**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ**

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ**

**«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук

Департамент программной инженерии

|  |  |  |
| --- | --- | --- |
| СОГЛАСОВАНО  Профессор  департамента программной инженерии  кандидат технических наук  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Е.М. Гринкруг  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2018 г. |  | УТВЕРЖДАЮ  Академический руководитель образовательной программы «Программная инженерия»    \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.В. Шилов  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2018 г. |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | ***Подп. и дата*** |  | | ***Инв. № дубл.*** |  | | ***Взам. инв. №*** |  | | ***Подп. и дата*** |  | | ***Инв. № подл*** | **RU.17701729.03.05-01** | | **РЕАЛИЗАЦИЯ ПОДМНОЖЕСТВА СТАНДАРТА ТРЕХМЕРНОЙ ГРАФИКИ СРЕДСТВАМИ БИБЛИОТЕКИ WEBGL**  **Руководство программиста**  **ЛИСТ УТВЕРЖДЕНИЯ**  **RU.17701729.03.05-01 33 01-1-ЛУ** | | |
|  |  | |
| Исполнитель:  студентка группы БПИ162  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / Казанцева А.Р. /  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2018 г. | |
|  | | |
|  | |  |

**2018**

|  |  |  |  |
| --- | --- | --- | --- |
| **УТВЕРЖДЕНО**  **RU.17701729.03.05-01 33 01-1** | |  |  |
| |  |  | | --- | --- | | ***Подп. и дата*** |  | | ***Инв. № дубл.*** |  | | ***Взам. инв. №*** |  | | ***Подп. и дата*** |  | | ***Инв. № подл*** | **RU.17701729.03.05-01** | | **ПРОГРАММА КОДИРОВАНИЯ И ДЕКОДИРОВАНИЯ**  **АЛГЕБРОГЕОМЕТРИЧЕСКИХ КОДОВ**  **Руководство программиста**  **RU.17701729.03.05-01 33 01-1**  **Листов 15** | | | | |
|  |  | | | |
|  | | | |
|  | | | | |

**2018**

|  |  |  |
| --- | --- | --- |
|  |  |  |

СОДЕРЖАНИЕ

[СОДЕРЖАНИЕ 3](#_Toc514232115)

[1. Назначение программы 4](#_Toc514232116)

[**1.1 Функциональное назначение** 4](#_Toc514232117)

[**1.2 Эксплуатационное назначение** 4](#_Toc514232118)

[2. Условия выполнения программы 4](#_Toc514232119)

[**2.1 Минимальный состав технических средств** 4](#_Toc514232120)

[**2.2 Минимальный состав программных средств** 4](#_Toc514232121)

[**2.3** **Требования к пользователю** 5](#_Toc514232122)

[3. Использование библиотеки 5](#_Toc514232123)

[3.3 Состав элементов 5](#_Toc514232124)

[3.4 Примеры использования 7](#_Toc514232125)

[**Инициализация сцены:** 7](#_Toc514232126)

[**Создание прямоугольного параллелепипеда и освещения:** 8](#_Toc514232127)

[**Изменение параметров:** 9](#_Toc514232128)

[**Загрузка произвольных моделей:** 9](#_Toc514232129)

[**Выбор типа навигации:** 10](#_Toc514232130)

[ПРИЛОЖЕНИЕ 1 12](#_Toc514232131)

[ПРИЛОЖЕНИЕ 2 14](#_Toc514232132)

[ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ 16](#_Toc514232133)

# **1. Назначение программы**

## **1.1 Функциональное назначение**

Библиотека предназначена для предоставления возможности работы с 3D-графикой непосредственно средствами Web-браузера, без установки каких-либо иных специальных программных средств.

Библиотека при подключении к проекту определяет наличие специальной трехмерной сцены1 в виде тега1 <canvas is=”my-scene”> </canvas>. При наличии хотя бы одного тега данного формата библиотека начинает взаимодействовать с графическим контекстом webgl1 этого тега и, опираясь на дочерние теги1, описываемые в соответствии со стандартом трехмерной графики X3D[12] и спецификацией данной библиотеки[13], отрисовывает трехмерные объекты, обращаясь к WebGL API1.

