

УДК 004.738.5:004.4WebGL
ББК 32.973.202-018.2
МЗЗ

МЗЗ Коичи Мацуда, Роджер Ли

WebGL: программирование трехмерной графики. / Пер. с англ. Киселев А. Н. – М.: ДМК Пресс, 2015. – 494 с.: ил.

ISBN 978-5-97060-146-4

WebGL является новой веб-технологией, позволяющей использовать в браузере преимущества аппаратного ускорения трехмерной графики без установки дополнительного программного обеспечения. WebGL основана на спецификации OpenGL и привносит новые концепции программирования трехмерной графики в веб-разработку.

Снабженная большим количеством примеров, книга показывает, что овладеть технологией WebGL совсем несложно, несмотря на то, что она выглядит незнакомой и инородной. Каждая глава описывает один из важнейших аспектов программирования трехмерной графики и представляет разные варианты их реализации. Отдельные разделы, описывающие эксперименты с примерами программ, позволяют читателю исследовать изучаемые концепции на практике.

Издание предназначено для программистов, желающих научиться использовать в своих веб-проектах 3D-графику.

УДК 004.738.5:004.4WebGL
ББК 32.973.202-018.2

Original English language edition published by Pearson Education, Inc., Permissions Department, One Lake Street, Upper Saddle River, New Jersey 07458. Copyright © 2013 Pearson Education, Inc. Russian-language edition copyright © 2014 by DМК Press. 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 978-0-321-90292-4 (англ.)
 ISBN 978-5-97060-146-4 (рус.)

Copyright © 2013 Pearson Education, Inc.
 © Оформление, перевод на русский язык,
 издание, ДМК Пресс, 2015



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
Основа 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 объектов299

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 объекты329

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 (g.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