

**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**

Пояснительная записка

ЛИСТ УТВЕРЖДЕНИЯ

RU.17701729.03.05-01 81 01-1-ЛУ

Подп. и дата	
Инв. № дубл.	
Взам. инв. №	
Подп. и дата	
Инв. № подл	RU.17701729.501610-01

Исполнитель:
студентка группы БПИ162
_____/Казанцева Анастасия Романовна /
« ____ » _____ 2018 г.

УТВЕРЖДЕНО
RU.17701729.03.05-01 81 01-1-ЛЮ

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

Пояснительная записка

RU.17701729.03.05-01 81 01-1

Листов 33

<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

1. ВВЕДЕНИЕ	5
1.1. Наименование библиотеки	5
1.2. Документы, на основании которых ведется разработка.....	5
2. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ.....	6
2.1. Назначение библиотеки	6
2.1.1. Функциональное назначение.....	6
2.1.2. Эксплуатационное назначение	6
2.2. Краткая характеристика области применения	6
3. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ	7
3.1. Постановка задачи на разработку библиотеки.....	7
3.2. Описание функционирования библиотеки.....	7
3.3. Описание и обоснование выбора использованных при разработке технологий	8
3.3.1. OBJ парсер.....	8
3.3.2. WebGL 1.0	8
3.3.3. JS Promise.....	9
3.3.4. WebWorkers API.....	9
3.4. Описание и обоснование выбора метода организации входных и выходных данных.....	10
3.5. Описание и обоснование выбора состава технических и программных средств	10
3.5.1. Состав технических и программных средств.....	10
3.5.2. Обоснование выбора технических и программных средств	11
3.6. Предполагаемая потребность	12
3.7. Экономические преимущества разработки по сравнению с отечественными и зарубежными образцами или аналогами	12
4. ИСТОЧНИКИ, ИСПОЛЬЗОВАННЫЕ ПРИ РАЗРАБОТКЕ	13
ПРИЛОЖЕНИЕ 1	16
ПРИЛОЖЕНИЕ 2	19
ПРИЛОЖЕНИЕ 3	21
ПРИЛОЖЕНИЕ 4	22
ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ	34

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.03.05-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

АННОТАЦИЯ

В данном программном документе приведена пояснительная записка к курсовой работе на тему Реализация подмножества стандарта трёхмерной графики средствами библиотеки WebGL», в результате которой разработана библиотека, предназначенная для упрощения работы с трёхмерной графикой в веб-разработке.

В разделе «Введение» указано наименование программы, краткое наименование программы и документы, на основании которых ведется разработка.

В разделе «Назначение и область применения» указано функциональное назначение программы, эксплуатационное назначение программы и краткая характеристика области применения программы.

В разделе «Технические характеристики» содержатся следующие подразделы: – постановка задачи на разработку программы; – описание алгоритма и функционирования программы с обоснованием выбора схемы алгоритма решения задачи и возможные взаимодействия программы с другими программами; – описание и обоснование выбора метода организации входных и выходных данных; – описание и обоснование выбора состава технических и программных средств.

В разделе «Ожидаемые технико-экономические показатели» указана предполагаемая потребность и экономические преимущества разработки по сравнению с отечественными и зарубежными образцами или аналогами

Настоящий документ разработан в соответствии с требованиями:

- 1) ГОСТ 19.101-77 Виды программ и программных документов [1];
- 2) ГОСТ 19.102-77 Стадии разработки [2];
- 3) ГОСТ 19.103-77 Обозначения программ и программных документов [3];
- 4) ГОСТ 19.104-78 Основные надписи [4];
- 5) ГОСТ 19.105-78 Общие требования к программным документам [5];
- 6) ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом [6];
- 7) ГОСТ 19.404-79 Пояснительная записка. Требования к содержанию и оформлению [7].

Изменения к Пояснительной записке оформляются согласно ГОСТ 19.603-78 [8], ГОСТ 19.604-78 [9].

Перед прочтением данного документа рекомендуется ознакомиться с терминологией, приведенной в Приложении 1 настоящей пояснительной записки.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.03.05-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

1. ВВЕДЕНИЕ

1.1. Наименование библиотеки

Наименование библиотеки: «easy_webgl».

Условное обозначение темы разработки: «Реализация подмножества стандарта трехмерной графики средствами библиотеки WebGL».

1.2. Документы, на основании которых ведется разработка

Приказ декана факультета компьютерных наук Национального исследовательского университета «Высшая школа экономики» № 2.3-02/1212-01 от 12.12.2017 "Об утверждении тем, руководителей курсовых работ студентов образовательной программы Программная инженерия факультета компьютерных наук".

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.03.05-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

2. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ

2.1. Назначение библиотеки

2.1.1. 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 `webgl`¹ 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.

