

**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ  
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук  
Департамент программной инженерии

**СОГЛАСОВАНО**  
Профессор  
департамента программной инженерии  
кандидат технических наук

**УТВЕРЖДАЮ**  
Академический руководитель  
образовательной программы  
«Программная инженерия»

\_\_\_\_\_ Е.М. Гринкруг  
«\_\_» \_\_\_\_\_ 2018 г.

\_\_\_\_\_ В.В. Шилов  
«\_\_» \_\_\_\_\_ 2018 г.

**РЕАЛИЗАЦИЯ ПОДМНОЖЕСТВА СТАНДАРТА ТРЕХМЕРНОЙ ГРАФИКИ  
СРЕДСТВАМИ БИБЛИОТЕКИ WEBGL**

**Руководство программиста**

**ЛИСТ УТВЕРЖДЕНИЯ**

**RU.17701729.03.05-01 33 01-1-ЛУ**

<i>Подп. и дата</i>	
<i>Инв. № дубл.</i>	
<i>Взам. инв. №</i>	
<i>Подп. и дата</i>	
<i>Инв. № подл</i>	RU.17701729.03.05-01

Исполнитель:  
студентка группы БПИ162  
\_\_\_\_\_ / Казанцева А.Р. /  
«\_\_» \_\_\_\_\_ 2018 г.

**2018**

**УТВЕРЖДЕНО**  
**RU.17701729.03.05-01 33 01-1**

**ПРОГРАММА КОДИРОВАНИЯ И ДЕКОДИРОВАНИЯ  
АЛГЕБРОГЕОМЕТРИЧЕСКИХ КОДОВ**

**Руководство программиста**

**RU.17701729.03.05-01 33 01-1**

**Листов 15**

<i>Инв. № подл</i>	<i>Подп. и дата</i>	<i>Взам. инв. №</i>	<i>Инв. № дубл.</i>	<i>Подп. и дата</i>
RU.17701729.03.05-01				

## СОДЕРЖАНИЕ

СОДЕРЖАНИЕ.....	3
1. Назначение программы.....	4
1.1 Функциональное назначение .....	4
1.2 Эксплуатационное назначение .....	4
2. Условия выполнения программы.....	4
2.1 Минимальный состав технических средств.....	4
2.2 Минимальный состав программных средств .....	4
2.3 Требования к пользователю .....	5
3. Использование библиотеки .....	5
3.3 Состав элементов.....	5
3.4 Примеры использования .....	7
Инициализация сцены: .....	7
Создание прямоугольного параллелепипеда и освещения: .....	8
Изменение параметров: .....	9
Загрузка произвольных моделей: .....	9
Выбор типа навигации: .....	10
ПРИЛОЖЕНИЕ 1 .....	11
ПРИЛОЖЕНИЕ 2 .....	13
ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ.....	15

## 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 сцены<sup>1</sup> в виде тега<sup>1</sup> `<canvas is="my-scene"> </canvas>`. При наличии хотя бы одного тега данного формата библиотека начинает взаимодействовать с графическим контекстом `webgl`<sup>1</sup> этого тега и, опираясь на дочерние теги<sup>1</sup>, описываемые в соответствии со стандартом трехмерной графики X3D[12] и спецификацией данной библиотеки[13], отрисовывает трехмерные объекты, обращаясь к WebGL API<sup>1</sup>.

### 1.2 Эксплуатационное назначение

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

- не используя никаких плагинов<sup>1</sup>;
- декларируя элементы непосредственно в привычном html-коде;
- не углубляясь в низкоуровневую работу с шейдерами<sup>1</sup> и GLSL<sup>1</sup>.

## 2. Условия выполнения программы

### 2.1 Минимальный состав технических средств

- 1) NVIDIA  $\geq$  257.21 или ATI/AMD  $\geq$  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
--------------	--------

<sup>1</sup> см. Приложение 1.

<b>Edge</b>	16 выше
<b>Firefox</b>	59 и выше
<b>Chrome</b>	49 и выше
<b>Safari</b>	TP, 11
<b>Яндекс.Браузер</b>	17 и выше
<b>iOS Safari</b>	10.3, 11.2, 11.3
<b>Chrome Android</b>	66
<b>UC for Android</b>	11.8
<b>Samsung Internet</b>	4, 6.2

2. 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.

### 2.3 Требования к пользователю

Для использования библиотеки не требуется никаких особых знаний.