## **1.2 Эксплуатационное назначение**

Подключенная к проекту библиотека определённые библиотекой (нестандартизованные) DOM-элементы[[1]](#footnote-1), определенные спецификацией[13] и обозначающие те или иные элементы стандарта трехмерной графики X3D[12], преобразует в трехмерные объекты и отрисовывает в DOM-элементе1 Canvas, тем самым позволяя работать с трехмерной графикой в веб-браузере:

* не используя никаких плагинов1;
* декларируя элементы непосредственно в привычном html-коде;
* не углубляясь в низкоуровневую работу с шейдерами1 и GLSL1.

# **2. Условия выполнения программы**

## **2.1 Минимальный состав технических средств**

1. NVIDIA >= 257.21 или ATI/AMD >= 10.6 или Intel driver версии от сентября 2010.

## **2.2 Минимальный состав программных средств**

1. один из следующих браузеров:

Таблица 1. Совместимость библиотеки с браузерами.

|  |  |
| --- | --- |
| Наименование | Версия |
| Edge | 16 выше |
| Firefox | 59 и выше |
| Chrome | 49 и выше |
| Safari | TP, 11 |
| Яндекс.Браузер | 17 и выше |
| iOS Safari | 10.3, 11.2, 11.3 |
| Chrome Android | 66 |
| UC for Android | 11.8 |
| Samsung Internet | 4, 6.2 |

1. операционная система Windows XP и более поздние версии, Mac OS X 10.5 и более поздние версии, Unix-подобная операционная система не позднее 2010 года выпуска.

## **Требования к пользователю**

Для использования библиотеки не требуется никаких особых знаний.

# **Использование библиотеки**

# **Состав элементов**

Таблица 2 . Теги и атрибуты

|  |  |  |
| --- | --- | --- |
| **Тег** | **Атрибуты** | **Значение атрибутов** |
| **my-scene**  Определяет трехмерную сцену. | fon-color | *Значение по умолчанию*: “255 255 255 0”  Цвет фона трехмерной сцены. |
| **my-camera**  Определяет камеру. *Обязательный элемент.* | view-angle | *Значение по умолчанию*: “60”  Угол фрустума в градусах. |
| z-far | *Значение по умолчанию*: “2000”  Максимальное видимое значение координат по оси z. |
| z-near | *Значение по умолчанию*: “1”  Минимальное видимое значение координат по оси z. |
| camera-rotation | *Значение по умолчанию*: “0 0 0”  Углы вращения по осям x,y,z для камеры. |
| camera-position | *Значение по умолчанию*: “0 0 0”  Начальная позиция камеры. |
| navigation-type | *Значение по умолчанию*: “”  Тип навигации в нашем мире. Принимает одно из двух значений: object или camera. |
| **my-directed-light**  Определяет направленный свет. | fon-light-color | *Значение по умолчанию*: “100 100 100”  Rgb-цвет фонового освещения (значения в пределах 0..255) |
| directed-light-color | *Значение по умолчанию*: “200 200 200”  Rgb-цвет направленного освещения (значения в пределах 0..255) |
| direction | *Значение по умолчанию:* “0 0 -1”  Направление направленного освещения. |
| **my-transform**  Определяет положение некоторого трехмерного объекта в пространстве сцены. | translation | *Значение по умолчанию:* "0 0 0"  Смещение относительно центра сцены. |
| rotation | *Значение по умолчанию:* "0 0 0"  Поворот относительно осей сцены. |
| scale | *Значение по умолчанию:* "1 1 1"  Масштаб объекта. |
| **my-shape**  Определяет форму некоторого трехмерного объекта.  *Обязательно имеет дочерний элемент.* | - | - |
| **my-indexed-face-set**  Определяет произвольную фигуру. | model | *Обязательный атрибут.*  Путь к .obj файлу - модели, которую необходимо отрендерить. |
| **my-box**  Определяет прямоугольный параллелепипед. | size | *Значение по умолчанию:* "10 10 10"  Размеры параллелепипеда по соответствующим сторонам. |
| **my-appearance**  Определяет внешний вид некоторого трехмерного объекта.  *Обязательно имеет дочерний элемент.* | - | - |
| **my-color**  Определяет цвет некоторого трехмерного объекта. | color | *Обязательный атрибут.*  *Значение по умолчанию:* "255 255 255"  Rgb-цвет (значения в пределах 0..255) |