2.1.2. Эксплуатационное назначение

Подключенная к проекту библиотека определённые библиотекой (нестандартизованные) DOM-элементы¹, 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.

2.2. Краткая характеристика области применения

Библиотека «easy_webgl», реализующая подмножество стандарта трехмерной графики X3D[12], - это системный программный продукт, упрощающий работу веб-разработчика с трехмерной графикой.

¹ см. Приложение 1.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.03.05-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

3. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

3.1. Постановка задачи на разработку библиотеки

Разрабатываемая библиотека должна:

- 1) Декларированные в html-коде теги ассоциировать с трехмерными объектами в соответствии со спецификацией[13];
- 2) Подчиняться стандарту трехмерной графики X3D[12], реализуя его подмножество.

3.2. Описание функционирования библиотеки

При запуске проекта, к которому подключена «easy_webgl», библиотека навигируясь по дереву DOM-элементов¹ ищет теги canvas с атрибутом¹ is в значении "my-scene". Согласно спецификации Custom Elements[14] именно такая формулировка html-тега позволяет создать собственный тег (DOM-элемент), унаследованный от тега (DOM-элемента), уже имеющегося в спецификации HTML[15].

При нахождении элементов с заданными параметрами библиотека создает объект класса Scene², проверяет структуру выстраивания элементов, дочерних данному, и в соответствии с их декларацией (положением в коде и атрибутивным наполнением) формирует в памяти массив вершин, массив цветов и некоторые другие вспомогательные данные.

При столкновении с тегом my-transform, запускается функция init³ класса Transform². Эта функция задает цепочку асинхронных вызовов, использующую технологию Promise, запускает эту цепочку для инициализации трехмерной сцены данными. Такой способ инициализации позволяет разделить обработку большого фрагмента кода на несколько небольших промежутков времени, чтобы программа не вызывала зависания. Особенно тяжелая операция, такая как парсинг файла формата .obj⁴ (т.е. преобразование данных из файла в нужный библиотеке формат), выводится в

¹ см. Приложение 1.

² см. Приложение 3.

³ см. Приложение 4.

⁴ см. Приложение 2.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.03.05-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

отдельный поток при помощи WebWorkers API¹[16]. В случае, когда браузер не поддерживает WebWorkers API, загрузка будет производиться асинхронно.

После получения всей необходимой информации библиотека отрисовывает трехмерную сцену, добавляя в нее объекты по мере их загрузки.

3.3. Описание и обоснование выбора использованных при разработке технологий

3.3.1. OBJ парсер

Парсер файлов формата .obj является частью библиотеки. Он делит строковое представление файла на строки по символу переноса строки ‘\n’ и, в соответствии с форматом .obj файла², преобразует данные из него в массивы нормалей, вершин и индексов, необходимых для передачи в качестве шейдерного атрибута¹ в вершинный шейдер¹.

3.3.2. WebGL 1.0

WebGL (Web-based Graphics Library) — программная библиотека для языка программирования JavaScript, позволяющая создавать на JavaScript интерактивную 3D-графику, функционирующую в широком спектре совместимых с ней веб-браузеров. За счёт использования низкоуровневых средств поддержки OpenGL¹, часть кода на WebGL может выполняться непосредственно на видеокартах. WebGL — это контекст элемента canvas HTML, который обеспечивает API 3D графики без использования плагинов[17].

Использована спецификация версии 1.0, выпущенная 3 марта 2011 года и использующая в основе OpenGL ES 2.0.[18]

WebGL 1.0 API – предоставляет широкий доступ к взаимодействию с видеокартой. Основной опорной точкой WebGL API является шейдерная программа, работающая как средство растеризации¹ непосредственно на видеокарте. Шейдерная программа в своем минимальном варианте состоит из двух шейдеров – вершинного и фрагментного. Оба шейдера написаны на GLSL (OpenGL Shading Language). Основной задачей программиста является правильно написать шейдеры, загрузить в них нужную информацию и связать DOM-элемент Canvas со скомпилированной шейдерной программой.

¹ см. Приложение 1.

² см. Приложение 2.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.03.05-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Для передачи информации в шейдеры в нашей библиотеке используются три основных способа[19]:

1. Атрибуты и буферы

Буфер – это массив данных последовательного доступа. Атрибут – переменная определяющая какого рода данные хранятся в массиве-буфере. Данные из атрибутов и буферов поступают в вершинный шейдер. Шейдерная программа выполняется отдельно для каждого элемента буфера.

2. Uniform-переменные

Переменные, передающиеся в вершинный шейдер и используемые на каждой итерации прохода по буферу.

3. Varying-переменные

Средства связи между фрагментного буфера с вершинным.

