

---

# Blend4Web. Руководство пользователя

*Выпуск v15.10*

ООО "Триумф"

29 October 2015

<b>1 Общие сведения</b>	<b>1</b>
1.1 Что такое Blend4Web . . . . .	1
1.2 “Движок” . . . . .	1
1.3 Графический движок, трехмерный движок . . . . .	2
1.4 Что такое WebGL . . . . .	2
1.5 Поддержка WebGL в браузерах . . . . .	2
1.6 Преимущества WebGL . . . . .	3
1.7 Что такое Blender . . . . .	3
1.8 3D моделирование . . . . .	3
1.9 Браузерные технологии . . . . .	4
1.10 Интерактивная графика . . . . .	5
1.11 Видео-карты и драйверы . . . . .	5
<b>2 Функционал движка</b>	<b>7</b>
2.1 Общее . . . . .	7
2.2 Текстуры . . . . .	7
2.3 Материалы . . . . .	7
2.4 Освещение . . . . .	8
2.5 Тени . . . . .	8
2.6 Система частиц . . . . .	8
2.7 Рендеринг наружных сцен . . . . .	9
2.8 Постпроцессинговые эффекты . . . . .	9
2.9 Анимация . . . . .	9
2.10 Оптимизация . . . . .	9
2.11 Звук . . . . .	10
2.12 Физика . . . . .	10
2.13 Событийная модель . . . . .	11
2.14 Визуальное программирование . . . . .	11
2.15 Прочее . . . . .	11
<b>3 Экспресс-установка</b>	<b>12</b>
3.1 Установка программы Blender . . . . .	12
3.2 Установка аддона движка . . . . .	13
3.3 Профиль Blend4Web . . . . .	14

---

3.4	Экспорт и просмотр сцены . . . . .	14
3.5	Установка новой версии аддона . . . . .	15
<b>4</b>	<b>Развёртывание среды разработки</b>	<b>16</b>
4.1	Распаковка архива . . . . .	16
4.2	Подключение аддона . . . . .	16
4.3	Профиль Blend4Web . . . . .	18
4.4	Локальный сервер разработки . . . . .	19
4.5	Запуск просмотрщика сцен и демо-приложений . . . . .	21
<b>5</b>	<b>Рабочий процесс</b>	<b>23</b>
5.1	Подготовка сцен . . . . .	23
5.2	Экспорт сцен . . . . .	24
5.3	Отображение сцен в просмотрщике . . . . .	24
5.4	Разработка приложения . . . . .	27
<b>6</b>	<b>Просмотрщик сцен</b>	<b>28</b>
6.1	Навигация . . . . .	28
6.2	Боковая панель . . . . .	28
6.3	Индикаторы . . . . .	31
<b>7</b>	<b>Веб-плеер</b>	<b>32</b>
7.1	Использование . . . . .	33
7.2	Навигация . . . . .	33
7.3	Панель управления . . . . .	33
7.4	Атрибуты . . . . .	34
7.5	Название сцены в заголовке . . . . .	35
7.6	Ошибки сцены . . . . .	35
<b>8</b>	<b>Аддон</b>	<b>38</b>
8.1	Опции экспорта . . . . .	38
8.2	Ошибки инициализации . . . . .	42
8.3	Ошибки совместимости . . . . .	43
8.4	Критические ошибки экспорта . . . . .	45
8.5	Некритические ошибки экспорта . . . . .	48
8.6	Прочие сообщения . . . . .	53
8.7	Перевод аддона . . . . .	53
<b>9</b>	<b>Объекты</b>	<b>55</b>
9.1	Типы . . . . .	55
9.2	Статические и динамические объекты . . . . .	55
9.3	Настройка . . . . .	56
9.4	Управление перемещением объектов . . . . .	61
9.5	Кватернионы . . . . .	62
9.6	Отличия систем координат Blender'a и Blend4Web . . . . .	64
9.7	Перемещение с помощью векторов TSR . . . . .	65
9.8	Копирование объектов (инстансинг) . . . . .	65
9.9	Выделение объектов . . . . .	67
9.10	Морфинг . . . . .	67

<b>10 Камера</b>	<b>69</b>
10.1 Режимы управления и общие настройки . . . . .	69
10.2 Скорость движения камеры . . . . .	70
10.3 Ограничения движения камеры . . . . .	70
10.4 Управление камерой через API . . . . .	77
<b>11 Материалы</b>	<b>80</b>
11.1 Параметры освещения . . . . .	81
11.2 Прозрачность . . . . .	82
11.3 Зеркальное отражение . . . . .	84
11.4 Опции рендеринга . . . . .	87
11.5 Специальные параметры движка . . . . .	88
11.6 Материалы гало (Halo) . . . . .	88
<b>12 Текстуры</b>	<b>90</b>
12.1 Типы текстур . . . . .	90
12.2 Общие настройки . . . . .	91
12.3 Диффузная текстура (diffuse map) . . . . .	92
12.4 Карта бликов (specular map) . . . . .	92
12.5 Карта нормалей (normal map) . . . . .	93
12.6 Карта высот (height map). Parallax mapping . . . . .	93
12.7 Карта смешивания (stencil map) . . . . .	94
12.8 Видео-текстуры . . . . .	96
12.9 Карта окружения (environment map) . . . . .	98
12.10 Карта зеркального отражения (mirror map) . . . . .	100
12.11 Текстура неба (skydome) . . . . .	101
12.12 Особые типы текстур . . . . .	103
<b>13 Нодовые материалы</b>	<b>107</b>
13.1 Стандартные ноды . . . . .	107
13.2 Дополнительные ноды . . . . .	108
<b>14 Освещение, тени и фон</b>	<b>116</b>
14.1 Типы затенения . . . . .	116
14.2 Редактор нормалей . . . . .	117
14.3 Освещение от источников света . . . . .	130
14.4 Освещение от окружающей среды . . . . .	131
14.5 Тени . . . . .	133
14.6 Цвет фона . . . . .	139
<b>15 Постпроцессинговые эффекты</b>	<b>142</b>
15.1 Размытие при движении . . . . .	142
15.2 Глубина резкости камеры . . . . .	143
15.3 Взаимное затенение . . . . .	144
15.4 Сумеречные лучи . . . . .	146
15.5 Эффект засветки ярких деталей . . . . .	147
15.6 Подсветка контура (outlining) . . . . .	148
15.7 Аналиф стереоизображение . . . . .	149
15.8 Коррекция цвета . . . . .	150

15.9 Свечение (glow) . . . . .	150
15.10 Сглаживание . . . . .	151
<b>16 Система частиц. Флюиды</b>	<b>153</b>
16.1 Использование . . . . .	154
16.2 Настройка . . . . .	155
16.3 Текстуры в системах частиц . . . . .	160
<b>17 Система частиц. Инстансинг</b>	<b>163</b>
17.1 Настройки системы частиц . . . . .	164
17.2 Травяной покров . . . . .	169
17.3 Листва деревьев . . . . .	171
<b>18 Анимация</b>	<b>174</b>
18.1 Управление анимацией . . . . .	174
18.2 Объектная анимация . . . . .	175
18.3 Скиннинг и скелетная анимация . . . . .	176
18.4 Вертекская анимация . . . . .	178
18.5 Параметризация источников звука . . . . .	179
18.6 Анимация над Value и RGB . . . . .	179
<b>19 Рендеринг наружных сцен</b>	<b>182</b>
19.1 Вода . . . . .	182
19.2 Атмосфера . . . . .	191
19.3 Ветер . . . . .	194
<b>20 Гамма-коррекция и альфа-композитинг</b>	<b>197</b>
20.1 Общее описание . . . . .	197
20.2 Человеческое зрение и мониторы . . . . .	198
20.3 Гамма . . . . .	198
20.4 Коррекция в нодовых материалах . . . . .	199
20.5 Альфа-композитинг . . . . .	200
<b>21 Звуковая подсистема</b>	<b>202</b>
21.1 Настройка звуковых источников . . . . .	202
21.2 Обработка и кодирование . . . . .	204
<b>22 Физика</b>	<b>206</b>
22.1 Подготовка к использованию . . . . .	206
22.2 Статический тип физики . . . . .	206
22.3 Динамический тип физики . . . . .	208
22.4 Ограничители (Constraints) . . . . .	210
22.5 Колесные транспортные средства . . . . .	212
22.6 Плавающие объекты . . . . .	214
22.7 Плавающие транспортные средства . . . . .	216
22.8 Особенности использования в приложениях . . . . .	217
<b>23 Нелинейная анимация</b>	<b>218</b>
23.1 Редактор NLA . . . . .	218
23.2 Управление через API . . . . .	220

<b>24 Визуальное программирование</b>	<b>221</b>
24.1 Визуальное программирование (Logic Editor) . . . . .	221
<b>25 Разработчикам приложений</b>	<b>228</b>
25.1 Hello world! . . . . .	228
25.2 Загрузка сцены в приложение . . . . .	229
25.3 Система модулей . . . . .	229
25.4 Быстрое создание приложений . . . . .	230
25.5 Система управления проектами . . . . .	231
25.6 Подготовка ресурсов . . . . .	239
25.7 Конвертация ресурсов . . . . .	242
25.8 Специфика неполноэкраных приложений . . . . .	245
25.9 Примеры кода . . . . .	246
25.10 Событийная модель . . . . .	247
25.11 Файловая структура SDK . . . . .	249
25.12 Загрузка локальных ресурсов . . . . .	252
25.13 Профили качества изображения . . . . .	252
<b>26 Разработчикам движка</b>	<b>254</b>
26.1 Стиль оформления кода . . . . .	254
26.2 Сборка движка . . . . .	255
26.3 Сборка аддона . . . . .	256
26.4 Зависимости . . . . .	256
26.5 Названия функций и переменных . . . . .	257
26.6 Отладка . . . . .	258
26.7 Компиляция шейдеров . . . . .	258
26.8 Обновление переводов аддона . . . . .	264
<b>27 Работа в команде с использованием Git</b>	<b>266</b>
27.1 Общие сведения . . . . .	266
27.2 Типичный рабочий процесс . . . . .	266
27.3 Индивидуальные настройки . . . . .	267
27.4 Проверка статуса . . . . .	267
27.5 Перед коммитом . . . . .	268
27.6 Подготовка к коммиту . . . . .	269
27.7 Коммит . . . . .	270
27.8 Синхронизация между репозиториями . . . . .	271
27.9 Разрешение конфликтов . . . . .	273
27.10 Тэги . . . . .	275
27.11 Другие полезные команды . . . . .	276
<b>28 Проблемы и решения</b>	<b>277</b>
28.1 Проблемы при запуске движка . . . . .	277
28.2 Ошибка инициализации WebGL . . . . .	278
<b>29 Замечания к релизам</b>	<b>282</b>
29.1 v15.10 . . . . .	282
29.2 v15.09 . . . . .	285
29.3 v15.08 . . . . .	290

29.4 v15.07 . . . . .	293
29.5 v15.06 . . . . .	297
29.6 v15.05 . . . . .	302
29.7 v15.04 . . . . .	306
29.8 v15.03 . . . . .	310
29.9 v15.02 . . . . .	314
29.10 v15.01 . . . . .	317
29.11 v14.12 . . . . .	321
29.12 v14.11 . . . . .	323
29.13 v14.10 . . . . .	325
29.14 v14.09 . . . . .	327

**Алфавитный указатель** **331**

## Общие сведения

---

### 1.1 Что такое Blend4Web

Blend4Web - это программная среда для подготовки и интерактивного отображения трехмерного аудиовизуального контента в браузерах, т.е. трехмерный «движок» (жаргонизм от англ. «engine»).

Платформа предназначена для создания визуализаций, презентаций, интернет-магазинов, игр и других “насыщенных” web-приложений.

Фреймворк Blend4Web имеет тесную интеграцию с пакетом 3D моделирования и анимации Blender (отсюда название). Отображение контента осуществляется средствами WebGL и других браузерных технологий, без использования плагинов.

Технически Blend4Web представляет собой программную библиотеку, подключающуюся в web-странице, аддон (дополнение) к программе Blender, а также инструменты для отладки и оптимизации.

3D движок Blend4Web разрабатывается сотрудниками ООО «Триумф» с 2010 г. Первый релиз движка состоялся 28 марта 2014 г.

### 1.2 “Движок”

Движок - это обособленная часть программного кода, используемая внешними приложениями для реализации нужного им функционала.

Типы движков: сайтовый движок, блоговый движок, движок интернет-магазина, wiki-движок, поисковый движок, игровой движок и т.д. Экономический смысл существования программных движков заключается в многократном использовании одного и того же функционала. Например, используя тот или иной движок, разработчики могут относительно дешево создавать интернет-магазины или игры.

## 1.3 Графический движок, трехмерный движок

Графический движок выполняет специализированные функции по отображению графики. Он является промежуточным звеном между:

- высокоуровневой прикладной частью (игровой логикой, бизнес-логикой) и
- низкоуровневой системной частью (например, графической библиотекой *WebGL* и нижлежащими *драйверами*).

Графический движок может объединяться со звуковой системой, физическим движком, системой, реализующей искусственный интеллект, сетевой системой, а также редактором сцен и логики, образуя интегрированный инструментарий для создания 3D приложений - **трехмерный движок**.

## 1.4 Что такое WebGL

WebGL (Web Graphics Library, т.е. графическая библиотека для использования в веб-приложениях) - одна из современных браузерных технологий, позволяющая создавать трехмерные графические веб-приложения. Другими словами WebGL - это “3D в браузере”.

## 1.5 Поддержка WebGL в браузерах

В настоящий момент технология WebGL в той или иной степени поддерживается во всех браузерах.

### 1.5.1 Полная поддержка

- Chrome
- Yandex Browser
- Firefox
- Opera
- Safari 8+

### 1.5.2 Экспериментальная поддержка

- Internet Explorer 11
- Microsoft Edge

### 1.5.3 Мобильные платформы

- Android
- BlackBerry
- Firefox OS
- iOS 8
- Sailfish OS
- Tizen
- Ubuntu Touch

## 1.6 Преимущества WebGL

- работает в браузерах без установки дополнительных программ (плагинов)
- кроссплатформенный, предназначен для работы во всех стационарных и мобильных системах
- является [открытым стандартом](#), не требует лицензионных отчислений
- поддерживается ведущими участниками рынка IT (Google, Apple, Microsoft, Nvidia, Samsung, Adobe и др.)
- основан на знакомой разработчикам технологии OpenGL
- интегрируется с другими *браузерными технологиями*

## 1.7 Что такое Blender

Blender - это популярная программа для создания 3D-моделей и анимации, бесплатная и с открытым кодом. Подготовленные в программе модели и сцены могут быть отображены, например, с помощью [трехмерного движка](#) на странице веб-сайта.

## 1.8 3D моделирование

Создание графических ресурсов требует наличия подготовленных специалистов - 3D-художников.

Типичный рабочий процесс может состоять из следующих этапов:

- подбор фотографий и/или создание концепта и скетчей (“спереди”-“сбоку”-“сверху”) будущей модели или сцены
- моделирование - создается трехмерная модель, состоящая из многоугольников (полигонов)

- текстурная развертка - на модели создается разметка для последующего наложения текстур (плоских изображений)
- текстурирование - на 3D-модель накладываются текстуры
- подбор материалов - назначение различным частям модели материалов и их настройка (например, деревянная дверь с металлической ручкой)
- риггинг (от англ. rigging, т.е. “оснастка”) - к модели прикрепляются управляющие элементы (“кости” “скелета”) с целью дальнейшей анимации
- анимация - модель приводится в движение с целью визуализации действий (например, персонажей)
- экспорт - выполняется на любом этапе с целью отображения 3D-модели в ее конечном виде, например, на веб-странице

Кроме того, в процессе создания 3D-моделей часто используются техники повышения реализма, требующие отдельных этапов:

- создание высокополигональной модели - создается детализированная версия модели
- “запекание” карты нормалей - детали из высокополигональной модели переносятся на основную модель в виде специальной текстуры (карты нормалей)
- создание карты отражения - различным частям модели назначается различный цвет и степень отражения света
- запекание карт окружения - производится с целью реализации эффекта отражения окружающей среды на поверхности модели
- настройка камеры и источников света на сцене
- настройка параметров физической симуляции - частицы, ткань

Затраты времени при изготовлении 3D-моделей и анимации зависят от их сложности и требуемого качества, и могут изменяться от 1-2 дней (например, игровой предмет) до 1-2 недель (например, детализированная модель самолета) и даже нескольких месяцев (реалистичные персонажи с наборами одежды, волос, лиц, с анимацией и настройкой пропорций фигуры).

## 1.9 Браузерные технологии

Браузер (от англ. “browser”, т.е. “просмотрщик”) - программа для воспроизведения содержимого сети Интернет. На заре развития интернет-технологий роль браузера сводилась к просмотру текстовых страниц с включениями статических изображений (“гипер-текст”). Современные браузеры представляют собой полнофункциональные платформы для создания мультимедийных веб-приложений.

Среди реализованных и перспективных возможностей браузеров, используемых в *Blend4Web*, можно отметить следующие технологии:

- трехмерная графика, WebGL

- типизированные массивы, [Typed Array](#)
- временной контроль анимации (`requestAnimationFrame`), [Timing control for script-based animations](#)
- двухмерная графика, [HTML Canvas 2D Context](#)
- обработка звука, [Web Audio API](#)
- загрузка бинарных данных, [XMLHttpRequest Level 2](#)
- полноэкранный режим, [Fullscreen](#)
- захват курсора мыши, [Pointer Lock](#)
- многопоточные вычисления, [Web Workers](#)
- ориентация и перемещение мобильных устройств, [Device Orientation](#)

Другие перспективные технологии:

- векторная графика, [Scalable Vector Graphics \(SVG\)](#)
- безопасный доступ к файлам, [File API](#), [File API: Directories and System](#)
- потоковое соединение между браузерами, [WebRTC](#)
- постоянное сетевое подключение, [The WebSocket API](#)
- игровые пульты, [Gamepad](#)

## 1.10 Интерактивная графика

Термин “интерактивный” в приложении к компьютерной графике означает, что пользователь имеет возможность взаимодействовать с постоянно меняющимся изображением. Например, пользователь может изменять направление взгляда в 3D сцене, перемещать объекты, инициировать анимацию и выполнять другие действия, обычно ассоциирующиеся с компьютерными играми.

Интерактивность графики достигается за счет частой смены изображений, так что действие пользователя (например, движение курсора или нажатие кнопки) в промежутках между кадрами приводит к изменению изображения в следующем кадре. Изображения должны сменять друг друга так часто, чтобы человеческий глаз не был способен распознать их по отдельности (быстрее 30 кадров в секунду).

Близким по смыслу термином является также “графика реального времени”, или “рендеринг реального времени” (от англ. `rendering`, т.е. “отображение”).

## 1.11 Видео-карты и драйверы

Интерактивная графика реализуется специализированной аппаратной частью современных компьютеров, называемой графическим процессором, который может быть

выполнен в виде отдельного устройства (видео-карты) или как часть центрального процессора.

Основные производители графических процессоров (в скобках указаны их торговые марки), для настольных компьютеров - NVidia (GeForce, Quadro), AMD (Radeon), Intel (HD), для мобильных устройств - ARM (Mali), PowerVR (SGX), Nvidia (Tegra), Qualcomm (Adreno).

Доступ программ к ресурсам графического процессора осуществляется через программу-посредника, называемого драйвером. Важным условием для корректной работы интерактивных графических программ является наличие в системе драйверов последней версии. Драйверы можно установить (или обновить), загрузив их с соответствующих сайтов производителей графических процессоров. Подробнее в разделе *Ошибка инициализации WebGL*.

### Функционал движка

---

## 2.1 Общее

- эффективный рендеринг трёхмерных сцен любой сложности и размеров
- формат хранения данных, учитывающий специфику работы WebGL и оптимизированный с целью минимизации времени загрузки
- удобная среда разработки трёхмерного контента на основе плагина к Blender

## 2.2 Текстуры

- текстурирование (texturing) - покрытие поверхности 3D объекта плоским изображением
- мультитекстурирование (multitexturing) - использование для объекта нескольких текстур
- рендеринг в текстуру (render-to-texture, RTT) для реализации вложения одной сцены в другую и постпроцессинговых эффектов
- анизотропная фильтрация для улучшения качества обзора поверхности под косыми углами (anisotropic filtering, AF, использовано стандартное расширение WebGL)
- поддержка текстурной компрессии (формат S3TC/DXT)
- видео-текстуры - загрузка и воспроизведение видео на текстуре
- Canvas-текстуры - возможность отрисовки двухмерной графики через Canvas API в текстуре

## 2.3 Материалы

- прозрачность материалов, сортировка по глубине при необходимости (z-sorting)

- улучшенная детализация рельефных поверхностей текстурами (использован метод parallax offset mapping)
- зависимость степени отражения от угла обзора - эффект Френеля (Fresnel)
- динамическое отражение
- поддержка нодовых материалов
- гало материал для рендеринга источников света и звезд (Halo material)

## 2.4 Освещение

- освещение несколькими источниками света
- типы источников света - прямой (directional), полусферический (hemisphere), точечный (point), конический (spot)
- диффузное (т.е. рассеянное) освещение объектов (diffuse lighting)
- рассеянное освещение от окружающей среды (ambient lighting)
- зеркальное отражение света от поверхности объектов (specular lighting)
- зеркальное отражение окружающей среды (environment mapping)
- дополнительная детализация картами нормалей к поверхности (normal mapping)

## 2.5 Тени

- статические падающие тени (light mapping)
- динамические падающие тени (использован метод shadow mapping)
- собственные тени — объекты отбрасывают тени сами на себя (self-shadowing)
- каскадные тени для больших сцен (cascaded shadow mapping, CSM)
- мягкие тени

## 2.6 Система частиц

- система частиц (particle system) для реализации эффектов, таких как огонь, дым, брызги и т.д.
- система частиц для расстановки (инстансинга) однородных объектов: трава, камни, листва деревьев и проч.

## 2.7 Рендеринг наружных сцен

- туман (fog)
- купол неба/окружающего пространства (skydome)
- эффект линз при направлении камеры на источник света (lens flares)
- рендеринг воды

## 2.8 Постпроцессинговые эффекты

- размытие при движении (motion blur)
- антиалиасинг - уменьшение зубчатости краев изображения в результате рендеринга в текстуру (использован метод fast approximate anti-aliasing, FXAA)
- стерео-изображение (анаглифное, 3D очки)
- рассеянное затенение от окружающей среды (ambient occlusion, используется метод SSAO)
- глубина резкости для камеры (DOF)
- сумеречные лучи (god rays)
- эффект засветки ярких объектов (bloom)
- подсветка контуров объектов (outlining)

## 2.9 Анимация

- сканинг (skinning) - деформация объекта с помощью системы костей
- анимация перемещения, вращения, масштабирования объектов, камер и источников света
- скелетная анимация (например, для тела персонажа)
- вертексная анимация (например, для симуляции ткани)
- процедурная анимация (например, изгибание растений на ветру)
- анимация текстурных координат (например, для визуализации волн воды)

## 2.10 Оптимизация

- оптимизация отсечением по зоне видимости (frustum culling)

- оптимизация уменьшением количества вызовов WebGL — батчинг, текстурные атласы (batching, texture atlases)
- оптимизация уменьшением уровня детализации объектов на удалении (level of detail, LOD)

## 2.11 Звук

- звуковой движок, основанный на Web Audio API
- поддержка различных форматов файлов с учётом различий браузеров
- гибкое управление воспроизведением, возможность приостановки звука
- позиционирование источников в трёхмерном пространстве
- эффект Допплера для движущихся объектов с возможностью отключения и компенсацией скачков в пространстве
- управление громкостью, скоростью и задержкой воспроизведения
- эффекты плавного перехода громкости (fade-in, fade-out, duck)
- качественное зацикливание звуков (looping)
- рандомизация звуковых параметров для улучшения восприятия повторяющихся звуков
- поддержка кроссфейдерной звуковой анимации
- динамический компрессор
- эффективное хранение и воспроизведение длинных музыкальных композиций
- инструменты для сведения (микширования) звуковой картины в реальном времени

## 2.12 Физика

- физика жестких тел - определение столкновений, движение, гравитация, определение высоты, опрокидывание
- система соединителей (ограничителей) - жёсткие, гибкие, пружинящие, поворотные, скользящие итд.
- система трассировки лучей
- физика плавания объектов и движения в толще воды
- физика колёсных транспортных средств
- физика плавучих транспортных средств

## 2.13 Событийная модель

- асинхронный фреймворк для написания логики приложений
- управление анимацией и искусственный интеллект животных и персонажей

## 2.14 Визуальное программирование

- инструмент NLA Script предоставляет возможность создавать интерактивные приложения, конструируя логические цепочки из простых блоков

## 2.15 Прочее

- поддержка математических кривых для моделирования удлиненных объектов (дороги, провода, река)
- выбор пользователем объектов на 3D сцене (picking)
- минификация (уменьшение объема) и обfuscация (сокрытие) кода, необходимые для коммерческого использования движка
- модульная структура исходного кода
- мощный шейдерный препроцессор с поддержкой модулей и функциональных блоков (нод)
- удобная система для быстрого развертывания новых 3D приложений
- опции для поддержки работы на широком спектре оборудования
- руководство пользователя и документация для программистов
- взаимодействие с пользователем — управление камерой, персонажем, действиями

## Экспресс-установка

---

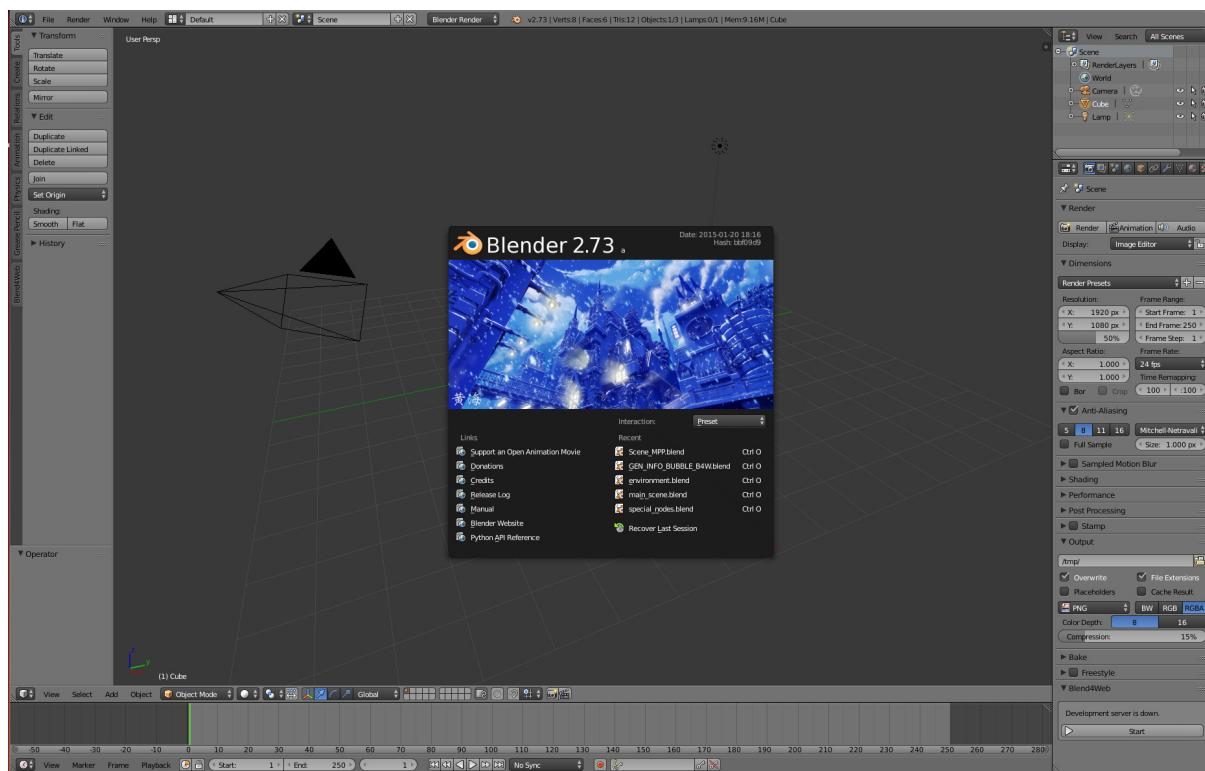
Экспресс-установка аддона Blend4Web подойдет обычным пользователям, которым не требуется разработка полноценных 3D приложений. Основной возможностью в этом случае является экспорт сцены в один HTML-файл с последующим просмотром в браузере, поддерживающем WebGL.

Для более серьезных задач необходима установка *комплекта средств разработки*.

### 3.1 Установка программы Blender

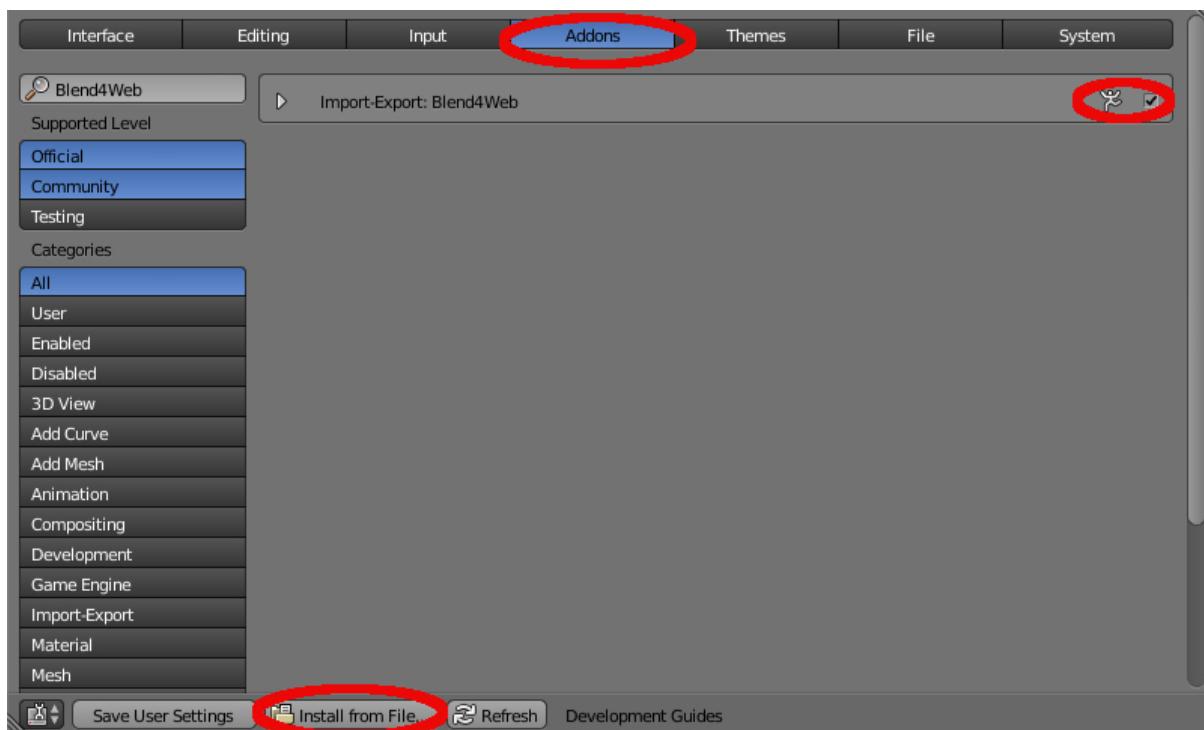
Создание 3D сцен осуществляется в графическом пакете Blender, который является программным продуктом с открытым исходным кодом и распространяется бесплатно.

Должна использоваться текущая стабильная версия Blender. Загрузить Blender можно с [официального сайта](#).



## 3.2 Установка аддона движка

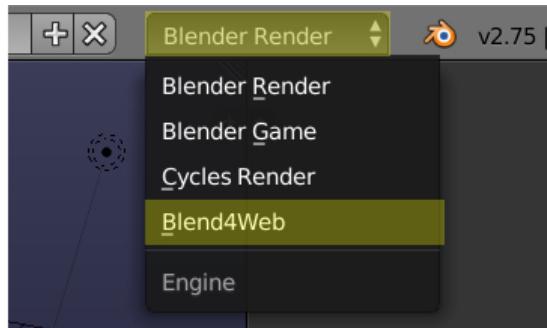
Запустить Blender, загрузить сцену по умолчанию **File > New**. Вызвать окно пользовательских настроек **File > User Preferences....** Во вкладке **Addons** нажать **Install from File...** и затем выбрать zip-архив с файлами аддона. После этого необходимо отметить галочку напротив **Import-Export: Blend4Web**.



Далее нажать Save User Settings и закрыть окно пользовательских настроек.

### 3.3 Профиль Blend4Web

Выберите профиль настроек Blend4Web в выпадающем меню верхней панели:



### 3.4 Экспорт и просмотр сцены

Созданную сцену можно экспортировать в формате HTML. Для этого нужно выбрать опцию File > Export > Blend4Web (.html) и указать путь экспорта. Полученный HTML файл можно открыть любым браузером, поддерживающим технологию WebGL.

**См.также:**

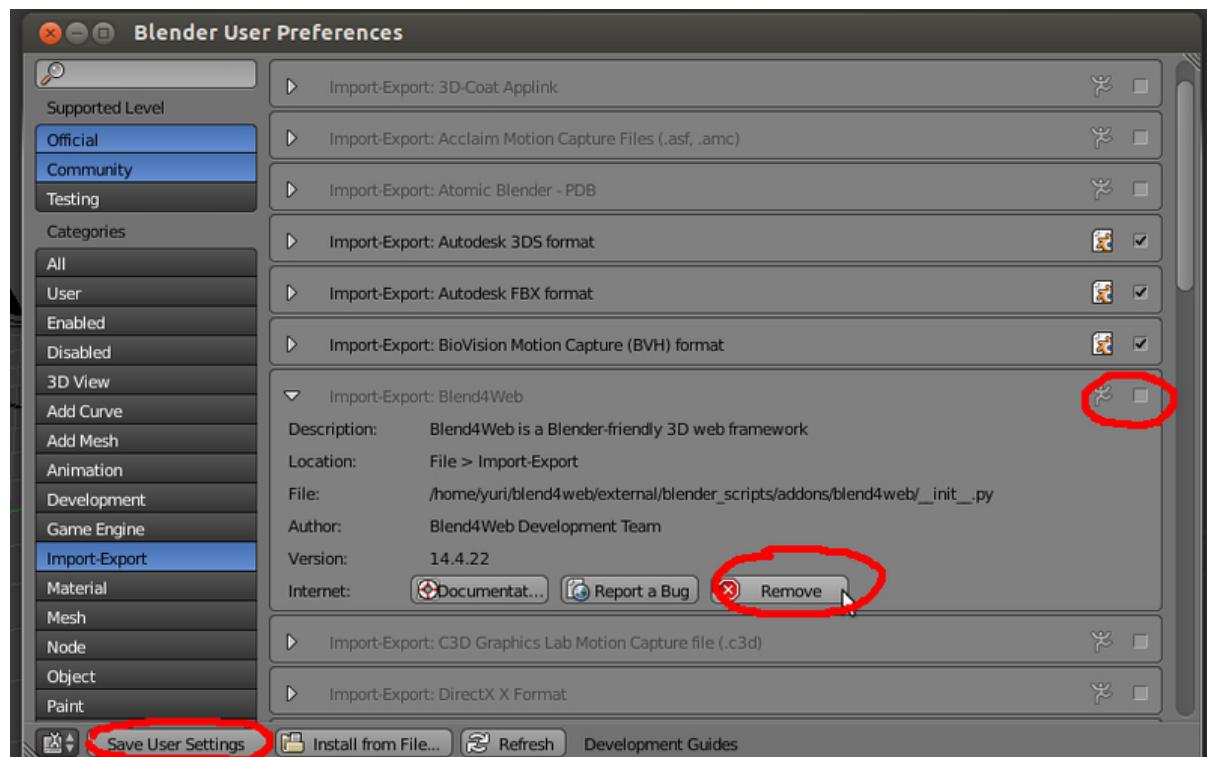
*Поддерэска WebGL в браузерах*

## 3.5 Установка новой версии аддона

Чтобы установить новую версию аддона, необходимо **сначала отключить, а затем удалить** старую версию.

Для отключения аддона запустить Blender, загрузить сцену по умолчанию **File > New**. Вызвать окно пользовательских настроек **File > User Preferences...**. Перейти на вкладку **Addons** и выбрать категорию **Import-Export**. Снять галочку напротив **Import-Export: Blend4Web**. Далее нажать **Save User Settings** и **перезапустить Blender**.

Затем для удаления аддона снова вызвать окно пользовательских настроек, раскрыть панель информации об аддоне Blend4Web и нажать кнопку **Remove**.



# Развёртывание среды разработки

---

Вариант установки с развёртыванием дистрибутива SDK рекомендуется для разработчиков 3D-приложений. Если же нужно быстрое ознакомление с возможностями Blend4Web, то, возможно, будет удобнее [экспресс-установка](#).

Перед установкой необходимо загрузить и установить совместимую версию Blender, согласно [таблице](#).

## 4.1 Распаковка архива

Стабильные версии дистрибутива поставляются в виде архива (`blend4web_sdk_free_YY_MM.zip` – бесплатный SDK, `blend4web_sdk_pro_YY_MM.zip` – коммерческий SDK). Достаточно распаковать данный архив в любое место на диске.

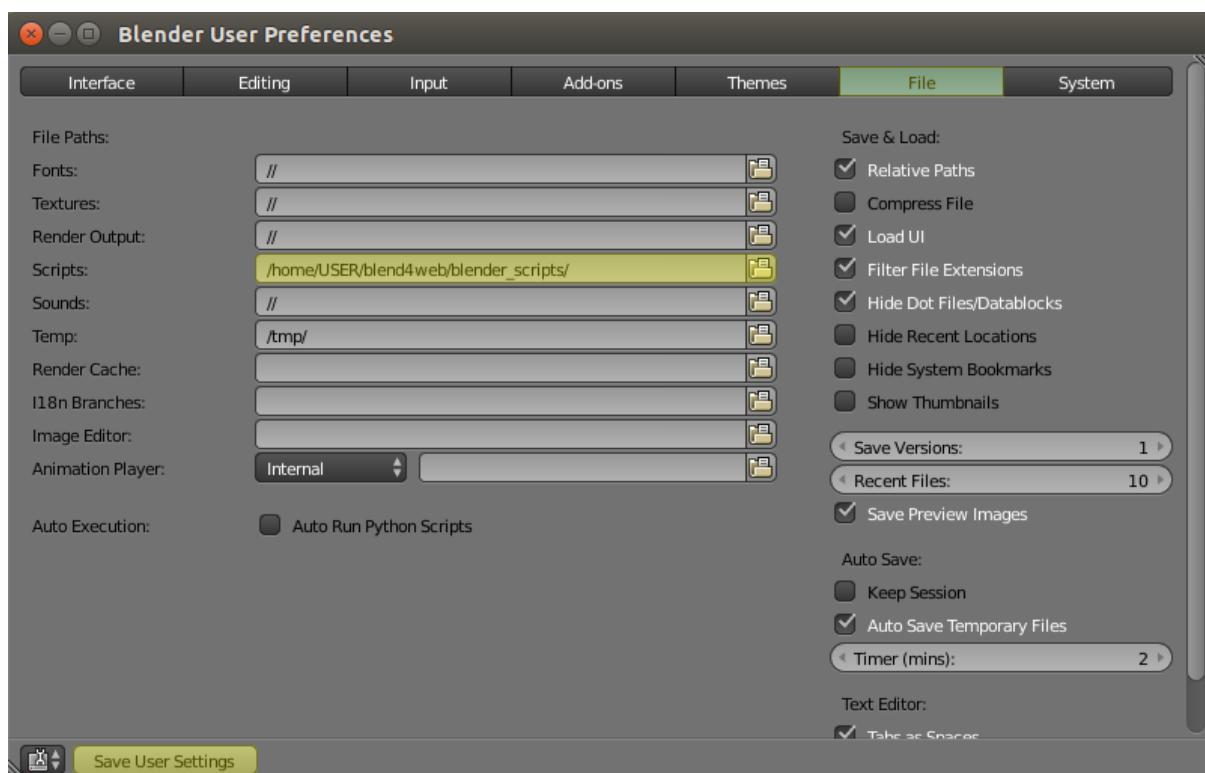
## 4.2 Подключение аддона

---

**Примечание:** Если аддон ранее был установлен с помощью [экспресс-установки](#), то рекомендуется прежде его удалить.

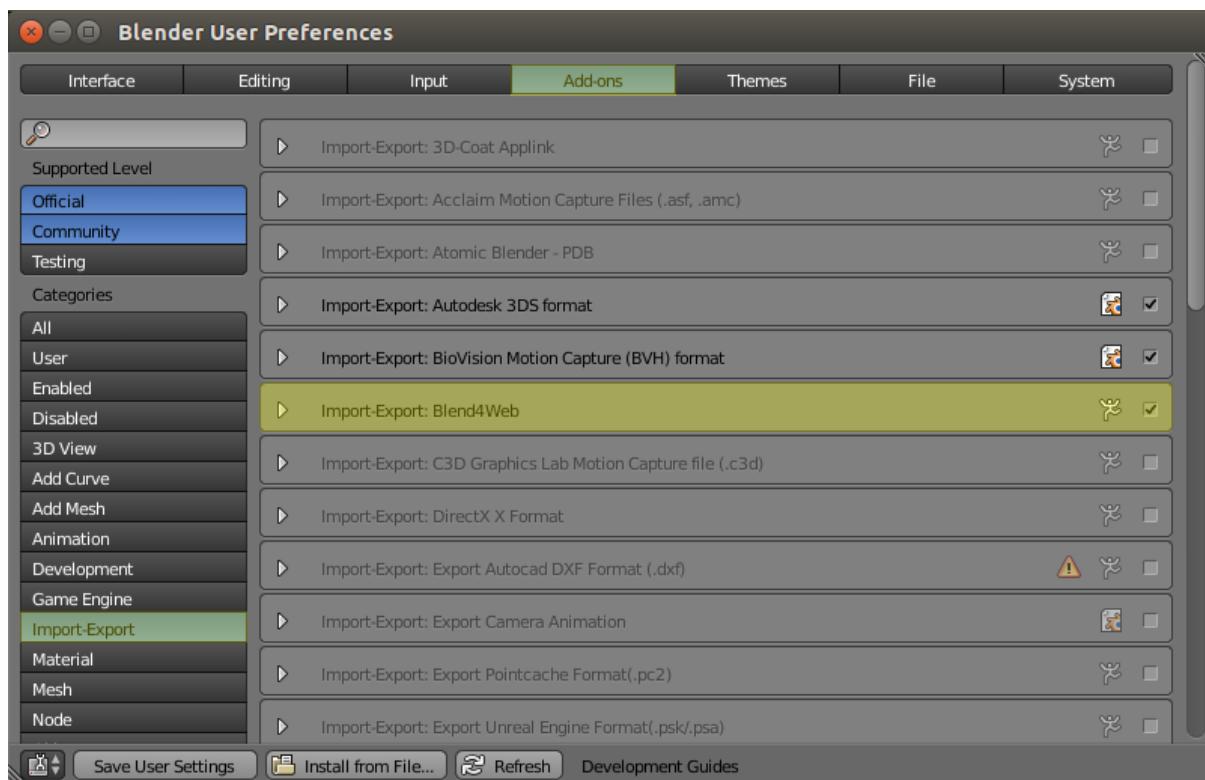
---

Запустить Blender, загрузить сцену по умолчанию `File > New` (горячие клавиши `Ctrl-N`). Вызвать окно пользовательских настроек `File > User Preferences...` (горячие клавиши `Ctrl-Alt-U`). Во вкладке `File` в поле `Scripts` выбрать путь к директории `blender_scripts`.



Нажать **Save User Settings** и **перезапустить** Blender.

Повторно загрузить сцену по умолчанию, вызвать окно пользовательских настроек, перейти на вкладку Addons и выбрать категорию Import-Export. Отметить галочку напротив Import-Export: Blend4Web.



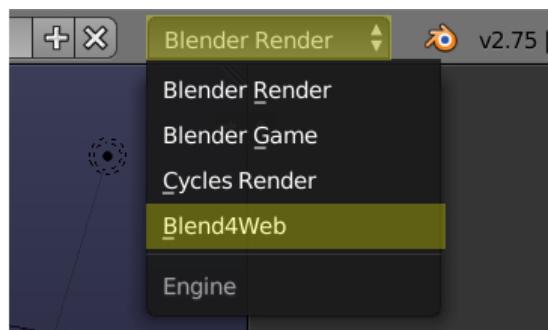
Нажать `Save User Settings`. Перезапуск Blender не требуется.

*Для проверки:*

В меню `File > Export` должны появиться опции `Blend4Web (.json)` и `Blend4Web (.html)`.

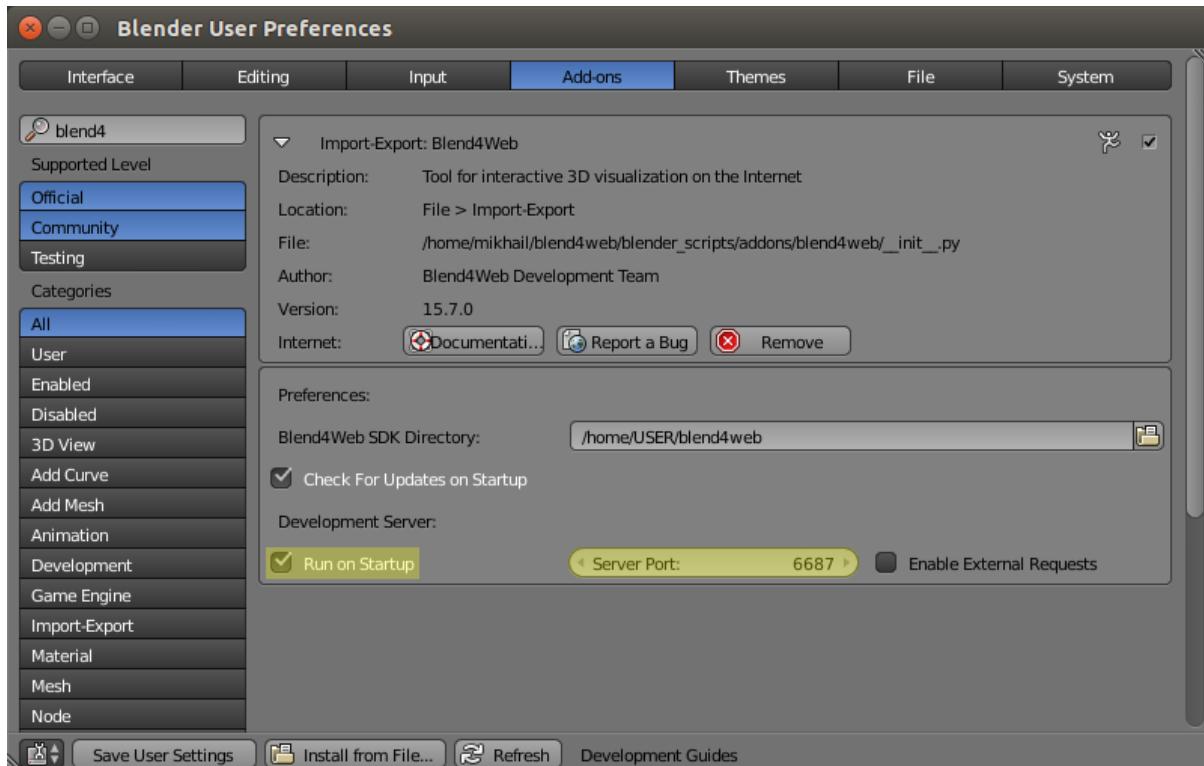
## 4.3 Профиль Blend4Web

Выберите профиль настроек Blend4Web в выпадающем меню верхней панели:



## 4.4 Локальный сервер разработки

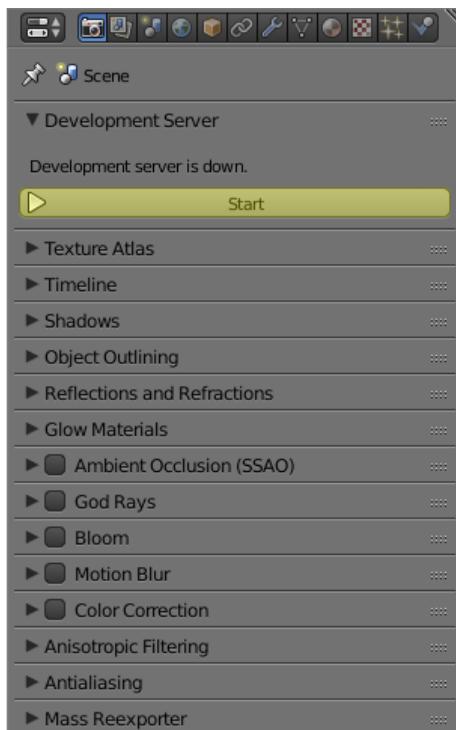
Настройки локального сервера разработки находятся в меню **File > User Preferences...** (горячие клавиши **Ctrl-Alt-U**). Здесь можно изменить номер порта, используемый при создании локального сервера разработки (6687 по умолчанию), или включить автоматический запуск сервера при запуске Blender'a: для этого достаточно включить опцию **Run on Startup** в настройках аддона.



После изменения настроек локального сервера разработчика требуется перезапуск Blender'a.

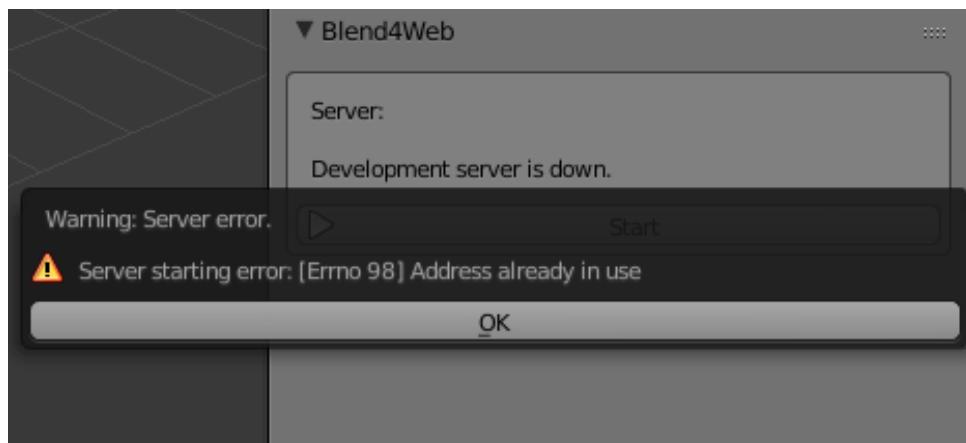
Имеется возможность сделать локальный сервер доступным из внешней сети. Для этого достаточно включить свойство **Enable External Requests**.

Если не был настроен автоматический запуск сервера, то его можно стартовать вручную по кнопке **Start** во вкладке **Render->Development Server**:



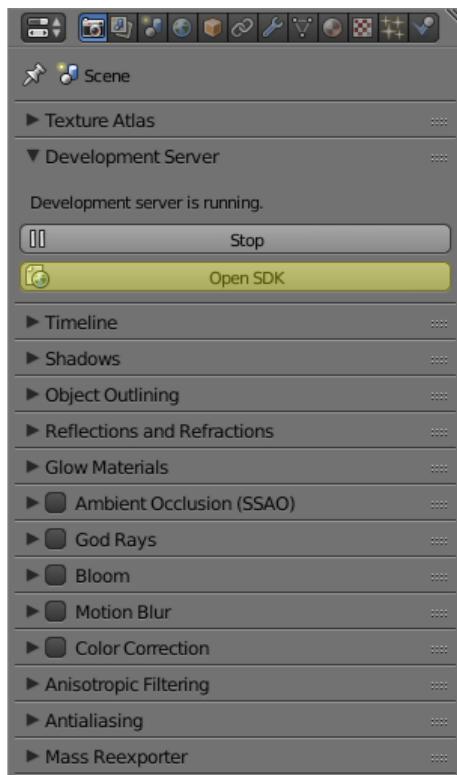
**Примечание:** Если путь к Blend4Web SDK не был указан, запуск локального сервера невозможен. При этом вместо кнопки **Start** будет выведено соответствующее сообщение.

В случае ошибки запуска сервера будет показано окно с описанием причины:



Чаще всего ошибка возникает из-за того, что указанный серверу адрес уже используется другим приложением.

Нажмите кнопку **Open SDK** для запуска в браузере обзорной веб-страницы Blend4Web SDK, доступной по адресу <http://localhost:6687>.

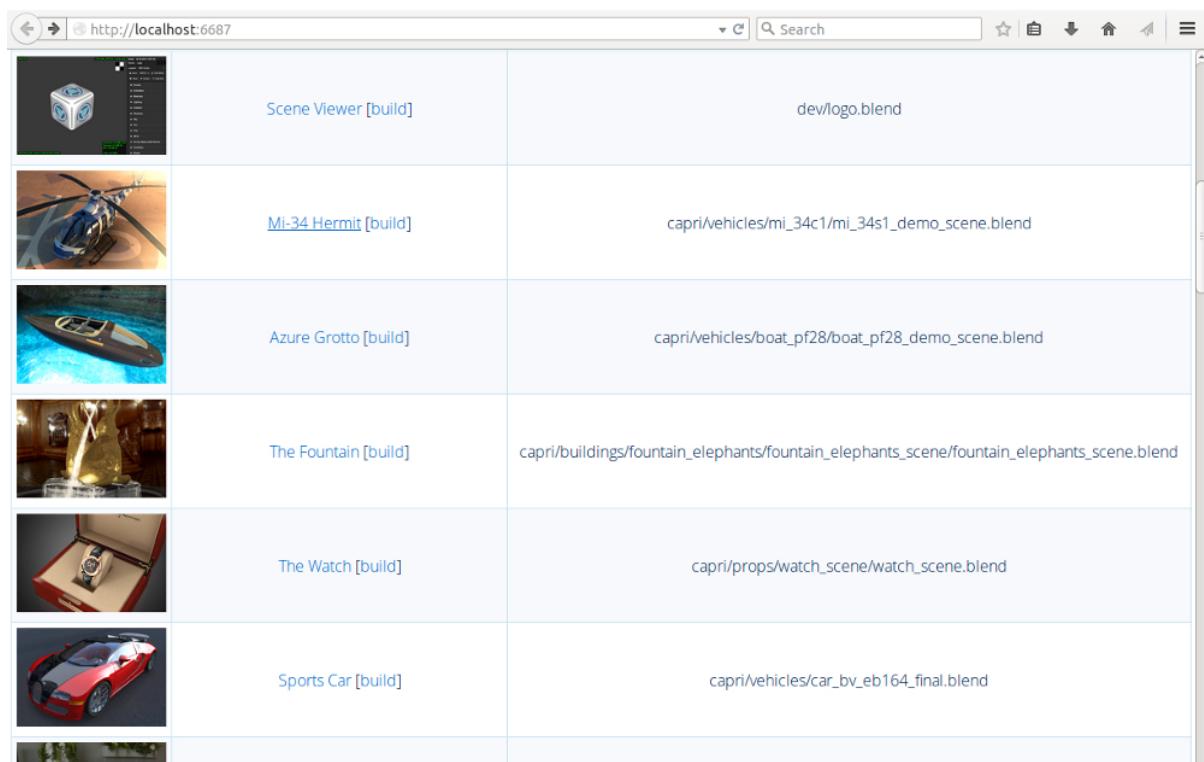


При этом будет автоматически запущен браузер, используемый в операционной системе по умолчанию.

Сервер может быть остановлен по нажатию на кнопку **Stop**. Сервер также прекращает работу при закрытии программы Blender.

## 4.5 Запуск просмотрщика сцен и демо-приложений

На обзорной странице имеются ссылки для запуска просмотрщика сцен и демо-приложений. Для их запуска необходим *браузер с поддержкой WebGL*.



**Примечание:** Если приложения SDK не отображаются корректно, или появляются сообщения об ошибках, необходимо предпринять действия, описанные в разделе *Проблемы при запуске движка*.

## Рабочий процесс

---

Создание любого продукта является творческим процессом, в котором могут участвовать множество людей, с различными навыками и опытом. Однако вне зависимости от его сложности и конечного результата, всегда можно выделить стадию производства, на которой создаётся основной объём ресурсов (ассетов) и исходного кода.

При использовании Blend4Web, производственный процесс можно представить следующим образом:

1. Подготовка трёхмерных сцен в программе Blender.
2. Экспорт сцен.
3. Запуск, настройка и отладка сцены в программе-просмотрике.
4. Создание целевого приложения.

### 5.1 Подготовка сцен

Помимо обычных операций по моделированию, текстурированию, анимации и т.д. должна быть осуществлена подготовка сцены для работы в движке.

Общие рекомендации:

1. Настоятельно рекомендуется сохранять blend-файлы в директории `blender/имя_проекта`. Там же должны находиться вспомогательные файлы, не предназначенные для загрузки в приложение (например, референсы).
2. Изображения и медиа-файлы должны быть внешними и находиться в директории `deploy/assets/имя_проекта`.
3. Файл, из которого будет осуществляться экспорт, должен содержать только необходимые в разрабатываемом приложении ресурсы (объекты, материалы, текстуры, анимацию и т.д.)
4. Составные части сцен должны иметь отличающие названия. Они не должны называться “Cube.001”, “Material”, “Armature”.

5. Рекомендуется практиковать добавление по ссылке (linking) компонентов из других файлов (библиотек).

## 5.2 Экспорт сцен

Для загрузки сцен, созданных с помощью пакета Blender, в движок, необходимо перевести их в формат, пригодный для чтения браузером. На данный момент используются текстовые файлы с расширением `.json`, в которые сохраняются экспортируемые структуры данных в формате JSON (JavaScript Object Notation). Данный файл, в свою очередь, ссылается на один бинарный файл с расширением `.bin`, содержащий массивы данных моделей, и на внешние ресурсы - текстуры и звуковые сэмплы.

Файлы `.json` и `.bin` создаются при экспорте, файлы текстур и медиа-файлы, как правило, должны быть размещены вручную. Внедренные в `.blend`-файл ресурсы автоматически размещаются рядом с экспортированными файлами.

Экспорт производится выбором в меню `File > Export` опции Blend4Web (`.json`). Быстрый доступ - поиск по `b4w export` (горячая клавиша ПРОБЕЛ).

Экспортные файлы рекомендуется размещать в директории, предназначеннной для развертывания приложения, например `deploy/assets/имя_проекта`.

Необходимо использовать относительные пути для внешних ресурсов (как правило, это происходит по умолчанию). В случаях, когда это не так, необходимо выполнить команду `File > External Data > Make All Paths Relative` (т.е. сделать все пути относительными). Использование абсолютных путей вместо относительных может приводить к ошибкам при попытках загрузки `.blend` и `.json` файлов на других компьютерах.

В момент экспорта происходит проверка сцены на предмет использования не поддерживаемых движком возможностей Blender'a. В таких случаях генерируется сообщение об ошибке. Перечень возможных ошибок экспорта перечислен в [соответствующем разделе](#).

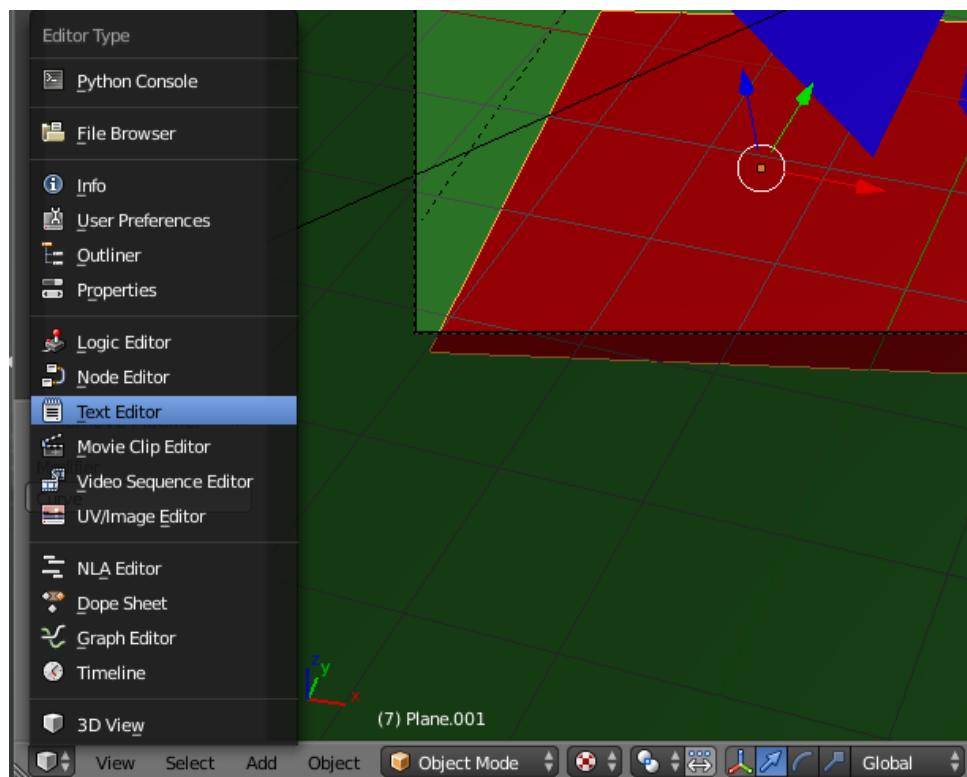
Опции экспорта подробно описаны в [соответствующем разделе](#).

## 5.3 Отображение сцен в просмотрщике

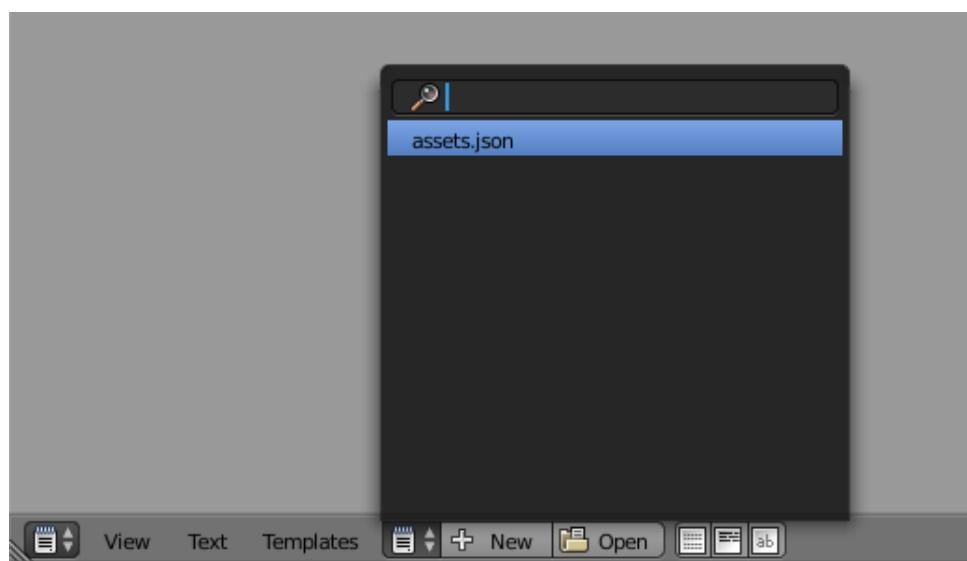
При использовании [локального сервера разработки](#) и опции экспорта `Run in Viewer` сцены показываются в просмотрщике сцен сразу после экспортации.

При необходимости долгосрочного хранения сцены в списке сцен просмотрщика, нужно вручную добавить запись в текстовой файл `apps_dev/viewer/assets.json`, который доступен при открытии Blender'a при использовании Blend4Web SDK.

Для редактирования файла откройте “Text Editor”:



В появившемся окне вы сможете выбрать файл assets.json и отредактировать его:



Для добавления новой сцены нужно знать категорию, в которой она должна отображаться. Категория обычно соответствует названию проекта и имени директории,

где хранятся соответствующие файлы.

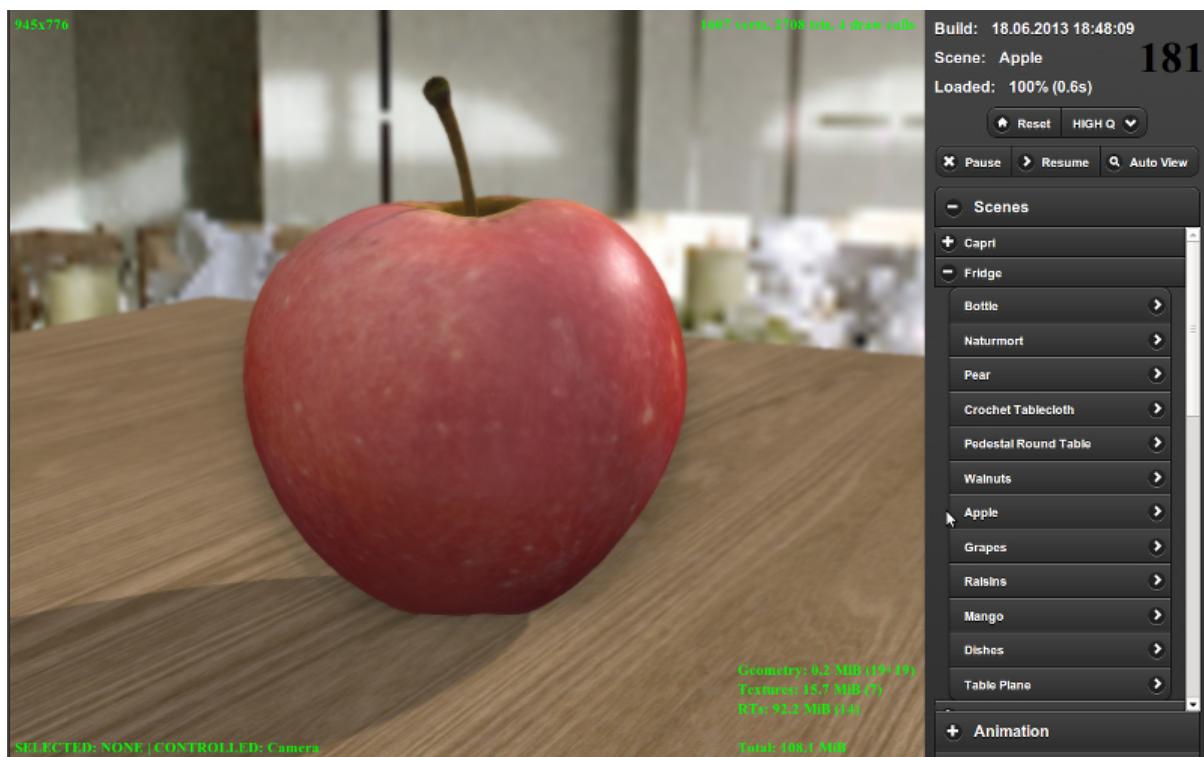
### 5.3.1 Пример

Ниже приведена примерная часть файла `assets.json`, в которой находятся два проекта “Capri” и “Fridge” с соответствующими сценами в каждом проекте:

```
{  
    name: "Capri",  
    items: [  
        {  
            name: "Baken",  
            load_file : "capri/props/baken/baken.json"  
        },  
        {  
            name: "Terrain",  
            load_file : "capri/landscape/terrain/terrain.json"  
        }  
    ]  
},  
{  
    name: "Fridge",  
    items: [  
        {  
            name: "Apple",  
            load_file : "fridge/fruits/apple/apple.json"  
        },  
        {  
            name: "Mango",  
            load_file : "fridge/fruits/mango/mango.json"  
        }  
    ]  
}
```

Добавление можно осуществить копированием и вставкой описания похожей сцены в нужной категории и последующим редактированием ее названия и пути к экспортному файлу.

В случае успешного добавления сцена должна появиться в списке сцен просмотрика в нужной категории.



## 5.4 Разработка приложения

На этой стадии создаётся приложение, в котором на языке JavaScript пишется логика по загрузке сцены и взаимодействию с ней пользователя. Документация для разработчиков приложений приведена в [соответствующем разделе](#).

## Просмотрщик сцен

---

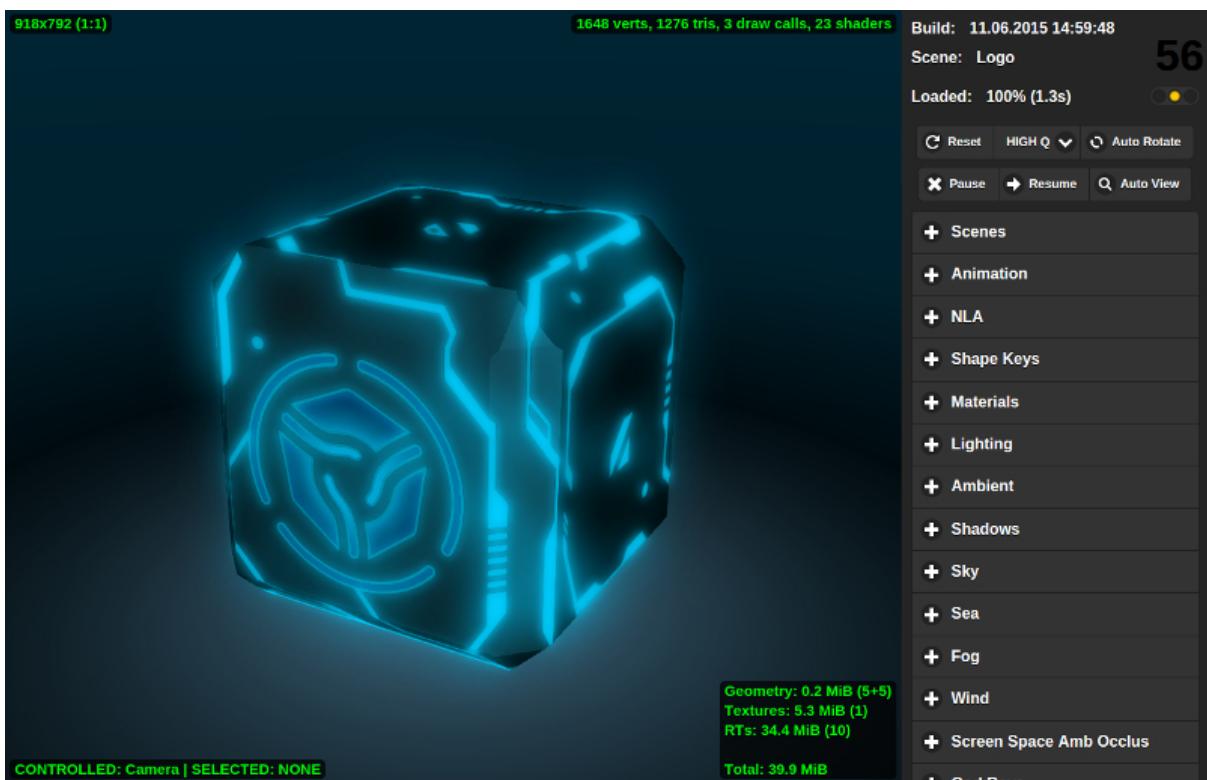
*Запуск просмотрщика сцен.*

### 6.1 Навигация

Управление камерой осуществляется мышью с нажатой кнопкой, а также клавишами W, A, S, D, R, F: вперед, влево, назад, вправо, вверх, вниз. Также поддерживаются стрелки и клавиши `numPad`. В режиме камеры `Target` возможно фокусирование на выделенный объект посредством нажатия клавишей Z или . (точка).

### 6.2 Боковая панель

Боковая панель содержит в себе три области: информационное табло, базовые кнопки управления и список выпадающих панелей, содержащий дополнительные элементы управления, разделённые по функциональному признаку.



### 6.2.1 Список элементов управления сверху вниз

**Build** Дата и время сборки движка. В версии для разработки показывает время загрузки страницы.

**Scene** Название загруженной сцены, взятое из файла assets.json. При наведении курсора мыши всплывает путь к файлу.

**Loaded** Процент и время загрузки.

**Reset** Кнопка удаляет сохраненное название последней просмотренной сцены и перезагружает страницу, возвращая к сцене по умолчанию.

**LOW Q - HIGH Q - ULTRA Q** Выпадающее меню выбора профиля работы движка.

**См.также:**

*Профили качества изображения*

**Pause** Приостановка рендеринга.

**Resume** Возобновление рендеринга.

**Auto View** Активация режима автоматического переключения сцен по списку, пауза между просмотрами 1 секунда.

**Scenes** Двухуровневый список категорий и сцен из файла `assets.json`.

**Animation** Управление анимацией. При просмотре анимированных моделей можно выбирать объект и переключать для него анимацию с помощью выпадающего меню, включать режим циклической анимации, останавливать и возобновлять анимацию, выставлять нужный кадр (при этом анимация должна быть остановлена).

**Materials** Настройка свойств материалов. Материал выбирается с помощью выпадающего меню. В настоящий момент поддерживается изменение ограниченного ряда свойств.

**Lighting** Настройка параметров прямого освещения. Источник света выбирается с помощью выпадающего меню. Поддерживается изменение цвета и интенсивности. Также на этой панели осуществляется настройка параметров времени суток и солнечного освещения.

**Ambient** Настройка параметров освещения от окружающей среды. Поддерживается изменение цветов в полусферической модели освещения и интенсивности.

**Shadows** Настройка параметров падающих теней, включая параметры каскадов теней и параметры смягчения краев теней.

**Sky** Настройка параметров динамического неба, таких как цвет, параметры рассеивания солнечного света и т.д.

**Sea** Настройка параметров рендеринга воды, включая переходы цвета по глубине и удаленности от берега, параметры пены, подповерхностного рассеивания, динамики волн и т.д.

**Fog** Настройка параметров тумана, включая плотность и цвет.

**Wind** Настройка параметров ветра, включая направление и силу.

**Screen Space Amb Occlus** Настройка параметров взаимного затенения.

**God Rays** Настройка параметров эффекта сумеречных лучей.

**Bloom** Настройка параметров эффекта засветки ярким светом.

**Depth of Field** Настройка параметров эффекта глубины резкости камеры.

**Color correction** Настройка параметров цветовой коррекции, включая яркость, контрастность, экспозицию и насыщенность.

**Anti-aliasing** Выбор метода сглаживания изображения (антиалиасинга).

**Audio** На панели находится переключатель режима микширования. После включения на сценах со звуковыми источниками появляется интерфейс микшера.

**Stereo View** На панели находится переключатель режима стерео-изображения.

**Debug** На панели находится ряд инструментов отладки сцен, в частности переключатель режима просмотра сетки, переключатель режима просмотра постпроцессинговых стадий.

## 6.3 Индикаторы

**Счетчик количества кадров в секунду** Находится в правом верхнем углу. Выводит усредненное и округленное значение за последние 1,5 секунды.

**Размер области рендеринга** Находится в левом верхнем углу. Выводит размер области рендеринга в пикселях.

**Выбранный объект и контролируемый объект** Находится в левом нижнем углу. Выводит название выбранного объекта и контролируемого объекта. Выбор объекта осуществляется мышью. Для получения прямого контроля над объектом (обычно в целях проверки физики) нужно нажать Q и выбрать объект. Движение объекта осуществляется клавишами W, A, S, D. Для выхода из режима контроля нужно нажать Q и “кликнуть” на пустом пространстве. Индикатор также показывает расстояние до выбранного объекта в единицах Blender'a (эквивалент метра).

**Индикатор сложности сцены** Находится в правом верхнем углу области рендеринга. Выводит количество вершин, треугольников и WebGL вызовов на основной рендер-сцене (т.е. не учитываются, например, вызовы по отрисовке динамических теней).

**Индикатор видео-памяти** Находится в правом нижнем углу области рендеринга. Выводит количество видео-памяти, занимаемой геометрией, текстурами, буферами с результатами рендеринга (render targets), а также суммарное количество занимаемой памяти.

**Индикатор ошибок при загрузке сцены** Находится под счетчиком количества кадров. Выводит ошибки и предупреждения при загрузке сцены. Красный свет означает ошибки, желтый - предупреждения, зеленый - загрузка сцены прошла успешно.



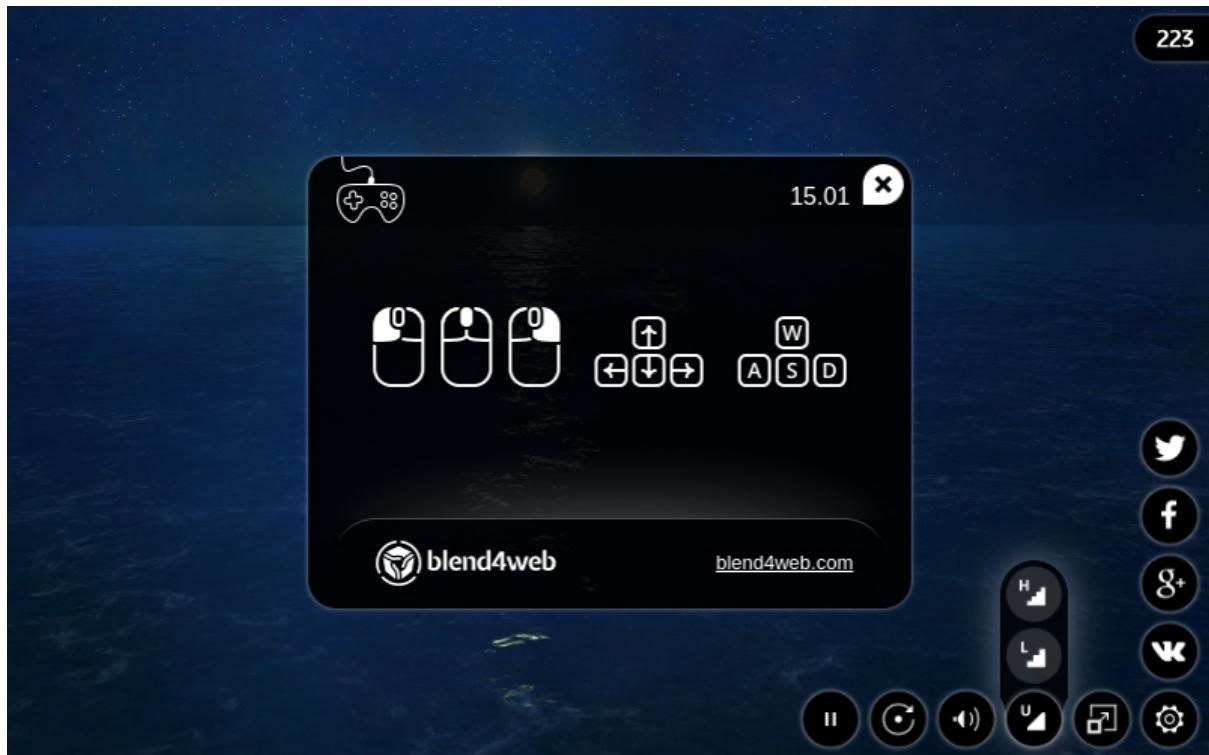
---

## Веб-плеер

---

Веб-плеер представляет собой специализированное приложение для отображения моделей и сцен в режиме демонстрации.

Десктопная версия:



Мобильная версия:



## 7.1 Использование

Директорию с файлами веб-плеера `deploy/apps/webplayer` можно скопировать из дистрибутива Blend4Web SDK и разместить на сайте. Полученные в результате экспорта файлы сцен можно разместить на сайте и указать путь к ним (абсолютный или относительный) с помощью параметра веб-плеера `load`.

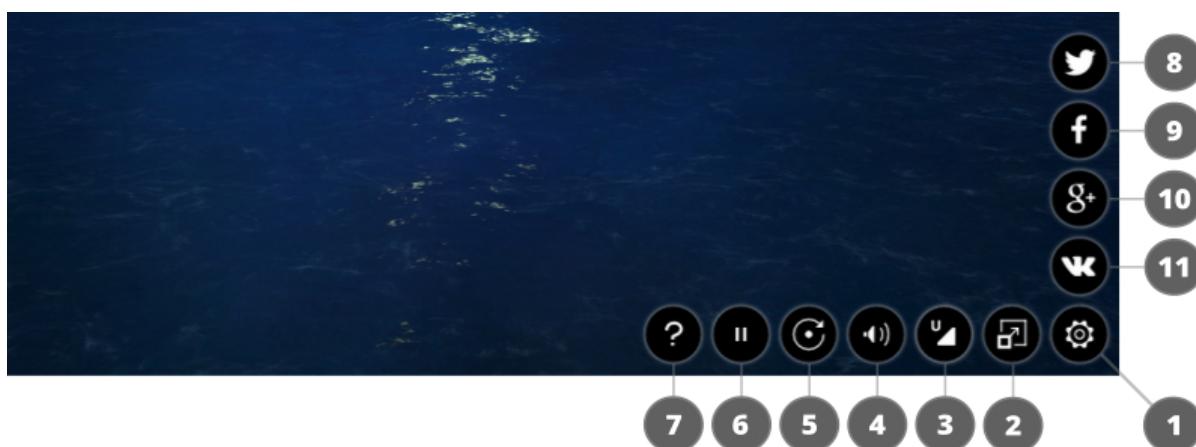
При экспорте в единый HTML файл интерфейс веб-плеера интегрируется автоматически.

## 7.2 Навигация

Управление камерой (в режимах `Target` и `Eye`) осуществляется мышью с нажатой кнопкой, а также клавишами `W`, `A`, `S`, `D`, `R`, `F`: вперед, влево, назад, вправо, вверх, вниз. Также поддерживаются стрелки и клавиши `numpad`.

## 7.3 Панель управления

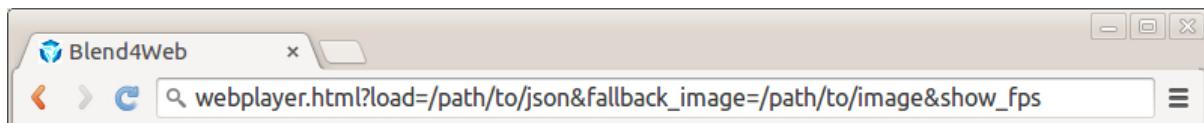
Ниже показана панель управления веб-плеера.



1. показ / скрытие панели управления;
2. включение / выключение полноэкранного режима;
3. установка качества сцены;
4. включение / выключение звука;
5. включение / выключение автоматического поворота камеры вокруг сцены;
6. запуск / остановка движка;
7. открытие окно помощи;
8. кнопка Твиттер;
9. кнопка Facebook;
10. кнопка Google+;
11. кнопка ВКонтакте.

## 7.4 Атрибуты

Веб-плеер принимает атрибуты из адресной строки браузера:



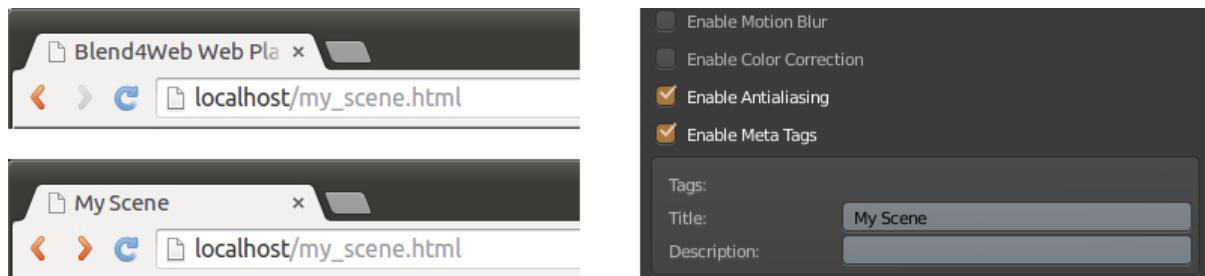
1. для загрузки сцены используется обязательный атрибут `load`, в который помещается относительный путь к JSON файлу.
2. необязательный атрибут `fallback_image` используется для установки фона, заменяющего сцену при возникновении ошибки WebGL.

3. необязательный атрибут `fallback_video` используется для установки видео, заменяющего сцену при возникновении ошибки WebGL. Может использоваться неоднократно с целью добавления видео файлов различных форматов.
4. необязательный атрибут `show_fps` применяется для отображения счетчика кадров в секунду в правом верхнем углу плеера.
5. необязательный атрибут `autorotate` используется для включения автоматического вращения камеры сразу после загрузки сцены.
6. необязательный атрибут `compressed_textures` используется для включения загрузки уменьшенных текстур и текстур в формате с компрессией (DDS).
7. необязательный атрибут `alpha` используется для установки прозрачности области отрисовки.

**Примечание:** Если указаны параметры `fallback_image` и `fallback_video`, приоритет имеет `fallback_image`.

## 7.5 Название сцены в заголовке

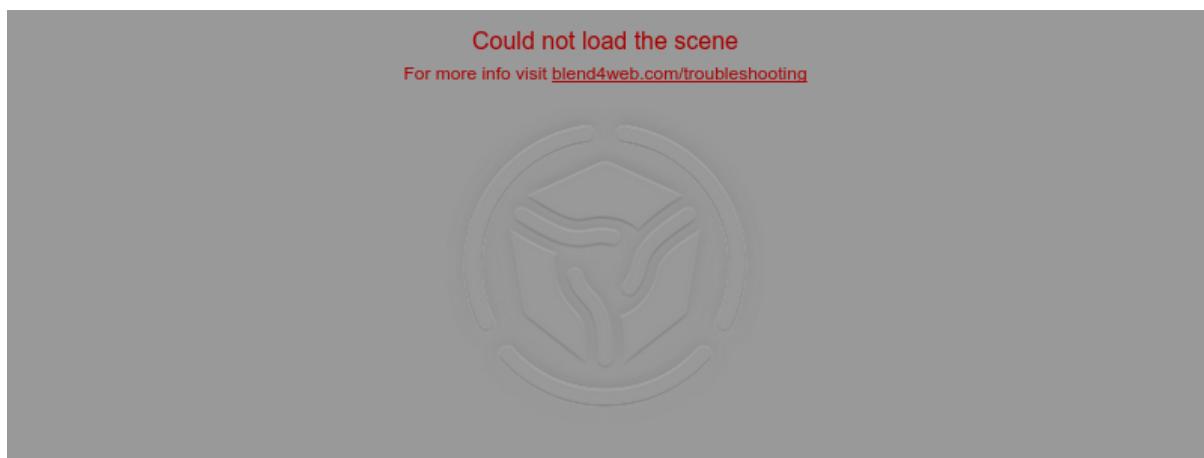
По умолчанию веб-плеер устанавливает тег `title` равным `Blend4Web Web Player`. С помощью мета-тега `title` в настройках сцены можно изменить значение по умолчанию.



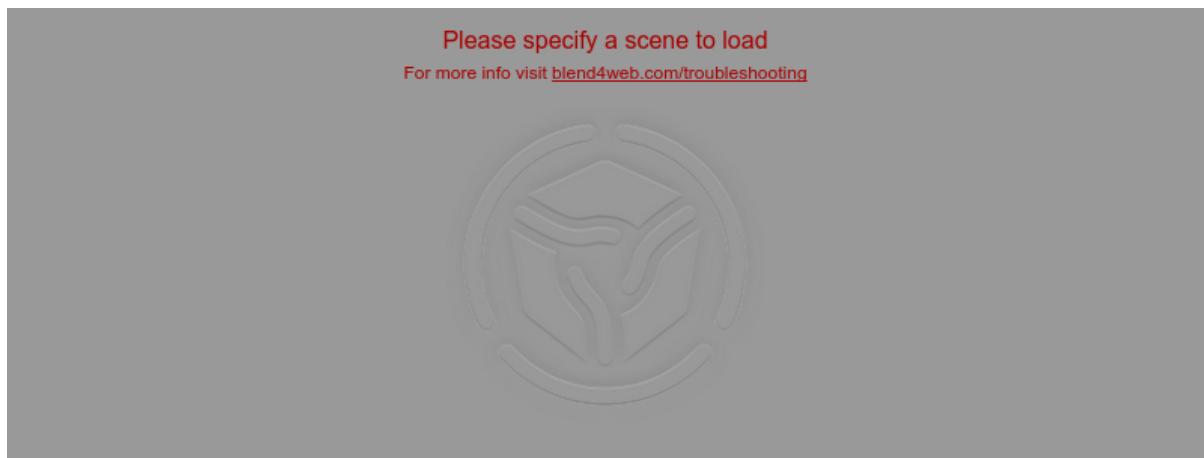
## 7.6 Ошибки сцены

В случае некорректного использования веб-плеера выводятся сообщения об ошибках.

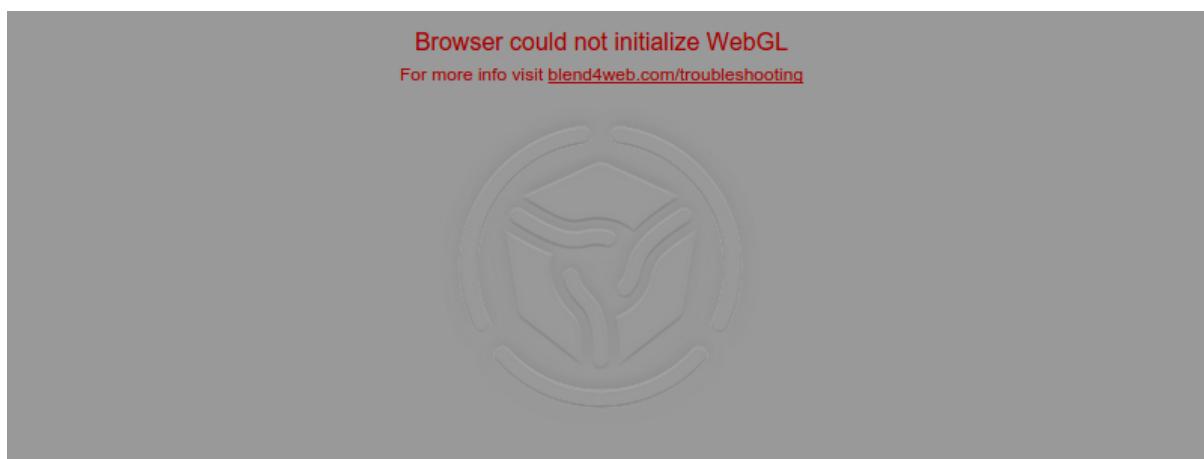
1. атрибутом `load` указан неверный путь к JSON файлу или данный файл неисправен;



2. атрибут `load` не найден или его значение пустое;



3. ошибка инициализации WebGL.