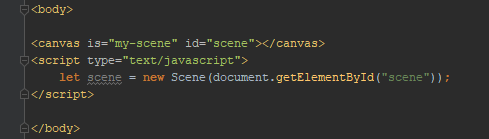
# **Примеры использования**

## **Инициализация сцены:**

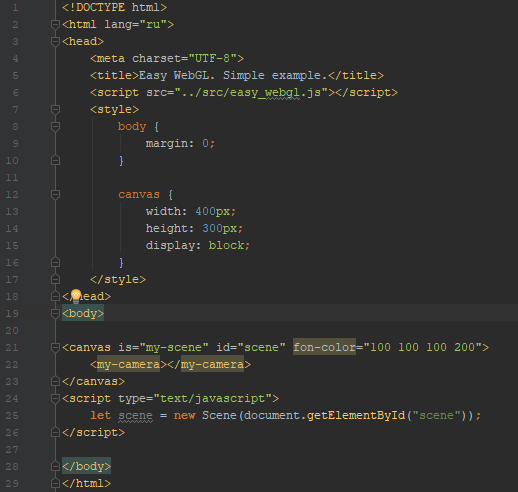
Для написания простейшей программы необходимо в коде html-страницы подключить библиотеку easy\_webgl и определить сцену следующим образом:



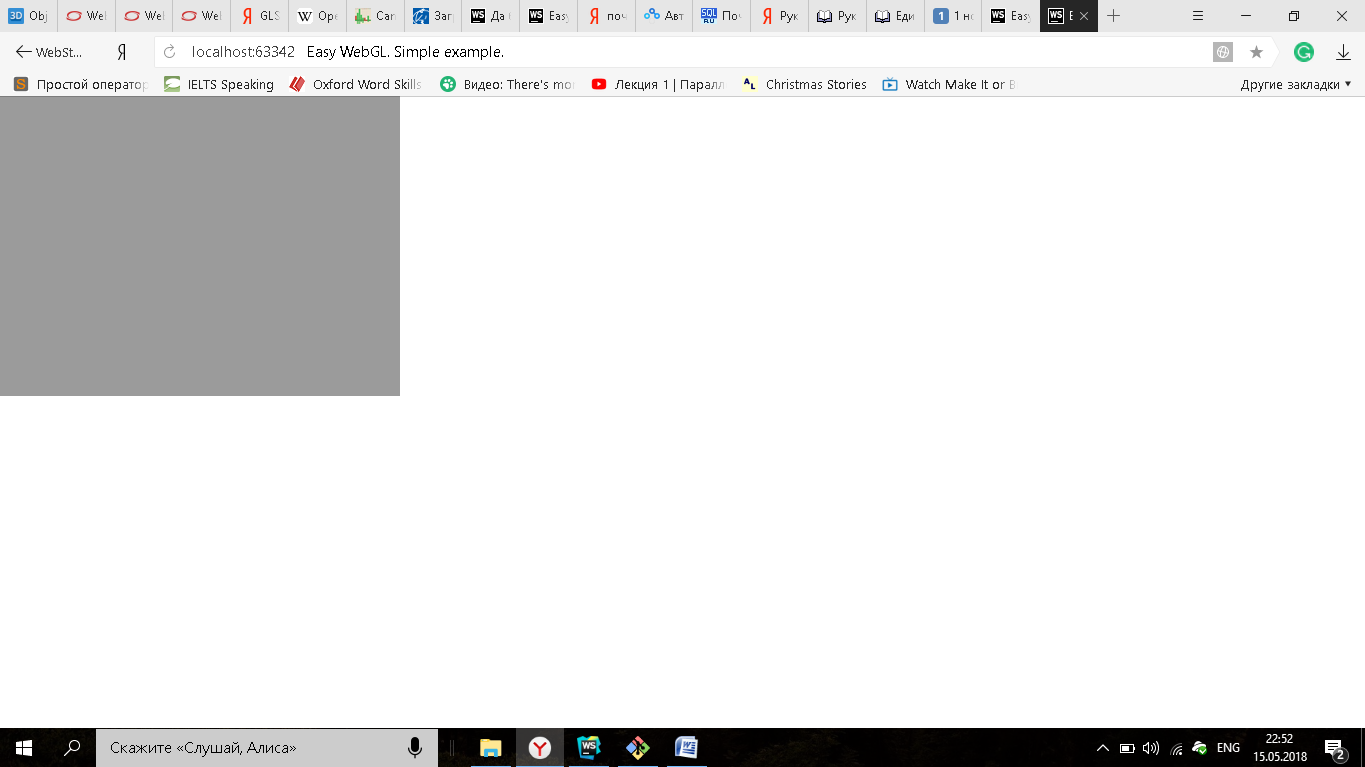
И написать скрипт, инициализирующий эту сцену:



