



Положительные отзывы к книге «WebGL: руководство по программированию»

«WebGL – одна из заключительных особенностей, приближающих веб-приложения к обычным, настольным приложениям, и книга «WebGL: программирование трехмерной графики» рассказывает о создании таких веб-приложений. Она охватывает все аспекты, связанные с использованием WebGL – JavaScript, OpenGL ES и базовые приемы программирования графики – помогая получить полное представление обо всем, что вам может потребоваться на начальном этапе. За веб-приложениями – будущее, и эта книга поможет вам попасть туда!»

Дэйв Шрайнер (Dave Shreiner), соавтор «The OpenGL Programming Guide, Eighth Edition»¹; редактор серии «OpenGL Library» (Addison Wesley).

«Стандарт HTML5 превратил Web в высокоэффективную платформу для создания весьма привлекательных интерактивных приложений, способных выполняться в самых разных системах. WebGL является одним из важнейших элементов HTML5, позволяющим веб-программистам использовать всю широту возможностей современных графических ускорителей. Поддержка WebGL разрабатывалась так, чтобы обеспечить максимальную надежность в любых веб-совместимых системах и послужить толчком для появления новых инноваций в разработке трехмерного веб-контента, приложений и пользовательских интерфейсов. Эта книга поможет веб-разработчикам понять, какие удивительные возможности несет в себе новая волна веб-функциональности и использовать их на практике.»

Нейл Треветт (Neil Trevett), вице-президент Mobile Content, NVIDIA; президент The Khronos Group

«Давая ясное описание, сопровождаемое отменными иллюстрациями, эта книга прекрасно справляется с задачей превращения такой сложной темы в простое и практичное руководство. Несмотря на всю сложность WebGL, данная книга яв-

¹ На русский язык было переведено только 4-е издание книги: Д. Шрайнер, М. Ву, Дж. Нейдер, Т. Девис «OpenGL. Руководство по программированию. Библиотека программиста. 4-е изд.», ISBN: 5-94723-827-6, 2006, Питер. – *Прим. перев.*

ляется доступным источником знаний даже для начинающих разработчиков, которым определенно стоит выбрать ее, прежде чем переходить к чему-то другому.»

Эван Барчард (Evan Burchard), автор «Web Game Developer's Cookbook» (Addison Wesley)

«Оба автора имеют большой опыт использования OpenGL. Они сумели распространить его на WebGL, в результате чего получилось отличное руководство, которое может пригодиться не только начинающим, но и опытным специалистам.»

Дэниел Хаэн (Daniel Haehn), программист, детская больница в г. Бостон

«WebGL: руководство по программированию просто и понятно излагает механику создания веб-приложений, воспроизводящих трехмерный графику, без использования массивных библиотек или фреймворков. Отличный источник информации для разработчиков, ищущих вводные руководства по интеграции концепций отображения трехмерный графики с ультрасовременными веб-технологиями.»

Брендон Джонс (Brandon Jones), программист, Google

«Великолепная работа блестящих исследователей. Коичи Мацуда (Koichi Matsuda) уверенно ведет новичков к овладению WebGL, и благодаря его усилиям любой сможет начать использовать эту восхитительную веб-технологию, несмотря на ее сложность. Он также включил в книгу описание основных понятий трехмерный графики, заложив фундамент для дальнейшего изучения этой темы. Эта книга станет отличным приобретением для любого веб-дизайнера.»

Крис Маррин (Chris Marrin), редактор спецификации WebGL Спец

«WebGL: руководство по программированию дает отличную возможность пройти путь от новичка до эксперта WebGL. Несмотря на простоту основных понятий, заложенных в WebGL, для работы с этой технологией требуется знание сложного математического аппарата трехмерный моделирования. Книга «WebGL: программирование трехмерной графики» поможет приобрести эти знания и научит применять их на практике. Если вы, в конечном счете, выберете другую библиотеку для работы с трехмерной графикой, отличную от WebGL, знания, полученные при прочтении этой книги, помогут вам быстрее понять, как она действует, и как использовать ее в своих приложениях для решения конкретных задач. И, даже если вы желаете создавать обычные приложения, использующие OpenGL и/или DirectX, «WebGL: программирование трехмерной графики» послужит вам хорошей первой ступенью, потому что большинство книг, посвященных теме программирования трехмерный графики, во многом отстали от современного уровня развития трехмерный технологий. «WebGL: программирование трехмерной графики» поможет вам заложить основы программирования современной трехмерный графики.»

Грегг Таварес (Gregg Tavares), разработчик реализации WebGL в Chrome

*Мудрость приходит с годами, прошлое не возвращается
и время как спираль из полудрагоценных камней...*

— Коичи Мацуда

Моей супруге, семье, друзьям, несущим радость в мою жизнь.

— Роджер Ли



ОГЛАВЛЕНИЕ

Положительные отзывы к книге «WebGL: программирование трехмерной графики»	5
Предисловие	17
Кому адресована эта книга.....	17
О чем рассказывается в этой книге	18
Структура книги	18
Браузеры с поддержкой WebGL.....	22
Примеры программ и сопутствующие ссылки	23
Типографские соглашения	23
Благодарности	23
Об авторах	24
Глава 1.	
Обзор WebGL	26
Достоинства WebGL	27
Вы можете заниматься разработкой приложений с трехмерной графикой, используя только текстовый редактор	28
Публикация приложений с трехмерной графикой не вызывает сложностей .	29
Вы можете использовать всю широту возможностей браузеров	29
Изучение и использование WebGL не вызывает никаких сложностей.....	29
История происхождения WebGL	30
Структура приложений на основе WebGL.....	31
В заключение	32
Глава 2.	
Первые шаги в WebGL	33
Что такое canvas?	33
Использование тега <canvas>	34
DrawRectangle.js.....	36
Самая короткая WebGL-программа: очистка области рисования	40
Файл HTML (HelloCanvas.html)	40
Программа на JavaScript (HelloCanvas.js)	41
Эксперименты с примером программы.....	46
Рисование точки (версия 1)	46
HelloPoint1.html	48
HelloPoint1.js	48
Что такое шейдер?	49
Структура WebGL-программы, использующей шейдеры	51

Инициализация шейдеров	53
Вершинный шейдер	55
Фрагментный шейдер	58
Операция рисования	58
Система координат WebGL	60
Эксперименты с примером программы	61
Рисование точки (версия 2)	62
Использование переменных-атрибутов	63
Пример программы (HelloPoint2.js)	64
Получение ссылки на переменную-атрибут	65
Присваивание значения переменной-атрибуту	66
Семейство методов <code>gl.vertexAttrib3f()</code>	68
Эксперименты с примером программы	70
Рисование точки по щелчку мышью	71
Пример программы (ClickedPoints.js)	72
Регистрация обработчиков событий	73
Обработка события щелчка мышью	75
Эксперименты с примером программы	78
Изменение цвета точки	79
Пример программы (ColoredPoints.js)	80
Uniform-переменные	82
Получение ссылки на uniform-переменную	83
Присваивание значения uniform-переменной	84
Семейство методов <code>gl.uniform4f()</code>	86
В заключение	86

Глава 3.

