

---

**Blend4Web. Руководство  
пользователя**  
*Выпуск v17.04.1*

ООО "Триумф"

15 May 2017

|   |           |
|---|-----------|
| <b>1 Общие сведения</b>                                     | <b>1</b>  |
| 1.1 Что такое Blend4Web . . . . .                           | 1         |
| 1.2 “Движок” . . . . .                                      | 1         |
| 1.3 Графический движок, трёхмерный 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 Возможности движка</b>                                 | <b>7</b>  |
| <b>3 Установка и обновление</b>                             | <b>14</b> |
| 3.1 Установка . . . . .                                     | 14        |
| 3.2 Обновление . . . . .                                    | 18        |
| <b>4 Рабочий процесс</b>                                    | <b>22</b> |
| 4.1 Создание нового проекта . . . . .                       | 23        |
| 4.2 Создание сцен . . . . .                                 | 23        |
| 4.3 Подготовка трёхмерных сцен в программе Blender. . . . . | 24        |
| 4.4 Экспорт сцен . . . . .                                  | 24        |
| 4.5 Разработка приложения . . . . .                         | 25        |
| 4.6 Сборка проекта . . . . .                                | 25        |
| 4.7 Размещение проекта. . . . .                             | 25        |
| <b>5 Управление проектами</b>                               | <b>27</b> |
| 5.1 Менеджер проектов . . . . .                             | 28        |
| 5.2 Расширенное управление проектами . . . . .              | 44        |
| <b>6 Просмотрщик сцен</b>                                   | <b>53</b> |
| 6.1 Навигация . . . . .                                     | 54        |
| 6.2 Боковая панель . . . . .                                | 54        |

|           |  |            |
|-----------|--|------------|
| 6.3       | Индикаторы . . . . .                                 | 66         |
| <b>7</b>  | <b>Веб-плеер</b>                                     | <b>68</b>  |
| 7.1       | Использование . . . . .                              | 70         |
| 7.2       | Навигация . . . . .                                  | 70         |
| 7.3       | Панель управления . . . . .                          | 70         |
| 7.4       | Атрибуты . . . . .                                   | 71         |
| 7.5       | Название сцены в заголовке . . . . .                 | 72         |
| 7.6       | Ошибки сцены . . . . .                               | 72         |
| <b>8</b>  | <b>Аддон</b>   | <b>75</b>  |
| 8.1       | Локальный сервер разработки . . . . .                | 75         |
| 8.2       | Запуск просмотрщика сцен и демо-приложений . . . . . | 78         |
| 8.3       | Форматы экспорта . . . . .                           | 79         |
| 8.4       | Опции экспорта . . . . .                             | 81         |
| 8.5       | Ошибки инициализации . . . . .                       | 85         |
| 8.6       | Ошибки совместимости . . . . .                       | 86         |
| 8.7       | Критические ошибки экспорта . . . . .                | 89         |
| 8.8       | Некритические ошибки экспорта . . . . .              | 90         |
| 8.9       | Прочие сообщения . . . . .                           | 96         |
| 8.10      | Перевод аддона . . . . .                             | 97         |
| <b>9</b>  | <b>Параметры сцены</b>                               | <b>99</b>  |
| 9.1       | Вкладка Render . . . . .                             | 101        |
| 9.2       | Вкладка Scene . . . . .                              | 113        |
| 9.3       | Вкладка World . . . . .                              | 121        |
| <b>10</b> | <b>Объекты</b>                                       | <b>126</b> |
| 10.1      | Типы . . . . .                                       | 126        |
| 10.2      | Статические и динамические объекты . . . . .         | 127        |
| 10.3      | Настройка . . . . .                                  | 128        |
| 10.4      | Настройки якорей . . . . .                           | 135        |
| 10.5      | Управление перемещением объектов . . . . .           | 137        |
| 10.6      | Выбор объектов . . . . .                             | 138        |
| 10.7      | Выделение объектов . . . . .                         | 139        |
| 10.8      | Копирование объектов (инстансинг) . . . . .          | 140        |
| 10.9      | Кватернионы . . . . .                                | 141        |
| 10.10     | Перемещение с помощью векторов TSR . . . . .         | 143        |
| 10.11     | Визуализация линий . . . . .                         | 144        |
| 10.12     | Уровни детализации (LOD) . . . . .                   | 145        |
| <b>11</b> | <b>Меши</b>  | <b>150</b> |
| 11.1      | Статические и динамические меши . . . . .            | 150        |
| 11.2      | Настройка . . . . .                                  | 151        |
| 11.3      | Морфинг . . . . .                                    | 152        |
| <b>12</b> | <b>Редактор нормалей</b>                             | <b>154</b> |
| 12.1      | Основные особенности редактора нормалей: . . . . .   | 157        |
| 12.2      | Интерфейс . . . . .                                  | 157        |

|           |   |            |
|-----------|---|------------|
| 12.3      | Activate . . . . .                            | 157        |
| 12.4      | Show Normals . . . . .                        | 158        |
| 12.5      | Rotation . . . . .                            | 158        |
| 12.6      | Режимы Absolute и Offset . . . . .            | 159        |
| 12.7      | Split Normals . . . . .                       | 160        |
| 12.8      | Average . . . . .                             | 161        |
| 12.9      | Restore . . . . .                             | 162        |
| 12.10     | Tree, Foliage и Face . . . . .                | 163        |
| 12.11     | Copy/Paste . . . . .                          | 165        |
| 12.12     | Copy From Mesh . . . . .                      | 166        |
| <b>13</b> | <b>Камера</b>                                 | <b>169</b> |
| 13.1      | Режимы управления и общие настройки . . . . . | 169        |
| 13.2      | Ограничения движения камеры . . . . .         | 171        |
| 13.3      | Привязка к камере . . . . .                   | 179        |
| 13.4      | Управление через API . . . . .                | 181        |
| <b>14</b> | <b>Материалы</b>                              | <b>187</b> |
| 14.1      | Параметры освещения . . . . .                 | 188        |
| 14.2      | Прозрачность . . . . .                        | 190        |
| 14.3      | Зеркальное отражение . . . . .                | 192        |
| 14.4      | Опции рендеринга . . . . .                    | 195        |
| 14.5      | Настройки выупорта . . . . .                  | 196        |
| 14.6      | Специальные параметры движка . . . . .        | 196        |
| 14.7      | Материалы гало (Halo) . . . . .               | 197        |
| 14.8      | Методы API для работы с материалами . . . . . | 198        |
| 14.9      | Наследование материалов . . . . .             | 201        |
| <b>15</b> | <b>Библиотека материалов</b>                  | <b>203</b> |
| 15.1      | Структура библиотеки . . . . .                | 204        |
| 15.2      | Структура материала . . . . .                 | 205        |
| 15.3      | Добавление материала в сцену . . . . .        | 209        |
| <b>16</b> | <b>Нодовые материалы</b>                      | <b>212</b> |
| 16.1      | Стандартные ноды . . . . .                    | 213        |
| 16.2      | Дополнительные ноды . . . . .                 | 214        |
| 16.3      | Ноды Cycles . . . . .                         | 226        |
| <b>17</b> | <b>Редактор логики</b>                        | <b>228</b> |
| 17.1      | Основы . . . . .                              | 228        |
| 17.2      | Control Flow . . . . .                        | 231        |
| 17.3      | Animation . . . . .                           | 237        |
| 17.4      | Camera . . . . .                              | 241        |
| 17.5      | Object . . . . .                              | 243        |
| 17.6      | Operations . . . . .                          | 250        |
| 17.7      | Sound . . . . .                               | 254        |
| 17.8      | Network . . . . .                             | 256        |
| 17.9      | Debug . . . . .                               | 261        |
| 17.10     | Deprecated . . . . .                          | 262        |

|   |            |
|---|------------|
| 17.11 Layout . . . . .                                    | 263        |
| 17.12 Отладка . . . . .                                   | 265        |
| <b>18 Освещение, тени и фон</b>                           | <b>266</b> |
| 18.1 Типы затенения . . . . .                             | 266        |
| 18.2 Освещение от источников света . . . . .              | 268        |
| 18.3 Освещение от окружающей среды . . . . .              | 269        |
| 18.4 Тени . . . . .                                       | 271        |
| 18.5 Цвет фона . . . . .                                  | 278        |
| <b>19 Спецэффекты</b>                                     | <b>281</b> |
| 19.1 Размытие при движении . . . . .                      | 281        |
| 19.2 Глубина резкости камеры . . . . .                    | 282        |
| 19.3 Взаимное затенение . . . . .                         | 285        |
| 19.4 Сумеречные лучи . . . . .                            | 286        |
| 19.5 Эффект засветки ярких деталей . . . . .              | 287        |
| 19.6 Подсветка контура (outlining) . . . . .              | 288        |
| 19.7 Свечение (glow) . . . . .                            | 289        |
| 19.8 Сглаживание . . . . .                                | 290        |
| <b>20 Стереоизображение</b>                               | <b>292</b> |
| 20.1 Активация . . . . .                                  | 293        |
| 20.2 Дополнительные настройки . . . . .                   | 294        |
| 20.3 Настройки HMD . . . . .                              | 294        |
| <b>21 Текстуры</b>  | <b>296</b> |
| 21.1 Типы текстур . . . . .                               | 296        |
| 21.2 Общие настройки . . . . .                            | 297        |
| 21.3 Диффузная текстура (diffuse map) . . . . .           | 298        |
| 21.4 Карта бликов (specular map) . . . . .                | 299        |
| 21.5 Карта нормалей (normal map) . . . . .                | 299        |
| 21.6 Карта высот (height map). Parallax mapping . . . . . | 300        |
| 21.7 Карта смешивания (stencil map) . . . . .             | 301        |
| 21.8 Видео-текстуры . . . . .                             | 303        |
| 21.9 Карта окружения (environment map) . . . . .          | 305        |
| 21.10 Карта зеркального отражения (mirror map) . . . . .  | 307        |
| 21.11 Текстура неба (skydome) . . . . .                   | 308        |
| 21.12 Особые типы текстур . . . . .                       | 310        |
| <b>22 Система частиц. Флюиды</b>                          | <b>314</b> |
| 22.1 Использование . . . . .                              | 315        |
| 22.2 Настройка . . . . .                                  | 316        |
| 22.3 Текстуры в системах частиц . . . . .                 | 321        |
| <b>23 Система частиц. Инстансинг</b>                      | <b>324</b> |
| 23.1 Настройки системы частиц . . . . .                   | 326        |
| 23.2 Травяной покров . . . . .                            | 331        |
| 23.3 Листва деревьев . . . . .                            | 333        |

|   |            |
|---|------------|
| <b>24 Анимация</b>                                      | <b>336</b> |
| 24.1 Управление анимацией . . . . .                     | 337        |
| 24.2 Объектная анимация . . . . .                       | 337        |
| 24.3 Скинниング и скелетная анимация . . . . .            | 339        |
| 24.4 Вертексная анимация . . . . .                      | 341        |
| 24.5 Анимация по умолчанию . . . . .                    | 343        |
| 24.6 Нелинейная анимация . . . . .                      | 343        |
| 24.7 Параметризация источников звука . . . . .          | 345        |
| 24.8 Анимация над Value и RGB . . . . .                 | 345        |
| <b>25 Природные эффекты</b>                             | <b>348</b> |
| 25.1 Вода . . . . .                                     | 349        |
| 25.2 Атмосфера . . . . .                                | 357        |
| 25.3 Lens Flare . . . . .                               | 360        |
| 25.4 Ветер . . . . .                                    | 362        |
| <b>26 Работа с цветом</b>                               | <b>366</b> |
| 26.1 Общее описание . . . . .                           | 366        |
| 26.2 Человеческое зрение и мониторы . . . . .           | 367        |
| 26.3 Гамма . . . . .                                    | 368        |
| 26.4 Коррекция в нодовых материалах . . . . .           | 368        |
| 26.5 Альфа-композитинг . . . . .                        | 369        |
| 26.6 Коррекция цвета . . . . .                          | 370        |
| <b>27 Звук</b>  | <b>372</b> |
| 27.1 Настройка звуковых источников . . . . .            | 373        |
| 27.2 Микшер . . . . .                                   | 376        |
| 27.3 Обработка и кодирование . . . . .                  | 377        |
| <b>28 Физика</b>  | <b>378</b> |
| 28.1 Подготовка к использованию . . . . .               | 378        |
| 28.2 Статический тип физики . . . . .                   | 379        |
| 28.3 Динамический тип физики . . . . .                  | 381        |
| 28.4 Ограничители (Constraints) . . . . .               | 382        |
| 28.5 Колесные транспортные средства . . . . .           | 384        |
| 28.6 Плавающие объекты . . . . .                        | 386        |
| 28.7 Плавающие транспортные средства . . . . .          | 388        |
| 28.8 Персонажи . . . . .                                | 389        |
| 28.9 Навигационные меши . . . . .                       | 393        |
| 28.10 Особенности использования в приложениях . . . . . | 397        |
| <b>29 Разработчикам приложений</b>                      | <b>398</b> |
| 29.1 Разработка приложений . . . . .                    | 398        |
| 29.2 Прозрачный фон . . . . .                           | 405        |
| 29.3 Конвертация ресурсов . . . . .                     | 407        |
| 29.4 Компрессия GZIP . . . . .                          | 413        |
| 29.5 Примеры кода . . . . .                             | 415        |
| 29.6 Загрузка медиаресурсов приложения . . . . .        | 415        |
| 29.7 Событийная модель . . . . .                        | 416        |

|   |            |
|---|------------|
| 29.8 Файловая структура SDK . . . . .                                   | 418        |
| 29.9 Загрузка локальных ресурсов . . . . .                              | 421        |
| 29.10 Профили качества изображения . . . . .                            | 421        |
| 29.11 Нестандартное расположение и ориентация элемента Canvas . . . . . | 423        |
| 29.12 Мобильные веб-приложения . . . . .                                | 424        |
| <b>30 Разработчикам движка</b>  | <b>426</b> |
| 30.1 Стиль оформления кода . . . . .                                    | 426        |
| 30.2 Сборка движка . . . . .  | 428        |
| 30.3 Сборка аддона . . . . .  | 428        |
| 30.4 Зависимости . . . . .  | 429        |
| 30.5 Названия функций и переменных . . . . .                            | 429        |
| 30.6 Отладка . . . . .  | 430        |
| 30.7 Компиляция шейдеров . . . . .                                      | 430        |
| 30.8 Обновление переводов аддона . . . . .                              | 436        |
| <b>31 Работа в команде с использованием Git</b>                         | <b>438</b> |
| 31.1 Общие сведения . . . . .   | 438        |
| 31.2 Типичный рабочий процесс . . . . .                                 | 439        |
| 31.3 Индивидуальные настройки . . . . .                                 | 439        |
| 31.4 Проверка статуса . . . . .   | 440        |
| 31.5 Перед коммитом . . . . .   | 440        |
| 31.6 Подготовка к коммиту . . . . .                                     | 441        |
| 31.7 Коммит . . . . .   | 442        |
| 31.8 Синхронизация между репозиториями . . . . .                        | 443        |
| 31.9 Разрешение конфликтов . . . . .                                    | 445        |
| 31.10 Тэги . . . . .  | 448        |
| 31.11 Другие полезные команды . . . . .                                 | 449        |
| <b>32 Проблемы и решения</b>  | <b>450</b> |
| 32.1 Поддержка WebGL . . . . .  | 450        |
| 32.2 Проблемы при запуске движка . . . . .                              | 451        |
| 32.3 Ошибка инициализации WebGL . . . . .                               | 452        |
| 32.4 Если ничего не помогло... . . . . .                                | 457        |
| 32.5 Известные проблемы . . . . .                                       | 457        |
| <b>33 Замечания к релизам</b>   | <b>460</b> |
| 33.1 v17.04 . . . . .   | 460        |
| 33.2 v17.02 . . . . .   | 463        |
| 33.3 v16.12 . . . . .   | 468        |
| 33.4 v16.11 . . . . .   | 471        |
| 33.5 v16.10 . . . . .   | 474        |
| 33.6 v16.09 . . . . .   | 477        |
| 33.7 v16.08 . . . . .   | 479        |
| 33.8 v16.07 . . . . .   | 482        |
| 33.9 v16.06 . . . . .   | 484        |
| 33.10 v16.05 . . . . .  | 487        |
| 33.11 v16.04 . . . . .  | 491        |
| 33.12 v16.03 . . . . .  | 493        |

|                             |            |
|-----------------------------|------------|
| 33.13 v16.02 . . . . .      | 496        |
| 33.14 v16.01 . . . . .      | 500        |
| 33.15 v15.12 . . . . .      | 504        |
| 33.16 v15.11 . . . . .      | 507        |
| 33.17 v15.10 . . . . .      | 510        |
| 33.18 v15.09 . . . . .      | 514        |
| 33.19 v15.08 . . . . .      | 519        |
| 33.20 v15.07 . . . . .      | 522        |
| 33.21 v15.06 . . . . .      | 526        |
| 33.22 v15.05 . . . . .      | 531        |
| 33.23 v15.04 . . . . .      | 534        |
| 33.24 v15.03 . . . . .      | 538        |
| 33.25 v15.02 . . . . .      | 542        |
| 33.26 v15.01 . . . . .      | 546        |
| 33.27 v14.12 . . . . .      | 549        |
| 33.28 v14.11 . . . . .      | 552        |
| 33.29 v14.10 . . . . .      | 553        |
| 33.30 v14.09 . . . . .      | 556        |
| <b>Алфавитный указатель</b> | <b>560</b> |



## Общие сведения

---

### 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, подключающую к веб-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.

### “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.

## 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 нижележащими *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**.

### 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”.

### Поддержка WebGL в браузерах

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

### Браузеры

- Chrome
- Yandex Browser
- Firefox
- Opera
- Safari 8+
- Internet Explorer 11
- Microsoft Edge

### Мобильные платформы

- Android

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

Более подробно поддержка WebGL описана в *соответствующем разделе* главы “Проблемы и решения”.

## 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*

## 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.

## 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, для 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).

## 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 в браузерах перспективных возможностей, 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](#)

## 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”).

## Видео-карты и драйверы

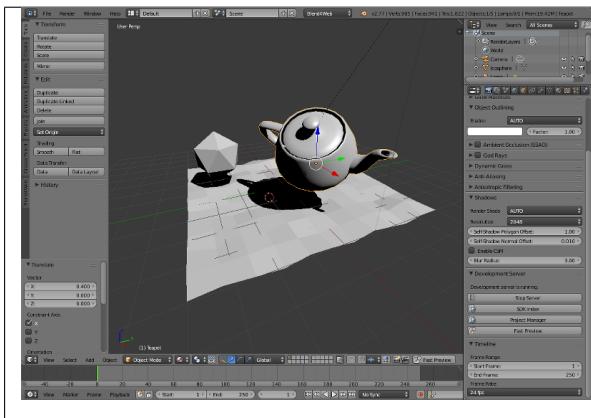
Интерактивная графика реализуется специализированной аппаратной частью современных компьютеров, называемой графическим процессором, который может быть

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*.

### Возможности движка



#### Интеграция с Blender

Создание интерактивных приложений никогда ещё не было таким удобным! Просто создавайте 3D-сцены в Blender и используйте их в Blend4Web с помощью следующих возможностей:

- Простой и быстрый экспорт сцен
- Предварительный просмотр сцены одним кликом
- Полная поддержка настроек материалов и окружения Blender
- Средства создания и управления проектами

|  |  |
|--|--|
|  |  |
| <p><b>Рендеринг</b></p> <p>Blend4Web предлагает большое количество способов сделать Ваши творения красивыми.</p> <p>В том числе:</p> <ul style="list-style-type: none"><li>• 3D-сцены любой сложности</li><li>• Каскады теней</li><li>• Динамическая растительность</li><li>• Системы частиц</li><li>• Уровни детализации (LOD)</li><li>• Постпроцессинговые эффекты</li><li>• Поддержка стереоизображения</li></ul> |  |

---



### Материалы

Мощная и простая в обращении система создания материалов позволяет создать любой материал, от самого простого до фотореалистичного.

Некоторые из её возможностей:

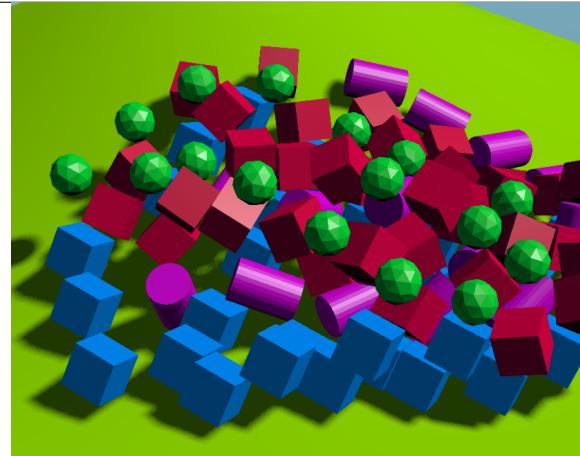
- Нодовые материалы
- Parallax Offset Mapping
- Рендеринг в текстуру
- Видеотекстуры
- Полная поддержка нод материалов Blender
- Отражения и преломления

## Анимация

Blend4Web поддерживает почти все анимационные техники, доступные в Blender, в том числе:

- Скелетная анимация
- Вертексная анимация
- Объектная анимация
- Процедурная анимация травы и крон деревьев
- Анимированные материалы
- Запекание анимации
- Управление анимацией с помощью API или редактора логики





### Физика

Если вы создаёте физическую симуляцию или просто хотите, чтобы ваша сцена смотрелась реалистично, мощная физическая система Blend4Web к вашим услугам.

В числе её возможностей:

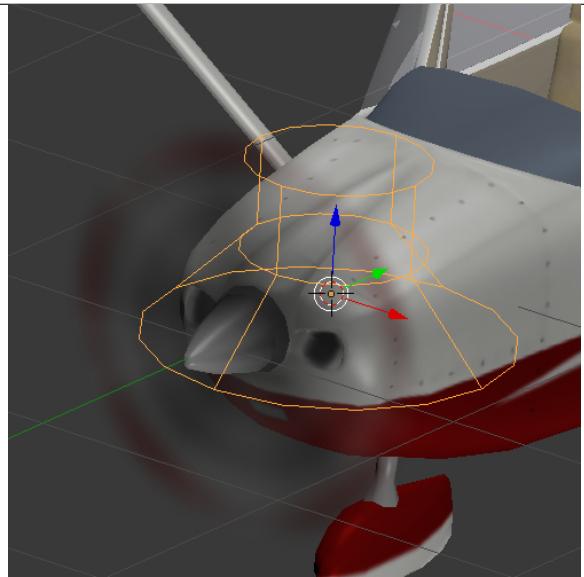
- Столкновения
- Физика твёрдого тела
- Система шарниров
- Трассировка лучей
- Плавающие объекты
- Транспортные средства

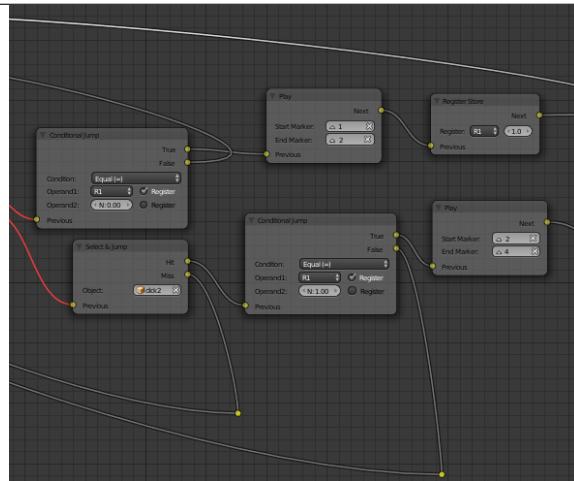
## Звук

Правильно подобранный звук позволяет сделать приложение более интересным и отзывчивым.

Возможности Blend4Web по созданию звуковой картины вашего творения включают:

- Гибкий контроль воспроизведения
- Позиционирование в трёхмерном пространстве
- Эффект Допплера
- Динамический компрессор
- Кроссфейдерная звуковая анимация
- Смешивание в реальном времени





## Редактор логики

С помощью визуального редактора логики вы можете создавать нодовые деревья для контроля поведения сцены и добавления в сцену интерактивности без какого-либо программирования!

Возможности редактора логики:

- Выбор объектов
- Управление анимацией
- Переменные и вычисления
- Обмен данными с сервером

## Программирование

Blend4Web использует JavaScript-сценарии для контроля всех аспектов работы приложения.

Некоторые возможности API Blend4Web:

- Модульная структура
- Событийная модель
- GLSL-шейдеры
- Минификация и обfuscация кода

```
"use strict"

// register the application module
b4w.register("simple_app", function(application) {

    // import modules used by the app
    var m_app      = require("app");
    var m_data     = require("data");

    /**
     * export the method to initialize the app (called at run)
     */
    exports.init = function() {
        m_app.init({
            canvas_container_id: "main_canvas_container",
            callback: init_cb,
            show_fps: true,
            console_verbose: true,
            autoresize: true
        });
    };
});
```

# Установка и обновление

---

## Содержание

- Установка и обновление
  - Установка
    - \* Установка программы *Blender*
    - \* Установка *Blend4Web SDK*
    - \* Установка аддона *Blend4Web*
    - \* Переключение в режим *Blend4Web*
  - Обновление
    - \* Обновление *SDK*
    - \* Обновление аддона
    - \* Обновление сохранённых проектов

## Установка

Существуют две версии фреймворка Blend4Web: Blend4Web SDK и аддон Blend4Web. Какую из них вам следует выбрать для работы, зависит от того, приложения какого типа вы планируете разрабатывать.

Если вы планируете использовать все возможности движка Blend4Web, вам следует установить Blend4Web SDK.

Если же вы планируете разрабатывать небольшие проекты или собираетесь использовать только некоторые возможности движка Blend4Web (например, редактор нормалей), вы можете использовать *аддон Blend4Web* вместо SDK.

Обратите внимание, что аддон поддерживает экспорт сцены только в формат *HTML* и имеет ограниченные возможности. Например, в нём отсутствует *менеджер проектов*, примеры сцен, документация и другие дополнительные ресурсы. Тем не менее, аддон располагает всем необходимым для создания простого приложения.

## Установка программы Blender

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

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

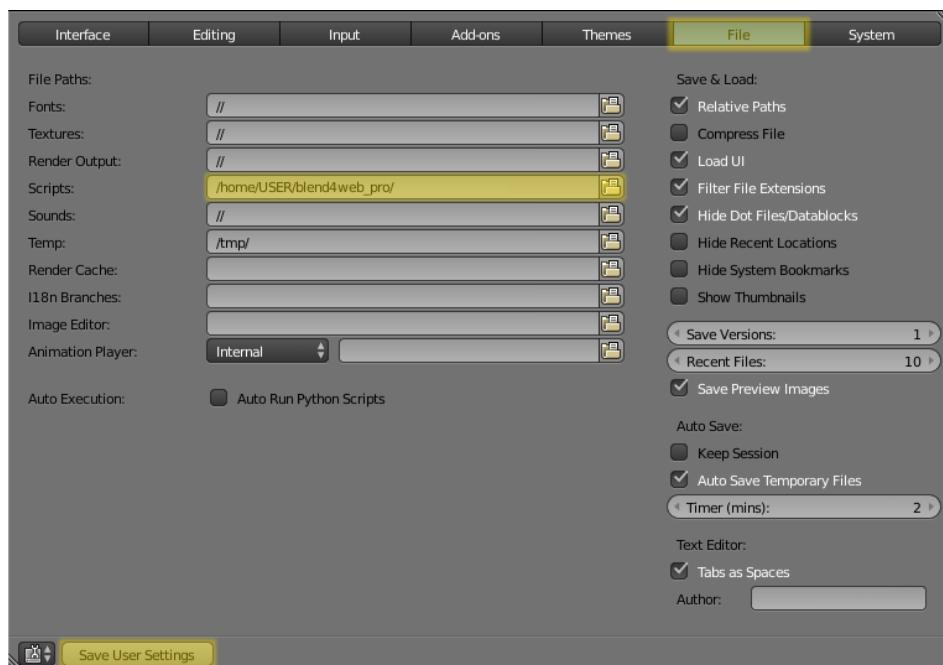
Должна использоваться текущая стабильная версия Blender. Загрузить Blender можно с официального сайта или с [сайта Blend4Web](#).



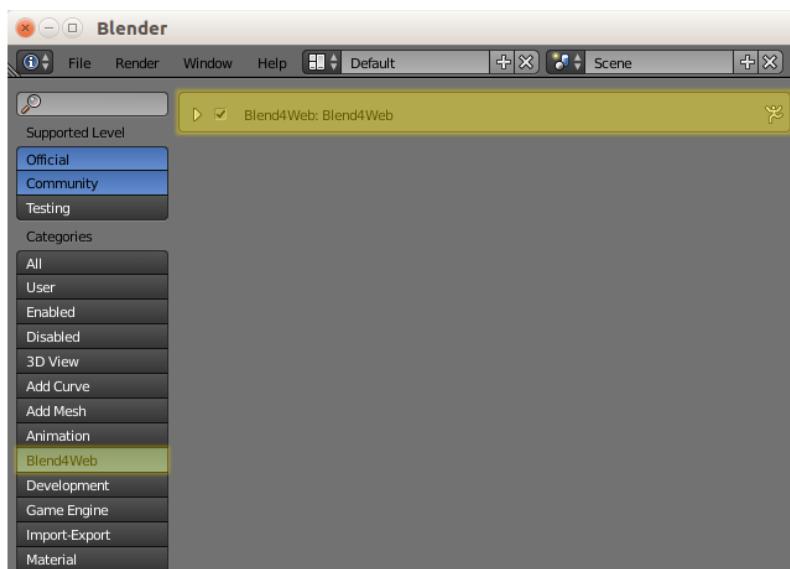
## Установка Blend4Web SDK

Стабильные версии дистрибутива поставляются в виде архива (`blend4web_ce_YY_MM.zip` – бесплатный SDK, `blend4web_pro_YY_MM.zip` – коммерческий SDK). Достаточно распаковать данный архив в любое место на диске.

1. Открыть Blender.
2. Вызвать окно пользовательских настроек `File > User Preferences....`
3. Открыть вкладку `File`.
4. В поле `Scripts` прописать путь к директории SDK.
5. Нажать `Save User Settings`



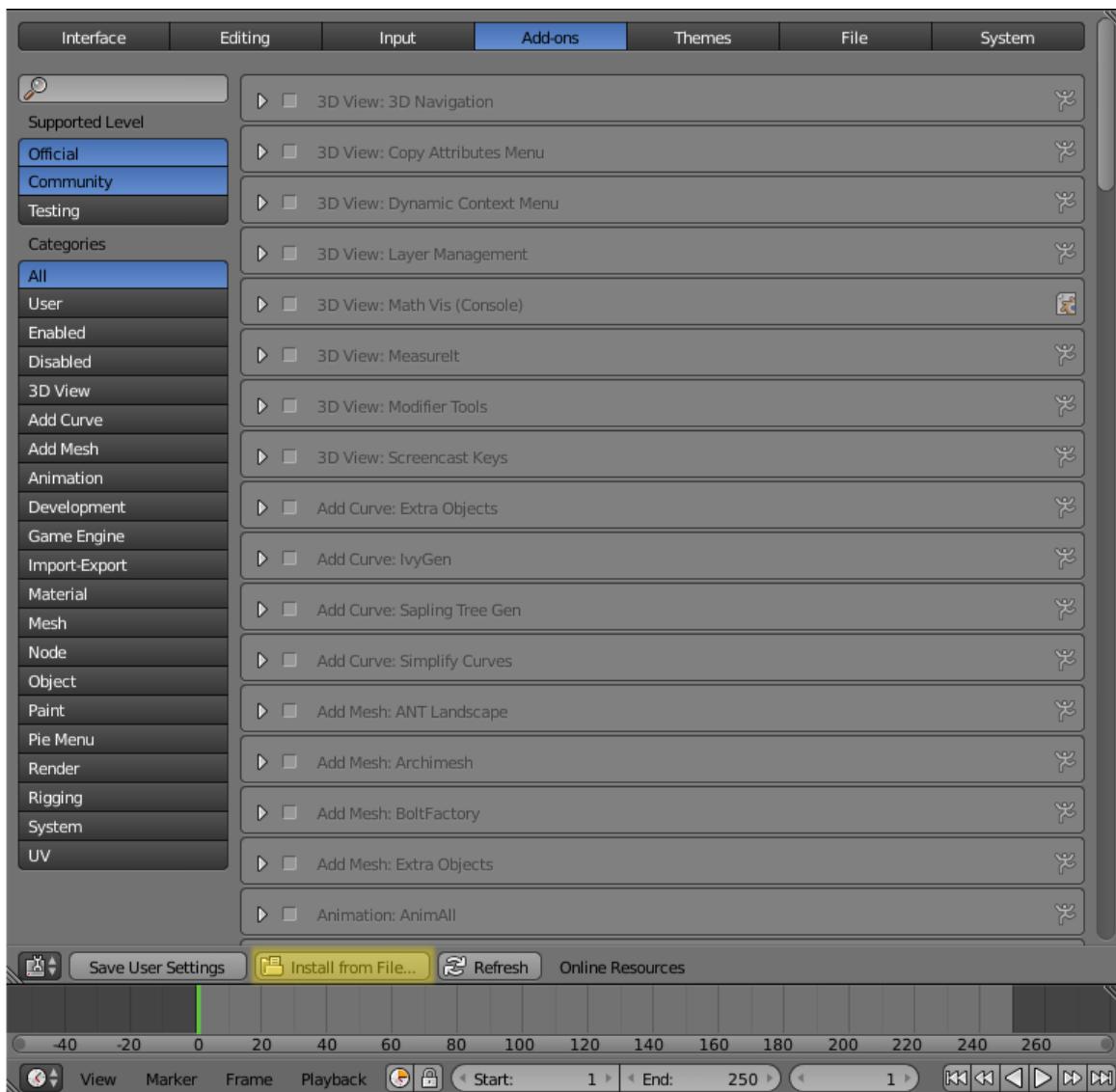
6. Перезапустить Blender.
7. Ещё раз вызвать окно пользовательских настроек **File > User Preferences**....
8. Открыть вкладку **Add-ons**.
9. Активировать аддон *Blend4Web*.
10. Ещё раз нажать **Save User Settings**



#### *Проверка:*

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

## Установка аддона Blend4Web



Устанавливается так же, как любой другой [аддон](#) Blender.

1. Открыть Blender.
2. Вызвать окно пользовательских настроек `File > User Preferences....`
3. Открыть вкладку `Add-ons`.
4. Нажать кнопку `Install From File....`
5. Выбрать архив, содержащий аддон Blend4Web, и нажать кнопку `Install From File....`
6. Нажмите кнопку `Save User Settings`.
7. Перезапустить Blender.
8. Ещё раз вызвать окно пользовательских настроек `File > User Preferences....`

9. Открыть вкладку Add-ons.
10. Найдите аддон *Blend4Web* в списке.
11. Активируйте его, поставив флажок слева от названия аддона.
12. Снова нажмите кнопку Save User Settings.

Теперь всё должно работать.

## Переключение в режим Blend4Web

Выберите профиль настроек Blend4Web в выпадающем меню верхней панели:



## Обновление

### Обновление SDK

Перед обновлением рекомендуется создать резервные копии всех разрабатываемых проектов с помощью *средств экспортации*.

При обновлении SDK следуйте этой инструкции:

1. Загрузите новую версию SDK.
2. Откройте Blender.
3. Откройте окно User Preferences.
4. Откройте вкладку Add-ons.
5. Деактивируйте аддон *Blend4Web*.

---

**Примечание:** Не рекомендуется использовать кнопку Remove для отключения аддона при обновлении SDK, так как это приведёт к полному удалению аддона, в том числе его файлов в SDK.

---

6. Разархивируйте новую версию SDK.

---

**Примечание:** Обновление может производиться одним из двух способов.

Во-первых, вы можете полностью удалить каталог SDK, после чего распаковать каталог с новой версией и прописать путь к ней в настройках Blender, как если бы вы устанавливали SDK с нуля.

Во-вторых, вы можете просто перезаписать устаревшие файлы SDK новыми. Этот подход считается менее “чистым”, но в большинстве случаев не должен приводить к проблемам.

---

7. В окне `User Preferences` откройте панель `File`.

8. В поле `Scripts` пропишите путь к директории SDK.

9. Нажмите кнопку `Save User Settings`.

10. Перезапустить Blender.

11. Снова откройте окно `User Preferences`.

12. Откройте вкладку `Add-ons`.

13. Активируйте аддон `Blend4Web`.

14. Снова нажмите кнопку `Save User Settings`.

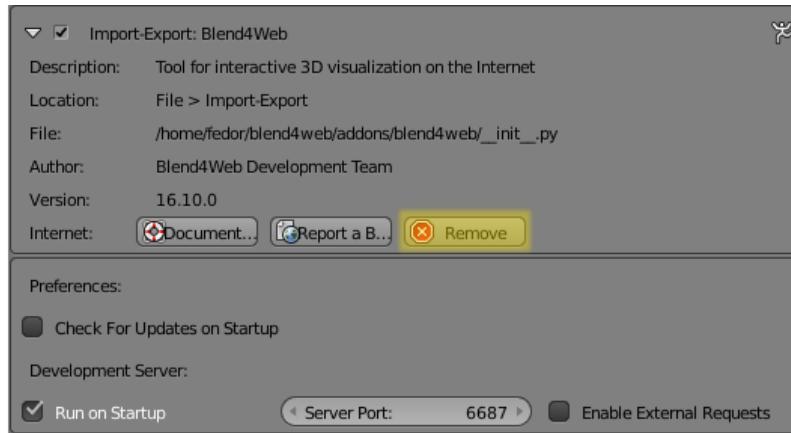
15. Проверьте, всё ли работает правильно.

После того, как обновление выполнено, сохранённые проекты можно импортировать с помощью средств [импорта проектов](#).

## Обновление аддона

В том случае, если вы используете только аддон Blend4Web, следуйте этой инструкции.

Перед установкой новой версии аддона рекомендуется удалить старую. Для этого раскройте информационную панель аддона Blend4Web и нажмите кнопку `Remove`. Удаление старой версии аддона перед установкой новой не обязательно, но рекомендуется для того, чтобы избежать возможных конфликтов.



1. Загрузите архив, содержащий новую версию аддона, и сохраните его в любом месте на жёстком диске.
2. Откройте Blender.
3. Откройте окно **User Preferences**.
4. Откройте панель **Add-ons**.
5. Установите новую версию аддона из загруженного архива.
6. Нажмите кнопку **Save User Settings**.
7. Перезапустить Blender.

Теперь всё должно работать.

## Обновление сохранённых проектов

После обновления SDK (или аддона) вы можете импортировать в Менеджер проектов проекты, экспортанные перед началом обновления. Чтобы сделать это, следуйте инструкции:

1. Откройте Менеджер проектов.
2. Импортируйте проекты с помощью кнопки **Import Project(s)**.
3. Повторно экспортируйте .blend-файлы с помощью ссылки **re-export scenes**.
4. Проверьте импортированные проекты на отсутствующие и неиспользуемые модули с помощью ссылки **check modules**. Если такие модули обнаружены, проблему можно попытаться исправить с помощью кнопки **Update Modules**.
5. Затем используйте команду **build** для того, чтобы собранные версии импортированных проектов нормально работали с новой версией SDK.
6. Если приложение использует тип привязки движка **Copy** или **Compile**, то его версию для разработки тоже необходимо собрать перед тем, как использовать с обновленным SDK (версии для разработки проектов с другими типами привязки движка должны нормально работать и без сборки).



# Рабочий процесс

---

## Содержание

- *Рабочий процесс*
  - *Создание нового проекта*
  - *Создание сцен*
  - *Подготовка трёхмерных сцен в программе Blender.*
    - \* *Отображение сцен в просмотрщике*
  - *Экспорт сцен*
  - *Разработка приложения*
  - *Сборка проекта*
  - *Размещение проекта.*

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. Создание нового проекта.
2. Создание сцен
3. Подготовка трёхмерных сцен в программе Blender.
4. Экспорт сцен.
5. Разработка приложения.
6. Сборка проекта.
7. Размещение проекта.

## Создание нового проекта

Разумеется, работа над проектом должна начинаться с его создания.

Наиболее простой способ - использовать *Менеджер проектов*. Для того, чтобы создать новый проект, нажмите кнопку Create New Project в верхней части главной страницы менеджера.

Другой способ - использовать консольную утилиту *project.py*. Этот подход предоставляет больше возможностей для тонкой настройки проекта и описан *здесь*.

## Создание сцен

Сцены содержатся в blend-файлах. При создании нового проекта менеджер проектов может автоматически создать blend-файл с базовой сценой. Также к проекту можно подключать дополнительные blend-файлы.

Сцены для Blend4Web создаются почти так же, как сцены для самого Blender. Процесс работы над сценой можно разделить на несколько этапов:

1. Создание объектов сцены
2. *Настройка материалов*
3. *Создание анимации*
4. *Создание нодовой логики* для сцены. Этот этап применим только к движку Blend4Web, но не к самому Blender.

Следует заметить, что не все из этих этапов необходимы при работе над проектом. Простая сцена может не содержать анимации или даже материалов (кроме стандартных).

Порядок этапов также не задан жёстко и может изменяться в определённых пределах (например, анимировать объект возможно только после его создания, но материал для него может быть создан и загодя).

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 `projects/имя_проекта/blender`. 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 `projects/имя_проекта/assets`.
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).

## Подготовка трёхмерных сцен в программе Blender.

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

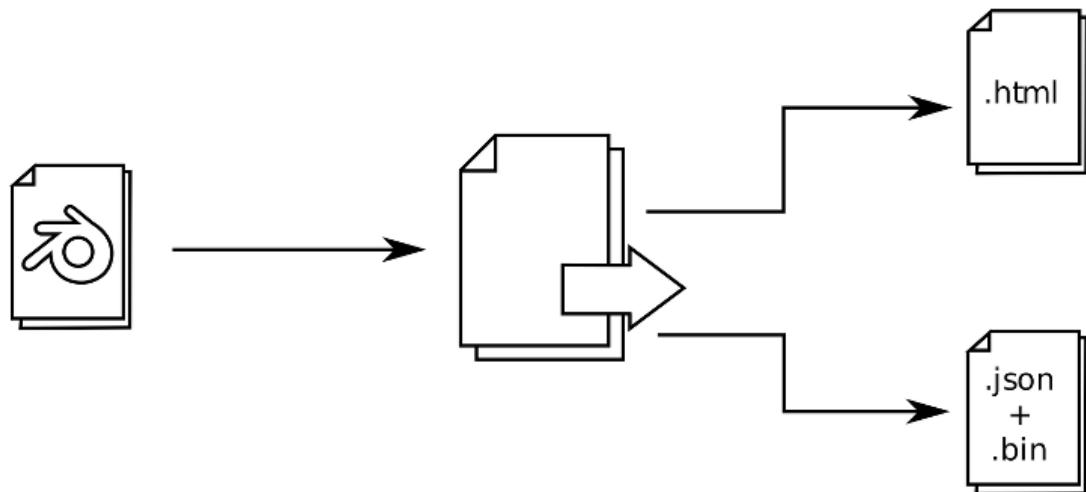
### Отображение сцен в просмотрщике

При использовании *локального сервера разработки* возможно сделать предварительный просмотр сцены используя кнопку *Fast Preview*, расположенную в нижней части окна 3D View Blender'a, а также на панели Render -> Development Server в Blender. В этом случае будет произведён автоматический экспорт сцены во временное хранилище с последующей загрузкой в просмотрщике.

Другой способ - использовать опцию экспорта сцены *Run in Viewer*. В этом случае сцена показывается в просмотрщике сразу после экспорта.

### Экспорт сцен

Процедура экспорта преобразует данные сцены из формата, понятного Blender, в формат, понятный Blend4Web.



Экспорт может осуществляться в два формата: JSON и HTML.

Формат HTML применяется для простых, самодостаточных проектов, в которых все ресурсы упакованы в один файл. Полученный файл легко можно разместить на веб-странице.

Формат JSON также может применяться для создания подобных приложений, но кроме этого он предлагает множество дополнительных возможностей для создания сложных проектов, требующих программирования. Различия между форматами и настройки экспорта описаны в разделе “[Форматы экспорта](#)”.

Для экспорта сцены, выберите в меню `File > Export` опцию Blend4Web (.json) или Blend4Web (.html) либо наберите `b4w export` в меню поиска команд (вызывается по клавише ПРОБЕЛ).

---

**Примечание:** Если сцена использует возможности Blender, не поддерживаемые движком Blend4Web, при экспорте могут возникнуть ошибки. Ошибки экспорта перечислены в [этой таблице](#).

---

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

## Разработка приложения

Этот раздел не актуален для проектов типа Web Player HTML и Web Player JSON.

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](#).

## Сборка проекта

После того, как работа над проектом завершена, его следует собрать, чтобы иметь возможность использовать проект вне SDK.

Сборка проекта запускается при нажатии на ссылку `build project` в [Менеджере проектов](#).

Также, перед сборкой проекта можно [конвертировать медиаресурсы](#), используемые в проекте, чтобы уменьшить его размер.

## Размещение проекта.

После того, как работа над проектом завершена, проект следует разместить. Размещение проекта осуществляется при нажатии на ссылку `deploy project` в списке команд [Operations](#), который находится справа от имени проекта в главном окне [Менеджера проектов](#).

При нажатии на эту ссылку Менеджер проектов выполняет все операции, необходимые для размещения проекта.

Размещённый проект имеет форму архива, содержащего все файлы проекта. Этот архив может быть сохранён на жёсткий диск при помощи стандартных средств сохранения файлов в веб-браузере.

Законченный проект может быть размещён на веб-сервере.

Для этого следует извлечь файлы проекта из архива, который вы загрузили из Менеджера проектов, и загрузить их на сервер (сохраняя внутреннюю структуру каталогов) при помощи FTP, SSH или любого другого протокола, поддерживаемого сервером. После этого проект следует разместить на веб-странице, например, при помощи контейнера `iframe`.

Размещение проектов на веб-сервере описано в *особом разделе* главы *Менеджер проектов*.

# Управление проектами

---

## Содержание

- Управление проектами
  - Менеджер проектов
    - \* Мастер создания проектов
    - \* Создание приложений для Веб-плеера
    - \* Информация о проекте
    - \* Редактирование проекта
    - \* Конфигурационный файл проекта
    - \* Импорт проектов
    - \* Экспорт проектов
    - \* Размещение проекта
  - Расширенное управление проектами
    - \* Зависимости
    - \* Список проектов
    - \* Структура проекта
    - \* Конфигурационный файл проекта (*.b4w\_project*)
    - \* Создание проекта
    - \* Разработка нескольких приложений в составе проекта
    - \* Сборка проекта
    - \* Автоматический экспорт blend-файлов
    - \* Конвертация ресурсов
    - \* Развертывание проекта
    - \* Удаление проекта
    - \* Обновление приложений на новые версии SDK

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 гото-

вых 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.

## Менеджер проектов

Приложение *Project Manager* доступно из раздела *Tools* главной страницы SDK. При запуске приложение выводит список из всех проектов, находящихся в установленном SDK.

| Project Manager |   |  |   |  |   |
|-----------------|---|--|---|--|---|
| Icon            | Project Name / Apps   | Project Directory / Build Directory                            | Blend Files   | JSON Files   | Operations  |
|                 | boat_pf28 (init) [edit] [config]<br>player:boat_pf28.json   | blender/capr/vehicles/boat_pf28                                | blender/capr/vehicles/boat_pf28/ [show]                 | deploy/assets/capr/vehicles/boat_pf28/ [init]<br>boat_pf28.json                        | re-export scenes<br>convert resources<br>deploy project<br>remove project                                   |
|                 | capri (init) [edit] [config]<br>dev: capri.html<br>build: capri.html  | apps/dev/capr<br>deploy/apps/capr                              | blender/capr/ [show]                                    | deploy/assets/capr/ [show]   | build project<br>check modules<br>re-export scenes<br>convert resources<br>deploy project<br>remove project |
|                 | car_bv_eb164 (init) [edit] [config]<br>player:car_bv_eb164.json   | blender/capr/vehicles/car_bv_eb164                             | blender/capr/vehicles/car_bv_eb164/ [show]              | deploy/assets/capr/vehicles/car_bv_eb164/ [init]<br>car_bv_eb164.json                  | re-export scenes<br>convert resources<br>deploy project<br>remove project                                   |
|                 | demos_postprocessing (init) [edit] [config]<br>player: demo_brown.json<br>player: gold_raps.json<br>player: motion_blue.json<br>player: ssao.json | apps/dev/demos_postprocessing<br>deploy/assets/postprocessing  | blender/postprocessing/ [show]                          | deploy/assets/postprocessing/ [show]   | re-export scenes<br>convert resources<br>deploy project<br>remove project                                   |
|                 | farm (init) [edit] [config]<br>dev: farm.html<br>build: farm.html   | apps/dev/farm<br>deploy/app/farm                               | blender/location_agriculture/ [show]                    | deploy/assets/location_agriculture/ [init]<br>exports/location_agriculture.json        | build project<br>check modules<br>re-export scenes<br>convert resources<br>deploy project<br>remove project |
|                 | firstperson (init) [edit] [config]<br>dev: firstperson.html<br>build: firstperson.html  | apps/tutorials/firstperson<br>deploy/app/tutorials/firstperson | blender/tutorials/firstperson/ [show]                   | deploy/assets/tutorials/firstperson/ [init]<br>firstperson.json                        | build project<br>check modules<br>re-export scenes<br>convert resources<br>deploy project<br>remove project |
|                 | mi_34c1 (init) [edit] [config]<br>player: mi_34c1.json<br>player: mi_34c1_demo_scene.json   | blender/capr/vehicles/mi_34c1                                  | blender/capr/vehicles/mi_34c1/ [show]                   | deploy/assets/capr/vehicles/mi_34c1/ [init]<br>mi_34c1.json<br>mi_34c1_demo_scene.json | re-export scenes<br>convert resources<br>deploy project<br>remove project                                   |
|                 | naturemorte (init) [edit] [config]<br>player: naturemorte.json  | blender/fridge<br>blender/fridge/naturemorte                   | blender/fridge/naturemorte/ [show]<br>naturemorte.blend | deploy/assets/fridge/naturemorte/ [init]<br>naturemorte.json                           | re-export scenes<br>convert resources<br>deploy project<br>remove project                                   |
|                 | petgors_tale (init) [edit] [config]<br>dev: petgors_tale.html<br>dev: quest.html<br>build: petgors_tale.html<br>build: quest.html                 | apps/dev/petgors_tale<br>deploy/apps/petgors_tale              | blender/petgors_tale/ [show]                            | deploy/assets/petgors_tale/ [show]   | build project<br>check modules<br>re-export scenes<br>convert resources<br>deploy project<br>remove project |
|                 | space_disaster (init) [edit] [config]<br>dev: space_disaster.html<br>build: space_disaster.html   | apps/dev/space_disaster<br>deploy/apps/space_disaster          | blender/space_disaster/ [show]                          | deploy/assets/space_disaster/ [init]<br>space_disaster.json                            | build project<br>check modules<br>re-export scenes<br>convert resources<br>deploy project<br>remove project |
|                 | watch_scene (init) [edit] [config]<br>player: watch_scene.json  | blender/capr/props/watch_scene                                 | blender/capr/props/watch_scene/ [show]                  | deploy/assets/capr/props/watch_scene/ [init]<br>watch_scene.json                       | re-export scenes<br>convert resources<br>deploy project<br>remove project                                   |

Команды для управления проектами находятся в верхней части страницы.

| Project Manager |                    |                   |                   |                     |      |
|-----------------|--------------------|-------------------|-------------------|---------------------|------|
| Back to Index   | Create New Project | Import Project(s) | Export Project(s) | Hide Stock Projects | Help |

### 5.1. Менеджер проектов

**Back to Index** Возвращает на индексную страницу Blend4Web SDK.

**Create New Project** Открывает мастер *создания новых проектов*.

**Import Project(s)** Открывает диалог *импорта проектов*.

**Export Project(s)** Открывает страницу *экспорта проектов*.

**Hide Stock Projects** Позволяет скрыть шаблонные проекты, присутствующие в SDK по умолчанию. Если они уже скрыты, место этой команды занимает команда **Show Stock Projects**.

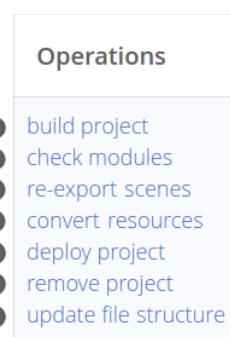
**Help** Вызывает *справку*.

Команды управления конкретным проектом находятся справа от него.

---

**Примечание:** Некоторые команды могут быть недоступны в зависимости от типа проекта.

---



1. Сборка проекта (недоступно для типов проекта WebPlayer JSON и WebPlayer HTML).
2. Проверка отсутствующих или неиспользуемых модулей API (недоступно для типов проекта WebPlayer JSON и WebPlayer HTML).

При активации этой опции менеджер проектов проверяет модули, используемые приложением, и выводит результаты проверки в консоль. Ниже приведены возможные предупреждения:

- (а) При отсутствии одного или нескольких модулей API в папке проекта выводится следующее сообщение:

*Module ‘module\_name’ is missing in the ‘project\_name.html’ project, please include it or run ‘Update Modules’.*

Project Manager

[Back to Projects](#)

Operation Report

Please carefully review the operation results:

Module 'addons/gp\_conf.js' is missing in the 'debugger.html', please include it or run 'Update Modules'.

[Update Modules](#)

Отсутствующие модули могут быть добавлены в проект при помощи кнопки **Update Modules**.

- (а) Если в проекте присутствует один или несколько модулей API, не используемых в программном коде, выводится следующее сообщение:

*Incorrect module ‘module\_name’ in the ‘project\_name.html’, please remove it or run ‘Update Modules’.*

Project Manager

[Back to Projects](#)

Operation Report

Please carefully review the operation results:

Incorrect module 'libs begin.js' in the 'viewer.html', please remove it or run 'Update Modules'.  
Incorrect module 'libs end.js' in the 'viewer.html', please remove it or run 'Update Modules'.

[Update Modules](#)

Модули, не используемые в проекте, могут быть удалены из него при помощи кнопки **Update Modules**.

- (а) В том случае, если никаких проблем не обнаружено, выводится следующее сообщение:

*Module check complete. No problems detected in the ‘module\_name.html’.*

Project Manager

[Back to Projects](#)

Operation Report

Please carefully review the operation results:

Module check complete. No problems detected in the 'flight.html'.

[Update Modules](#)

3. Реэкспорт блендеровских сцен в проекте.
4. *Конвертация медиа ресурсов.*

5. Загрузка архива проекта, готового к размещению на сервере.
6. Удаление проекта со всеми зависимостями.
7. *Обновить файловую структуру.* Доступно только для проектов, созданных при помощи более ранней версии Blend4Web, чем версия 16.12.

---

**Примечание:** Все пути берутся из *конфигурационного файла .b4w\_project*.

---

Рядом с именем проекта расположены ссылки на:

1. *страницу информации о проекте,*
2. *редактор файлов проекта и*
3. *страницу настроек проекта.*

Там же указан тип приложения. Приложение может иметь один из следующих типов:

**Player** Приложение запускается при помощи веб-плеера.

**Dev** Приложение для разработки

**Build** Скомпилированное приложение.

## Мастер создания проектов

Мастер создания новых проектов входит в состав приложения Project Manager и вызывается по ссылке [Create New Project] на главной странице этого приложения.

Create a new project

[Back to Projects](#)
[Help](#)

|  |   |
|--|---|
| <b>1</b><br>Project Name (required)<br><small>for better compatibility, use only letters and underscores</small>                         | <input type="text" value="my_project"/>   |
| <b>2</b><br>Project Title (optional)<br><small>choose a human-readable name</small>  | <input type="text"/>  |
| <b>3</b><br>Project Author / Company (optional)<br><small>choose a human-readable name</small>   | <input type="text"/>  |
| <b>4</b><br>Create Application Starter Files<br><small>create starter html/js/css files</small>  | <input checked="" type="checkbox"/>   |
| <b>5</b><br>Create Scene Starter Files<br><small>create starter blend/json/bin files</small>   | <input checked="" type="checkbox"/>   |
| <b>6</b><br>Use Material Library<br><small>copy material library blend and asset files into your project directory</small>               | <input type="checkbox"/>  |
| <b>7</b><br>Copy Project Management Script<br><small>simplifies project management for users who prefer using command line tools</small> | <input type="checkbox"/>  |
| <b>8</b><br>Application Type<br><small>if unsure, leave the default value<br/>see help for more info</small>                             | Custom Type<br><input checked="" type="radio"/> Copy <input type="radio"/> Compile <input type="radio"/> None<br><br>WebPlayer Type<br><input type="radio"/> Web Player JSON <input checked="" type="radio"/> Web Player HTML   |
| <b>9</b><br>JavaScript Obfuscation Level<br><small>if unsure, leave the default value<br/>see help for more info</small>                 | <input checked="" type="radio"/> Simple <input type="radio"/> Advanced <input type="radio"/> Whitespace Only  |
| <b>10</b><br>Web Player Params<br><small>default Web Player URL params</small>   | <input type="checkbox"/> Show FPS (show_fps)<br><input type="checkbox"/> Auto-rotate camera (autorotate)<br><input type="checkbox"/> Disable social network buttons (no_social)<br><input type="checkbox"/> Background transparency (alpha)<br><input type="checkbox"/> Use compressed textures (compressed_textures) |

[Create Project](#)

Кнопка [Back to Projects] используется для возвращения на главную страницу менеджера проектов, а кнопка [Help] *вызывает справку*.

1. Имя проекта. Текущим именем будут названы директории в “projects/имя\_проекта”, “projects/имя\_проекта/assets”, “projects/имя\_проекта/blender” и “projects/имя\_проекта/build”. Для лучшей совместимости рекомендуется использовать буквы и нижнее подчёркивание.
2. Название проекта. Это имя будет показываться в заголовке веб-браузера.
3. Имя автора проекта.
4. Добавить шаблоны приложения. В директорию проекта “projects/имя\_проекта” будут добавлены стандартные шаблоны приложения: html-файл, css-файл, js-файл.

5. Добавить шаблоны сцены. В директорию “projects/имя\_проекта/assets” будет добавлен стандартный json-файл; в директорию “projects/имя\_проекта/blender” - blend-файл.
6. Использовать библиотеку материалов. Blend-файлы библиотеки материалов будут скопированы в директорию проекта “blender/project\_name”, файлы медиаресурсов - в директорию “projects/project\_name/assets”.
7. Скопировать скрипт менеджера проектов. Скрипт project.py будет скопирован в директорию проекта.
8. Тип собираемого проекта. Здесь возможны варианты:
  - **Copy** - файлы движка прямо копируются в папку приложения из каталога `deploy/apps/common/`.

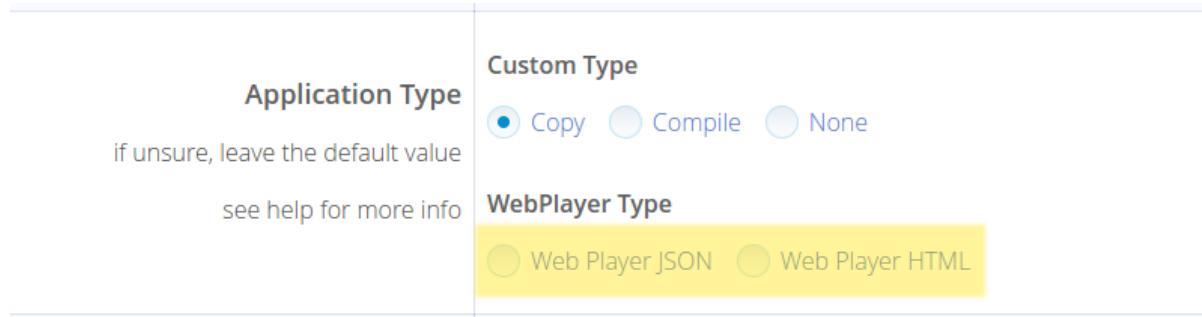
При сборке проекта компилируются только файлы приложения, но не движка (т.е. сам движок не может быть модифицирован);
  - **Compile** - исходники движка компилируются вместе со скриптами приложения.

Этот вариант применяется, если необходимо внести изменения в исходный код самого движка;
  - **None** - движок не копируется в директорию проекта, также менеджер проектов не производит никаких действий при сборке проекта. Разработчик приложения должен самостоятельно производить все необходимые операции;
  - **Web Player JSON** - json-файл, находящийся в проекте, запускается при помощи веб-плеера в составе SDK;
  - **Web Player 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.
10. URL-атрибуты Веб-плеера. Эта вкладка доступна только если параметр **Engine Binding Type** имеет значение **Web Player JSON** или **Web Player HTML**.
  - “Show FPS” - включает счётчик кадров в секунду в правом верхнем углу окна веб-плеера.
  - “Auto-rotate camera” включает автоматическое вращение камеры сразу после загрузки сцены.
  - “Disable social network buttons” используется для отключения кнопок социальных сетей на панели управления.

- “Background transparency” используется для установки прозрачности области отрисовки.
- “Use compressed textures” используется для включения загрузки уменьшенных текстур и текстур с компрессией (формат DDS).

## Создание приложений для Веб-плеера

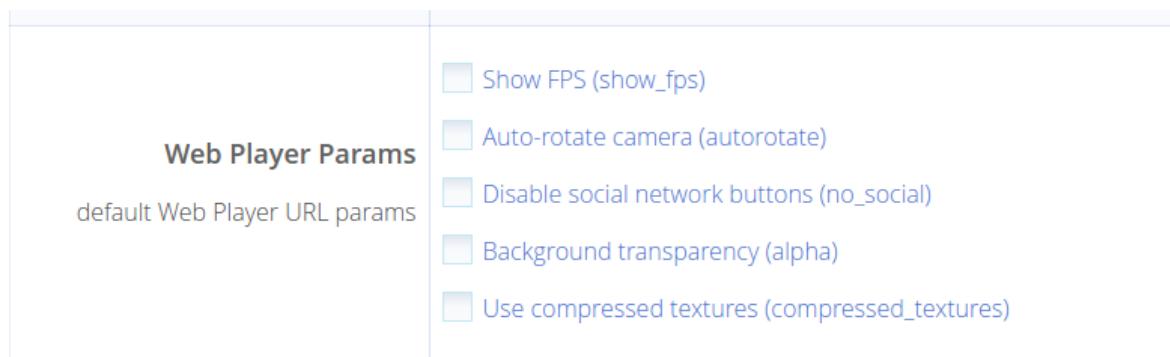
Главное преимущество приложений для Веб-плеера - простота размещения подобных приложений на веб-страницах.



Создать приложение для Веб-плеера несложно. Для этого достаточно при создании нового проекта выбрать опцию Web Player JSON или Web Player HTML на вкладке Application Type.

Проекты для веб-плеера обладают практически тем же набором параметров, что и проекты других типов. Единственное исключение - группа параметров, известная как Web Player Params.

## Настройки проекта



Эта вкладка доступна только в том случае, если включена опция Web Player JSON или Web Player HTML. Перечисленные здесь параметры по сути представляют собой URL-атрибуты, которые веб-плеер будет использовать при запуске проекта.

## Сборка и размещение проекта

После того, как работа над проектом завершена, проект может быть собран и размещён на веб-сервере. Этим операциями посвящён *отдельный раздел*.

## Информация о проекте

Эта страница содержит информацию о выбранном проекте и доступна по ссылке [info] рядом с его названием.

## Project Info

| Project Info                         |                                  |
|--------------------------------------|----------------------------------|
| Back to Projects                     |                                  |
| Help                                 |                                  |
| Config                               |                                  |
| Project Name                         | flight                           |
| Project Title                        | The Island                       |
| Project Author / Company             | Blend4Web                        |
| Project Icon                         | .b4w_icon.png                    |
| Applications                         | [detected automatically]         |
| Engine Binding Type                  | External                         |
| Size                                 | 130 Mb                           |
| Project Path (Development Directory) | apps_dev/flight                  |
| Project Config                       | apps_dev/flight/.b4w_project     |
| Build Directory                      | deploy/apps/flight               |
| Blend Directory(s)                   | blender/flight_over_island       |
| Assets Directory(s)                  | deploy/assets/flight_over_island |
| URL Params                           |                                  |
| JavaScript Obfuscation Level         | advanced                         |
| JS Compilation Pass-Through List     |                                  |
| CSS Compilation Pass-Through List    |                                  |
| Build Ignore List                    |                                  |
| Deployment Assets Directory          |                                  |
| Deployment Assets URL Prefix         |                                  |
| Deployment Ignore List               |                                  |

[Back to Projects](#)
[Help](#)
[Config](#)

## Команды

***Back to Projects*** Возвращает на главную страницу менеджера проектов.

***Help*** Вызывает *справку*.

## Параметры проекта

**Project Name** Имя проекта

**Project Title** Название проекта. Это имя будет показываться в заголовке веб-браузера.

**Project Author / Company** Имя автора проекта или название компании-разработчика проекта.

**Project Icon** Пиктограмма проекта.

**Applications** Приложения проекта.

**Engine Binding Type** Тип собираемого проекта.

**Project Path (Development Directory)** Каталог проекта.

**Project Config** Конфигурационный файл проекта.

**Build Directory** Каталог, содержащий скомпилированный проект.

**Blend Directory(s)** Каталоги, в которых располагаются blend-файлы проекта.

**Assets Directory(s)** Каталоги, в которых располагаются медиаресурсы проекта.

**URL Params** URL-параметры, применяемые при запуске приложения.

**JavaScript Obfuscation Level** Уровень оптимизации javascript.

**JS Compilation Pass-Through List** Список исключений для компиляции JavaScript-файлов проекта.

**CSS Compilation Pass-Through List** Список исключений для компиляции таблиц стилей проекта.

**Build Ignore List** Список исключений для сборок проекта.

**Deployment Assets Directory** Каталог, который будет содержать медиаресурсы размещённого проекта.

**Deployment Assets URL Prefix** URL-префикс пути к папке медиаресурсов размещённого проекта, возвращаемый функцией `get_assets_path()`.

**Deployment Ignore List** Список исключений для размещения проекта.

## Редактирование проекта

По ссылке [edit] рядом с названием проекта доступен простой веб-интерфейс для редактирования файлов проекта.

Edit project source files

[Back to Projects](#)
[New File](#)
[Help](#)

| Files   | Editor |
|---|--------|
| <pre> 1 [info] 2 author = 3 name = my_project 4 title = 5 icon = 6 7 [paths] 8 assets_dirs = projects/my_project/assets; 9 blend_dirs = projects/my_project/blender; 10 blender_exec = blender 11 build_dir = projects/my_project/build 12 deploy_dir = 13 14 [compile] 15 apps = 16 css_ignore = 17 engine_type = compile 18 ignore = 19 js_ignore = 20 optimization = simple 21 use_physics = 22 use_smaa_textures = 23 version = 24 25 [deploy] 26 assets_path_dest = assets 27 assets_path_prefix = assets 28 ignore = 29 override = 30 31 </pre> |        |
| <span><a href="#">Save</a></span> <span><a href="#">Save As...</a></span>   |        |

Левая часть окна редактора содержит список всех .html-, .css- и .js-файлов, находящихся в директории проекта (`projects/<мой_проект>`). Правая часть содержит текст выбранного файла (по умолчанию файл не выбран) с подсвеченным синтаксисом.

---

**Примечание:** Этот интерфейс может использоваться только для редактирования файлов версии для разработки, но не файлов собранной версии.

---

Новые файлы .html, .js и .css могут быть добавлены в директорию проекта при помощи кнопки [New File].

Для сохранения выбранного файла используются кнопки [Save] и [Save As...] в нижней части страницы.

## Конфигурационный файл проекта

Веб-интерфейс, позволяющий просматривать и изменять параметры проекта после его создания.

## Project Config

[Back to Projects](#)
[Help](#)

| Info                                     |  |
|--|--|
| <b>Name</b>                              | Environment Settings Example   |
| <b>Title</b>                             | Environment Settings Example   |
| <b>Author</b>                            |  |
| <b>Icon</b>                              | <br><input type="button" value="Choose File"/> No file chosen   |
| Build                                    |  |
| <b>Apps</b>                              |  |
| <b>Use Physics</b>                       | <input checked="" type="checkbox"/>  |
| <b>Engine Binding Type</b>               | <input type="radio"/> External <input checked="" type="radio"/> Copy <input type="radio"/> Web Player JSON<br><input type="radio"/> Compile <input type="radio"/> None <input type="radio"/> Web Player HTML |
| <b>JavaScript Obfuscation Level</b>      | <input checked="" type="radio"/> Simple <input type="radio"/> Advanced <input type="radio"/> Whitespace Only   |
| <b>JS Compilation Pass-Through List</b>  |  |
| <b>CSS Compilation Pass-Through List</b> |  |
| <b>Build Ignore List</b>                 |  |
| Paths                                    |  |
| <b>Developer Project Path</b>            | apps_dev/Environment Settings Example  |
| <b>Build Project Path</b>                | deploy/apps/Environment Settings Example   |
| <b>Blend Directory(s)</b>                | blender/Environment Settings Example   |
| <b>Assets Directory(s)</b>               | deploy/assets/Environment Settings Example   |
| <b>Blender Exec</b>                      | blender  |
| <b>Deployment Path</b>                   |  |
| Deploy                                   |  |
| <b>Deployment Assets Directory</b>       | assets   |
| <b>Deployment Assets URL Prefix</b>      | assets   |
| <b>Deployment Ignore List</b>            |  |
| <b>Override Deployment Path</b>          | <input type="checkbox"/>   |
| URL GET Params                           |  |
|  |  |

Параметры на этой странице включают все параметры, доступные при создании проекта, а также несколько дополнительных. Некоторые из параметров доступны только для чтения (и не могут быть изменены после того, как проект создан).

### Группа параметров Info

**Name** Имя проекта

Этот параметр доступен только для чтения.

**Title** Название проекта. Это имя будет показываться в заголовке веб-браузера.

**Author** Имя автора проекта или название компании-разработчика проекта.

**Icon** Пиктограмма проекта в Менеджере проектов.

Кнопка **Choose File** применяется для замены пиктограммы. По умолчанию пиктограмма не указана (в этом случае используется логотип Blend4Web).

### Группа параметров Build

**Apps** The list of the project's .HTML files that are used for compiling applications. Each .HTML file is considered a separate application.

**Use Physics** Enables and disables adding physics engine files to the project. Can be turned off if the project does not use physics.

This parameter is always disabled for a WebPlayer HTML type project and always enabled for a WebPlayer JSON type. For other types of projects, it can be set manually.

**Application Type** Тип собираемого проекта.

Этот параметр доступен только для чтения, т.к. проекты разных типов обладают различной внутренней структурой.

**JavaScript Obfuscation Level** Уровень оптимизации JavaScript для компиляции приложения.

This parameter is read-only for the None, WebPlayer HTML and WebPlayer JSON type projects.

**JS Compilation Pass-Through List** Список исключений для компиляции JavaScript-файлов проекта.

**CSS Compilation Pass-Through List** Список исключений для CSS-файлов проекта.

**Build Ignore List** Список исключений для сборок проекта.

## Группа параметров Paths

**Developer Project Path** Путь к версии приложения, предназначеннй для разработки

Этот параметр доступен только для чтения.

**Build Project Path** Путь к скомпилированному приложению.

**Blend Directory(s)** Path(s) to the blend file directory(ies).

**Assets Directory(s)** Путь(и) к каталогу(ам) медиаресурсов.

**Blender Exec** Путь к исполняемому файлу Blender.

**Deployment Path** Путь к каталогу, предназначенному для финального размещения приложения.

## Группа параметров Deploy

**Deployment Assets Directory** Каталог, который будет содержать медиаресурсы размещённого проекта.

**Deployment Assets URL Prefix** URL-префикс пути к папке медиаресурсов размещённого проекта, возвращаемый функцией `get_assets_path()`.

**Deployment Ignore List** Список исключений для размещения проекта.

**Override Deployment Path** Удалить директорию размещения, если она существует.

## URL GET Params

Здесь указываются дополнительные URL-параметры для ссылки на приложение, расположенной в интерфейсе менеджера проектов.

Изменённые параметры проекта сохраняются при помощи кнопки **[Save Config]**, расположенной в нижней части страницы.

## Импорт проектов

Средства импорта проектов вызываются по ссылке **[Import Project(s)]**.

Нажатие на эту ссылку вызывает стандартный диалог открытия файла, где можно выбрать файл проекта, который требуется импортировать.

## Экспорт проектов

Окно экспорта проекта вызывается по ссылке **[Export Project(s)]**.

**Export Project(s)**

|                                  |                                     |                      |
|----------------------------------|-------------------------------------|----------------------|
| <a href="#">Back to Projects</a> | <a href="#">Hide Stock Projects</a> | <a href="#">Help</a> |
|----------------------------------|-------------------------------------|----------------------|

| Select                   | Name         | Title        | Author    |
|--------------------------|--------------|--------------|-----------|
| <input type="checkbox"/> | azure_grotto | Azure Grotto | Blend4Web |
| <input type="checkbox"/> | capri        | Capri        | Blend4Web |
| <input type="checkbox"/> | capri_char   | Capri Char   | Blend4Web |

|                          |           |            |           |
|--------------------------|-----------|------------|-----------|
| <input type="checkbox"/> | webplayer | Web Player | Blend4Web |
| <input type="checkbox"/> | website   | Website    | Blend4Web |

Archive Name: .zip

[Export Project\(s\)](#)

## Команды

**Back to Projects** Возвращает на главную страницу менеджера проектов.

**Hide Stock Projects** Скрывает стоковые проекты. Если они уже скрыты, место этой команды занимает команда [**Show Stock Projects**].

**Help** Вызывает справку.

**Export Project(s)** Экспортирует выбранные проекты.

## Параметры проекта

**Select** Демонстрирует, выбран ли проект для экспорта.

**Name** Имя каталога проекта.

**Title** Заголовок проекта.

**Author** Имя автора проекта.

**Archive Name** Имя архива, в который будут запакованы экспортируемые проекты.

## Размещение проекта

После завершения работы над проектом, выберите опцию `deploy project` на панели **Operations**, на главной странице менеджера проектов. После этого проект будет экспортирован и запакован в архив.

## Project Manager

[Back to Projects](#)

### Operation Report

Please carefully review the operation results:

```
Compressing deployed project
Project Environment Settings Example has been deployed
```

[Download](#)

Этот архив затем может быть загружен при помощи кнопки **Download** и сохранён в папку, которую ваш браузер использует для хранения загруженных файлов. Следует заметить, что, в зависимости от размера архива, загрузка может быть достаточно долгой.

Чтобы разместить проект на веб-странице, распакуйте архив и загрузите файлы, находящиеся в нём на веб-сервер с помощью FTP, SSH или любого другого протокола, поддерживаемого сервером.

**Примечание:** После загрузки на веб-сервер должна сохраняться внутренняя структура файлов архива и их относительные пути.

После этого приложение типа Web Player HTML можно разместить на веб-странице с помощью контейнера **iframe**. Ниже приводится пример HTML-кода, который можно использовать для этого:

```
<!DOCTYPE html>
<html>
<head>
    <title>An Example Application</title>
</head>
<body>
    <iframe width="800" height="500" allowfullscreen src="/projects/myproject/myproject.html"></iframe>
</body>
</html>
```

Размещение JSON-проекта происходит почти так же, но вместо одного HTML-файла используется комбинация веб-плеера и JSON-файла, содержащего саму сцену.

```
<!DOCTYPE html>
<html>
<head>
    <title>Another Example Application</title>
</head>
<body>
    <iframe width="800" height="500" allowfullscreen src="/myproject/webplayer.html?load=/pro...></iframe>
</body>
```

```
</html>
```

## Расширенное управление проектами

Расширенное управление проектами используется опытными разработчиками, которым необходима как большая степень гибкости, так и возможность автоматизировать все рабочие процессы, связанные с разработкой приложений.

Для расширенного управления проектами следует использовать скрипт *project.py* и ручное редактирование конфигурационных файлов *.b4w\_project*.

Скрипт *project.py* расположен в Blend4Web SDK в директории *./apps\_dev/*. Если при создании проекта была выбрана опция *Copy Project Management Script*, то в директории самого проекта будет находиться копия этого скрипта.

### Зависимости

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
```

Для работы скрипта необходимо установить *java* и записать ее в переменную среды *PATH*.

---

**Примечание:** Java по умолчанию входит в Windows-версию SDK. Для Linux и macOS её требуется установить вручную.

---

При *конверсии ресурсов* также используется набор внешних приложений, описанных в *соответствующем разделе*.

### Список проектов

Список всех проектов в SDK доступен посредством следующей команды:

```
python3 project.py -р myproject list
```

## Структура проекта

Структура проекта, озаглавленного my\_project и размещённого в директории ./projects (в корне директории SDK), выглядит следующим образом:

```
blend4web/
    projects/
        my_project/
            assets/
                my_project.json
                my_project.bin
            blender/
                my_project.blend
            build/
                my_project.html
                my_project.min.css
                my_project.min.js
            .b4w_project
            my_project.js
            my_project.css
            my_project.html
```

Как видно, это приложение состоит из 3 различных директорий:

1. my\_project/assets. Содержит медиафайлы (текстуры, звуки, а также .bin- и .json-файлы) сцен проекта.
2. my\_project/blender. Содержит .blend-файлы сцен проекта.
3. my\_project/build. Содержит файлы собранного приложения.

Файл проекта .b4w\_project, а также .js-, .html'- и .css-файлы располагаются в корне каталога my\_project/.

Кроме того, операция развёртывания (deploy) может создавать ещё одну директорию, однако она обычно располагается за пределами SDK и её название и расположение зависит от структуры директорий на конечном сервере.

---

**Примечание:** Начиная с версии 16.12, структура проекта переработана. Проекты, использующие старую структуру, по-прежнему работают, но считаются устаревшими. Рекомендуется обновить структуру таких проектов с помощью кнопки **update file structure**.

Эта кнопка находится на панели Operations в правой части таблицы свойств проекта в главном окне Менеджера проектов. Кнопка доступна только для пользовательских (не стоковых) проектов, использующих устаревшую структуру.

---

## Конфигурационный файл проекта (.b4w\_project)

Конфигурационный файл проекта включает всю необходимую информацию о проекте, в том числе его заголовок, метаданные, каталоги, а также информацию для сборки и размещения приложения.

Конфигурационный файл проекта можно редактировать вручную, но более удобным будет использование интерфейса *редактирования проекта*.

```
[info]
author = Blend4Web
name = my_project
title = MyProject
icon =

[paths]
assets_dirs = projects/my_project/assets;
blend_dirs = projects/my_project/blender;
blender_exec = blender
build_dir = projects/my_project/build
deploy_dir =

[compile]
apps =
css_ignore =
engine_type = external
ignore =
js_ignore =
optimization = simple
use_physics =

[deploy]
assets_path_dest = assets
assets_path_prefix = assets
ignore =
override =
```

Это стандартный конфигурационный INI-файл, состоящий из разделов, включающих параметры и их значения.

### Раздел [info]

Содержит метаданные проекта:

*author* Имя автора проекта или название компании-разработчика проекта.

*name* Имя проекта

*title* Название проекта. Это имя будет показываться в заголовке веб-браузера.

*icon* Пиктограмма проекта.

## Раздел [paths]

Содержи пути проекта:

*assets\_dirs* Каталоги, в которых располагаются медиаресурсы проекта.

*blend\_dirs* Каталоги, в которых располагаются blend-файлы проекта.

*blender\_exec* Путь к исполняемому файлу Blender.

*build\_dir* Каталог, содержащий скомпилированный проект.

*deploy\_dir* Каталог развёртывания проекта.

## Раздел [compile]

*apps* Приложения проекта.

*css\_ignore* Список исключений для компиляции таблиц стилей проекта.

*engine\_type* Тип собираемого проекта.

*ignore* Список исключений для сборок проекта.

*js\_ignore* Список исключений для компиляции JavaScript-файлов проекта.

*optimization* Уровень оптимизации javascript.

*use\_physics* Указывает, использует ли проект физику. Использования физики включено по умолчанию.

*use\_smma\_textures* Указывает, использует ли проект текстуры SMMA. Не используется на данный момент.

*version* Версия проекта.

## Раздел [deploy]

*assets\_path\_dest* Каталог, который будет содержать медиаресурсы размещённого проекта.

*assets\_path\_prefix* URL-префикс пути к папке медиаресурсов размещённого проекта, возвращаемый функцией `get_assets_path()`.

*ignore* Список исключений для размещения проекта.

*override* Заменить существующую директорию вывода при развёртывании. Использовать с осторожностью.

## Раздел [url\_params]

Необязательный раздел настроек проектов для Веб-плеера. Содержит *URL-параметры*, применяемые при запуске приложения.

## Создание проекта

```
./project.py init my_project
```

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*).
- **-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 my_project
```

Эта команда создаст директорию *my\_project*, внутри которой будут располагаться следующие файлы: *my\_project.js*, *my\_project.css*, *my\_project\_dev.html* и *.b4w\_project*.

Файл *.b4w\_project* будет выглядеть следующим образом:

```
[info]
author = Blend4Web
name = my_project
title = MyProject
icon =

[paths]
assets_dirs = projects/my_project/assets;
blend_dirs = projects/my_project/blender;
blender_exec = blender
build_dir = projects/my_project/build
```

```

deploy_dir =

[compile]
apps =
css_ignore =
engine_type = external
ignore =
js_ignore =
optimization = simple
use_physics =
use_smaa_textures =
version =

[deploy]
assets_path_dest = assets
assets_path_prefix = assets
ignore =
override =

```

## Разработка нескольких приложений 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;
...

```

Если поле `apps` пусто, то каждый `html` файл из директории проекта будет считаться отдельным приложением.

## Сборка проекта

```
python3 project.py -p my_project compile
```

Собирает проект в директории `projects/my_project/build`.

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`.
- `v | --version`" добавляет версию к адресам скриптов и стилей.

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.

## Автоматический экспорт blend-файлов

```
python3 project.py -p my_project 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.

## Конвертация ресурсов

```
python3 project.py -p my_project 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.

Более подробно о процессе конвертации ресурсов сказано в *соответствующем разделе*.

## Развертывание проекта

```
python3 project.py -p my_project deploy DIRECTORY
```

Сохраняет проект во внешнюю директорию со всеми необходимыми зависимостями.

Доступные опции:

- `e | --assets-dest`" задаёт директорию для файлов ресурсов (по умолчанию "assets")

- E | --assets-prefix" задаёт URL-префикс для файлов ресурсов (по умолчанию "assets")
- o | --override" удаляет директорию, если она существует.
- s | --assets" записывает в конфигурационный файл путь к файлам ресурсов.
- t | --engine-type" (необязательная) записывает в конфигурационный файл тип движка.

## Удаление проекта

```
python3 project.py -p my_project remove
```

Удаляет проект. Удаляемые директории берутся из конфигурационного файла.

## Обновление приложений на новые версии SDK

В процессе обновления на новые версии движка чаще всего возникает две проблемы:

1. Несоответствие модулей новой и старой версии движка.
2. Несоответствие старых и новых API движка.

Для того, чтобы обновить список модулей, подключаемых в необfuscированной версии приложений, необходимо перейти в директорию с исходными файлами приложений `apps_dev/my_project`. 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>
```

Несоответствие API решается рефакторингом кода приложения. Все изменения подробно описываются в [замечаниях к релизам](#).

# Просмотрщик сцен

---

## Содержание

- *Просмотрщик сцен*
  - *Навигация*
  - *Боковая панель*
    - \* *Информационное табло*
    - \* *Базовые кнопки управления*
    - \* *Дополнительные панели*
      - *Tools & Debug*
      - *Animation*
      - *NLA*
      - *Shape Keys*
      - *Materials*
      - *Lighting*
      - *Ambient*
      - *Shadows*
      - *Sky*
      - *Sea*
      - *Mist*
      - *Wind*
      - *Wind Bending*
      - *Screen Space Amb Occlus*
      - *God Rays*
      - *Bloom*
      - *Motion Blur*
      - *Glow Materials*
      - *Depth of Field*
      - *Color correction*
      - *Audio*
      - *Stereo View*
      - *Input Devices*
    - *Индикаторы*

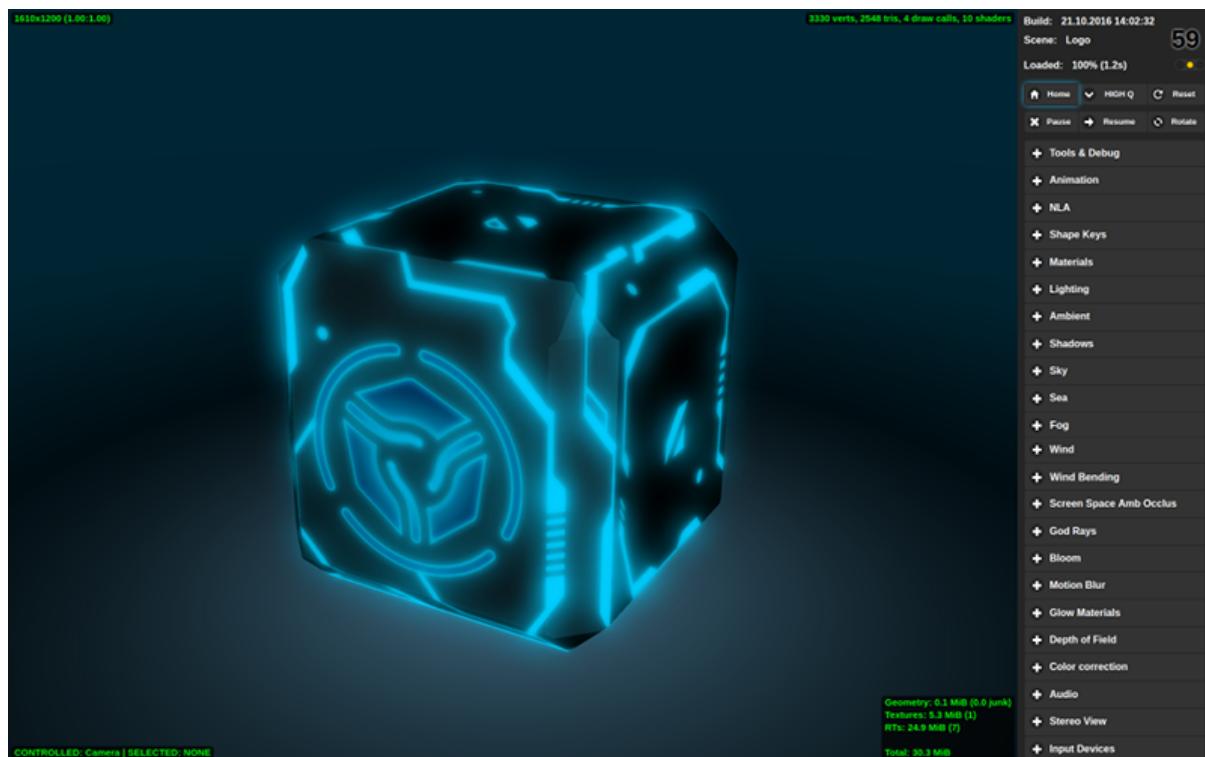
*Запуск просмотрщика сцен.*

## 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).

## 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.



Список элементов управления сверху вниз

## Информационное табло

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

**Scene** Название загруженной сцены. При наведении курсора мыши всплывает путь к файлу.

**Loaded** Процент и время загрузки.

## Базовые кнопки управления

**Home** Загружает последнюю сцену, экспортированную из Blender с помощью кнопки **Fast Preview**, или сцену по умолчанию, если Просмотрщик запущен не из Blender.

**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.

**Rotate** Включает и выключает автоматический поворот камеры вокруг сцены. По умолчанию отключён.

## Дополнительные панели

### Tools & Debug

На панели находится ряд инструментов отладки сцен, в том числе:

**Overview Outlining Mode** Активация этого параметра делает все объекты сцены *доступными для выбора* и включает проигрывание *анимации подсветки* при выборе объекта. Этот параметр включен по умолчанию.

**Auto View Mode** При нажатии на эту кнопку Просмотрщик загружает все сцены из списка **Scenes** с интервалом в одну секунду. Может применяться для проверки всего доступного контента на наличие консольных ошибок.

**Canvas Resolution Factor** Применяется для увеличения разрешения canvas'a. Значение этого параметра может меняться от 1 до 10. Значение по умолчанию равно 1.0 (разрешение canvas'a совпадает с физическим разрешением окна Просмотрщика).

---

**Примечание:** Максимальное доступное разрешение может быть ограничено веб-браузером.

---

**Make Screenshot** Применяется для снятия скриншотов окна Просмотрщика.

Скриншоты сохраняются в той же директории, которую веб-браузер использует для сохранения загруженных файлов.

**Debug View Modes** Список различных режимов визуализации, применяемых для отладки. Доступные режимы:

**NONE** - стандартный режим визуализации. Используется по умолчанию.

**OPAQUE WIREFRAME** - сцена окрашена плоским белым цветом с тёмной каркасной сеткой. Цвет сетки может быть настроен с помощью параметра *Wireframe Edge Color*.

**TRANSPARENT WIREFRAME** - сцена визуализируется так же, как в нормальном режиме, но с добавлением каркасной сетки. Как и в предыдущем режиме, цвет сетки может быть настроен с помощью параметра *Wireframe Edge Color*.

**FRONT BACK VIEW** - в этом режиме внешние и внутренние стороны полигонов окрашиваются разными цветами.

**BOUNDINGS** - в этом режиме к изображению добавляются каркасные сферы, окружающие *объекты* и *меши*. Сфера могут иметь синий (для статических объектов и мешей) или красный (для динамических объектов) цвет.

**CLUSTER VIEW** - в этом режиме каждый объект, принадлежащий к определённому кластеру (используемому для *батчинга*) окрашивается особым цветом.

**BATCH VIEW** - в этом режиме каждый батч окрашен особым (плоским) цветом. Цвета, используемые для визуализации, могут быть заменены на другие, случайно выбранные, цвета с помощью кнопки *Change Colors*.

**RENDER TIME** - в этом режиме сцена визуализируется при помощи разных цветов, отображающих сложность различных участков сцены. Наиболее простые для рендеринга части окрашиваются зелёным цветом, более сложные - жёлтым, а потребляющие наибольшее количество вычислительных ресурсов - красным.

**Change Colors** Случайным образом меняет цвета, используемые в режимах **CLUSTER VIEW** и **BATCH VIEW**. Применяется в случаях, когда части изображения трудно отличить друг от друга из-за того, что цвета, выбранные движком, слишком похожи.

**Render Time Threshold (ms)** Задаёт “пороговое” время для режима визуализации **RENDER TIME**. Если рендеринг какого-либо объекта занимает больше времени, чем указано этим параметром, этот объект будет окрашен красным цветом. Значение по умолчанию - 1.5 мс.

**Wireframe Edge Color** Используется для изменения цвета каркасной сетки в режимах **OPAQUE WIREFRAME** и **TRANSPARENT WIREFRAME**.

| SCENE "Scene"       |                |       |           |             |          |              |
|---------------------|----------------|-------|-----------|-------------|----------|--------------|
| Active              | Subscene       | Lamps | Size      | RenderCalls | Time     |              |
| (✓)                 | SHADOW_CAST    | 1     | 2048x2048 | 1 of        | 1        | 0.110        |
| (✓)                 | SHADOW_RECEIVE | 1     | 1725x925  | 2 of        | 2        | 0.446        |
| (✓)                 | MAIN_OPAQUE    | 1     | 1725x925  | 2 of        | 2        | 1.160        |
| (✓)                 | COLOR_PICKING  | 1     | 1x1       | 0 of        | 2        | 0.000        |
| (✗)                 | MAIN_BLEND     | 1     | 1725x925  | 0 of        | 0        | 0.000        |
| (✗)                 | DEBUG_VIEW     | 0     | 1725x925  | 0 of        | 4        | 0.000        |
| (✗)                 | OUTLINE_MASK   | 1     | 1725x925  | 0 of        | 2        | 0.000        |
| (✗)                 | POSTPROCESSING | 0     | 863x463   | 0 of        | 1        | 0.000        |
| (✗)                 | POSTPROCESSING | 0     | 863x463   | 0 of        | 1        | 0.000        |
| (✗)                 | POSTPROCESSING | 0     | 431x231   | 0 of        | 1        | 0.000        |
| (✗)                 | POSTPROCESSING | 0     | 431x231   | 0 of        | 1        | 0.000        |
| (✓)                 | OUTLINE        | 0     | 1725x925  | 1 of        | 1        | 1.401        |
| (✓)                 | ANTIALIASING   | 0     | 1725x925  | 1 of        | 1        | 0.520        |
| <hr/>               |                |       |           |             |          |              |
| <b>TOTAL ACTIVE</b> |                |       |           | <b>7 of</b> | <b>9</b> | <b>3.637</b> |

### HUD Info

Показывает список визуализируемых подсцен и их параметров, таких, как:

**SCENE** Имя текущей сцены.

**Active** Показывает, активна ли подсцена.

**Subscene** Имя подсцены.

**Lamps** Количество источников света в подсцене.

**Size** Разрешение, используемое для рендеринга подсцены.

**RenderCalls** Количество вызовов отрисовки в подсцене.

**Time** Время, которое движок тратит на рендеринг подсцены.

**TOTAL ACTIVE** Количество вызовов отрисовки во всех доступных подсценах и время, затрачиваемое движком на их рендеринг.

Включение этого параметра может снизить скорость рендеринга сцены. По умолчанию он выключен.

**GL Debug** Активация этого параметра заставляет движок обрабатывать низковолневые ошибки. Это может существенно снизить производительность. При первом запуске Просмотрщика этот параметр деактивирован, а при последующих запусках приложение запоминает его состояние.

**Min Capabilities Mode** При активации этого параметра Просмотрщик запускает сцену так, как если бы она работала на системе с низкой производительностью (например, на мобильном устройстве). Это позволяет проверить, какие материалы могут не работать так, как задумано, на низкопроизводительной системе. По умолчанию выключено.

## 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).

Доступные параметры:

**Play All** Проигрывает все доступные анимации.

**Stop All** Останавливает все анимации, которые воспроизводятся в настоящий момент.

**Object** Список присутствующих в сцене объектов, имеющих анимации.

**Animation Slot** Список всех доступных слотов, которые могут использоваться для применения какой-либо анимации к объекту.

**Animation** Список всех анимаций, доступных для выбранного объекта.

**Cyclic** При активации этого параметра анимация будет воспроизводиться циклически. В противном случае анимация остановится после одного проигрывания. По умолчанию включён.

**Range** Длинна выбранной анимации.

**Status** Состояние выбранное анимации, т.е. проигрывается ли она или остановлена.

**Playing Animation > Play** Воспроизводит выбранную анимацию.

**Playing Animation > Stop** Останавливает выбранную анимацию.

**Set Frame** Показывает текущий кадр анимации. Может применяться для задания текущего кадра.

**Skeletal Mix Factor** Задаёт степень смешивания скелетных анимаций. Значение по умолчанию - 1.0.

## NLA

Управление NLA-анимацией (если таковая присутствует в сцене). Для работы этой вкладки должна быть включена опция NLA в настройках сцены.

Доступные параметры:

**Range** Начальный и конечный кадры NLA-анимации.

**Status** Состояние NLA-анимации, т.е. проигрывается ли она или остановлена.

**Playing NLA > Play** Воспроизводит NLA-анимацию.

**Playing NLA > Stop** Останавливает проигрываемую NLA-анимацию.

**Set Frame** Если анимация остановлена, этот параметр может применяться для задания текущего кадра.

## Shape Keys

Управление доступными ключами формы. При просмотре моделей, обладающих ключами формы, можно выбрать объект, один из ключей и степень влияния ключа на объект.

Доступные параметры:

**Object** Список всех присутствующих в сцене объектов, имеющих ключи формы.

**Shape Key** Список всех ключей формы, доступных для выбранного объекта.

**Set Value** Этот параметр задаёт степень влияния выбранного ключа формы на форму выбранного объекта. Значение может изменяться от 0 до 1.

## Materials

Настройка свойств материалов. Материал выбирается с помощью выпадающего меню. В настоящий момент поддерживается изменение ограниченного ряда свойств, и только для стековых (не нодовых) материалов, присвоенных динамическим объектам (нодовые материалы и статические объекты не поддерживаются).

Доступные параметры:

**Material** Список всех стековых материалов, присутствующих в сцене.

**Color** Задаёт базовый цвет выбранного материала.

**Reflectivity** Задаёт силу зеркального отражения для выбранного материала. Значение по умолчанию - 0.

**Fresnel** Сила эффекта Френеля для отражения. По умолчанию равна нулю.

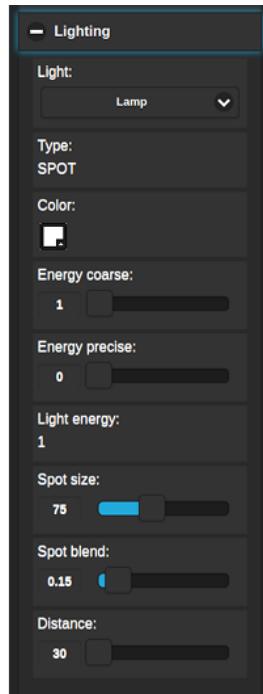
**Fresnel Factor** Фактор эффекта Френеля. По умолчанию равен нулю.

**Parallax Scale** Фактор влияния эффекта рельефа. Значение этого параметра может изменяться от 0 до 0.1, значение по умолчанию - 0.

**Parallax Steps** Количество итераций для расчёта эффекта рельефа. Значение этого параметра может изменяться от 1 до 30, по умолчанию он равен 5.

## 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.



Доступные параметры:

**Light** Список всех источников света, присутствующих в сцене.

**Type** Тип выбранного источника света. Этот параметр не может быть изменён в Просмотрщике.

**Color** Базовый цвет выбранного источника света. По умолчанию используется белый цвет (1; 1; 1).

**Energy Coarse** Этот параметр задаёт целую часть значения энергии источника света. Значение может изменяться от 0 до 100. Значение по умолчанию - 1.

**Energy Precise** Этот параметр задаёт дробную часть значения энергии источника света. Значение может изменяться от 0 до 1, по умолчанию равно 0.

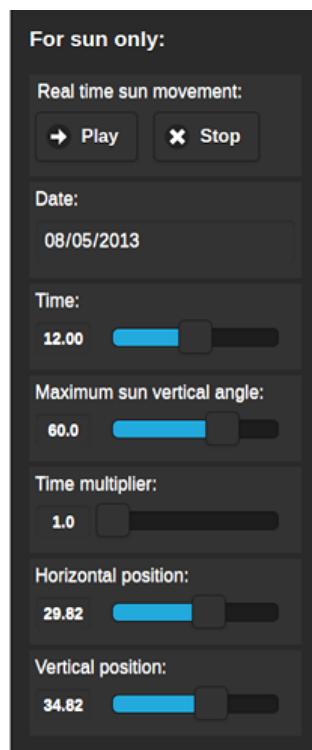
**Light Energy** Энергия источника света. Это значение вычисляется как сумма параметров Energy Coarse и Energy Precise. Если значения этих параметров совпадают с их значениями по умолчанию, энергия будет равна 1.

**Spot Size** Угол светового потока, излучаемого источником света. Значение по умолчанию - 75. Этот параметр доступен только для источников типа Spot.

**Spot Blend** Степень размытия границы пятна прожектора. Значение по умолчанию - 0.15. Этот параметр также доступен только для источников света типа Spot.

**Distance** Этот параметр задаёт расстояние (в метрах), на котором свет, излучаемый источником, будет иметь яркость, равную половине значения параметра Light Energy. По умолчанию равен 30.

Следующие настройки доступны только для источников света типа Sun:



**Real Time Sun Movement > Play** Включает движение солнца в реальном времени.

**Real Time Sun Movement > Stop** Останавливает движение солнца в реальном времени.

**Date** Задаёт текущую дату.

**Time** Этот параметр задаёт текущее время дня или ночи. По умолчанию равен 12.0.

**Maximum Sun Vertical Angle** Этот параметр задаёт максимальный вертикальный угол для источника света типа Sun. Значение по умолчанию - 60.

**Time Multiplier** Значение этого параметра задаёт скорость течения времени. Значение по умолчанию - 1.0 (время в сцене течёт с той же скоростью, что и в действительности).

**Horizontal Position** Горизонтальная позиция солнца на небесной сфере. Значение по умолчанию - 29.82.

**Vertical Position** Вертикальная позиция солнца на небесной сфере. Значение по умолчанию - 34.82.

## Ambient

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

Настройки, доступные для освещения от окружающей среды, совпадают с соответствующими настройками в Blender. Более подробно они описаны в *отдельном разделе*.

## Shadows

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

Настройки теней описаны в *соответствующем разделе* данного руководства.

## Sky

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

Просмотрщик сцен предоставляет пользователю доступ ко всем параметрам Blender, используемым для настройки процедурного неба. Эти параметры описаны в *соответствующем разделе*.

## Sea

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

Настройки рендеринга воды описаны в главе “*Природные эффекты*”. Просмотрщик использует те же настройки, что и сам движок Blend4Web.

## Mist

Настройка параметров тумана.

Просмотрщик сцен использует стандартные настройки рендеринга тумана, описанные в *отдельной главе*.

## Wind

Настройка параметров ветра, включая направление и силу.

В Просмотрщике используется тот же набор параметров, что и в Blender. Эти параметры описаны в *соответствующей главе*.

## Wind Bending

Настройка параметров эффекта Wind Bending.

Аналогичный набор параметров Wind Bending доступен в Blender и описан в главе “*Природные эффекты*”

## Screen Space Amb Occlus

Настройка параметров взаимного затенения.

Все доступные настройки описаны в главе “*Спецэффекты*”.

## God Rays

Настройка параметров эффекта сумеречных лучей.

Все параметры этого эффекта описаны в *соответствующем разделе*.

## Bloom

Настройка параметров эффекта глубины резкости камеры.

Этот эффект использует те же параметры, что и в Blender. Более подробно они описаны в *соответствующем разделе*.

## Motion Blur

Настройка эффекта размытия изображения при движении.

Параметры, используемые для настройки эффекта размытия, совпадают с таковыми в Blender. Они описаны в главе “*Спецэффекты*”.

## Glow Materials

Настройка эффекта свечения материалов.

Настройки этого эффекта описаны в *соответствующем разделе*.

## Depth of Field

Настройка параметров эффекта глубины резкости камеры.

Настройки описаны в *соответствующем разделе*.

## Color correction

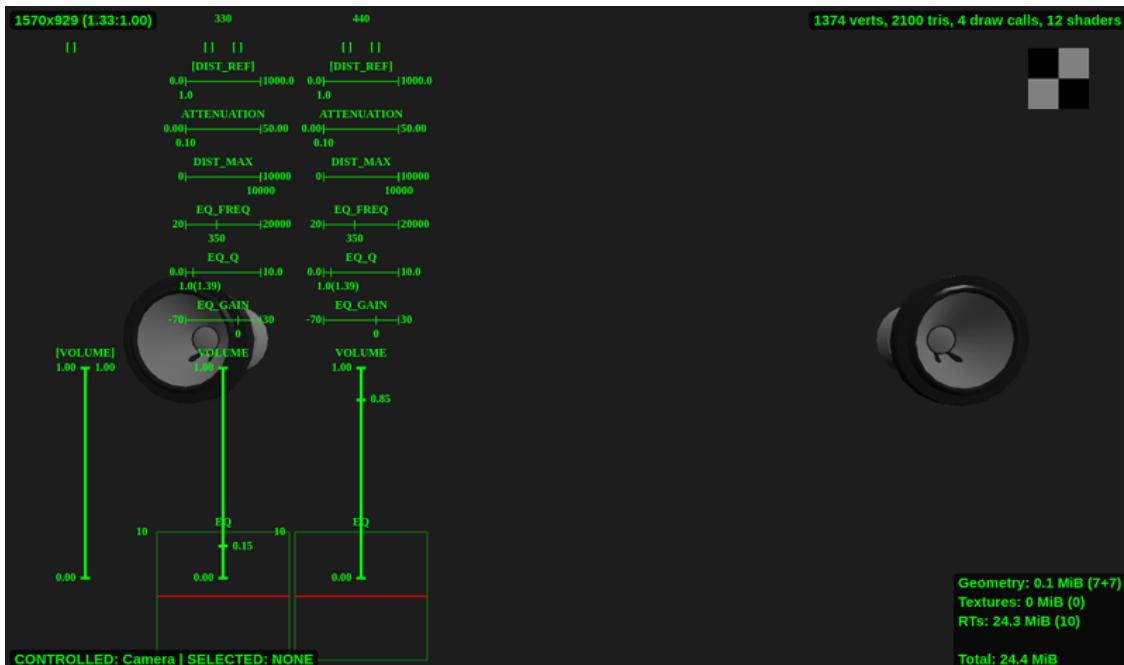
Настройка параметров цветовой коррекции, включая яркость, контрастность, экспозицию и насыщенность.

В самом Blender для цветокоррекции используется такой же набор параметров. Описание этих параметров находится в главе “*Работа с цветом*”.

## Audio

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

**Mixer** Включает эквалайзер, демонстрирующий различные параметры (в т.ч. громкость, частоту, расстояние, затухание) источников звука, присутствующих в сцене, и позволяет пользователю настраивать эти параметры с помощью дополнительной цифровой клавиатуры. Подробнее доступные параметры описаны в *соответствующем разделе* главы “Звук”.



По умолчанию этот параметр деактивирован.

## Stereo View

На панели находится переключатель режима стерео-изображения.

Доступные параметры:

**Stereo Types** Список доступных режимов рендеринга стереоизображения. На сегодняшний день доступно три режима:

NONE - стереорендеринг не используется.

ANAGLYPH - используется *анаглифное изображение*.

HMD - используется рендеринг для *шлемов виртуальной реальности*. Этот режим доступен только в том случае, если соответствующее устройство подключено к системе.

По умолчанию используется вариант NONE.

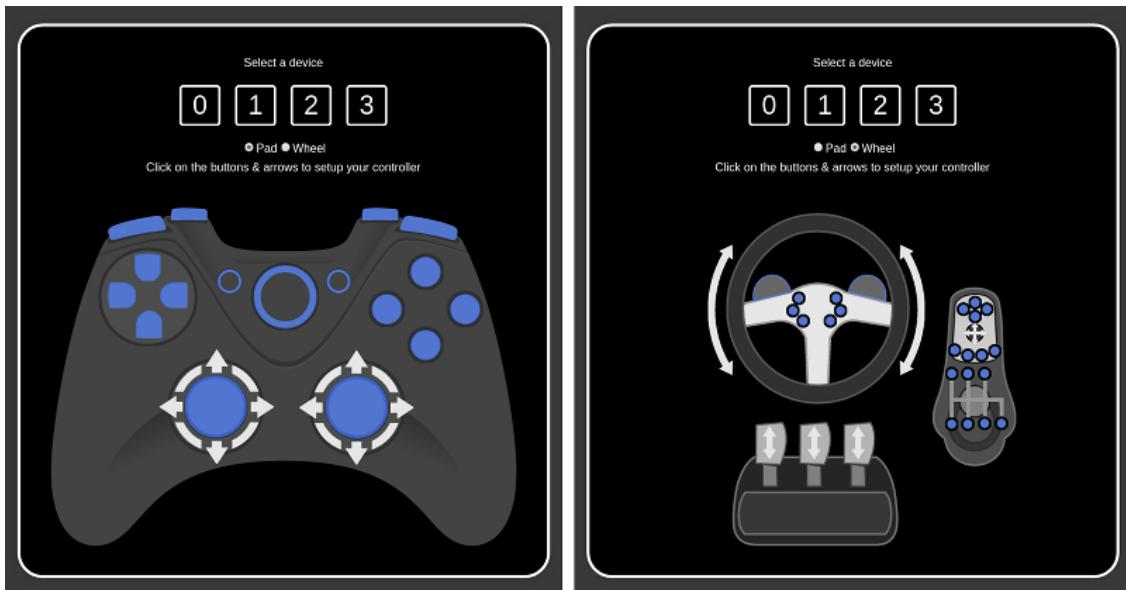
**HMD Settings** Показывает *окно настроек HMD*.

## Input Devices

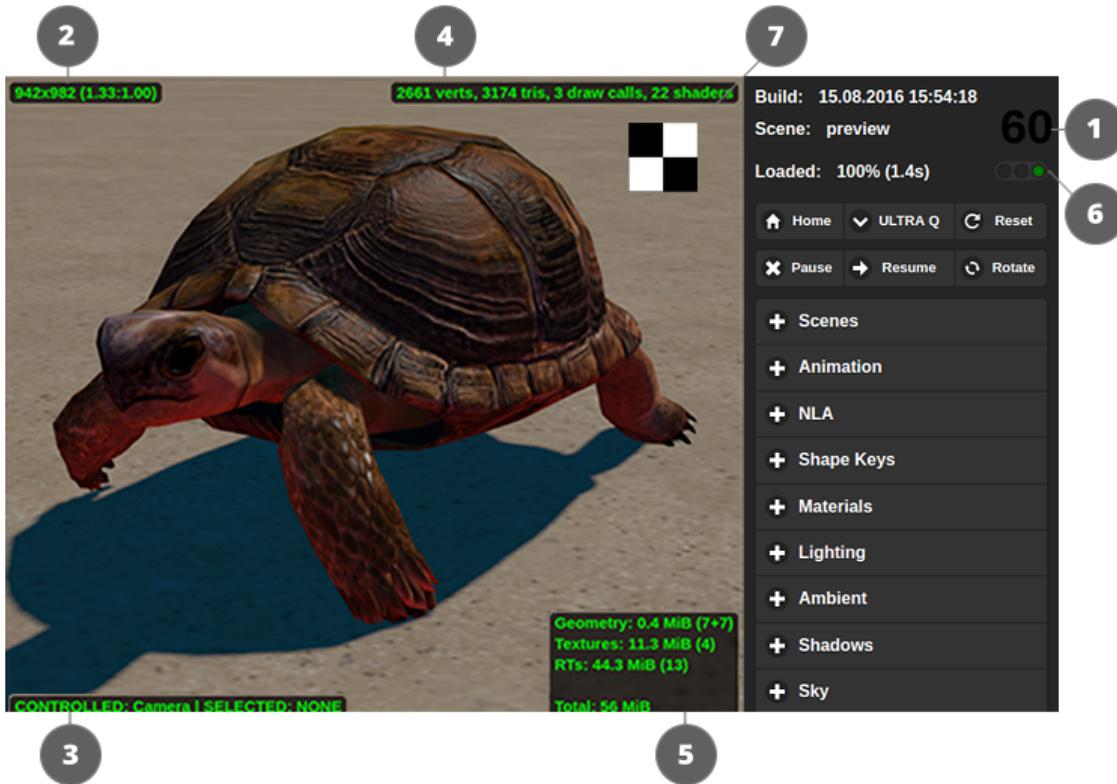
На этой панели находятся средства настройки устройств ввода.

**Gyroscope use** Включает использование гироскопов в приложении. По умолчанию отключено.

**Gamepad Settings** Вызывает окно настроек геймпадов.



## 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 вызовов отрисовки WebGL 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.

6. **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.

Ошибки загрузки и предупреждения можно увидеть в консоли браузера.

7. **Индикатор прозрачности фона** Находится в правом верхнем углу экрана, ниже индикатора сложности сцены. Этот инструмент может применяться для отладки, т.к. позволяет проверить, прозрачен ли фон сцены. Имеет вид небольшого квадрата с чёрно-белым "шахматным" узором.

Этот индикатор заметен только в том случае, если фон приложения - прозрачный. На скриншоте выше он изображён для того, чтобы дать читателю представление о том, как этот индикатор мог бы выглядеть в реальном приложении.

---

## Веб-плеер

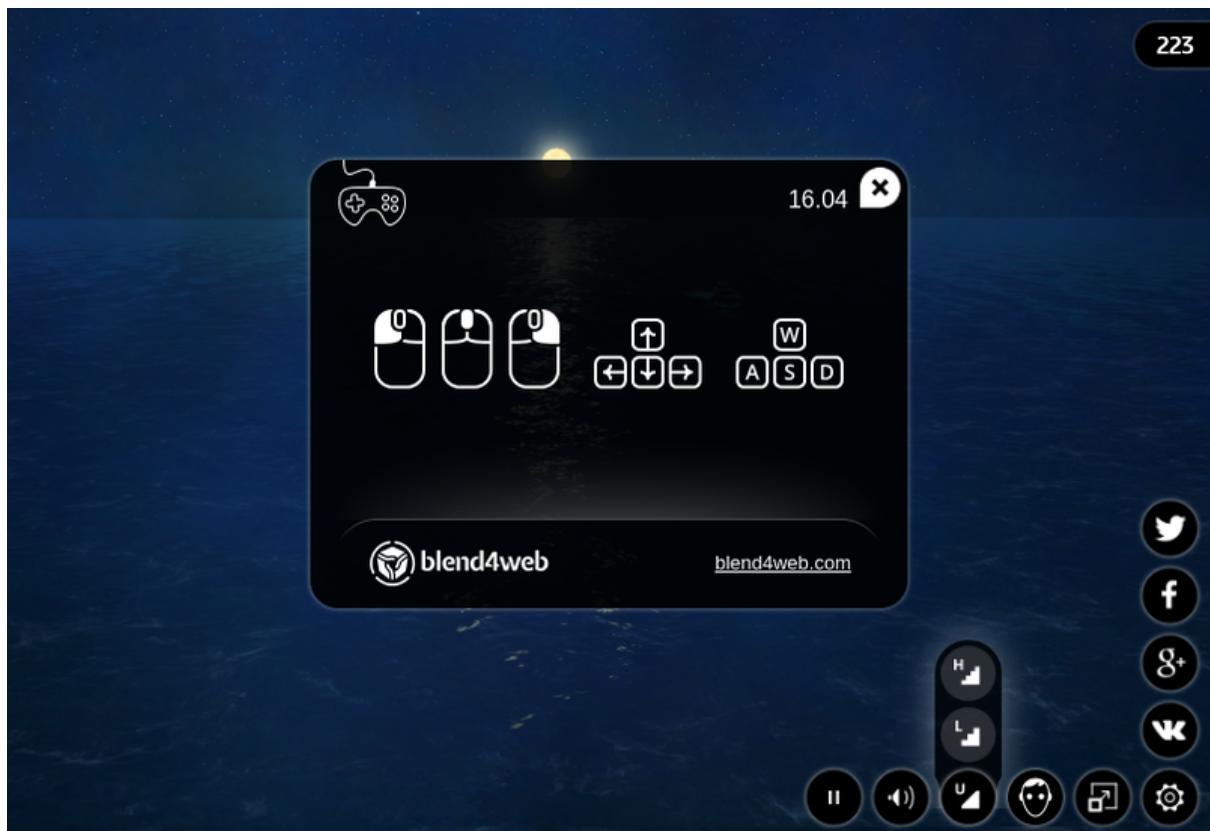
---

### Содержание

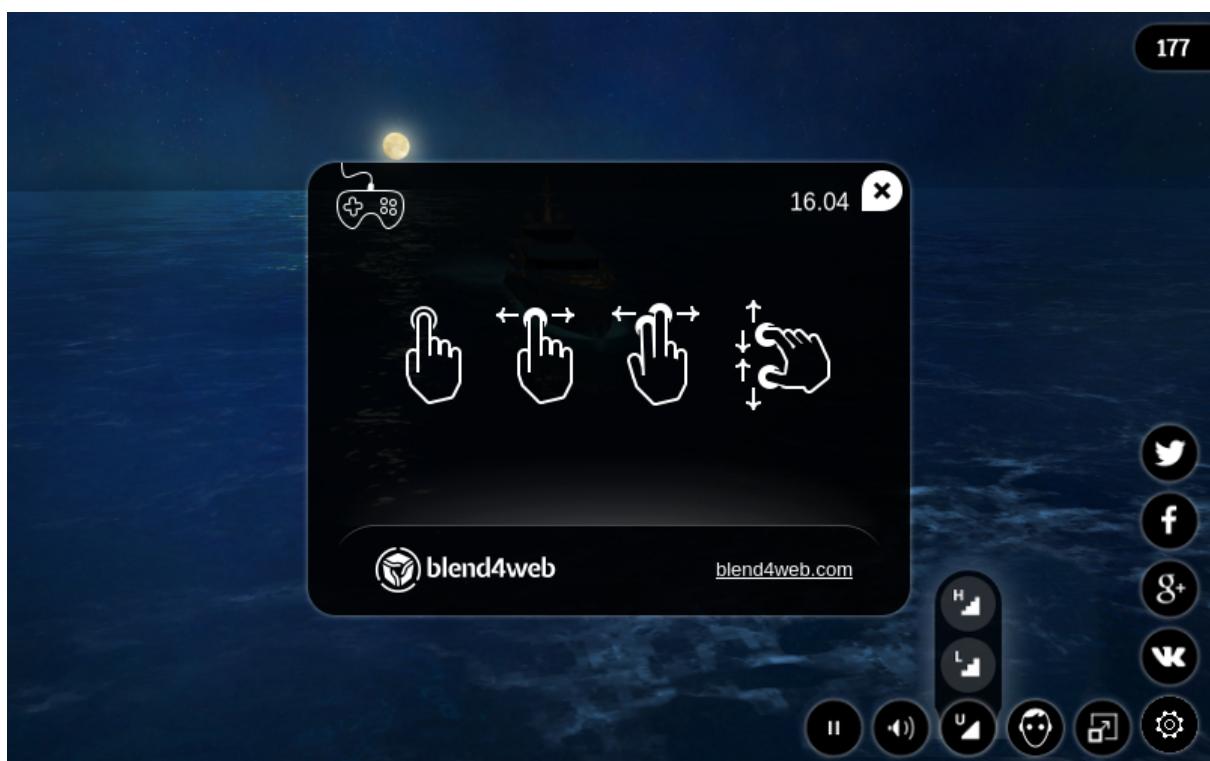
- *Веб-плеер*
  - Использование
  - Навигация
  - Панель управления
  - Атрибуты
  - Название сцены в заголовке
  - Ошибки сцены

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

Десктопная версия:



Мобильная версия:



## 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`.

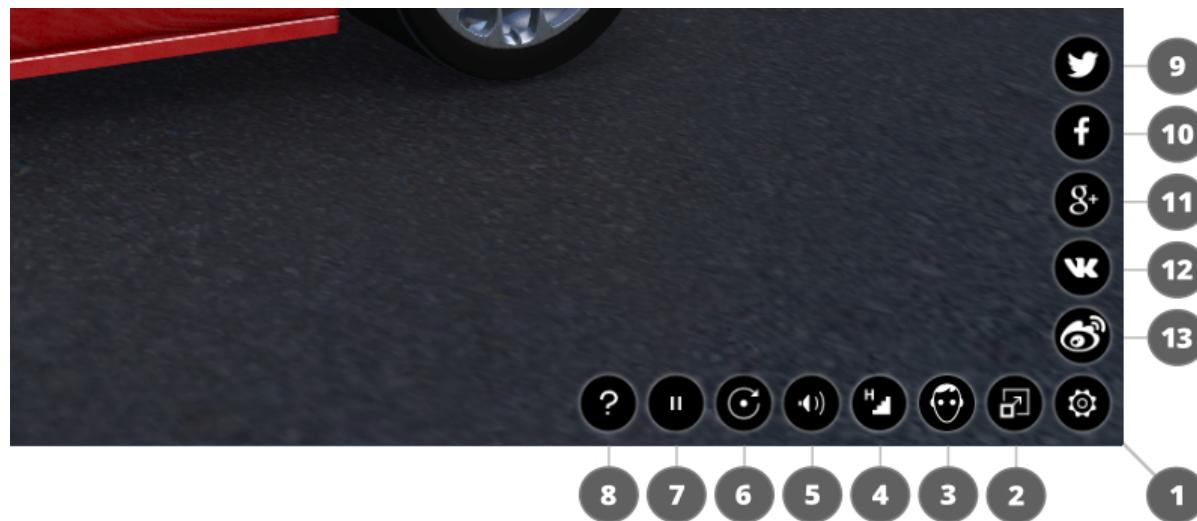
При экспорте в единый HTML файл интерфейс веб-плеера интегрируется автоматически.