## 3. Использование библиотеки

### 3.3 Состав элементов

Таблица 2 . Теги и атрибуты

Тег	Атрибуты	Значение атрибутов
<b>my-scene</b> Определяет трехмерную сцену.	fon-color	Значение по умолчанию: "255 255 255 0" Цвет фона трехмерной сцены.
<b>my-camera</b> 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.

<b>my-directed-light</b> 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.
<b>my-transform</b> 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.
<b>my-shape</b> Определяет форму некоторого трехмерного объекта. <i>Обязательно имеет дочерний элемент.</i>	-	-
<b>my-indexed-face-set</b> Определяет произвольную фигуру.	model	<i>Обязательный атрибут.</i> Путь к .obj файлу - модели, которую необходимо отрендерить.
<b>my-box</b> Определяет прямоугольный параллелепипед.	size	Значение по умолчанию: "10 10 10" Размеры параллелепипеда по соответствующим сторонам.
<b>my-appearance</b> Определяет внешний вид некоторого трехмерного объекта. <i>Обязательно имеет дочерний элемент.</i>	-	-
<b>my-color</b>	color	<i>Обязательный атрибут.</i>

Определяет цвет некоторого трехмерного объекта.		Значение по умолчанию: "255 255 255" Rgb-цвет (значения в пределах 0..255)
---	--	---

### 3.4 Примеры использования

#### Инициализация сцены:

Для написания простейшей программы необходимо в коде html-страницы подключить библиотеку `easy_webgl` и определить сцену следующим образом:

```
<canvas is="my-scene"></canvas>
```

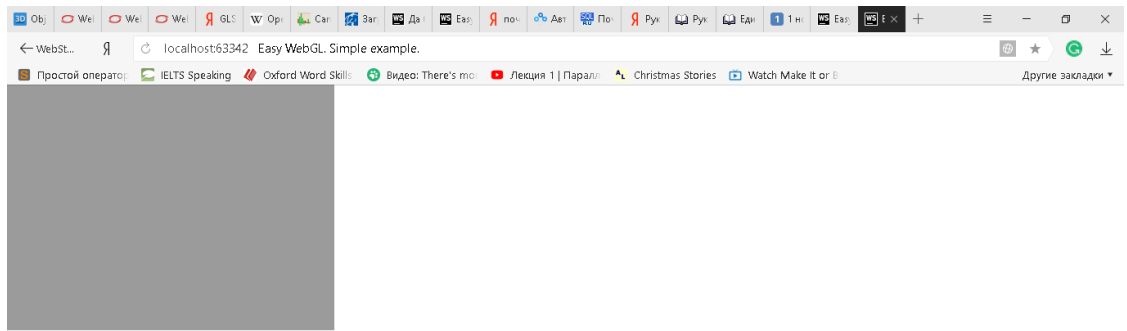
И написать скрипт, инициализирующий эту сцену:

```
<body>

  <canvas is="my-scene" id="scene"></canvas>
  <script type="text/javascript">
    let scene = new Scene(document.getElementById("scene"));
  </script>
</body>
```

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.

```
1  <!DOCTYPE html>
2  <html lang="ru">
3  <head>
4      <meta charset="UTF-8">
5      <title>Easy WebGL. Simple example.</title>
6      <script src="../src/easy_webgl.js"></script>
7      <style>
8          body {
9              margin: 0;
10         }
11
12         canvas {
13             width: 400px;
14             height: 300px;
15             display: block;
16         }
17     </style>
18 </head>
19 <body>
20
21     <canvas is="my-scene" id="scene" fon-color="100 100 100 200">
22         <my-camera></my-camera>
23     </canvas>
24     <script type="text/javascript">
25         let scene = new Scene(document.getElementById("scene"));
26     </script>
27
28 </body>
29 </html>
```