Рисование и преобразование треугольников	88
Рисование множества точек	88
Пример программы (MultiPoint.js)	90
Использование буферных объектов	93
Создание буферного объекта (<code>gl.createBuffer()</code>)	94
Указание типа буферного объекта (<code>gl.bindBuffer()</code>)	95
Запись данных в буферный объект (<code>gl.bufferData()</code>)	96
Типизированные массивы	98
Сохранение ссылки на буферный объект в переменной-атрибуте (<code>gl.vertexAttribPointer()</code>)	99
Разрешение присваивания переменной-атрибуту (<code>gl.enableVertexAttribArray()</code>)	101
Второй и третий параметры метода <code>gl.drawArrays()</code>	102
Эксперименты с примером программы	103
Привет, треугольник	104
Пример программы (HelloTriangle.js)	105
Простые фигуры	106
Эксперименты с примером программы	108
Привет, прямоугольник (HelloQuad)	109
Эксперименты с примером программы	110
Перемещение, вращение и масштабирование	111
Перемещение	112

Пример программы (TranslatedTriangle.js)	113
Вращение	115
Пример программы (RotatedTriangle.js)	117
Матрица преобразования: вращение	121
Матрица преобразования: перемещение	123
И снова матрица вращения	124
Пример программы (RotatedTriangle_Matrix.js)	125
Применение того же подхода для перемещения	128
Матрица преобразования: масштабирование	129
В заключение	130
Глава 4.	
Дополнительные преобразования и простая анимация	131
Перемещение с последующим вращением	131
Библиотека матричных преобразований: cuon-matrix.js	132
Пример программы (RotatedTriangle_Matrix4.js)	133
Объединение нескольких преобразований	135
Пример программы (RotatedTranslatedTriangle.js)	137
Эксперименты с примером программы	139
Анимация	140
Основы анимации	141
Пример программы (RotatingTriangle.js)	141
Повторяющиеся вызовы функции рисования (tick())	144
Рисование треугольника после поворота на указанный угол (draw())	145
Запрос на повторный вызов (requestAnimationFrame())	146
Изменение угла поворота (animate())	148
Эксперименты с примером программы	150
В заключение	151
Глава 5.	
Цвет и текстура	152
Передача другой информации в вершинные шейдеры	152
Пример программы (MultiAttributeSize.js)	153
Создание нескольких буферных объектов	155
Параметры stride и offset метода gl.vertexAttribPointer()	156
Пример программы (MultiAttributeSize_Interleaved.js)	157
Изменение цвета (varying-переменные)	160
Пример программы (MultiAttributeColor.js)	161
Эксперименты с примером программы	164
Цветной треугольник (ColoredTriangle.js)	165
Сборка и растеризация геометрических фигур	165
Вызовы фрагментного шейдера	169
Эксперименты с примером программы	170
Принцип действия varying-переменных и процесс интерполяции	171
Наложение изображения на прямоугольник	174
Координаты текстуры	176
Пример программы (TexturedQuad.js)	176
Использование координат текстуры (initVertexBuffers())	179
Подготовка и загрузка изображений (initTextures())	179

Подготовка загруженной текстуры к использованию в WebGL (loadTexture())	183
Поворот оси Y изображения	183
Выбор текстурного слота (gl.activeTexture())	184
Указание типа объекта текстуры (gl.bindTexture())	185
Настройка параметров объекта текстуры (gl.texParameteri())	187
Присваивание изображения объекту текстуры (gl.texImage2D())	190
Передача текстурного слота фрагментному шейдеру (gl.uniform1i())	192
Передача координат текстуры из вершинного шейдера во фрагментный шейдер	193
Извлечение цвета текселя во фрагментном шейдере (texture2D())	193
Эксперименты с примером программы	195
Наложение нескольких текстур на фигуру	196
Пример программы (MultiTexture.js)	197
В заключение	201

Глава 6.

Язык шейдеров OpenGL ES (GLSL ES)	203
Краткое повторение основ шейдеров	203
Обзор GLSL ES	204
Привет, шейдер!	205
Основы	205
Порядок выполнения	205
Комментарии	205
Данные (числовые и логические значения)	206
Переменные	206
GLSL ES – типизированный язык	207
Простые типы	207
Присваивание и преобразования типов	208
Операции	209
Векторы и матрицы	210
Присваивание и конструирование	211
Доступ к компонентам	213
Операции	216
Структуры	218
Присваивание и конструирование	219
Доступ к членам	219
Операции	219
Массивы	220
Семплы	221
Приоритеты операторов	221
Условные операторы и циклы	222
Инструкции if и if-else	222
Инструкция for	223
Инструкции continue, break, discard	223
Функции	224
Объявления прототипов	225
Квалификаторы параметров	226
Встроенные функции	227

Глобальные и локальные переменные	227
Квалификаторы класса хранения	228
Квалификатор const	228
uniform-переменные	230
varying-переменные	230
Квалификаторы точности	230
Директивы препроцессора.....	233
В заключение	235

Глава 7.

