
Blend4Web. User Manual

发布 v17.04

Triumph LLC

2017 年 04 月 26 日

1 概述	1
1.1 什么是 Blend4Web	1
1.2 “关于引擎”	1
1.3 图形引擎, 游戏引擎	2
1.4 什么是 WebGL	2
1.5 WebGL 的浏览器支持	2
1.6 WebGL 的优势	3
1.7 什么是 Blender	3
1.8 三维建模	4
1.9 浏览器技术	5
1.10 交互式图形	5
1.11 显卡和驱动程序	6
2 引擎功能	7
3 安装和更新	15
3.1 安装顺序	15
3.2 更新	19
4 工作流程	23
4.1 创建一个新项目	24
4.2 创建场景	24
4.3 准备一个在 Blend4Web 中使用的场景	25
4.4 导出场景	25
4.5 应用程序开发	26
4.6 构建项目	27

4.7	项目部署	27
5	项目管理	28
5.1	项目管理器	30
5.2	高级项目管理	46
6	场景预览器	56
6.1	导航	58
6.2	侧面板	58
6.3	指标	70
7	网络播放器	72
7.1	用法	74
7.2	导航	74
7.3	控制面板	75
7.4	属性	76
7.5	场景名称为标题	77
7.6	场景错误	77
8	插件	79
8.1	本地开发服务器	79
8.2	运行浏览器和演示	82
8.3	输出格式	83
8.4	导出选项	85
8.5	初始化错误	89
8.6	兼容性错误	90
8.7	关键导出错误	91
8.8	非致命导出的错误	93
8.9	其他消息	99
8.10	插件翻译	100
9	场景设置	103
9.1	渲染面板	105
9.2	场景面板	117
9.3	世界环境面板	125
10	物体	131
10.1	类型	132
10.2	静态和动态物体	133

10.3	设置	134
10.4	锚设置	141
10.5	物体变换 API	143
10.6	获取物体 API	144
10.7	物体选择	145
10.8	复制物体 (实例化)	146
10.9	四元数	147
10.10	通过 TSR 矢量移动	150
10.11	线渲染	150
10.12	Levels Of Detail	152
11	网格	157
11.1	静态和动态网格	157
11.2	设置	158
11.3	变形	159
12	法线编辑器	161
12.1	法线编辑器主要功能	164
12.2	界面	164
12.3	启用	164
12.4	显示法线	165
12.5	旋转	165
12.6	绝对和偏移模式	166
12.7	分割法线	167
12.8	平均	168
12.9	恢复	169
12.10	Tree (树), Foliage (树叶) 和 Face (面)	170
12.11	Copy (复制) /Paste (粘贴)	172
12.12	从网格复制	173
13	相机	176
13.1	移动方式和一般设置	176
13.2	限制摄像机的运动	178
13.3	视口对齐	186
13.4	相机控制 API	187
14	材质	194
14.1	灯光参数	195
14.2	透明	197

14.3 反射	199
14.4 渲染属性	202
14.5 视图属性	203
14.6 引擎特定参数	203
14.7 光晕材质	204
14.8 材质 API	205
14.9 材质继承	209
15 材质库	211
15.1 库结构	212
15.2 材质结构	214
15.3 添加材质到一个场景	218
16 节点材质	221
16.1 标准节点	222
16.2 游戏引擎特定节点	223
16.3 Cycles 节点	236
17 逻辑编辑器	237
17.1 基础	237
17.2 控制流	240
17.3 动画	246
17.4 相机 (Camera)	250
17.5 物体	252
17.6 操作	261
17.7 声音	265
17.8 网络	267
17.9 调试	273
17.10 已过时	274
17.11 布局	275
17.12 调试中	277
18 照明、阴影和背景	278
18.1 着色类型	278
18.2 用光源照明	280
18.3 环境照明 (环境)	281
18.4 阴影	284
18.5 Background	290

19 后期处理效果	293
19.1 运动模糊	293
19.2 景深	294
19.3 屏幕空间环境光遮蔽	296
19.4 上帝光芒	298
19.5 泛光效果	299
19.6 物体外轮廓	300
19.7 晕光	301
19.8 抗锯齿	302
20 立体渲染	304
20.1 激活	306
20.2 附加设置	306
20.3 HMD 设置	307
21 纹理	309
21.1 纹理类型	310
21.2 通用设置	311
21.3 漫射贴图	311
21.4 高光贴图	312
21.5 法线贴图	312
21.6 高度图。视差贴图	313
21.7 蒙版贴图	314
21.8 视频贴图	316
21.9 环境贴图	318
21.10 镜面贴图	320
21.11 天穹	321
21.12 特殊的纹理类型	323
22 粒子系统, 流体	328
22.1 用法	329
22.2 设置	330
22.3 粒子系统中的纹理	336
23 粒子系统, 实例化	340
23.1 粒子系统设置	342
23.2 草地	347
23.3 树叶	349

24 动画	353
24.1 动画控制	354
24.2 物体动画	354
24.3 蒙皮和骨骼动画	356
24.4 顶点动画	358
24.5 默认动画	360
24.6 非线性动画	360
24.7 音频源参数	362
24.8 值和 RGB 节点的动画	363
25 室外效果	365
25.1 水	367
25.2 大气	377
25.3 镜头光晕	380
25.4 风	382
26 色彩管理	386
26.1 Gamma 概述	386
26.2 人类视觉和显示器	387
26.3 伽玛公式	388
26.4 伽马在节点材质中	388
26.5 alpha 合成	389
26.6 色彩校正	391
27 音频	392
27.1 音源设置	393
27.2 调音台	397
27.3 处理和解码	398
28 物理	400
28.1 使用准备	401
28.2 静态物理类型	401
28.3 动态物理类型	403
28.4 约束	405
28.5 轮式的车辆	406
28.6 浮动物体	408
28.7 浮动交通工具或叫船舶	409
28.8 角色	411
28.9 导航网格	416

28.10 在应用程序中的使用	419
29 对应用程序开发人员	420
29.1 Application Development	421
29.2 背景透明	428
29.3 资源转换	431
29.4 GZIP Compression	438
29.5 示例代码	439
29.6 加载应用程序资产	440
29.7 事件驱动模型	440
29.8 SDK 文件结构	443
29.9 加载本地资源	446
29.10 品质配置	446
29.11 非标准画布位置和方向	448
29.12 移动 Web 应用程序	449
30 对于引擎开发者	451
30.1 代码样式	452
30.2 构建引擎	453
30.3 构建插件	454
30.4 依赖	454
30.5 命名函数和变量	455
30.6 调试中	456
30.7 着色器编译	456
30.8 更新插件翻译	462
31 团队工作。使用 Git	464
31.1 概述	465
31.2 典型的工作流	466
31.3 单独设置	466
31.4 检查状态	466
31.5 在提交前	467
31.6 准备提交	469
31.7 提交	470
31.8 存储库之间同步	471
31.9 解决冲突	473
31.10 标签	476
31.11 其他有用的命令	477

32 存在的问题及解决方案	478
32.1 WebGL 的支持	478
32.2 启动时的问题	479
32.3 WebGL 的无法初始化	481
32.4 更深入的故障排除	486
32.5 已知问题	486
33 发布说明	489
33.1 v17.04	489
33.2 v17.02	493
33.3 v16.12	498
33.4 v16.11	500
33.5 v16.10	504
33.6 v16.09	507
33.7 v16.08	509
33.8 v16.07	512
33.9 v16.06	514
33.10 v16.05	518
33.11 v16.04	522
33.12 v16.03	525
33.13 v16.02	528
33.14 v16.01	533
33.15 v15.12	537
33.16 v15.11	541
33.17 v15.10	545
33.18 v15.09	549
33.19 v15.08	555
33.20 v15.07	558
33.21 v15.06	563
33.22 v15.05	568
33.23 v15.04	572
33.24 v15.03	577
33.25 v15.02	582
33.26 v15.01	585
33.27 v14.12	589
33.28 v14.11	592
33.29 v14.10	594

33.30 v14.09	597
索引	601

概述

什么是 Blend4Web

Blend4Web 是一个面向 Web 的 3D 引擎 - 是一套在浏览器中进行制作交互式三维渲染图形和声音的软件框架。

该平台是用于可视化，演示，在线商店，游戏等丰富的互联网应用。

三维建模和动画工具（因此得名） - Blend4Web 框架与 Blender 紧密集成。内容是由 WebGL 的和其他的浏览器技术渲染而成，**不使用插件**。

技术上 Blend4Web 是网页库，Blender 插件以及进行调试和优化的一些工具。

自 2010 年以来凯旋有限责任公司 (Triumph LLC) 的员工开发了 Blend4Web 3D 引擎。该引擎首次发布于 2014 年 3 月 28 日。

“关于引擎”

引擎是被用于实现所需的功能使用的外部应用程序的软件代码的单独软件部分。

引擎的例子是：网站引擎，博客引擎，网上商店引擎，维基引擎，搜索引擎，游戏引擎等。软件引擎存在的经济原因是能多处使用相同的功能。例如开发人员可以用相对便宜的引擎来创建网上商店或游戏。

图形引擎，游戏引擎

图形引擎执行的显示图形特殊功能。它是介于以下几者之间的：

- 高级别应用部分（游戏逻辑，业务逻辑）和
- 低层系统的一部分（例如，图形库的WebGL 和drivers）.

图形引擎可以结合到音响系统，物理引擎，人工智能系统，网络系统和场景和逻辑编辑然后产生三维引擎--从而成为用于创作 3D 应用的集成环境。

什么是 WebGL

WebGL 的（Web 图形库）是现代浏览器技术，它允许创作 3D 图形应用程序。换句话说是 WebGL 是“浏览器中的 3D”。

WebGL 的浏览器支持

现在的 WebGL 在不同程度上被所有浏览器支持。

桌面浏览器

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

移动平台

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

有关详细信息请参阅[相关章节](#) 在问题和解决方案的篇章。

WebGL 的优势

- 工作在浏览器无需安装额外的软件（插件）
- 跨平台，适用于所有台式机和嵌入式系统
- [开放标准](#)，不要求授权费用
- 由 IT 市场的主要参与者支持（谷歌，苹果，微软，NVIDIA，三星，Adobe 和其他）
- 基于 OpenGL 对开发者来说非常熟悉
- 可以与其他[浏览器技术](#) 集成到一起

什么是 Blender

Blender 是一个流行的软件进行 3D 建模和动画，是自由和开放源码的。在此软件创建模型和场景的显示相当于[三维引擎](#) 在网页上的展示。

三维建模

创作图形资源，需要训练有素的专家 - 3D 艺术家。

典型的工作流程可以包括以下阶段：

- 选择的照片和/或创建的概念和草图来做模型或场景的特征（前视图 - 侧视图 - 顶视图）
- 建模 - 3D 模型由多边形创建
- UV 贴图 - 模型被解开铺平后进行的纹理制作（平面图像）
- 纹理 - 覆盖在 3D 模型上的肌理
- 材质设置 - 材料被分配给模型的不同部分和色调（例如，木门与金属柄）
- 装配 - 控制单元（“骨架骨头”）附着在模型之上控制模型动画
- 动画 - 让模型的运作动作可视化，比如-角色动画
- 输出 - 可以在任何阶段来显示 3D 模型的最终形态，例如，在网页上

此外，实践中的改善技术经常用于附加在创建 3D 模型的过程中：

- 创建高精度多边形模型 - 模型的详细版本创建
- “烘焙”一个法线贴图 - 高多边形模型细节转到主模型的一个特殊纹理中（法线贴图）
- 创建镜面贴图 - 不同的反射颜色和比率被分配给不同的模型部件
- 烘焙环境贴图 - 将模型表面周围环境反射记录
- 场景中设置摄像机和光源
- 物理模拟参数设置 - 粒子，布料

创作的 3D 模型和动画所需的时间取决于它们的复杂性和所需的质量，可能从 1-2 天不等（例如一个游戏项目）到 1-2 周（例如详细的飞机模型），甚至数月（现实人物与服装，发型，面部套，动画和特征参数设置）。

浏览器技术

浏览器是一种用于查看网络内容的软件。在互联网技术的初期浏览器的作用是查看文本页面与包含在内的静态图像（“超文本”）。现代浏览器是多媒体网络应用程序完整的平台。

其中在Blend4Web 使用的已经和可能使用的浏览器功能以下技术可以关注：

- 三维图形, [WebGL](#)
- [类型数组](#)
- 对于基于脚本的动画时序控制 (需要动画帧)
- 二维图形, [HTML 画布 2D 上下文](#)
- 声音处理, [网络音频 API](#)
- 二进制数据加载, [XMLHttpRequest Level 2](#)
- 全屏
- 指针锁定
- 多线程, [Web Workers](#)
- 设备的方向

其他有前途的技术：

- 可伸缩矢量图形 (SVG)
- 安全文件访问, [文件 API, File API: 目录和系统](#)
- 浏览器之间的实时通信, [WebRTC](#)
- 持久的网络连接, [WebSocket API](#)
- 游戏手柄

交互式图形

应用于计算机图形的术语“交互式”是指用户可以与不断变化的图像进行交互。例如，用户可以在 3D 场景改变视线方向，移动对象，触发动画和执行通常与计算机游戏相关

的其他动作。

图形交互是通过利用图像的频繁变化实现的，所以帧之间的用户动作（例如，鼠标移动或按压一键）导致图像在下一帧中发生变化。图片必须相互替换非常频繁，人眼无法识别它们单独（至少每秒 30 帧）。

“实时图形”或“实时渲染”，术语在含义上是相似。

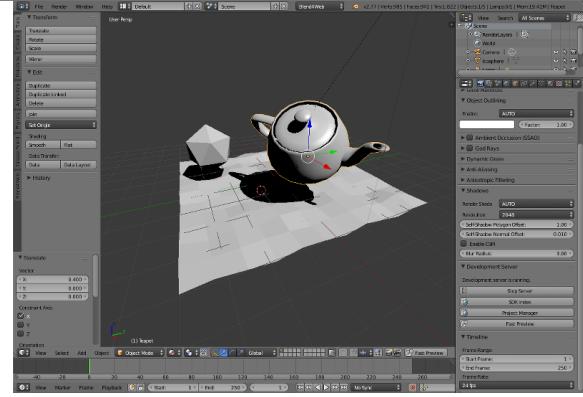
显卡和驱动程序

交互图形是由现代计算机一个特殊目的的硬件的一部分的所谓图形处理器可实现为分离装置（视频卡）或中央处理单元的一部分提供的。

主要图形处理器厂商的台式电脑：- NVidia (GeForce, Quadro), AMD (Radeon), Intel (HD), 嵌入式设备 - ARM (Mali), PowerVR (SGX), Nvidia (Tegra), Qualcomm (Adreno) (商标在括号中指定).

程序访问图形处理器资源，通过一个名为驱动中间程序进行。这是非常重要的交互式图形程序正确的工作有最新版本的系统中的驱动程序。驱动可以从相应的图形处理器厂商的网站进行安装（或升级）。内容的详细信息[WebGL 的无法初始化](#).

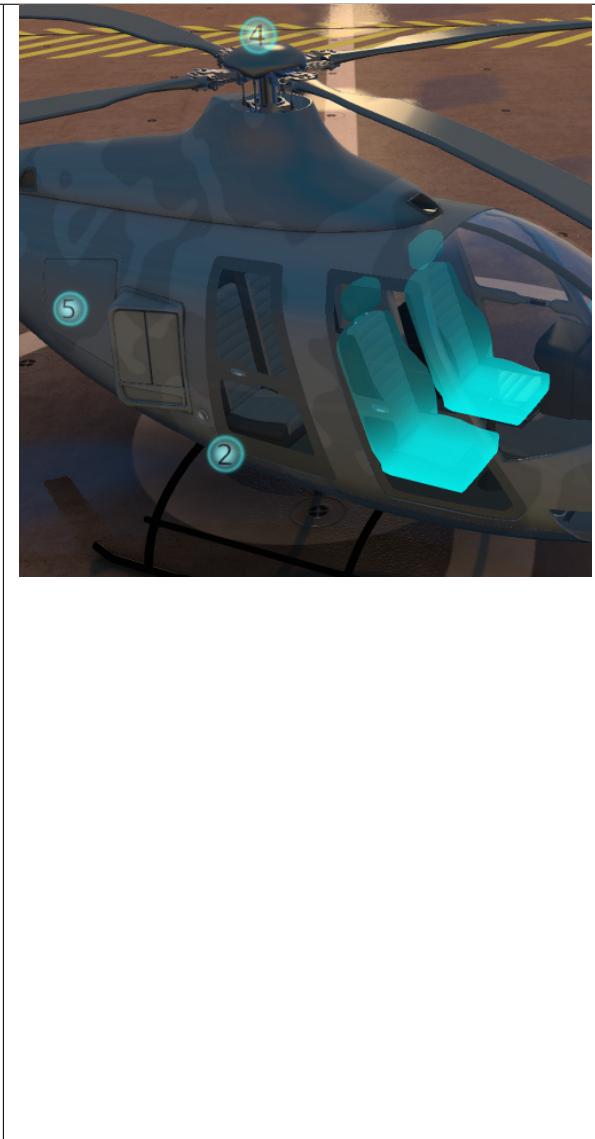
引擎功能

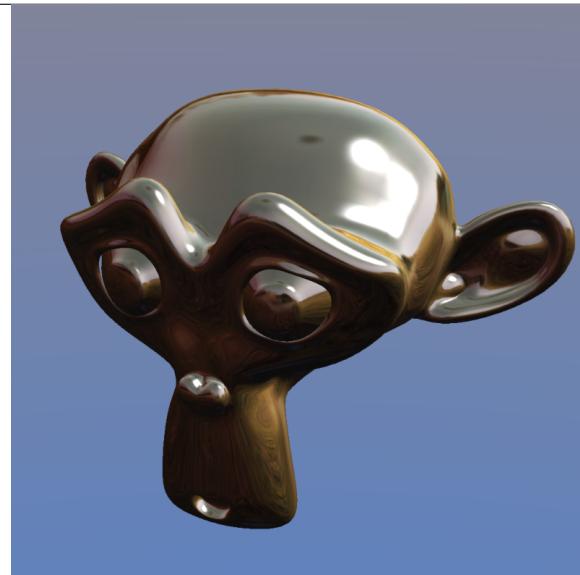


集成到 BLENDER 中

创建交互式应用程序从未如此简单！你可以在 Blender 中简单地创建一个 3D 场景，然后在 Blend4Web 的帮助下使用以下功能[F]

- 快速和方便的场景导出
- 单击即可场景预览
- 全面支持 Blender 的环境和材质设置
- 用于创建和管理项目的工具

<p>渲染</p> <p>Blend4Web 提供丰富的功能，使您的作品看起来令人惊叹。</p> <p>一些功能包括</p> <ul style="list-style-type: none">• 任何复杂的 3D 场景• 级联的阴影贴图• 动态的树叶• 粒子系统• 细节层次• 后期处理效果• 支持 VR	 A 3D rendering of a helicopter cockpit from a front-three-quarter perspective. The cockpit has a camouflage pattern. Inside, two blue 3D human models are seated in the seats. Numbered circles with arrows point to specific features: circle 1 points to the front window; circle 2 points to the side door handle; circle 3 points to the interior seat; circle 4 points to the top of the main rotor blade; and circle 5 points to a panel on the left side of the cockpit. The helicopter is parked on a tarmac with yellow runway markings.
--	---



材质

功能强大且易于使用的材质制作系统使可以让你创作需要的材质，从最简单的卡通类型到几乎照片真实的材质。

一些功能

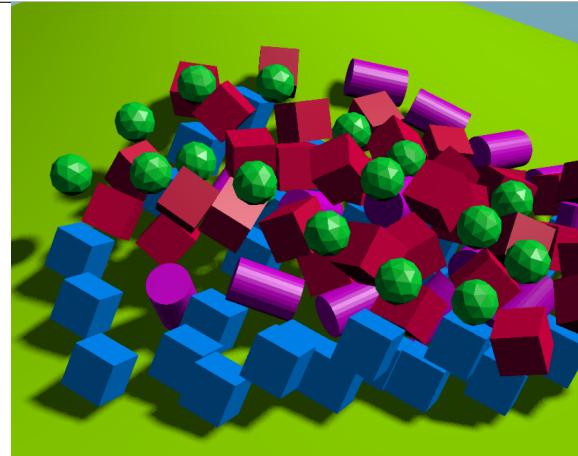
- 节点材质
- 视差偏移量贴图
- 渲染到纹理
- 视频纹理
- 全面支持 Blender 的材质节点
- 反射和折射

动画

Blend4web 支持几乎所有在 Blender 中存在的动画技术，包括：

- 骨骼动画
- 顶点动画
- 物体动画
- 风弯曲：
- 材质动画
- 动画烘焙
- 使用 API 或者逻辑编辑器的流程控制





物理

如果您正在创建一个物理模拟，或只是想要你的场景来表现现实，Blend4Web 的强大物理系统正为您服务。

功能包括：

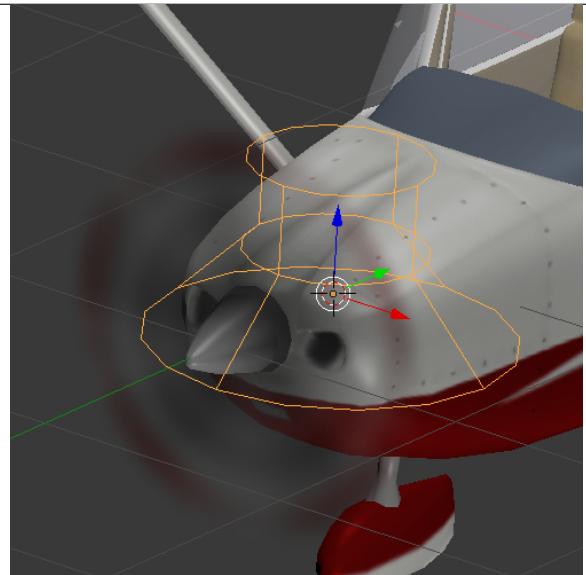
- 碰撞
- 刚体物理
- 铰链系统
- 光线投射
- 浮动物体
- 轮式车辆模拟

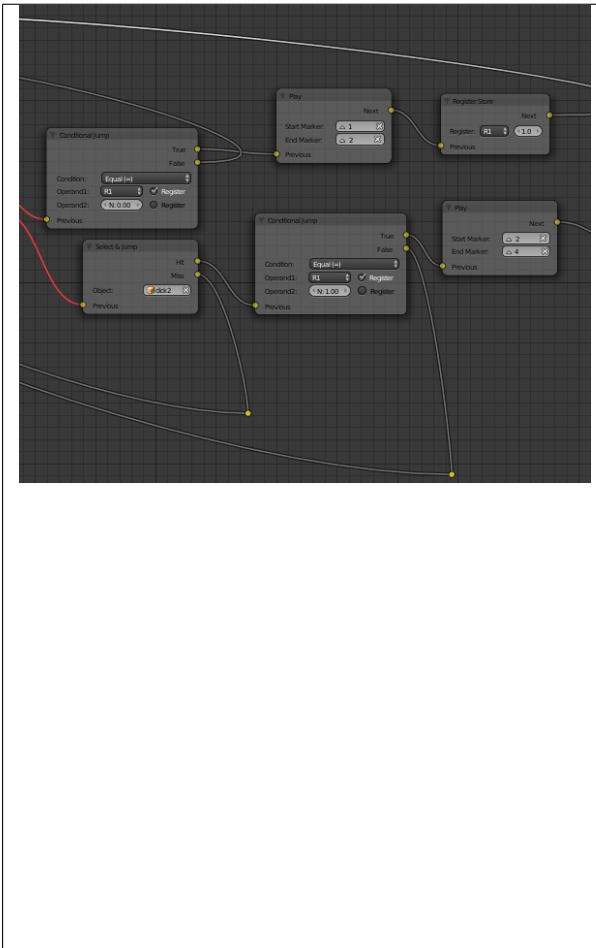
声音

声音在将制作的场景和应用变得对用户操作上更有反馈和让它们更有活力的重要元素。

这里是一些 Blend4Web 提供的你要控制作品会听起来怎么样的手段

- 灵活的播放控制
- 3D 定位
- 多普勒效应
- 动态压缩
- 声音淡入淡出动画
- 实时混音工具





逻辑编辑器

使用此可视化编辑器，您可以创建逻辑节点树来控制你的场景的行为，而无需为添加交互性而编写一行代码！

逻辑编辑器功能

- 物体拾取
- 动画控制
- 环境变量
- 开发服务器

脚本

Blend4Web 使用 JavaScript 方案来控制应用程序的各个方面。

一些 Blend4Web JavaScript API 功能包括

- 模块结构
- 事件驱动模型
- GLSL 着色器
- 用于商业用途引擎的代码缩减和混淆处理

```
"use strict"

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

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

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

安装和更新

目录

- 安装和更新
 - 安装顺序
 - * 安装 Blender
 - * 安装 Blend4Web SDK
 - * 安装 Blend4Web 插件
 - * 切换到 Blend4Web 模式
 - 更新
 - * 更新 SDK
 - * 更新插件
 - * 更新保存的项目

安装顺序

这里有的两个版本的 Blend4Web 框架：Blend4Web SDK 和 Blend4Web 插件。为了确定哪一种最适合你的需求，你需要决定哪种你正在计划开发应用程序。

如果你打算使用 Blend4Web 引擎所提供的所有功能，您应该安装 Blend4Web SDK。

如果你的目的是要开发小规模的项目，或者如果您打算只使用的 Blend4Web 引擎的功能的数量有限（例如法线编辑器），你可能会考虑安装 Blend4Web 插件 来代替。

注意：插件只能导出场景到 HTML 格式 和有限功能（例如，它没有项目管理，示例场景，用户手册，其他额外资源等等），但它仍然拥有一切让你可以创建一个简单的应用

程序。

安装 Blender

直接在 Blender 中创作 3D 场景，这是开放源码软件和免费分发软件。

安装之前，请下载并安装兼容的 Blender 版本，根据此 [表](#)。

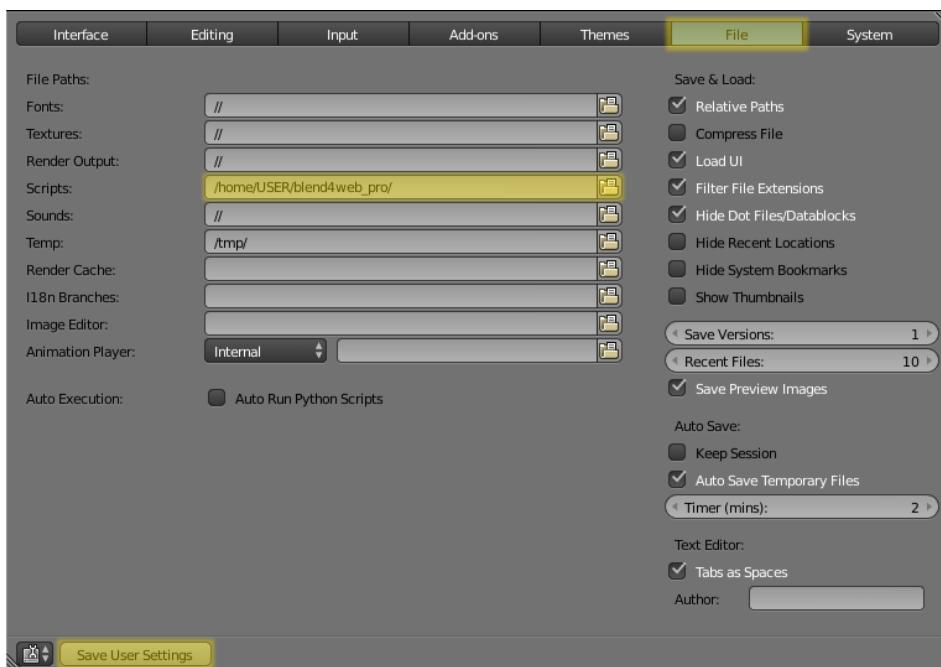
应使用当前的稳定版本的 Blender。它可以从 [官方网站](#) 或从 Blend4Web 网站 下载。



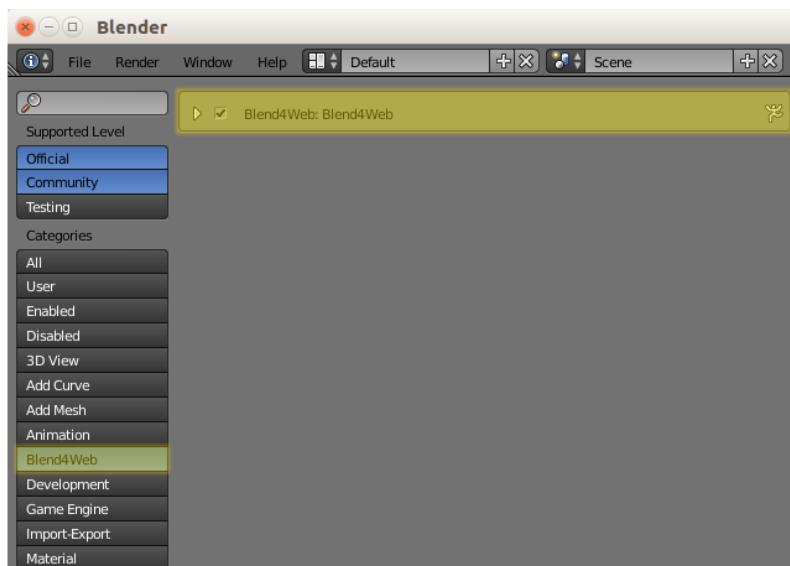
安装 Blend4Web SDK

稳定版作为压缩包发布 (blend4web_sdk_free_YY_MM.zip – 免费 SDK, blend4web_sdk_pro_YY_MM.zip – 商业 SDK)。只要在某处解开这个压缩包。

1. 打开 Blender。
2. 打开用户首选项面板 File > User Preferences....
3. 打开 File 选项卡。
4. 在 Scripts 中设置 SDK 目录的路径。
5. 单击 Save User Settings 按钮。



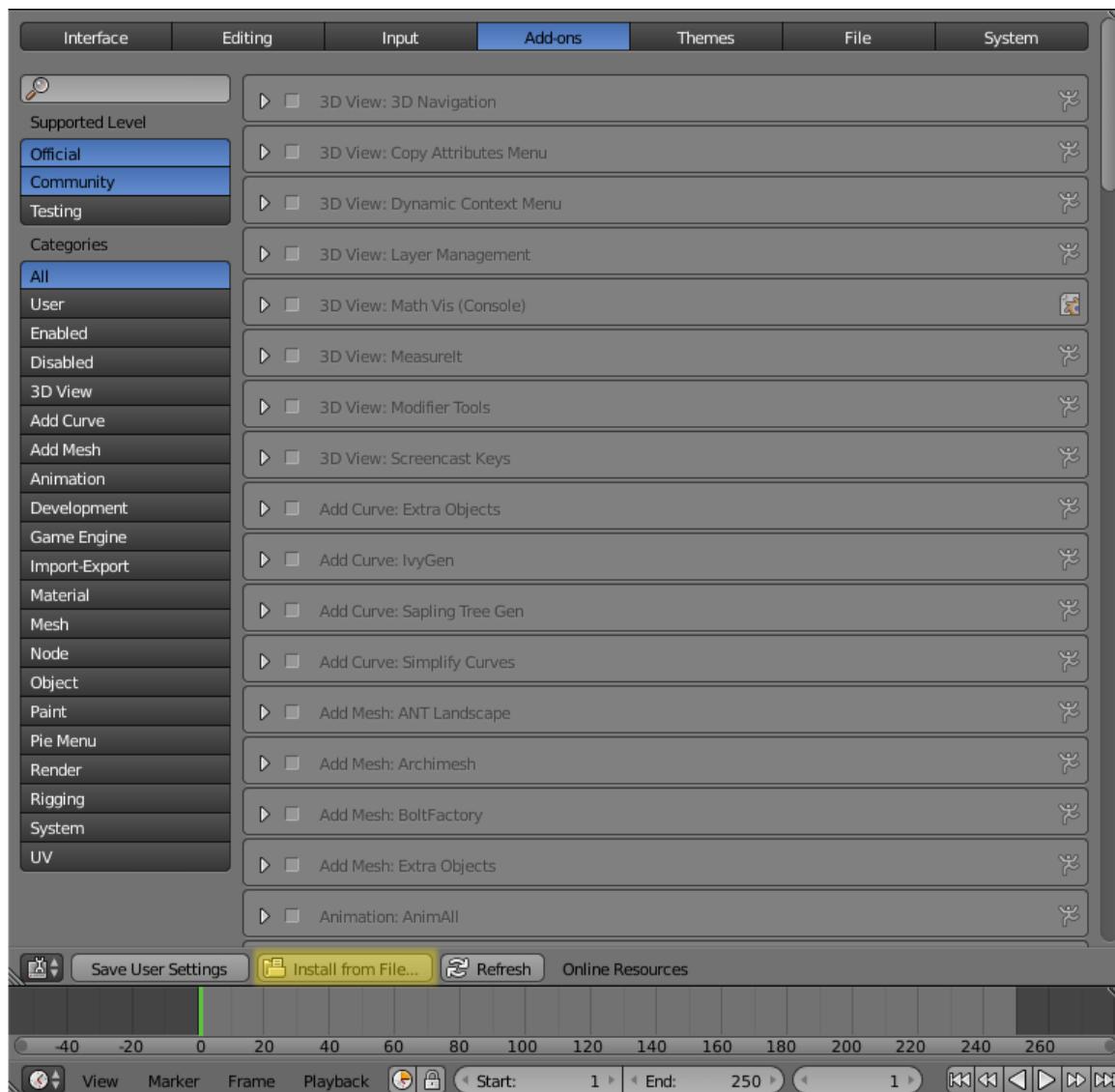
6. 重新启动 Blender。
7. 打开用户首选项面板 File > User Preferences....
8. 打开 File 选项卡。
9. 启用 Blend4Web 插件。
10. 单击 Save User Settings 按钮。



验证是否工作:

在 File > Export 菜单 Blend4Web (.json) 和 Blend4Web (.html) 选项应该出现。

安装 Blend4Web 插件



它可以和其他 [Blender 插件](#) 的安装方式相同。

1. 打开 Blender。
2. 打开用户首选项面板 [File > User Preferences....](#)
3. 打开 File 选项卡。
4. 按 [从文件安装...](#) 按钮。
5. 选择包含 Blend4Web 插件的文件，然后按 [从文件安装...](#) 按钮。
6. 单击 Save User Settings 按钮。
7. 重新启动 Blender。

8. 打开用户首选项面板 File > User Preferences....

9. 打开 File 选项卡。

10. 在列表中找到 Blend4Web 插件。

11. 勾选标题左侧的框以启用它..

12. 再按一次点击 Save User Settings 按钮。

现在，一切都应该能正常工作了。

切换到 Blend4Web 模式

为了查看该引擎设置，从顶部面板菜单中选择 Blend4Web：



更新

更新 SDK

在更新之前，我们建议您使用 `project export` 工具备份的所有开发中的项目。

要更新 SDK，请按照下列步骤操作：

1. 下载 SDK 的新版本。

2. 打开 Blender。

3. 打开 用户设置窗口。

4. 打开 Add-ons 面板。
5. 禁用 Blend4Web 插件。

注解: 我们建议不要使用 Remove 按钮禁用插件来更新 SDK，因为这样做会彻底删除插件，包括它的 SDK 文件。

6. 解压新版本 SDK。

注解: 有两种方法来更新 SDK。

首先，可以完全删除 SDK 文件夹中，在那之后，提取新版本文件夹并将路径设置为它在 Blender 设置，就像你第一次安装 SDK 一样。

第二，你可以简单地使用新的覆盖过时的 SDK 文件。此方法被认为是不太“纯净”，但在大多数情况下，它不应导致任何问题。

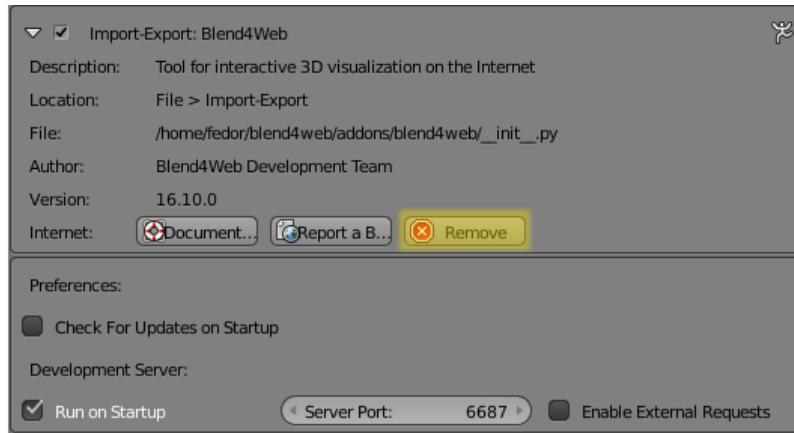
7. 在 用户设置窗口 中打开 File 面板。
8. 设置 Scripts 表格的路径 SDK 文件夹。
9. 单击 Save User Settings 按钮。
10. 重新启动 Blender。
11. 再次打开 用户设置窗口。
12. 打开 Add-ons 面板。
13. 启用 Blend4Web 插件。
14. 再按一次点击 Save User Settings 按钮。
15. 检查一切正常。

完成更新后，您可以使用 [项目导入](#) 工具导入所有已保存的项目。

更新插件

如果你只使用 Blend4Web 插件，按照下面使用说明来做。

安装新版本的插件之前，您可以先删除现有之一。要做到这一点，展开 Blend4Web 插件的信息选项卡，然后按 Remove 按钮。删除旧版本安装新的不是必需的，但这样做可以确保不会有任何版本冲突。



1. 下载包含新版本插件的文件，并将其保存到硬盘上的任何地方。
2. 打开 Blender。
3. 打开 用户设置窗口。
4. 切换到 Add-ons 面板。
5. 从你下载的存档安装新版本的插件。
6. 单击 Save User Settings 按钮。
7. 重新启动 Blender。

一切都应该能正常工作了。

更新保存的项目

你更新你的 SDK（或插件）后，您可以导入之前导出的项目更新回项目管理器。要执行此操作，请按照以下说明

1. 打开项目管理器
2. 导入您的项目使用 Import Project(s) 按钮。
3. Reexport the projects' .blend files using the re-export scenes link.
4. 对每个导入的项目使用 check modules 链接来确保在程序代码中有没有缺少或未使用的模块。如果有，你可以试着使用 Update Modules 按钮来修复。

5. 然后使用 build 命令，生成版本使导入的项目再次工作。
6. 开发者版本的项目使用 Copy 或 Compile 引擎绑定类型也可以构建出与新版本的 SDK 一起正常工作（与其他绑定类型的项目的开发版本没有它也可以工作）。

工作流程

目录

- 工作流程
 - 创建一个新项目
 - 创建场景
 - 准备一个在 Blend4Web 中使用的场景
 - * 在浏览器显示场景
 - 导出场景
 - 应用程序开发
 - 构建项目
 - 项目部署

开发任何产品是一个创造性的过程，需要许多参与者谁拥有不同的技能和经验。然而不论事情是多么的复杂，是它总是能够分离出许多生产阶段，其中的大部分资产和源代码是创作的目标。

当使用 Blend4Web 工作流程如下：

1. 创建一个新项目
2. 创建场景
3. 准备一个在 Blend4Web 中使用的场景
4. 导出场景。
5. 应用程序开发

6. 构建项目

7. 项目部署。

创建一个新项目

要在一个项目上工作，首先就是要创建它。

创建一个新项目最简单的方法是使用[项目管理器](#)。要做到这一点，在项目管理器的主窗口顶部按新建项目按钮。

另一种方法是使用 `project.py` 命令行脚本，这个更通用。这个脚本的工作描述[在这里](#)。

创建场景

场景都包含在 `blend` 文件。项目管理器可以在创建新项目时自动创建一个基本的 `blend` 场景文件。额外的 `blend` 文件可以被创建并附加到一个项目。

Blend4Web 的场景都跟 Blender 里创建场景都是以同样的方式，只有轻微的差异。处理一个场景的工作过程可以分为几个阶段：

1. 建模场景中的物体
2. [设置材质](#)
3. [给物体动画](#)
4. [创建场景逻辑](#)（这部分是 Blend4Web 引擎独有的，并不适用于普通 Blender 场景）

应该注意的是，并不是每一个项目都需要所有这些阶段。更简单的场景可能除了默认的功能不会有任何动画甚至任何的材质。

各个阶段的顺序也不是严格的，可以一定程度的互换（例如，你不可能在创建物体钱进行动画，但你可以预先为它创建一个材质）。

除了一般的阶段如建模，贴图，动画等，场景应该在做好在引擎中的准备。

一般建议：

1. 我们强烈建议将 Blend 文件保存在 projects/project_name/blender 目录中。程序中不需要加载的辅助文件（例如，引用）等，也该在这个文件夹下。
2. 图像和媒体文件应该是在外部和位于 projects/project_name/assets 目录。
3. 你将导出的文件应包含实际所需（对象、材料、材质、动画等）的应用程序中的资源。
4. 场景组件应该有不同的识别名称。他们不应该叫 “Cube.001”, “Material” 或者 “Armature”.
5. 我们建议您实践从其他文件（库）来链接组件。

准备一个在 Blend4Web 中使用的场景

除了一般的阶段如建模，贴图，动画等，场景应该在做好在引擎中的准备。

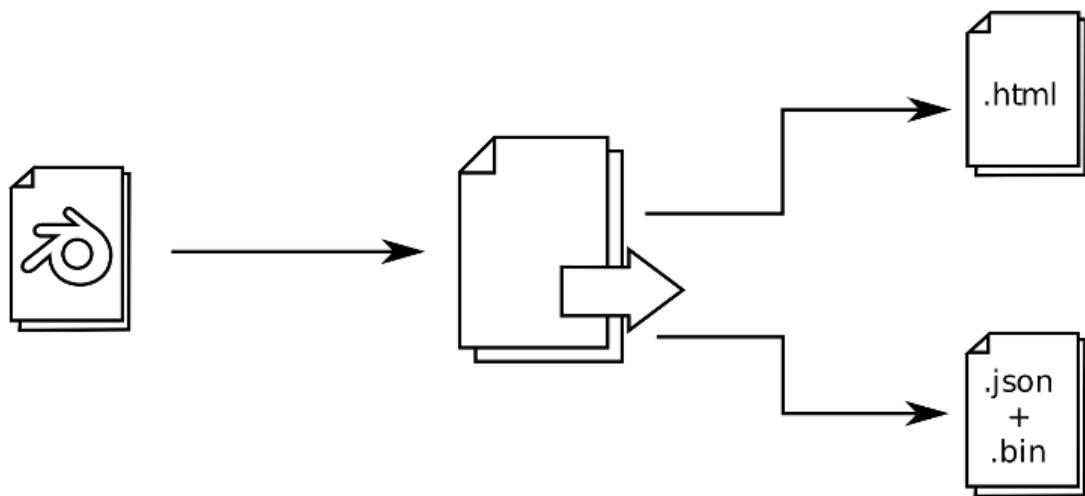
在浏览器显示场景

当使用[本地开发服务器](#) 有可能使用预览当前场景的 Fast Preview 按钮，位于 Blender 的 Render ->Development Server 面板。在这种情况下，场景将里面的一些临时存储被导出并在预览查看器应用程序加载。

另一种方法是使用[运行在预览查看器](#) 导出选项。在这种情况下，场景将导出后立即显示在预览查看器应用程序。

导出场景

导出过程将场景从 Blender 格式转换为 Blend4Web 所使用的格式之一。



有两种格式 JSON 和 HTML。

HTML 格式用于简单的单机应用中，所有的资源都打包成可以很容易地部署到一个网页的一个 HTML 文件。

JSON 格式，可用于创建各种项目，但它也提供了许多额外的功能使得它可以创建需要编程的复杂项目。格式的不同和导出选项在 [Export Formats](#) 一章描述。

若要导出场景，请从 [文件 > 导出](#) 菜单选择 Blend4Web (.json) 或 Blend4Web (.html)，或者在操作符搜索菜单（按空格键的时候出现）输入“b4w export”

注解: 如果场景使用 Blender 功能尚不由 Blend4Web 引擎支持，可能会在导出过程中发生错误。在导出错误中列出此表。

导出选项中详细描述在 [相应章节](#)。

应用程序开发

本节仅适用于除网络播放器的 HTML 和网络播放器 JSON 的其他项目类型。

在此阶段的应用已经被创建。逻辑场景加载和用户交互使用 JavaScript 编写。应用程序开发人员笔记中给出了 [相应章节](#)。

构建项目

一个项目的开发完成之后，应当构建以便它以后可以在 SDK 之外使用。

项目构建可以通过点击在[项目管理器](#) 中的 build project 链接执行。

构建项目之前，您可能希望将用在项目中的媒体资源[转换](#)，可以最大限度地缩减大小。

项目部署

项目开发完成后，该项目可被部署，这样它可以在 Blend4Web SDK 之外使用。项目部署通过点击[项目管理器](#) 主窗口的项目类右边栏中的 Operations 命令列表里的 deploy project 链接来执行。

点击这个链接会使得项目管理器执行所有必须的操作和部署项目。

部署的项目都有一个包含所有项目文件归档的形式。这个档案可以保存在硬盘的任何位置，只需通过你使用的浏览器的标准保存文件对话框。

一个完整的项目可以放置在 Web 服务器上。

要做到这一点，从项目管理器下载存档然后提取项目文件，并将其上传到 Web 服务器（同时保留内部文件结构）可以通过 FTP，SSH 或服务器支持的任何其他文件传输协议的方式。这样做后，你只需要把这个项目放在一个网页中使用，例如一个 iframe 容器。

部署项目到 Web 服务器的描述在[项目管理器](#) 章的相应节 中。

项目管理

目录

- 项目管理
 - 项目管理器
 - * 项目创建向导
 - * 创建一个 Web 播放器应用程序
 - * 项目信息
 - * 项目编辑
 - * 项目配置
 - * 项目导入
 - * 项目导出
 - * 部署项目
 - 高级项目管理
 - * 依赖
 - * 项目清单
 - * 项目结构
 - * 项目配置文件 (.b4w_project)
 - * 创建项目
 - * 开发一个项目中的多个应用程序
 - * 构建项目
 - * 自动 Blend 文件导出
 - * 资源转换
 - * 部署项目
 - * 移除项目
 - * 应用程序升级新版本的 SDK

自版本 15.09，SDK 中包含一种项目管理功能，使它成为可能

- 浏览完整列表和文件结构的项目；
- 启动应用程序，在查看器中运行导出的场景，在 Blender 中打开 blend 文件；
- 创建和配置新的应用程序，包括那些基于现成的模板；
- 构建应用程序并打包方便地部署在远程的服务器；
- 将应用程序资源（纹理、音频和视频文件）转换为其他格式，以确保跨浏览器和跨平台的性能；
- 自动重新导出应用程序的所有场景，包括导出到 JSON 和 HTML 格式；

- 删除项目。

项目管理器

SDK 的首页的 工具项中可以运行 项目管理应用。启动后，该应用程序将输出 SDK 中的所有当前项目列表。

Project Manager					
Icon	Project Name / Apps ▾	Project Directory / Build Directory	Blend Files	JSON Files	Operations
	boat_p08 (int) (edit) (config) player:boat_p08.json	blender/capri/vehicles/boat_p08	blender/capri/vehicles/boat_p08/ (show)	deploy/assets/capri/vehicles/boat_p08/ (file) boat_p08.json	re-export scenes convert resources deploy project remove project
	capri (int) (edit) (config) dev:capri.html built:capri.html	apps/dev/capri deploy/apps/capri	blender/capri/ (show)	deploy/assets/capri/ (show)	build project check modules re-export scenes convert resources deploy project remove project
	car_bv_eb164 (int) (edit) (config) player:car_bv_eb164.json	blender/capri/vehicles/car_bv_eb164	blender/capri/vehicles/car_bv_eb164/ (show)	deploy/assets/capri/vehicles/car_bv_eb164/ (file) car_bv_eb164.json	re-export scenes convert resources deploy project remove project
	demos_postprocessing (int) (edit) (config) player:blue Jean.json player:blue shirt.json player:gold cap.json player:motion_blue.json player:ssaa.json	apps/dev/demos_postprocessing deploy/assets/postprocessing	blender/postprocessing/ (show)	deploy/assets/postprocessing/ (show)	re-export scenes convert resources deploy project remove project
	farm (int) (edit) (config) dev:farm.html built:farm.html	apps/dev/farm deploy/apps/farm	blender/location_agriculture/ (show)	deploy/assets/location_agriculture/ (file) exports/location_agriculture.json	build project check modules re-export scenes convert resources deploy project remove project
	firstperson (int) (edit) (config) dev:firstperson.html built:firstperson.html	apps/dev/tutorials/firstperson deploy/apps/tutorials/firstperson	blender/tutorials/firstperson/ (show)	deploy/assets/tutorials/firstperson/ (file) firstperson.json	build project check modules re-export scenes convert resources deploy project remove project
	mi_34c1 (int) (edit) (config) player:mi_34c1.json player:mi_34c1_demo_scene.json	blender/capri/vehicles/mi_34c1	blender/capri/vehicles/mi_34c1/ (show)	deploy/assets/capri/vehicles/mi_34c1/ (file) mi_34c1.json mi_34c1_demo_scene.json	re-export scenes convert resources deploy project remove project
	naturemorte (int) (edit) (config) player:naturemorte.json	blender/fridge blender/fridge/naturemorte	blender/fridge/naturemorte/ (hex) naturemorte.blend	deploy/assets/fridge/naturemorte/ (hex) naturemorte.json	re-export scenes convert resources deploy project remove project
	petgors_table (int) (edit) (config) dev:petgors_table.html built:petgors_table.html built:quest.html	apps/dev/petgors_table deploy/apps/petgors_table	blender/petgors_table/ (show)	deploy/assets/petgors_table/ (show)	build project check modules re-export scenes convert resources deploy project remove project
	space_disaster (int) (edit) (config) dev:space_disaster.html built:space_disaster.html	apps/dev/space_disaster deploy/apps/space_disaster	blender/space_disaster/ (show)	deploy/assets/space_disaster/ (file) space_disaster.json	build project check modules re-export scenes convert resources deploy project remove project
	watch_scene (int) (edit) (config) player:watch_scene.json	blender/capri/props/watch_scene	blender/capri/props/watch_scene/ (show)	deploy/assets/capri/props/watch_scene/ (file) watch_scene.json	re-export scenes convert resources deploy project remove project

项目管理的命令位于页面的顶部。

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

[返回首页](#) 返回到 Blend4Web SDK 索引页。

[新建工程](#) 打开项目创建向导。

[导入项目](#) 打开project import 对话。

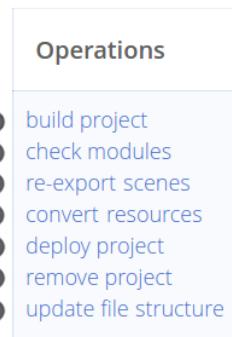
[导出项目](#) 打开project export 页。

隐藏项目 可以用来隐藏项目。如果已经隐藏了此类项目，则此命令替换 显示项目命令。

帮助 打开帮助文件.

用于管理特定项目的命令在它的右侧。

注解: 根据项目的类型的一些命令可能无法使用。



1. 编译项目（WebPlayer JSON 和 WebPlayer HTML 项目不可用）。
2. 检查是否缺少或不必要的 API 模块（WebPlayer JSON 和 WebPlayer HTML 项目不可用）。

激活此选项将使项目管理器检查应用程序使用的模块，并将检查结果的
结果输出到控制台。可能的警告消息如下：

- (a) 如果一个或多个 API 模块从项目的文件夹中丢失，将会显示以下信
息：

模块 ‘module_name’ 在 ‘project_name.html’ 项目丢失，请包
括它或运行 ‘Update Modules’.

The screenshot shows the 'Project Manager' interface with a 'Operation Report' section. It displays the following message:

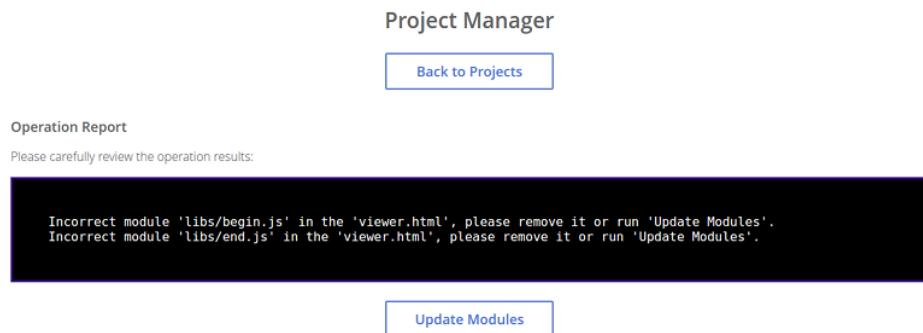
```
Module 'addons/gp_conf.js' is missing in the 'debugger.html', please include it or run 'Update Modules'.
```

Below the message is a blue 'Update Modules' button.

这些缺失的模块可以被添加到项目中的 Update Modules 按钮

- (a) 如果项目具有未在程序代码的任何地方使用的一个或多个的 API 模块，将显示以下信息：

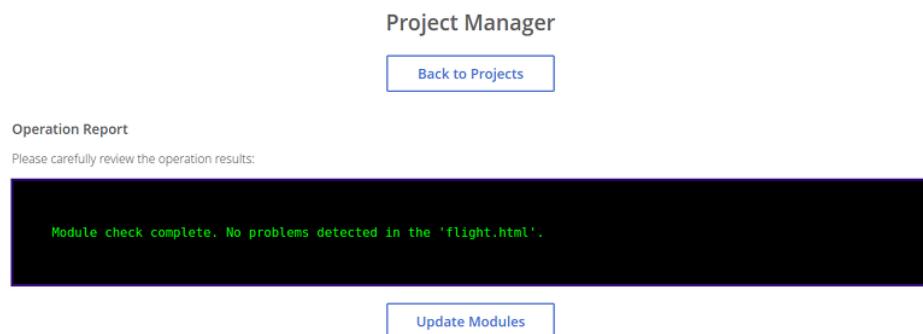
在 ‘project_name.html’ 中的 ‘不正确的模块 ‘module_name’’，
请删除它或运行 ‘Update Modules’.



这些不正确的模块可以从项目中用“更新模块”按钮删除。

- (a) 在已检测缺失/不需要的模块没有任何问题的情况下，将显示以下信息：

模块检查完毕。在 ‘module_name.html’ 中检测到没有问题



3. 从项目重新导出 blend 文件。
4. 转换媒体资源。
5. 导出并下载项目文档。
6. 移除项目
7. 更新文件结构. 仅可用于 Blend4Web 16.12 或更旧版本创建的项目。

注解: 项目的所有路径都是从它的.b4w_project 配置文件 中检索。

位于项目的名称旁边的是链接到

1. [项目信息页面](#),
2. [项目文件编辑器](#) 和
3. [项目配置器](#).

这里也指定应用程序类型。应用程序可以具有下列类型之一

播放器 可以使用 Web 播放器播放应用程序。

开发 开发中的应用。

构建 编译应用程序。

项目创建向导

该工具用于创建新的项目是列入项目管理应用程序中，由主页面上 [Create New Project] 按钮执行。

Create a new project

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

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

[Create Project](#)

[Back to Projects] 按钮可以用来返回到项目管理的主页面，而 [Help] 按钮可以用到访问帮助文件。

1. 项目名称习惯为“projects/project_name”，“projects/project_name/assets”，“projects/project_name/blender”和“projects/project_name/build”等名称的项目目录。
2. 在浏览器中所示的项目标题。
3. 项目作者的名字。
4. 添加应用程序模板。标准应用程序模板 html 文件，css 文件，js 文件将添加到

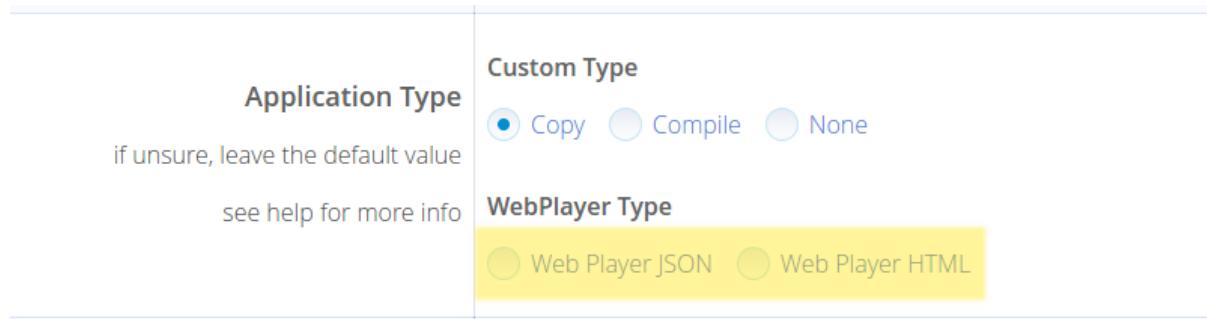
项目目录“projects/project_name”。

5. 添加场景模板。标准的 json 文件将添加到“projects/project_name/assets” 目录;blend 文件将添加到“projects/project_name/blender” 目录。
6. 使用材质库。材质库的 Blend 文件会复制到项目文件夹“projects/project_name/blender”，资产文件复制到“projects/project_name/assets” 目录。
7. 复制项目管理脚本。Project.py 脚本将复制到项目目录。
8. 项目的类型。有几个选项F
 - Copy - 引擎文件将直接从 deploy/apps/common/ 目录复制到应用文件夹。
当项目展开时，唯一的应用程序文件进行编译，而引擎文件仍保持不变（所以你不能修改引擎本身）；
 - Compile - 引擎源代码应用程序的脚本一同编译；
这个选项可以用于修改引擎本身的代码；
 - None - 项目管理器不会将引擎文件复制到应用程序文件夹，也不将它执行任何操作生成应用程序。应用程序开发人员将不得不手动执行他们所需要的一切；
 - Web Player JSON - 项目下的 json 文件通过 SDK 里的 web 播放器播放；
 - Web Player HTML - 项目被打包到单独的 html 文件，包含所有所需的资源。
9. Javascript 优化级别。
 - “简单” —— 变量的名称替换代码；
 - “先进” —— 进行了代码优化；
 - “只空格” —— 只有空格从代码中移除。
10. 网络播放器的 URL 属性。此选项卡仅当 Engine Binding Type 参数设置为 Web Player JSON 或 Web Player HTML 时可用。
 - “显示 FPS” 用于显示在玩家的右上角的 FPS 计数器。
 - “自动旋转摄像头” 让场景加载后立即自动旋转相机。
 - “禁用社交网络按钮” 禁用控制面板上的社交网络按钮。

- “背景透明” 启用渲染区域的透明度。
- “使用压缩纹理” 使精缩和压缩纹理加载（以 DDS 格式）。

创建一个 Web 播放器应用程序

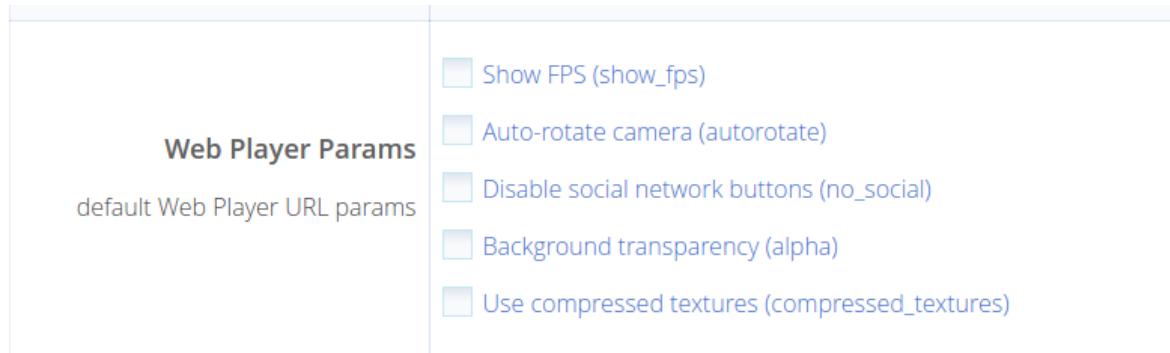
网络播放器应用的主要优点是易于在网站上部署这种应用。



创建一个 Web Player 应用程序很简单。所有你需要做的，当创建一个新的项目选择时在引擎绑定类型面板下面选择 Web Player JSON 或 Web Player HTML 选项。

可用于在 Web 播放器项目的参数大多是和任何其他类型的项目一样的。唯一的例外是被称为网络播放器 PARAMS 组参数。

项目设置



如果 Web Player JSON 或 Web Player HTML 选项已启用，此面板才可用。这里列出的参数在本质上是运行该项目的 Web 播放器应用程序将使用的 URL。

构建项目并部署

在一个项目上的工作完毕后，它能够生成和部署到 web 服务器。所有这一切都在相应章节 中描述

项目信息

此页包含有关选定项目的信息，可以在该项目的标题旁边的 [info] 链接查看。

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](#)

命令

回到项目 返回到项目管理的主页面。

帮助 打开帮助文件.

项目参数

项目名称 项目名称

项目的标题。 在浏览器中所示的项目标题。

项目作者 / 公司 该项目的作者或开发商公司的名称。

项目图标 该项目的图标。

应用程序 项目应用程序。

引擎绑定类型 设置项目类型

项目路径 (开发目录) 项目的目录。

项目配置 项目的配置文件。

构建目录 项目构建目录。

Blend 目录 项目的 blend 文件目录的位置。

资产目录 项目的媒体资产所在的位置目录。

URL 参数 URL 参数 列表用来启动应用程序。

JavaScript 混淆级别 Javascript 优化级别。

JS 编译通过列表 该项目的 JavaScript 文件汇编直通例外列表。

CSS 编译通过列表 该项目的样式表编译直通例外列表。

构建忽略列表 项目编译中的例外列表。

部署资产目录 资产将被放置部署项目的目录。

部署资产 URL 前缀 `get_assets_path()` 所报告的资产目录部署项目中 URL 路径前缀。

部署忽略列表 项目的部署例外列表。

项目编辑

简单的基于 Web 的界面可以编辑项目文件，通过项目的标题旁的 [编辑] 链接。

Edit project source files

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

Files	Editor
projects/my_project/b4w_project projects/my_project/my_project.js projects/my_project/my_project.html projects/my_project/my_project.css	<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>

[Save](#)
[Save As...](#)

编辑器窗口的左的部分包含所有从项目目录 (projects/my_project) 中 .html, .css 和 .js 文件的列表。右侧部分包含带有高亮语法的当前选定的项目文件（默认无选中的文件）的内容。

注解: 这个界面可以仅用于编辑项目的开发版文件，但不能编辑建好的版本。

新的 .html, .js 和 .css 文件可以通过单击 [New File] 按钮添加到项目目录。

可在页面底部找到了 [Save] 和 [Save As...] 按钮用来保存当前选定的项目文件。

项目配置

这是一个 web 界面，允许开发人员在创建项目后查看和更改项目设置。

Project Config

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

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

此页面包括项目创建过程中的所有设置以及一些额外的参数。一些参数是只读（因此创建项目后不能更改）。

Info 设置组

名字 项目名称

这是一个只读的参数。

名称 在浏览器中所示的项目标题。

作者 该项目的作者或开发商公司的名称。

图标 该项目在项目管理器中的图标。

其图标可以替换的选择文件按钮。如果图标不指定（因为它是默认情况），则使用 Blend4Web 标志。

Build 设置组

Apps 项目中用于编译应用程序的 .HTML 文件。每个 .HTML 文件被认为是一个单独的应用程序。

使用物理 启用和禁用添加物理引擎文件到该项目。如果该项目不使用物理可以关闭。

此参数在 WebPlayer HTML 类型项目中总是禁用但 WebPlayer JSON 类型中总是启用。对于其他类型的项目，它可以手动设置。

应用程序类型 设置项目类型

这是一个只读的参数，因为不同类型的项目有不同的内部结构。

JavaScript 混淆级别 编译应用程序时 Javascript 优化级别。

此参数为 None，WebPlayer HTML 和 WebPlayer JSON 类型项目设置为只读模式。

JS 编译通过列表 该项目的 JavaScript 文件汇编直通例外列表。

CSS 编译通过列表 项目的 CSS 文件直通例外列表。

构建忽略列表 项目编译中的例外列表。

Paths 设置组

Developer Project Path 开发的应用程序的路径。

这是一个只读的参数。

Build Project Path 编译应用程序的路径。

Blend 目录 blend 文件夹的路径

资产目录 资产文件夹的路径。

Blender Exec Blender 的可执行文件路径。

Deployment Path 为项目最终部署文件夹的路径。

Deploy 设置组

部署资产目录 资产将被放置部署项目的目录。

部署资产 URL 前缀 `get_assets_path()` 所报告的资产目录部署项目中 URL 路径前缀。

部署忽略列表 项目的部署例外列表。

Override Deployment Path 删除部署目录，如果它已经存在。

URL GET 参数

在项目管理器界面中，此字段中为 URL 链接指定额外的 GET 参数。

项目设置可以按屏幕底部的 [Save Config] 按钮保存。

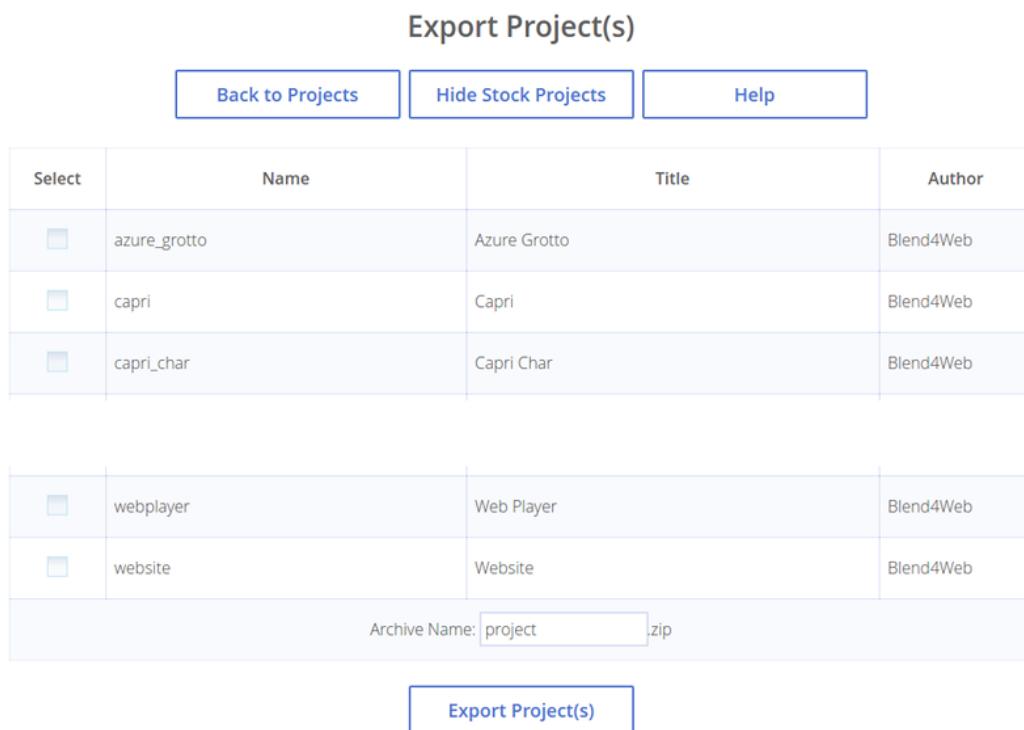
项目导入

导入项目工具是在 [Import Project(s)] 链接

点击该链接会打开标准的文件开启对话框，你可以选择需要导入的项目。

项目导出

项目输出窗口可以通过 [Export Project(s)] 链接访问。



命令

回到项目 返回到项目管理的主页面。

隐藏项目 隐藏项目。如果已经隐藏了此类项目，则此命令替换为 [显示项目] 命令。

帮助 显示帮助页。

导出项目 可以用来导出所选的项目。

项目参数

选择 显示项目是不是选择了要导出。

名字 该项目的目录的名称。

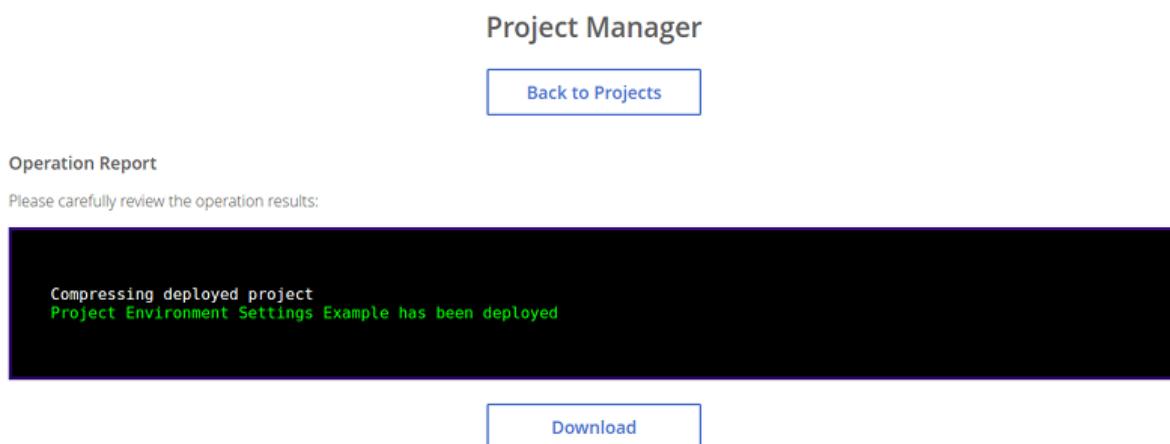
名称 项目的标题。

作者 项目作者的名字。

档案名称 导出的项目打包的档案名称。

部署项目

您完成您的项目后，在项目管理器的主页面从 Operations 面板选择 deploy project 选项。该项目将被导出，打包成一个单一的存档，并下载到您的网页浏览器存储所有下载文件的文件夹。



在那之后，该存档文件然后可以下载的到您的 web 浏览器存储下载的所有文件按下载按钮的位置的文件夹。应该指出的是，下载的过程可以产生大量的时间取决于归档文件的大小。

若要将项目放在网页上，你必须提取此文档，并用 FTP，SSH 或由支持的任何其他协议服务上传文件（html 文件和 assets 目录）发送到 Web 服务器。

注解： 将项目文件上传到 Web 服务器上，应保留对其文件的归档和相关路径的内部结构。

一个 Web 播放器的 HTML 应用程序然后可以使用一个 iframe 容器放置在任何网页。下面是可以用于此的 HTML 代码的示例：

```
<!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>
```

部署 JSON 项目也是同样地进行，但不是一个 HTML 文件，它是使用的 Web 应用程序播放器，并包含实际场景中的 JSON 文件的组合。

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

高级项目管理

高级项目管理器由经验丰富的开发人员使用，在需要更多的灵活性和需要自动化项目开发过程中使用。

高级项目管理器使用 project.py 脚本并手动编辑 .b4w_project 配置文件。

Project.py 脚本可以在 Blend4Web SDK 目录的 ./apps_dev/ 文件夹中找到。如果在创建项目时，设置了 Copy Project Management Script 选项，也可以在项目的根文件夹中找到该脚本。

依赖

项目管理系统在所有操作系统都可用。但是，某些操作可能需要安装附加依赖项。为了检查是否满足所有依赖项，请使用以下命令

```
./project.py check_deps
```

对于 MS Windows 用户：

```
python project.py check_deps
```

对于脚本运行它需要安装 Java 和 设置 PATH 系统变量。

注解： Java 是默认包含在 Windows 版本的 SDK。在 Linux 和 macOS 它应该手动安

装。

Resource Converter 也使用其自己的描述的外部工具集，在[另外章节](#) 描述.

项目清单

可以用下面的命令查看的 SDK 中的所有项目列表

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

项目结构

这是放置在 / projects 目录中的 SDK 文件夹的根目录下的一个项目名为 MY_PROJECT 结构以及，看上去的样子：

```
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
```

这个应用程序包括 3 个不同的目录。

1. my_project/assets。包含媒体文件（纹理和声音，还有 .bin 和 .json 文件）项目的场景。
2. my_project/blender. 包含项目的场景的 .blend 文件。

3. my_project/build. 包含构建应用程序的文件。

项目的主文件 .b4w_project, 还有 .js, .html 和 .css 文件, 放置在 my_project/ 文件夹的根下。

此外, 部署命令可以创建另一个目录中, 但它通常放置在 SDK 之外, 其名称和路径取决于目标服务器上的目录结构。

注解: 从版本 16.12 开始, 项目结构已经改变。仍然使用旧的结构的项目还可以工作, 而应该被视为过时。我们建议使用 update file structure 按钮, 这些项目的结构升级到一个新的文档结构。

这个按钮可以在 Operations 面板在项目管理器列表中的项目条目的右侧找到。它仅适用于使用淘汰的项目结构用户项目 (不是 stock 类型)。

项目配置文件 (.b4w_project)

项目配置文件包含项目的所有必要的信息, 包括姓名, 元数据, 目录, 应用程序创建和部署信息。

项目配置文件可以手动编辑, 但一个更方便的方法是使用[项目编辑](#)界面。

```
[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 =
```

这是一个标准的 INI 配置文件，其中包含的部分，属性和值。

模块 [info]

包含项目的元数据：

作者 该项目的作者或开发商公司的名称。

名字 项目名称

名称 在浏览器中所示的项目标题。

图标 该项目的图标。

模块 [paths]

包含项目路径：

资源目录 项目的媒体资产所在的位置目录。

blend_dirs 项目的 blend 文件目录的位置。

blender_exec Blender 的可执行文件路径。

build_dir 项目构建目录。

deploy_dir 项目部署的目录。

模块 [compile]

apps 项目应用程序。

css_ignore 该项目的样式表编译直通例外列表。

engine_type 设置项目类型

ignore 项目编译中的例外列表。

js_ignore 该项目的 JavaScript 文件汇编直通例外列表。

优化 Javascript 优化级别。

use_physics 指示您的项目是否使用物理或没有。默认 - 使用物理。

use_smaa_textures 指示您的项目是否使用 SMAA 纹理与否。当前未使用。

version 项目版本。

模块 [deploy]

assets_path_dest 资产将被放置部署项目的目录。

assets_path_prefix `get_assets_path()` 所报告的资产目录部署项目中 URL 路径前缀。

ignore 项目的部署例外列表。

override 在部署阶段替换现有的输出目录。请谨慎使用。

模块 [url_params]

网络播放器项目的可选部分。包含URL 参数 来启动项目应用.

创建项目

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

此命令将创建在当前目录指定名称的项目。默认情况下，项目目录只包括一个配置文件。

可用参数：

- -A | --copy-APP-templates (可选) 创建项目目录的标准应用模板 (my_project_dev.html, my_project.js, my_project.css).
- -C | --author (可选) 写作者的还是公司的配置文件名。
- -o | --optimization (可选) 写在配置文件中的脚本优化级别。
- -P | --copy 项目-script (可选) 在项目目录中创建 project.py 脚本的副本。
- -S | --copy-场景 templates (可选) 创建目录 deploy/assets/<my_project_name> 和 blender/<my_project_name> (my_project.json/.bin 和 my_project.blend 相应的).
- -T | --title“ (可选) 在配置文件写入标题。编译时,会将写入进 HTML 元素 “<title>”。
- -t | --engine-type (可选) 在配置文件中写入引擎类型。

示例：

```
./project.py init -AS -C Blend4Web -o simple -T MyProject -t external my_project
```

这个命令将创建一个名为 my_project 目录, 这里面以下文件将被放置: my_project.js, my_project.css, my_project_dev.html 和 .b4w_project.

该.b4w_project 文件看起来像:

```
[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 =

```

开发一个项目中的多个应用程序

一个项目可以包含多个应用程序。这可以通过用分号分隔的配置文件列表对应的 HTML 文件的形式提供:

```

...
[compile]
apps = myapp1;myapp2;
...
```

如果 apps 字段为空，项目目录中的每个 html 文件将被视为一个应用程序。

构建项目

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

构建在 projects/my_project/build 目录中的项目。

可用参数:

- “-a | --app” (可选) 指定一个 HTML 文件，相对于该项目的应用程序将建成。
- “-c | --css-ignore” (可选) 添加 CSS 样式异常为了不编译它们。
- “-j | --js-ignore” (可选) 添加脚本例外为了不编译它们。

- “-o | --optimization”（可选）指定 JavaScript 文件的优化级别: whitespace , simple (默认) 或 advanced。
- “-v | --version” 加版本和脚本和样式的路径。

编译器要求

- 一个 HTML 文件必须存放在目录的根目录，如果 -a 选项被禁用
- 脚本和样式可以存储在应用程序的根目录和子文件夹中

自动 Blend 文件导出

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

此命令将重新导出混合在 JSON 和 HTML 格式的文件。

可用参数:

- “-b | --blender_exec” Blender 可执行文件的路径。
- “-s | --assets” 用场景资源指定目录。

资源转换

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

外部资源（纹理，音频和视频文件）转换为另一种格式，以确保跨浏览器和跨平台性能。

可用参数:

- “-s | --assets” 用场景资源指定目录。

资源的转换中详细描述在[对应部分](#).

部署项目

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

与所有依赖一起保存了一个项目在外部目录。

可用参数:

- "-e | --assets-dest" 资产目的地目录（“资产”默认情况下）。
- "-E | --assets-prefix" 资产网址前缀（默认情况下“资产”）。
- "-o | --override" 如果它存在删除目录。
- "-s | --assets" 用场景资源指定目录。
- "-t | --engine-type"（可选）写在配置文件中的引擎类型。

移除项目

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

删除的项目。被删除的目录是从项目配置文件检索。

应用程序升级新版本的 SDK

当升级新的 SDK 版本往往两个问题就出来了：

1. 引擎的新和旧版本的模块不匹配。
2. 旧的和新的引擎 API 不匹配。

为了更新应用程序的开发者版本导入模块列表转到项目源代码目录 apps_dev/my_project 和执行模块列表生成脚本：

```
python3 ../../scripts/mod_list.py
```

对于 MS Windows 用户：

```
python ..\..\scripts\mod_list.py
```

注解：要运行的 Python 3.x 的需要安装在你的系统中的脚本。

控制台将打印模块列表 - 将它们复制并粘贴到主 HTML 文件：