## 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`.

## Панель управления

Ниже показана панель управления веб-плеера.



1. показ / скрытие панели управления;
2. включение / выключение полноэкранного режима;
3. включение / выключение стереоизображения;
4. установка качества сцены;
5. включение / выключение звука;

6. включение / выключение автоматического поворота камеры вокруг сцены;
7. запуск / остановка движка;
8. открытие окно помощи;
9. кнопка Твиттер;
10. кнопка Facebook;
11. кнопка Google+;
12. кнопка ВКонтакте;
13. кнопка Weibo.

## 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 `compressed_textures_pvr` используется для включения загрузки текстур с компрессией PVRTC. Применяется совместно с атрибутом `compressed_textures`.
8. необязательный атрибут `compressed_gzip` используется для включения загрузки сжатых при помощи GZIP ресурсов, таких как ".json.gz", ".bin.gz", ".dds.gz" и ".pvr.gz".

9. необязательный атрибут `alpha` используется для установки прозрачности области отрисовки.
10. необязательный атрибут `no_social` используется для отключения кнопок социальных сетей на панели управления.
11. необязательный атрибут `socials` используется для выборочного включения и отключения кнопок социальных сетей, а также для задания порядка, в котором они располагаются на экране. Этот атрибут должен выглядеть так: `socials=<...>`, где `<...>` - набор букв, соответствующих кнопкам социальных сетей, которые вы хотите включить (`f` - Facebook, `v` - VK, `t` - Twitter, `w` - Weibo, `g` - Google+).

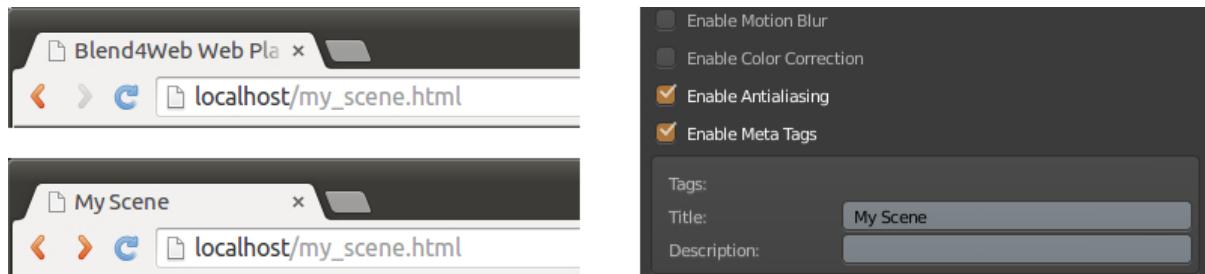
---

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

---

## Название сцены в заголовке

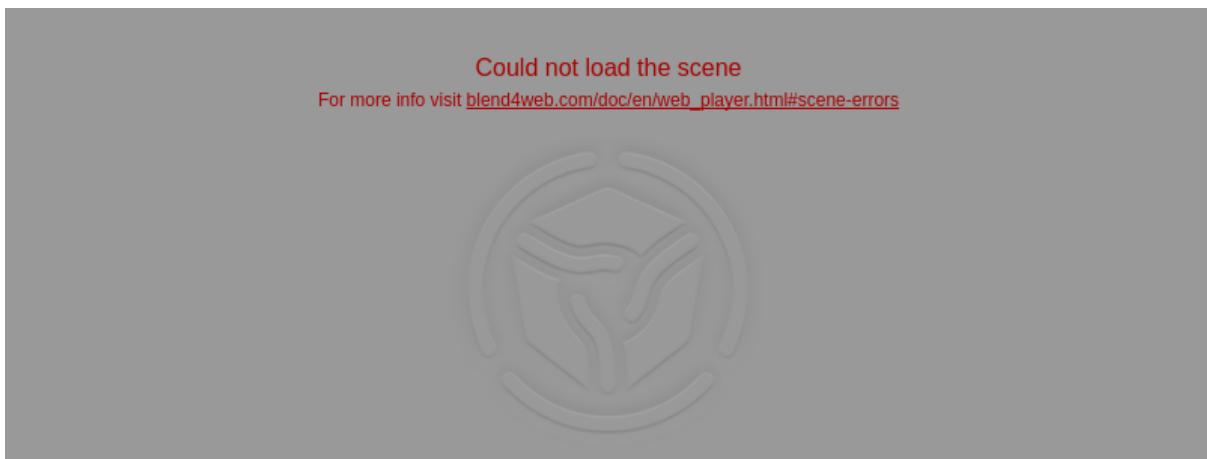
По умолчанию веб-плеер устанавливает тег `title` равным `Blend4Web Web Player`. С помощью мета-тега `title` в настройках сцены можно изменить значение по умолчанию.



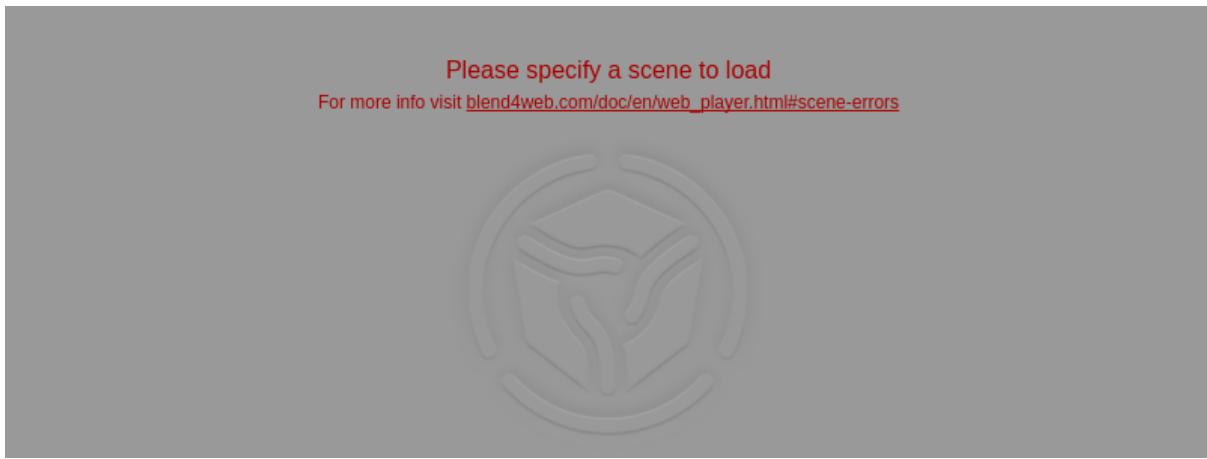
## Ошибки сцены

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

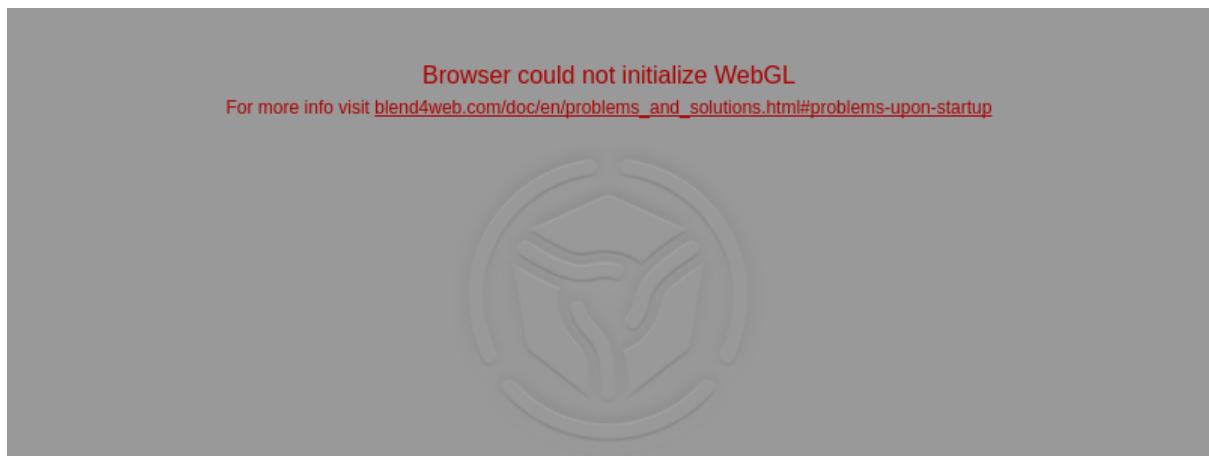
1. *атрибутом* `load` указан неверный путь к JSON файлу или данный файл неисправен;



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



3. Ошибка инициализации WebGL. Возможные решения описаны на *этой странице*.



## Аддон

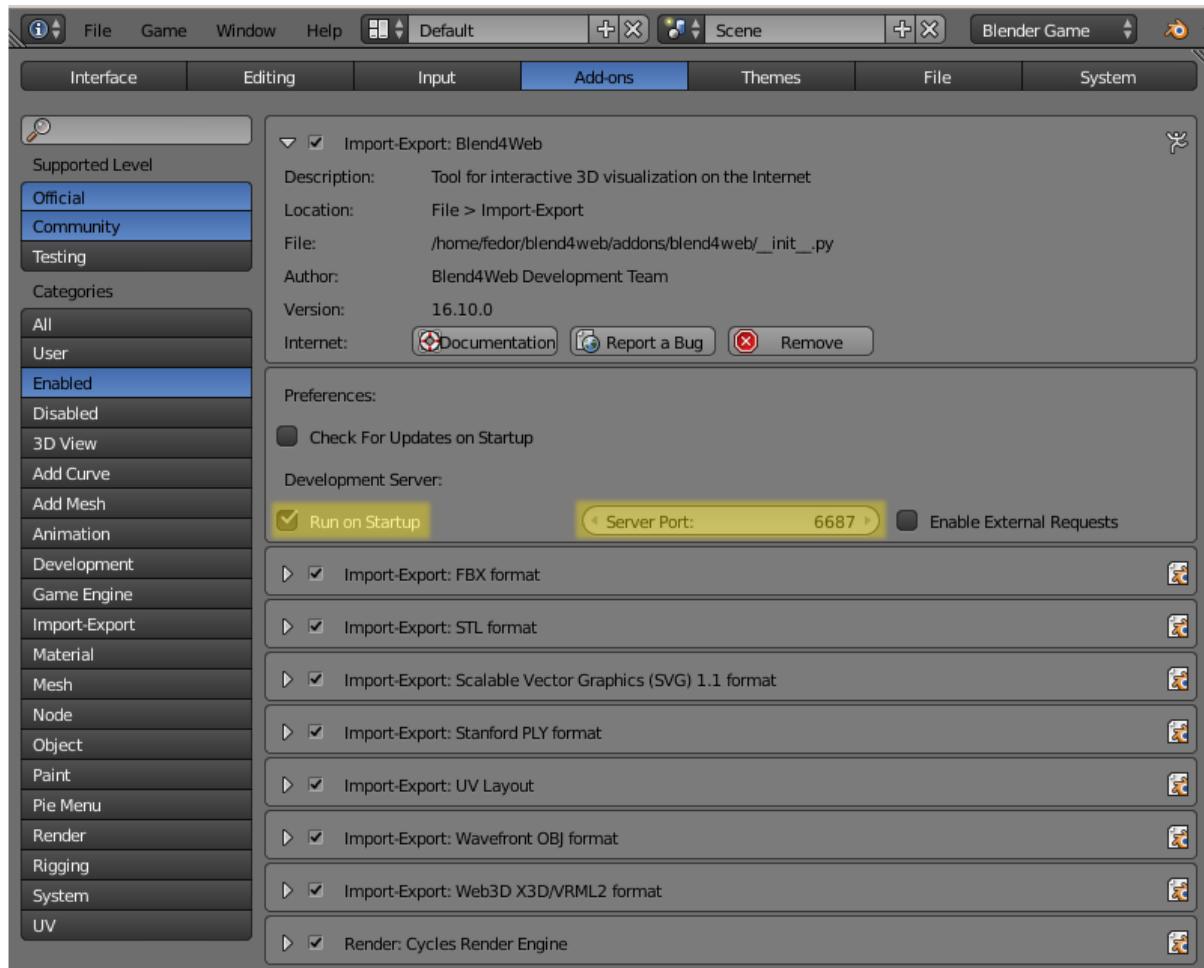
---

### Содержание

- *Аддон*
  - *Локальный сервер разработки*
  - *Запуск просмотрщика сцен и демо-приложений*
  - *Форматы экспорта*
    - \* *JSON*
    - \* *HTML*
  - *Опции экспорта*
  - *Ошибки инициализации*
  - *Ошибки совместимости*
  - *Критические ошибки экспорта*
  - *Некритические ошибки экспорта*
  - *Прочие сообщения*
  - *Перевод аддона*

## Локальный сервер разработки

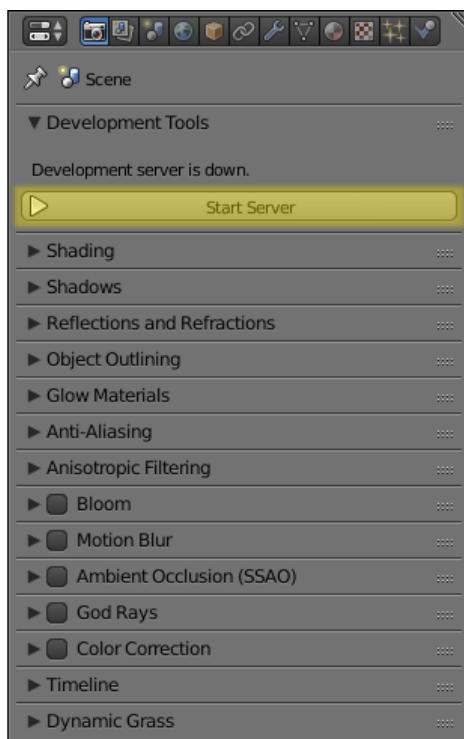
Настройки локального сервера разработки находятся в меню `File > User Preferences...` (горячие клавиши `Ctrl+Alt+U`). Здесь можно изменить номер порта, используемый при создании локального сервера разработки (6687 по умолчанию), или включить автоматический запуск сервера при запуске Blender'a: для этого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



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

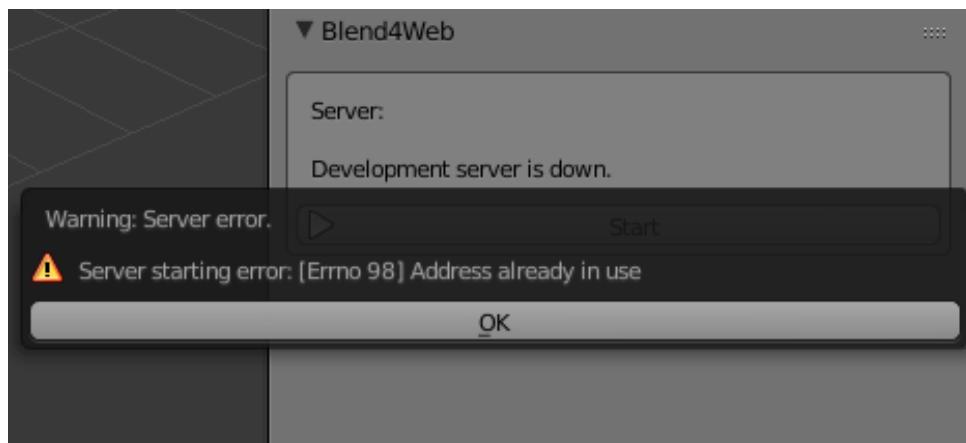
Имеется возможность сделать локальный сервер доступным из внешней сети. Для этого достаточно включить свойство **Enable External Requests**.

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



**Примечание:** Если запуск локального сервера невозможен, вместо кнопки **Start Server** 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 **SDK Index** 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.

Кнопка Project Manager открывает *менеджер проектов*.

Кнопка Fast Preview вызывает *просмотрщик сцен* и загружает в него сцену, открытую в Blender на данный момент.

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

## 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*.

### Blend4Web SDK Pro Free

WebGL: available

Development Server: available

| <b>Tools</b>             |                       |                |           |
|--------------------------|-----------------------|----------------|-----------|
| Project Manager          |                       | Scene Viewer   |           |
| Code Snippets            |                       | WebGL Report   |           |
| <b>Helpful Resources</b> |                       |                |           |
|                          | <i>English</i>        | <i>Русский</i> | <i>中文</i> |
| User Manual, HTML        | read                  | читать         |           |
| User Manual, PDF         | read                  | читать         | -         |
| API Reference            | read                  |                |           |
| Tutorials                | go                    | перейти        |           |
| FAQ                      | go                    | перейти        |           |
| Community Support Forums | go                    | перейти        |           |
| Support Email            | support@blend4web.com |                |           |

---

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

---

## Форматы экспорта

После того, как сцена закончена, её нужно преобразовать в формат, понятный движку Blend4Web.

На сегодняшний день поддерживаются два формата экспорта: JSON и HTML.

## JSON

При экспорте в этот формат создаётся файл `.json` (JavaScript Object Notation), в котором хранятся экспортруемые структуры данных и ссылки на внешние ресурсы (изображения, звуки и пр.), а также файл `.bin`, хранящий массивы данных моделей в бинарном формате.

Если медиаресурсы [запакованы](#) в [blend-файл](#), при экспорте они будут извлечены и размещены в каталоге проекта (внутри каталога SDK). При этом таким файлам будут автоматически присвоены имена (отличные от исходных), что может затруднить дальнейшую работу с ними.

Ресурсы рекомендуется размещать в отдельной директории внутри каталога SDK, например `projects/project_name/имя_проекта`.

---

**Примечание:** В соответствии с требованиями безопасности, сервер разработки имеет доступ только к каталогу SDK. Если медиаданные расположены в другом каталоге, сервер не сможет подключить их при экспорте (даже если в самом Blender они работают нормально).

---

Пути к внешним ресурсам должны быть относительными. Если это не так, следует выполнить команду `File > External Data > Make All Paths Relative`, иначе могут возникнуть проблемы с открытием файлов на других компьютерах.

Это основной формат для сложных проектов, которые могут включать несколько сцен и требуют написание программного кода на JavaScript. Работа с проектом описана в [соответствующем разделе](#).

## HTML

При экспорте в этот формат все ресурсы сцены упаковываются в один файл, имеющий расширение HTML. HTML-файл включает не только сцену, но и текстуры, звуки, сам движок Blend4Web и стандартный [веб-плеер](#). Такой файл может быть воспроизведен на любом компьютере или мобильном устройстве, имеющем браузер с поддержкой WebGL.

HTML-файлы не могут применяться для дальнейшей разработки, но и не требуют никаких дополнительных действий для запуска. Этот формат рекомендуется применять для относительно простых проектов небольшого размера.

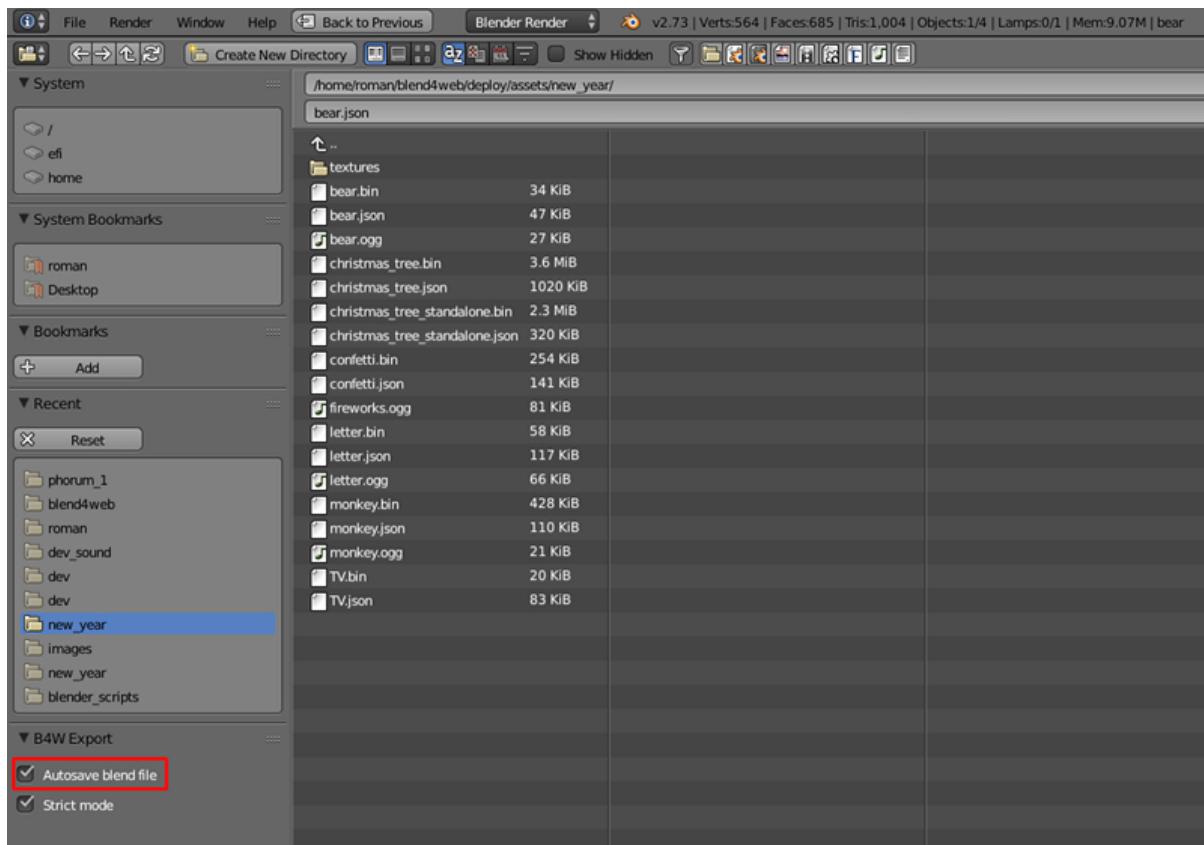
Однако следует заметить, что HTML-приложения не поддерживают следующие возможности:

- *Физику*
- *текстуры формата DDS*
- *текстуры min50*

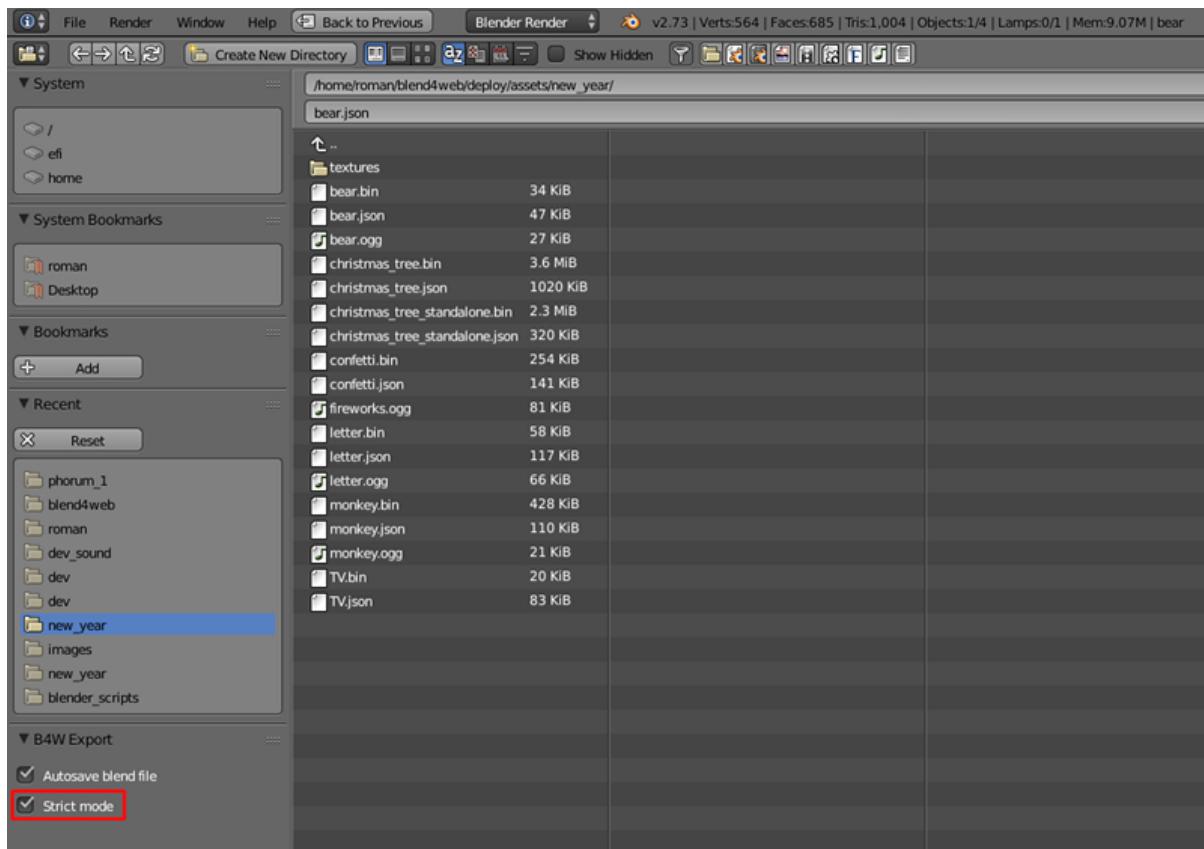
## Опции экспорта

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

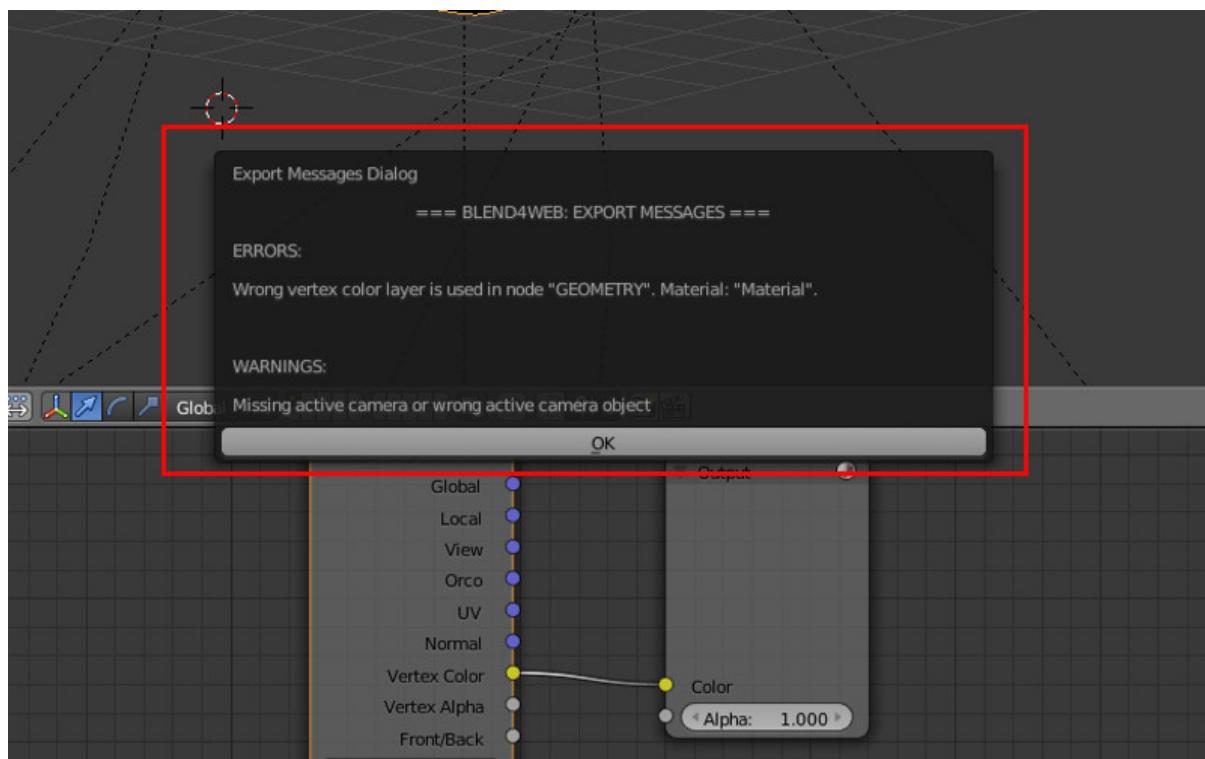
**Включено по умолчанию.** Осуществляется непосредственно после экспорта с целью поддержки соответствия между текущим содержимым blend-файла и экспортного файла. Кроме того, для удобства в blend-файле сохраняется относительный путь к экспортному файлу.



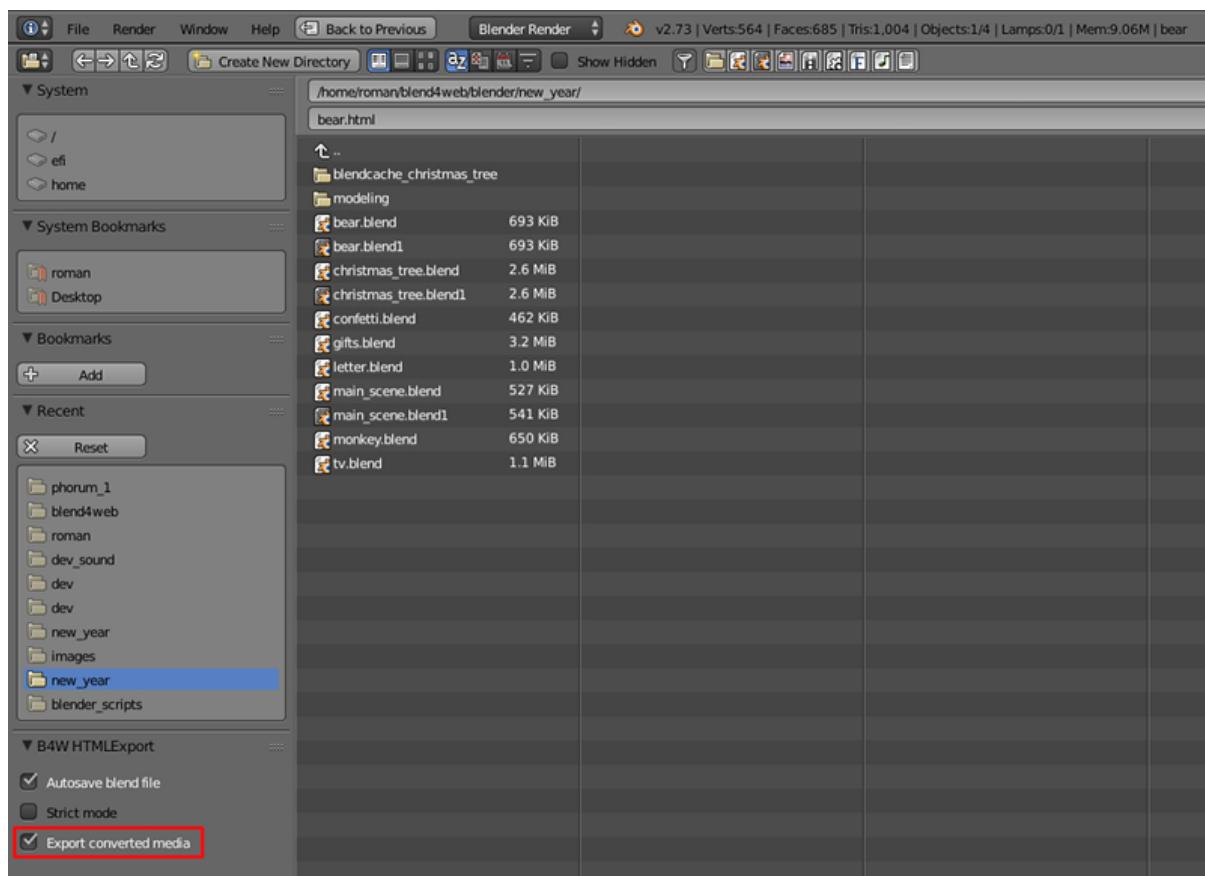
**Strict Mode** Данный режим блокирует экспорт при наличии ошибок и сообщений, требующих внимания пользователя. Режим включается при выставлении опции Strict Mode в меню экспорт:



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



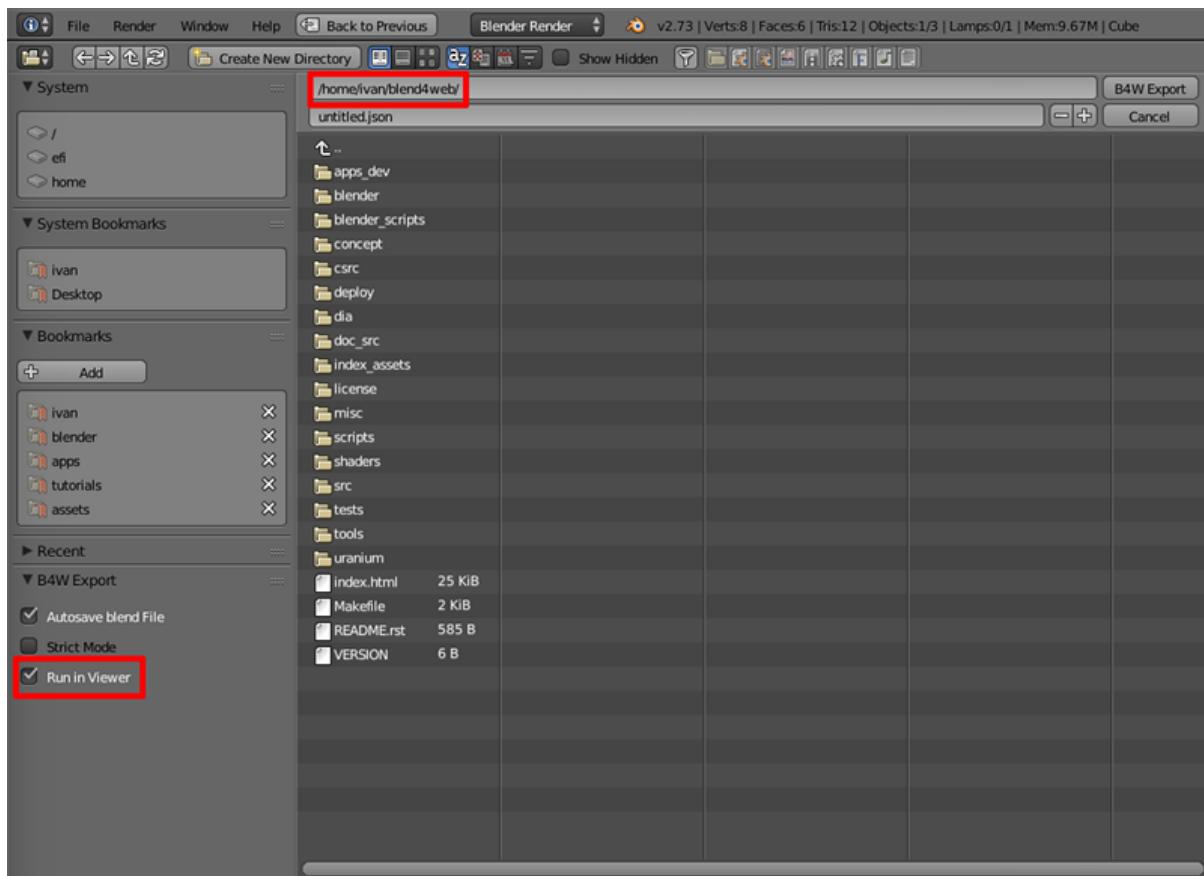
**Export Converted Media** Опция доступна при html-экспорте. Включение данного режима экспорта позволяет записать в HTML файл конвертированные медиафайлы разных форматов. Это необходимо использовать при создании кросбраузерных и кросплатформенных приложений при html-экспорте. При этом в html-файл будут 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.



## 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! Add-on 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 add-on. 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. |
| Incorrect add-on directory name.   | Некорректное имя директории аддона. Структура аддона в архиве была повреждена, либо была нарушена целостность архива.                                       |

## 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.

Версия движка сильно устарела по сравнению с версией аддона, которым был про-

изведен экспорт сцены. Сцена не будет загружена. Рекомендуется использовать последнюю версию движка и аддона.

| Сообщение об ошибке   | Причина   |
|---|---|
| JSON version is too old relative to B4W engine: VER_OLD, required: VER_NEW. Reexport scene with the latest B4W add-on 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 add-on 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.  |

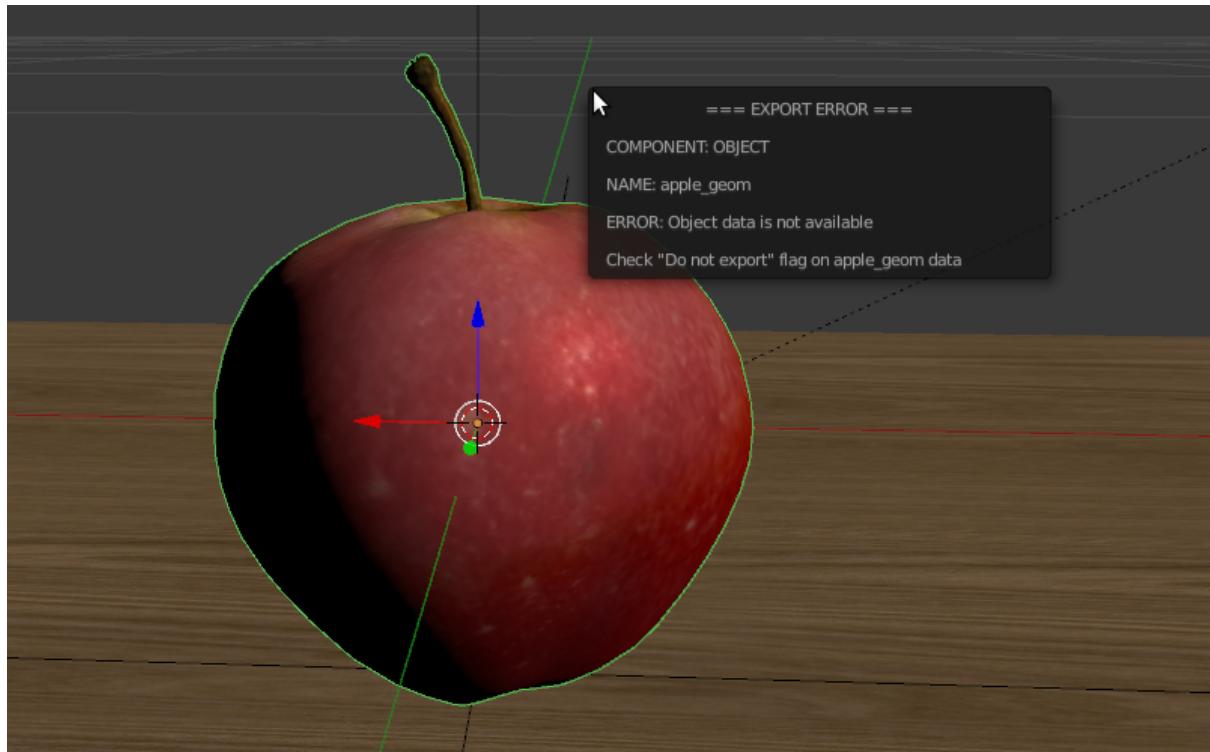
## Критические ошибки экспорта

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.

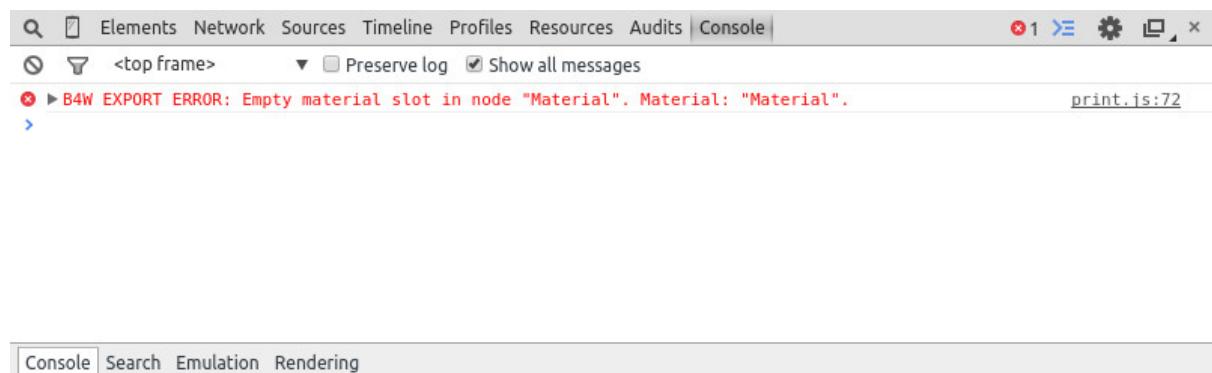


| Сообщение об ошибке  | Причина  |
|--|--|
| Export to different disk is forbidden  | Не разрешен экспорт в директорию, находящуюся на другом диске  |
| Incompatible objects with a shared mesh; The OBJECT_NAME object has both vertex groups and a shared mesh | Несовместимые объекты с общим мешем. Не допускается экспорт объекта с общим мешем и вертексными группами.<br>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 and Modifiers</code> (т.к. в этом случае при экспорте происходит полное копирование мешей). |
| Incorrect mesh; Corrupted file;<br>Wrong vertex color values   | Поврежденный файл: некорректное значение вертексного цвета.  |
| 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 <code>Normal Mapping</code> , но не имеет ноды <code>Material</code> .   |
| The mesh has a UV map but has no exported material   | Меш имеет текстурную развертку, но не имеет материала, который бы экспортировался.   |
| The mesh has a vertex color layer but has no exported material   | Меш имеет слой вертексного цвета, но не имеет материала, который бы экспортировался.   |
| No such file or directory  | Данная директория не существует.   |
| 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.<br>Требуется сделать <code>Clean Up</code> либо <code>Save</code> в панели <code>B4W Vertex Normals Editor</code> .  |
| Wrong overridden bounding box;<br>Check the mesh's bounding box values                                   | Указаны неверные размеры при переопределении <code>BoundingBox</code> для меша: минимальное значение больше максимального для хотя бы одного из измерений.   |

## 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.<br>Texture NAME.  | Данный идентификатор для объекта типа <b>Canvas</b> уже существует.   |
| Dupl group error for object<br>OBJECT_NAME. Objects from the<br>GROUP_NAME dupl group on the<br>OBJECT_NAME object cannot be<br>exported | Ни один из объектов группы<br>GROUP_NAME, выбранной для<br>дублирования на объекте<br>OBJECT_NAME, не экспортируется.<br>Требуется разрешить экспорт хотя бы<br>одного из объектов группы, либо<br>убрать дублирование группой. |
| Empty canvas texture ID for texture<br>NAME.   | Пустое поле идентификатора для<br>объекта типа <b>Canvas</b> .  |
| Empty material slot in node “NAME”.<br>Material: “NAME”.   | Не задан материал в ноде: “NAME”  |
| Environment map in the “NAME” world<br>texture slot cannot be a movie.   | Карта окружения не может быть<br>видео.   |
| Ignoring LODs after empty LOD for the<br>NAME object.  | Все LOD-объекты, следующие за<br>пустым слотом, были<br>проигнорированы.  |
| Incomplete mesh NAME; Dynamic grass<br>vertex colors required by material<br>settings  | Неполный меш: специальный материал<br>для ландшафта использует опции<br><b>Dynamic grass size</b> и/или <b>Dynamic<br/>grass color</b> , но у меша нет слоев<br>вертексного цвета с такими именами.                             |
| Incomplete mesh; Material settings<br>require vertex colors  | Неполный меш: материал меша имеет<br>включенную опцию вертексного цвета<br>( <b>Vertex Color Paint</b> ), но у меша нет<br>слоя вертексного цвета.  |
| Incorrect NLA script, falling back to<br>simple sequential NLA.  | Некорректный NLA-скрипт. Вместо<br>него будет использоваться стандартная<br>NLA-анимация.   |
| Incorrect vertex animation for mesh<br>NAME. Object has no vertex animation.   | Включен экспорт вертексной анимации<br>для объекта, но ни одной анимации не<br>имеется.   |
| Incorrect vertex animation for mesh<br>NAME. Unbaked “ANIM_NAME”<br>vertex animation.  | Включен экспорт вертексной анимации<br>для меша, но анимация ANIM_NAME<br>не содержит ни одного кадра.  |
| Incorrect mesh NAME; Wrong group<br>indices  | Меш содержит вершины, привязанные<br>к несуществующей группе.   |
| Incorrect mesh NAME; Wrong vertex<br>positions   | Поврежденный файл: некорректное<br>значение координаты вертекса.  |
| Incorrect mesh NAME; Wrong normals   | Поврежденный файл: некорректное<br>значение нормали.  |
| Incorrect mesh NAME; Wrong tangents  | Поврежденный файл: некорректное<br>значение тангенты.   |
| Incorrect mesh NAME; Wrong texture<br>coordinates  | Поврежденный файл: некорректное<br>значение текстурной координаты.  |
| Incorrect mesh NAME; Wrong vertex<br>group weights   | Поврежденный файл: некорректное<br>значение веса вертекса в вертексной<br>группе.   |

|   |   |
|---|---|
| Incomplete vehicle. The NAME vehicle doesn't have any chassis or hull                         | Моделируемое средство передвижения NAME является незавершенным: оно должно содержать один элемент <b>Chassis</b> или <b>Hull</b> .            |
| Incomplete vehicle. The NAME vehicle requires at least one bob                                | Моделируемое средство передвижения NAME является незавершенным: оно должно содержать хотя бы один элемент <b>Bob</b> .                        |
| Incomplete vehicle. The NAME vehicle requires at least one wheel                              | Моделируемое средство передвижения NAME является незавершенным: оно должно содержать хотя бы один элемент <b>wheel</b> .                      |
| 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 мира NAME 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 constraint has no target. Check "CONST HOLDER NAME".                                   | Для ограничителя объекта (вкладка <b>Object Constraints</b> ) не установлено свойство <b>Target Object</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".  |
| Object "NAME" has no data or data is broken. Change object type to EMPTY.                     | Объект с именем "NAME" не имеет меша, либо меш повреждён. Тип объекта изменён на <b>EMPTY</b> .   |
| 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 “ <b>to</b> ” 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.  |
| Particle system error. Unsupported render type “TYPE” for the HAIR particles “NAME” on object “OBJECT_NAME”. Particle system removed.                            | Ошибка системы частиц.<br>Неподдерживаемый тип рендера TYPE для EMITTER/HAIR системы частиц с именем PSYS_NAME, находящейся на объекте NAME. Система частиц удалена.  |
| Particle system error. Unsupported render type “TYPE” for the EMITTER particles “NAME” on object “OBJECT_NAME”. Particle system removed.                         | Ошибка системы частиц.<br>Неподдерживаемый тип рендера TYPE для EMITTER/HAIR системы частиц с именем PSYS_NAME, находящейся на объекте NAME. Система частиц удалена.  |
| Particle system error for “NAME”. Dupli object isn’t specified.  | Ошибка системы частиц. Не выбран объект, используемый в качестве частицы.   |
| Particle system error for “NAME”. Dupli group isn’t specified.   | Ошибка системы частиц: не указана группа, использующаяся для генерации частиц.  |
| Particle system error for “NAME”. Wrong dupli object type “TYPE”.  | Ошибка системы частиц. В качестве частицы выбран объект неподходящего типа. Поддерживаемые типы: MESH.  |
| Particle system error for “NAME”. Dupli object “NAME” has not been exported.   | Объект NAME, выбранный в качестве частицы, не экспортируется (на нем выбрана опция <b>Do not export</b> ).  |
| Particle system error for “NAME”. 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 <b>Do not export</b> , либо объекты имеют неподходящий тип. Поддерживаемые типы: MESH. |
| <b>8.8. Некритические ошибки экспорта</b><br>Sound file is missing in the SPEAKER object “NAME”. Converted to EMPTY.   | У объекта типа SPEAKER отсутствует звуковой файл. Тип объекта изменён на EMPTY.   |

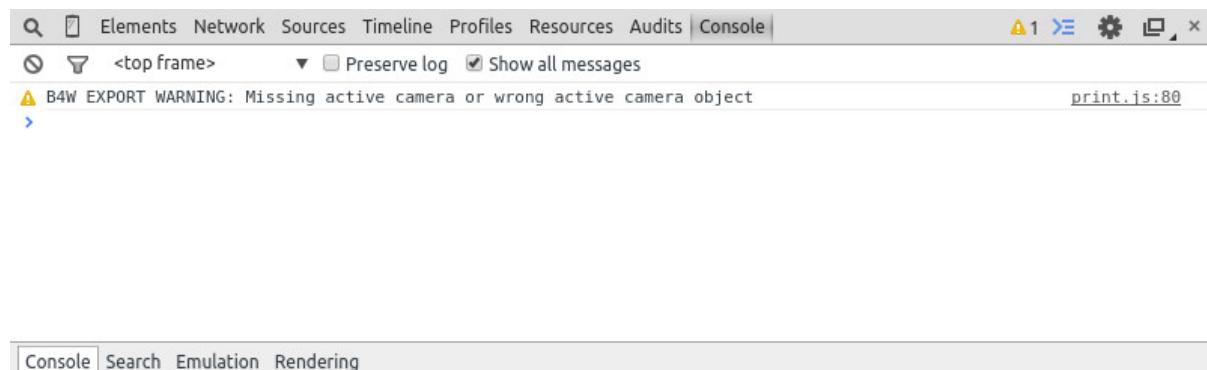
|   |   |
|---|---|
| The main scene NAME can not be rendered by another scene.<br>Material NAME has been removed.          | Основная сцена NAME не может отображаться другой сценой. Материал NAME был удален.  |
| The NAME action has decimal frames. Converted to integer.   | Анимация NAME содержит дробные значения кадров, что не поддерживается. Округлено до целых.  |
| The NAME armature modifier has a proxy object as an armature.<br>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.<br>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 вертексную анимацию и арматурный модификатор. Модификатор удален.   |
| The NAME object has the NAME armature modifier. It belongs to a not exported scene. Modifier removed. | В модификаторе NAME типа <b>Armature</b> не указан объект, либо объект не экспортируется. 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.<br>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 Cycles.                       |
| 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” texture type is not supported for world “NAME”.  | TEXTURE_TYPE тип текстуры не поддерживается для мира, WORLD_NAME “NAME”.  |
| Using B4W_REFRACTION node NODE_NAME with incorrect type of Alpha Blend. Material: NAME.               | Используется нодовый материал с неправильно заданным свойством Alpha Blend. 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 непустой текстурой.   |
| Wrong texture coordinates type in texture NAME. [Material: NAME.]  | Для текстур с изображением ( <b>image</b> ) поддерживаются следующие типы координат: <b>UV</b> , <b>Normal</b> и <b>Generated</b> .   |
| Wrong F-Curve interpolation mode for ACTION_NAME. Only BEZIER, LINEAR or CONSTANT mode is allowed for F-Curve interpolation. Switch to BEZIER. | Для режима интерполяции действий поддерживаются следующие типы: <b>BEZIER</b> , <b>LINEAR</b> и <b>CONSTANT</b> .   |
| Wrong vertex animation vertices count for mesh NAME. It doesn't match with the mesh vertices count for "ANIM_NAME".                            | Включен экспорт вертексной анимации, но число вершин покадрово в анимации <b>ANIM_NAME</b> не совпадает с числом вершин меша. Возможное решение - "перезапекание" анимации.   |

## 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 с именем “NAME” нет ни одного канала.   |
| The “NAME” camera has unsupported PANORAMIC type. Changed to PERSPECTIVE type.”                             | Панорамная камера не поддерживается. Будет использована перспективная камера.  |
| Unsupported texture type or texture is missing for Lens Flare material “NAME”                               | Текстура для материала Lens Flare с именем “NAME” отсутствует или имеет неподдерживаемый формат.   |
| Use of ENVIRONMENT_MAP as diffuse color is not supported. Use as mirror instead.                            | Текстура типа ENVIRONMENT MAP не может быть использована как источник диффузного цвета. Чтобы использовать текстуру в качестве карты зеркального отражения (mirror map), отключите настройку Diffuse > Color и включите Shading > Mirror на панели Textures > Influence. |

## 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    |
| Продолжается на следующей странице |          |

Table 8.1 – продолжение с предыдущей страницы

| Имя файла         | Язык                 |
|-------------------|----------------------|
| 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.

## Параметры сцены

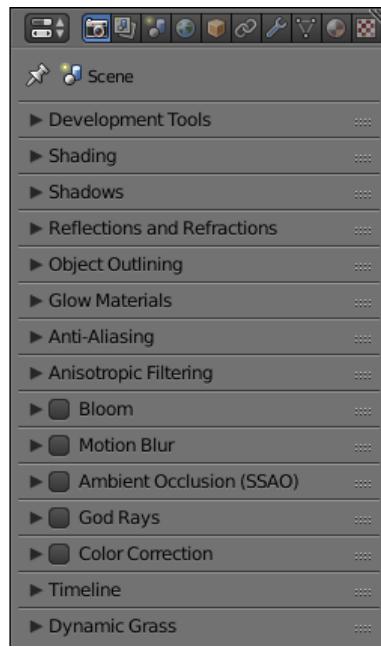
---

## Содержание

- *Параметры сцены*
  - *Вкладка Render*
    - \* *Development Tools*
    - \* *Shading*
    - \* *Shadows*
    - \* *Reflections and Refractions*
    - \* *Object Outlining*
    - \* *Glow Materials*
    - \* *Anti-Aliasing*
    - \* *Anisotropic Filtering*
    - \* *Bloom*
    - \* *Motion Blur*
    - \* *Ambient Occlusion (SSAO)*
    - \* *God Rays*
    - \* *Color Correction*
    - \* *Timeline*
    - \* *Dynamic Grass*
  - *Вкладка Scene*
    - \* *Scene*
    - \* *Units*
    - \* *Audio*
    - \* *Logic Editor*
    - \* *NLA*
    - \* *Meta Tags*
    - \* *Physics*
    - \* *Object Clustering & LOD*
    - \* *Object Selection*
    - \* *Anchors*
    - \* *Export Options*
  - *Вкладка World*
    - \* *Preview*
    - \* *World*
    - \* *Environment Lighting*
    - \* *Mist*
    - \* *Procedural Sky*
    - \* *Animation*
    - \* *Export Options*

Все параметры, определяющие вид и поведение сцены (а не отдельных объектов), находятся на трёх вкладках: **Render**, **Scene** и **World**.

## Вкладка Render



На этой вкладке находятся параметры сцены, связанные с рендерингом изображения.

### Development Tools

Настройки сервера разработки. Подробно описаны в *соответствующем разделе*.



## Shading

На этой панели находятся настройки затенения.



*Set Recommended Options* Применяется для достижения максимального соответствия

ствия между тем, как сцена выглядит в окне просмотра Blender и тем, как она выглядит в движке. При нажатии на кнопку:

- включается **World Space Shading**,
- режим материала переключается на **GLSL**,
- метод затенения объектов в окне просмотра переключается на **Material**,
- подгонка сенсора главной камеры переключается на **Vertical**.

**World Space Shading** Включает и выключает затенение в мировом пространстве (т.е. для расчёта освещённости объектов используется информация об освещении в мировом пространстве координат). Отключено по умолчанию.

## Shadows

Настройки теней. Подробно описаны в *соответствующем разделе*.



## Reflections and Refractions

Настройка эффектов отражения и преломления.



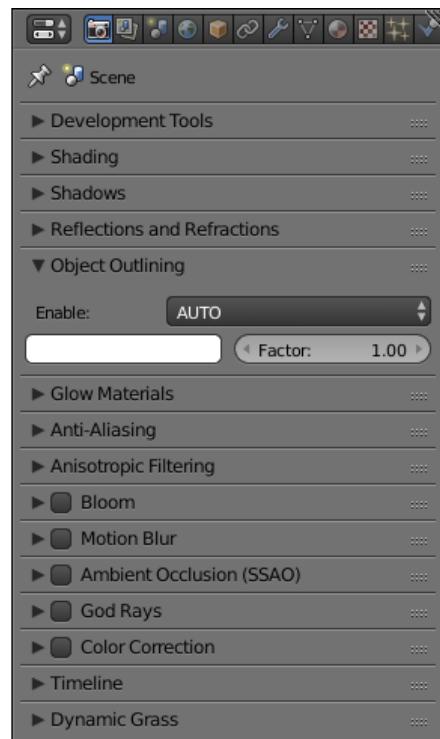
**Reflection** Настройки эффекта отражения. Может иметь значения **ON** и **OFF**. Значение по умолчанию - **ON**.

**Refraction** Настройки эффекта преломления. Может иметь значения **ON**, **OFF** и **AUTO**. Значение по умолчанию - **AUTO**.

**Quality** Настройки качества эффекта отражения и преломления. Может иметь значения **HIGH** (высокое качество), **MEDIUM** (среднее) и **LOW** (низкое). Значение по умолчанию - **MEDIUM**.

## Object Outlining

Настройки эффекта обводки. Подробно описаны в *соответствующем разделе*.



## Glow Materials

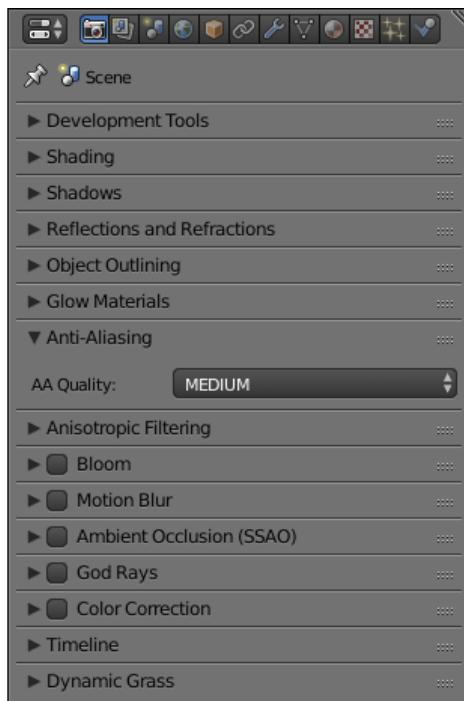
Настройки эффекта свечения материалов. Подробно описаны в *соответствующем разделе*.



**Примечание:** При динамической загрузке опция AUTO интерпретируется как OFF, если в главной сцене не было материалов с эффектом Glow. В этом случае, чтобы эффект Glow работал на динамически подгружаемых объектах, нужно выставить данный параметр в значение ON.

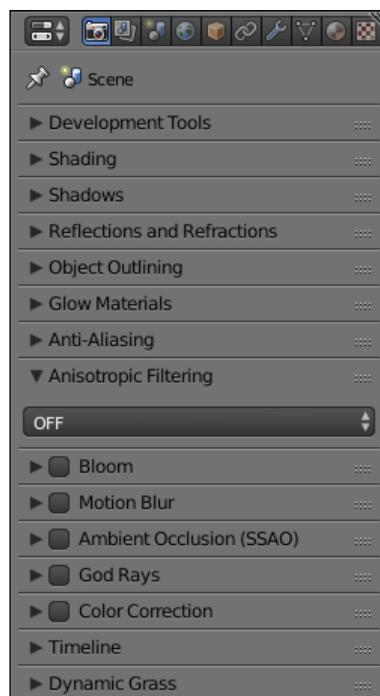
## Anti-Aliasing

Настройки сглаживания изображения (anti-aliasing). Подробно описаны в [соответствующем разделе](#).



## Anisotropic Filtering

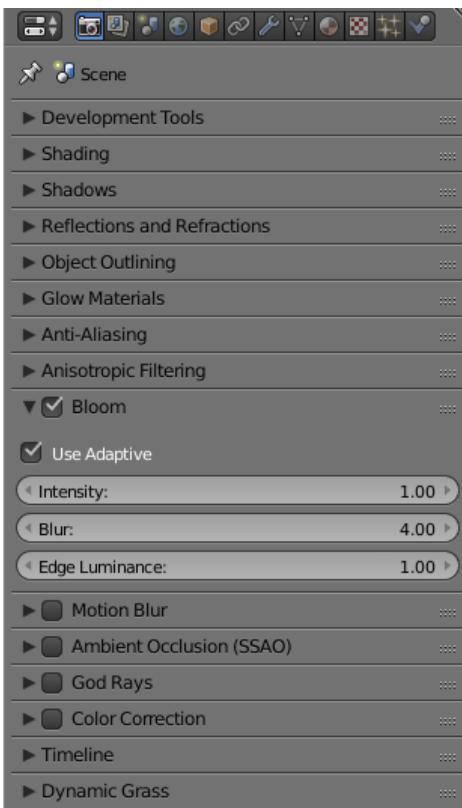
Используется для включения и выключения анизотропной фильтрации, а также для задания числа сэмплов, используемых для фильтрации. Анизотропная фильтрация отключена по умолчанию.



Доступные значения: 16x, 8x, 4x, 2x и OFF (значение по умолчанию).

## Bloom

Настройки эффекта засветки. Подробно описаны в *соответствующем разделе*.



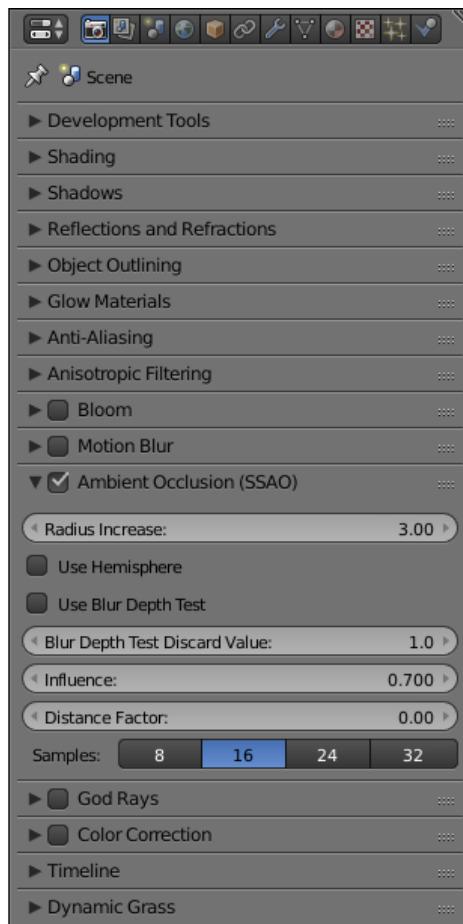
## Motion Blur

Настройки эффекта размытия изображения при быстром движении. Подробно описаны в *соответствующем разделе*.



## Ambient Occlusion (SSAO)

Настройки эффекта взаимного затенения (SSAO). Подробно описаны в *соответствующем разделе*.



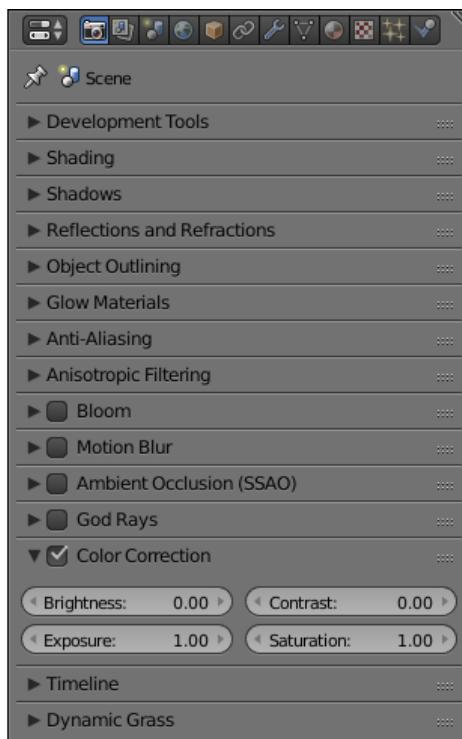
## God Rays

Настройки эффекта “сумеречных лучей”. Подробно описаны в *соответствующем разделе*.



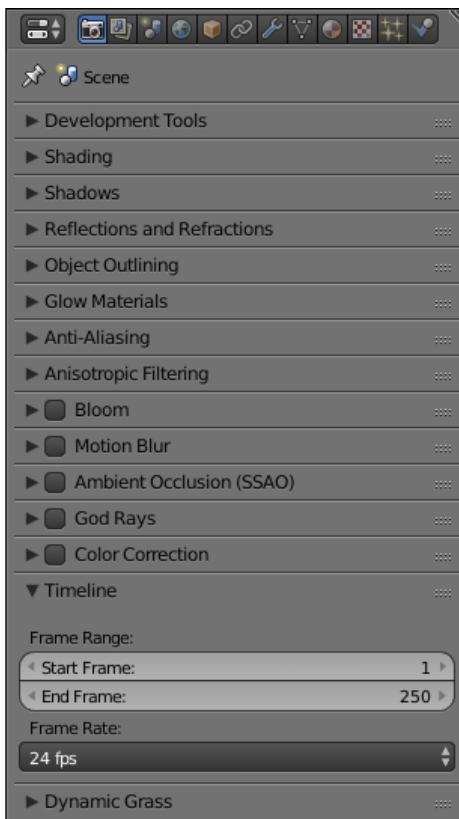
## Color Correction

Настройки цветокоррекции. Подробно описаны в *соответствующем разделе*.



## Timeline

Настройки таймлайна.



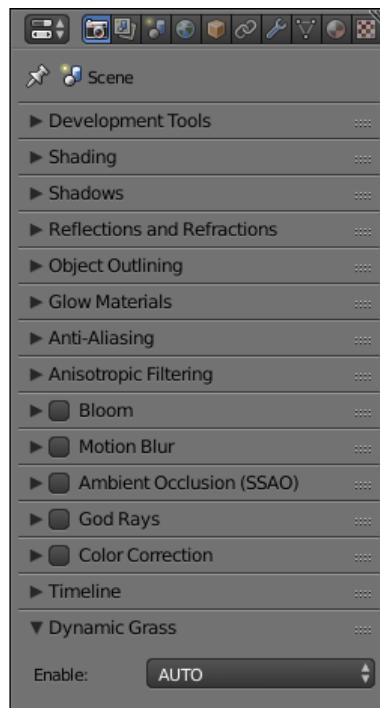
**Start Frame** Первый кадр таймлайна. По умолчанию 1.

**End Frame** Последний кадр таймлайна. По умолчанию 250.

**Frame Rate** Количество кадров в секунду. По умолчанию 24. Этот параметр определяет скорость воспроизведения анимации (но не скорость работы самой сцены).

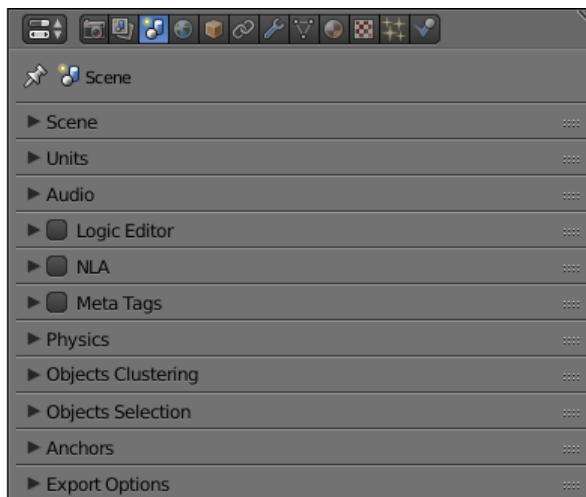
## Dynamic Grass

Включает и выключает эффект *травяного покрова*.



Доступные значения: ON, OFF и AUTO. Значение по умолчанию - AUTO.

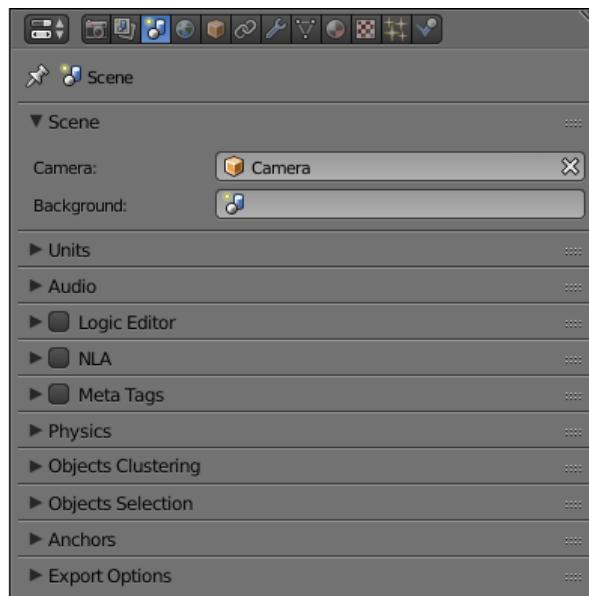
## Вкладка Scene



На этой вкладке находятся настройки, связанные с поведением сцены, в том числе настройки звука, физики, анимации и пр.

### Scene

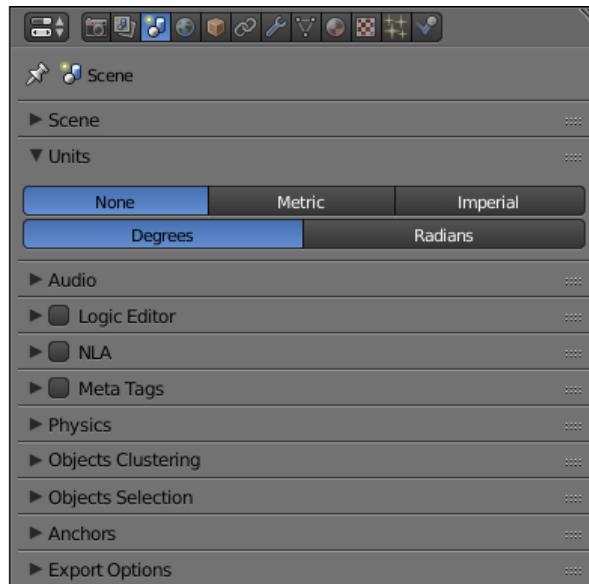
Настройки сцены.



**Camera** Камера, используемая для воспроизведения сцены.

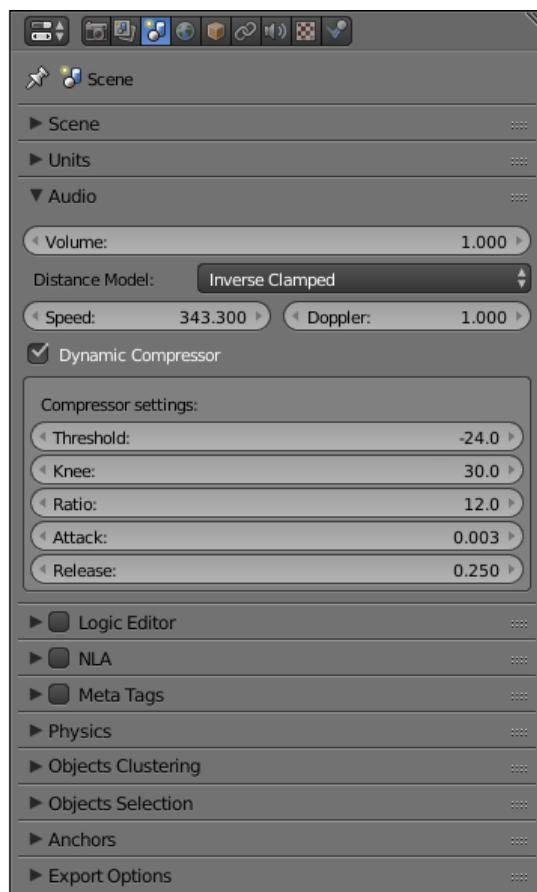
## Units

Настройки единиц измерения, используемых в сцене.



## Audio

Настройки параметров звука.



**Volume** Общая громкость звука в приложении. Значение может изменяться от 0 до 100, значение по умолчанию - 1.0.

**Distance Model** Модель расстояния, используемая для расчёта затухания. Движок поддерживает следующие модели:

**None** - затухание не используется, громкость звука не зависит от расстояния.

**Exponent Clamped** - экспоненциальная модель расстояния с отсечением.

**Linear Clamped** - линейная модель расстояния с отсечением.

**Inverse Clamped** - инверсивная модель расстояния с отсечением.

Следующие модели поддерживаются движком частично (работают так же, как соответствующие модели с отсечением):

**Exponent**

**Linear**

**Inverse**

Значение по умолчанию - **Inverse Clamped**.

**Speed** Задаёт скорость звука, которая используется для вычисления эффекта Допплера. Это значение измеряется в метрах в секунду и по умолчанию равно 343.3.

**Doppler** Этот параметр задаёт значение высоты тона, используемое для расчёта эффекта Допплера. По умолчанию значение равно 1.0.

**Dynamic Compressor** Сжатие динамического диапазона звукового сигнала. Применяется для того, чтобы сделать звучание более насыщенным и более равномерным по громкости. По умолчанию отключено.

**Threshold** Значение входного сигнала, при превышении которого включается компрессор. По умолчанию равно -24.

**Knee** Интервал над порогом (**Threshold**), где кривая громкости переходит в режим затухания. По умолчанию равен 30.

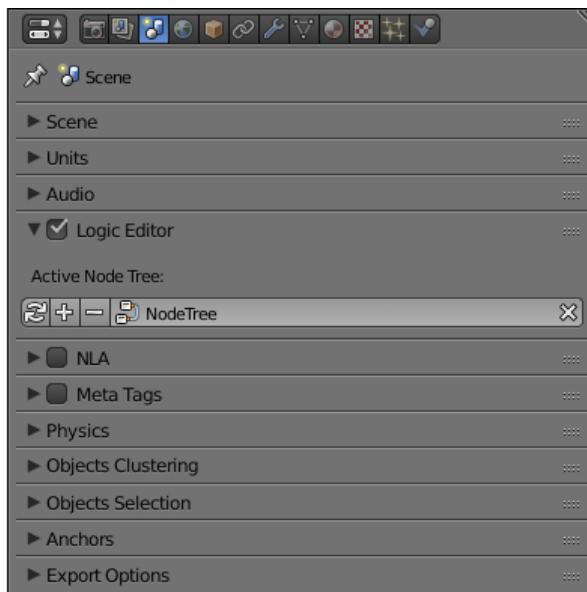
**Ratio** Фактор затухания компрессора. Значение по умолчанию равно 12.

**Attack** Время (в секундах) для затухания громкости на 10 дБ. По умолчанию равно 0.003.

**Release** Время (в секундах) для повышения громкости на 10 дБ. По умолчанию равно 0.25.

## Logic Editor

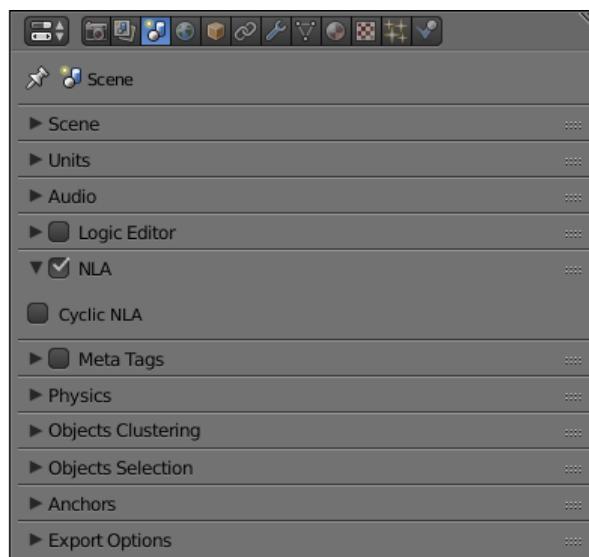
Настройки использования логических деревьев (создаваемых в *редакторе логики*) при воспроизведении сцены. По умолчанию отключено.



**Active Node Tree** Дерево логики, используемое при воспроизведении сцены.

## NLA

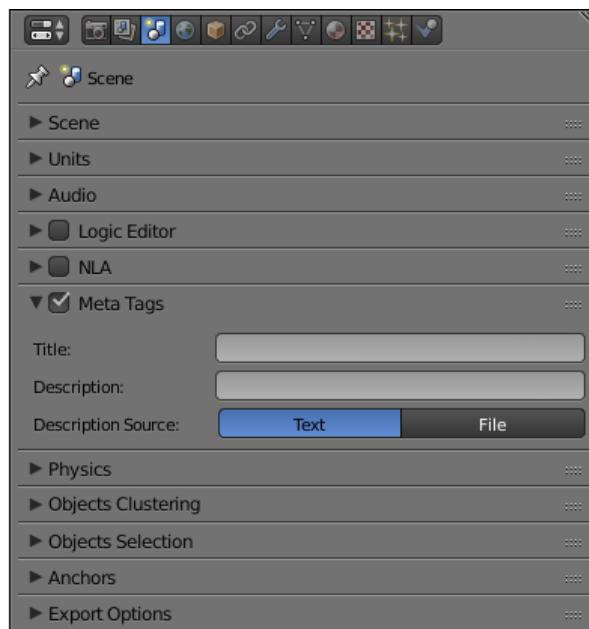
Настройки использования нелинейной анимации при воспроизведении сцены. По умолчанию отключено.



**Cyclic NLA** При активации этого параметра нелинейная анимация будет зацикленна.

## Meta Tags

Мета-теги приложения.



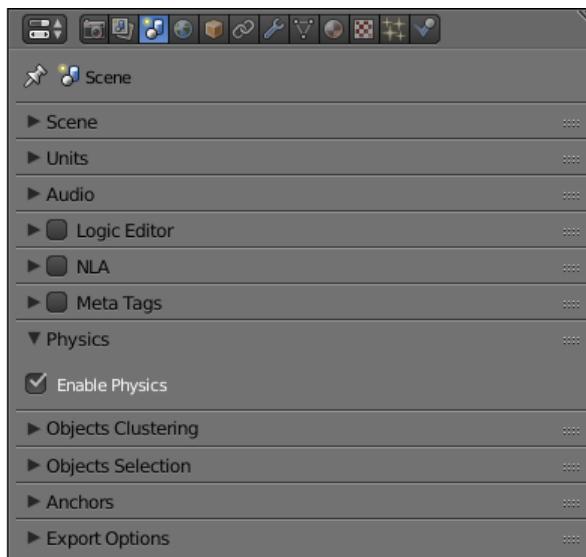
**Title** Заголовок приложения.

**Description** Описание приложения. Может быть текстовым или представлять собой ссылку на файл (если параметр **Description Source** имеет значение **File**).

**Description Source** Источник описания приложения. Описание может быть взято из файла или задано непосредственно в поле **Description**. Доступные значения - **Text** и **File**. Значение по умолчанию - **Text**.

## Physics

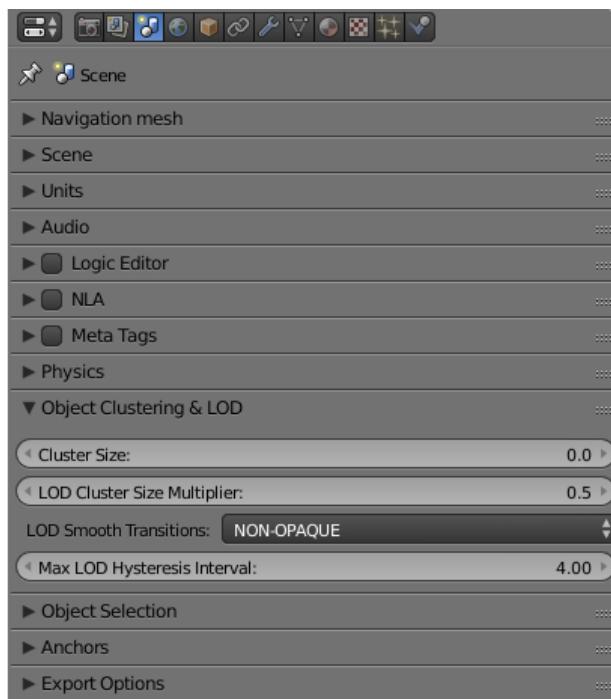
Настройки использования физики.



**Enable Physics** Использовать физику в приложении. Включено по умолчанию.

## Object Clustering & LOD

Settings for object clustering and Levels Of Detail.



**Cluster Size** The size of the cluster used for batching (in meters). Can be used for optimization purposes. If this parameter is set to zero, the engine will try to combine all objects in the scene. Set to zero by default.

**LOD Cluster Size Multiplier** Этот параметр используется для разбиения кластеров на части (в зависимости от расстояния LOD, которое настраивается индивидуально для каждого объекта) для того, чтобы переходы между различными уровнями детализации были менее заметны. Размеры меньших кластеров определяются на основании расстояний LOD, умноженных на значение этого параметра. Высокие значения увеличивают размер кластеров, что повышает производительность, но делает более заметными переходы между уровнями детализации. Низкие значения делают переходы менее заметными, но снижают производительность. Значение по умолчанию - 0.5.

**LOD Smooth Transitions** Определяет, какие объекты будут использовать плавные переходы при переключении уровней LOD. Может принимать следующие значения:

- OFF - плавные переходы отключены (наиболее быстрый вариант).
- NON-OPAQUE - плавные переходы используются для объектов с *материалами* типов Add, Alpha Clip, Alpha Blend, Alpha Sort и Alpha Anti-Aliasing.
- ALL - плавные переходы используются для всех объектов (наименее быстрый вариант).

Вариант “ALL” может существенно снизить производительность приложения, поэтому его следует использовать только тогда, когда это необходимо. Значение параметра по умолчанию - “NON-OPAQUE”.

#### Max LOD Hysteresis Interval

Длина (в метрах) интервала, который используется для переключения уровней LOD. Половина этого значения прибавляется к пороговому расстоянию или вычитается из него, образуя два разных пороговых расстояния для более высокого и более низкого уровней LOD. Делает менее заметным эффект “popping”. По умолчанию этот параметр равен 4.0.

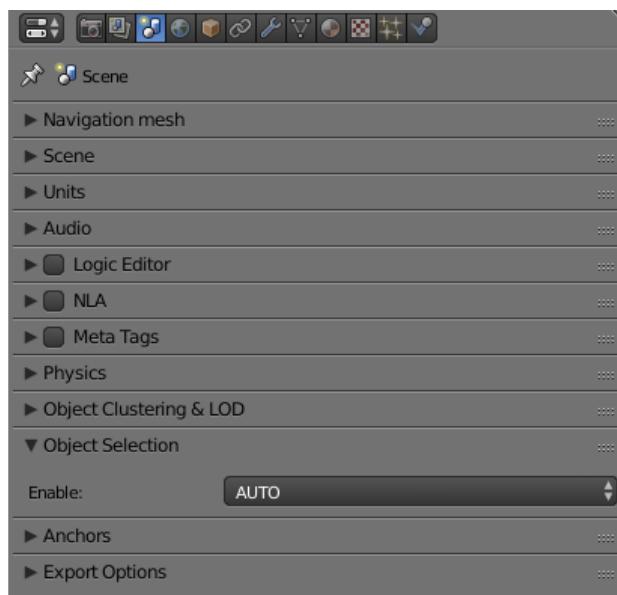
## Object Selection

Настройки параметров выбора объектов с помощью функции API `pick_object()` и *nodовой логики*.

---

**Примечание:** В *просмотрике сцен* выделение включено по умолчанию. Отключить его можно в разделе Tools & Debug.

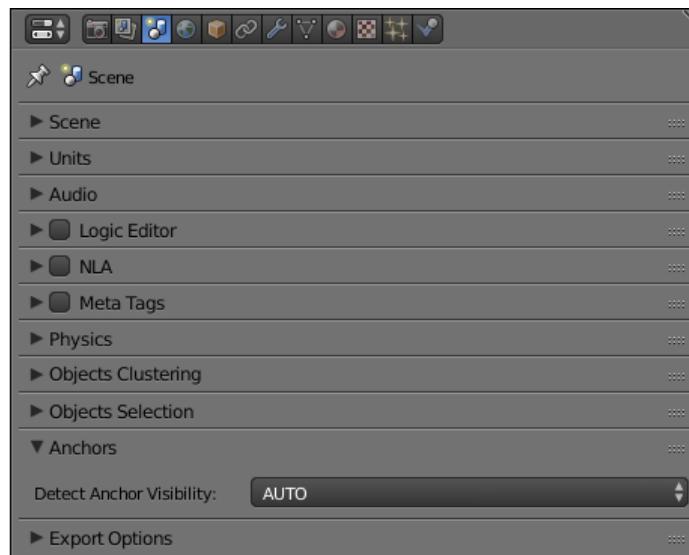
---



**Enable** Параметр, который определяет, могут ли объекты быть выбраны. Может иметь значения **ON**, **OFF** и **AUTO**. По умолчанию имеет значение **AUTO**.

## Anchors

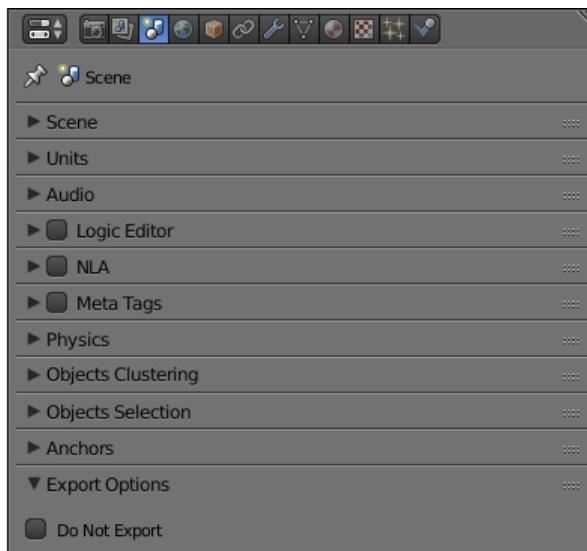
Настройки определения видимости *якорей*.



**Detect Anchors Visibility** Параметр определения видимости якорей. Возможные значения: **ON**, **OFF** и **AUTO**. Значение по умолчанию - **AUTO**.

## Export Options

Параметры экспорта настроек сцены.



**Do Not Export** Игнорировать настройки сцены при экспорте приложения.

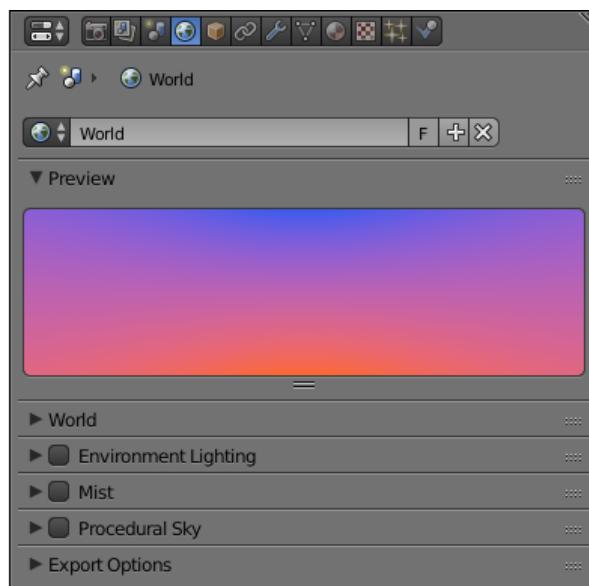
## Вкладка World



На этой вкладке находятся параметры окружающей среды, такие, как настройки неба, тумана и т.п.

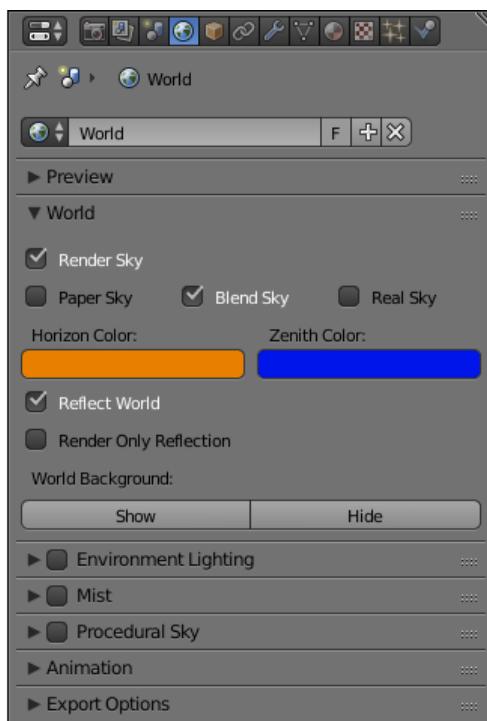
### Preview

Предварительный просмотр окружения.



## World

Настройки параметров неба.



**Render Sky** Отображение неба.

**Paper Sky** Если эта опция включена, градиент неба всегда будет отрисовываться сверху вниз относительно кадра независимо от угла наклона камеры.

**Blend Sky** Плавный переход от цвета горизонта к цвету неба в зените.

**Real Sky** Визуализация неба с горизонтом, зависящим от угла наклона камеры.

**Horizon Color** Цвет неба у горизонта.

**Zenith Color** Цвет неба в зените.

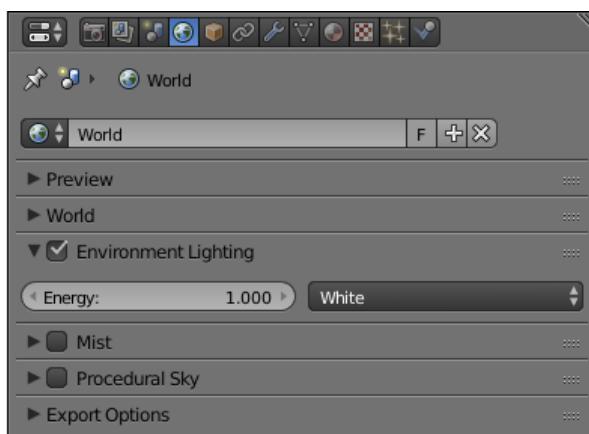
**Reflect World** Отображать небо при расчёте отражений.

**Render Only Reflection** Отображать небо *только* при расчёте отражений.

**World Background** Включает и выключает опцию **World Background** (находится на вкладке **Display** панели **3D View**). При включении этой опции цвета окружающей среды будут отображаться в окне просмотра так же, как в самом движке.

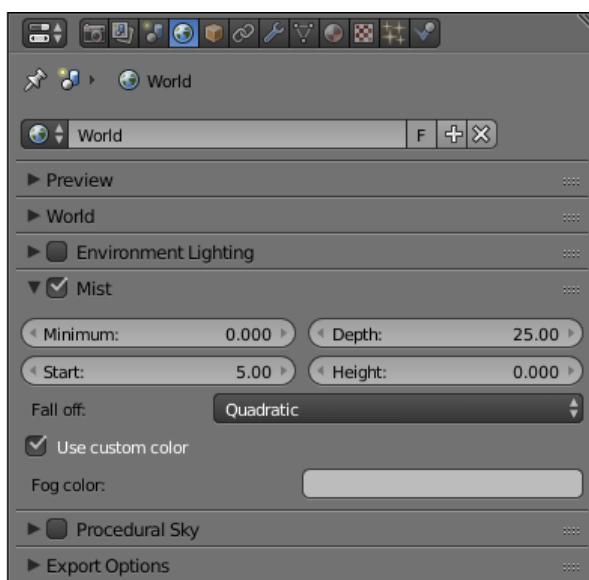
## Environment Lighting

Настройки параметров освещения от окружающей среды. Подробно описаны в [соответствующем разделе](#).



## Mist

Настройки параметров тумана.



**Minimut** Минимальная интенсивность тумана. По умолчанию равна 0.

**Depth** Расстояние от камеры, на котором эффект тумана достигает наибольшей интенсивности. По умолчанию равно 25.

**Start** Расстояние от камеры, на котором начинает проявляться эффект тумана. По умолчанию равно 5.

**Height** Определяет, насколько быстро плотность тумана падает с ростом высоты. По умолчанию равно 0.

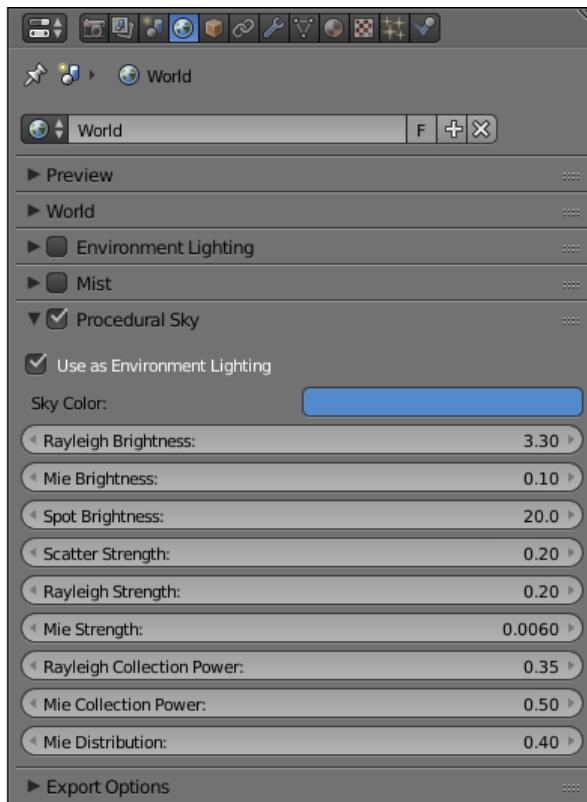
**Fall Out** Правило, по которому возрастает интенсивность тумана между его границами (определяются параметрами **Start** и **Depth**). Доступные значения: **Quadratic**, **Linear**, **Inverse Quadratic**. Значение по умолчанию - **Quadratic**.

**Use Custom Colors** Позволяет задать цвет тумана. По умолчанию включено. Если отключено, будет использоваться стандартный цвет 0.5, 0.5, 0.5.

**Fog Color** Цвет тумана. Может быть настроен, если активирована опция **Use custom colors**. По умолчанию установлен светло-серый цвет (0.5, 0.5, 0.5).

## Procedural Sky

Настройки процедурного неба. Подробно описаны в *соответствующем разделе*.



## Animation

Настройки анимации окружающей среды.



**Apply Default Animation** Если этот параметр включён, движок запускает проигрывание анимации окружающей среды после загрузки сцены.

**Behavior** Устанавливает поведение анимации окружающей среды. Доступные значения:

- **Loop** - анимация воспроизводится циклически.
- **Finish Stop** - анимация проигрывается один раз.
- **Finish Reset** - анимация проигрывается один раз, после чего сбрасывается на нулевой кадр.

Эта опция доступна только при включённом параметре **Apply Default Animation**.

## Export Options

Настройки экспорта параметров окружения.



**Do Not Export** Игнорировать настройки окружения при экспорте приложения.

## Объекты

### Содержание

- *Объекты*
  - *Типы*
  - *Статические и динамические объекты*
  - *Настстройка*
    - \* *Вкладка Object*
    - \* *Вкладка Physics*
  - *Настройки якорей*
  - *Управление перемещением объектов*
  - *Выбор объектов*
  - *Выделение объектов*
  - *Копирование объектов (инстансинг)*
    - \* *Простое копирование*
    - \* *Глубокое копирование*
    - \* *Удаление объектов*
  - *Кватернионы*
    - \* *Пример работы с кватернионами*
  - *Перемещение с помощью векторов TSR*
  - *Визуализация линий*
  - *Уровни детализации (LOD)*
    - \* *Общая информация*
    - \* *Специфика и отличия от BGE*
    - \* *Плавная смена LODов*
    - \* *Гистерезис*

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

### Типы

Движком поддерживаются объекты следующих типов:

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

При экспорте сцены объекты типов CURVE, TEXT, METABALL и SURFACE конвертируются в объекты типа MESH.

## Статические и динамические объекты

Объект может быть статическим или динамическим.

**Статические объекты** - объекты, которые невозможно перемещать, анимировать или изменять любым другим способом во время работы приложения. Для повышения быстродействия, такие объекты могут объединены друг с другом, если они имеют одинаковый материал.

**Динамические объекты** можно перемещать, анимировать и изменять другими способами во время работы приложения. Такие объекты также могут взаимодействовать с другими объектами, в том числе со статическими. Динамические объекты никогда не объединяются друг с другом или со статическими объектами.

Только объекты типов MESH и EMPTY могут быть как статическими, так и динамическими. Все остальные объекты, например CAMERA или ARMATURE, всегда динамические.

Статические объекты типа MESH рендерятся значительно быстрее динамических, поэтому для повышения производительности рекомендуется сводить количество динамических мешей к минимуму. Объекты других типов, как статические, так и динамические, не оказывают существенного влияния на производительность.

Объекты, имеющие анимацию, физику или родителя, являющегося динамическим объектом, рассматриваются как динамические. Динамическими считаются также объекты, контролируемые следующими логическими нодами:

- Play Animation
- Transform Object
- Move To

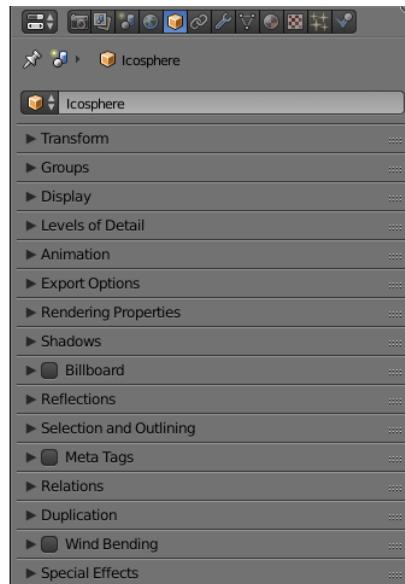
- Inherit Material

Методы API для перемещения, копирования и анимации (как самого объекта, так и его нодового материала) объектов могут применяться только к динамическим объектам. Чтобы обеспечить движение объекта, не имеющего настроек динамического объекта, в его настройках следует активировать опцию **Force Dynamic Object**.

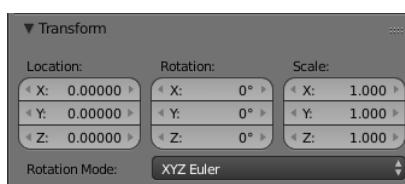
## Настройка

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

## Вкладка Object



## Панель Transform



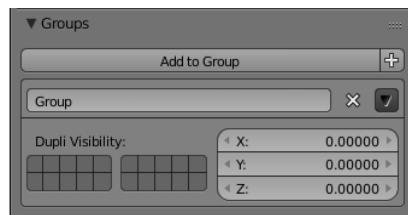
**Transform > Location** Координаты местоположения.

**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.

## Панель Groups

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



## Панель Level of Detail

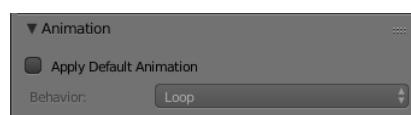


**Levels of Detail > Object** Объект, используемый для данного уровня детализации.

**Levels of Detail > Distance** Расстояние, на котором начинается использование данного уровня детализации.

Использование уровней детализации описано в *соответствующем разделе*.

## Панель Animation

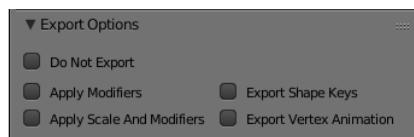


**Animation > Apply Default Animation** Начать воспроизведение в движке связанной с объектом анимации.

**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.

### Панель Export Options



**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** Экспортировать предварительно созданную и сохраненную вертексную анимацию. Актуально только для объектов типа MESH.

**Export Options > Export Shape Keys** Экспортировать 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.

---

### Панель Rendering Properties



**Rendering Properties > Hidden** Если это свойство активировано, объект будет невидим при запуске сцены.

**Rendering Properties > Hidden Children** Все потомки объекта, а также потомки его потомков будут невидимы при запуске сцены. Этот параметр доступен только если включен параметр **Rendering Properties > Hidden** и у объекта имеются потомки.

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

Этот параметр недоступен для объектов типа Empty.

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

Этот параметр недоступен для объектов типа **Empty**.

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

**Rendering Properties > Dynamic Geometry & Materials** Включает возможность использовать API для обновления геометрии объекта и наследования материалов.

**Rendering Properties > Line Rendering** Активирует использование объекта для рендеринга **линий**.

Этот параметр доступен только для объектов типа **Empty**.

## Панель Shadows

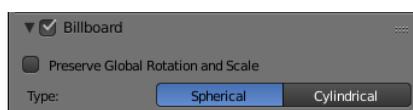


**Shadows > Cast** Объект будет отбрасывать тени.

**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.

## Панель Billboard



**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**.

## Панель Reflections

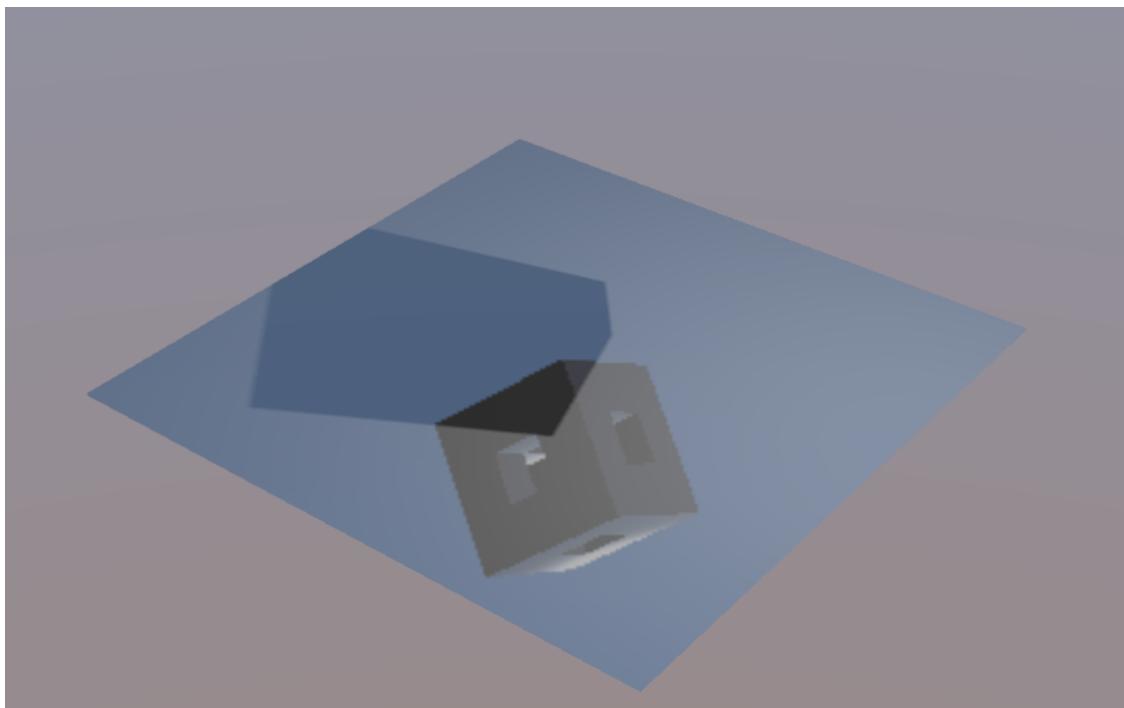


**Reflections > Reflective** При включении объект будет отражаться от зеркальных поверхностей.

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

---

**Примечание:** Если опция **Reflexible Only** включена одновременно с **Shadows > Cast Only**, движок не будет отображать сам объект, но будет расчитывать отражение объекта и тень от него, как показано на изображении ниже.

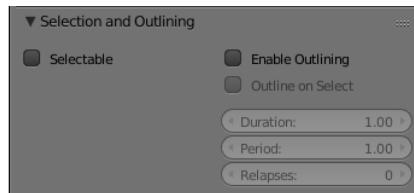


---

**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.

## Панель Selection and Outlining

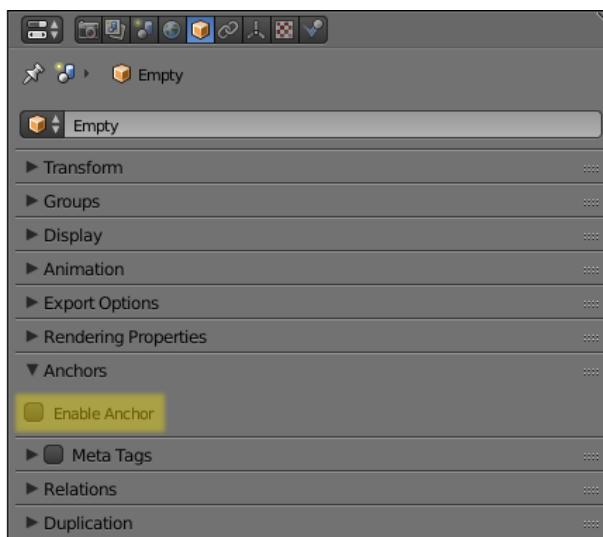


**Selection and Outlining > Selectable** Разрешить выделение объекта с помощью мыши или другого устройства ввода.

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

## Панель Anchors

**Anchors > Enable Anchor** Активирует интерфейс добавления якорей (двухмерных меток) к объектам. Доступно только для объектов типа EMPTY. Якоря подробно описаны в [соответствующем разделе](#).



## Панель Meta Tags



**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.

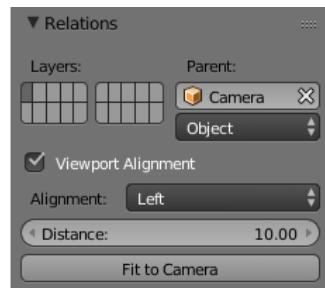
**Meta Tags > Description Source** Тип (источник) описания объекта: текст либо файл с текстом.

## Панель Relations



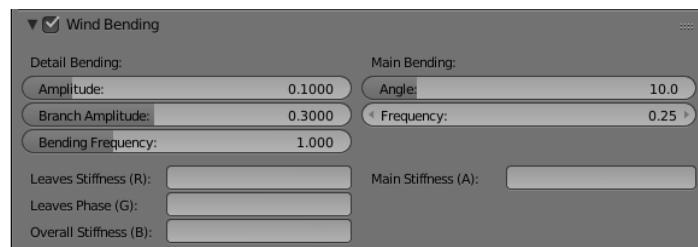
**Relations > Parent** Указатель на родительский объект.

Если в качестве родительского объекта выбрана камера, становятся доступными настройки **Viewport Alignment**.



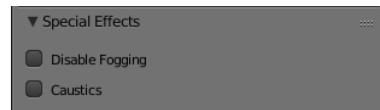
Они могут применяться для привязывания объекта к текущей камере. Описаны в [главе “Камера”](#).

## Панель Wind Bending



**Wind Bending** Включить процедурную анимацию под действием ветра. Подробно описана в разделе, посвященном [рендерингу наружных сцен](#).

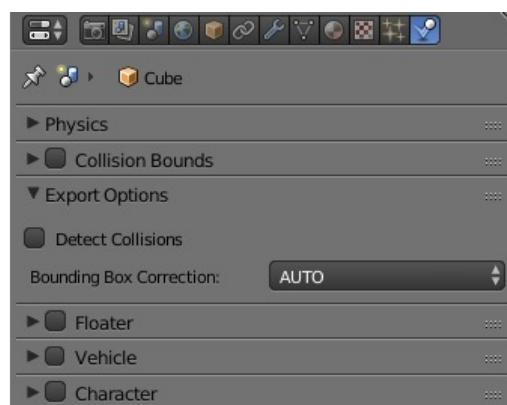
## Панель Special Effects



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

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

## Вкладка Physics



*Detect Collisions* Активировать связанную с объектом физику.

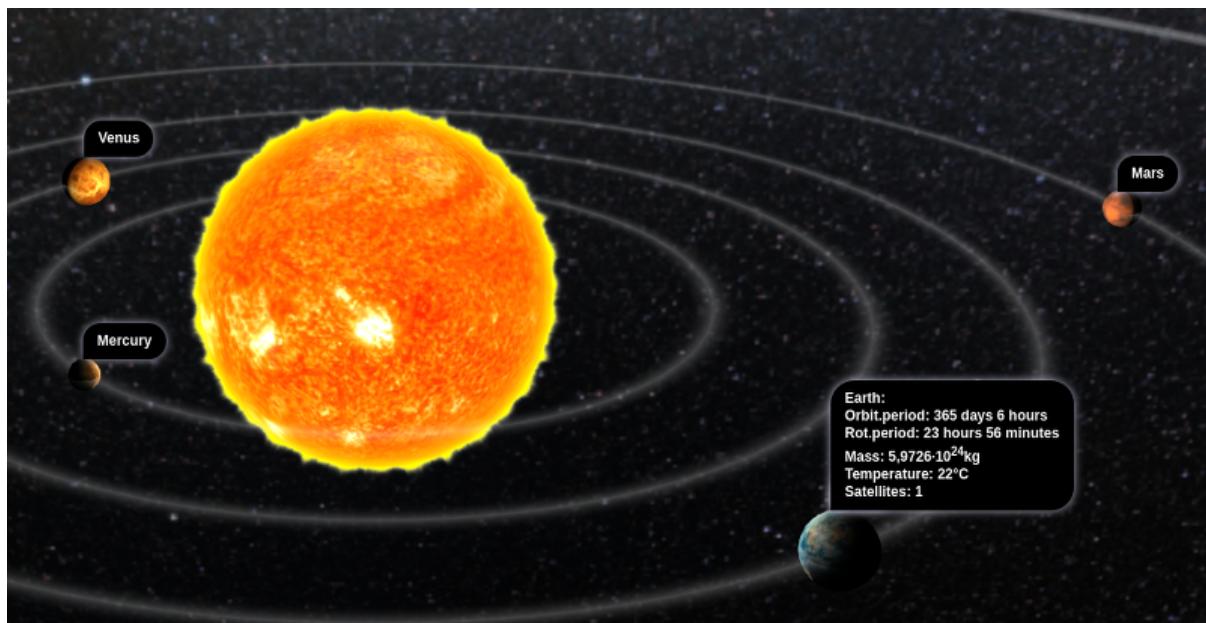
*Floating* Сделать объект плавучим. Подробное описание настроек плавучих объектов можно найти в разделе *физика*.

*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*.

## Настройки якорей

Якоря могут применяться для добавления поясняющих надписей к 3D-объектам. Вне зависимости от позиции камеры, надпись всегда располагается рядом с объектом и следует за ним в случае его анимации.



Аннотация может быть полностью создана в Blender. Для этого достаточно разместить в нужном месте объект типа **Empty** и включить его использование в качестве “якоря”. Текст аннотации можно изменить в полях **Title** и **Description**, доступных на вкладке **Meta Tags**.



**Enable Anchor** Активирует интерфейс добавления якорей (двухмерных меток) к объектам. Доступно только для объектов типа **EMPTY**.

**Type** Тип якоря

- **Annotation** - информация из *метатегов* объектов отображается в раскрывающихся информационных окнах стандартного дизайна.
- **Custom Element** - в качестве якоря можно назначить произвольный HTML-элемент с текущей веб-страницы
- **Generic** - якорь, положение которого можно определить с помощью API модуля `anchors`.

Значение по умолчанию - **Annotation**

**HTML Element ID** Задаёт ID HTML-элемента, который будет использоваться в качестве якоря. Доступно только если параметр **Type** имеет значение **Custom Element**.

**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.

**Max Width** Ограничить всплывающее информационное окно аннотации выбранным значением (в CSS пикселях). Доступно только если параметр Type имеет значение Annotation. Значение по умолчанию - 250.

## 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()** Установить коэффициент 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).

## Выбор объектов

Для того, чтобы производить какие-либо манипуляции с трёхмерным объектом, необходимо получить ссылку на него. Для этого используется ряд функций API. Ссылка на объект имеет тип `Object3D`.

**get\_object\_by\_name()** Получить ссылку на объект, основываясь на известном имени объекта.

```
// ...
var obj = m_scenes.get_object_by_name("Object");
// ...
```

**get\_object\_by\_dupli\_name()** Получить ссылку на скопированный объект, используя имя скопированного объекта и имя объекта типа `EMPTY`, который использовался для копирования.

```
// ...
var obj = m_scenes.get_object_by_dupli_name("Empty", "Object");
// ...
```

**get\_object\_by\_dupli\_name\_list()** Получить ссылку на скопированный объект, используя списки (массивы элементов типа `String`) имён скопированных объектов и объектов типа `EMPTY`, использованных для копирования.

```
// ...
var obj = m_scenes.get_object_by_dupli_name_list(["Empty1", "Empty2", "Object"]);
// ...
```

**get\_object\_name\_hierarchy()** Возвращает список (массив) имён предков заданного объекта, начиная от старшего предка и заканчивая самим объектом.

```
// ...
var names = m_scenes.get_object_name_hierarchy(obj);
// ...
```

**check\_object\_by\_name()** Проверить, существует ли в сцене объект с указанным именем.

```
// ...
var object_exists = m_scenes.check_object_by_name("Cube");
// ...
```

**check\_object\_by\_dupli\_name()** Проверить, существует ли в сцене скопированный объект с заданным именем и заданным именем объекта типа EMPTY, который использовался для копирования.

```
// ...
var object_exists = m_scenes.check_object_by_dupli_name("Empty", "Cube");
// ...
```

**check\_object\_by\_dupli\_name\_list()** Проверить, существует ли в сцене скопированный объект, используя списки (массивы элементов типа String) имён скопированных объектов и объектов типа EMPTY, использованных для копирования.

```
// ...
var object_exists = m_scenes.check_object_by_dupli_name_list(["Empty1", "Empty2", "Object"]);
// ...
```

**get\_object\_name()** Получить имя объекта, основываясь на известной ссылке на этот объект.

```
// ...
var object_name = m_scenes.get_object_name(obj);
// ...
```

## 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 *редактора логики*.

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*.