Вперед, в трехмерный мир	236
Что хорошо для треугольников, хорошо и для кубов	236
Определение направления взгляда	237
Точка наблюдения, точка направления взгляда и направление вверх.....	238
Пример программы (LookAtTriangles.js)	240
Сравнение LookAtTriangles.js с RotatedTriangle_Matrix4.js	243
Взгляд на повернутый треугольник с указанной позиции	245
Пример программы (LookAtRotatedTriangles.js)	246
Эксперименты с примером программы.....	247
Изменение точки наблюдения с клавиатуры.....	249
Пример программы (LookAtTrianglesWithKeys.js)	249
Недостающие части	252
Определение видимого объема в форме прямоугольного параллелепипеда ...	252
Определение видимого объема	253
Определение границ видимого объема в форме параллелепипеда.....	254
Пример программы (OrthoView.html)	256
Пример программы (OrthoView.js)	257
Изменение содержимого HTML-элемента из JavaScript	258
Вершинный шейдер	259
Изменение near или far	260
Восстановление отсеченных частей треугольников (LookAtTrianglesWithKeys_ViewVolume.js)	262
Эксперименты с примером программы.....	264
Определение видимого объема в форме четырехгранной пирамиды	265
Определение границ видимого объема в форме четырехгранной пирамиды	267
Пример программы (PerspectiveView.js)	269
Назначение матрицы проекции	271
Использование всех матриц (модели, вида и проекции).....	272
Пример программы (PerspectiveView_mvp.js)	274
Эксперименты с примером программы	276
Правильная обработка объектов переднего и заднего плана.....	277
Удаление скрытых поверхностей.....	280
Пример программы (DepthBuffer.js).....	282
Z-конфликт.....	283
Привет, куб	285
Рисование объектов с использованием индексов и координат вершин	287
Пример программы (HelloCube.js)	288

Запись координат вершин, цветов и индексов в буферный объект	291
Добавление цвета для каждой грани куба	293
Пример программы (ColoredCube.js)	295
Эксперименты с примером программы	296
В заключение	297
Глава 8.	
Освещение объектов	299
Освещение трехмерных объектов	299
Типы источников света	300
Типы отраженного света	302
Затенение при направленном освещении в модели диффузного отражения	304
Использование направления света и ориентации поверхности в модели диффузного отражения	305
Ориентация поверхности: что такое нормаль?	307
Пример программы (LightedCube.js)	309
Добавление затенения, обусловленного фоновым освещением	315
Пример программы (LightedCube_ambient.js)	316
Освещенность перемещаемого и вращаемого объекта	317
Волшебство матриц: транспонированная обратная матрица	319
Пример программы (LightedTranslatedRotatedCube.js)	320
Освещение точечным источником света	322
Пример программы (PointLightedCube.js)	323
Более реалистичное затенение: вычисление цвета для каждого фрагмента	326
Пример программы (PointLightedCube_perFragment.js)	327
В заключение	328
Глава 9.	
Иерархические объекты	329
Рисование составных объектов и управление ими	329
Иерархическая структура	331
Модель с единственным сочленением	332
Пример программы (JointModel.js)	333
Рисование иерархической структуры (draw())	337
Модель со множеством сочленений	339
Пример программы (MultiJointModel.js)	340
Рисование сегментов (drawBox())	343
Рисование сегментов (drawSegment())	345
Шейдер и объект программы: роль initShaders()	349
Создание объектов шейдеров (gl.createShader())	350
Сохранение исходного кода шейдеров в объектах шейдеров (gl.shaderSource())	351
Компиляция объектов шейдеров (gl.compileShader())	351
Создание объекта программы (gl.createProgram())	353
Подключение объектов шейдеров к объекту программы (gl.attachShader())	354
Компоновка объекта программы (gl.linkProgram())	355

Сообщение системе WebGL о готовности объекта программы (gl.useProgram())	356
Реализация initShaders()	357
В заключение	359

Глава 10.

Продвинутые приемы	360
Вращение объекта мышью	360
Как реализовать вращение объекта	361
Пример программы (RotateObject.js)	361
Выбор объекта	363
Как реализовать выбор объекта	364
Пример программы (PickObject.js)	365
Выбор грани объекта	368
Пример программы (PickFace.js)	368
Эффект индикации на лобовом стекле (ИЛС)	371
Как реализовать ИЛС	371
Пример программы (HUD.html)	372
Пример программы (HUD.js)	373
Отображение трехмерного объекта в веб-странице (3DoverWeb)	375
Туман (атмосферный эффект)	376
Реализация эффекта тумана	376
Пример программы (Fog.js)	377
Использование значения w (Fog_w.js)	379
Создание круглой точки	380
Как нарисовать круглую точку	380
Пример программы (RoundedPoints.js)	382
Альфа-смешивание	383
Как реализовать альфа-смешивание	383
Пример программы (LookAtBlendedTriangles.js)	384
Как должна действовать функция смешивания	385
Альфа-смешивание для трехмерных объектов (BlendedCube.js)	386
Рисование при наличии прозрачных и непрозрачных объектов	388
Переключение шейдеров	389
Как реализовать переключение шейдеров	390
Пример программы (ProgramObject.js)	390
Использование нарисованного изображения в качестве текстуры	394
Объект буфера кадра и объект буфера отображения	395
Как реализовать использование нарисованного объекта в качестве текстуры	397
Пример программы (FramebufferObject.js)	398
Создание объекта буфера кадра (gl.createFramebuffer())	399
Создание объекта текстуры, настройка его размеров и параметров	400
Создание объекта буфера отображения (gl.createRenderbuffer())	401
Связывание объекта буфера отображения с типом и настройка его размера (gl.bindRenderbuffer(), gl.renderbufferStorage())	401
Подключение объекта текстуры, как ссылки на буфер цвета (gl.bindFramebuffer(), gl.framebufferTexture2D())	403

Подключение объекта буфера отображения к объекту буфера кадра (gl.framebufferRenderbuffer())	404
Проверка корректности настройки объекта буфера кадра (gl.checkFramebufferStatus()).....	405
Рисование с использованием объекта буфера кадра	406
Отображение теней.....	407
Как реализуются тени	408
Пример программы (Shadow.js).....	409
Увеличение точности.....	414
Пример программы (Shadow_highp.js).....	415
Загрузка и отображение трехмерных моделей	417
Формат OBJ	419
Формат файла MTL	420
Пример программы (OBJViewer.js).....	421
Объект, определяемый пользователем	424
Пример программы (реализация анализа содержимого файла).....	425
Обработка ситуации потери контекста	432
Как реализовать обслуживание ситуации потери контекста	433
Пример программы (RotatingTriangle_contextLost.js).....	434
В заключение	436
Приложение А.	
В WebGL нет необходимости использовать рабочий и фоновый буферы.....	438
Приложение В.	
Встроенные функции в языке GLSL ES 1.0	441
Функции для работы с угловыми величинами и тригонометрические функции	441
Экспоненциальные функции.....	442
Общие функции.....	443
Геометрические функции	445
Матричные функции	446
Векторные функции.....	447
Функции для работы с текстурами.....	448
Приложение С.	
Матрицы проекций	449
Матрица ортогональной проекции	449
Матрица перспективной проекции	449
Приложение D.	
WebGL/OpenGL: лево- или правосторонняя система координат? ...	450
Пример программы CoordinateSystem.js	451
Удаление скрытых поверхностей и усеченная система координат	453
Усеченная система координат и видимый объем.....	454
Теперь все правильно?	456
В заключение	459