```
<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>
```

为了消除 API 不兼容性可能需要重构您的应用程序。所有的改变放在了[发行说明](#).

场景预览器

目录

- 场景预览器
 - 导航
 - 侧面板
 - * 信息面板
 - * 基本控制按钮
 - * 其他面板
 - 工具和调试
 - 动画
 - NLA
 - Shape Keys
 - 材质
 - 光照
 - 环境
 - 阴影
 - 天空
 - 海洋
 - 雾场
 - 风
 - 风弯曲
 - 屏幕空间环境光遮蔽
 - 上帝光芒
 - 泛光效果
 - 运动模糊
 - 发光材质
 - 景深
 - 颜色校正
 - 音频
 - 立体视图
 - 输入设备
 - 指标

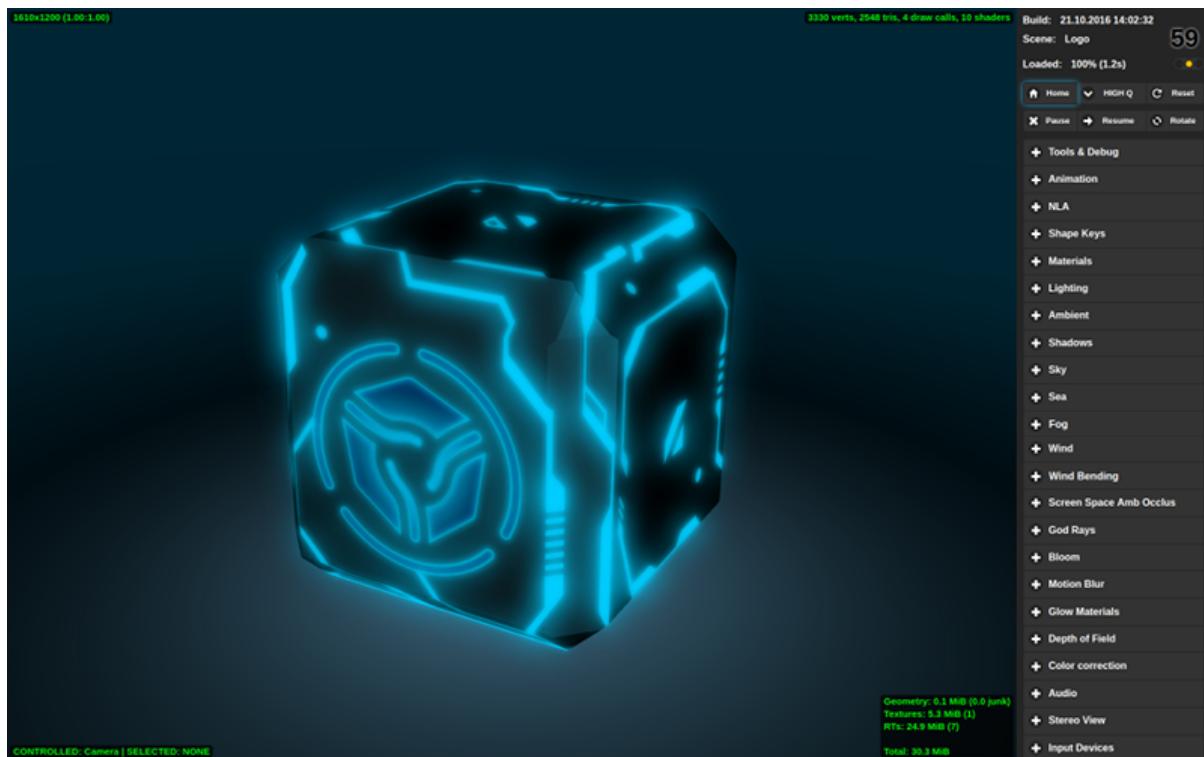
运行场景预览器.

导航

为了控制摄像机按住鼠标按钮并移动鼠标。也可以控制使用进行 W, A, S, D, R, F 键：前进，左，后，右，上，下。箭头和 numpad 键也可以使用。在 Target 相机模式有可能集中使用所选物体 Z 或 .(dot) 键。

侧面板

侧面板由三个区域组成：信息板，基本控制按钮和下拉面板与功能区分额外的控制元素列表。



控制元件列表中上到下依次

信息面板

建立 引擎建立日期和时间。在开发者版本显示的页面加载时间。

场景 加载的场景的名称。鼠标悬停时文件路径弹出。

加载 加载进度和时间。

基本控制按钮

首页 用 Fast Preview 按钮重载 Blender 导出的最新场景，或者，如果场景浏览器尚未从 Blender 运行，默认场景被加载。

重置 此按钮删除上次查看场景的保存名称和重新加载页面回显示默认场景。

LOW Q - HIGH Q - ULTRA Q 下拉菜单中选择了引擎的显示质量配置方式。

参见:

[品质配置](#)

暂停 暂停渲染。

恢复 恢复渲染。

旋转 启用和禁用相机围绕加载的场景。默认情况下禁用。

其他面板

工具和调试

此面板包含了一系列调试工具，其中包括：

概观外轮廓模式 打开此选项使所有场景中的物体可选择 并支持要在选择他们时候出现轮廓动画。此功能默认情况下启用。

自动浏览模式 按下这个按钮会导致场景查看器将 Scenes 列表的每一个场景在一秒钟内加载。此功能可用于检查控制台错误所有可用的内容。

画布分辨率因子 该参数可用于增加画布的分辨率。该值可以从 1 到 10 改变，默认值是 1.0（画布分辨率等于场景查看器窗口的物理分辨率）。

注解: 画布可用的最高分辨率可以通过 Web 浏览器来限制。

截屏 这个按钮可以用来把场景浏览器的窗口的屏幕截图。截图保存在 Web 浏览器用来存放下载的文件相同的文件夹。

调试视图模式 可用于调试目的不同的渲染模式的列表。可用模式包括：

NONE - 正常渲染模式。这是默认模式。

不透明 WIREFRAME - 将场景渲染成平白色带较深线框。用于渲染线框的颜色可以通过使用 Wireframe Edge Color 参数进行调整。

透明 WIREFRAME - 场景渲染为正常模式，但线框都在它的上面添加。像在前面的模式中，用于渲染线框的颜色可以通过使用 Wireframe Edge Color 参数进行调整。

正面背面 VIEW - 多边形的正面和背面渲染成不同颜色。

BOUNDINGS - 这种模式增加了线框球来罩着**物体** 和**网格**。球体可以是蓝色的（静态物体/目）或红色（动态物体）。

CLUSTER VIEW - 在这个视图模式，每个物体在特定的集群中（用在batching）用特定的颜色渲染。

BATCH VIEW - 在这种视图模式，每一批呈现视图使用一种特定的（平）颜色。在呈现时使用的颜色可以更改为随机生成的颜色，通过单击更改颜色按钮。

RENDER TIME - 场景是场景的不同部分的复杂性用不同颜色来渲染。最轻的部分使用绿色，较重 - 橙色色调，场景中非常消耗资源的部分用红色。

修改颜色 此按钮随机更改在 CLUSTER VIEW 和 BATCH VIEW 模式中使用的颜色。

在下列情况下此功能非常有用，当由引擎随机挑选的颜色太相似了，使它难以区分图像的部分。

渲染时间阈值（毫秒） 此参数为 渲染 TIME 视图模式设定一个“threshold”。如果引擎在一定的物体比设置的这个值花费更多的时间渲染的时候，这个物体会被染成红色。默认值是 1.5 毫秒。

线框边颜色 此参数允许你改变在 OPAQUE WIREFRAME 和 TRANSPARENT WIREFRAME 视图模式下使用线框边的颜色。

HUD 信息

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/>						
TOTAL ACTIVE				7 of	9	3.637

此按钮显示渲染子场景及其参数的列表，包含：

SCENE 当前场景的名称。

Active 显示子场景是否处于活动状态。

Subscene 子场景的名称。

Size 渲染的子场景的大小。

RenderCalls 在子场景中绘制的调用次数。

Time 引擎花费在渲染子场景的时间量。

TOTAL ACTIVE 所有子场景的绘制调用数和引擎花费在渲染它们的时间。

启用此参数可能会降低场景的性能。此选项默认为禁用。

GL Debug 该参数，当被激活时，使引擎处理低级错误。这可以显著降低性能。当你打开场景浏览器的第一次时它被禁用，设置后，应用程序将保存此选项的状态供下一次使用。

最小功能模式 此参数使场景查看器运行加载场景就好像是在一个低端系统上运行（在移动设备上，例如）。这使得艺术家，可以找出哪些材质场景是在低端配置可能无法正常工作。默认情况下禁用。

动画

动画控件。当观看动画模型时可以：选择一个物体，并使用下拉菜单切换的动画，循环切换动画模式，停止播放动画，设置所需的帧（动画应该被停止）。

可用参数：

Play All 按下此按钮将播放所有可用的动画。

Stop All 按下此按钮将停止当前正在播放的所有动画。

Object 场景中有动画的物体的列表。

动画槽 可用应用到物体上的动画的所有动画槽的列表。

动画 可供选择的物体使用的所有动画的列表。

循环 如果启用此参数，动画将重复播放。如果不是，动画将被播放一次后停止。默认情况下启用。

Range 这个值示出了所选择的动画的长度。

Status 这个值示出了动画的状态，即，是它正在播放或停止。

Playing Animation > Play 播放选定的动画。

Playing Animation > Stop 停止选定的动画。

Set Frame 显示出了动画的当前帧和可用于设置当前帧。

Skeletal Mix Factor 不同的骨骼动画混合在一起的因子。默认值是 1.0。

NLA

控制非线性动画的工具（如果它存在于场景）。在这个面板中工作，NLA 选项应该在场景的设置中启用。

可用参数：

Range 显示的 NLA 带的开始和结束帧。

Status 显示 NLA 动画的状态，即它正在播放或停止。

Playing NLA > Play 此按钮可用于播放 NLA 动画。

Playing NLA > Stop 这个按钮可以用来停止当前正在播放的 NLA 动画。

Set Frame 如果动画在此刻被停止，该参数可用于设置 NLA 动画的特定帧。

Shape Keys

形变键控制。当查看的模型有形变键，您可以选择一个物体，物体上的形变键和键对物体的影响。

可用参数：

Object 这个列表中包含场景中具有形变键的所有的物体。

形变关键帧 这个清单包含连接到一个选定的物体的所有形变键。

设置值 此参数设置的一个物体选定的形变键的影响程度。该值可以从零到 1 变化。

材质

材质属性设置。可以使用下拉菜单中选择一种材质。现在，这个功能只支持为数有限的属性和只有堆栈（非节点）材质应用于动态物体（节点材质和静态物体不支持）。

可用参数：

材质 目前在一个场景中的所有堆材质的清单。

Color 该参数定义一个选定的材质的基础的颜色。

反射率 该参数设定所选择的材质的反射率因子。它的值默认等于零。

菲涅尔 为反射菲涅耳效应的幕值。默认设置为零。

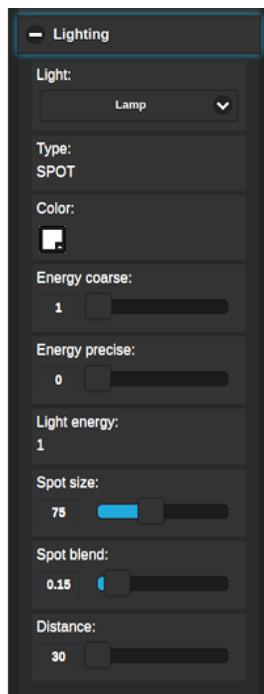
菲涅尔因子 菲尼尔 1 效果的一个因子。默认设置为零。

视差缩放 视差效果的缩放参数。这个值可以从零到 0.1 变化。默认值为 0。

视差步幅 用于构建视差表面的步数。值可以从 1 至 30 变化，默认值是 5。

光照

设置直接照明参数。可以使用下拉菜单中选择一种光源。支持更改颜色和强度。白天，太阳光照参数也可以在此面板上调整。



可用参数:

光源 这个列表包含当前场景中的所有光源。

类型 选择到的光源的类型。此参数不能在场景查看器来改变。

Color 所选光源基准颜色。默认情况下，使用了白色 (1; 1; 1)。

粗能量 此参数设置为光源的能量值的整数部分。这个参数的值可以从零到 100 变化，
默认值是 1。

精确能量 此参数设置为光源的能量值的小数部分。该参数的值可从零到 1 变化。默认
设置为零。

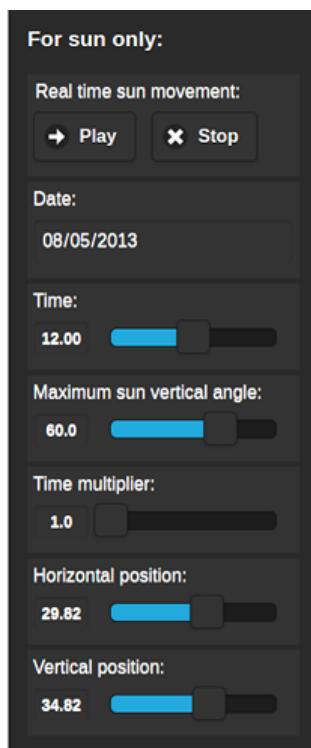
光能 此值的计算作为粗能量和精确能量参数的总和。如果他们都有默认值，它将会是
等于 1。

聚光尺寸 由光源发射的光束的角度。默认值是 75. 此参数仅适用于 Spot 型光源。

聚光混合 这定义了聚光灯边缘的柔和度。默认值为 0.15。此参数也是只可供“Spot”类
型的光源。

距离 此参数设置从其中由光源发出的光到只有由 Light Energy 定义的一半明亮的值的
距离 (米)。默认设置为 30。

以下参数仅适用于 Sun 式光源:



实时太阳运动 > 播放 按下此按钮激活实时太阳运动。

实时太阳运动 > 停止 按下此按钮停止实时太阳运动。

日期 此参数设置为当前日期。

Time 此参数设置一天的当前时间。默认值为 12.0。

最大太阳垂直转角 此参数设置 Sun 型光源的最大可能的垂直角度。默认值是 60。

时间倍数 此参数的值定义的时间流动的速度。默认值是 1.0 (时间以相同的速度，跟在现实生活中的流动相同)。

水平位置 在天穹中太阳的水平位置。默认值是 29.82。

垂直位置 在天穹中太阳的垂直位置。默认值是 34.82。

环境

环境照明参数的设置。支持一个半球的环境模型的颜色和强度的更改。

可用于环境照明设置跟在 Blender 中设置一样，在[专门章节](#) 描述。

阴影

设置阴影参数，包括阴影级联和阴影边缘柔化参数。

在手册的[相应章节](#) 描述了设置阴影选项

天空

为动态天空设置参数，如颜色、太阳光散射参数等。

场景浏览器能够访问用于设置所有 Blender 的程序天空参数。参数在[相应章节](#) 描述。

海洋

设置水渲染参数，包括通过深度和距离岸边的颜色过渡，泡沫和次表面散射参数设置、波动力学等。

[室外渲染](#) 章节描述了用于水渲染的设置。场景查看器使用跟 Blend4Web 引擎本身相同的设置。

雾场

设置雾参数。

场景查看器使用标准的雾场渲染设定，描述在[相应章节](#).

风

设置风参数，包括方向和强度。

场景浏览器所使用的的和在 Blender 中使用的是相同的参数。参数在[相应章节](#) 描述。

风弯曲

设置风弯曲效果参数。

同一套风弯曲参数在 Blender 中也可用。参考[室外渲染](#) 一章。

屏幕空间环境光遮蔽

环境光遮蔽参数的设置。

所有参数设置在[后期效果](#) 章节.

上帝光芒

黄昏的光线效果参数设置。

所有参数设置在[相应](#) 章节.

泛光效果

泛光效果参数设置。

场景浏览器所使用的的和在 Blender 中使用的是相同的参数。参数在[相应章节](#) 描述。

运动模糊

运动模糊效果设置。

场景浏览器所使用的的和在 Blender 中使用的是相同的参数。参数在[后期效果](#) 描述。

发光材质

发光材质的效果设置。

所有参数设置在[相应](#) 章节.

景深

设置景深效果的参数

所有参数设置在[相应](#) 章节.

颜色校正

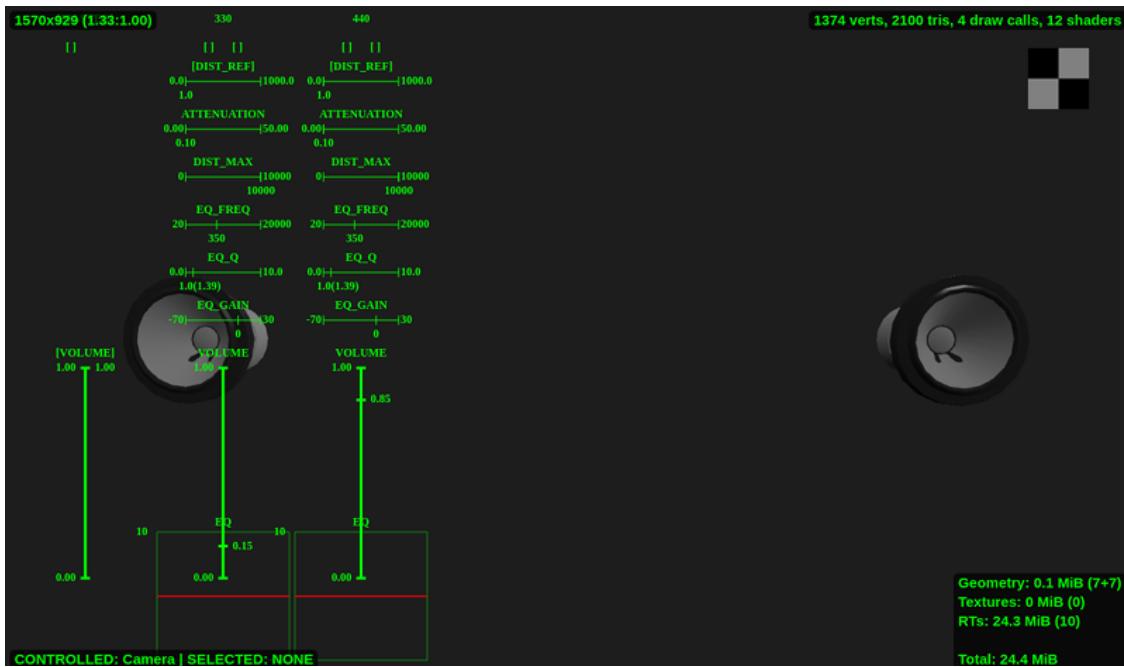
颜色校正参数，包括亮度、对比度、曝光和饱和度的设置。

场景浏览器所使用的的和在 Blender 中使用的是相同的参数。参数在[颜色管理](#) 描述。

音频

在面板上的混合模式的切换。启用该功能后，混频器的界面变得可见（仅适用于具有声源的场景）。

混合器 使显示各种（包括音量、频率、距离、衰减等）场景中的参数，发言者的均衡器，并允许用户通过使用数字键盘调整这些参数。更详细地介绍了可用的参数和可能性在[声音篇](#) 的相应章节



此功能默认是关闭的。

立体视图

有在面板上的立体模式的切换。

可用参数：

立体类型 列表中的所有可用的立体渲染模式。现在来说，支持三种类型 [F](#)

NONE - 不使用立体渲染。

ANAGLYPH - 立体图 渲染被使用。

HMD - 用于头戴式显示设备 渲染。如果系统具有访问的 HMD 设备此选项才是可用的。

默认设置为 NONE 。

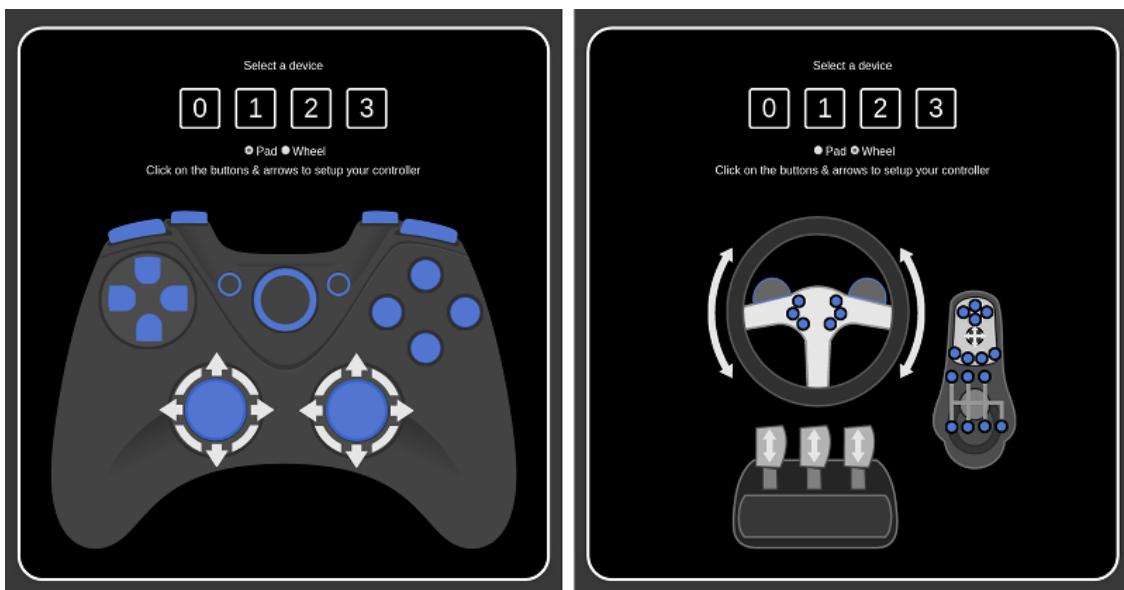
HMD 设置 显示在HMD 设置窗口.

输入设备

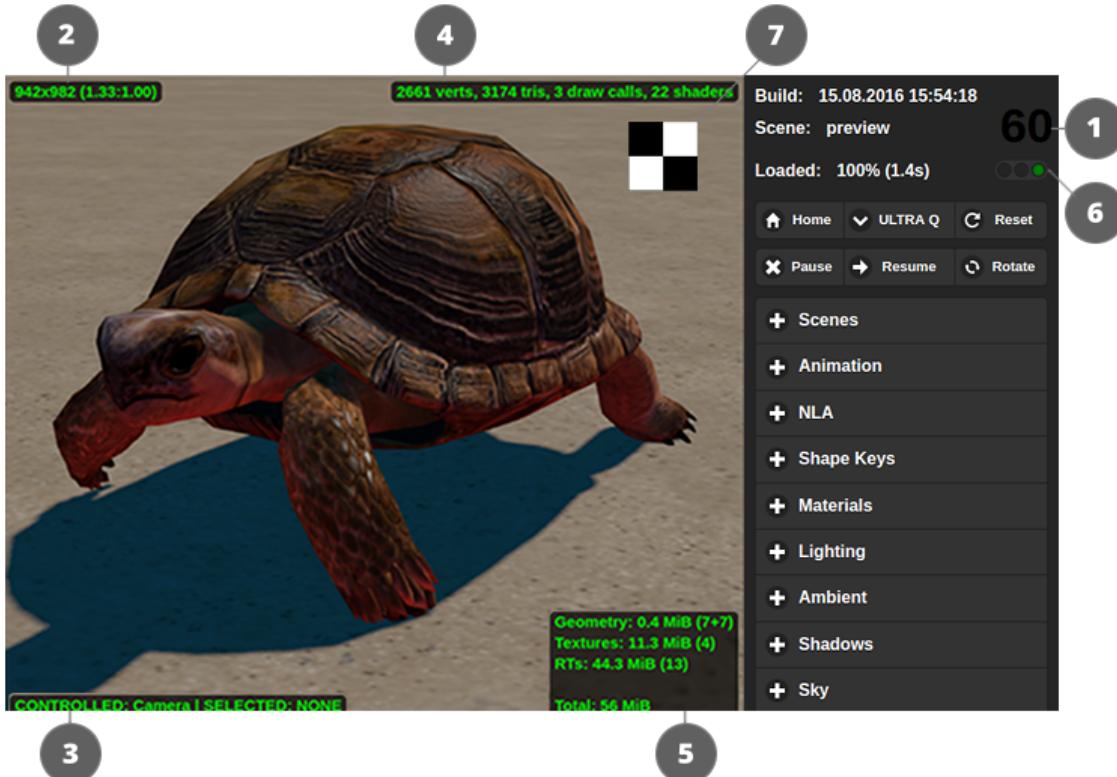
此面板包含用于设置输入设备的工具。

使用陀螺 使在应用程序中使用陀螺仪。默认情况下禁用。

游戏手柄设置 单击此按钮显示了游戏手柄的设置窗口。



指标



- 1. 每秒计数帧** 这是位于右上角。这显示最后 1.5 秒的平均和四舍五入的值。
- 2. 视维** 这是位于左上角。它显示像素视口的尺寸。
- 3. 选定的物体和控制物体** 位于左下角。它显示选择和控制物体的名称。物体选择可以用鼠标来执行。要直接控制的物体（通常为物理调试）按 Q 键并单击物体。物体的移动通过 W, A, S, D 键执行。要退出控制模式下，按 Q 键并单击一个空白的地方。该标示也显示所选物体的距离用 Blender 单位（相当于米）。
- 4. 场景复杂度指标** 位于渲染区域的右上角。它显示顶点的数目，三角形，WEBGL 绘制调用，并在主渲染场景中使用的着色器（阴影渲染调用不包括，例如）。
- 5. 显存指标** 位于渲染区域的右下角。它显示视频存储器的使用的几何形状，纹理，渲染目标的量，并且还总的内存使用情况。
- 6. 场景负载误差指标** 位于 FPS 计数器下。它显示在场景加载期间出现的错误和警告。红灯表示错误，黄色-警告和绿色表示场景已成功加载。

装载错误和警告可以在 Web 浏览器的控制台中查看。

7. **背景透明度指标** 位于屏幕的右上角，场景复杂度指标的右下面。这是一个调试工具，可用于以发现一个场景的背景是透明或没有。它显示为一个黑色和白色棋盘图案的小方块。

只有当该应用程序的背景为透明时这一指标是可见的。显示在上面的图片只给读者的总体思路，它将如何工作，要根据实际情况。

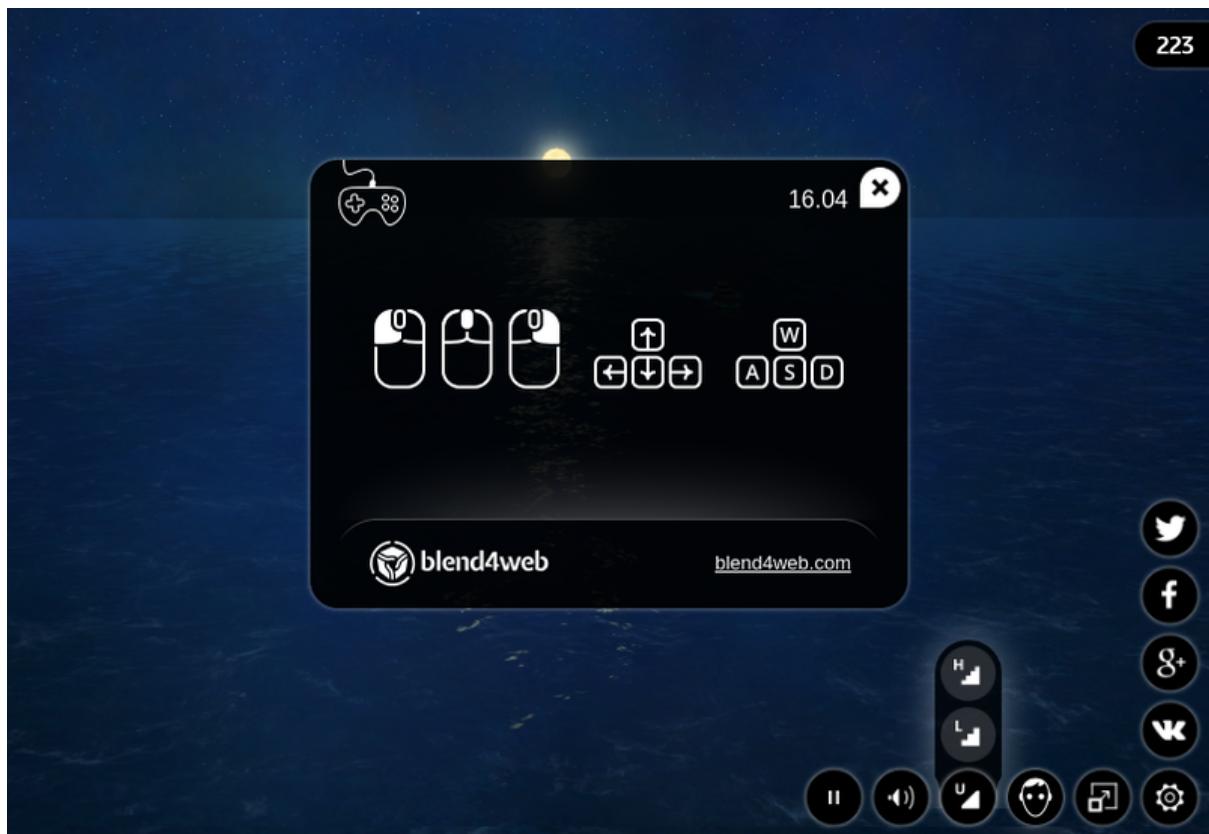
网络播放器

目录

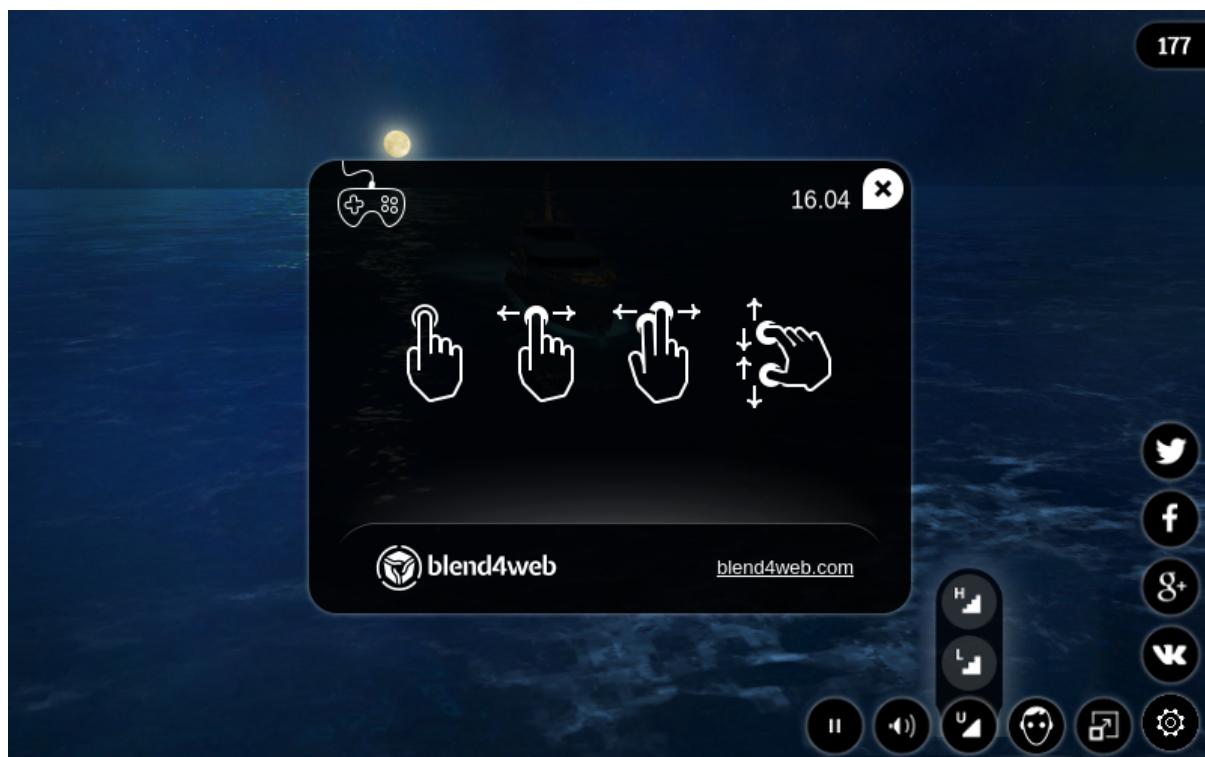
- 网络播放器
 - 用法
 - 导航
 - 控制面板
 - 属性
 - 场景名称为标题
 - 场景错误

网络播放器是演示模式的特殊应用可以渲染模型和场景。

桌面版本：



手机版：



用法

您可以复制包含网络播放文件的目录，即 `deploy/apps/webplayer`, 从 Blend4Web SDK 分布和您的网站进行部署。您可以将导出的场景文件在您的网站和指定的路径给他们（绝对或相对）与 `load` 网络播放器的参数。

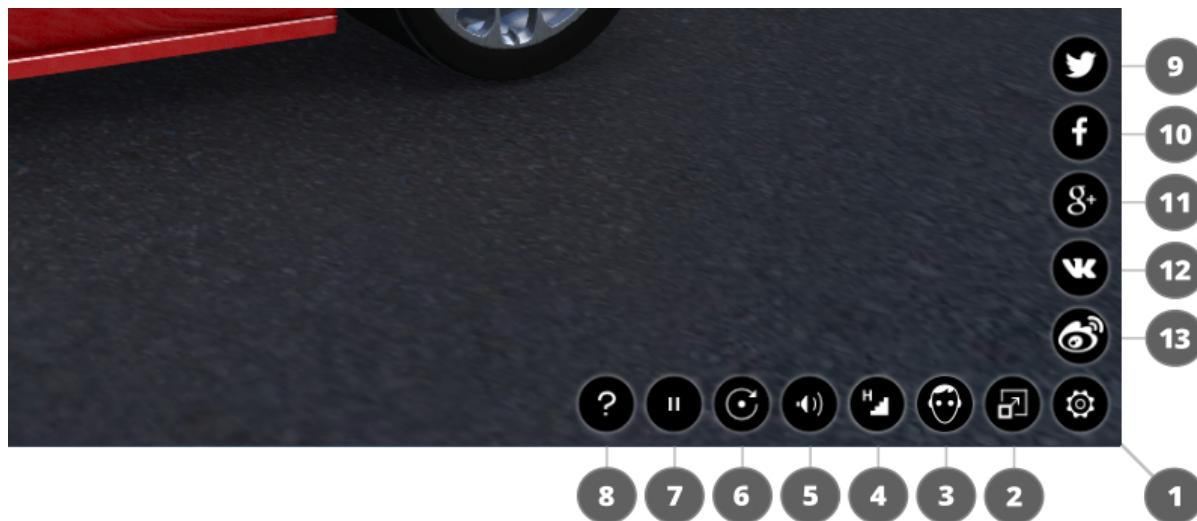
当您导出成一个 HTML 文件的 Web 播放器界面会自动集成到它。

导航

相机（在 Target 和 Eye 模式）由鼠标控制，它的按钮按下或键：W, A, S, D, R, F (向前, 左, 后, 右, 上, 下). numpad 键也支持。

控制面板

网络播放器的控制面板如下图所示。



1. 显示/隐藏控制面板;

2. 开/关全屏模式;

3. 立体渲染开/关;

4. 设置场景的质量;

5. 声音开/关;

6. 开启/关闭相机自动旋转模式;

7. 运行/停止引擎;

8. 打开帮助窗口;

9. 通过推特分享;

10. 通过 Facebook 分享;

11. 通过 Google+ 分享;

12. 通过 VK 分享。

13. 通过 Weibo 分享。

属性

网络播放器接收来自浏览器地址栏中的属性：



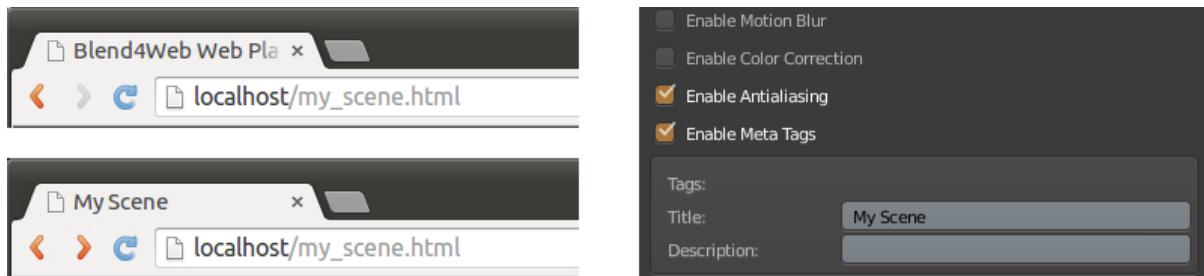
1. 特殊的属性 load 用于加载场景，这个属性包含相对路径 JSON 文件。
2. 在发生错误的 WebGL 的可选 fallback_image 属性用于设置的背景图像，而不是 3D 内容。
3. 在发生错误的 WebGL 的可选 fallback_video 属性用于设置背景视频，而不是 3D 内容。可用于多次添加更多的视频格式。
4. 可选 show_fps 属性是用来显示在玩家的右上角的 FPS 计数器。
5. 可选参数 autorotate 时，使用了场景加载后使自动相机旋转。
6. 该 compressed_textures 可选参数用于启用加载精缩和压缩纹理（在 DDS 格式）。
7. 该 compressed_textures_pvr 可选参数用于启用加载压缩为 PVRTC 格式的材质，该参数用在 compressed_textures 参数中。
8. the compressed_gzip optional parameter is used to enable loading of GZIP compressed resources such as ".json.gz", ".bin.gz", ".dds.gz" and ".pvr.gz".
9. 可选 alpha 参数用于启用对描画区域的透明度。
10. 可选的 no_social 属性用于在控制面板上禁用社交网络按钮。
11. 可选的 socials 属性用于选择性地启用和禁用控制面板上的社交网络按钮。此属性应该像 socials=<...>, 其中 <...> 是一组与您要启用的社交网络按钮字母 (f 为 Facebook v 为 VK, t 是 Twitter 的, w 是微博和 g 是 Google)。字母顺序设置定义了按钮将出现在屏幕上的顺序。

注解： 如果同时 fallback_image 和 fallback_video 参数指定 fallback_image 参数被使

用。

场景名称为标题

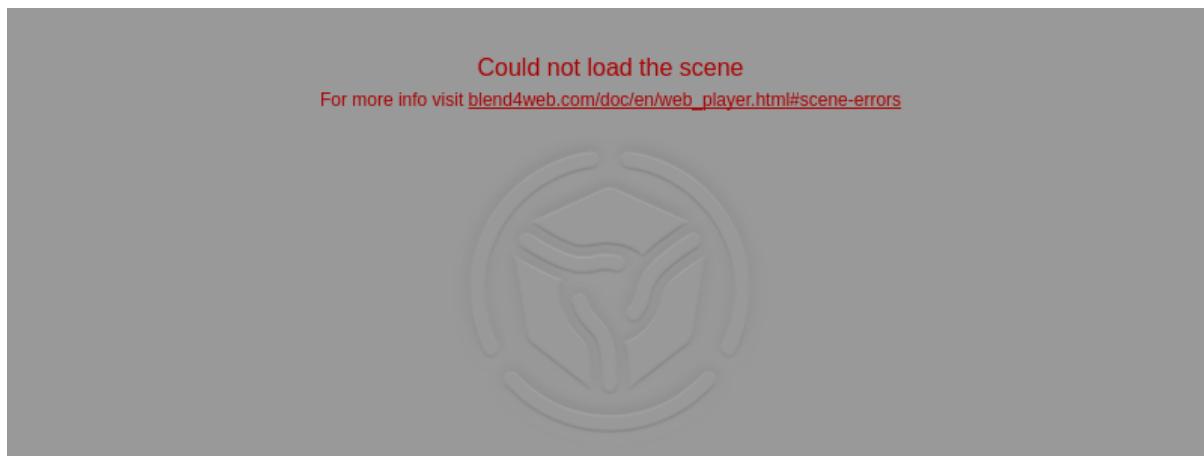
默认情况下，网络播放器具有 Blend4Web Web Player 称号。在 Blender 的场景中分配的 meta 标签 title ，您可以更改该值为别的东西。



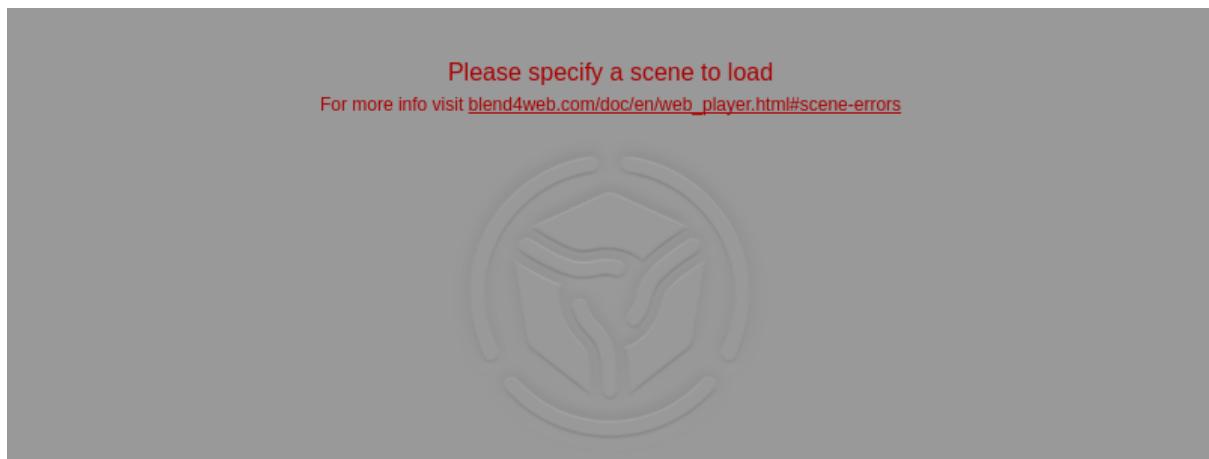
场景错误

如果播放器使用不当，它会显示相应的错误。

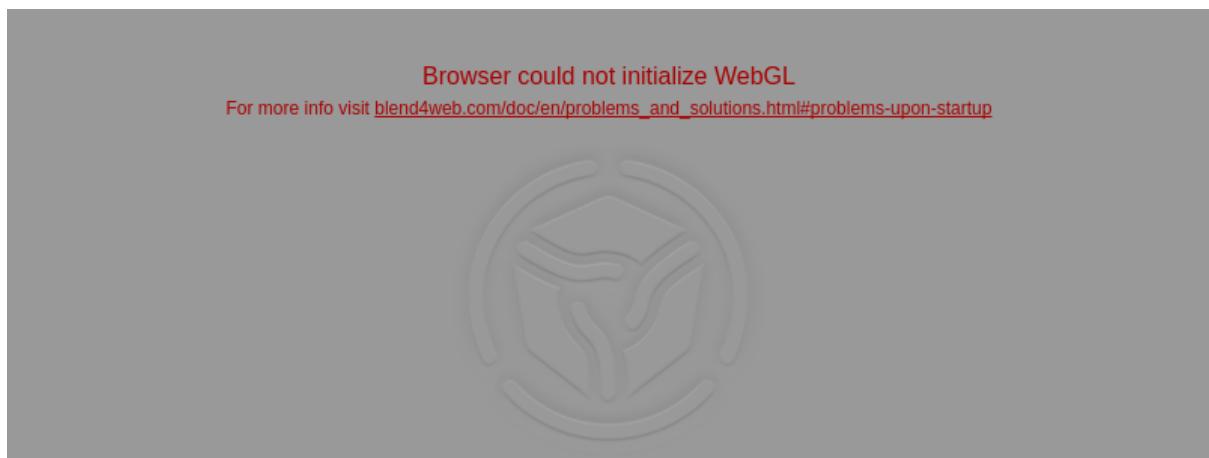
1. load 属性 指定了一个错误的路径指向 JSON 文件或文件已损坏;



2. load 属性 未找到或无效的;



3. WebGL 的初始化错误。请查看[这一页](#) 找到解决方法。



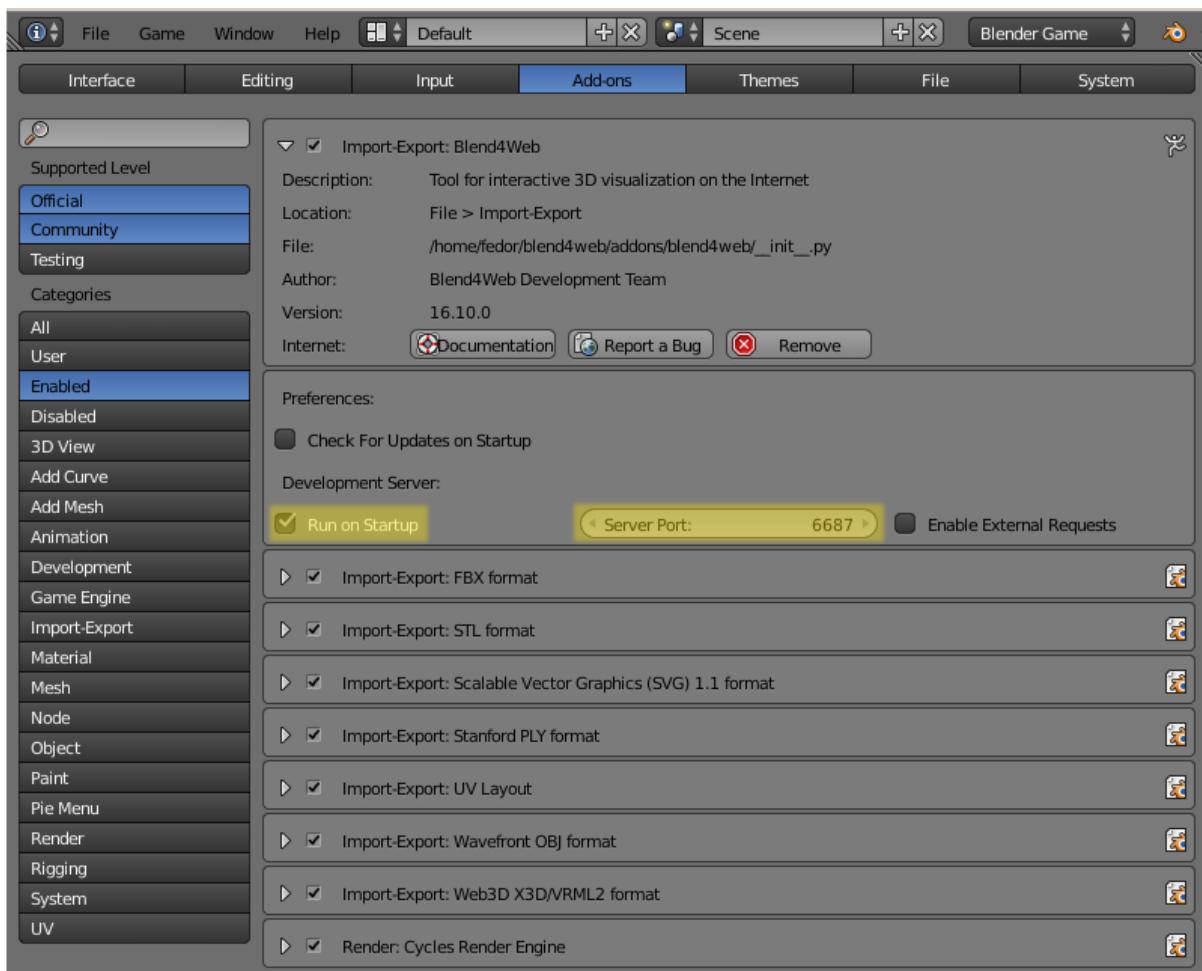
插件

目录

- 插件
 - 本地开发服务器
 - 运行浏览器和演示
 - 输出格式
 - * JSON
 - * HTML
 - 导出选项
 - 初始化错误
 - 兼容性错误
 - 关键导出错误
 - 非致命导出的错误
 - 其他消息
 - 插件翻译

本地开发服务器

本地开发服务器的设置中可以在 File > User Preferences... 找到 (快捷键 Ctrl-Alt-U)。在这里，您可以更改端口号被用来运行开发服务器（默认情况下 6687），可随 Blender 启动开启。要做到这一点，在插件上启用 Run on Startup 选项运行。



在改变了本地开发服务器设置后需要重新启动 Blender。

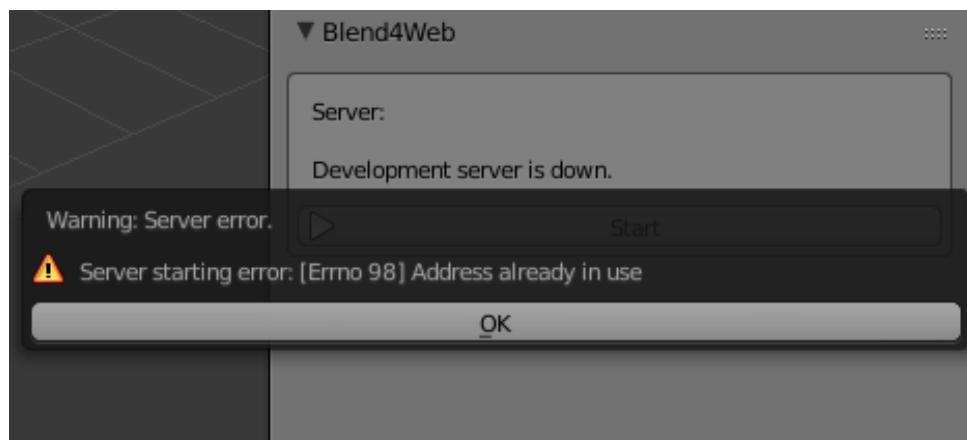
有可能为本地服务器来处理外部请求。要做到这一点启用 Enable External Requests 选项。

如果您选择不自动启动服务器，你总是可以手动启动：去 Render 选项卡，然后按 Development Server 面板上的 Start Server 按钮：



注解: 如果本地开发服务器无法启动。在这种情况下，相应的消息将被显示，而不是 Start Server 按钮。

如果服务器无法运行，将显示一条错误消息，说明了原因：



如果服务器端口已被一些其他的应用程序可能会出现此错误。

按 SDK Index 按钮，在浏览器中打开 Blend4Web SDK 的索引网页。本页面可在 <http://localhost:6687> 访问。



然后，对于您的操作系统的默认浏览器将启动。

Project Manager 按钮可用于打开[项目管理器](#).

Fast Preview 按钮加载当前正在 Blender 打开的场景到[查看器](#).

服务器可以通过按下 Stop Server 按钮被停止。当 Blender 被关闭也停止。

运行浏览器和演示

索引页包含用于启动场景浏览和演示应用程序的链接。需要一个[WebGL-适配浏览器](#) 运行这些应用程序。

Blend4Web SDK Pro Free

WebGL: available

Development Server: available

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

注解: 如果 SDK 应用程序显示不正确, 或显示错误消息, 请按照启动时的问题部分中的说明。

输出格式

场景制作完成后, 需要将其转换成由 Blend4Web 引擎所支持的格式。

目前, 两种格式都支持: JSON 和 HTML。

JSON

导出场景到这种格式会创建一个 `.json` (JavaScript 对象符号) 文件，其中包含所有导出的数据结构和链接到外部资源 (图片，声音等等)，还有一个 `.bin` 文件包含二进制格式的模型数据阵列。

如果媒体资源被 [打包到`.blend`文件](#)，它们会在导出过程中解压，并在项目的目录放置在隐藏的 `../.tmp/` 文件夹中 (SDK 目录内)。应当注意的是，这样的话文件的名称将被自动地改变，工作的时候可能变得复杂。

建议将所有资源存储在 SDK 目录中一个专门的文件夹，例如，它可以被称为 `projects/project_name/assets`。

注解：按照安全因素考虑，开发服务器只能访问 SDK 文件夹。如果媒体数据被放置在另一个文件夹，服务器将无法在部署的时候导出（即使是在 Blender 内工作正常）它。

外部资源路径应该是相对的。如果不是这种情况，执行 File > External Data > Make All Paths Relative 命令，否则在其他电脑上打开这个项目会有问题。

这是一个复杂的项目，其中包括多个场景，并且需要 JavaScript 编程的主要格式。对应的部分项目开发进一步说明在[corresponding section](#).

HTML

导出场景到这种格式会打包场景的所有的资源投入到一个文件带 HTML 扩展。这个 HTML 文件不仅包含场景本身，而且还材质，声音 Blend4Web 引擎和标准[web player](#). 像这样的文件可以在任何计算机和有支持 WebGL 的 Web 浏览器的任何移动设备上执行。

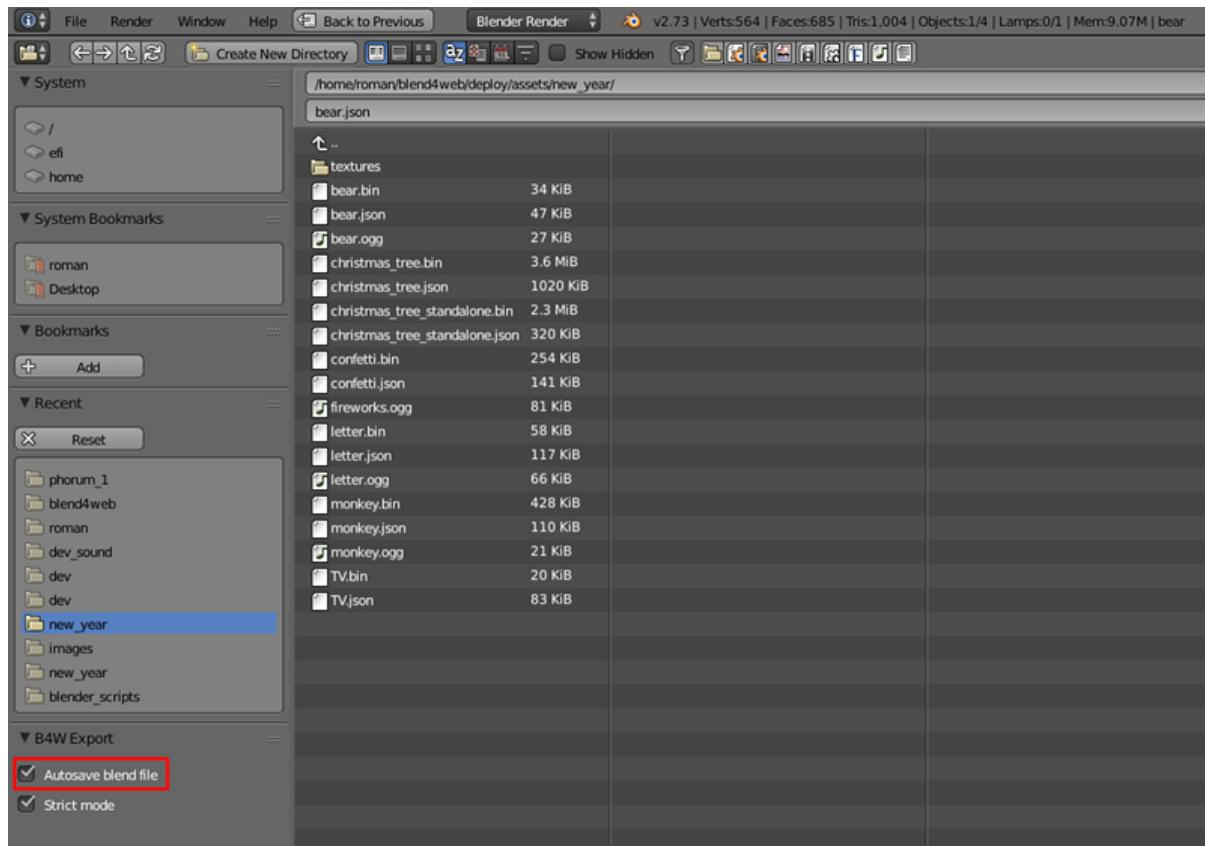
不能使用 HTML 文件做进一步的开发，但你也并不需要任何额外的操作来运行它们。这种格式对开发中等规模的相对简单的应用非常有用。

应当指出，不论如何，HTML 应用程序不支持以下功能：

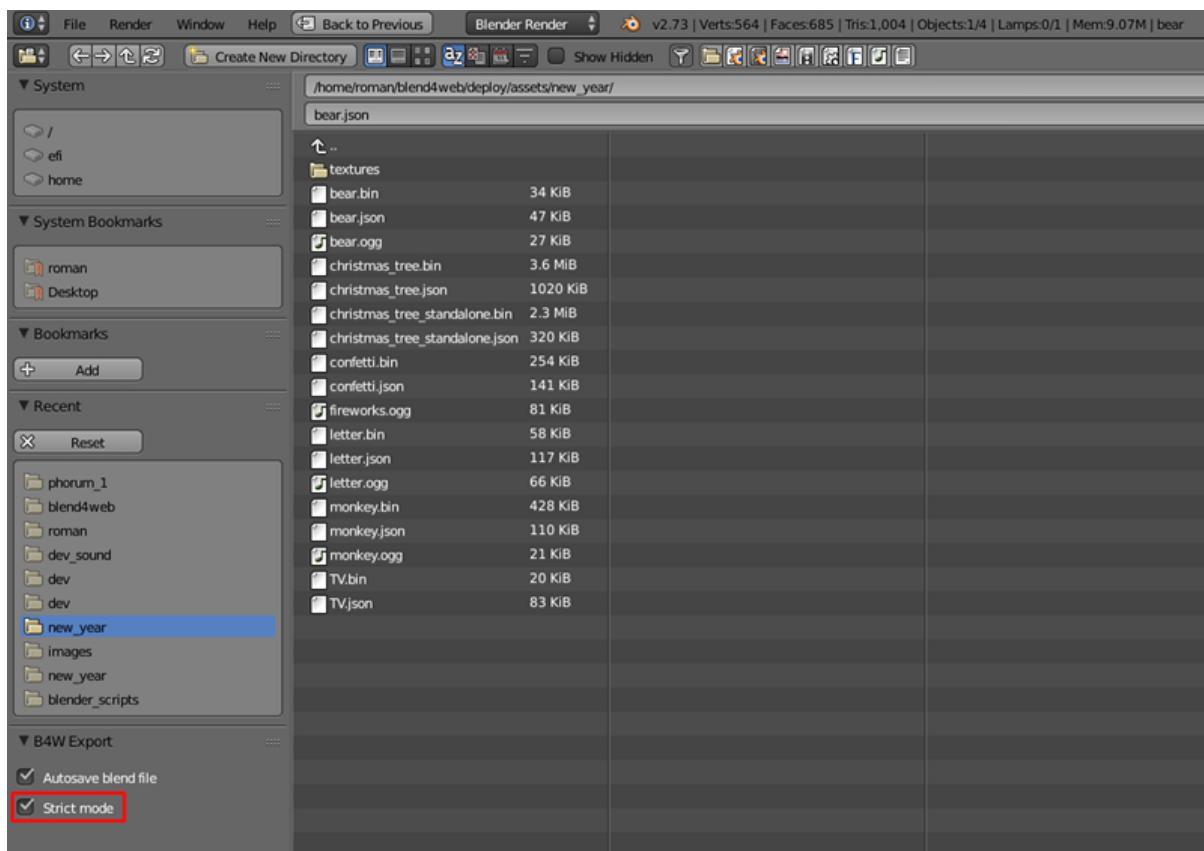
- 物理
- DDS 纹理
- min50 纹理

导出选项

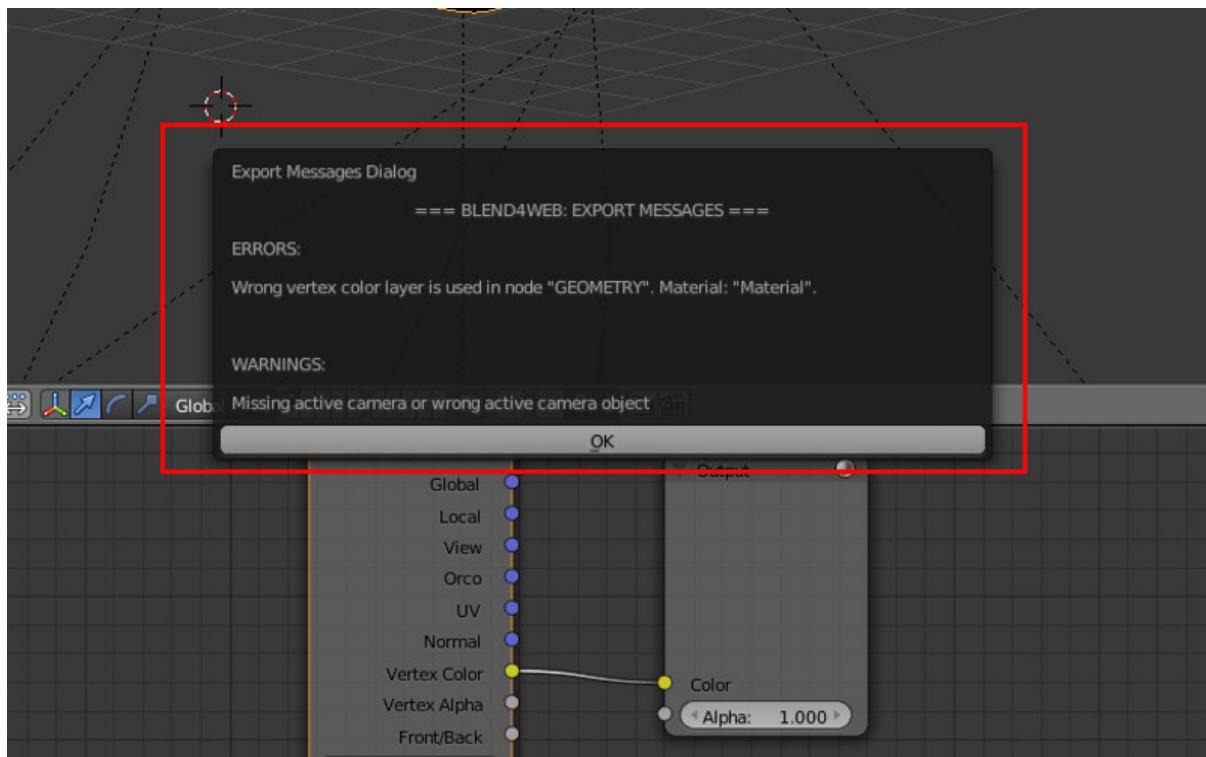
自动保存 Blend 文件 自动保存从导出的实际文件。默认情况下启用。自动保存正在执行的导出之后，以保证目前的混合文件和导出的文件内容之间的一致性。此外，相对路径导出的文件被保存方便。



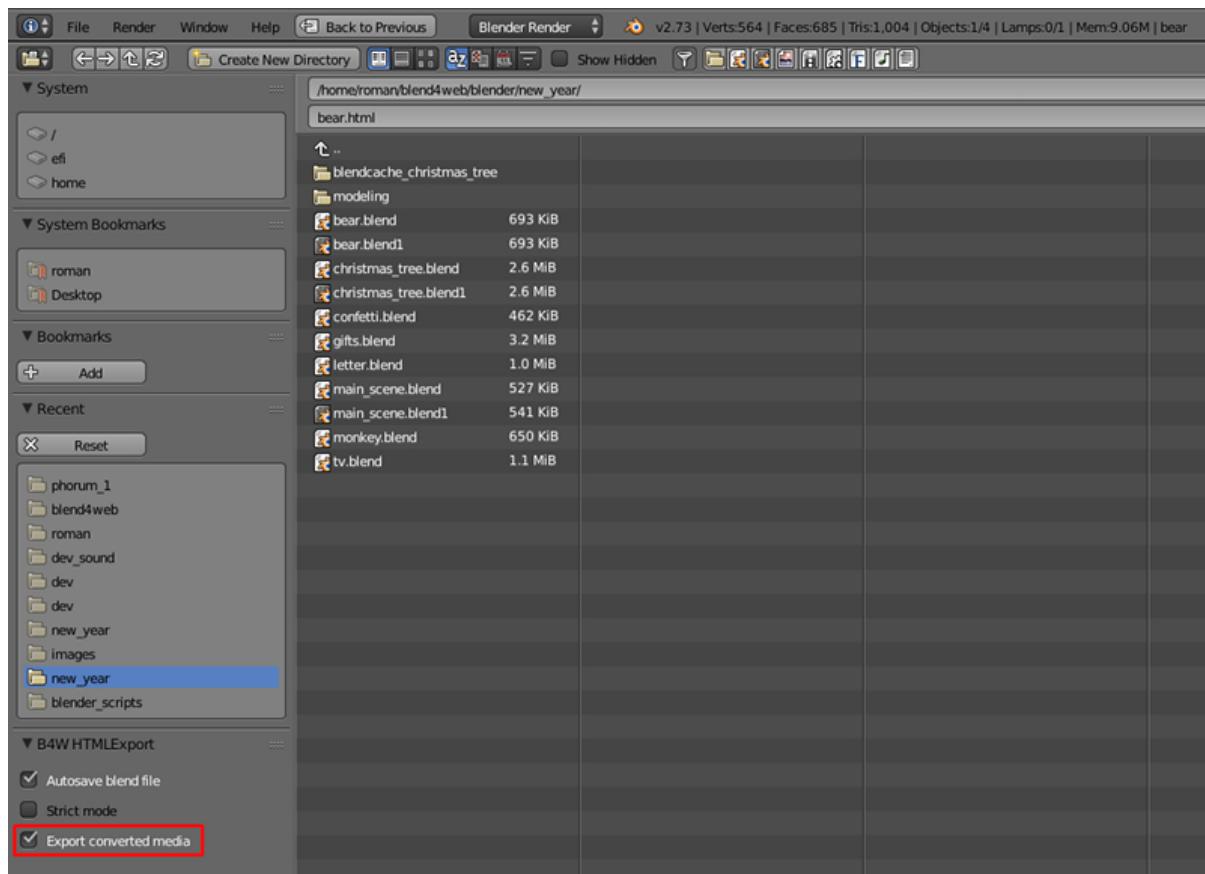
严格模式 这种模式可以防止输出是否有适用于用户的注意力的任何错误或消息。这种模式使能了 Strict Mode 在导出菜单设置：



如果有用户的注意力任何非关键性的错误或消息，一个对话窗口会显示如下：



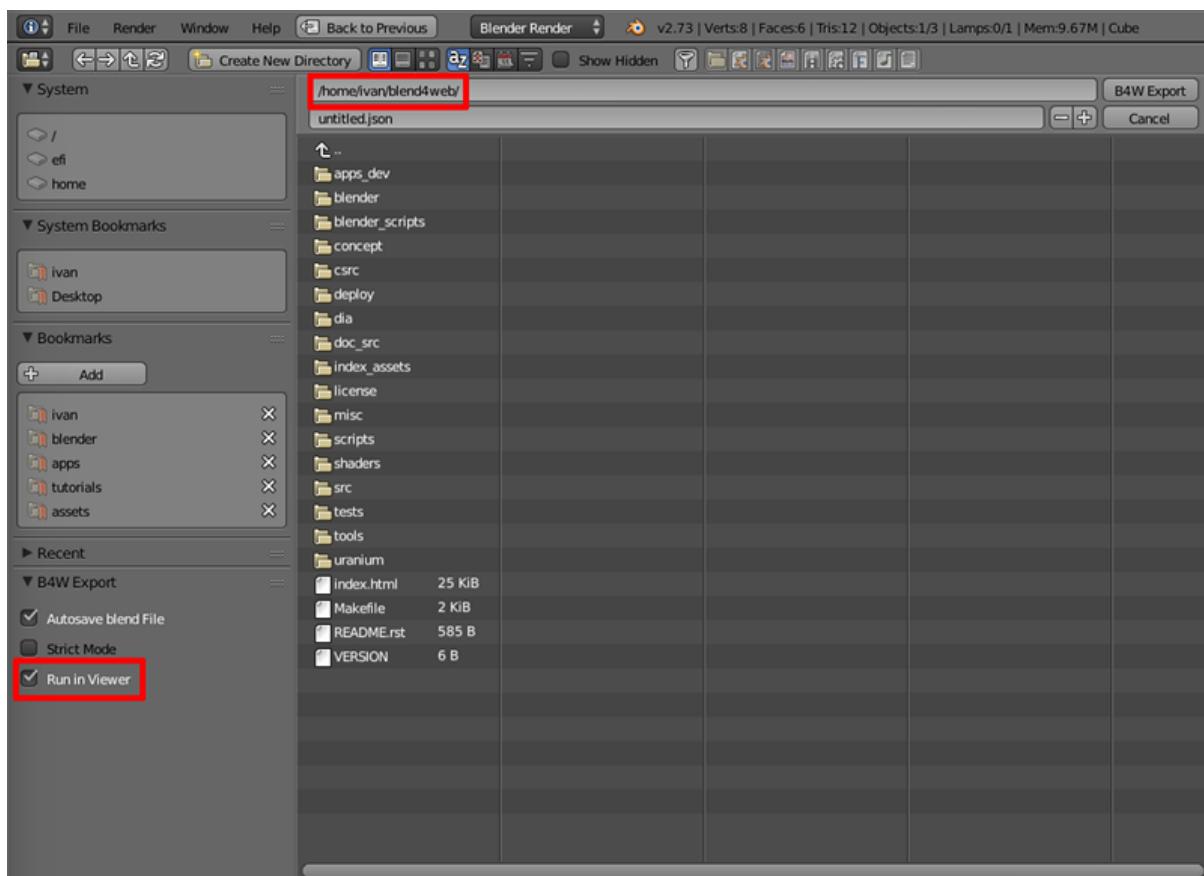
输出转换媒体 此选项可用于 HTML 导出。当启用该选项，不同格式的转换后 mediafiles 都写在 HTML 文件。使用不同的 mediafiles 是必不可少的，而使用 HTML 导出创造跨浏览器和跨平台的应用程序。这些文件可以由创建[转换器](#).



在预览器中运行 自动启动场景查看，导出的场景添加到它。

当使用[本地开发服务器](#)，可以在场景预览器中打开导出的.json文件。要做到这一点，选择在导出时 Blend4Web SDK 文件结构中的任何路径。

应该于 SDK 里面的目录。如果没有，这个选项将不会显示在菜单中。另外，如果本地开发服务器宿机也不会显示。



初始化错误

可能出现安装的加载项或当一个场景是在 Blender 打开时初始化错误。在这种情况下，一个对话窗口，显示错误的描述。



错误信息	原因
Blend4Web initialization error! Add-on is not compatible with the PLATFORM platform.	该 Blend4Web 插件和该平台不兼容。
Warning: Blender version mismatch. Blender VER_REQUIRED is recommended for the Blend4Web add-on. Current version is VER_CURRENT.	警告有关可能不符合当前 Blender 的版本号。它建议使用 VER_REQUIREDBlender 版本。当前的版本是 VER_CURRENT。
Incorrect add-on directory name.	插件目录的不正确的名称。在存档附加结构已损坏，或归档的全部被打乱。

兼容性错误

当试图在浏览器中查看一个场景，在下列情况下兼容性错误可能出现：如果版本的附加用于导出场景不同版本 Blend4Web 引擎，它尝试加载的场景，或者.bin 文件不对应于.json 文件。

相比版本的插件与该场景被导出引擎版本太旧。场景将不会被加载。我们建议您使用引擎和插件的最新版本。

错误信息	原因
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.	插件版本，与场景被导出版本相比太旧：VER_OLD。该引擎要求：VER_NEW。场景将不会被加载。我们建议您使用的加载项的最新版本重新导出场景。我们也建议使用引擎的最新版本。
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.	插件版本与场景的导出版本，是有点老了：VER_OLD。该引擎要求：VER_NEW。场景将被载入像往常一样，但是可能会出现一些错误。我们建议您使用的加载项的最新版本重新导出场景。我们也建议使用发动机的最新版本。
B4W engine version is too old relative to JSON. Can't load the scene. Update your engine version to fix it.	相比版本的插件与该场景被导出引擎版本太旧。场景将不会被加载。我们建议您使用引擎和插件的最新版本。
B4W engine version is a bit old relative to JSON. Some compatibility issues can occur. Update your engine version to fix it.	相比版本的插件与该场景被导出引擎版本是有点老了。场景将被载入像往常一样，但是可能会出现一些错误。我们建议您使用引擎和插件的最新版本。
BIN version does not match to JSON version: VER_BIN, required: VER_JSON. Couldn't load the scene. Reexport scene to fix it.	.bin 文件的版本太旧相对于.json 文件：VER_BIN, .json 文件版本VER_JSON。场景将不会被加载。我们建议您再导出该场景。
BIN version does not match to JSON version: VER_BIN, required: VER_JSON. Some compatibility issues can occur. Reexport scene to fix it.	.bin 文件的版本有点老相对于.json 文件：VER_BIN, .json 文件版本VER_JSON。可能会出现一些不兼容的错误。我们建议您再导出该场景。

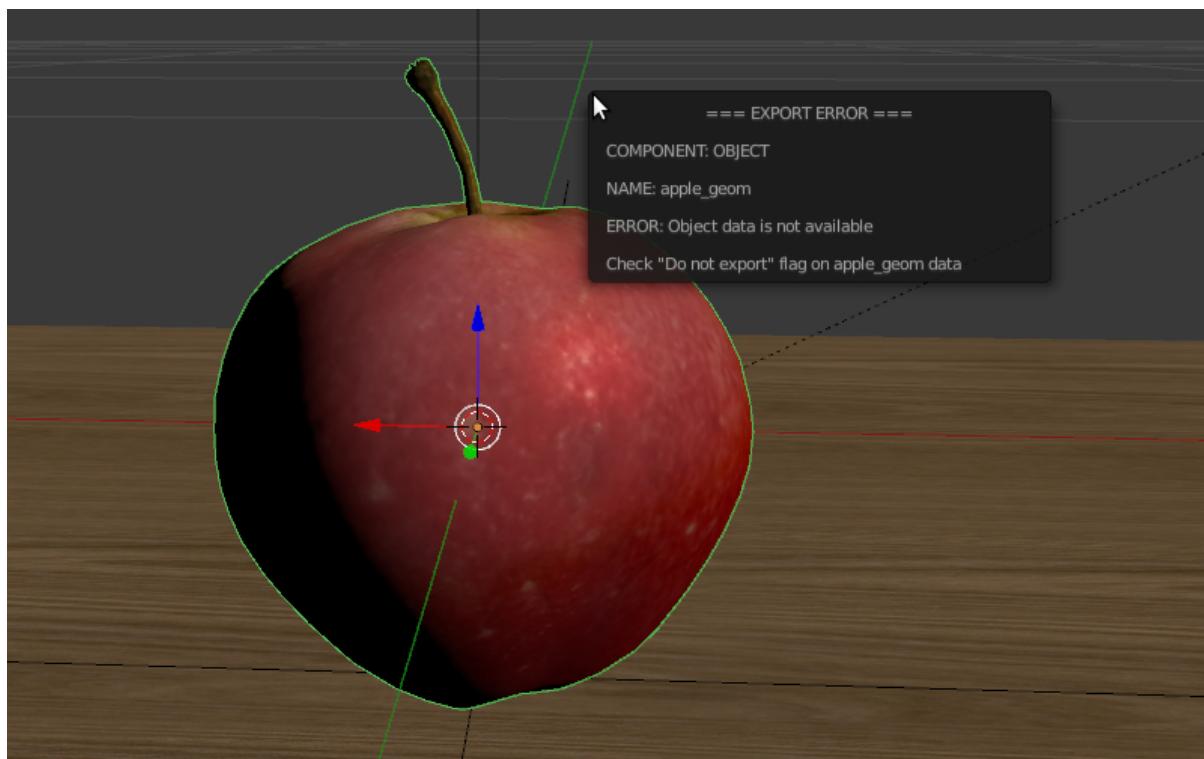
关键导出错误

在导出的情况下错误的 BLEND4WEB EXPORT ERROR 对话框出现描述该问题：

COMPONENT -组件（对象，网格，材质，纹理等），才造成了导出的错误的类型。

NAME -组件名称。

ERROR -在发生问题的简短描述。

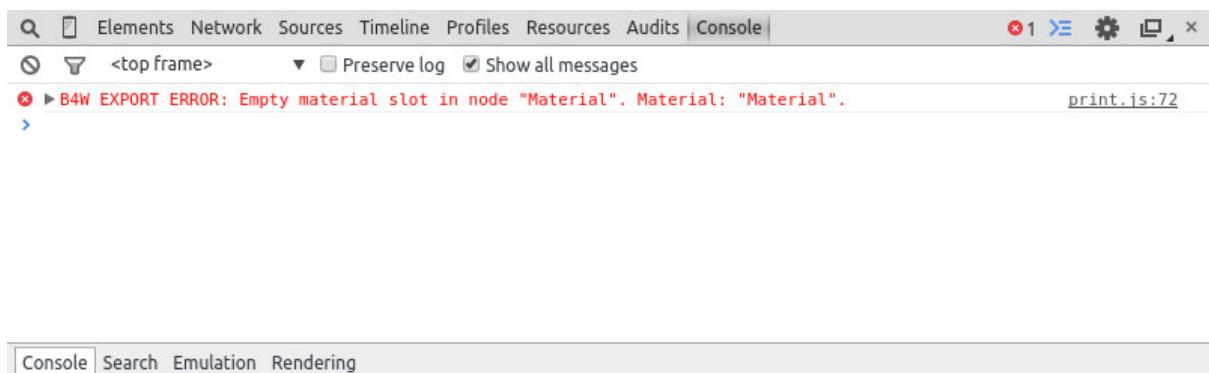


错误信息	原因
Export to different disk is forbidden	导出到位于不同的磁盘上的目录被禁止
Incompatible objects with a shared mesh; The OBJECT_NAME object has both vertex groups and a shared mesh	不兼容的对象有一个共享的网格。不允许导出的对象既具有共享网格和顶点组。例外：导出是可能的，如果一个对象具有 Apply modifiers, Export vertex animation, Export edited normals, Apply scale 选项打开（因为在这些情况下，网格的完全复制时）。
Incorrect mesh; Corrupted file: Wrong vertex color values	损坏的文件：不正确的顶点颜色值。
Loading of resources from different disk is forbidden	从不同的磁盘资源的装载是被禁止的。
The material has a normal map but doesn't have any material nodes	节点材质采用 Normal Mapping, 但没有 Material 节点。
The mesh has a UV map but has no exported material	该网格具有 UV 图层但并无材质导出。
The mesh has a vertex color layer but has no exported material	该网格具有顶点颜色层，但有无材质导出。
No such file or directory	文件或目录不存在。
Permission denied	没有访问权限到当前目录。
Wrong edited normals count; It doesn't match with the mesh vertices count	编辑法线的数目不匹配的网格顶点的数量。执行 Clean Up 或 Save 在 B4W Vertex Normals Editor 面板。
Wrong overridden bounding box; Check the mesh's bounding box values	错尺寸覆盖网格的时指定 BoundingBox: 最小值为大于尺寸中的至少一个最大值。

非致命导出的错误

在对比上述所列临界导出错误，这些错误不会禁止导出，但可以使显示不正确的场景。这些消息可以在浏览器的控制台中查看（与打开 F12）当一个场景被加载。该消息是这样的：

B4W EXPORT ERROR: Error message



错误信息	原因
Canvas texture ID NAME already exists. Texture NAME.	该 Canvas ID 已经存在。
Dupli group error for object OBJECT_NAME. Objects from the GROUP_NAME dupli group on the OBJECT_NAME object cannot be exported	其中没有被选定为重复的 OBJECT_NAME 对象的 GROUP_NAME 组中的对象可以被导 出。导出许可的组中的至少一个对象，或 删除的组的复制是必需的。
Empty canvas texture ID for texture NAME.	Canvas ID 是空的。
Empty material slot in node “NAME”. Material: “NAME”.	在“名称”节点空的材质插槽。
Environment map in the “NAME” world texture slot cannot be a movie.	环境贴图无法显示一个视频文件。
Ignoring LODs after empty LOD for the NAME object.	按照空槽，所有 LOD 对象被忽略（在 LOD 对象列表对象的名称）。
Incomplete mesh NAME; Dynamic grass vertex colors required by material settings	该 Dynamic grass size 和/或 Dynamic grass color 选择所使用的特殊地形材质， 但网格没有顶点着色这样的名字。
Incomplete mesh; Material settings require vertex colors	该 Vertex Color Paint 启用了网状材料的 选择，但网格没有顶点颜色层。
Incorrect NLA script, falling back to simple sequential NLA.	不正确的 NLA 脚本。返回到简单的序列 NLA。
Incorrect vertex animation for mesh NAME. Object has no vertex animation.	对象的顶点动画导出选项，但没有顶点 动画。
Incorrect vertex animation for mesh NAME. Unbaked “ANIM_NAME” vertex animation.	顶点动画导出是打开的网格，但 ANIM_NAME 动画没有任何帧。
Incorrect mesh NAME; Wrong group indices	该网格具有分配给不存在的顶点顶点组 的顶点。
Incorrect mesh NAME; Wrong vertex positions	损坏的文件：不正确的顶点坐标值。
Incorrect mesh NAME; Wrong normals	损坏的文件：不正确法线值。
Incorrect mesh NAME; Wrong tangents	损坏的文件：不正确正切值。
Incorrect mesh NAME; Wrong texture coordinates	损坏的文件：不正确的纹理坐标值。
Incorrect mesh NAME; Wrong vertex group weights	损坏的文件：不正确的顶点组权重值。

Incomplete vehicle. The NAME vehicle doesn't have any chassis or hull	模型的 NAME 车辆是不完整的，因为它应该包含一个 Chassis 或 Hull 元素。
Incomplete vehicle. The NAME vehicle requires at least one bob	模拟的 NAME 车辆是不完整的，因为它应该含有至少一种 Bob 元件。
Incomplete vehicle. The NAME vehicle requires at least one wheel	模拟的 NAME 车辆是不完整的，因为它应包含至少一个 Wheel 元件。
Invalid link found in node material. Material: "NAME".	"NAME" 节点材料包含节点之间的不正确连接。
No image in the NAME texture. [Material: NAME.]	纹理有没有图像。
No texture for the NAME particle settings texture slot.	在粒子设置 “纹理槽没有纹理。
No texture in the NAME world texture slot.	在 NAME 世界纹理插槽没有纹理
No texture in the texture slot. Material: NAME.	有一个在材料的纹理插槽没有纹理
Node material invalid: "NAME". Check sockets compatibility: "FROM_NODE" with "TO_NODE".	节点材质错误：之间的联系的输入和输出类型 FROM_NODE 和 TO_NODE 节点应该匹配。
Object "NAME" hasn't renderable data. Converted to EMPTY.	名为 "NAME" 的对象是失效的，比如没有多边形。对象的类型已更改为空。
Object constraint has no target. Check "CONST HOLDER NAME".	该 Target Object 为对象的约束（在 Object Constraints 选项卡）没有设置。
Object "NAME" has the mesh with shape keys. The property "Relative" of mesh has been enabled.	名为 "NAME" 的对象具有按键形状的网格。该网具有“相对”属性，使这是被禁止的。
Object "NAME" has no data or data is broken. Change object type to EMPTY.	名为 "NAME" 的对象是失效的，比如没有多边形。对象的类型已更改为空。
Packed media "FILE_NAME" has not been exported to "CONVERTED_FILE_PATH"	填充媒体文件 "FILE_NAME" 不能被转换为 "CONVERTED_FILE_PATH"。请解压缩这个文件，并将其转换。

Particle system error for “NAME”; The “NAME” vertex color specified in the from field is missing in the last of the “OBJECT_NAME” object’s vertex colors	该名称顶点颜色是在指定 from 领域，但它没有出现在 OBJECT_NAME 发射器。
Particle system error for “NAME”; The “NAME” vertex color specified in the to field is missing in the list of the “OBJECT_NAME” object’s vertex colors	这个 NAME 顶点颜色中指定 to 场，但它不存在于被选择作为一个粒子的 OBJECT_NAME 对象。
Particle system error for “NAME”; The “NAME” vertex color specified in the “to” field is missing in the “OBJECT_NAME” object (“GROUP_NAME” dupli group)	这个 NAME 顶点颜色中指定 to 区域，但它不存在于被选择作为一个粒子的 GROUP_NAME 组的 OBJECT_NAME 对象。
Particle system error for object “NAME”. Invalid dupli object “OBJECT_NAME”.	粒子系统错误的对象名称。无效 DUPLI 对象 OBJECT_NAME。
Particle system error. Unsupported render type “TYPE” for the HAIR particles “NAME” on object “OBJECT_NAME”. Particle system removed.	Particle system error. Unsupported render type TYPE for the HAIR particles PSYS_NAME on object NAME. The particle system has removed.
Particle system error. Unsupported render type “TYPE” for the EMITTER particles “NAME” on object “OBJECT_NAME”. Particle system removed.	Particle system error. Unsupported render type TYPE for the EMITTER particles PSYS_NAME on object NAME. The particle system has removed.
Particle system error for “NAME”. Dupli object isn’t specified.	粒子系统错误：没有对象被选中作为一个粒子。
Particle system error for “NAME”. Dupli group isn’t specified.	Particle system error: no group is selected as a particle.
Particle system error for “NAME”. Wrong dupli object type “TYPE”.	被选择为粒子不适当类型的对象。支持的类型：MESH.
Particle system error for “NAME”. Dupli object “NAME” has not been exported.	The NAME object which is selected as a particle can not be exported (the Do not export checkbox is set).
Particle system error for “NAME”. The “GROUP_NAME” dupli group contains no valid object for export.	被选择作为粒子的 GROUP_NAME DUPLI 组包含导出没有有效的对象。无论这样的对象具有 Do not export 启用复选框或对象的类型不适合。支持的对象类型：MESH.
8 Sound file is missing in the SPEAKER object “NAME”. Converted to EMPTY. ⁹⁷	扬声器没有附加声音。物体的类型已更改为空。 ⁹⁷
The lamp object “NAME” has	The lamp object “NAME” has

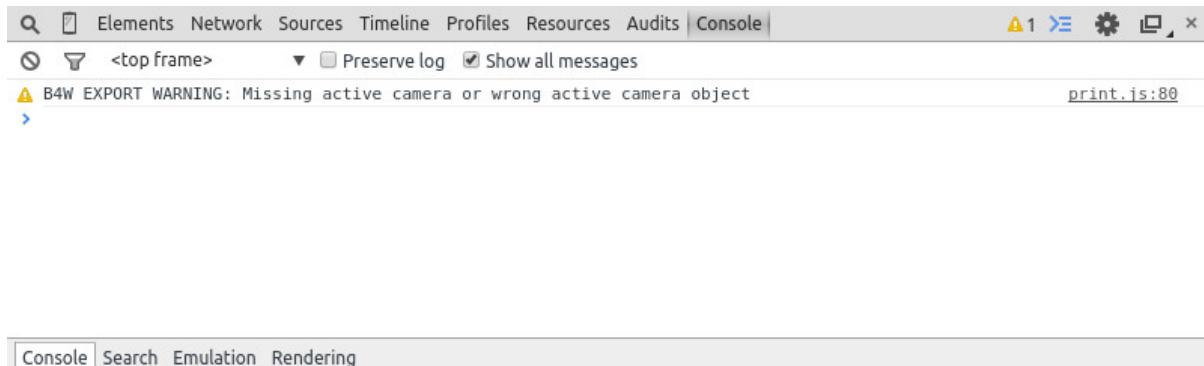
The main scene NAME can not be rendered by another scene. Material NAME has been removed.	主场景名称不能被其他场景渲染。材料名称已被删除。
The NAME action has decimal frames. Converted to integer.	操作具有十进制的帧。转换为整数。
The NAME armature modifier has a proxy object as an armature. Modifier removed.	一个骨架修改器具有一个代理对象作为骨架。 修改器已删除。
The NAME armature modifier has no armature object or it is not exported. Modifier removed.	The NAME Armature modifier has no armature object or it is not exported. Modifier removed.
该名称曲线修正没有对象。修改删除。	该名称曲线修正没有对象。修改删除。
The NAME curve modifier has unsupported curve object. Modifier removed.	该名称的曲线修饰有不支持的对象。修改删除。
The NAME object has the NAME armature modifier and a vertex animation. Modifier removed.	Name 对象既有顶点动画和不支持的电枢修改。其结果是，该改性剂已被删除。
该名称 Armature 修正没有骨架问题，也没有导出。修改删除。	The NAME object has the NAME armature modifier. The armature object in this modifier isn't specified or belongs to a not exported scene. Modifier removed.
The NAME LAMP node has no lamp object. Material: NAME.	在指定的名字错了对象 LAMP 节点。
The NAME node is not supported. The NAME material will be rendered without nodes. Material: NAME.	引擎不支持该节点与该名字，并因此节点材料将被关闭。经常出现这种情况时，循环节点使用。
The NAME object has NAME armature modifier which references the wrong group. Modifier removed.	一个物体应该是在同一组作为骨架，或两个物体应该明确地出现在场景中。
“TEXTURE_TYPE” texture type is not supported for world “NAME”.	不支持 TEXTURE_TYPE 纹理类型为世界 WORLD_NAME
Using B4W_REFRACTION node <small>8.8 非致命导致的错误</small> NODE_NAME with incorrect type of Alpha Blend. Material: NAME.	用不正确的透明混合属性节点材质。Alpha sort, Alpha blend 和 Add 使用的是“折射”节 ⁹⁸ 点时允许的。

Wind bending: vertex colors weren't properly assigned for "NAME". Properties were set to default values.	风弯曲参数设置：它必须指定或者所有顶点颜色层的名称 (Main stiffness (A), Leaves stiffness (R), Leaves phase (G), Overall stiffness (B)), 或主要的一个唯一的 (Main stiffness (A)), 或没有其中之一的。
Wrong "Height Map" input for the "NAME" B4W_PARALLAX node. Only link from the TEXTURE node with a non-empty texture is allowed.	错误的数据传递给该名称的“高度贴图”输入 B4W_PARALLAX 节点。只从一个非空的输出 TEXTURE 节点是允许的。
Wrong texture coordinates type in texture NAME. [Material: NAME.]	下列坐标类型的图像纹理支持：UV, Normal 和 Generated.
Wrong F-Curve interpolation mode for ACTION_NAME. Only BEZIER, LINEAR or CONSTANT mode is allowed for F-Curve interpolation. Switch to BEZIER.	The following types are supported for action interpolation mode: BEZIER, LINEAR and CONSTANT.
Wrong vertex animation vertices count for mesh NAME. It doesn't match with the mesh vertices count for "ANIM_NAME".	顶点动画导出被启用，但顶点在焙烤 ANIM_NAME 动画帧的数目不匹配网格顶点编号。可能的解决方案是“重新烘焙”的动画。

其他消息

这些消息可以在浏览器的控制台中查看（与打开 F12 一样）当一个场景被加载。该消息是这样的：

B4W EXPORT 警告：导出消息，这就需要用户注意



错误信息	原因
Missing active camera or wrong active camera object	在场景中没有激活的 (Camera 属性在 Scene 面板).
Missing world or wrong active world object	应该有至少有一个数据块的世界场景中。
NAME particle settings has the NAME texture rendering a scene. It has been replaced by the default texture.	粒子设置数据块名称包含被用于渲染场景到纹理名称。此纹理被替换为默认的纹理。
The action NAME has no fcurves.	The action with the name “NAME” has no fcurves.
The “NAME” camera has unsupported PANORAMIC type. Changed to PERSPECTIVE type.”	不支持全景相机。立体模式来代替。
Unsupported texture type or texture is missing for Lens Flare material “NAME”	不支持的纹理类型或纹理缺少镜头光晕材质“NAME”
Use of ENVIRONMENT_MAP as diffuse color is not supported. Use as mirror instead.	The ENVIRONMENT MAP texture can not be used as diffuse color. Disable the Diffuse > Color and enable the Shading > Mirror option on the Textures > Influence panel to use this texture as mirror map.

插件翻译

还有就是可以翻译插件到 Blender 所支持的语言。为了做到这一点，重命名文件“empty.po”，它位于目录 SDK/blender_scripts/addons/blend4web/locales，下表中的名称之一：

文件名	语言
ru_RU.po	俄语
ja_JP.po	日本
en_US.po	英语
nl_NL.po	荷兰
it_IT.po	意大利
de_DE.po	德语
fi_FI.po	芬兰
sv_SE.po	瑞典
fr_FR.po	法国
es.po	西班牙语
ca_AD.po	加泰罗尼亚
cs_CZ.po	捷克
pt_PT.po	葡萄牙语
zh_CN.po	简体中文
zh_TW.po	繁体中文
hr_HR.po	克罗地亚
sr_RS.po	塞尔维亚
uk_UA.po	乌克兰
pl_PL.po	波兰
ro_RO.po	罗马尼亚
ar_EG.po	阿拉伯
bg_BG.po	保加利亚语
el_GR.po	希腊语
ko_KR.po	朝鲜的
ne_NP.po	尼泊尔
fa_IR.po	波斯语
id_ID.po	印度尼西亚
sr_RS@latin.po	塞尔维亚拉丁语
ky_KG.po	吉尔吉斯
tr_TR.po	土耳其
hu_HU.po	匈牙利
pt_BR.po	巴西葡萄牙语
he_IL.po	希伯来语
et_EE.po	爱沙尼亚语
下页继续	

Table 8.1 – 续上页

文件名	语言
eo.po	世界语
es_ES.po	西班牙西班牙
am_ET.po	阿姆哈拉语
uz_UZ.po	乌兹别克
uz_UZ@cyrillic.po	乌兹别克西里尔
hi_IN.po	印地语
vi_VN.po	越南

然后打开该文件并编辑/翻译。

当翻译准备好了，你可以与我们联系，将他们作为插件的一部分。

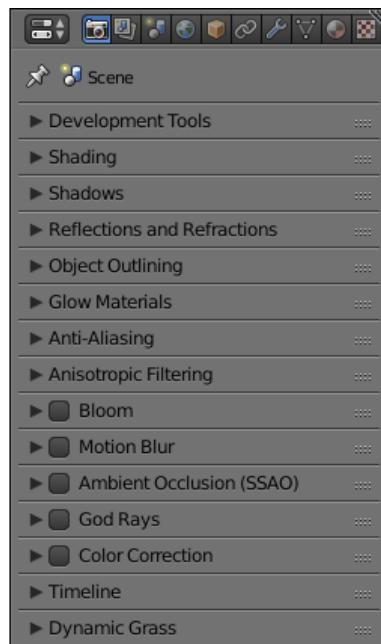
场景设置

目录

- 场景设置
 - 渲染面板
 - * 开发工具
 - * 著色
 - * 阴影
 - * 反射和折射
 - * 物体勾边
 - * 晕光材质
 - * 抗锯齿
 - * 各向异性过滤
 - * 布隆光晕
 - * 运动模糊
 - * 环境光遮蔽 (SSAO 方法)
 - * 上帝光芒
 - * 色彩校正
 - * 时间线
 - * 动态草
 - 场景面板
 - * 场景
 - * 单位
 - * 音频
 - * 逻辑编辑器
 - * NLA 片段
 - * Meta 标签
 - * 物理
 - * Object Clustering & LOD
 - * Object Selection
 - * 锚
 - * 导出选项
 - 世界环境面板
 - * 预览
 - * 世界环境
 - * 天光照明
 - * 雾场
 - * 程序天空
 - * 动画
 - * 导出选项

定义整个场景的外观和行为（不只是一个单一的对象）所有参数都能在三个面板里找到：Render 面板时，Scene 面板和 World 面板。

渲染面板



场景中的所有图像渲染参数都在此面板上找到。

开发工具

开发服务器设置。详尽的描述在[自己的章节](#).



著色

此面板包含各种着色选项。



推荐设置选项 该按钮用于实现 3D 场景在 Blender 的视口和 Web 浏览器的外观之间的最大一致性。按下此按钮：

- 启用 **世界空间着色**,
- 切换材质模式到 GLSL,
- 切换视口着色模式到 **材质** 和
- 设置主摄像头适合 **垂直**。

世界空间着色 此选项打开和关闭 **世界空间着色**（照明数据的世界空间演绎将用于对象阴影）。默认情况下禁用。

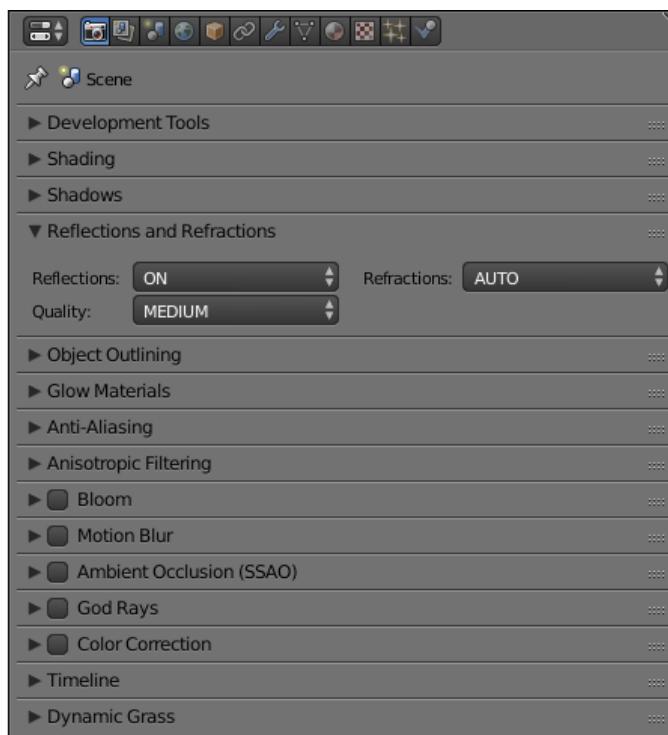
阴影

阴影设置。详尽的描述在[自己的章节](#).



反射和折射

反射和折射效果设置。



反射 反射效果设置。可设置 ON, OFF 或 AUTO。默认设置为 ON。

折射 折射效果设置。可设置 ON, OFF 或 AUTO。默认设置为 AUTO。

品质 对于反射效果的画质设置。可设置 HIGH (最高反射品质), MEDIUM (中等品质), 或 LOW (最低质量)。默认设置为 MEDIUM。

物体勾边

物体勾边设置。详尽的描述在[自己的章节](#).



晕光材质

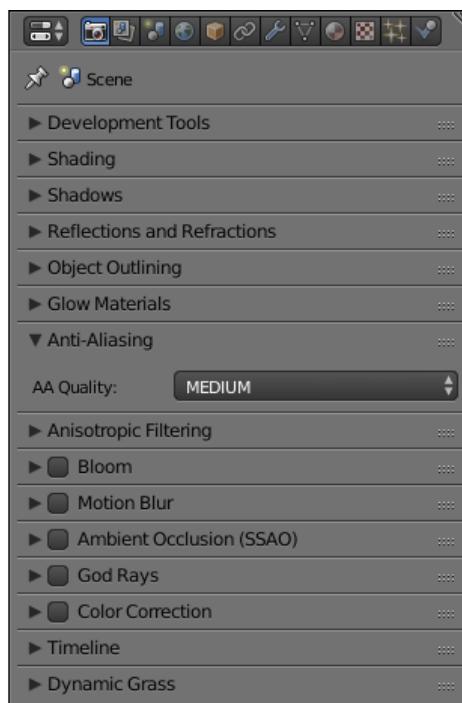
晕光材质效果设置。详尽的描述在自己的章节。



注解: 为动态加载物体, AUTO 设置被解析为 OFF, 当没有带辉光的材料在场景中时。如果使用的是动态加载, 你应该将这个参数设置为 ON.

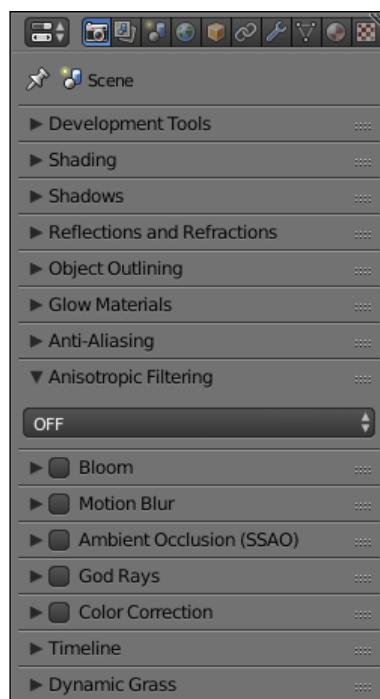
抗锯齿

抗锯齿的设置。详尽的描述在[自己的章节](#).



各向异性过滤

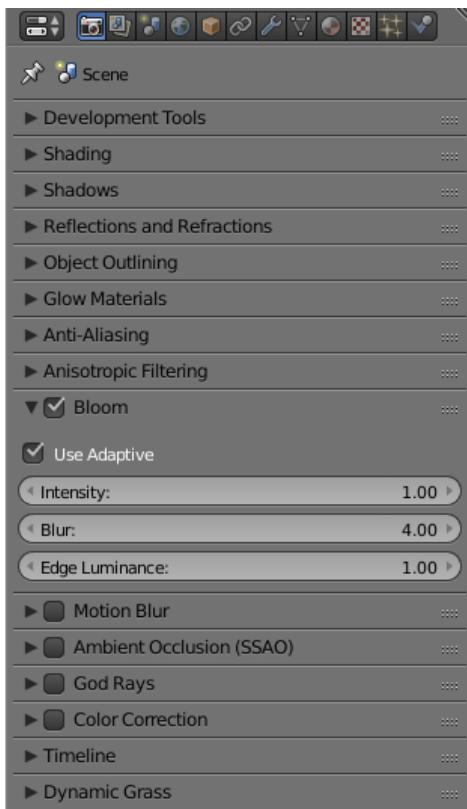
该参数可用于启用或确定停用各向异性过滤并且还设置用于它的纹理样本的数目。默认情况下，各向异性过滤被禁用。



可用值: 16x, 8x, 4x, 2x 和 OFF (默认值)。

布隆光晕

布隆光晕效果设置。详细描述在[自己的章节](#)。



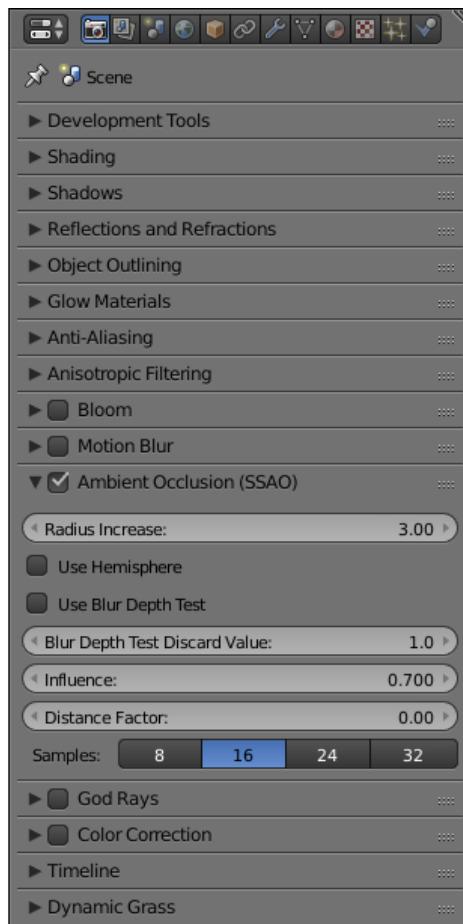
运动模糊

运动模糊设置。详尽的描述在[自己的章节](#)。



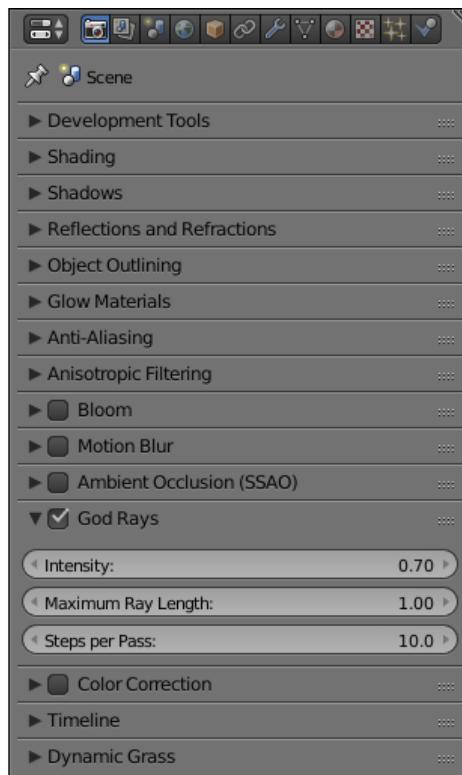
环境光遮蔽 (SSAO 方法)

屏幕空间环境光遮蔽 (SSAO) 的设置。详尽的描述在[自己的章节](#).



上帝光芒

上帝光芒效果设置。详尽的描述在自己的章节。



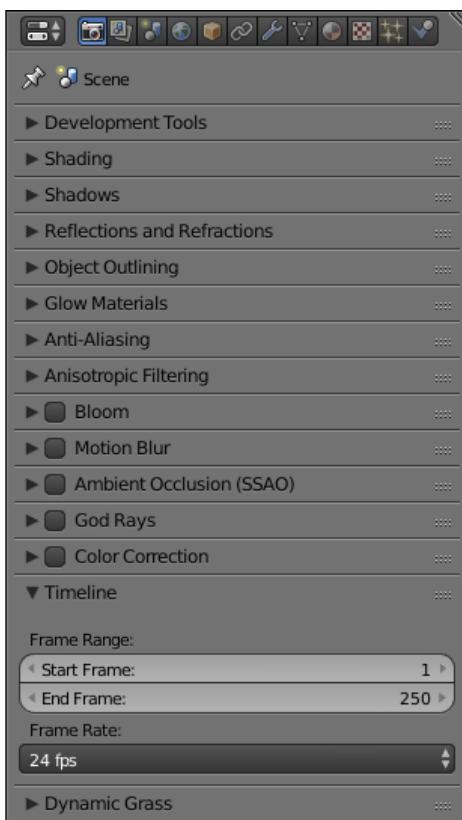
色彩校正

色彩校正设置。详尽的描述在[自己的章节](#).



时间线

时间线的设置。



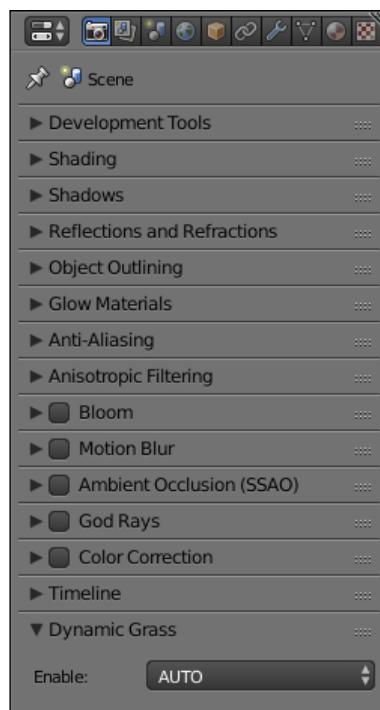
起始帧 时间轴的第一帧。默认设置为 1。

结束帧 时间轴的最后一帧。默认设置为 250。

帧率 每秒的帧的数目。默认设置为 24。此参数只影响动画播放的速度（而不是场景本身）。

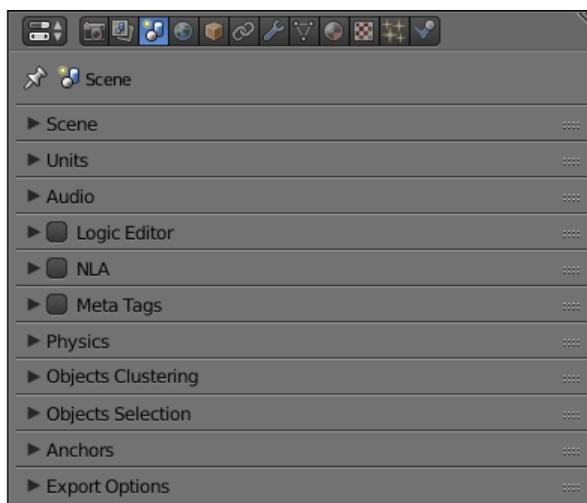
动态草

启用和禁用**动态草** 效果。



可能的值: ON, OFF 和 AUTO。默认设置为 AUTO。

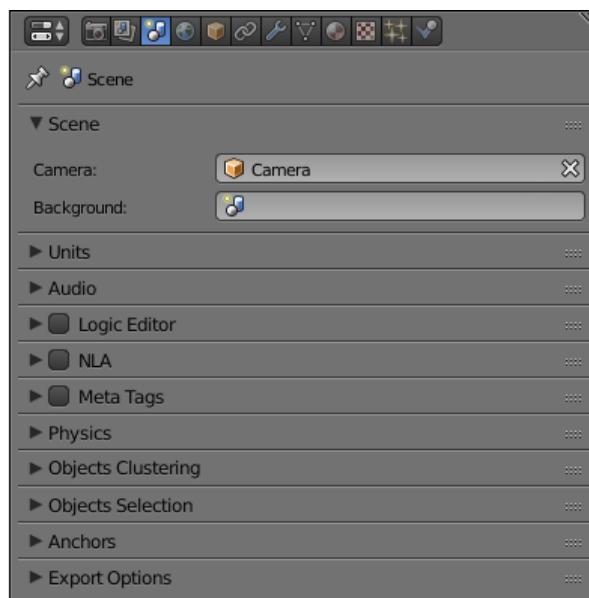
场景面板



关于场景行为的设置，包括音频，物理和动画设置，都在此面板上找到。

场景

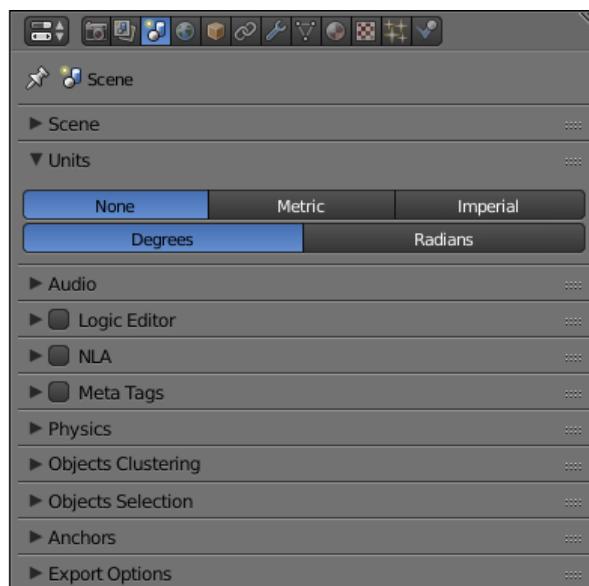
场景设置



相机 用于渲染场景的相机。

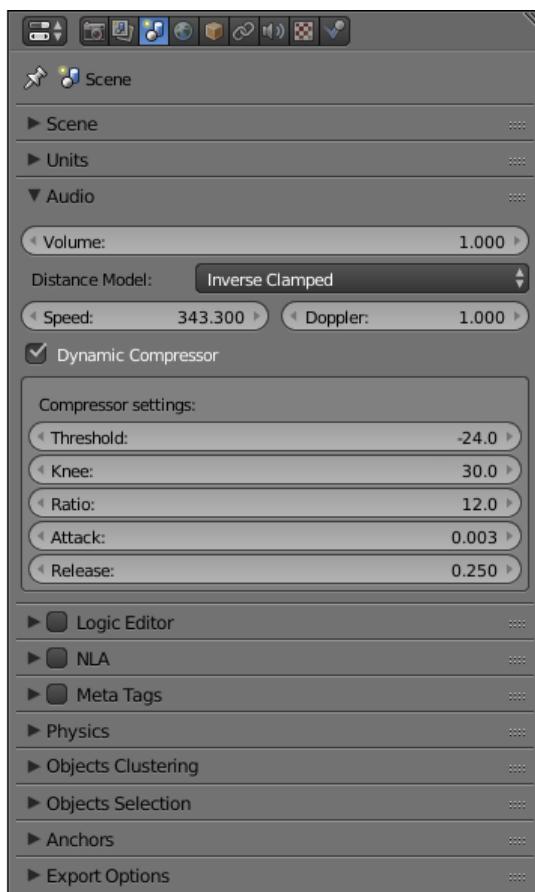
单位

设置场景中使用的度量单位。



音频

音频设置



音量 应用中的声音的主音量。此值介于 0 到 100 之间，并默认值是 1.0。

距离模型 用于计算距离衰减的距离的模型。以下型号的引擎支持：

None - 不使用距离模型，声音具有体积不变无论距离远近。

指数钳制 - 一个钳位指数距离模型。

线性钳制 - 被夹住的直线距离模型。

逆钳制 - 被夹住的逆距离模型。

以下型号部分支持（以同样的方式相应的 钳位型车型）：

F

线性

反

此参数默认设置为 逆钳制。

速度 此参数设置用于多普勒效应计算音速。它的值以米每秒为单位，默认设置为 343.3。

多普勒效应 这为多普勒效应计算距离因素。它的默认值，如果 1.0。

动态压缩 压缩音频信号的动态范围。该功能可用于使声音更加丰富。默认情况下禁用。

阈值 如果音频信号的幅度超过由该参数指定的值，压缩器将降低其等级。默认设置为-24 分贝。

拐点 反响曲线切换到递减模式时低于阈值的间隔。默认设置为 30。

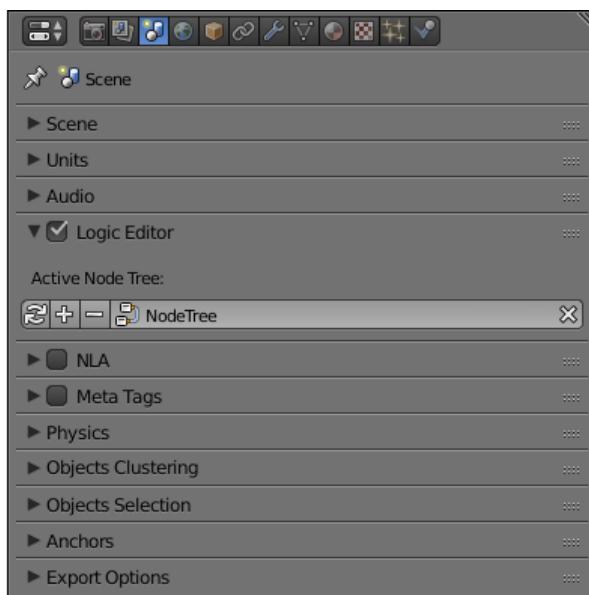
比率 增益衰减量。默认设置为 12。

起音 让压缩器减少增益 10 dB 的时间（以秒为单位）。默认设置为 0.003。

释放音 让压缩器提高增益 10 dB 的时间（以秒为单位）。默认设置为 0.25。

逻辑编辑器

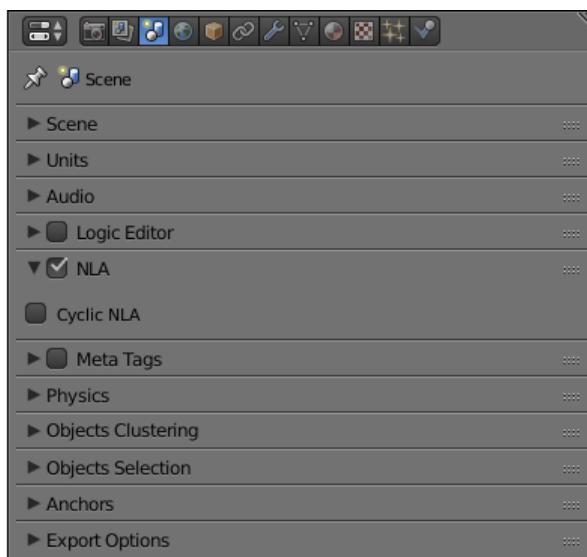
设置为使用逻辑节点树（在逻辑编辑器 中创建的）。默认情况下禁用。



主控节点树 即在场景的播放使用节点树。

NLA 片段

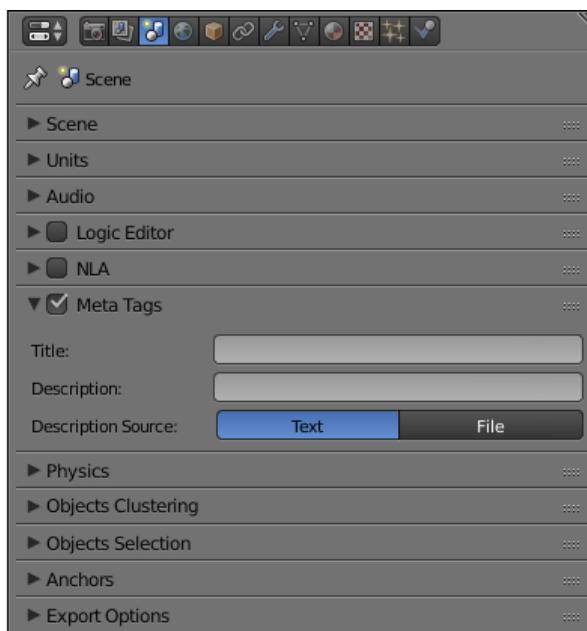
非线性动画播放设置。默认情况下禁用。



循环 NLA 如果该参数被启用，NLA 动画播放完后将被重复。

Meta 标签

应用程序的 meta 标签。



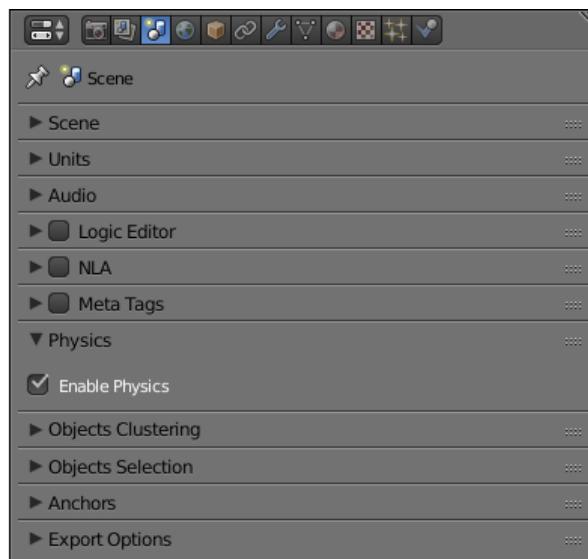
名称 应用程序的标题。

描述 应用程序的说明。可以是一个简单的文本或文本文件的链接（如果 Description Source 参数设置为 File 值）。

说明源 应用程序的说明的来源。说明可以从一个文件中加载，或直接在 Description 字段中写出。此参数可以有两个值，Text 和 File，它是默认设置为 Text。

物理

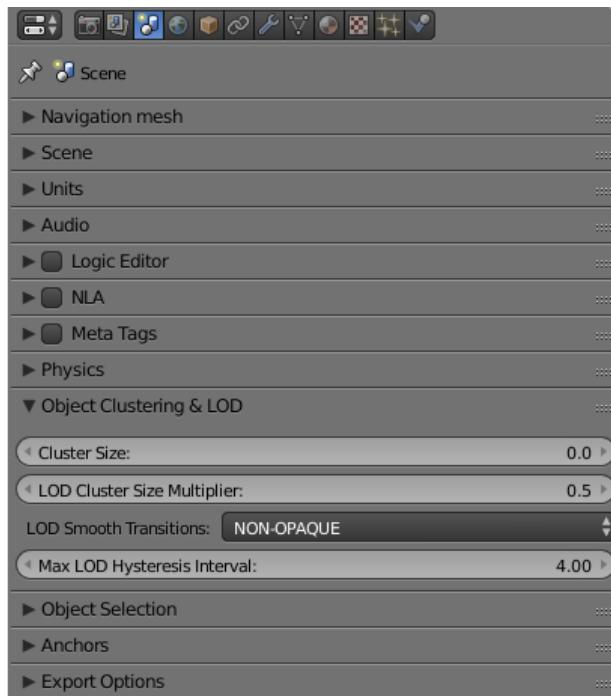
物理设置。



启用物理 允许在应用程序中使用物理。默认情况下启用。

Object Clustering & LOD

Settings for object clustering and Levels Of Detail.



簇大小 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 簇大小乘数 This parameter is used to subdivide clusters (based on LOD distance specified individually for each object) into smaller ones to make transitions between different levels of detail less noticeable. The size of that smaller clusters is defined by object LOD distance multiplied by the value of this parameter. Higher values lead to bigger clusters which increases performance but makes transition between LODs more noticeable, while lower values make said transitions less noticeable at the cost of decreased performance. Can be used for optimization purposes. Set to 0.5 by default.

LOD Smooth Transitions Defines what objects will use smooth transitions while switching their LOD levels. Has the following options:

- OFF - smooth transitions are disabled (fastest).
- NON-OPAQUE - use smooth transitions for objects with Add, Alpha Clip, Alpha Blend, Alpha Sort and Alpha Anti-Aliasing [materials](#).
- ALL - smooth transitions will be used for all objects (slowest).

Choosing the “ALL” value can noticeably decrease application performance, so use it with caution. Default value is “NON-OPAQUE”.

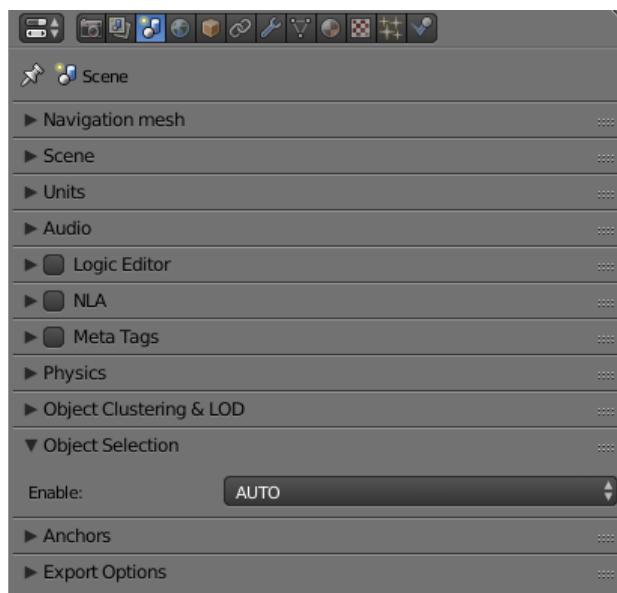
Max LOD Hysteresis Interval

The length of the interval (in meters) used for switching LOD levels. The half of this value is added/subtracted from the distance threshold in order to make two different thresholds for switching to the lower and to the higher LOD level. This should reduce LOD “popping” effect. Set to 4.0 by default.

Object Selection

物体选择设置。对象可以使用 API函数来选择两种 `pick_object()` 并用[逻辑节点](#)。

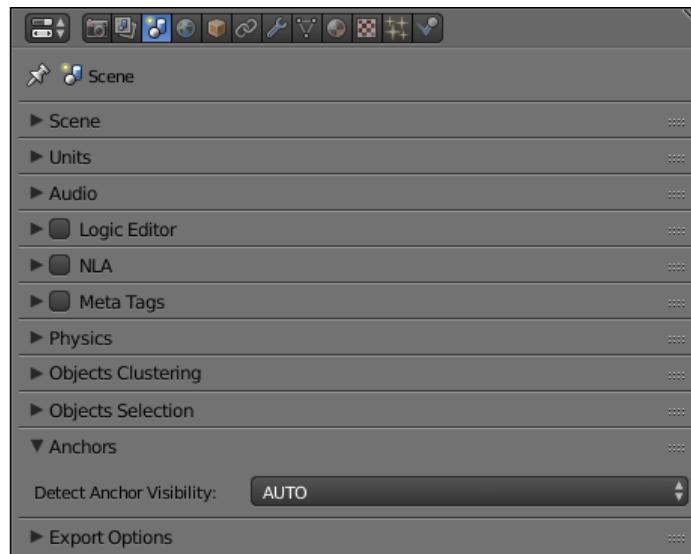
注解: 在[场景查看器](#), 选择默认情况下启用。您可以在 工具和除错面板将其关闭。



启用 定义物体可以或不能被选择的参数。它可以有 ON, OFF 或 AUTO 值。默认设置为 AUTO。

锚

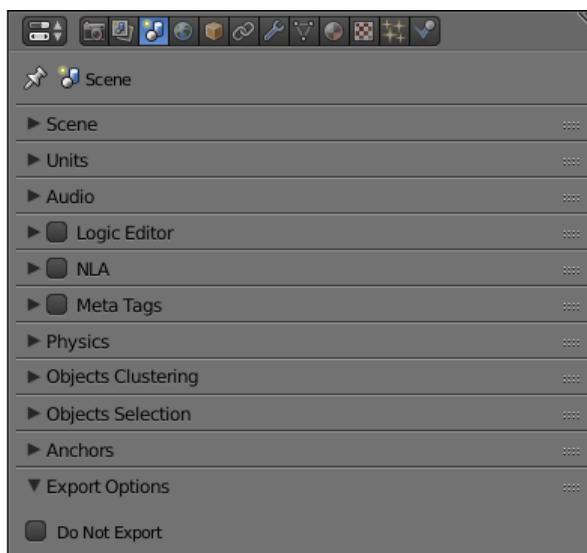
Anchor 可见性检测设置。



检测锚能见度 锚能见度检测参数。有 ON, OFF 或 AUTO 值。默认设置为 AUTO 。

导出选项

场景设置导出参数。



不要导出 如果启用此参数，场景设置将在导出过程中被忽略。

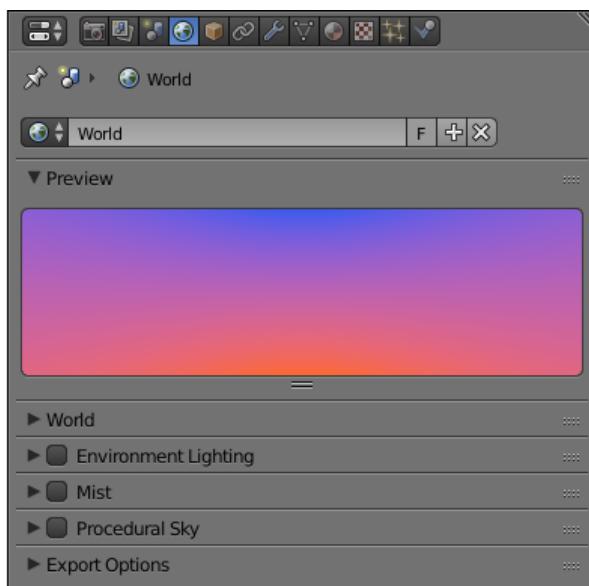
世界环境面板



环境设定，如天空，雾气和这样的设置在该面板上找到。

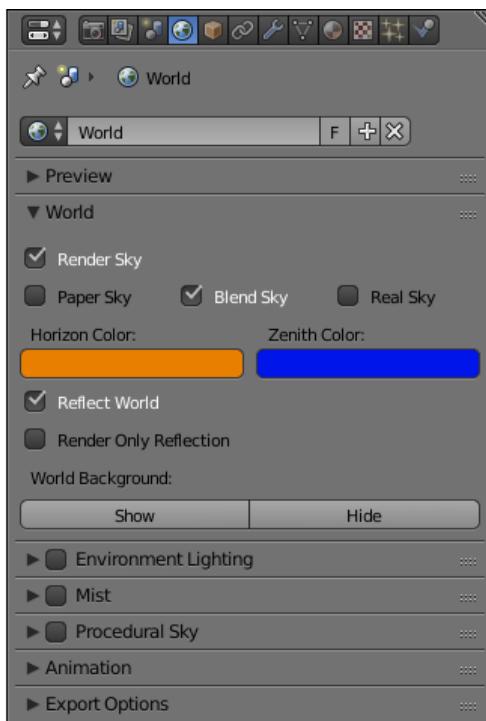
预览

环境预览。



世界环境

天空的设置。



渲染天空 如果启用此参数，引擎会在场景渲染天空。

墙纸天空 如果该参数被启用，天空梯度将总是从屏幕的顶部到底部，不管摄像机的位置和角度。

混色天空 地平线和天顶色之间的平滑过渡。

真实天空 天空渲染与地平线受相机角度影响。

水平颜色 水平线的颜色。

天顶色 天空顶的颜色。

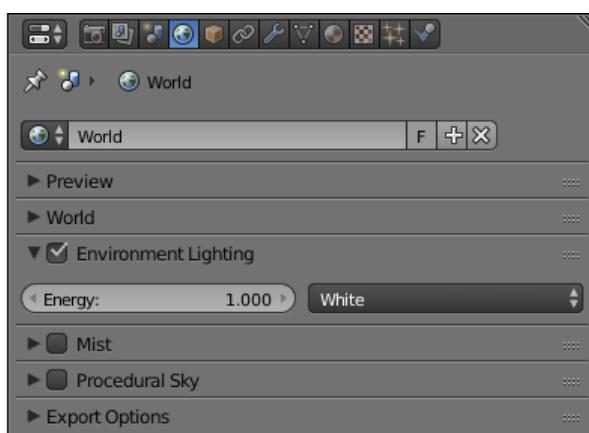
反射世界 渲染反射时渲染天空。

只渲染反射 在渲染的反射时 只渲染天空。

世界背景 启用和禁用 世界背景 (位于 3D View 面板下的 Display 选项卡)。当激活此选项，则背景颜色在视区窗口中渲染的方式跟在引擎里面渲染是一样的。

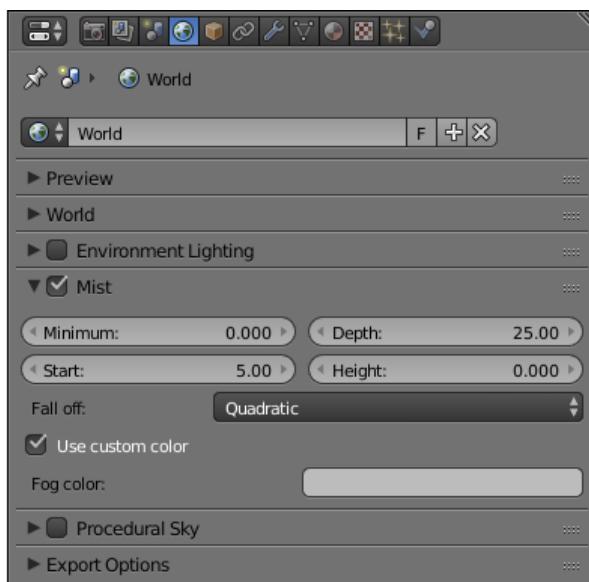
天光照明

环境照明设置。详尽的描述在[自己的章节](#)。



雾场

雾设置。



最小值 雾的最小强度。默认设置为零。

深度 在从相机这个距离的雾效果达到最大强度。默认设置为 25。

起始 雾效果开始出现在从相机这个距离。默认设置为 5。

高度 该参数快速指定如何雾强度随相机的高度增加而减小。默认设置为 0。

散去 该参数指定规则，根据该边界之间的雾的变化（由 Start 和 Depth 参数指定）的密度。可以有以下值：Quadratic, Linear, Inverse Quadratic。默认设置为 Quadratic。

使用自定义颜色 可用于设定的雾的颜色。默认情况下启用。如果禁用此参数，标准 (0.5, 0.5, 0.5) 的颜色将被使用。

雾颜色 颜色雾。, 如果 使用自定义颜色参数启用，就可以改变。浅灰色 (0.5, 0.5, 0.5) 默认情况下使用。

程序天空

程序天空设置。详尽的描述在[自己的章节](#)。



动画

环境动画设置。



应用默认动画 如果启用此参数，引擎将开始加载场景的时候播放动画。

行为 设置默认的动画的行为。可用选项有：

- Loop - 动画将重复播放。
- 完成停止 - 动画将播放一次。

- 完成重置 - 动画将播放一次，然后再切换回第一帧。

仅启用了 应用默认动画参数此选项可用。

导出选项

环境参数导出设置。



不要导出 如果启用此参数，环境设置将在导出过程中被忽略。

物体

目录

- 物体
 - 类型
 - 静态和动态物体
 - 设置
 - * 物体选项卡
 - * 物理标签
 - 锚设置
 - 物体变换 API
 - 获取物体 API
 - 物体选择
 - 复制物体 (实例化)
 - * 制作一个简单的复制
 - * 制作深复制
 - * 删除物体
 - 四元数
 - * 四元数例子
 - 通过 TSR 矢量移动
 - 线渲染
 - Levels Of Detail
 - * Overview
 - * Specifics and Differences From BGE
 - * Smooth LOD Switching
 - * Hysteresis

物体是在 3D 场景空间中放置的不同类型（网格，照相机，灯等）的组件。

类型

该引擎支持以下类型的物体：

- 网格 (mesh)
- 相机
- 灯光

- 空物体
- 骨架
- 声音
- 曲线
- 文本
- 融球
- 曲面

在场景导出时, CURVE, TEXT, METABALL 和 SURFACE 类型的物体转换为 MESH 类型的物体。

静态和动态物体

物体可以是静态或动态的。

静态物体是在运行程序时不能移动, 动画或以任何其它方式改变的物体。这样的物体可以提高性能, 如果它们具有相同的材质可以合并在一起。

动态物体在运行的应用程序可以移动, 动画或通过其手段改变。它们也可以与其他物体交互, 包括静态物体。动态物体不会彼此或与静态物体相结合。

只有 MESH 和 EMPTY 类型物体可以是静态的或是动态的。所有其他物体类型, 如 CAMERA 和 ARMATURE, 始终是动态的。

静态 MESH 物体比动态的渲染得快得多, 因此, 为了更好的性能, 最好将动态网格数保持在最低水平。任何其他类型的物体, 静态和动态, 不显著影响性能。

具有动画, 物理或父关系的是一个动态物体, 被认为是动态, 以及由下面的逻辑节点控制的物体:

- 播放动画
- 变换物体
- 移动到
- 继承材质

API 方法是关注物体的运动, 复制和动画 (包括物体和节点材料) 只能应用于动态物体。为了使物体的不动态设置的运动可能, 有必要激活设置 Force Dynamic Object 选项。

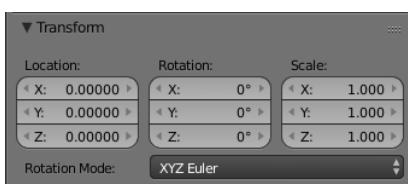
设置

以下是功能支持所有类型的物体: 变换, 数据引用, 父物体, 组成员和一套 Blend4Web 的特殊属性。

物体选项卡



变换面板



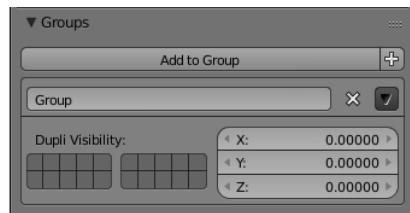
变换 > 位置 位置坐标。

变换 > 旋转 旋转角度。对于物体的旋转所有可用的模式可以使用 (旋转 Mode)。但是，只有 四元 (WXYZ) 和 XYZ Euler 都支持**物体动画**.

变换 > 缩放 缩放。所有 3 分量 (X, Y, Z) 应该是相同的。缩放不支持物理物体。

组面板

组 物体所属的物体组



Levels of Detail Panel

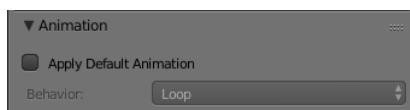


Levels of Detail > Object The object to use for this level of detail.

Levels of Detail > Distance The distance to begin using this level of detail.

Using levels of detail is described in the [dedicated section](#).

动画面板

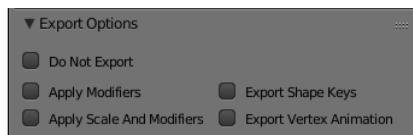


动画 > 应用默认动画 传到引擎开始后播放物体的动画。

动画 > 动画混合 只对骨骼物体。允许骨骼动画之间进行混合。

动画 > 行为 到达最后一帧时动画的行为：完成停止 - 停止 完成重置 - 停止并跳到零帧，Loop - 永远重复。

导出选项面板



导出选项 > 不导出 不要导出此物体。

导出选项 > 应用修改器 应用导出物体的修改器。如果使用 SKIN 修改器，我们建议导出前应用掉它，因为它会重置顶点颜色和 UV 层可能会导致错误。

导出选项 > 应用缩放和修改器 一旦导出，应用物体的缩放和修改器。

导出选项 > 导出顶点动画 导出以前创建并保存的顶点动画。只适用于 MESH 类型物体。

导出选项 > 导出形变键 导出形变键。只适用于 MESH 类型物体。

注解：以下属性是相互排斥的：应用修改器，应用缩放及修改器，导出顶点动画和导出形变键。

渲染属性面板



渲染属性 > 隐藏 与此属性启用的物体将在加载时被隐藏。

Rendering Properties > Hidden Children All of the object's children and children's children and so on will be hidden on load. Available only if the Rendering Properties > Hidden property is enabled and if the object has any children.

渲染属性 > 不要渲染 禁止物体的渲染（例如用于有用的物理物体）。

此参数不可用于 Empty 类型的物体。

渲染属性 > 禁止视椎体剔除 禁用视椎体剔除优化。

此参数不可用于 Empty 类型的物体。

渲染属性 > 强制动态物体 强制物体变成了动态物体。

渲染属性 > 动态几何体与材质 允许使用几何更新 API 和为物体继承材质。

渲染属性 > 线呈渲染 允许使用该物体到渲染中参数线条。

此参数仅供 Empty 类型物体。

阴影面板

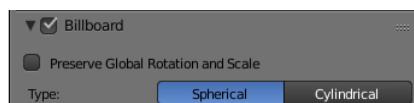


阴影 > 投射 该物体将投射阴影。

阴影 > 仅投射 该物体将投射阴影，但仍然看不到自己。成启用 **阴影 > 投射** 后可用。

阴影 > 接受 该物体将接收来自其他相邻物体的阴影。

广告牌面板



看板 使用物体作为看板（即自动定向面对摄像机）。

看板 > 保留全局方向和缩放 考虑到（世界空间）看板的物体的旋转和缩放。该物体沿 Blender 的 Y 轴查看时将朝明显的被引导到相机。启用 **看板** 复选框后可用。

看板 > 看板类型 看板方向模式。Spherical（默认）- 物体一面始终朝向相机，不管视角，Cylindrical - 类似于 Spherical，但旋转仅限于 Blender 的世界 Z 轴。启用看板后变得可用。

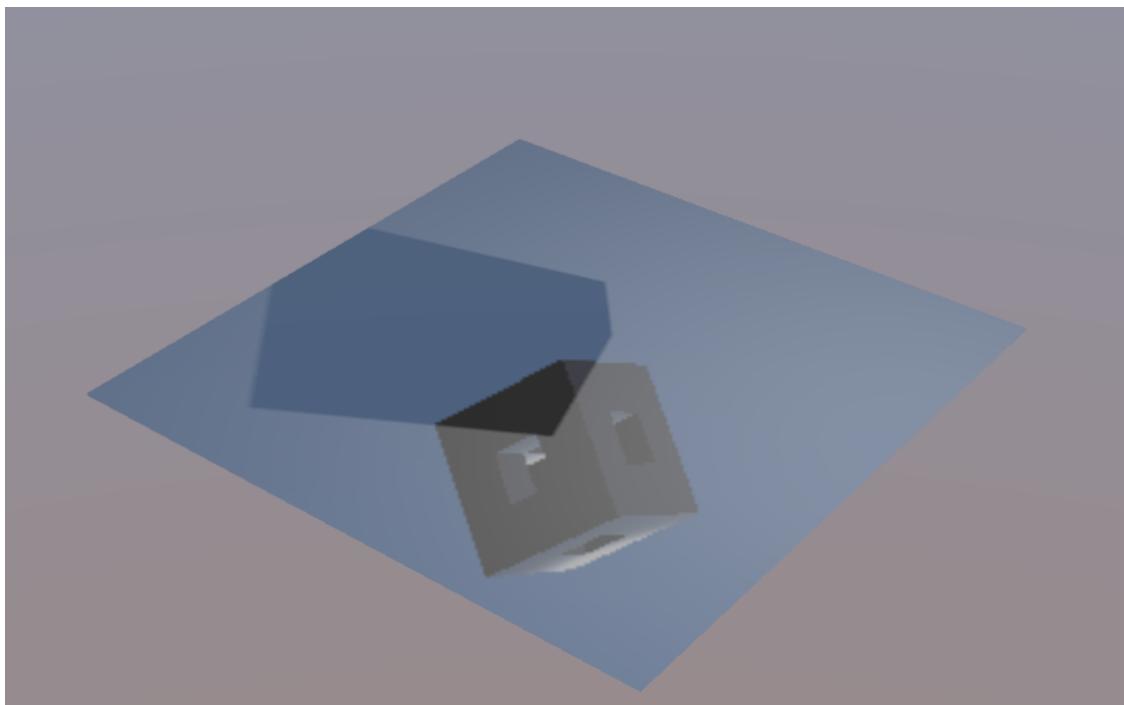
反射面板



反射 > 被反射 启用后物体被会在动态镜面中被反射。

反射 > 仅被反射 物体被反射但仍将看不见自身。开启 **反射 > 反射性** 后可用。

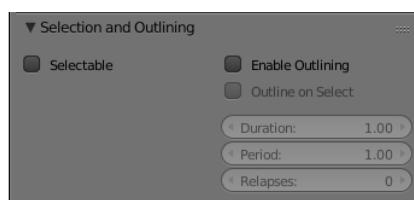
注解: 如果 **Reflexible Only** 参数与 **Shadows > Cast Only** 参数同时启用，引擎将不会渲染物体本身，而是将渲染它的影子和反射，像下面的图片所示。



反射 > 反射 当启用时该物体表面反射的其他物体。

反射 > 反射面 对于一个空物体定义了反射面的 **Name** 文本字段。启用 **反射 > 反射** 后可用。

选择和轮廓面板

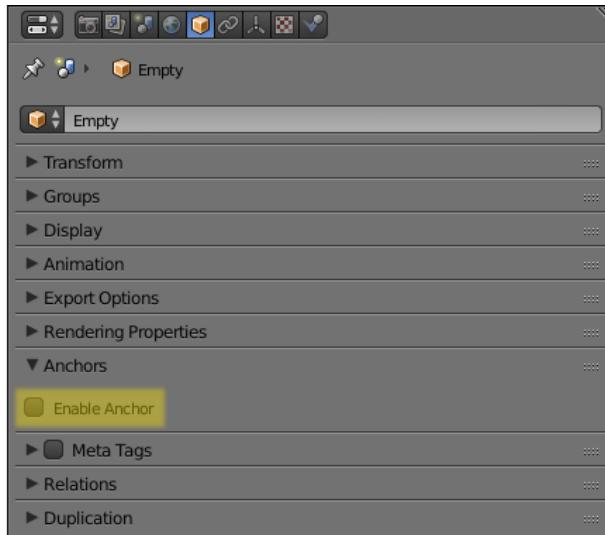


选择和外轮廓 > 可选择 启用**物体选择**，用鼠标或其他输入设备。

选择和外轮廓 > 启用外轮廓 启用**物体的外轮廓发光**。

锚面板

锚 > 启用锚 该参数将启用添加锚（2D 标签）到物体的界面。只可用于 EMPTY 物体。
在[相应章节](#) 描述。



元标签面板



元标签 添加物体的元标签的接口：

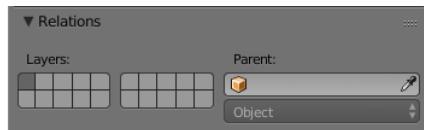
元标签 > 标题 物体的标题。

元标签 > 类别 物体的类别。

元标签 > 描述 物体描述。根据 说明源，此字段接受描述文本本身，或包含的文件的名称。

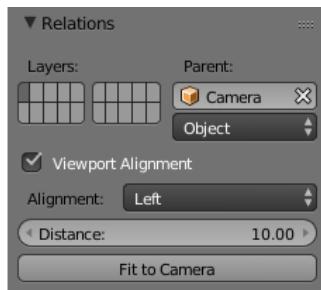
元标签 > 描述源 为说明来源类型：文本或文本文件。

关系面板。



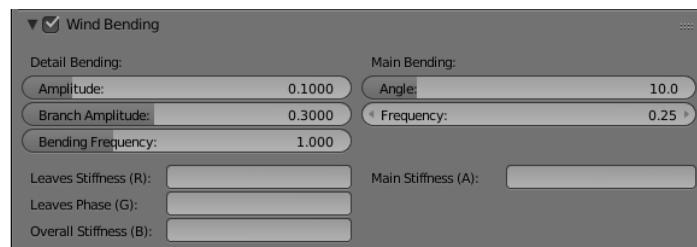
关系 > 父级 参照父物体。

如果父物体是一个摄像头， Viewport Alignment 设置将可见。



这些设置可被用于将物体对准的主控照相机。参考[相机章节](#) 的描述。

风弯曲面板



风弯曲 启用风弯曲程序动画。在[室外渲染](#) 页有完整的描述.。

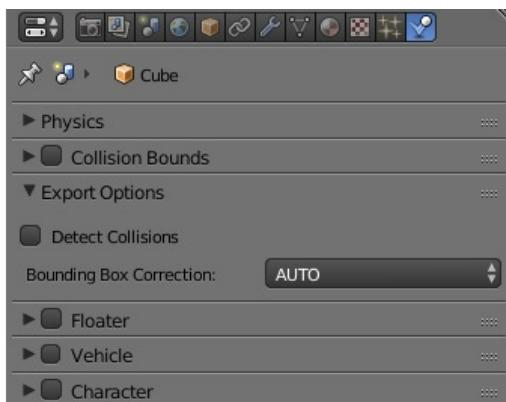
特效面板



特殊效果 > 关闭雾效 禁用物体的雾。

特殊效果 > 焦散 该物体将渲染自附近水的焦散效果。

物理标签



检测碰撞 激活物体的物理。

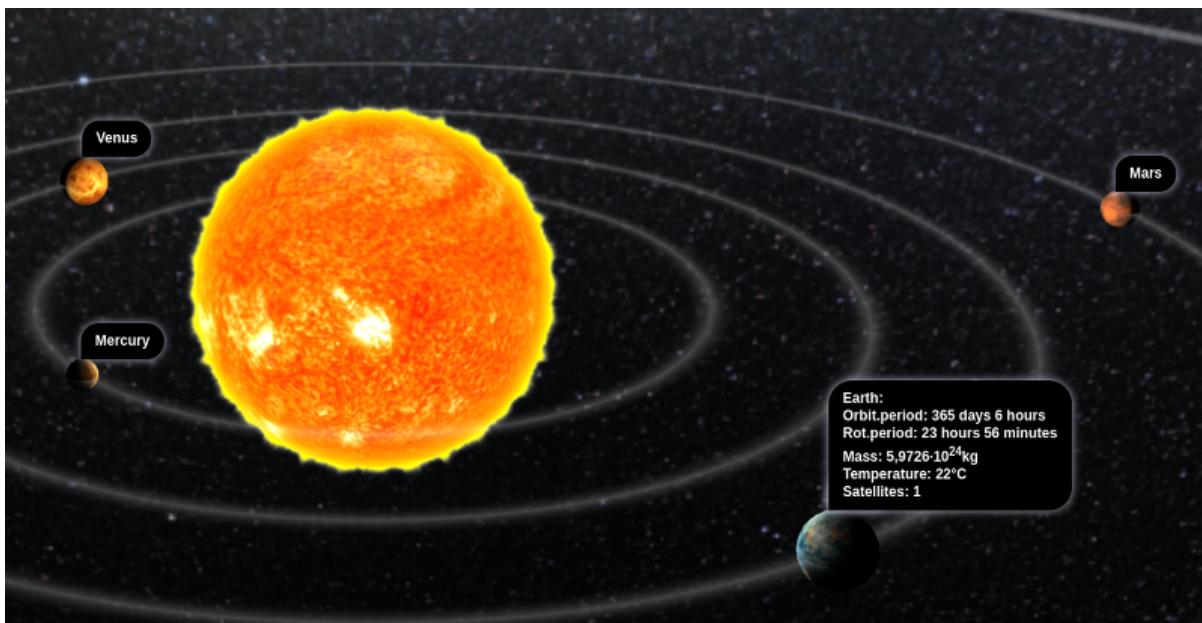
飘浮 使物体漂浮。在[physics](#) 一节对飘浮物体的设置进行了详细描述。

载具 使用物体作为车辆的一部分。该车辆设置中详细的描述在[物理](#) 一节。

角色 启用物体的角色物理特性。角色设置详细描述在[物理](#) 一节..。

锚设置

锚可以用来附加注释到 3D 物体。不管摄像机位置注释显示在物体附近，甚至在整个动画跟随它。



注释完全可以在 Blender 中创建。你需要做的是放置一个 Empty 物体在需要的位置，然后启用 Anchor 属性。对于注释文本可以在 Meta Tags 面板的 Title 和 Description 字段进行分配。



启用锚 使用此参数可添加锚（2D 标签）物体的界面。唯一适用于 EMPTY 物体。

类型 锚类型

- Annotation - 内容从元标签 所获得再分配给物体，显示出标准的部分。
- Custom Element - 从当前网页的任意 HTML 元素用作锚。
- Generic - 一个锚点的位置，可以使用 anchors API 模块进行检测。

默认值是 Annotation。

HTML 元素 ID 这指定了将被用作锚的 HTML 元素的 ID。这是仅当 Type 参数设置为 Custom Element.

可见检测 检测锚物体是否被其他物体重叠。打开此选项会降低性能，应该只在必要时使用。

最大宽度 此参数通过一个预定义的值来限制扩展信息窗口（测量用 CSS 像素）。只有当 Type 参数设置为 Annotation 时才可用，它被设置为默认值为 250。

物体变换 API

注解: 请确保您要变换的物体是[动态物体](#).

使用 `transform` 模块的下列方法在引擎中移动物体:

`get_translation()` 获得在世界空间的物体中心坐标。带一个参数的方法返回一个新向量（一个非优化选项），同时用两个参数的方法需要一个额外的矢量记录结果。

`get_translation_rel()` 类似于 `get_translation()` 方法，但如果此物体有父级，得到的坐标在父级的空间测量。

`set_translation()`, `set_translation_v()` 移动物体的中心到指定的位置。第一种方法需要单独的坐标作为参数，而第二个需要三元矢量（`Array` 或 `Float32Array`）。

`set_translation_rel()`, `set_translation_rel_v()` 类似 `set_translation()` 和 `set_translation_v()`，但如果这个物体有父级，设定坐标为测量父级的空间。

`get_rotation()` 获取在世界空间的物体的旋转四元数。类似 `get_translation()`，调用此函数也有两种选择。

`get_rotation_rel()` 获取物体在其父级空间的旋转四元数。类似 `get_translation_rel()`，调用此函数也有两种选择。

`set_rotation()`, `set_rotation_v()` 设置在世界空间物体的旋转四元数。第一个功能需要单独的坐标参数，而第二个需要一个四元矢量（`Array` 或 `Float32Array`）。

`set_rotation_rel()`, `set_rotation_rel_v()` 设置物体在其父的空间中的旋转四元数。第一个功能需要单独的坐标参数，而第二个需要一个四元矢量（`Array` 或 `Float32Array`）。

`get_scale()` 获取物体在世界空间的缩放。

`get_scale_rel()` 获取物体在其父的空间的缩放。

`set_scale()` 设置物体在世界空间的缩放。统一对应于物体的原始缩放。值小于统一值是平均缩小，比统一值大 - 扩大。注意，不是所有的物体可以被缩放。特别是不允许物理物体缩放。

`set_scale_rel()` 设置物体在其父的空间的缩放。

`set_rotation_euler()`, `set_rotation_euler_v()` 使用欧拉角设置物体的旋转。一个 intrinsic YZX 系统被使用 (这意味着角遵循 YZX 顺序和坐标原点旋转，并且每角采取了一个新的位置)。

`set_rotation_euler_rel()`, `set_rotation_euler_rel_v()` 使用欧拉角在其父的空间中设置物体的旋转。

`move_local()` 相对移动到原来的位置的对象 (在局部空间)。

`rotate_x_local()`, `rotate_y_local()`, `rotate_z_local()` 相对于物体原来位置的 (在本地空间) 旋转。

获取物体 API

对物体执行任何操作，您首先需要得到它 (找到到它的链接)。有几个 API 函数是做这个的。有链接的物体具有 Object3D 类型。

`get_object_by_name()` 按名称获取物体。

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

`get_object_by_dupli_name()` 通过空名称和 DUPLI 名称找复制物体。

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

`get_object_by_dupli_name_list()` 通过空名称和 DUPLI 名单 (String 类型元素的数组) 获得复制物体。

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

`get_object_name_hierarchy()` 用一个给定的物体，返回物体名层次队列 (从最高层的父物体到物体本身)。

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

`check_object_by_name()` 检查是否具有给定名称的物体在场景存在。

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

`check_object_by_dupli_name()` 检查复制物体是空的名字和 DUPLI 名字出现在场景。

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

`check_object_by_dupli_name_list()` 检查复制物体是空的名字和 DUPLI 名单 (String 类型元素的数组) 出现在场景。

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

`get_object_name()` 获取物体的名称。

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

物体选择

为了使一个特定的物体能被选择，需要 Selection and Outlining 面板上的 Selectable 复选框启用。

注解: 确保在 Scene > Object Outlining 面板被设置为 ON 或 AUTO。

物体选择也可以通过 API 编程方式，例如，在 scenes.js 模块有基于二维的画布坐标选择物体的 `pick_object` 函数。

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

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

或使用[逻辑编辑器](#).

如果选择的物体已启用了 Object > Selection 和轮廓面板上的 Enable Outlining 和 Outline on Select 选框，那么 pick_object 函数调用将激活外轮廓辉光动画。

注解: 如果选定的物体是透明的 (Blend, Add 和 Sort 透明度类型)，外轮廓发光只能在具有 Alpha 高于 0.5 的部分可见。

复制物体（实例化）

它通常需要复制（使的实例）中的应用程序的工作物体。

复制物体也有其局限性:

- 只有 MESH 物体可以被复制
- 物体应该是[动态](#)（启用 渲染属性 > 强制动态 Object）
- 源物体应该属于活动场景

制作一个简单的复制

在简单复制的情况下，新的物体将共享与原始物体的网格。因此，如果原始物体的网格被改变，则复制的物体的网格也将被改变。为了让简单复制可能的，但它足以打开“Blend4Web> 强制动态 Object”源物体的属性设置。

制作深复制

在深度复制的情况下，新物体将具有独特的性质，即它会有自己的网格。因此，如果原始物体的网格被改变，则复制的物体的网格将不被改变。另外，在复制的物体在画布纹理不同的纹理和同一个像它与简单复制的情况。为了使深度复制可能，要求源物体启用渲染属性 > 动态几何 复选框。|

在运行时复制物体可以使用 objects.js 模块的 copy 方法进行。此方法需要三个参数：源物体的 id，新物体一个唯一的名称和布尔值，以指定复制模式（即简单或深度）。默认情况下，简单复制将被执行。

新创建的物体，应添加到场景。这可以用 scenes.js 模块的 append_object 方法进行。新物体应该被传递给它作为参数。

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

删除物体

要删除物体，请使用 scenes.js 模块的 remove_object 方法。传递物体作为参数。动态网格-空类型的物体可以用这种方式删除。

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

四元数

四元数是用于执行旋转的四分量矢量。四元数优于其它旋转方法，例如：

- 四元数没有多解，并且不像欧拉角依赖于的旋转顺序。
- 四元数的内存使用是更有效的（少 2-4 倍，取决于所使用的矩阵）。
- 在一系列旋转的情况下，比矩阵有更好的计算效率。

- 数值稳定 - 从浮点数不准确而产生乘法错误的补偿。
- 便捷的插值方法。

四元有一些缺点:

- 用四元数来旋转向量比用矩阵来旋转向量要浪费计算成本。
- 很难将四元数用到非旋转变换上（例如透视或正交投影）。

该引擎具有许多功能，能更方便地跟四元数工作：

`quat.multiply` 四元数乘法。需要注意的是用 B 四元数左乘 A 四元数 $A * B$ 是用 A 旋转物体。比如，物体已经具有一定的旋转 B 上，我们加了新的 A 旋转。

`quat.setAxisAngle` 四元数是任意相对于任意轴（矢量）的任意角度旋转的另一种表现。

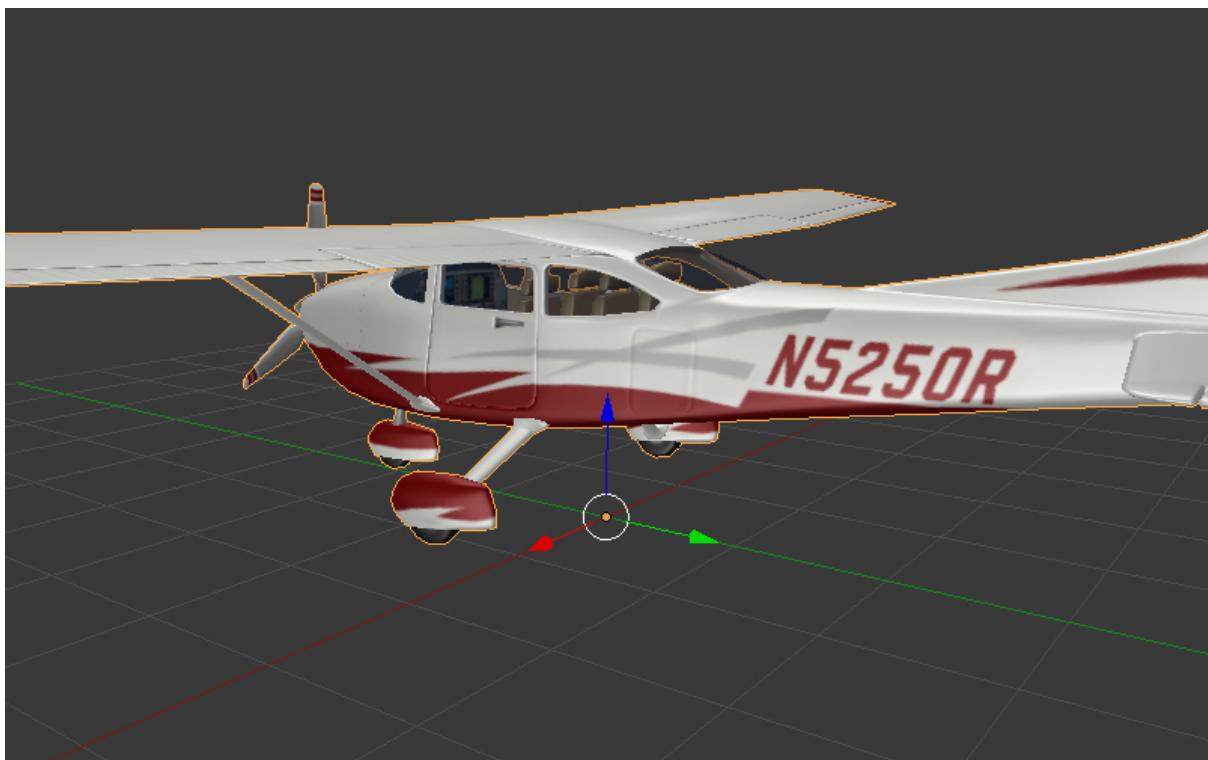
从向量的末端来观看时旋转的正方向定义为逆时针。例如 `quat.setAxisAngle([1, 0, 0], Math.PI/2, quat)` 用四元数对物体相对于 X 轴旋转了 90 度（如果从 X 轴末端查看是逆时针）。

`quat.slerp` 四元数球面插值。用于平滑物体的旋转和动画。

`util.euler_to_quat`, `util.quat_to_euler`. 从欧拉角转换到和转换回欧拉角。

四元数例子

我们需要在水平面向右旋转物体 60 度。我们有一个在 Blender 里有个名为“Cessna”的模型。



让我们在 `aircraft` 变量保存引用：

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

旋转：

- 顺时针转动对应于旋转到右侧（即在负方向）。
- 60 度 $= \pi/3$ 弧度.

因此，我们得到：

```
// 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);
```

优化的版本不创建新的物体：

```
// 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);
```

通过 TSR 矢量移动

有时用下面格式的矢量来移动物体是很方便的：

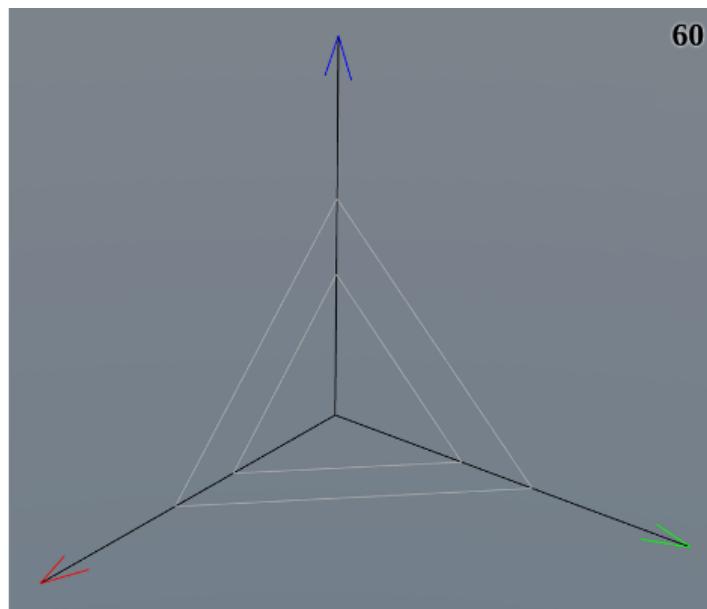
$$[T_x, T_y, T_z, S, R_x, R_y, R_z, R_w]$$

在这里 T_x, T_y, T_z - 位移向量的分量, S - 比例系数, R_x, R_y, R_z, R_w - 四元数向量的分量。这个向量故名：TSR 或 TSR-8。

该向量可通过 `tsr` 模块以及经由 `transform` 模块的 `set_tsr()`/`get_tsr()` 的方法操作。

线渲染

Blend4Web 引擎还具有功能选项能够渲染点集定义的线。



用于渲染线的 API 方法位于 `geometry` 模块，可以设置渲染线的样式（即其颜色和厚度）需要用 `set_line_params()` 的方法，位于 `material` 模块。

若要使用线条的渲染效果，至少需要在场景中存在一个 `Empty` 类型物体启用了 线渲染选项。

要渲染一条线，你首先必须 [获取链接](#) 到 `Empty` 物体。此物体的起始点将用作线在渲染时的中心坐标。

可以通过调用 `draw_line()` API 方法渲染线条。其第一个参数是指向 `Empty` 物体（见上面），而第二个是用于建立一条线的一列顶点数组。该方法也拥有第三个可选的参数，定义渲染模式`¶`方法由成对的顶点定义生成一条或一组线（后一种情况下的顶点总数应该是偶数）。

如果一个 `Empty` 物体在一条线渲染中不止一次使用，只有最后一个将可见。要渲染多条线，您将需要使用多个 `Empty` 物体。

下面的示例演示了使用线渲染上面的图片中的坐标轴物体的一部分`¶`

```

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, 5, 0, 0, 0, 0, 0, 0, 5]);

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

这个代码列表生成只有物体的主轴，因为列出所有元素将都使代码过长和重复。绘制物体的其他元素它以同样的方式。

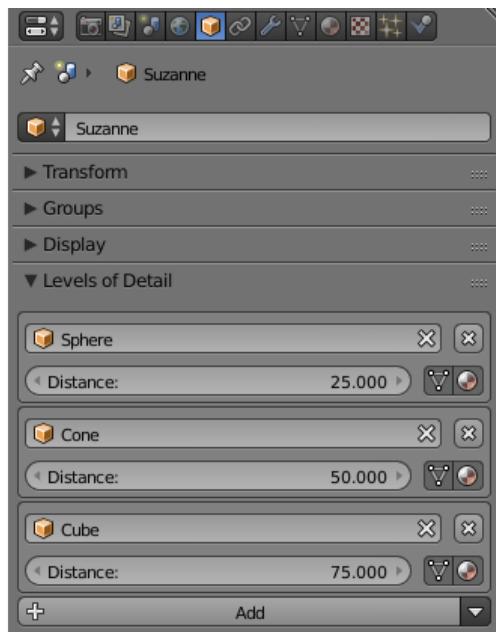
Levels Of Detail

Overview

LODs, or levels of detail, are versions of a single object with various amounts of detail. Levels of detail are used for the purpose of optimization, and the idea here is that if an object is far from the camera, you can render more simple version of it –for example, the one with less polygons and simpler material. This can reduce the burden on the GPU without decreasing the quality of the render, as most of the details of the original object won’t be visible from the distance.

LODs are often used in large-scale scenes with high object count. More compact scenes may not gain a major performance boost from using this feature. Video game locations such as islands, countrysides or town streets are most suitable for using LODs.

LODs can be set up directly in Blender at the Levels of Detail [panel](#).



There, you can set levels of detail and distances at which they will be used for an individual object. Each level of detail is a separate object. The main rule is obvious: the lower the level of detail is positioned in the list, the simpler the corresponding object should be.

You can make your objects simpler in different ways: by decreasing its polycount, by making the materials less complex (this is especially true for node materials) or by turning off various effect such as [shadows](#), [animations](#) or [Wind Bending](#).

The number of levels can be set as you see fit, depending on the overall scale of the scene.



Figure 10.1: Levels of detail for trees: original object at the left, its LOD on the right

Specifics and Differences From BGE

Setting up LODs and using them in the engine has some specifics. Also, BGE mode in Blender has its own LOD system that is mostly identical to the one used in Blend4Web, but still has some differences that are mentioned below.

1. When you are applying a LOD object to some other object, the position of the LOD object will not change. So, if you want the objects to occupy the same place in the runtime, you should manually place them in the same location in Blender. This is different from BGE, where a selected LOD object is automatically moved to the center of the main object.
2. If you need to set the same LOD object for multiple different objects (for example, same-type trees, building, cars or shrubs), you should make a copy of it for every object (i.e., make as many copies as there are objects that will use this object as a LOD) and set these copies as LODs. Unlike BGE, copies are not generated automatically when you select the same LOD object for multiple base objects. To simplify creating same-type object, we recommend to put the base object and all its LODs to one group and then duplicate this group using Empty objects.
3. If the object is supposed to change its position in the runtime, you should attach its LODs to it in Blender using parent-child link so LOD object would copy its transformations.

Smooth LOD Switching

One of the most notable shortcomings of the LOD system is the abrupt switching between levels of detail which is easy to notice and can be annoying.

The engine supports smooth transition between LODs based on alpha test.

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

It is intended for enabling smooth LOD transition and setting object types that this transition will be applied to. It should be noted that enabling this option might significantly decrease performance (it depends on the number of LOD objects and their materials).

注解: Smooth LOD transition is not always required. For example, if you instantly



Figure 10.2: Smooth transition between the original object and its LOD.

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.

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

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

Hysteresis

LODs are switched at certain distances from the object. These level borders are set up on the Levels of Detail panel. If the camera is moving near a border like this, it can cross it often, which results in frequent LOD switching. Under such circumstances, LOD switching becomes very apparent and thus undesirable.

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.

For example, if Max LOD Hysteresis Interval is set to 4 meters while the distance between detail levels is equal to the 20 meters, one level will be turned on at the distance of 18 meters and the other, at the distance of 22 meters. Using this option eliminates frequent LOD switching, as levels of detail no longer have definite border between them. The setting is adaptive and can adjust to different LOD distances, reducing the set interval, if needed.

网格

目录

- 网格
 - 静态和动态网格
 - 设置
 - * 覆盖包围体积
 - 变形

网格是的obje 物体 的一个子类。网格从某种意义上讲跟其他物体类型是不一样的，尽管他们都是在场景中可见的几何结构。用户在一个场景中看到的所有物体都是网格或在导出时 (比如 CURVE, TEXT, METABALL 和 SURFACE 类型的物体) 转换成网格的。

Blend4Web 插件 有几个工具用于网格编辑，包括法线编辑器.

这一章概述了 MESH 类型物体参数和设置，以及他们一起工作的 API 函数。

静态和动态网格

所有 MESH 物体可分为静态和动态。

静态物体是物体，具有相同的材质的网格可以合并在一起。

动态物体是物体，网格不可以互相合并。静态网格的合并-所谓的批处理-在场景被加载时自动执行，以优化绘制调用的次数。即使在场景中只有一个物体，也执行转换。由此产生的网格的中心位于原点。

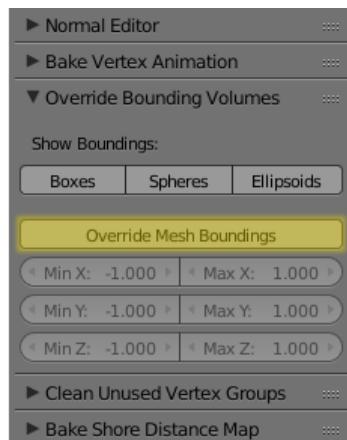
具有动画，物理或父级关系的物体，是一个动态物体，都被认为是动态的。

设置

网格含有**物体** 所有的设置和一些额外的设置。

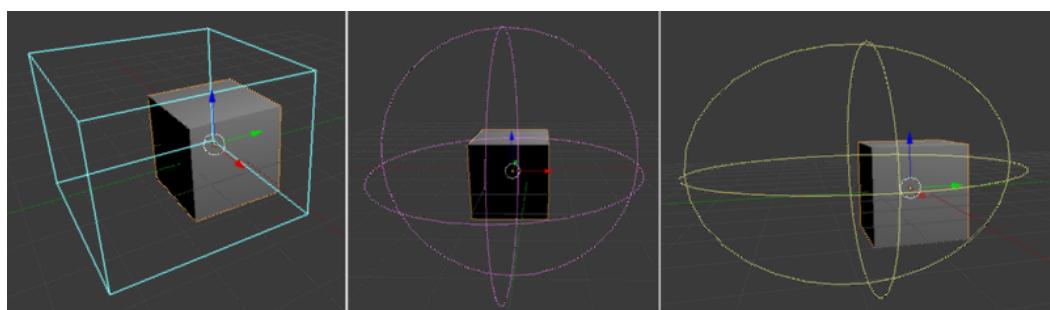
覆盖包围体积

这部分是用来覆盖包围盒。这样体积盒用于检查物体的可见性，来处理碰撞和物理交互。通过编辑它们，可以实现各种效果。



该 覆盖包围体积部分可以在 Blend4Web 面板上找到（在法线编辑器 旁边）。

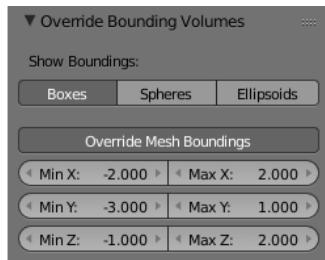
显示包围



显示在 Blender 视口中的包围盒。只有当 覆盖网格包围参数开启，启用该选项。包围盒可以呈现为长方体或为圆形或椭圆的组合。多个渲染方式可以在同一时间是启用（甚至所有三种方法可以同时启用）。

注解: 所选渲染方法不会影响包围盒的形状，只是影响它在视口中展示的方式。

覆盖网格包围



此按钮可以使基本网格包围覆盖。包围盒总是一个长方体并以物体的枢轴点为中心。

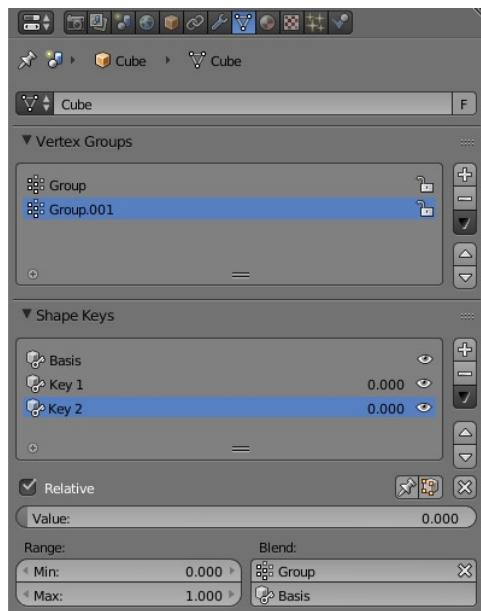
MinX and MaxX 包围盒的 X 尺寸。默认情况下，MinX = -1, MaxX = 1

MinY andMaxY 包围盒的 Y 尺寸。缺省情况下，MinY = -1, MaxY = 1.

