ДМК Пресс
издательство

ЯЗЫК ГЕОГРАФИЧЕСКОЙ РАЗМЕТКИ

KML

ДЖОЗИ ВЕРНЕКЕ

Google

Джози Вернеке

Язык географической разметки KML

Josie Wernecke

The KML Handbook

Geographic Vizualization for the Web

Addison-Wesley

Джози Вернеке

Язык географической разметки KML

Москва, 2010

УДК 32.973.26-018.2
ББК 004.4
В31

В31 **Джози Вернеке**

Язык географической разметки KML. – М.: ДМК Пресс, 2010. – 288 с.: ил.

ISBN 978-5-94074-554-9

KML – язык разметки географических данных для отображения в таких программах, как Google Earth или Microsoft Virtual Earth. С его помощью можно представлять на трехмерной панораме Земли самую разнородную информацию – от фотоотчета о своем путешествии по Европе до динамической картины распространения птичьего гриппа или дрейфа материков.

Возможности ограничены только вашим воображением!

В книге подробно, на конкретных примерах описываются возможности версии KML 2.2, а также приводится справочник всех элементов языка.

УДК 32.973.26-018.2
ББК 004.4

Original English language edition published by Syngress Publishing, Inc. Copyright © by Syngress Publishing, Inc. All rights reserved.

Все права защищены. Любая часть этой книги не может быть воспроизведена в какой бы то ни было форме и какими бы то ни было средствами без письменного разрешения владельцев авторских прав.

Материал, изложенный в данной книге, многократно проверен. Но поскольку вероятность технических ошибок все равно существует, издательство не может гарантировать абсолютную точность и правильность приводимых сведений. В связи с этим издательство не несет ответственности за возможные ошибки, связанные с использованием книги.

ISBN 978-0-321-52559-8 (англ.)
ISBN 978-5-94074-554-9

Copyright © by Pearson Education, Inc., 2009
© Перевод на русский язык, оформление, издание, ДМК Пресс, 2010

Посвящается Байрон

Содержание

Предисловие ... 12

Введение ... 14
Для кого предназначена эта книга ... 15
Что нужно знать для чтения этой книги ... 15
Что вы найдете в этой книге ... 15
Запуск примеров ... 16
Типографские соглашения ... 16
Благодарности ... 17

Глава 1. Первое знакомство ... 19
KML – международный стандарт ... 19
Является ли спецификация KML полной? ... 20
Изобилие ресурсов ... 21
Создание и распространение KML-файлов ... 21
Изложите свой рассказ в виде KML ... 21
Представление неба в KML ... 25
Hello, Earth ... 26
Просмотр примеров из этой книги ... 27
Экспериментируйте! ... 27
Структура KML-файла ... 28
Определение простых и составных элементов ... 28
Общие правила языка KML ... 29
Техника копирования и вставки ... 29
Что дальше? ... 30

Глава 2. Метки и информационные окна ... 31
Пометка места ... 31
Настройка KML-презентации ... 31
Точечные метки ... 32
Информационное окно по умолчанию ... 33
Модификация информационного окна ... 34
Добавление текста ... 34
Оформление абзацев ... 35
Выделение курсивом ... 35
Выделение полужирным шрифтом ... 35
Добавление гиперссылок ... 35
Добавление изображений ... 36
Пример ... 36
Простой шаблон информационного окна ... 37
Изменение цвета фона ... 38

Задание цвета ... 39
Цветоподборщик в Google Earth ... 40
Внутренние стили ... 41
Изменение цвета текста ... 41
Исключение маршрута ... 42
Добавление нестандартного значка ... 42
Развернутый шаблон информационного окна ... 43
Задание точки обзора ... 44
Элемент LookAt ... 44
Поиск и исправление ошибок ... 46
Облет меток в программе Google Earth ... 47
Перелет к новой метке ... 49
Анкеры ... 49
Другие потомки элемента Placemark ... 50
Синтаксис элемента <Placemark> ... 50
Элементы, производные от Feature ... 50
Упаковка KML-файлов в KMZ-архив ... 51
Рекомендуемая структура файла ... 52
Примеры ... 53
Что дальше? ... 54

Глава 3. Геометрия ... 55
Предварительный обзор ... 55
Познакомимся с координатами поближе ... 56
Долгота ... 57
Широта ... 57
Высота и режим высоты ... 57
Ломаные ... 58
Тесселяция ломаной ... 59
Экструзия ломаной ... 60
Экспериментирование с элементом <altitudeMode> ... 61
Программа Google Maps ... 62
Элемент LineStyle ... 62
Многоугольники ... 63
Простой многоугольник ... 63
Элемент PolyStyle ... 65
Многоугольники с дырками ... 67
Пример ... 67
Элемент MultiGeometry ... 68
Модели ... 70
Местоположение ... 72
Ориентация ... 72
Вращение в трехмерном пространстве ... 73
Масштаб ... 73
Карта ресурсов ... 74

Включение информации об авторе и источнике 75
Что дальше? 76

Глава 4. Стили и значки 77
Предварительный обзор 77
Путешествие по литературным произведениям 78
Составные части стилей 79
Иерархия наследования стилей 79
Элемент LineStyle 80
Элемент PolyStyle 81
Элемент IconStyle 83
Элемент LabelStyle 88
Элемент BalloonStyle 90
Элемент ListStyle 93
Разделяемые и внутренние стили 96
Документы 96
Как задается URL стиля 98
Определение эффекта наката с помощью карт стилей 100
Синтаксис элемента <StyleMap> 101
Сила точки 101
Простой пример 101
Более сложный пример: другие способы применения карты стилей 103
Применение карты стилей для упрощения картинки 105
Определение стилей во внешнем файле 105
Что дальше? 106

Глава 5. Наложения 107
Предварительный обзор 107
Что такое наложение? 108
Инструменты для создания наложений 110
Основные концепции 110
Синтаксис элемента <Overlay> 111
Цвет и прозрачность наложений 112
Порядок рисования 112
Значки: задание и обновление накладываемого изображения 113
Другой способ задания точки обзора: камера 113
Различия между элементами <LookAt> и <Camera> 113
Рельефное наложение 118
Синтаксис элемента <GroundOverlay> 118
Элемент <LatLonBox> 119
Высота и режим высоты 120
Пример элемента <GroundOverlay> 121
Экранное наложение 122
Синтаксис элемента <ScreenOverlay> 123

Задание единиц измерения x/y для накладываемых изображений и значков ... 123
Потомки элемента <ScreenOverlay> ... 124
Пример простого экранного наложения ... 125
Пример экранного наложения с элементом <ListStyle> ... 126
Фотоналожение ... 126
Гигапиксельные фотографии ... 127
Синтаксис элемента <PhotoOverlay> ... 127
Основные понятия: форма и точка ... 127
Поле зрения ... 128
Дополнительный материал: пирамиды для гигапиксельных изображений ... 130
Как работает пирамида изображений ... 130
Создание пирамиды изображений ... 131
Пример фотоналожения ... 134
Что дальше? ... 135

Глава 6. Сетевые ссылки ... 136
Предварительный обзор ... 136
Что такое сетевая ссылка? ... 137
Сетевая ссылка как механизм группировки ... 138
Локальные и удаленные сетевые ссылки ... 138
Для чего нужны сетевые ссылки? ... 139
Основы клиент-серверной архитектуры ... 139
Настройка веб-сервера ... 140
Абсолютные и относительные ссылки на файлы ... 142
Синтаксис элемента <NetworkLink> ... 143
Пример простой сетевой ссылки ... 146
Обновление сетевых ссылок ... 147
Обновление по времени или при обнаружении изменений ... 147
Обновление по событиям просмотра ... 148
Более сложный пример: модель ветрового переноса вулканического пепла ... 148
Отправка информации от клиента серверу ... 152
Отправка информации о текущем виде ... 152
Отправка серверу дополнительных данных с помощью <httpQuery> ... 153
Пример обновления по событиям просмотра ... 154
Контроллеры сетевых ссылок ... 156
Синтаксис элемента <NetworkLinkControl> ... 157
Пример переопределения поведения сервером ... 159
Что дальше? ... 161

Глава 7. Динамический KML ... 162
Предварительный обзор ... 162
Основные понятия ... 162

Идентификаторы ... 163
Фрагменты URI ... 163
Обновление в KML ... 164
Общее описание механизма обновления ... 164
Синтаксис элемента <Update>, вложенного в <NetworkLinkControl> ... 166
Создание нового элемента с помощью <Update> ... 167
Изменение элемента с помощью <Update> ... 169
Удаление элемента с помощью <Update> ... 170
Пример обновления с помощью сценария ... 172
Время и анимация в KML ... 175
Задание времени ... 177
Всеобщее скоординированное время (UTC) ... 178
Примеры задания времени ... 178
Синтаксис элемента <TimeStamp> ... 179
Синтаксис элемента <TimeSpan> ... 179
Временные штампы и GPS-данные ... 180
Пример использования временных штампов с данными от GPS-приемника ... 180
Пример использования временных штампов с анимированными данными о метках ... 183
Временные интервалы и рельефные наложения ... 185
Пример использования временных интервалов для анимации рельефных наложений ... 186
Что дальше? ... 188

Глава 8. Большие наборы данных ... 189
Предварительный обзор ... 189
Основные понятия ... 190
Ограничивающий параллелепипед для регионов ... 191
Диапазон уровней детализации (LOD) ... 192
Диапазон затухания ... 195
Детальное описание элемента <Lod> ... 195
Как регионы соотносятся с режимом обновления фрагмента ... 196
Каскадное определение регионов ... 196
Суперналожения ... 196
Регионы ... 196
Синтаксис элемента <Region> ... 197
Пример рельефного наложения с регионом ... 198
Регион для трехмерной модели ... 200
Регион для двумерного наложения на ненулевой высоте ... 201
Развернутый пример: упрощение набора данных ... 202
Суперналожения ... 207
Простой пример сетевой ссылки с ассоциированным регионом ... 207
Пример суперналожения ... 209
Подготовка данных для суперналожения ... 209

Швейцарская система общественного транспорта 214
Расширенные данные 215
Какой подход выбрать? 215
Добавление произвольных XML-данных в элемент Feature 216
Добавление нетипизированных пар имя/значение 216
Использование элемента <BalloonStyle> в качестве шаблона 218
Подстановка компонентов для элементов расширенных данных ... 219
Добавление типизированных данных в элемент Feature 221
Пример подстановки компонентов 224
Что дальше? 226

Приложение А. Справочное руководство по KML 227

Приложение В. Астрономические данные в KML 275

Алфавитный указатель 282

Предисловие

Тем, кому доводилось взбираться на высокую гору, знаком тот волшебный миг, когда окружающая местность внезапно отступает и открывается вид на землю, раскинувшуюся далеко внизу. Вот такое чувство я испытываю, когда пишу это предисловие. Я оглядываюсь на растянувшееся на десять лет восхождение к вершинам технологии геобраузинга – сначала была идея, потом патент, потом компания-«стартап», и, наконец, эта технология стала частью повседневной жизни сотен миллионов людей. Я мысленно устремляю взор к возвышающимся вдалеке пикам, которые вы и другие разработчики, использующие KML, покорите, основываясь на сделанном нами. А еще, заглядывая к себе в душу, я понимаю, как десятилетие виртуального исследования нашей планеты изменило мое восприятие, сделало меня терпимее и заставило по-новому уважать космический корабль под названием «Земля» и его команду. И это самое главное для меня.

Опыт наглядно демонстрирует, что блуждание по географическим картам (геобраузинг) сродни личному присутствию – настолько, что пользователи продуктов Google Earth и Google Maps часто, увидев на карте свой дом и округу, замечают, как Томас С. Эллиот в стихотворении «Литтл Гиддинг»[1], что «увидели свой край впервые».

Покрытие всего мира, детальность изображения и отображение топографических особенностей местности – все это делает геобраузинг похожим на настоящее путешествие. Дополнительную притягательность придает плавность перемещения и свобода выбора области исследования. Объединенные в геобраузере, эти характеристики позволяют по-новому взглянуть на старую-престарую присказку: «если бы ты там был, то понял бы». Теперь вы можете легко «отправиться туда» в любой момент времени, если под рукой есть персональный компьютер или мобильный телефон, а когда «доберетесь», то увидите нужную информацию в пространственном контексте и сможете исследовать интересующую вас область с разных точек зрения. Наконец-то любой человек получил такие возможности узнавать, чувствовать и понимать, какие раньше давало только реальное путешествие.

Это и легло в основу амбициозной идеи языка KML, разрабатываемого Открытым геопространственным консорциумом (Open Geospatial Consortium), – предложить популярный, всеохватывающий, международный стандарт, позволяющий описать, «что» находится «там» и «тогда» в геобраузере. В различных

---

[1] Одно из стихотворений, вошедших в цикл «Четыре квартета» – *Прим. перев.*

главах этой книги подробно описываются многочисленные формы этого «что», в том числе точки – на поверхности Земли, над ней, под ней и даже в космосе, линии – для представления дорог, тропинок и границ, залитые и оконтуренные области, текст, изображения, трехмерные объекты, например здания и суда, а также механизмы и кодировки, позволяющие обмениваться этими данными.

В совокупности эти элементы образуют всесторонний язык разметки и инфраструктуру для аннотирования объектов на Земле и других планетах самой разнородной информацией, представляющей интерес для человечества. Можно было бы провести аналогию с веб-браузером и языком HTML, однако имеется существенная разница – не имея HTML-файла, веб-браузер покажет лишь пустую страницу, тогда как геобраузер даже без KML-файла дает детальное представление мира, которое можно исследовать, получая удовольствие. Недостает ему лишь аннотаций, превращающих планету в рассказчика.

Если мысль об отсутствии необходимых аннотаций – скажем, информации о текущей обстановке на дорогах, о погоде, о местах расположения банкоматов вашего банка, о тенденциях роста температуры воды вблизи коралловых рифов, об истории путешествия Шеклтона[1], обо всех местечках, упомянутых в каком-нибудь романе Джейн Остин или пьесе Шекспира, или об ареале распространения вируса H5N1 – не дает вам покоя и побуждает к действию, то язык KML и эта книга – для вас. Поскольку именно вы сумеете воспользоваться всей мощью геобраузинга и геовеб для создания отдаленного материка, едва различимого с той вершины, на которой мы стоим сегодня, материка, где информация будет способна спасти нашу планету, изменить политику, дать людям образование и улучшить жизнь. И я благодарен вам за ту роль, которую вы сыграете в использовании виртуального мира для изменения реального.

Майкл Т. Джонс
Chief Technology Advocate
Google

---

[1] Известный исследователь Антарктики – *Прим. перев.*

Введение

Умение «видеть географически» означает способность к целостному восприятию постоянно изменяющегося мира. Это способность проникать глубоко в суть проблем человечества и окружающей среды. Это умение находить баланс между глобальным и местным осмыслением. Оно открывает возможность охватить самые насущные проблемы современности: функционирование естественных систем Земли, все усложняющееся взаимодействие между человеком и окружающей средой, природу социальной организации общества с присущим ей неравенством и борьбой за власть над людьми и природой.

Из проспекта «Почему мы выбираем географию?»,
географический факультет Ливерпульского университета

Первые формальные знания по географии я получила на уроках м-ра Грейнджера в восьмом классе. И влюбилась в эту науку. Меня очаровывает многообразие способов графического изображения карт, и я до сих пор поражаюсь тому, сколь разнородную информацию можно представить на карте. По счастливому стечению обстоятельств, два года назад меня включили в группу, работающую в компании Google над проектом «KML». Это была удивительно интересная работа, к тому же не менее поучительная, чем год, потраченный в колледже на лекции, семинары и самостоятельные занятия. Аббревиатура KML означает Keyhole Markup Language (язык географической разметки), это простой текстовый формат, первоначально применявшийся в программе Google Earth (а теперь и в ряде других геобраузеров).

Настоящая книга представляет собой попытку поделиться знаниями, которые я почерпнула у специалистов в Google. Когда я присоединилась к проекту KML, группа состояла всего из двух инженеров: Бента Хагемарка (Bent Hagemark) и Майкла Эшбриджа (Michael Ashbridge), он же Мэш (Mash). Перед Бентом и Мэшем была поставлена задача превратить существовавший на тот момент язык KML в формальную XML-схему, разработать убедительные примеры, демонстрирующие хороший стиль кодирования, и довести язык до состояния, в котором он мог бы претендовать на звание международного стандарта. Я должна была создать сайт, посвященный KML, и доработать документацию. С этой задачей я успешно справилась, но осталось стойкое ощущение, будто я раскрыла только верхушку айсберга. Ну а теперь вашему вниманию предлагается сам айсберг.

В этой книге я стремилась также поведать о некоторых вдохновляющих работах, выполненных блестящими умами в разных странах. Многие из них, являясь

крупными специалистами в своих областях, ничего не знали о XML, KML, да и вообще об основах программирования. Осознав, что KML способен вдохнуть жизнь в сухие цифры, никому не известные названия мест и плоские карты, они потратили немало времени и сил на эксперименты, стремясь постичь логику, скрытую за форматом данных Google. Надеюсь, что, имея под рукой эту книгу, больше так мучиться не придется.

Для кого предназначена эта книга

Эта книга предназначена для тех, кто хочет знать, как создаются представления для таких геобраузеров, как Google Earth, но плохо знаком с программированием. Полезную информацию для себя найдут и «опытные пользователи», которым интересны более развитые возможности языка. Главы следуют друг за другом в порядке возрастания сложности материала.

Что нужно знать для чтения этой книги

Предполагается, что вы умеете создавать, сохранять и загружать файлы в компьютер и в веб-браузер и что вы подключены к Интернету. Описываются некоторые элементы HTML, используемые для создания меток, но детального рассмотрения языка HTML вы здесь не найдете. Если этот язык вам совсем не знаком, то, вероятно, имеет смысл обратиться к другим источникам. Для работы с KML знание языка XML необязательно, все необходимые сведения будут приведены в тексте.

Если вы захотите организовать сервер для размещения KML-файлов, упоминаемых в ссылках (глава 6), то потребуется скачать, установить и сконфигурировать программный пакет с каким-нибудь веб-сервером, например Apache или lighttpd. В главе 6 имеется кое-какая информация по этому поводу, но с деталями лучше ознакомиться, изучив прилагаемую к конкретному продукту документацию.

Что вы найдете в этой книге

В главе 1 «Первое знакомство» приводится обзор различных применений KML, начиная с простых наборов меток, позволяющих украсить и дополнить географической информацией блоги и сайты. Здесь же рассматривается простенький пример «Hello, Earth», иллюстрирующий основные части KML-файла.

В главе 2 «Метки и информационные окна» описывается, как создавать собственные значки и симпатичные стили информационных окон. Здесь приводится подробная информация о задании цветов в KML и о создании архивов в формате KMZ.

В главе 3 «Геометрия» речь пойдет о задании координат и режима вычисления высоты. Объясняются также такие методы машинной графики, как тесселяция и экструзия. Включены примеры и пояснения ко всем геометрическим элементам, в том числе и моделям. Показано также, как добавлять элементы, описывающие автора и источник KML-файла.

В главе 4 «Стили и значки» рассказывается, как использовать общие стили и как создавать все возможные типы подстилей: значок, метка, линия, многоугольник, информационное окно и список.

Глава 5 «Наложения» посвящена созданию наложений: экранных, рельефных и фотографических. Здесь же рассматривается вопрос о специальной обработке сверхбольших (гигапиксельных) фотографий и о том, как с помощью элемента Camera задать точку обзора.

В главе 6 «Сетевые ссылки» речь пойдет о размещении KML-файлов на веб-сервере, где их можно периодически обновлять или обрабатывать с помощью написанных пользователем программ. Приводится также введение в контроллеры сетевых ссылок – элементы, которые позволяют управлять некоторыми аспектами получения файла по сетевой ссылке.

В главе 7 «Динамический KML» приводятся подробные примеры работы с механизмом обновления, позволяющим создавать, изменять и удалять элементы KML-файлов, которые ранее были загружены по сетевой ссылке. Здесь же описываются элементы задания времени, позволяющие создавать эффекты анимации в KML-файле.

В главе 8 «Большие наборы данных» содержится важная информация о регионах и нестандартных типах данных. Регион – это мощный механизм управления условиями, при которых данная деталь (feature) становится видимой. Если вас интересует создание нестандартного шаблона стиля всплывающего окна для своей KML-презентации, обязательно прочитайте раздел «Подстановка компонентов для элементов расширенных данных».

Приложение А «Справочное руководство по KML» содержит алфавитный перечень всех элементов и типов, упоминаемых в стандарте KML, с краткими описаниями и сведениями о синтаксисе составных элементов. Здесь же описывается базовая структура KML-файла и принятые в языке соглашения.

В приложении В «Данные о небесной сфере в KML» рассказано, как отображать астрономические данные в геобраузере. Описывается также синтаксис располагающихся в начале KML-файла «подсказок», которые говорят браузеру о том, что файл содержит данные о небесной сфере, и задают способ преобразования небесных координат для отображения в Google Earth и других геобраузерах.

Запуск примеров

На странице informit.com/title/0321525590 размещены тексты всех примеров, встречающихся в этой книге. Щелкните по ссылке для любого примера, чтобы запустить программу Google Earth и посмотреть презентацию. Для ознакомления с KML-кодом воспользуйтесь техникой копирования и вставки, описанной в главе 1.

Типографские соглашения

Примеры кода набраны шрифтом Courier. Синтаксис составных элементов набран тем же шрифтом, а чтобы было проще отличить от примеров, такие фрагменты напечатаны на сером фоне. Элементы, рассматриваемые в данной главе, выделяются полужирным шрифтом.

Этим значком обозначаются примеры кода, которые размещены на странице informit.com/title/0321525590.

Имена элементов KML набраны обычным шрифтом и заключены в угловые скобки (например, <Placemark>, <NetworkLink>, <GroundOverlay>). Для удобства чтения имена элементов иногда записываются строчными буквами и даются в переводе, если такая небрежность не приводит к двусмысленности (например, метка, сетевая ссылка, рельефное наложение).

Благодарности

Для меня было большой честью работать с командой Google Earth, состоящей из умных, творческих и великодушных людей. Во время работы над книгой меня терпеливо учили, конструктивно критиковали и бескорыстно просвещали многие сотрудники Google.

Первым в списке следует поставить Бента Хагемарка. За добродушно-веселым, дружелюбным нравом и мягким стилем общения скрывается склонный к безукоризненно строгим рассуждениям интеллект самого высокого класса. Именно он научил меня почти всему, что я знаю о KML, и он же всегда был готов читать и перечитывать мой текст столько раз, сколько у меня хватало сил писать и переписывать его. Майкл Эшбридж тоже с радостью приходил на помощь, когда бы я ни попросила, выказывая при этом неизменное чувство юмора. Мано Маркс (Mano Marks), который влился в команду KML вскоре после меня и отвечал за поддержку внешних разработчиков, без задержек просматривал все черновики и помогал мне осознать потребности аудитории. Я никогда не рискнула бы взяться за эту книгу и уж тем более не смогла бы ее закончить, если бы не поддержка Бента, Мэша и Мано.

Большое спасибо также всем членам команды, работающей над программой Google Earth, в особенности Джону Рольфу (John Rohlf), Франсуа Байи (Francois Bailly), Бренту Остину (Brent Austin), Грегу Кумби (Greg Coombe), Райану Скрентону (Ryan Scranton), Питеру Бэрчу (Peter Birch), Майклу Вейсс-Малику (Michael Weiss-Malik), Брайну Макклендону (Brian McClendon) и Майклу Т. Джонсу (Michael T. Jones). Я также благодарна членам команды, работающей над программой Google Earth Outreach, особенно Ребекке Мур (Rebecca Moore) и Дженифер Фоулкс (Jenifer Foulkes), за то, что они помогли мне найти содержательные примеры практического применения KML.

Одной из самых восхитительных сторон этого проекта был поиск интересных применений технологии KML в Сети. Отдельное спасибо всем авторам, которые любезно разрешили включить в текст свой код и примеры. Хотя место и ограничено, я хотела бы отметить создателей некоторых из приведенных в книге примеров (в порядке появления): Памела Фокс (Pamela Fox), Мано Маркс, Джон Бэйли (John Bailey), Питер Уэбли (Peter Webley), Вулканическая обсерватория Аляски (Alaska Volcano Observatory), Институт Джейн Гудолл (Jane Goodall Institute), Американский мемориальный музей Холокоста (United States Holocaust Memorial

Museum), Энджел Телло (Angel Tello), Джером Бург (Jerome Burg), Брайан Флад (Brian Flood), Стефан Джинс (Stefan Geens), Деклан Батлер (Declan Butler), Валерий Хронусов и Рон Блэйки (Ron Blakey), Джеймс Стаффорд (James Stafford), Бент Хагемарк, Майкл Эшбридж, Коллекция карт Дэвида Рамси, Антонио Рочча Грака (Antonio Rocha Graca).

Писать книгу – трудная работа, в которой мне очень помогали поддержка и понимание со стороны группы документирования Google EngDocs. Отдельное спасибо Тине Орндафф (Tina Ornduff), с которой мы делили комнатку в течение всего времени работы над проектом и которая постоянно воодушевляла меня. Занимаясь нескончаемой, как нам тогда казалось, писаниной, мы часто вспоминали книгу Энн Ламотт «Bird by Bird» («Птица за птицей»).

Издательство Addison-Wesley подобрало группу квалифицированных рецензентов: Уоррен Кэлли (Warren Kelly), Стивен Кемп (Stephen Kemp), Даниэль Маккиннон (Daniel McKinnon), Дженнифер Минник (Jennifer Minnick) и Боб Евчук (Bob Yewchuk). Я высоко ценю то, как добросовестно и быстро вы прочитывали очередные главы и отправляли мне полезные критические замечания. Благодарю также моего редактора Грэга Донка (Greg Doench), который не утратил веру в меня, когда в самом начале проекта я выбилась из графика, а также Мишель Хаусли (Michelle Housley) и Элизабет Райан (Elizabeth Ryan), которые не давали всем нам сбиться с пути.

Еще я поднимаю тост за моих друзей, в особенности Присциллу Хосперс (Priscilla Hospers) и Джуди Кафлин (Judy Coughlin), и за свою семью: сыновей Джеффа и Эвана, невестку Карин, сестру Рут – все вы проявляли терпение и заинтересованность, а ваша помощь была мне необходима. И наконец, от всего сердца благодарю Байрона за то, что он разделил со мной это путешествие.

Глава 1. Первое знакомство

Прочитав эту главу, вы сможете:

- ✓ дать простое, достаточное для непрофессионала, определение языка KML;
- ✓ перечислить четыре возможных способа применения KML-презентаций;
- ✓ найти в Интернете KML-файлы по интересующей вас тематике, а затем просмотреть их в геобраузере;
- ✓ создать простой KML-файл и поделиться им с друзьями.

Язык географической разметки KML (Keyhole Markup Language) представляет собой диалект XML, предназначенный для отображения информации в географическом контексте. Если веб-браузеры считывают и отображают HTML-файлы, то геобраузеры, например Google Earth, проделывают то же самое с KML-файлами. KML – это текстовый формат, понятный человеку. KML-файл можно создать с помощью простейшего текстового редактора, сохранить и затем просмотреть в геобраузере. Для усвоения основ KML не нужно быть техническим гением. Очень скоро вы сможете создать впечатляющие презентации с нанесением своих географических данных и графики на глобальную палитру, предоставляемую многими популярными (и бесплатными) геобраузерами.

KML – международный стандарт

Язык KML был создан в 2001 году компанией Keyhole как формат данных для разработанного ей геобраузера Earth Viewer. С тех пор KML эволюционировал и теперь принят в качестве международного стандарта для визуального представления географической информации. Официально он называется OpenGIS KML 2.2 Encoding Standard (OGC KML) и относится к ведению консорциума Open Geospatial Consortium (www.opengeospatial.org/standards/kml/). В настоящее время в сети World Wide Web можно найти десятки миллионов общедоступных KML-файлов.

Для простоты в этой книге большинство примеров KML демонстрируется с помощью программы Google Earth, как показано на рис. 1.1. Однако KML широко поддержан и другими приложениями, в том числе Microsoft Virtual Earth, Microsoft WorldWide Telescope, NASA WorldWind, ESRI ArcGIS Explorer, Google Maps, Google Maps for mobile, Adobe PhotoShop, Autodesk AutoCAD и Yahoo! Pipes. Более того, перечень геобраузеров, картографических приложений и мобильных устройств, поддерживающих KML, постоянно расширяется. Не на каж-

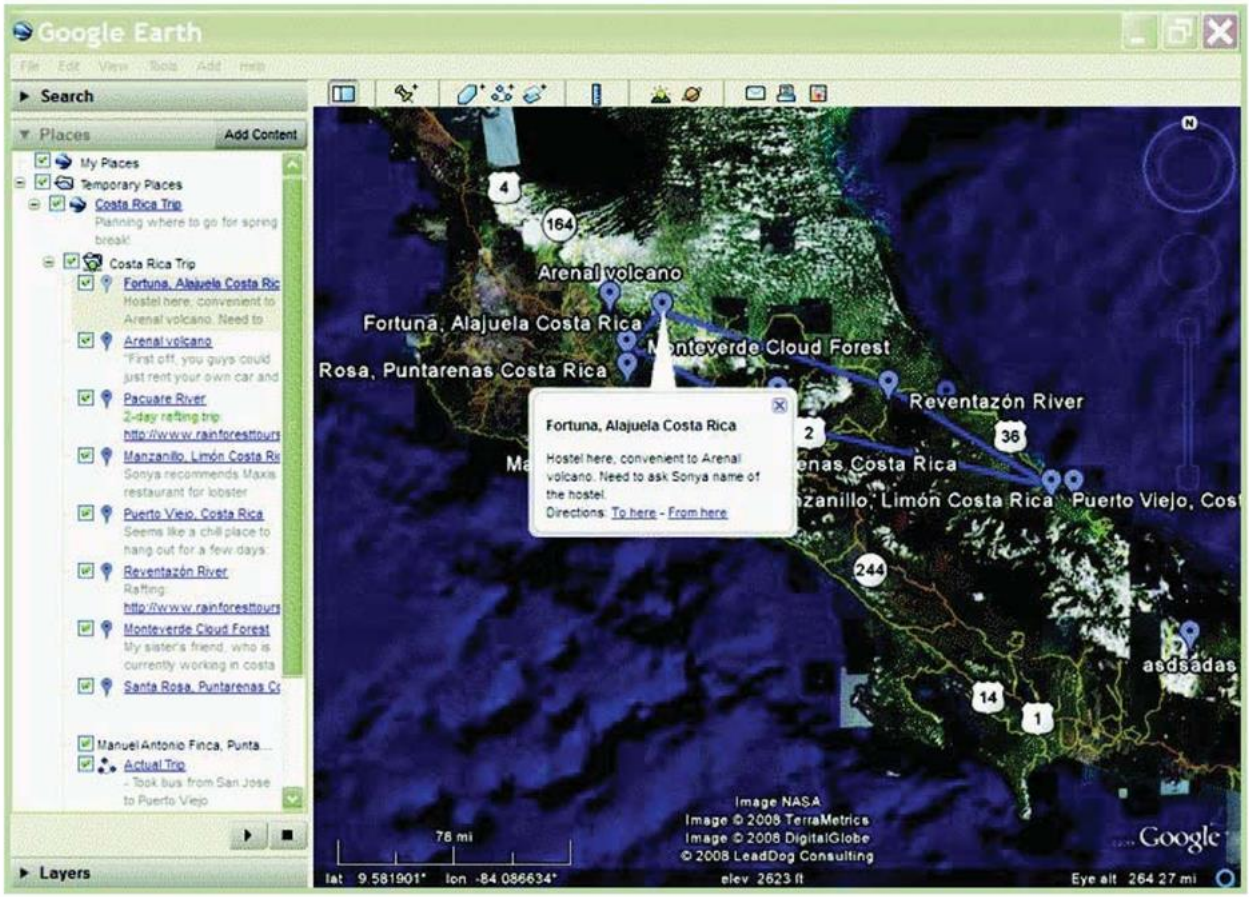


Рис. 1.1. Поделитесь своим опытом: путешествиями по городу или по миру, сведениями о местности, где вы проживаете, снятыми вами фотографиями. Синие значки обозначают предполагаемые остановки на маршруте по Коста-Рике. В информационных окнах приводятся советы путешественникам и ссылки на другие ресурсы. Этот файл изначально был подготовлен в программе My Maps компании Google – приложении для совместного создания двумерных карт, а затем импортирован в Google Earth (KML-файл создан Памелой Фокс)

дой платформе реализованы все возможности версии KML 2.2, поэтому если вы ставите перед собой конкретную цель, то не забывайте тестировать плоды своих трудов в предполагаемой системе или приложении. Между различными браузерами могут быть мелкие отличия, но основы KML всюду одинаковы. KML – это трехмерная система; обычно измерения в такой системе называются *длина*, *ширина* и *глубина*, но в данном контексте принято говорить о *долготе*, *широте* и *высоте*. Однако и двумерные приложения, например Google Maps и Google Maps для мобильных устройств, поддерживают подмножество KML.

Является ли спецификация KML полной?

Версия KML 2.2 полна, но спецификация KML продолжает развиваться и будет расширена под контролем консорциума Open Geospatial Consortium (OGC). Номер версии KML состоит из двух частей: главной и дополнительной (*majorVersion*.

minorVersion). Гарантируется, что версии с одной и той же главной частью совместимы. Официальное определение синтаксиса KML описывается *схемой* KML – формальным определением диалекта XML (см. www.opengeospatial.org/standards/kml/). Гарантируется, что текущая версия KML 2.2 будет полностью поддерживаться в схеме KML 2.3, которая сейчас разрабатывается. Дополнительную информацию о версиях KML см. в приложении А.

О будущем KML лучше всего узнавать на сайте OCG (www.opengeospatial.org/standards/kml/). Компании, предлагающие бесплатные геобраузеры, например Google и Microsoft, также предоставляют документацию по KML. На сайте этой книги (www.informit.com/title/9780321525598) периодически публикуется актуальная информация о последних разработках в области KML.

Изобилие ресурсов

Помимо официального сайта OGC и различных корпоративных сайтов, имеет смысл познакомиться с сообществом энтузиастов, ведущих информативные блоги, посвященные KML. Там вы найдете полезные советы, последние новости и фантастические примеры реального использования KML, которые послужат источниками знаний и вдохновения. Существующий уже довольно давно блог Google Earth Blog Фрэнка Тейлора (www.gearthblog.com) и блог Ogle Earth Стивена Джинса (www.ogleearth.com) – это только два примера ресурсов, содержащих богатейшую информацию по KML.

Создание и распространение KML-файлов

Создавать KML-файлы можно с помощью программы Google Earth, обладающей графическим интерфейсом специально для этой цели. А можно воспользоваться простым XML-редактором и вводить KML-код с нуля. KML-файлы модели и относящиеся к ним изображения можно упаковать в KMZ-архив, так чтобы весь материал находился в одном контейнере (см. главу 2). Если вы хотите поделиться своими KML или KMZ-файлами, то можете присоединить их к электронному письму в виде вложения, поместить в общую папку в домашней или корпоративной сети или выложить на веб-сервер. После того как вы правильно сконфигурируете веб-сервер и опубликуете адрес своего KML-файла, любой человек, установивший Google Earth (или какое-нибудь совместимое приложение), сможет созданный вами файл просмотреть.

Изложите свой рассказ в виде KML

В сообществе пользователей KML есть люди с самыми разными интересами и знаниями:

- случайные пользователи, которые создают KML-файлы, чтобы показать расположение своего дома, документировать свои путешествия и спланировать пеший или велосипедный поход;
- студенты и преподаватели, применяющие KML для изучения людей, мест и событий, как исторических, так и текущих;

- профессионалы в области недвижимости, архитекторы, агентства по развитию городской среды, которые используют KML для описания предлагаемых конструкций и визуализации планов;
- ученые, для которых KML является средством нанесения на карту ресурсов, моделей и тенденций, например извержений вулканов, погодных явлений, активности земной коры и месторождений полезных ископаемых (рис. 1.2);
- различные организации, например Национальное географическое общество, ЮНЕСКО или Смитсоновский институт, применяют KML для отображения поддерживаемых ими обширных наборов глобальных данных.

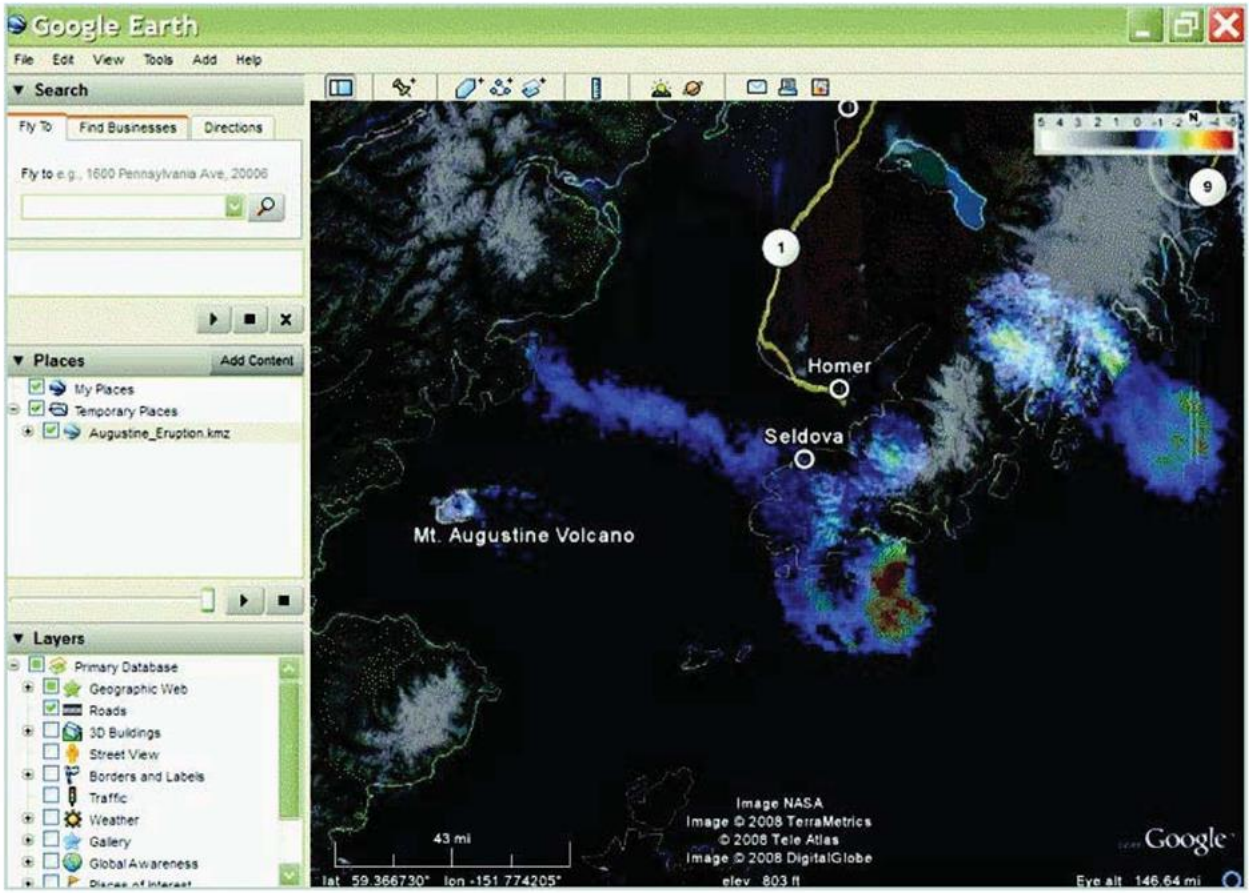


Рис. 1.2. Данные можно осмысленно представить. На этом рисунке показано, как Вулканическая обсерватория Аляски использует Google Earth для наложения на карту пеплового шлея, возникшего в результате извержения вулкана Августин. Цветом представлены температурные данные (фото публикуется с разрешения Джона Е. Бэйли, суперкомпьютерный центр арктического региона, Фэрбенкс, Аляска)

Вы можете использовать KML для нанесения собственных меток, геометрических данных, аннотаций и изображений на графическую основу, предоставляемую Google Earth. Если вы храните свои KML-файлы на сервере, то можно даже обновлять представление на компьютерах пользователей периодически или по мере изменения данных (см. обсуждение сетевых ссылок в главе 6). Файлы, размещенные на общедоступных серверах, индексируются поисковыми системами, чтобы пользователи веб могли их найти (рис. 1.3).

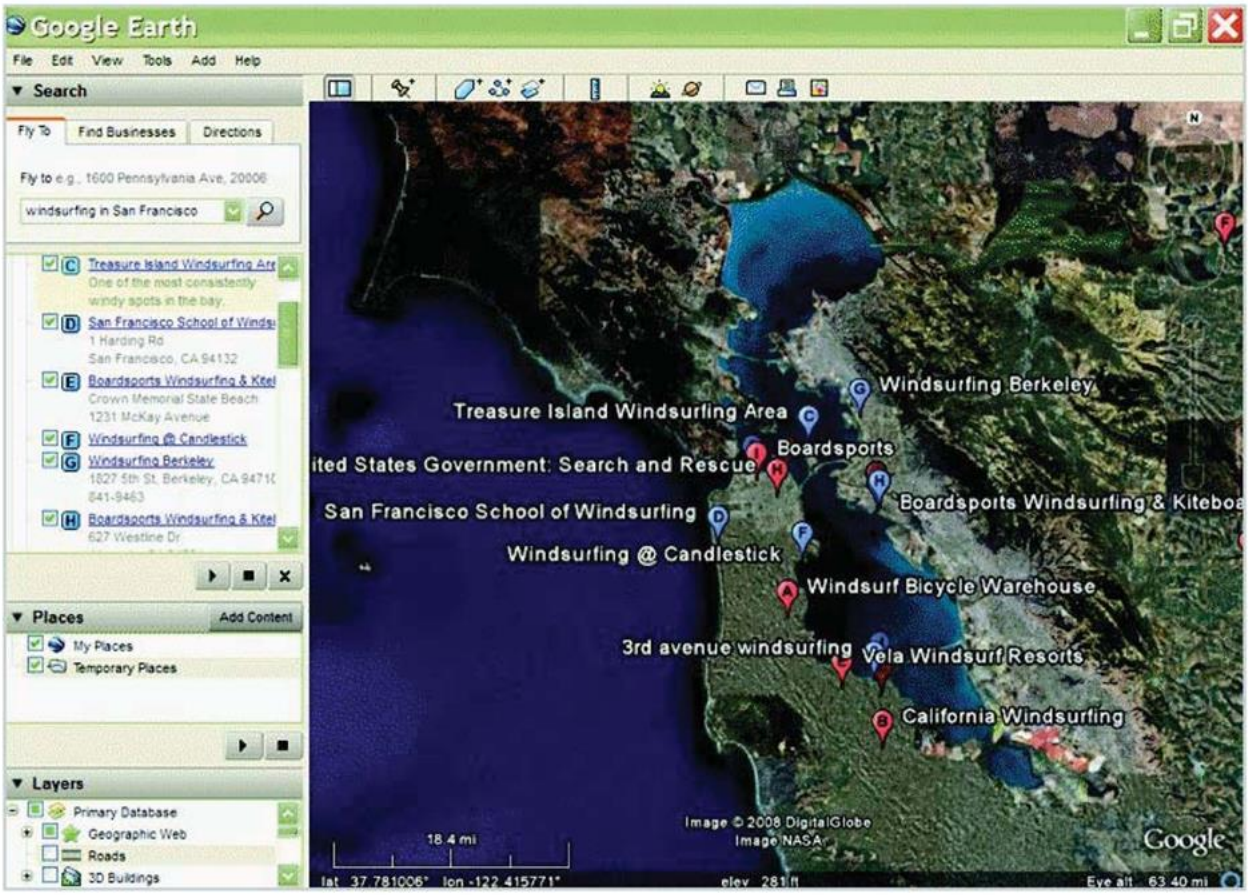


Рис. 1.3. Ищите информацию по конкретной теме или месту. На этом рисунке показано, как Google Earth выдает информацию о местах в окрестностях Сан-Франциско, где можно заниматься виндсерфингом

Можно без труда персонализировать свои KML-презентации, воспользовавшись стилями для описания значков, всплывающих информационных окон, цветов, линий, форм и надписей. KML позволяет также отображать определенные детали в зависимости от момента времени в заданном интервале и изменять отображение в соответствии с выбранным пользователем масштабом, увеличивая уровень детализации, когда зритель приближает изображение (рис. 1.4 и 1.5).

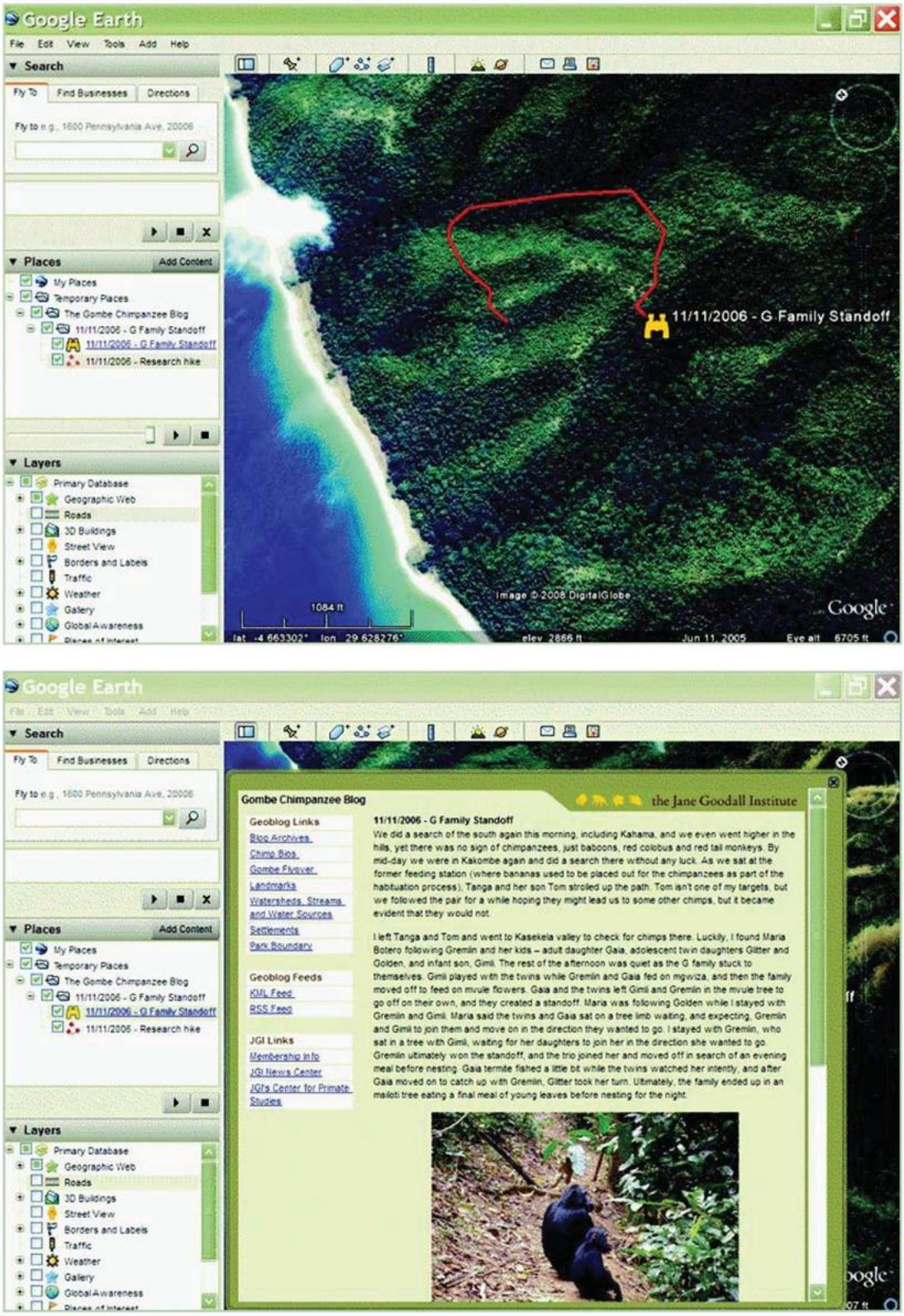


Рис. 1.4. Исследуйте мир... не вставая со стула. На верхнем рисунке показан путь семейства шимпанзе, которое изучали в Институте Джейн Гудолл в Национальном парке Гомбе в Африке. Щелчок по названию открывает описание, в котором приведена подробная информация о поведении животных в этот день (публикуется с разрешения Института Джейн Гудолл: http://gombeblog.janegoodall.org)

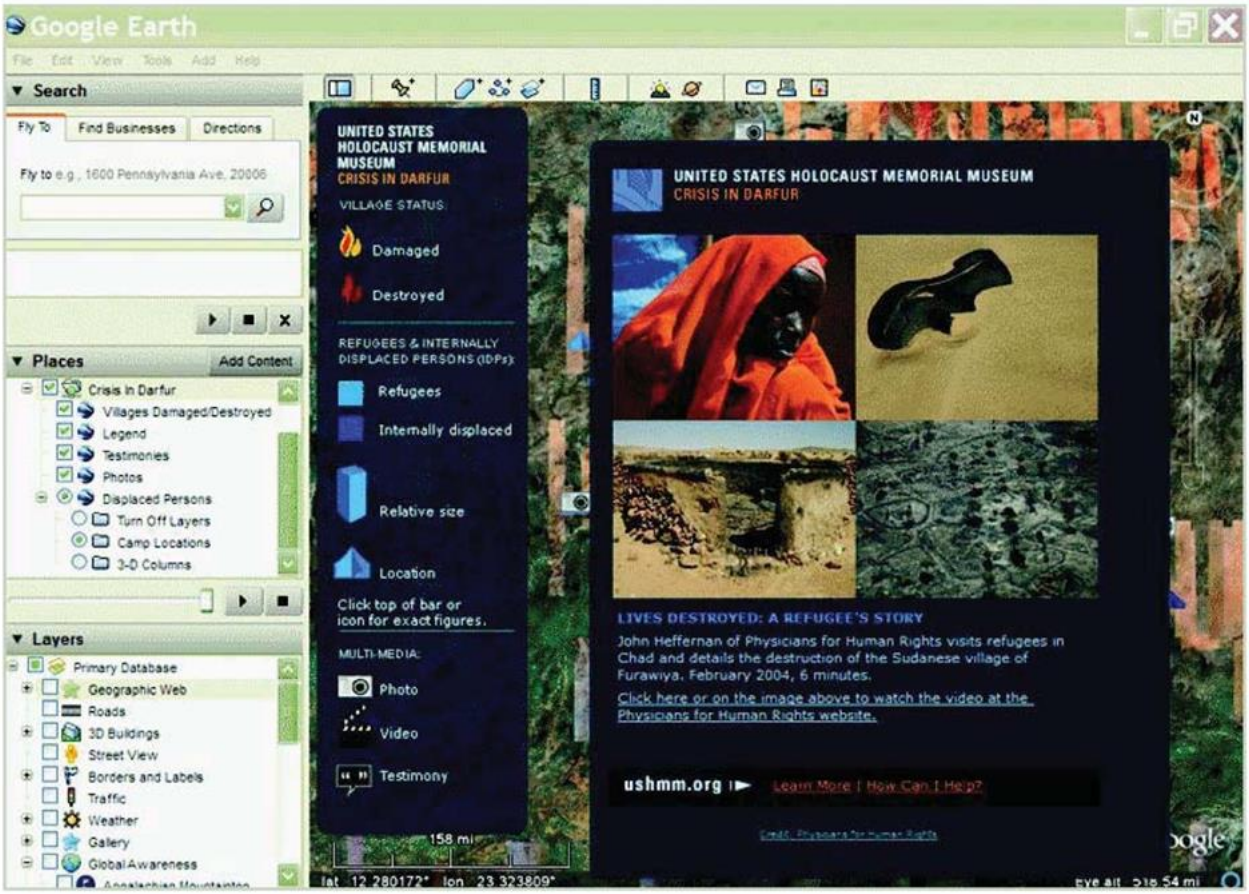


Рис. 1.5. Углубляйте свои представления. Специальные проекты, например инициатива Американского мемориального музея Холокоста, посвященная кризису в Дарфуре, призваны привлечь внимание мировой общественности к личным и глобальным трагедиям. Для этого публикуются фотографии, рассказы очевидцев и ссылки на видео – все в географическом контексте (публикуется с разрешения Американского мемориального музея Холокоста: www.ushmm.org)

Представление неба в KML

KML 2.2 поддерживает представление не только наземных, но и астрономических данных (рис. 1.6). Если поместить в начале KML-файла специальную подсказку (hint=»target=sky»), то браузер будет интерпретировать данные иначе, проецируя их на виртуальную небесную сферу, окружающую Землю. В режиме неба камера Google Earth направлена вверх – в небеса, а не вниз – на Землю. С точки зрения автора KML-файла, основное отличие заключается в необходимости выполнить определенные математические операции для преобразования астрономических координат (*прямое восхождение* и *склонение*) в земные (*долготу* и *широту*). Все остальное в KML работает одинаково в режиме неба и в режиме Земли.

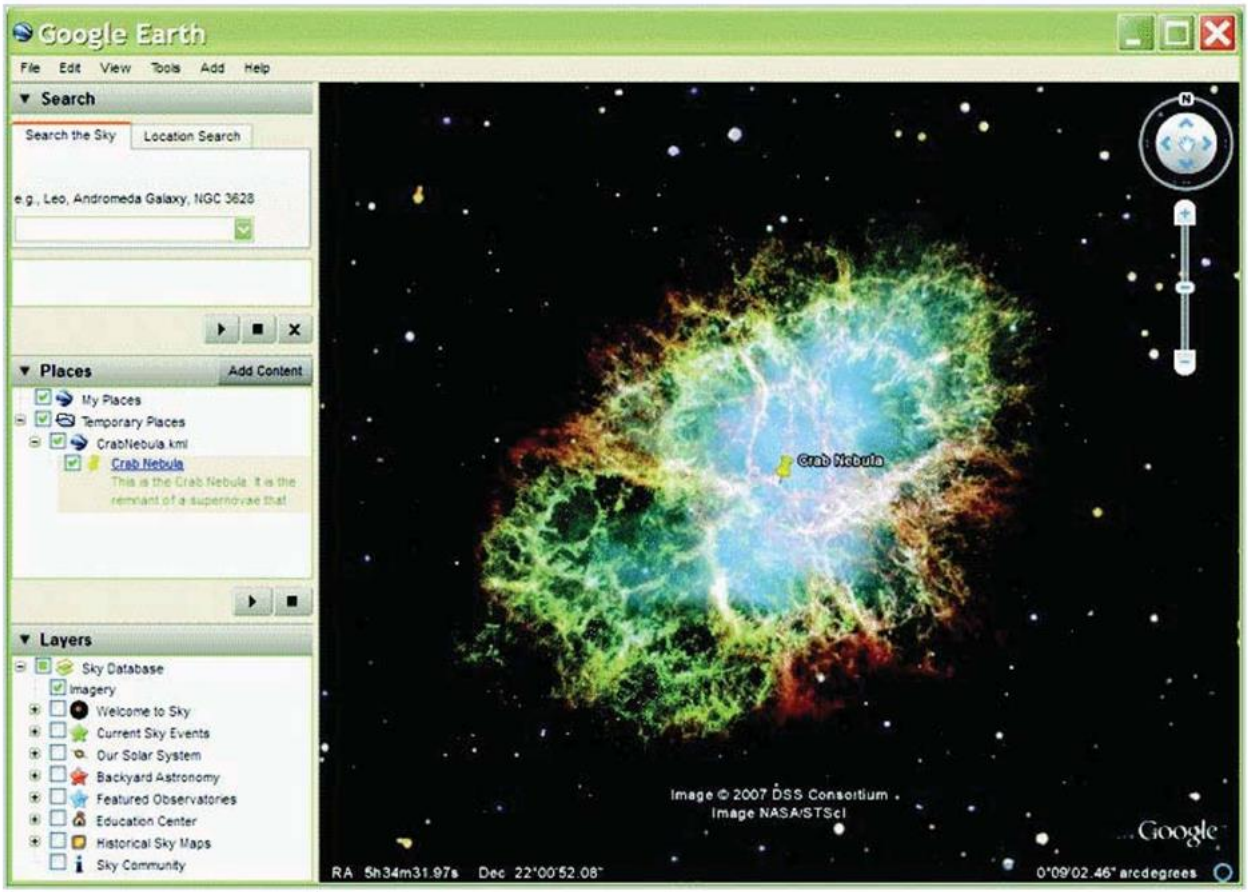


Рис. 1.6. Путешествие в космосе. Этот KML-файл содержит метку и значок, соответствующие знаменитой планетарной туманности. Чтобы отобразить небесные данные в геобраузере, включите в файл специальную подсказку и преобразуйте астрономические координаты в земные (см. приложение B)

Hello, Earth

Стандартная метка отображается в программе Google Earth в виде значка желтой канцелярской кнопки, «воткнутой» в определенную точку на земной поверхности. Обычно у метки имеется *имя*, идентифицирующее эту точку. Считается хорошим тоном включать также и *описание*. Описание отображается в результатах поискового запроса и помогает пользователю решить, нужен ли ему данный KML-файл.

В следующем примере мы создадим простую метку с именем «Hello, Earth». В описании содержится дополнительная информация о месте (рис. 1.7).

HelloEarth.kml

```
<?xml version="1.0" encoding="utf-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Placemark>
    <name>Hello, Earth</name>
    <description><![CDATA[Вот здесь мы разработали программу Google Earth!]]></description>
    <Point>
      <coordinates>
```

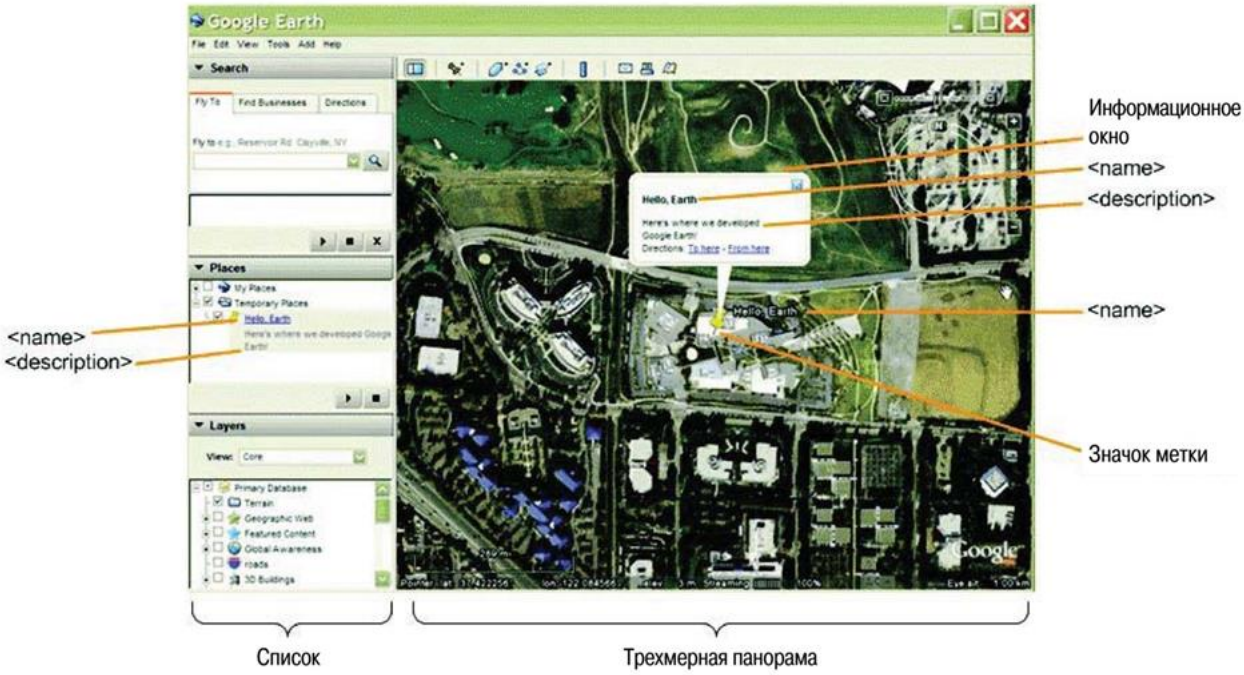


Рис. 1.7. Анатомия метки. Обычно метка сопровождается именем и описанием. Если у вас есть что рассказать, то во всплывающем окне будут располагаться относящийся к метке текст, изображения и ссылки на другие ресурсы в веб

```
        -122.084583,37.42227,0
      </coordinates>
    </Point>
  </Placemark>
</kml>
```

Просмотр примеров из этой книги

Чтобы просмотреть этот пример в программе Google Earth, сначала необходимо скачать и установить саму программу. Скачать ее можно бесплатно с сайта http://earth.google.com. Полные листинги примеров находятся на сайте этой книги по адресу www.informit.com/title/9780321525598. Для просмотра примера в Google Earth дважды щелкните по соответствующему ему файлу.

Экспериментируйте!

Если вам захочется поэкспериментировать самостоятельно, можете набрать текст примера в любом редакторе, который сохраняет файл, не добавляя информацию о форматировании, например в Блокноте (Notepad). При сохранении укажите имя файла с расширением *.kml* и откройте его в программе Google Earth на своем компьютере. Сохраненный файл можно впоследствии отредактировать. Например, попробуйте изменить содержимое тега <name>, еще раз сохраните файл и снова откройте его в Google Earth. Затем измените значения в элементе <coordinates>, сохранитесь и посмотрите, где окажется новая метка.

Самый лучший способ изучить KML – экспериментировать с имеющимися файлами, меняя разные значения и наблюдая за результатами в геобраузере. Если браузер ничего не показывает, значит, вы где-то допустили ошибку. В программе Google Earth есть механизм проверки ошибок, который, возможно, будет вам полезен. (Выберите из меню пункт **Инструменты** ⇒ **Настройки** и на вкладке **Общие** в разделе **Обработка ошибок KML** отметьте переключатель **Показывать сообщения об ошибках**[1].) Кроме того, для проверки своего KML-кода можете воспользоваться какой-нибудь специализированной программой. Одна такая программа, разработанная компанией Galdos Systems, имеется на сайте www.kmlvalidator.com.

Структура KML-файла

В начале любого KML-файла должны располагаться следующие две строки:

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
```

Если вы создаете KML-файл с нуля, обязательно поместите их в самое начало файла точно в таком виде, как показано выше. KML-файл может содержать только один элемент <kml>. Не забудьте про закрывающий тег </kml> в конце файла.

Файл из нашего примера содержит один элемент <Placemark> с тремя потомками. Имена элементов заключаются в угловые скобки < >:

Потомки элемента <Placemark>

<name>	Текст надписи, ассоциированной с меткой
<description>	Текст (и, возможно, изображения), содержащий дополнительную информацию о метке. Содержимое элемента <description> отображается в *информационном окне* (balloon). Это окно появляется, когда пользователь щелкает по имени метки на панели **Метки** или по значку метки в главном окне программы Google Earth, где представлена трехмерная панорама

На рис. 1.7 показано, как выглядят имя и описание на трехмерной панораме и на панели **Метки**.

Поскольку формат KML является частным случаем XML, то его структура подчиняется общим правилам XML. Любой элемент начинается с имени, заключенного в угловые скобки (<Placemark>). Элемент заканчивается тегом, в котором имени предшествует символ косой черты. Между этими ограничителями находится значение элемента.

Определение простых и составных элементов

В KML любое слово, заключенное в угловые скобки < >, считается *элементом*. Элементы, чьи имена начинаются с заглавной буквы, являются *составными*, то

[1] Поскольку программа Google Earth русифицирована, здесь и далее названия элементов интерфейса приводятся по-русски в соответствии с официальной версией – *Прим. перев.*

есть могут содержать внутри себя другие элементы. Например, во фрагменте ниже <Point> – составной элемент, в который *вложен* элемент <coordinates>:

```
<Point>
  <coordinates>-122.084583,37.42227,0</coordinates>
</Point>
```

Имена *простых элементов* начинаются со строчной буквы. Простые элементы не могут содержать внутри себя других элементов. Простой элемент содержит только *символьные данные* (в терминологии XML так называются буквы, цифры и символы, которые не используются для разметки). В файле HelloEarth.kml простыми являются элементы <name>, <description> и <coordinates>.

Составные элементы также называют *родительскими*, поскольку они содержат другие элементы. Простые же элементы называют *дочерними*, или *потомками*. В KML-файле принято записывать потомков с отступом от родителя на несколько пробелов, но это лишь соглашение для удобства восприятия человеком. Геобраузер на отступы не обращает никакого внимания.

Общие правила языка KML

При составлении KML-файлов следует помнить о нескольких общих правилах:

- регистр букв имеет значение. Каждый элемент должен записываться точно так, как указано в справочном руководстве по KML 2.2, с точностью до регистра (см. приложение А);
- порядок следования имеет значение. Элементы-потомки должны располагаться внутри своего родителя в том порядке, который описан в справочном руководстве по KML 2.2. Некоторых потомков можно опускать, но изменять порядок следования не разрешается;
- потомки могут встречаться только внутри тех родительских элементов, где они допустимы. Если вы будете придерживаться перечисленных в справочном руководстве синтаксических правил для конкретных элементов, то все будет нормально.

Техника копирования и вставки

Если вы хотите просмотреть KML-разметку какой-нибудь папки или метки, представленной в Google Earth, можете скопировать ее и затем вставить в текстовый редактор, например в Блокнот. (Интуитивно не очевидно, что можно скопировать графическую деталь из панорамы Google Earth, но при вставке в редактор она преобразуется в соответствующий фрагмент KML-разметки. Попробуйте сами – работает!) Чтобы посмотреть на KML-разметку какой-нибудь видимой детали, например метки Placemark, рельефного наложения GroundOverlay (изображения, накладываемого на рельефную основу), многоугольника Polygon (формы) или отрезка LineString (пути), выполните следующие действия:

1. Подведите курсор к интересующей вас детали на трехмерной панораме (или на панели **Метки**), чтобы выделить ее.

2. Щелкните правой кнопкой мыши и выберите из контекстного меню команду **Копировать**.
3. Откройте текстовый редактор и вставьте содержимое буфера обмена, например выбрав из меню команду **Правка** > **Вставить**.
   В редакторе появляется KML-разметка выбранной детали. (Напоминаю, что редактор не должен записывать в файл никакой дополнительной информации о форматировании.)
4. Сохраните файл, указав расширение *.kml* (например, *myHouseInPhila.kml*).

Что дальше?

В следующей главе вы узнаете о двух основных элементах KML – <Placemark> и <description>. Хотя создавать метки и всплывающие описания можно с помощью графического интерфейса пользователя, встроенного в геобраузер, в главе 2 рассказывается о том, как модифицировать KML-файл для достижения специальных эффектов, что открывает путь к эффективному созданию целых сайтов с единообразным внешним обликом. Мы поговорим также о том, как упаковать KML-файлы в KMZ-архив, чтобы ими было удобно обмениваться и публиковать в веб в виде единого компонента.

Глава 2. Метки и информационные окна

Прочитав эту главу, вы сможете:

- ✓ создать метку, описание которой содержит HTML-разметку;
- ✓ снабдить метку нестандартным значком;
- ✓ задать для метки внешнюю точку обзора с помощью элемента <LookAt>;
- ✓ создать набор меток и задать маршрут перемещения от одной метки к другой.

Пометка места

Элемент <Placemark> служит для отметки точки на земной поверхности. Простую метку, включая имя, описание, точку обзора и местоположение, можно создать с помощью графического интерфейса таких программ, как Google Earth или Google Maps. Но в этой главе мы пойдем дальше и покажем, как создается KML-файл, в котором с метками и маршрутами ассоциированы специальные значки, более стилизованные информационные окна и нестандартные эффекты.

Из примера «Hello, Earth» в главе 1 вы узнали о простых элементах, которые образуют информационное окно по умолчанию. Это окно вы получаете «даром», не создавая соответствующий элемент явно. Если же вы захотите настроить информационное окно, то придется создать элемент <BalloonStyle>, в который будут вложены элементы для задания цвета текста и фона, а также текстовые элементы. Обо всем этом рассказано в разделе «Изменение цвета фона» ниже в этой главе.

Настройка KML-презентации

Примером информативного и привлекательного стиля информационных окон для KML может служить блог Института Джейн Гудолл, в котором рассказывается о наблюдениях за жизнью шимпанзе в Национальном парке Гомбе в Африке. На рис. 2.1 представлены метки и заголовки различных статей в блоге на сайте www.janegoodall.org/gombe-chimp-blog/. Здесь используются два нестандартных значка. Значок шимпанзе ведет на страницу с данными об отдельных особях, а значок бинокля – на статьи в блоге натуралиста Эмили Вроблевски. На рис. 2.2 показан нестандартный стиль информационного окна, ассоциированный со всеми метками статей на данном сайте. В этом стиле для позиционирования двух текстовых колонок применена HTML-таблица и, кроме того, заданы цвета текста и фона. Демонстрируется также использование ссылки на изображение (для добавления фотографии) и ссылок на другие веб-страницы.

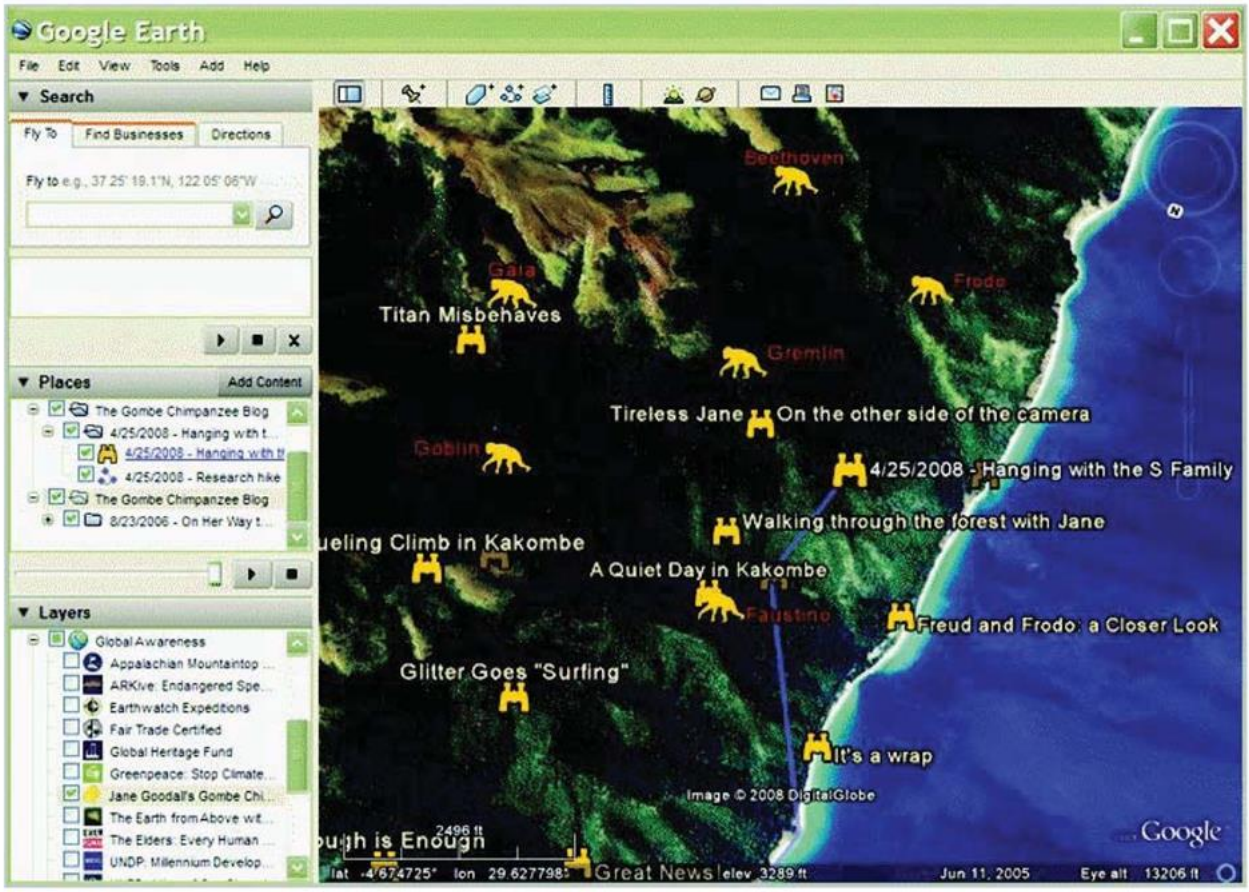


Рис. 2.1. Метки с нестандартными значками. В этом примере значок бинокля обозначает статьи в блоге, а значок шимпанзе – данные об отдельных животных (публикуется с разрешения Института Джейн Гудолл: www.janegoodall.org/gombe-chimp-blog/)

Точечные метки

То, что пользователь воспринимает как метку в программе Google Earth, на самом деле является элементом <Placemark>, в который вложен элемент <Point>. Такая точечная метка – единственный способ поместить значок и надпись на трехмерной панораме. По умолчанию в качестве значка выступает уже знакомая нам желтая канцелярская кнопка. В языке KML элемент <Placemark> может содержать один или несколько геометрических элементов, например <LineString>, <Polygon> или <Model>. Но лишь с элементом <Placemark>, в который вложен <Point>, ассоциируются значок и надпись. Значок обозначает местоположение точки.

Геометрические элементы внутри <Placemark> необязательны, но если они отсутствуют, то всплывающее окно оказывается в углу панорамы. На трехмерной панораме в Google Earth точечная метка является единственным объектом, по которому можно щелкнуть мышью и который демонстрирует эффект наката. Прочие геометрические элементы не представлены значками на панораме. Однако вы можете сопоставить геометрическому элементу значок, если воспользуетесь эле-

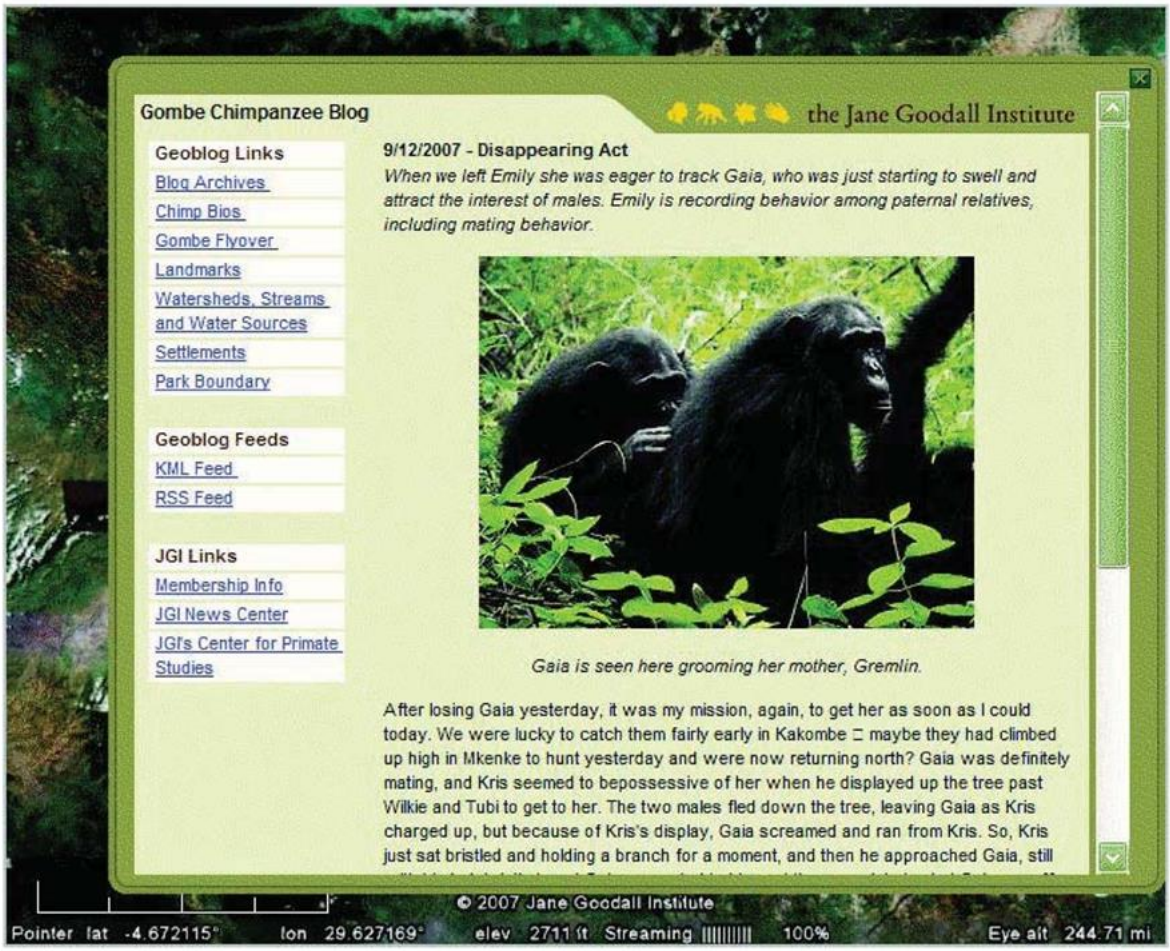


Рис. 2.2. Привлекательный стиль информационных окон. В этой презентации, подготовленной Институтом Джейн Гудолл, используется нестандартный стиль с двухколонной версткой, специальными шрифтами и цветами, гиперссылками и изображениями (публикуется с разрешения Института Джейн Гудолл; www.janegoodall.org)

ментом <MultiGeometry>, который содержит как <Point>, так и какой-нибудь другой геометрический элемент. (Дополнительную информацию о геометрических элементах с примерами использования см. в главе 3.)

Информационное окно по умолчанию

В этом разделе мы разложим простую метку на составные части, обращая особое внимание на то, как можно изменить внешний вид информационного окна и определить дополнительные стилевые элементы. По умолчанию всплывающее окно в Google Earth состоит из следующих компонентов:

- имя <name>, отображаемое полужирным шрифтом;
- описание метки <description>;
- ссылки на маршруты следования: *Сюда – Отсюда*.

По умолчанию фон окна белый, а «хвостик» указывает на точку с координатами метки, если элемент <Placemark> включает элемент <Point>.

Маршруты бывают полезны при описании путешествий по местности, где есть дороги, а люди пользуются автомобилями или другими транспортными средствами. Но если вы помечаете место для стоянки в Гималаях или места обитания шимпанзе в джунглях, то оставлять указания, как туда доехать, бессмысленно. О том, как опустить этот элемент, см. ниже в разделе «Исключение маршрута». Если элемент <Placemark> не содержит элемента <Point>, то маршрут автоматически опускается.

Модификация информационного окна

В этом разделе приводятся пошаговые инструкции о том, как модифицировать внешний вид информационного окна для некоторой метки.

Добавление текста

Элемент <description>, вложенный в <Placemark>, поддерживает подмножество языка HTML, описанное в справочном руководстве по KML 2.2 (см. приложение А). Если вы решите воспользоваться HTML, то проще всего окружить HTML-код скобками CDATA, как показано ниже в примере *SimpleTextBalloon.kml*.

Такая техника позволяет скрыть HTML от синтаксического анализатора KML. Если теги CDATA опустить, то придется экранировать угловые скобки (то есть подставлять специальные последовательности символов вместо спецификаторов HTML-разметки). Так, без CDATA надо было бы писать `<` вместо `<` и `>` вместо `>`. Следовательно, `<p>` превратилось бы в `<p>`. Такая нотация гораздо менее удобна как для записи, так и для чтения. В отсутствие CDATA аналогичные экранирующие последовательности нужно было бы употреблять вместо всех специальных символов HTML-разметки.

SimpleTextBalloon.kml

```
<?xml version="1.0" encoding="utf-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Document>
    <Placemark>
      <name>9/12/2007 - Феномен исчезновения</name>
      <description>
        <![CDATA[
          <p>Потеряв вчера Гайю, я сегодня должна была как можно скорее найти ее.
          К счастью, мы довольно быстро натолкнулись на них в Какомбе.</p>

          <p>Оба самца спустились с дерева, оставив Гайю в компании возбужденного
          Криса, но Гайя закричала и сбежала от Криса.</p>

          <p>К сожалению, расставшись с деревом ироко, они направились к водопаду,
          где безо всяких усилий вскарабкались по вьющимся растениям и буквально за
          несколько секунд оказались выше водопада.</p>
        ]]>
      </description>
      <Point>
        <coordinates>18.615106,-4.841399,0</coordinates>
      </Point>
    </Placemark>
  </Document>
</kml>
```

Оформление абзацев

Только что вы видели, как HTML-тег используется для вставки разрыва строки и промежутка между абзацами описания. Закрывается абзац тегом . Например:

```
<description>
  <![CDATA[
    <p>Оба самца спустились с дерева, оставив Гайю в компании возбужденного Криса,
    но Гайя закричала и сбежала от Криса.</p>

    <p>К сожалению, расставшись с деревом ироко, они направились к водопаду,
    где безо всяких усилий вскарабкались по вьющимся растениям и буквально за
    несколько секунд оказались выше водопада.</p>
  ]]>
</description>
```

Чтобы разорвать строку без добавления пустого промежутка, воспользуйтесь тегом <br/>.

Выделение курсивом

Чтобы выделить некоторые слова курсивом, используйте HTML-теги и :

```
<description>
  <![CDATA[
    <p><em>Потеряв вчера Гайю, я сегодня должна была как можно скорее найти ее.
    К счастью, мы довольно быстро натолкнулись на них в Какомбе.</em></p>

    <p>Оба самца спустились с дерева, оставив Гайю в компании возбужденного Криса,
    но Гайя закричала и сбежала от Криса.</p>

    <p>К сожалению, расставшись с деревом <em>ироко</em>, они направились к водопаду,
    где безо всяких усилий вскарабкались по вьющимся растениям и буквально
    за несколько секунд оказались выше водопада.</p>
  ]]>
</description>
```

Выделение полужирным шрифтом

Для выделения части текста полужирным шрифтом служат HTML-теги и :

```
<description>
  <![CDATA[
    <b>Институт Джейн Гудолл</b>
  ]]>
</description>
```

Добавление гиперссылок

Гиперссылками называются ссылки на другие ресурсы в веб. Для их добавления в описание используются стандартные HTML-теги. Когда пользователь щелкает по ссылке, страница с указанным в ней адресом (URL) открывается в новом окне браузера. Например:

```
<description>
  <![CDATA[
    <a href="http://www.discoverchimpanzees.org">
    Центр ИДГ по изучению примамов</a>
  ]]>
</description>
```

Текст гиперссылки располагается между тегами <a> и . Атрибут `href` открывающего тега <a> заключается в кавычки и содержит URL-ссылки. Значение тега <a> (строка, находящаяся между открывающим и закрывающим тегами элемента) содержит текст, который отображается в описании.

Добавление изображений

Чтобы добавить в информационное окно изображение, применяется HTML-тег <img>. Например:

```
<description>
  <![CDATA[
    <img src="janeGoodall.jpg" />
  ]]>
</description>
```

В данном случае файл с изображением находится в том же каталоге, что и KML-файл. Но ссылкой может быть также URL, ведущий на изображение, размещенное в произвольном месте веб (например, http://someServer/someDirectory/myPhoto.jpg). Помимо *.jpg*, поддерживаются графические форматы *.png*, *.tif* и *.gif*.

Пример

В файле *SimpleTextBalloon.kml* находится пример KML-кода, демонстрирующий применение в описании разрывов строк, выделения курсивом и полужирным шрифтом, гиперссылки и изображения:

SimpleTextBalloon.kml

```
<?xml version="1.0" encoding="utf-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
.
.
.
  <Placemark>
    <name>9/12/2007 - Феномен исчезновения</name>
    <description>
      <![CDATA[
        <p><em>Потеряв вчера Гайю, я сегодня должна была как можно скорее найти ее.
        К счастью, мы довольно быстро натолкнулись на них в Какомбе.</em></p>

        <p>Оба самца спустились с дерева, оставив Гайю в компании возбужденного Криса,
        но Гайя закричала и сбежала от Криса.</p>

        <p>К сожалению, расставшись с деревом ироко, они направились к водопаду,
        где безо всяких усилий вскарабкались по вьющимся растениям и буквально
        за несколько секунд оказались выше водопада.</p>
        <p>
        <a
      href="http://www.discoverchimpanzees.org>Центр ИДГ по изучению примамов</a></p>
        <br />
        <b>Институт Джейн Гудолл</b>
```

```
      ]]>
    </description>
    <Point>
      <coordinates>18.615106,-4.841399,0</coordinates>
    </Point>
  </Placemark>
</kml>
```

На рис. 2.3 показано, как этот пример выглядит в программе Google Earth.

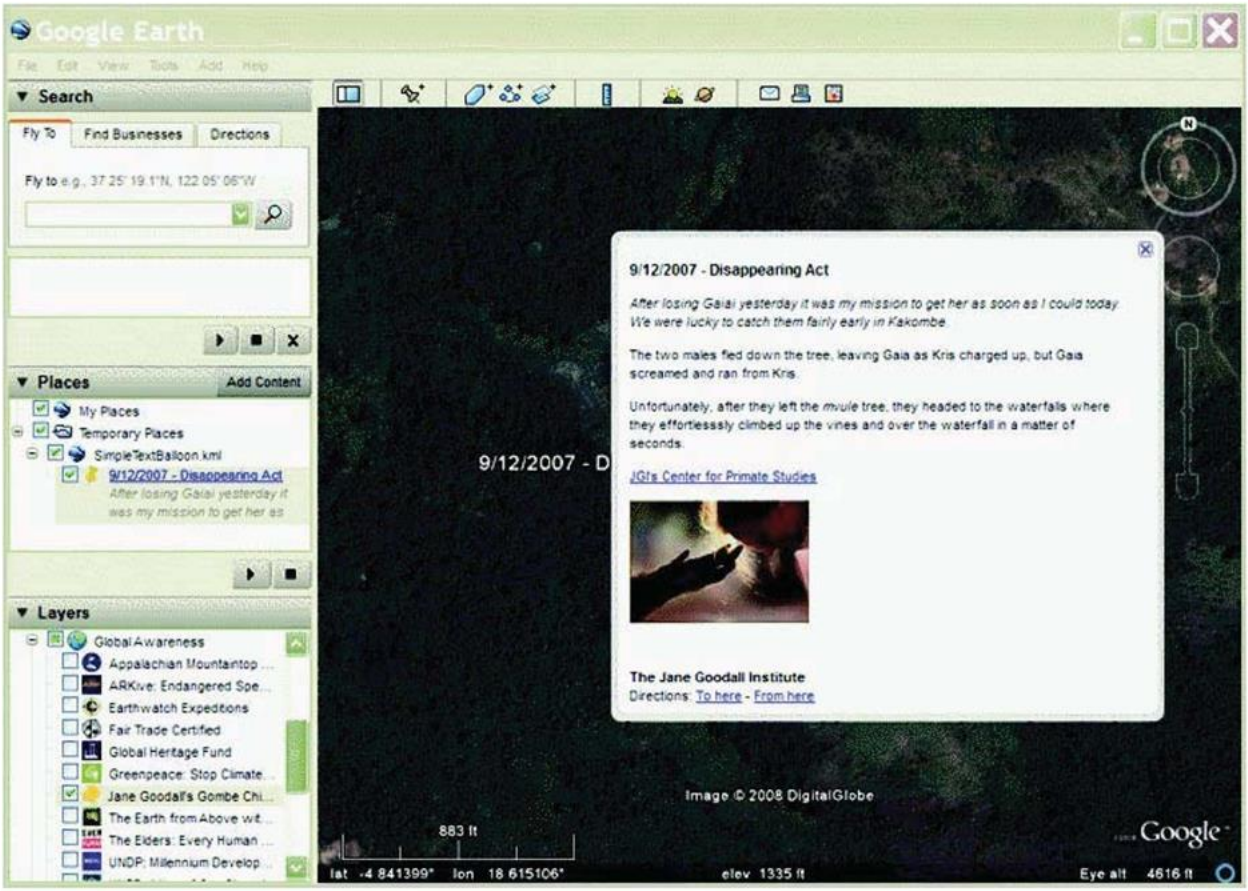


Рис. 2.3. Добавление во всплывающее описание текста, ссылок и изображений (публикуется с разрешения Института Джейн Гудолл; www.janegoodall.org)

Простой шаблон информационного окна

Для добавления текста, изображений и ссылок можете воспользоваться файлом *BalloonTemplate.kml* в качестве шаблона (рис. 2.4). Скопируйте текст из него в какой-нибудь простой редактор, например Блокнот (главное, чтобы он не добавлял информацию о форматировании при сохранении). Теперь подставьте в шаблон свой текст, изображения и ссылки. Сохраните файл с расширением *.kml* (например, *MyFavoritePlacemark.kml*) и откройте его в программе Google Earth или любом другом геобраузере, понимающем формат KML.

Рис. 2.4. Простой шаблон для задания текста и добавления ссылок и изображений

В Google Earth, чтобы перезагрузить файл и увидеть внесенные в него изменения, щелкните правой кнопкой мыши по имени метки в списке и выберите из контекстного меню команду **Восстановить**.

BalloonTemplate.kml

```
<?xml version="1.0" encoding="utf-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
<Document>
  <Placemark>
    <name>Имя метки</name>
    <description>
      <![CDATA[
        Первый абзац.
        <br/>

        Вот так добавляется ссылка:
        <p><a href="http://code.google.com/apis/kml">
          Сюда впишите текст ссылки.</a></p>

        <p>А вот так добавляется изображение:
        <img src="http://code.google.com/images/code_sm.png"></p>

        <p><b>Пример выделения полужирным шрифтом.</b></p>

        <p><em>Пример выделения курсивом.</em></p>
      ]]>
    </description>
  </Placemark>
</Document>
</kml>
```

Изменение цвета фона

Чтобы изменить цвет фона всплывающего окна, воспользуйтесь элементом <BalloonStyle>, который должен быть вложен в элемент <Style>. В примере ниже мы нарисуем окно цвета морской волны:

```
<Style>
  <BalloonStyle>
    <bgColor>fff8fa00</bgColor>
  </BalloonStyle>
</Style>
```

Задание цвета

Значение цвета в KML состоит из двух частей: величины *прозрачности* и собственно *цвета*, который складывается из *синей*, *зеленой* и *красной* составляющих. В текущей версии Google Earth игнорирует прозрачность фона информационного окна, поэтому мы будем говорить только о самих цветах. Информация о задании прозрачности приведена в главе 5.

Значение каждой составляющей цвета может изменяться от 0 до 255, причем 0 означает «отсутствие данной составляющей», а 255 – «полное насыщение».

В KML цвета записываются в *шестнадцатеричном виде* (то есть в системе счисления по основанию 16, где в каждой позиции может стоять цифра от *0* до *9* или одна из букв *a*, *b*, *c*, *d*, *e* и *f*). Любая из четырех компонент цвета (прозрачность и синяя, зеленая и красная составляющие) записывается двумя цифрами в следующем порядке:

aabbggrr,

где

aa – две цифры, определяющие прозрачность (*a* происходит от термина «альфа-канал», который часто применяется в машинной графике для обозначения прозрачности цвета). Значение `00` означает, что цвет полностью прозрачен, а значение `ff` – что цвет абсолютно непрозрачен;

bb – две цифры, определяющие синюю составляющую цвета;

gg – две цифры, определяющие зеленую составляющую цвета;

rr – две цифры, определяющие красную составляющую цвета.

В шестнадцатеричной записи `00` означает, что данной составляющей в цвете нет, а `ff` – что эта составляющая полностью насыщена.

В табл. 2.1 приведены значения основных цветов.

Таблица 2.1. Задание шестнадцатеричных значений цвета

Название цвета	Шестнадцатеричное значение в KML
Red (непрозрачный)	ff0000ff
Green (непрозрачный)	ff00ff00
Blue (непрозрачный)	ffff0000
White (непрозрачный)	ffffffff
Black (непрозрачный)	ff000000

Примечание

В HTML тоже используется представление цвета в виде комбинации красной, зеленой и синей составляющих. Однако в HTML они записываются в порядке *RGB*, тогда как в KML – в порядке *BGR*. Задав поисковой системе запрос «цвета HTML» (HTML colors), вы найдете в Сети таблицы шестнадцатеричных значений цветов в формате HTML. Поменяв порядок составляющих с RGB на BGR, вы сможете использовать их в KML-файлах. Еще одно различие между цветами в HTML и KML заключается в том, что KML не распознает стандартных названий цветов («red», «white», «pink» и т. д.), а HTML допускает их употребление вместо числовых значений.

В табл. 2.2 приведены значения некоторых цветов в KML.

Таблица 2.2. Задание шестнадцатеричных значений цветов

Название цвета	**Шестнадцатеричное значение в KML** ***(все цвета непрозрачные)***
Aqua (цвет морской полны)	ff00ffff
Cornflower Blue (васильковый)	ffed9564
DarkBlue (темно-синий)	ffb00000
DarkGreen (темно-зеленый)	ff2fffad
DarkMagenta (темно-пурпурный)	ff8b008b
DeepPink (темно-розовый)	ff9314ff
FireBrick (кирпичный)	ff2222b2
DarkOrange (темно-оранжевый)	ff008cff
Olive (оливковый)	ff008080
Tan (желто-коричневый)	ffccffff

Например, KML-код для задания желто-коричневого цвета фона выглядит следующим образом:

```
<Style>
  <BalloonStyle>
    <bgColor>ffccffff</bgColor>
  </BalloonStyle>
</Style>
```

Цветоподборщик в Google Earth

В программе Google Earth имеется цветоподборщик, позволяющий выбирать цвет линии, а также контура и заливки многоугольника. Хотя интерфейс не позволяет напрямую задать цвет фона информационного окна, тем не менее вы можете воспользоваться цветоподборщиком для нахождения нужного цвета, как описано ниже. Затем найденное значение следует подставить в элемент <bgColor>.

1. Создайте в программе Google Earth многоугольник. Щелкните по нему правой кнопкой мыши и выберите из контекстного меню пункт **Свойства**, затем перейдите на вкладку **Стиль, цвет** и щелкните по квадратику **Цвет**. Появится окно цветоподборщика.
2. Выберите нужный цвет. Предположим, что вы выбрали светло-желтый цвет: Red:255, Green:255 и Blue:127.
3. Применив технику копирования и вставки, скопируйте окрашенный многоугольник в текстовый редактор, например Блокнот.
4. В разметке вы увидите шестнадцатеричное значение выбранного цвета. В нашем случае KML-код будет выглядеть так:

```
<PolyStyle>
   <color>ff7fffff</color>
</Style>
```

Чтобы у информационного окна оказался фон такого цвета, укажите его код в элементе <bgColor>:

```
<Style>
  <BalloonStyle>
    <bgColor>ff7fffff</bgColor>
  </BalloonStyle>
</Style>
```

Внутренние стили

В главе 4 элемент <Style> будет рассмотрен подробно и, в частности, будет показано, как определить *разделяемый стиль*. А пока просто включите элемент <Style> внутрь элемента <Placemark>, как показано в файле *SimpleTextBalloonWithStyle.kml*. В этом же примере продемонстрировано применение комментариев в KML, они размещаются между тегами <!— и —>. Комментарий не отображается в геобраузере.

SimpleTextBalloonWithStyle.kml

```
<?xml version="1.0" encoding="utf-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Document>
  <Placemark>
    <name>9/12/2007 - Феномен исчезновения</name>
    <description>
      <![CDATA[
        <em>Потеряв вчера Гайю, я сегодня должна была как можно скорее найти ее.
        К счастью, мы довольно быстро натолкнулись на них в Какомбе.</em>
        <p>
        <b>Институт Джейн Гудолл</b>
      ]]>
    </description>
    <Style>
      <BalloonStyle>
        <bgColor>ff669999</bgColor>        <!-- default=ffffff ->
      </BalloonStyle>
    </Style>
    <Point>
      <coordinates>18.615106,-4.841399,0</coordinates>
    </Point>
  </Placemark>
  </Document>
</kml>
```

В KML порядок следования потомков внутри родительского элемента строго фиксирован. Так, внутри элемента <Placemark> элемент <Style> должен находиться после элемента <description>.

Изменение цвета текста

Цвет текста в информационном окне изменяется примерно так же, как цвет фона. Он задается с помощью элемента <textColor>, вложенного в <BalloonStyle>. Значение цвета записывается в шестнадцатеричном виде и состоит из тех же четырех компонент: *прозрачность* и *синяя*, *зеленая* и *красная* составляющие. Вот как можно задать темно-синий цвет на желтовато-сером фоне:

```
<Style>
  <BalloonStyle>
    <bgColor>ff669999</bgColor>
    <textColor>ff660000</textColor>
  </BalloonStyle>
</Style>
```

Исключение маршрута

Чтобы включить в информационное окно маршрут в виде ссылок *Сюда* и *Отсюда*, необходимо поместить следующую строку в любом месте внутри элемента <text>, вложенного в элемент <BalloonStyle>:

```
$[geDirections]
```

Если вы хотите исключить маршрут, то добавьте элемент <text> с *компонентами* для элементов <name> и <description>. Компоненты легко отыскать в файле, поскольку они начинаются со знака $, за которым следует имя элемента, заключенное в квадратные скобки:

```
<Style>
  <BalloonStyle>
    <text>
      <![CDATA[
        $[name]
        <p>
        $[description]
      ]]>
    </text>
  </BalloonStyle>
</Style>
```

Вместо компонента будут подставлены фактические значения, определенные в элементе <Placemark>. Таким образом, в данном примере `$[name]` заменяется строкой «9/12/2007 – Феномен исчезновения», а `$[description]` – текстом, начинающимся со слов «Потеряв вчера Гайю...».

Добавление нестандартного значка

Для добавления нестандартного значка воспользуйтесь элементом <IconStyle>, в котором задается изображение значка. Можете указать один из значков Google Earth, размещенных в веб, или сослаться на свою собственную картинку. Если вы выбрали свое изображение, то не забудьте включить содержащий его файл в KMZ-архив или выложить на веб-сервер, чтобы другие пользователи могли загрузить значок вместе с KML-файлом.

Совет

Чтобы указать стандартный значок Google Earth, откройте диалоговое окно **Редактирование метки** и щелкните по значку желтой канцелярской кнопки. Откроется окно с набором стандартных значков. При щелчке по любому значку его изображение появляется в верхней части окна. URL этого изображения можно подставить в качестве значения элемента <href>, вложенного в элемент <IconStyle>.

Ниже приведен пример использования элемента <IconStyle> для добавления значка бинокля, присутствующего в блоге Института Джейн Гудолл. Файл с изображением находится в одном каталоге с KML-файлом, поэтому задавать полный путь необязательно.