---

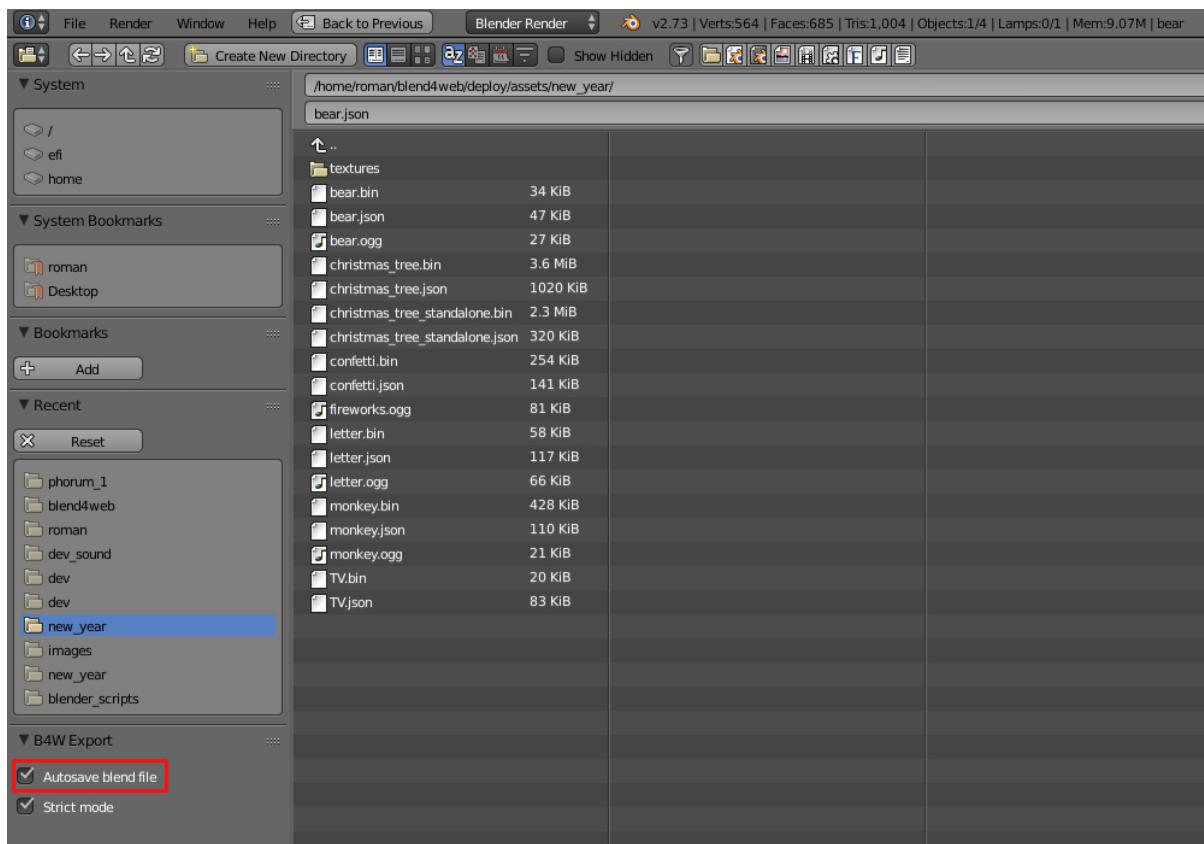
## Аддон

---

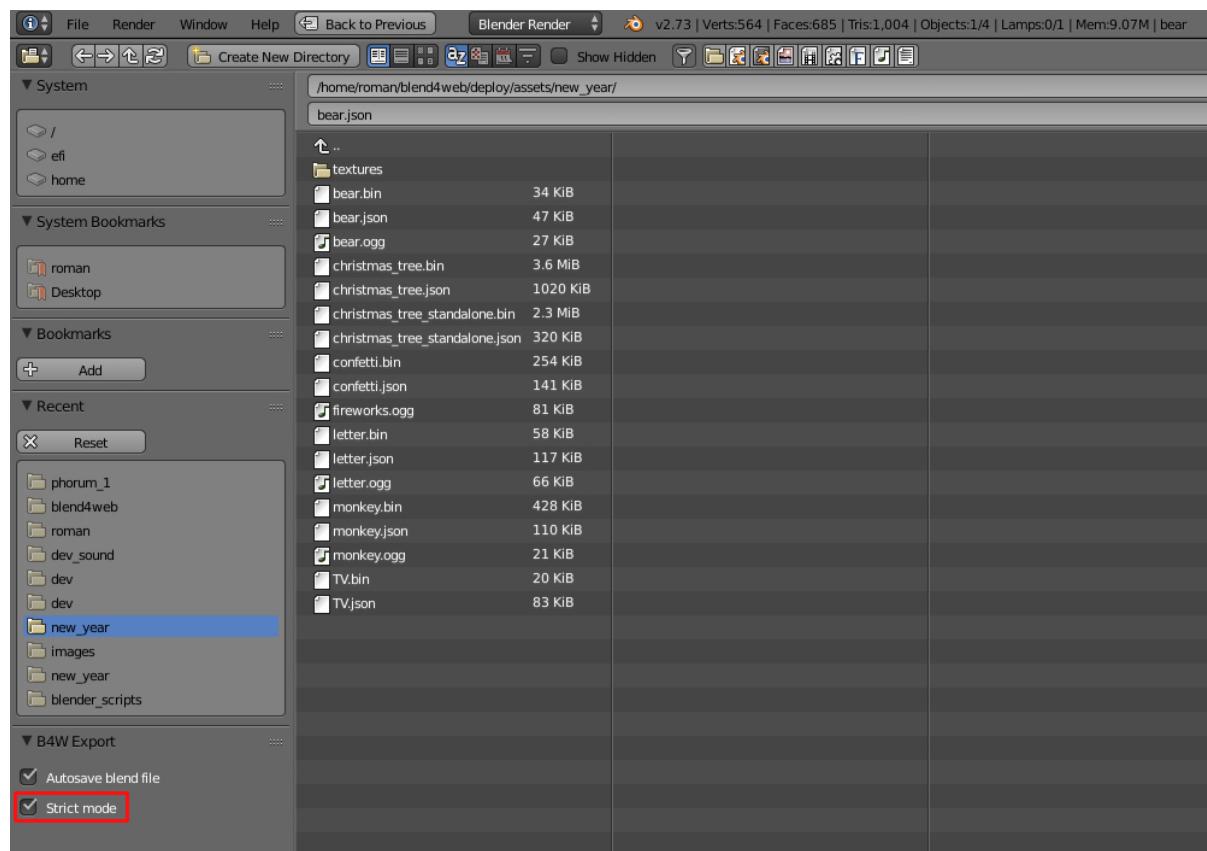
### 8.1 Опции экспорта

**Autosave blend File** Автосохранение файла, из которого осуществляется экспорт.

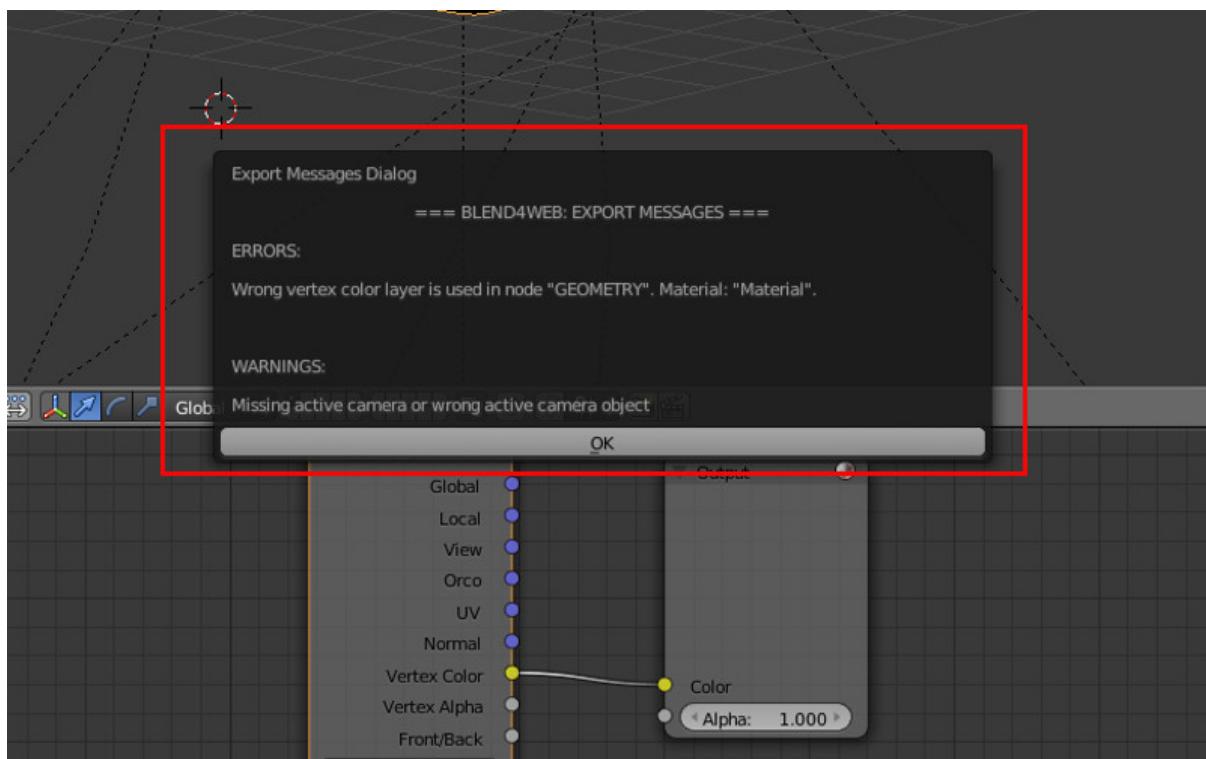
**Включено по умолчанию.** Осуществляется непосредственно после экспорта с целью поддержки соответствия между текущим содержимым blend-файла и экспортного файла. Кроме того, для удобства в blend-файле сохраняется относительный путь к экспортному файлу.



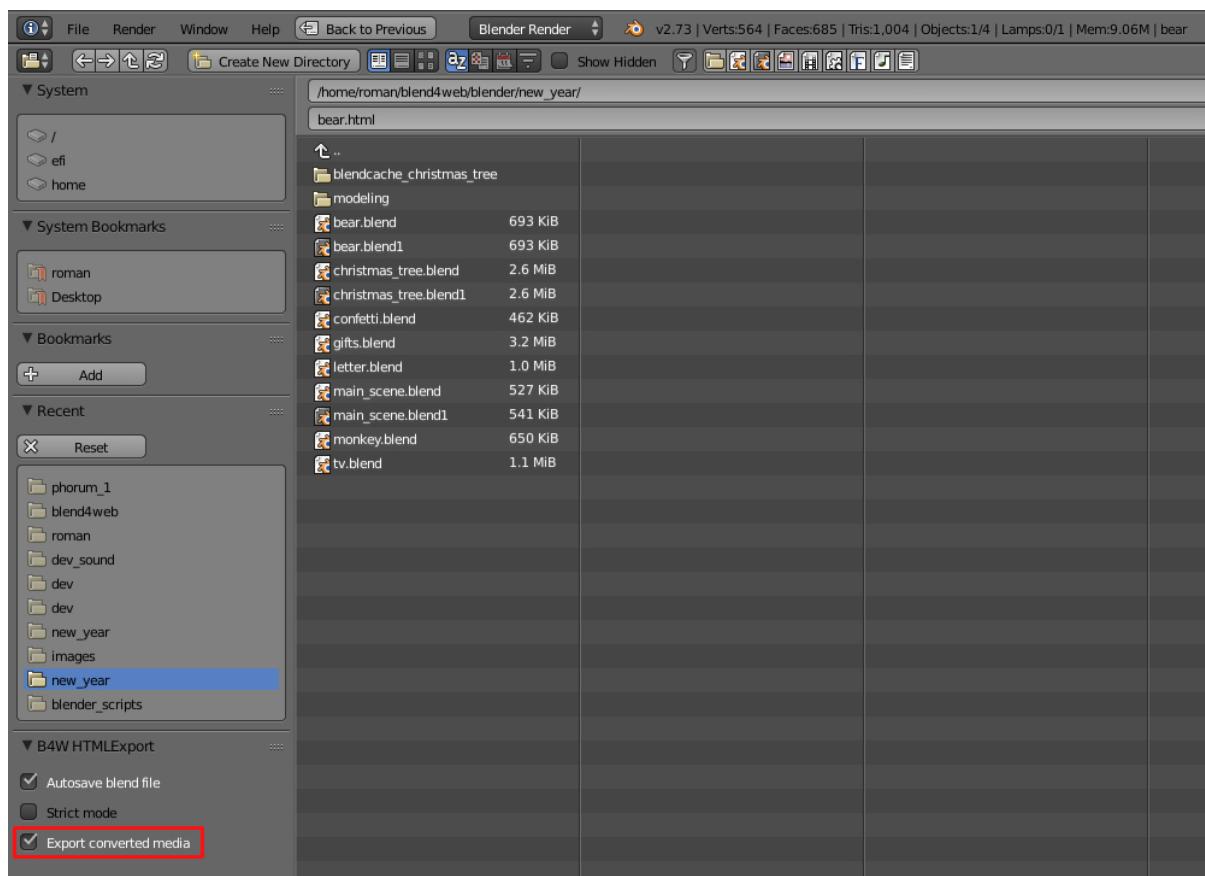
**Strict Mode** Данный режим блокирует экспорт при наличии ошибок и сообщений, требующих внимания пользователя. Режим включается при выставлении опции **Strict Mode** в меню экспорта:



При наличии некритических ошибок экспорта или сообщений, требующих внимания пользователя, вашему вниманию будет представлено диалоговое окно вида:



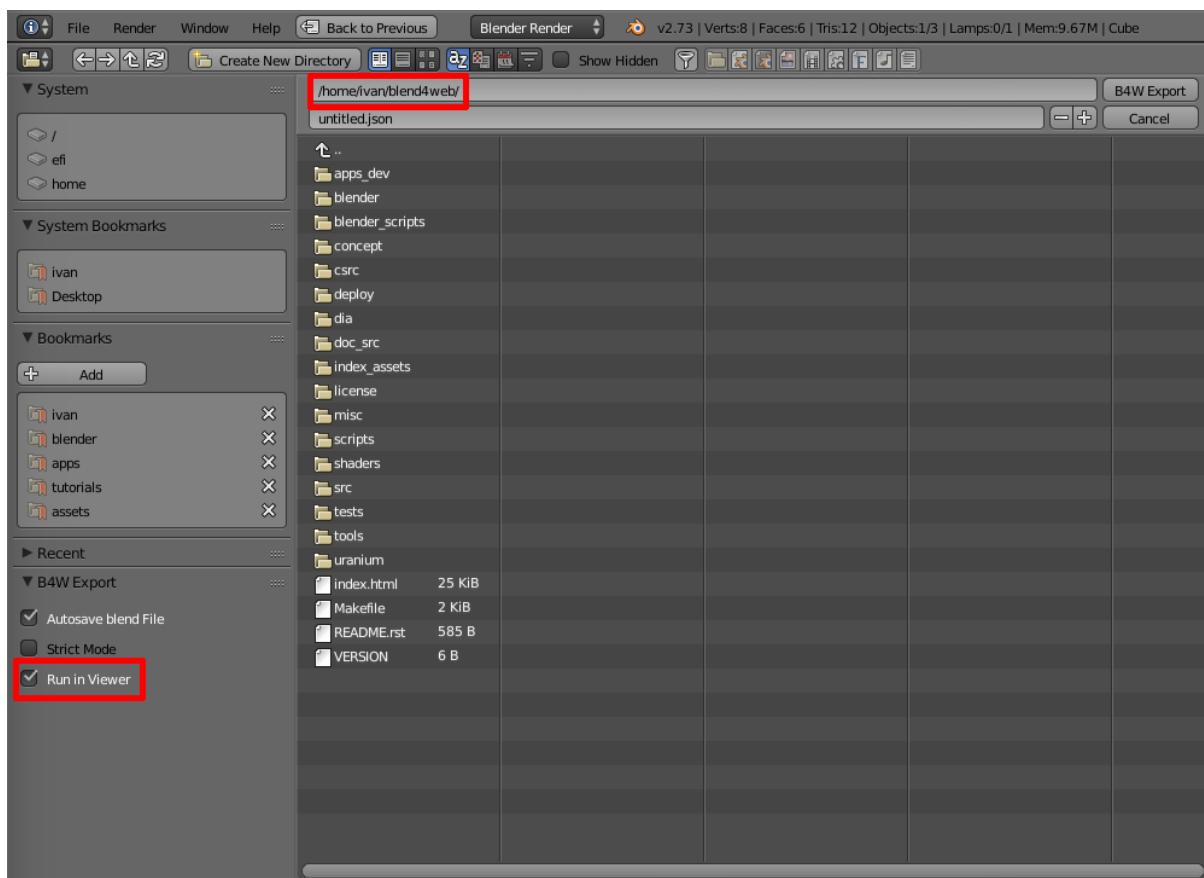
**Export Converted Media** Опция доступна при html-экспорте. Включение данного режима экспорта позволяет записать в HTML файл конвертированные медиафайлы разных форматов. Это необходимо использовать при создании кросбраузерных и кросплатформенных приложений при html-экспорте. При этом в html-файл будут записываться файлы, созданные с использованием [нашего конвертера](#).



**Run in Viewer** Автоматически запустить просмотрщик сцен и добавить в него экспортируемую сцену.

При использовании *локального сервера разработки*, имеется возможность открыть сцену, экспортированную в формате .json, в просмотрщике сцен. Для этого при экспорте необходимо выбрать любой путь, лежащий внутри файловой структуры Blend4Web SDK.

В качестве директории для экспорта может использоваться созданная пользователем директория внутри Blend4Web SDK. При несоблюдении этого условия опция не будет отображаться в меню экспорта. Также опция не будет отображаться если локальный web-сервер не запущен.



## 8.2 Ошибки инициализации

Ошибки инициализации могут проявляться при инициализации аддона, либо при загрузке сцены в Blender. При возникновении появится диалоговое окно с описанием ошибки.



Сообщение об ошибке	Причина
Blend4Web initialization error! Addon is not compatible with the PLATFORM platform.	Ошибка загрузки аддона. Аддон не совместим с платформой PLATFORM.
Warning: Blender version mismatch. Blender VER_REQUIRED is recommended for the Blend4Web addon. Current version is VER_CURRENT.	Предупреждение о возможной несовместимости с текущей версией Blender. Для работы рекомендуется версия Blender'a VER_REQUIRED. Текущая версия - VER_CURRENT.

## 8.3 Ошибки совместимости

Ошибки совместимости версий могут проявиться при просмотре сцены в браузере, если версия аддона, которым был произведен экспорт сцены, не соответствует версии движка Blend4Web, который пытается эту сцену загрузить, а также если .bin-файл не соответствует .json-файлу.

Сообщение об ошибке	Причина
JSON version is too old relative to B4W engine: VER_OLD, required: VER_NEW. Reexport scene with the latest B4W addon to fix it.	Версия аддона, которым была экспортирована сцена, сильно устарела: VER_OLD, движком требуется - VER_NEW. Сцена не будет загружена. Рекомендуется переэкспортировать сцену аддоном последней версии, а также использовать последнюю версию движка.
JSON version is a bit old relative to B4W engine: VER_OLD, required: VER_NEW. Some compatibility issues can occur. Reexport scene with the latest B4W addon to fix it.	Версия аддона, которым была экспортирована сцена, немного устарела: VER_OLD, движком требуется - VER_NEW. Сцена будет загружена, однако возможны различные ошибки. Рекомендуется переэкспортировать сцену аддоном последней версии, а также использовать последнюю версию движка.
B4W engine version is too old relative to JSON. Can't load the scene. Update your engine version to fix it.	Версия движка сильно устарела по сравнению с версией аддона, которым был произведен экспорт сцены. Сцена не будет загружена. Рекомендуется использовать последнюю версию движка и аддона.
B4W engine version is a bit old relative to JSON. Some compatibility issues can occur. Update your engine version to fix it.	Версия движка немного устарела по сравнению с версией аддона, которым был произведен экспорт сцены. Сцена будет загружена, однако возможны различные ошибки. Рекомендуется использовать последнюю версию движка и аддона.
BIN version does not match to JSON version: VER_BIN, required: VER_JSON. Couldn't load the scene. Reexport scene to fix it.	Версия загружаемого .bin-файла сильно устарела по сравнению с .json-файлом: VER_BIN, версия .json-файла: VER_JSON. Сцена не будет загружена. Рекомендуется переэкспортировать сцену.
BIN version does not match to JSON version: VER_BIN, required: VER_JSON. Some compatibility issues can occur. Reexport scene to fix it.	Версия загружаемого .bin-файла немного устарела по сравнению с .json-файлом: VER_BIN, версия .json-файла: VER_JSON. Могут возникнуть ошибки несовместимости. Рекомендуется переэкспортировать сцену.

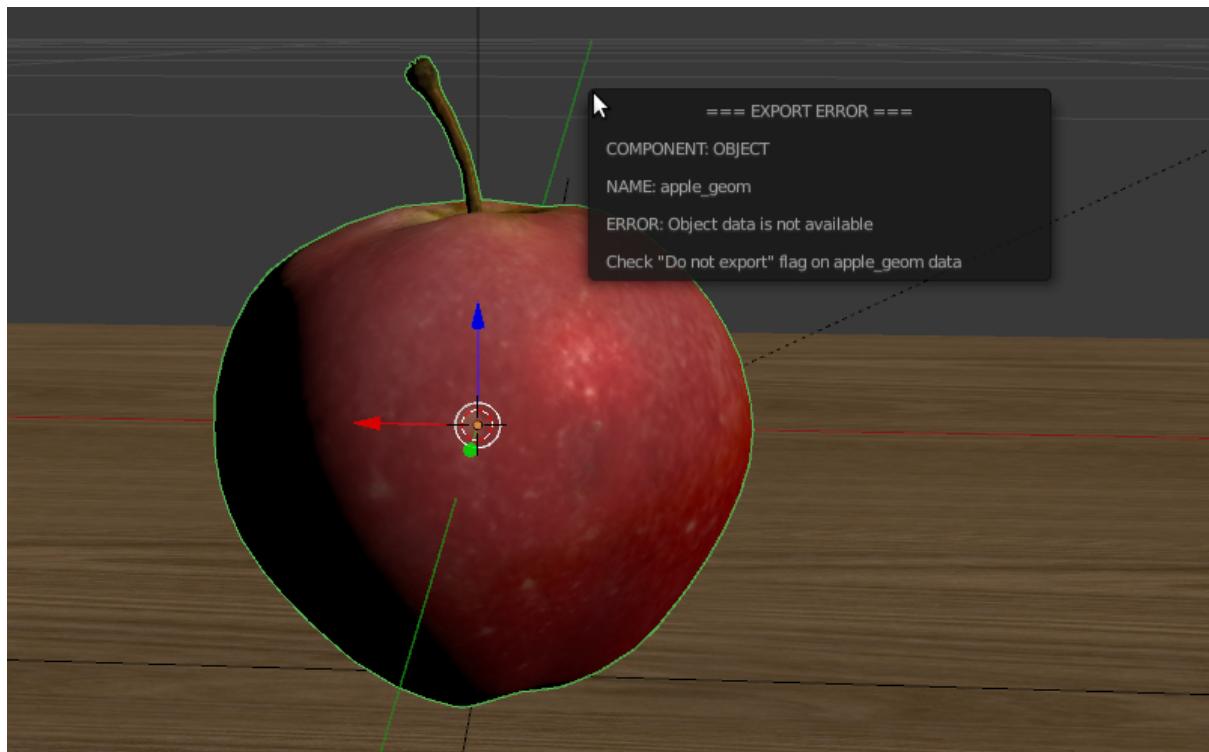
## 8.4 Критические ошибки экспорта

При возникновении ошибок во время экспорта появляется диалоговое окно BLEND4WEB EXPORT ERROR с описанием проблемы:

COMPONENT - тип компонента (объект, меш, материал, текстура и т.д.), при экспорте которого произошла ошибка.

NAME - имя компонента.

ERROR - краткое описание возникшей проблемы на англ. языке.



Сообщение об ошибке	Причина
Dupli group error; Objects from the GROUP_NAME dupli group on the OBJECT_NAME object cannot be exported	Ни один из объектов группы GROUP_NAME, выбранной для дублирования на объекте OBJECT_NAME, не экспортируется. Требуется разрешить экспорт хотя бы одного из объектов группы, либо убрать дублирование группой.
Export to different disk is forbidden	Не разрешен экспорт в директорию, находящуюся на другом диске
Incompatible objects with a shared mesh; The OBJECT_NAME object has both vertex groups and a shared mesh	Несовместимые объекты с общим мешем. Не допускается экспорт объекта с общим мешем и вертексными группами. Исключения: экспорт возможен, если на объекте включены опции <code>Apply modifiers</code> , <code>Export vertex animation</code> , <code>Export edited normals</code> , <code>Apply scale</code> (т.к. в этом случае при экспорте происходит полное копирование мешей).
Incomplete mesh; Material slot is empty	Неполный меш: пустой слот материала.
Incomplete vehicle. The NAME vehicle doesn't have any chassis or hull	Моделируемое средство передвижения NAME является незавершенным: оно должно содержать один элемент <code>Chassis</code> или <code>Hull</code> .
Incomplete vehicle. The NAME vehicle requires at least one bob	Моделируемое средство передвижения NAME является незавершенным: оно должно содержать хотя бы один элемент <code>Bob</code> .
Incomplete vehicle. The NAME vehicle requires at least one wheel	Моделируемое средство передвижения NAME является незавершенным: оно должно содержать хотя бы один элемент <code>wheel</code> .
Incorrect mesh; Corrupted file: Wrong group indices	Меш содержит вершины, привязанные к несуществующей группе.
Incorrect mesh; Corrupted file: Wrong vertice positions	Поврежденный файл: некорректное значение координаты вертекса.
Incorrect mesh; Corrupted file: Wrong normals	Поврежденный файл: некорректное значение нормали.
Incorrect mesh; Corrupted file: Wrong tangents	Поврежденный файл: некорректное значение тангенты.
Incorrect mesh; Corrupted file: Wrong texture coordinates	Поврежденный файл: некорректное значение текстурной координаты.
Incorrect mesh; Corrupted file: Wrong vertex group weights	Поврежденный файл: некорректное значение веса вертекса в вертексной группе.

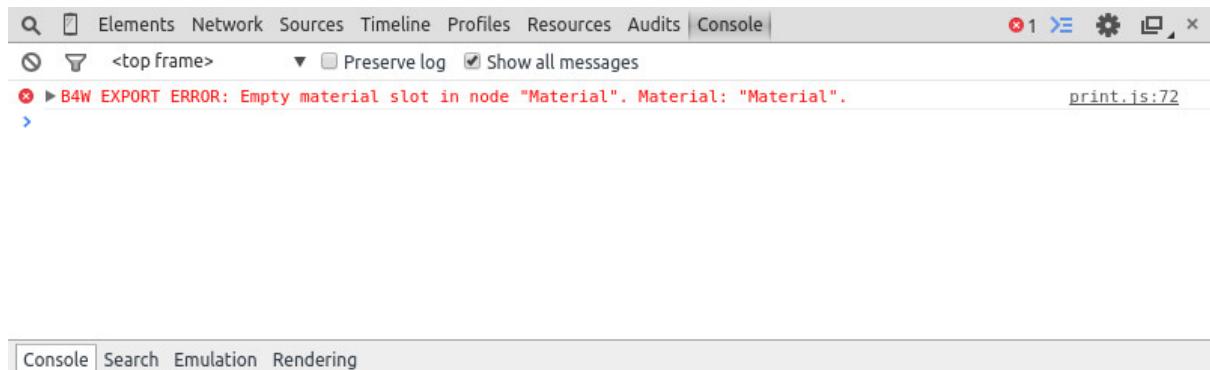
Incorrect mesh; Corrupted file; Wrong vertex color values	Поврежденный файл: некорректное значение вертексного цвета.
Incorrect vertex animation; Object has no vertex animation	Включен экспорт вертексной анимации для объекта, но ни одной анимации не имеется.
Incorrect vertex animation; Unbaked “ANIM_NAME” vertex animation	Включен экспорт вертексной анимации для меша, но анимация ANIM_NAME не содержит ни одного кадра.
Loading of resources from different disk is forbidden	Не разрешен экспорт ресурсов из директории, находящейся на другом диске.
The material has a normal map but doesn't have any material nodes	Нодовый материал использует Normal Mapping, но не имеет ноды Material.
The mesh has a UV map but has no exported material	Меш имеет текстурную развертку, но не имеет материала, который бы экспортовался.
The mesh has a vertex color layer but has no exported material	Меш имеет слой вертексного цвета, но не имеет материала, который бы экспортовался.
No such file or directory	Данная директория не существует.
Object constraint has no target	Для ограничителя объекта (вкладка Object Constraints) не установлено свойство Target Object.
Particle system error; Dupli group isn't specified	Ошибка системы частиц. Не выбрана группа, используемая в качестве частицы.
Particle system error; Dupli object isn't specified	Ошибка системы частиц. Не выбран объект, используемый в качестве частицы.

Particle system error; Dupli object OBJECT_NAME doesn't export	Ошибка системы частиц. Объект OBJECT_NAME, выбранный в качестве частицы, не экспортируется (на нем выбрана опция Do not export).
Particle system error; The GROUP_NAME dupli group contains no valid object for export	Ошибка системы частиц. Ни один подходящий объект из группы GROUP_NAME, выбранной в качестве частицы, не экспортируется. Либо на таких объектах выбрана опция Do not export, либо объекты имеют неподходящий тип. Поддерживаемые типы: MESH.
Particle system error. Unsupported render type TYPE for the EMITTER/HAIR particles PSYS_NAME on object NAME. Particle system removed.	Ошибка системы частиц. Неподдерживаемый тип рендера TYPE для EMITTER/HAIR системы частиц с именем PSYS_NAME, находящейся на объекте NAME. Система частиц удалена.
Particle system error; Wrong dupli object type TYPE_NAME	Ошибка системы частиц. В качестве частицы выбран объект неподходящего типа. Поддерживаемые типы: MESH.
Permission denied	Нет прав доступа к текущей директории.
Wrong edited normals count; It doesn't match with the mesh vertices count	Число редактируемых нормалей не совпадает с числом вершин меша. Требуется сделать Clean Up либо Save в панели B4W Vertex Normals Editor.
Wrong overridden bounding box; Check the mesh's bounding box values	Указаны неверные размеры при переопределении BoundingBox для меша: минимальное значение больше максимального для хотя бы одного из измерений.
Wrong vertex animation vertices count; It doesn't match with the mesh vertices count for "ANIM_NAME"	Включен экспорт вертекской анимации, но число вершин покадрово в анимации ANIM_NAME не совпадает с числом вершин меша. Возможное решение - "перезапекание" анимации.

## 8.5 Некритические ошибки экспорта

В отличие от критических ошибок экспорта, рассмотренных ранее, данные ошибки не препятствуют экспорту, однако могут приводить к некорректному отображению сцен. Сообщения выводятся в консоли браузера (горячая клавиша F12) при загрузке сцены. Сообщение имеет вид:

B4W EXPORT ERROR: Сообщение об ошибке



Сообщение об ошибке	Причина
Canvas texture ID NAME already exists. Texture NAME.	Данный идентификатор для объекта типа <b>Canvas</b> уже существует.
Empty canvas texture ID for texture NAME.	Пустое поле идентификатора для объекта типа <b>Canvas</b> .
Empty material slot in node "NAME". Material: "NAME".	Не задан материал в ноде: "NAME"
Environment map in the "NAME" world texture slot cannot be a movie.	Карта окружения не может быть видео.
Ignoring LODs after empty LOD for the NAME object.	В списке LOD объектов, настроенных для объекта NAME, были проигнорированы все LOD объекты, следующие за пустым.
Incomplete mesh NAME; Dynamic grass vertex colors required by material settings	Неполный меш: специальный материал для ландшафта использует опции <b>Dynamic grass size</b> и/или <b>Dynamic grass color</b> , но у меша нет слоев вертексного цвета с такими именами.
Incomplete mesh; Material settings require vertex colors	Неполный меш: материал меша имеет включенную опцию вертексного цвета ( <b>Vertex Color Paint</b> ), но у меша нет слоя вертексного цвета.
Incorrect NLA script, falling back to simple sequential NLA.	Некорректный NLA-скрипт. Вместо него будет использоваться стандартная NLA-анимация.
Invalid link found in node material. Material: "NAME".	Нодовый материал с именем "NAME" содержит некорректные связи между нодами.
No image in the NAME texture. [Material: NAME.]	У текстуры отсутствует изображение.
No texture for the NAME particle settings texture slot.	В текстурном слоте системы частиц отсутствует текстура.
No texture in the NAME world texture slot.	В текстурном слоте объекта <b>World</b> отсутствует текстура.
No texture in the texture slot. Material: NAME.	В текстурном слоте материала отсутствует текстура.
Node material invalid: "NAME". Check sockets compatibility: "FROM_NODE" with "TO_NODE".	Ошибка нодового материала. Типы входа и выхода связи между нодами <b>FROM_NODE</b> и <b>TO_NODE</b> не соответствуют друг другу.
Object "NAME" hasn't renderable data. Converted to EMPTY.	Объект с именем "NAME" является вырожденным, например, не имеет полигонов, поэтому его тип изменён на <b>EMPTY</b> .

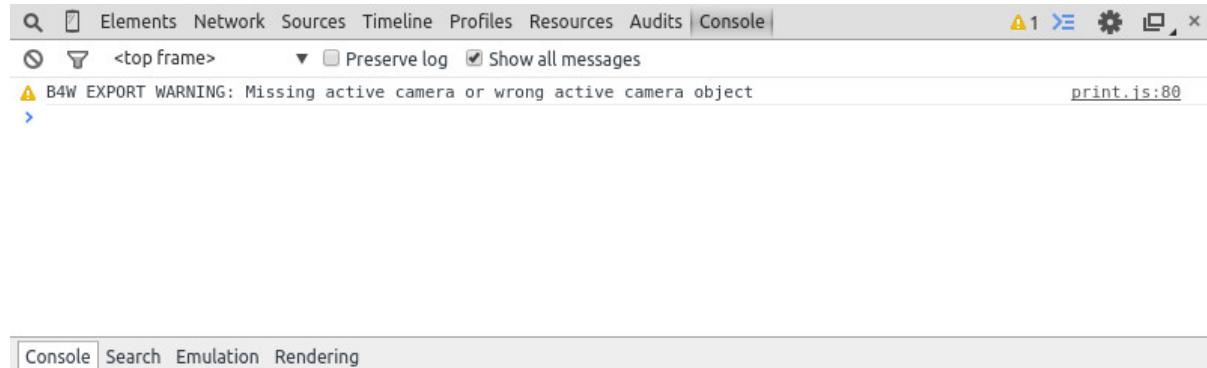
Object “NAME” has the mesh with shape keys. The property “Relative” of mesh has been enabled.	Объект с именем “NAME” имеет меш с опорными фигурами. На меше было включено свойство “Relative”.
Only 2 UV textures are allowed for a mesh; The mesh has N UVs.	Движком поддерживаются только до 2 UV текстур на каждый меш. Меш содержит UV текстуры в количестве N.
Packed media “FILE_NAME” has not been exported to “CONVERTED_FILE_PATH”	Запакованный медиа файл “FILE_NAME” не может быть конвертирован в “CONVERTED_FILE_PATH”. Требуется распаковать файл и произвести конвертацию.
Particle system error for “NAME”; The “NAME” vertex color specified in the <b>from</b> field is missing in the last of the “OBJECT_NAME” object’s vertex colors	Ошибка системы частиц. Вертексный цвет NAME указанный в поле <b>from</b> , отсутствует в эмиттере OBJECT_NAME.
Particle system error for “NAME”; The “NAME” vertex color specified in the <b>to</b> field is missing in the list of the “OBJECT_NAME” object’s vertex colors	Ошибка системы частиц. Вертексный цвет NAME указанный в поле <b>to</b> , отсутствует в объекте OBJECT_NAME, выбранном в качестве частицы.
Particle system error for “NAME”; The “NAME” vertex color specified in the “to” field is missing in the “OBJECT_NAME” object (“GROUP_NAME” dupli group)	Ошибка системы частиц. Вертексный цвет NAME указанный в поле <b>to</b> , не присутствует в объекте OBJECT_NAME группы GROUP_NAME, выбранной в качестве частицы.
Particle system error for object “NAME”. Invalid dupli object “OBJECT_NAME”.	Ошибка системы частиц для объекта NAME. Неверный дупли объект OBJECT_NAME.
Sound file is missing in the SPEAKER object “NAME”. Converted to EMPTY.	У объекта типа SPEAKER отсутствует звуковой файл. Тип объекта изменён на EMPTY.
The main scene NAME can not be rendered by another scene. Material NAME has been removed.	Основная сцена NAME не может отображаться другой сценой. Материал NAME был удален.
The main scene NAME can not be rendered by another scene. Texture NAME has been removed.	Основная сцена NAME не может отображаться другой сценой. Текстура NAME был удалена.
The NAME action has decimal frames. Converted to integer.	Анимация NAME содержит дробные значения кадров. Округлено до целых.
The NAME armature modifier has a proxy object as an armature. Modifier removed.	Модификатор арматуры имеет прокси объект в качестве арматуры.
The NAME armature modifier has no armature object or it is not exported. Modifier removed.	В модификаторе NAME типа <b>Armature</b> не указан объект, либо объект не экспортируется. Модификатор удален.

The NAME curve modifier has no curve object. Modifier removed.	В модификаторе NAME типа <b>Curve</b> не указан объект. Модификатор удален.
The NAME curve modifier has unsupported curve object. Modifier removed.	В модификаторе NAME типа <b>Curve</b> указан неподходящий объект. Модификатор удален.
The NAME object has the NAME armature modifier and a vertex animation. Modifier removed.	Объект NAME имеет вертекскую анимацию и арматурный модификатор. Модификатор удален.
The NAME LAMP node has no lamp object. Material: NAME.	В ноде NAME типа <b>LAMP</b> не указан подходящий объект.
The NAME node is not supported. The NAME material will be rendered without nodes. Material: NAME.	Нода с данным именем не поддерживается движком, поэтому нодовый материал будет отключён. Чаще всего проблемы подобного рода возникают при использовании нод <b>Cycles</b> .
The NAME object has NAME armature modifier which references the wrong group. Modifier removed.	Объект должен находиться в той же группе, что и арматура или оба объекта должны явно присутствовать на сцене.
TEXTURE_TYPE isn't supported, WORLD_NAME	TEXTURE_TYPE тип текстуры не поддерживается для мира, <b>WORLD_NAME</b>
Using B4W_REFRACTION node NODE_NAME with incorrect type of Alpha Blend. Material: NAME.	Используется нодовый материал с неправильно заданным свойством <b>Alpha Blend</b> . Допускается значение <b>Alpha sort</b> , <b>Alpha blend</b> и <b>Add</b> при использовании ноды "REFRACTION".
Wind bending: not all vertex colors exist for "NAME". Properties were set to default values.	Настройки процедурной анимации деревьев: должны существовать все указанные слои вертексных цветов.
Wind bending: vertex colors weren't properly assigned for "NAME". Properties were set to default values.	Настройки процедурной анимации деревьев; должны быть указаны названия всех слоев вертексных цветов ( <b>Main stiffness (A)</b> , <b>Leaves stiffness (R)</b> , <b>Leaves phase (G)</b> , <b>Overall stiffness (B)</b> ), либо только главного ( <b>Main stiffness (A)</b> ), либо ни одного из них.
Wrong "Height Map" input for the "NAME" B4W_PARALLAX node. Only link from the TEXTURE node with a non-empty texture is allowed.	На вход "Height Map" ноды NAME типа <b>B4W_PARALLAX</b> подаются некорректные данные. Разрешено подавать только выход из ноды типа <b>TEXTURE</b> с заполненной текстурой.
Wrong texture coordinates type in texture NAME. [Material: NAME.]	Для текстур с изображением (image) поддерживаются следующие типы координат: <b>UV</b> , <b>Normal</b> и <b>Generated</b> .

## 8.6 Прочие сообщения

Сообщения выводятся в консоли браузера (горячая клавиша F12) при загрузке сцены. Сообщение имеет вид:

B4W EXPORT WARNING: Сообщение экспорта, требующее внимания пользователя



Сообщение об ошибке	Причина
Missing active camera or wrong active camera object	На сцене отсутствует активная камера (свойство Camera на вкладке Scene).
Missing world or wrong active world object	На сцене должен быть хотя бы один мир.
NAME particle settings has the NAME texture rendering a scene. It has been replaced by the default texture.	NAME система частиц имеет NAME текстуру, которая отображает сцену. Эта текстура была заменена на стандартную.
The action NAME has no fcurves.	В анимации NAME нет ни одного канала.
The “NAME” camera has unsupported PANORAMIC type. Changed to PERSPECTIVE type.”	Панорамная камера не поддерживается. Будет использована перспективная камера.

## 8.7 Перевод аддона

Имеется возможность перевести аддон на язык, доступный в Blender'e. Для этого необходимо переименовать файл "empty.po", находящийся в директории: SDK/blender\_scripts/addons/blend4web/locales, на одно из имен, соответствующих таблице:

Имя файла	Язык
ru_RU.po	Russian
ja_JP.po	Japanese
en_US.po	English
nl_NL.po	Dutch
it_IT.po	Italian
de_DE.po	German
fi_FI.po	Finnish
sv_SE.po	Swedish
fr_FR.po	French
es.po	Spanish
ca_AD.po	Catalan
cs_CZ.po	Czech
pt_PT.po	Portuguese
zh_CN.po	Simplified Chinese
zh_TW.po	Traditional Chinese
hr_HR.po	Croatian
sr_RS.po	Serbian
uk_UA.po	Ukrainian
pl_PL.po	Polish
ro_RO.po	Romanian
ar_EG.po	Arabic
bg_BG.po	Bulgarian
el_GR.po	Greek
ko_KR.po	Korean
ne_NP.po	Nepali
fa_IR.po	Persian
id_ID.po	Indonesian
sr_RS@latin.po	Serbian Latin
ky_KG.po	Kyrgyz
tr_TR.po	Turkish
hu_HU.po	Hungarian
pt_BR.po	Brazilian Portuguese
he_IL.po	Hebrew
et_EE.po	Estonian
eo.po	Esperanto
es_ES.po	Spanish from Spain
am_ET.po	Amharic
uz_UZ.po	Uzbek
uz_UZ@cyrillic.po	Uzbek Cyrillic
hi_IN.po	Hindi
vi_VN.po	Vietnamese

Затем нужно открыть этот файл и написать перевод аддона.

Если вы пишете перевод аддона, вы можете прислать нам ваш .po файл любым доступным способом, чтобы он был включен в состав аддона.

## Объекты

---

Объекты служат целям размещения компонентов различного типа (мешей, камер, ламп и т.д.) в пространстве 3D сцены.

### 9.1 Типы

Движком поддерживаются объекты следующих типов:

- меш (mesh)
- камера (camera)
- лампа (lamp)
- пустой (empty)
- скелет (armature)
- источник звука (speaker)
- кривая (curve)
- текст (text)
- метасфера (metaball)
- поверхность (surface)

### 9.2 Статические и динамические объекты

Все объекты типа MESH можно разделить на статические и динамические.

**Статические объекты** - объекты, меши которых могут быть объединены друг с другом, если они имеют одинаковый материал.

**Динамические объекты** - объекты, меши которых не могут быть скомбинированы друг с другом.

Объединение статических объектов - т.н. батчинг (batching) - осуществляется автоматически при загрузке сцены в целях оптимизации количества вызовов отрисовки. Конвертация происходит, даже если на сцене присутствует всего один объект. Центр получившегося в результате конвертации меша находится в начале координат.

Среди объектов другого типа динамическими являются **CAMERA** и **ARMATURE**. Остальные - статические.

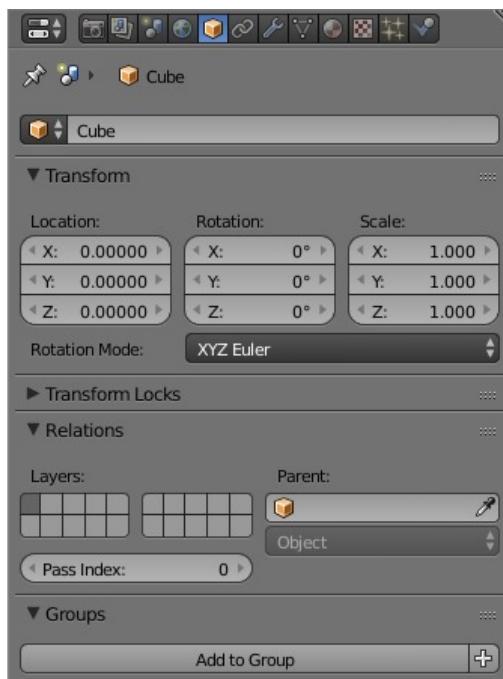
Объекты, имеющие анимацию, физику или родителя, являющегося динамическим объектом, рассматриваются как динамические.

Перемещение посредством API возможно только в случае динамических объектов. Чтобы обеспечить движение объекта, не имеющего настроек динамического объекта, в его настройках следует активировать опцию **Force Dynamic Object**.

## 9.3 Настройка

Для объектов всех типов поддерживаются расположение в пространстве, указатель на блок данных, родительский объект, принадлежность к группе и ряд свойств, специфичных для Blend4Web.

### 9.3.1 Вкладка Object



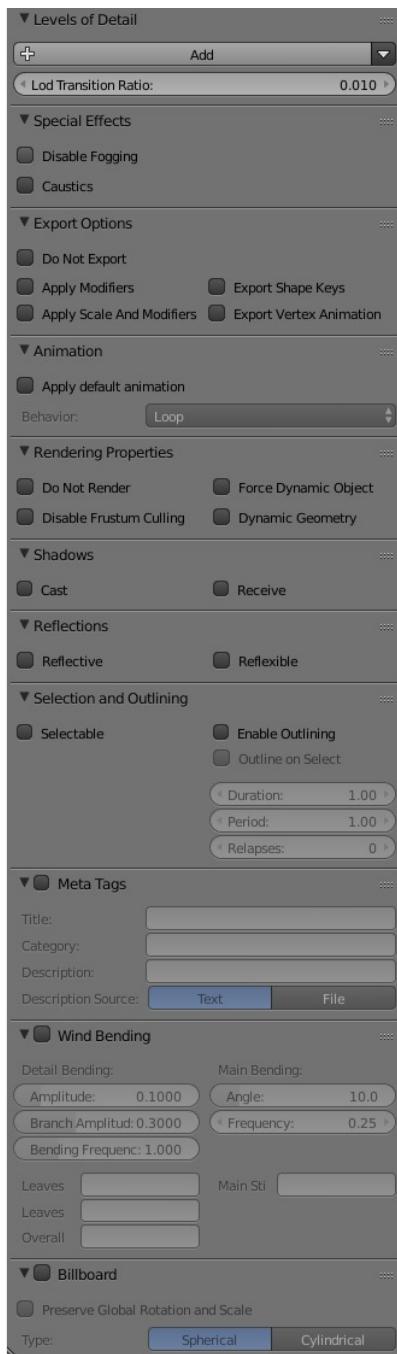
**Transform > Location** Координаты местоположения.

**Transform > Rotation** Углы вращения. При повороте объекта доступны все режимы **Rotation Mode**, но при создании *объектной анимации* поддерживаются только Quaternion (WXYZ) и XYZ Euler.

**Transform > Scale** Масштабирование. Все 3 компонента (x, y, z) должны быть одинаковы. Для физических объектов масштабирование не поддерживается.

**Relations > Parent** Указатель на родительский объект.

**Groups** Группы объектов, которым принадлежит данный объект.



**Levels of Detail > LOD Transition Ratio** Параметр для сглаживания переключения между LOD объектами. Определяет, на каком дополнительном расстоянии LOD объекты будут видны до замены последующими LOD объектами. Назначается для главного объекта. Измеряется в долях от радиуса охватывающей сферы объекта.

**Special Effects > Disable Fogging** Отключить туман для объекта.

**Special Effects > Caustics** На объекте будут видны эффекты от прилегающей к нему воды (каустика).

**Export Options > Do Not Export** Не экспортовать.

**Export Options > Apply Modifiers** Применить модификаторы объекта при экспорте. При наличии модификатора SKIN рекомендуется применить его перед экспортом, т.к. он сбрасывает слои вертексных цветов и UV-текстур, что может привести к ошибкам экспорта.

**Export Options > Apply Scale and Modifiers** Применить масштабирование и модификаторы объекта при экспорте.

**Export Options > Export Vertex Animation** Экспортовать предварительно созданную и сохраненную вертексную анимацию. Актуально только для объектов типа MESH.

**Export Options > Export Shape Keys** Экспортовать ключи деформации. Актуально только для объектов типа MESH.

---

**Примечание:** Свойства *Apply Modifiers*, *Apply Scale and Modifiers*, *Export Vertex Animation*, и *Export Shape Keys* взаимоисключают друг друга.

---

**Animation > Apply default animation** Начать воспроизведение в движке связанной с объектом анимации.

**Animation > Animation blending** Только для арматурных объектов. Разрешить плавный переход от одной скелетной анимации к другой.

**Animation > Behavior** Поведение анимации при достижении последнего кадра: Finish Stop - остановить, Finish Reset - остановить и сбросить в нулевой кадр, Loop - циклически повторять.

**Rendering Properties > Do Not Render** Отключить рендеринг объекта (например, вспомогательный объект физики).

**Rendering Properties > Disable Frustum Culling** Отключить оптимизацию отсечением по зоне видимости.

**Rendering Properties > Force Dynamic Object** Принудительно сделать объект *динамическим*.

**Rendering Properties > Dynamic Geometry** Разрешить изменение геометрии объекта через API Blend4Web.

**Shadows > Cast** Объект будет отбрасывать тени.

**Shadows > Cast Only** Объект будет отбрасывать тени, при этом сам будет оставаться невидимым. Становится доступным после включения Shadows > Cast.

**Shadows > Receive** Разрешить затенение от прилегающих объектов (получение теней) на объекте.

**Reflections > Reflexible** При включении объект будет отражаться от зеркальных поверхностей.

**Reflections > Reflexible only** При включении объект будет отражаться, при этом сам будет оставаться невидимым. Становится доступным после включения Reflections > Reflexible.

**Reflections > Reflective** При включении объект будет отражать своей поверхностью другие объекты.

**Reflections > Reflection Plane** Текстовое поле для названия пустого объекта, задающего плоскость отражения. Становится доступным после включения Reflections > Reflective.

**Selection and Outlining > Selectable** Разрешить *выделение объекта* с помощью мыши или другого устройства ввода.

**Selection and Outlining > Enable Outlining** Разрешить подсвечивание объекта (*эффект Outline*).

**Meta Tags** Интерфейс добавления мета-тегов на объект:

**Meta Tags > Title** Заголовок объекта.

**Meta Tags > Category** Категория объекта.

**Meta Tags > Description** Описание объекта. В зависимости от значения опции **Description Source** в это поле заносится либо текст описания непосредственно, либо имя файла, в котором содержится описание.

**Meta Tags > Description Source** Тип (источник) описания объекта: текст либо файл с текстом.

**Anchors > Enable Anchor** Опция доступна только для объектов типа EMPTY. Интерфейс добавления графических меток (якорей) к объектам:

**Anchors > Type** Тип якоря. Аннотация (*Annotation*) - информация из мета-тегов объектов отображается в раскрывающихся информационных окнах стандартного дизайна. Элемент (*Custom Element*) - в качестве якоря можно назначить произвольный HTML-элемент с текущей веб-страницы. Общий (*Generic*) - якорь, положение которого можно определить с помощью API модуля *anchors*.

**Anchors > Detect Visibility** Определять видимость якоря, т.е. реагировать на перекрытие объекта-якоря другими объектами. Активация опции приводит к падению производительности рендеринга и должна рассматриваться только в случае реальной необходимости.

**Anchors > Max Width** Используется для якорей-аннотаций. Ограничить всплывающее информационное окно аннотации выбранным значением (в

CSS пикселях).

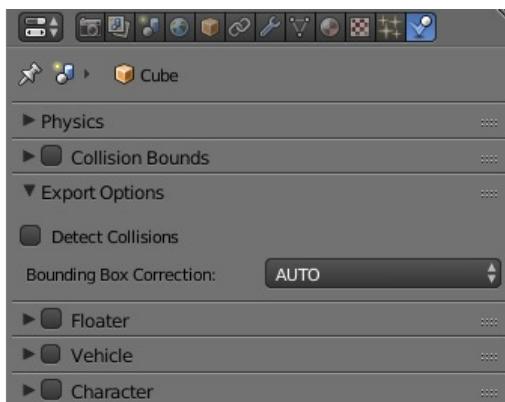
**Wind Bending** Включить процедурную анимацию под действием ветра.

**Billboard** Использовать объект в качестве билборда (т.е. автоматически ориентировать его относительно камеры).

**Billboard > Preserve Global Orientation and Scale** Билбординг объекта с учетом его ориентации и масштабирования в мировом пространстве. Объект всегда будет обращен к камере той стороной, которая видна при взгляде на него вдоль направления оси Y в системе координат Blender'a. Становится доступным после включения **Billboard**.

**Billboard > Billboard Type** Способ ориентирования билборда. **Spherical** (по умолчанию) - объект всегда ориентирован к камере одной стороной независимо от угла обзора, **Cylindrical** - аналогично **Spherical**, только вращение ограничено осью Z мирового пространства в Blender. Становится доступным после включения **Billboard**.

### 9.3.2 Вкладка Physics



**Detect Collisions** Активировать связанную с объектом физику.

**Floating** Сделать объект плавучим. Подробное описание настроек плавучих объектов можно найти в разделе *физика*.

**Vehicle** Активировать использование объекта в качестве составной части транспортного средства. Подробное описание настроек транспортных средств можно найти в разделе *физика*.

**Character** Активировать использование объекта в качестве физического каркаса игрового персонажа. Подробное описание настроек персонажа можно найти в разделе *физика*.

## 9.4 Управление перемещением объектов

**Примечание:** Необходимо обеспечить, чтобы объект, над которым выполняется преобразование, был *динамическим*.

Для управления перемещением объектов в движке предусмотрены следующие базовые функции модуля `transform`:

`get_translation()` Получить координаты центра объекта в мировой системе координат. Вариант с одним параметром возвращает новый вектор (неоптимизированный вариант), вариант с двумя требует отдельного вектора для записи результата.

`get_translation_rel()` Аналогично методу `get_translation()`, однако если у данного объекта имеется родительский объект, координаты берутся в относительной системе координат, связанной с этим родительским объектом.

`set_translation(), set_translation_v()` Переместить центр объекта в указанное место. Первая функция принимает в качестве параметров отдельные координаты, вторая - трёхмерный вектор (`Array` или `Float32Array`).

`set_translation_rel(), set_translation_rel_v()` Аналогично методам `set_translation()` и `set_translation_v()`, однако если у данного объекта имеется родительский объект, координаты задаются в относительной системе координат, связанной с этим родительским объектом.

`get_rotation()` Получить кватернион поворота объекта в мировой системе координат. По аналогии с `get_translation()` имеется два варианта вызова функции.

`get_rotation_rel()` Получить кватернион поворота объекта в системе координат, связанной с родительским объектом. По аналогии с `get_translation_rel()` имеется два варианта вызова функции.

`set_rotation(), set_rotation_v()` Установить кватернион поворота объекта в мировой системе координат. Первая функция принимает в качестве параметров отдельные координаты, вторая - четырёхмерный вектор (`Array` или `Float32Array`).

`set_rotation_rel(), set_rotation_rel_v()` Установить кватернион поворота объекта в системе координат, связанной с родительским объектом. Первая функция принимает в качестве параметров отдельные координаты, вторая - четырёхмерный вектор (`Array` или `Float32Array`).

`get_scale()` Получить значение коэффициента увеличения объекта в мировой системе координат.

`get_scale_rel()` Получить значение коэффициента увеличения объекта в системе координат, связанной с родительским объектом.

`set_scale()` Установить коэффициент увеличения объекта в мировой системе координат. Единица соответствует исходному состоянию. Значение меньше единицы - уменьшение. Значение больше единицы - увеличение. Не все объекты

могут быть увеличены. В частности, увеличение невозможно для физических объектов.

**set\_scale\_rel()** Установить коэффициент увеличения объекта в системе координат, связанной с родительским объектом.

**set\_rotation\_euler(), set\_rotation\_euler\_v()** Установить поворот объекта с помощью углов Эйлера. Используется **внутренняя** (intrinsic) система поворота YZX (то есть углы следуют в последовательности YZX, сама система отсчёта при этом вращается и занимает каждый раз новое положение).

**set\_rotation\_euler\_rel(), set\_rotation\_euler\_rel\_v()** Установить поворот объекта с помощью углов Эйлера в системе координат, связанной с родительским объектом.

**move\_local()** Переместить объект относительно своего первоначального положения (в локальной системе координат).

**rotate\_x\_local(), rotate\_y\_local(), rotate\_z\_local()** Повернуть объект относительно своего первоначального положения (в локальной системе координат).

## 9.5 Кватернионы

Кватернионы представляют собой четырёхмерные векторы, используемые для осуществления поворотов. Использование кватернионов обладает рядом преимуществ перед другими способами представления поворотов:

- Не имеет неоднозначности и зависимости от порядка применения поворотов, которые имеют место в случае использования углов Эйлера.
- Более эффективное использование памяти (от 2-х до 4-х раз меньше в зависимости от типа используемой матрицы).
- Высокая эффективность вычисления серии поворотов, чем при использовании матриц.
- Нейтрализация ошибок умножения, возникающих вследствие неточности чисел с плавающей запятой.
- Удобный метод интерполяции.

**Кватернионы имеют ряд недостатков:**

- Поворот вектора с помощью кватерниона более сложная в вычислительном плане операция чем поворот с использованием матрицы.
- Использование кватернионов для представления отличных от поворота преобразований (перспективная или ортогональная проекция) затруднено.

Для удобства работы с кватернионами в движке имеется ряд функций:

**quat.multiply** Умножение кватернионов. Умножение кватерниона А на кватернион Б слева А\*Б является поворотом на А. То есть у объекта уже имеется некий поворот Б, который мы дополняем новым поворотом на А.

*quat.setAxisAngle* Кватернион представляет собой иную форму записи поворота относительно произвольной оси (вектора) на произвольный угол. Положительное направление поворота отсчитывается против часовой стрелки, если смотреть с конца вектора. Например вызов `quat.setAxisAngle([1, 0, 0], Math.PI/3, quat)` сформирует кватернион, который может быть использован для осуществления поворота относительно оси X на 60 градусов (против часовой стрелки, если смотреть с конца оси X).

*quat.slerp* Сферическая интерполяция кватернионов. Используется для осуществления плавного разворота объектов и анимации.

*util.euler\_to\_quat, util.quat\_to\_euler*. Преобразование из углов Эйлера и наоборот.

### 9.5.1 Пример работы с кватернионами

Требуется повернуть объект на 60 градусов в горизонтальной плоскости вправо. В программе Blender имеется соответствующая модель с названием “Cessna”.



Сохраним ссылку на объект в движке в переменной `aircraft`:

```
var aircraft = m_scenes.get_object_by_name("Cessna");
```

О我们将 осуществим поворот:

- Расположение координатных осей в Blender'е и в движке различается, и при экспорте произойдет преобразование: [X Y Z] (Blender) -> [X -Z Y] (движок). Поэтому поворот нужно осуществлять относительно оси Y, а не Z.
- Повороту вправо соответствует поворот по часовой стрелке, т.е. в отрицательном направлении.
- $60$  градусов =  $\pi/3$  радиан.

В итоге получаем:

```
// compose quaternion
var quat_60_Y_neg = m_quat.setAxisAngle([0, 1, 0], -Math.PI/3, m_quat.create());

// get old rotation
var quat_old = m_transform.get_rotation(aircraft);

// left multiply: quat60_Y_neg * quat_old
var quat_new = m_quat.multiply(quat_60_Y_neg, quat_old, m_quat.create());

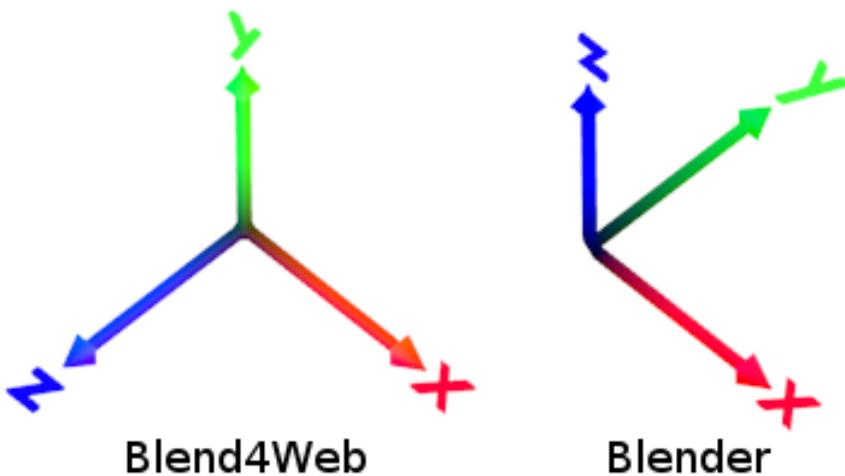
// set new rotation
m_transform.set_rotation_v(aircraft, quat_new);
```

Оптимизированный вариант, не приводящий к созданию новых объектов:

```
// cache arrays as global vars
var AXIS_Y = new Float32Array([0, 1, 0])
var quat_tmp = new Float32Array(4);
var quat_tmp2 = new Float32Array(4);
...
// rotate
m_quat.setAxisAngle(AXIS_Y, -Math.PI/3, quat_tmp);
m_transform.get_rotation(aircraft, quat_tmp2);
m_quat.multiply(quat_tmp, quat_tmp2, quat_tmp);
m_transform.set_rotation_v(aircraft, quat_tmp);
```

## 9.6 Отличия систем координат Blender'a и Blend4Web

В системе координат Blender'a вектор UP, указывающий направление вверх, сонаправлен с осью Z. В движке Blend4Web, так же как и вообще в OpenGL, это направление совпадает с направлением оси Y. Таким образом система координат движка оказывается повернутой относительно системы координат Blender'a на 90 градусов вокруг оси X.



Методы API используют именно систему координат движка, поэтому работа с API может отличаться от работы с настройками непосредственно в Blender'e.

## 9.7 Перемещение с помощью векторов TSR

Удобным способом управления перемещением объектов является использование специальных векторов, имеющих следующий формат:

$$[T_x, T_y, T_z, S, R_x, R_y, R_z, R_w]$$

Здесь  $T_x, T_y, T_z$  - компоненты вектора перемещения,  $S$  - масштаб,  $R_x, R_y, R_z, R_w$  - компоненты кватерниона поворота. Отсюда следует название вектора: TSR или TSR-8.

Для работы с данными векторами существует специальный модуль `tsr`, а также методы `set_tsr()`/`get_tsr()` модуля `transform`.

## 9.8 Копирование объектов (инстан싱)

Часто возникает необходимость копирования объектов (инстан싱) во время работы приложения.

**Существует ряд ограничений на копируемые объекты:**

- тип объекта должен быть MESH
- объект должен быть *динамическим* (включить настройку *Rendering Properties > Force Dynamic Object*)

- объект должен принадлежать активной сцене

### 9.8.1 Простое копирование

В случае простого копирования для объекта-копии будет использоваться геометрия объекта, с которого осуществлялось копирование. При изменении геометрии на исходном объекте, геометрия изменится и на объекте-копии.

### 9.8.2 Глубокое копирование

При глубоком копировании новый объект будет обладать уникальными для него свойствами, а именно, он будет иметь свою собственную геометрию. То есть при изменении геометрии на объекте-оригинале у объекта-копии геометрия изменяться не будет. Для такого типа копирования на исходном объекте нужно дополнительно выставить опцию *Rendering Properties > Force Dynamic Geometry*.

Для копирования объектов в модуле `objects.js` предусмотрена функция `copy`, в которую надо передать идентификатор копируемого объекта, уникальное имя нового объекта и булеву переменную, определяющую тип копирования: `true`- глубокое, `false` - простое. По умолчанию будет осуществлено простое копирование.

Созданный при копировании объект необходимо добавить на сцену. Для этого в модуле `scenes.js` существует функция `append_object`, куда нужно передать объект, полученный копированием.

```
// ...
var new_obj = m_objects.copy(obj, "New_name", true);
m_scenes.append_object(new_obj);
m_transform.set_translation(new_obj, 2, 0, 2);
// ...
```

### 9.8.3 Удаление объектов

Для удаления объектов, созданных копированием, в модуле `scenes.js` предусмотрена функция `remove_object`, куда надо передать объект. Возможно удаление только объектов-копий.

```
// ...
m_objects.remove_object(new_obj);
// ...
```

## 9.9 Выделение объектов

Чтобы разрешить выделение конкретного объекта, необходимо установить флаг **Selectable** в панели **Selection and Outlining**.

**Примечание:** Следует убедится в том, что в панели **Scene > Object Outlining** установлено состояние **ON** или **AUTO**.

Выделение объекта возможно программно через API, например, в модуле **scenes.js** имеется функция **pick\_object**, которая по координатам в canvas'е выделяет объект,

```
// ...
var x = event.clientX;
var y = event.clientY;

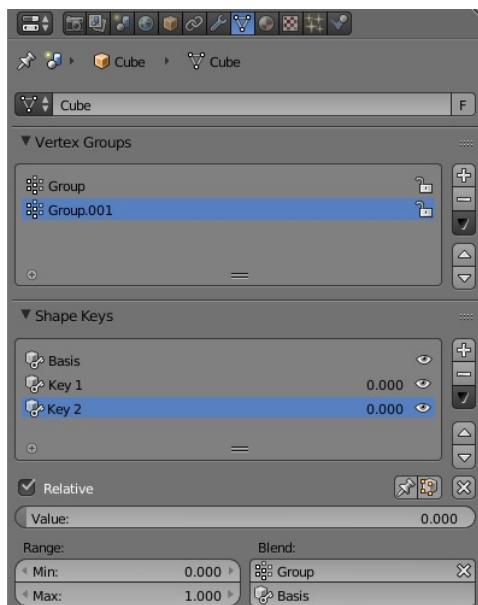
var obj = m_scenes.pick_object(x, y);
// ...
```

или с использованием *NLA-скрипта*.

Если для выделяемого объекта в панели **Object > Selection and Outlining** установлены флаги **Enable Outlining** и **Outline on Select**, то вызов функции **pick\_object** приведет к активации *Outline-анимации*.

## 9.10 Морфинг

Для создания ключей деформации (опорных фигур) перейдите во вкладку **Mesh > Shape keys**.



Движком поддерживаются все опции типа “Relative” для ключей деформации.

Для изменения значения ключа деформации, существует функция `apply_shape_key` в модуле `geometry.js`.

---

**Примечание:** На объекте должно стоять свойство *Export Shape Keys*.

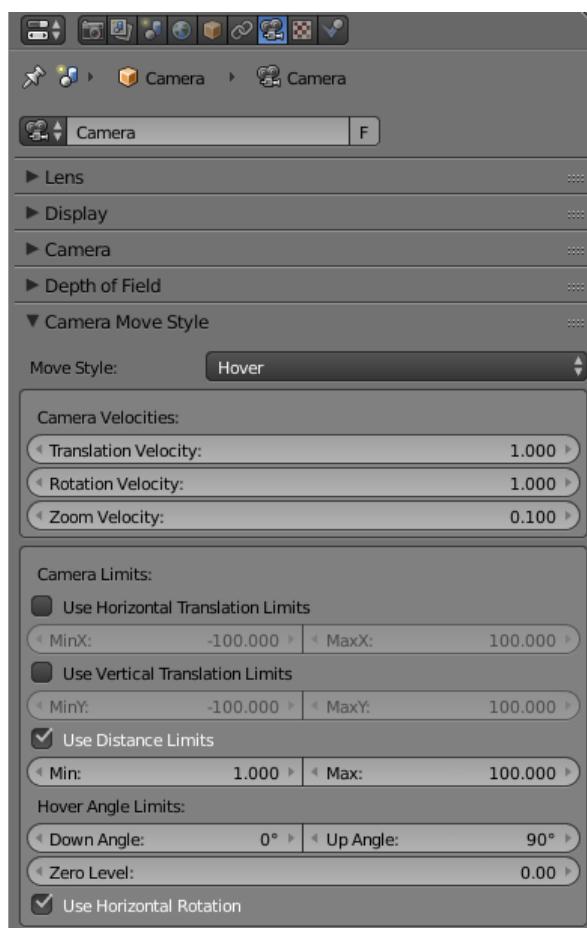
---

```
// ...
var obj = m_scenes.get_object_by_name("Object");
m_geometry.apply_shape_key(obj, "Key 1", 0.5);
// ...
```

## Камера

### 10.1 Режимы управления и общие настройки

Настройки камеры выставляются в панели Properties на вкладке Camera (Object Data).



*Camera Move Style > Move Style* Режим управления камерой:

- **Target** По умолчанию камера находится в режиме вращения вокруг некоторой точки (target). Положение точки вращения может быть изменено (см. [панорамирование](#)).
- **Eye** Режим взгляда позволяет осуществлять вращение и перемещение от первого лица.
- **Hover** В режиме нависания камера движется параллельно горизонтальной опорной плоскости.
- **Static** В статическом режиме изменение положения камеры осуществляется посредством анимации или через API.

*Camera Move Style > Target Location* Доступно в режиме Target. Позиция точки, относительно которой будет вращаться камера. Кнопка **Copy Cursor Location** позволяет скопировать текущее положение курсора.

*Depth of Field* Описано в разделе [Глубина резкости камеры](#).

## 10.2 Скорость движения камеры

Для камеры доступно несколько настроек скорости движения.

*Camera Move Style > Camera Velocities > Translation Velocity* Актуально для режимов Target, Eye, Hover. Задаёт скорость перемещения камеры.

*Camera Move Style > Camera Velocities > Rotation Velocity* Актуально для режимов Target, Eye, Hover. Задаёт скорость вращения камеры.

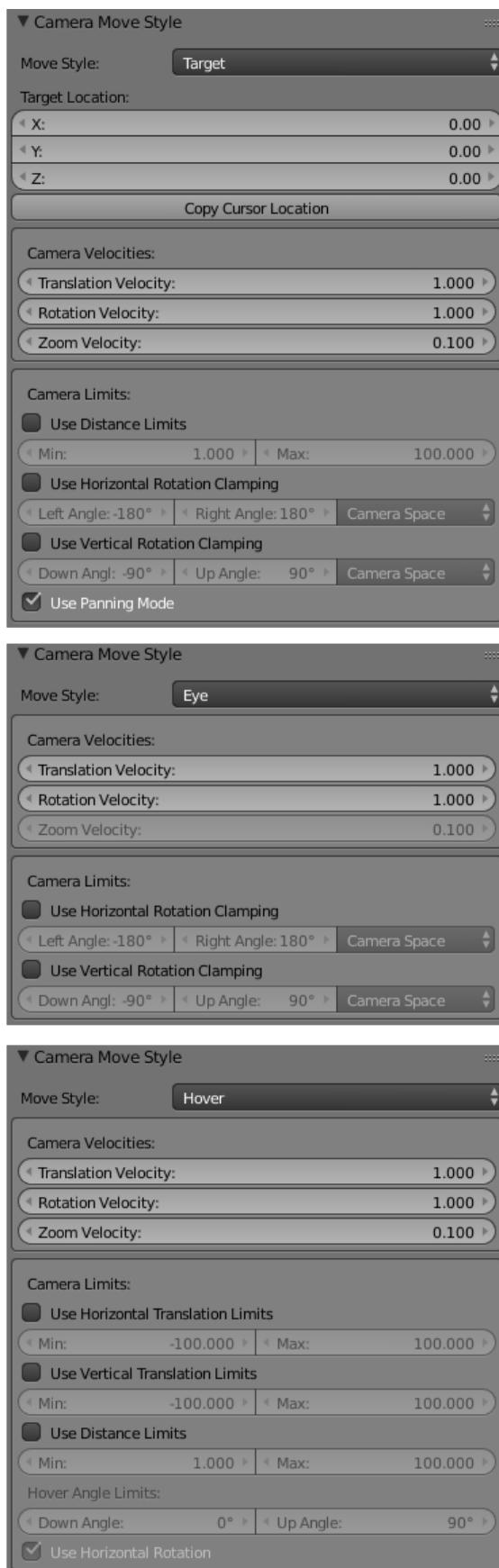
*Camera Move Style > Camera Velocities > Zoom Velocity* Актуально для режимов Target, Hover. Задаёт скорость приближения камеры к опорной точке.

Значение параметров по умолчанию: Translation Velocity, Rotation Velocity: 1; Zoom Velocity: 0.1.

Допустимые значения параметров: Translation Velocity, Rotation Velocity:  $[0, \infty)$ ; Zoom Velocity:  $[0, 1]$ .

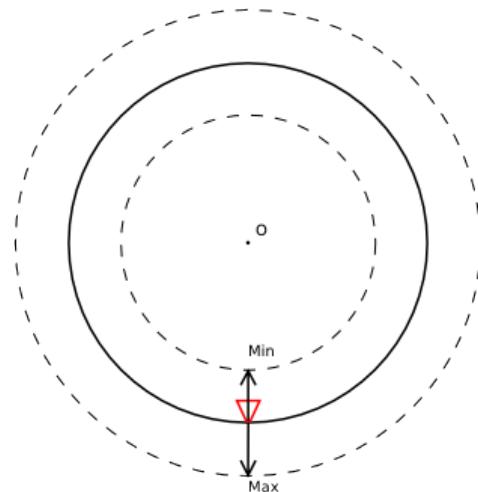
## 10.3 Ограничения движения камеры

Для камеры доступно несколько настроек, ограничивающих/изменяющих её движение тем или иным образом. Они объединены в группу Camera Limits.



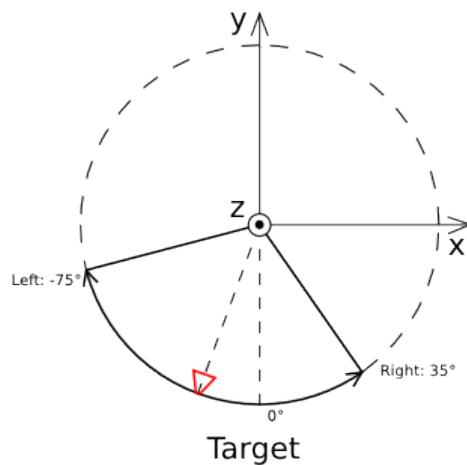
### Камера типа Target

*Camera Move Style > Camera Limits > Use Distance Limits* Задание минимального и максимального расстояний от камеры до точки вращения. Допустимые значения:  $Min \leq Max$ . По умолчанию отключено.



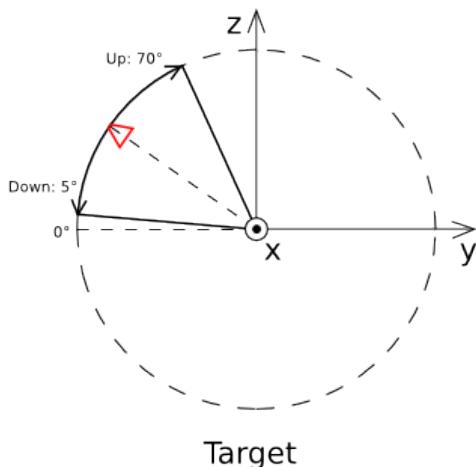
*Camera Move Style > Camera Limits > Use Horizontal Rotation Clamping*

Ограничение горизонтального (вокруг мировой оси Z в Blender'e) вращения камеры относительно соответствующей точки. Вращение происходит по дуге окружности определяемой значениями Left Angle и Right Angle. Дуга вращения соответствует движению из Left Angle в Right Angle против часовой стрелки. По умолчанию отключено.



*Camera Move Style > Camera Limits > Use Vertical Rotation Clamping*

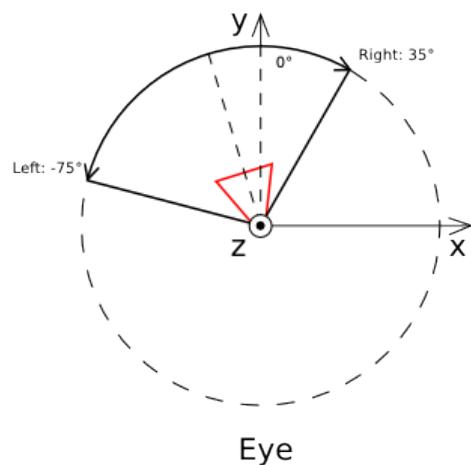
Ограничение вертикального (вокруг локальной оси X камеры в Blender'e) вращения камеры относительно соответствующей точки. Вращение происходит по дуге окружности определяемой значениями Down Angle и Up Angle. Дуга вращения соответствует движению из Down Angle в Up Angle по часовой стрелке. По умолчанию отключено.



**Camera Move Style > Camera Limits > Use panning mode** Разрешить панорамное перемещение камеры.

### Камера типа Eye

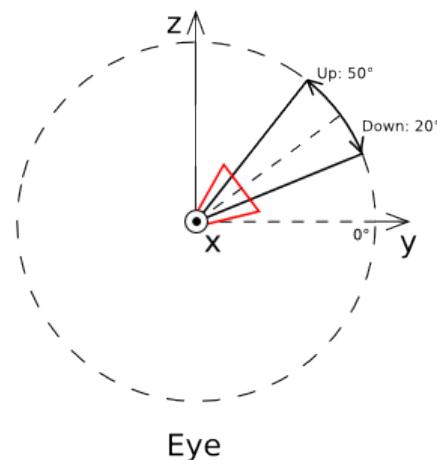
**Blend4Web > Use Horizontal Rotation Clamping** Ограничение горизонтального (вокруг мировой оси Z в Blender'е) вращения камеры относительно своего местоположения. Вращение происходит по дуге окружности определяемой значениями Left Angle и Right Angle. Дуга вращения соответствует движению из Left Angle в Right Angle по часовой стрелке. По умолчанию отключено.



**Camera Move Style > Camera Limits > Use Vertical Rotation Clamping**

Ограничение вертикального (вокруг локальной оси X камеры в Blender'е) вращения камеры относительно своего местоположения. Вращение происходит по дуге окружности определяемой значениями Down Angle и Up Angle. Дуга вращения соответствует движению из Down Angle в Up Angle против часовой

стрелки. По умолчанию отключено.



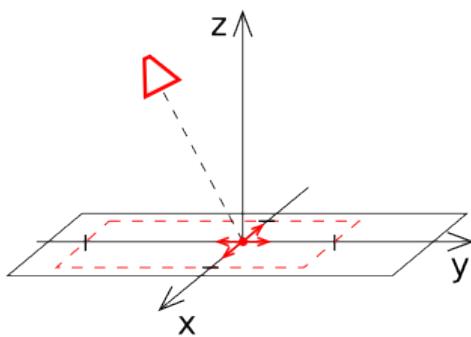
### Камера типа Hover

#### *Camera Move Style > Camera Limits > Use Horizontal Translation Limits*

Ограничение перемещения опорной точки вдоль оси X в мировых координатах в Blender'e. Допустимые значения:  $Min \leq Max$ . По умолчанию отключено.

#### *Camera Move Style > Camera Limits > Use Vertical Translation Limits*

Ограничение перемещения опорной точки вдоль оси Y в мировых координатах в Blender'e. Допустимые значения:  $Min \leq Max$ . По умолчанию отключено.



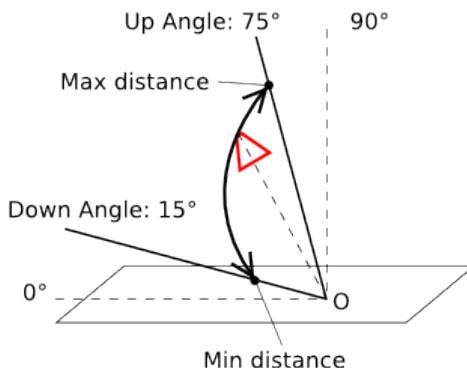
*Camera Move Style > Camera Limits > Use Distance Limits* Задание минимального и максимального расстояний от камеры до точки пересечения (опорная точка) направления взгляда камеры с горизонтальной опорной плоскостью (по умолчанию с плоскостью XOY мировых координат в Blender'e). Допустимые значения:  $Min \leq Max$ . При отключении камера будет свободно движ-

гаться вдоль оси Z мировых координат в Blender'e, опорная точка не будет задана, а настройки Camera Move Style > Camera Limits > Use Horizontal Translation Limits и Camera Move Style > Camera Limits > Use Vertical Translation Limits будут ограничивать позицию самой камеры. По умолчанию отключено.

**Camera Move Style > Camera Limits > Camera Angle Limits** Ограничения угла подъема камеры (угла между направлением взгляда камеры и горизонтальной плоскостью). Ставятся доступными при включении опции Camera Move Style > Camera Limits > Use Distance Limits. Допустимые значения:  $0 \leq \text{Down Angle} \leq \text{Up Angle} \leq 90$ . Значения по умолчанию: *Down Angle = 0, Up Angle = 90*.

**Camera Move Style > Camera Limits > Zero Level** Расстояние от опорной плоскости камеры до центра координат. Доступно при включении опции Camera Move Style > Camera Limits > Use Distance Limits.

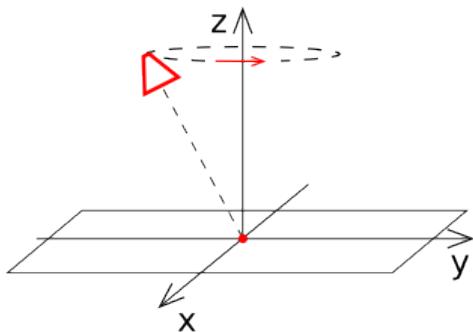
При включении настройки Blend4Web > Use Distance Limits лимиты по расстоянию и углу подъема будут действовать одновременно, задавая траекторию движения камеры в вертикальной плоскости.



Некорректное задание лимитов по расстоянию либо углу подъема приведет к отключению опции Camera Move Style > Camera Limits > Use Distance Limits.

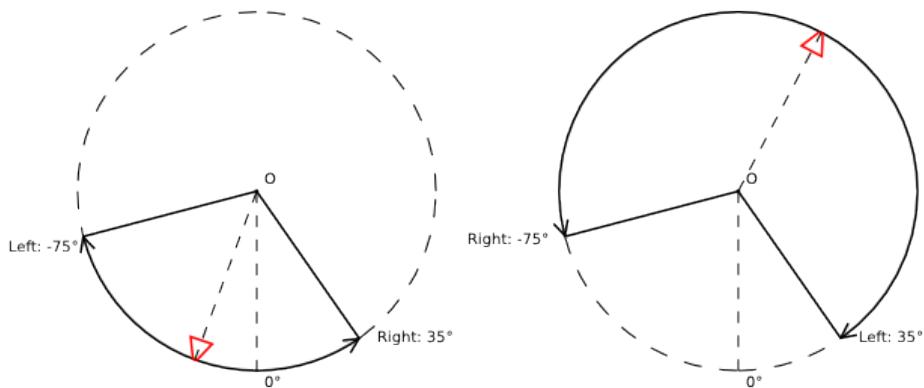
**Camera Move Style > Camera Limits > Use Horizontal Rotation**

Разрешить вращение камеры в плоскости XOY Blender'a относительно опорной точки. Ставятся доступными при включении опции Camera Move Style > Camera Limits > Use Distance Limits. По умолчанию включено.



### Особенности задания ограничений

- Для EYE/TARGET камеры перемена мест значений *Left/Right* или *Down/Up* приводит к движению по противоположной дуге окружности вращения.



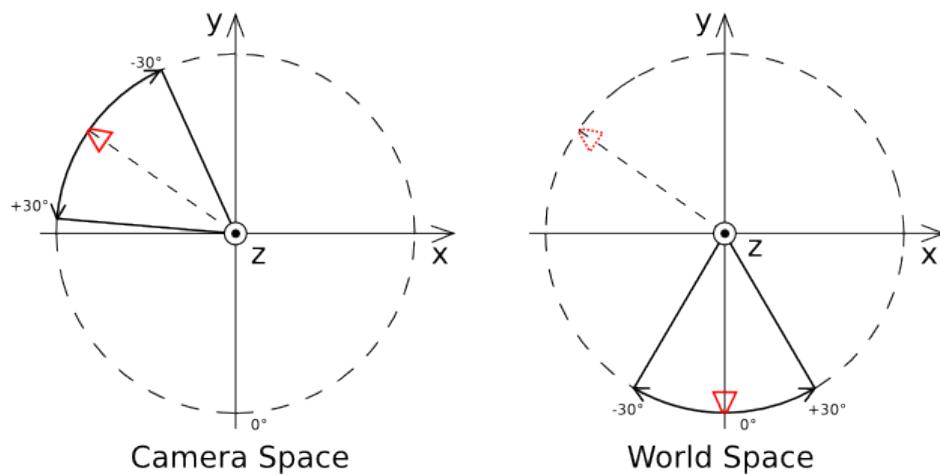
- При ограничении горизонтального и вертикального вращения камеры можно выбрать пространство координат:

**Camera Space** Все углы отсчитываются относительно начального положения и ориентации камеры.

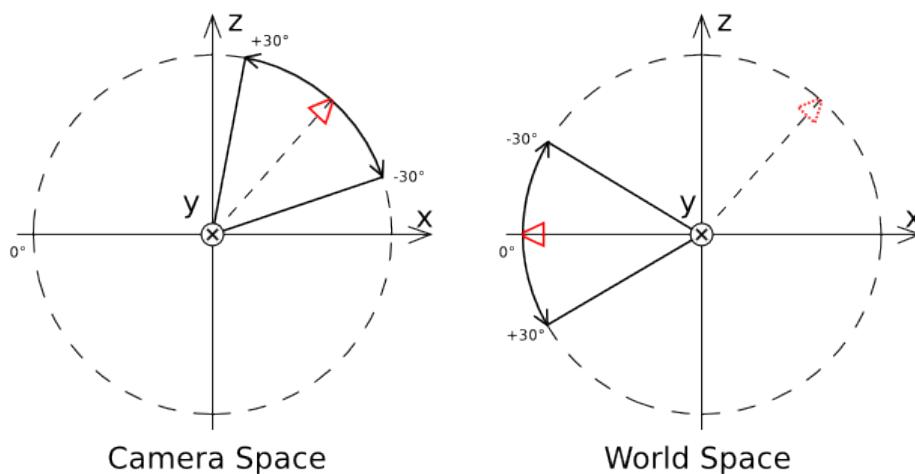
**World Space** Горизонтальные углы отсчитываются относительно направления оси Y в пространстве мировых координат, вертикальные - относительно плоскости XOY в мировом пространстве в Blender'e.

Значение по умолчанию: Camera Space.

Ограничения по горизонтали на примере TARGET камеры:



Ограничения по вертикали на примере TARGET камеры:

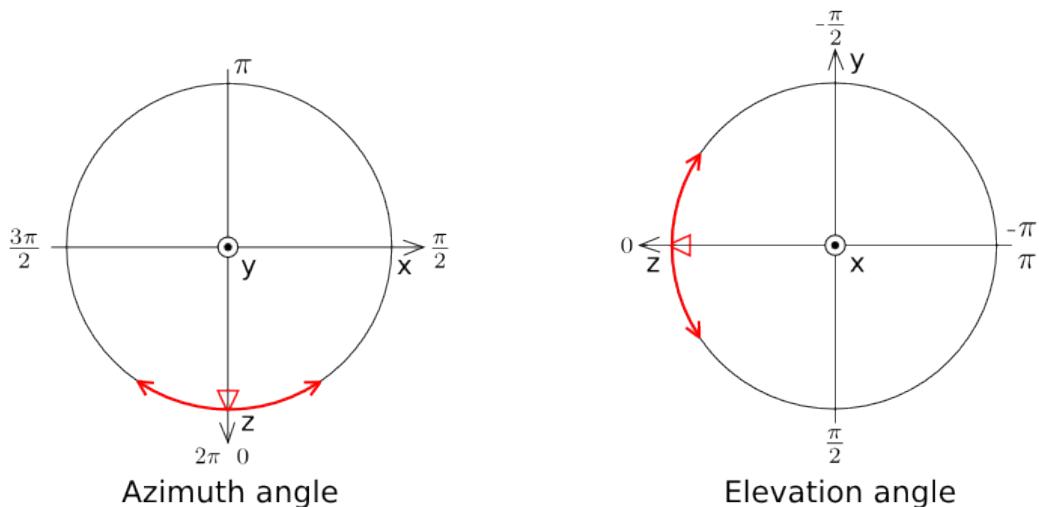


## 10.4 Управление камерой через API

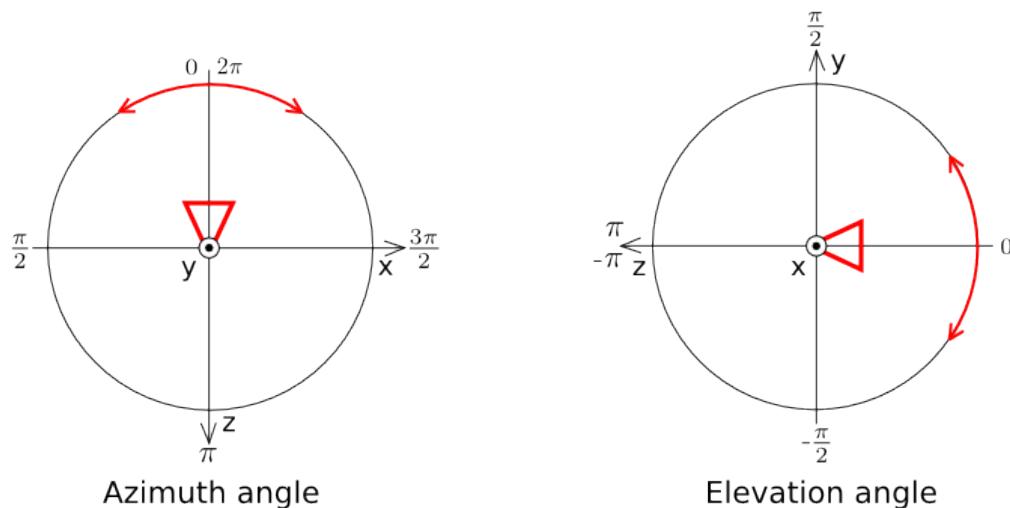
Основные функции управления камерой находятся в модуле `camera.js`.

При работе с камерой через API (вращение, выставление ограничений) отсчет угловых координат ведется в *системе координат движка* следующим образом:

Для типов TARGET/HOVER:

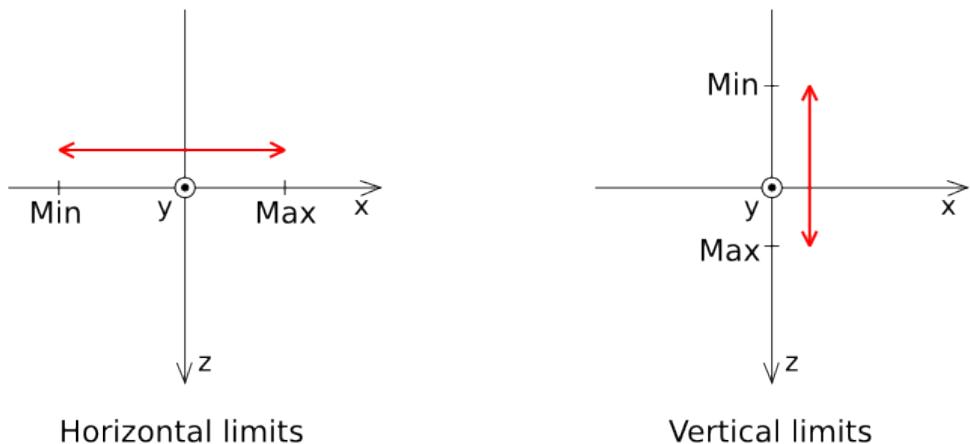


Для типа EYE:



При выставлении лимитов перемещения для камеры типа `HOVER` значения лимитов будут соответствовать осям системы координат движка:

- Horizontal Translation Limits - ось X
- Vertical Translation Limits - ось Z



### Изменение режима управления

Для изменения режима управления камерой (опция в Blender - *Blend4Web > Move Style*) достаточно использовать метод `set_camera_move_style()` модуля `app.js`:

```
// ...
var m_app = require("app");
var m_cam = require("camera");
// ...
m_app.set_camera_move_style(m_cam.MS_TARGET_CONTROLS);
// ...
```

Если приложение не использует функционал модуля `app.js`, для этих же целей можно использовать метод `set_move_style()` модуля `camera.js`:

```
// ...
var m_cam      = require("camera");
var m_scenes  = require("scenes");
// ...
var camera = m_scenes.get_active_camera();
m_cam.set_move_style(camera, m_cam.MS_TARGET_CONTROLS);
// ...
```

При изменении режима происходит сброс лимитов перемещения и позиции опорной точки (для камер типа `TARGET` и `HOVER`), поэтому их необходимо установить заново с помощью соответствующих методов модуля `camera.js`.