MinZ and MaxZ 包围盒的 z 方向。缺省情况下，MinZ = -1, MaxZ = 1

变形

变形目标可以用 Blender 的标准 网格 > 形变键界面进行添加。



该引擎支持所有“相对”类型下的形变键选项。

要设置一个形变键值，使用 `geometry.js` 模块的 `apply_shape_key` 方法。

注解：该物体必须具有 导出形变键参数启用。

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

法线编辑器

目录

- 法线编辑器
 - 法线编辑器主要功能
 - 界面
 - 启用
 - 显示法线
 - 旋转
 - 绝对和偏移模式
 - 分割法线
 - 平均
 - 恢复
 - Tree (树), Foliage (树叶) 和 Face (面)
 - Copy (复制) /Paste (粘贴)
 - 从网格复制

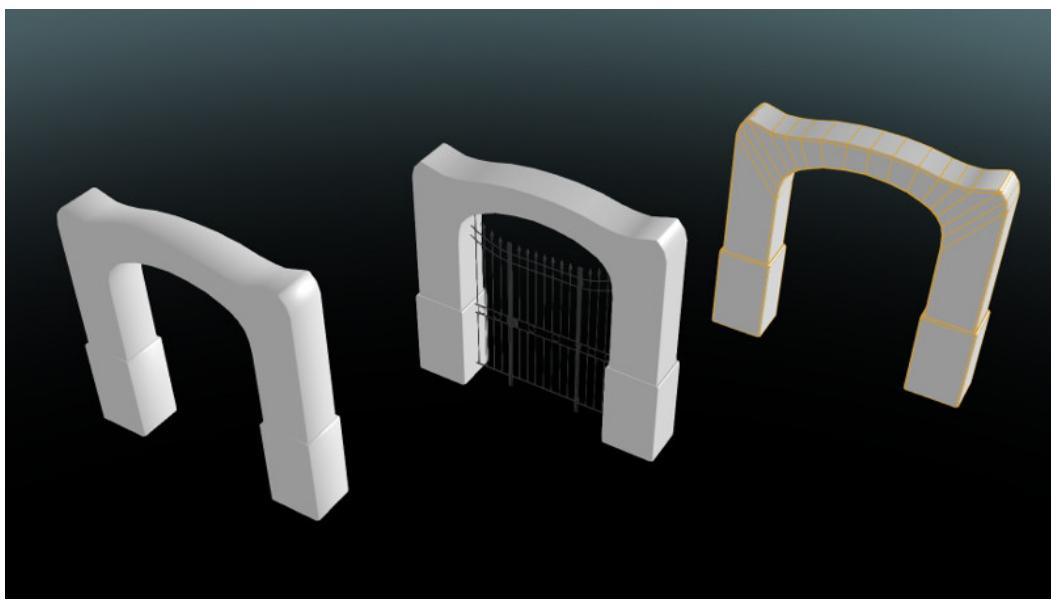
编辑法线来更改模型的著色而不更改其几何形状，是相当简单而有效的方法。

在某些情况下，使用法线编辑器，你可以做到这类似于使用法线贴图 的效果。同等的，编辑法线是首选，因为它是更有效的计算和消耗更少的显存。

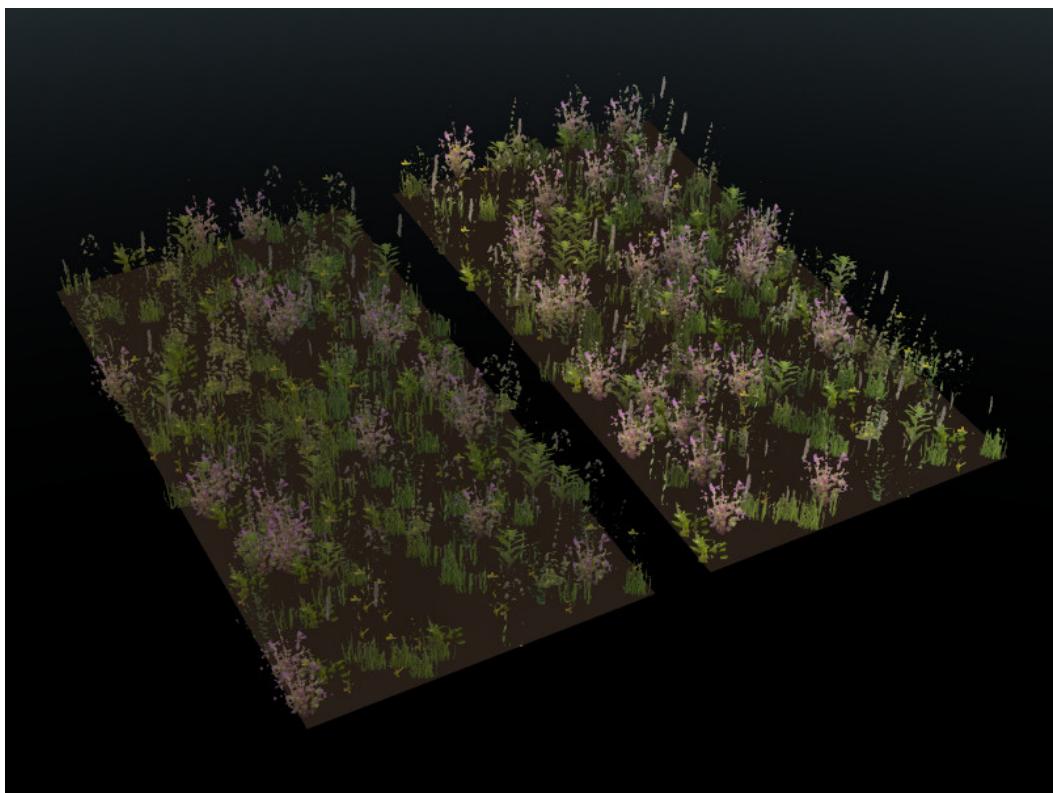
法线编辑器工作流示例：



左边-添加默认几何材质的树，右边-同一棵树但编辑过法线。



左边-普通着色的门，中间-带法线编辑的门，右边-带线框的门的几何形状。



左边-默认材质的草，右边-同样的草并编辑过法线。



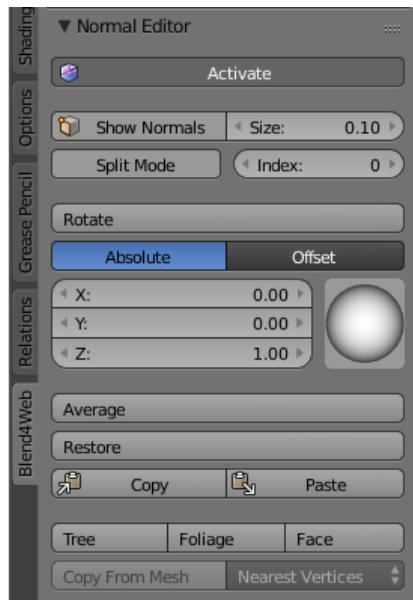
左边-默认材质的眼镜，右边-编辑过法线的眼镜。

法线编辑器主要功能

1. Blender 内的 datablock 数据块（2.74 版后可用）被用作于存储法线；
2. 法线编辑和由此产生的材质在 Edit Mode（编辑模式）下显示；
3. 所有更改应用全都立即自动保存；
4. 可在 Viewport 视图窗口通过 Shift+Ctrl+R 热键可直接旋转顶点法线，就像任何其他 Blender 里的旋转；
5. 编辑的法线会自动导出

界面

法线编辑器位于工具面板 Blend4Web > Normal Editor 上。开始编辑工作前 Shading: Smooth 模式应该启用和 Activate 按钮应该点击，或 自动 Smooth 标志应在网格设置中启用。



启用

该 Activate 按钮打开顶点法线编辑模式。

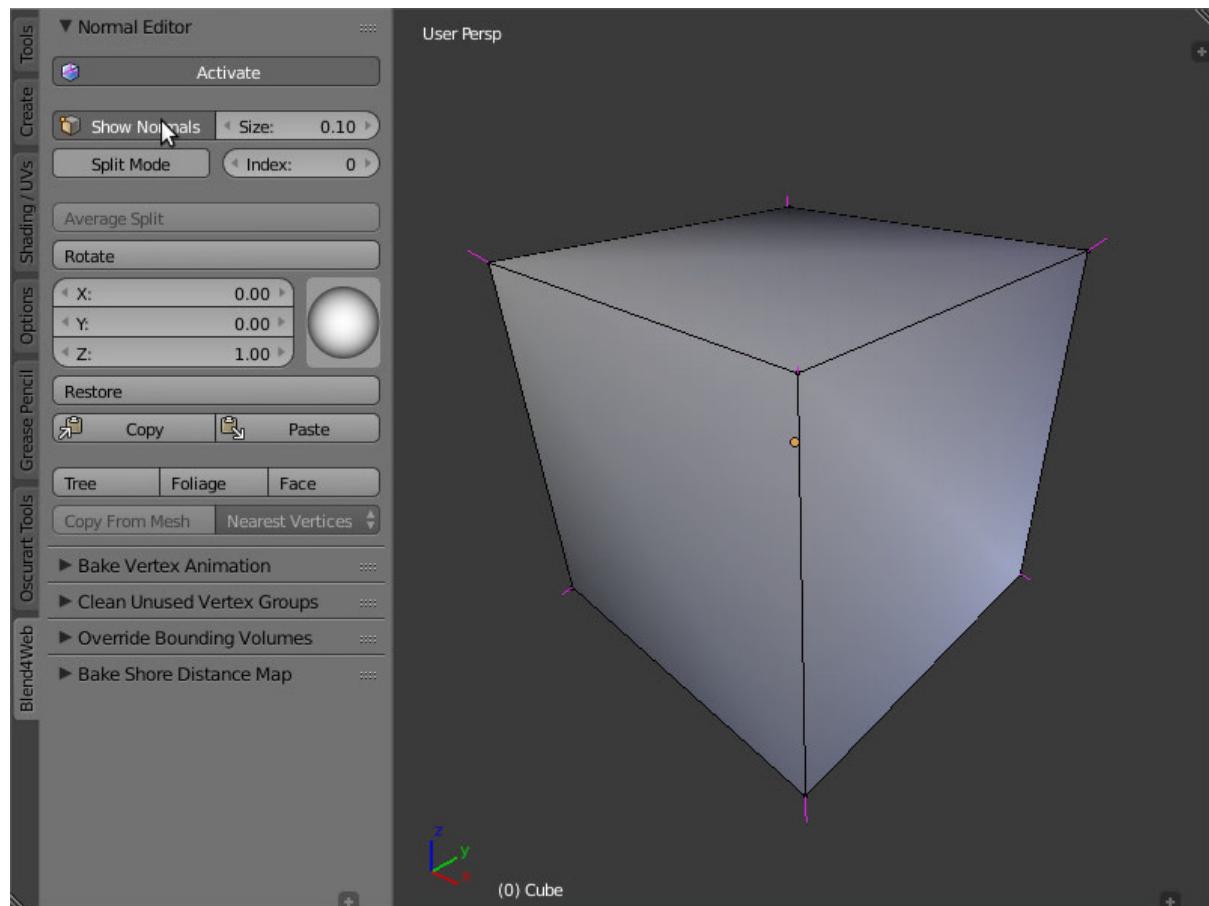
编辑法线需要在编辑模式下 Activate 启用按钮。启用后，物体的材质和导出将使用编

辑过的法线。换句话说，要在 Blend4Web 视图里查看根据材质的编辑后法线，此按钮必须处于启用状态。

显示法线

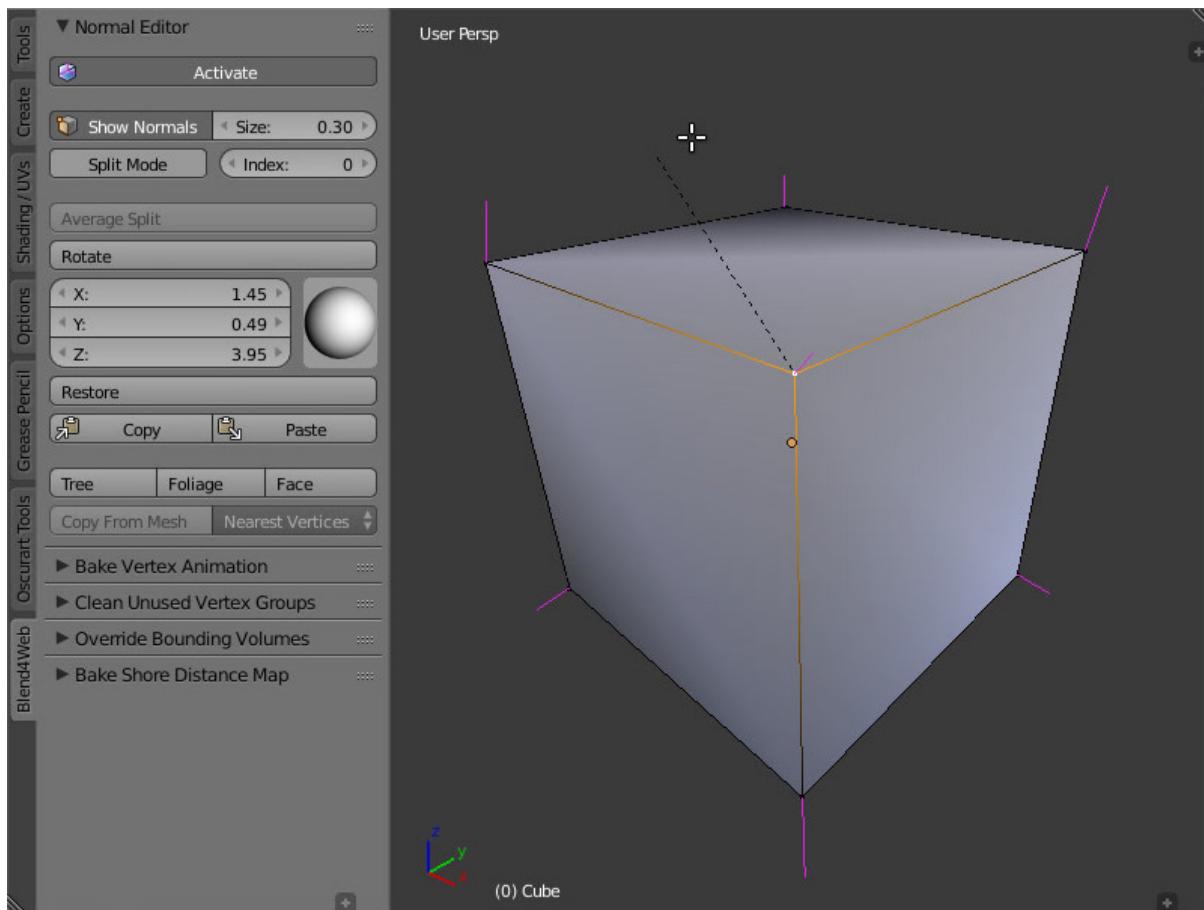
此按钮实际上重复原始 Blender 按钮。在窗口里打开显示法线，而 Size (尺寸) 栏允许您设置它们的长度。

您只要按下法线编辑器里的 Show Normals 按钮或 Blender 右侧面板中的 Mesh Display (网格显示) 里的法线显示。您也可以经由设置 Size 栏来调整合适的法线长度。



旋转

使用这些工具，您可以更改法线的方向。Rotate (旋转) 的功能也可通过 Shift+Ctrl+R 热键，允许类似于 Blender 的顶点法线旋转。



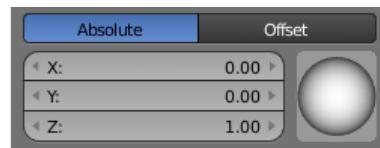
选择您想要编辑的一个或多个顶点，然后使用圆形视觉操纵器指定他们的法线方向数值。

Rotate 按钮提供了更简便的方法来操纵法线。旋转在屏幕空间执行。然而，正如任何在 Blender 的其他旋转，您可在旋转过程中隔离所需的轴（通过键入 X、Y 或 Z）。

绝对和偏移模式

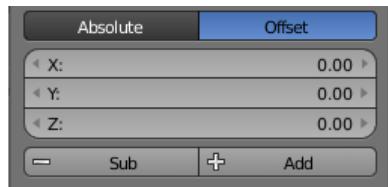
法线编辑器可以在两种不同的模式进行操作：绝对模式和补偿模式。当前选择的模式可以按已上述的“Rotate”按钮下放置在两个按钮中的一个进行切换。默认情况下，使用“Absolute”模式。

绝对模式



这是法线编辑器的默认模式。在这种模式下，一法线矢量的坐标是绝对的，也可以使用 X, Y 和 Z 字段或位于它们的旁边的视觉范围进行设置。

偏置模式

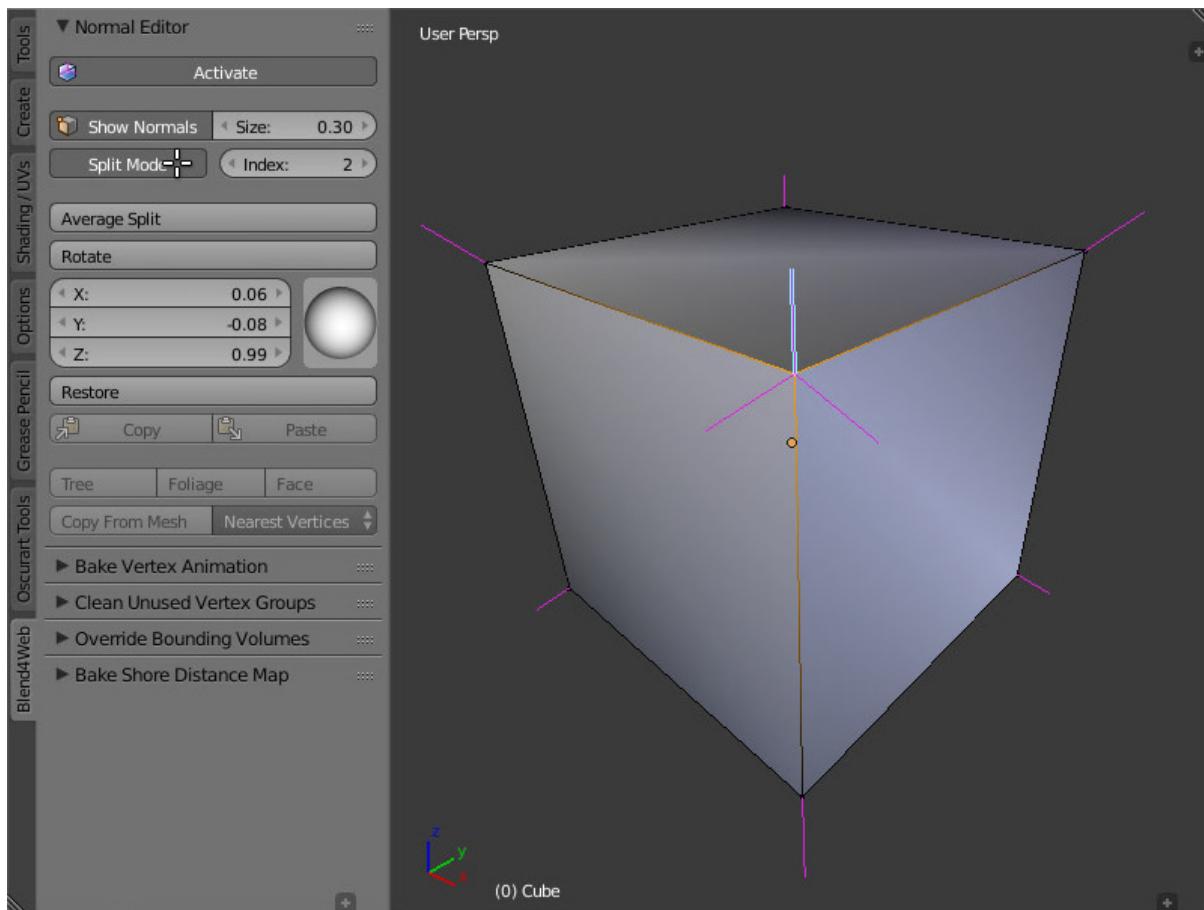


在这种模式下，一个用户定义的值被添加到法线矢量。

在 X, Y 和 Z 字段可用于设置向量的相应分量，而 Sub 和 Add 按钮定义是否该矢量应该从法线矢量中减去或加至它。

分割法线

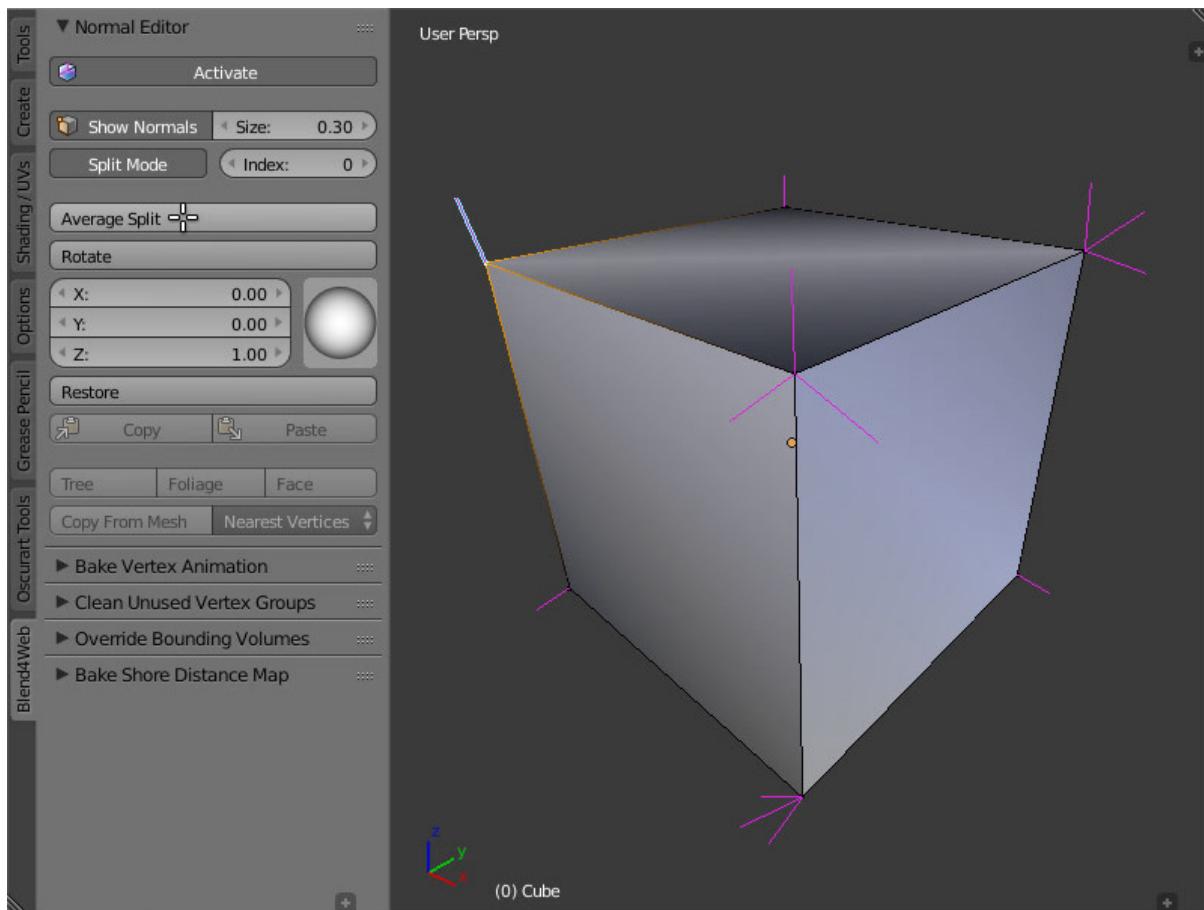
Split Normals (分割法线) 模式允许分别编辑每一面的顶点法线。Index 索引栏允许您在分开的法线间切换。



打开 Split Normals 模式、选择顶点并更改其法线方向。首先，索引为零的法线将被修改。接着，可以通过切换索引到下一个顶点法线并编辑它，下一步亦是如此。

平均

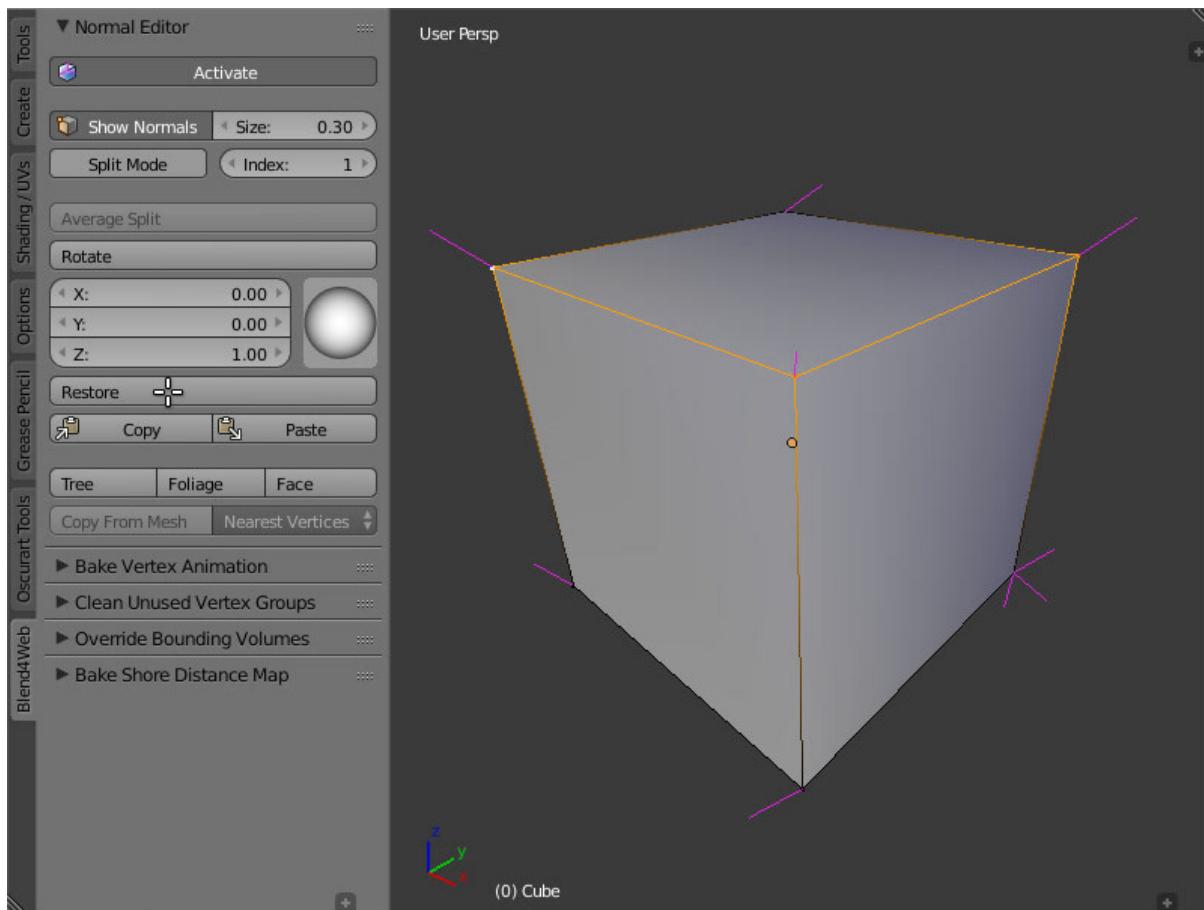
在 Average 按钮平均值其中被修改的顶点法线的方向。



结合几个分割顶点法线，为了获得这些法线的平均方向，您只需要选择所需的顶点，然后按 Average Split 按钮。

恢复

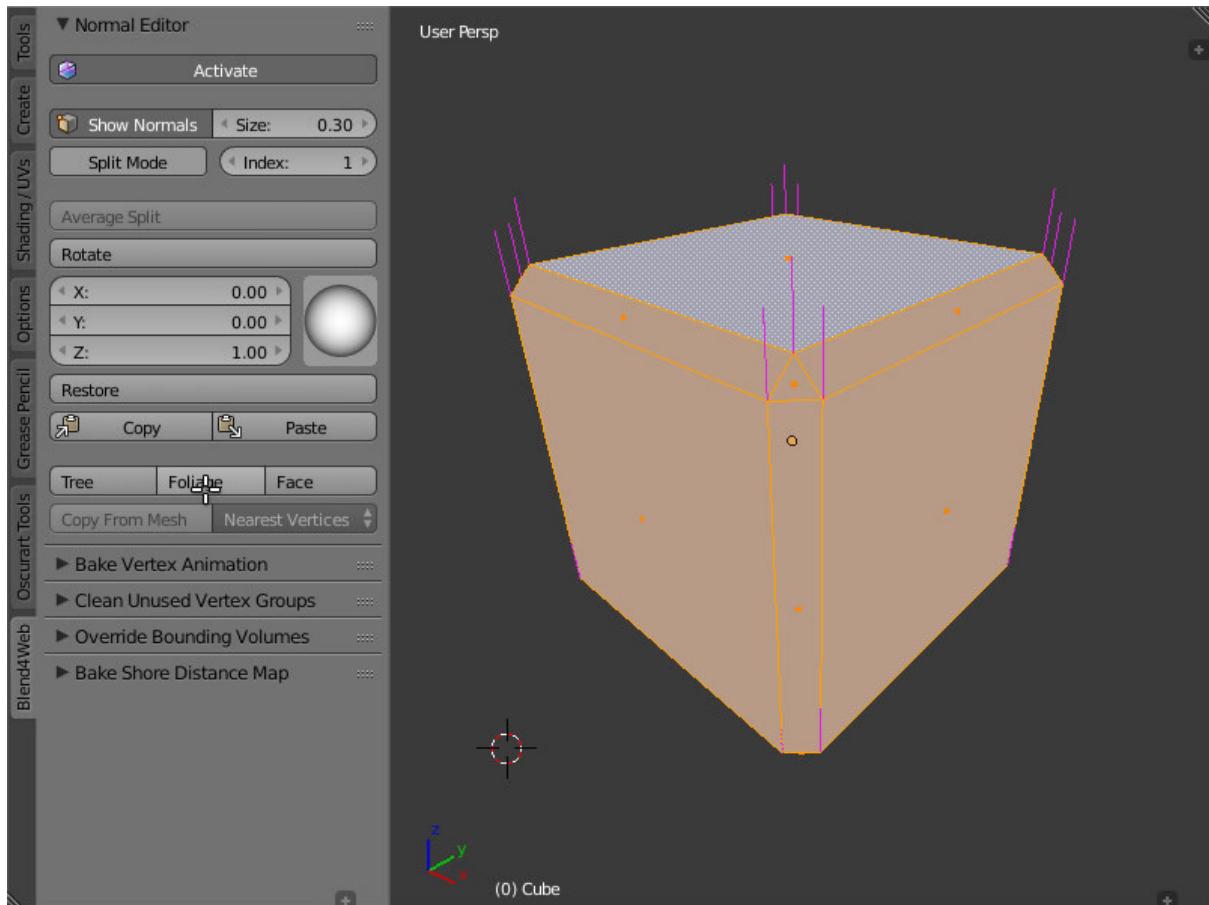
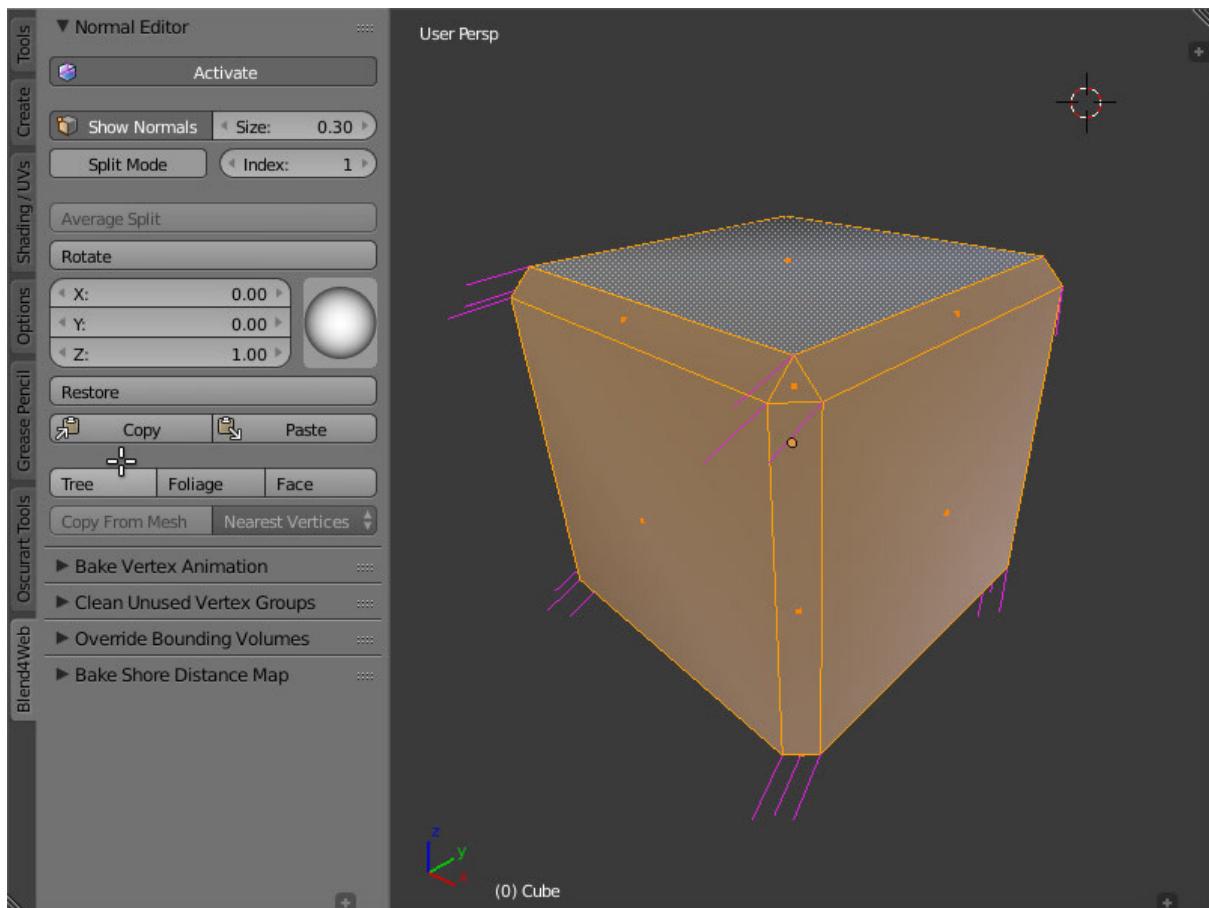
Restore (还原) 按钮将还原所选顶点法线的原始方向。

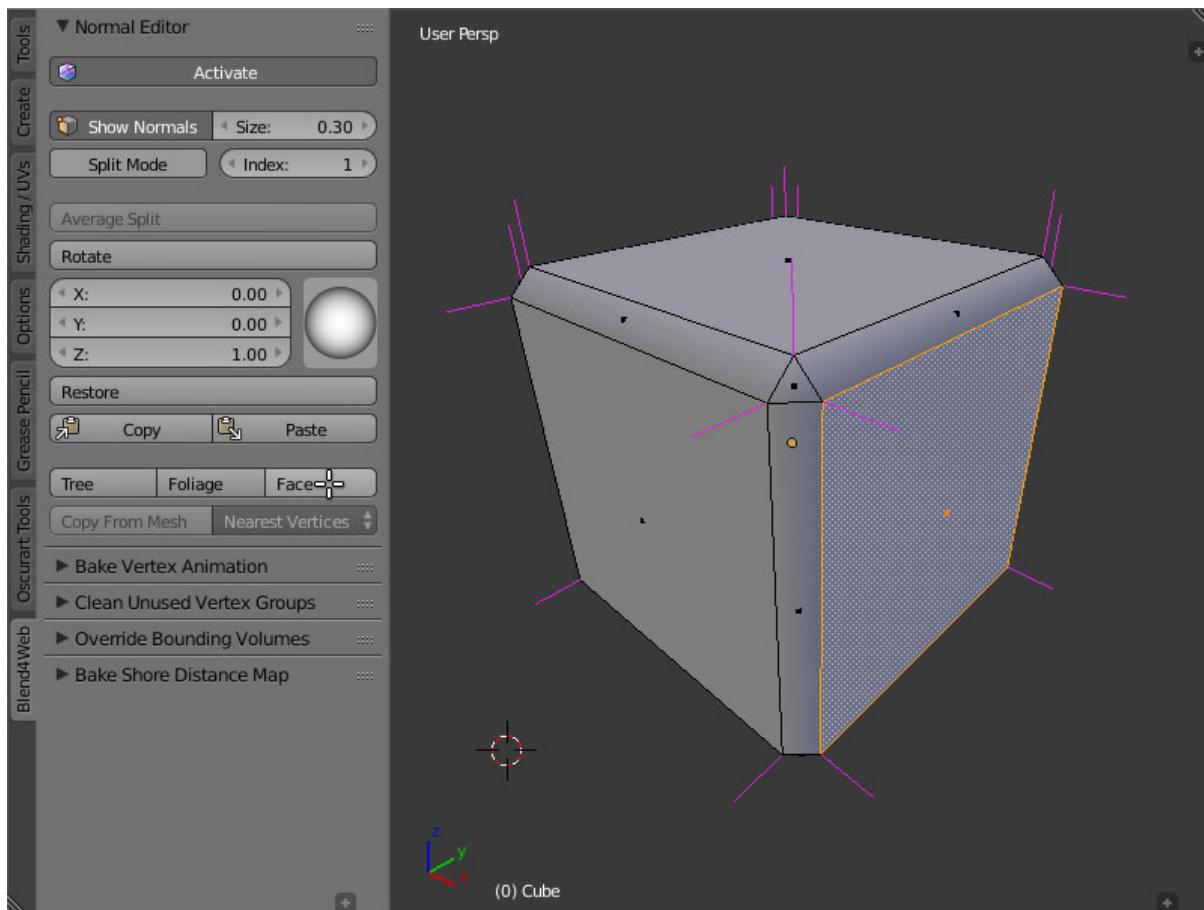


为了还原到其原始 (根据面的法线计算) 的法线的方向, 您需要选择所需的顶点, 然后单击 **Restore** 按钮。

Tree (树), Foliage (树叶) 和 Face (面)

Tree 按钮将选择的顶点法线导离 3D 游标。**Foliage** 按钮将法线硬直地朝上。**Face** 按钮将法线方向平行于面的法线。





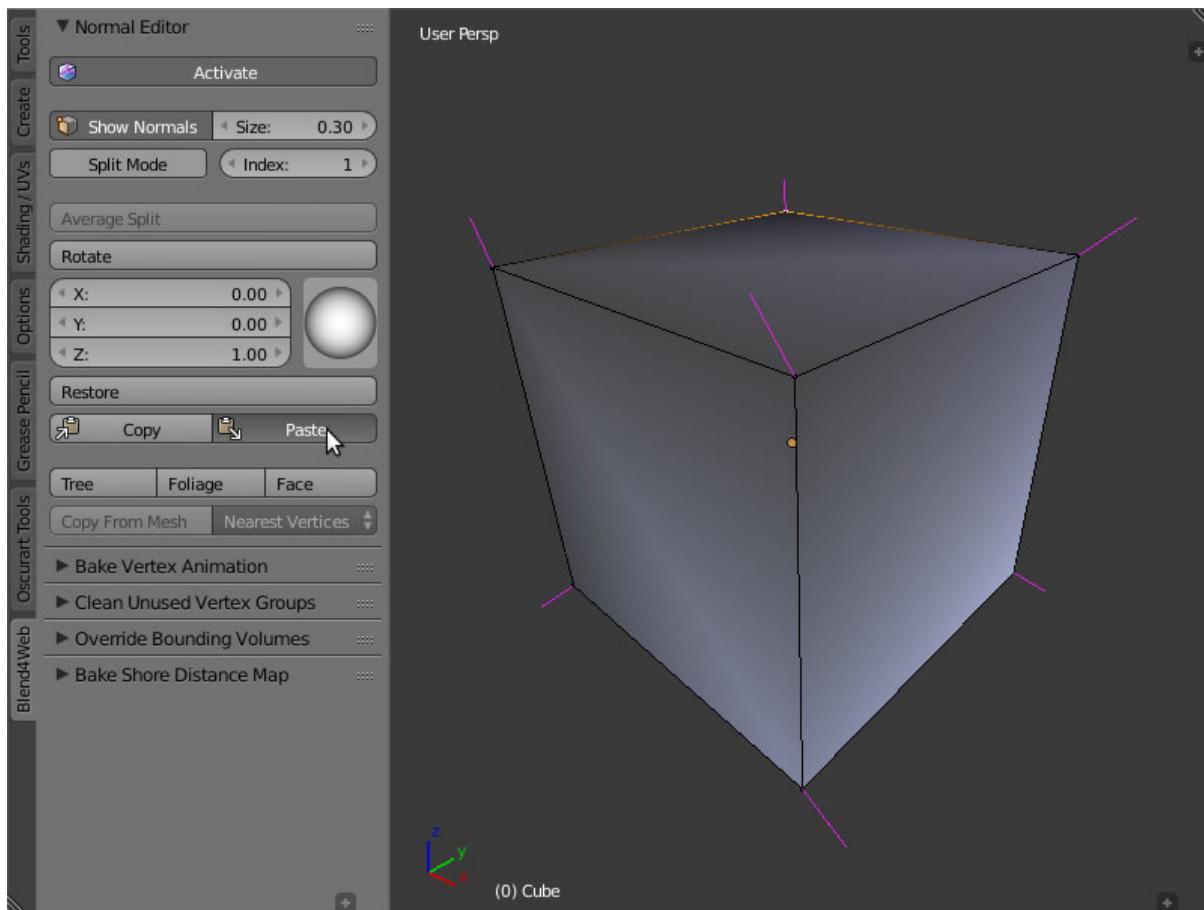
为了使用 Tree 功能, 请选择所需的顶点并将 3D 游标放于所需位置。然后单击 Tree 按钮, 接着所有选定的顶点仿佛他们从一个点射出般, 均朝着游标的反方向。

Foliage 功能相当容易使用: 只需选择顶点按下按钮, 此功能会将它们的法线笔直地朝上。

为了将法线方向平行于面的法向量, 只需选取所要的面并单击 Face 按钮。面的顶点法线将平行朝向面的法线。此功能仅适用于一次选取一个面。

Copy (复制) /Paste (粘贴)

拷贝一个到另从一个顶点的法线方向。

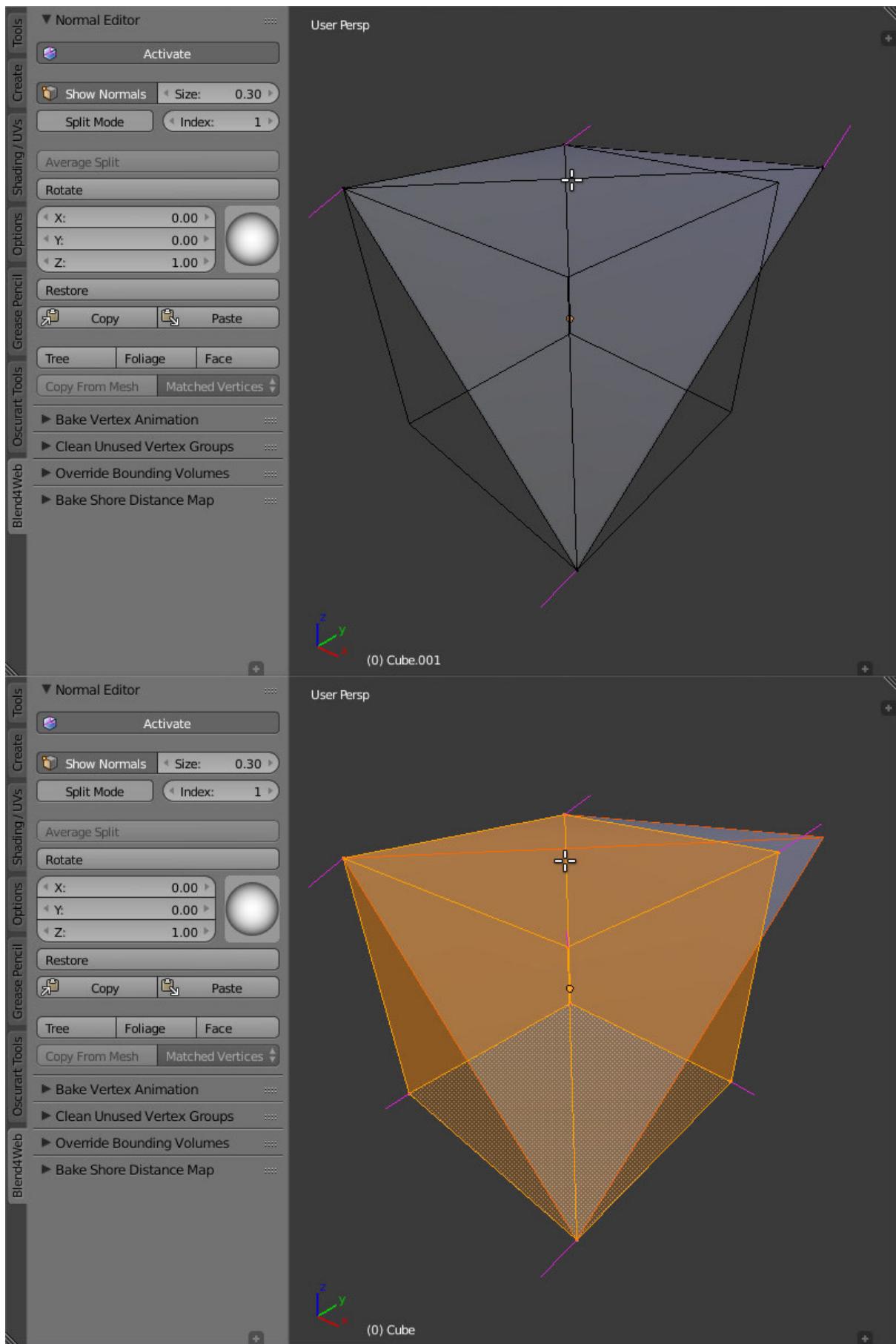


选择您想要复制的来源顶点后单击 Copy 按钮。接着选择您想要复制的顶点单击 Paste 按钮。您可从所选的顶点复制信息到许多不同的顶点上。此按钮在 Split Mode 不启用，亦无法复制分割顶点的数据。

从网格复制

这个功能允许您从一个物体复制法线到另一个上。有两种模式: Matched Vertices (相符顶点) 和 Nearest Vertices (最近顶点)。

Matched Vertices 模式只许在两个物体拥有相同的座标系时复制; 其余的顶点将被忽略。Nearest Vertices 模式是从来源物体顶点最靠近处复制法线。



此功能需在 Blender 物体模式中运作。首先，您需要选择复制的来源物体，接着是复制的对象。它也需要选择目标物体的顶点法线并复制到其中。

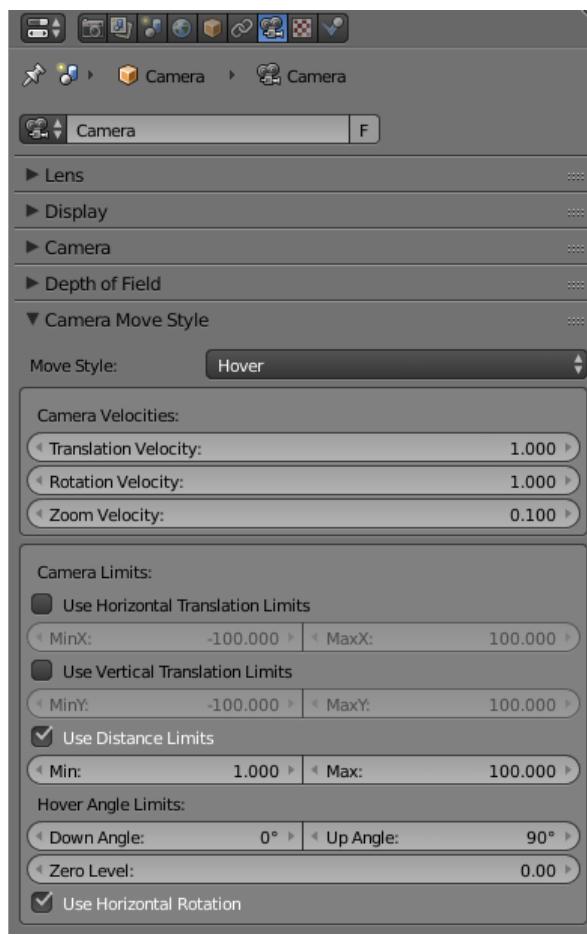
相机

目录

- 相机
 - 移动方式和一般设置
 - 限制摄像机的运动
 - * 目标类型相机
 - * 眼类型相机
 - * 悬停类型相机
 - 视口对齐
 - 相机控制 API
 - * 控制模式设置
 - * 平移和旋转
 - * 限制设置
 - * 角坐标计数

移动方式和一般设置

相机设置在 相机 (物体数据) 选项卡下 Properties 面板指定。



相机移动方式 > 移动方式 摄像机控制模式:

- **目标** 默认摄像机围绕一个固定点(目标)旋转。中心点的位置可以改变(见[相机摇动](#))。
- **眼睛** 在 Eye 模式允许以第一人称视角旋转和平移。
- **悬停** 在 Hover 模式, 相机将平行于水平面移动。通过使用附加的限制, 可以创建一个相机指向中心点的平滑移动路径。
- **静态** 在 Static 模式下, 相机可以通过动画或通过 API 调用移动。

看向光标 > 目标位置 在 Target 模式可用。这是在相机中心点的位置。该 看向 Cursor 按钮将当前的 3D 光标位置复制到该值。

景深 在[景深](#)章节中描述。

相机速度设置现在是特优

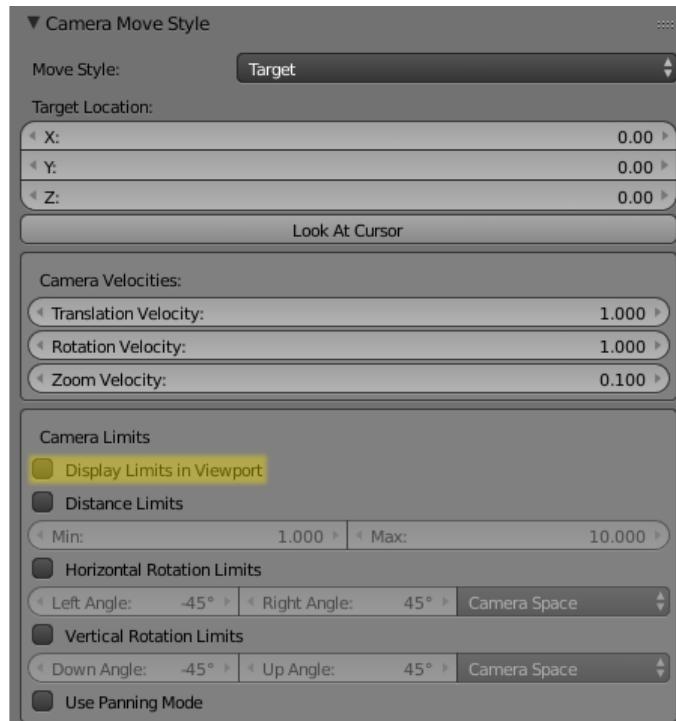
相机移动方式 > 相机速度 > 移动速度 提供 目标, 眼睛和 悬停型相机。设置摄像机平移的速度。可能的值区间: $[0, \infty)$ 。默认值是 1。

相机移动方式 > 相机速度 > 旋转速度 提供 目标, 眼睛和 悬停型相机。设置摄像机平移的旋转速度。可能的值区间: $[0, \infty)$ 。默认值是 1。

相机移动方式 > 相机速度 > 缩放速度 提供 目标和 悬停型相机。设置摄像机接近中心点的速度。可能的值区间: $[0, 0.99]$ 。默认值是 0.1。

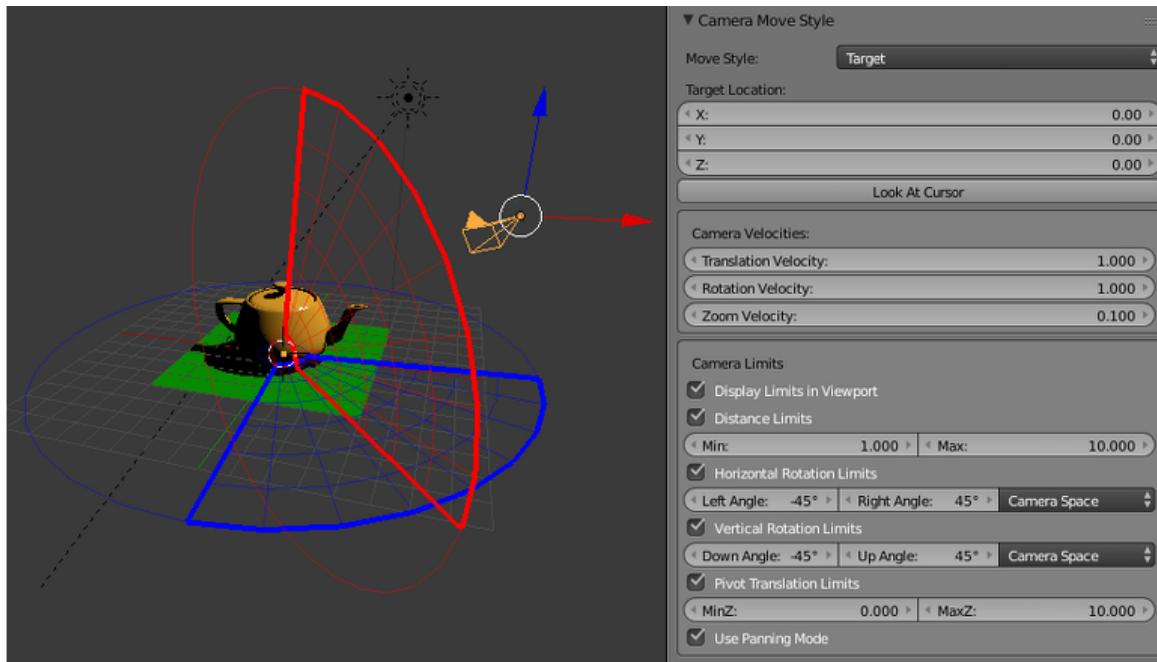
限制摄像机的运动

有几种限制/改变摄像机运动的设置。它们组合为 相机限制.

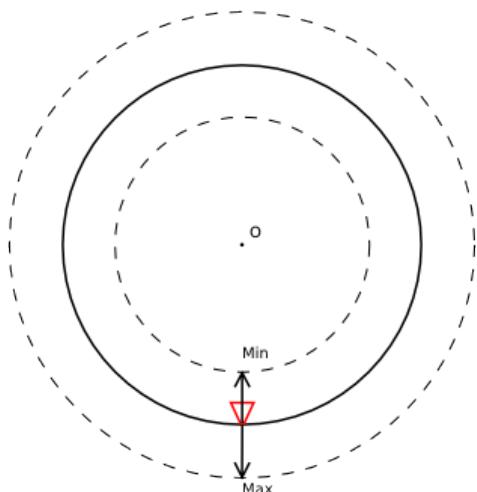


相机限制可以直接在 Blender 视图查看, 如果 Display Limits in Viewport 参数已启用。

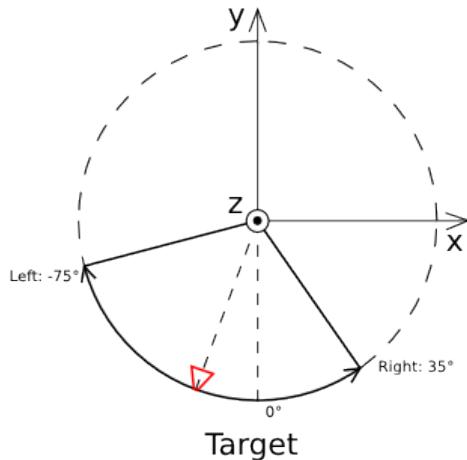
目标类型相机



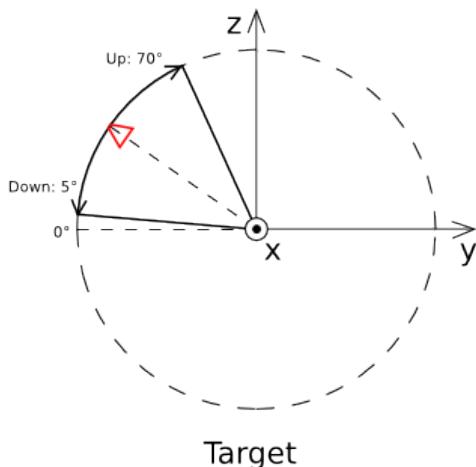
相机移动方式 > 相机限制 > 距离限制 从相机到中心点设置最小和最大距离。允许值：
最小 \leq 最大。默认值：最小 = 1, 最大 = 10. 默认情况下禁用。



相机移动方式 > 相机限制 > 水平旋转限制 限制摄像头水平（围绕在 Blender 的 Z 世界轴）对点的转动。旋转沿 左角 和 右角值之间的圆的弧进行。旋转弧对应于从 左角 到 右角逆时针运动。默认值：左角 = -45° , 直角 = 45° 。默认情况下禁用。



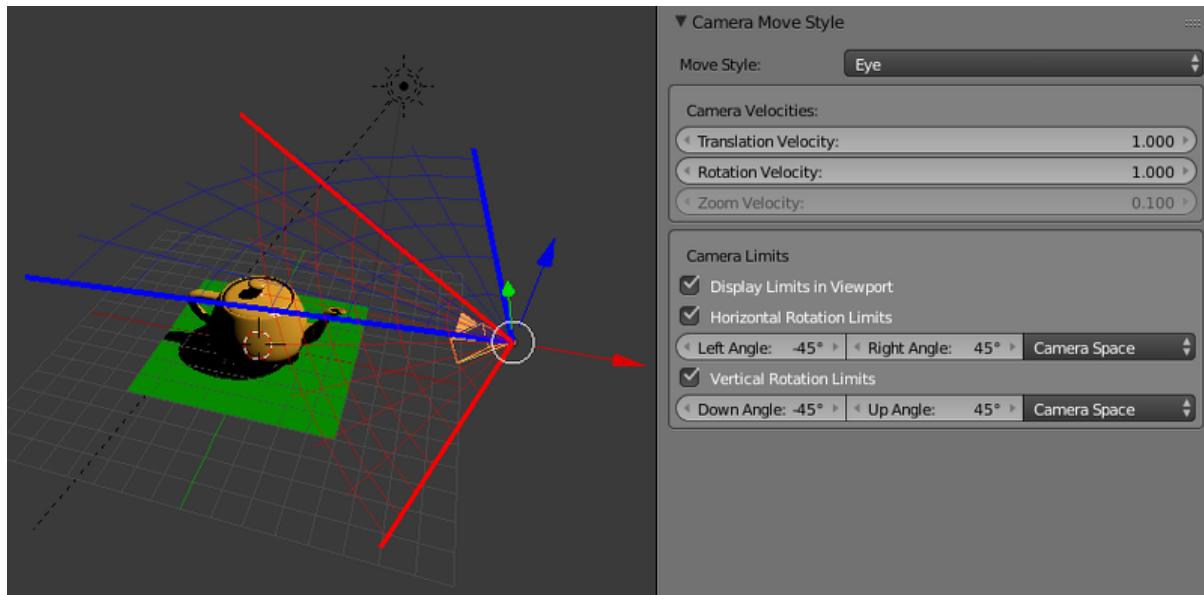
相机移动方式 > 相机限制 > 垂直旋转限制 限制摄像头水平（围绕在 Blender 的 Z 世界轴）对点的转动。旋转沿 左角 和 右角值之间的圆的弧进行。旋转弧对应于从 左角 到 右角逆时针运动。默认值：左角 = -45θ ，直角 = 45θ 。默认情况下禁用。



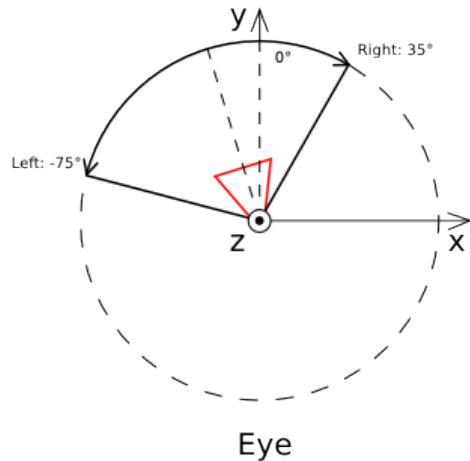
相机移动方式 > 相机限制 > 中心点移动限制 限制摄像机的中心点移动。默认值：最小 Z = 0, 最大 Z = 10. 在视口中不渲染。

相机移动方式 > 相机限制 > Use Panning Mode 使相机平移。

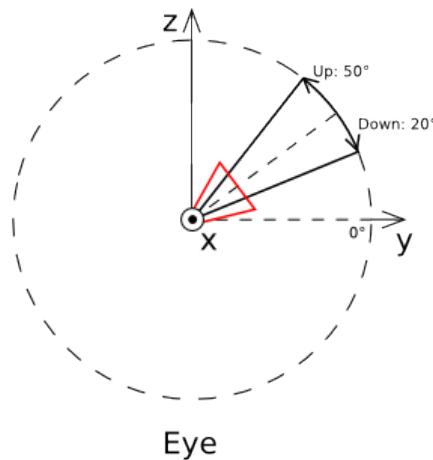
眼类型相机



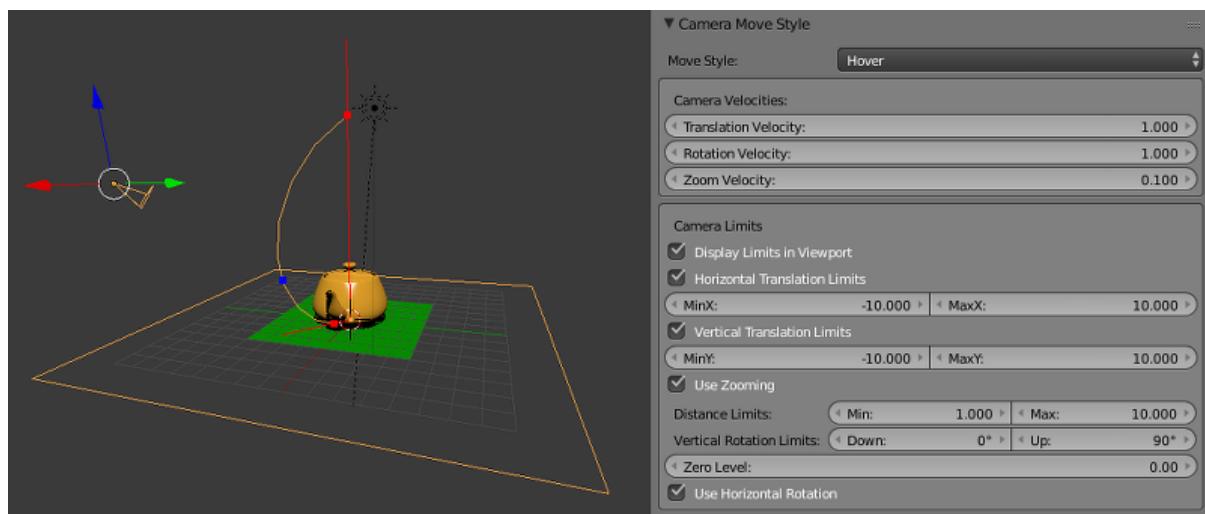
Blend4Web > Horizontal Rotation Limits 限制摄像头水平（围绕在 Blender 的 Z 世界轴）对点的转动。旋转沿 左角 和 右角值之间的圆的弧进行。旋转弧对应于从 左角 到 右角逆时针运动。默认值：左角 = -45° ，直角 = 45° 。默认情况下禁用。



相机移动方式 > 相机限制 > 垂直旋转限制 限制垂直（在 Blender 的局部 X 轴）摄像头围绕对应点转动。旋转沿 向下角 和 向上角值之间的圆的弧进行。旋转弧对应于逆时针运动从 向下角 到 最多角。默认值：下角 = -45° ，上角 = 45° 。默认情况下禁用。



悬停类型相机

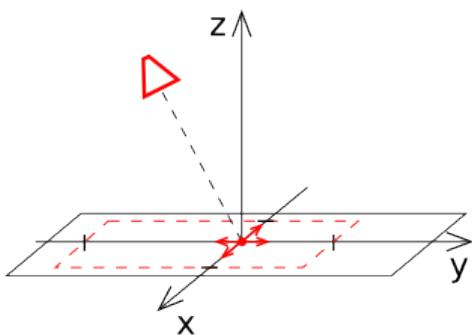


相机移动方式 > 相机限制 > 水平移动限制 沿着 Blender 的世界 X 轴支点的限制运动。

允许值: $\text{Min} \leq \text{Max}$. 默认值: $\text{MinX} = -10$, $\text{MaxX} = 10$. 默认情况下禁用。

相机移动方式 > 相机限制 > 垂直移动限制 沿着 Blender 的世界 X 轴支点的限制运动。

允许值: $\text{Min} \leq \text{Max}$. 默认值: $\text{MinY} = -10$, $\text{MaxY} = 10$. 默认情况下禁用。

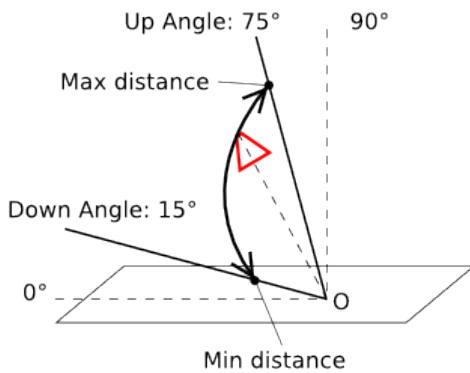


相机移动方式 > 相机限制 > Use Zooming 允许用户从中心点点调整摄像机的焦距和缩小。如果禁用此参数，相机将始终保持在离支点的距离固定。默认情况下禁用。

相机移动方式 > 相机限制 > 距离限制 设置最小和最大距离从相机到摄像机的视图方向与水平基准面（默认情况下 Blender 的世界 XOY 平面）之间的交点。允许值: $\text{Min} \leq \text{Max}$. 默认值: $\text{Min} = 1$, $\text{Max} = 10$. 默认情况下禁用。

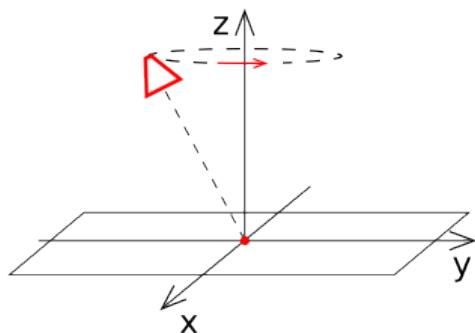
相机移动方式 > 相机限制 > 垂直旋转限制 限制抬高相机的角度（摄像机的视线与水平面之间的夹角可能的取值区间: $0^\circ \leq \text{Down} \leq \text{Up} \leq 90^\circ$. 默认值: $\text{Down} = 0^\circ$, $\text{Up} = 90^\circ$.

相机移动方式 > 相机限制 > Zero Level 基准面和坐标原点之间的距离。默认设置为零。如果 Blend4Web> 使用 Zooming 设置打开，同时设置距离和倾斜角度的限制，在垂直平面以限定在相机移动路径。

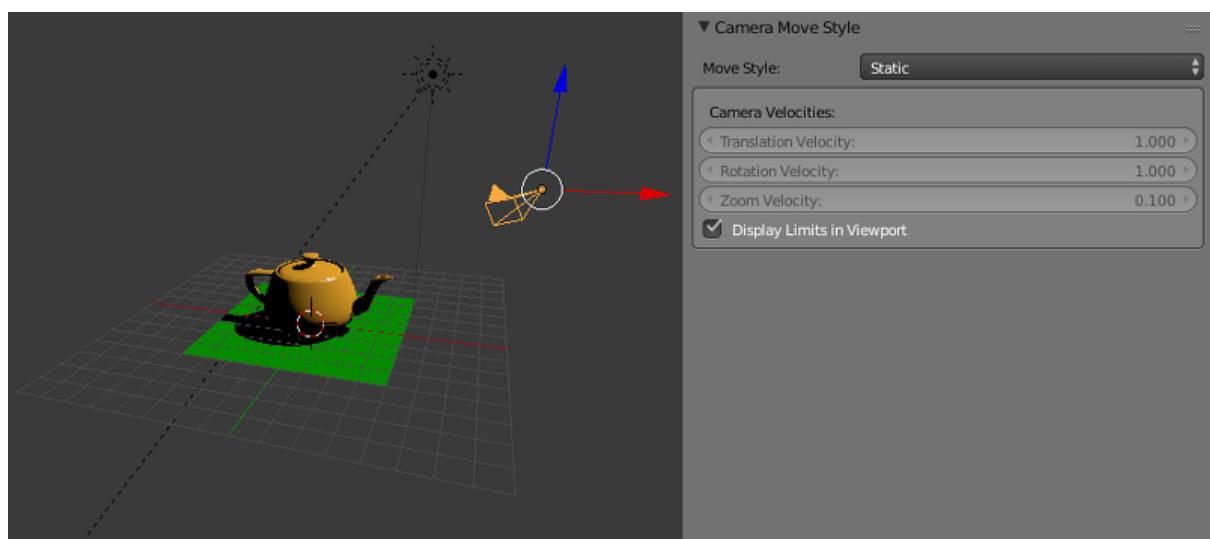


设置不正确的距离或仰角限制将禁用此参数。

相机移动方式 > 相机限制 > Use Horizontal Rotation 允许摄像头在 Blender 的 XOY 平面对着中心点旋转。默认情况下启用。



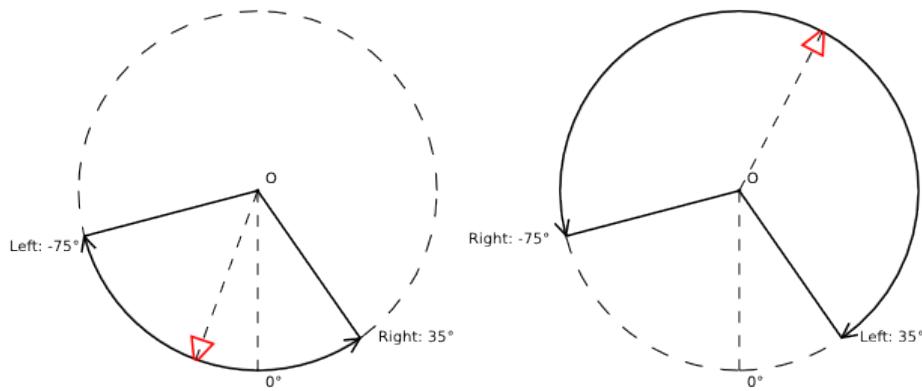
静态相机



这种类型的相机不具有限制，因为它不能直接由用户控制。

限制设置的特殊性

- EYE/TARGET 相机，交换 左/右或 下/上值导致移动圆弧相反。



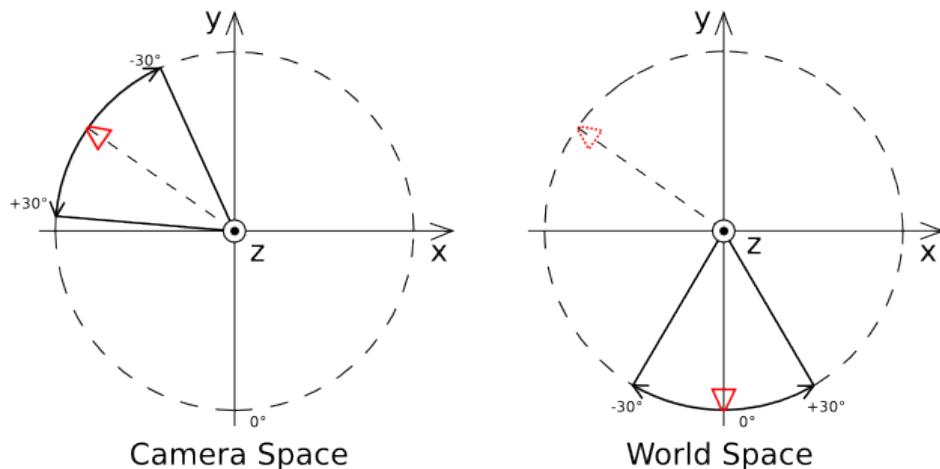
- 当限制相机的水平和垂直旋转，则可在下列坐标空间之间进行选择：

摄像机空间 所有的角度是相对于初始相机位置和取向测量。

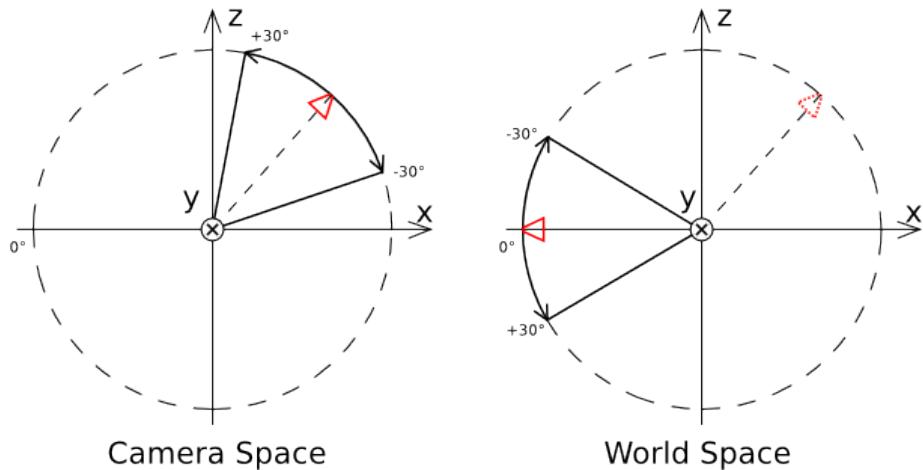
世界空间 水平角度是相对于世界空间 Y 轴测量；垂直角度是相对于世界空间中 Blender 的 XOY 平面上测量。

默认值：相机 Space。

水平限制目标相机的例子：

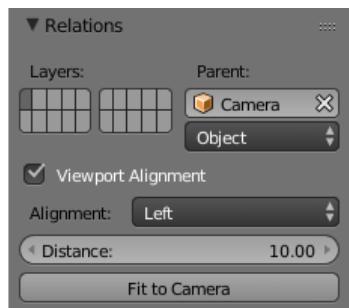


垂直限制目标相机的例子：

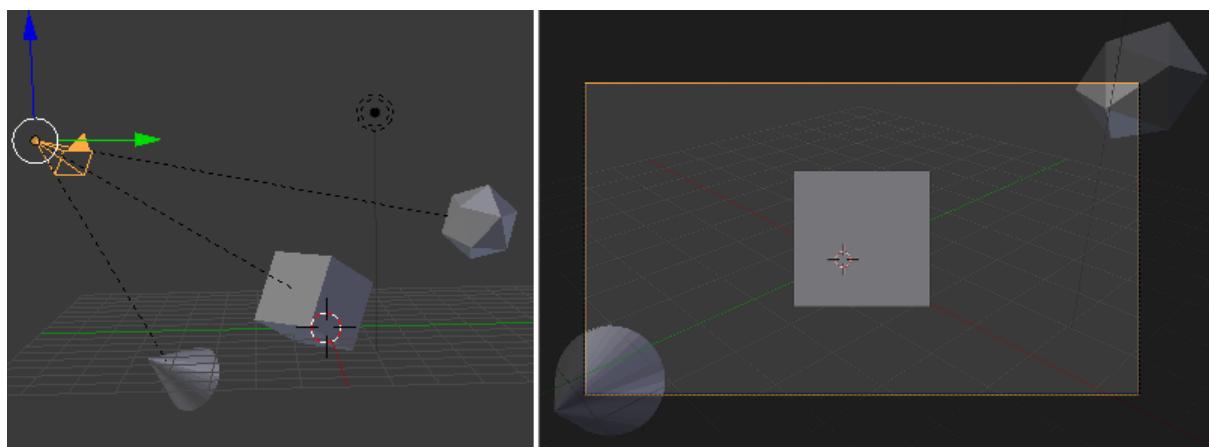


视口对齐

如果物体父级的相机的参数设置为 视口对齐，在 Object 面板下的 Relations 选项卡中。



使用这些设置，物体可以对齐到激活的摄像机，这样该物体将保持在屏幕同一个地方，无关摄影机的位置和转动。这是用于创建 UI 元素尤其有用。



这张照片显示可以如何使用视区对齐选项的示例。

视口对齐 此参数启用和禁用以下的所有参数。默认情况下禁用它。

对齐 此参数指定物体将与屏幕的哪一侧的对齐方式。它可以具有下列值之一。

- Top-Left - 将物体对齐到视口的顶部左上角。
- Top - 将物体对齐到视口的顶部。
- Top-Right - 将物体对齐到视口的顶部右上角。
- Left - 将物体对齐到视口的左边。
- Center - 将物体对齐到视口的中央。
- Right - 将物体对齐到视口的右侧。
- Bottom-Left - 将物体对齐到视口的底部左侧。
- Bottom - 将物体对齐到视口的底部。
- Bottom-Right - 将物体对齐到视口的底部右边。

默认值为 Center.

距离 此选项指定对齐的物体和相机之间的距离。默认设置为 10。

适合相机 按下此按钮，你可以使对齐的物体看起来在引擎中和 Blender 视口同样方式显示。它可以用于预览对齐结果。

相机控制 API

详细的 API 模块文档: [link](#).

照相机的位置和其空间取向是由控制模式来定义。在 EYE, TARGET 和 HOVER 模式，控制方式有若干限制：例如，相机的垂直轴总是沿着世界 Y 轴对齐，并且相机一直对焦到中心点。STATIC 模式相机有没有这样的限制，这使得它更适合当你需要更完整的控制的情况下，如程序动画。

相机控制主要功能都位于 `camera` 模块。其中一些（针对特定控制模式）的有哪些用适当的前缀开头的名字：`static_`, `eye_`, `target_` 和 `hover_`。其它功能可在任何模式下使用。

控制模式设置

若要更改控制模式，完全定义相机的行为，可以使用以下方法：`static_setup()`, `eye_setup()`, `target_setup()` 和 `hover_setup()`. 这些方法接收一个包含一组可选的参数的物体，可以用于设置相机的位置、旋转、可用的限制，等等。

```

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 });
```

HOVER 模式的主要特征在于，距离和仰角范围限制在一个确定的路径 来朝向中心点运动。相机会一直在这条路径上，所以其最终位置可能跟从 `hover_setup()` (基本的，它是投射到路径上) 方法返回的不一样。如果你需要设置相机的位置在一些路径，你可以使用 `hover_setup_rel()` 方法设置相对极限。

```
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 });
```

当前相机模式可以通过检查 `is_static_camera()`, `is_eye_camera()`, `is_target_camera()`, `is_hover_camera()` 或 `get_move_style()` 方法:

```
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!");

```

注解: 如果您需要更改相机的行为的某些方面, 你可以使用下面描述的方法。

平移和旋转

要旋转 STATIC 类型的相机, 您需要使用 `static_get_rotation()` 和 `static_set_rotation()` 方法。在这些方法中, 旋转通过一个四元数来定义。

```

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);

```

对于 EYE, TARGET 和 HOVER 类型摄像机, 旋转使用的是球面坐标 使用 `eye_rotate()`, `target_rotate()`, `hover_rotate()` 和 `rotate_camera()` 方法:

```

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);

```

摄像机的当前球面坐标可通过使用 `get_camera_angles()` 方法获得:

```

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];

```

获取摄像机的位置可以通过获得使用 `get_translation()` 和 `set_translation()` 方法。对于 TARGET 和 HOVER 模式，这意味着整个模型，包括摄像机的位置和中心点同时移动。

```

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);

```

其他相机移动方法：

- `target_set_trans_pivot()`, `target_set_pivot_translation()`,
`hover_set_pivot_translation()` - 同时设置相机的位置和中心点;
- `target_get_distance()`, `target_set_distance()` - 基于中心点移动位置;
- `static_set_look_at()`, `eye_set_look_at()` - 同时设置相机的位置和视点;

注解：因为相机是一个场景物体 `transform` 模块方法也可以使用。但是，由于每个控制模式以不同的方式影响相机运动，结果可能不同于那些预期。

限制设置

相机限制可供 EYE, TARGET 和 HOVER 类型相机。若要设置一个特定的限制，您需要使用特定的方法

眼睛	目标	悬停模式
eye_get_horizontal_limits()		hover_get_distance_limits()
eye_set_horizontal_limits()	get_distance_limits()	hover_set_distance_limits()
eye_get_vertical_limits()		hover_get_vertical_limits()
eye_set_vertical_limits()	get_set_distance_limits()	hover_set_vertical_limits()
	target_get_horizontal_limits()	hover_get_horiz_trans_limits()
	target_set_horizontal_limits()	hover_set_horiz_trans_limits()
	target_get_vertical_limits()	hover_get_vert_trans_limits()
	target_set_vertical_limits()	hover_set_vert_trans_limits()
	target_get_pivot_limits	
	target_set_pivot_limits	

```
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);
```

限制存在检测可以通过 `has_distance_limits()`, `has_horizontal_rot_limits()`,
`has_vertical_rot_limits()`, `has_horizontal_trans_limits()` and
`has_vertical_trans_limits()` 方法获得。

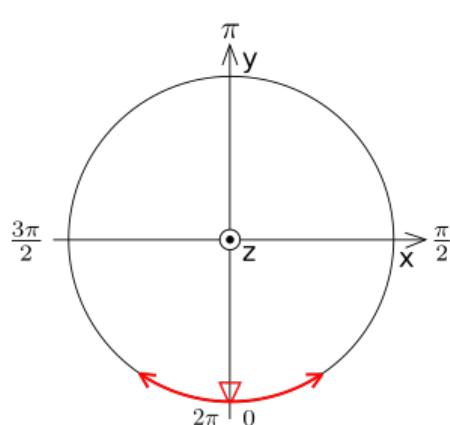
注解: 在 HOVER 模式下，相机总是有距离和仰角的限制。如果这些限制都没有设置，它们会自动计算锁定摄像头当前的位置相对中心点。

注解: 已启用的相机限制可以影响它的位置和空间取向, 由 API 函数设置。

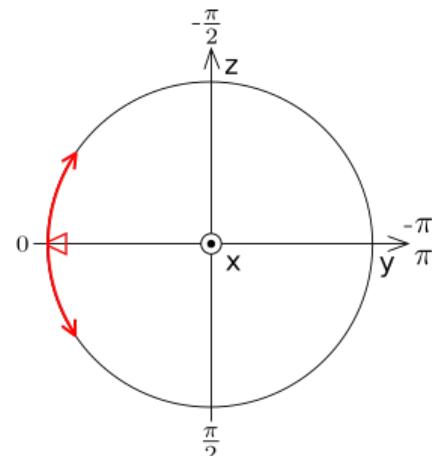
角坐标计数

当用相机的 API 的工作 (旋转, 设置限制), 各个角度都定义如下:

对于 TARGET / HOVER 相机:

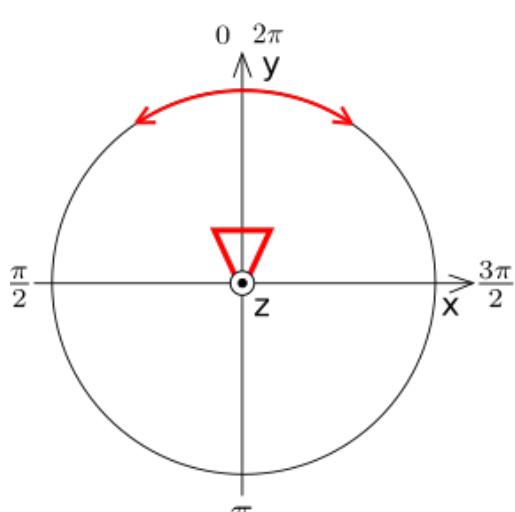


Azimuth angle

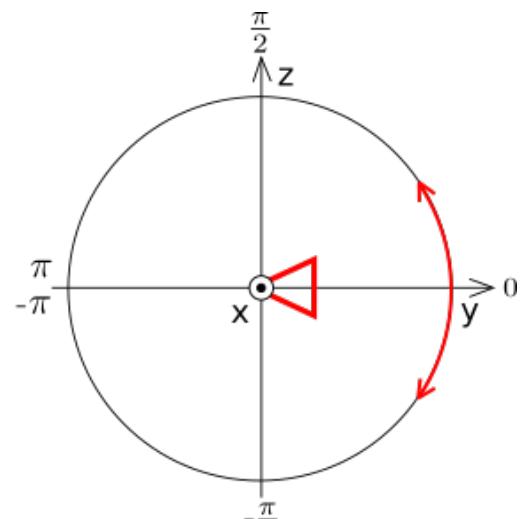


Elevation angle

对于 EYE 相机:



Azimuth angle



Elevation angle

材质

目录

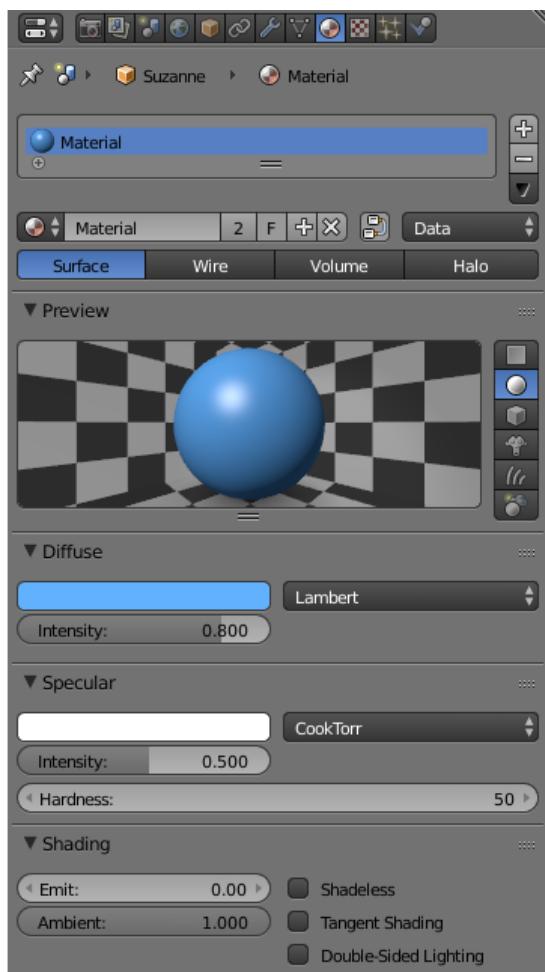
- 材质
 - 灯光参数
 - 透明
 - * 类型
 - * 附加设置
 - 反射
 - * 静态反射
 - * 动态反射
 - * 菲涅耳反射效果
 - 渲染属性
 - 视图属性
 - 引擎特定参数
 - 光晕材质
 - * 激活
 - * 附加设置
 - 材质 API
 - * 堆叠材质的方法
 - * 节点材质的方法
 - 材质继承
 - * 激活

材质是描述物体表面对光的反应，还包含有关其透明度、反射率、物理参数等信息。

网格可以有一个或多个材质。在多个材质的情况下，他们可以分配给不同多边形在编辑模式。要做这个选择所需的多边形，从列表中选择所需的材质，然后单击分配按钮。

支持以下材质类型: Surface , 光晕.

灯光参数



漫射 > 颜色 漫反射光的颜色。默认值为 0.8, 0.8 (0.8)。它可能与漫射映射贴图颜色进行交互。

漫射 > 强度 漫反射光的强度。默认值为 0.8。

漫射 > 着色 漫反射着色算法。默认值为 兰伯特.

反射 > 颜色 反射光的颜色。默认值是 1.0, 1.0 (1.0)。它可能与高光贴图颜色进行交互。

反射 > 强度 镜面反射光的强度。默认值是 0.5。

反射 > 硬度 反射着色计算公式中的指数。默认值为 50。请注意，在引擎中使用的公式和 Blender 中的略有不同。

反射 > 着色 高光着色算法。默认值为 CookTorr.

着色 > Emit 发光强度。默认值为 0.0。

着色 > 环境 影响材质的环境因素。默认值是 1.0。

着色 > 无光影 当启用时，材质不会对光起反应。默认为禁用。

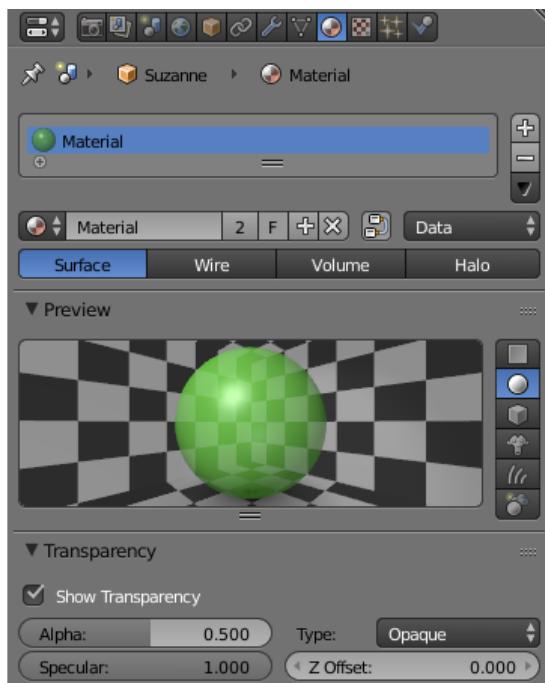
着色 > 切向着色 当启用此参数时，引擎将使用材质的切向量（而不是普通的向量）计算物体的颜色。这可以用于创建各向异性阴影效果。



在左边： 标准着色模型；右： 切向着色模型。

着色 > 双面光照 启用双面照明模式。此选项是适用于非透明物体的单层网格。

透明



类型

属性 > 材质面板菜单中的 透明，可以选择透明度的类型。

引擎支持以下的透明度实现类型（按性能升序排序）[F](#)

Alpha 排序 带渐变透明。引擎按三角与相机间的距离进行排序来正确渲染重叠的透明表面。此操作计算代价昂贵。建议将此功能用于封闭透明几何（瓶、汽车玻璃等）。

Alpha Anti-Aliasing Transparent with a gradient. This feature is implemented with the help of the Alpha to coverage technique. It's available only if MSAA is enabled (only on WebGL2-capable devices). The sorting of triangles is not performed. It is recommended to use this feature with a mask texture to visualize smaller details (tree leaves, grass).

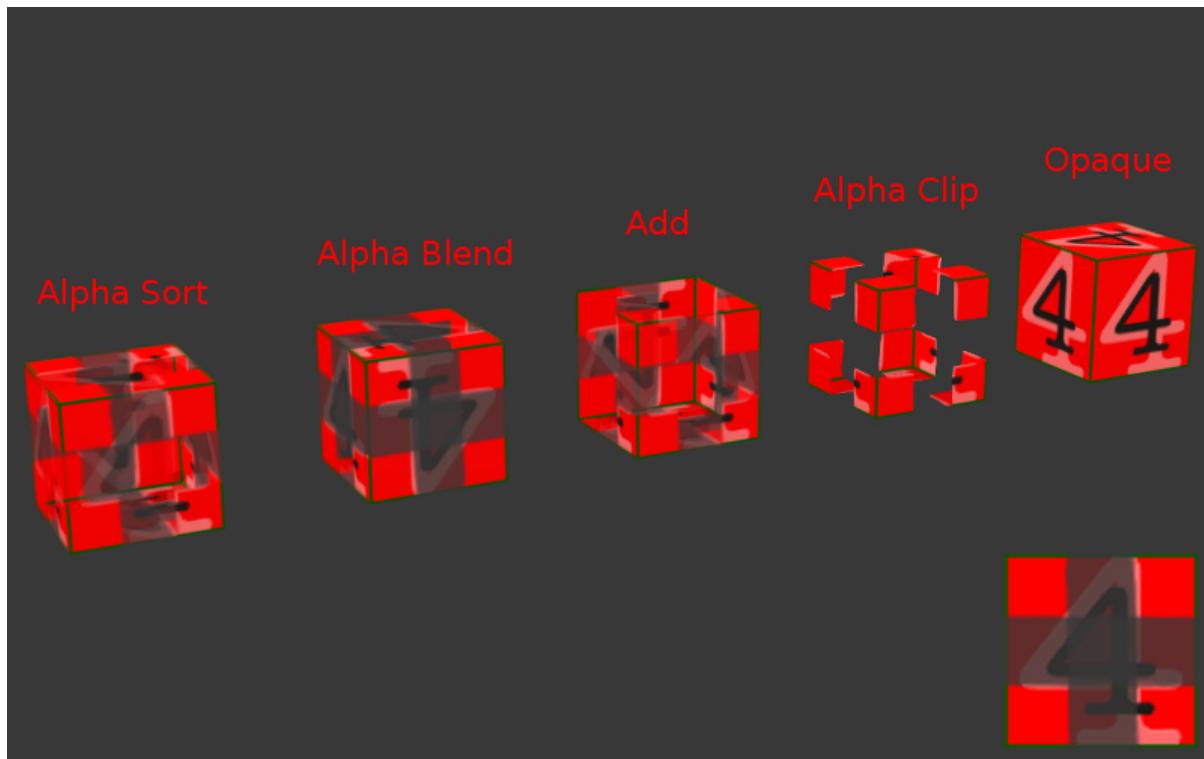
Alpha 混合 带渐变透明。不执行三角形排序。建议在无封闭透明几何体（水表面，贴花）使用此功能。

添加 带渐变透明。不执行三角形排序。引擎将禁用导致透明曲面以任意顺序渲染的深度缓冲区写入。建议在效果中（粒子系统，辉光）中使用此功能。

Alpha 剪切 没有渐变透明。如果 alpha 小于 0.5，引擎丢弃像素。不执行三角形排序。

建议使用此功能与遮罩纹理一起用来渲染小细节（树的树叶、草）。

不透明 非透明。Alpha 将被忽略。这是默认值。



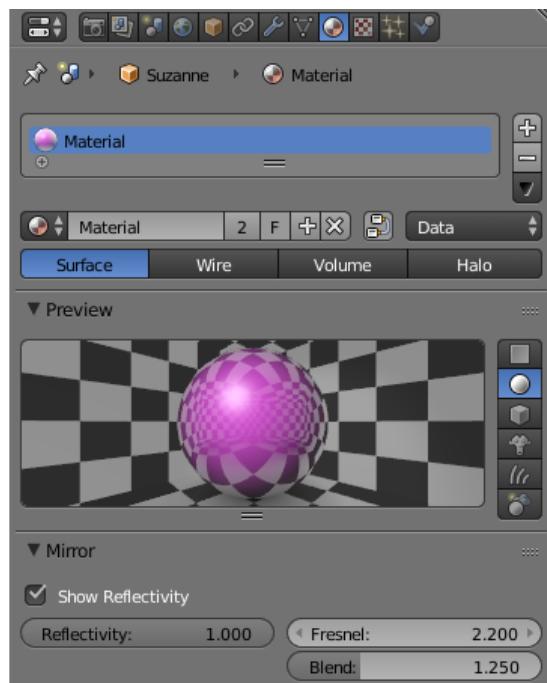
附加设置

透明度 > 显示透明度 启用透明度复选框是可以在 Blender 中查看透明物体。引擎忽略此选项- 引擎要用 Alpha Blend 选项。

透明度 > 阿尔法 材质的透明度级别。引擎忽略此参数（与 Blender 相反），如果有漫射纹理 - 改为使用纹理的 alpha 通道值。

透明度 > 阿尔法 此选项明确指定带有深度排序相关的 **不同材质** 物体的相对定位顺序。该选项可以有正负值。越遥远的物体用较小的参数值更能被正确的渲染。默认值为 0.0。

反射



静态反射

无论环境如何变化，表面任然渲染同样的一张图。只需使用[mirror map](#)便可激活。

参见:

[菲涅耳反射效果](#)

动态反射

表面反射了他们当前的位置中的选定的物体。引擎支持平面和球面反射。

注解: 如果您使用的是[节点材质](#)，如果 Material 或 Extended Material 节点出现在节点树，动态反射才有效。

激活

1. 检查 Render > Reflections 和折射面板上的 Reflections 设置。

2. 对反射物体，启用 Object > Reflections 面板上的 Reflective 选项。
 - 对平面的反射，Object > Reflections > Type 属性设置为 Plane。然后，添加一个空物体用作反射平面，例如通过执行 Add > Empty > Single Arrow。为方便起见重命名，在反射物体 Reflection plane 字段中指定其名称。
 - 用方块贴图反射，Object > Reflections > Type 属性设置为 Cube.
3. 反射物体如需要材质，设置 材质 > Mirror > Reflectivity 值。
 - 在 Blender 中显示的物体上的反射需要设置 Mirror > Reflectivity > Show Reflectivity 。引擎将忽略此选项。
4. 对反射性物体，启用 Object > Reflections 面板上的 Reflexible 复选框。

注解: 建议启用 World > Environment Lighting 复选框。

限制条件

因为优化性能的原因法线贴图和阴影在反射图像中将被忽略。

参见:

[菲涅耳反射效果](#)

菲涅耳反射效果

菲涅耳效应表现为依赖和反射光的入射角来表现反射的强弱。如果入射角是接近零（即光入射几乎表面成直角）光传递部分较大然而反射光部分小。相反如果入射角是接近 90 度（即光入射几乎平行于表面）几乎所有的光反射。

本引擎采用近似 Schlick 公式

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

R - 反射系数

R_0 - 在成直角的表面中观看的右夹角下的反射系数（即当 $\theta = 0$ ），

θ - 入射角（即等于反射角度下的光进入相机），它由引擎实时计算，

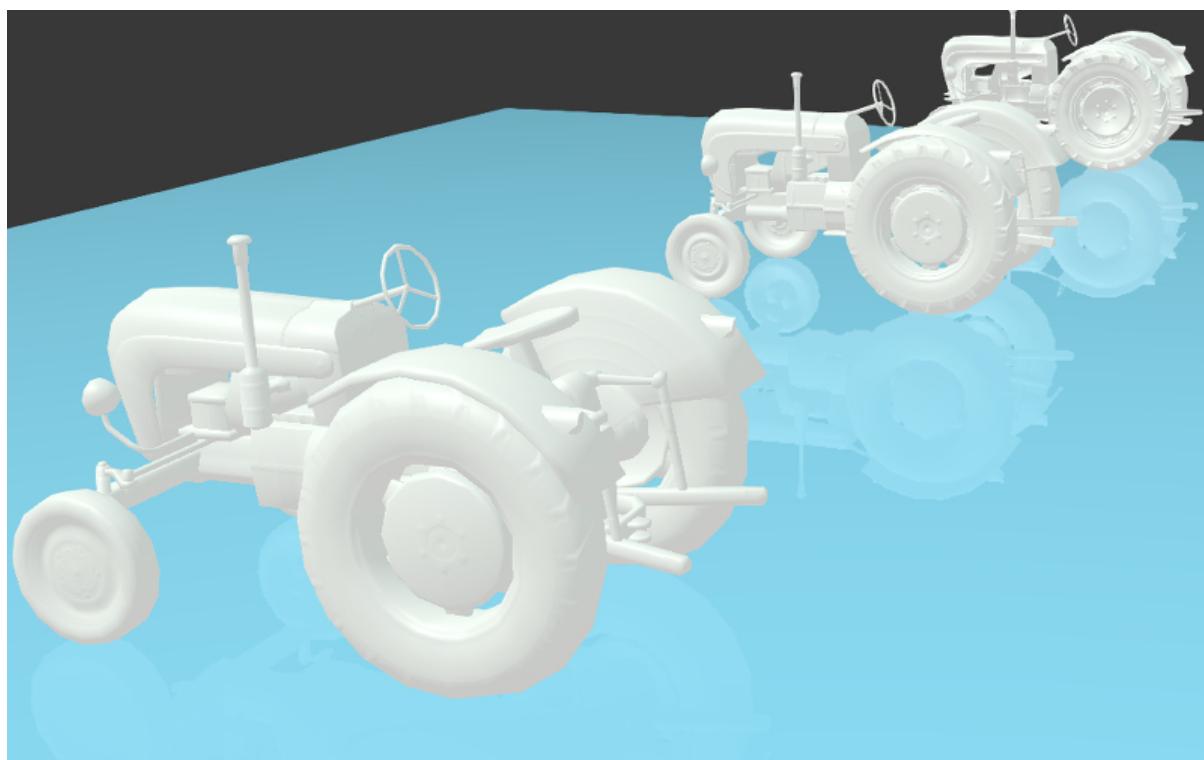
N-指数。

设置

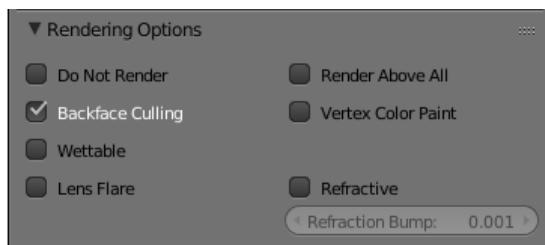
菲涅耳效应可以设置静态和动态的反射。

材质 > 镜面 > 菲尼尔 反射的菲涅尔指数。Schlick 方程中的 N 指数。在 blender 中被限制在 0 到 5 的值。设置为零的时候菲涅耳效应无法被观察到，所有的反射都是正常的。大于零的时候，在角度这是接近正确的角度观看时表面材料是弱反射性的。参数值越大，可以看到菲涅尔效应的观察角度越大。

材质 > 镜面 > 混合 菲涅耳反射系数。在 Schlick 公式中通过下面的表达式 $R_0 = 1 - Blend / 5$ 它减少到 R_0 。在 Blender 中它仅限于值从 0 到 5。此参数定义了菲涅耳效应强度: Blend 因子越大，更多的菲涅耳效应影响。如果它是等于零的菲涅耳影响观察不到。



渲染属性



材质 > 渲染选项 > 不要渲染 禁用渲染此物体。

材质 > 渲染选项 > 背面剔除 当启用时，引擎不会渲染多边形背面。默认情况下启用。

材质 > 渲染选项 > 湿润 为材质激活水润湿效果。

参见:

[水](#)

材料 > 渲染选项 > 镜头光晕 启用此参数可激活材质镜头光晕效果。

材质 > 渲染选项 > 渲染以上全部 材质被渲染在场景的所有物体的顶层。要求使用渐变透明度类型 (Add, Alpha Blend[“] 或者 Alpha Sort).

材质 > 渲染选项 > 顶点色绘制 复选框启用时，使用网格顶点颜色而不是物质的漫反射颜色。

材质 > 渲染选项 > 折射性 使物体折射。摄动因素可以设置 Refraction Settings 面板上的 Refraction Bump 选项设置。默认值为 0.001。

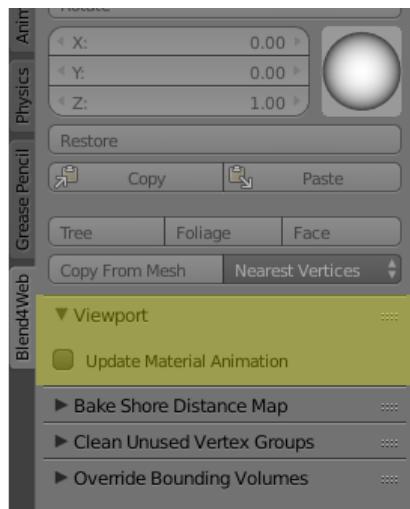
注解: 为了使用这种效果，Render > Reflections and Refractions > Refractions 面板上选择 ON 或者 AUTO 。该物体必须具有 Alpha Blend 透明类型。

参见:

[透明](#)

视图属性

Viewport 部分可以在 Blend4Web 面板上发现。



更新材质动画

在 Blender 视口更新动画着色。

引擎特定参数



材质 > 水 特殊材质水渲染.

材质 > 地面动态草 材质用于草渲染.

材质 > 碰撞 碰撞几何体的一种特殊材质。

参见:

[物理](#)

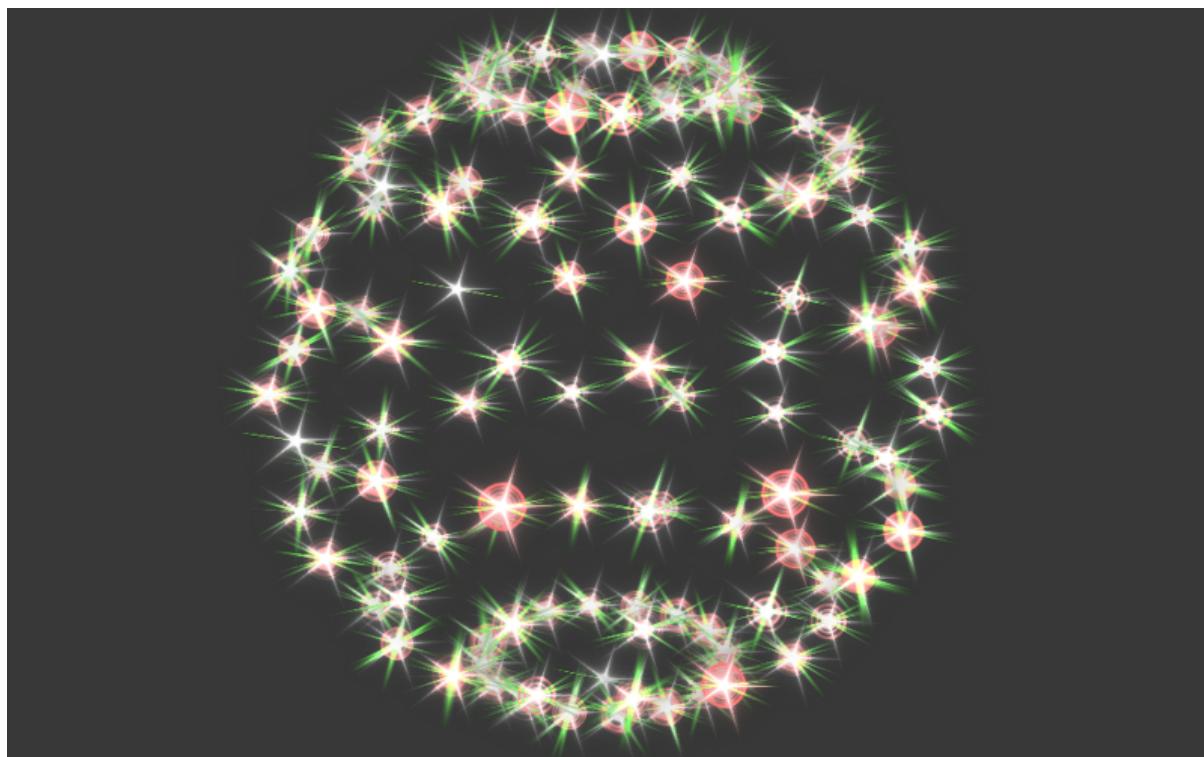
材质 > 导出选项 > 不要导出 材质不导出。

光晕材质

光晕材质在粒子系统和静态网格中使用。使用静态网格中的光晕是如下所述。

激活

在 Materials 选项卡下选择 光晕类型。建议也选择使用渐变的透明度类型 (Add, Alpha Blend 或者 Alpha Sort).



附加设置

光晕 > Alpha 材质透明度的参数。默认值是 1.0 (不透明)。

光晕 > 颜色 材质的颜色。默认值是 (0.8, 0.8, 0.8) (基本是白色)。

光晕 > 大小 粒子大小。默认值是 0.5。

光晕 > 硬度 计算渐变的指数。影响粒子的可见尺寸。默认值为 50。

光晕 > 环 使用环。可以设置相对数量和颜色。

光晕 > 线 使用线。可以设置相对数量和颜色。

光晕 > 星点 使用星星。可以设置边的数量。

光晕 > 特殊: 星 启用星空渲染模式。网格被固定的相对于相机。还需启用 Sun 的 Lamp > Dynamic Intensity 复选框。应用程序应通过 API 设置黑暗的时间长度。

光晕 > 混合高度 褪色的星星的高度范围。

光晕 > 最小高度 物体的局部坐标系中使星星可见的最小高度。

材质 API

用于设置和场景的材质，包括堆叠和节点，的所有 API 方法，位于材质 API 模块中。该模块所提供的每个方法的完整说明都在 [page API 参考页](#)。

注解: API 方法只能在添加到动态物体的材质上工作。

堆叠材质的方法

`material` API 模块包括控制堆栈材质的每一个方面的方法。

这里有材质 API 模块如何用来对一个物体的材质进行各种操作的几个示例：

获取物体的漫反射颜色：

```
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");
```

获取物体的高光颜色：

```
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");
```

获取其他参数：

```
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");
```

设置堆叠材质参数：

```
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});

```

节点材质的方法

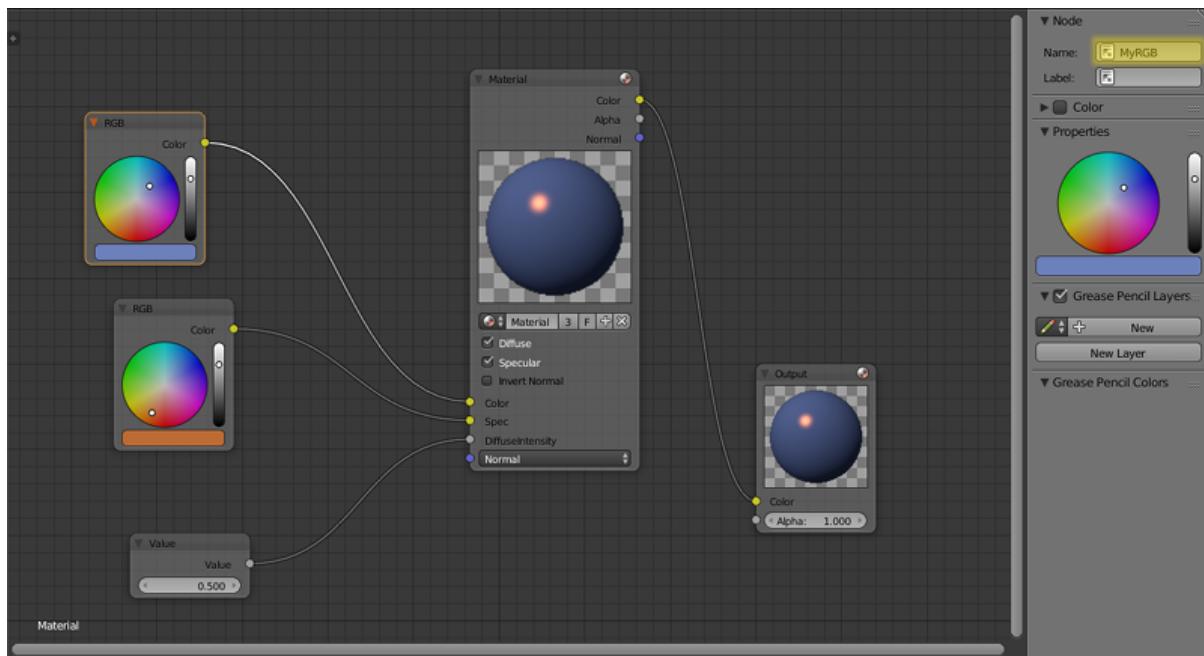
此时，API 方法只能影响 RGB 节点和数值值节点。不支持任何其他类型。

要改变一个特定的材质节点的值，您需要：

- 到物体的链接，使用这种特殊的材质，
- 该材质的名称，
- 节点本身的名字
- 和包含该节点（如果存在的话）的节点组的名称。

最好也给节点唯一的名称，否则你将无法获得它们的链接。一个节点的名称可以在在节点编辑器窗口的右侧侧面板的上部查看和改变。

注解： 在 Node 编辑器的主要窗口中的节点不显示节点的名称，只有它的类型。



这两个例子说明了一个节点材质的参数如何通过 API 方法来调节。

获取节点材质参数[F]

```
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"]);
```

设置节点材质参数:

```
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);
```

材质继承

Blend4Web 支持动态的材质切换。这个特性给用户一个机会可以配置模型外观非常对有用，诸如服装，一件家具，汽车或其他东西。继承材质特性可以在通过 `change_texture` 方法来调整堆叠材质和节点材质不够用的时候使用。

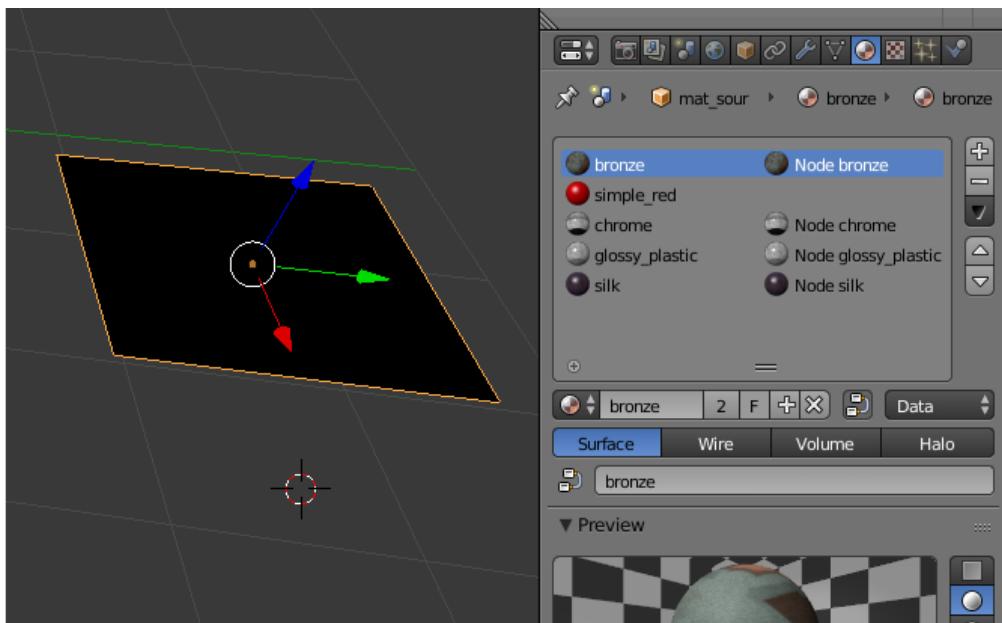
继承材质的优点是灵活，易于使用，这使得当你必须显着改变源材质的时候，它是一个更好的选择。它同时适用于 API 的脚本和逻辑节点，并且它可用于堆叠材质和节点材质。这个功能的缺点是，它可能无法像上面描述的其他方法一样快速工作，这在某些情况下可能是至关重要的。

激活

要使用继承材质，你需要两个物体。其中第一个是，具有将被继承材质的源物体。第二个是其具有需要更换材质的目标物体。这两个物体应该具有启用了 Dynamic Geometry & Material 选项（此选项可以在 Object 标签下可以找到，在 Rendering Properties 面板）。

注解: 当一个材质被继承后，它看起来和在 Blender 中被简单地分配给目标物体的完全一样。这意味着，需要事先将继承的每一个材质的所有 UV 贴图和顶点颜色层进行手动准备。对于源物体，这是没有必要的。

如果使用的是逻辑节点，继承材质可以用 Object->Inherit Material 节点执行。如果你正在使用 API，可以使用 `inherit_material` 方法完成。



如果应用程序需要许多不同的材质被改变，创建一个专用的容器物体可能是一个不错的办法。它可以是一个简单的平面网格，就像一个 Plane 物体，有你可能需要分配给它的每一个材质。这个物体可以隐藏在场景中，通过在 Object->Rendering Properties 面板中设置 Hidden 或者 Do Not Render 的选项。

材质库

目录

- 材质库
 - 库结构
 - 材质结构
 - * 核心材质
 - * 使用材质的一个例子
 - 添加材质到一个场景
 - * 如何添加整个材质
 - * 如何只添加核心材质

材质库是用户可以在其中找到基本和最常用的材质，在他或她的项目中使用或创建他或她自己的基于这些材质的材质库。

当创建一个新的项目，你可以插入材质库进去，然后直接从那里插入材质。



To do this, check the Use Material Library box when creating the project using the Project Manager. Then, an entire material library will be automatically copied to the /assets/material_library/ folder in the new project's directory.

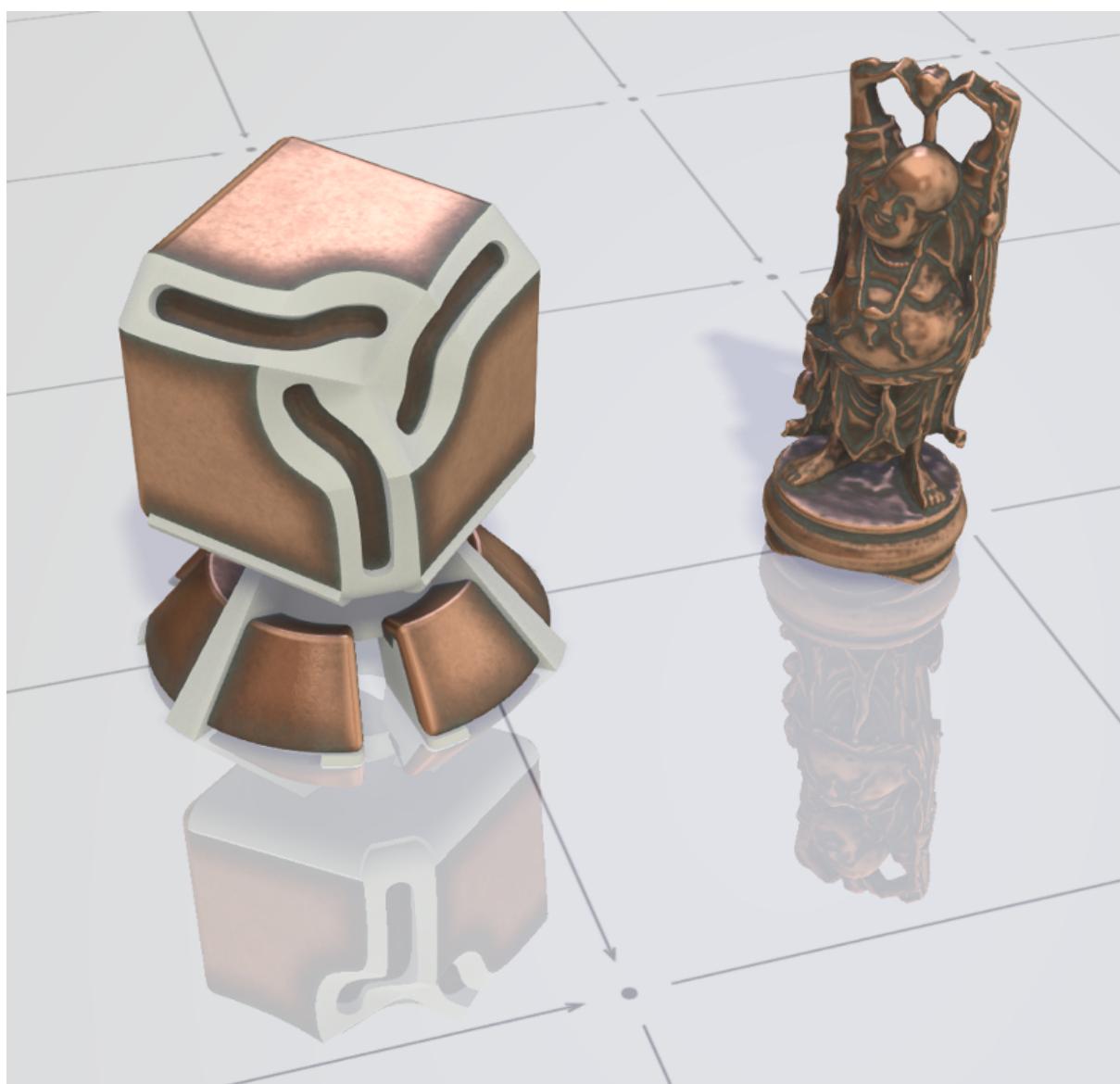
注解: It is also possible to use Material Library in a pre-existing project. All you have

to do is copy files from the `/blender/material_library/` folder to the `/assets/` folder in your project's directory. Then you can link the files that contain materials you need to your scene file and use them in the scene.

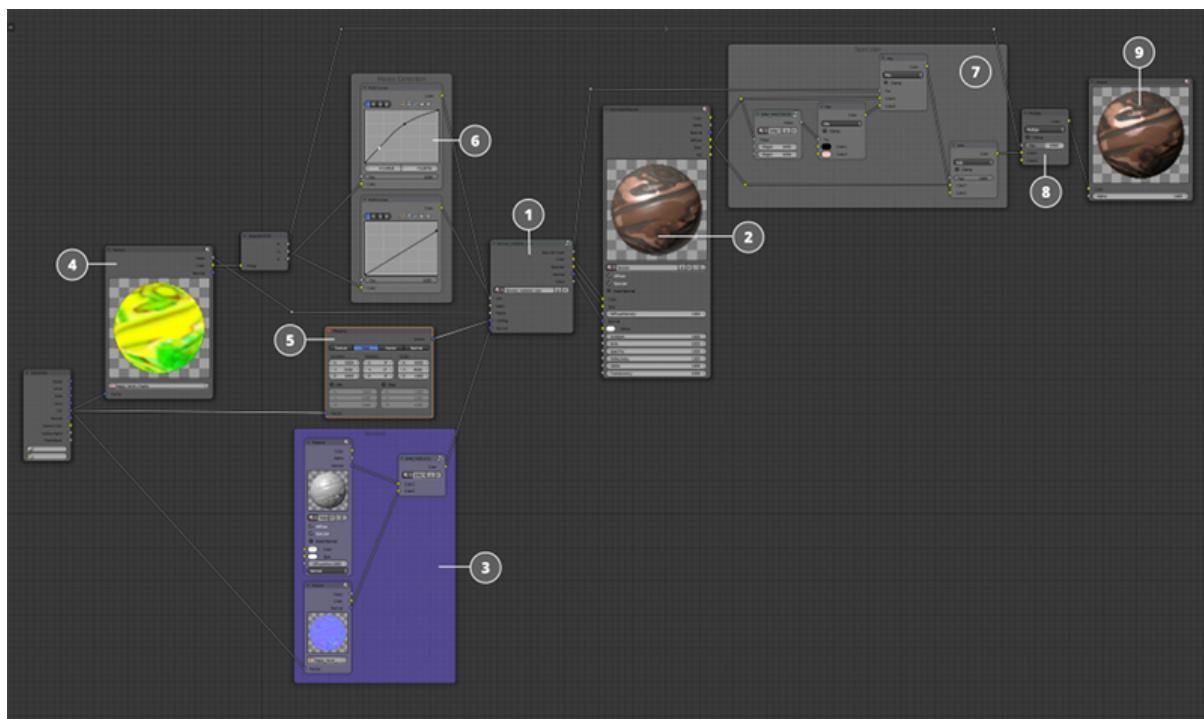
注解: Materials (or any other objects) can be both linked or appended to the scene. The difference is described [here](#).