```
<IconStyle>
  <Icon>
    <href>binoculars1.png</href>
```

```
  </Icon>
</IconStyle>
```

Если же файл с изображением хранится на удаленном сервере, то в элементе <href> следует задать его полный URL:

```
<IconStyle>
  <Icon>
    <href>http://www.kanegoodall.oeg/news/gombe-blog/
      assets/binoculars1.png
    </href>
  </Icon>
</IconStyle>
```

Развернутый шаблон всплывающего окна

Шаблон *AdvancedTemplate.kml* вы можете использовать для добавления текста, изображений, ссылок, цветов и значков в информационное окно метки. Наберите приведенный ниже текст в редакторе и подставьте в шаблон собственные цвета, ссылки и текст. Сохраните файл с расширением *.kml*, например *MyFavoritePlacemark.kml*, и откройте его в программе Google Earth или другом геобраузере, понимающем формат KML.

В Google Earth, чтобы перезагрузить файл и увидеть внесенные в него изменения, щелкните правой кнопкой мыши по имени метки в списке и выберите из контекстного меню команду **Восстановить**.

AdvancedTemplate.kml

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
<Document>
  <Placemark>
    <name>Имя метки</name>
    <description>
      <![CDATA[
        Первый абзац.
        </br>

        Вот как добавляется ссылка:
        <a href="http://code.google.com/apis/kml">
          Сюда впишите текст ссылки.</a>

        <p>
        А так вставляется изображение:
          <img src="http://code.google.com/images/code_sm.png">
        </p>

        <p>
          <b>Пример выделения полужирным шрифтом.</b>
        </p>

        <p>
          <em>Пример выделения курсивом.</em>
        </p>
      ]]>
    </description>
    <Style>
      <IconStyle>
        <Icon>
          <href>binoculars1.png</href>
        </Icon>
```

```
      </IconStyle>
      <BalloonStyle>
        <bgColor>ff669999</bgColor>          <!-- default=ffffffff ->
        <textColor>ff660000</textColor>      <!-- default=ff000000 ->
        <text>                               <!-- так исключается маршрут ->
          <![CDATA[
          $[name]
          <p>
          $[description]
          ]]>
        </text>
      </BalloonStyle>
    </Style>
    <Point>
      <coordinates>...</coordinates>
    </Point>
  </Placemark>
</Document>
</kml>
```

Задание точки обзора

Когда пользователь дважды щелкает по метке, Google Earth автоматически располагает панораму прямо над этой меткой. Расположением начальной точки обзора метки можно управлять с помощью элемента <LookAt> или <Camera>. В этом разделе описывается, как задать для метки элемент <LookAt>.

Создав набор меток, вы можете последовательно перемещаться от одной к другой, как описано в разделе «Облет меток в программе Google Earth» ниже. Дополнительную информацию о камерах см. в главе 5 «Наложения».

Элемент LookAt

Элемент <LookAt> описывает точку обзора, используя в качестве репера какую-нибудь точку на земной поверхности. Ниже показан синтаксис этого элемента:

```
<LookAt id="ID">
   <longitude>0.0</longitude>
   <latitude>0.0</latitude>
   <altitude>0.0</altitude>
   <heading>0.0</heading>
   <tilt>0.0</tilt>
   <range>0.0</range>
   <altitudeMode>clampToGround</altitudeMode>
</LookAt>
```

В этой книге во всех описаниях синтаксиса приведены значения по умолчанию для простых элементов. Более детальную информацию о синтаксисе KML-элементов см. в приложении А. Если элемент может принимать одно из дискретного набора перечисляемых значений, то в приложении приведен их полный список. (Так, потомок <altitudeMode> элемента <LookAt> принимает одно из перечисляемых значений.) В приложении А в комментариях приведена информация о типах значений каждого элемента.

У элемента <LookAt> могут быть следующие потомки:

<longitude>

Долгота точки, на которую мы смотрим из точки обзора. Значением является угловое расстояние в градусах относительно нулевого меридиана. Значения к западу от нулевого меридиана изменяются в диапазоне от –180° до 0, значения к востоку – от 0 до 180°.

<latitude>

Широта точки, на которую мы смотрим из точки обзора. Значением является угловое расстояние в градусах относительно экватора, оно изменяется в диапазоне от –90° до 90° (с севера на юг).

<altitude>

Высота от земной поверхности в метрах. Об интерпретации значения см. описание элемента <altitudeMode>.

<heading>

Направление (север, юг, восток, запад) в градусах. По умолчанию подразумевается 0 (север). Изменяется в диапазоне от 0 до 360°, при этом 90° – это восток, 180° – юг, 270° – запад. Допустимы также отрицательные значения (отсчет против часовой стрелки от 0).

<tilt>

Угол между направлением на точку обзора и нормалью к поверхности Земли (рис. 2.5). Изменяется от 0 до 90°. Значение не может быть отрицательным. Значение 0 означает, что точка обзора расположена прямо над обозреваемой точкой, а значение 90° – что взгляд направлен вдоль линии горизонта.

<range>

Расстояние в метрах от точки, заданной координатами <longitude>, <latitude> и <altitude>, до точки обзора (рис. 2.5).

<altitudeMode>

Определяет способ интерпретации значения в элементе <altitude>. Допустимы следующие значения:

`clampToGround` (по умолчанию)

Означает, что значение, заданное в элементе <altitude>, следует игнорировать и поместить точку обзора на поверхность Земли.

`relativeToGround`

Значение, заданное в элементе <altitude>, измеряется в метрах относительно поверхности Земли.

`absolute`

Значение, заданное в элементе <altitude>, измеряется в метрах над уровнем моря.

На рис. 2.5 показано, как эти параметры соотносятся с положением виртуальной камеры, размещенной в точке обзора.

Совет

Простейший способ создать элемент <LookAt> с нужными значениями – воспользоваться функцией **Снимок метки** в программе Google Earth. Выбрав на трехмер-

ной панораме какую-нибудь деталь, вы можете облететь ее, наклонить или повернуть карту, пока не будете удовлетворены получившимся обзором. Щелкните правой кнопкой мыши по детали и выберите из контекстного меню пункт **Снимок метки**. Воспользуйтесь описанной в главе 1 техникой копирования и вставки, чтобы посмотреть на значения параметров <LookAt> в текстовом редакторе.

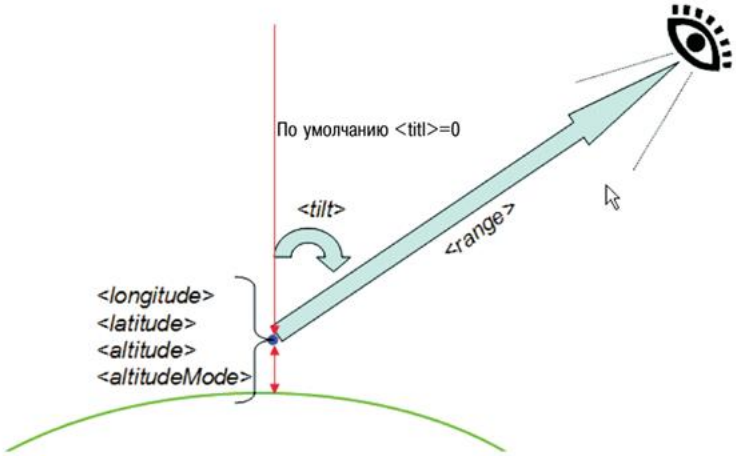


Рис. 2.5. Параметры элемента <LookAt>

Поиск и исправление ошибок

Если вы явно задали точку обзора для метки, то Google Earth показывает карту именно с этой точки, даже если метка, содержащая элемент <LookAt> или <Camera>, при этом не видна. Протестируйте свой KML-код и убедитесь, что указанная точка обзора дает желаемые результаты. Если вы задали элемент <LookAt>, а метка вдруг исчезла с панорамы, то, наверное, следует подправить параметры элемента.

Ниже приведен пример использования элемента <LookAt> для задания точки обзора для метки **Феномен исчезновения**:

```
<LookAt>
  <longitude>18.615083</longitude>
  <latitude>-4.841518</latitude>
  <altitude>0</altitude>
  <range>8969.164821</range>
  <tilt>57.480205</tilt>
  <heading>11.022992</heading>
</LookAt>
```

На рис. 2.6 показано, как эта метка выглядит из точки обзора, заданной таким элементом <LookAt>. Элемент <LookAt> должен находиться внутри элемента <Placemark> сразу после <description>.

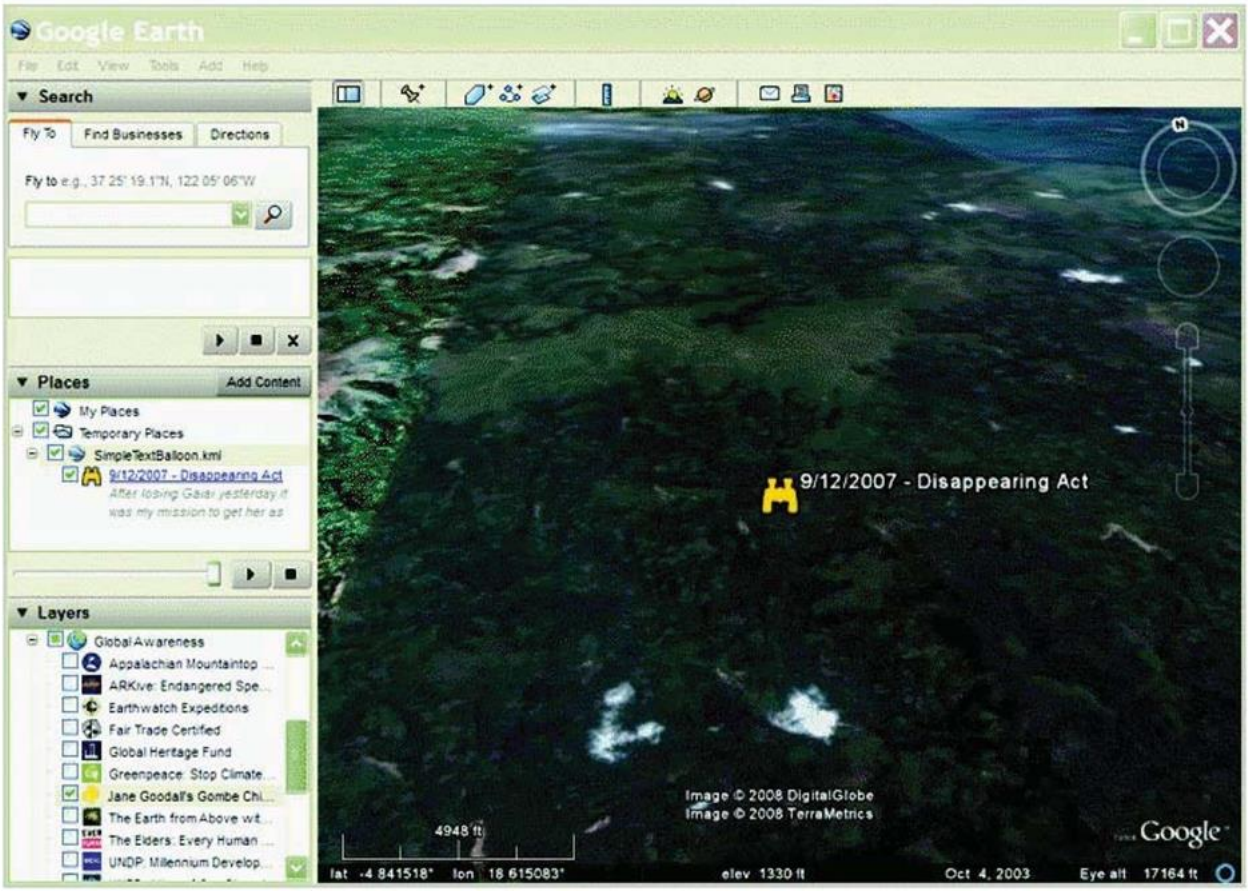


Рис. 2.6. Добавление <LookAt> и нестандартного значка в элемент <Placemark>

Облет меток в программе Google Earth

Для группировки нескольких меток служит элемент <Folder>. Как и <Placemark>, этот элемент может содержать потомков <name> и <description>, которые отображаются в списке **Метки**. Программа Google Earth позволяет перемещаться между метками в одной папке. Для этого следует щелкнуть по папке и нажать кнопку воспроизведения с треугольничком на панели со списком **Метки**. Если для меток были заданы точки обзора, то Google Earth автоматически совершит облет всех точек обзора. В файле *GroupingPlacemarks.kml* приведен пример элемента <Folder>, содержащего четыре метки <Placemark>, для каждой из которых с помощью элемента <LookAt> задана точка обзора.

GroupingPlacemarks.kml

```
<?xml version="1.0" encoding="utf-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Folder>
    <name>Облет меток</name>
    <open>1</open>
    <Placemark id="home">
      <name>Моя первая метка</name>
      <description>Здесь путешествие начинается.</description>
      <LookAt>
```

```
      <longitude>18.595265</longitude>
      <latitude>-4.809265</latitude>
      <altitude>0</altitude>
      <heading>-146.75992</heading>
      <tilt>52.843216</tilt>
      <range>13962.113122</range>
    </LookAt>
    <Style>
      <IconStyle>
        <Icon>
          <href>binoculars1.png</href>
        </Icon>
      </IconStyle>
    </Style>
    <Point>
      <coordinates>18.615106,-4.841399,0</coordinates>
    </Point>
  </Placemark>
  <Placemark id="lunch">
    <name>Моя вторая метка</name>
    <description>
      <![CDATA[
        Это вторая остановка в пути.
        <a href="#journeyEnd;balloonFlyto">
        Щелкните здесь</a> чтобы перейти в конец путешествия.
      ]]>
    </description>
    <LookAt>
      <longitude>18.598893</longitude>
      <latitude>-4.833605</latitude>
      <altitude>0</altitude>
      <heading>-111.390223</heading>
      <tilt>52.843802</tilt>
      <range>11566.909708</range>
    </LookAt>
    <Style>
      <IconStyle>
        <Icon>
          <href>binoculars1.png</href>
        </Icon>
      </IconStyle>
    </Style>
    <Point>
      <coordinates>18.605812,-4.856031,0</coordinates>
    </Point>
  </Placemark>
  <Placemark id="rest">
    <name>Моя третья метка</name>
    <description>
      <![CDATA[
        Едем дальше ... <a href="#lunch;balloonFlyto">
        Щелкните здесь</a>, чтобы вернуться к месту обеда.
      ]]>
    </description>
    <LookAt>
      <longitude>18.579639</longitude>
      <latitude>-4.8470967</latitude>
      <altitude>0</altitude>
      <heading>-79.190397</heading>
      <tilt>52.843246</tilt>
      <range>7431.249174</range>
    </LookAt>
    <Style>
      <IconStyle>
        <Icon>
```

```
            <href>binoculars1.png</href>
          </Icon>
        </IconStyle>
      </Style>
      <Point>
        <coordinates>18.59093,-4.863163,0</coordinates>
      </Point>
    </Placemark>
    <Placemark id="journeyEnd">
      <name>Моя четвертая метка</name>
      <description>Это последняя остановка.</description>
      <LookAt>
        <longitude>18.56632</longitude>
        <latitude>-4.721962</latitude>
        <altitude>0</altitude>
        <heading>-9.00462</heading>
        <tilt>52.843308</tilt>
        <range>18890.906055</range>
      </LookAt>
      <Style>
        <IconStyle>
          <Icon>
            <href>binoculars1.png</href>
          </Icon>
        </IconStyle>
      </Style>
      <Point>
        <coordinates>18.575425,-4.755368,0</coordinates>
      </Point>
    </Placemark>
  </Folder>
</kml>
```

Перелет к новой метке

Обратите внимание на атрибут *id*, присвоенный каждой метке в предыдущем примере. Он призван уникально идентифицировать элемент внутри KML-файла. На элемент, которому присвоен идентификатор, можно ссылаться как внутри файла, так и из других KML-файлов. Чтобы сослаться на элемент в том же файле по его идентификатору, укажите идентификатор с префиксом #, как показано в примере. А чтобы сослаться на элемент в другом файле, задайте полный путь, например:

```
http://myserver.com/TourOfFrenchChateaux.kml#PalaisRoyale
```

Анкеры

Поведение FlyTo, ассоциированное с гиперссылками в Google Earth, можно задать, включив в URL одну из следующих строк:

`;flyto` лететь к метке;

`;balloon` открыть информационное окно метки, но не лететь к ней;

`;balloonFlyto` открыть информационное окно метки и лететь к ней.

В файле *GroupingPlacemarks.kml* спецификация `balloonFlyto` встречается во второй и третьей метках.

Другие потомки элемента Placemark

К числу других часто употребляемых потомков элемента <Placemark> относятся следующие:

- элемент <visibility> определяет, будет ли метка присутствовать на панораме при первом показе файла в Google Earth. Если значение равно 1, то метка рисуется (и соответствующий ей флажок в списке **Метки** будет отмечен). Если значение равно 0, то метка не рисуется (и флажок не отмечен). Чтобы изменить значение на противоположное, пользователь должен щелкнуть по флажку в списке **Метки**;
- элемент <Snippet> содержит краткое описание метки, отображаемое под ее именем в списке **Метки**. Можно задать максимальное число видимых строк описания. Если этот элемент отсутствует, то в списке отображаются первые две строки полного описания из элемента <description>. В отличие от <description>, элемент <Snippet> не поддерживает HTML-разметку. Чтобы воспрепятствовать появлению содержимого элемента <description> в списке, поместите внутрь <Placemark> пустой элемент `<Snippet/>`.

Синтаксис элемента <Placemark>

```
<Placemark id="ID">
  <!-- наследуются от элемента Feature ->
  <name>...</name>
  <visibility>1</visibility>
  <open>0</open>
  <atom:author>...</atom:author>
  <atom:link />
  <address>...</address>
  <xal:AddressDetails>...</xal:AddressDetails>
  <phoneNumber>...</phoneNumber>
  <Snippet maxLines="2">...</Snippet>
  <description>...</description>
  <AbstractView>...</AbstractView>
  <TimePrimitive>...</TimePrimitive>
  <styleUrl>...</styleUrl>
  <StyleSelector>...</StyleSelector>
  <Region>...</Region>
  <ExtendedData>...</ExtendedData>
  <!-- специфичны для элемента Placemark ->
  <Geometry>...</Geometry>
</Placemark>
```

Элементы, производные от Feature

Feature (деталь) – абстрактный элемент, то есть сам по себе он никогда в KML-файле не встречается. *Абстрактные элементы* – это удобный механизм определения элементов, имеющих много общих черт. На рис. 2.7 показано также, что сам элемент Feature наследует другому базовому элементу, Object. Любой элемент, производный от Object, может иметь атрибут *id*. Как мы увидим в следующей главе, идентификаторы полезны для создания ссылок на метки. Они используются также для обновления значений элементов после того, как KML-файл загружен (см. главу 7 «Динамический KML»).

Элемент <Placemark> принадлежит семейству элементов, производных от Feature. В элементе Feature определен набор потомков, наследуемых всеми производными от него элементами. На рис. 2.7 показаны элементы, производные от Feature.

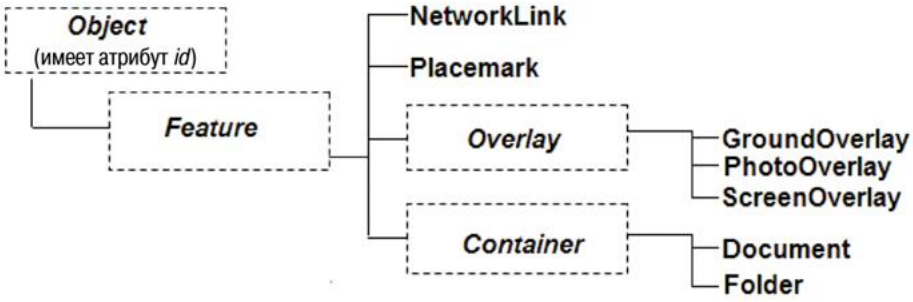


Рис. 2.7. Элемент Feature и производные от него

У каждого из этих элементов есть ряд общих потомков, а также некоторые дополнительные потомки, определяющие его уникальное поведение. Взгляните еще раз на синтаксис элемента <Placemark>, обращая внимание на то, какие потомки унаследованы от Feature.

Поведение элементов-потомков зависит от контекста, в котором они встречаются. Например, элемент <open> в каждом из элементов <Document>, <Folder> и <NetworkLink> определяет, открыт ли его родитель при первом отображении соответствующей детали (1 – открыт, 0 – закрыт). В любом другом элементе, например <Placemark>, потомок <open> просто игнорируется.

Упаковка KML-файлов в KMZ-архив

Многие KML-презентации включают дополнительные изображения, модели и текстуры, находящиеся в отдельных файлах. Например, можно создать нестандартный значок метки, а в информационном окне разместить относящиеся к метке фотографии. Если вы хотите поделиться своим KML-файлом, например отправить его по почте приятелю, то необходимо приложить к основному файлу и все дополнительные.

KMZ-архив представляет собой набор файлов, необходимых для создания одной KML-презентации. В этот архив включаются все локальные файлы, на которые есть ссылки в KML-файле. KMZ-архив – это автономный пакет, который не обязательно размещать на сетевом сервере, его можно отправлять по электронной почте и хранить в виде единого компонента. Помимо всего прочего, данные в KMZ-архиве сжаты, то есть его размер меньше суммарного размера отдельных файлов. Программа Google Earth может читать KML- и KMZ-файлы непосредственно, а также сохранять файлы в виде KMZ-архивов.

По умолчанию при создании KMZ-архива Google Earth присваивает основному KML-файлу имя *doc.kml*. Локальные файлы, на которые он ссылается, упаковываются в архив в формате ZIP. Такие ZIP-архивы можно создать и напрямую

с помощью Windows Explorer или Mac Finder. В общем случае вы просто отмечаете все файлы в папке, содержащей основной KML-файл, а затем выполняете команду типа **WinZip > Add to Zip file**. Формат ZIP поддерживают и многие другие приложения, например WinZip в Windows, Stuffit на платформе Mac и zip в Linux и Macintosh.

Примечание

Из KMZ-архива загружается только первый KML-файл (файл верхнего уровня, по умолчанию называемый *doc.kml*). Он может ссылаться на другие KML-файлы (например, в элементе <NetworkLink>, который описан в главе 6 «Сетевые ссылки»).

Создав ZIP-файл, измените его расширение на *.kmz* (так, файл *myHawaiiTrip.zip* следует переименовать в *myHawaiiTrip.kmz*). KMZ-архив можно загружать в программу Google Earth непосредственно, без предварительной распаковки.

Рекомендуемая структура файла

В этом разделе описывается рекомендуемый подход к организации иерархии файлов, на которые ссылается основной KML-файл. Если размер основного файла превышает 10 Кб, то упакуйте его в архив, даже если никаких ссылок на внешние файлы в нем нет. *Обязательно* создавайте KMZ-архив в случае, когда основной файл ссылается на другие (изображения, модели, текстуры). Ниже приведены общие рекомендации по созданию KMZ-архивов:

1. Создайте папку с осмысленным именем, которая потом будет целиком упакована в архив.
2. Поместите в эту папку основной KML-файл, назвав его *doc.kml*.
3. Создайте подпапку для изображений (если таковые имеются). Придумайте для нее осмысленное имя (например, *images/*).
4. Поместите файлы с изображениями в подпапку *images/*.
5. Создайте подпапки для других типов вспомогательных файлов (например, *models/*, *textures/*, *overlays/*). Можете назвать их по-другому, лишь бы было понятно, что там содержится.
6. В каждую папку поместите соответствующие файлы.
7. Не забывайте, что в файле *doc.kml* ссылки на элементы в подпапках *относительные*. Например, ссылка на JPEG-файл в подпапке *images/* должна быть записана так:

   ```
   <href>images/etnaErupting.jpg</href>
   ```

8. Перед тем как создавать архив, проверьте, что все ссылки работают, для чего загрузите файл *doc.kml* в Google Earth (дважды щелкнув по нему мышью).
9. Если все нормально, упакуйте все содержимое архивной папки.
10. Измените расширение имени файла с *.zip* на *.kmz*.

Наоборот, если вы собираетесь посмотреть, какие файлы находятся в KMZ-архиве, измените расширение с *.kmz* на *.zip* и распакуйте архив. Чтобы загрузить KMZ-архив в программу Google Earth, дважды щелкните по нему мышью (предварительно распаковывать его не нужно).

Совет

Расширение *.kmz* используется в имени самого KMZ-архива. Не употребляйте его в имени папки, находящейся внутри архива.

Примеры

Структура простого KMZ-архива иллюстрируется на примере файла *MtEtna.kmz*. В нем есть основной файл *doc.kml* и дополнительное наложение (*etnaErupting.jpg*), помещенное в папку *images/* (рис. 2.8).

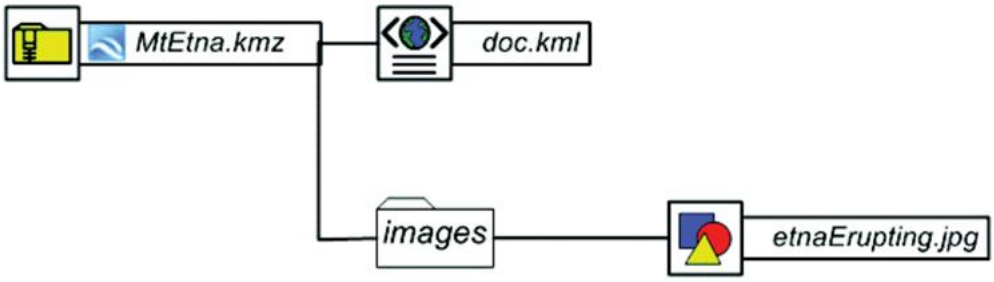


Рис. 2.8. Структура простого KMZ-архива. KML-файл верхнего уровня (корневой) называется doc.kml. Дополнительные файлы помещены в подпапки

На рис. 2.9 представлен более сложный KMZ-архив, следующий тем же рекомендациям по организации большого набора файлов.

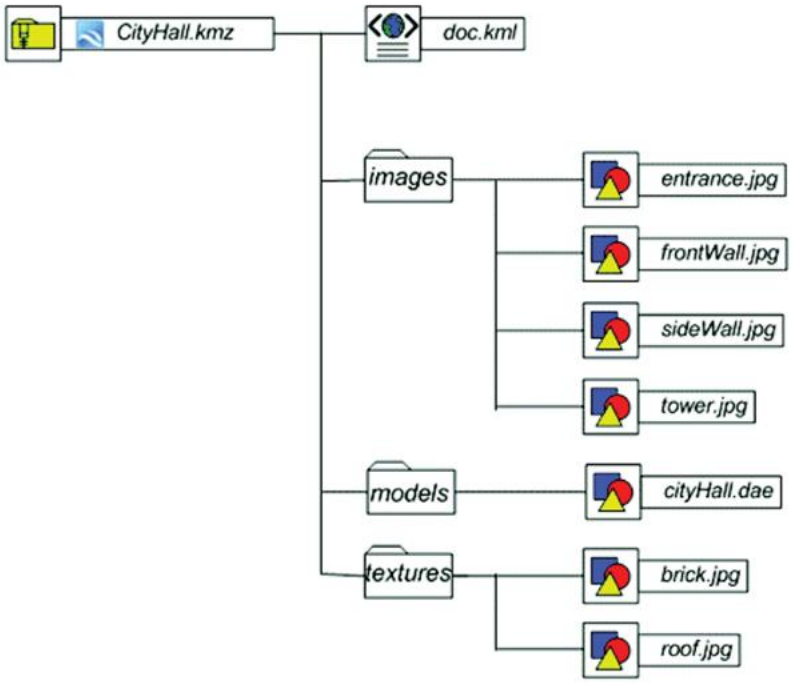


Рис. 2.9. Организация набора файлов в KMZ-архиве. Создайте подпапки с осмысленными именами. Не забудьте, что ссылки на дополнительные файлы из doc.kml должны быть относительными

Совет

Для задания путей к папкам используйте во всех операционных системах (даже в Windows) знак прямой косой черты, например: *images/people/SamSpade.jpg*. (На платформе Windows программа Google Earth самостоятельно преобразует прямую черту в обратную.) Не включайте абсолютные пути к локальным файлам, например C:\. При написании относительных путей можно использовать сокращение, например «..» означает «перейти вверх на один уровень».

Что дальше?

В следующей главе речь пойдет о семействе элементов, производных от абстрактного элемента Geometry. Мы более детально исследуем элемент <coordinates> и узнаем о том, как задавать дополнительные параметры для геометрических форм, в том числе о тесселяции, экструзии и режиме высоты.

Глава 3. Геометрия

Прочитав эту главу, вы сможете:

- ✓ создать путь и задать его толщину и цвет;
- ✓ создать многоугольник и задать его цвет и прозрачность;
- ✓ объяснить, что такое тесселяция и для чего она нужна;
- ✓ добавить в KML-файл простую модель;
- ✓ добавить в KML-файл элементы с информацией об авторе и источнике файла в веб.

Предварительный обзор

Вы уже видели, что элемент <Placemark> может содержать элемент <Point>, определяющий географическое местоположение. В этой главе мы узнаем, что метка может содержать и другие элементы из семейства Geometry, а также их комбинации:

- ❑ <Point>;
- ❑ <LineString>;
- ❑ <LinearRing>;
- ❑ <Polygon>;
- ❑ <MultiGeometry>;
- ❑ <Model>.

Элемент Geometry определяет фундаментальные свойства формы объекта, ассоциированного с меткой. В случае элементов <Point>, <LineString>, <LinearRing> и <Polygon> форма определяется *координатами*. Форма элемента <Model> определяется набором координат, хранящемся во внешнем файле. На форму объекта влияют и другие элементы:

- ❑ <altitudeMode> – определяет способ интерпретации высоты;
- ❑ <extrude> – задает, надо ли продлевать форму до земной поверхности;
- ❑ <tessellate> – определяет, надо ли разбивать длинные отрезки на меньшие части, чтобы они повторяли кривизну Земли.

Для формы можно определить также *стили*: *линейные* – для форм, составленных из линий, и *полигональные* – для многоугольников (заполненных форм с тремя или более сторонами). Все эти вопросы будут рассмотрены в данной главе более детально.

Познакомимся с координатами поближе

Элемент <Point> содержит одного потомка – <coordinates>, который определяет долготу, широту и высоту точки на Земле. Вот как выглядят координаты точки, расположенной на высоте 300 метров над мостом Золотые Ворота:

```
<Point>
  <coordinates>-122.478,37.8107,300</coordinates>
</Point>
```

Три значения, указанные в элементе <coordinates>, разделяются запятыми (без пробелов). Иногда их называют *кортежем*. На рис. 3.1 представлены широта и долгота в программе Google Earth.

Примечание

В программе Google Earth долгота задается первой, а широта – вслед за ней. Во многих других географических системах первой указывается широта.

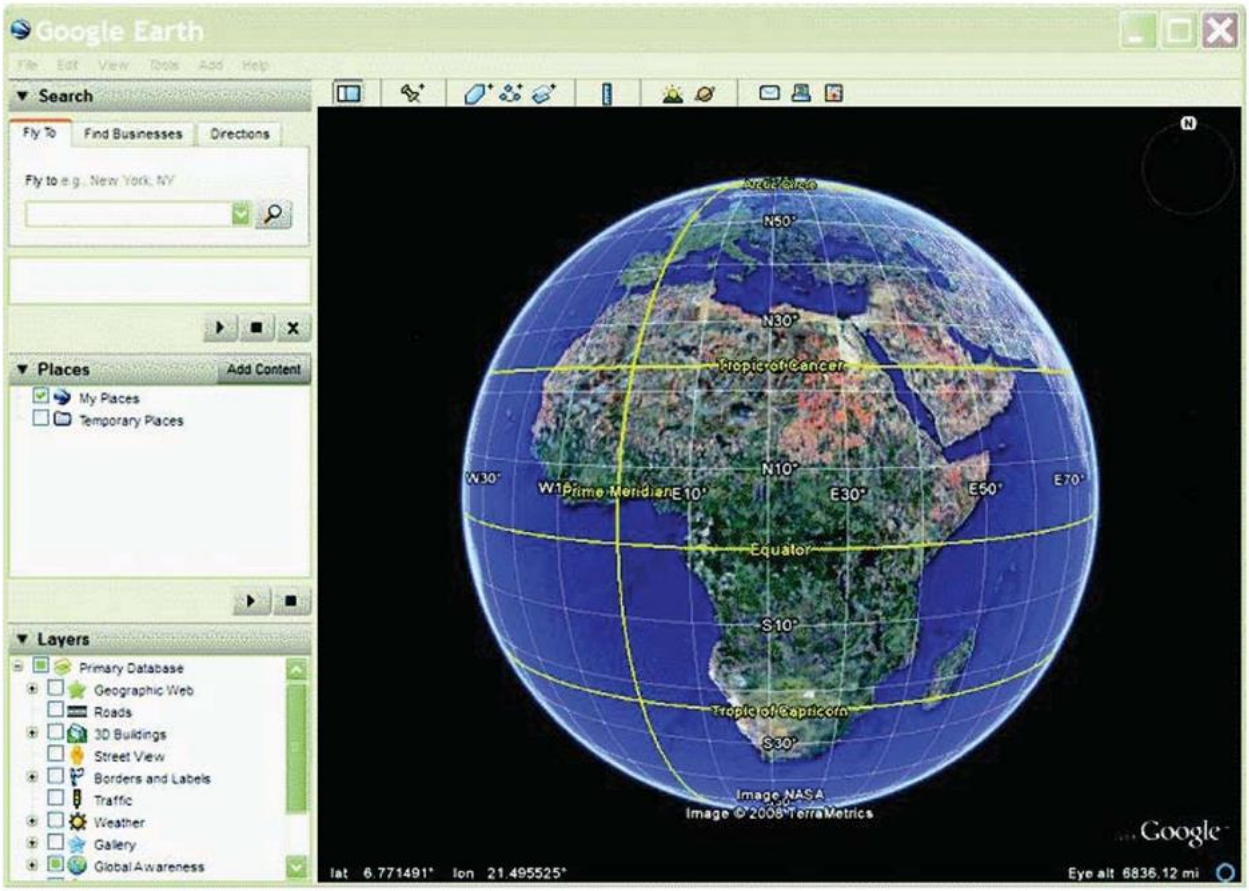


Рис. 3.1. Долгота и широта. Долгота отсчитывается от нулевого меридиана, проходящего через Гринвичскую обсерваторию в Англии. Широта отсчитывается от экватора

Долгота

Линии долгот тянутся с севера на юг, причем линия нулевой долготы совпадает с произвольно выбранным нулевым меридианом, который проходит через городок Гринвич в Англии. Долгота изменяется от 0 до 180 градусов к востоку от нулевого меридиана и от 0 до –180 градусов – к западу от него.

Широта

Линии широт тянутся с востока на запад, причем линия нулевой широты совпадает с экватором. Широта изменяется от 0 до 90 градусов в направлении от экватора к Северному полюсу и от 0 до –90 градусов – в направлении к Южному полюсу. В KML значения долготы и широты задаются в виде десятичной дроби.

Высота и режим высоты

Необязательное значение *высоты* определяет положение точки над уровнем моря (в метрах). Высоту можно задавать в разных элементах (<coordinates>, <Camera>, <LookAt>, геометрических элементах <Point>, <LineString>, <LinearRing>, <Polygon>, <MultiGeometry> и <Model>, элементах <GroundOverlay>, <PhotoOverlay> и <Region>). Значение высоты интерпретируется с учетом действующего режима высоты <altitudeMode>. По умолчанию <altitudeMode> принимает значение `clampToGround`, при котором задание высоты игнорируется. Если же вы хотите, чтобы геобраузер принимал высоту во внимание, то задайте одно из следующих значений:

- `relativeToGround` – высота отсчитывается от истинного уровня земной поверхности в данной точке;
- `absolute` – высота отсчитывается от уровня моря вне зависимости от рельефа местности в данной точке.

Рассмотрим, например, точку, для которой <altitude> равно 160 метров, расположенную в местности, возвышающейся на 40 метров над уровнем моря. Если <altitudeMode> принимает значение `absolute`, то высота считается равной 160 метров (то есть совпадает с заданным значением). Эта точка находится на высоте 120 метров над земной поверхностью. Но если режим <altitudeMode> равен `relativeToGround`, то высота отсчитывается от уровня поверхности, то есть в данном случае составляет 40 + 160 = 200 метров. Иными словами, точка находится на высоте 160 метров над поверхностью Земли, как показано на рис. 3.2.

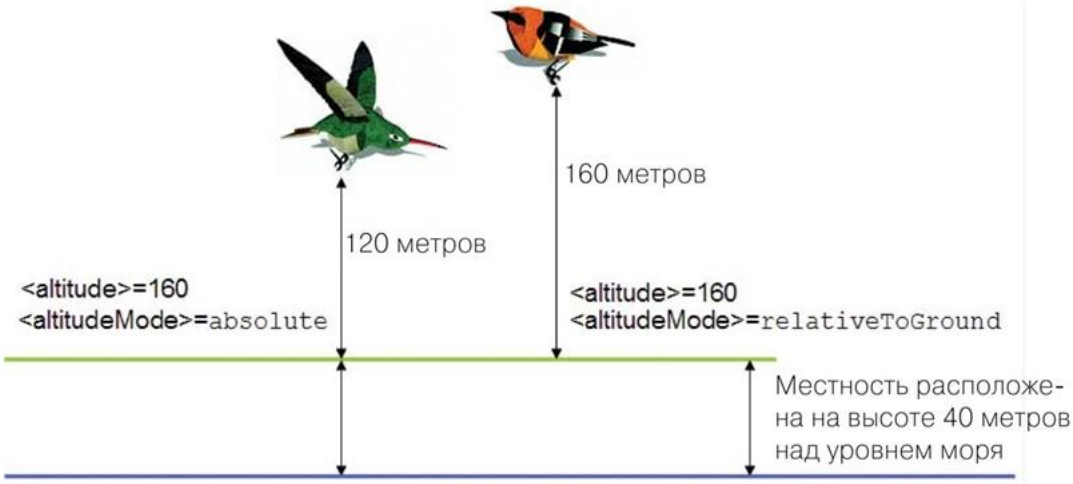


Рис. 3.2. На интерпретацию высоты оказывает влияние значение элемента <altitudeMode> (модель создана Генри Уилтоном, компания Google 3D WareHouse)

Ломаные

Элемент <LineString> представляет собой последовательность из двух или более координат. Точка, определяемая каждым набором координат, соединяется со следующей точкой отрезком прямой. Такие последовательности отрезков обычно называют *путями*[1]. На рис. 3.3 представлена карта тектонических плит в Калифорнии, созданная Геологической службой США. На ней для обозначения границ между плитами применены элементы <LineString> и <LineStyle>.

В файле *SimpleLine.kml* приведен KML-код для создания некоторых отрезков, образующих границу плиты Хуан де Фука, на которой расположено побережье в штатах Орегон и Вашингтон.

SimpleLineString.kml

```
<?xml version="1.0" encoding="utf-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Document>
    <name>Тектонические плиты</name>
    <Placemark>
      <name>Плита Хуан де Фука</name>
      <LineString>
        <coordinates>
          -130.597293,50.678292,0
          -129.733457,50.190606,0
          -130.509877,49.387208,0
          -128.801553,48.669761,0
          -129.156745,47.858658,0
          -128.717835,47.739997,0
        </coordinates>
      </LineString>
    </Placemark>
  </Document>
</kml>
```

[1] Применяется также термин «контур», но «путь» выбран исходя из терминологии, принятой в русифицированной версии программы Google Earth – *Прим. перев.*

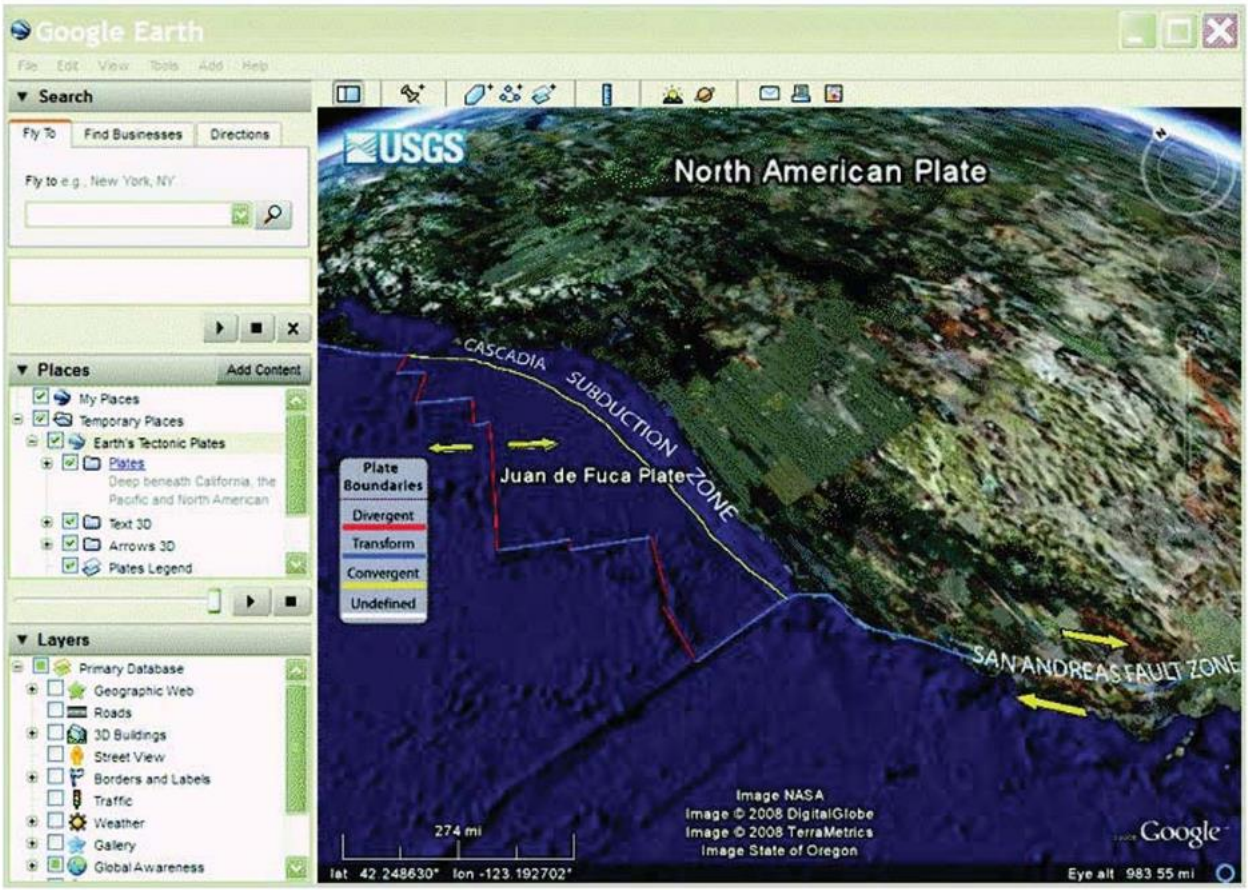


Рис. 3.3. Карта тектонических плит в Калифорнии, созданная Геологической службой США. Границы между плитами обозначены отрезками разных цветов. Легенда и логотип представляют собой накладываемые изображения (см. главу 5) (http://earthquake.usgs.gov/regional/nca/virtualtour/kml/Earths_Tectonic_Plates.kmz)

Тесселяция ломаной

Без тесселяции ломаная порождается путем рисования отрезков прямых между соседними точками. Если точки расположены сравнительно близко друг к другу, то такие отрезки обычно хорошо видны с большинства углов. Но если расстояние между точками велико, то соединяющий их отрезок уже не будет целиком лежать на поверхности Земли, поэтому некоторые его части могут оказаться невидимыми.

Чтобы этого не происходило, задайте значение 1 (то есть «да») в элементе <tessellate>, как показано в файле *TesselatedLineString.kml*.

TesselatedLineString.kml

```
<?xml version="1.0" encoding="utf-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Document>
    <name>Tessellated LineString</name>
    <Placemark>
      <name>Воображаемая плюма</name>
      <LineString>
```

```
        <tessellate>1</tessellate>
        <coordinates>
          -123.597293,50.678292,0
          -121.7178358,47.739997,0
        </coordinates>
      </LineString>
    </Placemark>
  </Document>
</kml>
```

Теперь замените в элементе <tessellate> 1 на 0 (то есть «нет»), сохраните файл и снова загрузите его в Google Earth. Попробуйте посмотреть на ломаную под разными углами – вы увидите, что средняя часть уходит под землю, если тесселяция не производилась. Чтобы эффект тесселяции проявлялся, элемент <altitudeMode> должен иметь значение `clampToGround`.

Экструзия ломаной

Экструдированный геометрический элемент продлевается до земной поверхности. Если он уже находится на поверхности, то продление невозможно, поэтому в случае, когда элемент <altitudeMode> равен `clampToGround`, эффект экструзии не наблюдается.

Если ломаная экструдирована, то из каждой точки проводится линия, соединяющая ее с центром Земли. На рис. 3.4 показана простая экструдированная ломаная. Ее KML-код находится в файле *ExtrudedLineString.kml*. Отметим, что в каждом координатном кортеже задана высота, а элемент <altitudeMode> равен `relativeToGround`.

ExtrudedLineString.kml

```
<?xml version="1.0" encoding="utf-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Document>
    <name>Extruded LineString</name>
    <Placemark>
      <name>Плума Хуан де Фука</name>
      <LineString>
        <extrude>1</extrude>
        <tessellate>1</tessellate>
        <altitudeMode>relativeToGround</altitudeMode>
        <coordinates>
          -130.597293,50.678292,2800
          -129.733457,50.190606,2800
          -130.509877,49.387208,2800
          -128.801553,48.669761,2800
          -129.156745,47.858658,2800
          -128.717835,47.739997,2800
        </coordinates>
      </LineString>
    </Placemark>
  </Document>
</kml>
```

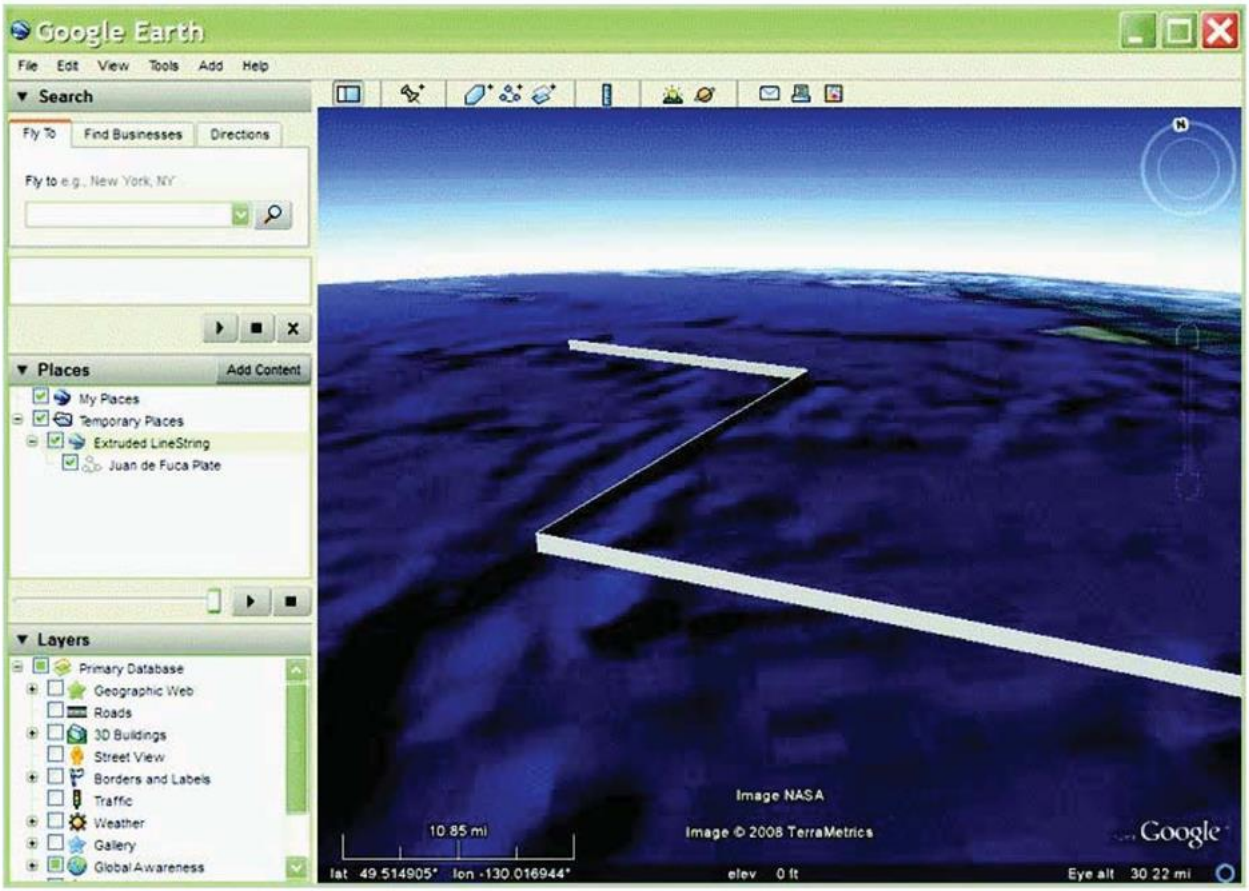


Рис. 3.4. Экструдированная ломаная. Чтобы эффект экструзии был виден, режим <altitudeMode> должен быть равен `relativeToGround` или `absolute`

Экспериментирование с элементом <altitudeMode>

Попробуйте скопировать к себе файл *LineStringWithAltitude.kml* и понаблюдайте, как влияет на отображение значение элемента <altitudeMode>: `clampToGround` (по умолчанию), `absolute` и `relativeToGround`.

LineStringWithAltitude.kml

```
<?xml version="1.0" encoding="utf-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Document>
    <name>Язык KML: глава 3</name>
    <Placemark>
      <name>Розовый путь с высотой</name>
      <Style>
        <LineStyle>
          <color>ffff55ff</color>
          <width>5</width>
```

```
        </LineStyle>
      </Style>
      <LineString>
        <tessellate>1</tessellate>
        <altitudeMode>relativeToGround</altitudeMode>
        <coordinates>
          -65.255616,-22.813902,3000
          .
          .
          .
          -64.6413,-22.712729,3000
        </coordinates>
      </LineString>
    </Placemark>
  </Document>
</kml>
```

Программа Google Maps

Поскольку Google Maps – приложение для показа двумерных карт, то элементы <altitude> и <altitudeMode> в нем игнорируются. Не имеют смысла в двумерном контексте и элементы <tessellate> и <extrude>. На рис. 3.5 показано, как файл *LineStringWithAltitude.kml* выглядит в режиме спутника в программе Google Maps.

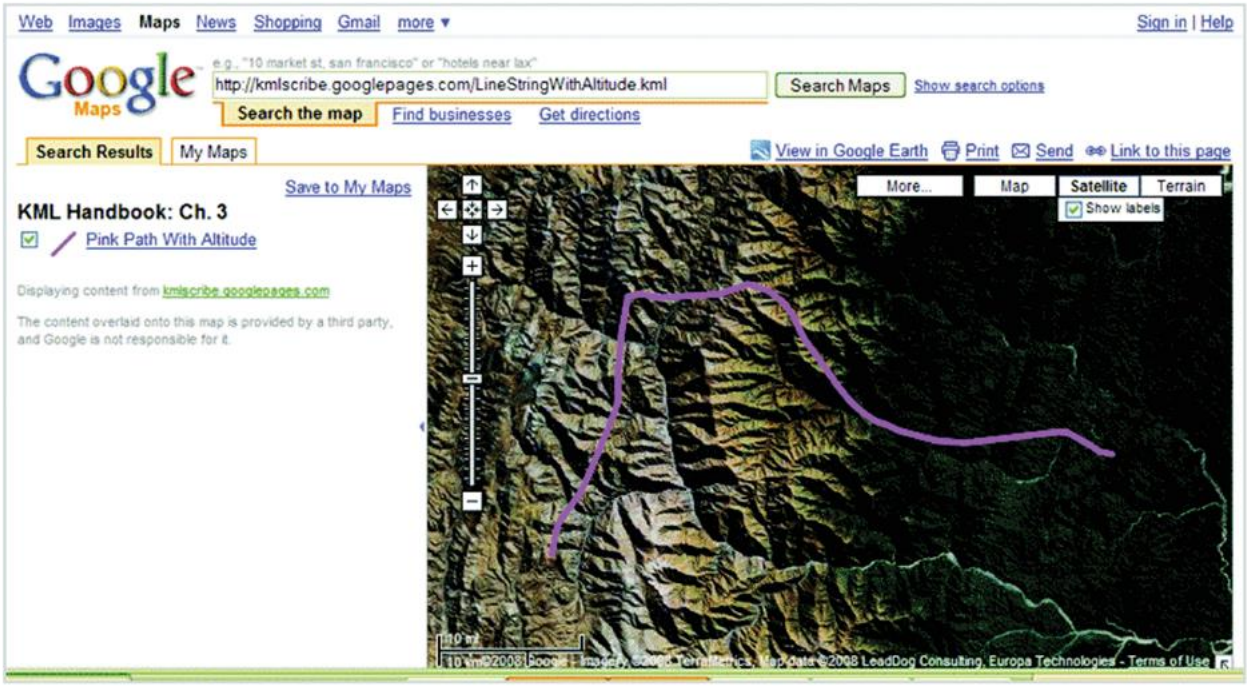


Рис. 3.5. Элемент LineString в программе Google Maps. Некоторые элементы, в частности <altitude> и <altitudeMode>, в двумерном контексте не применяются

Элемент LineStyle

В файле *LineStringWithAltitude.kml* иллюстрируется также применение элемента <LineStyle> для изменения цвета и толщины линии:

```
<Style>
  <LineStyle>
    <color>ffff55ff</color>   <!-- ярко-розовый ->
    <width>5</width>          <!-- толщина 5 пикселей ->
  </LineStyle>
</Style>
```

Цвет линии задается так же, как цвет фона или текста в элементе <BalloonStyle> (см. раздел «Изменение цвета фона» в главе 2). Значения цвета и прозрачности (альфа-канала) записываются в шестнадцатеричном виде и для каждой из четырех компонент изменяются в диапазоне от 00 до ff. Порядок следования – *aabbggrr*, где *aa* – альфа-канал, *bb* – синяя составляющая, *gg* – зеленая составляющая и *rr* – красная составляющая. Подробнее о прозрачности см. главу 5 «Наложения».

Элемент *width* содержит значение с плавающей точкой, определяющее толщину линии в пикселях.

Элемент <LineStyle> является потомком элемента <Style>, который может содержать и другие стилевые элементы, например <BalloonStyle>, описанный в главе 2. В этом примере показано, как сделать <Style> *внутренним* (inline) потомком Feature. В главе 4 рассказано о задании *разделяемых стилей*, на которые можно ссылаться из разных элементов Feature в одном KML-файле.

Многоугольники

Многоугольником называется замкнутая форма, ограниченная тремя или более отрезками прямых. Внутренние части многоугольника можно залить цветом. На рис. 3.6 показана составленная Гавайской вулканической обсерваторией (входит в состав Геологической службы США) карта, на которой с помощью многоугольников обозначены лавоопасные зоны на острове Гавайи.

Простой многоугольник

У простого многоугольника есть одна *внешняя граница*, которая описывается последовательностью координатных кортежей, начинающейся и заканчивающейся в одной и той же точке (то есть образующей замкнутую ломаную). Элемент <coordinates> является потомком элемента <LinearRing>. Ниже представлена базовая структура простого многоугольника, состоящего из одной внешней границы:

```
<Polygon>
  <outerBoundaryIs>
    <LinearRing>
     <coordinates>...</coordinates> <!-- долгота,широта[,высота] ->
    </LinearRing>
  </outerBoundaryIs>
</Polygon>
```

В файле *SimplePolygon.xml* показана структура одного из многоугольников на рис. 3.6. В нее входят следующие важные элементы:

<description>

Включает стандартную HTML-таблицу из четырех строк (описываемых тегами <tr>) и двух колонок (теги <td>).

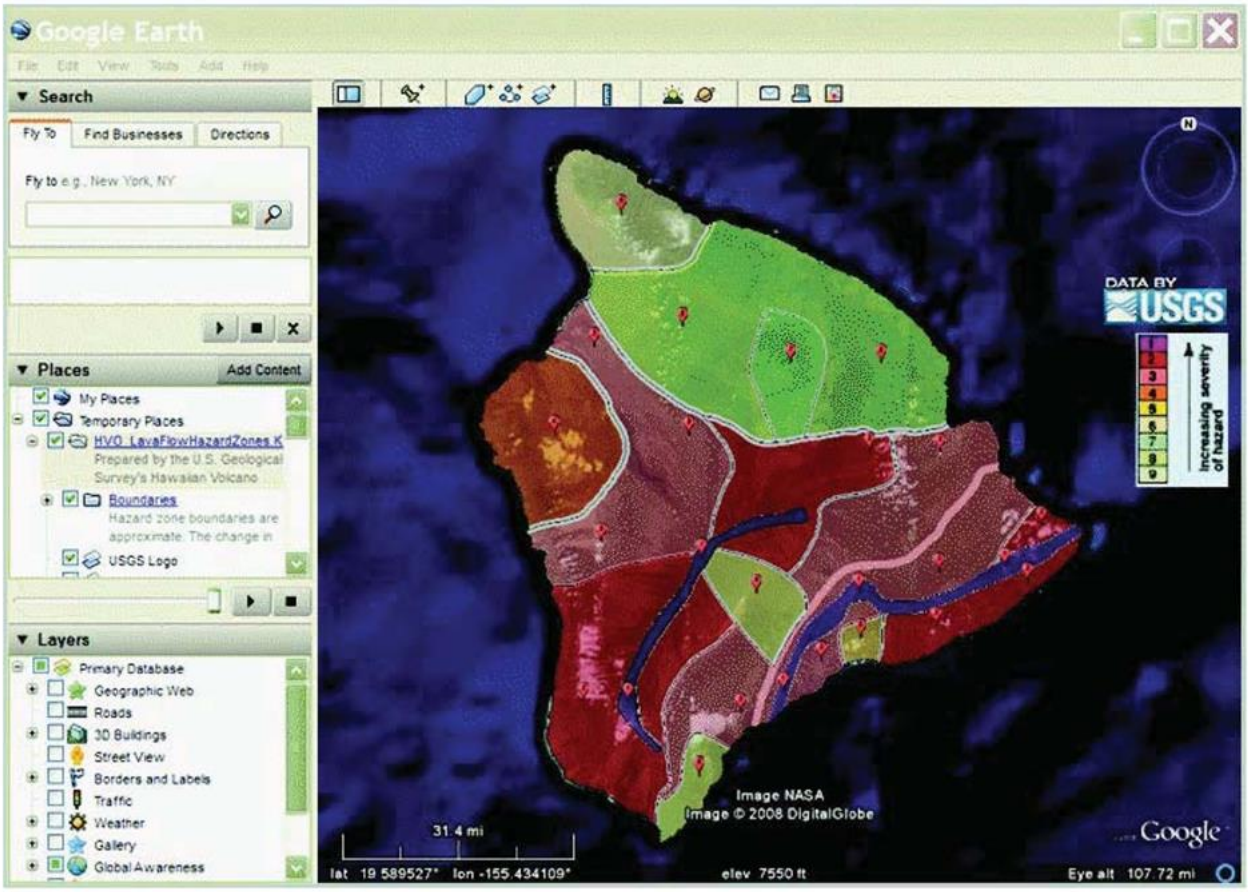


Рис. 3.6. Лавоопасные зоны на острове Гавайи (http://hvo.wr.usgs.gov/hazards/lavazones/HVO_LavaFlowHazrdZones.kmz)

<Style>

Включает элемент <LineStyle>, в котором говорится, что граница многоугольника состоит из белых линий, и элемент <PolyStyle>, в котором задан цвет морской волны для заливки внутренности многоугольника.

<Polygon>

Содержит элементы <outerBoundaryIs>, <LinearRing> и <coordinates>, определяющие геометрию метки.

Отметим также, что многоугольники, как и ломаные, могут иметь в качестве потомков элементы <extrude> и <tessellate>.

SimplePolygon.kml

```
<?xml version="1.0" encoding="utf-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Document>
    <name>Язык KML: глава 3</name>
    <Placemark>
      <name>Простой многоугольник</name>
      <description>
        <![CDATA[
          <br/><br/><br/>
          <table border="1" padding="0">
```

```
          <tr><td>ZONE</td><td>3</td></tr>
          <tr><td>pct_covered_since_1800</td><td>1-5%</td></tr>
          <tr><td>pct_covered_in_750_yrs</td><td>15-75%</td></tr>
          <tr><td>Explanation</td>
              <td>Области, менее опасные, чем Зона 2, в силу большей удаленности от
        кратеров, недавно проявлявших активность, и/или из-за топографических
        особенностей местности, уменьшающих вероятность того, что поток лавы
        сюда доберется.
              </td>
          </tr>
        ]]>
      </description>
      <Style>
        <LineStyle>
          <color>ffffffff</color>
        </LineStyle>
        <PolyStyle>
          <color>f7ebffd5</color>
        </PolyStyle>
      </Style>
      <Polygon>
        <extrude>1</extrude>
        <tessellate>1</tessellate>
        <outerBoundaryIs>
          <LinearRing>
            <coordinates>
              -155.087713,19.728869,0      <!-- первая точка ->
              .
              .
              .
              -155.087713,19.728869,0      <!-- последняя точка совпадает с первой ->
            </coordinates>
          </LinearRing>
        </outerBoundaryIs>
      </Polygon>
    </Placemark>
  </Document>
</kml>
```

Элемент PolyStyle

В примере простого многоугольника показано использование элемента <PolyStyle> для задания цвета внутренности многоугольника. Там же встречается элемент <LineStyle>, описывающий цвет линий, образующих внешнюю границу многоугольника. Ниже приведен синтаксис элемента <PolyStyle>:

```
<PolyStyle>
  <color>ffffffff</color>
  <colorMode>normal</colorMode>
  <fill>1</fill>
  <outline>1</outline>
</PolyStyle>
```

Потомки элемента <PolyStyle> определяются следующим образом:

<color>

Задает цвет многоугольника, полигональных экструзий и экструдированных линий.

<colorMode>

Может принимать значение `normal` или `random`. В последнем случае базовый цвет <color> рандомизируется (применяется равномерное распределение).

Для получения полностью случайных цветов задайте для элемента <color> белый цвет (00ffffff). По умолчанию подразумевается значение `normal`. (Пример использования этого элемента см. в разделе «Более сложный пример» главы 4.)

<fill>

Определяет, нужно ли заливать многоугольник цветом (1 – заливать, 0 – не заливать).

<outline>

Определяет, нужно ли оконтуривать многоугольник (1 – нужно, 0 – не нужно). Поскольку граница многоугольника состоит из *линий*, они рисуются с применением текущего стиля LineStyle.

На рис. 3.7 иллюстрируется случай, когда многоугольник экструдирован и элемент <outline> равен 1. Граница и линии, соединяющие точки с поверхностью, рисуются цветом, заданным в стиле <LineStyle> (в данном случае ff000000 – черным).

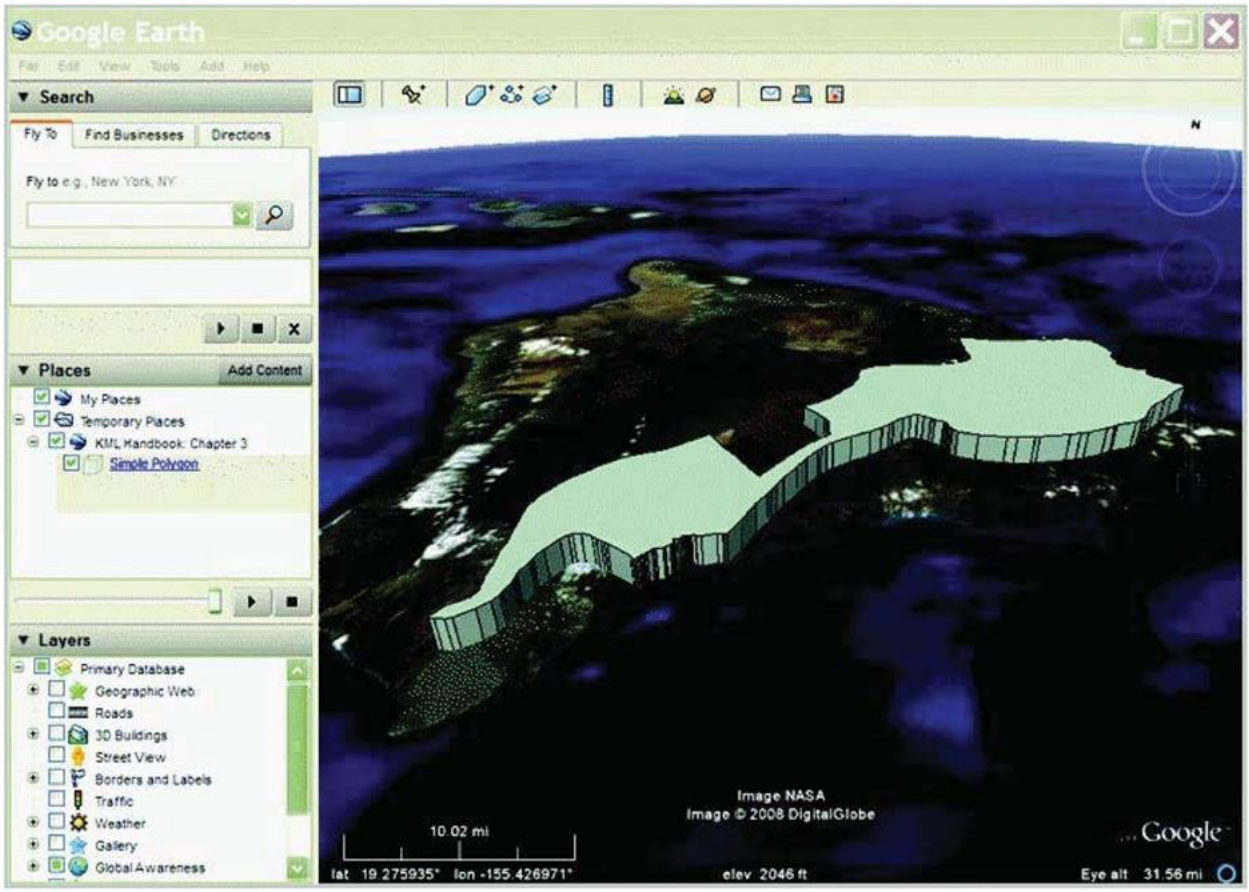


Рис. 3.7. Простой многоугольник с одной внешней границей. Многоугольник *экструдирован*, то есть внешняя граница продлена до земной поверхности

Многоугольники с дырками

Для описания дырки внутри многоугольника следует задать *внутреннюю границу*, то есть воспользоваться элементом <innerBoundary>. Многоугольник может содержать несколько внутренних границ. Базовая структура элемента <Polygon> с дыркой выглядит следующим образом:

```
<Polygon>
  <outerBoundaryIs>
    <LinearRing>
      <coordinates>...</coordinates><!-- долгота,широта[,высота] ->
    </LinearRing>
  </outerBoundaryIs>
  <innerBoundaryIs>
    <LinearRing>
      <coordinates>...</coordinates><!-- долгота,широта[,высота] ->
    </LinearRing>
  </innerBoundaryIs>
</Polygon>
```

Для внешней и внутренней границы действует одно и то же правило: первая и последняя точки должны совпадать. Элементы <Style> метки применяются ко всем элементам многоугольника <Polygon>. Если внутренняя граница лежит *снаружи* внешней, то она отображается в виде отдельного многоугольника. Программа Google Maps трактует все внутренние границы как отдельные многоугольники, но в левой панели показывает их внутри одной и той же метки.

Пример

На рис. 3.8 приведен пример многоугольника с двумя внутренними границами, образующими дырки. Каждая внутренняя граница описывается элементом <LinearRing>, в котором первая и последняя точки совпадают. Этот многоугольник создается файлом *PolyWithInnerAndOuterBoundaries.kml*.

PolyWithInnerAndOuterBoundaries.kml

```
<?xml version="1.0" encoding="utf-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Placemark>
    <name>Многоугольник с двумя дырками</name>
    <Polygon>
      <outerBoundaryIs>
        <LinearRing>
          <coordinates>
            -122.431938278749,37.8019857095478,0
            -122.431873068102,37.8016611830403,0
            -122.431497938539,37.8017138829201,0
            -122.431564485129,37.8020299537248,0
            -122.431938278749,37.8019857095478,0
          </coordinates>
        </LinearRing>
      </outerBoundaryIs>
      <innerBoundaryIs>
        <LinearRing>
          <coordinates>
            -122.431886016065,37.8019318219595,0
            -122.431842117396,37.8017208533219,0
```

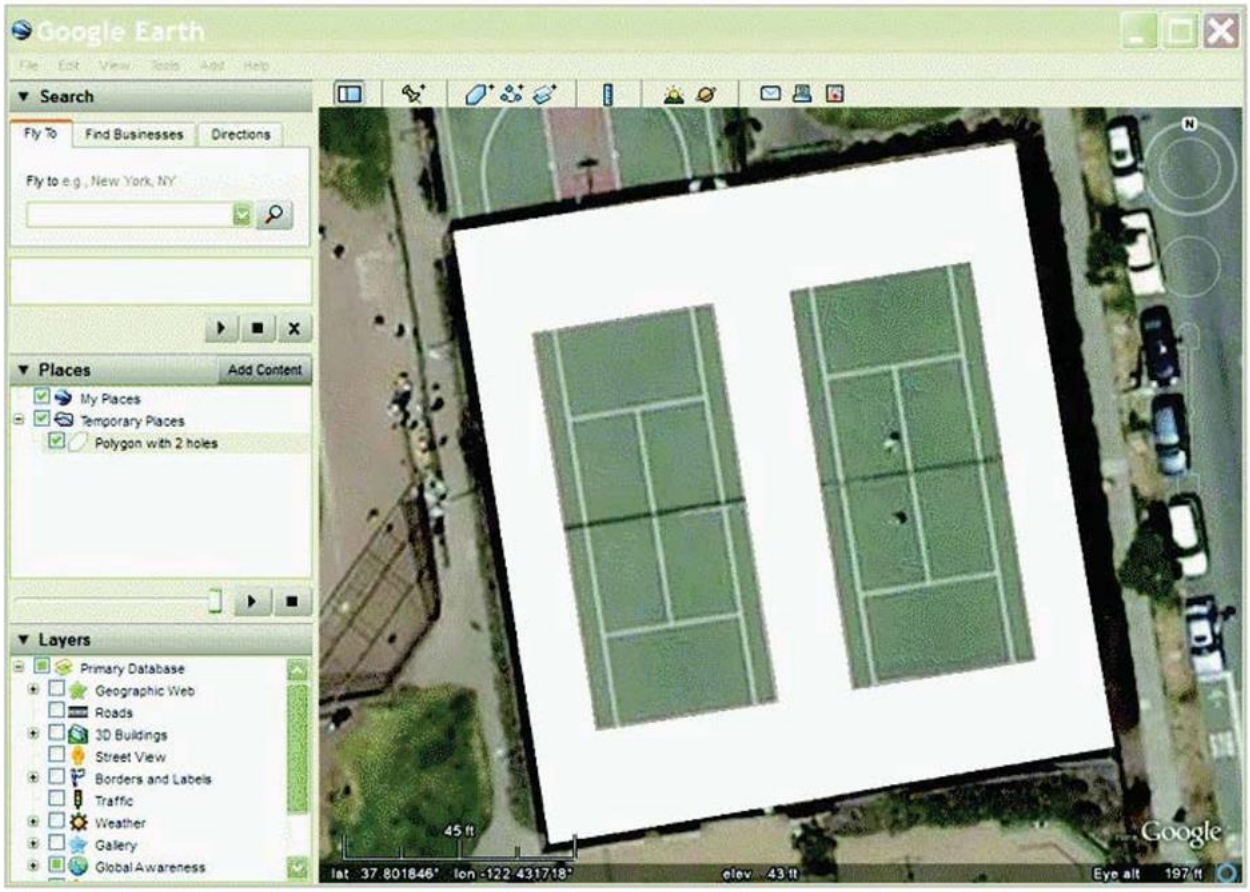


Рис. 3.8. Внутренние границы служат для определения дырок в многоугольнике

```
            -122.431719499379,37.8017366044116,0
            -122.431763224388,37.8019475540673,0
            -122.431886016065,37.8019318219595,0
          </coordinates>
        </LinearRing>
      </innerBoundaryIs>
      <innerBoundaryIs>
        <LinearRing>
          <coordinates>
            -122.431713379578,37.8019536360684,0
            -122.431671885143,37.8017428837353,0
            -122.431549296352,37.8017586463012,0
            -122.431592641627,37.8019684859395,0
            -122.431713379578,37.8019536360684,0
          </coordinates>
        </LinearRing>
      </innerBoundaryIs>
    </Polygon>
  </Placemark>
</kml>
```

Элемент MultiGeometry

Этот элемент представляет собой контейнер для нуля или более элементов Geometry. Чтобы добавить к многоугольнику значок, необходимо воспользоваться элементом <MultiGeometry>. В данном случае <Placemark> должен содержать

<MultiGeometry>, внутри которого находятся <Point> и <Polygon>. Если <MultiGeometry> содержит одновременно <Point> и <Polygon>, то вы получаете все свойства точечной метки, описанные в главе 2. Пользователь может щелкнуть по значку, открыв тем самым информационное окно, ассоциированное с точкой и многоугольником. Это продемонстрировано на примере файла *MultiGeometry.kml*.

MultiGeometry.kml

```
<?xml version="1.0" encoding="utf-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Placemark>
    <name>Аделаида</name>
    <description>Этот многоугольник показывает границы города.</description>
    <Style>
      <LineStyle>
        <color>ff000000</color>
      </LineStyle>
      <PolyStyle>
        <color>c733ffee</color>
      </PolyStyle>
    </Style>
    <MultiGeometry>
      <Point>
        <coordinates>138.6,-34.91,0</coordinates>
      </Point>
      <Polygon>
        <outerBoundaryIs>
          <LinearRing>
            <coordinates>
              138.64,-34.93 138.64,-34.94 138.63,-34.94 138.62,-34.94
              138.62,-34.95 138.62,-34.96 138.63,-34.96 138.62,-34.96
              138.61,-34.97 138.60,-34.97 138.59,-34.97 138.58,-34.97
              138.57,-34.97 138.57,-34.96 138.57,-34.95 138.57,-34.95
              138.57,-34.94 138.57,-34.93 138.57,-34.92 138.57,-34.92
              138.57,-34.91 138.56,-34.91 138.56,-34.90 138.57,-34.90
              138.57,-34.90 138.57,-34.89 138.57,-34.89 138.57,-34.89
              138.56,-34.88 138.56,-34.88 138.56,-34.88 138.56,-34.88
              138.57,-34.88 138.58,-34.87 138.58,-34.87 138.58,-34.86
              138.58,-34.85 138.58,-34.85 138.60,-34.85 138.61,-34.85
              138.63,-34.85 138.63,-34.85 138.64,-34.86 138.64,-34.87
              138.64,-34.87 138.63,-34.87 138.63,-34.88 138.62,-34.88
              138.62,-34.88 138.62,-34.89 138.62,-34.89 138.62,-34.89
              138.63,-34.89 138.63,-34.90 138.64,-34.90 138.64,-34.90
              138.64,-34.91 138.64,-34.91 138.64,-34.92 138.64,-34.92
              138.64,-34.93 138.64,-34.93
            </coordinates>
          </LinearRing>
        </outerBoundaryIs>
      </Polygon>
    </MultiGeometry>
  </Placemark>
</kml>
```

Визуализация этого элемента <MultiGeometry> представлена на рис. 3.9. Значок выглядит и работает так, будто он ассоциирован с многоугольником, хотя на самом деле является частью точечной метки. Чтобы убедиться в этом, закомментируйте элемент <Point>, поместив его в скобки <!— и —>, затем перезагрузите файл в Google Earth и обратите внимание, что значок и надпись исчезли.

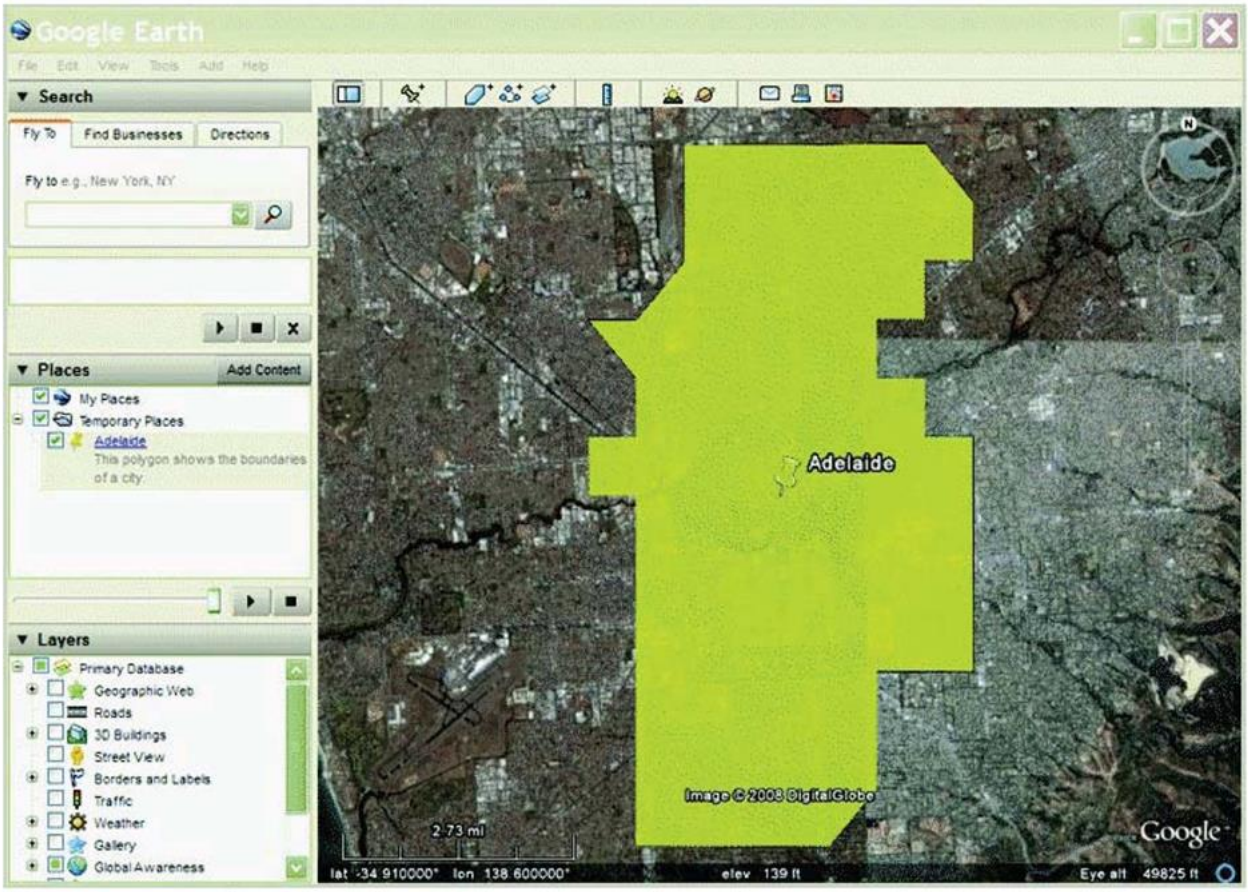


Рис. 3.9. Использование элемента MultiGeometry для добавления значка и надписи к многоугольнику

Модели

Для импорта трехмерной модели в Google Earth воспользуйтесь элементом <Model> и укажите в нем внешний файл, содержащий модель. Элемент <Model> используется точно так же, как любой другой геометрический элемент в KML, то есть должен быть потомком <Placemark> или <MultiGeometry>. В KML можно импортировать трехмерные модели – зданий, мостов, памятников и статуй, – представленные в формате COLLADA. Модели определяются независимо от Google Earth в собственном координатном пространстве. Это можно сделать в таких программах, как Google SketchUp, Autodesk 3D Studio Max, Maya или Softimage XSI. После импорта в Google Earth модель позиционируется, поворачивается и масштабируется для приведения к системе координат Земли. Положение и размеры модели, загруженной в Google Earth, можно изменять с помощью элемента <Update> (см. главу 7 «Динамический KML»).