### Материалы

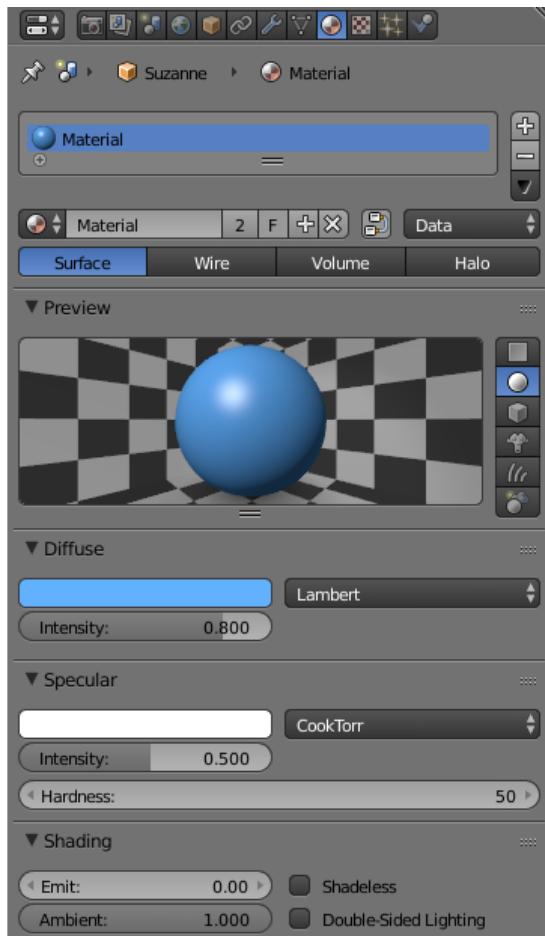
---

Материалы описывают реакцию поверхности объекта на освещение, а также содержат информацию о ее прозрачности, отражающей способности, физических и других параметрах.

Меши могут использовать один или несколько материалов. В случае использования нескольких материалов назначение их различным полигонам происходит в режиме редактирования **Edit Mode**. Для этого нужно выделить нужные полигоны, выбрать материал из списка и нажать кнопку **Assign**.

Поддерживаются следующие типы материалов: **Surface** (поверхность), **Halo** (гало).

## 11.1 Параметры освещения



**Diffuse > Color** Цвет диффузного (рассеянного) света. Значение по умолчанию (0.8, 0.8, 0.8). Может взаимодействовать с цветом диффузной текстуры.

**Diffuse > Intensity** Интенсивность диффузного (рассеянного) света. Значение по умолчанию 0.8.

**Diffuse > Shader** Алгоритм расчета диффузного (рассеянного) освещения. Значение по умолчанию *Lambert*.

**Specular > Color** Цвет отраженного света. Значение по умолчанию (1.0, 1.0, 1.0). Может взаимодействовать с цветом карты бликов.

**Specular > Intensity** Интенсивность отраженного света. Значение по умолчанию 0.5.

**Specular > Hardness** Степенный показатель в формуле расчета отраженного света (“жесткость” блика). Значение по умолчанию 50. Алгоритм применения в движке отличается от алгоритма применения в Blender’е.

**Specular > Shader** Алгоритм расчета отраженного освещения. Значение по умолчанию *CookTorr*.

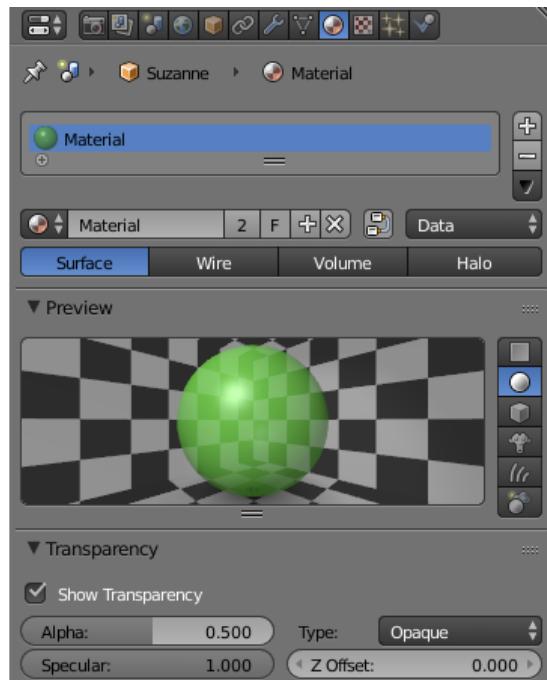
**Shading > Emit** Интенсивность эмиссии (излучения). Значение по умолчанию 0.0.

**Shading > Ambient** Фактор влияния освещения от окружающей среды на материал. Значение по умолчанию 1.0.

**Shading > Shadeless** При включении материал не реагирует на освещение. По умолчанию выключено.

**Shading > Double-sided Lighting** Включить двухстороннее освещение. Опция полезна для однослойных непросвечивающих объектов.

## 11.2 Прозрачность



### 11.2.1 Типы

Тип реализации прозрачности выбирается в меню Transparency на панели Properties > Material.

Движком поддерживаются следующие типы реализации прозрачности, перечисленные в порядке увеличения производительности:

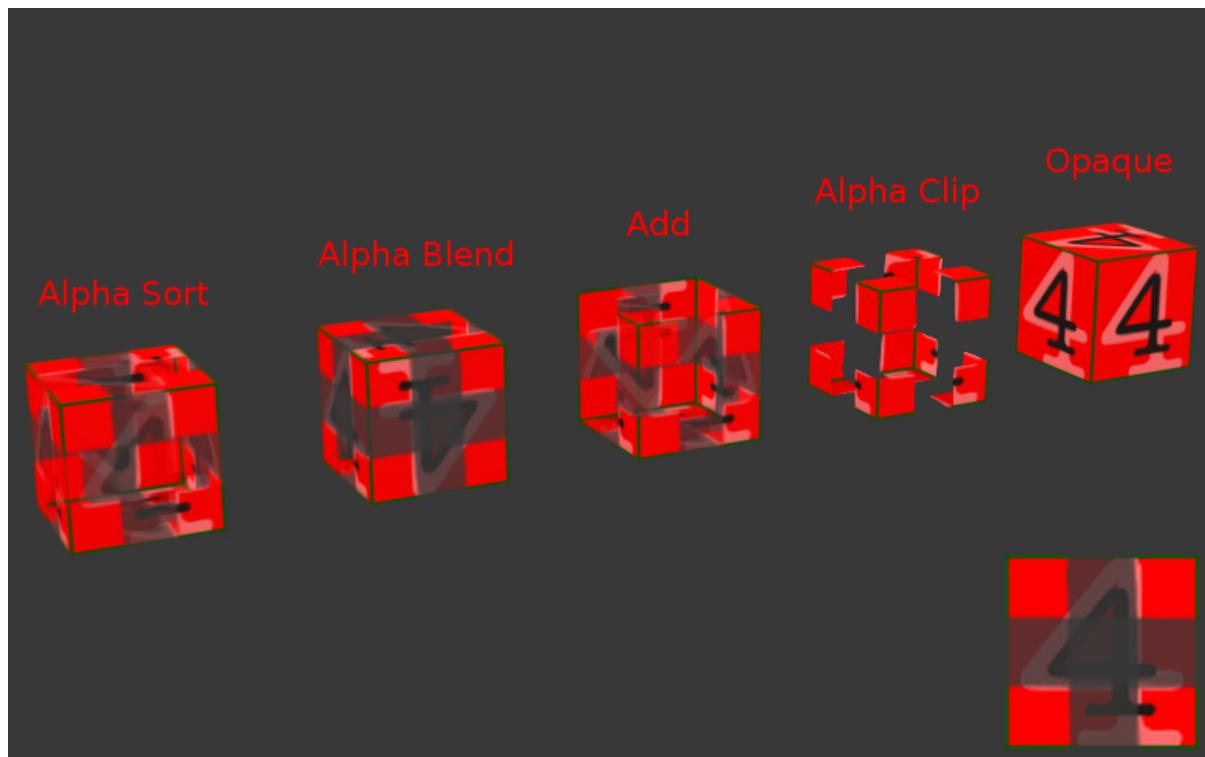
**Alpha Sort** Прозрачный с градиентом. Для корректного отображения перекрывания одних прозрачных поверхностей другими движком производится сортировка треугольников по дальности от камеры. Операция требует дополнительных затрат вычислительных ресурсов. Рекомендуется применять для замкнутых прозрачных объектов (бутилка, стекла автомобиля и т.д.).

**Alpha Blend** Прозрачный с градиентом. Сортировка треугольников не производится. Рекомендуется применять для незамкнутых прозрачных объектов (поверхность воды, декали).

**Add** Прозрачный с градиентом. Сортировка треугольников не производится. Движок отключает запись в буфер глубины, что приводит к произвольному порядку отображения прозрачных поверхностей. Рекомендуется применять для создания эффектов (системы частиц, светящиеся лучи).

**Alpha Clip** Прозрачный без градиента. Движок отбрасывает пиксели (discard) с прозрачностью менее 0.5. Сортировка треугольников не производится. Рекомендуется применять с текстурой в качестве маски для визуализации множества мелких деталей (листва деревьев, травы).

**Opaque** Непрозрачный. Альфа-канал игнорируется. Значение по умолчанию.



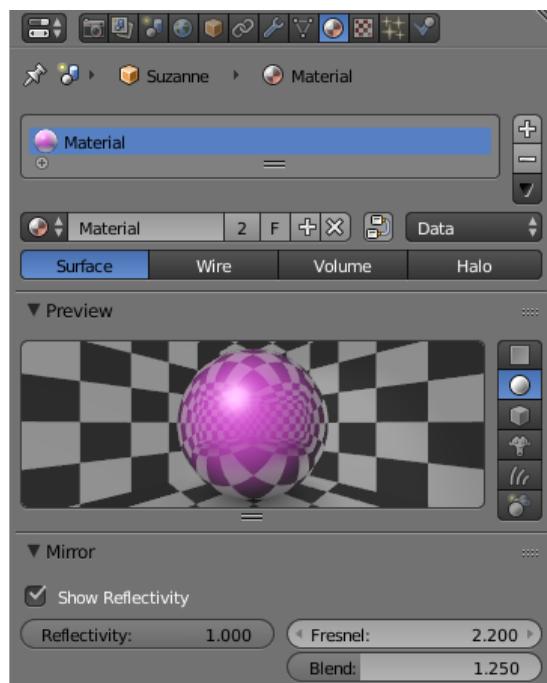
## 11.2.2 Дополнительные настройки

**Transparency > Show Transparency** Опция включения прозрачности требуется для отображения прозрачных объектов в Blender'e. Движок игнорирует эту опцию, используя вместо нее Alpha Blend.

**Transparency > Alpha** Уровень прозрачности материала. При наличии диффузной текстуры движок (в отличие от Blender'a) игнорирует этот параметр, используя вместо него значение прозрачности текстуры.

**Transparency > Z Offset, смещение по глубине** Используется для явного указания расположения прозрачных объектов с **разными** материалами относительно друг друга с целью сортировки по глубине. Может принимать отрицательные и положительные значения. Для корректного отображения дальние объекты должны иметь меньшее значение параметра, чем ближние. Значение по умолчанию 0.0.

## 11.3 Зеркальное отражение



### 11.3.1 Статическое отражение

Поверхность отражает одно и то же изображение вне зависимости от изменения окружающей среды. Для активации достаточно использовать *карту зеркального отражения*.

**См. также:**

*Эффект Френеля для отражения*

### 11.3.2 Динамическое отражение

Поверхность отражает текущее расположение определенных объектов. Поддерживаются отражения от плоскости и сферические отражения.

#### Активация

1. Проверить опцию **Reflections** на панели **Render > Reflections and Refractions**.
2. Для отражающих объектов на панели **Object > Reflections** выставить опцию **Reflective**.
  - Для плоских отражений выставить значение **Object > Reflections > Type** в **Plane**. Добавить пустой объект для задания плоскости отражения

Add > Empty > Single Arrow. Переименовать для удобства и указать его имя в поле Reflection plane отражающего объекта.

- Для кубических отражений выставить значение Object > Reflections > Type в Cube.
3. Для нужных материалов *отражающих* объектов выставить значение отражающей способности Material > Mirror > Reflectivity.
    - Mirror > Reflectivity > Show Reflectivity требуется для отображения отражений на объектах в Blender'e. Движок игнорирует эту опцию.
  4. Для *отражаемых* объектов на панели Object > Reflections выставить опцию Reflexible.

---

**Примечание:** Рекомендуется также включить использование освещения от окружающей среды World > Environment Lighting.

---

## Ограничения

В отраженном изображении игнорируется карта нормалей, тени.

**См.также:**

[Эффект Френеля для отражения](#)

### 11.3.3 Эффект Френеля для отражения

Эффект Френеля проявляется в зависимости интенсивностей проходящего и отраженного света от угла падения. Если угол падения близок к нулю (т.е. свет падает почти перпендикулярно поверхности), доля проходящего света велика, а отраженного мала. И наоборот, если угол падения близок к 90 градусам (т.е. свет падает почти параллельно поверхности), отражается почти весь свет.

Движок использует приближенную формулу Шлика:

$$R = R_0 + (1 - R_0)(1 - \cos \theta)^N, \text{ где}$$

R - коэффициент отражения,

R<sub>0</sub> - коэффициент отражения в случае обзора под прямым углом к поверхности (т.е. при  $\theta = 0$ ),

$\theta$  - угол падения (равный углу отражения, под которым свет попадает в камеру), рассчитывается движком в реальном времени,

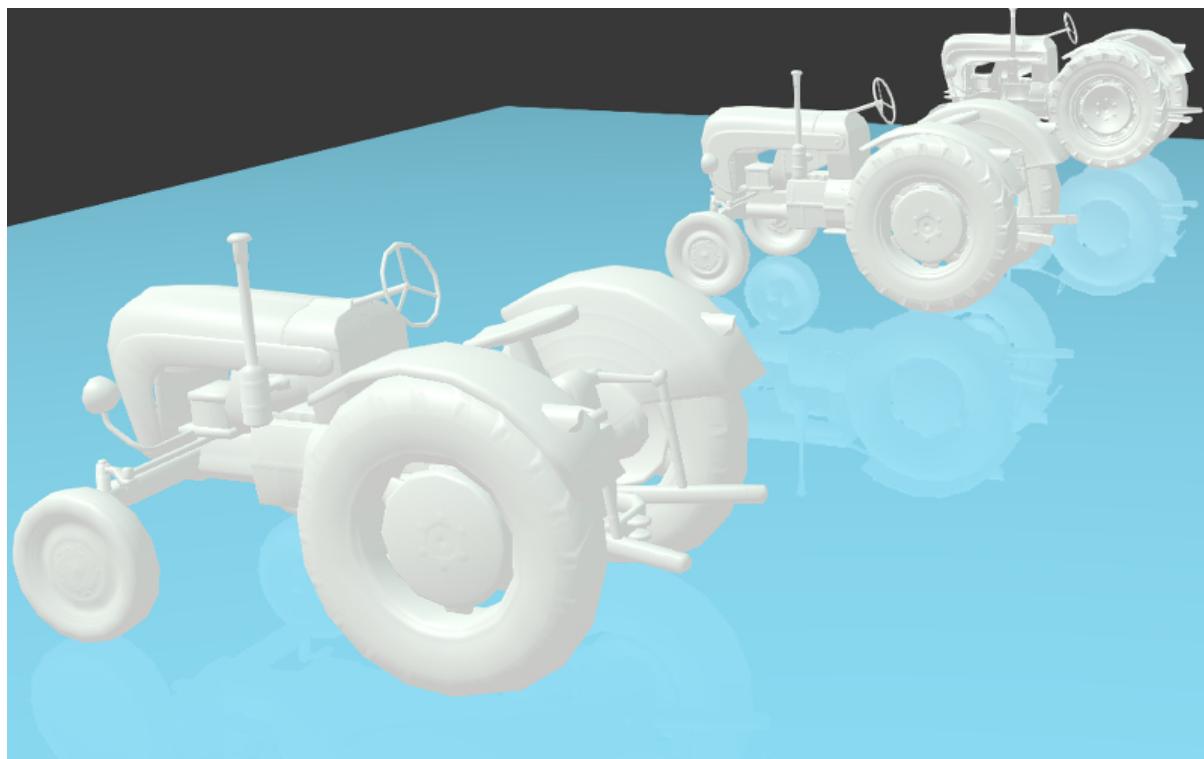
N - показатель степени.

## Настройка

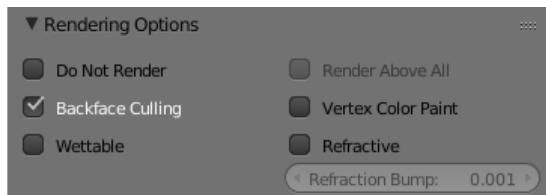
Эффект Френеля применяется как для статического, так и для динамического отражения.

**Material > Mirror > Fresnel** Степень Френеля для отражения. Показатель степени N в формуле Шлика. В пакете Blender ограничен значениями от 0 до 5. Если этот параметр равен нулю, эффект Френеля не проявляется, происходит *полное отражение* на всех углах. Если этот параметр больше нуля, при взоре поверхности под углами, близкими к прямому (почти перпендикулярно поверхности), материал становится менее отражающим. Чем больше этот параметр, тем больше отклонение угла от прямого, для которого наблюдается такой эффект.

**Material > Mirror > Blend** Фактор Френеля для отражения. Приводится к  $R_0$  в формуле Шлика:  $R_0 = 1 - \text{Blend} / 5$ . В пакете Blender ограничен значениями от 0 до 5. Этот параметр показывает интенсивность проявления эффекта Френеля: чем больше фактор Blend, тем сильнее влияние эффекта Френеля. Если он равен нулю, эффект Френеля не проявляется.



## 11.4 Опции рендеринга



**Material > Rendering Options > Do not Render** Отключить рендеринг объекта.

**Material > Rendering Options > Backface Culling** При включении обратная сторона полигона не отображается движком. По умолчанию включено.

**Material > Rendering Options > Wettable** Для материала активируется эффект смачивания водой.

См.также:

*Вода*

**Material > Rendering Options > Render above all** Материал отрисовывается поверх всех объектов на сцене. Требует тип прозрачности с градиентом (Add, Alpha Blend или Alpha Sort)

**Material > Rendering Options > Vertex Color Paint** Опция включает использование вертекского цвета меша вместо диффузного цвета.

**Material > Rendering Options > Refractive** Включить рефракцию. Коэффициент возмущения задаётся опцией Refraction Bump на панели Refraction Settings. Значение по умолчанию 0.001.

---

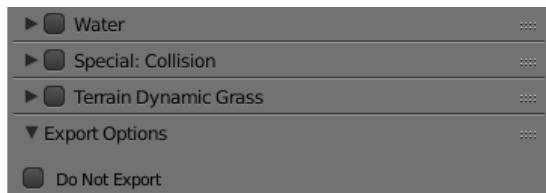
**Примечание:** Для отображения эффекта необходимо выбрать опцию ON или AUTO на панели Render > Reflections and Refractions > Refractions. Объект должен быть с типом прозрачности Alpha Blend.

---

См.также:

*Прозрачность*

## 11.5 Специальные параметры движка



**Material > Water** Специальный материал для *рендеринга воды*.

**Material > Terrain dynamic grass** Материал используется для *рендеринга травяного покрова*.

**Material > Collision** Специальный материал для физического объекта.

См.также:

*Физика*

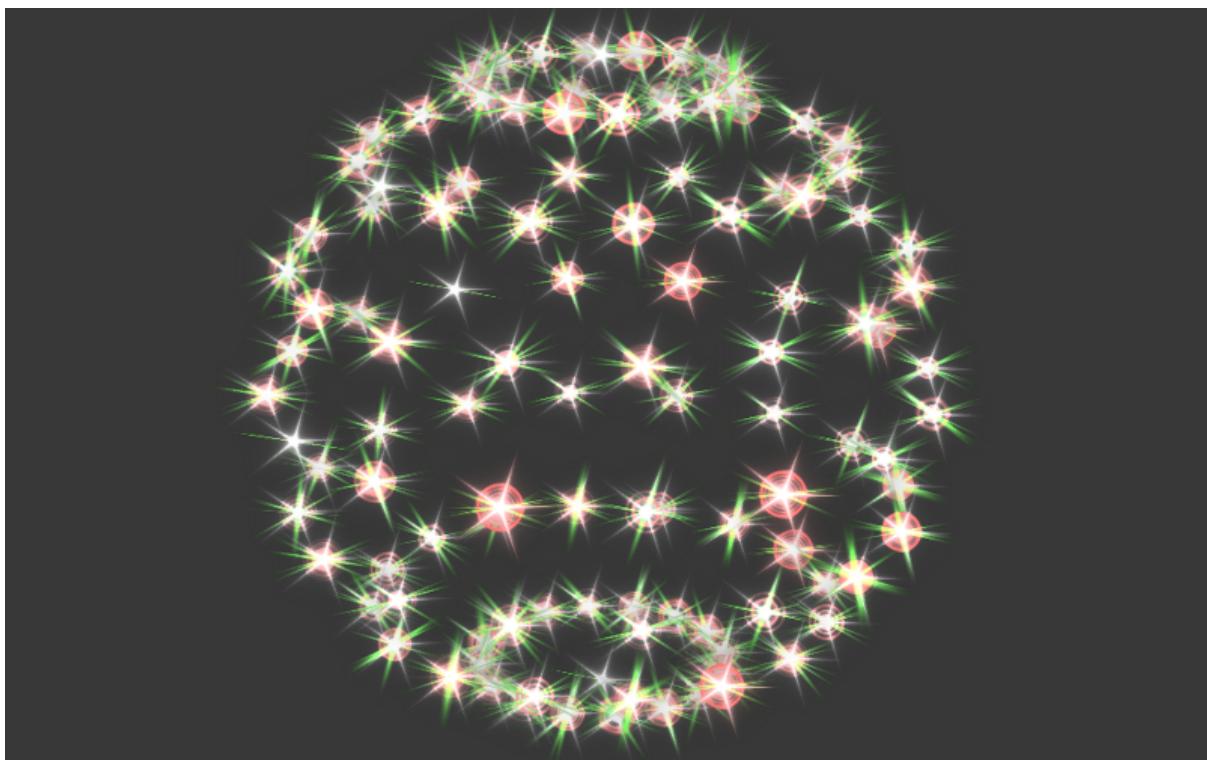
**Material > Export Options > Do Not Export** Не экспортовать.

## 11.6 Материалы гало (Halo)

Используются в системах частиц и в статических мешах. Ниже рассматривается использование гало на статических мешах.

### 11.6.1 Активация

Выставить тип **Halo** во вкладке **Materials**. Рекомендуется также выставить тип прозрачности с градиентом (**Add**, **Alpha Blend** или **Alpha Sort**).



### 11.6.2 Дополнительные настройки

**Halo > Alpha** Параметр прозрачности материала. Значение по умолчанию 1.0 (непрозрачный).

**Halo > Color** Цвет материала. Значение по умолчанию (0.8, 0.8, 0.8) (почти белый).

**Halo > Size** Размер частиц. Значение по умолчанию 0.5.

**Halo > Hardness** Показатель степени при расчете градиента. Влияет на видимый размер частиц. Значение по умолчанию 50.

**Halo > Rings** Использовать кольца. Настраивается относительное количество и цвет.

**Halo > Lines** Использовать линии. Настраивается относительное количество и цвет.

**Halo > Star Tips** Использовать звезды. Настраивается количество концов.

**Halo > Special: Stars** Включает режим рендеринга звездного неба, при этом меш неподвижен относительно камеры. Для лампы типа Sun необходимо также выставить опцию Lamp > Dynamic Intensity. Приложения должны установить ночное время суток, используя API.

**Halo > Blending Height** Диапазон высот, на котором происходит затухание яркости звезд.

**Halo > Minimum Height** Минимальная высота в локальном пространстве объекта, на которой видны звезды.

## Текстуры

---

Текстуры - подготовленные вручную или процедурно сгенерированные изображения, накладываемые на поверхность моделей с целью их детализации. Для соотнесения точек 3D поверхности и пикселей плоского изображения, как правило, используются текстурные развертки. По этой причине текстуры иногда называют картами.

Текстуры обычно помещаются в текстурные слоты *материалов*, также возможно их использование для параметризации *систем частиц* и визуализации *небесного свода*.

### 12.1 Типы текстур

Опция выбора типа текстуры Туре расположена во вкладке **Textures**. Дважды поддерживаются текстуры следующих типов:

#### 1. Image or Movie, изображение или фильм

- *диффузная (diffuse map)*
- *карта бликов (specular map)*, может также содержаться в альфа-канале диффузной текстуры
- *карта нормалей (normal map)*
- *карта высот (height map)*, может содержаться только в альфа-канале карты нормалей, используется для *реализации рельефной поверхности (parallax mapping)*
- *карта смещивания (stencil map)*
- *видеотекстура*

#### 2. Environment Map, карта окружения

- *карта зеркального отражения (mirror map)*
- *текстура неба (skydome)*
- используется для реализации одной из методик *освещения от окружающей среды*

### 3. None, пустая

- применена на кубе в стартовой сцене Blender'a. Также используется для *рендеринга сцены в текстуру*, а также для *рендеринга canvas-текстуры*.

### 4. Blend, градиент

- используется в *системах частиц*

## 12.2 Общие настройки

**Размер** Размер растром для текстур-изображений (длина и ширина изображения в пикселях) должен быть числом  $2^N$ , т.е. 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096 пикселов. Использование текстур других размеров (т.н. NPOT) поддерживается, но не рекомендуется. Для корректной работы компрессии текстур размер должен составлять не менее 4 пикселов. Как правило, используются изображения квадратной формы (например, 512 x 512 px), однако могут использоваться и прямоугольные (например, 4 x 128 px). Использование изображений размером более 2048 пикселов не рекомендуется.

**Image Mapping > Extension** Режим интерпретации текстурных координат (в WebGL - Wrap Mode). Доступен для текстур типа **Image or Movie**. В случае значения **Repeat** движок устанавливает для текстуры режим **REPEAT**. При этом целочисленная часть текстурных координат игнорируется, используется только дробная часть. Во всех остальных случаях (например, **Extend**) движок устанавливает **CLAMP\_TO\_EDGE**. При этом происходит ограничение текстурных координат отрезком [0, 1]. Значение по умолчанию **Repeat**.

**Mapping > Coordinates** Тип текстурных координат. Поддерживаются **UV** (использовать развертку), **Normal** (использовать направление на камеру, только для диффузных текстур, применяется для создания материалов в стиле **material capture, matcap**) и **Generated** (исходные недеформированные координаты объекта). Значение по умолчанию **Generated**.

**Mapping > Size** Масштабирование развертки по соответствующим осям. Значения по умолчанию 1.0.

**Export Options > Do Not Export** Не экспортовать текстуру.

**Export Options > Disable Compression** Отключить использование текстурной компрессии для данной текстуры. Применяется в случае, когда компрессия приводит к ухудшению качества изображения. В частности, рекомендуется отключать для текстур-масок, использующихся для смешивания различных характеристик материалов.

**Export Options > Shore Distance Map** Используется в *рендеринге наружных сцен*.

**Export Options > Anisotropic Filtering** Фактор анизотропной фильтрации для индивидуальной текстуры. Имеет приоритет перед аналогичной настройкой

для сцены. Значение по умолчанию **DEFAULT** (т.е. использовать настройки сцены).

**Water Foam** Текстура пены. Используется материалом для рендеринга воды.

---

**Примечание:** Для карт нормалей текстурная компрессия не применяется.

---

## 12.3 Диффузная текстура (diffuse map)

Диффузная текстура применяется для указания распределения цвета рассеянного света (модель Ламберта).

### 12.3.1 Активация

Выставить опцию **Diffuse > Color** на панели **Textures > Influence**.

### 12.3.2 Дополнительные настройки

**Influence > Diffuse > Color** Степень влияния текстуры на диффузный цвет.  
Значение по умолчанию 1.0.

**Influence > Blend** Тип взаимодействия с цветом материала (**Material > Diffuse > Color**), или с вертексным цветом, если включена опция **Vertex Color Paint**.  
Поддерживаются **Mix** (смешивается с цветом), **Multiply** (умножается на цвет).  
Значение по умолчанию **Mix**.

## 12.4 Карта бликов (specular map)

Карта бликов применяется для указания распределения цвета отраженного света (модель Фонга).

### 12.4.1 Активация

Выставить опцию **Specular > Color** на панели **Textures > Influence**.

### 12.4.2 Дополнительные настройки

**Influence > Specular > Color** Степень влияния текстуры на цвет отраженного света. Значение по умолчанию 1.0.

**Influence > Blend** Тип взаимодействия с цветом отраженного света материала (**Material > Specular > Color**). Поддерживается только **Mix** (смешивается с цветом). Значение по умолчанию **Mix**.

Карта бликов может быть упакована в альфа-канал диффузной текстуры в целях оптимизации. В этом случае для текстуры необходимо одновременно выставить опции **Diffuse > Color** и **Specular > Color**. Цветовой диапазон ограничен оттенками серого цвета.

## 12.5 Карта нормалей (normal map)

Карта нормалей применяется для указания распределения нормалей (перпендикуляров) к поверхности с целью увеличения уровня детализации ее рельефа. Информация о нормалях должна храниться в текстурном пространстве координат. Карты нормалей в объектном пространстве не поддерживаются.

### 12.5.1 Активация

Выставить опцию **Geometry > Normal** на панели **Textures > Influence**.

### 12.5.2 Дополнительные настройки

**Influence > Geometry > Normal** Степень участия карты в расчетах нормалей. Значение по умолчанию 1.0.

## 12.6 Карта высот (height map). Parallax mapping

Карта высот содержит информацию о распределении относительных высот рельефа. Более высокий уровень поверхности обозначается более светлым цветом. Карта высот в сочетании с картой нормалей требуются в качестве входящих данных для реализации рельефной поверхности (parallax mapping). Карта высот должна содержаться в альфа-канале карты нормалей.

### 12.6.1 Активация

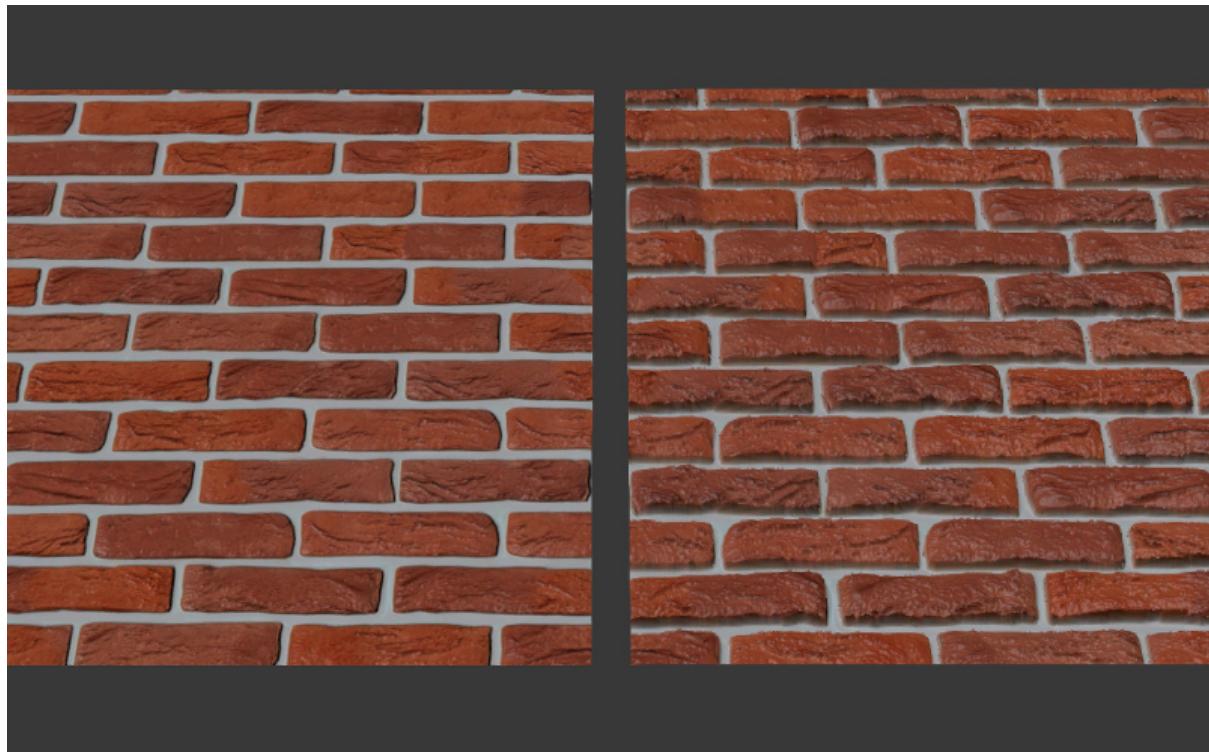
Для карты нормалей дополнительно к опции **Geometry > Normal** активировать панель **Parallax**.

### 12.6.2 Дополнительные настройки

**Parallax > Parallax Scale** Фактор влияния эффекта рельефной поверхности. Значение по умолчанию 0.03.

**Parallax > Parallax Steps** Количество итераций в расчетах рельефной поверхности. Большее значение приводит к лучшему качеству и к большим затратам вычислительных ресурсов. Значение по умолчанию 10.

**Parallax > Parallax LOD distance** Расстояние на котором виден эффект параллакса.



## 12.7 Карта смещивания (stencil map)

Специальная текстура (цветная или оттенков серого), содержащая информацию о распределении других текстур по поверхности.

### 12.7.1 Активация

1. В случае нодовых материалов карта смещивания должна использоваться соответствующим образом в нодовой структуре.
2. В случае обычных материалов карта смещивания должна располагаться в текстурном слоте между двумя смешиаемыми диффузными текстурами. Для текстуры смещивания необходимо одновременно выставить опции **RGB to Intensity** и **Stencil** на панели **Textures > Influence**.

## 12.7.2 Дополнительные настройки

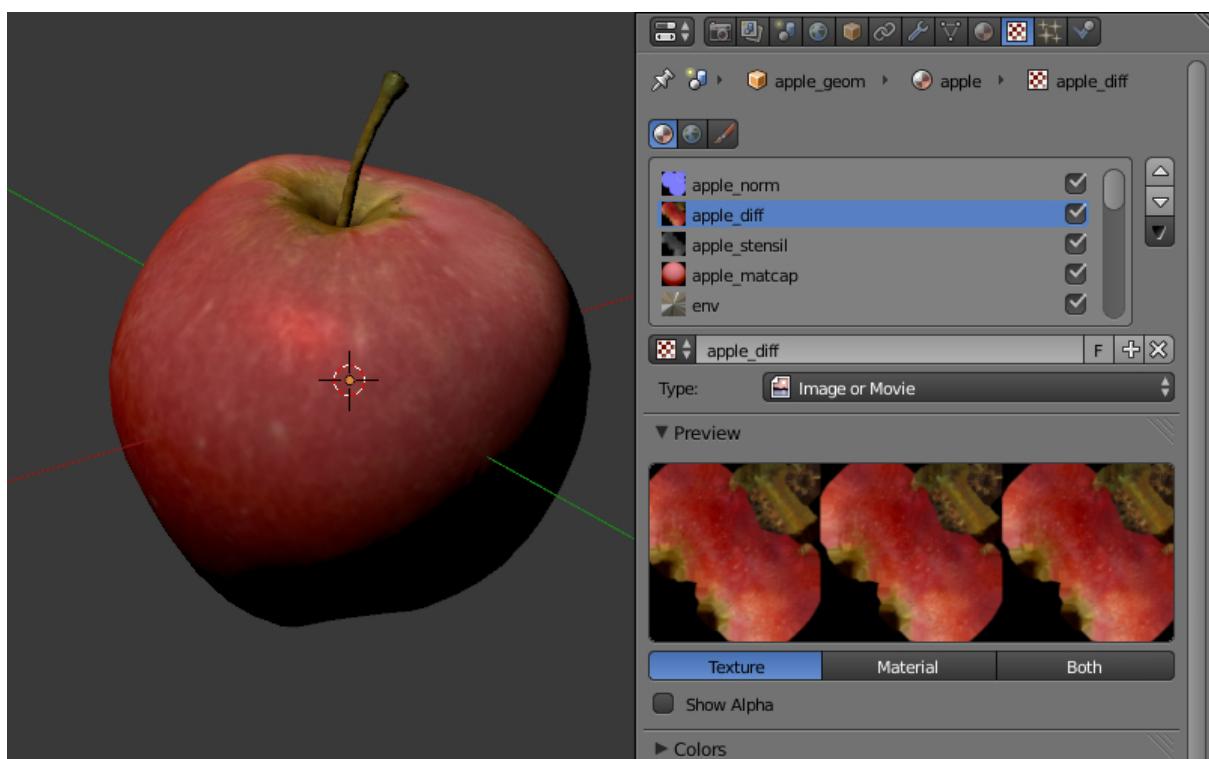
В случае обычных материалов для одной из смешируемых диффузных текстур поддерживается тип текстурных координат **Normal** (“matcap”).

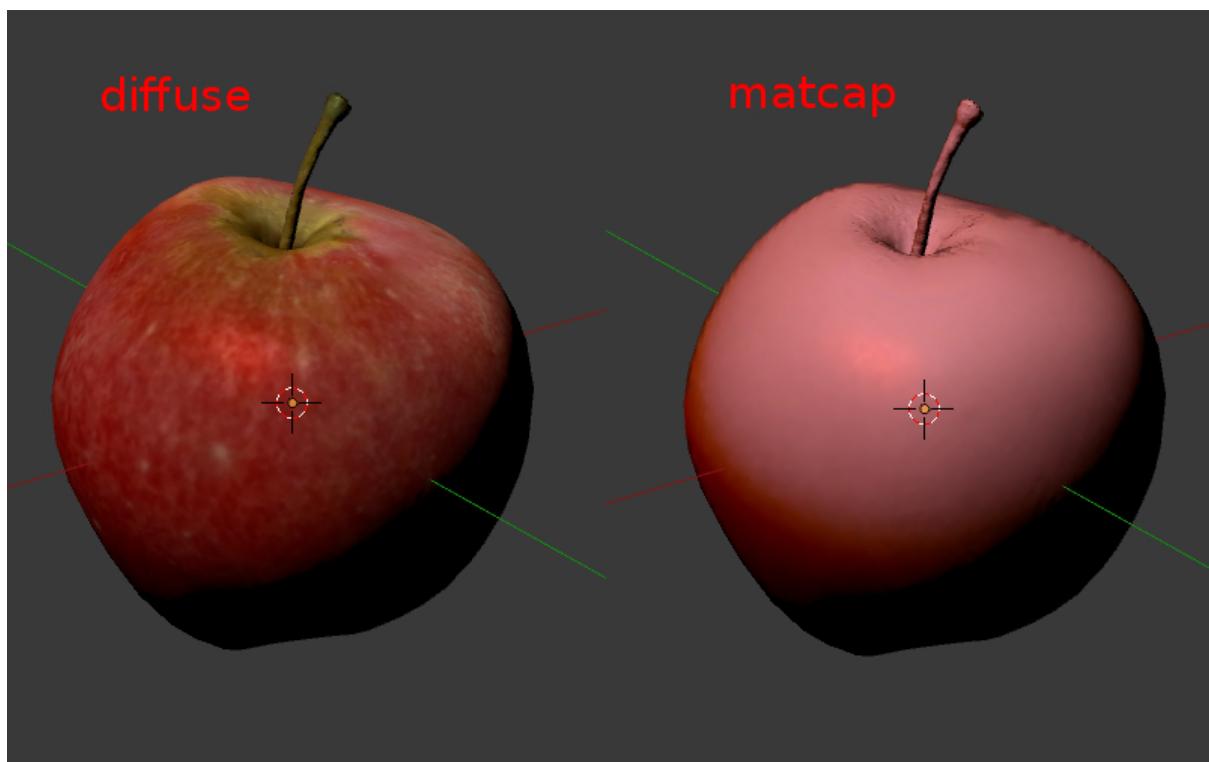
## 12.7.3 Ограничения

В случае обычных материалов движком интерпретируется только красный канал текстуры смеширования. Карта бликов или карта нормалей при их наличии смешианию не подвергаются. Настройка масштабирования **Mapping > Size** извлекается из первой текстуры и применяется ко всем остальным текстурам.

## 12.7.4 Пример

Материал яблока имеет текстуры: карту нормалей, диффузную текстуру с картой бликов в альфа-канале, карту смеширования, диффузную карту “matcap”, карту зеркального отражения.





## 12.8 Видео-текстуры

В качестве текстуры может применяться видеозапись. Для этого должен быть выбран тип текстуры **Image or Movie**.

---

**Примечание:** Видеотекстура поддерживает только воспроизведение видеоряда. Звуковое сопровождение следует накладывать, используя объект типа **SPEAKER**.

---

### 12.8.1 Поддерживаемые форматы (контейнеры):

- webm, кодек VP8 (Chrome, Firefox)
- m4v, кодек H.264 (Chrome, Safari, IE)
- ogv, кодек Theora (Chrome, Firefox)

В качестве базового формата рекомендуется использовать *WebM*, который является открытым стандартом, поддерживается многими браузерами, обеспечивает хорошее качество изображения.

---

**Примечание:** Файлы в форматах mp4 и ogg имеют различное расширение для звуковой и видео-информации: расширения *.mp4* и *.ogg* используются для звуков, *.m4v* и *.ogv* - для видео.

Конвертация ресурсов в различные форматы описывается в *соответствующем разделе*.

## 12.8.2 Настройка текстуры

При использовании видео-текстуры можно настроить следующие опции на панели **Texture > Image**:

**Image > Frames** Длина воспроизводимого фрагмента видеозаписи в кадрах.

**Image > Offset** Номер кадра, с которого будет начинаться воспроизведение видеозаписи.

**Image > Cyclic** Начинать воспроизводить видеозапись заново каждый раз после ее завершения.

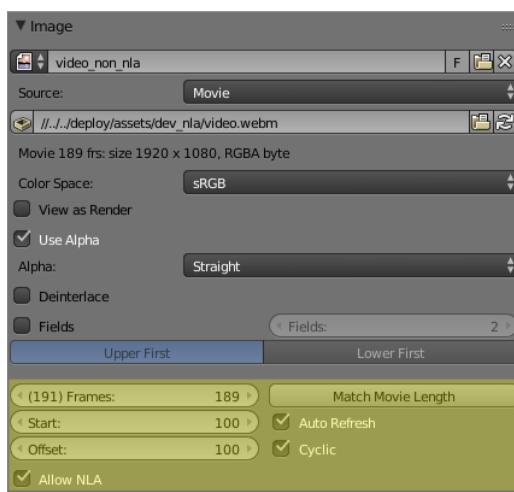
**Image > Allow NLA** Воспроизводить текстуру в рамках NLA-сценария. Необходимо также включить NLA глобально на сцене, активировав панель **Scene > NLA**. По умолчанию включено.

Для текстур, управляемых через NLA будет актуальна следующая опция:

**Image > Start** Задержка воспроизведения видеозаписи в кадрах при использовании нелинейной анимации.

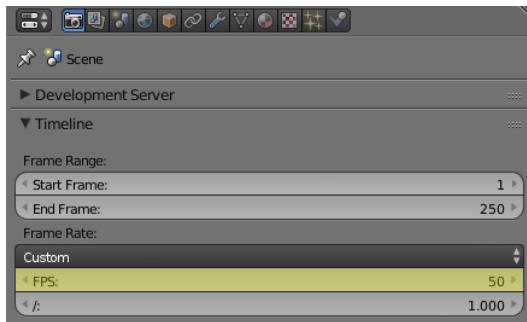
Для текстур, управляемых не через NLA будет актуальна следующая опция:

**Image > Auto Refresh** Воспроизводить видеозапись сразу после загрузки сцены.



Существует возможность ускоренного воспроизведения видеозаписей. Для этого необходимо задать значение свойства **Scene > Dimensions > Frame rate** отличным

от значения fps (кадров в секунду) для видеозаписи. При этом скорость воспроизведения видеозаписи расчитывается как отношение Frame rate сцены к fps видеозаписи.



**Примечание:** При использовании видео-текстур совместно с NLA кадры видео могут воспроизводиться не в точном соответствии со сделанными в Blender'е настройками. Возможно отставание видео в пределах 5-6 кадров, связанное с небольшой задержкой при старте/паузе тега <video>.

### 12.8.3 Особенности работы на мобильных устройствах

Присутствуют следующие особенности реализации на мобильных устройствах:

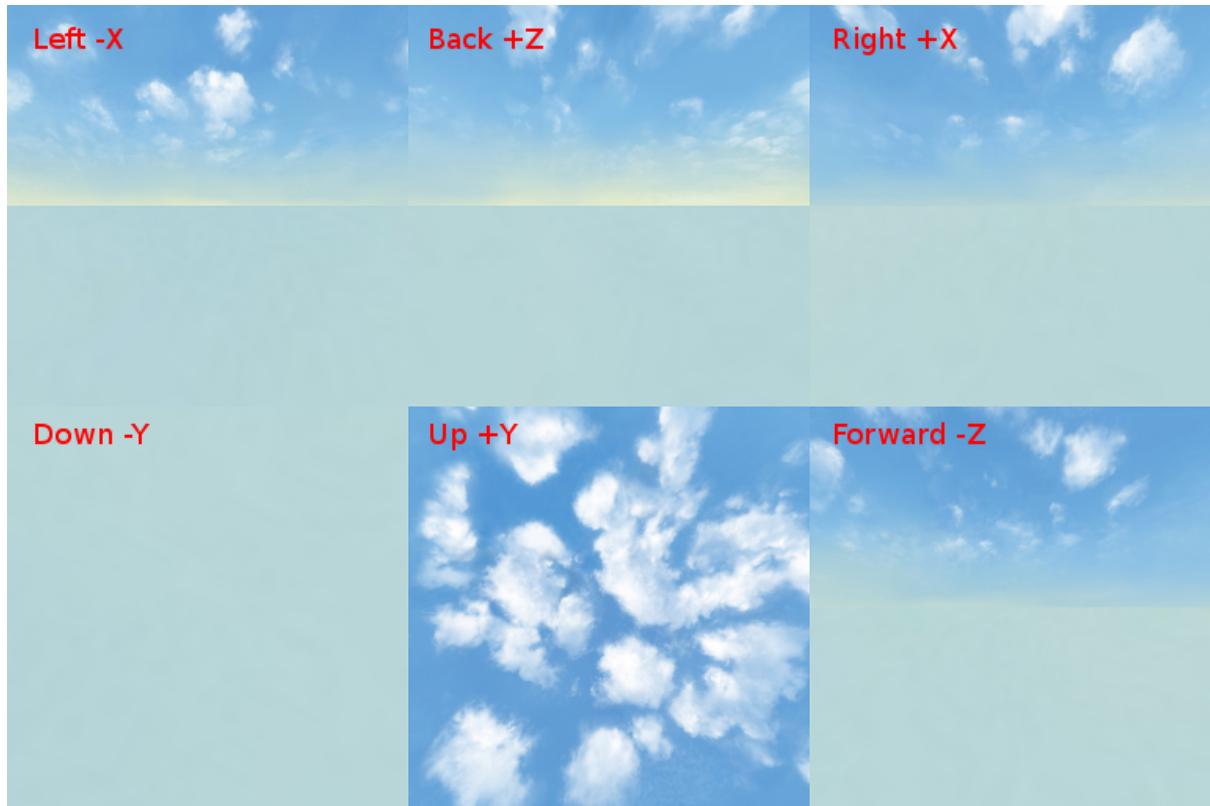
1. на iPhone используется специально разработанный видеоформат .seq, потому что данное устройство открывает все стандартные видеозаписи через свой видеопроигрыватель. Видео надо сконвертировать заранее, используя [наш конвертер](#).
2. на некоторых мобильных устройствах имеется поддержка воспроизведения только одного видеофайла.
3. не гарантируется стабильная работа при выставлении опции Offset в ненулевое значение.
4. не на всех устройствах поддерживается ускореное воспроизведение видео.
5. на iPad и iPhone отсутствует возможность управления громкостью звука видеофайла, поэтому следует убрать аудио дорожку из видео перед добавлением файла в Blender.

## 12.9 Карта окружения (environment map)

Применяется в качестве карты зеркального отражения (*mirror map*), в качестве статической текстуры неба (*skydome*), а также для реализации одной из методик освещения от окружающей среды.

В движке представлена кубической текстурой. Раstry для карт окружения должны содержать 6 спроецированных изображений окружающей среды, упакованных в 2 ряда по 3 (формат, используемый в Blender'е). Размер растром для каждого из изображений должен подчиняться правилу  $2^N$  (512, 1024 и т.п.).

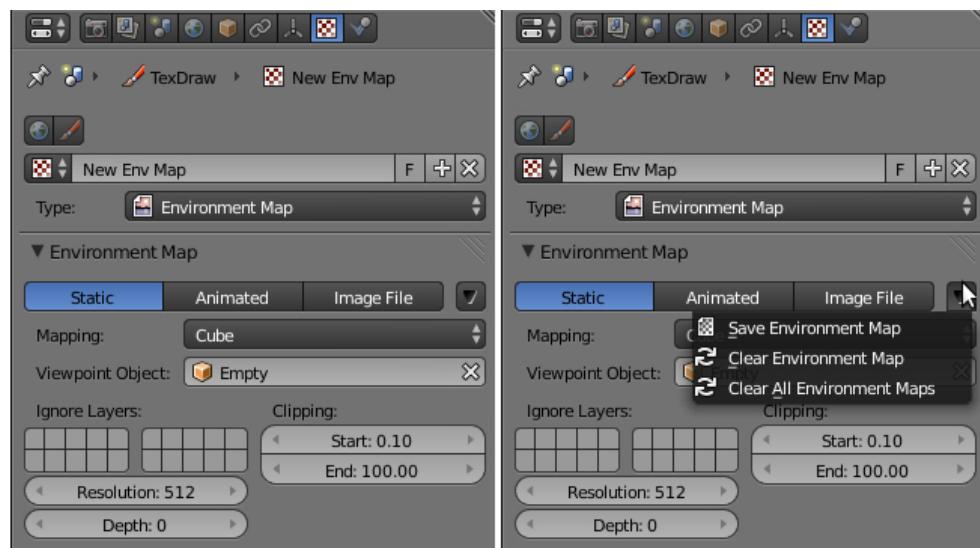
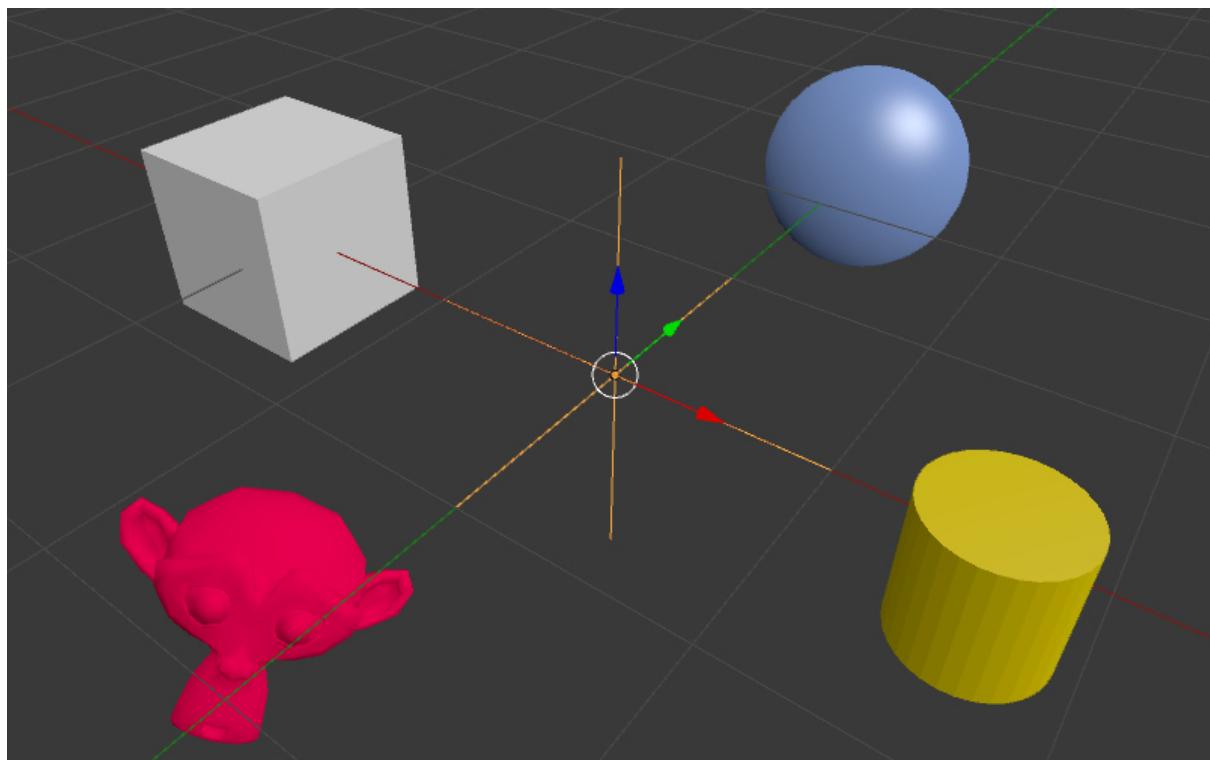
Во избежание проявления швов рекомендуется использовать формат без потери качества (PNG).



### 12.9.1 Создание карты окружения

Blender позволяет запекать сцену в карту окружения. Для этого:

1. Создать сцену для запекания.
2. Добавить пустой объект в предполагаемом центре обзора (**Add > Empty**).
3. Перейти во вкладку **World**, затем перейти во вкладку **Textures**, создать новую текстуру, выбрать тип **Environment Map**.
4. На панели **Environment Map** выбрать источник **Static**, выбрать созданный пустой объект в поле **Viewport Object**, установить разрешение  $2^N$  (512, 1024 и т.п.).
5. Выполнить рендеринг сцены F12 (требуется наличие камеры).
6. Сохранить карту окружения в файл.



## 12.10 Карта зеркального отражения (mirror map)

Применяется для визуализации отражающей способности поверхности. Представляет собой [карту окружения](#).

## 12.10.1 Активация

Выбрать тип текстуры (Type) Environment Map. Выставить опцию Shading > Mirror на панели Textures > Influence.

## 12.10.2 Дополнительные настройки

*Influence > Shading > Mirror* Степень влияния карты зеркального отражения.  
Значение по умолчанию 1.0.

См.также:

*Статическое отражение.*

## 12.11 Текстура неба (skydome)

Применяется для визуализации бесконечно удаленного окружения (например, небесного свода). Представляет собой *карту окружения*.

Также может применяться для реализации одной из методик *освещения от окружающей среды*.

## 12.11.1 Активация

Создать текстуру мира (world texture) с типом “Environment Map”. Выбрать опцию Export Options > Sky Texture Usage > SKYDOME, предварительно выставив опцию World > Render Sky на вкладке World.

---

**Примечание:** Поведение текстуры приближено к поведению рендера в Blender Internal. Поэтому, при стандартных настройках текстуры, она может не отображаться. Для отображения текстуры активируйте флаг в панели текстуры Influence > Horizon и установите значение Horizon равное 1.0.

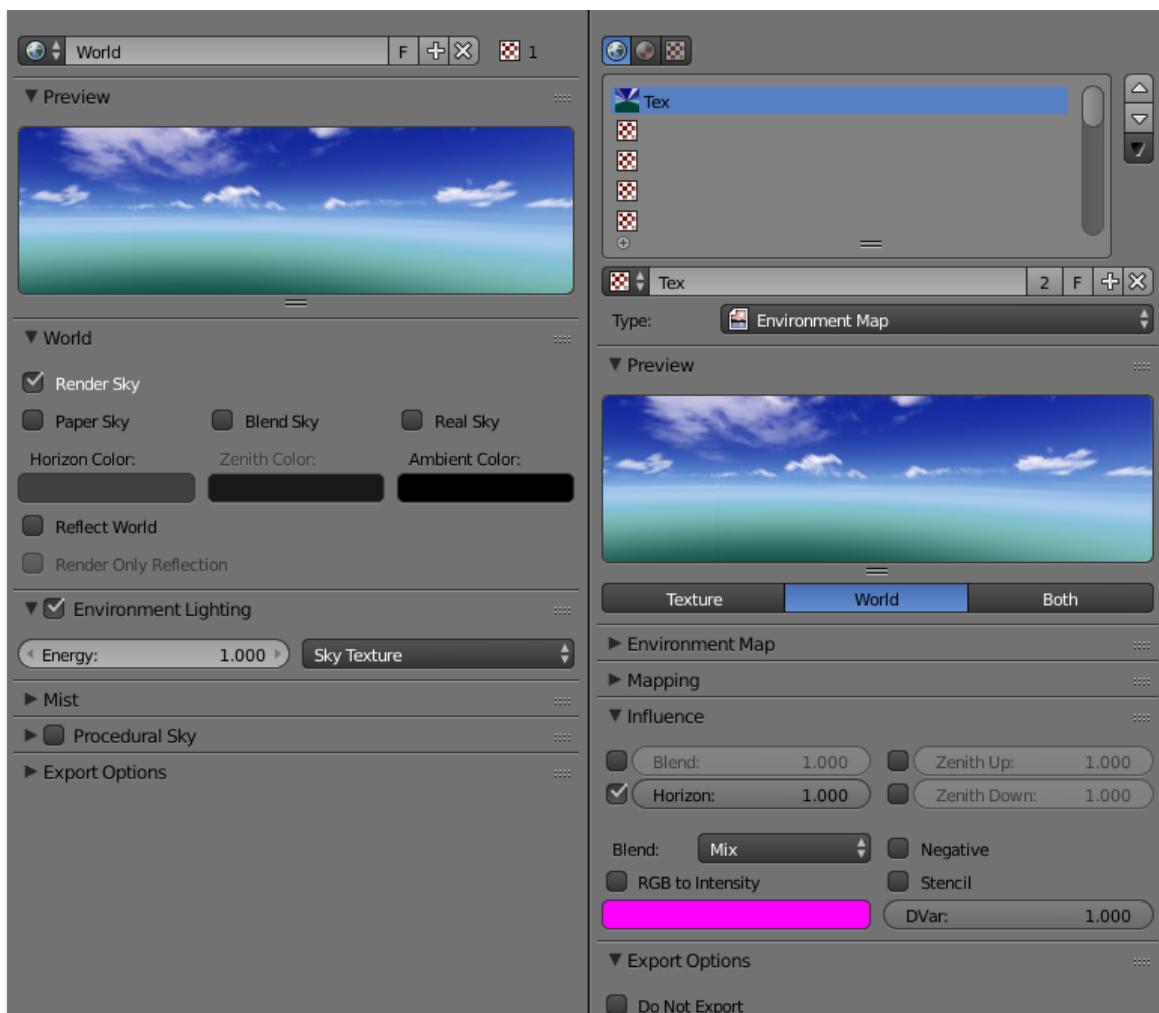
---

**Примечание:** Для имитации *освещения от окружающей среды* можно воспользоваться опцией Export Options > Sky Texture Usage > ENVIRONMENT\_LIGHTING. При этом необходимо также выбрать соответствующую опцию в настройках мира: Environment Lighting > Sky Texture.

---

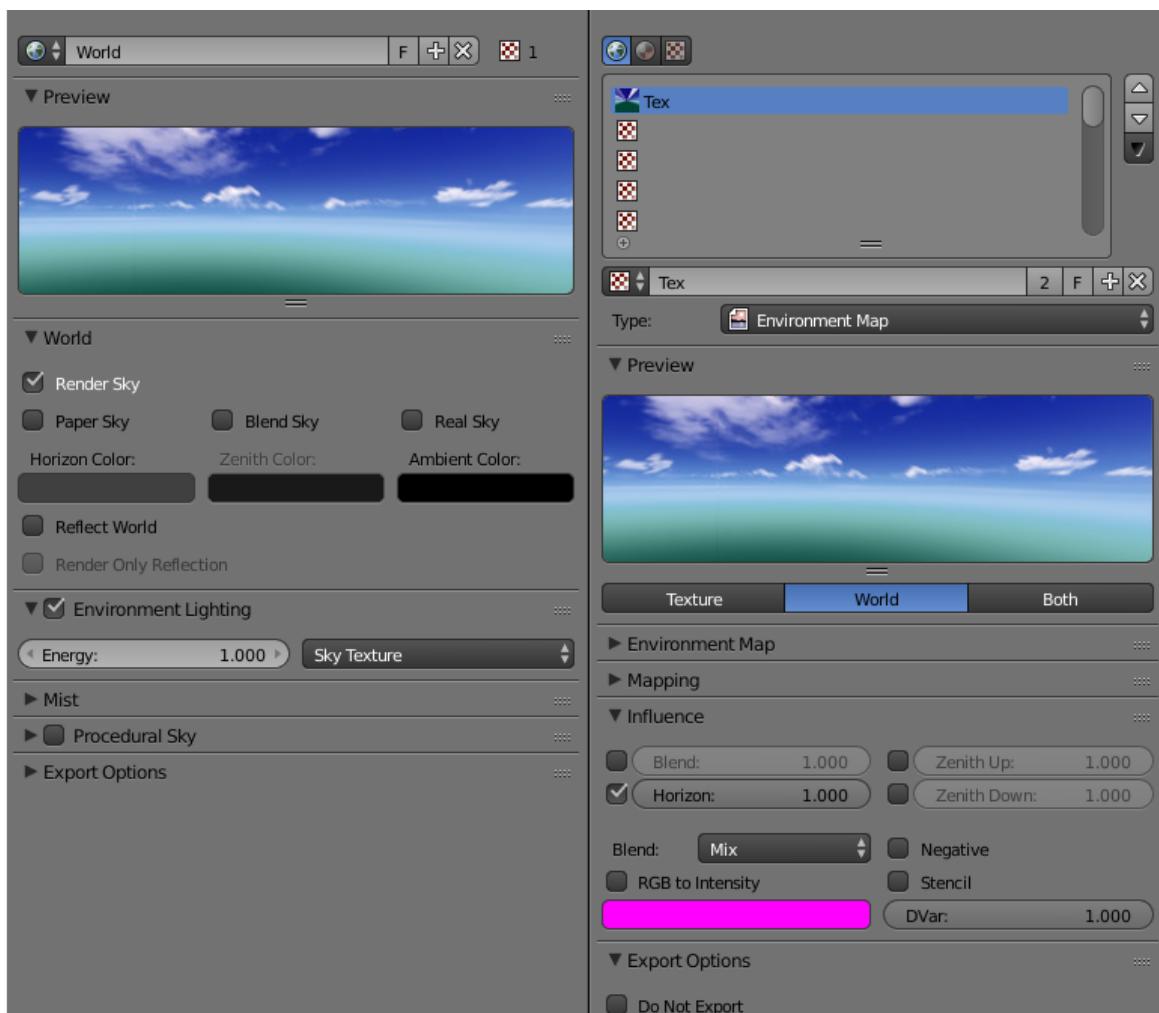
Для создания обоих эффектов от одной текстуры необходимо выставить опцию Export Options > Sky Texture Usage > BOTH.

---



## 12.11.2 Дополнительные настройки

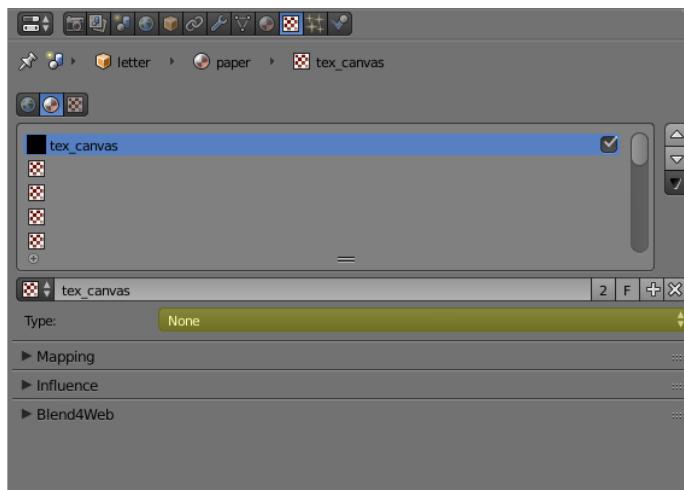
Движком поддерживаются параметры вкладки **Influence** для настройки отображения неба. Смешивание текстуры с цветом производится в зависимости от параметров **World > Horizon Color** (цвет горизонта) и **World > Zenith Color** (цвет зенита), а также флагов **Paper Sky**, **Blend Sky** и **Real Sky**. Поддерживаются все типы смешивания (**Mix**, **Add**, **Multiply** и т.д.).



**Примечание:** Параметры вкладки **Influence** влияют только на отображение неба. Они не влияют на отображение *освещения от окружающей среды*.

## 12.12 Особые типы текстур

Для использования особых типов текстур необходимо выставить тип текстуры **None** на панели **Textures**.



В меню **Textures > Export Options** имеется возможность задать основные свойства для данного типа текстуры:

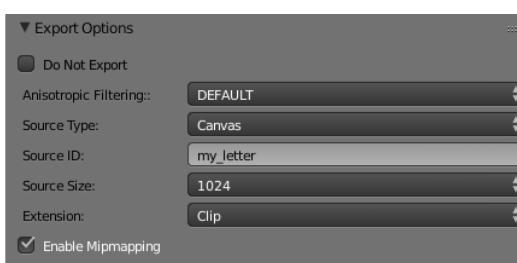
**Export Options > Source Type** Выбор особого типа текстуры: **Scene** - для рендеринга 3D сцены в текстуру, **Canvas** - использование HTML-элемента `<canvas>` и **None** - отсутствие текстуры.

**Export Options > Source ID** Имя сцены, которая будет отображена в текстуре, в случае 3D сцены либо идентификатор HTML-элемента `<canvas>` в случае Canvas-текстуры.

**Export Options > Source Size** Разрешение текстуры.

**Export Options > Extension** Режим интерпретации текстурных координат. Значение по умолчанию **Repeat**.

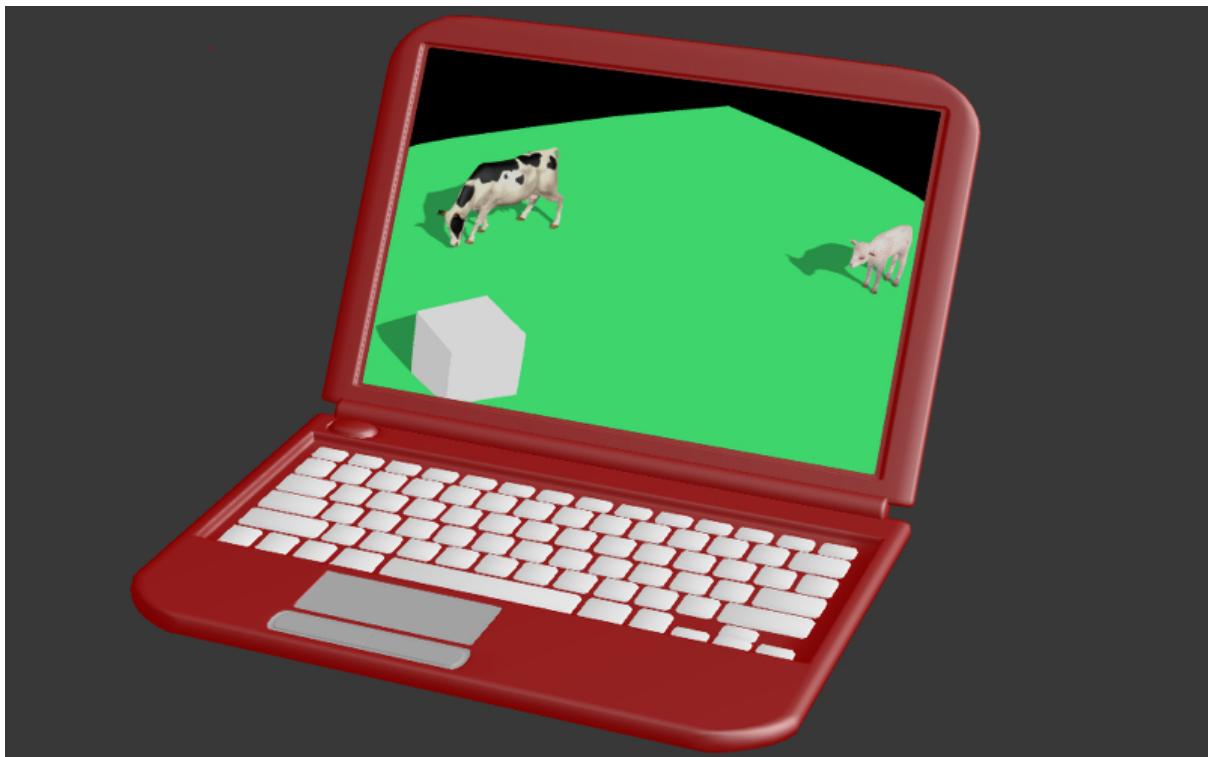
**Export Options > Enable Mipmapping** Включение MIP-текстурирования для Canvas-текстуры. По умолчанию включено.



### 12.12.1 3D сцена

Изображение 3D сцены может быть использовано в качестве текстуры на объекте другой (“главной”) сцены.

1. Создать дополнительную сцену-источник, переименовать для удобства, создать **World**, добавить нужные объекты, настроить вид из камеры.
2. В главной сцене для текстуры целевого объекта выставить тип **None**, в поле **Export Options > Source Type** указать тип **Scene**. В поле **Export Options > Source ID** указать название сцены-источника. В поле **Export Options > Source Size** указать размер текстуры в пикселях.



Также поддерживается циклическое отображение сцен друг в друга.

---

**Примечание:** В проекте должна присутствовать хотя бы одна сцена, которую не отображают другие.

---

### 12.12.2 Canvas

В качестве текстуры используется HTML-элемент `<canvas>`. Его редактирование осуществляется при помощи API.

В главной сцене для текстуры целевого объекта выставить тип **None**, в поле **Export Options > Source Type** указать тип **Canvas**. В поле **Export Options > Source ID** указать идентификатор объекта типа **Canvas**. В поле **Export Options > Source Size** указать размер текстуры в пикселях.

Для работы с этими текстурами используется модуль `textures`. Пример работы с ним описан ниже.

```
var m_tex = require("textures");
...
var ctx = m_tex.get_canvas_texture_context("canvas_id");
...
// operations with canvas context
...
m_tex.update_canvas_texture_context("canvas_id");
```

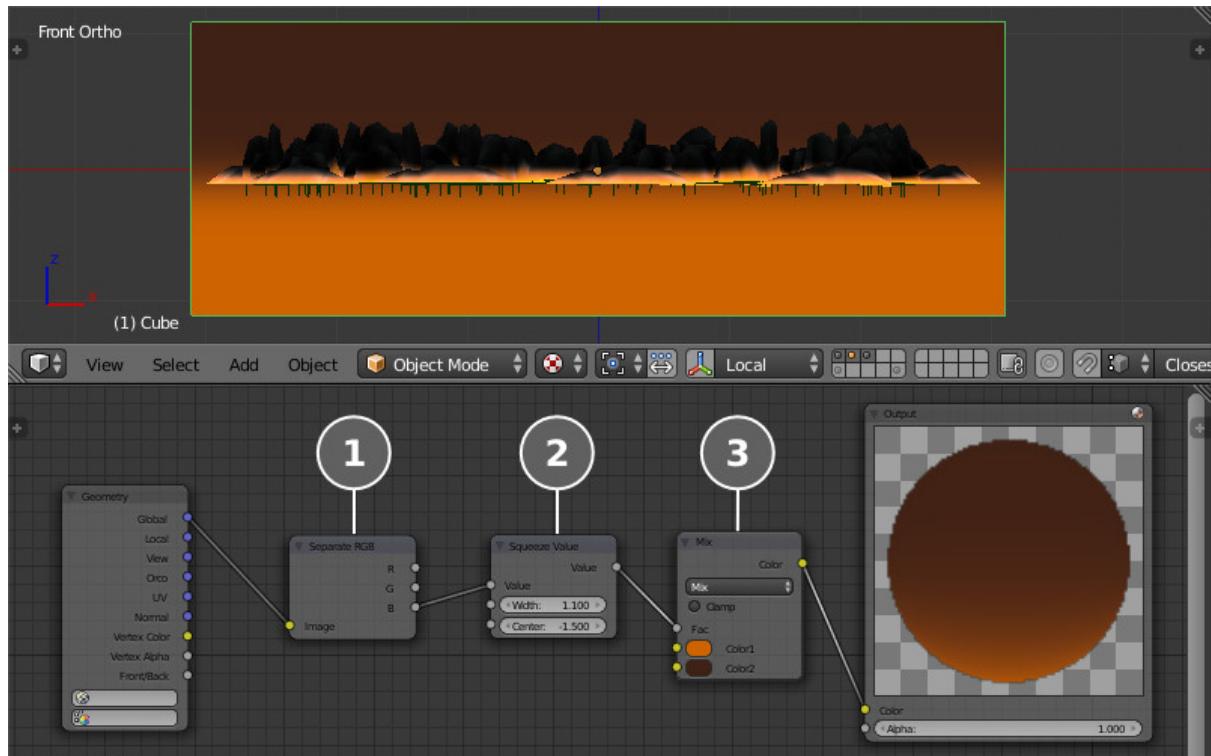
Для получения контекста используется функция `get_canvas_texture_context()`, куда передается идентификатор “`canvas_id`”, задаваемый пользователем в Blender. После преобразований с контекстом вызывается функция `update_canvas_texture_context()`, которая осуществляет визуализацию изменений элемента с идентификатором “`canvas_id`”.

---

## Нодовые материалы

---

Шейдерные ноды (Shader Nodes) существенно расширяют возможности стандартных материалов Blender, позволяя представить освещение как серию базовых преобразований.



### 13.1 Стандартные ноды

Полностью поддерживаются все возможности Blender, за исключением следующих случаев:

- **Geometry** - не поддерживается выход **Vertex Alpha**.
- **Lamp data** - не поддерживается выход **Shadow**.

- **Material, Extended Material** - допускается не больше одной ноды на материал, не поддерживаются входы **Refl**, **Ambient**, **SpecTra** и выход **A0**.
- **RGB Curves, Vector Curves, ColorRamp** и **Cycles**-ноды поддерживаются ограниченно.

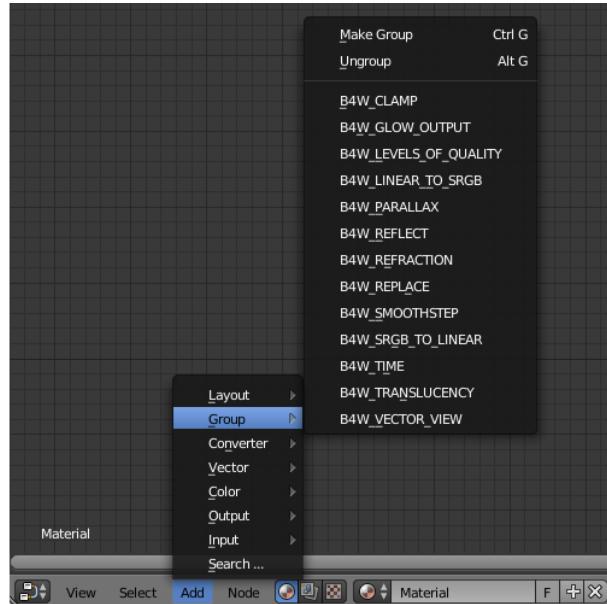
Кроме того, в контексте рендеринга в реальном времени, следует учитывать низкую производительность некоторых нод. Не рекомендуется к использованию:

- **Hue/Saturation**
- **MixRGB** типы **Burn**, **Dodge**, **Value**, **Saturation**, **Hue**, **Color**

Не рекомендуется создавать сложные материалы, особенно использующие большое количество нод **Geometry** и **Texture**.

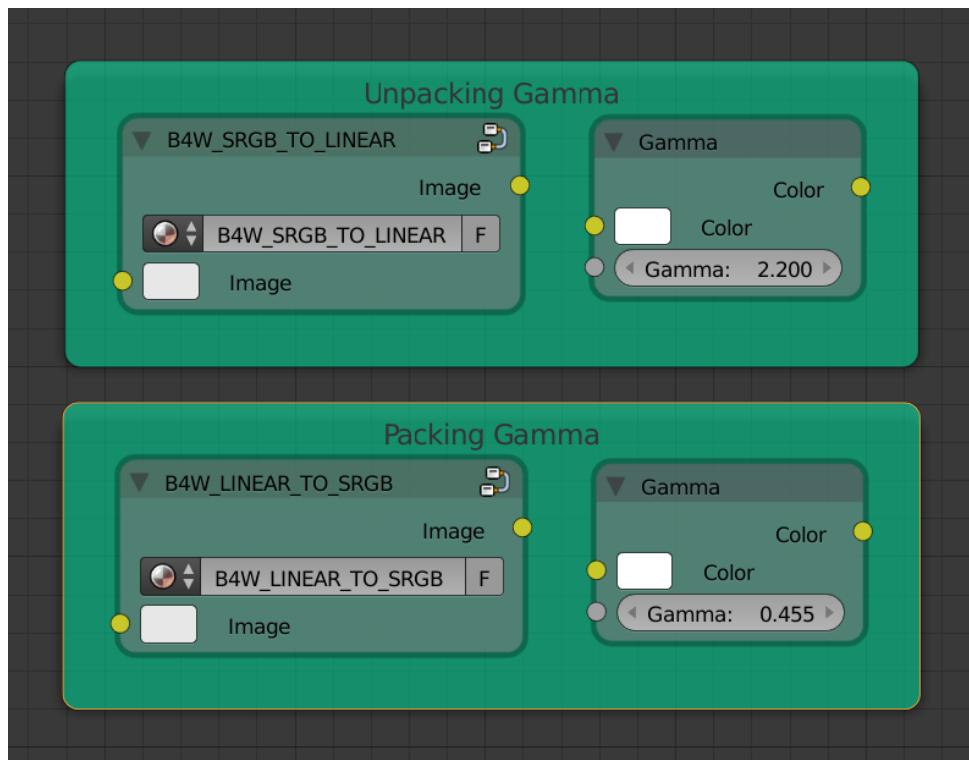
## 13.2 Дополнительные ноды

Дополнительные ноды расширяют функционал стандартных с учётом специфики работы движка. Ноды оформляются в виде нодовых групп (**Node groups** или **Node tree**) со специально выбранным именем и форматом входов. Для удобства, все дополнительные ноды добавляются в blend-файл при его открытии.



### 13.2.1 B4W\_LINEAR\_TO\_SRGB и B4W\_SRGB\_TO\_LINEAR

Преобразование цвета из линейного цветового пространства в пространство sRGB и наоборот. Функционал объявлен устаревшим с версии 15.04 - вместо этих нод должна использоваться нативная нода **Gamma**.

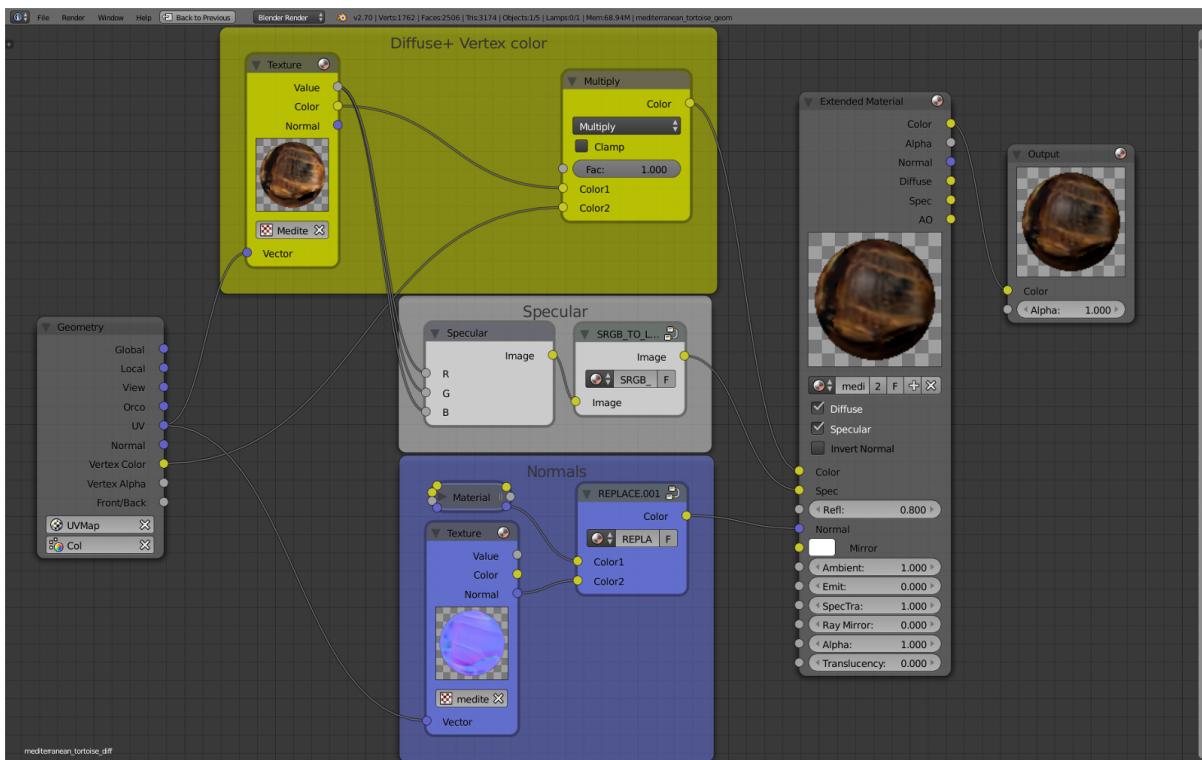


См.также:

*Коррекция в нодовых материалах*

### 13.2.2 B4W\_REPLACE

Осуществляет замену входов в зависимости от того, в какой среде (viewport Blender'a или движок) в данный момент работает текущая сцена. При работе в Blender вход **Color1** подключается к выходу **Color**, вход **Color2** игнорируется. При работе в движке входы меняются местами (**Color1** игнорируется, **Color2** подключается к выходу). Нода предназначена для отображения во viewport'е одной конструкции нодов, а в движке - другой.



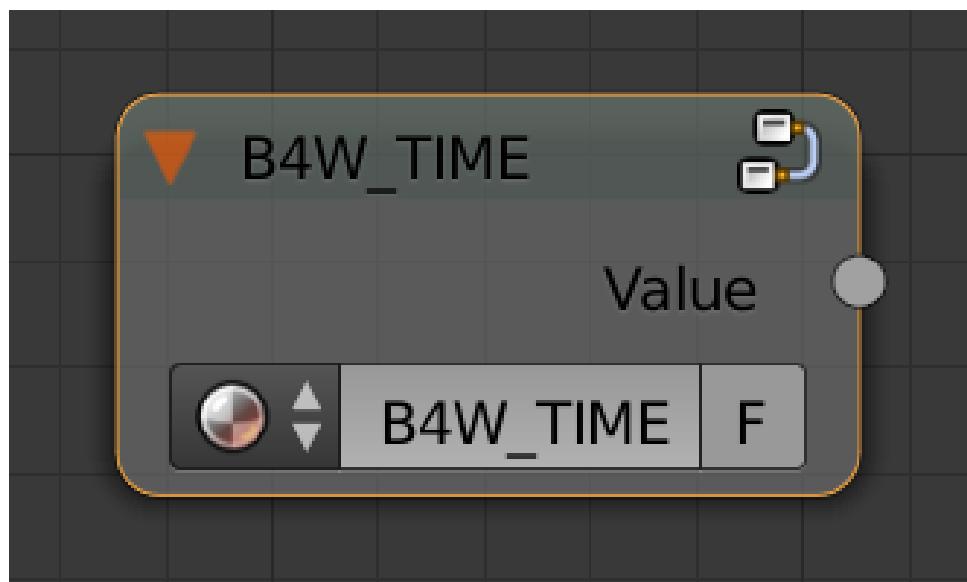
Используется, как правило, для подключения карт нормалей. Нодовые материалы Blender'a не поддерживают тангенциальное пространство координат, в связи с чем единственный способ корректного отображения карт нормалей во viewport'e - их подключение внутри нод Material.

### 13.2.3 B4W\_CLAMP

Осуществить операцию ограничения над входом. В результате, все элементы вектора на выходе получают значения от 0 до 1 включительно.

### 13.2.4 B4W\_TIME

Осуществляет отсчет времени с момента старта движка в секундах. Может использоваться для анимации любых параметров в нодовых материалах - UV координат, факторов смешивания, прозрачности и т.п.



**См.также:**

[Анимация под Value и RGB](#)

### 13.2.5 B4W\_VECTOR\_VIEW

Осуществить преобразование вектора в пространство камеры. Преобразование необходимо, поскольку при работе в движке большинство векторов определены в мировой системе координат (например нормали, направления источников освещения и т.д.). Преобразованный таким образом вектор нормали используется только для различных эффектов и не должен подключаться к входу ноды **Material** или **Extended Material**.

### 13.2.6 B4W\_PARALLAX

Реализует смещение текстурных координат в соответствии с картой высот.

#### Входные параметры

**UV** Исходные текстурные координаты.

**Height Map** RGBA текстура с картой высот в альфа канале.

**Scale** Коэффициент смещения текстурных координат.

**Steps** Количество шагов при генерации смещенных текстурных координат. Чем больше данное значение, тем выше качество получаемой текстуры.

**Lod Distance** Максимальное расстояние от камеры, на котором виден эффект.

#### Выходные параметры

**UV** Измененные текстурные координаты, которые используются как вход для текстурных нод.

### 13.2.7 B4W \_ TRANSLUCENCY

Реализует эффект полупрозрачности (только по отношению к источникам света) для тонких объектов, таких как ткань, листва, бумага и др. Эффект состоит из двух частей: засвечивание обратной по отношению к источнику стороны объекта и появление светового пятна непосредственно в том месте, где должен был находиться источник.

#### Входные параметры

**Color** Одноканальная текстура, определяющая неоднородность материала, белый - максимальный эффект просвечивания, черный - его отсутствие. По умолчанию используется белый.

**Backside Factor** Коэффициент коррекции цвета материала на обратной от источника света стороне. Основан на визуальном эффекте большей насыщенности цвета при просвечивании.

- *Backside Factor < 1* - коррекция в сторону освещения
- *Backside Factor = 1* - без коррекции
- *Backside Factor > 1* - коррекция в сторону затемнения

Значение по умолчанию: 1.

**Spot Hardness** Коэффициент размытия светового пятна. При увеличении размеры пятна уменьшаются, края становятся более резкими. Значение по умолчанию: 1000.

**Spot Intensity** Интенсивность светового пятна. При увеличении становится более ярким. Значение по умолчанию: 1.

**Spot Diffuse Factor** Коэффициент влияния диффузного цвета материала на цвет светового пятна.

- *Spot Diffuse Factor = 0* - световое пятно имеет диффузный цвет
- *Spot Diffuse Factor = 1* - световое пятно имеет белый цвет

Значение по умолчанию: 1.

## Выходные параметры

**Translucency** Выход должен быть подключен ко входу **Translucency** ноды **Extended Material**.

---

**Примечание:** Возможно некорректное поведение ноды, если используются *отредактированные нормали*.

---

## 13.2.8 B4W\_REFRACTION

Реализует эффект преломления.

### Входные параметры

**Normal** Карта нормалей, по которой происходит возмущение (сдвиг).

**Refraction bump** Коэффициент возмущения (сдвига) текстуры сцены позади объекта.

Значение по умолчанию: 0.001.

### Выходные параметры

**Color** Текстура сцены позади объекта с внесённым возмущением.

---

**Примечание:** Для отображения эффекта необходимо переключить опцию **Refractions** на панели **Render > Reflections and Refractions** в состояние **AUTO** или **ON**. Объект должен быть с типом прозрачности **Alpha Blend**.

---

**См.также:**

*Прозрачность*

## 13.2.9 B4W\_LEVELS\_OF\_QUALITY

Устанавливает значение выходного цвета в зависимости от установленного профиля качества изображения.

**См.также:**

*Профили качества изображения*

### Входные параметры

**HIGH** Значение, которое будет подано в выходной параметр **Color** при высоком и максимальном качестве изображения.

**LOW** Значение, которое будет подано в выходной параметр *Color* при низком качестве изображения.

#### Выходные параметры

*Color* Значение выходного цвета.

---

**Примечание:** Для отображения во виджете Blender'a используется входной параметр *HIGH*.

---

### 13.2.10 B4W\_SMOOTHSTEP

Осуществить мягкую интерполяцию двух значений, исходя из первого значения.

#### Входные параметры

*Value* Значение, на основе которого осуществляется интерполяция.

*Edge0* Первое значение для интерполяции.

*Edge1* Второе значение для интерполяции.

#### Выходные параметры

*Value* Сглаженное значение.

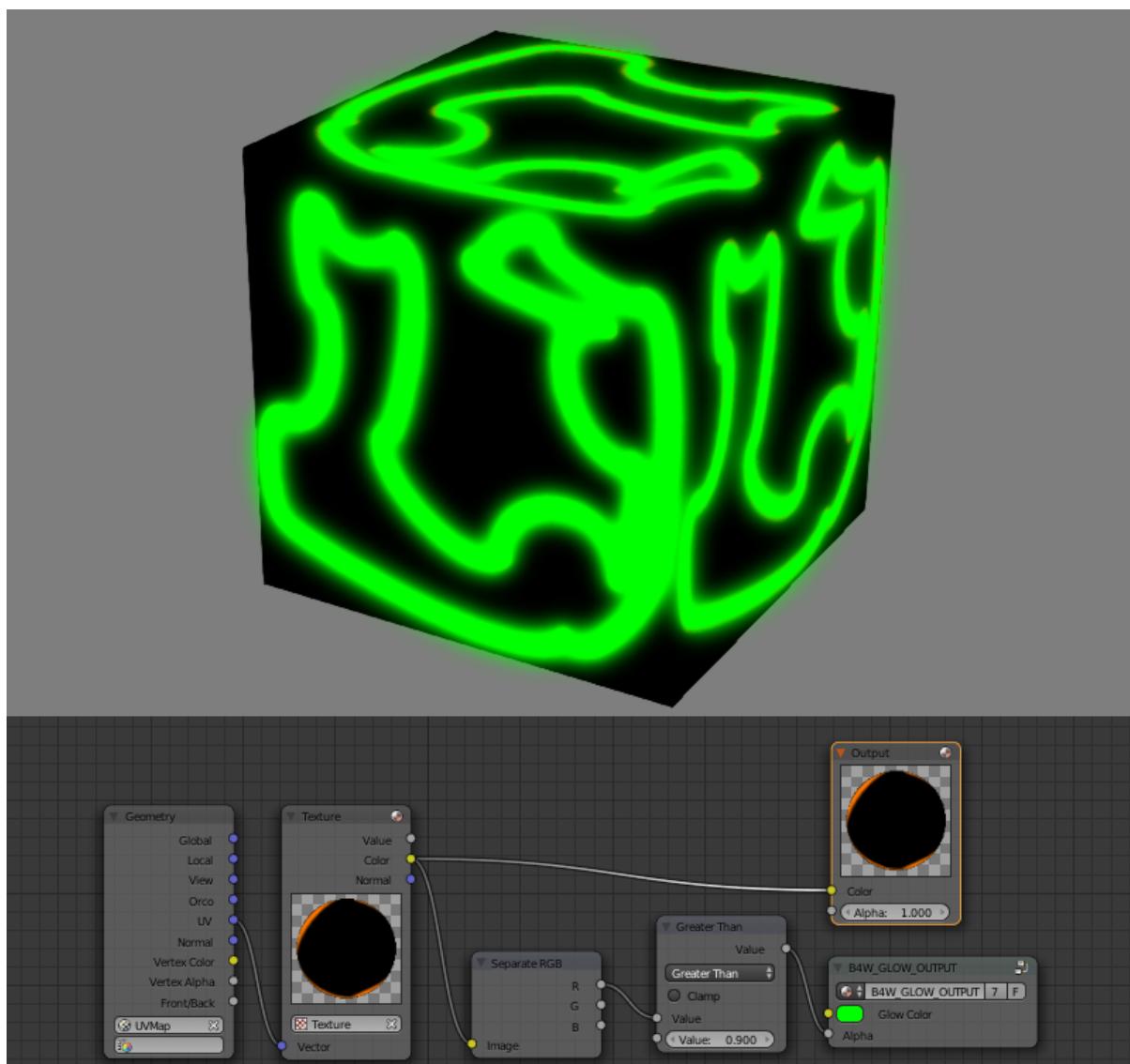
---

**Примечание:** Для корректной интерполяции входное значение ''Value'' должно лежать в диапазоне между ''Edge0'' и ''Edge1''.