---

**Примечание:** Если выделяемый объект полупрозрачен (имеет тип прозрачности `Blend Add` или `Sort`), outline-анимация будет видна только на тех частях объекта, для которых значение прозрачности `Alpha` выше 0.5.

---

## Копирование объектов (инстансинг)

Часто возникает необходимость копирования объектов (инстансинг) 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

### 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.

### 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. Также при глубоком копировании canvas-текстуры на копиях - именно разные

текстуры, в то время как при простом копировании - одна и та же. Для такого типа копирования на исходном объекте нужно дополнительно выставить опцию *Rendering Properties > 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);
// ...
```

## Удаление объектов

Для удаления объектов, в модуле `scenes.js` предусмотрена функция `remove_object`, куда надо передать объект. Возможно удаление динамических mesh- и empty-объектов.

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

## 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.
- Численная устойчивость - компенсация ошибок умножения, возникающих из-за неточности вычислений 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.

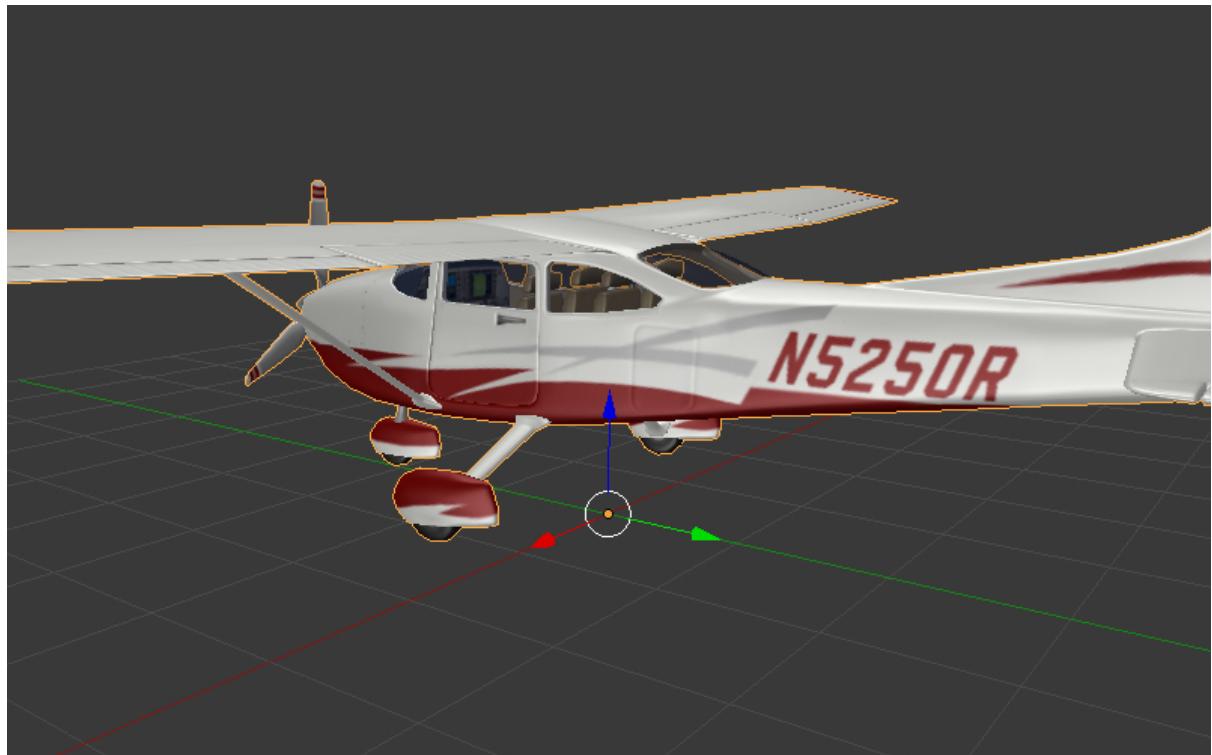
***quat.setAxisAngle*** Кватернион представляет собой иную форму записи поворота относительно произвольной оси (вектора) на произвольный угол. Положительное направление поворота отсчитывается против часовой стрелки, если смотреть с конца вектора. Например вызов `quat.setAxisAngle([1,0,0], Math.PI/3, 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.

## 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им поворот:

- 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_Z_neg = m_quat.setAxisAngle([0, 0, 1], -Math.PI/3, m_quat.create());

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

// left multiply: quat60_Z_neg * quat_old
var quat_new = m_quat.multiply(quat_60_Z_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_Z = new Float32Array([0, 0, 1])
var quat_tmp = new Float32Array(4);
var quat_tmp2 = new Float32Array(4);
...
// rotate
m_quat.setAxisAngle(AXIS_Z, -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);
```

## Перемещение с помощью векторов TSR

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

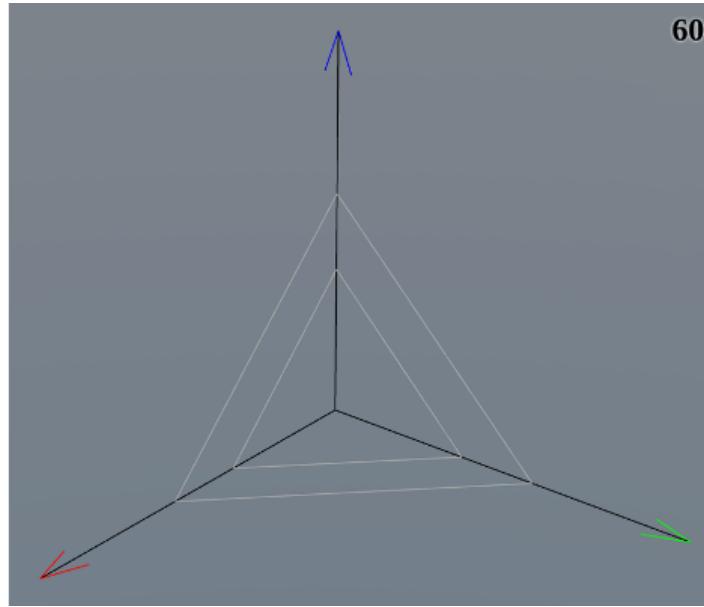
$$[T_x, T_y, T_z, S, R_x, R_y, R_z, R_w]$$

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\_ts()*/get\_ts()** модуля *transform*.

## Визуализация линий

Двигок Blend4Web также способен визуализировать линии, заданные в виде набора точек.



Методы API для собственно рендеринга линий находятся в модуле *geometry*, а стиль (т.е. толщина и цвет) линии устанавливается с помощью метода *set\_line\_params()* из модуля *material*.

Для рендеринга линий в сцене должен присутствовать по крайней мере один объект типа *Empty* с активированной опцией *Line Rendering*.

Для того, чтобы визуализировать линию, необходимо прежде всего *получить ссылку* на объект типа *Empty*. При рендеринге линий начальная точка этого объекта будет использоваться как центр координат.

Сама линия визуализируется при помощи метода API *draw\_line()*. Его первый параметр - ссылка на объект типа *Empty*, а второй - массив вершин, используемых для построения линии. У этого метода также есть третий (опциональный) параметр, определяющий, визуализируется ли непрерывная линия или набор отрезков, заданных парами вершин (в этом случае количество вершин должно быть чётным).

Если объект типа *Empty* используется для рендеринга более чем одной линии за раз, видна будет только последняя линия. Для рендеринга нескольких линий следует использовать несколько *Empty*.

Следующий пример демонстрирует визуализацию фрагмента фигуры, приведённой на изображении выше, с помощью линий:

```

var m_scenes      = require("scenes");
var m_material    = require("material");
var m_geometry    = require("geometry");
var m_rgba        = require("rgba");

...

//setting up Empty object for line rendering
var line_1 = m_scenes.get_object_by_name("MyEmpty_1");

//setting up style parameters for the lines
m_material.set_line_params(line_1, { width: 3
                                      color: m_rgba.from_values(0, 0, 0, 1.0)
                                    });

//coordinates for main axes
var points_1 = new Float32Array([0, 0, 0, 5, 0, 0, 0, 0, 0, 0, 0, 5, 0, 0, 0, 0, 0, 0, 0, 5]);

//drawing main axes
m_geometry.draw_line(line_1, points_1, true);

```

Этот фрагмент кода визуализирует только главные координатные оси, т.к. полный код был бы слишком длинным и монотонным. Прочие элементы объекта создаются тем же образом.

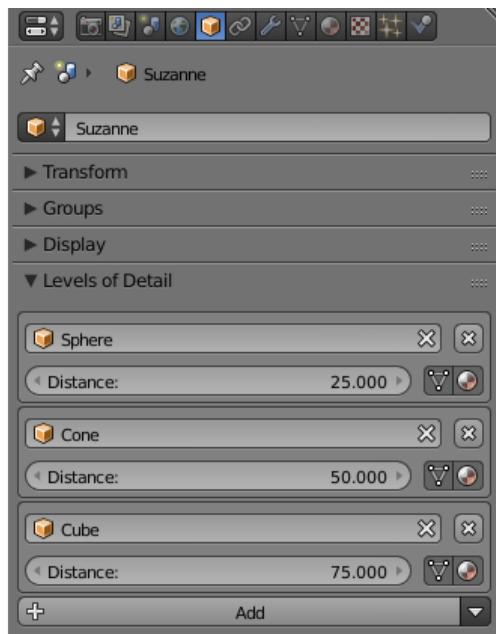
## Уровни детализации (LOD)

### Общая информация

ЛОДы (Levels Of Detail) представляют собой несколько уровней детализации одного объекта. Они применяются для оптимизации, которая состоит в том, что при удалении отрисовывается менее детальный объект, например, низкополигональный с примитивным материалом. Это позволяет снизить нагрузку на видеокарту, при этом не сильно потеряв в качестве изображения, т.к. на дальней дистанции многие детали оригинального объекта все равно не будут различимы.

Использование ЛОДов оправдано на больших сценах со множеством объектов, в других случаях выигрыш в FPS может быть и не таким существенным. Наиболее подходят для этого крупные игровые локации, вроде острова, сельской местности, города и т.д.

Настройка ЛОДов осуществляется в Блендере при помощи *панели Levels of Detail*.



Здесь для каждого конкретного объекта можно настроить его уровни детализации и расстояние, на котором будет происходить переключение между ними. В качестве такого уровня предлагается выбрать другой объект. Очевидное правило - чем ниже находится уровень в списке, тем более простым должен быть соответствующий объект.

Упрощать можно по-разному: можно уменьшать полигональность объекта, облегчать материалы (особенно подовые), не использовать эффекты вроде *теней*, *анимации*, *Wind Bending'a* и т.д. - все, что даст удовлетворительный результат при взгляде на этот объект с указанной дистанции.

Количество уровней стоит настраивать по своему усмотрению с учетом общего масштаба сцены.



Fig. 10.1: Уровни детализации деревьев: слева - оригиналъный объект, справа - его LOD

## Специфика и отличия от BGE

Есть ряд рекомендаций по настройке ЛОДов для использования их в движке. Также, как известно, в Блендере в режиме BGE присутствует и своя система ЛОДов, настраиваемая похожим образом, но с существенными отличиями, которые стоят упоминания.

1. При назначении ЛОД-объекта на другой объект, позиция ЛОДа никак не изменится. Поэтому, чтобы в движке объекты совпали, необходимо вручную разместить их в одном и том же месте в Блендере. Здесь кроется отличие от BGE, где при выборе ЛОДа, он автоматически смещается в центр объекта, который его использует.
2. Если требуется назначить один и тот же ЛОД-объект на множество разных объектов (например, однотипные деревья, дома, машины, кусты...), то необходимо его скопировать и назначать уже его копии, т.е. размножить в стольких экземплярах, сколько объектов его будут использовать. В отличие от BGE, где при выборе одного ЛОД-объекта для многих он автоматически дублируется, при загрузке в движке этого не происходит. Для упрощения создания однотипных объектов рекомендуется объединять главный объект и все его ЛОДы в одну группу, которую затем просто дублировать при помощи EMPTY-объектов.
3. Если предполагается, что объект будет изменять свое положение в пространстве, то стоит привязать к нему его ЛОДы родительской связью в Блендере, чтобы они повторяли его трансформации.

## Плавная смена ЛОДов

Один из наиболее заметных недостатков системы ЛОДов - резкая смена уровней детализации, очевидный и иногда раздражающий эффект.

В движке доступна возможность плавной смены уровней на основе альфа-теста.

The option used for this is called **LOD Smooth Transitions** and can be found on the **Scene->Object Clustering & LOD panel**.

С её помощью можно указать движку использовать ли плавную смену, и если да, то для каких типов объектов. Стоит отметить, что её включение может ощутимо снизить производительность приложения в зависимости от количества настроенных ЛОД-объектов и их материалов.

---

**Примечание:** Smooth LOD transition is not always required. For example, if you instantly transport your camera to some distant object, you probably don't want LOD switching to occur right in front of it. In cases like this, the `lod_leap_smooth_threshold` parameter should be used. It defines the threshold distance (in meters) that the camera can cover in one frame. If this value is exceeded, LODs are switched instantaneously. This can be used to disable smooth LOD transition when camera is teleported to a new location or simply moves very fast.



Fig. 10.2: Постепенная смена оригинального объекта LODом

```
var m_cfg = require("config");

m_cfg.set("lod_leap_smooth_threshold", 5); // 5 meters
```

## Гистерезис

Смена уровней детализации происходит на определённых дистанциях от объекта, иными словами, границах между уровнями, которые настраиваются на панели **Levels of Detail**. Если камера движется непосредственно вблизи такой границы, то может её часто пересекать, а значит и часто активировать смену ЛОД-объектов. Такая ситуация сделает процесс переключения уровней детализации очень заметной, что нежелательно.

This problem can be negated with the **Max LOD Hysteresis Interval** parameter located on the **Scene->Object Clustering & LOD panel**. It sets an interval (in meters) that is used as a gap between two LOD levels.

Например, при настройке **Max LOD Hysteresis Interval** равной 4м. и границе между уровнями детализации 20 м. переключение между ними в одну сторону будет происходить на расстоянии 18 м., а в другую - на 22 м. Таким образом будет решена проблема частого переключения между ЛОДами, благодаря тому, что между ними не будет четкой границы. Данная настройка адаптивна и подстроится под разную дистанцию ЛОД-объектов, уменьшая установленный интервал, если потребуется.

## Меши

### Содержание

- *Меши*
  - Статические и динамические меши
  - Настройка
    - \* Override Bounding Volumes
  - Морфинг

Меши (англ. Meshes) — подкласс *объектов*. Меши отличаются от объектов других типов тем, что имеют геометрическую структуру, видимую в сцене. Все видимые пользователю объекты сцены либо представляют собой меши, либо конвертируются в меши при экспорте (как объекты типов CURVE, TEXT, METABALL и SURFACE).

*Адд-он Blend4Web* обладает рядом инструментов, которые могут применяться для редактирования мешей. К таким инструментам, в частности, относится *редактор нормалей*.

В этой главе рассмотрены параметры и настройки объектов типа MESH, а также функции API для работы с ними.

## Статические и динамические меши

Все объекты типа MESH можно разделить на статические и динамические.

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

**Динамические меши** - меши, которые не могут быть скомбинированы друг с другом. Объединение статических мешей - 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.

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

## Настройка

Меша обладают всеми параметрами *объектов*, а также некоторыми дополнительными параметрами.

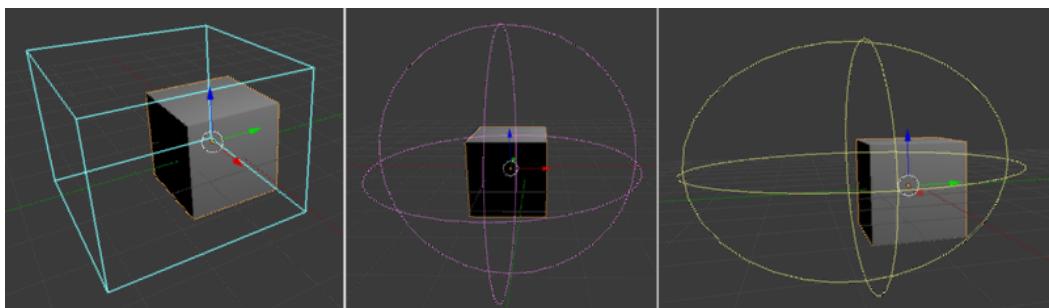
### Override Bounding Volumes

Эта вкладка используется для настройки ограничивающих объёмов. Такие объёмы используются при определении видимости объекта, для расчёта столкновений и физических взаимодействий. Манипулируя ими, можно добиться различных эффектов.



Раздел *Override Bounding Volumes* расположен на панели инструментов Blend4Web (рядом с *редактором нормалей*).

#### Show Boundings



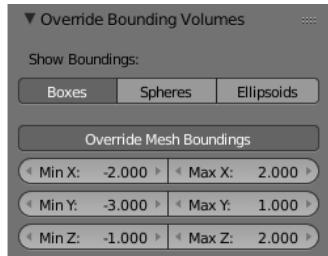
Отобразить ограничивающий объём во видпорте Blender. Работает только если активирована опция *Override Mesh Boundings*. Объём может быть представлен в виде параллелепипеда либо в виде комбинации окружностей или эллипсов. Одновременно может работать более одного метода визуализации (в том числе все три).

---

**Примечание:** Метод визуализации не влияет на форму ограничивающего объёма, только на его представление во видпорте.

---

### Override Mesh Boundings



Эта кнопка включает использование нестандартного ограничивающего объёма. Ограничивающий объём имеет форму параллелепипеда, центр которого совпадает с точкой привязки объекта.

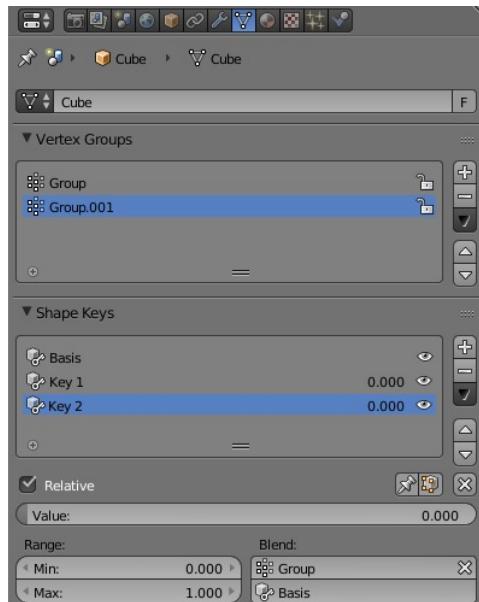
**MinX и MaxX** Размеры ограничивающего объёма по оси Z. По умолчанию  $MinX = -1$ ,  $MaxX = 1$ .

**MinY и MaxY** Размеры ограничивающего объёма по оси Y. По умолчанию  $MinY = -1$ ,  $MaxY = 1$ .

**MinZ и MaxZ** Размеры ограничивающего объёма по оси Z. По умолчанию  $MinZ = -1$ ,  $MaxZ = 1$ .

## 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);
// ...
```

# Редактор нормалей

---

## Содержание

- *Редактор нормалей*
  - Основные особенности редактора нормалей:
  - Интерфейс
  - Activate
  - Show Normals
  - Rotation
  - Режимы Absolute и Offset
  - Split Normals
  - Average
  - Restore
  - Tree, Foliage и Face
  - Copy/Paste
  - Copy From Mesh

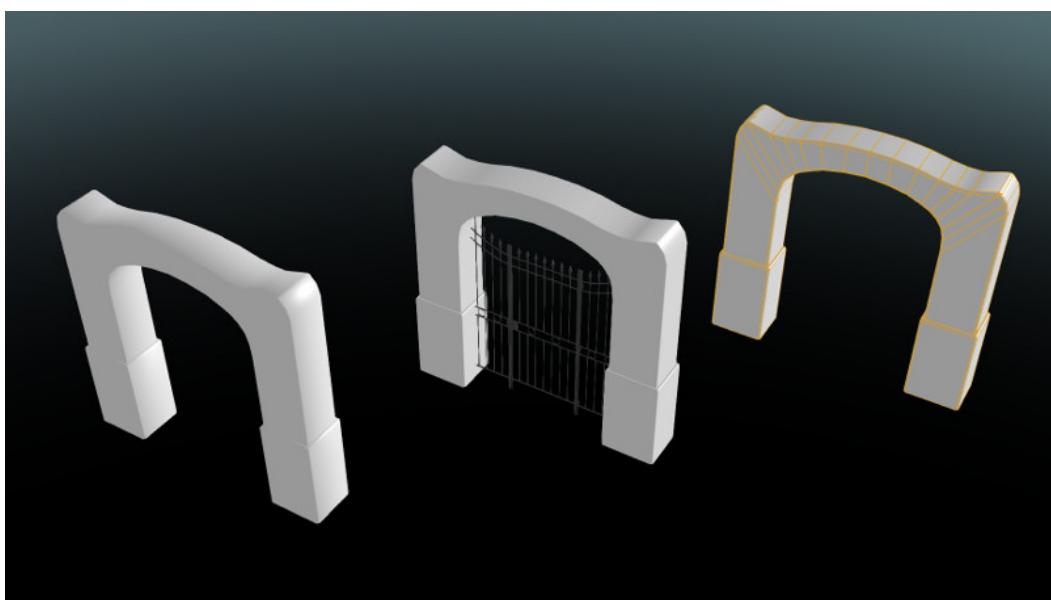
Редактирование нормалей - достаточно простой и эффективный способ изменить собственное затенение трехмерной модели, не усложняя ее геометрию.

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.

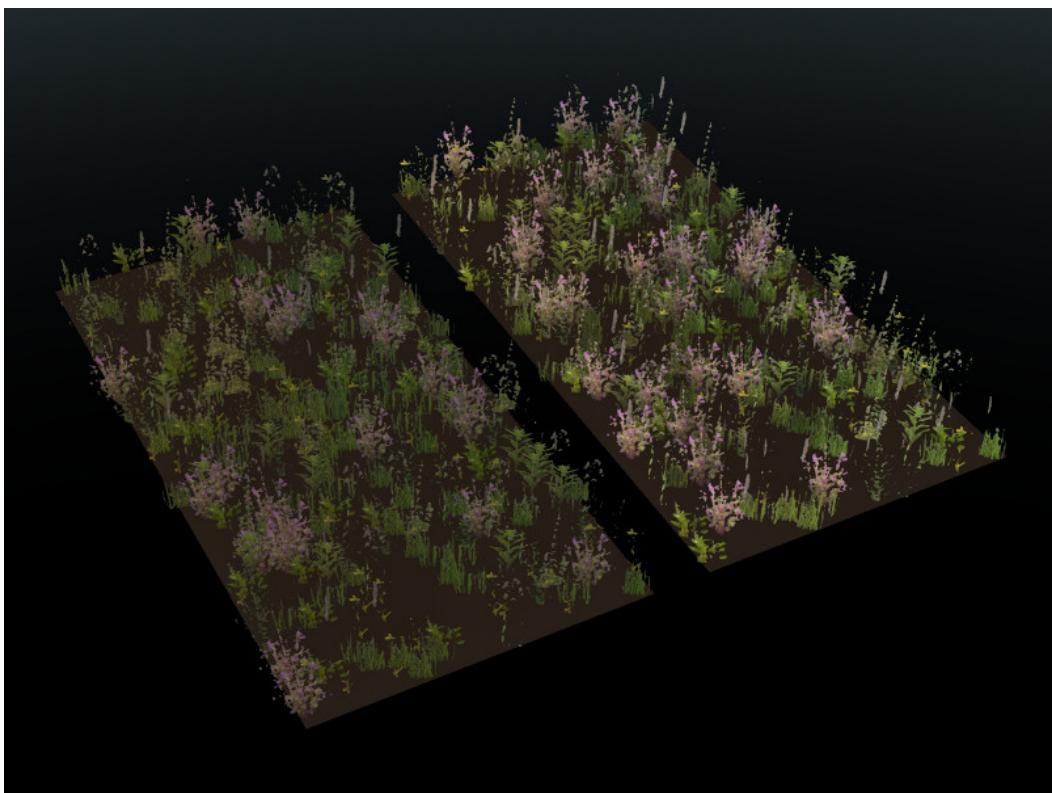
Пример работы редактора нормалей:



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



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



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



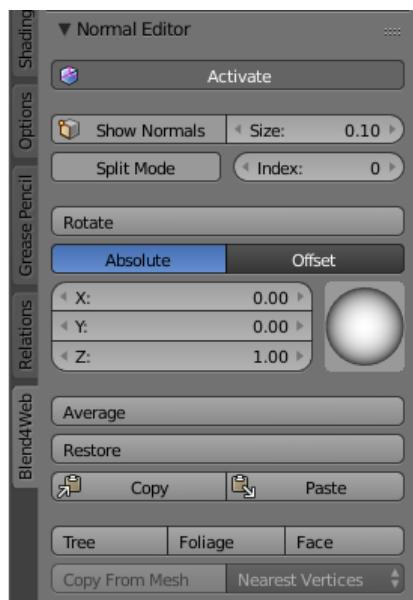
Слева - очки с обычным затенением, справа - затенение геометрии с редактированными нормалями.

## Основные особенности редактора нормалей:

1. для хранилища массива данных с направлениями вертексных 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.

## Интерфейс

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.



### Activate

Кнопка **Activate** включает режим редактирования вертексных нормалей.

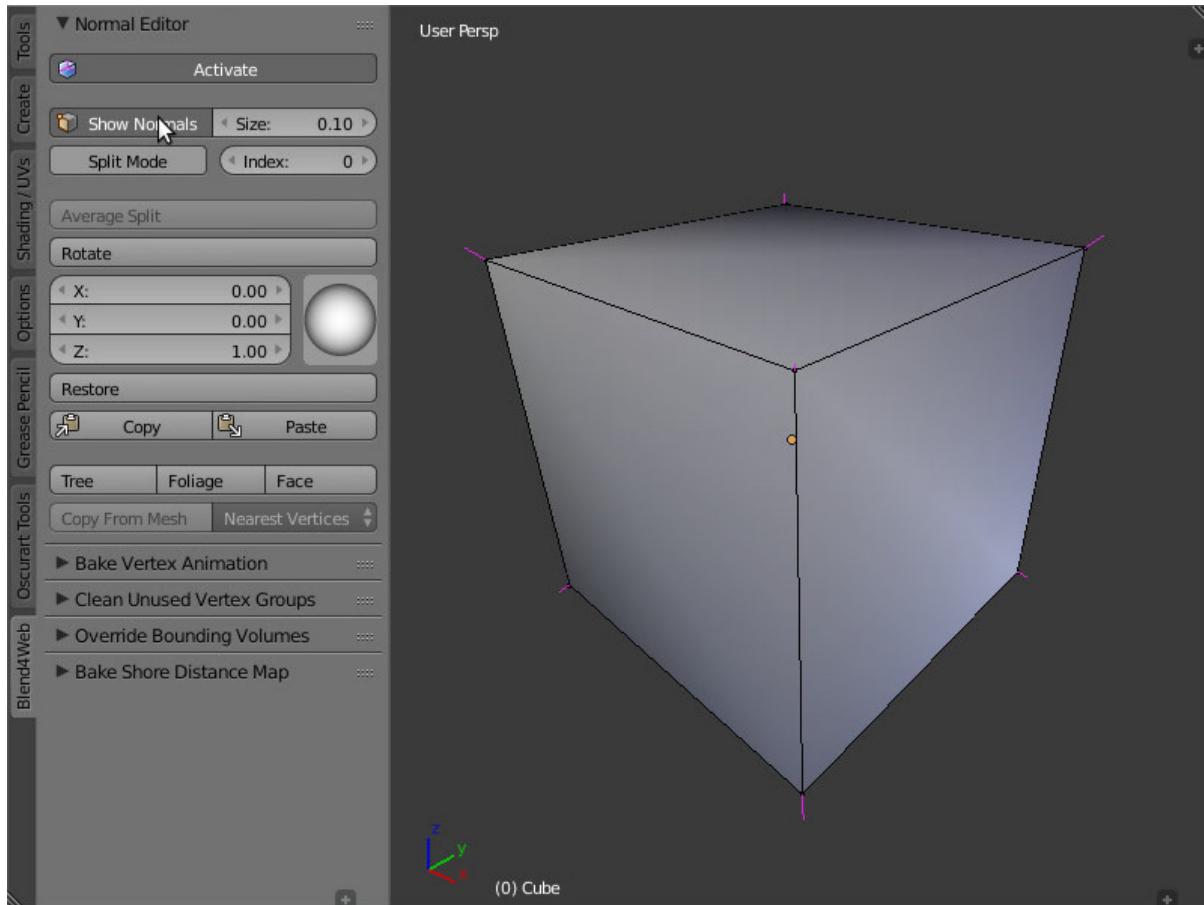
Достаточно в режиме **Edit Mode** нажать кнопу **Activate**, и можно приступать к редактированию вертексных нормалей. Пока она остаётся в активном режиме, затенение объекта и его экспорт будут осуществляться с учетом редактированных вертекс-

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

## Show Normals

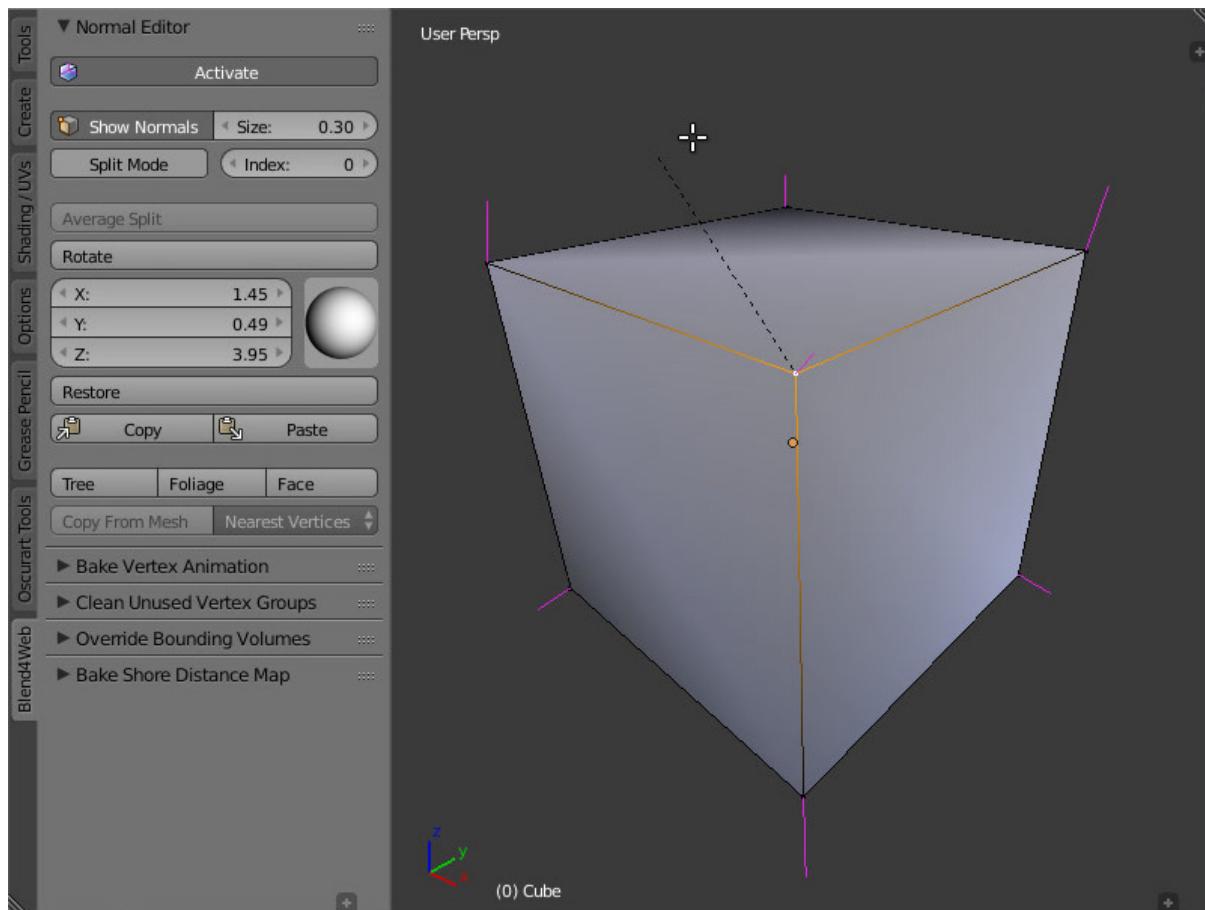
Для удобства в панель продублирована оригинальная кнопка Blender, включающая отображение маркеров вертексных нормалей во Viewport'e, а так же настройка их длины.

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**.



## Rotation

При помощи этого инструментария можно вращать нормаль каждого вертекса индивидуально. Функция **Rotate** также доступна по горячим клавишам **Shift+Ctrl+R** и позволяет вращать вертексные нормали в привычном для пользователя Blender режиме.



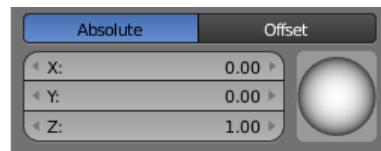
Выделив один или несколько вертексов, которые хотите изменить, вращайте их, используя визуальную сферу на панели редактора вертексных нормалей Blend4Web, или задайте направление каждой координате в цифровом значении.

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) и вводить угол поворота при помощи цифровых клавиш.

## Режимы Absolute и Offset

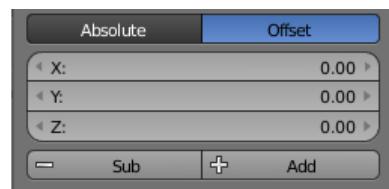
Редактор нормалей может функционировать в двух режимах: режиме **Absolute** и режиме **Offset**. Режим может быть выбран с помощью двух кнопок, расположенных прямо под вышеописанной кнопкой **Rotate**. По умолчанию используется режим **Absolute**.

## Режим Absolute



В этом режиме редактор нормалей работает по умолчанию. В нём координаты нормального вектора абсолютны и могут быть заданы с помощью полей X, Y и Z либо с помощью сферического контроллера, расположенного рядом с ними.

## Режим Offset

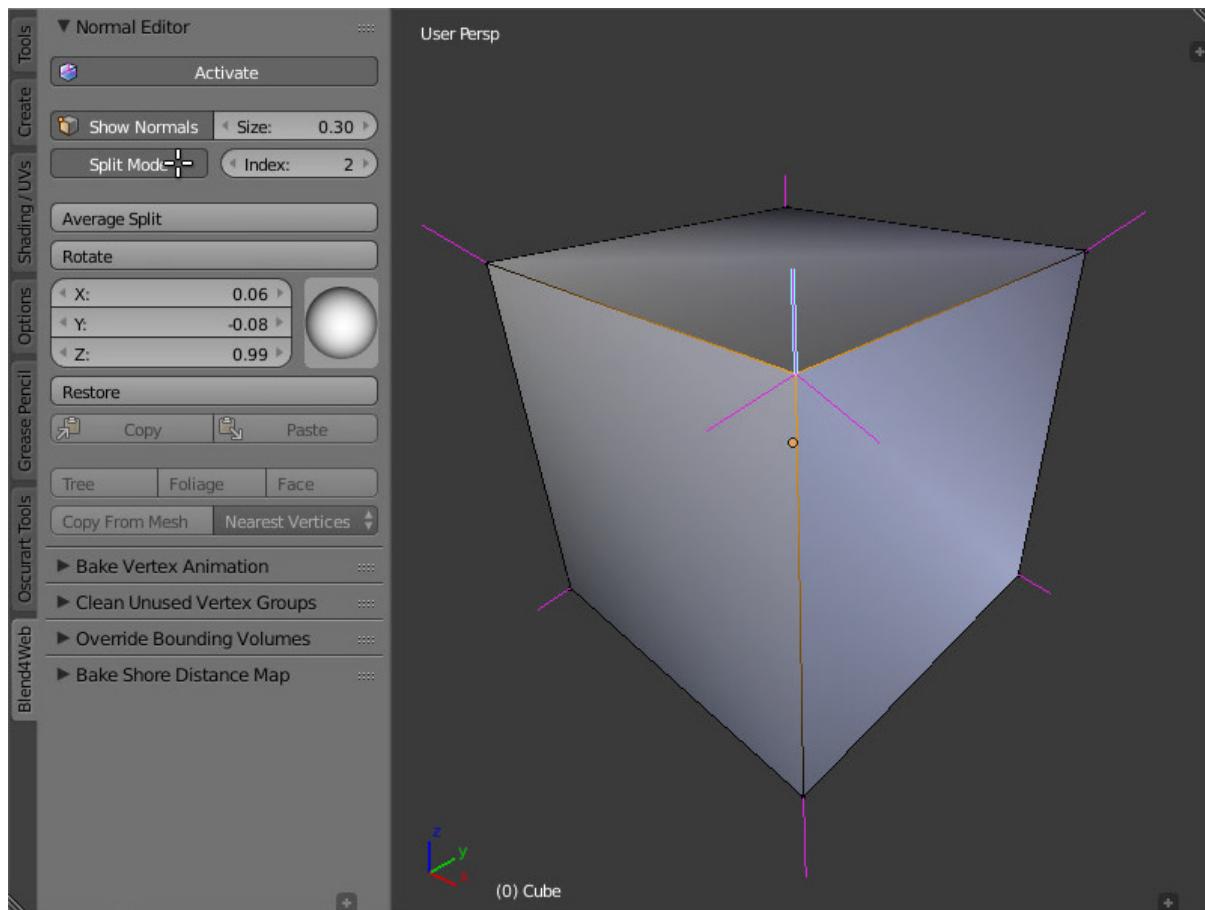


В этом режиме к нормальному вектору прибавляется вектор, заданный пользователем.

Поля X, Y и Z используются для задания соответствующих компонент вектора, а кнопки Sub и Add определяют, вычитается ли пользовательский вектор из нормального или складывается с ним.

## Split Normals

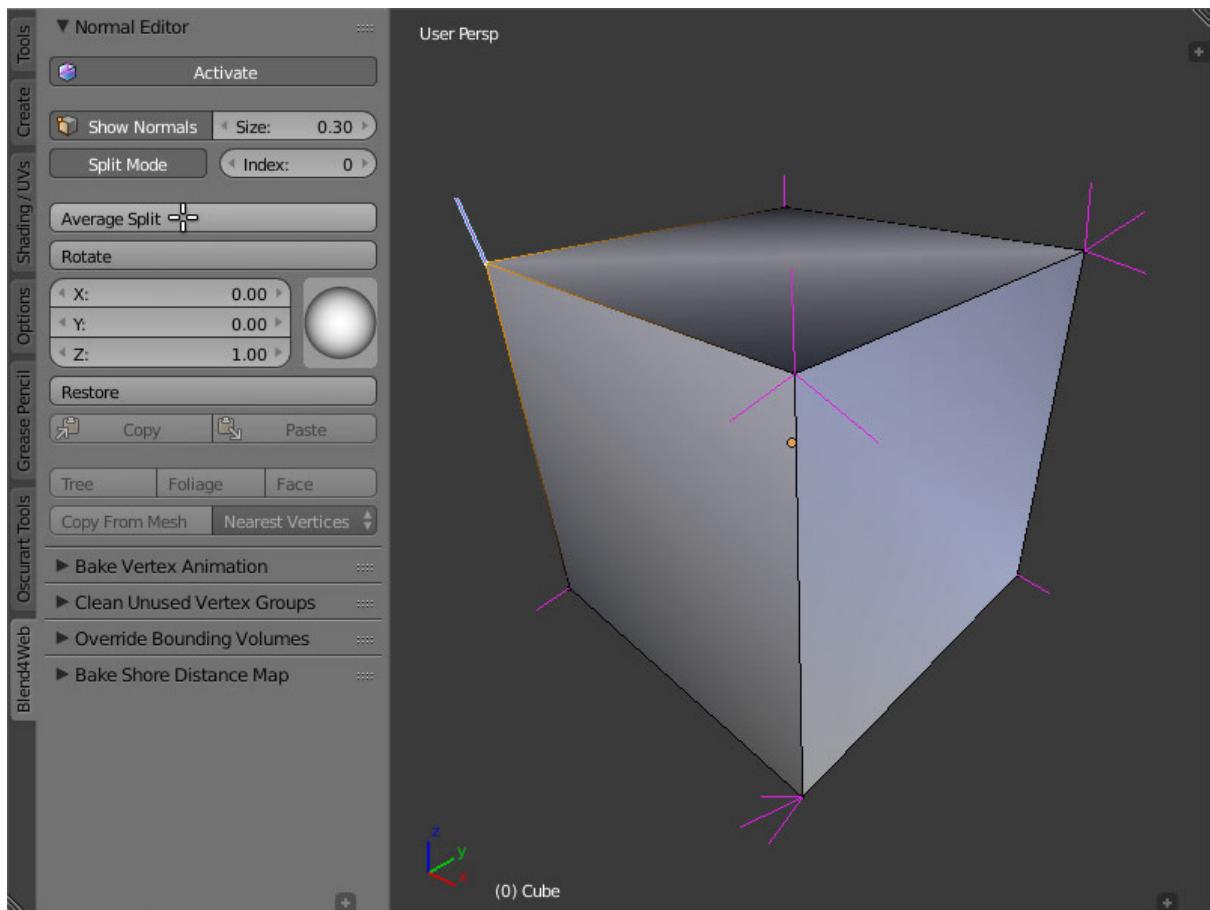
Режим Split Normals позволяет редактировать вертексные нормали индивидуально для каждой грани (Face в Blender), образующей редактируемую вершину (Vertex в Blender). Переключатель вертексных индексов позволяет перемещаться между нормалями разделенного вертекса.



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.

## Average

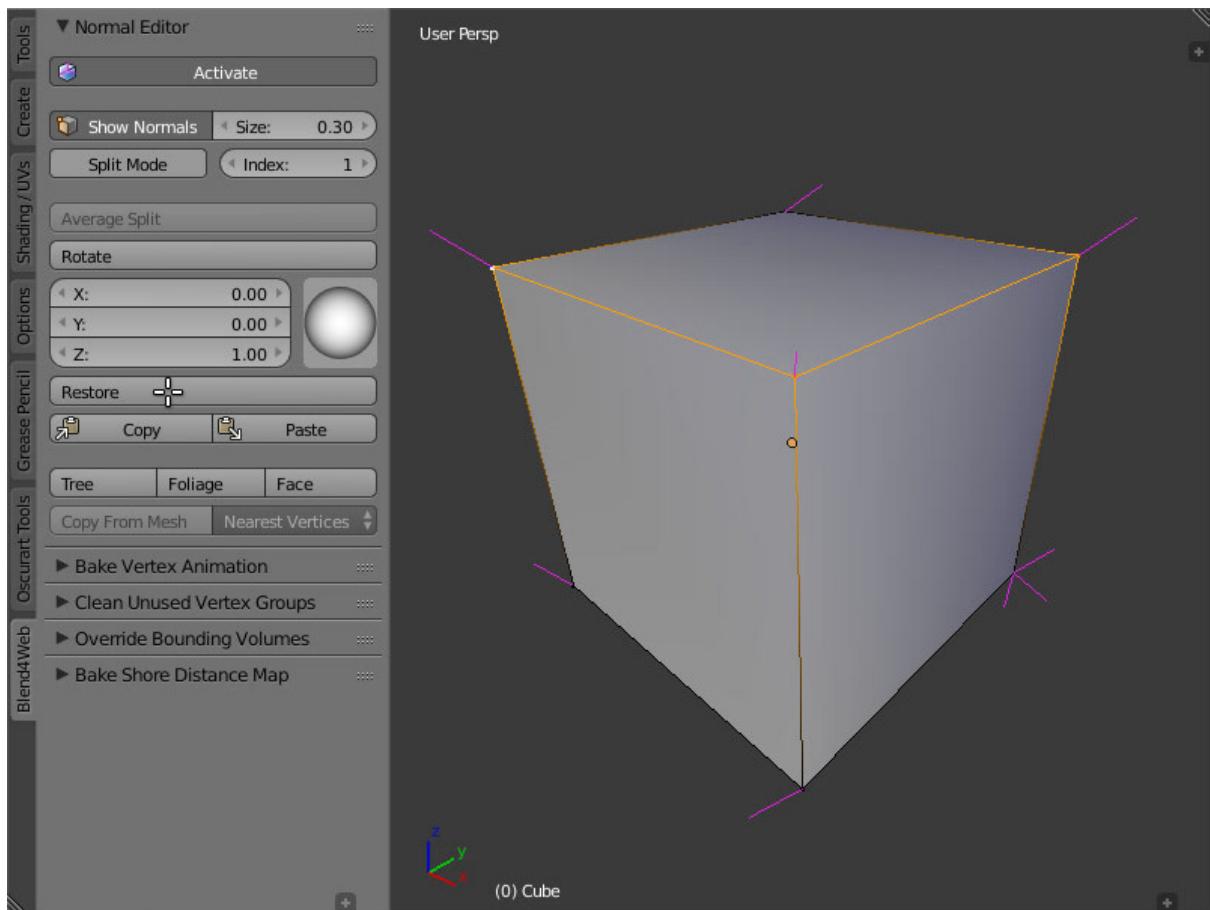
**Average** усредняет направление вертекской нормали, исходя из направлений нормалей, разделенных функцией **Split Normals**.



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**.

## Restore

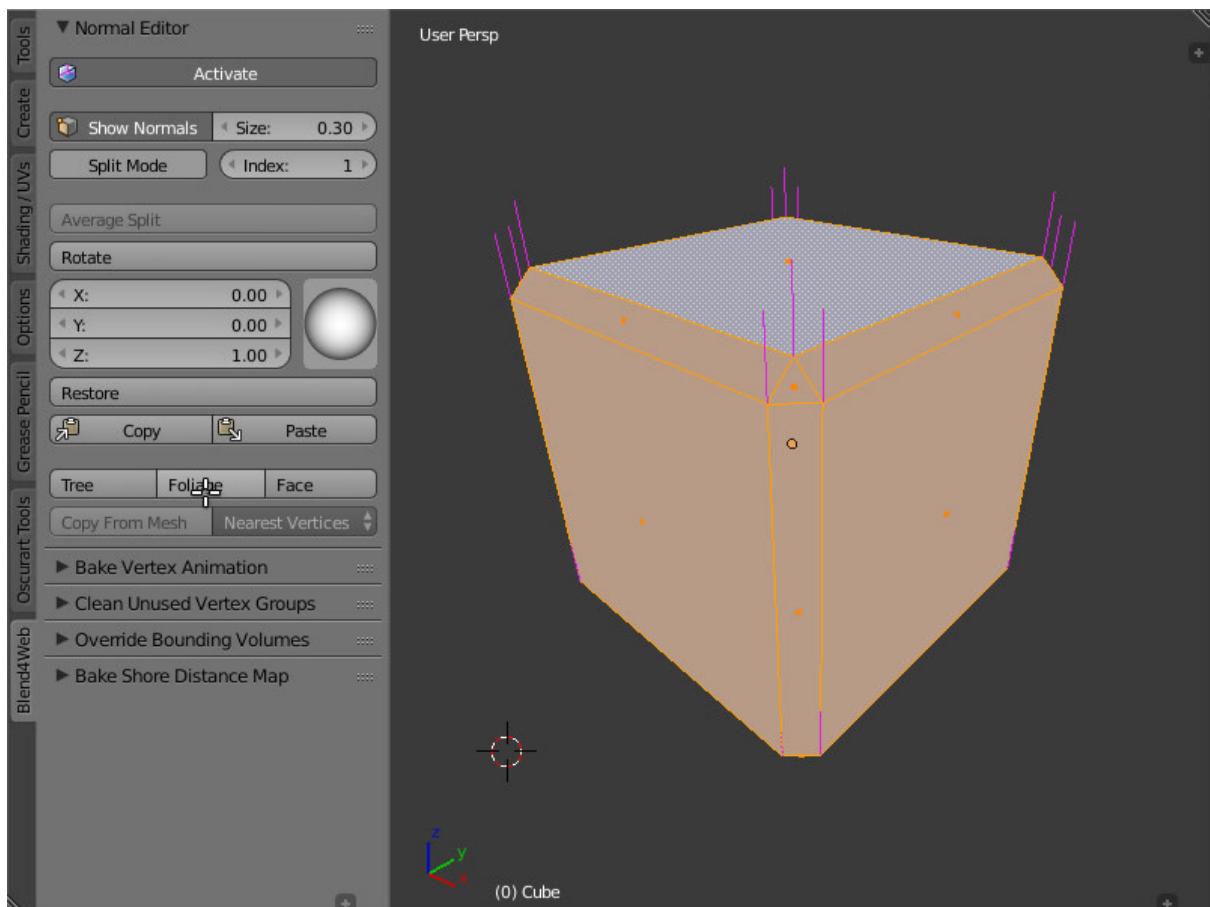
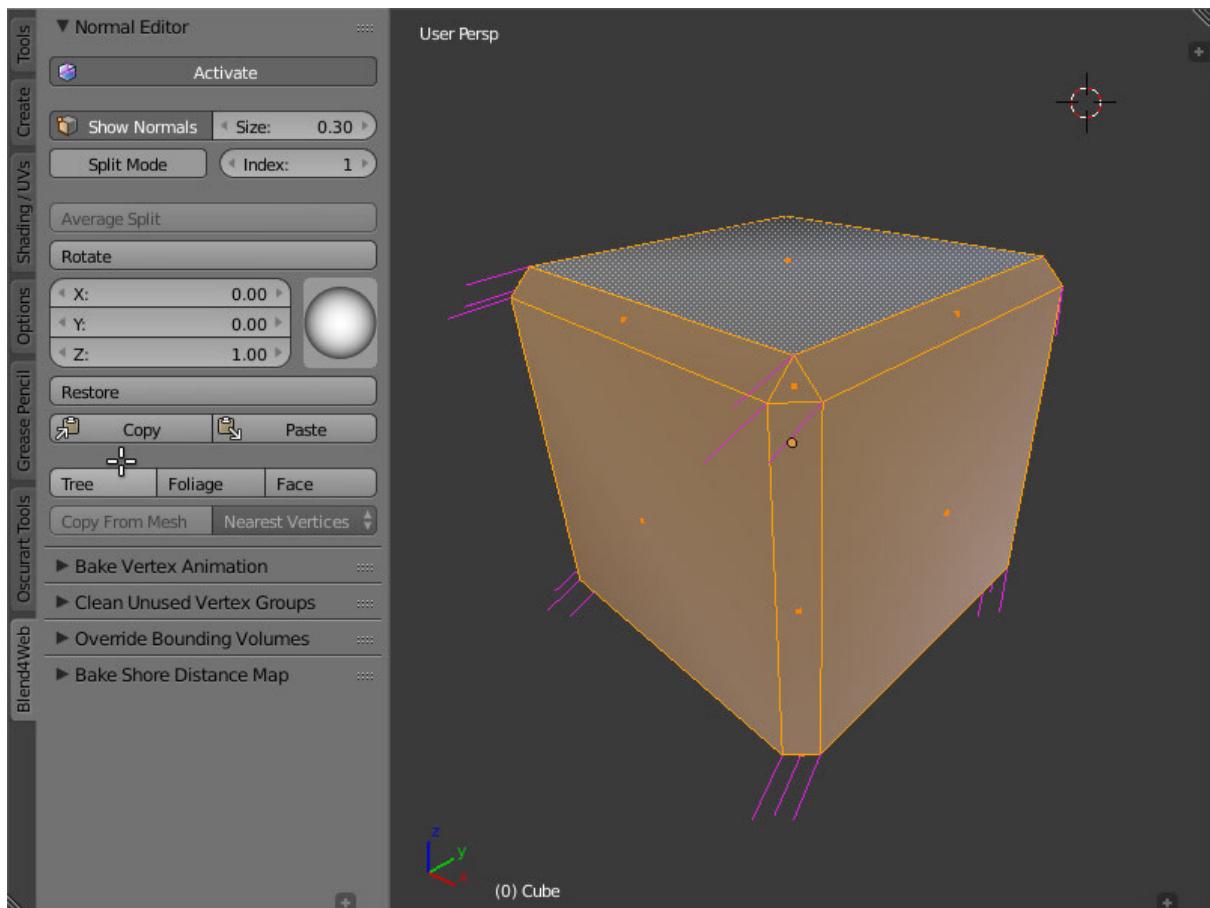
Кнопка **Restore** восстанавливает исходное направление нормалей выделенных вертексов.

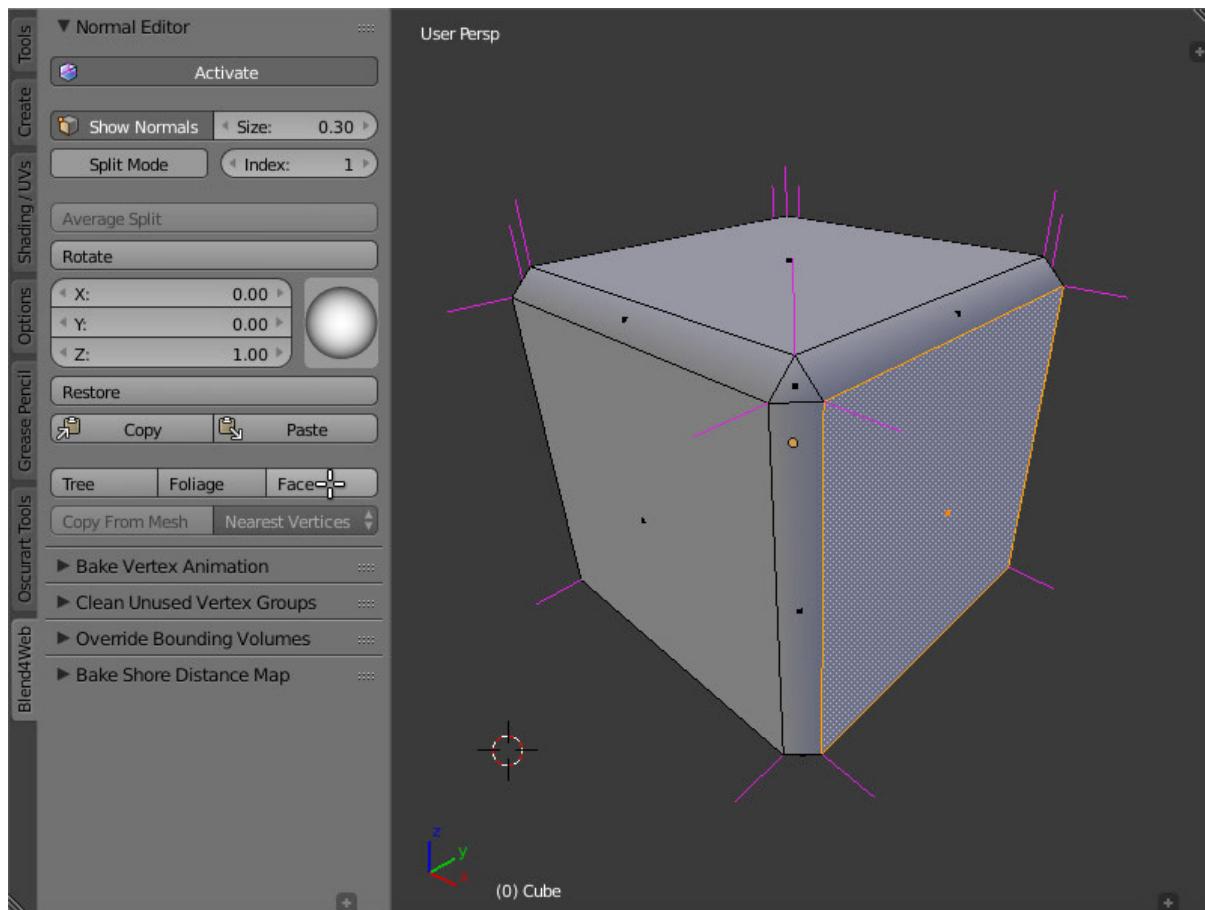


Для того, чтобы восстановить нормали к исходному направлению (которое рассчитывается на основе нормалей полигонов), нужно выделить нужные вертексы и нажать кнопку **Restore**.

## Tree, Foliage и Face

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.





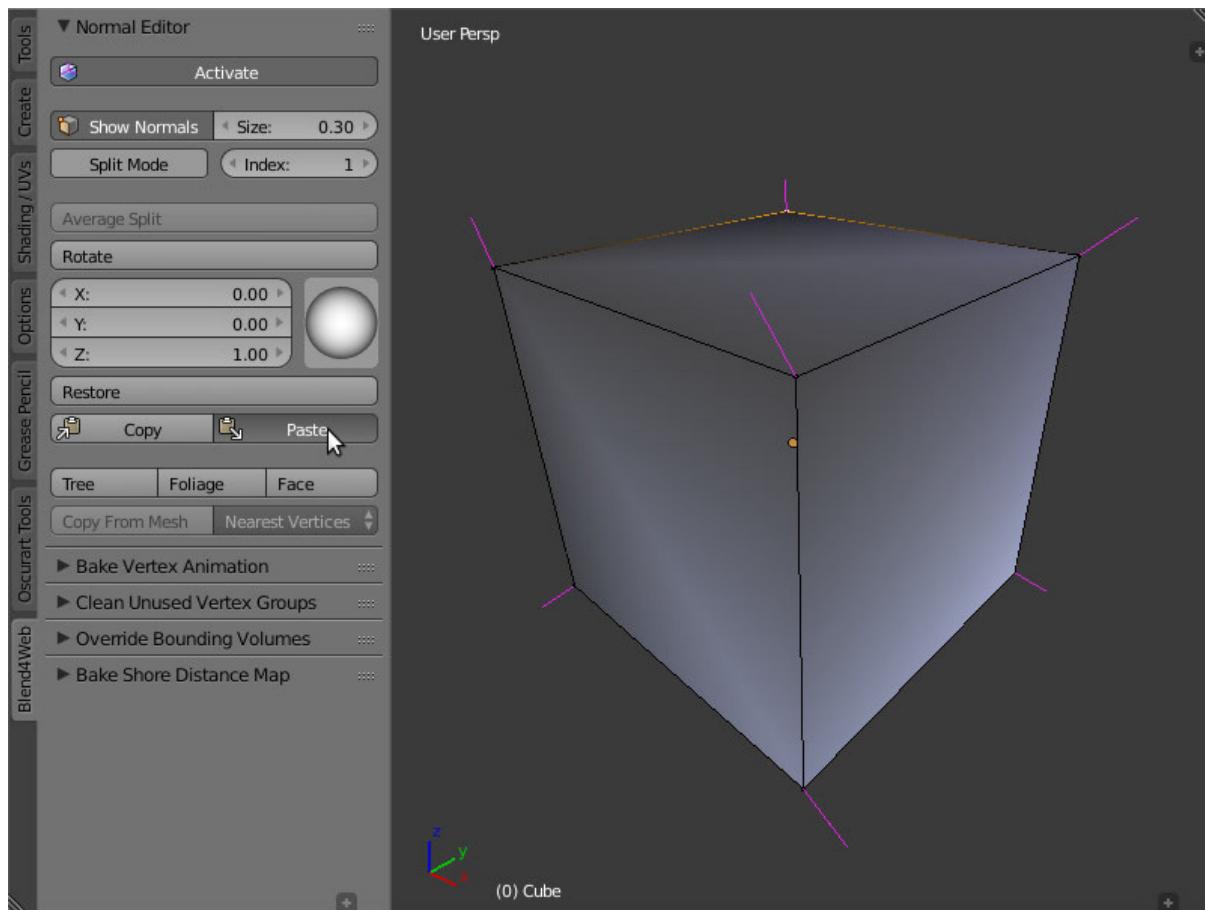
Чтобы воспользоваться функцией **Tree**, выделите нужные вертексы и расположите курсор в нужном вам месте. После нажатия кнопки **Tree** все выделенные вертексы будут направлены в сторону от курсора, как будто бы они выстреливают из одной точки.

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.

## Copy/Paste

Копирует направление нормали с одного вертекса на другой.

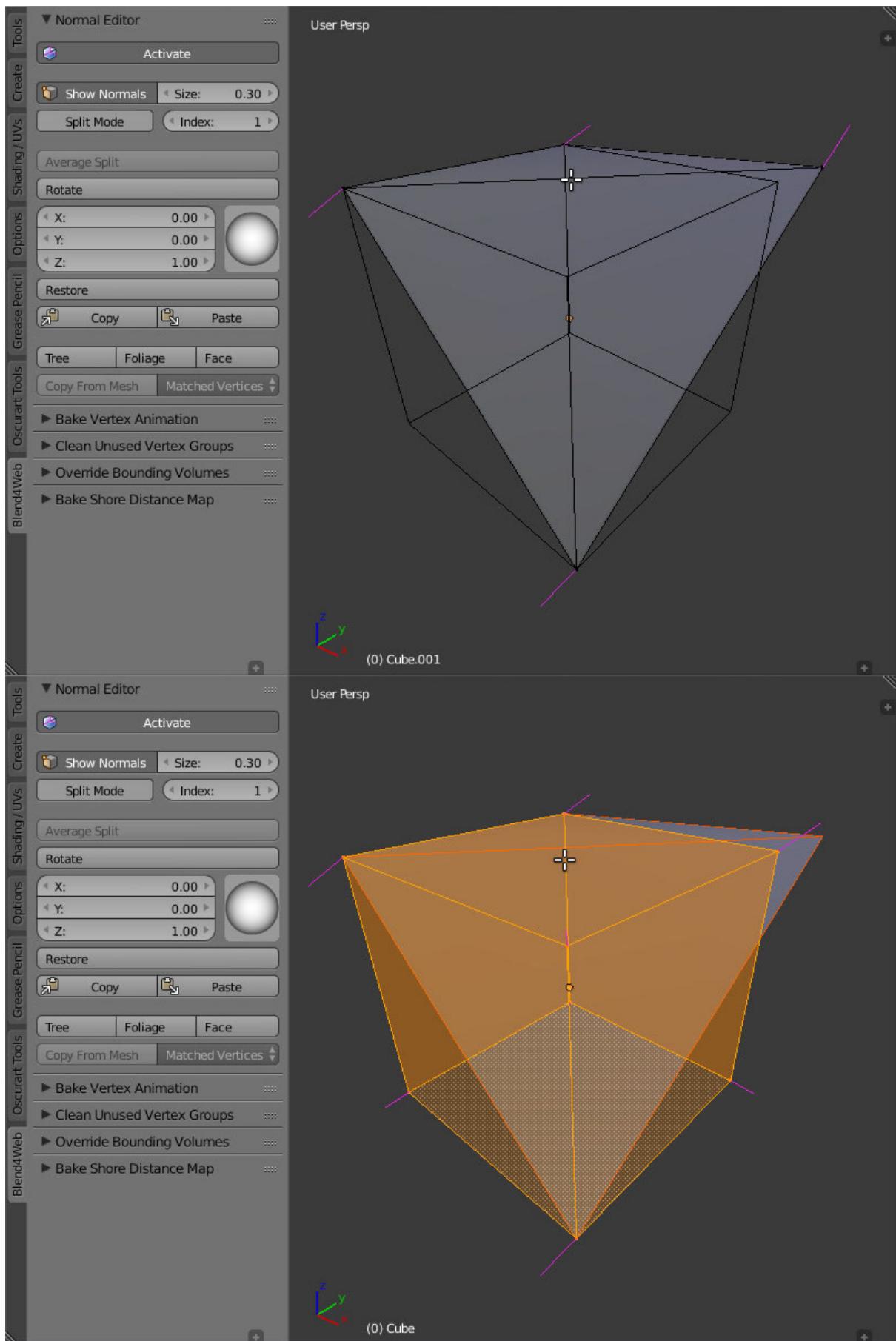


Выделите вертекс для копирования и нажмите **Copy**. Затем выделите вертекс, на который хотите копировать результат, и нажмите **Paste**. Вставлять скопированную информацию можно на множество разных вертексов. Кнопки не активны в режиме **Split Mode** и не переносят данные уже “разделенных” нормалей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

## Copy From Mesh

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.



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.

## Камера

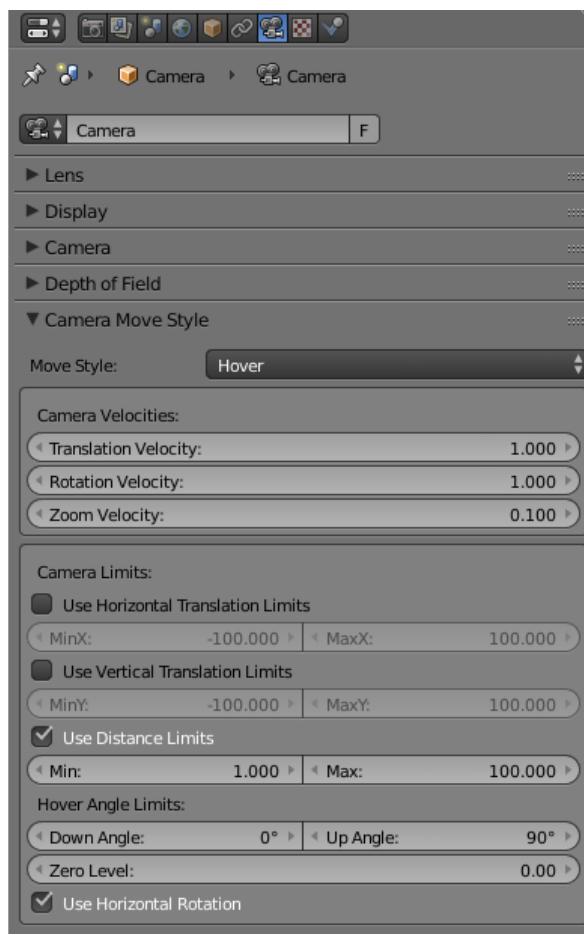
---

### Содержание

- Камера
  - Режимы управления и общие настройки
  - Ограничения движения камеры
    - \* Камера типа Target
    - \* Камера типа Eye
    - \* Камера типа Hover
  - Привязка к камере
  - Управление через API
    - \* Задание режима управления
    - \* Перемещение и поворот
    - \* Задание лимитов
    - \* Отсчет угловых координат

## 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. Дополнительными лимитами можно задать плавную траекторию движения камеры по направлению к опорной точке.
- **Static** В статическом режиме изменение положения камеры осуществляется посредством анимации или через API.

**Look At Cursor > Target Location** Доступно в режиме Target. Позиция точки, относительно которой будет вращаться камера. Кнопка Look At Cursor позволяет скопировать текущее положение курсора, а при нажатии на неё активная камера поворачивается в направлении указанной точки.

**Depth of Field** Описано в разделе *Глубина резкости камеры*.

Для камеры доступны настройки скорости движения:

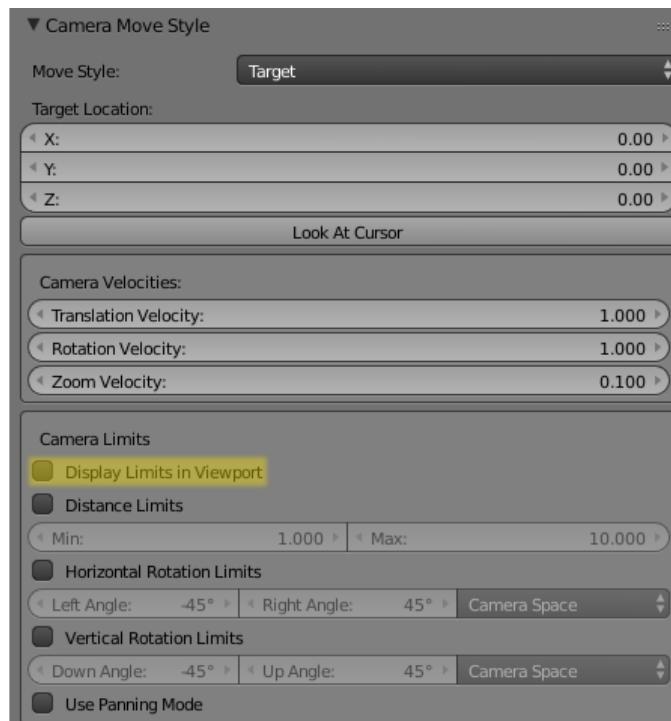
**Camera Move Style > Camera Velocities > Translation Velocity** Актуально для режимов Target, Eye, Hover. Задаёт скорость перемещения камеры. Интервал значений:  $[0, \infty)$ . Значение по умолчанию: 1.

**Camera Move Style > Camera Velocities > Rotation Velocity** Актуально для режимов Target, Eye, Hover. Задаёт скорость вращения камеры. Интервал значений:  $[0, \infty)$ . Значение по умолчанию: 1.

**Camera Move Style > Camera Velocities > Zoom Velocity** Актуально для режимов Target, Hover. Задаёт скорость приближения камеры к опорной точке. Интервал значений:  $[0, 0.99]$ . Значение по умолчанию: 0.1.

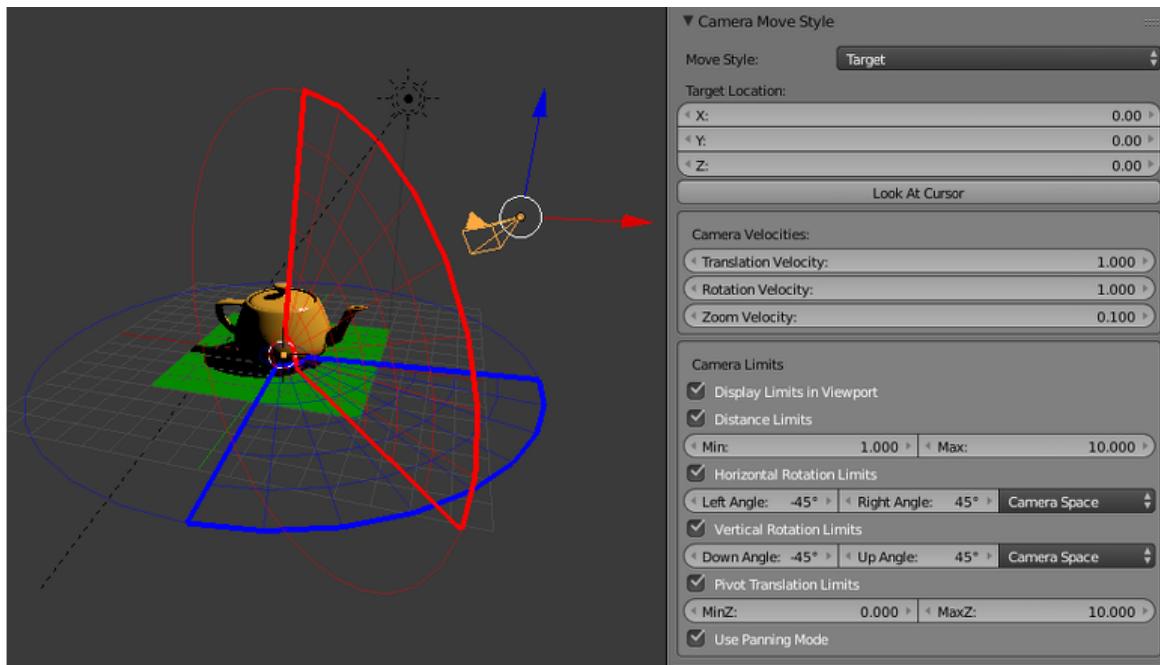
## Ограничения движения камеры

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

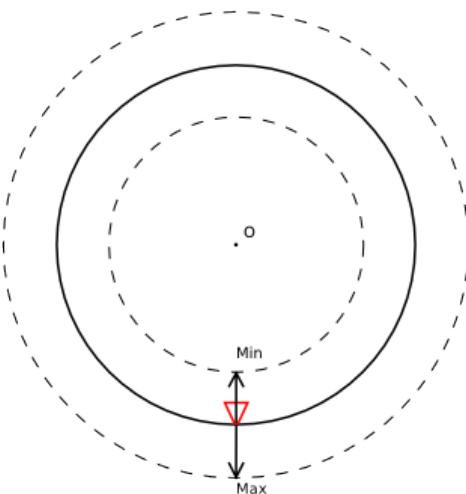


Ограничители перемещения камеры могут отображаться непосредственно во видоискателе Blender. Для этого нужно включить опцию **Display Limits in Viewport**.

## Камера типа Target

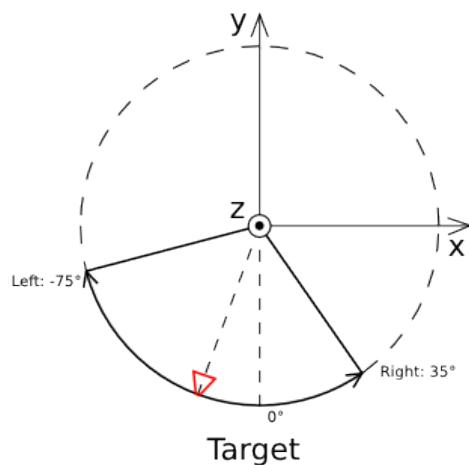


**Camera Move Style > Camera Limits > Distance Limits** Задание минимального и максимального расстояний от камеры до точки вращения. Допустимые значения:  $Min \leq Max$ . Значения по умолчанию:  $Min = 1$ ,  $Max = 10$ . По умолчанию отключено.



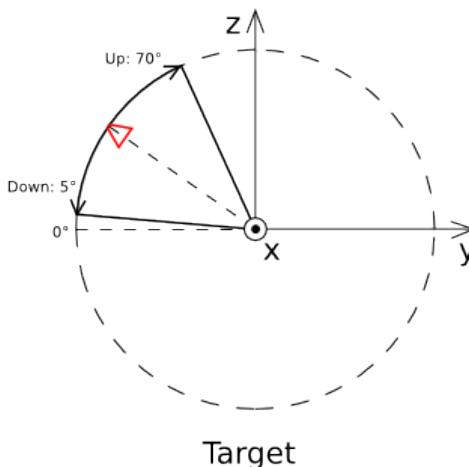
### Camera Move Style > Camera Limits > Horizontal Rotation Limits

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. Значения по умолчанию:  $Left Angle = -45^\circ$ ,  $Right Angle = 45^\circ$ . По умолчанию отключено.



#### *Camera Move Style > Camera Limits > Vertical Rotation Limits*

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. Значения по умолчанию:  $Left\ Angle = -45^\circ$ ,  $Right\ Angle = 45^\circ$ . По умолчанию отключено.

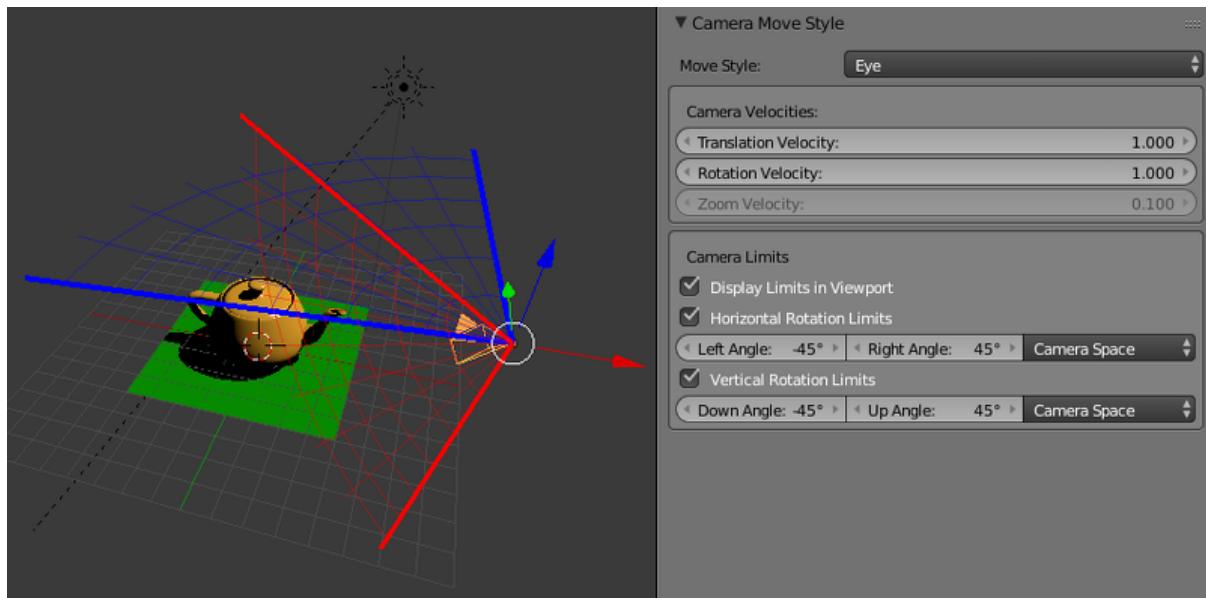


#### *Camera Move Style > Camera Limits > Pivot Translation Limits*

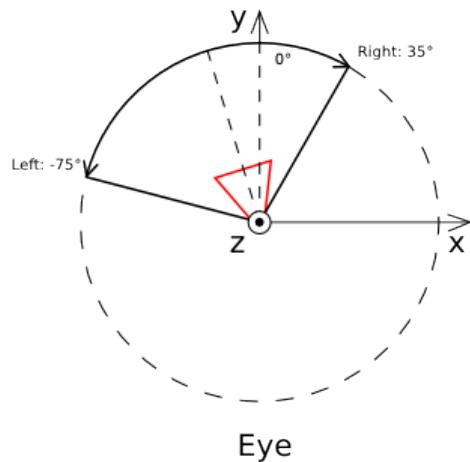
Ограничение перемещения точки привязки камеры. Значения по умолчанию:  $MinZ = 0$ ,  $MaxZ = 10$ . Не отображается во выпорте.

#### *Camera Move Style > Camera Limits > Use Panning Mode* Разрешить панорамное перемещение камеры.

## Камера типа Eye

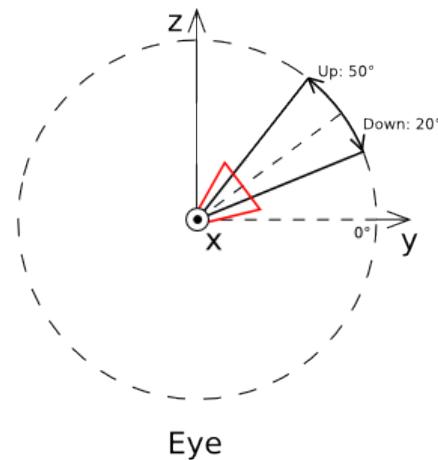


**Blend4Web > Horizontal Rotation Limits** Ограничение горизонтального (вокруг мировой оси Z в Blender'e) вращения камеры относительно соответствующей точки. Вращение происходит по дуге окружности определяемой значениями *Left Angle* и *Right Angle*. Дуга вращения соответствует движению из *Left Angle* в *Right Angle* против часовой стрелки. Значения по умолчанию: *Left Angle* =  $-45^\circ$ , *Right Angle* =  $45^\circ$ . По умолчанию отключено.

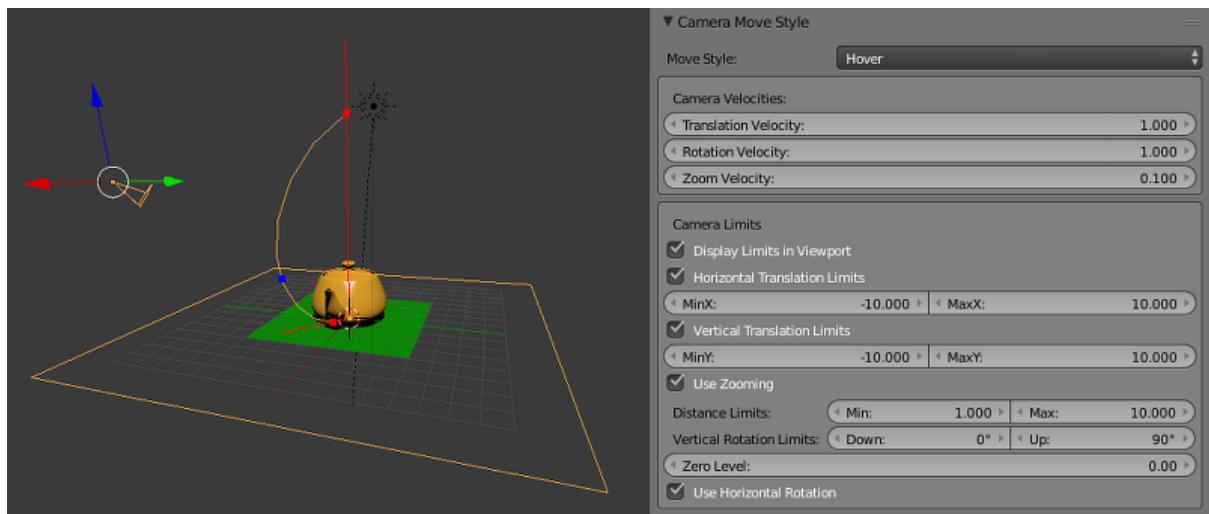


### Camera Move Style > Camera Limits > Vertical Rotation Limits

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. Значения по умолчанию: *Left Angle* =  $-45^\circ$ , *Right Angle* =  $45^\circ$ . По умолчанию отключено.



## Камера типа Hover

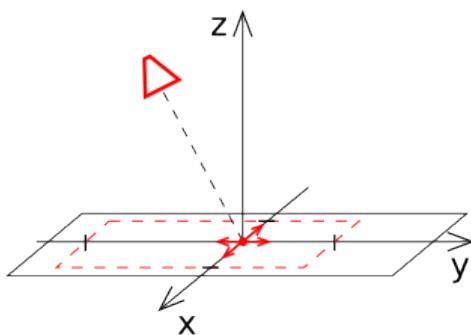


### *Camera Move Style > Camera Limits > Horizontal Translation Limits*

Ограничение перемещения опорной точки вдоль оси X в мировых координатах в Blender'e. Допустимые значения:  $Min \leq Max$ . Значения по умолчанию:  $MinX = -10$ ,  $MaxX = 10$ . По умолчанию отключено.

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

Ограничение перемещения опорной точки вдоль оси Y в мировых координатах в Blender'e. Допустимые значения:  $Min \leq Max$ . Значения по умолчанию:  $MinY = -10$ ,  $MaxY = 10$ . По умолчанию отключено.



**Camera Move Style > Camera Limits > Use Zooming** Позволяет пользователю приближать и удалять камеру относительно опорной точки. При отключении этого параметра положение камеры относительно опорной точки будет фиксированным. По умолчанию отключено.

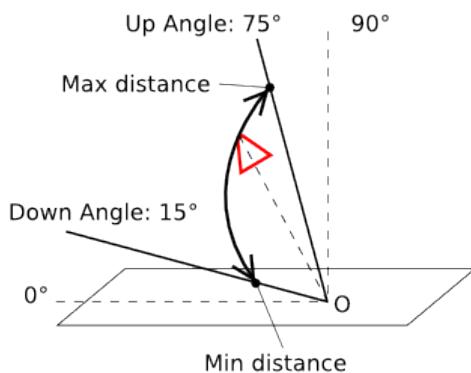
**Camera Move Style > Camera Limits > Distance Limits** Задание минимального и максимального расстояний от камеры до точки пересечения (опорная точка) направления взгляда камеры с горизонтальной опорной плоскостью (по умолчанию с плоскостью XOY мировых координат в Blender'e). Допустимые значения:  $Min \leq Max$ . Значения по умолчанию:  $Min = 1$ ,  $Max = 10$ . По умолчанию отключено.

#### **Camera Move Style > Camera Limits > Vertical Rotation Limits**

Ограничения угла подъема камеры (угла между направлением взгляда камеры и горизонтальной плоскостью). Допустимые значения:  $0^\circ \leq Down \leq Up \leq 90^\circ$ . Значения по умолчанию:  $Down = 0^\circ$ ,  $Up = 90^\circ$ .

**Camera Move Style > Camera Limits > Zero Level** Расстояние от опорной плоскости камеры до центра координат. По умолчанию равно нулю.

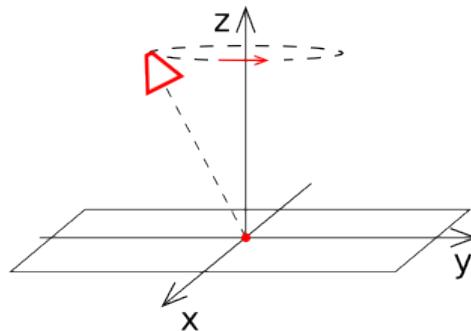
При включении настройки Blend4Web > Use Zooming лимиты по расстоянию и углу подъема будут действовать одновременно, задавая траекторию движения камеры в вертикальной плоскости.



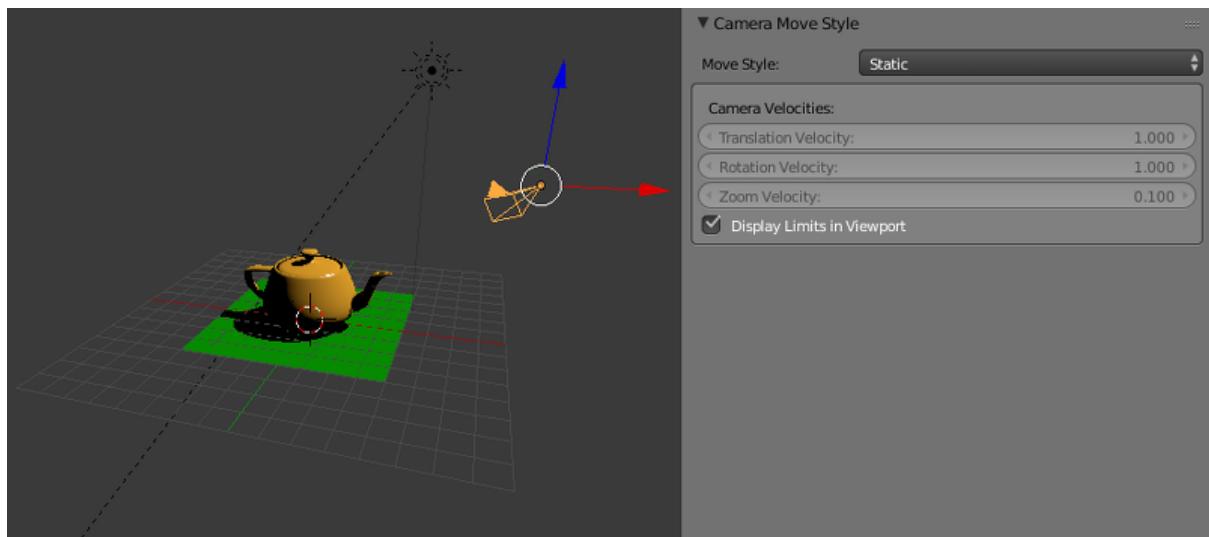
Некорректное задание лимитов по расстоянию либо углу подъема приведет к отключению этой настройки.

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

Разрешить вращение камеры в плоскости XOY Blender'a относительно опорной точки. По умолчанию включено.



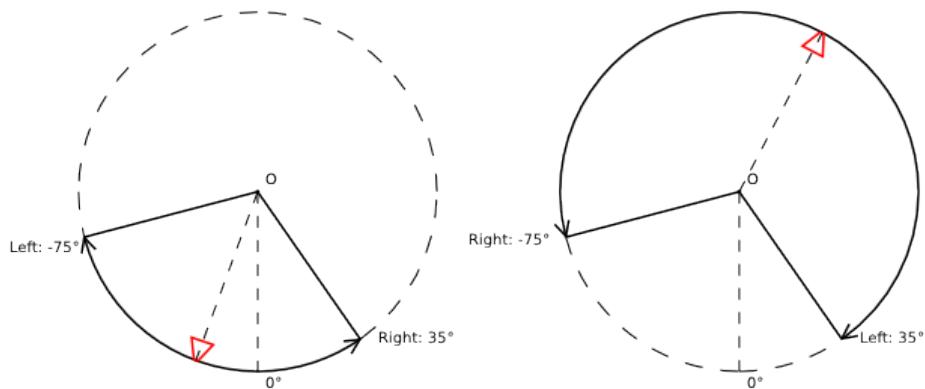
Камера типа Static



Камера этого типа не имеет ограничителей, так как не контролируется пользователем.

### Особенности задания ограничений

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



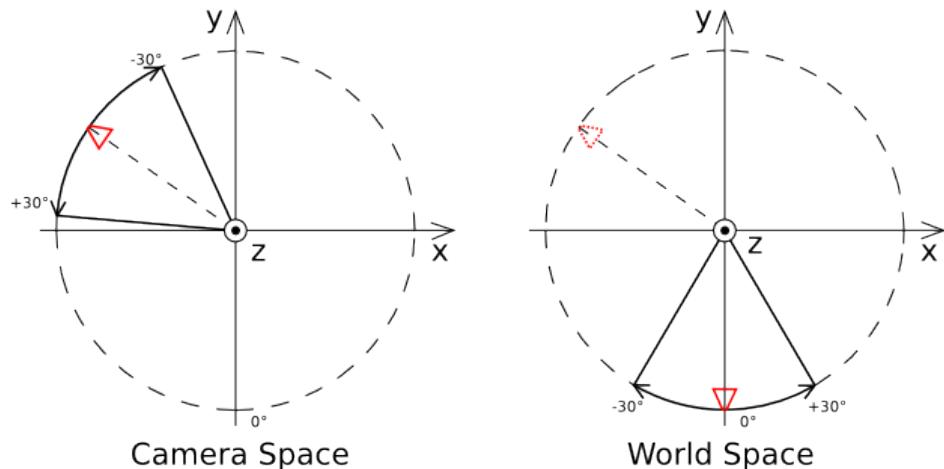
- При ограничении горизонтального и вертикального вращения камеры можно выбрать пространство координат:

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

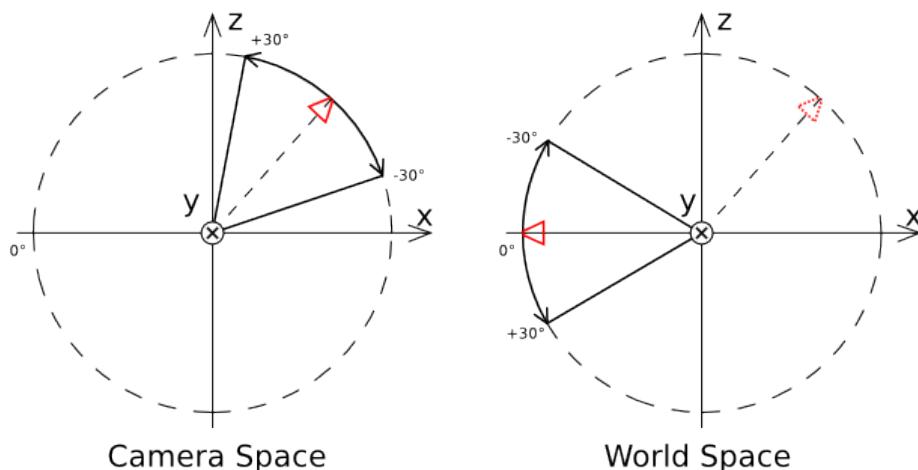
**World Space** Горизонтальные углы отсчитываются относительно направления оси Y в пространстве мировых координат, вертикальные - относительно плоскости XOY в мировом пространстве в Blender'e.

Значение по умолчанию: Camera Space.

Ограничения по горизонтали на примере TARGET камеры:

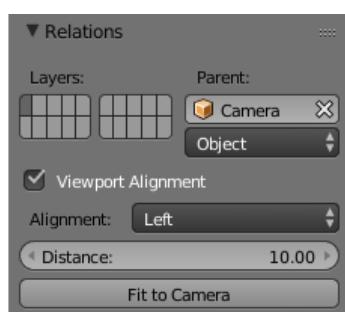


Ограничения по вертикали на примере TARGET камеры:

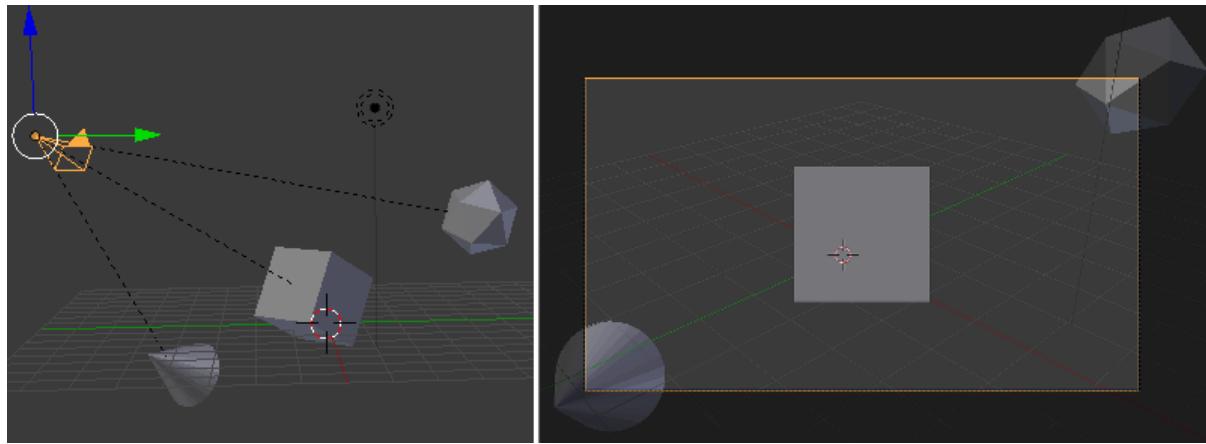


## Привязка к камере

Если родителем объекта является камера, для него доступен набор параметров, известный как *Viewport Alignment*. Эти параметры находятся во вкладке *Relations* на панели *Object*.



С их помощью объект можно выровнять относительно камеры таким образом, что он всегда будет оставаться в одной и той же точке экрана, вне зависимости от движения или поворота камеры. Может применяться, в частности, для создания элементов интерфейса.



Пример использования возможностей Viewport Alignment.

**Viewport Alignment** Этот параметр включает и выключает использование всех нижеперечисленных параметров. По умолчанию деактивирован.

**Alignment** Этот параметр определяет, к какой точке экрана будет привязан объект. Может иметь одно из следующих значений:

- Top-Left - привязывает объект к левому верхнему углу экрана
- Top - привязывает объект к верхней границе экрана
- Top-Right - привязывает объект к правому верхнему углу экрана
- Left - привязывает объект к левой стороне экрана
- Center - привязывает объект к центру экрана
- Right - привязывает объект к правой стороне экрана
- Bottom-Left - привязывает объект к левому нижнему углу экрана
- Bottom - привязывает объект к нижней границе экрана
- Bottom-Right - привязывает объект к правому нижнему углу экрана

Значение по умолчанию: **Center**.

**Distance** Расстояние между камерой и объектом, привязанным к ней. По умолчанию равно 10.

**Fit to Camera** Нажатие на эту кнопку расположит объект относительно камеры так, как он будет располагаться в движке. Может использоваться для предварительного просмотра результатов привязки.

## Управление через API

Подробная документация API модуля: [ссылка](#).

Положение и ориентация камеры в пространстве определяются конкретным режимом управления. В режимах EYE, TARGET и HOVER модель поведения накладывает ряд ограничений, например, таких как выравнивание вертикальной оси камеры вдоль мировой оси Y и постоянная ориентация камеры на опорную точку. Для STATIC камеры таких ограничений нет, поэтому она лучше подходит в случаях, когда нужен более полный контроль, например, при процедурной анимации.

Основные функции управления камерой находятся в модуле `camera`. Часть из них, относящаяся к конкретному режиму управления, начинается с соответствующих префиксов: `static_`, `eye_`, `target_` и `hover_`. Остальные методы применимы для всех режимов.

### Задание режима управления

Для изменения режима и полного определения модели поведения камеры можно воспользоваться следующими методами: `static_setup()`, `eye_setup()`, `target_setup()` и `hover_setup()`. Они принимают объект, содержащий набор optionalных параметров, позволяющих задать положение камеры, поворот, наличие лимитов и т.д:

```
var camera = m_scenes.get_active_camera();
var POS = new Float32Array([1,1,1]);
var LOOK_AT = new Float32Array([0,0,0]);
var EYE_HORIZ_LIMITS = { left: Math.PI/4, right: -Math.PI/4 };
var EYE_VERT_LIMITS = { down: -Math.PI/4, up: Math.PI/4 };
var TARGET_DIST_LIMITS = { min: 1, max: 10 };
var HOVER_DIST_LIMITS = { min: 1, max: 10 };
var HOVER_ANGLE_LIMITS = { down: 0, up: -Math.PI/4 };
var HOVER_HORIZ_TRANS_LIMITS = { min: -5, max: 3 };
var HOVER_VERT_TRANS_LIMITS = { min: -1, max: 1 };

...

// setup STATIC camera by defining the new position and the new look-at point
m_cam.static_setup(camera, { pos: POS, look_at: LOOK_AT });

// setup STATIC camera by defining the new look-at point and keeping the existing position
m_cam.static_setup(camera, { look_at: LOOK_AT });

// setup STATIC camera by defining the new position and keeping the existing orientation
m_cam.static_setup(camera, { pos: POS });

// setup EYE camera with horizontal rotation limits
m_cam.eye_setup(camera, { pos: POS, look_at: LOOK_AT,
    horiz_rot_lim: EYE_HORIZ_LIMITS });
```

```
// setup EYE camera with vertical rotation limits
m_cam.eye_setup(camera, { pos: POS, look_at: LOOK_AT,
    vert_rot_lim: EYE_VERT_LIMITS });

// setup TARGET camera with distance limits and panning mode
m_cam.target_setup(camera, { pos: POS, pivot: LOOK_AT,
    dist_lim: TARGET_DIST_LIMITS, use_panning: true });

// setup HOVER camera on a fixed distance (without zooming) relatively to its pivot
m_cam.hover_setup(camera, { pos: POS, pivot: LOOK_AT });

// setup HOVER camera with zooming (distance + angle limits)
m_cam.hover_setup(camera, { pos: POS, pivot: LOOK_AT,
    dist_lim: HOVER_DIST_LIMITS, hover_angle_lim: HOVER_ANGLE_LIMITS });

// setup HOVER camera with translation limits
m_cam.hover_setup(camera, { pos: POS, pivot: LOOK_AT,
    horiz_trans_lim: HOVER_HORIZ_TRANS_LIMITS,
    vert_trans_lim: HOVER_VERT_TRANS_LIMITS });

// setup HOVER camera with horizontal rotation enabled
m_cam.hover_setup(camera, { pos: POS, pivot: LOOK_AT, enable_horiz_rot: true });
```

Особенность камеры типа HOVER заключается в том, что лимиты, ограничивающие расстояние и угол подъёма, задают определённую *траекторию* движения по направлению к опорной точке. Камера всегда будет находиться на данной траектории, поэтому её итоговая позиция может отличаться от той, что была передана в методе `hover_setup()` (по сути происходит её проекция на траекторию). Если необходимо, чтобы заданное положение камеры сразу находилось на некой траектории, то можно воспользоваться методом `hover_setup_rel()` для относительного задания лимитов.

```
var camera = m_scenes.get_active_camera();
var POS = new Float32Array([1,1,1]);
var PIVOT = new Float32Array([0,0,0]);

...
// setup HOVER camera with maintaining the given camera position and
// ability to zoom-in and zoom-out equally
m_cam.hover_setup_rel(camera, { pos: POS, pivot: PIVOT, dist_interval: 2,
    angle_interval: Math.PI/4 });
```

Проверить текущий режим камеры можно, используя методы `is_static_camera()`, `is_eye_camera()`, `is_target_camera()`, `is_hover_camera()` или `get_move_style()`:

```
var camera = m_scenes.get_active_camera();

...
if (m_cam.is_static_camera(camera))
    console.log("STATIC camera!");
```

```
if (m_cam.get_move_style(camera) == m_cam.MS_EYE_CONTROLS)
    console.log("EYE camera!");
```

**Примечание:** Если требуется изменить лишь некоторые аспекты поведения камеры, то стоит воспользоваться отдельными методами, описанными ниже.

## Перемещение и поворот

Для поворота STATIC камеры следует использовать методы `static_get_rotation()` и `static_set_rotation()` - в них поворот описывается кватернионом:

```
var camera = m_scenes.get_active_camera();

...
// rotation through a quaternion
var _quat_tmp = new Float32Array(4);
var old_quat = m_cam.static_get_rotation(camera, _quat_tmp);
var new_quat = m_quat.rotateX(old_quat, Math.PI/2, old_quat)
m_cam.static_set_rotation(camera, new_quat);
```

Для EYE, TARGET и HOVER поворот осуществляется в *сферических координатах* с использованием методов `eye_rotate()`, `target_rotate()`, `hover_rotate()` и `rotate_camera()`:

```
var camera = m_scenes.get_active_camera();

...
// rotate by given delta angles
m_cam.eye_rotate(camera, Math.PI/6, Math.PI/2);

// set absolute rotation in world space
m_cam.eye_rotate(camera, Math.PI/6, Math.PI/2, true, true);
```

Текущие сферические координаты камеры можно получить, используя метод `get_camera_angles()`:

```
var camera = m_scenes.get_active_camera();
var _vec2_tmp = new Float32Array(2);

...
// get camera orientation in spherical coordinates
var angles = m_cam.get_camera_angles(camera, _vec2_tmp);
phi = angles[0];
theta = angles[1];
```

Доступ к позиции камеры предоставляют методы `get_translation()` и `set_translation()`. При этом для режимов TARGET и HOVER перемещение означает параллельный перенос всей модели, включающей позицию камеры и опорную точку.

```
var camera = m_scenes.get_active_camera();
var _vec3_tmp = new Float32Array(3);

...