На рис. 3.10 показана модель гробницы Тутанхамона, в которую включены текстуры камня и настенные изображения павианов по обе стороны погребальной камеры. Координаты вершин многоугольников, составляющих модель, хранятся в файле с расширением *.dae* в формате COLLADA. Эта модель была создана в программе 3D-моделирования Google SketchUp. В COLLADA-файле есть ссылки на

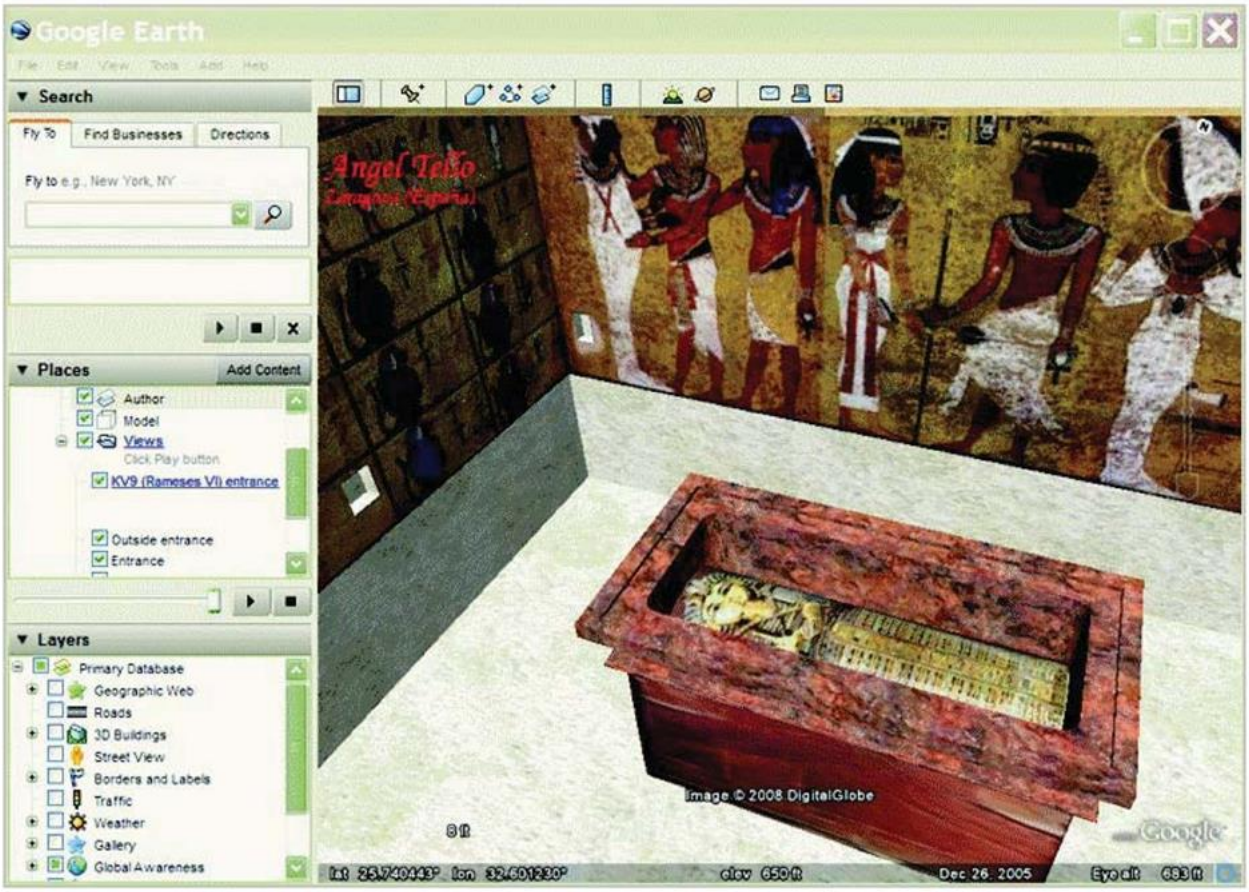


Рис. 3.10. Модель гробницы Тутанхамона с реалистичными текстурами и последовательностью точек обзора, позволяющей пользователю перемещаться из одной подземной камеры в другую (модель создал Энджел Телло, http://bbs.keyhole.com/ubb/download.php?Number=408502. Публикуется с разрешения автора)

внешние файлы текстур, представляющие собой JPEG-изображения, накладываемые на многоугольники, образующие модель. KML-файл, содержащий элемент <Model>, COLLADA-файл и все вспомогательные файлы текстур упакованы в KMZ-архив, который можно отправить по почте или скачать в виде единого пакета (см. главу 2 «Метки и всплывающие окна»).

Файл в формате COLLADA, содержащий геометрию модели, указан в элементе <Link>:

```
<Link>
  <href>KingTutsTomb3D.kmz/files/SUPreview2.dae</href>
</Link>
```

Ниже приведен полный синтаксис элемента <Model>:

```
<Model id="ID">
  <altitudeMode>clampToGround</altitudeMode>
  <Location>
    <longitude>0.0</longitude>
    <latitude>0.0</latitude>
    <altitude>0.0</altitude>
  </Location>
  <Orientation>
```

```
    <heading>0.0</heading>
    <tilt>0.0</tilt>
    <roll>0.0</roll>
  </Orientation>
  <Scale>
    <x>1.0</x>
    <y>1.0</y>
    <z>1.0</z>
  </Scale>
  <Link>...</Link>       <!-- включает href файла модели ->
  <ResourceMap>
    <Alias>
      <targetHref>...</targetHref>
      <sourceHref>...</sourceHref>
    </Alias>
  </ResourceMap>
</Model>
```

В следующих разделах описаны основные элементы и связанные с ними понятия.

<Location>

Задает начальное положение модели на Земле.

<Orientation>

Определяет, как нужно повернуть модель на Земле, чтобы она была правильно ориентирована.

<Scale>

Определяет, как нужно изменить размер модели на Земле.

<ResourceMap>

Задает соответствие между ссылками на текстуры в исходном COLLADA-файле и путями к файлам текстур в KMZ-архиве.

Местоположение

Элемент <Location> содержит уже знакомые вам стандартные спецификации координат в виде элементов <longitude>, <latitude> и <altitude>. Как обычно, интерпретация высоты <altitude> зависит от значения элемента <altitudeMode>, который должен быть первым потомком <Model>. В эту точку помещается начало координат модели (0,0,0).

Ориентация

На рис. 3.11 показана ориентация осей координат типичной модели. В данном случае ось $+x$ направлена вправо, ось $+y$ направлена в сторону наблюдателя и ориентирована на север, а ось $+z$ направлена вверх. (Это обычное в моделировании соглашение, но не требование.)

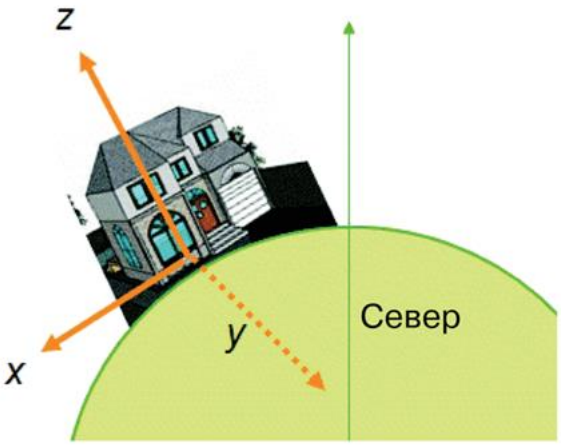


Рис. 3.11. Ориентация типичной модели (модель создана Майклом Массикоте, компания Google 3D Warehouse)

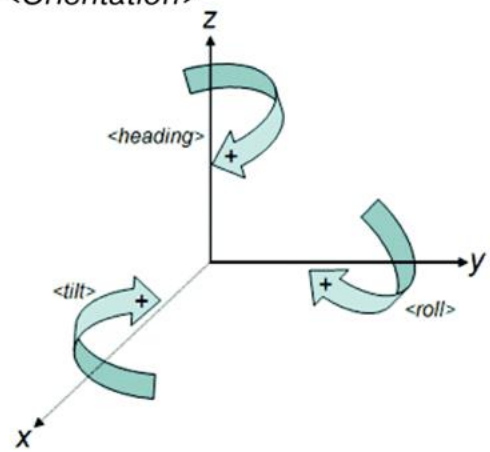
Вращение в трехмерном пространстве

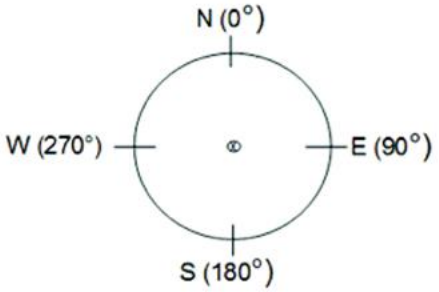
После того как модель привязана к нужной точке, она поворачивается. Порядок выполнения поворотов определяется следующими элементами:

1) <roll>;
2) <tilt>;
3) <heading>.

На рис. 3.12 показана интерпретация этих элементов.

Задание параметров элемента <Orientation>





Элементы *<heading>*, *<tilt>* и *<roll>* задаются в направлении по часовой стрелке (когда наблюдатель смотрит вдоль оси на начало координат)

Рис. 3.12. Повороты относительно осей *x*, *y* и *z*

<heading>

Задает угол поворота относительно оси *z*, представляющей собой эквивалент азимута (направления стрелки компаса). Поворот на 0 градусов означает север, на 90 градусов – восток, на 180 градусов – юг, на 270 градусов – запад.

<tilt>

Задает угол поворота относительно оси *x*. Положительным считается направление по часовой стрелке, когда вы смотрите вдоль оси *x* на начало координат. Значения измеряются в градусах и изменяются в диапазоне от 0 до 180.

<roll>

Задает угол поворота относительно оси *y*. Положительным считается направление по часовой стрелке, когда вы смотрите вдоль оси *y* на начало координат. Значения измеряются в градусах и изменяются в диапазоне от 0 до ±180.

Масштаб

Значения элементов <x>, <y> и <z>, являющихся потомками <Scale>, определяют коэффициент растяжения или сжатия модели в соответствующем направлении. Масштабирование с коэффициентом 1 оставляет модель неизменной в данном направлении. Масштабирование с коэффициентом 0.5 уменьшает модель в два раза. Масштабирование с коэффициентом 2 увеличивает модель в два раза.

Коэффициент масштабирования применяется в пространстве координат модели. Если модель создана во внешнем приложении, ее, возможно, придется уменьшить или увеличить, чтобы она «естественно» вписалась в систему координат геобраузера. Коэффициент масштабирования зависит от конкретной ситуации, поэтому для достижения нужного эффекта предстоит немного поэкспериментировать.

Карта ресурсов

Исходный COLLADA-файл модели содержит ссылки на использованные текстуры. Элемент <ResourceMap> (карта ресурсов) определяет соответствие между исходными местоположениями файлов и их новыми местоположениями в KML или KMZ-файле, содержащем модель и ее текстуры. Это позволяет перемещать и переименовывать файлы, не модифицируя исходный COLLADA-файл, ссылающийся на текстуры.

Карта ресурсов содержит один или несколько элементов <Alias>, отображающих путь к файлу текстуры, указанный в COLLADA-файле (*исходный* путь), на новый путь, указанный в KML/KMZ-файле (*конечный* путь). В элементе <Alias> эти пути представлены соответственно элементами <sourceHref> и <targetHref>.

<targetHref>

Задает файл текстуры, который должна прочитать программа Google Earth. Ссылка может быть сформирована относительно файла изображения в KMZ-архиве или являться абсолютным путем к файлу (например, в виде URL).

<sourceHref>

Путь к файлу текстуры, заданный в COLLADA-файле с расширением *.dae*.

Нередко в COLLADA-файлах используются относительные ссылки, тогда <targetHref> просто является копией <sourceHref>.

В следующем примере карты ресурсов показано, как Google Earth автоматически помещает в элемент <ResourceMap> правильные значения, когда вы сохраняете KMZ-архив на своем компьютере. Этот фрагмент взят из модели гробницы Тутанхамона (рис. 3.10), в которой для достижения реалистичного эффекта используются 16 текстур и изображений (см. файл KingTut.kmz).

KingTut.kmz

```
.
.
.
<ResourceMap>
  <Alias>
    <targetHref>
      C:/Documents And Settings/username/Local Settings/Temp/
      408502-Tutankhamon&apos;stomb3D.kmz/files/Material1.jpg
    </targetHref>
    <sourceHref>
      ../images/Material1.jpg
    </sourceHref>
  </Alias>
  <Alias>
    <targetHref>
      C:/Documents And Settings/username/Local Settings/Temp/
```

```
      408502-Tutankhamon&apos;stomb3D.kmz/files/
      Material1Stone_Brushed_Khaki_big.jpg
    </targetHref>
    <sourceHref>
      ../images/Material1Stone_Brushed_Khaki_big.jpg
    </sourceHref>
  </Alias>
  <Alias>
    <targetHref>
      C:/Documents And Settings/username/Local Settings/Temp/
      408502-Tutankhamon&apos;stomb3D.kmz/files/
      Material1Stone_Brushed_Khaki_big1noCulling.jpg
    </targetHref>
    <sourceHref>
      ../images/Material1Stone_Brushed_Khaki_big1noCulling.jpg
    </sourceHref>
  </Alias>
.
.
.
</ResourceMap>
.
.
.
```

Включение информации об авторе и источнике

Считается хорошим тоном включать информацию об авторе и адрес веб-страницы, на которой пользователи могут найти ваши KML-файлы. Наличие такой информации в самом KML-файле гарантирует надлежащее установление авторства даже в том случае, когда этот файл или содержащий его архив распространяются независимо. Эти сведения отображаются в результатах поиска географической информации как в геобраузере Google Earth, так и в других приложениях, например Google Maps.

Google Earth использует элементы *author*, *name* и *link* точно так же, как они определены в формате агрегирования Atom. Полная спецификация этого формата находится по адресу http://atompub.org. Вот пример использования элементов атрибуции в KML-файле:

```
<kml xmlns="http://www.opengis.net/kml/2.2"
     xmlns:atom="http://www.w3.org/2005/Atom">
  <Document>
    <atom:author>
      <atom:name>Angel Tello</atom:name>
    </atom:author>
    <atom:link>
      href="http://bbs.keyhole.com/ubb/download.php?Number=408502" />
.
.
.
  </Document>
</kml>
```

Чтобы воспользоваться этими элементами, необходимо включить в KML-файл ссылку на пространство имен Atom. Это делается в теге <kml> сразу после объявления пространства имен KML:

```
<kml xmlns="http://www.opengis.net/kml/2.2"
     xmlns:atom="http://www.w3.org/2005/Atom">
.
.
.
</kml>
```

Что дальше?

В следующей главе мы воспользуемся тем, что уже знаем о стилизации информационных окон, линий и многоугольников. Мы познакомимся с *разделяемыми стилями* – мощным механизмом разработки единообразных стилей презентации, который легко применить к широкому набору деталей. Мы расскажем также о стилях значков и надписей.

Глава 4. Стили и значки

Прочитав эту главу, вы сможете:

- ✓ перечислить все шесть элементов, определяющих подстили для детали KML;
- ✓ создать разделяемый стиль, применяющий один и тот же цвет ко всем многоугольникам;
- ✓ определить, чему отдается предпочтение в случае, когда в разделяемом и внутреннем стиле определены различные значения одного и того же элемента SubStyle;
- ✓ создать собственный простой стиль <BalloonStyle>;
- ✓ создать элемент <StyleMap>, который будет определять внешний вид значка при наведении на него мыши.

Предварительный обзор

Стили позволяют определить внешний вид значков, линий, многоугольников, информационных окон и элементов списка. Можно создать общий стиль для всех линий, многоугольников и информационных окон в презентации или несколько разных стилей для представления семантически различной информации. В KML все детали и геометрические элементы описываются вместе со стилями. В KML нет механизма стилизации, аналогичного CSS (каскадные таблицы стилей) или SLD (Styled Label Descriptor), который позволил бы отделить содержимое от представления. Дополнительную информацию по этому вопросу см. в разделе «Определение внешних стилей» ниже.

Стиль на самом деле представляет собой набор следующих *подстилей*:

- ❑ <IconStyle>;
- ❑ <LabelStyle>;
- ❑ <LineStyle>;
- ❑ <PolyStyle>;
- ❑ <BalloonStyle>;
- ❑ <ListStyle>.

В этой главе приводится описание всех подстилей с примерами использования. В главе 2 мы уже познакомились с элементами <BalloonStyle> и <IconStyle>, а в главе 3 – с <LineStyle> и <PolyStyle>, поэтому здесь они описываются конспективно. Вслед за обсуждением шести стилей-кирпичиков мы рассмотрим основные концепции, связанные со стилями:

- ❑ использование *внутренних* (inline) и *разделяемых* (shared) стилей, позволяющих эффективно применить один стиль ко многим деталям;
- ❑ как внутренний стиль позволяет *переопределить* свойства, заданные в разделяемом стиле (или принимаемые по умолчанию);

- различные способы задания элемента <styleUrl>;
- *карты стилей* – средство, позволяющее программе Google Earth реагировать на событие наведения мыши на точечную метку.

Путешествие по литературным произведениям

Джером Берг (Jerome Burg) разработал интересный способ изучения литературы на основе Google Earth, назвав его Google Lit Trips (http://googlelittrips.com). Ориентированная прежде всего на детей от 5 до 12 лет, эта программа показывает на карте пути, по которым следовали герои таких знаменитых произведений, как «Одиссея» Гомера, «Энеида» Вергилия, «Кандид» Вольтера, «Гроздья гнева» Стейнбека и «Портрет художника в юности» Джойса. Во всплывающих окнах представлены многочисленные факты, изображения и ссылки по пути следования. В результате книга оживает перед сидящими в виртуальном классе. На рис. 4.1 представлен маршрут Джоуда из «Гроздьев гнева».

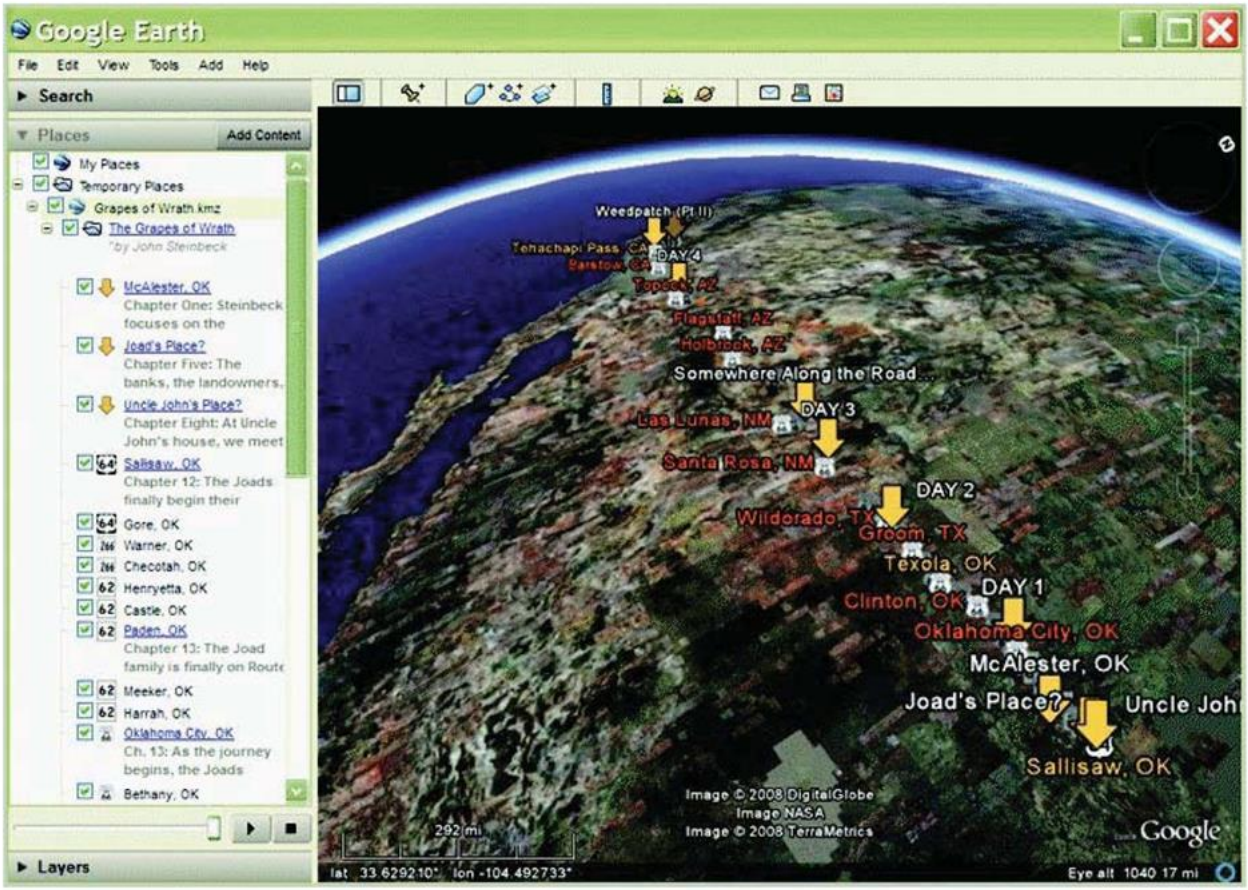


Рис. 4.1. Приложение Google Lit Trips дает возможность обучающемуся повторить многотрудное странствие семьи Джоуда в Калифорнию, описание в книге Стейнбека «Гроздья гнева». Подробные описания и ссылки на внешние ресурсы в информационных окнах позволяют глубже понять роман. (http://googlelittrips.com)

Составные части стилей

Элемент подстиля всегда вложен в элемент <Style>, который может содержать 0 или 1 подстиль каждого вида. В первой части главы мы рассмотрим синтаксис всех подстилей и приведем простые примеры их использования. Вот как выглядит синтаксис самого элемента <Style>:

```
<Style id="ID">
  <IconStyle>...</IconStyle>
  <LabelStyle>...</LabelStyle>
  <LineStyle>...</LineStyle>
  <PolyStyle>...</PolyStyle>
  <BalloonStyle>...</BalloonStyle>
  <ListStyle>...</ListStyle>
</Style>
```

Стиль не обязательно должен включать все подстили. Если какой-то подстиль не задан, используются значения, принимаемые по умолчанию. Элементы-потомки <Style> должны следовать в указанном выше порядке.

Элемент <Style> может быть потомком любого элемента, производного от Feature (см. раздел «Элементы, производные от Feature» в главе 2). Если <Style> используется таким образом, то он называется *внутренним стилем* (inline style). О других способах использования элемента <Style> см. раздел «Разделяемые и внутренние стили» ниже.

Иерархия наследования стилей

На рис. 4.2 представлена часть дерева элементов KML, содержащая классы подстилей. Элемент ColorStyle является *абстрактным*, то есть сам по себе никогда не встречается в KML-файле. Он предоставляет набор относящихся к цвету элементов, общих для всех четырех подстилей, находящихся на рис. 4.2 справа от него. Элементы <LineStyle>, <PolyStyle>, <IconStyle> и <LabelStyle> наследуют от ColorStyle следующие элементы:

- <color>;
- <colorMode>.

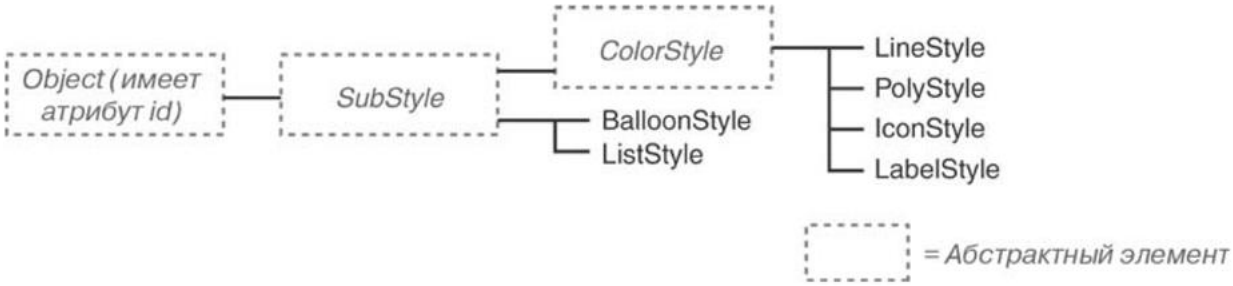


Рис. 4.2. Иерархия наследования элементов подстилей

Элементы, расположенные правее в дереве, называются *производными* от тех, что находятся слева от них.

Примечание

В описаниях синтаксиса составных элементов перечисляются все возможные потомки, поэтому помнить, кто кому наследует, необязательно. Мы затронули эту тему только для того, чтобы вы понимали, как работает иерархия элементов в KML.

Сначала рассмотрим четыре элемента, производных от ColorStyle.

Элемент LineStyle

Элемент <LineStyle> позволяет задать цвет и толщину линий, образующих *ломаные* (пути), контуры многоугольников и экструдированные отрезки, исходящие из точек. Ниже приведен синтаксис этого элемента:

```
<LineStyle id="ID">
  <color>ffffffff</color>
  <colorMode>normal</colorMode>
  <width>1.0</width>
</LineStyle>
```

Подробнее о задании элементов <color> и <width> см. главу 2.

Выбор случайного цвета

Элемент <colorMode> может принимать значение `normal` (по умолчанию) или `random`. Если он равен `normal`, то цвет линии определяется элементом <color>. Если же он равен `random`, то геобраузер вправе выбрать произвольный цвет при каждой загрузке KML-файла. Чтобы получить полностью случайные непрозрачные цвета, присвойте элементу <color> белый цвет (ffffffff). В примере PolyStyle из этой главы показано, как задавать случайный цвет. Дополнительные сведения о рандомизации цветов см. в описании элемента <ColorStyle> в приложении А.

Пример

В примере Lit Trips элемент LineStyle применяется для представления пути Джоуда линией оранжевого цвета толщиной 3 пикселя (рис. 4.3). В файле *LineStyle.kml* приведен код этого простого элемента <LineStyle>.

LineStyle.kml

```
<?xml version="1.0" encoding="utf-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Document>
    <name>LineStyle Example</name>
    <Placemark>
      <Style>
        <LineStyle>
          <color>ff0080ff</color>
          <width>3</width>
        </LineStyle>
      </Style>
      <LineString>
        <tessellate>1</tessellate>
        <coordinates>
          -119.039568538709,35.3512349410626,0
          -119.039393055414,35.3620902295471,0
```

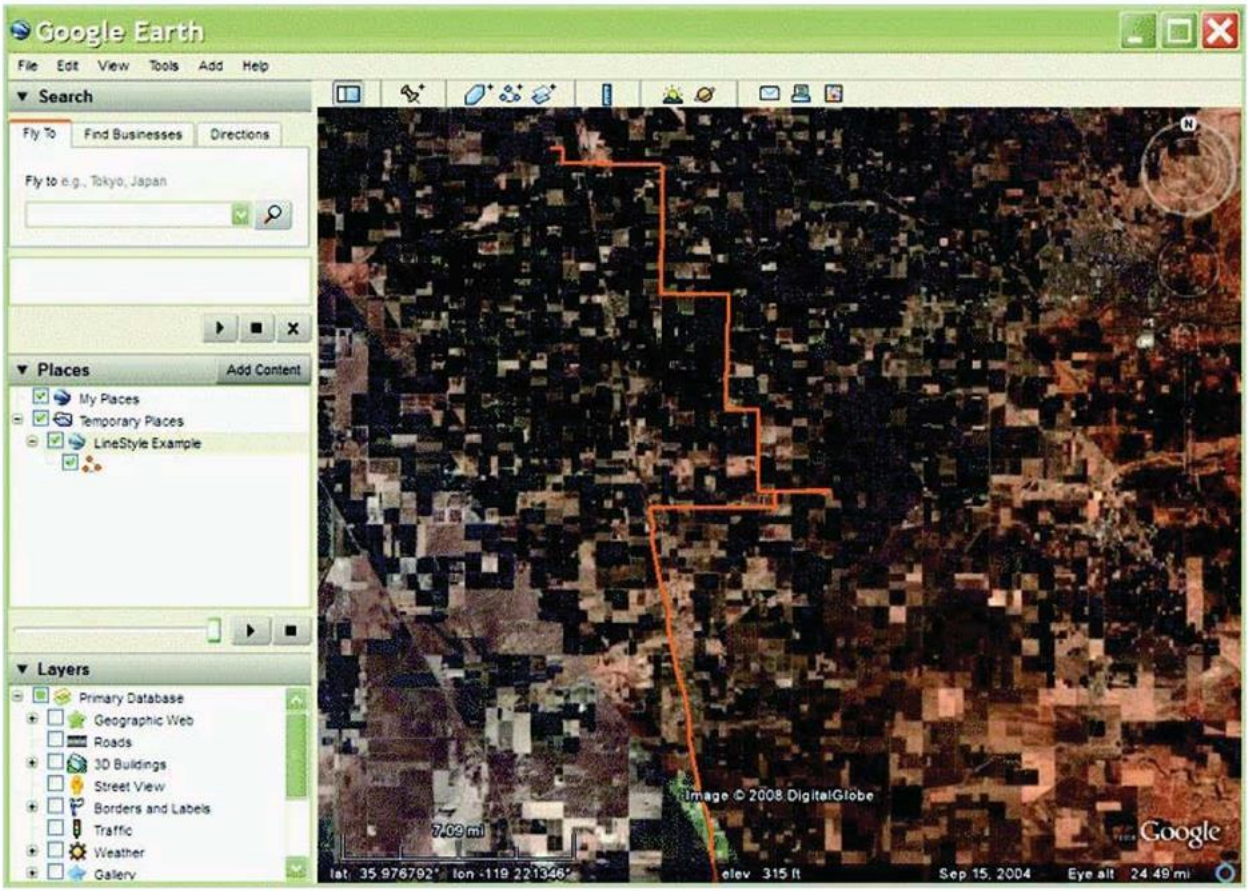


Рис. 4.3. Элемент <LineStyle>, задающий оранжевый цвет и толщину 3 пикселя

```
          -119.039826751439,35.3631086344801,0
          -119.04366070921,35.3721324303697,0
          .
          .
          .
          -119.347536971146,36.1306716810123,0
        </coordinates>
      </LineString>
    </Placemark>
  </Document>
</kml>
```

Элемент PolyStyle

Элемент PolyStyle позволяет задать цвет внутренности многоугольника. У него могут быть потомки, говорящие о том, нужно ли заливать многоугольник цветом и оконтуривать его. Если <outline> равен 1, то контур многоугольника рисуется цветом, заданным в элементе LineStyle.

Замечание о булевских значениях

По определению, булевское значение может быть истинным или ложным. В KML для представления значения «истина» («да») применяется 1 или слово `true`, а для представления значения «ложь» («нет») – 0 или `false`. Элементы <fill>

и <outline>, являющиеся потомками <PolyStyle>, – это примеры булевских значений.

Вот как выглядит синтаксис этого элемента:

```
<PolyStyle id="ID">
  <color>ffffffff</color>
  <colorMode>normal</colorMode>
  <fill>1</fill>
  <outline>1</outline>
</PolyStyle>
```

Значения по умолчанию

В описаниях синтаксиса, встречающихся в этой книге, указываются значения элементов, принимаемые по умолчанию. Если некоторый элемент опущен, то геобраузер использует определенное для него значение по умолчанию.

Пример

В файле *PolyStyle.kml* приведена карта штата Оклахома, образованная многоугольником с оранжевым цветом контура и элементом <colorMode>, равным `random`. Отметим, что цвет контура определяется элементом <color>, вложенным в <LineStyle>. На рис. 4.4 показано, как выглядит многоугольник, залитый случайным цветом.

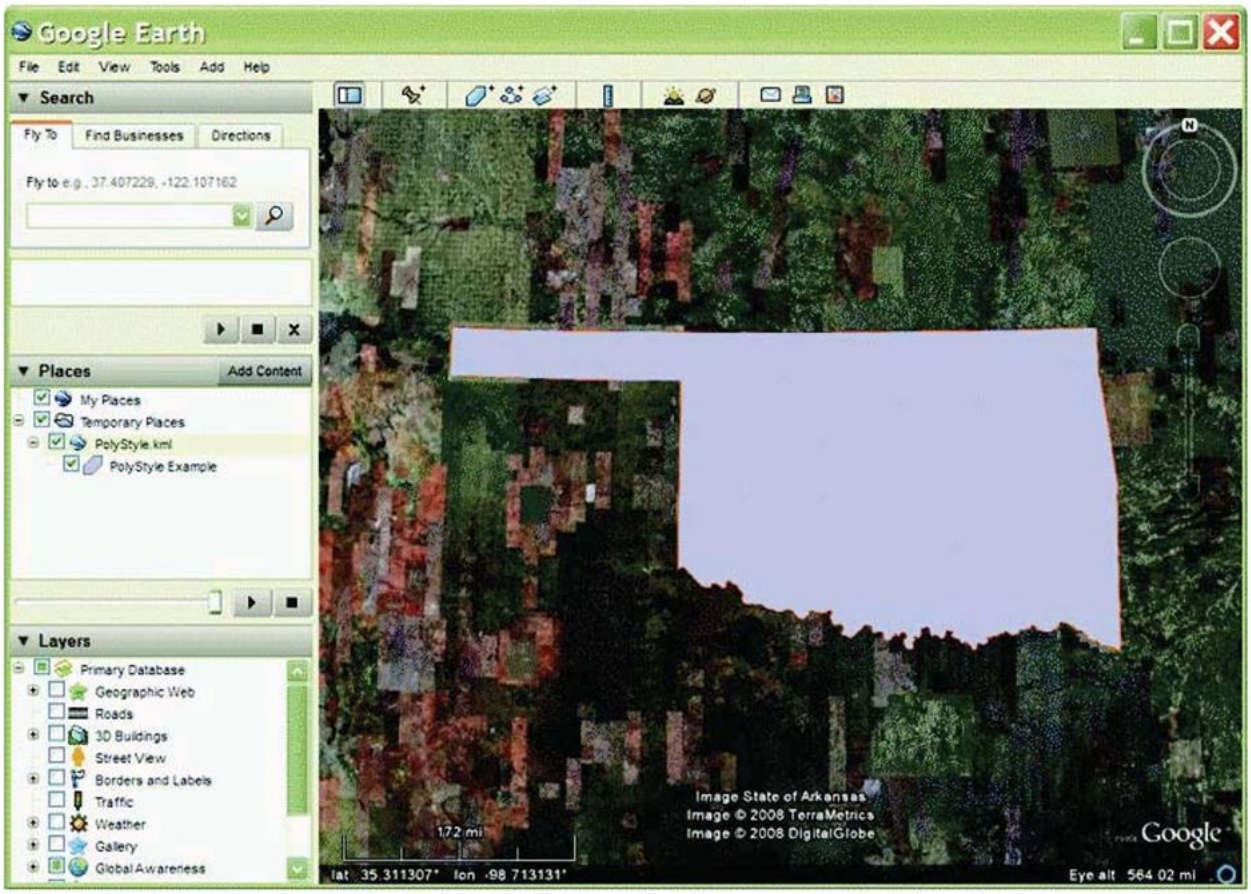


Рис. 4.4. Пример многоугольника <Polygon>, залитого случайным цветом

PolyStyle.kml

```
<?xml version="1.0" encoding="utf-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Document>
    <Placemark>
      <Style>
        <LineStyle>
          <color>ff0080ff</color>     <!-- оранжевый ->
          <width>3</width>
        </LineStyle>
        <PolyStyle>
          <colorMode>random</colorMode>
        </PolyStyle>
      </Style>
      <Polygon>
        <outerBoundaryIs>
          <LinearRing>
            <coordinates>
              -94.4393221493475,34.9291508772006,0
              -94.4285520120899,35.4005462695086,0
              -94.4684852110367,35.6410882624306,0
              -94.485934826059,35.7603104941298,0
              -94.5424172738563,36.1068358038929,0
              -94.5531136198804,36.1645252110653,0
              .
              .
              .
              -94.4393221493475,34.9291508772006,0
            </coordinates>
          </LinearRing>
        </outerBoundaryIs>
      </Polygon>
    </Placemark>
  </Document>
</kml>
```

Элемент IconStyle

Элемент <IconStyle> определяет способ рисования точечных меток. Файл с изображением значка задается в элементе <Icon>. Вложенный в него элемент <href> может содержать путь к локальному файлу или URL-файлу в Сети. Вот как выглядит синтаксис этого элемента:

```
<IconStyle id="ID">
  <color>ffffffff</color>
  <colorMode>normal</colorMode>
  <scale>1.0</scale>
  <heading>1.0</heading>
  <Icon>
    <href>...</href>
  <Icon>
  <hotSpot x="0.5" y="0.5"
    xunits="fraction" yunits="fraction"/>
</IconStyle>
```

Потомки элемента <IconStyle> определяются следующим образом:

<color>

Заданный здесь цвет (в формате *aabbggrr*) смешивается с цветом изображения, который указан в элементе <href>, вложенном в <Icon>.

<colorMode>

Может принимать значение `normal` или `random`. Значение `normal` означает, что нужно смешать цвет, заданный элементом <color>, с цветом значка, а значение `random` – что с цветом значка смешивается случайно выбранный цвет. По умолчанию подразумевается `normal`.

<scale>

Изменяет размер значка. Изображение значка (заданное потомком <href> элемента <Icon>) масштабируется с указанным коэффициентом в обоих направлениях. В случае значения 1 (принимается по умолчанию) значок не изменяется.

<heading>

Задает угол поворота по часовой стрелке. 0 означает север (не поворачивается), 90 – восток, 180 – юг (изображение перевернуто), 270 – запад. На рис. 4.5 иллюстрируются все четыре значения.

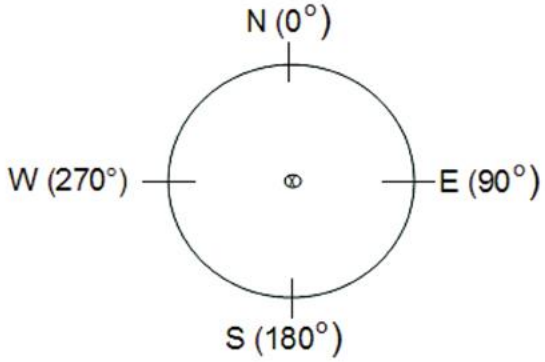


Рис. 4.5. Значения элемента <heading> задаются по часовой стрелке, причем 0 соответствует направлению на север

<Icon>

<href>

Задает изображение нестандартного значка. Элемент <Icon> может иметь единственного потомка <href>, который задает URL или путь к локальному файлу с изображением. Локальный файл может находиться на диске или в KMZ-архиве. Пути могут быть абсолютными или относительными. Как правило, указывается относительный путь, что позволяет упаковать значок и KML-файл в KMZ-архив, который можно отправить по почте или переместить на другой сервер, не изменяя путей. Абсолютные ссылки применяются, когда значки берутся из стандартного репозитория, который редко изменяется.

<hotSpot x="0.5" y="0.5" xunits="fraction" yunits="fraction" />

Элемент <hotSpot> определяет точку внутри значка, которая совмещается с точкой, указанной в элементе <Point> метки. По умолчанию таковой является центр значка. Но в некоторых случаях, например когда значок представляет собой стрелку, совмещать необходимо какую-то другую точку

(скажем, острие стрелки). Если пользователь будет наклонять или поворачивать панораму, то значок будет «вертеться» вокруг точки <hotSpot>. Элемент <hotSpot> можно задать одним из трех способов: в виде *количества пикселей*, в виде *доли* от размера значка или в виде *смещения* от правого верхнего угла значка. В примере из этой главы используется количество пикселей. (О других способах см. описание элемента <IconStyle> в приложении А.) Отметим, что внутри тега <hotSpot> находятся пары имя/значение. В XML такие пары называются *атрибутами*. Значение атрибута заключается в кавычки. Обратите также внимание на то, что весь элемент <hotSpot> записывается в виде одного самозакрывающегося тега:

```
<hotSpot x="5" y="9" xunits="pixels" yunits="pixels" />
```

Такая запись в точности эквивалентна следующей:

```
<hotSpot x="5" y="9" xunits="pixels" yunits="pixels"></hotSpot>
```

Активная точка значка

Ниже приведен простой пример, демонстрирующий использование активной точки (hotspot) значка. В этом KML-коде определяется нестандартный значок. Он взят из предлагаемой программой Google Earth коллекции значков, которая находится на сайте http://maps.google.com и доступна из интерфейса Google Earth. Значок масштабируется с коэффициентом 1.2 (то есть увеличивается):

```
<IconStyle>
  <scale>1.2</scale>
  <Icon>
    <href>http://maps.google.com/mapfiles/kml/shapes/arrow.png</href>
  </Icon>
</IconStyle>
```

Слева на рис. 4.6 изображена стрелка с подразумеваемым по умолчанию положением активной точки, когда центр значка совмещается с точкой метки. Если мы хотим, чтобы на метку указывало острие стрелки, то необходимо сделать острие активной точкой. Вот один из способов добиться этого:

```
<IconStyle>
  <scale>1.2</scale>
  <Icon>
    <href>http://maps.google.com/mapfiles/kml/shapes/arrow.png</href>
  </Icon>
  <hotSpot x="32" y="1" xunits="pixels" yunits="pixels" />
</IconStyle>
```

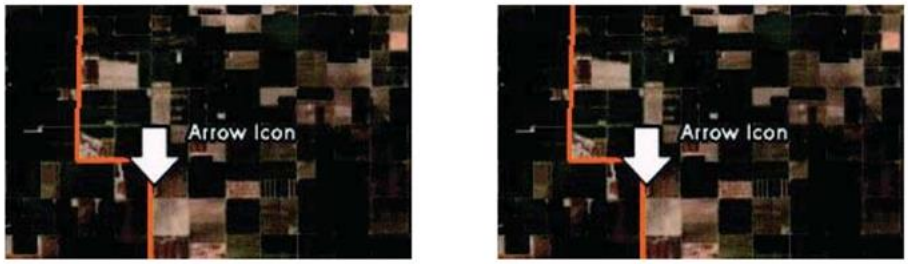


Рис. 4.6. Нестандартный значок с размещением по умолчанию (левый рисунок) и с определенной в элементе <hotSpot> активной точкой (правый рисунок)

Справа на рис. 4.6 показан результат этой модификации, теперь острие стрелки совпадает с точкой метки.

Большинство браузеров и приложений для обработки фотографий показывают размеры изображения в пикселях. Так, браузер Mozilla Firefox включает информацию о размерах в заголовок окна, в котором отображается картинка (рис. 4.7).

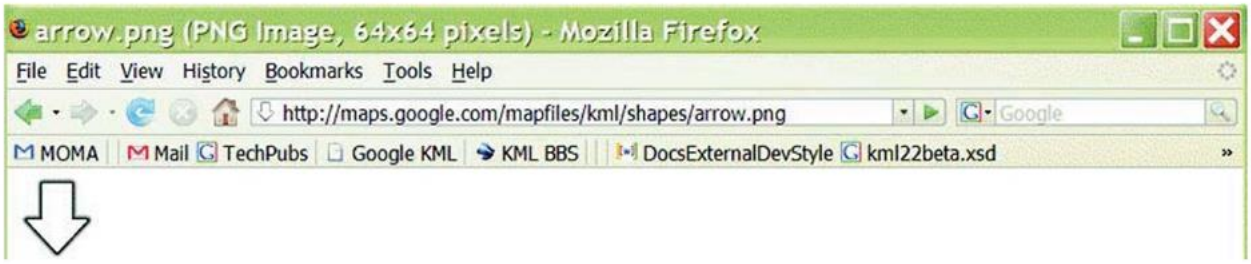


Рис. 4.7. Браузер Mozilla Firefox выводит информацию о формате и размере изображения (64 × 64 пикселя) в заголовке окна

На рис. 4.8 видно, что начало системы координат значка расположено в левом нижнем углу изображения, при этом ось x направлена вправо, а ось y – вверх. Расстояния по обеим осям измеряются в пикселях.

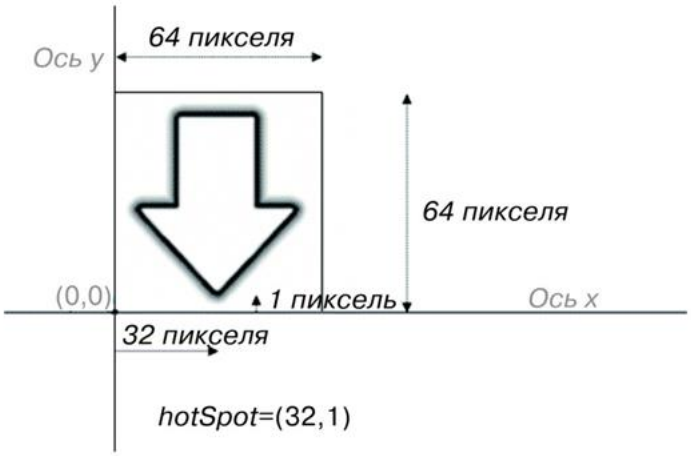


Рис. 4.8. Задание активной точки значка в пикселях. Начало координат находится в левом нижнем углу изображения

Пример

В презентации Google Lit Trips есть несколько примеров нестандартных значков: стрелка и значок исторического Маршрута 66[1] (рис. 4.9). В файле *IconStyle.kml* приведен код создания этих двух меток и значков, для которых определены разные стили.

---

[1] Первый трансконтинентальный маршрут, который протянулся через Америку с востока на юг, соединив Лос-Анджелес и Чикаго. В романе «Гроздья гнева» Дж. Стейнбек назвал его «Матерью американских дорог» – *Прим. перев.*

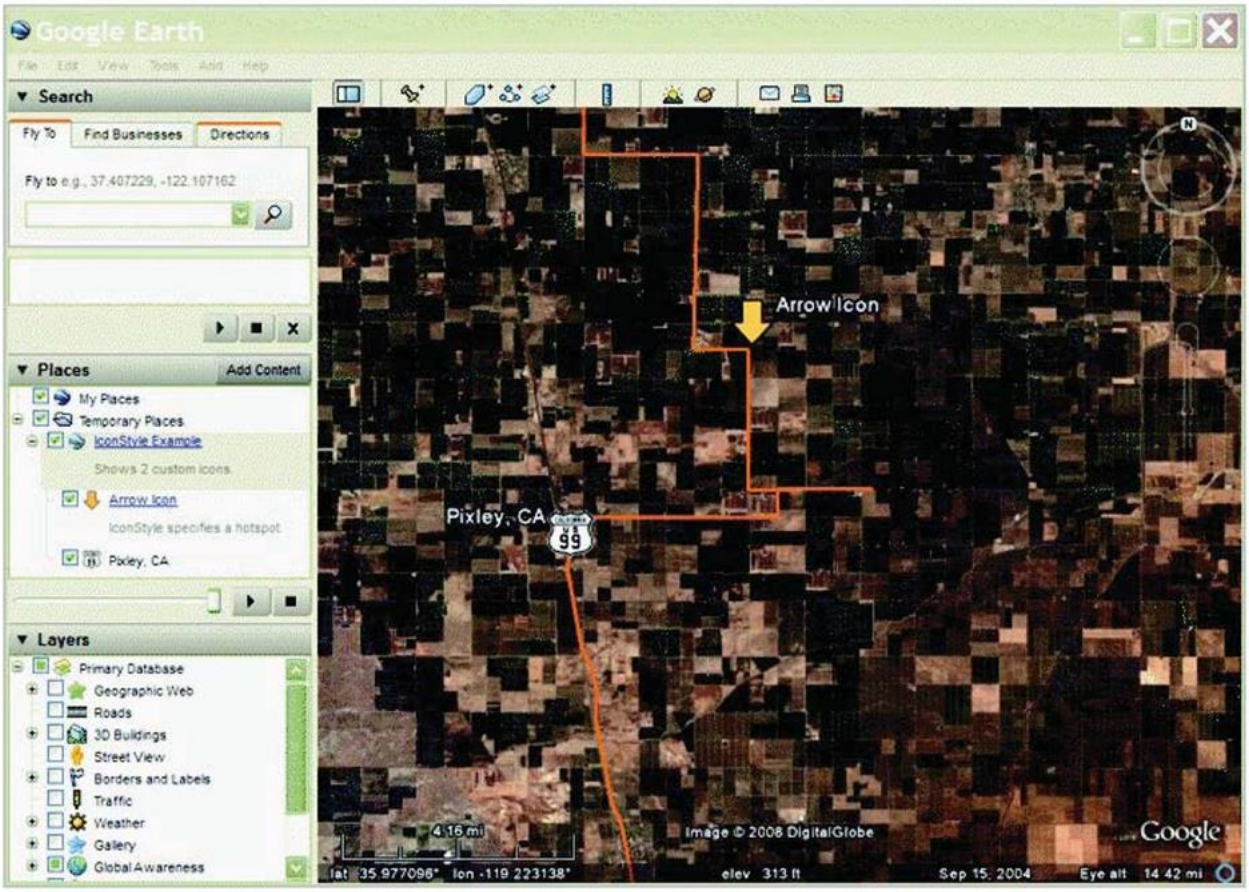


Рис. 4.9. Два нестандартных значка с различными внутренними стилями <IconStyle>

IconStyle.kml

```
<?xml version="1.0" encoding="utf-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Document>
    <name>Пример IconStyle</name>
    <open>1</open>
    <description>Демонстрируем 2 нестандартных значка.</description>
    <Placemark>
      <name>Значок стрелки</name>
      <description>В IconStyle задана активная точка.</description>
      <Style>
        <IconStyle>
          <color>ff66ccff</color>
          <scale>1.2</scale>
          <Icon>
            <href>http://maps.google.com/mapfiles/kml/shapes/arrow.png</href>
          </Icon>
          <hotSpot x="32" xunits="pixels" y="1" yunits="pixels"/>
        </IconStyle>
      </Style>
      <Point>
        <coordinates>-119.232195463845,36.0160219116952,0</coordinates>
      </Point>
    </Placemark>
    <Placemark>
      <name>Пиксли, штат Калифорния</name>
      <Style>
        <IconStyle>
          <scale>1.2</scale>
```

```
          <Icon>
            <href>http://www.gbcnet.com/ushighways/logos/99_old_shield.gif</href>
          </Icon>
        </IconStyle>
        <LineStyle>
          <color>ff0080ff</color>
          <width>3</width>
        </LineStyle>
        <PolyStyle>
          <colorMode>random</colorMode>
        </PolyStyle>
      </Style>
      <MultiGeometry>
        <LineString>
        .
        .
        .
        </LineString>
        <Polygon>
        .
        .
        .
        </Polygon>
        <Point>
          <coordinates>-119.290316474804,35.9670763464666,0</coordinates>
        </Point>
      </MultiGeometry>
    </Placemark>
  </Document>
</kml>
```

Элемент LabelStyle

Элемент <LabelStyle> определяет способ рисования имени <name> детали на трехмерной панораме. Можно задать цвет, режим интерпретации цвета и масштаб для надписи.

```
<LabelStyle id="ID">
  <color>ffffffff</color>
  <colorMode>normal</colorMode>
  <scale>1.0</scale>
</LabelStyle>
```

Потомки элемента <LabelStyle> определяются следующим образом:

<color>

Цвет надписи (в формате *aabbggrr*). По умолчанию принимается непрозрачный белый цвет (ffffffff).

<colorMode>

Может принимать значение `normal` или `random`. Значение `normal` означает, что нужно использовать цвет, заданный элементом <color>, а значение `random` – что цвет выбирается случайно. По умолчанию подразумевается `normal`.

<scale>

Надпись масштабируется с указанным коэффициентом. Если задано значение 1 (принимается по умолчанию), то размер надписи не изменяется.

В примере LabelStyle.kml надпись сделана зеленой, а ее размер чуть больше принимаемого по умолчанию (рис. 4.10).

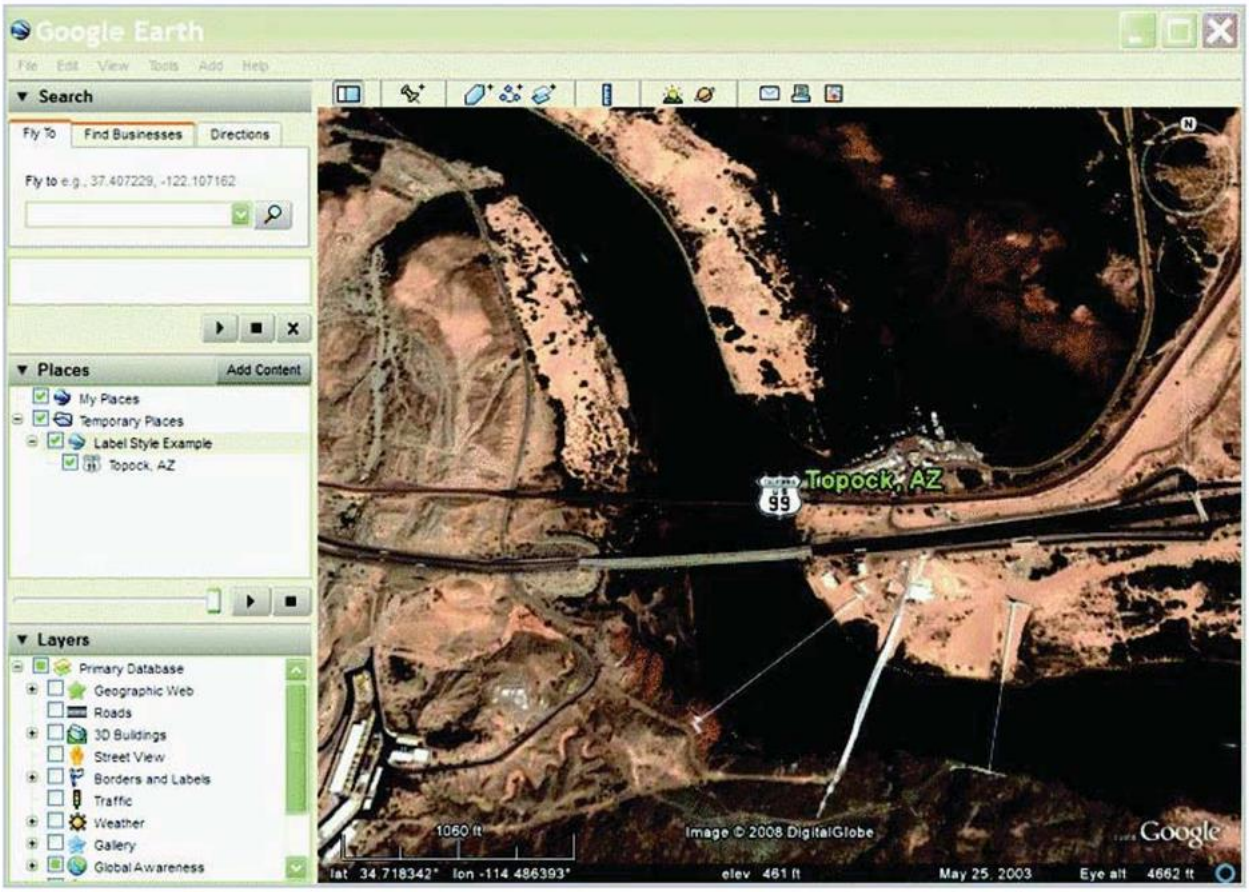


Рис. 4.10. Использование <LabelStyle> для раскрашивания и масштабирования надписи метки. В качестве текста надписи фигурирует значение элемента <name>

LabelStyle.kml

```
<?xml version="1.0" encoding="utf-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Document>
    <name>Пример LabelStyle</name>
    <Placemark>
      <name>Топок, штат Аризона</name>
      <open>1</open>
      <LookAt>
        <longitude>-114.486391641715</longitude>
        <latitude>34.7183318048167</latitude>
        <altitude>0</altitude>
        <heading>9.72352284417441e-15</heading>
        <tilt>3.06638923748042e-12</tilt>
        <range>9489.36670378269</range>
      </LookAt>
      <Style>
        <IconStyle>
          <scale>1.2</scale>
          <Icon>
            <href>http://www.gbcnet.com/ushighways/logos/99_old_shield.gif</href>
          </Icon>
        </IconStyle>
        <LabelStyle>
          <color>ff00ffaa</color>
          <scale>1.2</scale>
        </LabelStyle>
      </Style>
```

```
      <Point>
        <coordinates>-114.486391641715,34.7183318048167,0</coordinates>
      </Point>
    </Placemark>
  </Document>
</kml>
```

Элемент BalloonStyle

Обычно информационные окна ассоциируются с метками, но напомним, что допустимо ассоциировать их с любыми элементами KML, производными от Feature (*деталями*) (<Document>, <Folder>, <NetworkLink>, <ScreenOverlay>, <GroundOverlay> и <PhotoOverlay>; см. раздел «Элементы, производные от Feature» в главе 2). Элемент <BalloonStyle>, с которым мы познакомились в главе 2, описывается следующим синтаксисом:

```
<BalloonStyle id="ID">
  <bgColor>ffffffff</bgColor>
  <textColor>ff000000</textColor>
  <text>...</text>
  <displayMode>default</displayMode>
</BalloonStyle>
```

Потомки элемента <BalloonStyle> определяются следующим образом:

<bgColor>

Цвет фона информационного окна (в формате *aabbggrr*). По умолчанию принимается непрозрачный белый цвет (ffffffff).

<textColor>

Цвет текста в информационном окне (в формате *aabbggrr*). По умолчанию принимается черный цвет (ff000000).

<text>

Текст, отображаемый в информационном окне.

Внутри тега <text> могут находиться компоненты в следующем формате, позволяющем ссылаться на дочерний элемент типа Feature: `$[name]`, `$[description]`, `$[address]`, `$[id]`, `$[Snippet]`. Программа Google Earth ищет в текущей детали Feature элемент, соответствующий компоненту, и подставляет информацию из него в информационное окно. Компоненты очень полезны при разработке разделяемых стилей (см. раздел «Разделяемые и внутренние стили» ниже).

Чтобы включить в информационное окно маршрут *Сюда – Отсюда*, воспользуйтесь компонентом `$[geDirections]`. Чтобы предотвратить появление маршрута в информационном окне, включите элемент <text> с каким-нибудь содержимым или с компонентом `$[description]`, вместо которого будет подставлено содержимое элемента <description> из детали Feature.

Если элемент <text> не задан, то Google Earth отображает информационное окно по умолчанию, включая в него:

- имя <name>, заданное в детали;
- описание <description>, заданное в детали;

- стандартные ссылки *Сюда – Отсюда*, позволяющие вывести маршрут;
- текст отображается на белом фоне, а «хвостик» окна указывает на точку с координатами, заданными в детали (при условии их наличия).

<displayMode>

Может принимать значение `default` или `hide`. Если равен `default`, то в информационном окне отображается информация из элемента <text>. Если же режим <displayMode> равен `hide`, то информационное окно вообще не появляется. В программе Google Earth щелчок по значку в списке, соответствующему метке, для которой элемент <displayMode> равен `hide`, заставляет Google Earth «перелететь» к этой метке.

В примере *BalloonStyle.kml* элемент <BalloonStyle> добавлен в папку <Folder>. Щелчок по имени папки в списке **Метки** открывает это информационное окно. Для окна задан желтовато-коричневый цвет фона и коричневый цвет текста. Имя <name> и описание <description> заданы в отдельном элементе Feature и подставляются вместо компонентов. На рис. 4.11 показано, как этот пример визуализируется в Google Earth.

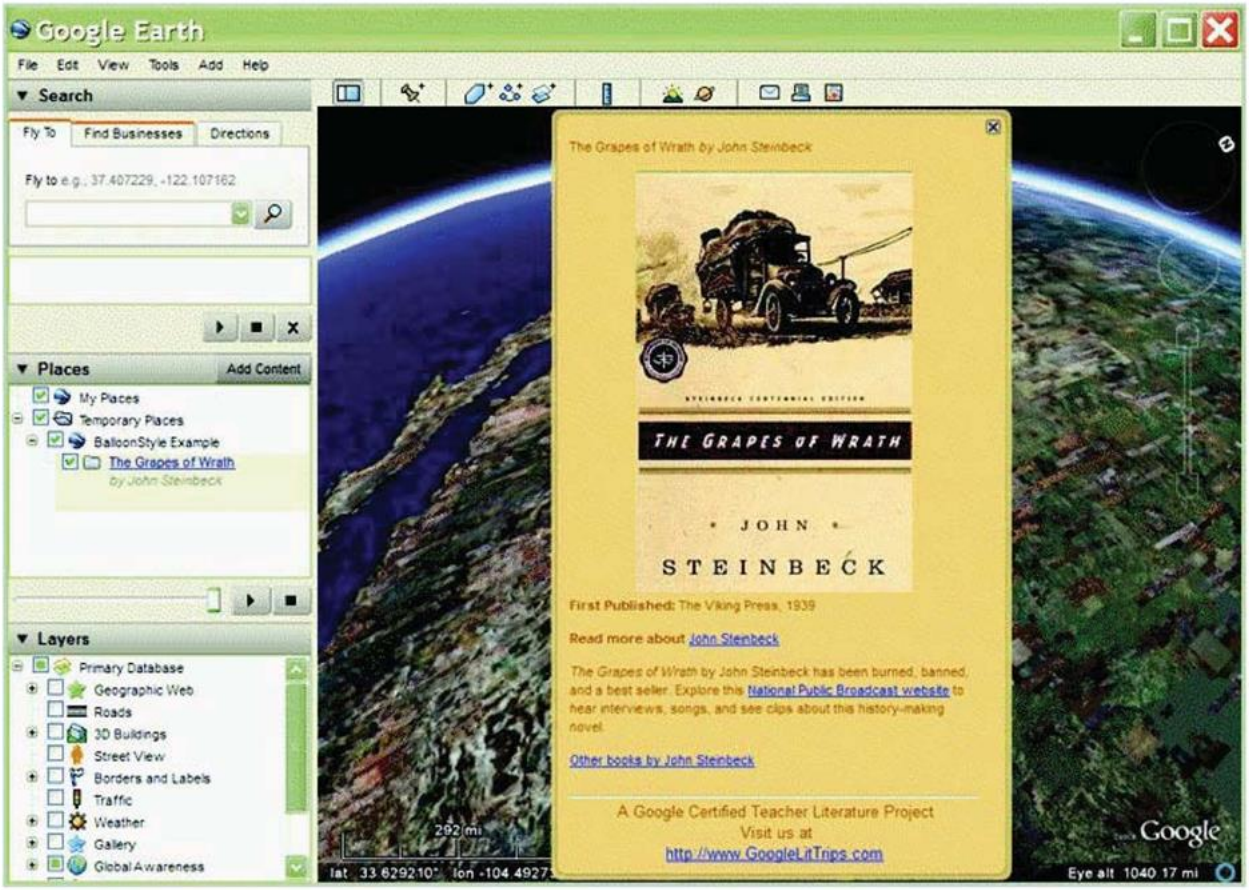


Рис. 4.11. Элемент <BalloonStyle> применяется для задания цвета фона и текста, текстовых элементов и верстки информационного окна, потенциально разделяемой многими деталями в KML-файле

BalloonStyle.kml

```
<?xml version="1.0" encoding="utf-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Document>
    <name>Пример BalloonStyle</name>
    <open>1</open>
    <Style id="tan_balloon_style">
      <BalloonStyle id="ID">
        <bgColor>ff87d1ec</bgColor>
        <textColor>ff0055aa</textColor>
        <text>
        <!-- имя и описание подставляются из другой детали ->
          <![CDATA[
            $[name]
            $[description]
          ]]>
        </text>
      </BalloonStyle>
    </Style>
    <Folder>
      <name>Гроздья гнева</name>
      <description><![CDATA[<i>Джон Стейнбек</i>
        <p><center>
        <img src="http://nmxs-images.forbesautos.com/streamer/advice/
           toptens/cars_literature/grapes_wrath2?WID=225
        "><br></center>
        <B>Первое издание: </b>The Viking Press, 1939
        <p>
        <B>Дополнительная информация </b><a href="http://nobelprize.org/
           nobel_prizes/literature/laureates/1962/steinbeck-bio.html">
        Джон Стейнбек</a>
        <p>
        Книга <i>Гроздья гнева</i> Джона Стейнбек сжигалась, запрещалась и была
    бестселлером. На сайте <a href="http://www.npr.org/programs/morning/features/
    patc/grapesofwrath/"> National Public Broadcast</a>можно прослушать интервью,
    песни и посмотреть клипы, посвященные этому вошедшему в историю роману.
        <p>
        <a href="http://books.google.com/books?q=Books+by+John+
           Steinbeck&ots=dpyZMKB6wY&sa=X&oi=print&ct=title">
        Другие книги Джона Стейнбека</a>
        <center><hr>
        <font style="font-size: 10pt" >Сертифицированный Google проект изучения
          литературы<br>Посетите наш сайт<br>
       http://www.GoogleLitTrips.com</font></center>]]></description>
      <LookAt>
        <longitude>-104.492732803477</longitude>
        <latitude>33.6292096680936</latitude>
        <altitude>0</altitude>
        <heading>-58.2589615598971</heading>
        <tilt>38.1065722637815</tilt>
        <range>2005783.42286775</range>
      </LookAt>
      <styleUrl>#tan_balloon_style</styleUrl>
    </Folder>
  </Document>
</kml>
```

Дополнительную информацию о шаблонах информационных окон см. в главе 8 «Большие наборы данных». Вы можете определить в KML собственные элементы данных и потом ссылаться на них с помощью $-компонент из элемента <text> точно так же, как это позволяют делать встроенные компоненты. См. пример элемента <Data> в главе 8.

Элемент ListStyle

Этот элемент определяет, как детали представляются в списке. В программе Google Earth список находится на панели **Метки** в левой части окна (рис. 4.12).

```
<ListStyle id="ID">
  <listItemType>check</listItemType>
  <bgColor>ffffffff</bgColor>
  <ItemIcon>
    <state>open</state>
    <href>...</href>
  <ItemIcon>
  <maxSnippetLines>2</maxSnippetLines>
</ListStyle>
```

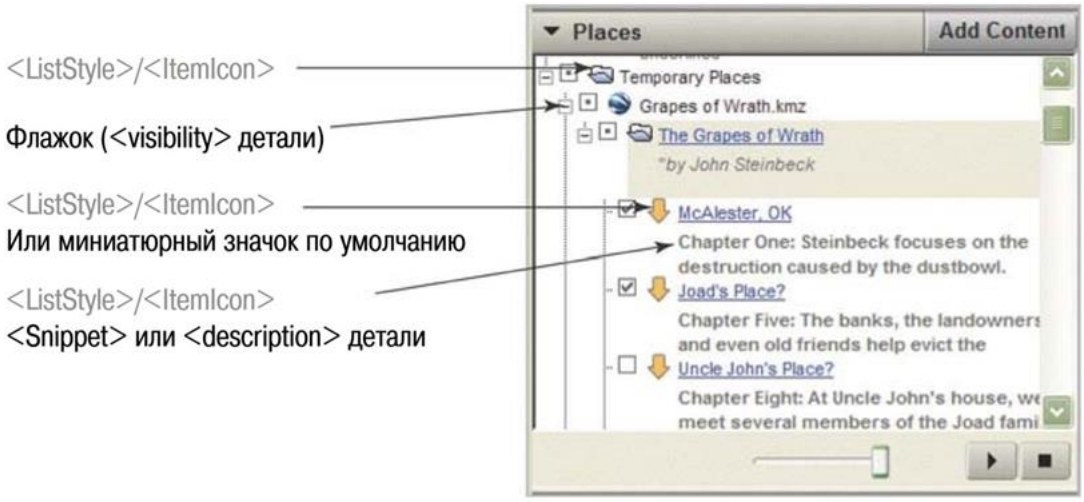


Рис. 4.12. Список в программе Google Earth

Потомки элемента <ListStyle> определяются следующим образом:

<listItemType>

Способ представления детали в списке. Допустимы следущие значения:

`check` (по умолчанию) (применим к любой детали)

Означает, что видимость детали определяется состоянием флажка. Если флажок отмечен, то деталь видима, если сброшен – невидима.

`radioFolder` (применим только к Document и Folder)

Означает, что в каждый момент времени может быть виден только один из потомков контейнера.

`checkHideChildren` (применим только к Document и Folder)

Означает, что нужно скрыть потомков контейнера в списке и включить флажок только для самого родительского контейнера. Если этот флажок отмечен, то все потомки контейнера видимы. Это значение бывает полезно для сетевых ссылок (см. главу 6), когда количество ссылок-потомков велико.

checkOffOnly (применим к любой детали)

Означает, что пользователь может выключить все элементы-потомки с помощью флажка контейнера, но не может все одновременно включить. Каждый элемент-потомок требуется включать по отдельности. Такой режим полезен для папок, содержащих много данных, и для глубоких иерархий сетевых ссылок (см. главу 6).

<bgColor>

Цвет фона для краткого описания в списке.

<ItemIcon>

По умолчанию в списке отображается тот же значок, что на трехмерной панораме. Элемент <ItemIcon> позволяет задать значок специально для списка.

<state>

Задает текущее состояние папки или сетевой ссылки, которая загружается из внешнего источника в Сети. Допустимы значения open, closed, error, fetching0, fetching1 и fetching2. Значения можно объединять, перечисляя их через пробел (не запятую).

<href>

Задает URL изображения значка, которое будет отображаться в списке. См. обсуждение абсолютных и относительных ссылок в разделе «Элемент IconStyle» выше.

<maxSnippetLines>

Задает максимальное число строк краткого описания, отображаемых в списке.

В примере *ListStyle.kml* определен элемент <Style> с именем «lit_trips_style» для папки <Folder> и шести находящихся в ней меток <Placemark>. На рис. 4.13 показано, как этот файл выглядит в программе Google Earth. Загрузите файл, затем откройте его в редакторе и модифицируйте, поочередно задав все возможные значения <listItemType>. После каждого изменения сохраните файл и перезагрузите его, щелкнув правой кнопкой мыши по имени папки в Google Earth и выбрав из меню команду **Восстановить**. Посмотрите, как разные значения <listItemType> влияют на представление меток в списке.

Примечание

Для краткости в этом примере <ListStyle> определен как *разделяемый стиль*. Определение его в виде внутреннего стиля увеличило бы файл на 120 строк, поэтому мы решили воспользоваться этой еще не рассматривавшейся техникой. Подробнее о создании и использовании разделяемых стилей см. следующий раздел.

ListStyle.kml

```
<?xml version="1.0" encoding="utf-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Document>
    <name>Пример ListStyle</name>
    <Style id="lit_trips_style">
      <IconStyle>
        <color>ff66ccff</color>
        <scale>1.4</scale>
        <Icon>
```

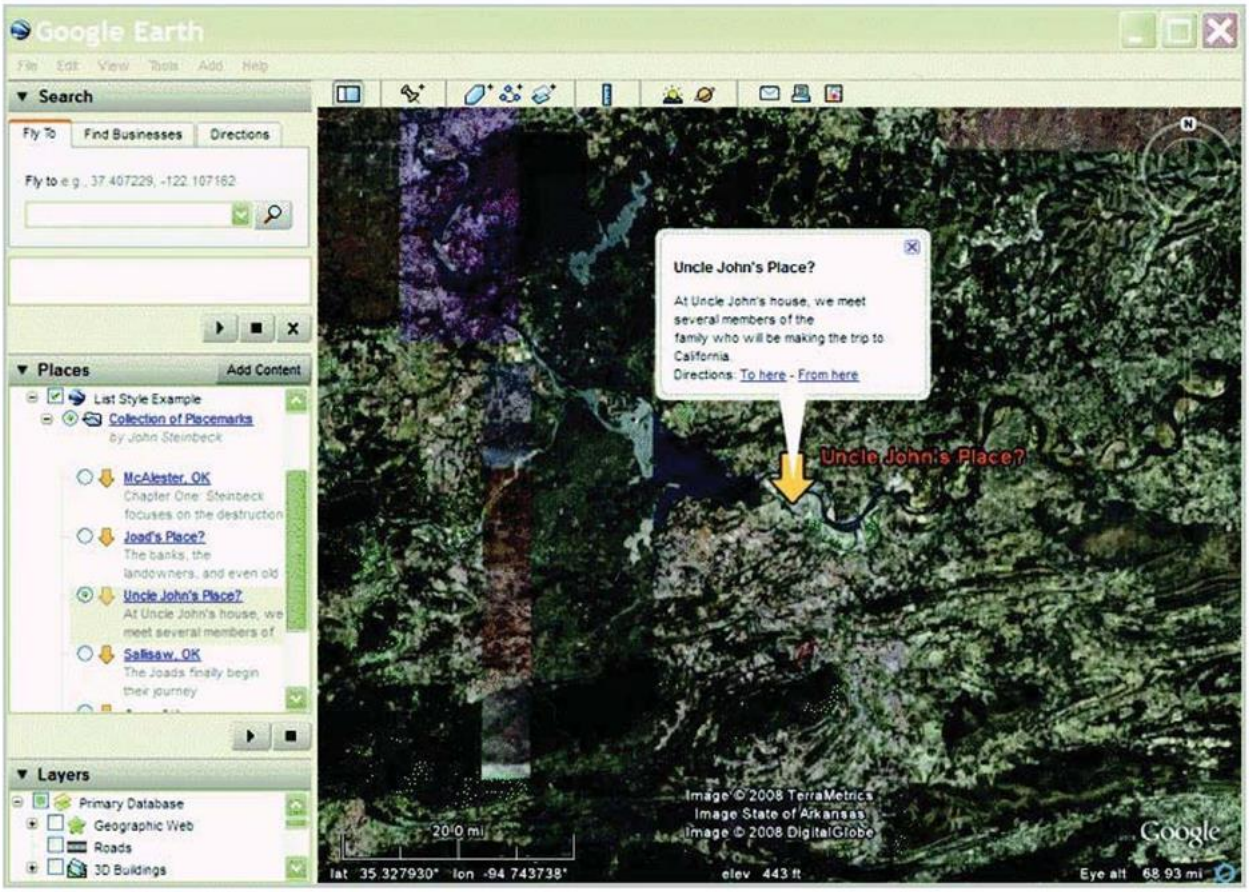


Рис. 4.13. Использование элемента <ListStyle> для создания переключателя меток в папке. В каждый момент времени пользователь может выбрать только одну метку. Кроме того, в <ListStyle> определен желтый фон для кратких описаний деталей в списке

```
        <href>http://maps.google.com/mapfiles/kml/shapes/arrow.png</href>
      </Icon>
      <hotSpot x="32" xunits="pixels" y="1" yunits="pixels"/>
    </IconStyle>
    <LabelStyle>
      <color>ff0080ff</color>
    </LabelStyle>
    <ListStyle>
      <listItemType>radioFolder</listItemType>
      <!-- поэкспериментируйте с другими типами элементов списка: checkOffOnly, ->
      <!-- checkHideChildren и check ->
      <bgColor>ffb0fffc</bgColor>
    </ListStyle>
  </Style>
  <Folder>
    <name>Набор меток</name>
    <description><![CDATA[<i>Джон Стейнбек</i>
    <center>
    <img src="http://nmxs-images.forbesautos.com/streamer/advice/
     toptens/cars_literature/grapes_wrath2?WID=225"><br></center>
    <B>Первое издание: </b>The Viking Press, 1939
    ]]></description>
    <styleUrl>#lit_trips_style</styleUrl>
    <Placemark>
      <name>МакАлистер, штат Оклахома</name>
      <description>Глава 1: Стейнбек рассказывает о разрушениях,
       причиненных пыльной бурей.</description>
```

```
        <styleUrl>#lit_trips_style</styleUrl>
        <Point>
          <coordinates>-95.7694436624162,34.9333328710693,0</coordinates>
        </Point>
      </Placemark>
      <Placemark>
        <name><![CDATA[Место Джоуда?]]></name>
        <description>Банки, землевладельцы и даже старые друзья
          помогают выселять фермеров.</description>
        <styleUrl>#lit_trips_style</styleUrl>
        <Point>
          <coordinates>-94.8617839285501,35.2632608217436,0</coordinates>
        </Point>
      </Placemark>
      <Placemark>
        <name><![CDATA[Место дяди Джона?]]></name>
        <description><![CDATA[В доме дяди Джона мы встречаем нескольких членов семьи,
          которые собираются в Калифорнию.]]></description>
        <styleUrl>#lit_trips_style</styleUrl>
        <Point>
          <coordinates>-94.7253165818111,35.3149226014067,0</coordinates>
        </Point>
      </Placemark>
      <Placemark>
        <name>Саллисо, штат Оклахома</name>
        <description>Семья Джоудс наконец отправляется в путешествие.</description>
        <styleUrl>#lit_trips_style</styleUrl>
        <Point>
          <coordinates>-94.7876387110315,35.459897002806,0</coordinates>
        </Point>
      </Placemark>
      <Placemark>
        <name>Гоур, штат Оклахома</name>
        <styleUrl>#lit_trips_style</styleUrl>
        <Point>
          <coordinates>-95.1169429897411,35.5291660360979,0</coordinates>
        </Point>
      </Placemark>
      <Placemark>
        <name>Уорнер, штат Оклахома</name>
        <styleUrl>#lit_trips_style</styleUrl>
        <Point>
          <coordinates>-95.3009513361342,35.4937000359785,0</coordinates>
        </Point>
      </Placemark>
    </Folder>
  </Document>
</kml>
```

Разделяемые и внутренние стили

До сих пор мы создавали в основном *внутренние стили*, определяемые внутри того элемента Feature, в котором используются. В этом разделе мы опишем способ определения стилей, которые могут *разделяться* несколькими элементами Feature, находящимися в том же или в другом файле, на этой же или на удаленной машине. Это эффективная и рекомендуемая техника.

Документы

Элемент <Document> является контейнером для деталей и разделяемых стилей. В него может быть вложено произвольное количество элементов <Style>.

(Но не помещайте разделяемые стили внутрь элемента <Folder>.) Чтобы создать стиль, разделяемый несколькими деталями, выполните следующие действия:

1. Определите все стили в элементе <Document>. Присвойте каждому стилю уникальный идентификатор id.
2. Из детали сошлитесь на id стиля, воспользовавшись элементом <styleUrl>.

Идентификатор id должен быть уникальным в пределах одного KML-файла. Для удобства задавайте осмысленные идентификаторы.

В примере *SharedStyles.kml* определен элемент <Style> с идентификатором «blue_arrow», в котором описаны синяя стрелка, красная надпись, оранжевые линии толщиной 3 пикселя и информационные окна с голубым фоном. Этот элемент вложен в элемент <Document>, и на него имеются ссылки из двух меток.

SharedStyles.kml

```
<?xml version="1.0" encoding="utf-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Document>
    <name>Пример SharedStyle</name>
    <open>1</open>
    <Style id="blue_arrow">
      <IconStyle>
        <color>ffff0000</color>
        <scale>1.4</scale>
        <Icon>
          <href>http://maps.google.com/mapfiles/kml/shapes/arrow.png</href>
        </Icon>
        <hotSpot x="32" xunits="pixels" y="1" yunits="pixels"/>
      </IconStyle>
      <LabelStyle>
        <color>ff0000ff</color>
      </LabelStyle>
      <LineStyle>
        <color>ff0080ff</color>
        <width>3</width>
      </LineStyle>
      <BalloonStyle>
        <bgColor>fffff8ce</bgColor>
      </BalloonStyle>
    </Style>
    <Placemark>
      <name>Метка 1</name>
      <styleUrl>#blue_arrow</styleUrl>
      <Point>
        <coordinates>-119.232195463845,36.0160219116952,0</coordinates>
      </Point>
    </Placemark>
    <Placemark>
      <name>Placemark 2</name>
      <styleUrl>#blue_arrow</styleUrl>
      <MultiGeometry>
      .
      .
      .
        <Point>
          <coordinates>-119.290316474804,35.9670763464666,0</coordinates>
        </Point>
      </MultiGeometry>
    </Placemark>
  </Document>
</kml>
```

Разделяемые стили не только способствуют повышению производительности, но и делают KML-файл короче и понятнее, поскольку определяются только один раз, вне зависимости от того, сколько на них имеется ссылок.

Как задается URL стиля

Если элемент <Style> определен в том же самом файле, то для ссылки на него достаточно указать идентификатор стиля с предшествующим знаком #, как показано ниже:

```
<styleUrl>#blue_arrow</styleUrl>
```

Если же стиль определен во внешнем файле, то укажите полный URL, поместив в конце идентификатор стиля с предшествующим знаком #. Например, URL

```
<styleUrl>http://server.com/fancyStylesFile.kml#clubs</styleUrl>
```

или

```
<styleUrl>http://server.com/fancyStylesFile.kmz#clubs</styleUrl>
```

ссылаются на стиль с идентификатором «clubs», который определен в файле с именем *fancyStylesFile.kml* (или в архиве *fancyStylesFile.kmz*), расположенном на сервере *server.com*. Отметим, что таким образом вы можете хранить на своем сервере стили, на которые другие разработчики ссылаются из файлов, находящихся на другом сайте. Однако, в отличие от CSS, вы не можете применить к детали в данном KML-файле стиль, находящийся в другом, внешнем файле. (Исходный файл должен содержать ссылку на внешний.)

Вот еще один пример:

```
<styleUrl>sportsclubs.kml#favorites</styleUrl>
```

Здесь мы ссылаемся на стиль с идентификатором «favorites», который определен в локальном файле, находящемся в том же каталоге, что и файл *sportsclubs.kml*.

Тонкость

Если стиль нужно применить к документу, то сам элемент <Document> должен содержать ссылку <styleUrl>, даже если <Style> является непосредственным потомком <Document>.

```
<Document>
  <Style id="myPrettyDocument">
    <ListStyle>...</ListStyle>
  </Style>
  <styleUrl>#myPrettyDocument</styleUrl>
  ...
</Document>
```

Переопределение заданных в стиле значений

Что происходит, когда какой-то подстиль определен одновременно в разделяемом и внутреннем стилях? Кому отдается предпочтение? Как вы, наверное, догадались, внутренний стиль приоритетнее разделяемого. В следующем приме-

ре в разделяемом стиле («blue_arrow») определен подстиль <IconStyle> с синей стрелкой и красной надписью. Элемент <Placemark> ссылается на этот стиль из вложенного элемента <styleUrl>, но одновременно включает собственный элемент <IconStyle>/<color>, в котором определен зеленый цвет. И этот элемент переопределяет значение <IconStyle>/<color>, заданное в разделяемом стиле (рис. 4.14).

OverridingStyles.kml

```
<?xml version="1.0" encoding="utf-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Document>
    <name>Переопределение параметров стиля</name>
    <Style id="blue_arrow">                        <!-- синий ->
      <IconStyle>
        <color>ffff0000</color>
        <scale>1.4</scale>
        <Icon>
          <href>http://maps.google.com/mapfiles/kml/shapes/arrow.png</href>
        </Icon>
        <hotSpot x="32" xunits="pixels" y="1" yunits="pixels"/>
      </IconStyle>
      <LabelStyle>
        <color>ff0000ff</color>                    <!-- красный ->
      </LabelStyle>
    </Style>
    <Placemark>
      <name>Метка 1</name>
      <styleUrl>#blue_arrow</styleUrl>
      <Style>
        <IconStyle>
          <color>ff00ff00</color>                  <!-- зеленый ->
        </IconStyle>
      </Style>
      <Point>
        <coordinates>-119.232195463845,36.0160219116952,0</coordinates>
      </Point>
    </Placemark>
  </Document>
</kml>
```

Рис. 4.14. Зеленый цвет, заданный во внутреннем элементе <IconStyle>, переопределяет синий цвет, заданный в разделяемом стиле. Поскольку значок стрелки и красный цвет метки ни в каком внутреннем стиле не переопределяются, то действуют значения, заданные в разделяемом стиле

Определение эффекта наката с помощью карт стилей

Вы обращали внимание на то, как изменяется размер значка или появляется текст надписи, когда вы наводите на элемент мышь в трехмерной панораме Google Earth? Это результат работы карты стилей. В элементе <StyleMap> определены два стиля: `normal` и `highlight`. Стиль `highlight` активируется, когда пользователь наводит мышь на метку, содержащую карту стилей. В остальное время действует стиль `normal`. Для создания карты стилей выполните следующие действия:

1. Создайте элемент <Style> для варианта **`normal`**. Не забудьте присвоить ему идентификатор `id` (например, *normalState*).
2. Создайте элемент <Style> для варианта **`highlight`**. Ему тоже присвойте идентификатор (например, *highlightState*).
3. Создайте элемент <StyleMap>. Из одного элемента <StyleMap>/<key> сошлитесь на стиль *normalState*, а из другого – на стиль *highlightState*.

Вот как должен выглядеть элемент <StyleMap>:

```
<StyleMap id="styleMapExample">
  <Pair>
    <key>normal</key>
    <styleUrl>#normalState</styleUrl>
  </Pair>
  <Pair>
    <key>highlight</key>
    <styleUrl>#highlightState</styleUrl>
  </Pair>
</StyleMap>
```

Поскольку для этого элемента <StyleMap> задан атрибут `id`, то на него можно сослаться из разных элементов <styleUrl> как на *разделяемый стиль*. По существу, использование <StyleMap> вводит в KML-файл два уровня ссылок. Просматривая KML-файл, в котором встречаются элементы <StyleMap>, вы должны найти, где создается <StyleMap> с конкретным `id`. Затем надо вернуться к началу файла и найти определения стилей, на которые ссылаются разделы `normal` и `highlight` (рис. 4.15).

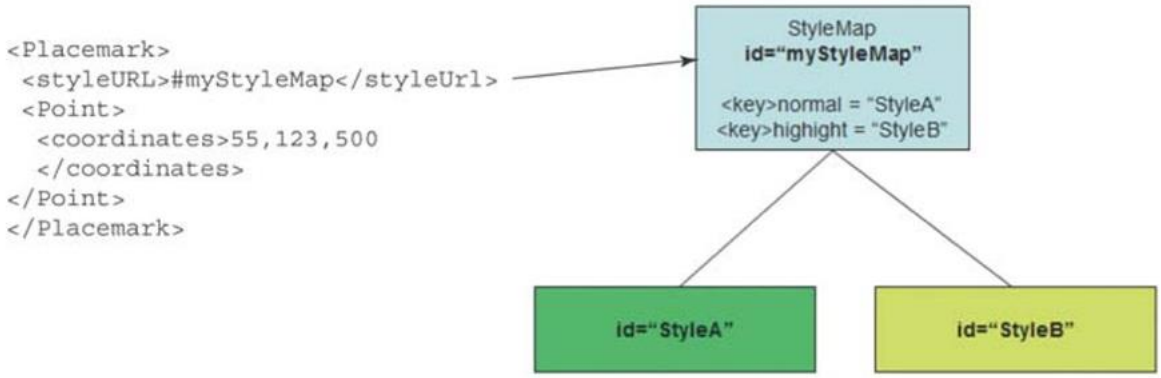


Рис. 4.15. Элемент <StyleMap> содержит две пары, одна из которых определяет стиль в «нормальном» состоянии, а другая – стиль в «выделенном» состоянии. Эффект наката реализуется заменой значка, заданного в одном стиле, ассоциированном с меткой, значком, заданным в альтернативном стиле

Стили для состояний normal и highlight могут включать некоторые (и даже все шесть) подстили, рассмотренные в первой части этой главы. Все стили составляются одинаково, вне зависимости от того, каким образом они используются: в элементе <StyleMap>, как внутренние стили в какой-то детали или как разделяемые стили, на которые ссылаются несколько деталей.

Синтаксис элемента <StyleMap>

Ниже приведен синтаксис элемента <StyleMap>:

```
<StyleMap id="ID">
  <Pair  id="ID">
    <key>normal</key>
        <!-- kml:styleStateEnum: normal или highlight ->
    <styleUrl>...</styleUrl> или <Style>...</Style>
  </Pair>
</StyleMap>
```

Примечание

Согласно спецификации KML, элемент <StyleMap> может быть вложен в любой элемент, производный от Feature. Но в текущей версии Google Earth (Release 4.3) элемент <StyleMap> срабатывает только при наведении мыши на значок, ассоциированный с точечной меткой (или на значок фотоналожения, о чем см. главу 5 «Наложения»). В настоящем обсуждении нас интересует прежде всего реализованное в данный момент поведение и порядок использования этого элемента.

Сила точки

Элемент <StyleMap> действует только при использовании в сочетании с меткой, содержащей внутри себя элемент <Point>. Метка может содержать и другие геометрические элементы, в этом случае все они, включая и точку, должны быть вложены в элемент <MultiGeometry> (см. раздел «Более сложный пример: другие способы применения карты стилей» ниже).

На элементы <styleUrl> внутри <StyleMap> можно ссылаться, как на любые другие элементы <styleUrl> (см. раздел «Как задается URL стиля» выше).

Простой пример

Ниже приведен простой пример использования карты стилей («my_arrow_style») для реализации того поведения значка стрелки и надписи, которое было описано выше. Когда пользователь наводит мышь на стрелку, та становится синей, а надпись – красной. В этом примере обе метки разделяют одну карту стилей. На рис. 4.16 показано, как изменяется внешний вид метки, когда мышь проходит над значком с надписью «Arrow Icon».