---

### 13.2.11 B4W\_GLOW\_OUTPUT

Применяет *эффект свечения (Glow)* к нодовому материалу. Помимо ноды *B4W\_GLOW\_OUTPUT* в нодовом материале должна присутствовать нода *Output*.



## Входные параметры

**Glow Color** Цвет свечения.

**Factor** Степень свечения.  $Factor \in [0, 1]$ .

- $Factor = 0$  - свечение отсутствует.
- $Factor \in (0, 1]$  - свечение цветом *Glow Color*.

---

## Освещение, тени и фон

---

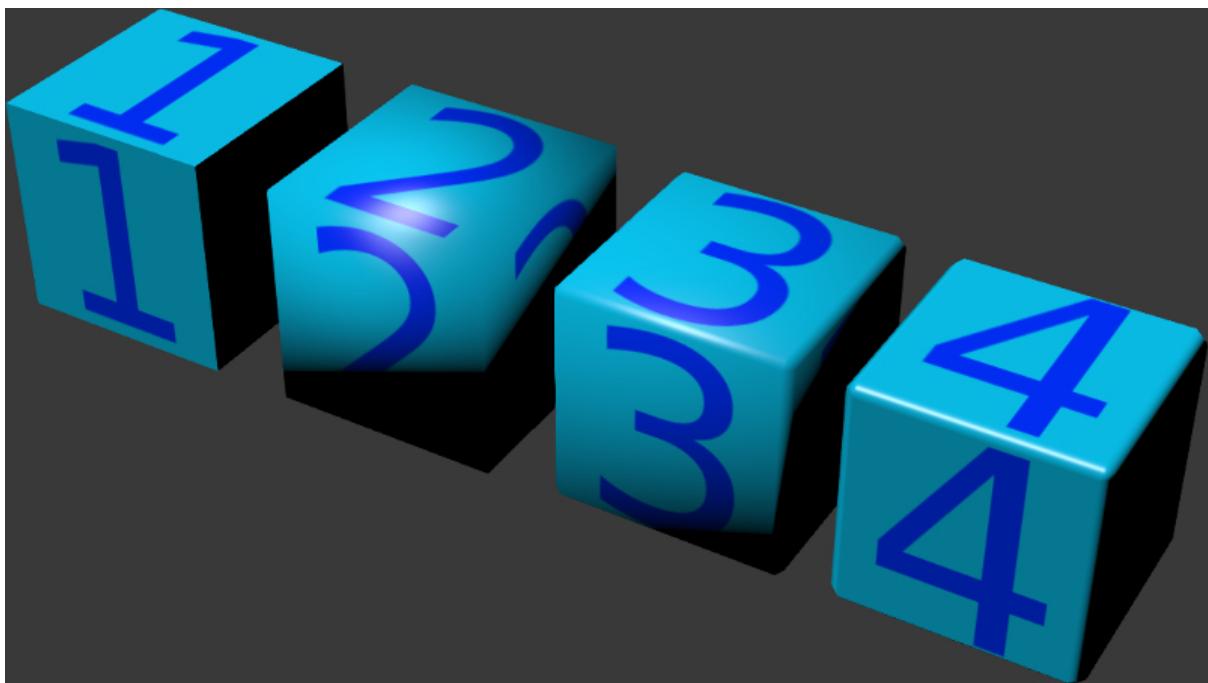
### 14.1 Типы затенения

Результат расчета освещения (затенения) зависит от направления векторов нормалей. Поддерживаются стандартные типы затенения, использующиеся в Blender, Shading: Flat (используются нормали граней) и Shading: Smooth (используются интерполированные нормали вершин), а также их комбинации.



Если желаемый результат недостижим стандартными средствами, можно воспользоваться [редактором нормалей](#).

Результат применения различных типов затенения и использования редактора нормалей:



1. Flat Shading
2. Smooth Shading
3. Smooth Shading + фаска
4. Smooth Shading + фаска + редактирование нормалей

## 14.2 Редактор нормалей

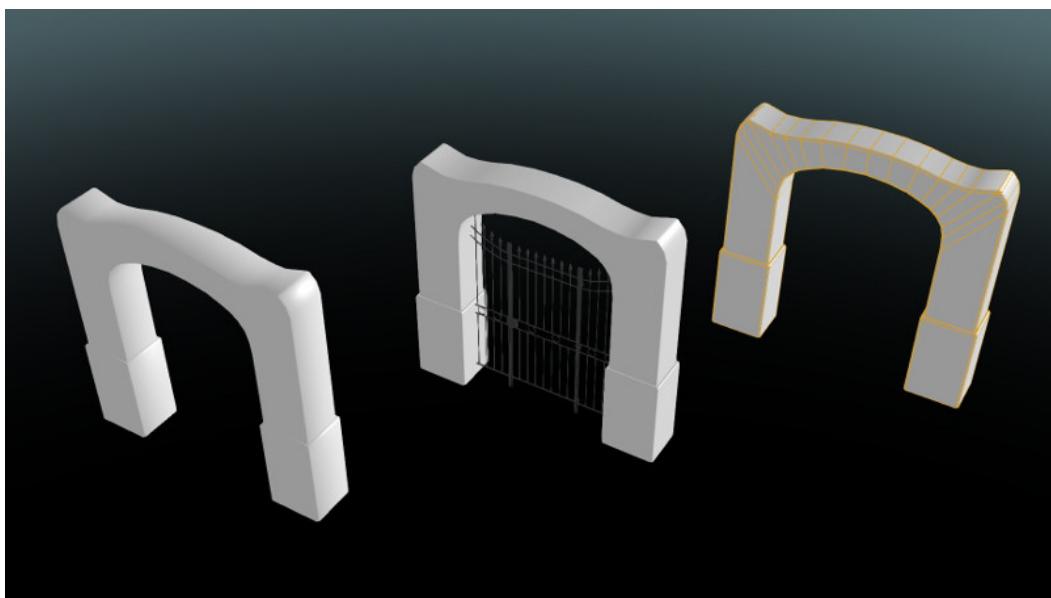
Редактирование нормалей - достаточно простой и эффективный способ изменить собственное затенение трехмерной модели, не усложняя ее геометрию.

С помощью редактора нормалей, в некоторых случаях, можно достичь результатов, аналогичных применению *карт нормалей*. При этом метод редактирования нормалей является более предпочтительным с точки зрения потребления вычислительных ресурсов и видео-памяти.

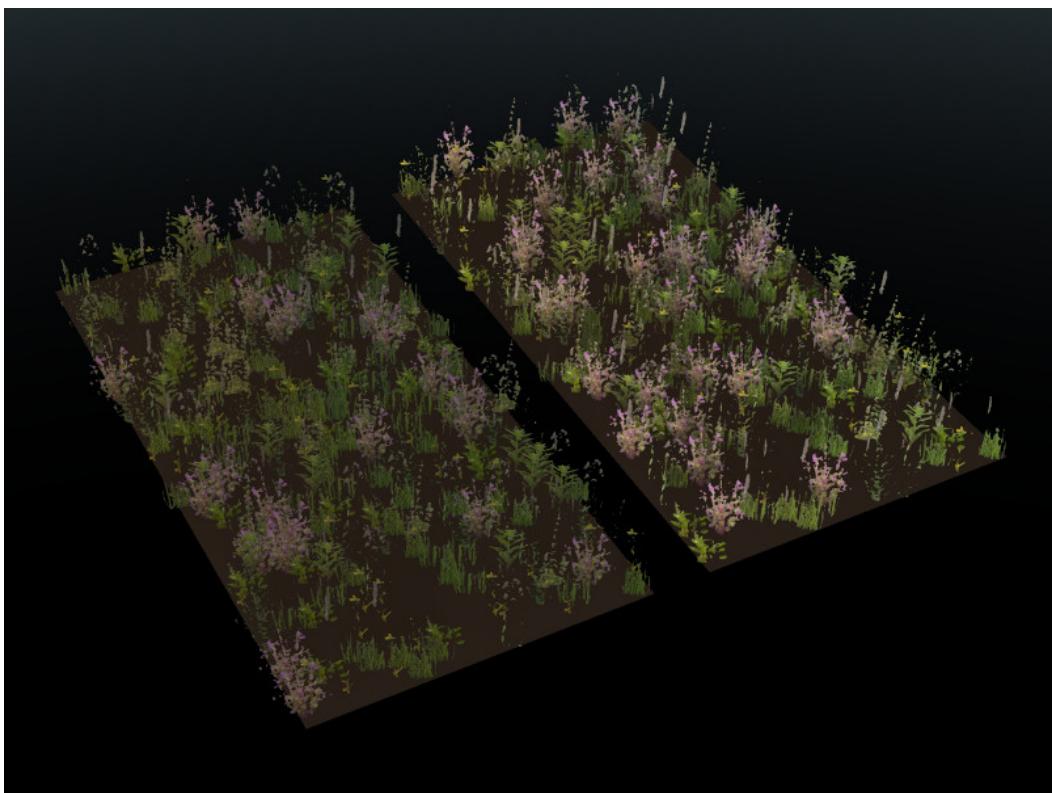
Пример работы редактора нормалей:



Слева - обычное затенение геометрии, справа - дерево с редактированными нормалями.



Слева - ворота с обычным затенением, в центре - ворота с редактированными нормалями, справа - геометрия модели ворот с сеткой.



Слева - обычное затенение геометрии травы, справа - трава с редактированными нормалями.



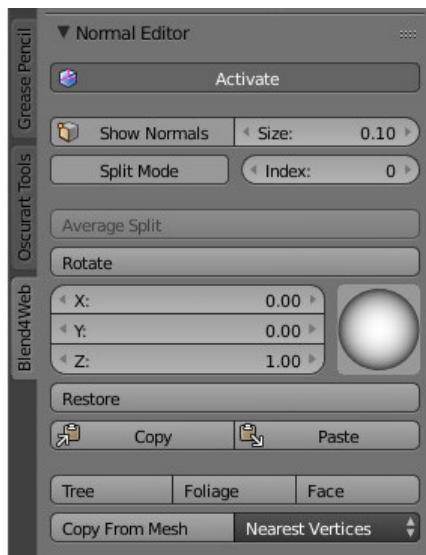
Слева - очки с обычным затенением, справа - затенение геометрии с редактированными нормалями.

### 14.2.1 Основные особенности редактора нормалей:

1. для хранилища массива данных с направлениями вертекных нормалей используется “родной” блок данных Blender (появился в версии 2.74);
2. редактирование нормалей и отображение затенения теперь происходит в режиме редактирования (**Edit Mode**);
3. все изменения записываются сразу же автоматически;
4. вращение нормали выделенного вертекса можно производить непосредственно в окне Viewport’а по горячим клавишам **Shift+Ctrl+R**, подобно остальным операциям вращения в Blender;
5. редактированные нормали экспортируются автоматически.

### 14.2.2 Интерфейс редактора вертекных нормалей

Интерфейс редактора нормалей располагается на панели инструментов Blend4Web > **Normal Editor**. Перед началом работы редактора для объекта необходимо включить режим затенения **Shading: Smooth** и нажать кнопку **Activate** на панели редактора нормалей или выставить галочку **Auto Smooth** в настройках меша.



### 14.2.3 Activate

#### *Описание*

Новая кнопка **Activate** включает режим редактирования вертекных нормалей.

#### *Использование*

Достаточно в режиме **Edit Mode** нажать кнопу **Activate**, и можно приступать к редактированию вертекных нормалей. Пока она остаётся в активном режиме, затенение объекта и его экспорт будут осуществляться с учетом редактированных вертек-

ных нормалей. Другими словами, после внесения изменений, для их отображения во Viewport'e Blender и в движке Blend4Web кнопка должна быть оставлена активной.

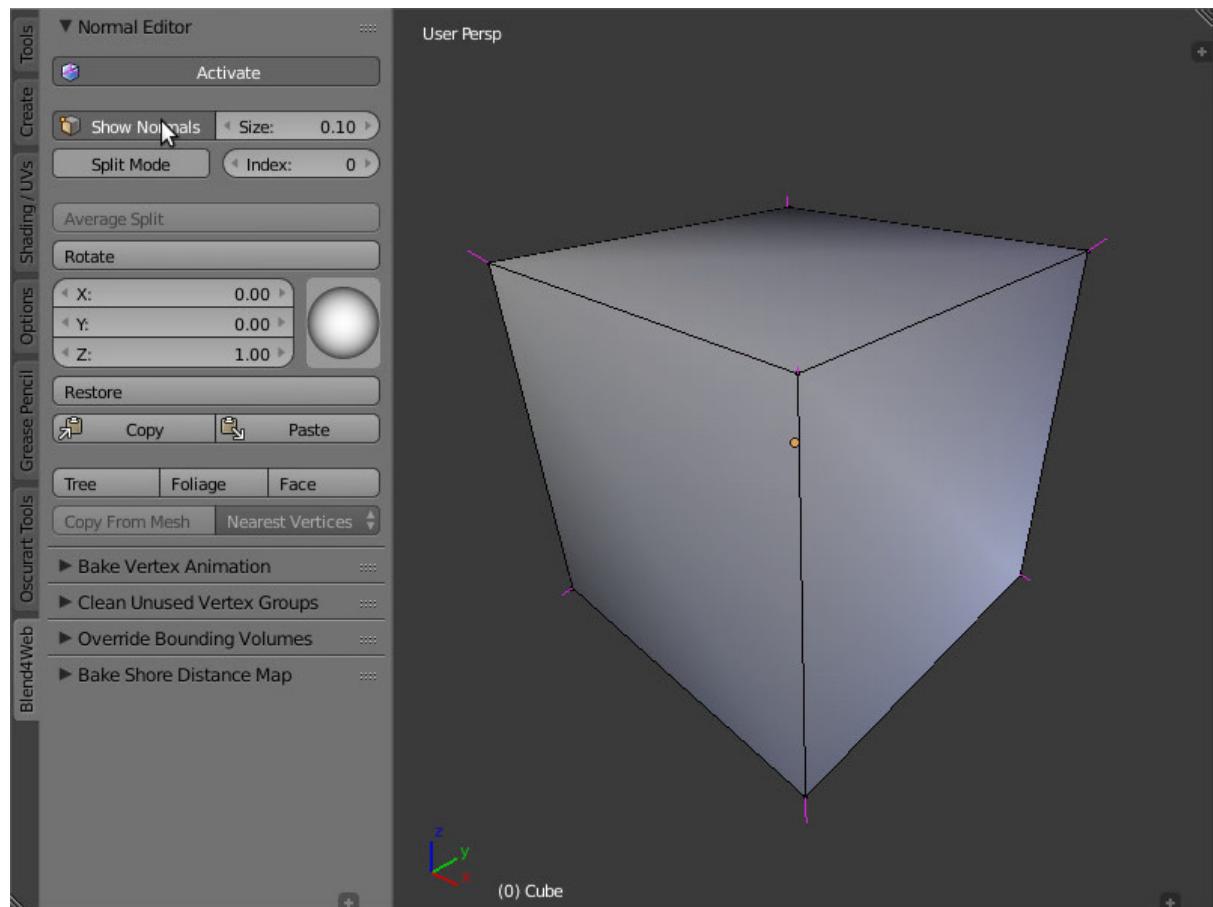
#### 14.2.4 Show Normals

##### *Описание*

Для удобства в панель продублирована оригинальная кнопка Blender, включающая отображение маркеров вертексных нормалей во Viewport'e, а так же настройка их длины.

##### *Использование*

Достаточно просто включить кнопку **Show Normals** прямо на панели редактора вертексных нормалей Blend4Web или в правой панели Blender в разделе **Mesh Display** и настроить комфортную длину маркера в поле **Size**.



#### 14.2.5 Вращение

##### *Описание*

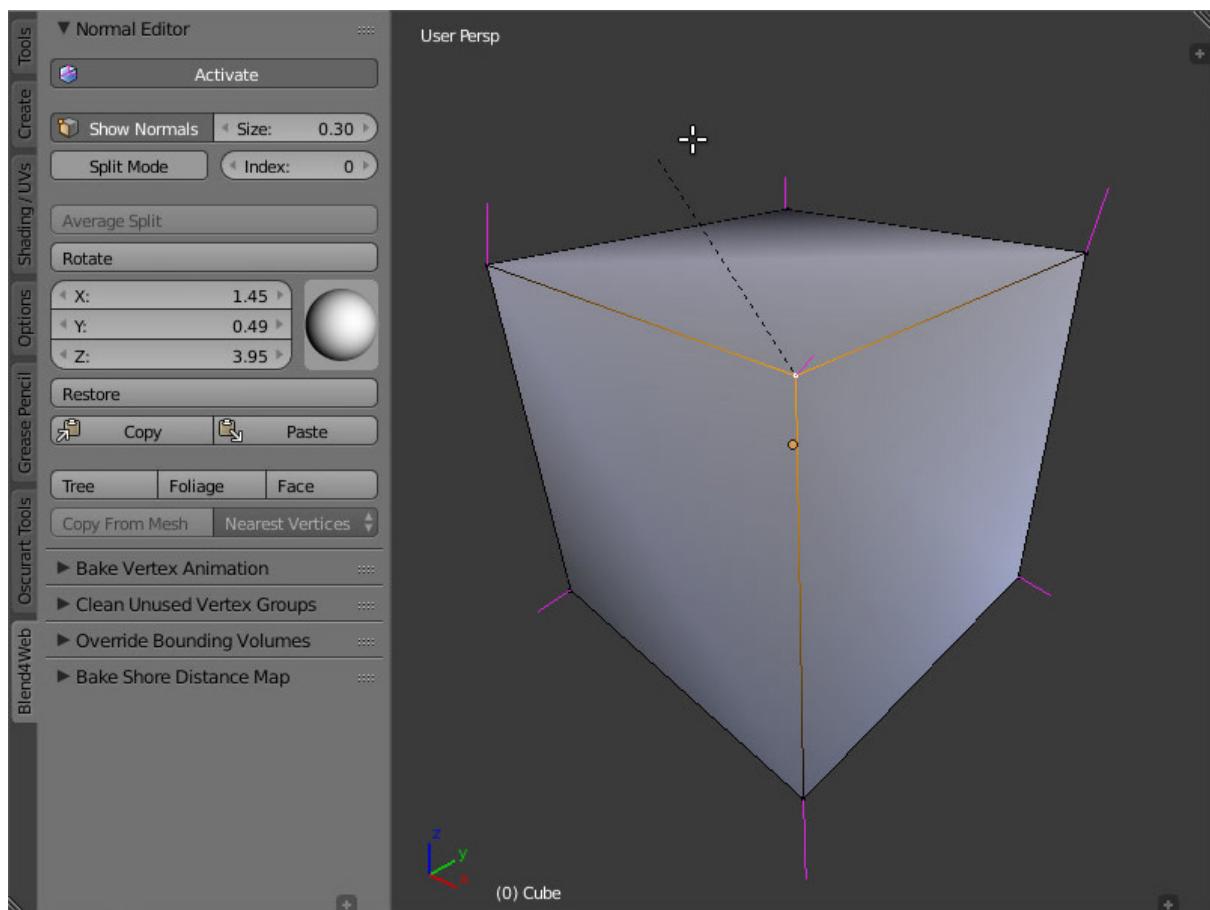
При помощи этого инструментария можно вращать нормаль каждого вертекса индивидуально.

Кнопка **Rotate** (горячие клавиши **Shift+Ctrl+R**) позволяет вращать вертексы нормали в привычном для пользователя Blender режиме.

#### *Использование*

Выделив один или несколько вертексов, которые хотите изменить, вращайте их, используя визуальную сферу на панели редактора вертексовых нормалей Blend4Web, или задайте направление каждой координате в цифровом значении.

Более удобный вариант манипуляции с вертексными нормалами предоставляет кнопка **Rotate**. Вращение происходит по координатам в пространстве экрана, но, как и при обычном вращении объектов в Blender, можно изолировать нужную ось координат и вращать только по ней (нажав X, Y или Z).



### 14.2.6 Split Normals

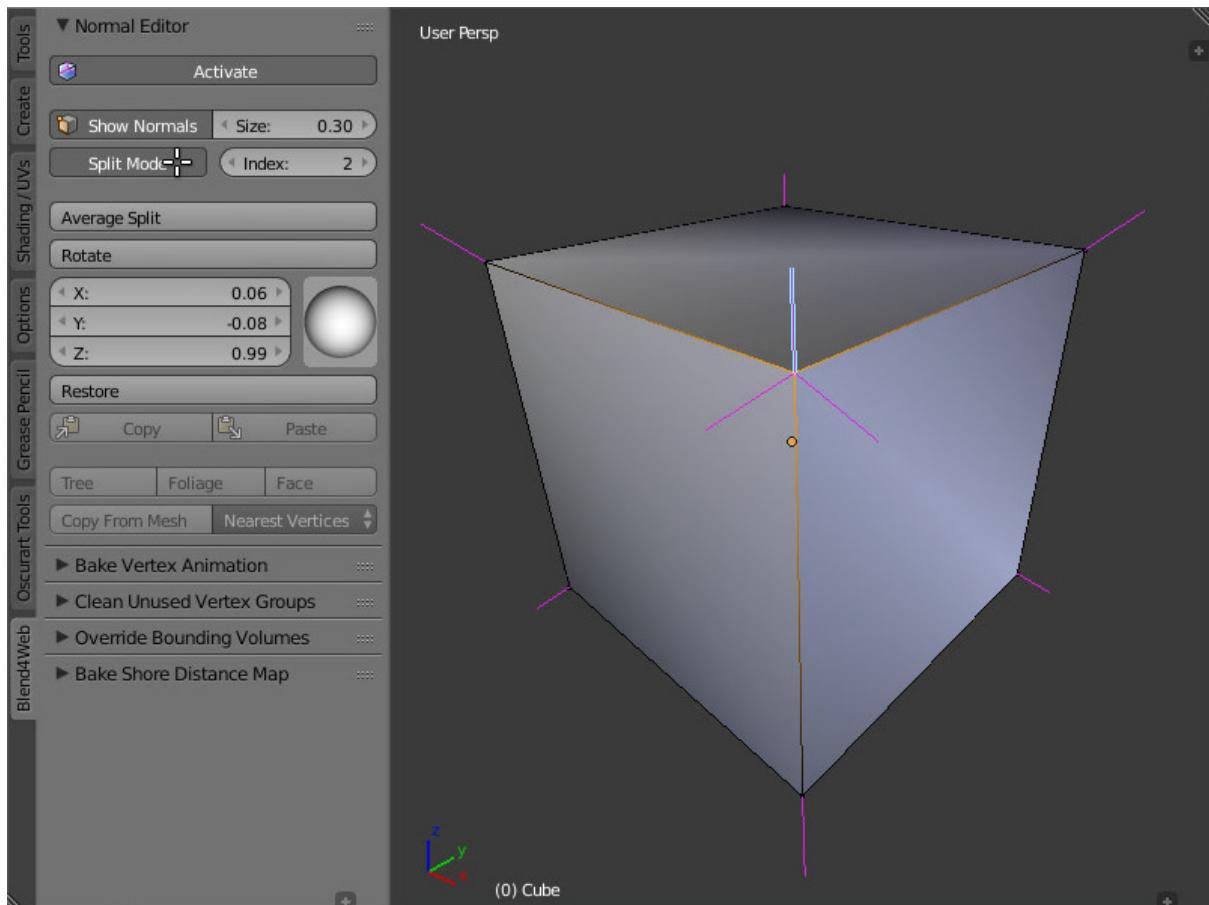
#### *Описание*

Режим **Split Normals** позволяет редактировать вертексы нормали индивидуально для каждой грани (**face** в Blender), образующей редактируемую вершину (**vertex** в Blender). Переключатель вертексных индексов позволяет перемещаться между нормалами разделенного вертекса.

#### *Использование*

Перейдя в режим **Split Normals**, выберите нужный вертекс и измените направление

его нормали. Прежде всего подвергнется модификации та нормаль, которая имеет наименьший индекс в очереди (то есть 0) редактируемого вертекса. Далее, перемещаясь по индексам, перейдите на следующую нормаль этого вертекса и редактируйте ее.



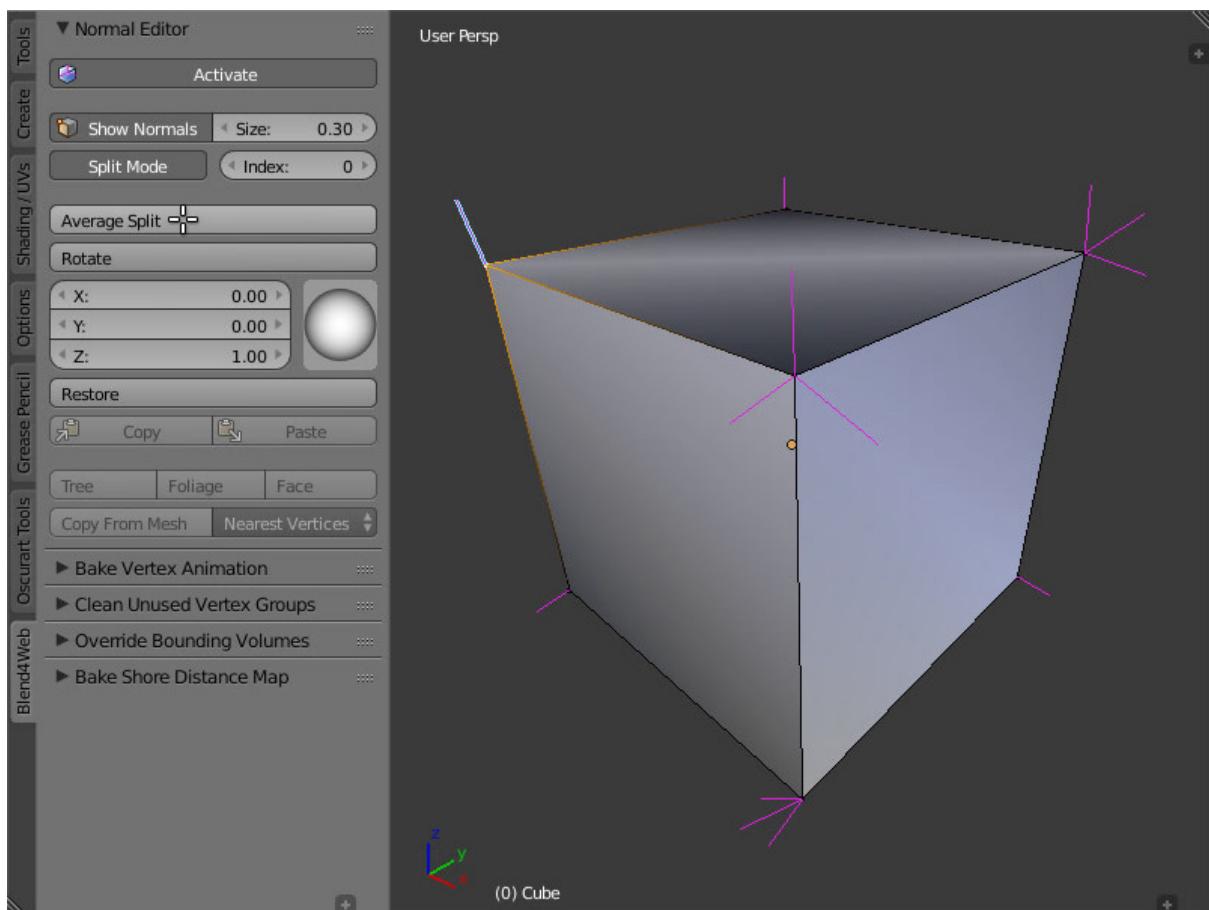
### 14.2.7 Average Split

#### *Описание*

**Average Split** усредняет направление вертексной нормали, исходя из направлений нормалей, разделенных функцией **Split Normals**.

#### *Использование*

Для того, чтобы объединить несколько разделенных вертексных нормалей в одну так, чтобы она имела их среднее направление, достаточно выделить нужный вертекс и нажать **Average Split**.



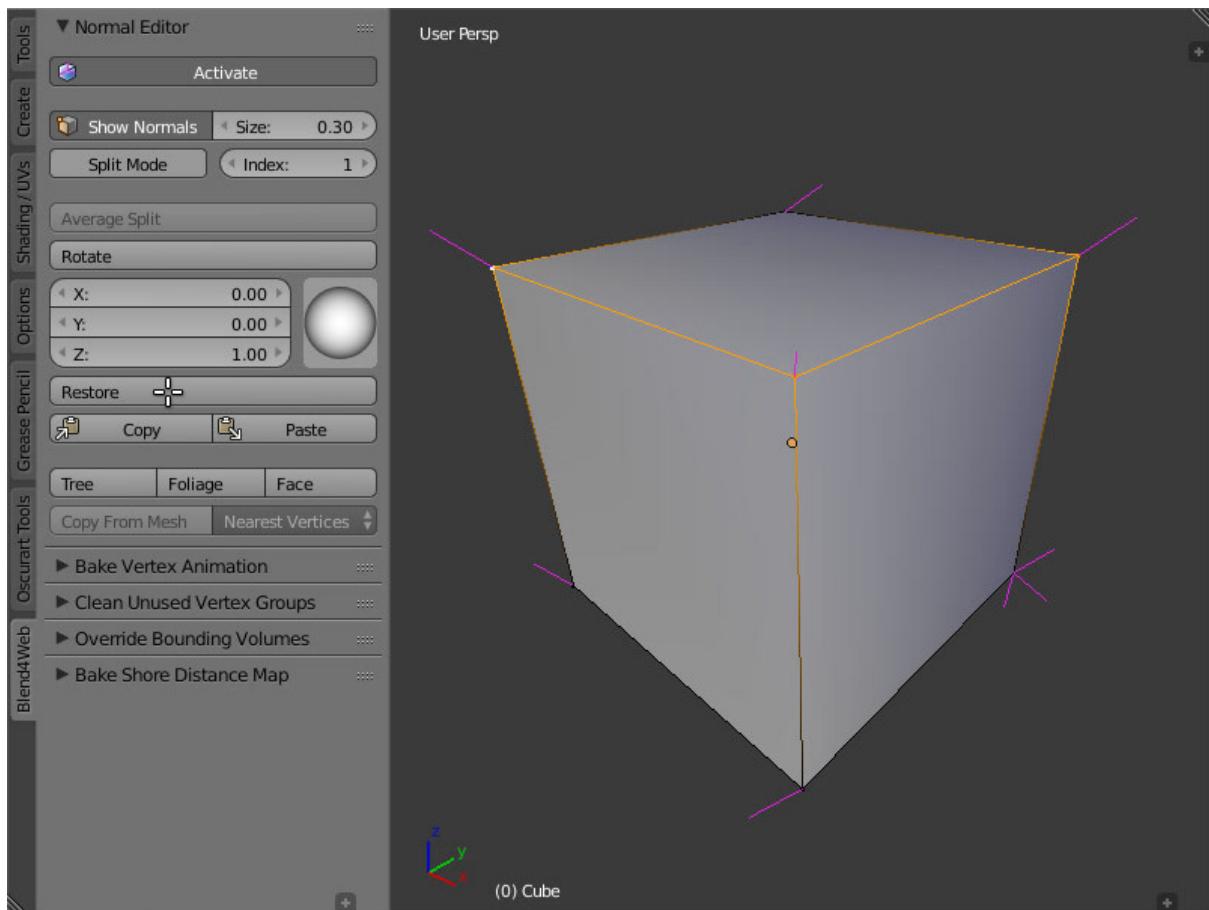
### 14.2.8 Restore

#### *Описание*

Кнопка **Restore** восстанавливает исходное направление нормалей выделенных вертексов.

#### *Использование*

Для того, чтобы восстановить нормали к исходному направлению (которое рассчитывается на основе нормалей полигонов), нужно выделить нужные вертексы и нажать кнопку **Restore**.



### 14.2.9 Tree, Foliage и Face

#### *Описание*

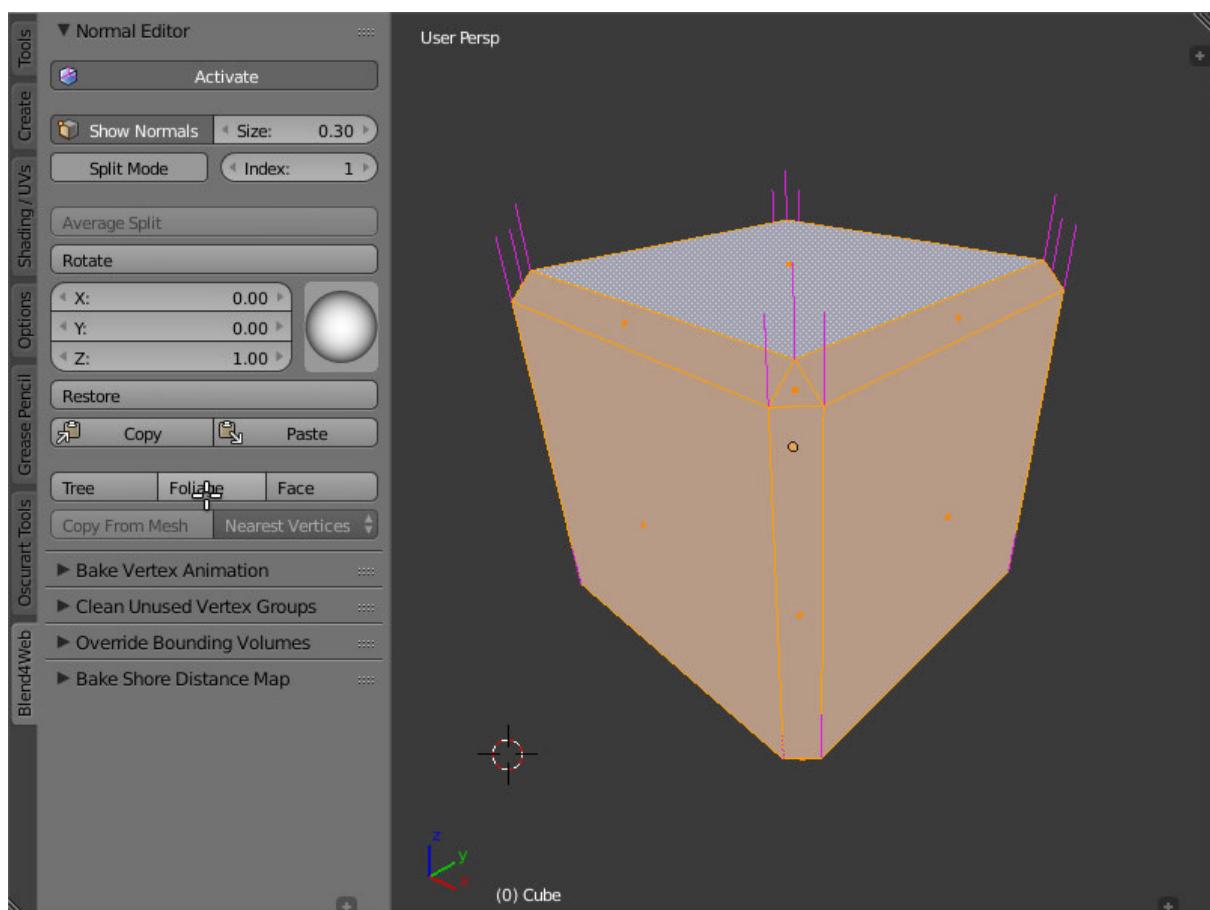
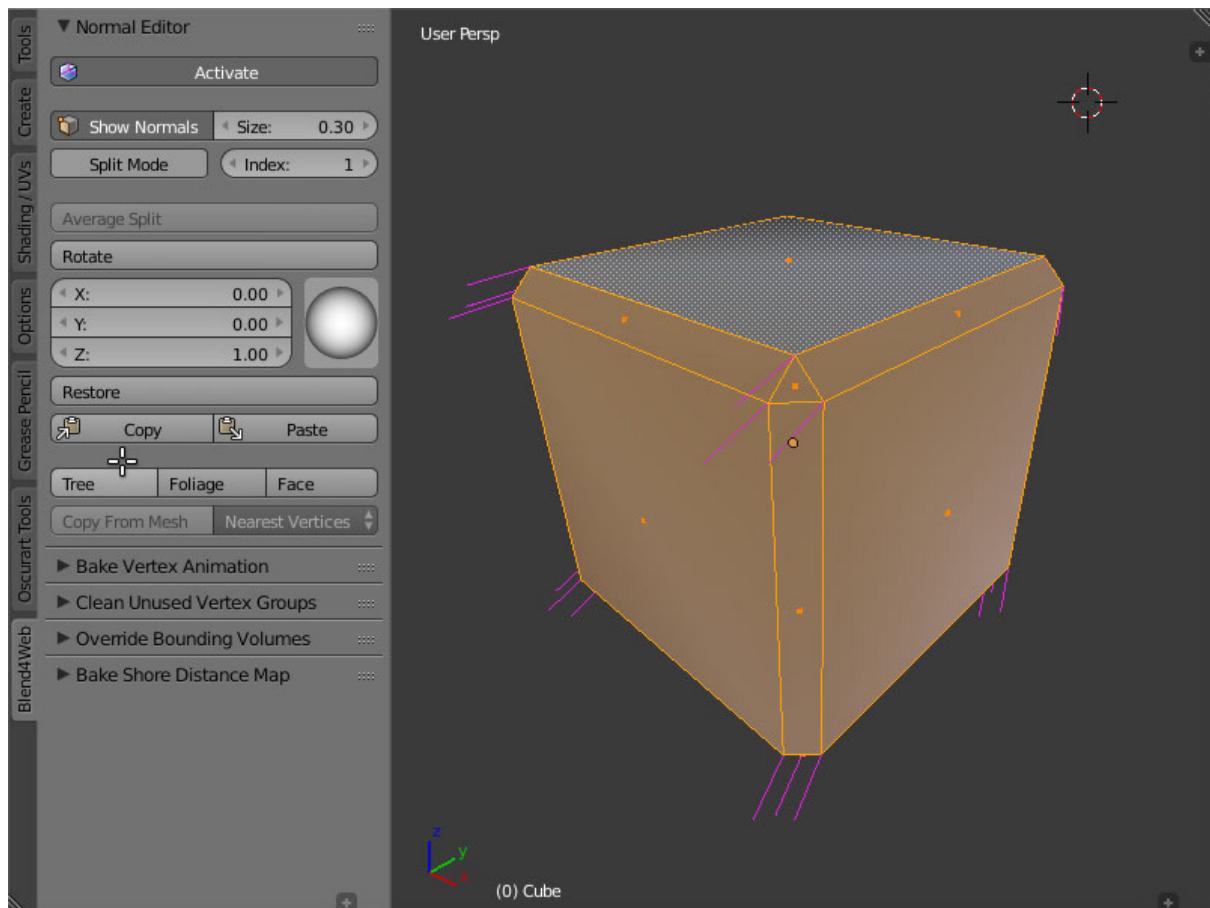
Функция **Tree** направляет нормали выделенных вертексов в сторону от 3D-курсора. **Foliage** направляет нормали вертексов строго вверх по глобальной оси Z. **Face** направляет нормали вертексов выделенного полигона параллельно нормали этого полигона.

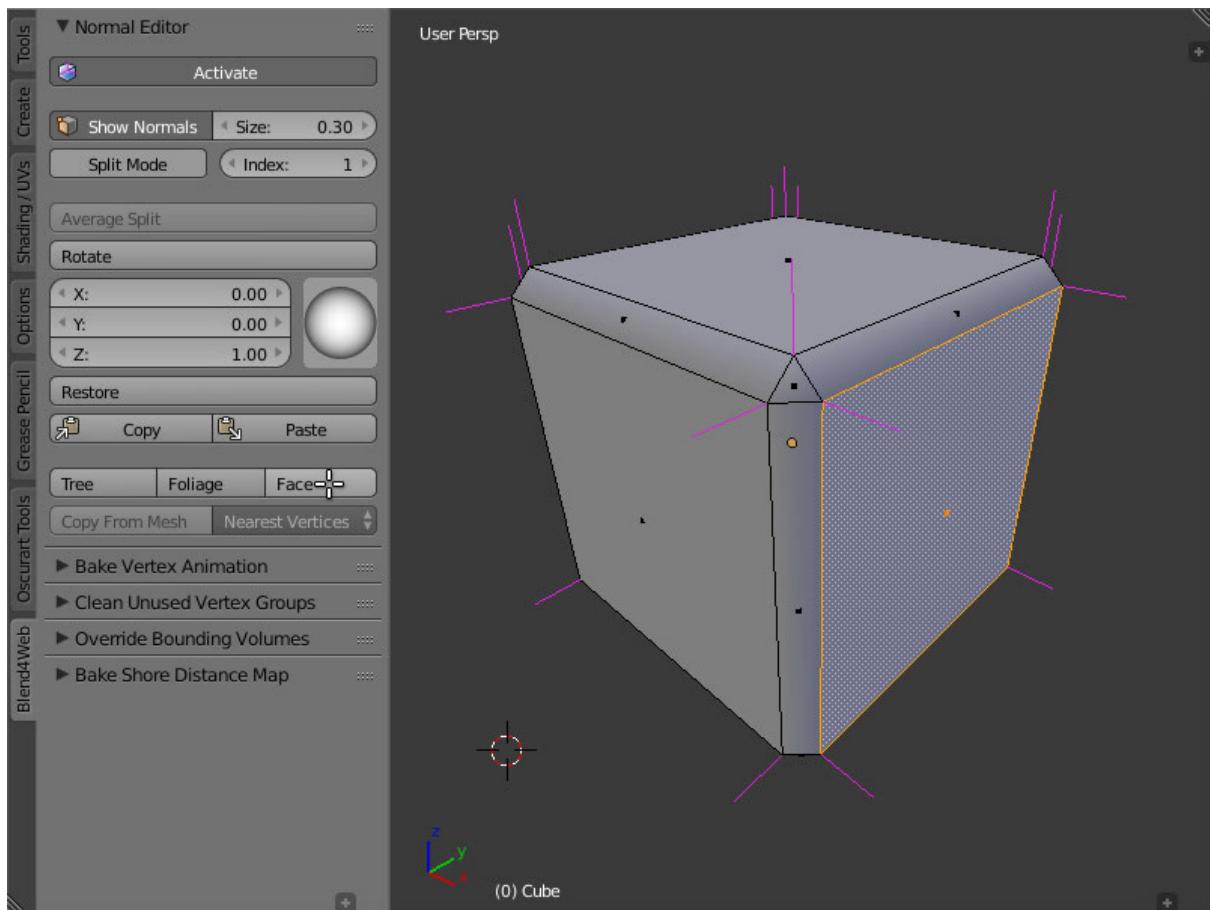
#### Использование

Чтобы воспользоваться функцией **Tree**, выделите нужные вертексы и расположите курсор в нужном вам месте. После нажатия кнопки **Tree** все выделенные вертексы будут направлены в сторону от курсора, как будто бы они выстреливают из одной точки.

Функция **Foliage** очень проста в использовании: выделите нужные вертексы и нажмите кнопку, и функция направит их строго вверх.

Чтобы направить нормали вертексов параллельно нормали полигона, выделите нужный полигон и нажмите **Face**. Все нормали вертексов, образующих этот полигон, будут направлены параллельно нормали самого полигона. Эта функция работает только с одним выделенным полигоном одновременно.





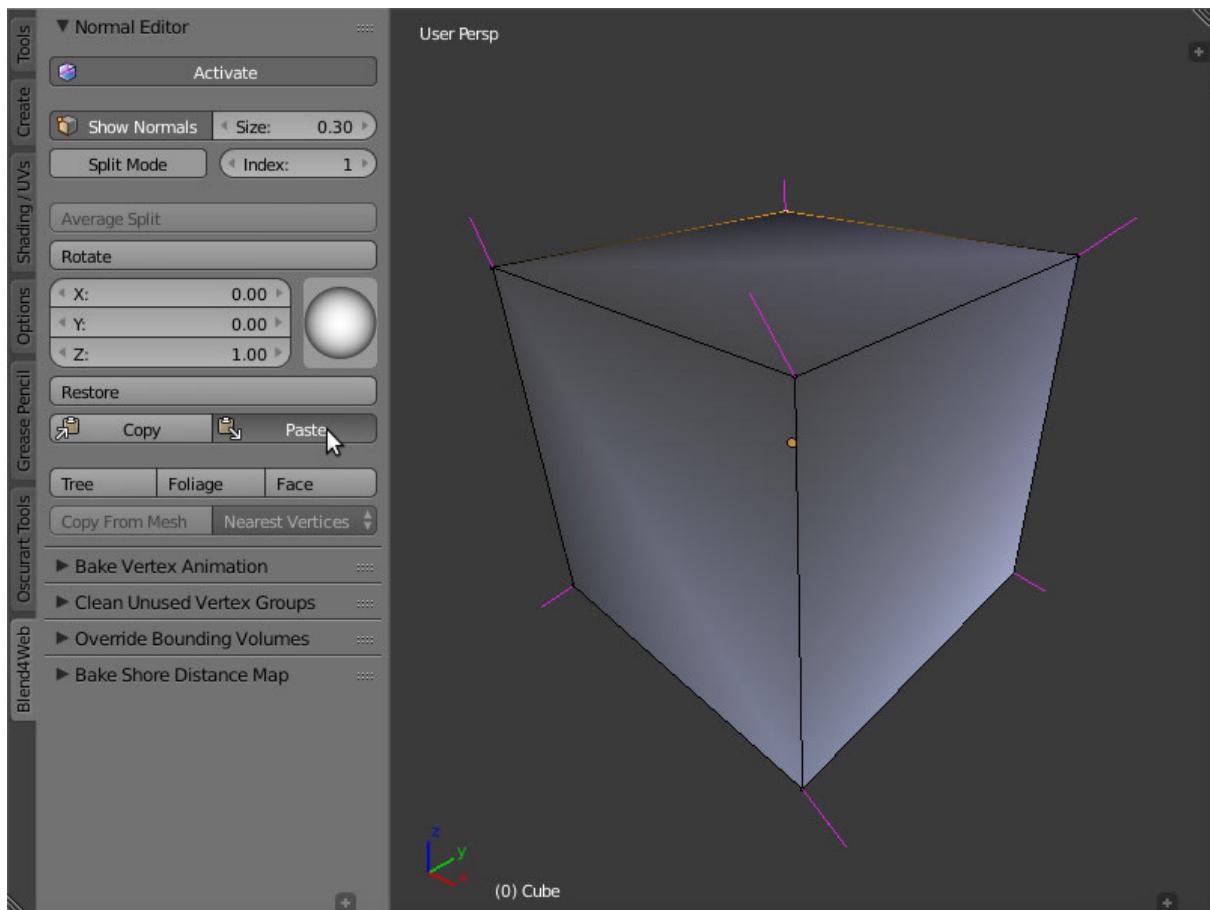
### 14.2.10 Copy/Paste

#### *Описание*

Копирует направление нормали с одного вертекса на другой.

#### *Использование*

Выделите вертекс для копирования и нажмите **Copy**. Затем выделите вертексы, на которых хотите копировать результат, и нажмите **Paste**. Вставлять скопированную информацию можно на множество разных вертексов. Кнопки не активны в режиме **Split Mode** и не переносят данные уже “разделенных” нормалей на другие вертексы.



### 14.2.11 Copy From Mesh

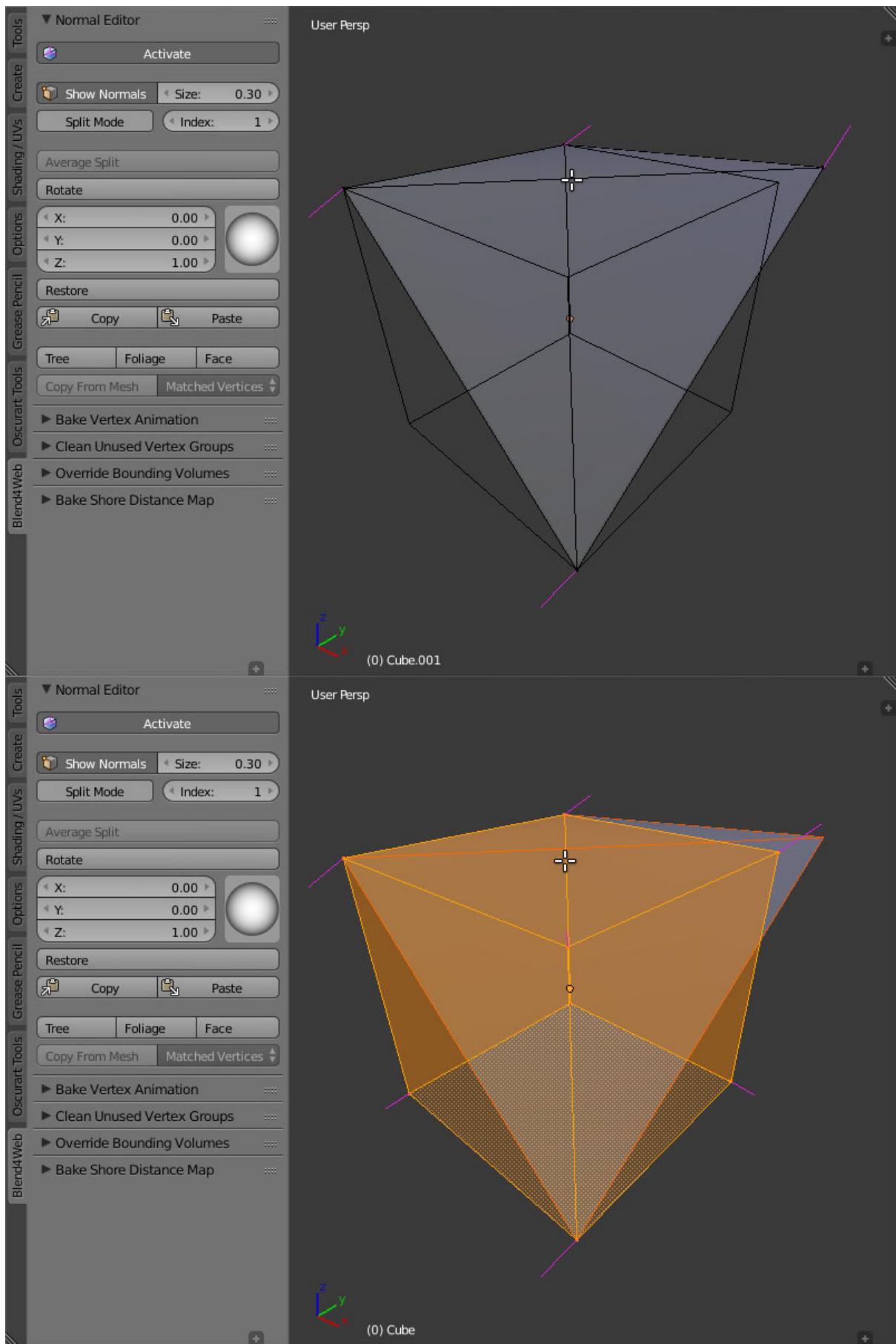
#### *Описание*

Эта функция позволяет копировать направление вертексных нормалей с одной геометрии на другую. Существует два режима копирования, **Matched Vertices** и **Nearest Vertices**.

В режиме **Matched Vertices** копируются направления нормалей с вертекса одной модели на вертекс другой, если у них одинаковые координаты; прочие вертексы игнорируются. В режиме **Nearest Vertices** копируются нормали с ближайших вертексов другой модели.

#### *Использование*

Работа с этой функцией производится в объектном режиме. Первым нужно выделить объект, с которого будет происходить копирование, далее объект, на который будет происходить копирование. Необходимо также выделить вертексы целевого объекта, на которые необходимо произвести перенос направления нормалей.



## 14.3 Освещение от источников света

На сцене может быть несколько (но не менее одного) источников света разного типа.

### 14.3.1 Типы источников света

Поддерживаются источники света следующих типов:

**Point** Точечный. Свет распространяется из одной точки равномерно во все стороны, с постепенным затуханием.

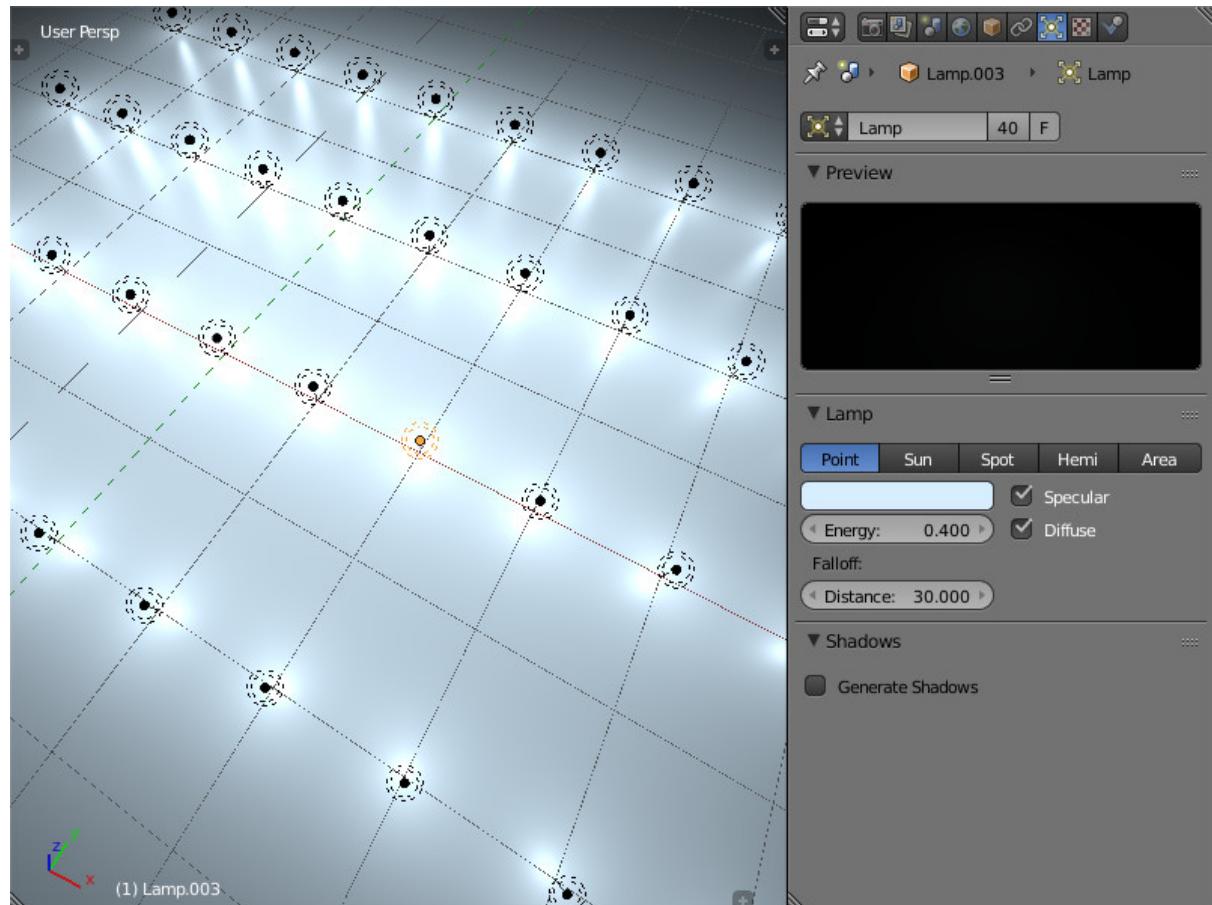
**Sun** “Солнце”. Свет распространяется из бесконечной плоскости прямолинейно в одном направлении, без затухания.

**Spot** Прожектор. Свет распространяется из одной точки, с ограничением угла распространения, с постепенным затуханием.

**Hemi** Полусфера. Свет распространяется из бесконечной полусферы, без затухания.

### 14.3.2 Настройка источников света

Производится во вкладке **Object Data** при выборе объекта-лампы.



**Color** Цветовая характеристика света. Значение по умолчанию (1.0, 1.0, 1.0) (белый).

**Energy** Интенсивность излучения. Значение по умолчанию 1.0.

**Falloff** Тип затухания. Значение экспортируется, но в движке всегда используется **Inverse Square** (обратный квадратичный). Применяется для источников света типа **Point** и **Spot**. Значение по умолчанию **Inverse Square**.

**Distance** Параметр затухания. Применяется для источников света типа **Point** и **Spot**. Значение по умолчанию 25.0.

**Specular** Создание отблеска на объектах. По умолчанию включено.

**Diffuse** Применение рассеянного освещения к объектам. По умолчанию включено.

**Spot Shape > Size** Угол конуса в градусах. Применяется для источников света типа **Spot**. Значение по умолчанию 45°.

**Spot Shape > Blend** Параметр смягчения края светового пятна. Применяется для источников света типа **Spot**. Значение по умолчанию 0.15.

**Dynamic intesity** Источник света используется для расчета изменения времени суток. Применяется для источников света типа “Солнце”. По умолчанию отключено.

**Shadows > Generate shadows** Источник света используется для расчета падающих теней. Применяется в случае наличия нескольких источников света. По умолчанию отключено.

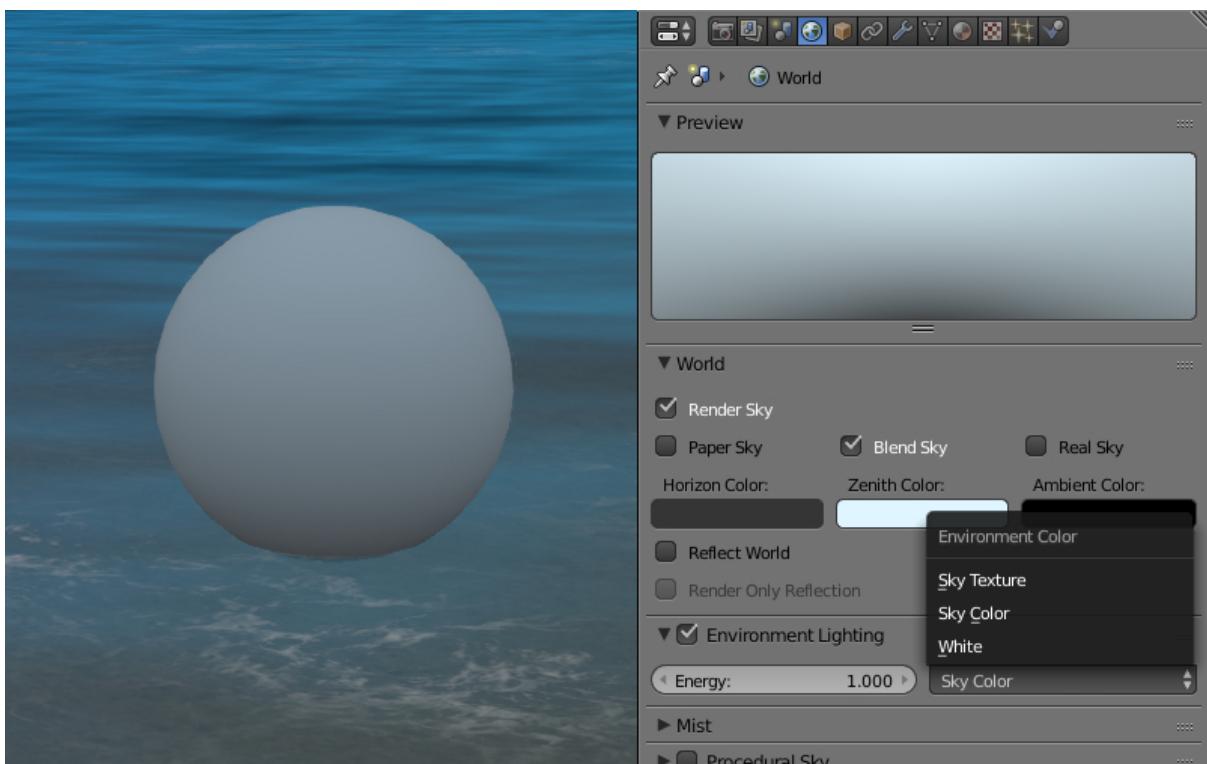
## 14.4 Освещение от окружающей среды

Движок поддерживает 3 метода симуляции рассеянного освещения от окружающих объектов.

1. “Плоское” освещение белым цветом.
2. Полусферическая модель освещения, в которой задается цвет горизонта и цвет зенита, в результате чего объекты заливаются градиентом между этими цветами в зависимости от направления нормалей.
3. Освещение с помощью *карты окружения* - т.н. методика image-based lighting.

### 14.4.1 Активация

Включить опцию **Environment Lighting** во вкладке **World**.



#### 14.4.2 Настройка

**World > Environment Lighting > Energy** Интенсивность освещения от окружающей среды. Значение по умолчанию 1.0.

**World > Environment Lighting > Environment Color** Выбор метода симуляции рассеянного освещения: **White** - “плоское” освещение белым цветом, **Sky Color** - полусферическая модель, **Sky Texture** - освещение с помощью *карты окружения*. Значение по умолчанию **White**.

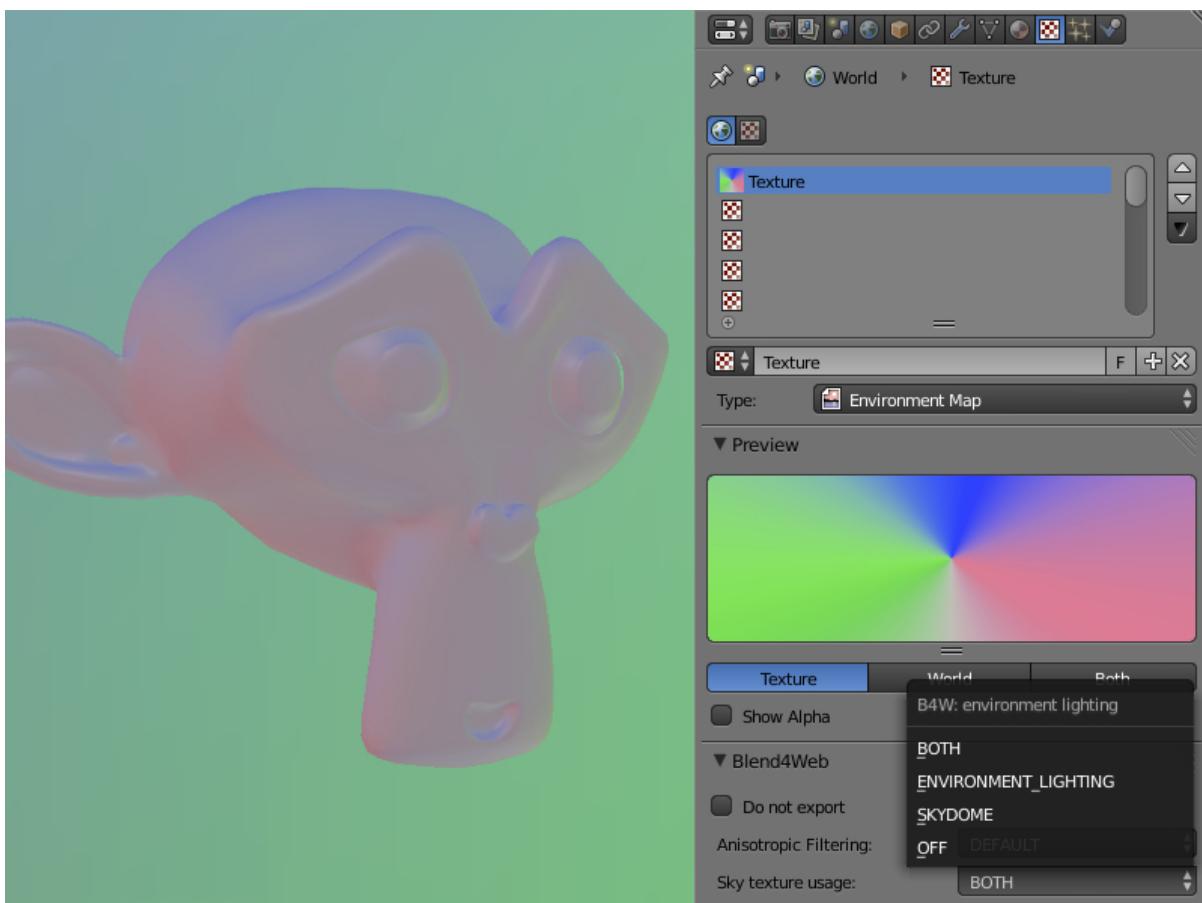
**World > Horizon Color и World > Zenith Color** Если выбрана полусферическая модель рассеянного освещения **Sky Color**, цвет горизонта и цвет зенита задаются цветоподборщиками **World > Horizon Color** (цвет горизонта) и **World > Zenith Color** (цвет зенита). При выборе цвета рекомендуется активировать опцию **World > Blend Sky**.

#### 14.4.3 Использование карты окружения

Для того, чтобы использовать карту окружения в целях реализации освещения от окружающей среды, необходимо:

1. Включить опцию **Environment Lighting** во вкладке **World**.
2. Выбрать метод **Environment Lighting > Sky Texture**.
3. Перейти из вкладки **World** во вкладку **Texture**.
4. Создать *карту окружения*, загрузить в нее соответствующее изображение.

5. Для карты окружения на панели Export Options для значения Sky Texture Usage выбрать ENVIRONMENT\_LIGHTING или BOTH (опция BOTH активирует также использование этой текстуры в качестве *текстуры неба*).



## 14.5 Тени

Тени - один из важнейших элементов при рендеринге конечного изображения. Они предоставляют зрителю не только информацию о силуэте объектов, но и об их высоте и взаимном расположении, положении источника света и т.д.

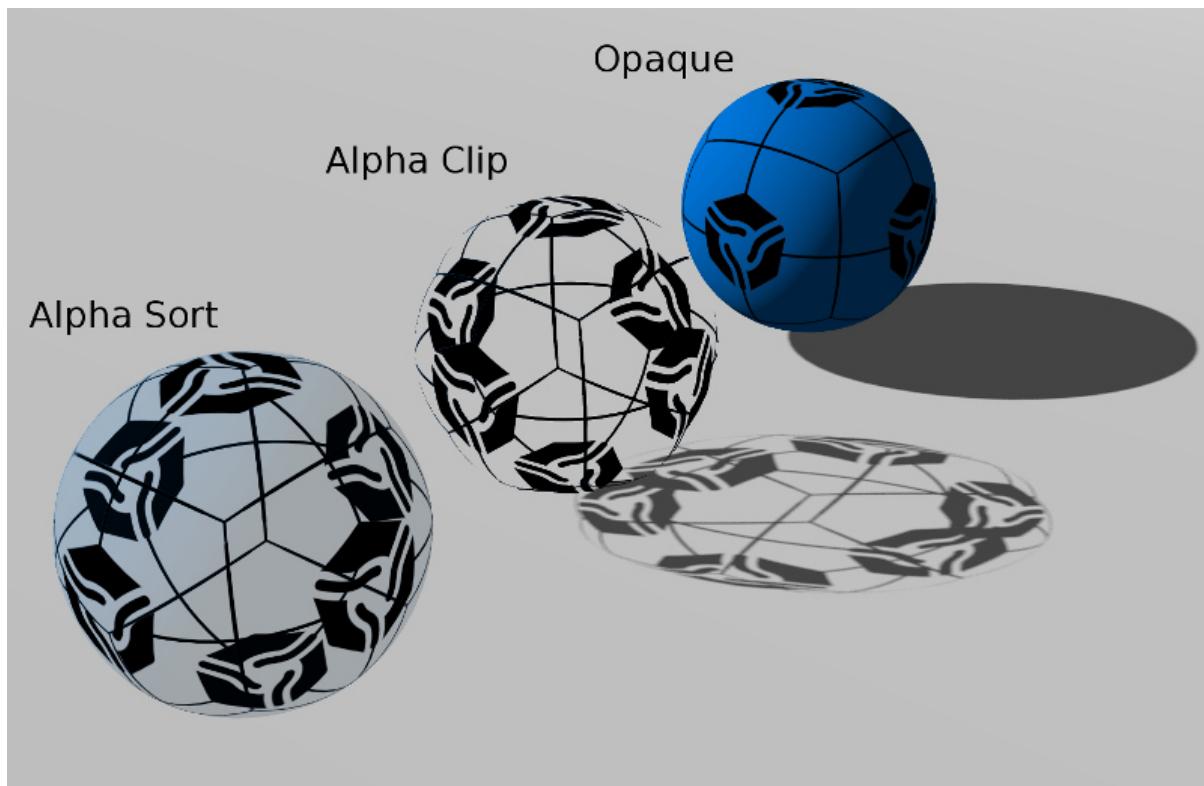
Платформой Blend4Web реализуются такие техники, как каскадные карты теней (CSM) и смягченные тени (PCF).

### 14.5.1 Активация

- На объектах, **отбрасывающих** тени, включить опцию Shadows: Cast во вкладке Object.

2. На объектах, **получающих** тени, включить опцию Shadows: Receive во вкладке Object.
3. Убедиться, что включена опция Shadows AUTO или ON во вкладке Render.

**Примечание:** Объекты, имеющие *прозрачные материалы с градиентом*, не отбрасывают теней.

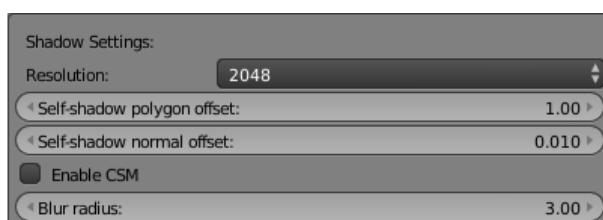


### 14.5.2 Настройка

**Направление** В случае наличия нескольких источников света рекомендуется указать, какой именно источник света будет использоваться для расчета падающих теней, включив опцию Shadows > Generate Shadows во вкладке Object Data при выборе объекта-лампы.

**Цвет** Цвет тени определяется настройками *освещения от окружающей среды*.

Во вкладке Render на панели Shadows находятся дополнительные настройки:



**Resolution** Разрешение используемой карты теней. Значение по умолчанию: 2048x2048px.

**Self-shadow polygon offset** Коэффициент смещения полигона в зависимости от ориентации к источнику света. Значение по умолчанию: 1.

**Self-shadow normal offset** Коэффициент смещения полигона по нормали. Значение по умолчанию: 0.01.

Последние две настройки служат для борьбы с артефактами самозатенения. Они проявляются на объектах, одновременно отбрасывающих и принимающих тени. Параметр **Self-shadow polygon offset** лучше справляется с артефактами во внутренних областях полигонов, а **Self-shadow normal offset** - в приграничных. Оба параметра приводят к искажению теней, поэтому рекомендуется держать их как можно меньшими.

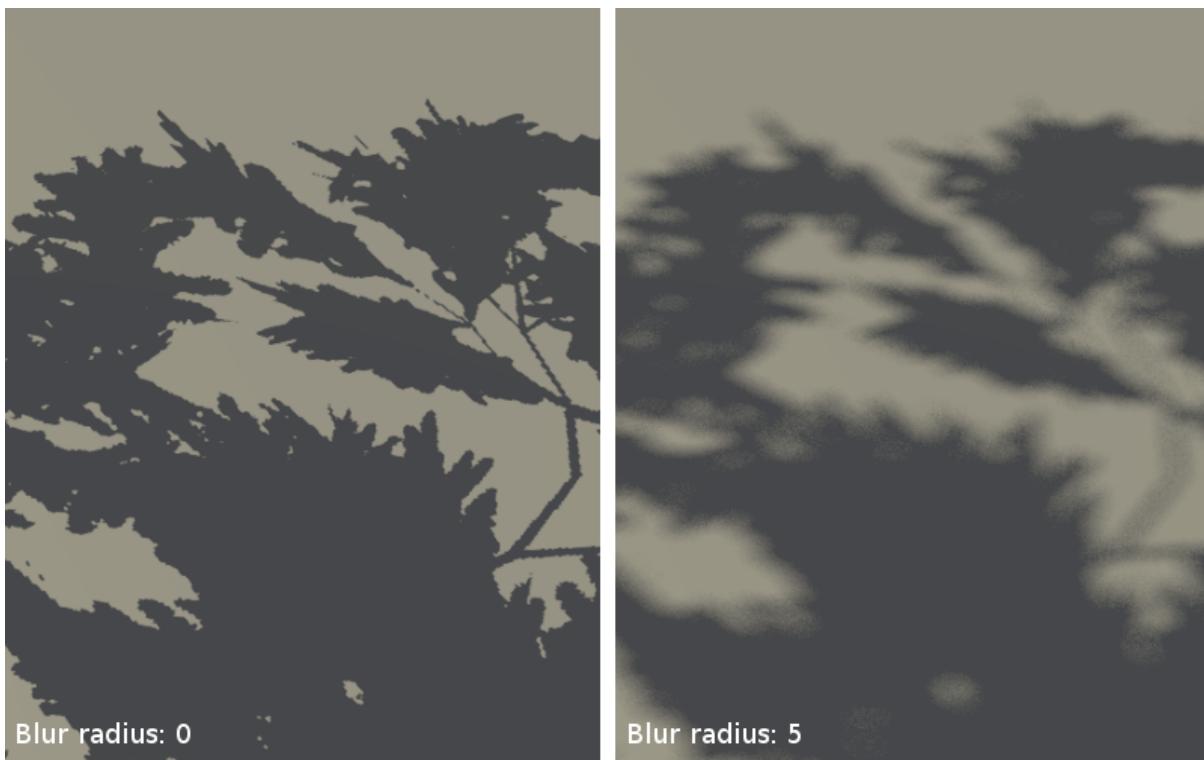


**Enable CSM** Включение каскадной модели теней; открывает доступ к расширенным настройкам. Отключено по умолчанию.

Эта опция позволяет выбрать один из следующих вариантов наложения теней:

- Стандартная модель, использующая одну оптимизированную карту теней, охватывающую всю сцену (**Enable CSM** отключена).
- Каскады теней (**Enable CSM** включена).

**Blur radius** Коэффициент размытия теней, позволяющий настроить смягченные тени. Значение по умолчанию: 3. Коэффициент 0 даст жесткие тени.



Смягченные тени могут повысить качество и реалистичность изображения. Они скрывают неизбежную при использовании основанных на изображениях техник зубчатость краев, особенно сильно проявляющуюся для карт теней низкого разрешения. Использование смягченных теней часто позволяет снизить разрешение без существенной потери качества.

## Стандартная модель

Этот вариант больше подходит для маленьких сцен, состоящих из небольшого числа объектов. Благодаря оптимизации на таких сценах можно добиться более высокого качества теней по сравнению с каскадной моделью. Данный вариант проще и быстрее настроить, а использование всего лишь одной карты теней положительно сказывается на производительности.

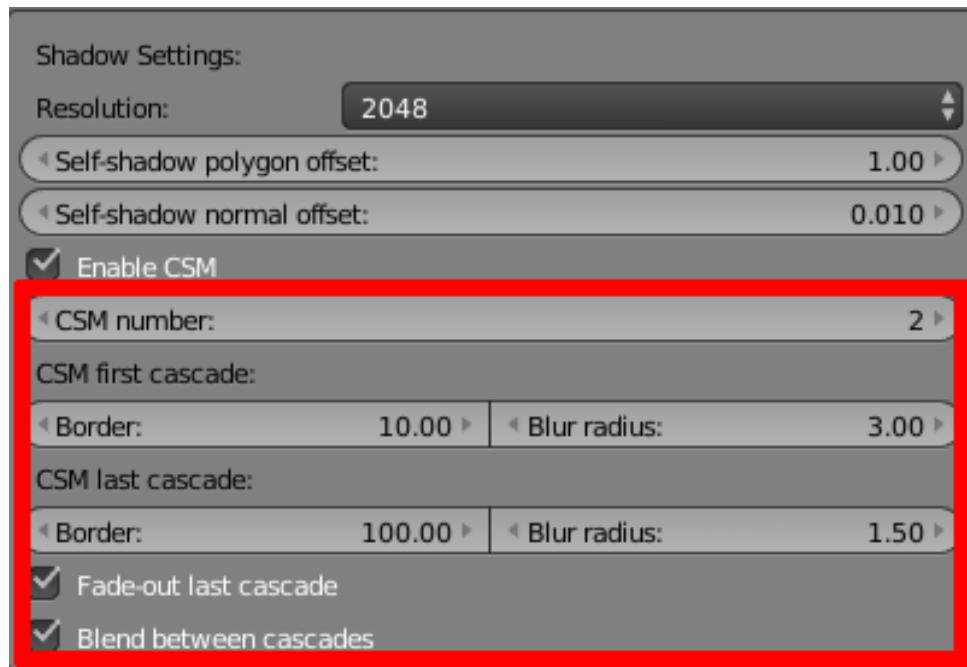
## Каскады теней

**Примечание:** Данные настройки поддерживаются только для источников освещения типа Sun. Для других источников каскады автоматически отключаются.

Для обеспечения приемлемого качества теней и одновременно покрытия значительных пространств необходимо использовать несколько стадий генерации теней (каска-

дов). При этом вблизи наблюдателя располагается каскад с наилучшим качеством, вдали от наблюдателя — с наихудшим. Этот вариант больше подходит для сцен среднего и большого размера, например, игровых уровней.

При включении предоставляет расширенные настройки:



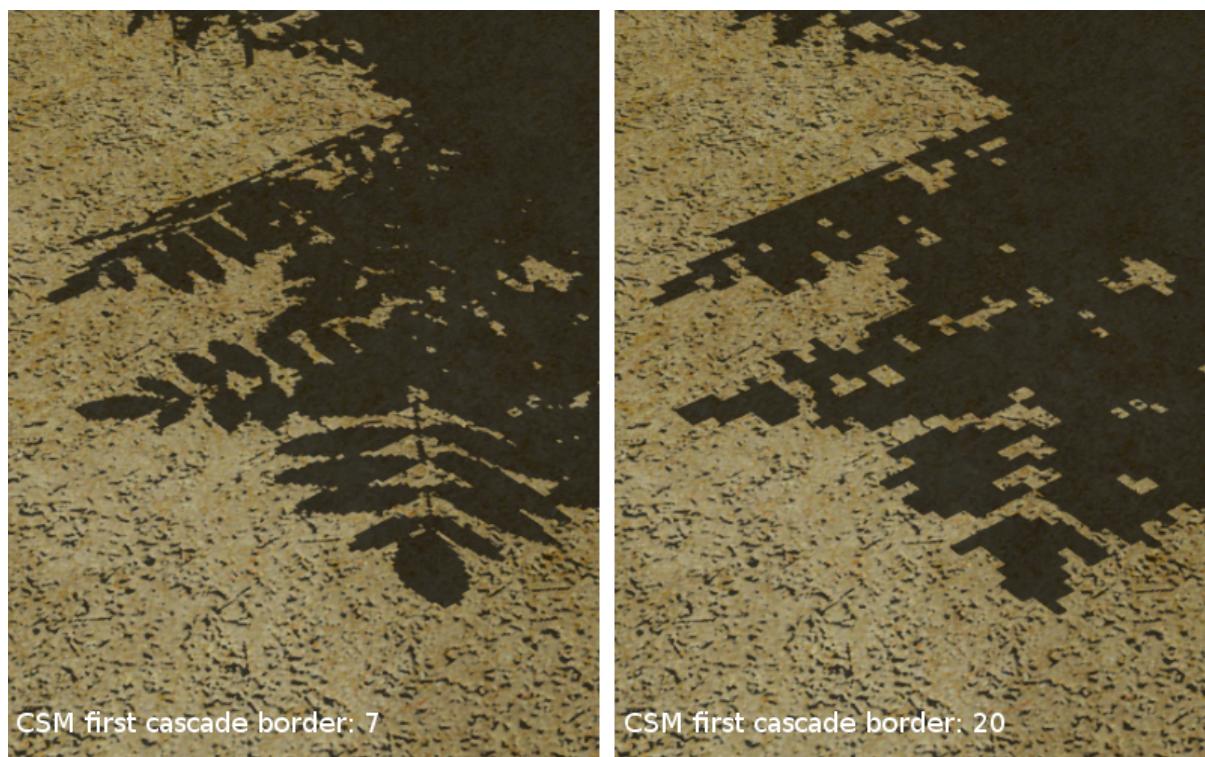
**CSM number** Количество каскадов теней. Поддерживается от 1 до 4 каскадов. Значение по умолчанию: 1.

**CSM first cascade border** Размер первого каскада. Значение по умолчанию: 10.0.

**CSM last cascade border** Размер последнего каскада. Значение по умолчанию: 100.0.

Размеры промежуточных каскадов интерполируются на основе последних двух параметров.

**Примечание:** При настройке следует помнить, что, увеличивая размер каскада, мы получаем на нем менее качественные тени. С другой стороны, уменьшение параметра **CSM first cascade border** приблизит к камере и сделает более заметными последующие менее детальные каскады. Уменьшение параметра **CSM last cascade border** приведет к исчезновению теней на более близком расстоянии от камеры. Однако, при использовании мягких теней качество в целом улучшится благодаря размытию на границах.



**CSM first cascade blur radius** Коэффициент размытия на первом каскаде. Значение по умолчанию: 3. Коэффициент 0 даст жесткие тени.

**CSM last cascade blur radius** Коэффициент размытия на последнем каскаде. Значение по умолчанию: 1.5. Коэффициент 0 даст жесткие тени.

Радиус размытия каждого промежуточного каскада интерполируется на основе этих параметров.

**Примечание:** Смягченные тени рекомендуется настраивать сначала на первом каскаде опцией **CSM first cascade blur radius**, а далее на всех остальных с помощью **CSM last cascade blur radius**. Часто на последнем каскаде может потребоваться размытие меньшее, нежели на первом. Это нужно для того, чтобы тени на последнем каскаде не стали слишком блеклыми из-за низкой детализации, к тому же это уменьшит нежелательные артефакты самозатенения.

**Fade-out last cascade** Плавное исчезновение последнего каскада. По умолчанию включено.

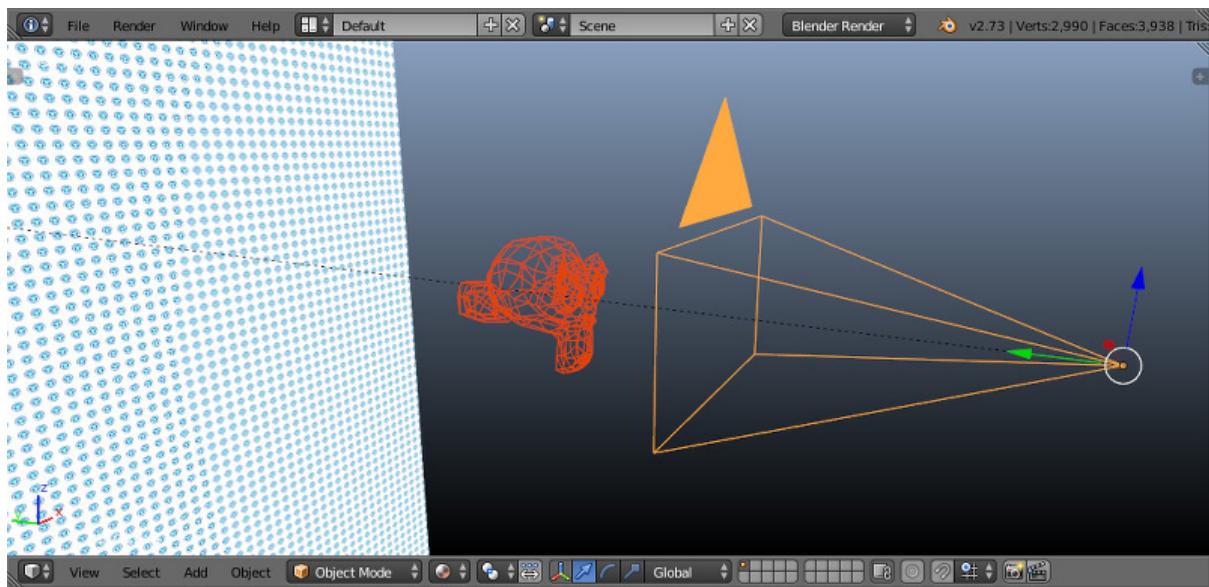
**Blend between cascades** Сглаживание границ между каскадами. По умолчанию включено.



## 14.6 Цвет фона

Цвет фона можно задать несколькими способами:

1. Установить параметры `Horizon Color` и `Zenith Color` на вкладке `World` в Blender, предварительно выставив опцию `World > Render Sky`.
2. Поместить сцену внутрь модели (например, куба или сферы) с направленными внутрь нормалями, с материалом и optionalной текстурой.
3. Разместить перед камерой поверхность с материалом и optionalной текстурой, присоединить ее к камере связью родитель-потомок. При необходимости настроить расстояние до поверхности, переднюю и заднюю плоскости отсечения камеры.



4. Использовать *текстуру неба*.
5. Настроить динамически генерируемую *атмосферу*.
6. Установить параметр движка `background_color`, используя программный метод `config.set()`, предварительно отключив опцию `World > Render Sky` на вкладке `World` в Blender. Установленное значение используется в качестве аргумента метода WebGL `clearColor()`. Для получения корректных результатов рекомендуется отключить прозрачность контекста WebGL (параметр `alpha`). Такая конфигурация используется по умолчанию в стандартном веб-плеере движка.

```
var m_cfg = b4w.require("config");
var m_main = b4w.require("main");

// gray
m_cfg.set("background_color", new Float32Array([0.224, 0.224, 0.224, 1.0]));
m_cfg.set("alpha", false);

m_main.init(...);
```

7. В качестве фона можно использовать любой HTML контент, находящийся позади элемента `canvas`, который используется для рендеринга. Для этого необходимо активировать прозрачность контекста WebGL (параметр `alpha`), предварительно отключив опцию `World > Render Sky` на вкладке `World` в Blender. Для получения корректных результатов рекомендуется выставить полностью прозрачный черный цвет фона. Такая конфигурация используется по умолчанию в стандартном *просмотрщике сцен* SDK движка.

```
var m_cfg = b4w.require("config");
var m_main = b4w.require("main");

m_cfg.set("background_color", new Float32Array([0.0, 0.0, 0.0, 0.0]));
m_cfg.set("alpha", true);
```

```
m_main.init(...);
```

См.также:

*Альтера-композитинг*

### Постпроцессинговые эффекты

---

#### 15.1 Размытие при движении

Эффект размытия при движении (motion blur) служит целям увеличения реализма интерактивной сцены. Он проявляется при движении камеры или объектов в виде “смазывания” изображения.



##### 15.1.1 Активация

Активировать панель Motion Blur во вкладке Render.

### 15.1.2 Дополнительные настройки

**Factor** Степень проявления эффекта. Чем выше значение, тем сильнее эффект размытия. Значение по умолчанию 0.01.

**Decay threshold** Степень плавности размытия. Чем выше значение, тем более резким будет эффект. Значение по умолчанию 0.01.

## 15.2 Глубина резкости камеры

Эффект глубины резкости камеры (depth of field, DOF) акцентирует внимание зрителя на части сцены. Проявляется в размытии изображения ближе и дальше от фокуса камеры.



### 15.2.1 Активация

1. Выбрать активную камеру, перейти на панель ее настроек во вкладке **Camera (Object Data)**.
2. Далее возможны два варианта:
  - На панели **Depth of Field** в опции **Focus** выбрать объект, на котором будет сфокусирована камера. В этом случае при удалении или приближении к этому объекту будет происходить соответствующая коррекция фокуса камеры.

- На той же панели установить ненулевое значение **Distance** (в метрах). В этом случае фокус камеры будет располагаться на заданном расстоянии от камеры и перемещаться вместе с ней.

### 15.2.2 Дополнительные настройки

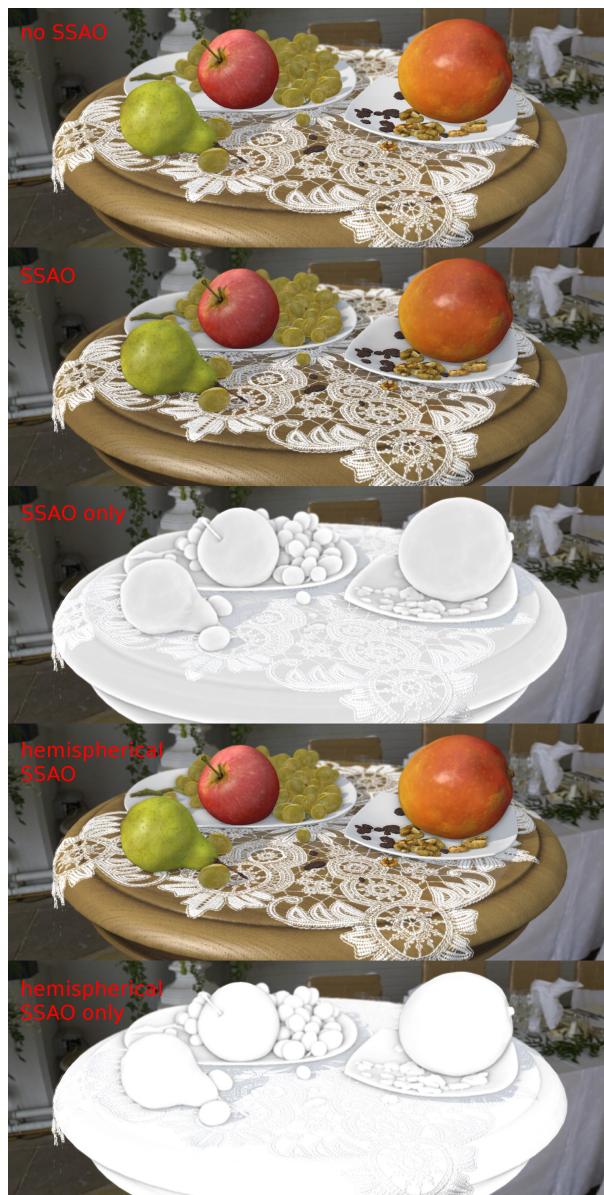
**Front** Расстояние от фокуса до ближней к камере плоскости, за которой происходит полное размытие (в метрах). Значение по умолчанию 1.0.

**Rear** Расстояние от фокуса до дальней от камеры плоскости, за которой происходит полное размытие (в метрах). Значение по умолчанию 1.0.

**Power** Степень размытия. Значение по умолчанию 3.0.

## 15.3 Взаимное затенение

Эффект взаимного затенения (screen-space ambient occlusion, SSAO) применяется с целью воспроизведения сложного переотражения света от объектов. Пространство между близкими объектами менее доступно для рассеянного света и поэтому затеняется сильнее.



### 15.3.1 Активация

Активировать панель Ambient Occlusion (SSAO) во вкладке **Render** и выставить параметр опции **Render Shadows** в положение **AUTO** или **ON** на панели **Render > Shadows**.

### 15.3.2 Дополнительные настройки

**Radius Increase** Фактор умножения радиуса сферического сэмплинга при переходе от внутреннего кольца к внешнему. Значение по умолчанию 3.0.

**Use Hemisphere** Использовать для расчёта затенения полусферический сэмплинг вместо сферического. Помимо этого используется другой закон затенения.

**Use Blur Depth Test** Если активировано - используется размытие SSAO на основе буфера глубины. Иначе - размытие по квадрату 4x4 по соседним пикселям.

**Blur Depth Test Discard Value** Влияние разницы глубины сэмлов на их вес в размытии. Используется при активированном параметре Use Blur Depth Test. Значение по умолчанию 1.0.