库结构

所有材质通过文件夹组织结构（由材质类别），并通过一个单独的文件（由材质）。每共 blend 文件包含一个基底材质和物体添加该材质，演示该材质可以与该材质相关联的物体上（例如，元宝为金材质）。



材质结构



1. 核心材质
2. Material 节点从核心接收任何有价值信息
3. 法线贴图可以加到任何一个材质核心
4. 纹理包含额外的遮罩信息给材质给予物体正确显示。大多数材质使用下面的遮罩：
 - 环境闭塞贴图
 - 污垢贴图
 - 磨损图
 - 绿锈贴图
 这些遮罩不是在每个材质中使用，需要时，才为着色器显示一种特定的方式。
5. 在 Mapping 节点用来控制 UV 映射的参数。
6. RGB Curve 节点可以在遮罩插入核心材质之前修正。
7. 具体的高光贴图在 Material 节点之后，用来实现不同程度的镜面模糊。

8. 烘焙的环境闭塞贴图是在输出节点之前加的着色器。

9. Output 节点。

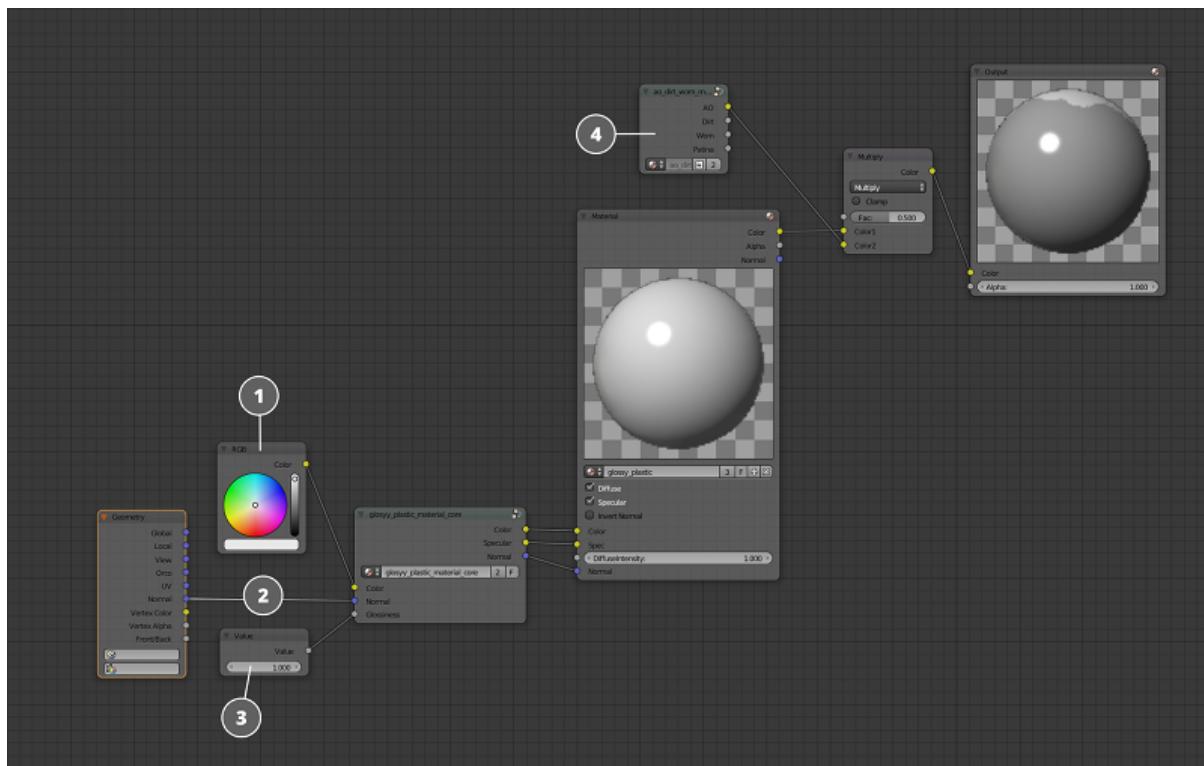
核心材质

这是包含节点结构形成一个着色器的主要节点组。本质上，这是材质本身的真实形式，与所有复杂节点交互包装成一组，方便使用。此节点组具有一定的输入和输出，以及具体的输入参数，来进行设置。

使用材质的一个例子

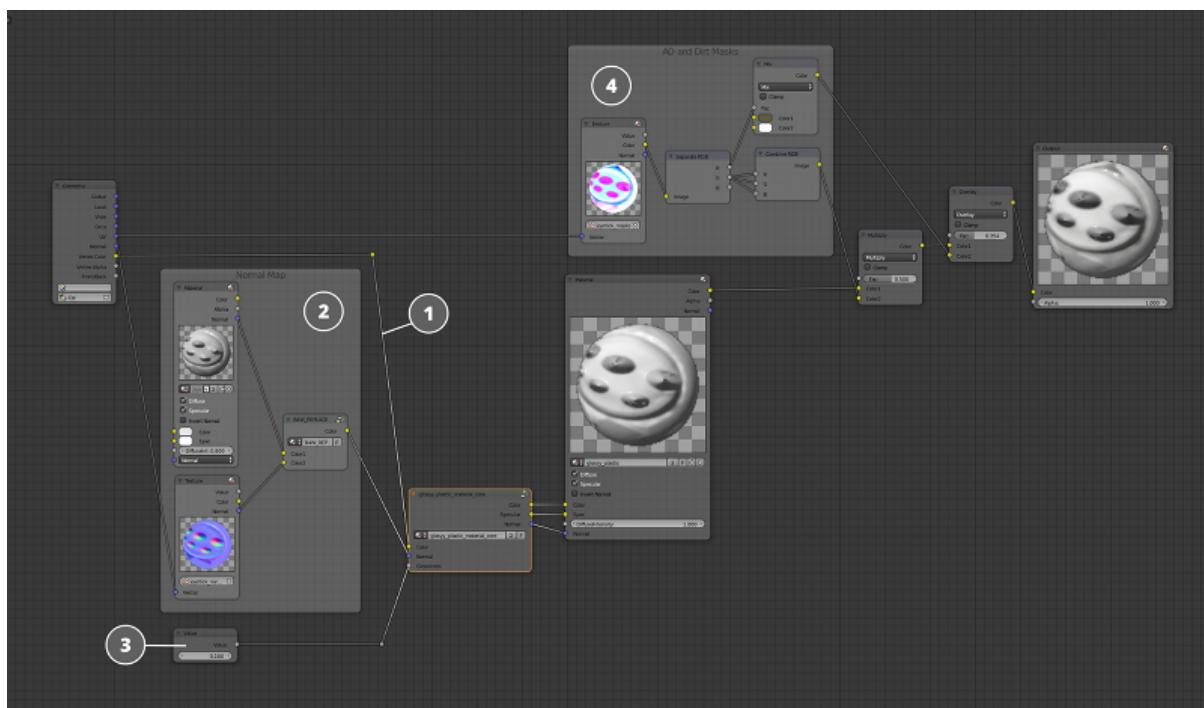


在这里，可以看到，设置塑料材质为一种特殊的演示物体的一个例子，而无需使用被插入核心材质中的纹理。



1. 基色用一个简单的“RGB”颜色节点创建并插入到核心材质输入。
2. 材质实现不使用任何法线贴图，所以几何数据插入到 Normal 输入。
3. 该核心材质具有控制光泽的材质上的量的某些参数。在这种情况下，这个参数的值设置为最大。
4. 该材质还采用了烘焙的环境闭塞贴图。

现在，让我们来看看使用相同材质在不同物体的情况 - 一个塑料手柄。

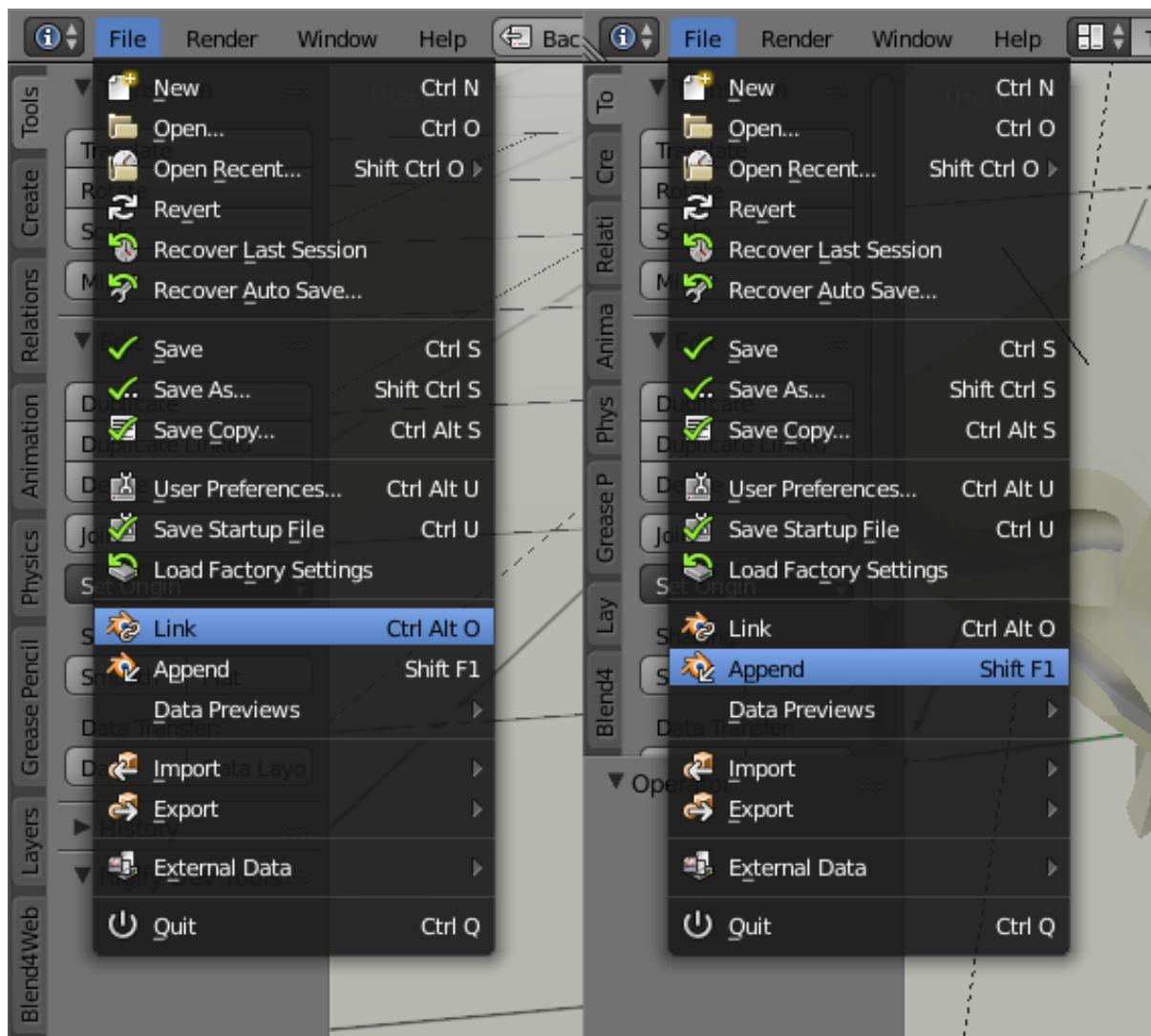


1. 物体的几何形状的顶点的颜色被用作这里的基本颜色。
2. 手柄的几何形状烘焙到了一张法线贴图上，并连接到核心材质的 Normal 输入。
3. 光泽的程度较低。
4. 替代演示物体的遮罩，专为游戏手柄模型烘焙的遮罩（环境光遮蔽和污垢）被使用。

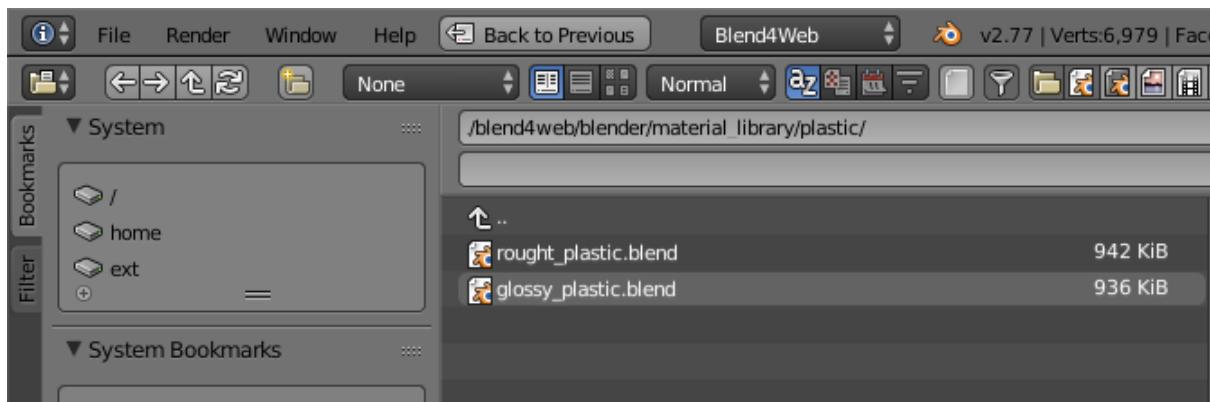
注解：应当指出的是，核心材质并没有改变以适应不同的几何形状。

添加材质到一个场景

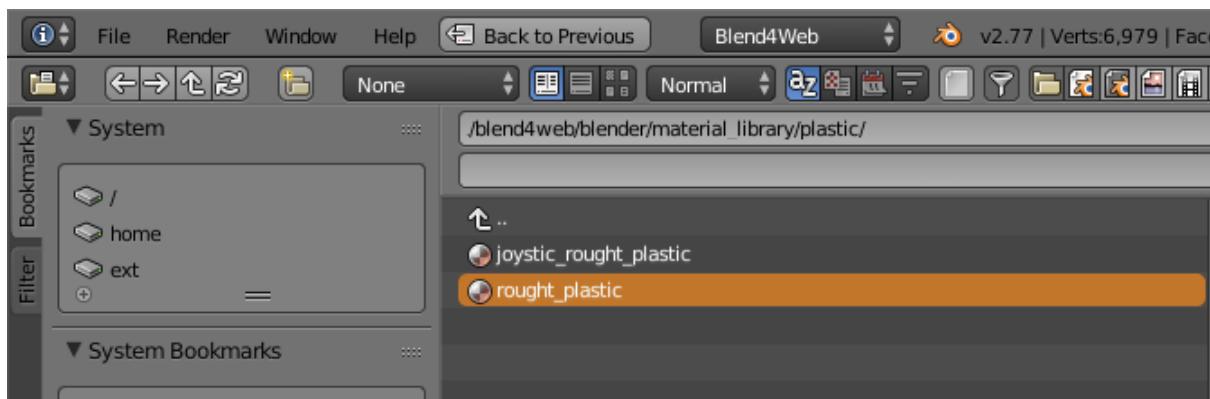
如何添加整个材质



选择 File > Append/Link 从菜单中选择，这取决于你想要做什么。如果你想保留修改物体的能力，它已被添加到场景后，使用 Append 。如果你不这样做，用 Link 。



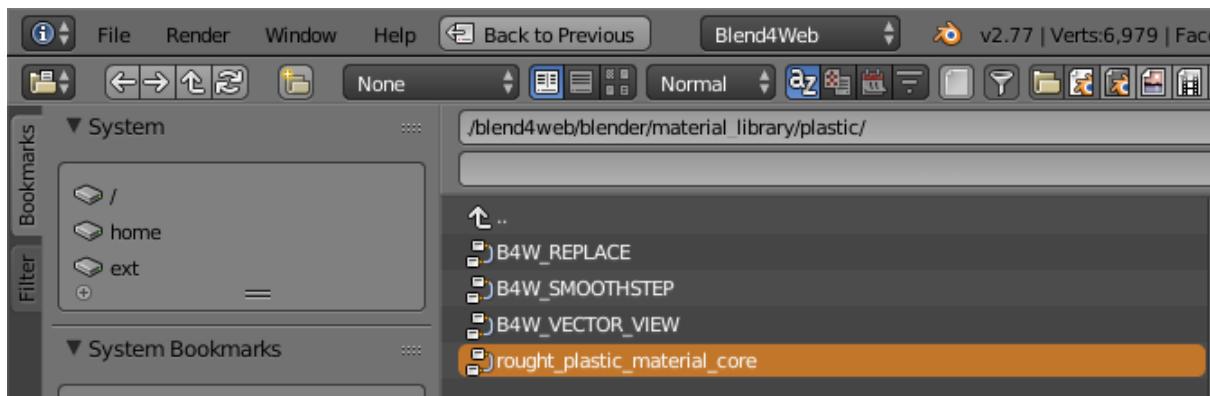
然后从 `/blend4web/blender/material_library/` 目录中选择包含所需要的材质 blend 文件。



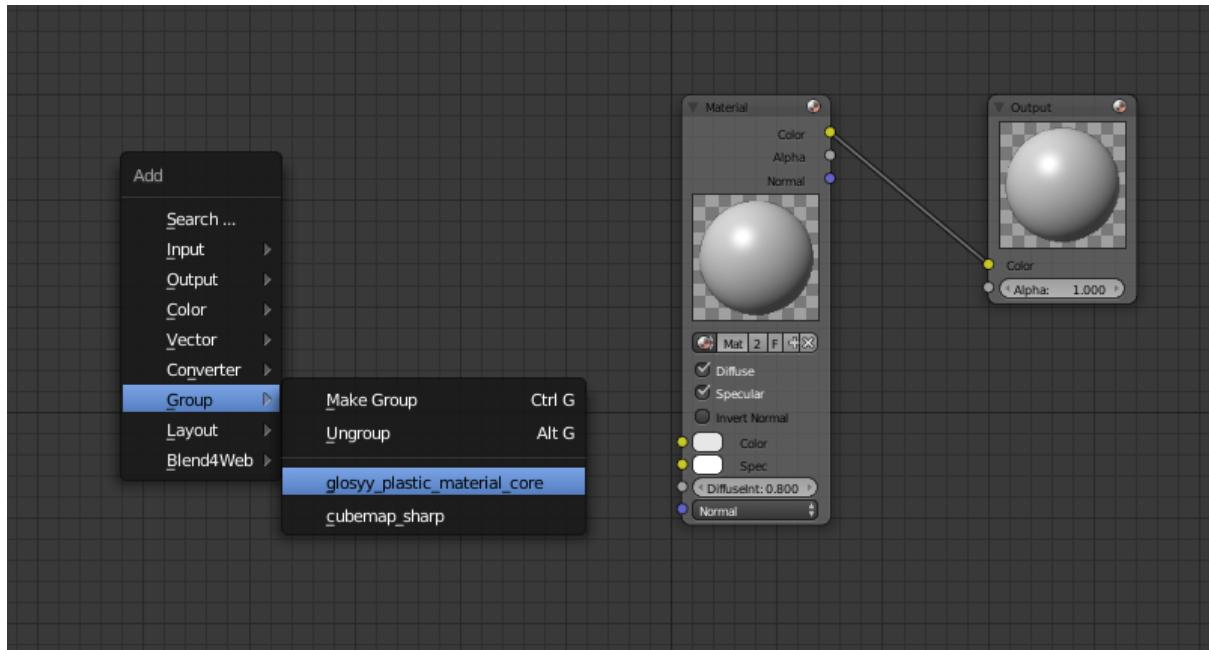
从材质列表中选择需要的材质，然后按 Append from Library 或 Link from Library. 追加。现在，该材质已经被添加到场景，并可以在任何物体上使用。

如何只添加核心材质

如果你想只添加核心材质，其他一切都自己设置的话，不用选择材质，选择 NodeTree 同时连接/追加，然后选择从库中您需要的节点组，然后按 Append from Library 或者 Link from Library 。



现在，这个节点组可在节点组列表中找到，您可以通过点击 Add=>Group 在节点编辑器界面添加材质列表中找到。



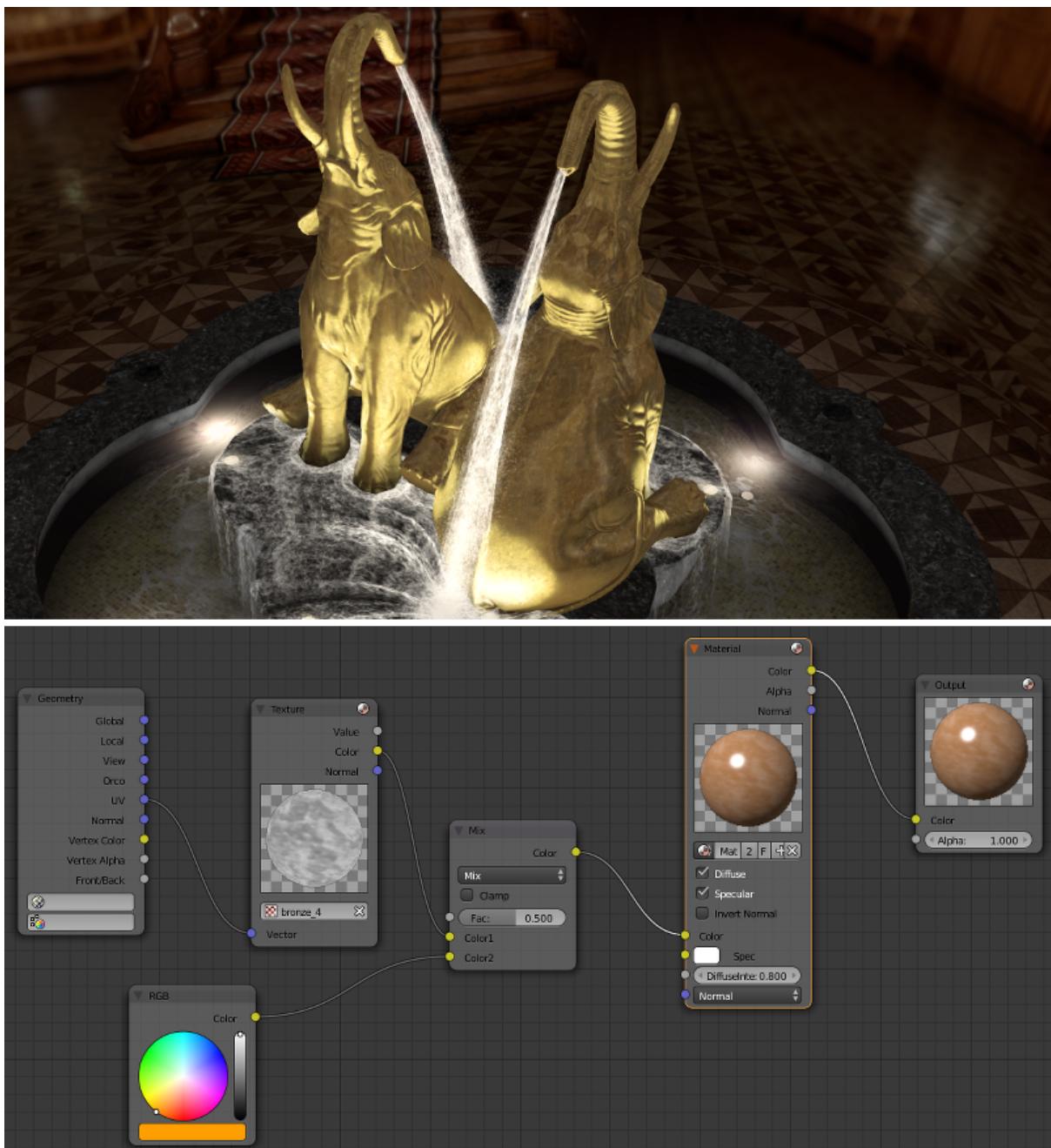
注解: 每一个材质不仅考虑遮罩和纹理，而且顶点颜色和 UV 贴图也有自己的要求。

节点材质

目录

- 节点材质
 - 标准节点
 - 游戏引擎特定节点
 - * 钳制 (B4W_CLAMP)
 - * 夜光输出 (B4W_GLOW_OUTPUT)
 - * 质量层级 (B4W_LEVELS_OF_QUALITY)
 - * 视差 (B4W_PARALLAX)
 - * 反射 (B4W_REFLECT)
 - * 折射 (B4W_REFRACTION)
 - * 替换 (B4W_REPLACE)
 - * 平滑步进 (B4W_SMOOTHSTEP)
 - * 时间 (B4W_TIME)
 - * 半透明 (B4W_TRANSLUCENCY)
 - * 矢量视图 (B4W_VECTOR_VIEW)
 - * 线性到 SRGB 和 SRGB 线性 (已取消)
 - Cycles 节点

着色器节点通过将现有的着色器进行基本变换和批处理，显著地扩展了 Blender 的标准材质的潜力。



标准节点

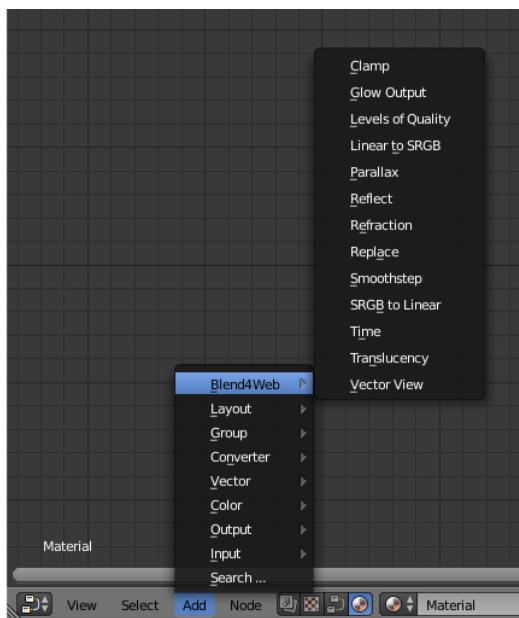
Blend4Web 支持所有标准的 Blender 节点，但他们中的一些工作的不够快，因此不建议在实时应用程序中使用。创建非常复杂的材质，尤其是使用大量的 Geometry 和 Texture 节点，也不推荐。

引擎能部分支持某些 Cycles 节点。本内容的更详尽描述在相应章节。

标准节点性能和支持度在 表中描述.

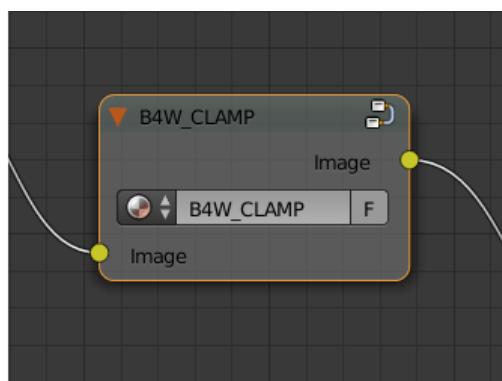
游戏引擎特定节点

引擎特定的节点扩展标准节点以支持额外的功能。这些节点采用特殊命名的名称和输入格式创建为节点组（节点组或 节点树）。为方便起见，被打开时，所有的特殊节点已经添加到 Blend 文件。



钳制 (B4W_CLAMP)

节点限制输出值。其结果是所有的输出矢量分量取从 0 到 1 中间的值。



输入参数

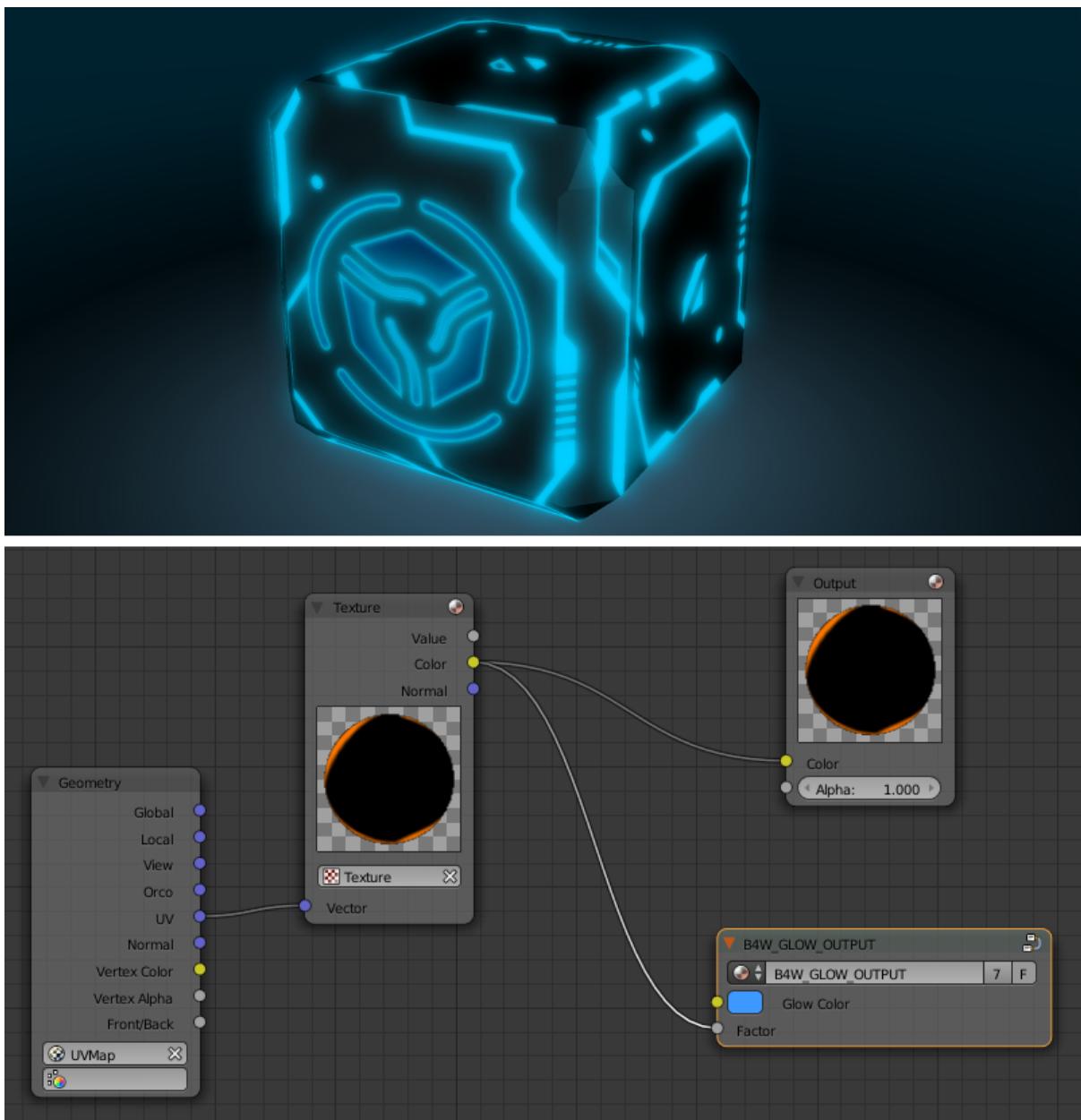
图像 关联向量

输出参数

图像 钳制向量

夜光输出 (B4W_GLOW_OUTPUT)

应用[发光效果](#) 到节点材质。除了 B4W_GLOW_OUTPUT 节点，该节点材质应具有输出节点。



输入参数

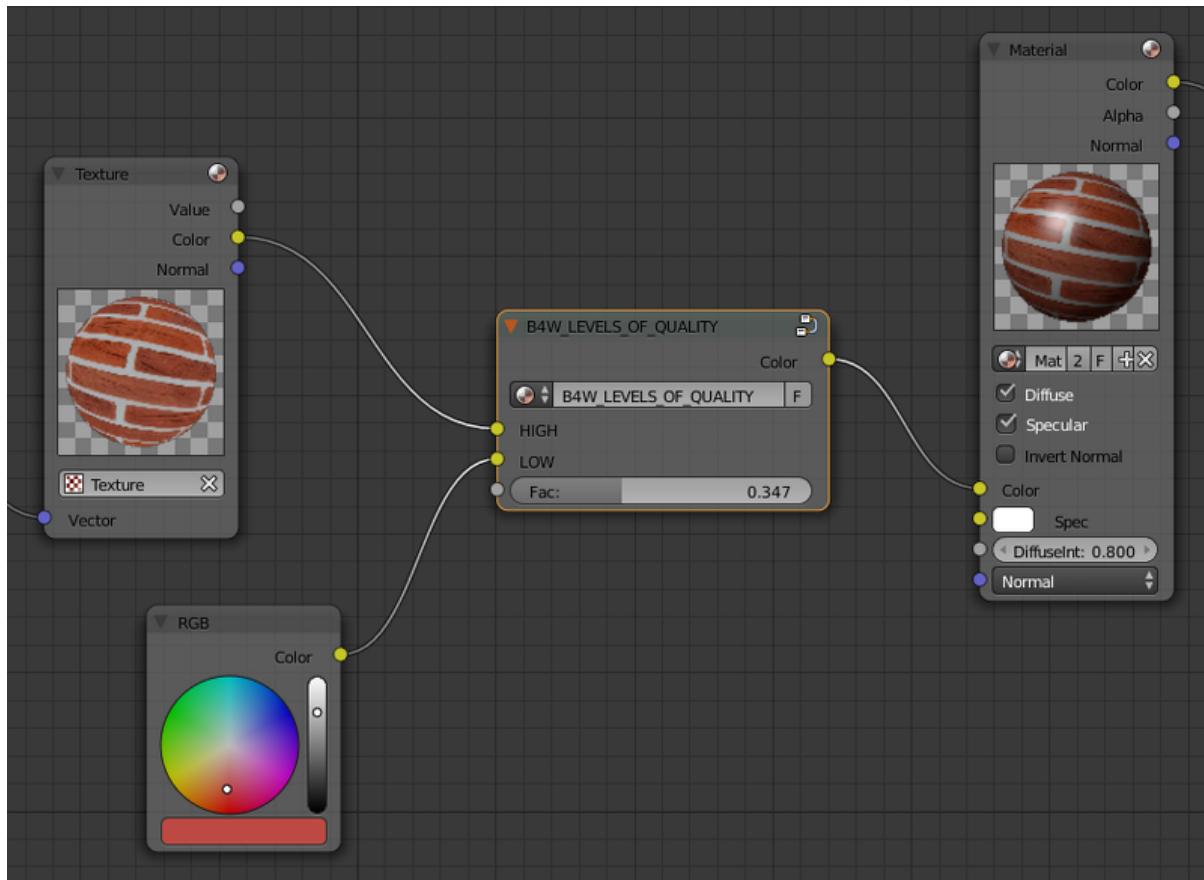
发光颜色 发光颜色

系数 发光的比例。系数 $\in [0, 1]$.

- 系数 = 0 - 没有发光。
- Factor $\in (0, 1]$ - 有一个发光，与 发光颜色同色。

质量层级 (B4W_LEVELS_OF_QUALITY)

设置根据当前的图像质量设置输出颜色。可用于，例如，与如果应用程序在移动装置上运行用更简单材质的更换复杂材质。



参见:

[品质配置](#)

输入参数

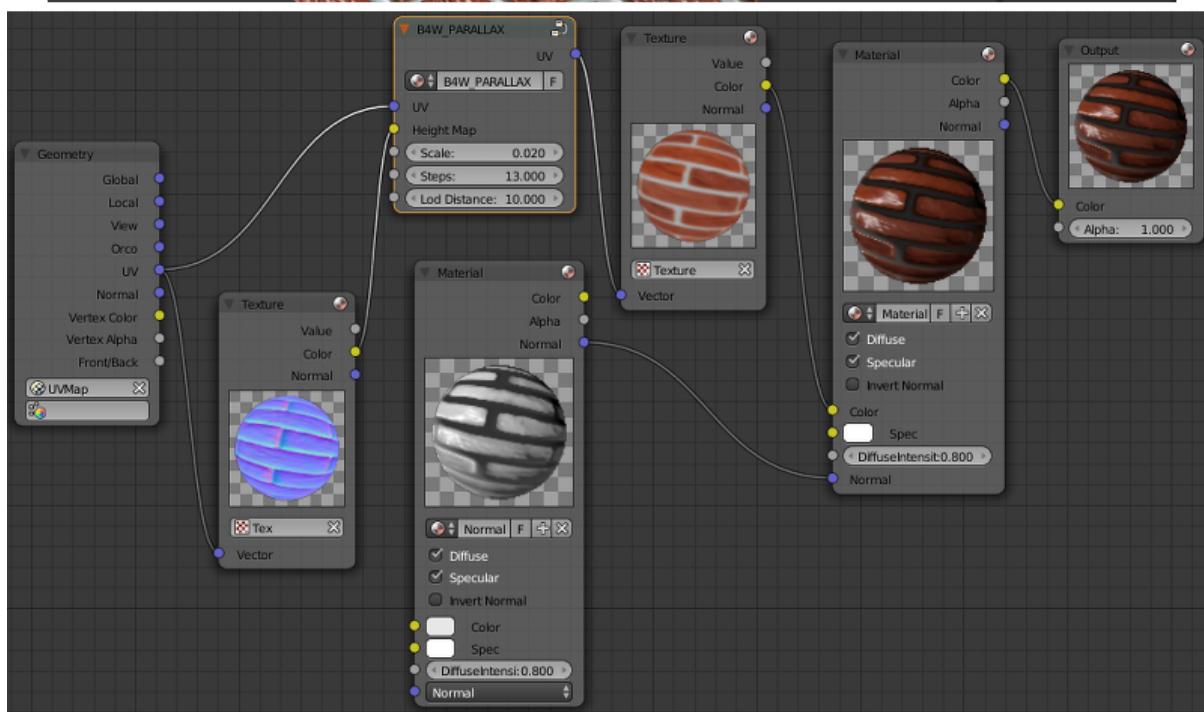
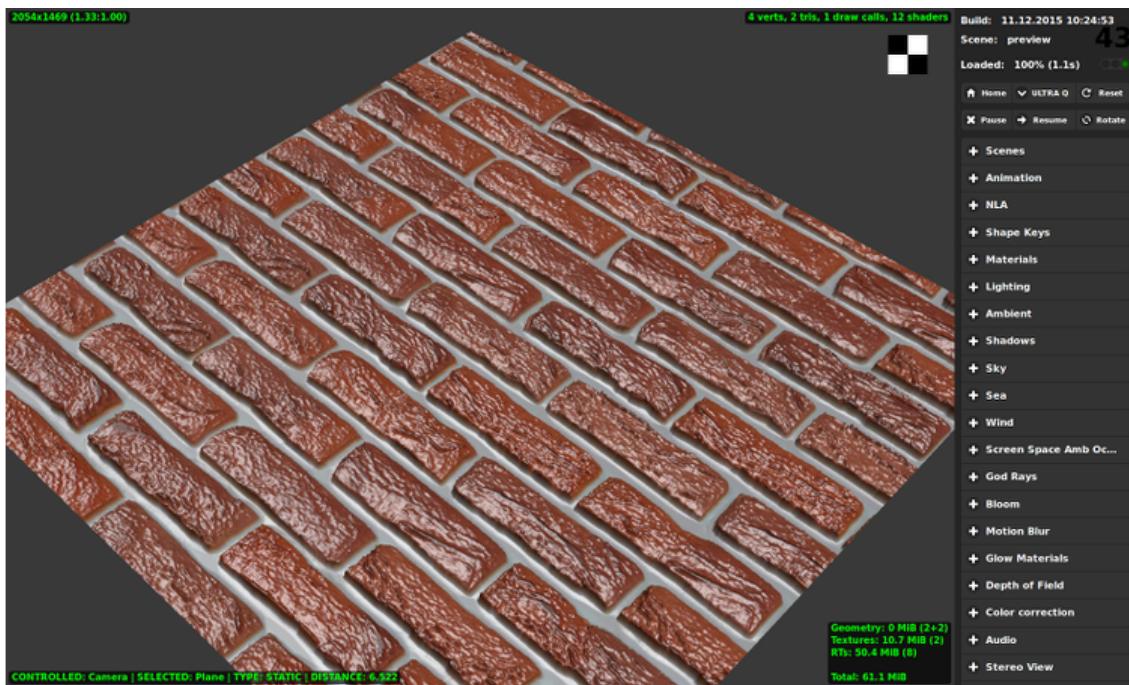
高质量 节点链接这个参数到 Color 参数高，最高的质量使用的情况。

低 节点链接这个参数到 Color 参数低质量的使用情况。

系数 此值指定什么质量设置 (HIGH 或 LOW) 将在 Blender 视口中可见。可以从 0 变化到 1，如果该值低于 0.5 时，HIGH 设置将是可见的，如果它是 0.5 或更高时，LOW 设置将是可见的。

视差 (B4W_PARALLAX)

该节点使用高度贴图实现纹理坐标偏移。



输入参数

UV 源纹理坐标。

高度贴图 RGBA 纹理压缩到 alpha 通道的高度图。

缩放 纹理坐标偏移的因子。

步进 迭代产生纹理的步数坐标偏移。这个值越大最后的质量越好。

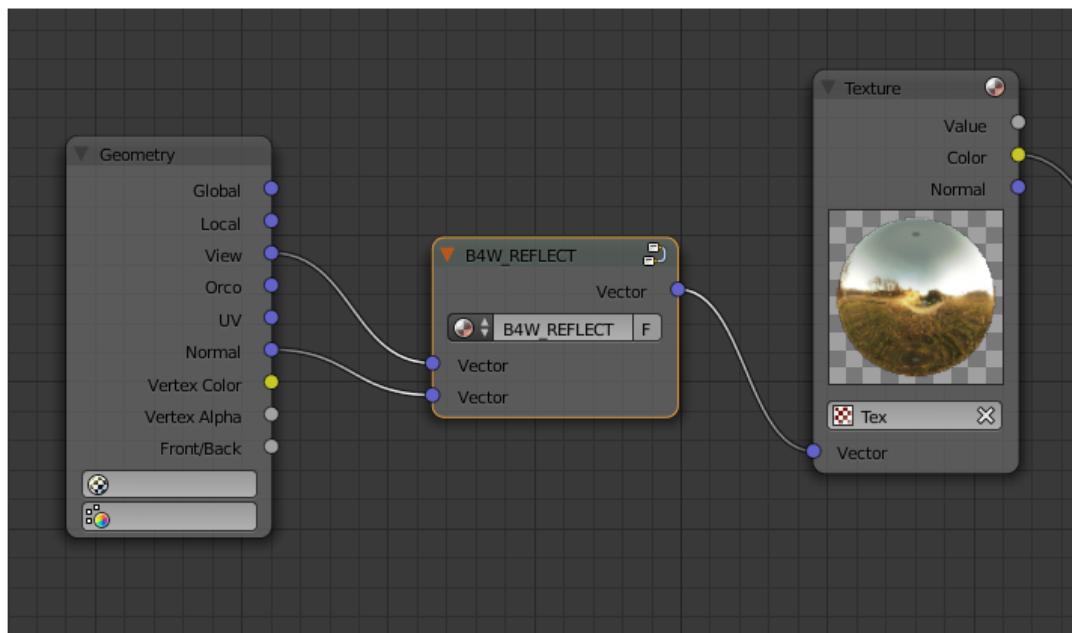
LOD 距离 从相距距离最远开始，效果能被观察到。

输出参数

UV 结果是纹理坐标用着输入纹理节点。

反射 (B4W_REFLECT)

计算入射矢量和法线矢量之间的反射方向。可用方块贴图应用到物体。



输入参数

矢量 入射向量。应连接至 Geometry 节点的 View 插座。

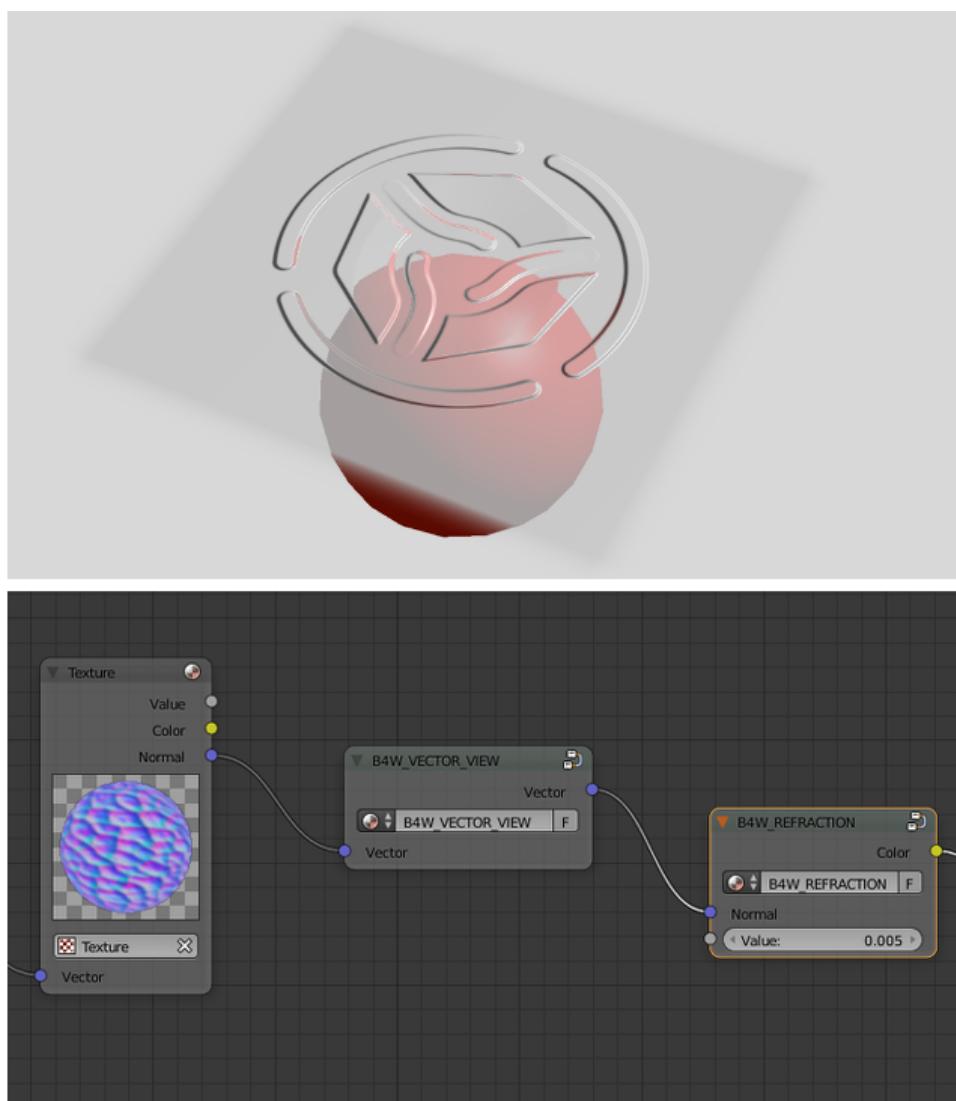
矢量 法向量。需要进行归一化以达到所期望的结果。应连接至 Geometry 节点的 Normal 插座。

输出参数

矢量 反射向量。应该连接到包含立方贴图的 Texture 节点的 Vector 插座。

折射 (B4W_REFRACTION)

一个物体产生折射效果。这种效果仅在 Blend4Web 引擎，而不是在 Blender 的视口。



输入参数

法线 添加扰动法线贴图。

折射凹凸 扰动强度值。

默认值是 0

输出参数

Color 对物体后面的材质产生扰动。

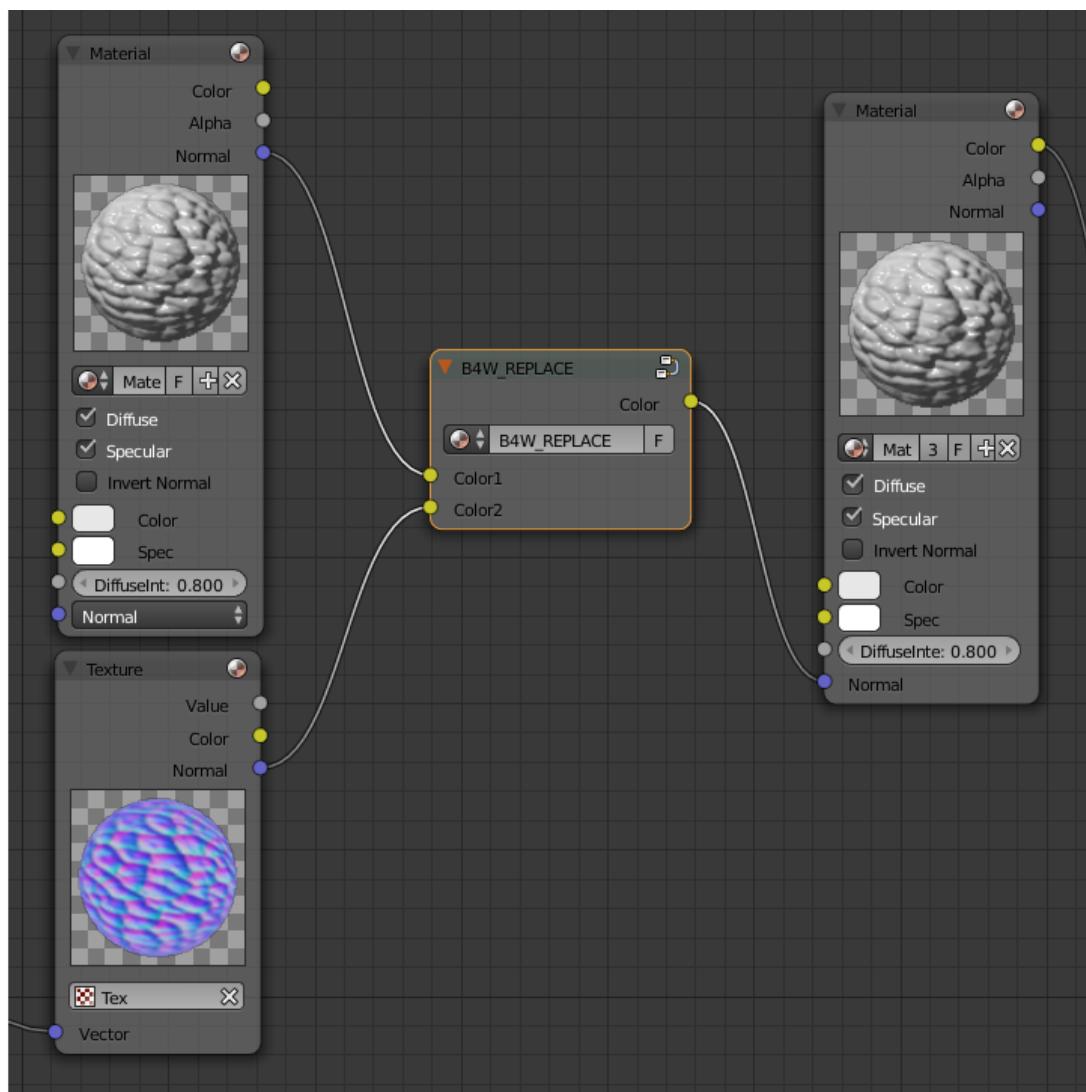
注解: 需要在 Render > Reflections and Refractions 面板设置 Refractions 选项值为 AUTO 或 ON 。物体的透明度类型必须设置为 Alpha Blend.

参见:

[透明](#)

替换 (B4W_REPLACE)

节点替换取决于工作环境 (即 Blender 视口或 Blend4Web) 的输入。在 Blender 工作时, Color1 输入被连接到 “Color“ 输出和 Color2 输入被忽略。相反在引擎中输入被替换时 (这时 Color1 被忽略, Color2 被连接到输出)。节点意欲在视口中显示的一个节点的结构和其他 - 在引擎中。



作为一项规则是用于法线贴图。Blender 的节点材质不支持坐标的切线空间。因此，唯一可能的在视口中来显示法线贴图正确的方法是在 Material 节点内的使用。

输入参数

Color1 节点的设置，这将是在 Blender 视口中可见。

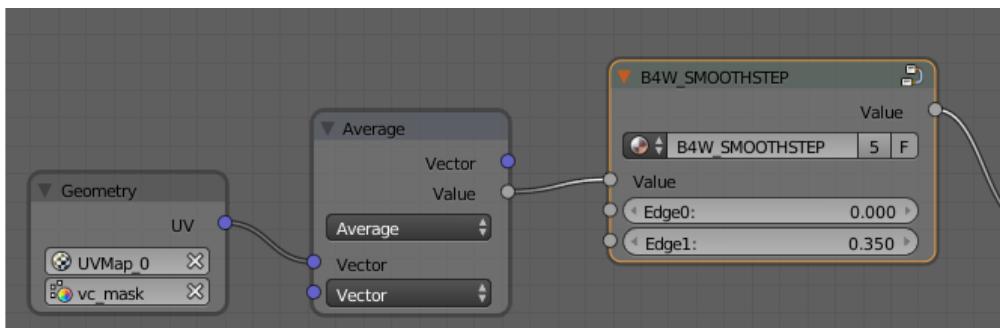
Color2 节点设置，就可以在 Blend4Web 引擎可见。

输出参数

Color 应连接到 Material 或 扩展 Material 节点的 Color 插座。

平滑步进 (B4W_SMOOTHSTEP)

进行基于第一值的两个输入值之间的平滑内插。



输入参数

值 决定插值的平滑度的值。

Edge0 首个插值。

Edge1 第二插值。

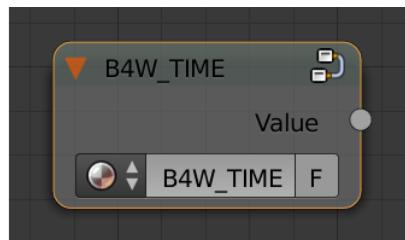
输出参数

值 插值。

注解: 对于正确的插值, 输入 Value 必须在 Edge0 和 Edge1 之间。

时间 (B4W_TIME)

提供时间表, 从引擎起动计数 (以秒计)。可用于在节点的材质, 如 UV 坐标动画任何参数, 混合的因子, 透明度等。



输入参数

无

输出参数

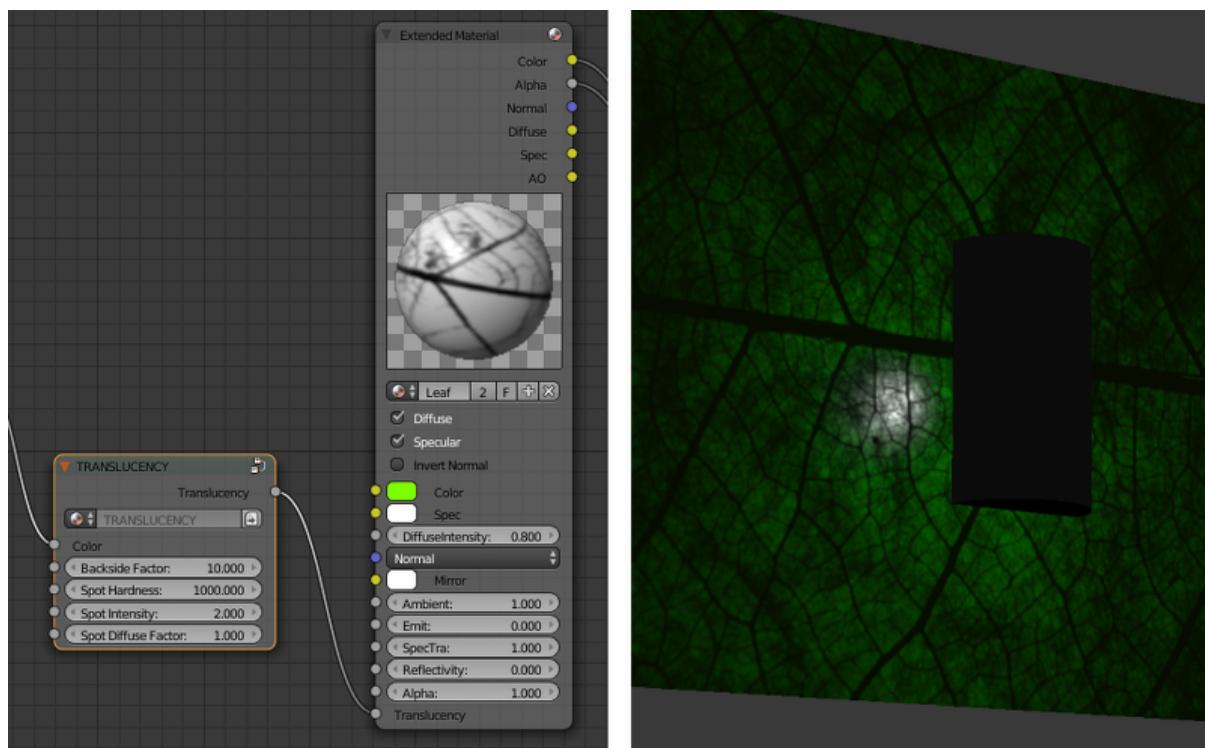
值 从引擎起动开始经过的时间（秒）。

参见：

[值和 RGB 节点的动画](#)

半透明 (B4W_TRANSLUCENCY)

节点实现了一个半透明效果（只对于光源）的薄的物体，如布，树叶，纸张等的效果，由两部分组成：1) 物体侧的增亮这是相对的光源和 2) 光点就在光源处的外观。



输入参数

Color 它定义了材质异质性单通道纹理 - 白色表示最大的半透明效果，而黑色代表不存在。白色默认情况下使用。

背面因子 为这是相对的光源侧的材质颜色校正系数。它描述了半透明区域的颜色丰富性的效果。

- 背面系数 < 1 - 增亮
- 背面系数 $= 1$ - 无修正
- 背面系数 > 1 - 变黑

默认值是 0

光点硬度 光点模糊的系数。这个值越大就越小，当场和更清晰的现货边缘。缺省值为 1000。

斑点强度 光斑强度。这个值越亮越大，光点。默认值是 1。

光斑漫射系数 光斑的颜色材质漫反射颜色影响。

- 光斑漫射系数 $= 0$ - 光斑有漫反射颜色
- 光斑漫射系数 $= 1$ - 光斑颜色为白色

默认值是 0

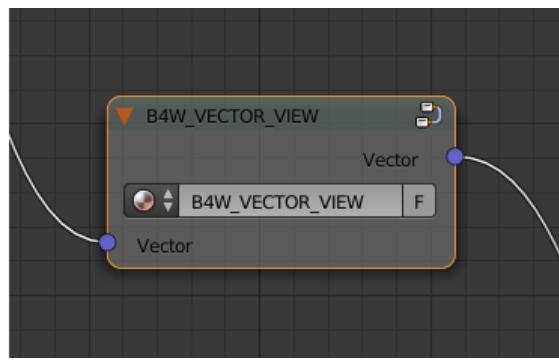
输出参数

半透明 输出应被连接到 扩展 Material 节点的 Translucency 输入。

注解: 如果网格法线被编辑了，这个节点不会正常工作。

矢量视图 (B4W_VECTOR_VIEW)

该节点将一个向量坐标转换到相机的空间。转换是必要的，因为引擎定义矢量坐标系在世界空间。如果正常矢量是由该节点转变它应该只对影响，不用于连接到 Material 或扩展 Material 节点的输出。



输入参数

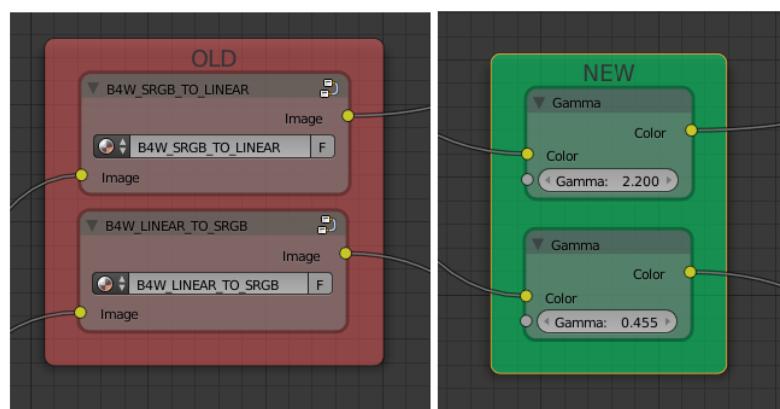
矢量 向量的坐标在世界空间。

输出参数

矢量 矢量坐标在摄像机空间。

线性到 sRGB 和 sRGB 线性（已取消）

从 sRGB 到线性空间转换颜色反之亦然。从版本 15.04 这个功能已经被宣布弃用。在较新的版本，具有 2.200 的数值内置 Gamma 节点应使用将颜色从 sRGB 转换为线性空间，并用 0.455 的值相同的节点将颜色从线性空间转换为 sRGB。



参见:

[伽马在节点材质中](#)

Cycles 节点

注解: Cycles 节点的支持是不建议在生产中使用环境的实验性功能。

此外应指出的是，在 Blend4Web 中使用 Cycles 节点会产生相似的图像，但不是完全跟使用 Cycles 渲染器本身创建的图像一样。

引擎支持以下 Cycles 节点F

- 材质输出 (只支持 Surface 和 Displacement 输入)；
- BSDF Diffuse;
- BSDF Glossy (只支持 GGX 分布; Roughness 参数不影响反射);
- Transparent BSDF;
- Mix Shader;
- Fresnel;
- Layer Weight;
- Image Texture;
- Environment Texture;
- Object Info;
- Bump.

部分支持以下节点F

- Texture Coordinates (From Dupli 参数不支持);
- UV Map (From Dupli 参数不支持);
- Geometry (Pointness 和 Parametric 参数不支持).
- Emission (不会影响场景中的打灯效果).

其他 Cycles 节点在大多数情况下在不会在 Blend4Web 中跟他们在 Blender 中一样良好工作。他们不会工作也可能引起材质的不正常显示和工作。但是，使用这些节点不会在应用程序工作流中导致不稳定。

逻辑编辑器

目录

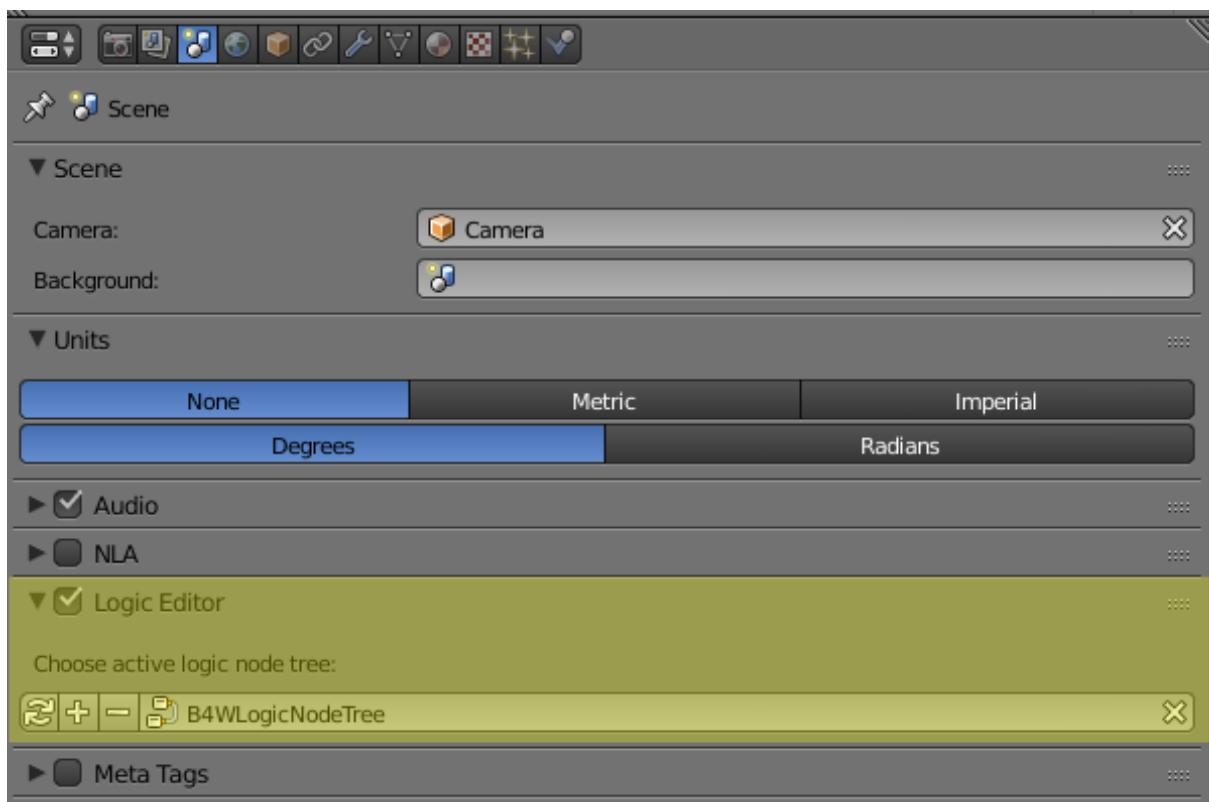
- 逻辑编辑器
 - 基础
 - 控制流
 - 动画
 - 相机 (Camera)
 - 物体
 - 操作
 - 声音
 - 网络
 - 调试
 - 已过时
 - 布局
 - 调试中

基础

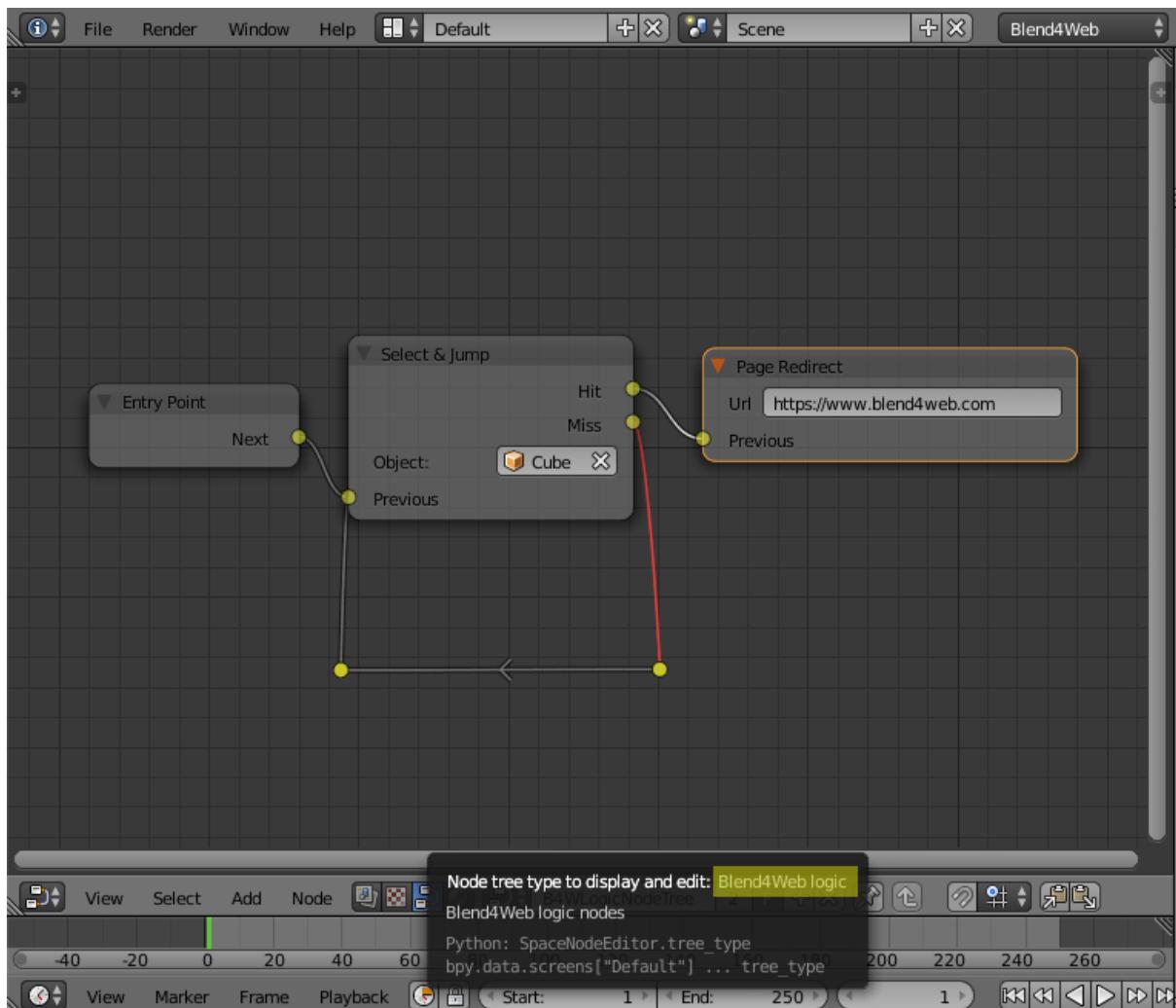
可视化编程是通过在 Blender 的节点编辑器创建逻辑节点树进行。这些节点可以无需任何编码也能显著的增强场景的功能性。



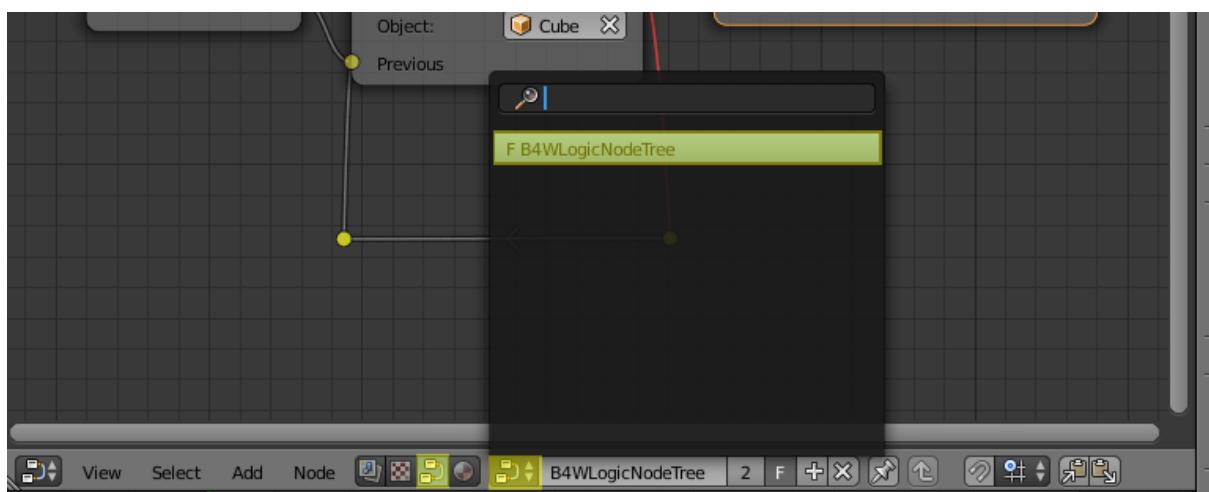
要在给定场景激活逻辑，在 Scene 选项卡中设置 Logic Editor 属性并追加/选择活动节点树。



要编辑逻辑树需要到 节点 Editor 区域并选择 Blend4Web Logic 树类型：



然后选择所需的节点树：

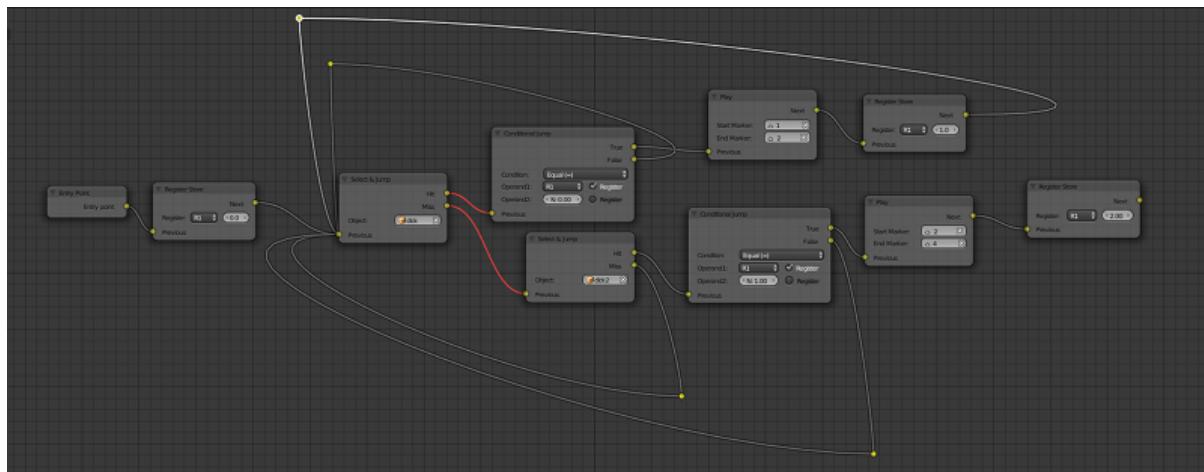


通过使用标准的 Blender 键盘快捷键 Shift+A 创建节点。

本身就是逻辑块的节点从只有一个输出没有输入的 Entry Point 节点开始。所有其他节点有输入和输出，并可以在逻辑树的任何地方插入。例外的是页面重定向节点，仅可以在末端节点树插入。有两个输出的节点允许分支，因此基于为此类节点指定条件可以在下一个层级做选择。

为实现复杂的逻辑有可以有数值或字符串值的变量。变量可以用于存储一些场景状态（例如，这可以是动画播放计数器，角色的健康点等）。

逻辑编辑器使用示例：

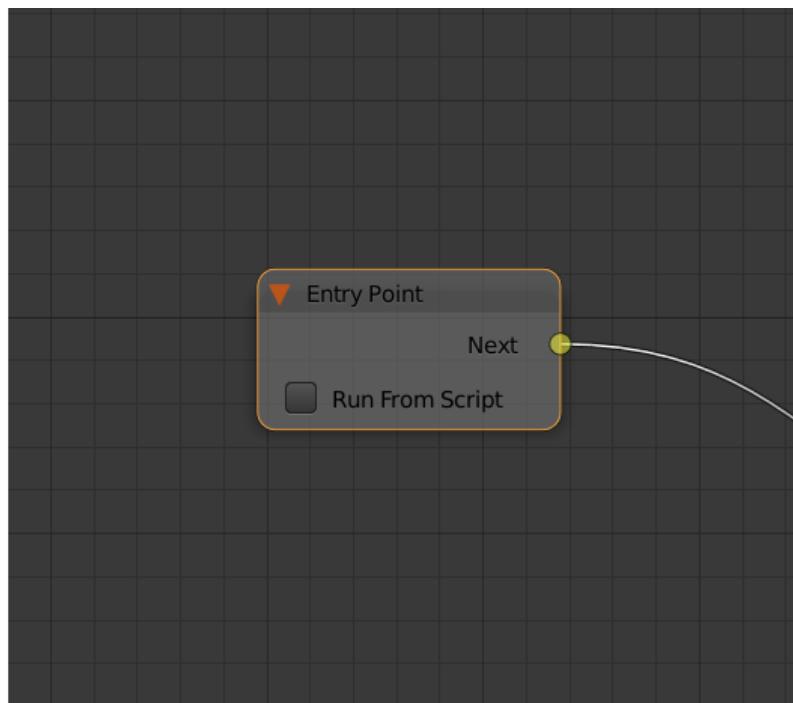


下文介绍了所有可用的节点。

控制流

入口点

这是该脚本开始的地方。通过使用多个人口点您可以创建多线程应用程序。



输入参数

空

输出参数

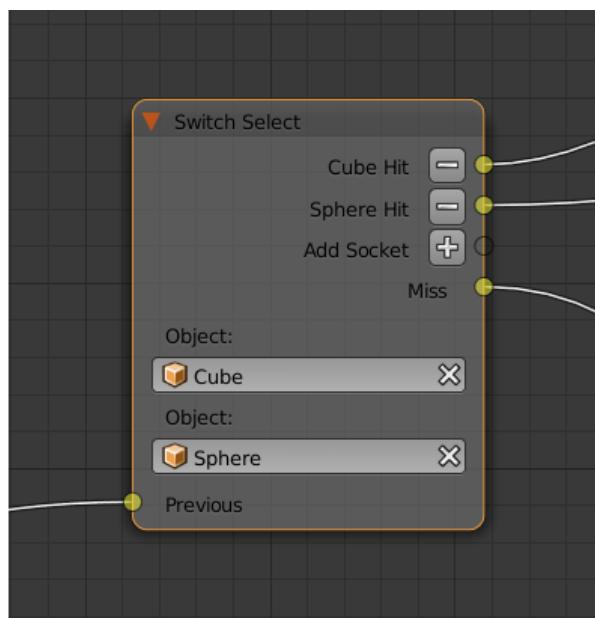
下一个 下一个节点。

内部参数

从脚本运行 如果启用此参数，切入点可以通过 API 使用 `run_entrypoint` 方法触发。

切换选择

可用于跟踪物体列表中的任何物体的选择。



输入参数

上一个 前一个节点。

输出参数

<object name> 命中 如果用户选择（用鼠标或触摸）一个物体的参数名称被设置关注，这个参数将传递到下一个节点的控制。该 Switch Select 节点在默认情况下有一个这样的参数，但可以添加新的和删除现有（节点甚至可以没有这样的参数）。

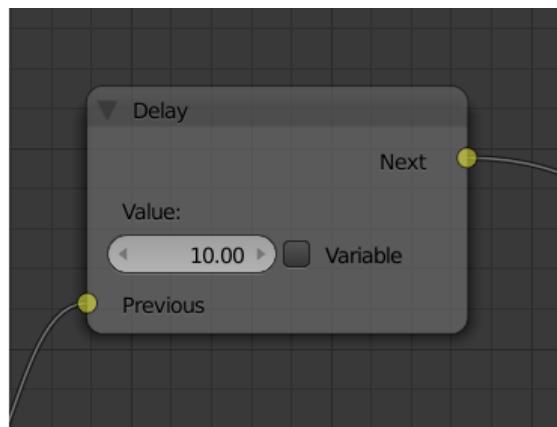
未击中 This parameter will pass the control to the next node when the user selects any object with the Selectable property enabled (or used by another Switch Select node), but not specified in the Switch Select node.

内部参数

物体 用户可以选择的物体之一。当您创建或删除命中参数时，这些参数会自动创建和删除 Hit 参数。这样的参数的数量总是等于 Hit 参数的数目。

延迟

在去下一个节点之前做一个延迟（以秒计）。



输入参数

上一个 前一个节点。

输出参数

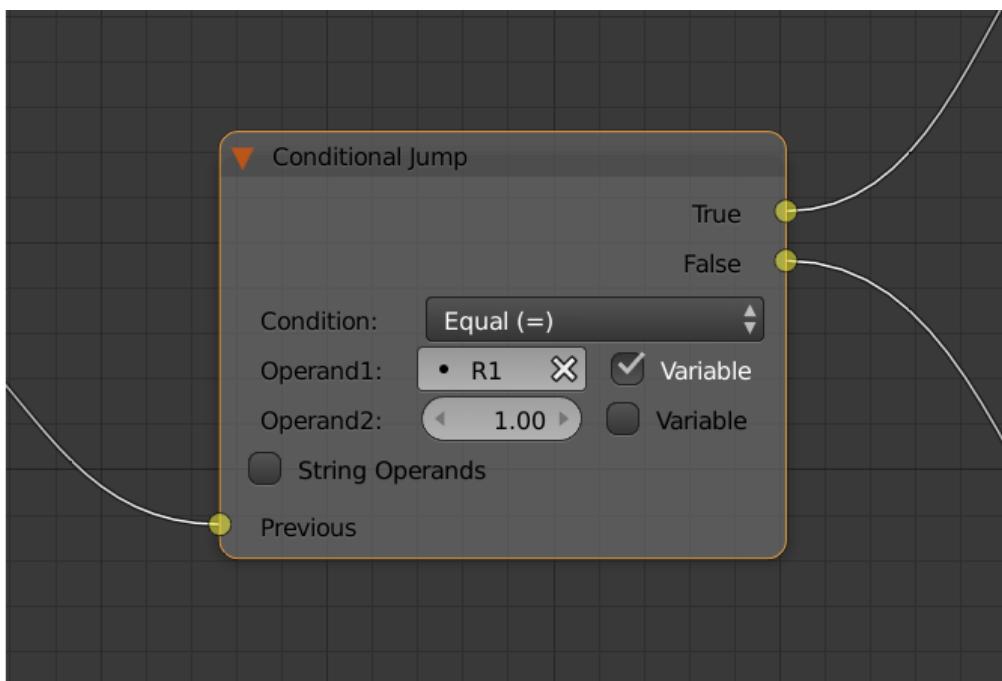
下一个 下一个节点。

内部参数

值 将在下一个节点激活之前的时间（以秒为单位）。默认情况设置为零。可以手动或通过一个变量来设置（如果启用了 Variable 参数）。

条件跳转

如果满足一定的条件转到指定的节点。参数（操作数）也可以是使用相应的开关激活的变量。



输入参数

上一个 前一个节点。

输出参数

真 条件为真。

假 条件是假的。

内部参数

条件 逻辑条件。可以有以下几种类型：

- Equal (=) - 第一个操作数等于第二个。
- Not Equal (!=) - 第一个操作数不等于第二个。
- Less Than (<) - 第一个操作数小于第二个。
- Greater Than (>) - 第一个操作数大于第二个。
- Less Than Or Equal (<=) - 第一个操作数小于或等于第二。
- Greater Than Or Equal (=>) - 第一个操作数是大于或等于第二个。

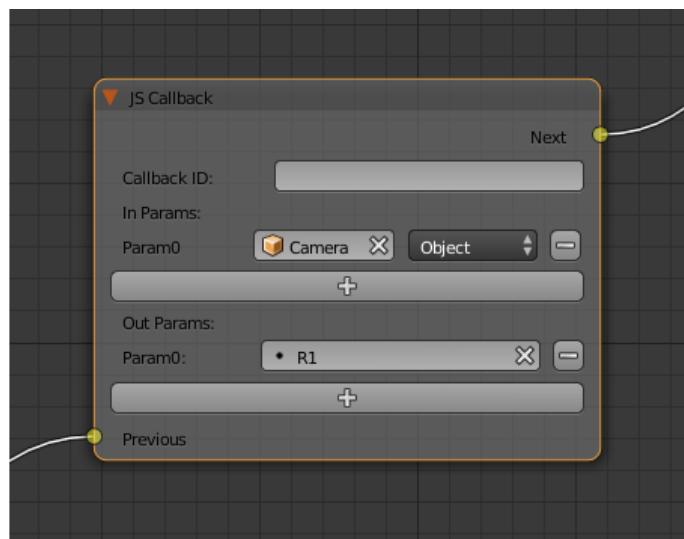
Operand1 First operand of the logical condition. Contains a numeric value of a string (if the String Operators parameter is enabled). Can be specified in the node or can be a link to one of the variables (if the Variable parameter at the right of it is enabled).

Operand2 逻辑条件的第二个操作数。和第一个以同样的方式工作。

String Operands If this parameter is enabled, first and second operands can use strings (set manually or by a variable) as their values.

JS 回调

可以用来调用您的 Blend4Web 应用程序中定义的自定义 JavaScript 回调。



输入参数

上一个 前一个节点。

输出参数

下一个 下一个节点。

内部参数

回调 ID 一个 JavaScript 函数将被节点调用的 ID。

输入参数 输入函数的参数的列表。每个参数可以是变量或到场景物体的链接。可以调整输入参数的数目。默认情况下，此列表是空的。

输入的参数传递给回调函数作为数组来作为第一个函数的参数。

参数 <param_number> 指定输入的参数。此参数可以是一个变量（如默认的 R1）或链接到一个场景物体，根据 Type 参数（这些参数每一个总是有相应的 Type 参数）的值。

类型 相应的输入参数的类型。它可以有两个值之一：Variable（在这种情况下，其中一个变量将用作参数）和 Object（链接至一个场景物体）。

输出参数 输出参数的列表。默认情况下为空。可以调整输出参数的数目。

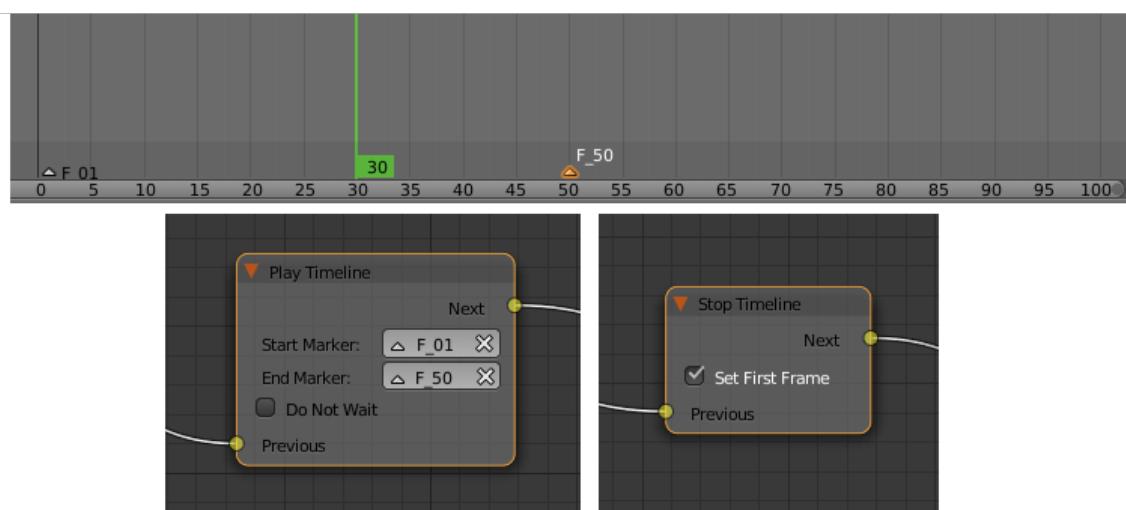
一个数组，包含输出参数作为回调函数的第二个参数。

参数 <param_number> 指定变量之一作为输出参数。默认情况下，使用 R1 变量。

动画

播放时间轴和停止时间线

可以用来控制 NLA 动画。'玩的时间线' 节点扮演 NLA 片段从指定的标记的帧开始。动画播放直至遇到下一个标记，或到场景的时间线的末尾。在那之后，控制传递到下一个节点。'停止时间线' 节点停止播放。



输入参数

上一个 前一个节点。

输出参数

下一个 下一个节点。

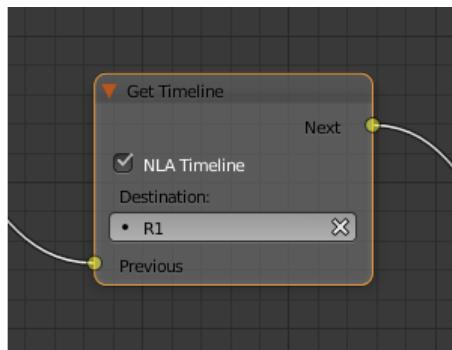
内部参数

起始标记 动画的第一帧。如果没有指定，动画从时间轴开始播放，并可能无法正常工作。

结束标记 动画的最后一帧。如果没有指定，动画播放到时间轴的结束，可能无法正常工作。

获取时间线

此节点可以用于获取 NLA 动画或时间线的当前帧。



输入参数

上一个 前一个节点。

输出参数

下一个 下一个节点。

内部参数

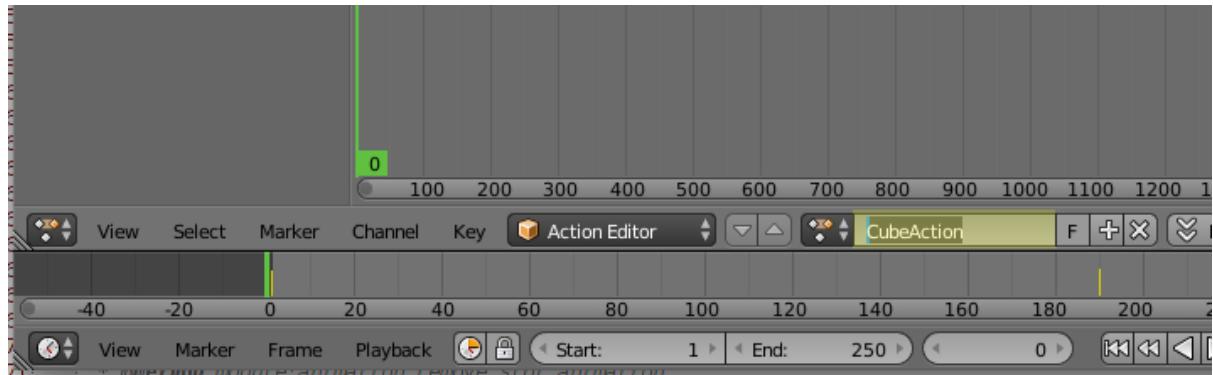
NLA Timeline 如果启用此参数，则该节点将返回当前动画的帧的 NLA。如果禁用了它，则该节点将返回当前帧的时间线。默认情况下启用。

目标 指定一个变量来存储当前帧的数量。默认设置为 R1。

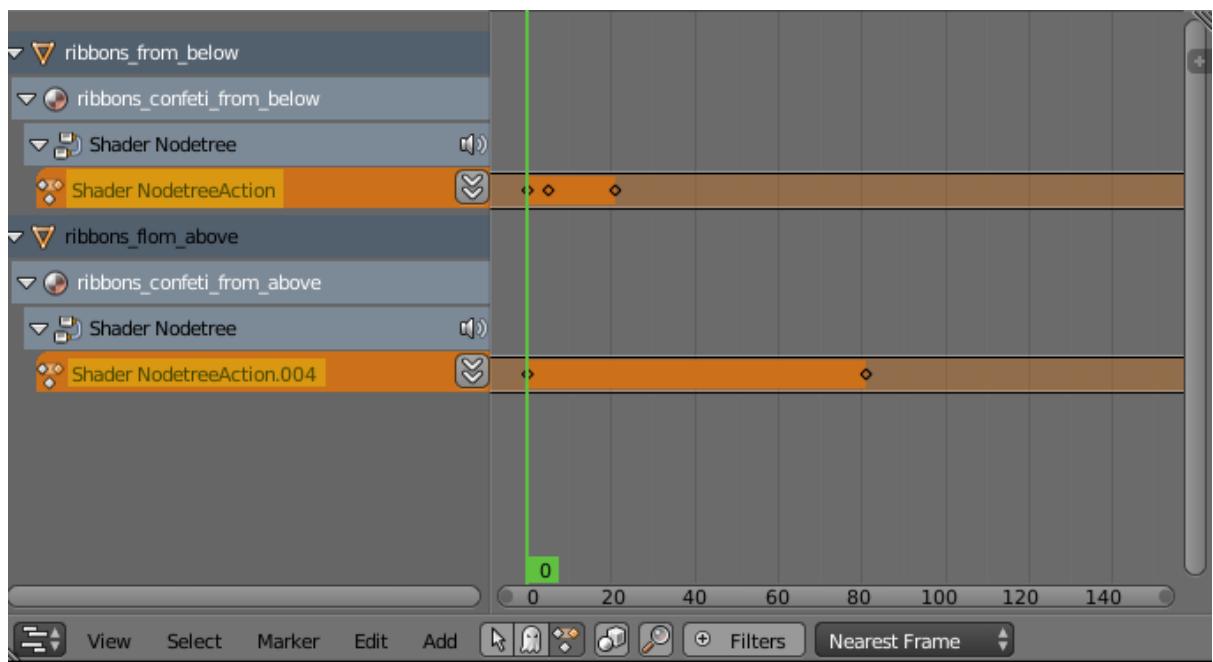
播放动画

可用于播放物体的动画。动画可以是以下类型之一：

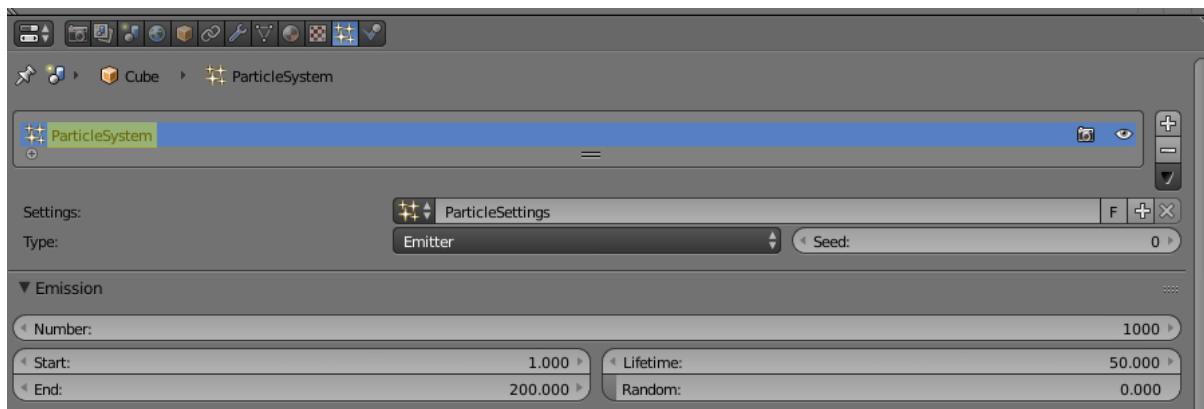
常规动作：



着色器动作：



粒子系统：



输入参数

上一个 前一个节点。

输出参数

下一个 下一个节点。

内部参数

物体 物体, 动画, 其中将播放的名称。

动画。名称 要播放动画的名字。如果未指定, 则整个时间轴将播放。

行为 指定动画行为。可以有以下值之一:

- 完成停止 - 动画将播放一次。
- 完成重置 - 动画将被播放一次, 则该物体将返回到起点。
- 循环 - 直到它被 停止 Animation 节点停止的动画将被反复播放。

不要等 如果该参数被启用, Play Animation 节点将控制传递到下一个节点上启动动画播放。如果不是, 控制将被传递到播放完成后才会进入下一个节点。

停止动画

可以用来停止物体的动画。

输入参数

上一个 前一个节点。

输出参数

下一个 下一个节点。

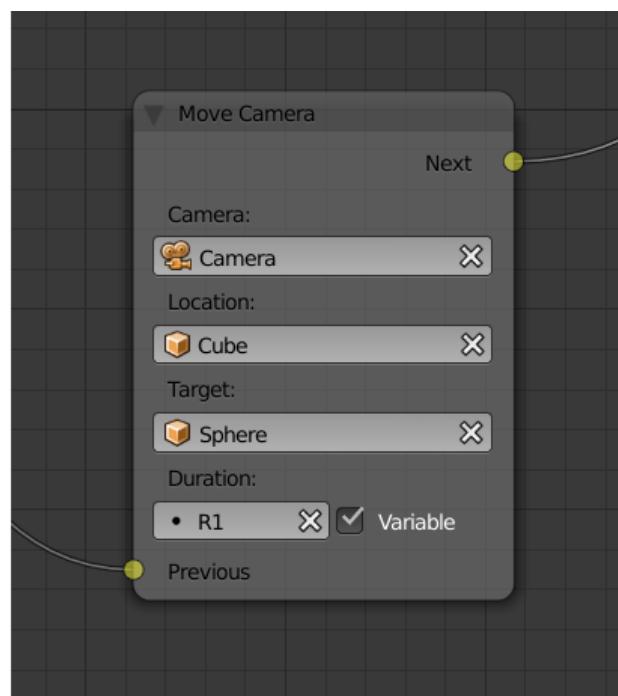
内部参数

设置第一帧 返回到第一帧的动画已经停止之后。

相机 (Camera)

移动相机

可用于移动相机，包括其位置的平滑插补。



输入参数

上一个 前一个节点。

输出参数

下一个 下一个节点。

内部参数

相机 将摄像机移动。

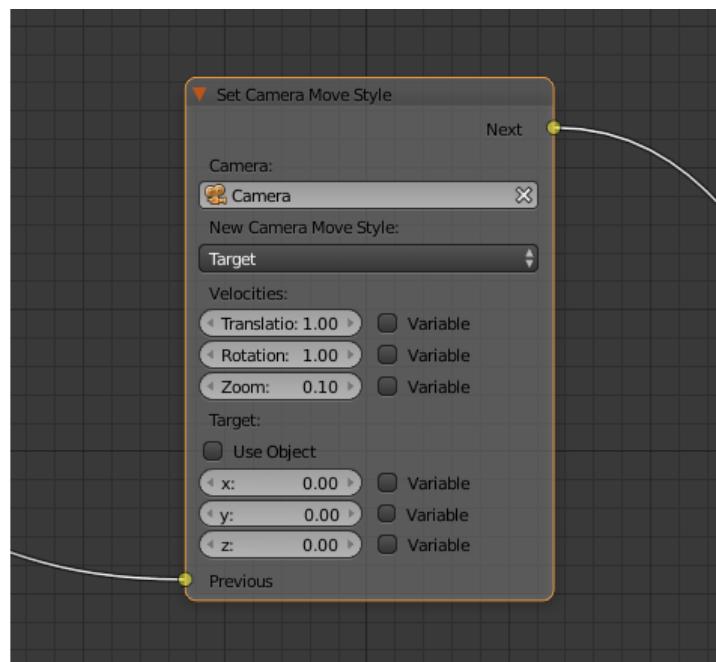
位置 相机将移动的物体。相机的坐标跟运动结束后的物体是相同的。

目标 相机将在指向该物体的方向移动。

持续 相机被转移到一个新的位置将花费的时间（秒）。默认设置为零（在这种情况下，相机实际上并没有移动，它只是改变了它的位置）。它可以手动指定或作为一个变量的一个链接（如果是启用了变量的参数）。

设置相机移动方式

可以用来改变所选摄像机的移动方式。



输入参数

上一个 前一个节点。

输出参数

下一个 下一个节点。

内部参数

相机 此参数指定的摄像机到的变化将被应用。

新相机移动方式 此参数指定相机将使用新的方式。四个选项可用：Hover, Eye, Target and Static.

以下选项只是 New Camera Move Style 参数未设置为 Static 时才可用

变换 设置相机变换速度。默认情况下，此参数设置为 1.0。

旋转 设置摄像机的旋转速度。默认情况下，此参数设置为 1.0。

放大 设置相机的变焦速度。默认值为 0.10。只有 New Camera Move Style 参数设置为 Hover 或 Target 时才可用。

以下参数用于指定目标或相机的轴心点和仅在 New Camera Move Style 参数设置为 Hover 或者 Target 时才可用

使用物体 此参数启用和禁用使用场景物体作为相机的目标或轴心点（取决于相机类型）。如果它激活，一个文本字段将显示在指定的物体。此参数默认被禁用的。

如果“使用物体”参数被禁用，以下三个选项可用：

x 相机的目标/轴心点的 X 分量。

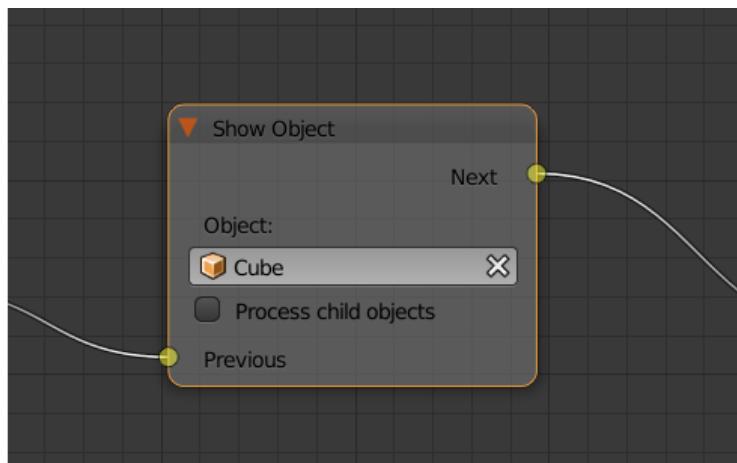
y 相机的目标/轴心点的 Y 分量。

z 相机的目标/轴心点的 Z 分量。

物体

显示物体

可用于显示 3D 物体。



输入参数

上一个 前一个节点。

输出参数

下一个 下一个节点。

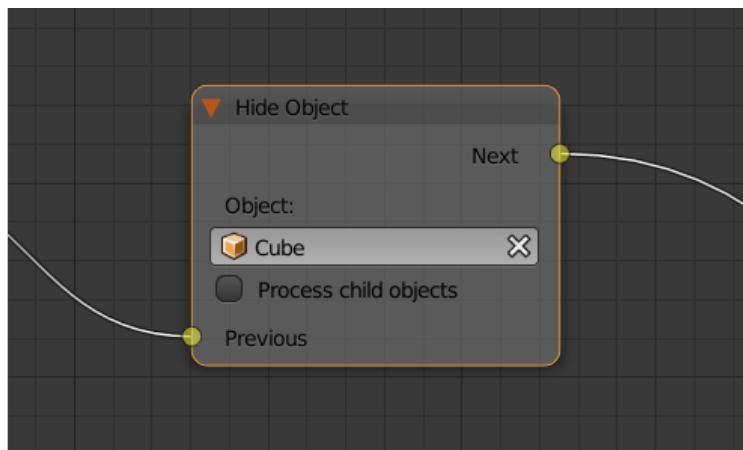
内部参数

物体 显示一个物体。

处理子物体 如果启用此参数，则将显示子物体。

隐藏物体

可以用来隐藏三维物体。



输入参数

上一个 前一个节点。

输出参数

下一个 下一个节点。

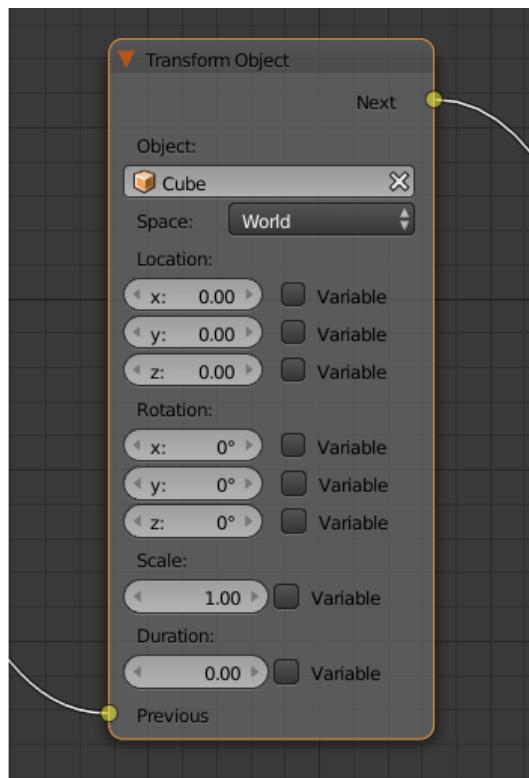
内部参数

物体 一个物体隐藏。

处理子物体 如果启用此参数，则将隐藏子物体。

变换物体

可以用来改变物体的位置，大小和旋转角度。



输入参数

上一个 前一个节点。

输出参数

下一个 下一个节点。

内部参数

物体 需要进行变换的物体。

空间 此参数定义将用于转换的物体的坐标空间。它可以具有下列值之一

- World - 全局坐标空间。
- Parent - 物体在 Object 参数中指定的父物体的本地坐标系统。父物体的原点作为坐标的中心，而它的旋转角度定义坐标轴的方向。
- Local - 所选物体的局部坐标空间。类似于 Parent 坐标空间，但在这种情况下，物体本身的起始点作为坐标原点。

默认设置为 World。

位置 如何沿着 X, Y 和 Z 轴移动物体。默认情况下，所有三个参数设置为零。值可以指定节点本身或通过变量（如果启用了变量选项）。

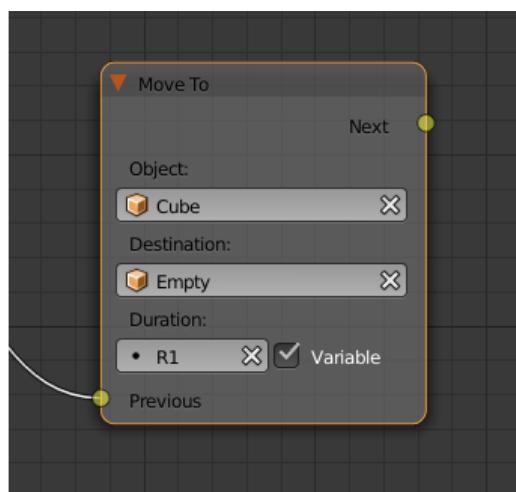
旋转 围绕 X, Y 和 Z 轴旋转物体。所有的三个值都设置为零，默认情况。可以指定节点中直接或通过变量（如果启用了变量选项）。

缩放 物体的大小。可以指定直接或通过一个变量（如果启用了变量参数）。默认设置为 1。

持续 变换需要的时间（以秒为单位）。它可以指定直接或用变量（若要做到这一点，变量参数应启用）。设置为零，默认情况。

移动到

可用于移动物体。



输入参数

上一个 前一个节点。

输出参数

下一个 下一个节点。

内部参数

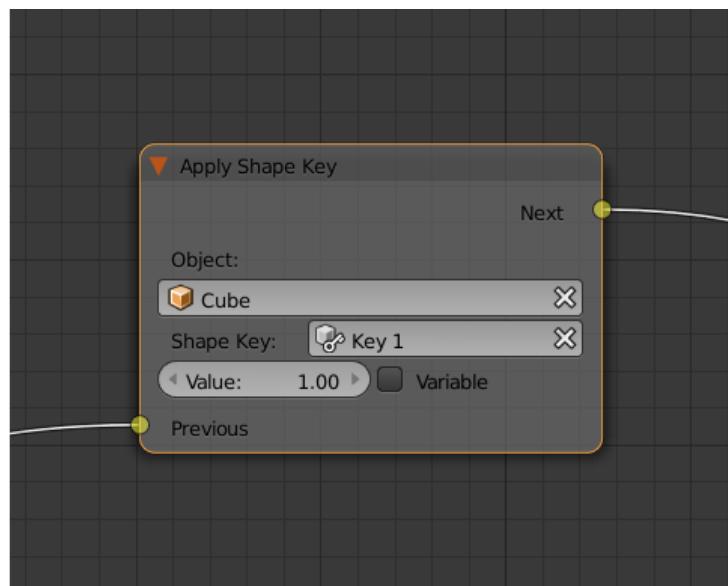
物体 你需要移动的物体。

目标 一个目标（另一个物体或光源，摄像机或其他东西）选择的物体会移动。物体的坐标和目标的运动完成后的将是相同的。

持续 该物体将移到新位置花的时间（以秒为单位）。默认情况下，此参数设置为零（和在这种情况下，物体实际上并不移动，它只是改变了其在一个时刻的位置）。它可以设置手动或用变量（仅当启用了变量参数可用）。

应用形状关键

设置形状关键因素。



输入参数

上一个 前一个节点。

输出参数

下一个 下一个节点。

内部参数

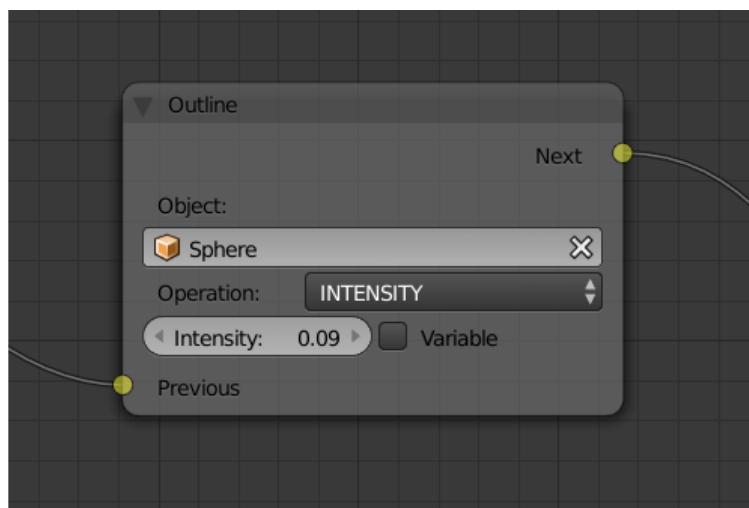
物体 待转化需要的物体。

形变关键帧 形状将被应用到物体的密钥。

值 多少的形状关键将影响物体。此值可以直接在节点中设置或使用的变量。值应该是 0 和 1 之间。

轮廓

控制物体概述效果。



输入参数

上一个 前一个节点。

输出参数

下一个 下一个节点。

内部参数

物体 物体的轮廓效果的任何更改将只适用于由该参数指定的物体。

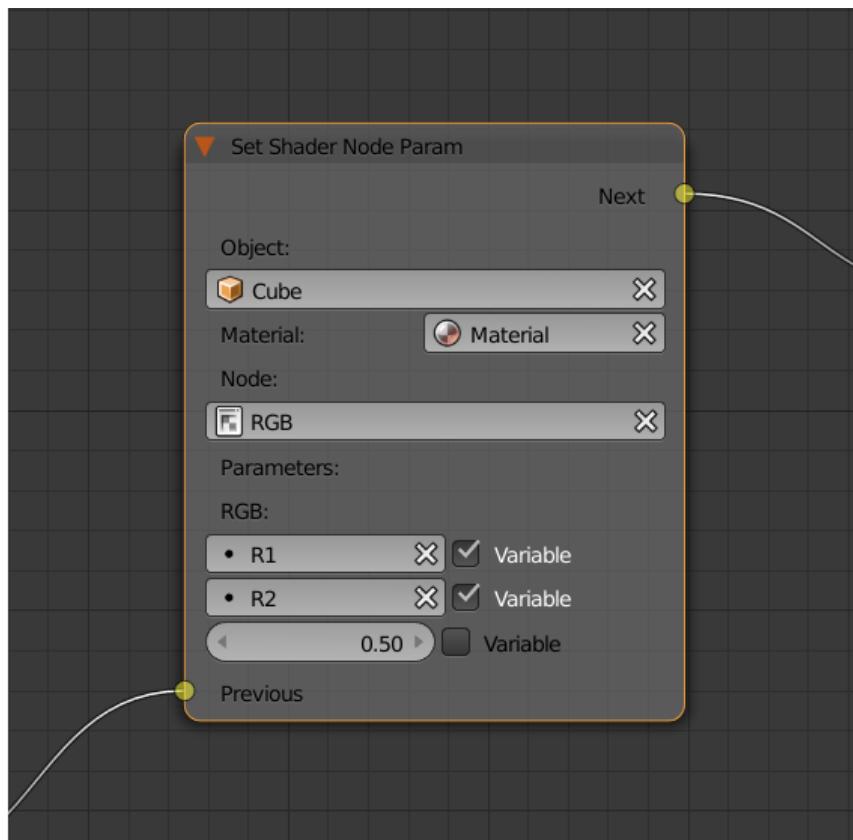
操作 指定将完成到该物体的轮廓的操作。这个参数可以具有以下值之一：

- PLAY 启用轮廓动画
- 停止禁用它
- INTENSITY 可以用来设置该物体的轮廓强度

强度 轮廓强度。如果 Operation 参数设置为 INTENSITY 时此参数才是可用的。值可以设置手动或通过变量（如果启用了变量参数）。

设置着色器节点参数

可用于设置着色器节点的值。目前，只有 Value 和 RGB 节点被支持。



输入参数

上一个 前一个节点。

输出参数

下一个 下一个节点。

内部参数

物体 有需要被编辑材质的物体。

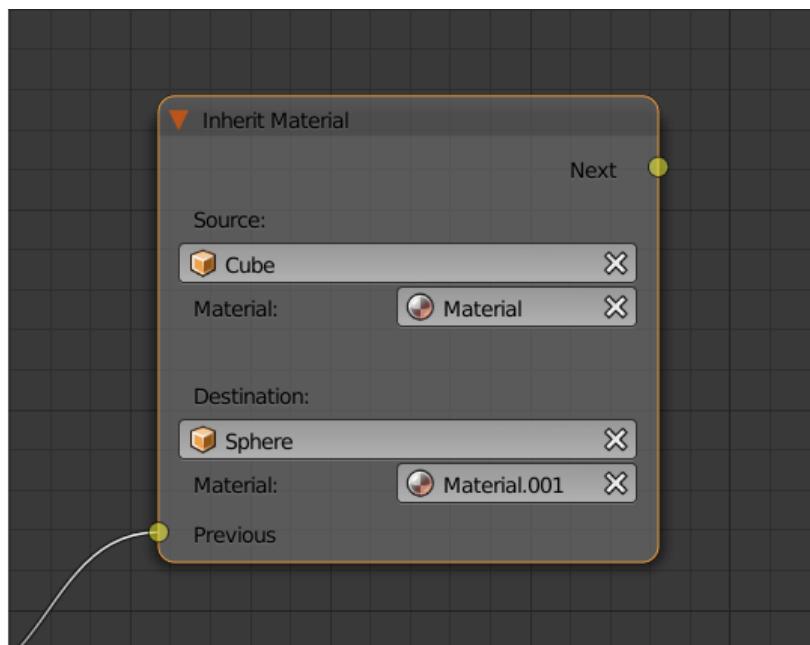
材质 需要材质进行编辑。它应使用的节点。

RGB 节点 可以改变参数的节点。现在，只有 Value 和 RGB 节点被支持。

参数 可编辑参数的选定节点。在设置时可以在节点本身或通过变量（如果启用了变量参数）。

继承材质

从一个材质拷贝属性到另一个。



输入参数

上一个 前一个节点。

输出参数

下一个 下一个节点。

内部参数

源 物体的材质可以被继承。

材质 要继承的材质。

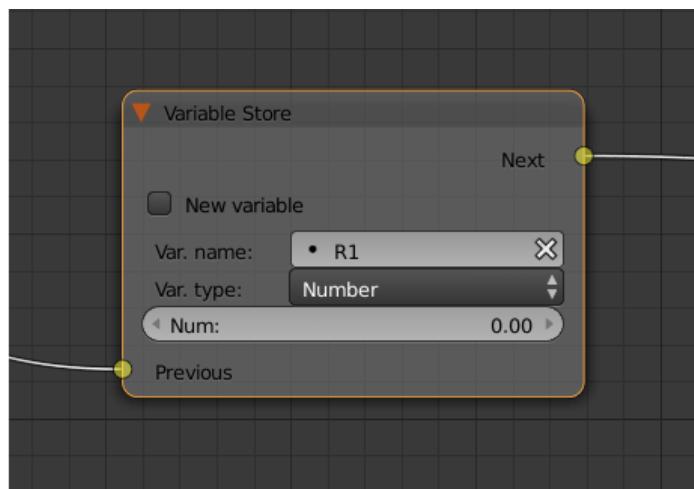
目标 一个物体，将继承选定的材质。

材质 将由继承所替代的材质。

操作

变量存储

保存数字或字符串值的变量。



输入参数

上一个 前一个节点。

输出参数

下一个 下一个节点。

内部参数

变量名称 变量的名称。可以从变量列表中选定或手动指定（如果 New variable 参数已启用）。

变量类型 变量的类型。此参数可以有以下两个值之一：Number（数值变量）和 String（字符串变量）。

新变量 如果启用此参数，则可以手动输入变量名称并不只是选择其中一个变量。这可以用于应用程序和服务器之间传输的数据。

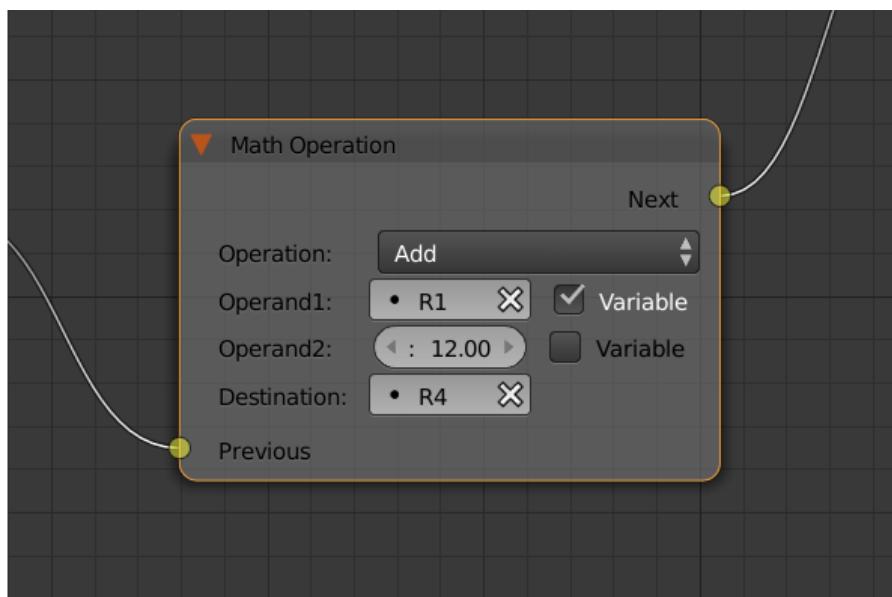
全局 启用此参数可使变量成全局变量。仅当启用了 New Variable 参数可用。



数/STR 数值或字符串（取决于 Var. type 参数值）的变量的值。

数学运算

执行数学运算，并将结果存储在变量中。任何参数（操作数）可以是一个数值或变量。



输入参数

上一个 前一个节点。

输出参数

下一个 下一个节点。

内部参数

操作 数学运算。可以有以下几种类型：

- 随机产生比第一操作数大比第二个小的随机值。
- 添加量操作数相加。
- 乘将操作数相乘。
- 减去从所述第一个数中减去第二个操作数。
- 除数由第二个把第一个操作数除。
- 正弦返回第一个操作数所定义的角度（以弧度为单位来衡量）的正弦值。
- 余弦返回第一个操作数所定义的角度（以弧度为单位来衡量）的余弦值
- 正切返回第一个操作数所定义的角度（以弧度为单位来衡量）的正切值。
- 反正弦返回第一个操作数的反正弦值。
- 反余弦返回第一个操作数的反余弦值。
- 反正切返回第一个操作数的反正切值。
- 对数返回用第二个操作数用作基底的第一个操作数的对数值。
- 最小返回一个较小的两个操作数。
- 最大返回两个操作数的其中大的那个。
- 四舍五入四舍五入第一个操作数。
- 求余返回第一个数被第二个数相除后的余数。
- 绝对值返回第一个操作数的绝对值。

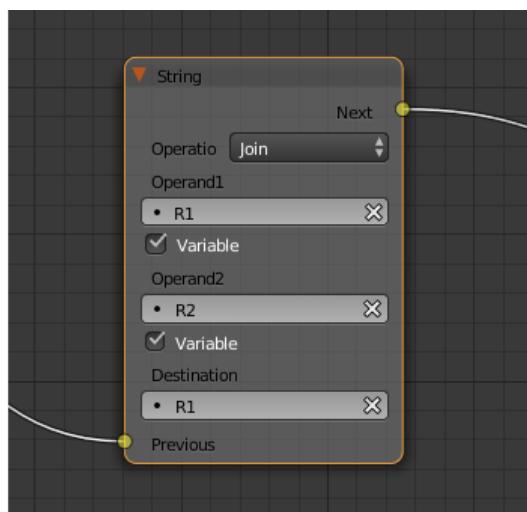
Operand1 对于第一个操作数。它可以在节点中指定或它可以链接到一个变量（如果启用了变量参数）。

Operand2 第二操作数。的工作方式与第一个同。

目标 该运算的结果将保存在此参数所指定的变量中。

字符串操作

可以用来执行一个包含两个字符串的操作，并将结果保存到一个变量。



输入参数

上一个 前一个节点。

输出参数

下一个 下一个节点。

内部参数

操作 您需要有两个字符串，其可具有以下值中的一个来执行操作：

- 加入 - 连接两个字符串合并为一个。

- **查找** - 如果第二个字符串在第一个里面出现就将在第一个里面出现的第二个的索引位置写入变量。如果没有匹配项，则将写入值为-1。应该指出的是，字符串的第一个符号具有索引为 0，不是 1。
- 替换用第三个字符串替换第一个字符串中出现的第二个字符串。
- 拆分用第二个字符串在第一个字符串中出现位置作为拆分标记将第一个字符串拆成两个
- 比较比较两个字符串。对于此操作，您需要指定一个逻辑条件。如果满足此条件，值为 1 将被输出到 Destination 变量，如果不是，零将被输出。

条件 要比较两个字符串的逻辑条件。只有 Operation 参数设置为 Compare，此参数才可用。和 Conditional Jump 节点的 Condition 参数相同的工作。

Operand1 第一个字符串。可以指定在该节点本身或与一个变量。

Operand2 第二个字符串。的工作方式与第一个相同。

操作数 3 只有 Operation 参数设置为 Replace，此参数才可用。可以用来指定第三个字符串，将替换在第一个中出现的第二个字符串。

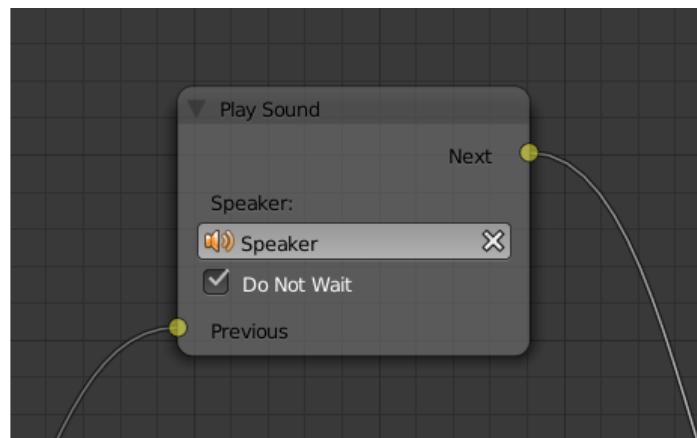
目标 一个变量来保存操作的结果。

Destination2 只有 Operation 参数设置为 Split，此参数才可用。指定的变量来保存第二个拆分（第一半将被保存到 Destination 参数所指定的变量）的字符串。

声音

播放声音

可以用来播放扬声器的声音。



输入参数

上一个 前一个节点。

输出参数

下一个 下一个节点。

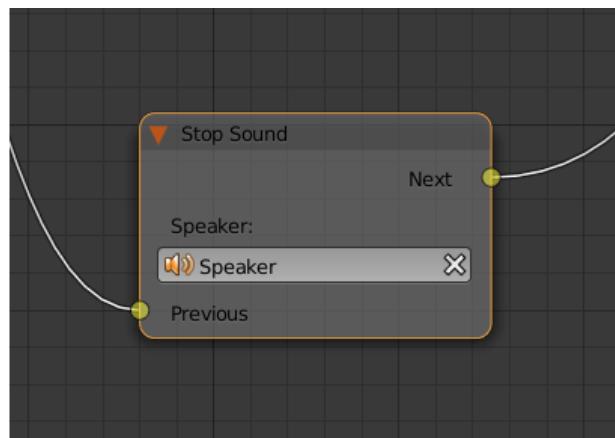
内部参数

扬声器 将启用的扬声器。

不要等 如果启用此参数，控制将声音开始播放后立即传递到下一个节点。如果未启用它，当播放完成控制只会通过。

停止声音

可以用来阻止扬声器的声音。



输入参数

上一个 前一个节点。

输出参数

下一个 下一个节点。

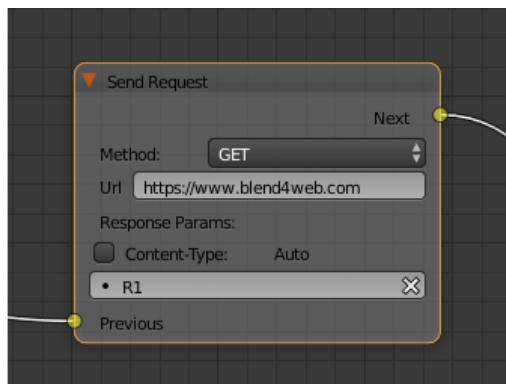
内部参数

扬声器 将被关闭的扬声器。

网络

发送请求

将 HTTP GET 请求发送到指定的 URL 和保存响应字段变量。



输入参数

上一个 前一个节点。

输出参数

下一个 下一个节点。

内部参数

方法 请求方法。可以有两个值之一：

- GET 用来从服务器请求数据。
- POST 被用于将数据发送到服务器。

默认设置为 GET 。

Url 将请求发送到的 web 地址。默认设置为 “<https://www.blend4web.com>” 。

响应参数 指定要保存接收到来自服务器的数据的变量。

注解: 从服务器收到的数据应该是 JSON 格式

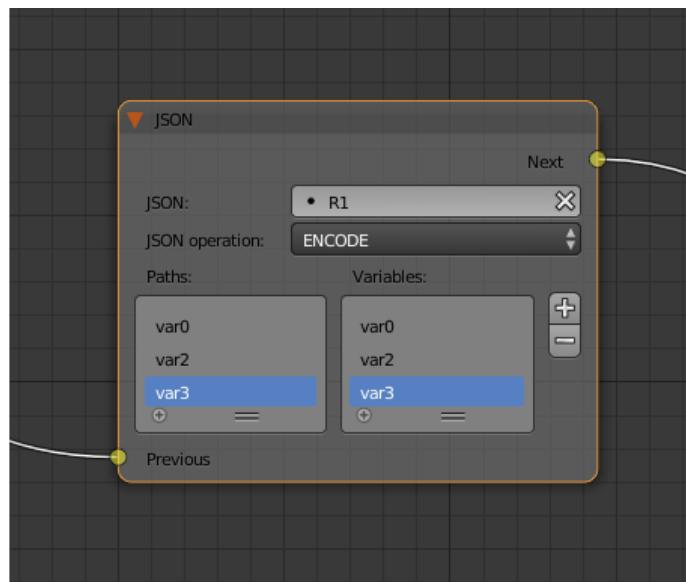
```
{
  "var0": 1,
  "var1": 10,
  "var2": 144
}
```

Content-Type 表示信息内容的媒体类型。一个类型和子类型，例如包括：text/plain。默认设置为 Auto 。可以用来重新分配 HTTP 请求的标题。

Request Params 指定将发送到服务器的包含 JSON 物体的变量。Method 参数设置为 Method 时才可用。默认值为 R1 。

JSON

此节点可以用于编码和解码复杂的 JSON 物体。



输入参数

上一个 前一个节点。

输出参数

下一个 下一个节点。

内部参数

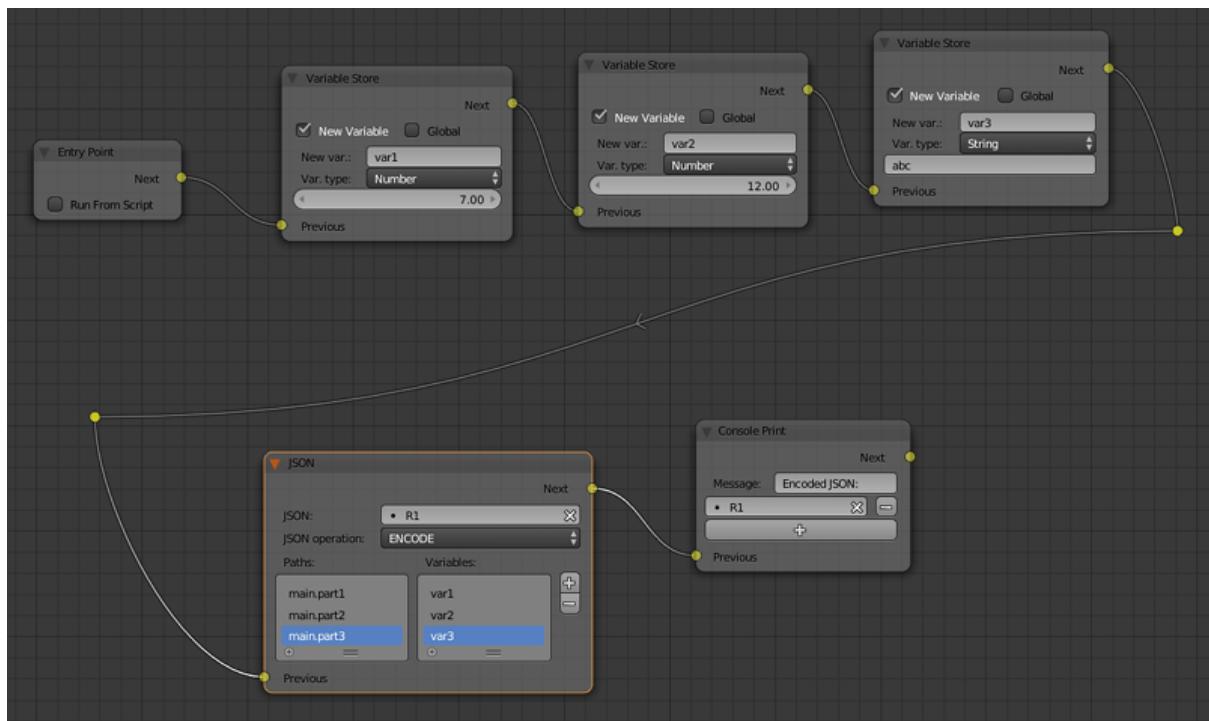
JSON 链接到一个您需要进行解码或编码的 JSON 物体。默认设置为 R1 。

JSON 操作 需要对 JSON 参数所指定的 JSON 物体执行的操作。可以有两个值之一： ENCODE 对 JSON 物体进行编码和 PARSE 进行解码。默认设置为 ENCODE 。

Paths 一个 JSON 物体内部的变量的路径列表。路径是用来定义一个 JSON 物体的内部结构。路径应该包括几个标识符（用逗号分隔），作为一个 JSON 字段路径。如果路径的片段的名称仅由数字组成，则该名称将被解释为数组索引。路径的创建和删除跟变量是同时的（变量列表），和一个路径总是对应于一个变量。这个列表可以用来进行编码和解码的 JSON 物体。默认情况下，列表是空的。

Variables 变量列表，将用于存储数据的解码或编码一个 JSON 物体（取决于 JSON Operation 参数的值）。变量总是有类似 var0, var1 等的名字，他们的数量和名称可以调整。默认列表为空。

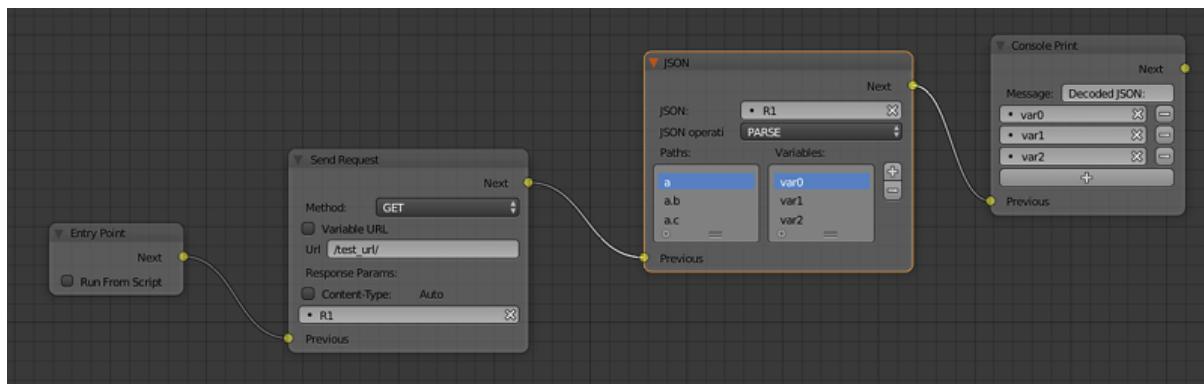
使用 JSON 逻辑节点编码 JSON 物体



在图片的逻辑节点设置在 R1 可变编码以上 JSON 物体，并将其存储。这样的 JSON 物体如下：

```
{
  "main": {
    "part1": 7,
    "part2": 12,
    "part3": "abc"
  }
}
```

使用 JSON 逻辑节点解码 JSON 物体



上图为逻辑节点的设置，从服务器获取 JSON 物体，并将其存储在 R1 变量，然后解码。这样的 JSON 物体如下：

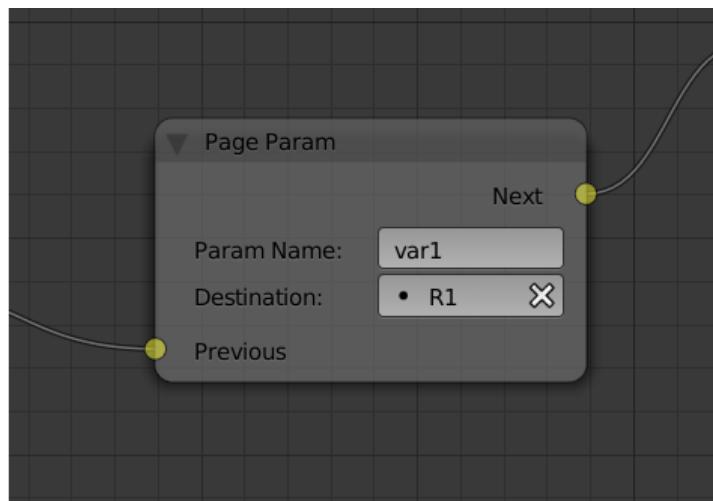
```
{
  "a": {
    "b": 17,
    "c": "abc"
  }
}
```

这解码 JSON 物体的 JSON 逻辑节点 var0, var1 和 var2 三个变量命名的结果（你不必事先创建的变量）将包含 JSON 物体的各个部分。在这个例子中，var1 变量的值是 17，则 var2 变量的值是 “abc”，而 “var0“ 变量包含 JSON 物体的以下片段：

```
{
  "b": 17,
  "c": "abc"
}
```

页面参数

允许在一个给定的变量中存储任何 web 页参数。



输入参数

上一个 前一个节点。

输出参数

下一个 下一个节点。

内部参数

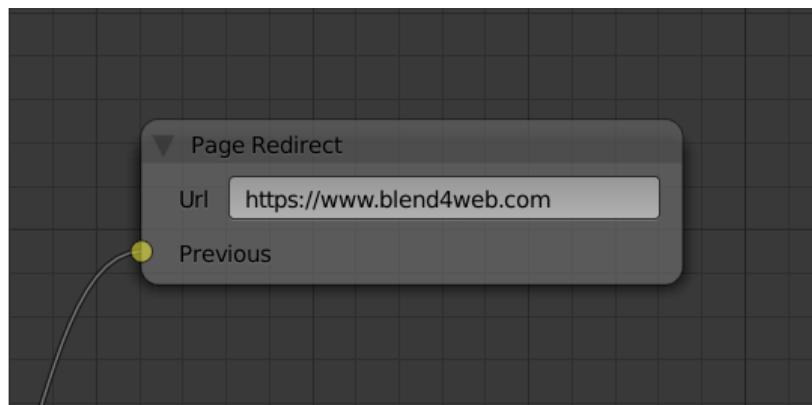
参数名称 网页参数的名称。

If the parameter specified in this field is presented in the URL, then its value will be saved to a variable specified by the Destination parameter.