StyleMap.kml

```
<?xml version="1.0" encoding="utf-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Document>
```

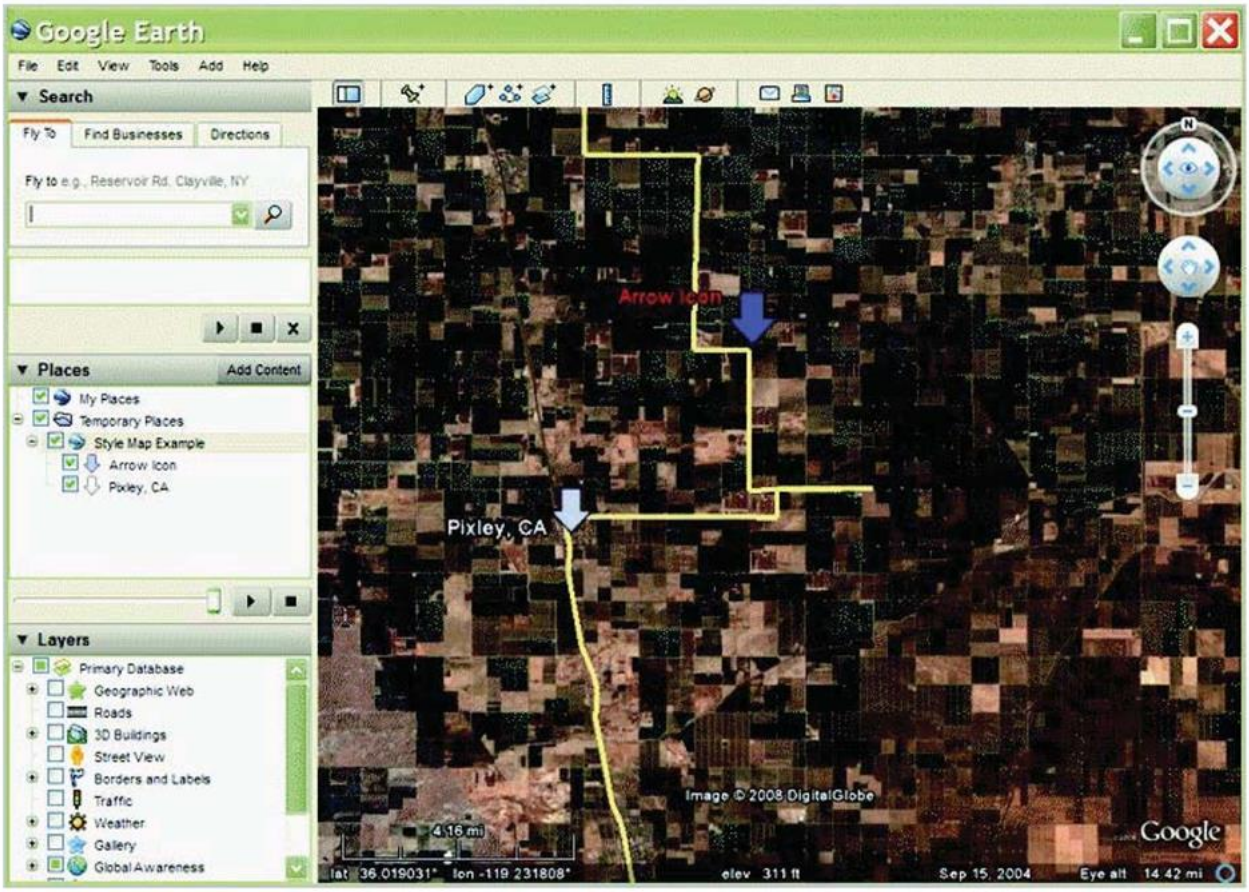


Рис. 4.16. Применение карты стилей для создания значков, изменяющих цвет при наведении мыши

```
<name>Пример карты стилей</name>
<open>1</open>
<Style id="highlight_arrow">
  <IconStyle>
    <color>ffff0000</color>
    <scale>1.4</scale>
    <Icon>
      <href>http://maps.google.com/mapfiles/kml/shapes/arrow.png</href>
    </Icon>
    <hotSpot x="32" xunits="pixels" y="1" yunits="pixels"/>
  </IconStyle>
  <LabelStyle>
    <color>ff0000ff</color>
  </LabelStyle>
  <LineStyle>
    <color>ff95eaff</color>
    <width>4</width>
  </LineStyle>
</Style>
<Style id="normal_arrow">
  <IconStyle>
    <color>ffffcdc0</color>
    <scale>1.2</scale>
    <Icon>
      <href>http://maps.google.com/mapfiles/kml/shapes/arrow.png</href>
    </Icon>
    <hotSpot x="32" xunits="pixels" y="1" yunits="pixels"/>
  </IconStyle>
```

```
      <LineStyle>
        <color>ff95eaff</color>
        <width>4</width>
      </LineStyle>
    </Style>
    <StyleMap id="my_arrow_style">
      <Pair>
        <key>normal</key>
        <styleUrl>#normal_arrow</styleUrl>
      </Pair>
      <Pair>
        <key>highlight</key>
        <styleUrl>#highlight_arrow</styleUrl>
      </Pair>
    </StyleMap>
    <Placemark>
      <name>Arrow Icon</name>
      <styleUrl>#my_arrow_style</styleUrl>
      <Point>
        <coordinates>-119.232195463845,36.0160219116952,0</coordinates>
      </Point>
    </Placemark>
    <Placemark>
      <name>Pixley, CA</name>
      <styleUrl>#my_arrow_style</styleUrl>
      <MultiGeometry>
        <LineString>
          <tessellate>1</tessellate>
          <coordinates>
            -119.039568538709,35.3512349410626,0
            -119.039393055414,35.3620902295471,0
            -119.039826751439,35.3631086344801,0
             .
             .
             .
          </coordinates>
        </LineString>
        <Point>
          <coordinates>-119.290316474804,35.9670763464666,0</coordinates>
        </Point>
      </MultiGeometry>
    </Placemark>
  </Document>
</kml>
```

Более сложный пример: другие способы применения карты стилей

Хотя реализовать эффект наката непосредственно для многоугольника не получится, можно имитировать его, создав метку, содержащую элементы <Point> и <Polygon> внутри элемента <MultiGeometry>, как показано в следующем примере. На рис. 4.17 изображены выделенный многоугольник и значок для одной из меток. Для выделенного многоугольника режим <colorMode> установлен в `random`, поэтому при каждой загрузке файла он будет выглядеть по-разному. Для многоугольника в «нормальном» состоянии цвет <color> прозрачный, поэтому он не виден, пока пользователь не наведет мышь на значок. Еще раз отметим, что эффект наката активируется только точечной меткой, никакие другие геометрические элементы внутри <Placemark> на это не способны.

Рис. 4.17. Наведение мыши на значок активирует стиль «highlight», заданный в карте стилей. Для проявления этого эффекта необходим элемент <Point>. В данном примере эффект наката для многоугольника имитируется за счет того, что элемент <Polygon> вложен в элемент <MultiGeometry> вместе с <Point>. На левом рисунке показан «нормальный» стиль, а на правом – «выделенный»

Применение карты стилей для смены стиля геометрического элемента

```
<?xml version="1.0" encoding="utf-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Document>
    <Style id="normal_state">
      <IconStyle>
        <Icon>
          <href>http://maps.google.com/mapfiles/kml/shapes/flag.png</href>
        </Icon>
        <hotSpot x="0.5" xunits="fraction" y="0" yunits="fraction"/>
      </IconStyle>
      <LineStyle>
        <color>00000000</color>
      </LineStyle>
      <PolyStyle>
        <color>00000000</color>
      </PolyStyle>
    </Style>
    <Style id="selected_state">
      <IconStyle>
        <Icon>
          <href>http://maps.google.com/mapfiles/kml/shapes/flag.png</href>
        </Icon>
        <hotSpot x="0.5" xunits="fraction" y="0" yunits="fraction"/>
      </IconStyle>
      <LineStyle>
        <color>ff00ffff</color>
        <width>4</width>
      </LineStyle>
      <PolyStyle>
        <colorMode>random</colorMode>
      </PolyStyle>
    </Style>
    <StyleMap id="highlight_the_state">
      <Pair>
        <key>normal</key>
```

```
      <styleUrl>#normal_state</styleUrl>
    </Pair>
    <Pair>
      <key>highlight</key>
      <styleUrl>#selected_state</styleUrl>
    </Pair>
  </StyleMap>
  <Placemark>
    <name>Oklahoma City</name>
    <styleUrl>#highlight_the_state</styleUrl>
    <MultiGeometry>
      <Polygon>
        <outerBoundaryIs>
          <LinearRing>
            <coordinates>
              -94.4393221493475,34.9291508772006,0
              -94.4285520120899,35.4005462695086,0
              -94.4684852110367,35.6410882624306,0
              .
              .
              .
              -94.4393221493475,34.9291508772006,0
            </coordinates>
          </LinearRing>
        </outerBoundaryIs>
      </Polygon>
      <!-- попробуйте закомментировать следующую точку и посмотрите, что получится
           при наведении мыши ->
      <Point>
        <coordinates>-97.52033,35.47204,0</coordinates>
      </Point>
    </MultiGeometry>
  </Placemark>
</Document>
</kml>
```

Применение карты стилей для упрощения картинки

Элемент <StyleMap> можно использовать для упрощения картинки, перегруженной метками. Для этого следует для стиля `normal` задать коэффициент масштабирования <scale> в элементе <LabelStyle> равным 0, а для стиля `highlight` – 1. В результате надпись метки будет появляться только при наведении мыши на значок этой метки.

Карту стилей можно использовать также для предоставления различной информации в зависимости от того, находится мышь над значком метки или нет. Например, в нормальном состоянии можно показывать простой значок, а в выделенном – целый набор значков, представляющих данные, относящиеся к метке.

Определение стилей во внешнем файле

Хотя язык KML не поддерживает многие аспекты CSS, он все же позволяет определить все стили во внешнем KML-файле, а затем явно ссылаться на них из других KML-файлов. KML-файл со стилями можно разместить не на том же компьютере, где находится KML-файл с элементами, ссылающимися на стили

с помощью <styleUrl>. Такое отделение содержимого от стилей часто бывает полезно, когда нужно обобществить стили и обновлять их централизованно. Отдельный элемент Feature содержит явную ссылку на свой (определенный вовне) элемент <Style>.

Что дальше?

В главе 5 мы узнаем, как создавать экранные, рельефные и фотографические наложения и манипулировать ими в KML. Наложение изображений на основу, предоставляемую Google Earth, позволяет добавлять собственные графические слои, представляя данные в ярком географическом контексте.

Глава 5. Наложения

Прочитав эту главу, вы сможете:

- ✓ искать в Интернете интересные аэрофотоснимки Земли, исторические или топографические карты, а также добавлять в KML-файл изображение, используемое в качестве рельефного наложения;
- ✓ создать экранное наложение из корпоративного логотипа и отобразить его в правом верхнем углу окна Google Earth;
- ✓ объяснить, как элемент <LatLonBox> используется в элементе <GroundOverlay>;
- ✓ добавить свою фотографию в Google Earth в составе элемента <PhotoOverlay>;
- ✓ создать элемент <Camera> и с его помощью представить вид на ваш дом или школу с интересной точки обзора.

Предварительный обзор

В этой главе рассказывается о создании трех видов наложений:

- ❑ рельефных;
- ❑ экранных;
- ❑ фотографических.

Мы начнем с описания каждого вида наложения и типичных примеров использования. Затем рассмотрим базовые концепции, применимые ко всем наложениям, а именно:

- ❑ цвет (детально обсуждается в главе 2);
- ❑ прозрачность;
- ❑ порядок рисования;
- ❑ значки.

В главе 2 вы научились задавать точку обзора с помощью элемента <LookAt>. А здесь мы рассмотрим еще один элемент, производный от AbstractView, – <Camera>. Точку обзора для любого наложения (и вообще любой детали) можно определить как с помощью <LookAt>, так и с помощью <Camera>.

Обсудив концепции, общие для всех наложений, мы далее рассмотрим каждое в отдельности. Будут описаны синтаксис и семантика, относящиеся к созданию и модификации рельефных, экранных и фотографических наложений, а также приведены примеры и диаграммы.

Что такое наложение?

Наложением называется изображение, накладываемое на базовую спутниковую карту, которую показывает Google Earth. В KML существуют три вида наложений.

- Рельефное наложение «натягивается» на земную поверхность или помещается на определенной высоте над ней. В обоих случаях рельефное наложение повторяет особенности рельефа. Наложение может быть снятой со спутника фотографией с высоким разрешением, изображением, отражающим метеорологические условия над Землей, двумерной картой, сохраненной в виде изображения, или многоугольником либо какой-то другой аннотацией, тоже представленной в виде изображения. Большинство геобраузеров поддерживают различные графические форматы рельефных наложений, в том числе BMP, DDS, GIF, JPG, PGM, 32-разрядный PNG, PPM, TGA и TIFF. На рис. 5.1 показано рельефное наложение, представ-

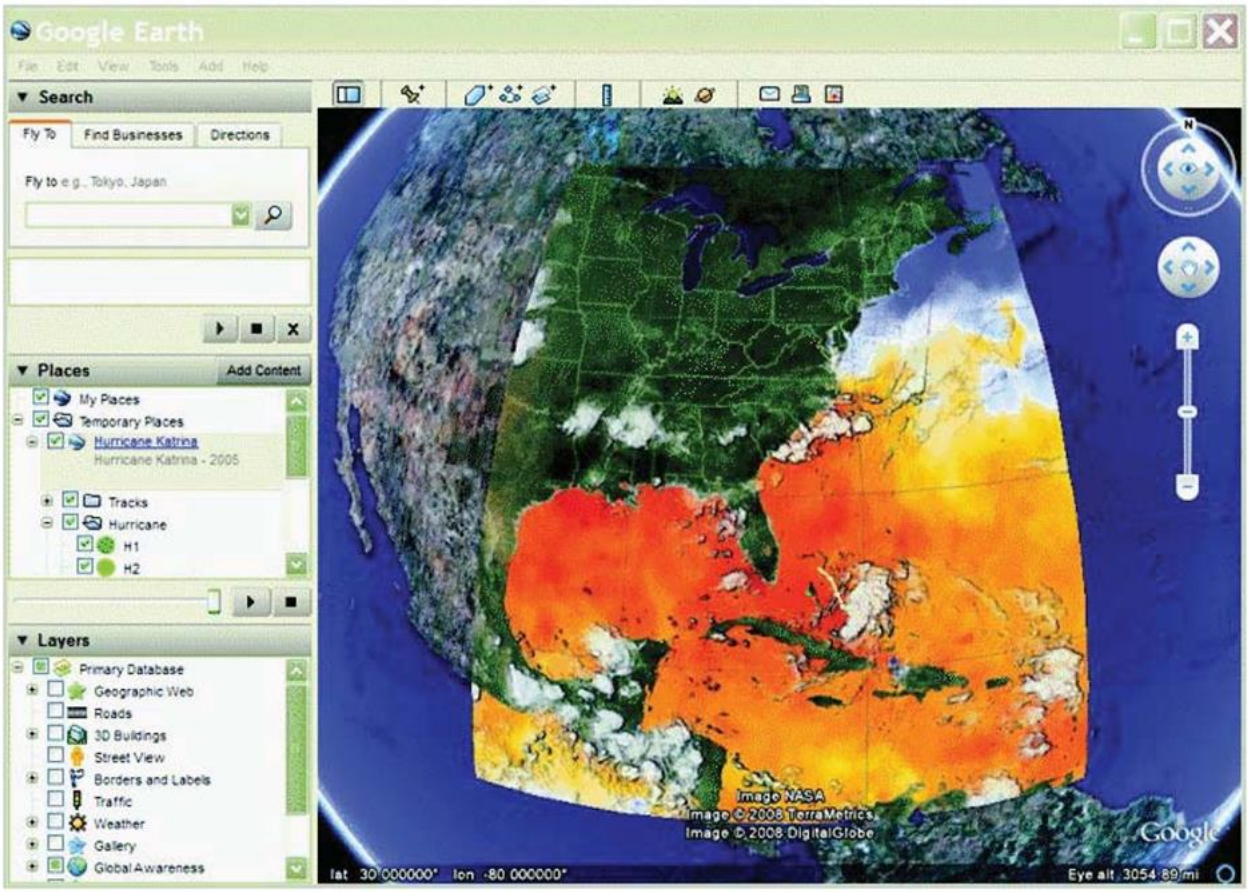


Рис. 5.1. Наложение со спутниковой фотографией урагана Катрина в сочетании с цветокодированным распределением температур на поверхности моря (KML-файл создан Брайаном Фладом, www.spatialdatalogic.com/CS/blogs/brian_flood/archive/2006 /09/22/Google-Earth-Time-Sample-_2Doo_-Animation-Part-2.aspx. Публикуется с разрешения автора)

ляющее собой спутниковую фотографию урагана Катрина в сочетании с цветокодированным распределением температур на поверхности моря (часть KML-анимации, подготовленной Брайаном Фладом).

- *Экранные наложения* – это двумерные изображения, как бы «приклеенные» к экрану. Таким способом часто добавляются логотипы, титры и легенды на картах. На рис. 5.2 показано экранное наложение с указанием источника изображения.
- *Фотоналожения* – это фотографии, добавленные в Google Earth в определенных точках. Они могут быть прямоугольными (выглядят как рекламные щиты на земной поверхности), цилиндрическими или сферическими. Пользователь может «влететь» в фотоналожение, детально исследовать его, а потом вернуться к основной карте геобраузера. Поддерживается добавление очень больших панорамных изображений, предварительно разбитых на несколько слоев для эффективного просмотра. На рис. 5.3 показана фотоэкскурсия по месту археологических раскопок, осуществляемых группой ученых из Британского музея в Ком-Фирине, Египет (www.britishmuseum.org/research/research_projects/ kom_firin/introduction.aspx). На этой KML-презентации, созданной Стефаном Джинсом, автором блога Ogle Earth, основаны примеры в настоящей главе.

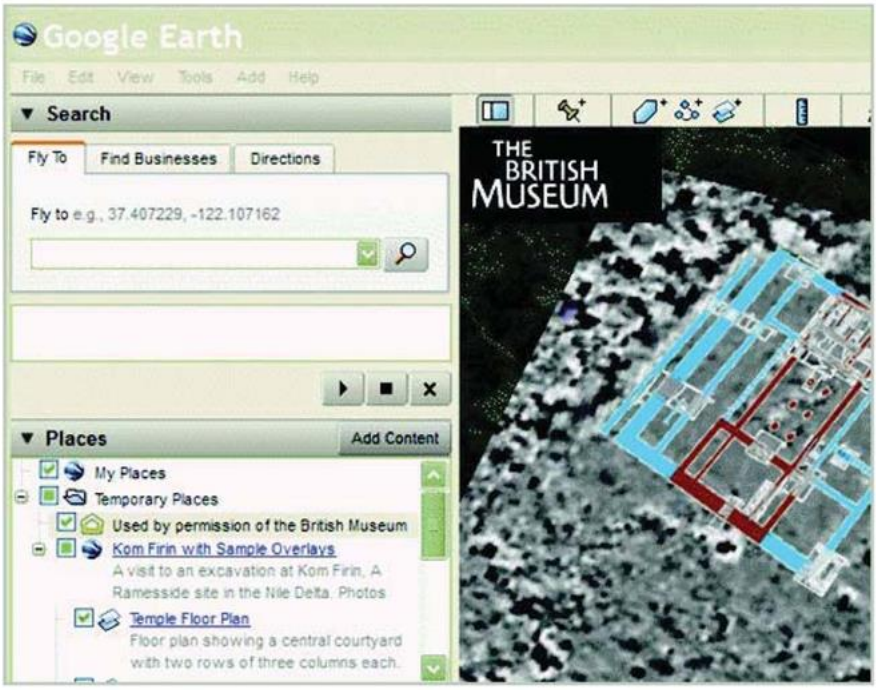


Рис. 5.2. Экранное наложение, служащее для указания источника графического материала (в данном случае Британский музей). В этом примере для отображения нестандартного значка в списке использован элемент <ListStyle> (см. описание элемента <ScreenOverlay> ниже)

Инструменты для создания наложений

В этой главе мы расскажем о представлении всех трех наложений в языке KML. Часто для создания наложения вы будете применять тот или иной программный инструмент, не трогая сам KML-файл. Программа Google Earth предлагает средства для добавления рельефных и фотографических наложений. Если же вы захотите добавить экранное наложение или реализовать программный интерфейс между имеющимся у вас графическим материалом и механизмом наложений в KML, то информация, изложенная в настоящей главе, очень пригодится.

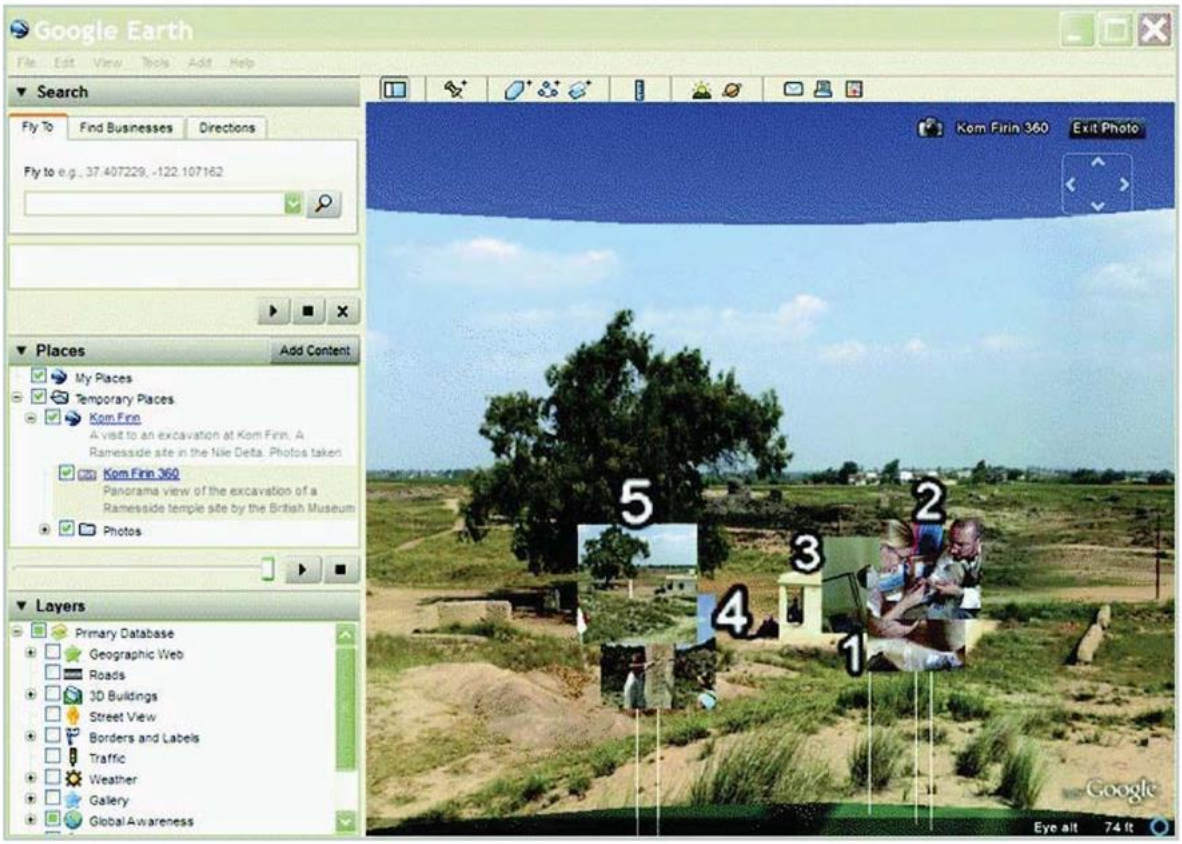


Рис. 5.3. Изобретательное применение нескольких разных техник, позволившее визуализировать археологические раскопки храмового комплекса Рамсеса II в местности Ком-Фирин в Западной Дельте Нила. Фоновое изображение представляет собой цилиндрическое фотоналожение, в которое пользователь Google Earth может «влететь» и детально его исследовать. Миниатюрные картинки – это значки меток, с которыми ассоциированы информационные окна и ссылки на сайт с фотографиями (публикуется с разрешения Стефана Джинса, www.ogleearth.com/2007/11/the_kom_firin_d.html)

Основные концепции

Наложения, как и метки, являются элементами, производными от Feature (рис. 5.4). Элемент Feature предоставляет ряд потомков, наследуемых всеми производными от него элементами.

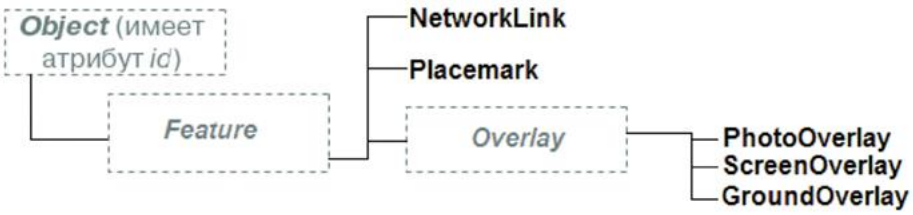


Рис. 5.4. Элемент Overlay и производные от него

Это означает, в частности, что у всех элементов, производных от Overlay (GroundOverlay, PhotoOverlay и ScreenOverlay), есть имя, описание, краткое описание и информационное окно.

Синтаксис элемента <Overlay>

Ниже приведен полный синтаксис элемента <Overlay>, в том числе все потомки элемента Feature, а также три потомка, определенных в самом элементе Overlay.

```
<!-- Overlay ->
  <унаследованы от элемента Feature>
  <name>...</name>
  <visibility>1</visibility>
  <open>0</open>
  <atom:author>...</atom:author>
  <atom:link>...</atom:link>
  <address>...</address>
  <xal:AddressDetails>...</xal:AddressDetails>
  <phoneNumber>...</phoneNumber>
  <Snippet maxLines="2">...</Snippet>
  <description>...</description>
  <AbstractView>...</AbstractView>
  <TimePrimitive>...</TimePrimitive>
  <styleUrl>...</styleUrl>
  <StyleSelector>...</StyleSelector>
  <Region>...</Region>
  <ExtendedData>...</ExtendedData>
  <!-- специфичны для Overlay ->
  <color>ffffffff</color>
  <drawOrder>0</drawOrder>
  <Icon>
    <href>...</href>
    <refreshMode>onChange</refreshMode>
    <refreshInterval>4.0</refreshInterval>
    <viewRefreshMode>never</viewRefreshMode>
    <viewRefreshTime>4</viewRefreshTime>
    <viewBoundScale>1</viewBoundScale>
    <viewFormat>...</viewFormat>
    <httpQuery>...</httpQuery>
  </Icon>
<!-- /Overlay ->
```

Следующие элементы-потомки являются общими для всех наложений:

- цвет (и прозрачность);
- порядок рисования;
- значок (в данном случае накладываемое *изображение*).

Значки: задание и обновление накладываемого изображения

Потомок <href> элемента <Icon> задает файл, содержащий накладываемое изображение. Он может находиться как в локальной файловой системе, так и на веб-сервере. Например, следующая разметка означает, что файл с изображением находится в одном каталоге с KML-файлом.

```
<href>clouds.jpg</href>
```

А эта разметка говорит, что файл находится на удаленном сервере WeatherStation.com:

```
<href>http://WeatherStation.com/clouds.jpg</href>
```

Если элемент <href> опущен или пуст, то рисуется прямоугольник того цвета и размера, который определен в наложении.

Элемент <Icon> содержит также элементы для задания условий, при которых наложение должно перегружаться (обновляться), а именно:

- при изменении любого параметра <Icon>;
- с заданной периодичностью;
- по истечении заданного срока;
- при изменении точки обзора пользователем.

Элемент <Icon>, вложенный в любой элемент наложения, может иметь тех же потомков, что элемент <Link>, вложенный в <NetworkLink>. Дополнительную информацию об обновлении по времени или при изменении точки обзора см. в главе 6 «Сетевые ссылки».

Другой способ задания точки обзора: камера

В главе 2 вы научились использовать элемент <LookAt> для описания точки обзора элемента Feature или NetworkLinkControl. Сейчас мы познакомимся еще с одним элементом, производным от AbstractView: <Camera>. Элемент Overlay, как и любой производный от Feature, может содержать в качестве потомка элемент AbstractView, задающий для него точку обзора. У AbstractView есть два производных элемента: <LookAt> и <Camera>.

Различия между элементами <LookAt> и <Camera>

Элементы <LookAt> и <Camera> применяются для задания точки обзора элемента Feature или NetworkLinkControl. Различаются они в двух отношениях:

- <LookAt> определяет точку обзора относительно рассматриваемой сцены, тогда как <Camera> – в терминах положения наблюдателя, смотрящего на сцену через виртуальную камеру;

- <Camera> предоставляет все шесть степеней свободы, то есть ее можно расположить в любой точке пространства и повернуть вокруг осей x, y и z (описывается ниже). Элемент <Camera> позволяет расположить точку обзора так, чтобы камера смотрела в небо, выше линии горизонта.

Синтаксис элемента <Camera>

Элемент <Camera> описывает положение и направление взгляда виртуальной камеры, которая рассматривает элемент Feature или содержимое элемента <NetworkLinkControl>. Положение камеры определяется элементами <longitude>, <latitude>, <altitude> и <altitudeMode>. Направление обзора определяется элементами <heading>, <tilt> и <roll>.

```
<Camera id="ID">
  <longitude>0.0</longitude>
  <latitude>0.0</latitude>
  <altitude>0.0</altitude>
  <heading>0.0</heading>
  <tilt>0.0</tilt>
  <roll>0.0</roll>
  <altitudeMode>clampToGround</altitudeMode>
</Camera>
```

Потомки элемента <Camera> определяются следующим образом:

<longitude>

Долгота виртуальной камеры (точки расположения глаза). Угловые расстояния измеряются в градусах от нулевого меридиана. Значения к западу от нулевого меридиана изменяются в диапазоне от –180° до 0°, к востоку – от 0° до 180°.

<latitude>

Широта виртуальной камеры. Измеряется в градусах к северу и югу от экватора и изменяется в диапазоне от –90° до 90°.

<altitude>

Расстояние до земной поверхности в метрах. Об интерпретации значения см. описание элемента <altitudeMode>.

<heading>

Направление (север, юг, восток, запад) в градусах. По умолчанию подразумевается 0 (север). Изменяется в диапазоне от 0 до 360°, при этом 90° – это восток, 180° – юг, 270° – запад. Допустимы также отрицательные значения (отсчет против часовой стрелки от 0).

<tilt>

Угол поворота камеры относительно оси x (наклон). Значение 0 означает, что взгляд направлен строго вниз к Земле (наиболее часто встречающийся случай). Значение 90° означает, что взгляд направлен к линии горизонта. Значения, большие 90°, соответствуют взгляду вверх, в небо. Значение <tilt> изменяется в диапазоне от 0° до +180°.

<roll>

Угол поворота камеры относительно оси z (крен). Значение изменяется в диапазоне от –180° до +180°.

<altitudeMode>

Определяет способ интерпретации элемента <altitude>. Допустимы следующие значения:

`clampToGround` (по умолчанию)

Означает, что значение, заданное в элементе <altitude>, следует игнорировать и поместить камеру на поверхность Земли.

`relativeToGround`

Значение <altitude> измеряется в метрах относительно поверхности Земли.

`absolute`

Значение <altitude> измеряется в метрах над уровнем моря.

Начальное положение камеры

На рис. 5.5 показана начальная ориентация камеры относительно осей *x*, *y* и *z*:

- ось *x* направлена вправо от камеры, в трехмерной машинной графике она называется *правосторонним вектором*;
- ось *y* определяет направление «вверх» относительно экрана и называется *вектором верха*;
- ось *z* направлена от центра экрана в точку расположения глаза. Камера смотрит вниз вдоль оси -*z*, которая называется *вектором взгляда*.

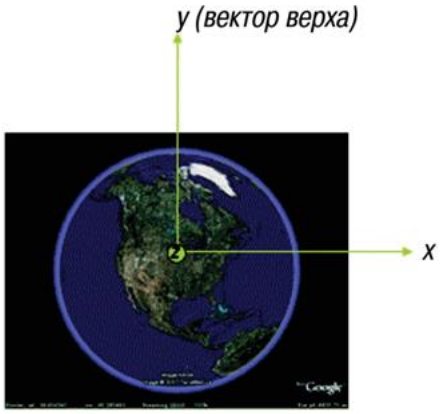


Рис. 5.5. Ориентация виртуальной камеры по умолчанию

Порядок выполнения поворотов важен. По умолчанию камера смотрит вдоль оси *z* в направлении Земли. Порядок выполнения поворотов определяется следующим образом:

1. <heading> повернуть относительно оси *z* (вектора взгляда);
2. <tilt> повернуть относительно оси *x*;
3. <roll> повернуть относительно (новой) оси *z* (нового вектора взгляда).

Отметим, что при каждом повороте камеры две ее оси меняют ориентацию. Рассмотрим несколько примеров поворота.

На рис. 5.6 (слева) камера смотрит вертикально вниз на египетские пирамиды. На рис. 5.6 (справа) показано, что получается, когда <heading> равно 90 градусов, то есть вектор верха камеры направлен на восток.

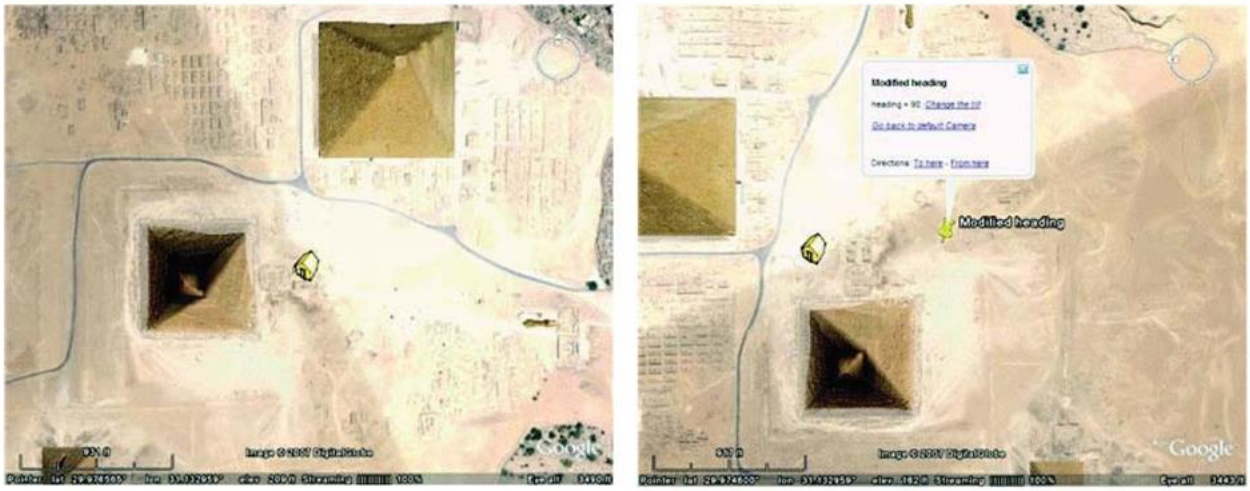


Рис. 5.6. На левом рисунке показано положение камеры по умолчанию, когда она смотрит вертикально вниз. На правом рисунке значение <heading> равно 90 градусов

На рис. 5.7 (слева) значение <tilt> равно 75 градусов, то есть камера повернута к горизонту. На рис. 5.7 (справа) значение <roll> равно 30 градусов, при этом картинка наклонена (пожалуй, это самый нежелательный эффект).

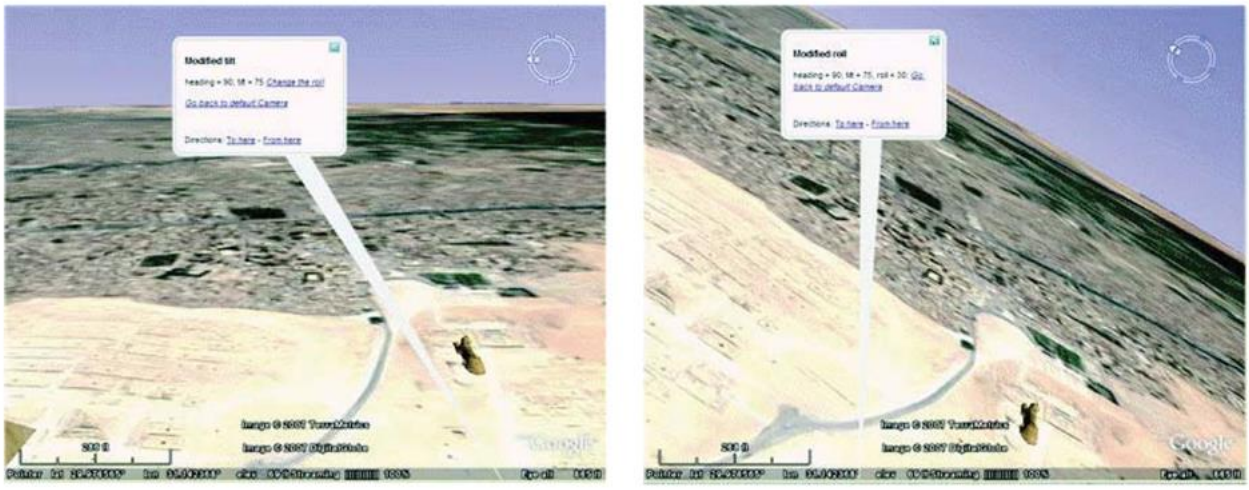


Рис. 5.7. На левом рисунке значение <tilt> равно 75 градусов. На правом рисунке значение <roll> равно 30 градусов

В файле *CameraRotation.kml* иллюстрируются эффекты различных поворотов камеры. Результаты изображены на рис. 5.6 и 5.7.

CameraRotations.kml

```
<?xml version="1.0" encoding="utf-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Document>
    <name>Повороты камеры</name>
    <Camera>              <!-- начальное положение камеры ->
      <longitude>31.132959</longitude>
      <latitude>29.976565</latitude>
      <altitude>1000</altitude>
      <heading>0</heading>
      <tilt>0</tilt>
      <roll>0</roll>
      <altitudeMode>relativeToGround</altitudeMode>
    </Camera>
    <Placemark id="defaultCamera">
      <name>Камера по умолчанию</name>
      <description><![CDATA[
          Ориентация камеры по умолчанию.
        <a href="#changeHeading;balloonFlyto">
                <em>Изменим направление</em></a>
        <p>
      ]]></description>
      <Camera>
        <longitude>31.132959</longitude>
        <latitude>29.976565</latitude>
        <altitude>1000</altitude>
        <heading>0</heading>
        <tilt>0</tilt>
        <roll>0</roll>
        <altitudeMode>relativeToGround</altitudeMode>
      </Camera>
      <Point>
        <coordinates>31.1324,29.9739,0</coordinates>
      </Point>
    </Placemark>

    <Placemark id="changeHeading">
      <name>Измененное направление</name>
      <description><![CDATA[
        heading = 90; <a href="#changeTilt;balloonFlyto">
        <em>Изменим наклон</em></a>
        <p>
        <a href="#defaultCamera;balloonFlyto"><em>Вернемся к камере по умолчанию</em></a>
        <p>
      ]]></description>
      <Camera>
        <longitude>31.132959</longitude>
        <latitude>29.9746</latitude>
        <altitude>1000</altitude>
        <heading>90</heading>
        <tilt>0</tilt>
        <roll>0</roll>
        <altitudeMode>relativeToGround</altitudeMode>
      </Camera>
      <Point>
        <coordinates>31.133,29.9746,0</coordinates>
      </Point>
    </Placemark>
    <Placemark id="changeTilt">
      <name>Измененный наклон</name>
      <description><![CDATA[
      .
      .
      .
      <Camera>
        <longitude>31.132959</longitude>
        <latitude>29.976565</latitude>
        <altitude>200</altitude>
        <heading>90</heading>
```

```
        <tilt>75</tilt>
        <roll>0</roll>
        <altitudeMode>relativeToGround</altitudeMode>
      </Camera>
      <Point>
        <coordinates>31.1337,29.9754,0</coordinates>
      </Point>
    </Placemark>
    <Placemark id="changeRoll">
      .
      .
      .
      <Camera>
        <longitude>31.132959</longitude>
        <latitude>29.976565</latitude>
        <altitude>200</altitude>
        <heading>90</heading>
        <tilt>75</tilt>
        <roll>30</roll>
        <altitudeMode>relativeToGround</altitudeMode>
      </Camera>
      <Point>
        <coordinates>31.1343,29.976,0</coordinates>
      </Point>
    </Placemark>
  </Document>
</kml>
```

Рельефное наложение

Рельефные наложения чаще всего создаются с помощью таких инструментов, как пользовательский интерфейс программы Google Earth, но можно сделать это и вручную, если вы знаете координаты границ изображения. В Google Earth рельефные наложения называются «накладываемыми изображениями».

Синтаксис элемента <GroundOverlay>

Ниже приведен синтаксис элемента <GroundOverlay> (описание потомков, наследуемых от Feature и Overlay, см. в разделе «Синтаксис элемента <Overlay>» выше и в приложении А).

```
<GroundOverlay id="ID">
  <!-- элементы, унаследованные от Feature ->
  .
  .
  .
  <!-- элементы, унаследованные от Overlay ->
  .
  .
  .
  <!-- элементы, специфичные для GroundOverlay ->
  <altitude>0.0</altitude>
  <altitudeMode>clampToGround</altitudeMode>
  <LatLonBox>
    <north>180.0</north>
    <south>-180.0</south>
    <east>180.0</east>
    <west>-180.0</west>
    <rotation>0.0</rotation>
  <LatLonBox>
</GroundOverlay>
```

Элемент <LatLonBox>

В <GroundOverlay> вложен элемент <LatLonBox>, который определяет границы изображения в терминах широты и долготы.

Синтаксис элемента <LatLonBox> определяется следующим образом (см. также рис. 5.8):

<north>

Широта северного края накладываемого изображения в градусах от 0 до ±90.

<south>

Широта южного края накладываемого изображения в градусах от 0 до ±90.

<east>

Долгота восточного края накладываемого изображения в градусах от 0 до ±180.

<west>

Долгота западного края накладываемого изображения в градусах от 0 до ±180.

<rotation>

Угол поворота накладываемого изображения вокруг центра в градусах от 0 до ±180. По умолчанию равен 0 (верхний край изображения направлен на север). Угол поворота отсчитывается против часовой стрелки.

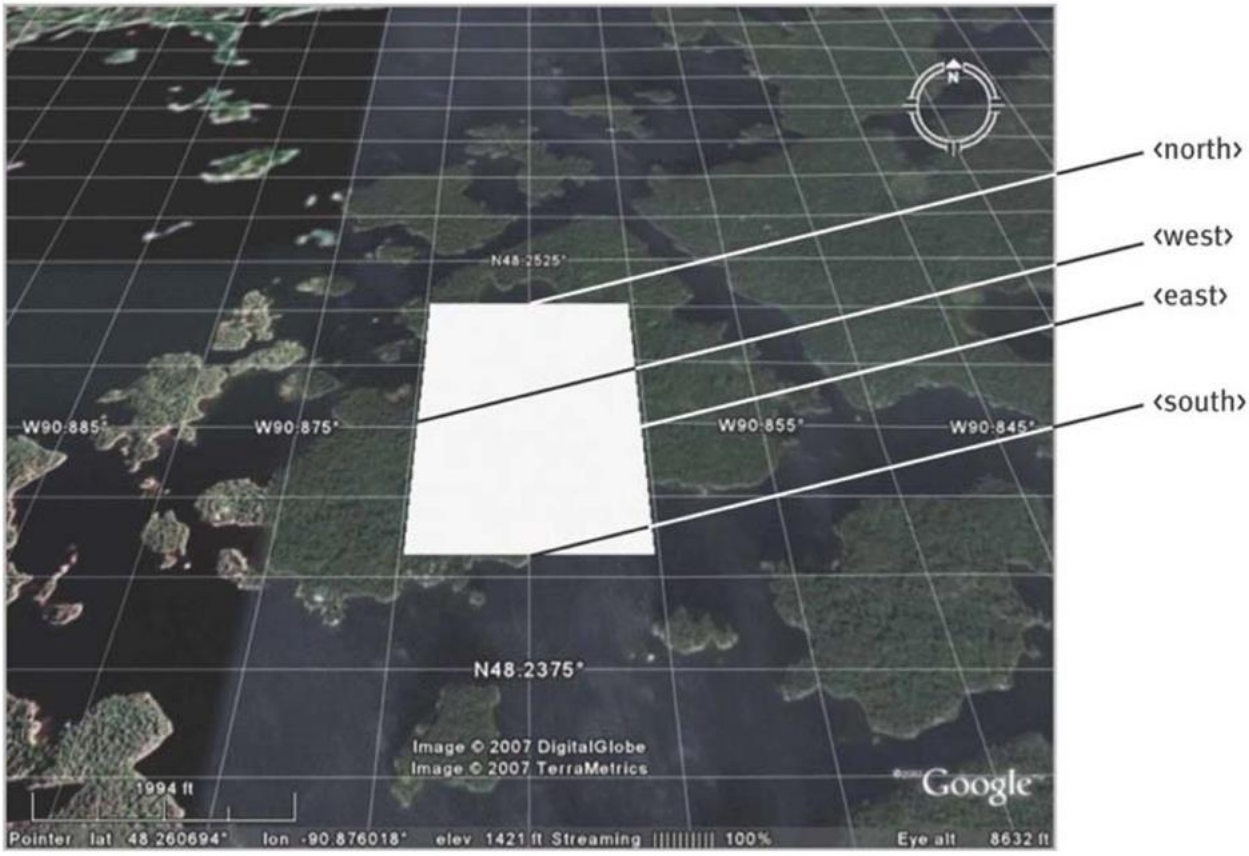


Рис. 5.8. Рельефное наложение без ассоциированного изображения

В следующем примере задается <LatLonBox> для элемента <GroundOverlay> без изображения.

GroundOverlayNoImage.kml

```
<?xml version="1.0" encoding="utf-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <GroundOverlay>
    <LatLonBox>
      <north>48.25</north>
      <south>48.24</south>
      <east>-90.86</east>
      <west>-90.87</west>
    </LatLonBox>
  </GroundOverlay>
</kml>
```

На рис. 5.8 показано, как выглядит это наложение в программе Google Earth с включенной координатной сеткой.

Примечание

В KML края наложения растягиваются до краев <LatLonBox>. Более изощренные ГИС (геоинформационные системы) позволяют точнее управлять расположением изображения вдоль рельефа. Один из способов добиться более точного «натягивания» изображения на земную поверхность заключается в том, чтобы разбить изображение на большое число маленьких прямоугольников и для каждого задать отдельное рельефное наложение с помощью <LatLonBox>.

Высота и режим высоты

Для рельефного наложения можно задать параметры <altitude> и <altitudeMode>. В главе 3 уже отмечалось, что значение <altitude> интерпретируется с учетом заданного режима высоты. В элементе <GroundOverlay> допустимы только режимы `clampToGround` (по умолчанию) и `absolute`.

Совет

Многоугольники, созданные на ненулевой высоте, не повторяют рельеф. Но если это необходимо, то вы можете сохранить многоугольник в виде изображения и поместить его внутрь элемента <GroundOverlay> на ненулевой высоте. Для задания цвета и прозрачности многоугольника используйте элемент <color>, вложенный в <Icon>.

Ниже приведен пример размещения элемента <GroundOverlay> (без изображения) на высоте 500 м над уровнем моря:

```
<?xml version="1.0" encoding="utf-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <GroundOverlay>
    <altitude>500</altitude>
    <altitudeMode>absolute</altitudeMode>
    <LatLonBox>
      <north>48.25</north>
      <south>48.24</south>
      <east>-90.86</east>
```

```
      <west>-90.87</west>
    </LatLonBox>
  </GroundOverlay>
</kml>
```

Пример элемента <GroundOverlay>

В следующем примере показаны два рельефных наложения, рисуемые одно поверх другого согласно порядку, заданному в элементе <drawOrder> (рис. 5.9). Верхний элемент был обработан приложением SnagIt, чтобы сделать белые элементы плана полностью прозрачными. Эту задачу способны решить многие программы обработки фотографий, например Photoshop или GIMP (GNU Image Manipulation Program). Не забудьте, что для сохранения прозрачности изображение надо сохранить в формате PNG или GIF. Нижнее изображение, магнитометрическая карта места раскопок, видна сквозь прозрачные области верхнего.

Рис. 5.9. Два рельефных наложения с различной информацией о раскопках в Ком-Фирине. На верхнем изображении показан план храма с прозрачными областями, позволяющими видеть расположенную ниже магнитометрическую карту (публикуется с разрешения Британского музея)

GroundAndImage.kml

```
<?xml version="1.0" encoding="utf-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Document>
    <name>Ком-Фирин с наложениями</name>
    <open>1</open>
    <description>Раскопки храма Рамсеса в Дельте Нила (местечко Ком-Фирин). Снимок
          сделан 6 октября 2007 года.
    </description>
    <GroundOverlay>
      <name>План храма</name>
      <description>На плане храма виден центральный двор с двумя рядами колонн по три
            в каждом. На месте проходов, вероятно, были ворота предположи-
            тельно шириной два метра.
      </description>
      <drawOrder>1</drawOrder>
      <Icon>
        <href>komfirin1.gif</href>
      </Icon>
      <LatLonBox>
        <north>30.865689</north>
        <south>30.86382</south>
        <east>30.491607</east>
        <west>30.48943</west>
        <rotation>-121.340933</rotation>
      </LatLonBox>
    </GroundOverlay>
    <GroundOverlay>
      <name>Магнитометрическая карта храма в Ком-Фирине</name>
      <description>Результаты магнитометрического обследования храма, окружающих
            укреплений и северных ворот. Публикуется с разрешения Британского музея.
      </description>
      <Icon>
        <href>3komfirin_3.jpg</href>
      </Icon>
      <LatLonBox>
        <north>30.866014</north>
        <south>30.862289</south>
        <east>30.492794</east>
        <west>30.488453</west>
        <rotation>-16.823374</rotation>
      </LatLonBox>
    </GroundOverlay>
  </Document>
</kml>
```

Экранное наложение

Экранное наложение фиксируется в конкретном месте экрана. Куда бы ни переходил пользователь в геобраузере, экранное наложение остается в одной и той же позиции. Как и в случае рельефных наложений, изображение задается с помощью потомка <href> элемента <Icon>. Если изображение не задано, то рисуется прямоугольник указанного размера, цвета и прозрачности.

Элемент <ScreenOverlay> позволяет задать следующие параметры:

- размер изображения;
- позицию изображения на экране;
- угол поворота изображения вокруг заданного центра (при желании).

Для позиционирования экранного наложения задаются две точки – одна на экране, другая на изображении. Изображение «наклеивается» на экран, так что эти две точки совмещаются.

В пользовательском интерфейсе Google Earth нет механизма для создания экранных наложений. Их можно создать непосредственно в KML, как описано ниже, или с помощью специализированного стороннего приложения.

Синтаксис элемента <ScreenOverlay>

Ниже описан синтаксис элемента <ScreenOverlay> (описание потомков, наследуемых от Feature и Overlay, см. в разделе «Синтаксис элемента <Overlay>» выше и в приложении А).

```
<ScreenOverlay id="ID">
  <!-- элементы, унаследованные от Feature ->
  .
  .
  .
  <!-- элементы, унаследованные от Overlay ->
  .
  .
  .
  <!-- элементы, специфичные для ScreenOverlay ->
  <overlayXY x="1.0" y="1.0" xunits="fraction" yunits="fraction"/>
  <screenXY x="1.0" y="1.0" xunits="fraction" yunits="fraction"/>
  <rotationXY x="1.0" y="1.0" xunits="fraction" yunits="fraction"/>
  <size      x="1.0" y="1.0" xunits="fraction" yunits="fraction"/>
  <rotation>0.0</rotation>
</ScreenOverlay>
```

Задание единиц измерения x/y для накладываемых изображений и значков

В первых четырех потомках элемента <ScreenOverlay> используются единицы измерения x и y, в точности аналогичные одноименным элементам в потомке <hotSpot> элемента <IconStyle> (см. главу 4). Ниже приводится более подробное описание всех трех единиц измерения: `fraction`, `pixels` и `insetPixels`.

- `fraction` – задает долю от линейного измерения изображения или экрана, отмеряемую от левого нижнего угла (рис. 5.10, верхний).
- `pixels` – задает количество пикселей в направлении оси x (по горизонтали) или y (по вертикали), отсчитываемое от левого нижнего угла (рис. 5.10, средний).
- `insetPixels` – задает количество пикселей в направлении оси x (по горизонтали) или y (по вертикали), отсчитываемое от правого верхнего угла (рис. 5.10, нижний).

Единицы измерения не обязаны быть одинаковыми – ни по осям, ни для экрана и изображения.

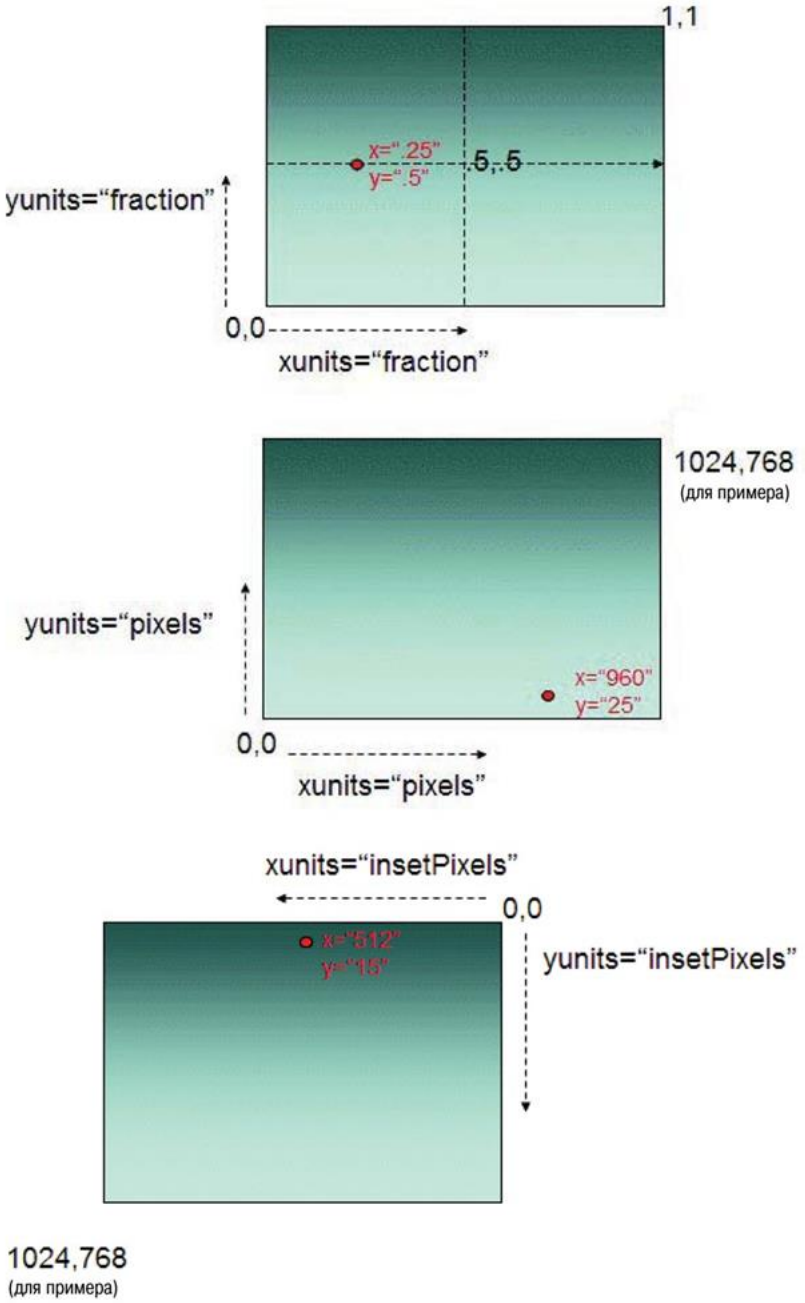


Рис. 5.10. Позицию экранного наложения можно задать в терминах *доли* экрана, количества *пикселей* (отсчитываемых от левого нижнего угла) или отступа (количество пикселей, отсчитываемых от правого верхнего угла). По разным осям можно задавать различные единицы измерения

Потомки элемента <ScreenOverlay>

Ниже приводится описание потомков элемента <ScreenOverlay>.

<overlayXY>

Задает точку внутри или вне накладываемого изображения, которая должна совмещаться с точкой на экране, заданной в элементе <screenXY>. В атрибу-

тах этого элемента задаются значения координат x и y, а также единицы измерения по осям. Атрибуты `xunits` и `yunits` могут принимать следующие значения: `fraction`, `pixels` и `insetPixels` (см. предыдущий раздел).

<screenXY>

Задает точку на экране, которая должна совмещаться с точкой изображения, заданной в элементе <overlayXY>. Атрибуты такие же, как для <overlayXY>.

<rotationXY>

Задает точку в координатах экрана, вокруг которой поворачивается экранное наложение. Например, (.5,.5) означает поворот вокруг центра экрана. По умолчанию поворот осуществляется вокруг правого верхнего угла экрана, то есть точки с координатами (1,1). Атрибуты такие же, как для <overlayXY>.

<size>

Задает размер накладываемого изображения. Атрибуты такие же, как для <overlayXY>. Их значения интерпретируются следующим образом:

–1 означает, что изменять размер изображения вообще не нужно;

0 означает, что следует сохранить пропорции;

n задает коэффициент масштабирования в направлении соответствующей оси.

<rotation>

Задает угол поворота накладываемого изображения. Значение представляет собой угол поворота в градусах против часовой стрелки, считая от направления на север (0). Допустимый диапазон – от –180 до +180 градусов (рис. 5.11).

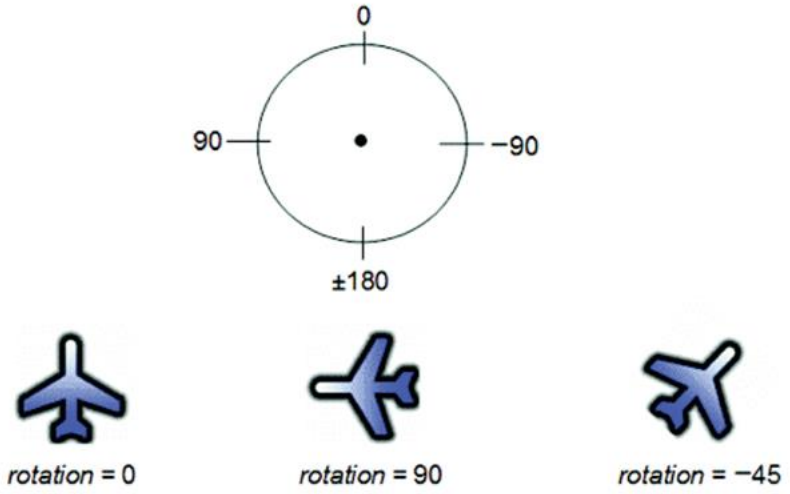


Рис. 5.11. Углы поворота экранного наложения задаются в градусах в диапазоне от –180 до +180. Для задания поворота вокруг точки, отличной от центра экрана, воспользуйтесь атрибутом «rotationXY»

Пример простого экранного наложения

В файле *ScreenOverlay.kml* приведен пример экранного наложения, размещенного в левом верхнем углу экрана.

ScreenOverlay.kml

```
<?xml version="1.0" encoding="utf-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <ScreenOverlay>
    <Icon>
      <href>BritishMuseumLogo.tif
      </href>
    </Icon>
    <overlayXY x="0" xunits="fraction" y="1" yunits="fraction"/>
    <screenXY x="0" xunits="fraction" y="1" yunits="fraction"/>
  </ScreenOverlay>
</kml>
```

Пример экранного наложения с элементом <ListStyle>

Пример *ScreenOverlayWithListStyle.kml* основан на предыдущем, но добавлен внутренний элемент <ListStyle>, в котором задан значок, отображаемый в списке. Кроме того, включен элемент <name>, с тем чтобы показываемая в списке информация об источнике бросалась в глаза.

ScreenOverlayWithListStyle.kml

```
<?xml version="1.0" encoding="utf-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <ScreenOverlay>
    <name>Публикуемся с разрешения Британского музея</name>
    <Style>
      <ListStyle>
        <ItemIcon>
          <href>http://maps.google.com/mapfiles/kml/shapes/homegardenbusiness.png</href>
        </ItemIcon>
      </ListStyle>
    </Style>
    <Icon>
      <href>BritishMuseumLogo.tif
      </href>
    </Icon>
    <overlayXY x="0" xunits="fraction" y="1" yunits="fraction"/>
    <screenXY x="0" xunits="fraction" y="1" yunits="fraction"/>
  </ScreenOverlay>
</kml>
```

Фотоналожение

Фотоналожения – это фотографии, встроенные прямо в ландшафт Земли. Это могут быть двумерные прямоугольники – эдакие «рекламные щиты» на основе Google Earth. Кроме того, фотоналожение можно спроецировать на цилиндр или сферу и создать тем самым виртуальную панораму, в которую пользователь может «влететь» и изучить во всех деталях. Фотоналожение помещается в указанной точке и ориентировано в направлении точки обзора. Если включить элемент-потомок <Point>, то с фотоналожением будет ассоциирован значок.

Гигапиксельные фотографии

Элемент <PhotoOverlay> умеет справляться с огромными изображениями, содержащими многие мегапиксели данных. Но для этого автор KML-файла должен предоставить частичные изображения, так чтобы геобраузер мог эффективно загрузить ту часть, которая относится к текущему фрагменту карты и при этом имеет нужный уровень детализации. Подробную информацию о создании *пирамиды изображений* см. ниже в разделе «Дополнительный материал: пирамиды для гигапиксельных изображений».

Синтаксис элемента <PhotoOverlay>

Ниже описан синтаксис элемента <PhotoOverlay> (описание потомков, наследуемых от Feature и Overlay, см. в разделе «Синтаксис элемента <Overlay>» выше и в приложении А).

```
<PhotoOverlay id="ID">
  <!-- элементы, унаследованные от Feature ->
  .
  .
  .
  <!-- элементы, унаследованные от Overlay ->
  .
  .
  .
  <!-- элементы, специфичные для PhotoOverlay ->
  <rotation>0.0</rotation>
  <ViewVolume>
    <leftFov>0.0</leftFov>
    <rightFov>0.0</rightFov>
    <bottomFov>0.0</bottomFov>
    <topFov>0.0</topFov>
    <near>0.0</near>
  </ViewVolume>
  <ImagePyramid>
    <tileSize>256</tileSize>
    <maxWidth>0</maxWidth>
    <maxHeight>0</maxHeight>
    <gridOrigin>lowerLeft</gridOrigin>
  </ImagePyramid>
  <Point>
    <coordinates>...</coordinates>
  </Point>
  <shape>rectangle</shape>
</PhotoOverlay>
```

В следующих разделах материал излагается не по порядку, а от простого к сложному. Когда вам понадобится вспомнить порядок следования потомков элемента <PhotoOverlay>, обратитесь к описанию синтаксиса в этом разделе или в приложении А.

Основные понятия: форма и точка

Изображение, входящее в состав фотоналожения, проецируется на форму <shape>, которая может быть прямоугольником `rectangle` (по умолчанию), ци-

линдром `cylinder` (для построения частичных или полных цилиндрических панорам) или сферой `sphere` (для построения сферических панорам). Основное изображение может быть снято панорамной камерой или представлять собой последовательность отдельных фотографий, «сшитых» специальной программой.

У элемента <PhotoOverlay> может быть потомок <Point>, который используется точно так же, как одноименный элемент, вложенный в <Placemark>. <Point> позволяет ассоциировать с фотоналожением значок и привязать наложение к конкретной точке. Значок можно стилизовать с помощью элементов <IconStyle> и <ListStyle>.

Поле зрения

Как вы уже знаете, с помощью элемента <LookAt> или <Camera> можно задать точку обзора для любой детали Feature. Элемент <PhotoOverlay> включает потомка <ViewVolume>, который определяет *поле зрения*. Задание поля зрения аналогично заданию диаметра отверстия объектива в физической камере. Если поле зрения мало, как в телескопическом объективе, то взгляд фокусируется на небольшом участке сцены. А если велико, как в широкоугольном объективе, то взгляд охватывает большую часть сцены.

Поле зрения для фотоналожения описывается четырьмя плоскостями, каждая из которых задается углом между ней и вектором взгляда. Эти плоскости называются верхней (top), нижней (bottom), левой (left) и правой (right) сторонами поля зрения, которое имеет вид усеченной пирамиды, как показано на рис. 5.12.

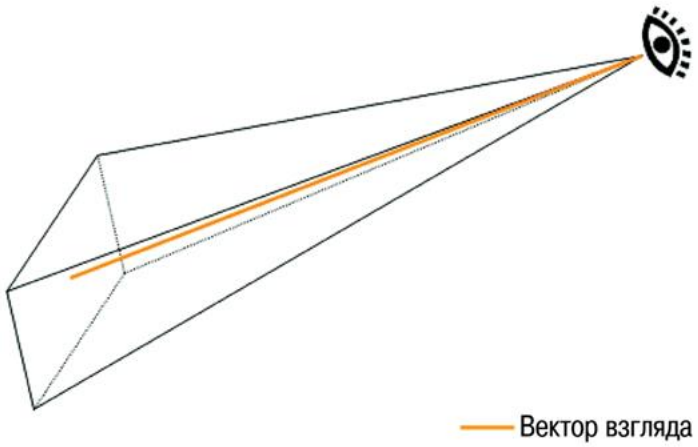


Рис. 5.12. Поле зрения определяется четырьмя плоскостями, задаваемыми относительно вектора взгляда

На рис. 5.13 показаны углы <rightFov> и <leftFov> (вид сверху), а также <topFov> и <bottomFov> (вид сбоку) этой пирамиды. Элемент <near> определяет расстояние в метрах вдоль направления взгляда между камерой и формой фотоналожения.

Типичная камера симметрична относительно вектора взгляда, то есть параметры задаются примерно так:

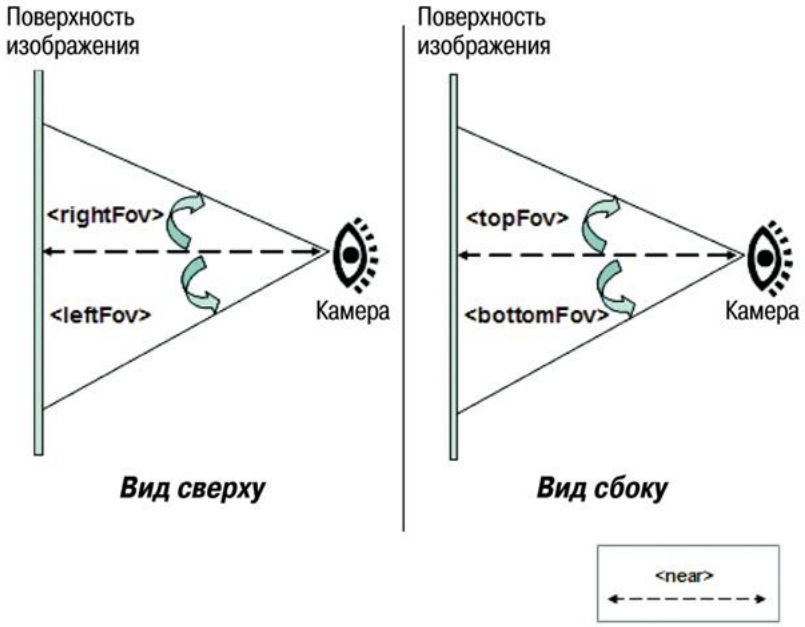


Рис. 5.13. Четыре плоскости определяют поле зрения, от которого зависит, какая часть фотографии видна

```
<ViewVolume>
  <near>1000</near>
  <leftFov>-60<leftFov>
  <rightFov>60<rightFov>
  <bottomFov>-60<bottomFov>
  <topFov>60<topFov>
</ViewVolume>
```

Поле зрения для прямоугольника

Для прямоугольника угол <topFov> должен быть меньше 90 градусов, а <bottomFov> – больше –90 градусов. Если хотя бы одно условие не соблюдается, то плоскость поля зрения вообще не будет пересекаться с изображением. Обычно значения <topFov> и <bottomFov> выбираются отрицательными.

Поле зрения для цилиндра

В случае цилиндрического изображения ось цилиндра совпадает с *вектором верха* (осью *y*), а радиус цилиндра равен <near>. Для цилиндров диапазон углов поля зрения такой:

```
-90 < bottomFov < topFov < 90
-180 < leftFov < rightFov < 180
```

Поле зрения для сферы

Центр сферического изображения совпадает с точкой расположения камеры (глазом). Радиус сферы равен <near>. Диапазоны углов такие же, как для цилиндра.

Подгонка поля зрения с помощью элемента <rotation>

Если фотография повернута и слегка отклоняется от требуемого горизонтального положения, то с помощью элемента <rotation> можно скорректировать ее так, чтобы она оказалась внутри поля зрения.

Дополнительный материал: пирамиды для гигапиксельных изображений

В этом разделе описывается, как подготовить очень большие изображения для эффективной загрузки в геобраузер. К счастью, существуют сторонние инструменты, позволяющие легко и быстро создавать пирамиды изображений. В качестве примеров можно назвать программу Google Earth PhotoOverlay Creator компании Digital Urban и программы, доступные для скачивания на сайте компании GigaPan по адресу www.gigapan.org.

Основная причина для создания пирамиды изображений – желание оптимизировать производительность при просмотре больших фотографий. На практике не имеет смысла создавать пирамиду, если размеры изображения не превышают 2K × 2K пикселей. Большинство современных графических карт могут справиться с изображениями размером до 4K × 4K пикселей. Если же размер больше, то стоит подумать о создании пирамиды изображений для фотоналожения.

Как работает пирамида изображений

Пирамида представляет собой иерархический набор изображений, являющихся копиями исходного с прогрессивно меньшим разрешением. Каждое изображение в пирамиде разбивается на ряд «плиток», чтобы можно было загружать только части, попадающие в поле зрения. Google Earth вычисляет текущую точку обзора и загружает те плитки, которые соответствуют расстоянию от наблюдателя до изображения. По мере того как точка обзора приближается к фотоналожению, Google Earth загружает плитки со все более высоким разрешением. Поскольку все пиксели исходного изображения невозможно увидеть на экране одновременно, предварительная обработка позволяет Google Earth достичь максимальной производительности, так как загружаются лишь те части, которые видны, и лишь те детали, которые пользователь может различить с данной точки.

Если изображение очень велико, то следует создать для него пирамиду и изменить элемент <href> внутри <Icon>, включив спецификации плиток в пирамиде. В следующих разделах описывается, как создать пирамиду вручную и как задать <href> для гигапиксельного изображения. На практике эту задачу лучше решать с помощью специальной программы.

Синтаксис элемента <ImagePyramid>

```
<ImagePyramid>
  <tileSize>256</tileSize>
  <maxWidth>0</maxWidth>
  <maxHeight>0</maxHeight>
  <gridOrigin>lowerLeft</gridOrigin><!-- lowerLeft или upperLeft->
</ImagePyramid>
```

<tileSize>

Размер плитки в пикселях. Плитки должны быть квадратными, а <tileSize> – степенью двойки. Рекомендуется выбирать размер 256 (по умолчанию) или 512. Исходное изображение разбивается на плитки указанного размера с разными разрешениями.

<maxWidth>

Ширина исходного изображения в пикселях.

<maxHeight>

Высота исходного изображения в пикселях.

<gridOrigin>

Задает начало нумерации плиток в каждом слое пирамиды. Значение `lowerLeft` (по умолчанию) означает, что плитка в строке 1 и столбце 1 должна считаться левым нижним углом сетки. Значение `upperLeft` означает, что нумерация начинается в левом верхнем углу сетки.

С помощью элементов <maxWidth> и <maxHeight> задается размер исходного изображения в пикселях. Ширина и высота могут быть произвольными числами, не обязательно степенями двойки. Недостающие пиксели можно сделать пустыми, как описано ниже.

Создание пирамиды изображений

Ниже предполагается, что размеры изображения в пикселях являются степенями двойки. Если это не так, то необходимо сначала дополнить изображение, как описано в разделе «Дополнение изображения» ниже. Затем нужно выполнить следующие действия:

1. Разбить исходное полноразмерное изображение на плитки заданного размера, например на блоки 256 × 256 пикселей.
2. Сжать изображение с коэффициентом 2.
3. Разбить новое изображение на плитки того же размера.
4. Повторять шаги 2 и 3, пока результирующее изображение не уместится целиком в одну плитку.

На рис. 5.14 показана небольшая пирамида, построенная таким способом.

Дополнение изображения

Если последняя плитка в строке неквадратная, то необходимо добавить столько прозрачных пикселей, чтобы сделать ее квадратной. Расположите изображение так, чтобы плитка (0,0) оказалась в начале координат. (Например, если начало координат находится в левом нижнем углу, то поместите изображение в левый нижний угол сетки.) Тогда строки и столбцы, нуждающиеся в дополнении, окажутся справа и сверху изображения. Для оптимальной фильтрации продублируйте последнюю строку (или столбец) с края изображения. Затем заполните пустующее место в плитках этой строки (или столбца), например черными пикселями.

Пример

В качестве примера рассмотрим изображение размером 3600 × 2700 пикселей (это составляет примерно 10 мегапикселей). Ниже описаны шаги по созданию для него пирамиды.

1. Если размер плитки равен 256 пикселей, то исходное изображение можно представить в виде сетки 16 ×16. (Это изображение в конечном итоге окажется на уровне 4 пирамиды.)

2. Добавьте пиксели, чтобы сделать квадратными неполные плитки в последнем столбце (правом) и последней строке (верхней в предположении, что <gridOrigin> равно `lowerLeft`).
3. Уменьшите изображение в два раза.
4. Разбейте новое изображение на плитки размером 256 пикселей. На этом уровне изображение представляет собой сетку 8×8 (уровень 3).
5. Уменьшите изображение на уровне 3 вдвое.
6. Разбейте новое изображение на плитки. На этом уровне получится сетка 4×4 (уровень 2).
7. Уменьшите изображение на уровне 2 вдвое.
8. Разбейте новое изображение на плитки. Теперь получается сетка 2×2 (уровень 1).
9. Уменьшите изображение на уровне 1 вдвое.
10. Получилось изображение размером 256×256 пикселей, и это последний уровень пирамиды (уровень 0).

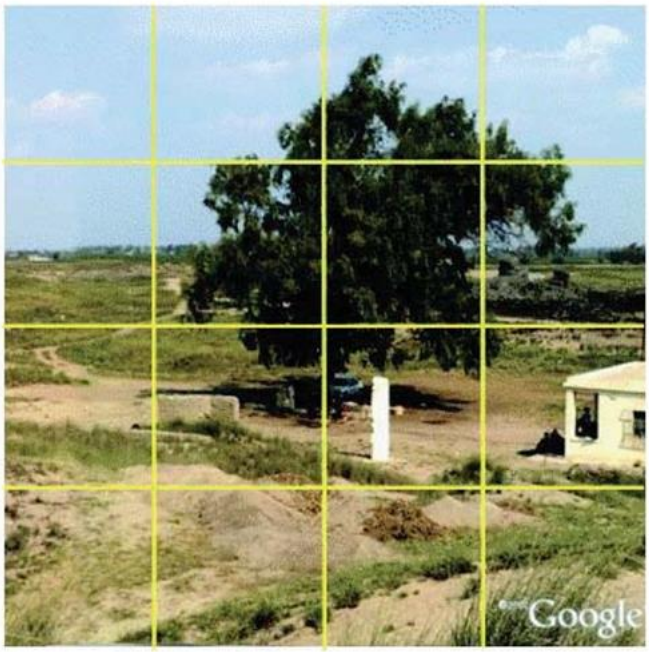
Пирамида для изображения размером 4096×4096 пикселей состоит из пяти уровней, как показано в табл. 5.2.

Таким образом, на уровне n имеется 2^n плиток в каждом направлении.

На рис. 5.14 показаны уровни 0, 1 и 2 построенной пирамиды.

Уровень 0

Уровень 1



Уровень 2

Рис. 5.14. Пирамида состоит из нескольких вариантов одного и того же изображения. На каждом уровне изображение разбивается на квадратные плитки. Поищите в Интернете программы, умеющие создавать такую пирамиду автоматически

Таблица 5.2. Пример пирамиды изображений с пятью уровнями

Уровень	Количество плиток	Размер изображения (в пикселях)
0	1	256 × 256
1	4 (сетка 2 × 2)	512 × 512
2	16 (сетка 4 × 4)	1024 × 1024
3	64 (сетка 8 × 8)	2048 × 2048
4	256 (сетка 16 × 16)	4096 × 4096

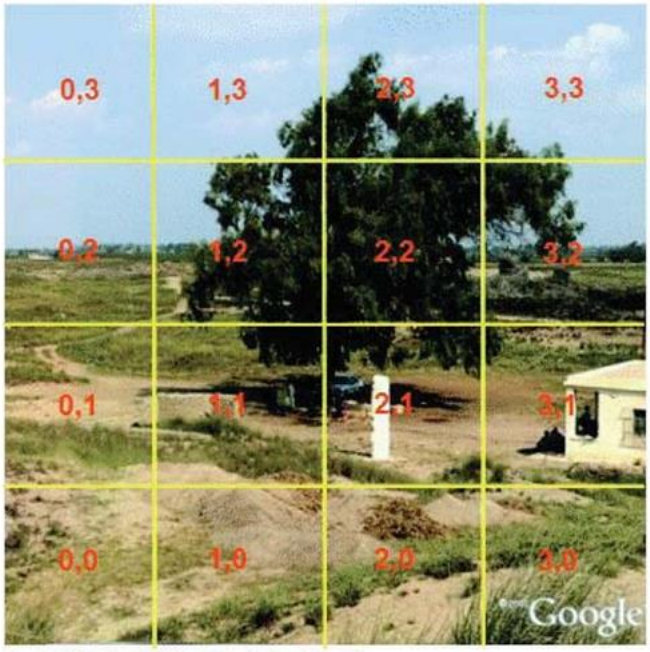
Нумерация плиток

Плитки на каждом уровне нумеруются, так чтобы Google Earth могла выбрать только те, что необходимы для текущей точки обзора. Каждая плитка идентифицируется тремя значениями:

- *x* номер строки;
- *y* номер столбца;
- *level* уровень пирамиды изображений; самый верхний уровень имеет номер 0.

По умолчанию начало координат (0,0) находится в левом нижнем углу сетки. Если для вашего изображения начало координат находится в левом верхнем углу, задайте элемент <gridOrigin> равным `upperLeft`.

На рис. 5.15 показана нумерация плиток на уровне 2 10-мегапиксельного изображения при условии, что начало координат находится в левом нижнем углу.



Уровень 2 (нумерация плиток)

Рис. 5.15. В этом примере начало координат сетки находится в левом нижнем углу изображения, а плитки нумеруются, начиная с левого верхнего угла и заканчивая правым нижним

Задание URL гигапиксельного изображения

Для гигапиксельных изображений в спецификации элемента <href> внутри <Icon> должны быть специальные метки для задания *x*, *y* и *level* тех плиток, которые Google Earth должна загрузить. Например, URL изображения может выглядеть так:

```
http://server.company.com/bigphoto/$[level]/row_$[x]_column_$[y].jpg
```

Google Earth подставит вместо этих меток нужные значения, когда будет отправлять HTTP-запрос. Например, если Google Earth потребуется плитка в строке 2, столбце 1 на уровне 3, то она обратится к следующему URL:

```
http://server.company.com/bigphoto/3/row_2_column_1.jpg
```

Прозрачность

Если ваше изображение непрозрачно, пользуйтесь форматом JPEG. Если часть изображения прозрачна, а другая часть непрозрачна, можно одни плитки сохранить в формате PNG, а другие – в формате JPEG. Применяйте формат PNG только для тех плиток, где имеются прозрачные пиксели. В случае смешения форматов не указывайте расширение файла в спецификации <href> для файла изображения в целом, а включайте его в имя файла каждой плитки.

Пример фотоналожения

В файле *PhotoOverlay.kml* приведен пример цилиндрического фотоналожения и точки обзора, заданной элементом <Camera>. Обратите внимание, что элемент <description> содержит ссылку на метку **Placemark**, что позволяет пользователю «влететь» в панорамное фотоналожение. Поскольку изображение очень велико, автор этого KML-файла Стефан Джинс создал пирамиду и включил описывающие ее параметры, так чтобы Google Earth могла затребовать отдельные плитки, а не загружать всю картинку целиком.

PhotoOverlay.kml

```
<?xml version="1.0" encoding="utf-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <PhotoOverlay id="kom_firin_360">
    <name>Ком-Фирин 360</name>
    <description>
      <![CDATA[
        <p>Панорама раскопок храма Рамсеса археологами Британского музея,
         снимок сделан 6 октября 2007.</p><p><a href="#kom_firin_360">Влететь в
         панораму</a>.</p>
      ]]>
    </description>
    <Camera>
      <longitude>30.4882478804576</longitude>
      <latitude>30.8634158327095</latitude>
      <altitude>20</altitude>
      <heading>53.5</heading>
      <tilt>90</tilt>
      <roll>0</roll>
    </Camera>
    <Style>
```

```
      <IconStyle>
        <Icon>
          <href>http://maps.google.com/mapfiles/kml/shapes/camera.png
           </href>
        </Icon>
      </IconStyle>
    </Style>
    <Icon>
      <href>http://www.ogleearth.com/komfirin360/komfirin360_$[level]_$[x]_$[y].jpg</href>
    </Icon>
    <ViewVolume>
      <leftFov>-180</leftFov>
      <rightFov>180</rightFov>
      <bottomFov>-24.5</bottomFov>
      <topFov>12.7</topFov>
      <near>65</near>
    </ViewVolume>
    <ImagePyramid>
      <maxWidth>16390</maxWidth>
      <maxHeight>1757</maxHeight>
      <gridOrigin>upperLeft</gridOrigin>
    </ImagePyramid>
    <Point>
      <altitudeMode>relativeToGround</altitudeMode>
      <coordinates>30.4882478804576,30.8634158327095,13</coordinates>
    </Point>
    <shape>cylinder</shape>
  </PhotoOverlay>
</kml>
```

Что дальше?

В главе 6 мы познакомимся с сетевыми ссылками – эффективным механизмом обобществления и обновления данных по Сети. Кроме того, сетевые ссылки позволяют организовать большие презентации в виде набора небольших файлов, которые можно загружать независимо – с локального диска или по Сети. Обсуждаются также дополнительные особенности элемента <Link>, дающие возможность обновлять сетевые ссылки в зависимости от выбранной пользователем точки обзора или вследствие изменения данных. Еще мы покажем, как геобраузер может передать определенную информацию серверу.

Глава 6. Сетевые ссылки

Прочитав эту главу, вы сможете:

- ✓ назвать несколько причин полезности сетевых ссылок;
- ✓ создать сетевую ссылку для получения KML-файла с веб-сервера;
- ✓ создать сетевую ссылку, которая каждые полчаса будет обновлять полученный KML-файл;
- ✓ создать сетевую ссылку, которая будет обновлять полученный KML-файл спустя 5 секунд после того, как пользователь перейдет в другое место на карте Google Earth;
- ✓ объяснить, в каких отношениях находятся веб-клиент и веб-сервер.

Предварительный обзор

Сетевая ссылка содержит URL какого-нибудь KML-файла или KMZ-архива. Когда программа Google Earth открывает исходный KML-файл, она также ищет и загружает KML-файл, на который имеется ссылка в элементе <NetworkLink>. Этот файл может быть *локальным* или *удаленным* (то есть находиться на веб-сервере). Наличие ссылки на файл, хранящийся на сервере, позволяет пользователям Google Earth просматривать данные, которые обновляются удаленно или требуют специальной обработки на стороне сервера (рис. 6.1).

Загруженный KML-файл может содержать:

- ❑ иерархию элементов, производных от Feature;
- ❑ один (необязательный) элемент <NetworkLink>, который управляет поведением исходного KML-файла.

Мы начнем эту главу с ответа на вопрос, для чего могут быть полезны сетевые ссылки. Мы обсудим основы взаимодействия между клиентом и сервером и дадим несколько советов по настройке веб-сервера для тестирования функциональности сетевых ссылок в KML. Но для завершения настройки вам придется ознакомиться с документацией, поставляемой в комплекте с веб-сервером.

Сетевая ссылка, загруженная в Google Earth, может быть обновлена (то есть загружена повторно), когда происходит какое-то событие при просмотре карты, или по времени, или при изменении элемента <Link>. Эту крайне важную особенность сетевых ссылок мы обсудим ниже и приведем примеры обновления ссылки по событиям просмотра.

Сетевые ссылки также позволяют Google Earth *отправлять данные серверу*, где написанный пользователем сценарий обработает их перед возвратом ответа клиенту. Будут рассмотрены потомки элемента <NetworkLink>, необходимые для реализации этой функциональности:

- ❑ <viewFormat>;

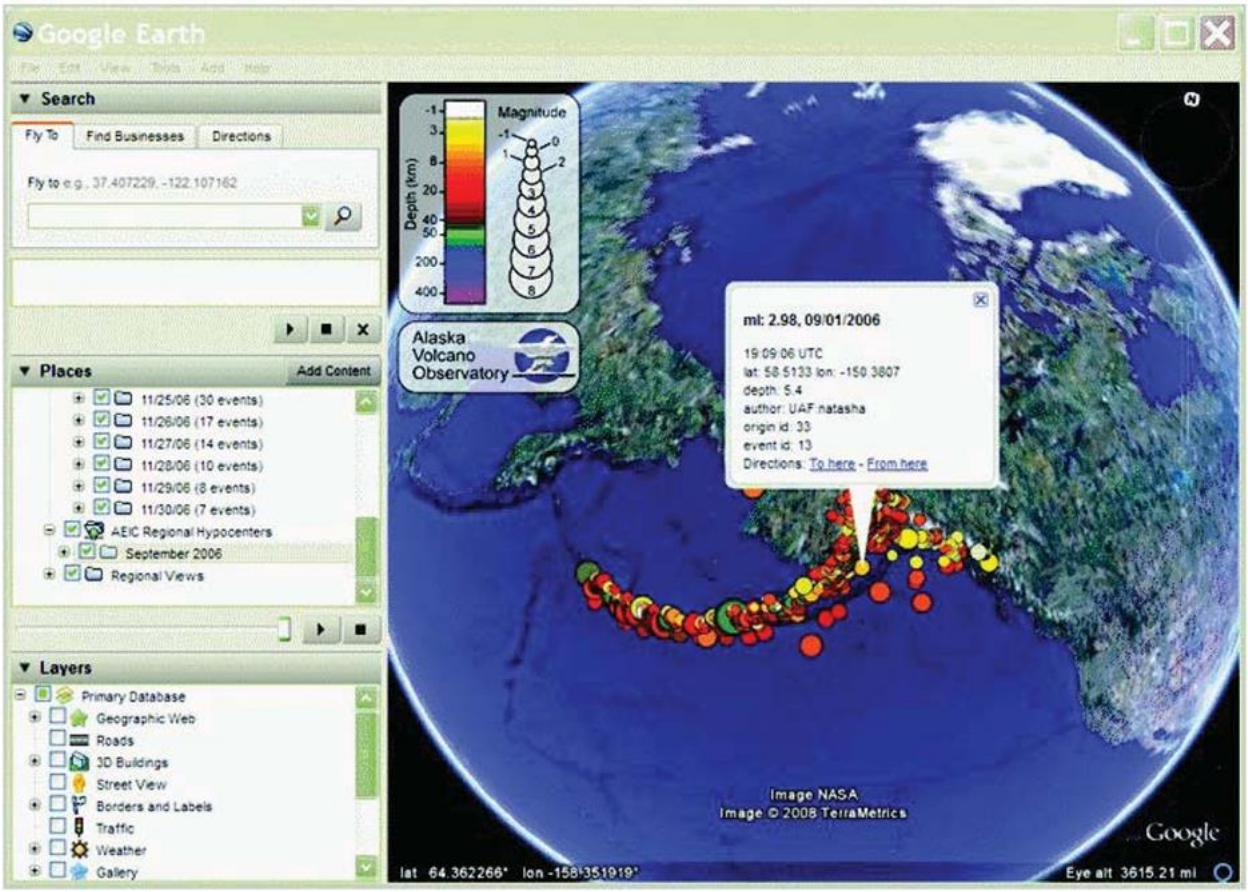


Рис. 6.1. Вулканическая обсерватория Аляски ведет сейсмический мониторинг зон вулканической активности на Аляске и Алеутских островах. По радиоканалам и телефонным линиям данные передаются в Университет Аляски в Фербэнксе и отделение Геологической службы США (USGS) в Анкоридже, где анализируются и обрабатываются. Актуальная информация публикуется на веб-сервере в виде KML-файла, чтобы с ней могли ознакомиться удаленные пользователи, установившие программу Google Earth (публикуется с разрешения Вулканической обсерватории Аляски и Университета Аляски в Фербэнксе, http://ge.images.alaska.edu)

- <viewBoundScale>;
- <httpQuery>.

Контроллером сетевой ссылки называется KML-файл, который содержит элемент <NetworkLink> и располагается на веб-сервере. Мы завершим эту главу рассмотрением того, как контроллеры сетевых ссылок задают или переопределяют их поведение и как контроллер может повлиять на некоторые элементы, содержащиеся внутри сетевой ссылки.