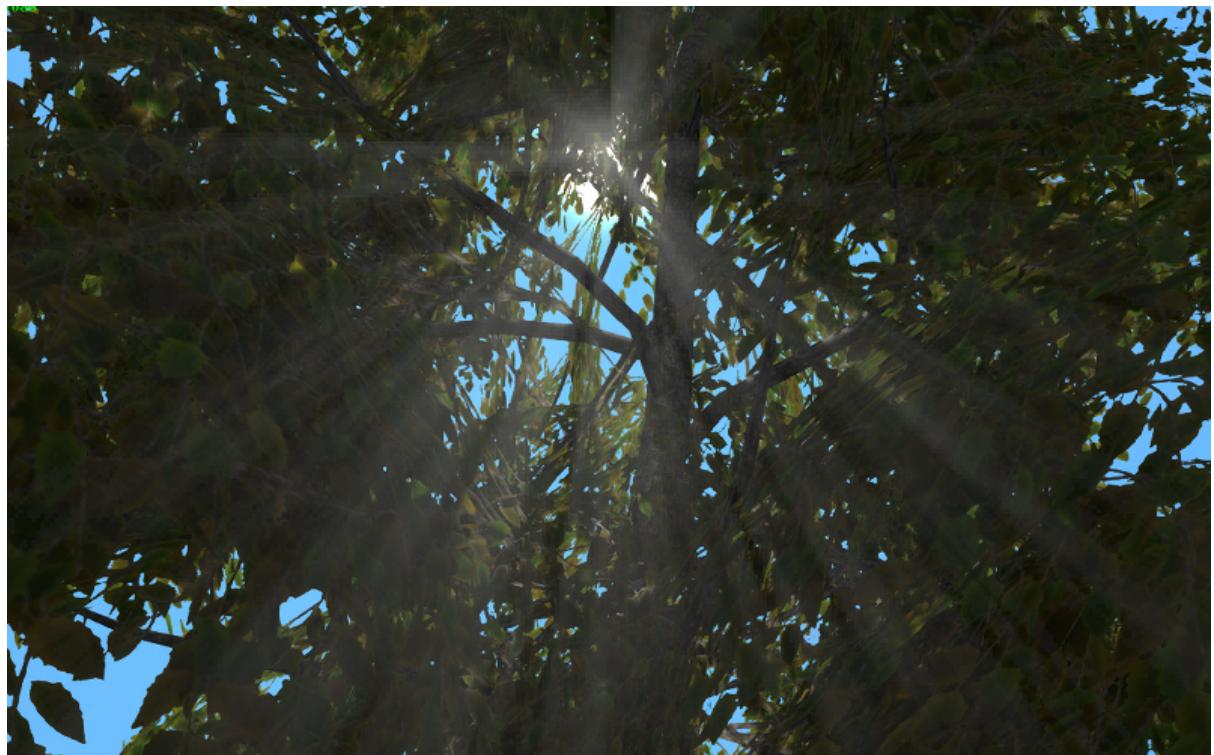
**Influence** Степень проявленности эффекта взаимного затенения. Значение по умолчанию 0.7.

**Distance Factor** Фактор уменьшения проявленности эффекта взаимного затенения с расстоянием. Значение по умолчанию 0.0 (т.е. уменьшения нет).

**Samples** Количество сэмплов (чем больше, тем лучше качество, но меньше производительность). Значение по умолчанию 16.

## 15.4 Сумеречные лучи

Эффект сумеречных лучей (stereoscopic rays, “god rays”) симулирует известное природное явление - свечение освещенных областей воздуха.



### 15.4.1 Активация

Активировать панель God Rays во вкладке Render.

### 15.4.2 Дополнительные настройки

**Intensity** Степень проявленности эффекта. Значение по умолчанию 0.7.

**Maximum Ray Length** Фактор длины лучей. Определяет шаг сэмплов радиального размытия. Значение по умолчанию 1.0.

**Steps Per Pass** Количество шагов на один сэмпл. Значение по умолчанию 10.0.

## 15.5 Эффект засветки ярких деталей

Эффект засветки (Bloom) проявляется при наличии на экране элементов с большой разницей в яркости. Вокруг ярких деталей создается светящийся ореол.



### 15.5.1 Активация

Активировать панель Bloom во вкладке Render.

### 15.5.2 Дополнительные настройки

**Key** Интенсивность эффекта свечения.

**Blur** Степень размытия засветки.

**Edge Luminance** Границное значение относительной яркости элемента, выше которого начинает проявляться эффект засветки.

## 15.6 Подсветка контура (outlining)

В результате применения эффекта подсветки контура вокруг объекта появляется светящийся ореол произвольного цвета.



### 15.6.1 Активация

Эффект подсветки контура активируется программно через API. Может быть реализован как эффект постоянного свечения, так и затухающего, пульсирующего и любой другой модели. Для включения возможности подсветки объектов необходимо убедиться, что в панели **Render > Object Outlining** поле **Enable** установлено в состояние **ON** или **AUTO**.

### 15.6.2 Дополнительные настройки

На панели **Object > Selection and Outlining**:

**Enable Outlining** Разрешить использование эффекта подсветки контура на конкретном объекте.

**Duration** Длительность анимации подсветки, сек. Значение по умолчанию 1.

**Period** Период повторения анимации подсветки, сек. Значение по умолчанию 1.

**Relapses** Количество итераций анимации подсветки. В случае 0 анимация будет повторяться бесконечно. Значение по умолчанию 0.

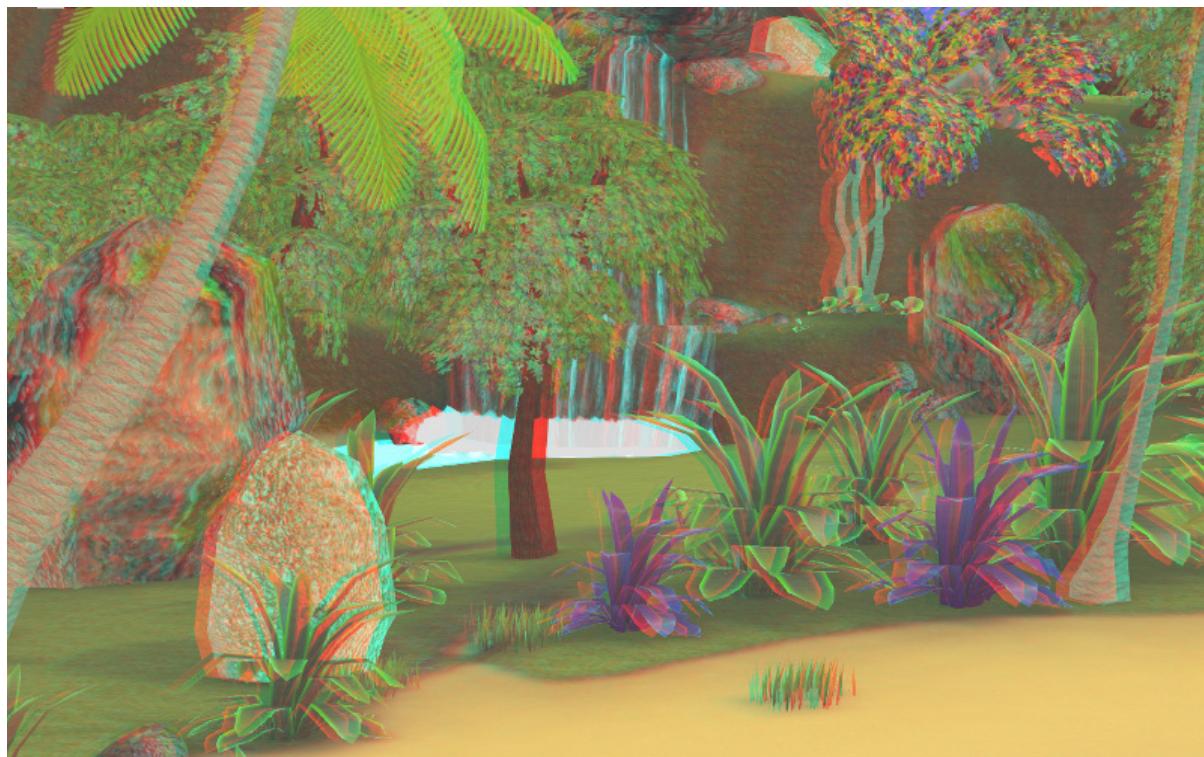
**Outline on Select** Активация анимации подсветки при выделении объекта. Для данного случая необходимо включить опцию **Selectable**. При необходимости реализации собственной модели подсветки объекта следует отключить эту опцию во избежание конфликта.

На панели **Render > Object Outlining**:

**Factor** Толщина и яркость ореола, окружающего объект. Падает с уменьшением параметра. Значение по умолчанию 1.

При управлении через API настройки на панели **Render > Object Outlining** воспринимаются как настройки по умолчанию.

## 15.7 Аналиф стереоизображение



### 15.7.1 Активация

Режим стереоизображения предназначен для просмотра контента в специальных очках и активируется приложением.

### 15.7.2 Дополнительные настройки

Нет.

## 15.8 Коррекция цвета



### 15.8.1 Активация

Активировать панель Color Correction во вкладке Render.

### 15.8.2 Дополнительные настройки

*Brightness* Яркость. Значение по умолчанию 0.0.

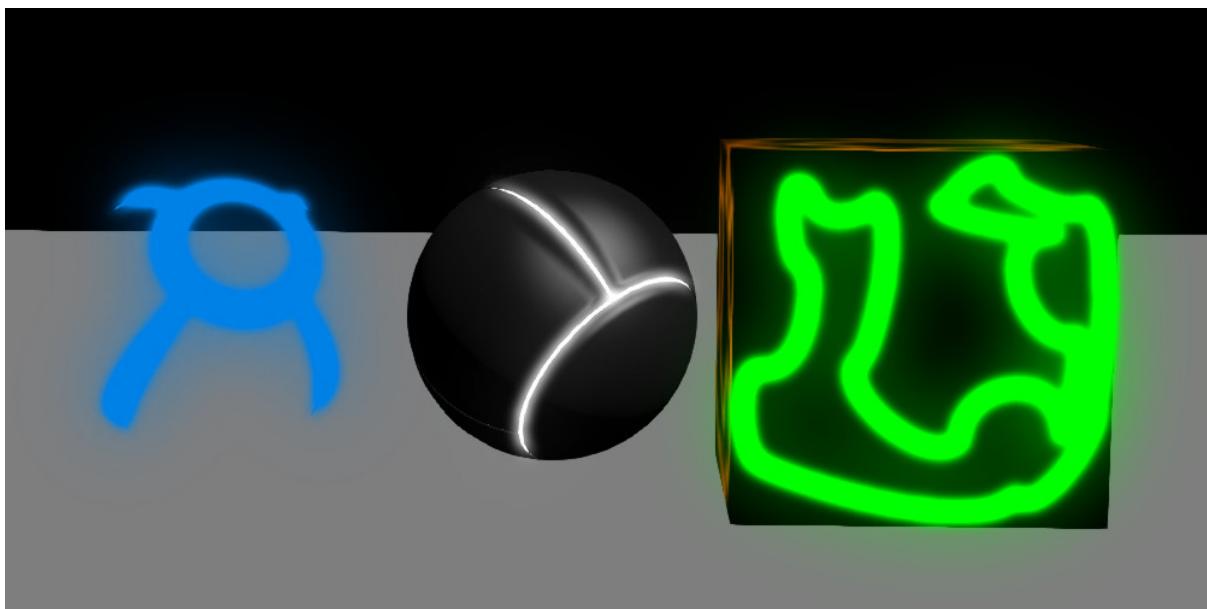
*Contrast* Контрастность. Значение по умолчанию 0.0.

*Exposure* Экспозиция. Значение по умолчанию 1.0.

*Saturation* Насыщенность. Значение по умолчанию 1.0.

## 15.9 Свечение (glow)

Эффект возникновения ореола вокруг светящихся объектов, возникающий вследствие рассеивания света в атмосфере и внутри человеческого глаза.



### 15.9.1 Активация

Добавить в нодовый материал ноду *B4W\_GLOW\_OUTPUT*. Для включения возможности свечения в панели Render > Glow Materials опция Enable должна быть установлена в состояние ON или AUTO.

### 15.9.2 Дополнительные настройки

***Small Mask: Intensity*** Интенсивность свечения по малой маске. Значение по умолчанию 2.0.

***Small Mask: Width*** Ширина свечения по малой маске. Значение по умолчанию 2.0.

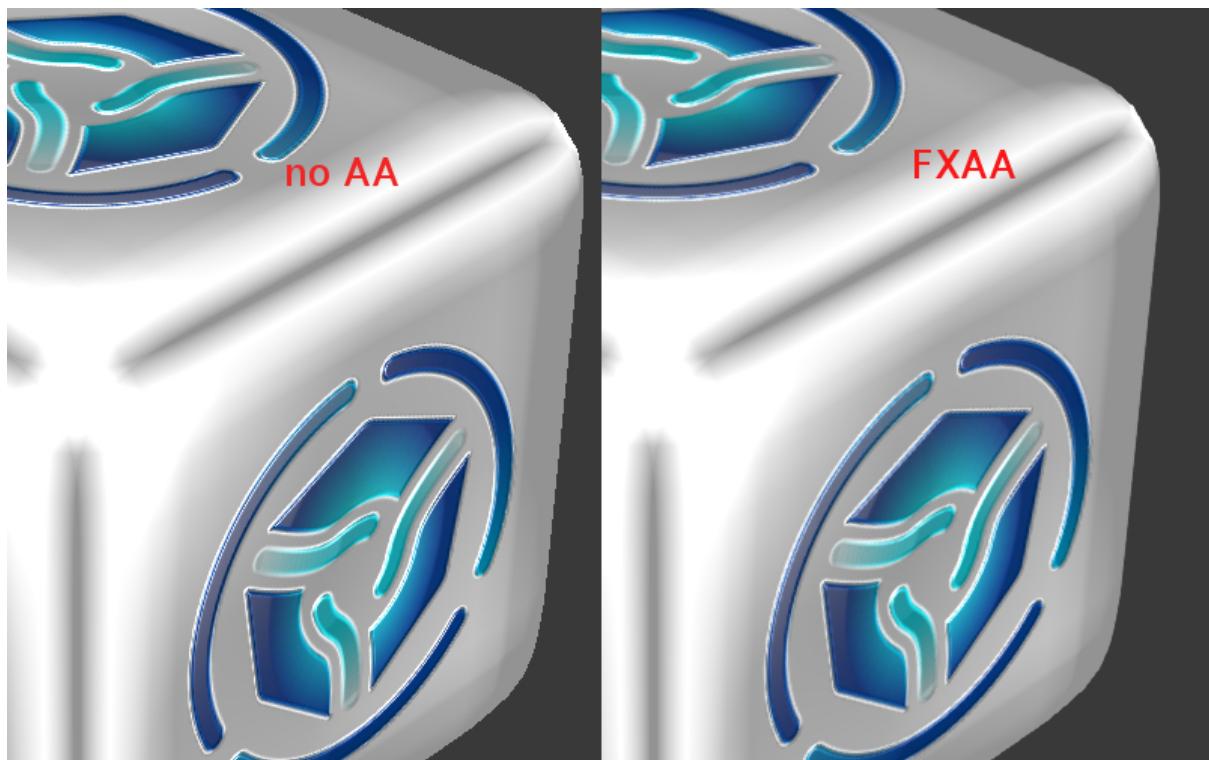
***Large Mask: Intensity*** Интенсивность свечения по большой маске. Значение по умолчанию 2.0.

***Large Mask: Width*** Ширина свечения по большой маске. Значение по умолчанию 6.0.

***Render Glow Over Transparent Object*** Отрисовывание эффекта свечения поверх прозрачных объектов. По умолчанию выключено.

## 15.10 Сглаживание

Сглаживание (anti-aliasing) необходимо для уменьшения влияния нежелательных артефактов рендеринга (“зубчатости”).



### 15.10.1 Активация

Выбрать один из профилей качества в поле AA Quality на панели Render > Anti-Aliasing:

- *None* - антиалиасинг отключен,
- *Low, Medium, High* - применяется антиалиасинг с низкими, средними и высокими параметрами качества соответственно.

Значение по умолчанию: *Medium*.

### 15.10.2 Дополнительные настройки

Итоговые настройки сглаживания назначаются композицией выбора профиля качества антиалиасинга и выбора профиля работы движка. При выборе профиля работы движка действуют следующие настройки:

- *низкое качество* - антиалиасинг отключен,
- *высокое качество и максимальное качество* - использовать алгоритм FXAA 3.11 (Fast Approximate Anti-Aliasing, Nvidia).

### Система частиц. Флюиды

---

Система частиц предназначена для визуализации явлений, обусловленных движением множественных малых объектов, таких как дым, огонь, брызги воды и др.



Необходимым элементом системы частиц является эмиттер - объект, определяющий местоположение и направление исходящего потока частиц.

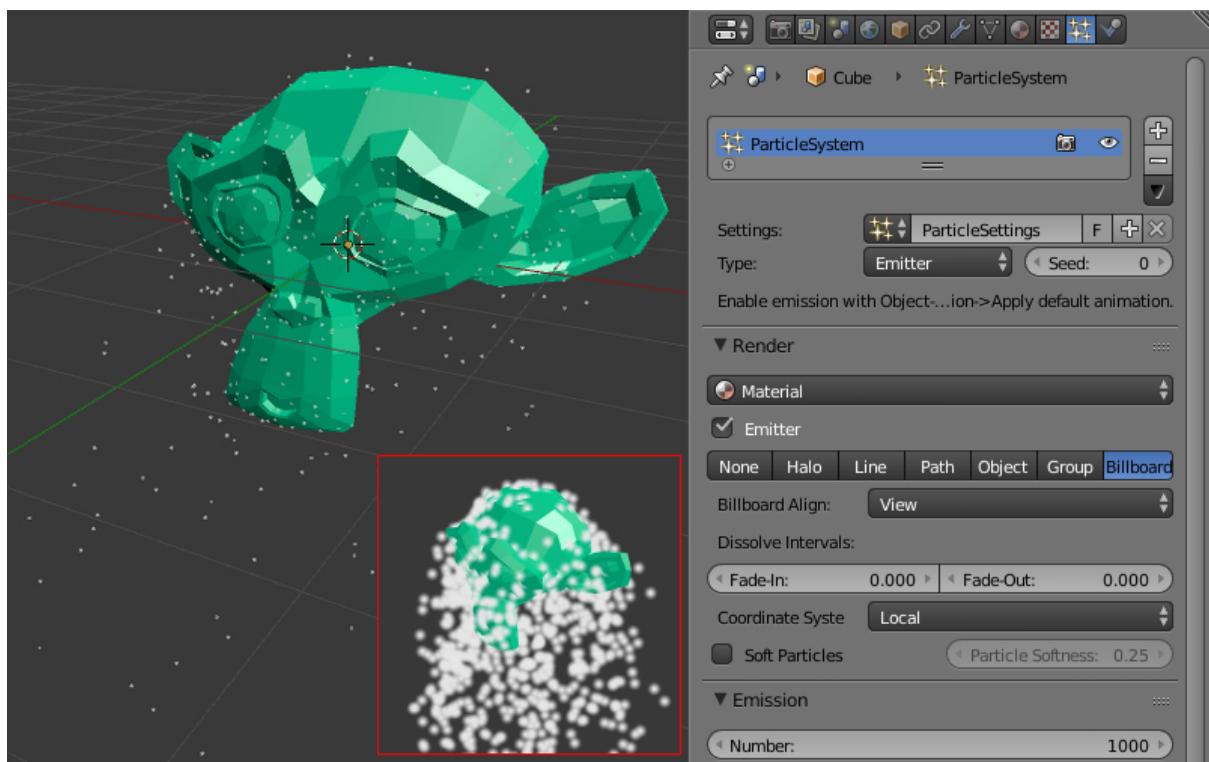
## 16.1 Использование

### 16.1.1 Необходимые этапы

1. Добавить на сцену меш - эмиттер.
2. Создать на эмиттере материал для частиц, например типа **Halo**. Поддерживается также материал типа **Surface** с обязательной диффузной текстурой.
3. Добавить на эмиттере систему частиц.
4. Инициализировать воспроизведение в движке. Возможны два варианта:
  - “циклическое испускание” - для системы частиц выставить опцию **Emission > Cyclic emission**.
  - “нециклическая анимация” - для эмиттера выставить опцию **Animation > Apply Default Animation**.

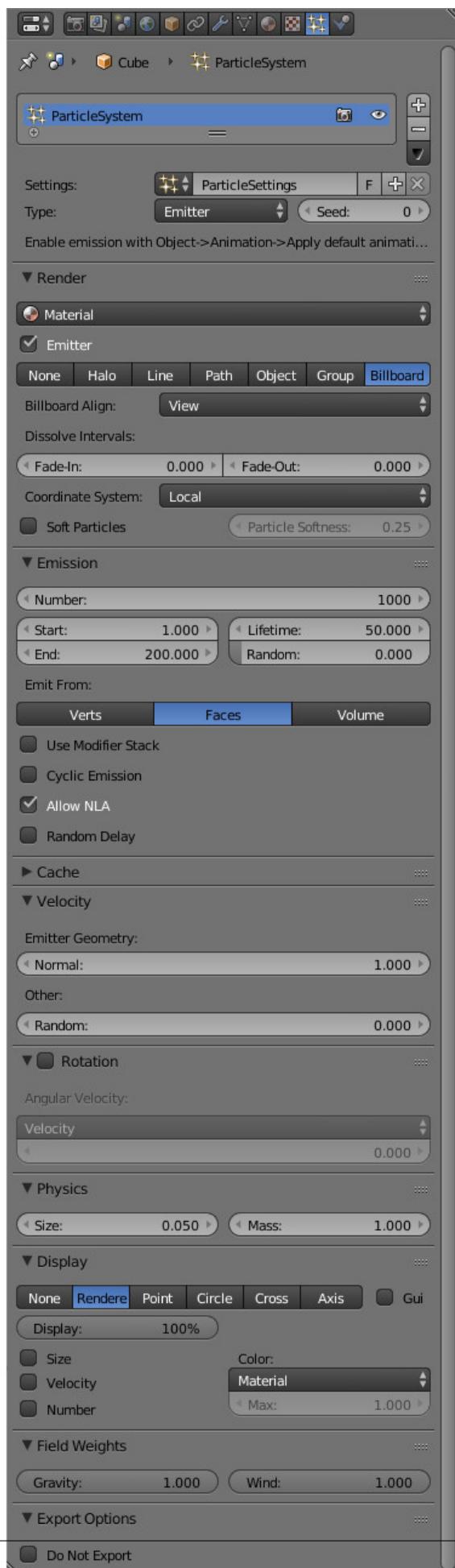
### 16.1.2 Рекомендуемые дополнительные настройки

1. Для материала частиц выставить тип прозрачности **Add**.
2. Если отображение эмиттера на сцене не требуется, отключить опцию **Particles > Render > Emitter**.
3. Если отображение эмиттера на сцене необходимо, для него можно использовать дополнительные материалы. В этом случае в настройках системы частиц нужно выбрать материал частиц **Particles > Render > Material**.
4. В случае использования для частиц материала типа **Surface**, к материалу необходимо подключить диффузную текстуру (обычно с альфа-каналом). В меню **Mapping > Coordinates** выбрать **UV**. Убедиться, что меш эмиттера имеет развертку.



## 16.2 Настройка

Параметры системы частиц настраиваются во вкладке **Particles**. Поддерживается несколько систем частиц на одном эмиттере.



### 16.2.1 Общие настройки

**Name** Название системы частиц. Значение по умолчанию “ParticleSystem”.

**Settings** Ссылка на блок данных с настройками системы частиц. Блоки данных с настройками могут быть общими для разных систем частиц.

**Type** Тип системы частиц: **Emitter** или **Hair**. Системы частиц типа **Hair** используются для создания множественных копий (инстансинга) объектов. Значение по умолчанию **Emitter**.

**Seed** Индекс в таблице случайных чисел, используемых для генерации системы частиц. Значение по умолчанию 0.

### 16.2.2 Настройки испускания

**Emission > Number** Количество частиц. Значение по умолчанию 1000.

**Emission > Start** Первый кадр, после которого начинается испускание частиц. Значение по умолчанию 1.0.

**Emission > End** Последний кадр, после которого прекращается испускание частиц. Значение по умолчанию 200.0.

**Emission > Lifetime** Время жизни частиц в кадрах. Значение по умолчанию 50.0.

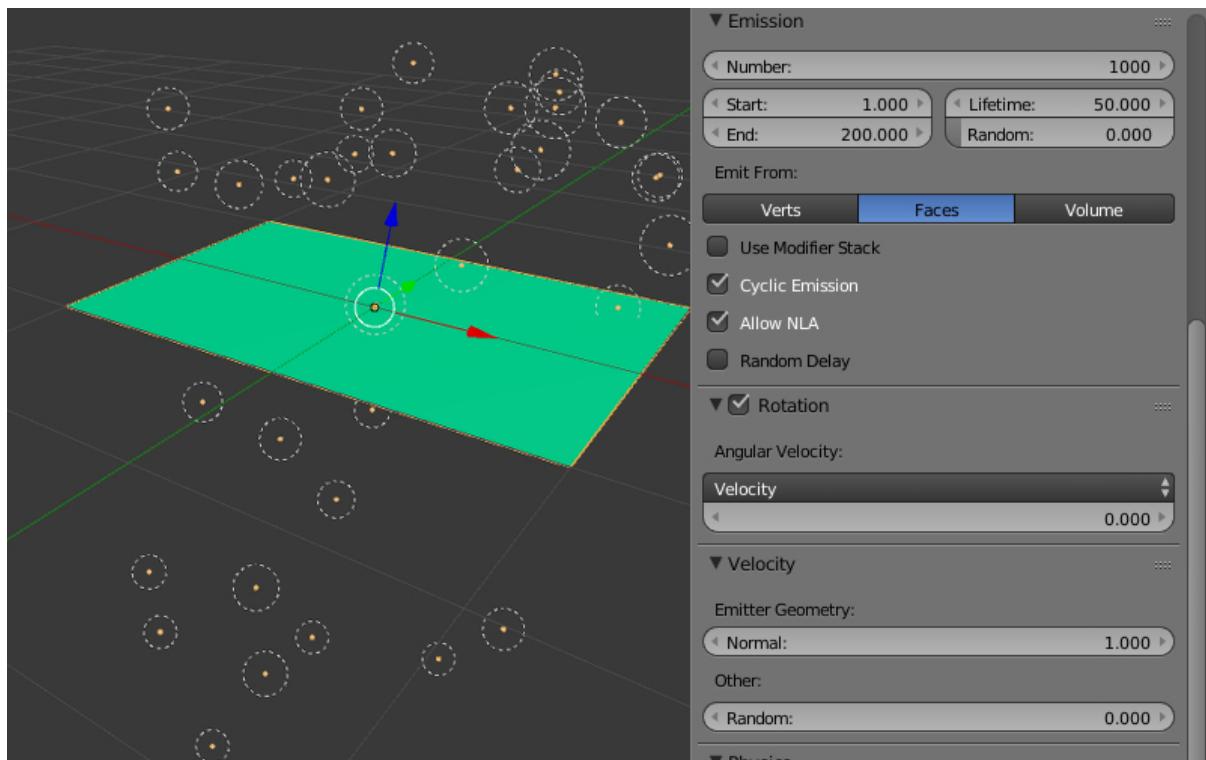
**Emission > Lifetime > Random** Фактор случайности для времени жизни. Значение по умолчанию 0.0.

**Emission > Emit From** Источник испускания. Поддерживаются вершины **Verts**, грани **Faces**. Значение по умолчанию **Faces**.

**Emission > Cyclic emission** Опция включает циклический режим испускания. Применяется для постоянных эффектов (дым, горение, брызги). Рекомендуется выставить нулевое значение **Emission > Start**. По умолчанию выключено.

**Emission > Random Delay** Опция устанавливает случайный характер времени испускания частиц. По умолчанию выключено.

**Emission > Emit From > Distribution** Настройки распределения испускания: **Jittered**, **Random**, **Grid**. Игнорируются движком. Всегда используется случайное распределение (**Random**). Значение по умолчанию **Jittered**.



### 16.2.3 Настройки направления

**Velocity > Emitter Geometry > Normal** Фактор влияния на испускание вдоль нормалей меша эмиттера. Значение по умолчанию 1.0.

**Velocity > Other > Random** Фактор случайности для направления испускания. Значение по умолчанию 0.0.

### 16.2.4 Настройки вращения

**Rotation > Angular Velocity > Mode** Режим собственного вращения биллбордов частиц. Система частиц типа Hair поддерживает все типы вращения, а система частиц типа Emitter только Velocity

**Rotation > Angular Velocity > Factor** Фактор скорости собственного вращения биллбордов частиц. Значение по умолчанию 0.0.

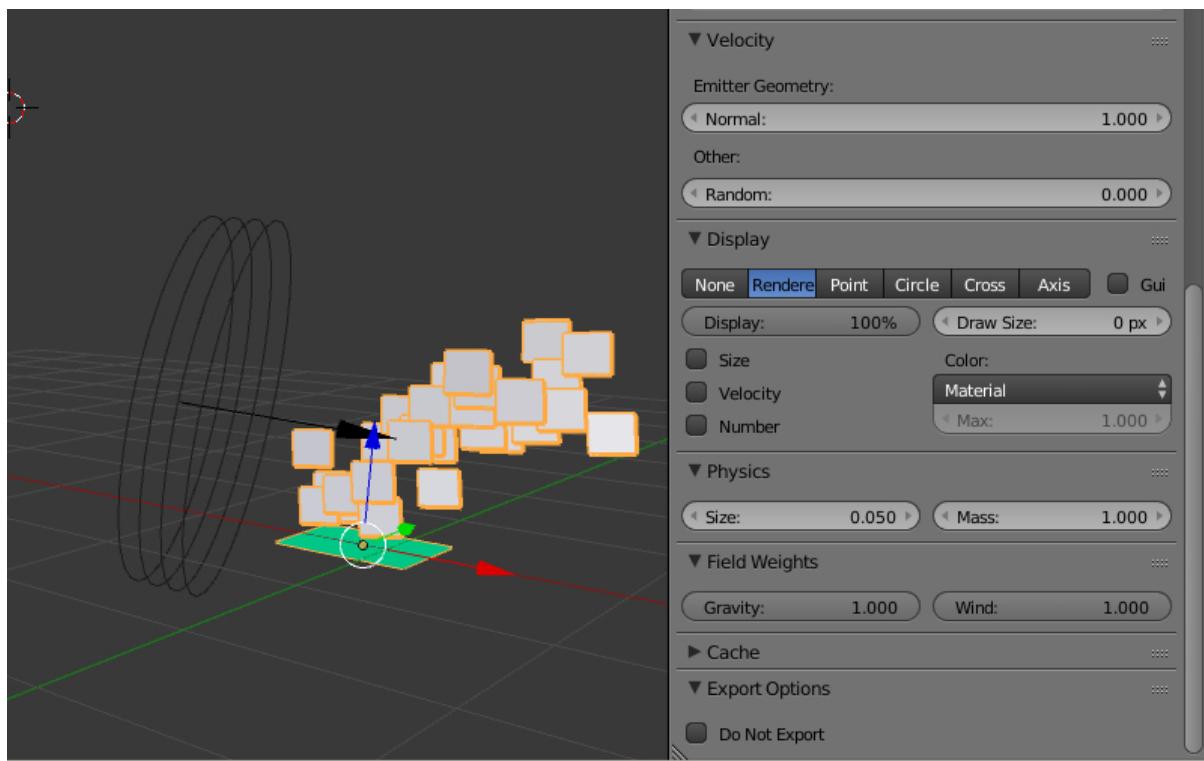
### 16.2.5 Настройки физики

**Physics > Type** Тип расчетов физики: No, Newtonian, Keyed, Boids, Fluid. Игнорируется движком. Всегда используется физика Ньютона (Newtonian). Значение по умолчанию Newtonian.

**Physics > Size** Размер частиц. Значение по умолчанию 0.05.

**Physics > Mass** Масса частиц. Влияет на взаимодействие с силовыми полями (в частности, с ветром). Значение по умолчанию 1.0.

**Physics > Forces > Brownian** Экспортируется, но не используется движком.



### 16.2.6 Настройки отображения

**Render > Material** Меню выбора материала частиц. Используется в случае использования эмиттером нескольких материалов. Значение по умолчанию Default Material.

**Render > Emitter** Опция включения отображения эмиттера на сцене. По умолчанию включено.

**Render > Type** Режим отображения частиц: None, Halo, Line, Path, Object, Group, Billboard. Движком различаются режимы Object и Group, использующиеся для инстансинга объектов и групп объектов, соответственно. Другие режимы игнорируются. Для удобства отображения биллбордов рекомендуется включать режим Billboard. Значение по умолчанию Halo.

**Render > Billboard align** Способ ориентирования биллбордов: View - поворачивать к камере, XY plane, YZ plane, ZX plane - ориентировать в соответствующей плоскости (в мировой системе координат Blender'a). Значение по умолчанию View.

**Render > Dissolve intervals > Fade-in и Fade-out** Начальный и конечный интервалы (в кадрах) для постепенного увеличения и уменьшения прозрачности частиц.

**Render > Coordinate system** Система координат испускаемых частиц: Local - использовать локальную систему координат эмиттера, World - использовать мировую систему координат.

**Render > Soft Particles** Включает так называемые “Мягкие частицы” билборды которых плавно растворяются при соприкосновении с поверхностями. С помощью слайдера Particle Softness настраивается степень проявленности этого эффекта.

### 16.2.7 Настройки влияния силовых полей

**Field Weights > Gravity** Фактор влияния гравитационного поля (земное притяжение). Значение по умолчанию 1.0.

**Field Weights > Wind** Фактор влияния ветра. Необходимо присутствие объекта силового поля (добавляется Add > Force Field) типа Wind (ветер). На систему частиц оказывают также настройки направления и силы ветра. Значение по умолчанию 1.0.

### 16.2.8 Специальные настройки движения

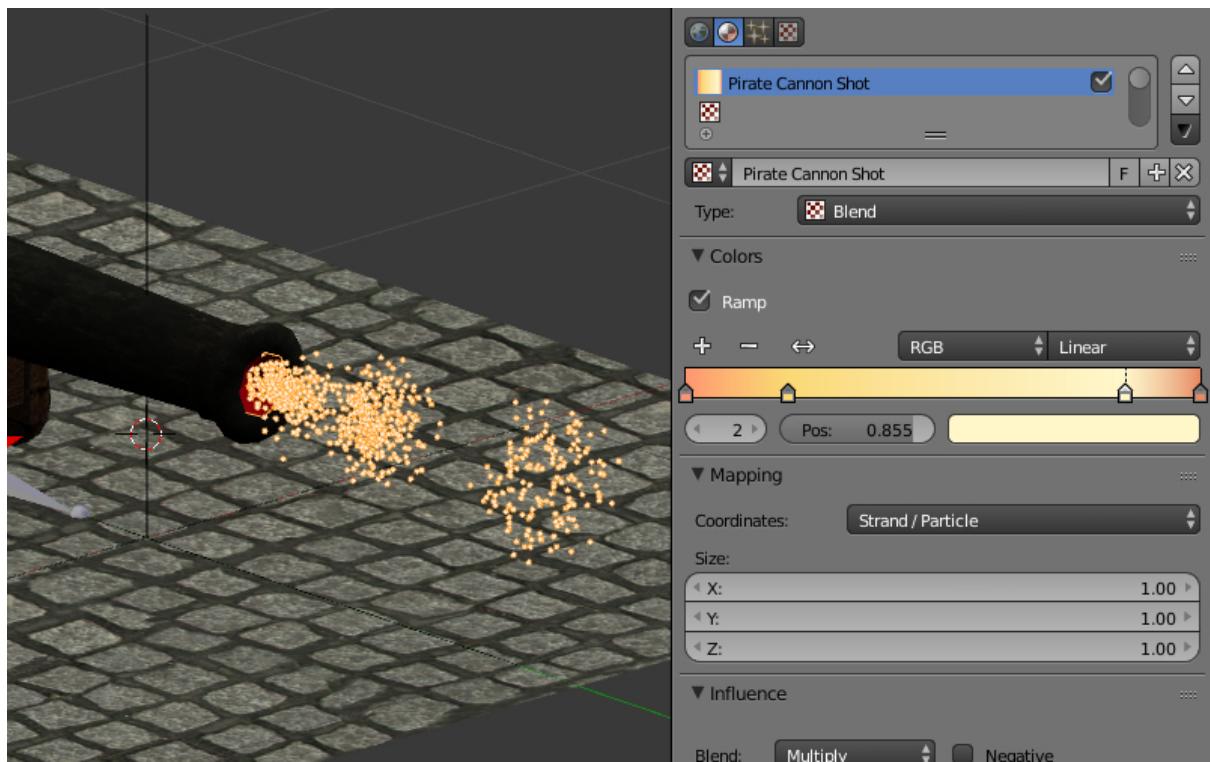
**Export Options > Do not export** Не экспортовать.

## 16.3 Текстуры в системах частиц

### 16.3.1 Текстуры материала частиц

В материалах частиц типа **Surface** **необходимо** наличие диффузной текстуры (обычно с альфа-каналом). В меню **Mapping > Coordinates** выбрать **UV**. Убедиться, что меш эмиттера имеет развертку.

В материалах частиц типа **Halo** **возможно** использование текстуры типа **Blend** с линейным (**Linear**) градиентом. В меню **Mapping > Coordinates** выбрать **Strand / Particle**. На текстуре необходимо включить использование рампы (**Ramp**). Допускается использование до 4 контрольных точек градиента.

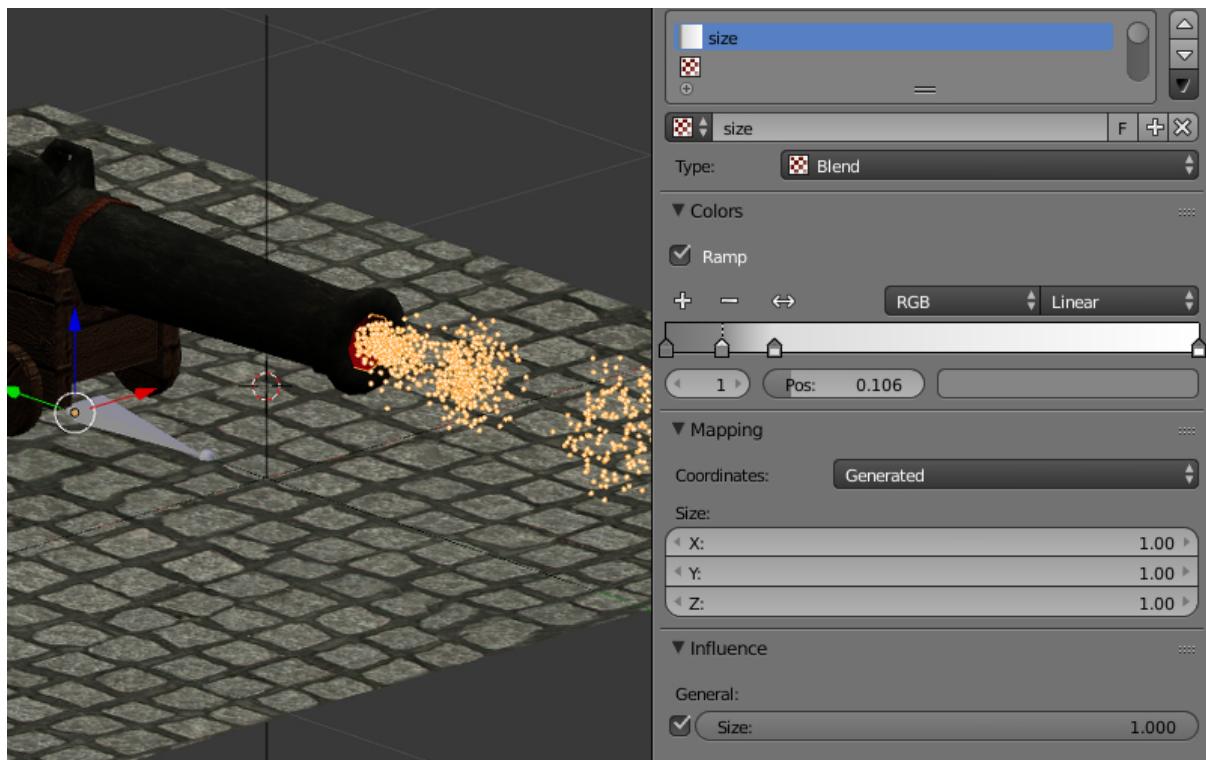


### 16.3.2 Текстуры системы частиц

Для настройки поведения системы частиц могут быть использованы текстуры. В отличие от текстур, используемых материалами частиц, такие текстуры относятся к блоку данных (datablock) системы частиц, а не к блоку данных материала. Чтобы создать текстуру системы частиц, необходимо из вкладки **Particles** перейти во вкладку **Textures**, после чего нажать **New**.

Поддерживаются только текстуры типа **Blend** с линейным (**Linear**) градиентом. На текстуре необходимо включить использование рампы (**Ramp**). Допускается использование до 4 контрольных точек градиента.

На панели **Influence** необходимо выбрать параметр, на который воздействует текстура. В настоящий момент поддерживается только **Size** (размер).



Результат применения текстур градиента для материала частиц и для системы частиц:



[Ссылка на модель](#)

---

## Система частиц. Инстансинг

---

Система частиц может использоваться для создания множественных копий объектов (инстансинга). Кроме сокращения времени на создание сцены, инстансинг позволяет сэкономить на времени загрузки и сократить потребление памяти сценой, нежели при использовании отдельных объектов.



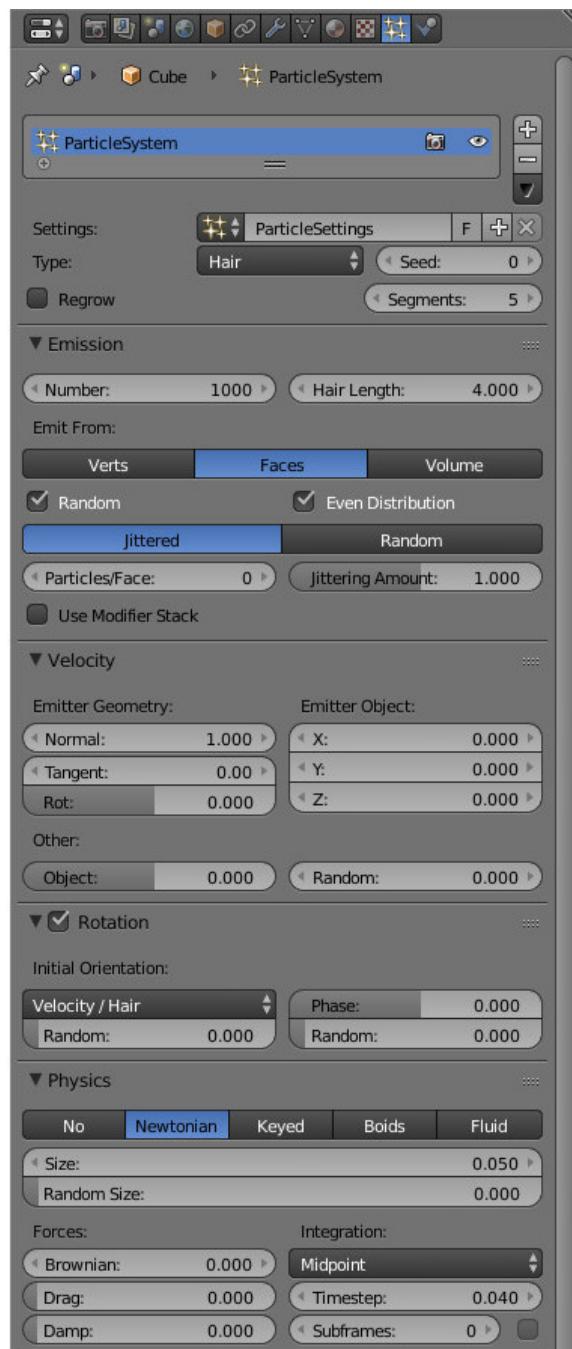
Использование системы частиц для инстансинга имеет ограничения:

- Невозможно осуществлять движение и анимацию объектов внутри системы частиц.
- Не допускается создавать иерархические сущности среди объектов внутри си-

стемы частиц за исключением дуплицирования групп.

- Не допускается инстансинг отличных от объектов-мешей сущностей.

## 17.1 Настройки системы частиц



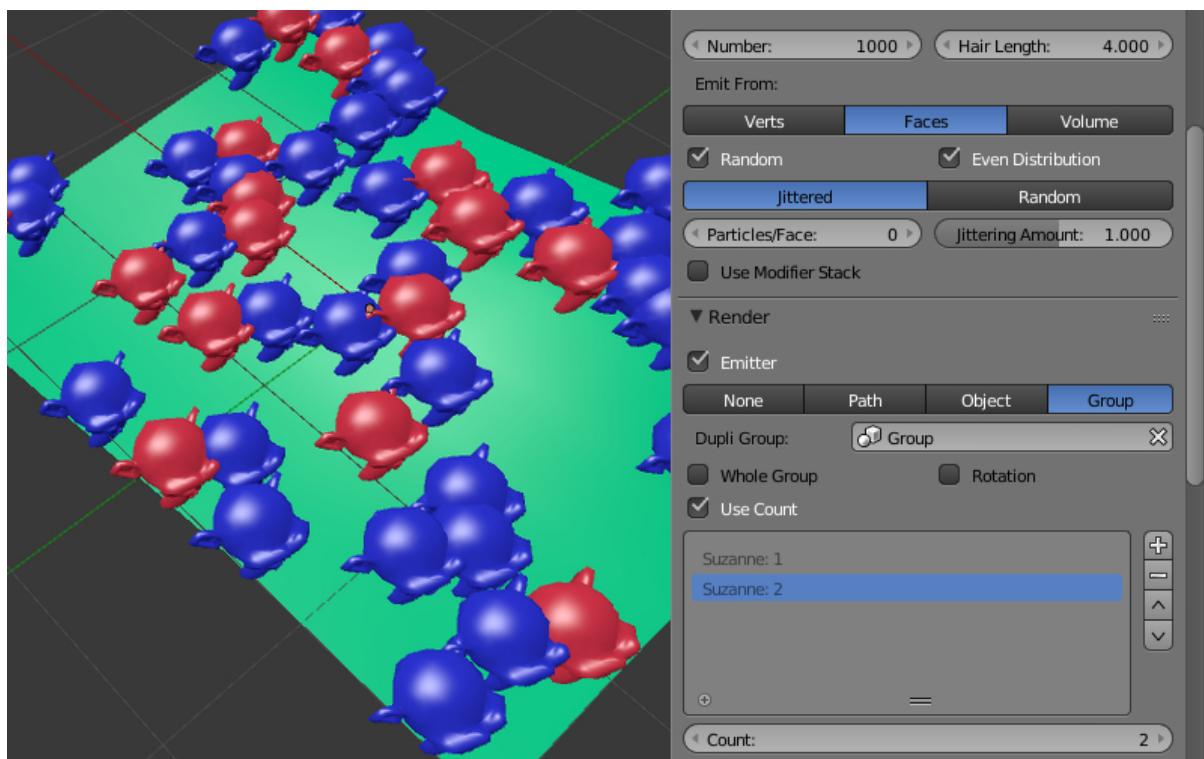


### 17.1.1 Активация

- На эмиттере создать систему частиц типа Hair.
- В панели Render выбрать тип отображения Object (или Group).
- В поле Dupli Object (или Dupli Group) выбрать объект (или группу объектов) для инстансинга. Поддерживаются как локальные, так и подключенные по ссылке объекты (или группы).

#### Рекомендуемые дополнительные настройки

- Для корректного отображения размера установить значение 1.0 для параметров Emission > Hair Length и Render > Size.



### 17.1.2 Настройки отображения

#### *Render > Use Count*

Опция доступна для групп объектов-частиц. При включении появляется интерфейс установки относительного количества входящих в группу объектов. Движок не воспроизводит точное местонахождение объектов заданных типов.

#### *Render > Randomize Location and Size*

Опция устанавливает случайный характер расположения и размеров объектов. Если опция включена, движок генерирует случайные координаты и размер (в пределах  $\pm 25\%$ ) объектов-частиц. Если опция выключена, производится экспорт и использование текущих координат и размеров объектов-частиц.

#### *Render > Randomize Initial Rotation*

Опция устанавливает случайный характер вращения объектов относительно оси определяемой параметром **Rotation Type**. Если опция включена, движок генерирует случайные углы вращения объектов-частиц. Если опция выключена, применяется ориентация согласно панели **Rotation**.

#### *Render > Rotation Type*

Ось случайного поворота объекта (опция доступна при включении **Render > Randomize Initial Rotation**). Возможны два варианта:

- **Z axis** - случайный поворот будет осуществлен относительно вертикальной оси Z
- **Random axis** - случайный поворот будет осуществлен относительно случайной оси

Значение по умолчанию **Z axis**.

#### *Render > Rotation Strength*

Коэффициент, определяющий диапазон случайных углов поворота, отсчитываемых от направления на камеру (опция доступна при включении **Render > Randomize Initial Rotation**). Например:

- **Rotation Strength = 1** - углы будут лежать в пределах  $[-\pi, \pi]$
- **Rotation Strength = 0.5** - углы будут лежать в пределах  $[-0.5 \cdot \pi, 0.5 \cdot \pi]$
- **Rotation Strength = 0.1** - углы будут лежать в пределах  $[-0.1 \cdot \pi, 0.1 \cdot \pi]$

Значение по умолчанию 1.

#### *Render > Billboard*

Включение биллбординга для частиц. По умолчанию выключено.

#### *Render > Billboard Type*

Тип биллбординга (опция доступна при включении **Render > Billboard**). Доступны три типа:

- **Basic** - простой односторонний биллбординг: частицы всегда будут повернуты лицевой стороной
- **Random** - случайный двусторонний биллбординг: частицы чаще всего будут повернуты лицевой, либо обратной стороной, реже - боком; присутствует небольшой случайный поворот; модель создана специально для инстансинга травы
- **Jittered** - односторонний биллбординг с колебанием частиц в плоскости, обращенной к наблюдателю; модель создана специально для инстансинга листьев деревьев

Значение по умолчанию **Basic**.

#### *Render > Jitter Amplitude*

Коэффициент амплитуды колебаний частиц (опция доступна при выборе типа **Jittered** в **Render > Billboard Type**). При увеличении параметра амплитуда растет. Значение по умолчанию 0.

#### *Render > Jitter Frequency*

Частота колебаний частиц, Гц (опция доступна при выборе типа **Jittered** в **Render > Billboard Type**). Значение по умолчанию 0.

#### *Render > Billboard Geometry*

Тип вращения биллбордов (опция доступна при включении **Render > Billboard**). Доступны два типа:

- **Spherical** - сферический биллбординг, полная ориентация частиц по отношению к наблюдателю, вращение ничем не ограничено;
- **Cylindrical** - цилиндрический биллбординг, вращение частиц только относительно оси Z

Значение по умолчанию **Spherical**.

### 17.1.3 Настройки динамической травы

#### *Dynamic Grass*

Опция включает режим динамического рендеринга травяного покрова. По умолчанию отключено.

#### *Dynamic Grass > Scale Threshold*

Минимальный размер частиц динамической травы. Частицы меньшего размера не отображаются. Опция доступна при включении **Dynamic Grass**.

### 17.1.4 Настройки наследования

#### *Properties Inheritance > Wind Bending*

Наследование частицами настроек **Wind Bending**:

- **Parent** - наследование с эмиттера
- **Instance** - наследование с объекта самой частицы

Значение по умолчанию **Parent**.

#### *Properties Inheritance > Shadows*

Наследование частицами настроек теней:

- **Parent** - наследование с эмиттера
- **Instance** - наследование с объекта самой частицы

Значение по умолчанию **Parent**.

#### *Properties Inheritance > Reflection*

Наследование частицами настроек отражений:

- **Parent** - наследование с эмиттера

- **Instance** - наследование с объекта самой частицы

Значение по умолчанию **Parent**.

#### *Properties Inheritance > Vertex color*

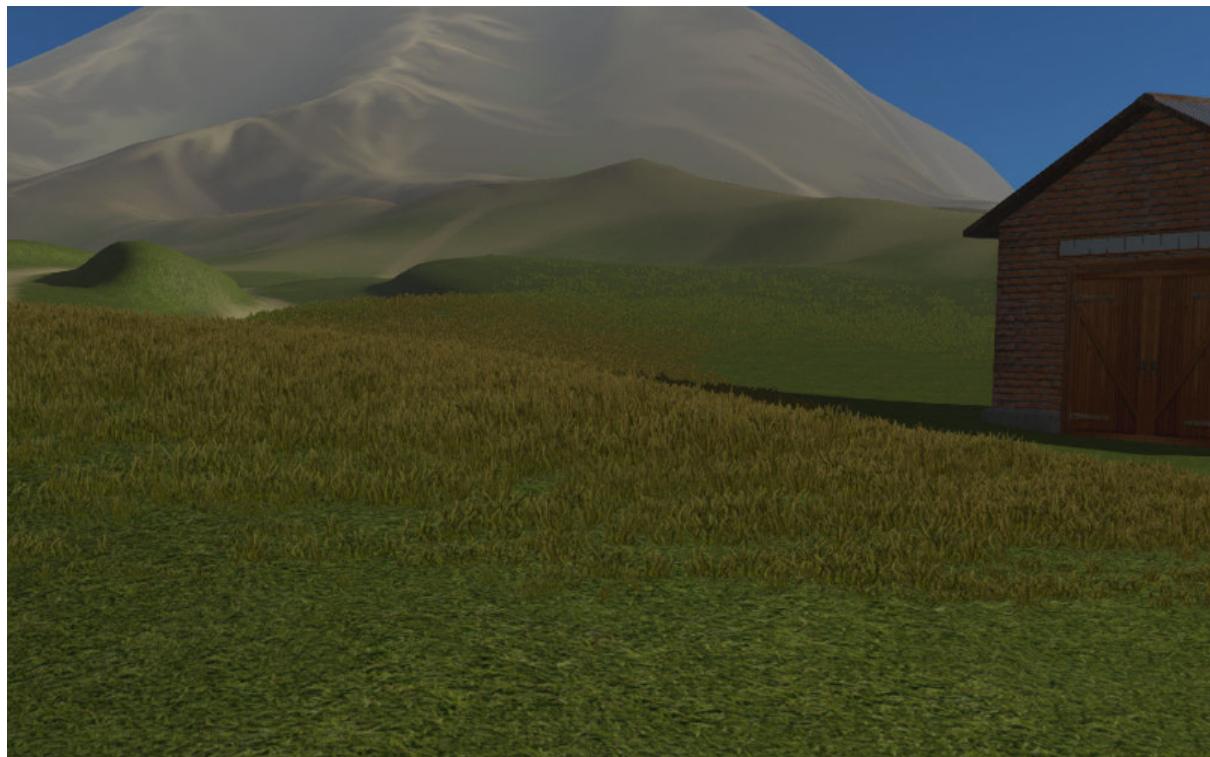
Наследование частицами вертексного цвета с эмиттера. Содержит два поля:

- **From** - имя существующего у эмиттера вертексного цвета
- **To** - имя существующего у частицы вертексного цвета

По умолчанию наследования не происходит.

## 17.2 Травяной покров

Инстансинг объектов может использоваться для визуализации травяного покрова на обширных площадях. При этом происходит отрисовка травы вблизи камеры по мере ее движения по ландшафту.



### Активация

1. На отдельном объекте-плоскости создать систему частиц для инстансинга объектов. Включить опцию **Dynamic Grass**.

- На предполагаемом материале ландшафта включить опцию **Terrain Dynamic Grass**.

## Настройка

Рекомендуется создать несколько плоскостей (например, 3) с размерами, соответствующими желаемому размеру каскада травяного покрова (например, 100, 150 и 250 м).

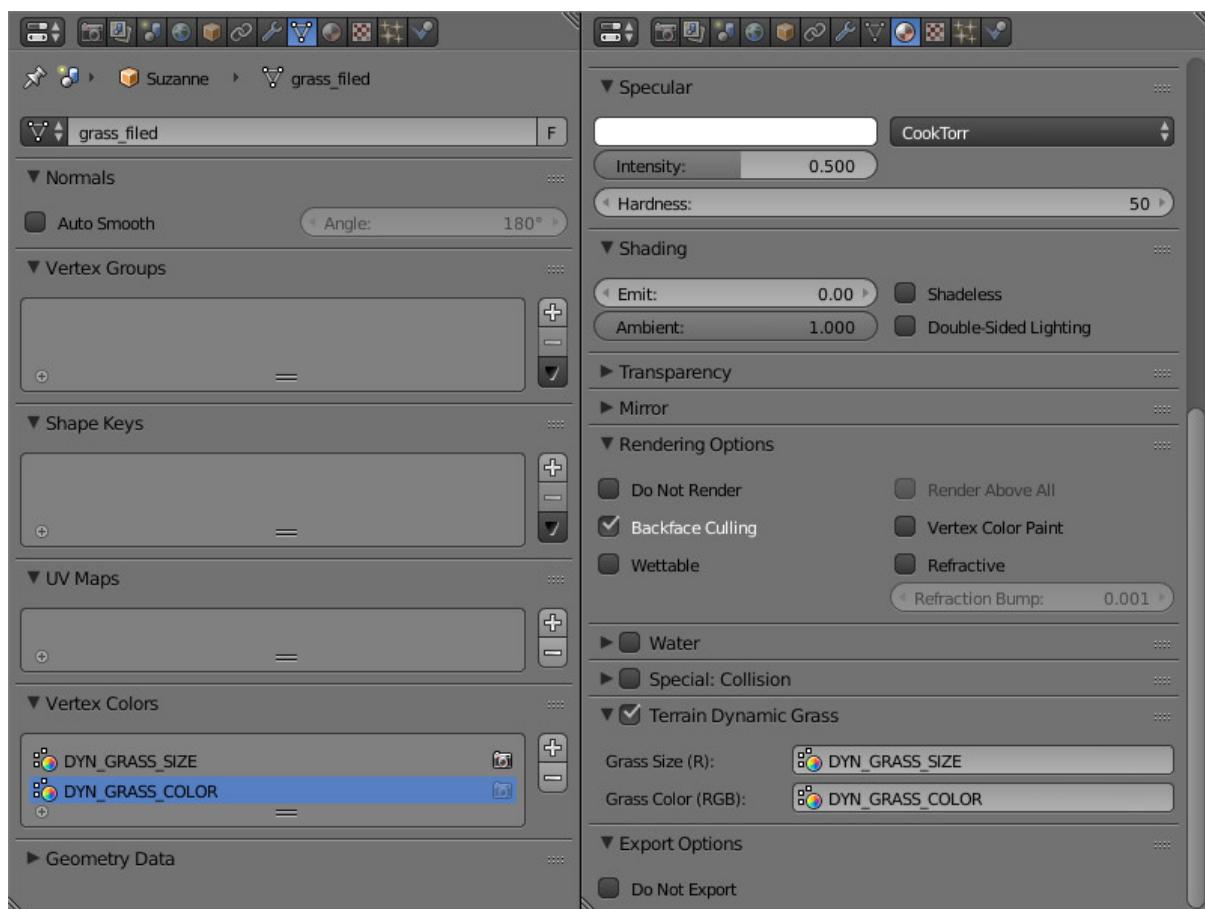
На **материале** ландшафта при включении опции **Terrain Dynamic Grass** становятся активными текстовые поля:

**Dynamic Grass Size (R)** Название слоя вертексного цвета меша ландшафта, предназначенного для модификации размера травяного покрова. Размер (“высота”) травяного покрова задается оттенками серого - чем светлее, тем больше.

**Dynamic Grass Color (RGB)** Название слоя вертексного цвета меша ландшафта, предназначенного для подкраски травяного покрова. Вертексный цвет умножается на цвет материала травы. Параметр **Influence > Blend** диффузной текстуры материала травы должен иметь значение **Multiply**.

Слои вертексного цвета с такими названиями должны существовать в меше ландшафта.

Рекомендуется также отключить отображение эмиттера (опция **Render > Emitter**).



## 17.3 Листва деревьев

Инстансинг хорошо подходит для отображения листвы на деревьях, и позволяет добиться более высокого уровня детализации.



### Активация

Осуществляется как описано выше в разделе **Настройки системы частиц -> Активация**. Здесь соответственно эмиттером будет выступать дерево, а частицами - ветки, листья и т.д.

Для эмиттера дополнительно можно сделать следующее:

- создать вертекскую группу, включающую вершины, на которых будут располагаться частицы
- создать слой вертекского цвета для настройки Wind Bending дерева и листвы
- создать слой вертекского цвета для наследования его частицами (можно использовать, например, для подкраски частиц)

### Настройка

1. *Настройки случайного поворота*

Если включена опция Render > Randomize Initial Rotation, то рекомендуется выставить вертикальную ось случайного поворота - Z axis (опция Render >

Rotation Type). Опция Render > Rotation Strength - на свое усмотрение.

### 2. Настройки биллбординга

Рекомендуется включить биллбординг, выставить тип Jittered (опция Render > Billboard Type) и сделать его сферическим - Spherical (опция Render > Billboard Geometry). Настройки Render > Jitter Amplitude и Render > Jitter Frequency - на свое усмотрение.

### 3. Настройки расположения частиц

Рекомендуется выставить опцию Emission > Emit From в значение Verts, а в Vertex Group > Density выбрать вертекскую группу эмиттера с вершинами для расположения частиц. Также нужно отключить опцию Render > Randomize Location and Size.

### 4. Настройки эффекта ветра

Рекомендуется включить наследование настроек из эмиттера - выставить Parent в опции Properties Inheritance > Wind Bending. Затем у эмиттера в панели Object выбрать опцию Wind Bending и настроить параметры бендинга. Для дерева достаточно указать параметры Wind Bending > Main Bending > Angle и Wind Bending > Main Bending > Frequency, а также вертексный цвет для бендинга - Wind Bending > Main Bending > Main Stiffness.

### 5. Настройки наследования вертексного цвета

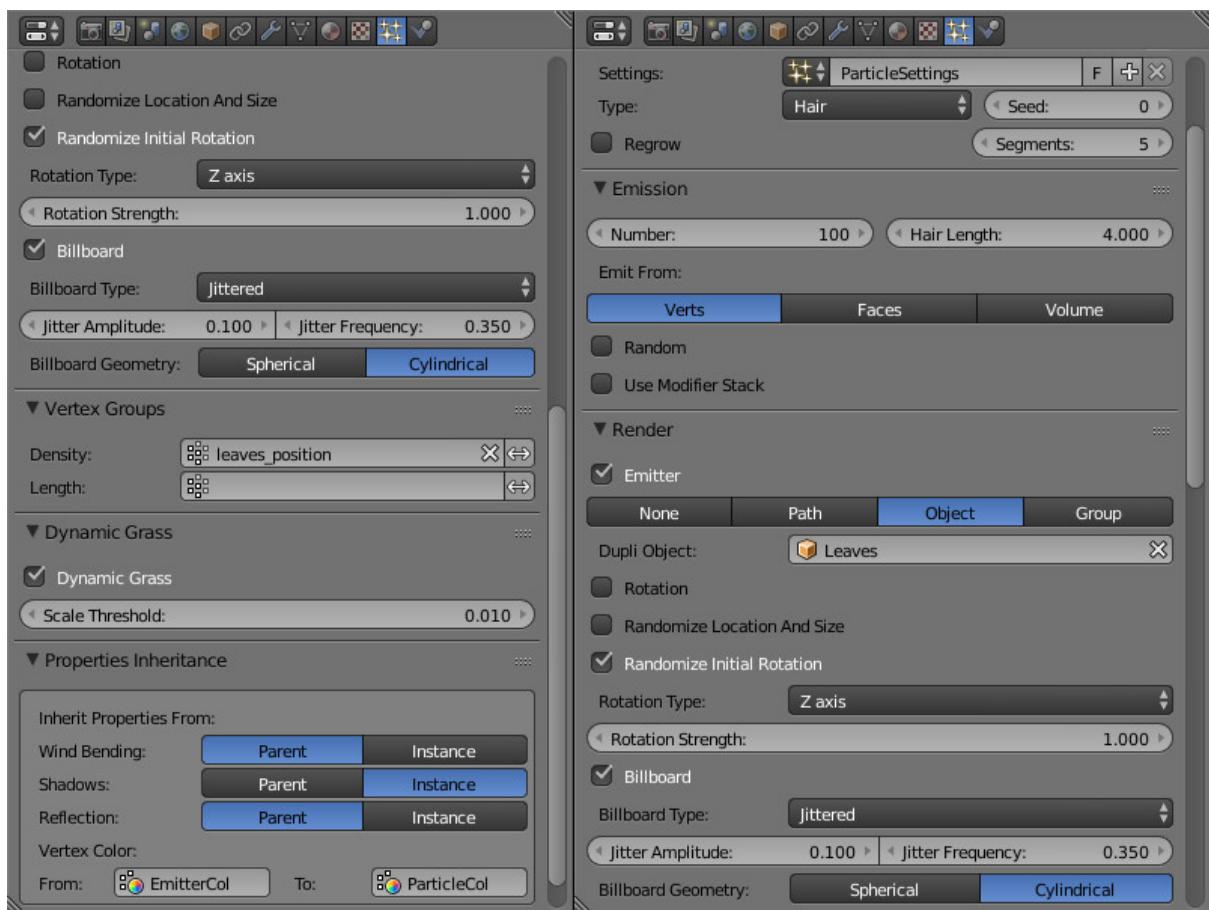
Для наследования частицами вертексного цвета эмиттера нужно указать имя цвета эмиттера и имя цвета частицы соответственно в полях Properties Inheritance > Vertex Color > From и Properties Inheritance > Vertex Color > To. При наследовании цвет ближайшей к частице вершины эмиттера из From будет скопирован и размножен в цвет To частицы.

Полученный таким образом вертексный цвет с именем Properties Inheritance > Vertex Color > To можно будет использовать в новом материале частицы для ее подкрашивания либо каких-то других эффектов.

### 6. Настройки размеров частиц в зависимости от весовых коэффициентов вертекской группы

Для создания зависимости между размером частиц и весовым коэффициентом вертекса из вертекской группы укажите на панели Vertex groups > Length имя вертекской группы.

Для управления эффектом необходимо задать весовые коэффициенты в выбранной вертекской группе.



## Анимация

---

В общем случае, к анимации относятся изменения параметров объектов во времени. Движком поддерживаются следующие типы анимации:

- Объектная анимация. Заключается в трансформации объекта в пространстве как единого целого.
- Скелетная анимация, то есть деформация геометрии объекта с помощью системы костей (скиннинг). Сюда же относится анимация костей в арматурном объекте с целью прикрепления объектов к костям.
- Вертексная анимация. Заключается в покадровой записи деформаций объекта с их последующим воспроизведением.
- Параметризация источников звука. Изменяемые параметры: громкость (*Volume*) и высота звука (*Pitch*).
- Анимация выходного значения ноды *Value* в нодовом материале.
- Процедурная анимация в виде колебаний объекта под действием ветра. Описано *отдельно*.
- Эмиссия частиц из источника. Описано в *соответствующем разделе*.

### 18.1 Управление анимацией

Управление анимацией в движке осуществляется одним из двух способов:

1. Автоматически, с помощью активации панели *Animation* и указания поведения анимации *Behavior* в свойствах объекта. В данном случае будет осуществлён поиск доступного метода и в случае положительного результата, объект анимируется с момента загрузки сцены. В случае скелетной анимации, по умолчанию воспроизводится актор, назначенный на объекте в окне *Action Editor*.
2. Программно, используя функции модуля движка *animation*.

Для отладки анимации имеет смысл использовать интерфейс *Animation* программы-просмотрщика, рассмотренный в *соответствующем разделе*.

## 18.2 Объектная анимация

Изменяемые параметры: координаты центра (**Location**), поворот (**Rotation**) и масштабирование (**Scale**).



Осуществляется с помощью добавления ключей анимации для движения объекта в программе Blender и их последующего воспроизведения в движке.

Поддерживаются следующие типы ключей:

- *Location*
- *Rotation* – необходимо осуществлять в режиме Quaternion(WXYZ) либо XYZ Euler.
- *Scale* – для получения корректных результатов, фактор масштабирования должен быть одинаковым вдоль любых из осей.
- *LocRot* – комбинация *Location* и *Rotation*.
- *LocScale* – комбинация *Location* и *Scale*.
- *LocRotScale* – комбинация *Location*, *Rotation* и *Scale*.
- *RotScale* – комбинация *Rotation* и *Scale*.

В случае анимации объекта-меша, необходимо включение опции **Force Dynamic Object** панели **Rendering Properties** на вкладке свойств объекта.

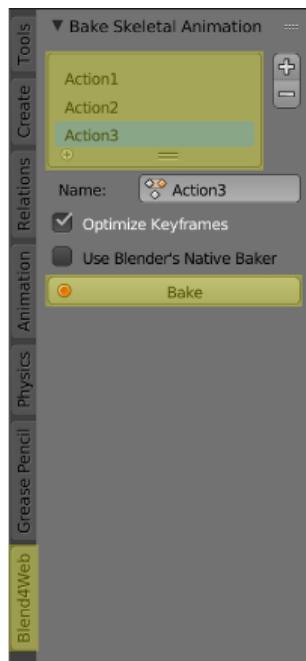
### 18.3 Скиннинг и скелетная анимация



Для осуществления скелетной анимации, кроме деформируемого объекта-меша требуется объект-арматура. Осуществляется в четыре этапа:

1. Создание скелета объекта в арматурном объекте.
2. Назначение вертексных групп в объекте-меше и их привязка к костям. Может быть осуществлено, например, методом “раскраски” весов (weight painting).

3. Анимация костей в арматурном объекте. Используются те же ключи, что и в случае объектной анимации.
4. В случае нетривиальных видов скелетной анимации, включающих инверсную кинематику, требуется стадия запекания анимационных акторов (блок Action в Blender). Запекание производится с помощью интерфейса Bake Skeletal Animation, расположенного на панели инструментов Blend4Web:

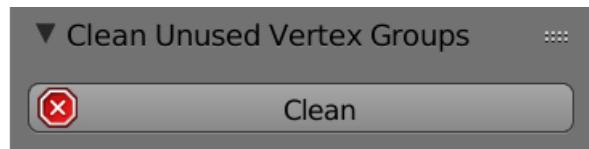


Запекание производится при выделенном арматурном объекте. Элементы интерфейса Bake Skeletal Animation:

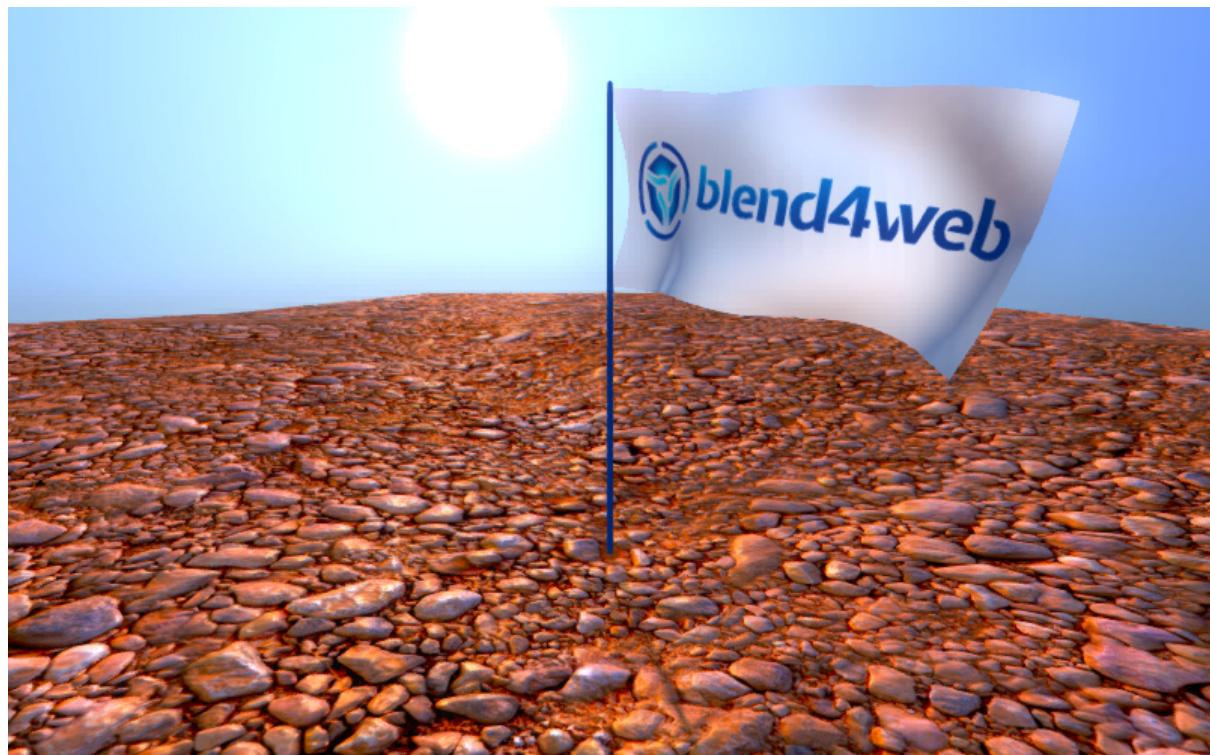
- Окно со списком запекаемых акторов – запекать только те акторы, которые указаны в списке, иначе запекать все возможные акторы.
- *Name* – имя текущего актора из списка запекаемых акторов.
- *Optimize Keyframes* – произвести оптимизацию ключей анимации после запекания. В случае получения некорректных результатов, рекомендуется отключить опцию.
- *Bake* – произвести запекание. После успешного окончания процесса на сцене появляются акторы с именами вида *ИМЯ\_B4W\_BAKED*. Данные акторы будут автоматически назначены на арматурном объекте и воспроизведены в движке. Стоит отметить, что работа подобных акторов в Blender не гарантируется.

**Примечание:** Движок поддерживает не более 4-х вертексных групп на каждом из вертексов, эти группы отбираются по величине влияния или “веса” вертекса. В процессе загрузки исходного файла со сценой “веса” вертексов проходят через процедуру нормализации, т.е. их сумма приводится в единице.

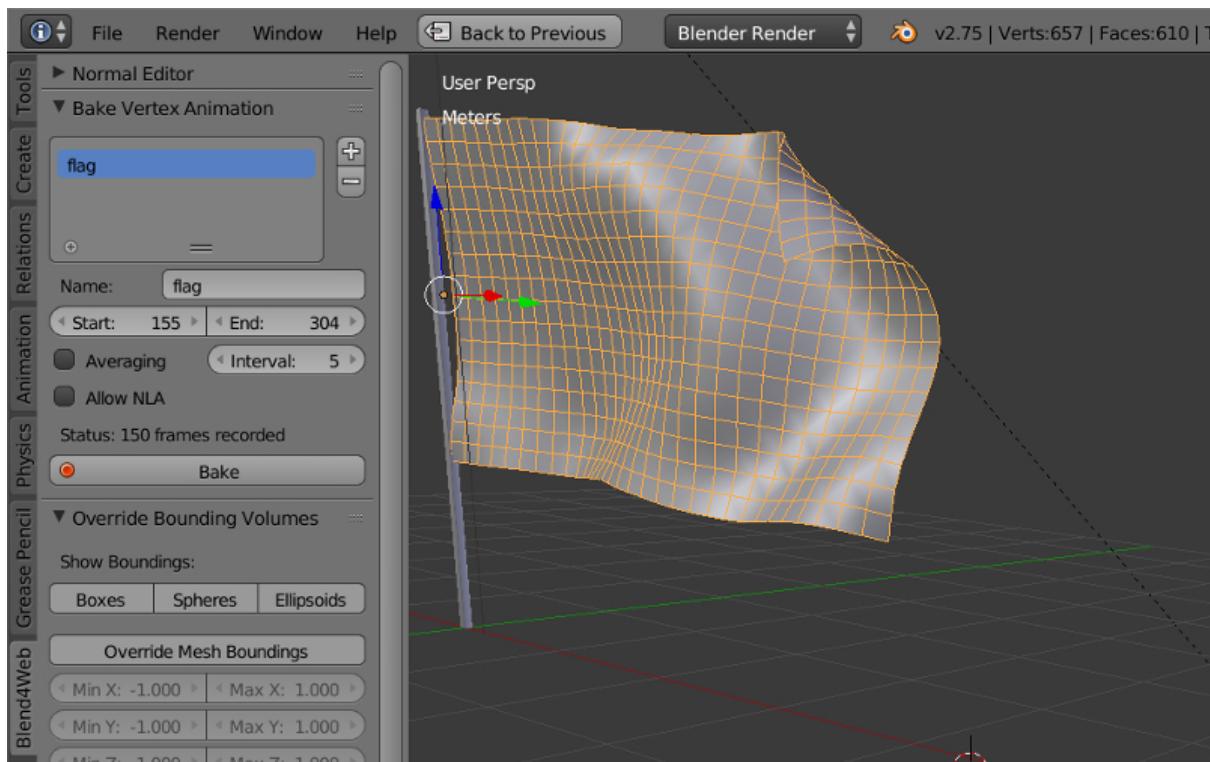
Для удаления вертексных групп, которые не используются арматурой, можно воспользоваться кнопкой **Clean Unused Vertex Groups** в одноименной панели.



## 18.4 Вертексная анимация



Позволяет записать любые изменения геометрии объекта-меша. Необходимо учитывать, что каждый кадр вертексной анимации эквивалентен мешу. Не рекомендуется создание длинной анимации для высокополигонального меша, поскольку это может привести к существенному возрастанию размера исходного и экспортируемого файлов, а также замедлить работу движка. Для запекания вертексной анимации предусмотрен инструмент **Bake Vertex Animation**, расположенный на панели инструментов Blend4Web.



## 18.5 Параметризация источников звука

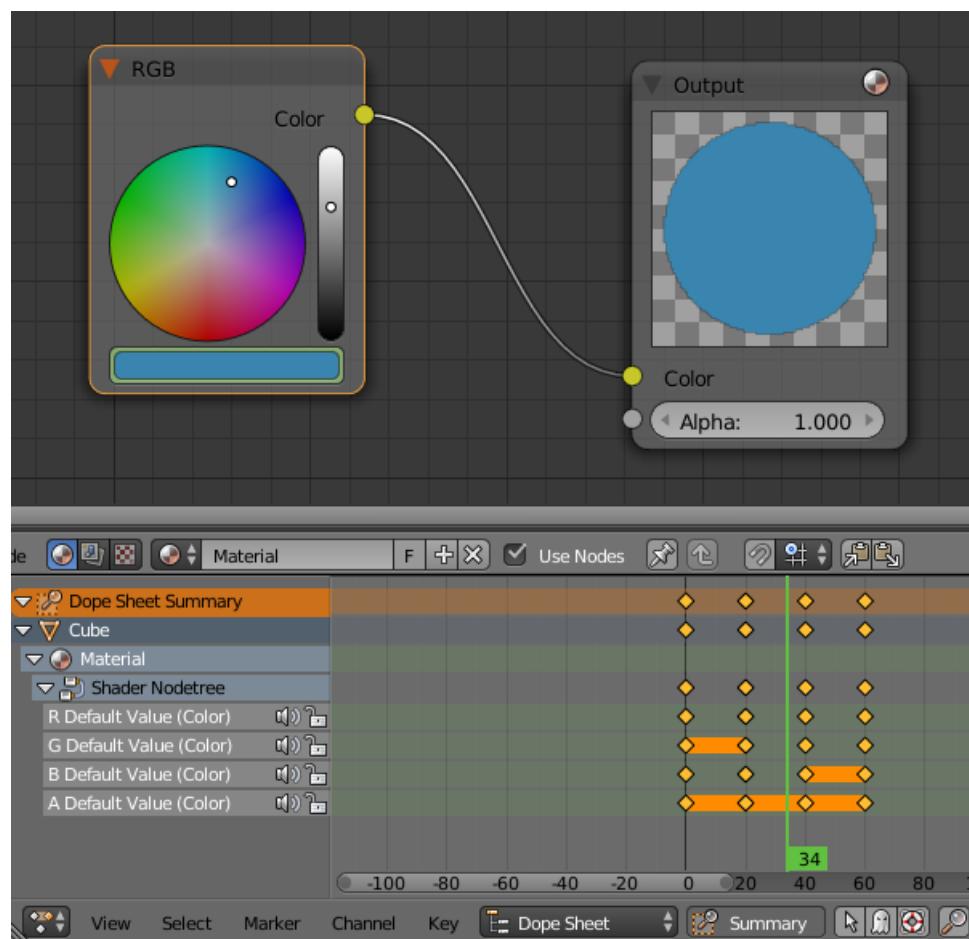
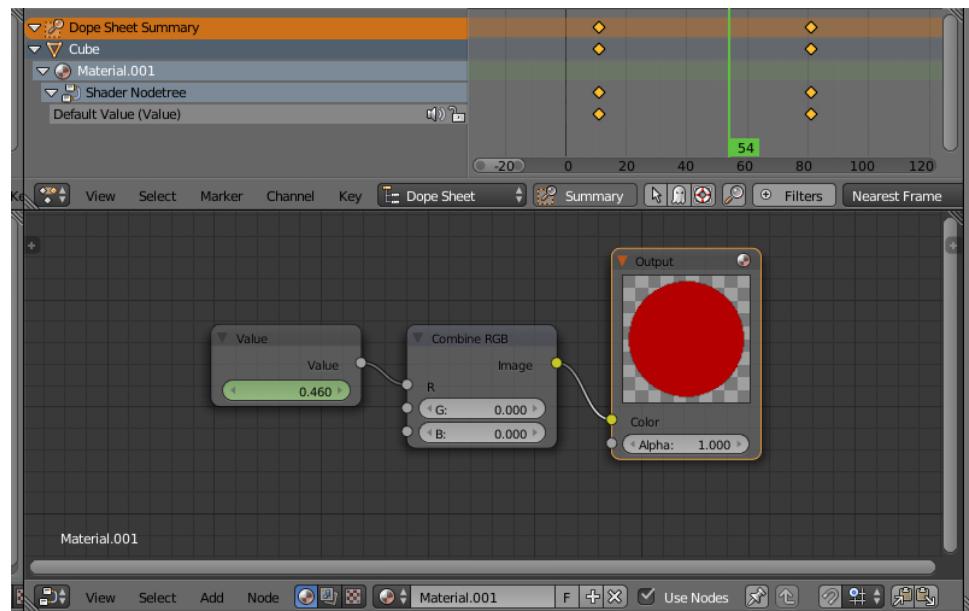
На объектах-спикерах дополнительно поддерживаются следующие типы анимационных ключей:

- *Volume* – громкость звука источника.
- *Pitch* – высота звука источника.

Параметризация источников звука по своей сути повторяет объектную анимацию.

## 18.6 Анимация нод Value и RGB

В нодовых материалах поддерживается воспроизведение анимационных ключей, проставленных на нодах *Value* и *RGB*.



---

**Примечание:** Анимация числовых и цветовых значений в других нодах не поддерживается.

---

Может быть также использована для создания треков в *редакторе нелинейной анимации*. Поддерживается несколько анимированных нод Value и RGB в одном материале. Значения нод могут быть также установлены программно с помощью методов `set_nodemat_value` и `set_nodemat_rgb` модуля `objects`.

**См.также:**

*B4W\_TIME*

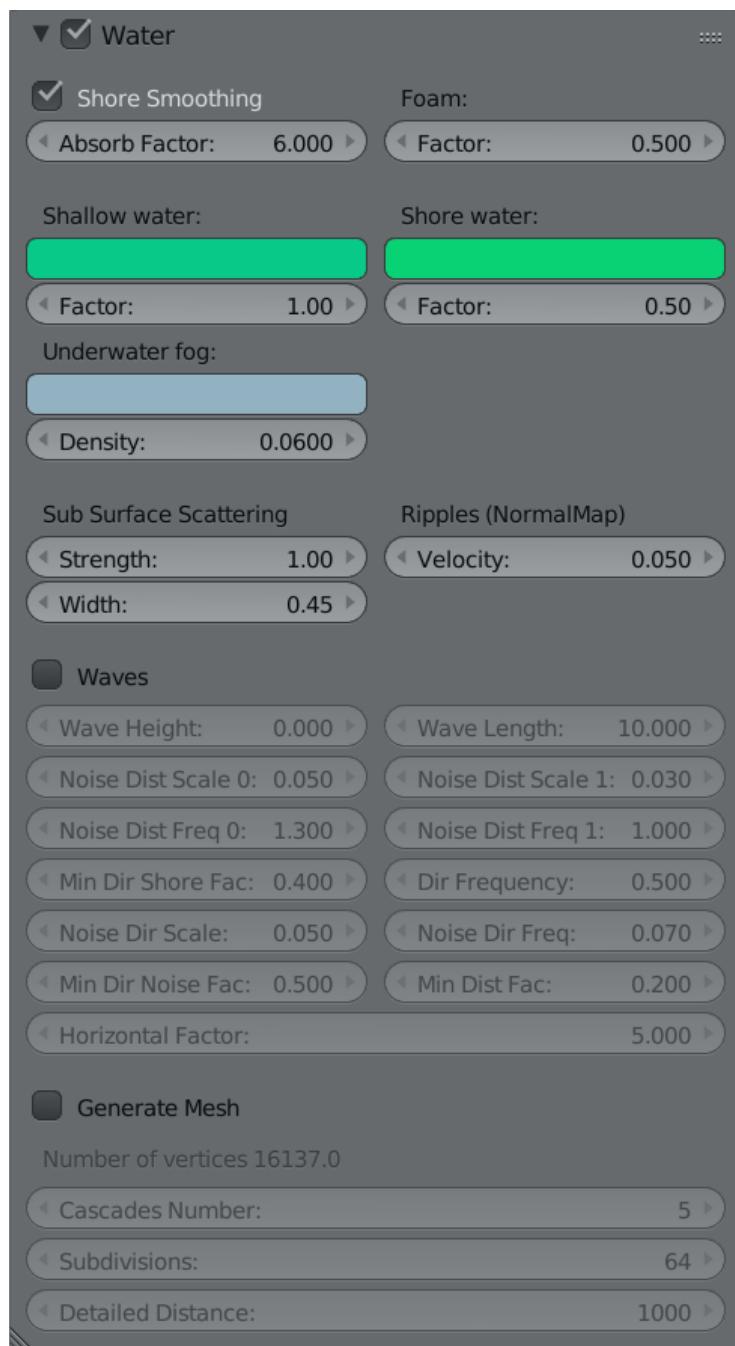
## Рендеринг наружных сцен

---

### 19.1 Вода

#### 19.1.1 Активация

Для предполагаемого материала воды активировать панель `Water` во вкладке `Material`.



### 19.1.2 Базовые настройки

**Прозрачность** Рекомендуется включить прозрачность с градиентом Transparency > Alpha Blend и настроить значение Alpha.

**Параметры освещения** Параметры освещения материала воды настраиваются как описано в разделе [Параметры освещения](#).

### 19.1.3 Динамика волн

Рябь на водной поверхности симулируется при помощи карт нормалей с анимированными развертками (в количестве от 0 до 4). Для текстур - карт нормалей используется только одно общее изображение, текстуры различаются параметрами **Mapping > Size**. Меш для воды должен иметь текстурную развертку.

### 19.1.4 Смачивание поверхностей

Осуществляется автоматически. Для включения эффекта на соответствующих материалах необходимо активировать опцию **Wettable** на панели **Rendering Options**.

### 19.1.5 Отражение и эффект Френеля

Для материала воды поддерживается как статическое, так и динамическое зеркальное отражение, с эффектом Френеля. См. раздел [Зеркальное отражение](#).



### 19.1.6 Сглаживание у береговой линии

Эффект заключается в том, что вода ближе к берегу становится прозрачнее.

**Water > Shore Smoothing** Включить сглаживание.

**Water > Absorb Factor** Коэффициент поглощения света водой. Чем он выше, тем прозрачнее вода.

### 19.1.7 Градиент цвета

Для создания цветного градиента на материале воды должна быть наложена текстура с включенной опцией **Export Options > Shore Distance Map**, генерируемая с помощью скрипта для [запекания параметров береговой линии](#).

**Shallow water > Color** Цвет воды на мелководье.

**Shallow water > Factor** Коэффициент примешивания цвета воды на мелководье.

**Shore water > Color** Цвет воды непосредственно у береговой линии.

**Shore water > Factor** Коэффициент примешивания цвета воды на береговой линии.

### 19.1.8 Преломление

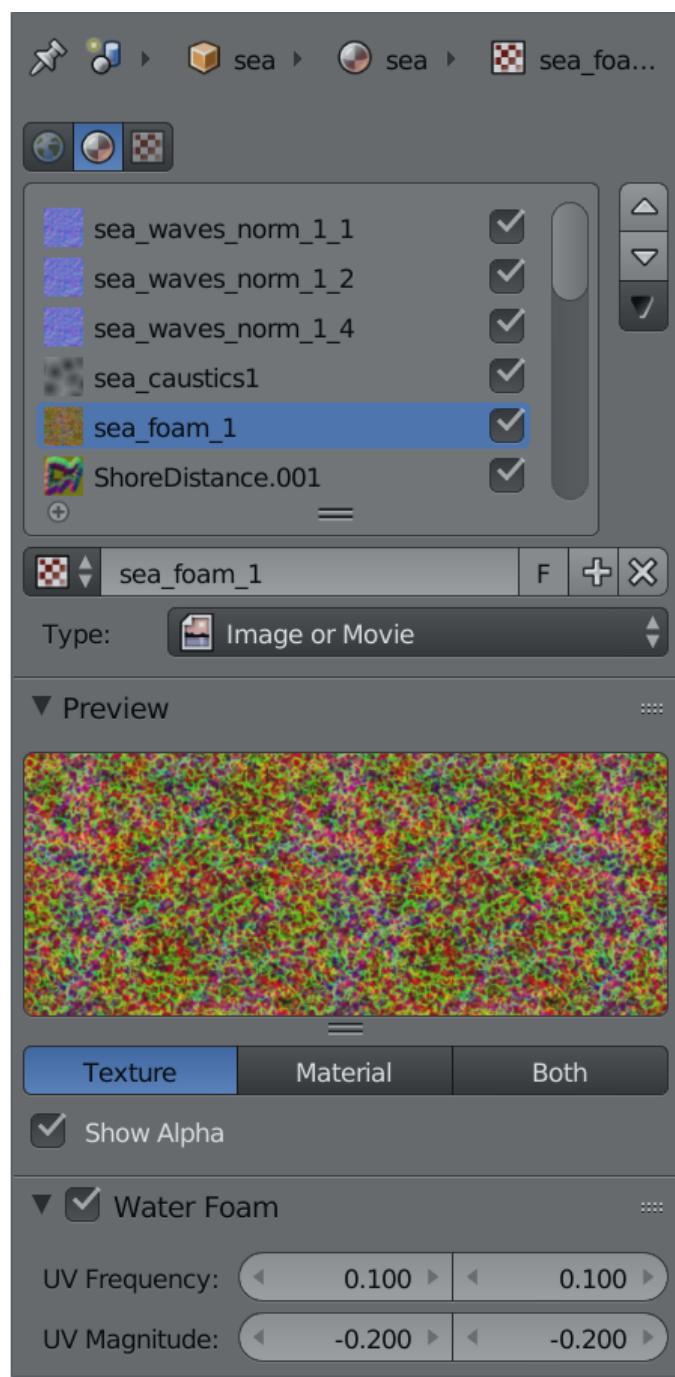
Во вкладке **Render** в панели **Reflections and Refractions** выставить опцию **Refractions** в положение **ON** или **AUTO**.



### 19.1.9 Пена

#### Активация

Для создания пены необходимо добавить в текстурный слот материала воды диффузную текстуру, в каждом из RGB - каналов которой следует расположить черно-белые текстуры пены. Для полученной текстуры необходимо активировать панель **Water Foam**.



