

ОБЗОР ПРИКЛАДНОГО ПРОГРАММНОГО ИНТЕРФЕЙСА WEBGL

Ковтунов Д.С. Email: Kovtunov640@scientifictext.ru

Ковтунов Дмитрий Сергеевич – студент,
кафедра космических информационных систем,
Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования
Московский технологический университет, г. Москва

Аннотация: в данной статье приведен общий обзор программной библиотеки, позволяющей создавать интерактивную 3D-графику для широкого спектра браузеров, – библиотеки WebGL. Основными вопросами, рассматриваемыми в данной статье, являются способы работы в WebGL. Также исследована терминология библиотеки, раскрыто понятие шейдера. Кроме того, рассмотрены способы получения данных шейдером, а также описаны компоненты, которые используют при работе в библиотеке 3D-графики. Раскрыты основные преимущества данной библиотеки.

Ключевые слова: WebGL, 3D-графика, шейдер, объекты, JavaScript, программирование.

REVIEW OF THE WEBGL APPLICATION PROGRAMMING INTERFACE

Kovtunov D.S.

¹Kovtunov Dmitry Sergeevich – Student,
SPACE INFORMATION SYSTEMS DEPARTMENT,
FEDERAL STATE BUDGET EDUCATIONAL INSTITUTION OF HIGHER EDUCATION MOSCOW TECHNOLOGICAL
UNIVERSITY, MOSCOW

Abstract: this article considers a general review of a program library that allows to create an interactive 3D-graphics for a wide range of browsers – the WebGL library. Main points that are considered in this article are ways of work in WebGL library. The terminology of the library was also researched, a term “shader” was revealed. Moreover, ways to get data with a shader, and components that uses 3D-graphics for a work in the WebGL library also were described. Main advantages of this library are revealed.

Keywords: WebGL, 3D-graphic, shader, objects, JavaScript, programming.

УДК 004.94

Современные технологии развиваются с каждым годом все интенсивнее. По сравнению с парой тройкой лет компьютерная графика шагнула на несколько шагов вперед.

На сегодняшний день одной из технологий, предназначенной для рисования и отображения интерактивной 2D- и 3D-графики в веб-браузерах является WebGL. Первоначально данная технология была основана OpenGL ES 2.0 версии спецификации OpenGL для таких устройств как iPhone от Apple и iPad.

Ее основная цель это обеспечение переносимости между различными операционными системами и устройствами. Данная спецификация была выпущена в 2011 году.

Технология WebGL позволяет рисовать графику в браузере, используя возможности видеокарты. WebGL – это средство растеризации [2]. На основе написанного кода оно отображает точки, линии и треугольники. Код, при помощи которого реализуют необходимые проекты, представлен в виде пар функций, совместная пара функций называется программой. Пара данных функций являются вершинным и фрагментным шейдером.

Вершинный шейдер вычисляет положение вершины. В зависимости от положений вершин, которые возвращает функция, WebGL затем можно растеризовать различные примитивы, включая точки, линии или треугольники.

Для процесса растеризации примитивов WebGL использует вторую функцию - фрагментный шейдер. Фрагментный шейдер вычисляет цвет по каждому пикселю примитива, отрисовываемого в данный момент.

API WebGL настраивается для работы данных функций. Для каждого объекта, который необходимо отрисовать устанавливаются настройки через вызов `gl.drawArrays` или `gl.drawElements`, который выполняет шейдеры на графическом процессоре.

Способы получения данных шейдером разделяют на:

- атрибуты – определяют, каким образом данные из буферов передаются в вершинный шейдер;
- буфер – массив бинарных данных, загруженных в графический процессор. Данные могут содержать как положение вершины, нормалей, так и координаты, цвета вершин и много другое [3]. Доступ к буферам не произвольный. Вместо этого вершинный шейдер выполняется заданное количество раз и

каждый раз, когда он выполняется, выбирается следующее значение каждого из указанных буферов и назначается атрибуту;

- текстуры – это массивы, к которым возможен доступ в программе шейдера. В большинстве случаев в текстуру помещают картинку. Помимо этого в нее можно поместить что-то отличное от набора цветов, так как текстуры являются простым набором данных;

- Uniform-переменные – позволяют данные из вершинного шейдера передать фрагментному. Во фрагментном шейдере получают интерполированные значения вершинного шейдера – это зависит от того, отображают ли точки линии или треугольники.