Что такое сетевая ссылка?

Сетевая ссылка в KML ссылается на KML-файл или KMZ-архив, находящийся на локальном диске или в Сети. Встретив такую ссылку, программа Google Earth загружает указанный в ней файл в память и отображает его на трехмерной

панораме. Загруженный файл, в свою очередь, может содержать сетевую ссылку на другой KML-файл. Тем самым создается иерархия KML-файлов, связанных сетевыми ссылками.

Сетевая ссылка как механизм группировки

Элемент <NetworkLink> является производным от Feature, то есть может иметь потомков <name>, <visibility>, <open>, <description>, <Camera> или <LookAt> и вообще любых потомков Feature, которые рассматривались в предыдущих главах. Кроме того, <NetworkLink> – это некий *механизм группировки*, работающий аналогично тому, как элемент <Folder> на панели списка в Google Earth. Детали (Feature), присутствующие в загруженном KML-файле, представляются в списке как потомки элемента <NetworkLink>. С сетевой ссылкой, как и с папкой, ассоциируются флажок и значок (рис. 6.2). Рассмотренный в главе 4 «Стили и значки» элемент <listItemType>, вложенный в <ListStyle>, можно применять и для настройки внешнего вида сетевой ссылки в списке. Если загруженный по ссылке KML-файл содержит сотни деталей, то имеет смысл создать для сетевой ссылки элемент <Style> и задать в нем параметр `checkOffOnly` или `checkHideChildren`, чтобы не загромождать список лишней информацией.

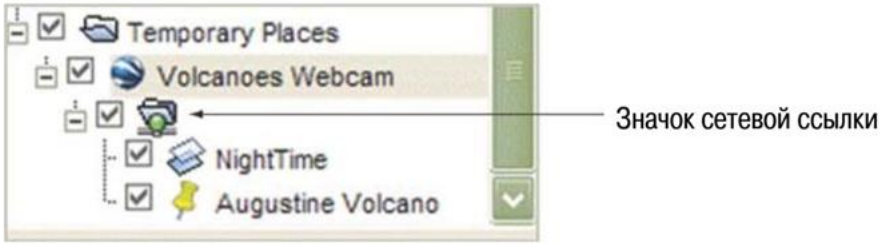


Рис. 6.2. С сетевой ссылкой ассоциирован значок в списке, и ведет она себя как папка. На панели списка в Google Earth объекты, находящиеся в связанном KML-файле, выглядят как потомки сетевой ссылки

Примечание

Хотя в Google Earth детали загруженного по ссылке KML-файла размещаются внутри папки или документа, этот контейнер верхнего уровня показывается в списке, лишь если ему присвоен идентификатор ID. Подробнее об идентификаторах см. главу 7.

Локальные и удаленные сетевые ссылки

Несмотря на упоминание о Сети в названии, сетевая ссылка может указывать не только на удаленный, но и на локальный файл. Сетевые ссылки полезны как способ разбиения очень больших файлов на меньшие части, с которыми проще иметь дело. Применяя относительные ссылки, группу файлов, связанных элементами <NetworkLink>, можно разместить локально (на своей машине) или удаленно, не меняя содержимого элемента <href>.

Для чего нужны сетевые ссылки?

Сетевые ссылки полезны в ряде случаев, в частности:

- если данные распределены по большой площади, то встроенная в элемент <NetworkLink> функция *обновления по событиям просмотра карты* позволяет Google Earth выполнять пространственный поиск, исходя из текущей точки обзора, и загружать только те данные, которые пользователь может увидеть;
- если данные часто изменяются, то функция *периодического обновления* позволяет своевременно обновлять отображаемые данные;
- для очень больших наборов данных механизм *регионов* элемента <NetworkLink> позволяет контролировать, какие части набора показываются на данном фрагменте карты (см. главу 8 «Большие наборы данных»);
- если вы уже передали большой объем данных, то можете воспользоваться встроенной в элемент <NetworkLinkControl> функцией *обновления*, чтобы вносить мелкие изменения в данные, загруженные по ссылке <NetworkLink> (см. главу 7 «Динамический KML»);
- сложный KML-файл можно разбить на несколько файлов и связать их ссылками. Это относится как к локальным, так и к удаленным файлам.

Примечание

Для краткости в этой главе загруженный по ссылке файл мы будем называть просто «KML-файлом», но все сказанное относится в равной мере и к KMZ-архивам. Google Earth автоматически распаковывает и загружает KMZ-архив точно так же, как простой KML-файл.

Основы клиент-серверной архитектуры

Программа Google Earth является по существу *веб-клиентом* (работающим на вашем локальном компьютере), который может через Интернет общаться с любыми *веб-серверами* (находящимися где-то в Сети). На рис. 6.3 показано, что Google Earth отправляет веб-серверам *запросы*, на которые те возвращают *ответы*.

Обычно веб-клиент общается с веб-сервером по протоколу HTTP (Hypertext Transfer Protocol). Когда клиент отправляет запрос, он на самом деле посылает серверу команду GET по протоколу HTTP. В состав этой команды входит имя запрашиваемого ресурса и другая необходимая информация. Запрашиваемый ресурс может быть HTML-файлом (возможно, содержащим JavaScript-сценарий), XML-файлом или даже программой, написанной на таких языках, как Perl или PHP. Такие программы пишутся в соответствии со спецификацией CGI (Common Gateway Interface – общий шлюзовой интерфейс).

В случае сетевой ссылки Google Earth посылает серверу запрос на KML-файл, как показано на рис. 6.4. Сервер возвращает KML-файл в качестве ответа. Здесь под «файлом» понимается поток байтов – не просто имя файла, а его содержимое.

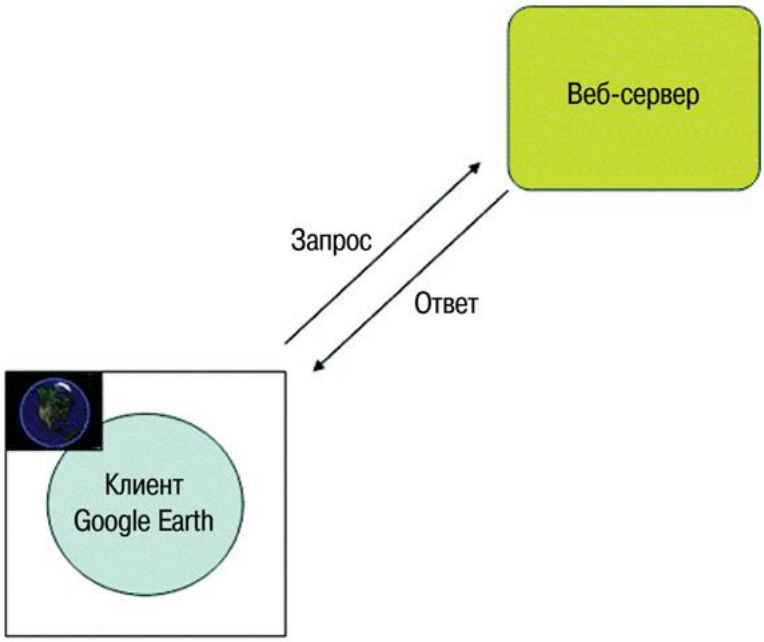


Рис. 6.3. Программа Google Earth – это веб-клиент, обменивающийся данными с веб-сервером

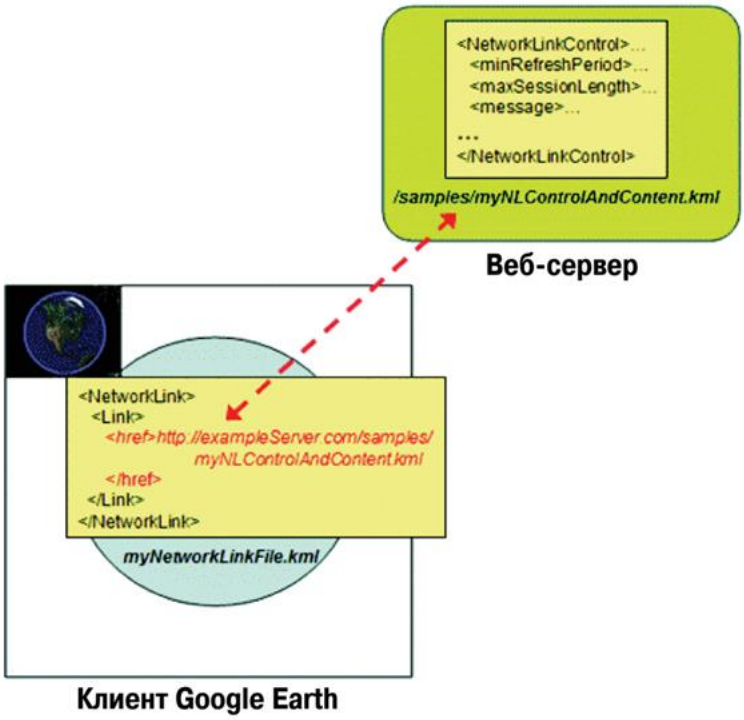


Рис. 6.4. Сетевая ссылка содержит URL KML-файла, который Google Earth скачивает и загружает в память

Настройка веб-сервера

Существует много различных веб-серверов; одни настраивать проще, другие – сложнее. К наиболее популярным можно отнести Apache, Microsoft Internet

Information Services, iPlanet, WebSite, WebStar и Lotus Domino. Программа **lighttpd** (произносится «лайти») (www.lighttpd.net), как следует из названия, является облегченным веб-сервером, который довольно легко установить и сконфигурировать.

Выберите какой-нибудь сервер, установите и сконфигурируйте его в соответствии с документацией. Если вас прежде всего интересует тестирование примеров KML, то подойдет и такой простой сервер, как lighttpd.

Конфигурирование MIME-типа KML/KMZ

MIME-тип сообщает операционной системе, какое приложение запускать для отображения конкретного файла. Аббревиатура MIME расшифровывается как «Multipurpose Internet Mail Extensions» (многоцелевые расширения электронной почты в Интернете). Хотя первоначально стандарт MIME предназначался для описания форматов электронной почты, теперь он применяется для идентификации различных типов содержимого. Для правильной работы сетевых ссылок веб-сервер необходимо сконфигурировать так, чтобы он распознавал типы содержимого KML и KMZ. Точный синтаксис описания MIME-типа в разных веб-серверах немного различается. MIME-тип KML описывается следующим образом:

```
application/vnd.google-earth.kml+xml
```

а KMZ – так:

```
application/vnd.google-earth.kmz
```

Например, для сервера Apache необходимо добавить следующие строки в конфигурационный файл *httpd.conf*:

```
AddType application/vnd.google-earth.kml+xml .kml
AddType application/vnd.google-earth.kmz .kmz
```

В случае сервера lighttpd нужно добавить в файл *lighttpd.conf* такие строки:

```
mimetype.assign = <
".kml" => "application/vnd.google-earth.kml+xml"
".kmz" => "application/vnd.google-earth.kmz"
```

Установка языков программирования и средств для работы с базами данных

Дополнительно нужно будет установить языки, на которых вы собираетесь писать серверные сценарии, и соответствующим образом сконфигурировать сервер. Пример сценария в этой главе написан на PHP, а для одного из примеров в главе 7 потребуется язык Python. Если сценарий обращается к базе данных, то необходимо будет установить и настроить соответствующую СУБД (для примера из этой главы необходима СУБД MySQL).

Стратегии тестирования сетевых ссылок

Если вы будете пользоваться относительными именами файлов, то сможете протестировать ссылки локально, а затем перенести всю иерархию файлов на веб-сервер, не меняя ничего внутри самих KML-файлов. И этим переход к промышленному режиму исчерпывается. В URL исходного файла, отправляемого пользователю, должен присутствовать префикс *http://*, чтобы геобраузер знал, что

нужно обратиться к веб-серверу. Однако внутри KML-файлов этот префикс уже не нужен (поскольку все пути в них относительные, а не абсолютные).

Абсолютные и относительные ссылки на файлы

Исходный KML-файл (тот, что содержит элемент <NetworkLink>) считается «корневым», то есть находящимся на вершине иерархии файлов. В элементе <href>, вложенном в элемент <Link> исходного файла, все внешние файлы задаются *относительно* корня.

Пусть, например, имеется файл *doc.kml*, находящийся в папке *C:\MyDocuments\Projects\CityPlanning*. Этот файл содержит элемент <NetworkLink> с вложенным элементом <Link>, который указывает на KML-файл *myHouses.kml*. Проще всего поместить оба файла в одну папку (на рис. 6.5 она называется *CityPlanning*). Тогда на связанный файл можно будет сослаться по имени:

```
<href>myHouses.kml</href>
```

Альтернативно можно завести подпапку внутри *CityPlanning* и назвать ее *philadelphia*. Если поместить файл *myHouses.kml* в эту подпапку, как показано на рис. 6.6, то элемент <href> в файле *doc.kml* нужно будет записать так:

```
<href>philadelphia/myHouses.kml</href>
```

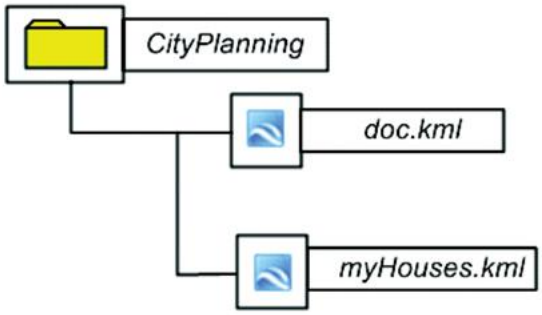


Рис. 6.5. Размещение связанного файла в одной папке с исходным

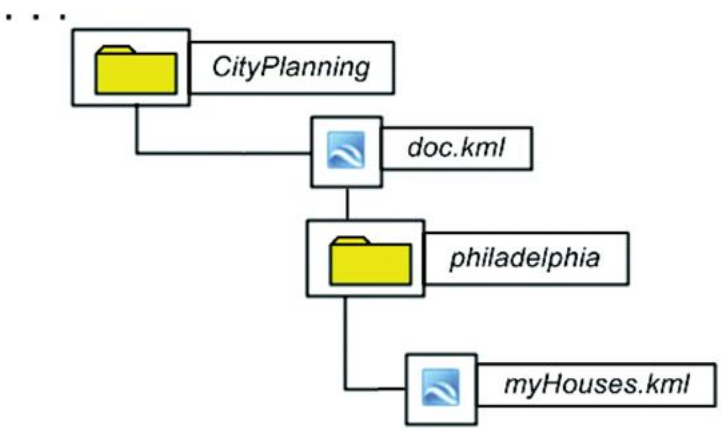


Рис. 6.6. Размещение связанного файла в подпапке той папки, в которой находится исходный файл

Эквивалентная абсолютная ссылка, то есть полная спецификация пути к файлу, выглядит так:

```
C:\MyDocuments\Projects\CityPlanning\Philadelphia\myHouses.kml
```

Она будет работать, если все файлы находятся на вашем компьютере. Но при переносе их на другой компьютер такая сетевая ссылка перестанет работать, если только вы не создадите на новой машине точно такую же структуру каталогов, как на исходной. Поэтому в URL всех файлов лучше использовать *относительные ссылки*.

Синтаксис элемента <NetworkLink>

Элемент <NetworkLink> может содержать все элементы, определенные для Feature, а также ряд собственных:

```
<NetworkLink id="ID">
  <!-- унаследованы от элемента Feature ->
  <name>...</name>
  <visibility>1</visibility>
  <open>0</open>
  <atom:author>...</atom:author>
  <atom:link>...</atom:link>
  <address>...</address>
  <xal:AddressDetails>...</xal:AddressDetails>
  <phoneNumber>...</phoneNumber>
  <Snippet maxLines="2">...</Snippet>
  <description>...</description>
  <AbstractView>...</AbstractView>
  <TimePrimitive>...</TimePrimitive>
  <styleUrl>...</styleUrl>
  <StyleSelector>...</StyleSelector>
  <Region>...</Region>
  <ExtendedData>...</ExtendedData>

  <!-- специфичны для NetworkLink ->
  <refreshVisibility>0</refreshVisibility>
  <flyToView>0</flyToView>
  <Link id="ID">
    <href>...</href>
    <refreshMode>onChange</refreshMode>
    <refreshInterval>4.0</refreshInterval>
    <viewRefreshMode>never</viewRefreshMode>
    <viewRefreshTime>4.0</viewRefreshTime>
    <viewBoundScale>1.0</viewBoundScale>
    <viewFormat>BBOX=
      [bboxWest],[bboxSouth],[bboxEast],[bboxNorth]
    </viewFormat>
    <httpQuery>...</httpQuery>
  </Link>
</NetworkLink>
```

<refreshVisibility>

Булевское значение (0 или 1). Значение 0 (по умолчанию) отдает управление видимостью детали в руки пользователя Google Earth. Значение 1 означает, что видимость детали должна сбрасываться при каждом обновлении элемента <Link>.

После обновления (повторной загрузки) KML-файла Google Earth освобождает память, занятую предыдущей версией, и строит визуальное представле-

ние заново. Поэтому, если всплывающее окно метки было открыто в момент обновления, оно закрывается.

Предположим, что в элементе <NetworkLink> параметр <refreshVisibility> равен 1, а загружаемый по этой ссылке файл содержит метку <Placemark>, в которой <visibility> равно 1. В таком случае после первой загрузки сетевой ссылки метка будет видна. Предположим далее, что пользователь сбросил ассоциированный с этой меткой флажок в списке, сделав ее невидимой. При обновлении ссылки метка перерисовывается заново и снова становится видимой, поскольку параметр <visibility> в KML-файле равен 1 (TRUE).

<flyToView>

Булевское значение (0 или 1). Значение 0 (по умолчанию) говорит, что элемент AbstractView (то есть <Camera> или <LookAt>) следует игнорировать. Значение 1 заставляет Google Earth «перелететь» к точке обзора, заданной в элементе <Camera> или <LookAt>, вложенном в элемент <NetworkLinkControl> в KML-файле (если таковой имеется). Если элемент <NetworkLinkControl> не содержит AbstractView, то Google Earth «перелетает» к точке обзора, заданной в элементе <Camera> или <LookAt>, который вложен в потомка Feature элемента <kml> в загруженном файле (напомним, что у элемента <kml> может быть не более одного потомка Feature). Если в элементе <kml> нет ни <Camera>, ни <LookAt>, то Google Earth использует точку обзора, подразумеваемую по умолчанию.

Если этот элемент равен 1, то Google Earth должна перелететь к точке обзора <LookAt>, заданной в родительском элементе <Document> в загруженном файле. Она не будет перелетать к точкам <LookAt>, указанным для меток, вложенных в <Document>, даже если на этом, более низком уровне иерархии элементы <LookAt> присутствуют. (Такая политика позволяет автору KML-файла в какой-то мере управлять начальной точкой обзора, а кроме того, оставляет Google Earth возможность применить эффективный механизм быстрого поиска подходящего элемента AbstractView.)

Отметим, что элемент <flyToView> следует применять с осторожностью, поскольку он лишает пользователя контроля над выбором точки обзора в Google Earth. В большинстве случаев более естественно положиться на мнение пользователя.

<Link>

<href>

В элементе <NetworkLink> URL, заданный параметром <href> в элементе <Link>, может вести на KML-файл или KMZ-архив, либо на программу, возвращающую такой файл (его содержимое, а не просто имя). Например, сервер может иметь доступ к большой базе данных. Получив от Google Earth запрос на сетевую ссылку, сервер запускает сценарий, который извлекает данные из базы, обрабатывает их и по результатам обработки создает и возвращает KML-файл.

Если заданный в <href> URL ведет на программу, то в нем *обязательно* должен быть указан префикс *http://*. Локально запускать программу, заданную в элементе <NetworkLink>, не разрешается.

<refreshMode>

Задает режим обновления: по времени или при изменении параметров, заданных в элементе <Link>. Допустимы следующие значения:

`onChange` (по умолчанию)

Обновлять после загрузки изображения и при изменении любого параметра, заданного в <Link>.

`onInterval`

Обновлять каждые *n* секунд, где *n* – значение параметра <refreshInterval>.

`onExpire`

Обновлять по истечении указанного срока. Этот режим основан на моменте времени, заданном в элементе <expires>, вложенном в элемент <NetworkLinkControl>, который будет описан ниже. Если элемент <expires> не задан, то Google Earth смотрит, указан ли какой-нибудь срок хранения в HTTP-заголовках (см. приложение А).

<refreshInterval>

Указывает, что изображение должно обновляться каждые *n* секунд.

<viewRefreshMode>

Задает режим обновления по событиям просмотра карты. Допустимы следующие значения:

`never` (по умолчанию)

Не обращать внимания на события просмотра. Параметры, заданные в элементе <viewFormat>, также игнорируются.

`onStop`

Обновлять изображение спустя *n* секунд после прекращения движения, где *n* – значение, заданное в элементе <viewRefreshTime>.

`onRequest`

Обновлять изображение, только когда пользователь явно запрашивает. (Например, в Google Earth пользователь может щелкнуть правой кнопкой мыши и выбрать из контекстного меню команду **Обновить**.)

`onRegion`

Обновлять изображение, когда становится активным определенный регион. См. главу 8 «Большие наборы данных».

<viewRefreshTime>

Определяет, сколько секунд ждать после смены точки обзора перед тем, как приступать к обновлению изображения (используется в режиме **<viewRefreshMode>**onStop**</viewRefreshMode>**).

Примечание

У элементов <Icon> и <Link> одинаковые наборы потомков. Но <Icon> используется внутри элементов наложения Overlay, а <Link> – совместно с элементами <NetworkLink> и <Model>.

В разделе «Отправка серверу дополнительных данных с помощью <httpQuery>» ниже мы рассмотрим оставшихся потомков элемента <Link> (и <Icon>).

Пример простой сетевой ссылки

В этом простом примере сетевая ссылка указывает на файл *AugustineWebcam.kml*, содержащий экранное наложение, которое связано с веб-камерой, направленной на вулкан Августин (рис. 6.7). Предполагается, что исходный и загружаемый файл находятся в *одном и том же каталоге*, поэтому можно пользоваться относительными ссылками. Если бы файлы находились в разных каталогах, то пути к файлам пришлось бы подправить.

Простая сетевая ссылка: исходный файл

```
<?xml version="1.0" encoding="utf-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Document>
    <name>Веб-камера, направленная на вулкан</name>
```

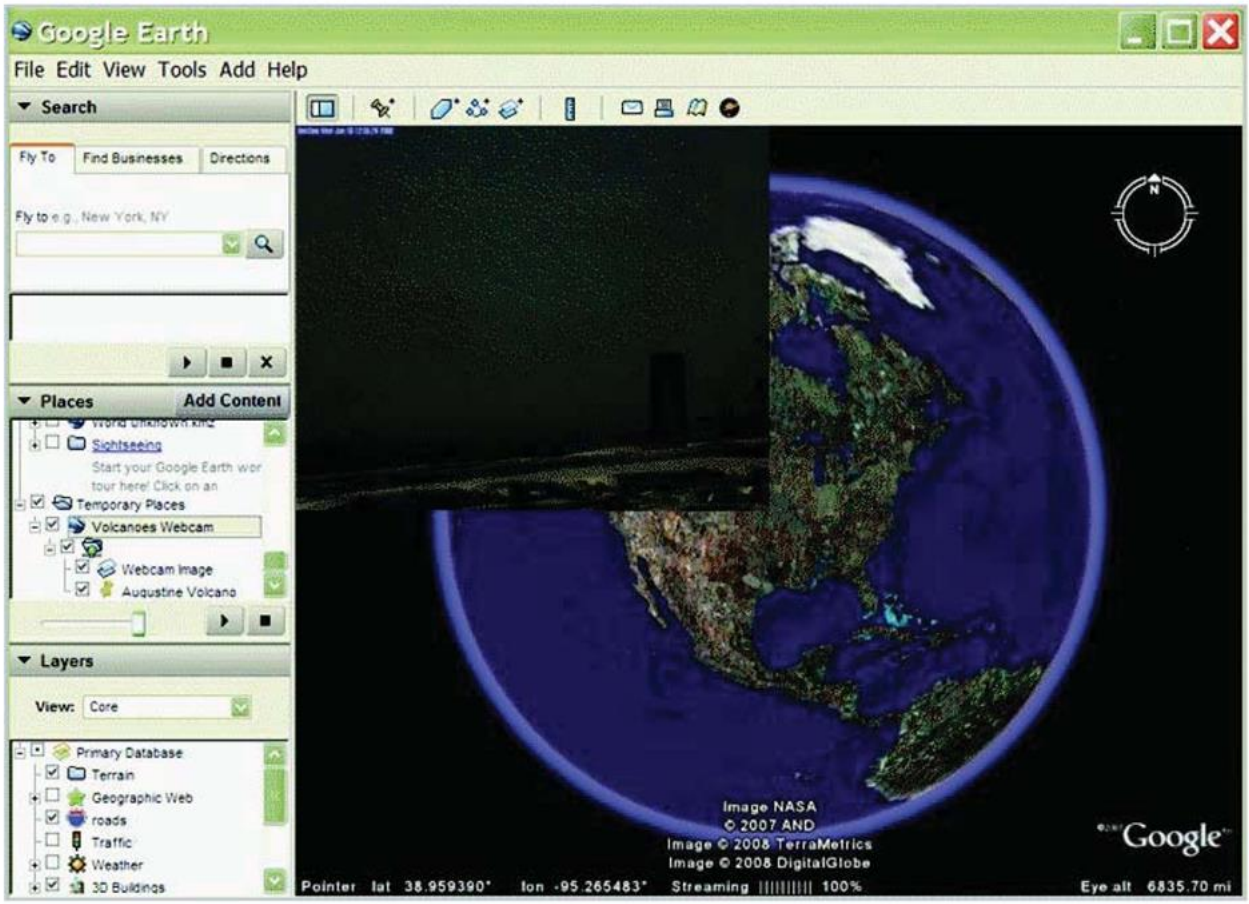


Рис. 6.7. По сетевой ссылке загружается KML-файл, содержащий изображение с веб-камеры, направленной на вулкан Августин на Аляске (публикуется с разрешения Вулканической обсерватории Аляски и Университета Аляски в Фербэнксе, http://ge.images.alaska.edu)

```
    <NetworkLink>
      <Link>
        <href>AugustineWebcam.kml</href>
      </Link>
    </NetworkLink>
  </Document>
</kml>
```

Простая сетевая ссылка: загружаемый файл (AugustineWebcam.kml)

```
<?xml version="1.0" encoding="utf-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Document>
    <name>Изображения вулкана Августин</name>
    <LookAt>
      <longitude>-133.768512314146</longitude>
      <latitude>56.447578192874</latitude>
      <altitude>0</altitude>
      <heading>-10.5952483593619</heading>
      <tilt>0</tilt>
      <range>7382344.56134118</range>
      <altitudeMode>relativeToGround</altitudeMode>
    </LookAt>
    <ScreenOverlay>
      <name>Изображение с веб-камеры</name>
      <Icon>              <!-- ссылка на веб-камеру ->
        <href>http://www.avo.alaska.edu/webcam/augustine.jpg</href>
      </Icon>
      <overlayXY x="0" xunits="fraction" y="1" yunits="fraction"/>
      <screenXY x="0" xunits="fraction" y="1" yunits="fraction"/>
      <size x="0.5" xunits="fraction" y="0.5" yunits="fraction"/>
    </ScreenOverlay>
    <Placemark>
      <name>Вулкан Августин</name>
      <Point>
        <coordinates>-153.456174,59.368623,0</coordinates>
      </Point>
    </Placemark>
  </Document>
</kml>
```

Обновление сетевых ссылок

Одно из достоинств сетевых ссылок заключается в возможности задать, с какой частотой и при каких условиях должен обновляться указанный в ссылке файл. Существуют два основных способа обновления:

- по времени или при обнаружении изменений;
- по событиям просмотра карты.

Обновление по времени или при обнаружении изменений

Если вы ожидаете, что данные или изображение будут изменяться часто, то имеет смысл задать параметр <refreshMode> в элементе <Link>, вложенном в <NetworkLink>. Можно указать один из трех режимов обновления: `onChange`, `onInterval` или `onExpire`.

В следующем фрагменте показано, как задать обновление по времени в рассмотренном выше примере простой сетевой ссылки. В результате изображение с веб-камеры будет перезагружаться каждый час.

```
<?xml version="1.0" encoding="utf-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Document>
    <name>Веб-камера, направленная на вулкан</name>
    <NetworkLink>
      <Link>
        <href>AugustineWebcam.kml</href>
        <refreshMode>onInterval</refreshMode>
        <refreshInterval>3600</refreshInterval>
      </Link>
    </NetworkLink>
  </Document>
</kml>
```

Обновление по событиям просмотра

Для задания режима обновления по событиям просмотра карты пользователем предназначены элементы <viewRefreshMode> и <viewRefreshTime>.

Вот, например, как можно обновить наложение спустя 2 секунды после того, как пользователь прекратит «дергать» Землю:

```
<Link>
  <href>http://localhost:3000/AugustineWebcam.kml</href>
  <viewRefreshMode>onStop</viewRefreshMode>
  <viewRefreshTime>2.0</viewRefreshTime>
<Link>
```

Элемент <viewRefreshMode> со значением `onStop` и заданным в элементе <viewRefreshTime> периодом обновления хорош с точки зрения производительности, поскольку сервер обновляет сетевую ссылку только после того, как пользователь прекратит перемещения по глобусу и сделает паузу, чтобы присмотреться к конкретному участку. В этом режиме путешествие пользователя не будет прерываться загрузкой файла.

Еще один пример обновления сетевой ссылки по событию просмотра приведен в файле *Pizza.kml*.

Более сложный пример: модель ветрового переноса вулканического пепла

Это реальное приложение, созданное Вулканической лабораторией Аляски (AVO), в котором применяются разнообразные возможности KML. Совместно с региональными отделениями AVO ведет мониторинг вулканической деятельности в Северной части Тихоокеанского побережья, включая Камчатку, Алеутские острова, материковую Аляску и северо-западную часть США. AVO – это совместный проект Геологической службы США (USGS), Геофизического института Университета Аляски в Фербэнксе (UAFGI) и Аляскинского отделения геологической и геофизической служб (ADGCS).

В примере используются сетевые ссылки для получения из обсерватории данных, которые обрабатываются и обновляются каждые шесть часов. Вулканологи Питер Уэбли и Джон Бэйли разработали компьютерную программу, любовно назвав ее «Puff» («Дуновение»), которая получает данные о направлении и скорости ветра и о частичках пепла, образовавшихся в результате извержений в каждом месте наблюдения, обрабатывает их в соответствии со сложной математической моделью и выводит информацию в виде трехмерной модели пеплового облака. Эту модель можно загрузить в Google Earth (рис. 6.8).

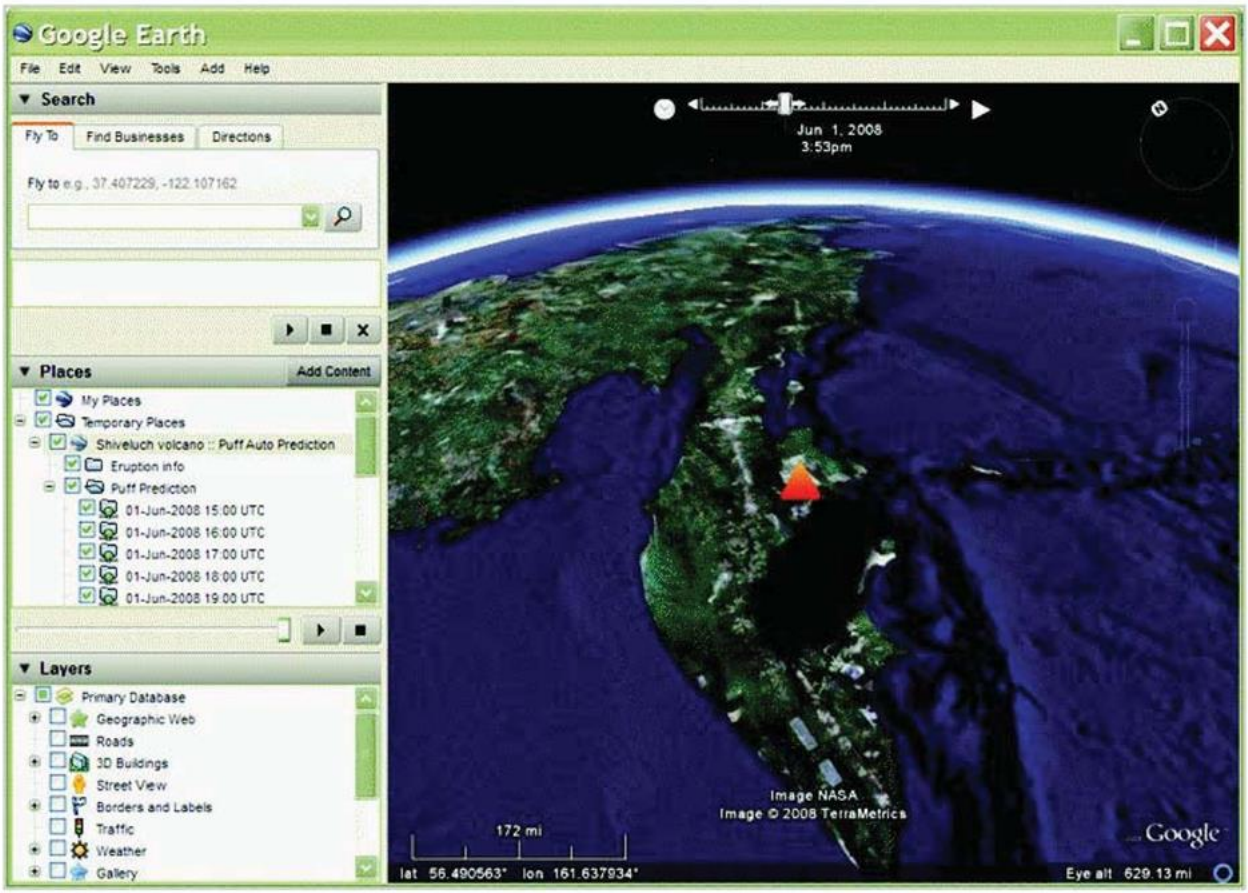


Рис. 6.8. Программа моделирования вулканического пепла Puff объединяет данные об извержениях вулканов с данными о метеорологических условиях и прогнозирует распространение частиц пепла. Эта информация очень важна для управления воздушным транспортом и мониторинга окружающей среды в целом (публикуется с разрешения Университета Аляски в Фербэнксе)

Программа периодически фиксирует данные о частичках пепла и метеоусловиях и выводит мгновенный снимок пеплового шлея в каждой точке наблюдения. У каждого снимка имеется временная метка, которая служит для Google Earth указанием «воспроизводить» файлы последовательно, как при демонстрации мультфильма. (Это лишь предварительное описание того, как используется элемент TimeSpan, подробности см. в главе 7.) Если вы хотите

посмотреть пример целиком, зайдите на сайт Вулканической лаборатории Аляски (http://puff.images.alaska.edu).

В пример, иллюстрирующий программу Puff, входит один KML-файл с сетевыми ссылками, по которому обновляемые данные скачиваются с сервера AVO. Ниже приведен фрагмент этого файла, содержащий элемент <NetworkLink> (для простоты элементы со стилем и описанием опущены).

PuffModel.kml

```
<?xml version="1.0" encoding="utf-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Document>
    <name>Вулкан Шивелуч :: прогноз программы Puff</name>
    <visibility>0</visibility>
    <LookAt>
      <longitude>161.36</longitude>
      <latitude>56.653</latitude>
      <altitude>3283</altitude>
      <heading>0</heading>
      <tilt>0</tilt>
      <range>250000</range>
      <altitudeMode>absolute</altitudeMode>
    </LookAt>
    <Folder>
      <name>Данные об извержении</name>
      <visibility>0</visibility>
      <Placemark>
        <description>...</description>
        <styleUrl>#PuffBalloon</styleUrl>
        <Style>
          <IconStyle>
            <Icon>
              <href>http://puff.images.alaska.edu/images/volcano2.png</href>
            </Icon>
          </IconStyle>
        </Style>
        <Point>
          <altitudeMode>absolute</altitudeMode>
          <coordinates>161.36,56.653,3283</coordinates>
        </Point>
      </Placemark>
    </Folder>
    <Folder>
      <name>Прогноз Puff</name>
      <NetworkLink>
        <name>12-Jan-2008 22:00 UTC</name>
        <visibility>0</visibility>
        <TimeSpan>
          <begin>2008-01-12T22:00:00Z</begin>
          <end>2008-01-12T23:00:00Z</end>
        </TimeSpan>
        <Link>
          <href>http://puff.images.alaska.edu/watch/Shiveluch/latest/avn-gfs/
          200811062200_kmz.kmz</href>
        </Link>
      </NetworkLink>
      <NetworkLink>
        <name>12-Jan-2008 23:00 UTC</name>
        <visibility>0</visibility>
        <TimeSpan>
          <begin>2008-01-12T11:00:00Z</begin>
          <end>2008-01-12T12:00:00Z</end>
        </TimeSpan>
        <Link>
          <href>http://puff.images.alaska.edu/watch/Shiveluch/latest/avn-gfs/
```

```
          200811062300_kmz.kmz</href>
        </Link>
      </NetworkLink>
.
.
.
    </Folder>
  </Document>
</kml>
```

KMZ-архив, упоминаемый в элементе <Link> в файле *PuffModel.kml*, содержит два файла: *doc.kml* (основной KML-файл) и *ash.tif*, в котором хранится растровое изображение, имитирующее пепловое облако. В реальности оно составлено из тысяч меток <Placemark>, каждая со своим значком, которые моделируют крохотный участок облака.

doc.kml (из архива 200801121100_kmz.kmz)

```
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Document>
    <name>Ash_Particles_for 200801121000</name>
    <styleUrl>#check-hide-children</styleUrl>
    <Style id="PuffBalloon">
      <BalloonStyle>
        <text>
          <![CDATA[$[description]]]>
        </text>
      </BalloonStyle>
    </Style>
    <Style id="check-hide-children">
      <ListStyle>
        <listItemType>checkHideChildren</listItemType>
      </ListStyle>
    </Style>
    <styleUrl>#check-hide-children</styleUrl>
    <Placemark>
      <styleUrl>#PuffBalloon</styleUrl>
      <description>
        .
        .
        .
      </description>
      <Style>
        <IconStyle>
          <Icon>
            <href>ash.tif</href>
          </Icon>
        </IconStyle>
      </Style>
      <Point>
        <altitudeMode>absolute</altitudeMode>
        <coordinates>160.701477,56.811301,2565.582002</coordinates>
      </Point>
    </Placemark>
    . <!-- и еще тысячи аналогичных элементов Placemark ->
    .
    .
  </Document>
</kml>
```

Вы можете скачать полный пример из Сети или зайти на сайт AVO и посмотреть на текущий трехмерный прогноз пеплового облака почти в реальном масштабе времени.

Отправка информации от клиента серверу

В этом разделе мы рассмотрим оставшиеся возможности элемента <Link>. Обсуждаемые здесь элементы используются, когда элемент <href>, вложенный в <Link>, указывает на удаленный файл, загружаемый по протоколу HTTP. К локальным файлам они неприменимы.

Для простоты изложения мы еще раз приведем синтаксис элемента <Link> (те части, которые описываются в этом разделе, выделены полужирным шрифтом):

```
<Link id="ID">
  <href>...</href>
  <refreshMode>onChange</refreshMode>
  <refreshInterval>4.0</refreshInterval>
  <viewRefreshMode>never</viewRefreshMode>
  <viewRefreshTime>4.0</viewRefreshTime>
  <viewBoundScale>1.0</viewBoundScale>
  <viewFormat>BBOX=
    [bboxWest],[bboxSouth],[bboxEast],[bboxNorth]
  </viewFormat>
  <httpQuery>...</httpQuery>
</Link>
```

Отправка информации о текущем виде

Клиент Google Earth использует элементы <viewFormat> и <viewBoundScale> для отправки серверу информации о текущей точке обзора. Зная это, CGI-сценарий на стороне сервера может эффективно сформировать ответ, соответствующий текущему состоянию просмотра.

Формат строки запроса

Посылаемый серверу URL загружаемого по ссылке файла состоит из трех частей:

- *href* (гипертекстовая ссылка), определяющая, какой файл загружать;
- форматная строка, формируемая из (а) параметров ограничивающего прямоугольника для текущего вида или (б) параметров, заданных вами в элементе <viewFormat>, описывающем текущий вид;
- вторая форматная строка, которая задается в элементе <httpQuery>.

Если в элементе <href> задан локальный файл, то <viewFormat> и <httpQuery> игнорируются. Пример элемента <NetworkLink>, посылающего строку запроса сценарию на сервере, см. в разделе «Обновление по событиям просмотра» ниже.

Информация об ограничивающем прямоугольнике (и не только)

По умолчанию клиент Google Earth добавляет к URL, заданному в элементе <href>, такие четыре параметра:

```
BBOX=[bboxWest],[bboxSouth],[bboxEast],[bboxNorth]
```

Они представляют западную, южную, восточную и северную стороны воображаемого прямоугольника, описанного вокруг *текущей области просмотра*. Он

называется *ограничивающим прямоугольником*, а формат соответствует общеупотребительной спецификации Web Map Service (WMS).

Если элемент <viewFormat> пуст, то серверу не посылается никакой информации об ограничивающем прямоугольнике.

Можно включить в строку запроса собственный набор параметров, описывающих область просмотра. В этом случае заданная вами строка будет использоваться *вместо* информации об ограничивающем прямоугольнике. Если вы хотите включить и ее тоже, то должны самостоятельно добавить указанные выше параметры, помимо своих собственных.

Элемент <viewBoundScale> позволяет масштабировать параметры ограничивающего прямоугольника перед отправкой их серверу. Значение, меньшее 1, означает, что нужно обработать область, меньшую текущей области просмотра, а значение, большее 1, – что нужно расширить границы области.

Дополнительные параметры в элементе <viewFormat>

В форматной строке можно использовать описанные ниже параметры. Имя параметра (в квадратных скобках) следует включить в состав значения элемента <viewFormat>, вложенного в <Link>. Когда программа Google Earth запрашивает у сервера ссылку, она подставляет вместо параметра соответствующее ему текущее значение.

[lookLon], [lookLat]

Долгота и широта точки, рассматриваемой из <LookAt>.

[lookatRange], [lookatTilt], [lookatHeading]

Значения, используемые в элементе <LookAt> (см. синтаксис <LookAt> в приложении А).

[lookatTerrainLon], [lookatTerrainLat], [lookatTerrainAlt]

Координаты в градусах/метрах точки обзора <LookAt> на местности.

[cameraLon], [cameraLat], [cameraAlt]

Координаты в градусах/метрах точки, в которой размещена камера <Camera>.

[horizFov], [vertFov]

Горизонтальное и вертикальное поле зрения камеры.

[horizPixels], [vertPixels]

Размер трехмерной панорамы Google Earth в пикселях.

[terrainEnabled]

Сообщает, показывает ли Google Earth в данный момент рельеф.

Отправка серверу дополнительных данных с помощью <httpQuery>

Элемент <httpQuery> используется для отправки серверу дополнительных данных о клиенте Google Earth. Если включить любой из указанных ниже параметров, то Google Earth подставит в запрос соответствующее значение.

[clientVersion]

Версия геобраузера, отправившего запрос.

[kmlVersion]

Версия KML, поддерживаемая геобраузером, отправившим запрос.

[clientName]

Название геобраузера, отправившего запрос.

[language]

Язык, используемый в данный момент геобраузером.

Например, желая сообщить серверу размеры окна панорамы Google Earth, текущее состояние флажка **Рельеф**, название и версию клиентского приложения, а также текущий установленный язык, вы должны включить в элемент <Link>, вложенный в <NetworkLink>, следующие параметры:

```
<Link>
  <href>http://localhost:3000/AugustineWebcamRevised.kml</href>
  <viewFormat>[horizPixels],[vertPixels],[terrainEnabled]</viewFormat>
  <httpQuery>[clientVersion],[clientName],[language]<httpQuery>
</Link>
```

Пример обновления по событиям просмотра

В этом примере, созданном Мано Марксом, метки генерируются на лету, исходя из текущего просматриваемого пользователем фрагмента. Написанный на PHP сценарий опрашивает базу данных MySQL, которая содержит список пиццерий в США. В результате создаются метки для всех пиццерий в радиусе 400 миль от центра просматриваемого фрагмента. Файл *Pizzas.kml* находится на машине клиента, а сценарий *genxml.php* – на сервере, там же, где база данных.

В файле *Pizzas.kml* имеется элемент <NetworkLink>, в котором параметр <viewRefreshMode> равен `onStop`. Если пользователь в течение 4 секунд не изменяет панораму, то Google Earth отправляет серверу информацию о текущем ограничивающем прямоугольнике в составе URL, который задан в <NetworkLink>. В элементе <Link> задано также значение для <httpQuery> (`radius=400`). Оно добавляется в строку запроса, отправляемую серверу, как описано в предыдущем разделе. Например, если пользователь задержался над Чикаго, то Google Earth пошлет серверу такую строку запроса:

```
http://localhost:8081/genkml.php?+BBOX=[-87.864613],[41.659033],
[-87.281816],[42.095123];?radius=400
```

Pizzas.kml

```
<?xml version="1.0" encoding="utf-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <NetworkLink>
    <name>Пример обновления по событию просмотра</name>
    <open>1</open>
    <Link>
      <href>http://localhost:8081/genxml.php?</href>
      <viewRefreshMode>onStop</viewRefreshMode>
      <httpQuery>radius=400</httpQuery>
    </Link>
  </NetworkLink>
</kml>
```

PHP-сценарий *genxml.php* опрашивает базу данных на сервере и возвращает записи обо всех пиццериях в радиусе 400 миль от центра прямоугольника, ограничивающего текущий просматриваемый фрагмент.

genxml.php

```
<?php
 $username = "joesmith";
 $password = "mypassword";
 $database = "testdb";
 $server = "dbtest.com";
error_reporting(0);
function parseLocation(){
$radius = $_GET["radius"];
if (!$radius) $radius = 200;
$bbox = $_GET["BBOX"];
$bbox = split(",",$bbox);
$west = $bbox[0];
$south = $bbox[1];
$east = $bbox[2];
$north = $bbox[3];
$center_lat = (($north - $south)/2) + $south;
$center_lng = (($east - $west)/2) + $east;
$location = array("center_lat"=>$center_lat,
"center_lng"=>$center_lng,"radius"=>$radius);
return $location;
}
$location = parseLocation();
$center_lat = $location["center_lat"];
$center_lng = $location["center_lng"];
$radius = $location["radius"];
// Начинаемся XML-файл, создать родительский узел
$dom = new DOMDocument("1.0");
$node = $dom->createElement("kml");
$kmlnode = $dom->appendChild($node);
$foldernode = $dom->createElement("Folder");
$parnode = $kmlnode->appendChild($foldernode);
$opennode = $dom->createElement("open","1");
$parnode->appendChild($opennode);
// Открываем соединение с MySQL-сервером
$connection=mysql_connect($server, $username, $password);
if (!$connection) {
die("Не удалось соединиться : " . mysql_error());
}
// Устанавливаем активную базу данных MySQL
$db_selected = mysql_select_db($database, $connection);
if (!$db_selected) {
die ("Не могу установить БД : " . mysql_error());
}
// Ищем строки в таблице markers
$query = "SELECT address, name, lat, lng, ( 3959 * acos( cos("
."radians(".$center_lat.") ) * cos( radians( lat ) ) * cos( radians( lng )"
."- radians(" . $center_lng . ") ) + sin( radians(".$center_lat.") ) *"
."sin( radians( lat ) ) ) ) AS distance FROM markers HAVING distance < "
.$radius. " ORDER BY distance LIMIT 0 , 20";
$result = mysql_query($query);
if (!$result) {
die("Недопустимый запрос: " . mysql_error());
}
//header("Content-type: application/vnd.google-earth.kml+xml");
// Перебираем найденные строки, добавляя для каждой узлы в XML-документ
while ($row = @mysql_fetch_assoc($result)){
$node = $dom->createElement("Placemark");
$placenode = $parnode->appendChild($node);
$namenode = $dom->createElement("name",htmlentities ($row["name"]));
$placenode->appendChild($namenode);
```

```
$descriptioncdata = $dom->createCDATASection("<b>Address:</b> " .
$row["address"] ."<br/><b>Distance:</b> " . $row["distance"]);
$descriptionnode=$dom->createElement("description");
$descriptionnode->appendChild($descriptioncdata);
$placenode->appendChild($descriptionnode);
$coor = $row["lng"] . "," . $row["lat"];
$pointnode = $dom->createElement("Point");
$placenode->appendChild($pointnode);
$coornode = $dom->createElement("coordinates", $coor);
$pointnode->appendChild($coornode);
}
echo $dom->saveXML();
?>
```

На рис. 6.9 изображен пример меток пиццерий, отображаемых, когда пользователь задержался в районе Чикаго.

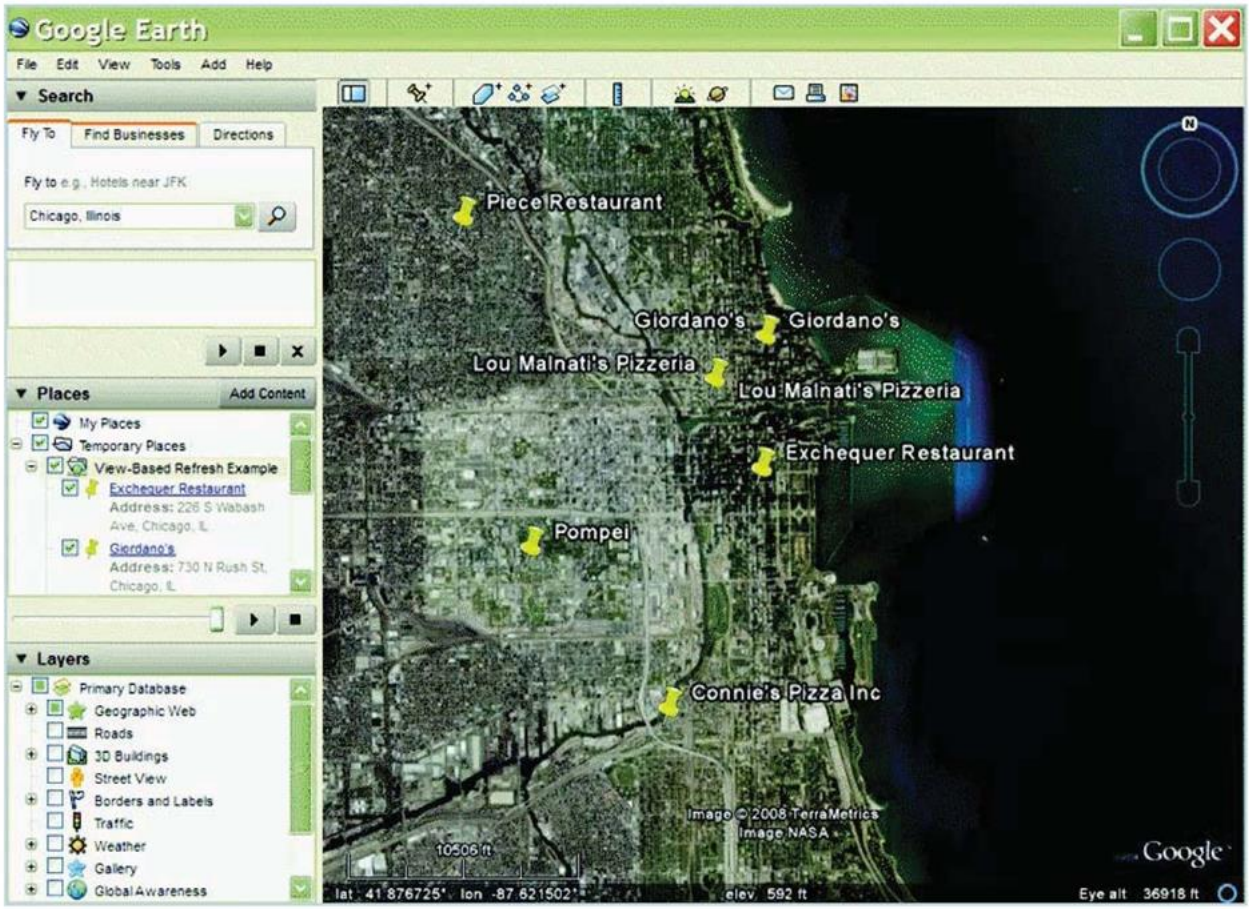


Рис. 6.9. KML-файл создается динамически по результатам запроса к базе данных о пиццериях, расположенных поблизости от текущей точки обзора; см. *Pizzas.kml* (пример написан Мано Марксом)

Контроллеры сетевых ссылок

Элемент <NetworkLinkControl> находится в KML-файле, загружаемом по ссылке с сервера. Он решает следующие три задачи:

- определяет или переопределяет, что происходит при загрузке по сетевой ссылке;
- воздействует на данные, возвращаемые сетевой ссылкой;
- воздействует на данные, загруженные по сетевой ссылке ранее и сейчас находящиеся в памяти Google Earth. (Об использовании контроллера сетевой ссылки для *обновления* данных, загруженных в Google Earth ранее, см. главу 7.)

На рис. 6.10 представлена схема обработки KML-файла, загружаемого по сетевой ссылке.

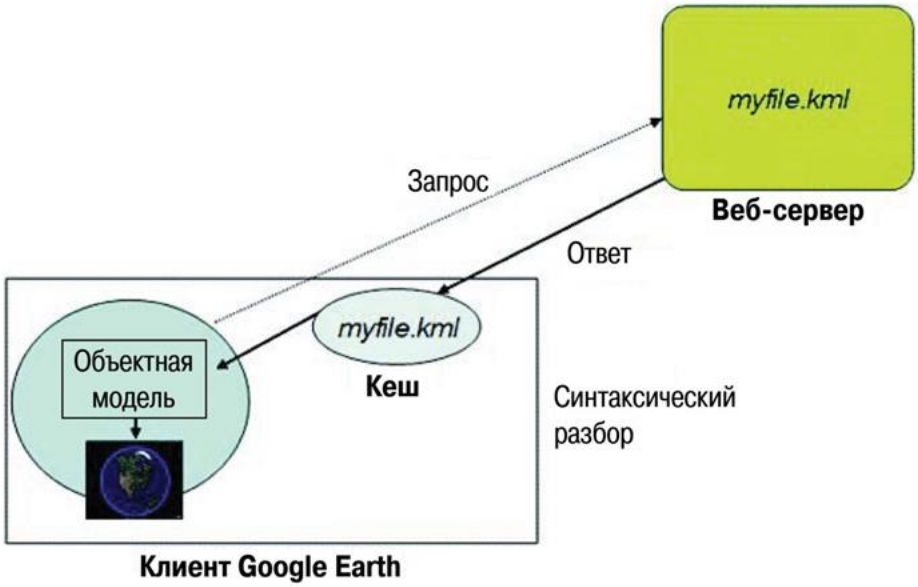


Рис. 6.10. Файл, загруженный по сетевой ссылке, обрабатывается в соответствии с условиями, заданными в элементе <NetworkLinkControl> (если таковой присутствует)

1. KML-файл загружается с сервера. Если полученный файл содержит элемент <NetworkLinkControl>, то ход процедуры загрузки управляется контроллерами, заданными в этом элементе.
2. Если файл находится в Сети, то он отправляется по протоколу HTTP и копируется в локальный системный кеш (временное хранилище на диске).
3. Далее клиент читает файл, разбирает его и загружает в оперативную память в виде *объектной модели*, которую уже можно отобразить на экране, то есть передать в буфер кадров (часть графической аппаратуры).
4. Компьютер рисует графическое изображение в области трехмерной панорамы Google Earth, визуализируя загруженный файл.

Синтаксис элемента <NetworkLinkControl>

Элемент <NetworkLinkControl> является непосредственным потомком элемента <kml>. Таких потомков у элемента <kml> может быть не более одного.

Ниже описывается синтаксис элемента <NetworkLinkControl>.

```
<NetworkLinkControl>
  <minRefreshPeriod>0.0</minRefreshPeriod>
  <maxSessionLength>-1.0</maxSessionLength>
  <cookie>...</cookie>
  <message>...</message>
  <linkName>...</linkName>
  <linkDescription>...</linkDescription>
  <linkSnippet maxLines="2">...</linkSnippet>
  <expires>...</expires>
  <Update>...</Update>  <!-- см. главу 7 ->
  <AbstractView>...</AbstractView><!-- <LookAt> или <Camera> ->
</NetworkLinkControl>
```

<minRefreshPeriod>

(по умолчанию = 0.0) Задает минимальное время между последовательными операциями загрузки файла. Значение задается в секундах (оно имеет тип `double`, то есть может быть дробным). Этот элемент позволяет серверу задерживать отправку файла, выбирая такой темп обновления, чтобы общая нагрузка была приемлемой. Значение, заданное в этом элементе, отменяет параметры обновления, указанные в элементе <Link>, вложенном в <NetworkLink> в KML-файле.

<maxSessionLength>

Задает максимальное время в секундах, в течение которого сохраняется соединение между клиентом и сервером. Принимаемое по умолчанию значение –1 означает, что соединение никогда не разрывается принудительно.

<cookie>

Сервер формирует строку <cookie>, которую клиент должен добавить в строку запроса URL при следующем обновлении сетевой ссылки. Эти данные можно использовать в серверном сценарии для уточнения алгоритма обработки, в частности для опроса номера версии и условной отправки файла. Точная последовательность действий выглядит так:

1. Сервер генерирует элемент <NetworkLinkControl> и формирует значение <cookie> (кук), например `"count=0"`.
2. Элемент <NetworkLink> на стороне клиент запоминает кук.
3. При очередном обращении к серверу для обновления сетевой ссылки клиент добавляет в конец URL строку `"count=0"`.
4. Сервер разбирает URL, пришедший с запросом, и ищет в нем кук.
5. Сервер возвращается к шагу 1 и, возможно, увеличивает счетчик на 1, то есть отправляет в качестве нового кука строку `"count=1"`.

Кук не обязательно должен иметь вид *имя=значение*, но такое соглашение понимают стандартные модули, с помощью которых реализуются CGI-сценарии. Отметим, что KML-кук не имеет ничего общего с HTTP-куком.

<message>

Сервер формирует строку <message>, которая отображается в диалоговом окне на экране пользователя в Google Earth при первой загрузке сетевой ссылки или при ее изменении контроллером сетевой ссылки (на стороне сервера). Отметим, что окно появляется не при каждом обновлении файла, чтобы не надоедать пользователю.

<linkName>

Сервер задает элемент <linkName>, который в Google Earth визуализируется как имя сетевой ссылки на панели списка. Он переопределяет элемент <name>, если таковой был задан внутри элемента <NetworkLink> в файле на стороне клиента.

<linkDescription>

Сервер задает элемент <linkDescription>, который в Google Earth визуализируется как описание сетевой ссылки на панели списка. Он переопределяет элемент <description>, если таковой был задан внутри элемента <NetworkLink> в файле на стороне клиента.

<linkSnippet maxLines=»2">

Сервер задает элемент <linkSnippet>, который в Google Earth визуализируется как краткое описание сетевой ссылки на панели списка. Он переопределяет элемент <Snippet>, если таковой был задан внутри элемента <NetworkLink> в файле на стороне клиента. Атрибут `maxLines` – это максимальное количество отображаемых строк. Он переопределяет значение `maxLines`, если таковое было задано в элементе <Snippet> внутри элемента <NetworkLink> на стороне клиента.

<expires>

Задает дату и время, когда ссылка должна быть обновлена. Этот параметр принимается во внимание, только если параметр <refreshMode> внутри <Link> равен `onExpire` (см. раздел «Обновление по времени или при обнаружении изменений» выше).

<Update>

<targetHref>
<Change>
<Create>
<Delete>

Этот элемент позволяет задать произвольное число элементов Change, Create и Delete для ранее загруженного по сетевой ссылке файла. О том, как специфицируются обновления, см. главу 7.

<AbstractView>

Задает элемент <Camera> или <LookAt>, являющийся точкой обзора для сетевой ссылки.

Пример переопределения поведения сервером

В этом примере показано, как сервер может переопределить параметры <name> и <description> сетевой ссылки. Также демонстрируется задание сообщения <message>, которое пользователь увидит при первой загрузке KML-файла. Результаты загрузки файла по этой сетевой ссылке представлены на рис. 6.11. Какое имя и описание вы видите – заданное в самой сетевой ссылке или в ее контроллере?

Поэкспериментируйте, изменяя различные в поля в элементах <NetworkLink> и <NetworkLinkControl> и наблюдая за тем, как контроллер переопределяет

Рис. 6.11. В этом примере показано, как контроллер сетевой ссылки может переопределить значения, заданные в самой сетевой ссылке (см. файлы *BasicNetworkLink.kml* и *AugustineWebcamRevised.kml*)

значения, заданные в исходной сетевой ссылке. Затем попробуйте заменить сообщение в контроллере. Если вы установили веб-сервер, то поместите файл *AugustineWebcamRevised.kml* на сервер, измените значение <href> в сетевой ссылке и попробуйте получить этот файл по Сети.

BasicNetworkLink.kml

```
<?xml version="1.0" encoding="utf-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Document>
    <name>Volcanoes Webcam</name>
    <NetworkLink>
      <name>Original Name on Client Side</name>
      <description>
      <![CDATA[
      This image is very dark in the winter!
      ]]>
      </description>
      <Link>
        <href>AugustineWebcamRevised.kml</href>
      </Link>
    </NetworkLink>
  </Document>
</kml>
```

AugustineWebcamRevised.kml

```
<?xml version="1.0" encoding="utf-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <NetworkLinkControl>
    <message>
      <![CDATA[
      This image is updated every 6 hours!
      ]]>
    </message>
```

```
    <linkName>Network Link Control Example</linkName>
    <linkDescription>
      <![CDATA[
      Can you see anything?
      ]]>
    </linkDescription>
  </NetworkLinkControl>
  <Document>
    <name>Augustine Webcam</name>
    <LookAt>
      <longitude>-133.768512314146</longitude>
      <latitude>56.447578192874</latitude>
      <altitude>0</altitude>
      <heading>-10.5952483593619</heading>
      <tilt>0</tilt>
      <range>7382344.56134118</range>
      <altitudeMode>relativeToGround</altitudeMode>
    </LookAt>
    <ScreenOverlay>
      <name>Webcam Image</name>
      <Icon>
        <href>http://www.avo.alaska.edu/webcam/augustine.jpg</href>
      </Icon>
      <overlayXY x="0" xunits="fraction" y="1" yunits="fraction"/>
      <screenXY x="0" xunits="fraction" y="1" yunits="fraction"/>
      <size x="0.5" xunits="fraction" y="0.5" yunits="fraction"/>
    </ScreenOverlay>
    <Placemark>
      <name>Augustine Volcano</name>
      <Point>
        <coordinates>-153.456174,59.368623,0</coordinates>
      </Point>
    </Placemark>
  </Document>
</kml>
```

Что дальше?

Вы познакомились с основами передачи информации с помощью сетевых ссылок и тем, как контроллер сетевой ссылки может воздействовать на процесс ее загрузки. В главе 7 мы рассмотрим механизм управления сетевыми ссылками более глубоко и покажем, как с помощью элемента <Update> можно создавать, изменять и удалять элементы в KML-файлах, ранее загруженных по ссылке и теперь находящихся в памяти геобраузера. Кроме того, мы объясним, как с помощью элементов задания времени <TimeSpan> и <TimeStamp> можно создавать динамические эффекты в KML. Наличие этих элементов в KML-файле приводит к загрузке деталей в заданной последовательности, что позволяет анимировать модели, наложения и значки иконок в Google Earth.

Глава 7. Динамический KML

Прочитав эту главу, вы сможете:

- ✓ объяснить, зачем нужны идентификаторы ID;
- ✓ воспользоваться контроллером сетевой ссылки для *изменения* значений в ранее загруженном и находящемся в памяти KML-файле;
- ✓ воспользоваться контроллером сетевой ссылки для *удаления* элемента из ранее загруженного KML-файла;
- ✓ воспользоваться контроллером сетевой ссылки для *добавления* элементов в ранее загруженный KML-файл;
- ✓ анимировать метку, так чтобы она перемещалась вдоль заданного пути;
- ✓ создать последовательность меток или рельефных наложений, управляемую ползунком, встроенным в Google Earth.

Предварительный обзор

В этой главе рассматриваются несколько приемов создания KML-объектов, которые перемещаются или изменяются заданным образом. Первый прием заключается в использовании элемента *Update* в контроллере сетевой ссылки для *изменения*, *удаления* и *добавления* элементов в ранее загруженный по ссылке файл. В этой главе применяется материал из главы 6, где были рассмотрены основы работы сетевых ссылок и их контроллеров.

Второй способ создания динамических эффектов в KML основан на использовании временных примитивов: элементов *TimeStamp* и *TimeSpan*. Временной примитив является потомком элемента Feature (<Placemark>, Overlay, Container и <NetworkLink>) и служит для того, чтобы ограничить время отображения детали на экране каким-то одним моментом или промежутком. Если KML-файл содержит детали с временными примитивами, то геобраузер автоматически выводит ползунок, соответствующий указанным в файле интервалам времени.

В этой главе вводится новое важное понятие – *идентификаторы* элементов, обновляемых контроллером сетевой ссылки. За разделом «Основные понятия» следуют два больших раздела: в первом подробно описывается работа с элементом Update, а во втором – временные примитивы.

Основные понятия

По большей части понятия, которые потребуются нам в этой главе, были введены в главе 6 «Сетевые ссылки». Здесь мы приведем дополнительную информацию о задании и использовании идентификаторов в KML.

Идентификаторы

Любой элемент KML, производный от <*Object*>, может иметь идентификатор (атрибут `id`). Они применяются, например, для уникальной идентификации *стилей*, чтобы на стиль можно было ссылаться из нескольких деталей (см. главу 4). Еще одно применение идентификаторов – однозначное задание элемента, модифицируемого операцией *Update*, о чем пойдет речь в этой главе. Третье применение идентификаторов в KML – *фрагменты URI* – будет описано ниже. В главе 8 «Большие наборы данных» мы расскажем о четвертом применении – идентификации элемента <Schema> для объявления нестандартных типов данных.

Идентификатор определяется внутри открывающего тега элемента. Вот, например, как задается идентификатор элемента <BalloonStyle> (см. главу 4):

```
<Style id="grayBlueScheme">
  <BalloonStyle>
    <bgColor>ffffabb2</bgColor>
    <textColor>ff8d8d8d</textColor>
    <displayMode>default</displayMode>
  </BalloonStyle>
</Style>
```

В элементах <Placemark> в том же файле можно сослаться на этот стиль по его идентификатору, предпослав последнему знак #:

```
<Placemark>
  <name>Сконфигурированная метка</name>
  <description>У этого информационного окна серо-синий фон</description>
  <styleUrl>#grayBlueScheme</styleUrl>
  <Point>
    <coordinates>-80.210836,34.679252,0</coordinates>
  </Point>
</Placemark>
```

В качестве идентификаторов лучше выбирать осмысленные строки, что-то сообщающие о самом элементе. Технически идентификатор в языке KML – это строка, следующая спецификации `NCName` в стандарте XML. Идентификатор должен начинаться с буквы или одного из символов: . (точка), - (дефис) и _ (подчеркивание). Идентификатор не может начинаться с цифры, но может содержать цифры внутри. Он не может содержать пробелов. Для записи идентификаторов применяется набор символов Unicode (в который входят, в частности, иероглифы).

Фрагменты URI

В KML элементы <href> и <styleUrl> могут содержать значения типа `anyURI`. URI (Uniform Resource Identifier – унифицированный идентификатор ресурса) – это более точный термин, описывающий то, что часто называют URL (Uniform Resource Locator – унифицированный локатор ресурса). URI описывает ресурс (KML-файл, HTML-файл, изображение, видеоролик и т. д.) в Интернете и состоит из нескольких частей:

Схема	Хост	Путь	Фрагмент	Строка запроса
http:// file:/	myServer.com	/greece/cities myDirectory/Jan/	#athens	?a=24&c=35

Со *схемой*, *хостом* и *путем* вы уже знакомы. *Фрагмент* применяется в анкерах деталей (глава 2) для обозначения конкретного объекта в KML-файле по его идентификатору (которому предшествует знак #). Когда пользователь щелкает по ссылке <href>, включающей фрагмент URI, геобраузер по умолчанию «перелетает» к детали, идентификатор которой соответствует фрагменту. Если для этой детали задан элемент <LookAt> или <Camera>, то деталь рассматривается с указанной точки обзора.

Например, включив следующий URI в описание всплывающего окна, мы создадим гиперссылку, щелчок по которой заставляет браузер открыть указанный KML-файл и «перелететь» к детали с идентификатором «myHomeTown».

```
<description>
  <![CDATA[
    <a href="http://myServer.com/biographies.kml#myHomeTown">
    Где все начиналось</a>
</description>
```

Внутри KML-файла фрагмент URI можно использовать сам по себе (например, `#historicMarker`) для ссылки на объекты в том же самом файле.

О строках запроса см. главу 6 «Сетевые ссылки».

Обновление в KML

Механизм обновления в языке KML полезен в случае, когда вы уже передали по сетевой ссылке большой объем данных и хотите произвести сравнительно небольшое изменение в ранее загруженном файле. Альтернативой элементу <Update> служит полная перезагрузка файла по времени или по факту изменения (см. главу 6). Тогда весь файл (включая модели, текстуры и изображения) удаляется из памяти, скачивается и снова загружается в память. Напротив, элемент <Update> позволяет изменить только указанные элементы. Следовательно, несмотря на то что затраты на подготовку инфраструктуры обновления несколько выше, для больших наборов данных это может заметно повысить производительность.

Существуют три вида обновлений:

- создание (Create);
- изменение (Change);
- удаление (Delete).

В одном элементе <NetworkLinkControl> может быть сколько угодно элементов каждого типа, причем обновления производятся в том порядке, в котором встречаются эти элементы. Обновляемый KML-файл должен быть предварительно загружен по сетевой ссылке, причем в целях безопасности файл, указанный в элементе <targetHref> контроллера этой сетевой ссылки, должен находиться на том же сервере, где и сам контроллер.

Общее описание механизма обновления

Прежде чем переходить к синтаксису элемента <Update>, полезно получить общее представление о том, как настраивается механизм обновления. Здесь имеет место своего рода косвенная адресация, поскольку одна сетевая ссылка на стороне

клиента запрашивает исходный KML-файл с сервера, тогда как другая ссылка, тоже на стороне клиента, запрашивает обновления KML. Вся схема изображена на рис. 7.1 и 7.2.

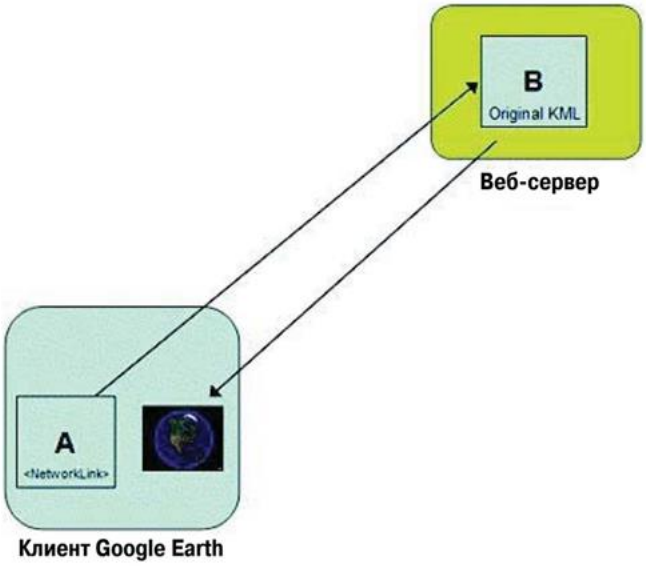


Рис. 7.1. Концептуально в обновлении участвуют четыре KML-файла. На шаге 1 по сетевой ссылке A загружается KML-файл B

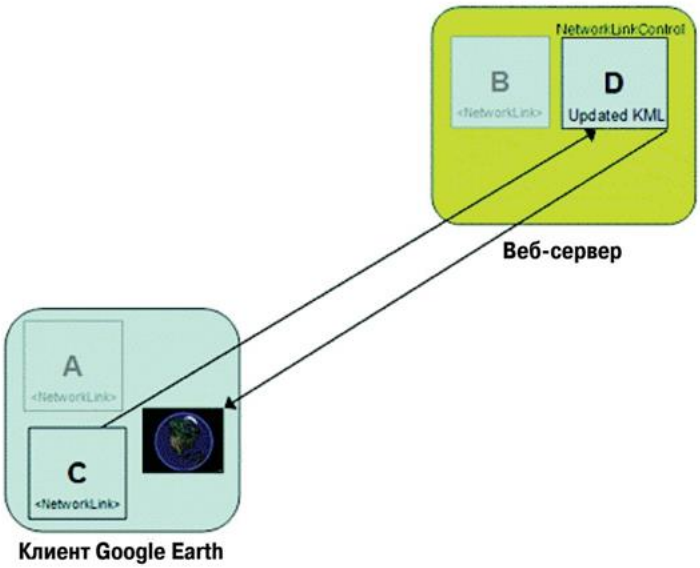


Рис. 7.2. На шаге 2 по сетевой ссылке C загружается KML-файл D с обновлениями

В интересах безопасности оба KML-файла загружаются по сетевым ссылкам, то есть исходный файл (B) и файл с обновлениями (D) должны находиться на одном и том же сервере. Это требование позволяет обеспечить целостность данных за счет ограничения доступа к обновляемому файлу.

Синтаксис элемента <Update>, вложенного в <NetworkLinkControl>

Ниже приведен синтаксис элемента <Update>, вложенного в <NetworkLinkControl>:

```
<NetworkLinkControl>
  .
  .
  .
  <Update>
    <targetHref>...</targetHref>
    <Change>...</Change>
    <Create>...</Create>
    <Delete>...</Delete>
  </Update>
  .
  .
  .
</NetworkLinkControl>
```

[Update]

У элемента <Update> может быть произвольное количество потомков типа <Change>, <Create> и <Delete>, причем выполняются они в порядке следования.

<targetHref>

Задает подлежащий обновлению KML-файл. Этот файл должен находиться на том же сервере, что и файл с обновлениями.

<Change>

Модифицирует значения в элементе KML-файла, ранее загруженного по сетевой ссылке. Потомком элемента <Change> может быть любой элемент, производный от Object. Этому элементу должен соответствовать какой-то элемент в исходном KML-файле, которому приписан идентификатор `id`. Идентификатор обновляемого элемента указывается в потомке <Change> с помощью атрибута `targetId`. В модифицируемом элементе заменяются только значения, перечисленные внутри <Change>, другие остаются без изменения. Если элемент <Change> применяется к набору координат, то текущие координаты заменяются новыми.

<Create>

Добавляет элемент в KML-файл, ранее загруженный по сетевой ссылке. Потомком элемента <Create> *должен* быть элемент, производный от Container (то есть <Document> или <Folder>). Этому элементу должен соответствовать какой-то элемент в исходном KML-файле, которому приписан идентификатор `id`. Потомок <Create> – это подлежащий созданию элемент, у него должен быть атрибут `targeted`, равный `id` элемента в исходном файле. После выполнения обновления новый элемент становится частью исходного файла. Для последующих обновлений этого элемента следует указывать URL исходного KML-файла (файла B на рис. 7.1).

<Delete>

Удаляет элемент из KML-файла, ранее загруженного по сетевой ссылке. Потомком элемента <Delete> *должен* быть элемент, производный от Feature (<NetworkLink>, <Placemark>, <GroundOverlay>, <PhotoOverlay>, <ScreenOverlay>, <Document> или <Folder>). Этому элементу должен соответствовать какой-то элемент в исходном KML-файле, которому приписан идентификатор `id`. Потомок <Delete> – это подлежащий удалению элемент, у него должен быть атрибут `targeted`, равный `id` элемента в исходном файле.

Для любой операции обновления необходимо задать два параметра:

- `targetHref`

  Задает подлежащий обновлению файл (в приведенных выше простых примерах файл B на рис. 7.1). Этот файл должен находиться на том же сервере, что и KML-файл с обновлениями. (Отметим, что хотя `targetHref` задает имя файла, обновление применяется к *содержимому* этого файла, уже загруженному в память Google Earth. Файл на диске остается неизменным.)
- `targetId`

  Задает идентификатор `id` объекта, подлежащего обновлению. Атрибут `targetId` применяется в элементах <Change>, <Create> и <Delete>, для того чтобы связать новые данные с ранее загруженными. (В простых примерах выше элемент <Update> является частью файла D на рис. 7.2.)

Создание нового элемента с помощью <Update>

В следующем примере показано, как добавить метку в KML-файл, ранее загруженный в Google Earth по сетевой ссылке.

Пример создания элемента

В примере ниже демонстрируется применение элемента <Update> для создания объекта. В его состав входят четыре файла, соответствующие файлам, изображенным на рис. 7.1 и 7.2 (первая буква имени файла говорит о том, какому именно из этих четырех файлов он соответствует).

Этот файл, находящийся на стороне клиента, содержит сетевую ссылку для загрузки в геобраузер исходного KML-файла.

aMyPlacemark.kml (загрузка по сетевой ссылке)

```
<?xml version="1.0" encoding="utf-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <NetworkLink>
    <name>Пример создания меток</name>
    <visibility>0</visibility>
    <description>Флажок, позволяющий увидеть метку</description>
    <Link>
      <href>http://localhost:3000/bOnePlacemark.kml</href>
    </Link>
  </NetworkLink>
</kml>
```

Этот файл, находящийся на сервере, содержит исходный KML-документ, загружаемый в геобраузер.

bOnePlacemark.kml (исходный KML-файл)

```
<?xml version="1.0" encoding="utf-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Folder id="targetFolder">
    <name>Все мои потомки</name>
    <Placemark>
      <name>Моя первая метка</name>
      <visibility>1</visibility>
      <Snippet>исходное краткое описание</Snippet>
      <description>исходное описание</description>
      <Point>
        <coordinates>130.166977,-14.40344,0</coordinates>
      </Point>
    </Placemark>
  </Folder>
</kml>
```

Этот файл, находящийся на стороне клиента, содержит сетевую ссылку для загрузки в геобраузер KML-файла с обновлениями.

cCreateMore.kml (сетевая ссылка на файл с обновлениями)

```
<?xml version="1.0" encoding="utf-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <NetworkLink>
    <name>Добавить метку</name>
    <visibility>0</visibility>
    <description>Флажок, позволяющий увидеть метку</description>
    <Link>
      <href>http://localhost:3000/dUpdatePlacemarkData.kml</href>
    </Link>
  </NetworkLink>
</kml>
```

Этот файл, находящийся на сервере, содержит новую метку, которая должна быть добавлена в элемент `"targetFolder"` в исходном KML-файле (*bOnePlacemark.kml*). Значение параметра `<targeHref>` в элементе <Update> должно в точности совпадать со значением <href> в элементе <Link>, вложенном в сетевую ссылку (в файле *aMyPlacemark.kml*). Не забудьте включить полный URL файла, начинающийся с префикса *http://*.

dUpdatePlacemarkData.kml (KML-файл с обновлениями)

```
<?xml version="1.0" encoding="utf-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <NetworkLinkControl>
    <Update>
      <targetHref>http://localhost:3000/bOnePlacemark.kml</targetHref>
      <Create>
        <Folder targetId="targetFolder">
          <Placemark>
            <name>Вновь созданная метка</name>
            <visibility>1</visibility>
            <Snippet>Сидней Австралия</Snippet>
            <description>
              новое описание, новое местоположение
            </description>
            <Point>
```

```
            <coordinates>151.209522,-33.861718,0</coordinates>
          </Point>
        </Placemark>
      </Folder>
    </Create>
  </Update>
</NetworkLinkControl>
</kml>
```

Изменение элемента с помощью <Update>

В следующем примере показано, как изменить координаты, а также полное и краткое описания метки, ранее загруженной в Google Earth по сетевой ссылке.

Пример изменения элемента

Следующие четыре файла соответствуют файлам, изображенным на рис. 7.2.

Этот файл, находящийся на стороне клиента, содержит сетевую ссылку для загрузки в геобраузер исходного KML-файла.

aPlacemark.kml (загрузка по сетевой ссылке)

```
<?xml version="1.0" encoding="utf-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <NetworkLink>
    <name>Моя местовая метка</name>
    <visibility>0</visibility>
    <description>Флажок, позволяющий увидеть метку</description>
    <Link>
      <href>http://localhost:3000/bOriginalPlacemark.kml</href>
    </Link>
  </NetworkLink>
</kml>
```

Этот файл, находящийся на сервере, содержит исходный KML-документ, загружаемый в геобраузер.

bOriginalPlacemark.kml (исходный KML-файл)

```
<?xml version="1.0" encoding="utf-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Placemark id="targetPlacemark">
    <name>Исходное имя метки</name>
    <visibility>1</visibility>
    <Snippet>исходное краткое описание</Snippet>
    <description>исходное описание</description>
    <Point>
      <coordinates>-122.000,37.000</coordinates>
    </Point>
  </Placemark>
</kml>
```

Этот файл, находящийся на стороне клиента, содержит сетевую ссылку для загрузки в геобраузер KML-файла с обновлениями.

cChangeMe.kml (сетевая ссылка на файл с обновлениями)

```
<?xml version="1.0" encoding="utf-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
```

```
  <NetworkLink>
    <name>Изменить мою метку</name>
    <visibility>0</visibility>
    <description>Флажок, позволяющий увидеть метку</description>
    <Link>
      <href>http://localhost:3000/dUpdatePlacemark.kml</href>
    </Link>
  </NetworkLink>
</kml>
```

Этот файл, находящийся на сервере, содержит новое имя, полное и краткое описания, а также координаты метки (с идентификатором `"targetPlacemark"`), которая была ранее загружена в геобраузер (*bOriginalPlacemark.kml*).

dUpdatePlacemark.kml (KML-файл с обновлениями)

```
<?xml version="1.0" encoding="utf-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <NetworkLinkControl>
    <Update>
      <targetHref>http://localhost:3000/bOriginalPlacemark.kml</targetHref>
      <Change>
        <Placemark targetId="targetPlacemark">
          <name>Новая метка</name>
          <visibility>1</visibility>
          <Snippet>Пересекаем океан</Snippet>
          <description>
            <![CDATA[
              новое описание, новое местоположение
            ]]>
          </description>
          <Point>
            <coordinates>140.837,-3.26921</coordinates>
          </Point>
        </Placemark>
      </Change>
    </Update>
  </NetworkLinkControl>
</kml>
```

Удаление элемента с помощью <Update>

В следующем примере показано, как удалить метку из файла, ранее загруженного в Google Earth по сетевой ссылке.

Пример удаления элемента

Следующие четыре файла соответствуют файлам, изображенным на рис. 7.2.

Этот файл, находящийся на стороне клиента, содержит сетевую ссылку для загрузки в геобраузер исходного KML-файла.

aPlacemark.kml (загрузка по сетевой ссылке)

```
<?xml version="1.0" encoding="utf-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <NetworkLink>
    <name>Пример удаления меток</name>
    <visibility>0</visibility>
    <description>Флажок, позволяющий увидеть метку</description>
    <Link>
```

```
      <href>http://localhost:3000/bOriginalPlacemarks.kml</href>
    </Link>
  </NetworkLink>
</kml>
```

Этот файл, находящийся на сервере, содержит исходный KML-документ, загружаемый в геобраузер.

bOriginalPlacemarks.kml (исходный KML-файл)

```
<?xml version="1.0" encoding="utf-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Folder>
    <Placemark id="eastMarker">
      <name>Восточная метка</name>
      <visibility>1</visibility>
      <Point>
        <coordinates>-67.99,-54.2312</coordinates>
      </Point>
    </Placemark>
    <Placemark id="westMarker">
      <name>Западная метка</name>
      <visibility>1</visibility>
      <Point>
        <coordinates>-73.0128,-54.3714</coordinates>
      </Point>
    </Placemark>
  </Folder>
</kml>
```

Этот файл, находящийся на стороне клиента, содержит сетевую ссылку для загрузки в геобраузер KML-файла с обновлениями.

cDeleteOne.kml (сетевая ссылка на файл с обновлениями)

```
<?xml version="1.0" encoding="utf-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <NetworkLink>
    <name>Удаление метки</name>
    <visibility>0</visibility>
    <Link>
      <href>http://localhost:3000/dRemoveOnePlacemark.kml</href>
    </Link>
  </NetworkLink>
</kml>
```

Этот файл, находящийся на сервере, определяет метку (с идентификатором `"eastMarker"`), которую следует удалить из ранее загруженного в геобраузер файла (*bOriginalPlacemarks.kml*).

dRemoveOnePlacemark.kml (KML-файл с обновлениями)

```
<?xml version="1.0" encoding="utf-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <NetworkLinkControl>
    <Update>
      <targetHref>http://localhost:3000/bOriginalPlacemarks.kml</targetHref>
      <Delete>
        <Placemark targetId="eastMarker"/>
      </Delete>
    </Update>
  </NetworkLinkControl>
</kml>
```

Пример обновления с помощью сценария

В этом примере, который написал Бент Хагемарк, используется сценарий на языке Python для генерации исходного KML-файла и его обновлений. (Таким образом, один сценарий *continents.py* заменяет файлы B и D на рис. 7.2.) В сценарии элемент `<cookie>`, вложенный в <NetworkLinkControl>, служит для хранения переменной `count` – индекса строки таблицы координат, хранящейся в переменной `continents`. Здесь мы имеем два файла: *continents.kml* (содержит сетевую ссылку для загрузки файла с сервера) и *continents.py* (находится на сервере, генерирует по запросу KML-документ и возвращает его клиенту). Описание последовательности событий – запрос файла клиентом и обновление – приведено после листингов.

continents.kml

```
<?xml version="1.0" encoding="utf-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <NetworkLink>
    <name>Путешествие по континентам</name>
    <visibility>0</visibility>
    <description>Отметьте флажок, чтобы начать путешествие</description>
    <flyToView>1</flyToView>
    <Link>
      <href>http://localhost:3000/cgi-bin/continents.py</href>
      <refreshMode>onInterval</refreshMode>
      <refreshInterval>6</refreshInterval>
    </Link>
  </NetworkLink>
</kml>
```

continents.py

```
#!/usr/bin/python

# Этот CGI-сценарий генерирует KML-документ.
# Если кук не обнаружен, то генерируется исходный KML-файл.
# Если обнаружена CGI-переменная "count=num", то генерируется KML-файл с обновлениями.

import cgi
import socket

# Имя хоста _должно_ совпадать со значением <href> в элементе NetworkLink/Link.
hostname = localhost:3000
href = "http://%s/cgi-bin/continents.py" % hostname
minrefreshperiod = 7  # seconds

# В этом списке хранятся параметры LookAt (name, lon, lat, range) для каждого континента.
continents = []
continents.append(("нигде", 0, 0, 0))  # запись с индексом 0 не используется
continents.append(("Австралия", 134.6, -20.47, 3500000))
continents.append(("Антарктида", 138.3, -86.37, 4700000))
continents.append(("Европа", 18.37, 49.18, 3400000))
continents.append(("Африка", 13.15, -1.01, 5600000))
continents.append(("Южная Америка", -67.09, -22.48, 5600000))
continents.append(("Азия", 102, 33.7, 4700000))
continents.append(("Северная Америка", -100.7, 31.7, 4200000))

# Значение <cookie> может оказаться в списке CGI-параметров.
fs = cgi.FieldStorage()
if fs.has_key("count"):
  # Мы нашли CGI-параметр count=val. Выделим его значение.
  # В исходном KML-файле устанавливается count=1. Увеличить на 1 для поиска первой записи.
  count = int(fs["count"].value) + 1
```

```
  # Индексируем список континентов. После записи 7 вернуться к первой.
  if count > 7:
    count = 1
else:
  # Кук не найден.  Установить значение count, так чтобы генерировался исходный KML.
  count = 0


def Original():
  k = []
  k.append("<kml>")
  k.append("<NetworkLinkControl>")
  k.append("<minRefreshPeriod>%d</minRefreshPeriod>" % minrefreshperiod)
  k.append("<cookie>count=0</cookie>")
  k.append("</NetworkLinkControl>")
  k.append("<Placemark>")
  k.append("<name>Перемещающаяся точка</name>")
  k.append("<Point id=\"pt0\">')
  k.append("<coordinates>0,0</coordinates>")
  k.append("</Point>")
  k.append("</Placemark>")
  k.append("</kml>")
  return "\n".join(k)

# При вызове этой функции справедливо условие 1 <= count <= 7
def Update(count, targethref):
  global continents
  name = continents[count][0]
  longitude = continents[count][1]
  latitude = continents[count][2]
  range = continents[count][3]
  k = []
  k.append("<kml>")
  k.append("<NetworkLinkControl>")
  k.append("<minRefreshPeriod>%d</minRefreshPeriod>" % minrefreshperiod)
  k.append("<message>Следующая остановка: %s</message>" % name)
  k.append("<cookie>count=%d</cookie>" % count)
  k.append("<Update>")
  k.append("<targetHref>%s</targetHref>" % targethref)
  k.append("<Change>")
  k.append("<Point targetId=\"pt0\">')
  k.append("<coordinates>%f,%f</coordinates>" % (longitude, latitude))
  k.append("</Point>")
  k.append("</Change>")
  k.append("</Update>")
  k.append("<LookAt>")
  k.append("<longitude>%f</longitude>" % longitude)
  k.append("<latitude>%f</latitude>" % latitude)
  k.append("<range>%d</range>" % range)
  k.append("</LookAt>")
  k.append("</NetworkLinkControl>")
  k.append("</kml>")
  return "\n".join(k)

print "Content-type: text/plain"
print

if count == 0:
  print Original()
else:
  print Update(count, href)
```

Последовательность событий в этом примере такова:

1. В первый раз загружается файл *continents.kml*.
2. Сетевая ссылка в файле *continents.kml* обращается к сценарию, указанному в элементе <href>.
3. В результате этого обращения сценарий выполняется в первый раз. В URL, отправленном серверу, элемента <cookie> еще нет.