## Настройка текстур

**Influence > Color** Фактор влияния цвета текстуры. Значение по умолчанию 1.0.

**Water Foam > UV Frequency** Частота колебаний анимированной развертки. Значение по умолчанию (1.0, 1.0).

**Water Foam > UV Magnitude** Амплитуда колебаний анимированной развертки. Значение по умолчанию (1.0, 1.0).

## Настройка материала

**Foam > Factor** Фактор общего влияния пены. Значение по умолчанию 0.5.

### 19.1.10 Каустика и хроматическая аберрация

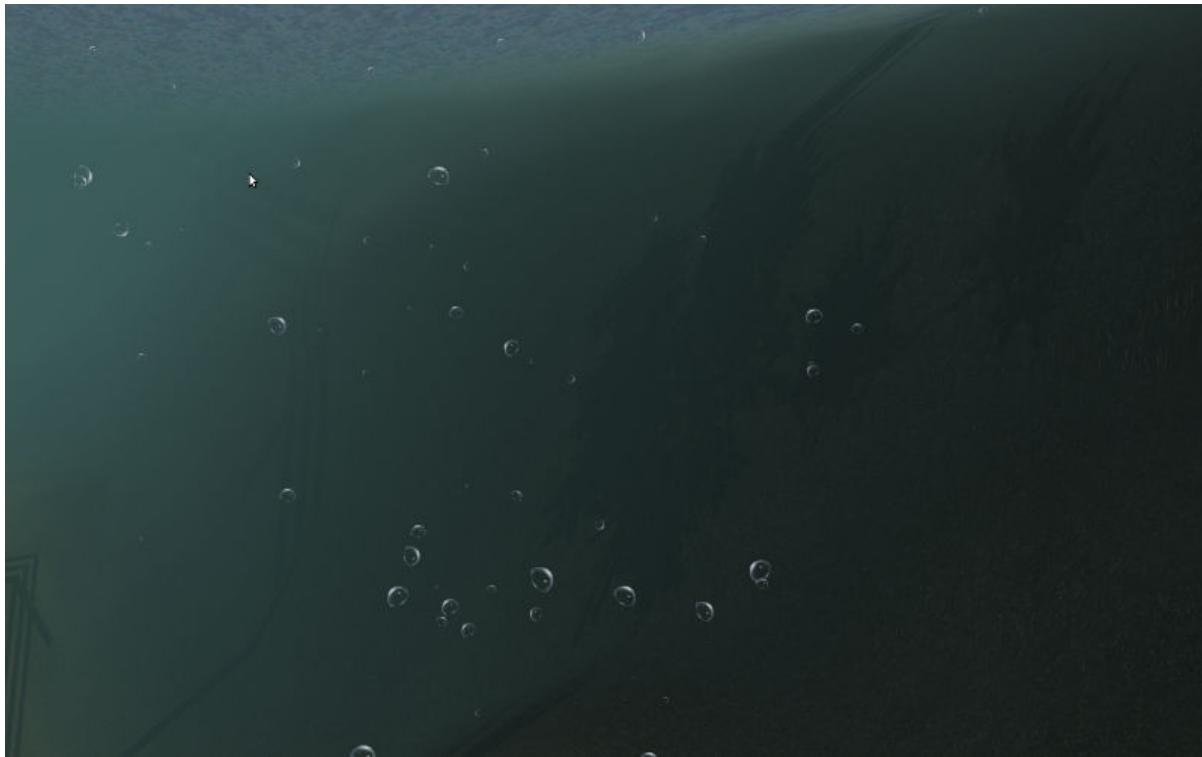
Для создания каустики на подводных объектах следует включить опцию **Caustics**. Так же на сцене должен присутствовать источник освещение типа **Sun**.



**Scale** Размер ячеек процедурной текстуры. Значение по умолчанию 0.25.

**Brightness** Фактор влияния каустики. Значение по умолчанию 0.5.

### 19.1.11 Подводная среда



#### Настройки видимости (“туман”)

***Underwater Fog > Color*** Цвет тумана. Значение по умолчанию (0.4, 0.6, 0.7).

***Underwater Fog > Density*** Экспоненциальный фактор, влияющий на плотность и расстояние. Значение по умолчанию 0.06.

Применяются также настройки [сумеречных лучей](#).

---

**Примечание:** Для корректного отображения водной поверхности необходимо выключить опцию **Rendering Options > Backface Culling**.

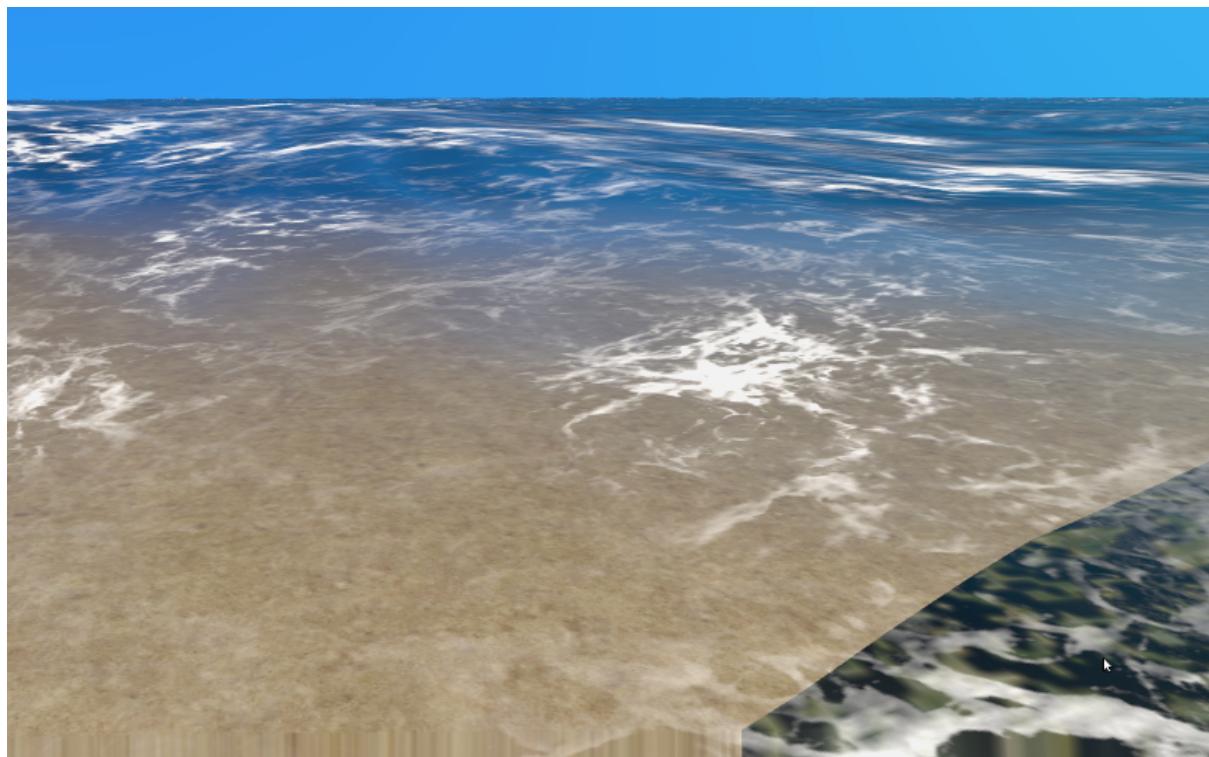
---



### 19.1.12 Объемные волны

#### Активация

Для создания объемных волн следует включить опцию Waves.



## Настройка

**Wave Height** Высота волн. Значение по умолчанию 0.0.

**Wave Length** Длина волн. Значение по умолчанию 10.0.

**Noise Dist Scale 0** Размер первого компонента волн, удаленных от берега.

**Noise Dist Scale 1** Размер второго компонента волн, удаленных от берега.

**Noise Dist Freq 0** Частота первого компонента волн, удаленных от берега.

**Noise Dist Freq 1** Частота второго компонента волн, удаленных от берега.

**Min Dir Shore Fac** Минимальный коэффициент уменьшения высоты прибрежных волн.

**Dir Frequency** Частота накатывания прибрежных волн.

**Noise Dir Scale** Размер шума на прибрежных волнах.

**Noise Dir Freq** Частота шума на прибрежных волнах.

**Min Dir Noise Fac** Минимальное значение шума на прибрежных волнах.

**Min Dist Fac** Минимальный коэффициент примешивания волн, удаленных от берега.

**Horizontal Factor** Коэффициент смещения прибрежных волн в направлении к берегу.

### 19.1.13 Настройки генерируемой поверхности

**Generate Mesh** Включить генерируемую поверхность.

**Cascades Number** Количество каскадов в генерируемой поверхности.

**Subdivisions** Количество подразделений в генерируемой поверхности.

**Detailed Distance** Максимальное расстояние от камеры до края последнего каскада.

### Создание текстуры с параметрами береговой линии

На панели инструментов (горячая клавиша “T”) во вкладке Blend4Web открыть панель **Bake Shore Distance Map**. Выставить настройки максимального расстояния до берега **Maximum Distance** и размера получаемой текстуры **Texture Size**. Выбрать сначала объект (или несколько объектов) ландшафта, затем объект воды. Нажать кнопку **Bake**.

В зависимости от размера текстуры и количества вершин в обрабатываемых мешах время выполнения скрипта варьируется от долей секунды до нескольких минут. Убедиться, что в меше воды создана текстура с названием **ShoreDistance**.

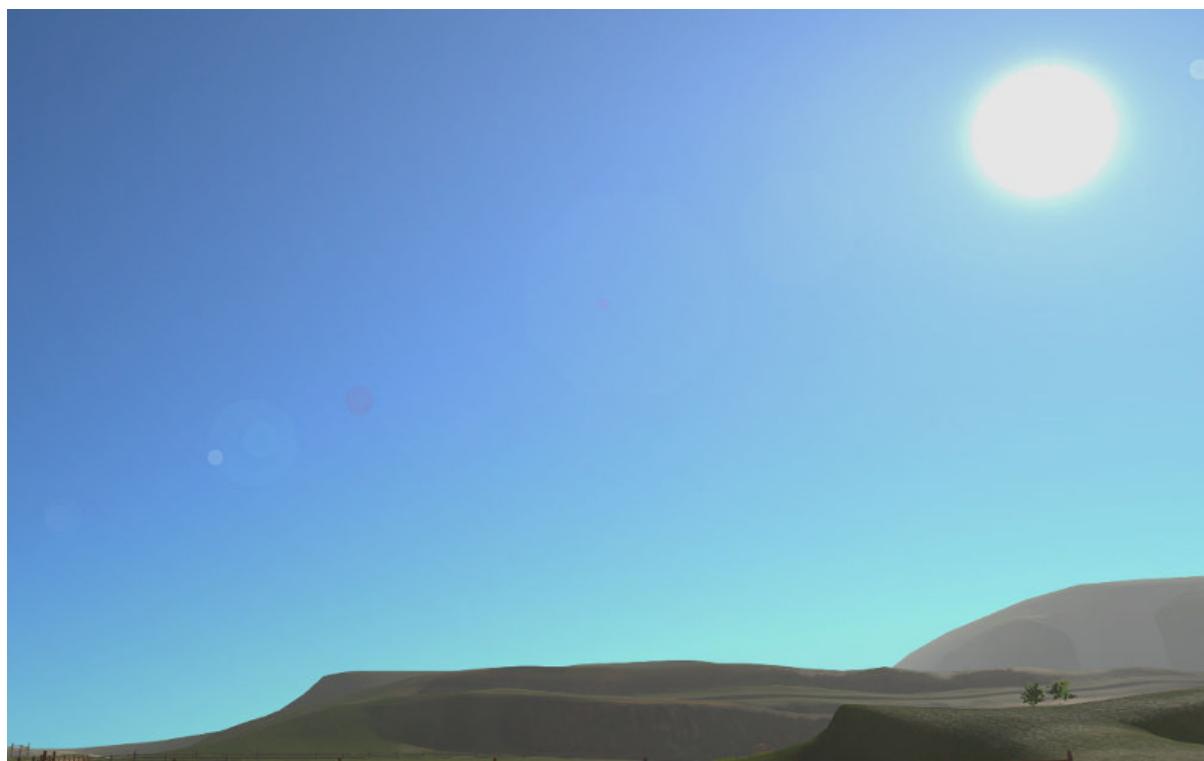
При вызове скрипта в материале воды сохраняются некоторые системные свойства. Поэтому, после его работы обязательно нужно сохранять сцену.

## 19.2 Атмосфера

### 19.2.1 Рассеивание

Во вкладке **World** активировать панель **Procedural Sky**, предварительно выставив опцию **World > Render Sky**. Если одновременно с этим используется статическая *текстура неба*, она будет заменена.

**Примечание:** Кроме того, процедурная текстура неба может быть использована для имитации рассеянного *освещения от окружающей среды*, по аналогии со статической *текстурой неба*. Для этого необходимо выставить опции **Procedural Sky > Use as Environment Lighting** и **Environment Lighting > Sky Texture**. Если текстура мира для рассеянного освещения уже существует, она будет заменена.



Движком поддерживаются следующие настройки:

**Procedural Sky > Sky Color** Базовый цвет неба. Значение по умолчанию (0.087, 0.255, 0.6) (голубой).

**Procedural Sky > Rayleigh Brightness** Яркость рэлеевского рассеяния (на малых частицах). Значение по умолчанию 3.3.

**Procedural Sky > Mie Brightness** Яркость рассеяния Ми (на крупных частицах). Значение по умолчанию 0.1.

**Procedural Sky > Spot Brightness** Яркость пятна солнца. Значение по умолчанию 20.0.

**Procedural Sky > Scatter Strength** Фактор рассеяния света. Значение по умолчанию 0.2.

**Procedural Sky > Rayleigh Strength** Фактор рэлеевского рассеяния. Значение по умолчанию 0.2.

**Procedural Sky > Mie Strength** Фактор рассеяния Ми. Значение по умолчанию 0.006.

**Procedural Sky > Rayleigh Collection Power** Степенной коэффициент рэлеевского рассеяния. Значение по умолчанию 0.35.

**Procedural Sky > Mie Collection Power** Степенной коэффициент рассеяния Ми. Значение по умолчанию 0.5.

**Procedural Sky > Mie Distribution** Распределение рассеяния Ми. Значение по умолчанию 0.4.

## 19.2.2 Туман

Движком поддерживаются стандартные параметры вкладки **World > Mist**. Помимо этого возможно переопределение цвета тумана.

**Mist > Use custom color** Использовать для тумана заданный цвет, вместо цвета **World > Horizon**. По умолчанию включено.

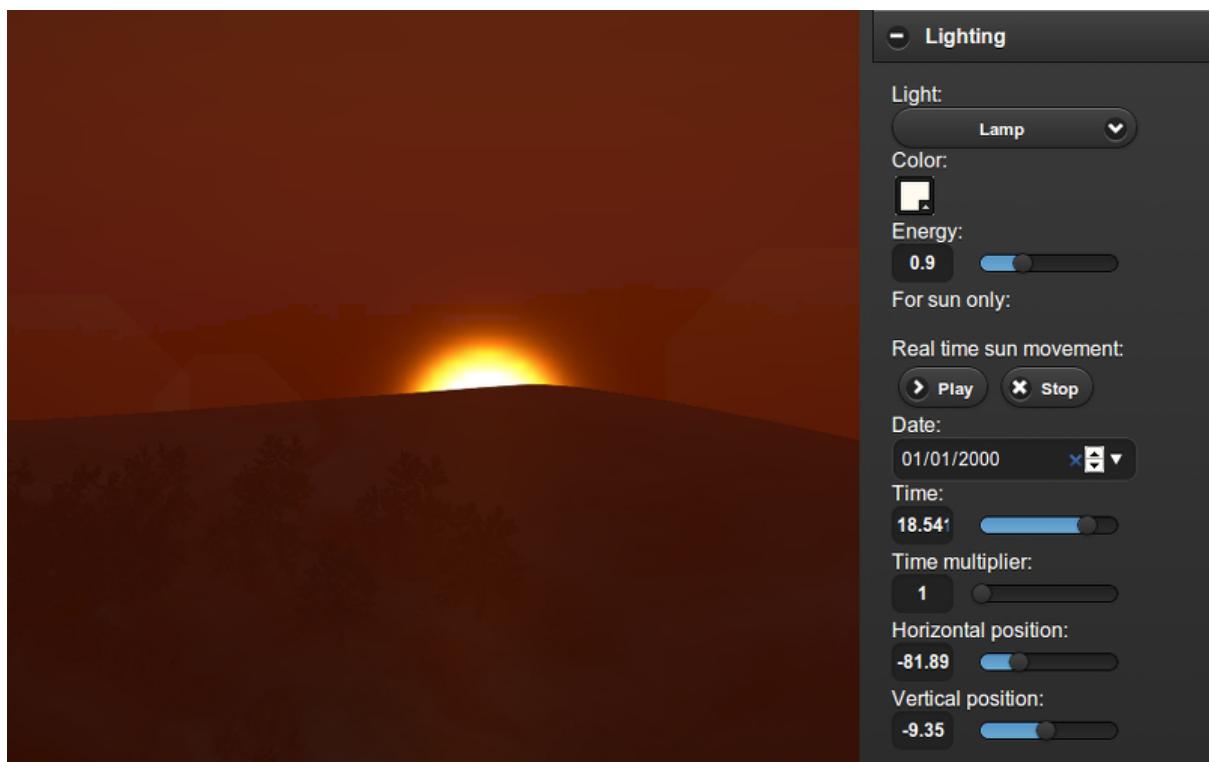
**Mist > Color** Цвет тумана. Значение по умолчанию (0.5, 0.5, 0.5) (серый).

При использовании динамического неба цвет тумана определяется цветом неба.

## 19.2.3 Время суток

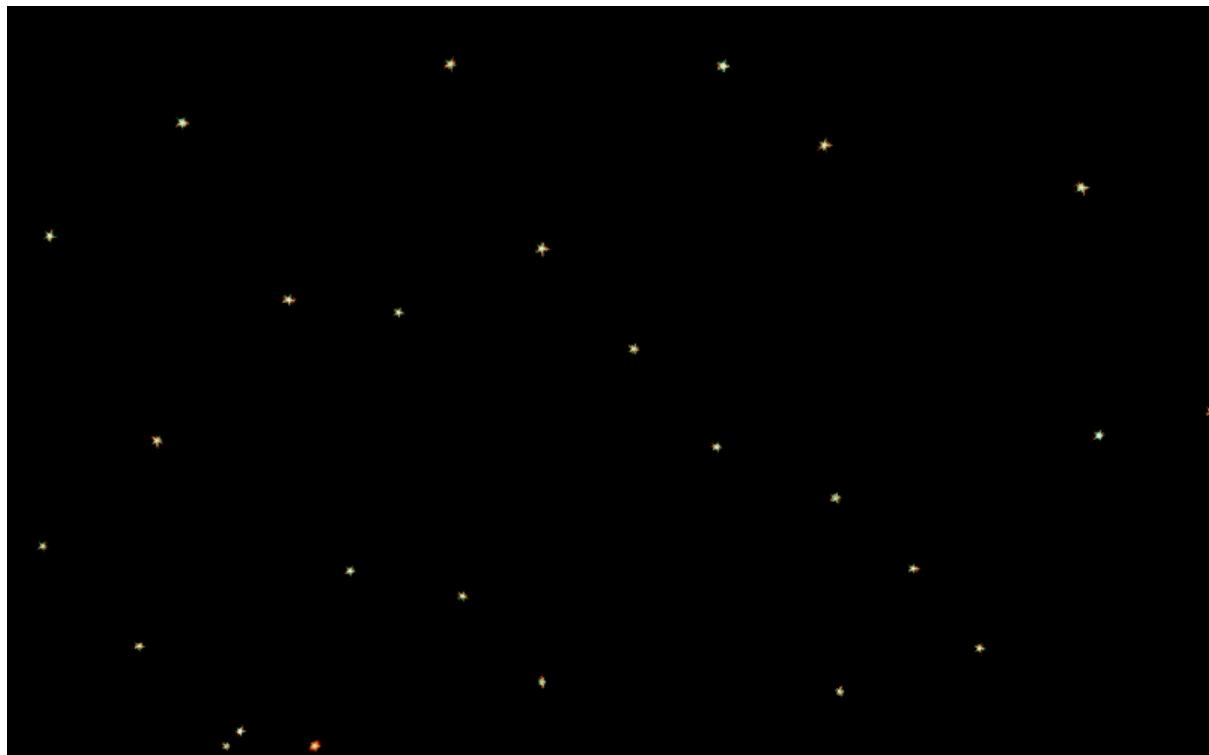
Для лампы необходимо выставить опцию **Dynamic Intensity**.

Время суток устанавливается приложениями с использованием соответствующего API. В частности, время суток может устанавливаться в интерфейсе **Lighting просмотрщика сцен**.



#### 19.2.4 Звезды

Настройка данного эффекта описывается в разделе *Материалы гало (Halo)*.



## 19.3 Ветер

**Сила и направление ветра оказывает воздействие на:**

- анимацию травы и крон деревьев
- динамику систем частиц
- частоту колебаний волн воды (в настоящий момент влияет только сила)

### 19.3.1 Активация

Добавить на сцену объект - силовое поле типа Wind.

### 19.3.2 Настройка

**Направление** Направление задается посредством вращения объекта - силового поля.

**Force Fields > Strength** Сила ветра. Располагается во вкладке Physics. Значение по умолчанию 1.0.

### 19.3.3 Анимация травы и крон деревьев

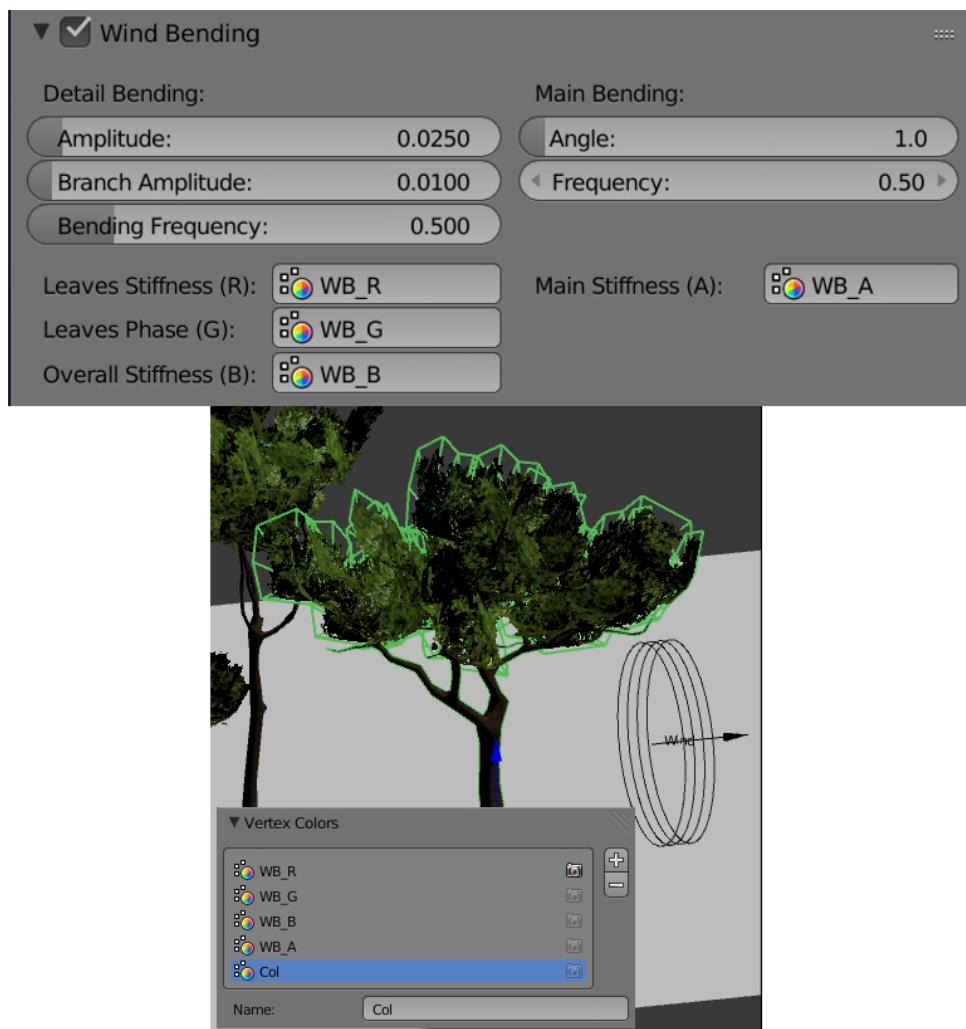
Подготовка ресурсов для рендеринга травы описана в разделе *Травяной покров*.

#### Активация

На объекте травы или дерева активировать панель Wind Bending.

#### Настройка

Интерфейс для настроек появляется после активации панели Wind Bending.



**Main bending > Angle** Амплитуда угла “основного” отклонения под действием ветра (в градусах). Значение по умолчанию 10.0.

**Main bending > Frequency** Частота “основного” отклонения под действием ветра. Значение по умолчанию 0.25.

**Main bending > Main Stiffness (A)** Текстовое поле для названия слоя вертекского цвета, содержащего информацию о жесткости “основного” отклонения. Может быть оставлено пустым.

**Detail bending > Amplitude** Амплитуда угла “детализованного” отклонения под действием ветра (в градусах). Значение по умолчанию 0.1.

**Detail bending > Branch Amplitude** Амплитуда угла отклонения ветвей под действием ветра (в градусах). Значение по умолчанию 0.3.

**Detail bending > Bending Frequency** Частота колебаний ветвей. Значение по умолчанию 1.0.

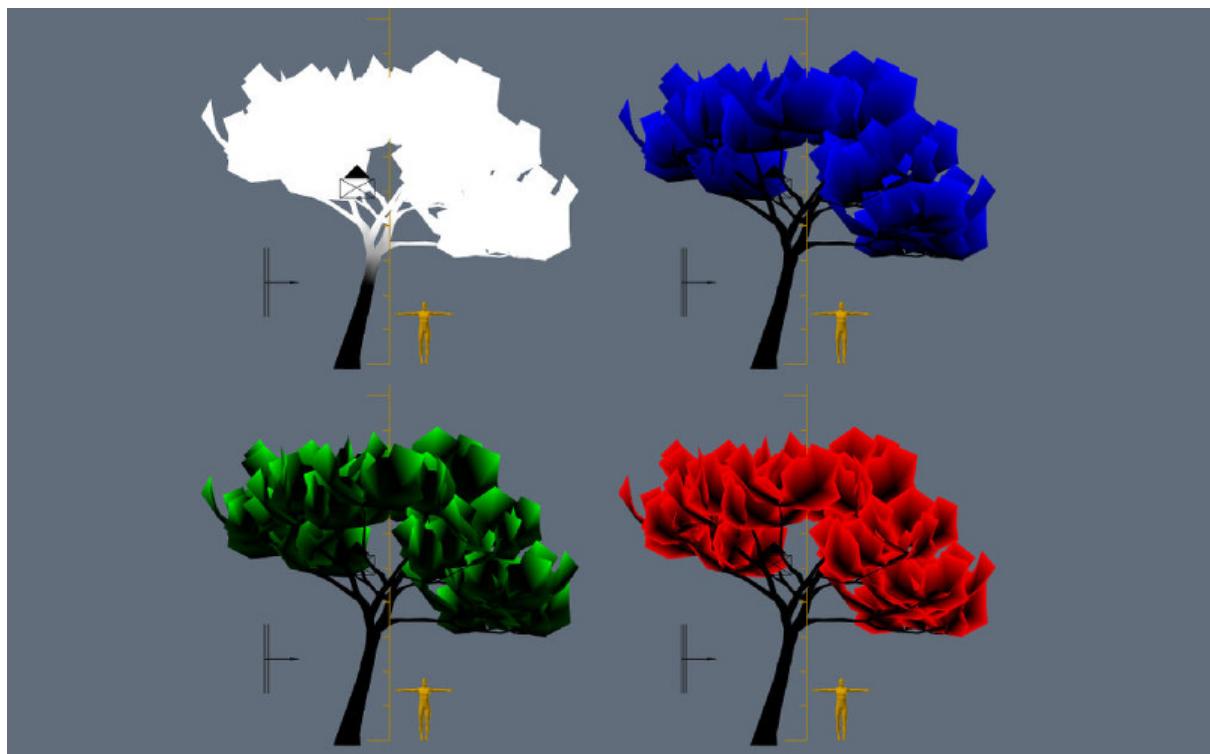
**Detail bending > Leaves Stiffness (R)** Текстовое поле для названия слоя вер-

тексного цвета, содержащего информацию о жесткости листвы. Может быть оставлено пустым.

***Detail bending > Leaves Phase (G)*** Текстовое поле для названия слоя вертексного цвета, содержащего информацию о фазе отклонения листвы. Может быть оставлено пустым.

***Detail bending > Overall Stiffness (B)*** Текстовое поле для названия слоя вертексного цвета, содержащего информацию об общей жесткости листвы. Может быть оставлено пустым.

Слои вертексных цветов с указанными в настройках названиями должны существовать в меше.



---

## Гамма-коррекция и альфа-композитинг

---

### 20.1 Общее описание

Сущность гамма-коррекции заключается в упаковке яркости канала изображения в 8 битах информации.

Стандартные (не HDR) изображения всегда сохраняются в нелинейном пространстве, где тёмные компоненты кодируются большим числом битов чем светлые. Это означает, что значению 0.5 от реальной интенсивности света (физической величины, называемой освещённость) будет соответствовать большее значение, содержащееся в каналах RGB (в самом простом случае  $0.5^{1/2.2} = 0.73$ ).

В противном случае 8 бит информации не достаточно для кодирования интенсивности света, что приведёт к тому, что тёмные тона будут отображаться некорректно (например, тёмный градиент будет иметь ступенчатый характер переходов между соседними цветами).

Таким образом, веб-браузеры, как и многие другие программы для просмотра и обработки изображений, работают в нелинейном цветовом пространстве. Однако трёхмерные движки и рендереры работают в линейном пространстве, поскольку только оно может адекватно представлять поведение света в реальном мире. Например, освещённость от двух одинаковых ламп будет ровно в два раза превышать освещённость только от одной.

Безусловно, в таком случае 8 бит информации не будет достаточно, что наглядно иллюстрирует таблица, в которой приведены примерные значения величин освещённости от реальных источников.

Описание	Освещённость, лк
Летом в полдень	17 000
Зимой в полдень	5 000
В пасмурный день	1 000
В светлой комнате	100
Ночью в полнолуние	0.2
В безлунную ночь	0.001

Blender при настройке сцены **Color Management > Display Device > sRGB** работает в линейном пространстве. Значения цветов материалов и настройки источников света соответствует физическим величинам. При работе с текстурами, за исключением карт нормалей необходимо выставить настройку изображения **Image > Input Color Space > sRGB**. В этом случае при рендеринге будет производится автоматическая распаковка изображения: **sRGB->Linear**.

## 20.2 Человеческое зрение и мониторы

Человеческое восприятие света нелинейно (человек лучше различает градации тусклого света чем яркого), однако свет, поступающий в глаз, по-прежнему должен подчиняться физическим законам (см. пример с лампочками).

Мониторы с электронно-лучевыми трубками (ЭЛТ) имеют нелинейную характеристику яркости от приложенного к их входу электрического напряжения, которое, в свою очередь, определяется значением канала цветности в видеопамяти. Подобную же характеристику копируют жидкокристаллические мониторы. Тем не менее свет, излучаемый такими мониторами, по-прежнему должен подчиняться физическим законам. Например, в идеальном случае при добавлении второго источника света на сцену в виртуальном мире, яркость пикселей на экране монитора должна увеличиваться в два раза.

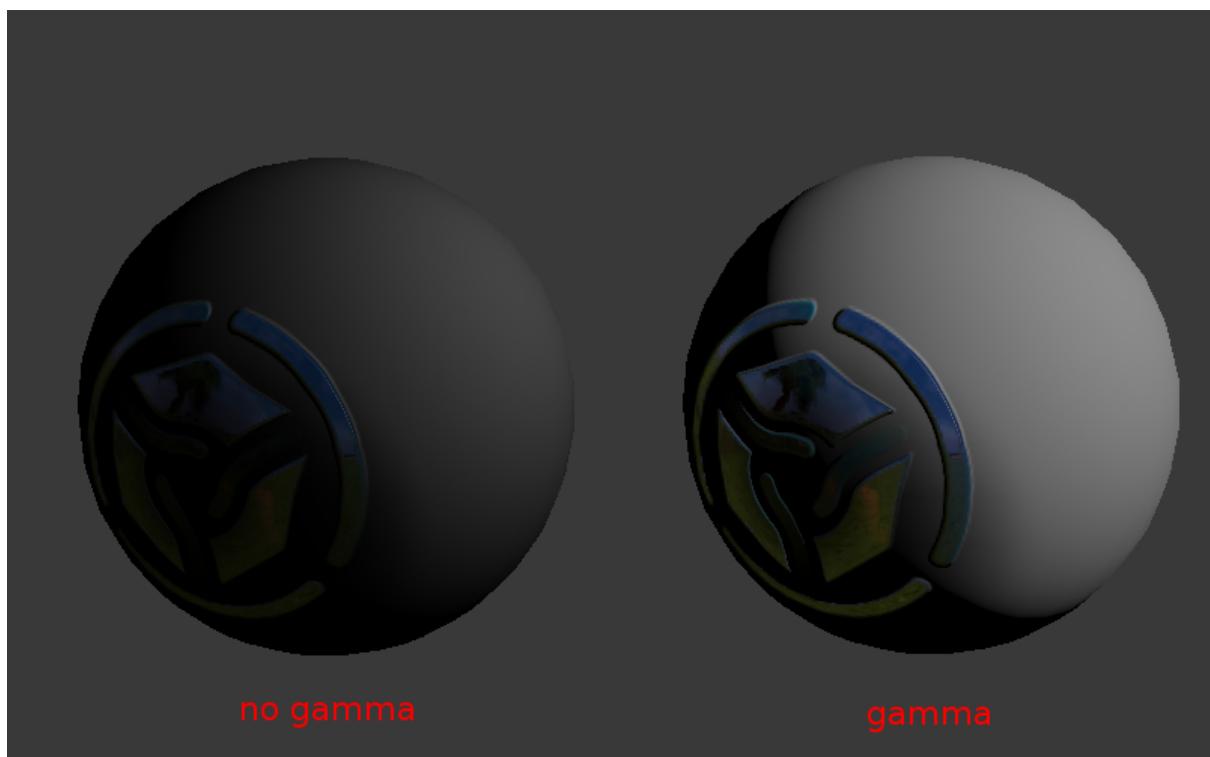
Таким образом, особенности восприятия человеческого глаза являются причиной, по которой стало возможным осуществлять упаковку каналов цветности, в то время как технические характеристики мониторов имеют вторичное значение по отношению к гамма-коррекции.

## 20.3 Гамма

Используется в следующей упрощенной формуле:

$$V_{\text{out}} = V_{\text{in}}^{\gamma}$$

$\gamma < 1$  - упаковывающая гамма,  $\gamma > 1$  - распаковывающая гамма. В наиболее простом случае используются значения 1/2.2 и 2.2 соответственно. Далее вместо термина “гамма-коррекция” будут использованы термины “упаковка” (**Linear -> sRGB**) и “распаковка” (**sRGB -> Linear**).



## 20.4 Коррекция в нодовых материалах

### 20.4.1 Ноды для окраски

При использовании текстур и вертексных цветов для окраски (не в качестве масок), необходима распаковка (sRGB -> Linear). Выход *Vertex Color* ноды *Geometry* и нода текстуры реализуют распаковку автоматически.

Отметим, что альфа-канал ноды текстуры коррекции не подвергается, его значения находятся в линейном пространстве.

### 20.4.2 Ноды для масок

Текстуры и вертексные цвета могут использоваться в качестве масок, т.е. для смешения цветов или других математических операций. В таком случае необходима упаковка (Linear -> sRGB).

Следует обратить внимание, что выход *Vertex Color* ноды *Geometry* и нода текстуры реализуют распаковку автоматически. Это приводит к необходимости дополнительного преобразования обратно в нелинейное пространство, для чего используется специальная нода *LINEAR\_TO\_SRGB* или нода *GAMMA* со значением  $\gamma = 1/2.2$ .

---

**Примечание:** Нода *LINEAR\_TO\_SRGB* считается устаревшей, следует использовать ноду *GAMMA*.

---

### 20.4.3 Карты нормалей

При использовании карт нормалей никакие преобразования не производятся.

### 20.4.4 Сводная таблица коррекции в нодовых материалах

Случай использования	Коррекция
Текстура для окраски	реализуется автоматически в node текстуры (альфа-канал коррекции не подвергается)
Текстура для маски	GAMMA со значением $\gamma = 0.45$
Вертексный цвет для окраски	реализуется автоматически
Вертексный цвет для маски	GAMMA со значением $\gamma = 0.45$
Карта нормалей	не требуется

## 20.5 Альфа-композитинг

### 20.5.1 Общие сведения

Физически корректный альфа-композитинг осуществляется по формуле [источник]:

$$C_o = C_a \alpha_a + C_b \alpha_b (1 - \alpha_a).$$

Формула отличается от классической операции смешивания (mix, выпуклая комбинация) наличием множителя  $\alpha_b$  во втором слагаемом. Таким образом, для осуществления альфа-композитинга должно быть известно не только значение  $\alpha_a$  пикселя-источника, но и значение  $\alpha_b$  пикселя, поверх которого осуществляется рендеринг.

В случае предварительного умножения значений  $\alpha$  на цветовые каналы (т.н. premultiplied alpha) формула принимает вид:

$$C_o = C_a + C_b (1 - \alpha_a).$$

Последняя формула используется также для расчёта результирующего значения  $\alpha_o$ :

$$\alpha_o = \alpha_a + \alpha_b (1 - \alpha_a).$$

Предварительное умножение цветовых каналов на значения  $\alpha$  позволяет сэкономить две операции умножения. Еще более существенным является тот факт, что полученная формула может использоваться многократно, без необходимости деления цвета  $C_o$  на значение  $\alpha_o$  на каждой последующей итерации.

### 20.5.2 Реализация

Функция смешивания в движке Blend4Web имеет вид:

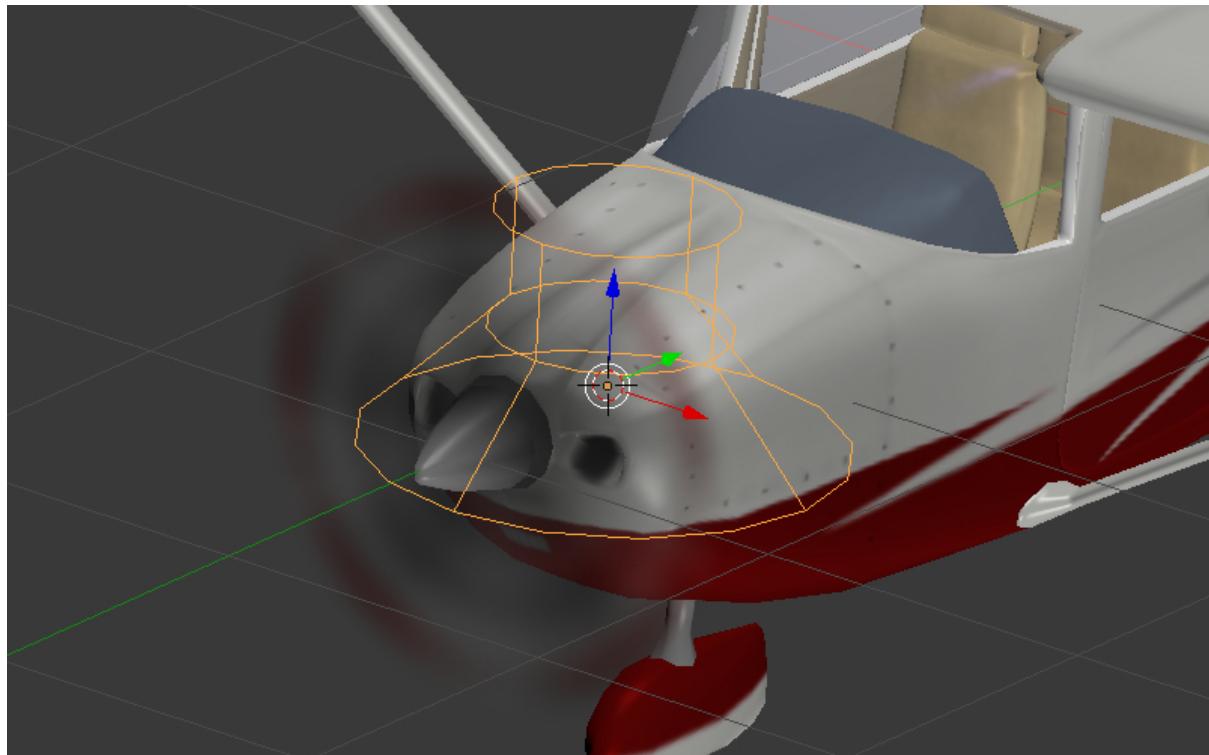
```
gl.blendFunc(gl.ONE, gl.ONE_MINUS_SRC_ALPHA);
```

Инициализация контекста WebGL производится с параметром *premultipliedAlpha = true* (что является значением по умолчанию). Кроме того, на выходе шейдеров производится умножение всех каналов цветности на значение  $\alpha$ .

## Звуковая подсистема

---

Создание звуковых источников осуществляется в Blender'е. Используется стандартный объект **Speaker**.



### 21.1 Настройка звуковых источников

Настройки спикера выставляются в панели **Properties** на вкладке **Object Data**.

Движком поддерживаются все стандартные для Blender параметры звука и некоторые специфические для движка опции.

*Speaker behavior:*

Поведение звукового источника.

**Positional** — высококачественный звук, допускающий позиционирование и имеющий направленность (конусность). Для рендеринга используется Web Audio API. Воспроизведение подобных звуков обладает наименьшей производительностью, поэтому их использовать целесообразно только для коротких сэмплов.

**Background Sound** — высококачественный всенаправленный звук без возможности позиционирования в пространстве. Для рендеринга используется Web Audio API. Более производителен, однако нецелесообразен для музыки.

**Background Music** — используется для воспроизведения музыки. Максимальная производительность вследствие использования тега `Audio`, минимальная гибкость.

На панели `Sound` доступны следующие настройки:

*Volume*

Громкость источника.

*Random Volume*

Дополнительная рандомизация громкости. Результирующее значение определяется аналогично задержке.

*Pitch*

Скорость проигрывания (высота) звука.

*Random Pitch*

Дополнительная рандомизация скорости проигрывания звука. Результирующее значение определяется аналогично задержке.

*Fade-In*

Интервал плавного включения звука.

*Fade-Out*

Интервал плавного выключения звука.

*Disable doppler*

Игнорировать смещение частоты источника при его перемещении.

*Cyclic play*

Зацикливать воспроизведение звука.

*Delay*

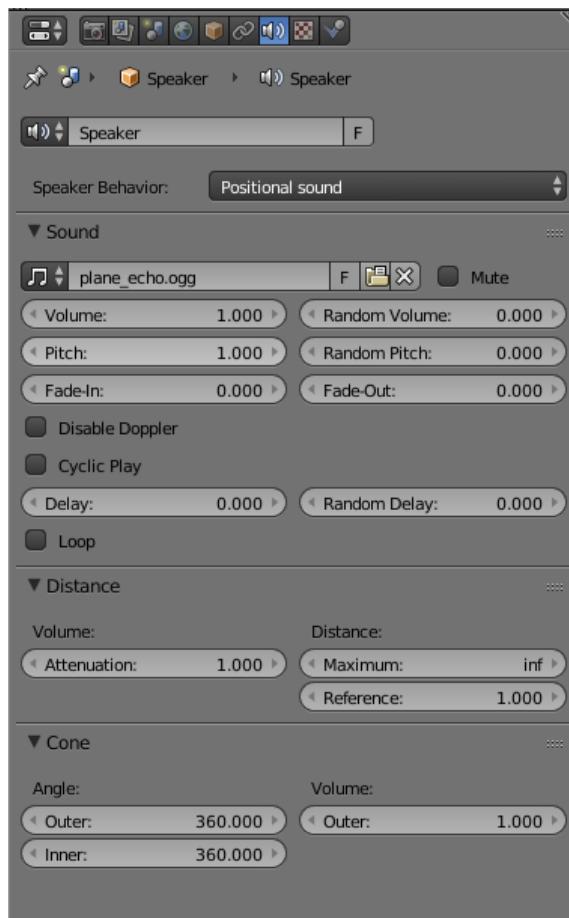
Задержка в секундах перед началом проигрывания звука.

*Random Delay*

Дополнительная рандомизация задержки, результирующее значение определяется по формуле  $Delay_{result} = Delay + Delay_{random} * Random_{[0-1]}$ .

### Loop

Зацокливать воспроизведение звука. Отличается от Cyclic Play, тем, что способен обеспечить нулевую задержку при повторении. Опция доступна только для звуковых источников с поведением Positional или Background Sound.



## 21.2 Обработка и кодирование

### 21.2.1 Поддерживаемые форматы (контейнеры):

- ogg, кодек Vorbis (Chrome, Firefox)
- mp3 (Chrome, Safari)
- mp4, кодек AAC (Chrome, Safari)

Рекомендуется использовать *Ogg*, который является открытым стандартом, поддерживается многими браузерами, обеспечивает хорошее качество звука. Оптимальным с точки зрения качества и совместимости является формат 48кГц/16бит. Одноканальный звук (моно) используется для хранения коротких сэмплов, двухканальный звук (стерео) - для музыкального сопровождения.

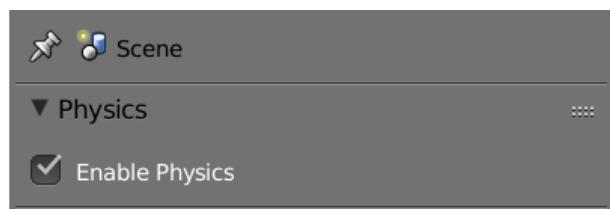
Конвертация ресурсов в различные форматы описывается в *соответствующем разделе*.

### Физика

---

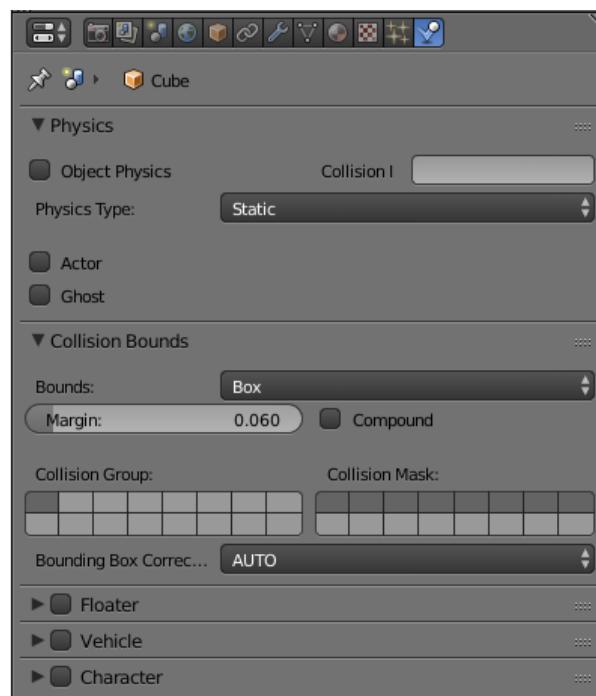
## 22.1 Подготовка к использованию

Для задействования физики на сцене необходимо установить флаг `Enable Physics` в панели `Physics` на сцене.

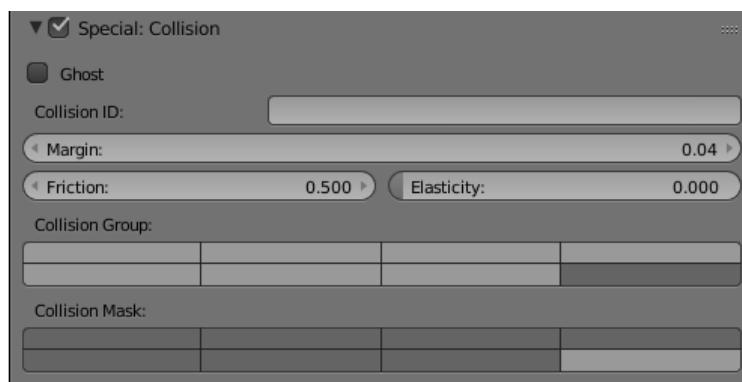


## 22.2 Статический тип физики

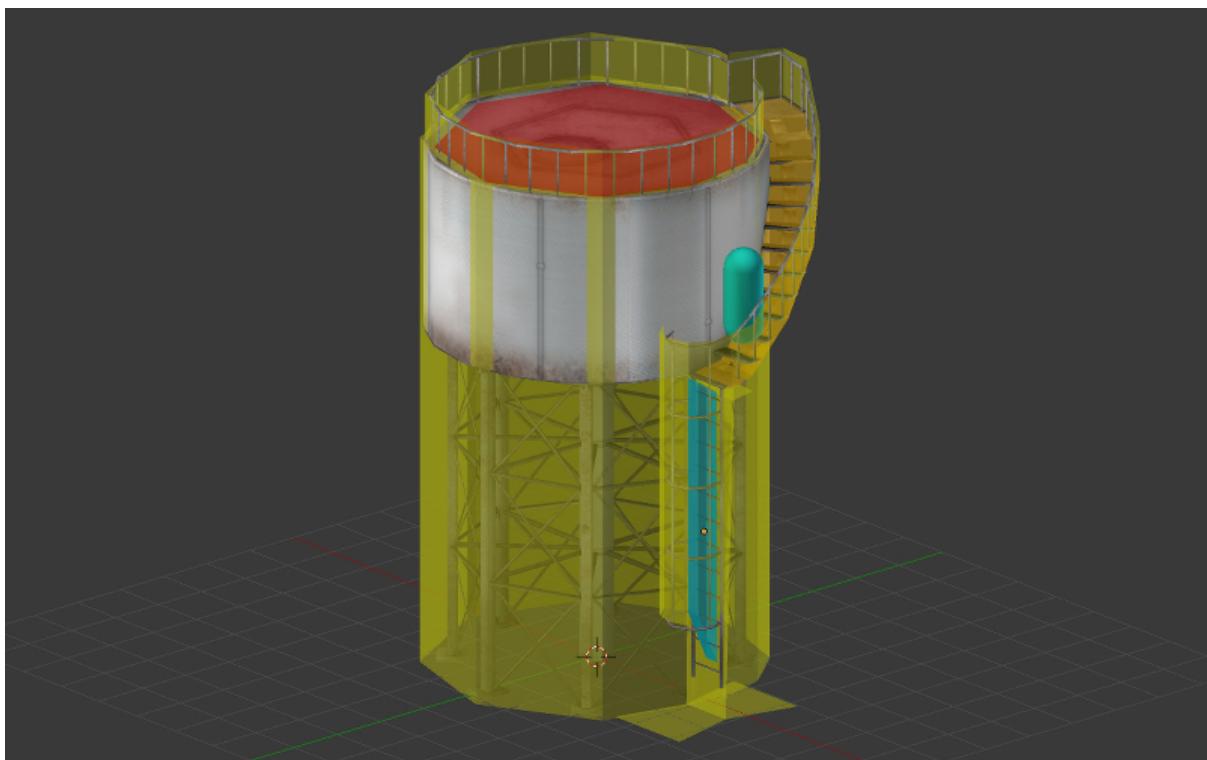
Может использоваться как ограничитель движения других объектов, например, для определения столкновений с ландшафтом, стенами и т.д. В настройках физики такого объекта для опции `Physics Type` должно быть выбрано значение `Static` (значение по умолчанию).



Меш может быть покрыт одним или несколькими физическими материалами. Во вкладке **Material** должна быть активирована панель **Special: Collision**.



Опция **Ghost** исключает материал из физических взаимодействий, но сообщает приложению о контакте с ним. Пример - определение, что игровой персонаж находится на вертикальной лестнице.



Поле **Collision ID** предназначено для определения столкновения со специфическим материалом, и может быть оставлено пустым. Пример использования **Collision ID** - определение нахождения игрового персонажа на разных типах покрытия ландшафта - трава, песок, деревянное покрытие и т.д.

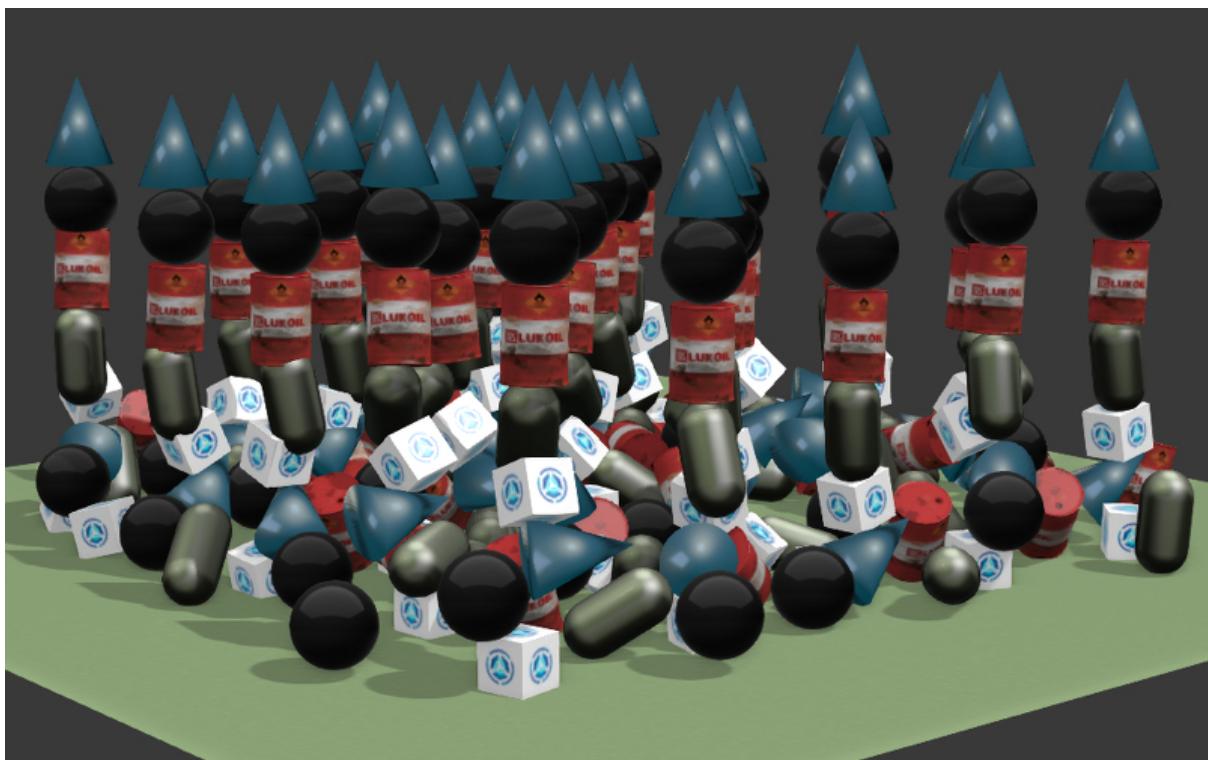
Поле **Margin** позволяет настроить ширину зоны, в которой происходит соударения с сеткой данного объекта. Данная настройка позволяет улучшить стабильность симуляций столкновений объектов.

Также в этой панели располагаются физические настройки материала. Поддерживаются следующие настройки: трение (**Friction**), упругость (**Elasticity**).

Поле **Collision Group** отвечает за физическую группу, к которой относится материал. Поле **Collision Mask** определяет все физические группы, с которыми будет взаимодействовать данный материал.

## 22.3 Динамический тип физики

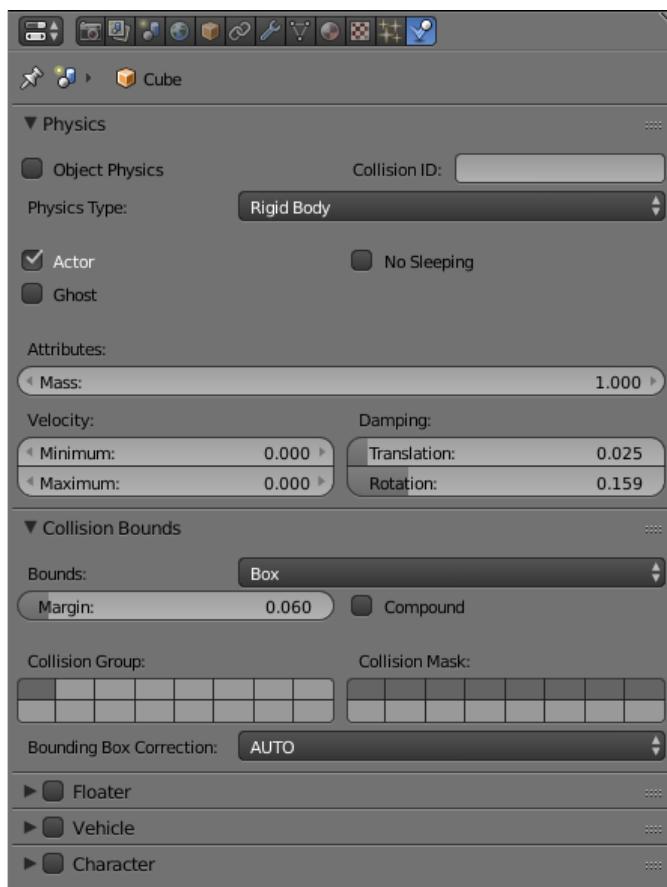
Предназначен для симуляции движения жесткого тела.



В настройках панели **Physics** такого объекта должен быть установлен флаг **Object Physics**. Поле **Collision ID** предназначено для определения столкновения со специфическим объектом (например, прикрепленный к камере объект для определения близости FPS персонажа к предметам), и может быть оставлено пустым.

Опция **Physics Type** может иметь значения: **Rigid Body** (с вращениями) или **Dynamic** (без вращений). В настройках **Collision Bounds** может быть выбран тип коллайдера. Поддерживаются: **Box**, **Capsule**, **Sphere**, **Cylinder**, **Cone**. Так же можно настроить: массу (**Mass**), демпфирование (**Damping**) - для перемещения (**Translation**) и вращения (**Rotation**).

Поле **Collision Group** отвечает за физическую группу, к которой относится объект. Поле **Collision Mask** определяет все физические группы, с которыми будет взаимодействовать данный объект.

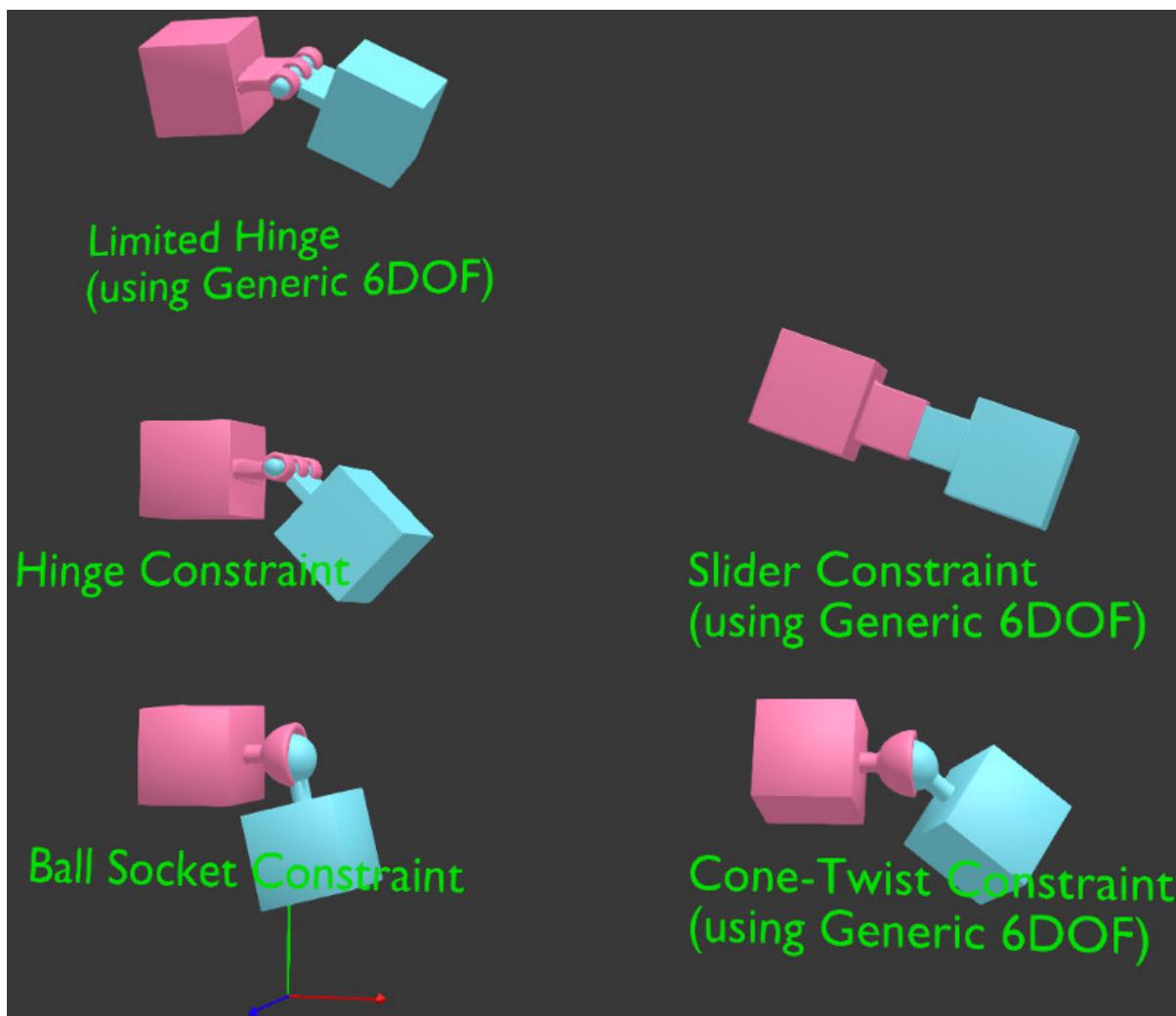


Для материала такого объекта поддерживаются: трение (**Friction**), упругость (**Elasticity**). В случае использования на одном меше нескольких материалов физические настройки считаются с первого из них.

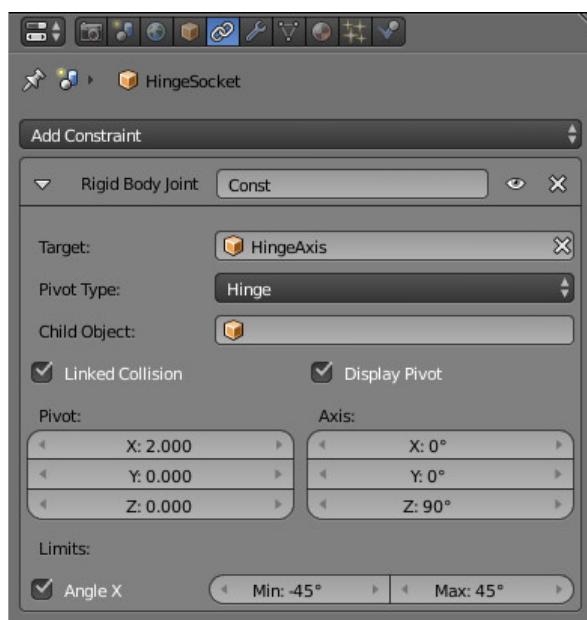
Для объекта-камеры должна использоваться настройка **Physics Type = Dynamic**, должен быть установлен флаг **Object Physics**.

## 22.4 Ограничители (Constraints)

Физические ограничители используются для уменьшения числа степеней свободы объектов.

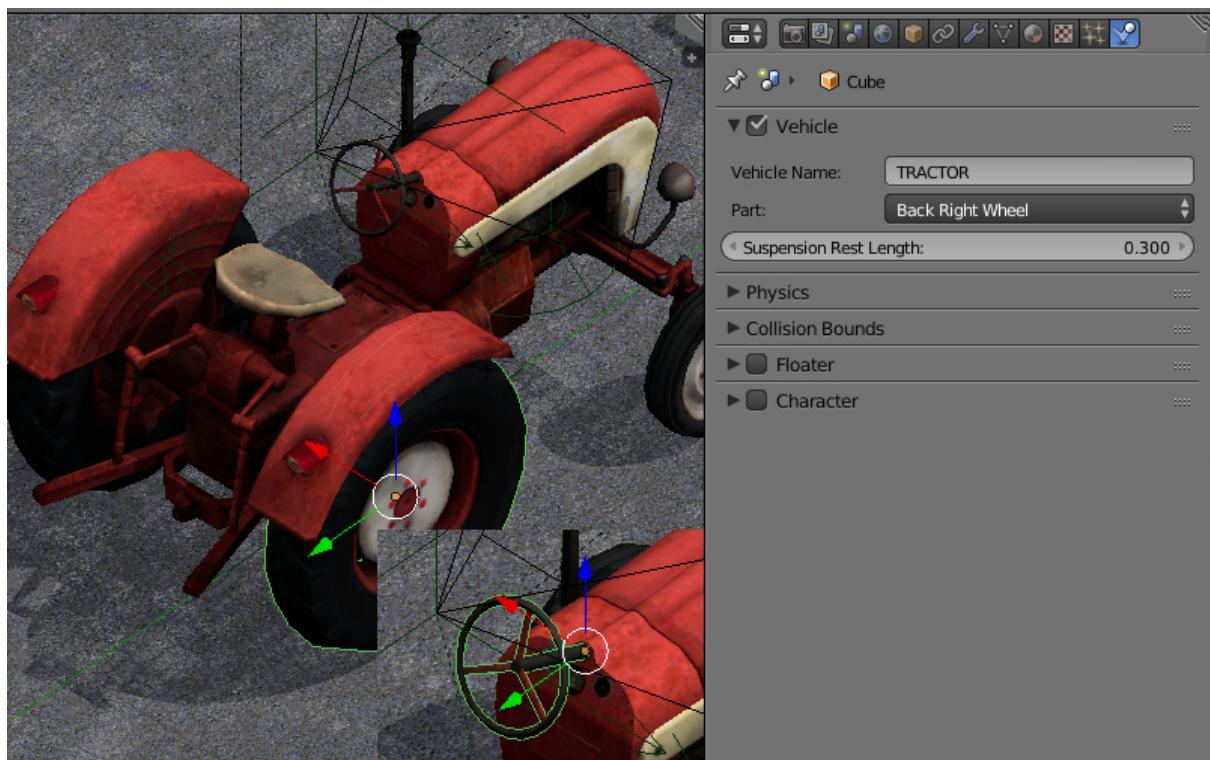


Установка физического ограничителя (Rigid Body Joint) на объект происходит в панели **Object Constraints**. Поддерживаемые типы (Pivot Type): **Ball**, **Hinge**, **Cone Twist**, **Generic 6 DoF**. Физический ограничитель можно установить на один из двух взаимодействующих объектов, при этом другой выступает в качестве цели (Target). Оба объекта могут быть со статическим и/или динамическим типом физики. В ограничителях (кроме **Ball**) могут настраиваться пределы перемещения и вращения.



## 22.5 Колесные транспортные средства

Модель транспортного средства (ТС) должна состоять из 6 отдельных объектов - шасси, 4 колеса, рулевое колесо. Центр меша шасси должен соответствовать центру масс. Центры мешей колес и рулевого колеса должны располагаться на осях вращения. Рулевое колесо должно быть ориентировано в локальной системе координат: X - ось вращения, Y - вправо, Z - вверх. Объекты могут иметь любые названия.



На всех 6 объектах нужно выставить **Part**, указать один и тот же идентификатор в поле **Vehicle Name**, выбрать соответствующий тип объекта - **Chassis**, **Steering Wheel**, **Back Right Wheel** и т.д. Для колес имеется также настройка компенсирующего хода подвески **Suspension Rest Length**.

Для шасси необходимо указать реалистичную массу (т.к. значение по умолчанию 1 кг). Для этого перейти в настройки физики, для опции **Physics Type** выбрать значение **Rigid Body**, и выставить нужное значение (например, 1000 кг) в поле **Mass**.

### 22.5.1 Параметры настройки для шасси

**Force Max** Максимальная движущая сила транспортного средства.

**Brake Max** Максимальный коэффициент торможения.

**Suspension Compression** Коэффициент демпфирования при растяжении подвески.

**Suspension Stiffness** Коэффициент жесткости подвески.

**Suspension Damping** Коэффициент амортизации подвески.

**Wheel Friction** Константа трения колес о поверхность. Для реалистичных Т.С. должен быть в районе 0.8. Но может быть значительно увеличен, для улучшения управляемости (1000 и более)

**Roll Influence** Снижает вращающий момент от колес, уменьшая вероятность переворота транспортного средства (0 - нет вращающего момента, 1 - реальное физическое поведение).

**Max Suspension Travel Cm** Максимальный ход подвески в сантиметрах.

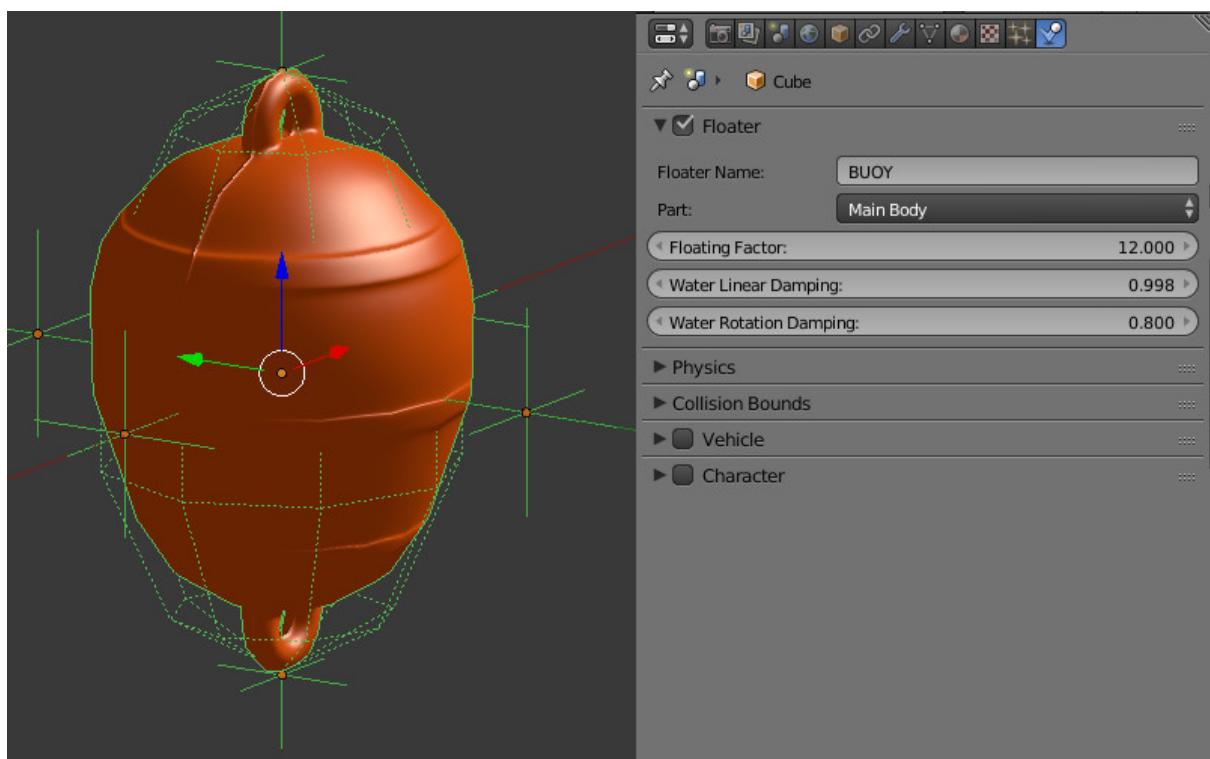
Для рулевого колеса(Steering Wheel) необходимо указать максимальный угол поворота(Steering Max) и передаточное отношение угла поворота руля к передним колесам (Steering Ratio). Максимальное значение угла поворота указывается в оборотах. Один оборот равен 360 градусам. Таким образом, поставив Steering Max равным единице, а Steering Ratio равным 10, максимальный поворот руля получится равным 360 градусам, а максимальный поворот передних колес 36 градусов.

На этом этапе можно произвести экспорт и загрузить сцену в движок. Рекомендуется создать дорожную поверхность с физическим материалом. В просмотрщике нажать клавишу Q для выбора контролируемого объекта, и выбрать шасси. Использовать W, A, S, D для управления.

Дополнительно можно настроить демпфирование Damping перемещения (Translation) и вращения (Rotation). Свойство влияет на скорость перемещения и инерционность ТС.

Настройка трения и эластичности физического материала дорожного покрытия не влияют на поведение ТС.

## 22.6 Плавающие объекты



Для того, чтобы объект мог плавать на поверхности воды (объекта с материалом **Water**), необходимо активировать панель **Floater**. Существует два типа частей плавающего объекта: **Main Body** - непосредственно сам плавающий объект и **Bob** - вспомогательный объект-поплавок, на который будет действовать выталкивающая из воды сила. Плавающий объект может иметь неограниченное количество объектов типа **Bob**. В качестве поплавков могут использоваться как меши, так и объекты типа **Empty**.

Всем объектам, входящим в состав одного плавающего объекта необходимо выставить одинаковое имя в поле **Floater Name**

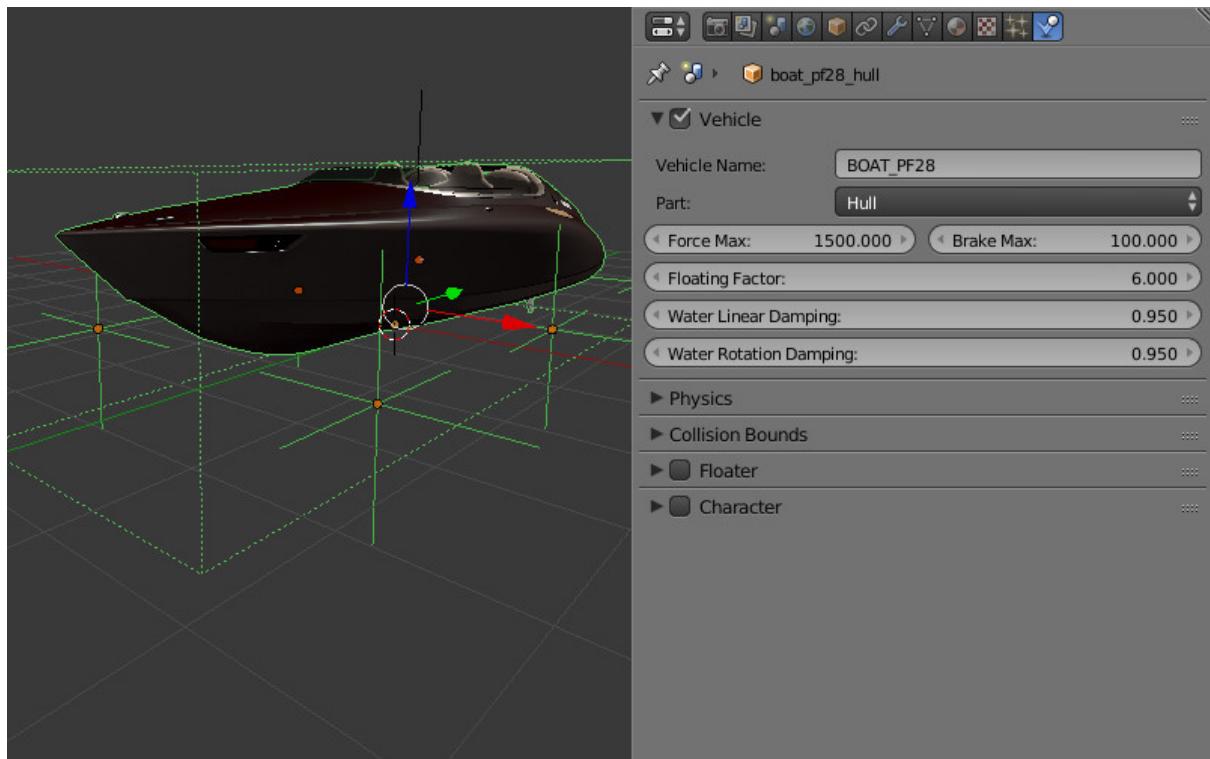
## 22.6.1 Параметры настройки плавающего объекта

**Floating Factor** Коэффициент выталкивания объекта из воды.

**Water Linear Damping** Демпфирование линейной скорости при нахождении объекта на поверхности воды (или под водой). Когда объект находится вне воды, используется значение из настроек физики.

**Water Rotation Damping** Демпфирование вращения при нахождении объекта на поверхности воды (или под водой). Когда объект находится вне воды, используется значение из настроек физики.

## 22.7 Плавающие транспортные средства



Плавающие транспортные средства используют часть параметров из настроек колесных транспортных средств **Vehicle** и все настройки как у объектов типа **Floater**. На основном объекте необходимо выставить *Part*, типа **Hull**. Так же как и плавающий объект плавающее транспортное средство требует наличия вспомогательных объектов типа **Bob**.

### 22.7.1 Параметры настройки плавающего транспортного средства

**Force Max** Максимальная движущая сила транспортного средства.

**Brake Max** Максимальный коэффициент торможения.

**Floating Factor** Коэффициент выталкивания объекта из воды.

**Water Linear Damping** Демпфирование линейной скорости при нахождении объекта на поверхности воды (или под водой). Когда объект находится вне воды, используется значение из настроек физики.

**Water Rotation Damping** Демпфирование вращения при нахождении объекта на

поверхности воды (или под водой). Когда объект находится вне воды, используется значение из настроек физики.

## 22.8 Особенности использования в приложениях

Физическая подсистема реализована в модуле **uranium.js** и загружается отдельно от основного кода движка. Модуль **uranium.js** представляет собой модификацию физического движка Bullet, портированную для работы в браузерах. Быстрое подключение физической подсистемы можно осуществить, разместив файлы **uranium.js** и **uranium.js.mem** в той же директории, где расположен исходный код движка, используемого в приложении.

В противном случае необходимо указать путь к модулю, используя конструкцию вида:

```
m_config.set("physics_uranium_path", ".../uranium.js");
```

---

**Примечание:** При разработке приложения *в составе SDK* путь к физическому движку определяется автоматически.

---

Если использование физики не требуется, рекомендуется отключить флаг **Enable Physics** в панели **Physics** на вкладке сцены в Blender'e. Также можно принудительно отключить загрузку модуля **uranium.js**, если до начала инициализации движка вызывать следующий метод:

```
m_config.set("physics_enabled", false);
```

---

## Нелинейная анимация

---

### 23.1 Редактор NLA

Редактор нелинейной анимации, имеющийся в программе Blender, позволяет в удобной форме задавать поведение сцены. С его помощью можно реализовать несложные сценарии. Таким образом исключается необходимость программирования простых сцен и приложений.



Движком поддерживается управление следующими сущностями:

- Любая анимация, параметры которой могут быть представлены с помощью *действий* (Action)

- Воспроизведение звуков
- Эмиссия частиц (в виде привязки к глобальной шкале времени)



### 23.1.1 Использование

1. Во вкладке **Scene** активировать панель **NLA**.
2. В редакторе **NLA Editor** задать необходимую конфигурацию поведения сцены.
3. На панели **Timeline** выбрать интервал времени анимации.

### 23.1.2 Дополнительные настройки

Настройка сцены *NLA > Cyclic NLA* позволяет активировать режим циклической NLA-анимации.

---

**Примечание:** Для использования вертекской анимации необходимо выставить опцию “Allow NLA” на *панели запекания вертекской анимации*.

---

### 23.1.3 Ограничения

- Одновременное проигрывание различных видов анимации для одного объекта не поддерживается.

## 23.2 Управление через API

Для управления нелинейной анимацией через API существует модуль `nla.js`.

```
// ...
var m_nla = require("nla");
// ...
m_nla.set_frame(150);
// ...
var frame = m_nla.get_frame();
// ...
m_nla.play();
// ...
m_nla.stop();
// ...
```

При использовании *Logic Editor* функции `set_frame`, `play`, `stop` недоступны.

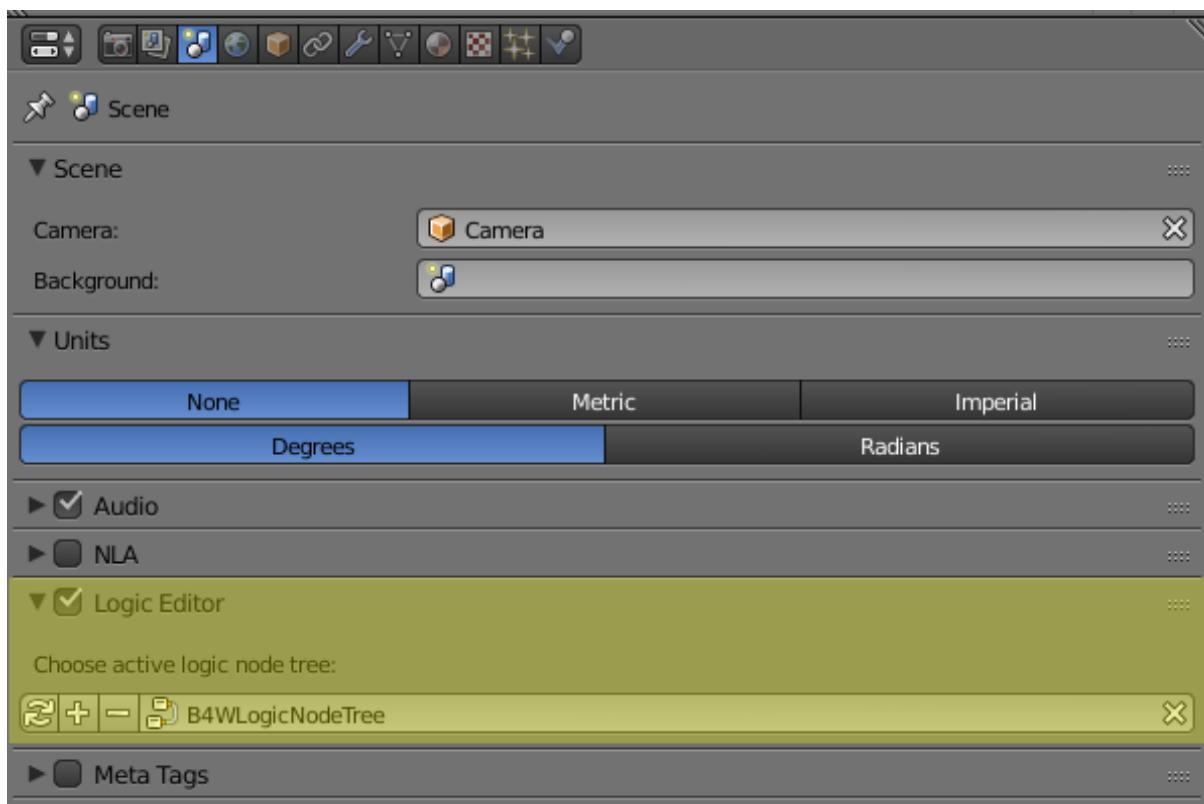
## Визуальное программирование

### 24.1 Визуальное программирование (Logic Editor)

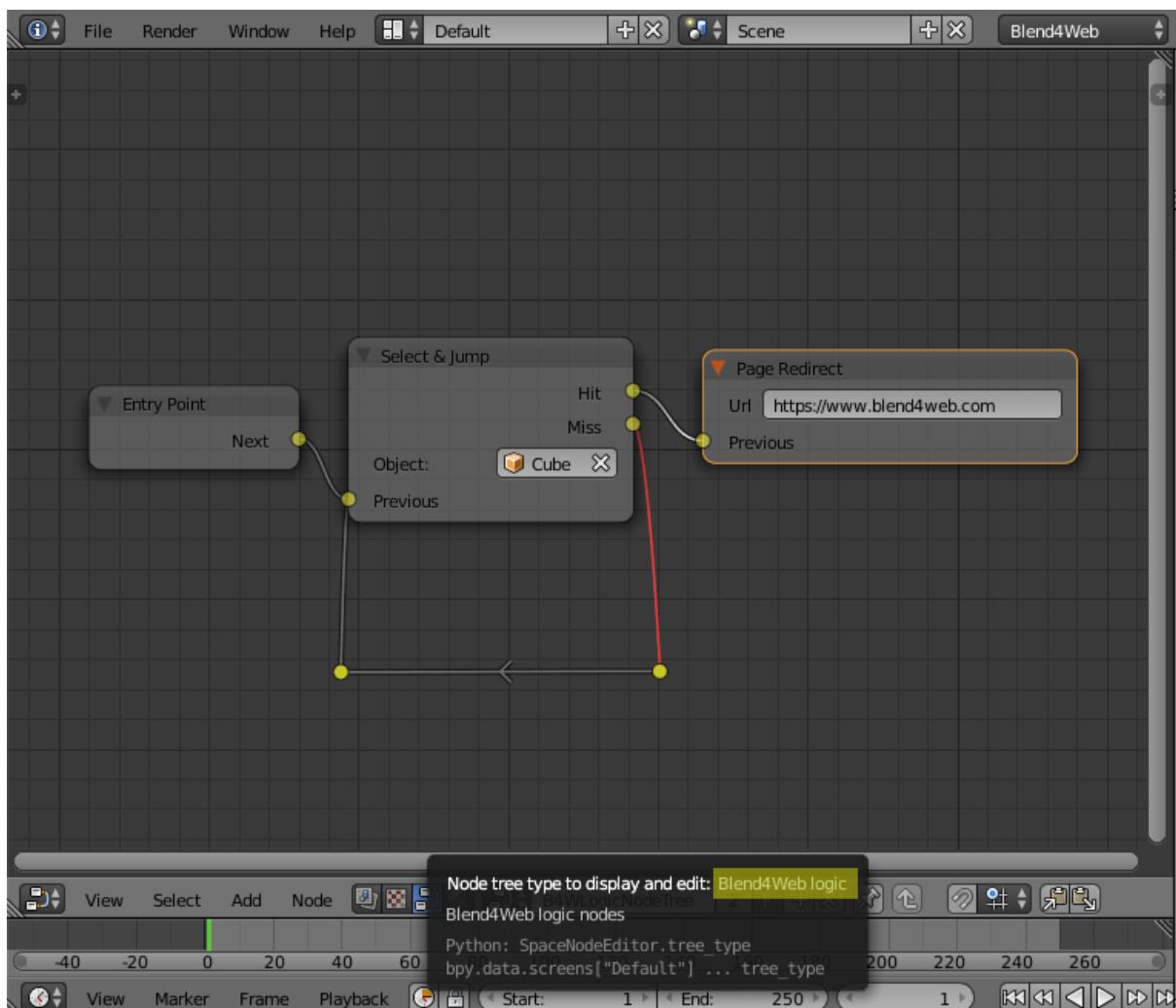
Осуществляется путем создания нодового дерева логики в окне нодового редактора. С помощью Logic Editor можно существенно расширить функциональность сцены, не прибегая при этом к программированию.



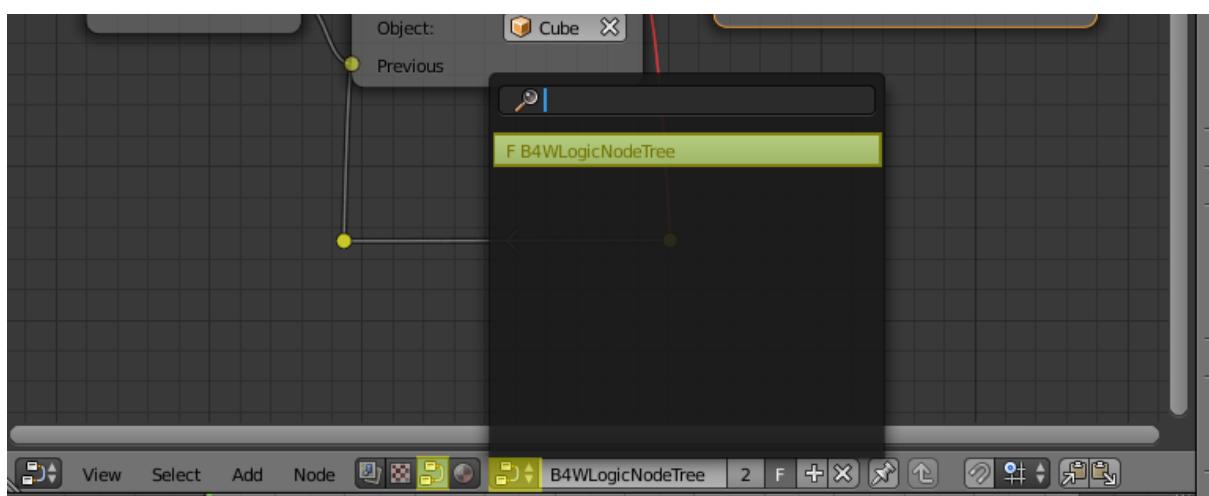
Для активации логики, на вкладке настроек сцены Blender необходимо установить галочку Logic Editor и добавить/выбрать активное нодовое дерево:



Для редактирования дерева логики необходимо перейти в Редактор Нод (Node Editor) выбрать тип нодового дерева Blend4Web logic:



Затем выбрать дерево для редактирования:

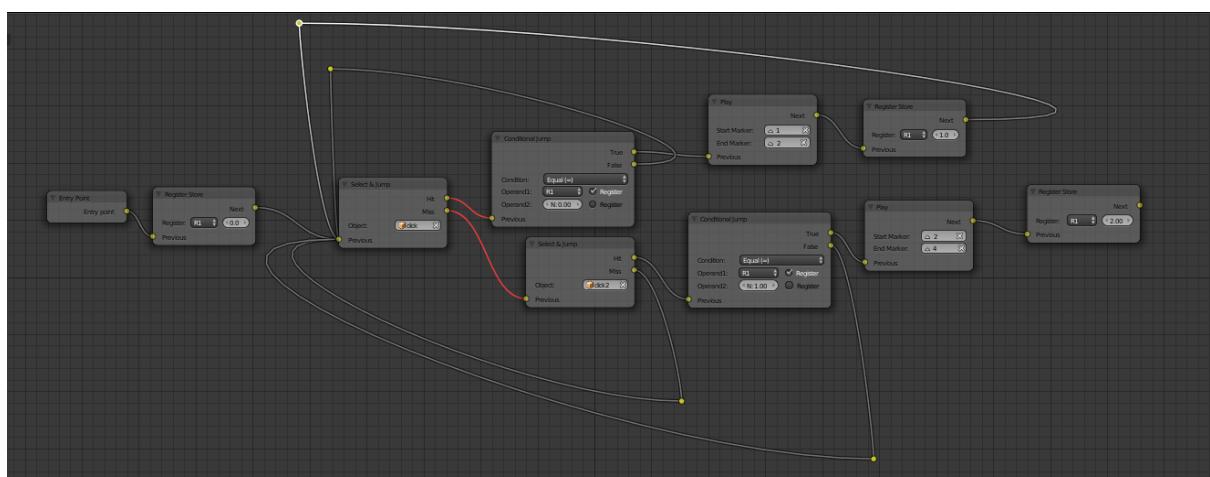


Создание нод осуществляется привычным для Blender сочетанием клавиш **Shift+A**.

Ноды - это логические блоки, выполнение которых начинается с **Entry Point**, которая имеет один единственный выход. Почти все остальные ноды помимо одного или двух выходов имеют также входы, таким образом они могут быть вставлены в любое место в дереве. Ноды без выходов, например **Page Redirect**, могут быть вставлены только в конец какой-либо ветви. Ноды с двумя выходами позволяют ветвить алгоритм в зависимости от успеха/провала указанного условия, например **Conditional Jump**.

Для реализации сложной логики предусмотрены числовые переменные, называемые регистрами. Каждый из 8 регистров может хранить в себе одно числовое значение. Регистры могут быть использованы для хранения какого-либо состояния сцены (например, это может быть счётчик проигрываемых анимаций, количество жизней, оставшихся у персонажа и т.д.).