Приложение Е.	
Транспонированная обратная матрица	460
Приложение F.	
Загрузка шейдеров из файлов	464
Приложение G.	
Мировая и локальная системы координат	466
Локальная система координат	466
Мировая система координат	467
Преобразования и системы координат	469
Приложение H.	
Настройка поддержки WebGL в веб-браузере	470
Словарь терминов	472
Список литературы	477
Предметный указатель	478



ПРЕДИСЛОВИЕ

WebGL – это технология рисования, отображения и интерактивного взаимодействия с трехмерной компьютерной графикой в веб-браузерах. Традиционно поддержка трехмерной графики ограничивалась высокопроизводительными компьютерами или специализированными игровыми консолями, а ее программирование требовало применения сложных алгоритмов. Однако, благодаря росту производительности персональных компьютеров и расширению возможностей браузеров стало возможным создание и отображение трехмерной графики с применением веб-технологий. Эта книга представляет собой исчерпывающий обзор возможностей WebGL и постепенно, шаг за шагом знакомит читателя с основами создания приложений WebGL. В отличие от других технологий для работы с трехмерной графикой, таких как OpenGL и Direct3D, WebGL предназначена для использования в веб-страницах и не требует установки специализированных расширений или библиотек. Благодаря этому можно сразу опробовать примеры программ, имея лишь стандартное окружение. А так как все основано на веб-технологиях, вновь созданные программы легко можно публиковать в Сети. Одно из преимуществ WebGL заключается в том, что приложения конструируются как веб-страницы, то есть одна и та же программа успешно будет выполняться на самых разных устройствах, таких как смартфоны, планшетные компьютеры и игровые консоли. Это означает, что WebGL будет оказывать все более усиливающееся влияние на сообщество разработчиков и станет одним из основных инструментов программирования графики.

Кому адресована эта книга

Мы писали эту книгу для специалистов двух основных категорий: веб-разработчиков, ищущих возможность добавления трехмерной графики в свои веб-страницы и приложения, и программистов, занимающихся проблемами реализации трехмерной графики, желающих понять, как применить свои знания в веб-окружении.

Для специалистов из первой категории – веб-разработчиков, хорошо знакомых со стандартными веб-технологиями, такими как HTML и JavaScript, и стремящихся внедрить трехмерную графику в свои веб-страницы или веб-приложения, – WebGL является простым и мощным решением. Эту технологию с успехом можно применять для добавления трехмерной графики в веб-страницы, с целью повысить качество пользовательского интерфейса веб-приложений, и даже для разработки сложнейших графических и игровых приложений, выполняющихся в веб-браузерах.

Специалистам из второй категории – программистам, имеющим опыт использования основных прикладных программных интерфейсов (API) создания трехмерной графики, таких как Direct3D или OpenGL, и интересующимся возможностью применения своих знаний в веб-окружении, – будет любопытно поближе познакомиться с возможностью создания сложных приложений для работы с трехмерными изображениями, выполняющимися в современных веб-браузерах.

Однако, мы постарались сделать книгу доступной для широкого круга читателей и использовали пошаговый подход к знакомству с особенностями WebGL, не предполагая наличия у читателей опыта программирования двух- или трехмерной графики. Мы полагаем, что она будет интересна также:

- обычным программистам, кому будет любопытно узнать, как идет развитие веб-технологий в направлении поддержки графических возможностей;
- студентам, изучающим приемы программирования двух- и трехмерной графики, потому что WebGL позволяет легко экспериментировать с графикой в веб-браузере и не требует установки и настройки полноценной среды программирования;
- веб-разработчикам, исследующим ультрасовременные возможности, поддерживаемые мобильными устройствами с новейшими версиями мобильных веб-браузеров.

О чем рассказывается в этой книге

Эта книга охватывает WebGL 1.0 API, наряду с соответствующими функциями JavaScript. Здесь вы узнаете, как связаны между собой HTML, JavaScript и WebGL, как устанавливать и запускать приложения WebGL и как программировать сложные трехмерные «шейдеры» на JavaScript. Книга подробно рассказывает, как описывать вершинные и фрагментные шейдеры, как реализовать разные усовершенствованные приемы отображения, такие как попиксельное освещение или затенение, а также об основных приемах взаимодействий, такими как выделение трехмерных объектов. В каждой главе создается несколько действующих приложений, на примере которых демонстрируются ключевые особенности WebGL. Закончив читать эту книгу, вы будете готовы писать приложения с применением WebGL, использующие всю мощь веб-браузеров и аппаратной поддержки графики.

Структура книги

Эта книга организована в виде последовательного описания WebGL API и связанных с ним функций, чтобы вы могли выстраивать свое здание знаний WebGL постепенно.

Глава 1 – Обзор WebGL

В этой главе дается короткое представление WebGL, отмечаются некоторые ключевые особенности и преимущества, а также дается история происхождения.

Завершается глава разъяснением взаимоотношений между WebGL, HTML5 и JavaScript, и как использовать веб-браузеры для исследования WebGL.

Глава 2 – Первые шаги в WebGL

В этой главе описывается элемент `<canvas>` и основные возможности WebGL на примере нескольких простых программ. Все программы написаны на JavaScript и используют WebGL для отображения и взаимодействий с простыми геометрическими фигурами на веб-странице. Примеры программ подчеркивают следующие важные моменты: (1) как WebGL использует элемент `<canvas>` и как рисовать на его поверхности; (2) как посредством JavaScript осуществляется связь между HTML и WebGL; (3) работа с простыми функциями рисования WebGL; и (4) роль шейдеров в WebGL.