4. Сценарий выполняется и отправляет клиенту следующий KML-документ (отметим, что в нем имеется элемент <cookie>, в котором `count` равно 0).

```
<kml>
  <NetworkLinkControl>
    <minRefreshPeriod>7</minRefreshPeriod>
    <cookie>count=0</cookie>
  </NetworkLinkControl>
  <Placemark>
    <name>Moving point</name>
    <Point id="pt0">
      <coordinates>0,0</coordinates>
    </Point>
  </Placemark>
</kml>
```

5. Когда пользователь в первый раз обновляет файл *continents.kml*, сетевая ссылка снова обращается к URL, указанному в элементе <href>, но на этот раз включает в строку запроса кук, полученный от сервера:

```
http://localhost:3000/cgi-bin/continents.py?count=0
```

6. Сценарий видит кук со значением `count=0`, увеличивает `count` на 1 и использует получившееся число как индекс в таблицу континентов. На первом месте в этой таблице находится Австралия. Поэтому клиенту Google Earth теперь отправляется такой KML-документ:

```
<kml>
  <NetworkLinkControl>
    <minRefreshPeriod>7</minRefreshPeriod>
    <message>Следующая остановка: Австралия</message>
    <cookie>count=1</cookie>
    <Update>
      <targetHref>http://localhost:3000/cgi-bin/continents.py</targetHref>
      <Change>
        <Point targetId="pt0">
          <coordinates>134.600000,-20.470000</coordinates>
        </Point>
      </Change>
    </Update>
    <LookAt>
      <longitude>134.600000</longitude>
      <latitude>-20.470000</latitude>
      <range>3500000</range>
    </LookAt>
  </NetworkLinkControl>
</kml>
```

7. При следующем обновлении `count=1`, поэтому сценарий присвоит `count` значение 2 и найдет в таблице вторую запись, где находится Антарктида. Следовательно, клиенту Google Earth будет отправлен такой документ:

```
<kml>
  <NetworkLinkControl>
    <minRefreshPeriod>7</minRefreshPeriod>
    <message>Следующая остановка: Антарктида</message>
    <cookie>count=2</cookie>
    <Update>
      <targetHref>http://localhost:3000/cgi-bin/continents.py</targetHref>
      <Change>
        <Point targetId="pt0">
          <coordinates>138.300000,-86.370000</coordinates>
        </Point>
      </Change>
    </Update>
    <LookAt>
```

```
        <longitude>138.300000</longitude>
        <latitude>-86.370000</latitude>
        <range>4700000</range>
      </LookAt>
    </NetworkLinkControl>
  </kml>
```

8. При последующих обновлениях процедура повторяется. Клиент возвращает серверу полученный от него же кук. Сервер увеличивает значение `count` на 1, находит в таблице `continents` соответствующую запись и подставляет данные из нее в генерируемый KML-документ. Когда счетчик достигнет значения 7, он будет сброшен обратно в 1.

Примечание

В этом примере от пользователя требуется вручную закрывать всплывающее окно для каждой новой точки, в противном случае открытые окна будут загромождать панораму. Обратите также внимание, что контроллер сетевой ссылки задает частоту обновления раз в 7 секунд (элемент <minRefreshPeriod>). Это разумная мера предосторожности, поскольку клиент, требующий обновления слишком часто, мог бы создать недопустимо высокую нагрузку на сервер. Напомним, что частота обновления, указанная в элементе <NetworkLinkControl>, отменяет значение, заданное на стороне клиента.

Время и анимация в KML

В этом разделе описывается, как с помощью элементов задания времени в KML ограничить видимость детали одним моментом времени (TimeStamp) или некоторым промежутком времени (TimeSpan). Элементы задания времени дают относительно простой способ реализовать анимацию в геобраузере. Если в KML-файле встречаются такие элементы, то геобраузер автоматически создает ползунок, концы которого соответствуют самому раннему и самому позднему из указанных в файле моментов времени. Затем пользователь может «воспроизвести» файл с помощью кнопки воспроизведения или буксируя сам ползунок.

Существует много способов применить элементы задания времени, поэтому примеры из этой главы следует рассматривать лишь как отправную точку. Наиболее часто используются следующие приемы:

- Импорт данных из GPS-приемника и использование элемента <TimeStamp> для каждой точки во времени и пространстве. Обычно с меткой ассоциируется временной штамп, позволяющий показать ее в определенный момент времени (см. пример «Анимированная метка» ниже).
- Ассоциирование деталей KML с временными штампами для создания графических представлений трендов или перемещений. В качестве примера см. созданный Декланом Батлером KML-файл, иллюстрирующий распространение птичьего гриппа (рис. 7.3).
- Анимация рельефных наложений путем задания последовательности временных интервалов для набора изображений. Примером может служить созданная Валерием Хронусовым визуализация данных, предоставленных Роном Блейки (рис. 7.4). На ней показано, как на протяжении 600 миллионов лет земные континенты эволюционировали к текущей конфигурации.

Рис. 7.3. Метки с временными штампами позволяют создать выразительную анимацию распространения по планете птичьего гриппа за последние несколько лет (визуализация в виде KML-файла создана Декланом Батлером, www.declanbutler.info)

Рис. 7.4. В этой анимации используется последовательность рельефных наложений, с каждым из которых ассоциирован временной интервал. Демонстрируется дрейф континентов на протяжении 600 миллионов лет (визуализация в виде KML-файла создана Валерием Хронусовым на основе данных, предоставленных Роном Блейки, Университет Северной Аризоны, http://earth.doogle.cpm/gallery/index.html)

- Анимация модели путем задания различных временных интервалов для последовательных положений модели. Эту технику иллюстрирует впечатляющая анимация гигантского колеса обозрения в Лондоне, созданная Джеймсом Стаффордом (рис. 7.5).

Рис. 7.5. Джеймс Стаффорд создал анимацию «Лондонского глаза» с помощью последовательности меток, в каждой из которых угол поворота колеса обозрения отличается. Промежутки времени, в течение которых отображается каждое состояние модели, заданы с помощью элементов <TimeSpan>, www.barnabu.co.uk (скорость воспроизведения контролируется пользователем)

Задание времени

В элементах <TimeStamp> и <TimeSpan> время задается в формате `dateTime`, который определен в стандарте XML Schema (www.w3.org/TR/xmlschema-2/#isoformats). Значение представляется в виде:

```
yyyy-mm-ddThh:mm:sszzzzzz
```

где

yyyy

Четырехзначный номер года (например, 1950).

mm

Двузначный номер месяца от 01 до 12 (например, февраль обозначается как 02).

dd

Двузначный порядковый номер дня в месяце от 01 до 31 в зависимости от месяца (например, 28).

T

Разделитель даты и времени.

hh

Двузначное число часов от 00 до 24. Если значение равно 24, то и минуты, и секунды должны быть равны 00.

mm

Двузначное число минут от 00 до 59.

ss

Двузначное число секунд (обычно от 00 до 59, хотя спецификация XML допускает и значение 60 для обозначения «високосной секунды»).

zzzzzz

Либо ***Z*** для обозначения того, что используется всеобщее скоординированное время (UTC), либо смещение относительно UTC (***+hh:mm*** или ***–hh:mm***). Дополнительную информацию о стандарте UTC см. в следующем разделе.

Формат `dateTime` позволяет также задавать одну дату (***yyyy-mm-dd***).

Всеобщее скоординированное время (UTC)

UTC – это мировой стандарт отсчета времени. Он был принят вместо Гринвичского времени (GMT) в 1972 году. UTC еще называют «зулусским временем» (Zulu time), поэтому в спецификации времени в XML (применяемой также и в KML) для обозначения UTC-времени употребляется буква «Z». Часовые пояса можно задавать с помощью смещения от UTC (с плюсом или минусом в зависимости от положения относительно начального меридиана).

Если вы хотите записать местное время со смещением от UTC, то поищите величину смещения в Интернете, например на следующей странице:

`http://www.timeanddate.com/worldclock/city.html`

Ниже приведены смещения для некоторых крупных городов:

- Нью-Йорк, США: UTC–5 часов (или –4 часа, когда действует летнее время);
- Сидней, Австралия: UTC+10 часов (или +11, когда действует летнее время);
- Пекин, Китай: UTC–8 часов (перехода на летнее время нет).

Примеры задания времени

Ниже приведены несколько примеров задания времени в формате XML:

Год

Вот как задается 1990 год:

```
<when>1990</when>
```

Год и месяц

Вот как задается май 1990 года:

```
<when>1990-05</when>
```

Год, месяц и день

Вот как задается 17 мая 1990 года:

```
<when>1990-05-17</when>
```

Дата и время (UTC)

В следующем примере `T` отделяет календарную дату от времени, а `Z` говорит, что задано UTC-время. Секунды обязательны.

```
<when>1990-05-17T07:30:25Z</when>
```

Дата и время (с приведением к UTC)

В следующем примере задано нью-йоркское время (10:30:15), а за ним смещение от UTC (–5 часов):

```
<when>1990-05-17T10:30:15-05:00</when>
```

Синтаксис элемента <TimeStamp>

Элемент <TimeStamp> представляет отдельный момент времени и может быть потомком любого элемента, производного от Feature. Ниже приводится его синтаксис:

```
<TimeStamp id="ID">
  <when>...</when>
</TimeStamp>
```

<when>

Задается элемент в формате `dateTime`, описанном выше.

Синтаксис элемента <TimeSpan>

Элемент <TimeSpan> представляет промежуток времени и может быть потомком любого элемента, производного от Feature. Ниже приводится его синтаксис:

```
<TimeSpan id="ID">
  <begin>...</begin>
  <end>...</end>
</TimeSpan>
```

<begin>

Задается элемент в формате `dateTime`, обозначающий начало промежутка времени, в течение которого отображается элемент Feature. Если элемент <begin> не задан, промежуток считается не ограниченным слева.

<end>

Задается элемент в формате `dateTime`, обозначающий конец промежутка времени, в течение которого отображается элемент Feature. Если элемент <end> не задан, промежуток считается не ограниченным справа.

Временные штампы и GPS-данные

Обычно временной штамп ассоциируется с точечной меткой. Отображение значка метки на короткий миг вдоль некоторого пути создает эффект анимации. Эта техника особенно полезна для визуализации координат, импортированных из GPS-приемника. Поскольку приемник опрашивает спутник через одинаковые промежутки времени, метка перемещается вдоль пути с постоянной скоростью.

Временные штампы обычно применяются для небольших наборов данных, отображаемых в нескольких местах (например, для меток с простыми значками, которые визуально перемещаются вдоль некоторого пути). В таких случаях часто одновременно видны несколько деталей, находящихся в близких точках в разные моменты времени. Ползунок в пользовательском интерфейсе Google Earth включает временное окно, которое отмечает текущий интервал и перемещается слева направо.

В следующих двух примерах элемент <TimeStamp> применяется для анимации метки. В первом примере используются данные, полученные от GPS-приемника, во втором – большой набор данных, созданный исследователями распространения птичьего гриппа.

Пример использования временных штампов с данными от GPS-приемника

В этом примере анимируется перемещение нестандартного значка (турист-байдарочник в профиль) вдоль пути. Координаты меток были получены от GPS-приемника. Временные штампы очень близки друг к другу, поэтому при воспроизведении KML-файла значок перемещается довольно быстро (рис. 7.6).

Для начала посмотрите, как в приведенном ниже фрагменте кода используются элементы <TimeStamp>, в которых дата и время заданы в формате UTC. Затем обратите внимание на элемент <Style>; будет полезно вспомнить, чему мы научились в главе 4 «Стили и значки». Первая метка содержит нестандартный значок – байдарочное весло, обозначенное «A», – который обозначает начальный пункт маршрута. Последняя метка содержит значок весла, обозначенный «B», и соответствует конечному пункту маршрута. Внутри элемента <Document> есть также элемент <ListStyle>, в который вложен <listItemType>, равный `checkHideChildren`. После того как стиль *определен*, на него можно *сослаться*, то есть применить к любым потомкам документов. В результате многочисленные метки в списке Google Earth скрыты.

Ниже приведен фрагмент кода этого примера:

TimeStampGPSExample.kml

```
<?xml version="1.0" encoding="utf-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Document>
    <name>Точки с временными штампами</name>
    <!-- определяем стили, используемые в метках и в списке ->
    <Style id="paddle-a">
```

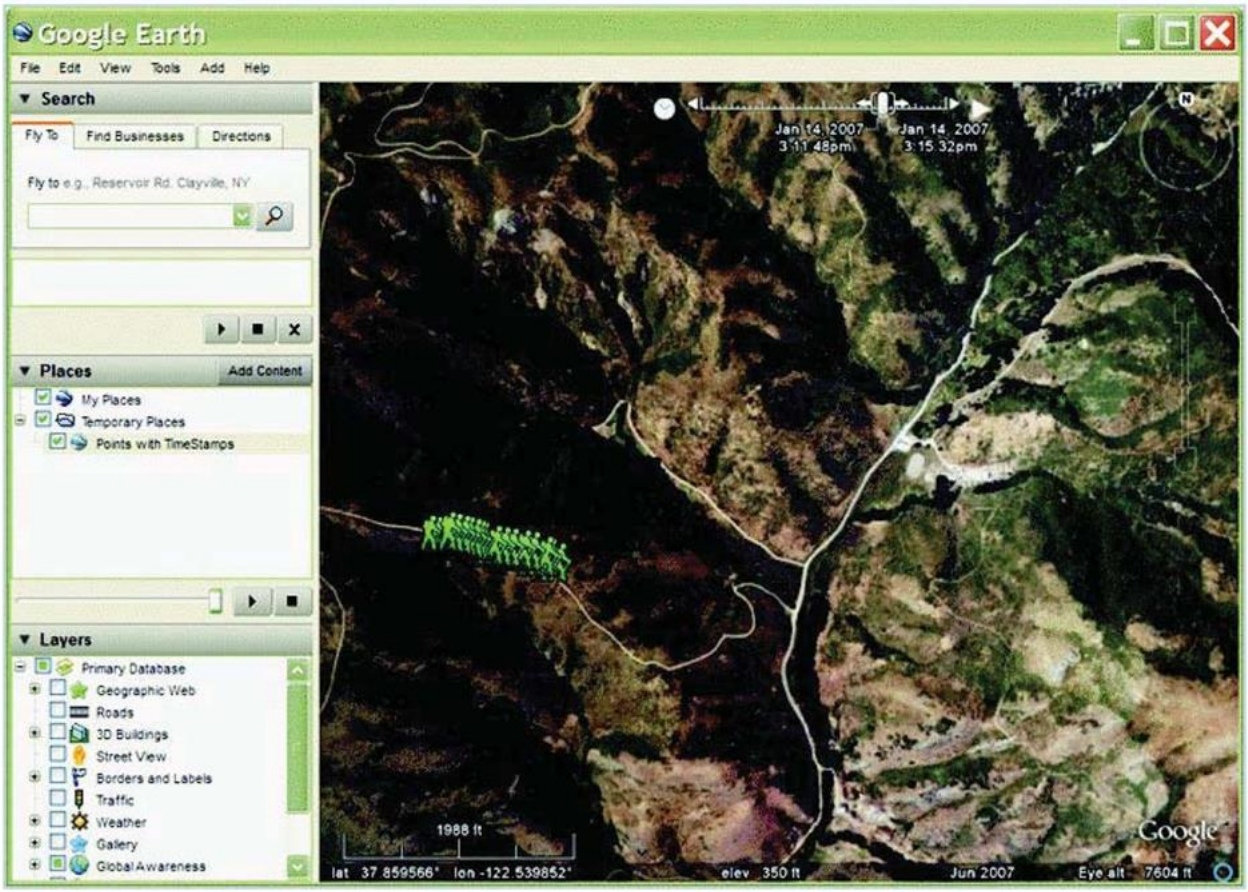


Рис. 7.6. Анимированная метка перемещается вдоль пути в соответствии с временными штампами, заданными в каждой метке. В элементе <Document> имеется элемент <ListStyle>, в который вложен <listItemType>, равный `checkHideChildren`. Это препятствует загромождению панели списка сотнями меток

```
    <IconStyle>
      <Icon>
        <href>http://maps.google.com/mapfiles/kml/paddle/A.png</href>
      </Icon>
      <hotSpot x="32" xunits="pixels" y="1" yunits="pixels"/>
    </IconStyle>
  </Style>
  <Style id="paddle-b">
    <IconStyle>
      <Icon>
        <href>http://maps.google.com/mapfiles/kml/paddle/B.png</href>
      </Icon>
      <hotSpot x="32" xunits="pixels" y="1" yunits="pixels"/>
    </IconStyle>
  </Style>
  <Style id="hiker-icon">
    <IconStyle>
      <Icon>
        <href>http://maps.google.com/mapfiles/ms/icons/hiker.png</href>
      </Icon>
      <hotSpot x="0" xunits="fraction" y="0.5" yunits="fraction"/>
    </IconStyle>
  </Style>
  <Style id="check-hide-children">
    <ListStyle>
```

```
      <listItemType>checkHideChildren</listItemType>
    </ListStyle>
  </Style>
  <styleUrl>#check-hide-children</styleUrl>
  <Placemark> <!-- первая метка, значок весла A ->
    <TimeStamp>
      <when>2007-01-14T21:05:02Z</when>
    </TimeStamp>
    <styleUrl>#paddle-a</styleUrl>
    <Point>
      <coordinates>-122.536226,37.86047,0</coordinates>
    </Point>
  </Placemark>
  <Placemark>
    <TimeStamp>
      <when>2007-01-14T21:05:20Z</when>
    </TimeStamp>
    <styleUrl>#hiker-icon</styleUrl>
    <Point>
      <coordinates>-122.536422,37.860303,0</coordinates>
    </Point>
  </Placemark>
  <Placemark>
    <TimeStamp>
      <when>2007-01-14T21:05:43Z</when>
    </TimeStamp>
    <styleUrl>#hiker-icon</styleUrl>
    <Point>
      <coordinates>-122.536688,37.860072,0</coordinates>
    </Point>
  </Placemark>
  <Placemark>
    <TimeStamp>
      <when>2007-01-14T21:06:04Z</when>
    </TimeStamp>
    <styleUrl>#hiker-icon</styleUrl>
    <Point>
      <coordinates>-122.536923,37.859855,0</coordinates>
    </Point>
  </Placemark>
  <Placemark>
    <TimeStamp>
      <when>2007-01-14T21:06:24Z</when>
    </TimeStamp>
    <styleUrl>#hiker-icon</styleUrl>
    <Point>
      <coordinates>-122.537116,37.85961,0</coordinates>
    </Point>
  </Placemark>
  <Placemark>
    <TimeStamp>
      <when>2007-01-14T21:06:46Z</when>
    </TimeStamp>
    <styleUrl>#hiker-icon</styleUrl>
    <Point>
      <coordinates>-122.537298,37.859336,0</coordinates>
    </Point>
  </Placemark>
  <Placemark>
    <TimeStamp>
      <when>2007-01-14T21:07:07Z</when>
    </TimeStamp>
    <styleUrl>#hiker-icon</styleUrl>
    <Point>
      <coordinates>-122.537469,37.85907,0</coordinates>
    </Point>
  </Placemark>
.
.
.
```

```
    <Placemark> <!-- последняя метка, значок весла B ->
      <TimeStamp>
        <when>2007-01-14T22:36:20Z</when>
      </TimeStamp>
      <styleUrl>#paddle-b</styleUrl>
      <Point>
        <coordinates>-122.536248,37.860445,0</coordinates>
      </Point>
    </Placemark>
  </Document>
</kml>
```

Пример использования временных штампов с анимированными данными о метках

Этот пример представляет собой небольшой фрагмент данных, участвующих в визуализации распространения птичьего гриппа, которую подготовил Деклан Батлер (рис. 7.3). В нем используется та же техника, что и выше, – с каждой меткой <Placemark> связан элемент <TimeStamp>. Но теперь мы не будем показывать перемещение метки вдоль пути, а посмотрим, как можно использовать временные штампы, чтобы сравнить относительные моменты времени и местоположения для дискретных событий.

Все метки относятся к Турции, с каждой ассоциирован стиль Style_13, определенный в оригинальном KML-файле для Турции. (В оригинальном файле каждой стране соответствует свой стиль.) Этот коротенький пример поможет вам освежить в памяти все, что вы знаете о стилях, картах стилей и разделяемых стилях.

Google Earth создает ползунок, охватывающий период от 30 сентября 2005 до 20 февраля 2007, поскольку этот набор меток создан именно для такого промежутка времени.

Для простоты мы опустили большинство информационных окон. Если вам интересно, как форматируются описания информационных окон, щелкните по метке для города Батман или Диярбакыр.

AvianFluExcerpt.kml

```
<?xml version="1.0" encoding="utf-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Document>
    <StyleMap id="Style_13">
      <Pair>
        <key>normal</key>
        <styleUrl>#Style_13n</styleUrl>
      </Pair>
      <Pair>
        <key>highlight</key>
        <styleUrl>#Style_13h</styleUrl>
      </Pair>
    </StyleMap>
    <Style id="Style_13n">
      <IconStyle>
        <scale>0.78125</scale>
        <Icon>
          <href>Style_13n.png</href>
        </Icon>
      </IconStyle>
```

```
    <LabelStyle>
      <scale>0</scale>
    </LabelStyle>
    <PolyStyle>
      <color>ff000000</color>
      <fill>1</fill>
      <outline>0</outline>
    </PolyStyle>
    <BalloonStyle>
      <bgColor>ffdcf5f5</bgColor>
      <textColor>ff000000</textColor>
      <text>$[description]</text>
    </BalloonStyle>
  </Style>
  <Style id="Style_13h">
    <IconStyle>
      <scale>0.78125</scale>
      <Icon>
        <href>Style_13h.png</href>
      </Icon>
    </IconStyle>
    <LabelStyle>
      <color>ff00ffff</color>
      <scale>1</scale>
    </LabelStyle>
    <PolyStyle>
      <color>ff000000</color>
      <fill>1</fill>
      <outline>0</outline>
    </PolyStyle>
    <BalloonStyle>
      <bgColor>ffdcf5f5</bgColor>
      <textColor>ff000000</textColor>
      <text>$[description]</text>
    </BalloonStyle>
  </Style>
  <Folder>
    <name>Data</name>
    <visibility>1</visibility>
    <Placemark id="pm288">
      <name>Adana</name>
      <description>Turkey</description>
      <TimeStamp>
        <when>2006-01-15</when>
      </TimeStamp>
      <styleUrl>#Style_13</styleUrl>
      <Point id="g288">
        <altitudeMode>clampToGround</altitudeMode>
        <coordinates>35.49832811,37.027542491,0</coordinates>
      </Point>
    </Placemark>
    <Placemark id="pm289">
      <name>Adiyaman</name>
      <description>Turkey</description>
      <TimeStamp>
        <when>2006-01-07</when>
      </TimeStamp>
      <styleUrl>#Style_13</styleUrl>
      <Point id="g289">
        <altitudeMode>clampToGround</altitudeMode>
        <coordinates>38.25,37.75,0</coordinates>
      </Point>
    </Placemark>
    <Placemark id="pm290">
      <name>Adiyaman</name>
      <description>Turkey</description>
      <TimeStamp>
        <when>2006-01-13</when>
      </TimeStamp>
      <styleUrl>#Style_13</styleUrl>
```

```
        <Point id="g290">
          <altitudeMode>clampToGround</altitudeMode>
          <coordinates>38.287163037,37.745354829,0</coordinates>
        </Point>
      </Placemark>
      <Placemark id="pm293">
        <name>Agri</name>
        <description>Turkey</description>
        <TimeStamp>
          <when>2005-12-31</when>
        </TimeStamp>
        <styleUrl>#Style_13</styleUrl>
        <Point id="g293">
          <altitudeMode>clampToGround</altitudeMode>
          <coordinates>44.135769502,39.62933027,0</coordinates>
        </Point>
      </Placemark>
      <Placemark id="pm294">
        <name>Agri</name>
        <description>Turkey</description>
        <TimeStamp>
          <when>2005-11-25</when>
        </TimeStamp>
        <styleUrl>#Style_13</styleUrl>
        <Point id="g294">
          <altitudeMode>clampToGround</altitudeMode>
          <coordinates>42.592256972,39.834565669,0</coordinates>
        </Point>
      </Placemark>
      <Placemark id="pm295">
        <name>Agri</name>
        <description>Turkey</description>
        <TimeStamp>
          <when>2005-11-29</when>
        </TimeStamp>
        <styleUrl>#Style_13</styleUrl>
        <Point id="g295">
          <altitudeMode>clampToGround</altitude>
          <coordinates>43.088015653,39.505741873,0</coordinates>
        </Point>
      </Placemark>
      <Placemark id="pm337">
        <name>Aksaray</name>
        <description>Turkey</description>
        <TimeStamp>
          <when>2006-01-07</when>
        </TimeStamp>
        <styleUrl>#Style_13</styleUrl>
        <Point id="g337">
          <altitudeMode>clampToGround</altitudeMode>
          <coordinates>33.8545596715,38.4068859713,0</coordinates>
        </Point>
      </Placemark>
.
.
.
    </Folder>
  </Document>
</kml>
```

Временные интервалы и рельефные наложения

Элементом <TimeSpan> можно воспользоваться для отображения последовательности многоугольников или накладываемых изображений с мгновенными переходами от предыдущего к последующему члену. Временные интервалы при-

меняются в тех случаях, когда в любой момент времени должна быть видна только одна деталь из нескольких, занимающих одно и то же положение на карте. Обычно эту технику применяют для демонстрации изменений многоугольников или рельефных наложений – например, чтобы показать траекторию сползания ледника, распространение вулканического пепла или последствия многолетней вырубки лесов.

Имейте в виду, что все элементы <TimeSpan> в KML-файле должны располагаться в одном месте, а интервалы не должны перекрываться. Если в наборе данных встречаются временные интервалы, то в пользовательском интерфейсе Google Earth появляется ползунок, внутри которого имеется стрелка, плавно перемещающаяся слева направо. Переход от предыдущей детали к последующей производится мгновенно.

В следующем примере демонстрируется применение элемента <TimeSpan> для анимации последовательности рельефных наложений.

Пример использования временных интервалов для анимации рельефных наложений

В этом примере анимируется последовательность рельефных наложений, созданная на основе данных, собранных Роном Блейки. Валерию Хронусову пришлось проявить изобретательность для задания моментов времени в элементах <TimeSpan>, поскольку спецификация `dateTime` допускает только четыре цифры года, а эта визуализация охватывает период в 600 миллионов лет.

В файл на сайте этой книги включены только три из 26 геологических эр, содержащихся в оригинальной визуализации Валерия Хронусова. Название и данные для каждой эры хранятся внутри отдельной папки <Folder>. Элемент <TimeSpan>, определенный внутри <Folder>, относится ко всем деталям в этой папке. Каждая папка содержит две детали Feature:

- элемент <GroundOverlay> с изображением структуры континента для этой эры;
- элемент <Placemark>, в котором <name> содержит надпись, отображаемую поверх рельефного наложения. Тут применен интересный трюк – точка, ассоциированная с меткой, расположена на высоте 1 000 000 метров, а параметр <scale> в элементе <LabelStyle> равен 8. Поскольку единственное назначение этой метки <Placemark> – предоставить надпись для <GroundOverlay>, значок (элемент <Icon>) для нее пуст.

Отметим еще, что в каждой папке <Folder> есть элемент <ListStyle>, определяющий цвет фона той записи в списке, которая соответствует данной метке. Цвета записей постепенно меняются от темно-оранжевого (для самых ранних эр) до желтого (современная эра). Потомок <listItemType> элемента <ListStyle> равен `checkHideChildren`, так как наличие в списке отдельных записей для <Placemark> и <GroundOverlay> только загромождало бы его (и даже могло бы ввести зрителя в заблуждение).

PaleoGlobeExcerpt.kml

```
<?xml version="1.0" encoding="utf-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Document>
    <name>Глобальная палеография</name>
    <description><![CDATA[Изображения публикуются с любезного разрешения
<a href="http://jan.ucc.nau.edu/~rcb7/">профессора Рона Блейки</a>]]>
    </description>
    <Folder>
      <name>Поздний докембрий - 560M</name>
      <open>1</open>
      <TimeSpan>
        <begin>2000-02-09</begin>
        <end>2000-02-29T23:00:00Z</end>
      </TimeSpan>
      <Style>
        <ListStyle>
          <listItemType>checkHideChildren</listItemType>
          <bgColor>ff0039ff</bgColor>
        </ListStyle>
      </Style>
      <Placemark>
        <name>Поздний докембрий - 560M</name>
        <Style>
          <IconStyle>
            <scale>0</scale>
            <Icon/>
          </IconStyle>
          <LabelStyle>
            <color>ff0039ff</color>
            <scale>8</scale>
          </LabelStyle>
        </Style>
        <Point>
          <extrude>1</extrude>
          <altitudeMode>relativeToGround</altitudeMode>
          <coordinates>0,89.99,1000000</coordinates>
        </Point>
      </Placemark>
      <Placemark>
        <name>Поздний докембрий - 560M</name>
        <Style>
          <IconStyle>
            <scale>0</scale>
            <Icon/>
          </IconStyle>
          <LabelStyle>
            <color>ff0039ff</color>
            <scale>8</scale>
          </LabelStyle>
        </Style>
        <Point>
          <extrude>1</extrude>
          <altitudeMode>relativeToGround</altitudeMode>
          <coordinates>0,-89.99,1000000</coordinates>
        </Point>
      </Placemark>
      <GroundOverlay>
        <name>Поздний докембрий - 560M</name>
        <drawOrder>4</drawOrder>
        <Icon>
          <href>files/560Marect.jpg</href>
        </Icon>
        <LatLonBox>
          <north>90</north>
          <south>-90</south>
          <east>180</east>
```

```
          <west>-180</west>
        </LatLonBox>
      </GroundOverlay>
    </Folder>
.
.
.
  </Document>
</kml>
```

Что дальше?

Примите поздравления! Вы закончили изучение основных компонентов KML. В последней главе мы рассмотрим две более сложные темы, относящиеся к KML-файлам, которые содержат большие наборы данных. *Регионы* позволяют задать дополнительные условия, при которых данные показываются (исходя из того, куда направлен взгляд пользователя и насколько далеко он находится от рассматриваемых объектов). Механизм *расширенных данных* применяется для пополнения KML новыми нестандартными типами данных таким образом, что их можно сохранить в файле и загрузить впоследствии.

Глава 8. Большие наборы данных

Прочитав эту главу, вы сможете:

- ✓ назвать три способа применения регионов;
- ✓ создать набор ссылок, включающий регионы, которые активизируют их;
- ✓ создать (или найти в Интернете) модель, которую можно показать на трех уровнях детализации, и с помощью регионов отобразить различные варианты модели;
- ✓ воспользоваться регионами для отображения набора рельефных наложений;
- ✓ объяснить, для чего нужны суперналожения. Назвать два элемента KML, которые используются в суперналожениях;
- ✓ придумать простой набор данных, в котором можно было бы воспользоваться механизмом расширенных данных для сохранения полей вместе с их типами.

Предварительный обзор

В этой главе мы рассмотрим два понятия, которые можно отнести к продвинутым особенностям KML: *регионы* и *расширенные данные*. Читателям, знакомым с программированием, вероятно, не составит труда разобраться в изложенном здесь материале. Тем же, кто далек от программирования, обсуждение полей и типов в разделе, посвященном расширенным данным, может показаться более сложным, но основные идеи должны быть понятны. Самая трудная из рассматриваемых тем – суперналожения, и связано это с тем фактом, что мы описываем все детали реализации. На самом деле практически всегда суперналожения создаются с помощью тех или иных программных инструментов, поэтому изложенные в этой главе детали представляют интерес главным образом для разработчиков, которым нужно понять, что происходит «под капотом».

В первой части главы мы рассмотрим элемент <Region>, который может быть вложен в любой элемент, производный от Feature. Регионы используются разными способами:

- ❑ чтобы не загромождать панораму в том случае, когда в одном месте сосредоточено чересчур много деталей (рис. 8.1);
- ❑ чтобы прояснить общую картину, распределив данные разных типов по слоям, которые можно показать или скрыть в зависимости от текущего вида;
- ❑ чтобы реализовать появление и исчезновение деталей;
- ❑ чтобы увеличить производительность за счет разбиения изображений с очень высоким разрешением на более мелкие фрагменты, которые загружаются только по мере попадания в поле зрения;

Рис. 8.1. Слишком много данных на одном экране! Швейцарская система общественного транспорта насчитывает 15 000 остановок, соединенных более чем 18 000 маршрутов. Если разнести эти данные по нескольким файлам, связав их сетевыми ссылками, ассоциированными с регионами, то получится много маленьких файлов, которые можно загружать по сети, при этом уровень детализации будет доступным для восприятия человеком (см. рис. 8.11, где показано решение этой задачи, предложенное Бентом Хагемарком)

- чтобы увеличить производительность за счет загрузки моделей с разными уровнями детализации в зависимости от того, сколько места на экране отведено в данный момент для показа модели.

Во второй части главы мы будем говорить об элементе <ExtendedData>, который предоставляет три разных способа пополнить KML нестандартными типами данных. Читатель, не интересующийся техническими подробностями, может пропустить этот раздел. Он представляет интерес прежде всего для пользователей, имеющих дело с очень большими базами данных, когда имена и типы полей помещаются в KML-файл и должны быть сохранены при его выгрузке. Такие наборы данных встречаются в самых разных приложениях – торговле недвижимостью, вулканологии, иммунологии, астрономии, геологии, – поэтому очень важно иметь простой и гибкий механизм расширения KML.

Основные понятия

В этом разделе мы рассмотрим основные понятия, относящиеся к регионам. Попросту говоря, регион определяет некую географическую область. Детали, в которые вложен элемент <Region>, видимы, когда удовлетворяются два условия:

- область, определенная в регионе, в данный момент находится в поле зрения;
- будучи спроецирована на экран, область, определенная в регионе, оказывается в заданном диапазоне LOD (см. раздел «Диапазон уровней детализации (LOD)» ниже).

Если оба условия выполнены, то говорят, что регион активен. Когда регион активен, связанные с ним детали рисуются на экране (если, конечно, параметр <visibility> равен 1).

Поскольку регионы позволяют управлять тем, когда деталь появляется и исчезает на текущем фрагменте панорамы, они оказываются очень полезны для построения таких представлений больших наборов данных, которые были бы одновременно привлекательны и легко воспринимались зрителем. Впрочем, в примерах из этой главы регионы применяются и к простым данным (даже к одному рельефному наложению) для создания интересных эффектов.

В отдельном разделе будут рассмотрены *суперналожения*, то есть рельефные наложения, в которых используются сетевые ссылки, ассоциированные с регионами. Введенные здесь базовые концепции применимы и к суперналожениям. На рис. 8.2 приведен пример суперналожения.

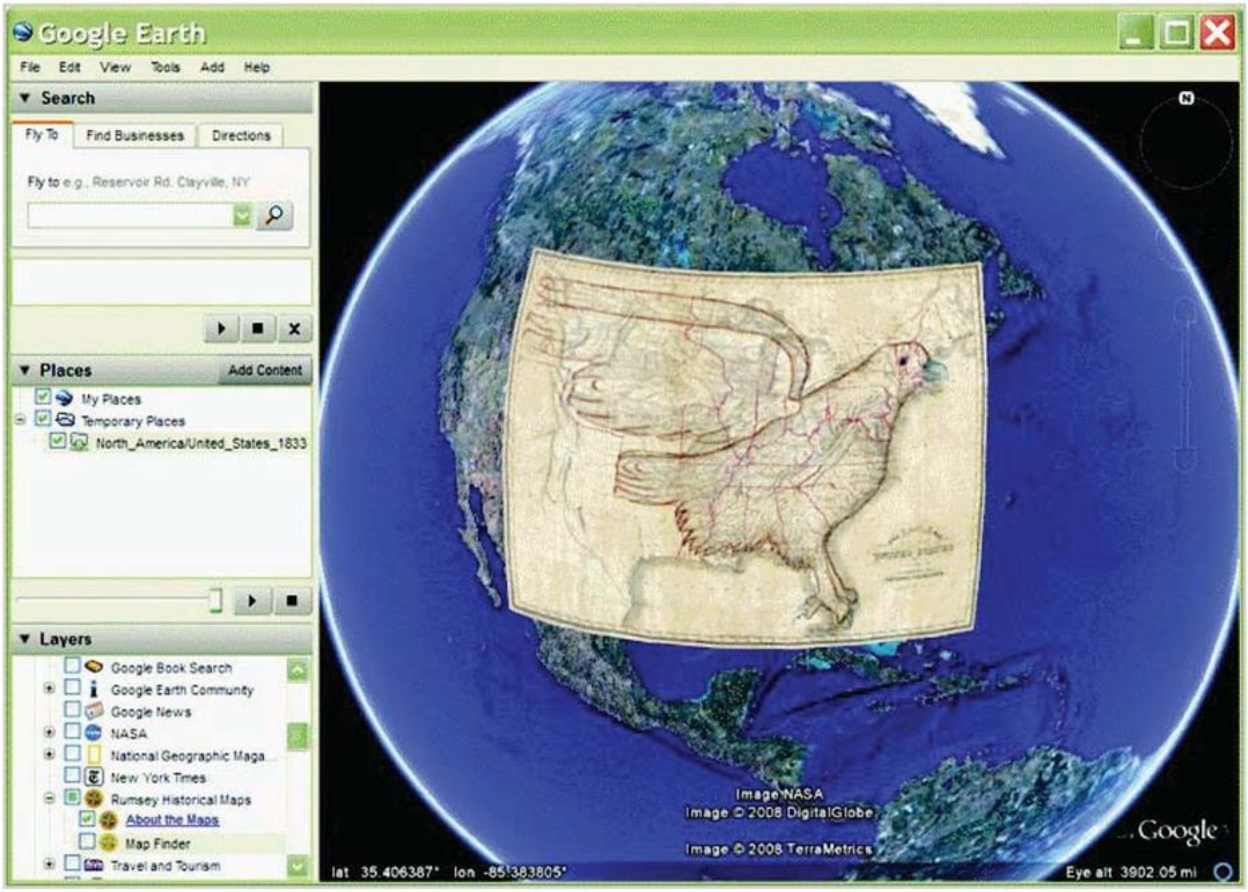


Рис. 8.2. На этом эстампе 1833 года изображен орел, наложенный на карту Соединенных Штатов. В Google Earth включен слой с картами из коллекции Дэвида Рамси, которые были преобразованы в набор суперналожений для эффективной прорисовки при любом разрешении (публикуется с разрешения коллекции карт Дэвида Рамси, www.davidrumsey.com)

Географическая область, занимаемая регионом, определяется во вложенном элементе <LatLonAltBox>. Это может быть либо плоский прямоугольник, заданный координатами *север/юг/восток/запад*, либо трехмерный параллелепипед, заданный координатами *север/юг/восток/запад/минимальнаяВысота/максимальнаяВысота* (рис. 8.3).

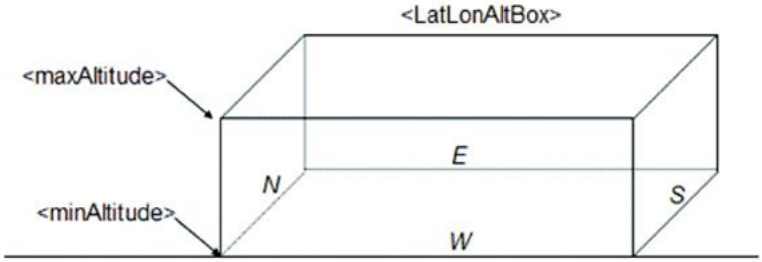


Рис. 8.3. Ограничивающий параллелепипед определяет область, занятую регионом

Для двумерного прямоугольника, расположенного на заданной высоте, также необходимы элементы <minAltitude> и <maxAltitude>. Тот и другой интерпретируются в соответствии со значением элемента <altitudeMode>, который рассматривался в главе 3. Эту область обычно называют *ограничивающим параллелепипедом*.

Примечание

Обычно деталь, включающая регион, располагается где-то внутри области, определенной элементом <LatLonAltBox>, но это не обязательно. <LatLonAltBox> региона и положение детали могут быть никак не связаны между собой.

Диапазон уровней детализации (LOD)

Аббревиатура *LOD* расшифровывается как *level of detail* (уровень детализации). Элемент <Lod> описывает, каким должен быть размер проекции региона на экран, чтобы регион считался активным.

Диапазон LOD ограничен снизу значением элемента <minLodPixels>, а сверху – значением элемента <maxLodPixels>. Измеряется LOD количеством пикселей в области, занятой проекцией региона на экран.

Значение элемента <minLodPixels> – это количество экранных пикселей, минимально необходимое для того, чтобы регион был виден. Геобраузер вычисляет размер проекции региона на экран, а затем – квадратный корень из площади региона. (Если, к примеру, регион является квадратным, расположен на экваторе, не наклонен и точка обзора находится прямо над регионом, то эта величина равна *ширине проекции региона*.) Если вычисленная величина попадает в диапазон между <minLodPixels> и <maxLodPixels> (и при этом прямоугольник <LatLonAltBox> находится в поле зрения), то регион активен. В противном случае регион неактивен, и ассоциированные с ним геометрические детали не рисуются.

```
        <rotationXY x="0" xunits="fraction" y="0" yunits="fraction"/>
        <size x="256" xunits="pixels" y="8" yunits="pixels"/>
      </ScreenOverlay>
      <ScreenOverlay>
        <name>512 px</name>
        <color>aaaaffaa</color>
        <Icon>
          <href></href>
        </Icon>
        <overlayXY x="0.5" xunits="fraction" y="0.5" yunits="fraction"/>
        <screenXY x="0.5" xunits="fraction" y="0.5" yunits="fraction"/>
        <rotationXY x="0" xunits="fraction" y="0" yunits="fraction"/>
        <size x="512" xunits="pixels" y="8" yunits="pixels"/>
      </ScreenOverlay>
    </Folder>
    <Folder>
      <name>По вермикали</name>
      <open>1</open>
      <styleUrl>#rf</styleUrl>
      <Placemark>
        <name>off</name>
      </Placemark>
      <ScreenOverlay>
        <name>64 px</name>
        <color>aaaaffaa</color>
        <Icon>
          <href></href>
        </Icon>
        <overlayXY x="0.5" xunits="fraction" y="0.5" yunits="fraction"/>
        <screenXY x="0.5" xunits="fraction" y="0.5" yunits="fraction"/>
        <rotationXY x="0" xunits="fraction" y="0" yunits="fraction"/>
        <size x="8" xunits="pixels" y="64" yunits="pixels"/>
      </ScreenOverlay>
      <ScreenOverlay>
        <name>128 px</name>
        <color>aaaaffaa</color>
        <Icon>
          <href></href>
        </Icon>
        <overlayXY x="0.5" xunits="fraction" y="0.5" yunits="fraction"/>
        <screenXY x="0.5" xunits="fraction" y="0.5" yunits="fraction"/>
        <rotationXY x="0" xunits="fraction" y="0" yunits="fraction"/>
        <size x="8" xunits="pixels" y="128" yunits="pixels"/>
      </ScreenOverlay>
      <ScreenOverlay>
        <name>256 px</name>
        <color>aaaaffaa</color>
        <Icon>
          <href></href>
        </Icon>
        <overlayXY x="0.5" xunits="fraction" y="0.5" yunits="fraction"/>
        <screenXY x="0.5" xunits="fraction" y="0.5" yunits="fraction"/>
        <rotationXY x="0" xunits="fraction" y="0" yunits="fraction"/>
        <size x="8" xunits="pixels" y="256" yunits="pixels"/>
      </ScreenOverlay>
      <ScreenOverlay>
        <name>512 px</name>
        <color>aaaaffaa</color>
        <Icon>
          <href></href>
        </Icon>
        <overlayXY x="0.5" xunits="fraction" y="0.5" yunits="fraction"/>
        <screenXY x="0.5" xunits="fraction" y="0.5" yunits="fraction"/>
        <rotationXY x="0" xunits="fraction" y="0" yunits="fraction"/>
        <size x="8" xunits="pixels" y="512" yunits="pixels"/>
      </ScreenOverlay>
    </Folder>
  </Document>
</kml>
```

Диапазон затухания

В элементе <Lod> задается также количество пикселей, используемое для перехода от прозрачного к непрозрачному состоянию и наоборот. Значение <minFadeExtent> – это расстояние, на котором геометрическая деталь совершает переход между полной непрозрачностью и полной прозрачностью. Это значение, выраженное в пикселях, относится к левому концу диапазона LOD.

Аналогично <maxFadeExtent> – расстояние, на котором геометрическая деталь совершает переход между полной непрозрачностью и полной прозрачностью на правом конце диапазона LOD. О вычислении этих значений см. следующий раздел.

Детальное описание элемента <Lod>

На рис. 8.4 приведено графическое представление потомков элемента <Lod>.

Следующий псевдокод показывает, как геобраузер применяет элементы, вложенные в <Lod>, для вычисления видимости и диапазонов затухания объектов, ассоциированных с регионом:

```
if (P < minLoadPixels)
   opacity = 0                                  // #1 на рис. 8.4
else if (P < minLoadPixels + minFadeExtent)
   opacity = (P - minLoadPixels)/minFadeExtent  // #2 на рис. 8.4
else if (P < maxLoadPixels - maxFadeExtent)
   opacity = 1                                  // #3 на рис. 8.4
else if (P < maxLoadPixels)
   opacity = (maxLoadPixels - P)/maxFadeExtent  // #4 на рис. 8.4
else
   opacity = 0                                  // #5 на рис. 8.4
```

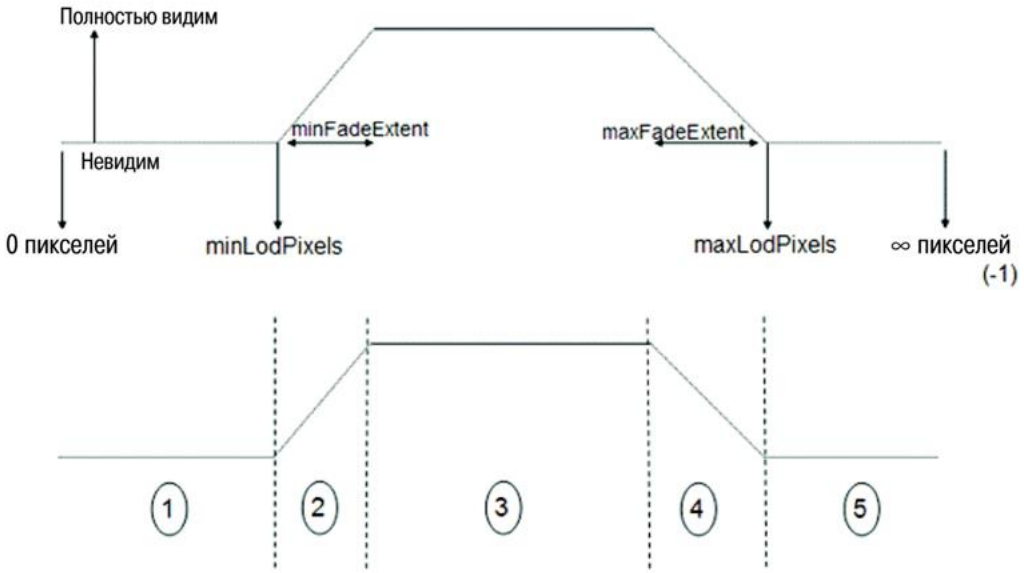


Рис. 8.4. Диапазон уровней детализации (LOD) ограничивает с двух сторон размер проекции региона на экране. Если проекция региона оказывается вне диапазона, то ассоциированные с ним детали не видны

Как регионы соотносятся с режимом обновления фрагмента

Элементы <Link> и <Icon> могут содержать элемент <viewRefreshMode>, одним из возможных значений которого является `onRegion`. Оно означает, что файл или сетевую ссылку следует обновить, когда ассоциированный с ней регион становится активен. Это очень эффективный механизм, так как KML-файл загружается лишь в том случае, когда регион попадает в поле зрения и удовлетворяет условиям видимости. Обновление в режиме `onRegion` лежит также в основе работы суперналожений, которые будут рассмотрены ниже.

Каскадное определение регионов

В иерархии Feature есть много элементов, для которых можно задавать регионы. Для элементов, производных от Container или NetworkLink, регионы, определенные в родительском элементе, применяются ко всем потомкам, если только для потомка не определен собственный регион. В файле *WineRegions.kml* ниже приведен пример задания региона локально и внутри родительского контейнера.

В KML механизм каскадирования применяется к регионам, элементам задания времени и элементам <atom> (глава 3). Например, элемент TimeSpan, заданный на уровне <Document>, применяется ко всем вложенным в документ деталям, если только не будет переопределен элементом TimeSpan в какой-нибудь детали, расположенной ниже в иерархии. На детали в файле, загруженном по сетевой ссылке, действуют те параметры регионов, временных примитивов и элементов <atom>, которые заданы в родительском файле, содержащем ссылку. (Отметим, что это поведение отличается от разделяемых стилей, которые не распространяются каскадно вниз по иерархии. Каждая деталь Feature должна явно сослаться на разделяемый стиль с помощью элемента <styleUrl>.)

Суперналожения

Суперналожение представляет собой набор рельефных наложений, загружаемый по сетевым ссылкам, ассоциированным с регионами. Говорят, что сетевая ссылка ассоциирована с регионом, если она содержит элемент <viewRefreshMode>, в который вложен элемент <Link>, имеющий значение `onRegion`. Такие наложения организуются в особую иерархию, обеспечивающую эффективную обработку геобраузером. В настоящей главе конструкция этой иерархии будет подробно описана. Возможно, вам потребуется специальная программа, например Regionator, для построения плиточной иерархии изображения.

Регионы

В этом разделе приведены простые примеры использования регионов для изменения способа представления данных в геобраузере.

Синтаксис элемента <Region>

```
<Region id="ID">
  <LatLonAltBox id="ID">
    <north>180.0</north>
    <south>-180.0</south>
    <east>180.0</east>
    <west>180.0</west>
    <minAltitude>0.0</minAltitude>
    <maxAltitude>0.0</maxAltitude>
    <altitudeMode>clampToGround</altitudeMode>
  </LatLonAltBox>
  <Lod id="ID">
    <minLodPixels>0.0</minLodPixels>
    <maxLodPixels>-1.0</maxLodPixels>
    <minFadeExtent>0.0</minFadeExtent>
    <maxFadeExtent>0.0</maxFadeExtent>
  </Lod>
</Region>
```

<LatLonAltBox>

Задает параллелепипед, ограничивающий область, представляющую интерес.

<north>

Широта северной стороны ограничивающего параллелепипеда. Выражена в градусах и принимает значение от –90 до +90.

<south>

Широта южной стороны ограничивающего параллелепипеда. Выражена в градусах и принимает значение от –90 до +90.

<east>

Долгота восточной стороны ограничивающего параллелепипеда. Выражена в градусах и принимает значение от –180 до +180.

<west>

Долгота западной стороны ограничивающего параллелепипеда. Выражена в градусах и принимает значение от –180 до +180.

<minAltitude>

Минимальная высота ограничивающего параллелепипеда региона, заданная в метрах и интерпретируемая в соответствии с <altitudeMode>.

<maxAltitude>

Максимальная высота ограничивающего параллелепипеда региона, заданная в метрах и интерпретируемая в соответствии с <altitudeMode>.

<altitudeMode>

Может принимать значения `clampToGround` (игнорировать задание высоты), `relativeToGround` (прибавить заданную высоту к высоте поверхности в данной точке) или `absolute` (считать, что задана высота над уровнем моря).

<Lod>

<minLodPixels>

Количество экранных пикселей, минимально необходимое для того, чтобы данный регион был видим. Google Earth вычисляет размер проекции региона на экран, а затем – квадратный корень из площади региона. Если вычисленная величина попадает в диапазон между <minLodPixels> и <maxLodPixels>

(и при этом прямоугольник <LatLonAltBox> находится в поле зрения), то регион «активен». В противном случае считается, что ассоциированная с регионом деталь находится слишком далеко от точки обзора и рисовать ее не имеет смысла.

<maxLodPixels>

Количество экранных пикселей, при превышении которого регион невидим. Значение –1, принимаемое по умолчанию, означает, что «регион активен, как бы велика ни была площадь его проекции».

<minFadeExtent>

Расстояние, на котором деталь переходит от полной прозрачности к полной непрозрачности и наоборот. Выражается в экранных пикселях и относится к левому концу диапазона уровней детализации.

<maxFadeExtent>

Расстояние, на котором деталь переходит от полной прозрачности к полной непрозрачности и наоборот. Выражается в экранных пикселях и относится к правому концу диапазона уровней детализации.

Пример рельефного наложения с регионом

В файле *HistoricOverlay.kml* строится регион для двумерного наложения на уровне земли. Черно-белое изображение, лежащее в основе этого наложения, содержит исторические данные о части городка Маунтин-Вью, относящиеся к 1991 году. Когда пользователь увеличивает масштаб этой области, наложение становится видимым. На рис. 8.5 показано, как выглядит наложение, когда в первый раз оказывается видимым. Чтобы наложение отчетливо выделялось на фоне карты, в примере используется элемент <LineString> белого цвета.

В данном случае <minLodPixels> равно 128, то есть элемент <GroundOverlay> становится видимым, когда занимаемая им площадь на экране не меньше 128 квадратных пикселей (<maxLodPixels> равно –1, то есть при дальнейшем увеличении масштаба наложение так и будет оставаться видимым).

HistoricOverlay.kmz

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://earth.google.com/kml/2.1">
  <Document>
    <name>Плоский регион</name>
    <Region>
      <LatLonAltBox>
        <north>37.430419921875</north>
        <south>37.41943359375</south>
        <east>-122.080078125</east>
        <west>-122.091064453125</west>
      </LatLonAltBox>
      <Lod>
        <minLodPixels>128</minLodPixels>
      </Lod>
    </Region>
    <GroundOverlay>
      <name>Mountain View DOQQ</name>
      <Icon>
        <href>files/image.JPEG</href>
```

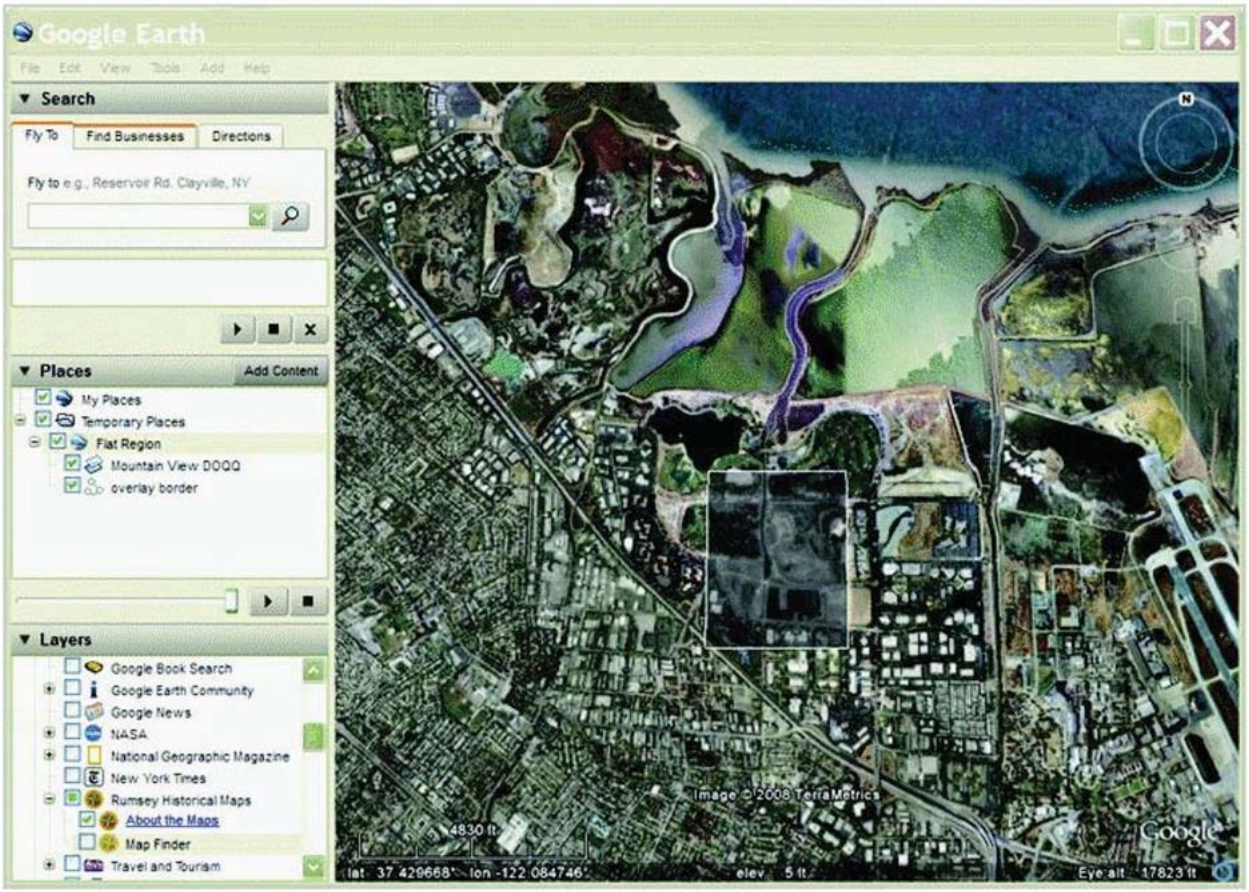


Рис. 8.5. Черно-белое рельефное наложение не видно до тех пор, пока занимаемая им на экране площадь не превысит <minLodPixels>. На этом рисунке показано, как выглядит изображение при минимальной площади (128 пикселей)

```
      </Icon>
      <LatLonBox>
        <north>37.430419921875</north>
        <south>37.41943359375</south>
        <east>-122.080078125</east>
        <west>-122.091064453125</west>
      </LatLonBox>
    </GroundOverlay>
  <Document>
</kml>
```

В данном случае включать в <LatLonAltBox> параметры <minAltitude> и <maxAltitude> необязательно, так как наложение расположено на уровне земли. Ограничивающий прямоугольник региона совпадает с границами рельефного наложения, заданными в его элементе <LatLonBox>.

Обратите также внимание, что регион в этом KML-файле является потомком изображения, видимость которого он определяет.

Попробуйте загрузить этот файл в геобраузер и поэкспериментировать с различными точками обзора. Посмотрите, как регион появляется и исчезает в зависимости от занимаемой им площади на экране. Вы увидите, что если увеличить наклон или немного отодвинуть точку обзора, то наложение пропадает, поскольку его площадь оказывается меньше <minLodPixels>.

Регион для трехмерной модели

В файле *UnitedNationsModel.kml* показано, как строится регион, содержащий трехмерные объекты на уровне земли. В элементе <LatLonAltBox> для этого региона величина <maxAltitude> равна 300 метров, то есть совпадает с высотой здания. Наверное, вы узнали комплекс зданий штаб-квартиры ООН в Нью-Йорке (рис. 8.6).

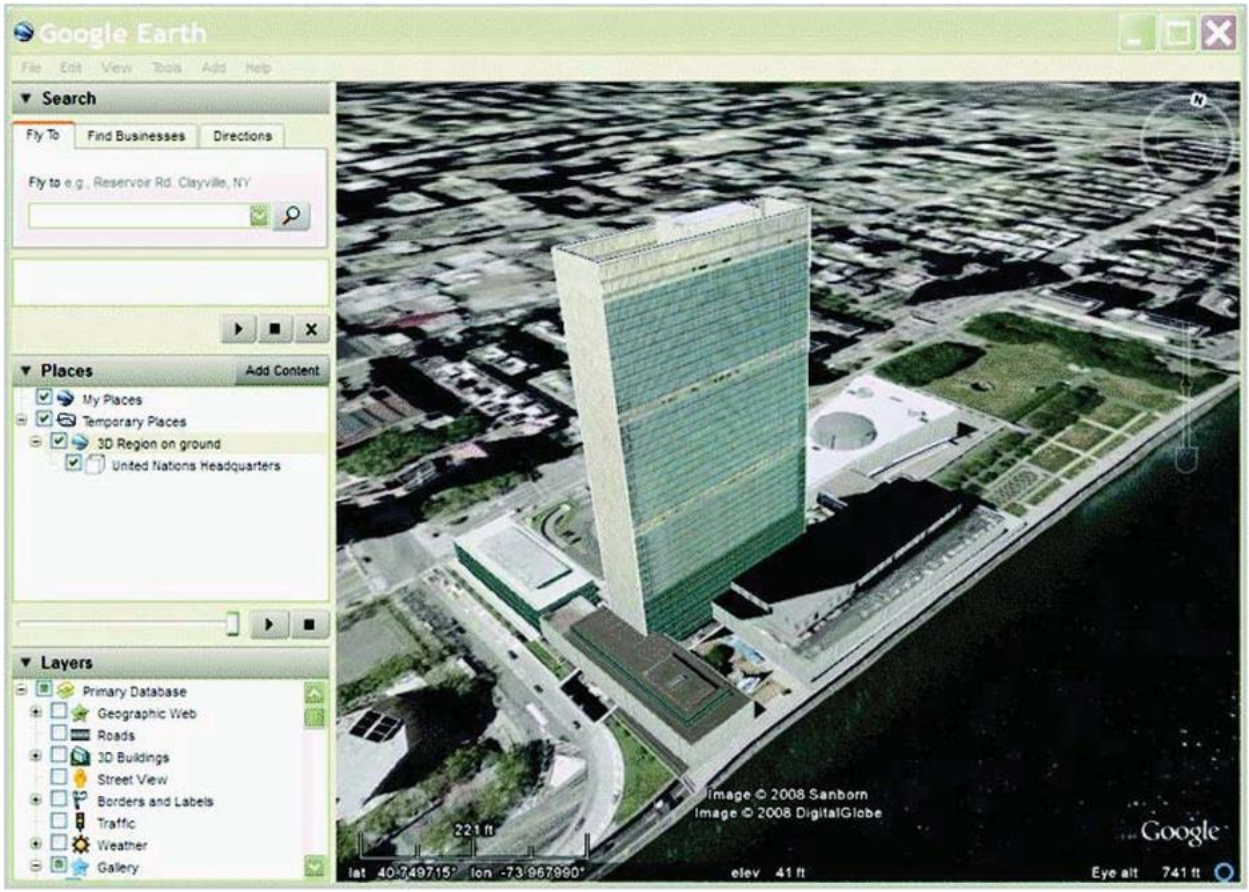


Рис. 8.6. Регионы представляют собой эффективный механизм отображения сложных моделей. Объекты видны, только когда регион активен

Важно отметить, что границы региона, заданные в <LatLonAltBox>, не обязательно должны совпадать с границами долготы и широты модели. Координаты модели задаются в ее собственной системе координат, начало которой может быть смещено относительно истинного положения модели на Земле.

UnitedNationsModel.kmz

```
<?xml version='1.0' encoding='UTF-8'?>
<kml xmlns="http://earth.google.com/kml/2.1">
  <Document>
    <name>Трехмерный регион на уровне земли</name>
    <Placemark>
      <name>Штаб-квартира ООН</name>
```

```
      <visibility>0</visibility>
      <Region>
        <Lod>
          <minLodPixels>128</minLodPixels>
        </Lod>
        <LatLonAltBox>
          <north>40.750683130314</north>
          <south>40.748162385230</south>
          <east>-73.966608428427</east>
          <west>-73.969476624071</west>
          <minAltitude>0</minAltitude>
          <maxAltitude>300</maxAltitude>
          <altitudeMode>absolute</altitudeMode>
        </LatLonAltBox>
      </Region>
      <Model>
        <altitudeMode>absolute</altitudeMode>
        <Location>
          <longitude>-73.967763927199</longitude>
          <latitude>40.749458312255</latitude>
          <altitude>0.406173708576</altitude>
        </Location>
        <Link>
          <href>models/un.dae</href>
        </Link>
      </Model>
    </Placemark>
  </Document>
</kml>
```

Регион для двумерного наложения на ненулевой высоте

В примере *CloudRegionAtAltitude.kml* показано, как выглядит двумерное наложение, расположенное выше уровня земли. Эта техника полезна, например, для показа метеорологических фронтов и схем движения воздушного транспорта. В данном случае мы видим небольшое скопление облаков на высоте 100 000 метров над уровнем моря.

CloudRegionAtAltitude.kmz

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://earth.google.com/kml/2.1">
  <Document>
    <name>Плоский регион на ненулевой высоте</name>
    <GroundOverlay>
      <name>Скопление облаков</name>
      <Region>
        <LatLonAltBox>
          <north>33.75</north>
          <south>22.5</south>
          <east>-45</east>
          <west>-56.25</west>
          <minAltitude>100000</minAltitude>
          <maxAltitude>100000</maxAltitude>
          <altitudeMode>absolute</altitudeMode>
        </LatLonAltBox>
        <Lod>
          <minLodPixels>128</minLodPixels>
        </Lod>
      </Region>
```

```
      <Icon>
        <href>files/image.PNG</href>
      </Icon>
      <altitude>100000</altitude>
      <altitudeMode>absolute</altitudeMode>
      <LatLonBox>
        <north>33.75</north>
        <south>22.5</south>
        <east>-45</east>
        <west>-56.25</west>
      </LatLonBox>
    </GroundOverlay>
  </Document>
</kml>
```

В элементе <LatLonAltBox> для этого региона оба параметра <minAltitude> и <maxAltitude> равны 100 000 метров (значения совпадают, так как наложение двумерное, то есть его толщина равна нулю). Параметр <altitudeMode> равен `absolute`, то есть высота отсчитывается от уровня моря.

Отметим, что параметр <altitude> в самом элементе <GroundOverlay> тоже равен 100 000 (то есть совпадает со значением высоты в ограничивающем параллелепипеде региона), и значения <altitudeMode> для региона и самого рельефного наложения одинаковы.

Развернутый пример: упрощение набора данных

В этом примере используется небольшое подмножество меток, взятое из обширного набора данных «Мировые регионы виноделия», который составил Антонио Рочча Грака (http://bbs.keyhole.com/ubb/showthreaded.php?Number=303649). В оригинальном файле определены 10 винодельческих районов и 187 меток, соответствующих официальным апелясьонам. Мы же включили только 39 меток и три района.

Регионы KML позволяют постепенно увеличивать объем выводимой информации в зависимости от того, что попадает в поле зрения. Не будь регионов, все данные оказались бы видны одновременно. В этом примере показано, как разбить данные на *слои*, соответствующие различным типам информации. Пример *WineRegions.kmz* ведет себя следующим образом:

1. Когда пользователь в первый раз переходит к Франции и смотрит на нее с очень большого расстояния, метки не видны.
2. По мере приближения точки обзора появляются метки пяти основных апелясьонов.
3. Если точка обзора продолжает приближаться, становятся видны розовые надписи с названиями трех регионов.
4. Наконец, на самом близком расстоянии появляются метки внутри каждого региона.

На рис. 8.7 показана увеличенная картина этой части Франции, когда все еще видна одна розовая надпись и появились апелясьоны, соответствующие местам производства известных винных марок (самый детальный уровень).

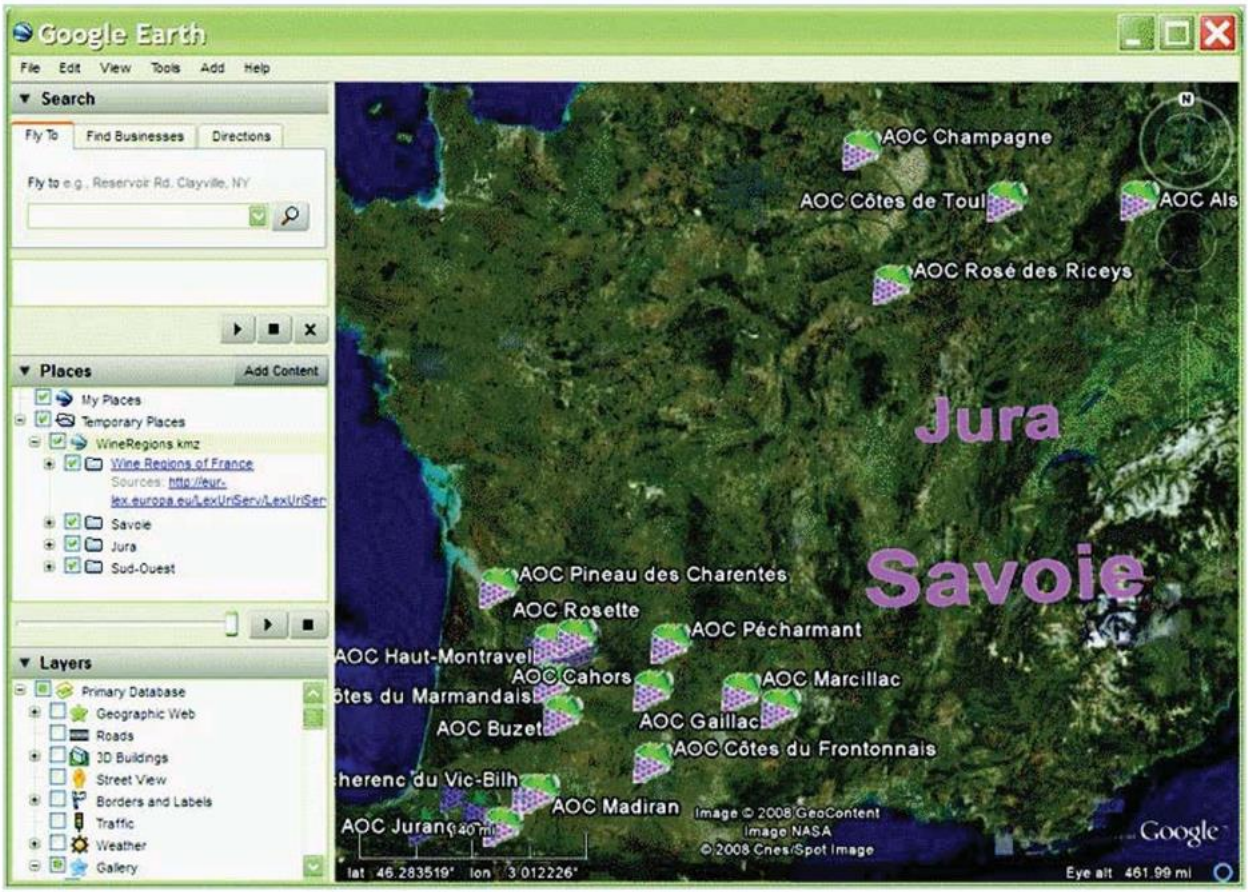


Рис. 8.7. По мере приближения точки обзора к различным частям Франции сначала появляется надпись с названием региона, а потом метки официальных винных апелясьонов. KML-регионы – это удобный способ избирательного представления данных в зависимости от текущей точки обзора (пример взят из работы Антонио Рочча Грака, http://bbs.keyhole.com/ubb/showthreaded.php? Number=303649)

Шаг 1: создание нескольких основных меток

В первой части примера мы определим KML-регион, охватывающий всю Францию. Этот регион влияет на видимость пяти меток, включенных в его папку. Эти метки становятся видимы, когда регион занимает не менее 256 квадратных пикселей на экране (при разрешении 1024 × 768 пикселей это составит примерно четверть экрана). Ниже приведен код этого региона, имеющего наибольший параллелепипед <LatLonAltBox> из всех встречающихся в рассматриваемом примере.

WineRegions.kmz

```
.
.
.
<Folder>
  <Region>
    <LatLonAltBox>
      <north>49.6936628301892</north>
      <south>42.8733871401169</south>
      <east>7.92566402137257</east>
```

```
      <west>-1.90120889491812</west>
    </LatLonAltBox>
    <Lod>
      <minLodPixels>256</minLodPixels>
      <maxLodPixels>-1</maxLodPixels>
    </Lod>
  </Region>
  <Placemark>
    <name>AOC Alsace</name>
    <styleUrl>#grapes</styleUrl>
    <Point>
      <coordinates>
        7.65924575839081,48.6604166508133,0
      </coordinates>
    </Point>
  </Placemark>
  <Placemark>
    <name>AOC Champagne</name>
    <styleUrl>#grapes</styleUrl>
    <Point>
      <coordinates>
        4.00300555110832,49.1547830422688,0
      </coordinates>
    </Point>
  </Placemark>
  <Placemark>
    <name>AOC Cфtes de Toul</name>
    <styleUrl>#grapes</styleUrl>
    <Point>
      <coordinates>
        5.88120562848141,48.6867220606491,0
      </coordinates>
    </Point>
  </Placemark>
  <Placemark>
    <name>AOC Pineau des Charentes</name>
    <styleUrl>#grapes</styleUrl>
    <Point>
      <coordinates>
        -0.522184443488129,45.3228752837428,0
      </coordinates>
    </Point>
  </Placemark>
  <Placemark>
    <name>AOC Rosй des Riceys</name>
    <styleUrl>#grapes</styleUrl>
    <Point>
      <coordinates>
        4.37071348058109,47.9806467702822,0
      </coordinates>
    </Point>
  </Placemark>
</Folder>
```

Совет

Вычислить географические границы <LatLonAltBox> можно очень просто: воспользуйтесь интерфейсом Google Earth для создания рельефного наложения <GroundOverlay>, размер которого совпадает с ограничивающим прямоугольником нужного вам региона (в данном случае Франции). Для наложения можно взять любое изображение, так как вас интересуют лишь линейные размеры. Затем скопируйте разметку <GroundOverlay> в текстовый файл и укажите ее размеры в качестве значений параметров <north>, <south>, <east> и <west> элемента <LatLonAltBox>. Если вас интересует также и высота, то добавьте в <LatLonAltBox> параметры <minAltitude> и <maxAltitude>.

Шаг 2: разнесение меток по регионам

В нашем примере используются метки, относящиеся к трем винодельческим районам Франции: Юра, Зюд-Ост и Савойя. Чтобы воспользоваться регионами, поместите каждый набор меток в папку с именем географической области. Затем определите регион, который охватывает эти метки, и задайте минимальную площадь (256 пикселей), при которой регион будет виден:

```
.
.
.
<Folder>
  <name>Savoie</name>
  <Region>
    <LatLonAltBox>
      <north>46.9636548058255</north>
      <south>44.6502126104117</south>
      <east>7.78667505979429</east>
      <west>4.24073167964791</west>
    </LatLonAltBox>
    <Lod>
      <minLodPixels>256</minLodPixels>
      <maxLodPixels>-1</maxLodPixels>
    </Lod>
  </Region>
  <Placemark>
    <name>AOC Bugey</name>
    <styleUrl>#grapes</styleUrl>
    <Point>
      <coordinates>5.70945945967014,45.7644687625724,0</coordinates>
    </Point>
  </Placemark>
  <Placemark>
    <name>AOC Crŭpy</name>
    <styleUrl>#grapes</styleUrl>
    <Point>
      <coordinates>6.31941116911709,46.295409776708,0</coordinates>
    </Point>
  </Placemark>
  <Placemark>
    <name>AOC Roussette du Bugey</name>
    <styleUrl>#grapes</styleUrl>
    <Point>
      <coordinates>5.27754272291607,46.1628519242604,0</coordinates>
    </Point>
  </Placemark>
.
.
.
```

На все метки в этой папке распространяется действие определенного в ней же региона.

Шаг 3: добавление слоя с названиями регионов

Чтобы помочь пользователю классифицировать данные и уменьшить количество меток на экране, видимых в каждый момент времени, можно добавить дополнительные слои, тем или иным способом фильтрующие данные. В нашем примере, когда точка обзора отдалена (и, следовательно, регион занимает меньше места на экране), появляется рельефное наложение, содержащее не сами метки, а розовые надписи с названиями географических областей. Названия районов Юра,

Зюд-Ост и Савойя – это изображения в формате PNG, содержащие розовые буквы на прозрачном фоне. Элементы <LatLonAltBox> для каждой надписи соответствуют элементам <LatLonAltBox> для региона меток, но диапазон LOD экранного наложения другой, поскольку надпись должна появляться до того, как станут видны сами метки. Точнее, в диапазоне LOD для Савойи левая граница равна 125 пикселей, а правая – 256 пикселей. Когда площадь наложения с метками становится больше максимума (256), соответствующий ей регион перестает быть активным и заменяется регионом с метками, для которого левая граница диапазона LOD как раз равна 256.

```
.
.
.
<GroundOverlay>
  <Region>
    <LatLonAltBox>
      <north>46.9636548058255</north>
      <south>44.6502126104117</south>
      <east>7.78667505979429</east>
      <west>4.24073167964791</west>
    </LatLonAltBox>
    <Lod>
      <minLodPixels>125</minLodPixels>
      <maxLodPixels>256</maxLodPixels>
    </Lod>
  </Region>
  <Icon>
    <href>labelImages/savoie.png</href>
  </Icon>
  <LatLonBox>
    <north>45.9514821841091</north>
    <south>44.9043980007738</south>
    <east>7.79996383091275</east>
    <west>3.58905953918123</west>
  </LatLonBox>
</GroundOverlay>
.
.
.
```

При работе с рельефными наложениями и регионами не забывайте учитывать реальное разрешение изображения. Если вы зададите для <maxLodPixels> значение, большее, чем разрешение изображения, то изображение растянется. Иногда этот эффект нежелателен, так как изображение может оказаться рваным или размытым.

Шаг 4: добавление эффекта затухания для надписей

Следующие две строчки разметки приводят к постепенному затуханию и проявлению надписей. Отредактируйте файл *doc.kml* в архиве *WineRegions.kmz* (предварительно замените расширение на *.zip*, а потом распакуйте архив), раскомментировав эти строки, и посмотрите, какой эффект оказывают эти элементы, вложенные в <Lod>:

```
<minFadeExtent>200</minFadeExtent>
<maxFadeExtent>200</maxFadeExtent>
```

Суперналожения

В этом разделе мы опишем, как создавать суперналожение – иерархию регионов и сетевых ссылок, которая служит для эффективной загрузки большого набора изображений. Суперналожения позволяют задействовать такие достоинства сетевых ссылок, как (1) умение определять, попадает ли данный регион в поле зрения, и (2) выяснять, достаточно ли велика площадь проекции некоторой области, чтобы показать ее на карте. Если оба условия выполнены, то регион активен, и по сетевой ссылке загружаются ассоциированные с ним данные. Если регион неактивен, то данные не загружаются.

Простой пример сетевой ссылки с ассоциированным регионом

Ниже приведен простой пример сетевой ссылки с ассоциированным регионом, на котором мы продемонстрируем несколько важных принципов, присущих этой технике. Параметр <viewRefreshMode> в родительском (инициирующем загрузку) файле *UkraineRegion.xml* равен `onRegion`, то есть загружать сетевую ссылку нужно лишь в случае, когда регион активен. Внутри большего региона (Украины) определен элемент <LatLonAltBox>. Когда этот регион становится активен, отображается ломаная вдоль границ, заданных в принадлежащем региону элементе <LatLonAltBox>. Данные для второй сетевой ссылки (*RomaniaRegion.xml*) загружаются лишь в том случае, когда активируется вложенный в нее регион. Если этот регион активен, то отображается описанная в файле *RomaniaRegion.xml* ломаная, соответствующая меньшему прямоугольнику <LatLonAltBox>.

Отметим, что в файле *RomaniaRegion.xml* определен также регион для ломаной, очерчивающей границы Румынии. В нашем примере этот регион можно было бы опустить, поскольку он совпадает с регионом, описанным в родительском файле (*UkraineRegion.xml*). Однако этот регион можно было бы заменить локальным, который отображался бы вместо региона из родительского файла. К тому же на случай, когда файл *RomaniaRegion.xml* загружается независимо, полезно иметь определенный в нем локальный регион.

UkraineRegion.kml

```
<?xml version="1.0" encoding="utf-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Document>
    <name>Вложенные регионы</name>
    <Region>
      <LatLonAltBox>
        <north>56.25</north>
        <south>45</south>
        <east>33.75</east>
        <west>22.5</west>
      </LatLonAltBox>
      <Lod>
        <minLodPixels>128</minLodPixels>
        <maxLodPixels>1024</maxLodPixels>
      </Lod>
```

```
    </Region>
    <Placemark>
      <name>UkraineRegion</name>
      <LineString>
        <tessellate>1</tessellate>
        <coordinates>
          22.5,45,0
          33.75,45,0
          33.75,56.25,0
          22.5,56.25,0
          22.5,45,0
        </coordinates>
      </LineString>
    </Placemark>
    <NetworkLink>
      <name>Сетевая ссылка на Румынию</name>
      <Region>
        <LatLonAltBox>
          <north>50.625</north>
          <south>45</south>
          <east>28.125</east>
          <west>22.5</west>
        </LatLonAltBox>
        <Lod>
          <minLodPixels>128</minLodPixels>
          <maxLodPixels>1024</maxLodPixels>
        </Lod>
      </Region>
      <Link>
        <href>RomaniaRegion.kml</href>
        <viewRefreshMode>onRegion</viewRefreshMode>
      </Link>
    </NetworkLink>
  </Document>
</kml>
```

RomaniaRegion.kml

```
<?xml version="1.0" encoding="utf-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Document>
    <name>Документ, описывающий Румынию</name>
    <Region>
      <LatLonAltBox>
        <north>50.625</north>
        <south>45</south>
        <east>28.125</east>
        <west>22.5</west>
      </LatLonAltBox>
      <Lod>
        <minLodPixels>128</minLodPixels>
        <maxLodPixels>1024</maxLodPixels>
      </Lod>
    </Region>
    <Placemark>
      <name>RomaniaRegion</name>
      <LineString>
        <tessellate>1</tessellate>
        <coordinates>
          22.5,45,0
          28.125,45,0
          28.125,50.625,0
          22.5,50.625,0
          22.5,45,0
        </coordinates>
      </LineString>
    </Placemark>
  </Document>
</kml>
```

Пример суперналожения

Чтобы посмотреть, как используется иерархия изображений в суперналожении, загрузите в Google Earth файл *Архивы Маунтин-Вью*. Это очень большой набор рельефных наложений, содержащий виды городка Маунтин-Вью, снятые в 1991 году (рис. 8.8). Поэкспериментируйте, приближая и отдаляя наложение, и посмотрите, как появляются и исчезают различные детали.

Чтобы загрузить Архивы Маунтин-Вью, зайдите на страницу http://kml-samples.googlecode.com/svn/trunk/kml_handbook и выберите файл mvdoqq.kml

Рис. 8.8. Пример использования сетевых ссылок с ассоциированными регионами для эффективной загрузки очень большого набора данных. В этом файле представлены старые фотографии городка Маунтин-Вью (на базе аэрофотосъемки 1991 года). Для отображения показанной на рисунке перспективы потребовалось загрузить только пять маленьких плиток (белыми линиями обозначены границы плиток)

Подготовка данных для суперналожения

В примере суперналожения оригинальное изображение Маунтин-Вью размером 7008 × 6720 пикселей разбито на сотни небольших рельефных наложений – *плиток*. Эти плитки организованы в пятиуровневую иерархию. Мы для простоты ограничимся более простой трехуровневой иерархией и набором из 21 наложе-

ния, но принципы остаются теми же. Отметим, что это лишь один из возможных подходов к созданию иерархий сетевых ссылок с ассоциированными регионами, есть и другие.

Для создания суперналожения вам понадобится:

1. Подготовить графический материал, разбив исходное изображение на более мелкие (рекомендованный размер одной плитки составляет 256 × 256 пикселей).
2. Создать KML-файлы, в которых описываются регионы, ссылки, сетевые ссылки и, в данном случае, файлы, содержащие рельефные наложения.

Примечание

В этом разделе детально описано, как подготовить иерархию сетевых ссылок с ассоциированными регионами вручную. Если вас серьезно интересует создание суперналожений, выберите какой-нибудь инструмент, умеющий создавать такие иерархии программно. Например, обратите внимание на утилиту Regionator и относящиеся к ней инструменты на сайте http://code.google.com/p/regionator.

Шаг 1: подготовка графического материала

Определитесь с размером плиток, на которые будет разбито исходное изображение. Google Earth загружает плитки, когда ассоциированный с ними регион становится активным. Мы остановимся на размере 256 × 256 пикселей – достаточно маленьком и управляемом.

1. Начните с исходного полноразмерного изображения. Разбейте его на четыре плитки, а каждую из получившихся – еще на четыре.
2. Продолжайте разбиение, пока не получатся плитки выбранного размера (в нашем случае 256 × 256 пикселей).

   Предположим, что размер исходного изображения составлял 1024 × 1024 пикселя. На рис. 8.9 показана получающаяся в результате разбиения иерархия.

Рис. 8.9. Для создания иерархии суперналожения разбейте исходное изображение на четыре части и продолжайте разбиение, пока не получатся плитки размером 256 × 256 пикселей

(Если размеры изображения не являются степенью двойки, то замостите его плитками, размеры которых – степень двойки, всюду, кроме, возможно, нижней строки и правого столбца.)

3. Затем масштабируйте каждую плитку в иерархии до выбранного размера (256 × 256 пикселей).

После масштабирования на плитках будет меньше деталей, но поскольку они ассоциированы с регионами, которые активируются при небольшом размере проекции, то пользователь эту утрату не заметит.

На рис. 8.10 показаны три уровня детализации для взятого в качестве примера изображения. Когда пользователь смотрит на регион с максимального расстояния, Google Earth показывает миниатюру. Эта миниатюра растягивается на весь прямоугольник <LatLonBox> (но так как площадь проекции мала – 256 квадратных пикселей, потеря визуальной информации не ощущается). Когда пользователь начинает приближать сцену, регион разбивается на четыре подрегиона. У каждой из четырех «плиток» тот же размер, что у миниатюры, но деталей гораздо больше.

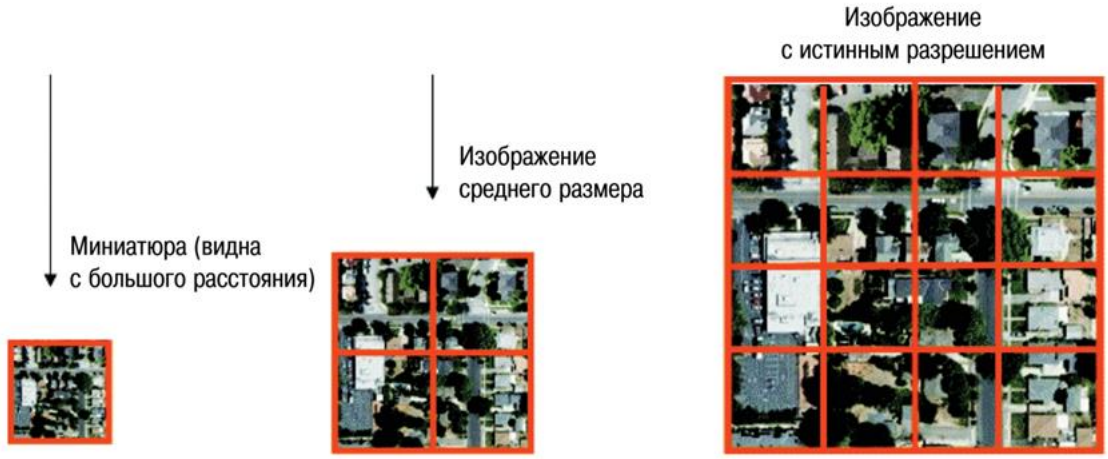


Рис. 8.10. После того как исходное изображение разбито на плитки уменьшающегося размера, перемасштабируйте все плитки к выбранному в самом начале размеру (например, 256 × 256 пикселей)

Если пользователь продолжит приближать область, то в поле зрения рано или поздно попадут части изображения с истинным разрешением. Когда это произойдет, зависит от диапазона LOD, указанного для данного региона. Отдаленные области по-прежнему показываются с меньшим уровнем детализации, поскольку их регионы еще не активировались. В Google Earth после загрузки примера *Архивы Маунтин-Вью* отметьте флажки против слова Boxes и меток A и B, для которых регионы обведены линиями, чтобы показать несколько отображаемых одновременно уровней иерархии.

Обратите внимание, что в этом примере для всех регионов значения <minLodPixels> и <maxLodPixels> одинаковы (на всех уровнях иерархии). Какой уровень иерархии загружать и какие плитки внутри региона показывать, определяет элемент <LatLonAltBox>.

Шаг 2: подготовка KML-файлов

Для каждого изображения подготовьте KML-файл, который ассоциирует элемент <GroundOverlay> с <Region> и <NetworkLink>. В каждом KML-файле из этого набора должны быть следующие элементы:

- <Region> (с <LatLonAltBox>, <minLodPixels> и <maxLodPixels>, чтобы геобраузер мог определить, активен ли данный регион);
- набор сетевых ссылок на дочерние файлы (плитки на следующем уровне иерархии);
- элемент <GroundOverlay> для данного <Region>.