目标 将用来保存参数的变量。

页转到

可以用来浏览器重定向到另一页。此节点总是标记节点树的结束，并且没有任何输出参数。



输入参数

上一个 前一个节点。

输出参数

空

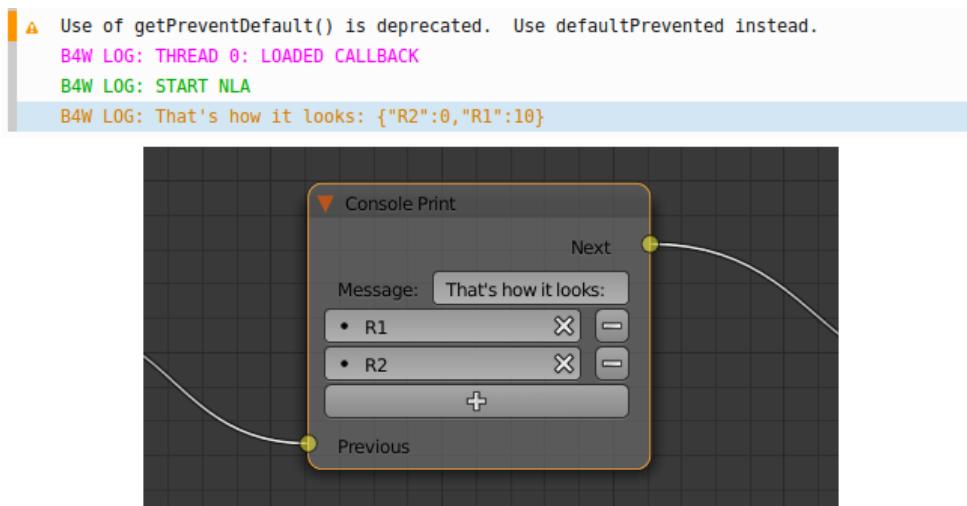
内部参数

Url 将打开的网页的 web 地址。默认设置为 “<https://www.blend4web.com>”。

调试

控制台打印

此节点打印变量的值和附加的文本到 Web 浏览器的控制台。它可用于调试目的。



输入参数

上一个 前一个节点。

输出参数

下一个 下一个节点。

内部参数

信息 将与值一起打印到控制台的消息。

<variable name> 将打印到控制台的变量。默认情况下，Console Print 节点都有一个这样的参数，但您可以添加新成员和删除现有的（节点甚至可能没有这种参数）。

已过时

选择（已过时）

注解： 已过时！不推荐使用。相反，推荐使用 Switch Select 节点。

它类似于 Select & Play 节点，除了过渡发生代替动画。这个功能可以实现一个复杂的逻辑，因为在这种情况下，存在这样的可能性，以确定所述用户选择的结果。

选择并播放时间轴（已过时）

注解: 已过时！不推荐使用。相反，建议使用 Switch Select 和 Play Timeline 节点。

等到用户选择的物体（在桌面上 - 用鼠标点击，移动设备 - 带有触摸）。如果物体，这是在本节点中指定时，选择 - 然后启动类似于 Play Timeline 节点动画。如果选择了任何其他物体 - 然后立即转移控制到下一个节点。

选择并播放动画（已过时）

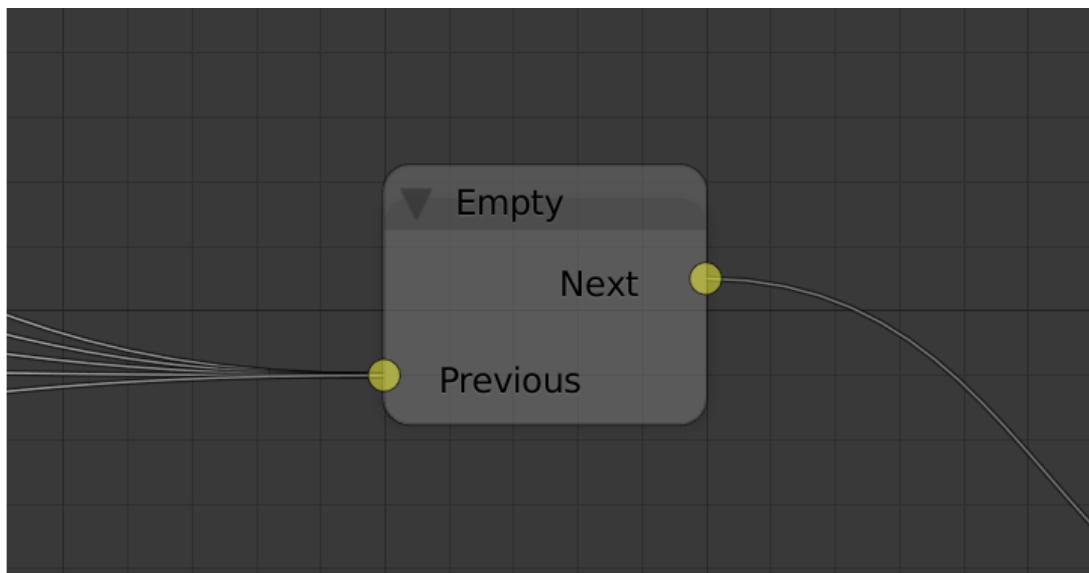
注解: 已过时！不推荐使用。相反，建议使用 Switch Select 和 Play Timeline 节点。

等到用户选择的物体（在桌面上 - 用鼠标点击，移动设备 - 带有触摸）。如果物体，这是在本节点中指定时，选择 - 然后启动类似于 Play Animation 节点动画。如果选择了任何其他物体 - 然后立即转移控制到下一个节点。

布局

空物体

这是一个不自身执行任何操作的简单的直通节点。它可以用于多个逻辑线程组合到一个或简单地使逻辑节点设置更容易阅读和理解。



输入参数

上一个 前一个节点（或多个节点）。

输出参数

下一个 下一个节点。

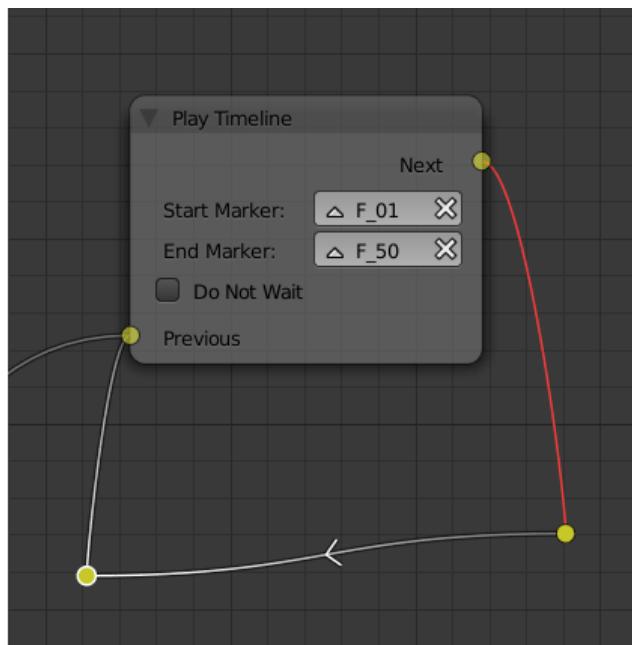
内部参数

空

路由

逻辑编辑器还具有 Reroute 元素，不从传递控制到下一个节点或者到下一个 Reroute 元件做任何处理的节点。这样的元件可以用来创建环状的结构，或使节点树更容易阅读和理解。

不同于 Empty 节点，Reroute 元素只能处理一个单一的逻辑线程。



注解: 输出参数不能被连接到相同的节点的输入参数。如果你需要做到这一点（例如做一个循环），你应该使用 Reroute 元素。

调试中

为了调试的目的，逻辑树中的一些节点可以被静音。要做到这一点，选择所需的节点，并按“M”键。静音节点不执行，并简单地传递控制到后面节点。如果静音节点有两个输出执行从负的结果（Miss, False）继续输出。

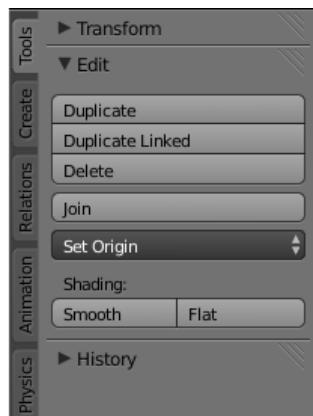
照明、阴影和背景

目录

- 照明、阴影和背景
 - 着色类型
 - 用光源照明
 - * 光源类型
 - * 光源设置
 - 环境照明（环境）
 - * 激活
 - * 设置
 - * 环境贴图方法
 - 阴影
 - * 激活
 - * 设置
 - Background

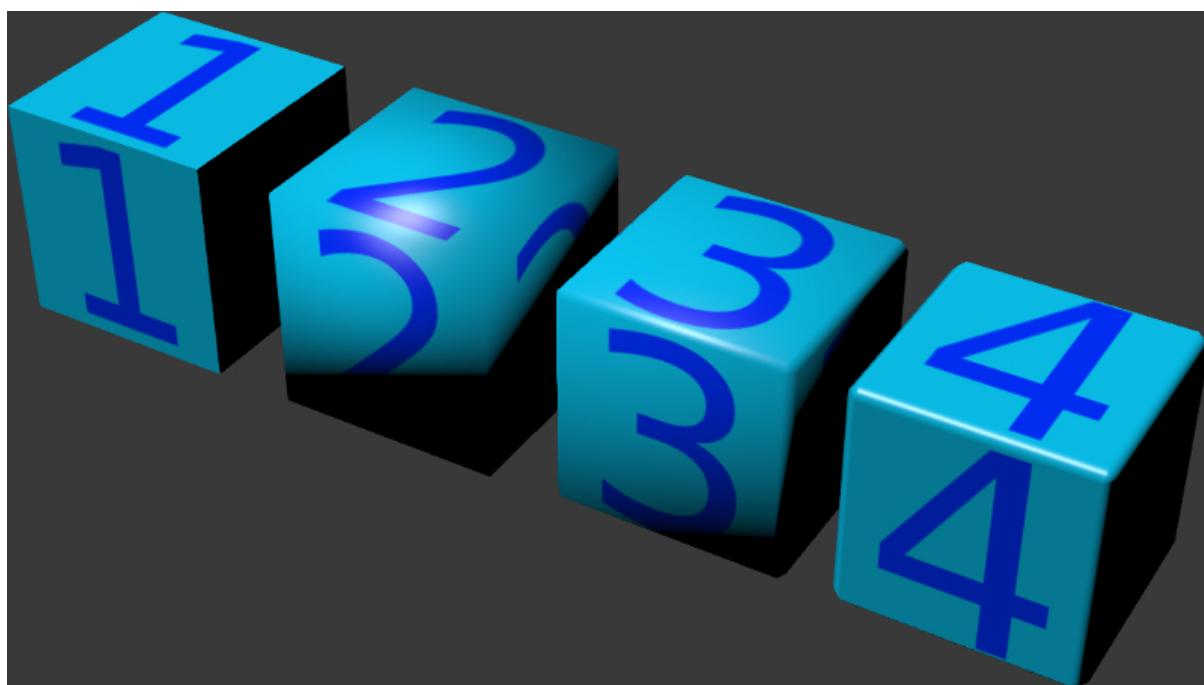
着色类型

照明（着色）取决于法线矢量的方向。支持标准 Blender 的阴影类型：Shading: Flat（面法线被使用），Shading: Smooth（插值顶点法线被使用）以及它们的组合。



如果使用标准工具没法达到所需的效果，您可以使用[normals editor](#).

不同的阴影类型和使用法线编辑的结果：



1. 平面着色
2. 光滑着色
3. 平滑阴影倒角
4. 平滑阴影倒角编辑法线

用光源照明

一个场景可以有多个（但不小于一个）的不同类型的光源。

光源类型

支持下面的光源类型：

Point 光从一个点均匀传播与逐渐衰减四面八方。

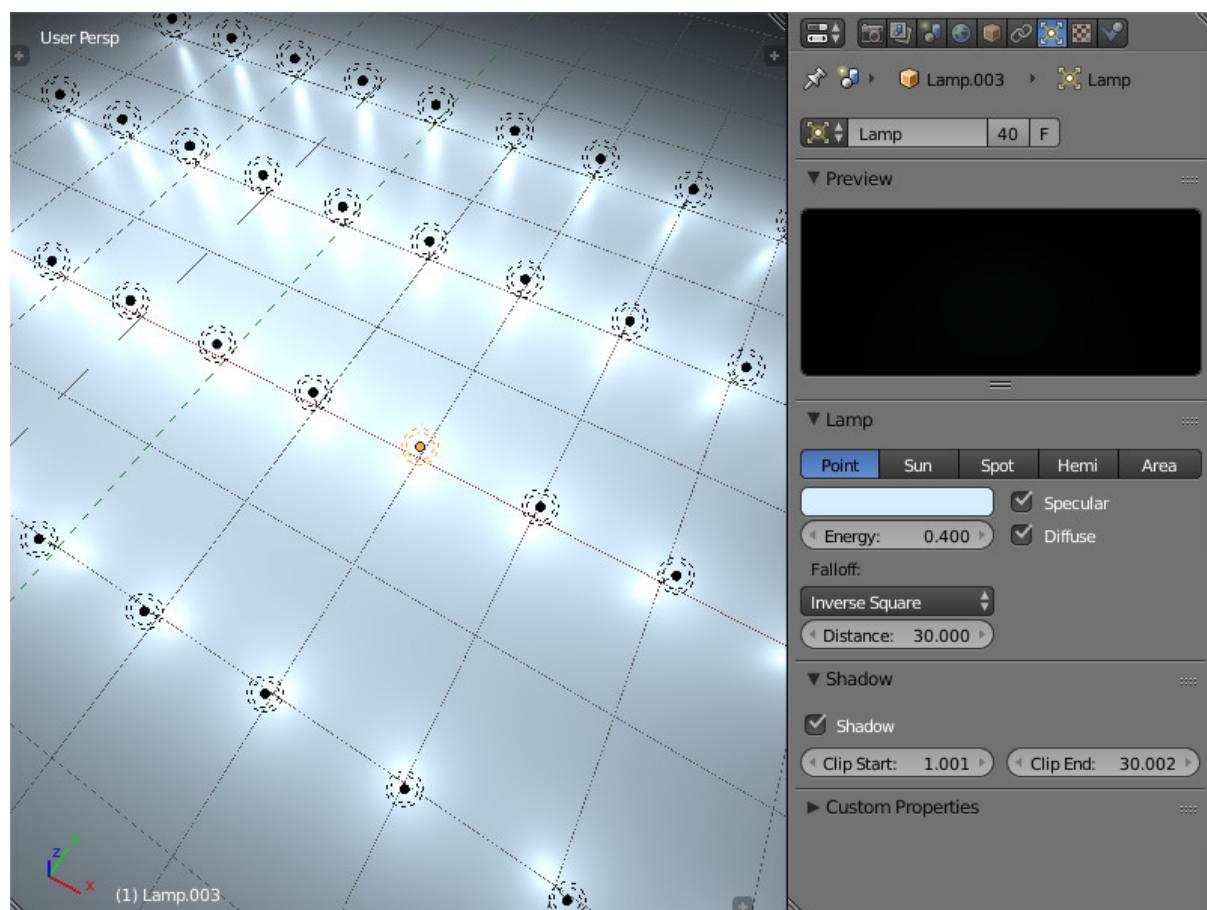
Sun 光从一个无限平面在一个方向无衰减传播。

Spot 光从角度限制内的一个点传播时，并逐渐衰减。

Hemi 半球形。光线从无限半球传播无衰减。

光源设置

当选择了一盏灯在 Object Data 标签中修改设置。



Color 灯光颜色。默认值是 (1.0, 1.0, 1.0) (即白色)。

Energy 辐射强度。缺省值为 1.0。

Falloff 衰减型。该值导出，但引擎始终使用 Inverse Square. 它适用于 Point 和 Spot 光源类型。默认值是 Inverse Square.

Distance 衰减参数。它适用于 Point 和 Spot 光源类型。默认值是 25.0。

Specular 创建镜面高光。默认情况下启用。

Diffuse 做漫反射着色。默认情况下启用。

Spot Shape > Size 锥角度。它适用于 Spot 光源类型。默认值是 45 度。

Spot Shape > Blend 模糊光斑边缘参数。它适用于 Spot 光源类型。默认值是 0.15。

Dynamic Intensity 使用该光源，用于计算每天的时间。仅适用于 Sun 光源类型。默认情况下禁用。

Shadow > Shadow 使用这个光源计算影子。当多个光源存在应该被使用。默认情况下禁用。

Shadow > Clip Start 此参数指定离光源的距离，低于该值物体不产生阴影的距离。默认值为 1.001。

Shadow > Clip End 此参数指定离光源的距离，超出后物体不产生阴影的距离。默认值是 30.002。

环境照明（环境）

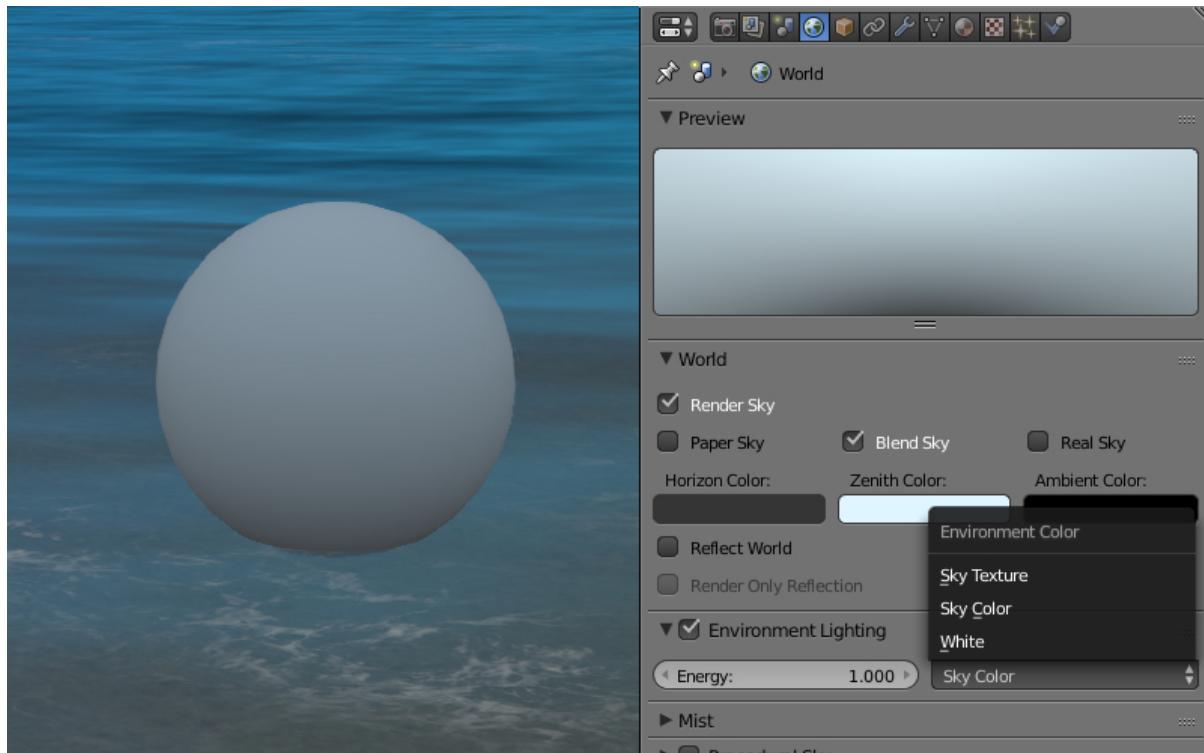
该引擎支持环境灯光模拟的 3 种方法。

1. 平白光照明。
2. 半球形的照明模式，即应当指定视野和天顶的颜色。物体会因为法线和方向填充这两种颜色组成的梯度色。
3. 灯光使用 [环境贴图](#) - 所谓的基于图像的照明。

Please note that environment lighting uses a simplified model which doesn't take into account mutual shadowing of objects.

激活

启用 World 选项卡上的 Environment Lighting 复选框。



设置

World > Environment Lighting > Energy 环境照明显亮度。缺省值为 1.0。

World > Environment Lighting > Environment Color 选择环境照明模拟的方法:

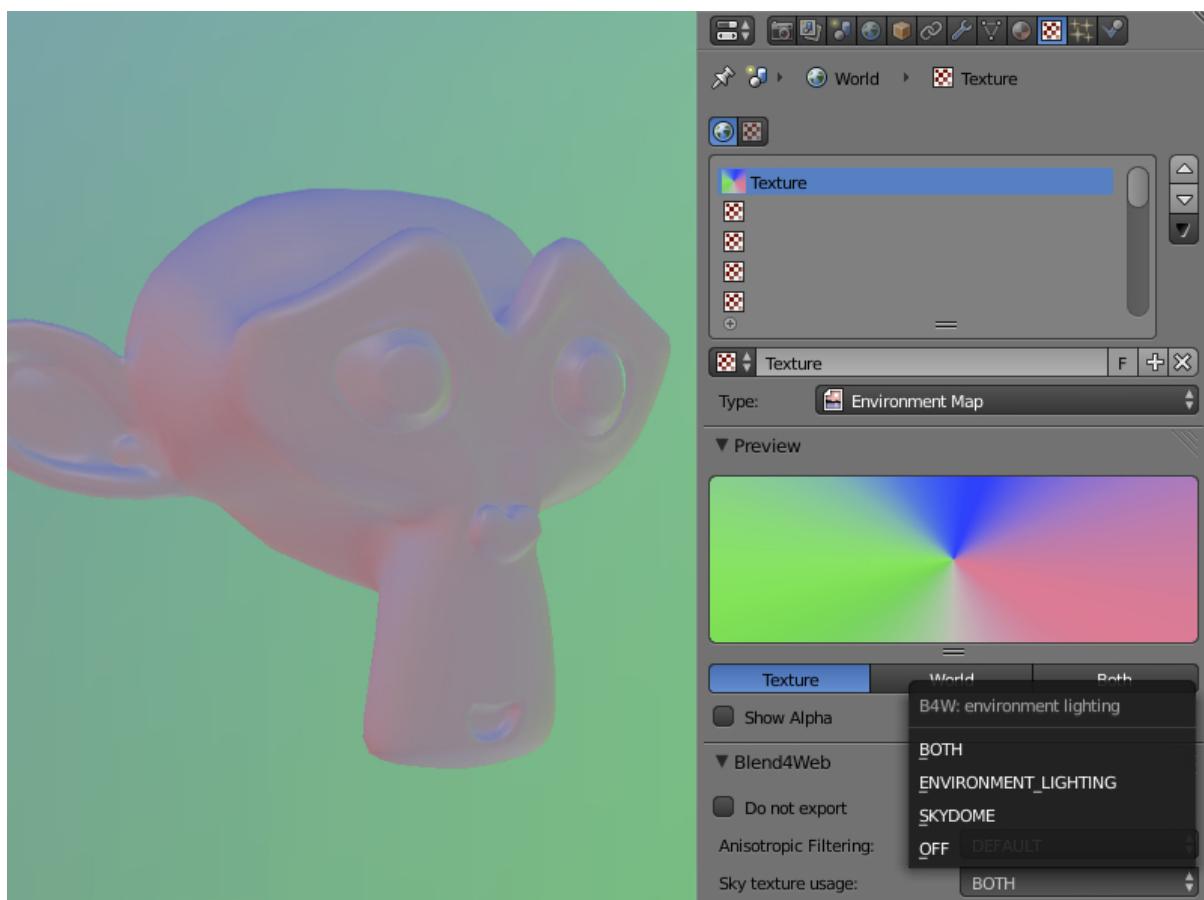
White - 平照明, Sky Color - 半球模型, Sky Texture - 照明使用的[环境贴图](#)。默认值是白色。

World > Horizon Color 和 World > Zenith Color 如果选择了半球模型 (Sky Color) 可以通过 World > Horizon Color 和 World > Zenith Color 颜色选取器指定的地平线和天顶角变化的颜色。建议以激活 World > Blend Sky 选项为更好的颜色选择。

环境贴图方法

要使用环境贴图进行环境照明：

1. 启用 World 选项卡上的 Environment Lighting 复选框。
2. 选择 环境照明 > 天空 Texture。
3. 从 World 选项卡转至 Texture 标签。
4. 创建环境贴图，加载相应的图像。
5. 对于环境贴图选择 ENVIRONMENT_LIGHTING 或 BOTH 为 Export Options 面板上的 Sky Texture Usage 值 (BOTH 选项，还可以使用该纹理作为天穹材质).

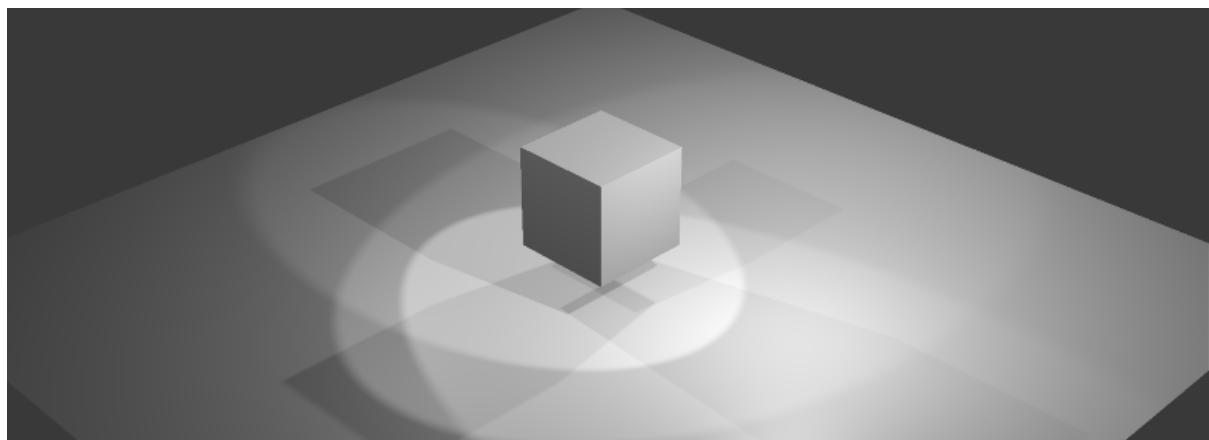


阴影

阴影对渲染最后的画面格外重要。它们所提供的观看者不仅与有关物体的轮廓的信息，也包含他们的高度和相对位置，光源位置等。

Blend4Web 实现以下阴影渲染技术：级联阴影贴图（CSM）和软阴影（PCF）。

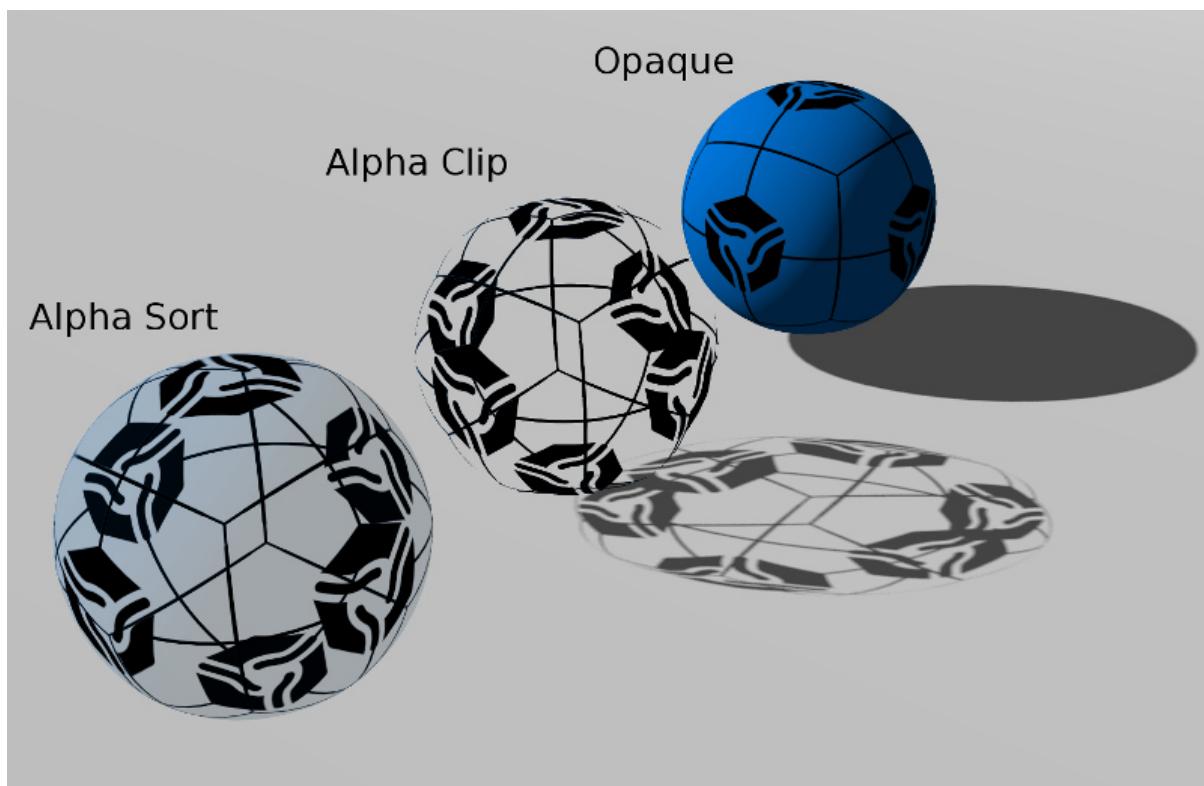
多达 4 个（或 3 如果 SSAO 启用）光源可以同时生成阴影。如果 阴影参数为被超过 4 光源启用，阴影只能从其中 4 个中生成的。



激活

1. 在 Object 选项卡下启用 Shadows: Cast 复选框为物体 **投射** 阴影。
2. 在 Object 选项卡下启用 Shadows: Receive 复选框为物体 **接收** 阴影。
3. 确保在 Render 选项卡中的 Shadows 选项的值为 AUTO 或 ON。

注解： 物体，其中带有渐变的透明材质，不投射阴影。

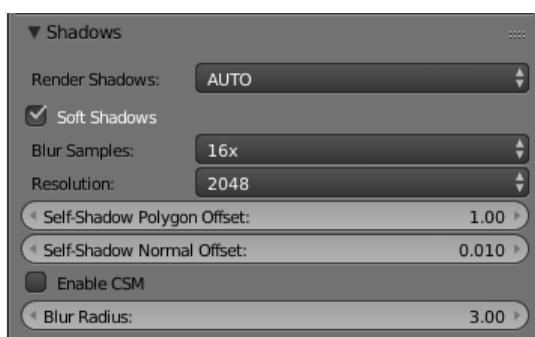


设置

Direction 如果有多个光源，建议指定用于阴影计算确切的光源，选择灯光物体通过在 Object Data 面板的 Shadow > Shadow 复选框来启用。

Color 环境照明阴影颜色是由[环境光照](#) 设置确定。

下列附加设置位于 Render 标签的 Shadows 面板上：



渲染阴影 启用和禁用阴影渲染。可设置 ON, OFF 和 AUTO。默认设置为 AUTO。

软阴影 此选项使平滑的阴影贴图。默认情况下启用。

分辨率 阴影贴图分辨率。默认值为 2048 x 2048px。

采样 用于平滑阴影贴图的样品数目。可用值为 4x, 8x 和 16x, 后者是默认值。

Self-Shadow Polygon Offset 相对于光源定向多边形移位系数。默认值是 1。

Self-Shadow Normal Offset 沿法线移动多边形系数。默认值是 0.010。

最后两个参数可以用于减少自阴影的瑕疵。这些瑕疵显示在同一时间投射和接收阴影的物体。Self-Shadow Polygon Offset 参数是更有效的对抗的多边形的内部区域瑕疵，虽然‘Self-Shadow 法线抵消’是更好地为边界地区。上述两个参数导致阴影扭曲，因此，我们建议让他们尽可能低的水平。



注解: 从 Point 光源生成的阴影方法与从 Spot 光源的相同，预计由源的 Rotation 参数指定只在一个方向投影。

Enable CSM 激活的级联阴影模型的使用；显示附加选项。默认情况下禁用。如果 Shadow 设置一个以上的光源启用将无法工作。Point 和 Spot 式光源只支持一个阴影级联。

此选项允许下面的阴影生成类型之间进行选择：

- 它使用一个单一的优化阴影贴图为整个场景通用模型 (Enable CSM 已关闭)。
- 阴影级联 (Enable CSM 被接通)。

Blur Radius 设置软化阴影模糊比例。默认值是 3。零值会产生硬阴影。



软化的阴影可以提高视觉质量和真实感。使用基于图像的技术，这是尤其明显的低分辨率的阴影贴图时，他们隐藏不可避免的齿形的边缘。经常使用软化阴影允许降低分辨率没有实质的质量损失。

通用阴影

此选项适合很小场景与数量有限的物体。通过优化应用这样的场景，可以用级联阴影达到更好的阴影质量。此外，此选项是简单和快速的设置，而使用单个阴影贴图极大地提高了性能。

级联阴影

注解：这些设置只支持 Sun 光源。级联对其他类型的光源关闭。

为提供可接受的阴影质量，同时也为覆盖了相当大的空间是需要影子生成（级联）使用多个阶段。因此，最好的质量级联位于靠近观察者，而最坏的质量级联的距离较远。这个选项适合中到大的场景，例如游戏关卡。

如果启用下列扩展设置会显示：



CSM Number 阴影级联的数量。从 1 至 4 个级联的支持。默认值是 1。

CSM First Cascade Border 第一梯级的大小。默认值是 10.0。

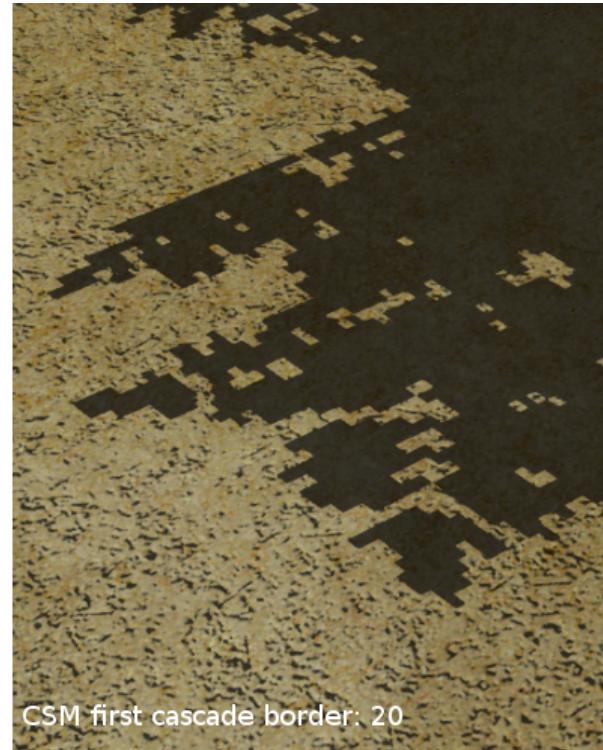
CSM Last Cascade Border 最后级联大小。默认值是 100.0。

中间级联的尺寸在两个上述参数之间内插。

注解: 记住当设置阴影时, 级联尺寸越大, 内部的阴影细节越少越差。另一方面, 降低 CSM First Cascade Border 参数使得靠近相机的级联缺乏细节。降低 CSM Last Cascade Border 参数强制在靠近相机的距离里阴影消失。然而, 当使用软阴影时, 由于边缘模糊的原因整体质量将得到改善。



CSM first cascade border: 7



CSM first cascade border: 20

CSM First Cascade Blur Radius 模糊比用于第一级联。默认值是 3。零值会产生硬阴影。

CSM Last Cascade Blur Radius 模糊比用于最后级联。默认值是 1.5。零值会产生硬阴影。

中间级联的模糊半径两个上述参数之间内插。

注解: 我们建议开始设置与第一梯级软化阴影 (使用 CSM First Cascade Blur Radius), 然后进行其他级联 (使用 CSM Last Cascade Blur Radius)。通常, 最后一个级联可能比第一个需要更少的模糊。这可能是必要的, 由于低分辨率以防止在最后一个级联的

阴影太褪色，。这也减少了不良自阴影假象。

Fade-out Last Cascade 平滑消失在最后一个级联。默认情况下启用。

Blend Between Cascades 平滑的级联之间的界限。默认情况下启用。

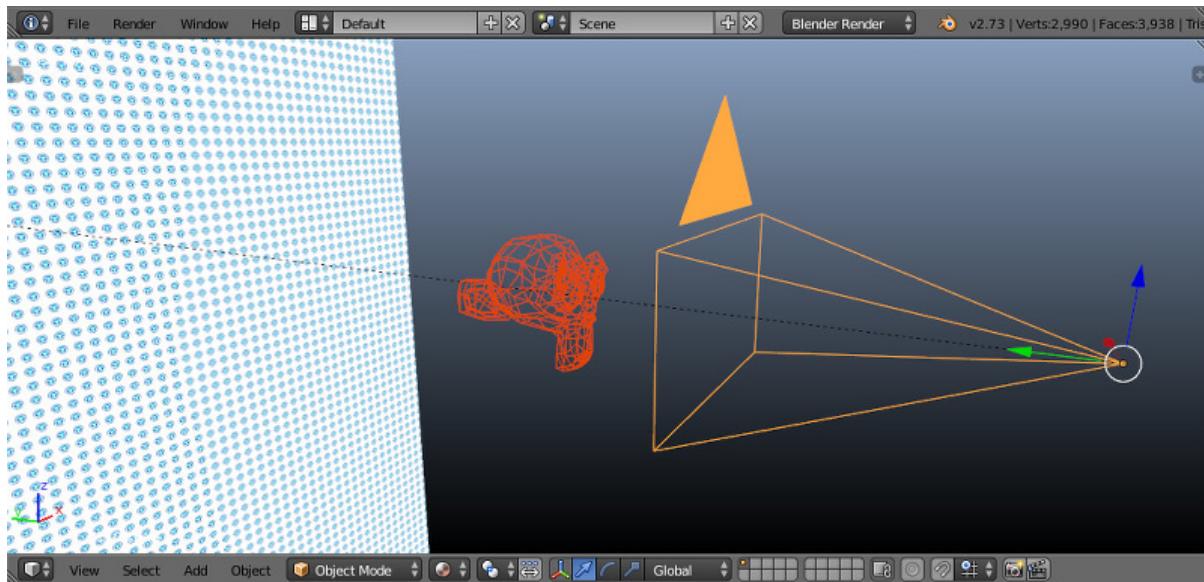


Background

您可以通过以下方式更改背景：

1. 启用 World > Render Sky, 然后在 Blender 的 World 选项卡下设置 Horizon Color 和 Zenith Color 。
2. 把整个场景放入一个（例如一个立方体或球体）法线指向内部模型的里面，带材料和一个可选的纹理。
3. 在相机前放置一个带一个可选的纹理的材质的面。设置父级为相机。如果需要，调

整到这个表面的距离，相机开始和结束剪切平面。



4. 使用skydome.
5. 建立程序产生大气.
6. 用 config.set() 方法设置引擎的 background_color 参数。请注意，Blender 的 World 选项卡下的 World > Render Sky 必须被禁止。这个值被用作 WebGL 的 clearColor() 方法的参数。对于正确的结果，建议将 WebGL 的背景透明度关闭 (alpha 参数)。这样的配置在默认情况下引擎的标准网络播放器使用。

```

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. 您可以在 canvas 元素之后使用任何 HTML 内容，作为背景进行渲染。要做到这一点，激活 WebGL 的背景透明度 (alpha 参数)。请注意，Blender 的 World 选项卡下的 World > Render Sky 必须禁止。想要正确的结果，建议设置完全透明的黑色背景色。这样的配置在 Blend4Web SDK 的 [场景查看器](#) 默认情况下使用。

```

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(...);
```

参见:

[alpha 合成](#)

后期处理效果

目录

- 后期处理效果
 - 运动模糊
 - 景深
 - 屏幕空间环境光遮蔽
 - 上帝光芒
 - 泛光效果
 - 物体外轮廓
 - 晕光
 - 抗锯齿

运动模糊

运动模糊效果可以用来改善交互式场景的真实感。当相机或物体移动，显示为画面模糊。



激活

激活 Render 选项卡上的 Motion Blur 面板。

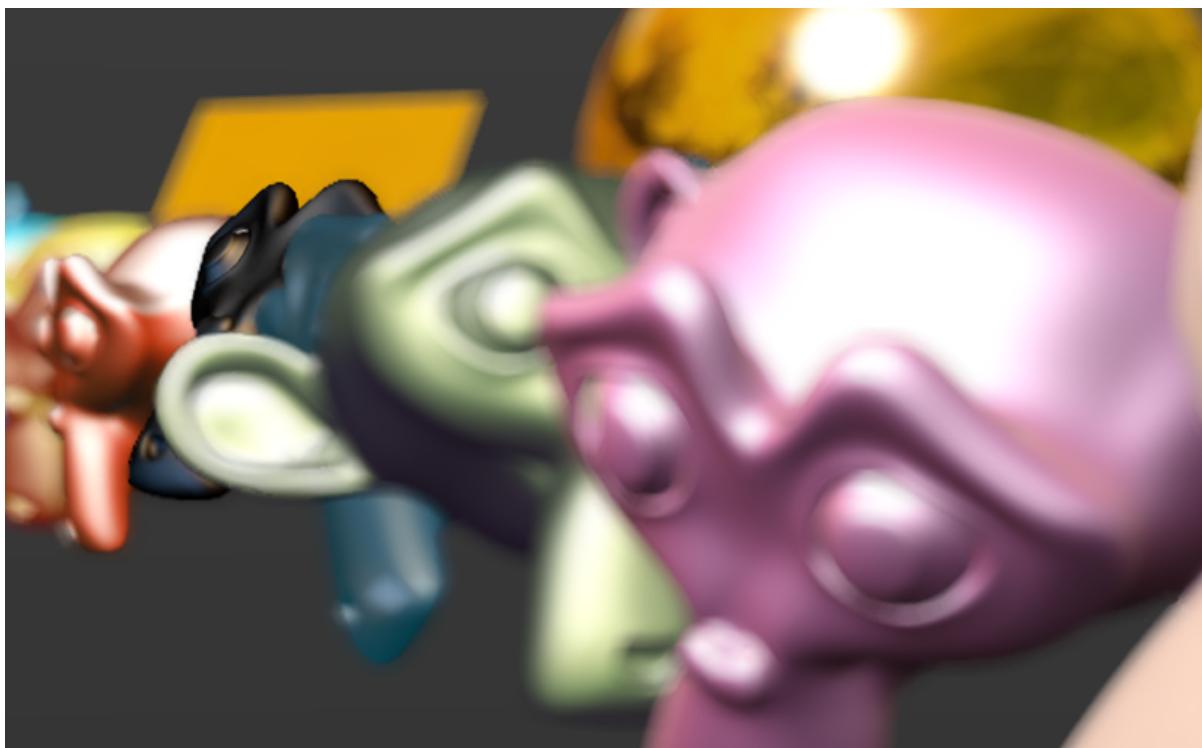
其他设置

因子 外观的影响比例。此值越高越强的运动模糊。

衰减阈值 模糊淡出率。较高的这个值有更鲜明的效果。默认值是 0.01。

景深

场景深度效应 (DOF) 可以用于强调场景的一部分。从相机对焦点之外的地方开始图像模糊。



激活

1. 选择一个主控摄像头，并转到它的设置面板 Camera (Object Data).
2. 然后有两种选择：
 - 在 Depth of Field 面板的 Focus 选项选择一个物体作为相机的对焦点。在这种情况下移动或接近这个物体，会导致相机的焦点相应的修正。
 - 同一面板上为 Distance 设置一个非零值 (Blender 中单位 = 米)。在这种情况下，相机焦点在相机的这个距离值，并与相机一起移动。

附加设置



焦点 设置的重点对象。默认情况下，此字段为空。

距离 此参数定义焦距。仅当 Focus 参数没有预先设置（如果相应的字段为空）有用。默认设置为零。

前端起始 此选项设置（以米为单位）焦点到最近的平面到模糊效应开始出现前的距离（相对于相机）。默认值是 1.0。

只有启用了 High Quality (Bokeh) 参数，此参数才可用。

前端结束 这个设置从相机和最近平面（相对于相机）背后的模糊效果达到最大强度（由 Power 的值定义）的距离。默认值为 5.0。

后方开始 这设置从焦点到最远的平面（相对于相机）的距离，模糊效果开始发生。默认值为 1.0.

只有启用了 High Quality (Bokeh) 参数，此参数才可用。

后方结束 这设置从相机到最远的平面（相对于相机）的距离，模糊达到最大强度（由 Power 的值定义）。默认值为 5.0。

指数 模糊的比率。此参数的值可从 0.1 至 10。默认值为 2.0。

高品质（背景虚化） 这使得高质量渲染景深效果。激活此选项

- 启用的散景效果，并使 Bokeh Intensity 参数可用来调整影响强度。
- 启用 Front Start 和 Rear Start 参数设置从相机里的影响开始发生软过渡之间的距离，它达到其全部的力量的距离。
- 启用 Foreground Blur 的参数，如下所述。

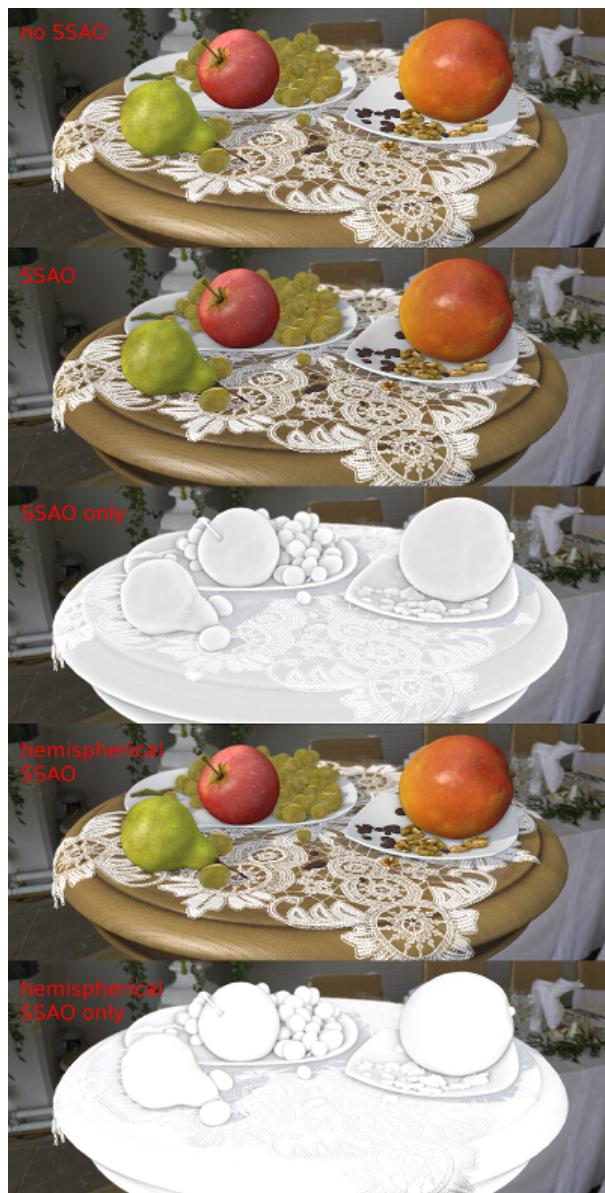
参数默认被禁用的。

散景强度 此值设置的散景效果的强度。它可以从 0 到 1.0 会发生变化。默认设置为 0.3。

前景模糊 启用此选项使引擎模糊前景物体的轮廓，提高质量的代价是性能略有降低作用。默认为禁用。

屏幕空间环境光遮蔽

屏幕空间环境光遮蔽 (SSAO) 效果，可用于仿造在物体周围复杂的光线反射。这种效果的基础是，近处的物体之间的空间为漫射光不易进入的，因而是更暗。



激活

激活 Render 标签下 Ambient Occlusion SSAO 面板，在 Render > Shadows 面板设置 Render Shadows 到参数 AUTO 或 ON 。

其他设置

半径增加 从内部采样环到外部采样环的球形采样半径的乘法系数。默认值是 3.0。

使用半球 使用半球形采样着色，而不是球形。它使用不同的着色方法。

使用模糊深度测试 使用边缘保持模糊到 SSAO 如果标志将被打开。否则，它采用模糊

平均为每个像素周围的 4x4 的矩形。

模糊深度测试丢弃值 对模糊量的采样之间的深度差异的影响。它采用 Use Blur Depth Test 启动标志。缺省值为 1.0。

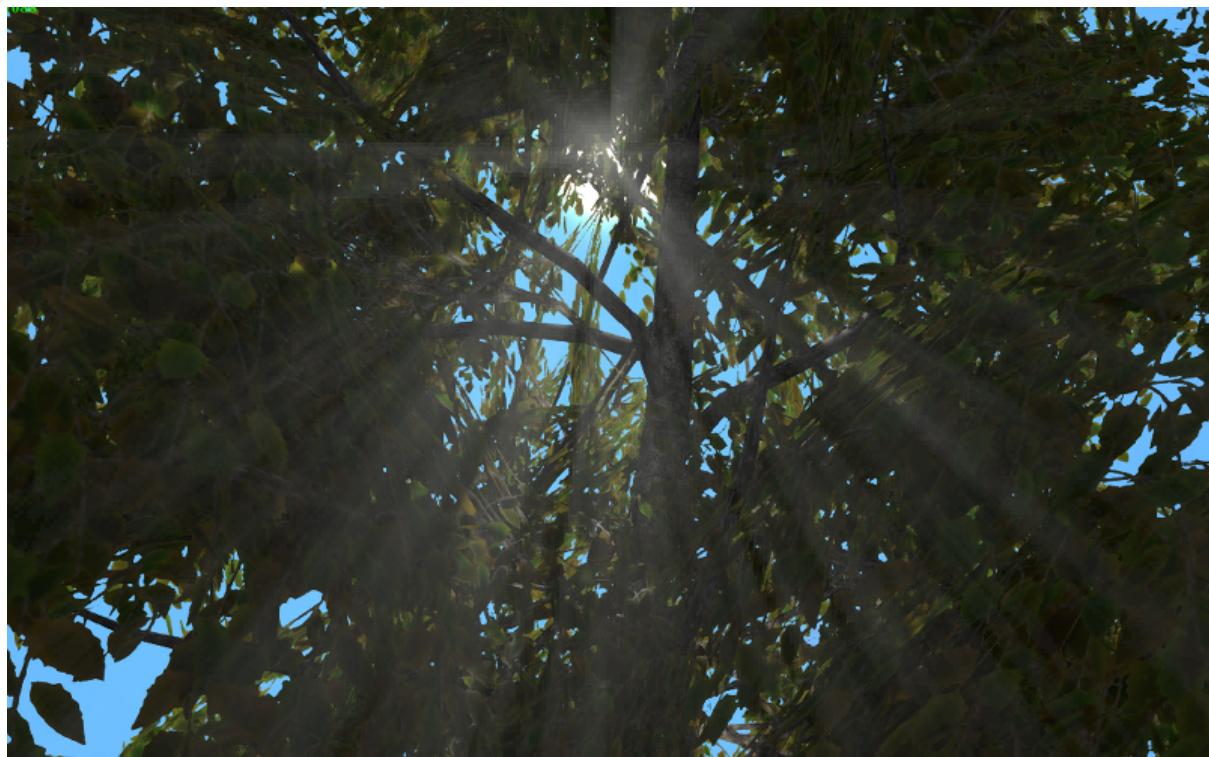
影响 SSAO 外观系数。默认值是 0.7。

距离系数 SSAO 衰减的距离系数。默认值是 0.0 (即没有衰减)。

采样 采样数 (越多采样有更好的质量, 但较差是性能)。默认值是 16。

上帝光芒

神光芒的效果 (又名曙暮辉) 模拟著名的自然现象 - 空气照明部分的闪耀。



激活

激活 Render 选项卡下的 God Rays 面板。

其他设置

强度 影响外观的系数。默认值是 0.7。

最大长度光线 光线长度的系数。限定径向模糊的样本之间的步长。缺省值为 1.0。

每通道步值 每个单个样本步长。默认值是 10.0。

泛光效果

泛光出现在画面中有一个非常不同的亮度元素。生成一个发光的光环围绕着明亮的细节。



激活

激活 Render 选项卡下的 Bloom 面板。

其他设置

Use Adaptive Use calculation of adaptive average luminance. Enabled by default.

强度 Bloom intensity. Set to 1.0 by default.

模糊 Bloom blurriness factor. Set to 4.0 by default.

边缘亮度 The boundary value of an element's relative brightness above which the bloom effect appears. Set to 1.0 by default.

物体外轮廓

作为轮廓辉光效应，围绕物体显示发光的光晕。



激活

外轮廓是通过 API 编程激活。不同的动画模型可以应用，如不断晕光，淡出的光晕，光晕脉动和任何其他。为了使某一物体上勾勒的效果，确保 Render > Object Outlining 面板的 Enable 属性设置为 ON 或 AUTO 。

其他设置

在 Object > Selection and Outlining 面板：

启用外轮廓 允许使用此物体的轮廓发光效果。

持续 辉光动画的持续时间，单位为秒。默认值是 1。

周期 辉光动画的重复周期，单位为秒。默认值是 1。

重复 辉光动画迭代次数。如果为零，动画将永远重复。缺省值是 0。

选择时外轮廓 在选择物体时激活辉光动画。在这种情况下，Selectable 选项必须启用。

在用户定义的辉光动画模型的情况下，以避免冲突此选项必须被禁用。

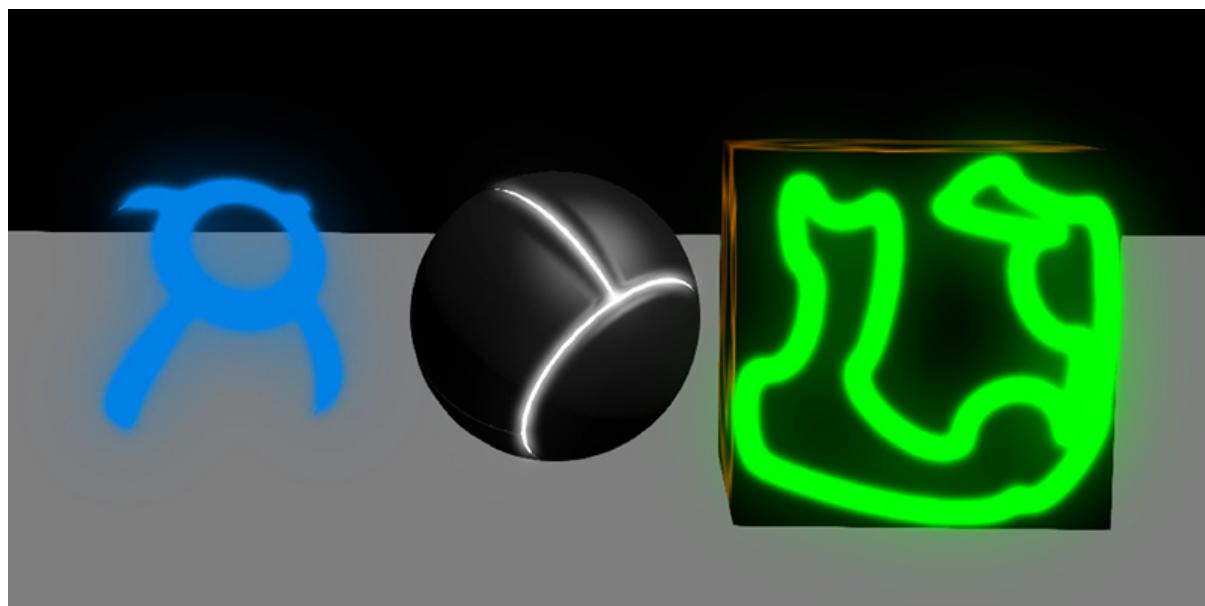
在 Render > Object Outlining 面板：

因子 当这个参数减小围绕物体的辉光的厚度亮度也减小。默认值是 1。

当通过 API 初始化辉光效果时 Render > Object Outlining 被作为默认值。

晕光

晕光的效果是通过观测发射光的物体在大气中的光散射进入人眼内后的成像。



激活

添加B4W_GLOW_OUTPUT 节点到节点材质。在 Render > Glow Materials 面板的 Enable Glow Materials 选项应设置为 ON 或 AUTO 。

其他设置

小遮罩：强度 通过较小的遮罩晕光获得的强度。默认值是 2.0。

小遮罩：宽 通过较小的遮罩晕光获得的宽度。默认值是 2.0。

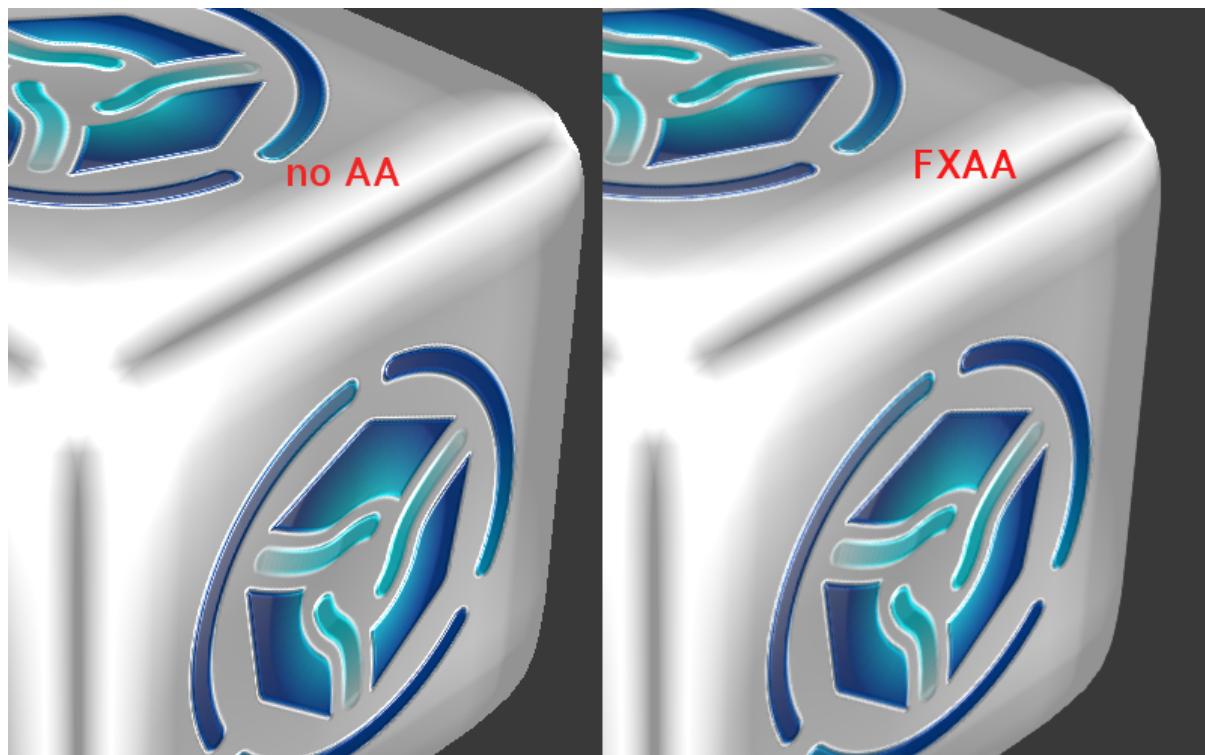
大遮罩：强度 通过较大遮罩晕光获得的强度。默认值是 2.0。

大遮罩：宽 通过较大遮罩晕光获得的宽度。默认值是 6.0。

在透明的物体渲染辉光 渲染透明物体的辉光效果。

抗锯齿

抗锯齿用于减少不良渲染表现（像素化差）。



激活

使用位于 Render > Anti-Aliasing 面板上的 AA Quality 菜单中选择质量配置文件。

- 无 - 关闭抗锯齿，
- 低, 中, 高 - 用给定质量配置文件启用抗锯齿。

中配置文件默认情况下使用。

其他设置

抗锯齿方法是在引擎性能配置文件选择时同时匹配的：

- 低质量 - 无抗锯齿，
- 高品质 * 和 * 超高品质 - 使用 FXAA 3.11 算法 (Nvidia 公司的快速近似抗锯齿)。

立体渲染

目录

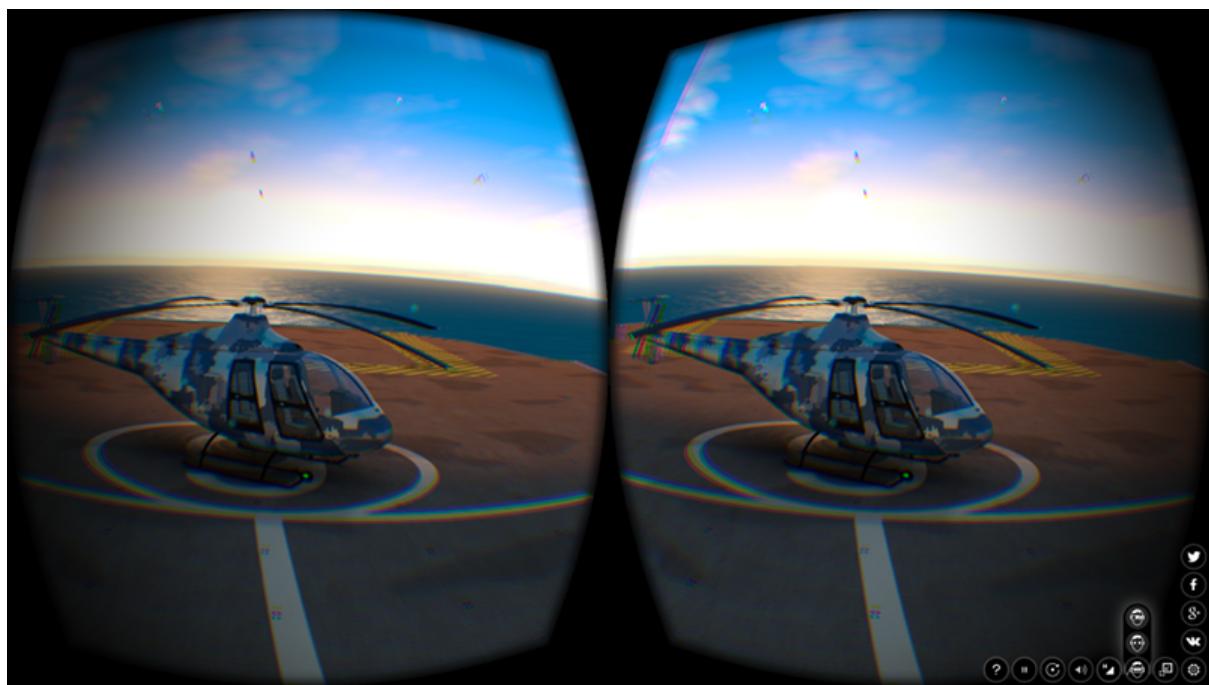
- 立体渲染
 - 激活
 - 附加设置
 - HMD 设置
 - * 配置
 - * 渲染设置
 - * 其他控制工具

立体渲染模式适用于使用特殊眼镜观看的内容。它是应用程序通过 API 来激活。

Blend4Web 支持立体图像渲染的两种技术 - 立体图像（红/蓝）和 HMD（头戴式显示器）。立体图（红/蓝）



HMD:



HMD 是一个实验性的功能，现在它只能用于 Eye 类型摄像机。

激活

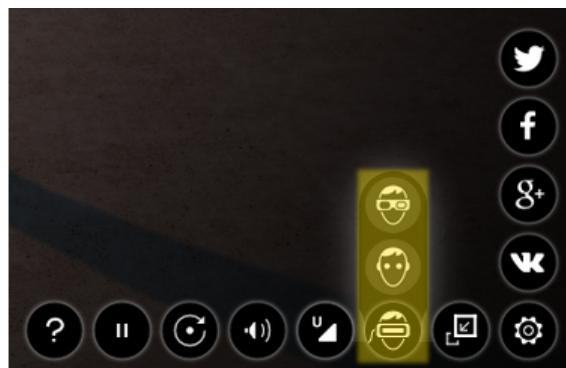
要使用 HMD 立体渲染，您需要安装 Oculus 的 [运行时工具](#)（支持两个版本：根目录下的 `chromium_webvr_v1_win64.7z` 和在 `Deprecated API` 文件夹下的）。Windows 和 MacOS 的二进制格式版本，可以在 Oculus 公司的网站上找到，而 Linux 版本应该从源代码进行编译。

就目前而言，HMD 的支持在 [Chromium 实验性版本](#) 和 [Firefox 每日版](#)。

[Chromium 设置](#).

[Firefox 设置](#).

要打开立体渲染，你需要选择某些特定选项中的设置，从右边的起第三栏，如图所示。



要让立体渲染正常工作，建议切换到全屏模式。

附加设置

无

HMD 设置



这组设置允许用户改变各种参数，以调整任何虚拟现实设备，让他们更好地查看某个三维应用程序或场景。有两种方法可以访问这些设置：

首先，它可以在Blend4Web 查看器 应用的立体视觉面板 下找到。

其次，它也可以在使用的应用程序中通过 `hmd_conf` 模块的 `show` 方法显示，但这样做需要一些编程。该方法的实例可在 `viewer.js` 和 `webplayer.js` 应用程序（在 `m_hmd_conf` 对象）中发现。

配置

这允许用户选择用于特定装置的配置文件。目前，以下模式可供选择：

- 自定义
- 纸板（2014 年）

- 纸板 (2015 年)

默认情况下，Custom 属性文件被选中。

渲染设置

托盘镜片中心的距离 此参数指定手机屏幕和摄像机透镜的中心之间的距离。

这个值可以从 0 到 50 变化， 默认值是 32。

瞳距 此参数指定用户的眼睛的瞳孔之间的距离。测量单位为毫米。

这个值可以从 0 到 100 变化， 默认值是 64。

屏幕到镜片距离 这指定了电话的屏幕和照相机透镜之间的距离。

这个值可以从 0 到 100 变化， 默认值是 50。

屏幕高度 屏幕的高度。

这个值可以从 0 到 150 变化， 默认值是 63。

屏幕宽度 屏幕的宽度。

这个值可以从 0 到 200 变化。缺省值是 107。

斜面宽 图像周围的边框的厚度。

这个值可以从 0 到 20 变化。缺省值是 3。

失真系数 这样做是为了用于补偿由 VR 设备透镜产生的图像失真。

这两个值可以从 0 到 1 变化， 默认值是 0.34 用于第一个参数和 0.55 用于第二个。

其他控制工具

保存 保存的设置。

取消 关闭设置界面，不保存设置。

恢复 恢复默认设置。

纹理

目录

- 纹理
 - 纹理类型
 - 通用设置
 - 漫射贴图
 - 高光贴图
 - 法线贴图
 - 高度图。视差贴图
 - 蒙版贴图
 - 视频贴图
 - 环境贴图
 - 镜面贴图
 - 天穹
 - 特殊的纹理类型

纹理是手工制作的或程序上生成的图像，可应用于模型表面，以添加更多的细节。依据规则，图像像素使用纹理映射的方法被分配到三维表面上。有时它们统称为贴图。

通常的纹理被放入材质 纹理插槽。他们可以也被用于粒子系统 参数化和用于创建天空纹理.

纹理类型

类型下拉菜单（用于选择纹理类型）位于 纹理选项卡下。引擎支持以下的纹理类型 [F]

1. 图像或影片

在这种情况下，纹理是由一个图像或视频文件中定义。下列文件格式支持：

- .PNG
- .JPG

它可用于以下目的：

- 漫射贴图
- 高光贴图, 这个也可以打包进漫射纹理的 alpha 通道
- 法线
- 高度图; 这必须打包进法线贴图的 alpha 通道; 它用于可视化的浮雕表面 (视差贴图).
- 蒙版贴图
- 视频纹理

2. 环境贴图

- 镜面贴图
- 天空纹理
- 用于实现环境照明 方法

3. 无

- 适用于 Blender 的默认场景的立方体。它也用于渲染场景到纹理 和渲染画布纹理.

4. 混合, 梯度

- 使用在粒子系统

通用设置

尺寸 图像纹理的位图尺寸（图像宽度和高度，以像素为单位）应为 2^N ，即 4、8、16、32、64、128、256、512、1024、2048、4096 px。与其他尺寸的（所谓 NPOT）纹理是支持但不是建议使用。为纹理压缩图像至少 4 像素。通常使用方形图像（例如 512 x 512 px），但是也可以使用矩形的（如 4 x 128 px）。不推荐使用图像大于 2048 像素的图。

图像贴图 > 扩展 纹理坐标解释模式（在 WebGL 交换模式）。这是可用于 图像或电影纹理类型。在 重复值设定的情况下由引擎设置纹理的 重复模式。在这种情况下纹理坐标的整数部分被忽略、小数部分被使用。在所有其他情况下（例如 扩展）引擎设置 CLAMP_TO_EDGE 模式。在这种情况下纹理坐标限制到 [0, 1] 段。默认值是 重复.

贴图 > 坐标 纹理坐标类型。支持的类型是 UV（使用 UV 映射），法线（直对相机，只供的漫反射贴图，用于创造 材质捕捉, matcap）和 生成。默认值是 生成。

贴图 > 大小 沿着各自的轴缩放 UV 贴图。默认值是 1.0。

导出选项 > 不导出 不导出纹理。

导出选项 > 禁用压缩 禁用这个纹理的纹理压缩。在 [纹理压缩](#) 恶化图像质量情况下使用。例如用于混合材质的不同部分建议禁用压缩遮罩纹理。

导出选项 > 海岸距离贴图 使用在室外渲染.

导出选项 > 各向异性过滤 各向异性过滤单独对纹理进行影响。对场景其他类似的参数有更高优先级别。默认值是 默认（即使用场景设置）。

水泡沫 泡沫的纹理。在水的材质中使用。

注解: 纹理使用法线贴图时纹理压缩被禁用。

漫射贴图

漫射贴图用于指定散射光的分布（Lambert 模型）。

激活

启用 纹理 > 影响面板上的 漫射 > 颜色复选框。

附加设置

影响 > 漫射 > 颜色 对漫反射颜色上的纹理的影响。默认值是 1.0。

影响 > 混合 与材质颜色相互作用的类型 (材质 > 漫反射 > 颜色), 或与顶点颜色如果启用了 顶点颜色绘制选框。支持以下类型: 混合 (混合的颜色), 相乘 (乘以颜色)。默认值是 混合。

高光贴图

高光贴图用于指定反射光的颜色分布 (Phong 模型)。

激活

启用 纹理 > 影响面板上的 镜面 > 颜色复选框。

附加设置

影响 > 镜面 > 颜色 对反射光颜色纹理的影响。默认值是 1.0。

影响 > 混合 反射光的颜色与材质的交互作用的类型 (材质 > 镜面 > 颜色)。混合 (用颜色混合) 是唯一受支持的类型。默认值是 混合。

出于优化的目的高光贴图可以打包到的漫射纹理的 alpha 通道中。在这种情况下是纹理同时启用 漫射 > 颜色和 镜面 > 颜色复选框。颜色范围被限制在灰色色调。

法线贴图

法线贴图用于指定的表面法线 (垂线) 分布为了立体的细节。关于法线信息应存储在纹理空间的坐标。法线贴图不支持在对象空间坐标的烘焙。

激活

设置 图像 > 彩色空间参数到 非 Color。

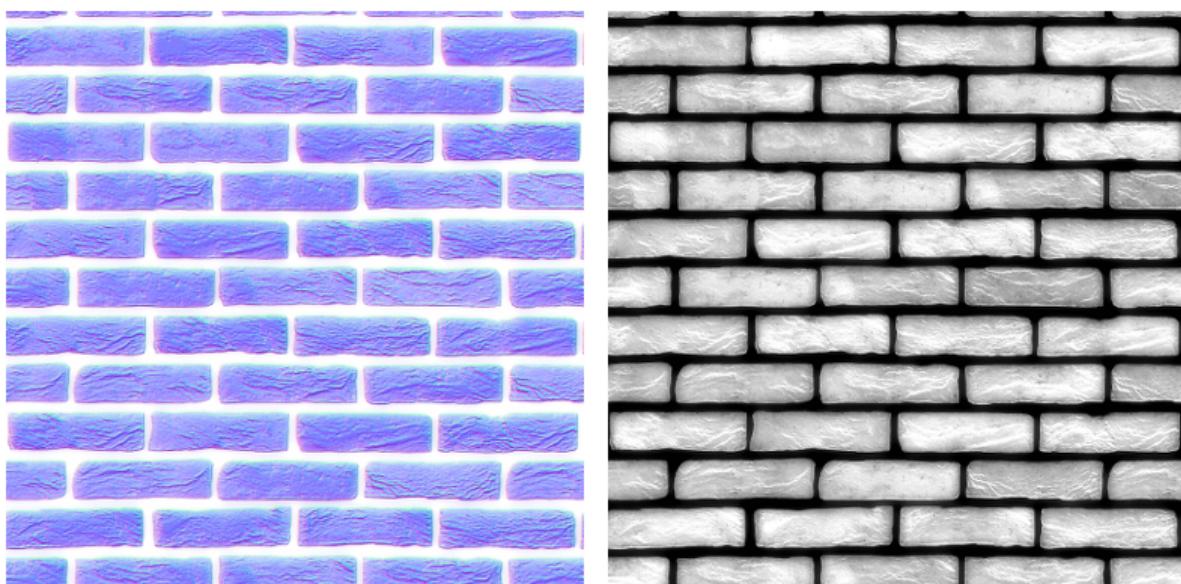
启用 纹理 > 影响面板上的 几何 > 法线复选框

附加设置

影响 > 几何 > 法线 法线贴图影响产生的法线计算。默认值是 1.0。

高度图。视差贴图

高度图包含相对宽缓的高度分布信息。表面的高度越高，颜色越明亮。高度图结合法线贴图都能展示立体效果（视差贴图）的实现。高度图应该出现在法线贴图的 alpha 通道。



激活

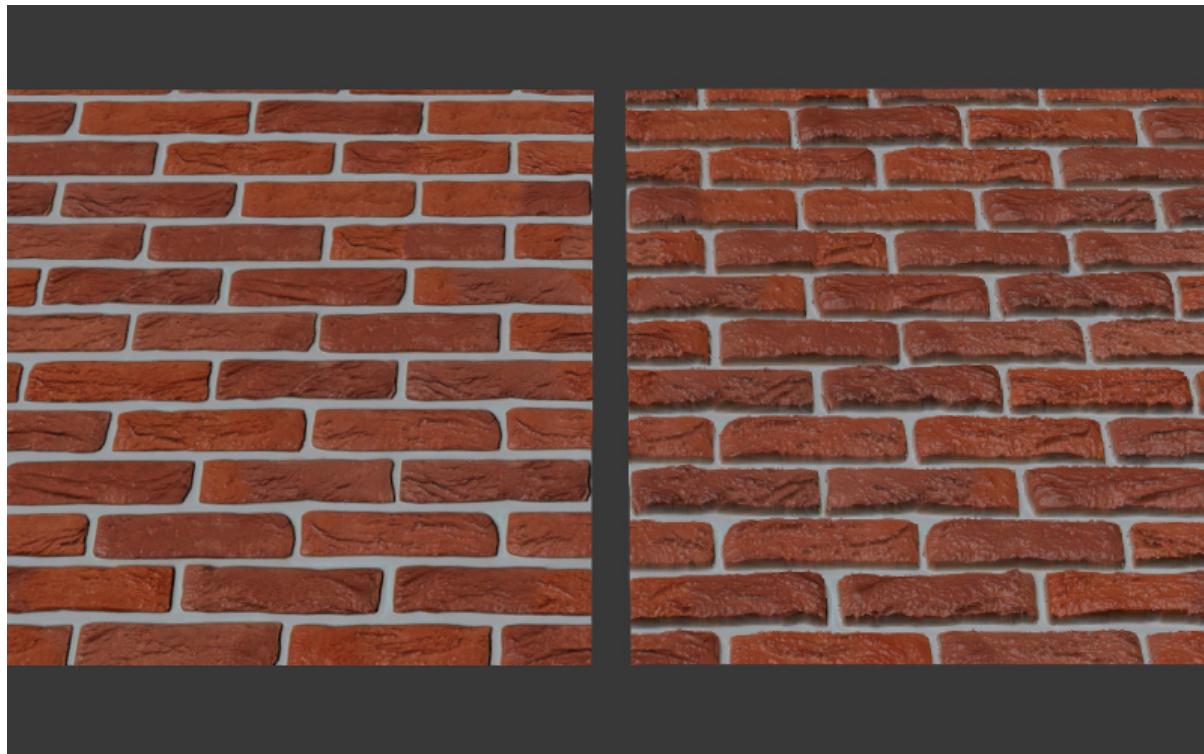
对于法线贴图在 几何 > 正常复选框的 视差面板启用。

附加设置

视差 > 视差缩放 表面立体效果的影响因素。默认值为 0.03。

视差 > 视差步 为立体表面计算的迭代次数。更大的值导致更好的质量但更多更昂贵的计算。

视差 > 视差 LOD 距离 观察者的距离视差效应。



蒙版贴图

特殊用途纹理 (彩色或灰度) 包含有关其他纹理表面的分布信息。

激活

1. 如果节点材质蒙版贴图应该用于相应的节点结构。
2. 在通用材质的情况下蒙版贴图应位于两个混合弥漫纹理之间的纹理插槽。蒙版贴图要求同时启用 纹理 > 影响面板上设置 RGB 到强度和 蒙版复选框。

附加设置

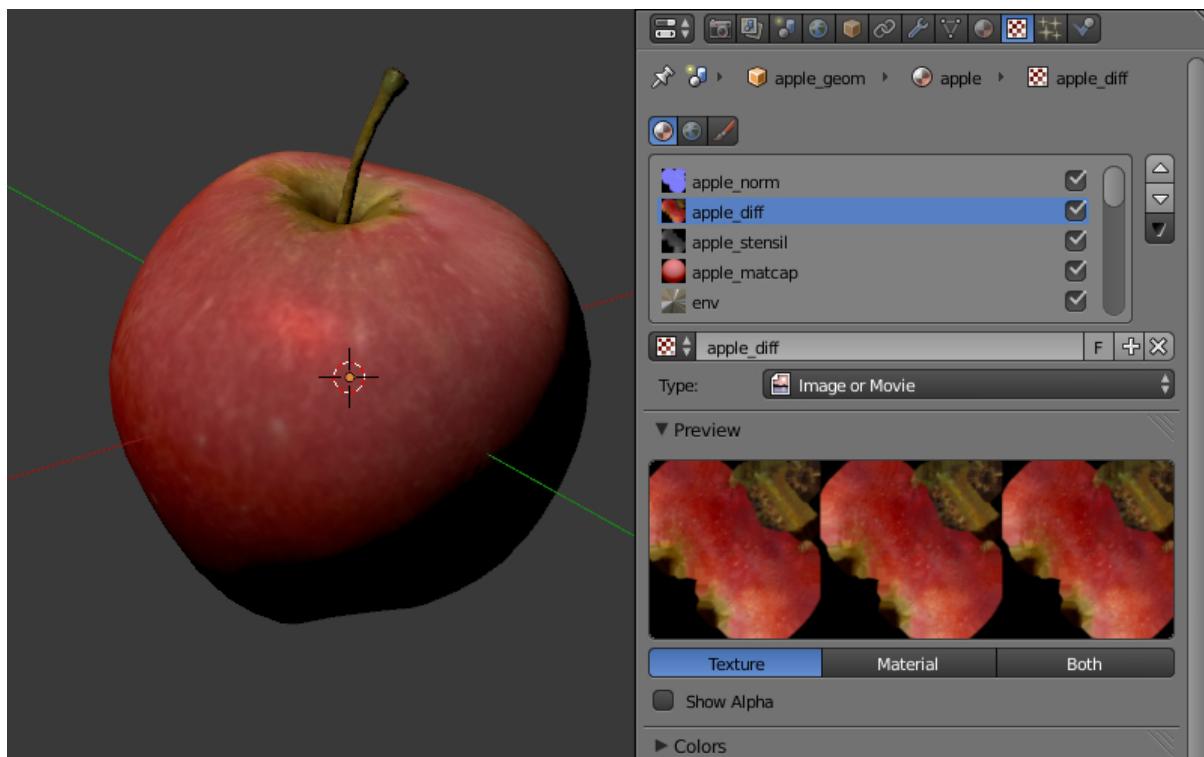
有一个混合的漫射纹理的通用的材质可以有 法线 (“matcap”) 纹理坐标类型。

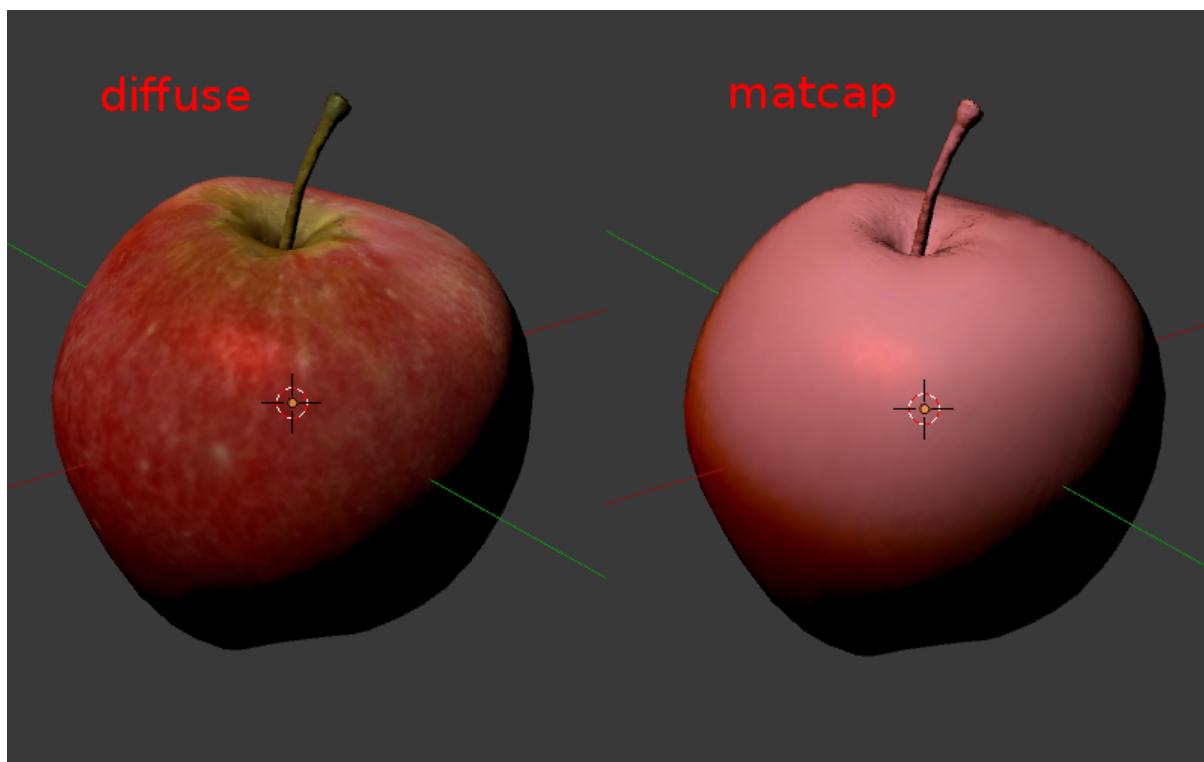
限制条件

在通用材质的情况下，引擎只是使用蒙版贴图的红色通道。高光贴图或法线贴图（如果有）不能混合在一起。映射 > 尺寸设置从第一个纹理提取并应用于所有剩余的纹理。

示例

苹果模型材质具有以下的纹理：法线贴图，alpha 通道有反射贴图的漫射贴图，蒙版贴图、漫射“matcap”贴图，环境贴图纹理。





视频贴图

如果选择了 图像或电影纹理类型，视频文件可以用作纹理。

注解： 视频纹理只是视频轨道支持播放。音频轨道应使用的 扬声器物体来播放。

支持的格式（容器） [E](#)

- webm VP8 编解码器 (Chrome、火狐浏览器)
- m4v, H.264 编解码器 (Chrome, Safari, IE)
- ogv, Theora 编解码器 (Chrome、火狐浏览器)

我们推荐使用 WebM 作为基本的格式。它是绝大多数浏览器所支持的开放标准，并提供不错的画质。

注解: 保存为 mp4 和 ogg 格式的文件有不同的音频和视频数据的扩展: .mp4 和 .ogg 扩展用于声音, .m4v 和 .ogv - 是视频。

不同格式之间的资源转换的描述在相应章节.

设置纹理

下面是 纹理 > 图像面板上的视频纹理可设置项:

图像 > 帧数 在帧中播放片段的长度。

图像 > 偏移 开始播放视频的帧数。

图像 > 循环 完成每次重新启动视频播放。

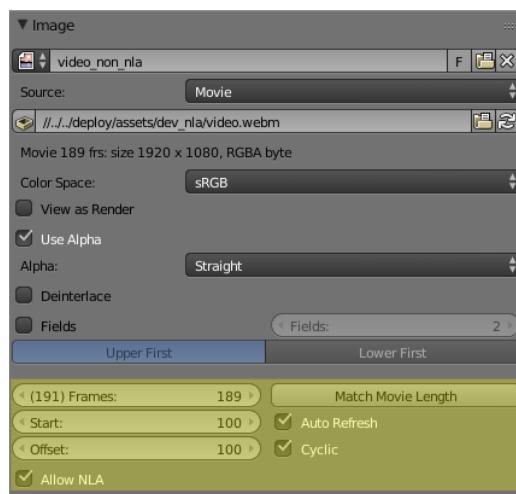
图像 > 允许 NLA 作为 NLA 轨道部分的播放出来的纹理。此外，在通过激活 场景 > NLA 全局场景设置中启用 NLA。默认情况下启用。

NLA 控制纹理以下选项也是适用的[F]

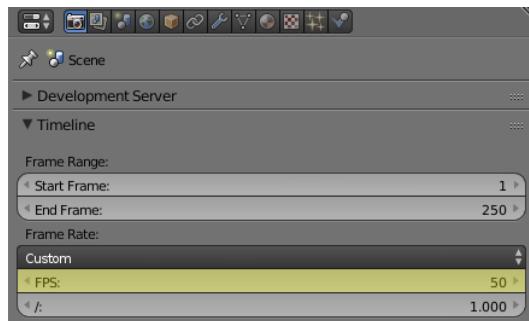
图像 > 开始 视频播放延迟（帧）时使用的非线性动画。

对于非 NLA 控制纹理以下选项也是适用的[F]

图像 > 自动刷新 场景加载后立即播放视频。



可以增加视频播放速率。要这样做需设置 **场景 > 尺寸 > 帧速率** 和视频的 FPS 值是不同的值。视频播放率是按比例增加到场景的 FPS 和视频的 FPS 的比例。



注解: 当视频纹理与 NLA 一起使用时, 视频播放可以不对应与 Blender 设置。即, 可能观察到 5-6 帧延迟是由于 `<video>` HTML 元素的开始/暂停的延迟。

移动设备的细节

移动设备特点如下

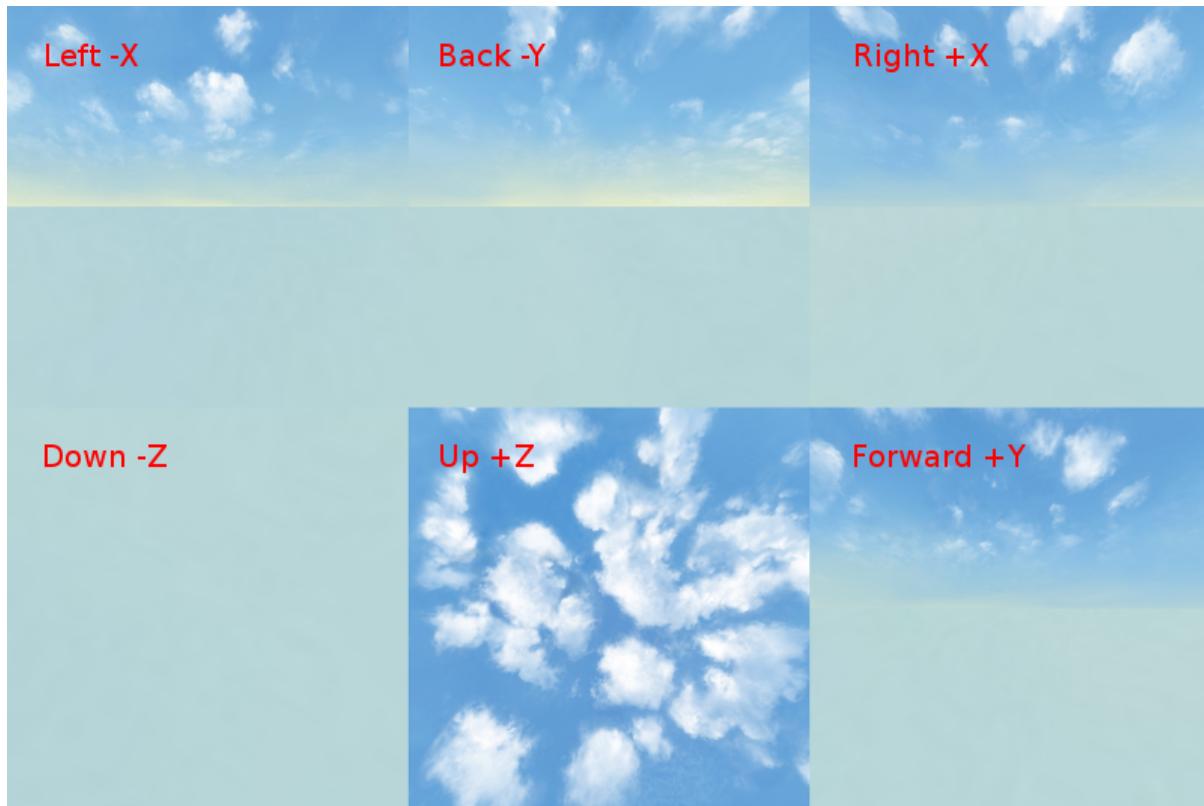
1. 视频纹理在 iPhone 上的正常运行是不可能的, 因为这些设备播放视频通过标准 iOS 视频播放器。你需要将你的视频转换为特殊的 ‘.seq’ 格式在这些设备上用, 通过使用我们[转换器](#).
2. 有些设备只支持播放一种视频文件。
3. 如果 偏移值不为零不能保证稳定运行。
4. 不是所有设备都支持更改视频播放速率。
5. iPad 和 iPhone 不能控制视频的音频音量, 在将文件添加到 Blender 前, 音频轨道应从视频中删除。

环境贴图

环境贴图可以用作镜面贴图, 作为一个静态天空纹理 (天穹) 也可以实现[环境照明](#) 方法。

引擎就默认作为立方体的纹理。环境贴图的位图应该包含环境图像 6 个方向投影, 封装在 2 行每行 3 件 Blender 格式)。为每个图像的位图大小应遵循 2^N 规则 (512、1024 等)。

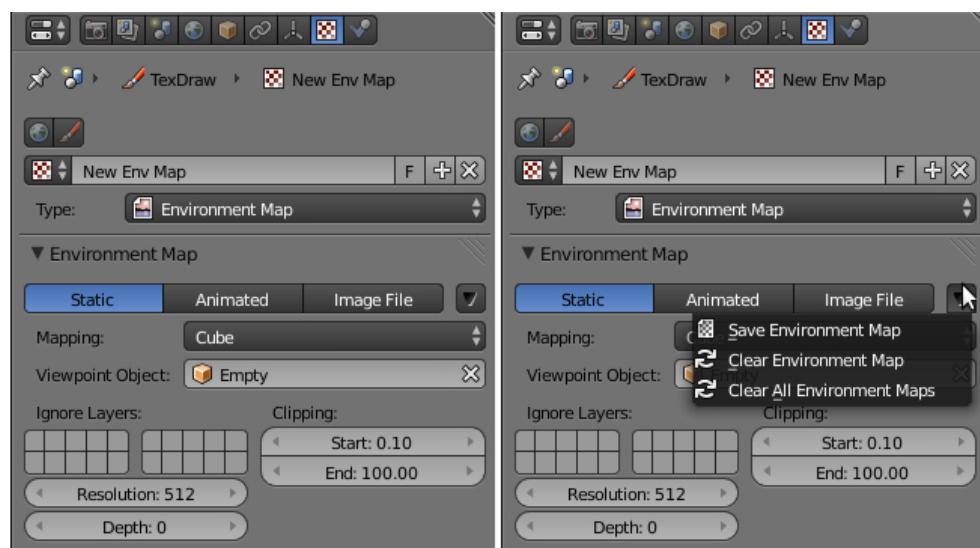
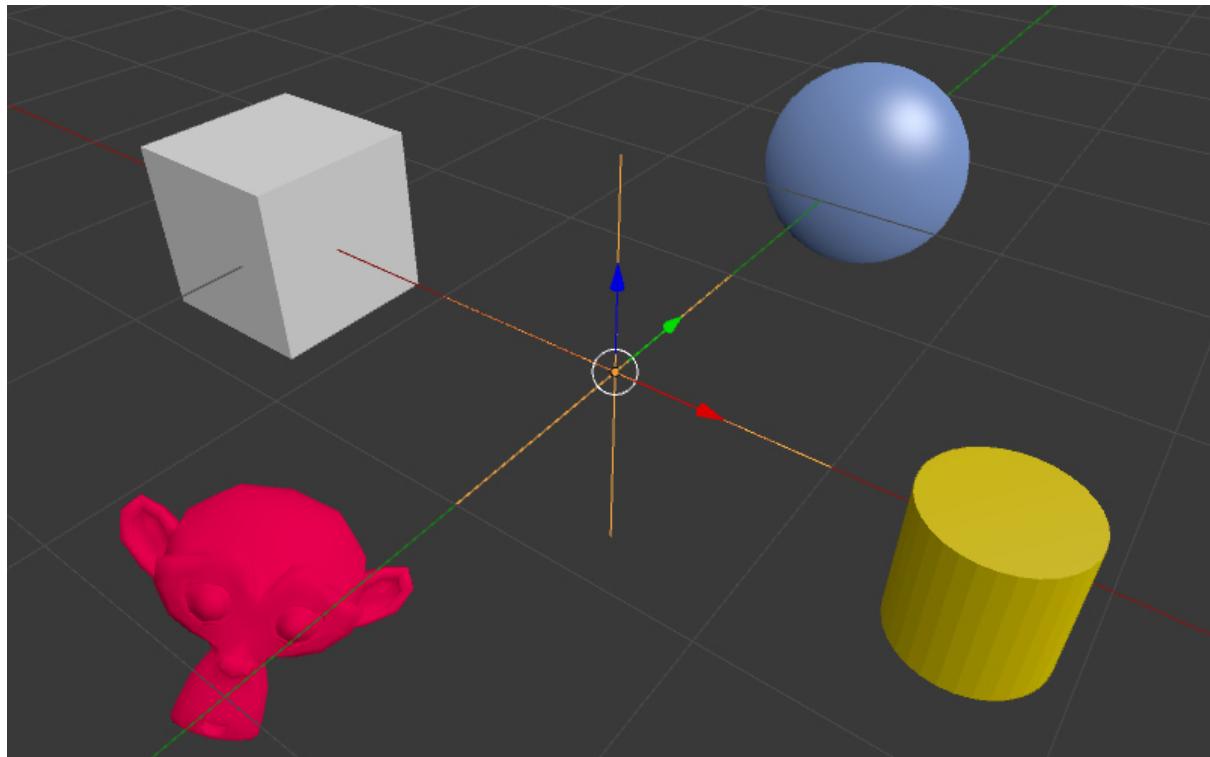
为了避免接缝，建议使用无损格式 (PNG)。



制作环境贴图

Blender 有烘焙场景到环境映射选项。要此操作 **F**

1. 创建一个场景进行烘焙。
2. 在视图默认点中添加一个空对象 (添加 > 空物体).
3. 转到 世界 选项卡，然后到 纹理 选项卡，用 环境贴图类型创建一个新的纹理。
4. 环境贴图面板上选择 静态的源，然后在 视口物体栏选择空物体，然后设置 2^N 维度 (512、1024 等)。
5. 按 ‘f12 键’ （相机是必需有的）渲染场景。
6. 将环境映射保存到一个文件。



镜面贴图

镜子贴图用于表面反射可视化。这是一个[环境贴图](#).

激活

选择 环境贴图纹理类型 (类型)。启用 纹理 > 影响面板上的 光影 > 镜子复选框。

附加设置

影响 > 光影 > 镜子 镜面贴图影响反射程度。默认值是 1.0。

参见:

[静态反射.](#)

天穹

天穹用于无穷远的环境可视化 (例如天空)。这是[环境贴图](#).

也可用来实现[环境照明](#) 方法。

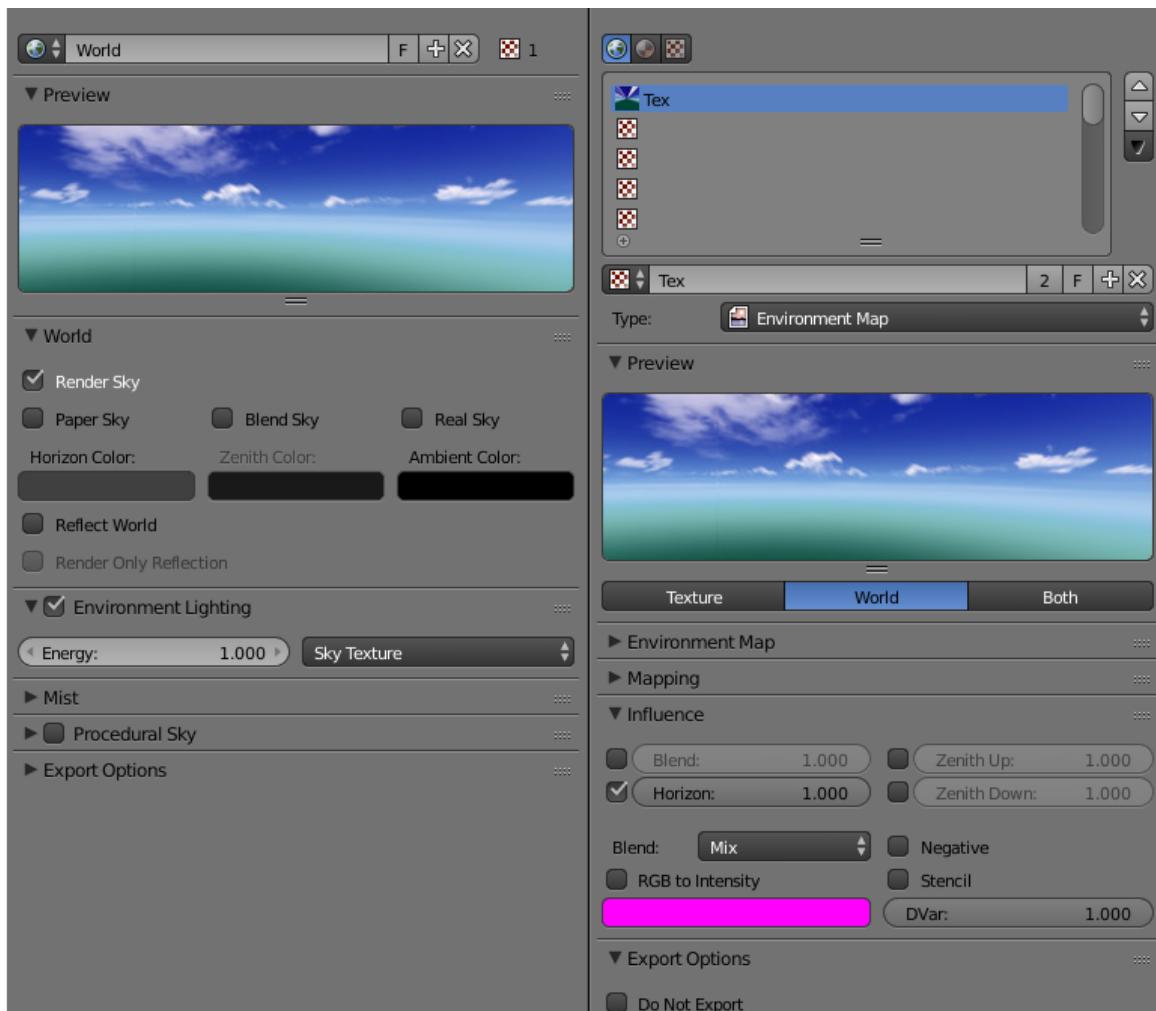
激活

用 环境贴图类型创建世界纹理。选择 导出选项 > 天空纹理使用 > 天穹选项。启用 世界选项卡下 世界 > 渲染天空。

注解: 纹理的行为是故意作出尽可能跟的 Blender 渲染器一致的效果。这就是为什么纹理可能不会显示在其默认设置。为使纹理可见, 设置面板上的 影响 > 地平线复选框, 将 地平线值设置为 1.0。

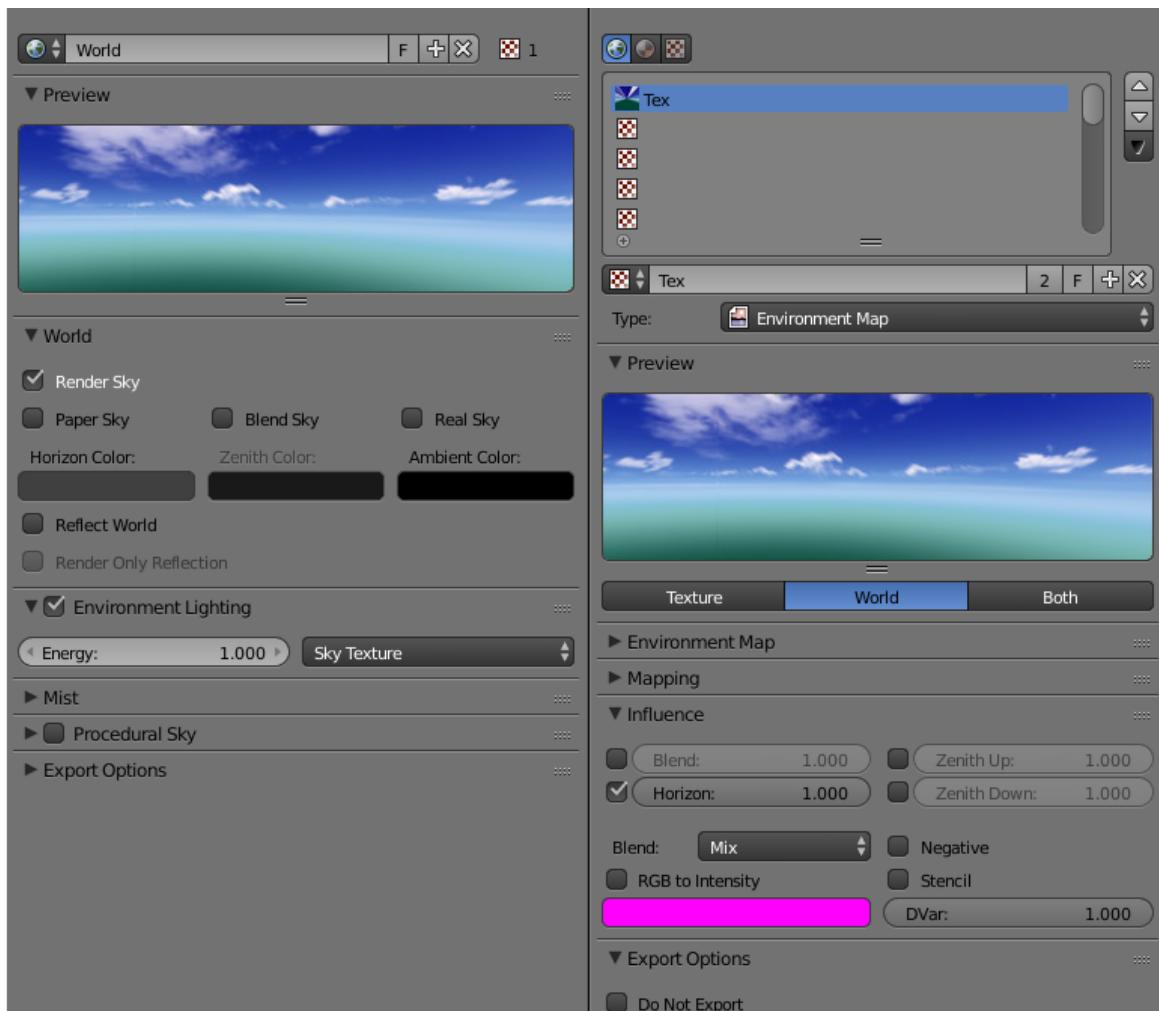
注解: 模仿[环境光照](#) 你可以选择 导出选项 > 天空纹理使用 > 环境光照选项。此外, 你应该在世界设置选中相应的选项 环境照明 > 天空纹理.

为天穹和环境照明都使用世界纹理, 请选择 导出选项 > 天空纹理使用 > 同时。



附加设置

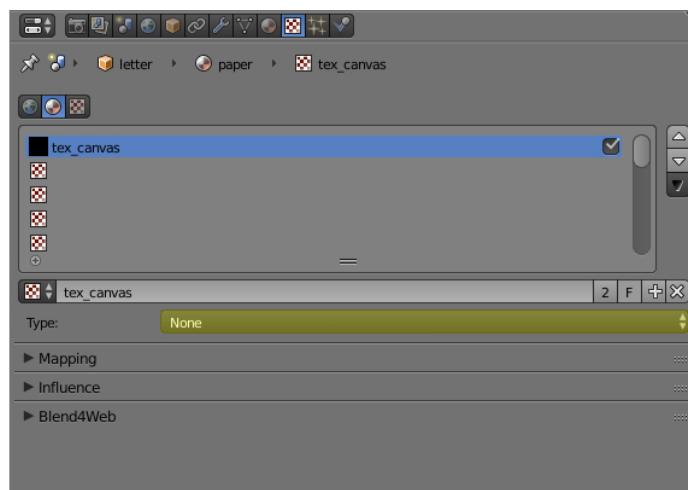
引擎还支持世界纹理的 影响面板的参数用于天空渲染。混合的世界纹理与颜色取决于 世界 > 地平线颜色 和 世界 > 天顶颜色 的参数，并且在 纸天空、混合天空 和 真正的天空 选项。支持所有的混合选项（混合、添加、乘等）。



注解: 影响面板参数只影响天空渲染。他们不会以任何方式影响环境照明

特殊的纹理类型

为了使用这种纹理，选择 无类型下的 纹理选项卡。



在 纹理 > 导出选项面板上，您可以对这些纹理设置属性

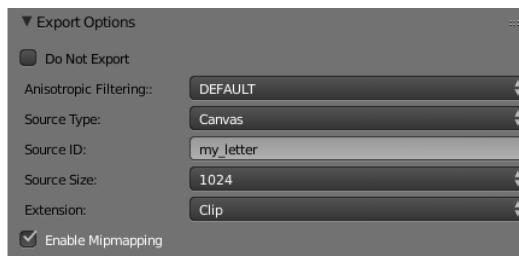
导出选项 > 源类型 选择纹理类型：场景 - 用于渲染 3D 场景成为纹理，画布 - 使用 `<canvas>`HTML 元素和“无” - 用于指明没有类型。

导出选项 > 源 ID 要渲染到纹理的场景名称（为 场景）或 `<canvas>` 的 HTML 元素 ID 的名称（为 画布）。

导出选项 > 源大小 纹理分辨率。

导出选项 > 扩展 纹理坐标解释模式。默认值是重复。

导出选项 > 启用纹理多级映射 (Mipmap) 为画布纹理启用纹理多级映射。默认情况下启用。



渲染到纹理

三维场景实时渲染的图像可以用作其他场景（“主”场景）的物体纹理。这种技术被称为渲染到纹理 (RTT)，可以通过以下步骤激活

1. 创建将被渲染到纹理的附加场景。
2. 为方便起见，给这一场景的唯一名称。
3. 场景中设置摄像机和光源
4. 在 3D 场景添加你需要的物体：
5. 场景中设置摄像机并设置
6. 然后，切换到主场景。
7. 选择目标物体，并为它创建一个 UV 贴图。
8. 创建一个纹理将作为渲染目标。
9. 设置纹理类型为 None
10. 对于 Mapping 选项卡下的 Coordinates 参数设置 UV 值。
11. 在 导出选项 > 源类型菜单中选择 场景类型。
12. 在 导出选项 > 源 ID 字段中指定源场景的名称。
13. 在 导出选项 > 源大小字段（以像素为单位）设置纹理的大小。



引擎还支持场景相互循环渲染。

注解: 项目应包含至少一个场景，不由任何其他场景渲染完成。

Canvas textures

一个 <canvas>HTML 元素可以用作纹理。它可以通过 API 修改。

设置主场景目标物体为 无类型的纹理，在 导出选项 > 源类型菜单中选择 画布类型。在 导出选项 > 源字号字段（以像素为单位）设置纹理的大小。

使用 纹理模块来处理这种纹理。请参阅下面的示例。