Глава 3 – Рисование и преобразование треугольников

Эта глава закладывает фундамент для дальнейших исследований особенностей рисования более сложных фигур и манипулирования этими фигурами в трехмерном пространстве. В этой главе рассматриваются: (1) критически важная роль треугольников в трехмерной графике и поддержка рисования треугольников в WebGL; (2) как с помощью множества треугольников рисуются другие простые фигуры; (3) простые преобразования, такие как перемещение, вращение и масштабирование с применением простых уравнений; и (4) как матричные операции упрощают преобразования.

Глава 4 – Дополнительные преобразования и простая анимация

В этой главе вы продолжите исследование преобразований и начнете объединять преобразования в анимации. Здесь вы: (1) познакомитесь с библиотекой матричных преобразований, скрывающей математические тонкости операций с матрицами; (2) научитесь пользоваться библиотекой, позволяющей просто и быстро комбинировать преобразования; и (3) исследуете анимационные эффекты и библиотеку, помогающую организовать анимацию с участием простых фигур. Приемы, представленные в этой главе, образуют основу для конструирования гораздо более сложных программ WebGL и будут использоваться в примерах программ в последующих главах.

Глава 5 – Цвет и текстура

Опираясь на фундамент, заложенный в предыдущих главах, вы более детально познакомитесь с WebGL, исследовав в этой главе следующие три темы: (1) как, помимо координат вершин, передать в вершинный шейдер дополнительную информацию, например, цвет; (2) преобразование фигуры во фрагмент, то есть процедура перехода от вершинного шейдера к фрагментному шейдеру, известная как процесс растеризации; и (3) как «натягивать» изображения (или текстуры) на поверхности фигур или объектов. Эта глава является завершающей в исследовании ключевых особенностей WebGL.

Глава 6 – Язык шейдеров OpenGL ES (GLSL ES)

В этой главе мы оторвемся от изучения примеров программ WebGL и исследуем основные особенности OpenGL ES Shading Language (GLSL ES) – языка программирования шейдеров. Здесь рассказывается о: (1) данных, переменных и типах переменных; (2) векторах, матрицах, структурах, массивах и образцах; (3) операторах, инструкциях управления потоком выполнения и функциях; (4) атрибутах, uniform-переменных и varying-переменных; (5) квалификаторах точности; и (6) препроцессоре и директивах. К концу этой главы вы будете понимать язык GLSL ES и знать, как его использовать для создания различных шейдеров.

Глава 7 – Вперед, в трехмерный мир

Эта глава является первым шагом в трехмерный мир и исследует последствия перехода из двухмерного мира в трехмерный. В частности, вы исследуете: (1) представление точки зрения пользователя в трехмерном мире; (2) как управлять объемом трехмерного пространства; (3) отсечение, или клиппинг (clipping); (4) объекты переднего и заднего плана; и (5) рисование трехмерных объектов – куб. Все эти проблемы оказывают большое влияние на рисование трехмерных сцен и их отображение. Именно решать их совершенно необходимо для создания неотразимых трехмерных сцен.

Глава 8 – Освещение объектов

Основное внимание в этой главе уделяется освещению объектов, исследованию разных источников освещения и их влияния на трехмерные сцены. Освещение играет важную роль в придании реалистичности трехмерным сценам, потому что правильно подобранное освещение придает ощущение глубины.

В этой главе обсуждаются следующие ключевые моменты: (1) тени, затенение и разнотипные источники света, включая точечные, направленные и рассеянные; (2) отражение света в трехмерных сценах и два основных типа отражения: рассеянное и фоновое; и (3) тонкости затенения и как реализовать эффект освещения, чтобы объекты выглядели трехмерными.

Глава 9 – Иерархические объекты

Эта глава завершает описание основных особенностей и приемов программирования с применением WebGL. К концу этой главы вы овладеете всеми основами WebGL и будете в состоянии самостоятельно создавать реалистичные и интерактивные трехмерные сцены. Основное внимание в этой главе уделяется иерархическим объектам, играющим важную роль, потому что позволяют перейти от простых объектов, таких как кубы и блоки, к более сложным конструкциям, которые можно использовать в роли персонажей игр, роботов и даже моделировать на их основе людей.

Глава 10 – Продвинутые приемы

В этой главе затрагиваются различные важнейшие приемы, основанные на приемах, изучаемых в предыдущих главах и позволяющие создавать интерактивные, реалистичные трехмерные изображения. Описания всех приемов сопровождаются

ся законченными примерами, которые вы можете использовать при разработке своих приложений WebGL.

Приложение А – В WebGL нет необходимости использовать рабочий и фоновый буферы

В этом приложении рассказывается, почему программы WebGL не нуждаются в перестановке рабочего и фонового буферов.

Приложение В – Встроенные функции в языке GLSL ES 1.0

Это приложение – справочник по всем встроенным функциям в языке OpenGL ES Shading Language.

Приложение С – Матрицы проекций

В этом приложении представлены матрицы проекций, сгенерированные функциями `Matrix4.setOrtho()` и `Matrix4.setPerspective()`.

Приложение D – WebGL/OpenGL: лево- или правосторонняя система координат?

В этом приложении рассказывается, как WebGL и OpenGL работают с системой координат. Здесь же поясняется, что обе технологии, WebGL и OpenGL, нейтральны к выбору системы координат.

Приложение E – Транспонированная обратная матрица

В этом приложении разъясняется, как с помощью операций обращения и транспонирования матрицы модели можно выполнять преобразования векторов нормали.

Приложение F – Загрузка шейдеров из файлов

В этом приложении рассказывается, как организовать загрузку программ шейдеров из файлов.

Приложение G – Мировая и локальная системы координат

В этом приложении описываются разные системы координат и их применение в трехмерной графике.

Приложение H – Настройка поддержки WebGL в веб-браузере

В этом приложении объясняется, как с помощью дополнительных настроек веб-браузера гарантировать правильное отображение графики, сгенерированной средствами WebGL.

Браузеры с поддержкой WebGL

На момент написания этих строк, технология WebGL поддерживалась браузерами: Chrome, Firefox, Safari и Opera. Печально, что некоторые распространенные браузеры, такие как IE9 (Microsoft Internet Explorer), не имеют такой поддержки. В процессе работы над этой книгой мы пользовались браузером Chrome, выпущенным компанией Google, который помимо WebGL поддерживает еще массу полезных возможностей, таких как консоль для отладки функций. Примеры программ, что приводятся в этой книге, опробованы в следующих окружениях (см. табл. П.1), но мы полагаем, что они будут работать в любых браузерах с поддержкой WebGL.