Ниже приведен фрагмент KML-файла верхнего уровня для примера *Архивы Маунтин-Вью*. В нем <maxLodPixels> равно –1, то есть площадь проекции, для которой регион виден, не ограничена сверху. В противном случае иерархия вообще могла бы никогда не активироваться (например, если пролетаете мимо близко к поверхности Земли, но не пересекая порог, заданный в <minLodPixels>).

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://earth.google.com/kml/2.1">
  <NetworkLink>
    <name>SuperOverlay: MV DOQQ</name>
    <Region>
      <LatLonAltBox>
        <north>37.44140625</north>
        <south>37.265625</south>
        <east>-121.9921875</east>
        <west>-122.16796875</west>
      </LatLonAltBox>
      <Lod>
        <minLodPixels>128</minLodPixels>
        <maxLodPixels>-1</maxLodPixels>
      </Lod>
    </Region>
    <Link>
      <href>http://mw1.google.com/mw-earth-vectordb/kml-samples/mv-070501/1.kml</href>
      <viewRefreshMode>onRegion</viewRefreshMode>
    </Link>
  </NetworkLink>
```

Далее показан еще один файл из архива фотографий Маунтин-Вью (*179.kml*). Он содержит пять тегов <href>: четыре ссылаются на KML-файлы следующего уровня иерархии изображений, а пятый – на файл изображения, ассоциированного с элементом <GroundOverlay> для этой плитки.

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://earth.google.com/kml/2.1">
  <Document>
    <Region>
      <Lod>
        <minLodPixels>128</minLodPixels>
        <maxLodPixels>-1</maxLodPixels>
      </Lod>
      <LatLonBox>
        <north>37.430419921875</north><south>37.41943359375</south>
        <east>-122.091064453125</east><west>-122.10205078125</west>
      </LatLonBox>
    </Region>
    <NetworkLink>
      <name>001120</name>
      <Region>
```

```
      <Lod>
        <minLodPixels>128</minLodPixels><maxLodPixels>-1</maxLodPixels>
      </Lod>
      <LatLonAltBox>
        <north>37.430419921875</north><south>37.4249267578125</south>
        <east>-122.0965576171875</east><west>-122.10205078125</west>
      </LatLonAltBox>
    </Region>
    <Link>
      <href>180.kml</href>
      <viewRefreshMode>onRegion</viewRefreshMode>
    </Link>
  </NetworkLink>
  <NetworkLink>
    <name>001121</name>
    <Region>
      <Lod>
        <minLodPixels>128</minLodPixels><maxLodPixels>-1</maxLodPixels>
      </Lod>
      <LatLonAltBox>
        <north>37.430419921875</north><south>37.4249267578125</south>
        <east>-122.091064453125</east><west>-122.0965576171875</west>
      </LatLonAltBox>
    </Region>
    <Link>
      <href>185.kml</href>
      <viewRefreshMode>onRegion</viewRefreshMode>
    </Link>
  </NetworkLink>
  <NetworkLink>
    <name>001122</name>
    <Region>
      <Lod>
        <minLodPixels>128</minLodPixels><maxLodPixels>-1</maxLodPixels>
      </Lod>
      <LatLonAltBox>
        <north>37.4249267578125</north><south>37.41943359375</south>
        <east>-122.0965576171875</east><west>-122.10205078125</west>
      </LatLonAltBox>
    </Region>
    <Link>
      <href>190.kml</href>
      <viewRefreshMode>onRegion</viewRefreshMode>
    </Link>
  </NetworkLink>
  <NetworkLink>
    <name>001123</name>
    <Region>
      <Lod>
        <minLodPixels>128</minLodPixels><maxLodPixels>-1</maxLodPixels>
      </Lod>
      <LatLonAltBox>
        <north>37.4249267578125</north><south>37.41943359375</south>
        <east>-122.091064453125</east><west>-122.0965576171875</west>
      </LatLonAltBox>
    </Region>
    <Link>
      <href>195.kml</href>
      <viewRefreshMode>onRegion</viewRefreshMode>
    </Link>
  </NetworkLink>
  <GroundOverlay>
    <drawOrder>5</drawOrder>
    <Icon>
      <href>179.JPEG</href>
    </Icon>
    <LatLonBox>
      <north>37.430419921875</north><south>37.41943359375</south>
```

```
        <east>-122.091064453125</east><west>-122.10205078125</west>
      </LatLonBox>
    </GroundOverlay>
  </Document>
</kml>
```

Швейцарская система общественного транспорта

Сетевые ссылки с ассоциированными регионами полезны также для эффективного отображения больших наборов меток. На рис. 8.1 представлена проблема, а на рис. 8.11 – ее решение. Этот пример, относящийся к швейцарской системе общественного транспорта, написал Бент Хагемарк. Здесь отображается информация о железнодорожных станциях и перегонах между ними, но KML-регионы позволяют существенно уменьшить объем единовременно загружаемых и показываемых пользователю данных. Изучите этот пример более пристально, обращая внимание также на использование элементов <MultiGeometry> и <StyleMap>.

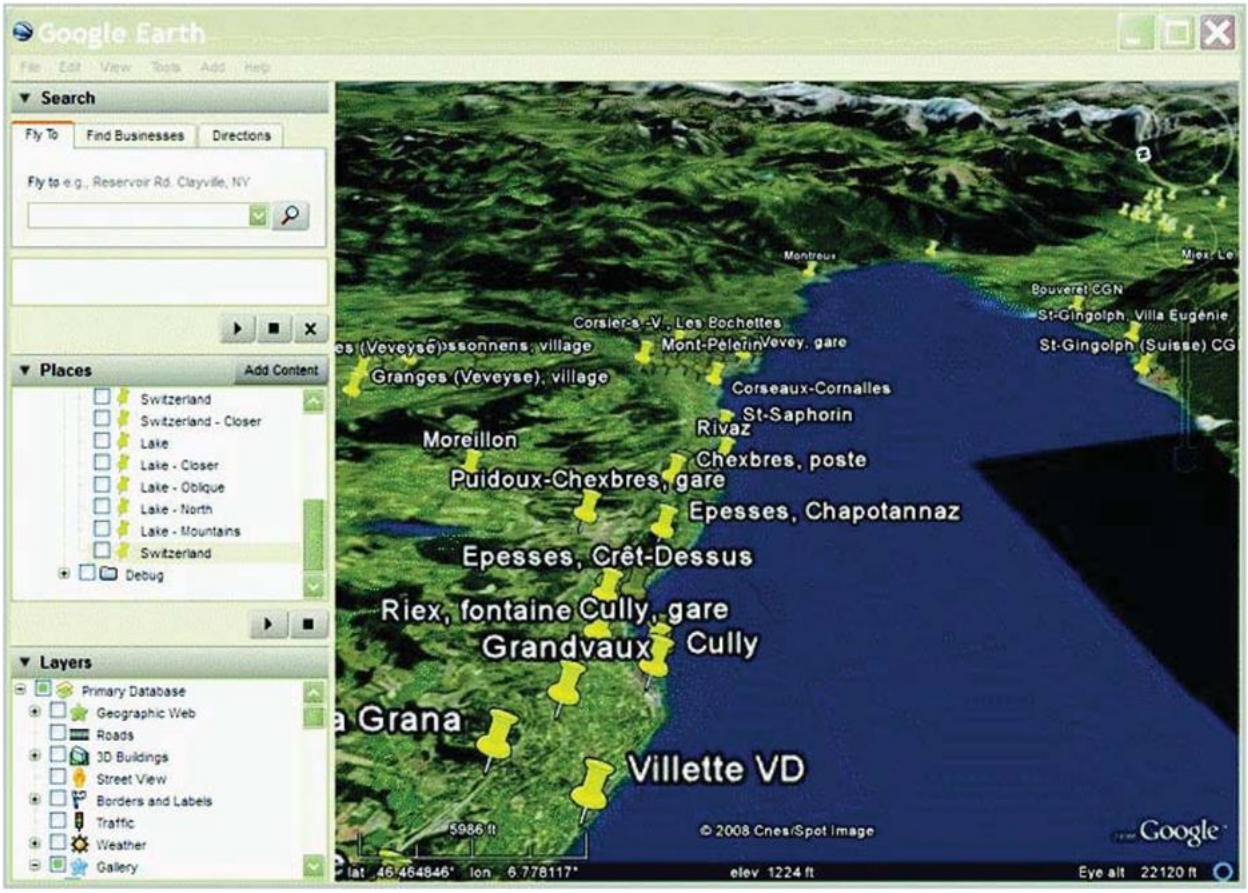


Рис. 8.11. С помощью сетевых ссылок с ассоциированными регионами этот KMZ-файл, описывающий швейцарскую систему общественного транспорта, отображает детальную информацию о станциях в районе того места, на которое смотрит пользователь. Когда пользователь меняет угол зрения, неинтересные ему станции уходят из поля зрения (пример написан Бентом Хагемарком, http://kml-samples.googlecode.com/svn/trunk/kml_handbook/ch-sbb.kml)

Чтобы загрузить пример, относящийся к швейцарской системе общественного транспорта, зайдите на страницу http://kml-samples.googlecode.com/svn/trunk/kml_handbook и выберите файл ch-sbb.kml

Расширенные данные

KML предлагает три способа добавить к детали данные нестандартного типа. Какой подход выбрать, зависит от особенностей данных, а также от того, как вы планируете их использовать в своей KML-презентации. Конкретно элемент <ExtendedData> предоставляет следующие механизмы:

- **Произвольные XML-данные** (простой) – позволяет *сохранять пользовательские данные* в KML-файле. Google Earth передает эти данные вместе с файлом, сохраняет их, но никак не использует.
- **Элемент <Data>** (простой) – позволяет включить в состав пользовательских данных, ассоциированных с элементом, производным от Feature (<NetworkLink>, <Placemark>, <GroundOverlay>, <PhotoOverlay>, <ScreenOverlay>, <Document> и <Folder>), *нетипизированные пары имя/значение*. По умолчанию эти пары отображаются в информационном окне. Кроме того, эту информацию можно использовать для подстановки вместо компонентов в элементе <text>, вложенном в <BalloonStyle> (см. раздел «Использование элемента <BalloonStyle> в качестве шаблона» ниже).
- **Элементы <Schema> и <SchemaData>** (более сложный) – позволяют ассоциировать с элементом Feature типизированные данные.

Примечание

Все три способа можно применять в одном файле. Если вы добавляете разнородные данные из различных источников, то, быть может, некоторые пользовательские данные стоит сделать типизированными (подход на основе Schema/SchemaData), тогда как другие – оставить нетипизированными (Data).

Какой подход выбрать?

Если вам необходимо только передать вместе с KML-файлом некоторые данные, а необходимости в их обработке геобраузером нет, то пользуйтесь механизмом сохранения произвольных XML-данных, который позволяет задать префикс пространства имен XML и затем передавать данные из этого пространства в текущем контексте KML. Дополнительную информацию см. в следующем разделе.

В большинстве случаев элемент <Data> предлагает самый простой и одновременно самый мощный механизм добавления нетипизированных данных в деталь KML. Этот метод относительно просто реализуется и обладает дополнительным достоинством: позволяет воспользоваться шаблоном <BalloonStyle>, который можно применить ко всем меткам в KML-файле. (Пример такого использования см. в разделе «Подстановка компонентов в элементах расширенных данных» ниже.)

Элементы <Schema> и <SchemaData> позволяют добавлять *типизированные пользовательские данные*. Они предназначены главным образом для пользовате-

лей, которым нужно передавать данные из ГИС (геоинформационных систем). Если у вас имеются типизированные данные, которые используются каким-нибудь внешним приложением, то для их добавления в Feature, вероятно, понадобятся элементы <Schema> и <SchemaData>. (Google Earth эту типизированную информацию никак не задействует, но другим программам она может быть необходима.) Обычному пользователю технические аспекты этого механизма, скорее всего, неинтересны. Как и <Data>, элемент <SchemaData> позволяет использовать BalloonStyle в качестве шаблона.

Добавление произвольных XML-данных в элемент Feature

Простейший способ добавить пользовательские данные в элемент Feature – включить их в качестве значения элемента <ExtendedData>. Google Earth сохраняет такие данные, но не обрабатывает их.

Добавляемые подобным образом данные должны включать *префикс пространства имен* (`xmlns:prefix="namespace"`), как показано в примере *SimpleUserData.kml*. Этот префикс можно указать как в элементе <kml>, так и в элементе <ExtendedData>. Префиксом внешнего пространства имен необходимо помечать все экземпляры пользовательских данных, как показано в примере ниже, где в качестве префикса выбрана строка «camp».

SimpleUserData.kml

```
<?xml version="1.0" encoding="utf-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2"
  xmlns:camp="http://example.com/camp">
  <!-- Примечание: адрес http://example.com/camp не существует, но указывать его
  необходимо, иначе возникнет ошибка при контроле XSD-схемы ->
  <Document>
    <name>Добавление произвольных данных в KML</name>
    <Placemark>
      <name>CampsiteData</name>
      <!-- Для ссылки на импортированную схему необходим префикс пространства имен ->
      <ExtendedData>
        <camp:number>14</camp:number>
        <camp:parkingSpaces>2</camp:parkingSpaces>
        <camp:tentSites>4</camp:tentSites>
      </ExtendedData>
      <Point>
        <coordinates>-114.041,53.7199</coordinates>
      </Point>
    </Placemark>
  </Document>
</kml>
```

Добавление нетипизированных пар имя/значение

Элемент <Data> предлагает простой, но мощный механизм добавления нетипизированных пар имя/значение в любой элемент KML, производный от Feature.

Синтаксис элемента <Data>

Элемент <Data> является потомком <ExtendedData>. Ниже описывается синтаксис последнего:

```
<ExtendedData>
  <Data name="string">
    <displayName>...</displayName>
    <value>...</value>
  </Data>
    .
    .
    .
</ExtendedData>
```

<Data name=»string»>

Атрибут `name` уникально идентифицирует элемент данных. Имя используется для идентификации пары имя/значение в пределах одного KML-файла. (Для вывода информации в геобраузере служит параметр <displayName>.)

<displayName>

Задает понятное пользователю имя элемента данных для отображения в информационном окне. Может содержать фрагмент CDATA, включающий пробелы, URL и гиперссылки.

<value>

Задает значение элемента данных (в виде строки).

В примере *NameValuePairs.kml* показана метка, содержащая пользовательские данные о нескольких лунках на поле для гольфа. На рис. 8.12 видно, как пары имя/значение отображаются во всплывающем окне с использованием форматирования по умолчанию. Так как параметр <displayName> не задан, то в качестве имени выступает значение атрибута `name`.

Рис. 8.12. По умолчанию пары имя/значение, содержащиеся в элементе <Data>, отображаются в информационном окне в виде таблицы

NameValuePairs.kml

```
<?xml version="1.0" encoding="utf-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Document>
    <name>Пример: поле для гольфа</name>
    <Placemark>
      <name>Clubhouse</name>
      <ExtendedData>
        <Data name="holeNumber">
          <value>1</value>
        </Data>
        <Data name="holeYardage">
          <value>234</value>
        </Data>
        <Data name="holePar">
          <value>4</value>
        </Data>
      </ExtendedData>
      <Point>
        <coordinates>-111.956,33.5043,0</coordinates>
      </Point>
    </Placemark>
    <Placemark>
      <name>By the lake</name>
      <ExtendedData>
        <Data name="holeNumber">
          <value>5</value>
        </Data>
        <Data name="holeYardage">
          <value>523</value>
        </Data>
        <Data name="holePar">
          <value>5</value>
        </Data>
      </ExtendedData>
      <Point>
        <coordinates>-111.95,33.5024,0</coordinates>
      </Point>
    </Placemark>
  </Document>
</kml>
```

Использование элемента <BalloonStyle> в качестве шаблона

В главе 4 вы узнали, что можно определить элемент <Style> один раз и присвоить ему идентификатор `id`. Затем на этот стиль можно ссылаться из разных мест данного (или какого-то другого) KML-файла с помощью элемента <styleUrl>. Определенный таким образом стиль называется *разделяемым*. В главе 2 было рассказано о том, что элемент <text>, вложенный в <BalloonStyle>, поддерживает *подстановку компонентов*. Вместо каждого экземпляра компонента можно подставить некоторое значение. Для стандартных компонентов определены следующие подстановки:

$[name]
: Подставляется значение элемента <name> из <Placemark>.

$[description]
: Подставляется значение элемента <description> из <Placemark>.

$[address]
Подставляется значение элемента <address> из <Placemark>.
$[id]
Подставляется значение атрибута id из <Placemark>.
$[Snippet]
Подставляется значение элемента <Snippet> из <Placemark>.
$[geDirections]
Подставляется пара ссылок на маршруты Сюда/Отсюда.

Подстановка компонентов для элементов расширенных данных

Программа Google Earth поддерживает также подстановку некоторых компонентов в элементах расширенных данных, встречающихся внутри элемента <text>, вложенного в <BalloonStyle>. Внутри элемента <text> можно ссылаться на следующие компоненты:

Переменная	Заменяется на	Пример
$[nameAttributeOfDataElement]	Содержимое элемента <value>	1, 234, 4
$[nameAttribute/displayName]	Содержимое элемента <displayName>	Номер лунки, расстояние до лунки в ярдах

В примере BalloonStyleTemplate.kml создается шаблон <BalloonStyle> для предыдущего примера с полем для гольфа. В информационное окно каждой метки Google Earth подставляет имя этой метки, после чего выводит информацию о номере лунки, паре[1] и расстоянии до лунки в ярдах (рис. 8.13). Шаблоны стилей информационных окон и подстановка компонентов – это очень полезный и экономный способ репликации сложных стилей для большого количества меток. Обязательно опробуйте этот метод, если в ваших информационных окнах отображается много данных.

BalloonStyleTemplate.kml

```
<?xml version="1.0" encoding="utf-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Document>
    <name>Data+BalloonStyle</name>
    <Style id="golf-balloon-style">
      <BalloonStyle>
        <text>
          <![CDATA[
          This is $[name]
          This is hole $[holeNumber]
          The par for this hole is $[holePar]
          The yardage is $[holeYardage]
```

[1] Пар – в гольфе: установленное количество ударов по мячу, необходимое хорошему игроку для проведения мяча в лунку – *Прим. перев.*

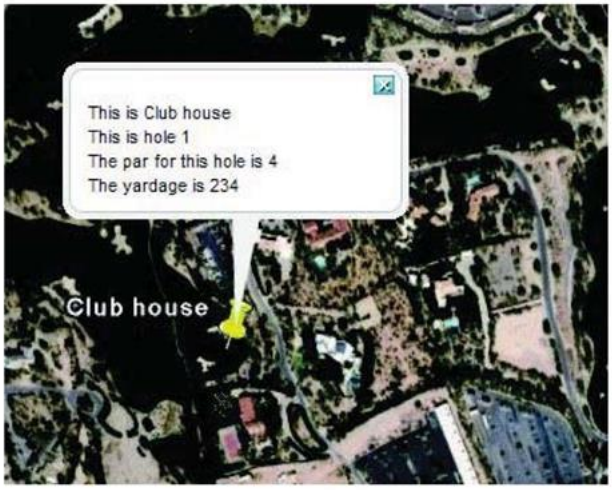


Рис. 8.13. Шаблоны стилей информационных окон – эффективный способ представить пользовательские данные при наличии большого количества меток. В данном примере применяется механизм подстановки компонентов для вывода уникальных значений в каждое информационное окно

```
          ]]>
        </text>
      </BalloonStyle>
    </Style>
    <!-- Пример разделяемого стиля
         Две метки пользуются одним и тем же шаблоном информационного окна ->
    <Placemark>
      <name>Club house</name>
      <styleUrl>#golf-balloon-style</styleUrl>
      <ExtendedData>
        <Data name="holeNumber">
          <value>1</value>
        </Data>
        <Data name="holeYardage">
          <value>234</value>
        </Data>
        <Data name="holePar">
          <value>4</value>
        </Data>
      </ExtendedData>
      <Point>
        <coordinates>-111.956,33.5043</coordinates>
      </Point>
    </Placemark>
    <Placemark>
      <name>By the lake</name>
      <styleUrl>#golf-balloon-style</styleUrl>
      <ExtendedData>
        <Data name="holeNumber">
          <value>5</value>
        </Data>
        <Data name="holeYardage">
          <value>523</value>
        </Data>
        <Data name="holePar">
          <value>5</value>
        </Data>
      </ExtendedData>
      <Point>
        <coordinates>-111.95,33.5024</coordinates>
      </Point>
    </Placemark>
  </Document>
</kml>
```

Добавление типизированных данных в элемент Feature

Элементы <Schema> и <SchemaData> позволяют добавлять типизированные данные в любой элемент, производный от Feature. Некоторым ГИС и другим приложениям этот механизм необходим, так как они имеют дело с типизированными данными.

Для добавления нестандартного типа в деталь KML необходимо сделать две вещи:

1. Создать элемент <Schema>, содержащий объявление нового типа.
2. Создать экземпляры нового типа с помощью элемента <SchemaData>.

Элемент <Schema> должен быть потомком <Document>. Элемент <ExtendedData> является потомком элемента Feature и содержит нестандартные данные.

Синтаксис элемента <Schema>

Элемент <Schema> содержит один или несколько элементов <SimpleField>. Внутри <SimpleField> элемент <Schema> объявляет тип и имя нестандартного поля. Там же может быть необязательный элемент <displayName> (понятное пользователю имя поля, которое может содержать пробелы и знаки препинания и отображается в геобраузере).

Элемент <Document> может содержать нуль или более элементов <Schema>. Синтаксис элемента <Schema> следующий:

```
<Schema name="string" id="ID">
  <SimpleField type="string" name="string">
    <displayName>...</displayName>
    <value>...</value>
  </SimpleField>
</Schema>
```

<Schema name=" " id=" ">

Атрибут `name` необязателен и применяется при подстановке компонентов, чтобы квалифицировать пространство имен. Атрибут `id` обязателен и должен быть уникален в пределах одного KML-документа. По атрибуту `id` производится ссылка на схему при создании экземпляров данного типа (в элементе <SchemaData>).

<SimpleField type=" " name=" ">

В элементе <SimpleField> обязательно должны присутствовать атрибуты `type` и `name`, в противном случае он игнорируется. Тип может принимать следующие значения: `bool`, `double`, `float`, `int`, `uint`, `short`, `ushort`, `string`.

Атрибут `name` применяется при подстановке компонентов в элементе <text>, вложенном в <BalloonStyle>.

<displayName>

Задает необязательное имя поля, которое отображается геобраузером. Для экранирования стандартной HTML-разметки пользуйтесь элементом CDATA.

Синтаксис элемента <SchemaData>

Элемент <SchemaData> должен быть потомком <ExtendedData>, синтаксис которого определяется следующим образом:

```
<ExtendedData>
    .
    .
    .
  <SchemaData schemaUrl="anyURI">
    <SimpleData name="">...</SimpleData> <!-- string ->
  </SchemaData>
  <namespace_prefix:other>...</namespace_prefix:other>
</ExtendedData>
```

<SchemaData schemaUrl=" ">

Этот элемент используется в сочетании с <Schema> для добавления данных нестандартного типа в любой элемент, производный от Feature. Элемент <Schema> (идентифицируемый атрибутом `schemaUrl`) объявляет тип данных. Сами объекты данных («экземпляры» нестандартного типа) определяются с помощью элемента <SchemaData>.

Атрибут `schemaUrl` ссылается по идентификатору `id` на элемент <Schema>, в котором объявлен тип. Точно так же элемент <styleUrl> ссылается по идентификатору на ранее объявленный стиль <Style>.

Значением `schemaUrl` может быть полный URL, ссылка на идентификатор схемы, определенной во внешнем KML-файле, или ссылка на идентификатор схемы, определенной в том же файле. Допустимы все следующие спецификации:

```
schemaUrl="http://host.com/PlacesIHaveLived.kml#my-schema-id">
schemaUrl="AnotherFile.kml#my-schema-id">
schemaUrl="#my-schema-id">                    <!-- тот же самый файл ->
```

Если значением `schemaUrl` является HTTP-адрес, то геобраузер скачивает указанный файл из Сети. Для каждой метки может быть задан только один экземпляр данного типа, определенного пользователем.

<SimpleData name=" ">

Присваивает значение полю нестандартного типа с именем, определяемым атрибутом `name`. Тип и имя этого поля должны быть объявлены в элементе <Schema>.

Пример данных нестандартного типа

В файле *TrailSchemaData.kml* объявлен пользовательский тип `ScenicVista`. Он содержит три поля: `TrailHeadName`, `TrailLength` и `ElevationGain`. Поле `TrailHeadName` имеет тип `string`, поле `TrailLength` – тип `double`, а поле `ElevationGain` – тип `int`. Метка с именем `Easy trail` содержит экземпляр этого типа, в котором `TrailHeadName` равно «Pi in the sky», `TrailLength` равно 3.14159, а `ElevationGain` равно 10.

TrailSchemaData.kml

```
<?xml version="1.0" encoding="utf-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Document>
    <name>ExtendedData+SchemaData</name>
    <open>1</open>

    <!-- Объявляем тип "TrailHeadType" с тремя полями ->
    <Schema id="TrailHeadTypeId" name="TrailHeadType">
      <SimpleField name="TrailHeadName" type="string">
        <displayName><![CDATA[<b>Trail Head Name</b>]]></displayName>
      </SimpleField>
      <SimpleField name="TrailLength" type="double">
        <displayName><![CDATA[<i>Length in miles</i>]]></displayName>
      </SimpleField>
      <SimpleField name="ElevationGain" type="int">
        <displayName><![CDATA[<i>Change in altitude</i>]]></displayName>
      </SimpleField>
    </Schema>

<!--
    <TrailHeadType>
      <TrailHeadName>...</TrailHeadName>
      <TrailLength>...</TrailLength>
      <ElevationGain>...</ElevationGain>
    <TrailHeadType>
->

    <!-- Создаем несколько меток с данными типа TrailHeadType ->
    <Placemark>
      <name>Easy trail</name>
      <ExtendedData>
        <SchemaData schemaUrl="#TrailHeadTypeId">
          <SimpleData name="TrailHeadName">Pi in the sky</SimpleData>
          <SimpleData name="TrailLength">3.14159</SimpleData>
          <SimpleData name="ElevationGain">10</SimpleData>
        </SchemaData>
      </ExtendedData>
      <Point>
        <coordinates>-122,37.002</coordinates>
      </Point>
    </Placemark>
    <Placemark>
      <name>Difficult trail</name>
      <ExtendedData>
        <SchemaData schemaUrl="#TrailHeadTypeId">
          <SimpleData name="TrailHeadName">Mount Everest</SimpleData>
          <SimpleData name="TrailLength">347.45</SimpleData>
          <SimpleData name="ElevationGain">10000</SimpleData>
        </SchemaData>
      </ExtendedData>
      <Point>
        <coordinates>-121.998,37.0078</coordinates>
      </Point>
    </Placemark>
  </Document>
</kml>
```

На рис. 8.14 показано информационное окно для одной из этих меток. Используется подразумеваемый по умолчанию стиль.

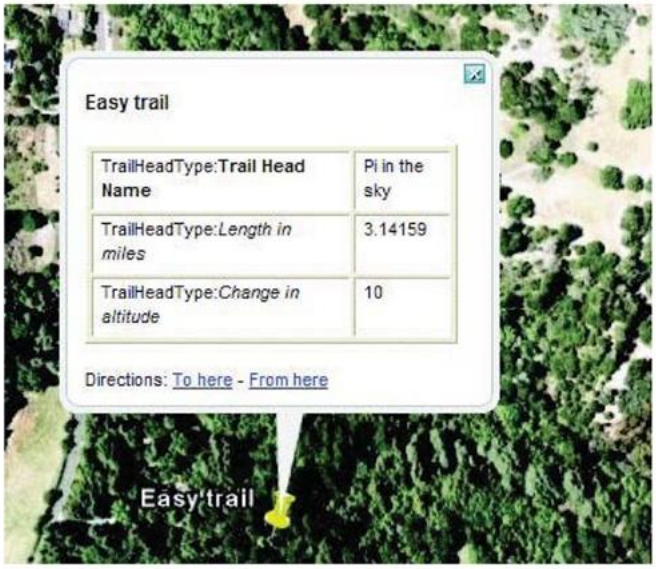


Рис. 8.14. По умолчанию данные нестандартного типа отображаются в информационном окне в табличной форме: в левом столбце – имя, в правом – значение

Пример подстановки компонентов

Механизм Schema/SchemaData поддерживает также подстановку компонентов в элементе <text>, вложенном в <BalloonStyle>. Элемент <displayName> внутри <Schema> позволяет задать понятное пользователю имя нестандартного типа данных. Изменение этой строки в элементе <Schema> отразится на всех элементах Feature, содержащих данный тип.

В элементе <text>, вложенном в <BalloonStyle>, для ссылки на <displayName> используется следующий синтаксис:

```
$[TYPENAME/TYPEFIELD/displayName]
```

где

`TYPENAME` – атрибут `name` в теге <Schema>;
`TYPEFIELD` – атрибут `name` в теге <SimpleField>;
`displayName` – элемент <displayName>, вложенный в <SimpleField>.

В файле *SchemaDataAndBalloonStyle.kml* к предыдущему примеру добавлен шаблон BalloonStyle. Как и в примере с полем для гольфа, шаблон создается из элементов <Style>, <BalloonStyle> и <text> и располагается в начале документа. Все метки в этом файле ссылаются на данный шаблон стиля.

На рис. 8.15 выводятся те же данные, что в предыдущем примере (рис. 8.14), но по созданному нами шаблону <BalloonStyle>.

SchemaDataAndBalloonStyle.kml

```
<?xml version="1.0" encoding="utf-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Document>
    <name>ExtendedData+SchemaData</name>
    <open>1</open>
    <!-- Создаем шаблон информационного окна, который ссылается на определенный пользователем тип -->
    <Style id="trailhead-balloon-template">
      <BalloonStyle>
        <text>
        <![CDATA[
          <h2>My favorite trails!</h2>
          <br/><br/>
          The $[TrailHeadType/TrailHeadName/displayName] is
          <i>$[TrailHeadType/TrailHeadName]</i>.
          The trail is $[TrailHeadType/TrailLength] miles.<br/>
          The climb is $[TrailHeadType/ElevationGain] meters.<br/><br/>
        ]]>
        </text>
      </BalloonStyle>
    </Style>

    <!-- Объявляем тип "TrailHeadType" с тремя полями -->
    <Schema id="TrailHeadTypeId" name="TrailHeadType">
      <SimpleField name="TrailHeadName" type="string">
        <displayName><![CDATA[<b>Trail Head Name</b>]]></displayName>
      </SimpleField>
      <SimpleField name="TrailLength" type="double">
        <displayName><![CDATA[<i>The length in miles</i>]]></displayName>
      </SimpleField>
      <SimpleField name="ElevationGain" type="int">
        <displayName><![CDATA[<i>change in altitude</i>]]></displayName>
      </SimpleField>
    </Schema>

    <!-- Создаем несколько меток с данными типа TrailHeadType -->
    <Placemark>
      <name>Easy trail</name>
      <styleUrl>#trailhead-balloon-template</styleUrl>
      <ExtendedData>
        <SchemaData schemaUrl="#TrailHeadTypeId">
          <SimpleData name="TrailHeadName">Pi in the sky</SimpleData>
          <SimpleData name="TrailLength">3.14159</SimpleData>
          <SimpleData name="ElevationGain">10</SimpleData>
        </SchemaData>
      </ExtendedData>
      <Point>
        <coordinates>-122,37.002</coordinates>
      </Point>
    </Placemark>
    <Placemark>
      <name>Difficult trail</name>
      <styleUrl>#trailhead-balloon-template</styleUrl>
      <ExtendedData>
        <SchemaData schemaUrl="#TrailHeadTypeId">
          <SimpleData name="TrailHeadName">Mount Everest</SimpleData>
          <SimpleData name="TrailLength">347.45</SimpleData>
          <SimpleData name="ElevationGain">10000</SimpleData>
        </SchemaData>
      </ExtendedData>
      <Point>
        <coordinates>-121.998,37.0078</coordinates>
      </Point>
    </Placemark>
  </Document>
</kml>
```

Что дальше?

Вот мы и дошли до конца той части этой книги, которую можно назвать «учебником». Приложение А «Справочное руководство по KML» будет вашим постоянным спутником при создании новых KML-файлов и совершенствовании ранее созданных. Настало время экспериментировать, пробовать и открывать для себя всю мощь KML. Не забывайте о быстро растущем сообществе пользователей KML – ищите вдохновляющие примеры в Сети. А потом создавайте свои творения и делитесь ими!

Приложение А
Справочное руководство по KML

В этой части книги содержится справочник по всем типам и элементам языка KML в алфавитном порядке. На рис. А-1 показано дерево наследования элементов KML. Элементы, находящиеся в нем правее, являются расширениями элементов слева от себя (наследуют им). Например, элемент Placemark – частный случай Feature. Он содержит все элементы, которые определены для Feature, и добавляет к ним собственные. В разделах, посвященных синтаксису элементов KML, указывается, какие дочерние элементы (потомки) унаследованы от родительских классов.

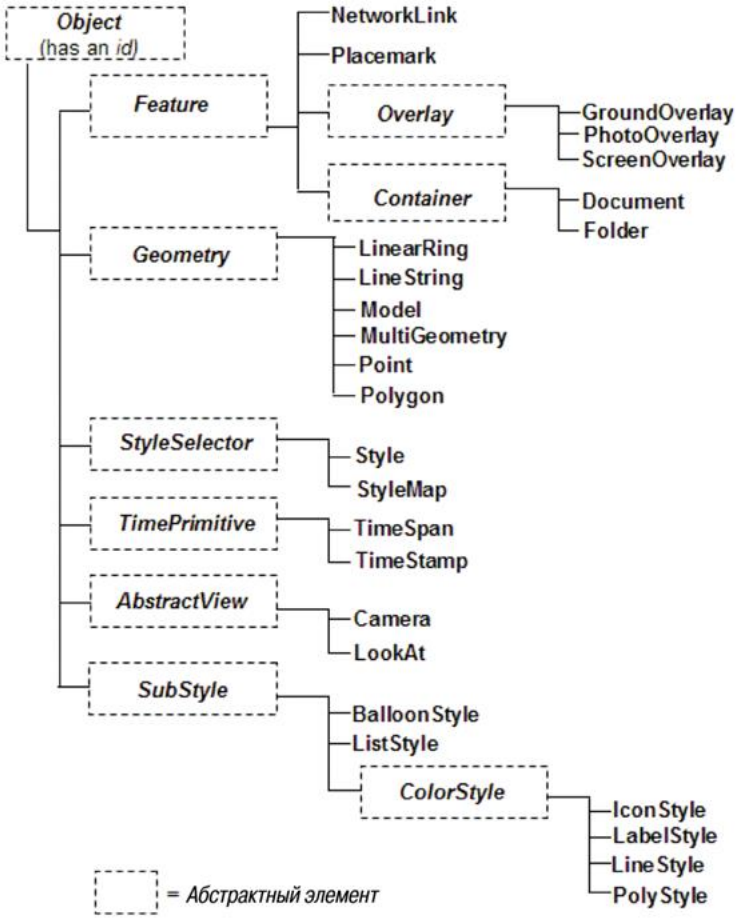


Рис. А-1. Дерево элементов KML. Элементы, находящиеся правее, наследуют потомков от родительских элементов слева от себя

Абстрактные элементы

На приведенной выше диаграмме классов также показано, какие элементы являются *абстрактными* (они обведены штриховой линией). Создать абстрактный элемент непосредственно нельзя. Это лишь эффективный способ описывать составные элементы, обладающие общими потомками. Каждый абстрактный элемент определяет некую *группу подстановок*. Например, любой элемент, производный от абстрактного элемента <Geometry>, можно подставить вместо любого другого элемента, также производного от <Geometry>. У производных элементов могут быть специфические задачи, но общие принципы использования всех элементов, наследующих одному и тому же предку, одинаковы.

Простые и составные элементы

Составной элемент может содержать внутри себя другие элементы. *Простой элемент* содержит строку или число в качестве значения и может обладать атрибутами, но других элементов включать не может. Имена составных элементов в KML начинаются с заглавной буквы, а имена простых – со строчной. В этом приложении приведен алфавитный перечень всех составных элементов в KML. Часто составной элемент называют *родителем* входящих в него простых элементов, а простой элемент – *потомком* своего составного родителя.

Основные синтаксические правила KML

С точки зрения грамматики и формата файла KML является диалектом XML и подчиняется общим правилам XML. Точнее:

- регистр букв имеет значение. Имена элементов должны записываться точно так, как указано в настоящем руководстве, с точностью до регистра;
- порядок следования имеет значение. Потомки составного элемента должны располагаться точно в таком порядке, как описано в настоящем руководстве;
- элементы-потомки могут быть вложены только в разрешенный для них родительский элемент. Для каждого составного элемента разрешенные потомки перечислены в разделе, посвященном синтаксису. (В подразделах «Содержит» и «Содержится в» приведена дополнительная информация о допустимых связях родитель/потомок.)

В программе Google Earth можно включить режим контроля, в котором выдаются предупреждения о стоящих не на месте элементах и других ошибках KML.

Заголовок

В настоящем руководстве описывается версия KML 2.2. В начале любого KML-файла должны располагаться такие строки:

```
<?xml version="1.0" encoding="utf-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
```

Если файл содержит небесные, а не земные координаты, то необходимо включить также атрибут `hint="target=sky"`, как показано ниже:

```
<kml xmlns="http://www.opengis.net/kml/2.2" hint="target=sky>
```

Значения по умолчанию

У каждого элемента KML есть значение по умолчанию. Если в составном элементе опущен какой-нибудь потомок, то предполагается, что он принимает значение по умолчанию. Обязательных элементов в KML 2.2 нет.

Как пользоваться настоящим руководством

Это руководство дополняет первые восемь глав книги, в которых вы научились вручную создавать и редактировать KML-документы. Разнообразные детали и нюансы намеренно исключены из описаний, чтобы текст оставался компактным и удобным для повседневного использования. Описания синтаксиса можно копировать непосредственно в KML-код, подставляя необходимые значения вместо подразумеваемых по умолчанию. В синтаксических разделах для каждого элемента указаны также KML-типы значений. В случае перечисляемых значений указываются все допустимые варианты.

Схема KML

Схема KML (*ogckml22.xsd*) находится по адресу www.opengeospatial.org. В скором времени ожидается расширение KML, поэтому заглядывайте на сайты этой книги и консорциума Open Geospatial Consortium, где публикуется самая свежая информация. Будущие версии KML гарантированно будут совместимы с предыдущими.

Не забывайте тестировать документы в разных геобраузерах

Некоторые геобраузеры и картографические приложения поддерживают только подмножество описанных здесь элементов KML. В настоящей книге документирована полная спецификация KML 2.2 в том виде, в котором она изложена в стандарте Open Geospatial Consortium (на сайте www.opengeospatial.org). Не забывайте тестировать свои KML-документы во всех геобраузерах и приложениях, в которых они могут использоваться.

Типы KML

В языке KML используются общие типы XML, такие как `boolean`, `string`, `double`, `float` и `int`. Помимо них, определен еще ряд типов. В следующей таблице перечислены некоторые наиболее употребительные типы и ссылки на элементы, в которых они используются.

Тип	Значение	Пример использования
altitudeModeEnum	clampToGround, relativeToGround, absolute	См. <LookAt> и <Region>
angle90	значение от –90.0 до 90.0	См. <latitude> в <Model>
angle180	значение от –180 до 180	См. <longitude> в <Model>
anglepos180	значение от 0.0 до 180.0	См. <tilt> в <LookAt>
angle360	значение от –360.0 до 360.0	См. <heading> в <Orientation>
color	шестнадцатеричное: *aabbggrr*	См. любой элемент, расширяющий ColorStyle
colorModeEnum	normal, random	См. любой элемент, расширяющий ColorStyle
dateTime	*dateTime*, *date*, *gYearMonth*, *gYear*	См. <TimeSpan> и <TimeStamp>
displayModeEnum	default, hide	См. <BalloonStyle>
gridOriginEnum	lowerLeft, upperLeft	См. <PhotoOverlay>
itemIconStateEnum	open, closed, error, fetching0, fetching1, fetching2	См. <state> в <ItemIcon>
listItemTypeEnum	radioFolder, check, checkHideChildren	См. <ListItemType> в <ListStyle>
refreshModeEnum	onChange, onInterval, onExpire	См. <Link>
shapeEnum	rectangle, cylinder, sphere	См. <PhotoOverlay>
styleStateEnum	normal, highlight	См. <StyleMap>
unitsEnum	fraction, pixels, insetPixels	См. <hotSpot> в <IconStyle>, <ScreenOverlay>
vec2	x=double xunits=*kml:unitsEnum* y=double yunits=*kml:unitsEnum*	См. <hotSpot> в <IconStyle>, <ScreenOverlay>
viewRefreshEnum	never, onRequest, onStop, onRegion	См. <Link>

AbstractView

Раздел 14.1 спецификации OGC KML 07-147

```
<!-- AbstractView id="ID" -> <!-- Camera, LookAt ->
<!-- /AbstractView ->
```

Описание

Это абстрактный элемент, который нельзя использовать в KML-файле непосредственно. Он является базовым для элементов <Camera> и <LookAt>.

<BalloonStyle>

```
<BalloonStyle id="ID">                  <!-- kml:color -->
  <bgColor>ffffffff</bgColor>           <!-- kml:color -->
  <textColor>ff000000</textColor>       <!-- kml:color -->
  <text>...</text>                      <!-- string -->
  <displayMode>default</displayMode>    <!-- kml:displayModeEnum -->
</BalloonStyle>
```

Описание

Определяет форматирование информационного окна с описанием. Если элемент <bgColor> задан, то он интерпретируется как цвет фона окна. См. главу 2 «Метки и информационные окна».

Элементы, специфичные для BalloonStyle

<bgColor>

Цвет фона информационного окна. Цвет и прозрачность (альфа-канал) записываются в шестнадцатеричном виде *aabbggrr*, где *aa* – альфа-канал (от 00 до ff), *bb* – синяя составляющая (от 00 до ff), *gg* – зеленая составляющая (от 00 до ff), *rr* – красная составляющая (от 00 до ff). Если альфа-канал равен 00, то цвет полностью прозрачен, а если ff – то полностью непрозрачен.

<textColor>

Цвет текста в информационном окне.

<text>

Текст, отображаемый в информационном окне. Внутри тега <text> могут находиться компоненты в следующем формате, позволяющем ссылаться на дочерний элемент типа Feature: `$[name]`, `$[description]`, `$[address]`, `$[id]`, `$[Snippet]`, `$[geDirections]` (маршрут). Геобраузер ищет в текущем элементе Feature элемент, соответствующий компоненту, и подставляет информацию из него в информационное окно. Чтобы предотвратить появление маршрута во всплывающем окне, включите элемент <text> с каким-нибудь содержимым или с компонентом `$[description]`, вместо которого будет подставлено содержимое элемента <description> из детали Feature. О том, как подставлять во всплывающее окно нестандартные компоненты, см. описание элемента <ExtendedData> (и раздел «Подстановка компонентов для элементов расширенных данных» в главе 8).

<displayMode>

Если равен `default`, то геобраузер отображает в информационном окне информацию из элемента <text>. Если же режим <displayMode> равен `hide`, то информационное окно вообще не появляется. (В программе Google Earth щелчок по значку в списке, соответствующем метке, для которой элемент <displayMode> равен `hide`, заставляет Google Earth «перелететь» к этой метке.).

Содержится в

<Style>

<Camera>

```
<Camera id="ID">
  <longitude>0.0</longitude>                    <!-- kml:angle180 -->
  <latitude>0.0</latitude>                      <!-- kml:angle90 -->
  <altitude>0.0</altitude>                      <!-- double -->
  <heading>0.0</heading>                        <!-- kml:angle360 -->
  <tilt>0.0</tilt>                              <!-- kml:anglepos180 -->
  <roll>0.0</roll>                              <!-- kml:angle180 -->
  <altitudeMode>clampToGround</altitudeMode>    <!-- kml:altitudeModeEnum -->
</Camera>
```

Описание

Определяет виртуальную камеру, обозревающую сцену. Этот элемент описывает положение камеры относительно земной поверхности, а также направление взгляда. Положение камеры определяется элементами <longitude>, <latitude>, <altitude> и <altitudeMode>, а направление взгляда – элементами <heading>, <tilt> и <roll>.

Элемент <Camera> предоставляет все шесть степеней свободы для контроля над полем зрения, то есть вы можете расположить камеру в произвольной точке пространства, а затем вращать ее вокруг осей x, y и z. И самое главное – вы можете наклонить камеру так, что она будет направлена выше линии горизонта, в небо.

Порядок выполнения поворотов важен. По умолчанию камера смотрит вдоль оси $-Z$ в направлении Земли. Перед тем как выполнять повороты, камера параллельно переносится вдоль оси Z на высоту <altitude>. Таким образом, порядок преобразований следующий:

1. <altitude> параллельный перенос вдоль оси z;
2. <heading> поворот относительно оси z;
3. <tilt> поворот относительно оси x;
4. <roll> поворот относительно оси z (еще раз).

Отметим, что при каждом повороте камеры две ее оси меняют ориентацию.

Элементы, специфичные для Camera

<longitude>

Долгота виртуальной камеры (точки расположения глаза). Угловые расстояния измеряются в градусах от нулевого меридиана. Значения к западу от нулевого меридиана изменяются в диапазоне от –180 градусов до 0, к востоку – от 0 до 180 градусов.

<latitude>

Широта виртуальной камеры. Измеряется в градусах к северу и югу от экватора и изменяется в диапазоне от –90 до 90 градусов.

<altitude>

Расстояние до земной поверхности в метрах. Об интерпретации значения см. описание элемента <altitudeMode>.

<heading>

Направление (азимут) камеры в градусах. По умолчанию подразумевается 0 (север). Изменяется в диапазоне от 0 до 360 градусов.

<tilt>

Угол поворота камеры относительно оси x (наклон). Значение 0 означает, что взгляд направлен строго вниз к Земле (наиболее часто встречающийся случай). Значение 90^0 означает, что взгляд направлен к линии горизонта. Значения, большие 90^0, соответствуют взгляду вверх, в небо.

<roll>

Угол поворота камеры относительно оси z (крен). Значение изменяется в диапазоне от –180 до +180 градусов.

<altitudeMode>

Определяет способ интерпретации элемента <altitude>. Допустимы следующие значения:

`clampToGround` (по умолчанию)

Означает, что значение, заданное в элементе <altitude>, следует игнорировать и поместить камеру на поверхность Земли.

`relativeToGround`

Значение <altitude> измеряется в метрах относительно поверхности Земли.

`absolute`

Значение <altitude> измеряется в метрах над уровнем моря.

Содержится в

Любой элемент, производный от Feature; <NetworkLinkControl>. Родительский элемент не может содержать одновременно <Camera> и <LookAt>.

ColorStyle

```
<!-- ColorStyle id="ID" -->
  <color>ffffffff</color>          <!-- kml:color -->
  <colorMode>normal</colorMode>    <!-- kml:colorModeEnum -->
<!-- /ColorStyle -->
```

Описание

Это абстрактный элемент, который нельзя использовать в KML-файле непосредственно. Он предоставляет набор элементов, наследуемых производными стилями (<LineStyle>, <PolyStyle>, <IconStyle> и <LabelStyle>).

Элементы, специфичные для ColorStyle

<color>

Цвет и прозрачность записываются в шестнадцатеричном виде в порядке *aabbggrr*, где *aa* – альфа-канал (от 00 до ff), *bb* – синяя составляющая (от 00 до ff), *gg* – зеленая составляющая (от 00 до ff), *rr* – красная составляющая (от 00 до ff). Если альфа-канал равен 00, то цвет полностью прозрачен, а если ff – то полностью непрозрачен.

<colorMode>

Может принимать значение `normal` (никакого эффекта) или `random`. В последнем случае базовый цвет <color> рандомизируется (применяется равномерное распределение).

- Для получения полностью случайных цветов задайте для элемента <color> белый цвет (00ffffff).
- Если в цвете задана одна составляющая (например, ff0000ff, что означает чисто красный цвет), то будут рандомизироваться значения только этой

составляющей. В данном случае цвет будет изменяться от 00 (в цвете нет красной составляющей) до ff (насыщенный красный цвет).

- Если заданы две или все три составляющие цвета, то рандомизация применяется независимо к каждой составляющей, и каждая будет варьироваться от 00 до ff.
- Прозрачность цвета определяется альфа-компонентой <color> и никогда не рандомизируется.

Container

```
<!-- Container id="ID" -->                <!-- Document, Folder -->
  <!-- наследуются от элемента Feature, см. <Feature> -->
  <name>...</name>                        <!-- string -->
  <visibility>1</visibility>              <!-- boolean -->
  <open>0</open>                          <!-- boolean -->
  <address>...</address>                  <!-- string -->
  <AddressDetails xmlns="urn:oasis:names:tc:ciq:xsdschema:xAL:2.0">...
      </AddressDetails>                   <!-- string -->
  <phoneNumber>...</phoneNumber>          <!-- string -->
  <Snippet maxLines="2">...</Snippet>     <!-- string -->
  <description>...</description>          <!-- string -->
  <AbstractView>...</AbstractView>        <!-- LookAt или Camera -->
  <TimePrimitive>...</TimePrimitive>      <!-- TimeSpan или TimeStamp -->
  <styleUrl>...</styleUrl>                <!-- anyURI -->
  <StyleSelector>...</StyleSelector>      <!-- Style или StyleMap -->
  <Region>...</Region>
  <atom:author>...</atom:author>
      <!-- xmlns:atom="http://www.w3.org/2005/Atom" -->
  <atom:link />

  <!-- специфичны для элемента Container -->
  <!-- 0 или более элементов Feature -->
<!-- /Container -->
```

Описание

Это абстрактный элемент, который нельзя использовать в KML-файле непосредственно. Он предоставляет набор элементов, наследуемых производными элементами (<Document> и <Folder>). Элемент Container может содержать один или несколько элементов Feature, что позволяет создавать вложенные иерархии.

<Document>

```
<Document id="ID">
  <!-- наследуются от элемента Feature -->
  <name>...</name>                  <!-- string -->
  <open>0</open>                    <!-- boolean -->
  <visibility>1</visibility>        <!-- boolean -->
  <atom:author>...</atom:author>    <!-- xmlns:atom -->
  <atom:link />                                    <!-- xmlns:atom -->
  <address>...</address>                           <!-- string -->
  <xal:AddressDetails...</xal:AddressDetails>      <!-- xmlns:xal -->
  <phoneNumber>...</phoneNumber>                   <!-- string -->
  <Snippet maxLines="2">...</Snippet>              <!-- string -->
  <description>...</description>                   <!-- string -->
  <AbstractView>...</AbstractView>                 <!-- LookAt или Camera -->
  <TimePrimitive>...</TimePrimitive>               <!-- TimeSpan или TimeStamp -->
  <styleUrl>...</styleUrl>                         <!-- anyURI -->
  <StyleSelector>...</StyleSelector>               <!-- Style или StyleMap -->
```

```
  <Region>...</Region>
  <ExtendedData>...</ExtendedData>

  <!-- специфичны для элемента Document -->
  <!-- 0 или более элементов Schema -->
  <!-- 0 или более элементов Feature -->
</Document>
```

Описание

Элемент <Document> является контейнером для деталей Feature, разделяемых стилей и схем. Этот элемент необходим, если в KML-файле используются разделяемые стили. Информацию о разделяемых стилях см. в главе 4 «Стили и значки». <Document> – единственный элемент, который может содержать разделяемые стили, и только он может содержать схемы. В KML-файле может быть несколько элементов <Document>.

Отметим, что разделяемые стили не оказывают влияния на детали, находящиеся внутри <Document>, если деталь не содержит явной ссылки на стиль в элементе <styleUrl>. Чтобы стиль применялся к самому элементу <Document>, последний должен содержать явную ссылку <styleUrl>.

Содержится в

<kml>

Содержит

- 0 или более элементов, производных от Feature;
- 0 или более элементов, производных от Schema.

<ExtendedData>

```
<ExtendedData>
  <Data name="string">
    <displayName>...</displayName>         <!-- string -->
    <value>...</value>                    <!-- string -->
  </Data>
  <SchemaData schemaUrl="anyURI">
    <SimpleData name="">...</SimpleData>   <!-- string -->
  </SchemaData>
  <namespace_prefix:other>...</namespace_prefix:other>
</ExtendedData>
```

Описание

Элемент <ExtendedData> предлагает три способа добавить данные нестандартного типа в деталь KML, а именно:

- сослаться на XML-элементы, определенные в других пространствах имен, указав в KML-файле внешнее пространство имен с помощью тега <*namespace_prefix*> (простой способ);
- включить нетипизированные пары имя/значение с помощью элемента <Data> (простой способ);
- объявить типизированные поля в элементе <Schema>, а затем создать их экземпляры, пользуясь элементом <SchemaData> (более сложный способ).

Все три способа можно применять в одном KML-файле или в одной детали Feature (при условии, что включаются данные разных типов). Дополнительную информацию см. в главе 8 «Большие наборы данных».

Элементы, специфичные для <ExtendedData>

<Data name="string">

Создает нетипизированную пару имя/значение. У имени могут быть два варианта: name и displayName. Атрибут `name` уникально идентифицирует данную пару в пределах одного KML-файла. Элемент <displayName> используется геобраузером для вывода отформатированного имени, возможно с пробелами и HTML-разметкой. Внутри элемента <text>, вложенного в <BalloonStyle>, вместо компонента в формате `$[name:displayName]` подставляется значение <displayName>. Если в таком же формате записать значение атрибута `name` элемента <Data> (например, `$[holeYardage]`), то вместо него будет подставлено ассоциированное с этим именем значение <value>. По умолчанию в информационном окне детали отображаются все ассоциированные с ней пары имя/значение.

<displayName>

Необязательный отформатированный вариант `name`, используемый только для отображения.

<value>

Значение в паре.

<SchemaData schemaUrl="anyURI">

Этот элемент используется в сочетании с <Schema> для добавления данных нестандартного типа в любой элемент, производный от Feature. Элемент Schema (идентифицируемый атрибутом `schemaUrl`) объявляет тип данных. Сами объекты данных («экземпляры» нестандартного типа) определяются с помощью элемента <SchemaData>.

Значением `schemaUrl` может быть полный URL, ссылка на идентификатор схемы, определенной во внешнем KML-файле, или ссылка на идентификатор схемы, определенной в том же файле.

Элемент <Schema> может быть только потомком <Document>. Элемент <ExtendedData> является потомком детали, содержащей данные нестандартного типа.

<SimpleData name=" ">

Присваивает значение полю нестандартного типа, определяемому атрибутом `name`. Тип и имя этого поля должны быть объявлены в элементе <Schema>.

<namespace_prefix:other>

Этот элемент позволяет добавлять нетипизированные данные. Не забудьте определить префикс пространства имен в элементе <kml> в начале файла или в качестве атрибута элемента <ExtendedData>. Этим префиксом необходимо помечать все экземпляры пользовательских данных. Добавленные таким способом данные сохраняются в KML-файле, но геобраузер никак их не использует.

Содержится в

Любой элемент, производный от Feature.

См. также

<Schema>

Feature

Раздел 9.1 спецификации OGC KML 07-147

```
<!-- Feature id="ID">

  <name>...</name>
  <open>0</open>
  <visibility>1</visibility>
  <atom:author>...</atom:author>
  <atom:link />
  <address>...</address>
  <xal:AddressDetails>...</xal:AddressDetails>
  <phoneNumber>...</phoneNumber>
  <Snippet maxLines="2">...</Snippet>
  <description>...</description>
  <AbstractView>...</AbstractView>
  <TimePrimitive>...</TimePrimitive>
  <styleUrl>...</styleUrl>
  <StyleSelector>...</StyleSelector>
  <Region>...</Region>
  <ExtendedData>...</ExtendedData>
<!-- /Feature -->
```

```
<!-- Document, Folder, NetworkLink,
       Placemark, GroundOverlay,
       PhotoOverlay, ScreenOverlay -->
       <!-- string -->
       <!-- boolean -->
       <!-- boolean -->
       <!-- xmlns:atom -->
       <!-- xmlns:atom -->
       <!-- string -->
       <!-- xmlns:xal -->
       <!-- string -->
       <!-- string -->
       <!-- string -->
       <!-- LookAt или Camera -->
       <!-- TimeSpan или TimeStamp -->
       <!-- anyURI -->
       <!-- Style или StyleMap -->
```

Описание

Это абстрактный элемент, который нельзя использовать в KML-файле непосредственно. Он предоставляет набор элементов, наследуемых производными элементами (Container, <Document>, <Folder>, <NetworkLink>, <Placemark>, Overlay, <GroundOverlay>, <PhotoOverlay>, <ScreenOverlay>).

Примечание

Элемент <Snippet> объявлен в версии KML 2.2 устаревшим и заменен на <snippet>. Однако к моменту публикации этой книги программа Google Earth еще не поддерживала элемент <snippet>. Поэтому мы продолжаем упоминать в руководстве вариант, начинающийся с заглавной буквы.

Элементы, специфичные для Feature

<name>

Определяемый пользователем текст, отображаемый на трехмерной панораме в виде надписи, ассоциированной с объектом (например, меткой, папкой или сетевой ссылкой).

<visibility>

Определяет, нужно ли рисовать деталь на панораме сразу после начальной загрузки. Чтобы деталь была видимой, элемент <visibility> должен быть равен 1 у всех ее предков.

<open>

Определяет, должен ли документ или папка быть открытым или закрытым на панели **Метки** сразу после начальной загрузки. 0 означает закрытый (по умолчанию), 1 – открытый. См. также <ListStyle>. Это параметр применим только к элементам <Document>, <Folder> и <NetworkLink>.

<atom:author>

Этот элемент определен в спецификации Atom Syndication. Ее полный текст можно найти на сайте *http://atompub.org*. В любом KML-файле, где этот элемент встречается, должно присутствовать следующее пространство имен:

```
xmlns:atom="http://www.w3.org/2005/atom"
```

(см. главу 3). Информация об авторе отображается в результатах поиска как в геобраузере, так и в картографических приложениях типа Google Maps.

<atom:link href=»»**>**

Задает URL сайта, на котором находится данный KML- или KMZ-файл. Не забудьте включить следующее пространство имен в KML-файл, где этот элемент встречается:

```
xmlns:atom="http://www.w3.org/2005/atom"
```

<address>

Строка, содержащая неструктурированный адрес, в котором указаны улица, город, штат и/или почтовый индекс. Тег <address> можно использовать для задания местоположения точки в качестве альтернативы широте и долготе. (Однако если элемент <Point> присутствует, то ему отдается предпочтение над <address>.)

<xal:AddressDetails>

Структурированный адрес в формате OASIS xAL 2.0 (eXtensible Address Language – расширяемый язык описания адреса) – международном стандарте форматирования адресов. Не забудьте включить следующее пространство имен в KML-файл, где этот элемент встречается:

```
xmlns:xal="urn:oasis:names:tc:ciq:xsdschema:xAL:2.0"
```

<phoneNumber>

Номер телефона в виде строки. Должен быть представлен в формате IETF RFC 3966.

<Snippet maxLines=»»>

Краткое описание детали. Если этот элемент задан, то текст отображается на панели списка вместо <description>. Внутри не должно быть HTML-разметки. У элемента <Snippet> имеется атрибут `maxLines` – целое число, равное максимальному количеству отображаемых строк. (См. выше примечание, касающееся элемента <Snippet>.)

<description>

Определяемый пользователем текст, который отображается во всплывающем окне (см. главу 2 «Метки и информационные окна»). Элемент <description> поддерживает как обычный текст, так и подмножество тегов HTML, в том

числе таблиц. Не поддерживаются никакие другие веб-технологии – скажем, динамическая разметка страниц (PHP, JSP, ASP), языки сценариев (VBScript, JavaScript) и другие языки программирования (Java, Python). Не поддерживается также видео.

При использовании HTML для превращения некоего слова в гиперссылку или для включения изображений необходимо либо употреблять HTML-компоненты (< и т. п.), либо погружать разметку в элемент CDATA, чтобы экранировать угловые скобки, апострофы и прочие специальные символы.

В HTML анкерный элемент <a> имеет атрибут `href`, определяющий URL. Если в нем указан KML-файл с расширением *.kml* или *.kmz*, то геобраузер самостоятельно загружает его при щелчке по ссылке. Если URL заканчивается расширением, неизвестным геобраузеру (например, *.html*), то URL передается веб-браузеру.

Значением `href` может быть фрагмент URL (то есть URL, в котором есть знак #, за которым следует идентификатор KML). Если пользователь щелкает по ссылке, содержащей такой фрагмент, то по умолчанию браузер совершает «перелет» к детали, идентификатор которой совпадает с фрагментом. Если в этой детали есть элемент <LookAt> или <Camera>, то она рассматривается с указанной точки обзора.

Описанное поведение можно уточнить, дописав в конец фрагмента URL одну из следующих строк:

`;flyto` (по умолчанию)

Перелететь к детали.

`;balloon`

Открыть информационное окно детали, но не перелетать к ней.

`;balloonFlyto`

Открыть информационное окно детали и перелететь к ней.

Атрибут `type` используется внутри элемента <a>, если `href` не заканчивается расширением *.kml* или *.kmz*, но ссылку требуется интерпретировать в контексте KML (например, речь идет о PHP-сценарии, порождающем KML-документ). Задайте его значение в следующем формате:

```
type="application/vnd.google-earth.kml+xml"
```

AbstractView

Определяет точку обзора, ассоциированную с элементом, производным от Feature. См. <Camera> и <LookAt>.

TimePrimitive

Ассоциирует с данной деталью промежуток времени (<TimeSpan>) или момент времени (<TimeStamp>).

<styleUrl>

URL элемента <Style> или <StyleMap>, определенного в документе. Если стиль определен в том же файле, указывайте ссылку в виде #-фрагмента. Если же

стиль определен во внешнем файле, указывайте полный URL, содержащий #-фрагмент. См. главу 4 «Стили и значки».

StyleSelector

Задает один или несколько элементов <Style> или <StyleMap>, которые служат для настройки внешнего вида любого элемента, производного от Feature или Geometry внутри <Placemark>. См. <BalloonStyle>, <ListStyle>, StyleSelector и стили, прозводные от ColorStyle. См. также главу 4 «Стили и значки».

<Region>

Детали и геометрические формы, ассоциированные с регионом, рисуются только в случае, когда регион активен. См. <Region>.

<ExtendedData>

Позволяет добавлять в KML-файл данные нестандартного типа. См. <ExtendedData>.

Содержится в

<kml>, <Folder>

<Folder>

```
<Folder id="ID">
  <!-- наследуются от элемента Feature -->
  <name>...</name>                                  <!-- string -->
  <open>0</open>                                    <!-- boolean -->
  <visibility>1</visibility>                        <!-- boolean -->
  <atom:author>...</atom:author>                    <!-- xmlns:atom -->
  <atom:link />                                     <!-- xmlns:atom -->
  <address>...</address>                            <!-- string -->
  <xal:AddressDetails...</xal:AddressDetails>       <!-- xmlns:xal -->
  <phoneNumber>...</phoneNumber>                    <!-- string -->
  <Snippet maxLines="2">...</Snippet>               <!-- string -->
  <description>...</description>                    <!-- string -->
  <AbstractView>...</AbstractView>                  <!-- LookAt или Camera -->
  <TimePrimitive>...</TimePrimitive>                <!-- TimeSpan или TimeStamp -->
  <styleUrl>...</styleUrl>                          <!-- anyURI -->
  <StyleSelector>...</StyleSelector>                <!-- Style или StyleMap -->
  <Region>...</Region>
  <ExtendedData>...</ExtendedData>

  <!-- специфичны для элемента Folder -->
  <!-- 0 или более элементов Feature -->
</Folder>
```

Описание

Папка служит для иерархической организации других деталей (<Document>, <Folder>, <Placemark>, <NetworkLink> и Overlay). Деталь видима только в том случае, когда видимы все ее предки.

Содержится в

<kml>, <Folder>

Содержит

0 или более элементов, производных от Feature.

Geometry

Раздел 10.1 спецификации OGC KML 07-147

```
<!-- Geometry id="ID" -->   <!-- Point, LinString, LinearRing, Polygon
                                MultiGeometry, Model -->
<!-- /Geometry -->
```

Описание

Это абстрактный элемент, который нельзя использовать в KML-файле непосредственно. Элемент-заглушка, наследуемый всеми геометрическими объектами.

<GroundOverlay>

```
<GroundOverlay id="ID">
  <!-- наследуются от элемента Feature -->
  <name>...</name>                              <!-- string -->
  <open>0</open>                                <!-- boolean -->
  <visibility>1</visibility>                    <!-- boolean -->
  <atom:author>...</atom:author>                <!-- xmlns:atom -->
  <atom:link />                                 <!-- xmlns:atom -->
  <address>...</address>                        <!-- string -->
  <xal:AddressDetails...</xal:AddressDetails>   <!-- xmlns:xal -->
  <phoneNumber>...</phoneNumber>                <!-- string -->
  <Snippet maxLines="2">...</Snippet>           <!-- string -->
  <description>...</description>                <!-- string -->
  <AbstractView>...</AbstractView>              <!-- LookAt или Camera -->
  <TimePrimitive>...</TimePrimitive>            <!-- TimeSpan или TimeStamp -->
  <styleUrl>...</styleUrl>                      <!-- anyURI -->
  <StyleSelector>...</StyleSelector>            <!-- Style или StyleMap -->
  <Region>...</Region>
  <ExtendedData>...</ExtendedData>

  <!-- наследуются от элемента Overlay -->
  <color>ffffffff</color>                       <!-- kml:color -->
  <drawOrder>0</drawOrder>                      <!-- int -->
  <Icon>...</Icon>

  <!-- специфичны для элемента GroundOverlay -->
  <altitude>0.0</altitude>                      <!-- double -->
  <altitudeMode>clampToGround</altitudeMode>
      <!-- kml:altitudeModeEnum: clampToGround или absolute -->
  <LatLonBox>
    <north>180.0</north>                        <!-- kml:angle90 -->
    <south>-180.0</south>                       <!-- kml:angle90 -->
    <east>180.0</east>                          <!-- kml:angle180 -->
    <west>-180.0</west>                         <!-- kml:angle180 -->
    <rotation>0.0</rotation>                    <!-- kml:angle180 -->
  <LatLonBox>
</GroundOverlay>
```

Описание

Этот элемент рисует изображение, наложенное на рельеф местности. Потомок <href> элемента <Icon> определяет файл с изображением. Этот файл может быть локальным или размещаться на веб-сервере. Если элемент <Icon> опущен или не содержит <href>, рисуется прямоугольник того цвета и размера, которые заданы в рельефном наложении. См. главу 5 «Наложения».

Элементы, специфичные для <GroundOverlay>

<altitude>

Расстояние до земной поверхности в метрах, интерпретируемое согласно значению элемента <altitudeMode>.

<altitudeMode>

Способ интерпретации <altitude>. Допустимые значения:

`clampToGround` (по умолчанию)

Задание высоты игнорируется, изображение накладывается прямо на поверхность Земли.

`absolute`

Считается, что задана высота над уровнем моря вне зависимости от реального возвышения рельефа, расположенного под наложением.

<LatLonBox>

Задает двумерный прямоугольник, описанный вокруг рельефного наложения.

<north>

Широта северной стороны ограничивающего прямоугольника.

<south>

Широта южной стороны ограничивающего прямоугольника.

<east>

Долгота восточной стороны ограничивающего прямоугольника.

<west>

Долгота западной стороны ограничивающего прямоугольника.

<rotation>

Угол поворота накладываемого изображения вокруг центра в градусах. Отсчитывается против часовой стрелки.

Содержится в

<Document>, <Folder>

<Icon>

```
<Icon id="ID">
  <href>...</href>                          <!-- string -->
  <refreshMode>onChange</refreshMode>
      <!-- kml:refreshModeEnum: onChange, onInterval или onExpire -->
  <refreshInterval>4.0</refreshInterval>     <!-- double -->
  <viewRefreshMode>never</viewRefreshMode>
      <!-- kml:viewRefreshModeEnum: never, onStop, onRequest, onRegion -->
  <viewRefreshTime>4</viewRefreshTime>       <!-- double -->
  <viewBoundScale>1</viewBoundScale>         <!-- double -->
  <viewFormat>...</viewFormat>               <!-- string -->
  <httpQuery>...</httpQuery>                 <!-- string -->
</Icon>
```

Описание

Определяет изображение, ассоциированное со стилем значка или наложением. У элемента <Icon> могут быть такие же потомки, как у <Link>. Потомок <href>

определяет файл изображения, используемого в качестве наложения или значка метки. Этот файл может быть локальным или размещаться на удаленном веб-сервере.

Элементы, специфичные для <Icon>

<href>

Задает HTTP-адрес или имя локального файла, содержащего значок.

<refreshMode>

Описание <refreshMode> и других нижеперечисленных элементов см. в разделе <Link>.

<refreshInterval>
<viewRefreshMode>
<viewRefreshTime>
<viewBoundScale>
<viewFormat>
<httpQuery>

Содержится в

<GroundOverlay>, <PhotoOverlay>, <ScreenOverlay>, <IconStyle>

<IconStyle>

```
<IconStyle id="ID">
  <!-- наследуются от элемента ColorStyle -->
  <color>ffffffff</color>                      <!-- kml:color -->
  <colorMode>normal</colorMode>
      <!-- kml:colorModeEnum: normal или random -->

  <!-- специфичны для элемента IconStyle -->
  <scale>1.0</scale>                           <!-- double -->
  <heading>1.0</heading>                       <!-- kml:angle360 -->
  <Icon>
    <href>...</href>
  <Icon>
  <hotSpot x="0.5" y="0.5"
    xunits="fraction" yunits="fraction"/>      <!-- kml:vec2 -->
</IconStyle>
```

Описание

Определяет, как рисуются значки, ассоциированные с точечными метками, – в списке и на трехмерной панораме Google Earth. В элементе <Icon> задается изображение значка. Элемент <scale> определяет масштабирование значка по осям x и y. Цвет, заданный в элементе <color>, смешивается с цветом изображения. См. главу 4 «Стили и значки».

Если в потомке <ItemIcon> элемента <ListStyle> тоже задан значок, то при отображении в списке ему отдается преимущество над значком, заданным в потомке <Icon> элемента <IconStyle>.

Элементы, специфичные для <IconStyle>

<scale>

Изменяет размер значка.

<heading>

Угол поворота (север, юг, восток, запад) в градусах. По умолчанию равен 0 (север).

<Icon>

Нестандартный значок. Если встречается внутри <IconStyle>, то единственным потомком должен быть элемент <href>.

<href>

HTTP-адрес или путь к локальному файлу с изображением значка.

<hotSpot x="0.5" y="0.5" xunits="fraction" yunits="fraction" **/>**

Определяет точку внутри значка, которая совмещается с точкой, указанной в элементе <Point> метки. Значения x и y можно задать тремя способами: в виде количества пикселей (`"pixels"`), в виде доли от размера значка (`"fraction"`) или в виде смещения `"insetPixels"`, то есть количества пикселей от правого верхнего угла значка. Значения x и y можно задавать в разных единицах измерения: например, x может быть выражено в пикселях, а y – в виде процентной доли. Начало системы координат совпадает с левым нижним углом значка.

x

Либо количество пикселей, либо доля от размера значка, либо количество пикселей от правого верхнего угла.

y

Либо количество пикселей, либо доля от размера значка, либо количество пикселей от правого верхнего угла.

xunits

Единица измерения x. Значение `fraction` означает, что x – доля от ширины значка. Значение `pixels` означает, что x измеряется в пикселях от левого нижнего угла, а значение `insetPixels` – что x измеряется в пикселях от правого верхнего угла.

yunits

Единица измерения y. Значение `fraction` означает, что y – доля от высоты значка. Значение `pixels` означает, что y измеряется в пикселях от левого нижнего угла, а значение `insetPixels` – что y измеряется в пикселях от правого верхнего угла.

Содержится в

<Style>

<kml>

Раздел 7.1 спецификации OGC KML 07-147

```
<kml xmlns="http://earth.google.com/kml/2.2" hint="target=sky">
  <NetworkLinkControl>
  <!-- 0 или 1 элемент Feature -->
</kml>
```

Описание

Корневой элемент KML-файла. Обязателен. Следует сразу же за декларацией xml в начале файла. Атрибут `hint` служит для Google Earth указанием на то, что файл содержит небесные, а не земные координаты (см. приложение B). (Этот атрибут следует включать только для астрономических данных.)

Элемент <kml> должен также включать описание пространств имен для всех внешних XML-схем, на которые есть ссылки в файле.

Содержит

- 0 или 1 деталь;
- 0 или 1 элемент <NetworkLinkControl>.

<LabelStyle>

```
<LabelStyle id="ID">
  <!-- наследуются от элемента ColorStyle -->
  <color>ffffffff</color>                    <!-- kml:color -->
  <colorMode>normal</colorMode>
      <!-- kml:colorModeEnum: normal или random -->

  <!-- специфичны для элемента LabelStyle -->
  <scale>1.0</scale>                         <!-- double -->
</LabelStyle>
```

Описание

Определяет способ отображения имени <name> детали на трехмерной панораме. Можно задать цвет, режим цвета и масштаб надписи. См. главу 4 «Стили и значки».

Элементы, специфичные для <LabelStyle>

<scale>

Коэффициент масштабирования надписи.

Содержится в

<Style>

<LinearRing>

```
<LinearRing id="ID">
  <extrude>0</extrude>                       <!-- boolean -->
  <tessellate>0</tessellate>                 <!-- boolean -->
  <altitudeMode>clampToGround</altitudeMode>
      <!-- kml:altitudeModeEnum: clampToGround
           relativeToGround или absolute -->
  <coordinates>...</coordinates>             <!-- кортежи lon,lat[,alt] -->
</LinearRing>
```

Описание

Описывает замкнутую ломаную, обычно внешнюю границу многоугольника. Может также применяться для описания внутренней границы многоугольника для создания в нем дырок. Элемент Polygon может содержать несколько элементов <LinearRing>, описывающих внутренние границы. См. главу 3 «Геометрия».

Элементы, специфичные для <LinearRing>

<extrude>

Определяет, нужно ли соединять LinearRing с поверхностью Земли. Для экструдирования этой геометрической формы параметр <altitudeMode> должен быть равен `relativeToGround` или `absolute`, а компоненты высоты в элементе <coordinates> должны быть больше 0 (то есть форма располагается над землей). Экструдируются только вершины ломаной, но не ее внутренняя часть. Линии, исходящие из вершин, направлены к центру земной сферы.

<tesselate>

Определяет, должна ли ломаная повторять кривизну Земли. Чтобы тесселяция была возможна, параметр <altitudeMode> должен быть равен `clampToGround`. Для очень длинных ломаных тесселяция должна быть включена, иначе отрезки уйдут под землю и не будут видны.

<altitudeMode>

Способ интерпретации высоты в элементах <coordinates>. Допустимые значения: `clampToGround`, `relativeToGround` и `absolute`. Эти режимы рассматриваются в главе 3 «Геометрия».

<coordinates>

Четыре или более кортежей, в каждом из которых задаются значения долготы, широты и высоты (с плавающей точкой). Высоту задавать необязательно. Кортеж не должен содержать пробелов. Последняя координата должна совпадать с первой. Широта и долгота выражаются в градусах.

Содержится в

<innerBoundaryIs>, <outerBoundaryIs>, <MultiGeometry>, <Placemark>

<LineString>

```
<LineString id="ID">
  <extrude>0</extrude>                          <!-- boolean -->
  <tessellate>0</tessellate>                    <!-- boolean -->
  <altitudeMode>clampToGround</altitudeMode>
      <!-- kml:altitudeModeEnum: clampToGround
           relativeToGround или absolute -->
  <coordinates>...</coordinates>                <!-- кортежи lon,lat[,alt] -->
</LineString>
```

Описание

Описывает последовательность соединенных отрезков. Для задания цвета, режима цвета и толщины линии используйте элемент <LineStyle>. Если ломаная экструдирована, то она продлевается до поверхности Земли, образуя многоуголь-

ник, напоминающий стенку или забор. Если экструзия задана, то для рисования самой ломаной применяется текущий стиль <LineStyle>, а экструдированные линии рисуются в соответствии с текущим стилем <PolyStyle>. Элемент <LineString> часто называют «путем».

Элементы, специфичные для <LineString>

<extrude>

Определяет, нужно ли соединять ломаную с поверхностью Земли. Для экструдирования этой геометрической формы параметр <altitudeMode> должен быть равен `relativeToGround` или `absolute`, а компоненты высоты в элементе <coordinates> должны быть больше 0 (то есть форма располагается над землей). Экструдируются только вершины ломаной, но не ее внутренняя часть. Линии, исходящие из вершин, направлены к центру земной сферы.

<tesselate>

Определяет, должна ли ломаная повторять кривизну Земли. Чтобы тесселяция была возможна, параметр <altitudeMode> должен быть равен `clampToGround`. Для очень длинных ломаных тесселяция должна быть включена, иначе отрезки уйдут под землю и не будут видны.

<altitudeMode>

Способ интерпретации высоты в элементах <coordinates>. Допустимые значения: `clampToGround`, `relativeToGround` и `absolute`. Эти режимы рассматриваются в главе 3 «Геометрия».

<coordinates>

Два или более кортежей, в каждом из которых задаются значения долготы, широты и высоты (с плавающей точкой). Высоту задавать необязательно. Кортежи отделяются друг от друга пробелами. Внутри кортежа пробелов быть не должно.

Содержится в

<MultiGeometry>, <Placemark>

- 0 или 1 элемент <NetworkLinkControl>.

<LineStyle>

```
<LineStyle id="ID">
  <!-- наследуются от элемента ColorStyle -->
  <color>ffffffff</color>                          <!-- kml:color -->
  <colorMode>normal</colorMode>
      <!-- kml:colorModeEnum: normal или random -->

  <!-- специфичны для элемента LabelStyle -->
  <width>1.0</width>                               <!-- double -->
</LineStyle>
```

Описание

Определяет стиль рисования (цвет, режим цвета и толщину линии) для всех линейных геометрических форм, к которым относятся в том числе границы окон-

туренных многоугольников и экструдированные линии значков, ассоциированных с метками (если экструзия включена). См. главу 4 «Стили и значки».

Элементы, специфичные для <LineStyle>

<width>

Толщина линии в пикселях.

Содержится в

<Style>

<Link> Раздел 13.1 спецификации OGC KML 07-147

```
<Link id="ID">
  <href>...</href>                                          <!-- string -->
  <refreshMode>onChange</refreshMode>
      <!-- kml:refreshModeEnum: onChange, onInterval или onExpire -->
  <refreshInterval>4.0</refreshInterval>                    <!-- double -->
  <viewRefreshMode>never</viewRefreshMode>
      <!-- kml:viewRefreshModeEnum: never, onStop, onRequest, onRegion -->
  <viewRefreshTime>4</viewRefreshTime>                      <!-- double -->
  <viewBoundScale>1</viewBoundScale>                        <!-- double -->
 <viewFormat>BBOX=[bboxWest],[bboxSouth],[bboxEast],[bboxNorth]
   </viewFormat> <!-- string -->
  <httpQuery>...</httpQuery>                                <!-- string -->
</Link>
```

Описание

Элемент <Link> определяет положение следующих объектов:

- KML-файлов, загружаемых по сетевым ссылкам;
- файлов с изображениями, используемыми в любом наложении Overlay (элемент <Icon> имеет такие же поля, как <Link>);
- файлов с моделями для элемента <Model>.

Файл загружается и обновляется условно, в зависимости от заданных в этом элементе параметров. Можно задать два разных набора параметров: одни связаны с временем (<refreshMode> и <refreshInterval>), другие – с текущим полем зрения «камеры» (<viewRefreshMode> и <viewRefreshTime>). Дополнительно элемент Link определяет, нужно ли масштабировать размеры ограничивающего прямоугольника, посылаемого серверу (<viewBoundScale>), и задает несколько необязательных параметров просмотра, которые можно отправить серверу (<viewFormat>), а также необязательные параметры, содержащие информацию о номере версии и языке геобраузера.

При запросе файла серверу посылается URL, состоящий из трех частей (см. пример в главе 6 «Сетевые ссылки»):

- *href* – гипертекстовая ссылка, указывающая, какой файл загрузить;
- *произвольная форматная строка*, формируемая из (а) параметров, заданных вами в элементе <viewFormat>, или (б) параметров ограничивающего прямоугольника (это режим по умолчанию, который выбирается, когда элемент <viewFormat> отсутствует);

- *вторая форматная строка*, которая задается в элементе <httpQuery>.

Если в элементе <href> задан локальный файл, то <viewFormat> и <httpQuery> игнорируются.

Элементы, специфичные для <Link> (и <Icon>)

<href>

Задает URL (HTTP-адрес или путь к локальному файлу). Если родителем <Link> является элемент NetworkLink, то <href> ссылается на KML-файл. Если родителем <Link> является элемент Model, то <href> ссылается на COLLADA-файл. Если родителем <Link> является элемент Overlay, то <href> ссылается на изображение. Разрешается использовать относительные URL, в таком случае адрес вычисляется относительно объемлющего KML-файла.

<refreshMode>

Задает режим обновления по времени:

`onChange` (по умолчанию)

Обновлять после загрузки изображения и при изменении любого параметра, заданного в <Link>.

`onInterval`

Обновлять каждые *n* секунд, где *n* – значение параметра <refreshInterval>.

`onExpire`

Обновлять по истечении указанного срока. Если в загруженном файле есть элемент <NetworkLinkControl>, то момент времени, указанный во вложенном в него элементе <expires>, имеет приоритет над сроком хранения, заданным в HTTP-заголовках. Если элемент <expires> отсутствует, то используется HTTP-заголовок `max-age` (если он имеется). Если и заголовка `max-age` нет, то используется HTTP-заголовок Expires (если имеется). (Дополнительную информацию о заголовках протокола HTTP см. в документе RFC2616, в котором описана версия HTTP 1.1.)

<refreshInterval>

Указывает, что файл должен обновляться каждые *n* секунд.

<viewRefreshMode>

Задает режим обновления по событиям просмотра карты. Допустимы следующие значения:

`never` (по умолчанию)

Не обращать внимания на события просмотра. Параметры, заданные в элементе <viewFormat>, также игнорируются.

`onStop`

Обновлять файл спустя *n* секунд после прекращения движения, где *n* – значение, заданное в элементе <viewRefreshTime>.

`onRequest`

Обновлять файл, только когда пользователь явно запрашивает. (Например, в Google Earth пользователь может щелкнуть правой кнопкой мыши и выбрать из контекстного меню команду **Обновить**.)

`onRegion`

Обновлять файл, когда становится активным определенный регион. См. <Region>.

<viewRefreshTime>

Определяет, сколько секунд ждать после остановки камеры, перед тем как приступать к обновлению изображения (см. <viewRefreshMode> и `onStop` выше).

<viewBoundScale>

Масштабирует параметры ограничивающего прямоугольника перед отправкой их серверу. Значение, меньшее 1, означает, что нужно обработать область, меньшую текущей области просмотра, а значение, большее 1, – что нужно расширить границы области.

<viewFormat>

Задает формат строки запроса, которая дописывается в конец элемента <href> (если <href> ссылается на локальный файл, то этот параметр игнорируется). Если <viewRefreshMode> равно `onStop`, а элемент <viewFormat> опущен, то в строку запроса автоматически дописывается следующая информация:

```
BBOX=[bboxWest],[bboxSouth],[bboxEast],[bboxNorth]
```

Она соответствует спецификации ограничивающего прямоугольника Web Map Service (WMS).

Если задан пустой элемент <viewFormat>, то в строку запроса ничего не дописывается.

Можно также задать свои собственные параметры, которые будут дописываться в строку запроса. Если указана форматная строка, то она используется *вместо* BBOX. Если вам нужно и то, и другое, то необходимо самостоятельно добавить входящие в BBOX параметры, помимо собственных. В форматной строке можно использовать любые из перечисленных ниже параметров (и геобраузер подставит соответствующее текущее значение при формировании строки запроса):

`[lookLon], [lookLat]`

Долгота и широта точки, рассматриваемой из <LookAt>.

`[lookatRange], [lookatTilt], [lookatHeading]`

Значения, используемые в элементе <LookAt> (см. описание параметров <range>, <tilt> и <heading> в <LookAt>).

`[lookatTerrainLon], [lookatTerrainLat], [lookatTerrainAlt]`

Координаты в градусах/метрах точки обзора <LookAt>.

`[cameraLon], [cameraLat], [cameraAlt]`

Координаты в градусах/метрах точки, в которой размещена камера.

`[horizFov], [vertFov]`

Горизонтальное и вертикальное поля зрения камеры.

`[horizPixels], [vertPixels]`

Размер трехмерной панорамы Google Earth в пикселях.

`[terrainEnabled]`

Сообщает, показывает ли Google Earth в данный момент рельеф.

<httpQuery>

Добавляет в строку запроса информацию, исходя из заданных параметров. (Геобраузер подставляет текущие значения в момент формирования строки запроса.) Поддерживаются следующие параметры:

```
[clientVersion]
[kmlVersion]
[clientName]
[language]
```

Содержится в

<Model>, <NetworkLink>

См. также

<NetworkLinkVontrol>, <Region>

<ListStyle>

```
<ListStyle id="ID">
  <listItemType>check</listItemType>    <!-- kml:listItemTypeEnum:check,
                                             checkOnly, checkHideChildren,
                                             radioFolder -->
  <bgColor>ffffffff</bgColor>           <!-- kml:color -->
  <ItemIcon>                            <!-- 0 или более элементов ItemIcon -->
    <state>open</state>
        <!-- kml:itemIconModeEnum:open, closed, error, fetching0,
             fetching1, fetching2 -->
    <href>...</href>                    <!-- string -->
  <ItemIcon>
  <maxSnippetLines>2</maxSnippetLines>
</ListStyle>
```

Описание

Определяет способ отображения детали Feature в списке. Список – это иерархия контейнеров и потомков.

Элементы, специфичные для <ListStyle>

<listItemType>

Способ представления детали в списке. Допустимы следущие значения:

`check` (по умолчанию)

Видимость детали определяется состоянием флажка.

`radioFolder`

Означает, что в каждый момент времени может быть виден только один из объектов, находящихся в контейнере.

`checkOffOnly`

Если задан для контейнера или сетевой ссылки, то пользователь не может одновременно включить все содержащиеся в нем объекты. Иными словами, пользователю разрешается выключить разом все элементы-потомки контейнера или сетевой ссылки, но включать их он может только по отдельности. Такой режим полезен для контейнеров или сетевых ссылок, содержащих много данных.

`checkHideChildren`

Использовать обычный флажок для управления видимостью, но не отображать потомков контейнера или сетевой ссылки в списке. Флажок позволяет делать объекты-потомки видимыми или невидимыми на панораме.

<bgColor>

Цвет фона для краткого описания <Snippet>. Цвет и прозрачность записываются в шестнадцатеричном виде. Диапазон значений каждой компоненты от 0 до 255 (в шестнадцатеричной записи от 00 до ff). Для альфа-канала 00 означает полную прозрачность, ff – полную непрозрачность. Цвет записывается в порядке aabbggrr, где aa – альфа-канал (от 00 до ff), bb – синяя составляющая (от 00 до ff), gg – зеленая составляющая (от 00 до ff), rr – красная составляющая (от 00 до ff).

<ItemIcon>

Значок, отражающий состояние папки <Folder> или ссылки <Link> в списке. Значки, ассоциированные с состояниями `open` и `closed`, используются для папок и сетевых ссылок. Значки, ассоциированные с состояниями `error`, `fetching0`, `fetching1` и `fetching2`, используются только для сетевых ссылок. См. главу 4 «Стили и значки».

<state>

Задает текущее состояние папки или сетевой ссылки. Допустимы значения `open`, `closed`, `error`, `fetching0`, `fetching1` и `fetching2`. Значения можно объединять, перечисляя их через пробел (не запятую).

<href>

Задает URL изображения значка, которое будет отображаться в списке.

<maxSnippetLines>

Максимальное число строк краткого описания, отображаемых в списке.

Содержится в

<Style>

<LookAt>

```
<LookAt id="ID">
  <longitude>0.0</longitude>        <!-- kml:angle180 -->
  <latitude>0.0</latitude>          <!-- kml:angle90 -->
  <altitude>0.0</altitude>          <!-- double -->
```

```
  <heading>0.0</heading>            <!-- kml:angle360 -->
  <tilt>0.0</tilt>                  <!-- kml:anglepos180 -->
  <range>0.0</range>                <!-- double -->
  <altitudeMode>clampToGround</altitudeMode>
      <!-- kml:altitudeModeEnum:clampToGround, relativeToGround, absolute -->
</LookAt>
```

Описание

Определяет точку обзора, ассоциированную с любым элементом, производным от Feature. Элемент <LookAt> позиционирует точку обзора относительно рассматриваемого объекта. В программе Google Earth, когда пользователь дважды щелкает по элементу в списке или по значку на трехмерной панормаме, производится «перелет» в соответствующую точку обзора. См. главу 2 «Метки и информационные окна».

Элементы, специфичные для <LookAt>

<longitude>

Долгота точки, на которую мы смотрим из точки обзора. Значением является угловое расстояние в градусах относительно нулевого меридиана. Значения к западу от нулевого меридиана изменяются в диапазоне от -180^0 до 0, значения к востоку – от 0 до 180^0.

<latitude>

Широта точки, на которую мы смотрим из точки обзора. Значением является угловое расстояние в градусах относительно экватора, оно изменяется в диапазоне от -90^0 до 90^0 (с севера на юг).