### 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.

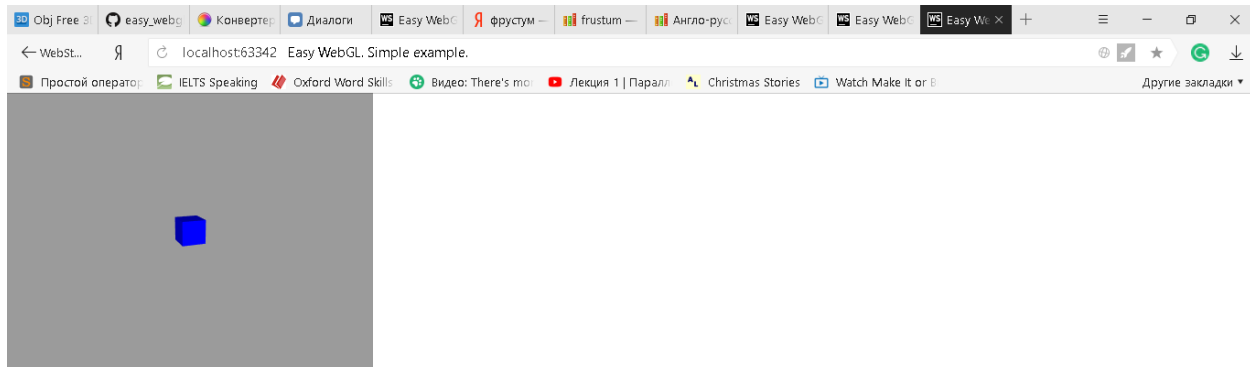
Итак, для создания простейшего кубика определим следующие элементы:

```
<canvas is="my-scene" id="scene" fog-color="100 100 100 200">
  <my-camera></my-camera>
  <my-directed-light></my-directed-light>
  <my-transform translation="0 0 -100" rotation="15 20 0">
    <my-shape>
      <my-box></my-box>
      <my-appearance>
        <my-color color="0 0 255"></my-color>
      </my-appearance>
    </my-shape>
  </my-transform>
</canvas>
```

Заметим, что у камеры, трансформы и цвета есть атрибуты. Атрибут для цвета обязателен, для трансформы 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. Оставляя 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:

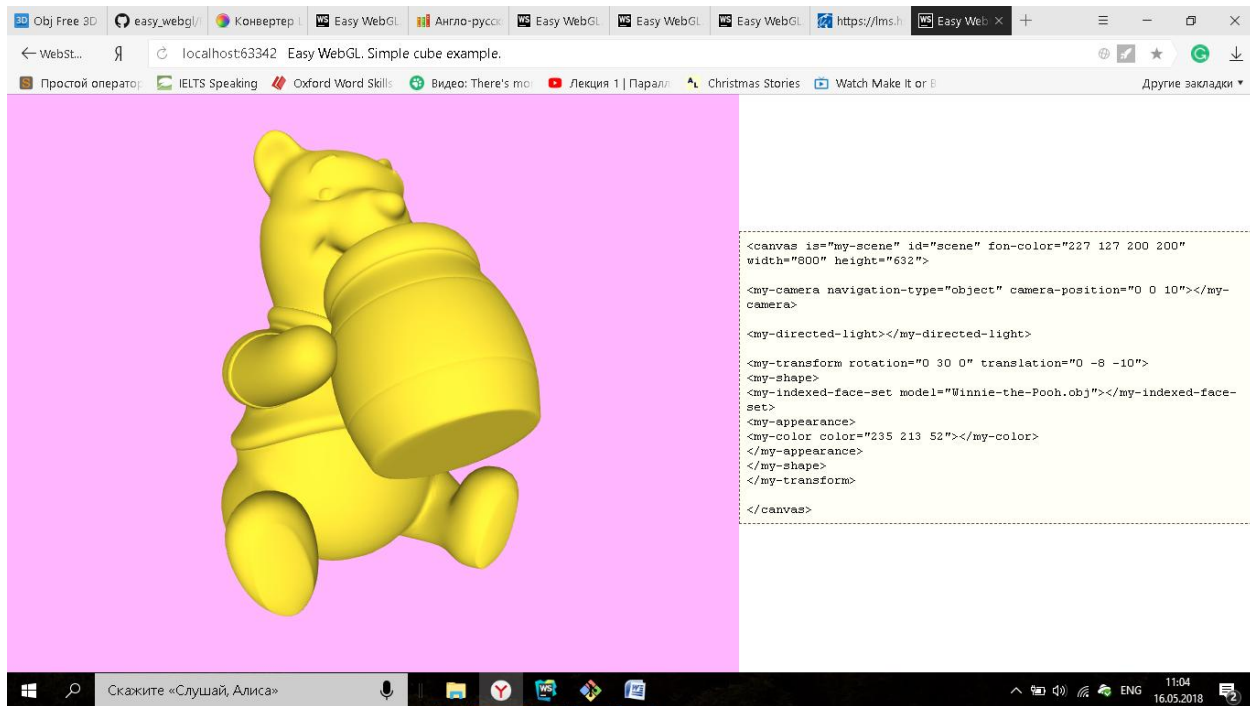
Для отрисовки более сложных форм в библиотеке предусмотрен тег my-indexed-face-set. Его использование предполагает наличие у программиста готовой 3D-модели в формате .obj<sup>1</sup>, которую он хочет добавить в свой проект.