Таблица П.1. Окружение, в которых опробовались примеры программ

Браузер	Chrome (25.0.1364.152 m)
ОС	Windows 7 и 8
Графическая карта	NVIDIA Quadro FX 380, NVIDIA GT X 580, NVIDIA GeForce GTS 450, Mobile Intel 4 Series Express Chipset Family, AMD Radeon HD 6970

На странице www.khronos.org/webgl/wiki/BlacklistsAndWhitelists можно найти постоянно обновляемый список графических карт, имеющих известные проблемы.

Чтобы проверить, поддерживает ли ваш браузер WebGL, запустите его и перейдите на веб-сайт этой книги: <https://sites.google.com/site/webglbook/>.

Перейдите по ссылке **Chapter 3** и выберите пример `HelloTriangle` в списке. Если вы увидите красный треугольник, как показано на рис. П.1, значит ваш браузер поддерживает WebGL.

Если ваш браузер не смог вывести красный треугольник, загляните в приложение Н, где описывается, как изменить настройки браузера, чтобы активизировать в нем поддержку WebGL.

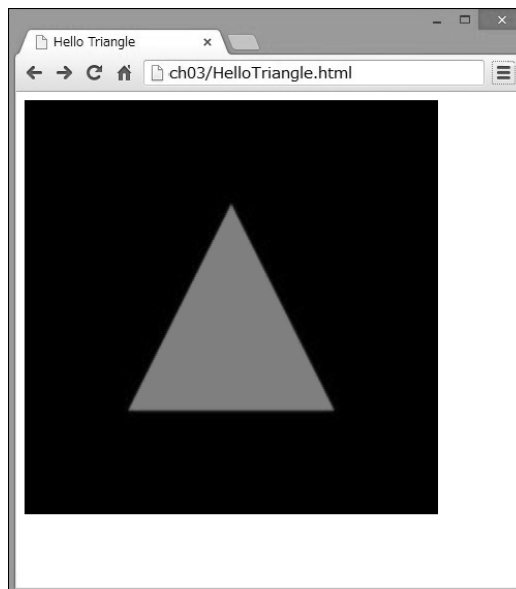


Рис. П.1. При выборе примера `HelloTriangle` должен появиться красный треугольник

Примеры программ и сопутствующие ссылки

Все примеры программ и сопутствующие ссылки, что приводятся в этой книге, доступны на вспомогательных веб-сайтах. Официальный сайт книги находится на сайте издательства: www.informit.com/title/9780321902924. Авторы поддерживают свой сайт книги: <https://sites.google.com/site/webglbook/>.

Последний из этих сайтов содержит ссылки на все примеры программ, что приводятся в этой книге. Вы можете опробовать каждый из них, просто щелкая на ссылках.

Если вы пожелаете внести свои изменения в программы, загрузите zip-файл со всеми примерами, доступный на втором сайте, на свой локальный диск. Обратите внимание, что каждый пример состоит из двух файлов – файла HTML и соответствующего ему файла JavaScript – в одной папке. Например, пример программы `HelloTriangle` состоит из двух файлов: `HelloTriangle.html` и `HelloTriangle.js`. Чтобы запустить программу `HelloTriangle`, выполните двойной щелчок на файле `HelloTriangle.html`.

Типографские соглашения

В этой книге используются следующие типографские соглашения:

- **Жирный шрифт** – используется для выделения ключевых терминов и важных слов.
- *Курсив* – используется для выделения имен параметров и ссылок.
- Моноширинный шрифт – используется для оформления примеров кода; имен методов, функций, переменных; параметров команд; имен объектов JavaScript; имен файлов; тегов HTML.

Благодарности

Нам было приятно получить помощь от многих талантливых личностей в процессе работы над этой книгой – и над начальной японской версией, и над последующей за ней английской версией.

Такафуми Канда (Takafumi Kanda) помог нам с многочисленными фрагментами кода для нашей библиотеки поддержки и примеров программ, без его участия мы просто не смогли бы закончить эту книгу. Ясуко Кикучи (Yasuko Kikuchi), Чи Онума (Chie Onuma) и Ючи Нишизава (Yuichi Nishizawa) оказали неоценимую помощь при работе над ранними версиями книги. Хочется заметить, что один глубокомысленный комментарий госпожи Кикучи буквально остановил работу над книгой и заставил нас полностью переписать несколько разделов, чтобы обеспечить большую техническую точность. Хироюки Танака (Hiroyuki Tanaka) и Казухира Оониши (Kazuhira Oonishi) (iLinx) оказали большую помощь с примерами

ми программ, а Терухиша Камачи (Teruhisa Kamachi) и Тетсуо Юшитани (Tetsuo Yoshitani) помогли в работе над разделами, посвященными HTML5 и JavaScript. Члены рабочей группы WebGL, особенно Кен Рассел (Ken Russell) (Google), Крис Марин (Chris Marin) (Apple) и Дэн Гинзбург (Dan Ginsburg) (AMD) любезно согласились ответить на массу технических вопросов. Мы получили привилегию воспользоваться поддержкой президента Khronos Group, Нейла Треветта (Neil Trevett), и весьма благодарны Хитоши Касаи (Hitoshi Kasai) (руководителю MIACIS Associates) за то, что связал нас с мистером Треветтом и членами рабочей группы WebGL. Мы также хотим выразить благодарность Ксавье Мишелю (Xavier Michel) и Макото Сато (Makoto Sato) (Софийский университет, Токио), оказавшим неоценимую помощь в переводе оригинального текста и решении проблем, возникавшим во время перевода. Английская версия книги тщательно была выверена Джеффом Гилбертом (Jeff Gilbert), Риком Рафии (Rick Rafey) и Даниэлем Ханом (Daniel Haehn), которые дали ценные технические замечания, позволившие значительно улучшить эту книгу. Кроме того, мы хотим выразить благодарность Лауре Левин (Laura Lewin) и Оливии Бесижио (Olivia Basegio) из издательства Pearson, помогавшим в организации публикации книги и обеспечивавшим неуклонное движение вперед.

Мы оба выражаем признательность авторам книг «Red Book» (OpenGL Programming Guide) и «Gold Book» (OpenGL ES 2.0 Programming Guide), выпущенных издательством Pearson, без которых эта книга была бы невозможной. Мы надеемся, что эта книга в какой-то степени послужит выражением нашей признательности.