Использована именно технология WebGL в связи с рядом преимуществ:

- не требует никаких дополнительных плагинов;
- выполняется непосредственно на видеокарте;
- имеет стандартизованное API.

3.3.3. JS Promise

Новейшее средство асинхронного программирования в JavaScript, введенное в стандарте ES6[21]. Библиотека производит вычисления, которые в зависимости от потребностей программиста, могут занимать и достаточно много времени, поэтому без асинхронности обойтись нельзя.

Использована именно технология Promise в связи с рядом преимуществ:

1. более удобное API (по сравнению с XMLHttpRequest);
2. хорошая поддержка в браузерах¹.

3.3.4. WebWorkers API

¹ Вывод о распространенности сделан в соответствии с информацией сайта <https://caniuse.com/>

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.03.05-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Единственное средство распараллеливания задач в JavaScript. Предоставляет доступ к средствам браузера для создания отдельного потока и запуска автономной программы на JavaScript. Основной код страницы обменивается информацией с кодом запущенным в WebWorker благодаря событийному механизму.

Парсинг больших файлов .obj подразумевает достаточно громоздкие и затратные по времени вычисления. Это обславливает использование данной технологии в проекте. Дополнительным плюсом является распространенность поддержки технологии WebWorker¹.

3.4. Описание и обоснование выбора метода организации входных и выходных данных

Основными входными данными для библиотеки служат определённые библиотекой (нестандартизованные) html-элементы, на основании которых и происходит отрисовка трехмерных объектов. Данный формат был выбран с целью сделать использование библиотеки наиболее интуитивным и наглядным. Обработка данных этого формата ведется в соответствии со спецификацией Custom Elements[14].

Также в качестве входных данных используются файлы формата .obj, описывающие необходимую для отрисовки геометрии. Выбор данного формата файла связан с простотой во внутренней структуре и распространенностью, что удобно для пользователя. Информация из файлов формата .obj получается путём парсинга через класс OBJLoader².

3.5. Описание и обоснование выбора состава технических и программных средств

3.5.1. Состав технических и программных средств

Для работы библиотеки необходим следующий состав программных средств³:

- 1) один из следующих браузеров:

¹ Вывод о распространенности сделан в соответствии с информацией сайта <https://caniuse.com/>

² см. Приложение 4.

³ Выбор параметров программных средств производится на основании данных о поддержке используемых библиотекой технологий с использованием сайта <https://caniuse.com/>

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.03.05-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

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

- 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.

Для работы библиотеки необходим следующий состав технических средств¹:

- 1) NVIDIA \geq 257.21 или ATI/AMD \geq 10.6 или Intel driver версии от сентября 2010.

3.5.2. Обоснование выбора технических и программных средств

При реализации библиотеки использованы следующие технологии WebGL API 1.0, WebWorkers, Custom Element, JS Promise и другие особенности языка JavaScript стандарта ES6[21].

Требования к техническим и программным средствам представлены в соответствии с поддержкой используемых технологий.

¹ Выбор параметров технических средств производится на основании данных о поддержке используемых библиотекой технологий с использованием источника 20.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.03.05-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

ОЖИДАЕМЫЕ ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ

В рамках данной работы расчет экономической эффективности не предусмотрен.

3.6. Предполагаемая потребность

Данная библиотека будет иметь спрос среди веб-разработчиков, желающих, не имея глубокого понимания основ трехмерной графики, использовать трехмерную графику в своих проектах декларативно, т.е. непосредственно в html-коде веб-страницы.

3.7. Экономические преимущества разработки по сравнению с отечественными и зарубежными образцами или аналогами

Данная библиотека:

- 1) распространяется бесплатно;
- 2) не требует вложения денежных средств во время использования;
- 3) имеет неограниченный срок службы;
- 4) опирается на стандарт трехмерной графики X3D[12].

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.03.05-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

4. ИСТОЧНИКИ, ИСПОЛЬЗОВАННЫЕ ПРИ РАЗРАБОТКЕ

1. ГОСТ 19.101-77 Виды программ и программных документов. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
2. ГОСТ 19.102-77 Стадии разработки. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
3. ГОСТ 19.103-77 Обозначения программ и программных документов. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
4. ГОСТ 19.104-78 Основные надписи. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
5. ГОСТ 19.105-78 Общие требования к программным документам. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
6. ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
7. ГОСТ 19.201-78 Техническое задание. Требования к содержанию и оформлению. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
8. ГОСТ 19.603-78 Общие правила внесения изменений. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
9. ГОСТ 19.604-78 Правила внесения изменений в программные документы, выполненные печатным способом. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
10. ГОСТ 15150-69 Машины, приборы и другие технические изделия. Исполнения для различных климатических районов. Категории, условия эксплуатации, хранения и транспортирования в части воздействия климатических факторов внешней среды. – М.: Изд-во стандартов, 1997.
11. ГОСТ 19.301-79 Программа и методика испытаний. Требования к содержанию и оформлению. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.03.05-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