// get camera position
var pos = m_cam.get_translation(camera, _vec3_tmp);

// set new position
var new_pos = m_vec3.set(1, 0, 2, pos);
m_cam.set_translation(camera, new_pos);
```

Дополнительные методы для перемещения камеры:

- `target_set_trans_pivot()`, `target_set_pivot_translation()`, `hover_set_pivot_translation()` - одновременное задание позиции и опорной точки камеры;
- `target_get_distance()`, `target_set_distance()` - смещение по расстоянию до опорной точки;
- `static_set_look_at()`, `eye_set_look_at()` - одновременное задание позиции и точки взгляда камеры;

---

**Примечание:** Т.к. камера является объектом сцены, то при необходимости можно использовать и методы модуля `transform`. Однако, результат может не всегда соответствовать ожиданиям из-за вмешательства модели поведения камеры в каждом конкретном режиме.

---

## Задание лимитов

Лимиты камеры присутствуют в режимах EYE, TARGET и HOVER. Для назначения конкретного лимита следует воспользоваться соответствующим методом:

| EYE                                      | TARGET                                      | HOVER                                       |
|--|---|---|
| <code>eye_get_horizontal_limits()</code> | <code>target_get_distance_limits()</code>   | <code>hover_get_distance_limits()</code>    |
| <code>eye_set_horizontal_limits()</code> | <code>target_set_distance_limits()</code>   | <code>hover_set_distance_limits()</code>    |
| <code>eye_get_vertical_limits()</code>   | <code>target_get_horizontal_limits()</code> | <code>hover_get_vertical_limits()</code>    |
| <code>eye_set_vertical_limits()</code>   | <code>target_set_horizontal_limits()</code> | <code>hover_set_vertical_limits()</code>    |
|  | <code>target_get_vertical_limits()</code>   | <code>hover_get_horiz_trans_limits()</code> |
|  | <code>target_set_vertical_limits()</code>   | <code>hover_set_horiz_trans_limits()</code> |
|  | <code>target_get_pivot_limits</code>        | <code>hover_get_vert_trans_limits()</code>  |
|  | <code>target_set_pivot_limits</code>        | <code>hover_set_vert_trans_limits()</code>  |

```

var camera = m_scenes.get_active_camera();
var _limits_tmp = {};
var EYE_HORIZ_LIMITS = { left: Math.PI/4, right: -Math.PI/4 };

...
// get limits
m_cam.eye_get_horizontal_limits(camera, _limits_tmp);

// set limits
m_cam.eye_set_horizontal_limits(camera, EYE_HORIZ_LIMITS);

```

Наличие лимитов можно проверить методами `has_distance_limits()`, `has_horizontal_rot_limits()`, `has_vertical_rot_limits()`, `has_horizontal_trans_limits()` и `has_vertical_trans_limits()`.

---

**Примечание:** В режиме HOVER у камеры всегда есть лимиты по дистанции и углу подъёма. В случае, если они не были заданы, лимиты автоматически рассчитываются так, чтобы зафиксировать камеру в текущем положении относительно опорной точки.

---

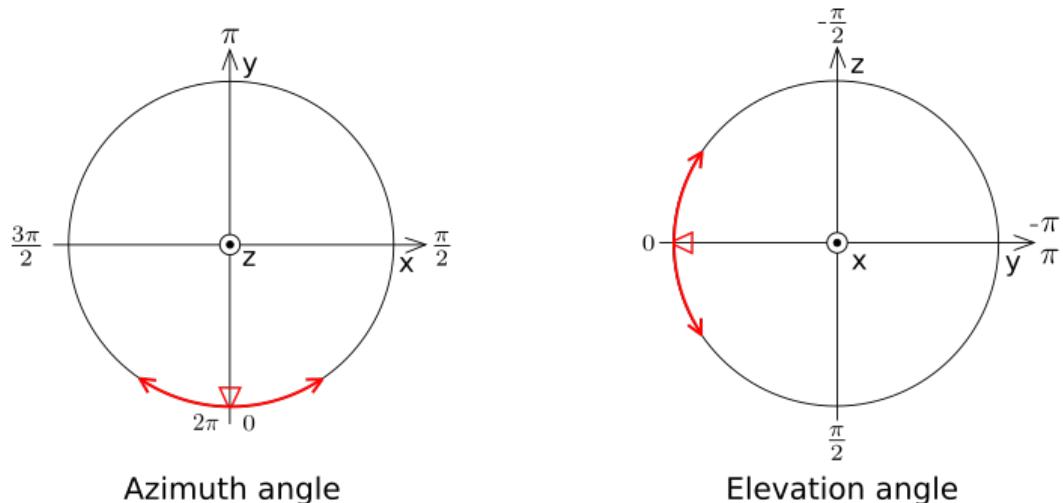
**Примечание:** Присутствующие на камере лимиты могут влиять на её позицию и ориентацию, выставляемые через API.

---

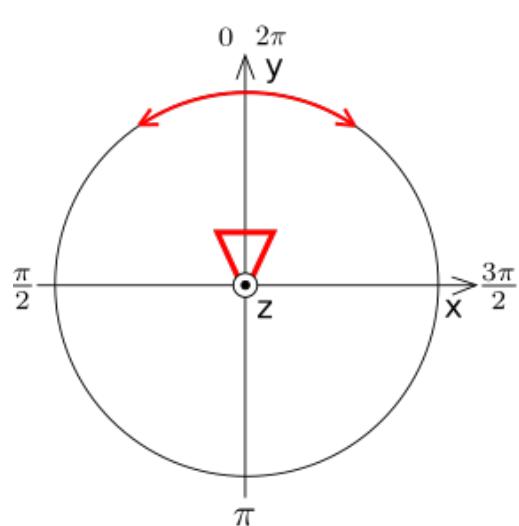
## Отсчет угловых координат

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

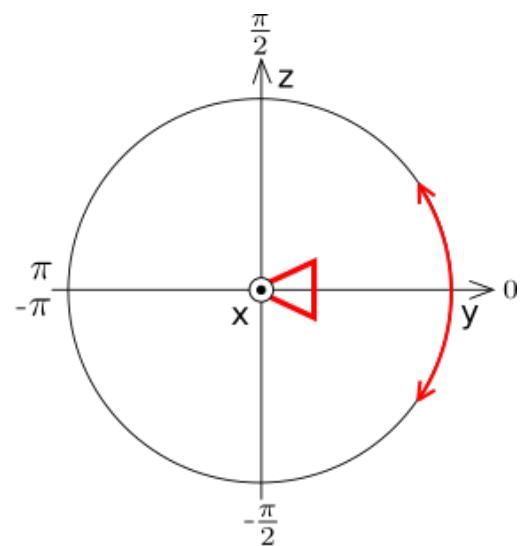
Для типов TARGET/HOVER:



Для типа EYE:



Azimuth angle



Elevation angle

## Материалы

### Содержание

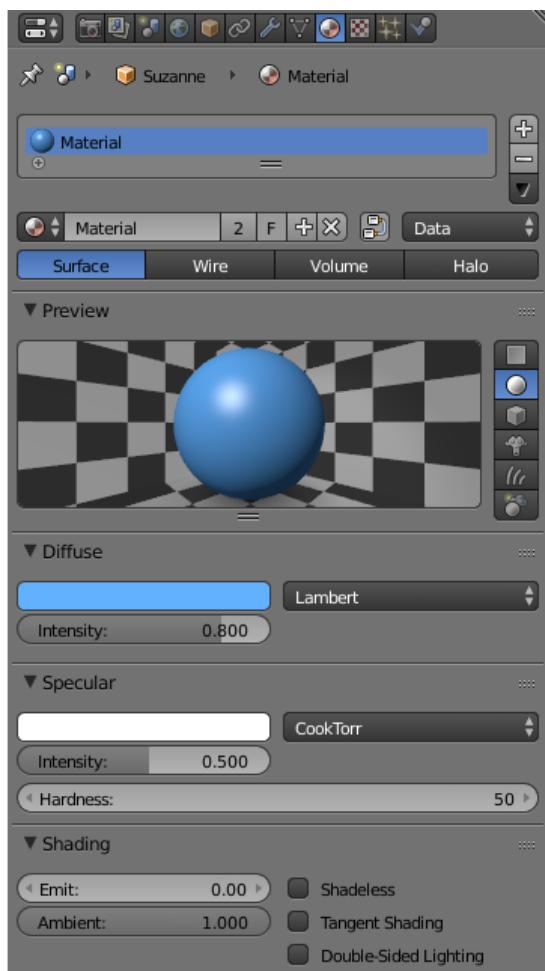
- *Материалы*
  - *Параметры освещения*
  - *Прозрачность*
    - \* *Типы*
    - \* *Дополнительные настройки*
  - *Зеркальное отражение*
    - \* *Статическое отражение*
    - \* *Динамическое отражение*
    - \* *Эффект Френеля для отражения*
  - *Опции рендеринга*
  - *Настройки вьюпорта*
  - *Специальные параметры движка*
  - *Материалы гало (Halo)*
    - \* *Активация*
    - \* *Дополнительные настройки*
  - *Методы API для работы с материалами*
    - \* *Методы для стандартных материалов*
    - \* *Методы для нововых материалов*
  - *Наследование материалов*
    - \* *Активация*

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).

## 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** Степенной 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.

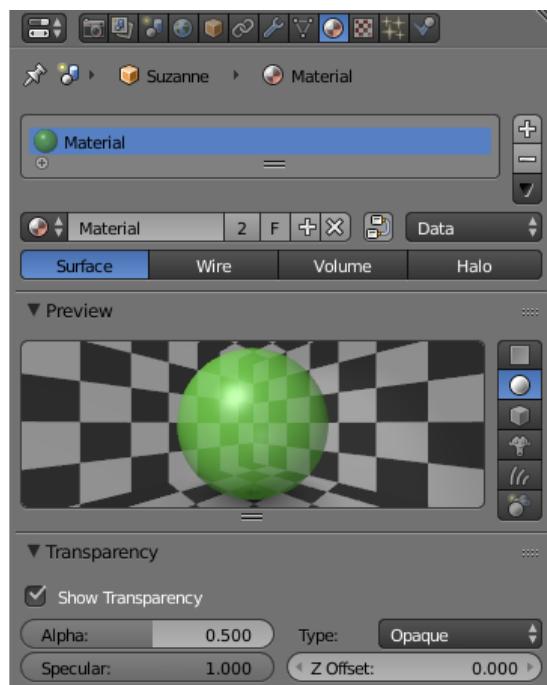
***Shading > Tangent Shading*** При включении этого параметра движок использует тангенциальный вектор материала (вместо нормального вектора) для расчёта итогового цвета объекта. Может применяться для имитации анизотропных поверхностей.



**Слева:** стандартное затенение, **справа:** тангенциальное затенение.

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

## Прозрачность



## Типы

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

Движком поддерживаются все типы реализации прозрачности (перечислены 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 (бутылка, стекло автомобиля и т.д.).

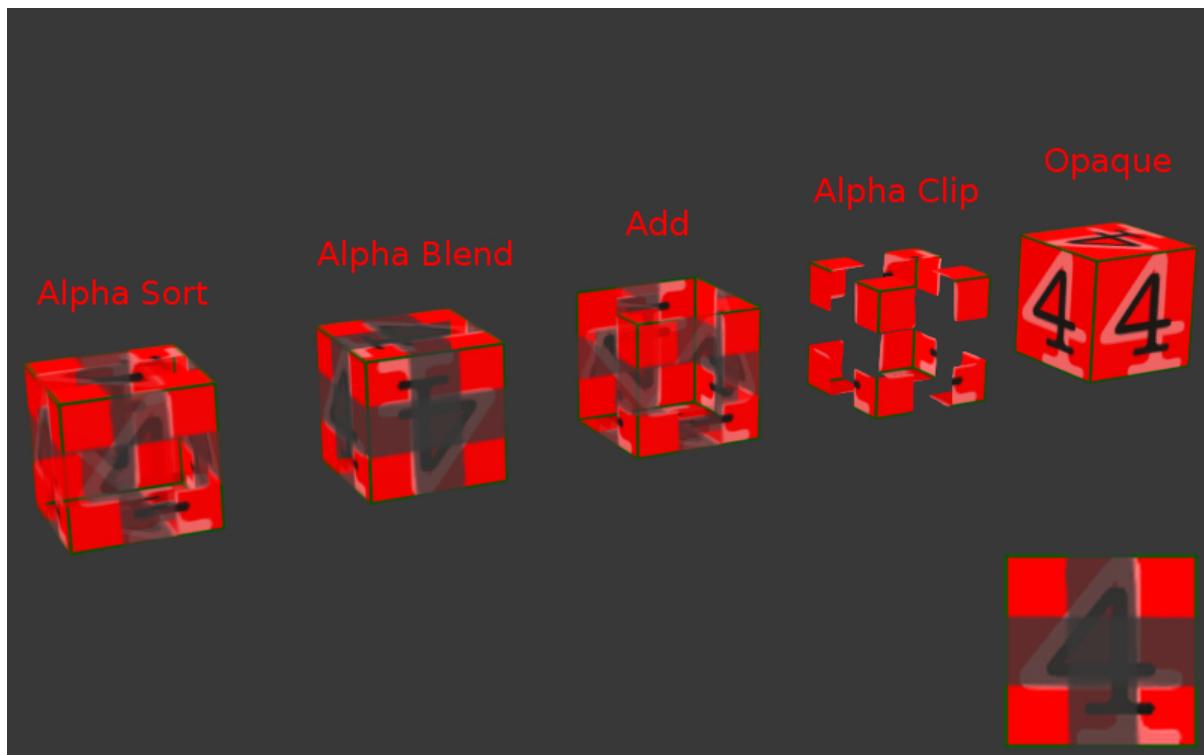
**Alpha Anti-Aliasing** Прозрачный с градиентом. Этот тип реализован при помощи технологии Alpha to coverage и доступен только при включённом MSAA (только на устройствах с поддержкой WebGL2). Сортировка треугольников не производится. Эту технику рекомендуется использовать совместно с текстурой-маской для изображения мелких деталей, таких, как трава или листья.

**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.



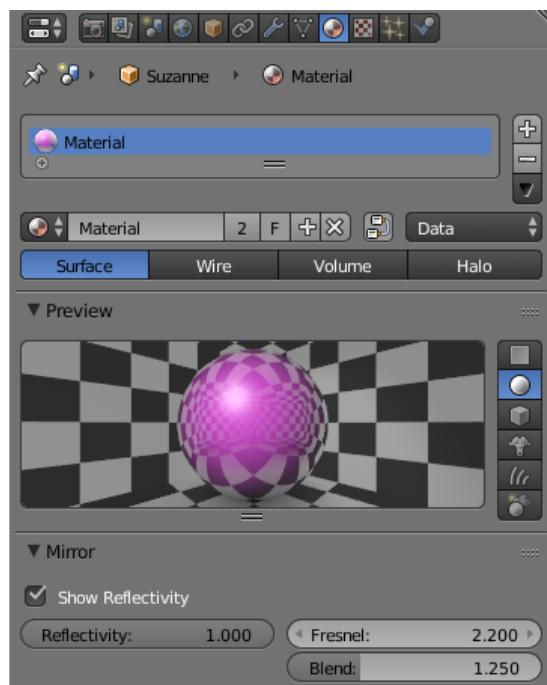
## Дополнительные настройки

**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** 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.

## Зеркальное отражение



### 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*

### Динамическое отражение

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

---

**Примечание:** При использовании *нодовых материалов* динамические отражения будут работать только в том случае, если в нодовом дереве присутствует нода Material или Extended Material.

---

### 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](#)

## 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,

$R_0$  - 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,

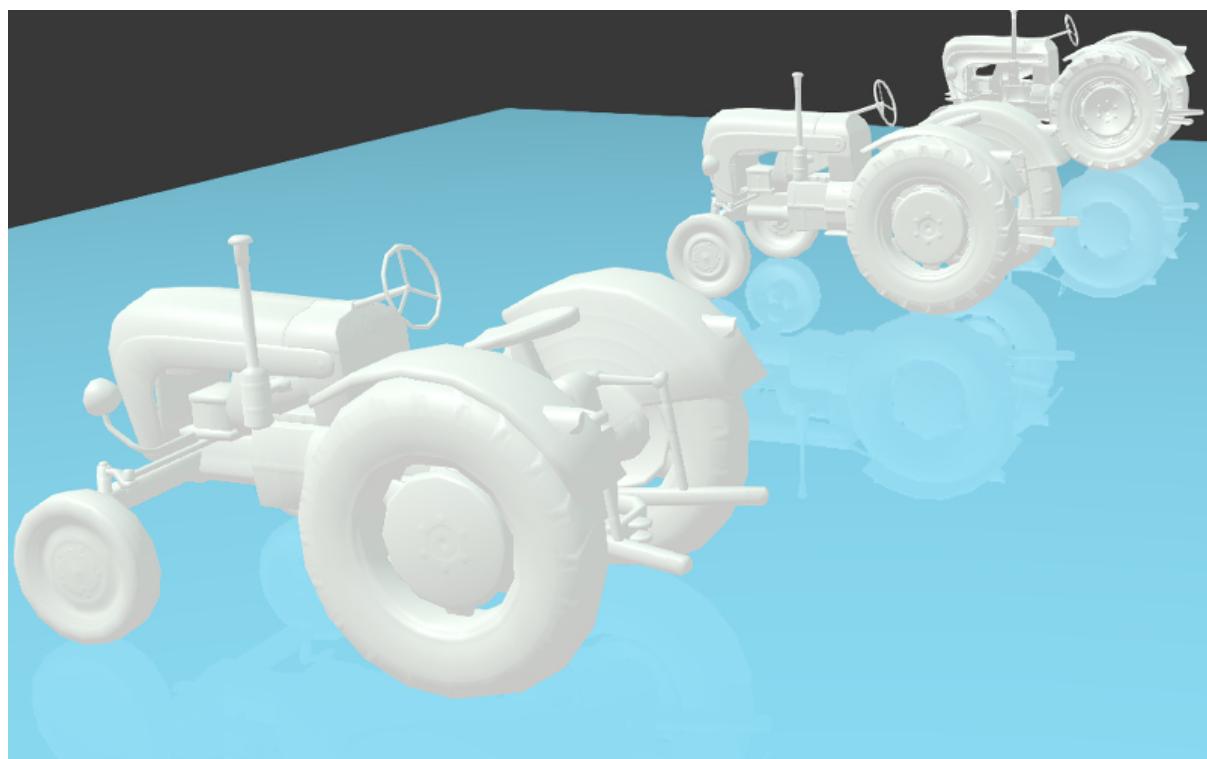
N - показатель степени.

## Настройка

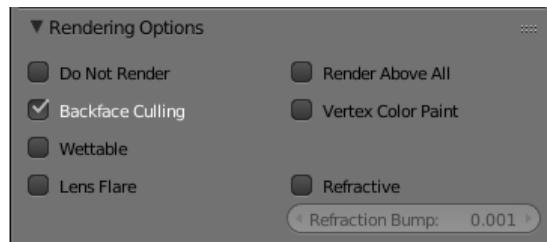
Эффект Френеля применяется как для статического, так и для динамического отражения.

**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.



## 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 > Lens Flare** Включает эффект Lens Flare для материала.

**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.

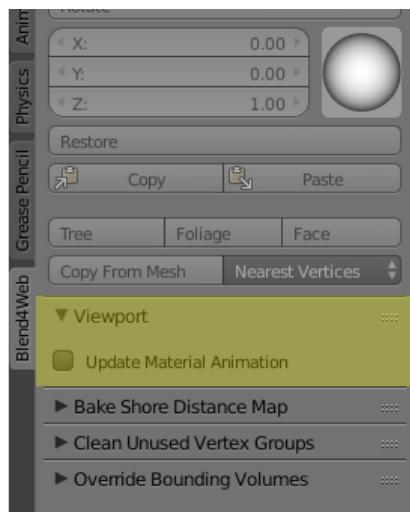
---

**См.также:**

*Прозрачность*

## Настройки выюпорта

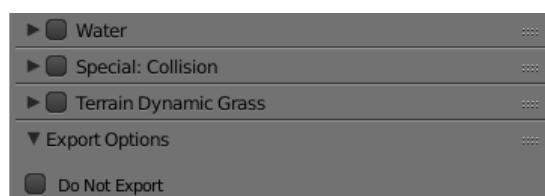
Раздел **Viewport** находится на панели Blend4Web.



### *Update Material Animation*

Обновлять анимированный шейдер во выюпорте Blender.

## 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*

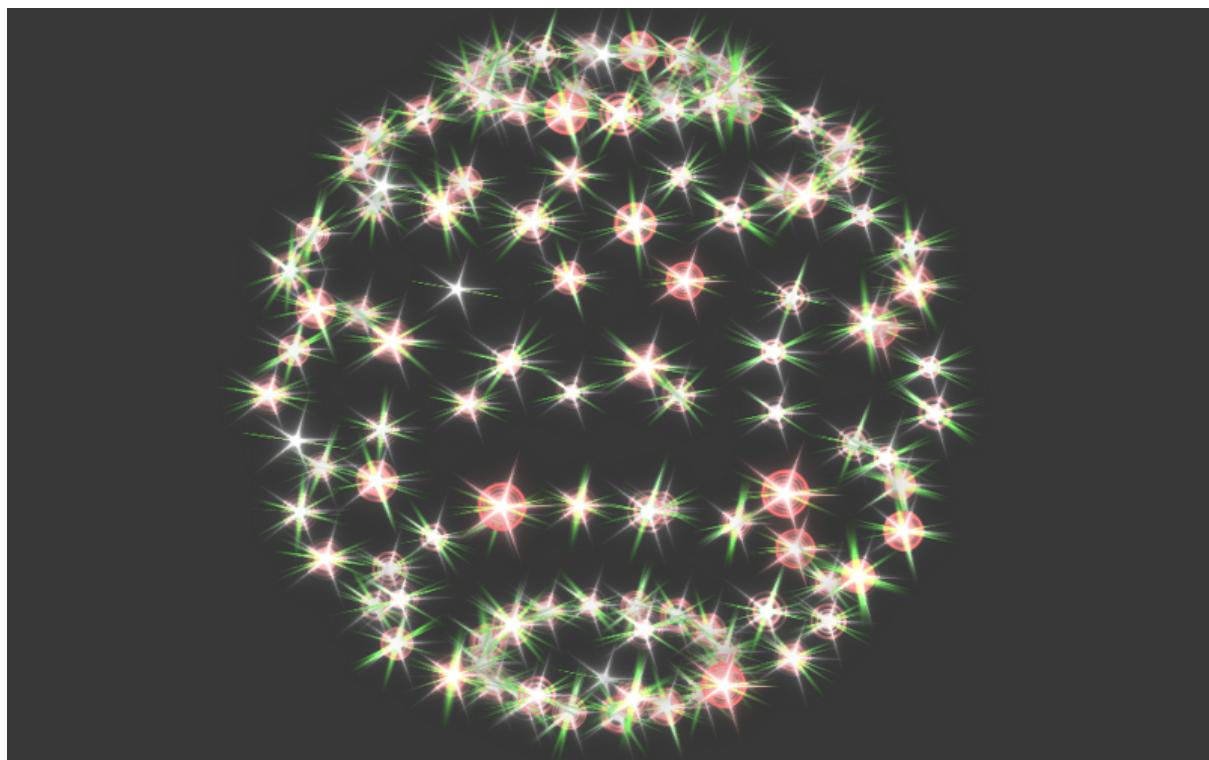
*Material > Export Options > Do Not Export* Не экспортировать.

## Материалы гало (Halo)

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

### Активация

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



### 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.

## Методы API для работы с материалами

Все методы API, используемые для настройки материалов, как стандартных, так и нодовых, находятся в модуле API `material`. Все методы этого модуля подробно описаны на [странице](#) документации API.

---

**Примечание:** Методы API могут работать только с материалами, которые используются с динамическими объектами.

---

### Методы для стандартных материалов

Модуль API `material` включает методы для управления практически любым параметром стандартного материала.

Ниже приводится ряд примеров, демонстрирующих, как модуль API `material` может использоваться для различных операций с материалом объекта:

Получение диффузного цвета объекта:

```
var m_scenes      = require("scenes");
var m_material   = require("material");

...

var cube = m_scenes.get_object_by_name("Cube");

var diffuse_color = m_material.get_diffuse_color(cube, "MyMaterial");
var diffuse_color_factor = m_material.get_diffuse_color_factor(cube, "MyMaterial");
var diffuse_intensity = m_material.get_diffuse_intensity(cube, "MyMaterial");
```

Получение цвета блика материала:

```
var m_scenes = require("scenes");
var m_material = require("material");

...

var cube = m_scenes.get_object_by_name("Cube");

var specular_color = m_material.get_specular_color(cube, "MyMaterial");
var specular_color_factor = m_material.get_specular_color_factor(cube, "MyMaterial");
var specular_hardness = m_material.get_specular_hardness(cube, "MyMaterial");
var specular_intensity = m_material.get_specular_intensity(cube, "MyMaterial");
```

Получение других параметров:

```
var m_scenes = require("scenes");
var m_material = require("material");

...

var cube = m_scenes.get_object_by_name("Cube");

var emit_factor = m_material.get_emit_factor(cube, "MyMaterial");
var alpha_factor = m_material.get_alpha_factor(cube, "MyMaterial");
var ambient_factor = m_material.get_ambient_factor(cube, "MyMaterial");

var extended_parameters = m_material.get_material_extended_params(cube, "MyMaterial");
```

Установка параметров материала:

```
var m_scenes = require("scenes");
var m_material = require("material");
var m_rgba = require("rgba");

...

var cube = m_scenes.get_object_by_name("Cube");

m_material.set_diffuse_color(cube, "MyMaterial", m_rgba.from_values(1.0, 0.0, 0.0, 1.0));
m_material.set_diffuse_color_factor(cube, "MyMaterial", 0.05);
m_material.set_material_extended_params(cube, "MyMaterial", {fresnel: 0,
                                                               fresnel_factor: 1.25,
                                                               parallax_scale: 0,
                                                               parallax_steps: "5.0",
                                                               reflect_factor: 0});
```

## Методы для нодовых материалов

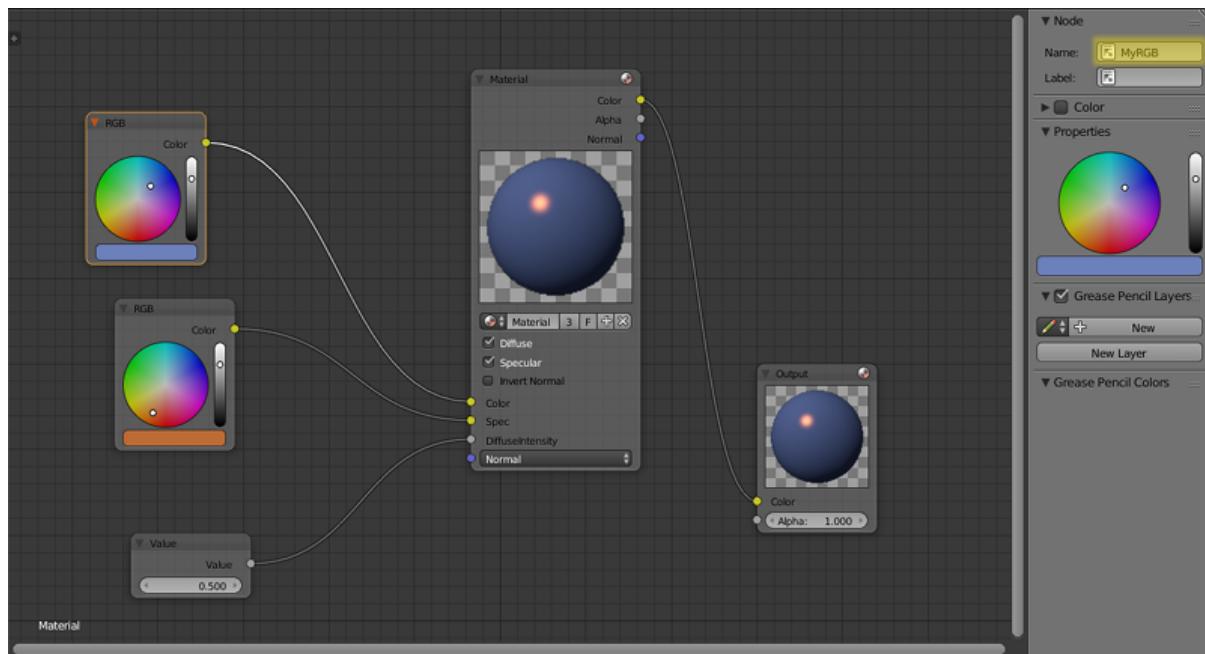
На данный момент поддерживаются только ноды типа RGB и Value. Все прочие типы не поддерживаются.

Для того, чтобы изменить значение той или иной ноды материала, требуется:

- объект, использующий данный материал,
- имя материала,
- имя самой ноды,
- а также имя группы нод, в которую входит данная нода (если есть).

Имя ноды можно узнать (и изменить) в верхней части боковой панели справа от главного окна редактора нод.

**Примечание:** Следует иметь в виду, что главное окно редактора нод не показывает имён нод, только их типы.



Два следующих примера демонстрируют, как параметры нодового материала могут быть настроены с использованием методов API:

Получение параметров нодового материала:

```
var m_scenes      = require("scenes");
var m_material   = require("material");

...

var cube = m_scene.get_object_by_name("Cube");

var rgb_node_1 = m_material.get_nodemat_rgb(cube, ["MyMaterial", "MyRGB"]);
var rgb_node_2 = m_material.get_nodemat_rgb(cube, ["MyMaterial", "MyRGB_2"]);
var value_node = m_material.get_nodemat_value(cube, ["MyMaterial", "MyValue"]);
```

Установка параметров нодового материала:

```

var m_scenes = require("scenes");
var m_material = require("material");

...

var cube = m_scene.get_object_by_name("Cube");

m_material.set_nodemat_value(cube, ["MyMaterial", "MyValue"], 0.8);
m_material.set_nodemat_rgb(cube, ["MyMaterial", "MyRGB"], 0.7, 0.9, 0.3);

```

## Наследование материалов

Blend4Web поддерживает динамическую смену материалов. Эта возможность очень полезна для т.н. приложений-конфигураторов, где пользователю предлагается настроить внешний вид какой-либо модели (одежда, мебель, автомобили и т.д.). В случае, если не хватает возможностей API по изменению стековых и нодовых материалов, описанных выше, а также метода `change_texture` и простой смены текстур, тогда можно воспользоваться наследованием.

Преимуществом наследования является простота и универсальность использования, что хорошо подходит, когда нужно довольно сильно изменить исходный материал. Также оно доступно как из внешнего API, так и в нодовой логике, и подходит одинаково как для нодовых, так и для стековых материалов. Недостатком может являться то, что операция наследования может быть хуже по производительности, чем другие вышеупомянутые методы, что может быть критически важным в ряде случаев.

### Активация

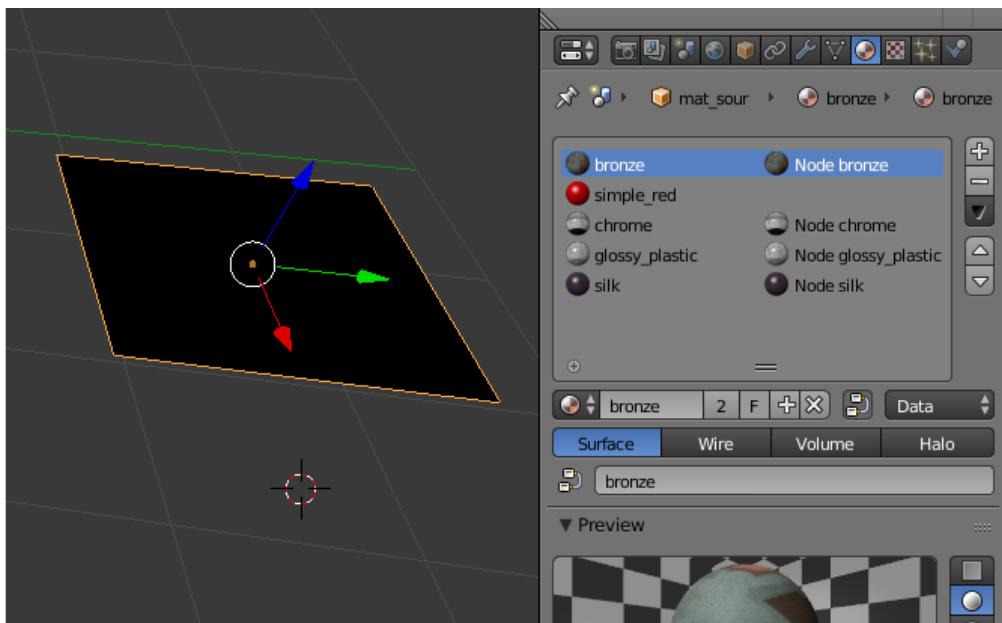
В наследовании участвуют 2 объекта. Первый - источник, на котором находится наследуемый материал, второй - целевой объект, на котором требуется произвести замену материала. Оба этих объекта требуют выставления опции `Dynamic Geometry & Materials` на панели `Object` в разделе `Rendering Properties`.

---

**Примечание:** После наследования материал на целевом объекте будет выглядеть также как, если бы он был просто назначен на него в самом Blender, т.е. при необходимости на целевом объекте нужно заранее подготовить UV-слои и слои вертексного цвета под все материалы, которые будут на него наследоваться. Для объекта источника это уже необязательно.

---

При использовании нодовой логики наследование можно осуществить с помощью ноды `Object->Inherit Material`. В API же для этого существует метод `inherit_material`.



Если в приложении требуется смена множества разных материалов, то хорошим подходом будет создание специального объекта-контейнера. Он может представлять собой простой объект типа **Plane**, на котором будут находиться все нужные материалы. При необходимости его можно скрыть на самой сцене, выставив на нём настройку **Hidden** или **Do Not Render** на панели **Object->Rendering Properties**.

## Библиотека материалов

### Содержание

- *Библиотека материалов*
  - *Структура библиотеки*
  - *Структура материала*
    - \* *Ядро материала*
    - \* *Пример использования*
  - *Добавление материала в сцену*
    - \* *Добавление целого материала*
    - \* *Добавление только ядра материала*

Библиотека материалов – это библиотека, где пользователь может найти базовые и наиболее часто используемые материалы для использования в своих проектах или для создания на их базе собственных материалов.

При создании нового проекта возможно перенести библиотеку материалов в новый проект автоматически и подключать материалы оттуда.



Для этого при создании нового проекта при помощи *менеджера проектов* надо выставить галочку *Use Material Library*. Тогда вся библиотека материалов будет автоматически скопирована в папку */assets/material\_library/* в каталоге нового проекта.

**Примечание:** Библиотека материалов может использоваться и в уже существующем проекте. Для этого достаточно скопировать файлы из папки */blender/material\_library/* в каталоге SDK в папку */assets/* в каталоге проекта. После этого файлы материалов можно подключить к файлу сцены и использовать материалы в сцене.

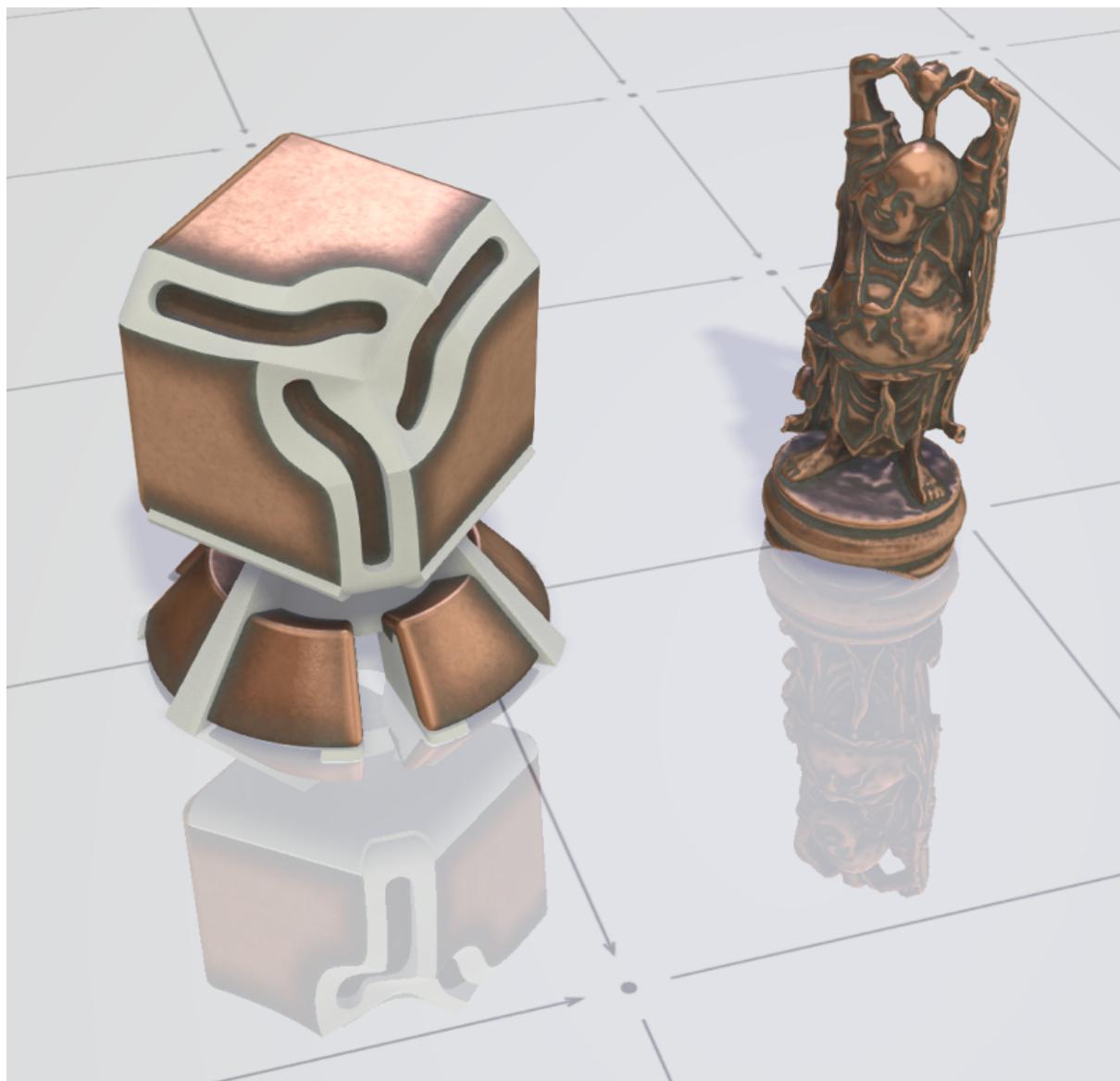
---

**Примечание:** Как и любые объекты, материалы можно добавлять к сцене с помощью команд `Link` и `Append`. Разница между ними описана [здесь](#).

---

## Структура библиотеки

Все материалы структурированы по папкам (по группам материалов) и отдельным файлам (по материалам). В каждом blend-файле находится один базовый материал (например `bronze`) и материал на дополнительном объекте (например `happy_recon_bronze`), который позволяет демонстрировать материал на характерном объекте, ассоциирующимся с данным материалом (например: материал золото — золотой слиток).



## Структура материала



1. Ядро материала
2. Нода **Material**, в которую подается вся нужная информация из ядра материала.
3. Карта нормалей, которую при необходимости можно подключить к любому ядру материала.
4. Текстура с необходимыми дополнительными масками для задания корректного поведения материала на поверхности объекта. В большинстве материалов использованы следующие маски:
  - Карта затенения
  - Карта загрязнения
  - Карта потёртостей
  - Карта патины
5. Нода **Mapping**, регулирующая параметры UV-развертки.
6. Ноды **RGB Curve**, позволяющие корректировать те или иные маски прежде чем они будут поданы в ядро материала.
7. Специфическая настройка карты спекуляра после ноды **Material**, для создания разной степени размытия блеска с использованием масок.
8. Запеченная карта затенения добавляется к шейдеру перед нодой **Output**.

Не все из них используются в каждом материале, только если требуется для создания корректного поведения того или иного шейдера.

5. Нода **Mapping**, регулирующая параметры UV-развертки.
6. Ноды **RGB Curve**, позволяющие корректировать те или иные маски прежде чем они будут поданы в ядро материала.
7. Специфическая настройка карты спекуляра после ноды **Material**, для создания разной степени размытия блеска с использованием масок.
8. Запеченная карта затенения добавляется к шейдеру перед нодой **Output**.

## 9. Нода Output

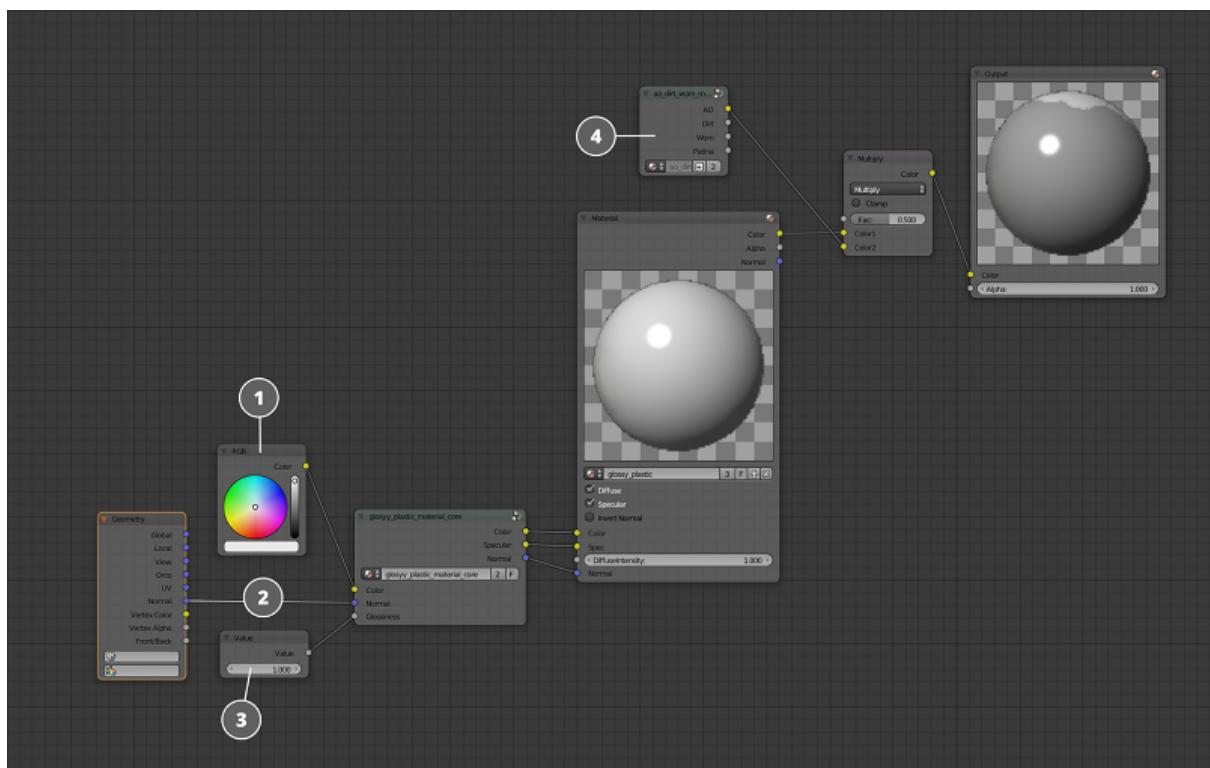
### Ядро материала

Основная нодовая группа, которая содержит нодовую структуру, формирующую шейдер. Это по сути и есть сам материал в его чистом виде, все сложные взаимодействия нодовых узлов упакованы в одну группу для удобного использования. Для её настройки нодовая группа имеет специальные входы и выходы, а так же специфические вводные параметры.

### Пример использования

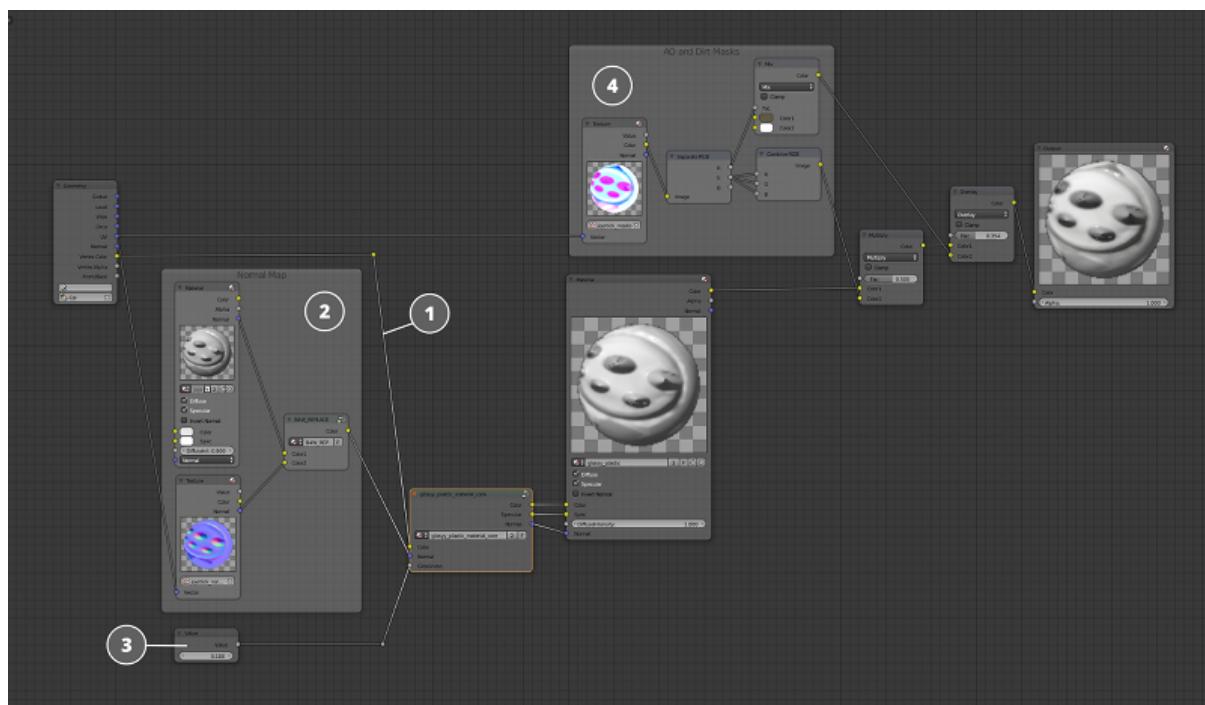


Здесь вы можете видеть пример настройки материала пластика на специальном демонстрационном объекте без использования текстур, подключаемых в ядро материала.



1. Базовый цвет задан простой нодой **RGB Color** и подан на вход в ядро материала.
2. В данной реализации материала не используется никаких карт нормалей и во вход **Normal** поданы данные нормалей с геометрии.
3. В данном ядре материала присутствует специфическая настройка, регулирующая уровень глянца на материале. В данном случае значение выставлено на максимум.
4. Также в материале используется запечатенная маска затенения.

Теперь рассмотрим использование того же материала на другом объекте - пластиковом джойстике.

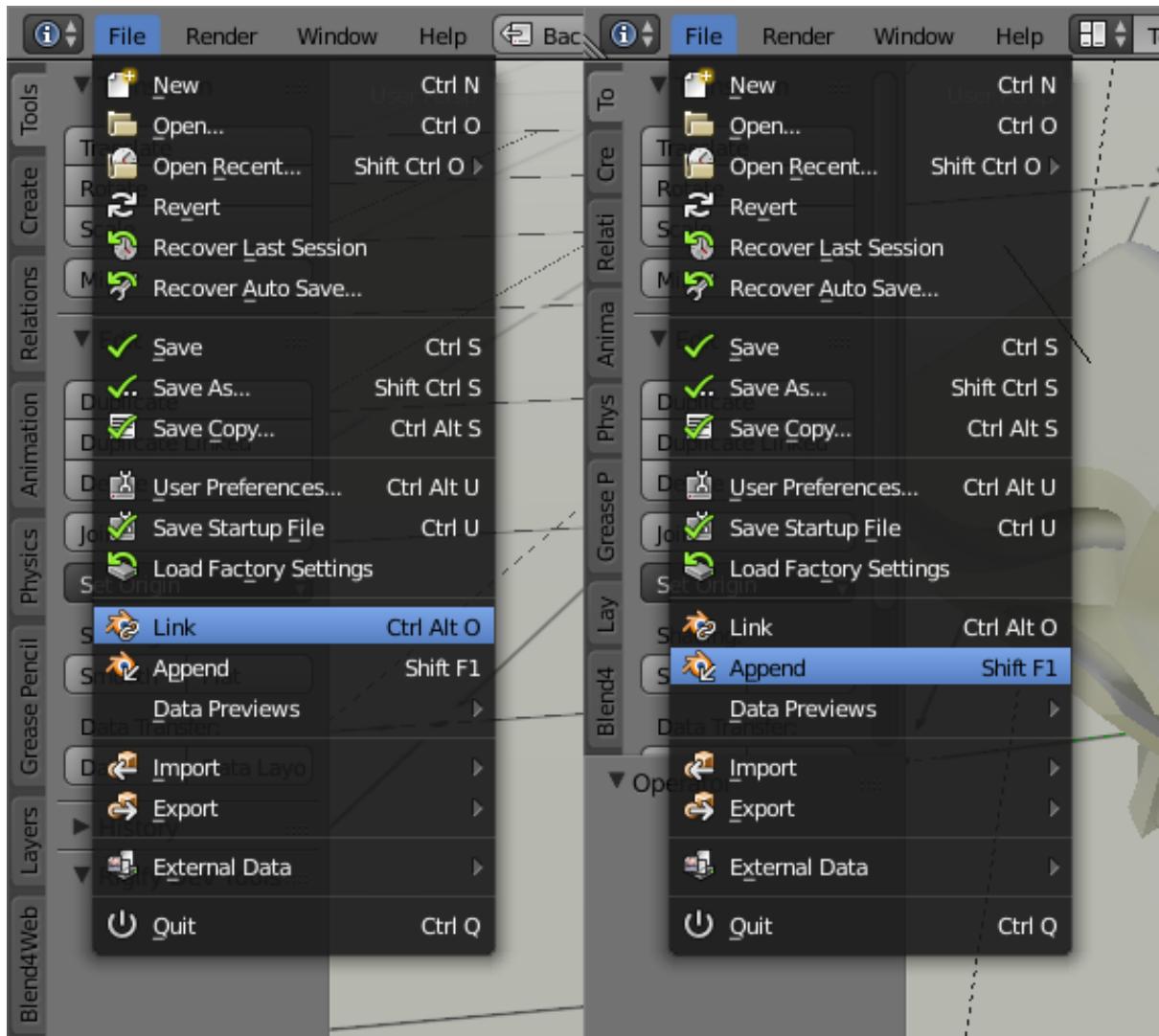


1. В качестве цвета здесь используется вертексный цвет с геометрии объекта.
2. Для геометрии джойстика была запечена карта нормалей, которая подана во вход `Normal` ядра материала.
3. Уровень глянца объекта настроен на меньшее значение.
4. Вместо масок демонстрационной модели используются маски (загрязнения и затемнения), запеченные специально для модели джойстика.

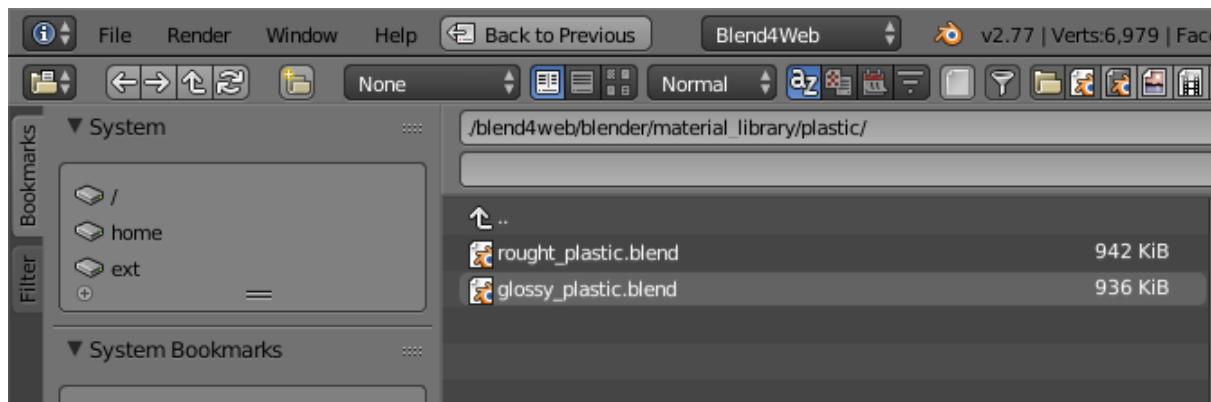
**Примечание:** Следует заметить, что содержимое ядра материала не менялось и не корректировалось под другую геометрию.

## Добавление материала в сцену

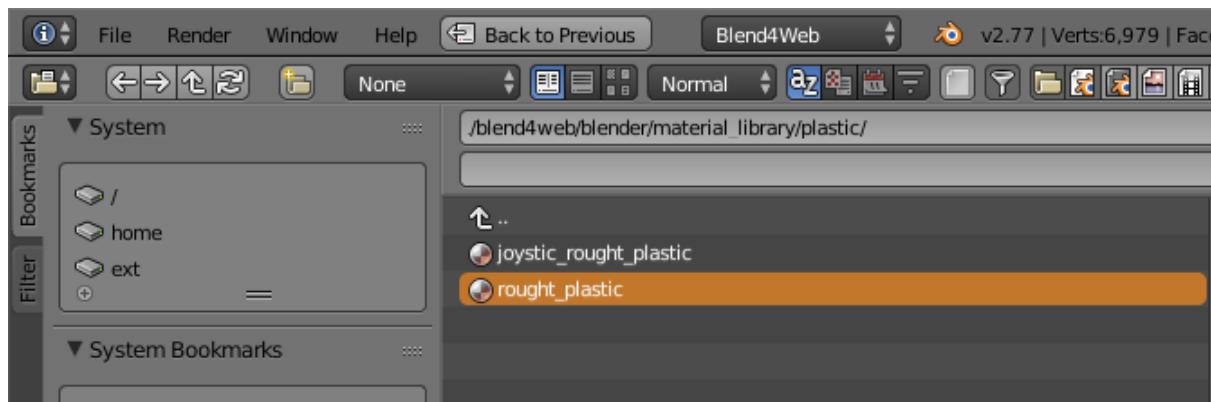
### Добавление целого материала



Выберете в меню **File** пункт **Append** или **Link**, в зависимости от того что вы хотите сделать - добавить объект полностью в ваш проект с возможностью изменять его (в этом случае выберите **Append**) или только прилинковать без необходимости изменений (в этом случае выберите **Link**).



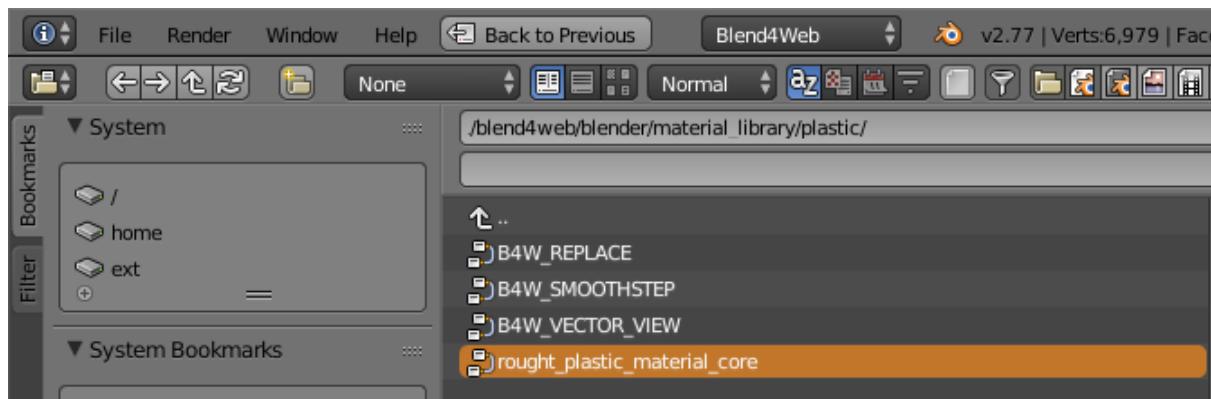
Далее выберите интересующий вас .blend-файл с материалом из директории `/blend4web/blender/material_library/`



Выберите сам материал в списке материалов и нажмите `Append from Library` или `Link from Library`. Теперь материал добавлен в вашу сцену и вы можете применить его к любому из объектов.

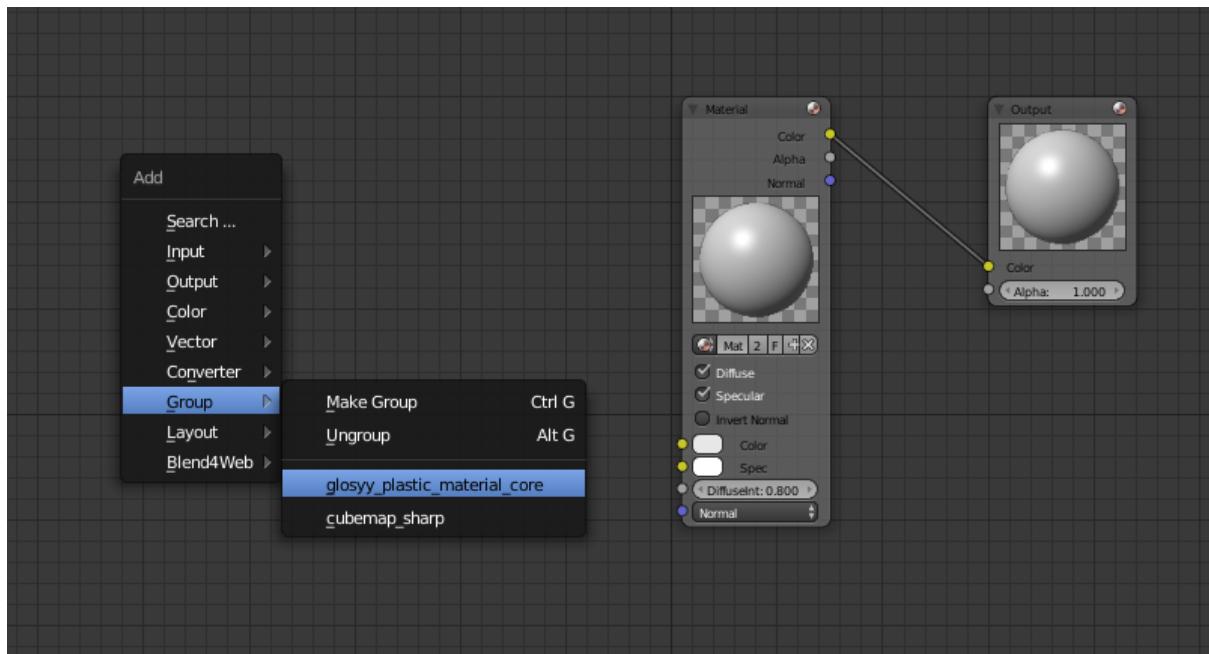
## Добавление только ядра материала

Если вы хотите добавить только ядро материала, а остальную часть настроить самостоятельно, при линковке или добавлении вместо материала выберите `NodeTree` и, выбрав интересующую вас нодовую группу, нажмите `Append from Library` или `Link from Library`.



Теперь эта нодовая группа находится в списке нодовых групп которые вы можете

добавить в свой материал через **Add=>Group** в окне Node Editor.



---

**Примечание:** У любого материала есть свои требования не только в отношении определенных масок и текстур, но и вертексных цветов и UV-разверток.

---

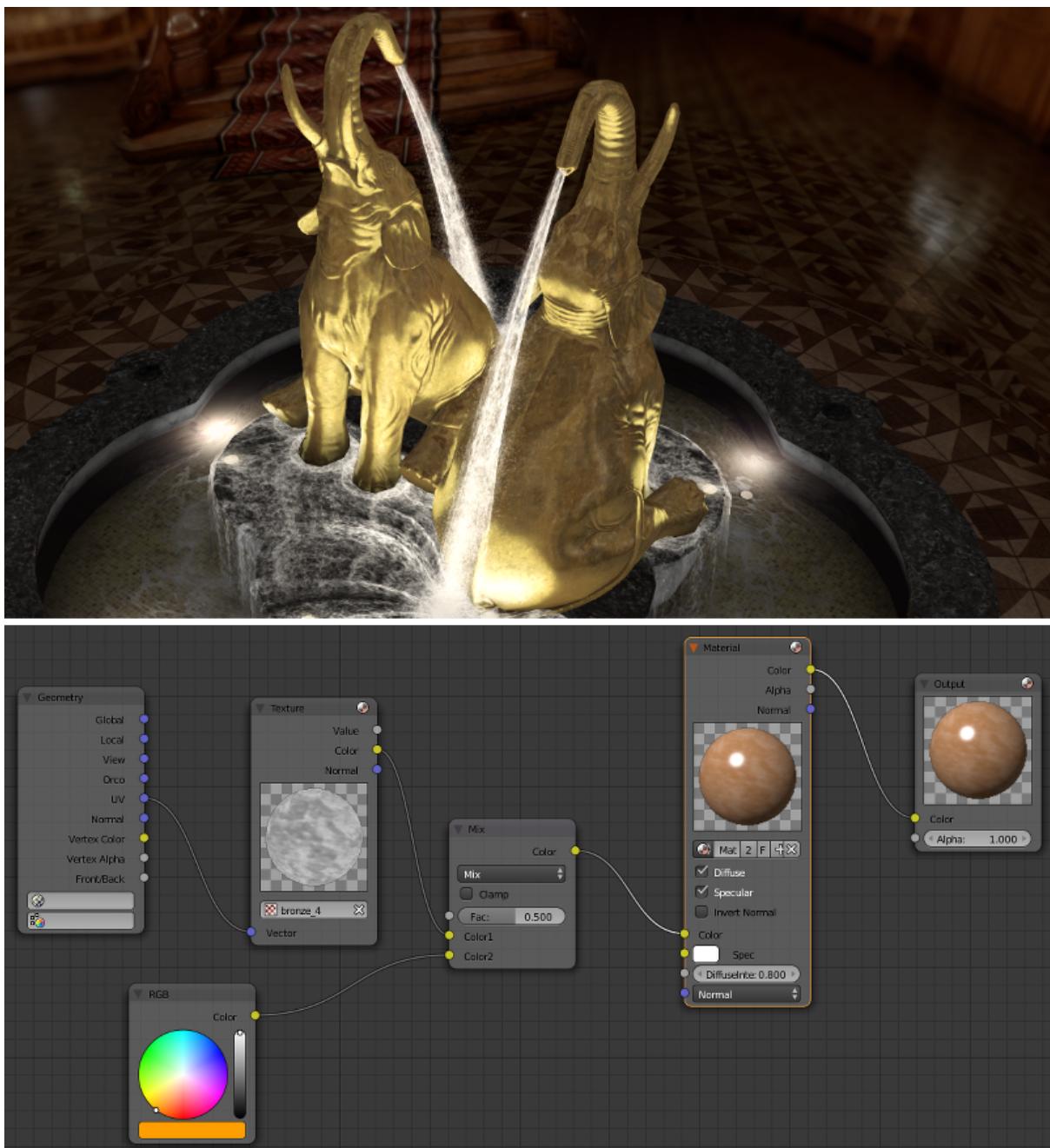
## Нодовые материалы

---

### Содержание

- *Нодовые материалы*
  - Стандартные ноды
  - Дополнительные ноды
    - \* Clamp (*B4W\_CLAMP*)
    - \* Glow Output (*B4W\_GLOW\_OUTPUT*)
    - \* Levels of Quality (*B4W\_LEVELS\_OF\_QUALITY*)
    - \* Parallax (*B4W\_PARALLAX*)
    - \* Reflect (*B4W\_REFLECT*)
    - \* Refraction (*B4W\_REFRACTION*)
    - \* Replace (*B4W\_REPLACE*)
    - \* Smoothstep (*B4W\_SMOOTHSTEP*)
    - \* Time (*B4W\_TIME*)
    - \* Translucency (*B4W\_TRANSLUCENCY*)
    - \* Vector View (*B4W\_VECTOR\_VIEW*)
    - \* Linear to SRGB и SRGB to Linear (*Deprecated*)
  - Ноды Cycles

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



## Стандартные ноды

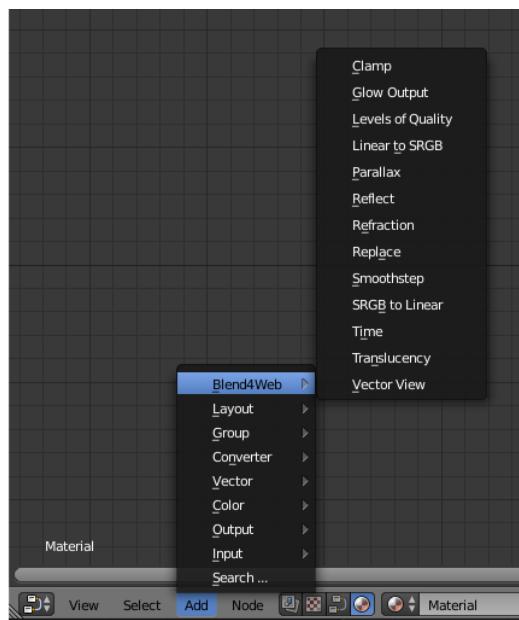
Blend4Web поддерживает все стандартные ноды Blender, однако некоторые из них работают недостаточно быстро и не рекомендуются к использованию в приложениях реального времени. Не рекомендуется создавать сложные материалы, особенно использующие большое количество нод **Geometry** и **Texture**.

Движок также частично поддерживает некоторые ноды Cycles. Более подробно этот вопрос рассмотрен в *соответствующей главе*.

Производительность и степень поддержки движком стандартных нод описаны в таблице.

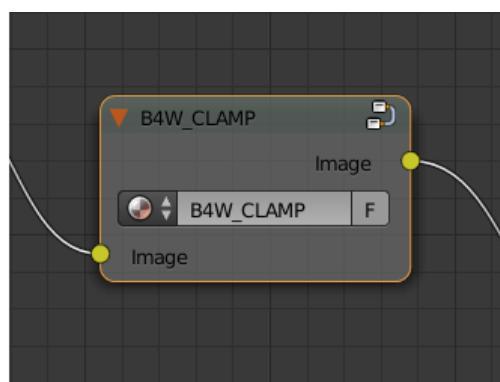
## Дополнительные ноды

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.



### Clamp (B4W\_CLAMP)

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



#### Входные параметры

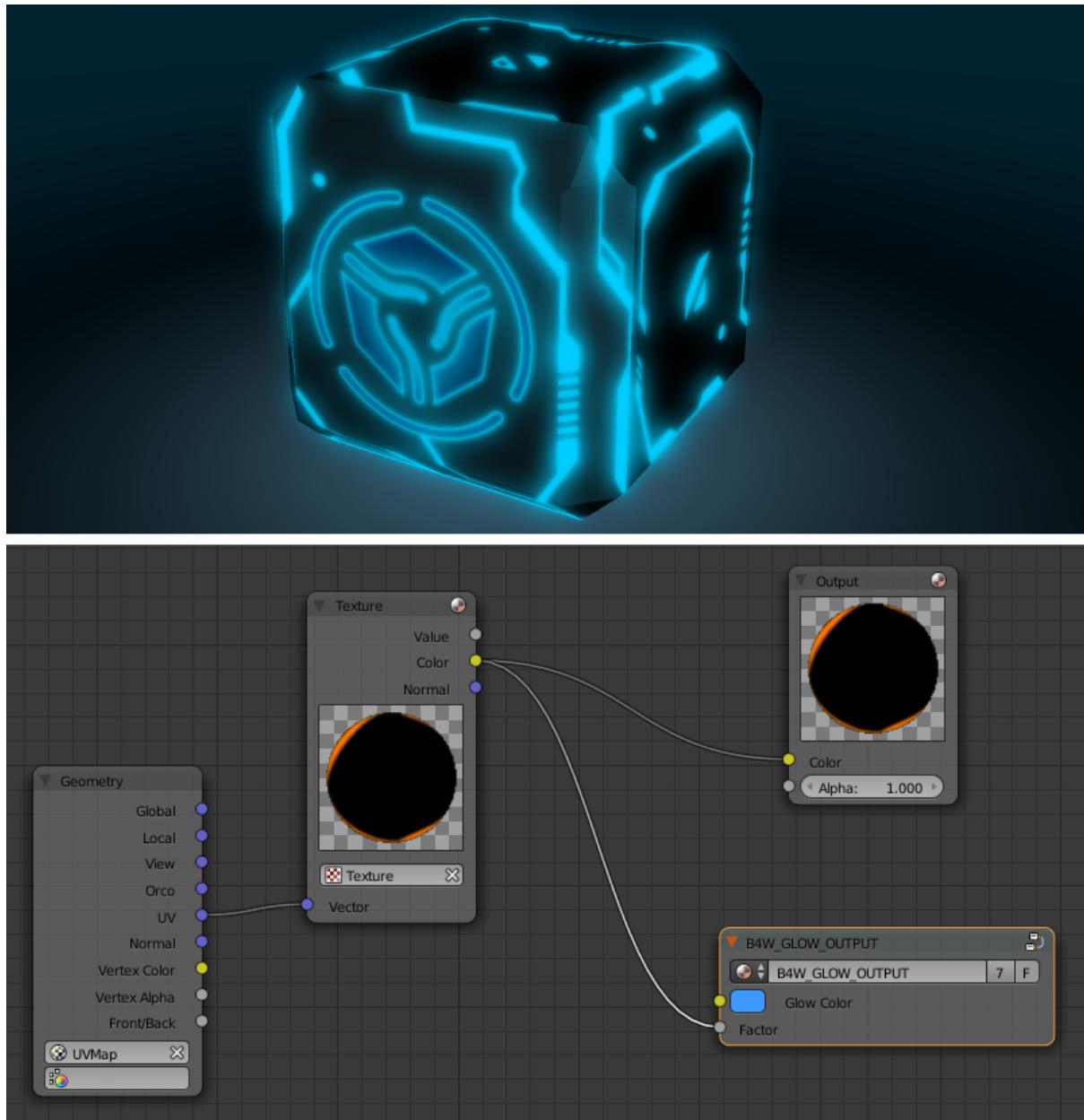
*Image* Заданный вектор.

## Выходные параметры

*Image* Вектор после ограничения.

## Glow Output (B4W\_GLOW\_OUTPUT)

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



## Входные параметры

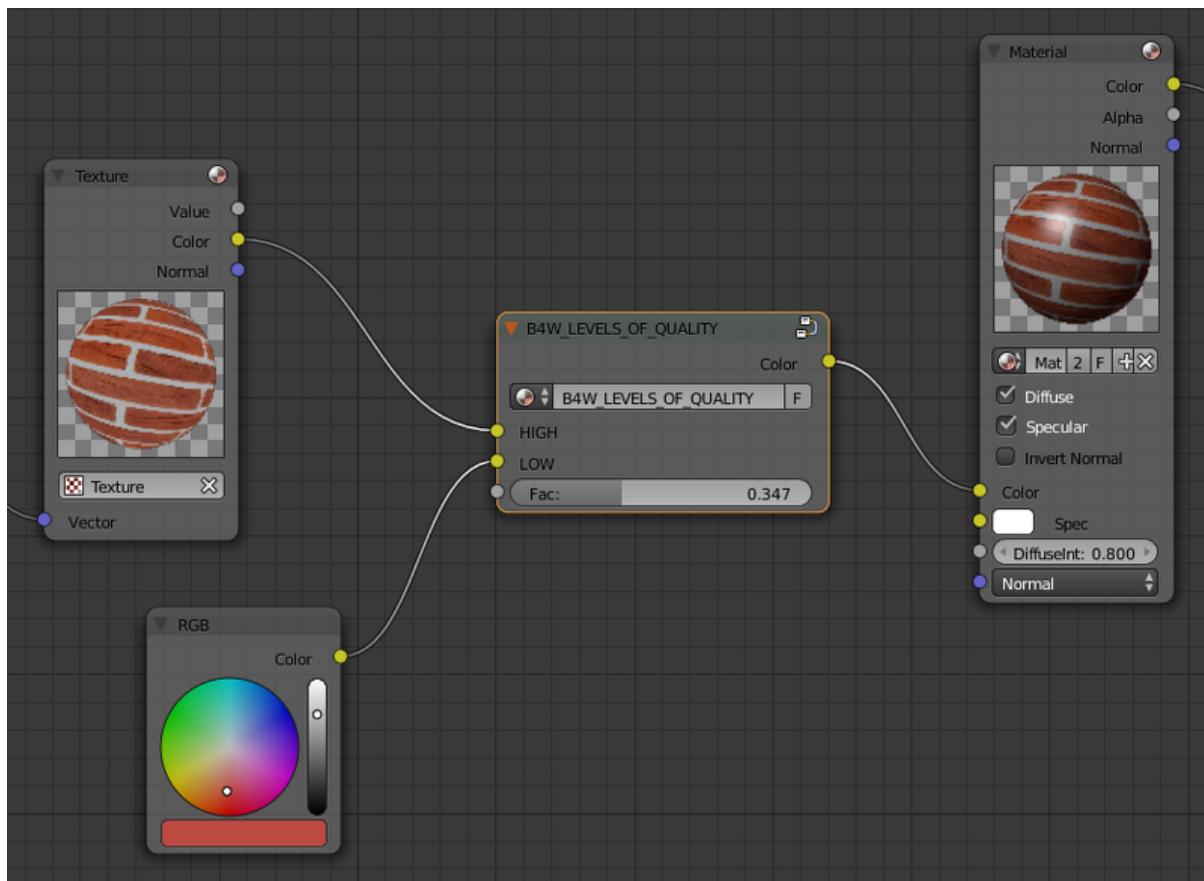
*Glow Color* Цвет свечения.

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

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

## Levels of Quality (B4W\_LEVELS\_OF\_QUALITY)

Устанавливает значение выходного цвета в зависимости от установленного профиля качества изображения. Может применяться, например, для автоматической замены сложного материала более простым при запуске приложения на мобильном устройстве.



См. также:

[Профили качества изображения](#)

### Входные параметры

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

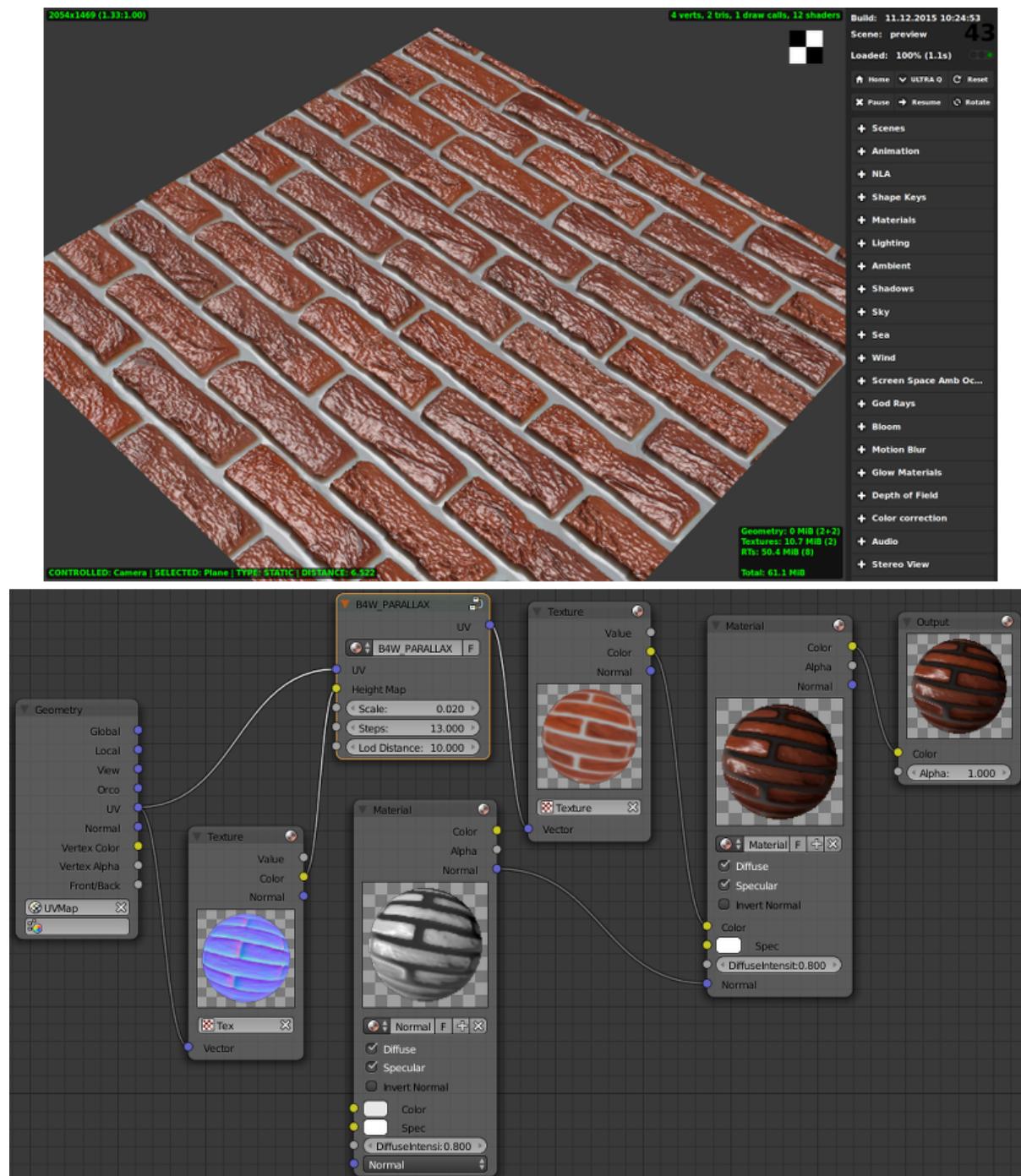
**LOW** Значение, которое будет подано в выходной параметр *Color* при низком качестве изображения.

**Fac** Значение, определяющее, какое из значений (**HIGH** или **LOW**) будет отображаться в Blender. Может принимать значения от 0 до 1. При значении параметра менее

0.5 будет виден вариант HIGH, при значении 0.5 и выше будет виден вариант LOW.

## Parallax (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.

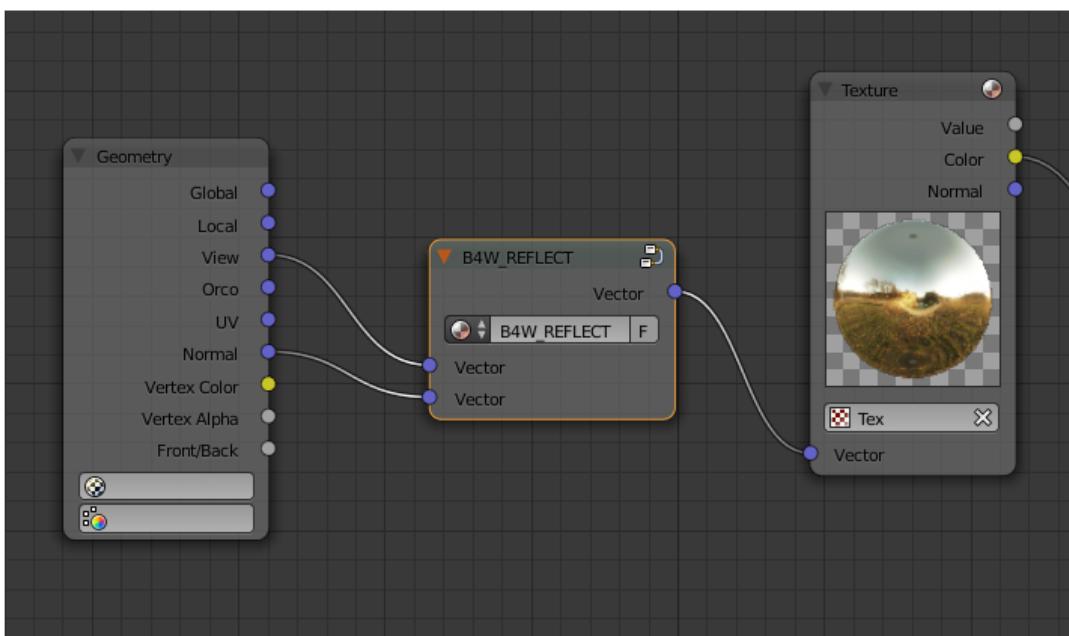
**Lod Distance** Максимальное расстояние от камеры, на котором виден эффект.

## Выходные параметры

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

## Reflect (B4W\_REFLECT)

Вычисляет отражение заданного вектора относительно заданной нормали. Может применяться для наложения кубической карты (кубемар) на объект.



## Входные параметры

**Vector** Заданный вектор. Должен быть подключён ко входу **View** ноды **Geometry**.

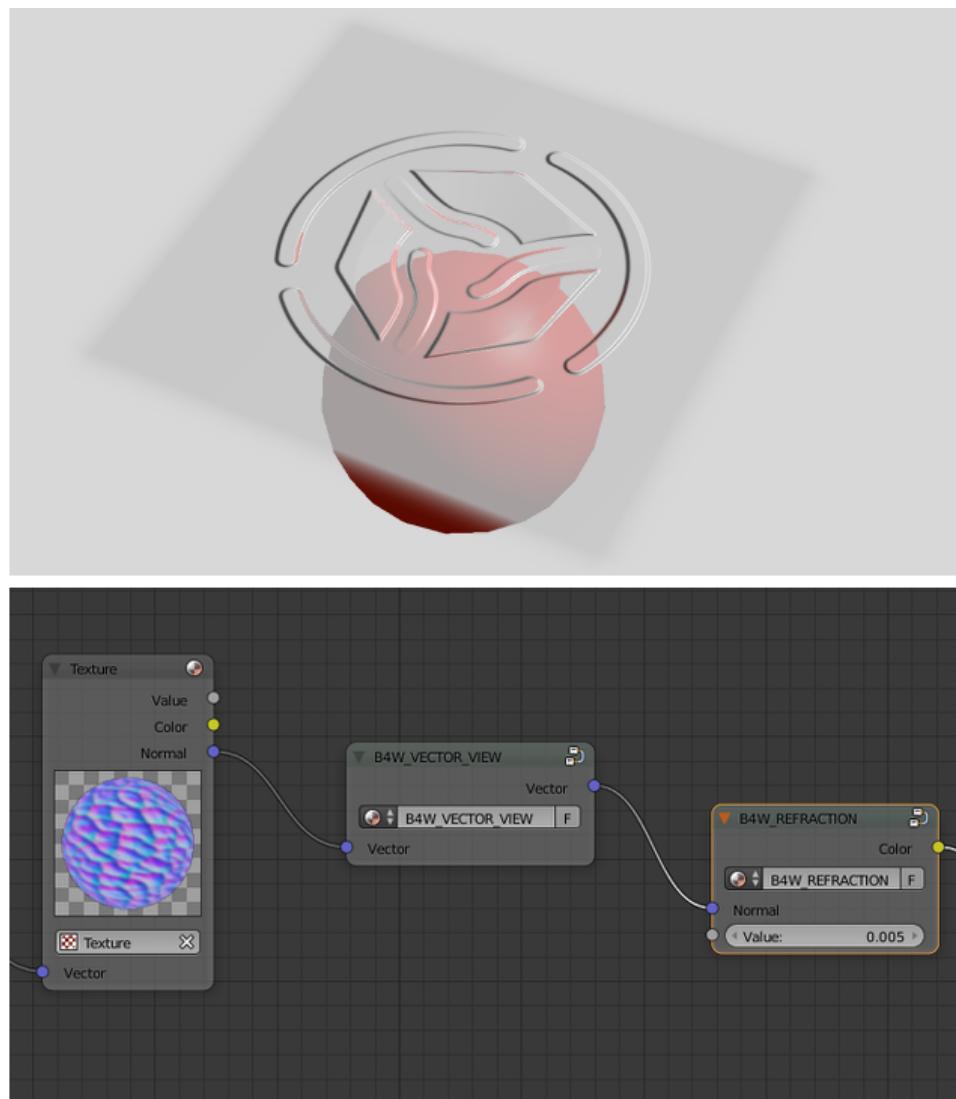
**Vector** Заданная нормаль. Для получения желаемого результата, вектор должен быть нормирован. Должен быть подключён ко входу **Normal** ноды **Geometry**.

## Выходные параметры

**Vector** Отражённый вектор. Следует подключать к входному параметру **Vector** ноды **Texture**, содержащего кубическую карту.

## Refraction (B4W\_REFRACTION)

Реализует эффект преломления. Этот эффект виден только в движке Blend4Web, но не в Blender.



### Входные параметры

**Normal** Вектор нормали в пространстве камеры, по которому 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.

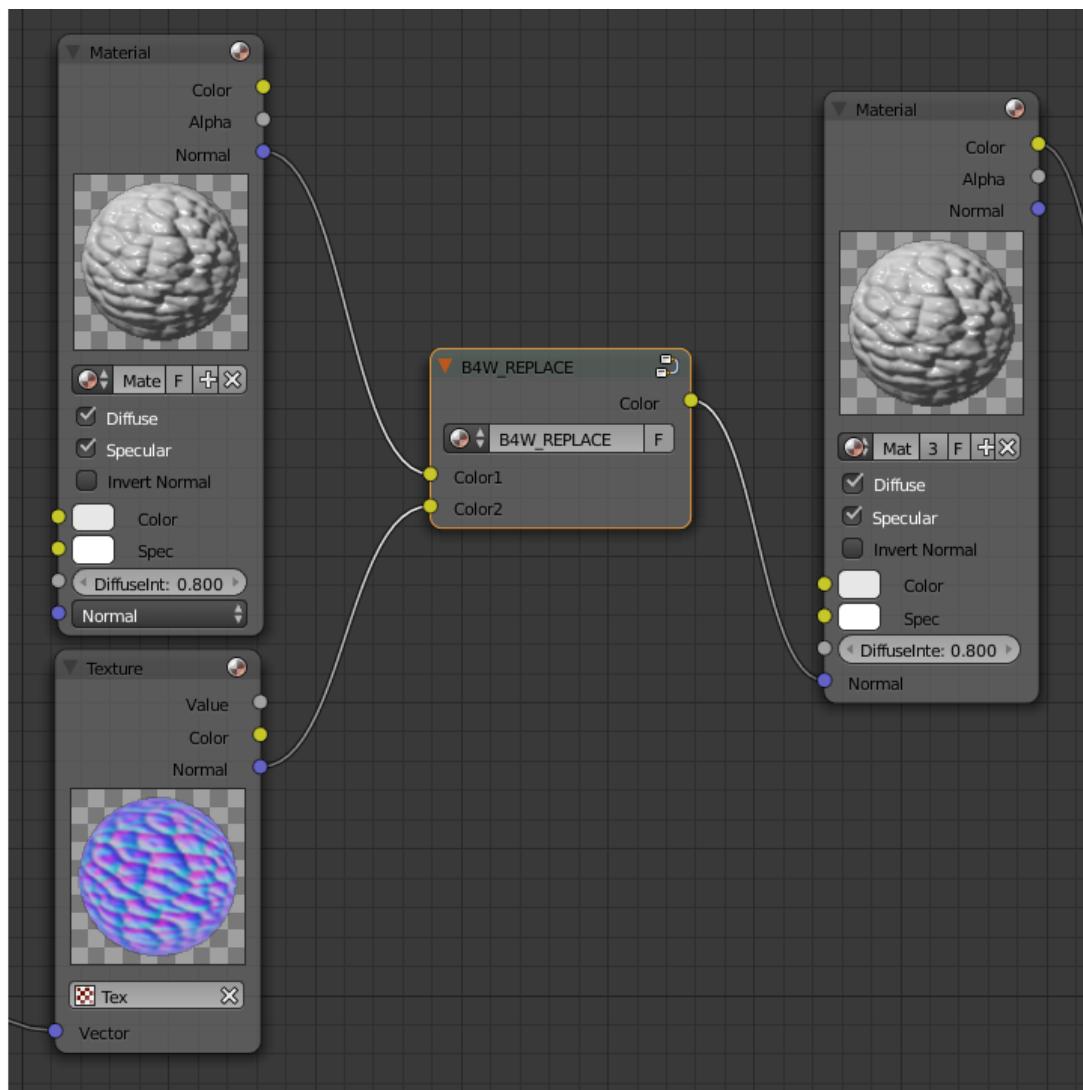
**Примечание:** Для отображения эффекта необходимо переключить опцию Refractions на панели Render > Reflections and Refractions в состояние AUTO или ON. Объект должен быть с типом прозрачности Alpha Blend.

**См.также:**

*Прозрачность*

## Replace (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'а не поддерживают тангентное пространство координат, в связи с чем единственный способ корректного отображения карт нормалей во viewport'e - их подключение внутри ноды Material.

### Входные параметры

**Color1** Конструкция нод, видимая во виджете Blend4Web.

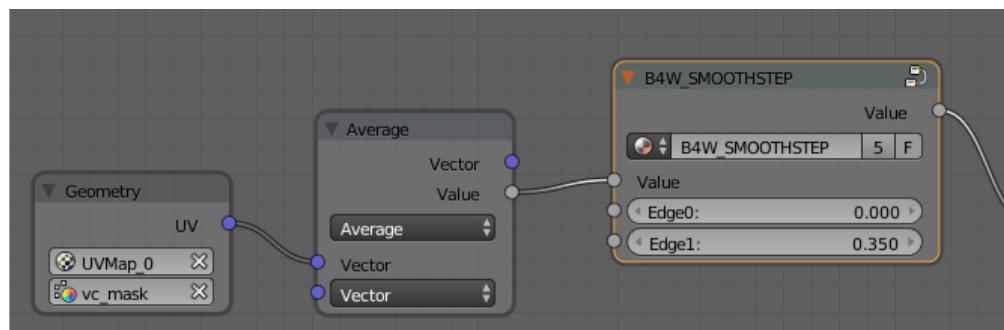
**Color2** Конструкция нод, видимая объекта в движке Blend4Web.

### Выходные параметры

**Color** Следует подключать к выходу Color ноды Material или Extended Material.

## Smoothstep (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.

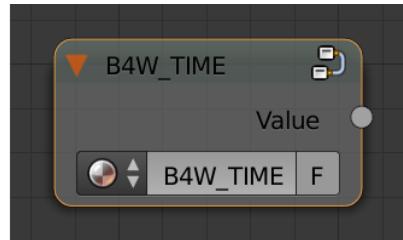
---

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

---

## Time (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.



### Входные параметры

Отсутствуют.

### Выходные параметры

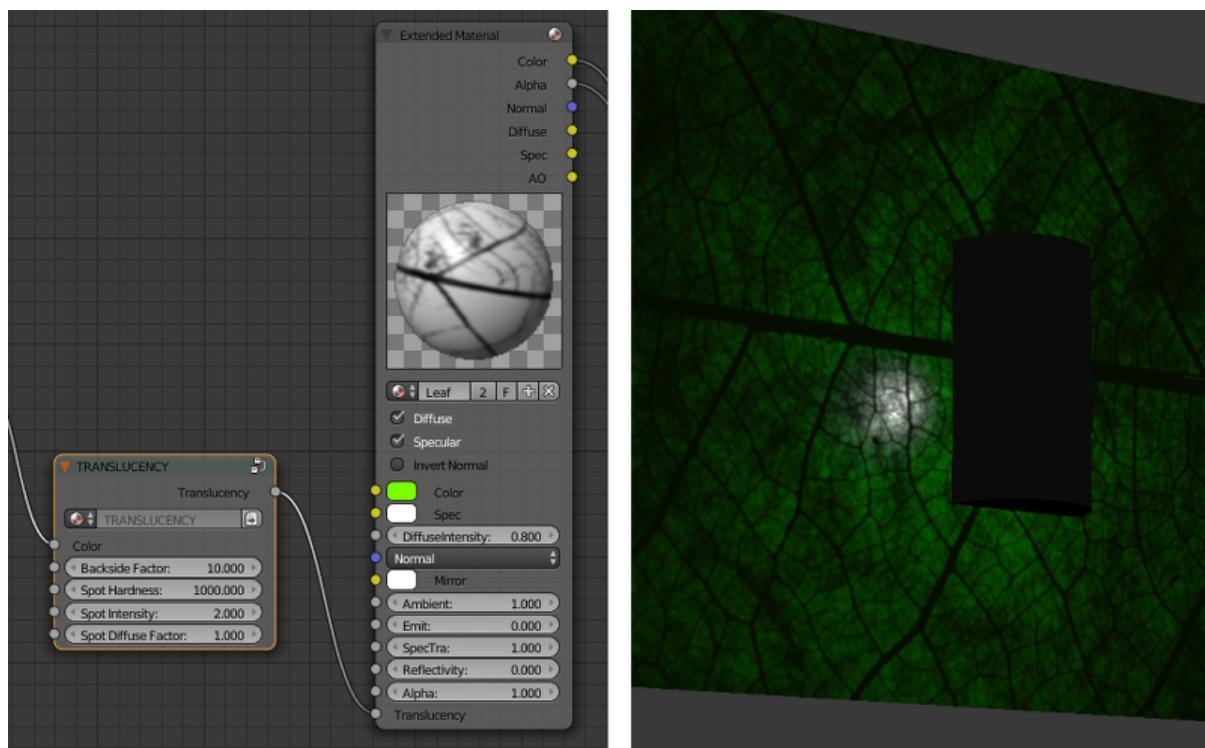
**Value** Время (в секундах), прошедшее с момента старта движка.

См.также:

*Анимация нод Value и RGB*

## Translucency (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 стороне 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**.

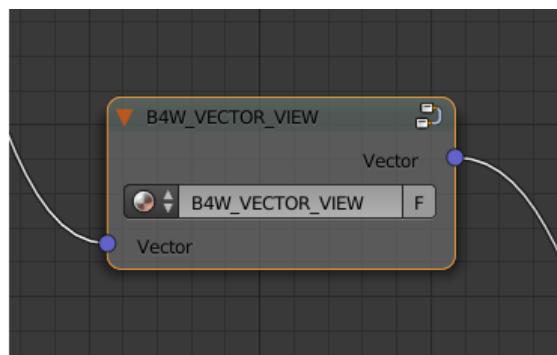
---

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

---

## Vector View (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**.



### Входные параметры

**Vector** Вектор в мировой системе координат.

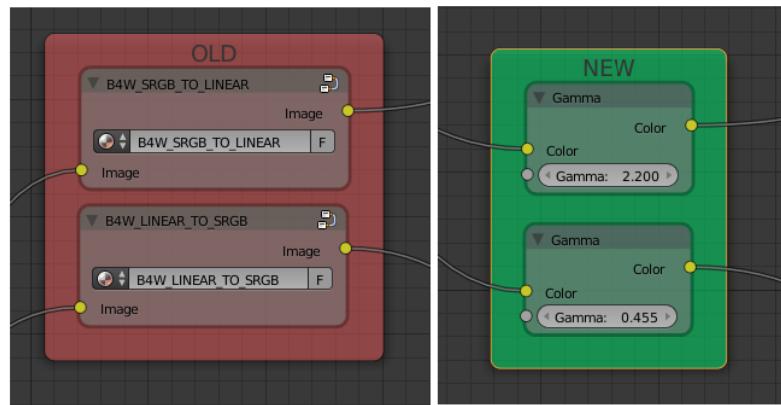
### Выходные параметры

**Vector** Вектор в системе координат камеры.

## Linear to sRGB и sRGB to Linear (Deprecated)

Преобразование цвета из линейного цветового пространства в пространство sRGB и наоборот. Функционал объявлен устаревшим с версии 15.04. В более новых версиях для преобразования цвета из sRGB в линейное пространство следует использовать

нативную ноду **Gamma** со значением 2.200, а для преобразования из линейного пространства в sRGB - ту же ноду со значением 0.455.



**См.также:**

*Коррекция в нодовых материалах*

## Ноды Cycles

**Примечание:** Поддержка нод Cycles - экспериментальная возможность, которая на данный момент не рекомендуется к использованию при разработке серьёзных приложений.

Следует также заметить, что при использовании нод Cycles в Blend4Web конечный результат будет близок к изображению, созданному в самом Cycles, но всё же не идентичен ему.

Движок поддерживает следующие ноды Cycles:

- **Material Output** (поддерживаются только входы **Surface** и **Displacement**);
- **BSDF Diffuse**;
- **BSDF Glossy** (поддерживается только распределение **GGX**; параметр **Roughness** не влияет на отражения);
- **Transparent BSDF**;
- **Mix Shader**;
- **Fresnel**;
- **Layer Weight**;
- **Image Texture**;
- **Environment Texture**;
- **Object Info**;

- **Bump.**

Следующие ноды поддерживаются частично:

- **Texture Coordinates** (параметр **From Dupli** не поддерживается);
- **UV Map** (параметр **From Dupli** не поддерживается);
- **Geometry** (параметры **Pointness** и **Parametric** не поддерживаются).
- **Emission** (не влияет на освещённость сцены).

Прочие ноды **Cycles** в большинстве случаев не будут работать в Blend4Web так же, как они работают в Blender. Кроме того, они могут не работать вообще или даже приводить к некорректной работе материала, в котором используются. Однако использование таких нод не вызовет нестабильности в работе приложения.

# Редактор логики

---

## Содержание

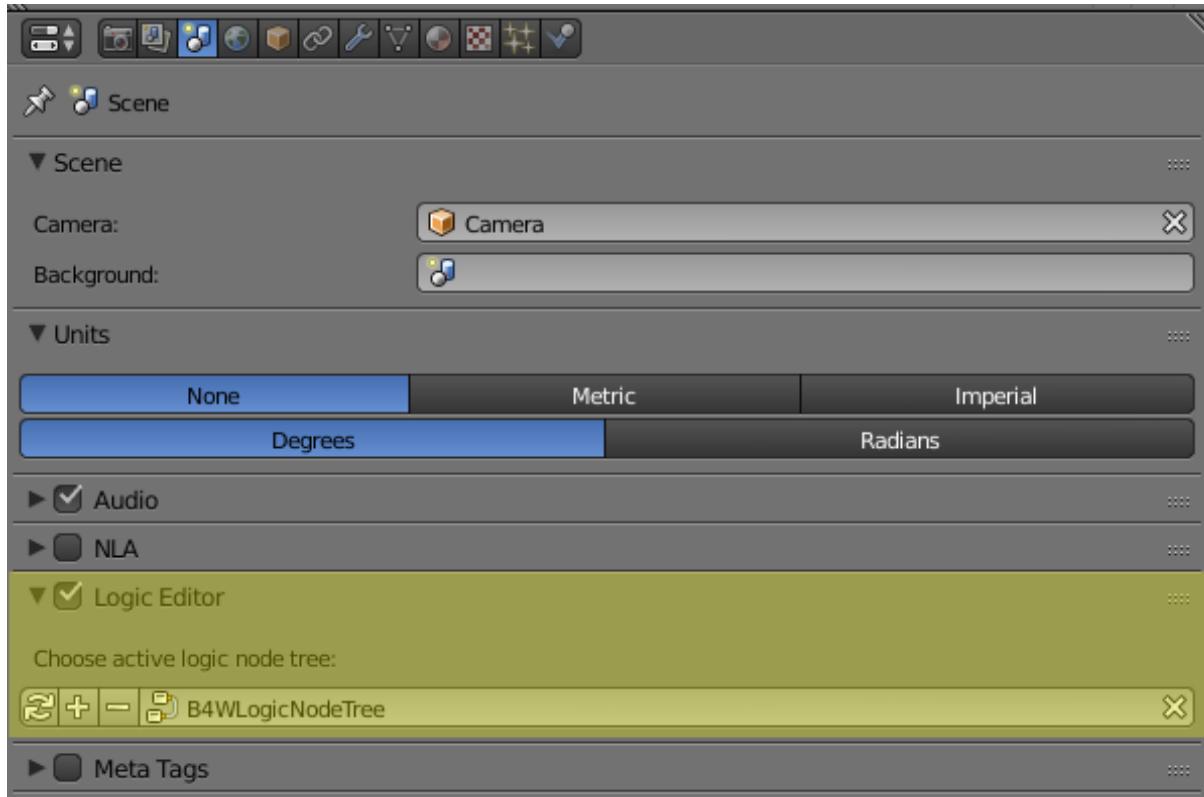
- *Редактор логики*
  - Основы
  - Control Flow
  - Animation
  - Camera
  - Object
  - Operations
  - Sound
  - Network
  - Debug
  - Deprecated
  - Layout
  - Отладка

## Основы

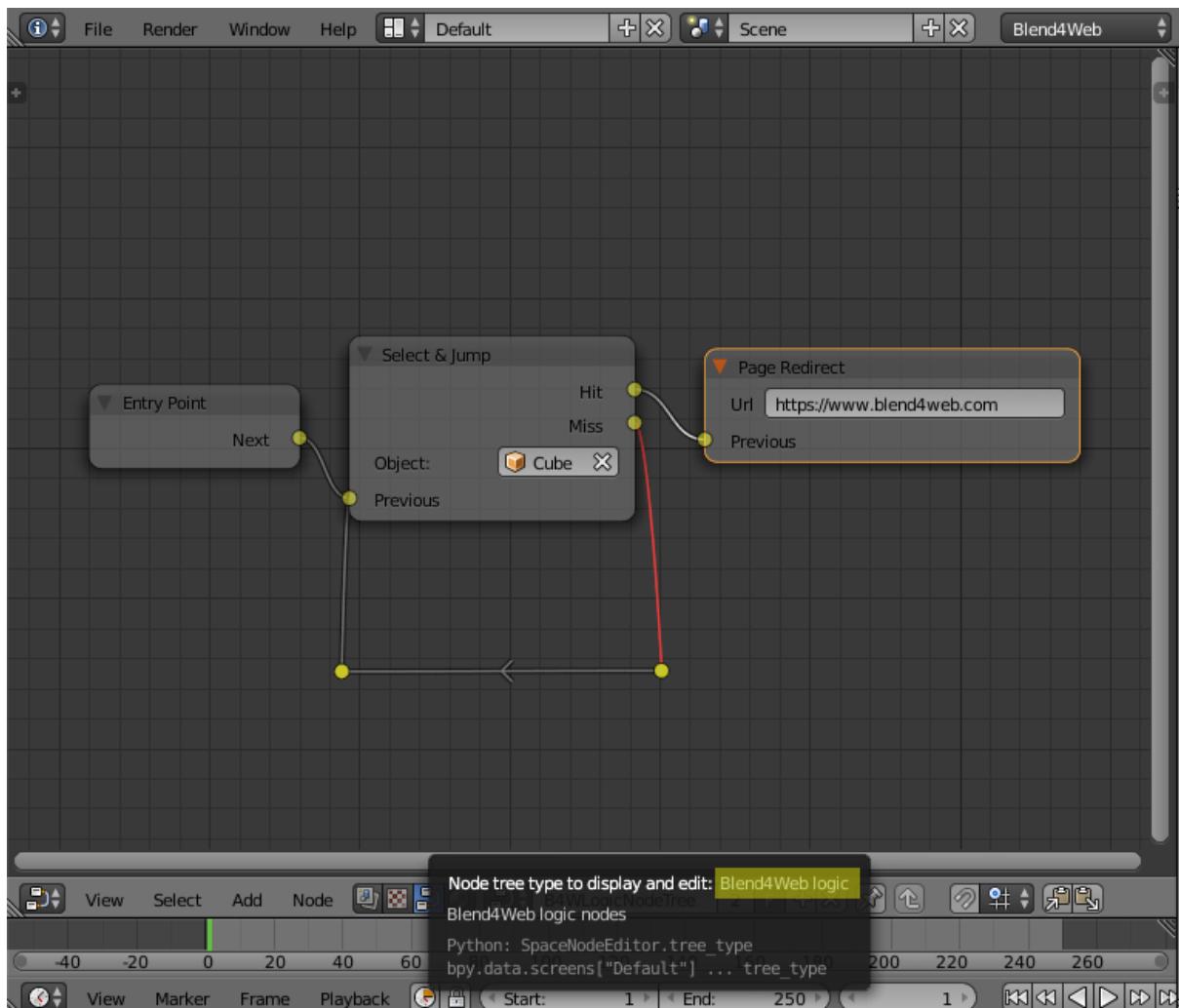
Визуальное программирование осуществляется путем создания нодового дерева логики в окне нодового редактора. С помощью Logic Editor можно существенно расширить функциональность сцены, не прибегая при этом к программированию.



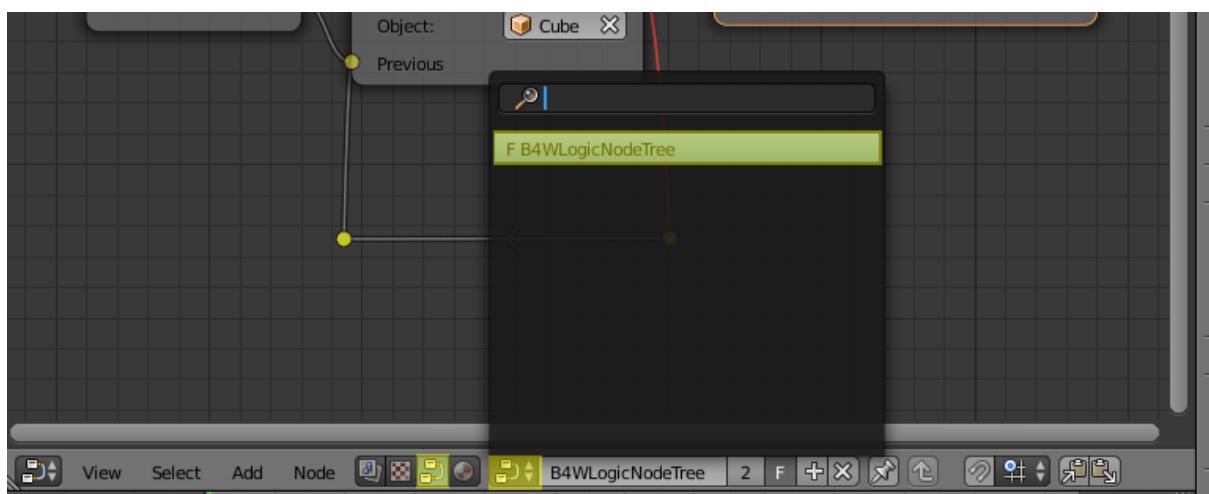
Для активации логики, на вкладке настроек сцены Blender необходимо установить галочку **Logic Editor** и добавить/выбрать активное нодовое дерево:



Для редактирования дерева логики необходимо перейти в Редактор Нод (Node Editor) выбрать тип нодового дерева Blend4Web logic:



Затем выбрать дерево для редактирования:

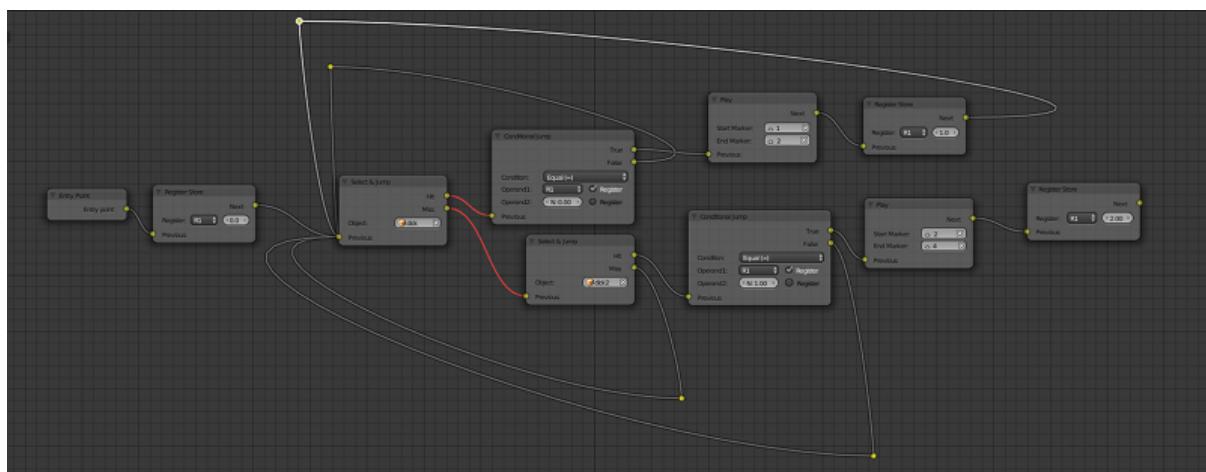


Создание нод осуществляется привычным для Blender сочетанием клавиш **Shift+A**.

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 переменные, которые могут иметь как числовые, так и строковые значения. Переменные 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:

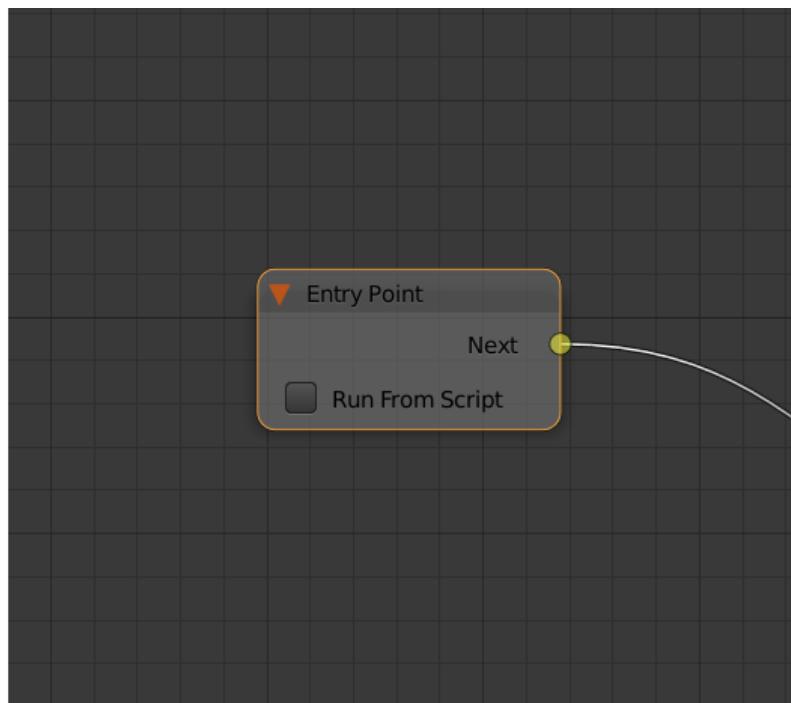


Ниже перечислены все доступные ноды.

## Control Flow

### Entry Point

Является точкой начала выполнения скрипта. Использование нескольких точек входа позволяет эмулировать многопоточное выполнение.



### Входные параметры

Отсутствуют.

### Выходные параметры

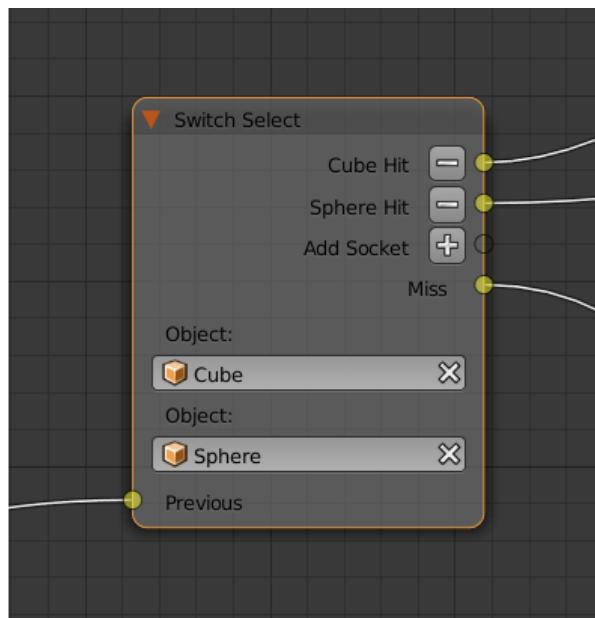
*Next* Следующая нода.

### Внутренние параметры

**Run From Script** Если этот параметр активирован, точка входа может быть запущена с помощью API, используя метод `run_entrypoint`.

## Switch Select

Позволяет отследить событие выбора какого-либо объекта из списка объектов.



## Входные параметры

**Previous** Предыдущая нода.

## Выходные параметры

**<имя объекта> Hit** Этот параметр передаёт управление следующей ноде в случае, если выбран объект, упомянутый в его названии. По умолчанию нода **Switch Select** имеет один параметр этого типа, также присутствует возможность создавать новые и удалять уже существующие (нода может и не иметь ни одного такого параметра).

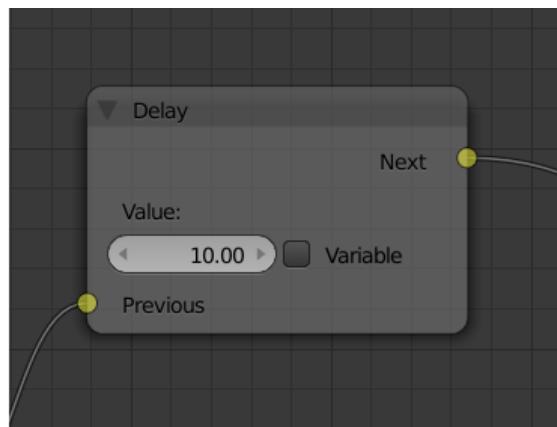
**Miss** This parameter will pass the control to the next node when the user selects any object with the **Selectable** property enabled (or used by another **Switch Select** node), but not specified in the **Switch Select** node.

## Внутренние параметры

**Object** Один из объектов, доступных для выбора пользователю. Такие параметры создаются и удаляются автоматически при создании или удалении выходных параметров типа **Hit**.

## Delay

Позволяет установить задержку (в секундах) перед выполнением следующей ноды.



## Входные параметры

**Previous** Предыдущая нода.

## Выходные параметры

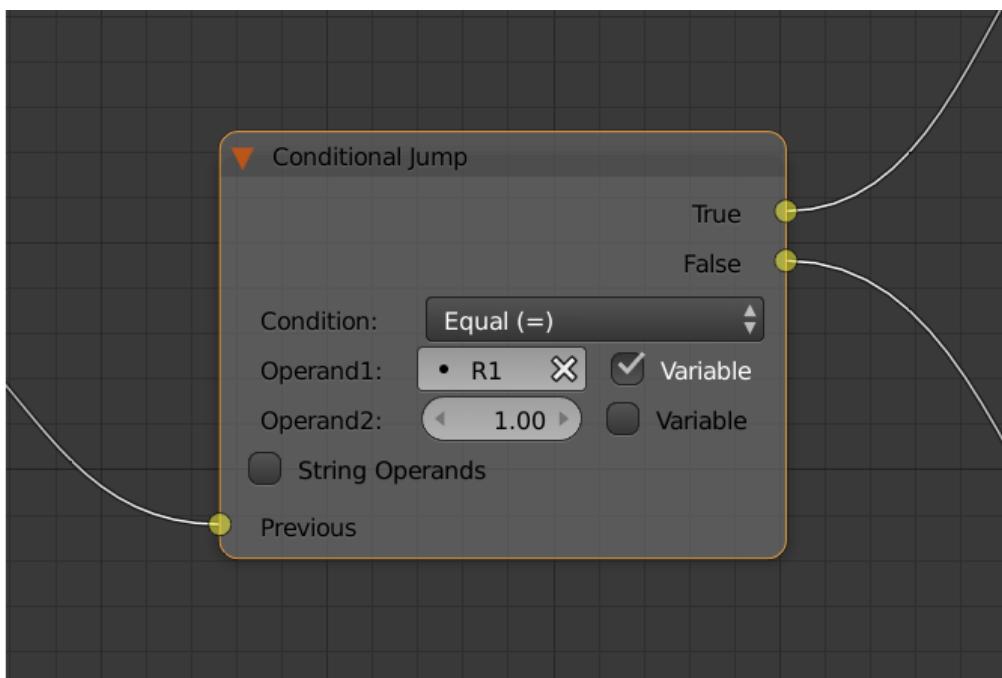
**Next** Следующая нода.

## Внутренние параметры

**Value** Время (в секундах), на которое задерживается передача управления следующей ноде. По умолчанию равно нулю. Может быть задано в явном виде или же взято из переменной (в случае, если активирован параметр **Variable**).

## Conditional Jump

Перейти к указанной ноде в случае выполнения выбранного условия. В качестве параметров условия (операндов) могут выступать также переменные, которые активируются с помощью соответствующих переключателей.



## Входные параметры

**Previous** Предыдущая нода.

## Выходные параметры

**True** Утверждение истинно.

**False** Утверждение ложно.

## Внутренние параметры

**Condition** Логическое утверждение. Может иметь один из следующих типов:

- *Equal (=)* - первый operand равен второму.
- *Not Equal ( $\neq$ )* - первый operand не равен второму.
- *Less Than ( $<$ )* - первый operand меньше второго.
- *Greater Than ( $>$ )* - первый operand больше второго.
- *Less Than Or Equal ( $\leq$ )* - первый operand меньше либо равен второму.
- *Greater Than Or Equal ( $\geq$ )* - первый operand больше либо равен второму.

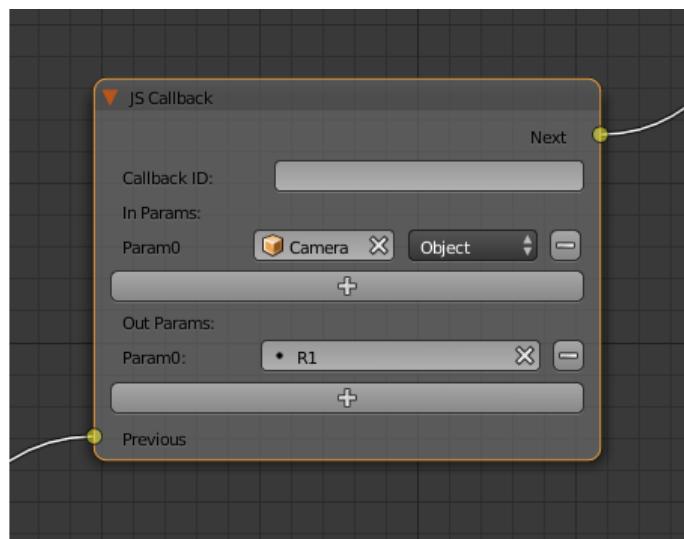
**Operand1** First operand of the logical condition. Contains a numeric value of a string (if the **String Operators** parameter is enabled). Can be specified in the node or can be a link to one of the variables (if the **Variable** parameter at the right of it is enabled).

**Operand2** Второй operand логического выражения. Аналогичен первому.

**String Operands** If this parameter is enabled, first and second operands can use strings (set manually or by a variable) as their values.

## JS Callback

Используется для вызова callback-функции, заранее заданной в вашем Blend4Web-приложении.



### Входные параметры

**Previous** Предыдущая нода.

### Выходные параметры

**Next** Следующая нода.

### Внутренние параметры

**Callback ID** ID JavaScript-функции, которая будет вызываться нодой.

**In Params** Список входных параметров функции. Каждый параметр может представлять собой либо переменную, либо ссылку на объект сцены. Количество входящих параметров может быть настроено. По умолчанию список пуст.

Входные параметры передаются callback-функции в виде массива, который используется в качестве первого аргумента функции.

**Param <param\_number>** Задаёт входящий параметр. Параметр может быть как переменной (R1 по умолчанию), так и ссылкой на объект сцены, в зависимости от значения параметра Type (у каждого входного параметра всегда есть соответствующий параметр Type).

**Type** Тип соответствующего входного параметра. Может иметь одно из двух значений: **Variable** (в этом случае в качестве параметра будет использована одна из переменных) или **Object** (ссылка на объект сцены).

**Out Params** Список выходных параметров. По умолчанию пуст. Количество выходных параметров поддаётся настройке.

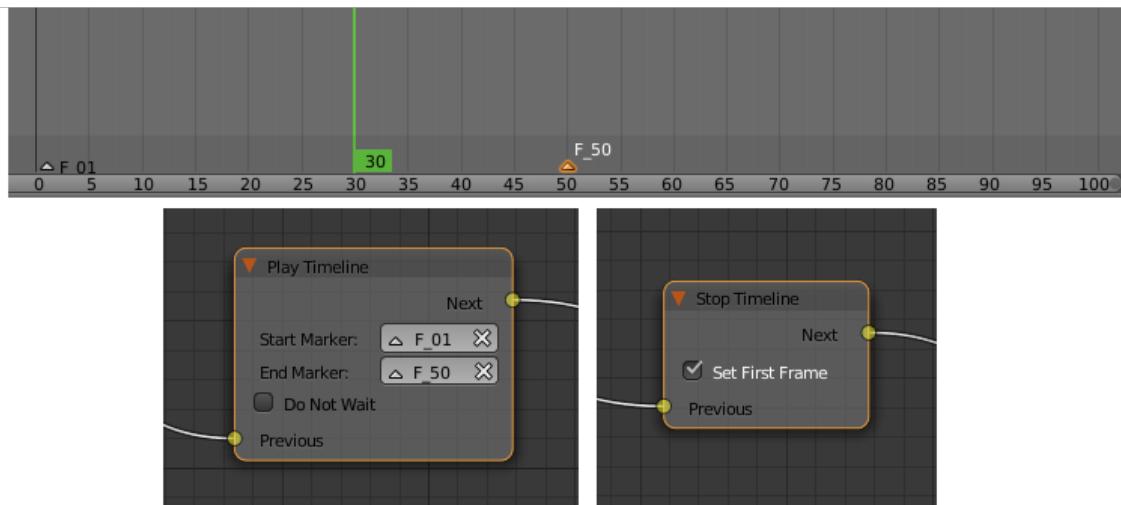
Массив, состоящий из выходных параметров, используется в качестве второго аргумента callback-функции.

**Param <param\_number>** Задаёт переменную, которая будет использоваться в качестве одного из выходных параметров. По умолчанию используется переменная R1.

## Animation

### Play Timeline и Stop Timeline

Позволяют управлять NLA анимацией. **Play Timeline** проигрывает 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. **Stop Timeline** останавливает воспроизведение.



#### Входные параметры

**Previous** Предыдущая нода.

#### Выходные параметры

**Next** Следующая нода.

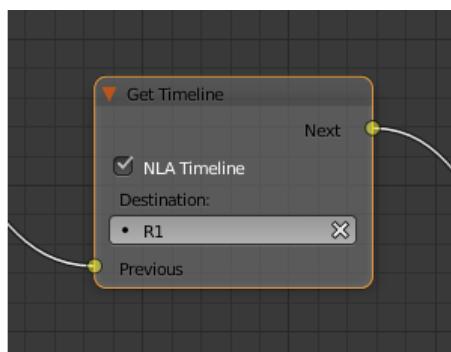
## Внутренние параметры

**Start Marker** Кадр, с которого должна воспроизводиться анимация. Если не указан, анимация воспроизводится с первого кадра и может работать некорректно.

**End Marker** Кадр, на котором проигрывание анимации должно остановиться. Если не указан, анимация воспроизводится до окончания шкалы времени и может работать некорректно.

## Get Timeline

Эта нода используется для получения текущего кадра NLA-анимации или таймлайна.



### Входные параметры

**Previous** Предыдущая нода.

### Выходные параметры

**Next** Следующая нода.

### Внутренние параметры

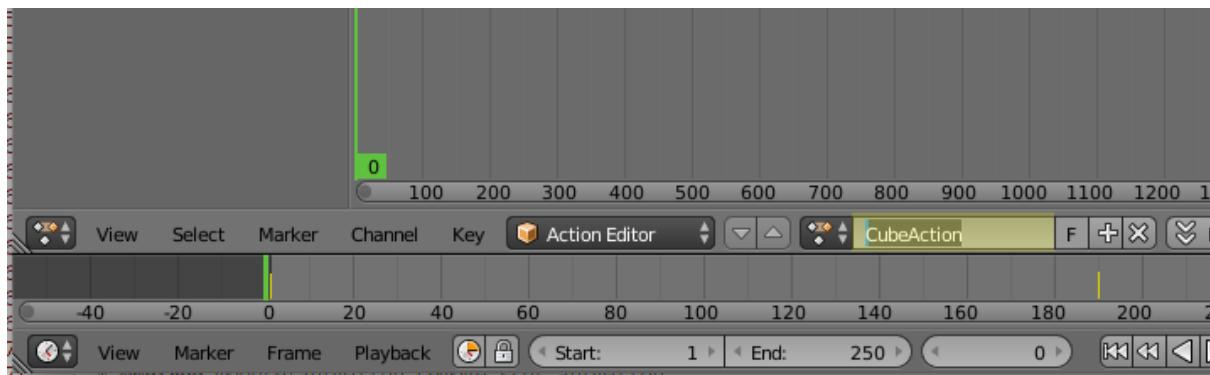
**NLA Timeline** Если этот параметр активирован, нода будет возвращать текущий кадр NLA-анимации. Если параметр не активирован, нода будет возвращать текущий кадр таймлайна. Включено по умолчанию.

**Destination** Задаёт переменную, в которую будет сохранено значение текущего кадра. По умолчанию имеет значение R1.

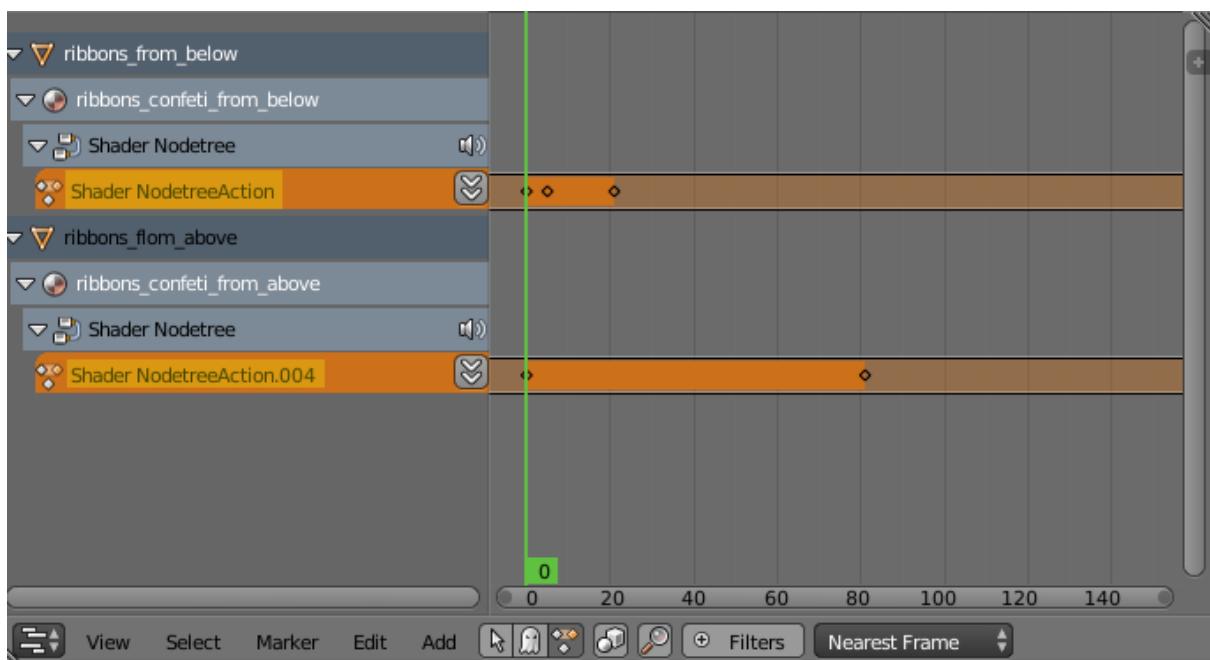
## Play Animation

Используется для воспроизведения анимации объекта. Проигрываемая анимация может иметь один из трёх типов:

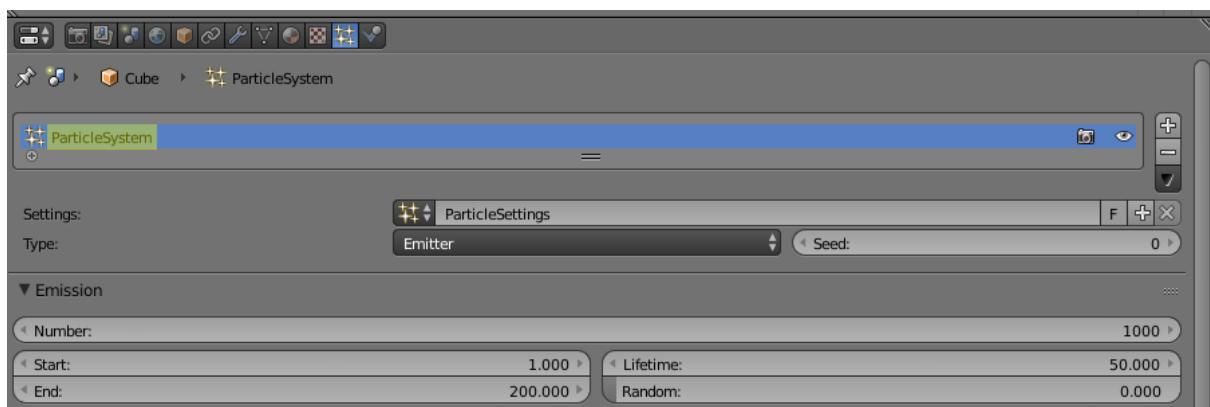
Обычный Action:



Шейдерный Action:



Система частиц:



### Входные параметры

**Previous** Предыдущая нода.

## Выходные параметры

**Next** Следующая нода.

## Внутренние параметры

**Object** Имя объекта, анимацию которого следует воспроизвести.

**Anim. Name** Имя анимации, которую следует воспроизвести. В случае, если имя анимации не указано, проигрывается вся временная шкала.

**Behavior** Задаёт поведение анимации. Может иметь одно из следующих значений:

- *Finish Stop* - анимация воспроизводится один раз.
- *Finish Reset* - анимация воспроизводится один раз, после её завершения объект возвращается в исходное состояние.
- *Loop* - анимация воспроизводится циклически до тех пор, пока не остановлена с помощью ноды **Stop Animation**.

**Do Not Wait** Если этот параметр активирован, нода **Play Animation** передаст управление следующей ноде сразу после начала воспроизведения анимации. В противном случае управление будет передано только после окончания анимации.

## Stop Animation

Используется для остановки анимации объекта.

## Входные параметры

**Previous** Предыдущая нода.

## Выходные параметры

**Next** Следующая нода.

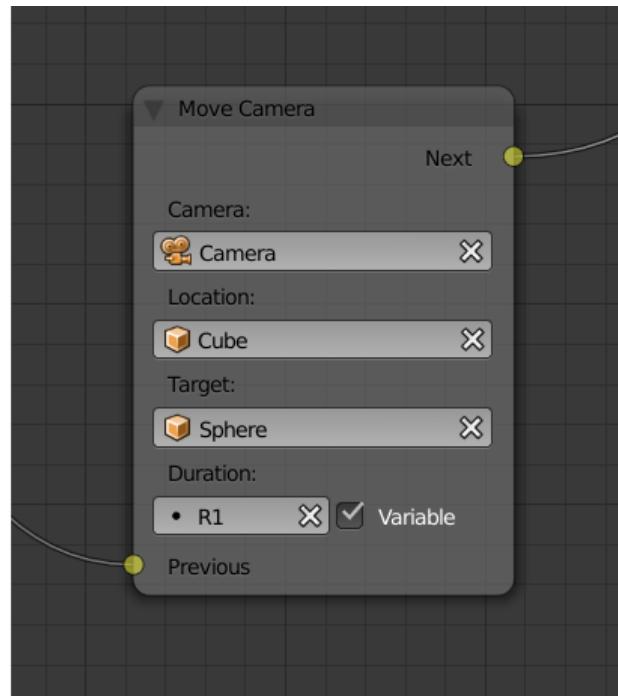
## Внутренние параметры

**Set First Frame** Вернуться к первому кадру после того, как анимация остановлена.

## Camera

### Move Camera

Позволяет перемещать камеру, в том числе с плавной интерполяцией положения.



#### Входные параметры

**Previous** Предыдущая нода.

#### Выходные параметры

**Next** Следующая нода.

#### Внутренние параметры

**Camera** Камера, которую необходимо переместить.

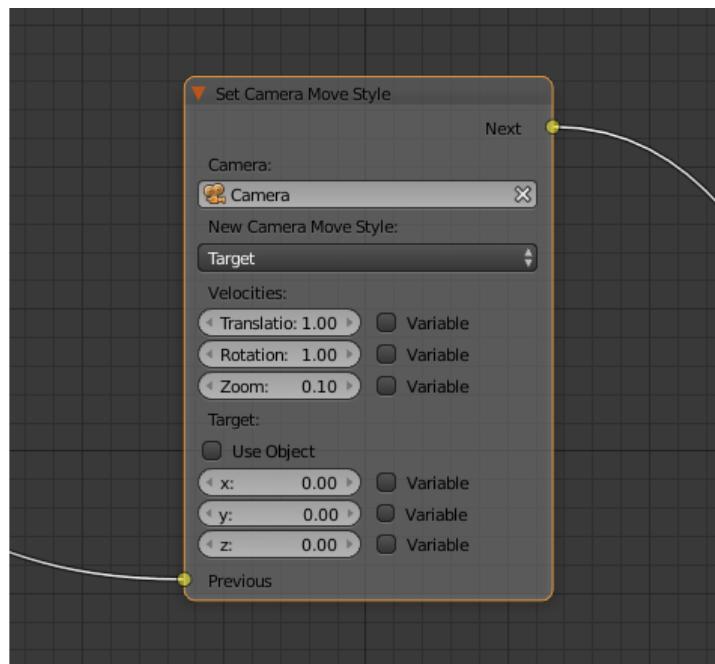
**Location** Объект, к которому перемещается камера. После того, как перемещение завершится, координаты камеры и объекта будут совпадать.

**Target** Объект, в направлении которого камера будет развернута после перемещения.

**Duration** Время (в секундах), которое займёт перемещение. По умолчанию равно нулю (в этом случае камера перемещается мгновенно). Может быть задано вручную или с помощью ссылки на переменную (эта возможность появляется при активации параметра **Variable**).

## Set Camera Move Style

Используется для задания нового режима перемещения камеры.



### Входные параметры

**Previous** Предыдущая нода.

### Выходные параметры

**Next** Следующая нода.

### Внутренние параметры

**Camera** Камера, режим перемещения которой будет изменён.

**New Camera Move Style** Задаёт новый режим перемещения выбранной камеры. Доступно четыре варианта: Hover, Eye, Target и Static.

Следующие параметры недоступны в том случае, если параметр New Camera Move Style имеет значение Static:

**Translation** Задаёт скорость перемещения камеры. Значение по умолчанию - 1.0.

**Rotation** Задаёт скорость поворота камеры. Значение по умолчанию - 1.0.

**Zoom** Задаёт скорость приближения и удаления камеры. Значение по умолчанию - 0.10. Этот параметр доступен только в том случае, если параметр New Camera Move Style имеет значение Hover или Target.

Следующие параметры используются для задания цели или точки привязки камеры. Они доступны только если параметр **New Camera Move Style** имеет значение **Hover** или **Target**.

**Use Object** Эта опция включает и выключает использования объекта сцены в качестве цели или точки привязки камеры (в зависимости от типа камеры). Если опция включена, появляется текстовое поле для задания объекта. Эта опция деактивирована по умолчанию.

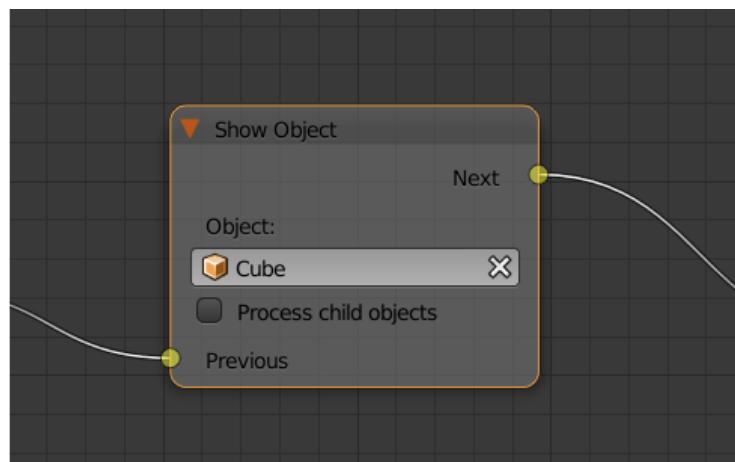
Если параметр **Use Object** деактивирован, становятся доступны следующие три опции:

- x* X-координата цели или точки привязки камеры.
- y* Y-координата цели или точки привязки камеры.
- z* Z-координата цели или точки привязки камеры.

## Object

### Show Object

Используется для отображения объектов.



#### Входные параметры

**Previous** Предыдущая нода.

#### Выходные параметры

**Next** Следующая нода.

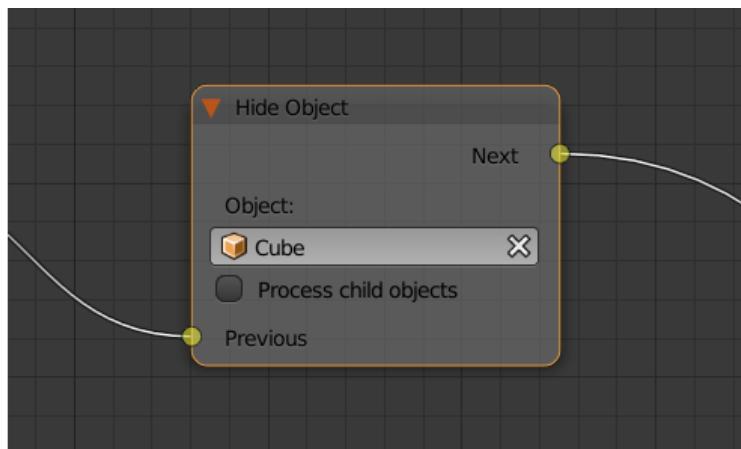
## Внутренние параметры

*Object* Объект, который нужно отобразить.

*Process child objects* Если этот параметр включён, объекты-потомки также будут показаны.

## Hide Object

Используется для скрытия объекта.



## Входные параметры

*Previous* Предыдущая нода.

## Выходные параметры

*Next* Следующая нода.

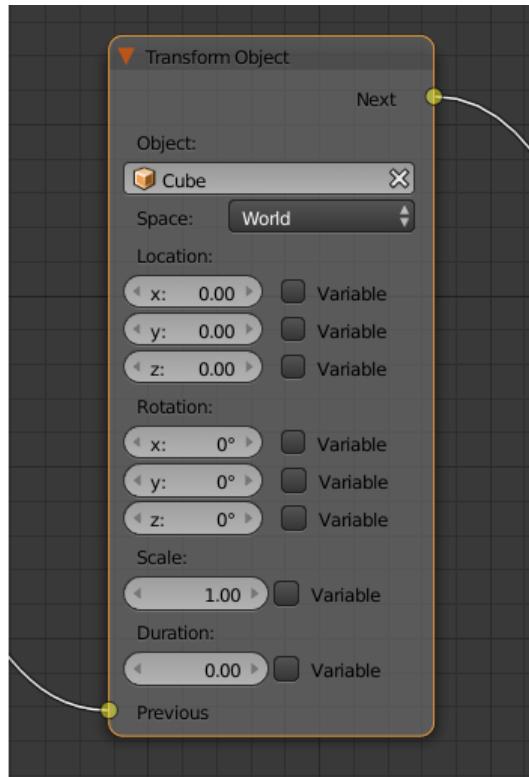
## Внутренние параметры

*Object* Объект, который нужно скрыть.

*Process child objects* Если этот параметр включён, объекты-потомки также будут скрыты.

## Transform Object

Используется для изменения координат, размеров и углов поворота объекта.



## Входные параметры

**Previous** Предыдущая нода.

## Выходные параметры

**Next** Следующая нода.

## Внутренние параметры

**Object** Объект, который нужно трансформировать.

**Space** Этот параметр определяет систему координат, которая будет использована для трансформации объекта. Может принимать одно из следующих значений:

- **World** - глобальная система координат.
- **Parent** - система координат объекта, который является предком объекта, указанного в параметре **Object**. Точка привязки родительского объекта используется как центр координат, а углы наклона задают направления координатных осей.
- **Local** - локальная система координат выбранного объекта. Схожа с системой координат родительского объекта, но в данном случае точка привязки самого объекта используется в качестве начала координат.

Значение по умолчанию - **World**.

**Location** Перемещение объекта по осям X, Y и Z. По умолчанию все три значения равны нулю. Значения могут быть заданы в самой ноде либо с помощью переменной (если включена опция Variable).

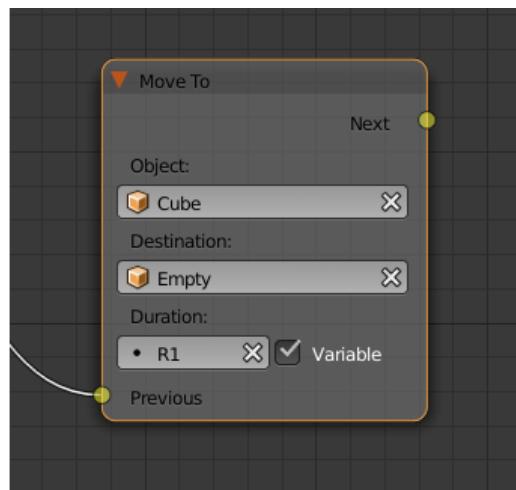
**Rotation** Углы поворота объекта вокруг осей X, Y и Z. По умолчанию все три угла равны нулю. Значения могут быть заданы непосредственно в ноде или через переменные (в том случае, если включена опция Variable).

**Scale** Масштаб объекта. Может быть задан напрямую или с помощью переменной (если включена опция Variable). По умолчанию равен нулю.

**Duration** Время (в секундах), которое займёт трансформация. Может быть задано как напрямую, так и посредством переменной (для этого должна быть включена опция Variable). По умолчанию равно нулю.

## Move To

Позволяет перемещать объекты.



### Входные параметры

**Previous** Предыдущая нода.

### Выходные параметры

**Next** Следующая нода.

### Внутренние параметры

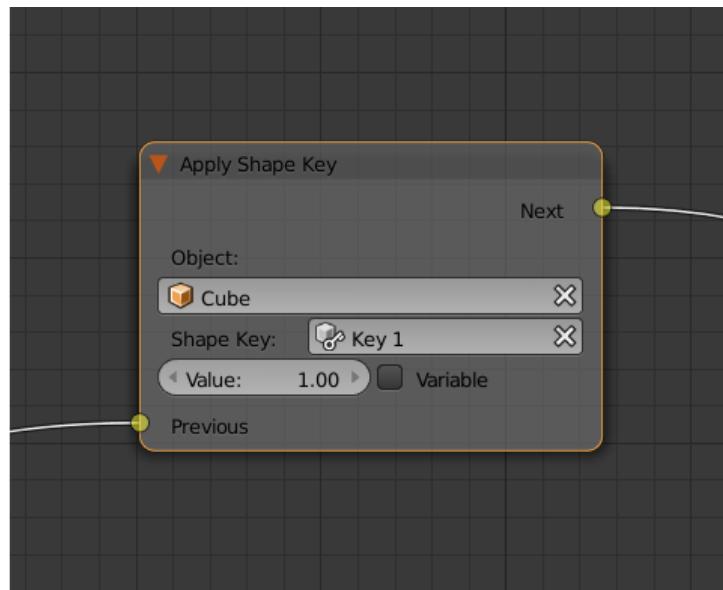
**Object** Объект, который требуется переместить.

**Destination** Цель (объект, источник света, камера и т.п.), к которой перемещается объект. После того, как перемещение завершится, координаты объекта и цели будут совпадать.

**Duration** Время (в секундах), которое займёт перемещение. По умолчанию равно нулю (в этом случае объект не движется, а мгновенно перемещается к точке назначения). Может быть задано как в самой ноде, так и с помощью переменной (если активирован параметр **Variable**).

## Apply Shape Key

Позволяет задать значение параметра Shape Key.



### Входные параметры

**Previous** Предыдущая нода.

### Выходные параметры

**Next** Следующая нода.

### Внутренние параметры

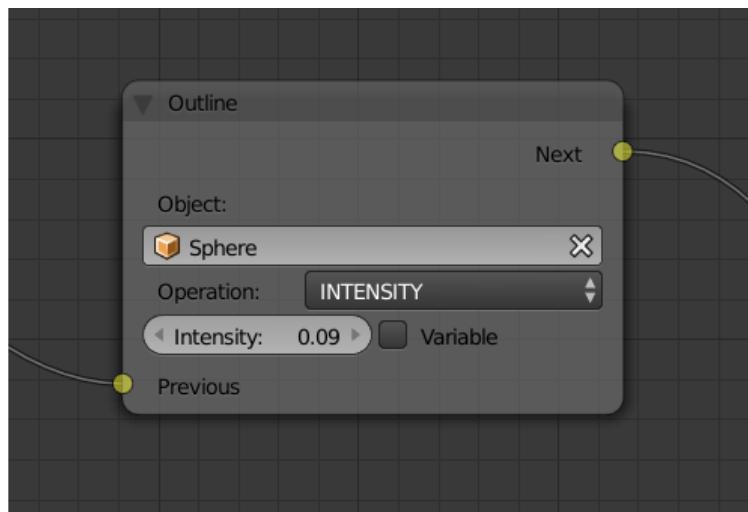
**Object** Объект, к которому нужно применить трансформацию.

**Shape Key** Ключ формы, который нужно применить к объекту.

**Value** Степень влияния выбранного ключа формы на объект. Значение может быть задано в самой ноде или взято из переменной. Изменяется в пределах от 0 до 1.

## Outline

Позволяет управлять эффектом обводки объекта.



### Входные параметры

**Previous** Предыдущая нода.

### Выходные параметры

**Next** Следующая нода.

### Внутренние параметры

**Object** Объект, эффект обводки которого требуется настроить.

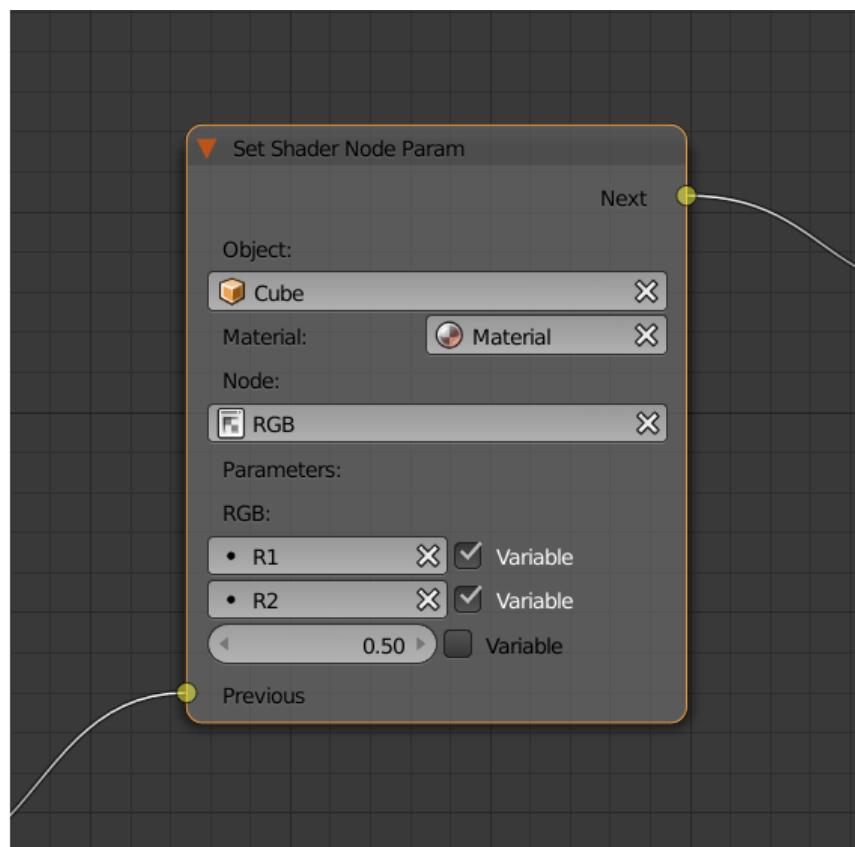
**Operation** Операция, которую следует произвести с обводкой объекта. Этот параметр может принимать одно из следующих значений:

- *PLAY* - включает анимацию обводки
- *STOP* - выключает её
- *INTENSITY* - позволяет настраивать интенсивность обводки

**Intensity** Интенсивность обводки объекта. Этот параметр появляется только если параметр **Operation** имеет значение **INTENSITY**. Интенсивность может быть задана численно или взята из переменной (для этого нужно активировать параметр **Variable**).

## Set Shader Node Param

Позволяет установить значение одного или нескольких параметров шейдерной ноды. В данный момент поддерживаются ноды **Value** и **RGB**.



## Входные параметры

**Previous** Предыдущая нода.

## Выходные параметры

**Next** Следующая нода.

## Внутренние параметры

**Object** Объект, материал которого нужно отредактировать.

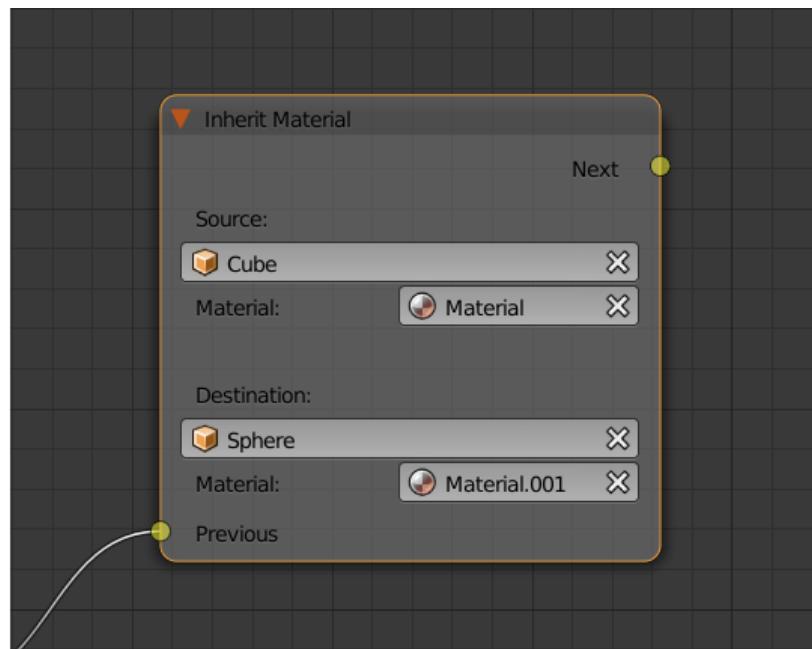
**Material** Материал, параметры которого нужно изменить. Должен использовать ноды.

**Node** Нода, параметры которой нужно изменить. На данный момент поддерживаются только ноды **Value** и **RGB**.

**Parameters** Доступные для редактирования параметры выбранной ноды. Значения этих параметров могут быть заданы непосредственно в самой ноде или взяты из указанных переменных (для этого нужно активировать параметр **Variable**).

## Inherit Material

Позволяет скопировать атрибуты одного материала на другой.



### Входные параметры

**Previous** Предыдущая нода.

### Выходные параметры

**Next** Следующая нода.

### Внутренние параметры

**Source** Объект, которому присвоен материал, который требуется скопировать.

**Material** Материал, который требуется скопировать.

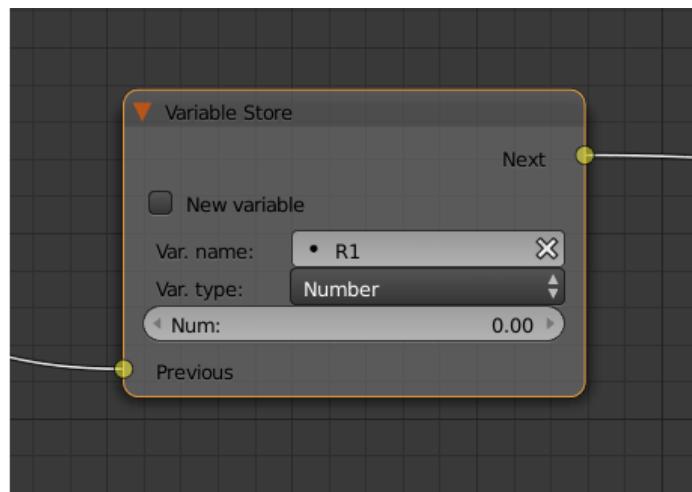
**Destination** Объект, которому требуется присвоить скопированный материал.

**Material** Материал, который будет заменен скопированным материалом.

## Operations

### Variable Store

Сохраняет числовое или строковое значение в переменную.



### Входные параметры

**Previous** Предыдущая нода.

### Выходные параметры

**Next** Следующая нода.

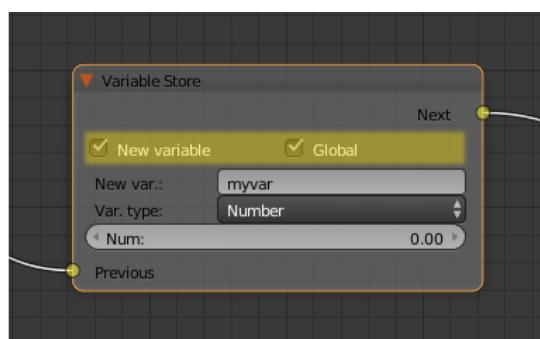
### Внутренние параметры

**Var. name.** Имя переменной. Может быть выбрано из списка переменных либо задано самостоятельно (если включен параметр **New variable**).

**Var. type** Тип переменной. Может иметь значения **Number** (численная переменная) и **String** (строковая переменная).

**New Variable** Если этот параметр активирован, имя переменной может быть задано самостоятельно (а не выбрано из списка переменных). Может применяться для обмена данными с сервером.

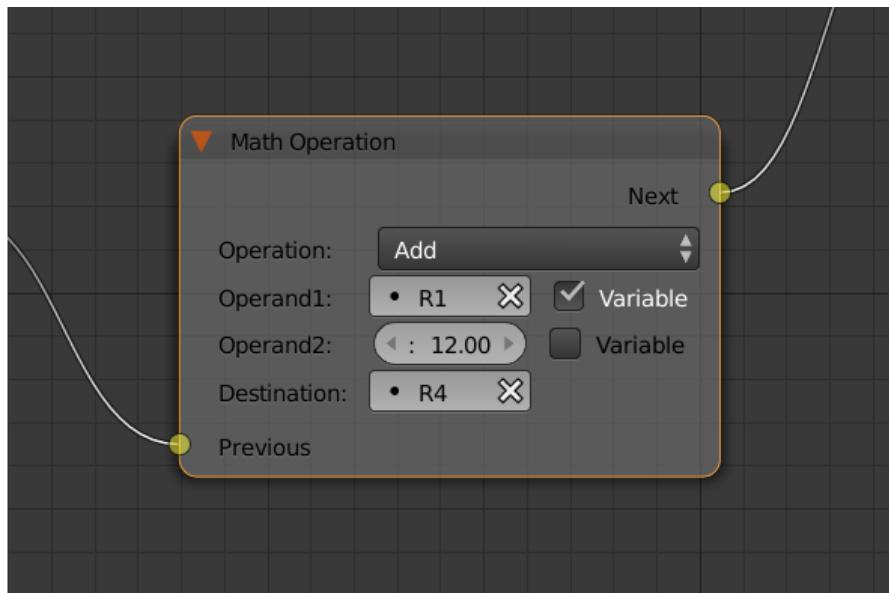
**Global** Включение этого параметра делает переменную глобальной. Доступен только если выбран параметр **New Variable**.



**Num./Str.** Численное либо строковое (в зависимости от значения параметра **Var. type**) значение, присваиваемое переменной.

## Math Operation

Выполнить математическую операцию и сохранить результат в регистр. Любой из параметров (операндов) может быть либо числовым значением, либо переменной.



### Входные параметры

*Previous* Предыдущая нода.

### Выходные параметры

*Next* Следующая нода.

### Внутренние параметры

**Operation** Математическая операция, производимая над operandами. Может иметь один из следующих типов:

- *Random* - генерация случайной величины больше первого операнда и меньше второго.
- *Add* - сумма operandов.
- *Multiply* - произведение operandов.
- *Subtract* - разность operandов.
- *Divide* - частное operandов.
- *Sin* - синус угла (в радианах), заданного первым operandом.
- *Cos* - косинус угла (в радианах), заданного первым operandом.
- *Tan* - тангенс угла (в радианах), заданного первым operandом.

- *ArcSin* - арксинус значения, заданного первым операндом.
- *ArcCos* - арккосинус значения, заданного первым операндом.
- *ArcTan* - арктангенс значения, заданного первым операндом.
- *Log* - логарифм первого операнда; в качестве базы используется второй операнд.
- *Min* - возвращает наименьший из двух операндов.
- *Max* - возвращает наибольший из двух операндов.
- *Round* - округляет первый operand.
- *Mod* - остаток от деления первого операнда на второй.
- *Abs* - значение первого операнда по модулю.

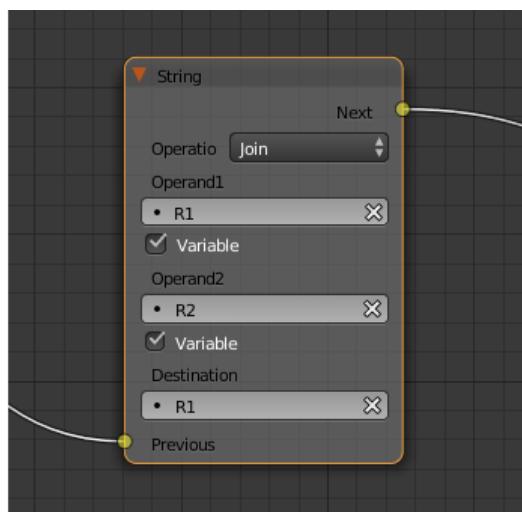
**Operand1** Первый operand. Может представлять собой численное значение либо ссылку на одну из переменных (в случае, если активирован параметр **Variable**).

**Operand2** Второй operand. Аналогичен первому.

**Destination** Переменная, в которую помещается результат операции.

## String Operation

Выполнить операцию над строковыми переменными и сохранить результат в переменную.



### Входные параметры

**Previous** Предыдущая нода.

## Выходные параметры

**Next** Следующая нода.

## Внутренние параметры

**Operation** Операция, которую нужно произвести над строками. Может иметь одно из следующих значений:

- *Join* - объединить первую строку со второй.
- *Find* - записывает в переменную позицию первого вхождения второй строки в первую. Если вхождений нет, в переменную записывается результат -1. Следует иметь в виду, что символы строки нумеруются, начиная с нуля, а не с единицы.
- *Replace* - найти содержимое второй строки в первой и заменить третьей.
- *Split* - разделить первую строку на две части, используя первое вхождение второй строки как маркер места разделения.
- *Compare* - сравнить первую строку со второй. Для работы этой операции требуется определить логическое условие. Если условие выполняется, в **Destination** будет записана единица, если нет - ноль.

**Condition** Логическое условие, по которому сравниваются строки. Этот параметр появляется только если параметр **Operation** имеет значение **Compare**. Аналогично параметру **Condition** ноды **Conditional Jump**.

**Operand1** Первая строка. Может быть задана в самой ноде или с помощью переменной.

**Operand2** Вторая строка. Аналогична первой.

**Operand3** Этот параметр появляется только если параметр **Operation** имеет значение **Replace**. Здесь задаётся строка, которой будет заменена вторая строка.

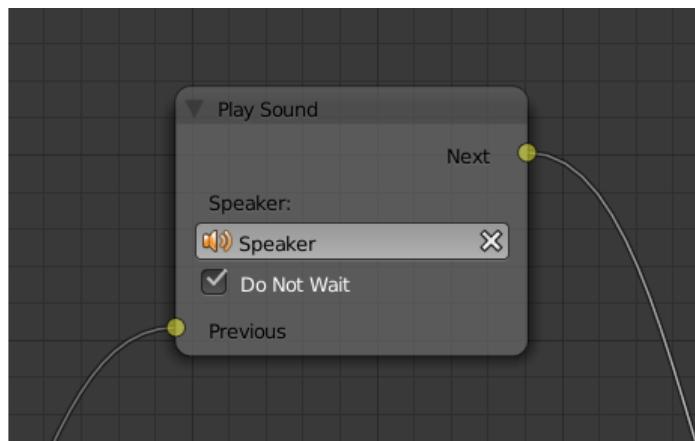
**Destination** Переменная, в которую будет сохранён результат операции.

**Destination2** Этот параметр появляется только в том случае, если параметр **Operation** имеет значение **Split**. Определяет переменную, в которую будет сохранена вторая половина разделённой строки.

## Sound

### Play Sound

Позволяет воспроизводить звук спикера.



### Входные параметры

**Previous** Предыдущая нода.

### Выходные параметры

**Next** Следующая нода.

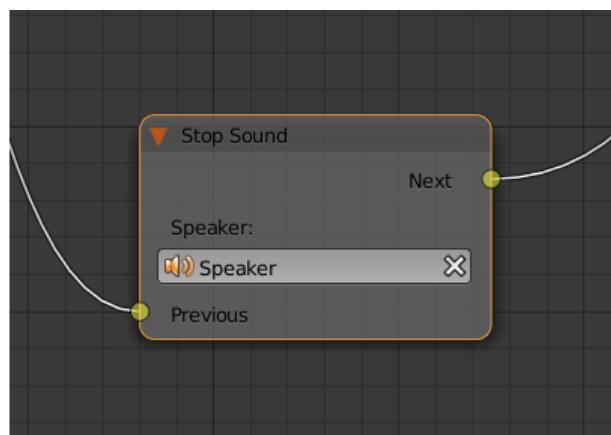
### Внутренние параметры

**Speaker** Источник звука, который требуется активировать.

**Do Not Wait** Если этот параметр активирован, управление будет передано следующей ноде сразу же после того, как началось воспроизведение звука. В противном случае это произойдёт только после того, как звук закончится.

## Stop Sound

Позволяет останавливать звук спикера.



## Входные параметры

*Previous* Предыдущая нода.

## Выходные параметры

*Next* Следующая нода.

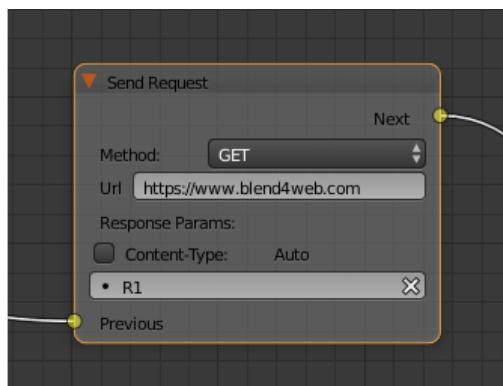
## Внутренние параметры

*Speaker* Источник звука, который нужно отключить.

# Network

## Send Request

Позволяет отправить HTTP GET запрос на указанный URL и поместить значения полей ответа в переменную.



## Входные параметры

*Previous* Предыдущая нода.

## Выходные параметры

*Next* Следующая нода.

## Внутренние параметры

**Method** Метод отправки запроса. Может иметь одно из двух значений:

- *GET* - отправляет запрос на получение данных с сервера.

- *POST* - отправляет данные на сервер.

Значение по умолчанию - *GET*.

**Url** Адрес, на который следует отправить запрос. По умолчанию имеет значение “<https://www.blend4web.com>”.

**Response Params** Задаёт переменную, в которую будут сохранены данные с сервера.

**Примечание:** Информация, полученная с сервера, должна быть в формате JSON:

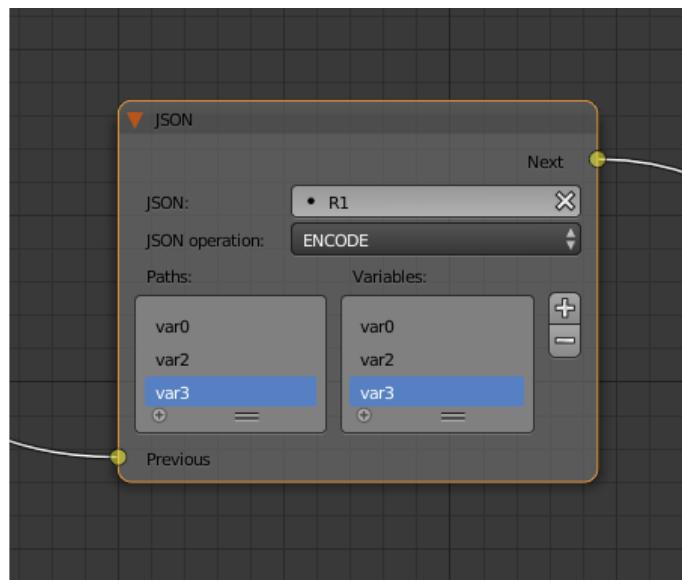
```
{
    "var0": 1,
    "var1": 10,
    "var2": 144
}
```

**Content-Type** Определяет формат запроса. Состоит из типа и подтипа, например: *text/plain*. Значение по умолчанию - *Auto*. Используется для переназначения заголовка HTTP-запроса.

**Request Params** Задаёт переменную, содержащую JSON-объект, который будет отправлен на сервер. Этот параметр доступен только если параметр **Method** имеет значение *POST*. Значение по умолчанию - *R1*.

## JSON

Используется для кодирования и декодирования сложных JSON-объектов.



## Входные параметры

*Previous* Предыдущая нода.

## Выходные параметры

*Next* Следующая нода.

## Внутренние параметры

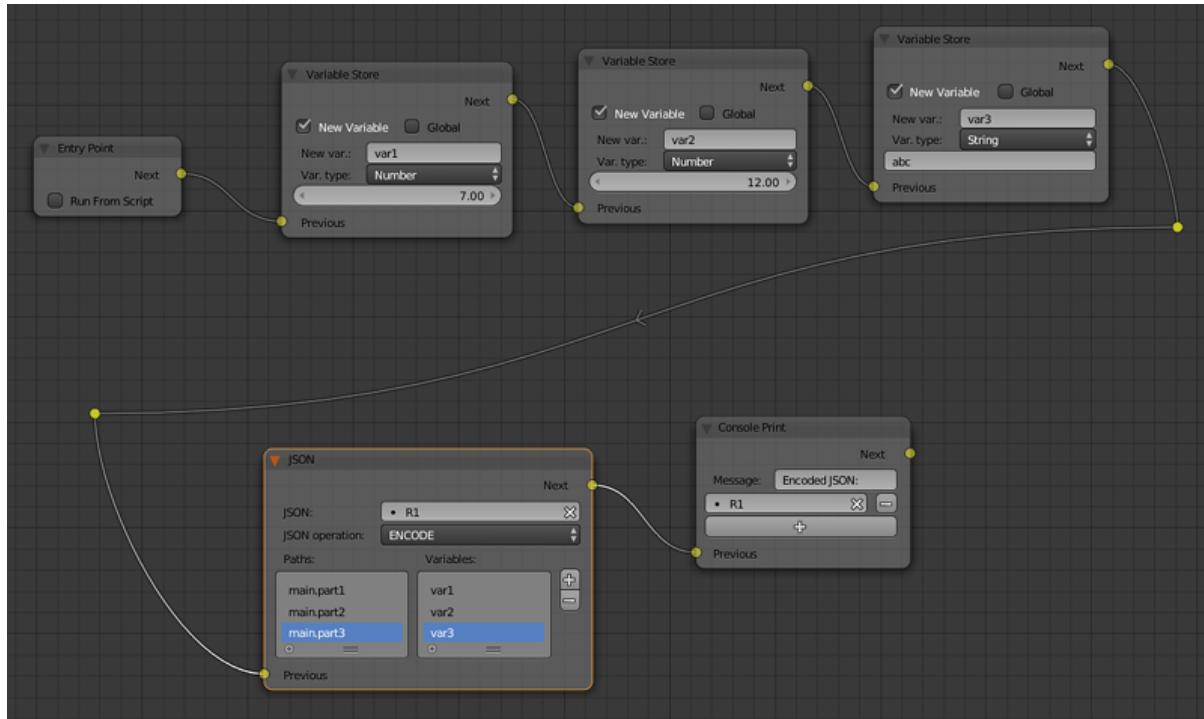
**JSON** Ссылка на JSON-объект, который нужно кодировать или декодировать. По умолчанию имеет значение R1.

**JSON Operation** Операция, которую нужно произвести над JSON-объектом, указанном в параметре **JSON**. Может иметь одно из двух значений: **ENCODE** кодирует объект, **PARSE** декодирует его. Значение по умолчанию - **ENCODE**.

**Paths** Список путей к переменным внутри объекта JSON. Пути используются для определения внутренней структуры объекта. Путь должен состоять из нескольких идентификаторов (разделённых точками), указывающих адрес поля JSON. Если имя фрагмента пути состоит только из цифр, такое имя интерпретируется как индекс массива. Пути создаются и удаляются вместе с переменными (в списке **Variables**); один путь всегда соответствует одной переменной. Список путей может использоваться как для кодирования, так и для декодирования объектов JSON. По умолчанию список пуст.

**Variables** Список переменных, которые будут использованы для кодирования или декодирования JSON-объекта (в зависимости от значения параметра **JSON Operation**). Имена и количество переменных могут быть настроены. Этот список также пуст по умолчанию.

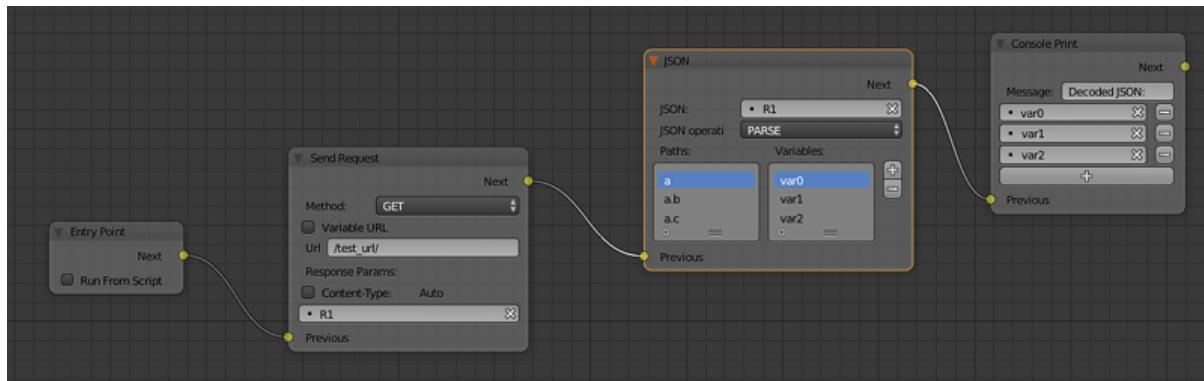
## Кодирование объекта JSON с помощью логической ноды JSON.



Логическое дерево на изображении выше кодирует объект JSON и сохраняет его в переменной R1. Закодированный объект JSON будет иметь следующий вид:

```
{
  "main": {
    "part1": 7,
    "part2": 12,
    "part3": "abc"
  }
}
```

## Декодирование объекта JSON с помощью логической ноды JSON.



Логическое дерево на изображении выше получает объект JSON с сервера, сохраняет его в переменной R1 и декодирует объект. Объект JSON имеет следующий вид:

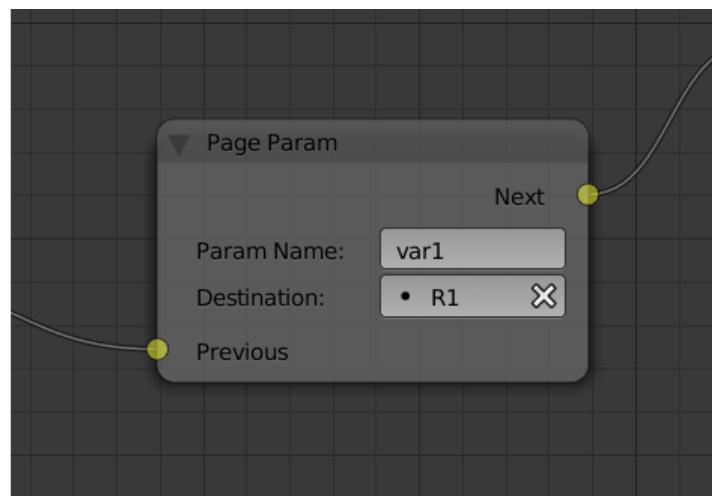
```
{
  "a": {
    "b": 17,
    "c": "abc"
  }
}
```

При декодировании объекта JSON в этом примере создаются три переменные (создавать их заранее не требуется) с именами `var0`, `var1` и `var2`, которые содержат разные части объекта JSON. В данном случае переменная `var1` будет иметь значение, равное 17, переменная `var2` будет иметь значение “abc”, а переменная `var0` будет содержать следующий фрагмент объекта JSON:

```
{
  "b": 17,
  "c": "abc"
}
```

## Page Param

Позволяет сохранить произвольный параметр веб-страницы в выбранной переменной.



### Входные параметры

**Previous** Предыдущая нода.

### Выходные параметры

**Next** Следующая нода.

## Внутренние параметры

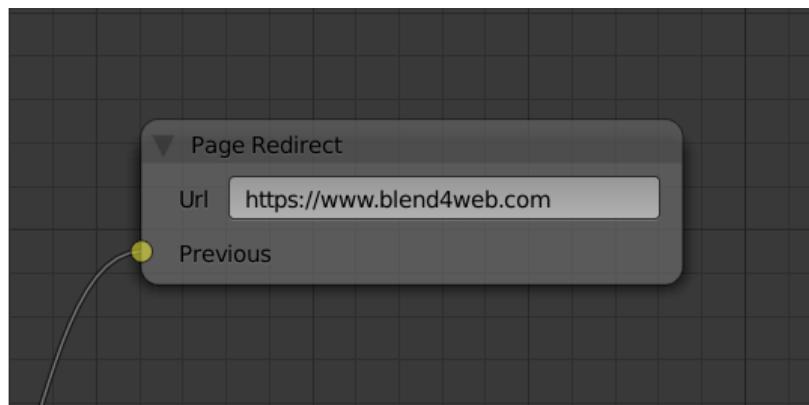
**Param Name** Имя параметра веб-страницы.

Если параметр с данным именем присутствует в URL-строке, то его значение будет сохранено в переменную, определенную полем **Destination**.

**Destination** Переменная, в которую следует сохранить параметр.

## Page Redirect

Служит для перенаправления на другие веб-страницы. По этой причине нода всегда находится в конце логической цепочки и не имеет выходных параметров.



### Входные параметры

**Previous** Предыдущая нода.

### Выходные параметры

Отсутствуют.

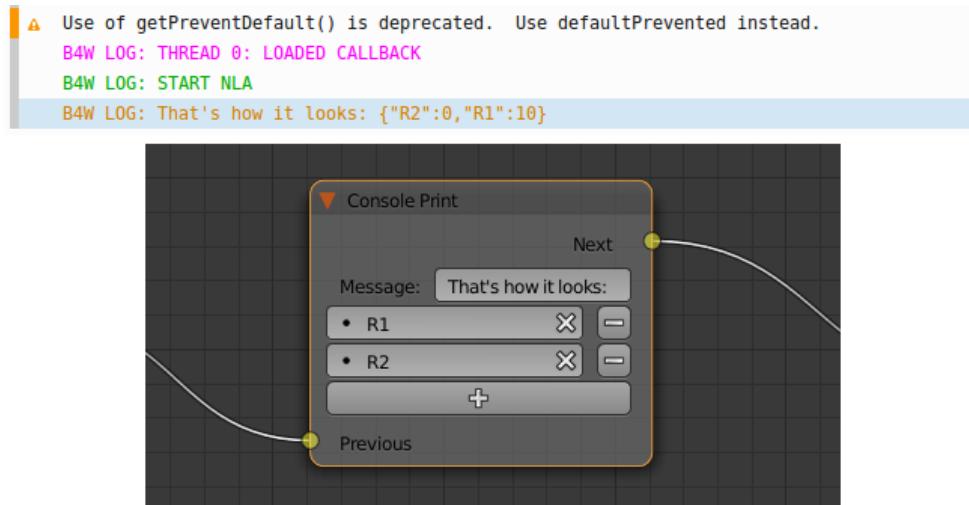
## Внутренние параметры

**Url** Адрес страницы, которая будет открыта. По умолчанию имеет значение “<https://www.blend4web.com>”.

## Debug

### Console Print

Позволяет выводить значения переменных и дополнительную информацию в консoli браузера. Может использоваться для отладки логики.



## Входные параметры

**Previous** Предыдущая нода.

## Выходные параметры

**Next** Следующая нода.

## Внутренние параметры

**Message** Сообщение, которые будет напечатано в консоли вместе со значениями переменных.

<имя переменной> Имя одной из переменных, значение которой нужно вывести в консоли. По умолчанию нода **Console Print** имеет один параметр такого типа, также имеется возможность создавать новые и удалять уже существующие (нода может и не иметь ни одного параметра этого типа).

## Deprecated

### Select (Deprecated)

---

**Примечание:** Устарел! Не рекомендуется использовать! Взамен следует использовать ноду **Switch 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.

## Select & Play Timeline (Deprecated)

---

**Примечание:** Устарел! Не рекомендуется использовать! Взамен следует использовать комбинацию нод `Select` и `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.

## Select & Play Animation (Deprecated)

---

**Примечание:** Устарел! Не рекомендуется использовать! Взамен следует использовать комбинацию нод `Select` и `Play Timeline`.

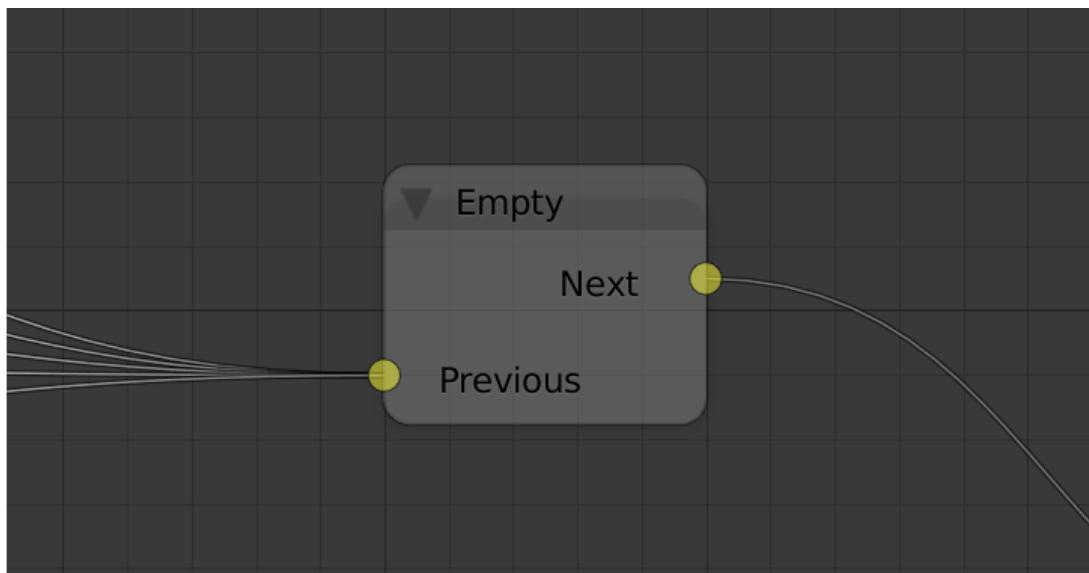
---

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 Animation`, если же был выбран любой другой объект - немедленно передать управление следующей ноде.

## Layout

### Empty

Простая нода, не выполняющая никаких действий кроме передачи управления следующей ноде. Может применяться для объединения нескольких логических нитей в одну или для того, чтобы сделать логическое дерево более простым для понимания.



### Входные параметры

*Previous* Предыдущая нода (либо несколько нод).

### Выходные параметры

*Next* Следующая нода.

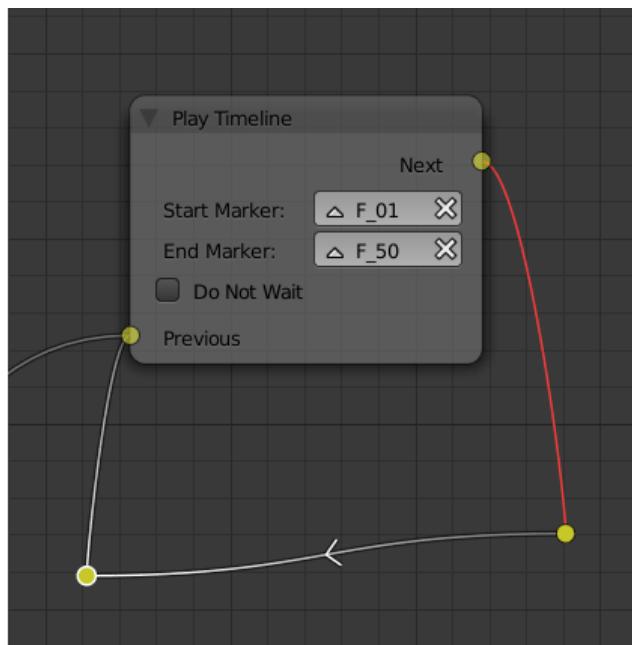
### Внутренние параметры

Отсутствуют.

## Reroute

В редакторе логики также присутствуют элементы **Reroute** - специальные ноды, который не выполняют никаких операций, а только передают управление следующей ноде (или следующему элементу **Reroute**). Они могут применяться для создания циклических структур или для того, чтобы сделать логическую цепочку более понятной и лёгкой для восприятия.

В отличие от ноды **Empty**, через элемент **Reroute** может проходить только одна логическая нить.



---

**Примечание:** Выходной параметр не может быть подключён к входному параметру той же ноды. В случае, когда это требуется сделать (например, для того, чтобы зациклить выполнение какой-либо операции), следует использовать Reroute.

---

## 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).

# Освещение, тени и фон

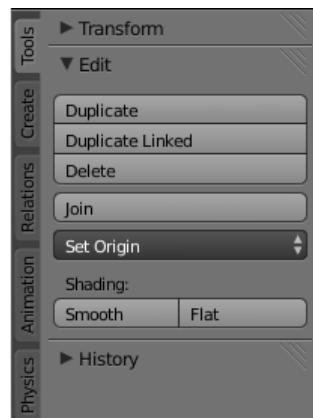
---

## Содержание

- *Освещение, тени и фон*
  - *Типы затенения*
  - *Освещение от источников света*
    - \* *Типы источников света*
    - \* *Настройка источников света*
  - *Освещение от окружающей среды*
    - \* *Активация*
    - \* *Настройка*
    - \* *Использование карты окружения*
  - *Тени*
    - \* *Активация*
    - \* *Настройка*
  - *Цвет фона*

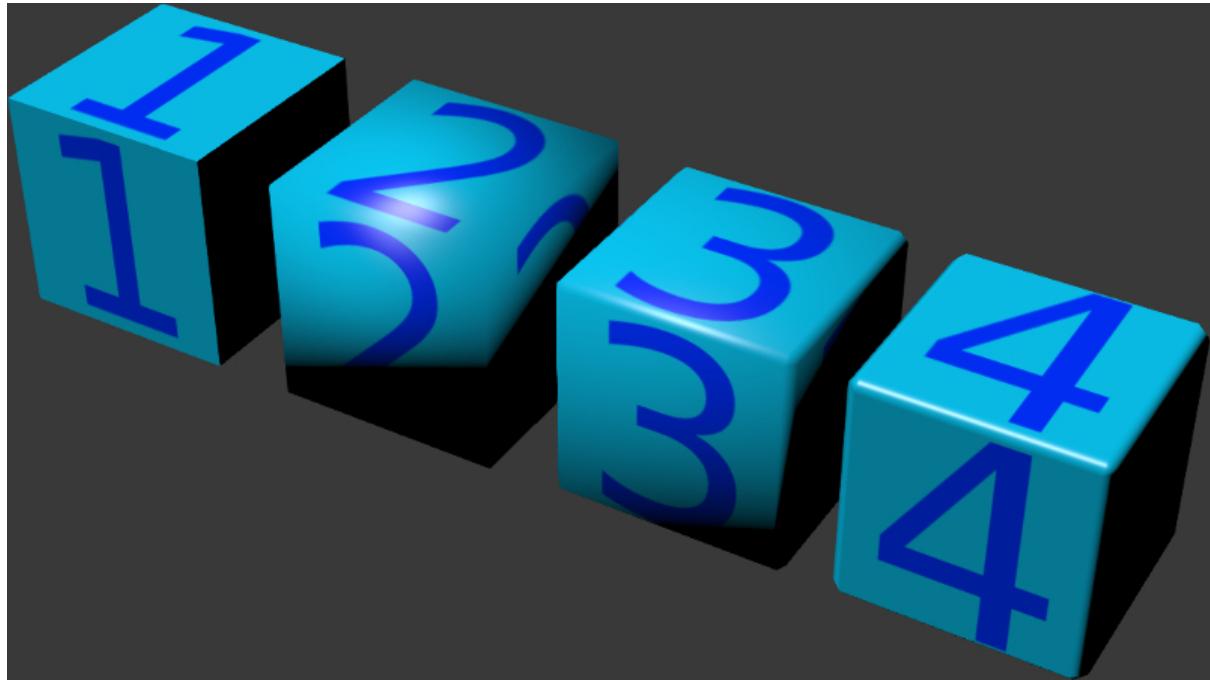
## Типы затенения

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

## Освещение от источников света

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

### Типы источников света

Поддерживаются источники света следующих типов:

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

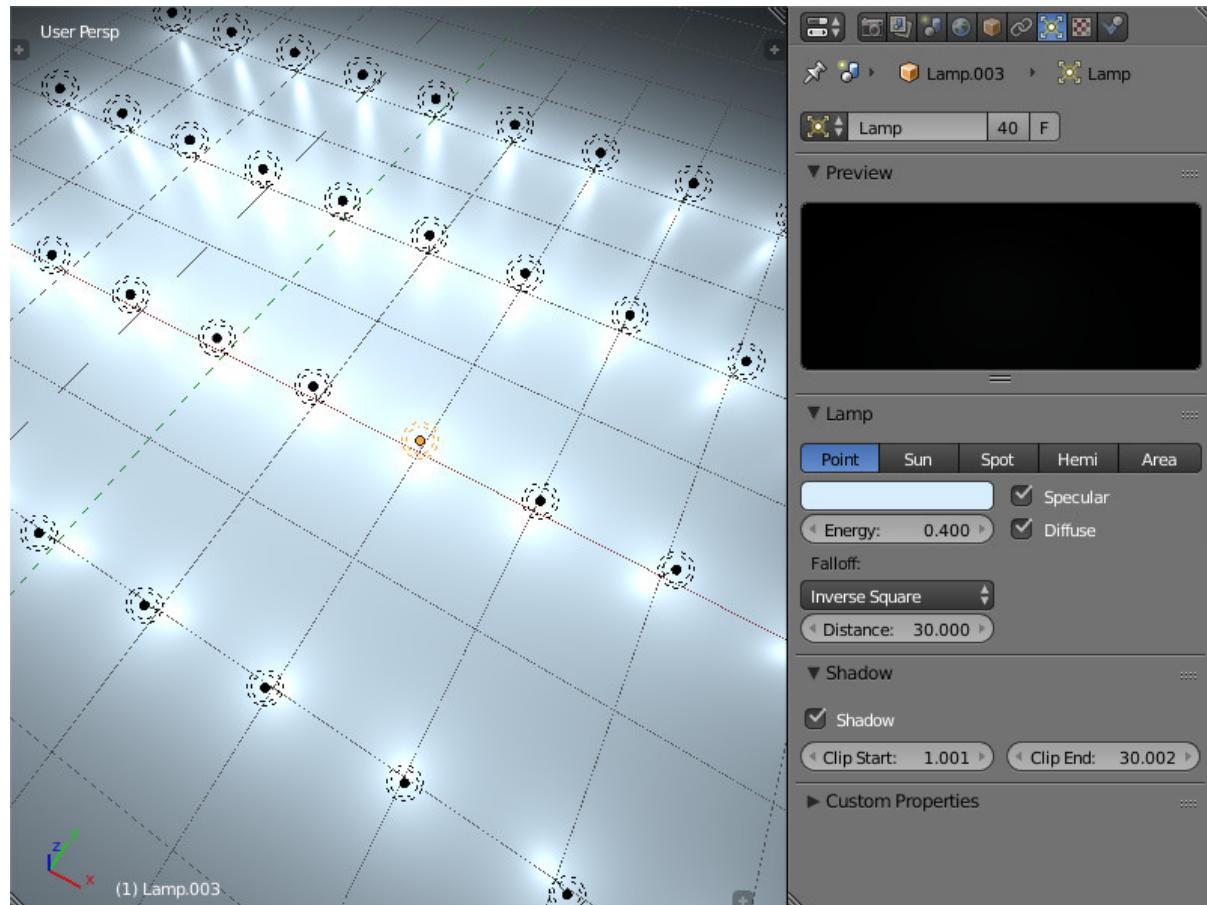
**Sun** “Солнце”. Свет распространяется из бесконечной плоскости прямолинейно в одном направлении, без затухания.

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

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

### Настройка источников света

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



**Цвет** 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 30.0.

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

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

**Shadow > Clip Start** Расстояние между источником света и ближней плоскостью отсечения. Используется при генерации теней. Значение по умолчанию - 1.001.

**Shadow > Clip End** Расстояние между источником света и дальней плоскостью отсечения. Используется при генерации теней. Значение по умолчанию - 30.002.

## Освещение от окружающей среды

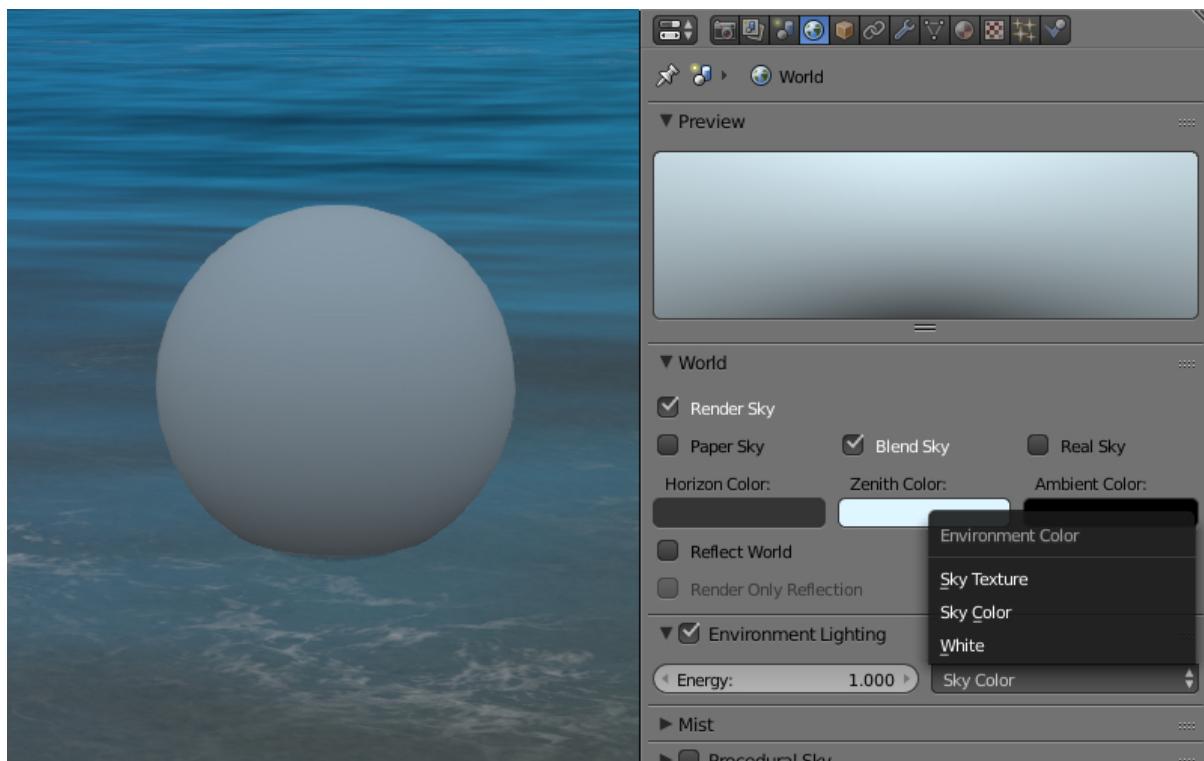
Движок поддерживает 3 метода симуляции рассеянного освещения от окружающих объектов.

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.

Обратите внимание, что освещение от окружающей среды рассчитывается по упрощённой модели, в которой объекты не затеняют друг друга.”

## Активация

Включить опцию Environment Lighting во вкладке World.



## 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.

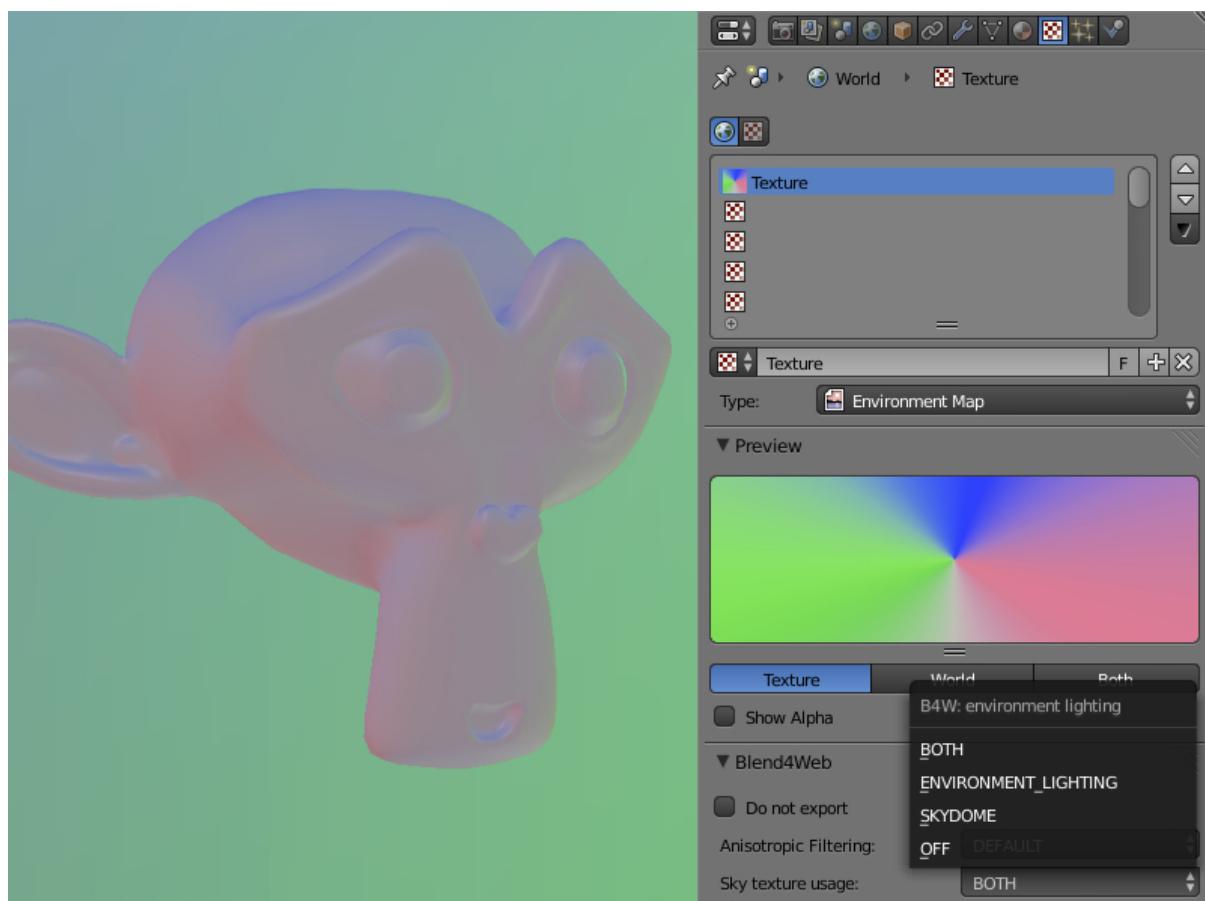
**World > Horizon Color и World > Zenith Color** Если выбрана полусферическая модель рассеянного освещения Sky Color, цвет горизонта и цвет зенита задаются цветоподборщиками World > Horizon Color (цвет горизонта) и World > Zenith Color (цвет зенита). При выборе цвета рекомендуется активировать опцию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

## 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*).

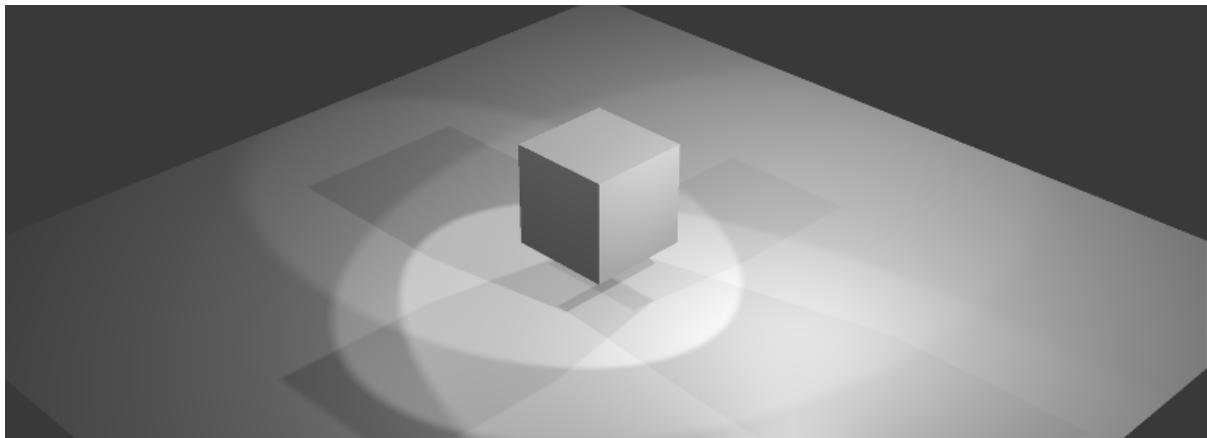


## Тени

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).

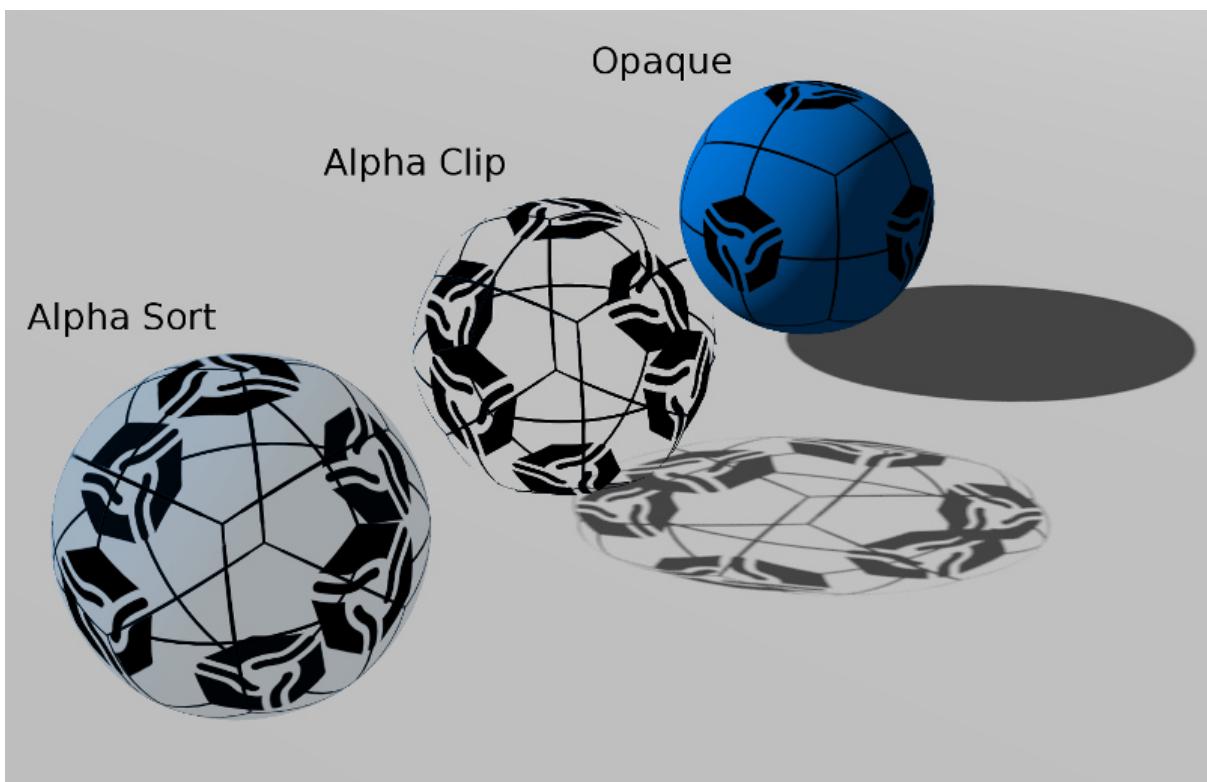
Генерировать тени могут не более четырёх (или трёх, если включен SSAO) источников света одновременно. Если источников света больше (и у всех включён параметр Shadow), будут видны тени только от четырёх из них.



## Активация

1. 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.

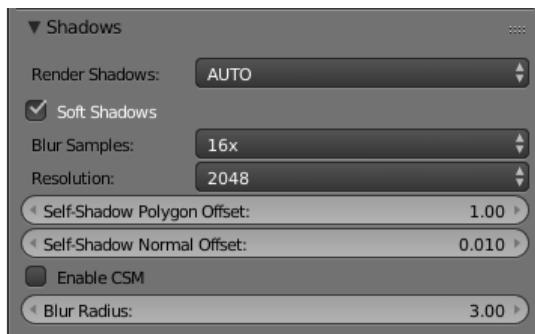


## Настройка

**Направление** В случае наличия нескольких источников света рекомендуется указать, какой именно источник света будет использоваться для расчета падающих теней, включив опцию **Shadow > Shadow** 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:



**Render Shadows** Включает и выключает рендеринг теней. Возможные значения: **ON**, **OFF** и **AUTO**. Значение по умолчанию - **AUTO**.

**Soft Shadows** Включает размытие карт теней. По умолчанию включено.

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

**Blur Samples** Количество сэмплов, используемых для размытия карты теней. Возможные значения - 4x, 8x и 16x (значение по умолчанию).

**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: 0.010.

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.



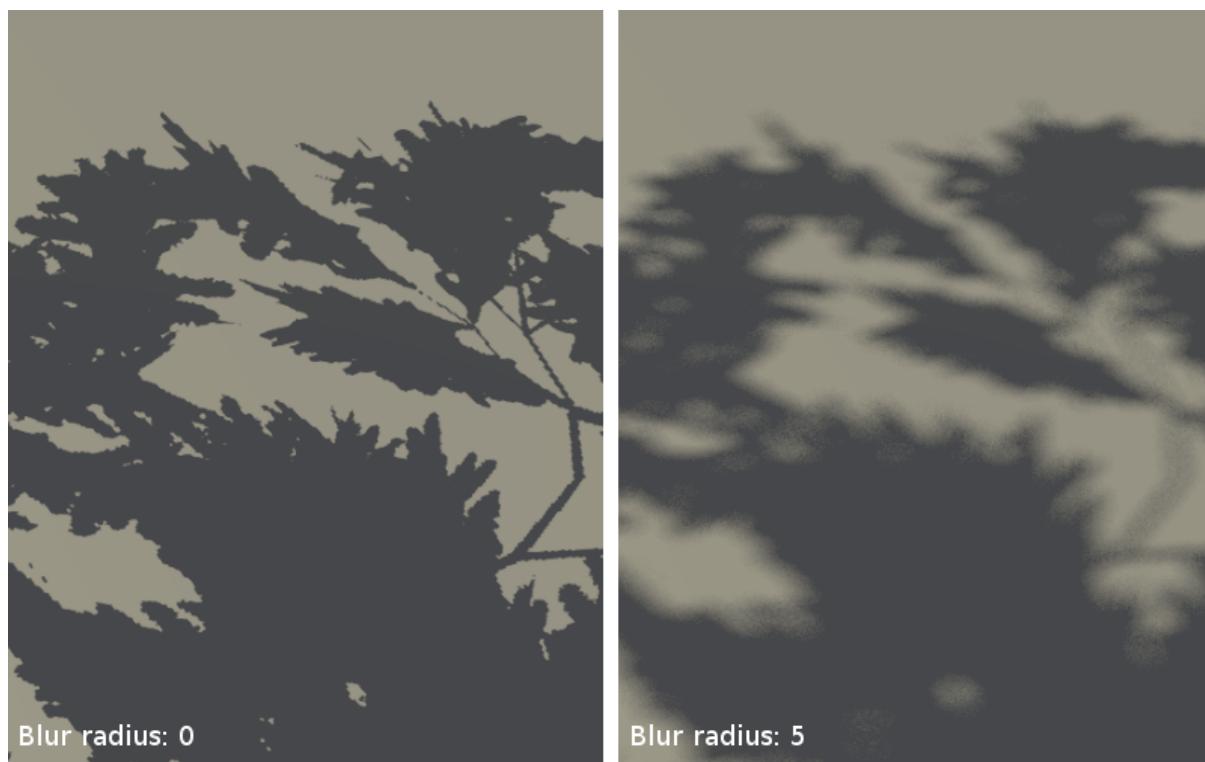
**Примечание:** Тени для точечных (**Point**) источников света генерируются так же, как и для направленных (**Spot**), и падают только в одном направлении, определяемом параметром **Rotation**.

**Enable CSM** Включение каскадной модели теней; открывает доступ к расширенным настройкам. Отключено по умолчанию. Не работает, если опция **Shadow** включена более чем для одного источника света. Для источников света типа **Point** и **Spot** поддерживается только один каскад теней.

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

---

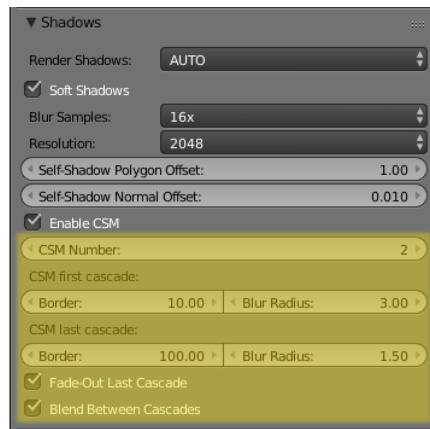
**Примечание:** Данные настройки поддерживаются только для источников освещения типа Sun. Для других источников каскады автоматически отключаются.

---

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.

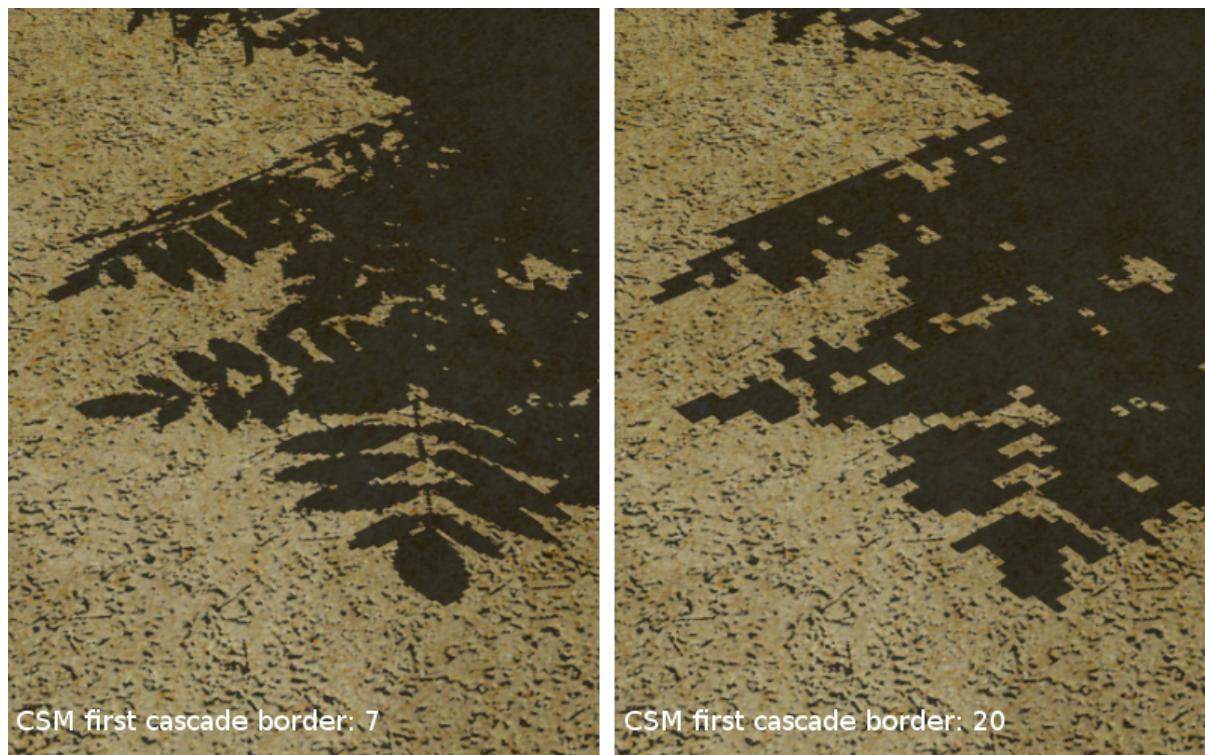
**CSM Last Cascade Border** Размер последнего каскада. Значение по умолчанию: 100.0.

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.

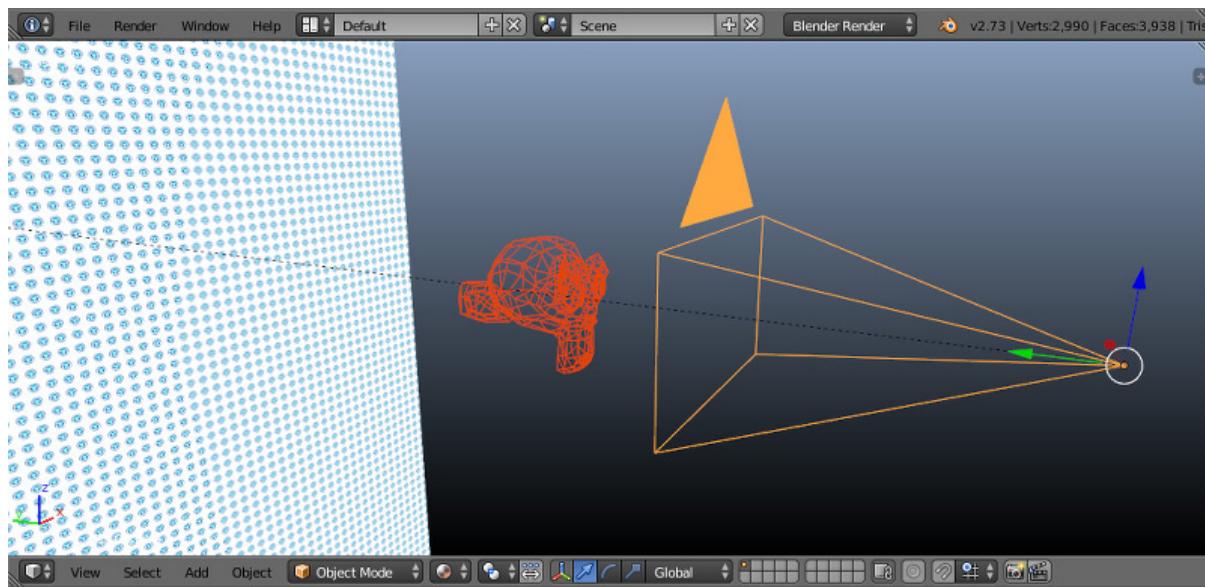
**Blend Between Cascades** Сглаживание границ между каскадами. По умолчанию включено.



## 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*

---

## Спецэффекты

---

### Содержание

- *Спецэффекты*
  - *Размытие при движении*
  - *Глубина резкости камеры*
  - *Взаимное затенение*
  - *Сумеречные лучи*
  - *Эффект засветки ярких деталей*
  - *Подсветка контура (outlining)*
  - *Свечение (glow)*
  - *Сглаживание*

### 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.



## Активация

Активировать панель **Motion Blur** во вкладке **Render**.

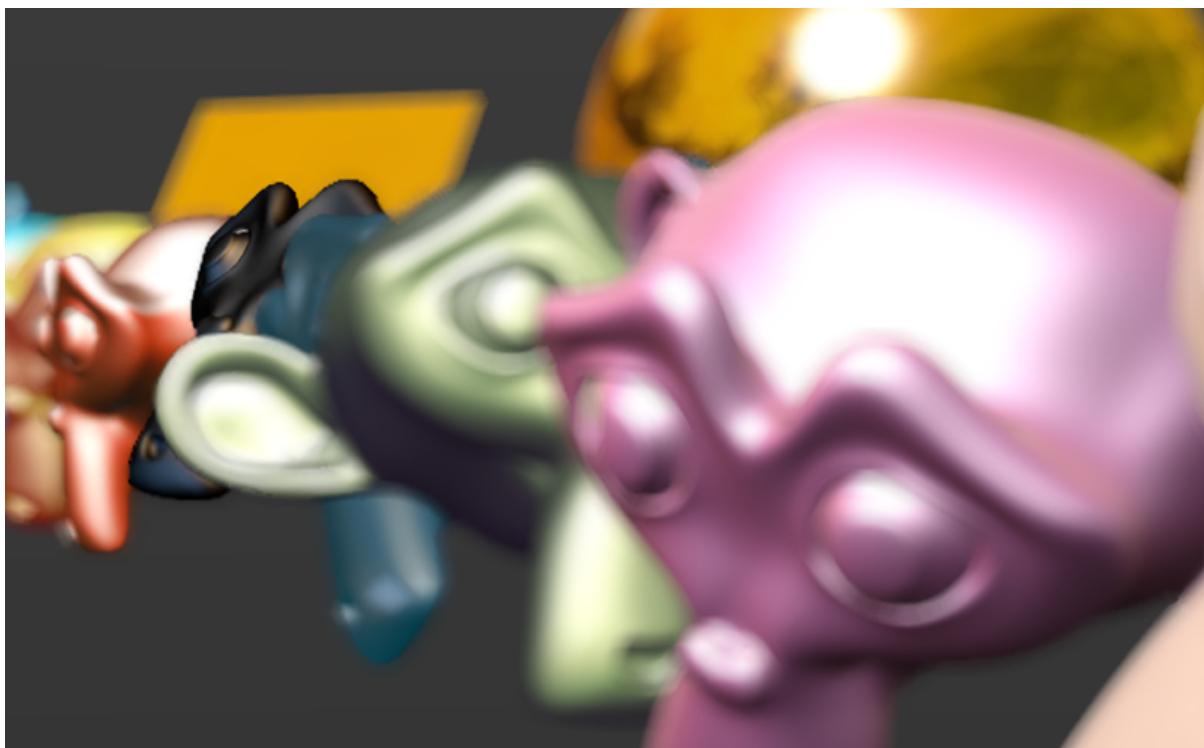
## Дополнительные настройки

**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.

## Глубина резкости камеры

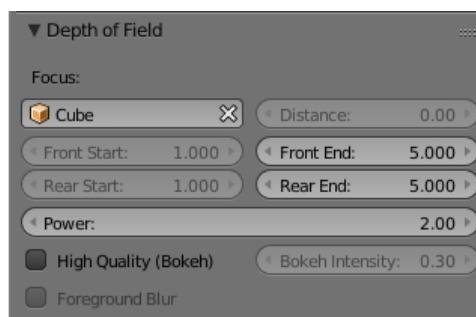
Эффект глубины резкости камеры (depth of field, DOF) акцентирует внимание зрителя на части сцены. Проявляется в размытии изображения ближе и дальше от фокуса камеры.



## 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.

## Дополнительные настройки



**Focus** Задаёт объект, на котором фокусируется камера. Это поле пусто по умолчанию.

**Distance** Задаёт расстояние фокусировки. Этот параметр доступен только в том случае, если не установлен параметр **Focus** (соответствующее поле пусто). По умолчанию расстояние до точки фокуса равно нулю.

**Front Start** Задаёт расстояние (в метрах) от фокуса до ближней к камере плоскости, за которой начинает проявляться эффект размытия. Значение по умолчанию 1.0.

Этот параметр доступен только при включённой опции **High Quality (Bokeh)**.

**Front End** Задаёт расстояние (в метрах) от фокуса до ближней к камере плоскости, за которой происходит полное размытие (сила размытия задаётся параметром **Power**). Значение по умолчанию 5.0.

**Rear Start** Задаёт расстояние (в метрах) от фокуса до дальней от камеры плоскости, за которой начинает проявляться эффект размытия. Значение по умолчанию 1.0.

Этот параметр доступен только при включённой опции **High Quality (Bokeh)**.

**Rear End** Задаёт расстояние (в метрах) от фокуса до дальней от камеры плоскости, за которой происходит полное размытие (сила размытия задаётся параметром **Power**). Значение по умолчанию 5.0.

**Power** Сила размытия. Этот параметр может принимать значения от 0.1 до 10. Значение по умолчанию - 2.0.

**High Quality (Bokeh)** Включает высококачественный рендеринг эффекта глубины резкости. Активация этого параметра:

- Включает эффект боке и опцию **Bokeh Intensity**, которая может применяться для настройки силы этого эффекта.
- Включает параметры **Front Start** и **Rear Start**, предназначенные для установки мягкого перехода между расстоянием от камеры, за которым эффект размытия начинает проявляться, и расстоянием, за которым он достигает максимальной силы.
- Включает параметр **Foreground Blur**, описанный ниже.

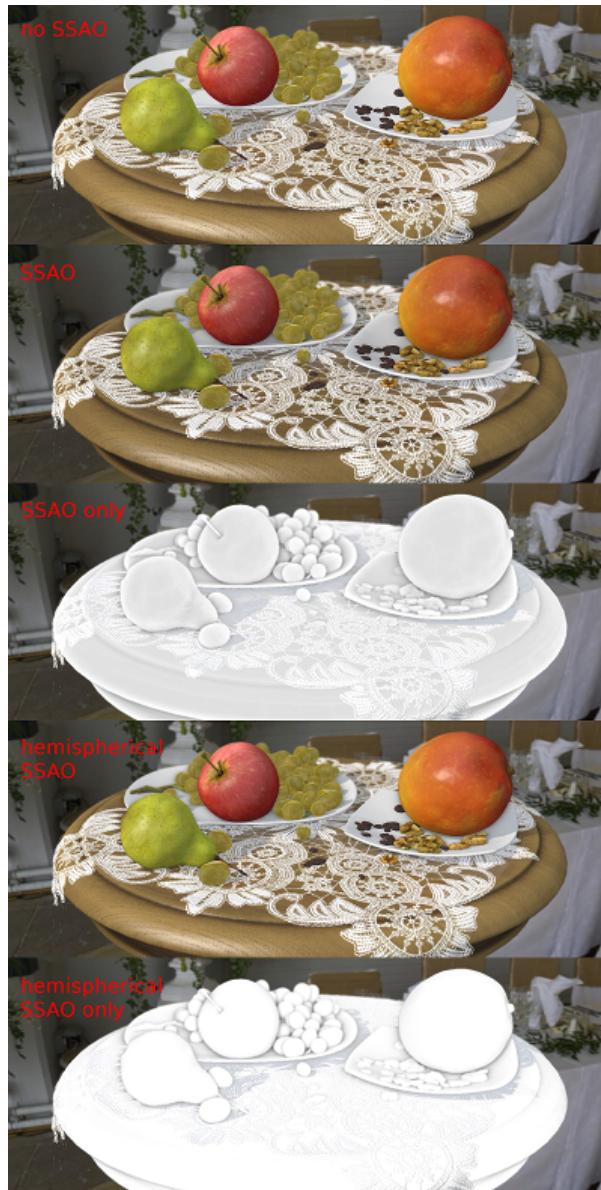
По умолчанию этот параметр деактивирован.

**Bokeh Intensity** Устанавливает силу эффекта боке. Значение может меняться от 0 до 1, значение по умолчанию - 0.3.

**Foreground Blur** При включении этого параметра движок размывает границы объектов, находящихся на переднем плане, увеличивая качество эффекта размытия за счёт некоторого снижения производительности. По умолчанию отключен.

## 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.



## Активация

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

## 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 сэмплов на их вес в размытии. Используется при активированном параметре Use Blur Depth Test. Значение по умолчанию 1.0.

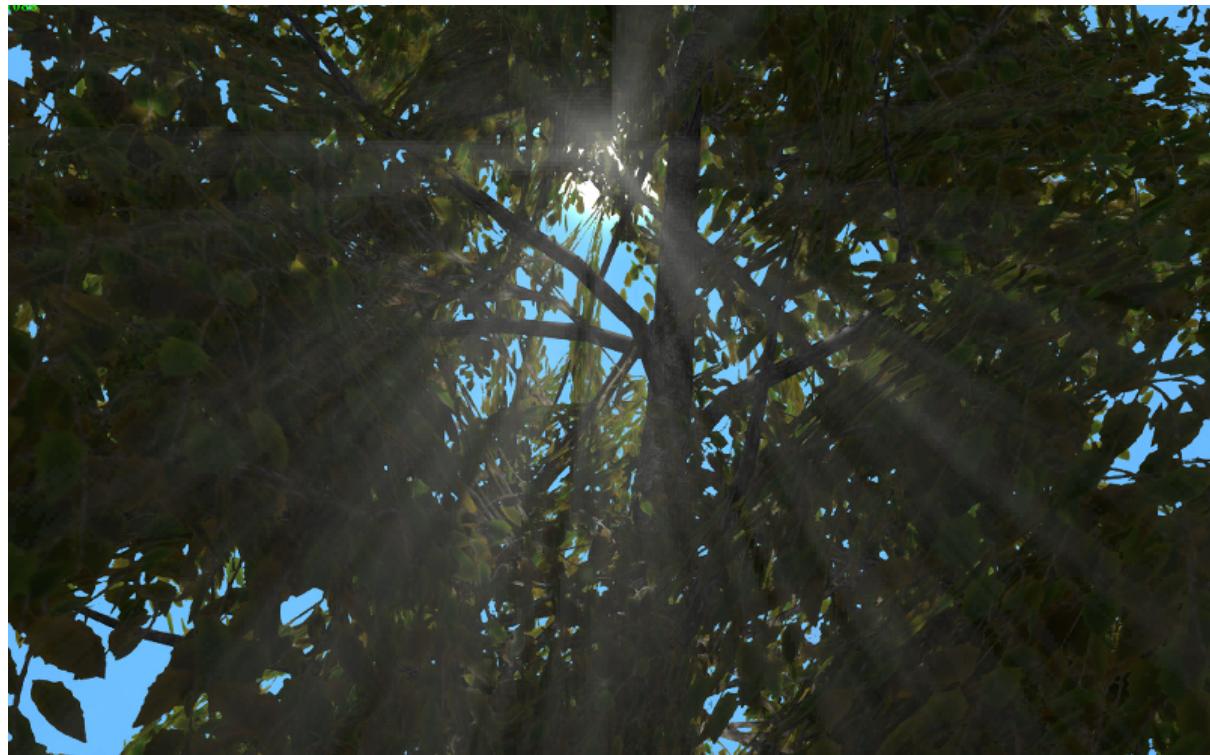
**Influence** Степень проявленности эффекта взаимного затенения. Значение по умолчанию 0.7.

**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.

## Сумеречные лучи

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



## Активация

Активировать панель God Rays во вкладке Render.

## Дополнительные настройки

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

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

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

## Эффект засветки ярких деталей

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



## Активация

Активировать панель Bloom во вкладке Render.

## Дополнительные настройки

**Use Adaptive** Включает расчёт адаптивной средней яркости. По умолчанию включено.

**Intensity** Интенсивность засветки. По умолчанию равна 1.0.

**Blur** Степень размытия засветки. По умолчанию равна 4.0.

**Edge Luminance** Границное значение относительной яркости элемента, выше которого начинает проявляться эффект засветки. По умолчанию равно 1.0.

## Подсветка контура (outlining)

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



## 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 Selection поле Enable установлено в состояние ON или AUTO.

## 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.

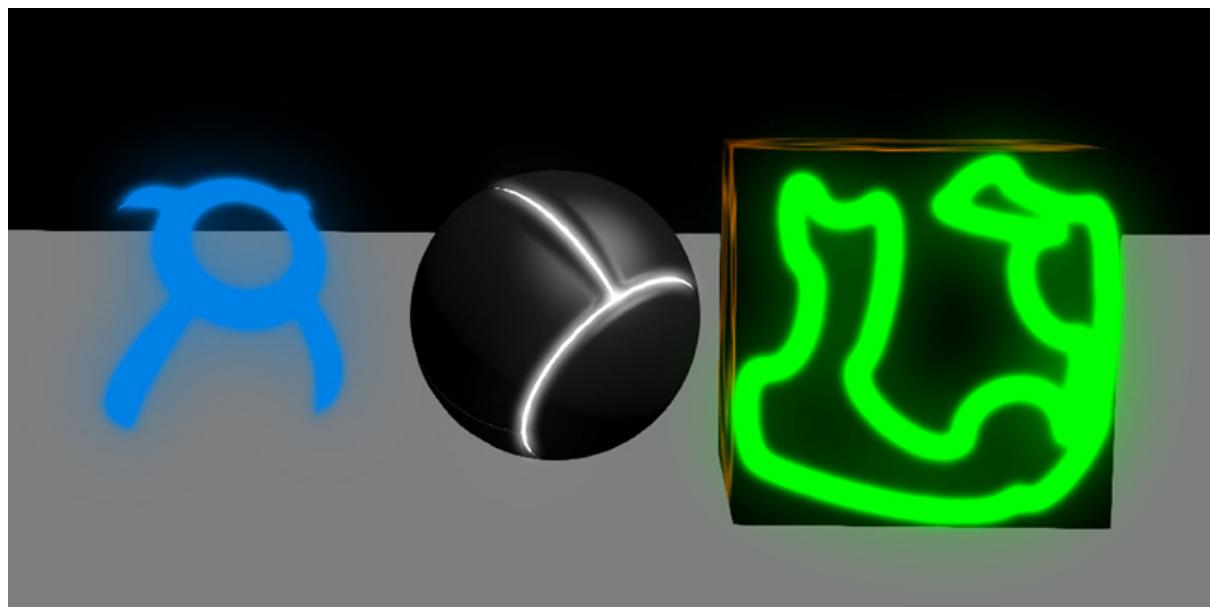
На панели Render > Object Outlining:

**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.

## Свечение (glow)

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



## 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.

## 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.

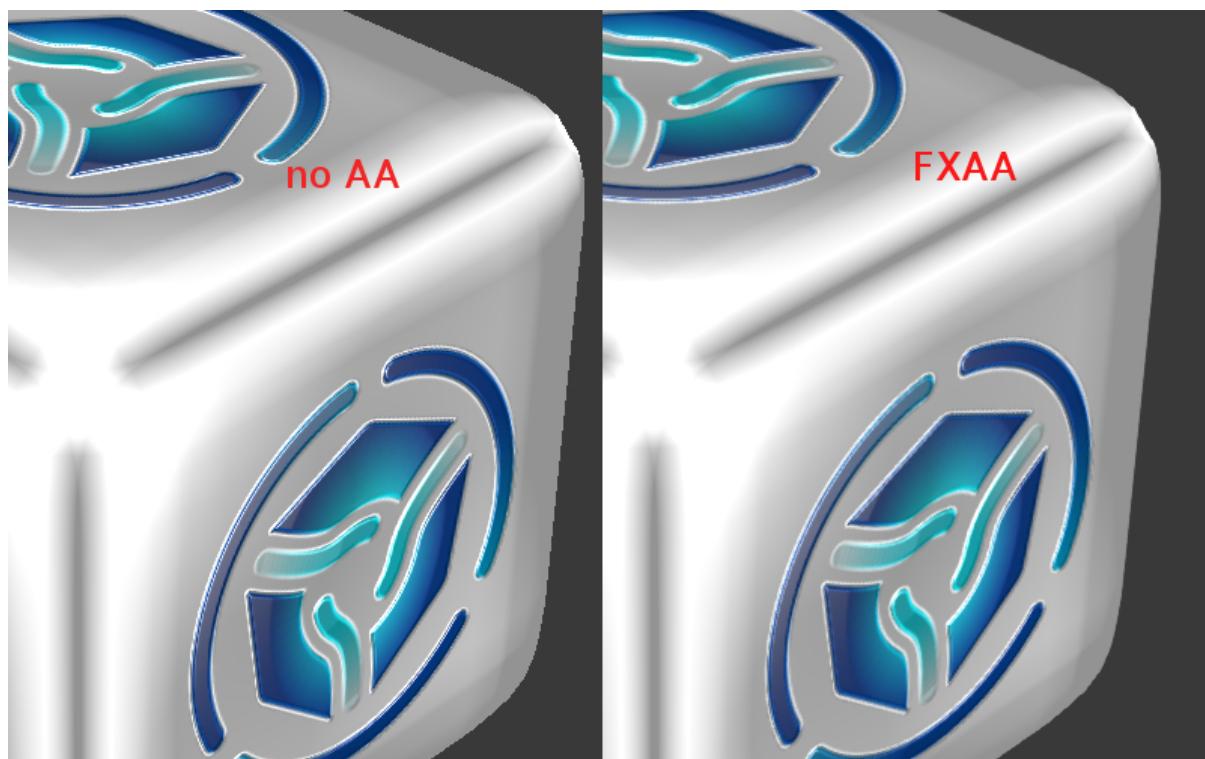
*Large Mask: Intensity* Интенсивность свечения по большой маске. Значение по умолчанию 2.0.

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

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

## Сглаживание

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



## 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*.

## 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).

## Стереоизображение

### Содержание

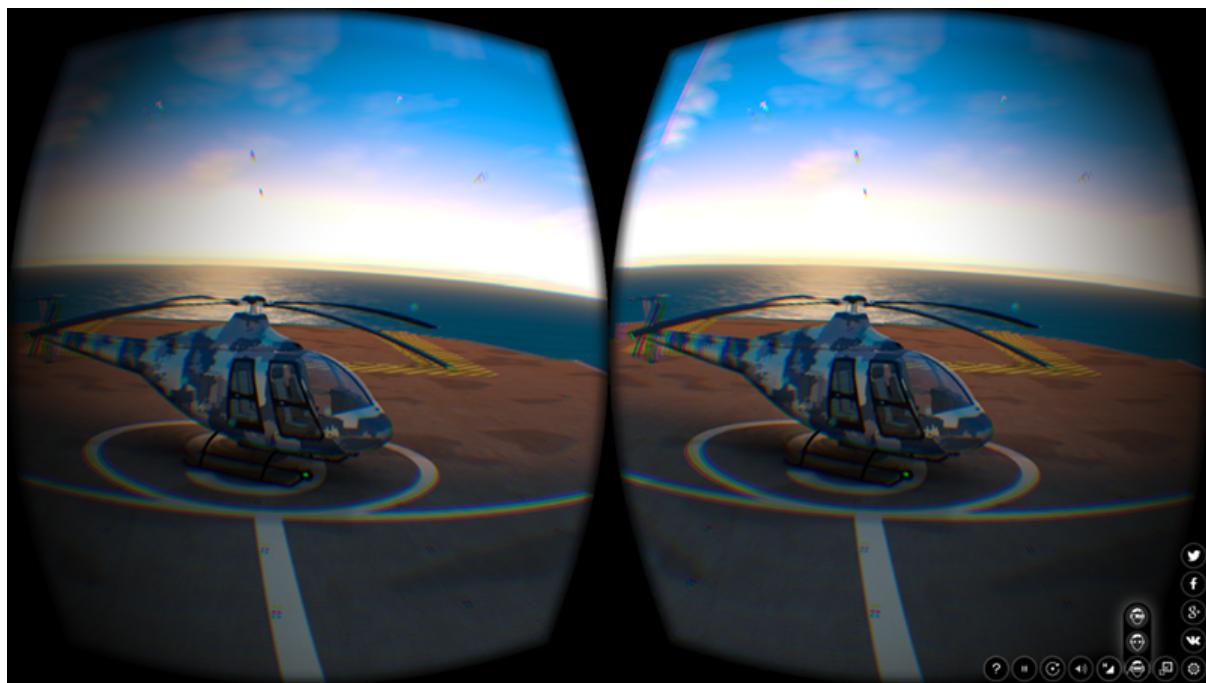
- Стереоизображение
  - Активация
  - Дополнительные настройки
  - Настройки HMD
    - \* Profile
    - \* Настройки рендеринга
    - \* Другие инструменты управления

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

Blend4Web поддерживает два стереорежима - анаглифное изображение и HMD (Head-mounted display, шлем виртуальной реальности). Анаглиф:



HMD:



HMD - экспериментальная возможность, она работает только с камерами типа Eye.

## Активация

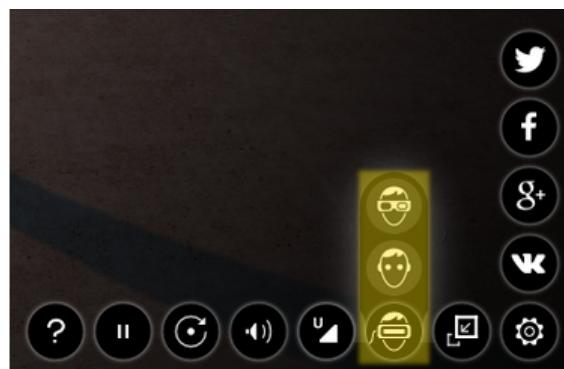
Для работы стереорежима HMD нужно установить [runtime-утилиту](#) от Oculus (поддерживаются две версии: *chromium\_webvr\_v1\_win64.7z* в корневой директории и версия в папке *Deprecated API*). Версии для Windows и macOS присутствуют на сайте Oculus в бинарном формате, версию утилиты для Linux пользователю понадобится собирать из исходников самостоятельно.

На сегодняшний день HMD поддерживается в экспериментальных сборках Chromium и в nightly build Firefox.

Настройки Chromium.

Настройки Firefox.

Чтобы включить стереорежим, нужно выбрать соответствующий пункт в настройках приложения, в третьем столбце справа, как показано на иллюстрации.



Для корректной работы рекомендуется включить полноэкранный режим.

## Дополнительные настройки

Отсутствуют.

## Настройки HMD



Эта группа настроек позволяет пользователю регулировать различные параметры для максимально комфортного просмотра 3D-сцены или приложения на своём VR-устройстве. Получить доступ к настройкам можно двумя способами:

Во-первых, с помощью *панели Stereo View* в *Просмотрщике сцен Blend4 Web*.

Во-вторых, с использованием в приложении метода `show` из модуля `hmd_conf`, для чего требуется написание программного кода. Примеры использования этого метода можно увидеть в приложениях `viewer.js` и `webplayer.js` (в объекте `m_hmd_conf`).

## Profile

Позволяет пользователю выбрать профиль для определённого устройства. На сегодняшний день поддерживаются следующие профили:

- Custom
- Cardboard (2014)
- Cardboard (2015)

По умолчанию выбран профиль *Custom*.

## Настройки рендеринга

***Tray to lens-center distance*** Задаёт расстояние между корпусом телефона и линзами камеры.

Значение может меняться от 0 до 50. Значение по умолчанию - 32.

***Interpupillary distance*** Задаёт расстояние (в миллиметрах) между зрачками глаз пользователя.

Значение может меняться от 0 до 100. Значение по умолчанию - 64.

***Screen to lens distance*** Задаёт расстояние между экраном телефона и линзами камеры.

Значение может меняться от 0 до 100. Значение по умолчанию - 50.

***Screen height*** Высота экрана.

Значение может меняться от 0 до 150. Значение по умолчанию - 63.

***Screen width*** Ширина экрана.

Значение может меняться от 0 до 200. Значение по умолчанию - 107.

***Bevel width*** Ширина рамки вокруг изображения.

Значение может меняться от 0 до 20. Значение по умолчанию - 3.

***Distortion coefficients*** Используются для компенсации искажений, вносимых в изображение линзами VR-устройства.

Оба значения могут меняться от 0 до 1. Значения по умолчанию: 0.34 для первого параметра и 0.55 для второго.

## Другие инструменты управления

***Save*** Сохраняет настройки.

***Cancel*** Закрывает окно настроек без сохранения изменений.

***Reset*** Восстанавливает настройки по умолчанию.

## Текстуры

### Содержание

- *Текстуры*
  - *Типы текстур*
  - *Общие настройки*
  - *Диффузная текстура (diffuse map)*
  - *Карта бликов (specular map)*
  - *Карта нормалей (normal map)*
  - *Карта высот (height map). Parallax mapping*
  - *Карта смешивания (stencil map)*
  - *Видео-текстуры*
  - *Карта окружения (environment map)*
  - *Карта зеркального отражения (mirror map)*
  - *Текстура неба (skydome)*
  - *Особые типы текстур*

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*.

## Типы текстур

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

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

В этом случае текстура задаётся с помощью растрового изображения или видеофайла. Поддерживаются следующие графические форматы:

- .PNG
- .JPG

Может использоваться в следующих случаях:

- *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*

# Общие настройки

**Размер** Размер растров 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 (т.е. формата текстур DDS) 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.

---

## Диффузная текстура (diffuse map)

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

### Активация

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

### 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*.

## Карта бликов (specular map)

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

### Активация

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

### 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.

## Карта нормалей (normal map)

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

### Активация

Выставить параметр *Image > Color Space* в значение *Non-Color*.

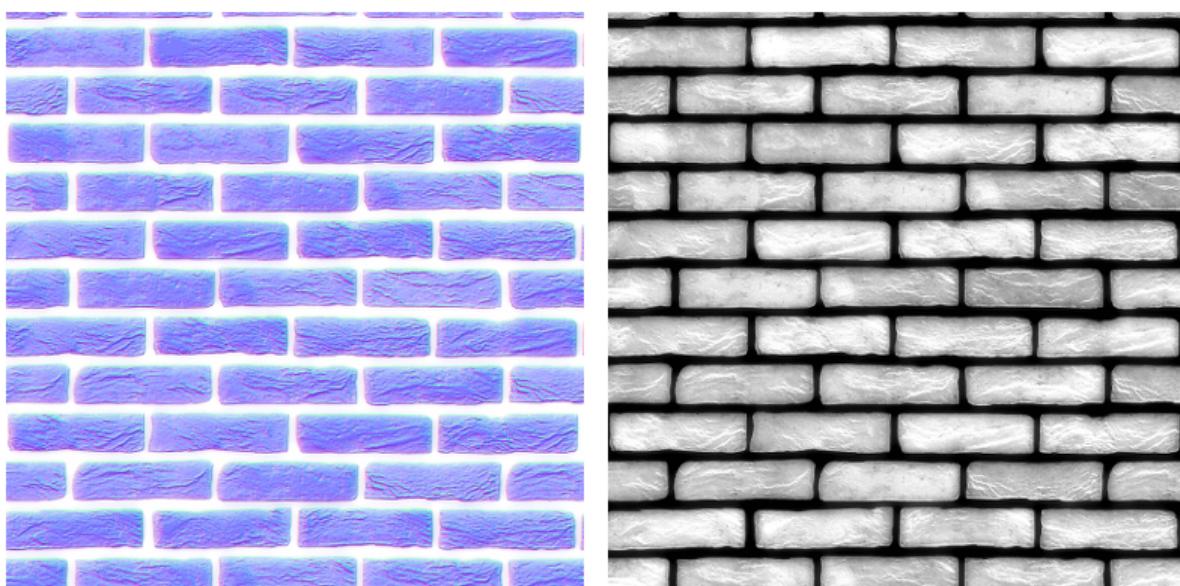
Выставить опцию *Geometry > Normal* на панели *Textures > Influence*.

## Дополнительные настройки

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

## 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.



## Активация

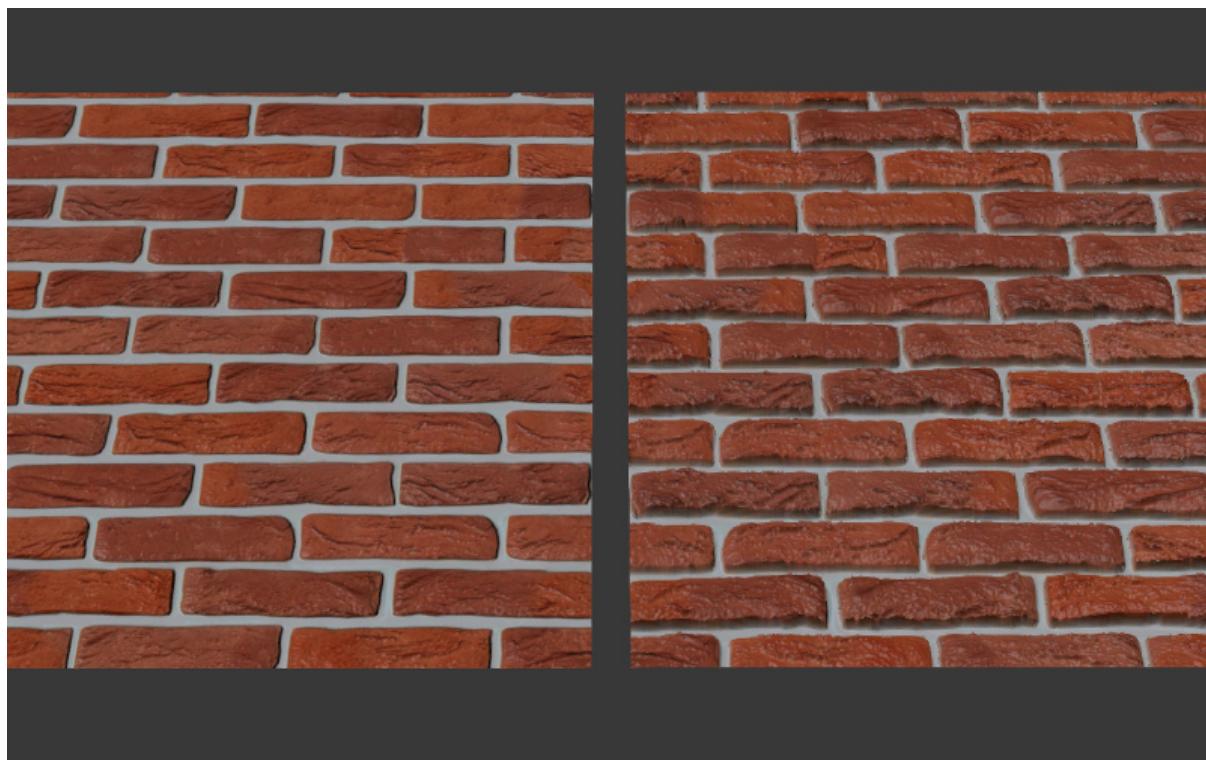
Для карты нормалей дополнительно к опции **Geometry > Normal** активировать панель **Parallax**.

## 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.



## Карта смещивания (stencil map)

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

### 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**.

### Дополнительные настройки

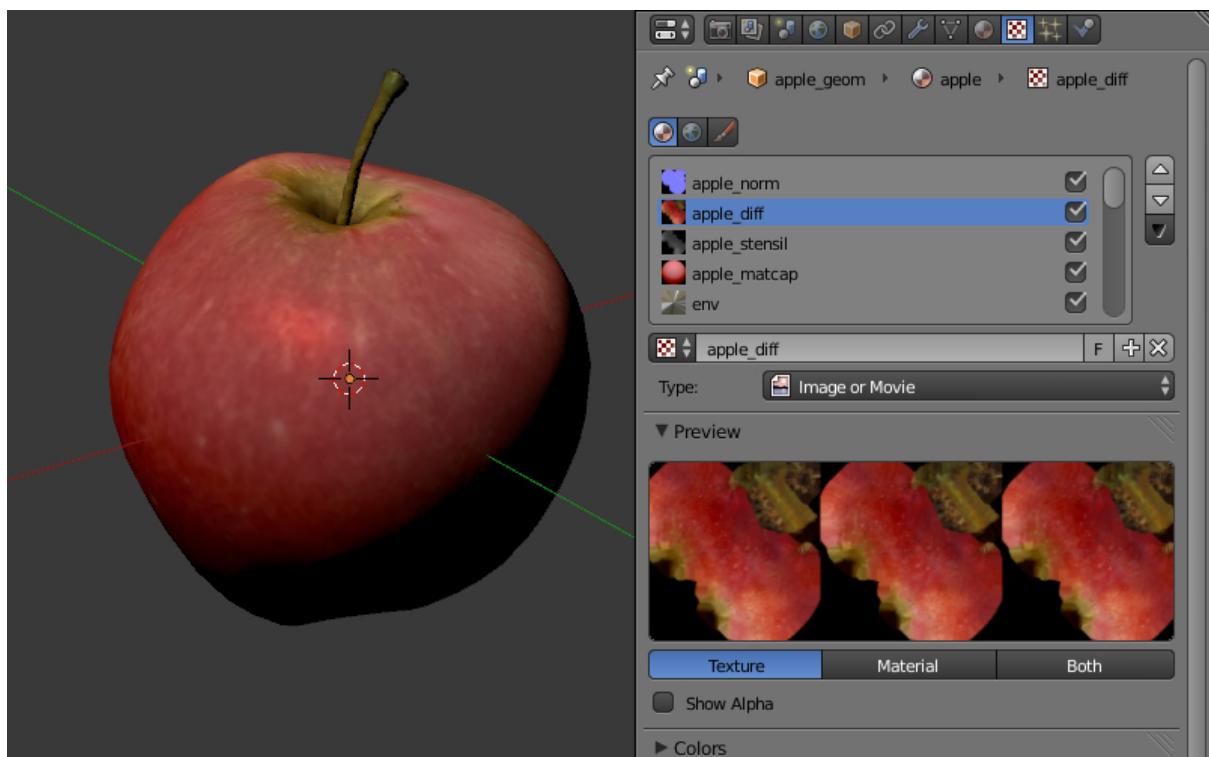
В случае обычных материалов для одной из смешиемых диффузных текстур поддерживается тип текстурных координат **Normal** (“matcap”).

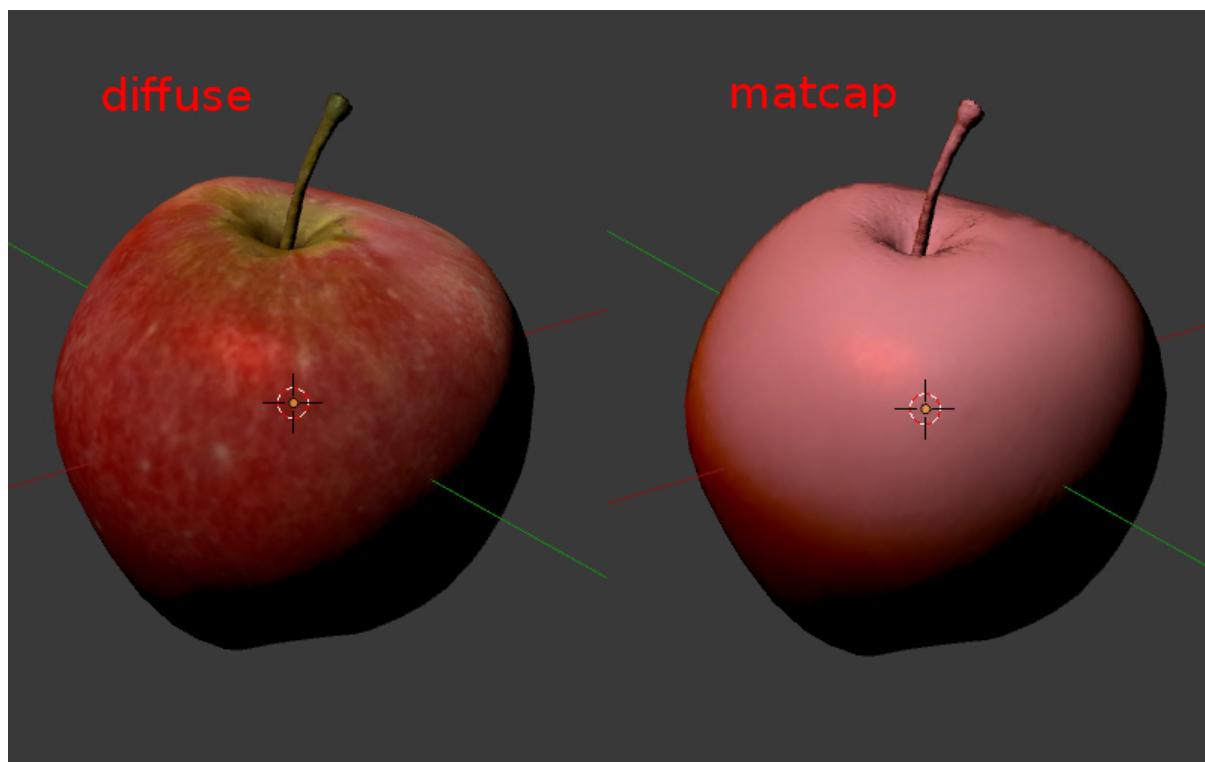
## Ограничения

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

## Пример

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





## 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**.

---

### 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*.

## 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.

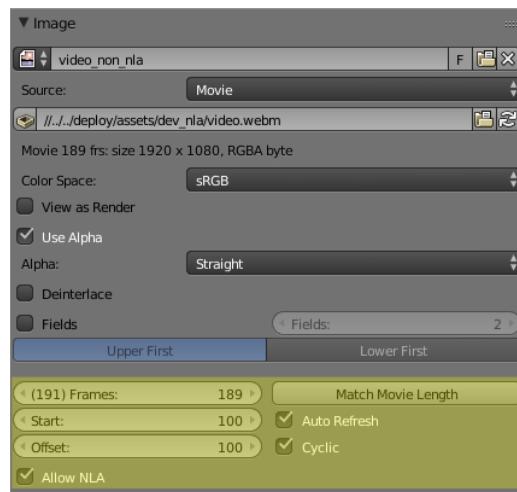
**Image > Allow NLA** Воспроизводить текстуру в рамках NLA-сценария. Необходимо также включить NLA глобально на сцене, активировав панель **Scene > NLA**. По умолчанию включено.

Для текстур, управляемых через NLA будет актуальна следующая опция:

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

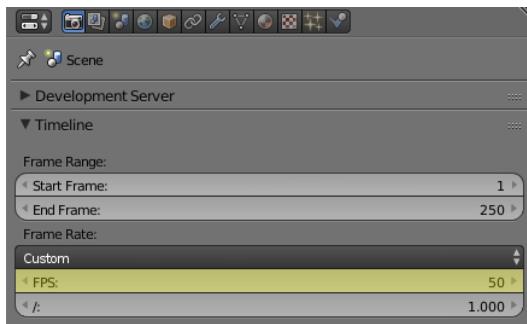
Для текстур, управляемых не через NLA будет актуальна следующая опция:

**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 рассчитывается 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>`.

---

## Особенности работы на мобильных устройствах

Присутствуют следующие особенности реализации на мобильных устройствах:

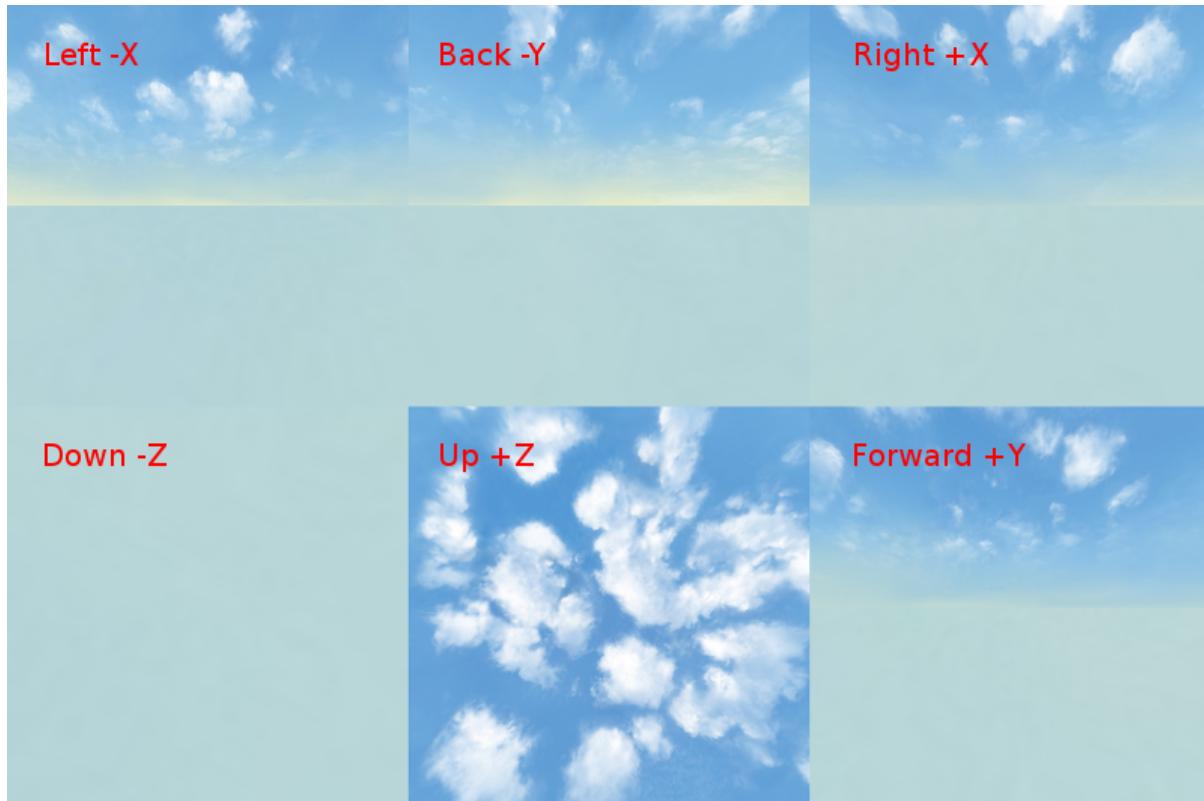
- на iPhone используется специально разработанный видеоформат **.seq**, потому что данное устройство открывает все стандартные видеозаписи через свой видеопроигрыватель. Видео надо сконвертировать заранее, используя [наш конвертер](#).
- на некоторых мобильных устройствах имеется поддержка воспроизведения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

## 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'e). Размер раstrов для каждого из изображений должен подчиняться правилу  $2^N$  (512, 1024 и т.п.).

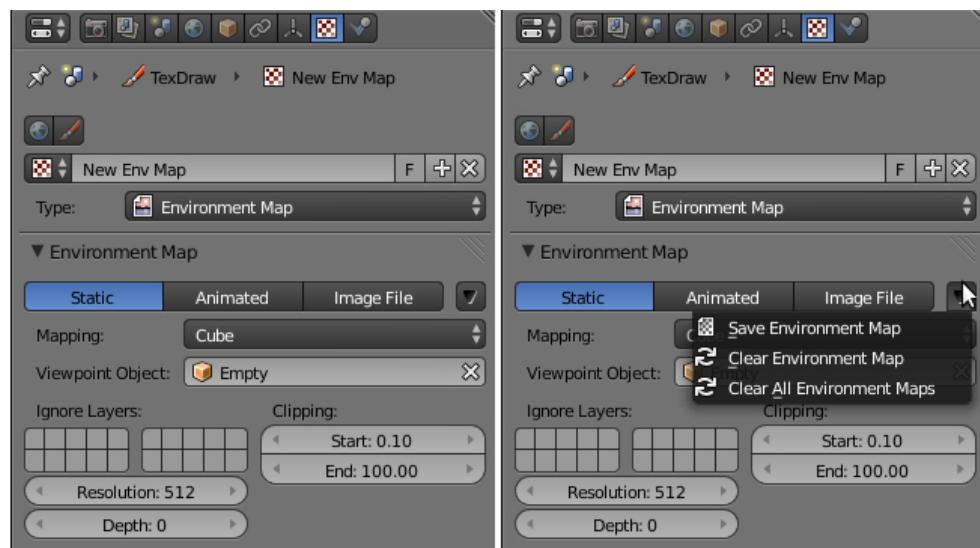
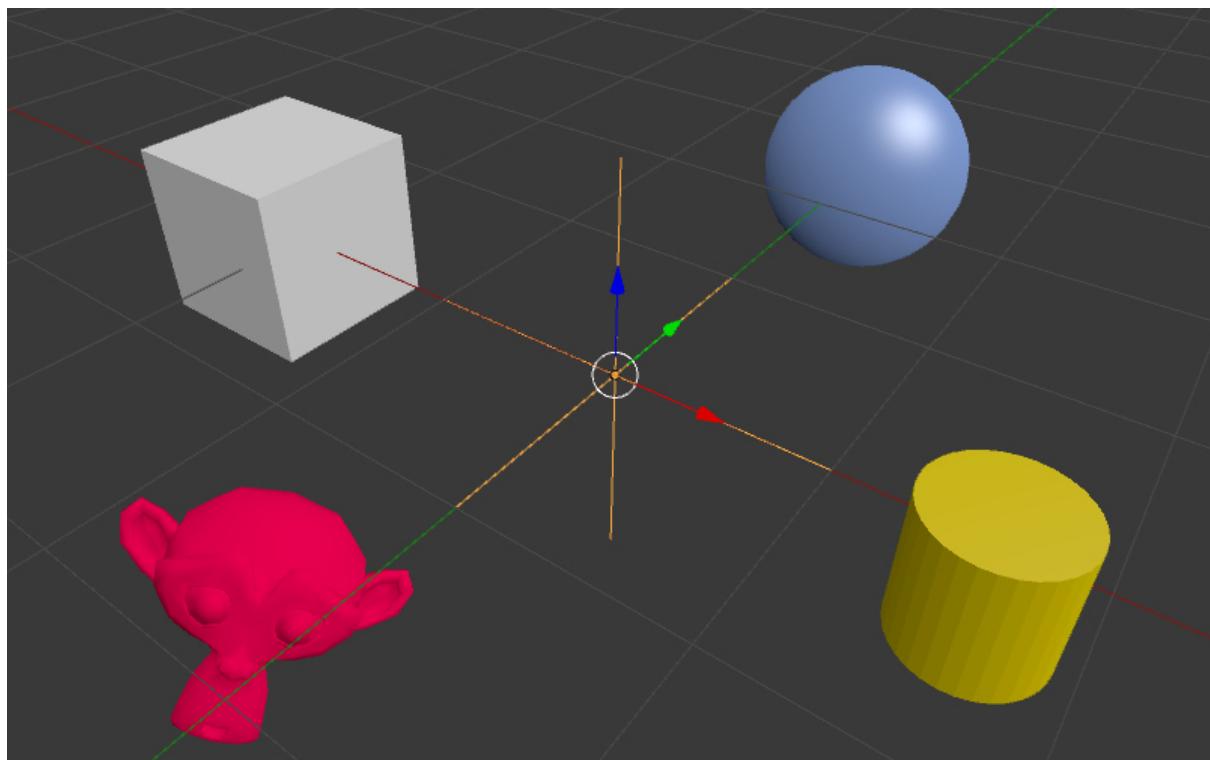
Во избежание проявления швов рекомендуется использовать формат без потери качества (PNG).



## 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.



## Карта зеркального отражения (mirror map)

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

## Активация

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

## Дополнительные настройки

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

См.также:

*Статическое отражение.*

## 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*.

## 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.

---

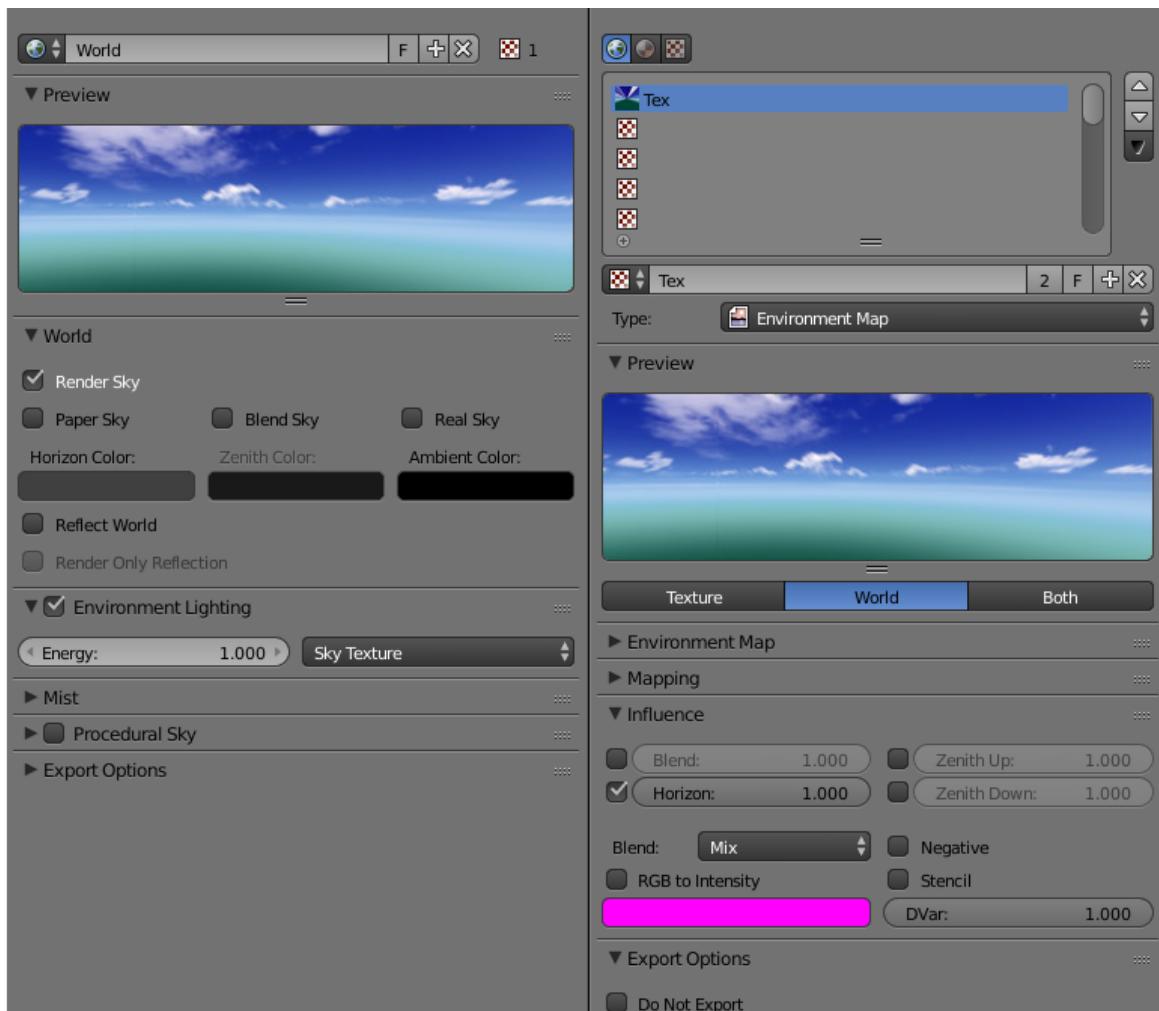
**Примечание:** Поведение текстуры приближено к поведению рендера в Blender Internal. Поэтому, при стандартных настройках текстуры, она может не отображаться. Для отображения текстуры активируйте флаг в панели текстуры Influence > Horizon и установите значение Horizon равное 1.0.

---

**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.

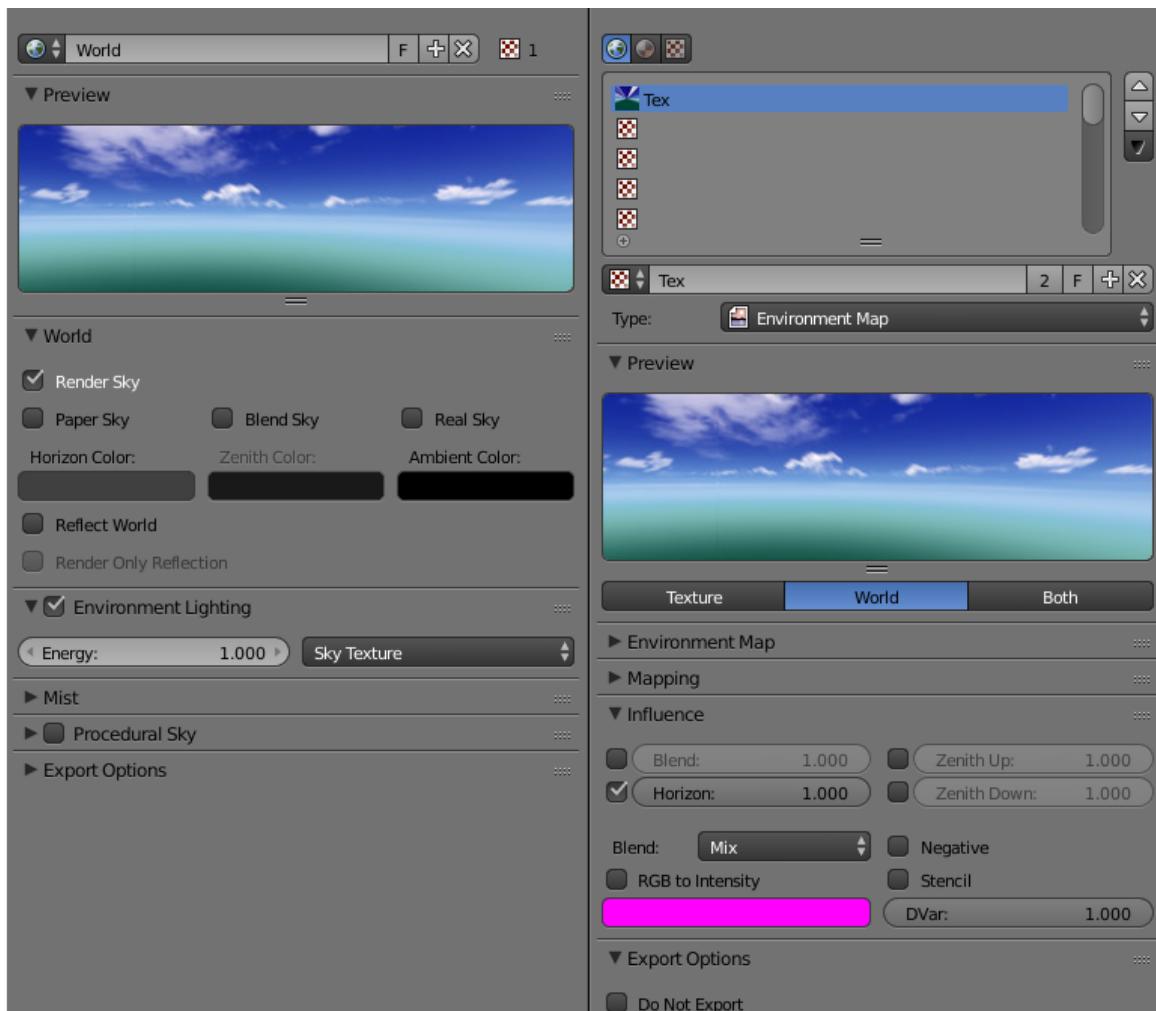
Для создания обоих эффектов от одной текстуры необходимо выставить опцию Export Options > Sky Texture Usage > BOTH.

---



## Дополнительные настройки

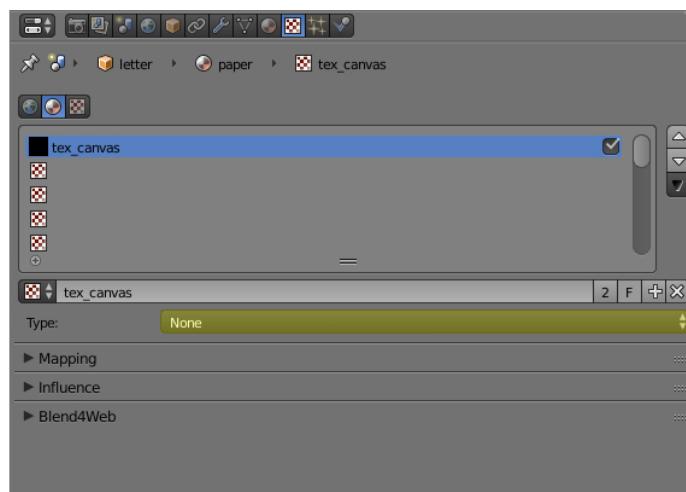
Движком поддерживаются параметры вкладки **Influence** для настройки отображения неба. Смешивание текстуры с цветом производится в зависимости от параметров **World > Horizon Color** (цвет горизонта) и **World > Zenith Color** (цвет зенита), а также флагов **Paper Sky**, **Blend Sky** и **Real Sky**.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

## 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:

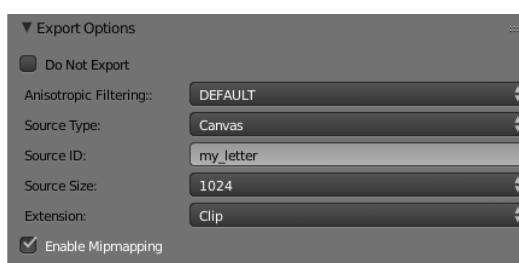
**Export Options > Source Type** Выбор особого типа текстуры: **Scene** - для рендеринга 3D сцены в текстуру, **Canvas** - использование HTML-элемента `<canvas>` и **None** - отсутствие текстуры.

**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.



## Рендеринг в текстуру

Изображение 3D сцены может быть использовано в качестве текстуры на объекте другой (“главной”) сцены. Этот приём известен как “рендеринг в текстуру”. Для того, чтобы использовать его, выполните следующие действия:

1. Создаёте новую сцену, которая будет использоваться для рендеринга в текстуру.
2. Для удобства переименуйте сцену.
3. Создайте для сцены группу настроек **World**.
4. Добавьте в сцену необходимые объекты.
5. Добавьте в сцену камеру и настройте её.
6. Затем переключитесь на главную сцену.
7. Выберите нужный вам объект и создайте для него UV-развёртку.
8. Создайте текстуру, которая будет использоваться для рендеринга.
9. Для этой текстуры выставьте тип **None**.
10. На панели **Mapping**, установите параметр **Coordinates** в значение **UV**.
11. В поле **Export Options > Source Type** указать тип **Scene**.
12. В поле **Export Options > Source ID** указать название сцены-источника.
13. 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.

---

## Canvas-текстуры

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 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 obj = m_scenes.get_object_by_name("NAME");
var ctx = m_tex.get_canvas_ctx(obj, "TEXTURE_NAME");
...
// operations with canvas context
...
m_tex.update_canvas_ctx(obj, "TEXTURE_NAME");
```