Пример кода объявления произвольной модели:

```
<my-transform rotation="0 30 0" translation="0 -8 -10">
  <my-shape>
    <my-indexed-face-set model="Winnie-the-Pooh.obj"></my-indexed-face-set>
    <my-appearance>
      <my-color color="235 213 52"></my-color>
    </my-appearance>
  </my-shape>
</my-transform>
```

Результат:

<sup>1</sup> см. Приложение 2.



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 называемое «брождение» по сцене).

Для навигации используются следующие элементы управления:

- в режиме *object*:
  - **стрелки, PageUp, PageDown** - вращение элемента по всем его осям.
  - **Home, End** - для приближения и отдаления объекта.
- в режиме *camera*:
  - **Shift + стрелки, PageUp, PageDown** - вращение камеры по всем ее осям.
  - **стрелки, Home, End** - перемещение вдоль осей камеры во все три стороны.

**ПРИЛОЖЕНИЕ 1**

**ТЕРМИНОЛОГИЯ**

*Таблица 3. Терминология*

<b>Термин</b>	<b>Определение</b>
<b>Трёхмерная сцена</b>	Трёхмерная сцена - это часть 3D-мира, подлежащая расчёту и выводу на экран в соответствии с текущей точкой наблюдения.
<b>HTML-Тег</b>	HTML-тег (HTML-элемент)- основная структурная единица веб-страницы, написанная на языке HTML.
<b>Графический контекст WebGL</b>	Графический контекст - вспомогательный объект для взаимодействия графического приложения, операционной системы и видеокарты.
<b>WebGL API</b>	WebGL API - программный интерфейс для отображения трёхмерной графики интернет-браузерами[18]
<b>Дочерние теги</b>	Дочерние тэги - тэги, являющиеся прямыми потомками данного элемента в DOM и объявленные внутри данного элемента.
<b>Стандарт X3D</b>	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]
<b>Custom Elements</b>	Custom Elements – это спецификация, описывающая определение пользовательских элементов [14]
<b>DOM-элементы</b>	Объекты DOM, соответствующие HTML тегам страницы
<b>Canvas</b>	Canvas — элемент HTML, предназначенный для создания растрового изображения при помощи скриптов, обычно на языке JavaScript [15]
<b>Плагин</b>	Плагин — 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]
<b>HTML</b>	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]
<b>GLSL</b>	GLSL (OpenGL Shading Language) — язык высокого уровня для программирования шейдеров.[23]
<b>Шейдер</b>	Шейдер (англ. shader — затеняющая программа) — компьютерная программа, предназначенная для исполнения процессорами видеокарты (GPU). [24]
<b>Вершинный шейдер</b>	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.
<b>Фрагментный шейдер</b>	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.
<b>дерево DOM-элементов</b>	Дерево DOM элементов – структура объектов, описывающая структуру HTML документа
<b>WebWorkers API</b>	Программный интерфейс, позволяющий запускать на WEB-странице фоновые задачи, не влияющие на производительность страницы. [16]
<b>OpenGL</b>	OpenGL (Open Graphics Library) — спецификация, определяющая платформу-независимый (независимый от языка программирования) программный интерфейс для написания приложений, использующих двумерную и трёхмерную компьютерную графику.[26]
<b>растеризация</b>	Растеризация — это перевод изображения, описанного векторным форматом в пиксели или точки, для вывода на дисплей или принтер. Процесс, обратный векторизации.

<b>JS Promise</b>	Объект Promise (обещание) используется для отложенных и асинхронных вычислений.
<b>API</b>	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]
<b>DOM</b>	ОМ (от англ. Document Object Model — «объектная модель документа») — это независимый 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]
<b>Фрустум</b>	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

#### 2. 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 ...

...

#### 3. Нормали (x,y,z); нормали могут быть не нормированными.

vn 0.707 0.000 0.707

vn ...

...

#### 4. 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.

#### 4.1. 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).

#### 4.2. Вершины / Текстурные координаты

Наряду с вершинами могут сохраняться соответствующие индексы текстурных координат.

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

#### 4.3. Вершины / Текстурные координаты / Нормали

Также допустимо сохранение соответствующих индексов нормалей.

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

#### 4.4. 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 полностью. Описываются только части, необходимые для работы библиотеки.

## ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

[illegible]

[illegible]