12. Don Brutzman, Leonard Daly X3D: Extensible 3D Graphics for Web Authors [Книга]. - [441 с.] : Elsevier Inc., 2007.
13. «Реализация подмножества стандарта трехмерной графики средствами библиотеки WebGL» Руководство программиста, 2018.
14. Custom Elements [Электронный ресурс]: W3C Working Group Note / W3C — MIT/CSAIL — США, НИУ ВШЭ - Россия, 1994 — Режим доступа: <https://www.w3.org/TR/custom-elements/>. (дата обращения: 15.05.18)
15. HTML [Электронный ресурс]: W3C Recommendation / W3C — MIT/CSAIL — США, НИУ ВШЭ - Россия, 1994 — Режим доступа: <https://www.w3.org/TR/html5/>. (дата обращения: 14.12.17)
16. Web worker // Wikipedia, The Free Encyclopedia. [2018—2018]. Дата обновления: 13.05.2018. URL: <https://en.wikipedia.org/?oldid=840991490> (дата обращения: 13.05.2018).
17. WebGL Overview [Электронный ресурс] // Khronos Group. [2018-2018]. URL: <https://www.khronos.org/webgl/> (дата обращения: 10.05.2018).
18. WebGL Specification [Электронный ресурс] // Khronos Group. [2017-2018]. URL: <https://www.khronos.org/registry/webgl/specs/latest/1.0/> (дата обращения: 18.12.2017).
19. BlacklistsAndWhitelists [Электронный ресурс] // WebGL Public Wiki. [2017-2018]. URL: <https://www.khronos.org/webgl/wiki/BlacklistsAndWhitelists> (дата обращения: 03.06.2017).
20. ECMAScript 2015 Language Specification [Электронный ресурс] // Ecma International [2015-2015] URL: <https://www.ecma-international.org/ecma-262/6.0/> (дата обращения: 10.05.2018).
21. X3D [Электронный ресурс] // Википедия. [2015—2015]. Дата обновления: 23.11.2015. URL: <https://ru.wikipedia.org/?oldid=74708271> (дата обращения: 23.11.2015).
22. Плагин [Электронный ресурс] // Википедия. [2018—2018]. Дата обновления: 27.03.2018. URL: <https://ru.wikipedia.org/?oldid=91740638> (дата обращения: 27.03.2018).
23. OpenGL Shading Language [Электронный ресурс] // Википедия. [2018—2018]. Дата обновления: 14.05.2018. URL: <https://ru.wikipedia.org/?oldid=92660665> (дата обращения: 14.05.2018).

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.03.05-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

24. Шейдер [Электронный ресурс] // Википедия. [2018—2018]. Дата обновления: 25.02.2018. URL: <https://ru.wikipedia.org/?oldid=91160905> (дата обращения: 25.02.2018).
25. Object Files (.obj)[Электронный ресурс] // Martin Reddy. URL: <http://www.martinreddy.net/gfx/3d/OBJ.spec> (дата обращения: 14.05.2018).
26. OpenGL [Электронный ресурс] // Википедия. [2018—2018]. Дата обновления: 14.05.2018. URL: <https://ru.wikipedia.org/?oldid=92660655> (дата обращения: 14.05.2018).
27. API [Электронный ресурс] // Википедия. [2018—2018]. Дата обновления: 13.03.2018. URL: <https://ru.wikipedia.org/?oldid=91492448> (дата обращения: 13.03.2018).
28. Document Object Model [Электронный ресурс] // Википедия. [2018—2018]. Дата обновления: 10.05.2018. URL: <https://ru.wikipedia.org/?oldid=92566096> (дата обращения: 10.05.2018).
29. Obj [Электронный ресурс] // Википедия. [2018—2018]. Дата обновления: 14.05.2018. URL: <https://ru.wikipedia.org/?oldid=92658282> (дата обращения: 14.05.2018).

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.03.05-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

ПРИЛОЖЕНИЕ 1

ТЕРМИНОЛОГИЯ

Таблица 2. Терминология

Термин	Определение
Трёхмерная сцена	Трёхмерная 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	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]
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

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.03.05-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

	текст отображается на экране монитора компьютера или мобильного устройства. [15]
GLSL	GLSL (OpenGL Shading Language) — язык высокого уровня для программирования шейдеров.[23]
Шейдер	Шейдер (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.
дерево 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 платформу-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

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.03.05-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

	использования во внешних программных продуктах. Используется программистами при написании всевозможных приложений. [27]
DOM	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]
Фрустум	Фрустум - часть геометрического тела, заключённая между двумя секущими плоскостями.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.03.05-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

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 ...