Для получения контекста используется функция `get_canvas_ctx()`, куда передается объект и имя текстуры, задаваемое в Blender. После преобразований с контекстом вызывается функция `update_canvas_ctx()`, которая осуществляет визуализацию изменений на указанном объекте, в указанной текстуре.

---

**Примечание:** Если в Blender на разных объектах была назначена одна и та же текстура типа Canvas, то и после загрузки в движке будет одна общая текстура, а не несколько уникальных. Результат работы с ней будет заметен на всех использующих её объектах. Это может быть полезно с точки зрения оптимизации. Если же требуется противоположное поведение, то следует назначить разные текстуры ещё в Blender, либо использовать [глубокое копирование](#) уже непосредственно в приложении.

---

---

## Система частиц. Флюиды

---

### Содержание

- Система частиц. Флюиды
  - Использование
    - \* Необходимые этапы
    - \* Рекомендуемые дополнительные настройки
  - Настройка
    - \* Общие настройки
    - \* Настройки испускания
    - \* Настройки направления
    - \* Настройки вращения
    - \* Настройки физики
    - \* Настройки отображения
    - \* Настройки влияния силовых полей
    - \* Специальные настройки движка
  - Текстуры в системах частиц
    - \* Текстуры материала частиц
    - \* Текстуры системы частиц

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.

## Использование

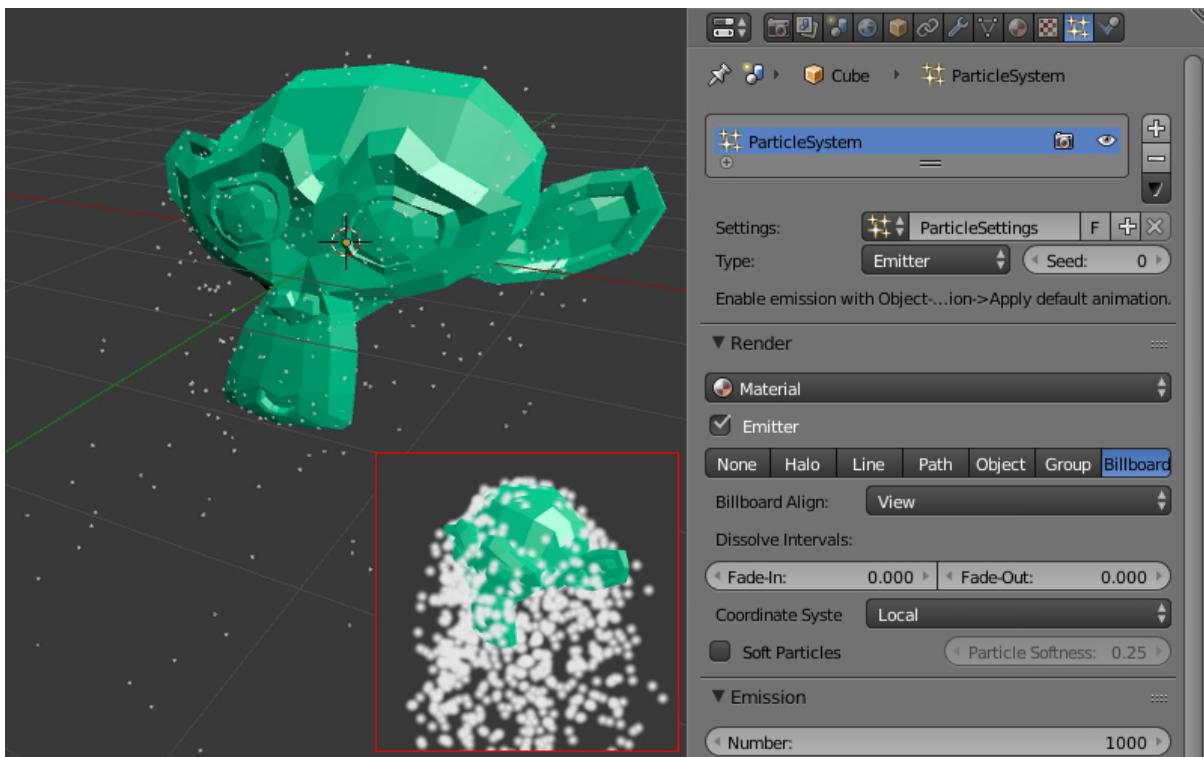
### Необходимые этапы

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**, для эмиттера - **Animation > Apply Default Animation**.

- “нециклическая анимация” - для эмиттера выставить опцию Animation > Apply Default Animation.

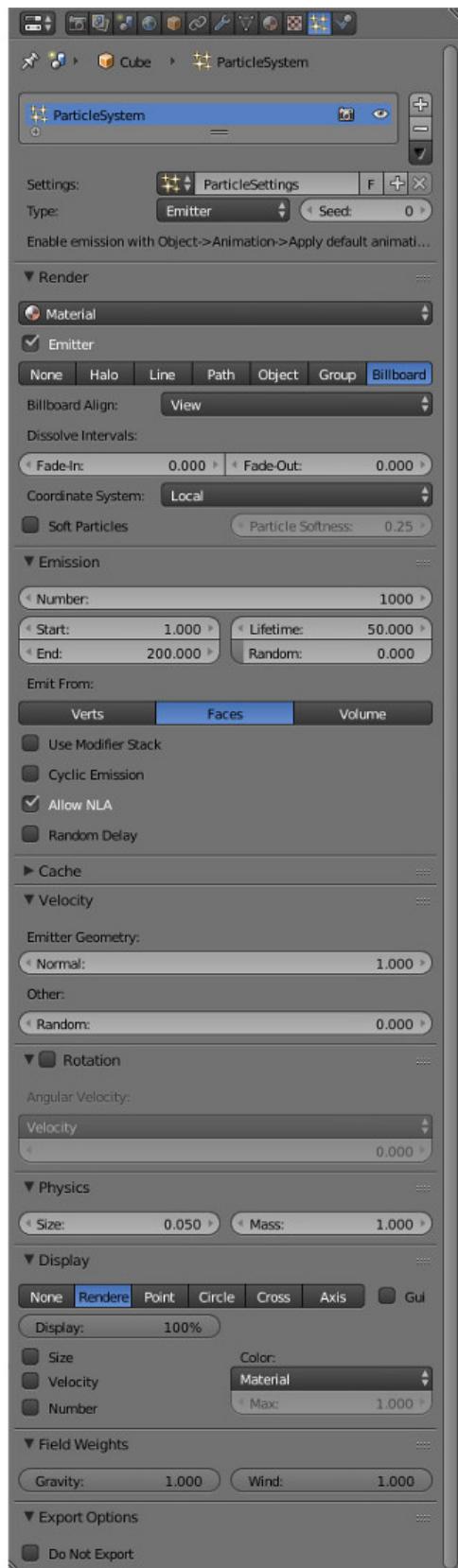
## Рекомендуемые дополнительные настройки

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.



## Настройка

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



## Общие настройки

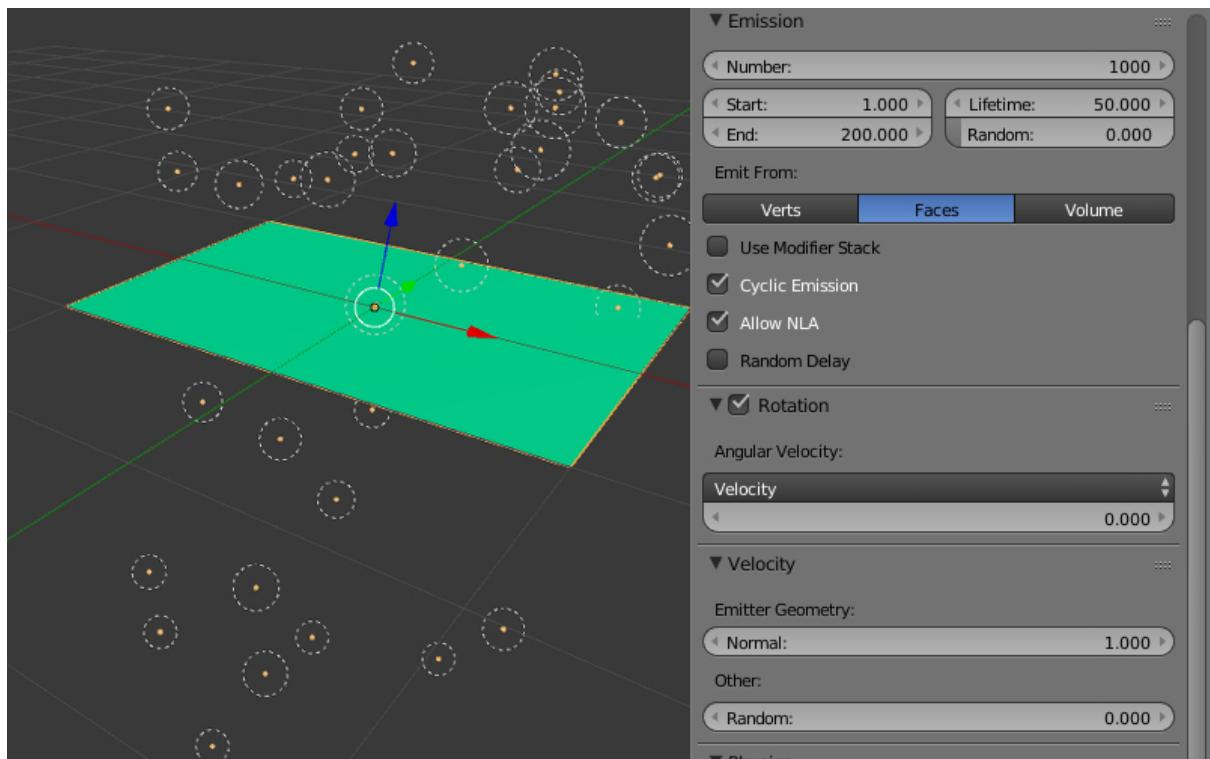
**Name** Название системы частиц. Значение по умолчанию “ParticleSystem”.

**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.

## 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 > Allow NLA** Если эта опция включена, испускание частиц может контролироваться с помощью NLA. По умолчанию активировано.

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

## Настройки направления

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

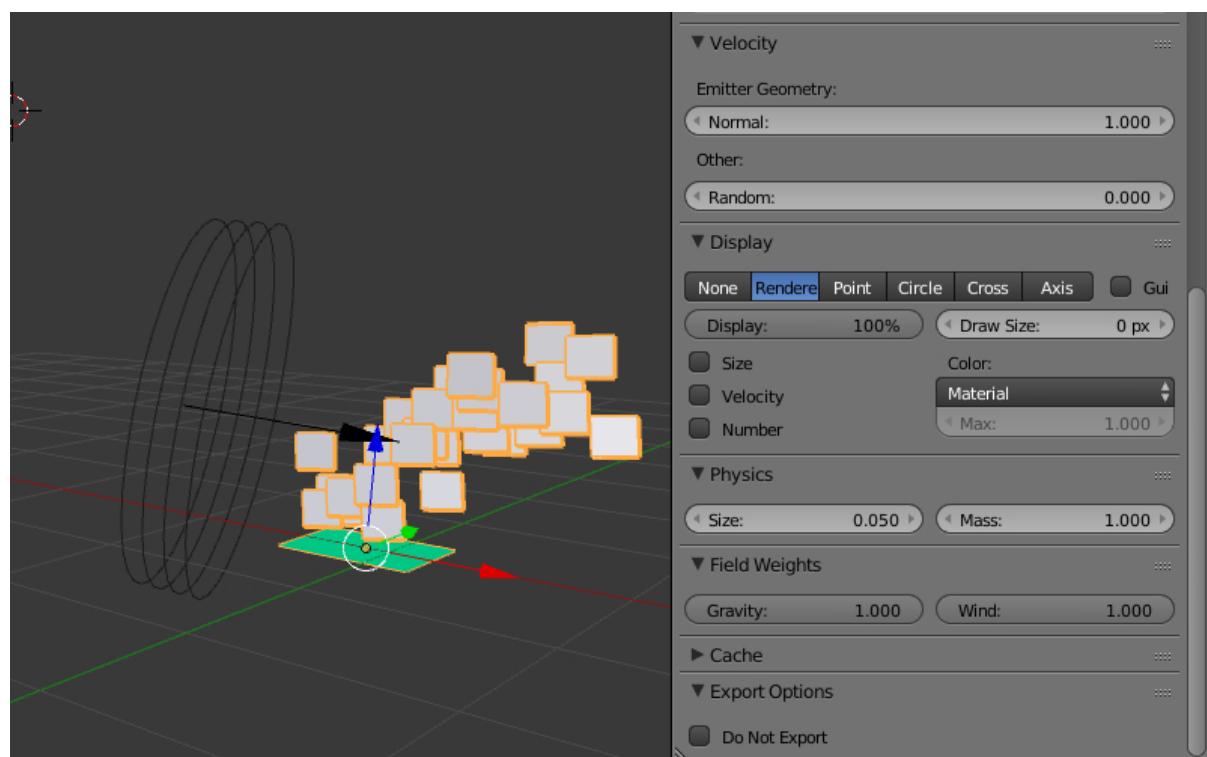
**Velocity > Other > Random** Фактор случайности для направления испускания. Значение по умолчанию 0.0.

## Настройки вращения

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

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

## Настройки физики



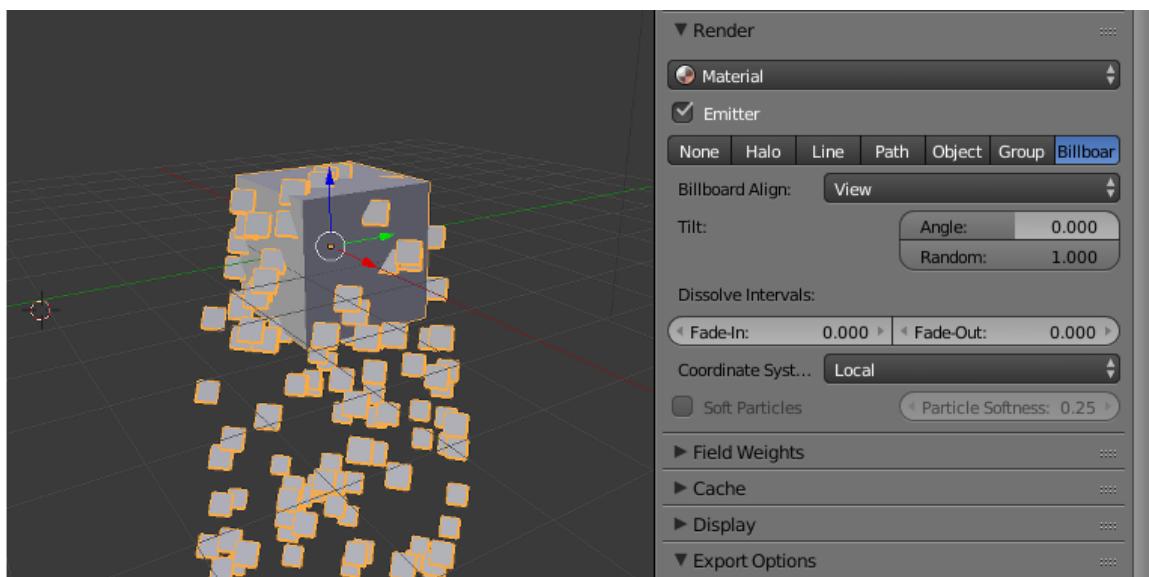
**Physics > Type** Тип расчетов физики: No, Newtonian, Keyed, Boids, Fluid. Игнорируется движком. Всегда используется физика Ньютона (Newtonian). Значение по умолчанию Newtonian.

**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.

## 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 билбордов рекомендуется включать режим Billboard. Значение по умолчанию Halo.

**Render > Billboard Align** Способ ориентирования билбордов: 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 > Tilt** Эта группа параметров управляет начальным наклоном билбордов. Включает два параметра:

**Angle** Угол поворота билбордов. Значение может меняться от -1 до 1. Значение, равное 1, поворачивает билборд на 180 градусов (т.е., переворачивает его вверх ногами). По умолчанию значение равно нулю.

**Random** Случайное изменение угла наклона. Может принимать значения от 0 до 1. По умолчанию равно нулю.

**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.

## 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.

## Специальные настройки движка

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

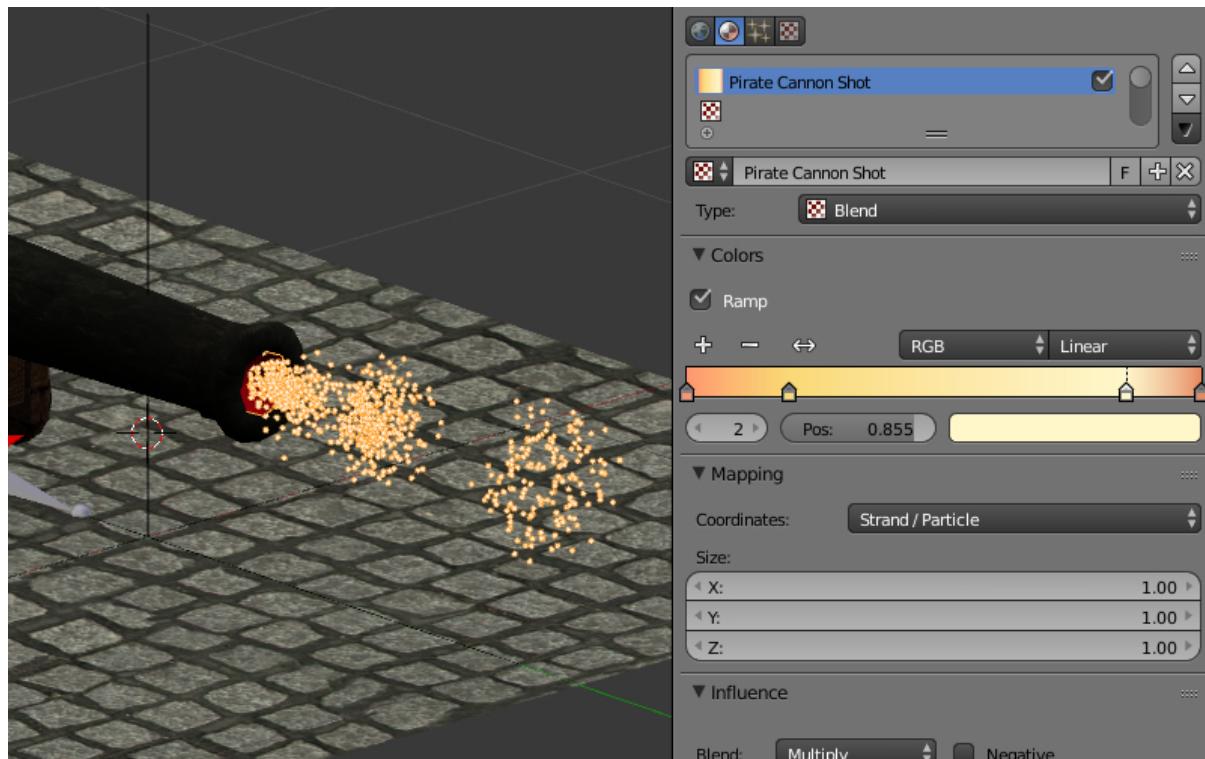
## Текстуры в системах частиц

Системы частиц поддерживают как стандартные, так и нодовые материалы типа Surface и Halo. При использовании материала типа Halo также поддерживаются все его дополнительные настройки, такие как Rings, Lines и Star Tips.

## 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).

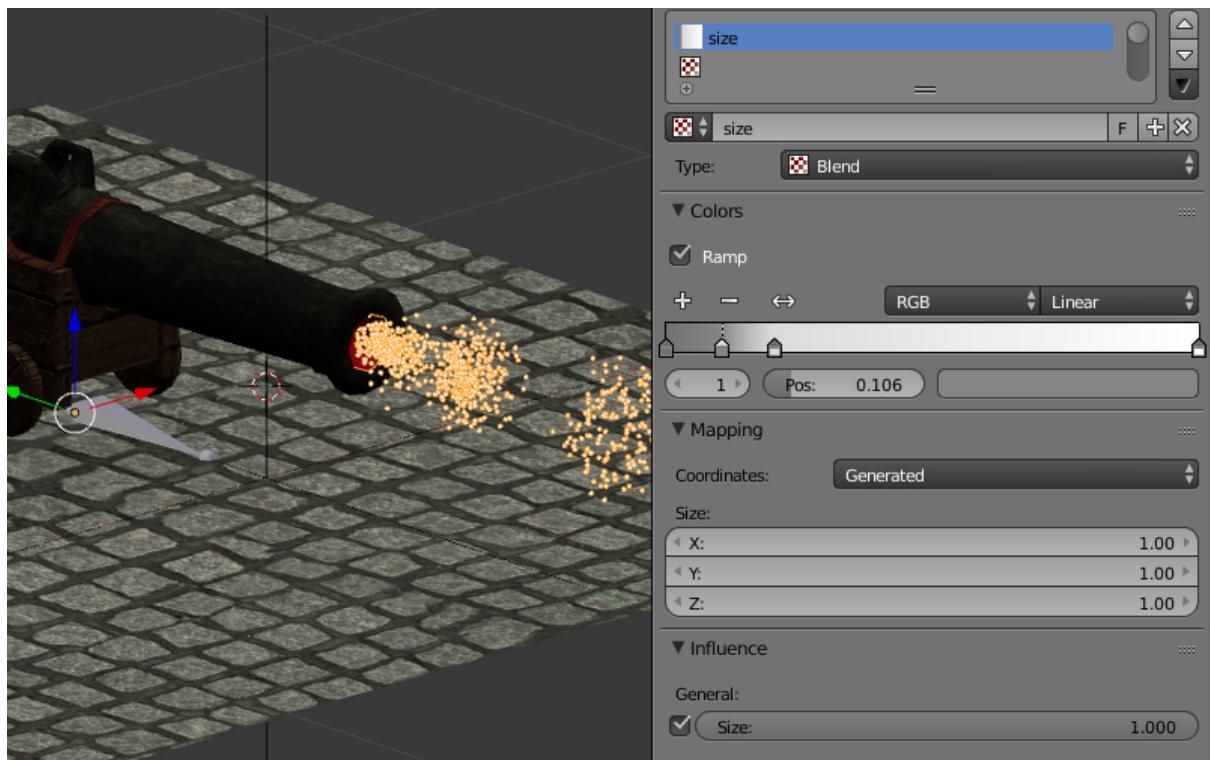


## Текстуры системы частиц

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

---

### Содержание

- *Система частиц. Инстансинг*
  - *Настройки системы частиц*
    - \* *Активация*
    - \* *Настройки отображения*
    - \* *Настройки динамической травы*
    - \* *Настройки наследования*
  - *Травяной покров*
  - *Листья деревьев*

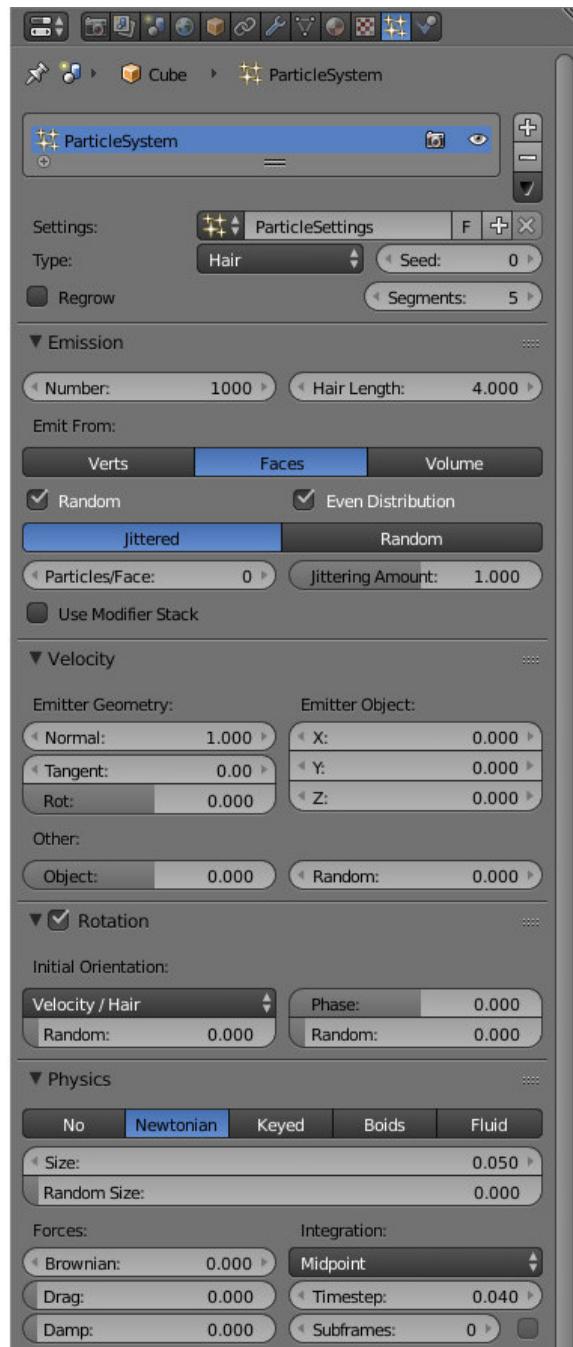
Система частиц может использоваться для создания множественных копий объектов (инстансинга). Кроме сокращения времени на создание сцены, инстансинг позволяет сэкономить на времени загрузки и сократить потребление памяти сценой в сравнении с использованием отдельных объектов.



Использование системы частиц для инстансинга имеет ограничения:

- Невозможно осуществлять движение и анимацию объектов внутри системы частиц.
- Не допускается создавать иерархические сущности среди объектов внутри системы частиц за исключением дуплицирования групп.
- Не допускается инстансинг отличных от объектов-мешей сущностей.

## Настройки системы частиц



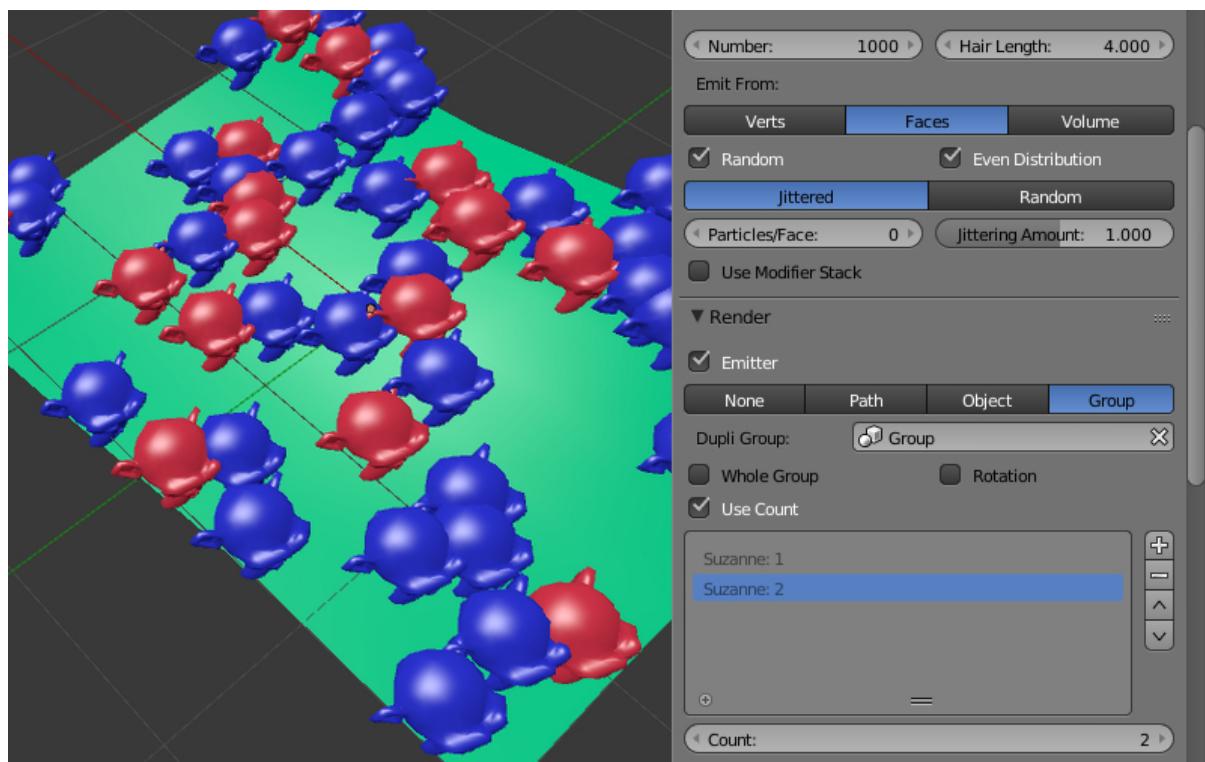


## Активация

- 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.



## 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 билбординга для частиц. По умолчанию выключено.

#### *Render > Billboard Type*

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

- **Basic** - простой односторонний билбординг: частицы всегда будут повернуты лицевой стороной
- **Random** - случайный двусторонний билбординг: частицы чаще всего будут повернуты лицевой, либо обратной стороной, реже - боком; присутствует небольшой случайный поворот; модель создана специально для инстансинга травы
- **Jittered** - односторонний билбординг 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 билбордов (опция доступна при включении **Render > Billboard**). Доступны два типа:

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

Значение по умолчанию **Spherical**.

## 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**.

## 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*

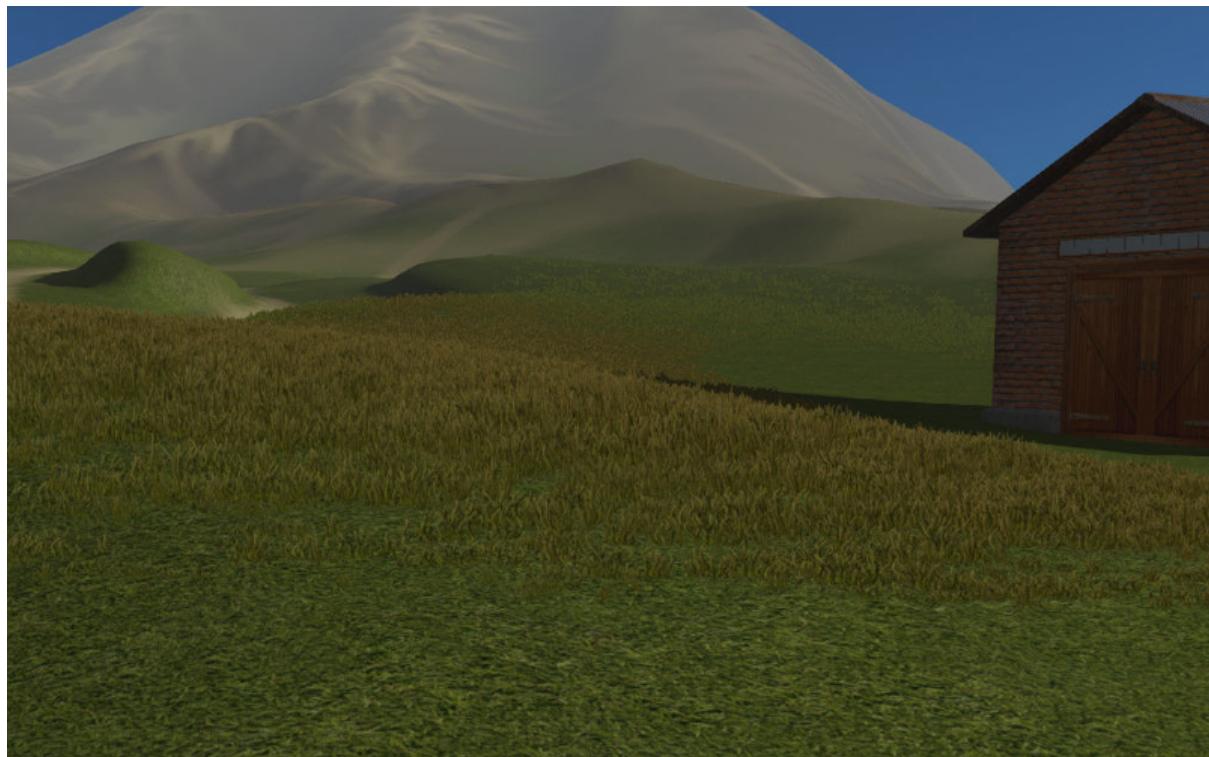
Наследование частицами вертексного цвета с эмиттера. Содержит два поля:

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

По умолчанию наследования не происходит.

## 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).

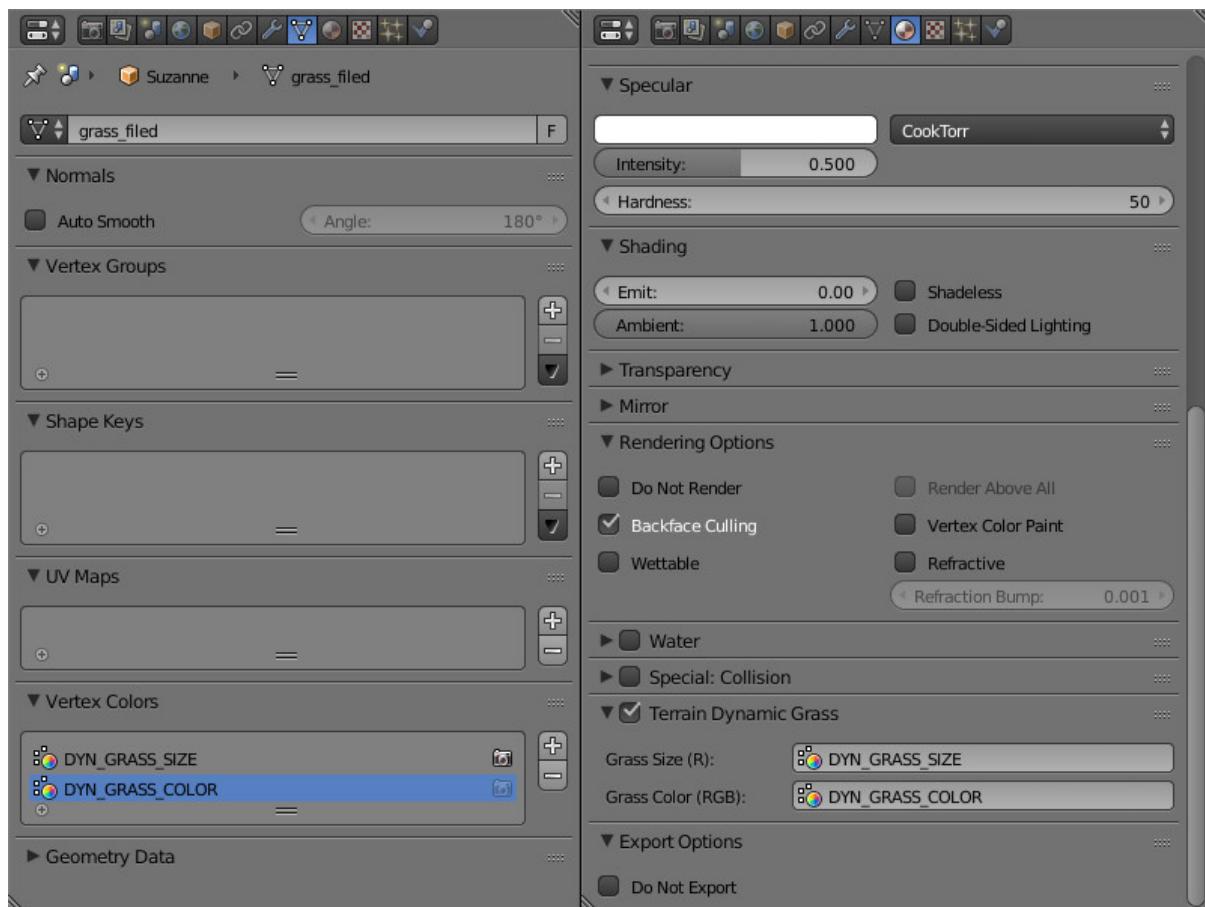
На **материале** ландшафта при включении опции **Terrain Dynamic Grass** становятся активными текстовые поля:

**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**).



## 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 билбординга

Рекомендуется включить билбординг, 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

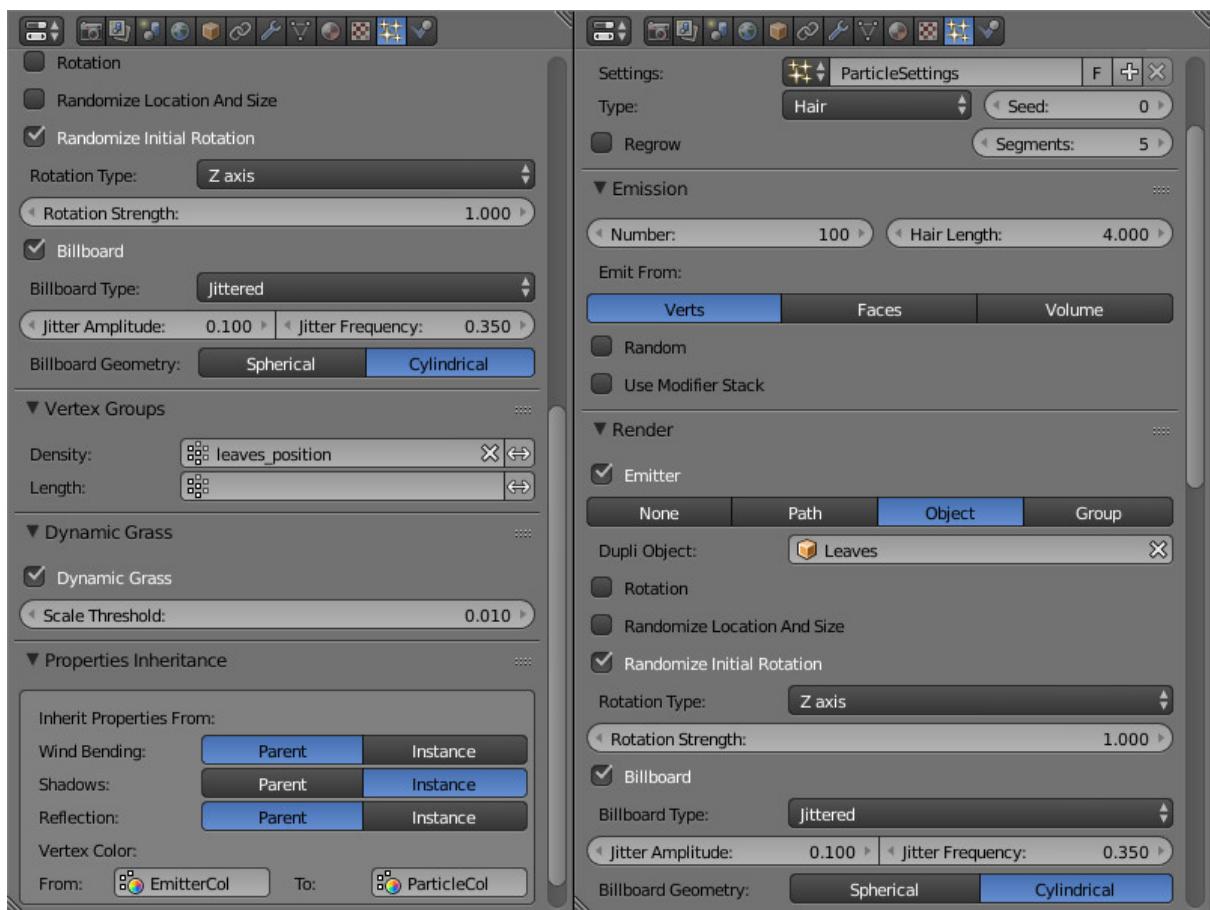
Для наследования частицами вертексного цвета эмиттера нужно указать имя цвета эмиттера и имя цвета частицы соответственно в полях Properties Inheritance > Vertex Color > From и Properties Inheritance > Vertex Color > To. При наследовании цвет ближайшей к частице вершины эмиттера из From будет скопирован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

### Содержание

- *Анимация*
  - Управление анимацией
  - Объектная анимация
  - Скиннинг и скелетная анимация
    - \* Параметры запекания
  - Вертексная анимация
    - \* Параметры запекания
  - Анимация по умолчанию
  - Нелинейная анимация
    - \* Редактор NLA
    - \* Управление через API
  - Параметризация источников звука
  - Анимация нод Value и RGB

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*.

## 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*.

## 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.

## 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:



## Параметры запекания

Запекание производится при выделенном арматурном объекте.

В окне со списком запекаемых акторов приводятся те акторы, которые следует запечь. Если список пуст, запекаются 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.

**Use Blender's Native Baker** Использовать для запекания анимации средства самого Blender (возможность Bake Action), а не Blend4Web. Настройки средства запекания Blender отличаются от настроек Blend4Web.

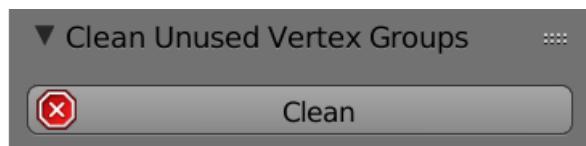
**Bake** Произвести запекание. После успешного окончания процесса на сцене появляются акторы с именами вида *B4W\_BAKED\_ИМЯ\_АКТОРА*. Данные акторы будут автоматически назначены на арматурном объекте и воспроизведены в движке. Стоит отметить, что работа подобных акторов в Blender не гарантируется.

---

**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.



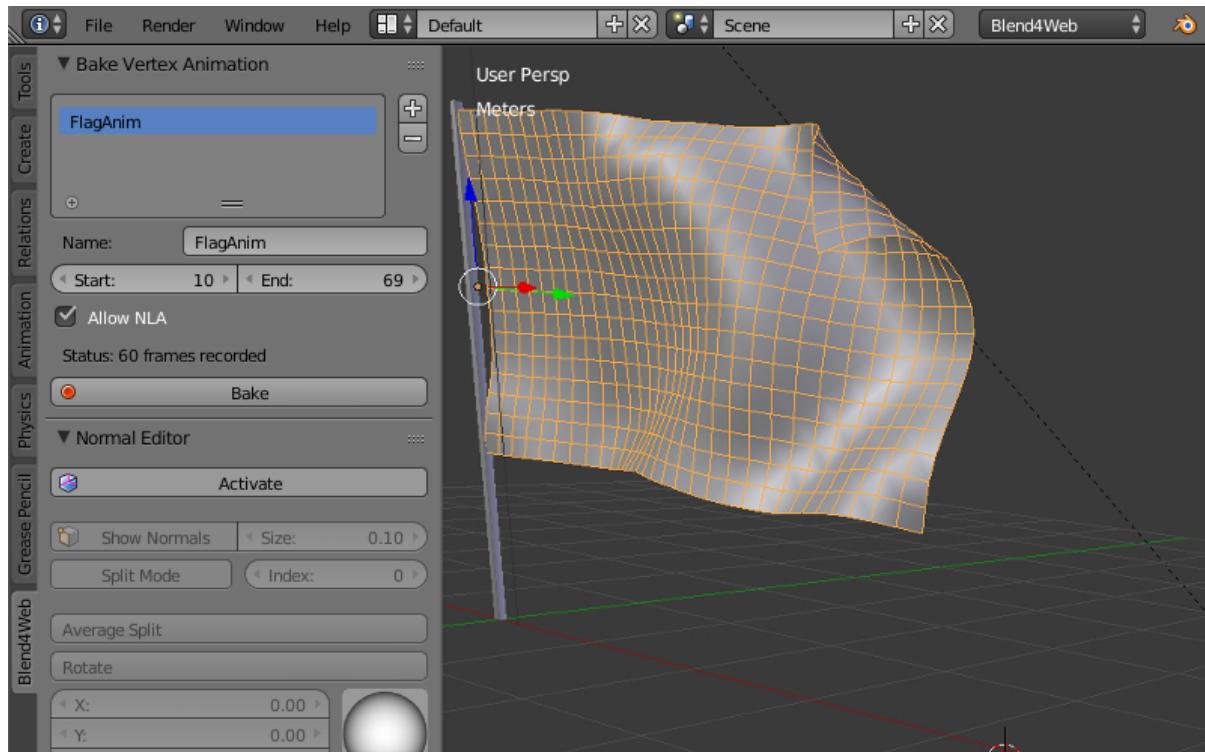
В Blend4Web присутствует начальная поддержка ограничителей для костей. На сегодняшний день поддерживается только один тип ограничителя, **Copy Transform**. Это позволяет привязывать арматуру к различным объектам, в том числе физическим (ragdoll). Поддержка других ограничителей ожидается в будущих версиях.

## 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 вертекской анимации предусмотрен инструмент **Bake Vertex Animation**, расположенный на панели инструментов Blend4Web.



## Параметры запекания

**Name** Имя, которое будет присвоено запечённой анимации.

**Start** Кадр, с которого начнётся запекание анимации.

**End** Кадр, которым закончится запекание анимации.

**Allow NLA** Позволяет управлять запечённой анимацией при помощи [NLA](#).

**Status** В этой строке приводится количество кадров, из которых состоит запечённая анимация. Если запекание ещё не было произведено, вместо количества кадров приводится слово **Empty**.

**Bake** Запекание анимации производится при нажатии на эту кнопку.

---

**Примечание:** Для корректной работы вершинной анимации следует активировать опцию **Export Vertex Animation** (раздел **Export Options** панели **Object**) у соответствующего объекта.

## Анимация по умолчанию

В Blend4Web присутствует возможность автоматически проигрывать анимацию объекта. Для этого нужно активировать параметр **Apply Default Animation** на вкладке **Animation** панели **Object** того объекта, анимацию которого требуется включить. Поддерживается скелетная и объектная анимация.



Проигрывание анимации начинается сразу после запуска приложения и осуществляется аналогично ноде *Play Timeline*, но без возможности задать начальный и конечный кадр (в качестве таковых всегда выступают первый и последний кадры таймлайна). Также присутствует возможность задать поведение анимации, аналогично ноде *Play Animation*.

## Нелинейная анимация

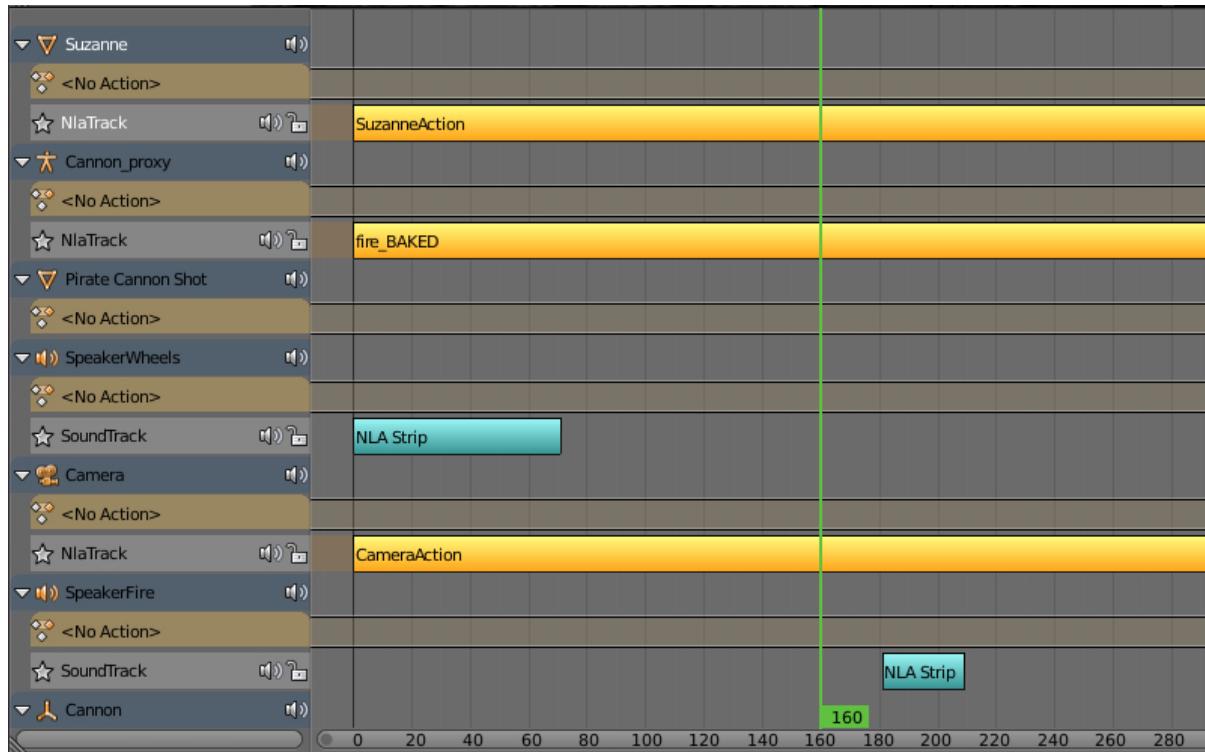
### Редактор NLA

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)



## 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.

## 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*.

---

## Ограничения

- Одновременное проигрывание различных видов анимации для одного объекта не поддерживается.

## 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.

## Параметризация источников звука

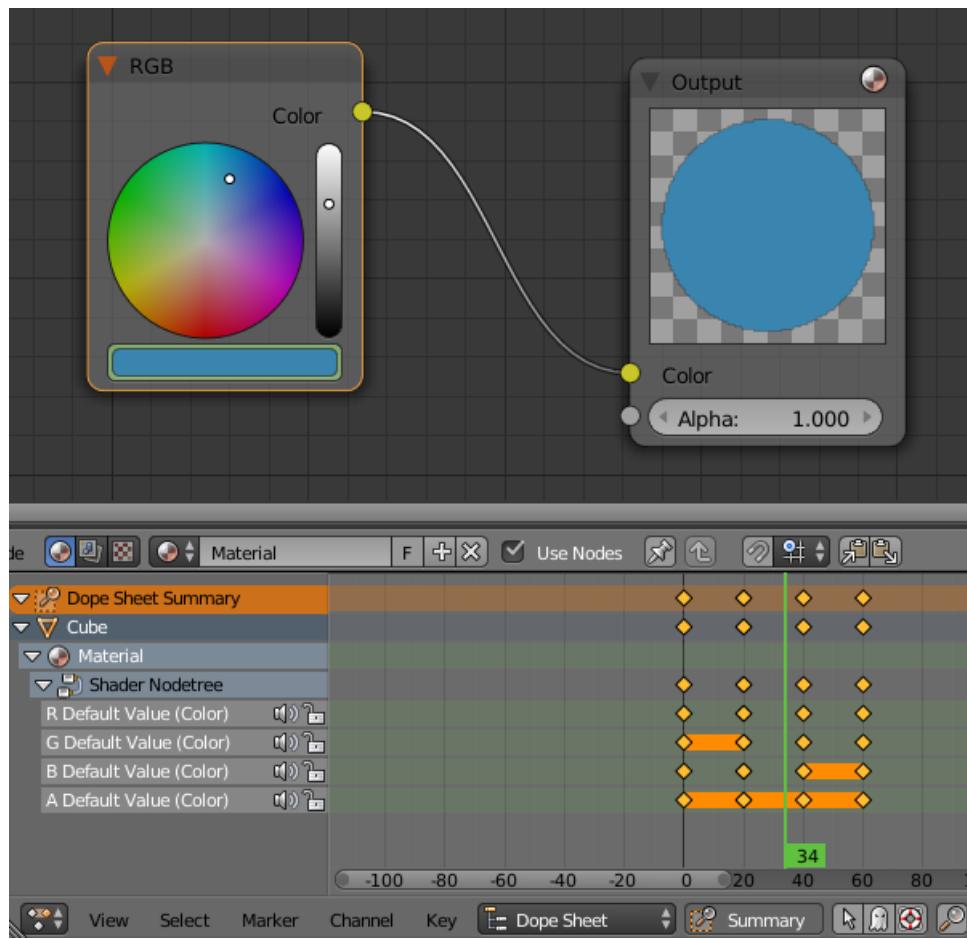
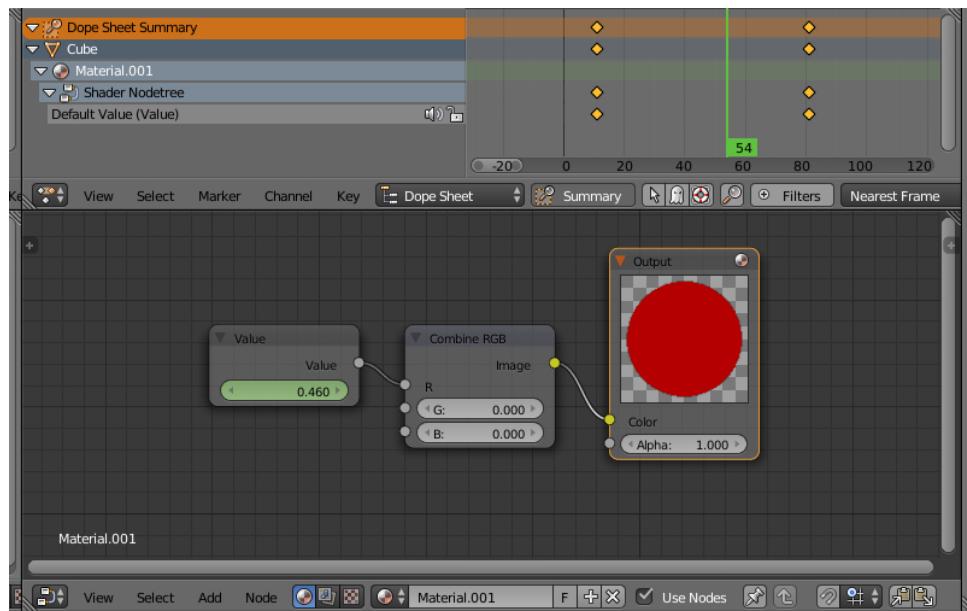
На объектах-спикерах дополнительно поддерживаются следующие типы анимационных ключей:

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

Параметризация источников звука по своей сути повторяет объектную анимацию.

## Анимация над 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:**

*Time (B4W\_TIME)*

## Природные эффекты

### Содержание

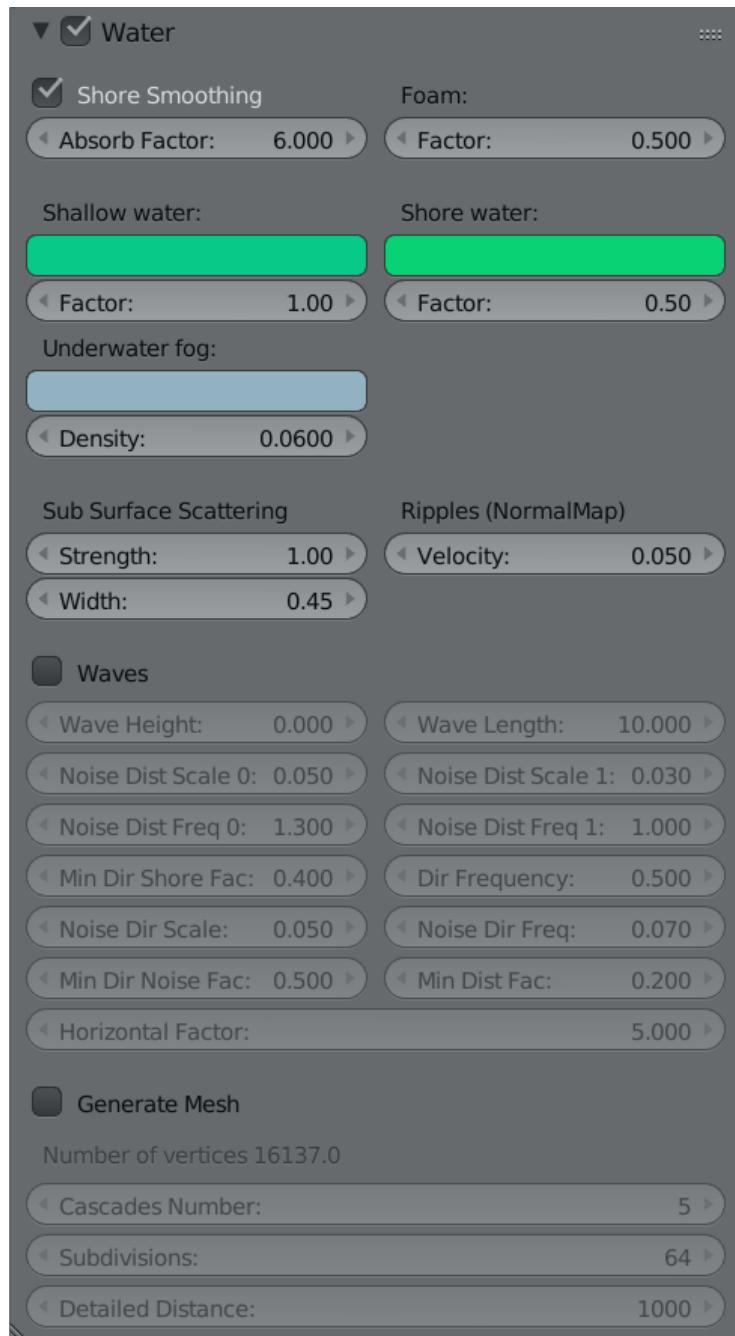
- Природные эффекты
  - Вода
    - \* Активация
    - \* Базовые настройки
    - \* Динамика волн
    - \* Смачивание поверхностей
    - \* Отражение и эффект Френеля
    - \* Сглаживание у береговой линии
    - \* Градиент цвета
    - \* Преломление
    - \* Пена
    - \* Каустика и хроматическая аберрация
    - \* Подводная среда
    - \* Объемные волны
    - \* Настойки генерируемой поверхности
  - Атмосфера
    - \* Рассеивание
    - \* Туман
    - \* Время суток
    - \* Звезды
  - Lens Flare
    - \* Настойки эффекта
    - \* Объект для бликов линз
  - Ветер
    - \* Активация
    - \* Настойка
    - \* Анимация травы и крон деревьев

В Blend4Web присутствует ряд эффектов окружающей среды, которые могут быть полезны при создании открытых пространств.

## Вода

### Активация

Для предполагаемого материала воды активировать панель Water во вкладке Material.



### Базовые настройки

**Прозрачность** Рекомендуется включить прозрачность с градиентом (установить значение Alpha Blend для параметра Transparency > Type) и настроить зна-

чение Alpha.

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

## 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.

## Смачивание поверхностей

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

## Отражение и эффект Френеля

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



## 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. Значение этого параметра может находиться в пределах от 0 до 100. Значение по умолчанию - 6.

## 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. По умолчанию используется синий (0.008, 0.222, 0.8) цвет.

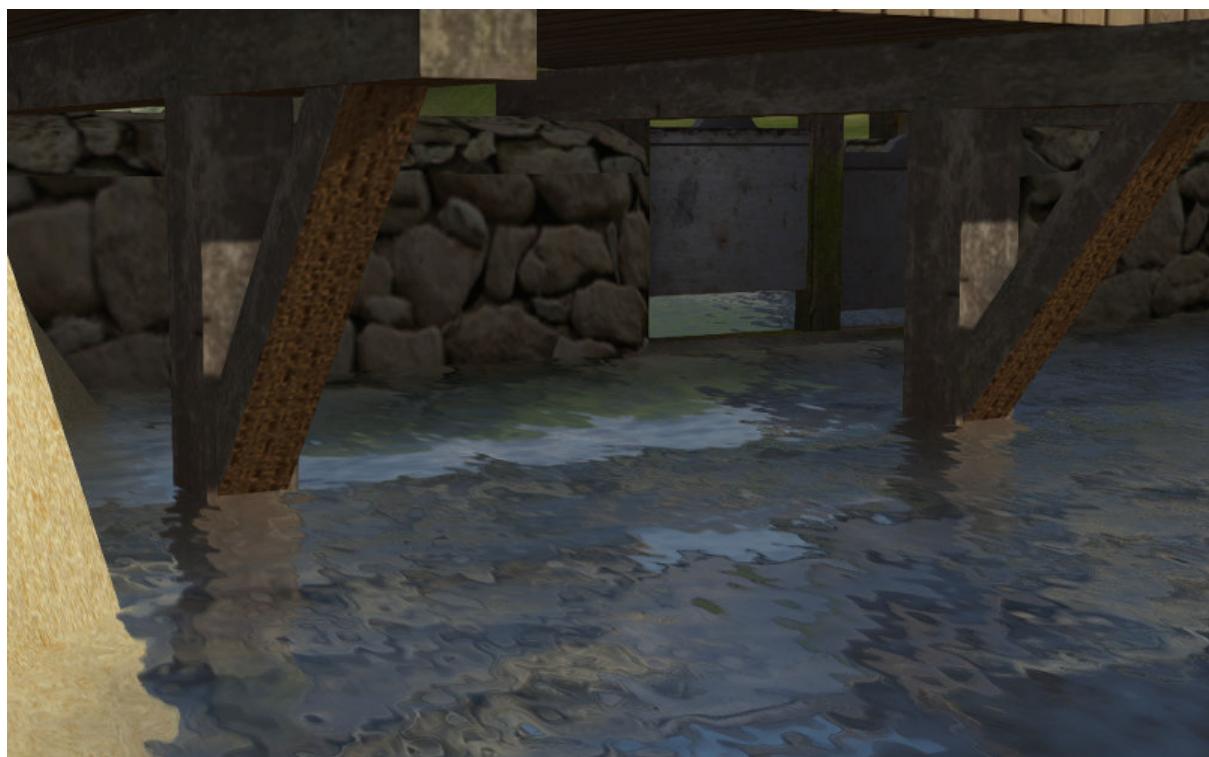
**Shallow Water > Factor** Коэффициент примешивания цвета воды на мелководье. Этот параметр может принимать значения от 0 до 2, значение по умолчанию - 1.

**Shore Water > Color** Цвет воды непосредственно у береговой линии. По умолчанию используется тёмно-синий (0.003, 0.19, 0.57) цвет.

**Shore Water > Factor** Коэффициент примешивания цвета воды на береговой линии. Возможные значения - от 0 до 2, значение по умолчанию - 0.5.

## Преломление

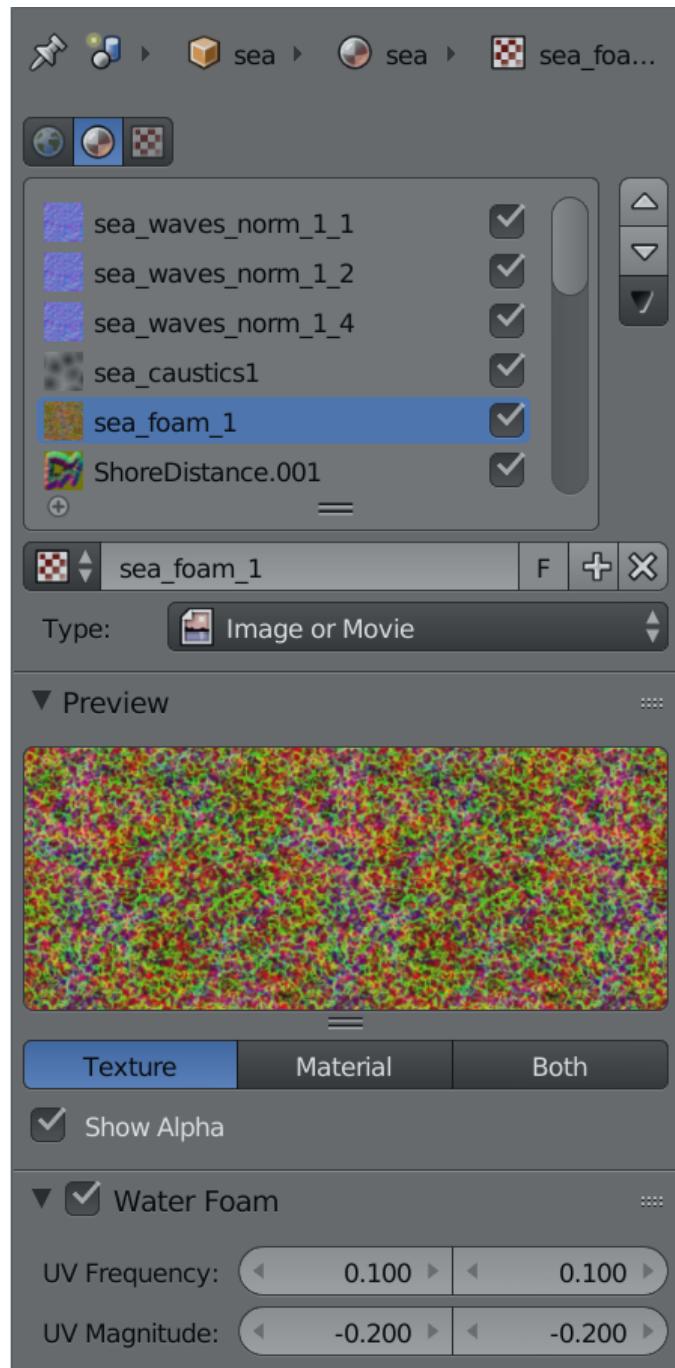
Во вкладке **Render** в панели **Reflections and Refractions** выставить опцию **Refractions** в положение **ON** или **AUTO**.



## 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

**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.

## Каустика и хроматическая aberrация

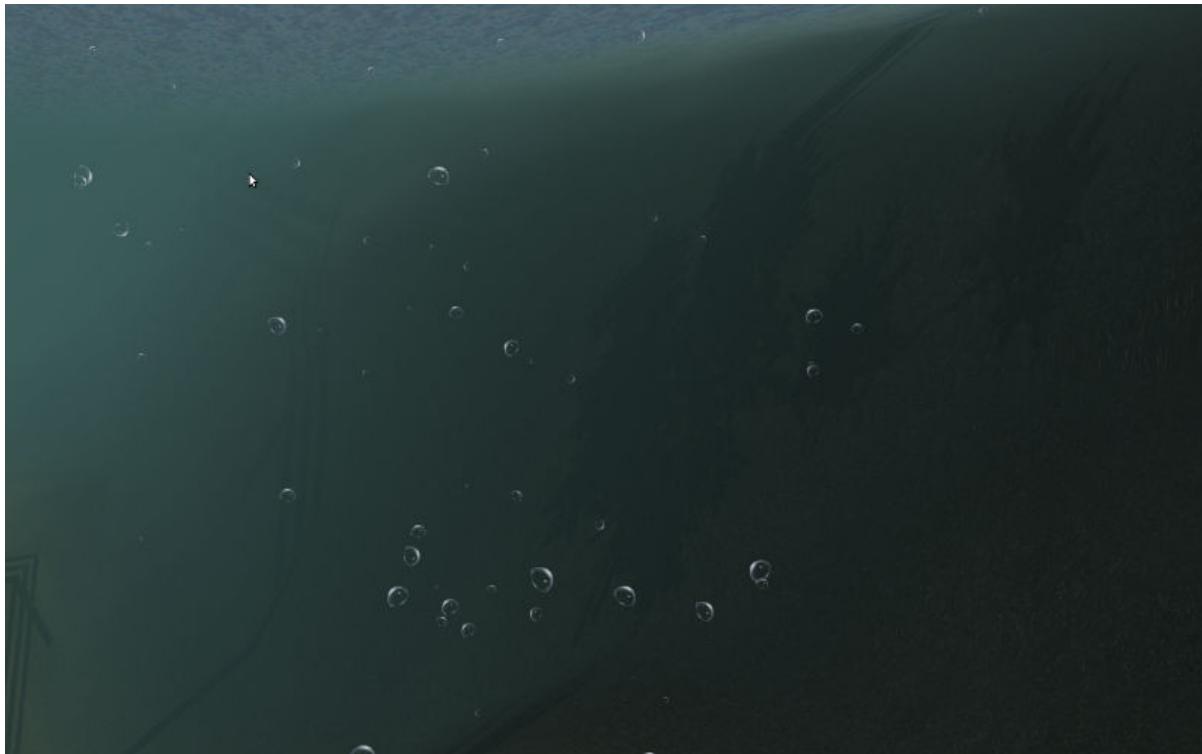
Для создания каустики на подводных объектах следует включить опцию **Caustics**. Так же на сцене должен присутствовать источник освещение типа **Sun**.



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

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

## Подводная среда



### Настройки видимости (“туман”)

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

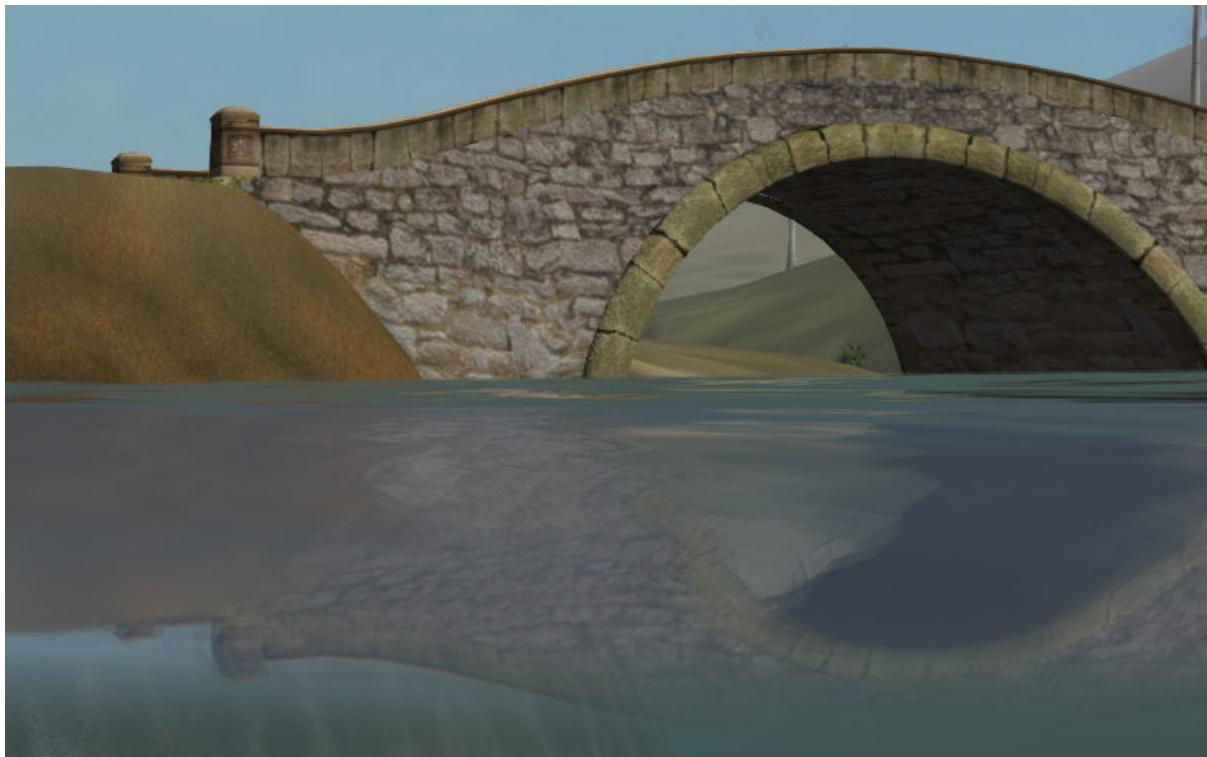
***Underwater Fog > Density*** Экспоненциальный фактор, влияющий на плотность и расстояние. Значение по умолчанию 0.06.

Применяются также настройки [сумеречных лучей](#).

---

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

---



## Объемные волны

### Активация

Для создания объемных волн следует включить опцию Waves.

---

**Примечание:** Направление объемных волн определяется силовым полем типа Wind, присутствующим в сцене.

Если в сцене отсутствует силовое поле типа Wind, объемные волны будут двигаться только в направлении, заданном по умолчанию. Это направление не может быть изменено.

---



## Настройка

**Wave Height** Высота волн. Возможные значения - от 0 до 5. Значение по умолчанию 0.0.

**Wave Length** Длина волн. Возможные значения - от 0.01 до 200. Значение по умолчанию 10.0.

**Noise Dist Scale 0** Размер первого компонента волн, удаленных от берега. Этот параметр может принимать значения от 0 до 1. Значение по умолчанию - 0.05.

**Noise Dist Scale 1** Размер второго компонента волн, удаленных от берега. Возможные значения - от 0 до 1. Значение по умолчанию - 0.03.

**Noise Dist Freq 0** Частота первого компонента волн, удалённых от берега. Этот параметр может принимать значения от 0 до 10. Значение по умолчанию - 1.3.

**Noise Dist Freq 1** Частота второго компонента волн, удаленных от берега. Возможные значения - от 0 до 10, значение по умолчанию - 1.

**Min Dir Shore Fac** Минимальный коэффициент уменьшения высоты прибрежных волн. Диапазон значений - от 0 до 1, значение по умолчанию - 0.4.

**Dir Frequency** Частота накатывания прибрежных волн. Значение этого параметра может изменяться от 0 до 10. Значение по умолчанию - 0.5.

**Noise Dir Scale** Размер шума на прибрежных волнах. Возможные значения - от 0 до 1, значение по умолчанию - 0.05.

**Noise Dir Freq** Частота шума на прибрежных волнах. Возможные значения - от 0 до 1. Значение по умолчанию - 0.07.

**Min Dir Noise Fac** Минимальное значение шума на прибрежных волнах. Этот параметр может принимать значения от 0 до 1. Значение по умолчанию - 0.5.

**Min Dist Fac** Минимальный коэффициент примешивания волн, удаленных от берега. Значение этого параметра может меняться от 0 до 1 и по умолчанию равно 0.2.

**Horizontal Factor** Коэффициент смещения прибрежных волн в направлении к берегу. Этот параметр может принимать значения от 0 до 10 и равен 5 по умолчанию.

## Настройки генерируемой поверхности

**Generate Mesh** Включить генерируемую поверхность.

**Cascades Number** Задаёт количество каскадов в генерируемой поверхности. Этот параметр может принимать значения от 1 до 20. Значение по умолчанию - 5.

**Subdivisions** Количество подразделений в генерируемой поверхности. Значение по умолчанию - 64. Минимальное возможное значение для этого параметра - 2, а максимальное - 512.

**Detailed Distance** Этот параметр задаёт максимальное расстояние от камеры до края последнего каскада. Значение параметра может меняться от 1 до 5000. Значение по умолчанию - 1000.

## 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.

## Атмосфера

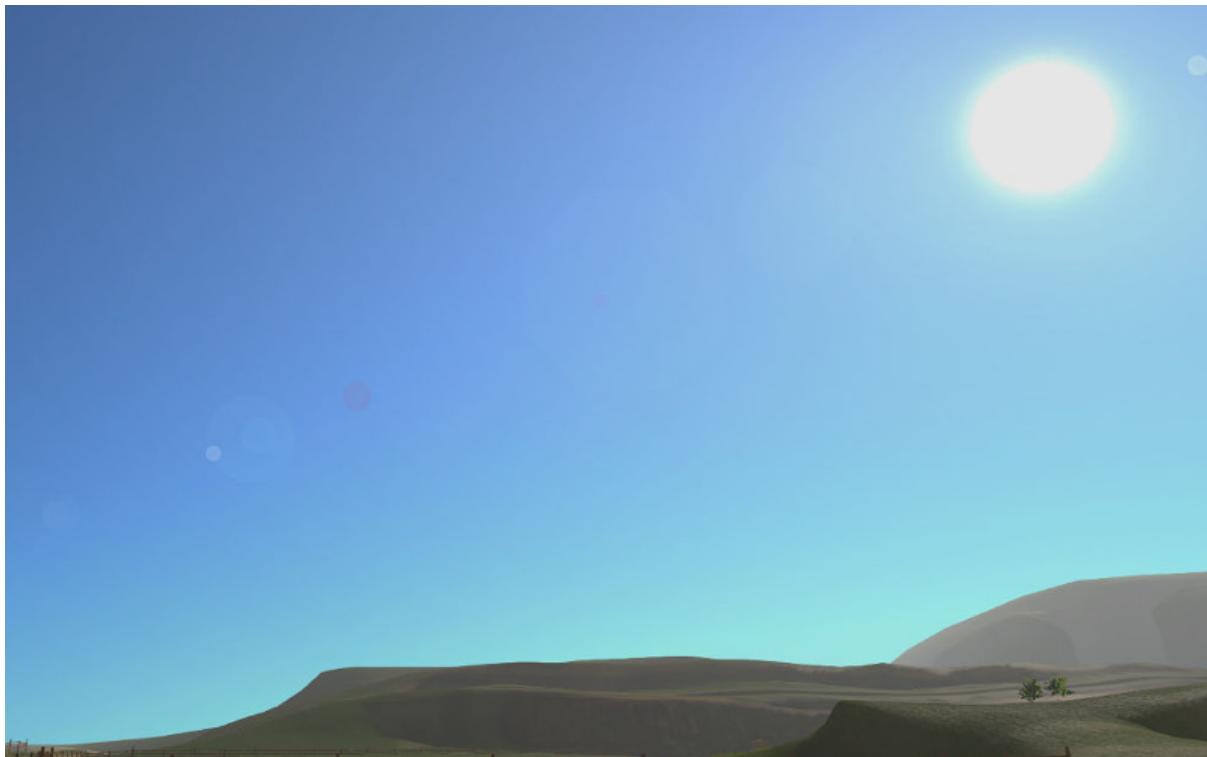
### Рассеивание

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.

## Туман

Движком поддерживаются стандартные параметры вкладки **World > Mist**. Помимо этого возможно переопределение цвета тумана.

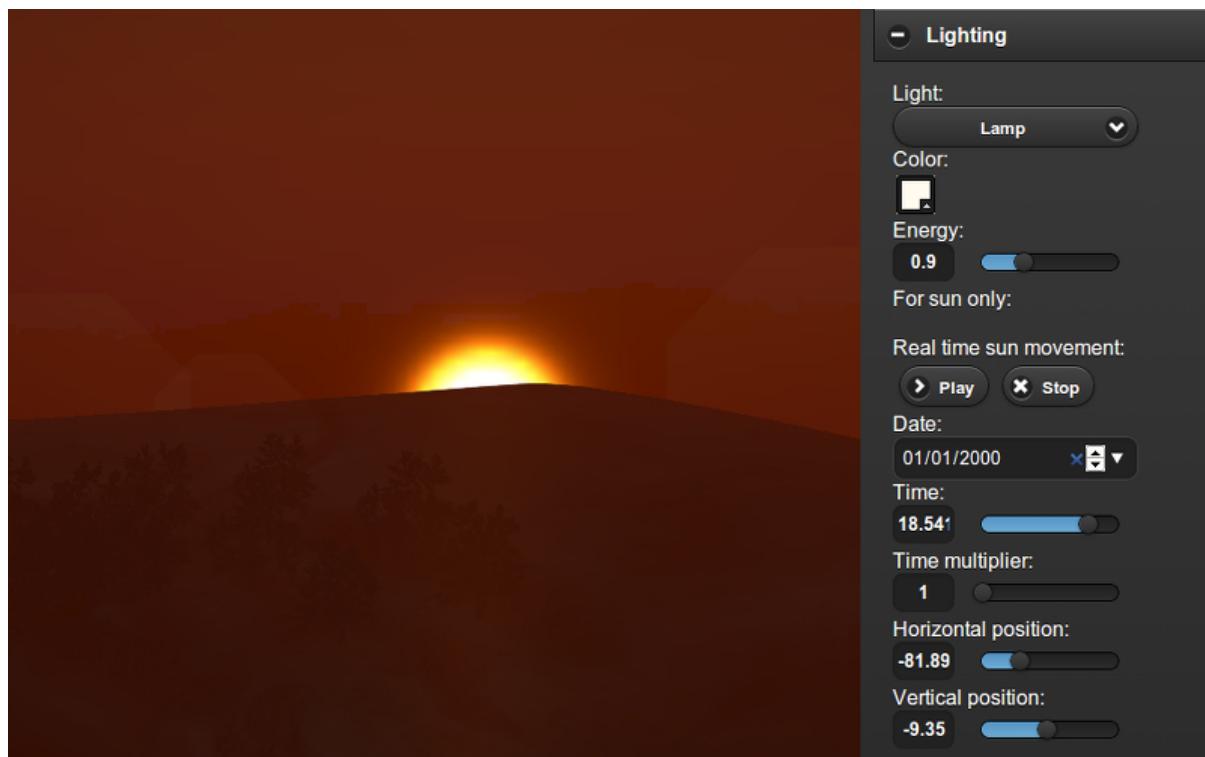
Настройки тумана описаны в главе [Параметры сцены](#).

При использовании динамического неба цвет тумана определяется цветом неба.

## 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**.

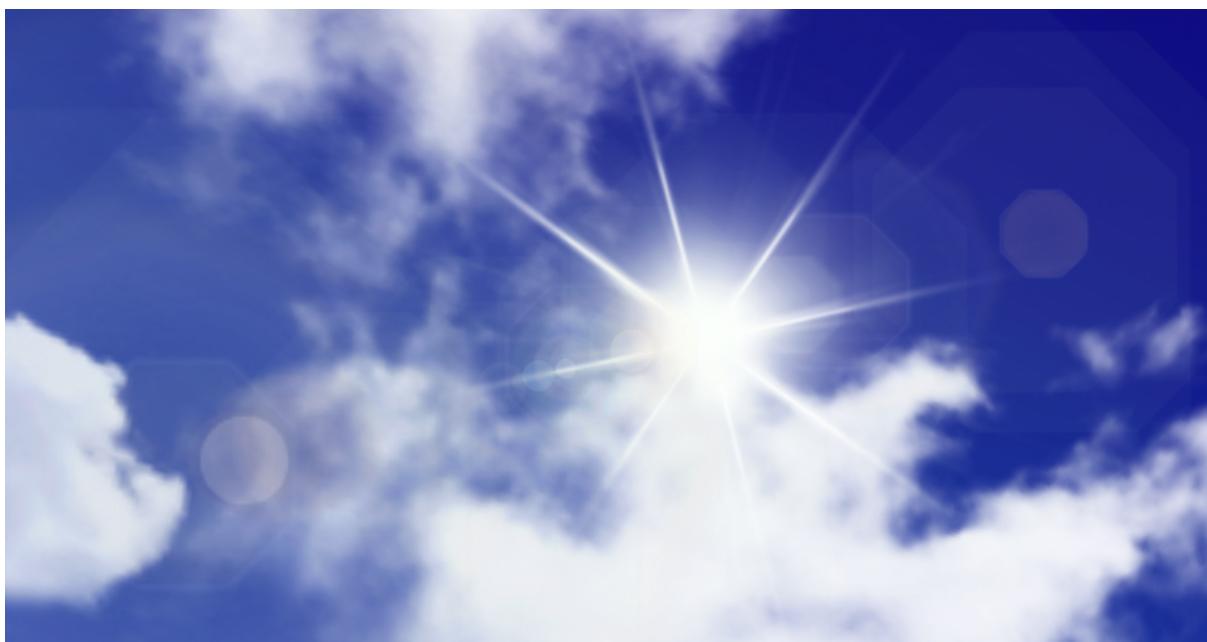


## Звезды

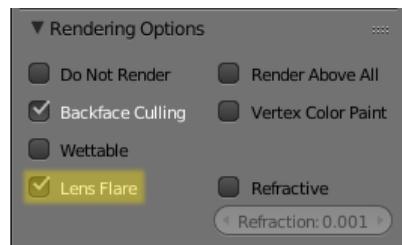
Настройка данного эффекта описывается в разделе *Материалы гало (Halo)*.



## Lens Flare



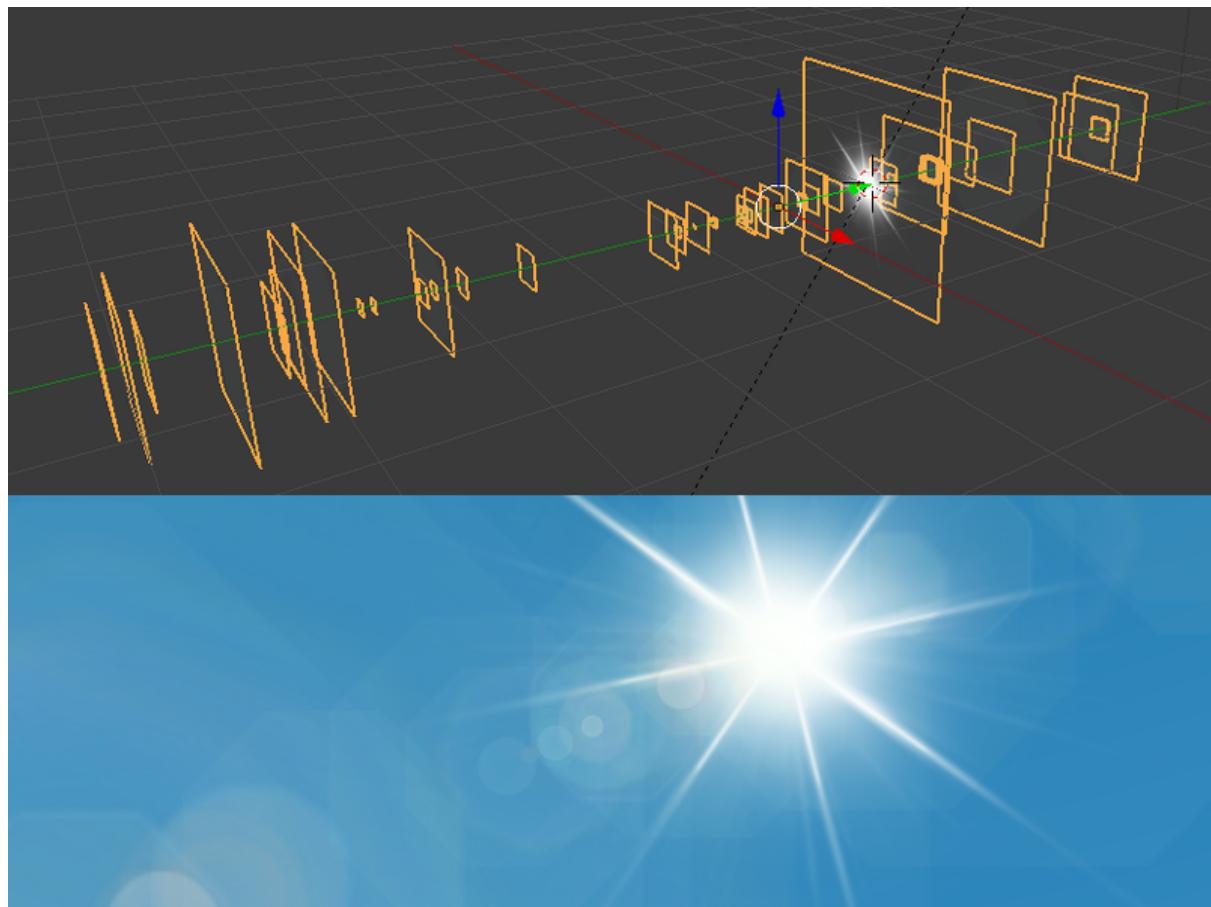
Включается с помощью опции **Lens Flare** на вкладке **Rendering Options** панели **Material**.



**Примечание:** Этот эффект работает только в том случае, если в сцене присутствует источник света типа Sun.

## Настройки эффекта

### Объект для бликов линз



Объект используется при визуализации эффекта. По сути, такой объект представляет собой набор плоскостей, каждая из которых содержит один из элементов блика. Нормаль каждой плоскости должна быть направлена в сторону -Y. Каждый элемент блика может двигаться по экрану с поворотом камеры, единственным исключением является центральный элемент (яркая белая "звезда" на изображении выше). Эта плоскость считается центром объекта, и её локальная координата Y должна быть равна 1.

Расположение и пространственная ориентация самого объекта при визуализации не учитывается.

Для корректной работы объекта следует включить опцию **Disable Frustum Culling**.

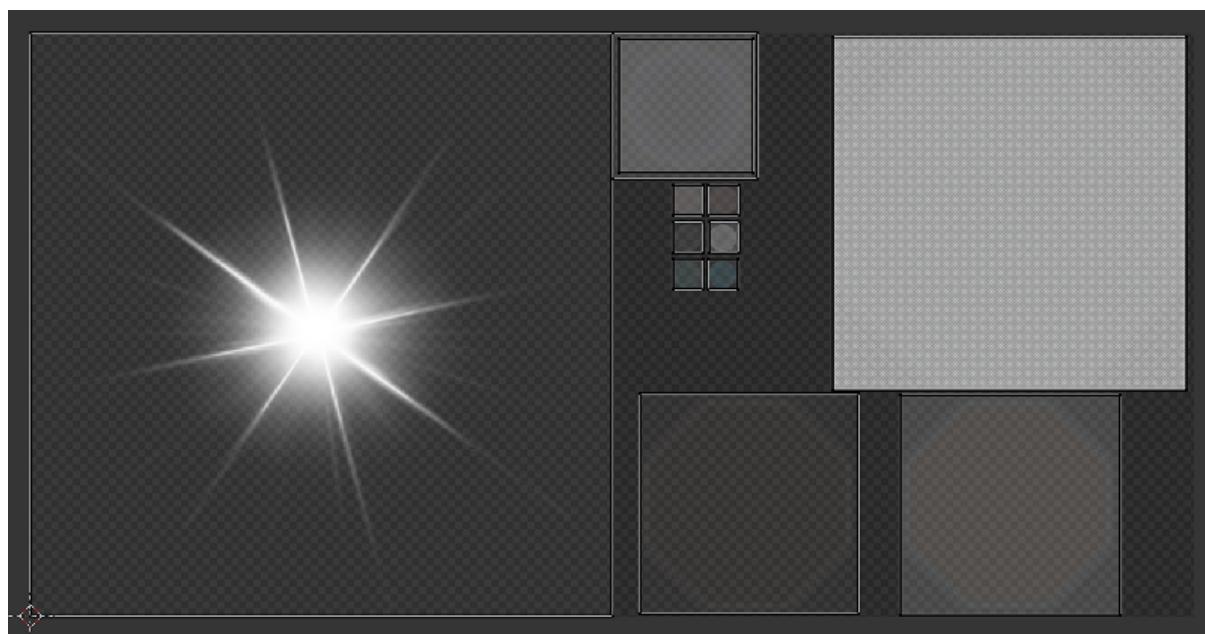
Пример использования подобного объекта можно увидеть в нашем демо **Island**.

## Настройки материала

Материал объекта для бликов линз должен иметь тип прозрачности **Alpha Blend**. Значение параметра **Alpha** должно быть равно нулю.

*Нодовые материалы* не поддерживаются.

## Настройки текстуры



Пример использования текстуры для создания эффекта линз.

Все элементы, используемые для изображения блика в приложении, должны находиться на одной текстуре. Текстура должна обладать каналом прозрачности и влиять на значения **Color** и **Alpha**.

## 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)

## Активация

Добавить на сцену объект - силовое поле типа Wind.