Для того, чтобы убедиться, что всё работает окрасим сцену, определим простейшую камеру и придадим сцене размеры. Размеры передаем элементу canvas через CSS или атрибуты. Для окраски сцены используем специальный атрибут fon-color. Тег my-camera – обязательный тег в трехмерной сцене, определяющий камеру, т.е. зрителя.



Запустив данный код увидим сцену на экране:



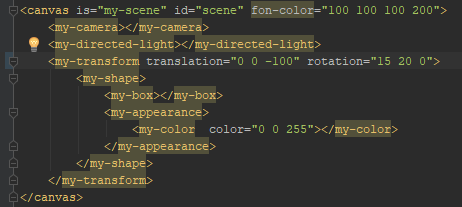
## **Создание прямоугольного параллелепипеда и освещения:**

Теперь определим первую фигуру на нашей сцене. Пусть это будет куб. Для использования библиотеки необходимо запомнить, что у каждого трехмерного объекта есть три важных составляющих:

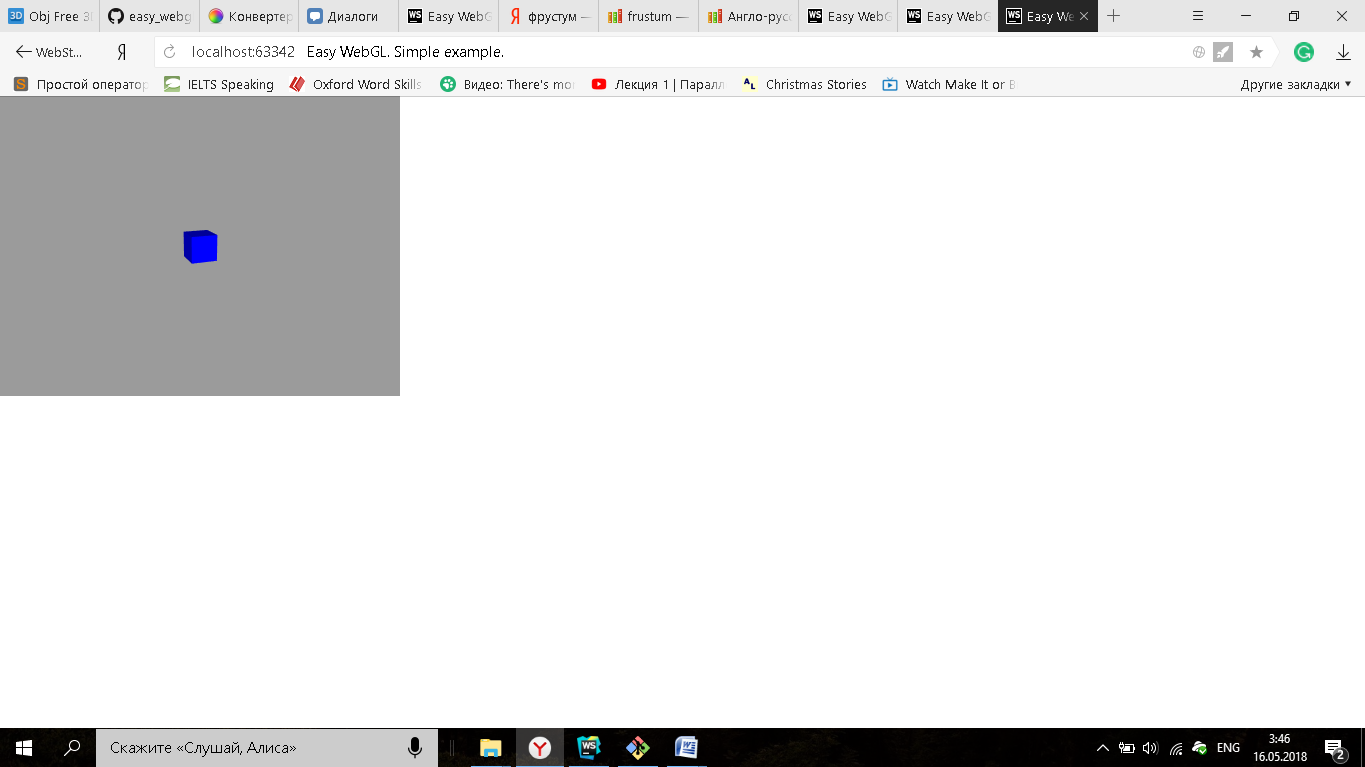
* Положение в мире (определяется тегом my-transform);
* Форма (определяется тегом my-shape и обязательно дополняется дочерними);
* Внешний вид (определяется тегом my-appearance shape и обязательно дополняется дочерними).