Об авторах

Доктор Коичи Мацуда (Koichi Matsuda) имеет большой опыт в области разработки пользовательских интерфейсов и новых мультимедийных программных продуктов. Много лет занимался разработкой продуктов и исследованиями в таких компаниях, как NEC, Sony Corporate Research и Sony Computer Science Laboratories. В настоящее время занимает должность руководителя отдельной исследовательской группы, занимающейся изысканиями в области пользовательских интерфейсов и взаимодействий человека с машиной. В свое время занимался проектированием трехмерного виртуального мира «PAW» (Personal Agent-oriented virtual World – личный агент-ориентированный виртуальный мир), с самого начала был вовлечен в разработку стандарта VRML97 (ISO/IEC 14772-1:1997)¹, и продолжает участвовать в жизни обоих сообществ – VRML и X3D (предшественник WebGL). Является автором 15 книг по компьютерным технологиям. Перевел 25 книг на японский язык. Считается опытным экспертом в области пользовательских интерфейсов, взаимодействий человека с машиной, обработки естественных языков, построения развлекательных сетевых служб и интерфейсов агентных систем. Всегда следит за появлением новых возможностей в области

¹ VRML (Virtual Reality Modeling Language – язык моделирования виртуальной реальности). – *Прим. перев.*

технологий. Помимо профессиональной деятельности любит посещать горячие термальные источники, отдыхать на море летом, увлекается виноделием, а также рисованием иллюстраций к книгам издательства MANGA. Защитил докторскую диссертацию в Токийском университете. Связаться с ним можно по адресу: WebGL.prog.guide@gmail.com.

Доктор Роджер Ли (Rodger Lea) – адъюнкт-профессор в центре Media and Graphics Interdisciplinary Centre, в Университете Британской Колумбии (University of British Columbia), с интересом в области мультимедиа и распределенных вычислений. Имеет более чем 20-летний опыт руководства исследовательской группой в академической и промышленной среде, занимался разработкой ранних версий трехмерных миров, участвовал в работе над стандартом VRML97, разрабатывал мультимедийные операционные системы, участвовал в создании прототипов интерактивного цифрового телевидения и в разработке стандартов домашних мультимедийных сетей. Опубликовал более 60 статей и написал 3 книги, является держателем 12 патентов. В настоящее время занимается исследованием растущего «Интернета вещей» (Internet of Things, IoT)², но сохраняет влечение ко всему, что связано с графикой и мультимедиа.

² Концепция вычислительной сети физических объектов («вещей»), оснащённых встроенными технологиями для взаимодействия друг с другом или с внешней средой. – *Прим. перев.*

ГЛАВА 1.

Обзор WebGL

WebGL – это технология, позволяющая рисовать, отображать и взаимодействовать со сложной, интерактивной трехмерной компьютерной графикой в веб-браузерах. Традиционно поддержка трехмерной графики ограничивалась высокопроизводительными компьютерами или специализированными игровыми консолями, а ее программирование требовало применения сложных алгоритмов. Однако, благодаря росту производительности персональных компьютеров и расширению возможностей браузеров стало возможным создание и отображение трехмерной графики с применением веб-технологий. WebGL, в сочетании с HTML5 и JavaScript, делает трехмерную графику доступной для веб-разработчиков и открывает возможность создания веб-приложений следующего поколения, с простыми и понятными пользовательскими интерфейсами и веб-контентом. Некоторые примеры таких интерфейсов показаны на рис. 1.1. В ближайшие годы можно ожидать появления программ, использующих WebGL, на самых разных устройствах, от стандартных ПК до бытовой электроники, смартфонов и планшетных компьютеров.



Рис. 1.1. Сложная трехмерная графика в браузере. © 2011 Хиромаса Хори (Hiromasa Horie) (слева), © 2012 Коичи Мацуда (Kouichi Matsuda) (справа)

HTML5, последняя версия стандарта HTML, дополнила традиционный HTML средствами для работы с 2-мерной графикой, сетевых взаимодействий и доступа к локальному хранилищу. С появлением HTML5 браузеры быстро развились из простых инструментов отображения в сложные платформы для приложений. Однако с развитием появилась потребность в интерфейсах и графических возможностях, не ограничивающихся двумя измерениями. По этой причине была создана технология WebGL, занявшая центральное место в создании слоя отображения для новых приложений с трехмерной графикой, выполняющихся под управлением браузеров.

Традиционно для воспроизведения трехмерной графики требовалось создавать автономные приложения на таких языках программирования, как C или C++, использующие мощные графические библиотеки, такие как OpenGL и Direct3D. Однако, с появлением WebGL трехмерную графику стало возможным воспроизводить на стандартных веб-страницах, используя знакомые языки HTML и JavaScript – с небольшим объемом кода.

Особенно важно, что WebGL поддерживается браузерами, как технология по умолчанию для отображения трехмерной графики, что дает возможность использовать ее непосредственно, без необходимости устанавливать специальные расширения или библиотеки. Но самое замечательное, что основой всего этого является браузер, то есть вы можете запускать приложения WebGL на самых разных платформах, от быстродействующих ПК до бытовой электроники, планшетных компьютеров и смартфонов.

Эта глава познакомит вас с технологией WebGL в общих чертах, обозначит ключевые ее особенности и достоинства, а так же расскажет о происхождении WebGL. Кроме того, здесь вы узнаете, как связаны между собой WebGL, HTML5 и JavaScript, и познакомитесь со структурой программ, использующих WebGL.

Достоинства WebGL

С развитием HTML, веб-разработчики получали новые возможности для создания все более и более сложных веб-приложений. Первоначально язык HTML предлагал только возможность отображения статического контента, но с добавлением поддержки языка сценариев JavaScript появилась возможность реализации более сложных взаимодействий и отображения динамического контента. Стандарт HTML5 добавил дополнительные возможности, включая поддержку 2-мерной графики в виде тега `canvas`. Это дало возможность размещать на страницах разные графические элементы, от танцующих мультипликационных персонажей до вывода анимации, соответствующей вводу пользователя.