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.03.05-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

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.

$f\ v1/vt1\ v2/vt2\ v3/vt3\ v4/vt4\ \dots$

4.3. Вершины / Текстурные координаты / Нормали

Также допустимо сохранение соответствующих индексов нормалей.

$f\ v1/vt1/vn1\ v2/vt2/vn2\ v3/vt3/vn3\ v4/vt4/vn4\ \dots$

4.4. Вершины / / Нормали

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

$f\ v1//vn1\ v2//vn2\ v3//vn3\ v4//vn4\ \dots$

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.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.03.05-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

ПРИЛОЖЕНИЕ 3

ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ КЛАССОВ

Описание и функциональное назначение классов.

Таблица 3

Класс	Назначение
Scene	Определяет трехмерную сцену.
Camera	Определяет камеру в нашей сцене.
DirectedLight	Определяет направленный свет для трехмерной сцены.
Transform	Определяет положение геометрии в пространстве сцены.
Shape	Определяет геометрию.
Appearance	Определяет внешний вид фигуры.
Color	Определяет цвет однотонной фигуры.
Box	Определяет форму прямоугольного параллелепипеда.
IndexedFaceSet	Определяет произвольную форму на сцене.
Utils	Класс вспомогательных функций.
Algebra	Класс вспомогательных алгебраических функций, совершающий операции над матрицами 4*4 и векторами 3.
OBJLoader	Парсер файлов формата .obj.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.03.05-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

ПРИЛОЖЕНИЕ 4

ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ ПОЛЕЙ, МЕТОДОВ И СВОЙСТВ

Таблица 4. Поля и методы класса Scene.

ПОЛЯ		
Наименование	Тип	Назначение
fragmentShader	WebGLShader	Ссылка на скомпилированный фрагментный шейдер на GLSL.
vertexShader	WebGLShader	Ссылка на скомпилированный вершинный шейдер на GLSL.
program	WebGLProgram	Скомпилированная и прилинкованная шейдерная программа на GLSL.
sceneElement	HTMLElement	Ссылка на DOM-элемент, который иллюстрирует этот объект.
gl	WebGLRenderingContext	Графический контекст.
color	number[]	Фоновый цвет сцены.
objects	Array	Все объекты сцены.
cameras	Camera[]	Все камеры сцены.
activeCamera	Camera	Активная камера.
light	DirectedLight	Направленный свет.
fragmentShaderSource	string	Текст фрагментного шейдера на GLSL.
vertexShaderSource	string	Текст вершинного шейдера на GLSL.
МЕТОДЫ		
Наименование	Модификатор	Описание
-	constructor	Создает экземпляр Scene. * @param sceneElement Ссылка на DOM-элемент, который иллюстрирует этот объект.
registerAll	-	Регистрация всех кастомных HTML-элементов.
init	-	Инициализирует данные, используя атрибуты тега my-scene. Проверяет наличие обязательных тегов. Запускает инициализацию дочерних элементов.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.03.05-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

resize	-	Подгоняет размер canvas под экран
addObject	-	<p>Добавление трехмерного объекта на сцену.</p> <ul style="list-style-type: none"> * @param transform {Transform} Положение в пространстве добавляемого объекта. * @param shape {Shape} Форма добавляемого объекта. * @param appearance {Appearance} Внешний вид добавляемого объекта.
drawScene	-	Отрисовывает сцену.
createShader	-	<p>Создание и компиляция шейдера.</p> <ul style="list-style-type: none"> * @param type Тип шейдера. * @param source {String} Код шейдера. * @returns {WebGLShader} Шейдер.
createProgram	-	<p>Собираем программу для видеокарты из двух шейдеров.</p> <ul style="list-style-type: none"> * @param vertexShader {WebGLShader} Вершинный шейдер. * @param fragmentShader {WebGLShader} Фрагментный шейдер. * @returns {WebGLProgram} Программа для видеокарты.
fillAttribute	-	<p>Заполняет атрибут и буфер данными.</p> <ul style="list-style-type: none"> * @param program {WebGLProgram} Текущая программа для видеокарты. * @param name {String} Наименование атрибута в шейдере. * @param data Массив данных для передачи, приведенный к нужному типу. * @param size {Number} Количество компонент массива на итерацию. * @param type Тип массива данных. * @param normalize {Boolean} Нуждаются ли данные в нормализации. * @param stride {Number} Дополнительное перемещение по данным относительно итерации. * @param buf_offset {Number} Элемент массива, с которого начнется первая итерация.
fillIndexAttribute	-	Заполняет атрибут индексов.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.03.05-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

		* @param indices {number[]} Данные для передачи
--	--	---

Таблица 5. Поля и методы класса Scene.