## Настройка

**Направление** Направление задается посредством вращения объекта - силового поля.

**Force Fields > Strength** Сила ветра. Располагается во вкладке Physics. Значение по умолчанию 1.0.

## Анимация травы и крон деревьев

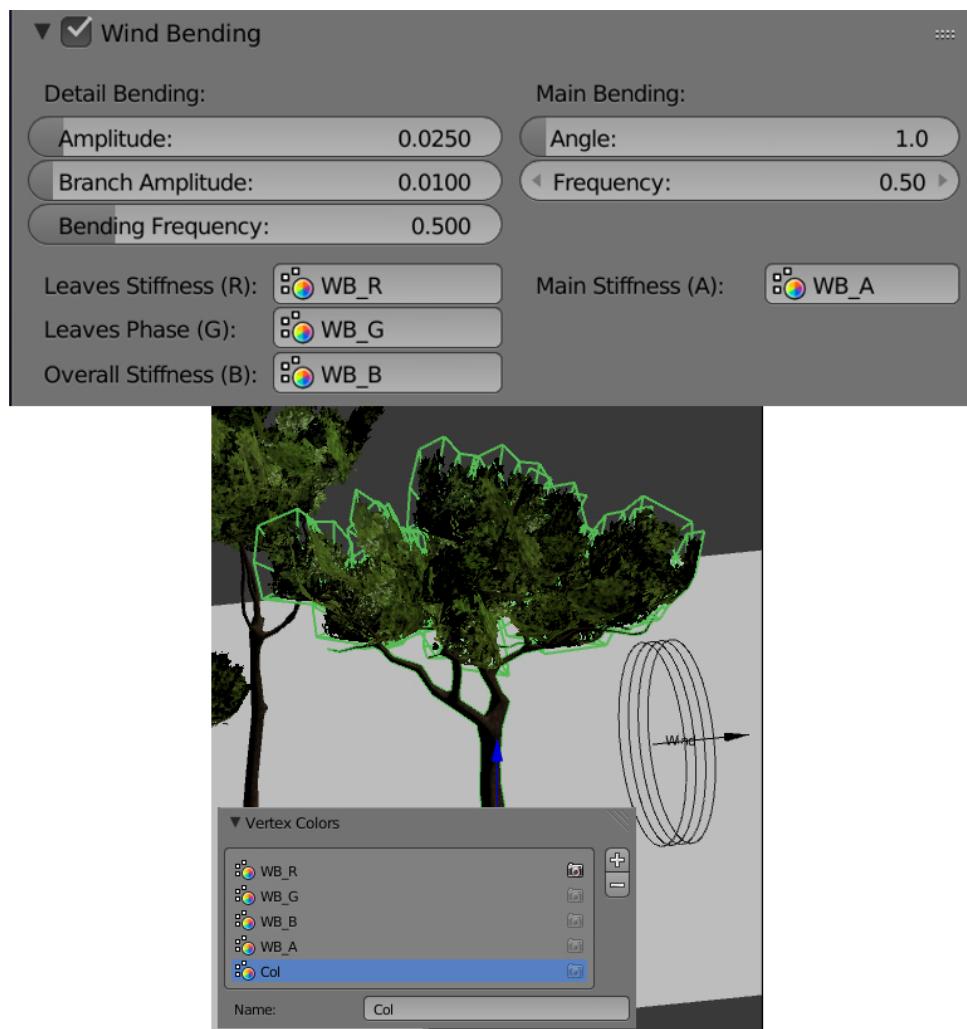
Подготовка ресурсов для рендеринга травы описана в разделе *Травяной покров*.

## 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 вертексного цвета, содержащего информацию о жесткости “основного” отклонения. Может быть оставлено пустым.

**Detail Bending > Amplitude** Амплитуда угла “детализированного” 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.

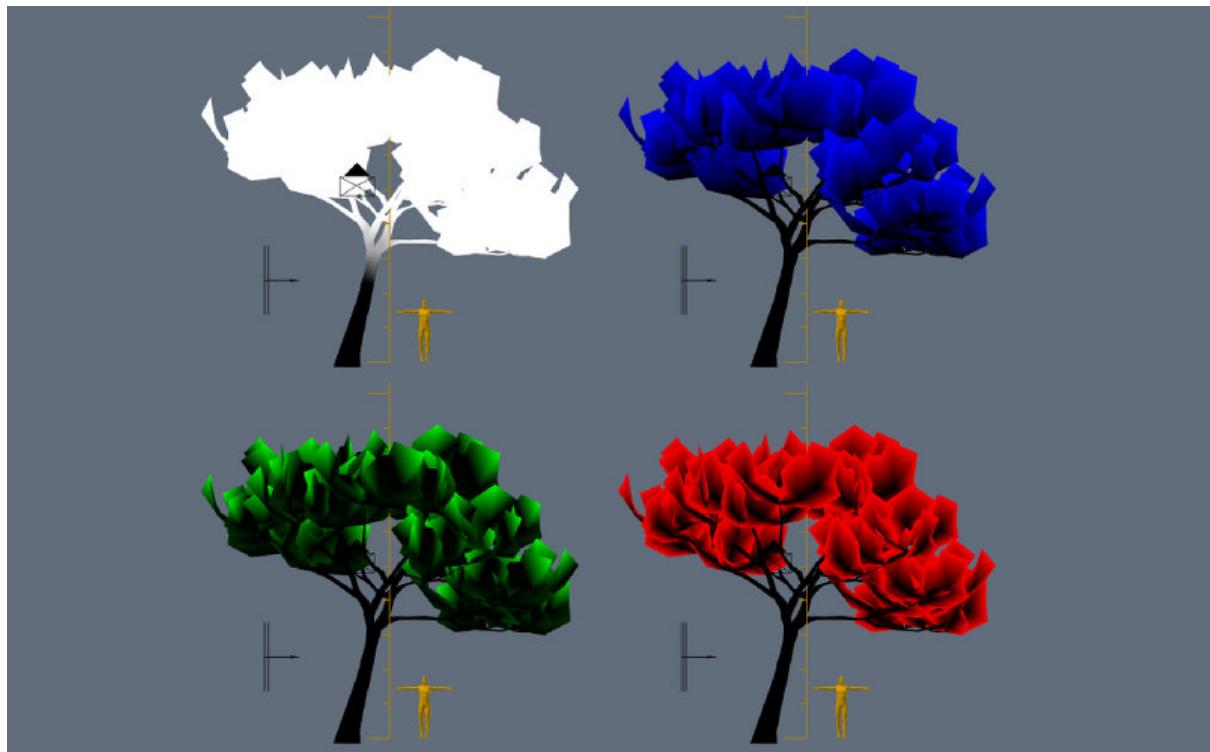
**Detail Bending > Leaves Stiffness (R)** Текстовое поле для названия слоя вер-

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.



## Работа с цветом

---

### Содержание

- Работа с цветом
  - Общее описание
  - Человеческое зрение и мониторы
  - Гамма
  - Коррекция в нодовых материалах
    - \* Ноды для окраски
    - \* Ноды для масок
    - \* Карты нормалей
    - \* Сводная таблица коррекции в нодовых материалах
  - Альфа-композитинг
    - \* Общие сведения
    - \* Реализация
  - Коррекция цвета
    - \* Активация
    - \* Дополнительные настройки

### 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`.

## 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.

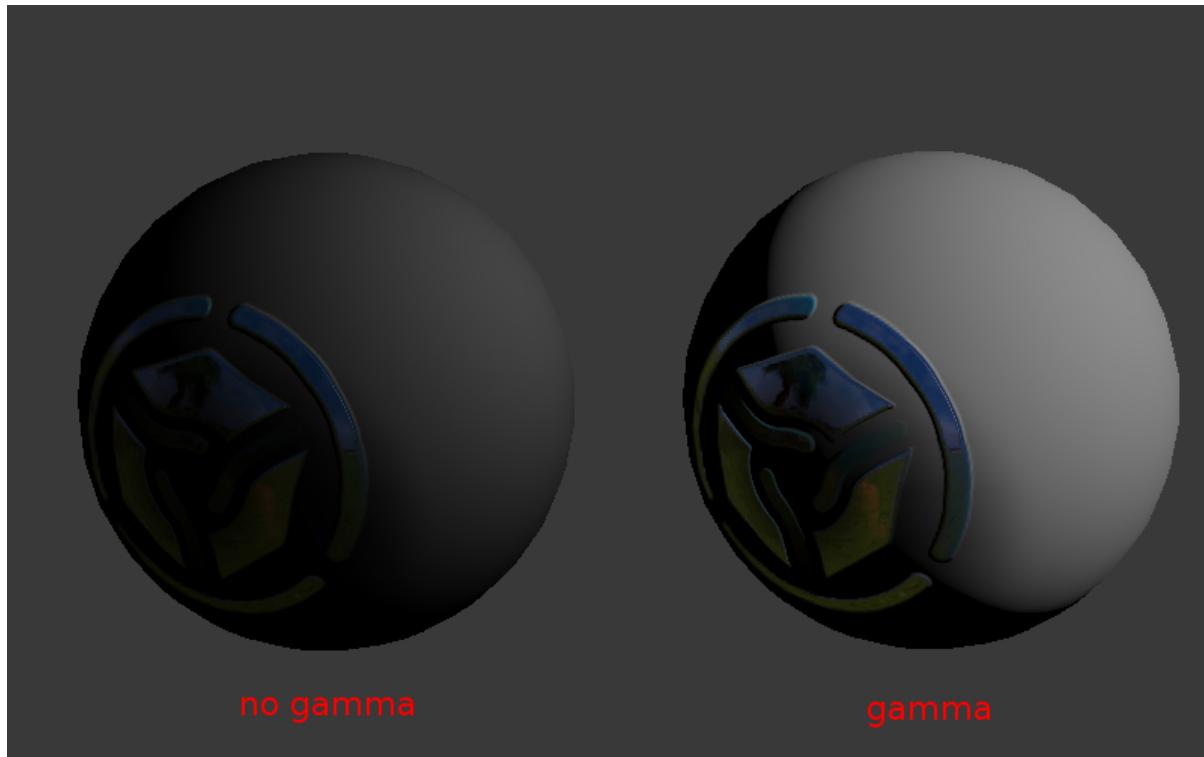
Таким образом, особенности восприятия человеческого глаза являются причиной, по которой стало возможным осуществлять упаковку каналов цветности, в то время как технические характеристики мониторов имеют вторичное значение по отношению к гамма-коррекции.

## 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  $\rightarrow$  sRGB) и “распаковка” (sRGB  $\rightarrow$  Linear).



## Коррекция в нодовых материалах

### Ноды для окраски

При использовании текстур и вертексы цветов для окраски (не в качестве масок), необходима распаковка (sRGB  $\rightarrow$  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.

### Ноды для масок

Текстуры и вертексы 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*.

---

## Карты нормалей

При использовании карт нормалей никакие преобразования не производятся.

## Сводная таблица коррекции в нодовых материалах

| Случай использования        | Коррекция   |
|-----------------------------|---|
| Текстура для окраски        | реализуется автоматически в ноде 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  |

## Альфа-композитинг

### 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.

## 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  $\alpha$ .

## Коррекция цвета



## Активация

Активировать панель Color Correction во вкладке Render.

## Дополнительные настройки

*Brightness* Яркость. Значение по умолчанию 0.0.

*Contrast* Яркость. Значение по умолчанию 0.0.

*Exposure* Экспозиция. Значение по умолчанию 1.0.

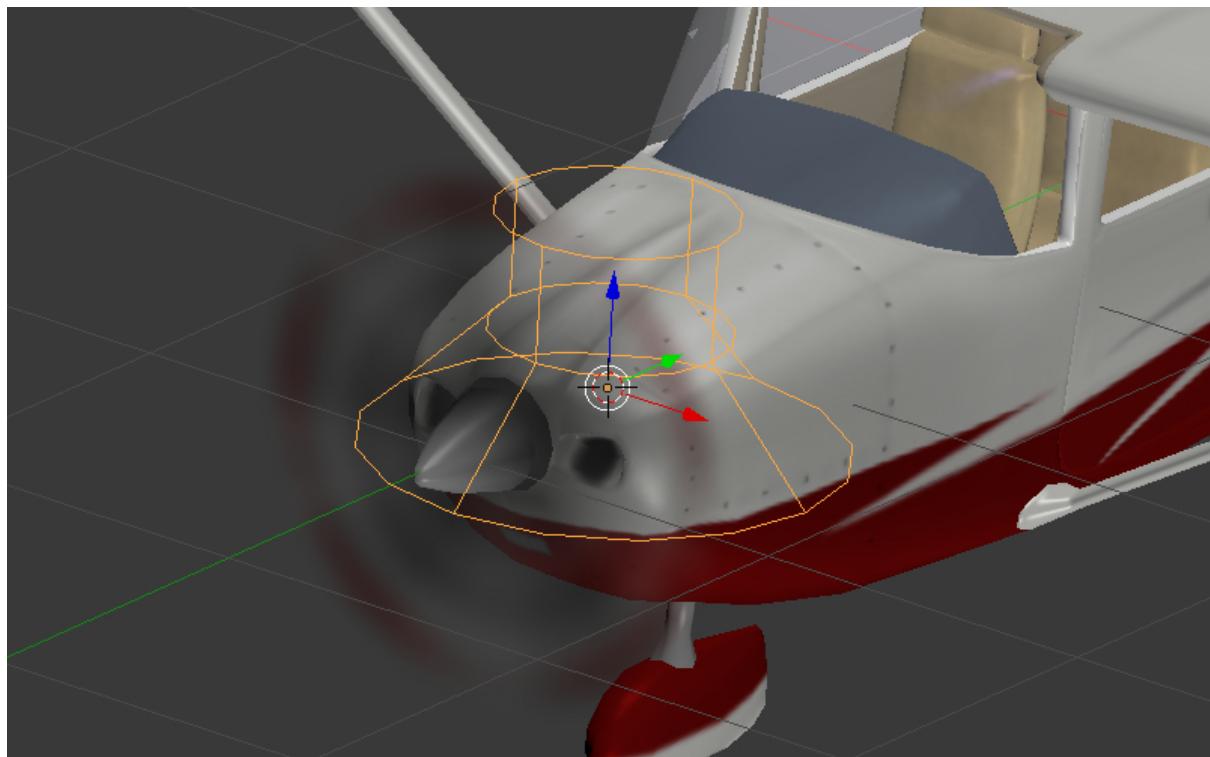
*Saturation* Экспозиция. Значение по умолчанию 1.0.

## Звук

### Содержание

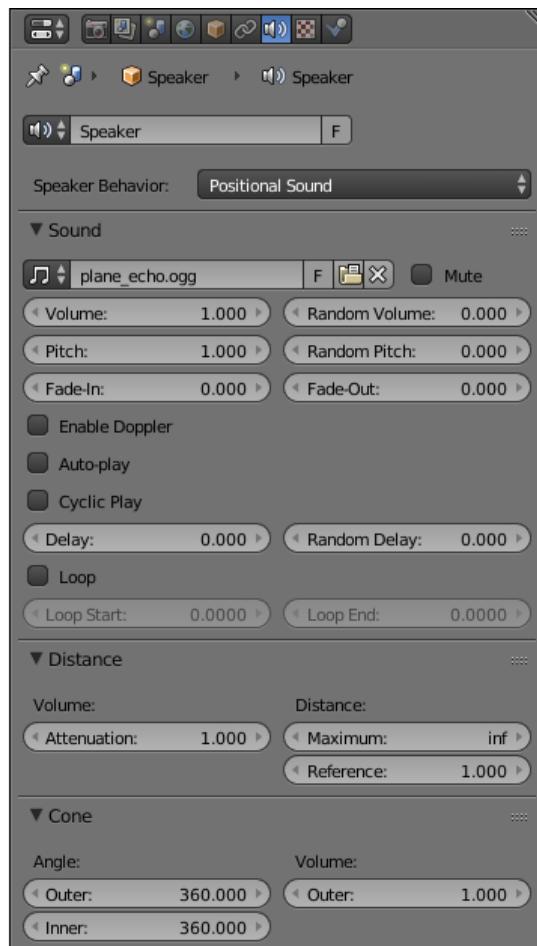
- *Звук*
  - Настройка звуковых источников
    - \* Вкладка *Sound*
    - \* Вкладка *Distance*
    - \* Вкладка *Cone*
  - Микшер
  - Обработка и кодирование
    - \* Поддерживаемые форматы (контейнеры):

Создание звуковых источников осуществляется в Blender'e. Используется стандартный объект **Speaker**.



## 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.

### Вкладка Sound

#### *Mute*

Отключает воспроизведение звука.

#### *Speaker Behavior:*

Поведение звукового источника.

**Positional** — высококачественный звук, допускающий позиционирование и имеющий направленность (конусность). Для рендеринга используется Web Audio API. Воспроизведение подобных звуков обладает наи-

меньшей производительностью, поэтому их использовать целесообразно только для коротких сэмплов. Это значение выбрано по умолчанию.

**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.

*Enable Doppler*

Включает смещение частоты источника при его перемещении.

*Auto-play*

Включает воспроизведение по умолчанию.

*Cyclic Play*

Зацикливать воспроизведение звука.

*Loop*

Зацикливать 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*.

*Loop Start*

Задаёт начальную точку фрагмента звукового файла, который будет зациклен. Измеряется в секундах, отсчитываемых от начала звукового файла.

#### *Loop End*

Задаёт начальную точку фрагмента звукового файла, который будет зациклен. Также измеряется в секундах.

#### *Delay*

Задержка в секундах перед началом проигрывания звука.

#### *Random Delay*

Дополнительная рандомизация задержки, результирующее значение определяется по формуле  $Delay_{result} = Delay + Delay_{random} * Random_{[0-1]}$ .

## Вкладка Distance

#### *Attenuation*

Задаёт силу, с которой расстояние влияет на громкость. По умолчанию это значение равно 1.0.

#### *Maximum*

Максимальная дистанция для расчёта громкости.

#### *Reference*

Задаёт эталонное расстояние, на котором громкость звука равна 100% от общего значения.

## Вкладка Cone

Группа параметров Angle:

#### *Outer*

Угол внешнего конуса (в градусах). За пределами этого конуса громкость определяется параметром *Outer* группы *Volume*. Между границами внешнего и внутреннего конусов значение громкости интерполируется.

#### *Inner*

Угол внутреннего конуса (в градусах). Внутри этого конуса громкость составляет 100%.

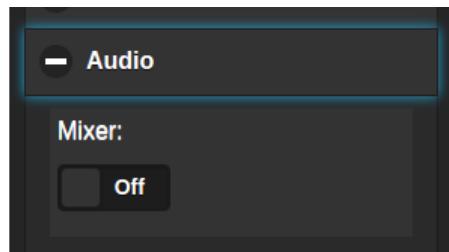
Группа параметров Volume:

#### *Outer*

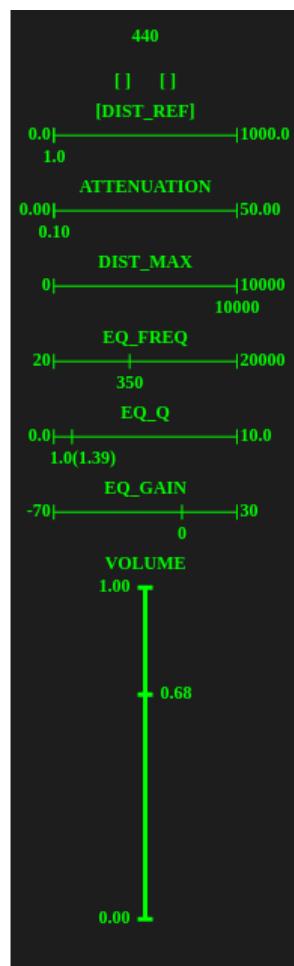
Задаёт значение громкости за пределами внешнего конуса.

## Микшер

Этот инструмент - часть интерфейса *Просмотричика сцен*.



Его включение вызывает интерфейс эквалайзера, содержащий различные параметры источников звука, присутствующих в сцене.



Доступные параметры:

**DIST\_REF** Этот параметр задаёт эталонную дистанцию для выбранного источника звука. Доступен только для источников типа Positional Sound.

**ATTENUATION** Этот параметр задаёт степень затухания звука, издаваемого вы-

бранным спикером. Доступен только для источников типа Positional Sound.

***DIST\_MAX*** Это значение задаёт максимальную дистанцию для выбранного спикера. Если расстояние от камеры до спикера превышает это расстояние, спикер не будет издавать звука. Доступно только для источников типа Positional Sound.

***EQ\_FREQ*** Центральная частота, к которой применяется усиление.

***EQ\_Q*** Фактор Q. Это значение управляет шириной полосы усиливаемых частот. Увеличение значения сужает полосу.

***EQ\_GAIN*** Это значение задаёт усиление (измеряемое в dB), которое применяется к звуку, издаваемому спикером.

***VOLUME*** Громкость звука, издаваемого спикером.

Ниже этого списка параметров демонстрируется частотная характеристика параметрического эквалайзера.

Микшер управляется с помощью дополнительной цифровой клавиатуры.

- Клавиши 8, 4, 6 и 2 работают так же, как клавиши-стрелки, и используются для переключения между источниками звука и выбора их параметров.
- Клавиши + и - увеличивают и уменьшают значение выбранного параметра.
- Клавиша 7 выключает выбранный спикер.
- Клавиша 9 выключает все спикеры, кроме выбранного.

## Обработка и кодирование

### 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*.

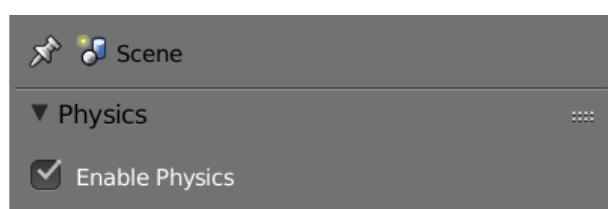
## Физика

### Содержание

- *Физика*
  - *Подготовка к использованию*
  - *Статический тип физики*
  - *Динамический тип физики*
  - *Ограничители (Constraints)*
  - *Колесные транспортные средства*
    - \* *Параметры настройки для шасси*
  - *Плавающие объекты*
    - \* *Параметры настройки плавающего объекта*
  - *Плавающие транспортные средства*
    - \* *Параметры настройки плавающего транспортного средства*
  - *Персонажи*
    - \* *Параметры настройки объекта-персонажа*
    - \* *Базовый персонаж*
    - \* *Управление персонажем с помощью API*
  - *Навигационные меши*
    - \* *Создание навигационных мешей*
    - \* *Использование навигационных мешей*
  - *Особенности использования в приложениях*

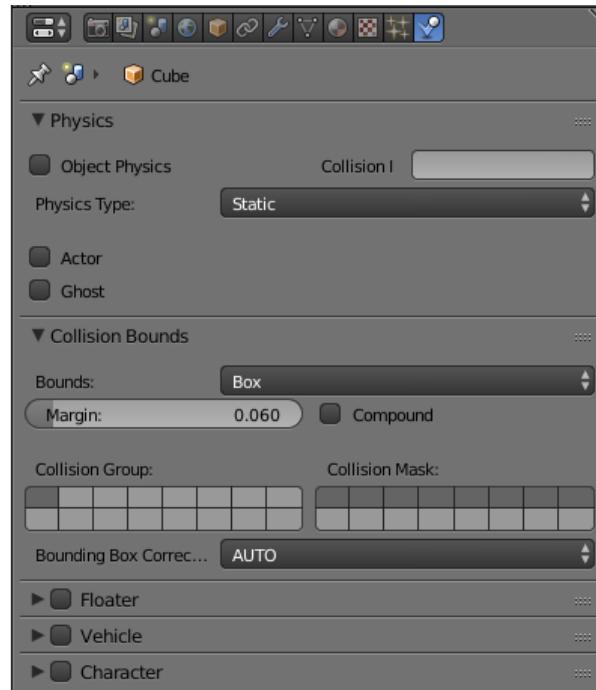
## Подготовка к использованию

Для задействования физики на сцене необходимо установить флаг `Enable Physics` в панели `Physics` на сцене.

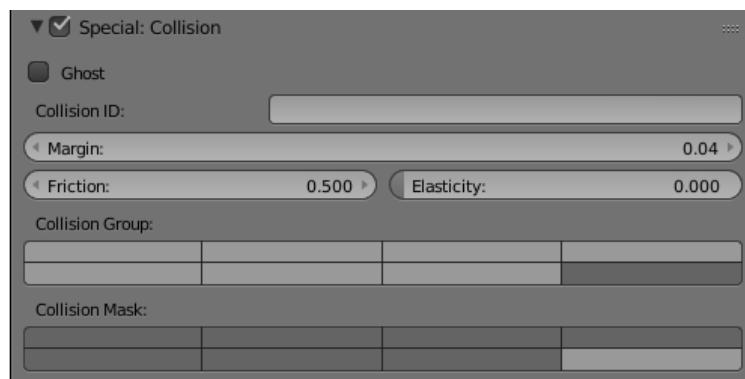


## Статический тип физики

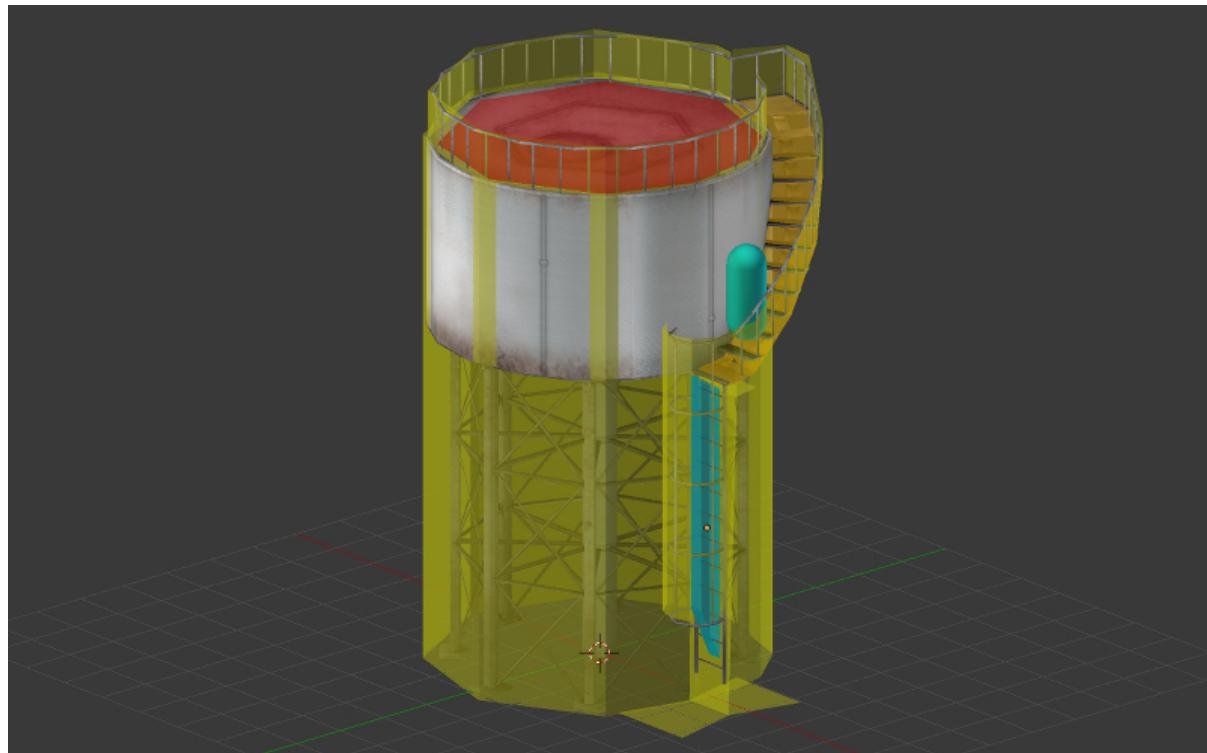
Может использоваться как ограничитель движения других объектов, например, для определения столкновений с ландшафтом, стенами и т.д. В настройках физики такого объекта для опции Physics Type должно быть выбрано значение **Static** (значение по умолчанию).



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.

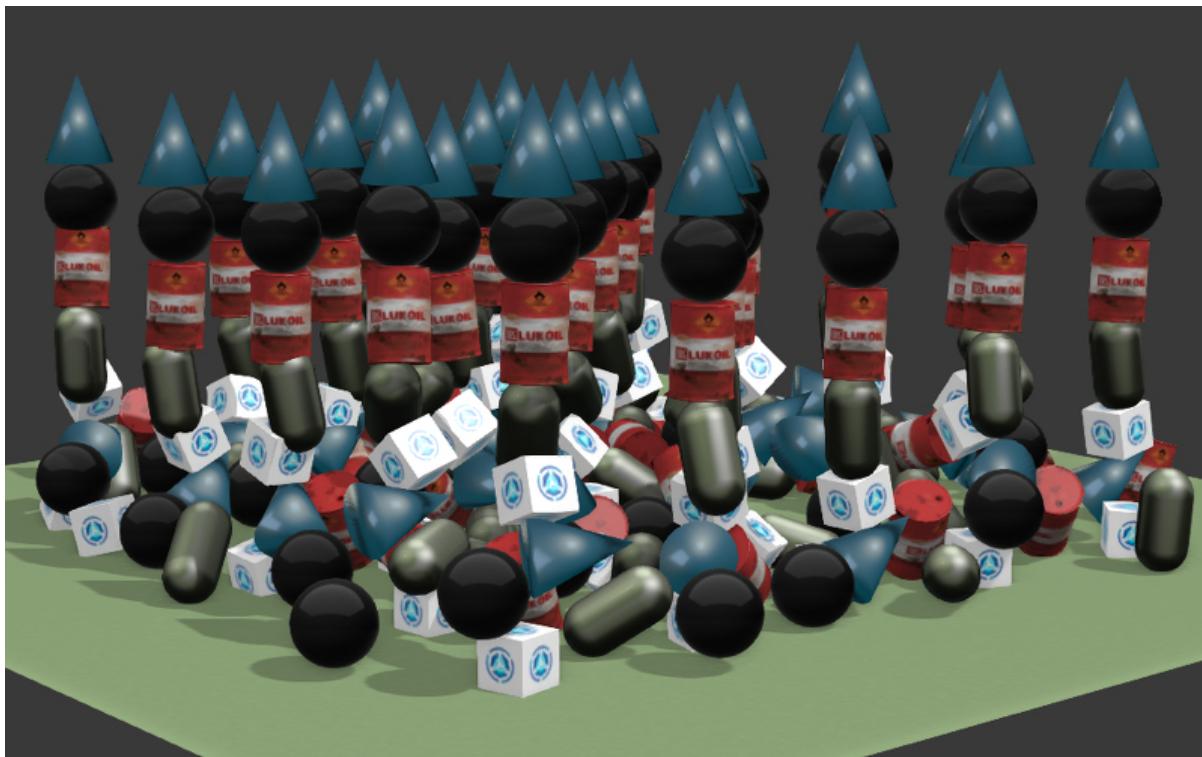
Поле **Margin** позволяет настроить ширину зоны, в которой происходит соударения с сеткой данного объекта. Данная настройка позволяет улучшить стабильность симуляций столкновений объектов.

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.

## Динамический тип физики

Предназначен для симуляции движения жесткого тела.

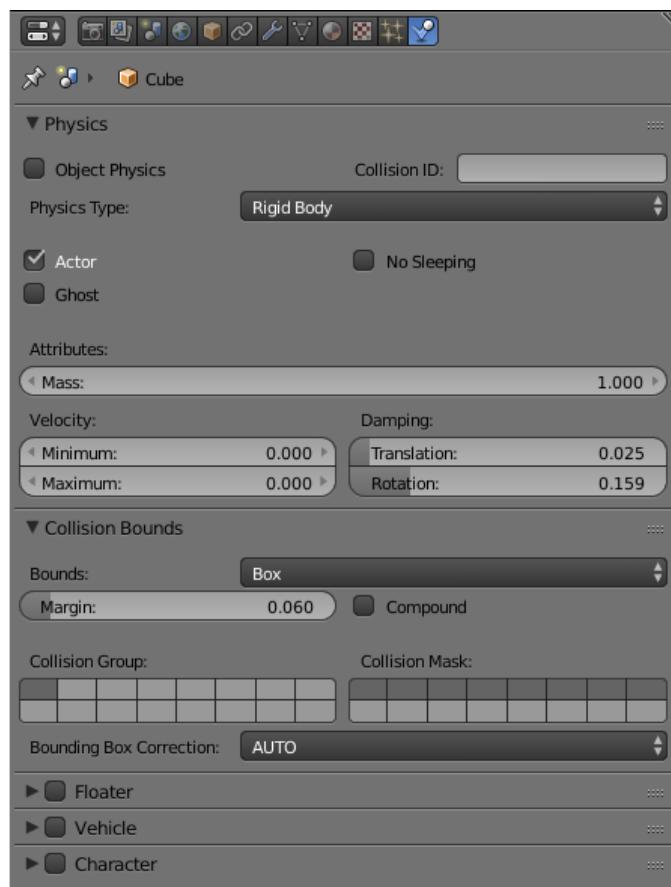


В настройках панели **Physics** такого объекта должен быть установлен флаг **Object Physics**. Поле **Collision ID** предназначено для определения столкновения со специфическим объектом (например, прикрепленный к камере объект для определения близости FPS персонажа к предметам), и может быть оставлено 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.

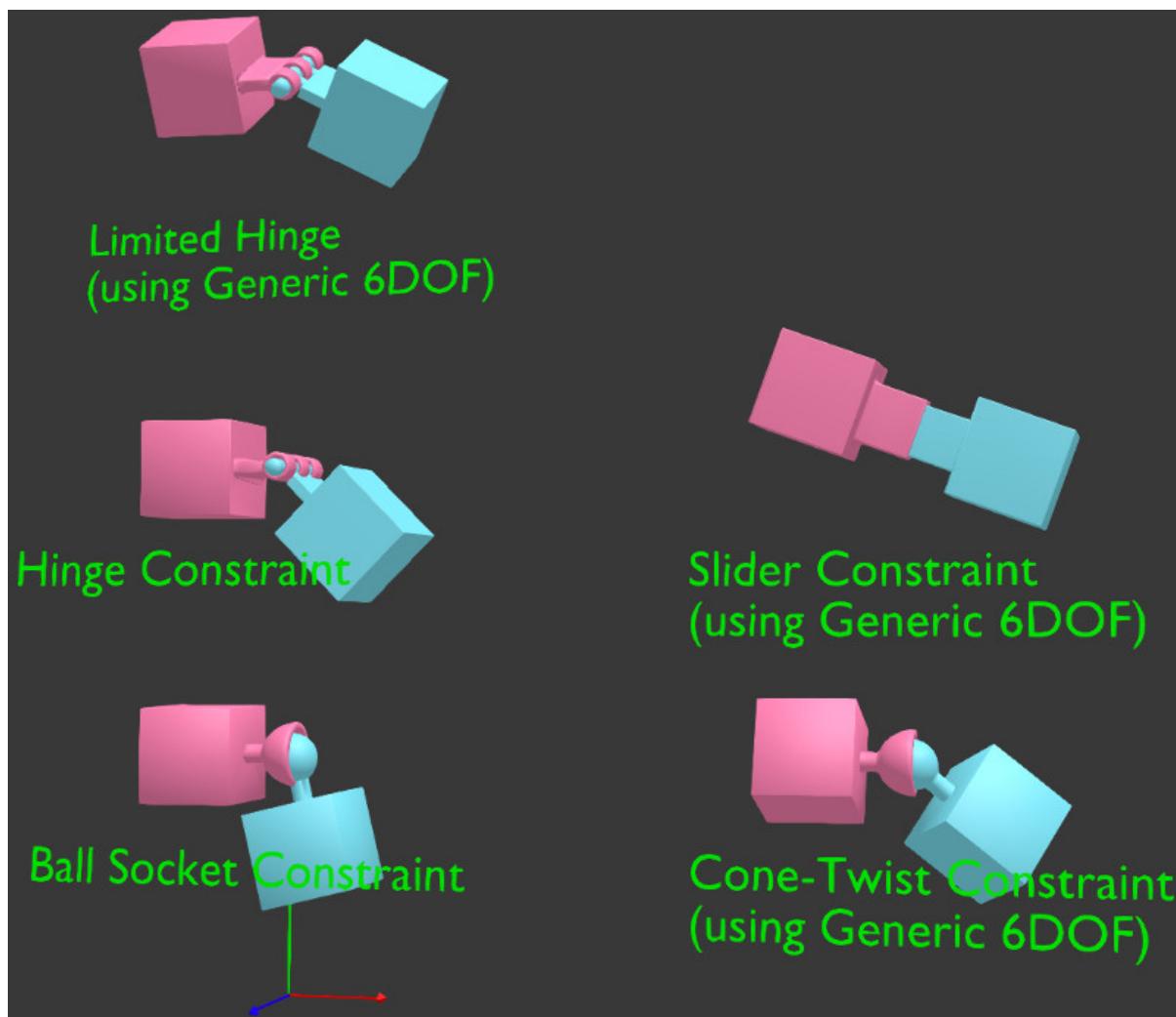


Для материала такого объекта поддерживаются: трение (**Friction**), упругость (**Elasticity**). В случае использования на одном меше нескольких материалов физические настройки считаются с первого из них.

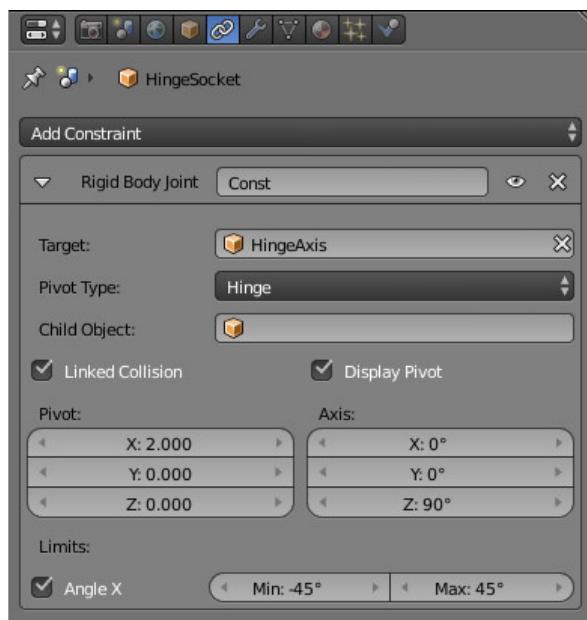
Для объекта-камеры должна использоваться настройка **Physics Type = Dynamic**, должен быть установлен флаг **Object Physics**.

## Ограничители (Constraints)

Физические ограничители используются для уменьшения числа степеней свободы объектов.

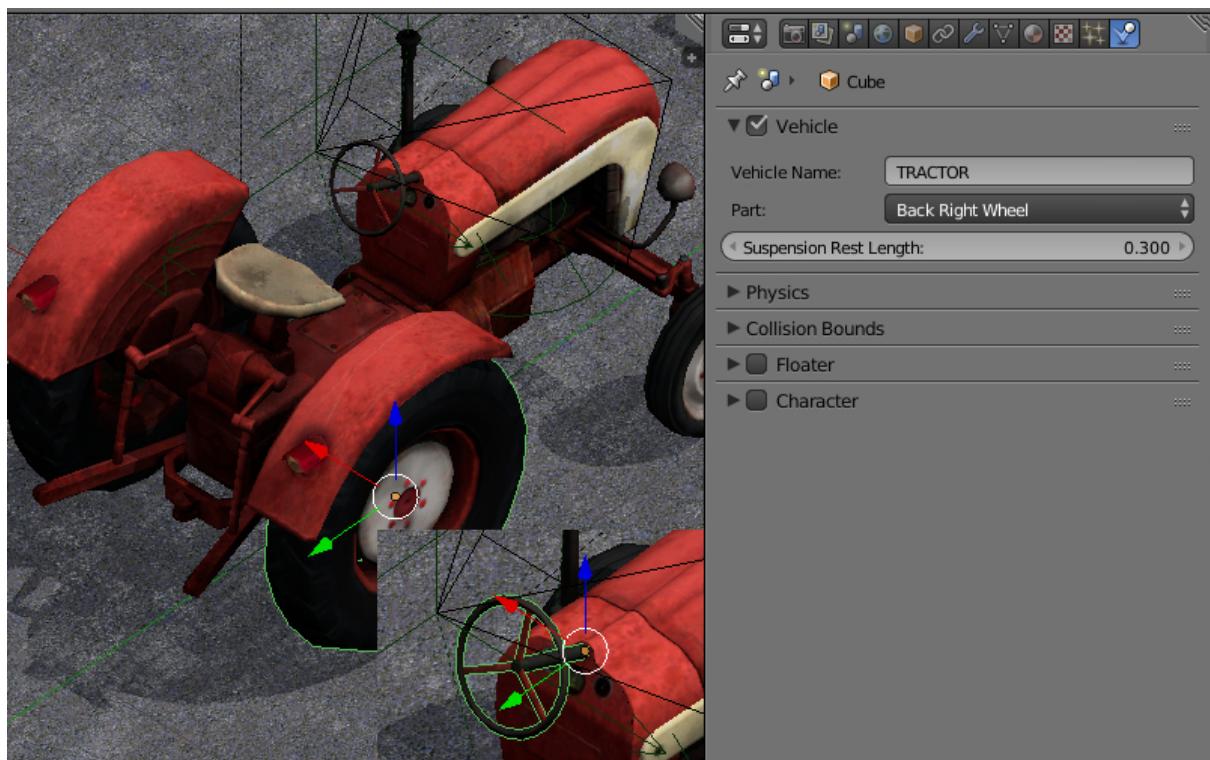


Установка физического ограничителя (Rigid Body Joint) на объект происходит в панели **Object Constraints**. Поддерживаемые типы (Pivot Type): **Ball**, **Hinge**, **Cone Twist**, **Generic 6 DoF**. Физический ограничитель можно установить на один из двух взаимодействующих объектов, при этом другой выступает 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.



## 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**.

## 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.

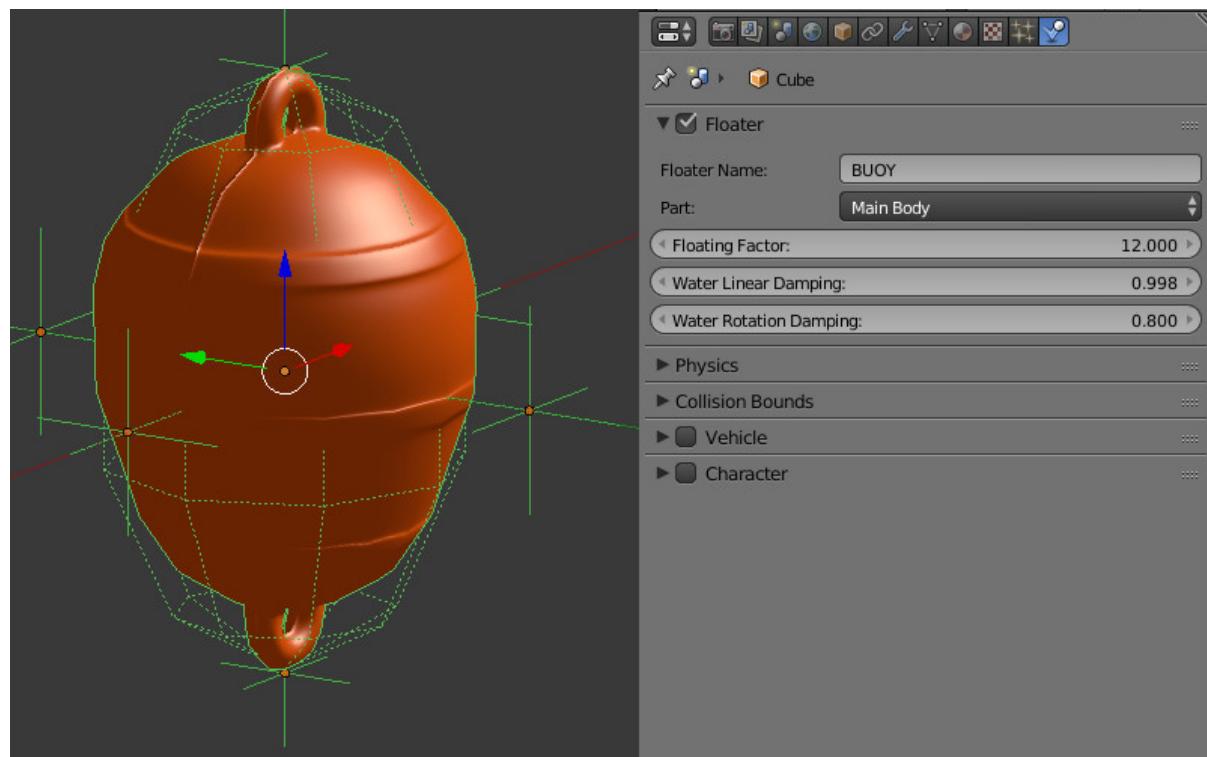
Для рулевого колеса(Steering Wheel) необходимо указать максимальный угол поворота(Steering Max) и передаточное отношение угла поворота руля к передним колесам (Steering Ratio). Максимальное значение угла поворота указывается в оборотах. Один оборот равен 360 градусам. Таким образом, поставив Steering 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.

## 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**

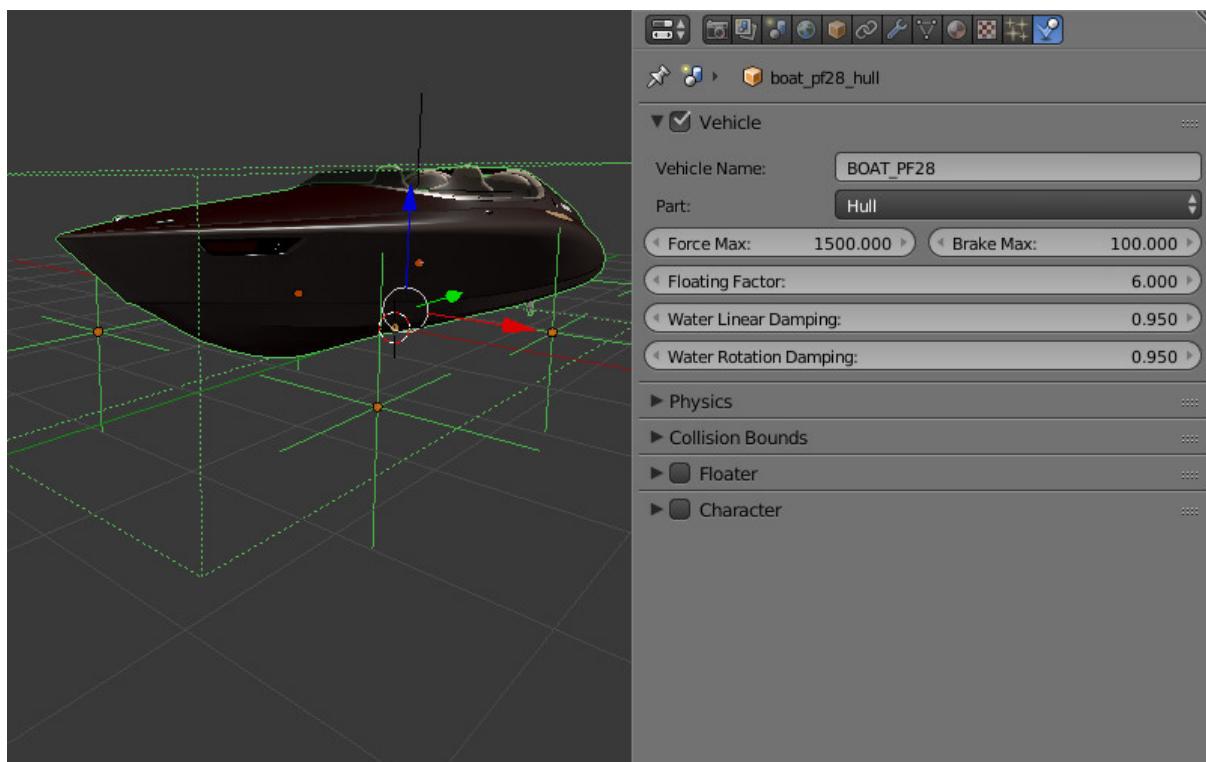
## Параметры настройки плавающего объекта

**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.

## 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**.

### Параметры настройки плавающего транспортного средства

**Force Max** Максимальная движущая сила транспортного средства.

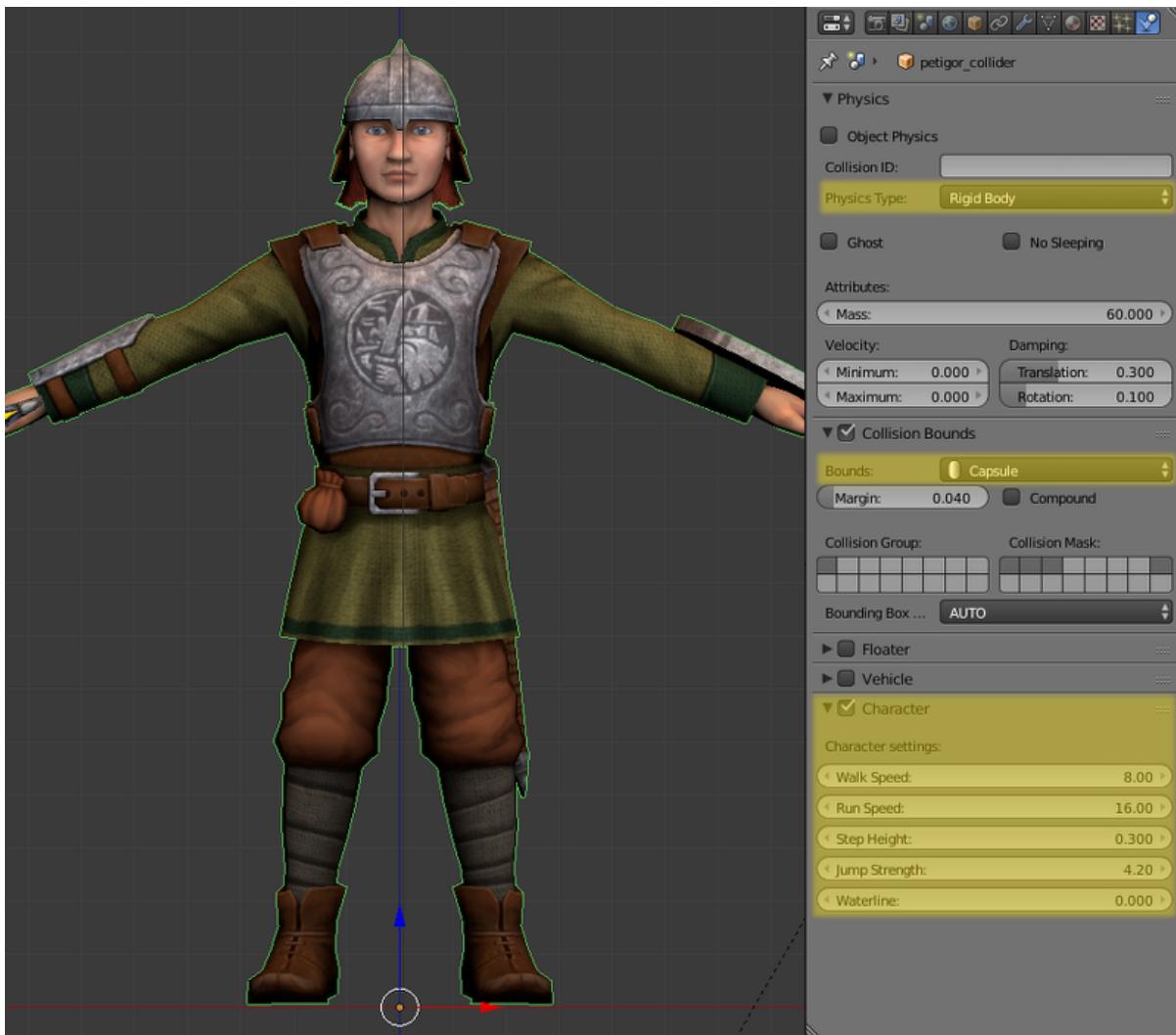
**Brake Max** Максимальный коэффициент торможения.

**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.

## Персонажи

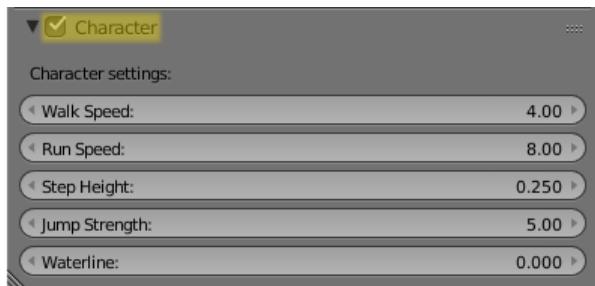


Эта возможность включается с помощью флагка **Character** в заголовке одноимённой вкладки на панели **Physics**.

С точки зрения движка персонаж представляет собой полноценный физический объект, который может сталкиваться с другими объектами, и на который могут воздействовать физические силы, такие как гравитация и сила Архимеда.

Для управления объектом-персонажем предусмотрено несколько методов API, содержащих слово *character* в названии. Эти методы находятся в модуле API [physics](#).

## Параметры настройки объекта-персонажа



Все настройки, перечисленные в этом разделе, становятся доступны при включении панели **Character**

**Walk Speed** Задаёт скорость ходьбы персонажа (измеряется в метрах в секунду).

Минимальное возможное значение параметра - 0, максимальное - 10.

Значение по умолчанию - 4.

**Run Speed** Скорость бега персонажа (измеряется в метрах в секунду). Значение может изменяться от нуля до 20.

По умолчанию равна 8.

**Step Height** Максимальная высота (в метрах) препятствия, которое может преодолеть персонаж. Значение этого параметра может изменяться от 0 до 1.

Значение по умолчанию - 0.250.

**Jump Strength** Сила прыжка персонажа. Минимальное значение - 0, максимальное - 50.

По умолчанию равна 5.

**Waterline** Этот параметр устанавливает ватерлинию для объекта-персонажа. Расстояние до ватерлинии отсчитывается от центра физического примитива объекта (не от центра меша). Если объект погружен в воду ниже ватерлинии, он будет плавать. Значение параметра может изменяться от -2 до 2.

Значение по умолчанию равно нулю.

## Базовый персонаж

Движок Blend4Web также предоставляет возможность быстрой настройки персонажа, управляемого пользователем от первого лица. Такой персонаж обладает только самыми основными функциями, но его создание не требует программирования.

Для создания базового персонажа следуйте инструкции:

1. Выберите объект, который вы хотите использовать в качестве персонажа.
2. Включите физику для выбранного объекта и установите тип физики на **Dynamic** или **Rigid Body**.
3. Включите для объекта опцию **Character**.

4. Создайте новую камеру или выберите уже имеющуюся, и установите тип камеры Eye.

Теперь в сцене будет присутствовать базовый персонаж с прикреплённой камерой. Такой персонаж управляется практически так же, как обычная камера типа Eye:

- Клавиши WASD перемещают персонажа.
- Направление взгляда управляется с помощью клавиш-стрелок или перемещения мыши с зажатой левой кнопкой.
- Нажатие клавиши С включает и выключает режим полёта (включён по умолчанию).

---

**Примечание:** В сцене может присутствовать только один базовый персонаж. Если базовых персонажей больше одного, движок будет использовать первого из них и игнорировать остальных.

---

## Управление персонажем с помощью API

Базовые персонажи обладают только стандартным набором возможностей. Если вам требуется более тонкая настройка поведения персонажа, следует использовать методы модуля `fps`.

Наиболее важный из них - метод `enable_fps_controls()`, который, как понятно из его названия, используется для инициализации персонажей в сцене. Он может быть использован следующим образом:

```
var m_fps = require("fps");
m_fps.enable_fps_controls();
```

Для включения управления персонажем этот метод следует вызывать при запуске приложения (из функции `load_cb`).

---

**Примечание:** Метод может конфликтовать с методом `enable_camera_controls()`, который используется для активации базового персонажа, описанного в предыдущем разделе. Эти два метода не следует использовать одновременно.

---

Опциональные параметры метода:

`character` - задаёт объект-персонаж. Ссылку на объект можно получить при помощи метода `get_first_character()`. Следует использовать, если в сцене присутствует более одного персонажа.

`element` - определяет HTML-элемент, к которому метод добавляет сенсоры.

`motion_cb` задаёт callback-функцию, которая вызывается, когда персонаж меняет направление движения.

`gamepad_id` - задаёт ID геймпада, подключённого к системе.

`forward_sens`, `backward_sens`, `right_sens`, `left_sens`, `jump_sens`, `fly_sens` - эти параметры используются для задания массивов типов сенсоров для различных действий персонажа (например, ходьбы в разных направлениях, прыжков и т.п.).

`rotation_cb` задаёт callback-функцию, которая вызывается при повороте камеры.

`lock_camera` - если этот параметр имеет значение `true`, камера будет прикреплена к персонажу.

Другие важные методы: `set_cam_sensitivity()` и `set_cam_smooth_factor()`. Первый из них управляет чувствительностью камеры (измеряемой значением от 0 до 100), а второй - определяет, насколько плавно камера движется (плавность движения изменяется от 0 до 1.0).

## Состояния персонажей

Состояние персонажа - это постоянная величина, обозначающая действие, которое персонаж совершает в данный момент (т.е. идёт ли персонаж, или летит, или совершает какое-то другое действие). Каждый персонаж, присутствующий в сцене, всегда имеет одно и только одно состояние.

Доступные состояния:

`CS_CLIMB` - персонаж лезет по лестнице.

`CS_FLY` - персонаж летит.

`CS_RUN` - персонаж бежит.

`CS_STAY` - персонаж не двигается.

`CS_WALK` - персонаж идёт.

Текущее состояние персонажа можно получить при помощи метода `get_character_state()`. С помощью метода `switch_state()` персонажу может быть присвоено то или иное состояние. Пример ниже демонстрирует использование этих двух методов:

```
var m_fps = require("fps");

var current_state = m_fps.get_character_state();

if (current_state == m_fps.CS_FLY)
    m_fps.switch_state(m_fps.CS_WALK);
```

## Привязка действий

В модуле `fps` также есть средства для привязки различных действий к событиям персонажа. Для этого используется метод `bind_action()`:

```
var m_fps = require("fps");
var m_ctl = require("controls");

var action_cb = function(value) {
    console.log("Q key pressed.");
}

m_fps.bind_action(m_fps.AT_PRESSED, [m_ctl.KEY_Q], action_cb);
```

Метод обладает следующими параметрами:

- первый параметр задаёт тип действия:

`AT_CONTINUOUS` регистрирует текущее действие пользователя, например, перемещение мыши, продолжительное нажатие кнопки на клавиатуре, отключение стика геймпада и т.п.

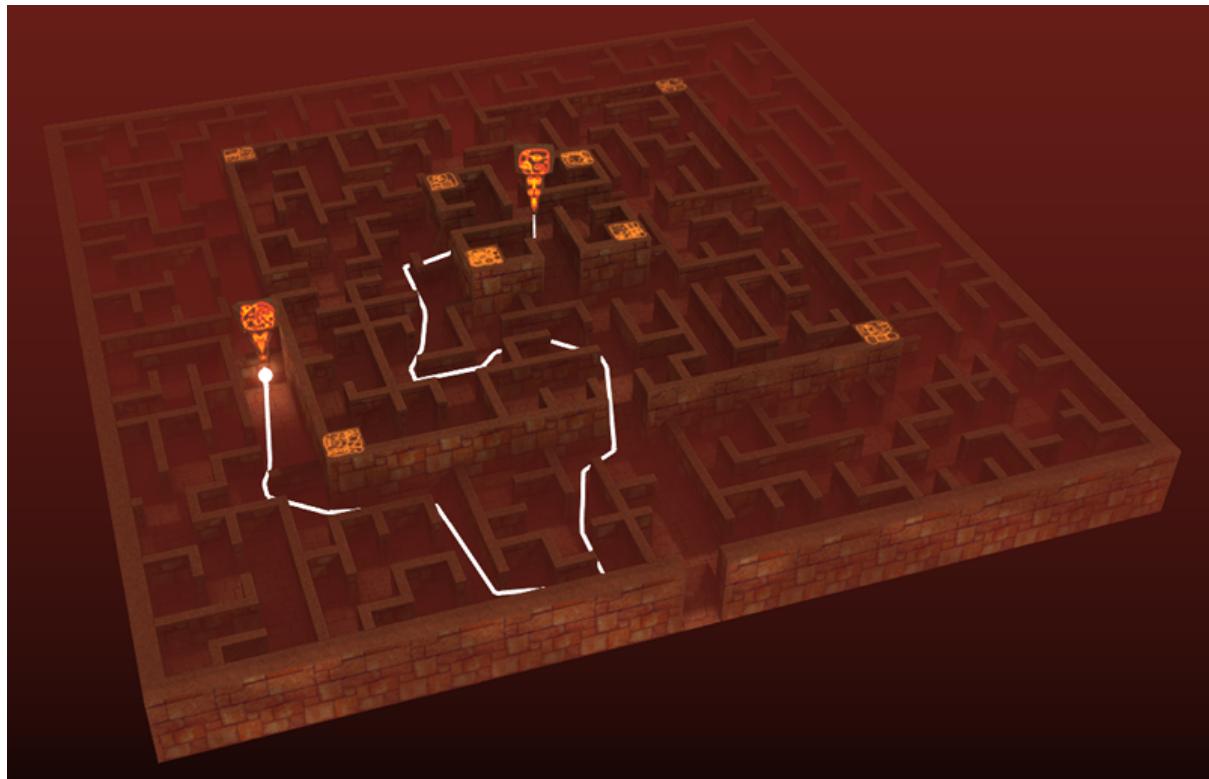
`AT_PRESSED` - однократное действие, например, нажатие кнопки.

`AT_RELEASED` - событие, возникающее, когда пользователь отпускает кнопку.

- второй параметр - массив типов сенсоров (клавиши, кнопки геймпада и мыши).
- `action_cb` - callback-функция, которая вызывается каждый раз, когда пользователь совершает действие, определённое первым параметром.

## Навигационные меши

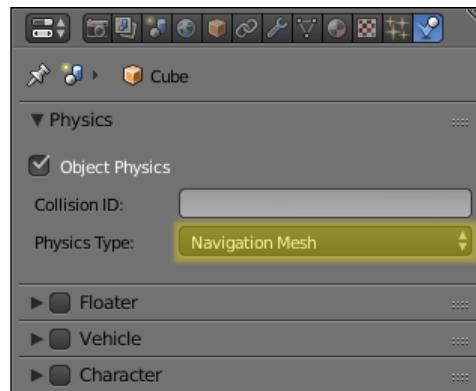
Навигационные меши (*навмеши*) - трёхмерные объекты, используемые для упрощения построения траекторий, в частности, за счёт ликвидации дополнительных вычислений, таких как расчёт столкновений.



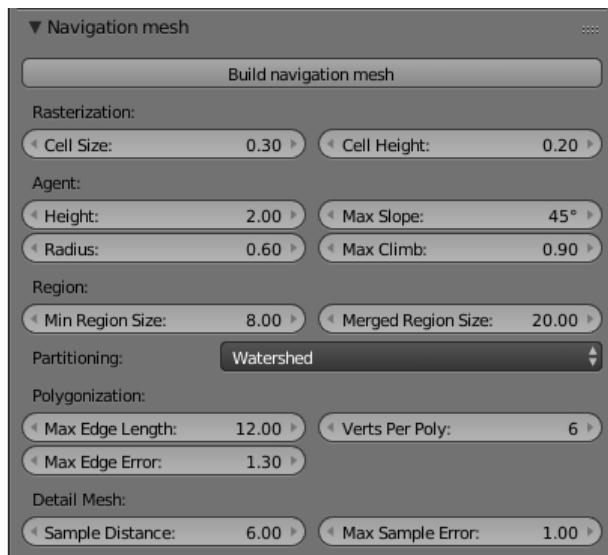
В этом примере (взятом из Code Snippets) навигационный меш используется для построения пути через лабиринт.

## Создание навигационных мешей

Чтобы использовать объект в качестве навмеша, активируйте параметр **Object Physics** на панели **Physics** и выберите **Navigation Mesh** из списка **Physics Type**.



Навигационные меши могут создаваться вручную или генерироваться с помощью специального инструмента.




---

**Примечание:** Этот инструмент изначально присутствует в Blender и не является частью движка Blend4Web.

---

Интерфейс создания навигационных мешей находится на соответствующей панели вкладки **Scene**, но так как он предназначен только для использования в интерактивных приложениях, панель доступна только в том случае, если выбран тип движка Blend4Web или Blender Game (не Blender Internal, Cycles или любая другая система рендеринга).

Главный инструмент, находящийся на этой панели - кнопка **Build Navigation Mesh**. Как и предполагает её название, она используется для генерации навигационных мешей, основанных на выбранном объекте. Навмеш создаётся как отдельный объект, располагающийся над оригинальным объектом, который никак не изменяется при генерации.

Помимо этой кнопки, на панели **Navigation Mesh** также находится несколько групп параметров для тонкой настройки создаваемых навмешей.

Группа параметров **Rasterization**:

**Cell Size** Растеризованный размер клетки. Повышение значения увеличивает размер полигонов навмеша. Максимальное значение - 1.0. Значение по умолчанию - 0.3.

**Cell Height** Растеризованная высота клетки (расстояние между навмешем, и объектом, на котором он основан). Значение по умолчанию - 0.2.

Группа параметров **Agent**:

**Height** Минимальная высота, необходимая агенту для перемещения. Значение по умолчанию - 2.0.

**Radius** Радиус агента. Значение по умолчанию - 0.6.

**Max Slope** Максимальный проходимый угол наклона. Измеряется в градусах, значение по умолчанию - 45.

**Max Climb** Максимальная высота между клетками сетки, на которую может взобраться агент. Значение по умолчанию - 0.9.

Группа параметров **Region**:

**Min Region Size** Минимальный размер региона (регионы меньшего размера удаляются). Значение по умолчанию - 8.0.

**Merged Region Size** Минимальный размер региона для объединения (регионы меньшего размера объединяются). Значение по умолчанию - 20.0.

**Partitioning** Метод разбиения навигационного меша. Поддерживаемые методы:

**Monotone** — наиболее быстрый метод. Может генерировать длинные узкие полигоны.

**Layers** — более сложный, но достаточно быстрый метод, генерирующий более оптимальные сетки треугольников, чем метод Monotone.

**Watershed** — классический Recast-алгоритм, создающий наилучшие разбиения. Этот вариант выбран по умолчанию.

Группа параметров **Polygonization**:

**Max Edge Length** Максимальная длина ребра контура. Значение по умолчанию - 12.0.

**Max Edge Error** Максимальное отклонение от контуров клеток. Значение по умолчанию - 1.3.

**Verts Per Poly** Максимальное количество вершин на полигон. Значение по умолчанию - 6.

Группа параметров **Detail Mesh**:

**Sample Distance** Интервал сэмплирования детализированной полисетки. Значение по умолчанию - 6.0.

**Max Sample Error** Максимальное отклонение сэмплирования при упрощении детализированной полисетки. Значение по умолчанию - 1.0.

## Использование навигационных мешей

После создания навмеша его можно использовать для расчёта траекторий между двумя точками. Для этого используются два метода из модуля `physics:navmesh_get_island()` и `navmesh_find_path()`. Первый из них применяется для получения ссылки на текущий остров (ближайший независимый сегмент навмеша), а второй — для получения самой траектории (в виде массива координат).

С примером использования навигационного меша для построения траекторий можно ознакомиться в приложении *Code Snippets* (образец *Pathfinding*).

## 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);
```

## Разработчикам приложений

### Table of Contents

- *Разработчикам приложений*
  - *Разработка приложений*
    - \* *Структура кода приложения*
  - *Прозрачный фон*
  - *Конвертация ресурсов*
    - \* *Команды*
    - \* *Аргументы*
    - \* *Зависимости*
    - \* *Форматы данных*
  - *Компрессия GZIP*
  - *Примеры кода*
  - *Загрузка медиаресурсов приложения.*
  - *Событийная модель*
    - \* *Сенсоры*
    - \* *Пример*
  - *Файловая структура SDK*
  - *Загрузка локальных ресурсов*
  - *Профили качества изображения*
  - *Нестандартное расположение и ориентация элемента Canvas*
  - *Мобильные веб-приложения*

## Разработка приложений

Для упрощения разработки рекомендуется пользоваться *менеджером проектов*, с помощью которого можно быстро *создать* базовый каркас приложения и получить шаблонный код, достаточный для загрузки простой сцены с базовым управлением камерой.

## Структура кода приложения

Процесс инициализации и загрузки приложения состоит из нескольких последовательных стадий, что отражено в структуре кода приложения. При использовании менеджера проектов каждый вновь созданный проект типов **Copy** и **Compile** будет содержать главный JS-файл, который находится внутри SDK по следующему пути: `./projects/ИМЯ_ПРОЕКТА/ИМЯ_ПРОЕКТА.js`.

Этот файл содержит шаблонный код, оформленный в виде модуля. Он регистрируется специальной конструкцией:

```
b4w.register("my_module", function(exports, require) {

    // module code
    //...

});
```

Таким образом, код модуля является содержимым функции, принимающей параметры `exports` и `require`.

1. `require` - метод, служащий для подгрузки библиотечных модулей движка. В шаблонном примере уже загружается ряд модулей:

Наиболее важные из них это `app` и `data`. Модуль `app` служит для упрощения инициализации приложения, а `data` предоставляет методы API для загрузки данных 3D-сцены.

---

**Примечание:** При именовании для удобства используется специальный префикс `m_` (`m_app`, `m_data`, ...), чтобы указать, что соответствующая переменная является модулем.

---

2. `exports` - объект, служащий для доступа к функциям модуля извне, например, из других модулей. В данном случае внешней сделана только одна функция `init`:

```
b4w.register("my_module", function(exports, require) {

    ...

    exports.init = function() {
        m_app.init({
            canvas_container_id: "main_canvas_container",
            callback: init_cb,
            show_fps: DEBUG,
            console_verbose: DEBUG,
            autoresize: true
        });
    }
});
```

```
    ...
});
```

С неё начинается инициализации приложения, и её вызов происходит как раз вне модуля:

```
b4w.register("my_module", function(exports, require) {

    ...

    exports.init = function() {
        m_app.init({
            canvas_container_id: "main_canvas_container",
            callback: init_cb,
            show_fps: DEBUG,
            console_verbose: DEBUG,
            autoresize: true
        });
    }

    ...
});

// import the app module and start the app by calling the init method
b4w.require("my_module").init();
```

Далее срабатывает метод `app.init` - библиотечный метод, который создаст HTML-элемент Canvas и выполнит все необходимые действия для инициализации WebGL. Он может принимать множество настроек, из которых наиболее важны следующие:

- `canvas_container_id` - id HTML-элемента - контейнера, внутри которого будет создан Canvas; по умолчанию используется элемент с id `main_canvas_container`, который присутствует в главном HTML-файле приложения.
- `callback` - функция, вызываемая по завершении инициализации.

Когда инициализация приложения будет закончена, вызовется функция `init_cb`, указанная в параметре `callback`:

```
function init_cb(canvas_elem, success) {

    if (!success) {
        console.log("b4w init failure");
        return;
    }

    m_preloader.create_preloader();

    // ignore right-click on the canvas element
    canvas_elem.oncontextmenu = function(e) {
        e.preventDefault();
        e.stopPropagation();
    }
}
```

```

        return false;
    };

    load();
}

```

В неё передаются следующие параметры:

- canvas\_elem - созданный HTML-элемент Canvas, на котором будет осуществляться рендеринг 3D-контента
- success - флаг успешности инициализации; значение `false` означает невозможность работы приложения вследствие ошибок инициализации (например, если устройством не поддерживается WebGL).

---

**Примечание:** Метод `app.init` назначает инициализацию на событие `window.onload`, поэтому в функции `init_cb` уже будет доступно все DOM-дерево HTML-документа.

---

Теперь можно начинать загрузку 3D-сцены. Это происходит в функции `load`, вызываемой из `init_cb`:

```

var APP_ASSETS_PATH = m_cfg.get_assets_path("my_project");

...

function load() {
    m_data.load(APP_ASSETS_PATH + "my_project.json", load_cb, preloader_cb);
}

```

Для загрузки используется метод `data.load`, в который первым параметром подается путь к файлу 3D-сцены. Путь к JSON-файлу должен указываться относительно главного HTML-файла приложения. Проекты, созданные в менеджере проектов, имеют отдельную директорию для ресурсов, где стандартно располагаются JSON-файлы, и относительный путь к этой директории можно легко получить. В коде шаблонного приложения это делается заранее с введением глобальной переменной `APP_ASSETS_PATH`, которая затем и используется в `data.load`.

Второй параметр метода - функция `load_cb`, вызываемая после того, как завершится загрузка и подготовка к рендерингу 3D-сцены:

```

function load() {
    m_data.load(APP_ASSETS_PATH + "my_project.json", load_cb, preloader_cb);
}

function load_cb(data_id, success) {

    if (!success) {
        console.log("b4w load failure");
        return;
    }
}

```

```
m_app.enable_camera_controls();

// place your code here

}
```

Её вызов означает переход от загрузки к началу работы приложения и, соответственно, началу рендеринга сцены. Так как этот момент - самый ранний, когда будут доступны данные 3D-сцены, то это подходящее место для инициализации и подготовки всего, что связано со сценой, объектами, анимацией и т.д. Например, здесь включается стандартная модель управления камерой при помощи метода `enable_camera_controls`.

## Написание логики приложения

После инициализации и загрузки 3D-сцены приложение будет работать, реализуя заданную программистом логику: взаимодействие через устройства ввода, манипуляции с объектами сцены, управление поведением камеры и т.д.

Рассматривая процесс *загрузки приложения*, можно выделить несколько мест, пригодных для решения той или иной задачи.

1. Метод `app.init`, стартующий инициализацию, принимает параметры конфигурации движка. Поэтому можно заняться конфигурированием прямо перед его вызовом, например, на основе параметров из URL-строки:

```
b4w.register("my_module", function(exports, require) {

    ...

    exports.init = function() {
        var url_params = m_app.get_url_params();
        if (url_params && "show_fps" in url_params)
            var show_fps = true;
        else
            var show_fps = false;

        m_app.init({
            canvas_container_id: "main_canvas_container",
            callback: init_cb,
            autoresize: true
            show_fps: show_fps
        });
    }

    ...
});

b4w.require("my_module").init();
```

2. Так как инициализация происходит по событию `window.onload`, то по её завер-

шении в функции `init_cb` будет доступно всё DOM-дерево. Таким образом, в этот момент уже можно осуществлять какие-либо подготовительные действия, например, заниматься созданием/настройкой элементов интерфейса. Однако, 3D-сцена ещё не загружена, и её объекты ещё не доступны.

- По окончании загрузки 3D-сцены вызовется функция `load_cb`. В этот момент становятся доступными объекты сцены, поэтому все действия, связанные с ними, уже можно реализовывать в этой функции. Ряд конкретных примеров можно увидеть в приложении *Code Snippets*.

Для добавления логики в приложение можно пользоваться разными способами, прибегая к стандартному браузерному API и к API движка:

- Обычную обработку ввода с клавиатуры/мыши/геймпада можно реализовать стандартными обработчиками событий посредством метода `addEventListener`. В более сложных случаях можно воспользоваться API модуля `input`. Также в движке есть метод `add_click_listener`, который одновременно отслеживает как нажатие мыши, так и нажатие на сенсорном экране, поэтому будет удобен при написании приложений совместимых как с настольными компьютерами, так и с мобильными устройствами.
- Действия, которые нужно выполнять каждый кадр, растянутые во времени (например, процедурная анимация) или отложенные можно реализовать такими методами, как `set_render_callback`, `append_loop_cb`, `animate` и `set_timeout`. Также можно использовать стандартные `setTimeout` и `setInterval`.
- Для сложной логики, зависящей и от действий пользователя и от состояния 3D-сцены, можно применять *событийную модель* движка на основе системы сенсоров.

## Система модулей

Движок Blend4Web имеет модульную структуру, а именно: набор методов API движка разбит на отдельные модули, каждый из которых можно при необходимости подключить в приложение с помощью метода `require`. Собственный код при разработке приложения рекомендуется организовывать таким же образом в виде отдельных модулей.

- Регистрация модулей.

Отдельный модуль представляет собой блок кода, обернутый особой конструкцией, с помощью которой он регистрируется:

```
b4w.register("my_module1", function(exports, require) {
    // module code
    ...
});

b4w.register("my_module2", function(exports, require) {
    // module code
})
```

```
}); ...
...
```

Для регистрации используется метод `register`. Регистрация собственных модулей допустима только если их имена не пересекаются с именами имеющихся модулей API. При необходимости проверка наличия модуля с некоторым именем может быть осуществлена с помощью метода `module_check`:

```
if (b4w.module_check("my_module"))
    throw "Failed to register module: my_module";

b4w.register("my_module", function(exports, require) {

    // module code
    ...

});
```

## 2. Подгрузка модулей.

Подгружать собственные модули можно также как и библиотечные через метод `require`:

```
b4w.register("my_module1", function(exports, require) {

    var mod2 = require("my_module2")
    ...

});
```

## 3. Инициализация приложения.

Инициализация приложения на Blend4Web обычно происходит с помощью подобного вызова:

```
b4w.require("my_module").init();
```

Здесь подгружаемый модуль `my_module` и его внешняя функция `init` являются своего рода точкой входа в приложение.

---

**Примечание:** Подгрузка модуля в глобальной области видимости происходит при помощи той же самой функции `require`, доступной как метод глобального объекта `b4w`: `b4w.require("ИМЯ_МОДУЛЯ")`.

---

## 4. Использование нескольких модулей.

После создания нового проекта в *менеджере проектов* шаблонный JS-файл приложения будет содержать один единственный модуль. Однако, часто при разработке приложения может потребоваться логически разбить код на несколько модулей. В

этом случае можно создавать как несколько модулей в одном файле, так и по принципу: один модуль - один файл.

Если в приложении используется больше одного модуля, то нужно помнить, что регистрировать модули необходимо до инициализации, иначе при попытке вызвать ещё не зарегистрированный модуль движок выдаст ошибку. Для этого в случае нескольких JS-файлов скрипт, начинающий инициализацию (содержащий точку входа в приложение), должен подключаться в главный HTML-файл приложения последним.

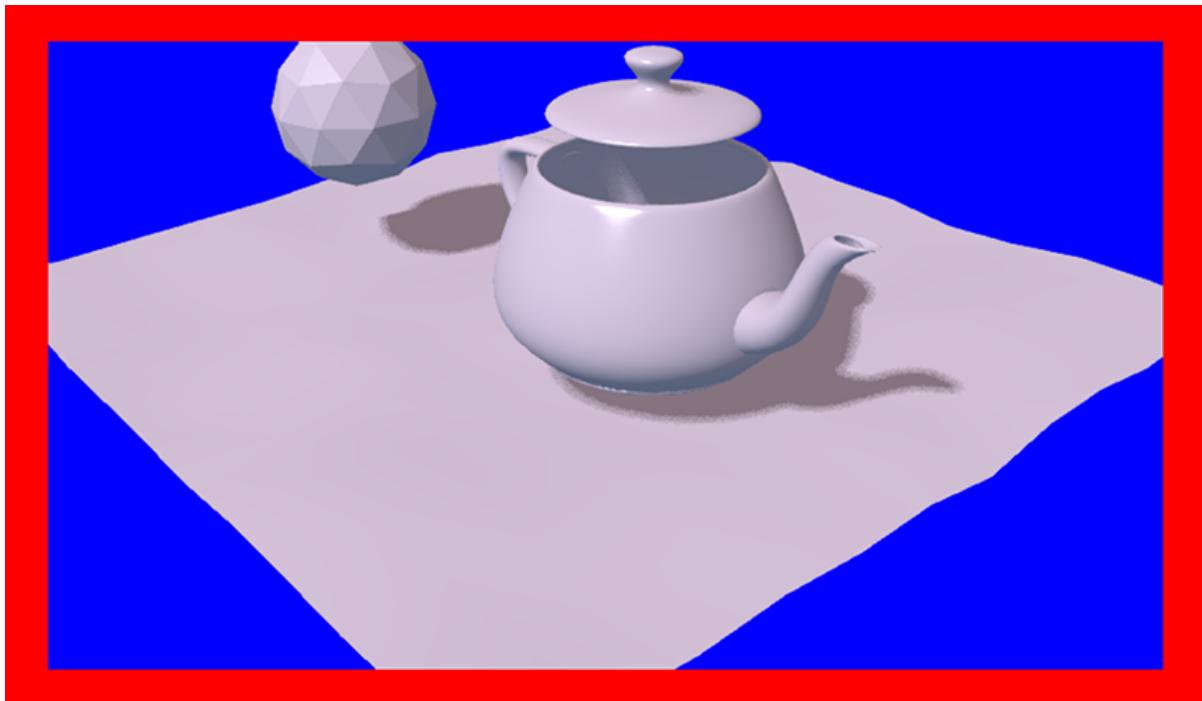
## Прозрачный фон

Параметры `background_color` и `alpha` передаются в метод `init`, который вызывается из callback-функции загрузчика (т.е., из функции, которая вызывается сразу после загрузки сцены), например, так:

```
m_app.init ({
    alpha: true,
    background_color: [0.7, 0.7, 0.7, 1]
    //this method sets the background to an opaque light gray color
});
```

Комбинация параметров, передаваемых в метод, определяет, как смешиваются цвета приложения Blend4Web и HTML-страницы. Возможны следующие варианты:

1. `alpha = false` Цвет фона определяется параметром `background_color` приложения Blend4Web, цвет фона HTML-страницы не принимается в расчёт.



2. `alpha = true` Фон HTML-страницы влияет на фон приложения Blend4Web в зависимости от его прозрачности, выражаемой четвёртой компонентой параметра `background_color` (`alpha = background_color[3]`, не путать с

параметром `alpha`, упомянутым выше).

**background\_color[3] = 1** Приводит к тому же результату, как если бы прозрачность была отключена (`alpha = false`)

**background\_color[3] = 0** Используется аддитивное смешение.



Изображение выше демонстрирует HTML-страницу, содержащую приложение Blend4Web с синим [0, 0, 1] цветом, который смешивается с красным (Red) цветом фона самой страницы, создавая более светлый оттенок.

**background\_color[3] > 0** Также используется аддитивное смешивание, но цвет `background_color` имеет большее влияние.



На этом изображении приводится та же HTML-страница с тем же приложением Blend4Web, но параметр `alpha` имеет значение 0.5, из-за чего фон приложения имеет более тёмный оттенок.

Более подробно смешивание прозрачности описано в главе “*Работа с цветом*”.

Параметр `alpha` включен по умолчанию, а в качестве `background_color` используется прозрачный чёрный цвет `[0, 0, 0, 0]`, т.е. в этом случае в качестве фона приложения используется фон HTML-страницы без какого-либо влияния фона самого приложения.

Прозрачный фон также может использоваться в *приложениях для Веб-плеера*: в них за прозрачность отвечает *URL-атрибут* `alpha`. Для использования этой возможности необходимо при создании приложения активировать параметр `Background transparency (alpha)` из группы параметров *Web Player Params*.

При использовании рендеринга неба canvas приложения Blend4Web полностью перекрывается объектами (в т.ч. небом), так что фон приложения будет полностью непрозрачным вне зависимости от настроек прозрачности.

---

**Примечание:** Рендеринг неба по умолчанию включен в новой сцене, созданной при помощи *Менеджера проектов*. Не забудьте отключить его, если вы собираетесь использовать прозрачный фон.

---

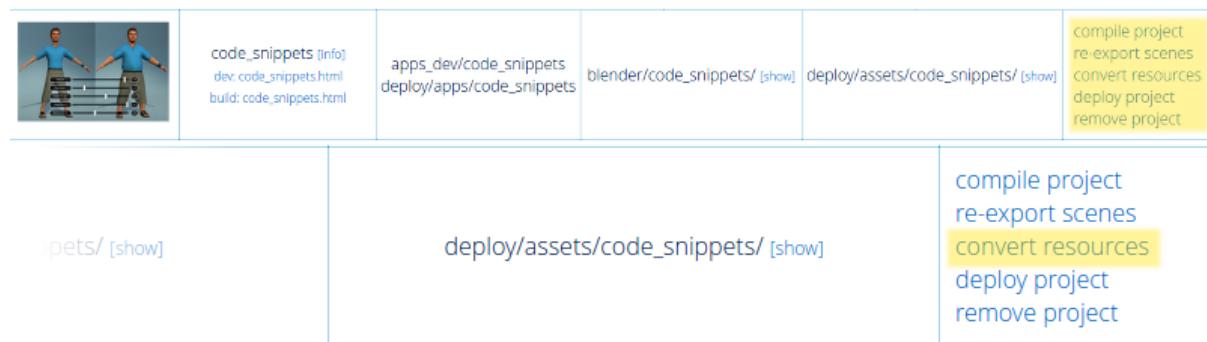
## Конвертация ресурсов

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.

Запустить скрипт можно одним из двух способов.

Во-первых, вы можете воспользоваться менеджером проектов. Кнопка `Convert Resources` находится на главной странице *менеджера проектов*, в разделе `Operations`, расположеннном в правой части экрана.



Во-вторых, вы можете запустить скрипт вручную:

```
> cd <path_to_sdk>/scripts
> python3 converter.py [options] resize_textures | convert_dds | convert_media
```

Для пользователей ОС Windows:

```
cd <path_to_sdk>\scripts
python converter.py [options] 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**.

## Команды

Команды для конвертации:

- `resize_textures` уменьшает размер текстур.
- `convert_dds` преобразует текстуры в *формат DDS*.
- `convert_pvr` преобразует текстуры в *формат PVR*.
- `convert_media` преобразует аудио- и видеофайлы в *альтернативные форматы*.
- `convert_gzip` generates GZIP-compressed versions of ".json" and ".bin" scene files and ".dds" and ".pvr" textures.

Команды для чистки:

- `cleanup_textures` удаляет текстуры низкого разрешения, создаваемые командой `resize_textures`.
- `cleanup_dds` удаляет текстуры в формате DDS, создаваемые командой `convert_dds`.
- `cleanup_pvr` удаляет текстуры в формате PVR, создаваемые командой `convert_pvr`.
- `cleanup_media` удаляет аудио- и видеофайлы в альтернативных форматах, создаваемые командой `convert_media`.
- `cleanup_gzip` удаляет файл, сжатые при помощи GZIP, создаваемые командой `convert_gzip`.

Команды для сжатия изображений:

- `compress_png` сжимает файлы PNG с целью уменьшить их размер. Для работы этой команды требуется установить и прописать в переменной окружения PATH утилиту OptiPNG.

Прочие команды:

- `check_dependencies` проверяет *зависимости конвертера*.

## Аргументы

- `-d, --dir <dir_path>` включает использование альтернативного каталога для хранения сконвертированных файлов. Путь к директории задаётся параметром `<dir_path>`.
- `-j, --jobs <jobs>` задаёт количество задач (потоков), запущенных одновременно. Если этот параметр не задан, или если его значение равно нулю, количество потоков рассчитывается автоматически на основе количества процессоров.
- `-v, --verbose` включает выдачу дополнительной информации о конвертируемых файлах. Например, при преобразовании текстур в формат DDS скрипт будет показывать процент готовности для каждого файла отдельно.

## 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 сообщение вида:

*Couldn't find PROGRAM\_NAME.*

### Linux

Список необходимых программ можно посмотреть в таблице:

| Название             | Пакет в дистрибутиве Ubuntu 16.04 |
|----------------------|-----------------------------------|
| ImageMagick          | imagemagick                       |
| NVIDIA Texture Tools | libnvtt-bin                       |
| Libav                | libav-tools                       |
| FFmpeg               | ffmpeg                            |
| PVRTC                | устанавливается вручную           |

**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.

## macOS

Пользователи macOS 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 консоли:

```
> 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
```

## Форматы данных

Преобразование происходит по схеме:

**для аудио (convert\_media):**

- ogg (ogg, oga) -> mp4
- mp3 -> oga
- mp4 (m4v, m4a) -> oga
- webm -> m4a

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):**

- ogg (ogg, oga) -> m4v / seq
- mp3 -> webm / seq

- mp4 (m4v, m4a) -> webm / seq
- webm -> m4v / 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/pvr
- jpg -> dds/pvr
- bmp -> dds/pvr
- gif -> dds

Пример файла на входе: `file_name.jpg`, пример файла на выходе: `file_name.jpg.dds`.

В целях оптимизации работы приложения существует возможность использования текстур `min50` (уменьшенных вдвое) и текстур в форматах DDS и PVRTC. Для этого при инициализации приложения необходимо передать следующие параметры:

```
exports.init = function() {
    m_app.init({
        // ...
        assets_dds_available: true,
        assets_min50_available: true,
        // ...
    });
    // ...
}
```

---

**Примечание:** Для использования текстур, сжатых в формате *PVRTC*, необходимо заменить эту строку кода

```
assets_dds_available: true,
```

следующей:

```
assets_pvr_available: true,
```

---

В этом случае движок будет загружать текстуры в формате PVRTC, если они есть в папке `./assets/`.

## Формат компрессии текстур DDS

Текстуры в формате DDS занимают меньше места в памяти (в 4 раза меньше для изображения в формате RGBA и в 6 раз меньше для изображения в формате RGB), но

при их использовании следует иметь в виду, что:

- Текстуры DDS могут не работать на некоторых устройствах, особенно мобильных, т.к. не все такие устройства поддерживают расширение `WEBGL_compressed_texture_s3tc`;
- поскольку сжатие в формате DDS осуществляется с потерями, могут быть заметны артефакты сжатия, особенно на картах *нормалей* и *смешивания*; для таких текстур рекомендуется *отключать сжатие*.



Артефакты сжатия формата DDS особенно хорошо заметны на границах тени.

При экспорте сцены из Blender в формат JSON (но не в формат HTML), DDS-текстуры будут подключены автоматически, если они имеются в наличии.

Текстуры могут быть конвертированы в формат DDS с помощью *менеджера проектов* или с помощью скрипта `scripts/converter.py`, который был описан выше.

## Формат компрессии текстур PVRTC

PVRTC - другой формат компрессии текстур, преимущественно используемый на устройствах iOS. В некоторых случаях он позволяет создавать файлы текстур размером вдвое меньше, чем при использовании формата DDS.

Движок поддерживает два метода сжатия: 2-bpp (два бита на пиксель) и 4-bpp (четыре бита на пиксель).

Как и в случае с форматом DDS, текстуры, сжатые с использованием алгоритма PVRTC, могут не работать на некоторых устройствах, особенно мобильных, так как для корректной работы этого формата сжатия требуется поддержка расширения WebGL `IMG_texture_compression_pvrtc`.

Библиотеки и SDK PVRTC доступны в версиях для операционных систем Windows, Linux и macOS. Дистрибутивы можно загрузить с веб-сайта [Power VR Insider](#).

Движок Blend4Web использует консольную утилиту PVRTC. Для того, чтобы использовать её, необходимо добавить путь к ней в переменную окружющей среды PATH, например, так:

```
export PATH = <InstallDir>\PVRTexTool\CLI\<PLATFORM>\
```

Здесь `<InstallDir>` - директория, в которой находится PVRTexTool, а `<PLATFORM>` - папка, содержащая версию утилиты, подходящую для вашей операционной системы, например, `\Windows_x86_64\` для 64-битной версии Windows.

---

**Примечание:** В Windows переменные окружающей среды можно установить из диалогового окна Система (для Windows 8 и 10) или Свойства (для Windows 7 и Vista), выбрав Дополнительные параметры системы -> Переменные среды, или с помощью консольных команд:

```
SET PATH = <InstallDir>\PVRTexTool\CLI\<PLATFORM>\
```

---

После этого текстуры можно будет преобразовать в формат PVR с помощью команды `convert_dds` скрипта `converter.py`.

## Формат видео SEQ

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.

## Компрессия GZIP

Типичное приложение на Blend4Web может использовать ресурсы разных форматов, начиная от стандартных HTML, JS, CSS, PNG, JPEG и заканчивая специфическими для движка json- и bin-файлами, содержащими данные сцены, а также сконвертированными изображениями в форматах DDS/PVR. Как в случае больших, так и малых по размеру приложений может встать вопрос уменьшения объёма ресурсов скачиваемых браузером, и, соответственно, уменьшения времени ожидания пользователем.

Обычно на сервере, содержащем веб-приложение, настраивается кеширование, что во многом решает эту проблему. Также на сервере может быть настроена GZIP компрессия для определённых форматов файлов, позволяя передавать их в запакованном виде, экономя сетевой трафик.

Если говорить о специфических для движка форматах, то GZIP компрессию имеет смысл применять для JSON, BIN, DDS и PVR. JSON и BIN - могут содержать большие объемы информации как главные файлы сцены. DDS и PVR тоже могут быть довольно велики (особенно по сравнению со стандартными PNG/JPEG), а так как это файлы текстур, то в приложении их может использоваться довольно много. Также не менее важно то, что файлы данных форматов хорошо подвергаются сжатию. Лучшим вариантом решения будет настройка использования GZIP для этих форматов на сервере.

Однако, если по каким-либо причинам это сделать невозможно, то есть вариант включить это на уровне конкретного приложения.

Движок поддерживает загрузку запакованных ресурсов с расширением .gz. В проекте типа WebPlayer JSON для этого нужно передать URL-параметр `compressed_gzip`. При разработке же собственного приложения необходимо будет передать конфигурационный параметр `assets_gzip_available` при инициализации приложения:

```
var m_app = require("app");

m_app.init({
    canvas_container_id: "main_canvas_container",
    callback: init_cb,
    show_fps: DEBUG,
    console_verbose: DEBUG,
    autoresize: true,
    assets_gzip_available: true
});
```

При этом будет требоваться наличие сжатых .gz файлов рядом с оригинальными. Например:

```
my_project/
  assets/
    my_scene.json
    my_scene.json.gz
    my_scene.bin
    my_scene.bin.gz
```

То же самое касается файлов .dds и .pvr и их сжатых вариантов .dds.gz и .pvr.gz.

---

**Примечание:** При отсутствии сжатого .gz файла движок загрузит оригинальный с соответствующим сообщением в консоли браузера.

---

Для того, чтобы сгенерировать все необходимые файлы, запакованные при помощи GZIP, достаточно выполнить `команду convert resources` из интерфейса менеджера проектов. Также это можно сделать и из консоли, запустив скрипт `./scripts/converter.py` с командами `compress_gzip` - для генерации и `cleanup_gzip` - для удаления.

## Примеры кода

В составе SDK присутствует приложение Code Snippets, демонстрирующее примеры использования функционала движка.

На данный момент приложение включает в себя следующие примеры:

- Bone API - пример управления положением отдельных костей скелета
- Camera Animation - создание процедурной анимации камеры
- Camera Move Styles - переключение режимов управления камерой
- Canvas Texture - пример работы с canvas-текстурой
- Change Image - пример изменения текстуры на лету
- Custom Anchors - процедурное создание аннотаций
- Dynamic Geometry - процедурное изменение геометрии
- Gamepad - пример управления персонажем с помощью геймпада
- Gyro (Mobile Only) - пример работы с гироскопом мобильных устройств
- Instancing - копирование объектов сцены
- Lines - рендеринг процедурных линий
- Material API - изменение свойств материалов и замена материалов объекта
- Morphing - использование ключей деформации объекта
- Multitouch (только для мобильных устройств) - использование мультитач-сенсора
- Pathfinding - расчёт пути с использованием навигационного меша
- Ray Test - использование функционала испускания лучей для определения преград
- VR - пример приложения виртуальной реальности
- Webcam - пример использования потока медиаданных с веб-камеры

Приложение Code Snippets доступно по пути `./apps_dev/code_snippets/code_snippets_dev.html`.  
Также оно доступно по ссылке из файла `index.html` в корне SDK.

## Загрузка медиаресурсов приложения.

Для упрощения обслуживания проектов и развертывания сервера храните медиаданные приложений (сцены, текстуры, звуки и т.п.) отдельно от прочих файлов проекта (JavaScript, CSS, HTML и пр.). Папка медиаресурсов располагается по адресу `projects/my_project/assets` в каталоге SDK.

Для загрузки файлов из директории (например, с помощью метода `load()`) используйте метод `get_assets_path()`:

```
m_data.load(m_config.get_assets_path("my_project") + "loaded_scene.json", load_cb);
```

Такой подход гарантирует, что приложение (как разрабатываемое, так и собранное или развёрнутое) сможет загружать медиаресурсы вне зависимости от того, на какой стадии разработки оно находится.

## Событийная модель

Событийная модель предоставляет унифицированный интерфейс для описания изменения состояний 3D сцены, упрощая обработку событий физики и действий пользователя.

### Сенсоры

Основным блоком событийной модели является сенсор (`sensor`). Сенсор является программной сущностью, которая выдаёт на выходе единственное числовое значение, в большинстве случаев это либо 1 (единица), либо 0 (ноль). Некоторые сенсоры также несут полезную нагрузку (`payload`), которую можно получить в функции-обработчике множества с помощью соответствующего API. Например, сенсор трасировки лучей (Ray Sensor) предоставляет относительную длину луча пересечения.

Опрос значений сенсоров не доступен пользователю в виде 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 `create_sensor_manifold` в документации по API):

- Объект-носитель множества (например, бросаемый объект).
- Уникальный идентификатор множества (например, “IMPACT”).
- Тип вызова функции-обработчика (варианты: `CT_POSITIVE` - положительный результат логической функции, `CT_CONTINUOUS` - каждый кадр при положительном результате логической функции и один раз при нулевом, `CT_LEVEL` - любое изменение значения результата логической функции, `CT_SHOT` - одномоментный скачок результата логической функции, `CT_TRIGGER` - переключение результата логической функции, `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.

Более подробно об API, используемых в событийной модели движка, описано в документации модуля [controls](#).

## 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'e 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).

## Файловая структура SDK

### addons

**blend4web** аддон Blender

**apps\_dev** исходный код приложений SDK

**code\_snippets** исходные файлы приложения Code Snippets

**scripts** исходные файлы самих примеров использования API Blend4Web

**dairy\_plant** исходные файлы приложения “Молочный завод” (доступно только в SDK Pro)

**demos\_animation** исходные файлы анимационных примеров

**demos\_environment** исходные файлы примеров работы с окружающей средой

**demos\_interactivity** исходные файлы примеров интерактивности

**demos\_materials** исходные файлы примеров материалов

**demos\_media** исходные файлы примеров работы с медиаресурсами

**demos\_particles** исходные файлы примеров работы с частицами

**demos\_physics** исходные файлы примеров работы с физикой

**demos\_postprocessing** исходные файлы примеров работы со спецэффектами

**farm** исходные файлы приложения “Ферма” (доступно только в SDK Pro)

**fashion** исходные файлы приложения “Показ мод” (доступно только в SDK Pro)

**flight** исходные файлы приложения “Остров”

**new\_year** исходные файлы открытки “С новым годом 2015”

**project.py** скрипт для разработчиков приложений

**space\_disaster** исходные файлы приложения Space Disaster

**tutorials** исходные файлы обучающих примеров Blend4Web

**victory\_day\_2015** исходные файлы открытки “День победы 70”

**viewer** исходные файлы приложения для просмотра сцен Viewer

**webplayer** исходные файлы веб-плеера

**website** исходные файлы приложений, размещаемых на официальном сайте Blend4Web

**blender** исходные файлы сцен в формате Blender

**csrc** исходный код бинарной части экспортёра движка и других утилит на языке C

**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** медиаданные приложения: сцены, текстуры, звуковые файлы

**doc** настоящее руководство пользователя в формате HTML, собирается автоматически из *doc\_src*

**webglreport** Приложения, выводящее информацию о WebGL

**distfiles** списки сборщика дистрибутивов

**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 и документации

**projects** каталог пользовательских проектов

**README.rst** файл README

**scripts** скрипты

**check\_resources.py** скрипт для проверки и сообщения о неиспользуемых ресурсах (изображения и звуки, на которые ссылаются экспортруемые 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

**tmp** директория для временных файлов (например, для Fast Preview)

**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

## 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.

## 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 (используется более высокое качество, скорость работы ниже).
- *пользовательское качество* (`P_CUSTOM`) - любой параметр качества может иметь любое доступное значение. Используется, если необходимо вручную задать некоторые параметры. По умолчанию используются настройки, аналогичные профилю `High`.



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
});
```

## Нестандартное расположение и ориентация элемента Canvas

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);
// ...
```

- 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 получить корректно рассчитанные координаты. Это можно сделать при помощи вышеупомянутых методов `get_coords_x()` и `get_coords_y()` с `true` в качестве третьего параметра.

```
var m_mouse = require("mouse");

// ...
var x = m_mouse.get_coords_x(event, false, true);
var y = m_mouse.get_coords_y(event, false, true);
// ...
```

Другой вариант - использовать метод `client_to_canvas_coords()`, как на следующем примере:

```
var m_cont = require("container");

var _vec2_tmp = new Float32Array(2);

// . . .
var coords_xy = m_cont.client_to_canvas_coords(event.clientX, event.clientY, _vec2_tmp);
// . . .
```

## Мобильные веб-приложения

Возможность масштабирования всей веб-страницы на мобильных устройствах также может приводить к смещению элемента Canvas. Описанные выше решения годятся и для этого случая, однако, в качестве альтернативы можно запретить масштабирование вовсе и избежать подобных проблем. Для этого достаточно в заголовке страницы добавить следующий мета-тег:

```
<meta name="viewport" content="width=device-width, initial-scale=1.0, user-scalable=no">
```

В некоторых случаях может потребоваться зафиксировать определённую ориентацию мобильного устройства (например, если вы не хотите создавать отдельные варианты интерфейса для ландшафтной и портретной ориентации). Поскольку штатный API для фиксации экрана - экспериментальный и не распространён широко, это проще сделать при помощи правил CSS.

Для сохранения ориентации в ландшафтном режиме можно установить вращение элемента `<html>` на 90 градусов в портретном режиме:

```
@media (orientation : portrait) {
    html {
        position: absolute;
        width: 100vh;
        height: 100vw;
        overflow: hidden;

        -moz-transform: rotate(90deg);
        -ms-transform: rotate(90deg);
        -webkit-transform: rotate(90deg);
        transform: rotate(90deg);
        -moz-transform-origin: top left;
        -ms-transform-origin: top left;
        -webkit-transform-origin: top left;
        transform-origin: top left;

        left: 100%;
    }
}
```

В свою очередь, правило CSS для использования только портретного режима будет выглядеть следующим образом:

```
@media (orientation : landscape) {  
    html {  
        position: absolute;  
        width: 100vh;  
        height: 100vw;  
        overflow: hidden;  
  
        -moz-transform: rotate(-90deg);  
        -ms-transform: rotate(-90deg);  
        -webkit-transform: rotate(-90deg);  
        transform: rotate(-90deg);  
        -moz-transform-origin: top left;  
        -ms-transform-origin: top left;  
        -webkit-transform-origin: top left;  
        transform-origin: top left;  
  
        top: 100%;  
    }  
}
```

## Разработчикам движка

### Содержание

- *Разработчикам движка*
  - Стиль оформления кода
    - \* Примеры
  - Сборка движка
  - Сборка аддона
  - Зависимости
  - Названия функций и переменных
  - Отладка
  - Компиляция шейдеров
    - \* Валидация
    - \* Обfuscация
    - \* Оптимизация
    - \* Директивы *import/export*
    - \* Рекомендации и ограничения
    - \* Расширения WebGL
    - \* Ошибки компилятора
  - Обновление переводов аддона

### 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).

## 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.

## 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
```

## Сборка аддона

Бинарные сборки аддона Blend4Web подготовлены для следующих платформ: Linux x32/64, macOS 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.

## Зависимости

В таблице ниже перечислены все зависимости, в порядке убывания важности для разработки.

| Название             | Ubuntu 16.04 package   | 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    |
| ImageMagick          | imagemagick  | конвертация текстур                       |
| NVIDIA Texture Tools | libnvt-bin   | конвертация текстур                       |
| Libav                | libav-tools  | конвертация медиаресурсов                 |
| NVIDIA Cg Toolkit    | nvidia-cg-toolkit  | отладка шейдеров                          |
| OptiPNG              | optipng  | оптимизация PNG-файлов                    |
| Emscripten           | из исходных текстов EMSDK  | сборка Uranium                            |
| Gnuplot              | gnuplot  | отладка                                   |
| Graphviz             | graphviz   | отладка                                   |
| xsel                 | xsel   | отладка                                   |
| Sphinx               | python3-sphinx   | сборка документации                       |
| sphinx-intl          | устанавливается через PIP v3 (pip3 install sphinx-intl)                                    | сборка документации (перевод)             |
| TeX Live             | texlive texlive-latex-extra<br>texlive-lang-cyrillic texlive-lang-chinese<br>texlive-xetex | сборка документации (PDF-версия)          |
| JSDoc 3              | устанавливается через NPM (npm install -g jsdoc)   | сборка документации (документация на API) |
| PEG.js               | из исходных текстов PEG.js   | сборка препроцессора шейдеров             |

## 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)

*switch\_* switch flag's binary value

*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

## 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.

## 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

## 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 ”,”.

## 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).

## 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.

---

## 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 grot

#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 объявление, необходимые для импорта, а после него может использовать экспортные 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.

## 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 обфускации 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`. Так же 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.

## Расширения WebGL

Работа обфускатора может зависеть от используемых WebGL-расширений, если они каким-либо образом влияют на шейдерный язык. На данный момент поддерживаются следующие расширения:

- `OES_standard_derivatives`

## 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 обfuscация include-файла FILE_NAME.   |
| Error! Extension NAME is unsupported in obfuscator. File: ‘FILE_NAME’.                                       | Ошибка! WebGL-расширение с именем NAME, использованное в файле FILE_NAME, не поддерживается обfuscатором.   |
| 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.<br>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.<br>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.<br>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.   |

## 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

### Содержание

- Работа в команде с использованием Git
  - Общие сведения
  - Типичный рабочий процесс
  - Индивидуальные настройки
  - Проверка статуса
  - Перед коммитом
    - \* Проверка изменений (текстовых файлов)
    - \* Восстановление файлов
    - \* Посторонние файлы
  - Подготовка к коммиту
    - \* Добавление файлов
    - \* Удаление файлов
  - Коммит
  - Синхронизация между репозиториями
    - \* Из удаленного - в локальный
    - \* Из локального - в удаленный
  - Разрешение конфликтов
    - \* Общие сведения
    - \* Порядок действий
    - \* Бинарные файлы
    - \* Текстовые файлы
    - \* Корректирующий коммит
  - Тэги
  - Другие полезные команды

### 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`.

## 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.
5. 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
```

## 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.

## 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")
```

## Перед коммитом

### 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);
+
    }
}
```

## 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.

## 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.

## Подготовка к коммиту

### 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`.

## 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`.

## 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**.

## Синхронизация между репозиториями

### 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 к 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.

## 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 gfxteam@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 gfxteam@lixer:blend4web.git
 ! [rejected]      master -> master (non-fast-forward)
error: failed to push some refs to 'gfxteam@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`.

## Разрешение конфликтов

### 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.

## 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.

## 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.

## 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 экспортанные текстовые файлы сцен (заканчивающиеся на `.json`) проще повторно экспортовать.