Создание технологии WebGL стало еще одним шагом вперед, позволившим отображать и манипулировать трехмерной графикой на веб-страницах с помощью JavaScript. Благодаря WebGL, разработчики могут создавать продвинутые пользовательские интерфейсы, трехмерные игры и использовать трехмерную графику для визуализации различной информации из Интернета. Несмотря на внушительные возможности, WebGL отличается от других технологий доступностью и простотой в использовании, что способствует ее быстрому распространению. В частности:

- вы можете заниматься разработкой приложений с трехмерной графикой, используя только текстовый редактор и браузер;
- вы легко можете опубликовать свое приложение с трехмерной графикой, используя стандартные веб-технологии, и сделать его доступным для своих друзей и других пользователей Сети;
- вы можете использовать всю широту возможностей браузеров;
- изучение и использование WebGL не вызывает никаких сложностей, потому что имеется огромное количество обучающих материалов и примеров программного кода.

Вы можете заниматься разработкой приложений с трехмерной графикой, используя только текстовый редактор

Одной из замечательных особенностей разработки приложений с использованием WebGL является отсутствие необходимости в установке среды разработки. Как уже говорилось выше, так как WebGL является встроенной технологией, для разработки приложений с трехмерной графикой не требуется устанавливать специализированные инструменты, такие как компиляторы и компоновщики. Чтобы опробовать примеры программ, которые приводятся в этой книге, нужен всего лишь браузер с поддержкой WebGL. Если вы захотите поэкспериментировать с этими приложениями или написать свое, достаточно иметь стандартный текстовый редактор (например, Notepad или TextEdit). На рис. 1.2 демонстрируется приложение WebGL, выполняющееся в браузере Chrome, и файл HTML, открытый в редакторе Notepad. Файл с разметкой HTML загружает сценарий на языке JavaScript (RotateObject.js), использующий возможности технологии WebGL, который так же можно редактировать в простом текстовом редакторе.



Браузер (Chrome)



Notepad

Рис. 1.2. Для разработки WebGL-приложений с трехмерной графикой необходимы только эти инструменты

Публикация приложений с трехмерной графикой не вызывает сложностей

Традиционно, приложения с трехмерной графикой разрабатывались на таких языках, как С или С++ и компилировались в выполняемые файлы для конкретных платформ. Это означает, например, что версия для Macintosh не будет работать в Windows или Linux. Кроме того, пользователям требовалось установить не только само приложение, но еще и дополнительные библиотеки, что только добавляло сложностей для тех, кто стремился поделиться своим приложением.

Приложения на основе WebGL, напротив, благодаря тому, что состоят из файлов HTML и JavaScript, легко могут распространяться, так как для этого достаточно всего лишь разместить их на веб-сервере, как обычные веб-страницы, или отправить файлы HTML и JavaScript по электронной почте. Например, на рис. 1.3 показаны примеры WebGL-приложений, опубликованных компанией Google и доступных по адресу: <http://code.google.com/p/webglsamples/>.



Рис. 1.3. Примеры WebGL-приложений, опубликованных компанией Google (с разрешения автора, Грегга Тавареса (Gregg Tavares), Google)

Вы можете использовать всю широту возможностей браузеров

Поскольку WebGL-приложения фактически являются веб-страницами, появляется возможность использовать всю широту возможностей веб-браузеров, таких как добавление кнопок, вывод диалогов и текста, проигрывание аудио- и видеороликов, а также обмен данными с веб-серверами. Все эти дополнительные особенности доступны изначально, тогда как в традиционных приложениях с трехмерной графикой их необходимо реализовывать явно.

Изучение и использование WebGL не вызывает никаких сложностей

Спецификация WebGL основана на открытом стандарте OpenGL, который уже много лет используется в качестве основы для разработки графических приложений, видеоигр и систем автоматизированного проектирования. Название «WebGL» можно интерпретировать как «OpenGL для веб-браузеров». Так как

технология OpenGL используется на самых разных платформах уже более 20 лет, и за эти годы было написано огромное количество учебников, справочников, статей и примеров программ, использующих ее, которыми с успехом можно пользоваться и при изучении WebGL.

История происхождения WebGL

Из технологий отображения трехмерной графики на персональных компьютерах наибольшее распространение получили Direct3D и OpenGL. Direct3D – составная часть пакета технологий Microsoft DirectX – это технология отображения трехмерной графики, предназначенная для использования на платформе Windows. Она является лицензионным программным интерфейсом (API) и контролируется корпорацией Microsoft. Альтернативная ей технология OpenGL получила гораздо более широкое распространение, благодаря ее открытости. Реализации OpenGL доступны для Macintosh, Linux и различных устройств, таких как смартфоны, планшетные компьютеры и игровые консоли (PlayStation и Nintendo). Имеются также реализации для Windows, которые представляют прямые альтернативы Direct3D.

Первоначально технология OpenGL была разработана в компании Silicon Graphics Inc. и опубликована как открытый стандарт в 1992 году. С тех пор вышло несколько версий стандарта OpenGL, оказавшего большое влияние на развитие трехмерной графики, создание программных продуктов и даже киноиндустрию. Последней версией OpenGL для настольных ПК на момент написания этих строк была версия 4.3. Несмотря на то, что технология WebGL корнями уходит в OpenGL, в действительности она является дальнейшим развитием версии OpenGL для встраиваемых систем, таких как смартфоны и игровые консоли. Эта версия, известная как OpenGL ES (for Embedded Systems – для встраиваемых систем), создана в 2003–2004 годах и затем была обновлена в 2007 году (ES 2.0) и в 2012 (ES 3.0). WebGL основана на версии ES 2.0. В последние годы быстро растет число устройств и процессоров, поддерживающих спецификацию, в число которых входят смартфоны (iPhone и Android), планшетные компьютеры и игровые консоли. Отчасти такой успех обусловлен добавлением в OpenGL ES новых особенностей, а также удалением множества устаревших и ненужных функций, что сделало спецификацию более легковесной и при этом достаточной для создания выразительной и привлекательной трехмерной графики.

На рис. 1.4 показано, как связаны между собой OpenGL, OpenGL ES 1.1/2.0/3.0 и WebGL. Так как сама спецификация OpenGL продолжала непрерывное развитие от версии 1.5 к версиям 2.0 ... 4.3, спецификация OpenGL ES была стандартизована как подмножество определенных версий OpenGL (OpenGL 1.5 и OpenGL 2.0).

Как показано на рис. 1.4, переход к версии OpenGL 2.0 означался появлением новой важной особенностью – поддержкой программных шейдеров. Эта поддержка была перенесена в OpenGL ES 2.0 и стала одним из основных элементов спецификации WebGL 1.0.