```
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");
```

使用 `get_canvas_texture_context()` 来获取上下文-此方法需要的“`canvas_id`”标识符应在 Blender 中定义。当与上下文操作, `update_canvas_ctx()` 功能被调用, 渲染“`canvas_id`”元素的修改。

注解: 如果在 Blender 中画布类型材质分配给了几个不同的物体, 待引擎加载后它仍然将一个纹理而不是几个不同的。应用于它的任何更改将应用于所有使用此纹理的物体, 可用于优化的目的。万一不需要这种效果, 你在 Blender 中应指派不同的纹理, 或在引擎启动后使用深度复制。

粒子系统, 流体

目录

- 粒子系统, 流体
 - 用法
 - * 必要的步骤
 - * 推荐其他设置
 - 设置
 - * 基本设置
 - * 发射设置
 - * 方向设置
 - * 旋转设置
 - * 物理设置
 - * 渲染设置
 - * 支持设置力场的影响
 - * 引擎特定的设置
 - 粒子系统中的纹理
 - * 粒子材质的纹理
 - * 粒子系统中的纹理

粒子系统旨在可视化现象如由烟、火、水溅、引起的其他无数小物体的运动。



一个粒子系统要求有发射器-一个物体来定义粒子流的位置和传出的方向。

用法

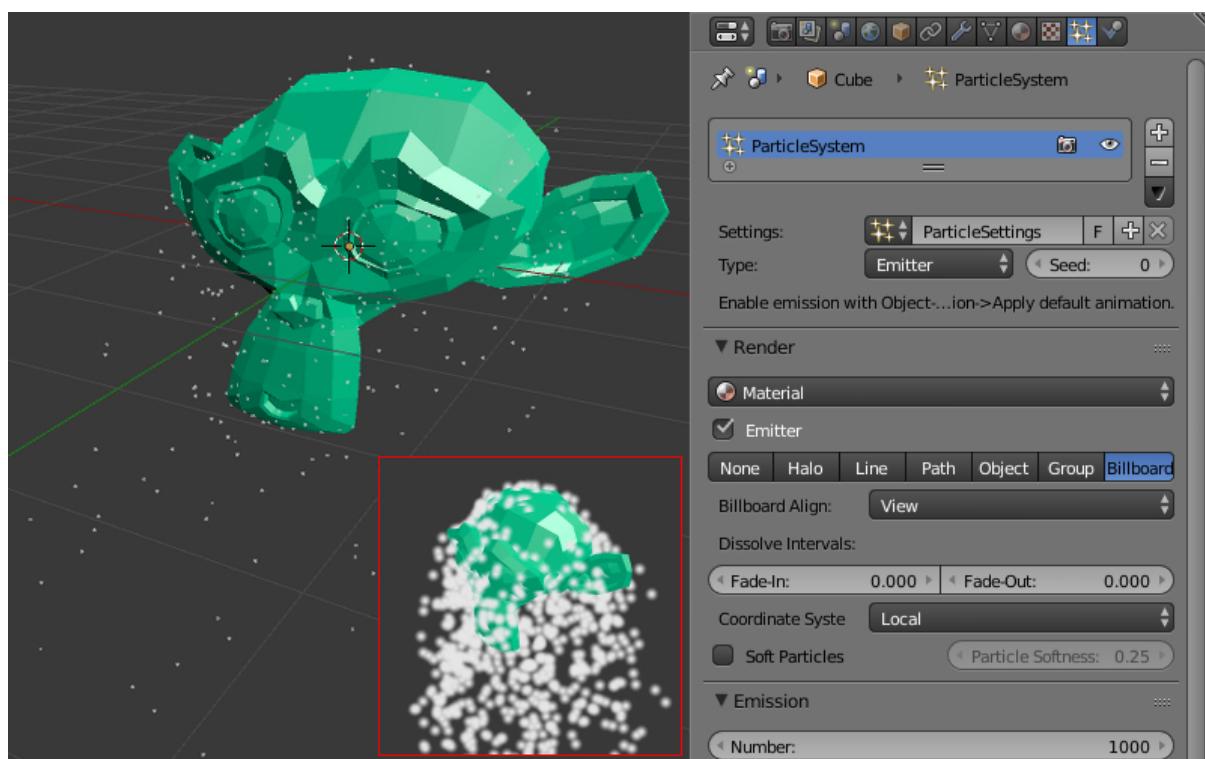
必要的步骤

1. 添加网格发射器到场景。
2. 在发射器上创建一个粒子材质，例如 Halo 类型上。此外支持必要的漫射纹理的 Surface 材质类型。
3. 在发射器上添加一个粒子系统。
4. 启动引擎播放。有两个选项可供选择：

- “cyclic emission” - 对粒子系统启用 Emission > Cyclic emission 对发射器启用 Animation > Apply Default Animation 。
- “non-cyclic animation” - 启用发射器的 Animation > Apply Default Animation 选项。

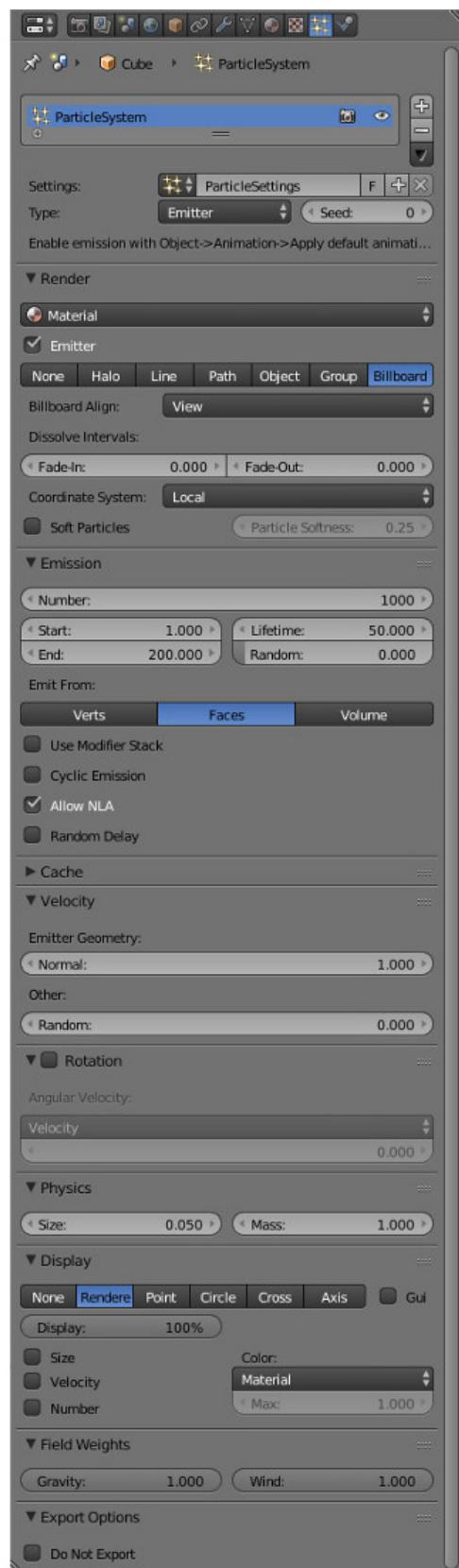
推荐其他设置

1. 设置粒子材质 Add 透明度类型。
2. 如果需要，禁用发射器渲染，使用 Particles > Render > Emitter 复选框。
3. 如果一个场景的发射器需要用到额外材质。在这种情况下，在粒子设置的 Particles > Render > Material 菜单选择粒子材质。
4. 如果使用了 Surface 材质类型它需要（通常与 alpha 通道）添加漫射纹理到该材质。在 Mapping > Coordinates 菜单中选择 UV 。确保发射器的网格有 UV 层。



设置

粒子系统参数可以在 Particles 选项卡下进行设置。支持每个发射多个粒子系统。



基本设置

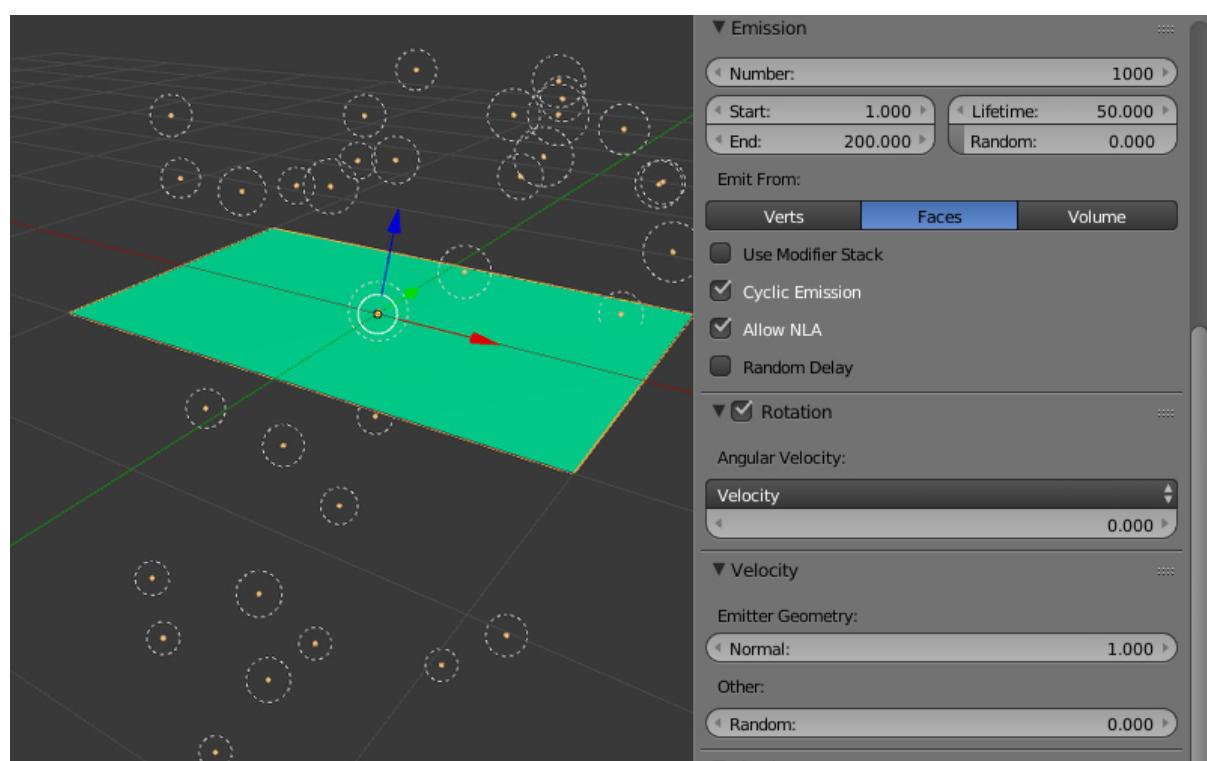
名字 粒子系统名称。默认名称是“ParticleSystem”。

设置 参照粒子系统的设置数据块。该数据块的设置可以在不同粒子系统之间共享。

类型 粒子系统类型: Emitter 或 Hair. Hair 粒子系统可以用于创建许多副本的（称为实例化）的物体。默认值为 Emitter.

随机种 粒子系统产生的随机数的表格索引。缺省值是 0。

发射设置



Emission > Number 粒子的数目。默认值为 1000。

Emission > Start 粒子发射开始的第一帧。缺省值为 1.0。

Emission > End 粒子发射结束的最后一帧。默认值为 200.0。

Emission > Lifetime 粒子以帧为单位的寿命。默认值是 50.0。

Emission > Lifetime > Random 生存期随机因子。默认值为 0.0。

Emission > Emit From 发射源类型。支持以下类型 Verts (从顶点发出), Faces (从多边形发出)。默认值为 Faces。

Emission > Cyclic emission 该选项启用循环发射模式。它可用于永久效果 (如烟, 燃烧, 水溅)。建议设置的 Emission > Start 的值为零。默认为禁用。

Emission > Allow NLA 激活此选项将允许粒子发射由 NLA 进行控制。默认情况下启用。

Emission > Random Delay 该选项允许启用粒子随机发射时间。默认情况下禁用。

方向设置

Velocity > Emitter Geometry > Normal 沿着发射器的网格法线的发射影响因子。缺省值为 1.0。

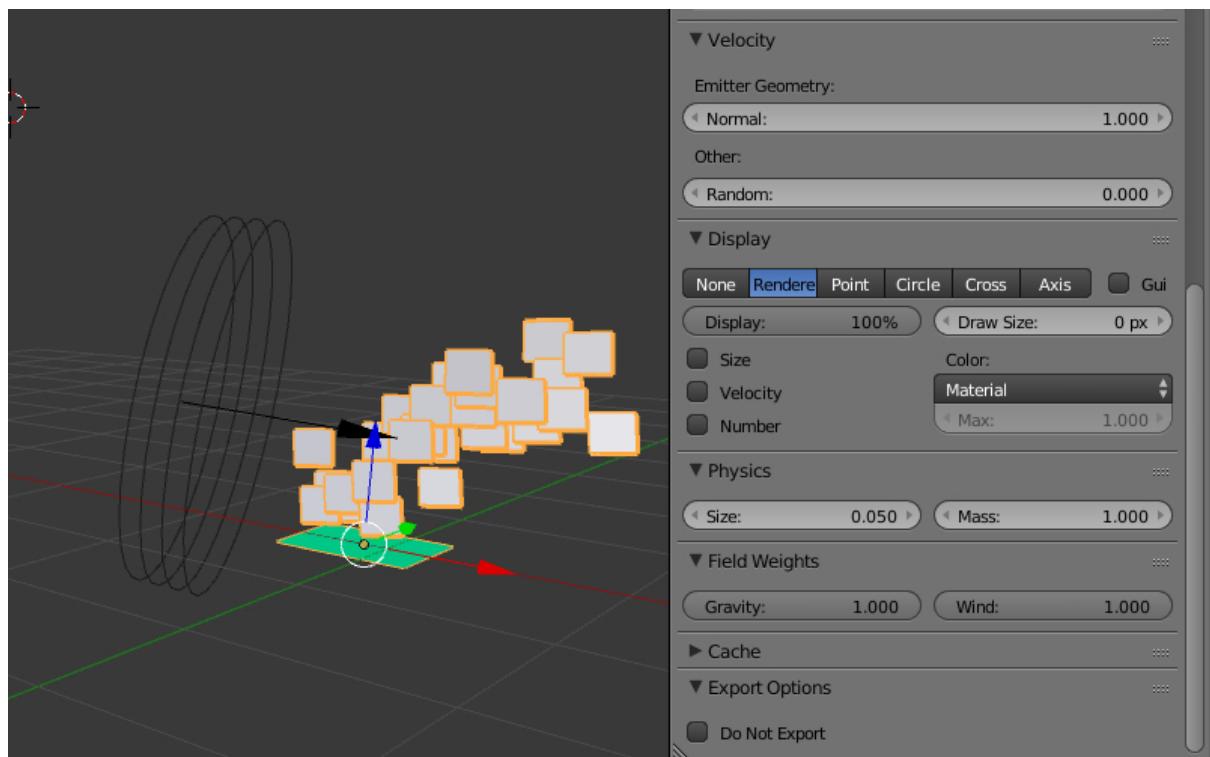
Velocity > Other > Random 随机的发射方向的因子。默认值是 0.0。

旋转设置

Rotation > Angular Velocity > Mode 模式为粒子广告牌的自转。Hair 粒子系统支持所有类型的旋转, Emitter 只支持 Velocity。

Rotation > Angular Velocity > Factor 粒子广告牌旋转速度的因子。默认值为 0.0。

物理设置



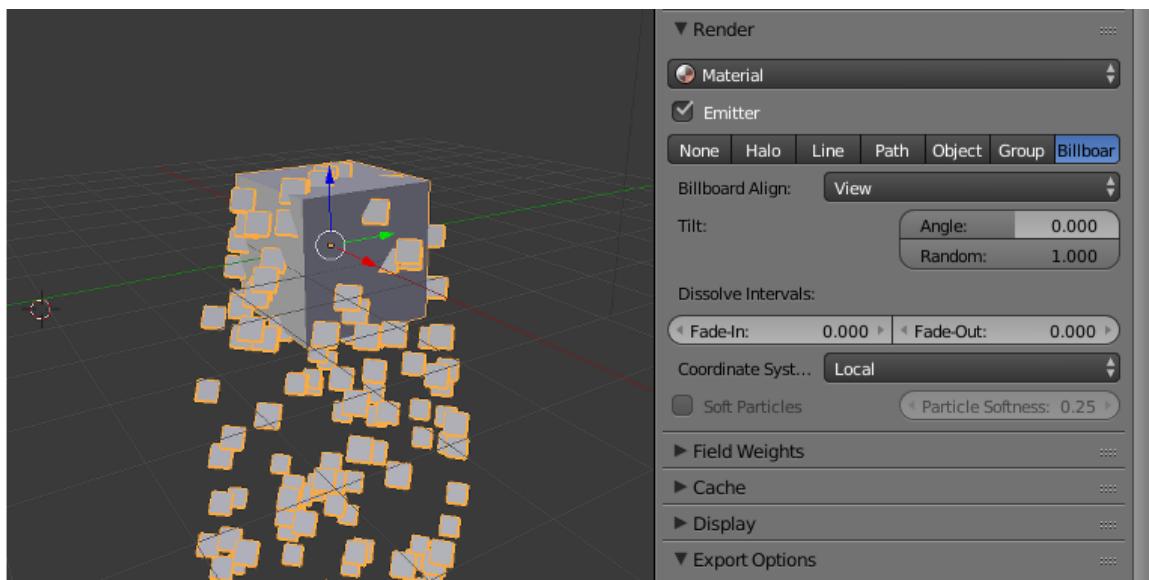
Physics > Type 物理计算类型: No, Newtonian, Keyed, Boids, Fluid. 由引擎忽略。总是用 Newtonian 。默认值为 Newtonian 。

Physics > Size 粒子大小。默认值为 0.05。

Physics > Mass 粒子质量。影响与力场（风能）的互动。默认值是 1.0。

Physics > Forces > Brownian 导出但是不在引擎中使用。

渲染设置



Render > Material 选择粒子材质的菜单。用于发射器有多个材质的情况下粒子材质。
默认值为 Default Material。

Render > Emitter 启用场景中发射器渲染。默认情况下启用。

Render > Type 粒子渲染模式: None, Halo, Line, Path, Object, Group, Billboard。
该引擎支持 Object 和 Object 和 Group 分别实例化模式。其他模式将被忽略。推荐使用 Billboard 模式，广告牌方便的显示。默认值是 Halo。

Render > Billboard Align 广告牌的方式取向: View - 按照相机, XY plane, YZ plane, ZX plane - 对准相应的平面（在世界坐标搅拌机系统）。默认值是 View。

Render > Tilt 这组参数控制初始的广告牌飞机倾斜。它是由两个参数 **角度** 广告牌平面的旋转角度。它的值可以从 -1 到 1。旋转 180 度的值为 1（即颠倒广告牌）。设置为零，默认情况。
随机 倾斜的随机变化。它的值可以从零到 1 更改，默认情况设置为零。

Render > Dissolve intervals > Fade-in and Fade-out 开始和结束的时间间隔（以帧测量的）为逐渐增加和减小粒子的透明度。

Render > Coordinate System 发射粒子的坐标系统: Local - 使用局部坐标发射物体的系统，World - 使用世界坐标系。

Render > Soft Particles 支持软粒子，它的广告牌与表面接触时能够消融。使用 Particle Softness 滑块来调整这种效果。

支持设置力场的影响

Field Weights > Gravity 重力影响因素 (地球引力)。缺省值为 1.0。

Field Weights > Wind 风的影响因子。Wind 力场源需要存在 (可以 Add > Force Field 添加使用)。一个粒子系统也受到风的方向和强度影响。默认值是 1.0。

引擎特定的设置

Export Options > Do not export 不要导出。

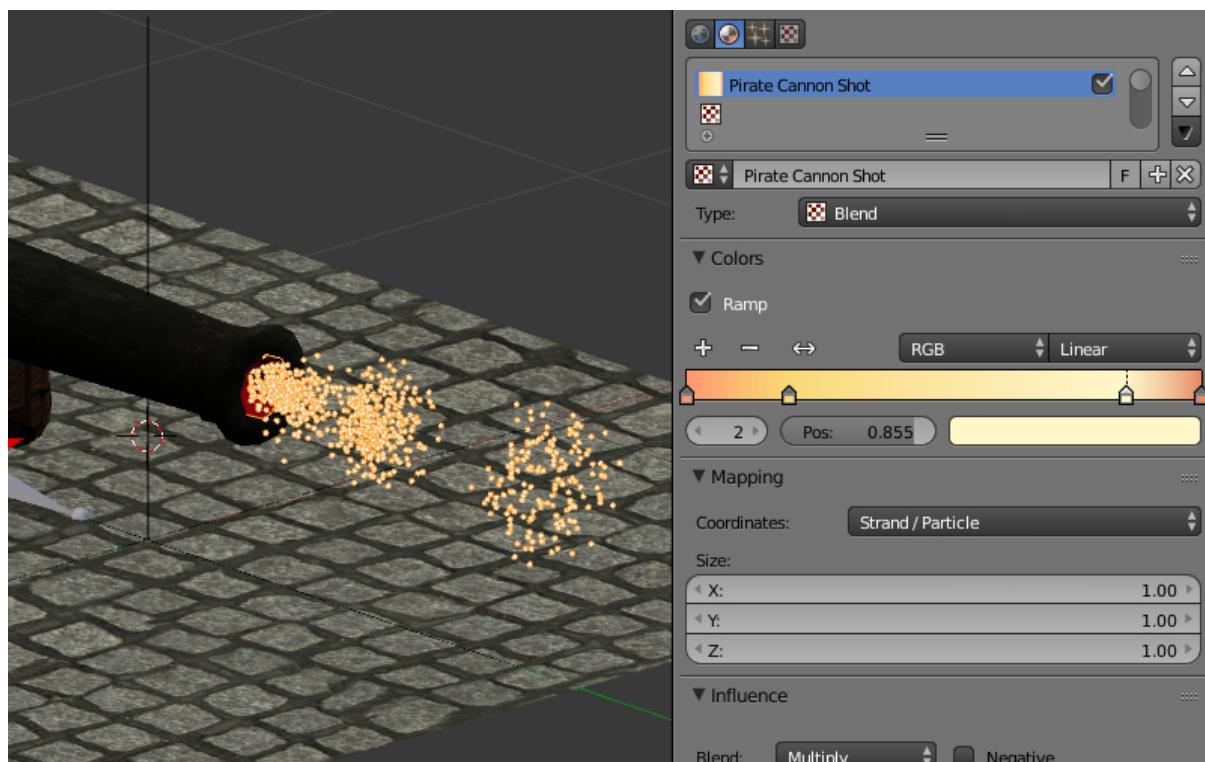
粒子系统中的纹理

粒子系统支持标准和节点的材质。支持的材质类型包括 Surface 和 Halo。此外支持 Halo 类型的材质，如 Rings, Lines 和 Star Tips，所有其他设置。

粒子材质的纹理

Surface 粒子材质它是 **要求**有漫射纹理 (通常带 alpha 通道)。在 Mapping > Coordinates 菜单中选择 UV 选项。请确保发射器的网格有 UV 层。

对于 Halo 粒子的材质，这是 **可以**使用 Blend 纹理一个 Linear 梯度。在 Mapping > Coordinates 菜单中选择 Strand / Particle 选项。它要求对纹理启用 Ramp。

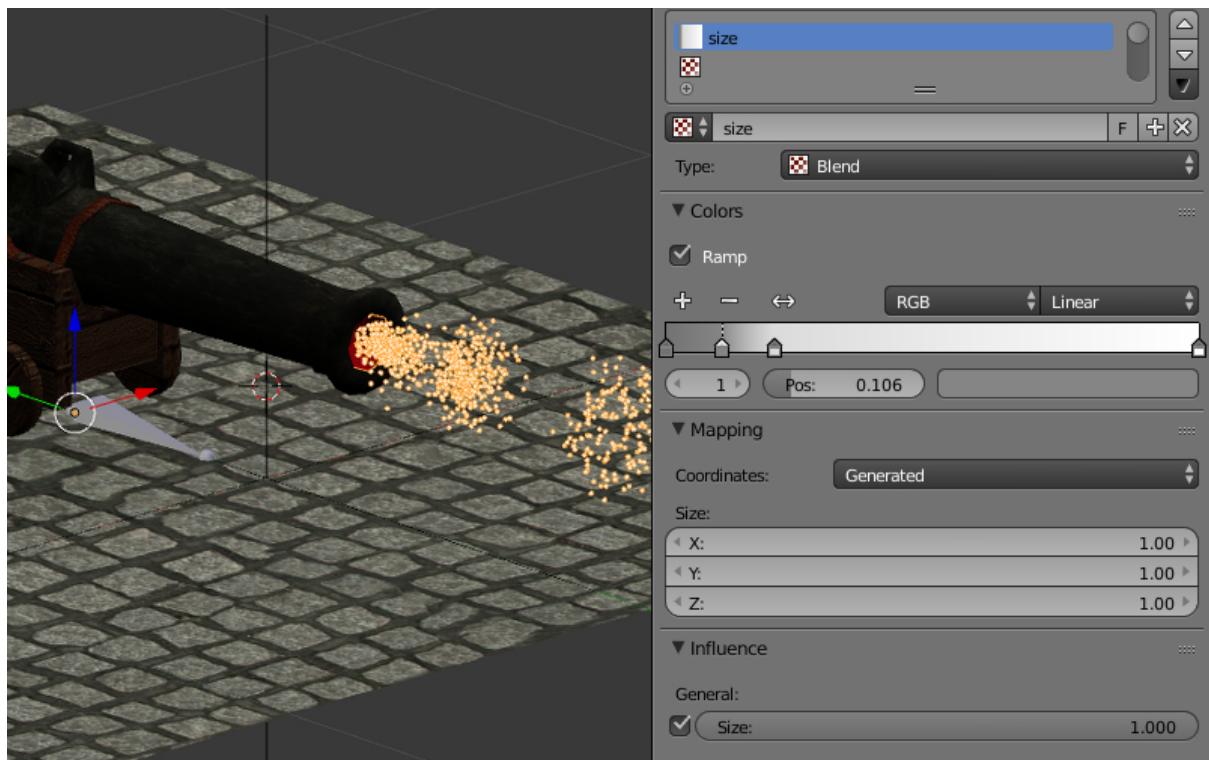


粒子系统中的纹理

纹理也可以用于设置粒子系统的行为。与纹理粒子材质不同的是这种纹理属于粒子系统数据块，不到材质数据块。若要创建纹理的粒子系统，它需要去 **从 Textures 选项卡的 Particles 选项卡**，然后单击新建按钮。

纹理唯一支持的类型是 Blend 带 Linear 梯度。纹理的 Ramp 应该启用。支持多达 4 个梯度控制点。

Influence 面板上选择参数受纹理的影响。目前唯一支持的参数是 Size.



粒子材质使用渐变纹理和粒子系统的结果



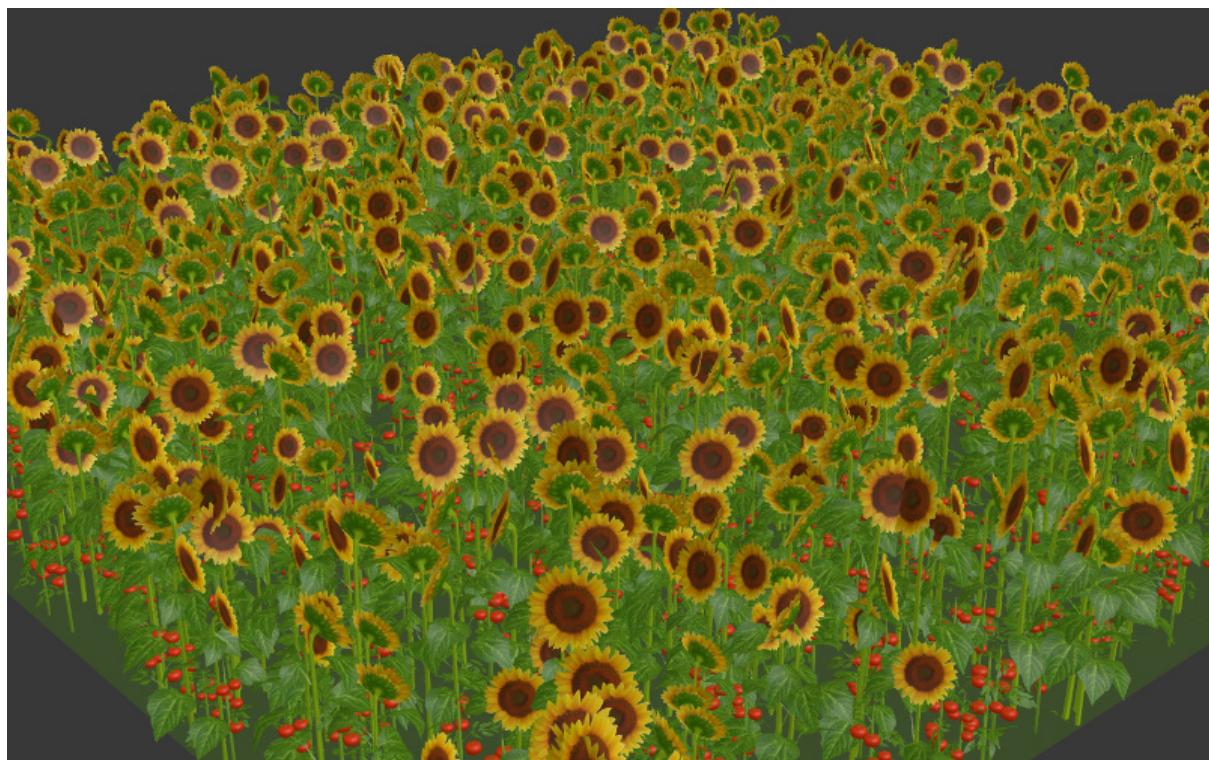
原始模型是从这里取得

粒子系统. 实例化

目录

- 粒子系统. 实例化
 - 粒子系统设置
 - * 激活
 - * 显示设置
 - * 动态草设置
 - * 继承设置
 - 草地
 - 树叶

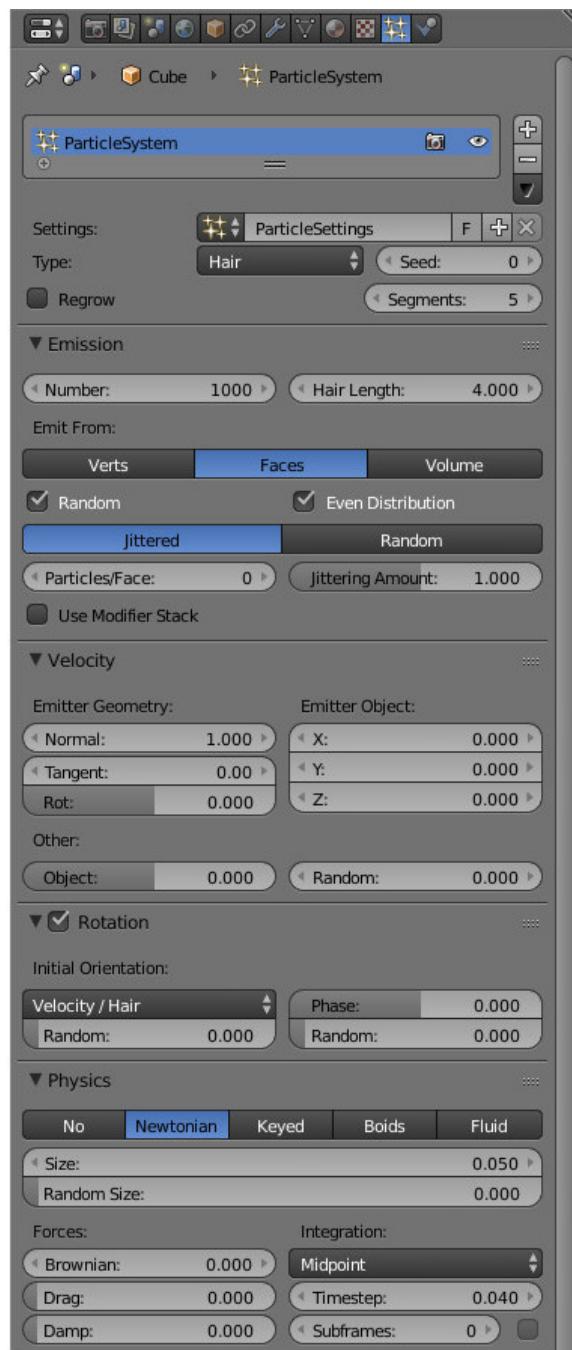
一个粒子系统可以用于创建多个物体的副本（所谓的实例）。这种技术简化了场景创作，与使用单个物体相比可以缩短加载时间和内存消耗。

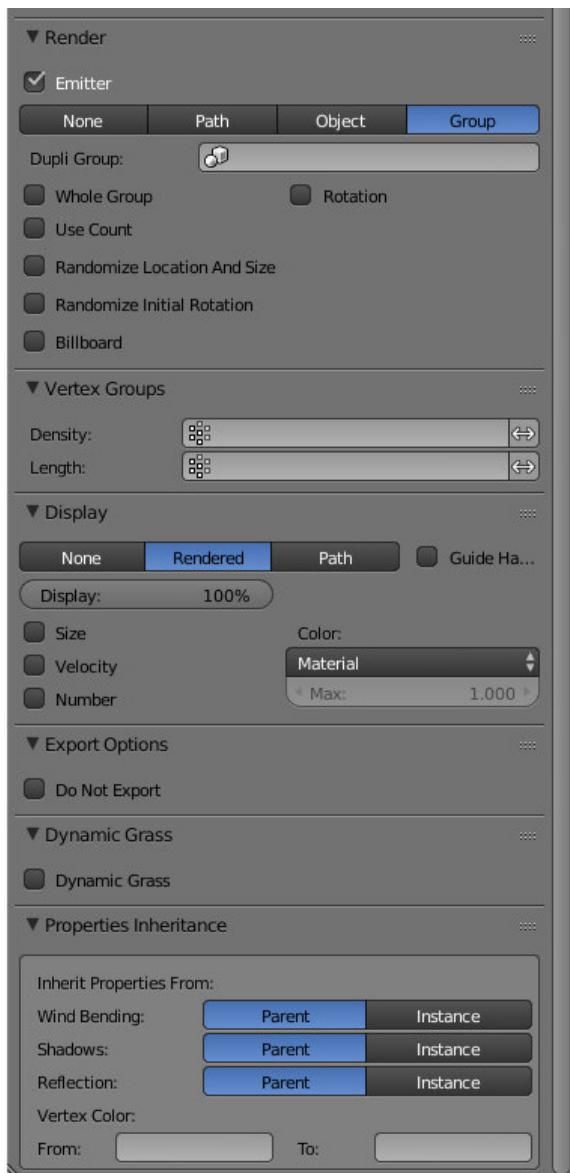


使用粒子系统实例化有一定的局限性，如：

- 粒子系统内的物体的移动和动画是不允许的。
- 粒子系统内的物体是没有父级关系的，除了 dupli-groups。
- 非网格物体的实例化是不可能的。

粒子系统设置



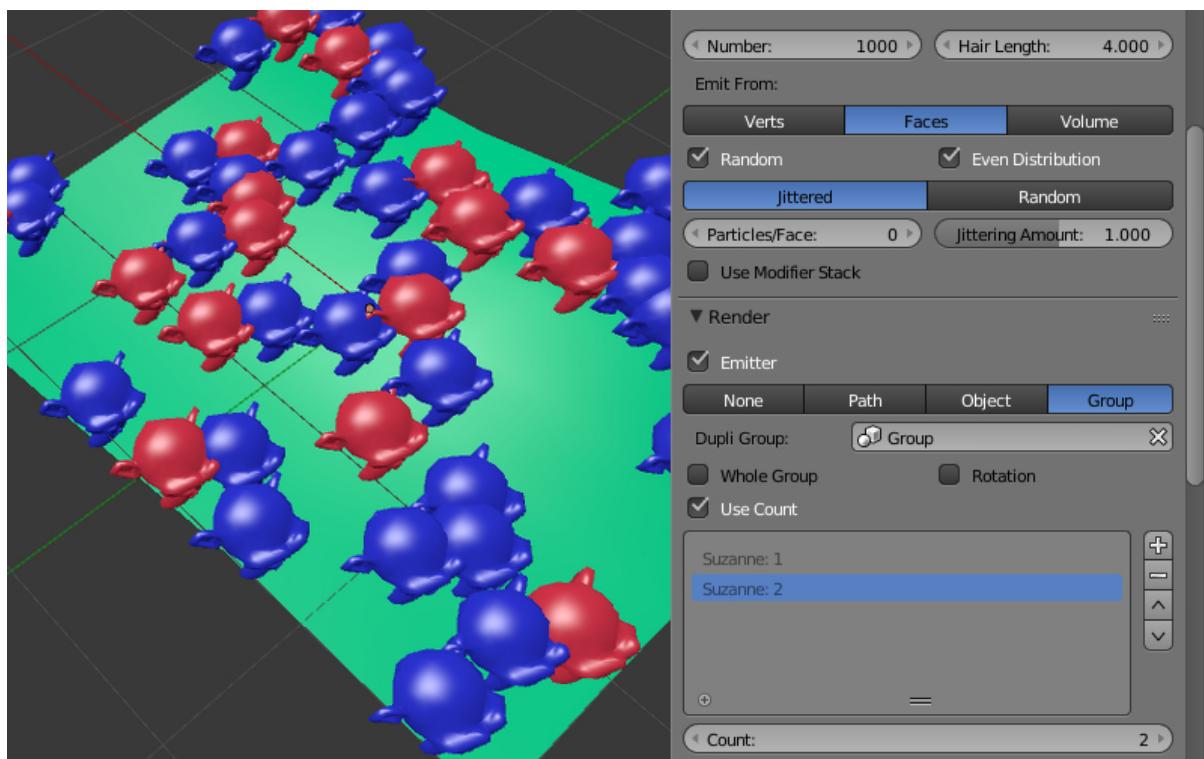


激活

1. 在发射器创建 Hair 类型的粒子系统。
2. 在 Render 面板中选择 Object (或 Group) 渲染类型。
3. 在 DUPLI Object 字段 (或在 DUPLI Group 字段) 选择用于实例化的物体 (或物体组)。支持本地和链接的物体 (或组)。

建议的额外设置

1. 为了在视口中显示正确的尺寸，设置 Emission > Hair Length 和 Render > Size 参数为 1.0。



显示设置

Render > Use Count

该选项可用于粒子物体组。当启用时，用于设置一个组中的物体的相对数的界面变得可见。该引擎不会重现某些类型的物体的确切位置。

Render > Randomize Location and Size

该选项使随机化的位置和对象的大小。如果启用引擎产生粒子物体随机坐标和大小（仅限于 $\pm 25\%$ 范围内）。如果禁用，粒子物体的准确坐标和尺寸在导出后使用。

Render > Randomize Initial Rotation

这个选项随机化由 Rotation Type 限定的轴线的物体的旋转。如果启用，引擎产生的粒子随意旋转角度。如果禁用，转动使用 Rotation 面板设置。

Render > Rotation Type

随机物体的旋转轴（属性时可用 Render > Randomize Initial Rotation 启用）。有两个选项：

- Z axis - 物体是随机绕垂直 Z 轴旋转
- Random axis - 物体是随机绕随机轴线旋转

默认值是 Z axis

渲染 > 旋转强度

系数限定随机旋转角的范围 - 从朝着摄像机的方向计数。Render > Randomize Initial Rotation 复选框勾选的时候被启用。例子：

- 旋转强度 = 1 - 角度将躺在内 $[-\pi, \pi]$ 范围
- 旋转强度 = 0.5 - 角度将躺在内 $[-0.5 \cdot \pi, 0.5 \cdot \pi]$ 范围
- 旋转强度 = 0.1 - 角度将躺在内 $[-0.1 \cdot \pi, 0.1 \cdot \pi]$ 范围

默认值是 1。

渲染 > 广告牌

启用对粒子的广告牌。默认情况下禁用。

渲染 > 广告牌类型

布告板类型。在 Render > Billboard 选项被启用时可用。三种类型可供选择：

- Basic - 简单片面地产生广告牌：粒子将与他们的前面转向观察者
- Random -Random - 随机双面产生广告牌：粒子将被更多地开启他们的前面或后面面向观察者更少的用边来对着观察者；也将有一个小的随机转；这种模式是专门为草实例化
- Jittered - Jittered - 片面产生广告牌沿着其转动到观察者的平面波浪粒子；这种模式是专为树叶的实例化

默认值是 Basic。

Render > Jitter Amplitude

系数限定了粒子振荡幅度。由 Render > Billboard Type 菜单中选择 Jittered 类型时可用。此参数是越大，越大的振幅。缺省值是 0。

Render > Jitter Frequency

粒子振荡频率单位赫兹。由 Render > Billboard Type 菜单中选择 Jittered 类型时可用。缺省值是 0。

Render > Billboard Geometry

广告牌旋转方式 (Render > Billboard 复选框勾选时该选项可用)。两种类型可供选择：

- Spherical - 球形广告牌即粒子完全面向观察者和他们的旋转是无限的;
- Cylindrical - 圆柱形广告牌即粒子只绕垂直 Z 轴旋转;

默认值是 Spherical。

动态草设置

Dynamic Grass

此选项可启用动态小草渲染模式。默认情况下禁用。

Dynamic Grass > Scale Threshold

最小尺寸为动态的草粒子。更小的粒子也不会被渲染。如果启用了 Dynamic Grass 的选项。

继承设置

属性继承 > 风弯曲

粒子继承了 Wind Bending 设置：

- Parent - 从发射器继承
- Instance - 从粒子物体本身继承

默认值是 Parent。

属性继承 > 阴影

通过粒子继承阴影设置：

- Parent - 从发射器继承
- Instance - 从粒子物体本身继承

默认值是 Parent。

属性继承 > 反射

通过粒子继承反射设置：

- Parent - 从发射器继承
- Instance - 从粒子物体本身继承

默认值是 Parent。

属性继承 > 顶点颜色

继承发射器顶点颜色。包含两个字段：

- From - 发射器的现有顶点颜色名称
- To - 粒子现有的顶点颜色名称

默认没有继承。

草地

物体的实例化可用于可视化各种草。在这种情况下，草在相机在地形上移动时，相机靠近的时候渲染。



激活

1. 在一个独立的平面物体创建一个粒子系统再物体实例化。启用 Dynamic Grass 选项。
2. 为所谓的地形材质启用 Terrain Dynamic Grass 选项。

设置

建议创建具有少量的平面（例如 3）对应于所需草的级联（例如 100, 150 和 250 米）的尺寸。

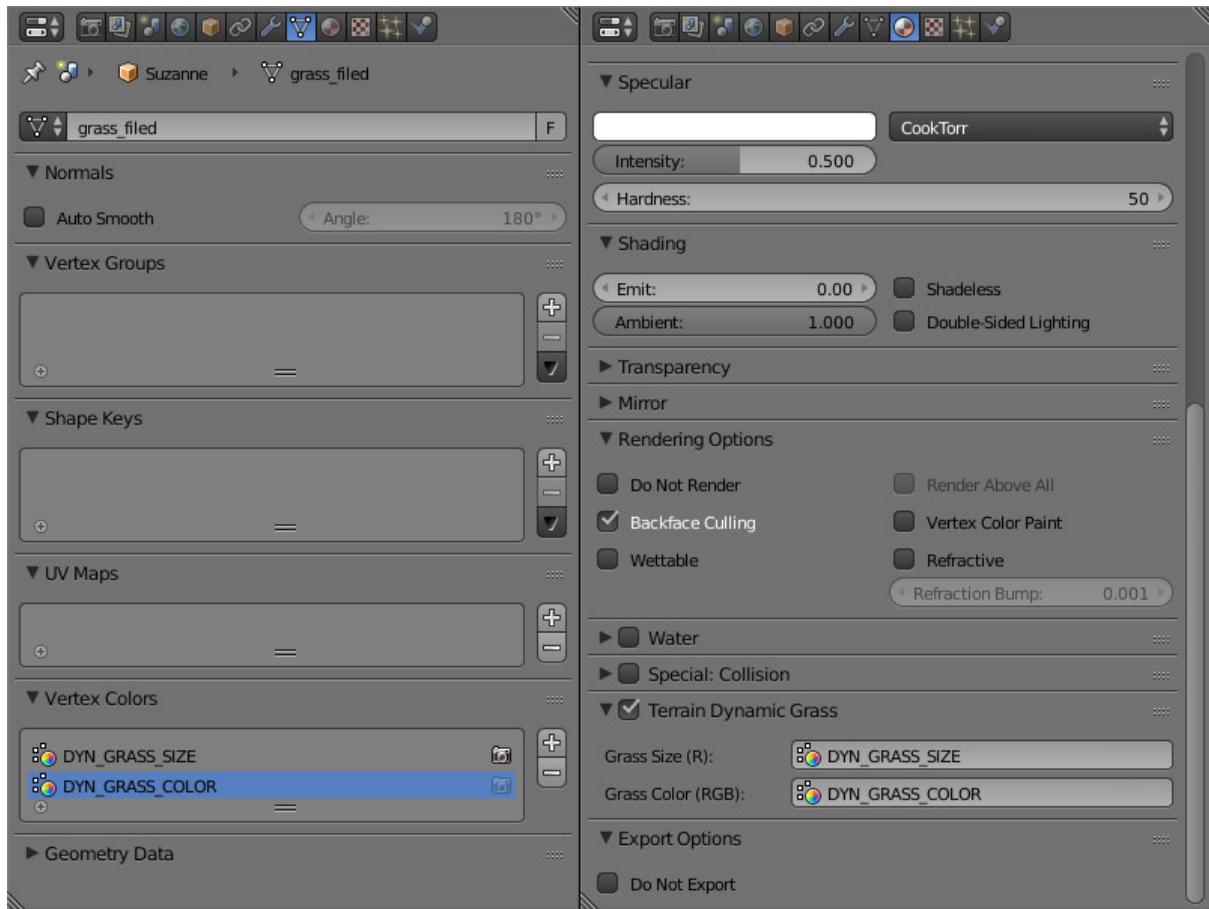
对于景观的 **材质**，当 Terrain Dynamic Grass 选项启用时，以下文本字段被激活：

动态草尺寸 (R) 景观网格的顶点颜色图层名称其目的是为修改草大小。草的大小（即高度）由灰色色调定义 - 亮颜色草越高。

动态彩色草 (RGB) 地形网格的顶点色层的名字是用作草的着色。顶点颜色与草材质颜色相乘。为草材质的漫反射纹理的 Influence > Blend 参数应该有 Multiply 值。

用这样的名字顶点颜色层应在景观网格存在。

此外，还建议禁用发射器（的 Render > Emitter 选项）的渲染。



树叶

实例化适合很好的渲染树的叶子，获得较好的层级显示。



激活

在 Particle system setup -> Activation 部分描述执行（见上文）。在这种情况下，树是发射器然后叶子和小分支是粒子。

此外，以下的操作可用于发射器来执行：

- 创建顶点组，其包括在其上的粒子将被放置的顶点
- 创建树的风弯曲参数和叶子顶点着色层
- 创建顶点着色层由粒子继承（例如它可以用于着色的粒子）

设置

1. 随机旋转设置

如果 初始随机 Rotation 复选框被启用，则建议选择垂直轴随机旋转 - Z axis （通过使用 Rotation Type 菜单）。该 旋转强度值可随意设定。

2. 布告板设置

建议启用广告牌，设置其类型为 Jittered（通过使用 Render > Billboard Type 菜单），并使其球形 - Spherical（通过使用 渲染 > 广告牌 Geometry 菜单）。该 渲染 > 抖动 Amplitude 和 渲染 > 抖动 Frequency 值可随意设定。

3. 粒子位置设置

建议选择从 Emission > Emit From 菜单中的 Verts 值，并选择发射器的顶点组（在 Vertex Group > Density 字段）定义粒子的位置。请注意，在 Render > Randomize Location and Size 复选框应禁用。

4. 风效果设置

建议从发射器启用继承设置 - 在 Properties Inheritance > Wind Bending 菜单选择 Parent。在 Object 面板为发射器启用 Wind Bending 再设置弯曲参数。对一颗树指定 Wind Bending > Main Bending > Angle 和 Wind Bending > Main Bending > Frequency 参数就够了，在 Wind Bending > Main Bending > Main Stiffness 指定顶点色名称用来弯曲。

5. 顶点颜色继承设置

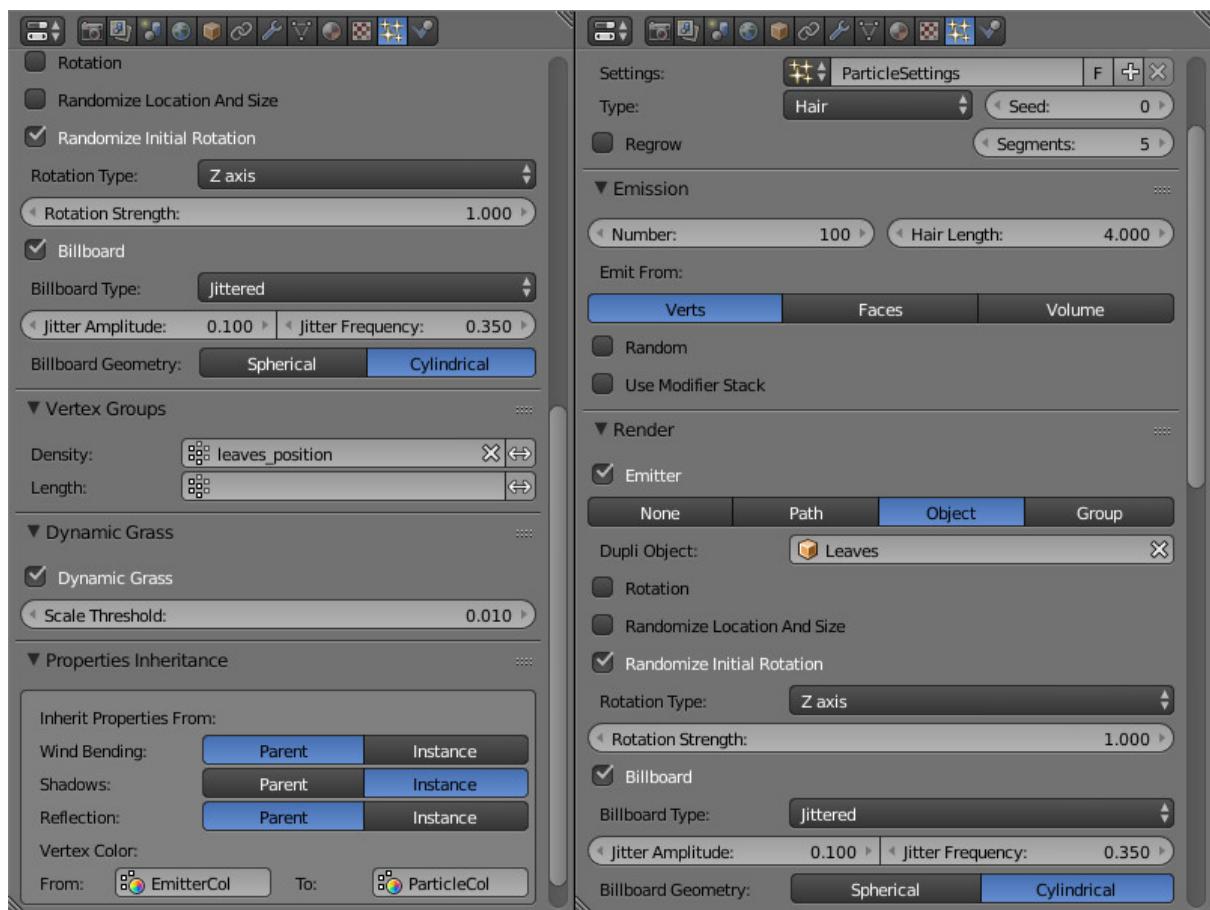
对于由粒子继承发射器的顶点颜色，则需要在分别在 Properties Inheritance > Vertex Color > From 和 Properties Inheritance > Vertex Color > To 同时指定发射器的顶点颜色名称。其结果是，发射器的顶点的颜色最接近的是粒子（在 From 字段指定）将被复制并传播到粒子的 To 顶点颜色层。

在 Properties Inheritance > Vertex Color > To 字段中指定的名称所产生的顶点颜色层所用的粒子的节点材质可用于它的着色和任何其他效果。

6. 通过顶点组权重设置粒子的大小

为了创建粒子和顶点组权重的大小之间的依赖，在 Vertex groups > Length 字段选择所需的顶点组的名称。

影响可通过选定的顶点组中设定的权重进行调整。



动画

目录

- 动画
 - 动画控制
 - 物体动画
 - 蒙皮和骨骼动画
 - * 烘焙参数
 - 顶点动画
 - * 烘焙参数
 - 默认动画
 - 非线性动画
 - * 非线性编辑器
 - * 通过 API 控制
 - 音频源参数
 - 值和 RGB 节点的动画

一般动画是物体的参数随时间变化。引擎支持以下类型的动画

- 物体动画意味着物体作为一个整体变换。
- 骨骼动画，即使用骨头让物体变形。一个单独的骨骼物体动画也支持（为父级到骨头）。
- 顶点动画。物体的变形可以被记录到帧，并在引擎然后重新生成。
- 音频源参数。扬声器的 音量和 音速可以进行动画处理。

- 节点材质的 值节点输出的动画。
- 风弯曲 - 程序性动画。单独在 [中描述](#)。
- 粒子发射。所述在[相应章节](#).

动画控制

有两种方法在引擎中控制动画 [\[F\]](#)

1. 自动，激活 动画面板并在物体的属性中选择 行为参数。在这种情况下将由引擎选择相应的动画方法，一个场景加载之后物体的动画就会开始播放。骨骼动画会播放在 Action Editor 窗口默认赋予物体的那个动作。
2. 在应用程序中通过 动画模块的 API 来工作。

使用 Animation 界面调整动画非常有用。已经放在[相应章节](#).

物体动画

可以进行动画处理的参数是中心坐标 (位置), 旋转和 缩放.



可以在 Blender 中为物体添加运动动画关键帧，然后在引擎中重现。

支持以下关键帧类型 [F](#)

- 位置
- 旋转 – Quaternion(WXYZ) 或者 XYZ Euler 模式是必需的。
- 缩放——要有正确的结果，比例因子在所有 3 轴应该是相同的。
- LocRot – 位置和 旋转的组合。
- LocScale – 位置和 缩放的组合。
- LocRotScale – 位置 旋转和 缩放的组合。
- RotScale – 旋转和 缩放的组合。

如果网格物体可进行动画，它需要在物体属性选项卡上的 Rendering Properties 面板激活 Force Dynamic Object 选项。

蒙皮和骨骼动画



骨骼动画同时需要一个网格物体和骨架物体。以下四个步骤应开展

1. 在骨架物体中创建该物体的“骨架”。
2. 分配好网格物体的顶点组并将它们链接到骨头。这可以由权重绘制完成。
3. 在姿态模式下将骨架物体中骨骼制作动画。可以使用物体动画相同的关键帧类型。

- 当使用反向动力学 (IK) 或其他非平常的结构时, 需要额外的步骤就是烘焙动画 (在 Blender 中 Action 数据块)。可以使用 Blend4Web 标签下的 Bake Skeletal Animation 接口执行烘焙 **F**



烘焙参数

烘焙需要骨架物体处于选中状态。

将需要烘焙的动作都放到列表里面。如果该列表为空, 烘焙所有的动作。

名字 在列表中将要烘焙的当前动作名字。

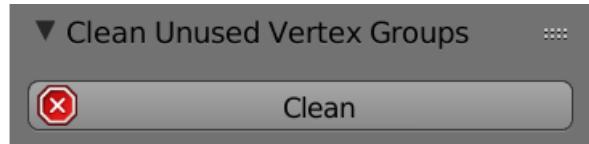
优化关键帧 烘焙后优化动画关键帧。引起不正确结果的情况下, 建议将此选项关闭。

使用 Blender 的原生烘焙 使用 Blender 的动画烘焙功能而不是 Blend4Web 的 (Bake Action 选项)。Blender 动画烘焙设置不同于 Blend4Web 的。

烘焙 执行烘焙。如果过程成功完成, 具有 B4W_BAKED_ACTOR_NAME 类型的动作名字会出现在场景。这些动作可以指派给骨架物体并在引擎中播放。把这些动作用在 Blender 本身是没有任何意义的。

注解: 引擎支持每个顶点达 4 个顶点组。如果顶点组数目超过 4 最有影响力的顶点组被选中。当场景加载的顶点权重归一化即它们的总和减少到 1。

要移除骨架不需要的顶点组, 请使用按钮 Remove Clean Unused Vertex Groups.

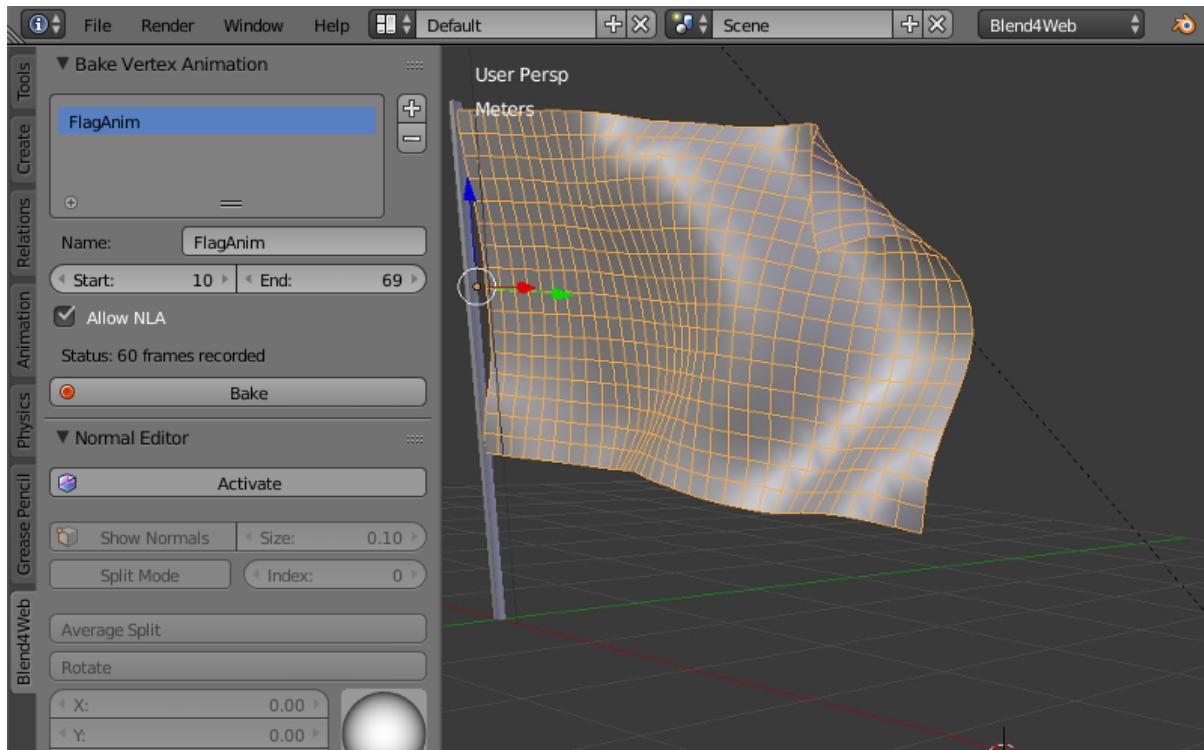


Blend4Web 才开始对骨架预算进行初步支持。目前, 唯一受支持的约束类型是 Copy Transform。它可以用于附加到物体, 产生影响, 如布娃娃的骨架。后续版本中将进一步添加支持其他类型的约束。

顶点动画



允许记录网格物体任何几何变化。请注意，每个顶点动画帧都是作为一个网格。建议不要用高模网格做长动画，源和导出的文件的大小会非常大，并且还会减慢引擎的工作。一个特殊的工具用于烘焙顶点动画 - Bake Vertex Animation - 在 Blend4Web 工具面板上。



烘焙参数

名字 将分配给烘焙动画的名称。

起始 烘焙从这帧开始。

结束 烘焙将在这一帧结束。

允许 NLA 允许使用[NLA](#) 来控制烘焙的动画。

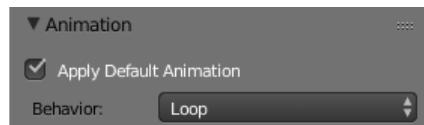
状态 在此字符串里，显示的已经烘焙动画的帧数。如果烘焙并没有执行，则会显示 Empty 一词。

烘焙 按下此按钮执行动画烘焙。

注解: 要让顶点动画正常工作，应启用所选物体的 Export Vertex Animation 参数 (Object 面板的 Export Options 部分)。

默认动画

Blend4Web 也有可以自动播放物体动画的选项。要做到这一点，您需要启用要进行动画处理的物体的 Animation 部分的 Apply Default Animation 参数。支持骨骼和物体的动画类型。



应用程序启动后动画就开始播放，和Play Timeline 节点非常相似，只是没有选项来设置开始和结束标记（相反，它总是从时间线第一帧开始和最后一个结束）。您还可以设置动画的行为，像Play Animation 节点。

非线性动画

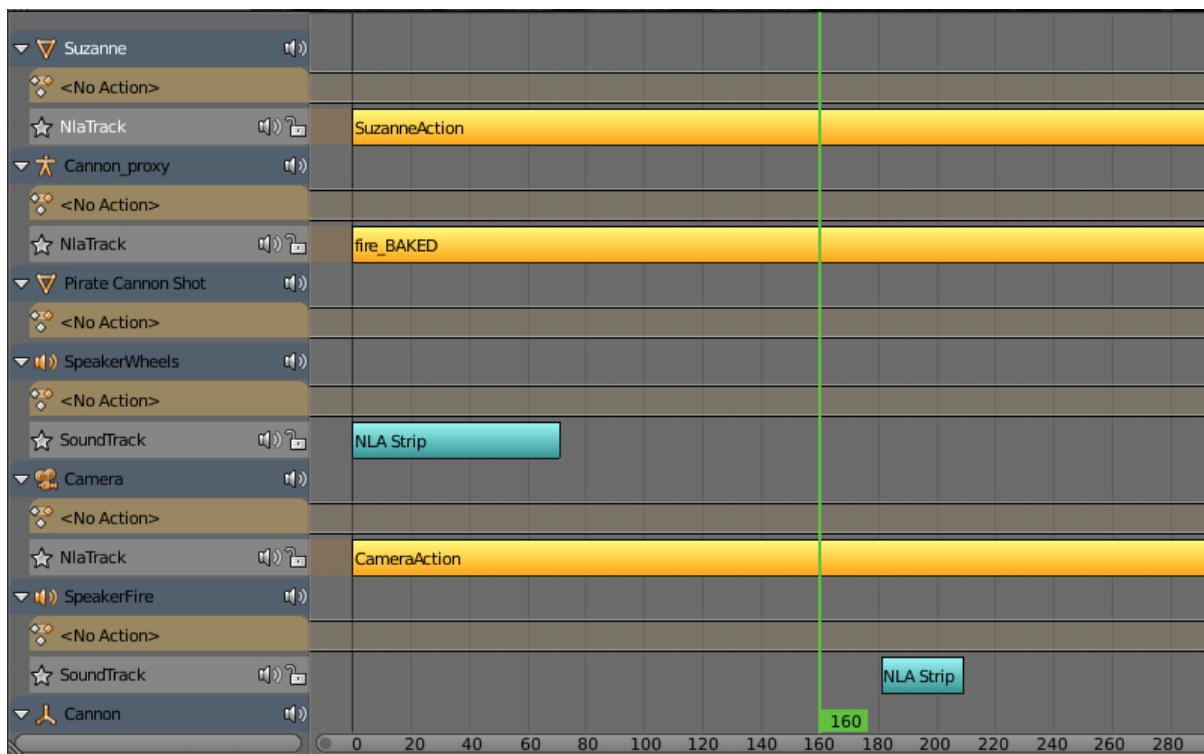
非线性编辑器

Blender 的非线形编辑让我们方便的设置场景的行为。在它的帮助下，我们可以实现简单的场景。这种方式下简单的场景和应用程序是不用编程的。



引擎支持控制以下实体**¶**

- 可以用动作表示的任何参数动画
- 声音播放
- 粒子发射（在与全局时间轴连接的形式）



用法

1. Scene 选项卡下的 NLA 面板激活。
2. 在 NLA Editor 设置场景所需的行为。
3. 选择时间轴面板上的动画时间间隔。

其他设置

NLA > Cyclic NLA 场景设置激活循环 NLA 动画模式。

注解: 为了使用顶点动画, 在[vertex animation panel](#) 上启用“Allow NLA”选项。

限制条件

- 不支持同一物体不同类型的动画同时播放。

通过 API 控制

可以通过 API 方法的‘nla.js’模块控制非线性动画播放。

```
// ...
var m_nla = require("nla");
// ...
m_nla.set_frame(150);
// ...
var frame = m_nla.get_frame();
// ...
m_nla.play();
// ...
m_nla.stop();
// ...
```

请注意, 如果Logic Editor 已经使用用, set_frame, play, stop 方法不可用。

音频源参数

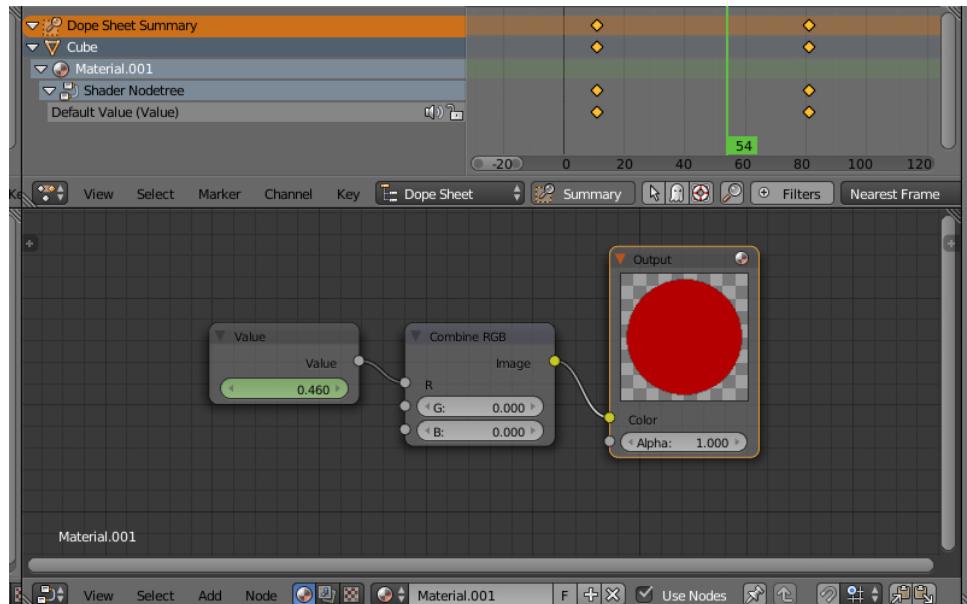
此外对于扬声器物体支持以下动画关键类型 

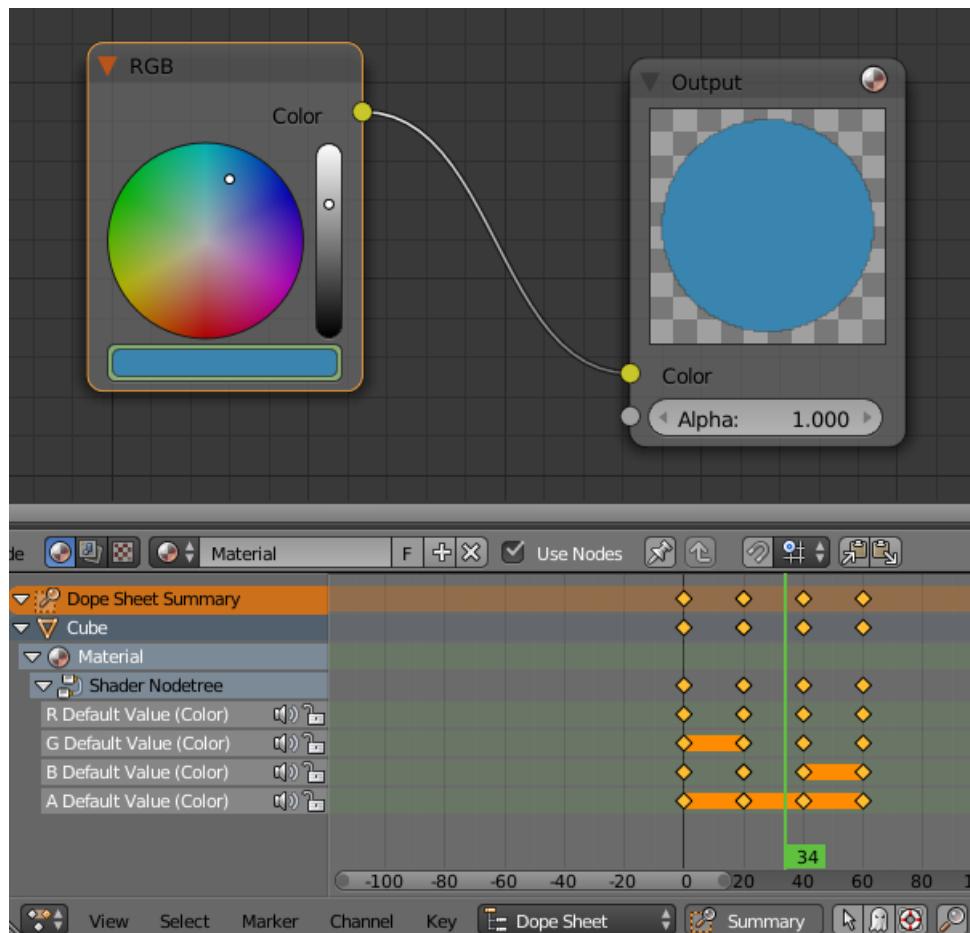
- 音量
- 音速

音频源参数在本质上遵循物体动画。

值和 RGB 节点的动画

支持节点材质中插入关键帧到 Value 和 RGB 节点的播放。





注解: 其他节点上的数值和颜色值的动画不受支持。

也可以在[非线性动画编辑器](#) 中创建动画轨道。每个材质支持多个 Value 或 RGB 节点动画。这些节点的值可以也通过 API 修改，使用 objects 模块的 set_nodemat_value 和 set_nodemat_rgb 方法。

参见:

[时间 \(B4W_TIME\)](#)

室外效果

目录

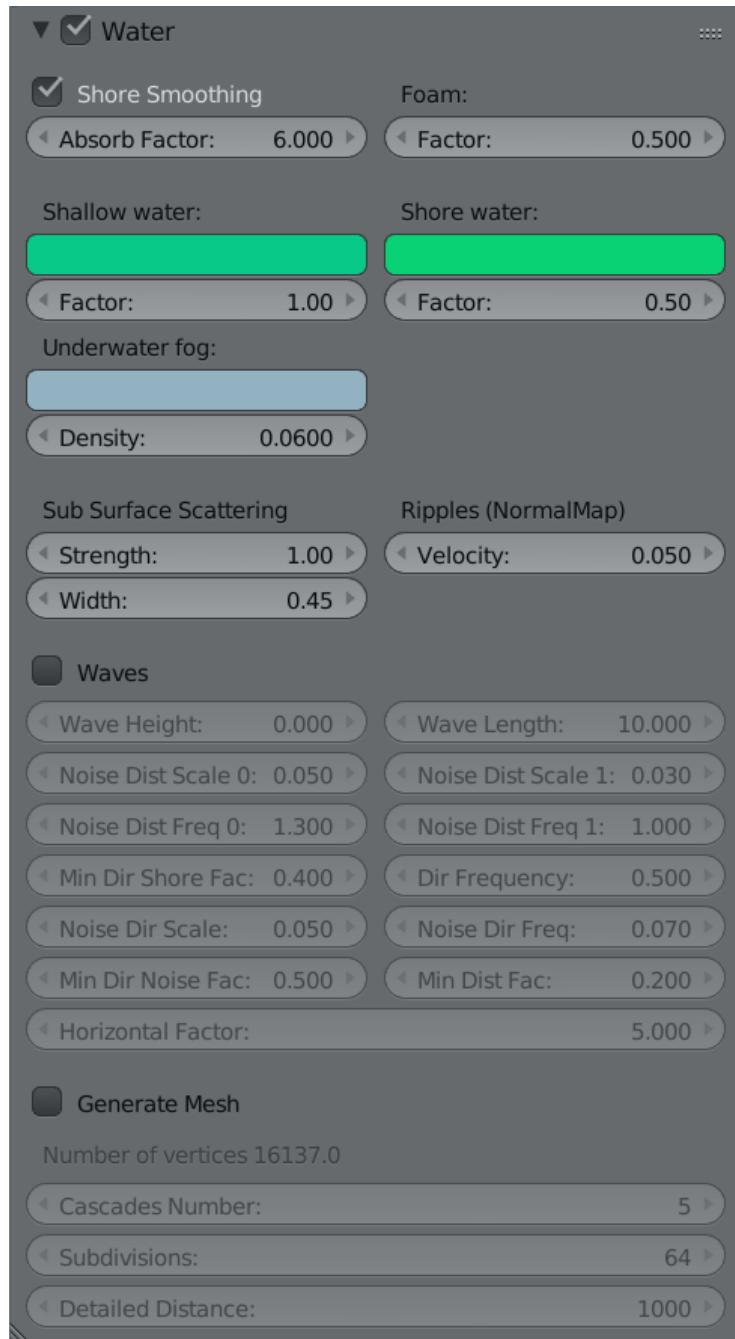
- 室外效果
 - 水
 - * 激活
 - * 基本设置
 - * 波浪运动
 - * 表面润湿
 - * 反射与菲涅尔效应
 - * 平滑的海岸线
 - * 颜色渐变
 - * 折射
 - * 泡沫
 - * 焦散和色差
 - * 水下环境
 - * 体积波
 - * 地表生成设置
 - 大气
 - * 散射
 - * 雾
 - * 当日时间
 - * 星星
 - 镜头光晕
 - * 镜头光晕设置
 - * 镜头光晕物体
 - 风
 - * 激活
 - * 设置
 - * 草和树叶的动画

Blend4Web 有几个环境的效果，对创建室外场景非常有用。

水

激活

对于所谓水材质激活 Material 选项卡下的 Water 面板选项。



基本设置

透明 推荐使用渐变透明度（设置 Transparency > Type 参数的 Alpha Blend 值）和调整的 Alpha 值。

灯光参数 水材质照明参数可如[灯光参数](#)章节所描述的进行设置。

波浪运动

水上的波纹通过法线贴图的动画 UV（从 0 到 4 个）模拟。对于法线贴图使用了唯一共享的图像 - 纹理只能由 Mapping > Size 参数不同来调节。水网格必须有一个 UV 层。

表面润湿

是自动进行的。要激活效果打开 Rendering Options 面板 Wettable 选项上所需要的材质。

反射与菲涅尔效应

水材质支持静态和动态的反射以及菲涅耳效应。请参阅：[参考反射](#)章节.



平滑的海岸线

效果影响海岸线附近水 - 它变得更加透明。

水 > 海岸平滑 启用平滑。

水 > 吸收因子 这是水的吸光系数。越高它是更透明的水。这个值可以从 0 到 100 的变化。默认值是 6。

颜色渐变

对于颜色渐变的水材质必须具有 Export Options > Shore Distance Map 选项启用。该纹理可以通过[烘焙海岸线参数](#) 用脚本生成.

浅水 > 颜色 设置浅水的颜色。缺省情况下，蓝 (0.008, 0.222, 0.8) 被使用。

浅水 > 因子 设置浅层水的颜色混合的因子。这个值可以从 0 到 2 变化，默认设置为 1。

滨水 > 颜色 这个决定在岸边水的颜色。缺省情况下，深蓝色 (0.003, 0.19, 0.57) 被使用。

滨水 > 因子 设置水在靠近海岸线的颜色混合的因子。这个值可以从 0 到 2 变化，默认设定为 0.5。

折射

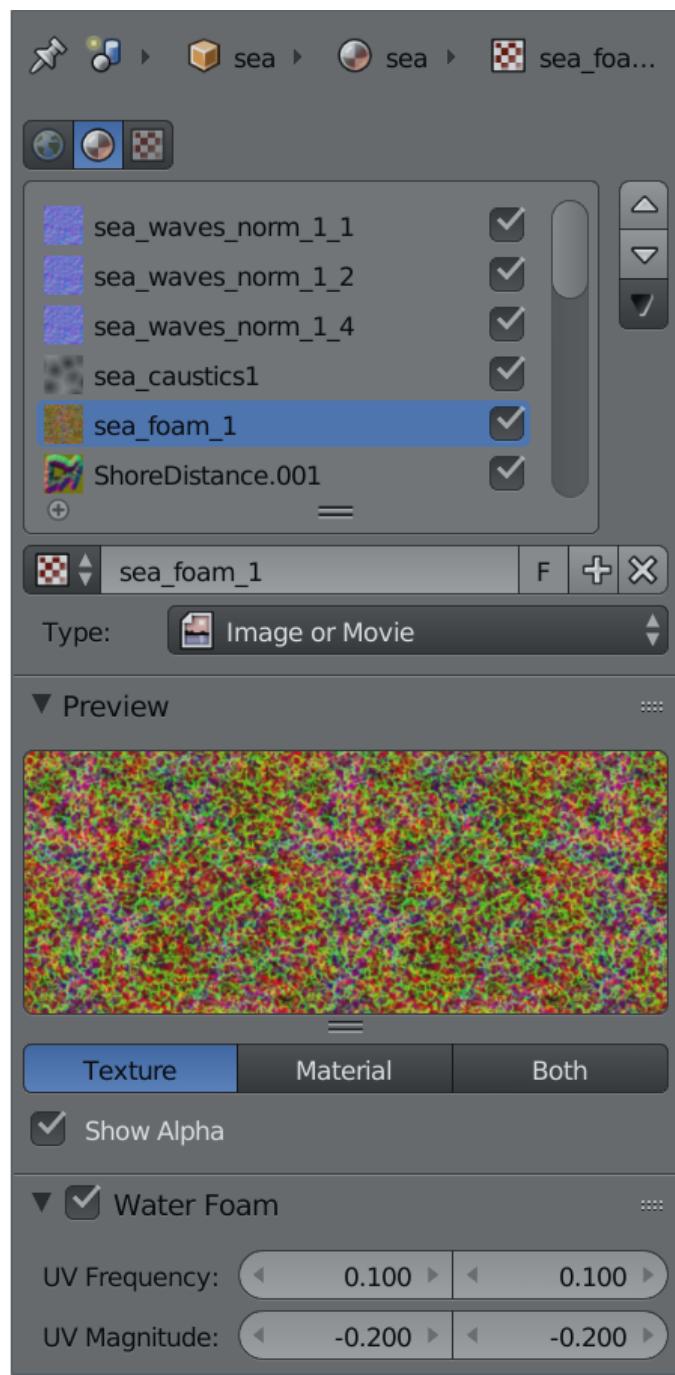
在 Reflections and Refractions 面板 Render 选项卡中设置 Refractions 选项 ON 或 AUTO。



泡沫

激活

要生成泡沫就要在水材质槽中加入一个漫射纹理。这个图像中的每个 RGB 通道必须包含有 BW 的泡沫纹理。然后在 Water Foam 面板启用。



设置纹理

水泡沫 > UV 频率 动画 UV 坐标的振荡频率。默认值是 (1.0, 1.0)。

水泡沫 > UV 幅度 动画 UV 坐标的振荡幅度。默认值是 (1.0, 1.0)。

建立材质

泡沫 > 因子 一般泡沫体的影响因子。默认值是 0.5。

焦散和色差

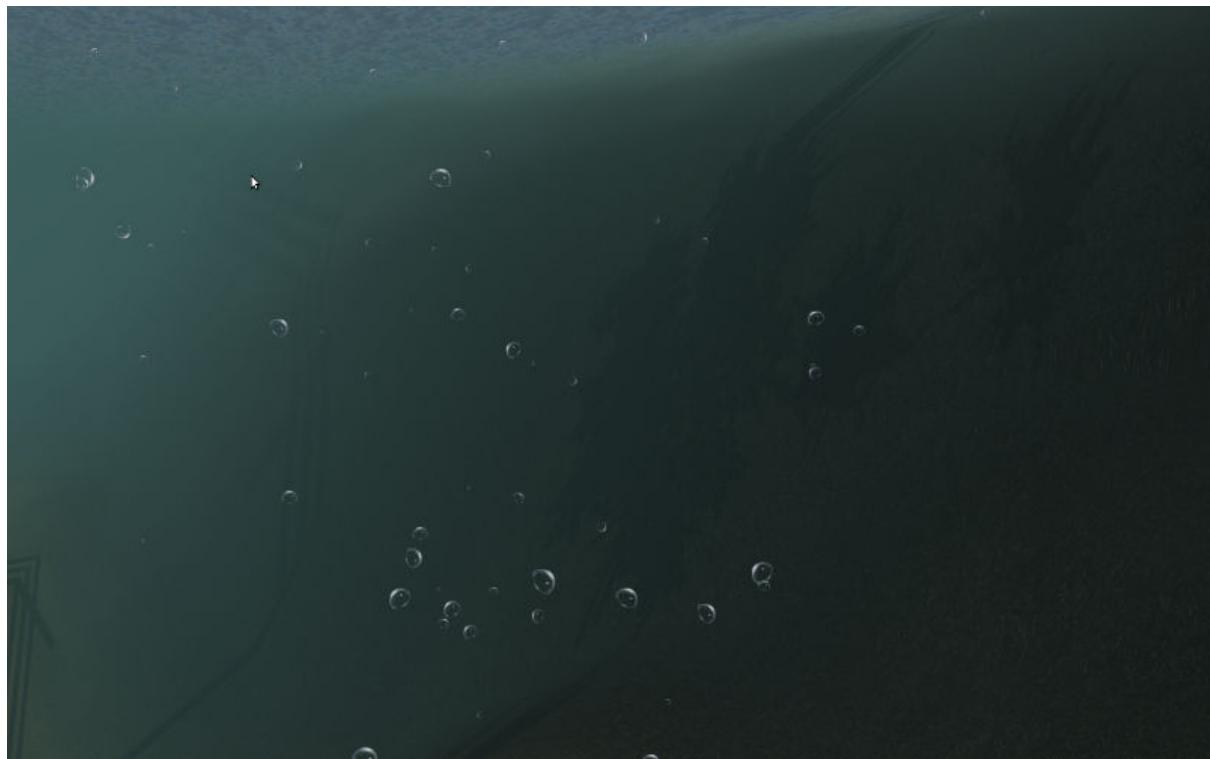
要建立焦散效果开启水材质的 Caustics 选项。另外，必须有至少一个 Sun 在场景中。



缩放 程序纹理的单元大小。默认值是 0.25。

亮度 焦散影响因子。默认值是 0.5。

水下环境



可见性设置 (“雾”)

水下雾 > 颜色 雾的颜色。默认值是 (0.4, 0.6, 0.7)。

水下雾 > 密度 其影响密度和能见距离指数因子。默认值是 0.06。

上帝光芒 效果设置也可用。

注解: 想要正确水面渲染效果必须关闭 Rendering Options > Backface Culling 选项。



体积波

激活

为了使用程序浪 Waves 选项必须打开。

注解: 程序波的方向是从场景中的 Wind 力场的方向一致。

如果场景没有 Wind 力场，程序浪将仅在不能改变的默认的方向移动。



设置

浪高 波高度可变化从 0 到 5 的默认值是 1.0。

波长度 波长可以从 0.01 至 200 变化， 默认值是 10.0。

噪音 DIST 缩放 0 这是开放水浪的第一组分的大小。这个参数可以从 0 到 1 变化， 默认值是 0.05。

噪音 DIST 缩放 1 这是开放的水浪的第二成分的大小。这个参数可以从 0 到 1 变化， 默认值是 0.03。

噪声 DIST 频率 0 这将设置开放式水浪的第一部分的频率。这个参数可以从 0 到 10 变化， 默认值是 1.3。

噪声 DIST 频率 1 这将设置开放水波的第二成分的频率。这个参数可以从 0 到 10 变化，并且默认设置为 1。

最小 Dir 河岸因子 这是在岸边波的最小高度减少系数。这个参数可以从 0 到 1 变化，并且默认设定为 0.4。

Dir 频率 设置靠着岸边的波浪翻滚的频率。这个参数可以从 0 到 10 变化，并且默认设定为 0.5。

噪音 Dir 缩放 设置岸边海浪的噪音大小。这个参数可以从 0 到 1 变化，并且默认设置为 0.05。

噪音 Dir 频率 设置岸波的噪音频率。这个参数可以变化从 0 到 1 的默认值是 0.07。

最小 Dir 噪声因子 岸边波浪的最小噪音。这个参数可以从 0 到 1 变化，默认值是 0.5。

最小 Dist 因子 这台混合打开水浪的最低系数。这个参数可以从 0 到 1 变化，默认值是 0.2。

水平因子 该系数表明了岸波向海岸线方向移动的程度。这个参数可以从 0 到 10 变化，
默认值是 5。

地表生成设置

生成网格 这使得产生的表面。

级联数据 这个描述所产生的表面上的级联的数量。这个参数的值可以从 1 至 20 变化，
默认值是 5。

细分 这是生成的网格的细分数目。默认值是 64。此参数是最低值是 2，而最高的可能
值是 512。

详细距离 这指定从相机到最后级联的边的最大距离。这个参数的值可以从 1 至 5000 变
化， 默认值是 1000。

烘焙海岸线数据到纹理

在 Blend4Web 选项卡下的工具面板（快捷键 “T”）打开 Bake Shore Distance Map 面板。
设定参数：到岸最大距离（最大 Distance），以及最终纹理大小（纹理 Size）。选择一个地形物体（或多个物体），然后 - 水物体。点击 Bake 按钮。

根据所处理的网格中的纹理大小和顶点的数量，脚本的执行时间可能会有所不同从几
秒到几分钟。确保水网格的纹理名 ShoreDistance 已经创建。

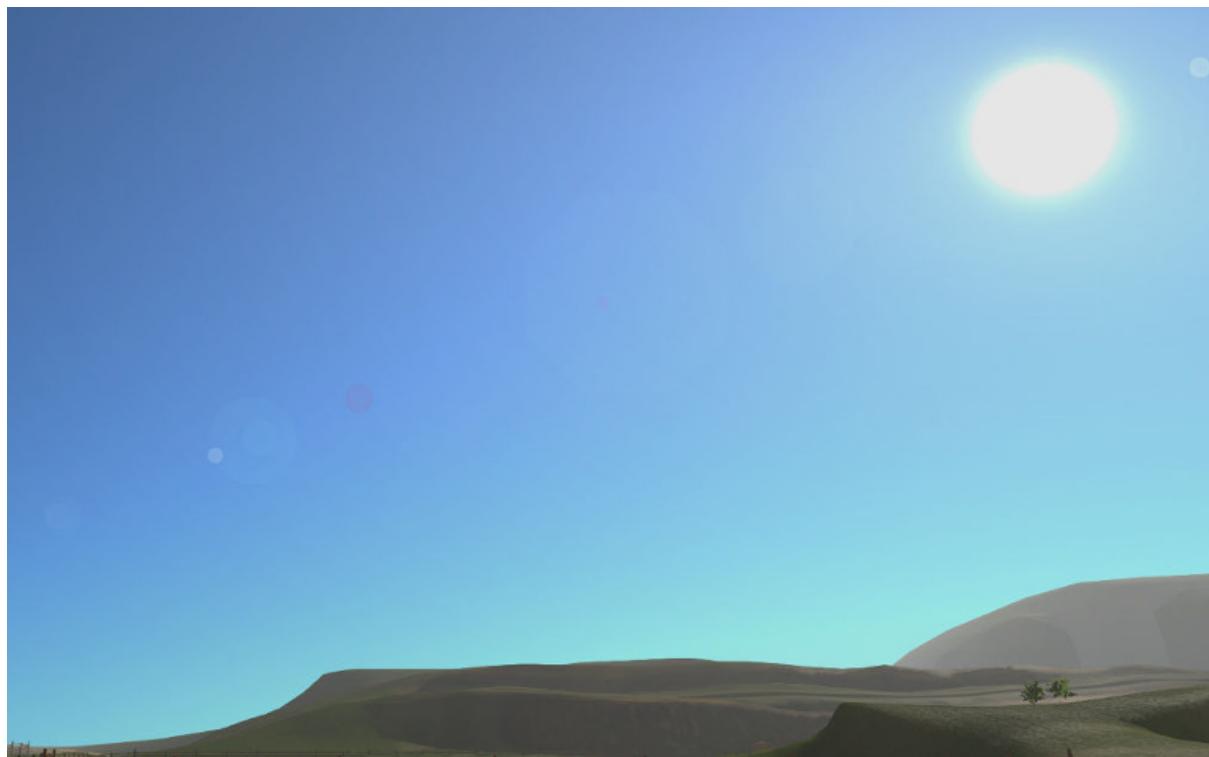
脚本执行一些系统属性保存在水材质。因此，该脚本完成工作后，必须保存场景。

大气

散射

启用 World > Render Sky, 然后激活世界标签下 Procedural Sky 面板。请注意，如果一静态天穹材质 同时正在被使用，它将被替换。

注解: 此外，一个程序的天空纹理可以用来模仿散射环境灯光 类似于static skydome texture。要做到这一点，启用 Procedural Sky > Use as Environment Lighting and Environment Lighting > Sky Texture 选项。如果世界纹理的环境照明已经存在，它会被替换。



支持设置：

程序天空 > 天空颜色 基础天空的颜色。默认值是 (0.087, 0.255, 0.6) (蓝色)。

程序天空 > 瑞利亮度 瑞利散射亮度（即小颗粒散射）。默认值是 3.3。

程序天空 > 三重亮度 Mie 散射亮度（即大颗粒散射）。缺省值为 0.1。

程序天空 > 曝光亮度 Sun SPOT 的亮度。默认值是 20.0。

程序天空 > 散射强度 光散射系数。默认值是 0.2。

程序天空 > 瑞利强度 瑞利散射系数。默认值是 0.2。

程序天空 > 三重强度 米氏散射系数。默认值是 0.006。

程序天空 > 瑞利收集强度 瑞利散射指数。默认值是 0.35。

程序天空 > mie 收集强度 米氏散射指数。默认值是 0.5。

程序天空 >Mie 分布 米氏散射分布。默认值是 0.4。

雾

该引擎支持位于 World > Mist 面板上的标准参数。除此之外，可能覆盖雾的颜色。

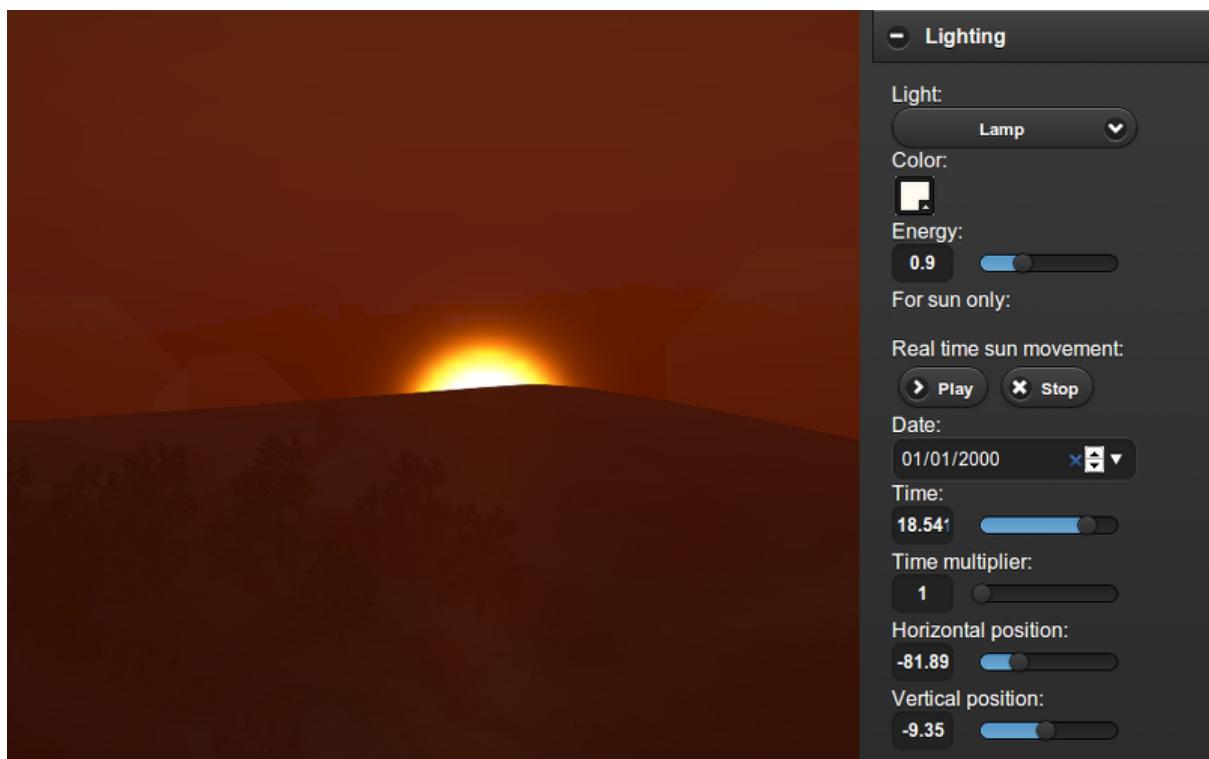
Fog settings are described in the [Scene Settings](#) chapter.

当使用动态天穹，雾的颜色是由天空的颜色定义。

当日时间

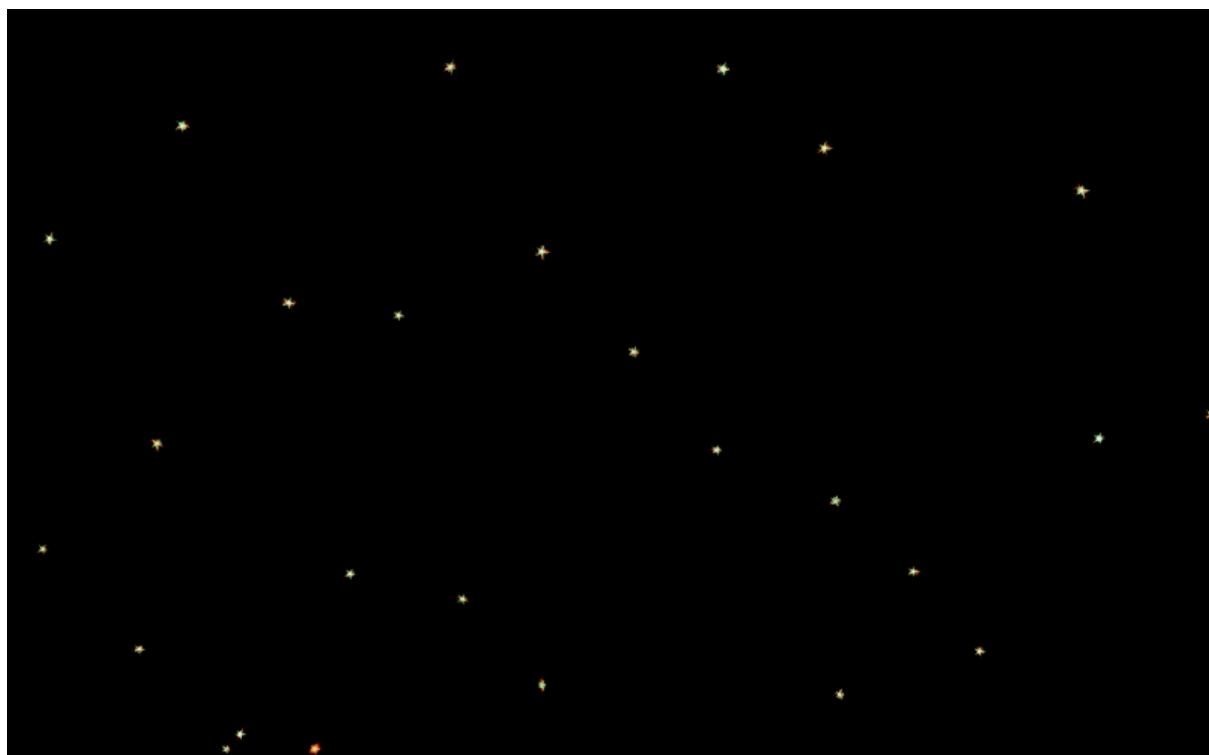
启用灯的 Dynamic Intensity 选项。

一天中的时间可以在应用程序通过 API 进行设置。一天中的时间可以用[场景查看器](#) 的 Lighting 界面来设置。



星星

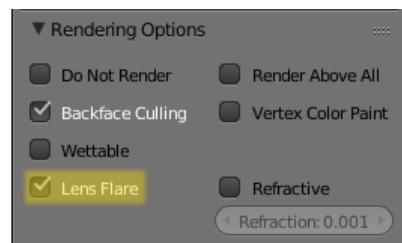
星的设置描述在[光晕材质](#)



镜头光晕



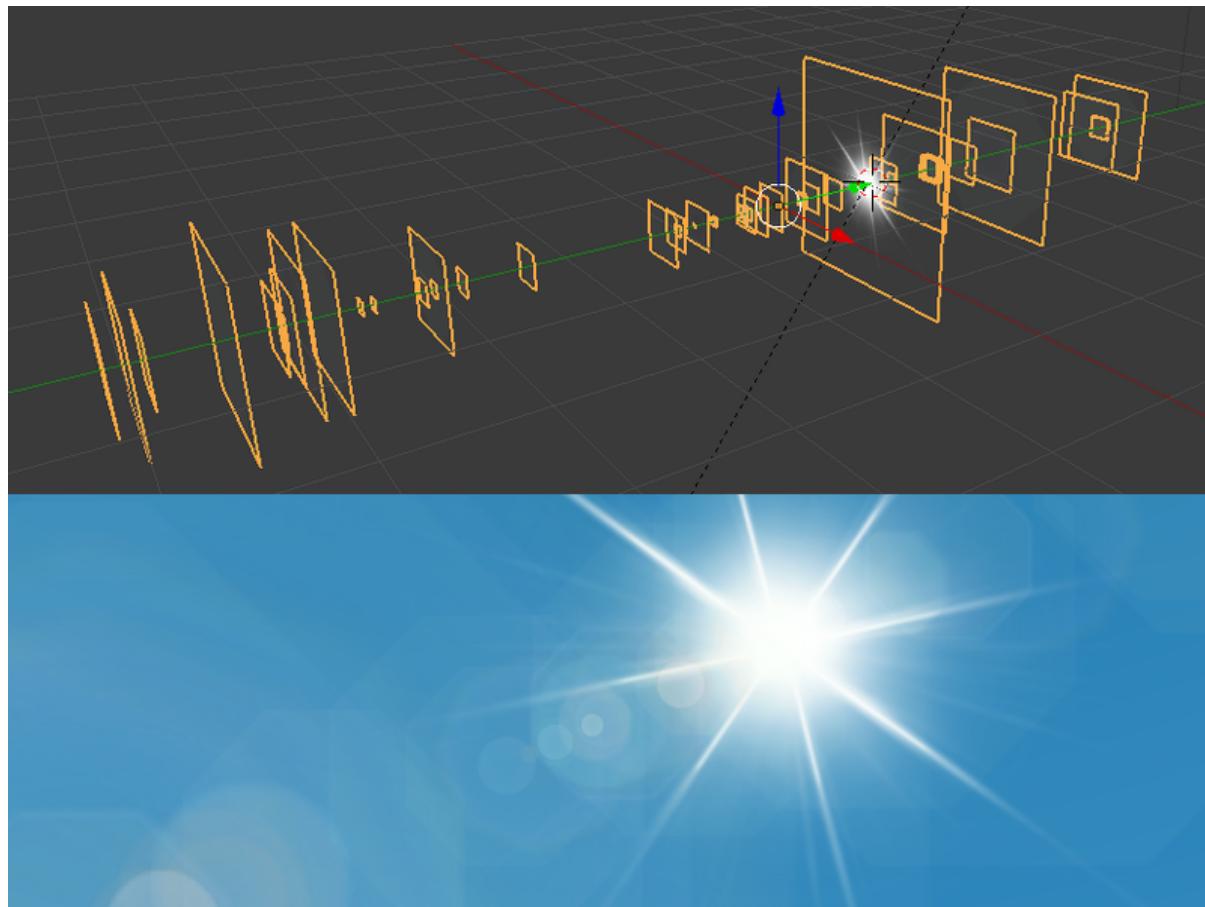
此功能可以通过 Lens Flare 位于 Material 面板的 渲染 Options 复选框启用。



注解: 只有场景中存在 Sun 型光源才会有效果。

镜头光晕设置

镜头光晕物体



镜头眩光物体是用于实际渲染的效果。从本质上说，这样的一个物体由几个平面与每个平面包含一个耀斑元素。每一平面的法线矢量应指向“-Y”方向。每一个元素都可以在屏幕上滑动和对相机的旋转，唯一的例外是中央平面（明亮的白色的“星”在上面的图片）。这个平面作为镜头眩光物体的中心，应该有一个本地的 Y 坐标设置为 1。

在渲染过程中，物体本身的位置和空间方向没有考虑。

镜头光晕物体要正常工作，Disable Frustum Culling 选项应该被激活。

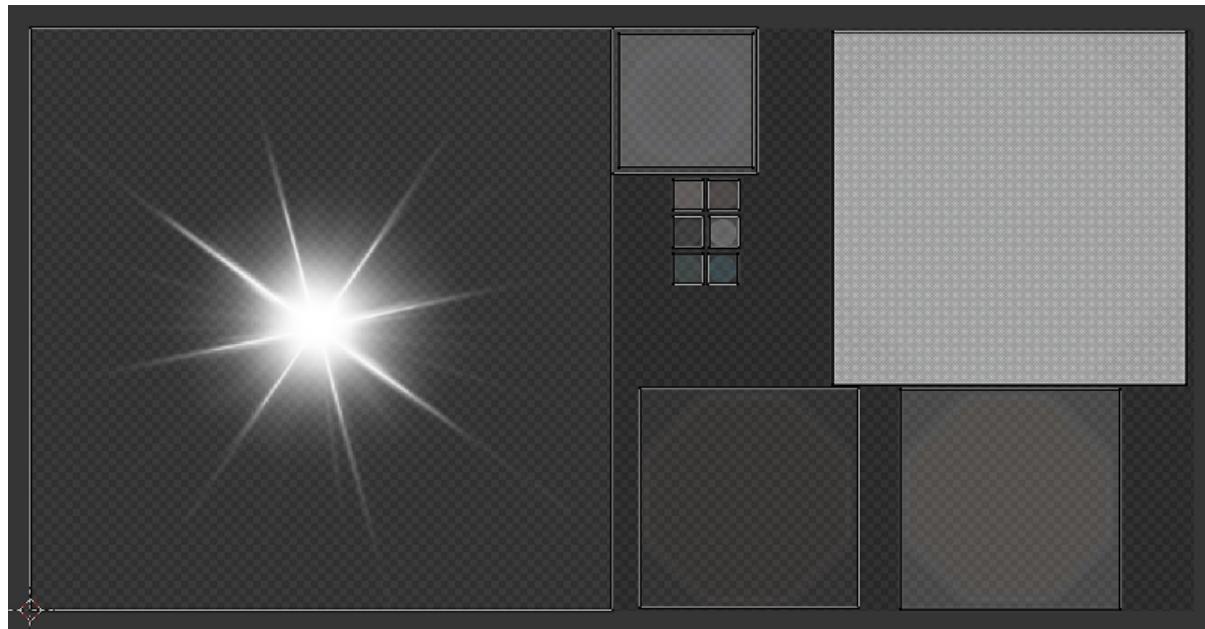
这样的物体的一个例子可以看到，例如，在我们的 [岛](#)演示。

材质设置

用于镜头光斑物体的材质应当具有 Alpha Blend 透明度类型。Alpha 参数应当被设置为零。

节点材质 不被支持。

纹理设置



用来产生镜头光斑效果的纹理的例子。

用于镜头眩光物体纹理应包含在实际的应用程序的用于形成光斑的形状所有元素。纹理图像应该使用 Alpha 通道对 Color 和 Alpha 值都有影响。

风

风的强度和方向的影响：

- 草和树叶的动画
- 粒子系统动力学
- 水浪翻滚频率 (现在只考虑强度)

激活

添加 Wind 类型的力场物体。

设置

方向 方向可以通过旋转力场物体进行设定。

力场 > 强度 风力强度。位于 Physics 选项卡下。缺省值为 1.0。

草和树叶的动画

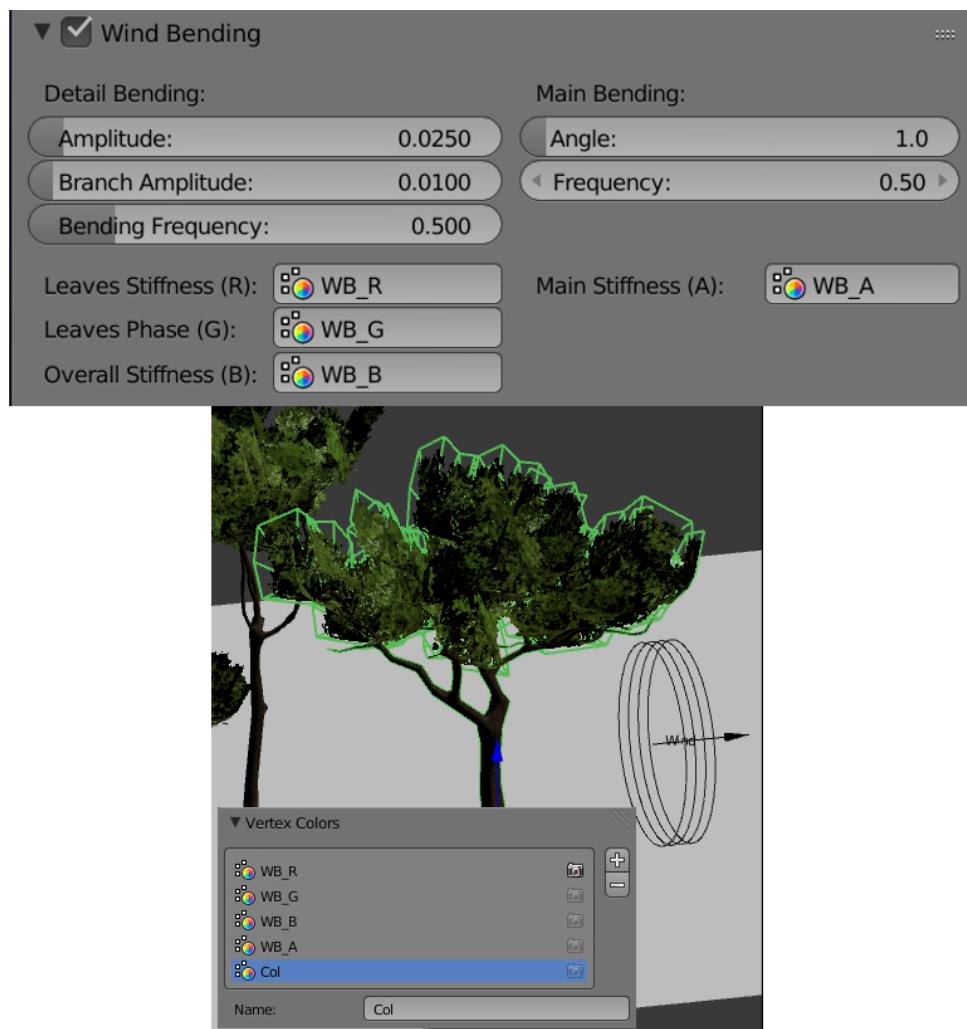
[草地](#) 章节讲述了草渲染创作需要的资源。

激活

在 Wind Bending 面板为草或树物题启用。

设置

打开 Wind Bending 面板后设置面板可见。



主要折弯 > 角 风的影响（度）下的“主”角偏差幅度。默认值是 10.0。

主要弯曲 > 频率 风的影响下，“主”偏差的频率。默认值是 0.25。

主要弯曲 > 主刚度 (A) 指定其中包含有关“主”偏差的刚度信息的顶点颜色层的名称文本字段。可以留空。

详细弯曲 > 振幅 在“细节”偏差造成的风的影响（度）的角度幅度。缺省值为 0.1。

详细折弯 > 分公司振幅 由风造成的影响（度）的分支偏差的角度幅度。默认值是 0.3。

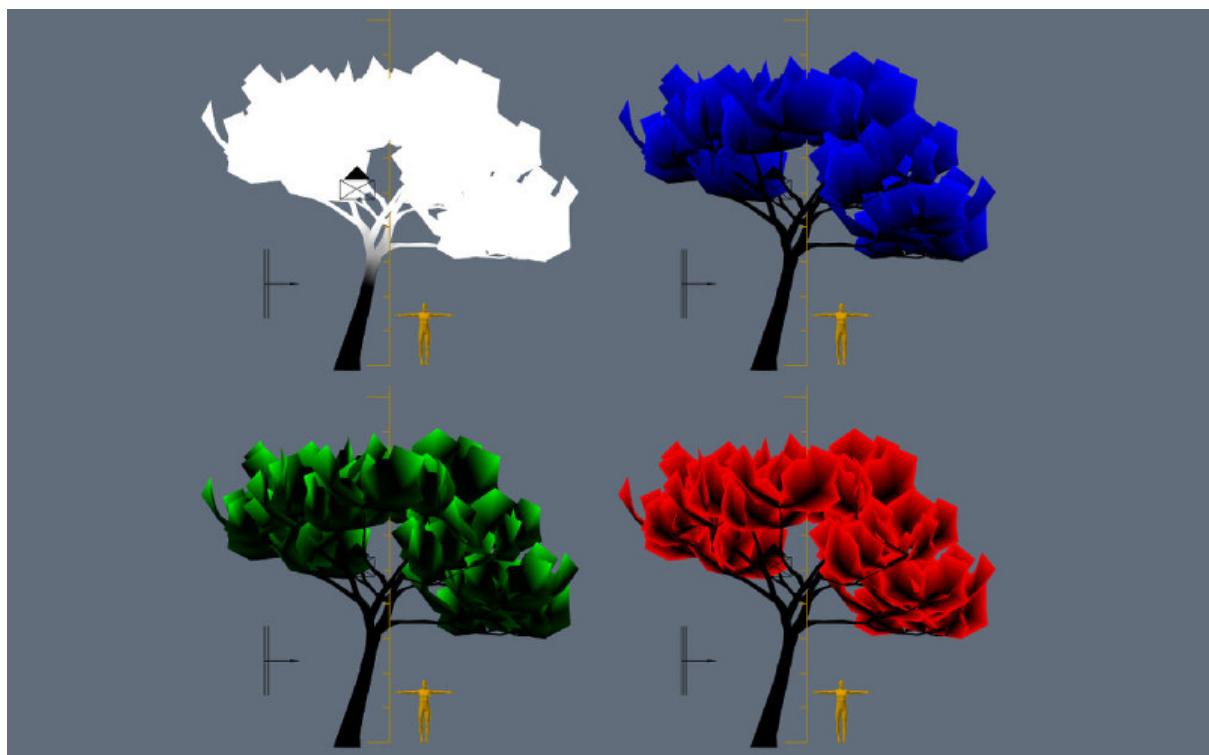
详细弯曲 > 弯曲次数 弯曲频率的细节。默认值是 1.0。

详细折弯 > 叶刚度 (R) 指定其中包含有关叶子的刚度信息的顶点颜色层的名称文本字段。可以留空。

详细折弯 > 叶阶段 (G) 指定其中包含有关叶偏差的相位信息的顶点颜色层的名称文本字段。可以留空。

详细折弯 > 整体刚度 (B) 指定其中包含有关叶片的整体刚度信息的顶点颜色层的名称文本字段。可以留空。

如果指定顶点着色层其名称应该是存在于该网格。



色彩管理

目录

- 色彩管理
 - Gamma 概述
 - 人类视觉和显示器
 - 伽玛公式
 - 伽马在节点材质中
 - * 为颜色的节点
 - * 为遮罩的节点
 - * 法线贴图
 - * 汇总表
 - alpha 合成
 - * 概述
 - * 实现
 - 色彩校正
 - * 激活
 - * 其他设置

Gamma 概述

伽马校正的实质是将图像亮度通道打包进 8 比特信息中。

标准（非 HDR）图像总是存储在较暗的部分比亮的那些部分使用更多的比特编码的非线性颜色空间。这意味着，较大的 RGB 值对应于实际的光强度（物理量称为照度）的

0.5 - 在简单的情况下该值等于 $0.5 ^ (1 / 2.2) = 0.73$ 。

否则，光强的 8 位的信息将不足以进行编码。这将导致深色调的不正确渲染。例如，深色渐变会显得加强。

因此，网络浏览器和其他软件一样，也是在非线性空间观看和处理图像。但是，3D 引擎和渲染器在线性空间的工作，这是表现现实世界中光的行为的唯一正确途径。例如，来自两个相同的灯的照度超过了从一个灯完全照度两倍。

无疑，在这种情况下 8 位的信息是不够的。这可以从一些真实光源的近似照度值示表中清楚地看到。

描述	照度, 光通量
夏季中午	17 000
冬季中午	5 000
日常天	1 000
在明亮房间	100
月圆夜	0.2
没有月亮的光	0.001

当 Color Management > Display Device > sRGB 选项为场景启用时，Blender 工作于线性空间。材质的颜色和灯设置对应到相应的物理值。对于纹理图像（除法线贴图），则需要选择 Image > Input Color Space > sRGB 选项。在这种情况下，在渲染的时候执行自动图象解包 (sRGB 的 -> 直线)。

人类视觉和显示器

尽管人的视觉是非线性（人识别较暗光色调比亮的要好），进入眼睛的光仍然服从物理定律（参照灯的例子）。

在 CRT 显示器的亮度是根据施加到显示器的输入非线性地施加到显示屏的电压（由在显存中的色彩通道值确定的电压）。液晶显示器模仿相同的特性。但是这类显示器发出的光服从物理定律。例如将第二光源到虚拟场景会引起亮度加倍（在完美的情况）。

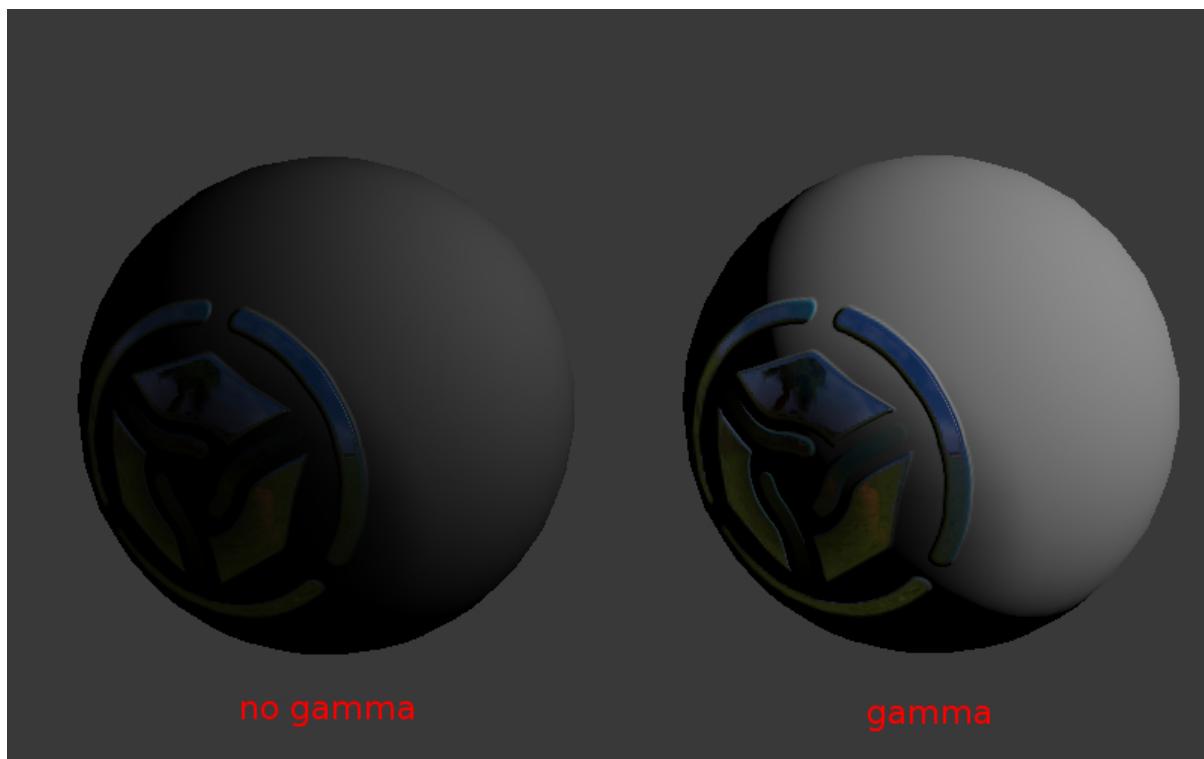
因此，人眼的感知特性是能够打包颜色通道的原因。同时，显示器的技术特征有二次伽马校正的意义。

伽玛公式

采用以下简化公式：

$$V_{\text{out}} = V_{\text{in}}^{\gamma}$$

$\gamma < 1$ - 打包伽马值, $\gamma > 1$ - 拆包伽马。在最简单的情况下 $1/2.2$ 和 2.2 的值被分别使用。在下文中, 用术语“打包”(Linear -> sRGB) 和“解包”(sRGB -> Linear) 的术语用来代替“伽马校正”。



伽马在节点材质中

为颜色的节点

当纹理和顶点颜色用于色彩 (不用于遮罩) 拆包 (sRGB -> 线性) 是必需的。纹理节点和几何节点的顶点颜色输出实现自动拆包。

需要注意的是纹理节点的 alpha 通道没有得到纠正。它的值是线性空间。

为遮罩的节点

纹理和顶点颜色可作为遮罩即对于一些数学运算的输入数据可以使用。在这样的情况下，需要打包操作。

请记住，一个纹理节点和 几何节点的 顶点色输出实现自动拆包。这导致了额外回到非线性空间中的转换，需要使用 LINEAR_TO_SRGB 或 $\gamma = 1 / 2.2$ GAMMA 节点。

注解：LINEAR_TO_SRGB 和 SRGB_TO_LINEAR 节点已被弃用。应使用 GAMMA 节点。

法线贴图

法线贴图无任何转换。

汇总表

使用案例	校正
纹理着色	在纹理节点自动执行 (Alpha 通道不校正)
纹理遮罩	GAMMA 值 $\gamma = 0.45$
顶点颜色着色	自动实现
作为遮罩顶点的颜色	GAMMA 值 $\gamma = 0.45$
法线	不需要

alpha 合成

概述

Physically correct alpha compositing is performed according to the formula [source]:

$$C_o = C_a \alpha_a + C_b \alpha_b (1 - \alpha_a).$$

这个方程和经典的混合操作（又名凸包含成）不同，因为它在第二被加数具有 α_b 乘数。因此渲染的时候不仅是原始像素的 α_a 值应该是已知的用于 alpha 合成，而且要叠合的

像素的 α_b 也是一样。

公式在预乘的情况下颜色通道的 α 值（称为预乘 alpha）就变成下面：

$$C_o = C_a + C_b(1 - \alpha_a).$$

最后一个公式用于计算出 α_o 值：

$$\alpha_o = \alpha_a + \alpha_b(1 - \alpha_a).$$

Preliminary multiplication of the color channels by the α values allows to save two multiplication operations. The more significant thing is that the derived formula can be used repeatedly without the need to divide the C_o color by the α_o value on each consequent iteration.

实现

在 Blend4Web 使用的混合功能如下：

```
gl.blendFunc(gl.ONE, gl.ONE_MINUS_SRC_ALPHA);
```

WebGL 的上下文初始化是使用 `premultipliedAlpha = true` 参数（即默认值）进行的。由所有颜色通道相乘 α 值在着色器进行输出。

色彩校正



激活

激活 Render 选项卡下的 颜色校正面板。

其他设置

亮度 默认值是 0.0。

对比度 默认值是 0.0。

曝光度 默认值是 1.0。

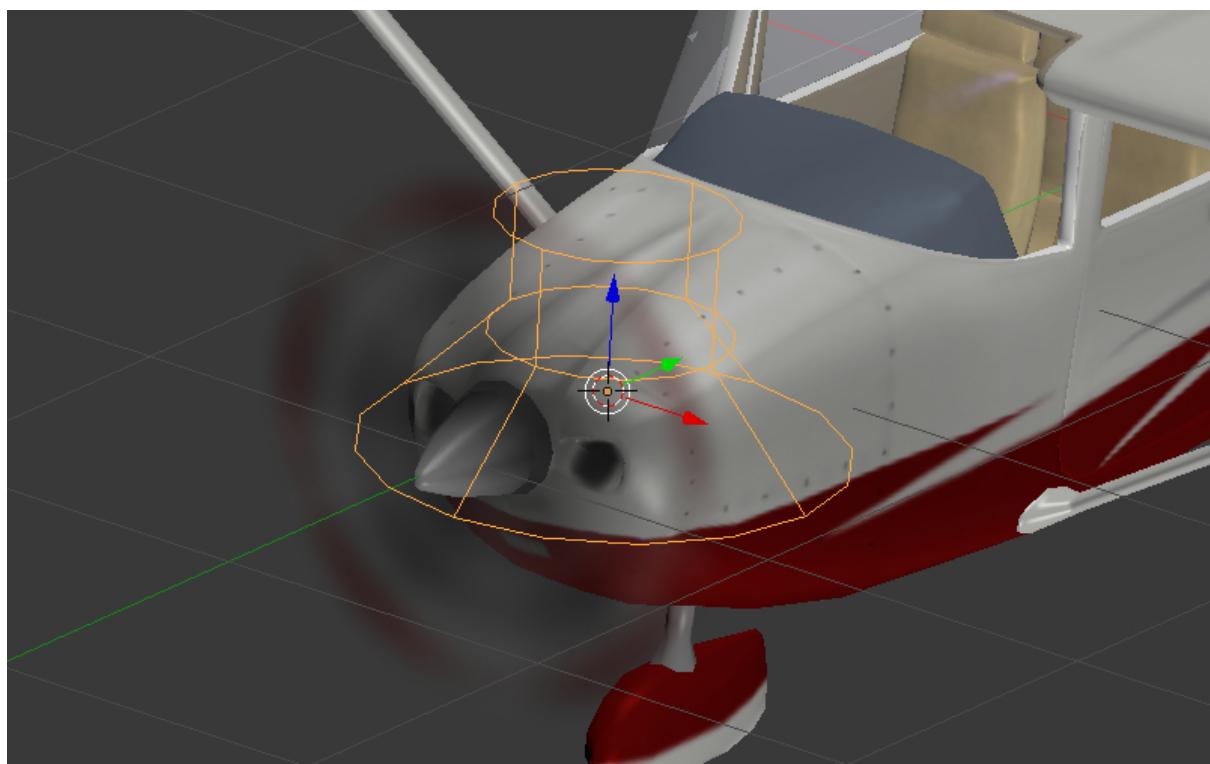
饱和度 默认值是 1.0。

音频

目录

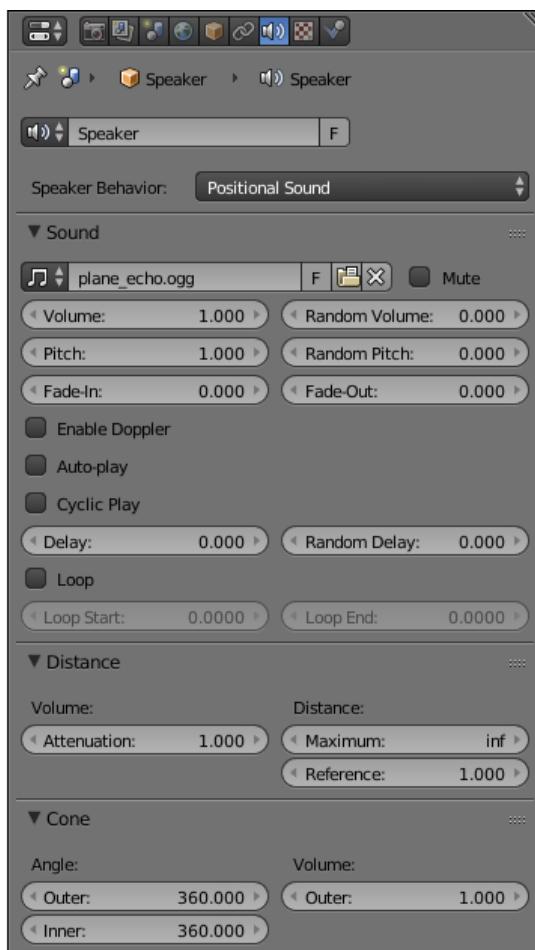
- 音频
 - 音源设置
 - * 声音选项卡
 - * 距离标签
 - * 圆锥标签
 - 调音台
 - 处理和解码
 - * 支持的格式 (容器) [F](#)

在 Blender 中创建音频源。使用标准的扬声器物体。



音源设置

扬声器参数可以在 属性面板的 物体数据选项卡下设置。



引擎支持标准的所有 Blender 声音参数和一些引擎特定设置。

声音选项卡

静音

启用此参数静音扬声器。

扬声器行为:

音频源的行为。

Positional ---高品质的声音空间定位和方向性（锥度）。通过 Web Audio API 用于声音渲染。这种声音的播放性能是最少，因此只将其用于短采样。这是默认值。

Background Sound — 高质量全方位声音没有空间定位。用 Web 音频 API 进行声音渲染。它是更好的性能，但对音乐不是有效。

Background Music — 用来播放音乐。最大的性能，因为使用的音频 HTML

标记，但具有最低限度的灵活性。

下列可用选项在 Sound 面板上 

音量

扬声器音量

随机音量

添加随机音量。所得到的值延迟播放。

音速

声音播放速度。

随机音速

声音播放速度的附加随机化。所得到的值被延迟计算。

淡入

淡入的时间间隔。

淡出

淡出的时间间隔。

启用多普勒

打开后，源移动会产生频移现象。

自动-播放

该选项启用默认播放。

循环播放

循环的声音播放功能。

循环

循环的声音播放功能。与 Cyclic play 选项相反，它保证零延迟后重复。选项仅可用于声源的 Positional 或 Background Sound 的行为。

循环启动

标记将循环源片段的起点。这个值是在从源文件的开始以秒计数测定。

循环结束

标志着源片段的结束点。该值也是以秒为单位。

延迟

声音播放开始前的延迟。

随机延迟

额外的延迟随机化。根据公式计算所得的值 $Delay_{result} = Delay + Delay_{random} * Random_{[0-1]}$.

距离标签

衰减

此参数定义的距离有多强影响音量。默认值是 1.0。

最大值

对于音量计算的最大距离。

参考

此设置基准距离在该音量为 100%。

圆锥标签

角组的参数:

Outer

在度外锥角。超出这个锥体体积是外锥体积。内和外锥体之间的体积内插。

Inner

度内锥角。锥体内的体积为 100%。

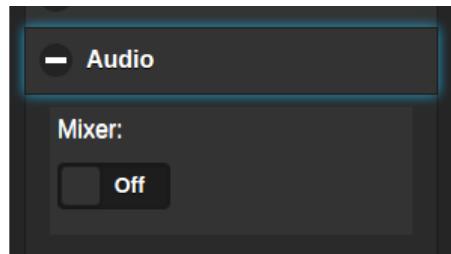
卷组的参数:

Outer

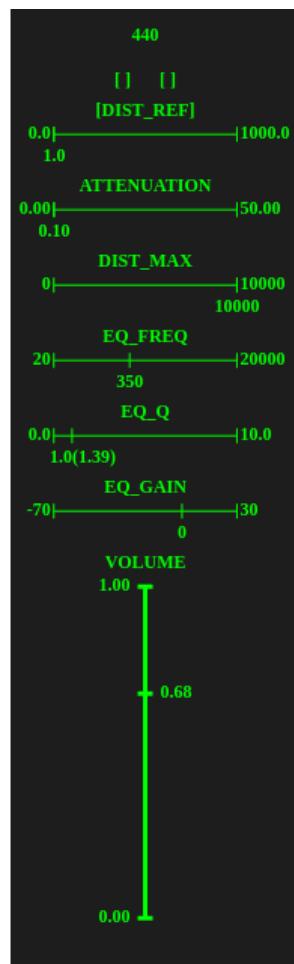
这台外锥外卷。

调音台

参考现场查看器此功能可以在发现。



使它显示包含当前场景中的扬声器的各种参数均衡器接口。



可用于混音的参数包括：

DIST_REF 这个值设置一个特定扬声器的基准距离。仅适用于“阵地 Sound”型扬声器。

ATTENUATION 此值设置为所选的扬声器发出的声音的衰减因子。可只为‘位置声音’键入扬声器。

DIST_MAX 这个值示出了用于一个特定扬声器的最大距离。如果从相机到扬声器的距离比这个数目大的扬声器不会发出任何声音。仅适用于 Positional Sound 型扬声器。

EQ_FREQ 当施加升压的中心频率。

EQ_Q 品质因数 Q。此值控制的频带宽度，下午两点会提振。增加此参数的值，减少宽度。

EQ_GAIN 此值控制，将被应用到由扬声器发出的声音的提升（以 dB 为单位测量）。

VOLUME 由扬声器发出的声音的音量。

下面这个列表，一个参数均衡器的频率特性被示出。

可以使用数字键盘控制混音器。

- 键 8, 4, 6 和 2 行为像箭头键和用于不同的扬声器和它们的参数之间的切换。
- 键 + 和 - 增减一个所选参数的值。
- 该 7 键可用于静音选定的扬声器。
- 该 9 键可以用来制作选定的扬声器梭罗（仅此扬声器会发出声音）。

处理和解码

支持的格式（容器）

- ogg Vorbis 编码解码器（Chrome、火狐浏览器）
- mp3 （Chrome, Safari）
- mp4、AAC 编码解码器（Chrome, Safari）

这推荐使用 Ogg，它是一个开放式标准，在浏览器常见并提供好音质。在质量和兼容性方面的最佳格式是 48 kHz/16 位。单声道声音（单声道）用于存储拍摄的样本，而双声道音响（立体声）用于音乐播放。

不同格式之间的资源转换的描述在相应章节.

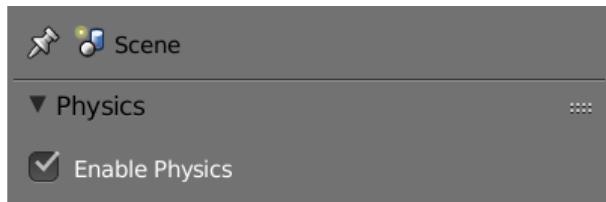
物理

目录

- 物理
 - 使用准备
 - 静态物理类型
 - 动态物理类型
 - 约束
 - 轮式的车辆
 - * 底盘设置
 - 浮动物体
 - * 浮动物体设置
 - 浮动交通工具或叫船舶
 - * 船舶设置
 - 角色
 - * 角色设置
 - * 基本角色
 - * Controlling Characters with API
 - 导航网格
 - * 创建导航网格
 - * 使用导航网格
 - 在应用程序中的使用

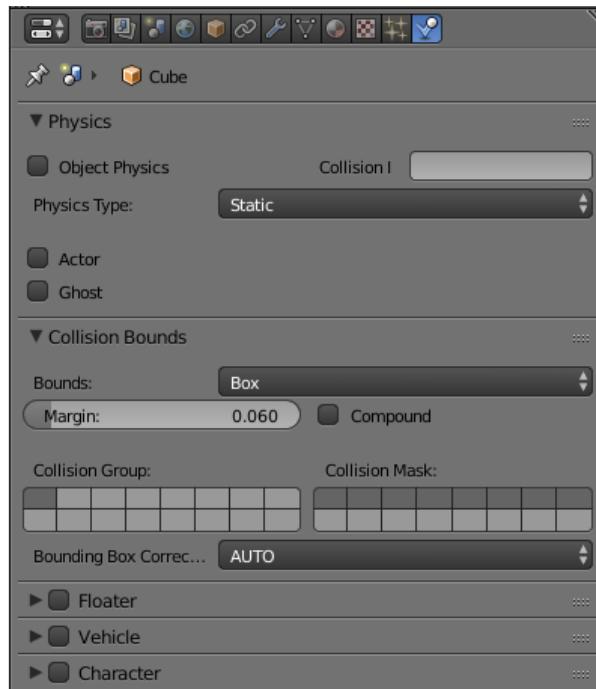
使用准备

为了在场景中启用物理, 请使用在 Blender 场景选项卡下 Physics 面板的 Enable Physics 复选框。

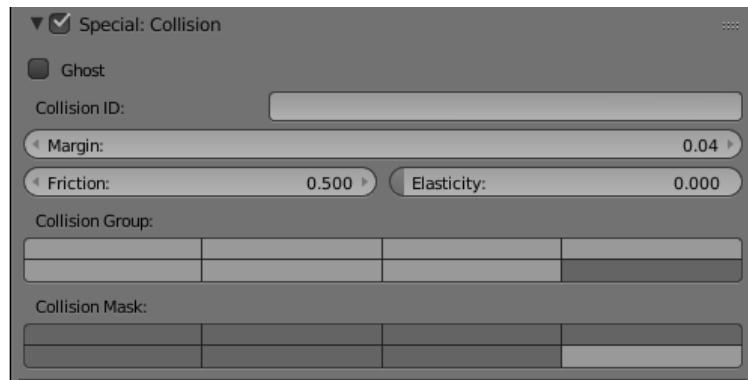


静态物理类型

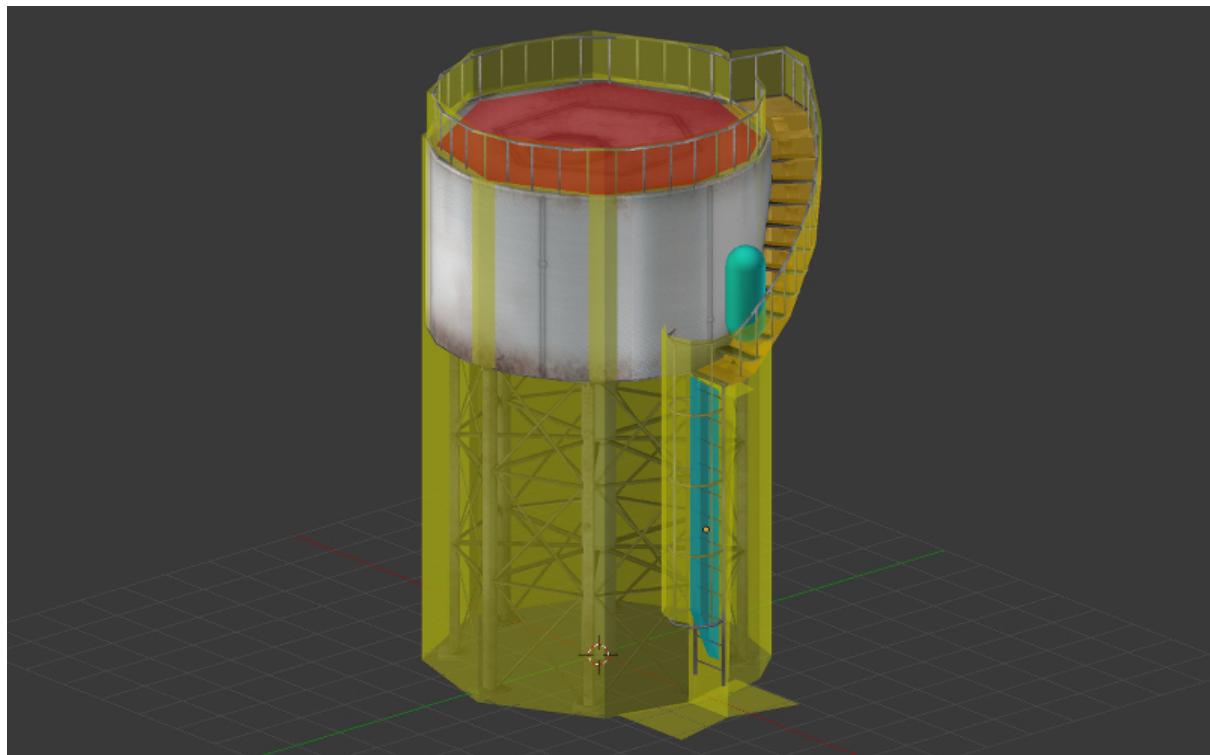
可以用来限制其他物体的运动, 例如与地形、墙壁碰撞等。在物理学中的此类物体的 Static 值 (默认设置) 设置应选择 Physics Type 选项。



一个网格可以分配到一个或多个物理材质。根据 Material 面板的 Special: Collision 选项卡必须被激活。



Ghost 选项排除物理相互作用的材质，但仍将通知应用它的接触。示例 - 检测角色坐在垂直梯子上。



Collision ID 的字段用于检测与指定的材质碰撞，可以留空。Collision ID 用法的一个示例，检测地形表面的角色位于-草，沙子，木制涂层等等。

Margin 字段允许自定义在哪里网格的碰撞区域的宽度。此选项提高了物理碰撞模拟稳定性。

还有，在此面板中有一些材质物理设置。支持以下设置：Friction, Elasticity.

Collision Group 字段对应于材质属于的物理组。Collision Mask 字段定义这种材质将与之交互的所有物理组。

动态物理类型

专为刚体运动仿真。

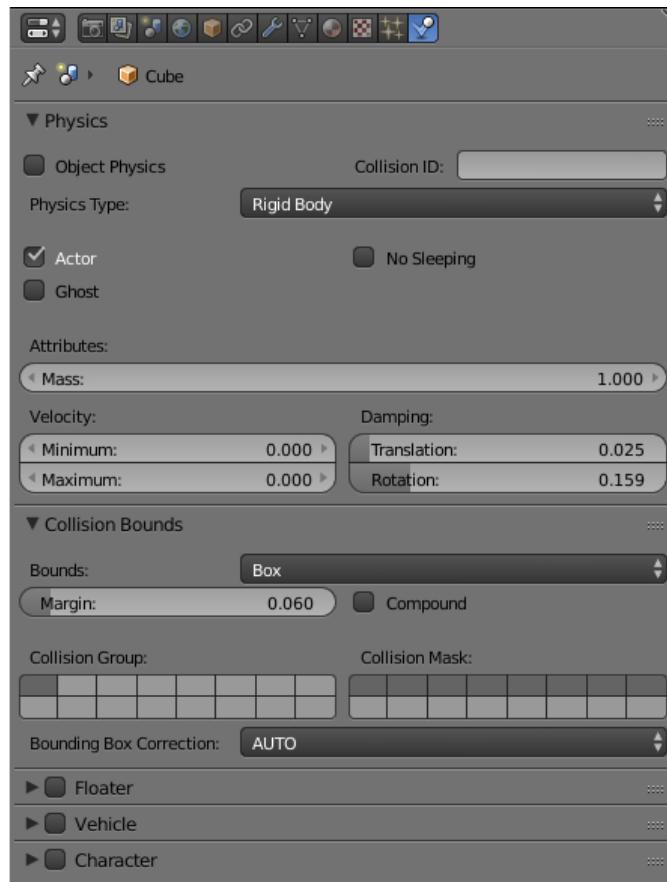


Object Physics 复选框必须在物体 Physics 面板下启用。Collision ID 字段用于检测与特定的物体的碰撞（例如，用于检测 FPS 角色和不同项目接近程度），可以留空。

在这样一个物体的 刚体（旋转）或 Dynamic（不旋转）的值的物理设置可以在 Physics Type 选项选择。在 Collision Bounds 设置碰撞体类型可以选择 - 支持的类型有：Box, Capsule, Sphere, Cylinder, Cone。同时，下面的物理参数可以设置：Mass, Damping - 为 Translation 和 Rotation。

Collision Group 字段对应于物体所属的物理组。

Collision Mask 字段定义此物体将与之交互的所有物理组。

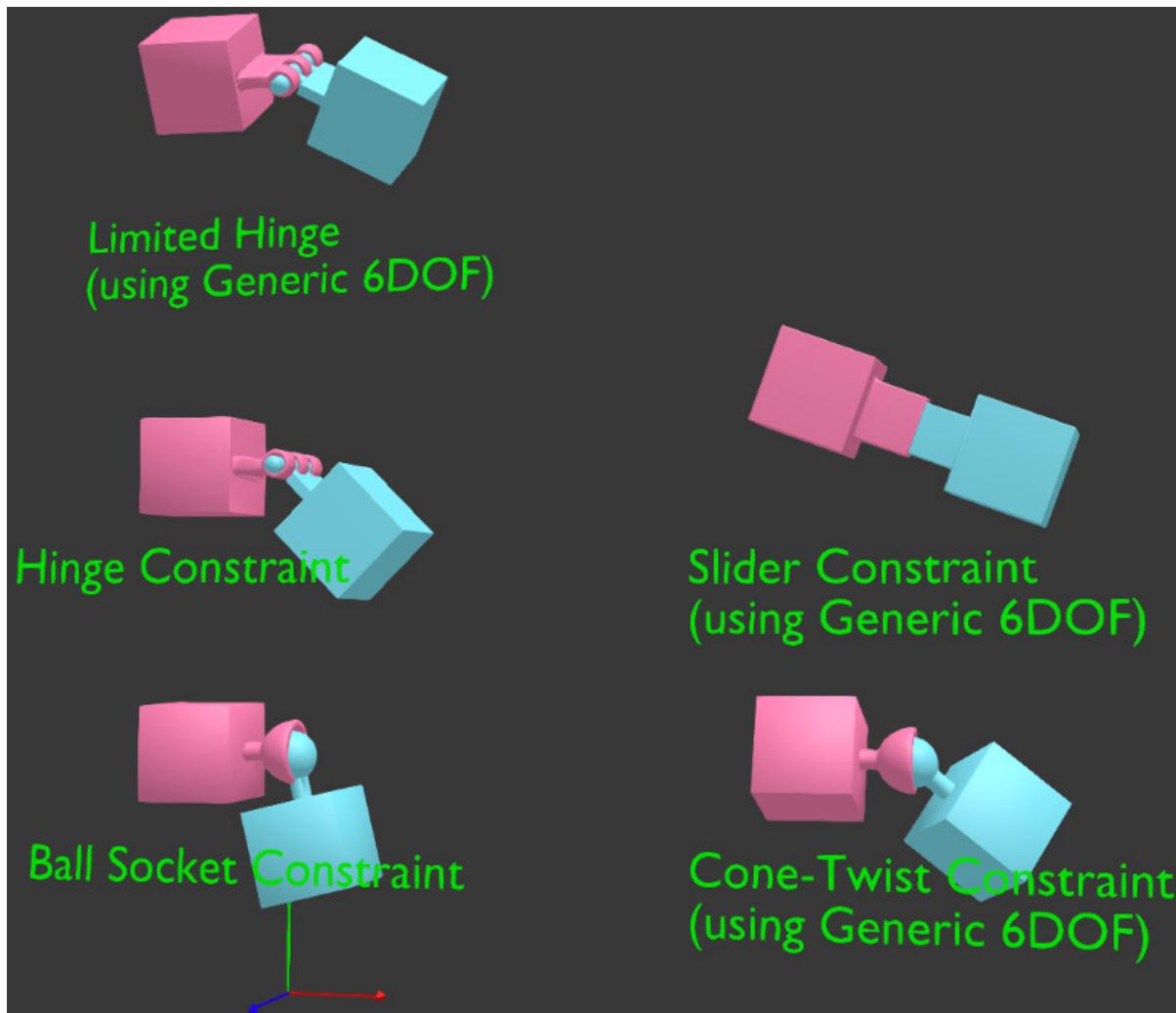


这类物体的材质支持 Friction 和 Elasticity。当多个材质广泛应用于一个单独的网格时，物理参数取自其中的第一项。

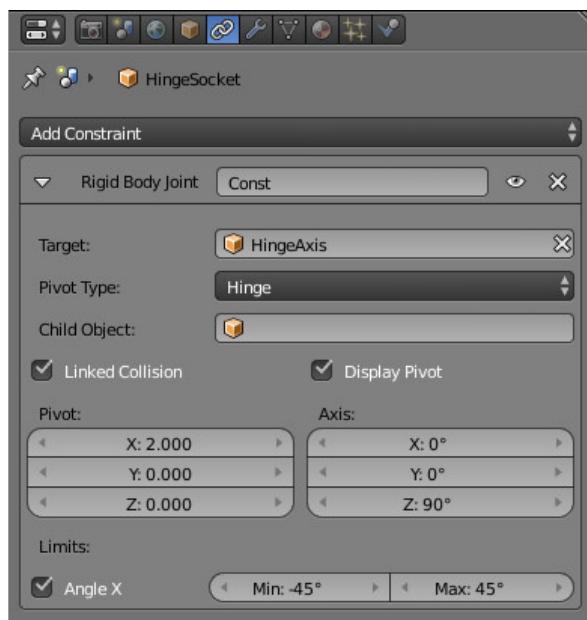
因为相机物体 Physics Type = Dynamic 参数必须使用，并且必须启用 Object Physics 复选框。

约束

物理约束用来限制物体的自由度。

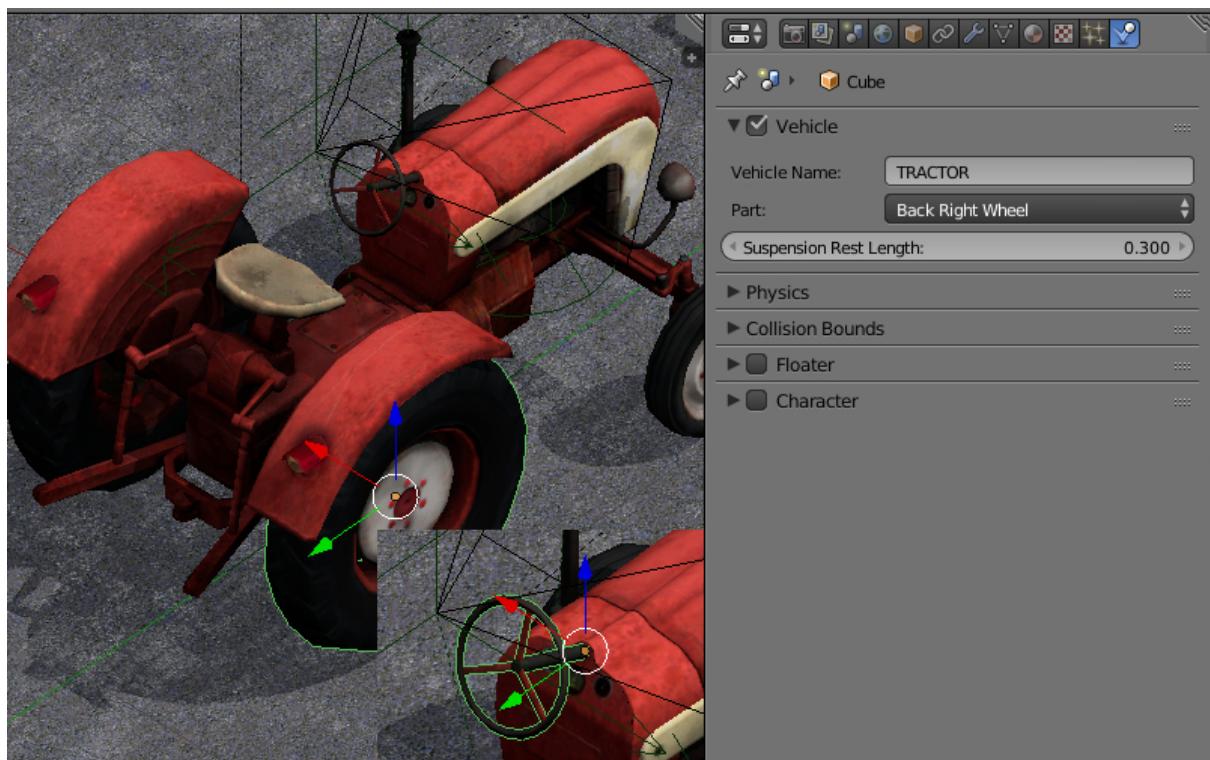


向物体添加物理约束 (Rigid Body Joint) 可以在 Object Constraints 面板上执行。受支持的类型 (Pivot Type) 是: Ball, Hinge, Cone Twist, Generic 6 DoF. 物理约束可以添加到两个相互作用的物体之一，而另一个物体作为 Target。这两个物体可以有静态和/或动态的物理类型。在约束 (除外 Ball)，可以设置的平移和旋转的限制。



轮式的车辆

车辆的模型必须包含 6 单独的物体 - 机箱、4 个轮子和一个方向盘。机箱的网格中心应该对应于质量的中心。车轮对和方向盘的中心网格应位于旋转轴。方向盘应该面向在本地空间的坐标 -X- 旋转轴, Y- 向右和 Z- 向上。该物体可以具有任何名称。



所有的 6 个物体选择 Part，在 Vehicle Name 字段中指定相同的 id，选择正确的物体类型- Chassis, Steering Wheel, Back Right Wheel 等等。Suspension Rest Length 设置对车轮可用。

有必要指定机箱的真实质量（因为默认值是仅 1 公斤）。做这件事先到物理设置，选择物理类型 Rigid Body 的 Physics Type 选项并指定 Mass 所需的值（例如，1000 公斤）字段。

底盘设置

Force Max 车辆最大驱动力。

Brake Max 最大制动系数。

Suspension Compression 阻尼系数为悬浮舒展。

Suspension Stiffness 悬架刚度系数。

Suspension Damping 悬架阻尼系数。

Wheel Friction 车轮和表面之间的摩擦系数。现实车辆它应该是 0.8 左右。但是，显著提高它能够实现更好的控制 (1000 和更多)。

Roll Influence 减少车轮的扭矩降低车辆倾覆概率 (0-无力矩、1-真正的物理行为)。

Max Suspension Travel Cm 以厘米为单位的最大悬挂行程。

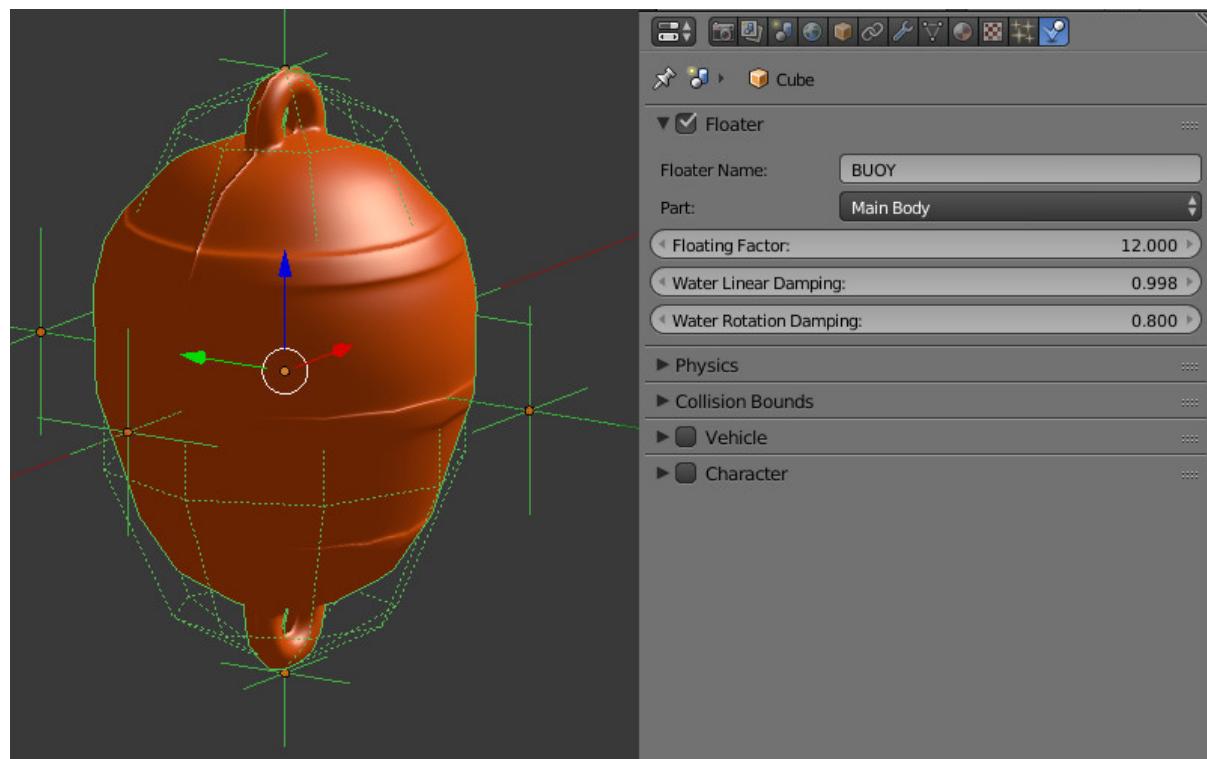
对 Steering Wheel 必须指定的是方向盘的最大角度 (Steering Max) 以及方向盘和轮胎转动之间的比值 (Steering Ratio). 最大方向盘转角设置值为圈。一圈等于 360 度。因此，如果 Steering Max 等于一以及 Steering Ratio 等于 10 的话，方向盘转 360 度前轮转 36 度。

在这个时候你可以导出并加载场景到引擎。我们建议您创建一个物理材质的路面。要在查看器中选择控制的物体按 Q 并选择机箱。使用 W, A, S, D 用作控件。

此外，我们可以调整 Damping of Translation and Rotation。这将影响汽车惯性与速度。

道路路面材质的摩擦和弹性不会影响车辆的行为。

浮动物体



为了使要浮在水面上（用 Water 材质物体）的物体，就必须使 Floater 面板启用。有两种类型的浮动物体：Main Body - 动物体本身和 Bob - 辅助 bob-物体让漂浮执行。浮动物体可以有无限的多个 Bob 物体。可以是网格或 Empty 物体。

相同的浮动物体包含的一部分的所有物体必须 Floater Name 字段中都具有相同的名称。

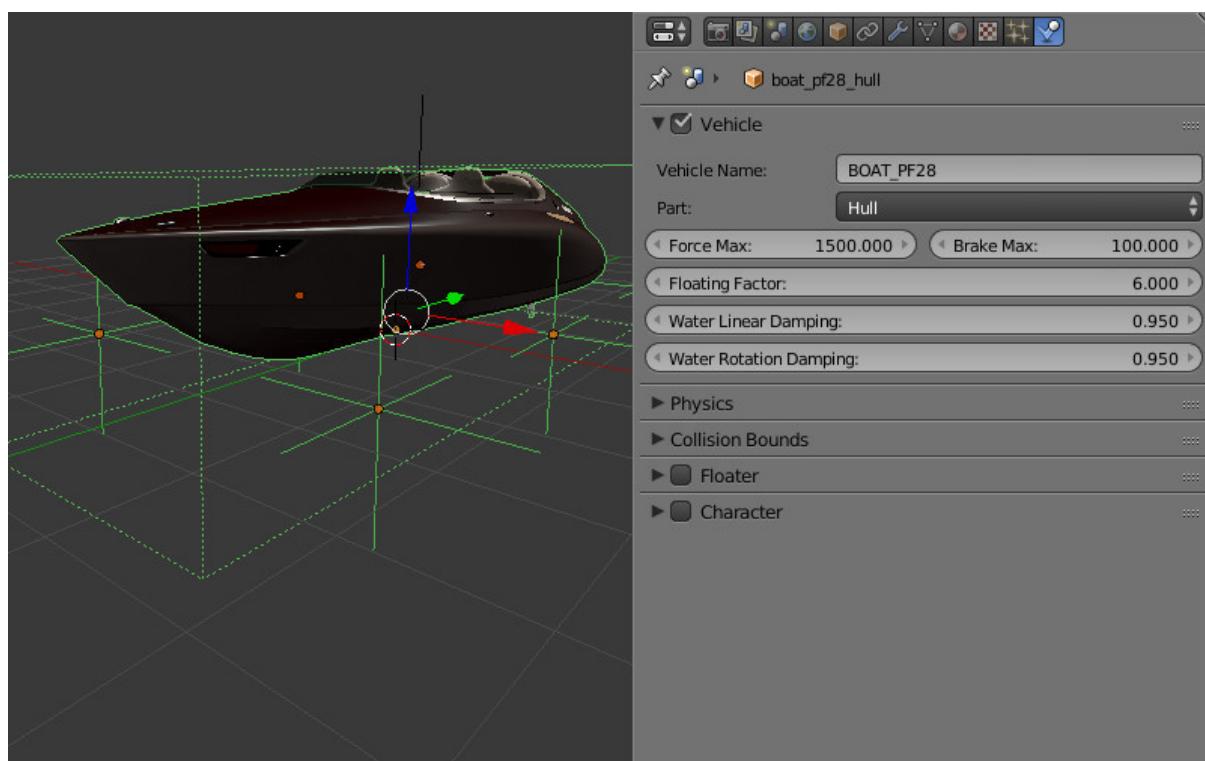
浮动物体设置

Floating Factor 浮力系数。

Water Linear Damping 物体在水面上时线速度阻尼（或水下）。当物体不是在水中时，则使用物理设置。

Water Rotation Damping 物体在水面上时的旋转阻尼（或水下）。当物体不是在水中时，则使用物理设置。

浮动交通工具或叫船舶



船舶使用 Vehicle 设置的一些参数和 Floater 设置的所有设置。它是必要的主要物体上 Part 类型的 Hull 设置。类似于浮动物体船只需要辅助 Bob 物体。

船舶设置

Force Max 车辆最大驱动力。

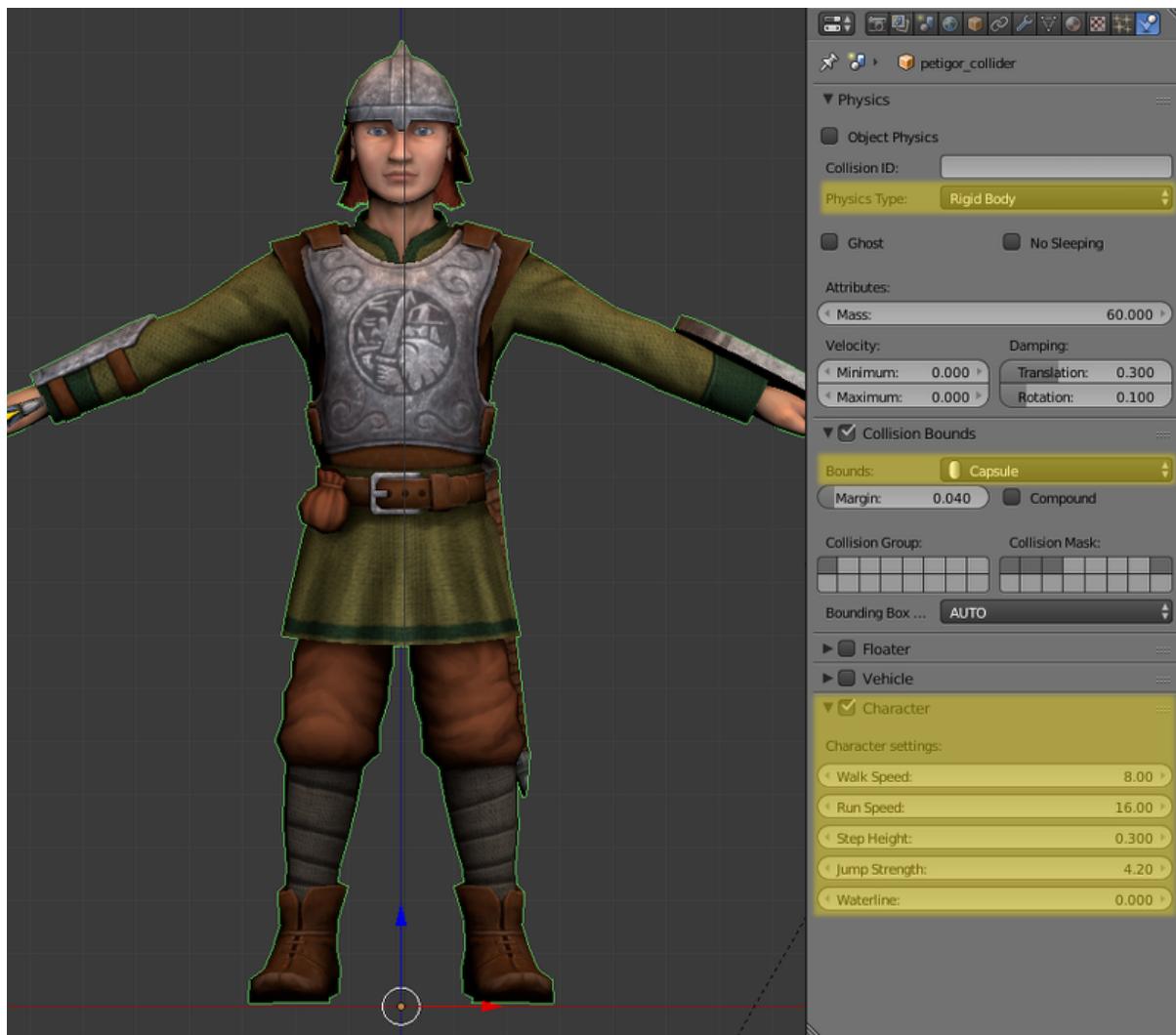
Brake Max 最大制动系数。

Floating Factor 浮力系数。

Water Linear Damping 物体在水面上时线速度阻尼（或水下）。当物体不是在水中时，则使用物理设置。

Water Rotation Damping 物体在水面上时的旋转阻尼（或水下）。当物体不是在水中时，则使用物理设置。

角色

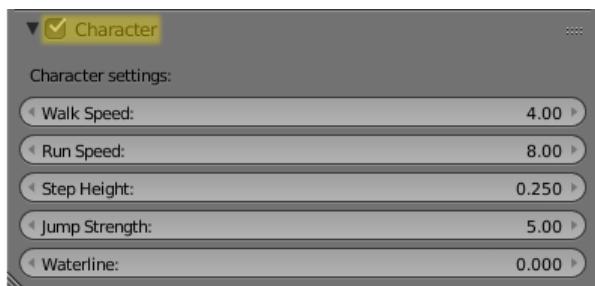


这个功能可以通过点击 Physics 面板上的同名标签的标题 Character 复选框被激活。

从引擎的角度来看，角色本身是一类适当的物理物体，该物体可以与其他物理物体发生碰撞或受到物理力如重力和浮力的影响。

角色物体可以使用多种 API 方法，由他们的名字 character 关键字容易控制。所有这些方法驻留在 physics API 模块。

角色设置



所有设置列出在这里在激活 Character 选项卡后可用。

Walk Speed 此参数设定一个角色行走的速度（以米每秒为单位）。这个参数的可能的最低值是零，而最高为 10。

它的默认值是 4。

Run Speed 这是奔跑角色（以米每秒为单位测得）的速度。其值可以从零到 20 变化。

此参数默认设置为 8。

Step Height 这是一个该角色可以超越（米）的障碍最大高度。这个参数的值可以从零到 1 变化。

此参数的默认值是 0.250。

Jump Strength 此参数设置人物跳跃的力量。其最小值是零，而其最大值为 50。

默认设置为 5。

Waterline 这个参数设置的角色物体的水线。水线从物体（不是物体的网格）的物理形状的中心测量的。如果物体被浸入水在该线以下的，它会漂浮。其值可以改变从 -2 到 2。

默认情况设置为零。

基本角色

Blend4Web 引擎有快速建立用户控制的第一人称角色的选项。这样的角色将只有非常基本的功能，但也可以没有任何编程的情况下设置。

要建立一个基本角色，请按照下列步骤操作：

1. 选择您想要作为一个角色使用一个物体。

2. 启用选定物体的物理和物理设置类型到 Dynamic 或 Rigid Body.
3. 启用该物体的角色选项。
4. 创建一个新的相机或选择一个现有的和其类型设置为 Eye .

现在，带有摄像头的基本角色将出现在您的场景。

该角色可大致和常规 Eye 式照相机相同的方式作为进行控制：

- WASD 键移动角色。
- 摄像头角度由方向键控制或移动鼠标按住左按钮。
- 按 C 键切换飞行模式（默认启用）。

注解: 只有一个基本角色可存在于场景中。如果场景中有多个角色，该引擎将使用他们的第一个基本角色，而忽略其他。

Controlling Characters with API

Basic character described above can only provide generic functionality. If you need more control over character behavior, you should use methods from the `fps` module.

The most important of them is the `enable_fps_controls()` method which, as its name suggests, is used to initialize characters in the scene. It can be used as follows:

```
var m_fps = require("fps");
m_fps.enable_fps_controls();
```

This method should be used at the start of the application (in the `load_cb` function) to enable character controls.

注解: This method may conflict with the `enable_camera_controls()` method which is used to enable basic character described in the previous section. These two methods should not be used simultaneously.

The method also has the following optional parameters:

character sets the character object. The link to the character can be retrieved by calling the `get_first_character()` method. Should be used if more than one character is present in the scene.

`element` specifies the HTML element to which the method adds listeners.

`motion_cb` specifies the callback function that is called when character changes its direction.

`gamepad_id` specifies the ID of the gamepad plugged to the system.

`forward_sens`, `backward_sens`, `right_sens`, `left_sens`, `jump_sens`, `fly_sens` - these parameters are used to set arrays of sensor types for specific character action like walking in different directions, jumping and so on.

`rotation_cb` specifies the callback function that is called when a character or camera rotates.

`lock_camera` - setting this parameter to true will parent the scene camera to the character.

Other important methods are `set_cam_sensitivity()` and `set_cam_smooth_factor()`. The first one of them sets the sensitivity of the camera (defined by a numeric value in the range from zero to 100). The second method defines how smooth the camera moves (defined by a value in the range from 0 to 1.0).

Character States

Character states are constant values that represent current behavior of the character (does it currently walk or fly or do anything else). Every character present in a scene always have one and only one state.

Available character states:

`CS_CLIMB` - a character is climbing.

`CS_FLY` - a character is flying.

`CS_RUN` - a character is running.

`CS_STAY` - a character does not move.

`CS_WALK` - a character is walking.

Current state of a specific character can be retrieved using the `get_character_state()` method, while a specific state can be assigned to a character with the `switch_state()` method. The following example shows how these two methods can be used:

```
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);
```

Action Binding

The `fps` module also provides means for binding various actions to character events. The `bind_action()` method is used for this:

```
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);
```

This method features the following parameters:

- the first parameter is the type of action:

`AT_CONTINUOUS` - an input type that detects a continuous user action such as mouse movement, keyboard key held down, gamepad stick tilt etc.

`AT_PRESSED` - this input type detects a discrete user action, e.g. pressing a button.

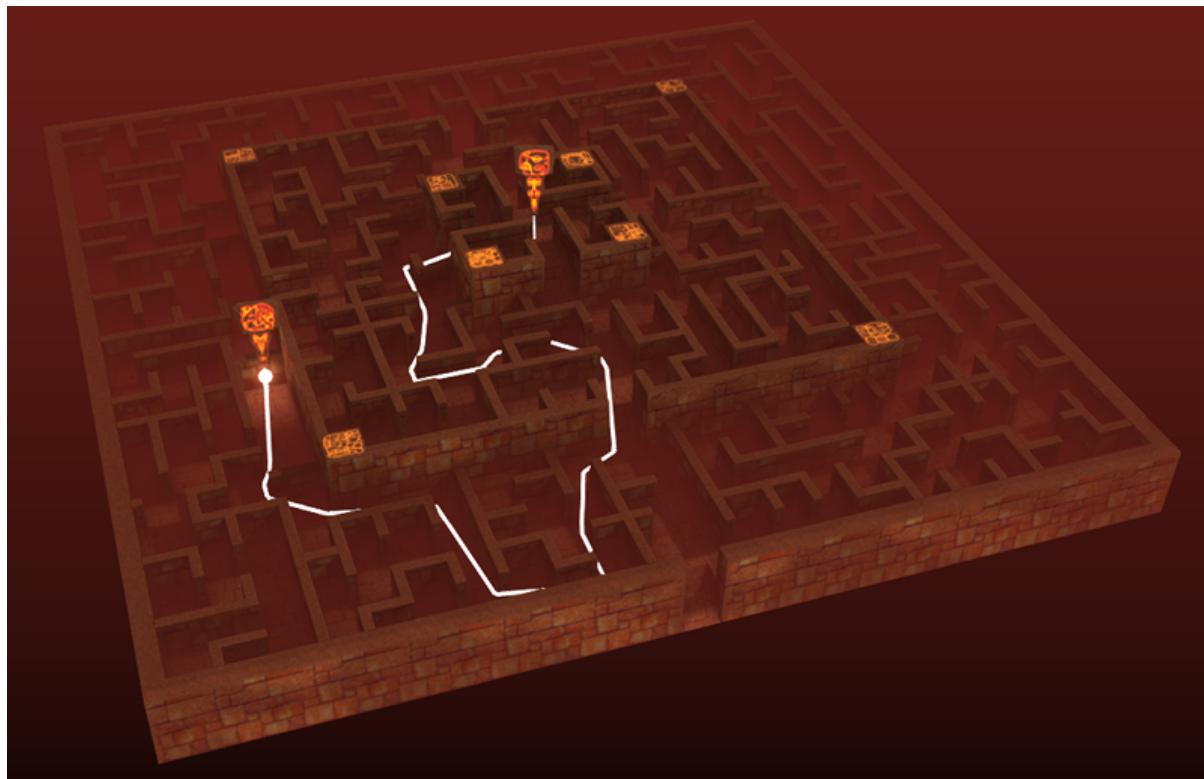
`AT_RELEASED` - an input type that detects button (mouse, gamepad or keyboard) being released.

- the second parameter defines an array of sensor types (such as keyboard keys, gamepad buttons or mouse actions).

- `action_cb` is a callback function that will be called each time user performs action defined by the first parameter.

导航网格

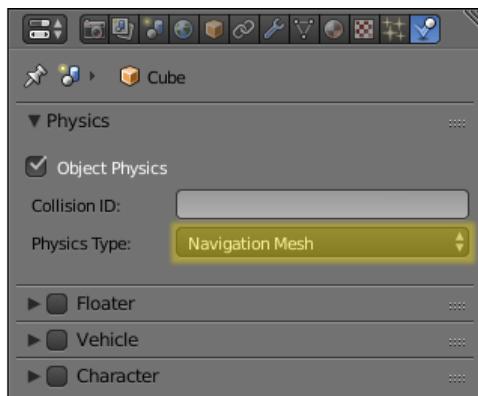
导航网格（通常缩写 navmeshes）是用来消除需要额外计算比如碰撞等使寻路更加简便的网格物体。



在此示例中（取自代码片段），导航网格用于生成通过迷宫的路径。

创建导航网格

若要将物体生成 navmesh，打开 Physics 面板的 Object Physics 参数，从 Physics Type 列表中选择 Navigation Mesh。



导航网格也可以使用特定的工具手动创建或生成自动。



注解: 此工具是原产于 Blender 并不是 Blend4Web 引擎的一部分。

此导航网格生成接口可以发现专用面板下的 Scene 选项卡上，但此功能是为了实时应用程序，它仅当引擎类型设置为 Blend4Web 或 Blender Game 才显示（Blender 内部、Cycles 或任何其他渲染引擎不会有显示）。

在此面板上的主要工具是 Build Navigation Mesh 按钮。正如标题所暗示的它用来建设基于当前选定对象的导航网格。Navmesh 是作为一个单独的物体放置在所选的物体上面，因此保留该物体创建的。

除了此按钮 Navigation Mesh 面板为用户提供了几组参数将生成 navmesh 进行微调。

Rasterization (栅格化) 组的参数 **F**

Cell Size 栅格化单元格大小。更大的值会导致更大的 navmesh 多边形。最高的可能值是 1.0。默认值为 0.3。

Cell Height 棚格单元格高度 (navmesh 和生成它的对象之间的距离)。默认值为 0.2。

Agent 组的参数F

Height 最小高度，代理还能走路。默认值为 2.0。

Radius 代理的半径。默认值为 0.6。

Max Slope 代理能够行走的坡度的最大角度。以度为单位。默认值为 45。

Max Climb 网格单元格之间的最大高度，代理可以爬上。默认值为 0.9。

Region 组的参数F

Min Region Size 一个区域的最小大小 (较小的区域将被删除)。默认值为 8.0 级。

Merged Region Size 一个区域的最小大小 (较小的地区，将合并)。默认值为 20.0。

Partitioning 用于分区导航网格的方法。受支持的方法是F

Monotone 是最快的方法。使用它可以生成细细长长的多边形。

Layers —速度较慢，但仍然相当快的方法，可产生更好的三角剖分比单调 (Monotone) 分区。

Watershed —经典重铸生成最好的剖分分区方法。此选项是默认选中的。

Polygonization 组的参数F

Max Edge Length 边缘轮廓最大长度。默认值为 12.0。

Max Edge Error 从轮廓到单元格的距离最大误差。默认值为 1.3。

Verts Per Poly 每个多边形顶点的最大数目。默认值为 6。

Detail Mesh 组的参数F

Sample Distance 详细信息网格采样间距。默认值是 6.0。

Max Sample Error 最大误差的网格简化算法。默认值为 1.00。

使用导航网格

Navmesh 创建后，它然后可用于在应用程序中计算两个点之间的运动轨迹。这可以从 physics 模块的两种方法得到 `navmesh_get_island()` 和 `navmesh_find_path()`。前者用

来检索链接到当前的岛（最接近独立网段的 navmesh），而后者返回以数组形式的实际路径中的坐标。

使用导航网格的寻路的例子可以在[代码摘要](#)（相对应的摘要名字叫 Pathfinding）。

在应用程序中的使用

物理系统是在 uranium.js 模块中实现，并分别从引擎的主要代码加载。Uranium.js 模块本身是 Bullet 物理引擎的修改版，移植在浏览器中工作。为了激活物理系统，这就需要将 uranium.js 和 uranium.js.mem 文件跟应用程序的源代码放在同一个目录。

另一个方法是通过使用下面的 API 方法显式指定 uranium.js 模块的加载路径

```
m_config.set("physics_uranium_path", ".../uranium.js");
```

注解：当开发应用程序[使用了 SDK](#)，自动检测到物理引擎的路径。

如果您的应用程序不使用物理，我们建议您在 Blender 场景选项卡中 Physics 面板下关闭 Enable Physics 的标志。它也是可以通过调用下面的方法初始化引擎前强行禁用加载 uranium.js 模块

```
m_config.set("physics_enabled", false);
```

对应用程序开发人员

目录

- 对应用程序开发人员
 - Application Development
 - * Application Code Structure
 - 背景透明
 - 资源转换
 - * 命令
 - * 参数
 - * 依赖
 - * 数据格式
 - GZIP Compression
 - 示例代码
 - 加载应用程序资产
 - 事件驱动模型
 - * 传感器
 - * 示例
 - SDK 文件结构
 - 加载本地资源
 - 品质配置
 - 非标准画布位置和方向
 - 移动 Web 应用程序

Application Development

To simplify development process, we recommend to use [Project Manager](#). It can be used to quickly [create](#) a simple application with a generic code that is enough to load a simple scene and enable basic camera controls.

Application Code Structure

The process of initializing and loading an application is separated into several stages, which is reflected in the code of the application. If you are using Project Manager, a newly created Copy or Compile type project will include a main JS file, which will be placed in the SDK. The path to the file will look like this: `./projects/PROJECT_NAME/PROJECT_NAME.js`.

This file contains generic code shaped as a module. This module can be registered using the certain structure:

```
b4w.register("my_module", function(exports, require) {  
    // module code  
    //...  
});
```

So, the code of a module is contained within a function that accepts exports and require parameters.

1. require is the method used for loading engine modules. The generic example mentioned above loads several modules:

The most important ones of them are the app and data modules. The app module simplifies application initialization while the data module contains API methods for loading 3D scene data.

注解: To make module naming more convenient, m_ prefix is often used (m_app, m_data etc.) to show that the variable is an engine module.

2. exports is an object used for gaining access to module's functions from outside (for example, from other modules). In this case, only the init function has been made external:

```
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
        });
    }

    ...
});
```

Application initialization begins with this function, and it is called outside of the module:

```
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();
```

After this, the `app.init` method is called. It creates Canvas HTML element and performs all necessary action for initializing WebGL. This method has many different attributes, the most important ones of which are:

- `canvas_container_id` set the id of the HTML element that acts as a container for the Canvas element. By default, an element with the `main_canvas_container` ID is used (this element is located in the main HTML file of the application).
- `callback` is the function that is called after finishing initialization.

When application initialization is completed, the `init_cb` function set by the `callback` parameter is called:

```
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();
}
```

It has following parameters:

- `canvas_elem` is the created Canvas HTML element that will be used for rendering 3D content
- `success` it the flag that indicates the success of the initialization. The `false` value

meant that the application is unable to work due to initialization errors (for example, WebGL is not supported by the device).

注解: The `app.init` methods sets initialization to the `window.onload` event, so the `init_cb` function will have access to the entire DOM tree of the HTML document.

Now we can begin loading 3D scene. This is done in the `load` function that is called from the `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);
}
```

The `data.load` method is used for loading. The first parameter of this method is the path to a 3D scene file. The path to a JSON file should be relative to the main HTML application file. Projects created in Project Manager have a dedicated asset folder, and you can easily obtain the path to it. This is done in the generic example code by introducing the `APP_ASSETS_PATH` global variable that is later used in the `data.load`.

The second parameter of the method is the `load_cb` function which is called after the 3D scene is loaded and prepared for rendering.

```
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

}
```

Calling this function means that the application has finished loading and now starts scene rendering. As this is the very first moment when the 3D scene data will be available, it is a suitable moment for initializing and preparing everything related to the scene, its object, animations and other things. For example, standard camera controls can be enabled here with the `enable_camera_controls` method.

Writing Application Logic

After initializing and loading 3D scene the application will proceed to work according to the logic set by the programmer such as interacting with input devices, manipulating scene objects, controlling camera behavior and so on.

By observing the application loading process, we can determine several places suitable for performing various tasks.

1. The `app.init` method used for starting the initialization accepts engine configuration parameters. So you can configure the engine right before calling this method using URL attributes as a base:

```
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. Initialization is started by the window.onload action, which means that after it is completed, the `init_cb` function will have access to the whole DOM tree. At this moment, you already can perform some preparations such as creating and setting up interface elements. However, the 3D scene itself is not yet loaded, and its objects are not yet available.
3. After the 3D scene is loaded, the `load_cb` function is called. At this moment, all scene objects become available, thus any action that concerns them can be implemented in this function. Some examples can be found in the [Code Snippets](#) application.

Logic can be added to the application by using either browser or engine API:

1. Basic keyboard/mouse/gamepad input can be implemented with standard event handlers by using the `addEventListener` method. In more complex cases you can use the `input` API module. The engine also features the `add_click_listener` method that registers both mouse clicks and touch screen events which makes it useful for writing applications compatible with desktop and mobile devices alike.
2. Time-stretched events that have to be performed at every frame (procedural animation, for example) can be implemented with methods such as `set_render_callback`, `append_loop_cb`, `animate` and `set_timeout`. Standard `setTimeout` and `setInterval` methods can also be used.
3. For complex logic that takes into account both user actions and the state of the 3D scene can use engine's `event-driven model` that is based on sensor system.

模块系统

Blend4Web engine is built upon modular structure: all engine API methods are separated into several modules. If necessary, a module can be plugged into the application with the `require` method. We recommend to structure the code of the actual application into

modules as well.

1. Registering Modules

A module is essentially a code block wrapped by a specific structure that is used to register it:

```
b4w.register("my_module1", function(exports, require) {

    // module code
    ...
});

b4w.register("my_module2", function(exports, require) {

    // module code
    ...
});
```

The `register` method is used for registering modules. You can only register custom modules if their names do not coincide with the regular API modules. If necessary, the `module_check` method to check if a module with a given name is present:

```
if (b4w.module_check("my_module"))
    throw "Failed to register module: my_module";

b4w.register("my_module", function(exports, require) {

    // module code
    ...
});
```

2. Loading Modules

Custom modules, just like regular ones, can be plugged in with the `require` method:

```
b4w.register("my_module1", function(exports, require) {

    var mod2 = require("my_module2")
```

```
...
});
```

3. Application Initialization

Application initialization in Blend4Web is usually done with a call like this:

```
b4w.require("my_module").init();
```

Here, the my_module custom module and its init external function do, in a certain sense, act as the entry point to the application.

注解: In the global visibility scope a module can be loaded with the same `require` method available as a method of the global b4w object: `b4w.require("MODULE_NAME")`.

4. Using Multiple Modules

After a project is created in the Project Manager, its generic application JS-file will contain only one module. However, while developing the application, you might need to separate your code into several logic parts. In this case, you can either create several modules inside one file, or you can create several files each one of them contain one module.

If your application uses multiple modules, keep in mind that every one of them should be properly registered before initialization starts, or you will get an engine error if you try to call a module that is not yet registered. If you are using several JS files, the script that starts the initialization (contains application entry point) should be the last one to be plugged into the main HTML application file.

背景透明

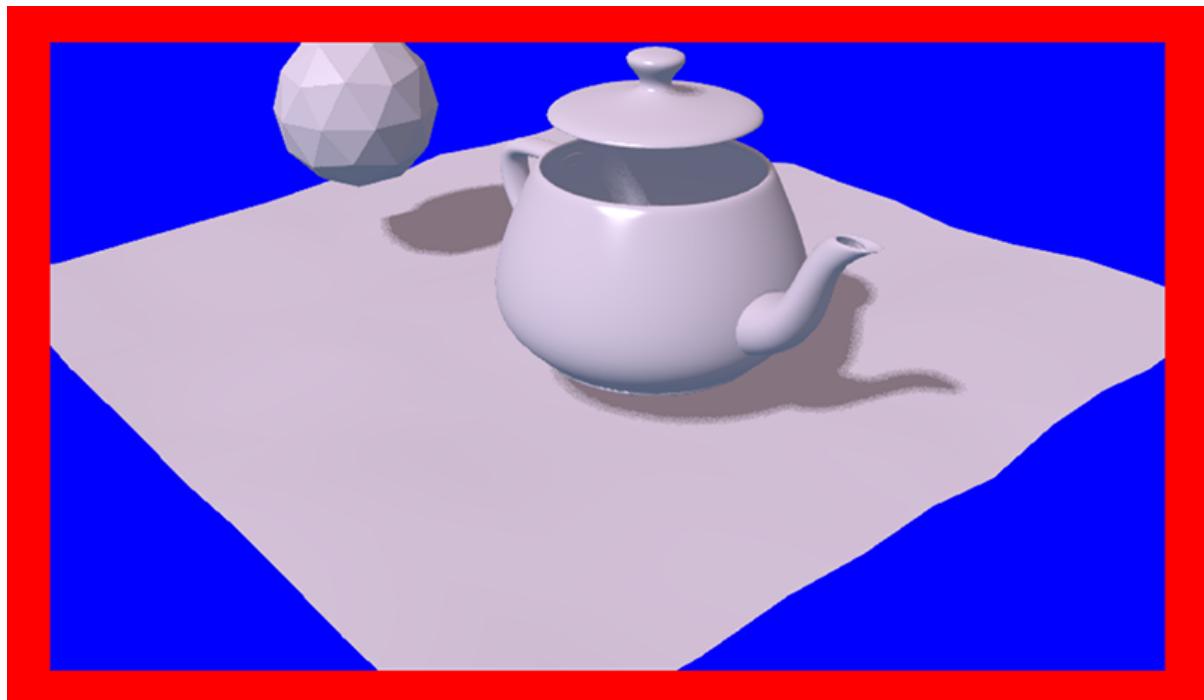
`background_color` 和 `alpha` 参数传递到 `init` 方法放在加载回调函数中（场景加载后调用的函数），像这样

```
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  
});
```

参数传递给该方法的组合定义了 Blend4Web 应用如何和 HTML 应用程序的背景融合在一起。可用的选项包括 **F**

1. `alpha = false` 背景的颜色由 Blend4Web 应用程序的 `background_color` 定义，不考虑 HTML 应用程序的背景。



2. `alpha = true` HTML 应用程序的背景可能会影响基于其透明度由 Blend4Web 应用的 ‘`background_color`’ 参数定义的四个分量的背景 (`alpha = background_color[3]`，不能混同于上面提到的 `alpha` 参数)。

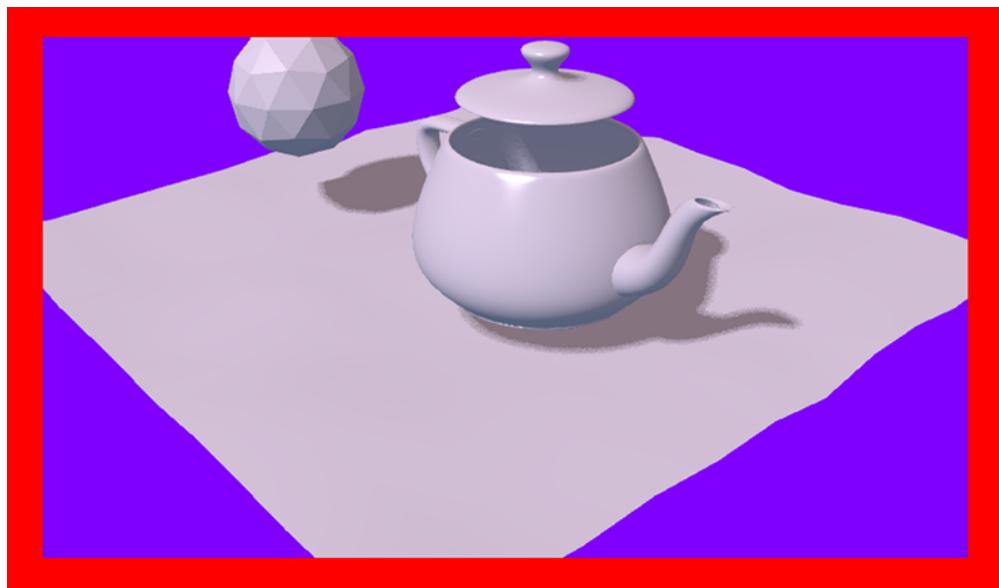
`background_color[3] = 1` 这产生相同的结果，如同 `alpha` 参数已关闭 (`alpha = 假`)

`background_color[3] = 0` 使用额外的混合



上面图片显示一个包含与页面的红色（‘红’）颜色生产了紫色基调 [0, 0, 1] 的蓝色背景的 Blend4Web 应用程序的 HTML 页面。

`background_color[3] > 0` 额外的混合用 `background_color` 有更大的影响。



这张照片显示相同的 HTML 页面相同的 Blend4Web 应用程序，但是，`alpha` 值设置为 0.5，导致应用程序背景较暗的色调。

在[颜色管理](#) 章节更详细地介绍了 `alpha` 混合机制。

默认情况下，`alpha` 参数设置为 `true`，`background_color` 设置为透明黑色 [0, 0, 0, 0]，这意味着应用程序将具有 HTML 背景并没有影响到的 Blend4Web 应用程序的背景。

背景透明度还可以利用在[网页播放应用](#) 利用 `alpha URL` 熟悉。若要使用此功能，您首先需要创建应用程序时在[网页播放器参数](#) 组中启用 `Background transparency (alpha)`

参数。

如果 Blend4Web 应用程序使用天空渲染，物体（包括天空），将完全覆盖在应用画布上，背景将是完全不透明的不受 alpha 设置影响。

注解： 默认情况下，在[项目管理器](#) 中创建 Blend4Web 场景中启用天空渲染。别忘了，为了使用透明的背景，您将需要手动禁用天空渲染。

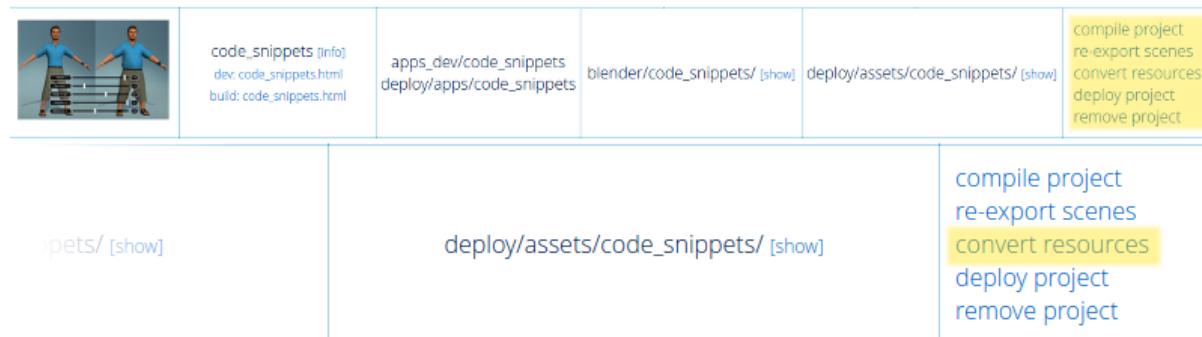
资源转换

目前，浏览器不完全支持所有可能的媒体格式，所以为了创建跨浏览器应用程序（也为优化的目的），我们建议您使用的资源转换器。

支持更广泛的平台，一个 Python 脚本 (scripts/converter.py) 为将源代码文件转换成其他格式后发布。

有两种方式来运行此脚本。

首先，您可以使用的项目管理系统自动运行它。Convert Resources 按钮可以在[Project Manager](#) 主页面中的发现，在操作选项卡中在屏幕的右侧。



其次，您可以手动运行该脚本

```
> cd <path_to_sdk>/scripts
> python3 converter.py [options] resize_textures | convert_dds | convert_media
```

对于 MS Windows 用户：

```
cd <path_to_sdk>\scripts
python converter.py [options] resize_textures | convert_dds | convert_media
```

注解: 要运行的 Python 3.x 的需要安装在你的系统中的脚本。

与-d 参数可以指定路径，其中转换将在哪一个目录发生。

要排除从资源转换的一些目录，它足以把一个文件名为 “.b4w_no_conv“。这不会影响嵌套目录转换。

resize_textures 参数用于为 LOW 模式降低纹理分辨率。

命令

转换的命令 

- resize_textures 将纹理转换到低分辨率。
- convert_dds 将纹理转换到 DDS 格式.
- convert_pvr 将纹理转换到 PVR 格式.
- convert_media 将音频和视频文件转换为替换格式.
- convert_gzip generates GZIP-compressed versions of ".json" and ".bin" scene files and ".dds" and ".pvr" textures.

资源清理命令:

- cleanup_textures 删除由 resize_textures 命令生成的低分辨率纹理。
- cleanup_dds 删除由 convert_dds 命令生成的 DDS 纹理文件。
- cleanup_pvr 删除由 convert_pvr 命令生成的 PVR 纹理文件。
- cleanup_media 删除由 convert_media 命令生成的替换格式的音频和视频文件。
- cleanup_gzip removes GZIP-compressed files generated by the convert_gzip command.

图像压缩命令:

- compress_png 压缩 PNG 文件，以减少它们的大小。此选项需要 OptiPNG 工具已经安装并在 PATH 环境变量完成设置。

其它命令:

- check_dependencies 检查转换依赖

参数

- -d, --dir <dir_path> 允许使用其他目录来存储转换后的文件。该 <dir_path> 值指定到此目录的路径。
- -j, --jobs <jobs> 指定要同时运行的作业（线程）的数量。如果该参数被设置为零或者未指定，数字会自动根据 CPU 的数量来计算。
- -v, --verbose 能够输出更多在转换文件时的详细信息。例如，转换成纹理 DDS 格式时，该脚本将显示每个要转换的文件进度百分比。

依赖

请确保你已经安装了所有的依赖转换器。你可以用下面的命令做到这一点：

```
> python3 <path_to_sdk>/scripts/converter.py check_dependencies
```

如果未安装某些程序，将显示以下信息：

找不到 PROGRAM_NAME。

Linux

依赖关系的列表如下表：

名称	Ubuntu 16.04 的包
ImageMagick	imagemagick
NVIDIA 纹理工具	libnvt-bin
Libav	libav-tools
FFmpeg	FFmpeg
PVRTC	手动安装

注解： Linux 用户还可以安装的用于优化加载媒体文件的包 qt-faststart。

Windows

用于 MS Windows 用户来说是不必要安装的，因为它们已经存在于 SDK 来安装这些软件包。

macOS

macOS 用户可以安装 brew 包管理器，然后再安装任何缺失的软件包。

在安装软件包之前，使用这些命令安装的 libpng 和 libjpeg 的库：

```
> brew install libpng
> brew install libjpeg
```

现在，您可以安装需要的依赖着手：

```
> brew install imagemagick
> brew install --with-theora --with-libvpx --with-fdk-aac ffmpeg
```

为了安装 NVIDIA 纹理工具，克隆使用以下命令库：

```
> git clone https://github.com/TriumphLLC/NvidiaTextureTools.git
```

现在你可以编译和安装包：

```
> cd NvidiaTextureTools
> ./configure
> make
> make install
```

数据格式

如下进行转化：

音频 (convert_media):

- ogg (ogg,oga) -> mp4
- mp3 -> oga
- mp4 (m4v,m4a) -> oga
- webm -> m4a

我们推荐使用 ogg 作为基础格式。在这种情况下，跨浏览器的兼容性要求的唯一的转换将是 ogg 到 mp4 。输入文件的实例：file_name.ogg，例如一个输出文件：file_name.altconv.mp4。

视频 (convert_media):

- ogg (ogv, oga) -> m4v / seq
- mp3 -> webm / seq
- mp4 (m4v, m4a) -> webm / seq
- webm -> m4v / seq

我们推荐使用 WebM 作为基础格式。在这种情况下，跨浏览器的兼容性要求的唯一的转换将是 webm 到 m4v (webm 到 seq 为 iPhone)。输入文件的实例：file_name.webm，例如一个输出文件：file_name.altconv.m4v。

为图像 (convert_dds):

- png -> dds/pvr
- jpg -> dds/pvr
- bmp -> dds/pvr
- png -> dds

输入文件的实例：file_name.jpg，例如一个输出文件：file_name.altconv.jpg.dds。

For the purpose of optimizing application performance it's possible to use min50 (halved) and DDS or PVRTC (compressed) textures. In order to do this, we need to pass the following parameters during initialization of the application:

```
exports.init = function() {
  m_app.init({
    // ...
    assets_dds_available: true,
    assets_min50_available: true,
    // ...
  });
  // ...
}
```

注解：如果您打算将使用的纹理压缩成PVRTC 格式，然后替换下面这行代码

assets_dds_available: true,

跟着下列：

```
assets_pvr_available: true,
```

This will tell the engine to load PVRTC textures, if such are present in the ./assets/ folder.

DDS 纹理压缩

DDS 纹理需要较少的内存（4 倍少 RGBA 数据，6 倍 RGB 数据），但使用它们具有以下缺点

- ‘DDS’ 纹理可能无法工作在一些设备上，尤其是移动的并不是所有支持 WEBGL_compressed_texture_s3tc 延伸；
- DDS 是一种有损压缩格式，压缩文物可能是可见的尤其是在 [正常](#) 和 [模具](#) 图；它建议 [禁用压缩](#) 对于这种纹理。



DDS 压缩失真，特别是可见的阴影边缘上的一个例子。

JSON 格式（而不是 HTML 格式），从 Blender 导出场景，DDS 纹理将自动插入，如果它们存在。

纹理可以转换为 DDS 格式，使用 [项目管理器](#) 或上文所述的 scripts/converter.py 脚本。

PVRTC 纹理压缩

PVRTC 是另一种主要在 iOS 设备上使用的纹理压缩格式。在某些情况下它可以产生相同的纹理图像比其他 DDS 格式的小两倍。

该格式具有由该引擎支持的两个压缩设置 [F] 2-bpp (两位 / 像素) 和 4-bpp (四位 / 像素)。

因为是 DDS 格式，纹理压缩算法 PVRTC 可能在某些平台上不工作，特别是移动设备，因为使用这种压缩格式需要支持 IMG_texture_compression_pvrta WebGL 扩展。

PVRTC 库和 SDK，在 Windows、Linux 和 macOS 系统相似。可以从 [Power VR Insider](#) 网页下载安装包。

Blend4Web 引擎使用控制台 PVRTC 工具。若要使用它，你需要的路径添加到路径环境变量，类似于以下内容 [F]

```
export PATH = <InstallDir>\PVRTexTool\CLI\<PLATFORM>\
```

哪里 <InstallDir> 是 PVRTexTool 安装目录，<PLATFORM> 是包含对应于您的操作系统，例如，\Windows_x86_64\64 位 Windows 操作系统的工具的版本的文件夹。

注解：在 Windows 系统中，环境变量可以设置‘系统’（在 Windows 10 和 8）或属性（在 Windows 7 和 Vista）对话窗口中，通过选择高级系统设置->环境变量，或使用控制台命令 [F]

```
SET PATH = <InstallDir>\PVRTexTool\CLI\<PLATFORM>\
```

在此之后，您将能够通过使用 converter.py 脚本与 convert_dds 命令转换为 PVR 格式的纹理。

SEQ 视频格式

在 .seq 文件格式用于顺序视频。这是适用于 IE 11 和 iPhone，因为他们目前缺少标准的视频格式支持的纹理。采用 DDS 格式的图像与其他格式相比是更理想的。

该引擎可以使用，如果他们有以下名称，手动由用户创建的文件：file_name.altconv.m4v, file_name.altconv.mp3 等等。此类文件应放在同一目录中的媒体文件在 Blender 中使用。

您还可以使用的免费和跨平台的应用 [Miro 视频转换器](#) 转换媒体文件。

GZIP Compression

Typical Blend4Web application can use various resource formats from standard HTML, JS, CSS, PNG or JPEG files to engine-specific JSON- and BIN-files that contain scene data. Compressed DDS/PVR image formats are also an option. Both big and small applications benefit from decreasing the size of the resources, as this also decreases the loading time.

Usually, loading time can be decreased by setting up caching on the server that contains web application. You can also enable GZIP compression for various file types.

Speaking of the specific file types, GZIP compression should be used for JSON, BIN, DDS and PVR files. JSON and BIN files, being the main scene files, can contain large amounts of data, while DDS and PVR also can be quite large (at least when compared to standard PNG and JPEG files), and there can be quite a lot of them.

But if for some reason GZIP compression cannot be set up on the server, it can be enabled in the application itself.

The engine can load compressed resources in the form of .gz files. To use this feature in a WebPlayer JSON type project, you need to pass the compressed_gzip URL parameter. If you are developing your own application, you need to pass the assets_gzip_available configuration parameter during the initialization.

```
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
});
```

Compressed .gz files should be placed alongside the original ones, for example, like this:

```
my_project/
assets/
    my_scene.json
```

```
my_scene.json.gz
my_scene.bin
my_scene.bin.gz
```

This also applies to the .dds and .pvr files and their compressed counterparts .dds.gz and .pvr.gz.

注解: If a compressed .gz is not present, engine will load the original file and output a corresponding message to the console.

GZIP compressed files can be generated with the convert resources command that can be found in the Project Manager interface. This can also be done in the console by running the `./scripts/converter.py` script with the compress_gzip (for compressing resources) or the cleanup_gzip (for removing compressed files) command.

示例代码

SDK 中包含的代码片段应用程序，它演示了如何使用引擎的功能。

目前，这款应用程序包含以下例子：

- Bone API - 个体的骨骼位置控制的一个例子
- Camera Animation - 程序相机动画
- Camera Move Styles - 相机改变控制方式
- Canvas Texture - 用帆布质地的工作
- 更改图像 - 即时更改纹理图像
- Custom Anchors - 创建自定义的注解
- Dynamic Geometry - 程序几何修改
- 手柄 - 经由手柄控制角色的一个例子
- Gyro (Mobile Only) - 与移动设备的陀螺仪工作
- Instancing - 复制现场运行时对象

- 线 - 程序线渲染
- Material API - 调整材质的性能和替换对象的材质
- Morphing - 利用形状键
- 多点触控（仅限移动设备）- 使用移动设备的多点触控传感器
- 寻路 - 计算路径，并使用导航网格的例子
- Ray Test - 光线投射的障碍物检测使用
- VR - a VR application example
- 网络摄像头 - 利用来自 Web 摄像头的媒体流

代码片段的应用在 `./apps_dev/code_snippets/code_snippets_dev.html`. 它可以通过使用位于 SDK 根的 `index.html` 文件的链接也可以运行。

加载应用程序资产

为了简化项目维护和服务器部署总是将你的项目文件（的 JavaScript, CSS, HTML 等）和您的应用程序资源文件（导出的场景，纹理，声音等）分开。你的 SDK 这里面的资产目录位于 `projects/my_project/assets`.

要（通过使用如 `load()`）从该目录中加载文件使用 `get_assets_path()` 方法：

```
m_data.load(m_config.get_assets_path("my_project") + "loaded_scene.json", load_cb);
```

这样可以确保您的应用程序将独立于目前的开发阶段寻找资产（开发，构建或部署）。

事件驱动模型

事件驱动模型提供一个通用接口描述的三维场景变化的状态。它简化了加工的物理事件和用户操作。

传感器

事件驱动模型的基本单元是一个传感器。传感器是一种编程实体和只能是活动的 (1, 一种) 或无效 (0, 零)。一些传感器可以携带可以在歧管的回调函数与对应的 API 接收

一个有效载荷。例如，光线跟踪传感器（线传感器）提供的交点射线的相对长度。

用户不能直接通过外部 API 控制传感器。相反，所有的传感器都必须存在于一个或多个集合-所谓的传感器流形。一个流形是与场景对象相关联的逻辑容器。它通过执行一个回调函数生成一组定义的传感器事件响应。要定义的流形，它须有以下信息（请参见的 ‘controls.create_sensor_manifold’ 功能描述 API 文档）[F](#)

- 一个对象带流行（如抛出的对象）。
- 流形（如”冲击”）的唯一 id。
- 回调的执行模式（选项有：CT_POSITIVE - 逻辑功能正结果，CT_CONTINUOUS - 以正逻辑功能的结果，并用一次结果为零每一帧，CT_LEVEL - 任何逻辑，函数结果改变，CT_SHOT - 一期逻辑函数结果的变化，CT_TRIGGER - 逻辑函数的结果开关，CT_CHANGE - 任何传感器值变化）。
- 传感器阵列。
- 逻辑函数来定义其中执行回调函数传感器状态的组合。
- 一个回调函数。
- 一个可选参数传递到回调函数。

你可以阅读更多有关在 [controls](#) 模块文档引擎 API。

示例

让我们考虑到 insonify 任务被投掷的石头的影响。不同的媒体（例如地形和墙）应产生独特的声音影响。有碰撞网格与 Blender 场景中的物理材质，材质的 id 是”地形”和”墙”。也是被抛在场景中的物理对象，该对象被命名为”石头”。

让我们定义碰撞传感器的每个介质类型的声音产生。

```
// 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");
```

传感器添加到阵列。使用 OR 逻辑的逻辑功能。放置在回调函数的声音处理的代码。创建传感器歧管的“影响” id 和 CT_SHOT 类型。

```
// 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);
```

当“石头”的物体与“地形”或“墙”的任何物理材质碰撞时，被执行的回调函数。在这个函数中，我们通过自己的传感器阵列中获得指数两个传感器的值（0 - “地形”，1 - “墙”）。传感器值 = 1（活性）是指发生碰撞与相应的物理材质。其结果是，相应的声音产生（未示出的代码）。

SDK 文件结构

addons

blend4web Blender 插件

apps_dev SDK 应用程序源代码

code_snippets Code Snippets 应用程序的源文件

scripts Blend4Web API 用法示例的源文件

dairy_plant 乳品厂演示源文件（仅适用于 SDK 专业版）

demos_animation 基本的动画演示的项目文件

demos_environment 基本的环境演示项目文件

demos_interactivity 基本交互演示的项目文件

demos_materials 基础材质演示项目文件

demos_media 基本的媒体演示项目文件

demos_particles 基本粒子演示项目文件

demos_physics 基本的物理演示项目文件

demos_postprocessing 基本的后处理演示项目文件

farm 农场演示源文件（仅适用于 SDK 专业版）

fashion 时装秀演示源文件（仅适用于 SDK 专业版）

flight 岛演示的源文件

new_year 在 2015 年新年贺卡的源文件

project.py 脚本的应用程序开发人员

space_disaster 空间灾害应用程序的源文件

tutorials Blend4Web 教程的源文件

victory_day_2015 胜利日 70 周年贺卡的源文件

viewer 浏览器应用程序的源文件

webplayer 在 Web 应用程序播放器的源文件

website 从 Blend4Web 官方网站上的应用程序的源文件

blender Blender 的场景源文件

csrc 源代码（C 语言）引擎出口的二进制部分和其他公用设施

deploy 在服务器上部署（场景源文件编译的应用程序和文档）的资源目录

api_doc 为开发人员 API 文档（自动构建，基于引擎的源代码）

apps 用于部署 3D 应用；目录重复 apps_dev

common 已编译引擎文件。从 SDK 的所有应用程序共享的（因此名称）。

assets 应用程序资产 [场景、纹理和声音]

doc 从自动生成的 HTML 格式的当前用户手册 doc_src

webglreport WebGL 报告应用程序

distfiles 构建发布文件列表

doc_src 当前手册的源文件用 reST 格式书写

index.html 和 index_assets 主要 SDK 网页文件

license 与许可证的文本文件

Makefile 建立引擎，应用程序和文档的 makefile 文件

projects 用户项目目录

README.rst README 文件

scripts 脚本

check_resources.py 脚本的检查和未使用的资源报告（图像和声音通过导出的文件中引用）

compile_b4w.py 构建引擎代码和应用程序的脚本

converter.py 脚本减半纹理尺寸，压缩纹理成 DDS 格式，声音文件转换成 MP4 和 Ogg 格式

custom_json_encoder.py JSON 的 Python 模块的分支，排序顺序相反的键

gen_glmatrix.sh 脚本基于 glMatrix 2 的源代码生成所述数学模块

graph.sh 为当前场景图 SVG 生成器，用于调试渲染

make_dist.py 发行构建脚本

memory.sh 脚本，用于检查存储器 (RAM) 和视频存储器 (VRAM)

mod_list.py 脚本用于生成的模块列表在新的应用程序使用

plot.sh 调试信息图形生成器

process_blend.py 从 SDK 所有场景自动再导出脚本

remove_alpha_channel.sh 去除图像的 alpha 通道的脚本

screencast.sh 屏幕录像脚本

shader_analyzer.py 开始其计算着色器的复杂度的本地 Web 服务器脚本

translator.py 构建插件翻译脚本

shaders 引擎 GLSL 着色

src 引擎的内核的主要源代码

addons 引擎插件的源代码

ext 形成了引擎的 API 外部声明的源代码

libs 库的源代码

tmp 临时文件（例如快速预览）目录

tools various tools for building the engine, apps or convert resources

converter_utils 二进制构建的工具，资源转换

closure-compiler 谷歌闭包编译器，它的扩展和他们的生成器

glsl

compiler 编译器为引擎的 GLSL 着色

pegjs 在 PEG.js 解析器生成的语法实现了 GLSL 预处理器，也是脚本从这些生成语法解析器模块

yuicompressor 实用压缩 CSS 文件

uranium Uranium 物理引擎的源代码和脚本建筑 (Bullet 的分支)

VERSION 包含引擎的当前版本

加载本地资源

该引擎的渲染器是一个 Web 应用程序，当你在浏览器中查看 HTML 文件它就工作。初始化之后，资源（场景，纹理）加载。这个过程是受 [同源策略](#) 规则。这条规则特别规定从本地目录加载禁止。

自 15.02 版，Blend4Web SDK 包括[开发服务器](#)解决加载本地资源的问题。

品质配置

几个质量配置文件，以支持具有不同功能的平台上实现。

- 低质量 (P_LOW) - 功能的范围被关闭（如阴影，动态反映，后处理），纹理的大小采用发行版本时减半，抗锯齿被禁用
- 高品质 (P_HIGH) - 由现场所要求的所有功能的使用，抗锯齿的方法是 FXAA
- 最高的质量 (P_ULTRA) - 渲染分辨率加倍，阴影贴图分辨率的增加，抗锯齿的方法是 FXAA（使用更高质量的设置和工作速度较慢）。
- 自定义质量 (P_CUSTOM) - 任何质量参数可以设置为任何可能值。当你需要手动设置一定的质量参数，使用此选项。默认情况下，它使用相同的设置“High”质量配置文件。



切换品质配置可在运行时的 WebGL 上下文的初始化之前进行。默认的配置文件是“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(...);
```

应用程序开发人员还可以在引擎初始化参数的 app.js 附加插件时设置 quality：

```
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
});
```

非标准画布位置和方向

画布元件，开始执行渲染时，可改变相对于浏览器窗口的位置。这可能是由于一些操作在 DOM 树，或者作为页面滚动的结果，特别是用于非全屏 web 应用。

在大多数情况下，这将不会影响该应用的任何的性能。然而，涉及到鼠标或触摸位置某些 DOM 事件可以携带不正确的信息。这是因为从相应的事件中获得的坐标是相对于浏览器窗口的原点测量，同时引擎的工作与 Canvas 元素本身（它的坐标原点位于左上角）的坐标空间。

- 如果画布元素的左上角与浏览器窗口左上角匹配并且固定在它的位置（不能移动），那么就足以使用 `event.clientX` 和 `event.clientY` 输入事件的坐标或 `get_coords_x()` 和 `get_coords_y()` 方法。

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

- 在更多复杂的画布元素操作中（滚动的页面元素，从浏览器窗口顶层角移动，在 DOM 树中的更改）这就需要获得正确计算的坐标。可以通过使用上述:`b4wref:mouse.get_coords_x()` and `get_coords_y()` 并将 `true` 作为第三个参数值的方法

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

在这种情况下，另一个选择是使用 `client_to_canvas_coords()` 方法，如下所示

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

移动 Web 应用程序

创建一个 web 应用程序的用户界面时，应考虑缩放整个网页和更改移动设备方向的选项。在这里建议，阻止缩放和保持网页的纵横比不变以避免可能出现的问题。通过将以下 meta-tag 添加到页标题得以实现

```
<meta name="viewport" content="width=device-width, initial-scale=1.0, user-scalable=no">
```

有时你可能需要锁定移动设备到某一种方向，尤其是当你不想要使用横向和纵向的接口类型。由于本机屏幕锁定 API 是试验性的并不普遍，可能使用 CSS 规则来实现更容易。

若要保持方向仅有横向模式，可以设置纵向模式中的元素旋转 `<html>` 90 度

```
@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%;  
}  
}
```

同样，纵向模式下的 CSS 规则只应该如下所示④

```
@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%;  
    }  
}
```

对于引擎开发者

目录

- 对于引擎开发者
 - 代码样式
 - * 例子
 - 构建引擎
 - 构建插件
 - 依赖
 - 命名函数和变量
 - 调试中
 - 着色器编译
 - * 验证
 - * 混淆
 - * 优化
 - * 导入/导出指令
 - * 建议和限制
 - * WebGL 的扩展
 - * 编译错误
 - 更新插件翻译

代码样式

这款引擎采用结构化程序。该代码由模块组织。没有用 OOP 方法都，没有定义的类，不进行继承等。

使用 **K & R 风格** 除了对于一个组合操作的括号开口被放置在同一行上，例如：

```
function foo_bar() {
    // ...
}

if (a > b) {
    // ...
}
```

4 个空格用于缩进（不允许 tab）。

例子

下划线符号在函数和变量名称中使用：

```
var foo_bar = 123; // correct
var fooBar = 123; // wrong
```

所有的全局变量以下划线开头：

```
var _foo_bar = null;
```

这些常量用大写字母表示，从不以下划线开头：

```
var FOO_BAR = 100;
```

外部 API 方法和属性的名称均在点之后写的。为了避免混淆他们必须用 @ cc_externs 标签字段：

```
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;

    ...
}
```

注释仅用英语。注释风格 - JSDoc。

构建引擎

构建之前, 请确保您的系统已安装所有需要的依赖关系 (参见[table](#)).

要编译引擎, 并纳入 SDK, 请执行以下命令 (在 SDK 根目录) 的应用程序:

```
make compile
```

完整的构建, 包括将资源 (纹理, 声音和视频), 编辑和转换的文档, 可以用下面的命令执行:

```
make build
```

构建发布档案:

```
make dist
```

所有上述操作可以用一个命令来执行:

```
make all
```

构建插件

二进制的 Blend4Web 插件版本是可用于以下平台: Linux 的 X32 / 64, macOS 的 x64, Windows X32 / 64。同时用户可以自行编译插件。

要做到这一点, Python 3.X (如果它是跟 blender 相同的版本更好) 和 C 编译器是必需的。在 Linux 下, 只需安装 python3-dev 和 build-essential 的程序包。

路径相对于库根:

- 构建脚本: csrc/b4w_bin/build.py
- Blend4Web 插件: addons/blend4web/

该构建过程开始于以下方式:

```
python3 ./csrc/b4w_bin/build.py
```

由于构建的结果, 你会得到一个名为二进制文件:

b4w_bin_[PLATFORM]_[ARCHITECTURE].[STANDARD_EXTENSION],

和插件位于同一目录中。例如: b4w_bin_Linux_64.so。在此之后, 插件已经准备好这个平台下使用。

依赖

所有依赖于下表中列出的重要性顺序递减。

名称	Ubuntu 的 16.04 包	用途
Bash	默认情况下包含	脚本解释器
Python 3	默认情况下包含	脚本解释器
NodeJS	nodejs	着色器编译
Java	默认的 JRE	编译和混淆引擎模块
ImageMagick	imagemagick	转换纹理
NVIDIA 纹理工具	libnvtt-bin	转换纹理
Libav	libav-tools	转换资源
NVIDIA CG 工具包	NVIDIA-CG-工具包	着色器的调试
OptiPNG	optipng	优化 PNG 文件
Emscripten	from EMSDK source 代码	构建 Uranium
Gnuplot	gnuplot	偵錯
Graphviz	graphviz	偵錯
xsel	xsel	偵錯
Sphinx	python3-sphinx	构建手册
sphinx-intl	使用 PIP v3 安装 (pip3 install sphinx-intl)	构建手册 (国际)
TeX Live	texlive texlive-latex-extra texlive-lang-cyrillic texlive-lang-chinese texlive-xetex	构建说明书 (PDF 版)
JSDoc 3	使用 NPM 进行安装 (npm install -g jsdoc)	构建 API 文档
PEG.js	从 PEG.js 源代码	着色器预处理

命名函数和变量

当创建新的函数和变量，建议使用以下前缀和后缀。

init_ 创建抽象物体

create_ 创建实际的物体

update_ 更新现有物体的状态

attach_/detach_ 添加/删除临时物体属性

append_/_remove_ 添加/删除临时属性同类的现有属性

insert_/_pop_ 添加/删除数组元素（按索引访问）

switch_ 切换标志的二进制值

apply_/_clear_ 操作标志，二进制值或任意参数的操作

set_/_get_ 设置/获取属性/变量值

_tmp 全局变量 - 缓存在一个简单的物体的形式（数组，向量）

_cache 全局变量 - 缓存在一个复杂的物体的形式

调试中

引擎调试在 debug.js 模块方法进行。

当前的渲染图的结构可以用 b4w.debug.scenegraph_to_dot() 调用，例如，在浏览器的控制台被保存在 DOT 格式。调用此方法后，保存控制台输出与.gv 扩展名的文件。为了得到在视觉形式图表中的 graphviz 的是必需的工具。转换为 SVG 格式是使用命令执行：

```
> dot -Tsvg graph.gv -o graph.svg
```

其中，graph.gv 是保存的图形文件的名称。

着色器编译

在引擎中使用的所有着色器由编译器处理。编译器执行以下三个主要步骤：

- shader 代码的验证，
- 它的混淆处理，
- 优化

为了编译运行，在 SDK 根处键入以下命令执行：

```
> make compile_shaders
> make verify_shaders
```

- make compile_shaders - 执行验证，混淆，优化和最后，编译着色器的导出，
- make verify_shaders - 只执行验证，混淆和优化。

在编译期间首先进行着色文本的语法分析（解析）。相应的解析器自动创建基于语法，使用 PEG.js 引擎。然后将由着色器验证，混淆和优化根据解析器数据，之后该着色器中抽象语法树（AST），用于在引擎直接装载的形式输出。

资源库中的主要文件的位置：

- 初步的语法 - glsl_utils / pegjs / glsl_parser.pegjs
- 解析器生成脚本 - glsl_utils / pegjs / gen_nodejs.sh
- 解析器 - glsl_utils / 编译器 / glsl_parser.js

验证

编译器执行与着色器代码验证以下程序：

- 报告有关未使用的变量和函数（死码）
- 检查着色器的语法，
- 检查着色器的一致性导入/导出机制，
- 去除奇数或重复的标记：空格，endlines 和分号。

混淆

混淆 minifies 的 GLSL 代码，使得它很难理解它。到目前为止以下过程实现：

- 具有较短的单符号（与支持导入/导出机制的）替换用户定义的标识符，两符号等的名称。

优化

优化构成以下程序：

- 除去大括号，不能在除了创建本地作用域任何方式有用（这一功能是用于处理节点/灯指示），

- 内部功能的优化 - 创建共享局部变量，以取代原来由程序员创建的。

删除未使用大括号的一个例子：替换下面的代码

```
void function(){
    int a;
    {
        a = 1;
    }
}
```

与这些代码

```
void function(){
    int a;
    a = 1;
}
```

临时局部变量的低数量是由在不同情况下重复使用它们来实现的。例如，下面的代码

```
int function(){
    int a = 1;
    int b = a + 3;
    return b;
}
```

Blender 内置渲染引擎的纹理绘图类型已被其它三个模块所取代：

```
int function(){
    int _int_tmp0 = 1;
    _int_tmp0 = _int_tmp0 + 3;
    return _int_tmp0;
}
```

注解：对于结构和数组的局部变量未优化这种方式。

导入/导出指令

导入/导出指令用于组织，结构和增加着色器代码的可读性中包含文件。它们在文件的开头指定，应该看起来大约是这样的：

```
#import u_frame_factor u_quatsb u_quatsa u_transb u_transa a_influence
#import qrot

#export skin
```

在 `# import` 指令定义了一组声明的包含文件之外，但可以从里面被访问的 ID。这里有一个限制，但：这样的 id 必然是上述地方，其中包括文件链接的地方声明。

在 `# export` 指令定义了一组可以从该文件外部访问的 ID。这样的 id 必然在这个文件中声明。

因此，它使用的着色器包含文件必须拥有所有必要的连接位置之前，进口的声明，以后它可以使用导出的标识。

IDS 既可以是变量名和函数名。如果没有导入/导出命令中，它在默认情况下认为，包括文件不使用外部声明和内部的人的使用不允许。

建议和限制

由于以下几个原因：预处理，处理多个着色器和包括文件和由于编译器的功能的需要 - 其可能保证只有当一些规则和限制方面的着色源推崇的输出代码的工作码：

1. 为了描述这是由引擎在运行中定义的常量，有必要使用 “`# var`” 特殊指令。例如：

```
#var AU_QUALIFIER uniform
AU_QUALIFIER float a;
```

这里的语法是类似的 `#define` 指令。所述 `#var` 指令的一点是，它定义了值允许解析初始着色。它是不相关的这将是什么（在上面的例子如“制服”或“属性”），因为在这个水平上它是未知反正。不过，最好是指定一个或多或少适当的描述，不是随心所欲。

注解： 该 `#var` 指令没有必要不是在着色的代码，但在预处理器表达式中使用的常数。

2. 在需要的时候使用导入/导出指令。

3. 内置的功能，不能超载 - 只有用户的。
4. 变量不应与的内置函数名称，或者主要（即使它不导致错误）中声明。
5. 的 #var 和 #define 指令不能被用于以这样运营商更换单个符号：“”，“-”，“*”=”，“/ =”，“=”，“- =”，“==”，“<=”，“> =”，“! =”，“&&”，“||”，“^^”。

例如：

```
#var EQUAL =
...
a *EQUAL b;
...
```

6. #include 指令的使用不应该包含文件的混淆中产生歧义。当多个着色器纳入相同的文件，上面定义的指令（如 #var 或者将 #define）可以对任何人影响，可能发生这种情况。此外，最好不要使用未声明的函数和变量中包含文件。
7. 多级包括或相同的多个纳入到包括不支持相同的着色器。
8. 着色器的故障也可以通过使用非平凡的预处理，例如，创建一个无效的 GLSL 代码引起的：

```
#if TYPE
void function1() {
#else
void function1(int i) {
#endif
...
}
```

9. 不要声明与这样的名字变量 node_ [NODE_NAME] _var_ [IN_OUT_NODE]，其中 NODE_NAME ---一些节点的名称，IN_OUT_NODE ---输入或输出的名称节点。
10. 重复使用 # nodes_main, # nodes_global 或 # lamps_main 指令是不是一个单一的着色器内禁止。
11. 在 # nodes_main, # nodes_global 和 # lamps_main 指令，建议在该文件中使用，包含这些着色器节点的说明，例如，在同一个包含文件。这是必要的正确的着色器验证。

WebGL 的扩展

如果他们以某种方式影响着色语言所使用的编译可能取决于 WebGL 的扩展。目前以下扩展名由编译器的支持：

- OES_standard_derivatives

编译错误

在发生错误时，编译器会在控制台输出相应的消息。

可能发生的错误的表：

错误信息	原因
错误！含糊不清的包含文件“FILE_NAME”。	在 “FILE_NAME” 含糊不清包含文件。
错误！扩展名是在混淆器不支持。文件：“FILE_NAME”。	在 FILE_NAME 文件中使用的名称 WebGL 的扩展不被混淆器支持。
错误！包括 “FILE_NAME” 找不到。	该 FILE_NAME 包含文件找不到。
Error! Undeclared TYPE: ‘NAME’. File: ‘FILE_NAME’.	FILE_NAME 文件中错误。类型 TYPE (变量, 函数, 结构等) 的未声明标识符名称。
Error! Undeclared TYPE: ‘NAME’. Importing data missed. File: ‘FILE_NAME’.	类型 TYPE (变量, 函数, 结构等) 的未声明标识符名称。声明中缺少在 FILE_NAME 所需的标识符包括根据 “# import” 指令文件。
Error! Undeclared TYPE: ‘NAME’. Possibly exporting needed in include file ‘INCLUDE_NAME’. File: ‘FILE_NAME’.	在 FILE_NAME 文件有错误。类型 TYPE (变量, 函数, 结构等) 的未声明标识符名称。可能出口到 INCLUDE_NAME include 文件应该被允许。
Error! Undeclared TYPE: ‘NAME’. Possibly importing needed. File: ‘FILE_NAME’.	类型 TYPE (变量, 函数, 结构等) 的未声明标识符名称。也许应指定为 FILE_NAME 进口包含文件。
Error! Unused export token ‘NAME’ in include file ‘FILE_NAME’.	未声明的标识符名称是允许在出口 FILE_NAME 包含文件。

Error! Using reserved word in TYPE 'NAME'. File: 'FILE_NAME'.	错误 FILE_NAME 文件。的预留 ID 用于声明 类型 TYPE (变量, 函数, 结构等) 的标识符 名称。
Error! 'all' extension cannot have BEHAVIOR_TYPE behavior. File: 'FILE_NAME'.	对于指定的 #extension 指令在 FILE_NAME 文件 all WebGL 的扩展不支持的行为 BEHAVIOR_TYPE。
Syntax Error. ERROR_MESSAGE. File: FILE_NAME, line: LINE_NUMBER, column: COL_NUMBER.	解析 FILE_NAME 着色器在线路 LINE_NUMBER 列 COL_NUMBER 语法错 误。最初的错误描述在 ERROR_MESSAGE 是引用。代码清单来自各地的相应线路连接到 邮件采取的 (注意哪些指定哪个才是真正的错 误后, 一点点就行 pegjs 解析器的特点)。
Warning! Function 'NAME' is declared in [include]file FILE_NAME, but never used.	未使用的函数名在 FILE_NAME 文件中声明。
Warning! Include file 'FILE_NAME' not used in any shader, would be omitted!	该 FILE_NAME 包含文件未在任何着色器的 使用, 所以它会从模糊版本被排除在外。
Warning! Unused import token 'NAME' in include file 'FILE_NAME'.	未使用的 ID 名称在 FILE_NAME 导入包含文 件。
Warning! Variable 'NAME' is declared in include file FILE_NAME, but never used.	未使用的变量名称在 FILE_NAME 文件中声 明。

更新插件翻译

如果您需要更新现有的所有.po 文件, 运行不带参数的 SDK / scripts 目录中的脚本 translator.py :

```
> python3 translator.py
```

为了更新现有的.po 文件, 以支持的语言代码作为参数运行脚本:

```
> python3 translator.py ru_RU
```

为了查看支持的语言列表, 运行脚本如下:

```
> python3 translator.py help
```

在任何情况下，该文件 empty.po 将在运行脚本进行更新。

更新后的.po 文件可以如常编辑/转换。

团队工作。使用 Git

目录

- 团队工作。使用 Git
 - 概述
 - 典型的工作流
 - 单独设置
 - 检查状态
 - 在提交前
 - * 检查（文本文件）的变化
 - * 回滚文件
 - * 不需要的文件
 - 准备提交
 - * 添加文件
 - * 删除文件
 - 提交
 - 存储库之间同步
 - * 从远程到本地
 - * 从本地到远程
 - 解决冲突
 - * 概述
 - * 可采取的步骤
 - * 二进制文件
 - * 文本文件
 - * 纠正提交
 - 标签
 - 其他有用的命令

概述

为了组织团队工作可以使用 Git 版本控制系统。使用 Git 与其他的方式进行协作相比有大量好处

- 保存更改历史，方便的回滚到以前版本
- 同步用户间的修改，自动合并修改
- 可以用于大型的二进制文件

Git 是一个分布式的系统，每个开发人员或设计师有他自己的本地存储库（存储）。可以在本地存储库之间同步，通过中央的（“共享”）存储，放在于一个专用的机器（服务器）。对服务器的访问可以通过 SSH 协议进行。

尽管有很多图形用户界面为 Git 初学者准备，但这里解释如何使用标准控制台的 git 如何工作。

典型的工作流

1. 可以在本地仓库创建、添加或删除文件。
2. 当一段确定的工作完结后需要将修改修复（提交）和/或同步到你同事那。
3. 文件已准备好提交，即已更改的审计、新的和已删除的文件和也全部为更改重置了。
4. 执行提交。
5. 本地更改上载到共享存储并提供给同事。

一组有限的 Git 命令，建议作为创作应用程序和它们的图形资源，展示如下。

非常有必要在执行命令前切换代码库，例如

```
> cd ~/blend4web
```

单独设置

新用户可以使用这个命令设置他的姓名和电子邮件

```
> git config --global user.name "Ivan Petrov"
> git config --global user.email ipetrov@blend4web.com
```

设置数据将用于更新日志。

检查状态

建议在执行的所有操作之前，、在进步和之后都检查存储库状态。

使用此命令可以检查状态

```
> git status
```

如果所有提交了都执行并且没有新的文件 git status 命令的结果

```
# On branch master
# Your branch is ahead of 'origin/master' by 2 commits.
#
nothing to commit (working directory clean)
```

如果有变化可能的 git status 结果。例如: apps_dev/firstperson/firstperson.js 和 doc_src/git_short_manual.rst 文件更和创建了新文件 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")
```

在提交前

检查（文本文件）的变化

如果是文本文件的情况下，建议在执行提交之前查看介绍了的变化。

检查在整个目录中的更改内容

```
> git diff
```

或者特定的文件

```
> git diff apps_dev/firstperson/firstperson.js
```

一个文本文件的 git diff 命令可能的结果

```
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);
+ }
```

回滚文件

如果文件被更改或删除，但需要恢复（到最新的提交状态），请使用命令

```
> git checkout doc_src/git_short_manual.rst
> git checkout 123.txt
```

引起的变更将会被取消 - 这就是为什么应谨慎执行此命令。

不需要的文件

如果文件列在 未跟踪文件 (git 状态)，但它不需要版本控制，它应删除或移到工作目录之外。

准备提交

添加文件

如果你对所做的更改满意，添加所需的改变和/或新文件到提交。

```
> git add apps_dev/firstperson/firstperson.js
> git add 123.txt
```

再次检查状态F

```
> git status
```

git status 命令后用 git add 添加一些文件的可能的结果F

```
# 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
#
```

你可以看到 `apps_dev/firstperson/firstperson.js` 和 `123.txt` 文件添加到提交
`doc_src/git_short_manual.rst` 文件不添加。为了使事情更容易，建议使用 `git checkout` 添加此类文件的提交或取消其更改。

删除文件

在执行了 git status 命令后，一些文件可以被标记为删除，例如F

```
# 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")

```

在此情况下，如果删除该文件应该被记录下来（即输入提交），执行 git rm 命令，例如 **F**

```
> git rm 123.txt
```

如果该文件被意外删除，需要恢复它，使用 git checkout 命令。

提交

执行提交命令**F**

```
> git commit
```

文本编辑器窗口会显示（例如， nano 或 vim），必须在其中输入英文的提交评论。

```

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

```

保存更改并退出编辑器 (在 nano 是 Ctrl + O, 然后 Ctrl + X; 在 vim ZZ, 或按 ESC: wq)。

在提交后它建议重新检查状态。如果 git status 命令将返回 无可提交的东西，工作文件夹干净.

存储库之间同步

从远程到本地

所有提交执行后，需要从远程（“共享”）资源库中加载所做的更改到本地库

```
> git pull
```

如果远程资源库中的任何更改 git pull 命令的结果

```
Already up-to-date.
```

如果远程资源库中包含的更改、同步的结果是成功的话，git pull 命令的结果

```

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 lixer: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

```

如果你希望查找你同事所做的更改，使用下面的命令

```
> git diff dbf3877..9f9700c
```

此命令的参数 - 在案例 dbf3877..9f9700c - 显示哪些提交之间确切地进行了更改。此参数可以方便地在控制台的 git pull 结果中选择再用鼠标点击（中间的按钮）粘贴到需要的地方。

还可以查看更新日志

```
> git log
```

git pull 命令并不总是能同步成功。当有冲突时 git pull 的结果

```

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.

```

在冲突时所采取的步骤如下

从本地到远程

在那之后所做的更改，应从本地资源库上载到远程库（“共享”），一个能为同事提供所做的更改的库。

```
> git push
```

如果远程资源库中已经包含了所有的本地更改的 git push 命令的结果[F]

```
Everything up-to-date
```

如果同步成功后 git push 命令的结果[F]

```
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
```

如果同步不成功是因为 git pull 命令没有首先执行时，git push 的结果[F]

```
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.
```

你应该执行 git pull 命令。

其他开发人员可以使用 git pull 命令收到上传到中央存储库中的更改。

解决冲突

概述

如果这两个条件都满足，就会发生同步冲突

1. 同一文件同时在本地和远程资源库中更改，并
2. 所做的更改的自动合并没有发生因为所做的更改在文件的同一个地方。

典型案例[F]

1. 二进制文件（纹理，blend 文件）已被两个开发人员独立修改
2. 引起的修改在同一文本文件的同一行
3. 一位开发已修改该文件，而另一位又移动它，等等。

虽然同步冲突是正常，单如果太过频繁发生会导致工作缓慢。在开始共享二进制文件，处理有关，执行更多的同步时建议要通知你的同事。需要将工作分开分发给开发人员能够减少这种共享的文件数目之间的工作。要做到这一点可以特别地将场景的资源从独自的各个 blend 文件链接到一个主文件中。

可采取的步骤

存储库处于冲突状态时，不建议执行任何文件操作（修改、删除）。

第一件事是要执行 git status 命令。

```
# 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")
```

冲突的文件的列表可以在“Unmerged paths”一节中找到。

以下步骤的顺序在二进制文件和文本文件之间是不同的。

二进制文件

在这个阶段，冲突的二进制文件是和尝试同步本地资源库之前处于相同的状态。文件是完全功能的（例如他们可以被图形编辑器打开）。

有冲突的二进制文件的情况下，需要整理（与同事或自己）哪个文件应该留下，哪个要丢弃。选择可以使用 git checkout 命令执行。

选择该文件的本地版本（- -我们的）。要确保本地可以打开它。

```
> git checkout --ours blender/landscape_objects/Fallen_tree.blend
```

选择该文件的远程版本（**-他们的**）。确保远程可以打开它。

```
> git checkout --theirs blender/landscape_objects/Fallen_tree.blend
```

再次选择该文件的本地版本（**-我们的**）。

```
> git checkout --ours blender/landscape_objects/Fallen_tree.blend
```

最终你必须坚持正确的文件版本。万一这样对你的工作可能有风险，可以在存储库之外保存废弃的工作文件。

文本文件

在这一阶段 Git 特别指出了冲突文本格式的文件本地和远程的变化。此类文本的文件并不可行作为一项规则

示例。一个开发人员在应用程序文件中更改场景名称从“蓝蜥蜴”到“绿蜥蜴”并将更改上传到中央存储库中。另一个开发人员在同一行更改“蓝蜥蜴”为“红蜥蜴”，执行提交和执行 git pull 命令。最后来提交修改的开发人员将负责解决冲突。他的版本的应用程序文件会有下面这根线

```
<<<<<< HEAD
    "name": "Red Lizard",
=====
    "name": "Green Lizard",
>>>>> 81bf4e2d5610d500ad4d2a2605ee7e61f759f201
```

在文本文件冲突的情况下，可以采取下列步骤。有更改和没更改的编辑源代码的文件的两部分。另一方面，很容易重新导出已导出的场景文本文件（后缀名为 .json）。

纠正提交

选择所需的文件或编辑所做的更改后，请将它们添加为提交

```
> git add blender/landscape_objects/Fallen_tree.blend
> git status
```

添加冲突文件提交后，git status 命令可能的结果

```
# 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)
```

执行提交。建议留下默认注释F

```
> git commit
> git status
```

```
# On branch master
# Your branch is ahead of 'origin/master' by 8 commits.
#
nothing to commit (working directory clean)
```

解决冲突后，从远程资源库中的更改成功应用到本地资源库。现在-包括刚解决的冲突的本地存储库中的更改可以上传到远程资源库中，使用 `git push` 命令。

标签

标签用于指向一个确切的提交，例如，指定到一个稳定的产品版本。

查看标记的列表F

```
> git tag
```

从 2013 年 6 月 3 日创建一个标记为发行，指向同一个稳定的产品版本提交F

```
> git tag R130603 67bb597f7ed1643ed0220d57e894f28662e614e5
```

检查提交标签信息F

```
> git show --shortstat R130603
```

回滚到该标记...

```
> git checkout R130603
```

... 并回车：

```
> git checkout master
```

从远程资源库中同步标签F

```
> git push --tags
```

删除标记（如果错误地创建）F

```
> git tag -d R130603
```

其他有用的命令

检查 2012 年 1 月日志，显示文件名称但不合并提交F

```
> git log --after={2012-01-01} --before={2012-01-31} --name-only --no-merges
```

存在的问题及解决方案

目录

- 存在的问题及解决方案
 - WebGL 的支持
 - 启动时的问题
 - WebGL 的无法初始化
 - 更深入的故障排除
 - 已知问题

WebGL 仍然是一个相对较新的技术，可能与软件和硬件的各种组合下不完全正常工作。本章介绍了 Blend4Web 引擎的用户可能会遇到的问题，并提供了解决这些问题的常见问题。

WebGL 的支持

如果您使用的是台式机或笔记本电脑，您的系统必须有一个支持的 DirectX 9.0c 和 OpenGL 2.1 的 GPU，如：

- 的 Nvidia GeForce 6XXX 系列或更高。
- AMD / ATI 的 Radeon R500 (X1xxx) 系列或更高。
- 英特尔 GMA 950 或更高。

如果您使用的是移动设备上的 WebGL，请检查您的设备是否是 兼容性列表。

您还需要有支持 WebGL 技术的 Web 浏览器。

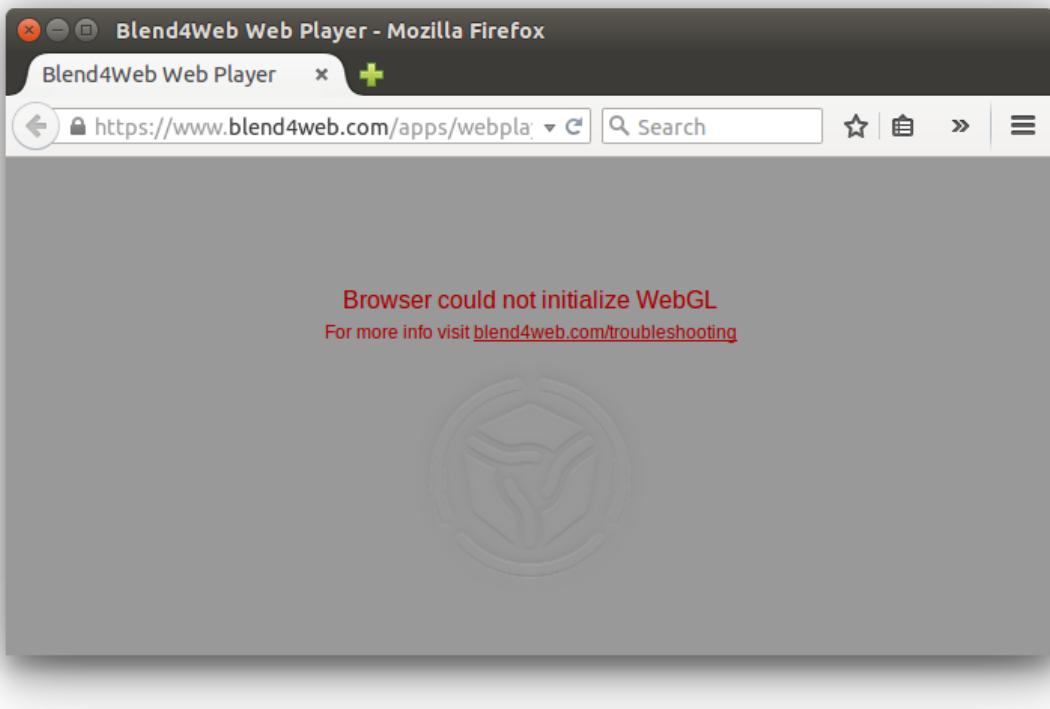
下面的 Web 浏览器支持 WebGL：

- 谷歌浏览器（诉 9 或更高版本）
- Mozilla Firefox 浏览器（V 4.0 或更高版本）
- Safari 浏览器（8.0 版或更高版本）
- Android 版 Chrome 浏览（25 节或更高版本）
- Internet Explorer（11 或更高）
- Microsoft Edge
- Opera (v. 12 or higher)
- UC Browser (experimental)
- Yandex 浏览器

我们也建议使用的最近的 web 浏览器的版本，以避免兼容性问题。

启动时的问题

1. “浏览器无法初始化 WebGL” 消息显示.



请按照[WebGL 的无法初始化](#)章节列出的说明.

2. 用户界面和背景显示但是默认场景没有渲染出来。同时 * <http://get.webgl.org/> * 网站和其他 WebGL 应用正常工作.

可能的原因:

- 该引擎尝试加载移动或删除的资源文件。
- 您正在使用旧版本的视频驱动程序。
- 您正在使用不完全支持 WebGL 的开源驱动。

对于 Linux 用户 - 由于目前的开源驱动程序不完整的 OpenGL 实现，建议使用 Nvidia 和 AMD 显卡的专用的驱动程序最新版本。

- 您正在使用过时的操作系统，如 Windows XP。
- 浏览器没有设置加载本地资源。在这种情况下，这个问题可以通过使用本地网络服务器是固定的。请参阅[加载本地资源](#)部分。

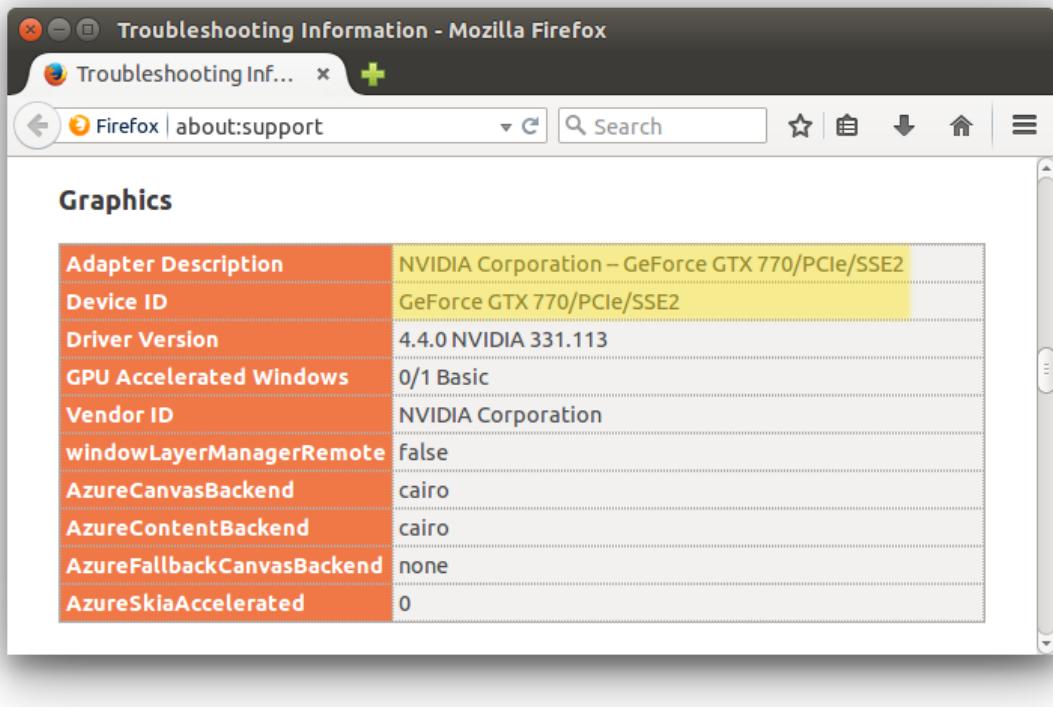
WebGL 的无法初始化

使用最近的 Chrome 或 Firefox 浏览 <http://get.webgl.org/> 页时出现的问题。我能做什么？

1. 安装最新的更新，为您的系统（用于 MS Windows 看 [指南](#)）。在微软 Windows 的情况下安装最新的 DirectX [运行](#)。重启。
2. 建议及时更新显卡驱动程序。要检测你的显卡和它的供应商，请输入 `about:gpu`（或 `chrome://gpu`），在 Chrome 浏览器的地址栏中...

Initialization time	66
Sandboxed	true
GPU0	VENDOR = 0x10de, DEVICE= 0x1184
Optimus	false
AMD switchable	false
Driver vendor	NVIDIA
Driver version	331.113
Driver date	
Pixel shader version	4.40
Vertex shader version	4.40
Machine model name	
Machine model version	
GL_VENDOR	NVIDIA Corporation
GL_RENDERER	GeForce GTX 770/PCIe/SSE2

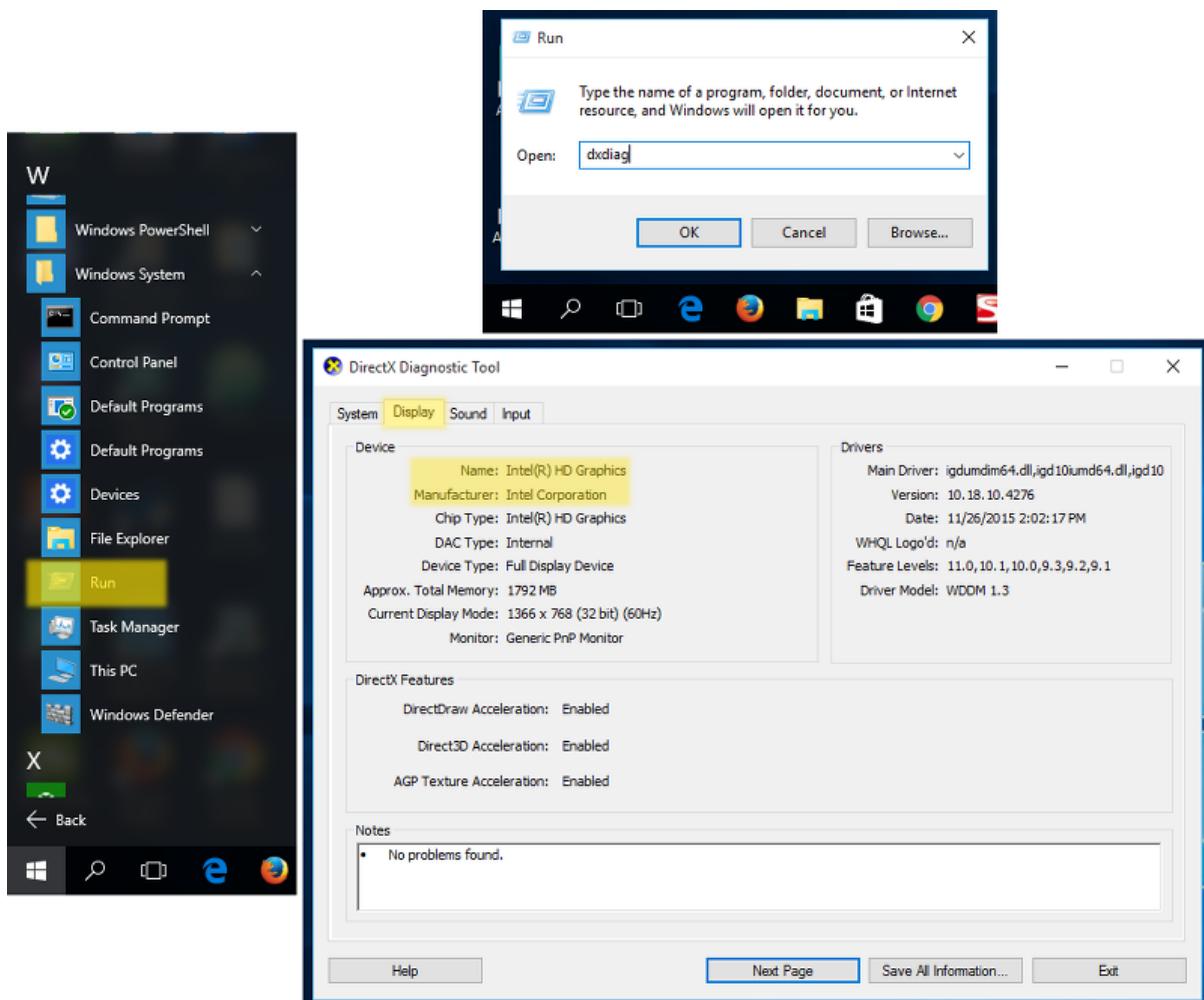
或 Firefox ...



对于 Windows，您可以运行 DirectX 诊断工具叫 dxdiag.

要做到这一点，请按照下列步骤操作：

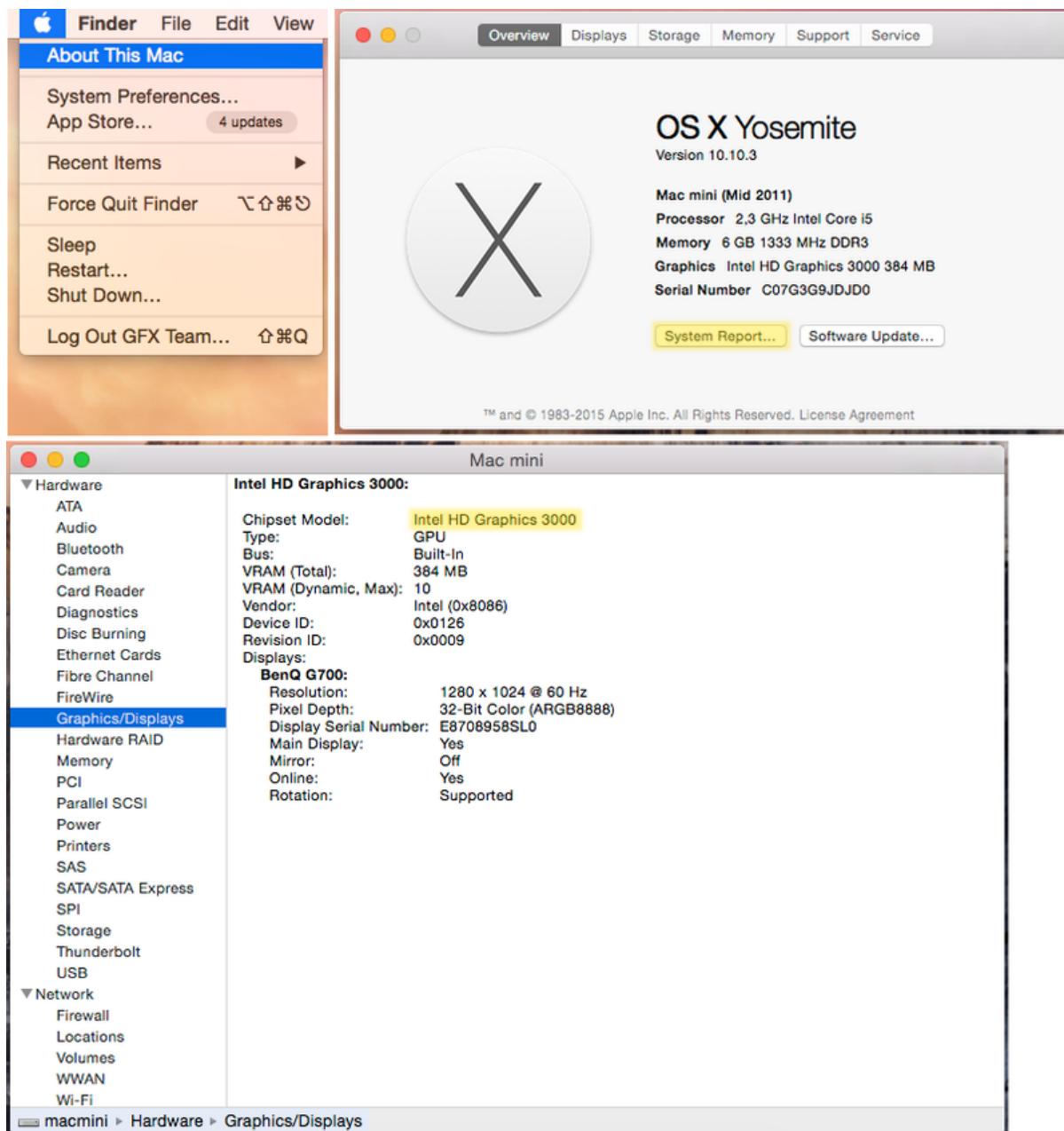
1. 从开始菜单中选择 Run 命令
2. 类型输入 dxdiag 给 Open 字段，然后按 Enter 打开 DirectX 诊断工具
3. 打开 Display 面板。在那里，你可以找到制造商，型号以及关于你的显卡的其他信息。



适用于 macOS，你可以检查系统报告。

要做到这一点，请按照下列步骤操作：

1. 从 Apple 菜单中选择 关于本 Mac。
2. 点击 系统 Report 按钮。
3. 在 Hardware 部分中选择 显卡 / Displays。



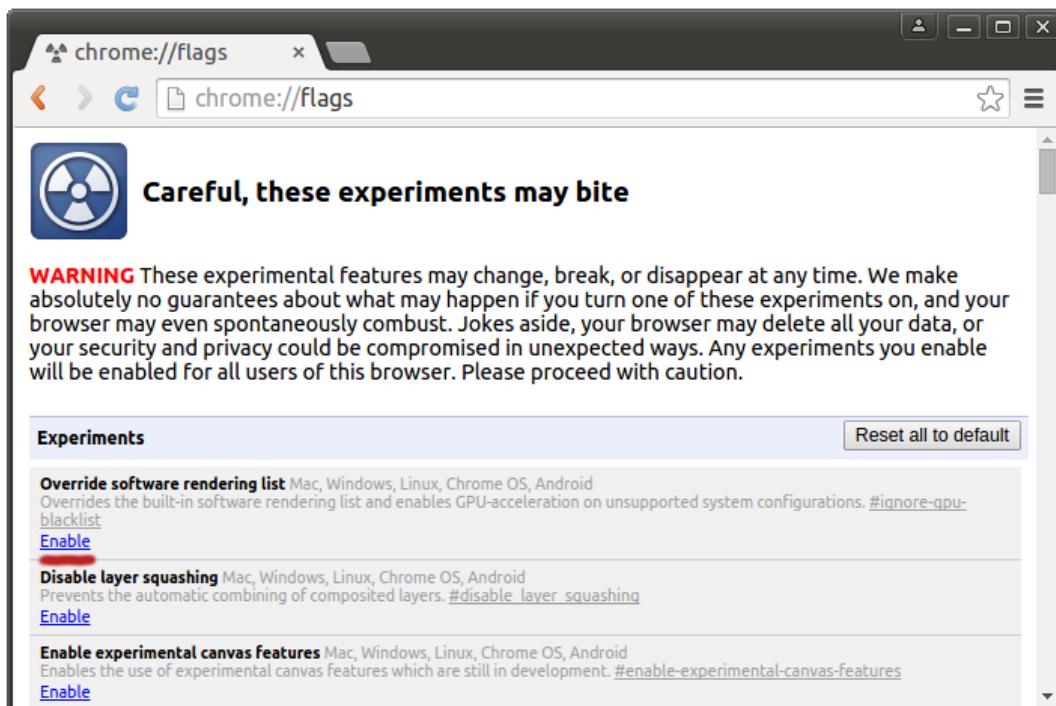
下载驱动程序从相应的支持中心（例如 [Intel](#), [Nvidia](#), [AMD / ATI](#)）。安装驱动程序后，重新启动系统。

3. 如果上述措施没有帮助初始化渲染（或者没办法更新系统）尝试改变浏览器设置。

对于 Chrome：

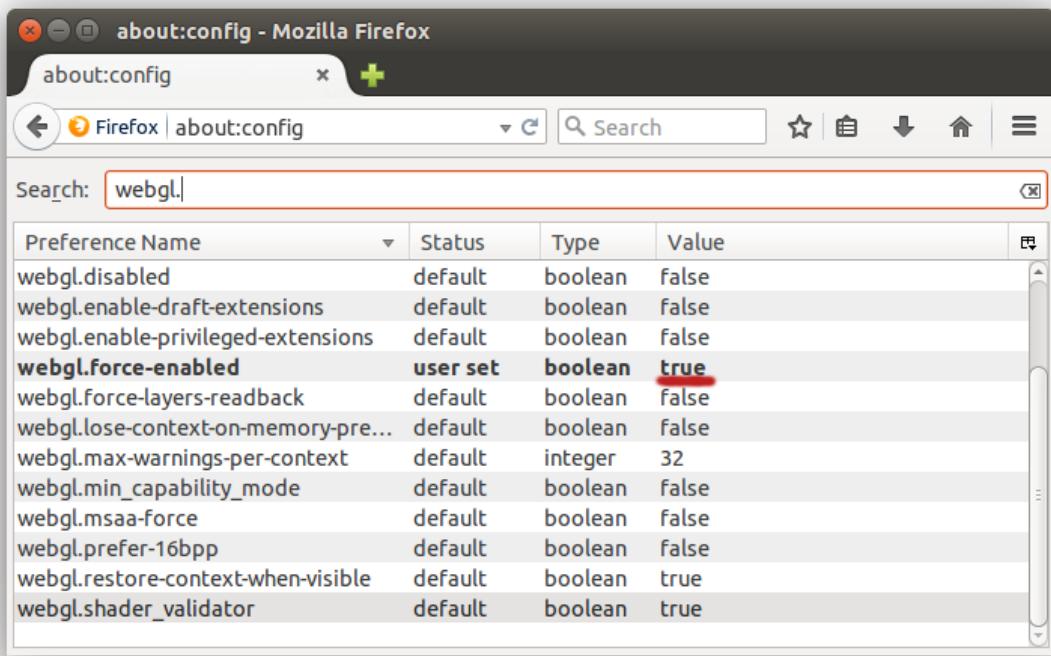
输入 `about:flags`（或者 `chrome://flags`）到浏览器的地址栏中，单击 `Enable` 下 `Override`

software rendering list 选项，并重启浏览器。



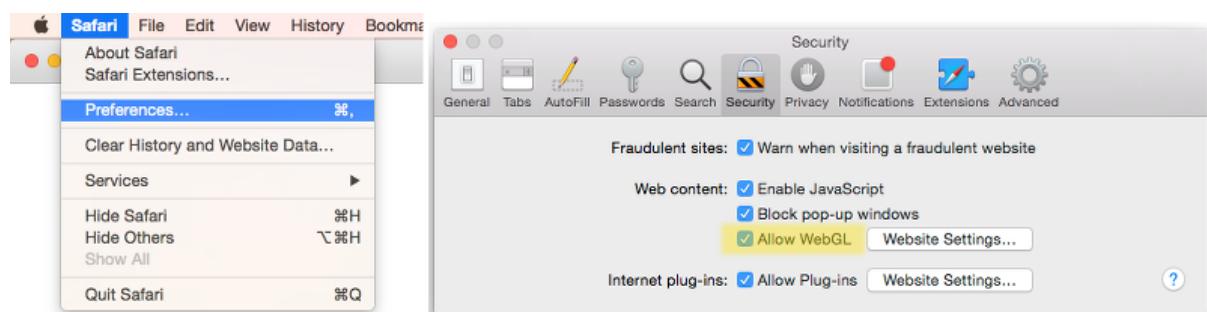
对于 Firefox：

输入 about:config 到浏览器的地址栏，搜索 webgl.force-enabled 参数，并在其上双击从 false 切换到值为 True 。



对于 Safari 浏览器

选择 Preferences 从 Safari 浏览器菜单中选择 Security 选项卡，并确保 允许 WebGL 复选框被启用。



更深入的故障排除

如果上面帮您解决您所遇到的问题都没有的话，请访问 Blend4Web 社区论坛 和 错误报告中留言 线程。我们的团队一定会帮助你。

已知问题

- 更新插件时候的问题。

强烈建议安装插件/ SDK 的新版本后重新启动 Blender。

- NVIDIA 驱动程序 331 在 Linux 中可引起 WebGL 的错误。
- 在一些平台上的改变纹理过滤。

在 iPad 和 IE 浏览器上屏蔽了一个不正确的纹理滤镜，为了让材质带 alpha clip 类型的透明类型。

- 使用 Mali GPU 的有些设备需要在浏览器设置手动激活的 WebGL。
- 对于本地开发服务器在苹果 macOS 和 Blender2.76 时，您可能需要安装 Python 3.4。这是由于在 Blender <https://developer.blender.org/T46623> 的错误。此错误已在 Blender2.76b 修复，所以建议更新 Blender。
- 使用 Nouveau 驱动时骨骼动画可能工作不正确。
- 透明纹理可能不能够在 IE11 和微软 Edge Web 浏览器和 iPad 上正确显示。

问题是关于 [transparent materials](#)。图像错误在 alpha 通道值接近或等于零的区域中可见。要解决这个问题，建议增加 Alpha 通道的值，直到错误不再可见（按价值 0.01 增加它 0.05 应足以在大多数情况下）。

- WebGL 的 Linux 上的 Chromium 崩溃 NVIDIA GeForce 400/500 系列 GPU 与驱动迟于 355。

此问题是由 Chromium 沙箱和 NVIDIA 最新的驱动程序不兼容引起的。
解决的办法是降级驱动到 340xx 版本。

- 音频问题。

音频不会在谷歌浏览器的某些版本的 HTML 导出的情况下，“背景音乐”的扬声器不工作。目前，音箱类型将自动更改为“背景声音”作为一种解决方法。

运行过多实例的情况下 (> 6)，当有音频方面的错误 B4W 引擎（例如，许多浏览器选项卡或在一个页面上的许多内部框架），谷歌浏览器。通过禁用一个应用程序的音频如果音频上下文不能为它创建防止错误。

暂停音频可能无法在谷歌浏览器版本 50-52 Nexus 6 和 9 的 Nexus 这样的移动设备上工作。它应该在 Chrome53 工作，了解详情：
<https://bugs.chromium.org/p/chromium/issues/detail?id=616772>。

- QQ 浏览器不会在此刻支持 WebGL。

- WebGL 是不稳定，有时会在 Linux 的 Mesa11.x 的驱动程序与英特尔的 GPU 上崩溃。降级到 Mesa10.x 的驱动程序可能会有帮助。
- 有关在 VR 模式上 GearVR 触摸传感器上移动手指的信息不可用。

三星的 Internet 浏览器不在 VR 模式的触摸传感器上不触发任何移动手指的任何事件。

- 由放置在无效（隐藏）层粒子系统发射的粒子可以有自己的位置设为零，作为非活性层不被 Blender 更新。

这个问题可以通过导出一个场景之前或通过简单地点击层激活层以使 Blender 更新它们来解决。

- Mali 400 系列的 GPU 上谷歌 Chrome 浏览器渲染 Alpha 混合物体不正确。

发布说明

v17.04

新功能

- Support for world materials in Cycles.

Now, sky rendering can be set up via Cycles material nodes. Background node is used to control the resulting color of the environment. Spherical environment maps with Equirectangular and Mirror Ball projection types are also supported. World materials can be animated using Value and RGB nodes both directly or by NLA. Use Nodes option for world objects is duplicated in the World properties panel.

- Experimental WebAssembly support for physics engine.

From now on you can use the optimized version of the Uranium.js physics engine (our JavaScript port of the Bullet engine) compiled with WebAssembly. The new `physics_use_wasm` config property is available to enable this feature. The feature is considered experimental and disabled by default.

- Experimental support for VR controllers.

Added `enable_controllers`. VR code snippet has been updated. The `create_gamepad_orientation_sensor` and `create_gamepad_position_sensor` methods have been added.

- Support for Alpha Anti-Aliasing blend mode for materials.

This is a variant of the Alpha Clipping technique which looks much better on compatible hardware. This mode requires that MSAA is enabled, hence it can work only on WebGL2-capable devices. In other cases, Alpha Clip blend mode will be used as a fallback.

- Support for loading of GZIP-compressed resources.

Instead of the standard ".json" and ".bin" files, the application can also load compressed files ".json.gz" and ".bin.gz". The same applies to DDS and PVR textures. This can be turned on in the Web Player by using the "compressed_gzip" URL parameter. To do this in a custom application, one should pass the "assets_gzip_available" flag in the `init` method. Compressed files can be generated by the Project Manager command "convert resources". However, GZIP-compressed resources should be used only if it's not possible to set up such compression on a server. More information can be found in the corresponding documentation section: [GZIP Compression](#).

- LOD system improvements.

From now on LOD level switching can be performed smoothly by using alpha test, which makes the transitions less noticeable. The corresponding option is called LOD Smooth Transitions and can be tuned in the Scene -> Object Clustering & LOD panel. Keep in mind that enabling this for all objects can significantly reduce application performance.

Also, a new option called Max LOD Hysteresis Interval has been added. Its intent is to eliminate continuous LOD switching when the camera moves near the border between 2 LOD levels. This option replaced the old LOD Transition Ratio parameter from the Object->Levels of Detail panel.

See the documentation for detailed information: [Levels Of Detail](#).

- RGBA shadows support.

It allows shadows on some devices that do not support depth texture.

- Bloom effect improvements.

New bloom algorithm has been implemented. Now it works with all lamp

types and is not dependent on the direction of light. Also, adaptive average luminance calculation is available.

- Supported Non-Colour Data textures in Filmic Blender.
- Fallback for compressed textures.

When a server returns the 404 error for gzip/dds/pvr textures, the engine now tries to load non-compressed images.

- Loading stages debug flag

The `init` method now has the `debug_loading` flag which allows to track and debug loading stages through the console.

- Improvements in Normal Editor.

Factor option has been added for some operators for blending the initial and resulting states of normals.

The option to use Face operator with multiple polygons has been added.

New operator Scale has been added for scaling normal along axes.

The functionality of some operators has been improved and they have been renamed in correspondence to their new possibilities: Tree -> Cursor 3D, Foliage -> Axis.

- Added API for rendering normals of dynamic objects.

The `show_normals`, `hide_normals` methods of the `debug` module have been added.

变更

- Removed warnings for the force fields of the None type.
- The `get_sky_params` method now returns null for non-procedural sky.
- The Use Custom Color property in the Mist panel is now disabled by default.
- The `append_object` method now works for non-copied objects as well.
- SSAO Influence upper limit has been increased to 2.0.

- Canvas is no longer resized to 320x240 if one or both of its sizes have zero dimensions.
- Dimensions of the cube sky render target are adjusted dynamically after an environment texture image is changed.

修复

- Improved Russian translation for addon entries in the Help menu.
- Fixed some inaccuracies in the API documentation.
- Fixed Do Not Export option for objects used as LOD levels. Previously, it didn't work.
- Fixed problem with FPS decreasing in the Viewer app after selecting an object.
- Fixed LOD switching for dynamic objects, when more than one LOD level was rendered at the same time.
- Identical packed images/sounds are unpacked during export into a single file instead of multiple identical files.
- Fixed Normal Map node strength clamping. Values less than 0 now have no influence on the normal.
- Unnecessary fog updates for world animation were removed.
- Empty objects with dynamic materials no longer crash the engine.
- Fixed texture cloning bug which caused memory leaks.
- Fixed batch sorting by alpha_clip param.
- Fixed Foam + ShoreMap texture crash in water material.
- 着色器分析仪进行了改进。
- Fixed Project Manager server behavior for nonstandard python environment.
- Fixed incorrect rendering of materials that use normal mapping.
- Fixed texture filtering for right eye in VR mode.
- Fixed crashes on IE11 in case of using `client_to_element_coords`.

- Fixed `is_hideable`, `show_object`, `hide_object` methods.
- Fixed crash for Cycles materials with non-connected Surface outputs.

v17.02

新功能

- 项目管理器改进。

为了简化项目配置一套全新的可视化配置器已经实现。要配置您的项目只需点击项目管理器的主页上的项目名称附近的 [config] 链接。项目配置器在非破坏性的方式下工作，并允许您更改当前项目的只可编辑参数。

加入克隆项目的功能。

添加 JavaScript 源映射。源映射让使用浏览器为工具的开发者可以将优化的精缩代码“映射”显示为非精缩代码。在构建项目的时候使用 `project.py` 脚本带 `-b` 标志来生成源映射。

- 用户界面的改进。

为了使创建 Blend4Web 内容的任务更有效，我们细化了 Blender 的 UI。

Render 和 Help 菜单已为 Blend4Web 渲染引擎进行修改。

已添加有关不支持编辑器的消息。

‘刚体’面板为 Blend4Web 渲染引擎已经从 View3d 工具移除。

Blend4Web 荣誉已被添加到启动画面。

- 对于 Cycles 材质的进一步支持。

对于 Displacement 输出支持已被添加。该输出是用来方便地添加凹凸材质。

透明 BSDF 节点的支持已被添加。此节点是在 Shader 类别。它是用来增加无折射透明度，直接通过表面看出去，好像几何在那个位置。

- 扩展了材质继承。

The `inherit_material` method was greatly improved to support node materials. It requires the source and the target objects to have the Dynamic Geometry & Materials option (which previously was named Dynamic Geometry) enabled on the Object panel. This option also means that all UV and vertex color layers are exported to be available to use in any material that is applied to the target object. More information about using this functionality can be found in the [corresponding chapter](#) in the documentation.

- “代码片段”改进。

克隆片段的功能已被添加。现在可以创建一个基于代码片段的新项目。它提供了使用现成的模板的进一步开发的可能性。

- 相机改进。

Horizontal 和 Auto 相机适配支持已经添加。

- 风弯曲设置 API。

Wind Bending parameters can now be set with `set_wind_bending_params`, and they can be extracted with `get_wind_bending_params`. Note that this API works only with dynamic objects. A special menu for Wind Bending was added to the Viewer interface.

- 影子模糊和反射质量配置

新参数: `shadow.blur.samples` 和 `reflection.quality` 已添加到 `config` 模块。它们可以通过 `set` 方法设置。

- 支持 Blend4Web 插件定制。

为了简化 Blend4Web 插件自定义一个新的字段 `b4w_custom_prop` 可以添加到物体或场景数据块。该选项可以通过 `get_custom_prop` 或 `get_custom_prop` 方法导出为 Blend4Web 格式而且运行时可以访问。关于如何使用此功能的例子可以在 SDK 的 `addons/blend4web/custom_prop_example.py` 文件中找到。

- 更改导出 UV 层的执行过程。

每一个网格只能有 2 个 UV 层的限制已被移除。此外，如果未指定 UV 层，则使用主控层代替列表中的第一层。未使用的 UV 图层将不会被导

出，除非物体属性 Dynamic Geometry & Materials 被启用。

变更

- 项目管理器变更。

“External” 引擎类型被废弃并通过“Copy”型自动替换。

- 带空插槽的材质能正确导出，不阻断导出过程。
- Blender 插件现在增加一个名为 Blend4Web 的独立插件类。
- 不兼容的纹理和约束现在打印了 Blender 的 UI 警告。
- 两个动画间的骨骼动画混合现在正常了。此功能仍处于试验阶段，API 可随时更改。
- 静态物理物体现在可以正确地影响一个场景当动态地加载或卸载时。
- fps 插件收到了一些改进。`enable_fps_controls` 方法现在有几个设置参数。请参阅 API 文档了解更多详情。
- API 变更。

`container` 模块的 `set_canvas_offsets`, `update_canvas_offsets`, `force_offsets_updating` 方法已被宣布为过时，并将在以后的版本中删除。使用 `client_to_canvas_coords` 或 `client_to_element_coords` 方法。

`screen` 模块已添加。

`requestFullscreen_hmd`, `enableSplitScreen`, `disableSplitScreen`, `requestFullscreen`, `exitFullscreen`, `checkFullscreen` 方法已经宣布弃用。使用 `requestFullscreen_hmd` `requestSplitScreen` `exitSplitScreen`, `requestFullscreen`, `exitFullscreen`, `checkFullscreen` 替代。

`drawMixerStrip`, `plotArray`, `shot` 方法已被宣布弃用并移动到 `screen` 模块。`hud`, `screenshooter` 模块已被宣布弃用。

`setHmdFov` 方法已被宣布弃用。

`groundProjectQuat` 方法已更名为 `groundProjectCamQuat`.

下列废弃方法已经移除:

```

camera.set_move_style,      camera.has_vertical_limits,
camera.has_horizontal_limits,
camera.move_pivot,          camera.set_velocity_params,      camera.get_velocity_params,
camera.clear_vertical_limits, camera.clear_horizontal_limits,   camera.clear_hover_angle_limits,
camera.set_look_at,          camera.rotate_eye_camera,       camera.rotate_target_camera,
camera.set_trans_pivot,      camera.set_pivot,           camera.zoom_object,
camera.rotate_hover_camera,  camera.get_eye,            camera.get_pivot,
camera.hover_cam_set_translation, camera.set_hover_pivot,   camera.get_hover_angle_limits,
camera.get_cam_dist_limits,   camera.apply_hover_angle_limits, camera.apply_vertical_limits,
camera.apply_distance_limits, camera.clear_distance_limits, camera.get_vertical_limits,
camera.get_horizontal_limits, controls.register_keyboard_events, controls.register_mouse_events,
controls.register_wheel_events, controls.register_touch_events, controls.register_device_orientation,
controls.unregister_keyboard_events, controls.unregister_mouse_events,
controls.unregister_wheel_events, controls.unregister_touch_events,
controls.unregister_device_orientation, app.resize_to_container,
app.set_camera_move_style,    app.enable_controls,
app.disable_controls,        hmd.get_hmd_device,      hmd.reset,
animation.get_first_armature_object, animation.get_bone_translation,
constraints.get_parent,      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.

```

添加了 `add_click_listener`, `remove_click_listener` 方法。

- VR 模式变更。

切换到 VR 模式现在可以无需在 VR 功能适配的浏览器（支持 WebVR 的浏览器或者移动浏览器）里重新加载应用程序来完成。

切换到 VR 模式时，相机自动旋转被禁用。

添加了对 WebVR 1.1 的支持。

- 如果应用中的物理学被禁用的话，`is_character` 方法不再返回 `true`。

- 该段非标准画布位置和方向 被重写，以反映最近的引擎变化。

此外，新的段落移动网络应用程序 添加，描述有关的方向和一个浏览器页缩放的某些方面。

- The Material API paragraph has been added to the user manual. It describes how object materials can be adjusted using API methods from the material module.
- 使用在 set_ssao_params 和 get_ssao_params 方法中的 SSAOPParams 物体的结构修改得更加一致。
- object_distance 方法已被宣布弃用，从现在开始改为使用 distance 方法。
- create_ray_sensor 和 new_event_track 方法不再接受非物理物体作为参数，因为那样会导致引擎崩溃。
- 视频纹理不再通过 change_image 方法进行改变。

修复

- 修正了当锚点描述不只包含文本节点的错误。
- 修正了由 Emitter 粒子系统使用了非有效材质崩溃的错误。
- 修正了相同的同名链接的物体和组的一些错误。
- 移除了由 Blend4Web 创建在其他渲染引擎中的重复 Simplify 面板。
- 修正了不兼容的纹理来源的材质缓存。
- 修正了发射粒子法线。
- 修正了在前一个版本中渲染到纹理实效的问题。
- 修正了几个 prefetch 方法引起的问题。
- 修正了 Ultra 模式的 SRGB 色彩校正。
- 修复了 iOS 浏览器 FPS 降低的问题。
- 修正了二进制加载在 tmp 目录不允许执行。这个问题是通过使用标准的 Blender 的 tmp 路径来修复，可配置了。
- 更正确的扩展了节点编辑点 Add 菜单。
- assets.json 已从所有 blend 文件中删除。

- 修正了按 F8 的时候二进制模块要重载的问题。
- 修正了 HMD 配置。
- 修正了在 Blender 中的场景层的可见性在导出之后被改变的问题。
- 修正了在 Firefox 中视频纹理的错误行为。
- 修正了动态草的阴影和反射。
- 修正了低质量配置被选中时的着色器编译崩溃。
- 修正了物体的选择和带节点材质物体的阴影，其中的物体有阿尔法剪辑混合模式时的问题。
- 修正了检查不支持的纹理尺寸。
- 修正了带形变键的物体的重载包围盒功能的问题。
- 修正了 LOD 物体的选择和外轮廓。
- 修正物体在 Blender 有塌陷的阵列修改器会引起引擎崩溃。
- 修复 `set_ssao_params` 方法引起的故障。
- 用于编译引擎脚本和应用程序的谷歌闭包编译器已更新到最新版本。这有助于发现和修复了几个小错误。
- 修正了物体带有毛发或发射粒子系统时应用形变键会导致引擎崩溃的问题。
- 修正了作为锚的 EMPTY 物体的“Hidden”属性。

v16.12

新功能

- 项目管理器改进。

New File 和 Save As 按钮添加到项目文件编辑器。另外，为了简化导航，编辑器高亮显示编辑过的文件。

项目管理器的应用构建工具最小化编译的 HTML 文件，以加快他们的加载。

- 新的 Cycles 材质节点。

Emission 节点的支持已被添加。此节点是来自 Shader 类别。它负责该材质的发光部分。节点输入包括 Color, Strength。在 Blend4Web, 使用此节点材质只是模拟的表面的外观和不是动态光源。

- 改进 LOD 物体的渲染。

从现在起的 LOD 物体通过它们的 LOD 距离参数和它们的比例来分批次。这样，引擎尽可能有效地渲染 LOD 物体，试图保持合理数量的来合并物体。对于调整此行为 LOD Cluster Size Multiplier 参数加入到 Scene->Objects Clustering 面板。

- 新的 API 方法 `prefetch` 和 `unfetch`.

第一个允许预加载场景（纹理，声音，场景文件）的资源，并对其进行缓存。第二允许清理缓存。

- 支持 VR 设备里的物体选择。

`pick_center` 方法已被添加。它允许用户获取物体中使用物体拾取功能的视口的中心。此方法既可以用于通用目的也能在 VR 的设备上使用。

- 改进的插件界面。

现在单独的插件不显示开发服务器面板。已添加扩展警告，即开发服务器不可用在独立的插件里。

- 纹理缓存

现在所有的图像的纹理都被缓存，并且如果请求一个新的纹理具有类似性质的，它将从缓存取出。这导致节省内存和显著加快对多个物体的 `change_image` 方法。

变更

- 项目管理器变更。

项目管理器文件的结构升级。现在，新的项目都放在 `projects` 目录。所有的资产都放在项目文件夹。

请注意 `get_std_assets_path()` 不再指定新的文件结构中项目资产目录。用 `get_assets_path()` 方法替换。

upgrade file structure 按钮添加到升级旧项目到新的结构。

现在，项目管理编辑器内的文本缩进用空格代替制表符。

- 不支持的区域类型的灯物体从现在开始导出过程中更改为太阳的类型。在这种情况下，相关的错误消息，将在浏览器控制台进行打印。

修复

- 修正了在移动浏览器的视频纹理错误。
- 修正了对于静态物体透明的 alpha 排序类型的错误。
- 修正了使用法线贴图材质渲染不正确的错误。
- 修正了一个当单个物体的 LOD 在进行切换时具有非常明显的延迟和没有 LOD 在那个情况下渲染出来。
- 修正了 WebGL 的 2.0 动态草的错误。
- 修正了 Wacom 数位板互动的错误。
- 修正了使用快速预览时“导出到不同的磁盘被禁止”的信息被显示出来。
- 修正了 DUPLI 组的导出过程中无任何附加物体时会奔溃。
- 透明材质的大于 1.0 的 Alpha 值现在正确地处理。
- PARALLAX 节点不再让节点树崩溃，如果其输入纹理有任何输出节点除了 PARALLAX 节点本身。
- 着色器节点树现在导出时预清洗，以清除未使用的节点。
- 修复在 VR 模式下视口对齐。

v16.11

新功能

- 添加了部分支持 Cycles 渲染器节点的材质。当前支持的 Cycles 节点的列表

Material Output 输出节点是类似于 Blender Internal 的 Output 除了它利用着色器类型输入的插槽。Surface 节点输入是当前唯一被支持的。此节点定义材质，如基于物理模型的渲染 (PBR) 材质。

BSDF Diffuse 节点是从 Shader 类别创建。它负责材质照明的漫反射部分，并产生不可见的反射。节点输入包括 Color, Roughness, Normal. 此节点有一个着色器类型输出插槽。

BSDF Glossy 节点出自 Shader 类别. 它是负责材质照明的镜面反射部分和环境的反射。节点输入包含 Color, Roughness, Normal. 此节点有一个着色器类型输出插槽。现在只支持反射分布类似是 GGX。这个节点自动为物体启用实时立方体反射，它使用材质而不是设置为反射。反射环境和物体应根据标准 b4w 工作流程配置。粗糙度目前不影响反射。

Mix Shader 节点出自 Shader 类别。此节点用于混合从着色类别的输出节点。节点输入包括 Fac, 定义了两个 Shader 输入的混合比。此节点有一个着色器类型输出插槽。

Fresnel 节点出自 Input 类别。此节点计算多少光反射材质的层，在那里剩余部分会折射通过层。由此产生的权重值可以用于分层着色器在 Mix Shader 节点。它是依赖于表面法线和查看方向之间的角度。节点输入包括 IOR (折射率) 和 Normal 。此节点具有一个标量类型输出插槽。

Layer Weight 节点出自 Input 类别。此节点定义通常用于与混合着色器节点的分层着色器的权重值。节点输入包括 Blend 和 Normal。节点输出包括 Fresnel 和 Facing.

其他受支持的节点包括 Image Texture, Environment Texture, Object Info, Bump.

其他部分受支持的节点包括 Texture Coordinate (From Dupli 参数不支持), UV Map (From Dupli 参数不支持), Geometry (Pointness 和 Parametric 输出不支持).

节点支持在以前的 b4w 版本，用于 Cycles 和 Blender Internal，也将在新的 PBR 材质里正常工作。此类节点包括 Color Ramp, Normal Map, Camera Data, Particle Info, RGB, Value, 节点出自 Convertor 类别（除了黑体和波长节点），出自 Vector 类别节点，出自 Color 类别节点（除了 Light Falloff 节点）。

- 项目管理器改进。

添加项目文件编辑器。现在可以在项目管理器中编辑 CSS、JavaScript、HTML 和.b4w_project 文件。

- 新的第一人称 (fps) 插件被添加。

有助于更容易创建第一人应用程序的插件。有两个主要方法: `enable_fps_controls` and `disable_fps_controls`. 第一个将创建默认的手柄和键盘控制, 启用鼠标和触摸摄像机的运动, 如果 VR 模式启用, 启动虚拟摄像机的旋转。第二个禁用这些控件。新的插件里面也有其他一些方法: `bind_action`, `set_character_state_changing_cb`, `set_cam_smooth_factor`, `set_cam_sensitivity`, `set_plock_enable_cb`, `set_plock_disable_cb`.

- 增强了 3D 导航网格。

现在 `physics` 模块的 `navmesh_find_path` 方法用来发现 3d 路径, 它使我们能够构建垂直表面上的路径。以前, 只能够建立在水平面上的路径。`navmesh_find_path` 方法的参数已被更改。导航网格初始化时间已减少。A* 算法进行了改进。

- Blend4Web 插件可用性改进。

插件中的 Blend4Web SDK 目录属性字段已被删除。现在, Blend4Web SDK 目录路径自动解析。

变更

- API 变更。

`shot` 方法的参数和 `canvas_data_url` 方法已被更改。

一个新的函数 `get_camera_angles_dir` 已经添加进 `camera` 模块。它允许从给定的方向下相机视图矢量得到相机的球面坐标。

- 简化的环境设置。

水现在使用默认风当风物体是没加在场景中。现在水物体永远是动态。太阳物体是缺席在场景中时程序天空使用默认太阳方向。

- NPC AI 现在初始化时缓存所有动画。

这有点增加加载时间, 但消除了 npc 动画造成的实时延迟。

- Fast Preview 按钮现在在 Cycles 渲染器面板出现。
- `get_all_objects`, `get_object_by_name` 和 `check_object_by_name` 方法不再返回引擎内部元物体，已经不在应用程序中使用。

修复

- 项目管理器与旧的构建类型 update 兼容性被添加。
- 在项目管理器中用特定的编码的错误得到修正。
- 在项目管理器中相同的模块名称冲突的错误得到修复。
- 镜面着色 bug 修复。

该 bug 发生在当两个或更多的灯被用于 Blinn 镜面型材质，被修复。

- NLA 卸载错误得到修复。

现在所有属于卸载场景的物体，从 NLA 移除。

- 立体闪烁 bug 修复。

使用运动模糊效果该 bug 出现。

- 使用快速预览时，在临时文件夹中用相同名字覆盖掉资源的问题被修复。
- 下载截图时网络错误得到修复。
- 修复发生在某些场景中的低质量模式的着色器崩溃。
- 修复场景没有 MESH 物体时引擎崩溃。
- 修复 `track_to_target` 以前执行不正确缩放动画的方法。
- 修正了一个罕见有关空粒子纹理插槽，这可能导致导出崩溃。
- 修正了程序线的渲染。
- 修复 Play Animation 逻辑节点当第一次播放后动画停止。
- 修复 navmesh 模块没有 ‘TypedArray’ 的 `indexOf` 方法时在 web 浏览器里崩溃。
- 修复资源转换时如果 ffprobe 找不到 Project Manager 会有无声故障。

v16.10

新功能

- 增加了导航网格的支持。

有两种方法被添加到 physics 模块：为获得最接近导航网格段的 navmesh_get_island 和为寻路的 navmesh_find_path 。

两种类型的路径是可用的：一条基于三角形中心的路径，和更优化的 - pulled string。

请参见 代码片段应用的例子。

- 新的逻辑节点 设置相机移动类型。

这个节点允许改变的相机的移动方式和速度。为 Target 和 Hover 照相机类型的目标参数可以设置为单独的坐标或设置目标物体。

- 切向着色支持编辑法线。

切线着色现在支持编辑法线。

- 提高项目管理器的可用性。

现在的应用，Blend 文件和项目资产会在一个新的浏览器卡片中打开。让你的项目管理器留在原来同一个窗口，使工作效率更高。

编译项目命令改名为 构建项目。这个新的名称更适合于该命令的性质。

- 新的环境纹理混合类型。

现在，所有的纹理混合类型，支持环境照明。

- Sphere 标志现在支持 Point 和 Spot 灯。

此标志允许，指定的光的强度下降到零的距离。

- 旋转的包围框。

现在，旋转边界框用于域剔除计算。此外，旋转椭球体边界框现在支持动态物体。

- JS Callback 逻辑节点现在可以同步调用。

当回调返回是 true 时该节点冻结整个节点树的执行，返回是 true 时节点完成执行。

- 新的 API 方法 `get_fog_params` 和 `set_fog_params`.

这些方法允许在一个场景控制薄雾。雾参数包含 `fog_intensity`, `fog_depth`, `fog_start` 和 `fog_height` 属性。

- 在新的 API 方法 `update`.

这种方法允许更新锚位置。

- 增加了对不同类型的 VBO 缓存的支持。

有些属性被更改为类型 UNSIGNED BYTE 和 SHORT 而不是 FLOAT 无质量损失并降低总 GPU 内存的消耗。这也会影响性能，可以增加某些演示的帧速率。将导出的顶点色从 SHORT 变为 UNSIGNED BYTE 类型随着导出的 .bin 文件的大小也略下降。

- 在 ULTRA 模式正确的 SRGB 转换函数现在用来代替一个简化的。

效果在暗区比较明显，其中简化函数不产生精确的颜色。

- 在 3D VIEW 面板添加了特别的按钮来启用/禁用 世界背景选项。

这些按钮布置在 World 标签，如果 Render Sky 选项已设置应该被使用。启用 世界背景显示在视口中的世界颜色（和在引擎中一样）。

变更

- 在 SDK 内重构项目。

现在所有项目（包括教程）在 SDK 包内遵循标准的项目管理器目录层次结构。

- 从预览器应用程序中删除场景列表。

同样的功能（浏览和查看项目资产）可以使用项目管理器进行。

- API 文档 `input` 模块已增强，增加了使用功能的示例。
- 降低了 `batches` 数从而优化场景渲染。

这种优化主要影响阴影和头发粒子。使用此功能增加了场景的帧速率。

- 现在，在默认情况下键盘设备连接到文档物体。
- 新增 None 作为一个新的引擎绑定类型。

这意味着在构建阶段项目管理器不会改变你的项目。

- 删除了 Update 引擎绑定类型。

使用 Copy 引擎绑定类型和项目管理器的部署功能做相同事。

- 新增轻量级的 SDK 版本。

现在 Blend4Web PRO 和 Blend4Web CE 的 SDK 可用轻量版本，其中不包括演示应用程序和教程。这些版本建议给互联网连接速度较慢的用户。

- 为 Intel HD Graphics 3000 显卡启用深度纹理，它允许在此设备上启用使用这种效果的阴影，上帝光芒，景深和其他一些特效。
- 现在，NLA 动画从 顶点动画本身的帧的开始/结束取值。

这使得一个物体的多个顶点动画可由 NLA 控制。

- 一些移动设备（包括 IOS）现在不强制使用低质量的节点材质。
- API documentation for `BloomParams` has been added. Some parameters have been renamed.
- 更改 `mouse_click`, `MOUSE_MOVE`, `touch_click` 和 `TOUCH_MOVE` 传感器装置。现在是一个字典，包含绝对坐标（coords）对上述所有；`which` 为点击鼠标；`gesture` 为 `TOUCH_MOVE`。
- `default_AND_logic_fun` 和 `default_OR_logic_fun` 逻辑函数现在 `controls` 模块中可用。
- 删除了二进制模块交叉编译的过时脚本。
- API 方法 `set_nodemat_value`, `get_nodemat_value`, `set_nodemat_rgb` 和 `get_nodemat_rgb` 已经废弃并移动到 `material` 模块。

修复

- 修正了 `quat_to_euler` 函数。
- 修复 法线贴图节点用非单元强度参数的不正确的行为。

- 修正了在逻辑节点中物体的某些运行时检查。
- 修正了当多种材质之间共享纹理时无法对一个物体修改纹理 (`change_image` 函数)。
- CookTorr 镜面模式现在看起来跟在 Blender 中一致。
- 在非深度复制物体中 Alpha Sort 材质现在可以正确地表现。
- 修正了 `create_pline_from_point_vec` 和 `set_pline_initial_point` 方法。
- 基于 `npc_ai` 模块行为的行走角色现在不会掉进地底下了。
- 修正了当两个或两个以上锚类型为 Custom Element 可以引用同一元素的 ID 的错误。
- 修复球面广告牌反射。
- 在 Firefox 修正了在暂停了 背景声和 位置声音箱然后恢复声音出的问题。

v16.09

新功能

- 网络播放器改进

设置一个放置在加载场景的右下角的社交网络按钮的选项。要做到这一点，你需要在应用程序启动之前指定 `socials 属性`。

- 坐标系统变更。

现在使用 Blender 的坐标空间来替代使用 OpenGL 的。这一新行为可能在应用程序逻辑中引入各种不兼容的情况。建议开发人员查看他们的应用程序并进行更改，根据新的坐标空间 (Z 矢量向上)。

- 优化的几何渲染。

通过实施一种在 GPU 内存中存储数据的新算法进行了优化几何渲染。现在法线和切线数据已存储于 TBN 四元数。

- 项目管理器改进。

添加新的 ‘update_modules’ 命令到 project.py 实用程序。此命令允许用户更新内部开发的应用程序的引擎的模块。此功能大大简化了更新项目文件到较新的 Blend4Web 版本的过程。

- 自动质量侦测

通过在 app 模块指定自动 (P_AUTO) 质量配置文件，可以具体取决您的硬件规格然后加载低或高质量的应用程序。此外，低级别性能基准测定可以使用执行 test_performance 方法。

- PVRTC 转换支持。

现在支持 PVRTC 转换。这允许开发人员在 iOS 和基于 PowerVR 的 Android 设备上使用压缩的纹理。

- 增加阴影质量设置。

现在就可以设置不同的质量配置的软阴影 [16 x, 8 x, 4 x]。

- Render 标签里添加了 Shading 面板。它包含了 World Space Shading 选项和 Set Recommended Options 按钮来自动配置 Blender，为更好的 Blend4Web 体验。
- 添加了对 Blender 2.78 的支持。

World Space Shading 选项搭配环境照明支持使得在视口中的 GLSL 模式渲染和 b4w 的视口预览最大程度的一致。

完全支持新的 Normal Map 着色器节点。

变更

- 指定着色器指令系统被重构。此功能可以减少引擎加载时间和简化调试。
- 在某些设备上使用 “translate3d” CSS 属性渲染锚被加快了。
- 着色器验证已得到改进，从而降低假阴性的错误的数量。已添加更多的错误说明。
- 着色器计算的主要部分从矩阵转到 TSR 来提高性能。
- 渲染选项卡面板的顺序被重新安排。

修复

- 修正了如果在场景中有许多锚会在三星互联网浏览器中崩溃的问题。
- 修复了网络播放器“alpha”属性不起作用的问题。
- 修复编译/链接着色器错误消息。
- 修复了在 Transform Object 逻辑节点中的变量值定义的角度旋转问题。
- 修复了 Play Animation 逻辑节点在烘焙和非烘焙版本的动作并存时的行为问题。
- 修复平面反射的质量设置。
- 修复了 `get_translation_rel` 和 `get_rotation_rel` 方法。
- 修复 `set_dof_params` 方法是 `dof_distance` 设置。
- 修复了当同时启用 Do Not Render 和 Enable Outlining 两个选项的物体在被选择后导致引擎崩溃的问题。
- 修复粒子系统的物理

v16.08

新功能

- 现在支持材质的 切向着色选项。

此选项可以用于模仿各向异性材质的表面，抛光的金属、头发等。

- 现在支持着色器节点 Normal Map 。

此节点允许颜色数据从纹理转换为法线贴图。输入的数据可以在切线、物体或世界坐标空间。Strength 参数控制从物体的默认法线与纹理所提取的法线的混合值。

纹理图像的色彩空间应设置为 Non-Color，使 Blender 的视口预览与最终 Blend4Web 场景中渲染的效果一致。

- 引擎 CPU 优化。

有在引擎的核心系统进行了几个 CPU 和 GC (垃圾回收器) 优化。OES_vertex_array_object 扩展在相应的硬件上获得支持。统一和着色器的使用也被优化以减少的 WebGL 调用总量。这应该改善场景加载时间和渲染响应能力，特别是在慢的 CPU 上。

- 查看器应用程序的界面改进。

Tools & Debug 面板已被移动，现在位于 Scenes 面板下面。

Min capabilities mode 按钮已经被添加到 Tools & Debug 面板。此参数使预览器应用程序能够运行加载的场景，就好像它在低端的系统（如 iOS 设备）上运行。这允许艺术家要找出哪个场景中的材质可能无法按意图在低端的配置工作。

- 法线编辑器的改进。

为法线编辑器添加 Offset 模式。

添加 Average 操作支持非分裂法线。

增加键入法线旋转角度就像输入物体的旋转角度一样。

- 在新的 API 方法 `textures` 模块。

添加 `get_texture_names` 方法。它使我们能够得到的所有物体的纹理名称。

- 在 `lights` 模块中新的 API 方法。

添加 `get_light_color`, `set_light_color`, `get_light_energy` 和 `set_light_energy` 方法。这些用来处理一盏灯的颜色和能量值。

- Hidden 物体属性的支持。

此标志将会在场景加载时隐藏物体。

- 对 GLSL ES 3.0 着色器的支持。

从现在起引擎自动选择应该用哪个版本的 OpenGL 着色语言来编译/链接着色器。它取决于 WebGL 上下文。WebGL 1 使用 GLSL ES 1 版本和 WebGL 2 用 GLSL ES 3。引擎的渲染系统和宏预处理器也改为与这些更改兼容。

此功能解除对执行有关 WebGL2 的不兼容问题的新功能的限制。

- 着色器分析仪进行了改进。

着色器分析仪 (方法 `analyze_shaders`) 现在打印低级程序代码集有助于审查和优化着色器。

变更

- 景深 (DOF) 散景效果算法的改进。

强度泄漏（或像素出血）瑕疵，当焦点的前景物体出现‘leak’到模糊的背景，减少了。

添加 Foreground Blur 属性。当启用时，它减少了非聚焦的前景物体在聚焦的背景上出现尖锐的外轮廓显示。

前景的 Front Start, Front End 属性和背景的 Rear Start, Rear End 属性，使我们能够指定一个模糊开始，并达到最大值的距离。

- 默认情况下反射世界。
- 着色器验证算法改进提升。

如果材质不能在低端的设备上渲染，取而代之的是调试错误（粉红色）材质，它会在生产环境删除。

修复

- 修复转化顶点组长度。

修复了为顶点组长度的毛发粒子系统的转化操作。

- 修复了移动物体在法线编辑器中的法线旋转问题。
- 修复了罕见的使用了逻辑节点的场景在加载的时候会让引擎崩溃的问题。
- 修复查看器应用程序中材质面板。有些材质，允许进行编辑却被禁用的问题。
- 修复在移动设备上引擎崩溃问题。
- 修复了移动浏览器上有闪烁问题。
- 动态水物体加载崩溃已得到修复。
- 修复对材质的反射的法线影响。
- 当在 iPad 上敲击网络播放器时禁用灰色高亮。

v16.07

新功能

- Optimized Particle system rendering.

Now WebGL instancing capabilities are used (provided by the ANGLE_instanced_arrays extension or WebGL 2.0) to render Object-type particles. This type of rendering is more memory efficient and, in some cases, also improves rendering performance.

- Reduced input latency on mobile devices.

Now the engine ignores mouse events that represent actions that already have been handled by internal Blend4Web touch-event handlers on mobile versions of Chrome, Firefox, Safari. It reduces delays in user actions.

- Audio system improvements.

Automatic audio context creation. The Audio checkbox has been removed from the addon. If necessary, scene audio context is created automatically.

Doppler effect implementation. In recent versions of WebAudio spec Doppler effect was removed. It's now supposed that application developers should implement this effect themselves. From this release you can use in-engine implementation. A new option called Enable Doppler has been added, replacing the deprecated Disable Doppler.

Reworked audio interface. New settings have been designed to be as close as possible to the native Blender settings. This includes support for such settings as Speed, Doppler and Distance Model.

New Auto-play speaker option. This option enables speaker playback by default.

Preliminary support for complex audio loops. Using new Loop Start and Loop End options as well as `loop_stop` API method you can create complex audio loops, which include start, loop and stop sections in one audio buffer. For example, you can create basic ADSR (attack, decay, sustain, release) envelopes using this new API.

- Support for multi-touch selection in selection sensor.

Now selection sensors are able to use multi-touch selection in Event-Driven Model.

- Node materials animation improvements.

Now any node animation can be applied to any node material or its nested node groups. Also, there is a new `apply_ext()` method that allows specifying material or a node group that is to be animated.

- A new method has been added into the `preload` module: `create_preloader`.
- Configuration parameters `max_fps`, `max_fps_physics`, `use_min50`, `anisotropic_filtering`, `shadows`, `reflections`, `refractions`, `ssao`, `dof`, `god_rays`, `bloom` and `motion_blur` have been added to the `config` API module.

变更

- Color picking optimization.

Now color picking uses very narrow frustum and a small framebuffer size (1×1 pix). Also, amount of `pick_object` calls has been reduced. This improves performance of selection sensor.

- Depth-of-field (DOF) effect improvements.

DOF effect performance has been increased.

An experimental DoF effect algorithm has been added. It varies the amount of blur depending on depth and produces bokeh effect on blurred objects. New algorithm can be enabled via Bokeh property from the camera properties panel in Blender.

- API 变更。

The `create_simple_preloader` method of the `preload` module has been declared deprecated and will be removed in future releases.

修复

- Fixed the broken Canvas Resolution Factor slider in the Scene Viewer.

- Fixed `get_matrix`, `set_matrix`, `get_matrix_rel`, `set_matrix_rel` methods of the transform module.
- Fixed definition of the `resize_to_container` method of the container module.
- Fixed performance regression caused by resizing the canvas.
- Fixed cameras linked from other scenes or dupli-groups not present in the scene.
- Fixed incorrect canvas alpha with Bloom post effect.
- Fixed Viewer Stop All animation button.
- Fixed object picking for stack material.
- Fixed glow effect on Safari.

v16.06

新功能

- Fast Preview improvements.

If necessary, the development server copies all external resources into the `tmp` directory. This allows previewing scenes which are placed outside the SDK (another directory, flash drive, etc).

- 项目管理器改进。

Support for material library. A new project option has been added allowing users to copy the material library sources into the project directory.

Added the new `--ignore` command property to the `project.py` utility. This option allows users to ignore files during compilation or deployment.

Added the new `check_modules` command to the `project.py` utility. This command allows users to check missing or no longer required modules.

- Experimental support of the GearVR virtual reality headset.

Support for new WebVR API 1.0 has been added to the engine allowing the use of GearVR devices.

- Support for GIF and BMP textures.

Non-animated GIF and BMP images can now be used as textures.

- Resource Converter improvements.

Now many more media extensions are supported. For detailed information see [the documentation](#).

- Automatic export path determination in Blender addon.

When projects, created using Project Manager, are exported for the first time, a path to the assets directory is automatically resolved.

- Support for parallel animations in the Logic Editor.

It is now possible to apply several parallel animations with the Logic Editor. Previously, only one animation per object was allowed in the Logic Editor. Now, an object can have one animation of each type. The maximum number of possible animations is 8.

- Improved Viewer profiling capabilities.

Added a special mode for profiling objects' rendering time in the Viewer application.

- Lens Flare material property.

A new material property has been appended to the material render panel. Note, that this works only when there is a Sun light source in a scene.

- Clip Start and Clip End options for light sources.

The Clip Start and Clip End properties have been supported for the shadow settings of Point and Spot lamps.

- HTML meta elements in Web Player app.

New HTML meta elements have been added into the WebPlayer HTML templates for compatibility with different social networks.

变更

- Several material nodes are now using Blender's viewport world space.

- Geometry

The Normal output provides data in Blender's world coordinate space.

The View output provides data in Blender's view coordinate space.

- Texture

The Vector input for environment textures receives data in Blender's world coordinate space.

The Normal output provides data in Blender's world coordinate space.

- Material

The Normal input receives data in Blender's world coordinate space.

The Normal output provides data in Blender's world coordinate space.

- Lamp Data

The Light Vector output provides data in Blender's world coordinate space.

- B4W Vector View

The node's input receives data in Blender's world coordinate space.

- B4W Reflect

The first node's input (used for view vectors) receives data in Blender's view coordinate space.

The second node's input (used for normals) takes data in Blender's world coordinate space.

The node's output provides data in Blender's world coordinate space.

- Now color picking and anchors are disabled in stereo-mode.
- Now keyboard sensors do not active when using browser shortcuts.
- API 变更。

The `resize_to_container` method of the `app` module has been declared deprecated and will be removed in future releases. `container` module's `resize_to_container()` method should be used in its stead.

- The Bloom Key option has been renamed Intensity.
- Deprecated functionality.

The following methods: `textures.get_canvas_texture_context()`, `textures.update_canvas_texture_context()` have been removed.

The following logic nodes: Select, Select & Play Timeline, Select & Play Animation have been removed.

- Export errors and warnings now include links to the documentation.
- Proper reporting of incorrect addon directory.

An incorrect addon directory name now generates a corresponding warning message.

- Dealing with cases when water is used without wind.

If the water shader is used without wind, a warning message appears.

- Debug console error messages for unsupported image, video and audio formats have been added.

修复

- Fixed decreased performance in stereo-mode.
- Fixed `window.screen.orientation.angle` obfuscation.
- Fixed rendering issue in Firefox browsers with enabled WebGL 2.0 context.
- Fixed right-eye rendering in stereo-mode.
- Fixed rendering for glow materials with the Terrain Dynamic Grass option enabled.
- Removed redundant angular velocity of particles if the Rotation option is disabled.
- Fixed rendering particles with non-node, non-Opaque materials.
- Fixed several GPU memory leaks.
- Fixed engine crash when using wrong callback id in the JS Callback logic node.
- Fixed engine crash in the rare case, when a node material has several Texture nodes with the same texture.

- Fixed the bug which caused the Diffuse Intensity input of the Material and Extended Material nodes to ignore the incoming link.

v16.05

新功能

- Experimental HMD configurator.

Add-on `hmd_conf.js` has been added. It allows users to set up HMD parameters manually. For now, this configurator has been implemented in the Viewer app and can be executed by the button located on the Stereo View panel. For more information see API documentation on the `hmd_conf` module.

- Experimental configurator for gamepads and other game controllers.

Add-on `gp_conf` has been added to API. It contains two main functions: `show` and `hide`. For now it supports controllers and steering devices.

To test this configurator in the Viewer app activate the Gamepad Settings check box on the Input Devices panel.

- New code snippets.

New code snippet demos have been added. Their names are Gamepad, Webcam and Change Image. The first one can help you to integrate gamepads into your project, the second shows you the Camera API in action and the third one is the implementation of the new texture functions.

- A new Vector Transform material node.

This node can be used to convert input data between object, world and camera coordinate spaces.

- A new Empty logic node.

This node can be used to simplify rerouting complex logic node configurations.

- Dynamic replacement of texture images.

A new method `change_image` has been added. It's possible now to replace textures and cubemap bitmaps via API.

- Batching based on a new clustering algorithm.

Batching, that is, the process of combining similar objects for performance reasons, is now based on a new clustering algorithm applied at export. This improves the whole batching process by making it “smarter” and more optimized in comparison with the old regular grid batching.

- 项目管理器改进。

To maintain naming consistency all command line options in `project.py` utility are now specified with `-` symbol instead of underscore, e.g `--engine_type` option has become `--engine-type`.

Added the new `--assets-dest` command property to `project.py` utility. This option allows users to specify a destination directory for storing assets in deployed projects.

- New materials in Material Library

4 new materials have been added to the Pro SDK Material Library: Gold, Ribbed Glass, Silk and Velvet.

变更

- UI and export usability improvements for objects with no option from the Export Options list selected.

`Export Shape Keys` is selected automatically after adding a shape key.

`Export Vertex Animation` is selected automatically after baking vertex animation.

`Apply Scale and Modifiers` is automatically used during the export process for non-uniform scaled objects, which meet the following requirements:

Has no vertex animation.

Has no parent object.

Has no skinning.

Object physics is disabled.

Apply Modifiers is automatically used during the export process for objects, which have modifiers and meet the following requirements:

Has no vertex animation.

Has no skinning.

- Z sorting in Alpha Sort materials is now performed based on the sizes of objects.

This can lead to more frequent sorting updates which impairs performance, but fixes sorting issues on small objects.

- New frustum culling algorithm.

Now we calculate frustum culling using specific mesh materials instead of objects.

- Gamepad sensor changes.

New button and axis identifiers have been added to the `input` module. Also, gamepad sensors have been renamed as `create_gamepad_btn_sensor` and `create_gamepad_axis_sensor`.

- Entry Point logic nodes with the Run From Script option selected can now be called from API multiple times.
- API 变更。

The `create_rotation_preloader` method of the `preload` module has been declared deprecated and will be removed in further releases.

The `create_advanced_preloader` method of the `preload` module has been declared deprecated and will be removed in further releases.

The `set_config` method has been added to the `input` module.

The `register_device` method of the `input` module has been declared deprecated and will be removed in further releases.

The `gyro_use` flag of the `config` module has been removed (not needed anymore).

Now functions of the `storage` module have an optional last parameter.

The `is_ie11` method has been added to the `util` module.

- Now the state of a keyboard button is stored in the payload of the keyboard sensor.

For more information see API documentation on the `create_keyboard_sensor` method.

- The link to the troubleshooting page in the WebPlayer “Browser could not initialize WebGL” message has been changed to a more appropriate one.

Now it refers one to the “Problems and Solutions” page in the documentation.

- The documentation has been considerably expanded with detailed descriptions for Anchor and Viewport Alignment features.

修复

- Fixed issue with incorrect automatic updates in addon.
- Fixed anchors behavior with logic nodes.
- Fixed JS Callback logic node obfuscation.
- Fixed issue with incorrect sensor removal in `controls.remove_sensor_manifolds` method.
- Fixed issue with incorrect age and compression ratio in distribution zip files.
- Fixed multi-sampling issue in Firefox browsers.

Disabled partially supported multi-sampling in Firefox WebGL 2.0 implementation.

- Fixed error when canvas resolution wouldn't change when Anti-Aliasing quality settings were changed.
- Fixed error when the diffuse intensity value would be applied to shadeless materials.
- Fixed engine crash in the case when `B4W_GLOW_OUTPUT` and `B4W_REFRACTION` nodes were used in the same material.

- Fixed the duplication of the Custom Properties panel in Blender's interface for the Font, Curve, Lattice, Armature and MetaBall data types.
- Fixed some errors in the compiled version of the engine.
- Fixed issues with input and controls modules.
- Fixed issue when node materials using the ColorRamp node were incorrectly batched.
- Fixed incorrect HTTP server shutdown.

v16.04

新功能

- Support for gamepads and controllers.

It's possible now to use gamepads and controllers as input devices. New functions have been added to work with these devices. The first one is `create_gamepad_btn_sensor`. It handles gamepad buttons. The second one is `create_gamepad_axes_sensor`. It handles gamepad axes.

- Node Logic Editor improvements.

Logic node JS Callback has been added. It allows to call custom JavaScript callback defined in your B4W application. Input and output parameters are supported for callbacks.

An option Run From Script has been added to Entry Point node.

Module `logic_nodes` has been added to API. It contains methods to control Node Logic Editor.

Method `append_custom_callback` has been added. It allows to register custom JavaScript callbacks to be used in JS Callback logic node.

Method `remove_custom_callback` has been added. It allows to remove registered custom JavaScript callback.

Method `run_entrypoint` has been added. It allows to activate Entry Point node from API.

- A new function has been added into the controls module: `create_hmd_position_sensor`.

This function allows to create a special sensor, which can track the position of an HMD device.

- The options Tilt Angle and Tilt Random are now supported for the Emitter particle systems.
- Reflection for transparent objects.

Before this release only opaque objects could be reflected. Now, transparent objects are also supported.

- Updated math modules.

Math modules `vec3`, `vec4`, `quat`, `mat3`, `mat4` are now based on `glMatrix` v2.3.1. This new version introduces `hermite`, `bezier`, `slerp`, `fromRotationTranslationScale` and `fromRotationTranslationScaleOrigin` methods.

- A new flag Bake only deform bones has been added to the Skeletal Animation Baker.

Previously, it was impossible to bake bones without the deform flag. This feature can be helpful in cases when some object is parented to the non-deforming bone.

- New GL Debug switch in the Viewer app.

Viewer application now has the GL Debug switch which allows to disable GL error checking. This increases the performance and can be useful while profiling a scene.

- Enable WebGL 2.0 for Firefox browser.

It's now possible to use experimental WebGL 2.0 context in Firefox browser.

- New `prevent_caching` engine configuration option.

This option enables/disables assets caching.

- New physics function has been added.

`apply_force_world` function applies a constant force to the object in the world space.

变更

- Viewer's HUD Info now has detailed info on each column.
- Node Logic Editor changes.
 - An option to select between Number and String operand types has been added to the Conditional Jump node.
- The activation of the VR mode from now automatically changes the camera type to EYE for better user experience.

修复

- Fixed anchors behavior.
- Fixed a cubemap issue for some old NVIDIA GPUs in Firefox.
- Fixed the bug for particle systems with the “Length” vertex group specified when the emitter mesh has the “Apply Modifiers” option checked.
- Fixed the behavior of the B4W_GLOW_OUTPUT node for transparent materials.
- Fixed grass map for a single flat grass terrain object.
- Refraction vectors for stack and node materials now use correct view normal.
- Fixed God Rays Blender interface tab.
- Viewer Sky parameters updates have been fixed.
- Fixed incorrect stereo (anaglyph and HMD) rendering in specific cases.
- The Page Param node bug, which always wrote result to the variable R1, has been fixed.
- Updating variable's scope in Node Logic Editor has been fixed.
- Fixed crash with dynamically loaded scenes while using Move Camera logic node.

v16.03

新功能

- Node Logic Editor improvements.

Now you can use global variables for sharing information between threads.

This significantly expands the capabilities and allows to create more complex interactive applications.

Logic node JSON has been added. It allows to parse and encode complex JSON objects.

Logic node Get Timeline has been added. It allows to get current frame from NLA or global timeline.

Logic nodes Play Animation and Stop Animation have received environment animation mode.

- New option Update Material Animation for updating animated node shaders in viewport.

This option is useful for those who often use animated shader nodes. Just turn it on to see material animation in Blender viewport.

- Separate Project Manager server from Blender addon.

Starting from this release it's possible to run the Project Manager server as a standalone application using `project_server.py` script. This eliminates the requirement to start Blender to be able to run the Project Manager.

- Add title/description to all Blend4Web demos.

This simplifies searching our demos in Google and other search engines.

- New modules have been added.

The first new mathematical module that was added is called `math`. For more information, please take a look at [the API doc](#).

Also, an `input` module has been added. This module provides an interface for input devices: mouse — `DEVICE_MOUSE`, keyboard

— DEVICE_KEYBOARD, touchscreen device — DEVICE_TOUCH, gyroscope device — DEVICE_GYRO, head-mounted device — DEVICE_HMD. For more information see API documentation on the input module.

- Changed policy for backward compatibility with previous Blender releases.

Starting from this release we will strive to keep the addon compatibility with previous Blender versions.

- HTML links have been supported in an anchor description.
- Objects from secondary scenes(rendered to textures) can now be controlled with logic nodes.
- The Dynamic Grass option was added to the render interface.

There are 3 options available: ON to enable dynamic grass constantly, OFF to disable dynamic grass completely and AUTO to automatically detect objects with dynamic grass.

变更

- API 变更。

The `enable_controls` and `disable_controls` of `app` add-on have been declared deprecated.

Several methods of `controls` module have been declared deprecated: `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`.

The `create_touch_click_sensor` and `create_hmd_quat_sensor` methods have been added to the `controls` module.

The `get_hmd_device` and `reset_device` methods of `hmd` add-on have been declared deprecated.

Several methods have been added to `util` module: `deg_to_rad`, `rad_to_deg`, `quat_to_ordered_angles`.

The `get_world_by_name` method has been added to the `scenes` module.

- API for camera has been changed.

The `calc_ray()` method was changed. Now it works with parametric lines.

- Static physics behavior has been changed.

Objects, which have material with the `Material -> Special: Collision` property enabled, can be fully unloaded. Also, these objects can be moved, rotated, etc, as long as they are dynamic.

- Rendering optimizations.

Now we use bounding ellipsoids instead of spheres to frustum cull static objects.

- Node Logic Editor changes.

Node `Send Request` has been simplified. JSON parsing and encoding routine has moved to new JSON node.

- Material updates, e.g. animated or affected by dynamic lights were optimized.
- Generated water mesh has been enabled on devices without `OES_depth_texture` support

修复

- Fixed webplayer menu behavior.
- Fixed rare development server crash when the response headers contained a specific date (29Feb).
- Fixed Refractions which was set to “ON” in the absence of refractive objects.
- Fixed water material position for dynamic objects.
- Fixed crash for generated water mesh without waves.
- Fixed incorrect output for the `TEXTURE` node with no texture selected.

v16.02

新功能

- Extended support of Emitter type particle systems.

The support for the node materials that can be used to set particle shading parameters with sequences of basic blocks (including the Particle Info node). This function is available for the particle systems with the Billboard rendering type.

The shader for particles with the Halo type rendering has been rewritten. The support for the Rings, Lines and Star Tips parameters has been added. Maximum particle size limit (caused by the hardware limitations on some platforms) has been removed.

Maximum number of gradient control points limit in the Ramp procedural texture (used for coloring particles) has been removed.

- Extended node material support.

Two new nodes, Vector Curves and RGB Curves, have been added.

The ColorRamp node support has been added. For now, this node supports Linear and Constant type interpolation.

The Particle Info node support has been added. For now, it is fully supported by the Emitter type particle systems with the Billboard rendering type.

- 项目管理器改进。

- An option to export multiple projects into one archive.

This function makes updating the SDK and migrating projects from one workstation to another significantly easier.

- New project deploying options.

Project deployment is required for uploading finished project to the server, sending them by mail and so on. Starting with the current release, project deployment can be performed directly from the

Project Manager's graphical interface. During deployment, projects are packed into a zip archive.

Also, starting with the current release, deployment is available for any type of project (including External type).

- Improvements in the Web Player HTML and Web Player JSON type projects.

Now, during the creation of these projects, you can set the Web Player application parameters such as FPS counter, automatic camera rotation, turning off social network buttons and so on.

Also, projects of these types can now be created with a bundle option which means that all application resources will be located in the same directory.

- Graphical interface improvements.

An option to add and show application icons has been added to make navigation easier and to give users a quick preview of an application in development.

An option to view project info. By clicking the [info] link located at the right side of the project name, a list of detailed information regarding the project can be accessed.

For convenience, the type of a project is now indicated by the prefix at the right side of the link: player: for Web Player HTML or Web Player JSON type projects, dev: for projects under development and build: for compiled (obfuscated) version of an application.

The elements of the Project Manager interface now have pop-up tips.

- The possibility to animate environment settings.

The possibility to animate environment parameters located in the Sky (Horizon Color, Zenith Color), Environment Lighting (Energy), and Mist (Minimum, Start, Depth, Height, Fog Color) tabs has been added. These functions are also available for the NLA animation.

The Animation tab has been added to Blender's World panel. It has Apply Default Animation and Behavior parameters.

For all exported environments, WORLD type meta-object are added to

the scenes. These objects can be used to control the animation of the environment settings. You can access these objects the same way you can access any standard object in the scene.

- 相机改进。

An option to set the limits for vertical movement of camera's pivot point has been added. This function is also available via the API by using the `target_set_pivot_limits()` and `target_get_pivot_limits()` methods.

The new `static_setup()`, `eye_setup()`, `target_setup()`, `hover_setup()` and `hover_setup_rel()` methods have been added for changing and complete setup of the camera behavior. At the same time, the `set_move_style()` has been declared deprecated.

The `target_switch_panning()` method has been added for controlling camera panning, and the `get_view_vector()` method has been added for retrieving the camera's line of sight vector.

The examples of use the camera API are now described in the [corresponding chapter of the documentation](#). This chapter can also be accessed from the `camera.js` API module documentation page.

- Node Logic Editor improvements.

The Content-Type option has been added to the Send Request node. It can be used to reassign the title field of an HTTP request.

- A new sensor has been added to the controls module.

Gyro Quat (`create_gyro_quat_sensor` method) is the sensor for working with gyroscopes on mobile devices. It can be used to handle device rotation quaternion.

- Experimental UC Browser support has been added.
- User Manual has been reworked and expanded significantly.

变更

- The Generate Shadows light source parameter has been renamed to Shadow and now also enables shadow rendering in the Blender Viewport.

- Design of the SDK main page has been changed.

Project list is now located in the Project Manager. Links to run frequently used applications, Viewer and Code Snippets, have been added.

- The `get_coords_x()` and `get_coords_y()` can now receive the `target_touches` parameter.

In case of multitouch, this parameter can be used to use only the touches that are inside of the current target element (the `targetTouches` parameter).

- `camera` API module changes.

The `has_vertical_limits()` and `has_horizontal_limits()` methods have been declared deprecated. The `has_vertical_rot_limits()` and `has_horizontal_rot_limits()` are recommended to use instead of them.

- Changes in the naming rules for the files of the projects under development.

Now, the HTML files of the projects under development (located in the `apps_dev` directory) do not include the `_dev` suffix in their names. To distinguish these projects from the compiled ones, prefixes has been added to the Project Manager.

- When you open the SDK main page, presence of the local development server is checked. If it starts from the local file system, an appropriate warning is shown.

修复

- Incorrect behavior of the `canvas_resolution_factor` engine parameter on Apple iOS devices has been fixed.
- Engine error that occurred in case of absence of the selected object in the Show Object and Hide Object logic nodes has been fixed.
- The incorrect coordinate rounding along the Y- and Z-axis while using variables as parameters in the Transform Object logic node has been fixed.
- Node logic blocking in case some nodes are not linked with the Entry Point node has been fixed.
- The issue with the rendering of dynamic objects the RTT-scenes has been fixed.

- Fullscreen switching issue in the Safari browser has been fixed.
- The issue with the Add-on interface (caused by Emitter type particle systems without a material assigned to the object) has been fixed.
- The Factor output of the B4W_GLOW_OUTPUT node in the materials with the Alpha Clip type transparency now works correctly.
- Dupli groups with None duplication type are not exported and not rendered.
- The `append_ray_test()` method now works correctly if its first parameter is an empty object.
- Several water settings has been optimized and fixed.
- Physical objects that have parent object will now correctly update their position in the case the physics is disabled in the scene.
- The behavior of the Camera Data and B4W Vector View nodes for reflected objects has been fixed.
- NLA animation of the RGB node in materials has been fixed.
- Now, HAIR type particle system will not be rendered if the emitter object has the Do Not Render parameter enabled.
- The work of the `auto_rotate()` method for EYE type cameras has been fixed.
- The export of Hair type particles from non-active scenes in Blender has been fixed.
- Runtime libraries for Windows have been added to fix the resource conversion error.
- Shader generation error that occurred in case there were more than 10 textures in the material has been fixed.

Known Issues

Starting with this release, the list of all known problems and possible solutions is located in the [dedicated chapter](#).

v16.01

新功能

- Line rendering.

Procedurally generated line rendering is now supported. Special type of object, activated by the Line Renderer option in blender, is provided for it, as well as several API functions: `draw_line`, `get_line_params` and `set_line_params`.
- New logic editor nodes.
 - The Transform Object node can be used to move an object in world, parent or local coordinate space.
 - The String Operation node can be used to perform operations with string constants and variables, like the Math Operation node.
- Simplified SDK installation.

Now, to [install SDK](#), you just need to specify the path to SDK in the File->Scripts section of the User Preferences panel.
- WebPlayer application improvements.

An option to turn off social network buttons located in the bottom-right corner of a loaded scene. To do this, you need to specify the `no_social` attribute before the application starts.

Escape characters in the Web Player's address bar are now processed correctly.
- Rendering quality improvements for Head-Mounted Displays (HMD).

Means to correct distortion and disable chromatic aberration while using Head-Mounted Displays have been added. Use `set_hmd_params()` function to set up these parameters.
- Two new sensors have been added to the `controls` module.

Touch Rotate (`create_touch_rotate_sensor()` method) sensor can be used to process rotation using two fingers on touch devices.

Callback (`create_callback_sensor()` method) is a sensor whose value is defined by a callback function called every frame.

- Resource converter utility improvements.

The `-verbose` and `-jobs` parameters have been added to the `resource converter utility` `converter.py`, which can be used to output detailed information and to set maximum number of the parallel processes during conversion, respectively.

- Several API methods have been added.

`is_line()` checks whether the object has LINE type.

`is_idle()` checks whether the scene loader has finished all planned tasks.

`hover_switch_horiz_rotation()` can be used to enable and disable horizontal rotation of HOVER type cameras.

- Node tree refresh algorithm has been optimized.

变更

- Using cubic reflections has been simplified.

Cubic reflection is now rendered from the object's geometric center and not from its Blender origin point. Also, flat reflection is no longer rendered when rendering cubic reflection, which caused artifacts before.

- Changes in API.

The following methods have been added to the `camera` module: `get_vertical_axis()`, `set_vertical_axis()`. “EYE” type camera rotation to a “phi” angle is now performed along the vector returned by the `get_vertical_axis()` function.

The `correct_up()` method now accepts the `strict` flag as a parameter. This parameter can be used to align camera codirectionally with the `y_axis` vector and not simply parallel to it.

`eye_get_vertical_limits`, `eye_get_horizontal_limits`, `target_get_vertical_limits` and `target_get_horizontal_limits` methods

can now return limits set in both world and local coordinate spaces. This can be defined by the local parameter.

- Logic node changes.

The nodes have been separated into categories.

- An option to select between Number and String variable types has been added to the Variable Store node.
 - An option to use string variables to store the entire body of the server request and response has been added to the Send Request node.
- Deprecated Mass Reexporter tool has been removed.

The Mass Reexporter tool panel has been removed, as automatic scene reexport function is already present in the Project Manager (re-export scenes operation).

- Deprecated functionality.

The following methods: 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() have been removed.

The `set_camera_move_style()` method and `SPACE_WORLD` and `SPACE_LOCAL` constants have been declared deprecated.

The `has_vertical_limits()` and `has_horizontal_limits()` methods have also been declared deprecated. The new methods `has_vertical_rot_limits()`, `has_horizontal_rot_limits()`, `has_vertical_trans_limits()` and `has_horizontal_trans_limits()` have

been added to replace them.

修复

- Positioning and flickering of the particle systems with world coordinates have been fixed.
- Front Facing for cubic reflections has been fixed.
- Stereo mode rendering errors have been fixed.
- Incorrect positioning in the child object animation has been fixed.
- Scene loading freeze in the Safari browser while using .ogg audio files with Background Music type speakers has been fixed.
- The Move Camera node bug, which disabled Duration parameter after .blend file reopening, has been fixed.
- System crash while exporting particle system with complex mesh emitter has been fixed.
- The Switch Select node error that led to incorrect switches has been fixed.
- Several camera limit rendering issues in the Blender viewport have been fixed.
- The issue with exporting materials attached directly to the object and not to the mesh has been fixed.
- The RenderCallback function set by the `set_render_callback()` method has been fixed.

The RenderCallback function are now called right before rendering the current frame, so the scene and objects are up-to-date.

- Issues with rendering shadows from multiple light sources of different types (such as POINT and SUN) have been fixed.
- Environment texture rendering while using the texture as the world map and in a stock material at the same time has been corrected.

Known Issues

- Problems with updating of the add-on.

It's strongly advised to restart Blender after installing a newer version of Addon/SDK.

- NVIDIA 331 driver in Linux can cause WebGL errors.
- Changed texture filtering on some platforms.

An incorrect texture filtering was disabled on iPad and Internet Explorer for materials with Alpha Clip type of transparency.

- Some devices with Mail GPU require manual WebGL activation in browser settings.
- For the local development server to work on Apple macOS and Blender 2.76, you may need to install [Python 3.4](#). This is due to a bug in Blender <https://developer.blender.org/T46623>. This bug has been fixed in Blender 2.76b, so updating it is advised.
- Skeletal animation can work incorrectly while using Nouveau drivers.

v15.12

新功能

- Support for shadows from multiple sources.

Support for shadows from multiple sources has been added. This feature can be used to greatly improve the realism of scenes lit by multiple light sources. Its functions and limitations are described in the [documentation](#).

- Experimental support of the HMD (Head-mounted display).

Experimental support for the head-mounted displays with the WebVR API has been added. For now, using this technology requires a web browser with WebVR support and an Oculus Rift device. More details in the [documentation](#).

- Aligning objects with the camera in the viewport.

The new Viewport Alignment option can be used to attach objects to the camera in the viewport. This feature can be used to create UI elements attached to the center, edge or corner of the screen. Unlike parent-child alignment, the position of an aligned object will automatically change when screen resolution or aspect ratio is changed.

Functionality of the `append_stiff_viewport()` method used for Stiff Viewport constraint has also been expanded.

- An option to show camera limits in Blender viewport.

To make the task of setting camera limits up simpler, the Display limits in viewport option has been added to the Data panel of the camera object. When enabled, it will show the limits right in the Blender viewport. Default settings of the limits has also been changed.

- Anaglyph rendering improvements.

New anaglyph rendering algorithm has been implemented. It has better color rendering. Also, plane of convergence of the left and right image for a TARGET type camera will now be calculated automatically based on a point around which the camera is rotated.

- New logic editor nodes.
 - Move To: Can be used to move an object to another object.
 - Console Print: Can be used to print various information (including variables values) to the web browser console.
- A Stereo View button has been added to the Web Player.

Can be used to switch to the HMD mode (if the web browser supports this technology) or to anaglyph mode.

- New methods for working with objects.

The new methods `get_selectable_objects` and `get_outlining_objects` have been added to make working with objects simpler.

- Dynamic objects can now be deleted.

Before, only objects copied by the user could have been deleted.

- The option to choose an annotation of an object based on the canvas coordinates has been added.

The `pick_object` function now returns the object's annotation, if the object is present in the sent coordinates.

- A new method to initialize media resources for mobile devices has been added.

Before, forced canvas block was used as a workaround of the mobile browsers restriction of loading media resources (video and audio), and a user had to click a popup icon to start the application. Now, you can manually initialize media resources by using `activate_media` method. This function can be useful if the application already have elements that require user input, such as Start button, sound mute button and so on.

- New sensor manifold type - `CT_POSITIVE`

The new `CT_POSITIVE` sensor manifold type activates event handler if the logic function result isn't zero.

变更

- The Fast Preview button has been duplicated in the UI low panel.

This button can be used to preview the scene without switching to the Development Server scene settings panel.

- Canvas texture API has been changed.

Now working with a canvas texture is simpler. New methods `get_canvas_ctx` and `update_canvas_ctx` have been added. The `get_canvas_texture_context` and `update_canvas_texture_context` methods have been declared deprecated and are not recommended to use. The Source ID interface field for the canvas texture has also been removed.

- Wrong behavior of the shadows from Alpha Clip materials has been fixed.

Incorrect behavior of the shadows casted by the objects with Alpha Clip materials without transparency value has been fixed.

修复

- Incorrect physics of a copied object has been fixed.

Incorrect physics behavior of a copied object (which occurred if the object was moved before being added to the scene) has been fixed.

- Incorrect fog behavior if a water plane was added to the scene has been fixed.
- Misplacing of the Blender interface panels in basic scene for a new project has been fixed.
- Cubemap rendering issue on the NVIDIA GeForce 200 series GPU has been fixed.
- Engine workflow on iPhone (4, 4S and 5), iPad (2nd, 3rd and 4th generations) and iPad Mini (1st and 2nd generations) has been improved.
- Incorrect rendering of the shadows casted by billboard objects has been fixed.

Known Issues

- Problems with updating of the add-on.

It's strongly advised to restart Blender after installing a newer version of Addon/SDK.

- NVIDIA 331 driver in Linux can cause WebGL errors.
- Changed texture filtering on some platforms.

An incorrect texture filtering was disabled on iPad and Internet Explorer for materials with Alpha Clip type of transparency.

- Incorrect Depth texture behavior on Windows OS.

Depth textures are unstable on Google Chrome 32 bit on Windows. This problem is fixed in the beta version of the browser.

- Some devices with Mali GPU require manual WebGL activation in browser settings.
- For the local development server to work on Apple macOS and Blender 2.76, you may need to install Python 3.4. This is due to a bug in Blender

<https://developer.blender.org/T46623>. This bug has been fixed in Blender 2.76b, so updating it is advised.

v15.11

新功能

- Initial support of WebGL 2.0 technology.

New API features are automatically provided with compatible browser and hardware

- New Logic Editor Features:

- Camera animation.

The option to smoothly change camera's position has been added to the Move Camera node.

- New: behavior animation.

The Play Animation node can now add Behavior animation to an object.

- New Stop Animation node

Can be used to stop object's animation

- New Stop Sound node

Can be used to stop a speaker's sound playback

- New Stop Timeline node

Can be used to stop NLA animation

- MSAA support

This antialiasing algorithm is used in systems compatible with WebGL 2.0. For now, it allows the highest image quality on the ULTRA settings by using 16x MSAA algorithm.

- Viewer interface improvements.

A Home button has been added. When pushed, it will open the basic scene specified in the application's URL parameters, or, if there isn't any, a placeholder scene with Blend4Web cubic logo will be opened.

The Reset button has been moved to the right side of the application's upper panel and will now reset not only internal but also URL parameters (such as loaded basic scene) of the Viewer.

The Auto View Mode button used for automatic walkthrough of all scenes has been moved to the Tools & Debug panel.

- High precision rendering performance measure mechanic.

Using the WebGL extension EXT_disjoint_timer_query allows high precision timers to be created, which can be used to measure the performance of GPU operations. With the activation of the debug mode in the Viewer ("HUD Info" option in the Tools & Debug panel), the time (in milliseconds) the GPU needs to render part of the scene (subscene) is shown in the last column of the debug information.

- Web player improvements.

Now, the cursor changes appearance while the camera is moving.

- There are numerous additions and improvements in the documentation.

变更

- The Copy Cursor Location button has been renamed to Look At Cursor, and its behavior has also been changed.

Now, by pressing this button, you can rotate the camera to the pointed direction.

- Changes API.

For compatibility with third-party applications, `:b4wref:transform.set_matrix`, `get_matrix`, `set_matrix_rel` and `get_matrix_rel` methods have been added, but for optimization purposes, using `set_tsr`, `get_tsr`, `set_tsr_rel` and `get_tsr_rel` methods is recommended.

Now the `hide_object` and `show_object` functions will, by default, change the visibility of an object and all its children. To change only the current object's visibility, ignore `_children` parameter with true value should be sent.

The following method has been declared deprecated and will be removed in further releases: `create_sep()` (`tsr` module). `tsr` module's `set_sep()` method should be used instead of it.

camera module API has been changed considerably. Deprecated methods `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()` have been removed. The following methods have also been declared deprecated: `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`. In their place, we recommend using methods starting with camera type prefix: `target_...`, `eye_...`, `static_...`, `hover_....`. Methods `set_velocity_params` and `get_velocity_params` have also been declared deprecated, and instead of them, using `set_velocities` and `b4wref:camera.get_velocities`, respectively, is recommended

For working with the distance between the current TARGET or HOVER camera and the pivot point, `target_get_distance`, `target_set_distance` and `hover_get_distance` methods have been added

- In the Viewer, Play All and Stop All buttons have been added for playing and stopping all animations, respectively.
- Logic editor changes:
 - Select node and Select & Play Animation and Select & Play Timeline nodes based on it, have been declared deprecated. As a replacement, using the combination of Switch Select, Play Animation and Play Timeline nodes is

recommended.

- SDK examples, which were using aforementioned deprecated nodes, have been updated.
- For the logic nodes behavior transparency, Cyclic NLA option has been switched off.
- Behavior of the HOVER camera in the absence of limits has been changed.
 - HOVER camera will now be hanging over pivot point at a fixed distance and fixed angle.
- Glow effect changes.

Light sources will now influence glow effect, if Material or Extended Material nodes are present.

修复

- Project Manager errors were fixed.

Links to the developer's applications included in the project are now showing in the list.

Formatting in the compiled applications' HTML files has been fixed.

- Skinning on the mobile platforms has been improved.
- Addon's translation files connection error has been fixed.

An error that could have happened on some devices because of system's inability to process addon's translation files, has been fixed.

- Particle objects shadow casting improvements.

Fix behavior of the shadows casted by "Hair" particles (used as billboards) while changing the size of the main canvas.

- Engine workflow on the Mali 400 series GPU has been fixed.
- Flat reflections and fog for double_sided_lighting materials have been fixed.
- Local Storage limited quota on the Safari browser in the incognito mode no longer causes error.

- Functionality of the Render Glow Over Transparent Objects' option in the Blender's glow settings has been restored.
- Several video texture playback issues have been fixed.

Known Issues

- Problems with updating of the add-on.

It's strongly advised to restart Blender after installing a newer version of Addon/SDK.

- NVIDIA 331 driver in Linux can cause WebGL errors.
- Changed texture filtering on some platforms.

An incorrect texture filtering was disabled on iPad and Internet Explorer for materials with Alpha Clip type of transparency.

- Incorrect Depth texture behavior on Windows OS.

Depth textures are unstable on Google Chrome 32 bit on Windows. This problem is fixed in the beta version of the browser.

- Some devices with Mail GPU require manual WebGL activation in browser settings.
- You may require to install Python 3.4 on the systems with Apple macOS and Blender 2.76. This issue is connected with Blender bug <https://developer.blender.org/T46623>.

v15.10

新功能

- New Project Manager features.

It is now possible to export/import projects. This simplifies updating projects to newer versions of the SDK and also allows users to run and debug projects on various platforms. You can also share your projects with other developers in an efficient way.

You can now create WebPlayer JSON and WebPlayer HTML projects. Such options allow you to create projects specifically for WebPlayer app which do not require writing any additional code.

To simplify navigation between your projects, two features have been added. The first one is the possibility to sort projects by name, and the second one is the possibility to show/hide SDK's stock projects.

It is now possible to remove projects from the SDK.

Improved Windows support. We now provide a Windows version of Java with the SDK, so you are no longer required to install any additional dependencies in order to build projects.

- Anti-aliasing improvements.

A new option AA Quality has been added to the Render > Anti-Aliasing panel in Blender. This option allows you to select quality level of anti-aliasing. Also, the anti-aliasing algorithm is now based on FXAA 3.11, which increases both quality and performance especially for HIGH and ULTRA quality profiles.

- Node Logic Editor improvements.

- Move Camera node.

Allows you to specify the camera's location and target.

- Play Sound node.

Allows you to play back sound from a selected speaker.

- Switch Select node.

This node works the same way as Select but allows you to select multiple objects in more flexible and convenient way.

- Support for random numbers generation has been added to the Math Operation node.
 - Support for POST requests in the Send Request node.

- Repeat mapping type for non-power-of-two textures is now supported.

Repeat mapping type is now supported for non-power-of-two textures

(i.e. textures whose dimensions are not 256, 512, 1024, etc). Also the mipmapping technique (trilinear filtering) is also supported for such textures.

- Automatic rescaling of textures is now performed if their dimensions exceed platform limits.

In the cases when texture dimensions exceed platform limits, textures will be automatically downscaled. The only exception is compressed textures.

- New button Fast Preview to perform fast scene previews.

The button is located on the Render > Development Server panel.

- Support for Intensity and Color animation of lamp objects.

It is now possible to animate intensity and color of lamp objects, both when using conventional and NLA animation.

变更

- Project Manager interface has been improved.

Improved UI, added Development Server > Project Manager button to run the Project Manager in the default browser.

- Speakers functioning has been improved.

The `is_play()` method now correctly notifies about finishing sound playback, with a minimal delay.

- Some new export warnings have been added.

Upon exporting some objects, their type will be changed to EMPTY in case of empty geometry or in the case when the sound file for the SPEAKER object does not exist. Messages on such facts are now displayed in the browser console.

- Displaying the object selector for logic nodes has been improved.
- Gamma correction behavior in node materials has been changed.

Gamma correction in node materials is now performed differently because of changes in Blender 2.76.

- Changes API.

The `resize` method is now deprecated and will be deleted from the `main` module. The `resize` method from the `container` module should be used instead.

修复

- Project Manager errors were fixed.
- Stability on mobile devices has been improved for scenes with too many lamps.
- Stability on Unix systems has been increased.

Stability on Unix-systems using AMD GPUs and open source drivers has been increased.

- An error related to annotations being added to a scene has been fixed.

Fixed an error which occurred when annotations without the Object -> Meta Tags property were added to the scene.

- Fixed an error with incorrect audio playback during browser tabs switch.
- Several video texture errors have been fixed.
- Fixed an error occurred when an empty node group was used inside a node material.
- Fixed an error with a particle system which use an object with LOD as a particle.
- The Delay node error when using a variable as a parameter has been fixed in the logic editor.
- Fixed an error in the logic editor which occurred upon deleting an Entry Point node.
- Fixed an error in the logic editor with duplicated variables in the dropdown list.
- Fixed an error in scenes which use both the Play Animation logic node and the switched off NLA flag.
- The behavior of the Math->Power and Gamma shader nodes has been fixed and is now consistent with Blender.

Known Issues

- Problems with updating of the add-on.

It's strongly advised to restart Blender after installing a newer version of Addon/SDK.

- NVIDIA 331 driver in Linux can cause WebGL errors.
- Changed texture filtering on some platforms.

An incorrect texture filtering was disabled on iPad and Internet Explorer for materials with Alpha Clip type of transparency.

- Incorrect Depth texture behavior on Windows OS.

Depth textures are unstable on Google Chrome 32 bit on Windows. This problem is fixed in the beta version of the browser.

- Some devices with Mail GPU require manual WebGL activation in browser settings.
- You may require to install Python 3.4 on the systems with Apple macOS and Blender 2.76. This issue is connected with Blender bug <https://developer.blender.org/T46623>.

v15.09

新功能

- Project Management.

A new project management system allows one to:

- show the list and info about existing projects
- execute apps, load scenes in the Viewer app, load scene files in Blender.
- create and configure new applications, optionally using prefabricated app/scene starter files to simplify creating new applications
- build applications, create versions suitable for deployment on a server

- convert application resources (textures, sounds and video) to alternative formats
- automatically reexport all application scenes including json and html files

There are two methods to manage projects: by using project.py utility which has a simple self-documented command line interface, or using a GUI web application, which is run on the developer server. This server does not require any additional configuration and as before is executed by the Open SDK button in Blender.

The project management system works on all operating systems, however, some operations may require additional dependencies. To find out more about the configuration of this system check out the following [topic](#) in documentation.

- Controlling armature bones.

Enhanced features to control armature objects.

Added a new `armature` module which includes methods to get or assign bone positions both in armature coordinate space (`get_bone_tsr()`, `set_bone_tsr()`) and in bone-relative coordinate space (`get_bone_tsr_rel()`, `set_bone_tsr_rel()`). By using these methods, it's possible to program armature behavior. For example, animate interactive characters or create sophisticated armature-based objects with multiple moving parts.

Support for `COPY_TRANSFORMS` constraints on bones. This allows bones to follow movements of any objects located on the scene, e.g create physically simulated “Ragdoll” objects.

- New logic editor nodes.
 - Play Animation: play animation of an object.
 - Send Request: send HTTP GET request to a server and parse its response.
 - Inherit Material: copy material properties from one object to another.
 - Set Shader Node Param: change outputs of Value and RGB shader nodes.
 - Delay: delay program execution for a given amount of time.

- Apply Shape Key: change shape key value for an object.
- Outline: control object outlining effect.
- New Allow NLA settings for video textures, which enables/disables NLA-animation for a given texture.
- New features in the material module.

New APIs added `set_specular_color_factor` and `get_specular_color_factor`.

- New features on the SDK index web page.

Index page of the SDK now contains a Tools section, which in turn includes Project Manager and WebGL Report tools.

- Web player improvements.

A new optional alpha setting has been added, which is used to set rendering canvas transparency. The [Outline on Select](#) feature is now supported.

- New app compilation type update has been added to project.py utility.

This type of app compilation allows one to update the engine inside a project directory. The app itself is not compiled with this option. The feature is useful for basic applications and tutorials.

- New features in the app module.

A new param `disable_zoom` has been added to the `enable_camera_controls()` method. This parameter disables zoom movements of the camera.

A new method `queue_animate()` has been added to the `app` module.

- New features in the scenes module.

A new method `has_picking_subs()` has been added to the `scenes` module.

- New features in the debug module.

A new method `fake_load()` has been added to the `debug` module.

变更

- Add-ons (such as `app`, `mouse` and others) are now a part of the compiled engine version:
 - `b4w.min.js` - advanced optimization (`b4w.full.min.js` previously)
 - `b4w.simple.min.js` - simple optimization
 - `b4w.whitespace.min.js` - optimization of whitespaces used in the code

The correct version is chosen according to the application compilation settings.

- Node Logic Editor improvements.
 - Play -> Play Timeline; Select & Jump -> Select, Register Store -> Variable Store nodes were renamed.
 - Now it is possible to create user-defined variables apart from register-variables.
- The rarely used combine engine building method was removed.
- Documentation for the `resource converter` was revised.
- Documentation for the `addon translator` was revised
- The export of UV-layers and vertex colors was changed.

Now the behavior of UV-layers and vertex colors resembles that in Blender even more.

- Support for NLA-animation and video-textures was extended. They act similar to those in Blender.
- A message about the lack of animation channels was added to the `export` errors.
- A non-critical export error on the selection of unsupported Render Type in particle systems was added.

- The Specular Color Factor property is now being inherited during material inheritance (`inherit_material` API method).
- Changes API.

The following methods are marked as deprecated and will be removed in future releases: `is_camera`, `is_mesh`, `is_armature`.

The following methods of the `objects` module should be used instead: `is_camera`, `is_mesh`, `is_armature`

Also, there are the following new methods: `is_speaker`, `is_lamp` and `is_empty`.

The `get_object_by_dupli_name_list` method now returns null, if the `name_list` parameter is given incorrectly. The `get_object_name_hierarchy` method for receiving the full list of names with respect to object duplication was added. The value returned by this method is a valid input for the `get_object_by_dupli_name_list` function.

The `wireframe_mode` parameter in the `set_debug_params` method now has a value of one of the following constants: `WM_NONE`, `WM_OPAQUE_WIREFRAME`, `WM_TRANSPARENT_WIREFRAME`, `WM_FRONT_BACK_VIEW`, `WM_DEBUG_SPHERES`.

- NLA Animation Behavior for dynamically loaded scenes changes.

If there are objects with NLA-animation in such scenes, they are now influenced by the NLA settings of the main scene. Previously, such animation was not supported.

- The refactoring of the `objects`' internal structure was continued.
- Export error was added.

Now when an object with a type other than “Mesh” is used as a “Hair” particle, a non-critical `export` error will occur.

- Licensing information was added to the distribution sources.
- Now texture slots with Environment Maps containing video textures are not exported.

A [non-critical export error](#), stating that a video cannot be used as an Environment Map, was added.

修复

- Render Above All option now works correctly with node materials.
- Fixed `remove_object()` function.
- “Hair” particles have become more stable.

Fixed the bug that appeared while using an object with the “Hair” particle modifier.

- Improved Windows Phone support.
- Fixed the bug that appeared while copying physical objects.
- Fixed the bug with Orco vector output when an object has zero scale in one or several axes.
- Fixed the bug in particle emitters: it appeared when an object with physics settings was chosen as a particle.
- Fixed the bug in NLA animation: it could not start from a frame other than 0 before.
- Fixed Lamp Data behavior: previously the information about light sources was not always refreshed during scene loading.
- Fixed the bug in calculations of Normal vector output in node Geometry on the back side of a polygon.
- Fixed the bug of Orco vector output in node Geometry that appeared if object was translated relatively to its origin in Blender.
- Fixed calculation of the last frame of NLA animation for video textures.
- Fixed the engine bug related to different setting combinations of objects’ selection and outlining.
- Wind bending effect has become more stable.
- Fixed bugs in Alpha Clip materials rendering.

- Fixed specular texture reproduction during material inheritance.
- Fixed the bug with light sources in apps with multiple scenes.

Known Issues

- Problems with updating of the add-on.

It's strongly advised to restart Blender after installing a newer version of Addon/SDK.

- NVIDIA 331 driver in Linux can cause WebGL errors.
- Changed texture filtering on some platforms.

An incorrect texture filtering was disabled on iPad and Internet Explorer for materials with Alpha Clip type of transparency.

- Some devices with Mail GPU require manual WebGL activation in browser settings.

v15.08

新功能

- Node-based logic editor.

This new logic editor allows to add interactivity to your apps more easily by using a tree of connected nodes. The NLA Script tool which was previously used for this purpose has been removed, the old scripts are automatically converted to the newer format during blend file loading.

- Support for various shading models inside MATERIAL and MATERIAL_EXT nodes.

Now the diffuse and specular shading models are selected based on materials inside such nodes. This is different from the previous behavior when the shading was the same for all nodes and was assigned by node material itself. This feature allows mixing different basic materials (non-textured) inside node-based ones.

- Improvements in transformation API.

New `transform` methods to perform relative objects transformations:

`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()`.

Methods without `_rel` suffix now always perform transformations in world coordinate space, even if they are children of other objects.

Also new methods to perform local rotations `rotate_x_local()`, `rotate_y_local()`, `rotate_z_local()` have been added to `transform` module.

- Improved support for OS Windows.

Support for OS Windows in `resource converter`. Now Windows users can create cross-browser applications which use media resources in different formats. Also the `application builder` app can also be run in Windows.

- Support for the Microsoft Edge browser.
- Support for physics simulation in the main (non-worker) browser process.

This feature is useful for eliminating delays in physics calculations in some mobile browsers. Earlier, all simulations took place in separate Worker threads. The feature is controlled by `physics_use_worker` parameter of the engine's initialization.

- Support for bounding box recalculation.

A new method `update_boundings()` has been added to the `objects` module. This method performs recalculations of the object's boundings such as box, sphere, ellipsoid, cylinder, cone and capsule.

- Documentation improvements.

A new theme has been selected for the user manual. This theme improves documentation readability on displays with different screen resolutions.

Greatly improved and extended documentation for `application developers`.

- New APIs in `camera_anim` module.

New methods `stop_cam_moving()` and `stop_cam_rotating()` in the `b4wmod:camera_anim` module.

- Binary compatibility checks.

Now the version checks between .bin and .json files and the current engine version are performed while scenes load.

变更

- Changes in node material editor.

Custom shader nodes have been moved from the Group menu to the Blend4Web menu. Also, an error with duplicated menu entries has been fixed.

- Refactoring of object structure.

Complete refactoring of an object structure has been initiated. Object structures now have strong typing and include less redundant data. This improves overall engine performance.

- Changes API.

The following methods are now deprecated and will be removed in future engine releases: `get_object_dg_parent()` (`scenes` module), `get_parent()` (`constraints` module). Instead, it is recommended to use `get_parent()` and `get_dg_parent()` from the `objects` module.

- Improved interaction between the development server and multiple Blender instances.

修复

- Fixed a compilation error in the `project.py` utility when a project is compiled to some external directory.
- Fixed a web server error which could arise due corrupted Windows registry.
- Fixed an export freeze for some corrupted .blend files.
- Fixed incorrect non-uniform scale warnings upon exporting Metaball objects.

- Fixed an error with Auto View mode in the Viewer app if were an empty scene category.
- Fixed rendering of dynamically loaded materials if they have the same name.
- Fixed an export error for scenes using Copy Transforms constraints.
- Fixed an error with rendering depth textures in the Microsoft Edge browser.
- Fixed a bug with touch events in the Microsoft Edge browser.

Known Issues

- In the logic editor, some of looped links are highlighted in red. This issue has only cosmetic effect and can be safely ignored.

v15.07

新功能

- Support for Alpha Clip transparency for node materials.

It is now possible to specify a transparency mask for Alpha Clip materials using node logic.

- Support for soft particles.

The new property Soft Particles has been added to the Render panel of the Emitter particle system. When activated, this property renders smooth edges near opaque objects located on the scene. This effect is available only for materials with Alpha Sort, Alpha Blend or Add type of transparency.

- New preprocessor for shader instructions.

This new preprocessor has more straightforward architecture and works faster, which, in turn, greatly reduces overall scene loading time, especially in cases with many different shaders.

- Improved Viewer app performance.

Performance problems caused by event handling in jQuery Mobile library have been fixed.

- Zero level property for HOVER cameras.

This property represents a Z coordinate of the reference plane in which the camera's pivot point is located.

- New sensor manifold type: CT_CHANGE.

Along with CT_CONTINUOUS, CT_TRIGGER, CT_SHOT and CT_LEVEL manifold types the new CT_CHANGE type can now be used. This type allows to execute a manifold's callback right after any of the sensors' values in the manifold has been changed.

- New APIs in camera_anim module.

New API methods such as `move_camera_to_point()`, `rotate_camera()`, `is_moving()` and `is_rotating()` have been added to module `camera_anim`.

- A new option to assign materials to objects.

Until now it has not been possible to assign materials on objects using Object property. Now, this feature is supported.

- Support for external requests to the local development server.

This feature is enabled by the Enable external requests property in addon preferences.

- New method `clear_animation()` in the `time` module.
- New camera APIs.

APIs of camera module has been extended by `get_vertical_limits()` and `has_vertical_limits()` methods which are used to get or check the camera's vertical rotation and translation limits.

- Documentation improvements.

New user manual design optimized for devices of all kind.

The differences between coordinate systems used in Blender and Blend4Web are now specified in documentation.

- Support for addon i18n.

This feature allows translation of addon interface to different languages.
The translation into Russian is almost complete.

变更

- Various usability improvements in the addon interface.
- New `force_container_ratio` option in the `init()` method.
- Simplified usage of water caustics.

Water caustics effect is now activated by the Caustics option located on the material's Water panel.

- New way of evaluating mouse movement sensors.

Now all elements inside the Canvas Container element (e.g Anchors) do not lock mouse events, thus allowing us to prevent glitches during camera movement.

- Option to change quality of reflections.

Now it is possible to choose one of three predefined reflection quality modes: LOW, MEDIUM or HIGH.

- New properties `--project` and `--app` in the `project.py` utility.
- New animation APIs.

New methods to simplify frame setting:
`set_first_frame()`/`set_last_frame()`.

- New behavior of plane reflections.

It is now possible to omit specifying the reflection normal by leaving the Reflection Plane option empty. In this case, local Z axis will be used instead.

- New SSAO behavior.

Now there is no need to have shadow casters in the scene to use the SSAO effect.

- New texture anisotropic filtering settings.

There is a new Default value for anisotropic filtering option assigned to textures. By selecting this value you can use the anisotropic filtering specified on the scene. If you need to disable anisotropic filtering on all textures, again, use the setting from the scene.

- New specular alpha shading behavior to match Blender's.
- The physics mask/group, assigned for collision materials, has been extended from 8 to 16 bits.
- New asynchronous shader loader for developer version of Blend4Web.
- Elimination of shaders module.

The methods of this module is now available from the `debug` module.

- New script for batch processing exported json/html files.

The scripts `reexporter.py` and `resaver.py` have been combined into `process_blend.py`, which has options for processing exported json/html files.

- Minor refactoring and improvements in API documentation.

修复

- Fixed a bug with incorrect Canvas element size appeared on the engine's startup.
- Fixed a lighting bug on objects with the negative Scale option.

This condition is properly handled and reported to the user (in the browser console).

- Fixed a bug with object's vertex animation.
- Fixed a bug with animation played in reverse.
- Fixed an error with incorrect state of Collision and Ray sensors which appeared after deleting physics objects.
- Environment textures with dimensions exceeding the supported ones are now processed correctly.

Now, such textures are not being turned off but rendered in reduced scale instead. Dimensions are reduced synthetically for NVIDIA GeForce 8000

GPUs on Windows Chrome.

- Fixed an issue with incorrectly reported error which appeared while loading scenes with missing camera/world.
- Fixed a Blender startup error with the world missing from the scene.
- Fixed a bug in the Webplayer app when the sound button was missing in some scenes.
- Fixed a bug in the scenes when motion blur is the only effect to appear.
- Fixed the `get_material_extended_params()` method in the `material` module.
- Fixed Firefox Mobile crashes when using shadows. Improved overall stability for this browser.
- Improved reexporter stability.

Known Issues

- Problems with updating of the add-on.

It's strongly advised to restart Blender after installing a newer version of Addon/SDK.

- NVIDIA 331 driver in Linux can cause WebGL errors.
- There is a [bug](#) with video textures on Chrome 43 for Android.

Please update your Chrome browser to Beta or wait until the next Chrome update.

- Fixed issues with the Background Music speakers in the scenes exported to html.

Currently, there is a bug in Google Chrome ([Issue 511251](#)), related to an error with audio file origin. Currently, this issue is partially resolved by our workaround with forced `crossOrigin` attribute on audio sources.

- Changed texture filtering on some platforms.

An incorrect texture filtering was disabled on iPad and Internet Explorer for materials with Alpha Clip type of transparency.

v15.06

新功能

- New add-on user interface.

Add-on interface has been redesigned. It is now activated by the new rendering profile, Blend4Web, which only contains panels and options explicitly supported by the engine. To simplify navigation, the old multi-line Blend4Web panels have been re-grouped into the smaller ones, based on functionality.

Also, there is a new feature to automatically assign graphic effects required for a scene. In particular, shadows, refraction, Glow and Outline effects acquired the new AUTO property which automatically activates them if some objects or materials located on the scene require such effects.

- New normal editor.

The normal editor has been substantially upgraded. Now it is fully compatible with the native Blender datablock used to store normals. This new editor has more efficient UI and also allows to edit split normals.

- Support for new material nodes.

Orco and Local outputs of Geometry node are now supported. There are also some preliminary steps to support RGB Curves, Vector Curves, ColorRamp and Cycles nodes.

- New shading models.

Minnaert/Toon diffuse shaders and Blinn specular shader have been implemented. Thus, starting from this release the engine supports all shading models of Blender.

- Support for Blender 2.75
- Multiple physics improvements.

Code for collision detection has been rewritten. Now it is possible to determine colliding objects, and also the coordinates and the normal at the collision point.

Improved ray casting API. In particular, one can specify an option to perform an automatic cleanup of the ray test object and also another option to cast a ray through multiple objects. As in the case of collision detection, this new API allows to determine the target object and the position/normal of the hit point. There is also a new possibility to cast rays from point to point in global space, without requirement to specify a source object.

Extended possibilities of Collision and Ray sensors.

Support for deleting physics objects and automatic recalculation of collision/ray tests after physics objects have been added/removed.

- A new tool for reexporting multiple scenes.

A new Mass Reexporter tool has been added to addon. This tool allows to automatically reexport all scenes from the specified list of directories.

- Possibility to check for updates.

You can now enable a Check for Updates on Startup option in addon settings to perform automatic checks for the new versions of Blend4Web.

- API to control Motion Blur postprocessing effect.

New methods to control Motion Blur effect `get_mb_params()` and `set_mb_params()` has been added to `scenes.js` module.

- Support for Timeline markers.

To extract frame numbers from timeline markers a new method `marker_frame()` has been implemented in `scenes.js` module.

- New NLA APIs.

A new set of methods: `set_range()`, `reset_range()`, `set_cyclic()` and `clear_callback()` has been added to `nla.js` module. Also, it's now possible to specify callback in `play()` method.

- New API to change Canvas resolution.

To change Canvas resolution it's sufficient to execute method `set()` from `config.js` module with the following parameters: `canvas_resolution_factor` and `value`, where `value` is the new resolution of

Canvas. This feature is particularly useful for creating high-definition screenshots.

- Support for Vertex Groups -> Length option in Hair particle system.
- New API documentation.

To document types used by Blend4Web applications we developed a new formal type system. For example, in our old API docs 3D vectors had `Float32Array` type. Now they have formal `Vec3` type. This solution allows us to formulate more clear and intelligible descriptions for API and, more importantly, helps our users to develop more readable and reliable applications.

- New colors API.

To work with colors in efficient way two new modules: `rgb` and `rgba` have been created. They include APIs to create new color vectors and convert them between different profiles.

- Automatic determination of path to SDK.

Addon option Blend4Web SDK Directory is filled up automatically, if the addon is located in default `blender_scripts` directory inside SDK.

- API for correct calculation of Canvas 2D coordinates.

For proper manipulations with mouse cursor and touchscreen devices the engine requires correct 2D Canvas coordinates.

Details about calculation and use cases of such coordinates are described in the separate [topic](#).

To support this feature the following methods have been added to engine's APIs: `client_to_canvas_coords()`, `set_canvas_offsets()`, `update_canvas_offsets()` and `force_offsets_updating()`. Also, there is a new configuration option: `track_container_position`.

- API to change smooth factors of the camera's movement.

To change smooth factors of the camera's movement the following APIs have been implemented: `set_plock_smooth_factor()` and `get_plock_smooth_factor()` in `mouse.js` module (for

Pointer Lock mode) and also set_camera_smooth_factor() and get_camera_smooth_factor() in app.js addon (for general use cases).

- New favicon picture has been added to Webplayer.

变更

- New API spec for collision detection and ray casting.

Methods append_collision_test() and append_ray_test(), as well as create_collision_sensor() and create_ray_sensor() now have new spec, which is incompatible with the previous versions of Blend4Web engine. All developers should consider upgrading their applications to match this new behavior.

- Rendering to texture changes.

It's now possible to render scene into texture cyclically, e.g. when two scenes render one into another. The main constraint here is the requirement to have at least one scene, into which is nothing is being rendered.

- The option Apply Default Animation is now disabled if an object already have an NLA animation attached.

In cases when the object already have an NLA animation attached, the option Apply Default Animation is disabled to eliminate possible animation conflicts.

- Changed Hemi lamp behavior.

If the object is being illuminated by the Hemi lamp, the Lambert shading model will always be applied to its materials. This is done to match the Blender's behavior.

- Support for exported-to-HTML video textures in Firefox browser.

Since Firefox 38 an error with video textures exported to base64 was fixed, so it's now possible to use them in such browsers.

- Changed assignment of UV layers to match Blender's behavior.

Missing from node materials UV layers are determined automatically as it's done in Blender.

- Improved stability of exported to HTML video textures.
- Optimizations of CSM shadows.
- Depth shader optimizations.
- Billboard objects optimizations.
- Configuration option resolution_factor was renamed to render_resolution_factor.
- Improved support for particle emitters which have several materials.

Distribution of the particles to emitters as well as vertex colors inheritance is being done taking into account each material i.e mesh part.

- Changed Wind Bending inheritance on particle systems.

If Wind Bending Inheritance property is set to Instance then Wind bending property for emitter object is not switched off anymore.

- Updated messages about addon/engine version incompatibilities.

For more info see [version errors](#).

- Remove sensor locks API.

Unused sensor locks API was removed from controls.js module.

- Fixed behavior of node materials with missing Output node.

修复

- Fixes in screenshooter.js addon.

Fixed an error with impossibility to take a screenshot.

- Fixed a bug in set_frame() method from nla.js module.

Fixed set_frame() inaccuracy.

- Improved exported stability.

- Fixed a bug with addon removal/update on Windows.

Refined binary loader in addon.

- Fixed a bug with shading from SPOT/POINT lamps.

- Fixed incorrect behavior of coordinate calculations in methods `get_coords_x()` and `get_coords_y()` from mouse.js addon.
- Fixed calculations of alpha channel in Outline effect.
- Fixed Wind Bending effect error.
- Fixed an error when particle's Scale was not taken into account on particle systems.
- Fixed synchronization error on animated EMITTER particle systems.
- Fixed a bug with shadows on billboard objects.
- Fixed incorrect exporting of Override Mesh Boundings settings.
- Fixed a bug with billboard rendering on iPad.

Known Issues

- Problems with updating of the add-on.
It's strongly advised to restart Blender after installing a newer version of Addon/SDK.
- NVIDIA 331 driver in Linux can cause WebGL errors.
- There is a [bug](#) with video textures on Chrome 43 for Android.

Please update your Chrome browser to Beta or wait until the next Chrome update.

v15.05

新功能

- Glow effect.
Supported [an effect](#) which occurs when the light scatters in the atmosphere and inside of the human eye and looks like a halo around glowing objects.
- The local development server runs automatically.

A new option has been added to the addon settings. This option turns on automatic start of [local development server](#) upon opening Blender. With the help of this functional web applications in development can be run without any preparations.

- Cube reflections.

Apart from plane reflections, there are now cube reflections available. There is a new Reflection Type option available when Reflective flag is set on the object. Setting it to Cube turns this feature on.

- More NLA options supported.

Added support of Blender's NLA tools: Scale, Muted, Reversed and Repeat. The support of these tools broadens capabilities of interactive scene developers.

Furthermore, to control NLA through API a new `nla.js` module was added. This module contains methods like `play()`, `stop()`, `get_frame()`, `set_frame()` that can play/stop NLA and get/set the current frame. New methods have been involved in implementing control panel interface in the Viewer app.

- Increased rendering possibilities for sky textures.

Influence parameters for sky texture rendering are now supported. Those parameters are: Blend, Horizon, Zenith Up, Zenith Down, “Blend”, Negative, RGB to Intensity, DVar.

- In node materials, the engine now correctly processes connections between inputs and outputs of different types.

It is now possible to connect outputs of one type with inputs of another type ([Non-critical error](#)). Now Blender's native behavior is supported by the engine.

- For Hair particles the options on the Rotation panel are now supported.

Now the engine fully supports the state of particles set in Blender. In particular not only location and scale are supported now, but rotation as well.

- Some demos for postprocessing effects demonstration were added.

Examples were prepared for the following effects: Bloom, Depth of Field, God Rays, Motion Blur and SSAO.

- Added a new module container.js.

DOM tree elements can be added with a specific depth relative to the depth of the canvas element with the help of container.js's insert_to_container() method. This method replaces CSS property z-index because location depth of elements is now determined by their position in the container element.

- Improvements in the physical engine.

Margin property of physical elements and materials is now supported. This option allows for improved stability of object collision simulation. Bullet engine was updated to version 2.83.

- API for changing camera controls mode.

Added methods of [changing movement style of the camera](#). The control mode switch example can be found in [Code Snippets](#) app in the “Camera Move Styles” section. Also has been added set_hover_pivot() method. This method allows to shift the control point of HOVER camera.

变更

- Reorganized SDK's scene list.

All the scenes in the Viewer app has been sorted by groups: App contains finished apps, Demos contains demo-scenes and examples, Tutorial Exports contains tutorials source files.

- Added syntax highlight in [Code Snippets](#) app.

[Code Snippets](#) app now has a new design. Also it was optimized for low-resolution screens.

- API controls of video and canvas textures have been changed.

A new parameter data_id has been added to these textures' control methods. This parameter contains an ID of dynamically loaded scene.

- Handling of animated bone excess has been changed.

The skeletal animation now just turns off when exceeding the maximum number of bones. It resulted in shader compilation error and unstable application behavior before.

- Some particle system properties have been renamed and now they are turned off by default.

In particular, Hair particle system's properties Randomize Location And Size and Randomize Initial Rotation are turned off by default now.

- Doppler effect for speakers is now turned off in some browsers.

Doppler effect support in Web Audio has been pronounced as deprecated and will be removed in Chrome starting from version 45. Other browsers still support this functionality.

- Changed objects' behavior when both skeletal and vertex animation are applied.

If an object has both armature modifier and vertex animation applied on it, the armature modifier won't be exported.

- Rendering of particle system procedural animation (Wind Bending effect) has been optimized.
- The main .json and .bin scene files loading error handling has been improved.
- Windows 32 version of Blend4Web addon is now compiled natively.

This feature improves compatibility of the add-on with such systems.

修复

- The error that led to the wrong height of the description element in module "anchors.js" has been fixed.
- Support for Epiphany and other WebKit-based browsers.

Achieved by fixing the code which works differently in the different JavaScript engines.

- Gestures on Internet Explorer 11 were disabled for Microsoft Windows touch devices.

Previously, gestures usage (Windows Touch Gestures) was leading to unnecessary HTML-elements scaling and movement on such configurations. It is expected that correct gestures' behavior will be supported in further browser releases.

- Vertex animation with animated armature bake error was fixed.
- The error with rendering billboard objects on the iPad has been fixed.
- Node material's NLA animation applied to several objects was fixed.
- Fixed a bug related to the Motion Blur effect.

Known Issues

- Problems with updating of the add-on.

It's strongly advised to restart Blender after installing a newer version of Addon/SDK.

v15.04

新功能

- Deformations by using Shape Keys (Morphing).

Added support for object's [Morph targets](#) (known as [Shape Keys](#) in Blender). To apply such keys, use the `set_shape_key_value` method of the `geometry.js` module. Simple example of how to use such functionality is given in the [Code Snippets](#) app.

- Support for Horizon Color and Zenith Color background settings.

It's now possible to tweak scene background by using the Horizon Color and Zenith Color properties as well as the Paper Sky, Blend Sky and Real Sky options directly from Blender.

- Support for the Gamma node.

We have implemented the Gamma node back in Blender v2.74. Now this node is finally supported by Blend4Web.

- Various improvements in the Anchors tool.

It's now possible to limit the pixel size of an annotation. Added support for dynamic loading/unloading of Anchors. Implemented the possibility to hide and show Anchors by using the `show()`/`hide()` API functions and/or by the NLA Script tool.

- Shader optimizations.

Shader compiler improvements. Added the following features: local variables optimizations, brackets removal. Improved the performance of node materials.

- Physics engine optimizations.

To save the load time, the physics modules are now loaded only when explicitly required. Overall size of the modules has been decreased by 20%.

- Extended tools for physics debugging.

The new `physics_stat()` method has been added to the `debug` module. This method returns physics statistics such as the number of physics objects (separated by type), amount of geometry and other info. It is now also possible to display the number of physics iterations per second aka Physics FPS (activated in the config module).

- The new API method to attach objects to the camera independently from the camera's aspect ratio or the field of view.

Implemented in the `append_stiff_viewport()` method of the `constraints` module.

- The new module to perform transformations: “`tsr.js`”.

This new module makes it possible to apply a variety of transformations to objects by using versatile TSR vectors. Each TSR vector combines translation, scale and rotation (hence the name). These vectors may be used instead of matrices as a more convenient and effective way to apply transformations.

- The possibility to exclude any directories from being converted by the resource converter.

To exclude some directory from being converted by the [resource converter](#), it is enough to place a file named `.b4w_no_conv` to this directory.

变更

- The API documentation has changed its appearance. Links for quick access to methods and properties were added.
- Skeletal animation now takes into account the relative translation of an armature and a skinned object.

Native Blender's behavior is now supported. There is now no need to position an armature and an animated object in the same place and with the same rotation and scale.

- Independent translation, rotation and scale animations are now supported.

The animation system no more forces keyframes to be present in every channel, which makes it possible to save original values in unused channels or change them with API.

- World background support.

Background and sky can be enabled with the [Sky Settings > Render Sky](#) option under the World tab. It is turned off by default.

- The Uranium physics engine now consists of two modules.

There are now two physics modules instead of one: `uranium.js` - the engine's code and `uranium.js.mem` - the file for memory initialization. Both modules must be placed in the same directory.

- The glow effect together with its related components was renamed to Outline.

The new name better describes the principle of this effect: highlighting objects' edges.

- Limiting the camera translation using the “`append_semi_stiff_cam`” method of the “`constraints.js`” module is now possible only for the “EYE” type.

- Local Development Server in Blender changes.

Now, instead of the standard Python SimpleHTTPServer, the Tornado web server is used as the [local development server](#). The new server has greater performance and it also offers more options to disable browser cache.

- Keyboard controls for sliders in the Viewer application were added.

You can now control sliders with < and > keys.

- Changes in the “update_object_animation” method of the “animation.js” module.

The optional “force_update” parameter was added. It forces animated objects to be updated even when their animation is not being played back.

- API changes in the mouse.js module.

The enable_mouse_hover_outline and disable_mouse_hover_outline methods were added.

The following methods were declared deprecated and will be removed in the next releases: enable_mouse_hover_glow and disable_mouse_hover_glow.

- API changes in the scenes.js module.

The outlining_is_enabled, set_outline_intensity, get_outline_intensity, apply_outline_anim, apply_outline_anim_def, clear_outline_anim, set_outline_color and get_outline_color methods were added.

The following methods were declared deprecated and will be removed in the next releases: set_glow_intensity, get_glow_intensity, apply_glow_anim, apply_glow_anim_def, clear_glow_anim, set_glow_color and get_glow_color.

- Some settings were changed in the Object > Blend4Web panel.

The Enable Outline option was added to enable using the outline effect on the given object. Also, the Outline on Select option was added to activate glow animation when the object is selected (previously this behavior was defined by the Selectable flag).

- New settings were added to the Scene > Blend4Web panel.

The Enable Object Outlining option was added to control the overall possibility of outlining. Similarly, the new Enable Object Selectable option controls the overall possibility of objects' selection.

- Now the following object properties: Apply Scale, Apply Modifiers, Export Vertex Animation, Export Edited Normals and Export Shape Keys are mutually exclusive.
- API changes in modules.

The new `is_armature` method was added to the API of the `util.js` module. It checks if the given object is of the ARMATURE type.

The new `get_parent` method was added to the API of the `constraints.js` module. It returns the parent object of the given object.

修复

- Fixed a bug in the “anchors.js” module which caused objects’ descriptions to disappear.
- Fixed a bug in the Animation Baker script that occurred when there were armature objects in hidden layers.
- Fixed the camera’s behavior while using “`append_semi_stiff_cam`” method of the “`constraints.js`” module.

Fixed correction of the camera’s vertical axis relative to the parent object. Also the original camera orientation is now being taken into account. This can require some adjustments of the camera’s rotation limits that are passed to this function.

- Fixed a bug with reloading of the playlist when it was empty.
- Fixed the buggy behavior of physical objects that occurred after deleting at least one of them from the scene.
- Fixed a bug that occurred when there were zero-scaled objects instanced through DupliGroups.
- Fixed a compilation error of the water shader occurred on Windows and some mobile devices.

- Fixed a bug that occurred when there were duplicates of animation keyframes.
- Actions from different files sharing one name can be now used for NLA animation.
- Fixed duplication of event listeners that occurred when the “pointerlock” function was repeatedly called.
- Fixed behavior of the “Alpha sort” transparency type for dynamic objects.
- Fixed an add-on compilation error that occurred on Windows without C++ 2010 runtime installed.
- Fixed a bug with billboard rendering on iPad.

v15.03

新功能

- New tool for adding annotations to 3D objects.

Now it is possible to assign anchors to empty objects. These anchors can be of three different types: “Annotations” - information from object’s meta-tags is used, “Custom Element” - a custom HTML-element from the current web-page can be used as annotation, “Generic” - an invisible anchor with coordinates calculated using anchors.js module API.

- Animation and API methods for Value and RGB nodes in node materials.

Now it is possible to animate not only Value nodes but also RGB nodes. Also, the corresponding API methods for changing such nodes were added in the objects.js module.

- New “Code Snippets” application.

This [application](#) was created to simplify access to the examples of engine’s functionality. It is also possible to look at the examples’ scripts. This application can be launched from the index.html file located in the Blend4Web SDK’s root directory.

- New control functions for the Glow effect.

New APIs were added in the scenes module: `get_glow_intensity()` and `get_glow_color()`.

- Improvements in the Scene Viewer.

Design of the “Home” button was changed. A new button “All objects selectable” was added. It allows to turn off automatic “Selectable” option assignment for all scene objects. Also, it is now possible to see the total number of shaders on the loaded scene.

- Dynamic copying of scene objects (instancing).

Now it is possible to dynamically copy and remove scene objects (to create and remove instances).

- Handling errors related to the `B4W_PARALLAX` Blend4Web-specific node.

In case of incorrect usage of the `B4W_PARALLAX` node, an `export` error warning is generated.

- New options in the application’s builder.

There are now new options in the application builder: `-j` and `-c`. They add scripts and styles correspondingly to the exceptions in order to be not compiled.

- Experimental Blend4Web render engine.

It can be turned on in the addon settings using the “Register Blend4Web render engine (Experimental)” flag. This mode is designed to simplify customization of scene properties. Also, it simplifies the interface by removing unsupported panels. At the moment, it is not possible to edit shader node tree in the Blend4Web render mode.

变更

- Origin for counting off the camera limits has been changed.

Setting camera movement limits via API now perfectly corresponds to values measured in the engine’s coordinate system. Setting horizontal limits for the TARGET camera in the Blender’s world space has been changed. Thus, it may require changeovers for old scenes.

- Documentation update for the camera settings.
- Horizontal and vertical limits of the camera rotation are completely independent from each other.
- Some APIs in the camera.js module were changed.

APIs of the camera.js module have undergone a number of changes.

New methods were added: `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`.

The following methods were declared as deprecated and will be removed in the next releases: `rotate_pivot`, `rotate_hover_cam`, `rotate`, `set_eye_params`, `get_angles`, `translate_hover_cam_v`, `set_hover_cam_angle`.

The `set_ortho_scale` and `get_ortho_scale` methods now print error message when they are applied to the Orthographic camera. The behavior of the `get_hover_angle_limits` method was also changed. This method now returns angle limits for the HOVER camera in the [down, up] format instead of [up, down] as it was before.

- The lighting system was significantly optimized.

Many of the lamp props are calculated at the compile time now. Now, there is no 4 lamp restriction imposed for some mobile devices.

- The HTML layout method was changed for apps using the app module.

Now, upon initializing an application using the app.js module, the dimensions of the created `<canvas>` element are completely determined by the size of the container element. Thus, if a `<div>` element is used as a container, the size of `<canvas>` will be zero by default since div's default size is zero. You can set correct parameters for the container with CSS and inline-style. Also, you have to use `resize_to_container()` method from the app module when the container is changed. The same effect may be achieved if the `autoresize` option is set upon initializing the application (in the `app.init()` function). The low level method for changing the element's dimensions with `main.resize()` function is still supported.

- Now, the `enable_controls()` function from the `app` module should be called without any parameters.
- Deprecated API methods were removed.

`textures.js` module : `stop_video`.

`scenes.js` module: `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`.

`physics.js` module: `set_character_dist_to_water`.

`material.js` module: `set_batch_param`, `set_max_bones`, `max_bones`.

`main.js` module: `set_shaders_dir`, `set_texture_quality`.

`data.js` module: `get_bpy_world`.

`controls.js` module: `sensor_make_positive`, `sensor_make_negative`.

`camera.js` module: `change_eye_target_dist` (deprecated `MS_CONTROLS` was also removed).

- There are now new conditions to allow changing object's position via API.

The functions from the `transform.js` module related to changing object position can be now applied to [dynamic objects](#) only.

- You can now use TEXTURE nodes without textures.

In this case the rendering of the node material completely corresponds to Blender.

- Updated the procedure of compatibility checks for versions of exported files and the engine itself.

The engine will report about the scene's incompatibility by printing [messages in the browser console](#).

- The “Do Not Batch” property was renamed to “Force Dynamic Object”

This option instructs the engine that the object must be dynamic regardless of other settings. Now its name is more clear.

修复

- Fixed camera autorotate feature of the web player.
- Fixed an error related to the fullscreen mode in the Web Player.
- Fixed an error related to the determination of the camera's angular coordinates in some positions.
- Fixed an error with camera autorotation if the horizontal limits are enabled.
- Fixed an error when Blend4Web-specific nodes were being added multiple times to a .blend file.
- Fixed a bug with replacing materials by using the “inherit_material” function from the “material.js” module.
- Fixed an error occurred while rendering reflections on an object which has been changed through the “material.js” module APIs.
- Fixed generation of the debugging wireframe spheres.
- Fixed optimization of the TEXTURE nodes in node materials.
- Fixed “Clamp” option behavior in the MixRGB (Linear Light) node.
- Fixed an export error occurred when an object shares its mesh with another object and one of the following flags is set to true: “Apply Scale”, “Apply Modifiers”, “Export Vertex Animation” or “Export Edited Normals”.
- Fixed an error with “Blend4Web > Preserve Global Orientation and Scale” option on some mobile devices.
- Fixed fog rendering error in some versions of Chrome/Firefox under Windows.

Known Issues

- Problems with updating of the add-on.
It's strongly advised to restart Blender after installing a newer version of Addon/SDK.
- Video textures do not work in Firefox for scenes exported as HTML files.

v15.02

新功能

- The local development server can be run from Blender.

It's possible now to run local development server when using Blend4Web SDK. This server allows fast access to the Blend4Web SDK content and also make it possible to [automatically open exported scenes](#) in the Viewer application.

- Support for Spot lights shadows.

Shadows for Spot lamps were processed in the same way as for Sun lamps. Now the calculations are performed in the same way as it's done in blender i.e the light scattering is taken into account.

- Added/improved support for “Metaball”, “Surface” and “Curve” objects.

Support for Metaball, Surface and Curve objects was added. Objects of these types are automatically converted into meshes during export. Support for Curve objects in modifiers was preserved.

- Social networks buttons are added to the Web Player.

These buttons allow placing a link and a description to the 3D scene in one of the four popular social networks.

- Added support for editing the list of Viewer scenes directly in Blender.

It is now possible to [edit assets.json file](#) with a list of Viewer scenes inside Blender. This works only when using Blend4Web SDK.

- Screenshots can now be taken in the Viewer application.
- New fallback_video option is added to the Web Player.

Now the user can choose a video file to play instead of 3D content on systems without WebGL support. It is possible with the help of the new `fallback_video=/path/to/video/` option.

- Improved rendering to texture functionality.

Added support for rendering scenes into several textures at a time. Scenes now can have any nesting level.

- Billboards now can save orientation and scale in world coordinates.

To use the feature you need to set [Blend4Web > Preserve global orientation and scale](#) flag in the object's settings panel.

- Improvements on the main SDK web page.

It is now possible to find out the version of the SDK and check the system for WebGL compatibility on the main web page.

- Added support for the Clamp flag in MATH and MIX_RGB nodes

At first this functionality was released in Blender 2.73, and now it's also supported in the engine.

- Considerable improvements in rendering quality on systems without depth-texture support.

Supported rendering features on systems without depth-texture support were extended. There are such effects available now: reflections, bloom, glow, motion blur, anti-aliasing.

- New documentation is added.

Added documentation for the vec3, vec4, quat, mat3, mat4 modules and for the global namespace b4w. Documentation web pages design was improved.

- Support for several engine instances on the same web page.

Several engine instances can now work simultaneously, by specifying the namespace on engine's initialization stage.

- Possibility to use SDK on Apple macOS.

On macOS all SDK functionality including engine and applications building, resource conversation and documentation generation is now available.

- The new set_trans_pivot() method is added to the camera module.

This function allows setting an arbitrary position of the pivot point and the position of the camera of the Target type.

- A new “version” property is added to the “project.py” utility.

This property allows adding a version to the scripts and styles of the compiled application.

变更

- Now it is possible to add options with the same names via browser address bar.

A new optional parameter `allow_param_array` is added to `get_url_params()` function of `app` module. It is set to false by default. Setting this parameter to true leads to a merge of identical functions into a massive, other way the last one will be used.

- Improved “Background Music” speaker.

Now the user can specify parameters of delay and playback time.

- Blend-file now includes special Blend4Web nodes by default.

Now there's no need to add [Blend4Web special nodes](#) into a file. It is available in both SDK and addon versions of Blend4Web.

- Changed export of empty “Mesh” objects.

Now Mesh objects without polygons are exported as Empty.

- Changes in the “light.js” module.

Added function `get_light_type`; functions `get_light_params` and `set_light_params` now get object LAMP instead of objects name. Also the user now can change `spot_blend`, `spot_size` and `distance` properties of the SPOT light source through those functions.

- Improved refractions on LOW quality settings.

New simplified (without distortion) refraction model is now used when LOW quality is chosen.

- Shader nodes optimization.

- Now automatic camera rotation can be turned off by touching touch screen.

修复

- Corrected behavior of the preloader for the Web Player.

Some artifacts could be visible on B4W logo while opening Web Player.

- Fixed an error in rendering of one-cascaded shadows. The error was related to appearing of a hard non-lighted line on the cascade border.
- Fixed an error related to names collision while linking objects in Blender.
- Fixed an error with optimization of SDK apps.
- Fixed export error of flat shaded meshes on Linux x32.
- Fixed incorrect behavior of Target camera in particular cases.
- Fixed an error when using a shadow map with size exceeding device limits.
- Fixed an error that leads to FPS drop in Firefox 35/36 on Windows when shadows are turned ON.

Known Issues

- Problems with updating of the add-on.

It's strongly advised to restart Blender after installing a newer version of Addon/SDK.

- Video textures do not work in Firefox for scenes exported as HTML files.

v15.01

新功能

- Support for panning on touchscreen devices.

Panning is performed by swiping two fingers on the surface of the screen.

- Support for “Text” objects.

These objects are now automatically converted to meshes during export.

- Extended support for the NLA Script tool.

Added new logic slots: Show Object and Hide Object used for hiding and showing the objects, Page Redirect - for redirecting to other webpages , Page Param - for storing any webpage parameter in given numerical register. Simplified usage of Select & Jump and Select & Play slots. Now it's not required to specify Selectable property on selectable objects.

- Support for high definition displays (HDPI, Retina).

The HDPI mode allows to achieve considerable improvement of picture quality on devices with high resolution. This mode is activated automatically upon application startup if ULTRA quality profile has been selected. If necessary, high resolution can be turned on for other quality profiles.

- Support orthographic camera scaling.

An API to change the Orthographic camera scale has been added (Orthographic scale in Blender).

- “autorotate” option has been added to the webplayer.

The [option](#) autorotate is used to turn on the automatic camera rotation as soon as the scene loads.

- Simplified keyboard control mode has been added to function “enable_camera_controls.

The mode is enabled by passing the optional parameter disable_letter_controls Thus, the keyboard controls with letter keys (WASD and so on) will be turned off. This feature can be used in cases when you need to use the letter keys for purposes other than moving the camera.

- Support for gyroscope on mobile devices.

To work with gyroscope on mobile devices the two sensors was implemented. The first sensor allows to operate with current device position compared with the previous one (position delta). It's created by using the function `create_gyro_delta_sensor` from “controls.js” module. The second sensor returns current device angle and created by `create_gyro_angles_sensor` from “controls.js” module. It's worth to mention that all angles are given in radians. Also the special addon “gyro-

scope.js” was created. This addon implements simple camera movements due to device rotation. You can find an example of using such feature in our Viewer application by selecting the “Gyroscope” menu option.

- New “Do not Render” property has been added to material settings.

Enabling the property allows to hide parts of the scene objects which use such material.

- Support for video-textures on IE 11 and iPhone.

The support is achieved by creating a new video-sequence format, *.seq.

For more info check the following [topic in documentation](#).

- Support for “title” tag in Web Player.

The Web Player’s title (shown as web browser header) is now extracted from JSON file of the loaded scene. For more info about this feature check the following [topic in documentation](#).

- Support for meta tags in Blender.

It’s now possible to append meta tag information to scenes and objects in Blender. Possible tags for scenes are “title” and “description”. Possible tags for objects are “title”, “description” and “category”.

- Added support for execution of user-defined functions every frame.

To help users to create sophisticated application the new function `append_loop_cb` has been added to “main.js” module. This function allows to execute given callback every frame. This callback has two parameters: time since the application start and time delta between current and previous frame. Both parameters are in seconds. To cancel the callback execution every frame you should remove it by using the `remove_loop_cb` function from the module “main.js”.

- Added support for simple preloader screen animation.

To create an application with animated preloader pass an option “`preloader_fadeout`” with the value “true” to the function `create_simple_preloader` from “preloader.js” module.

- Added support to export converted media files to HTML file.

Now then you export HTML files it's possible to store converted files in them. To do so you need to enable "Export Converted Media" option in the [export options](#).

- Added support for using min50 and dds textures in Web Player.

To enable this feature pass "compressed_textures" parameter to Web Player.

变更

- SDK file hierarchy was simplified.

The external directory was removed, all its content was moved to the upper level - in the root directory of the SDK. The file with the list of the scenes used by the Viewer application is now located in apps_dev/viewer directory.

- Changed camera autorotation behavior (experimental).

If the camera limits are present, the camera smoothly slows down when approaching the limits, then moves in the opposite direction.

- The usage of "Special: Collision" property was changed.

Earlier, enabling the option automatically resulted in objects' hiding.
Now, to do the same thing, you have to specify Do not Render property in the material settings.

- Changed suffix for converted media files.

Old *.lossconv.* suffix was replaced by *.altconv.*.

- Behavior of "Do not render" object property was changed.

Now, when the option is activated, an object's physics is not disabled.
The object simply becomes invisible.

- Improved the SDK structure.

Free and Pro SDK now come with the new and more polished examples;
old and arid examples were dropped.

修复

- When using “Panoramic” camera type in Blender the camera automatically obtains “Perspective” type when exported.
- Fixed bug with “Target” camera dragging in rare cases.
- Minor fixes in the “B4W Anim Baker” addon.
- Fixed issue with sound for scenes with multiple cameras.
- Improved stability of “Timer” sensors in “controls” module.
- Fixed issue when browsing exported HTMLs in IE browser.
- Video texture optimizations. Now the video texture is not updated for suspended video playback.
- Fixed rendering issue in node materials with “REFRACTION“ node.

Known Issues

- Problems with updating of the add-on.
It's strongly advised to restart Blender after installing a newer version of Addon/SDK.
- Video textures do not work in Firefox for scenes exported as HTML files.
- Slow and unstable rendering of depth textures in Firefox 35.

In various scenes the FPS is degraded when shadows are turned on. There is also an incorrect rendering of transparent materials. The issue is reported [here](#) and is to be fixed in the future browser updates.

v14.12

新功能

- Camera velocity settings are now available.

Now [camera movement velocity](#) can be set up, including translation, rotation and zooming. Velocity for all camera types (Target, Hover, Eye) can be tweaked both using Blender UI and through Blend4Web API.

- Mipmapping is now supported for Canvas textures.

We have added mipmapping support for Canvas textures.

- Full support for the “MAPPING” node.

Now all Vector type options available for the MAPPING node are supported, including Texture, Point, Vector and Normal.

- Glow on mouse over.

API in the mouse.js module were created for the effect of outlining the objects under the mouse pointer. Also, for purposes of controlling this effect, enable_mouse_hover_glow() and disable_mouse_hover_glow() methods were added. The objects should have Object > Blend4Web > Selectable checkbox enabled.

- A brand new app building system.

Now, the users can develop their apps right in the SDK thanks to the new project.py utility. [This script](#) makes it possible to build the apps together with the engine, to minify JavaScript and CSS files and to export the final apps to be deployed on a server.

变更

- Removed support for deprecated “UV translation velocity” texture settings.

We recommend to use node materials instead.

- Removed deprecated “Levels of Detail” user interface.

This functionality can be used through Blender’s standard “Levels of Detail” tool.

- The pointerlock.js add-on was renamed into mouse.js.
- A mouseup event is now fired when the mouse pointer is leaving the app viewport.

So the problem with broken camera controls is no longer observed.

- Error message about “Clear parent inverse” is no more shown.

Before, when using parenting, it was required to reset translation, rotation and scale of child objects (Object > Parent > Clear Parent Inverse). Now such transformation is natively supported by the engine.

- “Apply scale” option no longer applies modifiers.

As before, [Apply modifiers](#) should be used in order to apply modifiers.

- Use of normal maps in node materials no longer requires a Material or Extended Material node.

In some cases (e.g. refraction) normal maps can be used in shadeless materials.

修复

- Fixed audio playback error occurred when using NLA.

This happened due to insufficient float number precision.

- Fixed incorrect rendering of light sources on mobile devices.
- Layering shadows is fixed when multiple active light sources are present.

Now the shadows are calculated like in Blender, that is areas illuminated by other light sources are not darkened.

- Node material rendering error has been fixed.

The error occurred when a MATERIAL node (or MATERIAL_EXT) with a linked (from another .blend file) material was used.

- Animation baker (“B4W Animation Bake” operator) no longer resets an armature pose.

Now, when using the [animation baking tool](#), the armature pose is being left intact.

- Fixed jerky camera movement upon application startup.
- Fixed error with incorrect determination of the camera’s horizontal movement limits.

- Fixed error occurred when unused textures were exported.

Known Issues

- Problems with updating of the add-on.

It's strongly advised to restart Blender after installing a newer version of Addon/SDK.

- Video textures do not work in Firefox for scenes exported as HTML files.

v14.11

新功能

- Video textures support.

Video textures are now supported for Image or Movie textures.

- Frame rate.

Frame rate for animation and video textures can now be changed through the Scene > Dimensions > Frame rate option.

- Canvas textures support.

A canvas HTML element can be now used as a [texture](#). The workflow is described in the manual.

- Camera panning.

In the mode when the camera is rotating around a single point (Target) the users now have the ability to move the pivot point within the view plane (so called camera panning) while the right or middle mouse buttons are pressed. This function is turned on by default and can be turned off in Blender settings at need.

- New camera control mode - Hover.

The Hover mode is now available when the camera is gliding over the horizontal plane (including zooming in and out). This camera mode makes

it possible to realize scenarios for a convenient viewing of scenes which are spread in two dimensions (rooms, game levels).

- The SDK now contains a root index HTML webpage for simplifying navigation within the distribution.
- The resource converter now has the ability to convert videos.
- We have added a build system which was absent in previous public SDK distributions.
- The export Strict mode is implemented in the add-on.

Activating Strict mode gives the possibility to display all possible errors and warnings connected with incorrect scene settings. The option is useful for final scene debugging for getting the most correct and optimized resource files.

- Audio playback support for iOS devices.

变更

- The webplayer's "bg" parameter is renamed to "fallback_image".

This option also has changed its behavior. If the fallback_image is defined the error message that WebGL is unavailable is not shown anymore, instead the user sees just the image.

- If there are no sound sources in the scene the sound mute button is no longer shown in the webplayer.
- Generic materials workflow is now more predictable.
- The "mouse_down" sensor provides the code of the mouse button pressed. This code can be obtained from the payload sensor's parameter.
- Hair particle systems can be now exported significantly faster.

修复

- Normal maps now work with Generated and Normal texture coordinates.

Using UV layers is no more required for normal maps.

- Fixed the problem with the wrong path to the physics engine in the webplayer.

This error arose when `uranium.js` was moved out of the directory containing the main HTML file of the webplayer.

- In the add-on we have fixed the problem with packed textures. Export crashed when the “Automatically Pack Into .blend” option was enabled.

Known Issues

- Problems with updating of the add-on.

It's strongly advised to restart Blender after installing a newer version of Addon/SDK.

v14.10

新功能

- A new Web Player.

The new minimalistic Web Player design blends perfectly with any 3D scenes. It has a simplified user interface and build-in help. The Web Player works on all devices including mobile ones.

- Improved shadows.

It's now possible to choose a non-cascaded shadow model, based on a single optimized shadow map. Such model is easier to configure and suits well for relatively small scenes. For more info see the [docs](#).

- Many NLA system improvements.

It's now possible to create a complex logic using the Conditional Jump, Register Store, Math Operation NLA Script logic blocks and register-stored variables.

It's now possible to use all types of supported animations in the NLA, including sound playback, vertex animation and particle emission. It's now possible to play different animation types simultaneously.

- Supported objects billboarding.

The objects received the new set of [options](#), allowing to configure billboarding.

- The “XYZ Euler” mode is supported for animating rotations.

Object and skeletal animations now support the XYZ Euler mode for rotations.

- Support for the GENERATED texture coordinates.
- Support for Cross-origin resource sharing (CORS).
- Scene export process simplified.

The range of material export errors are now not blocking the export. Instead, this material will be highlighted pink at scene loading. Detailed error descriptions can be found in the [manual](#).

- Added support for the “Do not export” option for particle systems.
- Improved stability on iOS devices.

变更

- Changed SDK path setting for HTML export.

It's now needed to set the SDK path setting for HTML export to SDK root directory. Earlier it was required to provide the full path to embed application. Pay attention, the old behavior is not supported anymore.

- Deprecated the “UV translation velocity” option.

The option will be removed since version 14.12.

- Removed option “Do not export” from the “Object data” panel.
- Removed “Blend4Web > Animation > Cyclic” option from the object properties panel.

Instead, one should use Blend4Web > Animation > Behavior option located in the same place. Scenes with default animations may work incorrectly, so they need behavior property to be set to Cyclic.

- Modified SSAO algorithm realization.

The new implementation is much faster and shows better quality. The settings of the algorithm are changed too. For more info see the [manual section](#).

修复

- Fixed rendering error for HALO materials.
- Fixed a rendering error when an object with the enabled “Force Dynamic Object” property has a parent object.
- Fixed error with keyboard shortcuts in Blender.

Fixed error with inability to assign keyboard shortcuts in Blender for export menu items:File->Export->Blend4Web(.json) and File->Export->Blend4Web(.html).

- Fixed crash when loading textures with size exceeding device limits.
- Fixed node material errors resulting in unstable engine behavior.
- Fixed error in node materials that contained complex Node Groups.
- Fixed errors of shaders compilation on devices with mobile graphics Qualcomm Adreno 305.
- Fixed rendering error when using REFRACTION nodes in transparent materials.
- Fixed an issue in “B4W Vertex Anim Baker” tool when current frame reset was occurred after using bake.

Known Issues

- Problems with updating of the add-on.

It's strongly advised to restart Blender after installing a newer version of Addon/SDK.

- Normal maps don't work for Generated texture type.

It is necessary to use UV mapping for normal maps.

v14.09

新功能

- ABSOLUTE type support for the MATH node.
- Support for LEVELS_OF_QUALITY special node.

Allows to control the material's complexity depending on the quality profile which is specified by the user upon engine start.

- Support for SMOOTHSTEP special node.

Simplifies the creation of some effects in node materials.

- Node groups support.

Node groups allow the sharing of node blocks between materials.

- The ability to output intermediate rendering results for debugging.

The rendering result for a certain stage can be now output above the main picture. This can be set up in the config.js module through the debug_subs options.

- The logic for controlling Blender's NLA animation using a visual editor has been implemented.

The NLA Script tool has been added to Blender's interface to allow the implementing of simple scenarios using visual blocks, for example playing an animation in response to the user actions.

- Multiple sensor system improvements.

It is now permitted to register sensor manifolds globally using a controls module method with no connection to any object. To do this null should

be passed into the corresponding API. The sensor logic is processed in a more predictable and robust way according to the sequence in which their manifolds are created. Callbacks of the user action events are now assigned using the `register_<input_type>_events()` functions. To these functions it is now possible to pass the `prevent_default` flag which allows to unblock the standard browser behavior for the corresponding events.

- The Web Player now supports physics.

Works only in the Web Player version where JSON files are loaded separately. Physics still not supported in the single HTML files.

- Skeletal animation mixing is now supported.

The `animation.js` module now contains API for smooth transitions between skeletal animations: `get_skel_mix_factor()` - for getting the current mixing factor value and `set_skel_mix_factor()` - for setting it.

- The Value node can now be animated in node materials.

The functionality is similar to other animation types. Working in NLA is also supported.

- Diffuse and Specular lamp's properties are now supported.
- The possibility to render a transparent object above other objects on the scene.

Activated with the Render above all checkbox for transparent materials (i.e. not Opaque).

- Scale is applied automatically to the object mesh.

Activated by enabling the Apply scale checkbox in the object settings.

- High quality profile (including shadows, dynamic reflections and anti-aliasing) has been implemented for iOS.

变更

- Shadow rendering improved.

Shadow rendering system is significantly changed: it is now based on the Stable Cascaded Shadow Maps technique. This technique allows to

greatly diminish the flickering of shadow edges when the camera moves.

Smoothing is implemented between cascades. Also shadows of the last cascade fade out at distancing. Softened shadows are rendered using the Percentage Closer Shadows technique. The shadows' user settings are reworked and simplified. Now it's possible to tweak the size of shadow maps, blur ratio and the setting for removing self-shadowing artifacts.

The new settings are [documented](#) in detail.

- In the Web Player graphics quality settings are now saved independently for each scene.
- The behavior of the app configuration parameters has been changed: `physics_uranium_path`, `smaa_search_texture_path` and `smaa_area_texture_path`.

These parameters are now calculated automatically depending on the running HTML files location, if they haven't been overridden during the app's initialization.

- Transition is completed to the system of modules which are linked via `b4w.require()` call.

This also means that starting from the current version it's impossible to call modules in the engine's release version using the old `b4w.<module>` namespaces. For compatibility purposes the `ns_compat.js` add-on has been implemented, the linking of which allows to restore the old behavior.

- The Web Player's control panel can now be hidden.
- Skeletal animation is now applied to armature objects only.

There is no need to apply skeletal animation to MESH objects. If they are linked to some animated armature, their skinning will be automatic.

- Demos and tutorials are updated according to the newly implemented features.

修复

- The preloader didn't disappear in case of a loading error (texture or sound file).
- Lagging during scaling and turning on mobile devices is fixed.

- TARGET-type camera shimmering has been removed for small turnings.
- EYE-type camera controls was fixed for mobile devices.
- The Farm demo controls are improved for Safari browser.
- Errors concerning using the unsupported shading models in node materials are now fixed.
- “Selectable” option now works for the objects without materials.
- There is no longer need to enable “Force Dynamic Object” for the objects that are animated using NLA.
- The particle system error when the object being instanced is parented to another object, has been fixed.

Known Issues

- Problems with updating of the add-on.
It's strongly advised to restart Blender after installing a newer version of Addon/SDK.
- Armature animation mixing doesn't work with some browsers.

If skeletal animation mixing API brings unexpected errors, it is necessary to override standard Math.sign function as follows:

```
var m_util = require("util");
Math.sign = m_util.sign;
```

- genindex
- search

B

Blend4Web 模式, 1

Blender, 3

G

git, 463

准备提交, 467

单独设置, 466

存储库之间同步, 471

提交, 470

标签, 476

检查状态, 466

添加和删除文件, 468

解决冲突, 473

R

release notes, 489

RTT, 323

S

SSAO, 296

W

WebGL, 2

优势, 3

浏览器支持, 2

三维建模, 3

三维引擎, 1

上帝光芒, 298

交互式图形, 5

传感器, 440

流形, 441

光源, 279

光照, 277

动画, 352

包围, 60

反射, 198

动态, 199

菲涅尔效应, 200

静态, 199

四元数, 147

图形引擎, 1

场景, 102

天穹, 321

导出, 25

导出错误警告, 93

错误, 91

屏幕空间环境光遮蔽, 296

工具

调试, 59

引擎, 1

引擎功能, 6

批, 60

抗锯齿, 302
 插件, 78
 初始化错误, 89
 显卡, 6
 景深, 294
 景深, DOF, 294
 曙暮辉, 298
 材质, 193
 光晕, 204
 具体参数, 203
 反射, 198
 渲染属性, 201
 灯光参数, 195
 节点, 220, 222, 223
 视图属性, 202
 透明, 196
 材质捕捉, 311
 材质捕捉 (matcap) , 311
 查看器
 启动, 82
 法线, 160
 编辑器, 160
 浏览器, 4
 浏览器技术, 4
 海岸线, 376
 海岸线数据, 376
 渲染, 60
 渲染到纹理, 323
 漫射贴图, 311

 版本
 错误, 90
 物体变换 API, 142
 着色器编译, 456
 WebGL 的扩展, 460
 优化, 457
 建议和限制, 459

 混淆, 457
 输入/输出指令, 458
 错误, 461
 验证, 457
 立体影片, 303
 簇, 60
 粒子系统
 实例化, 339
 流体, 327
 纹理, 308
 天空, 321
 法线贴图, 312
 渲染到, 323
 漫反射, 311
 环境贴图, 318
 类型, 309
 蒙版贴图, 314
 设置, 310
 镜面贴图, 320
 高光贴图, 312
 高度贴图, 313
 线框, 60
 网络播放器, 71

 背景, 290
 色彩校正, 390
 获取物体, 144
 著色, 278
 类型, 278
 视差贴图, 313
 轮廓辉光, 300
 运动模糊, 293

 透明, 196
 类型, 197
 设置, 198

预览, 55, 60

添加场景, 25

驱动程序, 6