Каждая из этих составляющих выражается определенным тегом, который может иметь дополняющие его дочерние теги.

Итак, для создания простейшего кубика определим следующие элементы:



Заметим, что у камеры, трансформа и цвета есть атрибуты. Атрибут для цвета обязателен, для трансформа и камеры нет, но если мы ничего не укажем ни там, ни там, то ничего и не увидим. Это происходит т.к. согласно предыдущему разделу и бокс, и камера имеют начальную позицию в начале координат. Оставлять их на своих местах все равно что пытаться увидеть свои внутренние органы. Поэтому мы двигаем камеру на зрителя(в нашей системе координат изначально X смотрит строго вправо, Y – строго вверх, а Z – на зрителя. Центр координат совпадает с центром экрана). Таким образом, мы двигаем камеру от себя, а бокс на себя.



Благодаря атрибуту rotation мы видим, что фигура имеет объем.

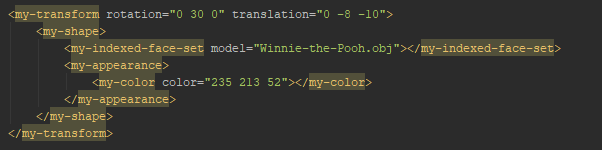
## **Изменение параметров:**

Изменение параметров можно производить согласно информации из предыдущего раздела.