ПОЛЯ		
Наименование	Тип	Назначение
cameraElement	HTMLElement	Ссылка на DOM-элемент, который иллюстрирует этот объект.
scene	Scene	Ссылка на трехмерную сцену, которой принадлежит камера.
aspect	number	Отношение ширины окна отрисовки к высоте.
fieldOfViewDegrees	number	Угол отображения области видимости в радианах.
zFar	number	Модуль самого дальнего индекса, видимого камере, по Z.
zNear	number	Модуль ближайшего индекса, видимого камере, по Z.
cameraRotation	number[]	Углы поворота камеры в градусах.
cameraPosition	number[]	Позиция камеры в мире.
navigationType	string	Тип навигации по трехмерной сцене.
cameraMatrix	number[]	Матрица камеры.
projectionMatrix	number[]	Матрица, формирующая перспективу.
МЕТОДЫ		
Наименование	Модификатор	Описание
-	constructor	Создает экземпляр Camera. * @param cameraElement Ссылка на DOM-элемент, который иллюстрирует этот объект. * @param scene {Scene} Трехмерная сцена, в которой определен объект.
init	-	Инициализирует данные для шейдеров, используя атрибуты тега my-camera. Определяет параметры фрустума: * атрибут "view-angle" - число - угол фрустума

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.03.05-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

		<p>в градусах - по умолчанию 60;</p> <p>* атрибут "z-far" - число - максимальное видимое значение координат по оси z - по умолчанию 2000;</p> <p>* атрибут "z-near" - число - минимальное видимое значение координат по оси z - по умолчанию 1.</p> <p>И параметры камеры:</p> <p>* атрибут "camera-rotation" - массив - углы вращения по осям x,y,z для камеры - по умолчанию "0 0 0";</p> <p>* атрибут "camera-position" - массив - начальная позиция камеры - по умолчанию "0 0 0".</p> <p>* А так же "navigation-type", принимающий одно из двух значений: object или camera. По умолчанию навигация по сцене не производится.</p>
setCameraParams	-	Пересчитывает матрицу камеры в случае изменения параметров.
defineNavigation Type	-	По атрибуту navigation-type тега my-camera задает тип навигации по трехмерной сцене.

Таблица 6. Поля и методы класса DirectedLight.

ПОЛЯ		
Наименование	Тип	Назначение
directedLightElement	HTMLElement	Ссылка на DOM-элемент, который иллюстрирует этот объект.
scene	Scene	Трехмерная сцена, в которой определен объект.
fonLightColor	number[]	Цвет фонового освещения.
directedLightColor	number[]	Цвет направленного освещения.
lightDirection	number[]	Направление света.
МЕТОДЫ		
Наименование	Модификатор	Описание
-	constructor	<p>Создает экземпляр DirectedLight.</p> <p>* @param directedLightElement Ссылка на DOM-</p>

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.03.05-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

		элемент, который иллюстрирует этот объект. * @param scene {Scene} Трехмерная сцена, в которой определен объект.
init	-	Инициализирует объект, используя данные атрибутов. Если есть устанавливает параметры этого освещения: * атрибут "fon-light-color" - массив - rgb-цвет фонового освещения (значения в пределах 0..255) - по умолчанию "100 100 100"; * атрибут "directed-light-color" - массив - rgb-цвет направленного освещения(значения в пределах 0..255) - по умолчанию "200 200 200"; * атрибут "direction" - массив - направление направленного освещения - по умолчанию "0, 0, -1".

Таблица 7. Поля и методы класса Transform.

ПОЛЯ		
Наименование	Тип	Назначение
transformElement	HTMLElement	Ссылка на DOM-элемент, который иллюстрирует этот объект.
scene	Scene	Трехмерная сцена, в которой определен объект.
translation	number[]	Перенос оюекта относительно начала координат сцены.
rotation	number[]	Поворот объекта относительно его осей X, Y, Z.
scale	number[]	Масштабирование объекта относительно его осей X, Y, Z.
МЕТОДЫ		
Наименование	Модификатор	Описание
-	constructor	Создает экземпляр Transform. * @param transformElement Ссылка на DOM-элемент, который иллюстрирует этот объект. * @param scene {Scene} Трехмерная сцена, в

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.03.05-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

		<p>которой определен объект.</p> <p>* @param info Данные о дополнительном смещении, если есть родительский Transform.</p>
init	-	<p>Инициализирует данные, используя атрибуты тега my-transform.</p> <p>Атрибуты:</p> <p>* translation - массив - определяет смещение относительно центра сцены - по умолчанию "0 0 0"</p> <p>* rotation - массив - определяет поворот относительно осей сцены - по умолчанию "0 0 0"</p> <p>* scale - массив - определяет масштаб объекта - по умолчанию "1 1 1"</p>
getMatrix	-	<p>Считает матрицу вида для трехмерного объекта.</p> <p>* @param cameraMatrix {number[]} Матрица камеры.</p> <p>* @param projectionMatrix {number[]} Матрица проекции.</p> <p>* @returns {number[]} Матрица вида.</p>
getNormalMatrix	-	<p>Считает матрицу нормалей для трехмерного объекта.</p> <p>* @returns {number[]} Матрица нормалей.</p>

Таблица 8. Поля и методы класса Shape.