Пример использования Logic Editor:



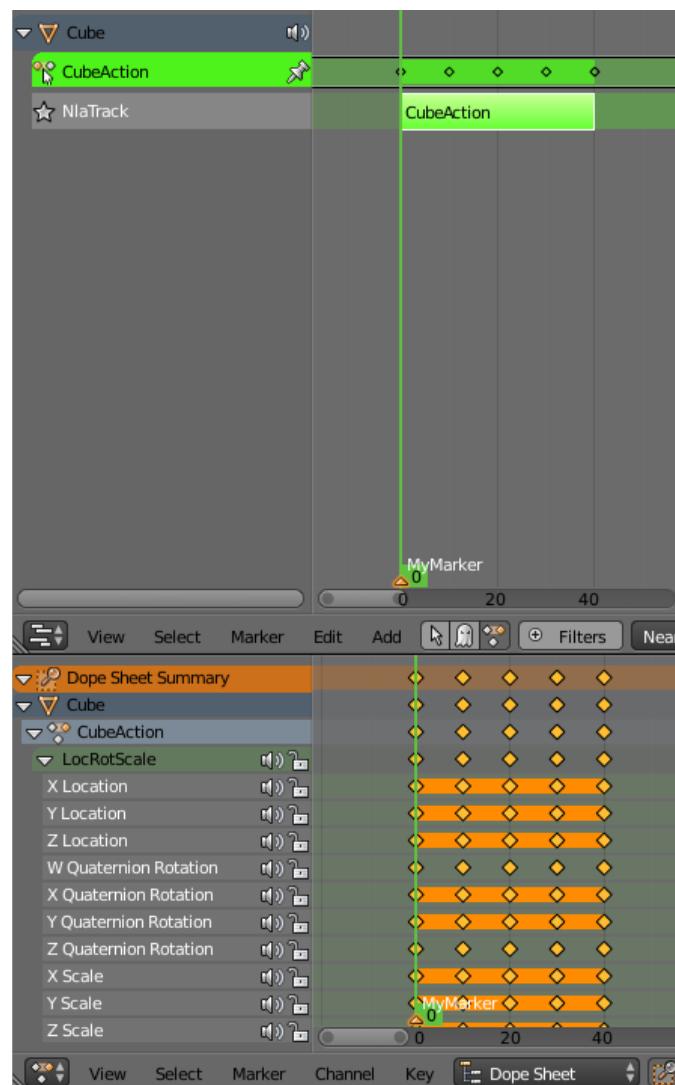
Ниже рассмотрены все возможные варианты нод.

### 24.1.1 Entry Point

Является точкой начала выполнения скрипта.

### 24.1.2 Play Timeline

Проигрывать участок NLA, начиная с кадра, на который указывает маркер. Анимация воспроизводится до следующего маркера, либо до конца шкалы времени сцены, после чего управление переходит к следующей ноде.



### 24.1.3 Play Animation

Проигрывает объектную анимацию. Присутствует возможность указать название анимации.

### 24.1.4 Select & Play Timeline

Ожидать, когда пользователь выберет объект (с помощью клика мышью в версии движка для десктопа, либо нажатия на экране в мобильной версии). Если выбран объект, указанный в параметрах ноды, начать играть анимацию аналогично ноде **Play Timeline**, если же был выбран любой другой объект - немедленно передать управление следующей ноде.

### 24.1.5 Select & Play Animation

Ожидать, когда пользователь выберет объект (с помощью клика мышью в версии движка для десктопа, либо нажатия на экране в мобильной версии). Если выбран объект, указанный в параметрах ноды, начать играть анимацию аналогично ноде *Play Animation*, если же был выбран любой другой объект - немедленно передать управление следующей ноде.

### 24.1.6 Select

Аналогично ноде *Select & Play*, за исключением того, что вместо анимации осуществляется переход. Указанная функция позволяет реализовать более сложную логику, поскольку появляется возможность распознавания результата выбора пользователя.

### 24.1.7 Conditional Jump

Перейти к указанной ноде в случае выполнения выбранного условия. В качестве параметров условия (операндов) могут выступать также регистры, которые активируются с помощью соответствующих переключателей.

### 24.1.8 Variable Store

Записать числовое значение в переменную.

### 24.1.9 Math Operation

Выполнить математическую операцию и сохранить результат в регистр. Любой из параметров (операндов) может быть либо числовым значением, либо регистром.

### 24.1.10 Show Object и Hide Object

Используются для скрытия и отображения объектов.

### 24.1.11 Page Redirect

Служит для перенаправления на другие веб-страницы.

### 24.1.12 Page Param

Позволяет сохранить произвольный параметр веб-страницы в выбранном числовом регистре.

### 24.1.13 Send Request

Позволяет отправить HTTP GET запрос на указанный URL и поместить значения полей ответа в переменные.

### 24.1.14 Inherit Material

Позволяет скопировать атрибуты одного материала на другой.

### 24.1.15 Set Shader Node Param

Позволяет установить значение шейдерной ноды. В данный момент поддерживаются ноды Value и RGB.

### 24.1.16 Delay

Позволяет установить задержку перед выполнением следующей ноды.

### 24.1.17 Apply Shape Key

Позволяет задать значение параметра Shape Key.

### 24.1.18 Outline

Позволяет управлять эффектом обводки объекта.

### 24.1.19 Отладка

Для отладки имеется возможность отключения (muting) нод. Для этого необходимо выделить ноду и нажать горячую клавишу M. Отключенная нода не выполняется, но при этом передает управление следующей ноде. Если у отключенной ноды имеется несколько выходов, то будет происходить выполнение ветки со стороны выхода отрицательного результата (Miss, False).

---

## Разработчикам приложений

---

### 25.1 Hello world!

Простейшее приложение на основе Blend4Web может иметь вид:

```
<!DOCTYPE html>
<html>
<head>
<script src="b4w.min.js"></script>
<script>
function hello() {
    var m_version = b4w.require("version");
    document.body.innerHTML = "Hello, Blend4Web " + m_version.version() + "!";
}
</script>
</head>

<body onload="hello()"></body>

</html>
```

Приложение выводит сообщение и текущую версию движка в окне браузера. Рассмотрим представленный пример. Базовый код движка подключается с помощью тега `<script src="...".js>`. Далее, приложение ожидает окончания загрузки страницы и выводит сообщение в окне браузера. В данном примере используется единственный модуль `version`, в котором находится одноимённая функция `version()`. Подробную информацию о предназначении модулей и функций движка можно найти в [документации по API](#).

Файл `b4w.min.js` со скомпилированным кодом движка необходимо скопировать из директории SDK `deploy/apps/common` и разместить в той же директории, что и представленный HTML-файл.

## 25.2 Загрузка сцены в приложение

Для того, чтобы загрузить трёхмерную сцену, требуется выполнить следующую последовательность действий:

1. Разместить на странице элемент <canvas>, на котором будет производиться рендеринг.
2. После загрузки страницы, для инициализации контекста WebGL, вызвать функцию `m_main.init()` с идентификатором созданного элемента.
3. Вызвать функцию `m_data.load()` для загрузки трёхмерной сцены.

```
<!DOCTYPE html>
<html>
<head>
<script src="b4w.min.js"></script>
<script>
function hello() {
    var m_main = b4w.require("main");
    var m_data = b4w.require("data");

    var canvas_elem = document.getElementById("canvas_id");
    m_main.init(canvas_elem);
    m_data.load("some_scene.json");
}
</script>
</head>

<body onload="hello()"><canvas id="canvas_id"></canvas></body>

</html>
```

---

**Примечание:** Следует отметить, что реальное приложение должно включать в себя проверку ошибок, настройку движка перед инициализацией, а также базовую систему взаимодействия с пользователем.

---

## 25.3 Система модулей

Несмотря на то, что движок предоставляет прикладному программисту API в объёме десятков модулей, в процессе работы он занимает в глобальном пространстве имён единственный объект `b4w`. При необходимости обращения к модулю, последний импортируется с помощью вызова функции `b4w.require()`.

Допустима регистрация сторонних модулей, если их имена не пересекаются с имеющимися. Регистрация происходит посредством вызова `b4w.register()`. Проверка наличия модуля с некоторым именем может быть осуществлена с помощью `b4w.module_check()`.

Пример:

```
// check if module exists
if (b4w.module_check("my_module"))
    throw "Failed to register module: my_module";

// register my_module
b4w.register("my_module", function(exports, require) {

    // import module "version"
    var m_version = require("version");

    // export print_build_date() from module "my_module"
    exports.print_build_date = function() {
        // exec function date() from module "version"
        console.log("Engine build date: " + m_version.date());
    }
});

// import module "my_module"
var m_my_module = b4w.require("my_module");

// exec function print_build_date() from module "my_module"
m_my_module.print_build_date();
```

## 25.4 Быстрое создание приложений

Поскольку создание приложения с нуля может быть достаточно сложной операцией, особенно для начинающих пользователей, в движке существует специальное дополнение `app`:

```
<!DOCTYPE html>
<html>
<head>
<script src="b4w.full.min.js"></script>
<script>

var m_app = b4w.require("app");
var m_data = b4w.require("data");

m_app.init({
    canvas_container_id: "container_id",
    callback: load_cb
})

function load_cb() {
    m_data.load("some_scene.json", loaded_cb);
}
```

```

function loaded_cb() {
    m_app.enable_controls();
    m_app.enable_camera_controls();
}

</script>
</head>

<body>
    <div id="container_id" style="width: 350px; height: 200px;"></div>
</body>

</html>

```

В данном случае модуль `app` создаст элемент `<canvas>` внутри контейнера с указанным идентификатором `container_id`, осуществит инициализацию движка при загрузке страницы и сообщит о её окончании с помощью обработчика `load_cb()`.

Далее загружается сцена `some_scene.json`, аналогично предыдущему примеру, с тем лишь отличием, что по окончании загрузки этой сцены инициализируется подсистема управления и активируются средства для перемещения камеры с помощью клавиатуры и мыши (сенсорного экрана для мобильных устройств).

При использовании модуля `app` необходимо явно задавать размеры контейнерного элемента, поскольку в противном случае создаваемый элемент `<canvas>` будет иметь нулевые размеры.

## 25.5 Система управления проектами

Начиная с версии 15.09, SDK включает систему управления проектами, которая позволяет:

- просматривать список и внутреннюю структуру имеющихся проектов;
- запускать приложения, просматривать сцены в программе-просмотрщике, загружать исходные файлы сцен в Blender;
- создавать и конфигурировать новые приложения, в том числе на основе готовых шаблонов;
- осуществлять сборку приложения и его конверсию в форму, удобную для последующего размещения на сервере;
- конвертировать ресурсы приложения (текстуры, звуковые и видео-файлы) в альтернативные форматы для обеспечения кросбраузерности и кросплатформенности;
- автоматизировать повторный экспорт всех сцен из состава приложения, включая файлы в формате JSON и HTML.
- удалять проекты.

Управлять проектами возможно двумя способами: с помощью консольной утилиты *project.py*, имеющей простой формат вызова и подробную справку, либо в графической форме с помощью веб-приложения, выполняющегося на локальном сервере разработчика. Сервер не требует каких-либо дополнительных настроек и запускается автоматически при старте Blender. Главная страница приложения Project Manager открывается с помощью кнопки **Project Manager** в Blender.

Система управления проектами работает на всех операционных системах, однако для некоторых операций может потребоваться установка дополнительных зависимостей. Проверить, установлены ли необходимые зависимости, можно с помощью команды:

```
./project.py check_deps
```

Для пользователей ОС Windows:

```
python project.py check_deps
```

### 25.5.1 Project Manager

Данное графическое приложение позволяет осуществлять базовые функции управления проектами, такие как просмотр, компиляция, экспорт сцен и конвертация ресурсов. Разработчикам, которым требуется полный набор функций для управления проектами, следует использовать скрипт *project.py* и ручное редактирование конфигурационных файлов *.b4w\_project*.

Приложение *Project Manager* доступно из раздела *Tools* главной страницы SDK. При запуске приложение выводит список из всех проектов, находящихся в установленном SDK.

Команды для управления проектами находятся в верхней части страницы, для создания нового проекта необходимо нажать на ссылку **[Create New Project]**; для загрузки внешнего проекта - **[Import Project Archive]**; чтобы скрыть или показать проекты, изначально входящие в состав SDK - **[Hide/Show Stock Projects]**.

#### Project Manager



Команды управления конкретным проектом находятся справа от него

## Operations

1

compile project

2

reexport scenes

3

convert resources

4

export project archive

5

remove project

1. Компиляция проекта (недоступно для типов проекта WebPlayer JSON и WebPlayer HTML).
2. Реэкспорт блендеровских сцен в проекте.
3. Конвертация медиа ресурсов.
4. Экспортирование и дальнейшая загрузка проекта.
5. Удаление проекта со всеми зависимостями.

**Примечание:** Все пути берутся из конфигурационного файла .b4w\_project.

Далее подробно рассмотрен мастер создания новых проектов.

**Project Manager**

[← Back to Projects](#)
[Help](#)

1	<b>Project Name (required)</b> <i>for better compatibility, use only letters and underscore symbols</i>	<input type="text"/>
2	<b>Project Title (optional)</b> <i>choose a human-readable name</i>	<input type="text"/>
3	<b>Project Author / Company (optional)</b> <i>choose a human-readable name</i>	<input type="text"/>
4	<b>Create Application Starter Files</b> <i>create starter html/js/css files</i>	<input type="checkbox"/>
5	<b>Create Scene Starter Files</b> <i>create starter blend/json/bin files</i>	<input type="checkbox"/>
6	<b>Copy Project Management Script</b> <i>simplifies project management for users who prefer using command line tools</i>	<input type="checkbox"/>
7	<b>Bundled Project</b> <i>all project files will be placed into a single directory good for small projects such as tutorials etc</i>	<input type="checkbox"/>
8	<b>Engine Binding Type</b> <i>if unsure leave the default value see help for more info</i>	<input checked="" type="radio"/> External <input type="radio"/> Copy <input type="radio"/> WebPlayer JSON  <input type="radio"/> Compile <input type="radio"/> Update <input type="radio"/> WebPlayer HTML
9	<b>JavaScript Obfuscation Level</b> <i>if unsure, leave the default value see help for more info</i>	<input checked="" type="radio"/> Simple <input type="radio"/> Advanced <input type="radio"/> Whitespace Only
<a href="#" style="border: 1px solid #ccc; padding: 2px 10px; border-radius: 5px;">✓ Create Project</a>		

1. Имя проекта. Текущим именем будут названы директории в “apps\_dev/имя\_проекта”, “deploy/assets/имя\_проекта”, blender и “deploy/apps/имя\_проекта”. Для лучшей совместимости рекомендуется использовать буквы и нижнее подчеркивание.
2. Название проекта. Это имя будет показываться в заголовке веб-браузера.
3. Имя автора проекта.
4. Добавить шаблоны приложения. В директорию проекта “apps\_dev/имя\_проекта” будут добавлены стандартные шаблоны приложения: html-файл, css-файл, js-файл.
5. Добавить шаблоны сцены. В директорию “deploy/assets/имя\_проекта” будет добавлен стандартный json-файл; в директорию “blender/имя\_проекта” - blend-файл.
6. Скопировать скрипт менеджера проектов. Скрипт project.py будет скопирован в директорию проекта.
7. Все файлы проекта будут находиться в одной директории. Предпочтительно использовать для маленьких проектов, таких как уроки и примеры. Для данной

опции доступен только тип проекта - “update”.

8. Тип собираемого проекта. Здесь возможны варианты:

- “External” - движок берется из директории “deploy/apps/common/”. Компилируются только файлы приложения;
- “Copy” - движок копируется в директорию собранного приложения. Компилируются только файлы приложения;
- “Compile” - исходники движка компилируются вместе со скриптами приложения;
- “Update” - заменяется движок в директории проекта;
- “Webplayer JSON” - json-файл, находящийся в проекте, запускается при помощи веб-плеера в составе SDK;
- “Webplayer HTML” - проект состоит из html-файла, внутри которого находятся все необходимые ресурсы.

9. Уровень оптимизации javascript.

- “Simple” - в коде заменяются имена переменных;
- “Advanced” - производится оптимизация кода.
- “Whitespace Only” - в коде удаляются только пробелы;

## 25.5.2 Структура проекта

```
blend4web/
    apps_dev/
        myproject/
            project.py
            .b4w_project
            myproject.js
            myproject.css
            myproject_dev.html
    blender/
        myproject/
            myproject.blend
    deploy/
        apps/
            my_project/
                myproject.js
                myproject.css
                myproject.html
        assets/
            myproject/
                myproject.json
                myproject.bin
```

### 25.5.3 Конфигурационный файл .b4w\_project

Если при запуске скрипта *project.py* не были заданы параметры проекта, то они берутся из конфигурационного файла.

```
[info]
author =
name =
title =

[paths]
assets_dirs =
blend_dirs =
blender_exec =
build_dir =
deploy_dir =

[compile]
apps =
css_ignore =
engine_type = external
js_ignore =
optimization = simple
use_physics =
use_smaa_textures =
version =

[deploy]
assets_path_prefix =
remove_exist_ext_dir =
```

### 25.5.4 Создание проекта

```
./project.py init myproject
```

Команда создает проект с указанной именем в текущей директории. По умолчанию в директории проекта будет находиться только конфигурационный файл.

Доступные опции:

- **-A | --copy-app-templates** (необязательная) создает в директории проекта стандартные шаблоны приложения (*<имя проекта>.dev.html*, *<имя проекта>.js*, *<имя проекта>.css*).
- **-B | --bundle** (необязательная) все файлы проекта будут размещены в одной директории.
- **-C | --author** (необязательная) записывает в конфигурационный файл имя автора или компании.

- **-o | --optimization** (необязательная) записывает в конфигурационный файл тип оптимизации скриптов.
- **-P | --copy-project-script** (необязательная) создает скрипт *project.py* в директории проекта.
- **-S | --copy-scene-templates** (необязательная) создает в директориях *deploy/assets/<имя проекта>* и *blender/<имя проекта>* (необязательная) стандартные шаблоны сцены (*<имя проекта>.json/.bin* и *<имя проекта>.blend* соответственно).
- **-T | --title"** (необязательная) записывает в конфигурационный файл заголовок проекта. При сборке он будет добавлен в html-тэг *<title>*.
- **-t | --engine-type** (необязательная) записывает в конфигурационный файл тип собираемого приложения.

Пример:

```
./project.py init -AS -C Blend4Web -o simple -T MyProject -t external myproject
```

Данная команда создаст директорию *myproject*, в которой будут находиться файлы: *myproject.js*, *myproject.css*, *myproject\_dev.html* и *.b4w\_project*.

Файл *.b4w\_project* будет выглядеть следующим образом:

```
[info]
author = Blend4Web
name = myproject
title = MyProject

[paths]
assets_dirs = deploy/assets/myproject;
blend_dirs = blender/myproject;
blender_exec = blender
build_dir = deploy/apps/myproject
deploy_dir =

[compile]
apps =
css_ignore =
engine_type = external
js_ignore =
optimization = simple
use_physics =
use_smaa_textures =
version =

[deploy]
assets_path_prefix =
remove_exist_ext_dir =
```

### 25.5.5 Создание приложения в составе проекта

В проекте может содержаться несколько приложений. Для этого необходимо в конфигурационном файле указать соответствующие HTML-файлы через точку с запятой:

```
...
[compile]
apps = myapp1;myapp2;
...
```

### 25.5.6 Сборка проекта

```
python3 project.py -p myproject compile
```

Собирает проект в директории `deploy/apps/myproject`.

Доступные опции:

- `a | --app"` (необязательная) указывает на HTML-файл, относительно которого будет собираться приложение для проекта.
- `c | --css-ignore"` (необязательная) добавляет в исключения стили, которые не будут скомпилированы.
- `j | --js-ignore"` (необязательная) добавляет в исключения скрипты, которые не будут скомпилированы.
- `o | --optimization"` (необязательная) используется для указания метода оптимизации js-файлов. Доступные варианты: `whitespace`, `simple` (по умолчанию) и `advanced`.
- `t | --engine-type"` (необязательная) определяет тип компилируемого приложения. Доступны четыре варианта: `external` (по умолчанию), `copy`, `compile`, `update`.
- `v | --version"` добавляет версию к адресам скриптов и стилей.

### 25.5.7 Автоматический экспорт ресурсов

```
python3 project.py -p myproject reexport
```

Повторно экспортирует blend-файлы в форматах JSON и HTML.

Доступные опции:

- `b | --blender_exec"` путь к исполняемому файлу blender.
- `s | --assets"` определяет директорию с ресурсами сцены.

### 25.5.8 Конвертация ресурсов

```
python3 project.py -p myproject convert_resources
```

Конвертирует внешние ресурсы (текстуры, звуковые и видео-файлы) проекта в альтернативные форматы для обеспечения кросбраузерности и кросплатформенности.

Доступные опции:

- `s | --assets"` определяет директорию с ресурсами сцены.

### 25.5.9 Список проектов

```
python3 project.py -p myproject list
```

Выводит список проектов, находящихся в директории `apps_dev`.

### 25.5.10 Развёртывание проекта

```
python3 project.py -p myproject deploy
```

Сохраняет проект во внешнюю директорию со всеми необходимыми зависимостями.

Доступные опции:

- `d | --dir"` директория для развертывания проекта.
- `e | --assets-path"` путь к файлам ресурсов.
- `o | --override"` (необязательная) удаляет директорию, если она существует.
- `s | --assets"` определяет директорию с ресурсами сцены.
- `t | --engine-type"` (необязательная) определяет тип развертываемого приложения.

### 25.5.11 Удаление проекта

```
python3 project.py -p myproject remove
```

Удаляет проект. Удаляемые директории берутся из конфигурационного файла.

## 25.6 Подготовка ресурсов

Следующим этапом является *подготовка сцены*. Наиболее удобно хранить все ресурсы, используемые приложением, в одной директории. Затем производится построение сцены в Blender'е и *экспорт сцены* в директорию с ресурсами.

Для создания кросбраузерного и кросплатформенного приложения требуется *конвертация ресурсов*.

На этапе создания приложения удобно использовать необфусцированную версию движка, потому что при этом можно легко отслеживать логические ошибки в коде приложения. Для ее подключения необходимо перейти в директорию для созданного приложения:

```
cd <path_to_sdk>/apps_dev/example
```

Для пользователей ОС Windows:

```
cd <path_to_sdk>\apps_dev\example
```

Затем необходимо вызвать скрипт, генерирующий пути к файлам движка:

```
python3 ../../scripts/mod_list.py
```

Для пользователей ОС Windows:

```
python ..\..\scripts\mod_list.py
```

---

**Примечание:** Стоит отдельно отметить, что для запуска скриптов требуется интерпретатор языка Python версии 3.x

---

В консоли появится список модулей, которые необходимо скопировать и вставить в главный HTML-файл:

```
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1">
    <script type="text/javascript" src="../../src/b4w.js"></script>
    <script type="text/javascript" src="../../src/anchors.js"></script>
    <script type="text/javascript" src="../../src/animation.js"></script>
    <script type="text/javascript" src="../../src/assets.js"></script>
    <script type="text/javascript" src="../../src/batch.js"></script>
    <script type="text/javascript" src="../../src/boundings.js"></script>
    <script type="text/javascript" src="../../src/camera.js"></script>
    .
</head>
```

## 25.6.1 Написание логики приложения

Следующим шагом станет написание скриптов с логикой работы приложения, которые необходимо подключить в главном HTML-файле:

```
<head>
    .
    .
    <script type="text/javascript" src="example.js"></script>
</head>
```

Стоит отметить отдельно, что при загрузке .json-файлов удобнее всего использовать функцию `get_std_assets_path()` из модуля `config.js`:

```
m_data.load(m_config.get_std_assets_path() + "example/example.json", load_cb);
```

После сборки готового приложения путь к данным изменится, а использование функции `get_std_assets_path()` позволит избежать возникновения проблемы с неверным путем.

## 25.6.2 Сборка готового приложения

В составе SDK присутствует скрипт `apps_dev/project.py`, предназначенный для сборки приложений.

Удобнее всего первым шагом осуществить компиляцию приложения, чтобы сразу же проверить его работоспособность. Сделать это можно выполнив команду:

```
./project.py -p example -o advanced
```

Для пользователей ОС Windows:

```
python project.py -p example -o advanced
```

---

**Примечание:** Для работы скрипта необходимо установить `java` и записать ее в переменную среды `PATH`

---

По умолчанию директория проекта полностью копируется в директорию `deploy/apps/project_name/`. Скрипты и стили компилируются относительно родительской директории.

## 25.6.3 Экспорт приложения за пределы SDK

Для экспорта готового приложения за пределы SDK (для последующего развертывания) необходимо выполнить команду:

```
./project.py -p example -o advanced -b copy -s ../deploy/assets/example -d ../my_apps
```

Для пользователей ОС Windows:

```
python project.py -p example -o advanced -b copy -s ..\deploy\assets\example -d ..\my_apps
```

Требования, накладываемые компилятором:

- В корне директории должен находиться единственный html-файл, если не указана опция `-a`.
- Скрипты и стили могут находиться как в корне проекта (приложения), так и во вложенных папках.

## 25.7 Конвертация ресурсов

Существующие браузеры не полностью поддерживают основные форматы медиаданных, поэтому для создания кроссбраузерных приложений, а также с целью оптимизации, необходимо использовать конвертер ресурсов.

В состав дистрибутива включен Python скрипт (`scripts/converter.py`) для конвертации исходных файлов в другие форматы с целью расширения спектра поддерживаемых платформ, а также для уменьшения размера ресурсов. Для конвертации необходимо выполнить команду:

```
> python3 <path_to_sdk>/scripts/converter.py [-d file_path] resize_textures | convert_dds | c
```

Для пользователей ОС Windows:

```
cd <path_to_sdk>\scripts
python converter.py [-d file_path] resize_textures | convert_dds | convert_media
```

---

**Примечание:** Стоит отдельно отметить, что для запуска скриптов требуется интерпретатор языка Python версии 3.x

---

С помощью опции `-d` можно указать путь к директории, в которой будет производится конвертация.

При необходимости исключить некоторую директорию при конвертации, достаточно разместить в ней файл с именем `.b4w_no_conv`. На конвертацию во вложенных директориях это не повлияет.

Аргумент `resize_textures` используется для изменения размера текстур в режиме **LOW**.

### 25.7.1 Зависимости

Убедитесь, что у вас установлены все необходимые для конвертации программы. Это можно сделать следующей командой:

```
> python3 <path_to_sdk>/scripts/converter.py check_dependencies
```

Если какая-либо программа отсутствует, то будет выведено сообщения вида:

*Couldn't find PROGRAM\_NAME.*

**Linux**

Список необходимых программ можно посмотреть в таблице:

Название	Пакет в дистрибутиве Ubuntu 15.04
ImageMagick	imagemagick
NVIDIA Texture Tools	libnvtt-bin
Libav	libav-tools
FFmpeg	ffmpeg

**Примечание:** Пользователи Linux могут дополнительно установить пакет qt-faststart, служащий для оптимизации загрузки медиаданных.

## Windows

Для пользователей ОС Windows нет необходимости устанавливать эти пакеты, так как они уже находятся в составе SDK.

## Mac OS

Пользователи Mac OS могут установить менеджер пакетов brew, а затем с его помощью установить некоторые недостающие пакеты.

Перед началом установки пакетов произведите установку библиотек libpng и libjpeg, выполнив следующие команды в консоле:

```
> brew install libpng
> brew install libjpeg
```

Теперь можно приступать к установке необходимых зависимостей:

```
> brew install imagemagick
> brew install --with-theora --with-libvpx --with-fdk-aac ffmpeg
```

Для установки NVIDIA Texture Tools необходимо склонировать репозиторий, выполнив следующую команду:

```
> git clone https://github.com/TriumphLLC/NvidiaTextureTools.git
```

Теперь можно произвести сборку и установку пакета:

```
> cd NvidiaTextureTools
> ./configure
> make
> make install
```

## 25.7.2 Формат данных

Преобразование происходит по схеме:

**для аудио (convert\_media):**

- ogg -> mp4
- mp3 -> ogg
- mp4 -> ogg

Рекомендуется использовать в качестве базового формата ogg, в этом случае для обеспечения кросс-браузерной совместимости потребуется только преобразование из ogg в mp4. Пример файла на входе: file\_name.ogg, пример файла на выходе: file\_name.altconv.mp4.

**для видео (convert\_media):**

- webm -> m4v
- m4v -> webm
- ogv -> webm
- webm -> seq
- m4v -> seq
- ogv -> seq

Рекомендуется использовать в качестве базового формата WebM, в этом случае для обеспечения кросс-браузерной совместимости потребуется только преобразование из `webm` в `m4v` (из `webm` в `seq` для iPhone). Пример файла на входе: `file_name.webm`, пример файла на выходе: `file_name.altconv.m4v`.

**для изображений (convert\_dds):**

- png -> dds
- jpg -> dds

Пример файла на входе: `file_name.jpg`, пример файла на выходе: `file_name.jpg.dds`.

В целях оптимизации работы приложения существует возможность использования `min50` (уменьшенных вдвое) и DDS текстур. Для этого при инициализации приложения необходимо передать следующие параметры:

```
exports.init = function() {
    m_app.init({
        // ...
        assets_dds_available: true,
        assets_min50_available: true,
        // ...
    });
    // ...
}
```

Файл формата `.seq` представляет собой раскадрированное видео. Применяется на IE 11 и iPhone, поскольку на них возникают трудности при использовании видео стандартного формата в качестве текстуры. Использование dds-формата для изображений является более оптимальным по сравнению с другими форматами.

Движком могут использоваться файлы, созданные пользователем вручную и имеющие следующие наименования: `file_name.altconv.m4v`, `file_name.altconv.mp3` и т.д. Такие файлы необходимо размещать в одной директории с медиафайлом, используемым в Blender'e.

Вы также можете использовать бесплатную кросплатформенную программу [Miro Video Converter](#) для конвертации медиаданных.

## 25.8 Специфика неполноэкраных приложений

Элемент Canvas, на котором осуществляется рендеринг, может изменять своё местоположение относительно окна браузера. Это может происходить в результате манипуляций, проводимых над DOM-деревом, либо в результате скроллинга страницы, что особенно актуально для неполноэкраных приложений.

В большинстве случаев это не будет никак сказываться на работе приложения. Однако для событий, связанных с положением курсора мыши или позицией касания на touch-устройстве, возможно получение некорректных результатов. Это происходит, потому что координаты, получаемые из соответствующих событий, принадлежат системе отсчета относительно окна браузера, а движок работает с координатами в системе отсчета именно Canvas элемента (верхний левый угол элемента).

- Если верхний левый угол Canvas'a совпадает с верхним левым углом окна браузера, и его местоположение не будет изменяться, то достаточно использовать координаты event.clientX и event.clientY соответствующего события либо функции API `get_coords_x()` и `get_coords_y()`:

```
var m_mouse = require("mouse");

// ...
var x = event.clientX;
var y = event.clientY;
// ...
var x = m_mouse.get_coords_x(event);
var y = m_mouse.get_coords_y(event);
// ...
```

- Если на странице присутствует только скроллинг окна браузера, то достаточно использовать координаты event.pageX и event.pageY:

```
// ...
var x = event.pageX;
var y = event.pageY;
// ...
```

- В случае более сложных манипуляций с положением элемента Canvas (скроллинг отдельных элементов страницы, смещение относительно левого верхнего угла окна браузера, манипуляции с DOM-деревом) требуется корректный пересчет координат. Чтобы получить координаты, подходящие для использования в движке, можно провести преобразование при помощи метода `client_to_canvas_coords()`:

```
var m_cont = require("container");
var _vec2_tmp = new Float32Array(2);
// ...
var canvas_xy = m_cont.client_to_canvas_coords(event.clientX, event.clientY, _vec2_tmp);
// ...
```

Для получения координат в системе отсчета Canvas элемента, движок должен знать его положение относительно окна браузера. Однако, если оно будет меняться во время работы приложения (тот же скроллинг), то необходимо будет как-то пересчитывать позицию Canvas'a. Для того, чтобы это происходило автоматически, нужно выставить настройку `track_container_position` при инициализации приложения:

```
exports.init = function() {
    m_app.init({
        // ...
        track_container_position: true,
        // ...
    });
    // ...
}
```

При её использовании в некоторых браузерах (например, Firefox) возможно незначительное снижение производительности вследствие частого обращения к DOM-дереву. Если этот момент критичен, то вместо флага `track_container_position` можно пользоваться методами `force_offsets_updating()`, `update_canvas_offsets()` или более низкоуровневым `set_canvas_offsets()` для обновления положения элемента Canvas вручную, когда это действительно необходимо:

```
var m_cont = require("container");
// ...
m_cont.force_offsets_updating();
// ...
m_cont.update_canvas_offsets();
// ...
m_cont.set_canvas_offsets(offset_left, offset_top);
// ...
```

## 25.9 Примеры кода

В составе SDK присутствует приложение Code Snippets, демонстрирующее примеры использования функционала движка.

На данный момент приложение включает в себя следующие примеры:

- Canvas Texture - пример работы с canvas-текстурой
- Camera Animation - создание процедурной анимации камеры

- Camera Move Styles - переключение режимов управления камерой
- Custom Anchors - процедурное создание аннотаций
- Dynamic Geometry - процедурное изменение геометрии
- Gyro (Mobile Only) - пример работы с гироскопом мобильных устройств
- Instancing - копирование объектов сцены
- Material API - изменение свойств материалов и замена материалов объекта
- Morphing - использование ключей деформации объекта

Приложение Code Snippets доступно по пути `SDK/apps_dev/code_snippets/code_snippets_dev.h`.  
Также оно доступно из файла `index.html` в корне SDK.

## 25.10 Событийная модель

Событийная модель предоставляет унифицированный интерфейс для описания изменения состояний 3D сцены, упрощая обработку событий физики и действий пользователя.

### 25.10.1 Сенсоры

Основным блоком событийной модели является сенсор (`sensor`). Сенсор является программной сущностью, и может быть только активным (1, единица) или неактивным (0, ноль). Некоторые сенсоры несут полезную нагрузку (`payload`). Например, сенсор трассировки лучей (`Ray Sensor`) предоставляет относительную длину луча пересечения.

Управление сенсорами не доступно пользователю в виде открытого API. Вместо этого каждый сенсор должен присутствовать в одном или нескольких множествах (`sensor manifold`). Множество является логическим контейнером, ассоциированным с объектом на сцене. Оно генерирует ответ на определенный набор событий сенсоров в виде вызова функции-обработчика. Для определения множества необходимо иметь следующую информацию (см. также описание функции `controls.create_sensor_manifold()` в документации по API):

- Объект-носитель множества (например, бросаемый объект).
- Уникальный идентификатор множества (например, “IMPACT”).
- Тип вызова функции-обработчика (варианты: `CT_CONTINUOUS` - непрерывный, `CT_LEVEL` - уровень, `CT_SHOT` - одномоментный, `CT_TRIGGER` - переключающий, `CT_CHANGE` - любое изменение любого из сенсоров).
- Массив сенсоров.
- Логическая функция, определяющая при какой комбинации состояний сенсоров вызывается функция-обработчик.

- Функция-обработчик.
- Необязательный параметр, который может быть передан в функцию-обработчик.

## 25.10.2 Пример

Поставлена задача озвучить удар бросаемого камня так, чтобы при ударе о различные среды (например, земля и стена) выводился характерный звук. На сцене в Blender'е имеются ограничивающие меши с физическими материалами, их идентификаторы “TERRAIN” и “WALL”. На сцене также присутствует бросаемый физический объект с названием “Stone”.

Определим по одному сенсору соударения (Collision Sensor) для каждой среды, по типу издаваемого звука.

```
// import the modules
var m_scenes = b4w.require("scenes");
var m_controls = b4w.require("controls");

// get the object being thrown
var stone = m_scenes.get_object_by_name("Stone");

// create the sensors
var sensor_impact_terrain = m_controls.create_collision_sensor(stone, "TERRAIN");
var sensor_impact_wall     = m_controls.create_collision_sensor(stone, "WALL");
```

Добавим сенсоры в массив. В качестве логической функции используем логическое ИЛИ. В обработчике напишем код для воспроизведения звука. Создадим множество сенсоров с идентификатором “IMPACT” и типом CT\_SHOT (одномоментный).

```
// array of the sensors
var impact_sens_array = [sensor_impact_terrain, sensor_impact_wall];

// manifold logic function
var impact_sens_logic = function(s) {return (s[0] || s[1]);};

// callback
var impact_cb = function(obj, manifold_id, pulse) {

    // NOTE: it's possible to play both sounds simultaneously

    if (m_controls.get_sensor_value(obj, manifold_id, 0) == 1) {
        // ...
        console.log("play the terrain impact sound");
    }

    if (m_controls.get_sensor_value(obj, manifold_id, 1) == 1) {
        // ...
        console.log("play the wall impact sound");
    }
}
```

```

}

// create the manifold
m_controls.create_sensor_manifold(stone, "IMPACT", m_ctl.CT_SHOT,
    impact_sens_array, impact_sens_logic, impact_cb);

```

При столкновении объекта “Stone” с любым из физических материалов “TERRAIN” или “WALL” происходит вызов функции-обработчика. Внутри этой функции получим значения обоих сенсоров по их индексу в массиве сенсоров (0 - “TERRAIN”, 1 - “WALL”). Значение сенсора = 1 (активный) означает, что произошло столкновение с соответствующим физическим материалом. В результате воспроизводится соответствующий звук (код не показан).

## 25.11 Файловая структура SDK

**apps\_dev** исходный код приложений

**code\_snippets** исходные файлы примеров кода

**dairy\_plant** исходные файлы приложения “Молочный завод” (доступно только в SDK Pro)

**fashion** исходные файлы приложения “Показ мод” (доступно только в SDK Pro)

**firstperson** исходные файлы приложения “Ферма” (доступно только в SDK Pro)

**flight** исходные файлы приложения “Остров”

**gallery** исходные файлы приложения “Натюрморт” (доступно только в SDK Pro)

**new\_year** исходные файлы открытки “С новым годом 2015”

**Makefile** файл для сборки всех приложений из состава SDK

**project.py** скрипт для разработчиков приложений

**victory\_day\_2015** исходные файлы открытки “День победы 70”

**viewer** исходные файлы приложения для просмотра сцен Viewer

**assets.json** метаданные с информацией о сценах, загружаемых просмотрщиком сцен Viewer

**webplayer** исходные файлы веб-плеера

**blender** исходные файлы сцен в формате Blender

**blender\_scripts** экспортер и вспомогательные скрипты для Blender’а

**csrc** исходный код бинарной части экспортёра движка и других утилит на языке С

**deploy** директория с ресурсами для размещения на сервере (исходные файлы сцен, скомпилированные приложения и документация)

**api\_doc** документация API движка для разработчиков (собирается автоматически, на основе исходного кода движка)

**apps** 3D-приложения, предназначенные для развертывания, директория дублирует *apps\_dev*

**common** Файлы скомпилированного движка. Используются приложениями из состава SDK (отсюда и название).

**assets** загружаемые ресурсы: сцены, текстуры, звуковые файлы

**doc** настоящее руководство пользователя в формате HTML, собирается автоматически из *doc\_src*

**globals\_detect** вспомогательный код для определения глобальных переменных

**tutorials** исходные файлы обучающих примеров

**doc\_src** исходный код настоящего руководства пользователя на языке разметки reST

**index.html** и **index\_assets** файлы главной веб-страницы SDK

**license** файлы с текстами лицензионных соглашений

**Makefile** файл сборки для компиляции движка, приложений, документации, развертывания на удаленном сервере (недоступен в бесплатной версии)

**README.rst** файл README

**scripts** скрипты

**blend4web.lst**, **blend4web\_sdk\_free.lst** и **blend4web\_sdk\_pro.lst** (оpционально)  
списки файлов для сборки дистрибутивов

**check\_resources.py** скрипт для проверки и сообщения о неиспользуемых ресурсах (изображения и звуки, на которые ссылаются экспортимые файлы)

**compile\_b4w.py** скрипт для сборки кода движка и приложений

**converter.py** скрипт, осуществляющий: уменьшение разрешения текстур вдвое, компрессию текстур в формат DDS, конвертацию звуковых файлов в форматы mp4 и ogg

**custom\_json\_encoder.py** форк Python-модуля json, сортирует ключи по алфавиту в обратном порядке

**gen\_glmatrix.sh** скрипт для генерации математического модуля на основе исходных файлов из репозитория glMatrix 2

**graph.sh** генератор текущего графа сцены в формате svg, используется для отладки рендеринга

**make\_dist.py** сборщик дистрибутивов

**memory.sh** скрипт для проверки обычной (RAM) и видео-памяти (VRAM)

**mod\_list.py** скрипт для генерации списка модулей, используемых в приложениях

**plot.sh** построитель графиков отладочной информации

**process\_blend.py** скрипт для автоматического переэкспорта всех сцен из состава SDK

**remove\_alpha\_channel.sh** скрипт для удаления альфа-канала изображения

**screencast.sh** скрипт для записи видео с экрана

**shader\_analyzer.py** скрипт, запускающий локальный веб-сервер, который осуществляет подсчет сложности шейдеров

**translator.py** скрипт для сборки файлов с переводами аддона

**shaders** GLSL-шейдеры движка

**src** основной исходный код ядра движка

**addons** исходный код дополнений движка

**ext** исходный код внешних объявлений, формирующих API движка

**libs** исходный код библиотек

**tools** Различные инструменты для сборки движка, приложений и конвертации ресурсов

**converter\_utils** сборки утилит для конвертации ресурсов

**closure-compiler** компилятор Google Closure, файлы исключений к нему, генераторы файлов исключений

**glsl**

**compiler** компилятор GLSL-шейдеров движка

**pegjs** грамматики парсер-генератора PEG.js для реализации препроцессора GLSL, а также скрипт для генерации модулей парсеров из этих грамматик

**yuicompressor** утилита для сжатия файлов CSS

**uranium** исходный код и скрипты сборки физического движка Uranium (форк Bullet)

**VERSION** содержит текущую версию движка

## 25.12 Загрузка локальных ресурсов

Рендерер движка является Web-приложением, и его работа происходит при просмотре HTML-файла в браузере. После инициализации происходит загрузка ресурсов (сцен, текстур), которая подчиняется [правилу ограничения домена](#), запрещающему, в частности, загрузку из локальной директории.

Начиная с версии 15.02, в состав Blend4Web SDK входит *сервер разработки*, решавший проблему загрузки локальных ресурсов.

## 25.13 Профили качества изображения

Для поддержки различных по функциональности платформ в движке реализовано несколько профилей качества изображения:

- *низкое качество (P\_LOW)* - отключен ряд функций (тени, динамическое отражение, постпроцессинг), размер текстур для сборочной версии уменьшен вдвое, антиалиасинг отключен
- *высокое качество (P\_HIGH)* - используются все запрошенные сценой функции, метод антиалиасинга FXAA
- *максимальное качество (P\_ULTRA)* - вдвое увеличено разрешение рендеринга, увеличено разрешение карт теней, метод антиалиасинга SMAA



Переключение профилей качества осуществляется программно, до инициализации контекста WebGL. Профиль по умолчанию P\_HIGH.

```
var m_cfg = b4w.require("config");
var m_main = b4w.require("main");

m_cfg.set("quality", m_cfg.P_LOW);
m_main.init(...);
```

Разработчики приложений могут также установить параметр **quality** при инициализации движка с использованием дополнения app.js:

```
var m_cfg = b4w.require("config");
var m_app = b4w.require("app");

m_app.init({
    canvas_container_id: "body_id",
    quality: m_cfg.P_HIGH
});
```

---

## Разработчикам движка

---

### 26.1 Стиль оформления кода

В движке применяется структурное программирование. Код организуется в модули. Подходы ООП не используются, классы не определяются, наследование не осуществляется и т.п.

Используется K&R стиль, за исключением того, что открывающая скобка для составного оператора ставится на той же строке, например:

```
function foo_bar() {
    // ...
}

if (a > b) {
    // ...
}
```

Для выравнивания используются 4 пробела (табуляция запрещена).

#### 26.1.1 Примеры

В именах переменных и функций используется знак подчеркивания:

```
var foo_bar = 123; // correct
var fooBar = 123; // wrong
```

Все глобальные переменные начинаются со знака подчеркивания:

```
var _foo_bar = null;
```

Константы пишутся прописными буквами и никогда не начинаются со знака подчеркивания:

```
var FOO_BAR = 100;
```

Для внешних API названия методов и свойств задаются через точку. Поля, требующие защиты от обfuscации, помещаются в специальный тэг `@cc_externs`:

```
exports.FOO_BAR = 123;

exports.foo_bar = function() {

}

/**
 * Set properties.
 * @method module:properties.set_props
 * @param {Object} foo Foo object
 * @cc_externs props_1 props_2
 * @cc_externs props_3 props_4
 */
exports.set_props = function(foo) {

    var bar_1 = foo.props_1;
    var bar_2 = foo.props_2;
    var bar_3 = foo.props_3;
    var bar_4 = foo.props_4;

    ...
}
```

Комментарии только на английском языке. Стиль комментирования - JSDoc.

## 26.2 Сборка движка

Перед сборкой необходимо убедиться, что в системе присутствуют все необходимые зависимости, для чего следует свериться с [таблицей](#).

Для компиляции движка и входящих в SDK приложений достаточно выполнить команду из корневой директории SDK:

```
make compile
```

Полная сборка, включающая конвертацию ресурсов (текстур, звуков и видео), компиляцию и подготовку документации вызывается командой:

```
make build
```

Сборка архивов с дистрибутивами:

```
make dist
```

Все вышеперечисленные операции могут быть выполнены одной командой:

```
make all
```

## 26.3 Сборка аддона

Бинарные сборки аддона Blend4Web подготовлены для следующих платформ: Linux x32/64, OS X x64, Windows x32/64. В то же время пользователи имеют возможность произвести сборку самостоятельно.

Для этого необходимо наличие Python 3.x (желательно, чтобы версия была эквивалентна используемой в Blender) и компилятора языка C (в Linux достаточно установить пакеты python3-dev и build-essential).

**Пути относительно корневой директории SDK:**

- скрипт сборки: `csrc/b4w_bin/build.py`
- аддон Blend4Web: `blender_scripts/addons/blend4web/`

Запуск сборки осуществляется следующим образом:

```
python3 ./csrc/b4w_bin/build.py
```

Результатом сборки будет бинарный файл с именем:

`b4w_bin_[ПЛАТФОРМА]_[АРХИТЕКТУРА].[СТАНДАРТНОЕ_РАСШИРЕНИЕ]`,

размещенный в каталоге с аддоном. Пример: `b4w_bin_Linux_64.so`. После этого аддон станет готовым к использованию на данной платформе.

## 26.4 Зависимости

Для ведения эффективной разработки движка и приложений, необходим ряд сторонних программ (зависимостей). Большинство этих зависимостей находится в составе современных дистрибутивов GNU/Linux, таких как Ubuntu. В других Unix-подобных системах (Apple OS X, FreeBSD) их установка из исходных кодов или иных источников не представляет существенных проблем.

В таблице ниже перечислены все зависимости, в порядке убывания важности для разработки.

Название	Пакет в дистрибутиве Ubuntu 14.04	Назначение
Bash	в составе по умолчанию	интерпретатор скриптов
Python 3	в составе по умолчанию	интерпретатор скриптов
NodeJS	nodejs	компиляция шейдеров
Java	default-jre	компиляция и обfuscация модулей движка
Emscripten	из исходных текстов EMSDK	сборка Uranium
ImageMagick	imagemagick	конвертация ресурсов
NVIDIA Texture Tools	libnvtt-bin	конвертация ресурсов
NVIDIA Cg Toolkit	nvidia-cg-toolkit	отладка шейдеров
Libav	libav-tools	конвертация ресурсов
Gnuplot	gnuplot	отладка
Graphviz	graphviz	отладка
xsel	xsel	отладка
Sphinx	sphinx-doc	сборка документации (HTML-версия)
sphinx-intl	устанавливается с помощью PIP	сборка документации (перевод)
TeX Live	texlive, texlive-latex-extra texlive-lang-cyrillic	сборка документации (PDF-версия)
JSDoc 3	из исходных текстов JSDoc	сборка документации (документация на API)
PEG.js	из исходных текстов PEG.js	препроцессинг шейдеров

## 26.5 Названия функций и переменных

Рекомендуется при создании новых функций и переменных использовать следующие префиксы и суффиксы.

*init\_* создание абстрактного объекта

*create\_* создание конкретного объекта

*update\_* обновить состояние имеющегося объекта

*attach\_ / detach\_* добавить/удалить временное свойство к объекту

*append\_ / remove\_* добавить/удалить временное свойство к уже существующим подобного рода

*insert\_ / pop\_* добавить/удалить элемент массива (доступ по индексу места)

*apply\_ / clear\_* операция с флагом, бинарной величиной или произвольным параметром

*set\_ / get\_* установить/получить значение свойства/переменной

*\_tmp* глобальная переменная - кеш в виде простого объекта (массив, вектор)

cache глобальная переменная - кеш в виде сложного объекта

## 26.6 Отладка

Отладка движка производится с помощью методов модуля `debug.js`.

Структура текущего рендер-графа может быть сохранена в формате DOT с помощью вызова `b4w.debug.scenegraph_to_dot()`, например, в консоли браузера. После вызова данного метода содержимое консоли сохранить в файл с расширением `.gv`. Чтобы получить граф в графическом виде, необходим набор утилит `graphviz`. Преобразование в формат SVG выполняется с помощью вызова:

```
> dot -Tsvg graph.gv -o graph.svg
```

где `graph.gv` имя файла с сохранённым графиком.

## 26.7 Компиляция шейдеров

Используемые в движке шейдеры подвергаются обработке компилятором. Компилятор выполняет 3 основных процедуры:

- валидацию кода шейдеров,
- обfuscацию кода шейдеров,
- оптимизацию кода шейдеров.

Для запуска компиляции требуется выполнить одну из команд в корневой директории SDK:

```
> make compile_shaders
> make verify_shaders
```

- **make compile\_shaders** - проверка, обfuscация, оптимизация и экспорт скомпилированных шейдеров,
- **make verify\_shaders** - проверка, обfuscация и оптимизация.

В процессе компиляции сначала осуществляется синтаксический анализ (парсинг) текста шейдера. Соответствующий парсер создается автоматически на основе грамматики с помощью генератора `PEG.js`. Далее по данным парсинга производится валидация, обfuscация и оптимизация шейдеров, после чего шейдеры экспортируются в виде абстрактного синтаксического дерева (Abstract Syntax Tree, AST) для непосредственной загрузки движком.

Расположение основных файлов в репозитории:

- исходная грамматика - `glsl_utils/pegjs/glsl_parser.pegjs`
- скрипт генерации парсера - `glsl_utils/pegjs/gen_nodejs.sh`
- парсер - `glsl_utils/compiler/glsl_parser.js`

## 26.7.1 Валидация

Компилятор шейдеров выполняет следующие процедуры, связанные с проверкой кода:

- вывод сообщений о неиспользуемых переменных и функциях (dead code),
- проверка синтаксиса шейдеров,
- проверка шейдеров на соответствие import/export-механизма,
- удаление лишних пробелов, переводов строк и повторяющихся символов ”,”.

## 26.7.2 Обфускация

Обфускация служит для сокращения объема и затруднения понимания GLSL-кода. На данный момент в нем реализована следующая процедура:

- замена пользовательских идентификаторов более короткими односимвольными, двухсимвольными и т.д. именами (с поддержкой import/export-механизма).

## 26.7.3 Оптимизация

Оптимизация заключается в выполнении следующих процедур:

- удаление фигурных скобок, которые не несут функциональной нагрузки, но порождают новые области видимости (даный функционал полезен при обработке директив node/lamp),
- внутрифункциональная оптимизация, связанная с использованием малого числа буферных локальных переменных взамен локальных переменных, заданных программистом.

Примером удаления бесполезных фигурных скобок может служить замена кода

```
void function(){
    int a;
    {
        a = 1;
    }
}
```

следующим кодом

```
void function(){
    int a;
    a = 1;
}
```

Использование малого числа буферных локальных переменных заключается в том, что они повторно используются в разных контекстах. Например, следующий код

```
int function(){
    int a = 1;
    int b = a + 3;
    return b;
}
```

будет заменен на

```
int function(){
    int _int_tmp0 = 1;
    _int_tmp0 = _int_tmp0 + 3;
    return _int_tmp0;
}
```

---

**Примечание:** Не производится оптимизация локальных переменных структур и переменных массивов.

---

## 26.7.4 Директивы import/export

В целях упорядочивания, структурирования и повышения удобочитаемости кода шейдеров в include-файлах используются директивы `import` и `export`. Они указываются в начале файла и должны выглядеть примерно следующим образом:

```
#import u_frame_factor u_quatsb u_quatsa u_transb u_transa a_influence
#import qrot

#export skin
```

Директива `#import` определяет набор идентификаторов, которые объявлены вне этого include-файла, но доступны для использования в нем. Имеется ограничение: такие идентификаторы должны быть обязательно объявлены где-либо выше места подключения include-файла.

Директива `#export` определяет набор идентификаторов, доступных для использования вне данного файла. Такие идентификаторы должны быть обязательно объявлены в этом файле.

Таким образом, шейдер, использующий include-файл, обязан до места подключения содержать объявления, необходимые для импорта, а после него может использовать экспортимые идентификаторы.

Идентификаторами могут быть как имена переменных, так и имена функций. По умолчанию при отсутствии директив `import/export` считается, что include-файл не использует внешние объявления и не предоставляет пользование внутренними.

## 26.7.5 Рекомендации и ограничения

В связи с наличием препроцессинга, необходимостью совместной обработки нескольких шейдеров и include-файлов, а также особенностями реализации компилятора

гарантировать работоспособность полученного на выходе кода можно только при соблюдении ряда правил или ограничений на текст исходных шейдеров:

1. Обязательное использование специальной директивы `#var` для описания констант, определяемых движком в момент запуска. Например:

```
#var AU_QUALIFIER uniform
AU_QUALIFIER float a;
```

Синтаксис здесь схож с директивой `#define`. Смысл директивы `#var` в том, чтобы определяемое ею значение позволило распарсить исходный шейдер. Что это будет конкретно (например, ‘uniform’ или ‘attribute’ в примере выше), не важно, т.к. на этом этапе оно все равно неизвестно. Однако, желательно указывать более-менее подходящее описание, а не что-то совершенно произвольное.

---

**Примечание:** Для констант, используемых не в коде шейдера, а в выражениях препроцессинга, директива `#var` не обязательна.

---

2. Использование при необходимости директив `import/export`.
3. Не следует перегружать встроенные функции, только пользовательские.
4. Не следует объявлять переменные с именем одной из встроенных функций, либо `main` (даже если это не приводит к ошибке).
5. Нельзя использовать директивы `#var` и `#define` для замены отдельных символов в таких операторах, как: “`++`”, “`-`”, “`*=`”, “`/=`”, “`+=`”, “`-=`”, “`-=`”, “`<=`”, “`>=`”, “`!=`”, “`&&`”, “`||`”, “`^^`”.

Например:

```
#var EQUAL =
...
a *EQUAL b;
...
```

6. Использование директивы `#include`, не должно приводить к неоднозначности при обfuscации содержимого include-файла. Это может произойти в том случае, когда один и тот же файл включается в несколько разных шейдеров, и в каком-то из них могут повлиять определенные выше директивы, вроде `#var` или `#define`. Также не стоит использовать в include-файле необъявленные функции и переменные.
7. Использование вложенных `include`'ов или множественного включения одного и того же `include`'а в один и тот же шейдер не поддерживается.
8. К неработоспособности шейдера может привести нетривиальное использование препроцессинга, например, создающее невалидный GLSL-код:

```
#if TYPE
void function1() {
#else
void function1(int i) {
#endif
```

```
...  
}
```

9. Не следует объявлять переменные с именами вида `node_[NODE_NAME]_var_[IN_OUT_NODE]`, где `NODE_NAME` — название некоторой ноды, `IN_OUT_NODE` — название одного из входов или выходов ноды.
10. Не разрешается множественное использование одной и той же директивы `#nodes_main`, `#nodes_global` или `#lamps_main` в одном шейдере.
11. Директивы `#nodes_main`, `#nodes_global` и `#lamps_main` рекомендуется использовать в том же файле, в котором содержится описание шейдерных нод, например, в одном и том же `include`-файле - это необходимо для корректной валидации шейдеров.

## 26.7.6 Поддержка расширений WebGL

Работа обfuscатора может зависеть от используемых WebGL-расширений, если они каким-либо образом влияют на шейдерный язык. На данный момент поддерживаются следующие расширения:

- `OES_standard_derivatives`

## 26.7.7 Ошибки компилятора

В случае ошибки компилятор выведет соответствующее сообщение в консоли.

Перечень возможных ошибок:

Сообщение об ошибке	Причина
Error! Ambiguous obfuscation in include file ‘FILE_NAME’.	Ошибка! Неоднозначная обфускация include-файла FILE_NAME.
Error! Bad preprocessing collision while obfuscation identifier: ‘NAME’. Varying/uniform or varying/attribute qualifiers combination. File: ‘FILE_NAME’.	Ошибка в файле FILE_NAME. Невозможность обфускации переменной с именем NAME из-за переопределения при препроцессинге. Переопределение одной и той же переменной с разными квалификаторами. Недопустимые комбинации: varying/uniform, varying/attribute.
Error! Extension NAME is unsupported in obfuscator. File: ‘FILE_NAME’.	Ошибка! WebGL-расширение с именем NAME, использованное в файле FILE_NAME, не поддерживается обфускатором.
Error! Include ‘FILE_NAME’ not found.	Ошибка! При подключении не найден include-файл FILE_NAME.
Error! Undeclared TYPE: ‘NAME’. File: ‘FILE_NAME’.	Ошибка в файле FILE_NAME. Необъявленный идентификатор типа TYPE (переменная, функция, структура, ...) с именем NAME.
Error! Undeclared TYPE: ‘NAME’. Importing data missed. File: ‘FILE_NAME’.	Ошибка! Необъявленный идентификатор типа TYPE (переменная, функция, структура, ...) с именем NAME. Отсутствует объявление идентификатора, требуемого в include-файле FILE_NAME согласно директиве #import.
Error! Undeclared TYPE: ‘NAME’. Possibly exporting needed in include file ‘INCLUDE_NAME’. File: ‘FILE_NAME’.	Ошибка в файле FILE_NAME. Необъявленный идентификатор типа TYPE (переменная, функция, структура, ...) с именем NAME. Возможно требуется разрешить его экспорт в include-файле INCLUDE_NAME.
Error! Undeclared TYPE: ‘NAME’. Possibly importing needed. File: ‘FILE_NAME’.	Ошибка! Необъявленный идентификатор типа TYPE (переменная, функция, структура, ...) с именем NAME. Возможно требуется указать его как импортируемый в include-файле FILE_NAME.
Error! Unused export token ‘NAME’ in include file ‘FILE_NAME’.	Ошибка! В include-файле FILE_NAME разрешен для экспорта необъявленный идентификатор с именем NAME.

Error! Using reserved word in TYPE 'NAME'. File: 'FILE_NAME'.	Ошибка в файле FILE_NAME. Использование зарезервированного слова при объявлении идентификатора типа TYPE (переменная, функция, структура, ...) с именем NAME.
Error! 'all' extension cannot have BEHAVIOR_TYPE behavior. File: 'FILE_NAME'.	Ошибка! Директива #extension, указанная для всех (all) WebGL-расширений в файле FILE_NAME, не поддерживает поведение BEHAVIOR_TYPE.
Syntax Error. ERROR_MESSAGE. File: FILE_NAME, line: LINE_NUMBER, column: COL_NUMBER.	Ошибка синтаксиса в строке LINE_NUMBER, столбце COL_NUMBER при парсинге шейдера FILE_NAME. Исходное описание ошибки приведено в ERROR_MESSAGE. В сообщении прилагается листинг кода в окрестности соответствующей строки (следует учитывать особенность pegjs-парсеров, указывающих чуть далее места, вызвавшего ошибку).
Warning! Function 'NAME' is declared in [include ]file FILE_NAME, but never used.	В файле FILE_NAME объявлена функция NAME, которая нигде не используется.
Warning! Include file 'FILE_NAME' not used in any shader, would be omitted!	Include-файл FILE_NAME не используется ни в одном из шейдеров, поэтому будет исключен из закомпилированной версии.
Warning! Unused import token 'NAME' in include file 'FILE_NAME'.	Идентификатор с именем NAME импортируется в include-файле FILE_NAME, но нигде не используется.
Warning! Variable 'NAME' is declared in include file FILE_NAME, but never used.	В файле FILE_NAME объявлена переменная NAME, которая нигде не используется.

## 26.8 Обновление переводов аддона

При необходимости обновить все существующие .ro файлы, запустите скрипт *translator.py* из директории SDK/scripts без параметра:

```
> python3 translator.py
```

Для обновления существующего .ro файла необходимо вызвать скрипт с передачей ему одного из поддерживаемых языков:

```
> python3 translator.py ru_RU
```

Для просмотра списка поддерживаемых языков вызовите скрипт следующим образом:

```
> python3 translator.py help
```

При вызове скрипта в любом случае будет обновлен файл *empty.po*.

После обновления .po файлы могут быть отредактированы/переведены.

## Работа в команде с использованием Git

---

### 27.1 Общие сведения

Для организации командной работы над проектом может быть использована система контроля версий файлов Git. Использование Git имеет ряд преимуществ перед другими способами организации совместной работы:

- сохранение полной истории изменений файлов с возможностью возврата к предыдущим версиям
- синхронизация изменений между пользователями и автоматическое слияние изменений
- возможность работы с бинарными файлами большого объёма

Git - распределенная система, и каждый разработчик или дизайнер имеет собственный локальный репозиторий (хранилище). Синхронизация между локальными репозиториями может осуществляться через центральное “общее” хранилище, которое можно разместить на специально выделенной для этой цели машине (сервере). К серверу может быть организован доступ по протоколу SSH.

Хотя для Git существует множество графических утилит, упрощающих работу начинающих пользователей, здесь мы рассмотрим работу со штатной консольной утилитой, вызываемой командой `git`.

### 27.2 Типичный рабочий процесс

1. В ходе работы в локальных репозиториях создаются, изменяются или удаляются файлы.
2. По завершении некоторого логического этапа работы возникает необходимость фиксации изменений (коммит) и/или синхронизации с коллегами.
3. Проводится подготовка файлов к коммиту - учет измененных, новых и удаленных файлов, а также сброс изменений.
4. Осуществляется коммит.

- Локальные изменения загружаются в общее хранилище и становятся доступными для коллег.

Далее описывается ограниченный набор команд Git, рекомендуемых к использованию при создании приложений и графических ресурсов.

Перед выполнением команд необходимо перейти в репозиторий, например:

```
> cd ~/blend4web
```

## 27.3 Индивидуальные настройки

Новый пользователь может установить имя и почтовый адрес командами:

```
> git config --global user.name "Ivan Petrov"
> git config --global user.email ipetrov@blend4web.com
```

Установленные данные будут использоваться в логе изменений.

## 27.4 Проверка статуса

Перед началом, в процессе или после выполнения любых операций рекомендуется проверять текущее состояние репозитория.

Проверить статус можно командой:

```
> git status
```

Результат команды `git status`, если все коммиты проведены и нет новых файлов:

```
# On branch master
# Your branch is ahead of 'origin/master' by 2 commits.
#
nothing to commit (working directory clean)
```

Возможный результат команды `git status`, если имеются изменения. Например, файлы `apps_dev/firstperson/firstperson.js` и `doc_src/git_short_manual.rst` изменены, и создан новый файл `123.txt`:

```
# On branch master
# Changes not staged for commit:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working directory)
#
#       modified:   apps_dev/firstperson/firstperson.js
#       modified:   doc_src/git_short_manual.rst
#
# Untracked files:
#   (use "git add <file>..." to include in what will be committed)
```

```
#  
#   123.txt  
no changes added to commit (use "git add" and/or "git commit -a")
```

## 27.5 Перед коммитом

### 27.5.1 Проверка изменений (текстовых файлов)

Перед совершением коммита в случае текстовых файлов рекомендуется просмотреть внесенные изменения.

Проверить, что изменилось, во всей директории:

```
> git diff
```

или только в определенном файле:

```
> git diff apps_dev/firstperson/firstperson.js
```

Возможный результат команды `git diff` для текстового файла:

```
diff --git a/apps_dev/firstperson/firstperson.js b/apps_dev/firstperson/firstperson.js
index 4381c99..44b3b15 100644
--- a/apps_dev/firstperson/firstperson.js
+++ b/apps_dev/firstperson/firstperson.js
@@ -557,8 +557,9 @@ function enable_camera_control_mode() {
        var cam_view_down = CAMERA_MOVE_UPDOWN * (Math.sin(_passed_time) - 1);

        b4w.camera.translate_view(obj, 0, cam_view_down, cam_view_angle);
-    } else
+    } else {
        b4w.camera.translate_view(obj, 0, 0, 0);
+    }
}
```

### 27.5.2 Восстановление файлов

Если файл был изменен или удален, но его необходимо восстановить (до состояния, зафиксированного последним коммитом), следует использовать команду:

```
> git checkout doc_src/git_short_manual.rst
> git checkout 123.txt
```

Внесенные изменения будут отменены, поэтому эту команду необходимо выполнять с осторожностью.

### 27.5.3 Посторонние файлы

Если файл значится в списке **Untracked files** (команда `git status`), но контроль версий для него не нужен, его следует удалить или переместить за пределы рабочей директории.

## 27.6 Подготовка к коммиту

### 27.6.1 Добавление файлов

Если изменения устраивают, добавить нужные измененные и/или новые файлы для коммита:

```
> git add apps_dev/firstperson/firstperson.js  
> git add 123.txt
```

Снова проверить статус:

```
> git status
```

Возможный результат команды `git status` после добавления некоторых файлов командой `git add`:

```
# On branch master  
# Changes to be committed:  
#   (use "git reset HEAD <file>..." to unstage)  
#  
#       new file:   123.txt  
#       modified:  apps_dev/firstperson/firstperson.js  
#  
# Changes not staged for commit:  
#   (use "git add <file>..." to update what will be committed)  
#   (use "git checkout -- <file>..." to discard changes in working directory)  
#  
#       modified:  doc_src/git_short_manual.rst  
#
```

Видно, что для коммита добавлены файлы `apps_dev/firstperson/firstperson.js` и `123.txt`, а файл `doc_src/git_short_manual.rst` остался недобавленным. Для упрощения работы рекомендуется либо добавлять такие файлы для коммита, либо отбрасывать их изменения командой `git checkout`.

### 27.6.2 Удаление файлов

Некоторые файлы могут быть отмечены как удаленные из Git после выполнения команды `git status`, например:

```
# On branch master
# Your branch is ahead of 'origin/master' by 2 commits.
#
# Changes not staged for commit:
#   (use "git add/rm <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working directory)
#
#       deleted:    123.txt
#
no changes added to commit (use "git add" and/or "git commit -a")
```

В таком случае, если удаление файла должно быть зафиксировано (т.е. войти в коммит), выполнить команду `git rm`, например:

```
> git rm 123.txt
```

Если же файл был удален по ошибке, и его необходимо вернуть, нужно использовать команду `git checkout`.

## 27.7 Коммит

Выполнить коммит командой:

```
> git commit
```

Появится окно текстового редактора (например, `nano` или `vim`), в котором нужно ввести комментарий к коммиту на английском языке.

```
GNU nano 2.2.6                                         File: .git/COMMIT_EDITMSG

My commit message
# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
# On branch master
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
#       new file:   123.txt
#       modified:   apps_dev/firstperson/firstperson.js
#
# Changes not staged for commit:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working directory)
#
#       modified:   doc_src/git_short_manual.rst
#
^G Get Help          ^O WriteOut        ^R Read File      ^Y Prev Page
^X Exit             ^J Justify         ^W Where Is       ^V Next Page
```

Сохранить изменения и выйти из редактора (в **nano** Ctrl+O, затем Ctrl+X; в **vim** ZZ, или ESC :wq).

После совершения коммита рекомендуется снова проверить статус. Коммит совершен правильно, если команда `git status` отображает `nothing to commit, working directory clean`.

## 27.8 Синхронизация между репозиториями

### 27.8.1 Из удаленного - в локальный

После того как все коммиты сделаны, необходимо загрузить изменения из удаленного (“общего”) репозитория в локальный:

```
> git pull
```

Результат команды `git pull`, если в удаленном репозитории нет изменений:

```
Already up-to-date.
```

Результат команды `git pull`, если в удаленном репозитории были изменения, и синхронизация прошла успешно:

```
remote: Counting objects: 151, done.
remote: Compressing objects: 100% (101/101), done.
remote: Total 102 (delta 74), reused 0 (delta 0)
Receiving objects: 100% (102/102), 69.77 MiB | 4.87 MiB/s, done.
Resolving deltas: 100% (74/74), completed with 32 local objects.
From liixer:blend4web
    dbf3877..9f9700c  master      -> origin/master
Updating dbf3877..9f9700c
Fast-forward
  apps_dev/firstperson/firstperson.js           |  338 +-+
  .../location_agriculture.blend               | Bin 25601626 -> 25598644 bytes
  ...
  src/controls.js                             |   38 +-+
  src/data.js                                |     5 +
  src/physics.js                            |  185 +-+
19 files changed, 1452 insertions(+), 2767 deletions(-)
create mode 100644    deploy/assets/location_agriculture/textures/rotonda_02_diff.png
```

При желании можно посмотреть, какие изменения были внесены коллегами, командой:

```
> git diff dbf3877..9f9700c
```

Параметр этой команды - в данном случае `dbf3877..9f9700c` - указывает, между какими именно коммитами просматриваются изменения. Этот параметр удобно выделить в результатах команды `git pull` и вставить щелчком мыши (средняя кнопка) в консоли в нужном месте.

Также можно просмотреть лог изменений:

```
> git log
```

Команда `git pull` не всегда приводит в успешной синхронизации. Результат команды `git pull` в случае наличия конфликтов:

```
remote: Counting objects: 11, done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 6 (delta 5), reused 0 (delta 0)
Unpacking objects: 100% (6/6), done.
From liixer:blend4web
  ff715c2..dbf316a  master      -> origin/master
warning: Cannot merge binary files: blender/landscape_objects/Fallen_tree.blend (...)
Auto-merging blender/landscape_objects/Fallen_tree.blend
CONFLICT (content): Merge conflict in blender/landscape_objects/Fallen_tree.blend
Automatic merge failed; fix conflicts and then commit the result.
```

Порядок действий при возникновении конфликтов описан далее.

### 27.8.2 Из локального - в удаленный

Затем нужно загрузить изменения из локального репозитория в удаленный (“общий”), чтобы локальные изменения стали доступными для коллег.

```
> git push
```

Результат команды `git push`, если в удаленном репозитории уже есть все локальные изменения:

```
Everything up-to-date
```

Результат команды `git push`, если синхронизация прошла успешно:

```
Counting objects: 25, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (14/14), done.
Writing objects: 100% (14/14), 1.23 KiB, done.
Total 14 (delta 11), reused 0 (delta 0)
To gftteam@lixer:blend4web.git
  9f9700c..fa1d6ac  master -> master
```

Результат команды `git push`, если синхронизация не прошла, потому что сначала не была выполнена команда `git pull`:

```
To gftteam@lixer:blend4web.git
 ! [rejected]      master -> master (non-fast-forward)
error: failed to push some refs to 'gftteam@lixer:blend4web.git'
To prevent you from losing history, non-fast-forward updates were rejected
Merge the remote changes (e.g. 'git pull') before pushing again. See the
'Note about fast-forwards' section of 'git push --help' for details.
```

Необходимо выполнить команду `git pull`.

Изменения, загруженные в центральный репозиторий, могут быть получены другими участниками разработки с помощью команды `git pull`.

## 27.9 Разрешение конфликтов

### 27.9.1 Общие сведения

Конфликты синхронизации происходят, если выполнены оба условия

1. один и тот же файл был изменен как в локальном, так и в удаленном репозитории, и
2. автоматическое слияние изменений не произошло, поскольку изменения находятся в одном и том же месте файла.

Типичные случаи:

1. бинарный файл (текстура, blend-файл) независимо изменен двумя участниками разработки
2. в текстовой файл в одной и той же строке были внесены разные изменения
3. один участник разработки изменил файл, а другой - переместил его и т.п.

Хотя конфликты синхронизации - нормальное явление, слишком частое их возникновение замедляет работу. Рекомендуется ставить коллег в известность о начале работ с общими бинарными файлами, а также чаще проводить синхронизацию. Необходимо эффективно распределять работу между участниками разработки, чтобы таких общих файлов было как можно меньше. Этого можно добиться, в частности, подключением всех ресурсов сцены (*linking*) из отдельных blend-файлов в один мастер-файл.

### 27.9.2 Порядок действий

Не рекомендуется производить какие-либо действия с файлами (изменять, удалять), пока репозиторий находится в конфликтном состоянии.

Первое что необходимо сделать - выполнить команду `git status`.

```
# On branch master
# Your branch and 'origin/master' have diverged,
# and have 7 and 1 different commit each, respectively.
#
# Unmerged paths:
#   (use "git add/rm <file>..." as appropriate to mark resolution)
#
#       both modified:    blender/landscape_objects/Fallen_tree.blend
#
no changes added to commit (use "git add" and/or "git commit -a")
```

Список конфликтующих файлов отображен в разделе `Unmerged paths`.

Дальнейший порядок действий различен для бинарных и текстовых файлов.

### 27.9.3 Бинарные файлы

На данном этапе конфликтующие бинарные файлы находятся в том состоянии, в котором они находились в локальном репозитории до попытки синхронизации. Файлы полностью функциональны (например, открываются графическими редакторами).

В случае конфликта бинарных файлов необходимо выяснить с коллегами или самостоятельно, какую из версий оставить, а какую отбросить. Выбор осуществляется командой `git checkout`.

Выбрать локальную версию файла (`- --ours`). Его можно открыть и убедиться в этом.

```
> git checkout --ours blender/landscape_objects/Fallen_tree.blend
```

Выбрать удаленную версию файла (`- --theirs`). Его можно открыть и убедиться в этом.

```
> git checkout --theirs blender/landscape_objects/Fallen_tree.blend
```

Снова выбрать локальную версию файла (`- --ours`).

```
> git checkout --ours blender/landscape_objects/Fallen_tree.blend
```

В итоге необходимо остановиться на нужной версии файла. При угрозе потери работы можно сохранить отбрасываемую версию файла вне репозитория.

### 27.9.4 Текстовые файлы

На данном этапе в конфликтующие текстовые файлы Git'ом вносятся как локальные, так и удаленные изменения одновременно, в особом формате. Такие текстовые файлы как правило, не работоспособны.

Пример. Один участник разработки изменил имя сцены с “Blue Lizard” на “Green Lizard” в файле приложения и загрузил изменения в центральный репозиторий. Другой участник разработки изменил в той же строке “Blue Lizard” на “Red Lizard”, совершил коммит и выполнил команду `git pull`. В результате именно на этого участника ложится ответственность по разрешению конфликта. В его файле приложения будут находиться строки:

```
<<<<< HEAD
      "name": "Red Lizard",
=====
      "name": "Green Lizard",
>>>>> 81bf4e2d5610d500ad4d2a2605ee7e61f759f201
```

В случае конфликта текстовых файлов можно поступить следующим образом. Файлы, содержащие исходный код, необходимо отредактировать с учетом или без учета внесенных обеими сторонами изменений. В то же время экспортированные текстовые файлы сцен (заканчивающиеся на **.json**) проще повторно экспортовать.

### 27.9.5 Корректирующий коммит

После выбора нужных файлов или редактирования изменений, добавить их для коммита:

```
> git add blender/landscape_objects/Fallen_tree.blend
> git status
```

Возможный результат выполнения `git status` после добавления конфликтующих файлов для коммита:

```
# On branch master
# Your branch and 'origin/master' have diverged,
# and have 7 and 1 different commit each, respectively.
#
nothing to commit (working directory clean)
```

Выполнить коммит, комментарий рекомендуется оставить предложенный по умолчанию:

```
> git commit
> git status
```

```
# On branch master
# Your branch is ahead of 'origin/master' by 8 commits.
#
nothing to commit (working directory clean)
```

Конфликты разрешены, изменения из удаленного репозитория успешно применены в локальном репозитории. Теперь изменения в локальном репозитории, - включающие только что разрешенный конфликт, - можно загрузить в удаленный репозиторий командой `git push`.

## 27.10 Тэги

Тэги (метки) предназначены для указания на определенный коммит, например, с целью обозначения стабилизированной версии продукта.

Просмотреть список тэгов:

```
> git tag
```

Создать тэг для релиза от 3 июня 2013 г., указывающий на коммит со стабильной версией проекта:

```
> git tag R130603 67bb597f7ed1643ed0220d57e894f28662e614e5
```

Просмотреть информацию о коммите тэга:

```
> git show --shortstat R130603
```

Перейти к тэгу...

```
> git checkout R130603
```

...и вернуться:

```
> git checkout master
```

Синхронизировать тэги с удаленным репозиторием:

```
> git push --tags
```

Удалить тэг (при ошибочном создании):

```
> git tag -d R130603
```

## 27.11 Другие полезные команды

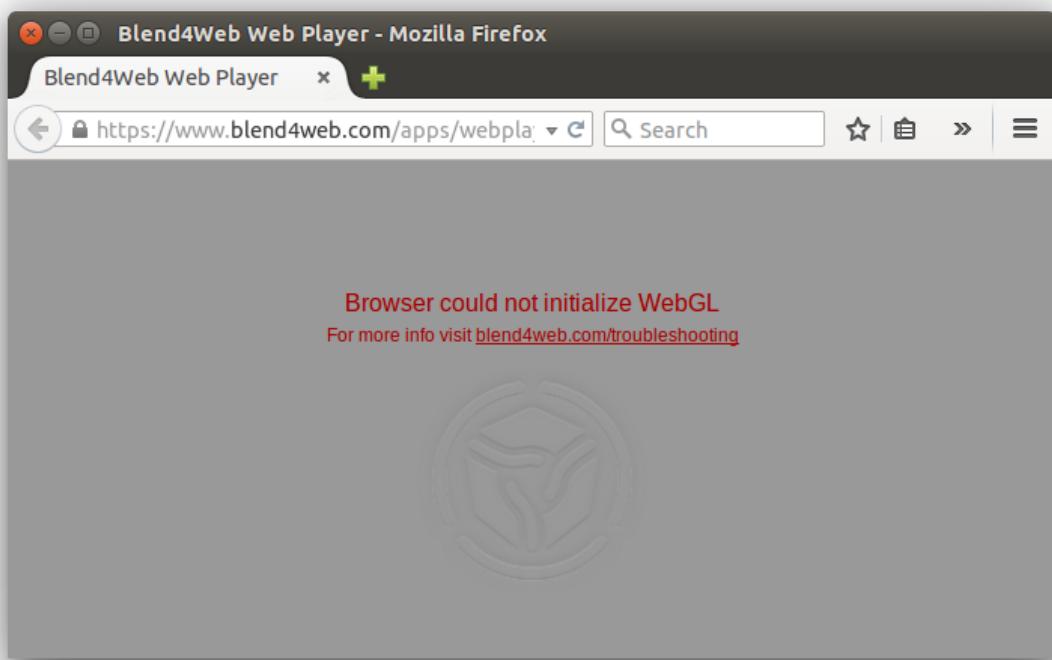
Просмотреть лог за январь 2012 г, показывать имена файлов, без коммитов слияния:

```
> git log --after={2012-01-01} --before={2012-01-31} --name-only --no-merges
```

## Проблемы и решения

### 28.1 Проблемы при запуске движка

1. Появляется сообщение “*Browser could not initialize WebGL.*”



Следует выполнить действия, описанные в разделе *Ошибка инициализации WebGL*.

2. Видны элементы интерфейса или пустой экран, но сцена не отображается. При этом тестовый сайт <http://get.webgl.org/> и другие WebGL приложения работают корректно.

Вероятные причины:

- Не используется локальный веб-сервер или браузер не настроен для работы с локальными ресурсами. См. раздел *Загрузка локальных ресурсов*.

- Файлы ресурсов, которые пытается загрузить движок, были перемещены или удалены.
- Используются старые версии драйверов.
- Используются открытые драйвера, не обеспечивающие поддержку WebGL.

Для пользователей Linux - ввиду неполной реализации OpenGL стека в драйверах с открытым кодом в настоящий момент рекомендуется использовать проприетарные драйверы текущей версии для графических процессоров Nvidia и AMD.

- Используется устаревшая операционная система, такая как Windows XP.

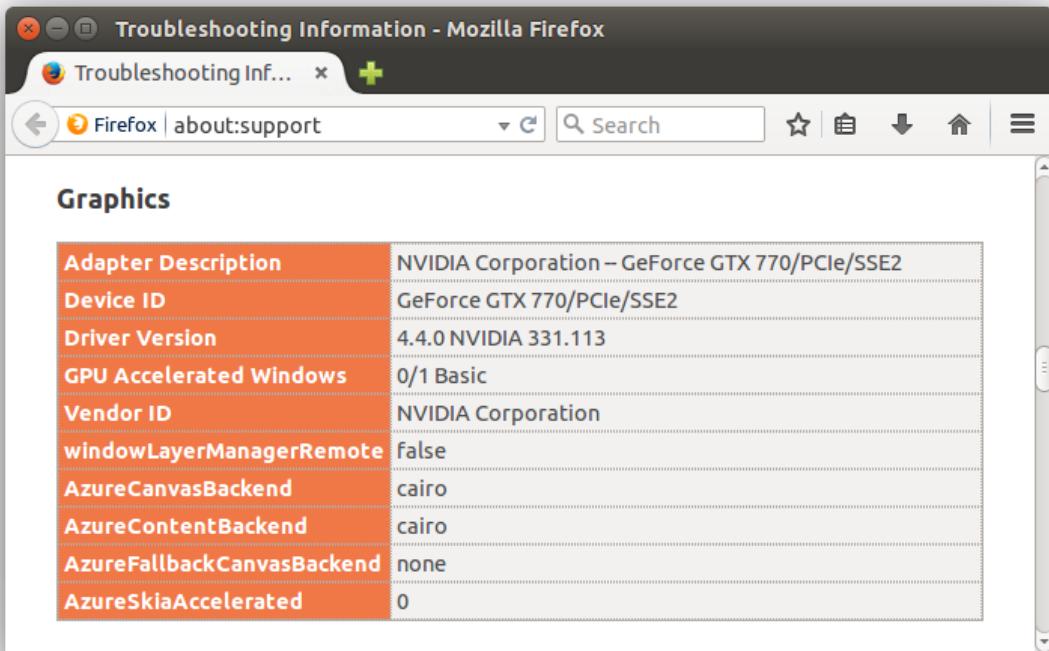
## 28.2 Ошибка инициализации WebGL

Сайт <http://get.webgl.org/> при просмотре в браузерах Chrome или Firefox последней версии сообщает о проблемах. Что делать?

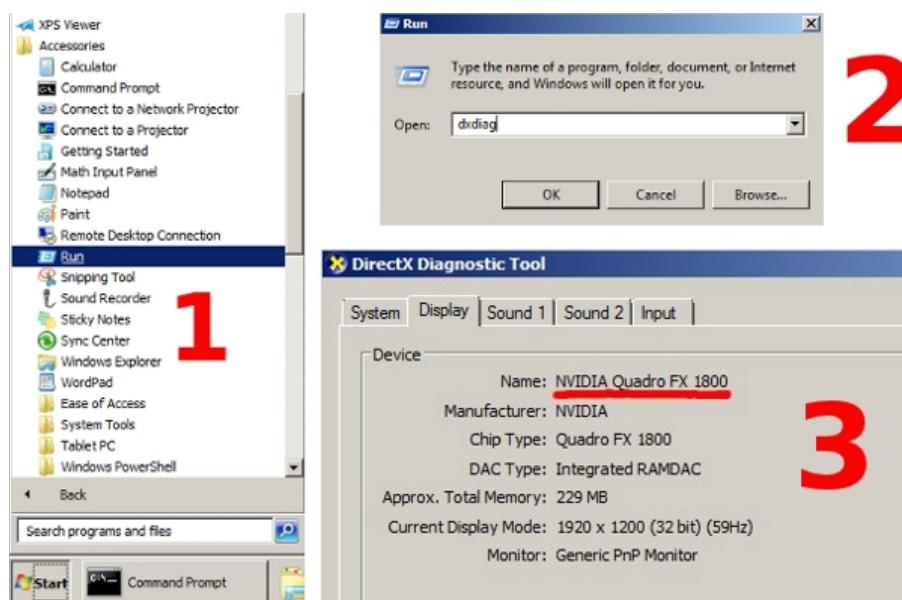
1. Установить доступные обновления для системы (для Windows см. [инструкцию](#)). В случае Windows установить последнюю версию [DirectX](#). Перезагрузить систему.
2. Рекомендуется проводить своевременное обновление драйверов для графических карт. Чтобы определить тип и производителя карты, можно ввести **about:gpu** (или **chrome://gpu**) в адресную строку браузера Chrome...

Driver Information	
<b>Initialization time</b>	66
<b>Sandboxed</b>	true
<b>GPU0</b>	VENDOR = 0x10de, DEVICE= 0x1184
<b>Optimus</b>	false
<b>AMD switchable</b>	false
<b>Driver vendor</b>	NVIDIA
<b>Driver version</b>	331.113
<b>Driver date</b>	
<b>Pixel shader version</b>	4.40
<b>Vertex shader version</b>	4.40
<b>Machine model name</b>	
<b>Machine model version</b>	
<b>GL_VENDOR</b>	NVIDIA Corporation
<b>GL_RENDERER</b>	GeForce GTX 770/PCIe/SSE2

или Firefox...



Для операционных систем семейства Windows можно воспользоваться средством диагностики DirectX dxdiag.

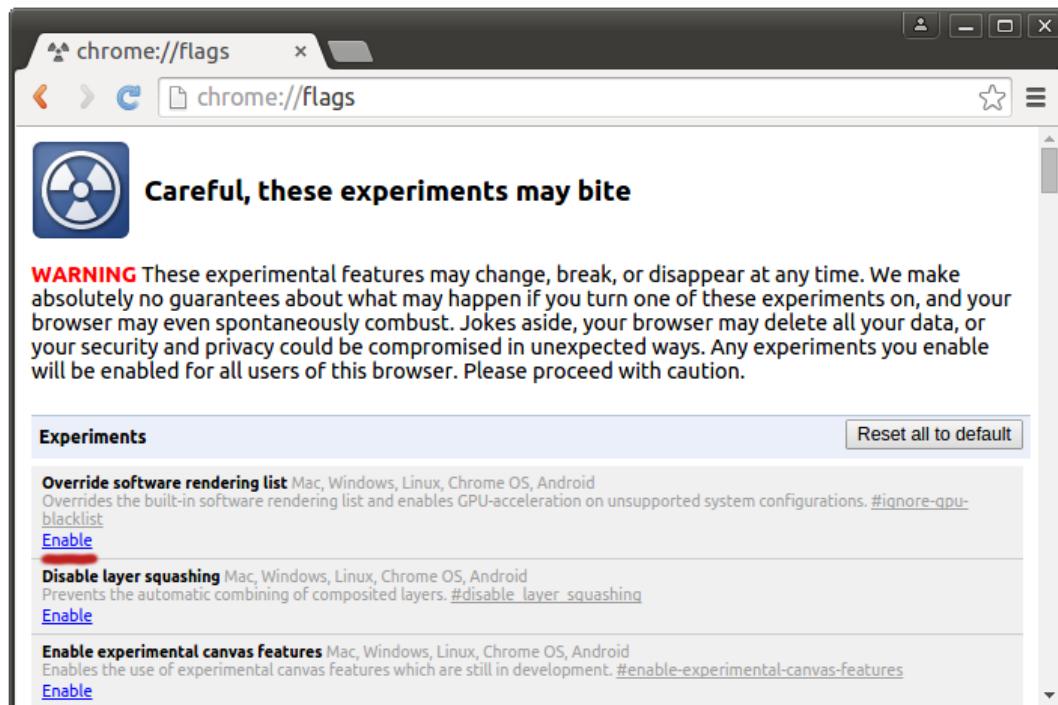


Необходимо загрузить драйверы с соответствующего центра поддержки (например, Intel, Nvidia, AMD/ATI). После установки драйверов перезагрузить систему.

3. Если в результате вышеперечисленных действий инициализировать рендеринг не удается (или нет возможности обновить систему), можно попробовать изменить настройки браузера.

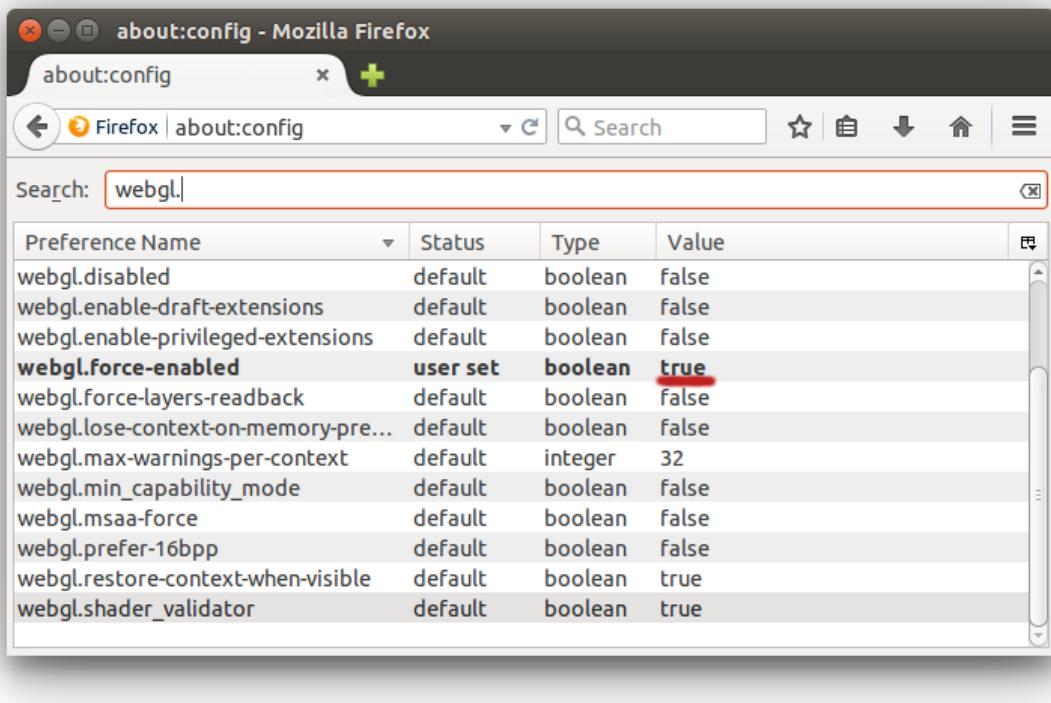
*B Chrome:*

Ввести **about:flags** (или **chrome://flags**) в адресную строку браузера, нажать Включить (Enable) под опцией Переопределение списка программного рендеринга (Override software rendering list) и перезапустить браузер.



*B Firefox:*

Ввести **about:config** в адресную строку браузера, найти параметр **webgl.force-enabled** и переключить его двойным щелчком мыши из **false** в **true**.



## Замечания к релизам

---

### 29.1 v15.10

#### 29.1.1 Новый функционал

- Новый функционал в системе управления проектами.

Проекты теперь можно экспортить и импортировать. Это существенно упрощает обновление на новые версии SDK, обеспечивает возможность запускать и отлаживать проекты на различных системах, а также позволяет обмениваться проектами с другими разработчиками.

Стали доступны два новых типа проектов: `WebPlayer JSON` и `WebPlayer HTML`. Обе опции позволяют создавать проекты, не требующие программирования и работающие внутри приложения `WebPlayer`.

Для удобства навигации по собственным проектам была добавлена функция сортировки по имени, а также скрытия проектов, изначально присутствующих в составе SDK.

Добавлена возможность удалять проекты из списка.

Улучшена поддержка ОС Windows, благодаря включению Java в состав SDK, что позволяет собирать приложения без необходимости установки внешних зависимостей.

- Улучшения антиалиасинга.