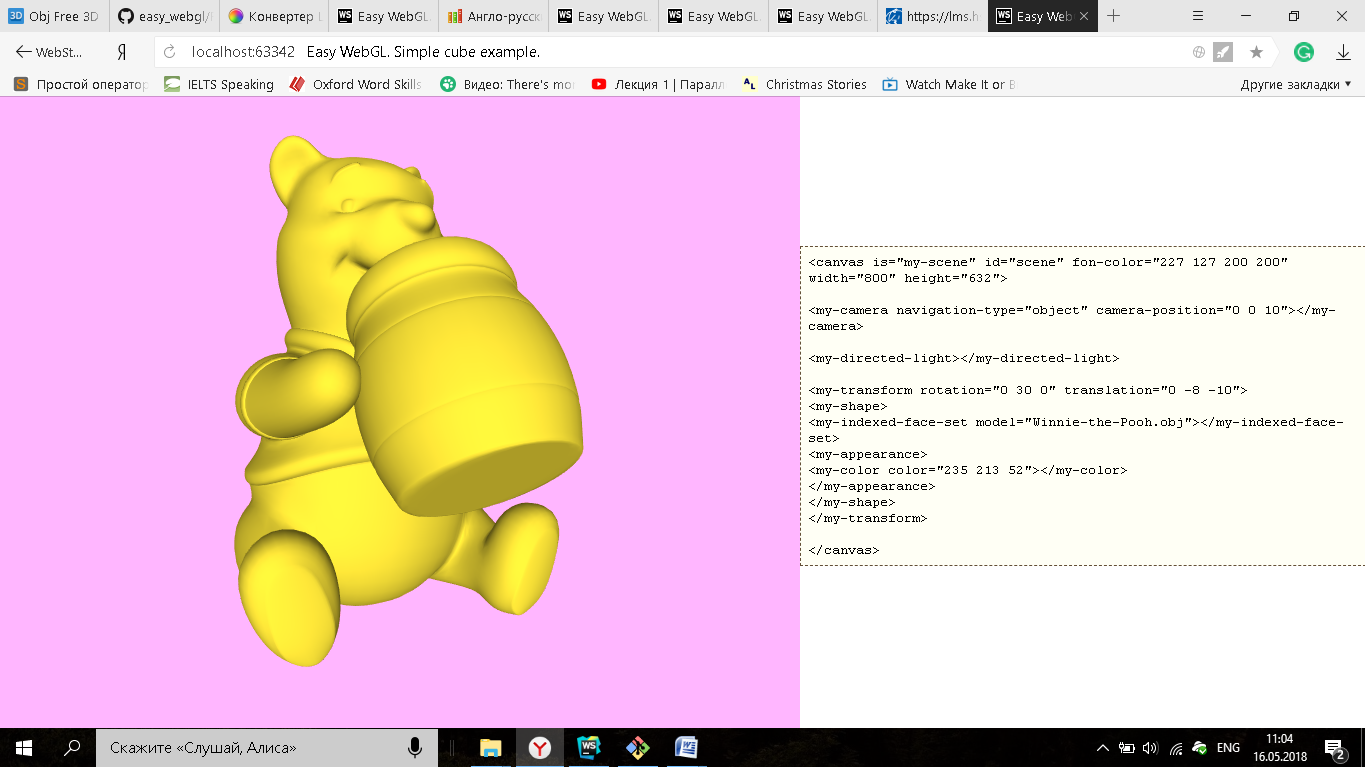
## **Загрузка произвольных моделей:**

Для отрисовки более сложных форм в библиотеке предусмотрен тег my-indexed-face-set. Его использование предполагает наличие у программиста готовой 3D-модели в формате .obj[[2]](#footnote-2), которую он хочет добавить в свой проект.

Пример кода объявления произвольной модели:



Результат:

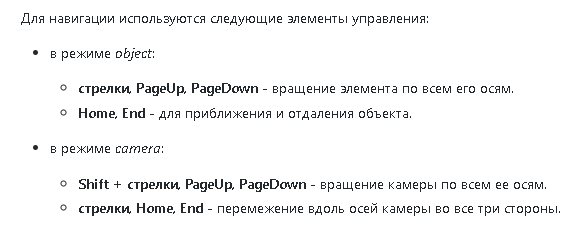


Для конфигурации параметров модели обратитесь к списку атрибутов предыдущего раздела.