<altitude>

Высота от земной поверхности в метрах. Об интерпретации значения см. описание элемента <altitudeMode>.

<heading>

Направление (север, юг, восток, запад) в градусах. По умолчанию подразумевается 0 (север). Изменяется в диапазоне от 0 до 360^0.

<tilt>

Угол между направлением на точку обзора и нормалью к поверхности Земли. Изменяется от 0 до 90^0. Значение <tilt> не может быть отрицательным. Значение 0 означает, что точка обзора расположена прямо над обозреваемой точкой, а значение 90^0 – что взгляд направлен к линии горизонта.

<range>

Расстояние в метрах от точки, заданной координатами <longitude>, <latitude> и <altitude>, до точки обзора (рис. 2.5).

<altitudeMode>

Определяет способ интерпретации значения в элементе <altitude>. Допустимы следующие значения: `clampToGround`, `relativeToGround` и `absolute`. Описание этих режимов см. в главе 3 «Геометрия».

Содержится в

Любой элемент, производный от Feature; <NetworkLinkControl>

<Model>

```
<Model id="ID">
  <altitudeMode>clampToGround</altitudeMode>
      <!-- kml:altitudeModeEnum:clampToGround, relativeToGround, absolute -->
  <Location>
    <longitude>0.0</longitude>      <!-- kml:angle180 -->
    <latitude>0.0</latitude>        <!-- kml:angle90 -->
    <altitude>0.0</altitude>        <!-- double -->
  </Location>
  <Orientation>
    <heading>0.0</heading>          <!-- kml:angle360 -->
    <tilt>0.0</tilt>                <!-- kml:anglepos180 -->
    <roll>0.0</roll>                <!-- kml:angle180 -->
  </Orientation>
  <Scale>
    <x>1.0</x>                      <!-- double -->
    <y>1.0</y>                      <!-- double -->
    <z>1.0</z>                      <!-- double -->
  </Scale>
  <Link>...</Link>
  <ResourceMap>
    <Alias>
      <targetHref>...</targetHref>  <!-- anyURI -->
      <sourceHref>...</sourceHref>  <!-- anyURI -->
    </Alias>
  </ResourceMap>
</Model>
```

Описание

Трехмерный объект, описанный в COLLADA-файле (на который ссылается элемент <Link>). COLLADA-файлы имеют расширение *.dae*. Модель создается в собственном пространстве координат, а затем позиционируется и масштабируется в геобраузере. См. главу 3 «Геометрия».

Элементы, специфичные для <Model>

<altitudeMode>

Определяет способ интерпретации значения в элементе <altitude>. Допустимы следующие значения: `clampToGround`, `relativeToGround` и `absolute`. Описание этих режимов см. в главе 3 «Геометрия».

<Location>

Задает точные координаты начала координат модели в элементах <longitude>, <latitude> и <altitude>. Широта и долгота выражаются в стандартной проекции WGS84. Высота измеряется над поверхностью Земли в метрах и интерпретируется согласно параметру <altitudeMode>.

<longitude>

Долгота начала координат модели.

<latitude>

Широта начала координат модели.

<altitude>

Высота начала координат модели.

<Orientation>

Определяет, как нужно повернуть систему координат модели, чтобы она была правильно ориентирована относительно Земли.

<heading>

Угол поворота относительно оси *z* (нормали к поверхности Земли). Значение 0 (по умолчанию) соответствует северу. Положительным считается направление вращения по часовой стрелке, угол поворота измеряется в градусах и изменяется в диапазоне от 0 до ±360.

<tilt>

Угол поворота относительно оси *x*. Положительным считается направление вращения по часовой стрелке, угол измеряется в градусах и изменяется в диапазоне от 0 до ±180.

<roll>

Угол поворота относительно оси *y*. Положительным считается направление вращения по часовой стрелке, угол измеряется в градусах и изменяется в диапазоне от 0 до ±180.

<Scale>

Масштабирование модели по осям *x*, *y* и *z* в пространстве координат модели.

<x>

Коэффициент масштабирования по оси *x*.

<y>

Коэффициент масштабирования по оси *y*.

<z>

Коэффициент масштабирования по оси *z*.

<ResourceMap>

Задает 0 или более элементов <Alias>, каждый из которых определяет соответствие между путями к файлам текстур в исходном COLLADA-файле и в KML- или KMZ-файле, содержащем данную модель. Этот элемент позволяет перемещать и переименовывать файлы текстур, не изменяя исходного COLLADA-файла, который на эти текстуры ссылается. Один элемент <ResourceMap> может содержать несколько отображений различных (исходных) COLLADA-файлов на один и тот же (конечный) KMZ-файл.

<Alias>

Устанавливает соответствие между <sourceHref> и <targetHref>:

<targetHref>

Задает файл текстуры, который должен прочитать геобраузер. Ссылка может быть сформирована относительно файла изображения в KMZ-архиве или являться абсолютным путем к файлу (например, в виде URL).

<sourceHref>

Путь к файлу текстуры, заданный в COLLADA-файле с расширением *.dae*.

Содержится в

<MultiGeometry>, <Placemark>

<MultiGeometry>

```
<MultiGeometry id="ID">
  <!-- 0 или более элементов Geometry -->
</MultiGeometry>
```

Описание

Контейнер для нуля или более геометрических элементов, ассоциированных с одной и той же деталью.

Содержится в

<MultiGeometry>, <Placemark>

Содержит

0 или более геометрических элементов.

<NetworkLink>

```
<NetworkLink id="ID">
  <!-- наследуются от элемента Feature -->
  <name>...</name>                                  <!-- string -->
  <visibility>1</visibility>                        <!-- boolean -->
  <open>0</open>                                    <!-- boolean -->
  <atom:author>...</atom:author>                    <!-- xmlns:atom -->
  <atom:link>...</atom:link>                        <!-- xmlns:atom -->
  <address>...</address>                            <!-- string -->
  <xal:AddressDetails>...</xal:AddressDetails>      <!-- xmlns:xal -->
  <phoneNumber>...</phoneNumber>                    <!-- string -->
  <Snippet maxLines="2">...</Snippet>               <!-- string -->
  <description>...</description>                    <!-- string -->
  <AbstractView>...</AbstractView>                  <!-- Camera или LookAt-->
  <TimePrimitive>...</TimePrimitive>                <!-- TimeSpan или TimeStamp -->
  <styleUrl>...</styleUrl>                          <!-- anyURI -->
  <StyleSelector>...</StyleSelector>                <!-- Style или  StyleMap -->
  <Region>...</Region>
  <ExtendedData>...</ExtendedData>

  <!-- специфичны для NetworkLink -->
  <refreshVisibility>0</refreshVisibility>          <!-- boolean -->
  <flyToView>0</flyToView>                          <!-- boolean -->
  <Link>...</Link>
</NetworkLink>
```

Описание

Ссылается на KML-файл или KMZ-архив, находящийся на локальном компьютере или в Сети. Внутри элемента <Link> можно задать параметры обновления файла. См. главу 6 «Сетевые ссылки» и главу 8 «Большие наборы данных».

Элементы, специфичные для <NetworkLink>

<refreshVisibility>

Значение 0 отдает управление видимостью детали в руки пользователя геобраузера. Значение 1 означает, что видимость детали должна сбрасываться при каждом обновлении сетевой ссылки.

<flyToView>

Значение 0 говорит, что элемент AbstractView следует игнорировать. Значение 1 заставляет геобраузер «перелететь» к точке обзора, заданной в элементе <Camera> или <LookAt>, вложенном в элемент <NetworkLinkControl> в загруженном KML-файле. Подробности см. в главе 6.

<Link>

Задает местоположение KML- или KMZ-файла.

Содержится в

Любой элемент, производный от Container.

<NetworkLinkControl>

```
<NetworkLinkControl>
  <minRefreshPeriod>0.0</minRefreshPeriod>      <!-- double -->
  <maxSessionLength>-1.0</maxSessionLength>     <!-- double -->
  <cookie>...</cookie>                          <!-- string -->
  <message>...</message>                        <!-- string -->
  <linkName>...</linkName>                      <!-- string -->
  <linkDescription>...</linkDescription>        <!-- string -->
  <linkSnippet maxLines="2">...</linkSnippet>   <!-- string -->
  <expires>...</expires>                        <!-- kml:dateTime -->
  <Update>...</Update>                          <!-- Change, Create, Delete -->
  <AbstractView>...</AbstractView>              <!-- <LookAt> или <Camera> -->
</NetworkLinkControl>
```

Описание

Управляет поведением файлов, загруженных по сетевой ссылке. Когда файл загружается по сетевой ссылке <NetworkLink>, обрабатываются все элементы <NetworkLinkControl> в загруженном файле. Элементы <linkName>, <linkDescription> и <linkSnippet> относятся к файлу, содержащему исходную ссылку <NetworkLink> (см. главу 6 «Сетевые ссылки»).

Элементы, специфичные для <NetworkLinkControl>

<minRefreshPeriod>

Задает минимальное время (в секундах) между последовательными операциями загрузки файла. Этот параметр позволяет серверу задерживать отправку файла, выбирая темп обновления, соответствующий ожидаемой частоте изменения данных.

<maxSessionLength>

Задает максимальное время в секундах, в течение которого сохраняется соединение между клиентом и сервером. Принимаемое по умолчанию значение –1 означает, что соединение никогда не разрывается принудительно.

<cookie>

Этот элемент позволяет дописать некоторую строку в конец строки запроса URL при следующем обновлении сетевой ссылки. Эти данные можно использовать в серверном сценарии для уточнения алгоритма обработки, в частности для опроса номера версии и условной отправки файла. См. главу 6 «Сетевые ссылки».

<message>

Вы можете доставить сообщение, которое отображается в диалоговом окне, например указания по использованию сетевой ссылки. Сообщение появляется при первой загрузке ссылки геобраузером, а также при ее изменении в элементе <NetworkLinkControl>.

<linkName>

Вы можете управлять именем сетевой ссылки с сервера, так что изменения, произведенные на стороне клиента, переопределяются сервером.

<linkDescription>

Вы можете управлять описанием сетевой ссылки с сервера, так что изменения, произведенные на стороне клиента, переопределяются сервером.

<linkSnippet maxLines=»2">

Вы можете управлять кратким описанием сетевой ссылки с сервера, так что изменения, произведенные на стороне клиента, переопределяются сервером. Атрибут `maxLines` – это целое число, задающее максимальное количество отображаемых строк.

<expires>

Вы можете задать дату и время, когда ссылка должна быть обновлена. Эта спецификация принимается во внимание, только если параметр <refreshMode> внутри элемента <Link>, вложенного в <NetworkLink>, равен `onExpire`. См. описание потомка <refreshMode> элемента <Link>.

<Update>

Элемент <Update> позволяет задать последовательность элементов <Change>, <Create> и <Delete> для ранее загруженного по сетевой ссылке KML- или KMZ-файла. Потомки элемента <Update> обрабатываются в порядке следования. См. также главу 7 «Динамический KML».

<targetHref><Snippet>

Определяет подлежащий обновлению KML-файл, который должен находиться на том же сервере, где и файл с обновлениями.

<Change>

Изменяет значения в элементе KML, который уже был загружен по сетевой ссылке. Потомком элемента <Change> может быть любой элемент, производный от Object. Этому элементу должен соответствовать какой-то элемент в исходном KML-файле, которому приписан идентификатор `id`. Идентификатор обновляемого элемента указывается в потомке <Change> с помощью атрибута `targetId`. В модифицируемом элементе заменяются только значения, перечисленные внутри <Change>, остальные остаются без изменения. Если элемент <Change> применяется к набору координат, то текущие координаты заменяются новыми.

<Create>

Добавляет элемент в KML-файл, ранее загруженный по сетевой ссылке. Потомком элемента <Create> *должен* быть элемент, производный от Container (то есть <Document> или <Folder>). Этому элементу должен соответствовать какой-то элемент в исходном KML-файле, которому приписан идентификатор `id`. Потомок <Create> – это подлежащий созданию элемент, у него должен быть атрибут `targeted`, равный `id` элемента в исходном файле. После выполнения обновления новый элемент становится частью исходного файла. Для последующих обновлений этого элемента следует указывать URL исходного KML-файла.

<Delete>

Удаляет элемент из KML-файла, ранее загруженного по сетевой ссылке. Потомком элемента <Delete> *должен* быть элемент, производный от Feature (<NetworkLink>, <Placemark>, <GroundOverlay>, <PhotoOverlay>, <ScreenOverlay>, <Document> или <Folder>). Этому элементу должен соответствовать какой-то элемент в исходном KML-файле, которому приписан идентификатор `id`. Потомок <Delete> – это подлежащий удалению элемент, у него должен быть атрибут `targeted`, равный `id` элемента в исходном файле.

<AbstractView>

Если параметр <flyToView> вызывающего элемента <NetworkLink> равен 1, то геобраузер совершает «перелет» к точке обзора, указанной в данном элементе <Camera> или <LookAt>.

Содержится в

<kml>

См. также

<NetworkLink>

Object

Раздел 8.1 спецификации OGC KML 07-147

```
<!-- абстрактный элемент, непосредственно не создается -->
<!-- Object id="ID" targetId="NCName" -->
<!-- /Object -->
```

Описание

Это абстрактный базовый класс, который невозможно создать в KML-файле непосредственно. Он предоставляет атрибут `id`, уникально идентифицирующий элемент KML, и атрибут `targetId`, позволяющий сослаться на объект, уже загруженный в геобраузер и являющийся целью операции обновления. Если вы собираетесь пользоваться механизмом <Update> для обновления некоторого объекта, то ему обязательно должен быть присвоен атрибут `id`.

Overlay

Раздел 11.1 спецификации OGC KML 07-147

```
<!-- Overlay -->                    <!-- GroundOverlay, PhotoOverlay, ScreenOverlay -->

  <!-- наследуются от элемента Feature -->
  <name>...</name>                                <!-- string -->
  <open>0</open>                                  <!-- boolean -->
  <visibility>1</visibility>                      <!-- boolean -->
  <atom:author>...</atom:author>                  <!-- xmlns:atom -->
  <atom:link />                                   <!-- xmlns:atom -->
  <address>...</address>                          <!-- string -->
  <xal:AddressDetails...</xal:AddressDetails>     <!-- xmlns:xal -->
  <phoneNumber>...</phoneNumber>                  <!-- string -->
  <Snippet maxLines="2">...</Snippet>             <!-- string -->
  <description>...</description>                  <!-- string -->
  <AbstractView>...</AbstractView>                <!-- LookAt или Camera -->
```

```
  <TimePrimitive>...</TimePrimitive>          <!-- TimeSpan или TimeStamp -->
  <styleUrl>...</styleUrl>                    <!-- anyURI -->
  <StyleSelector>...</StyleSelector>          <!-- Style или StyleMap -->
  <Region>...</Region>
  <ExtendedData>...</ExtendedData>

  <!-- специфичны для элемента Overlay -->
  <color>ffffffff</color>                     <!-- kml:color -->
  <drawOrder>0</drawOrder>                    <!-- int -->
  <Icon>
    <href>...</href>                          <!-- anyURI -->
    <refreshMode>onChange</refreshMode>
        <!-- kml:refreshModeEnum: onChange, onInterval или onExpire -->
    <refreshInterval>4.0</refreshInterval>    <!-- double -->
    <viewRefreshMode>never</viewRefreshMode>
        <!-- kml:viewRefreshModeEnum: never, onStop, onRequest, onRegion -->
    <viewRefreshTime>4</viewRefreshTime>      <!-- double -->
    <viewBoundScale>1</viewBoundScale>        <!-- double -->
    <viewFormat>...</viewFormat>              <!-- string -->
    <httpQuery>...</httpQuery>                <!-- string -->
  </Icon>

<!-- /Overlay -->
```

Описание

Это абстрактный элемент, который нельзя использовать в KML-файле непосредственно. Он является базовым типом для изображений, накладываемых на поверхность планеты или на экран. Элемент <Icon> задает собственно изображение и может быть сконфигурирован так, чтобы это изображение перезагружалось по таймеру или при смене положения камеры. Этот элемент включает еще параметры, управляющие относительным порядком расположения нескольких наложений, а также цветом и прозрачностью базового изображения.

Элементы, специфичные для Overlay

<color>

Значение цвета вместе с прозрачностью записывается в шестнадцатеричном виде в порядке *aabbggrr*, где *aa* – альфа-канал, *bb* – синяя, *gg* – зеленая, *rr* – красная составляющая. Каждая компонента изменяется в диапазоне от 0 до 255 (от 00 до ff). Если альфа-канал равен 00, то цвет полностью прозрачен, а если ff – то полностью непрозрачен.

<drawOrder>

Задает взаимный порядок перекрывающихся изображений. Порядок определяется относительно камеры и не имеет ничего общего с высотой наложений. Наложение с максимальным значением <drawOrder> прорисовывается последним (то есть оказывается ближайшим к камере).

<Icon>

Задает изображение, ассоциированное с наложением. См. <Link>.

<href>

Определяет, где находится накладываемое изображение. Файл может находиться как в локальной файловой системе, так и на веб-сервере. Если элемент <Icon> опущен или не содержит <href>, то рисуется прямоугольник того цвета и размера, которые заданы в рельефном или экранном наложении.

<refreshMode>

Описание <refreshMode> и прочих перечисленных ниже элементов см. в разделе <Link>.

<refreshInterval>

<viewRefreshMode>

<viewRefreshTime>

<viewBoundScale>

<viewFormat>

<httpQuery>

<PhotoOverlay>

```
<PhotoOverlay>
  <!-- наследуются от элемента Feature -->
  <name>...</name>                                <!-- string -->
  <open>0</open>                                  <!-- boolean -->
  <visibility>1</visibility>                      <!-- boolean -->
  <atom:author>...</atom:author>                  <!-- xmlns:atom -->
  <atom:link />                                   <!-- xmlns:atom -->
  <address>...</address>                          <!-- string -->
  <xal:AddressDetails...</xal:AddressDetails>     <!-- xmlns:xal -->
  <phoneNumber>...</phoneNumber>                  <!-- string -->
  <Snippet maxLines="2">...</Snippet>             <!-- string -->
  <description>...</description>                  <!-- string -->
  <AbstractView>...</AbstractView>                <!-- LookAt или Camera -->
  <TimePrimitive>...</TimePrimitive>              <!-- TimeSpan или TimeStamp -->
  <styleUrl>...</styleUrl>                        <!-- anyURI -->
  <StyleSelector>...</StyleSelector>              <!-- Style или StyleMap -->
  <Region>...</Region>
  <ExtendedData>...</ExtendedData>

  <!-- наследуются от элемента Overlay -->
  <color>ffffffff</color>                         <!-- kml:color -->
  <drawOrder>0</drawOrder>                        <!-- int -->
  <Icon>
    <href>...</href>                              <!-- anyURI -->
    ...
  </Icon>

  <!-- специфичны для элемента Overlay -->
  <rotation>0.0</rotation>                        <!-- kml:angle180 -->
  <ViewVolume>
    <leftFov>0.0</leftFov>                        <!-- kml:angle180 -->
    <rightFov>0.0</rightFov>                      <!-- kml:angle180 -->
    <bottomFov>0.0</bottomFov>                    <!-- kml:angle90 -->
    <topFov>0.0</topFov>                          <!-- kml:angle00 -->
    <near>0.0</near>                              <!-- double -->
  </ViewVolume>
  <ImagePyramid>
    <tileSize>256</tileSize>                      <!-- int -->
    <maxWidth>0</maxWidth>                        <!-- int -->
    <maxHeight>0</maxHeight>                      <!-- int -->
    <gridOrigin>lowerLeft</gridOrigin>            <!-- lowerLeft или upperLeft-->
  </ImagePyramid>
  <Point>...</Point>
  <shape>rectangle<shape>                         <!-- rectangle, cylinder, sphere -->
</PhotoOverlay>
```

Описание

Элемент <PhotoOverlay> позволяет расположить фотографию на поверхности Земли и задать параметры ее просмотра. Фотоналожение может быть прос-

тым двумерным прямоугольником, частичным или полным цилиндром либо сферой (для сферических панорам). Наложение располагается в заданной точке и ориентировано в направлении точки обзора.

Поскольку элемент <PhotoOverlay> является производным от Feature, он может содержать один из двух элементов, производных от AbstractView: <Camera> или <LookAt>. Элемент <Camera> (как и <LookAt>) задает точку и направление обзора (называемое также *вектором взгляда*). Фотоналожение позиционируется относительно точки обзора. Точнее, плоскость двумерного прямоугольного изображения перпендикулярна вектору взгляда.

URL самого изображения задается в элементе <Icon>, который должен содержать элемент <href> с адресом файла изображения. Если изображение очень велико, то <href> содержит специальный URL, служащий индексом пирамиды изображений с различным разрешением. См. главу 5 «Наложения».

Элементы, специфичные для <PhotoOverlay>

<rotation>

Корректирует расположение фотографии в поле зрения. Полезен в случае, если фотография была повернута и слегка отклоняется от требуемого горизонтального положения.

<ViewVolume>

Определяет, какая часть текущей сцены видима. Задание поля зрения аналогично заданию диаметра отверстия объектива в физической камере. Если поле зрения мало, как в телескопическом объективе, то взгляд фокусируется на небольшом участке сцены. А если велико, как в широкоугольном объективе, то взгляд охватывает большую часть сцены.

<leftFov>

Угол в градусах между направлением взгляда камеры и левой стороной поля зрения.

<rightFov>

Угол в градусах между направлением взгляда камеры и правой стороной поля зрения.

<bottomFov>

Угол в градусах между направлением взгляда камеры и нижней стороной поля зрения.

<topFov>

Угол в градусах между направлением взгляда камеры и верхней стороной поля зрения.

<near>

Расстояние в метрах вдоль направления взгляда между точкой обзора камеры и фотоналожением.

<ImagePyramid>

Для очень больших изображений необходимо построить пирамиду, то есть иерархически организованный набор изображений, каждое из которых пред-

ставляет собой вариант исходного изображения с постепенно уменьшающимся разрешением. Все изображения в пирамиде разбиты на плитки, так чтобы картинку можно было загружать не целиком, а по частям. Геобраузер вычисляет текущую точку обзора и загружает лишь те плитки, разрешение которых соответствует расстоянию до зрителя.

При задании пирамиды изображений необходимо также модифицировать элемент <href> внутри <Icon>, включив в него спецификацию тех плиток, которые нужно загружать. См. главу 5 «Наложения».

<tileSize>

Размер плитки в пикселях. Плитки должны быть квадратными, а <tileSize> – степенью двойки. Рекомендуется выбирать размер 256 (по умолчанию) или 512. Исходное изображение разбивается на плитки указанного размера с разными разрешениями.

<maxWidth>

Ширина исходного изображения в пикселях.

<maxHeight>

Высота исходного изображения в пикселях.

<gridOrigin>

Задает начало нумерации плиток в каждом слое пирамиды. Значение `lowerLeft` (по умолчанию) означает, что плитка в строке 1 и столбце 1 должна считаться левым нижним углом сетки.

<Point>

Элемент <Point> выполняет те же функции, что одноименный элемент внутри тега <Placemark>. Он служит для рисования значка, отмечающего позицию фотоналожения. Стиль значка определяется элементами <styleUrl> и <StyleSelector> точно так же, как для <Placemark>.

<coordinates>

Точка, в которой рисуется значок данного фотоналожения.

<shape>

Фотоналожение проецируется на форму, определяемую элементом <shape>, который может принимать следующие значения:

`rectangle` (по умолчанию)

Для обычных фотографий.

`cylinder`

Для панорамных фотографий, спроецированных на цилиндрическую поверхность.

`sphere`

Для сферических панорамных фотографий.

Содержится в

<Folder>, <Document> или <kml>

<Placemark>

```
<Placemark id="ID">
  <!-- наследуются от элемента Feature -->
  <name>...</name>                                  <!-- string -->
  <visibility>1</visibility>                        <!-- boolean -->
  <open>0</open>                                    <!-- boolean -->
  <atom:author>...</atom:author>                    <!-- xmlns:atom -->
  <atom:link />                                     <!-- xmlns:atom -->
  <address>...</address>                            <!-- string -->
  <xal:AddressDetails>...</xal:AddressDetails>      <!-- xmlns:xal -->
  <phoneNumber>...</phoneNumber>                    <!-- string -->
  <Snippet maxLines="2">...</Snippet>               <!-- string -->
  <description>...</description>                    <!-- string -->
  <AbstractView>...</AbstractView>                  <!-- Camera или LookAt -->
  <TimePrimitive>...</TimePrimitive>                <!-- TimeSpan или TimeStamp -->
  <styleUrl>...</styleUrl>                          <!-- anyURI -->
  <StyleSelector>...</StyleSelector>                <!-- Style или StyleMap -->
  <Region>...</Region>
  <ExtendedData>...</ExtendedData>

  <!-- специфичны для элемента Placemark -->
  <Geometry>...</Geometry>
      <!-- Point, LineString, LinearRing, Polygon, MultiGeometry, Model -->
</Placemark>
```

Описание

Элемент <Placemark> (метка) – это деталь Feature, с которой ассоциирован геометрический объект. В Google Earth метка присутствует в списке на панели **Метки**. Элементу <Placemark>, в который вложен элемент <Point>, соответствует значок, отмечающий эту точку на трехмерной панораме Земли.

Отметим, что <Placemark> – единственный элемент в KML, куда можно помещать геометрические объекты.

На трехмерной панораме в Google Earth точечная метка – единственный объект, по которому можно щелкнуть или накатить мышь. Другие геометрические объекты не представлены значками на панораме. Чтобы дать пользователю возможность по чему-то щелкнуть мышью, вы должны создать элемент MultiGeometry, содержащий одновременно <Point> и какой-то другой геометрический объект. См. также <Icon>.

Содержится в

<Document>, <Folder>

Содержит

0 или 1 элемент <Geometry> (с одной меткой можно ассоциировать несколько геометрических элементов, воспользовавшись тегом <MultiGeometry>).

<Point>

```
<Point id="ID">
  <!-- специфичны для элемента Point -->
  <extrude>0</extrude>                                  <!-- boolean -->
  <altitudeMode>clampToGround</altitudeMode>
      <!-- kml:altitudeModeEnum:clampToGround, relativeToGround, absolute -->
  <coordinates>...</coordinates>                        <!-- lon,lat[,alt] -->
</Point>
```

Описание

Географическое местоположение, определяемое долготой, широтой и высотой (необязательной). Если высота не задана, предполагается, что она равна 0. Если элемент <Point> вложен в <Placemark>, то эта точка определяет позицию имени и значка метки. Говорят, что точка экструдирована, если она соединена линией с поверхностью Земли. Эта линия рисуется текущим стилем <LineStyle>.

Элементы, специфичные для <Point>

<extrude>

Определяет, соединена ли точка линией с поверхностью Земли. Экструзия возможна, только если параметр <altitudeMode> равен `relativeToGround` или `absolute`, при этом высота в элементе <coordinates> должна быть больше 0 (то есть точка находится над землей). Экструдированная линия соединяет точку с центром земной сферы.

<altitudeMode>

Определяет способ интерпретации компонентов элемента <coordinates>. Возможные значения: `clampToGround`, `relativeToGround`, `absolute`. См. главу 3 «Геометрия», где эти режимы обсуждаются подробно.

<coordinates>

Кортеж, состоящий из чисел с плавающей точкой, которые соответствуют долготе, широте и высоте (именно в таком порядке). Долгота и широта выражаются в градусах, высота (необязательная) – в метрах над уровнем моря. По умолчанию высота равна 0. Значение высоты игнорируется, если режим <altitudeMode> задан равным `clampToGround`. Между отдельными компонентами кортежа не должно быть пробелов.

долгота	от –180 до 180
широта	от –90 до 90

Содержится в

<MultiGeometry>, <Placemark>

<Polygon>

```
<Polygon id="ID">
  <extrude>0</extrude>                <!-- boolean -->
  <tessellate>0</tessellate>          <!-- boolean -->
  <altitudeMode>clampToGround</altitudeMode>
      <!-- kml:altitudeModeEnum: clampToGround
           relativeToGround или absolute -->
  <outerBoundaryIs>
    <LinearRing>
      <coordinates>...</coordinates> <!-- lon,lat[,alt] -->
    </LinearRing>
  </outerBoundaryIs>                  <!-- кортежи lon,lat[,alt] -->
  <innerBoundaryIs>                   <!-- может включать несколько внутренних границ -->
    <LinearRing>
      <coordinates>...</coordinates> <!-- lon,lat[,alt] -->
    </LinearRing>
  </innerBoundaryIs>                  <!-- кортежи lon,lat[,alt] -->
</Polygon>
```

Описание

Многоугольник <Polygon> определяется внешней границей и 0 или более внутренними границами. Границы являются замкнутыми ломаными <LinearRing>. Если многоугольник экструдирован, то его граница продлевается до поверхности Земли, образуя нечто вроде стены или забора. Экструдированная граница рисуется в соответствии со стилем <PolStyle>, который задает цвет, режим цвета и заливку.

Элементы, специфичные для <Polygon>

<extrude>

Определяет, нужно ли соединять многоугольник с поверхностью Земли. Для экструдирования многоугольника параметр <altitudeMode> должен быть равен `relativeToGround` или `absolute`, а компоненты высоты в элементе <coordinates> должны быть больше 0 (то есть многоугольник располагается над землей). Экструдируются только вершины многоугольника, но не его внутренняя часть (так, прямоугольник превращается в параллелепипед с пятью гранями). Линии, исходящие из вершин, направлены к центру земной сферы.

<tesselate>

Определяет, должен ли многоугольник повторять кривизну Земли. Чтобы тесселяция была возможна, параметр <altitudeMode> должен быть равен `clampToGround`. Для очень протяженных многоугольников тесселяция должна быть включена, иначе стороны уйдут под землю и не будут видны.

<altitudeMode>

Способ интерпретации высоты в элементах <coordinates>. Допустимые значения: `clampToGround`, `relativeToGround` и `absolute`. Эти режимы рассматриваются в главе 3 «Геометрия».

<outerBoundaryIs>

Содержит элемент <LinearRing>.

<innerBoundaryIs>

Содержит элемент <LinearRing>. Многоугольник может содержать несколько элементов <innerBoundaryIs>, которые образуют дырки в нем.

<LinearRing>

См. раздел <LinearRing>.

Содержится в

<MultiGeometry>, <Placemark>

<PolyStyle>

```
<PolyStyle id="ID">
  <!-- наследуются от элемента ColorStyle -->
  <color>ffffffff</color>                          <!-- kml:color -->
  <colorMode>normal</colorMode>
      <!-- kml:colorModeEnum: normal или random -->

  <!-- специфичны для элемента ColorStyle -->
  <fill>1</fill>                                   <!-- boolean>
  <outline>1</outline>                             <!-- boolean>
</PolyStyle>
```

Описание

Задает стиль рисования всех многоугольников, в том числе и режим полигональной экструзии (в котором многоугольник выглядит как стена здания) или линейной экструзии (выглядит как сплошной забор).

Элементы, специфичные для <PolyStyle>

<fill>

Задает способ заливки многоугольника.

<outline>

Определяет, нужно ли оконтурировать многоугольник. Контур рисуется текущим стилем <LineStyle>.

Содержится в

<Style>

<Region>

```
<Region id="ID">
  <LatLonAltBox id="ID">
    <north>180.0</north>                          <!-- kml:angle180 -->
    <south>-180.0</south>                         <!-- kml:angle180 -->
    <east>180.0</east>                            <!-- kml:angle180 -->
    <west>180.0</west>                            <!-- kml:angle180 -->
    <minAltitude>0.0</minAltitude>                <!-- double -->
    <maxAltitude>0.0</maxAltitude>                <!-- double -->
    <altitudeMode>clampToGround</altitudeMode>
      <!-- kml:altitudeModeEnum: clampToGround
           relativeToGround или absolute -->
  </LatLonAltBox>
  <Lod id="ID">
    <minLodPixels>0.0</minLodPixels>              <!-- double -->
    <maxLodPixels>-1.0</maxLodPixels>             <!-- double -->
    <minFadeExtent>0.0</minFadeExtent>            <!-- double -->
    <maxFadeExtent>0.0</maxFadeExtent>            <!-- double -->
  </Lod>
</Region>
```

Описание

Элемент <Region> содержит ограничивающий параллелепипед (<LatLonAltBox>), который описывает представляющую интерес область с помощью географических координат и высоты. Кроме того, <Region> содержит элемент <Lod> (level of detail – уровень детализации), определяющий диапазон достоверности региона в терминах размера экранной проекции. Говорят, что регион «активен», если ограничивающий параллелепипед находится в поле зрения пользователя и удовлетворяются требования LOD. Геометрические объекты и наложения, ассоциированные с регионом, рисуются лишь в том случае, когда регион активен. Если параметр <viewRefreshMode> равен `onRegion`, то ссылка <Link> или значок <Icon> загружаются лишь при условии активности региона. В иерархии элемента Container или <NetworkLink> регион <Region> (если таковой имеется) оказывает влияние на содержащую его деталь Feature. Если Feature не содержит элемента

<Region>, то используется элемент <Region>, вложенный в ближайшего предка. См. главу 8 «Большие наборы данных».

Элементы, специфичные для <Region>

<LatLonAltBox>

Задает параллелепипед, ограничивающий область, представляющую интерес. Ниже описаны значения по умолчанию и обязательные поля.

<north>

Широта северной стороны ограничивающего параллелепипеда. Выражена в градусах и принимает значение от –90 до +90.

<south>

Широта южной стороны ограничивающего параллелепипеда. Выражена в градусах и принимает значение от –90 до +90.

<east>

Долгота восточной стороны ограничивающего параллелепипеда. Выражена в градусах и принимает значение от –180 до +180.

<west>

Долгота западной стороны ограничивающего параллелепипеда. Выражена в градусах и принимает значение от –180 до +180.

<minAltitude>

Минимальная высота ограничивающего параллелепипеда, заданная в метрах и интерпретируемая в соответствии с <altitudeMode>.

<maxAltitude>

Максимальная высота ограничивающего параллелепипеда, заданная в метрах и интерпретируемая в соответствии с <altitudeMode>.

<altitudeMode>

Определяет способ интерпретации <minAltitude> и <maxAltitude>. Может принимать значения `clampToGround`, `relativeToGround` или `absolute`. Обсуждение этих режимов см. в главе 3 «Геометрия».

<Lod>

Lod означает «Level of Detail» (уровень детализации). Элемент <Lod> описывает размер проекции региона на экран, необходимый для того, чтобы регион считался «активным». Кроме того, в нем задаются выраженные в пикселях расстояния, на которых происходит переход между прозрачностью и непрозрачностью.

<minLodPixels>

Количество экранных пикселей, минимально необходимое для того, чтобы данный регион был видим. Геобраузер вычисляет размер проекции региона на экран, а затем – квадратный корень из площади региона. (Если, например, регион квадратный, не наклонен и точка обзора находится прямо над ним, то эта величина будет равна длине стороны проекции региона.) Если вычисленная величина попадает в диапазон между <minLodPixels> и <maxLodPixels> (и при этом прямоугольник <LatLonAltBox> находится в поле зрения), то регион «активен».

<maxLodPixels>

Количество экранных пикселей, при превышении которого регион невидим. Значение –1, принимаемое по умолчанию, означает, что «регион активен, как бы велика ни была площадь его проекции».

<minFadeExtent>

Расстояние, на котором деталь переходит от полной прозрачности к полной непрозрачности и наоборот. Выражается в экранных пикселях и относится к левому концу диапазона уровней детализации.

<maxFadeExtent>

Расстояние, на котором деталь переходит от полной прозрачности к полной непрозрачности и наоборот. Выражается в экранных пикселях и относится к правому концу диапазона уровней детализации.

Содержится в

Любой элемент, производный от Feature.

<Schema>

```
<Schema name="string" id="ID">
  <SimpleField type="string" name="string">
    <displayName>...</displayName>                <!-- string -->
    <value>...</value>
  </SimpleField>
</Schema>
```

Описание

Определяет схему KML, применяемую для добавления данных нестандартных типов в детали Feature. Атрибут `id` обязателен и должен быть уникален в пределах KML-файла. См. главу 8 «Большие наборы данных».

Элементы, специфичные для <Schema>

<**SimpleField** type=" " name=" ">

Объявление нестандартного поля, в котором должны присутствовать как тип, так и имя этого поля. Если хотя бы один атрибут отсутствует, то поле игнорируется. Тип может принимать следующие значения: `bool`, `double`, `float`, `int`, `uint`, `short`, `ushort`, `string`.

<displayName>

Необязательное имя поля, которое отображается геобраузером. Для экранирования стандартной HTML-разметки пользуйтесь элементом [CDATA].

Содержится в

<Document>

См. также

<ExtendedData>

<ScreenOverlay>

```
<ScreenOverlay id="ID">
  <!-- наследуются от элемента Feature -->
  <name>...</name>                                <!-- string -->
  <open>0</open>                                  <!-- boolean -->
  <visibility>1</visibility>                      <!-- boolean -->
  <atom:author>...</atom:author>                  <!-- xmlns:atom -->
  <atom:link />                                   <!-- xmlns:atom -->
  <address>...</address>                          <!-- string -->
  <xal:AddressDetails...</xal:AddressDetails>     <!-- xmlns:xal -->
  <phoneNumber>...</phoneNumber>                  <!-- string -->
  <Snippet maxLines="2">...</Snippet>             <!-- string -->
  <description>...</description>                  <!-- string -->
  <AbstractView>...</AbstractView>                <!-- LookAt или Camera -->
  <TimePrimitive>...</TimePrimitive>              <!-- TimeSpan или TimeStamp -->
  <styleUrl>...</styleUrl>                        <!-- anyURI -->
  <StyleSelector>...</StyleSelector>              <!-- Style или StyleMap -->
  <Region>...</Region>
  <ExtendedData>...</ExtendedData>

  <!-- наследуются от элемента Overlay -->
  <color>ffffffff</color>                         <!-- kml:color -->
  <drawOrder>0</drawOrder>                        <!-- int -->
  <Icon>...</Icon>

  <!-- специфичны для элемента GroundOverlay -->
  <overlayXY x="double" y="double" xunits="fraction" yunits="fraction"/>
      <!-- vec2 -->
      <!-- xunits и yunits могут принимать значения:
           fraction, pixels или insetPixels -->
  <screenXY x="double" y="double" xunits="fraction" yunits="fraction"/>
      <!-- vec2 -->
  <rotationXY x="double" y="double" xunits="fraction" yunits="fraction"/>
      <!-- vec2 -->
  <size       x="double" y="double" xunits="fraction" yunits="fraction"/>
      <!-- vec2 -->
  <rotation>0.0</rotation>                        <!-- kml:angle180 -->
</ScreenOverlay>
```

Описание

Этот элемент рисует накладываемое изображение, зафиксированное на экране. Обычно таким образом рисуют компас, логотип или «данные на лобовом стекле». Размер экранного наложения определяется элементом <size>. Позиция выбирается так, чтобы точка изображения с координатами <overlayXY> совместилась с точкой на экране с координатами <screenXY>. Затем изображение поворачивается на <rotation> градусов вокруг точки на экране с координатами <rotationXY>.

Потомок <href> элемента <Icon> задает адрес файла с накладываемым изображением. Файл может быть локальным или находиться на веб-сервере. Если элемент <Icon> опущен или не содержит <href>, то рисуется прямоугольник цвета и размера, определяемого экранным наложением.

Элементы, специфичные для <ScreenOverlay>

<overlayXY x="double" y="double" xunits="fraction" yunits="fraction"/>

Задает точку внутри или вне накладываемого изображения, которая должна совмещаться с точкой на экране, заданной в элементе <screenXY>. Начало ко-

ординат находится в левом нижнем углу изображения. Подробное описание атрибутов `xunits` и `yunits` см. в главе 5 «Наложения».

<screenXY x="double" y="double" xunits="fraction" yunits="fraction"/>

Задает точку на экране, которая должна совмещаться с точкой изображения, заданной в элементе <overlayXY>. Начало экранной системы координат находится в левом нижнем углу экрана.

<rotationXY x="double" y="double" xunits="fraction" yunits="fraction"/>

Задает точку в координатах экрана, вокруг которой поворачивается экранное наложение.

<size x="double" y="double" xunits="fraction" yunits="fraction"/>

Задает размер накладываемого изображения:

–1 означает, что изменять размер изображения вообще не нужно;

0 означает, что следует сохранить пропорции;

n задает коэффициент масштабирования в направлении соответствующей оси.

<rotation>

Задает угол поворота накладываемого изображения. 0 означает, что поворачивать не надо. Значение выражается в градусах и отсчитывается против часовой стрелки от направления на север (0). Допустимый диапазон – от –180 до +180 градусов. Центр вращения задается в элементе <rotationXY>, а по умолчанию равен (.5,.5).

Содержится в

<Document>, <Folder>

<Style>

```
<Style id="ID">
  <IconStyle>...</IconStyle>
  <LabelStyle>...</LabelStyle>
  <LineStyle>...</LineStyle>
  <PolyStyle>...</PolyStyle>
  <BalloonStyle>...</BalloonStyle>
  <ListStyle>...</ListStyle>
</Style>
```

Описание

Элемент <Style> определяет адресуемую группу стилей, на которые можно ссылаться из элементов <StyleMap> и Feature. Стили влияют на представление геометрических объектов на трехмерной панораме и внешний вид деталей в списке на панели **Метки**. Разделяемые стили вкладываются в элемент <Document>, каждый из них должен иметь атрибут `id`, по которому можно сослаться на стиль из детали, где он используется.

Для ссылки на стиль по идентификатору служит элемент <styleUrl>.

Содержится в

Любой элемент, производный от Feature.

Содержит

<IconStyle>, <LabelStyle>, <LineStyle>, <PolyStyle>, <BalloonStyle>, <ListStyle>

<StyleMap>

```
<StyleMap id="ID">
  <Pair  id="ID">
    <key>normal</key>
        <!-- kml:styleStateEnum: normal или highlight -->
    <styleUrl>...</styleUrl> или <Style>...</Style>
  </Pair>
</StyleMap>
```

Описание

Элемент <StyleMap> устанавливает соответствие между двумя стилями. Обычно он используется для создания обычного и выделенного стиля метки, так чтобы выделенная версия применялась, когда пользователь наводит мышь на значок в геобраузере. См. главу 4 «Стили и значки».

Элементы, специфичные для StyleMap>

<Pair>

Определяет пару имя/значение, сопоставляющую режим (обычный или выделенный) ранее определенному элементу <styleMap>. Элемент <Pair> содержит два элемента (оба обязательные):

<key>

Задает режим: `normal` или `highlight`.

<styleUrl>

Ссылка на стиль. Ссылки на локальные стили, определенные в том же KML-документе, содержат идентификатор стиля, которому предшествует знак #. Для ссылок на стили, определенные во внешнем файле, применяется полный URL в дополнение к #-фрагменту.

Содержится в

Любой элемент, производный от Feature.

StyleSelector *Раздел 12.1 спецификации OGC KML 07-147*

```
<!-- StyleSelector id="ID" -->                <!-- Style или StyleMap -->
<!-- /StyleSelector -->
```

Описание

Это абстрактный элемент, который не может использоваться в KML-файле непосредственно. Он является базовым для <Style> и <StyleMap>. См. главу 4 «Стили и значки».

См. также

<Style>, <StyleMap>

TimePrimitive

Раздел 15.1 спецификации OGC KML 07-147

```
<!-- TimePrimitive id="ID" -->                    <!-- TimeSpan, TimeStamp -->
<!-- /TimePrimitive -->
```

Описание

Это абстрактный элемент, который не может использоваться в KML-файле непосредственно. Он является базовым для <TimeSpan> и <TimeStamp>.

См. также

<TimeSpan>, <TimeStamp>

<TimeSpan>

```
<TimeSpan id="ID">
  <begin>...</begin>      <!-- kml:dateTime -->
  <end>...</end>          <!-- kml:dateTime -->
</TimeSpan>
```

Описание

Представляет промежуток времени, ограниченный с двух сторон значениями типа `dateTime`.

Если какой-то из элементов <begin> или <end> отсутствует, промежуток считается не ограниченным с соответствующей стороны.

Тип `dateTime` определяется спецификацией XML Schema (см. документ «XML Schema Part 2: Datatypes Second Edition»). Значение записывается в виде **yyyy-mm-dd**T**hh:mm:ss**zzzzzz, где `T` – разделитель даты и времени, а часовой пояс равен либо *Z* (UTC-время), либо смещению относительно UTC (*zzzzzz*) в формате ±*hh:mm*. Можно также задавать только дату. См. главу 7 «Динамический KML».

Элементы, специфичные для <TimeSpan>

<begin>

Задает начало промежутка времени. В случае отсутствия промежуток считается не ограниченным слева.

<end>

Задает начало промежутка времени. В случае отсутствия промежуток считается не ограниченным справа.

Содержится в

Любой элемент, производный от Feature.

<TimeStamp>

```
<TimeStamp id="ID">
  <when>...</when>        <!-- kml:dateTime -->
</TimeStamp>
```

Описание

Представляет один момент времени. Тип `dateTime` определяется спецификацией XML Schema (см. документ «XML Schema Part 2: Datatypes Second Edition»). Точность временного штампа определяется значением типа `dateTime` в элементе `<when>`. См. главу 7 «Динамический KML».

Элементы, специфичные для <TimeStamp>

<when>

Задает момент времени. Значение имеет тип `dateTime` и может быть представлено в любой из следующих форм:

- dateTime (*YYYY-MM-DDThh:mm:ssZ*) – время с точностью до секунды;
- date (*YYYY-MM-DD*) – дата с точностью до дня;
- gYearMonth (*YYYY-MM*) – дата с точностью до месяца;
- gYear (*YYYY*) – год.

Содержится в

Любой элемент, производный от Feature.

Приложение В
Астрономические данные в KML

Можно создавать KML-файлы для отображения небесных объектов: звезд, созвездий, планет, Луны и галактик. В настоящем приложении объясняется, как создать такой KML-файл для геобраузера, поддерживающего астрономические данные, например Google Earth или Microsoft WorldWide Telescope. Сделать нужно две вещи:

- включить в элемент <kml>, располагающийся в начале KML-файла, подсказку, которая говорит о том, что файл содержит астрономические, а не земные данные;
- преобразовать небесные координаты в земные.

Режим неба

Пользователь Google Earth может управлять переходом в режим неба с помощью пункта меню **Вид > Переход к просмотру неба** или графической кнопки **Переключение между Землей, небом и планетами**. При переходе в режим неба Google Earth начинает показывать изображения неба, сфотографированные различными наземными и космическими телескопами. Небо отображается с точки зрения наблюдателя, находящегося в центре земной сферы. Такая модель позволяет пользователю рассматривать небо над головой, а также часть небесной сферы, которая обычно видна только с другой стороны Земли.

Координаты

Небесные координаты называются прямым восхождением (ПВ) и склонением (СКЛ). Прямое восхождение соответствует долготе и равно расстоянию от точки, где Солнце пересекает небесный экватор в точке весеннего равноденствия. Прямое восхождение измеряется в часах от 0 до 24, причем один час ПВ равен углу, на который поворачивается небо над данной точкой земной поверхности за один час. ПВ равно 0 в том месте на небе, где Солнце пересекает небесный экватор в весеннее равноденствие, и увеличивается к востоку от нее.

Склонение – это аналог широты, 0 градусов соответствует точкам на небесном экваторе. Значения изменяются в диапазоне от -90^0 (точка, расположенная прямо над Южным полюсом) до $+90^0$ (точка, расположенная прямо над Северным полюсом).

Идея системы координат ПВ/СКЛ состоит в том, что координаты напрямую отображаются на видимое движение звезд вследствие вращения Земли. Если вы находитесь на Земле на широте 20.0, то точка зенита в небе над вами тоже будет иметь склонение, равно 20.0. Точка зенита перемещается по небу, не меняя склонения, пока не окажется в той же точке неба в конце сидерического дня.

На рис. В.1 показана карта Google Sky с линиями постоянства прямого восхождения и склонения.

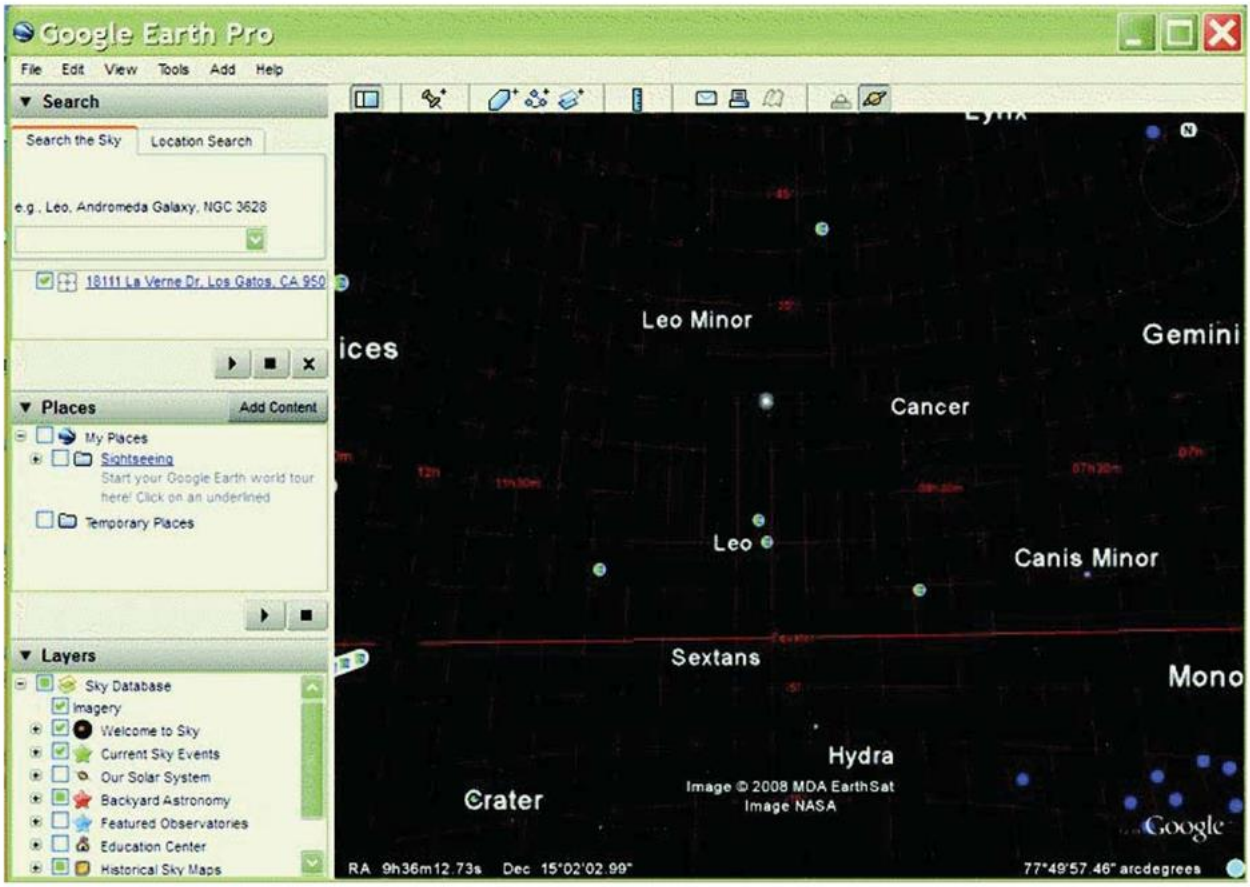


Рис. В.1. В режиме неба прямое восхождение соответствует долготе, а склонение – широте

Поддерживаемые элементы

Посмотрите в документации по своему геобраузеру, какие элементы KML он поддерживает для астрономических данных. Так, Google Earth, Release 4.3 в режиме неба поддерживает следующие элементы:

- Placemark;
- GroundOverlay;
- LineString;
- Polygon;
- MultiGeometry;
- LinearRing;

- Point;
- стилистические элементы;
- контейнерные элементы.

Отметим, однако, что элементы <tilt> и <roll> в текущей версии игнорируются.

Атрибут hint

Если KML-файл содержит астрономические данные, не забудьте добавить в элемент <kml> атрибут `hint`:

```
<kml smlns="http://www.opengis.net/kml/2.2" hint="target=sky">
```

При загрузке файла с подсказкой «target=sky» Google Earth спрашивает пользователя, хочет ли он перейти в режим неба (если в данный момент режим иной).

Преобразование небесных координат для отображения в Google Earth

Для правильного отображения в Google Earth в режиме неба прямое восхождение (часы/минуты/секунды) необходимо преобразовать в градусы долготы. Это делается с помощью несложных вычислений.

Преобразование прямого восхождения

Для преобразования прямого восхождения из диапазона от 0 до 24 в градусы в диапазоне от –180 до +180 применяется следующая формула, в которой hour, minute и second – час, минута и секунда прямого восхождения:

```
(hour + minute/60 + second/3600)*15 - 180
```

Преобразование склонения

Склонение в точности совпадает с долготой и изменяется от –90 градусов к югу от небесного экватора до + 90 градусов к северу от него.

Вычисление range для элемента LookAt

При использовании элемента <LookAt> для астрономических данных необходимо выполнить следующие вычисления для нахождения значения элемента `range`. Основная формула выглядит так:

```
r = R*(k*sin(β/2) - cos(β/2) + 1)
```

где

r – значение range, заданное в элементе <LookAt>;

R – радиус небесной сферы (или, в данном случае, Земли, поскольку мы смотрим на небо, находясь в ее центре);

k равно 1/tg(α/2), или 1.1917536;

α – угловой размер вида в Google Earth, когда камера находится в центре небесной сферы (Земли);
β – требуемый размер изображения неба в дуговых секундах.

Вот несколько примеров значений range:

- большая спиральная галактика (Галактика Подсолнух): 20–30 км;
- большое шаровое скопление (M15): 20–30 км;
- туманность Андромеды: 200 км;
- планетарная туманность (туманность Сова): 5–10 км;
- большая туманность (Трехраздельная туманность): 10–30 км;
- одна хаббл-метка (секстет Сейферта): 2–5 км;
- открытое звездное скопление (скопление Улей): 30–60 км;
- небольшая спиральная галактика: 5–10 км;
- Большое Магелланово Облако: 400–500 км.

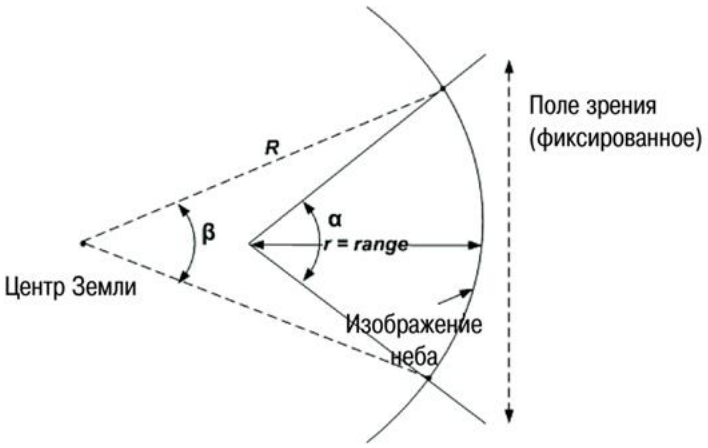


Рис. В.2. Когда в KML представляются астрономические данные, поле зрения фиксировано, так как виртуальная точка обзора помещена в центр земной сферы. Поэтому мы изнутри рассматриваем поверхность Земли, на которую спроецированы небесные данные

Пример метки

В файле CrabNebula.kml содержится описание туманности Краба для Google Earth.

CrabNebula.kml

```
<?xml version="1.0" encoding="utf-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2" hint="target=sky">
<Document>
  <Style id="CrabNebula">
    <BalloonStyle>
      <text><center><b>$[name]</b></center><br/>$[description]</text>
    </BalloonStyle>
  </Style>
  <Placemark>
    <name>Туманность Краба</name>
    <description>
      <![CDATA[
```

```
        Это туманность Краба - остатки взрыва сверхновой, который наблюдался на Земле
        в 1054 году. Дополнительную информацию о туманности Краба вы найдете, заглянув
        в следующие слои:
        <ul>
          <li> <b>Любительская астрономия</b>
          <li> <b>Показ снимков телескопа Хаббла</b>
          <li> <b>Жизнь звезды</b>
        </ul>
        Исследуйте небо с удовольствием!
      ]]>
    </description>
    <LookAt>
      <longitude>-96.366783</longitude>
      <latitude>22.014467</latitude>
      <altitude>0</altitude>
      <range>10000</range>
      <tilt>0</tilt>
      <heading>0</heading>
    </LookAt>
    <styleUrl>#CrabNebula</styleUrl>
    <Point>
      <coordinates>-96.366783,22.014467,0</coordinates>
    </Point>
  </Placemark>
</Document>
</kml>
```

На рис. B.3 показано, как этот файл выглядит в Google Earth.

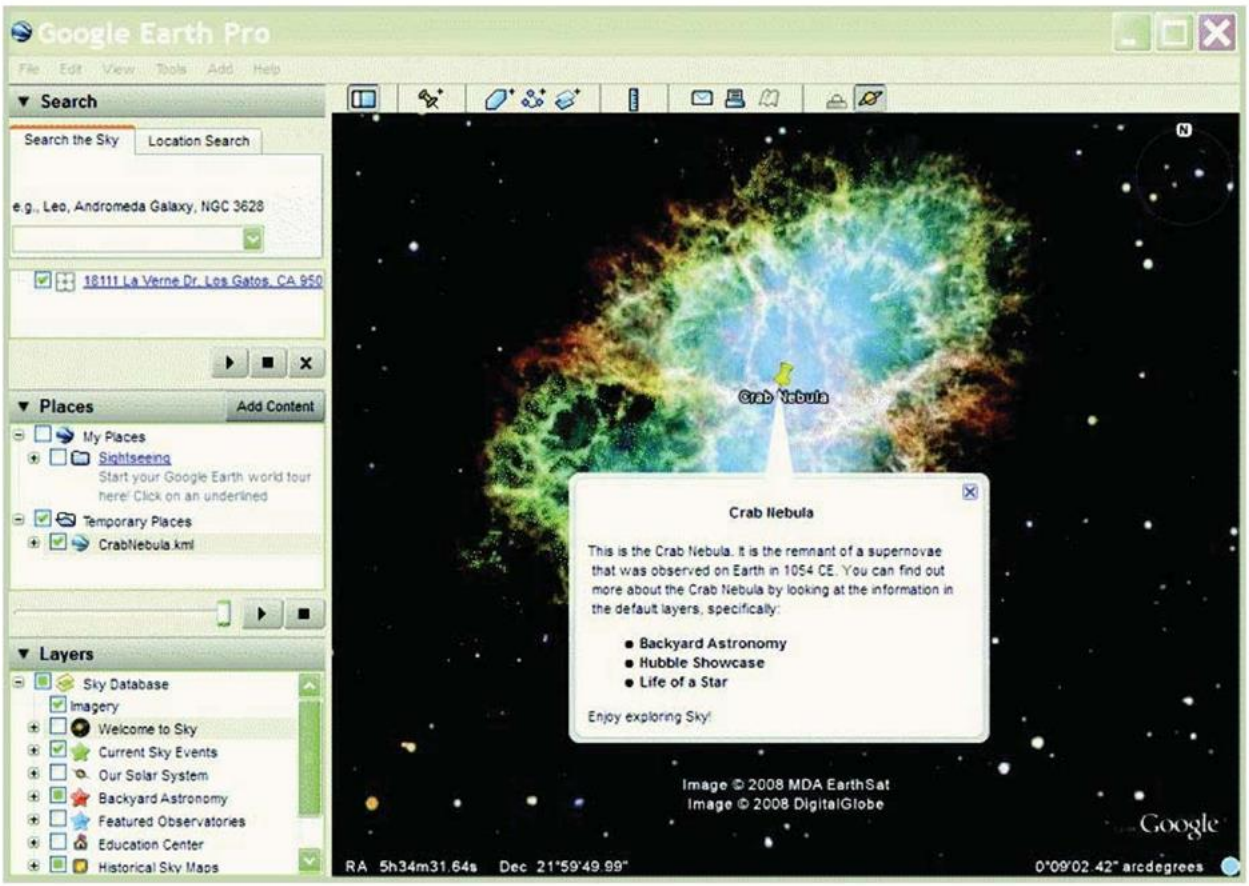


Рис. B.3. Для показа небесных данных в Google Earth необходимо выполнить простой пересчет параметров элемента <LookAt>. Информационные окна создаются в режиме неба точно так же, как в «режиме Земли»

Наложение на небо в KML

Существуют два основных способа создать изображение, накладываемое на небесные данные в Google Earth:

- воспользовавшись встроенным пользовательским интерфейсом, совместить накладываемое изображение с графической основой, предоставляемой Google Earth. Этот метод хорошо работает для относительно небольших участков основы. Не забудьте сохранить созданный в Google Earth файл, не выходя из режима неба;
- воспользоваться специальными утилитами, например пакетом **wcs2kml** с открытыми исходными текстами, который переводит данные из проекции WCS (World Coordinate System – внешняя система координат) в KML. Этот метод обычно применяется для больших изображений и оказывается

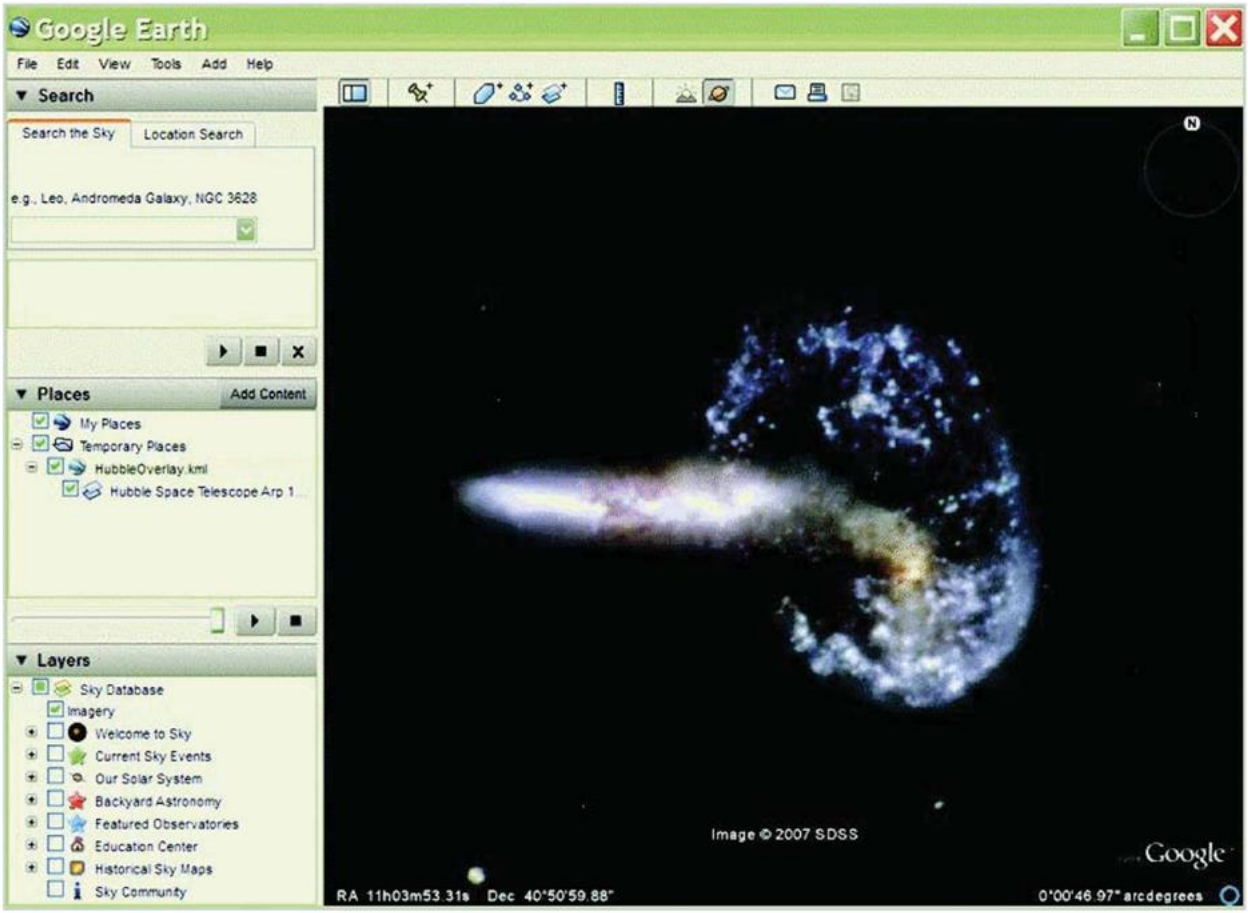


Рис. В.4. Накладываемое изображение объекта Arp 148 – результата столкновения двух галактик, удаленных от Земли примерно на 500 миллионов световых лет, www.spacetelescope.org/images/archive/top100/. Фотография сделана НАСА, ESA, группа наследия Хаббла (STScI/AURA)-ESA/Hubble Coolaboration и А. Эвансом (Университ штата Вирджиния, Шарлотсвилль/NRAO/Университет Стоуни Брук), К. Нолл (STScI) и Дж. Уэстфал (Калифорнийский технологический институт). Публикуется с разрешения

проще, чем выполнение математических вычислений для преобразования координат. В исходном файле координаты записаны в формате FITS (Flexible Image Transport System – гибкая система передачи изображений), часто применяемом в астрономии.

В следующем примере накладываемое изображение создается из фотографии, сделанной телескопом Хаббла.

HubbleOverlay.kml

```
<?xml version="1.0" encoding="utf-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2" hint="target=sky">
<Document>
  <GroundOverlay>
    <name>Телескоп Хаббла, наложение Arp 148</name>
    <LookAt>
      <longitude>-14.0270365795</longitude>
      <latitude>40.8499721419</latitude>
      <heading>0.00</heading>
      <range>2240.10965285</range>
      <altitudeMode>absolute</altitudeMode>
    </LookAt>
    <LatLonBox>
      <north>40.85706883329747</north>
      <south>40.84297545048161</south>
      <west>-14.03641921662847</west>
      <east>-14.01765394233431</east>
      <rotation>0</rotation>
    </LatLonBox>
    <Icon>
      <href>warped_image.png</href>
    </Icon>
  </GroundOverlay>
<Document>
<kml>
```

Алфавитный указатель

;balloon, анкер, 49
;balloonFlyto, анкер, 49
[cameraAlt], 153
[cameraLat], 153
[cameraLon], 153
[clientName], 154
[clientVersion], 153
[horizFov], 153
[horizPixels], 153
[kmlVersion], 154
[language], 154
[lookatHeading], 153
[lookatRange], 153
[lookatTerrainAlt], 153
[lookatTilt], 153
[lookLat], 153
[lookLon], 153
[terrainEnabled], 153
[vertFov], 153
[vertPixels], 153
<address>, 238
<Alias>, 74, 255
<altitude>, 45, 57, 114, 120, 232, 242, 253, 254
<altitudeMode>, 45, 55, 57, 115, 120, 197, 233, 242, 246, 247, 253, 254, 265, 266, 268
<atom:author>, 238
<atom:link href="">, 238
<BalloonStyle>, 31, 77, 90, 231
  использование в качестве шаблона, 218
<begin>, 179, 273
<bgColor>, 90, 94, 231, 252
<bottomFov>, 128, 262
<Camera>, 113, 232
  отличия от <LookAt>, 113
  синтаксис, 114
<Change>, 166, 169, 258
<color>, 65, 83, 88, 112, 233, 260
<colorMode>, 65, 80, 84, 88, 233
<cookie>, 158, 257
<coordinates>, 27, 29, 56, 64, 246, 247, 263, 265
  порядок задания в KML, 56
<Create>, 166, 167, 258
<Data name= >, 217, 236
<Data>, 215
  синтаксис, 217
<Delete>, 167, 170, 259
<description>, 28, 29, 63, 238
  переопределение, 159
<displayMode>, 91, 231
<displayName>, 217, 221, 224, 236, 269
<Document>, 96, 221, 235
<drawOrder>, 112, 260
<east>, 119, 197, 242, 268
<end>, 179, 273
<expires>, 159, 258
<ExtendedData>, 216, 222, 235, 240
<extrude>, 55, 246, 247, 265, 266
<fill>, 66, 267
<flyToView>, 144, 256
<Folder>, 47, 240
<gridOrigin>, 131, 263
<GroundOverlay>, 241
  в сочетании с <Region>, 198
  пример, 120
  синтаксис, 118
<heading>, 45, 73, 84, 114, 232, 244, 253, 255
<hotSpot>, 85, 244
<href>, 43, 84, 94, 113, 249, 252
<httpQuery>, 137, 146, 152, 153, 251
<Icon>, 83, 84, 113, 242, 243, 260
<IconStyle>, 42, 77, 83, 243
<ImagePyramid>, 262
  дополнение изображения, 131
  пример, 131
  принцип работы, 130
  синтаксис, 130
  создание пирамиды, 131
<innerBoundaryIs>, 266
<ItemIcon>, 94, 252
<key>, 272
<kml>, 245
<LabelStyle>, 77, 88, 245
<latitude>, 45, 114, 232, 253, 254
<LatLonAltBox>, 192, 197, 198, 267
<LatLonBox>, 119, 120, 242
<leftFov>, 128, 262
<LinearRing>, 55, 64, 246
<LineString>, 32, 55, 58, 246
<LineStyle>, 77, 93, 251
  в сочетании с экранным наложением, 126
<Link>, 144, 248, 257
<linkDescription>, 159, 258

<linkName>, 159, 258
<linkSnippet maxLines= >, 159, 258
<listItemType>, 93, 251
<Location>, 72, 254
<Lod>, 192, 195, 197, 268
  описание, 195
<longitude>, 45, 114, 232, 253, 254
<LookAt>, 44, 253
  и астрономические данные, 277
  отличия от <Camera>, 113
  поиск и исправление ошибок, 46
<maxAltitude>, 192, 197, 268
<maxFadeExtent>, 195, 198, 269
<maxHeight>, 131, 263
<maxLodPixels>, 192, 198, 269
<maxSessionLength>, 158, 257
<maxWidth>, 131, 263
<message>, 158, 257
<minAltitude>, 192, 197, 268
<minFadeExtent>, 195, 198, 269
<minLodPixels>, 197, 268
<minRefreshPeriod>, 158, 257
<Model>, 32, 55, 71, 254
  вращение, 73
<MultiGeometry>, 33, 55, 68, 101, 256
<name>, 28, 237
  переопределение, 159
<namespace_prefix:other>, 236
<near>, 128, 262
<NetworkLink>, 136, 256
  и регионы, 139, 207
  локальные и удаленные, 138
  назначение, 138
  обновление, 136, 139, 147
  определение, 137
  потомки, 136, 138
  пример, 146
  синтаксис, 143
<NetworkLinkControl>, 137, 257
  описание, 156
  синтаксис, 157
  функция обновления, 139, 164
<north>, 119, 197, 242, 268
<open>, 51, 238
<Orientation>, 72, 254
<outerBoundaryIs>, 266
<outline>, 66, 267
<overlayXY>, 124, 270
<Pair>, 272
<phoneNumber>, 238
<PhotoOverlay>, 261
  синтаксис, 127
  характеристики, 127
<Placemark>, 28, 31, 264
  геометрические элементы в, 32
  разнесение по регионам, 205
  с потомком <Point>, 32
<Point>, 29, 55, 84, 128, 263, 264
  внутри метки, 32, 101
<Polygon>, 32, 55, 64, 265
<PolyStyle>, 64, 77, 81, 266
<range>, 45, 253
<refreshInterval>, 145, 249
<refreshMode>, 145, 249
<refreshVisibility>, 143, 256
<Region>, 189, 240, 267
  в сочетании с рельефным наложением, 198
  для двумерного наложения на ненулевой высоте, 201
  для трехмерной модели, 200
  и <NetworkLink>, 207
  и режим <viewRefreshMode>, 196
  каскадирование, 196
  надписи, 205
  ограничивающий параллелепипед, 192
  описание, 190
  применение, 189
  развернутый пример, 202
  синтаксис, 196
<ResourceMap>, 72, 74, 255
<rightFov>, 128, 262
<roll>, 73, 114, 232, 255
<rotation>, 119, 125, 129, 242, 262, 271
<rotationXY>, 125, 271
<Scale>, 72, 73, 255
<scale>, 84, 88, 243, 245
<Schema name= >, 221
<Schema>, 215, 221
  синтаксис, 221, 269
<SchemaData>, 215, 221, 236
  синтаксис, 222
<ScreenOverlay>, 109, 270
  позиция, 124
  применение, 122
  примеры, 125
  синтаксис, 123
  элементы-потомки, 124
<screenXY>, 125, 271
<shape>, 127, 263
  поле зрения, 129
<SimpleData name= >, 222, 236
<SimpleField type= >, 221, 269
<size>, 125, 271
<Snippet maxLines= >, 238
<Snippet> и <snippet>, 237
<sourceHref>, 74, 255
<south>, 119, 197, 242, 268
<state>, 94, 252
<Style>, 63, 64, 271
  иерархия наследования, 79
  подстили, 77

синтаксис, 79
<StyleMap>, 100, 272
и точечные метки, 101
применение, 103
пример, 101
синтаксис, 101
<styleUrl>, 98, 101, 239, 272
<targetHref>, 74, 166, 255
<tessellate>, 55, 59, 246, 247, 266
<text>, 90, 219, 231
<textColor>, 90, 231
<tileSize>, 130, 263
<tilt>, 45, 73, 114, 232, 253, 255
<TimeSpan>, 177, 273
и рельефные наложения, 185
синтаксис, 179
<TimeStamp>, 177, 273
анимированные метки, 183
данные от GPS-приемника, 180
и метки, 180
синтаксис, 179
<topFov>, 128, 262
<Update>, 159, 258
виды обновления, 164
изменение элементов, 169
механизм работы, 164
применение сценариев, 172
синтаксис, 166
создание элементов, 166
удаление элементов, 170
<value>, 217, 236
<viewBoundScale>, 137, 153, 250
<viewFormat>, 136, 153, 250
<viewRefreshMode>, 145, 148, 196, 249
<viewRefreshTime>, 145, 148, 250
<ViewVolume>, 128, 262
<visibility>, 50, 237
<west>, 119, 197, 242, 268
<when>, 179, 274
<x>, 255
<xal:AddressDetails>, 238
<y>, 255
<z>, 255
absolute, как значение, 45, 57, 197, 242
AbstractView, 159, 230, 239, 259
AdvancedTemplate.kml, 43
altitudeModeEnum, тип, 230
aMyPlacemark.kml, 167
angle180, тип, 230
angle360, тип, 230
angle90, тип, 230
anglepos180, тип, 230
Apache, сервер, 140
aPlacemark.kml, 169, 170
Atom, формат агрегирования, 75, 238
AugustineWebcam.kml, 147
AugustineWebcamRevised.kml, 160
author, элемент, 75
AvianFluExcerpt.kml, 183
BalloonStyle.kml, 92
BalloonStyleTemplate.kml, 219
BalloonTemplate.kml, 38
BasicNetworkLink.kml, 160
bOnePlacemark.kml, 168
boolean, тип, 229
bOriginalPlacemark.kml, 169
bOriginalPlacemarks.kml, 171
CameraRotations.kml, 117
cChangeMe.kml, 169
cCreateMore.kml, 168
cDeleteOne.kml, 171
checkHideChildren, 93, 252
checkOffOnly, 94, 252
clampToGround, значение, 45, 197, 242
CloudRegionAtAltitude.kmz, 201
COLLADA, формат файла, 70, 254
color, тип, 230
colorModeEnum, тип, 230
ColorStyle, 79, 233
Container, 234
continents.kml, 172
continents.py, 172
CrabNebula.kml, 278
CSS (каскадные таблицы стилей), 77, 105
cylinder, 128, 263
поле зрения для, 129
dae, суффикс COLLADA-файлов, 70
dateTime, тип, 230
Digital Urban, компания, 130
displayModeEnum, тип, 230
doc.kml, из PuffModel.kmz, 151
doc.kml, имя файла по умолчанию, 51
double, тип, 229
dRemoveOnePlacemark.kml, 171
dUpdatePlacemark.kml, 170
dUpdatePlacemarkData.kml, 168
eXtensible Address Language, 238
ExtrudedLineString.kml, 60
Feature, 50, 110
добавление нестандартных данных, 215, 221
обозримость, 189
элементы, специфичные для, 237
float, тип, 229
fraction, единица измерения, 123
Galdos Systems, компания, 28
Geometry, 32, 55, 241
использование совместно с картой стилей, 103
GET, команда, 139
GigaPan, компания, 130
GMT, Гринвичское время, 178

Google Earth, 27
  как веб-клиент, 139
  цветоподборщик, 40
Google Earth Blog, 21
Google Lit Trips, 78, 86
Google Maps, 62
GPS-данные с временными штампами, пример, 180
gridOriginEnum, тип, 230
GroundAndImage.kml, 122
GroupingPlacemarks.kml, 47
HelloEarth.kml, 26
highlight, стиль, 100
hint, атрибут, 277
HistoricOverlay.kmz, 198
HTTP, протокол, 139
HubbleOverlay.kml, 281
IconStyle.kml, 87
id, атрибут, 50
insetPixels, единица измерения, 123
int, тип, 229
itemIconStateEnum, тип, 230
Keyhole, компания, 19
KingTut.kmz, 74
KML (Keyhole Markup Language), 14
  блоги, 21
  будущие версии, 21
  дерево элементов, 227
  динамический, 162
  заголовки, 228
  комментарии, 69
  обновление, 164
  обратная совместимость, 229
  описание, 19
  поддержка, 19
  презентации, 23
  применение, 21
  синтаксические правила, 228
  спецификация, 20
  справочное руководство, 227
  схема, 229
  типы, 229
  утилиты проверки синтаксиса, 28
  файлы, 21
KML-файлы
  автор и источник, 75
  структура, 28
  тип MIME, 141
KMZ-архивы, 51
  описание, 51
  структура, 51
  тип MIME, 141
LabelStyle.kml, 89
lighttpd, веб-сервер, 141
LineStringWithAltitude.kml, 61
LineStyle, 62, 64, 77, 80, 247
LineStyle.kml, 80
link, элемент, 75
listItemTypeEnum, тип, 230
ListStyle.kml, 94
maxLines, атрибут, 159
MIME, типы, 141
Mozilla Firefox, 86
MultiGeometry.kml, 69
My Maps, пример, 20
name, элемент, 75
NameValuePairs.kml, 218
never, значение, 145
normal, стиль, 100
OASIS xAL 2.0, 238
Object, абстрактный базовый класс, 259
ogckml22.xsd, 229
Ogle Earth, блог, 21
onChange, значение, 145, 147
onExpire, значение, 147, 159, 196
onExpiree, значение, 145
onInterval, значение, 145, 147
onRegion, значение, 145
onRequest, значение, 145
onStop, значение, 145
Open Geospatial Consortium (OpenGIS), 19, 229
Overlay, 260
  синтаксис, 111
PaleoGlobeExcerpt.kml, 187
PhotoOverlay.kml, 134
pixels, единица измерения, 123
Pizzas.kml, 154
PolyStyle.kml, 83
PolyWithInnerAndOuterBoundaries.kml, 67
PuffModel.kml, 150
radioFolder, 93, 252
rectangle, 127
  поле зрения для, 129
refreshModeEnum, тип, 230
relativeToGround, значение, 45, 57, 197
RomaniaRegion.kml, 208
SchemaDataAndBalloonStyle.kml, 225
ScreenOverlay.kml, 126
ScreenRulers.kml, 193
shapeEnum, тип, 230
SharedStyles.kml, 97
SimpleLineString.kml, 58
SimplePolygon.kml, 64
SimpleTextBalloon.kml, 36
SimpleTextBalloonWithStyle.kml, 41
SimpleUserData.kml, 216
sphere, 128, 263
  поле зрения для, 129
string, тип, 229
StyleMap.kml, 101
StyleSelector, атрибут, 240

StyleSelector, элемент, 272
styleStateEnum, тип, 230
targetId, 167
TesselatedLineString.kml, 59
TimePrimitive, атрибут, 239
TimePrimitive, элемент, 273
TimeStampGPSExample.kml, 180
UkraineRegion.kml, 207
UnitedNationsModel.kmz, 200
unitsEnum, тип, 230
URI, фрагменты, 163
UTC (всеобщее скоординированное время, 178
vec2, тип, 230
viewRefreshEnum, тип, 230
WCS (внешняя система координат), 280
wcs2kml, 280
Web Map Service, 153
WineRegions.kmz, 203
x, значение, 244
xunits, значение, 244
y, значение, 244
yunits, значение, 244
ZIP-архивы, 51
Абзацы, промежуток между, 35
Абсолютные ссылки на файлы, 142
Абстрактные элементы, 50, 79, 228
Активная точка значка, 85
Анимация, использование временных примитивов, 175
Анимированное рельефное наложение, 186
Анимированные метки, 183
Анкеры, для поведения FlyTo, 49
Архив фотографий Маунтин-Вью, пример, 198, 209
Астрономические координаты
  в KML, 25, 275
  преобразование, 277
Атрибуты, 85
Булевские значения, определение, 81
Внутренние стили, 41, 96
Время
  задание в формате XML, 178
  стандарты, 178
Вулканическая обсерватория Аляски, 22, 23, 137, 146, 148
Выделение многоугольников, 103
Геобраузинг, 12
Гигапиксельные изображения, 127
  задание URL, 134
  пирамиды для, 130
  прозрачность, 134
Гиперссылки, добавление в информационное окно, 35
Гробница Тутанхамона, пример, 70
Группы подстановок, 228
Динамический KML
  идентификаторы, 162
  описание, 162
Долгота, 56, 57
Дрейф континентов, пример, 176
Заголовки, 228
Затухание, 195, 198
  добавление эффекта, 206
Значения по умолчанию, 229
Значки
  нестандартные, 42
  обновление, 113
Зулусское время, 178
Идентификаторы, 162
Институт Джейн Гудолл, пример, 24, 31, 32, 37
Информационные окна, 28
  добавление гиперссылок, 35
  добавление изображений, 36
  добавление текста, 34
  задание цвета, 39
  по умолчанию, 31, 33, 90
  типографическое оформление, 35
Каскадирование регионов, 196
Клиенты, 139
  взаимодействие с сервером, 152
Коллекция карт Дэвида Рамси, 191
Ком-Фирин, пример, 110, 121
Комментарии в KML, 69
Копирование и вставка, 29
Кортеж, 56
Кризис в Дарфуре, пример, 25
Курсив, 35
Лавоопасные зоны, пример, 64
Линейные стили, 55
Лондонский глаз, пример, 177
Маршруты, исключение, 42
Метки, 26, 51
  анимация, 180
  перелет к, 49
  потомки, 50
  синтаксис, 50
Многоугольники, 63
  внешняя граница, 63
  внутренняя граница, 67
  выделение, 103
  дырки, 67
  на ненулевой высоте, 120
  простые, 63
Набор данных, упрощение, 202
Накладываемые изображения, 118
  для неба, 280
Наложения, 108
  виды, 107
  обновление, 113
  общие свойства, 111
  создание, 110
Небесные координаты
  в KML, 25, 275

преобразование, 277
Нестандартные данные
добавление, 215, 221
типизированные, 222
Нестандартные значки, 42
Нумерация плиток, 133
Обновление, 136, 147
по событиям просмотра, 154
Обновление по событиям просмотра, 139, 148
Объектная модель, 157
Ограничивающий параллелепипед, 192
Ограничивающий прямоугольник, 153
Оклахома, пример, 82
Относительные ссылки на файлы, 142
Периодическое обновление, 139, 145, 148
Плитки, нумерация, 133
Поворот
камеры, 115
модели, 73
Подстановка компонентов, 218
с применением <Schema> и <SchemaData>, 224
Подстили, 77
Поле зрения, 128
поворот, 129
Полигональные стили, 55, 65
Полужирный шрифт, 35
Порядок следования, важность, 29, 228
Потомки (элементы), 29, 228
Прозрачность
цвета, 39
шестнадцатеричная нотация, 112
Произвольные XML-данные, 215, 216
Простые элементы, 29, 228
Прямое восхождение (ПВ), 275
преобразование, 277
Пути, 58
Разделяемые стили, 41, 94, 96, 100, 218
Разрыв строки, 35
Расширенные данные, 189
различные подходы к выбору, 215
Регистр букв в именах, 29, 228
Режим контроля, 228
Режим неба, 25, 275
использование <LookAt>, 277
координаты, 277
наложения, 280
поддерживаемые элементы, 276
пример, 278
Рельефные наложения, 108
в сочетании с <TimeSpan>, 185
Родительские элементы, 29, 228
Сан Франциско, виндсерфинг, пример, 23
Серверы, 139
взаимодействие с клиентом, 152
конфигурирование, 141
переопределение поведения, 159
Сеть
веб-серверы, 140
клиент-серверная архитектура, 139
отправка информации по, 152
тестирование ссылок, 141
Символьные данные, 29
Склонение (СКЛ), 275
преобразование, 277
Снимок метки, 45
Составные элементы, 28, 228
Ссылки на файлы, 142
Стили
геометрических форм, 55
определение во внешнем файле, 105
переопределение, 98
разделяемые, 77, 96
Строка запроса, 152
Структурированные адреса, 238
Суперналожения, 191, 196
подготовка KML-файлов, 212
подготовка данных, 209
применение, 207
пример, 209
Текст
добавление в информационное окно, 34
цвет, 41
Текстуры, 74
Тектонические плиты, пример, 59
Текущий вид, отправка информации о, 152
Тесселяция, 59
Типы KML, 229
Толщина линии, 63
Точка обзора, задание, 44
Трехмерная модель, 200
Уровень детализации (LOD), 192
Фотоналожения, 109
создание, 130
Фрагменты URI, 163
Цвет
в KML и HTML, 39
линии, 63
представление, 39
случайный, 80
текста, изменение, 41
фона информационного окна, 38
цветоподборщик в Google Earth, 40
Швейцарская система общественного транспорта, пример, 126, 155, 190, 214
Шестнадцатеричная нотация, 39, 112
Широта, 56, 57
Экструзия, 60
Элементы
значения, 29
типы, 28, 228

Книги издательства «ДМК Пресс» можно заказать в торгово-издательском холдинге «АЛЬЯНС-КНИГА» наложенным платежом, выслав открытку или письмо по почтовому адресу: **123242, Москва, а/я 20** или по электронному адресу: **orders@alians-kniga.ru**.

При оформлении заказа следует указать адрес (полностью), по которому должны быть высланы книги; фамилию, имя и отчество получателя. Желательно также указать свой телефон и электронный адрес.

Эти книги вы можете заказать и в Internet-магазине: **www.alians-kniga.ru**.

Оптовые закупки: тел. **(495) 258-91-94, 258-91-95**; электронный адрес **books@alians-kniga.ru**.

Джози Вернеке

Язык географической разметки KML

Главный редактор *Мовчан Д. А.*
dm@dmk-press.ru
Корректор *Синяева Г. И.*
Верстка *Чаннова А. А.*
Дизайн обложки *Мовчан А. Г.*

Подписано в печать 28.12.2009. Формат 70×100 $^{1}/_{16}$.
Гарнитура «Петербург». Печать офсетная.
Усл. печ. л. 27. Тираж 500 экз.
№

Web-сайт издательства: www.dmk-press.ru

"Способ изложения информации понравится учителям, любителям, веб-дизайнерам — всем, кто стремится украсить свои материалы за счет использования карт."
Уоррен Келли, пастор

"Эта книга могла бы стать стандартным учебником по рассматриваемому предмету, а также классическим справочным руководством."
Томас Дафф, ведущий разработчик

"Эта книга написана так хорошо и читается так легко, что работать с ней — одно удовольствие."
Дэниэл Маккиннон, программист

Язык KML возник как формат представления данных для программы Google Earth, но в процессе эволюции превратился в полноценный международный стандарт описания любого географического содержимого – «HTML для географии». Он уже поддерживается самыми разнообразными приложениями: от Microsoft Virtual Earth и NASA WorldWind до Photoshop и AutoCAD. KML позволяет делать фантастические вещи, а эта книга научит вас пользоваться им на практических примерах, работающих в лучших современных приложениях для онлайновой картографии.

Основываясь на опыте работы с создателями KML, Вернеке обучает читателя технике, полезной всем: от программистов до агентов по продаже недвижимости, ученых, студентов, архитекторов и т.д.

Рассматриваются следующие темы:

- Включение обогащенного содержимого во всплывающие окна, ассоциированные с метками.
- Создание изображений, накладываемых на основу, предоставляемую стандартными геобраузерами.
- Генерирование анимация, составленных из меток, наложений и моделей.
- Обновление содержимого карты через Интернет.
- Управление большими наборами данных с помощью регионов и нестандартных типов.
- Полный справочник по языку KML: элементы, типы, синтаксис, структура файлов и принятые соглашения.

Джози Вернеке, старший технический писатель в компании Google. Работала с создателями первых версий KML. Она автор книг «The Inventor Mentor» и «The Inventor Toolmaker» и соавтор учебника «The VRML 2.0 Handbook» (вместе с Джедом Хартманом).

Internet-магазин: *www.alians-kniga.ru*
Книга - почтой:
Россия, 123242, Москва, а/я 20
e-mail: orders@alians-kniga.ru
Оптовая продажа: *"Альянс-книга"*
(495)258-9194, 258-9195
e-mail: books@alians-kniga.ru

ДМК Пресс издательство

Addison-Wesley
Pearson Education

978-5-94074-554-9
9 785940 745549