Для WebGL основой являются две вещи - координата пространства отсечения и цвет. Для этого используют два шейдера - вершинным задаются координаты пространства отсечения, фрагментный же отвечает за цвет.

Координаты пространства отсечения всегда находятся в диапазоне от -1 до +1 вне зависимости от размера canvas.

В основном большинство 3D движков способны генерировать шейдеры на лету, для этого используются различные шаблоны.

Основными преимуществами WebGL являются:

- автоматическое управление памятью. Если сравнивать WebGL со сходным продуктом OpenGL, где некоторые операции освобождения и выделения проводят вручную, в WebGL такая необходимость отсутствует. Таким образом при выходе JavaScript переменной из области видимости, память, занимаемая ей, автоматически освобождается. Данные действия облегчают работу программиста, значительно уменьшают размер кода, что делает его более понятным и структурированным.

- программирование JavaScript. Данный язык программирования является «родным» для многих программистов. Работая с JavaScript можно получить доступ ко всем DOM-элементам. С ними легко работать, в отличие от общения с апплетами. Программирование WebGL на JavaScript позволяет лучше интегрировать приложения WebGL с другими библиотеками JavaScript [4].

- проницаемость. Современные технологии позволяют устанавливать веб-браузерам, которые поддерживают JavaScript на планшеты и смартфоны.

- производительность. Приложения WebGL сопоставимы эквивалентным автономным приложениям. Это происходит благодаря возможности доступа WebGL к локальным аппаратным ускорителям графики.

- нулевая компиляция. Так как WebGL написан на JavaScript, уходит необходимость компилирования кода предварительно. За счет этого появляется возможность изменять работу на лету и следить за изменениями в 3D веб-приложении.

Для работы в библиотеке 3D-графики в WebGL используют следующие компоненты:

- Canvas. Данный HTML5- элемент является заполнителем. Доступ к нему осуществляется при помощи объектной модели документа (DOM) через JavaScript.

- Объекты. Сущности, состоящие из треугольников и составляющие часть сцены.

- Свет. Особенностью 3D мира является невозможность что-либо увидеть без света. При помощи шейдеров WebGL моделирует свет на сцене. В соответствии с физическими законами те или иные 3D – объекты поглощают свет.

- Камера. Данный компонент является холстом в 3D-мире. При помощи нее можно исследовать сцену.

Библиотека WebGL многогранна и позволяет создавать необычные эффекты, сцены и игры.

Примерами работы WebGL являются:

- Nucleal – один из красивейших элементов слайд-шоу, использующийся для перехода между фотографиями в галерее.

- Google Maps Cube – игра, в которой на улицах Google Map гоняют шарик в лабиринте из улиц, метка доставляется в определенные места. Все это располагается на кубе.

- Cube Slam – в данной игре пользователь играет в аэрохоккей с медведем. Teach Me to Fly – симуляция полета персонажа между небоскребами.

- Но одним из самых ярких примеров работы WebGL является Google Maps, который использует данную библиотеку для отрисовки векторной карты [1].

WebGL имеет достаточно простой API. При этом программист может выполнять как простые вещи при помощи данной технологии, так и справляться со сложными задачами, используя только две функции – вершинный и фрагментный шейдер.

На сегодняшний день WebGL концептуальное средство растеризации, которое довольно легко понять и освоить.

Список литературы / References

1. Впечатляющие примеры WebGL. [Электронный ресурс]. Режим доступа: <https://habrahabr.ru/post/190388/> (дата обращения: 19.04.2018).
2. *Паризи Тони*. WebGL: Up and Running: создание 3D-графики для веб-сайта/ Тони Паризи. O'Reilly Media, Inc, 2012. 211с.
3. *Мацуда Ли*. WebGL. Программирование трехмерной графики / Мацуда Ли. ДМК-Пресс, 2015.
4. Основы работы с WebGL [Электронный ресурс]. Режим доступа: <https://metanit.com/web/webgl/> (дата обращения: 19.04.2018).