## 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`.

## 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
```

Синхронизовать тэги с удаленным репозиторием:

```
> git push --tags
```

Удалить тэг (при ошибочном создании):

```
> git tag -d R130603
```

## 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

---

## Содержание

- *Проблемы и решения*
  - Поддержка WebGL
  - Проблемы при запуске движка
  - Ошибка инициализации WebGL
  - Если ничего не помогло...
  - Известные проблемы

Поскольку на сегодняшний день WebGL - всё ещё достаточно новая технология, она может работать некорректно с некоторыми комбинациями программного и аппаратного обеспечения. В этой главе перечислены наиболее распространённые проблемы, с которыми может столкнуться пользователь движка Blend4Web, и приведены способы их исправления.

## Поддержка WebGL

Если вы используете настольный компьютер или ноутбук, вам требуется видеокарта с поддержкой DirectX 9.0c и OpenGL 2.1, такая, как:

- Nvidia GeForce серии 6xxx или выше.
- AMD/ATi Radeon серии R500 (X1xxx) или выше.
- Intel GMA 950 или выше.

Если вы используете WebGL на мобильном устройстве, проверьте, присутствует ли ваше устройство в [списке совместимости](#).

Вам также потребуется веб-браузер с поддержкой технологии WebGL.

WebGL поддерживают следующие веб-браузеры:

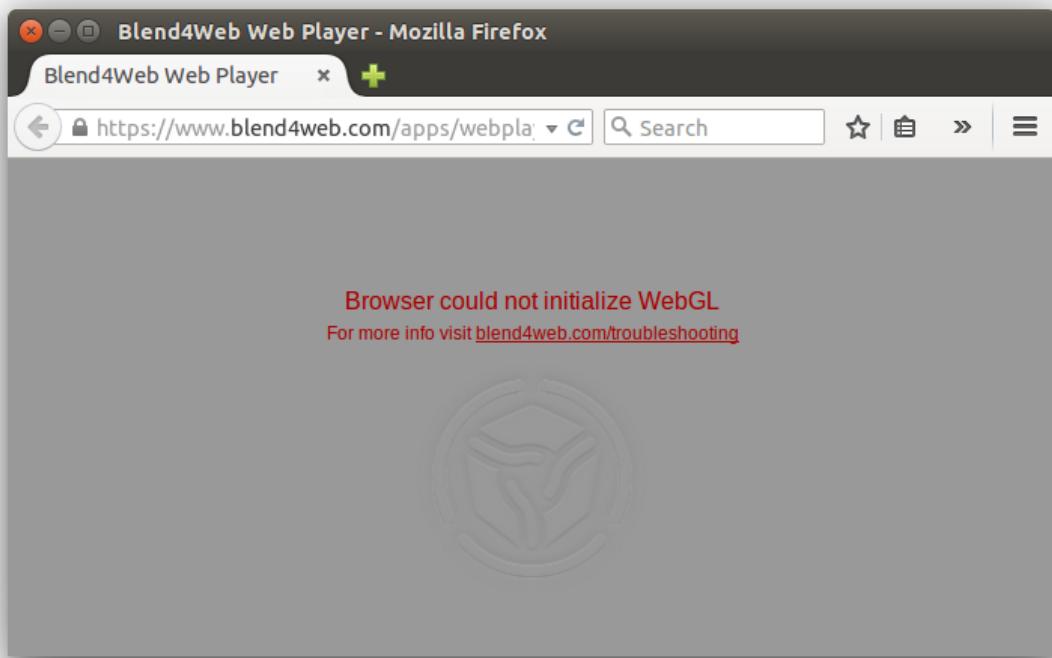
- Google Chrome (начиная с версии 9)
- Mozilla Firefox (начиная с версии 4.0)

- Safari (начиная с версии 8.0)
- Chrome for Android (начиная с версии 25)
- Internet Explorer (начиная с версии 11)
- Microsoft Edge
- Opera (начиная с версии 12)
- UC Browser (экспериментальная поддержка)
- Yandex Browser

Мы также рекомендуем использовать новейшие версии веб-браузеров для предотвращения проблем с совместимостью.

## 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:

- Файлы ресурсов, которые пытается загрузить движок, были перемещены или удалены.

- Используются старые версии драйверов.
- Используются открытые драйвера, не обеспечивающие поддержку WebGL.

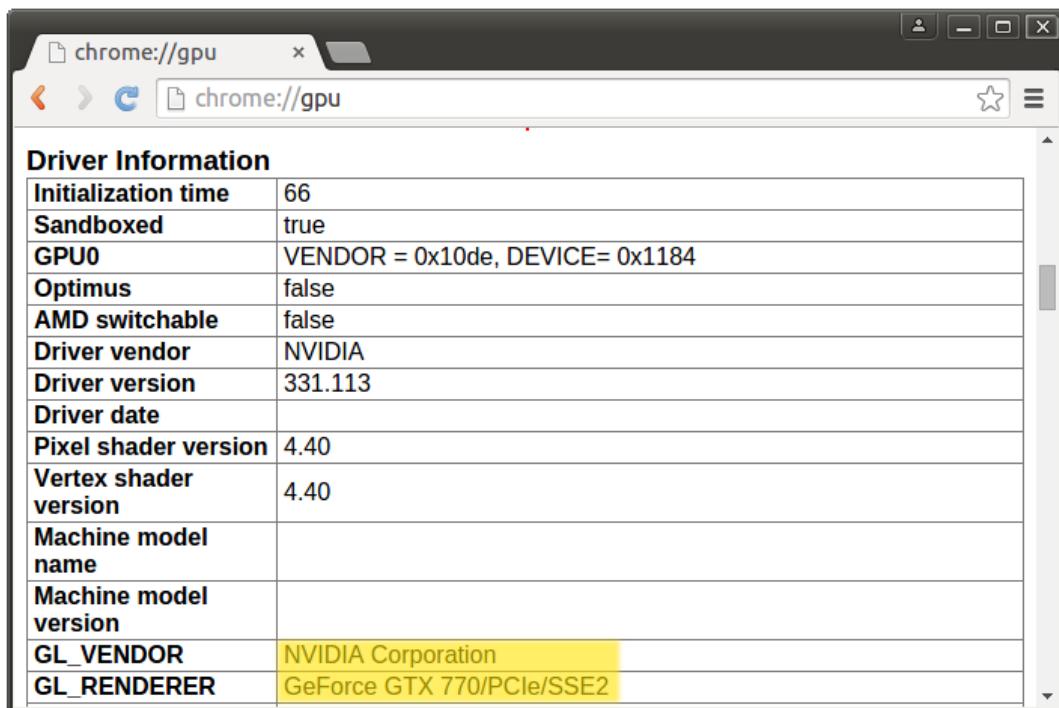
Для пользователей Linux - ввиду неполной реализации OpenGL стека в драйверах с открытым кодом в настоящий момент рекомендуется использовать проприетарные драйверы текущей версии для графических процессоров Nvidia и AMD.

- Используется устаревшая операционная система, такая как Windows XP.
- Браузер не настроен для работы с локальными ресурсами. В этом случае проблема может быть решена использованием локального веб-сервера. См. раздел [Загрузка локальных ресурсов](#).

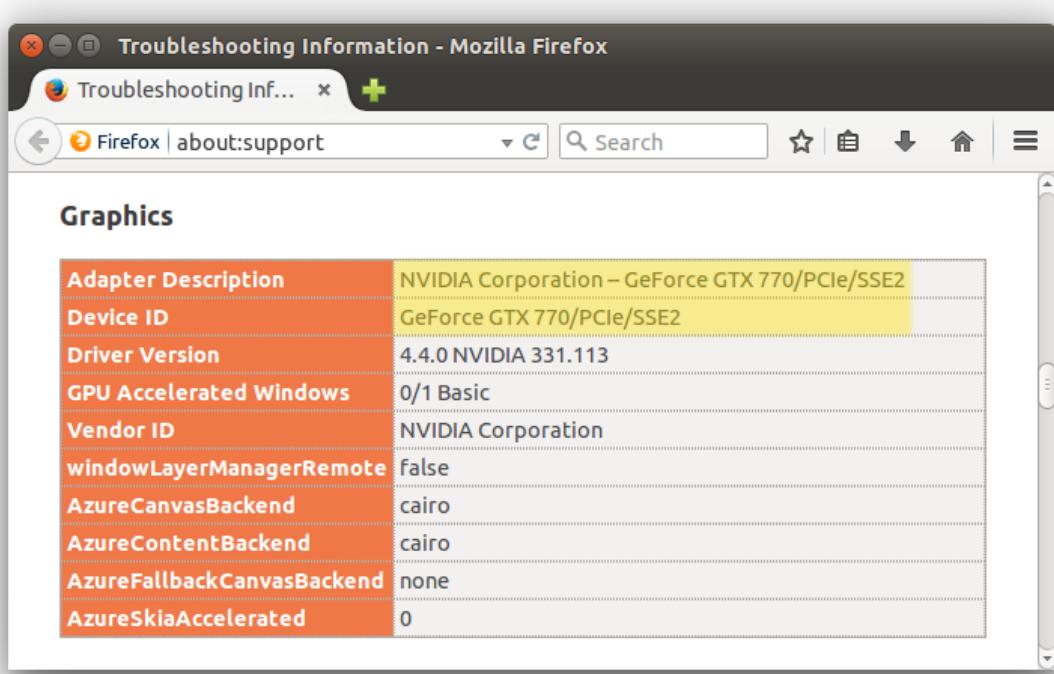
## Ошибка инициализации WebGL

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...



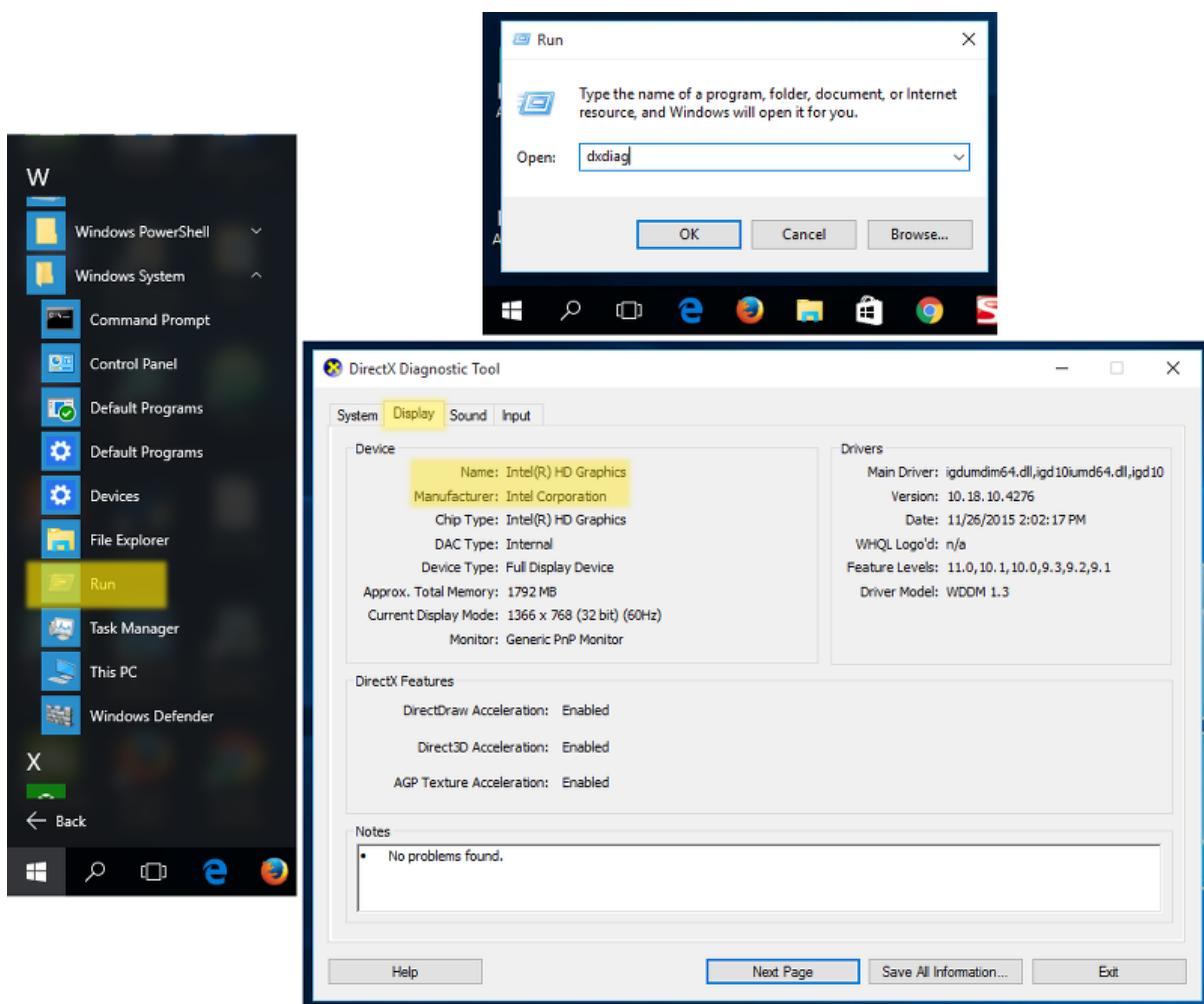
или Firefox...



Для операционных систем семейства Windows можно воспользоваться средством диагностики DirectX **dxdiag**.

Для этого выполните следующие действия:

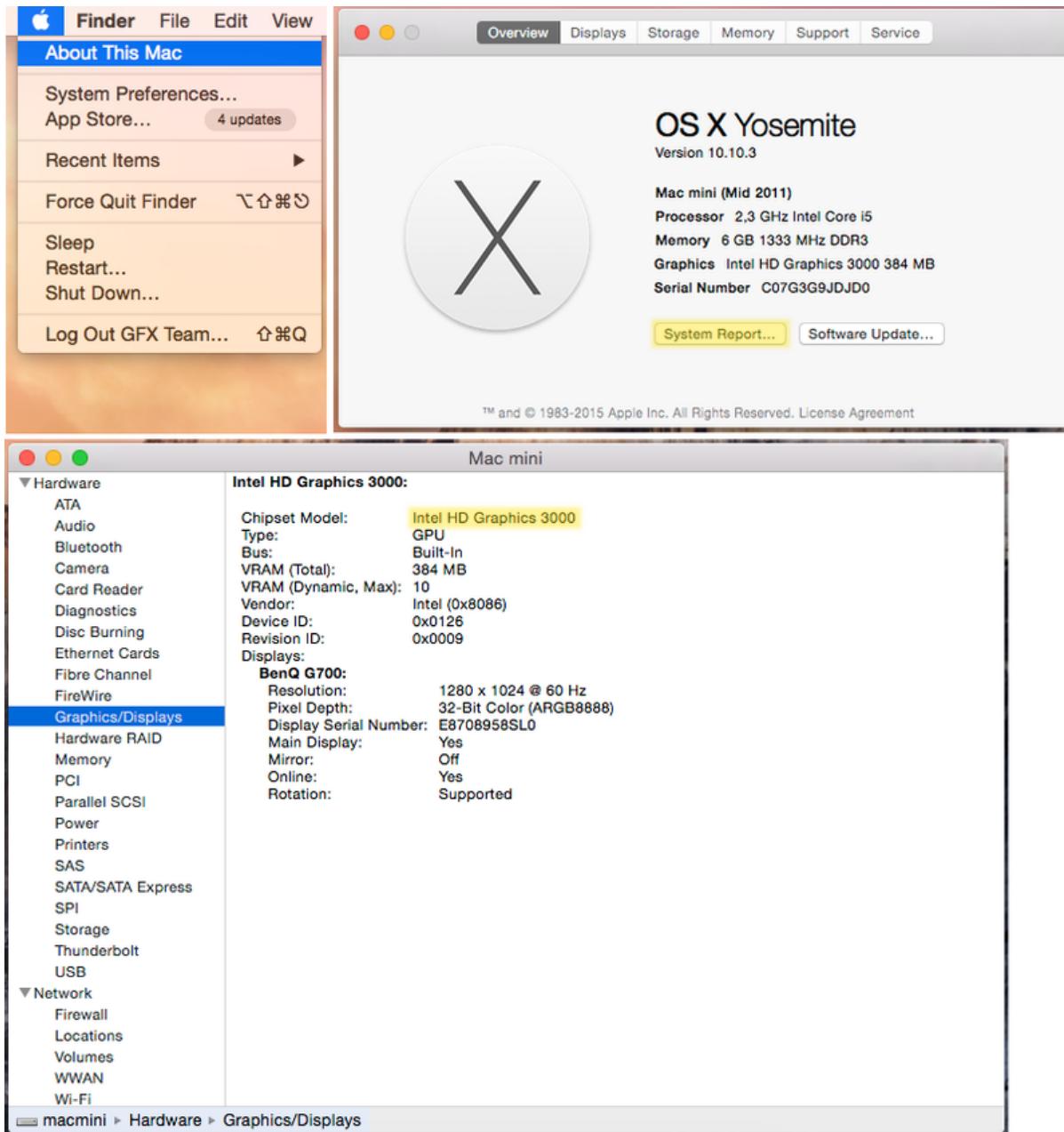
1. Выберете команду Run из стартового меню
2. В окне Open введите *dxdiag* и нажмите Enter чтобы вызвать утилиту диагностики DirectX
3. Откройте панель Display. Там вы можете найти информацию о производителе, модели и других параметрах вашей видеокарты.



В macOS вам следует ознакомиться с System Report.

Для этого выполните следующие действия:

1. Выберете **About This Mac** из меню Apple.
2. Нажмите кнопку **System Report**.
3. Выберете пункт **Graphics/Displays** из раздела **Hardware**.

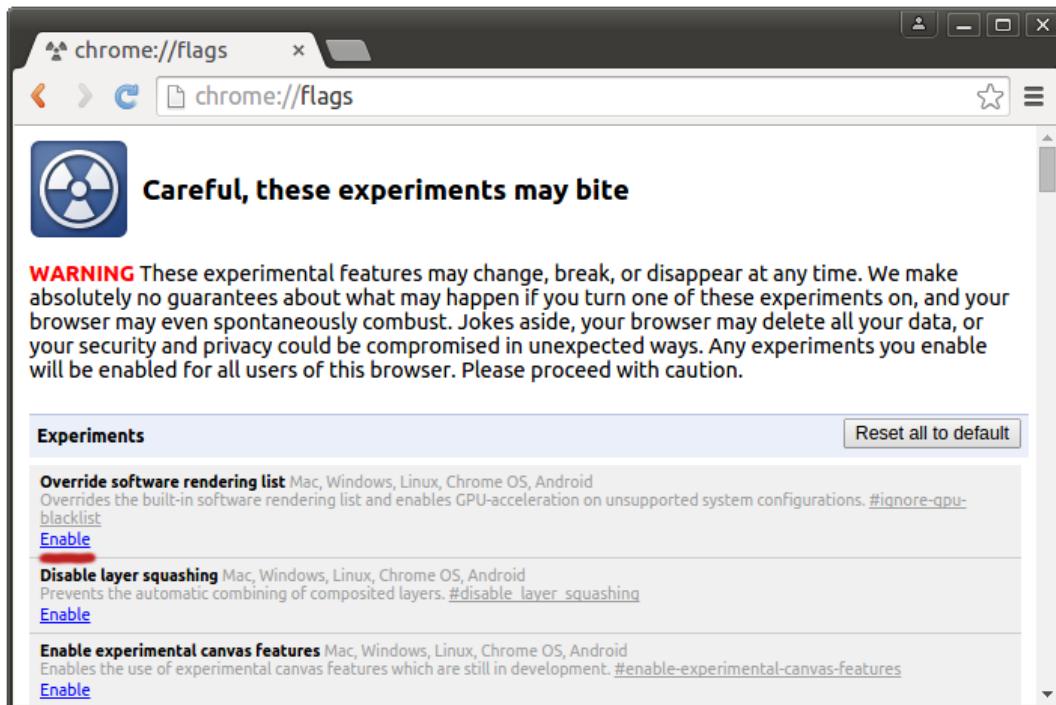


Необходимо загрузить драйверы с соответствующего центра поддержки (например, Intel, Nvidia, AMD/ATI). После установки драйверов перезагрузить систему.

3. Если в результате вышеперечисленных действий инициализировать рендеринг не удается (или нет возможности обновить систему), можно попробовать изменить настройки браузера.

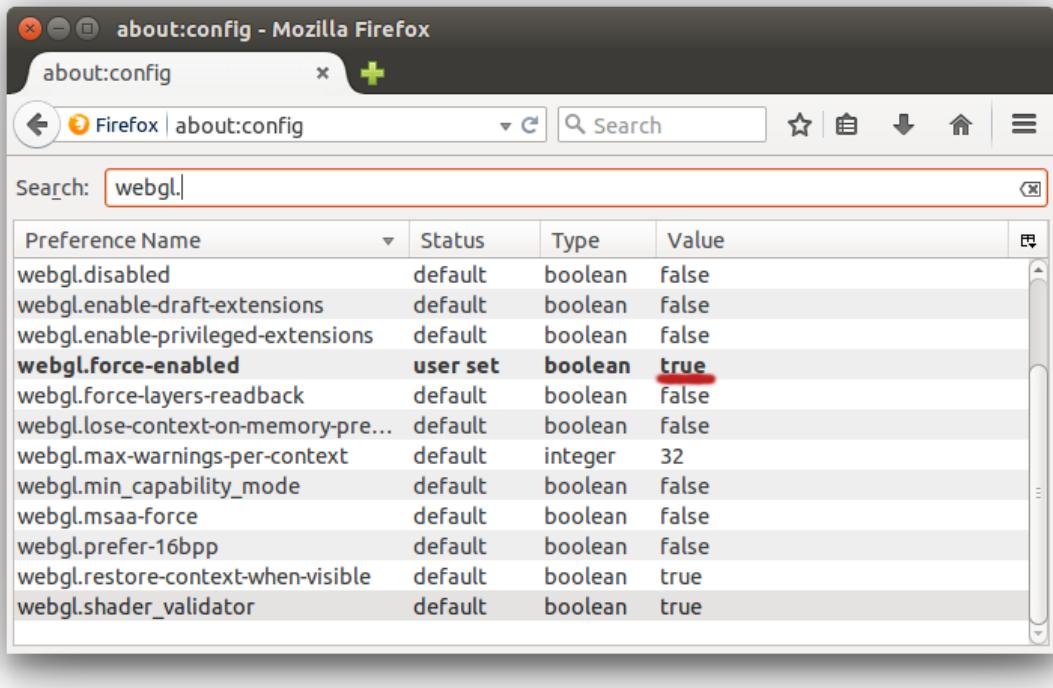
*B Chrome:*

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.



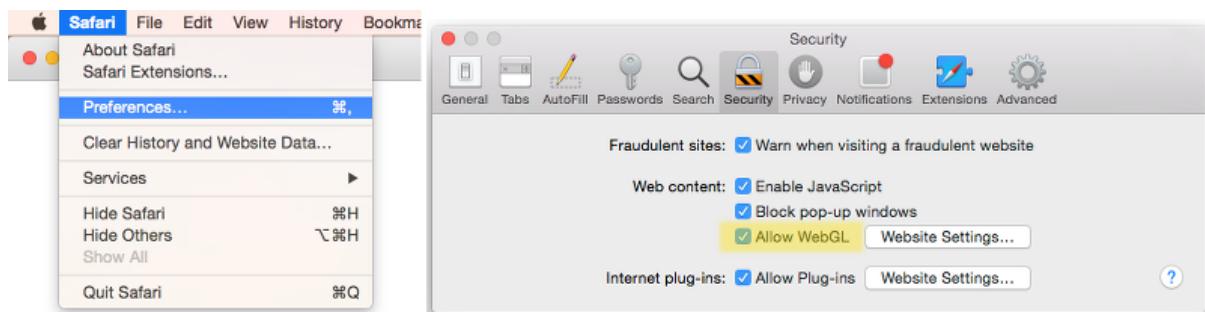
*B Firefox:*

Ввести **about:config** в адресную строку браузера, найти параметр `webgl.force-enabled` и переключить его двойным щелчком мыши из `false` в `true`.



### Для Safari

Выберите пункт **Preferences** из меню Safari, затем выберите вкладку **Security** и убедитесь, что опция **Allow WebGL** включена.



## Если ничего не помогло...

Если ни один из перечисленных выше способов не помог вам устранить проблемы, посетите [форум сообщества Blend4Web](#) и оставьте сообщение в разделе “Ошибки”. Наша команда сделает всё возможное, чтобы помочь вам.

## Известные проблемы

- Проблемы с обновлением старых версий аддона.

Рекомендуется перезагрузка Blender после обновления SDK или 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.
- На системах Apple macOS 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>. В Blender 2.76b эта проблема устранена, поэтому рекомендуем обновиться.
- При использовании драйверов Nouveau скелетная анимация может работать некорректно.
- В браузерах IE11 и Microsoft Edge, а также на iPad могут некорректнорендериться текстуры с прозрачностью.

Проблема актуальна для *прозрачных материалов*. Артефакты заметны на тех участках, где значение альфа-канала равно нулю или отличается от него незначительно. Для исправления рекомендуется поднимать значение альфа-канала пока артефакты не исчезнут (как правило, бывает достаточно величин в диапазоне от 0.01 до 0.05).

- Сбой работы WebGL в браузере Chromium под Linux на GPU Nvidia GeForce серий 400/500 с драйверами версии выше 355.

Проблема вызвана несовместимостью последних драйверов Nvidia с сэндбоксом Chromium. В качестве решения рекомендуется понизить версию драйверов до 340-х.

- Проблемы со звуком.

В некоторых версиях Google Chrome звук не воспроизводится для источников звука с типом “Background Music”, если используется экспорт в HTML. В качестве решения в настоящий момент тип источника звука автоматически меняется на “Background Sound”.

Ошибка аудиоконтекста проявляется, если запущено много (больше 6) копий движка Blend4Web (например, на нескольких вкладках браузера или в нескольких iframe'ах на одной странице). Ошибка предотвращена путем отключением звука в приложении, если для него не может быть создан аудиоконтекст.

На некоторых мобильных устройствах (в частности, Nexus 6 и Nexus 9) в браузере Google Chrome версий 50-52 может не работать пауза. В версиях Chrome начиная с 53 проблем быть не должно. Более подробная информация: <https://bugs.chromium.org/p/chromium/issues/detail?id=616772>.

- QQ Browser на данный момент не поддерживает WebGL.
- WebGL может работать нестабильно и давать сбои при использовании драйверов Mesa 11.x в Linux/Chrome на видеокартах Intel. Понижение версии драйверов до 10.x может решить проблему.
- При использовании тач-сенсора GearVR в VR-режиме недоступна информация о движениях пальцев.

Samsung Internet Browser не регистрирует события, связанные с движением пальцев на тач-сенсоре в режиме VR.

- Частицы, испускаемые системами частиц, находящимися на неактивных (скрытых) слоях, могут иметь нулевые координаты, т.к. неактивные слои не обновляются Blender'ом.

Проблема может быть решена с помощью активации слоёв перед экспортом или выбора необходимых слоёв вручную (в этом случае Blender их обновит).

- Объекты с типом прозрачности Alpha Blending некорректно отображаются в браузере Google Chrome на устройствах с GPU Mali серии 400.

---

## Замечания к релизам

---

### v17.04

#### Новый функционал

- Поддержка материалов окружающей среды в *Cycles*.

Теперь рендеринг неба можно настраивать с помощью нод *Cycles*.

Для управления итоговым цветом окружения применяется нода *Background*. Поддерживаются также сферические карты окружающей среды с типами проекции *Equirectangular* и *Mirror Ball*. Материалы окружающей среды могут быть анимированы при помощи нод *Value* и *RGB* (поддерживается как обычная анимация, так и NLA). Опция *Use Nodes* продублирована на панели настроек *World*.

- Экспериментальная поддержка WebAssembly в физическом движке.

Теперь вы можете использовать оптимизированную версию физического движка *Uranium.js* (нашего порта движка *Bullet* на *JavaScript*). Эта возможность включается при помощи нового конфигурационного параметра `physics_use_wasm`. Возможность считается экспериментальной и по умолчанию отключена.

- Экспериментальная поддержка VR-контроллеров

Добавлен метод `enable_controllers`. Обновлен при мер VR из приложения *Code Snippets*. Также добавлены методы `create_gamepad_orientation_sensor` и `create_gamepad_position_sensor`.

- Добавлена поддержка материалов с типом прозрачности *Alpha Anti-Aliasing*.

На совместимом оборудовании этот вариант техники *Alpha Clipping* выглядит намного лучше стандартного. Для использования этого типа прозрачности требуется поддержка MSAA, поэтому он работает только на устройствах, поддерживающих WebGL2. В других случаях вместо него будет использоваться тип прозрачности *Alpha Clip*.

- Поддержка загрузки ресурсов с компрессией GZIP.

Теперь приложение может загружать сжатые файлы ”.json.gz” и ”.bin.gz” вместо стандартных ”.json” и ”.bin”. То же относится и к текстурам в форматах DDS и PVR. Эта возможность включается в веб-плейере при помощи *URL-параметра* “compressed\_gzip”. В обычном приложении она активируется передачей флага “assets\_gzip\_available” в методе `init`. Сжатые файлы генерируются при помощи команды “convert resources” в Менеджере проектов. Эту возможность, однако, следует использовать только в тех случаях, когда сжатие не может быть настроено на сервере. Более подробная информация доступна в разделе документации “*Компрессия GZIP*”.

- Улучшения системы LOD.

Появилась возможность плавного переключения уровней LOD на основе техники alpha test. Соответствующая опция называется **LOD Smooth Transitions** и расположена на панели **Scene -> Object Clustering & LOD**. Следует иметь в виду, что включение этого параметра для всех объектов может существенно снизить скорость работы приложения.

Также появилась новая опция **Max LOD Hysteresis Interval**. Она предназначена для устранения слишком частых переключений уровней LOD в ситуации, когда камера движется рядом с границей между двумя уровнями LOD. Эта опция заменяет старый параметр **LOD Transition Ratio** на панели **Object->Levels of Detail**.

Подробное описание находится в разделе “*Уровни LOD*”.

- Поддержка RGBA-теней.

Это позволяет использовать тени на устройствах, не поддерживающих текстуры глубины.

- Улучшения эффекта Bloom.

Реализован новый алгоритм расчёта эффекта Bloom. Он работает со всеми типами источников света и не зависит от направления освещения. Также добавлена возможность расчёта адаптивной средней яркости.

- Поддержка текстур типа **Non-Colour Data** в Filmic Blender.
- Резервный вариант для сжатых текстур.

Теперь, если сервер возвращает ошибку 404 при обращении к текстурам в форматах gzip/dds/pvr, движок попытается загрузить несжатые изображения.

- Отладочный флаг этапов загрузки

В метод `init` добавлен флаг `debug_loading`, позволяющий отслеживать этапы загрузки через консоль.

- Улучшения в редакторе нормалей.

Для смешивания итоговых и начальных положений нормалей в некоторых операторах появился параметр **Factor**.

Появилась возможность использовать оператор **Face** с несколькими полигонами.

Появился новый оператор **Scale**, предназначенный для масштабирования нормалей по осям координат.

Функциональность ряда операторов переработана, а их названия изменены в соответствии с новыми возможностями: **Tree -> Cursor 3D**, **Foliage -> Axis**.

- Добавлен API для визуализации нормалей динамических объектов.

В модуль `debug` добавлены методы `show_normals` и `hide_normals`.

## Изменения

- Удалены предупреждения для объектов Force Field типа **None**.
- Теперь метод `get_sky_params` возвращает значение `null` для непроцедурного неба.
- Теперь параметр **Use Custom Color** на панели **Mist** по умолчанию отключён.
- Теперь метод `append_object` работает также и с недуплицированными объектами.
- Верхняя граница диапазона значений параметра SSAO **Influence** увеличена до 2.0.
- Размер canvas'а больше не меняется на 320x240 в случаях, когда одна или обе его стороны имеют нулевые размеры.
- Теперь размеры кубической карты неба изменяются автоматически при изменении текстуры.

## Исправления

- Исправлен русский перевод пунктов меню “Справка” в аддоне.
- Исправлен ряд неточностей в документации API.
- Исправлен параметр **Do Not Export** для объектов с уровнями LOD (ранее он не работал).
- Исправлено снижение FPS в Просмотрщике сцен после выбора объекта.
- Исправлено переключение уровней LOD для динамических объектов (ранее движок мог визуализировать более одного уровня LOD одновременно).

- Теперь одинаковые запакованные изображения и звуковые файлы при экспорте распаковываются в один файл, а не в несколько одинаковых файлов.
- Исправлено ограничения параметра strength ноды Normal Map. Теперь значения ниже 0 не влияют на нормали.
- Удалены ненужные обновления тумана при анимации окружающей среды.
- Использование объектов типа Empty с динамическими материалами не приводит к крэшу движка.
- Исправлена ошибка клонирования текстур, вызывавшая утечки памяти.
- Исправлена сортировка батчей по параметру alpha\_clip.
- Исправлен крэш текстуры Foam + ShoreMap в материале воды.
- Исправлено преобразование пространства в редакторе нормалей.
- Исправлено поведение Менеджера проектов в нестандартном Python-окружении.
- Исправлен некорректный рендеринг материалов, использующих карты нормалей.
- Исправлена фильтрация текстур для правого глаза в режиме VR.
- Исправлен крэш на IE11 при использовании client\_to\_element\_coords.
- Исправлены методы is\_hideable, show\_object и hide\_object.
- Исправлен крэш Cycles-материалов с несоединёнными выходами Surface.

## v17.02

### Новый функционал

- Улучшения в менеджере проектов.

Добавлен визуальный конфигуратор проекта. Конфигуратор позволяет редактировать только те настройки, которые доступны для данного проекта, кроме того, все изменения обратимы. Для вызова конфигуратора достаточно нажать ссылку *[config]* рядом с названием проекта на главной странице Менеджера проектов.

Появилась команда clone project.

Появились карты кода (source maps) JavaScript. Карты кода позволяют просматривать сжатый программный код в первоначальном виде, используя особую “карту”. Для генерации карты кода следует установить флаг *-b* для скрипта *project.py* при сборке проекта.

- Улучшения интерфейса.

Чтобы упростить создания Blend4Web-контента, мы переработали интерфейс Blender.

Меню *Render* и *Help* переработаны для движка Blend4Web.

Добавлены консольные сообщения о неподдерживаемых редакторах.

Панель ‘Rigid Body’ отсутствует в интерфейсе при использовании движка Blend4Web.

Номер версии Blend4Web добавлен на экран загрузки Blender.

- Дальнейшая поддержка материалов *Cycles*.

Добавлена поддержка выхода *Displacement*. Он применяется для быстрого добавления рельефа к материалу.

Добавлена поддержка ноды *Transparent BSDF*. Нода относится к категории *Shader* и применяется для создания прозрачных материалов без рефракции: свет проходит через такую поверхность, как если бы её не было.

- Расширение поддержки наследования материалов.

Метод *inherit\_material* был серьёзно усовершенствован и теперь поддерживает нодовые материалы. Для его использования необходимо включить опцию *Dynamic Geometry & Materials* (ранее известную как *Dynamic Geometry*) на панели *Object* как для целевого объекта, так и для объекта-источника. Кроме того, теперь экспортируются все UV-карты и слои вертексного цвета, которые также будут доступны для материала, применяемого к целевому объекту. Более подробно эта возможность описана в *соответствующей главе* документации.

- Улучшения “Примеров кода”.

Появилась функция “*Make project*”. Она позволяет создать новый проект, основанный на примере кода, что даёт пользователю возможность использовать готовые примеры для дальнейшей разработки.

- Улучшения камеры.

Добавлена поддержка методов подгонки изображения *Horizontal* и *Auto*.

- API для настройки Wind Bending.

Теперь параметры Wind Bending могут быть установлены при помощи метода *set\_wind\_bending\_params* и получены при помощи *get\_wind\_bending\_params*. Следует заметить, что эти функции работают только для динамических объектов. Кроме того, отдельное меню для настройки Wind Bending добавлено в *Просмотрщик сцен*.

- Настройка качества отражений и размытия теней.

В модуль *config* добавлены новые параметры *shadow\_blur\_samples* и *reflection\_quality*. Они могут быть заданы при помощи метода

set.

- Поддержка индивидуальной настройки аддона Blend4Web.

Для упрощения настройки аддона Blend4Web появилась возможность добавлять поля `b4w_custom_prop` в блоки данных Object и Scene. Эти параметры экспортируются в формат Blend4Web и доступны при помощи методов `scenes.get_custom_prop` и `objects.get_custom_prop`. Пример использования можно увидеть в файле `addons/blend4web/custom_prop_example.py`, входящем в SDK.

- Изменения экспорта UV-слоев.

Удалено ограничение в два UV-слоя на меш. Кроме того, если UV-слой не задан, по умолчанию используется активный, а не первый в списке, как раньше. Неиспользуемые UV-слои не экспортируются, за исключением случаев, когда для объекта активирован параметр `Dynamic Geometry & Materials`.

## Изменения

- Изменения в Менеджере проектов.

Тип движка “External” объявлен устаревшим и автоматически заменяется на тип “Copy”.

- Теперь пустые слоты материалов экспортируются корректно и не останавливают процесс экспортации.
- Аддон перемещён в отдельную категорию Blend4Web.
- В интерфейс Blender добавлены предупреждения о неподдерживаемых ограничителях и форматах текстур.
- Теперь смешивание скелетных анимаций корректно работает с двумя анимациями. Эта возможность остаётся экспериментальной, API для неё может меняться в будущем.
- Теперь при динамической загрузке или выгрузке статичных физических объектов последние корректно влияют на сцену.
- Ряд улучшений в модуле `fps`. В метод `enable_fps_controls` добавлено несколько параметров для настройки. Подробнее изменения описаны в документации API.
- Изменения API

Методы `set_canvas_offsets`, `update_canvas_offsets` и `force_offsets_updating` модуля `container` объявлены устаревшими и будут удалены в будущих релизах. Вместо них следует использовать метод `client_to_canvas_coords` или `client_to_element_coords`.

Добавлен модуль `screen`.

Методы `request_fullscreen_hmd`, `enable_split_screen`, `disable_split_screen`, `request_fullscreen`, `exit_fullscreen` и `check_fullscreen` объявлены устаревшими и будут удалены в будущих релизах. Вместо них следует использовать методы `request_fullscreen_hmd`, `request_split_screen`, `exit_split_screen`, `request_fullscreen`, `exit_fullscreen` и `check_fullscreen`.

Методы `draw_mixer_strip`, `plot_array` и `shot` объявлены устаревшими и перемещены в модуль `screen`. Модули `hud` и `screenshooter` объявлены устаревшими.

Метод `set_hmd_fov` объявлен устаревшим.

Метод API `util.ground_project_quat` переименован в `ground_project_cam_quat`.

Удалены следующие устаревшие методы: `camera.set_move_style`,  
`camera.has_vertical_limits`,  
`camera.move_pivot`,  
`camera.get_velocity_params`,  
`camera.clear_vertical_limits`,  
`camera.set_look_at`,  
`camera.rotate_target_camera`,  
`camera.zoom_object`, `camera.set_pivot`, `camera.rotate_hover_camera`,  
`camera.get_hover_cam_pivot`, `camera.get_eye`, `camera.get_pivot`,  
`camera.hover_cam_set_translation`, `camera.set_hover_pivot`,  
`camera.get_hover_angle_limits`,  
`camera.apply_vertical_limits`,  
`camera.apply_distance_limits`,  
`camera.get_vertical_limits`,  
`camera.get_horizontal_limits`,  
`controls.register_mouse_events`,  
`controls.register_touch_events`,  
`controls.unregister_keyboard_events`,  
`controls.unregister_wheel_events`,  
`controls.unregister_device_orientation`,  
`controls.unregister_mouse_events`,  
`controls.unregister_touch_events`,  
`controls.unregister_device_orientation`,  
`app.resize_to_container`,  
`app.set_camera_move_style`,  
`app.disable_controls`,  
`hmd.reset`,  
`animation.get_first_armature_object`,  
`animation.get_bone_translation`,  
`util.line_plane_intersect`, `util.is_mesh`, `util.is_armature`, `tsr.create_sep`,  
`sfx.is_play`,  
`scenes.check_object`,  
`scenes.get_object_dg_parent`,  
`nla.check_nla_scripts`,  
`main.resize`,  
`main.global_timeline`,  
`main.get_canvas_elem`.

Добавлены методы `add_click_listener` и `remove_click_listener`.

- Изменения режима VR.

Теперь переключение в режим VR происходит без перезагрузки приложения (при использовании мобильных браузеров или браузеров с

поддержкой WebVR).

Отключена авторотация камеры в режиме VR.

Добавлена поддержка WebVR 1.1.

- Теперь метод `is_character` не возвращает значение `true`, если приложение не использует физику.
- Параграф *Нестандартное расположение и ориентация элемента Canvas* переписан в соответствии с изменениями в движке.

Кроме того, добавлен новый параграф *Мобильные веб-приложения*, затрагивающий некоторые аспекты ориентации и масштабирования страницы браузера.

- В документацию добавлен раздел *Методы API для работы с материалами*. В нём описывается использование методов модуля API `material`.
- Структура объекта `SSAOParams`, используемого в методах `set_ssao_params` и `get_ssao_params`, стала более корректной.
- Метод `object_distance` объявлен устаревшим. Вместо него следует использовать метод `distance`.
- Методы `create_ray_sensor` и `new_event_track` больше не принимают нефизические объекты в качестве аргументов (это приводило к крэшу движка).
- Теперь видеотекстуры не могут изменяться при помощи метода `change_image`.

## Исправления

- Исправлен баг, приводивший к тому, что описание якоря состояло не только из текстовых нод.
- Исправлен крэш, возникавший при использовании некорректных материалов системой частиц типа `Emitter`.
- Исправлены ошибки, возникавшие при использовании связанных объектов или групп с одинаковыми именами.
- Удалена дублирующая панель `Simplify`, которая создавалась Blend4Web в других системах рендеринга.
- Исправлено кэширование текстур при использовании несовместимых источников.
- Исправлены нормали частиц эмиттера.
- Исправлен рендеринг в текстуру, который в предыдущем релизе работал некорректно.
- Ряд исправлений в методе `prefetch`.
- Исправлена SRGB-коррекция цвета в режиме `Ultra`.
- Исправлено снижение FPS в браузерах для iOS.

- Исправлена загрузка исполняемых файлов в случае, если директория tmp не поддерживает их запуск. В этом случае используется стандартная tmp-директория Blender, путь к которой может быть задан вручную.
- Более корректное раскрытие меню *Add* в нодовом редакторе.
- Файл assets.json удалён из всех blend-файлов.
- Исправлена перезагрузка бинарного модуля при нажатии F8.
- Исправлена работа конфигуратора HMD.
- Исправлен баг, приводивший к тому, что видимость слоёв сцены в Blender могла изменяться после экспорта.
- Исправлено некорректное поведение видеотекстур в Firefox.
- Исправлены тени и отражения для динамической травы.
- Исправлен крэш при компиляции шейдеров, происходивший в том случае, если был выбран профиль качества LOW.
- Исправлен выбор объектов и тени от объектов с нодовыми материалами, использующими режим прозрачности Alpha Clip.
- Исправлена проверка неподдерживаемых размеров текстур.
- Исправлена функция Override Bounding Volumes для объектов с ключами формы.
- Исправлен выбор и подсветка LOD-объектов.
- Исправлен крэш движка, возникавший, если к какому-либо объекту в Blender применялся модификатор Array.
- Исправлены баги, связанные с методом `set_ssao_params`.
- Google Closure Compiler, используемый для компиляции скриптов движка и приложений, обновлён до последней версии. Это помогло исправить несколько мелких багов.
- Исправлен крэш движка при применении ключа формы к объекту с системой частиц типа HAIR или EMITTER.
- Исправлено свойство “Hidden” объектов типа EMPTY, используемых в качестве якорей.

## v16.12

### Новый функционал

- Улучшения в менеджере проектов.

В редактор файлов проекта добавлены кнопки **New File** и **Save As**. Кроме того, теперь редактор подсвечивает изменённые файлы для упрощения навигации.

Теперь сборщик приложений Менеджера проектов уменьшает скомпилированные HTML-файлы, что ускоряет их загрузку.

- Поддержка новых нод **Cycles**.

Добавлена поддержка ноды **Emission**. Эта нода относится к категории **Shader** и отвечает за излучение материалом света. Среди входных параметров этой ноды - **Color** и **Strength**. В Blend4Web эта нода влияет только на вид материалов, в которых используется, и не превращает объекты в источники света.

- Улучшения рендеринга LOD-объектов.

Теперь объекты с уровнями детализации объединяются в батчи в зависимости от их пропорций и настроек расстояния. Таким образом, движок визуализирует LOD-объекты с максимальной эффективностью, стараясь сохранять достаточное количество объектов, объединённых в батчи. Для настройки этой возможности на панель *Scene->Objects Clustering* добавлен параметр **LOD Cluster Size Multiplier**.

- Новые методы API **prefetch** и **unfetch**.

Первый метод используется для предварительной загрузки и кэширования ресурсов сцены (текстур, звуков, файлов сцены и т.п.). Второй метод применяется для очистки кэша.

- Поддержка выбора объектов на VR-устройствах.

Добавлен метод **pick\_center**. Этот метод использует функцию выбора объектов и позволяет пользователю выбрать объект, находящийся в центре окна приложения. Он работает как на VR-устройствах, так и в обычных приложениях.

- Улучшения интерфейса аддона.

Из аддона удалена панель сервера разработки. Добавлены подробные предупреждения о том, что сервер разработки недоступен в аддоне.

- Кэширование текстур.

Теперь все текстурные изображения кэшируются. Если запрашивается новая текстура с похожими параметрами, она берётся из кэша. Это снижает расход памяти и значительно увеличивает быстродействие метода **change\_image** для нескольких объектов.

## Изменения

- Изменения в Менеджере проектов.

Переработана структура файлов Менеджера проектов. Теперь новые проекты размещаются в директории `projects`. Все медиаданные размещаются в каталоге проекта.

Обратите внимание, что в новой файловой структуре метод `get_std_assets_path()` больше не возвращает путь к директории проекта. Вместо него следует использовать метод `get_assets_path()`.

Для *перевода старых проектов на новую структуру* добавлена кнопка `upgrade file structure`.

Текст в редакторе Менеджера проектов теперь выделяется пробелами, а не табуляциями.

- Теперь источники света с неподдерживаемым типом `Area` при экспорте преобразуются в источники типа `Sun`. При этом в консоли браузера выводится соответствующее сообщение об ошибке.

## Исправления

- Исправлен баг с видеотекстурами в мобильной версии Chrome.
- Исправлен баг с типом прозрачности “Alpha Sort” для статических объектов.
- Исправлен баг, приводивший к некорректному рендерингу материалов, использующих карты нормалей.
- Исправлен баг, приводивший к тому, что уровни детализации объекта переключались с заметной задержкой, во время которой не выводился ни один из уровней.
- Исправлен баг с динамической травой в WebGL 2.0.
- Исправлена ошибка взаимодействия с планшетами Wacom.
- Исправлено сообщение “Export to different disk is forbidden”, которое появлялось при использовании функции быстрого просмотра.
- Исправлен крэш при экспорте дупли-групп без привязанных объектов.
- Теперь прозрачные материалы корректно воспринимают значения прозрачности более 1.0.
- Нода `Parallax` больше не вызывает крэш нодового дерева в случае, если её входящая текстура используется в каких-либо `Outout`-нодах за исключением самой ноды `Parallax`.
- Теперь нодовые материалы очищаются от неиспользуемых нод при экспорте.
- Исправлена привязка к камере в режиме VR.

## v16.11

### Новый функционал

- Добавлена частичная поддержка материалов, использующих ноды **Cycles**. Список нод **Cycles**, поддерживаемых на данный момент:

Нода **Material Output** работает почти так же, как **Output** из **Blender Internal**, однако использует вход шейдерного типа. На данный момент поддерживается только вход **Surface**. Эта нода определяет материал как использующий физический рендеринг (PBR-материал).

Нода **BSDF Diffuse** из категории **Shader**. Отвечает за диффузную составляющую освещения материала и не создаёт видимых отражений. Нода имеет входы **Color**, **Roughness** и **Normal**, а также один выход шейдерного типа.

Нода **BSDF Glossy** из категории **Shader**. Отвечает за блики на материале и за отражение окружающей среды. нода обладает входами **Color**, **Roughness** и **Normal**, а также одним выходом шейдерного типа. На данный момент поддерживается только распределение GGX. Использование этой ноды автоматически включает использование кубических отражений для объекта, использующего материал, даже если для этого объекта не включены отражения. Отражаемые объекты и окружение следует задавать с помощью стандартных настроек Blend4Web. На данный момент параметр **Roughness** не влияет на рендеринг отражений.

Нода **Mix Shader** из категории **Shader**. Используется для смешивания выходных данных нод, принадлежащих к той же категории. Два шейдерных входа и вход **Fac**, определяющий степень смешивания, а также один выход шейдерного типа.

Нода **Fresnel** из категории **Input**. Рассчитывает, какая часть падающего на материал света отражается от его поверхности (другая часть преломляется, проходя сквозь поверхность). Полученный весовой коэффициент может затем использоваться многослойными шейдерами с нодой **Mix Shader**. Результат зависит от угла между нормалью поверхности и вектором взгляда. Нода обладает входами **IOR** (коэффициент преломления) и **Normal**, а также одним выходом скалярного типа.

Нода **Layer Weight** из категории **Input**. Определяет весовые коэффициенты, используемые многослойными шейдерами с нодой **Mix Shader**. Входы - **Blend** и **Normal**. Выходы - **Fresnel** и **Facing**.

Другие поддерживаемые ноды - **Image Texture**, **Environment Texture**, **Object Info** и **Bump**.

Частично поддерживаемые ноды: **Texture Coordinate** (не поддерживается параметр *From Dupli*), **UV Map** (не поддерживается пара-

метр *From Dupli*) и **Geometry** (не поддерживаются выходы *Pointness* и *Parametric*).

Ноды, поддержка которых была добавлена в предыдущих релизах Blend4Web (и которые используются как **Cycles**, так и **Blender Internal**), также нормально работают с новыми PBR-материалами. Среди таких нод - **Color Ramp**, **Normal Map**, **Camera Data**, **Particle Info**, **RGB**, **Value**, ноды категории **Convertor** (за исключением нод **Blackbody** и **Wavelength**), ноды категории **Vector** и ноды категории **Color** (за исключением **Light Falloff**).

- Улучшения в менеджере проектов.

Появился редактор файлов проекта. Теперь файлы CSS, JavaScript, HTML, а также файл .b4w\_project можно редактировать прямо в Менеджере проектов.

- Добавлен новый аддон fps.

Использование этого аддона упрощает создание приложений с видом от первого лица. Аддон содержит два основных метода: `enable_fps_controls` и `disable_fps_controls`. Первый из них активирует стандартную схему управления с помощью геймпада и клавиатуры, включает управление камерой при помощи мыши и тач-интерфейса, а также поворот камеры с помощью VR-устройства (если включен VR-режим). Второй метод отключает всё перечисленное. Аддон также содержит дополнительные методы `bind_action`, `set_character_state_changing_cb`, `set_cam_smooth_factor`, `set_cam_sensitivity`, `set_plock_enable_cb` и `set_plock_disable_cb`.

- Улучшения навигационных мешей.

Теперь метод `navmesh_find_path` модуля `physics` строит траекторию в трёх измерениях, что позволяет строить траектории на вертикальных поверхностях (ранее этот метод мог использоваться только с горизонтальными поверхностями). Также изменены параметры метода `navmesh_find_path` и улучшен алгоритм A\*.

- Аддон Blend4Web стал удобнее в использовании.

Поле Blend4Web SDK Directory удалено из свойств аддона. Теперь путь к директории Blend4Web SDK строится автоматически.

## Изменения

- Изменения API

Изменены параметры методов `shot` и `canvas_data_url`.

В модуль `camera` добавлена новая функция `get_camera_angles_dir`. Она возвращает сферические координаты камеры с заданного направления вектора взгляда камеры.

- Упрощена настройка окружающей среды.

Теперь вода использует базовые настройки направления и силы ветра, если в сцене отсутствует соответствующее силовое поле и всегда является динамическим объектом. При отсутствии в сцене солнца для процедурного неба используется положение солнца по умолчанию.

- Теперь NPC AI подхватывает все анимации при загрузке.

Это несколько увеличивает время загрузки, но устраняет задержки, вызываемые анимацией NPC.

- Теперь кнопка `Fast Preview` доступна в режиме Cycles.
- Теперь методы `get_all_objects`, `get_object_by_name` и `check_object_by_name` не возвращают внутренние мета-объекты движка, не предназначенные для использования в приложении.

## Исправления

- Теперь Менеджер проектов совместим с устаревшим типом привязки движка `update`.
- Исправлен баг с определённой кодировкой в Менеджере проектов.
- Исправлен баг, приводивший к конфликту модулей с одинаковыми именами в Менеджере проектов.
- Исправлена ошибка рендеринга бликов на материале.

Эта ошибка возникала, когда материал с бликами типа Blinn освещался двумя или более источниками света.

- Исправлен баг с выгрузкой NLA.

Теперь из NLA удаляются все объекты, принадлежащие к выгруженной сцене.

- Исправлен баг с мерцанием в стереорежиме.

Этот баг возникал при использовании эффекта Motion Blur.

- Исправлена перезапись ресурсных файлов с одинаковыми именами в директории Temp при использовании Fast Preview.
- Исправлена ошибка сети, возникавшая при загрузке скриншота.
- Исправлен крэш шейдера при запуске некоторых сцен с настройкой качества LOW.
- Исправлен крэш движка при запуске сцен без объектов типа MESH.
- Исправлен метод `track_to_target`, который создавал некорректную анимацию приближения.
- Исправлена редкая ошибка с пустыми слотами текстур частиц, иногда приводившая к крэшу при экспорте.

- Исправлен рендеринг процедурных линий.
- Исправлена ошибка логической ноды `Play Animation`, приводившая к тому, что анимация останавливалась после первого проигрывания.
- Исправлен краш в модуле navmesh, происходивший в веб-браузерах, не поддерживающих метод `indexOf` для `TypedArray`.
- Исправлена ошибка Менеджера проектов, возникавшая при конвертации ресурсов в случае отсутствия `ffprobe`.

## v16.10

### Новый функционал

- Добавлена поддержка навигационных сеток.

В модуль `physics` добавлены два метода: `navmesh_get_island` возвращает ближайший сегмент навигационной сетки, а `navmesh_find_path` применяется для поиска пути.

Доступны два типа пути: основанный на центрах треугольников и более оптимальный `pulled string`.

Пример использования входит в состав приложения `Code Snippets`.

- Новая логическая нода `Set Camera Move Style`.

Эта нода может использоваться для изменения скорости и типа перемещения камеры. В качестве “цели” для камер типа `Target` и `Hover` может использоваться как объект, так и координаты точки.

- Поддержка тангенциального затенения для отредактированных нормалей.

Теперь тангенциальное затенение корректно работает с отредактированными нормалями.

- Улучшения Менеджера проектов.

Теперь приложения, `blend`-файлы и медиаданные проекта открываются в новых вкладках. Сам Менеджер проектов при этом остаётся в том же окне браузера, что увеличивает эффективность работы.

Команда `compile project` переименована в `build project`. Новое название точнее описывает функцию кнопки.

- Новые типы смешивания текстур окружения.

Теперь текстуры, используемые для расчёта освещения от окружающей среды, поддерживают все типы смешивания.

- Поддержка опции `Sphere` для источников света `Point` и `Spot`.

Эта опция позволяет задать дистанцию, на которой интенсивность света становится нулевой.

- Повёрнутые ограничивающие объёмы.

Теперь для отсечения невидимых объектов используются повёрнутые ограничивающие кубы. Кроме того, для динамических объектов теперь поддерживаются повёрнутые ограничивающие эллипсоиды.

- Теперь логическая нода JS `Callback` может вызываться синхронно.

Для остановки выполнения нодового дерева достаточно передать `true` из callback-функции. Когда нода закончит работу, передайте `false`.

- Новые методы API `get_fog_params` и `set_fog_params`.

Методы применяются для управления туманом в сцене. Параметры тумана включают такие свойства, как `fog_intensity`, `fog_depth`, `fog_start` и `fog_height`.

- Новый метод API `update`.

Метод позволяет обновлять позиции якорей.

- Добавлена поддержка разных типов буферов VBO.

Типы некоторых атрибутов изменены с `FLOAT` на `UNSIGNED BYTE` и `SHORT`, что снижает потребление памяти GPU, но не влияет на качество изображения. Это также влияет на производительность и может увеличить частоту кадров в некоторых сценах. Кроме того, размеры экспортируемых `.bin`-файлов несколько уменьшены за счёт того, что теперь цвета вертексов хранятся в переменных типа `UNSIGNED BYTE`, а не `SHORT`.

- Теперь в режиме `ULTRA` используется правильная, а не упрощённая функция SRGB-конверсии.

Эффекта наиболее заметен на тёмных участках изображения, где упрощённая функция даёт неточные цвета.

- Добавлены кнопки для включения и выключения опции `World Background` на панели 3D VIEW.

Кнопки находятся на вкладке `World` и должны использоваться при включенной опции `Render Sky`. При активированной опции `World Background` цвета окружающей среды видны в окне просмотра также, как в движке.

## Изменения

- Реорганизация проектов в SDK.

Теперь все проекты в SDK (в том числе обучающие) используют стандартную иерархию директорий Менеджера проектов.

- Из Просмотрщика сцен удалён список сцен.

Аналогичные возможности (просмотр сцен и данных проекта) теперь доступны в Менеджере проектов.

- Улучшена документация модуля API `input`, добавлены примеры использования функций.

- В целях оптимизации рендеринга сцен уменьшено количество *батчей*.

В наибольшей степени это влияет на тени и частицы типа Hair. Сцены, использующие эти возможности, теперь работают быстрее.

- Теперь клавиатура по умолчанию привязана к объекту `document`.

- Добавлен новый тип привязки движка `None`.

При выборе этого варианта Менеджер проектов не будет изменять Ваши проекты при сборке.

- Удалён тип привязки движка `Update`.

Для достижения аналогичных результатов вместо него следует использовать тип `Copy`.

- Добавлены облегчённые версии SDK.

Теперь доступны облегчённые версии пакетов Blend4Web PRO и Blend4Web CE, не включающие уроки и демонстрационные приложения. Пользователям с медленным интернет-каналом рекомендуется использовать эти версии.

- Добавлена поддержка текстуры глубины для Intel HD Graphics 3000, что делает возможным использование на этих устройствах таких эффектов, как тени, глубина резкости, сумеречные лучи и др.

- Теперь NLA-анимации получают значения начального и конечного кадров из самой *вертекской* анимации.

Это позволяет использовать для одного объекта несколько вертексовых анимаций, контролируемых NLA.

- Исправлено принудительное использование нод материалов низкого качества на некоторых мобильных устройствах (в том числе на iOS).

- API documentation for `BloomParams` has been added. Some parameters have been renamed.

- Изменены выходные данные сенсоров `mouse_click`, `mouse_move`, `touch_click` и `touch_move`. Теперь они представляют собой словарь с абсолютными координатами (`coords`) для всех вышеперечисленных сенсоров, а также `which` для обработчиков щелчков мыши и `gesture` для `touch_move`.

- В модуль `controls` добавлены логические функции `default_AND_logic_fun` и `default_OR_logic_fun`.

- Удалены устаревшие скрипты для кросс-компиляции бинарных модулей.

- Методы API `set_nodemat_value`, `get_nodemat_value`, `set_nodemat_rgb` и `get_nodemat_rgb` объявлены устаревшими и перемещены в модуль `material`.

## Исправления

- Исправлена функция `quat_to_euler`.
- Исправлена некорректная работа ноды `Normal Map` с неединичным параметром силы.
- Исправлено несколько проверок объектов логическими нодами.
- Исправлена невозможность замены текстуры на объекте (функция `change_image`), в случае, если эта текстура используется несколькими материалами.
- Теперь блики, рассчитываемые по модели `CookTorr`, выглядят так же, как в `Blender`.
- Теперь `Alpha Sort`-материалы корректно отображаются на простых (не “глубоких”) копиях объектов.
- Исправлены методы `create_pline_from_point_vec` и `set_pline_initial_point`.
- Теперь персонажи, управляемые с помощью модуля `npc_ai`, не проваливаются под землю при ходьбе.
- Исправлен баг, приводивший к тому, что два и более якоря типа `Custom Element` могли ссылаться на один и тот же id элемента.
- Исправлены отражения сферических билбордов.
- Исправлено воспроизведение звука после паузы для спикеров типа `Background Sound` и `Positional Sound` в браузере `Firefox`.

## v16.09

### Новый функционал

- Улучшения в интерфейсе веб-плеера.

Добавлена возможность настройки социальных кнопок, расположенных в правом нижнем углу загруженной сцены. Для этого перед запуском приложения необходимо настроить `атрибут socials`.

- Изменения координатной системы.

Теперь движок использует систему координат `Blender` вместо системы `OpenGL`. Это может вызывать несовместимости в логике приложений, поэтому мы рекомендуем разработчикам проверить свои приложения и внеси в них изменения в соответствии с новой координатной системой (ось `Z` направлена вверх).

- Оптимизация рендеринга геометрии

Для оптимизации рендеринга геометрии используется новый метод хранения данных в памяти GPU. Теперь данные о нормалях и касательных хранятся в TBN-кватернионах.

- Улучшения в менеджере проектов.

В утилиту *project.py* добавлена команда `update_modules`. С её помощью пользователь может обновить модули движка, используемые приложением. Эта возможность значительно упрощает процесс обновления файлов проекта до новых версий Blend4Web.

- Автоматическое определение качества изображения.

При установке профиля качества изображения AUTO (`P_AUTO`) в модуле `app` приложение может быть запущено как с низкими, так и с высокими настройками качества, в зависимости от конфигурации системы. Кроме того, метод `test_performance` позволяет запустить низкоуровневый тест производительности.

- Поддержка компрессии текстур PVRTC.

Добавлена поддержка алгоритма PVRTC, что позволяет использовать компрессию текстур на iOS и на Android-устройствах, основанных на PowerVR.

- Добавлены настройки качества теней.

Появилась возможность установки разных профилей качества для мягких теней: 16x, 8x и 4x.

- На вкладку `Render` добавлена панель `Shading`. Она содержит опцию `World Space Shading` и кнопку `Set Recommended Options` для автоматической настройки Blender на большую совместимость с Blend4Web.

- Добавлена поддержка Blender 2.78.

Опция `World Space Shading` и поддержка освещения от окружающей среды в режиме GLSL делают вид сцены в окне просмотра предельно близкой к виду сцены в самом движке.

Теперь новая шейдерная нода `Normal Map` поддерживается полностью.

## Изменения

- Переработана система присвоения шейдерных директив. Это упрощает отладку и уменьшает время загрузки движка.
- Рендеринг якорей на некоторых устройствах ускорен благодаря использованию свойства CSS “translate3d”
- Улучшена валидация шейдеров, что снижает количество ошибок второго рода. Также добавлены новые описания ошибок.

- Для увеличения производительности большая часть вычислений в шейдерах переведена с матриц на TSR.
- Изменён порядок панелей на вкладке **Render**.

## Исправления

- Исправлено зависание Samsung Internet Browser при наличии в сцене большого количества якорей.
- Исправлен не работавший атрибут Веб-плеера “alpha”.
- Исправлено сообщение об ошибке компиляции/линковки шейдеров.
- Исправлен поворот объекта с помощью логической ноды **Transform Object** при использовании переменной для задания угла поворота.
- Исправлено поведение логической ноды **Play Animation** в ситуации, когда одновременно присутствуют запечённый и незапечённый варианты анимации.
- Исправлены настройки качества для плоских отражений.
- Исправлены методы `get_translation_rel` и `get_rotation_rel`.
- Исправлен параметр `dof_distance` метода `set_dof_params`.
- Исправлен баг, приводивший к крашу движка при выборе объекта с опциями `Do Not Render` и `Enable Outlining`, включёнными одновременно.
- Исправлена физика систем частиц.

## v16.08

### Новый функционал

- Добавлена поддержка опции **Tangent Shading** для материалов.

Эта опция может использоваться для имитации анизотропных поверхностей, например, полированного металла, волос и т.п.

- Добавлена поддержка шейдерной ноды **Normal Map**.

Эта нода может применяться для преобразования цветовой информации текстуры в карту нормалей. Поддерживаются входящие данные в тангенциальном, объектном или мировом координатном пространстве. Параметр **Strength** управляет степенью смешивания значений, извлечённых из текстуры, с нормалями самого объекта.

Для того, чтобы изображение в окне Blender выглядело так же, как в сцене Blend4Web, цветовое пространство текстуры должно иметь тип **Non-Color**.

- Оптимизация движка.

Произведена оптимизация ядра движка и “сборщика мусора”. Добавлена поддержка расширения `OES_vertex_array_object` для соответствующего оборудования. Также, с целью уменьшения числа вызовов отрисовки, произведена оптимизация использования шейдеров и постоянных. Всё это должно уменьшить время загрузки сцен и отзывчивость рендеринга, особенно на системах с медленным CPU.

- Улучшения интерфейса Просмотрщика сцен.

Теперь панель `Tools & Debug` находится прямо под панелью `Scenes`.

На панель `Tools & Debug` добавлена кнопка `Min capabilities mode`. Этот параметр заставляет Просмотрщик сцен воспроизводить загруженную сцену так, как если бы она работала на системе с низкой производительностью (например, на iOS-устройстве). Это позволяет узнать, какие материалы в сцене могут работать не так, как задумано, на системе с низкой производительностью.

- Улучшения редактора нормалей.

Добавлен режим редактирования нормалей `Offset`.

Операция `Average` теперь может применяться к неразделённым нормалям.

Добавлена возможность ручного ввода угла поворота нормали, как при повороте обычного объекта.

- Новый метод API в модуле `textures`.

Добавлен метод `get_texture_names`. Он позволяет получить имена всех текстур объекта.

- Новый метод API в модуле `lights`.

Добавлены методы `get_light_color`, `set_light_color`, `get_light_energy` и `set_light_energy`. Они используются для работы с цветом и световой энергией источника света.

- Добавлена поддержка свойства объектов `Hidden`.

Включение этого свойства сделает объект невидимым при запуске сцены.

- Поддержка шейдеров GLSL ES 3.0.

Теперь движок автоматически выбирает, какую версию шейдерного языка OpenGL использовать для компиляции шейдеров. Выбор зависит от контекста WebGL: для WebGL 1 используется GLSL ES 1.0, для WebGL 2 - GLSL ES 3.0. Шейдерная система движка и препроцессор также были изменены для совместимости с обоими вариантами.

Это снимает ограничения на добавление в движок новых возможностей WebGL 2 без проблем с совместимостью.

- Улучшения Анализатора Шейдеров.

Теперь анализатор шейдеров (метод `analyze_shaders`) выводит низкоуровневый ассемблерный код, что полезно для оценки и оптимизации шейдеров.

## Изменения

- Улучшения алгоритма расчёта боке для эффекта глубины резкости.

Уменьшена интенсивность эффекта *pixel bleeding*, приводившего к тому, что находящиеся в фокусе объекты на переднем плане “вытекали” на размытый фон.

Добавлен параметр **Foreground Blur**. Его включение делает менее заметными чёткие силуэты объектов вне фокуса на фоне объектов в фокусе.

Расстояния, на которых эффекта размытия начинает проявляться и достигает максимальной интенсивности, теперь можно контролировать с помощью параметров **Front Start** и **Front End** (для передней плоскости) и **Rear Start** и **Rear End** (для задней плоскости).

- Рендеринг отражения окружающей среды теперь включен по умолчанию.
- Улучшения алгоритма валидации шейдеров.

Если материал не может быть отрендерен на низкопроизводительных системах, он заменяется специальным материалом розового цвета при отладке и удаляется из финального продукта.

## Исправления

- Исправлена инверсия длины для групп вершин.

Исправлена инверсия оператора систем частиц типа *HAIR* для вершинных групп, задающих длину.

- Исправлен поворот нормалей для преобразованных объектов в *Редакторе нормалей*.
- Исправлен редкий краш движка при загрузке сцены, использующей нодовую логику.
- Исправлена панель материалов в Просмотрщике сцен. Ранее она не работала для некоторых материалов, подлежащих редактированию.
- Исправлен краш движка на мобильных устройствах.
- Исправлено мерцание экрана в мобильных браузерах.
- Исправлен краш при загрузке динамической воды.
- Исправлено влияние карт нормалей на отражения для стековых материалов.
- Отключена серая подсветка при нажатии в Web Player на iPad.

## v16.07

### Новый функционал

- Оптимизация рендеринга систем частиц.

Теперь для рендеринга частиц типа `Object` используются возможности инстансинга WebGL (расширение `ANGLE_instanced_arrays` или WebGL 2.0). Использование инстансинга снижает потребление памяти и, в некоторых случаях, увеличивает скорость рендеринга.

- Снижение задержки ввода на мобильных устройствах.

Теперь на мобильных версиях Chrome, Firefox и Safari движок игнорирует события мыши, соответствующие событиям, которые уже были обработаны внутренними сенсорами Blend4Web. Это уменьшает время реакции на действия пользователя.

- Улучшения аудиосистемы.

Автоматическое создание аудиоконтекста. Флажок `Audio` удалён из аддона, при необходимости аудиоконтекст создаётся автоматически.

Реализация эффекта Допплера. Эффект Допплера отсутствует в современных версиях WebAudio, так как теперь предполагается, что разработчики приложений должны реализовывать этот эффект самостоятельно. Начиная с данного релиза, вы можете использовать реализацию,строенную в движок. Для этого добавлена опция `Enable Doppler`, которая заменяет старую `Disable Doppler`.

Переработан аудиоинтерфейс. Поддержка таких параметров, как `Speed`, `Doppler` и `Distance Model` максимально приближает его к собственному интерфейсу Blender.

Новая опция спикера `Auto-play`. Позволяет включить воспроизведение аудио по умолчанию.

Предварительная поддержка сложных аудиопотоков. Новые опции `Loop Start` и `Loop End`, а также метод API `loop_stop`, позволяют сложные петли, включающие начальную, закольцованную и конечную секции в одном аудиобуфере. С помощью нового API можно создавать, например, стандартную огибающую ADSR (Attack, Decay, Sustain, Release).

- Поддержка полисенсорного выбора в сенсоре выбора.

Теперь сенсор выбора событийной модели поддерживает мультитач-выбор.

- Новые возможности анимации нодовых материалов.

Теперь любая нодовая анимация может применяться к любому нодовому материалу или его вложенной нодовой группе. Кроме того,

появился новый метод `apply_ext()`, позволяющий задать материал или нодовую группу, которую требуется анимировать.

- В модуль `preloader` добавлен новый метод `create_preloader`.
- В модуль API `config` добавлены конфигурационные параметры `max_fps`, `max_fps_physics`, `use_min50`, `anisotropic_filtering`, `shadows`, `reflections`, `refractions`, `ssao`, `dof`, `god_rays`, `bloom` и `motion_blur`.

## Изменения

- Оптимизация выделения объектов.

Теперь для выделения объектов использует очень узкий усечённый конус и малый размер фреймбуфера ( $1 \times 1$  пиксель). Кроме того, снижено количество вызовов метода `pick_object`. Всё это увеличивает производительность сенсора выбора.

- Улучшения эффекта глубины резкости (Depth-of-field, DOF).

Увеличена производительность эффекта глубины резкости.

Добавлен экспериментальный алгоритм расчёта DOF. Теперь степень размытия изменяется в зависимости от расстояния, а на размытых объектах появляется эффект боке. Новый алгоритм включается с помощью опции `Bokeh` на панели свойств камеры в Blender.

- Изменения API

Метод `create_simple_preloader` модуля `preloader` объявлен устаревшим и будет удалён в будущих релизах.

## Исправления

- В Просмотрщике сцен исправлен не работавший слайдер `Canvas Resolution Factor`.
- Исправлены методы `get_matrix`, `set_matrix`, `get_matrix_rel` и `set_matrix_rel` модуля `transform`.
- Исправлено определение метода `resize_to_container` модуля `container`.
- Исправлено снижение производительности при изменении размеров `canvas`'а.
- Исправлена ошибка, приводившая к тому, что в сцене отсутствовали камеры из других сцен или дупли-групп.
- Исправлена некорректная прозрачность `canvas`'а при использовании эффекта засветки.
- Исправлена кнопка `Stop All` на вкладке `Animation` Просмотрщика сцен.
- Исправлено выделение объектов для стековых материалов.

- Исправлен эффект свечения в браузере Safari.

## v16.06

### Новый функционал

- Улучшения Fast Preview.

При необходимости сервер разработки копирует все внешние ресурсы во временную папку. Это делает возможным предварительный просмотр сцен, расположенных вне SDK (в других папках, на flash-накопителях и пр.)

- Улучшения в менеджере проектов.

Поддержка библиотеки материалов. Новая опция позволяет скопировать исходные файлы библиотеки материалов в директорию проекта.

В утилиту *project.py* добавлена новая опция `--ignore`. Эта опция даёт пользователю возможность игнорировать файлы при компиляции или развёртывании проекта.

В утилиту *project.py* добавлена новая команда `check_modules`. Эта команда позволяет проверить отсутствующие или неиспользуемые модули.

- Экспериментальная поддержка системы виртуальной реальности *GearVR*.

Теперь движок поддерживает API WebVR 1.0, что делает возможным использование устройств с GearVR.

- Поддержка текстур в форматах GIF и BMP.

Теперь изображения в форматах BMP и GIF (без анимации) могут использоваться в качестве текстур.

- Улучшения в утилите для конвертации ресурсов.

Теперь поддерживается большое количество расширений медиафайлов. Более подробная информация доступна в [документации](#).

- Автоматическое определение пути экспорта в аддоне Blender.

При первом экспорте проекта, созданного с помощью менеджера проектов, путь к каталогу медиаданных будет установлен автоматически.

- Поддержка параллельных анимаций в Редакторе логики

Теперь Редактор логики поддерживает использование нескольких параллельных анимаций. Ранее к объекту могла применяться только одна анимация. Теперь для одного объекта поддерживается по одной анимации каждого типа. Максимальное количество анимаций объекта, таким образом, достигает 8.

- Улучшения отладочных возможностей Просмотрщика сцен.

Добавлен специальный режим для отладки времени рендеринга объектов в Просмотрщике.

- Опция **Lens Flare** для материалов.

Новое свойство находится на вкладке опций рендеринга материала.

Следует заметить, что эта возможность работает только при наличии в сцене источника света типа **Sun**.

- Параметры источников света Clip Start и Clip End

Параметры Clip Start и Clip End поддерживаются для настроек теней источников света типа **Point** и **Spot**.

- HTML-элементы в Веб-плеере.

Новые мета-элементы добавлены к HTML-шаблонам Веб-плеера для совместимости с различными социальными сетями.

## Изменения

- Ноды материалов теперь работают в мировом пространстве координат выпорта Blender.

- **Geometry**

Выход **Normal** выдаёт данные в мировом пространстве координат Blender.

Выход **View** выдаёт данные в пространстве координат камеры Blender.

- **Texture**

Вход **Vector**, используемый для текстур окружающей среды, принимает данные в мировом пространстве координат Blender.

Выход **Normal** выдаёт данные в мировом пространстве координат Blender.

- **Material**

Вход **Normal** принимает данные в мировом пространстве координат Blender.

Выход **Normal** выдаёт данные в мировом пространстве координат Blender.

- **Lamp Data**

Выход **Light Vector** выдаёт информацию в мировом пространстве координат Blender.

- **B4W Vector View**

Нода принимает данные в мировом пространстве координат Blender.

- **B4W Reflect**

Первый вход ноды (используется для вектора взгляда) принимает информацию в пространстве координат камеры Blender.

Второй вход ноды (применяется для нормалей) также принимает информацию в мировом пространстве координат.

Выход ноды выдаёт данные в мировом пространстве координат Blender.

- Якоря и выбор цвета отключены в стереорежиме.
- Сенсор клавиатуры больше не активируется при использовании горячих клавиш браузера.
- Изменения API

Метод `resize_to_container` модуля `app` объявлен устаревшим и будет удалён в последующих релизах. Вместо него следует использовать метод `resize_to_container()` модуля `container`.

- Параметр `Key` эффекта засветки переименован в `Intensity`.
- Устаревший функционал.

Были удалены следующие методы: `textures.get_canvas_texture_context()`, `textures.update_canvas_texture_context()`.

Были удалены следующие логические ноды: `Select`, `Select & Play Timeline`, `Select & Play Animation`.

- Теперь сообщения об ошибках экспорта и предупреждения включают ссылки на документацию.
- Отчёт о некорректной директории аддона.

Теперь некорректное имя директории аддона приводит к появлению соответствующего предупреждения.

- Обработка ситуации применения воды без ветра.

Если шейдер воды используется без ветра, появляется предупреждение.

- Добавлены консольные сообщения об ошибках при использовании неподдерживаемых форматов изображений, видео- и аудиофайлов.

## Исправления

- Исправлено падение производительности в стереорежиме.
- Исправлена обfuscация `window.screen.orientation.angle`.

- Исправлена ошибка рендеринга в браузерах Firefox с включённым контекстом WebGL 2.0.
- Исправлен рендеринг изображения для правого глаза в стереорежиме.
- Исправлен рендеринг светящихся материалов с включённой опцией **Terrain Dynamic Grass**.
- Теперь параметр угловой скорости частиц становится недоступным (поскольку избыточен), если деактивирована опция **Rotation**.
- Исправлен рендеринг частиц со стандартными (не нодовыми) материалами с типом прозрачности, отличным от **Opaque**.
- Исправлен ряд причин утечки памяти на GPU.
- Исправлен крэш движка, возникавший при использовании некорректного значения **id** в логической ноде **JS Callback**.
- Исправлен крэш движка, происходивший в редких случаях при наличии в новом материале нескольких нод **Texture** с одной и той же текстурой.
- Исправлена ошибка, из-за которой вход **Diffuse Intensity** нод **Material** и **Extended Material** игнорировал входящее значение.

## v16.05

### Новый функционал

- Экспериментальный конфигуратор HMD.

Был добавлен аддон `hmd_conf.js`. Он даёт пользователям возможность вручную устанавливать параметры HMD. На данный момент конфигуратор доступен в Просмотрщике сцен и может быть вызван нажатием кнопки, расположенной на панели *Stereo View*. За более подробной информацией обратитесь к документации разработчика по модулю API `hmd_conf`.

- Экспериментальный конфигуратор геймпадов и контроллеров

В API был добавлен аддон `gp_conf`. Он содержит две главные функции: `show` и `hide`. На данный момент он поддерживает контроллеры и рули.

Чтобы протестировать этот конфигуратор, активируйте опцию *Gamepad Settings* на панели *Input Devices* Просмотрщика сцен.

- Новые примеры кода.

Добавлены новые примеры кода. Их названия - *Gamepad*, *Webcam* и *Change Image*. Первое из них поможет вам добавить поддержку геймпадов в ваш проект, второе демонстрирует возможности API камеры, а третье - применение новых функций для работы с текстурами.

- Новая шейдерная нода **Vector Transform**.

Эта нода может применяться для преобразования входных данных между координатными пространствами объекта, мира или камеры.

- Новая логическая нода **Empty**

Эта нода может применяться для упрощения перенаправлений в сложных логических деревьях.

- Программная замена текстурных изображений.

Добавлен новый метод `change_image`. С его помощью можно заменять текстуры и кубические карты средствами API.

- Новый кластерный алгоритм батчинга.

Теперь батчинг (процесс объединения похожих объектов для повышения производительности) основан на новом кластерном алгоритме, который применяется при экспорте. Это оптимизирует весь процесс батчинга и делает его более “интеллектуальным” по сравнению со старым алгоритмом, основанным на регулярной сетке.

- Улучшения в менеджере проектов.

Теперь для сохранения однородности в именах всех опций командной строки в утилите `project.py` используется символ - вместо символа подчеркивания, например, опция `--engine_type` теперь выглядит как `--engine-type`.

В утилиту `project.py` добавлена новая опция `--assets-dest`. Эта опция даёт пользователю возможность задать директорию, в которой будут храниться медиаданные развертываемого проекта.

- Новые материалы в Библиотеке Материалов.

В Библиотеку Материалов для PRO-версии SDK добавлено четыре новых материала: Золото, Рифленое Стекло, Шелк и Бархат.

## Изменения

- Улучшения интерфейса и повышение удобства экспорта объектов, для которых не выбрана ни одна из опций списка `Export Options`.

Опция `Export Shape Keys` выбирается автоматически при добавлении ключа формы.

Опция `Export Vertex Animation` выбирается автоматически после запекания вершинной анимации.

Опция `Apply Scale and Modifiers` автоматически применяется во время экспорта ко всем объектам, которые соответствуют следующим условиям:

Не имеют вершинной анимации.

Не имеют родительского объекта

Не имеют скелета.

Физика объекта отключена.

Опция **Apply Modifiers** автоматически применяется во время экспорта ко всем объектам, к которым применяются модификаторы и которые соответствуют следующим условиям:

Не имеют вершинной анимации.

Не имеют скелета.

- Теперь Z-сортировка для материалов с типом прозрачности **Alpha Sort** выполняется с учётом размеров объектов.

Это может привести к снижению производительности из-за более частых обновлений сортировки, но исправляет ошибки сортировки для объектов маленьких размеров.

- Новый алгоритм отсечения по зоне видимости.

Теперь зона видимости рассчитывается с использованием материалов мешей, а не объектов.

- Изменения сенсоров геймпада.

В модуль `input` добавлены новые идентификаторы кнопок и осей. Кроме того, имена сенсоров геймпада изменены на `create_gamepad_btn_sensor` и `create_gamepad_axis_sensor`.

- Теперь логическая нода **Entry Point** с параметром **Run From Script** может вызываться через API многократно.

- Изменения API

Метод `create_rotation_preloader` модуля `preloader` объявлен устаревшим и будет удалён в последующих релизах.

Метод `create_advanced_preloader` модуля `preloader` объявлен устаревшим и будет удалён в последующих релизах.

В модуль `input` добавлен метод `set_config`.

Метод `register_device` модуля `input` объявлен устаревшим и будет удалён в последующих релизах.

Удалён флаг `gyro_use` модуля `config` (больше не требуется).

Теперь функции модуля `storage` имеют optionalный последний параметр.

В модуль `util` добавлен метод `is_ie11`.

- Теперь состояние клавиатурной клавиши хранится в полезной нагрузке сенсора клавиатуры.

За более подробной информацией обратитесь к разделу документации разработчика, посвящённому методу `create_keyboard_sensor`.

- Ссылка на отладочную страницу в сообщении Веб-плеера “Browser could not initialize WebGL” заменена на более подходящую.

Теперь она ведёт на раздел документации “Проблемы и решения”.

- Документация значительно доработана, добавлены подробные описания возможностей якорей и Viewport Alignment.

## Исправления

- Исправлена проблема с некорректными автоматическими обновлениями в аддоне.
- Исправлено поведение якорей с логическими нодами.
- Исправлена обfuscация логической ноды JS Callback.
- Исправлена проблема с некорректным удалением сенсоров в методе `controls.remove_sensor_manifolds`.
- Исправлена проблема с некорректным возрастом и степенью сжатия zip-файлов дистрибутива.
- Исправлен мультисэмплинг в браузерах Firefox.

Частично поддерживаемый в реализации WebGL 2.0 для браузеров Firefox мультисэмплинг отключён.

- Исправлена ошибка, приводившая к тому, что разрешение canvas'a не менялось в зависимости от настроек качества антиалиасинга.
- Исправлена ошибка, приводившая к тому, что интенсивность диффузного цвета применялась к материалу с типом затенения Shadeless.
- Исправлен крэш движка при использовании нод B4W\_GLOW\_OUTPUT и B4W\_REFRACTION в одном и том же материале.
- Исправлено дублирование панели Custom Properties в интерфейсе Blender для объектов типа Font, Curve, Lattice, Armature и MetaBall.
- Исправлено несколько ошибок в скомпилированной версии движка.
- Исправлены ошибки в модулях input и controls.
- Исправлен некорректный батчинг нодовых материалов при использовании ноды ColorRamp.
- Исправлена некорректная остановка HTTP-сервера.

## v16.04

### Новый функционал

- Поддержка геймпадов и контроллеров

Появилась возможность использовать геймпады и контроллеры в качестве устройств ввода. Для работы с этими устройствами были добавлены новые функции. Первая из них - `create_gamepad_btns_sensor`. Она обрабатывает нажатия на кнопки геймпада. Вторая функция - `create_gamepad_axes_sensor`, она обрабатывает оси геймпада.

- Улучшения нодового редактора логики.

Была добавлена логическая нода *JS Callback*. Она может применяться для вызова callback-функций, заданных в вашем B4W-приложении. Для callback-функций поддерживаются входные и выходные параметры.

В ноду *Entry Point* была добавлена опция *Run From Script*.

В API был добавлен модуль `logic_nodes`. Он содержит методы для работы с нодовым редактором логики.

Добавлен метод `append_custom_callback`. Он применяется для регистрации JavaScript-функций с тем, чтобы их можно было использовать в логической ноде *JS Callback*.

Добавлен метод `remove_custom_callback`. Он применяется для удаления зарегистрированных callback-функций.

Добавлен метод `run_entrypoint`. Он позволяет активировать ноду *Entry Point* из API.

- В модуль `controls` была добавлена новая функция `create_hmd_position_sensor`.

Эта функция позволяет создать особый сенсор для отслеживания позиции HMD-устройств.

- Теперь для систем частиц типа `Emitter` поддерживаются опции `Tilt Angle` и `Tilt Random`.
- Отражения для прозрачных объектов.

Ранее только непрозрачные объекты могли иметь отражения. Начиная с этого релиза, полупрозрачные объекты также поддерживаются.

- Обновления математических модулей.

Математические модули `vec3`, `vec4`, `quat`, `mat3` и `mat4` теперь основаны на `glMatrix v2.3.1`. Новая версия добавляет следующие методы: `hermite`, `bezier`, `slerp`, `fromRotationTranslationScale` и `fromRotationTranslationScaleOrigin`

- В инструмент запекания скелетной анимации был добавлен флаг `Bake only deform bones`.

Эта возможность полезна в случае, когда некоторые объекты являются потомками недеформированных “костей”. До этого запечь “кости” без выставленного флага `deform` было невозможно.

- Новый переключатель `GL Debug` в Просмотрщике.

В Просмотрщике сцен появился переключатель `GL Debug`, который может использоваться для отключения проверок GL-ошибок. Это увеличивает производительность и может быть полезно при профайлинге сцены.

- Поддержка WebGL 2.0 в браузере Firefox.

Появилась возможность использовать экспериментальный контекст WebGL 2.0 в браузере Firefox.

- Новая опция настроек движка: `prevent_caching`

Включает и выключает кэширование медиаданных.

- Добавлена новая физическая функция.

Функция `apply_force_world` применяет к объекту постоянную силу в мировом пространстве координат.

## Изменения

- Теперь опция Просмотрщика `HUD Info` выводит подробную информацию в каждом столбце.
- Изменения редактора логики.
  - В node `Conditional Jump` добавлена возможность выбора типа переменной между `Number` и `String`.
- Теперь активация VR-режима автоматически меняет тип камеры на `EYE` для лучшего восприятия сцены пользователем.

## Исправления

- Исправлено поведение якорей.
- Исправлена ошибка в работе кубических карт на некоторых старых GPU производства NVIDIA в браузере Firefox.
- Исправлен баг системы частиц с назначенной вершинной группой “Length” в случае, когда для объекта-эмиттера активирована опция “Apply Modifiers”.
- Исправлено поведение ноды `B4W_GLOW_OUTPUT` для прозрачных материалов.

- Исправлена карта травы для единственного плоского объекта ландшафта.
- Векторы рефракции для нодовых и стековых материалов теперь используют правильную нормаль вида.
- Исправлена интерфейсная панель God Rays в Blender.
- Исправлено обновление параметров неба в Просмотрщике.
- Исправлен некорректный рендеринг стереоизображения (анаглиф и HMD) в некоторых специфических случаях.
- Исправлен баг ноды `Page Param`, приводивший к тому, что результат всегда записывался в переменную `R1`.
- Исправлено обновление области видимости переменных в нодовом редакторе логики.
- Исправлен крэш динамически загружаемых сцен при использовании логической ноды `Move Camera`.

## v16.03

### Новый функционал

- Улучшения нодового редактора логики.

Появилась возможность использовать глобальные переменные для обмена информацией между потоками. Это позволяет создавать более сложные интерактивные приложения.

Добавлена логическая нода `JSON`. Она может использоваться для кодирования и декодирования сложных JSON-объектов.

Добавлена логическая нода `Get Timeline`. Она может использоваться для получения текущего кадра NLA или общего таймлайна.

Логические ноды `Play Animation` и `Stop Animation` теперь могут применяться для анимации настроек окружения.

- Новый параметр `Update Material Animation` для обновления анимированных нодовых материалов во вьюпорте.

Эта возможность полезна тем, кто часто использует анимированные ноды материалов. Достаточно включить её, и анимация материалов будет воспроизводиться во вьюпорте Blender.

- Менеджер проектов отделён от аддона Blender.

Начиная с этого релиза, менеджер проектов может быть запущен как отдельное приложение с помощью скрипта `project_server.py`. Теперь не требуется запускать Blender для того, чтобы запустить менеджер проектов.

- Добавлены заголовки и описания ко всем демо-приложениям Blend4Web.

Это упрощает поиск демо-приложений с помощью Google и других поисковых систем.

- Добавлен ряд новых модулей.

Первый из них - математический модуль `math`. Чтобы больше узнать о нём, обратитесь к [документации разработчика](#).

Кроме него, также добавлен модуль `input`. Этот модуль содержит средства для работы с устройствами ввода: мышью — `DEVICE_MOUSE`, клавиатурой — `DEVICE_KEYBOARD`, тачскринами — `DEVICE_TOUCH`, гироскопами — `DEVICE_GYRO`, устройствами виртуальной реальности — `DEVICE_HMD`. Узнать о нём больше можно в разделе документации разработчика, посвящённом модулю `input`.

- Изменён подход к обратной совместимости с предыдущими версиями Blender.

Начиная с этого релиза, мы постараемся обеспечить совместимость аддона с предыдущими версиями Blender.

- В описаниях объектов теперь поддерживаются HTML-ссылки
- Объекты, находящиеся на вспомогательных сценах (используемых для рендеринга в текстуру), теперь могут контролироваться логическими нодами.
- В интерфейс рендеринга была добавлена опция `Dynamic Grass`.

Для этого параметра доступны три значения: *ON* включает постоянное отображение динамически генерируемой травы, *OFF* полностью отключает его, *AUTO* позволяет движку автоматически определять объекты с динамической травой.

## Изменения

- Изменения API

Методы `enable_controls` и `disable_controls` аддона `app` были объявлены устаревшими.

Следующие методы модуля `controls` были объявлены устаревшими: `register_keyboard_events`, `register_mouse_events`, `register_wheel_events`, `register_touch_events`, `register_device_orientation`, `unregister_keyboard_events`, `unregister_mouse_events`, `unregister_wheel_events`, `unregister_touch_events`, `unregister_device_orientation`.

В модуль `controls` были добавлены методы `create_touch_click_sensor` и `create_hmd_quat_sensor`.

Методы `get_hmd_device` и `reset_device` модуля `hmd` были объявлены устаревшими.

Следующие методы были добавлены в модуль `util`: `deg_to_rad`, `rad_to_deg`, `quat_to_ordered_angles`.

В модуле `scenes` был добавлен метод `get_world_by_name`.

- Были внесены изменения в API для работы с камерой.

Метод `calc_ray()` был изменён. Теперь он может работать с параметрическими линиями.

- Изменено поведение физики типа `Static`.

Теперь объекты, которым присвоен материал со включённым параметром *Material -> Special: Collision*, могут быть полностью выгружены. Кроме того, такие объекты можно перемещать, вращать и пр. пока они остаются динамическими.

- Оптимизация рендеринга.

Теперь для отсечения статических объектов используются ограничивающие эллипсоиды, а не сферы.

- Изменения редактора логики.

Нода *Send Request* была упрощена. Кодирование и декодирование `JSON` вынесено в новую ноду `JSON`.

- Оптимизировано обновление анимированных материалов и материалов, освещаемых динамическими источниками света.
- Генерация меша воды теперь работает на устройствах без поддержки `OES_depth_texture`.

## Исправления

- Исправлено поведение меню веб-плеера.
- Исправлен крэш сервера разработки, возникавший в случае, когда заголовки ответа содержали определённую дату (29 февраля).
- Исправлен параметр `Refractions`, который имел значение “ON” при отсутствии объектов с рефракцией.
- Исправлено позиционирование материала воды для динамических объектов.
- Исправлен крэш, возникавший при создании меша воды без волн.
- Исправлена некорректная работа выхода ноды `TEXTURE` в случае, если не выбрана текстура.

## v16.02

### Новый функционал

- Расширение поддержки системы частиц типа Emitter.

Поддержка нодовых материалов, позволяющих задавать параметры шейдинга частиц с помощью последовательности базовых блоков (включая ноду *Particle Info*). Функционал доступен для систем частиц с типом рендеринга *Billboard*.

Переписан шейдер для частиц с типом рендеринга *Halo*. Добавлена поддержка настроек *Rings*, *Lines* и *Star Tips*. Убрано ограничение на максимальный размер частиц, связанный с аппаратными ограничениями на некоторых платформах.

Ликвидирован лимит на максимальное количество точек в процедурной текстуре (*Ramp*), используемой для подкраски частиц.

- Расширение поддержки нодовых материалов.

Реализованы две новые ноды, основанные на использовании кривых: *Vector Curves* и *RGB Curves*.

Добавлена поддержка ноды *ColorRamp*, в которой реализованы типы интерполяции *Linear* и *Constant*.

Добавлена поддержка ноды *Particle Info*. На данный момент её функционал поддерживается в полном объёме для систем частиц типа *Emitter* (с установленным типом рендеринга *Billboard*).

- Улучшения в менеджере проектов.

- Возможность экспортировать сразу нескольких проектов в один архив.

Данная функциональность существенно упрощает обновление SDK, а также перенос проектов с одной рабочей машины на другую.

- Новые возможности для развёртывания (deploy) проектов.

Развёртывание проектов необходимо для загрузки готовых приложений на сервер, отправки по почте и т.д. В этом релизе развёртывание проектов можно делать непосредственно из графического интерфейса Project Manager, проект при этом упаковывается в zip-архив.

Также начиная с этого релиза развёртывание доступно для всех без исключения типов проектов (включая External).

- Улучшения в проектах типа Web Player HTML и Web Player JSON.

При создании таких проектов появилась возможность задавать параметры приложения Web Player, такие как возможность отоб-

ражения счётчика кадров, автоматический поворот камеры, выключение кнопок социальных сетей и др.

Кроме того, проекты этих типов могут создаваться с опцией **bundle**, означающей, что все ресурсы приложения будут располагаться в одной и той же директории.

- Улучшения графического интерфейса.

Поддержка добавления и отображения иконок проектов, что упрощает навигацию и позволяет быстро оценить внешний вид разрабатываемых приложений.

Просмотр информации о проекте. При переходе по ссылке **[info]**, расположенной справа от имени проекта, выводится список с подробной информацией о разрабатываемом проекте.

Для удобства определения типа запускаемых приложений, справа от ссылки запуска отображается префикс: **player:** - приложение типа Web Player HTML или Web Player JSON, **dev:** - префикс для запуска разрабатываемой версии приложения, **build:** - префикс для запуска скомпилированной (обфусцированной) версии приложения.

Интерфейсные элементы менеджера проекта теперь снабжены всплывающими подсказками.

- Возможность анимирования настроек окружения.

Добавлена возможность анимации параметров окружения из разделов **Sky** (**Horizon Color**, **Zenith Color**), **Environment Lighting** (**Energy**) и **Mist** (**Minimum**, **Start**, **Depth**, **Height**, **Fog Color**). Данная функциональность поддерживается также и в NLA-анимации.

В панель настроек **World** в Blender добавлен раздел **Animation** с параметрами **Apply Default Animation** и **Behavior**.

Для всех экспортируемых миров на сценах создаются соответствующие метаобъекты типа **WORLD**, которые можно использовать для управления анимацией настроек окружения. Взаимодействие с данными метаобъектами аналогично взаимодействию со стандартными объектами сцены.

- Улучшения камеры.

Добавлена возможность назначать лимиты на вертикальное перемещение опорной точки TARGET-камеры в Blender. Функциональность также поддерживается через API с помощью методов **target\_set\_pivot\_limits()** и **target\_get\_pivot\_limits()**.

Добавлены новые методы **static\_setup()**, **eye\_setup()**, **target\_setup()**, **hover\_setup()** и **hover\_setup\_rel()** для смены и полного задания модели поведения камеры. Метод **set\_move\_style()** в то же время объявлен устаревшим.

Также были добавлены методы `target_switch_panning()` - для управления режимом панорамного перемещения, `get_view_vector()` - для получения вектора направления взгляда камеры.

Примеры работы с API камеры теперь описаны в *соответствующем разделе документации*, ссылка на который также доступна со страницы API документации модуля `camera.js`.

- Улучшения нодового редактора логики.

В node `Send Request` добавлена опция `Content-Type`, позволяющая переназначить поле заголовка HTTP-запроса.

- В модуль `controls` добавлен новый сенсор.

`Gyro Quat` (метод `create_gyro_quat_sensor()`) - сенсор для работы с гироскопом на мобильных устройствах, позволяет оперировать кватернионом поворота устройства.

- Добавлена экспериментальная поддержка UC Browser.
- Документация была значительно переработана и дополнена.

## Изменения

- Параметр ламп `Generate Shadows` переименован в `Shadow` и теперь также активирует отображение теней в Blender Viewport.
- Изменён дизайн главной страницы SDK.

Список приложений теперь находится в менеджере проектов. Добавлены ссылки для запуска часто используемых приложений: просмотричика (*Viewer*) и примеров с кодом (*Code Snippets*).

- Методы `get_coords_x()` и `get_coords_y()` теперь принимают параметр `target_touches`.

Данный параметр позволяет в случае мультитача использовать только те касания, которые произошли внутри текущего target-элемента (свойство `targetTouches`).

- Изменения API модуля `camera`.

Объявлены устаревшими методы `has_vertical_limits()` и `has_horizontal_limits()`. Вместо них следует использовать `has_vertical_rot_limits()` и `has_horizontal_rot_limits()`.

- Изменение правил именования файлов разрабатываемых приложений

Теперь html-файлы разрабатываемых приложений (расположенных в директории `apps_dev`) не содержат в своём имени суффикса `_dev`, отличать данные приложения от скомпилированных позволяет префикс, указываемый в *Project Manager*.

- При переходе на главную страницу SDK проверяется наличие локального сервера разработки. В случае запуска из локальной файловой системы выводится соответствующее предупреждение.

## Исправления

- Исправлена ошибка с неверным поведением параметра движка `canvas_resolution_factor` на устройствах с системой Apple iOS.
- Исправлена ошибка движка при отсутствии выбранного объекта в логических нодах `Show Object`, `Hide Object`.
- Исправлена ошибка округления координат по осям Y и Z при использовании переменных в качестве параметров в логической ноде `Transform Object`.
- Исправлена блокировка нодовой логики при наличии нод, не связанных с `Entry Point`.
- Исправлена ошибка отображения динамических объектов на неосновных сценах при рендеринге сцены в текстуру.
- Исправлена ошибка перехода в полноэкранный режим в Safari.
- Устранена ошибка интерфейса аддона, связанная с системой частиц типа `Emitter` при отсутствии на объекте материала.
- Исправлена работа входа `Factor` в ноде `B4W_GLOW_OUTPUT` на материале с прозрачностью типа `Alpha Clip`.
- Дупли группы с типом дупликации `None` не экспортируются и не отрисовываются.
- Исправлен метод `append_ray_test()` при сообщении ему пустого объекта в качестве первого параметра.
- Исправлены и оптимизированы некоторые настройки воды.
- Физические объекты, имеющие родительский объект, теперь корректно обновляют позицию в случае, если на сцене отключена физика.
- Исправлено поведение нод `Camera Data` и `B4W Vector View` для отраженных объектов.
- Исправлена NLA анимация ноды `RGB` в материалах.
- Система частиц типа `HAIR` больше не будет отображаться, если на объекте-эмиттере выставлен флаг `Do Not Render`.
- Исправлена работа метода `auto_rotate()` для случая EYE камеры.
- Исправлен экспорт частиц типа `Hair`, находящихся на неактивной сцене в Blender.
- Добавлены runtime библиотеки для Windows, без которых возникала ошибка при конвертации ресурсов.

- Исправлена ошибка генерации шейдера нодового материала в случае наличия в материале более 10 текстур.

## Известные проблемы

Начиная с этого релиза описание всех известных проблем и возможных решений находится в *отдельном подразделе*.

# v16.01

## Новый функционал

- Рендеринг линий.

Теперь поддерживается возможность отрисовки линий, задаваемых процедурным образом. Для этого предусмотрен специальный тип объекта, активируемый с помощью опции *Line Rendererer* в Blender, а также ряд API `draw_line`, `get_line_params` и `set_line_params`.

- Новые ноды редактора логики.

- **Transform Object**: Позволяет перемещать объект в мировой, родительской или локальной системе координат.
- **String Operation**: Позволяет проводить операции над строковыми константами и переменными аналогично ноде **Math Operation**.

- Упрощена установка SDK.

Теперь при *установке SDK* в меню *User Preferences* в разделе *File->Scripts* достаточно указать путь к SDK.

- Улучшения в приложении WebPlayer.

Добавлена возможность отключать социальные кнопки, отображаемые в правой части загруженной сцены. Для этого необходимо *передать атрибут no\_social* при запуске приложения.

Экранированные символы в адресной строке плеера теперь обрабатываются корректно.

- Улучшение качества отрисовки для шлемов виртуальной реальности (HMD).

Для шлемов виртуальной реальности добавлен механизм коррекции дисторсии и устранения хроматической aberrации. Для установки пользовательских настроек коррекции следует использовать функцию `set_hmd_params()`.

- В модуль `controls` добавлены два новых сенсора.

*Touch Rotate* (метод `create_touch_rotate_sensor()`) - сенсор, обрабатывающий событие поворота с помощью двух пальцев на touch-устройствах.

*Callback* (метод `create_callback_sensor()`) - сенсор, значение которого определяется callback-функцией, вызываемой каждый кадр.

- Улучшения в утилите для конвертации ресурсов.

В *утилите для конвертации ресурсов converter.py* добавлены параметры `-verbose` и `-jobs` служащие соответственно для вывода детализированной информации и задания максимального количества параллельно выполняемых процессов в процессе конвертации.

- Добавлен ряд методов API.

`is_line()` - проверка, является ли объект специальным объектом типа `LINE`.

`is_idle()` - проверка, завершил ли загрузчик сцен все запланированные действия.

`hover_switch_horiz_rotation()` - возможность включать/выключать горизонтальное вращение камеры типа `HOVER`.

- Оптимизирован алгоритм обновления нодового дерева.

## Изменения

- Упрощено применение Кубических отражений.

Теперь кубическое отражение рендерится относительно геометрического центра объекта, а не относительно его центра в Blender. Так же в кубическом отражении больше не отрисовываются плоские отражения, что раньше приводило к нежелательным артефактам.

- Изменения API.

В модуль `camera` добавлены следующие методы: `get_vertical_axis()`, `set_vertical_axis()`. Теперь вращение на угол “ $\phi$ ” камеры типа “EYE” методом `eye_rotate()` происходит вокруг вектора, выдаваемого функцией `get_vertical_axis()`.

Метод `correct_up()` теперь принимает в качестве параметра флаг `strict`. Этот параметр позволяет ориентировать камеру сонаправленно с передаваемым вектором `y_axis`, а не просто параллельно ему.

Методы `eye_get_vertical_limits`, `eye_get_horizontal_limits`, `target_get_vertical_limits` и `target_get_horizontal_limits` теперь могут возвращать лимиты, заданные в мировых и в локальных координатах относительно камеры. Это определяется параметром `local`.

- Изменения в редакторе логики.

Добавлено распределение нод по категориям.

- В node **Variable Store** добавлена возможность выбора типа переменной между **Number** и **String**.
  - В node **Send Request** добавлена возможность использовать строковые переменные для хранения тела запроса и ответа сервера целиком.
- Удаление устаревшего инструмента *Mass Reexporter*.

Панель инструмента *Mass Reexporter* удалена, поскольку функционал автоматического реэкспорта сцен уже имеется в инструменте *Project Manager* (операция `re-export scenes`).

- Устаревший функционал.

Были удалены методы: `mouse.enable_mouse_hover_glow()`, `mouse.disable_mouse_hover_glow()`, `anim.get_actions()`, `anim.get_current_action()`, `anim.set_current_frame_float()`, `anim.get_current_frame_float()`, `anim.get_frame_range()`, `anim.cyclic()`, `anim.is_cyclic()`, `anim.update_object_animation()`, `controls.remove_sensor_manifolds()`, `main.redraw()`, `scenes.set_glow_intensity()`, `scenes.get_glow_intensity()`, `scenes.apply_glow_anim()`, `scenes.apply_glow_anim_def()`, `scenes.clear_glow_anim()`, `scenes.set_glow_color()`, `scenes.get_glow_color()`, `sfx.speaker_play()`, `sfx.speaker_stop()`, `sfx.speaker_playback_rate()`, `sfx.get_speakers()`, `trans.set_rotation_quat()`, `trans.set_rotation_quat_v()`, `trans.get_rotation_quat()`.

Объявлены устаревшими метод `set_camera_move_style()` и константы `SPACE_WORLD` и `SPACE_LOCAL`.

Также объявлены устаревшими методы `has_vertical_limits()` и `has_horizontal_limits()`. Вместо них следует использовать наиболее подходящий из новых методов: `has_vertical_rot_limits()`, `has_horizontal_rot_limits()`, `has_vertical_trans_limits()`, `has_horizontal_trans_limits()`.

## Исправления

- Исправлено позиционирование и мерцание для систем частиц с мировыми координатами.
- Исправлен Front Facing для кубических отражений.
- Исправлены ошибки рендеринга для стерео-режимов.
- Исправлено некорректное позиционирование в анимации дочерних объектов.
- Исправлено зависание загрузки сцены в Safari при использовании аудио файлов формата .ogg в источниках звука типа **Background Music**.

- Исправлен баг ноды `Move Camera` из-за которого не оказывало влияние свойство `duration` после переоткрытия blend файла.
- Исправлен крэш при экспорте систем частиц с эмиттером со сложным мешем.
- Исправлена ошибка ноды `Switch Select` в результате которой осуществлялся неверный переход.
- Исправлен ряд ошибок при отображении лимитов камеры во видпорте Blender.
- Исправлена ошибка экспорта материалов, привязанных непосредственно к объекту, а не к мешу.
- Исправлен вызов функции `RenderCallback`, задаваемой методом `set_render_callback()`.

Теперь `RenderCallback` будет выполняться прямо перед рендерингом текущего кадра, поэтому в момент вызова состояние сцены и объектов будет актуальным.

- Исправлен рендеринг теней от нескольких источников света разных типов (например, `POINT` и `SUM`).
- Исправлен рендеринг `Environment` текстур при их одновременном использовании в качестве текстуры мира и в стековом материале.

## 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.
- На системах Apple macOS 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>. В Blender 2.76b эта проблема устранена, поэтому рекомендуем обновиться.
- При использовании драйверов Nouveau скелетная анимация может работать некорректно.

## v15.12

### Новый функционал

- Поддержка теней от нескольких источников.

Осуществлена поддержка теней от нескольких источников освещения. Функционал позволяет существенно улучшить реалистичность сцен, освещённых множеством источников. Подробнее про функционал и сопутствующие ограничения можно прочитать в [документации](#).

- *Экспериментальная поддержка шлемов виртуальной реальности (HMD).*

Добавлена экспериментальная поддержка шлемов виртуальной реальности с использованием WebVR API. На данный момент для использования этой технологии необходимо наличие браузера с поддержкой WebVR и устройства Oculus Rift. Подробнее в соответствующем разделе документации.

- Привязка объектов к камере в пространстве экрана.

Новый инструмент Viewport Alignment для привязки объектов к камере в пространстве экрана. Этот инструмент позволяет в удобной форме создавать элементы пользовательского интерфейса, привязанные к краю, углу или центру экрана. В отличие от привязки родитель-потомок, положение объекта автоматически изменяется при изменении разрешения и пропорций экрана.

Также был расширен функционал метода `append_stiff_viewport()` для назначения ограничителя Stiff Viewport.

- Отображение лимитов, назначаемых на камере во виджете Blender.

Для упрощения задания ограничений добавлена опция `Display limits in viewport` на панели Data объекта-камеры. При её активации, лимиты будут отображаться непосредственно во виджете Blender. Также были изменены их значения, выставляемые по умолчанию.

- Улучшение рендера аналифного стереоизображения.

Был реализован новый алгоритм рендера для режима аналиф с улучшенным цветом изображения. Также плоскость сходимости левого и правого изображения для камеры типа TARGET теперь считывается автоматически на основе опорной точки камеры.

- Новые ноды редактора логики.

- `Move To`: Позволяет перемещать объект в направлении другого объекта.
- `Console Print`: Позволяет выводить полезную информацию (в том числе значения переменных) в консоль браузера.

- В веб-плеер добавлена кнопка переключения в стерео-режим.

Позволяет переключаться в режим HMD (при наличии поддержки со стороны браузера), либо в режим аналифного стереоизображения.

- Добавлены новые методы для работы с объектами.

Были добавлены новые методы `get_selectable_objects` и `get_outlining_objects`, облегчающие работу с объектами.

- Добавлена возможность удалять динамические объекты.

Раньше можно было удалять только те объекты, которые были скопированы пользователем.

- Добавлена возможность взятия объекта аннотации по координатам на канвасе.

Теперь функция `pick_object` возвращает объект-аннотацию, если он располагается в указанных координатах.

- Добавлен новый метод для инициализации медиаресурсов на мобильных устройствах.

Раньше для обхода ограничений мобильных браузеров по инициализации медиаресурсов (аудио и видео) применялось принудительное блокирование канваса с необходимостью нажатия на всплывающую пиктограмму начала воспроизведения. Теперь можно самостоятельно инициализировать медиаресурсы, используя метод `activate_media`. Данный функционал может быть полезен, в частности, если мобильное приложение уже включает элементы, требующие нажатия пользователя (кнопка “Старт”, кнопка включения звука и т.д.).

- Новый тип множества сенсоров (Sensor Manifold) - `CT_POSITIVE`.

Новый тип множества `CT_POSITIVE` предполагает вызов обработчика событий при ненулевом значении логической функции.

## Изменения

- Кнопка `Fast Preview` для быстрого предпросмотра сцены продублирована на нижней панели интерфейса.

Позволяет осуществить просмотр сцены без необходимости совершать переход к панели настроек сцены `Development Server`.

- Было изменено API для работы с канвас-текстурой.

Теперь работать с канвас-текстурой стало удобнее. Были добавлены новые методы `get_canvas_ctx` и `update_canvas_ctx`. Методы `get_canvas_texture_context` и `update_canvas_texture_context` признаны устаревшими и не рекомендуются к использованию. Также для канвас-текстуры больше не присутствует в интерфейсе поле `Source ID`.

- Исправлено неверное поведение теней для Alpha Clip материалов.

Было исправлено неверное поведение теней для Alpha Clip материалов при невыставленном значении прозрачности на материале.

## Исправления

- Исправлено неверное поведение физики скопированного объекта.

Было исправлено поведение физики скопированного объекта при перемещении его до добавления на сцену.
- Исправлено некорректное поведение тумана при добавлении на сцену воды
- Исправлено некорректное расположение интерфейсных панелей Blender в стандартной сцене, создаваемой для новых проектов.
- Исправлена ошибка при рендеринге кубмар-текстур на видеокартах NVIDIA GeForce 200 Series.
- Улучшена работа движка на iPhone 4, 4S, 5; iPad 2,3,4 поколений; iPad Mini 1 и 2 поколений.
- Исправлен некорректный рендеринг теней для billboard-объектов.

## 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 macOS 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>. В Blender 2.76b эта проблема устранена, поэтому рекомендуем обновиться.

## v15.11

### Новый функционал

- Начальная поддержка технологии WebGL 2.0.

Возможности новых API предоставляются автоматически при наличии совместимого оборудования/браузера.

- Новые возможности редактора логики.

- Возможность анимации камеры.

В ноду `Move Camera` добавлена возможность плавной смены положения камеры.

- Возможность задания поведения объектной анимации.

В ноду `Play Animation` добавлена возможность задания *Поведения (Behavior)* анимации.

- Новая нода `Stop Animation`.

Позволяет останавливать объектную анимацию.

- Новая нода `Stop Sound`.

Позволяет останавливать воспроизведение звука спикера.

- Новая нода `Stop Timeline`.

Позволяет останавливать NLA анимацию.

- Поддержка антиалиасинга MSAA.

Данный алгоритм сглаживания применяется на конфигурациях, совместимых с WebGL 2.0. На данный момент он позволяет получить максимальное качество изображения в режиме ULTRA, благодаря действованию алгоритма 16x MSAA.

- Улучшения в интерфейсе программы-просмотрщика.

Добавлена кнопка `Home`, при нажатии на которую осуществляется переход на базовую сцену, указанную в URL-параметрах приложения, либо, при её отсутствии, на стандартную сцену с кубическим лого Blend4Web.

Кнопка `Reset` вынесена в правую часть верхней панели приложения и теперь, кроме внутренних параметров, также сбрасывает и URL-параметры просмотрщика (например, загружаемую базовую сцену).

Кнопка `Auto View Mode` для автоматического прохода всех сцен вынесена на панель `Tools & Debug`.

- Высокоточный механизм оценки производительности рендеринга.

Использование расширения WebGL EXT\_disjoint\_timer\_query позволяет реализовать высокоточные таймеры, способные оценить производительность операций, выполняемых на GPU. При активации режима отладки в программе-просмотрщике (опция **HUD Info** на панели **Tools & Debug**) в последней колонке отладочной информации выводится время в миллисекундах, потраченное GPU на рендеринг части сцены (`subscene`).

- Улучшения в интерфейсе веб-плеера.

Добавлено изменение вида курсора при перемещении камеры.

- Многочисленные добавления и улучшения в документации.

## Изменения

- Кнопка **Copy Cursor Location** переименована в **Look At Cursor**, изменено её поведение.

Теперь при нажатии на данную кнопку камера поворачивается в направлении указанной точки.

- Изменения API.

Для совместимости со сторонними приложениями добавлены методы `set_matrix`, `get_matrix`, `set_matrix_rel`, `get_matrix_rel`, однако, в целях оптимизации рекомендуется использовать `set_tsr`, `get_tsr`, `set_tsr_rel`, `get_tsr_rel`.

Теперь функции `hide_object` и `show_object` по умолчанию изменяют видимость объекта и всех его дочерних объектов. Для изменения видимости конкретного объекта необходимо передать параметр `ignore_children` со значением `true`.

Следующий метод объявлен устаревшим и будет удален в последующих релизах: `create_sep()` (модуль `tsr`). Вместо него теперь следует использовать метод модуля `tsr`: `set_sep()`.

API модуля `camera` претерпело существенные изменения. Были удалены устаревшие методы: `is_camera()`, `rotate_pivot()`, `rotate_hover_cam()`, `get_hover_cam_angle()`, `set_hover_cam_angle()`, `translate_hover_cam_v()`, `set_eye_params()`, `rotate()`, `get_angles()`.

Следующие методы объявлены устаревшими: `set_look_at`, `rotate_eye_camera`, `rotate_target_camera`, `set_trans_pivot`, `zoom_object`, `set_pivot`, `rotate_hover_camera`, `get_hover_cam_pivot`, `get_eye`, `get_pivot`, `hover_cam_set_translation`, `set_hover_pivot`, `get_hover_angle_limits`, `get_cam_dist_limits`, `apply_vertical_limits`, `clear_hover_angle_limits`, `apply_hover_angle_limits`, `apply_distance_limits`, `clear_distance_limits`, `get_vertical_limits`, `apply_horizontal_limits`, `get_horizontal_limits`, `clear_vertical_limits`, `clear_horizontal_limits`. Вместо них следует использовать методы, начинающиеся с префикса, соответствующего типу камеры:

`target_...`, `eye_...`, `static_...`, `hover_...`. Также устаревшими объявлены методы `set_velocity_params` и `get_velocity_params`; вместо них следует использовать `set_velocities` и `get_velocities` соответственно.

Помимо этого были добавлены методы `target_get_distance`, `target_set_distance` и `hover_get_distance` для работы с расстоянием от текущего положения TARGET/HOVER камеры до pivot-точки.

- В приложении *Viewer* добавлены кнопки *Play All* и *Stop All* для запуска и остановки всей анимации.
- Изменения редактора логики:
  - Нода *Select* и базирующиеся на ней ноды *Select & Play Animation* и *Select & Play Timeline* теперь являются устаревшими. В качестве замены рекомендуется использовать комбинации нод *Switch Select*, *Play Animation* и *Play Timeline*.
  - Обновлены примеры в SDK, использующие вышеуказанные устаревшие ноды.
  - Для прозрачности поведения нодовой логики воздействие опции *Cyclic NLA* отключено.
- Изменено поведение HOVER-камеры при отсутствии у неё лимитов.
  - Теперь HOVER-камера будет зависать над pivot-точкой на фиксированном расстоянии и под фиксированным углом.
- Изменение поведения *glow* эффекта.

Теперь источники освещения влияют на *glow* эффект при наличии нод *Material*, *Extended Material*.

## Исправления

- Исправлены ошибки в системе управления проектами.

В списке теперь выводятся ссылки на девелоперские приложения в составе проекта.

В html-файлах скомпилированных приложений исправлена верстка.

- Улучшена работа скининга на мобильных платформах.
- Исправлена ошибка подключения файлов переводчика аддона.

Исправлена ошибка, возникающая при подключении аддона на некоторых устройствах, связанная с невозможностью обработки системой файлов переводчика аддона.

- Исправлено некорректное поведение теней от объектов системы частиц.

Было исправлено поведение теней от объектов системы частиц типа “Hair”, используемых в качестве билбордов, при изменении размеров основного канваса.

- Исправлена работа движка на GPU Mali 400 серии.
- Исправлены плоские отражения и туман для материалов, использующих `double_sided_lighting`
- Исправлена ошибка, возникавшая при ограниченной квоте в Local Storage. Проявлялось на Safari в режиме инкогнито.
- Восстановлена функциональность опции `Render Glow Over Transparent Objects` в [настройках](#) эффекта Glow в Blender.
- Исправлен ряд ошибок воспроизведения видеотекстур.

## 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 macOS 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>.

## v15.10

### Новый функционал

- Новый функционал в системе управления проектами.

Проекты теперь можно экспортировать 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.

- Улучшения нодового 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 НПОТ-текстур.

Теперь доступен тип маппинга **Repeat** для НПОТ-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.

## 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`.

## 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*.

## 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 macOS 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>.

## v15.09

### 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 интерфейсе веб-плеера.

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()`.

## 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*.

- Улучшения 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 дуплицирования 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`.

- Изменено поведение 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.

## 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 `Orco` 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.

## 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.

## v15.08

### 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 смешивание 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 кроссбраузерные 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, сфера, 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.

## 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.

## 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 touchскрина на некоторых устройствах в браузере Microsoft Edge.

## Известные проблемы

- В редакторе логики при зацикливании под часть соединений подкрашиваются красным цветом. Данный дефект никак не сказывается на работе инструмента и может быть проигнорирован.

## v15.07

### 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 слаживают 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*.

Устранены 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 плоскости, в которой лежит точка вращения камеры.

- Новый тип множества сенсоров (*Sensor Manifold*) - **CT\_CHANGE**

Наряду с типами **CT\_CONTINUOUS**, **CT\_TRIGGER**, **CT\_SHOT** и **CT\_LEVEL** появился новый тип множества **CT\_CHANGE**, который предполагает вызов 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 движка. 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 вертикальных лимитов поворота и перемещения.

- Улучшения в 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.

## 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 экспортованных 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.

## 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 вертекской 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.

## 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.
- 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*.

## v15.06

### 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, расположенном 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.

## 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'а для нодового материала при отсутствии ноды *Output*.

## 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 экспортёра.

- Исправлена ошибка удаления/обновления аддона на Windows.

Переработана система загрузки бинарной библиотеки.

- Исправлен ошибка 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 Bindings*.

- Исправлен баг рендеринга billboard-объектов на iPad.

## 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.
- 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.

## v15.05

### 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 вы-

ставлять 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()`. Реализованный 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`.

## 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 вертексной анимации.

При наличии у объекта арматурного модификатора и вертексной анимации одновременно, арматурный модификатор экспортоваться 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.

## 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 обрабатываемых 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 подовых материалов, используемая на нескольких объектах.

- Исправлен баг эффекта Motion Blur.

## Известные проблемы

- Проблемы с обновлением старых версий аддона.

Рекомендуется перезагрузка Blender после обновления аддона.

## v15.04

### 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.

- Оптимизации 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`.

## 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 простоянки 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 камеры типа `"EYE"`.
- Добавлена возможность запуска локального сервера разработки непосредственно из 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 *Ouline 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.

## 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.

## v15.03

### 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 под 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.

## 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 экспортированных 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.

## 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 автоворотения 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 мешем 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 рендера тумана в отдельных версиях *Chrome* и *Firefox* под *Windows*.

## Известные проблемы

- Проблемы с обновлением старых версий аддона.  
Рекомендуется перезагрузка Blender после обновления аддона.
- Не работают видеотекстуры в *Firefox* при экспорте сцены в формате *HTML*.

## v15.02

### 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 просмотра 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, стлаживание.

- 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 macOS.*

В macOS теперь доступны все функции SDK, включая сборку движка и приложений, конвертацию ресурсов и генерирование документации.

- *В модуль камера 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.

## 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.

## 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 имен 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.

## Известные проблемы

- Проблемы с обновлением старых версий аддона.

Рекомендуется перезагрузка Blender после обновления аддона.

- Не работают видеотекстуры в Firefox при экспорте сцены в формате HTML.

## v15.01

### 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 слов 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.

## 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 автоворота 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 скрытию 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.

## 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”.

## Известные проблемы

- Проблемы с обновлением старых версий аддона.

Рекомендуется перезагрузка Blender после обновления аддона.

- 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.

## v14.12

### 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-textur*.

- 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 экспортить готовое приложение для последующего развертывания на сервере.

## 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 нодовом материале больше не требует наличия нод Material или Extended Material.*

В некоторых случаях (например, рефракция) карты нормалей могут применяться в материалах без освещения.

## 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 нодовых материалов.*

Ошибка возникала при использовании нод 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.*

## Известные проблемы

- Проблемы с обновлением старых версий аддона.

Рекомендуется перезагрузка Blender после обновления аддона.
- *Не работают видеотекстуры в Firefox при экспорте сцены в формате HTML.*

## v14.11

### 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'e.

- *Новый режим управления камерой - Hover.*

Реализован режим скольжения камеры над плоскостью (**Hover**), включающий 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.*

## 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 сенсоре “mouse\_down” добавлен код нажатой кнопки мыши, который доступен в поле payload сенсора.
- Значительно оптимизирован экспорт систем частиц типа Hair.

## 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 движку 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”.

## Известные проблемы

- Проблемы с обновлением старых версий аддона.

Рекомендуется перезагрузка Blender после обновления аддона.

## v14.10

### 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 можно проигрывать параллельно.

- *Поддержка билбординга объектов.*

Объектам добавлены [опции](#), позволяющие настроить билбординг.

- *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 экспорта 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 частиц.*
- *Улучшена стабильность работы приложений на устройствах с мобильной операционной системой Apple iOS.*

## 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](#).

## 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 запечатания при помощи инструмента “B4W Vertex Anim Baker”.*

## Известные проблемы

- Проблемы с обновлением старых версий аддона.

Рекомендуется перезагрузка Blender после обновления аддона.

- *Карты нормалей не работают с типом текстурных координат Generated.*

При использовании карт нормалей необходимо использовать UV-развертку.

## v14.09

### 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 множества. 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.*

## 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](#).

- Настстройка качества графики в веб-плеере теперь сохраняется для каждой сцены независимо.

- Изменено поведение настроек конфигурации приложения: `physics_uraniun_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.

## 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 новом 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.

## Известные проблемы

- Проблемы с обновлением старых версий аддона.

Рекомендуется перезагрузка Blender после обновления аддона.

- *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

Аддон

ошибки инициализации, 85

Коррекция цвета, 370

Кватернионы, 141

Сумеречные лучи, 286

Управление перемещением объектов, 137

Выбор объектов, 138

аддон, 74

анаглиф, 291

анимация, 335

береговая линия, 357

браузер, 4

браузерные технологии, 4

цвет фона, 278

диффузная текстура, 298

драйверы, 5

движок, 1

экспорт, 24

ошибки, 88

предупреждения об ошибках экспорта, 90

глубина резкости камеры (depth of field), 282

графический движок, 1

интерактивная графика, 5

источники света, 267

компиляция шейдеров, 430

Расширения WebGL, 434

директивы import/export, 432

обfuscation, 431

оптимизация, 431

ошибки, 434

рекомендации и ограничения, 433

валидация, 431

материалы, 186

Параметры освещения, 187

гало, 197

ноды, 211, 213, 214

опции рендеринга, 194, 195

прозрачность, 189

специальные параметры, 196

зеркальное отражение, 191

освещение, 265

подсветка контура (outline glow), 288

просмотрщик, 52

добавление сцен, 24

прозрачность, 189

Настройка, 191

типы, 190

размытие при движении (motion blur), 281

рендеринг в текстуру, 310

сцена, 98

сенсор, 416

множество, 416

сглаживание, 290

система частиц

флюиды, 313

инстансинг, 323

сумеречные лучи (crepuscular rays), 286

текстура неба (skydome), 308

текстуры, 295

диффузная, 298

карта бликов, 299

карта нормалей, 299

карта окружения, 305

карта смешивания, 301

карта высот, 300

карта зеркального отражения, 307

настройки, 297

небо, 308

рендеринг в, 310

- типы, 296  
трехмерный движок, 1  
веб-плеер, 67  
версия  
    ошибки, 86  
видео-карта, 5  
возможности движка, 6  
взаимное затенение (screen-space ambient occlusion), 284  
замечания к релизам, 460  
затенение, 266  
    типы, 266  
зеркальное отражение, 191  
    динамическое, 192  
    эффект Френеля, 193  
    static, 192  
3D моделирование, 3
- B**
- batch, 56  
Blend4Web, 1  
Blender, 3  
boundings, 56
- C**
- cluster, 56
- D**
- DOF, 282
- G**
- git, 437  
    добавление и удаление файлов, 441  
    индивидуальные настройки, 439  
    коммит, 442  
    подготовка к коммиту, 440  
    проверка статуса, 439  
    разрешение конфликтов, 445  
    синхронизация между репозиториями, 443  
    тэги, 448
- M**
- matcap, 298  
material capture, 298
- N**
- normals, 153  
    editor, 153