На панель `Render > Anti-Aliasing` добавлено поле `AA Quality`, позволяющее выбирать профили качества [антиалиасинга](#). Улучшена технология антиалиасинга; теперь в качестве технологии антиалиасинга используется алгоритм, основанный на FXAA 3.11, что позволило увеличить качество и производительность, особенно при выборе высоких уровней детализации графики.

- Новый функционал редактора логики:

- Новая нода `Move Camera`.

Позволяет задать положение и целевую точку для камеры.
  - Новая нода `Play Sound`.

Позволяет проигрывать звук выбранного спикера.
  - Новая нода `Switch Select`.

Подобна ноде `Select`, но обладает большей гибкостью и удобством, позволяя выбрать серию объектов.
  - В ноду `Math Operation` редактора логики добавлена операция `Random`, позволяющая генерировать случайные числа.
  - В ноду `Send Request` добавлена возможность отправки `POST` запроса с заданными параметрами.
- Добавлена возможность использования типа маппинга `Repeat` для NПОТ-текстур.

Теперь доступен тип маппинга `Repeat` для NПОТ-текстур (текстур, размеры которых не являются степенью числа 2). Также для этих текстур теперь применяется МПР-текстурирование.
  - Добавлено автоматическое изменение размеров текстур при превышении ими лимитов WebGL.

В случае, когда размер текстуры превышает лимиты WebGL, будет использовано уменьшение размера текстуры до максимально допустимого (кроме случая с использованием DDS-текстур).
  - Добавлена кнопка `Fast Preview` для быстрого предпросмотра сцены.

Кнопка находится на панели `Render > Development Server`.
  - Поддержка анимации свойств `Intensity` и `Color` для источников освещения.

Теперь возможно анимировать интенсивность и цвет источников освещения как в обычной, так и в NLA-анимации.

## 29.1.2 Изменения

- Улучшен интерфейс системы управления проектами.

Изменена тема оформления, добавлена кнопка для быстрого запуска менеджера проектов в `Blender Development Server > Project Manager`.
- Улучшена работа звуковых источников.

Метод `is_play()` теперь корректно и с минимальными задержками отображает факт завершения воспроизведения.
- Добавлено несколько сообщений об ошибках экспорта.

При экспорте ряда объектов их тип может смениться на `EMPTY`, например, в случае вырожденной геометрии или отсутствия звукового файла на объекте типа `SPEAKER`. Сообщения об этом теперь выводятся в консоль браузера.

- Улучшено отображение селектора объектов для нод редактора логики.
- Изменено поведение гамма-коррекции в нодовых материалах.

В связи с изменениями в Blender 2.76 необходимо иначе проводить *гамма-коррекцию в нодовых материалах*.

- Изменения API.

Объявлен устаревшим и будет удален в последующих релизах метод `resize` из модуля `main`. Вместо него следует использовать соответствующий метод `resize` из модуля `container`.

### 29.1.3 Исправления

- Исправлены ошибки в системе управления проектами.
- Повышена стабильность работы на мобильных платформах при большом количестве лампочек на сцене.
- Повышена стабильность работы на Unix-подобных системах.

Улучшена работа движка на Unix-подобных системах, использующих видеокарты AMD и открытые видеодрайверы.

- Исправлена ошибка, связанная с добавлением аннотаций на сцену.

Исправлена ошибка, возникающая при добавлении аннотаций на сцену без свойства *Object -> Meta Tags*.

- Исправлена ошибка, связанная с некорректным воспроизведением звуков при переключении вкладок браузера.
- Исправлен ряд ошибок, связанных с работой видеотекстур.
- Исправлена ошибка, проявляющаяся при использовании пустой нодовой группы в нодовом материале.
- Исправлена ошибка, если на системе частиц в качестве частицы был выбран объект, имеющий LOD-объекты.
- В редакторе логики исправлена ошибка ноды `Delay`, связанная с использованием переменной в качестве параметра.
- В редакторе логики исправлена ошибка, возникающая при удалении `Entry Point`.
- В редакторе логики исправлена ошибка дублирования переменных в выпадающем списке выбора переменных.

- Исправлена ошибка, возникающая в экспортированной сцене, использующей логическую ноду `Play Animation` и отключенный флаг `NLA`.
- Исправлено и установлено поведение аналогичное Blender для следующих нод нодовых материалов: `Math->Power`, `Gamma`.

### 29.1.4 Известные проблемы

- Проблемы с обновлением старых версий аддона.

Рекомендуется перезагрузка Blender после обновления аддона.

- Драйверы NVIDIA версии 331 в Linux могут приводить к программным ошибкам WebGL.
- Изменение фильтрации текстур на некоторых устройствах.

При использовании *iPad* или браузера *Internet Explorer* отключена некорректная фильтрация текстур, которые используются материалами с типом прозрачности *Alpha Clip*.

- Некорректное поведение Depth-текстуры на ОС Windows.

При использовании стабильной версии Google Chrome 32bit на ОС Windows могут возникать проблемы с поддержкой Depth-текстуры. В настоящее время эта проблема исправлена для бета-версии браузера.

- Некоторые устройства, использующие GPU Mali, требуют принудительного включения WebGL в настройках браузера.
- На системах Apple OS X и Blender 2.76 для работы локального сервера разработки может потребоваться установка [Python 3.4](#). Это связано с ошибкой Blender <https://developer.blender.org/T46623>.

## 29.2 v15.09

### 29.2.1 Новый функционал

- Система управления проектами.

Новая система управления проектами позволяет:

- просматривать список и внутреннюю структуру имеющихся проектов
- запускать приложения, просматривать сцены в программе-просмотрщике, загружать исходные файлы сцен в Blender
- создавать и конфигурировать новые приложения, в том числе на основе готовых шаблонов

- осуществлять сборку приложения и его конверсию в вид, необходимый для последующего размещения на сервере
- конвертировать ресурсы приложения (текстуры, звуки и видео) в альтернативные форматы
- производить автоматический реэкспорт всех сцен из состава приложения, включая json и html файлы

Управлять проектами возможно двумя способами: с помощью консольной утилиты *project.py*, имеющей простой формат вызова и подробную справку, либо в графической форме с помощью веб-приложения, выполняющегося на сервере разработчика. Сервер не требует каких-либо дополнительных настроек и, также как и раньше, запускается с помощью кнопки *Open SDK* в Blender.

Система управления проектами работает на всех операционных системах, однако для некоторых операций может потребоваться установка дополнительных приложений. Подробнее о работе и настройке системы можно узнать в соответствующем разделе документации [для разработчиков приложений](#).

- Управление костями арматурного объекта.

Существенно расширены возможности для управления арматурными объектами.

Добавлен новый модуль `armature`, в котором присутствуют методы для получения и назначения позиций костей в координатах объекта арматуры: `get_bone_tsr()`, `set_bone_tsr()` и в относительных координатах кости: `get_bone_tsr_rel()`, `set_bone_tsr_rel()`. С помощью данных функций возможно создавать приложения с программным управлением элементами скелета. Например, теперь возможно процедурно анимировать персонажей или упрощенно назначать позиции элементов сложных конструкций, которые имеют в качестве управляющего каркаса арматуру.

Обеспечена поддержка ограничителей типа `COPY_TRANSFORMS` для костей. Кости могут копировать положение любых объектов на сцене, в том числе и физических. Таким образом можно например применить технику Ragdoll, чтобы получить физически корректные скелеты.

- Новые ноды редактора логики.

- Play Animation: позволяет проигрывать анимацию объекта.
- Send Request: позволяет отправить HTTP GET запрос на указанный адрес и осуществить простейший разбор ответа.
- Inherit Material: позволяет скопировать атрибуты материала с одного объекта на другой.
- Set Shader Node Param: позволяет изменить значение шейдерного нода; в данный момент поддерживаются нода Value и RGB.

- Delay: позволяет осуществить задержку перед выполнением следующей ноды.
- Apply Shape Key: позволяет выставить значение shape key для объекта.
- Outline: позволяет управлять “обводкой” объекта.
- Для видеотекстур добавлена настройка `Allow NLA`, позволяющая выбирать использование NLA-анимации индивидуально для каждой текстуры.
- Добавлен новый функционал в модуле *material*.

Добавлены методы `set_specular_color_factor` и `get_specular_color_factor`.

- Расширение функционала веб-приложения для разработчиков.

На корневой страницы веб-приложения для разработчиков появился новый раздел *Tools*, включающий инструменты *Manage Projects* (управление проектами) и *WebGL Report* (просмотр возможностей WebGL браузера).

- Улучшения в приложении *Web player*.

Добавлен необязательный атрибут `alpha`, использующийся для установки прозрачности области отрисовки. Добавлено поведение *Outline on Select*.

- В утилите *project.py* добавлен тип сборки приложения *update*.

Позволяет обновить движок в директории приложения без компиляции самого приложения. Опция используется в простых приложениях и уроках.

- Добавлен новый функционал в модуль *app*.

В методе `enable_camera_controls()` добавлен необязательный параметр `disable_zoom`. Данный параметр позволяет отключить приближение камеры.

В модуле *app* был добавлен метод `queue_animate()`.

- Добавлен новый функционал в модуль *scenes*.

В модуле *scenes* был добавлен метод `has_picking_subs()`.

- Добавлен новый функционал в модуль *debug*.

В модуле *debug* был добавлен метод `fake_load()`.

## 29.2.2 Изменения

- Аддоны (такие как *app*, *mouse* и другие) теперь являются частью скомпилированных версий движка:
  - *b4w.min.js* - расширенная (advanced) оптимизация (ранее *b4w.full.min.js*)

- b4w.simple.min.js - простая (simple) оптимизация
- b4w.whitespace.min.js - оптимизация пробелов, используемых в коде (whitespace)

Выбор нужной версии определяется исходя из настроек *компиляции приложения*.

- Изменения нодового редактора логики:
  - Переименованы ноды: *Play* -> *Play Timeline*; *Select & Jump* -> *Select*, *Register Store* -> *Variable Store*.
  - Помимо переменных-регистров теперь можно создавать пользовательские переменные, имеющие произвольные имена.
- В утилите *project.py* удален малоиспользуемый тип сборки движка *combine*.
- Улучшена документация по *конвертеру ресурсов*.
- Улучшена документация по *переводу аддона*.
- Изменен экспорт UV-слоев и вертексных цветов.

Теперь поведение UV-слоев и вертексных цветов после экспорта еще больше приближено к Blender'у.

- Расширена поддержка NLA-анимации для видеотекстур, поведение которых теперь более приближено к поведению в Blender'e.
- Добавлено *сообщение экспорта* об отсутствии в анимации каналов.
- Добавлена некритическая ошибка экспорта при выборе неподдерживаемого типа *Render Type* в системе частиц.
- При наследовании материала (метод API *inherit\_material*) теперь наследуется параметр *Specular Color Factor*.
- Изменения API.

Следующие методы объявлены устаревшими и будут удалены в последующих релизах: *is\_camera*, *is\_mesh*, *is\_armature*.

Вместо них следует использовать соответствующие методы модуля *objects*: *is\_camera*, *is\_mesh*, *is\_armature*.

Также добавлены новые методы: *is\_speaker*, *is\_lamp* и *is\_empty*.

Метод *get\_object\_by\_dupli\_name\_list* теперь возвращает *null*, если параметр *name\_list* неточно задан. Добавлен метод *get\_object\_name\_hierarchy* для получения полного списка имен с учетом дублирования объектов; возвращаемое этой

функцией значение будет валидным параметром для функции `get_object_by_dupli_name_list`.

В методе `set_debug_params` параметр `wireframe_mode`, ранее бывший строковым, теперь должен быть одной из констант: `WM_NONE`, `WM_OPAQUE_WIREFRAME`, `WM_TRANSPARENT_WIREFRAME`, `WM_FRONT_BACK_VIEW`, `WM_DEBUG_SPHERES`.

- Изменения поведения NLA-анимации для динамически загружаемых сцен.

Если на таких сценах присутствуют объекты с NLA-анимацией, то теперь они подчиняются настройкам NLA главной сцены. Ранее такая анимация не поддерживалась.

- Продолжен рефакторинг внутренней структуры объектов, начатый в предыдущем релизе.
- Добавлена ошибка экспорта.

Теперь при использовании объекта, с типом отличным от “Mesh”, в качестве частицы типа “Hair”, будет возникать *некритическая ошибка экспорта*.

- В исходные файлы дистрибутива добавлена информация о лицензировании.
- Теперь при экспорте сцены слоты с картами окружения (Environment Map) содержащими видеотекстуры не экспортируются.

Добавлена *некритическая ошибка экспорта*, говорящая о невозможности использования видео в качестве карты окружения.

### 29.2.3 Исправления

- Опция `Render Above All` теперь корректно отображается на нодовых материалах.
- Исправлена функция `remove_object()`, некорректно очищавшая массив объектов.
- Улучшена стабильность работы частиц типа “Hair”.

Исправлена ошибка, возникающая при использовании объекта с модификатором в качестве частицы типа “Hair”.

- Улучшена поддержка Windows Phone.
- Исправлена ошибка, возникающая при копировании физических объектов.
- Исправлена ошибка реализации текстурных координат типа `Orcos` в случае, когда объект имеет нулевой размер по какой-либо из осей координат.
- Исправлена ошибка движка, когда в системе частиц в качестве частицы был выбран объект, имеющий физические настройки.

- Исправлена ошибка, при которой NLA-анимация могла начинаться не с нулевого кадра.
- Исправлено поведение ноды **Lamp Data**: ранее могла не обновляться информация об источниках освещения в процессе загрузки сцены.
- Исправлена ошибка расчета выхода **Normal** в ноде **Geometry** для задней поверхности полигона.
- Исправлена ошибка расчета выхода **Orco** в ноде **Geometry**, появляющаяся при смещении меша относительно его центра в Blender'e.
- Поправлен расчет конечного кадра NLA-анимации для видеотекстур.
- Исправлена ошибка движка, связанная с выставлением различных комбинаций настроек выделения (*selection*) и подсвечивания (*outlining*) объектов.
- Улучшена стабильность эффекта **Wind bending**.
- Исправлены ошибки рендеринга **Alpha Clip** материалов.
- Исправлен перенос текстуры спекуляра (**Specular Map**) при наследования материала.
- Исправлена ошибка связанная с источниками освещения в приложениях с несколькими сценами.

#### 29.2.4 Известные проблемы

- Проблемы с обновлением старых версий аддона.  
Рекомендуется перезагрузка Blender после обновления аддона.
- Драйверы NVIDIA версии 331 в Linux могут приводить к программным ошибкам WebGL.
- Изменение фильтрации текстур на некоторых устройствах.  
При использовании *iPad* или браузера *Internet Explorer* отключена некорректная фильтрация текстур, которые используются материалами с типом прозрачности **Alpha Clip**.
- Некоторые устройства, использующие GPU Mali, требуют принудительного включения WebGL в настройках браузера.

### 29.3 v15.08

#### 29.3.1 Новый функционал

- Нодовый редактор логики.

Нодовый редактор логики позволяет добавлять интерактивность в приложение в более удобной для восприятия форме. Использовавшийся ранее для этих целей слотовый инструмент NLA Script удалён, при этом старые скрипты, созданные при помощи NLA Script, автоматически конвертируются в новый формат при открытии blend-файла.

- Поддержка различных моделей освещения для нод MATERIAL и MATERIAL\_EXT.

Теперь модель диффузного и спекулярного освещения для каждой ноды будет соответствовать настройкам выбранного в ней материала, в то время как раньше тип освещения был одинаков для всех и соответствовал настройкам самого нодового материала, содержащего данные ноды. Данное нововведение позволяет осуществлять смешиваемые различных стековых материалов (за исключением текстурированных) в составе нодового.

- Улучшения в API для перемещения объектов.

В модуле `transform` были добавлены новые методы для перемещения объектов относительно родительских: `set_translation_rel()`, `set_translation_rel_v()`, `get_translation_rel()`, `set_rotation_rel()`, `set_rotation_rel_v()`, `get_rotation_rel()`, `set_scale_rel()`, `get_scale_rel()`, `set_tsr_rel()`, `get_tsr_rel()`.

Методы без префикса `_rel` теперь служат исключительно для перемещения объектов в мировой системе координат, независимо от того, являются ли они дочерними по отношению к родительским объектам или нет.

Также были добавлены методы для осуществления локальных поворотов относительно выбранных осей: `rotate_x_local()`, `rotate_y_local()`, `rotate_z_local()`.

- Улучшена поддержка ОС Windows.

Добавлена поддержка *конвертера ресурсов* для ОС Windows. Теперь пользователи этой операционной системы смогут создавать кросбраузерные приложения, использующие медиаресурсы. Также была добавлена поддержка Windows в *сборщике приложений*.

- Поддержка работы движка в браузере Microsoft Edge.
  - Поддержка симуляции физики в основном потоке выполнения.
- Данная функция служит для устранения больших задержек расчёта физики в браузерах мобильных устройств. Ранее симуляция физики всегда производилась в отдельных Worker-процессах. Функция управляется параметром инициализации движка `physics_use_workers`.
- Добавлена возможность пересчета ограничивающих поверхностей (bounding box).

В модуле `objects` добавлен метод `update_boundings()`, осуществляющий пересчет ограничивающих поверхностей (параллелепипед, сфе-

ра, эллипсоид, цилиндр, конус, капсула).

- Улучшения в документации.

Выбрана новая тема оформления документации, что позволило существенно улучшить её читаемость на экранах, имеющих различные разрешения.

Существенно расширена и дополнена документация *для разработчиков приложений*.

- Добавлен новый функционал в модуле *camera\_anim*.

В модуле *camera\_anim* были добавлены методы *stop\_cam\_moving()*, *stop\_cam\_rotating()*.

- Добавлена проверка бинарной совместимости.

Теперь проверяется совместимость .bin-файла с .json-файлом и текущей версией движка.

### 29.3.2 Изменения

- Изменения в интерфейсе редактора материалов.

Шейдерные ноды Blend4Web перенесены из **Group** в новый пункт **Blend4Web**. Исправлена ошибка, связанная с их дублированием в интерфейсе.

- Рефакторинг объектов.

Начат рефакторинг представления объектов Blender'a в движке. Объекты теперь будут более строго типизированными, а также будут избавлены от лишних данных, что в итоге приведет к повышению производительности движка.

- Изменения API.

Следующие методы объявлены устаревшими и будут удалены в последующих релизах: *get\_object\_dg\_parent()* (модуль *scenes*), *get\_parent()* (модуль *constraints*). Вместо них теперь следует использовать методы модуля *objects*: *get\_parent()* и *get\_dg\_parent()*.

- Улучшено взаимодействие с сервером нескольких открытых одновременно экземпляров Blender'a.

### 29.3.3 Исправления

- В утилите *project.py* исправлена ошибка компиляции проекта во внешнюю директорию.
- Устранена ошибка работы локального сервера, возникающая на ОС Windows при наличии поврежденной записи в реестре.

- Устранена проблема зависания экспорта при наличии поврежденных данных в .blend-файле.
- Исправлен экспорт метаобъектов. Теперь больше не выводятся сообщения о неуниформном масштабе объекта.
- Исправлена ошибка режима **Auto View** в приложении **Viewer** при наличии пустой категории сцен.
- Исправлена ошибка рендеринга динамически загруженных материалов, имеющих одинаковое имя.
- Исправлена ошибка экспорта при наличии в сцене констрайнта **Copy Transforms**.
- Исправлена ошибка, возникающая при использовании depth-текстур, в браузере Microsoft Edge.
- Исправлен баг, связанный с отсутствием отклика при использовании touch-скрина на некоторых устройствах в браузере Microsoft Edge.

### 29.3.4 Известные проблемы

- В редакторе логики при зацикливании нод часть соединений подкрашивается красным цветом. Данный дефект никак не сказывается на работе инструмента и может быть проигнорирован.

## 29.4 v15.07

### 29.4.1 Новый функционал

- Реализована поддержка типа прозрачности **Alpha Clip** для нодового материала.

Теперь маска прозрачности на материале **Alpha Clip** может быть задана процедурным образом, используя логику нодовых блоков.

- Реализован эффект мягких частиц.

Системы частиц типа **Emitter** получили новое свойство **Soft Particles**, находящееся на панели **Render**. При его включении частицы сглаживают края вблизи непрозрачных объектов. Данный эффект работает только для материалов с прозрачностью типа **Alpha Sort**, **Alpha Blend** или **Add**.

- Новый препроцессор шейдерных инструкций.

Новый препроцессор имеет более простую структуру и обладает лучшей производительностью, что приводит к уменьшению общего времени загрузки сцены, особенно при использовании множества различных шейдеров.

- Существенно улучшена производительность приложения *Viewer*.

Устраниены проблемы, приводящие к замедлению работы просмотри-  
щика, связанные с обработкой событий в библиотеке jQuery Mobile.
- Добавлена возможность выбора нулевого уровня для камеры типа *HOVER*.

Данный параметр представляет собой Z-координату опорной плоско-  
сти, в которой лежит точка вращения камеры.
- Новый тип множества сенсоров (Sensor Manifold) - **CT\_CHANGE**

Наряду с типами **CT\_CONTINOUS**, **CT\_TRIGGER**, **CT\_SHOT** и **CT\_LEVEL** по-  
явился новый тип множества **CT\_CHANGE**, который предполагает вы-  
зов обработчика событий при любом изменении значений сенсоров,  
составляющих данное множество.
- Добавлен новый функционал в модуле *camera\_anim*.

В модуле *camera\_anim* были добавлены методы  
`move_camera_to_point()`, `rotate_camera()`, `is_moving()` и  
`is_rotating()`.
- Реализована поддержка возможности привязки материалов к объектам.

Ранее при назначении пользователя материала **Object** возникала  
ошибка движения. Теперь данная ситуация корректно обрабатывается.
- Реализована возможность обработки внешних запросов для локального сервера разработчика.

Для включения этого функционала необходимо в настройках аддона  
проставить свойство *Enable external requests*.
- В модуле *time* был добавлен метод `clear_animation()`.
- Новые методы API камеры.

API модуля *camera* расширено методами `get_vertical_limits()` и  
`has_vertical_limits()` для получения и проверки наличия вертикалъ-  
ных лимитов поворота и перемещения.
- Улучшения документации.

Новый дизайн документации, оптимизированный для всех видов  
устройств.

Добавлено *описание* различий систем координат Blender'a и  
Blend4Web.
- Поддержка мультиязычности аддона.

Это позволит перевести аддон на различные языки. Был осуществлен  
частичный перевод аддона на русский язык.

## 29.4.2 Изменения

- Многочисленные изменения в интерфейсе аддона, нацеленные на повышение удобства использования.
- Новая опция `force_container_ratio` при инициализации приложения (в методе `init()`).
- Упрощенное включение каустики на воде.

Каустика на воде теперь подключается с помощью флага `Caustics` на панели материала `Water`. Текстуры типа `Voronoi`, использовавшиеся для этого ранее, не поддерживаются.

- Новый способ расчёта сенсоров, связанных с движением мыши.

Теперь любые элементы, входящие в контейнер элемента `Canvas` (например якори), не блокируют обработку событий мыши, что позволяет избежать задержек при движении камеры.

- Добавлена возможность изменения качества отражений.

Теперь при использовании отражений имеется возможность выбрать одну из трех степеней качества: `LOW`, `MEDIUM` или `HIGH`.

- В утилиту `project.py` добавлены новые свойства `--project` и `--app`.
- Новые API анимации.

Добавлены функции `set_first_frame()`/`set_last_frame()`, упрощающие выставление текущего кадра на начало и конец анимации.

- Изменено поведение плоских отражений.

Теперь в случае, когда в поле `Reflection Plane` не указана нормаль к плоскости отражения, больше не возникает ошибка экспорта. По умолчанию берется локальная ось Z объекта.

- Изменено поведение SSAO.

Теперь для использования эффекта постобработки SSAO объекту больше не надо быть генератором тени. Достаточно ее принимать.

- Изменения настроек параметров анизотропной фильтрации текстуры.

Теперь в настройках анизотропной фильтрации текстуры присутствует значение `Default`. Выбор этого значения приводит к использованию настроек анизотропной фильтрации со сцены. Для отключения анизотропной фильтрации текстур следует отключить анизотропную фильтрацию для сцены.

- Поведение прозрачности для спекулярной модели шейдинга соответствует Blender'у.
- Размер маски и группы, назначаемых на материалах, участвующих в расчёте физики, увеличен с 8 до 16.

- Теперь в версии движка для разработчиков шейдеры загружаются асинхронно.
- Удаление модуля *shaders*.

Функции этого модуля теперь доступны из модуля `debug`.

- Скрипт упрощенной пакетной обработки экспортов json и html файлов.

Скрипты `reexporter.py` и `resaver.py` были объединены в один скрипт `process_blend.py`, который может вызываться с соответствующими флагами для обработки json и html файлов.

- Мелкие улучшения и исправления в документации разработчика.

### 29.4.3 Исправления

- Исправлена ошибка неправильных размеров элемента Canvas на начальном этапе загрузки.

- Исправлен баг освещения, когда у объекта выставлен отрицательный *Scale*.

Добавлена обработка этой ситуации с выводом сообщения в консоль.

- Исправлена ошибка применения вертексной анимации к объекту.

- Исправлена ошибка анимации при её обратном воспроизведении.

- Исправлена ошибка с некорректным состоянием сенсоров типа *Collision* и *Ray* после удаления объектов со сцены.

- Корректная обработка *Environment* текстур имеющих размер, превышающий максимально поддерживаемый на данной конфигурации.

Теперь подобные текстуры не отключаются, а отрисовываются со сниженным разрешением. Для видеокарт серии NVIDIA GeForce 8000 на Windows Chrome искусственно снижается разрешение.

- Динамически загруженные сцены больше не выдают сообщения об отсутствующей камере или мире.

- Исправлена ошибка аддона при старте, проявлявшаяся при отсутствии мира на сцене.

- Исправлен баг в приложении *Webplayer*, когда была недоступна кнопка включения/отключения звука на некоторых сценах.

- Исправлен баг, возникающий при использовании *motion blur*, без использования других эффектов постобработки.

- Исправлена работа метода `get_material_extended_params()` модуля `material`.

- Исправлено падение Firefox Mobile при использовании теней, улучшена стабильность работы браузера.

- Улучшена стабильность реэкспортера.

## 29.4.4 Известные проблемы

- Проблемы с обновлением старых версий аддона.

Аддоны с версией меньше 15.06 требуется удалять перед обновлением на более свежие версии, особенно это касается систем Windows. Удаление старой версии должно сопровождаться перезагрузкой Blender.

- Драйверы NVIDIA версии 331 в Linux могут приводить к программным ошибкам WebGL.
- Версии Google Chrome 43.x под Android имеют баг с рендерингом видеотекстур.

Баг уже исправлен в бета-версии и должен исчезнуть с будущими обновлениями браузера.

- Проблемы с использованием источников звука типа `Background Music` при экспорте в HTML.

В настоящее время в браузере Google Chrome имеется баг ([Issue 511251](#)), при котором возникает ошибка с источником аудиофайла. Во избежание проблем с воспроизведением аудиозаписи, при использовании этого браузера на аудиофайл навешивается атрибут `crossOrigin`.

- Изменение фильтрации текстур на некоторых устройствах.

При использовании *iPad* или браузера *Internet Explorer* отключена некорректная фильтрация текстур, которые используются материалами с типом прозрачности *Alpha Clip*.

## 29.5 v15.06

### 29.5.1 Новый функционал

- Новый интерфейс аддона.

Интерфейс аддона был полностью переработан, для чего был создан специализированный профиль *Blend4Web*, при переключении в который все панели содержат только те опции, которые поддерживаются движком. Кроме того, на смену многострочным панелям Blend4Web пришли небольшие тематические разделы, в которых значительно проще ориентироваться.

Кроме того, реализована возможность автоматически определять необходимость включения эффектов, настраиваемых на сцене. В частности, тени, эффекты преломления, эффекты свечения *Glow* и *Outline* в настройках рендера имеют опцию *AUTO*, позволяющую автоматически активировать их при наличии на сцене соответствующих объектов и материалов.

- Новый редактор нормалей.

Редактор нормалей был полностью переработан. Теперь он совместим с нативным хранилищем редактируемых нормалей Blender. Редактор имеет более удобный интерфейс пользователя, а также позволяет редактировать раздельные (split) нормали.

- Поддержка новых нод.

Были поддержаны недостающие выходы ноды *Geometry: Orco* и *Local*. Начата работа по поддержке нод *RGB Curves*, *Vector Curves*, *ColorRamp*, а также всех нод движка Cycles.

- Поддержка новых моделей шейдинга.

Были поддержаны модели *Minnaert* и *Toon* для диффузного шейдинга и *Blinn* для спекулярного. Таким образом, начиная с этого релиза движком поддерживаются все без исключения модели шейдинга Blender.

- Поддержка Blender 2.75.

- Улучшения в физической подсистеме.

Переработан код определения столкновений. Теперь имеется возможность определить объекты, участвующие в столкновениях, а также позицию и нормаль точки контакта.

В системе трассировки лучей расширены возможности API, в частности, возможно указание опции автоматической очистки связанного в трассировкой объекта (теста), а также функция трассировки через несколько объектов, расположенных один за другим. Как и в случае с API для определения столкновений, теперь возможно определить позицию и нормаль точки контакта. Сама трассировка лучей может производиться от точки до точки в глобальном пространстве, без участия вспомогательных объектов-источников.

Расширены возможности физических сенсоров *Collision* и *Ray*.

Реализована возможность удаления физических объектов со сцены, а также пересчёт тестов столкновений и трассировки лучей после добавления или удаления физических объектов.

- Инструмент для автоматического переэкспорта множества сцен.

В аддон добавлен инструмент *Mass Reexporter*, позволяющий осуществить автоматический переэкпорт сцен из указанного списка директорий.

- Возможность автоматической проверки обновлений.

Автоматическая проверка новой версии Blend4Web будет осуществляться при включенном флагке *Check for Updates on Startup* в настройках аддона.

- API для управления параметрами эффекта постобработки *Motion Blur*.

В модуле `scenes.js` были добавлены функции `get_mb_params()` и `set_mb_params()` для управления эффектом *Motion Blur*.

- Поддержка *Timeline*-маркеров.

В модуль `scenes.js` была добавлена функция `marker_frame()`, которая возвращает значение кадра в маркере, расположенному на шкале времени Blender.

- Добавлен новый функционал в модуле `nla.js`.

В модуле `nla.js` были добавлены функции `set_range()`, `reset_range()`, `set_cyclic()` и `clear_callback()`. Так же появилась возможность передать callback в функцию `play()`.

- Появилось API для изменения разрешения Canvas'a.

Для изменения разрешения элемента Canvas достаточно вызвать функцию `set()` из модуля `config.js`, в которую необходимо передать параметры `canvas_resolution_factor` и `value`, где `value` - новое значения фактора разрешения Canvas'a. Данный функционал удобно использовать, когда требуется повышенное разрешение скриншота.

- Поддержка опции *Vertex Groups* → *Length* для системы частиц типа *Hair*.
- Новый формат документации разработчика.

Разработана новая система формальных типов данных, используемых в целях документирования. Например, если раньше трёхмерный вектор имел в документации тип `Float32Array`, теперь он имеет формальный тип `Vec3`, унаследованный от `Float32Array`. Данное нововведение позволяет более точно описывать используемые в движке типы, что позволяет создавать более надёжный и качественный код приложений.

- Новые API для работы с цветом.

Для работы с цветом реализованы два новых модуля: `rgb` и `rgba`. Они содержат API для создания и конвертации цветов из различных цветовых профилей.

- Автоматическое определение пути к SDK в аддоне.

Путь к SDK автоматически заполняется если аддон подключен из директории SDK `blender_scripts`.

- API для корректного расчета 2D-координат над Canvas'ом.

При работе с позицией курсора или точкой касания на touch-устройстве движку требуются корректно рассчитанные координаты над Canvas-элементом.

Тонкости расчета и случаи, когда их необходимо учесть, описаны в соответствующем *разделе* документации.

В связи с этим в модуле `container.js` добавлены методы `client_to_canvas_coords()`, `set_canvas_offsets()`, `update_canvas_offsets()` и `force_offsets_updating()`. Также добавлена настройка конфигурации приложения `track_container_position`.

- API для изменения плавности движения камеры.

В аддоне `mouse.js` добавлены методы `set_plock_smooth_factor()` и `get_plock_smooth_factor()`, позволяющие управлять плавностью движения камеры при использовании режима Pointer Lock. При отсутствии этого режима можно воспользоваться аналогичными методами `set_camera_smooth_factor()` и `get_camera_smooth_factor()` аддона `app.js`.

- Поддержка иконки favicon в просмотрщике Webplayer.

## 29.5.2 Изменения

- Новый формат API для определения столкновений и трассировки лучей.

Функции `append_collision_test()` и `append_ray_test()`, а также `create_collision_sensor()` и `create_ray_sensor()` имеют новый формат вызова, несовместимый с предыдущими версиями движка. Разработчикам настоятельно рекомендуется ознакомится с новыми API и внести соответствующие правки в приложения.

- Изменения рендеринга сцен в текстуру.

Теперь доступен циклический рендеринг сцен в текстуру, например, когда две сцены рендерят друг друга. Основным условием является то, что имеется хотя бы одна основная сцена, которую никто не рендерит.

- Отключение опции `Apply Default Animation` при наличии нелинейной анимации у объекта.

В случае использования объектом нелинейной анимации, на объекте отключается опция `Apply Default Animation`, которая приводила к конфликту с другими типами анимации.

- Изменение поведения лампы типа *Hemi*.

В случае использования лампы типа *Hemi*, для объектов будет использоваться шейдинговая модель *Lambert*, что соответствует поведению Blender.

- Поддержка экспорта в HTML-формат для видеотекстур в Firefox.

Поскольку в Firefox, начиная с 38.0 версии, устранена ошибка, связанная с видеофайлами, записанными в base64, видеотекстуры больше не будут отключаться при экспорте в HTML.

- Использование UV-слоев в движке приближено к поведению Blender.

При отсутствии явно указанных UV-слоёв в нодовом материале, они определяются автоматически, в соответствии с поведением Blender.

- Улучшена стабильность видеотекстуры при HTML-экспорте.
- Оптимизация рендеринга теней при использовании CSM-схемы.
- Оптимизация этапа рендеринга глубины в шейдерном конвейере.
- Оптимизация рендеринга billboard-объектов.
- Параметр конфигурации приложения `resolution_factor` переименован в `render_resolution_factor`.
- Улучшена поддержка объекта-эмиттера системы частиц, имеющего несколько материалов.

Распределение частиц по мешу эмиттера, а также *наследование вертексного цвета* теперь всегда происходит с учетом каждого материала и соответствующей ему части меша.

- Изменено наследование свойства `Wind Bending` на системе частиц.

При выставлении данного свойства в положение `Instance` у объекта-эмиттера теперь не отключается собственный “Wind Bending”.

- Обновление сообщений об ошибках несовместимости версий аддона и движка.

*Описание ошибок* в документации.

- Удалены API sensor locks.

Из модуля `controls.js` удалены неиспользуемые API, связанные с блокировкой сенсоров: *sensor locks*.

- Поддержка поведения Blender'a для нодового материала при отсутствии ноды `Output`.

### 29.5.3 Исправления

- Исправлена работа аддона `screengrabs.js`.

Была исправлена ошибка, приводящая к невозможности сделать скриншот.

- Исправлена работа функции `set_frame()` из модуля `nla.js`.

Была исправлена погрешность выставления кадров функцией `set_frame()`.

- Улучшена стабильность экспортатора.

- Исправлена ошибка удаления/обновления аддона на Windows.

Переработана система загрузки бинарной библиотеки.

- Исправлен ошибки освещения от источников типа `SPOT/POINT`.

- Исправлено некорректное определение экранных координат методами `get_coords_x()` и `get_coords_y()` аддона `mouse.js`.
- Исправлен расчет alpha-канала у эффекта *Outline*.
- Исправлена ошибка эффекта *Wind Bending*.
- Исправлена ошибка системы частиц, при которой не учитывался *Scale* самой частицы.
- Исправлена ошибка, связанная с нарушением синхронизации анимации системы частиц типа *EMITTER*.
- Исправлен баг наложения теней на billboard-объекты.
- Исправлен некорректный экспорт настроек *Override Mesh Boundings*.
- Исправлен баг рендеринга billboard-объектов на iPad.

## 29.5.4 Известные проблемы

- Проблемы с обновлением старых версий аддона.

Аддоны с версией меньше 15.06 требуется удалять перед обновлением на более свежие версии, особенно это касается систем Windows. Удаление старой версии должно сопровождаться перезагрузкой Blender.

- Драйверы NVIDIA версии 331 в Linux могут приводить к программным ошибкам WebGL.
- Версии Google Chrome 43.x под Android имеют баг с рендерингом видеотекстур.

Баг уже исправлен в бета-версии и должен исчезнуть с будущими обновлениями браузера.

## 29.6 v15.05

### 29.6.1 Новый функционал

- Эффект свечения (*Glow*).

Реализован *эффект* возникновения ореола вокруг светящихся объектов, возникающий вследствие рассеивания света в атмосфере и внутри человеческого глаза.

- Автоматический запуск локального сервера разработки.

В настройках аддона появилась опция, включающая автоматический запуск *локального сервера разработки* при старте программы Blender. Функционал позволяет запускать разрабатываемые веб-приложения в браузере без проведения какой-либо подготовительной работы.

- *Кубические отражения*

Помимо плоских отражений теперь возможно настроить также и кубические отражения. Для этого в настройках объекта при включенном свойстве **Reflective** появилась опция **Reflection Type**, где можно выбрать тип отражений **Cube**.

- *Поддержка опций нелинейной анимации.*

Добавлена поддержка имеющегося в Blender инструментария для манипулирования полосами NLA: **Scale** (масштабирование), **Muted** (сделать неактивной), **Reversed** (проигрывать в обратном направлении) и **Repeat** (повторять). Данные опции позволяют существенно расширить инструментарий разработчиков интерактивных сцен.

Кроме того, для управления нелинейной анимацией через API был создан новый модуль **nla.js**. В него входят такие методы как **play()**, **stop()**, **get\_frame()**, **set\_frame()**, позволяющие проигрывать и останавливать нелинейную анимацию, а также получать и выставлять текущий кадр. Новые методы были задействованы в приложении **Viewer** для реализации интерфейса панели управления **NLA**.

- *Расширение возможностей отображения текстур неба.*

Добавлена поддержка параметров вкладки **Influence** для настройки отображения текстуры неба. К таким параметрам относятся: **Blend**, **Horizon**, **Zenith Up**, **Zenith Down**, **Blend**, **Negative**, **RGB to Intensity**, **DVar**.

- *Добавлена обработка соединений входов с выходами разного типа в нодовом материале.*

Теперь в нодовых материалах разрешено соединять входы одного типа с выходами другого типа в виде *некритической ошибки экспорта*. В движке для данного случая реализовано поведение аналогичное Blender Internal Render.

- *Добавлена поддержка опций на панели Rotation для системы частиц типа Hair.*

Теперь положение частиц, заданное в Blender, полностью воспроизводится движком. В частности, теперь поддерживаются не только позиционирование и масштабирование, но и поворот частиц.

- *Добавлены примеры для демонстрации возможностей постпроцессинговых эффектов.*

Примеры подготовлены для следующих эффектов: Bloom, Depth of Field, God Rays, Motion Blur и SSAO.

- *Добавлен новый модуль container.js.*

С помощью методов модуля **container.js** можно добавлять DOM-элементы дерева на определенную глубину относительно **canvas** элемента. Для этого используется метод **insert\_to\_container()**. Реа-

лизованный метод позволил полностью отказаться от использования CSS-свойства `z-index`, поскольку глубина расположения элементов теперь определяется исключительно их позицией в контейнерном элементе.

- *Улучшения в физическом движке.*

Добавлена поддержка свойства `Margin` для физических объектов и материалов. Данная настройка позволяет улучшить стабильность симуляций столкновений объектов. Физический движок Bullet обновлен до версии 2.83.

- *API для изменения режима управления камерой.*

Добавлены методы `изменения режима управления` камерой. Смену режимов управления демонстрирует пример “Camera Move Styles” в приложении `Code Snippets`. Также в модуль `camera.js` добавлен метод `set_hover_pivot()`, позволяющий перемещать опорную точку для камеры типа `HOVER`.

## 29.6.2 Изменения

- *Переработан список сцен из состава SDK.*

Все сцены в приложении `Viewer` были рассортированы по группам: *App* - готовые приложения, *Demos* - демонстрационные сцены-примеры, *Tutorial Exports* - исходные файлы уроков.

- *Добавлена подсветка синтаксиса в приложение `Code Snippets`.*

Помимо появления подсветки синтаксиса, немного изменился дизайн самого приложения `Code Snippets`, а также улучшена его работа на экранах с низким разрешением.

- *Изменено управление видео и canvas-текстурами через API.*

В методах управления этими текстурами появился параметр `data_id`, соответствующий номеру динамически загружаемой сцены.

- *Изменилось поведение при превышении максимального числа анимационных костей.*

Теперь при превышении максимального числа костей скелетная анимация отключается. Ранее это приводило к ошибке компиляции шейдера и нарушению работы приложения.

- *Некоторые свойства систем частиц были переименованы и теперь отключены по умолчанию.*

В частности, по умолчанию теперь отключены свойства `Randomize Location And Size` и `Randomize Initial Rotation` для систем частиц типа `Hair`.

- *В некоторых браузерах отключен эффект Допплера для источников звука.*

Поддержка эффекта Допплера в Web Audio объявлена устаревшей и будет удалена в браузере Chrome начиная с версии 45. В остальных браузерах данный функционал по-прежнему доступен.

- *Изменено поведение объектов при наличии одновременно скелетной и вертекской анимации.*

При наличии у объекта арматурного модификатора и вертекской анимации одновременно, арматурный модификатор экспортиться не будет.

- *Оптимизирован рендеринг систем частиц с процедурной анимацией (эффект Wind Bending).*
- *Улучшена обработка ошибки загрузки главного json- и bin-файлов сцены.*
- *32-разрядный addon для Windows собран без использования кросскомпиляции на нативной системе.*

Данное нововведение улучшает совместимость аддона с указанными системами.

### 29.6.3 Исправления

- *Исправлена ошибка, приводящая к установке неправильной высоты элемента с описанием в модуле “anchors.js”.*
- *Поддержка работы в Epiphany и других браузерах, работающих на движке WebKit.*

Обеспечена посредством исправления блоков кода, по-разному обрабатывающихся различными движками JavaScript.

- *Отключено использование жестов для touch-устройств на базе Microsoft Windows при использовании Internet Explorer 11.*

Раньше использование жестов (Windows Touch Gestures) в данной конфигурации приводило к нежелательному масштабированию и перемещению HTML-элементов. Ожидается, что корректная работа жестов будет обеспечена в будущих версиях браузера.

- *Исправлена ошибка запекания вертекской анимации при наличии анимационного скелета.*
- *Исправлена ошибка рендеринга billboard-объектов на iPad.*
- *Исправлена NLA анимация нодовых материалов, используемая на нескольких объектах.*
- *Исправлен баг эффекта Motion Blur.*

## 29.6.4 Известные проблемы

- *Проблемы с обновлением аддона в Blender.*

Настоятельно не рекомендуется устанавливать новую версию аддона, не удалив предварительно старой, особенно это касается систем Windows. Удаление старой версии должно сопровождаться перезагрузкой Blender.

## 29.7 v15.04

### 29.7.1 Новый функционал

- *Добавлена поддержка ключей деформации (Shape Keys).*

Была осуществлена поддержка *ключей деформации* (Shape Keys) в Blender'е. Для применения ключей в модуль `geometry.js` была добавлена функция `set_shape_key_value`. Пример использования данного функционала доступен в приложении *Code Snippets*.

- *Добавлена поддержка фона с использованием Horizon Color и Zenith Color.*

Теперь поддерживается настройка фона сцены с использованием *Horizon Color* и *Zenith Color*, а также флагов *Paper Sky*, *Blend Sky* и *Real Sky*.

- *Добавлена поддержка ноды Gamma.*

Силами команды Blend4Web в Blender v2.74 была добавлена поддержка ноды *Gamma*. Теперь эта нода также доступна и в движке.

- *Улучшения в инструменте для создания подписей Anchors.*

В настройках в Blender добавлена возможность ограничивать размеры подписей. Реализована поддержка динамической загрузки/выгрузки подписей. Добавлена возможность сокрытия подписей с помощью штатных методов `show()`/`hide()` или в инструменте NLA Script.

- *Оптимизация шейдеров.*

Улучшение шейдерного компилятора. Добавлен следующий функционал: оптимизация локальных переменных, удаление избыточных фигурных скобок. Улучшение производительности нодовых материалов.

- *Оптимизация физического движка *uranium.js*.*

Модуль с физическим движком загружается только в тех случаях, когда это необходимо. Размер модуля сокращён на 20%.

- *Расширены инструменты для отладки физики.*

Добавлен метод `physics_stat()` модуля `debug` для вывода статистики о количестве физических объектов (по типам), сложности геометрии и прочей информации. Для визуальной оценки производительности физической симуляции добавлена возможность вывода числа итераций: Physics FPS (активируется в модуле `config`).

- *Новый API для присоединения объектов к камере независимо от соотношений сторон и углов обзора камеры.*

Реализуется с помощью метода `append_stiff_viewport()` модуля `constraints`.

- *Новый модуль для применения трансформаций “tsr.js”.*

Позволяет применять произвольные трансформации к объектам с помощью универсальных векторов TSR, включающих в себя перемещение, масштабирование и поворот. Данные векторы могут быть использованы в качестве более удобной и эффективной замены матриц.

- *Добавлена возможность игнорирования директорий для конвертера ресурсов.*

Была добавлена возможность игнорирования директорий для `конвертера ресурсов`. Для этого надо разместить в этой директории файл с именем `.b4w_no_conv`.

## 29.7.2 Изменения

- *Изменен внешний вид документации API движка. Появились ссылки для быстрого перехода к описаниям методов и свойств.*
- *Скелетная анимация учитывает взаимное расположение арматуры и объекта.*

Теперь поддерживается родное поведение Blender. Больше нет необходимости размещать арматуру и анимируемый объект в одной точке и с одинаковым поворотом и масштабом.

- *Объектная анимация теперь позволяет анимировать независимо перемещение, поворот и масштаб.*

В объектной анимации теперь нет жесткой простановки ключей на все каналы, что позволяет сохранять изначальные значения в незатронутых каналах или изменять их через API.

- *Опциональная поддержка фона.*

Активация фона или неба регулируется опцией `Sky Settings > Render Sky` на вкладке `World`. По умолчанию отключено.

- *Физический движок Uranium теперь состоит из двух модулей.*

Для работы физики теперь требуется два модуля: код движка `uranium.js` и файл инициализации памяти `uranium.js.mem`. Оба модуля должны находиться в одной директории.

- Эффект *Glow* и все связанные с ним компоненты движка были переименованы в *Outline*.

Новое имя более ясно описывает суть эффекта: подсвечивание объекта на его границах.

- Применение ограничителя движения при помощи функции “*append\_semi\_stiff\_cmt*” модуля “*constraints.js*” теперь разрешено только для камера типа “*EYE*”.
- Изменения локального сервера разработки в *Blender*.

В качестве локального сервера теперь вместо стандартного SimpleHTTPServer на Python используется Tornado Web Server. Новый сервер имеет высокую производительность, а также содержит расширенные настройки для отключения кеширования браузерами.

- Добавлено управление слайдерами с помощью клавиатуры в приложении *Viewer*.

Теперь управление слайдером осуществляется при помощи клавиш < и >.

- Изменения в методе “*update\_object\_animation*” модуля “*animation.js*”.

Добавлен необязательный параметр “*force\_update*”, заставляющий обновлять состояние анимированного объекта даже при остановленной анимации.

- Изменения API модуля *mouse.js*.

Добавлены методы: *enable\_mouse\_hover\_outline*, *disable\_mouse\_hover\_outline*.

Следующие методы объявлены устаревшими и будут удалены в последующих релизах: *enable\_mouse\_hover\_glow*, *disable\_mouse\_hover\_glow*.

- Изменения API модуля *scenes.js*.

Добавлены методы: *outlining\_is\_enabled*, *set\_outline\_intensity*, *get\_outline\_intensity*, *apply\_outline\_anim*, *apply\_outline\_anim\_def*, *clear\_outline\_anim*, *set\_outline\_color*, *get\_outline\_color*.

Следующие методы объявлены устаревшими и будут удалены в последующих релизах: *set\_glow\_intensity*, *get\_glow\_intensity*, *apply\_glow\_anim*, *apply\_glow\_anim\_def*, *clear\_glow\_anim*, *set\_glow\_color*, *get\_glow\_color*.

- В панели *Object > Blend4Web* изменены настройки.

Добавлен флаг *Enable Outline*, который устанавливает возможность использования эффекта *подсветки контура* на данном объекте. Так же добавлен флаг *Outline on Select*, устанавливающий активацию анимации подсветки при выделении объекта (раньше данное поведение определялось флагом *Selectable*).

- В панель *Scene > Blend4Web* добавлены новые настройки.

Добавлено поле *Enable Object Outlining*, которое управляет возможностью подсветки контура объектов. Аналогично, новое поле *Enable Object Selectable* управляет возможностью выделения объектов.

- Теперь свойства объекта *Apply Scale*, *Apply Modifiers*, *Export Vertex Animation*, *Export Edited Normals* и *Export Shape Keys* взаимно исключают друг друга.
- Изменение API модулей.

В API модуля `util.js` добавился метод `is_armature`, позволяющий проверить, является ли данный объект объектом типа **ARMATURE**.

В API модуля `constraints.js` добавился метод `get_parent`, возвращающий для данного объекта родительский объект.

### 29.7.3 Исправления

- Исправлена ошибка, приводящая к исчезновению описания объекта в модуле `"anchors.js"`.
- Исправлена ошибка запекателя скелетной анимации при наличии арматурных объектов на скрытых слоях.
- Исправлено поведение камеры при использовании функции `"append_semi_stiff_cam"` модуля `"constraints.js"`.

Исправлено вертикальное выравнивание камеры относительно родительского объекта. Также теперь учитывается начальная ориентация камеры, что может потребовать корректировки лимитов вращения, передаваемых в эту функцию.

- Исправлена ошибка обновления плейлиста, если он оказывался пустым.
- Исправлено некорректное поведение физических объектов при удалении хотя бы одного из них со сцены.
- Исправлен баг при наличии нулевого масштабирования у объекта, размноженного через `DupliGroup`.
- Исправлена ошибка компиляции шейдера воды на Windows и некоторых мобильных устройствах.
- Исправлена ошибка при наличии дублированных ключей анимации.
- В анимации *NLA* теперь можно использовать акторы из разных файлов, имеющие одинаковое имя.
- Устранено размножение обработчиков перемещения мыши при многократном вызове `"pointerlock"`.
- Исправлен тип прозрачности `"Alpha Sort"` для динамических объектов.
- Исправлена ошибка сборки аддона, приводящая к неработоспособности на системах Windows без установленного *C++ 2010 runtime*.

- Исправлен баг рендеринга billboard-объектов на iPad.

## 29.8 v15.03

### 29.8.1 Новый функционал

- Инструмент для создания подписей к трёхмерным объектам.

Теперь имеется возможность создавать подписи к трёхмерным объектам (“якори”). Данные якори могут быть трёх различных типов: Аннотации (Annotation) - используется информация из мета-тегов, назначенных на объектах, Элементы (Custom Element) - в качестве якоря можно назначить произвольный HTML-элемент с текущей веб-страницы и Общий (Generic) - якорь, положение которого можно определить с помощью API модуля anchors.

- Анимация и возможность управления через API значениями нод типа Value и RGB.

Теперь помимо анимации нод типа Value возможна анимация нод RGB. Так же в модуле objects.js появились соответствующие функции для изменения значений таких нод.

- Новое приложение “Code Snippets”.

Это [приложение](#) было создано для более быстрого и легкого просмотра уроков по функционалу движка. В нем также можно ознакомиться со скриптами, составляющими основу логики работы приведенных уроков. Приложение доступно из файла index.html, находящегося в корне Blend4Web SDK.

- Новые функции управления эффектом Glow.

В модуль scenes были добавлены новые API: `get_glow_intensity()` и `get_glow_color()`.

- Улучшения в просмотрщике сцен Viewer.

Изменён дизайн кнопки перехода на “домашнюю” сцену, добавлена кнопка “All objects selectable”, позволяющая отключить автоматическое назначение свойства “Selectable” для всех объектов на сцене. Также в просмотрщике теперь можно увидеть суммарное количество шейдеров, выполняющихся на загруженной сцене.

- Динамическое копирование объектов сцены.

Появилась возможность динамического копирования и удаления объектов сцены (инстансинг).

- Обработка ошибок, связанных со специальной нодой B4W\_PARALLAX.

Добавлена обработка некорректного использования ноды B4W\_PARALLAX в виде [некритической ошибки экспорта](#).

- *Новые опции в сборщике приложений.*

В сборщике приложений появились новые опции `-j` и `-c`, которые позволяют добавлять в исключения, соответственно, скрипты и стили, не требующие компиляции.

- *Экспериментальный Blend4Web рендер движок.*

Включается в настройках аддона: флаг “Register Blend4Web render engine (Experimental)”. Предназначен для упрощения редактирования свойств объектов. Упрощает интерфейс, отключив неподдерживаемые аддоном панели. В данный момент в режиме Blend4Web рендера отсутствует возможность редактирования шейдерного нодового дерева.

## 29.8.2 Изменения

- *Изменение точки отсчета лимитов перемещения камеры.*

Выставление лимитов перемещения камеры через API теперь полностью соответствует *отсчету значений* в системе координат движка. Выставление в Blender'е горизонтальных лимитов для камеры типа TARGET в пространстве координат World Space изменилось, поэтому может потребовать перенастройки для старых сцен.

- *Обновление документации по настройкам камеры.*
- *Горизонтальные и вертикальные лимиты вращения камеры теперь полностью независимы друг от друга.*
- *Изменения API модуля camera.js.*

API модуля camera.js претерпело ряд изменений.

Добавились новые методы: `is_target_camera`, `is_eye_camera`, `is_hover_camera`, `rotate_camera`, `rotate_target_camera`, `rotate_eye_camera`, `rotate_hover_camera`, `get_camera_angles`, `hover_cam_set_translation`.

Объявлены устаревшими и будут удалены в последующих релизах: `rotate_pivot`, `rotate_hover_cam`, `rotate`, `set_eye_params`, `get_angles`, `translate_hover_cam_v`, `set_hover_cam_angle`.

Методы `set_ortho_scale` и `get_ortho_scale` при применении их к камере отличного от Orthographic типа теперь выводят сообщение об ошибке. Также изменено поведение метода `get_hover_angle_limits`, который теперь возвращает угловые лимиты для HOVER камеры в формате [down, up] вместо [up, down] как было раньше.

- *Значительно оптимизирована система освещения.*

Многие свойства источников света теперь обрабатываются на этапе компиляции. Теперь нет ограничения в 4 источника для некоторых мобильных устройств.

- Изменён способ вёрстки приложений, использующих модуль `app`.

При инициализации приложения с использованием модуля `app` размеры создаваемого элемента `<canvas>` теперь полностью определяются размерами элемента-контейнера. Это нужно иметь в виду, например, в случае использования элемента `<div>` в качестве контейнера, поскольку по умолчанию его высота равна нулю. Задать правильные размеры контейнера можно с использованием CSS, например с помощью inline-стиля. Также, при любых преобразованиях контейнера теперь необходимо вызывать функцию модуля `app.resize_to_container()`. Аналогичного эффекта можно добиться, если задать опцию `autoresize` при инициализации приложения (в функции `app.init()`). Поддержка низкоуровневого метода изменения размеров элемента `<canvas>` с использованием функции `main.resize()` сохранена.

- Функция модуля `app.enable_controls()` теперь вызывается без параметров.
- Удалены устаревшие методы API.

Модуль `textures.js`: `stop_video`.

Модуль `scenes.js`: `add_object`, `get_screen_scenes`, `set_light_pos`, `set_light_direction`, `set_dir_light_color`, `get_lights_names`, `remove_all`, `check_collision`, `check_ray_hit`, `get_appended_objs`, `get_object_by_empty_name`.

Модуль `physics.js`: `set_character_dist_to_water`.

Модуль `material.js`: `set_batch_param`, `set_max_bones`, `max_bones`.

Модуль `main.js`: `set_shaders_dir`, `set_texture_quality`.

Модуль `data.js`: `get_bpy_world`.

Модуль `controls.js`: `sensor_make_positive`, `sensor_make_negative`.

Модуль `camera.js`: `change_eye_target_dist` (также удалена устаревшая константа `MS_CONTROLS`).

- Новые условия для изменения положения объектов через API.

Функции модуля `transform.js`, связанные с изменением положения объекта, доступны только для динамических объектов.

- Разрешено использование ноды типа `TEXTURE` с отсутствующей текстурой.

Отображение нодового материала в этом случае целиком соответствует Blender'у.

- Обновление механизма проверки совместимости версий экспортованных файлов и движка.

Движок будет сигнализировать о несовместимости загружаемой сцены специальными сообщениями в консоли браузера.

- Свойство объекта “*Do not batch*” переименовано в “*Force Dynamic Object*”

Данная опция сообщает движку о том, что объект обязан быть динамическим независимо от других настроек. Теперь она носит более понятное имя.

### 29.8.3 Исправления

- Исправлена ошибка автоповорота камеры в веб-плеере.
- Исправлена ошибка полноэкранного режима в веб-плеере.
- Исправлена ошибка определения угловых координат камеры в некоторых положениях.
- Исправлена ошибка автоворота камеры при наличии горизонтальных лимитов, ограничивающих поворот.
- Исправлена ошибка множественного добавления специальных нод в blend-файл.
- Исправлена ошибка замены материала на объекте функцией “*inherit\_material*” из модуля “*material.js*”.
- Исправлены ошибки при отражении зеркальной поверхностью объекта, имеющего свойства, измененные с использованием модуля “*material.js*”.
- Исправлена генерация сферических WIREFRAME-объектов в режиме отладки сцены.
- Исправлена процедура оптимизации нод типа *TEXTURE* в нодовом материале.
- Исправлена работа опции “*Clamp*” в ноде типа *MixRGB(Linear Light)*.
- Исправлена ошибка, возникающая при экспорте, если на сцене присутствуют объекты с разделяемым мешием и настройками “*Apply Scale*”, “*Apply Modifiers*”, “*Export Vertex Animation*” или “*Export Edited Normals*”.
- Исправлена ошибка при использовании опции “*Blend4Web > Preserve global orientation and scale*” на ряде мобильных устройств.
- Исправлена ошибка рендеринга тумана в отдельных версиях *Chrome* и *Firefox* под *Windows*.

### 29.8.4 Известные проблемы

- Проблемы с обновлением аддона в *Blender*.

Настоятельно не рекомендуется устанавливать новую версию аддона *Blender*, не удалив предварительно старую версию, особенно это касается систем *Windows*.

- Не работают видеотекстуры в *Firefox* при экспорте сцены в формате *HTML*.

## 29.9 v15.02

### 29.9.1 Новый функционал

- Добавлена возможность запуска локального сервера разработки непосредственно из Blender.

При использовании Blend4Web SDK имеется возможность запуска локального сервера разработки. Сервер предоставляет быстрый доступ к содержимому Blend4Web SDK, а также позволяет автоматически открывать экспортированные сцены в просмотрщике сцен.

- Поддержка теней от источников света типа Spot.

Ранее на лампах Spot тени строились так же, как для ламп типа Sun. Теперь расчет производится аналогично тому, как это делается в Blender, с учетом расхождения лучей света в пространстве.

- Добавлена/расширена поддержка объектов типа “Metaball”, “Surface” и “Curve”.

Была добавлена поддержка объектов типа Metaball, Surface и Curve. Объекты этих типов автоматически конвертируются в меши при экспорте. Поддержка использования объектов типа Curve в модификаторах сохранена.

- В веб-плеер добавлены кнопки социальных сетей.

Данные кнопки позволяют разместить ссылку на 3D сцену, а также ее описание в одной из четырех популярных социальных сетей.

- Добавлена возможность редактирования файла списка сцен для просмотрищика непосредственно в Blender.

Была добавлена возможность быстрого редактирования файла assets.json для приложения Viewer через Blender, если используется Blend4Web SDK.

- Добавлена возможность изготовления скриншотов в приложении Viewer.
- В веб-плеер добавлена опция fallback\_video.

С помощью опции fallback\_video=/path/to/video/ можно указывать видео-файл, который будет проигрываться вместо 3D контента на системах без поддержки WebGL.

- Расширение функционала рендеринга в текстуре.

Добавлена поддержка рендеринга сцен сразу в несколько текстур. Сцены теперь могут иметь любую степень вложенности.

- Для билбордов появилась возможность сохранять ориентацию и масштабирование объектов в мировом пространстве.

Для этого нужно выставить опцию *Blend4Web > Preserve global orientation and scale* на панели настроек объекта.

- Улучшения на главной веб-странице *SDK*.

Теперь на главной странице можно узнать версию используемого *SDK* и проверить систему на совместимость с *WebGL*.

- Добавлена поддержка флага *Clamp* в нодах *MATH* и *MIX\_RGB*.

Функциональность была реализована сначала в *Blender 2.73*, и теперь в движке.

- Улучшено качество рендеринга при отсутствии аппаратной поддержки текстуры глубины.

Увеличены возможности рендеринга для устройств, не поддерживающих текстуру глубины. Стали доступны эффекты, такие как отражение, свечение, засветка, размытие при движении, сглаживание.

- Добавлена новая документация.

Добавлена документация по модулям *vec3*, *vec4*, *quat*, *mat3*, *mat4*, а также глобальному пространству имён *b4w*. Улучшен дизайн веб-страниц документации.

- Добавлена возможность иметь несколько запущенных копий движка на одной странице.

Несколько приложений теперь могут работать одновременно, указывая имя пространства имён при инициализации.

- Возможность использования *SDK* в *Apple OS X*.

В *OS X* теперь доступны все функции *SDK*, включая сборку движка и приложений, конвертацию ресурсов и генерирование документации.

- В модуль *camera* добавлен метод *set\_trans\_pivot()*.

Данная функция позволяет устанавливать произвольное положение опорной точки и положение камеры типа *Target*.

- В утилиту ‘‘*project.py*’’ добавлено новое свойство ‘‘*version*’’.

Использование данного свойства позволяет добавлять версию к скриптам и стилям скомпилированного приложения.

## 29.9.2 Изменения

- Теперь в приложение можно добавлять опции с одинаковыми именами через адресную строку браузера.

В функции *get\_url\_params()* модуля *app* появился необязательный параметр *allow\_param\_array*, по умолчанию равный *false*. При назначении этому параметру *true* одинаковые опции будут объединены в массив, в противном случае будет использоваться последняя.

- Улучшения работы источников звука типа “Background Music“.

Теперь для таких источников поддерживается возможность указания параметров задержки и времени воспроизведения. Также появилась возможность зацикливать их воспроизведение.

- Специальные ноды аддона Blend4Web теперь присутствуют в blend-файле по умолчанию.

Теперь нет необходимости добавлять *специальные ноды* аддона в файл. Это доступно как при использовании Blend4Web SDK, так и при экспресс-установке аддона.

- Изменен экспорт пустых объектов типа “Mesh”.

Теперь объекты типа Mesh, не имеющие полигонов, экспортируются как Empty.

- Изменения модуля “light.js”.

Была добавлена новая функция `get_light_type`, а также теперь в функции `get_light_params` и `set_light_params` передается не имя объекта, а сам объект типа LAMP. Также теперь доступны для изменения через эти функции свойства `spot_blend`, `spot_size`, `distance` для источника света типа SPOT. Для источника света типа POINT теперь доступно для изменения свойство `distance`.

- Улучшение рефракции при низких настройках качества.

При активации профиля качества LOW, теперь используется упрощённая модель рефракции (без искажений).

- Оптимизация нодовых шейдеров.
- Теперь режим автоматического вращения камеры отключается при касании сенсорного экрана.

### 29.9.3 Исправления

- Исправлено поведение прелоадера для веб-плеера.

При открытии веб-плеера в новой вкладке могли наблюдаться артефакты на логотипе B4W.

- Исправлена ошибка рендеринга однокаскадных теней, связанная с возникновением жесткой неосвещенной линии по границе всего каскада.
- Исправлена ошибка, связанная с коллизией имён при линковке объектов в Blender'е.
- Исправлена ошибка с недостаточной оптимизацией приложений из состава SDK.
- Исправлена проблема экспорта мешей с плоским шейдингом (*flat shading*) в Linux x32.

- Исправлено некорректное поведение Target камеры в отдельных случаях.
- Исправлена ошибка с попыткой использования карты теней размером большим, чем поддерживаемый конкретным устройством.
- Исправлена ошибка, приводящая к снижению FPS в Firefox 35/36 под Windows при включении теней.

## 29.9.4 Известные проблемы

- Проблемы с обновлением аддона в Blender.

Настоятельно не рекомендуется устанавливать новую версию аддона Blender, не удалив предварительно старую версию, особенно это касается систем Windows.

- Не работают видеотекстуры в Firefox при экспорте сцены в формате HTML.

## 29.10 v15.01

### 29.10.1 Новый функционал

- Поддержка режима панорамирования для устройств с сенсорным экраном.

Движение камеры в режиме панорамирования осуществляется параллельным перемещением двух пальцев по поверхности экрана.

- Поддержка объектов типа “Text”.

Данные объекты теперь автоматически конвертируются в меш при экспорте.

- Расширение функционала инструмента NLA Script.

Добавлены новые логические слоты: Show Object и Hide Object используются для скрытия и отображения объектов, Page Redirect - для перенаправления на другие веб-страницы, Page Param - для сохранения произвольного параметра веб-страницы в выбранном числовом регистре. Упрощено использование слотов Select & Jump и Select & Play - теперь для выделяемых объектов нет необходимости указывать свойство Selectable.

- Поддержка экранов сверхвысокого разрешения (HIDPI, Retina).

Режим HIDPI позволяет достичь существенного улучшения качества картинки на устройствах с высоким разрешением. Режим включается автоматически при запуске приложений с настройками качества ULTRA. При необходимости, высокое разрешение может быть включено и для других профилей качества.

- Поддержка масштабирования для ортографической камеры.

Добавлено API для изменения масштабирования камеры типа Orthographic (Orthographic scale в Blender'e).

- В веб-плеер добавлена опция autorotate.

Опция autorotate используется для автоматического включения вращения камеры сразу после загрузки сцены.

- В функцию “enable\_camera\_controls” из аддона “app.js” добавлен облегченный режим управления с клавиатурой.

Включить режим можно передав необязательный параметр disable\_letter\_controls. При этом будет отключено управление посредством алфавитно-цифровых клавиш клавиатуры (WASD и т.д.). Указанная функциональность может быть использована для тех случаев, когда в приложении необходимо использовать клавиши для целей, не связанных с перемещением камеры.

- Поддержка работы с гироскопом на мобильном устройстве.

Добавлены два сенсора для работы с гироскопом на мобильных устройствах. Первый сенсор позволяет оперировать с разностью текущего положения устройства с предыдущим, создается функцией create\_gyro\_delta\_sensor из модуля “controls.js”. Второй сенсор - с текущим значением угла. Создается функцией create\_gyro\_angles\_sensor из модуля “controls.js”. Стоит отметить, что все значения вычисляются в радианах. Также был создан аддон “gyroscope.js”, в котором реализовано движение камеры при угловых перемещениях устройства. С примером работы данного функционала для мобильных устройств можно ознакомиться в нашей программе “Viewer”, включив опцию “Gyroscope” в меню программы.

- Добавлено новое свойство “Do not Render” в настройках материала.

Включение данного свойства позволяет не отображать на сцене все объекты, использующие этот материал.

- Поддержка видеотекстур на IE 11 и iPhone.

Поддержка добавлена посредством создания простого формата видео-секвенции \*.seq. Более полную информацию можно посмотреть [в соответствующем разделе документации](#).

- Поддержка тега “title” в веб-плеере.

Тег “title” для веб-плеера, необходимый для вывода названия сцены в браузере, теперь берется из JSON-файла сцены. С более подробной информацией можно ознакомиться [в соответствующем разделе документации](#).

- Поддержка мета-тегов в Blender.

В Blender появилась возможность добавлять мета-теги к сцене и к объектам. Для сцены это “title” и “description”, для объекта “title”, “description” и “category”.

- Добавлена возможность вызова пользовательских функций каждый кадр.

Для возможности создания более сложных приложений в модуле “main.js” была добавлена функция `append_loop_cb`, позволяющая вызывать переданную в нее функцию каждый кадр. При этом в эту функцию каждый кадр будут передаваться два параметра: время с начала старта приложения и разница во времени между текущим кадром и предыдущим. Обе величины измеряются в секундах. Для того, чтобы больше не вызывать переданную функцию необходимо удалить ее вызовом функции `remove_loop_cb` из модуля “main.js”.

- Добавлена анимация простого экрана загрузки приложения.

Для создания приложения с анимированным экраном загрузки в функцию `create_simple_preloader` из модуля “preloader.js” необходимо передать опцию “`preloader_fadeout`” со значением “`true`”.

- Добавлена возможность экспорта конвертированных медиаданных в HTML-файл.

Теперь при экспорте в `html` имеется возможность записать конвертированные данные в файл. Для этого при экспорте необходимо задать свойство “`Export converted media`” в *опциях экспорта*.

- Добавлена возможность использования `min50` и `dds` текстур при просмотре сцен через `webplayer`.

Для этого необходимо передать атрибут “`compressed_textures`” при запуске приложения.

## 29.10.2 Изменения

- Упрощена файловая структура *SDK*.

Директория `external` больше не существует, всё её содержимое перемещено на уровень выше - в корневую директорию *SDK*. Файл со списком сцен для просмотрика `assets.json` теперь находится в директории `apps_dev/viewer`.

- Изменено поведение автоворотения камеры (экспериментально).

При наличии ограничений на горизонтальное вращение камера, приближаясь к границам, плавно замедляется и продолжает движение в обратном направлении.

- Изменено поведение настройки “*Special: Collision*“ в настройках материала.

Ранее включение опции автоматически приводило к сокрытию объектов, теперь для этой цели необходимо использовать настройку материала `Do not Render`.

- Изменен суффикс конвертированных медиаданных.

Суффикс `*.lossconv.*` заменен на `*.altconv.*`.

- *Изменена работа опции “Do not render” на объекте.*

Теперь выставление данной опции на объекте не приводит к потере физических свойств объекта. Объект всего лишь становится невидимым.

- *Доработана комплектация SDK.*

В бесплатной и коммерческой версиях SDK были добавлены новые сцены с примерами использования движка; устаревшие и малоинформационные сцены исключены.

### 29.10.3 Исправления

- *При выставлении в Blender’е камере типа “Panoramic” в движке принудительно используется камера типа “Perspective”.*
- *Исправлено дрожание камеры типа “Target” в отдельных случаях.*
- *Небольшие исправления в работе аддона “B4W Anim Baker”.*
- *Исправлена проблема со звуком на сценах с несколькими камерами.*
- *В модуле “controls” улучшена стабильность работы сенсоров типа “Timer”.*
- *Устранена ошибка, возникающая при просмотре сцены через IE при экспорте в HTML.*
- *Оптимизирована работа видеотекстуры. Теперь не производится обновление видеотекстуры при остановке видео.*
- *Исправлена ошибка рендеринга нодового материала, содержащего ноду “REFRACTION”.*

### 29.10.4 Известные проблемы

- *Проблемы с обновлением аддона в Blender.*

Настоятельно не рекомендуется устанавливать новую версию аддона Blender, не удалив предварительно старую версию, особенно это касается систем Windows.

- *Не работают видеотекстуры в Firefox при экспорте сцены в формате HTML.*
- *Текущая реализация depth-текстур в Firefox 35 приводит к значительному снижению FPS и другим неисправностям.*

В различных сценах наблюдается снижение FPS, например, при включении теней. Также некорректно отображаются тени для текстур с использованием прозрачности, например, ALPHA\_CLIP. Соответствующая ошибка, возможно, будет исправлена в ближайшее время в будущих обновлениях браузера.

## 29.11 v14.12

### 29.11.1 Новый функционал

- *Поддержка настройки скоростей камеры.*

Осуществлена поддержка пользовательской *настройки скоростей* перемещения, вращения, масштабирования камер типа Target, Hover, Eye. Настройка может быть выполнена как в интерфейсе Blender'a, так и посредством API Blend4Web.

- *Поддержка MIP-текстурирования для Canvas-текстур.*

Осуществлена поддержка MIP-текстурирования для *Canvas-текстур*.

- *Полная поддержка ноды “MAPPING”.*

Осуществлена поддержка всех возможных значений *Vector type* для ноды MAPPING: Texture, Point, Vector и Normal.

- *Glow-эффект при наведении курсора мыши на объект.*

Для эффекта подсветки объекта под курсором мыши доступно API модуля `mouse.js`. Управление эффектом осуществляется с помощью функций `enable_mouse_hover_glow()` и `disable_mouse_hover_glow()`. При этом на объекте должно быть выставлено свойство `Object > Blend4Web > Selectable`.

- *Новая система сборки приложений.*

Теперь пользователи имеют возможность вести разработку приложений непосредственно в составе SDK, благодаря новой утилите `project.py`. *Данный скрипт* позволяет собирать приложения вместе с движком, минимизировать JavaScript и CSS-файлы, а также экспорттировать готовое приложение для последующего развертывания на сервере.

### 29.11.2 Изменения

- *Удалена поддержка устаревшей опции текстур “UV translation velocity”.*

Вместо неё рекомендуется использовать функционал нодовых материалов.

- *Удалена поддержка устаревшего интерфейса “Levels of Detail”.*

Аналогичный функционал теперь реализуется штатным инструментом Blender “Levels of Detail”.

- *Аддон `pointerlock.js` переименован в `mouse.js`.*

- Добавлено срабатывание события *mouseup* при покидании курсора мыши вьюпорта приложения.

Таким образом была исправлена наблюдавшаяся ранее проблема с нарушением контроля камеры.

- Сообщение об ошибке “*Clear parent inverse*” более не появляется.

Ранее при использовании отношения родитель-потомок требовалось сбрасывать перемещение, вращение и масштаб объекта-потомка (*Object > Parent > Clear Parent Inverse*). Указанная трансформация теперь поддерживается движком.

- Использование “*Apply scale*” теперь не приводит к применению модификаторов.

Для применения модификаторов по-прежнему следует использовать флаг *Apply modifiers*.

- Использование карты нормалей в новом материале больше не требует наличия под *Material* или *Extended Material*.

В некоторых случаях (например, рефракция) карты нормалей могут применяться в материалах без освещения.

### 29.11.3 Исправления

- Исправлена ошибка воспроизведения звуков через инструмент *NLA*.

Ошибка проявлялась вследствие недостаточной точности чисел с плавающей запятой.

- Исправлен некорректный рендеринг источников света на мобильных устройствах.

- Исправлено наложение тени при нескольких активных источниках освещения.

Теперь тень накладывается так же как в Blender, не затеняя те участки, куда попадает свет от других источников.

- Исправлена ошибка отображения нововых материалов.

Ошибка возникала при использовании под **MATERIAL** и **MATERIAL\_EXT** с добавленным по ссылке (из другого .blend-файла) материалом.

- Исправлен сброс позы арматуры при использовании “*B4W Animation Bake*”.

Теперь при использовании инструмента *запекания анимации* арматурный объект остаётся в позе, в которой он находился до запекания.

- Исправлено резкое движение камеры после старта приложения.

- Исправлена ошибка, связанная с неверным определением лимитов горизонтального вращения камеры.

- Исправлена ошибка, связанная с экспортом неиспользуемых текстур.

## 29.11.4 Известные проблемы

- Проблемы с обновлением аддона в Blender.

Настоятельно не рекомендуется устанавливать новую версию аддона Blender, не удалив предварительно старую версию, особенно это касается систем Windows.

- Не работают видеотекстуры в Firefox при экспорте сцены в формате HTML.

## 29.12 v14.11

### 29.12.1 Новый функционал

- Поддержка видео-текстур.

Осуществлена поддержка *видео-текстур* для текстур типа **Image or Movie**.

- Частота кадров.

Добавлена поддержка изменения частоты кадров для анимации и видеотекстур. Частоту кадров можно задать опцией **Scene > Dimensions > Frame rate**.

- Поддержка текстур типа Canvas.

В качестве *текстуры* используется HTML-элемент **Canvas**. Работа с ней *описана* в документации.

- Camera panning.

В режиме вращающейся вокруг точки камеры (**Target**) теперь имеется возможность перемещения точки вращения в плоскости обзора (т.н. панорамирование) при зажатой правой или средней кнопках мыши. Функция включена по умолчанию и при необходимости выключается в настройках в Blender'е.

- Новый режим управления камерой - Hover.

Реализован режим скольжения камеры над плоскостью (**Hover**),ключающий возможность приближения и удаления. Этот режим камеры позволяет реализовывать сценарии удобного просмотра протяженных в двух измерениях сцен (помещений, игровых уровней).

- В SDK добавлена корневая веб-страница *index.html* для упрощения навигации по дистрибутиву.

- Поддержка преобразования форматов видео-файлов в конвертере ресурсов.

- Добавлена система сборки, ранее отсутствовавшая в публичных дистрибутивах *SDK*.
- Поддержка режима экспорта “*Strict mode*” в аддоне.

Активация режима **Strict mode** позволяет вывести все возможные ошибки и предупреждения, связанные с некорректной настройкой сцены. Опция полезна для финальной отладки сцены с целью получения максимально корректных и оптимизированных ресурсных файлов.

- Поддержка воспроизведения звуков на устройствах с *iOS*.

## 29.12.2 Изменения

- Опция “*bg*” веб-плеера переименована в “*fallback\_image*”.

Также претерпела изменение логика поведения опции. При указании *fallback\_image* информация об отсутствии поддержки WebGL теперь не выводится, вместо неё пользователю демонстрируется изображение.

- В веб-плеере опция отключения звука теперь не показывается для сцен, не имеющих источников звука.
- Улучшена стабильность работы стековых материалов.
- В сенсор “*mouse\_down*” добавлен код на жатой кнопки мыши, который доступен в поле *payload* сенсора.
- Значительно оптимизирован экспорт систем частиц типа *Hair*.

## 29.12.3 Исправления

- Карты нормалей теперь работают с типами текстурных координат *Generated* и *Normal*.

При использовании карт нормалей теперь нет необходимости использовать UV-развертку.

- Исправлена проблема с неверными путями к физическому движску в веб-плеере.

Ошибка проявлялась при перемещении файла *uranium.js* из директории с главным HTML-файлом веб-плеера.

- В аддоне исправлена проблема с упакованными текстурами, проявляющаяся при экспорте при включённой опции “*Automatically Pack Into .blend*”.

## 29.12.4 Известные проблемы

- *Проблемы с обновлением аддона в Blender.*

Настоятельно не рекомендуется устанавливать новую версию аддона Blender, не удалив предварительно старую версию, особенно это касается систем Windows.

## 29.13 v14.10

### 29.13.1 Новый функционал

- *Новый веб-плеер.*

Новый минималистичный дизайн веб-плеера удачно сочетается с любыми трёхмерными сценами, имеет упрощённый интерфейс и встроенную подсказку по управлению. Веб-плеер работает на всём спектре оборудования, включая мобильные устройства.

- *Улучшение системы теней.*

Добавлена возможность выбора некаскадной модели теней на основе одной оптимизированной карты теней. Такая модель проще в настройке и хорошо подходит для небольших сцен. Подробности в [документации](#).

- *Многочисленные улучшения в системе нелинейной анимации (NLA).*

Появилась возможность создания сложной логики в инструменте NLA Script с помощью блоков Conditional Jump, Register Store и Math Operation, а также переменных, сохраняемых в регистрах.

Возможность использования в NLA всех типов анимации, поддерживаемых движком, включая воспроизведение звуков, вертекскую анимацию и эмиссию частиц. Анимацию различных типов теперь возможно проигрывать параллельно.

- *Поддержка биллбординга объектов.*

Объектам добавлены [опции](#), позволяющие настроить биллбординг.

- *Поддержка режима “XYZ Euler” для анимации поворота.*

Объектная и скелетная анимация теперь поддерживают режим XYZ Euler для анимации поворота.

- *Поддержка текстурных координат типа GENERATED.*
- *Добавлена поддержка загрузки ресурсов с удаленного сервера (Cross-origin resource sharing).*
- *Упрощен процесс экспортса сцен.*

Ряд ошибок, возникающих при экспорте материала, теперь не блокирует экспорт. Вместо этого при загрузке сцены данный материал будет отображаться розовым цветом. Более подробное описание ошибок доступно в [документации](#).

- Добавлена поддержка опции “*Do not export*” для системы частич.
- Улучшена стабильность работы приложений на устройствах с мобильной операционной системой Apple iOS.

### 29.13.2 Изменения

- В SDK изменился путь для настройки экспорта в HTML.

В настройках аддона, установленного через SDK, теперь достаточно указывать путь непосредственно к корневой директории SDK. Ранее требовалось указывать путь к приложению embed. Обратите внимание, старое поведение более не поддерживается.

- Объявлена устаревшей опция текстур “*UV translation velocity*”.
- Опция будет удалена начиная с версии 14.12.
- Удалена опция “*Do not export*” на панели “*Object data*”.
  - Опция *Blend4Web>Animation>Cyclic* на вкладке свойств объектов удалена.

Вместо неё следует использовать опцию *Blend4Web > Animation > Behavior*, расположенную там же. В сценах, где циклическая анимация назначается по умолчанию, возможно нарушение работы анимации, соответственно в них необходимо установить поведение в *Cyclic*.

- Изменена реализация алгоритма *SSAO*.

Новая реализация работает значительно быстрее и показывает лучшее качество изображения. Параметры настройки алгоритма также изменились. Подробнее в [документации](#).

### 29.13.3 Исправления

- Исправлена ошибка рендеринга материала типа *HALO*.
- Исправлена ошибка рендеринга объекта при выставлении опции “*Force Dynamic Object*”, если он имел родительский объект.
- Исправлена ошибка назначения горячих клавиш на экспорт аддона *Blend4Web*.

Устранена ошибка, не позволяющая назначить горячие клавиши на экспорт в форматы JSON и HTML: *File->Export->Blend4Web(.json)* и *File->Export->Blend4Web(.html)*.

- Исправлена ошибка загрузки текстур, превышающих допустимый размер для данного устройства.
- Исправлены ошибки анимации нодовых материалов, приводившие к нестабильной работе движка.
- Устранена ошибка нодовых материалов, содержащих сложные группы нод (*Node Groups*).
- Исправлены ошибки компиляции шейдеров на устройствах с мобильной графикой *Qualcomm Adreno 305*.
- Исправлена ошибка рендеринга при использовании ноды *REFRACTION* на непрозрачных материалах.
- Исправлен сброс значения текущего кадра анимации после её запечатывания при помощи инструмента “*B4W Vertex Anim Baker*”.

## 29.13.4 Известные проблемы

- Проблемы с обновлением аддона в *Blender*.

Настоятельно не рекомендуется устанавливать новую версию аддона *Blender*, не удалив предварительно старую версию, особенно это касается систем Windows.

- Карты нормалей не работают с типом текстурных координат *Generated*.

При использовании карт нормалей необходимо использовать UV-развертку.

## 29.14 v14.09

### 29.14.1 Новый функционал

- Поддержка типа *ABSOLUTE* для ноды *MATH*.
- Поддержка специализированной ноды *LEVELS\_OF\_QUALITY*.

Позволяет управлять сложностью материала в зависимости от профиля качества, указанного пользователем при загрузке движка.

- Поддержка специализированной ноды *SMOOTHSTEP*.

Упрощает создание некоторых эффектов в нодовом материале.

- Поддержка нодовых групп.

Нодовые группы позволяют повторное использования блока нод одного материала в составе другого.

- Возможность вывода промежуточных результатов рендеринга с целью отладки.

Результат рендеринга конкретной стадии теперь можно вывести поверх основного изображения. Настройка осуществляется в модуле config.js опциями группы debug\_subs.

- *Реализация логики управления NLA-анимацией в Blender с помощью визуального редактора.*

В интерфейс Blender добавлен инструмент NLA Script, который позволяет с помощью визуальных блоков реализовывать простые сценарии, например осуществить управление текущей анимацией в ответ на действия пользователя.

- *Многочисленные улучшения системы сенсоров.*

В модуле controls разрешено регистрировать множества сенсоров глобально, без привязки к конкретному объекту, для чего в соответствующие API необходимо подавать параметр null. Обеспечена более предсказуемая и надёжная обработка логики сенсоров, в соответствии с последовательностью создания их множеств. Обработчики событий, связанных с действиями пользователя теперь назначаются с помощью функций вида register\_<inputtype>\_events(). В данные функции теперь можно подавать флаг prevent\_default, позволяющий заблокировать стандартное поведение браузера, происходящее после возникновения соответствующих событий.

- *Поддержка использования физики в веб-плеере.*

Работает только в версии веб-плеера с раздельной загрузкой JSON-файлов. В одиночных HTML файлах физика не поддерживается.

- *Поддержка смешивания различных скелетных анимаций.*

В модуле animation.js добавились API, позволяющие осуществлять плавные переходы между скелетными анимациями: get\_skel\_mix\_factor() - для получения текущего значения коэффициента смешивания и set\_skel\_mix\_factor() - для его назначения.

- *Поддержка анимации ноды Value в нодовых материалах.*

Функциональность работает аналогично другим видам анимации. Поддерживается работа в NLA.

- *Поддержка параметров Specular и Diffuse в источниках освещения.*
- *Возможность рендеринга полупрозрачного объекта поверх остальных объектов на сцене.*

Активируется с помощью опции Render above all для прозрачных материалов (т.е. не opaque).

- *Автоматическое применение масштаба к мешу объекта.*

Реализуется включением опции Apply scale в настройках объекта.

- *Поддержка профиля высокого качества (включая тени, динамическое отражение и антиалиасинг) для iOS.*

## 29.14.2 Изменения

- Улучшение рендеринга теней.

Существенно изменена система отрисовки теней, которая теперь основывается на технике **Stable Cascaded Shadow Maps**. Техника позволяет существенно уменьшить мерцание краев теней при движении камеры. Между каскадами реализовано сглаживание. Также тени последнего каскада плавно исчезают при удалении. Для рендеринга мягких теней реализована техника **Percentage Closer Shadows**. Переработана и упрощена система пользовательских настроек теней. Теперь можно задавать размеры карт теней, степень размытия, настройки для компенсации ошибок самозатенения. Новые настройки подробно [документированы](#).

- Настройка качества графики в веб-плеере теперь сохраняется для каждой сцены независимо.
- Изменено поведение настроек конфигурации приложения: `physics_uranium_path`, `smaa_search_texture_path` и `smaa_area_texture_path`.

Данные настройки конфигурации теперь рассчитываются автоматически в зависимости от местоположения запускаемого HTML-файла, если они не были переопределены при инициализации приложения.

- Завершён переход на систему модулей, подключаемых через вызов `b4w.require()`.

Это также означает, что начиная с текущей версии в релиз-версии движка модули нельзя вызывать с помощью устаревшего интерфейса `b4w.<module>`. С целью совместимости создан аддон `ns_compat.js`, подключение которого позволяет восстановить старое поведение.

- Добавлена возможность скрытия панели управления в веб-плеере.
- Скелетная анимация применяется только к объекту арматуры.

Нет необходимости применять скелетную анимацию к объектам типа MESH. Если они привязаны к какой-либо анимированной арматуре, скиннинг будет происходить автоматически.

- Демонстрационные приложения и обучающие материалы приведены в соответствие с новым функционалом.

## 29.14.3 Исправления

- Индикатор загрузки не скрывался, если в процессе происходила ошибка загрузки ресурса (текстуры либо звукового файла).
- Исправлены лаги при масштабировании и повороте на touch-устройствах.
- Устранено дрожание камеры типа TARGET при небольших поворотах.
- Исправлено управление камерой типа EYE на мобильных устройствах.

- Улучшено управление в приложении Ферма в браузере *Safari*.
- Исправлены ошибки при использовании неподдерживаемых моделей освещения на нодовых материалах.
- Для объектов без материалов теперь работает опция “Selectable”.
- Более нет необходимости включать “Force Dynamic Object” для объектов, анимирующихся с использованием *NLA*.
- Исправлена ошибка для систем частиц, в которых дуплицируемый объект входит в иерархию с другими объектами.

#### 29.14.4 Известные проблемы

- Проблемы с обновлением аддона в *Blender*.

Настоятельно не рекомендуется устанавливать новую версию аддона *Blender*, не удалив предварительно старую версию, особенно это касается систем Windows.

- Смешивание скелетной анимации не работает в некоторых браузерах.

Если при использовании API смешивания скелетной анимации возникают необработанные ошибки, следует переопределить стандартную функцию *Math.sign* следующим образом:

```
var m_util = require("util");
Math.sign = m_util.sign;
```

- genindex
- search

## Symbols

- Управление перемещением объектов, 60  
аддон, 37  
    ошибки инициализации, 42  
    установка, 13  
    установка новой версии, 15  
анаглиф, 149  
анимация, 173  
береговая линия, 190  
браузер, 4  
браузерные технологии, 4  
цвет фона, 139  
драйверы, 5  
движок, 1  
эффект Френеля, 85  
экспорт, 14, 24  
    ошибки, 44  
    предупреждения об ошибках экспорта, 48  
глубина резкости камеры (depth of field), 143  
графический движок, 1  
интерактивная графика, 5  
источники света, 129  
компиляция шейдеров, 258  
    директивы import/export, 260  
    обfuscation, 259  
    оптимизация, 259  
    ошибки, 262  
    расширения WebGL, 262  
    рекомендации и ограничения, 260  
    валидация, 259  
коррекция цвета, 149  
кватернион, 62  
материалы, 79  
    гало, 88  
ноды, 106–108  
опции рендеринга, 86  
параметры освещения, 80  
прозрачность, 82  
специальные параметры, 87  
зеркальное отражение, 83  
нормали, 117  
редактор, 117  
освещение, 115  
параметры берега, 190  
подсветка контура (outline glow), 147  
профиль Blend4Web, 14  
просмотрщик, 27  
    добавление сцен, 24  
    запуск, 21  
прозрачность, 82  
    настройка, 83  
    типы, 82  
размытие при движении (motion blur), 142  
сенсор, 247  
    множество, 247  
стлаживание, 151  
система частиц  
    флюиды, 152  
    инстансинг, 162  
стереоизображение, 149  
сумеречные лучи (crepuscular rays), 146  
текстуры, 89  
    диффузная, 92  
    карта бликов, 92  
    карта нормалей, 93  
    карта окружения, 98  
    карта смешивания, 94  
    карта высот, 93  
    карта зеркального отражения, 100  
    настройки, 91

- небо, 101  
рендеринг в, 103  
типы, 90  
трехмерный движок, 1  
веб-плеер, 31  
версия  
    ошибки, 43  
видео-карта, 5  
взаимное затенение (screen-space ambient occlusion), 144  
замечания к релизам, 282  
затенение, 116  
    типы, 116  
зеркальное отражение, 83  
    динамическое, 84  
    эффект Френеля, 85  
    статическое, 84  
3D моделирование, 3
- B**
- Blend4Web, 1  
Blender, 3  
    установка, 12
- D**
- diffuse map, 92  
DOF, 143
- E**
- environment map, 98
- G**
- git, 265  
    добавление и удаление файлов, 269  
    индивидуальные настройки, 267  
    коммит, 270  
    подготовка к коммиту, 268  
    проверка статуса, 267  
    разрешение конфликтов, 273  
    синхронизация между репозиториями, 271  
    тэги, 275  
god rays, 146
- H**
- halo, 88  
height map, 93
- M**
- matcap, 91
- material capture, 91  
mirror map, 100
- N**
- normal map, 93
- P**
- parallax mapping, 93
- R**
- render-to-texture, 103  
RTT, 103
- S**
- sensor, 247  
    manifold, 247  
skydome, 101  
specular map, 92  
SSAO, 144  
stencil map, 94
- W**
- WebGL, 2  
    поддержка в браузерах, 2  
    преимущества, 3