## **Выбор типа навигации:**

Тип навигации по трехмерной сцене можно выбрать задав значение атрибуту navigation-type тега my-camera. Отсутствие атрибута равносильно статичной картинке на экране.

Атрибут может принимать одно из двух возможных значений: "object" (всё движение объектов производится вокруг своих осей) или "camera" (так газываемое «брождение» по сцене).



**ПРИЛОЖЕНИЕ 1**

**ТЕРМИНОЛОГИЯ**

*Таблица 2. Терминология*

|  |  |
| --- | --- |
| Термин | Определение |
| Трехмерная сцена | Трехмерная сцена - это часть 3D-мира, подлежащая расчёту и выводу на экран в соответствии с текущей точкой наблюдения. |
| HTML-Тег | HTML-тег (HTML-элемент)- основная структурная единица веб-страницы, написанная на языке HTML. |
| Графический контекст WebGL | Графический контекст - вспомогательный объект для взаимодействия графического приложения, операционной системы и видеокарты. |
| WebGL API | WebGL API - программный интерфейс для отображения трёхмерной графики интернет-браузерами[18] |
| Дочерние теги | Дочерние тэги - тэги, являющиеся прямыми потомками данного элемента в DOM и объявленные внутри данного элемента. |
| Стандарт X3D | X3D — это стандарт ISO, предназначенный для работы с трёхмерной графикой в реальном времени, открытый и не требующий отчислений. В X3D возможно кодировать сцену используя синтаксис XML, равно как и Open Inventor-подобный синтаксис VRML97, а также расширенный интерфейс прикладного программирования.[21] |
| Custom Elements | Custom Elements – это спецификация, описывающая определение пользовательских элементов [14] |
| DOM-элементы | Объекты DOM, соответствующие HTML тегам страницы |
| Canvas | Canvas — элемент HTML, предназначенный для создания растрового изображения при помощи скриптов, обычно на языке JavaScript [15] |
| Плагин | Плаги́н — независимо компилируемый программный модуль, динамически подключаемый к основной программе и предназначенный для расширения и/или использования её возможностей. Плагины обычно выполняются в виде библиотек общего пользования. [22] |
| HTML | HTML (от англ. HyperText Markup Language — «язык гипертекстовой разметки») — стандартизированный язык разметки документов во Всемирной паутине. Большинство веб-страниц содержат описание разметки на языке HTML (или XHTML). Язык HTML интерпретируется браузерами; полученный в результате интерпретации форматированный текст отображается на экране монитора компьютера или мобильного устройства. [15] |
| GLSL | GLSL (OpenGL Shading Language) — язык высокого уровня для программирования шейдеров.[23] |
| Шейдер | Ше́йдер (англ. shader — затеняющая программа) — компьютерная программа, предназначенная для исполнения процессорами видеокарты (GPU).  [24] |
| Вершинный шейдер | Вершинный шейдер оперирует данными, связанными с вершинами многогранников, например, с координатами вершины (точки) в пространстве, с текстурными координатами, с цветом вершины, с вектором касательной, с вектором бинормали, с вектором нормали. Вершинный шейдер может использоваться для видового и перспективного преобразования вершин, для генерации текстурных координат, для расчёта освещения и т. д. |
| Фрагментный шейдер | Пиксельный (Фрагментный) шейдер работает с фрагментами растрового изображения и с текстурами — обрабатывает данные, связанные с пикселями (например, цвет, глубина, текстурные координаты). Пиксельный шейдер используется на последней стадии графического конвейера для формирования фрагмента изображения. |
| дерев DOM-элементов | Дерево DOM элементов – структура объектов, описывающая структуру HTML документа |
| WebWorkers API | Программный интерфейс, позволяющий запускать на WEB-странице фоновые задачи, не влияющие на производительность страницы. [16] |
| OpenGL | OpenGL (Open Graphics Library) — спецификация, определяющая платформо-независимый (независимый от языка программирования) программный интерфейс для написания приложений, использующих двумерную и трёхмерную компьютерную графику.[26] |
| растеризация | Растеризация — это перевод изображения, описанного векторным форматом в пиксели или точки, для вывода на дисплей или принтер. Процесс, обратный векторизации. |
| JS Promise | Объект Promise (обещание) используется для отложенных и асинхронных вычислений. |
| API | API (программный интерфейс приложения, интерфейс прикладного программирования) (англ. application programming interface, API [эй-пи-ай]) — набор готовых классов, процедур, функций, структур и констант, предоставляемых приложением (библиотекой, сервисом) или операционной системой для использования во внешних программных продуктах. Используется программистами при написании всевозможных приложений. [27] |
| DOM | OM (от англ. Document Object Model — «объектная модель документа») — это независящий от платформы и языка программный интерфейс, позволяющий программам и скриптам получить доступ к содержимому HTML-, XHTML- и XML-документов, а также изменять содержимое, структуру и оформление таких документов. [28] |
| Фрустум | Фрустум - часть геометрического тела, заключённая между двумя секущими плоскостями. |

**ПРИЛОЖЕНИЕ 2**

**ОПИСАНИЕ ФОРМАТА ФАЙЛА .OBJ**

1. **Комментарии**

Строки, начинающиеся с решётки(#), — это комментарии.

# Это комментарий

1. **Список вершин**, с координатами (x,y,z[,w]), w является не обязательным и по умолчанию 1.0.

v 0.123 0.234 0.345 1.0

v ...

...

1. **Нормали (x,y,z); нормали могут быть не нормированными.**

vn 0.707 0.000 0.707

vn ...

...

1. **Определения поверхности (сторон)**

f 1 2 3

f 3/1 4/2 5/3

f 6/4/1 3/5/3 7/6/5

f 6//1 3//3 7//5

f ...

...

Определение сторон

Поверхность определяется в списке вершин, текстурных координат и нормалей. Полигоны, такие как квадрат, могут быть определены с помощью более 3 вершин/текстурных координат/нормалей.

* 1. **Вершины**

Строка, начинающаяся с f, представляет собой индекс Поверхности. Каждая поверхность (полигон) может состоять из трех или более вершин.

f v1 v2 v3 v4 ...

Индексация начинается с первого элемента, а не с нулевого, как принято в некоторых языках программирования, также индексация может быть отрицательной. Отрицательный индекс указывает позицию относительно последнего элемента (индекс -1 указывает на последний элемент).

* 1. **Вершины / Текстурные координаты**

Наряду с вершинами могут сохраняться соответствующие индексы текстурных координат.

f v1/vt1 v2/vt2 v3/vt3 v4/vt4 ...

* 1. **Вершины / Текстурные координаты / Нормали**

Также допустимо сохранение соответствующих индексов нормалей.

f v1/vt1/vn1 v2/vt2/vn2 v3/vt3/vn3 v4/vt4/vn4 ...

* 1. **Вершины / / Нормали**

При отсутствии данных о текстурных координатах допустима запись с пропуском индексов текстур.

f v1//vn1 v2//vn2 v3//vn3 v4//vn4 ...

Наличие всех параметров необязательно. При отсутствии какого-либо параметра программа автоматически устанавливает его по умолчанию.[25][29]

Описание файла приведено не полностю. Описываются только части, необходимые для работы библиотеки.

# **ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Лист регистрации изменений | | | | | | | | | |
| Номера листов (страниц) | | | | | Всего листов (страниц в докум.) | № документа | Входящий № сопроводительного докум. и дата | Подп. | Дата |
| Изм. | Измененных | Замененных | Новых | Аннулированх |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |

1. см. Приложение 1. [↑](#footnote-ref-1)
2. см. Приложение 2. [↑](#footnote-ref-2)