МЕТОДЫ		
Наименование	Модификатор	Описание
init	static	<p>Запускает асинхронную инициализацию дочерних элементов.</p> <p>* @param shapeElement Ссылка на DOM-элемент, который иллюстрирует этот объект.</p> <p>* @param transform {Transform} Данные о положении фигуры в пространстве сцены.</p> <p>* @param scene {Scene} Трехмерная сцена, в которой определен объект.</p>

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.03.05-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Таблица 9. Поля и методы класса Appearance.

МЕТОДЫ		
Наименование	Модификатор	Описание
init	static	Запускает инициализацию тега, описывающего цвет текущей формы. * @param appearanceElement Ссылка на элемент, с которым мы в данный момент работаем. * @param {Number} vertexCount Количество вершин, которые необходимо покрасить.

Таблица 10. Поля и методы класса Color.

ПОЛЯ		
Наименование	Тип	Назначение
colors	number[]	Список цветов вершин фигуры.
МЕТОДЫ		
Наименование	Модификатор	Описание
-	constructor	Создает экземпляр Color. Получает атрибут "color" - массив - rgb-цвет (значения в пределах 0..255) - по умолчанию белый * @param colorElement Ссылка на DOM-элемент, который иллюстрирует этот объект. * @param vertexCount {number}

Таблица 11. Поля и методы класса Box.

ПОЛЯ		
Наименование	Тип	Назначение
size	number[]	Размер по трем осям.
boxElement	HTMLElement	Ссылка на DOM-элемент, который иллюстрирует этот объект.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.03.05-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

vertices	number[]	Массив вершин фигуры.
normals	number[]	Массив нормалей фигуры.
indices	number[]	Массив индексов поверхностей фигуры.
МЕТОДЫ		
Наименование	Модификатор	Описание
-	constructor	Создает экземпляр Box. * @param boxElement Ссылка на DOM-элемент, который иллюстрирует этот объект. * @param func {Function} Функция-колбэк, которая должна оповестить об окончании загрузки данных фигуры.
init	-	Инициализирует трехмерный объект Box, используя атрибуты тега my-box. Задает вершины и нормали, необходимые для отрисовки прямоугольного параллелепипеда. Из атрибута size вида массив получают размеры параллелепипеда по соответствующим сторонам. * По умолчанию size="10 10 10". * Центр прямоугольного параллелепипеда совпадает с началом координат.

Таблица 12. Поля и методы класса IndexedFaceSet.

ПОЛЯ		
Наименование	Тип	Назначение
src	string	Путь к модели, которую будет отрисовывать данный объект.
indexedFaceSet Element	HTMLElement	Ссылка на DOM-элемент, который иллюстрирует этот объект.
vertices	number[]	Массив вершин фигуры.
normals	number[]	Массив нормалей фигуры.
indices	number[]	Массив индексов поверхностей фигуры.
МЕТОДЫ		

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.03.05-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Наименование	Модификатор	Описание
-	constructor	Создает экземпляр IndexedFaceSet. * @param indexedFaceSetElement Ссылка на DOM-элемент, который иллюстрирует этот объект.
init	-	Инициализирует объект, используя данные атрибутов. Получает путь к .obj файлу - модели, которую необходимо отрендерить, через атрибут model.

Таблица 13. Поля и методы класса Algebra.

МЕТОДЫ		
Наименование	Модификатор	Описание
translation	static	Создает матрицу переноса. * @param tx {number} Перенос по оси X. * @param ty {number} Перенос по оси Y. * @param tz {number} Перенос по оси Z. * @returns {number[]} Матрица переноса.
xRotation	static	Создает матрицу поворота по оси X. * @param angleInRadians {number} Угол поворота в радианах по оси X. * @returns {number[]} Матрица поворота.
yRotation	static	Создает матрицу поворота по оси Y. * @param angleInRadians {number} Угол поворота в радианах по оси Y. * @returns {number[]} Матрица поворота.
zRotation	static	Создает матрицу поворота по оси Z. * @param angleInRadians {number} Угол поворота в радианах по оси Z. * @returns {number[]} Матрица поворота.
scaling	static	Создает матрицу масштабирования. * @param sx {number} Масштаб по оси X. * @param sy {number} Масштаб по оси Y. * @param sz {number} Масштаб по оси Z. * @returns {number[]} Матрица масштабирования.
translate	static	Выполняет перенос матрицы на заданные значения. * @param m {number[]} Исходная матрица

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.03.05-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

		<ul style="list-style-type: none"> * @param tx {number} Перенос по оси X. * @param ty {number} Перенос по оси Y. * @param tz {number} Перенос по оси Z. * @returns {number[]} Результирующая матрица.
xRotate	static	<p>Выполняет поворот матрицы по оси X.</p> <ul style="list-style-type: none"> * @param m {number[]} Исходная матрица. * @param angleInRadians {number} Угол поворота по оси X в радианах. * @returns {number[]} Результирующая матрица.
yRotate	static	<p>Выполняет поворот матрицы по оси Y.</p> <ul style="list-style-type: none"> * @param m {number[]} Исходная матрица. * @param angleInRadians {number} Угол поворота по оси Y в радианах. * @returns {number[]} Результирующая матрица.
zRotate	static	<p>Выполняет поворот матрицы по оси Z.</p> <ul style="list-style-type: none"> * @param m {number[]} Исходная матрица. * @param angleInRadians {number} Угол поворота по оси Z в радианах. * @returns {number[]} Результирующая матрица.
scale	static	<p>Выполняет масштабирование матрицы на заданные параметры.</p> <ul style="list-style-type: none"> * @param m {number[]} Исходная матрица * @param sx {number} Масштабирование по оси X. * @param sy {number} Масштабирование по оси Y. * @param sz {number} Масштабирование по оси Z. * @returns {number[]} Результирующая матрица.
perspective	static	<p>Создает матрицу перспективы по заданным параметрам.</p> <ul style="list-style-type: none"> * @param fieldOfViewInRadians {number} Угол отображения области видимости в радианах. * @param aspect {number} Отношение ширины сцены к высоте. * @param near {number} Наименьший индекс, видимый камере, по Z. * @param far {number} Наибольший индекс, видимый камере, по Z. * @returns {number[]} Матрица перспективы.
multiply	static	<p>Перемножает две матрицы в обратном порядке.</p> <ul style="list-style-type: none"> * @param b {number[]} Матрица-правый множитель. * @param a {number[]} Матрица-левый множитель. * @returns {number[]} Результат матричного умножения.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.03.05-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

identity	static	Создает единичную матрицу. * @returns {number[]} Единичная матрица.
inverse	static	Создает обратную матрицу. * @param m {number[]} Исходная матрица. * @returns {number[]} Матрица обратная исходной.
normalFromMat4	static	Преобразует матрицу вида(4*4) в нормальную матрицу(3*3). * @param a {number[]} Исходная матрица вида. * @returns {number[]} Нормальная матрица для исходной.
normalize	static	Нормирует исходный вектор. * @param v {number[]} Исходный вектор. * @returns {number[]} Нормированный вектор.

Таблица 14. Поля и методы класса Utils.

МЕТОДЫ		
Наименование	Модификатор	Описание
getRadians	static	Перевод из градусов в радианы. * @param angle {number} Угол в градусах. * @returns {number} Угол в радианах.
checkArrayAttribute	static	Проверка значения на принадлежность к множеству массивов из трех вещественных чисел. Бросает исключение в случае не принадлежности. * @param value Проверяемое значение. * @param tagName {string} Имя тега, откуда значение пришло. * @param attributeName {string} Имя атрибута, откуда значение пришло.
checkNumberAttribute	static	Проверка значения на принадлежность к вещественным числам. Бросает исключение в случае не принадлежности. * @param value Проверяемое значение. * @param tagName {string} Имя тега, откуда значение пришло. * @param attributeName {string} Имя атрибута, откуда значение пришло.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.03.05-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

checkNumber	static	Проверка значения на принадлежность к вещественным числам. * @param value Проверяемое значение. * @returns {boolean} Флаг принадлежности к вещественным числам.
--------------------	--------	---

Таблица 15. Поля и методы класса OBJLoader.

ПОЛЯ		
Наименование	Тип	Назначение
vertices	number[]	Массив вершин загружаемой модели.
normals	number[]	Массив нормалей загружаемой модели.
indices	number[]	Массив индексов поверхностей загружаемой модели.
МЕТОДЫ		
Наименование	Модификатор	Описание
-	constructor	Создает экземпляр OBJLoader.
calculateNormals	-	По заданным индексам поверхностей и вершинам формирует массив нормалей.
parse	-	Преобразует текстовое представление .obj файла в массив вершин, нормалей и индексов. * @param objectData {string} Текстовое представление .obj файла.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.03.05-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

[illegible]

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.03.05-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата