Blend4Web. User Manual

Release 14.07

Triumph LLC

July 23, 2014

Contents

CHAPTER 1

Overview

1.1 What's Blend4Web

Blend4Web is a web-oriented 3D engine - a software framework for authoring and interactive rendering of three-dimensional graphics and audio in browsers.

The platform is intended for visualizations, presentations, online-shops, games and other rich internet applications.

The Blend4Web framework is integrated tightly with Blender - a 3D modeling and animation tool (hence the name). The content is rendered by means of WebGL and other browser technologies, without the use of plugins.

Technically Blend4Web is a library for web pages, a Blender addon and some tools for debugging and optimization.

The Blend4Web 3D engine has been developed by Triumph LLC employees since 2010. The engine was first released on March 28 2014.

1.2 About Engines

An engine is a separate part of software code which is used by external applications for implementing the required functionality.

Engine examples are: site engine, blog engine, online shop engine, wiki engine, search egine, game engine etc. The economical reason for the existance of software engines is multiple usage of the same functionality. For example developers may create relatively cheap online shops or games using one or another engine.

1.3 Graphics Engine, Game Engine

A graphics engine performs special functions in displaying graphics. It is an intermediary between:

- high-level application part (game logic, business logic) and
- low-level system part (for example, the graphics library WebGL and underlying drivers).

A graphics engine may be combined with the sound system, the physics engine, the artificial intelligence system, the networking system and the scene and logic editors producing a three-dimensional engine - an integrated environment for authoring 3D applications.

1.4 What's WebGL

WebGL (Web Graphics Library) is one of the modern browser technologies which allows authoring 3D graphics applications. In other words WebGL is "3D in a browser".

1.5 WebGL Browsers Support

At the moment WebGL is supported in to a varying degree by all browsers.

1.5.1 Full Support

- Yandex Browser
- Chrome
- Firefox
- Opera

1.5.2 Experimental Support

- Internet Explorer 11
- Safari 8

1.5.3 Mobile Platforms

- Android (on the majority of modern devices)
- BlackBerry
- Firefox OS
- iOS 8
- Tizen
- Ubuntu Touch

- WebOS

1.6 Advantages of WebGL

- works in browsers without installing additional software (plugins)
- crossplatform, intended for all desktop and embedded systems
- open standard, does not require licensing fees
- supported by the leading participants of the IT market (Google, Apple, Microsoft, Nvidia, Samsung, Adobe and others)
- based on OpenGL which is familiar to developers
- can be integrated with other browser technologies

1.7 What's Blender

Blender is a popular piece of software for 3D modeling and animation and is free and open source. Models and scenes which are created in this software can be displayed, for example, by means of a three-dimensional engine on a web page.

1.8 3D Modeling

Authoring graphics resources requires trained specialists - 3D artists.

A typical workflow may include the following stages:

- choosing photos and/or creating concepts and sketches (views from the front - from the side - from the above) of the future model or scene
- modeling - a 3D model consisting of polygons is created
- UV mapping - the model is unwrapped for further overlaying of textures (flat images)
- texturing - textures are overlayed on the 3D model
- materials setup - materials are assigned for different parts of the model and tuned (for example, a wooden door with a metal handle)
- rigging - the controlling elements ("skeletal bones") are attached to the model to animate it
- animation - the model is set in motion to visualize actions for example - of characters
- export - can be performed on any stage to display the 3D model in its final form, for example, on a web page

In addition, realism improving techniques are often used in the process of creating 3D models which require additional stages:

- creating a high-poly model - a detailed version of the model is created
- "baking" of a normal map - details from the high-poly model are transferred to the main model in the form of a special texture (normal map)
- creating a specular map - different reflection color and ratio are assigned to different model parts
- baking environment maps - is performed to visualize the surrounding environment reflection on the model surface
- setting up the camera and the light sources on the scene
- physical simulation parameters setup - particles, cloth

The time required to author 3D models and animation depends on their complexity and required quality and may vary from 1-2 days (for example a game item) to 1-2 weeks (for example a detailed aircraft model) and even to several months (realistic characters with clothing, hair, face sets, with animation and figure parameters setup).

1.9 Browser Technologies

Browser is a program for viewing Internet content. At the dawn of Internet technologies the browser's role was to view text pages with the inclusion of static images ("hyper-text"). Modern browsers are full-scale platforms for multimedia web applications.

Among the already implemented and promising browser features which are used in Blend4Web the following technologies can be noted:

- three-dimensional graphics, WebGL
- Typed Array
- Timing control for script-based animations (requestAnimationFrame)
- two-dimensional graphics, HTML Canvas 2D Context
- sound processing, Web Audio API
- binary data loading, XMLHttpRequest Level 2
- Fullscreen
- Pointer Lock
- multithreading, Web Workers

Other promising technologies:

- Scalable Vector Graphics (SVG)
- safe file access, File API, File API: Directories and System
- real-time communication between browsers, WebRTC

- persistent network connection, The WebSocket API
- Gamepad

1.10 Interactive Graphics

Applied to computer graphics the term “interactive” means that the user can interact with a constantly changing image. For example the user can change the view direction in a 3D scene, move the objects, trigger animation and carry out other actions normally associated with computer games.

Graphics interactivity is achieved by utilizing a frequent change of images, so the user action (for example a mouse movement or the pressing of a key) between frames leads to the image changing in the next frame. Images must replace each other so frequently that the human eye could not recognize them individually (at least 30 frames per second).

“Real-time graphics” or “real-time rendering” are also similar in meaning to the term.

1.11 Video Cards and Drivers

Interactive graphics is provided by a special-purpose hardware part of modern computers so called graphics processor which can be implemeted as a discrete device (video card) or as a part of the central processing unit.

Main graphics processors vendors for desktop computers are: - NVidia (GeForce, Quadro), AMD (Radeon), Intel (HD), for embedded devices - ARM (Mali), PowerVR (SGX), Nvidia (Tegra), Qualcomm (Adreno) (trade marks are specified in brackets).

Program access to graphics processor resources is carried out via an intermediate program called driver. It’s important for the correct working of interactive graphics programs to have drivers of the latest version in the system. Drivers can be installed (or upgraded) from corresponding websites of graphics processors vendors. See detailed info in the section WebGL Failed to Init.

CHAPTER 2

The Engine Features

2.1 Texturing

- texture mapping i.e. applying of a flat image to a 3D object surface
- multitexturing i.e. using multiple textures for an object
- render-to-texture, RTT for displaying one scene in another and for postprocessing effects
- anisotropic filtering, AF for enhancing the quality of surfaces at oblique viewing angles (standard WebGL extension is used)
- texture compression support (S3TC/DXT format)

2.2 Materials

- materials transparency, sorting by depth if required (z-sorting)
- enhanced detailing of relief surfaces with textures (parallax offset mapping method is used)
- Fresnel effect - dependency of reflectivity on viewing angles
- dynamic reflection
- node materials support
- halo material for rendering light sources and stars

2.3 Lighting

- multiple light sources
- light source types - directional, hemisphere, point, spot
- diffuse lighting

- ambient lighting aka environment lighting
- specular lighting - light reflection from surface
- environment mapping - surface reflects the environment
- normal mapping - additional surface detailing by textures

2.4 Shadows

- static shadow mapping (light mapping)
- dynamic shadow mapping
- self-shadowing - objects cast shadows on themselves
- cascaded shadow mapping, CSM for large scenes
- soft shadows

2.5 Particle System

- particle system for implementing effects such as fire, smoke, splashes etc
- particle system for instancing similar objects: grass, stones, tree leaves etc

2.6 External Scenes Rendering

- fog
- skydome for skies or environment
- lens flares effect
- water rendering

2.7 Postprocessing Effects

- motion blur
- anti-aliasing - image edges enhancement (fast approximate anti-aliasing, FXAA method is used)
- stereo (anaglyph method, 3D glasses required)
- ambient occlusion (SSAO method is used)
- depth of field, DOF

- crepuscular rays (god rays)
- bloom - bright light effect
- glow

2.8 Animation

- skinning - object deformation with a system of bones
- animating location, rotation and scale of objects, cameras and light sources
- skeletal animation (e.g. for a character's body)
- vertex animation (e.g. for cloth simulation)
- procedural animation (e.g. foliage wind bending)
- texture coordinates animation (e.g. for visualizing water waves)

2.9 Optimization

- frustum culling - invisible objects are not rendered
- batching, texture atlases - WebGL calls number are reduced
- level of detail, LOD - far objects are less detailed

2.10 Audio

- audio engine based on the Web Audio API
- various file formats support depending on browsers
- flexible playback control, sound pause/resume
- positioning sources in a three-dimensional space
- Doppler effect for moving objects with a possibility to turn it off and with space jump compensation
- flexible control of playback volume, speed and latency
- fade-in, fade-out, duck
- high quality sound looping
- randomizing sound parameters to improve loop percepetion
- cross-fader sound animation support
- dynamic compressor

- efficient long soundtrack storage and playback
- tools for real-time mixing

2.11 Physics

- rigid body physics - collision detection, realistic movement, gravity, height detection, torsion
- various constraints types - rigid, hinges, springs, pivots, sliding etc
- ray tracing
- floating and underwater movement physics
- wheeled vehicles simulation
- watercraft simulation

2.12 Event-Driven Model

- asynchronous framework for application logic authoring
- animation control and artificial intelligence of characters and animals

2.13 Other

- math curves support for modeling long objects (roads, wires, rivers)
- picking objects on the 3D scene with the mouse
- code minification and obfuscation for commercial use of the engine
- use of git - a distributed version control system
- module structure of source code
- powerful shader preprocessor with modules and functional blocks (nodes) support
- convenient system to deploy new 3D applications quickly
- tight integration with Blender - a 3D modeling and animation tool
- support options for a broad range of equipment
- user manual and API documentation
- user interaction - camera, character, actions control

CHAPTER 3

Quick Install

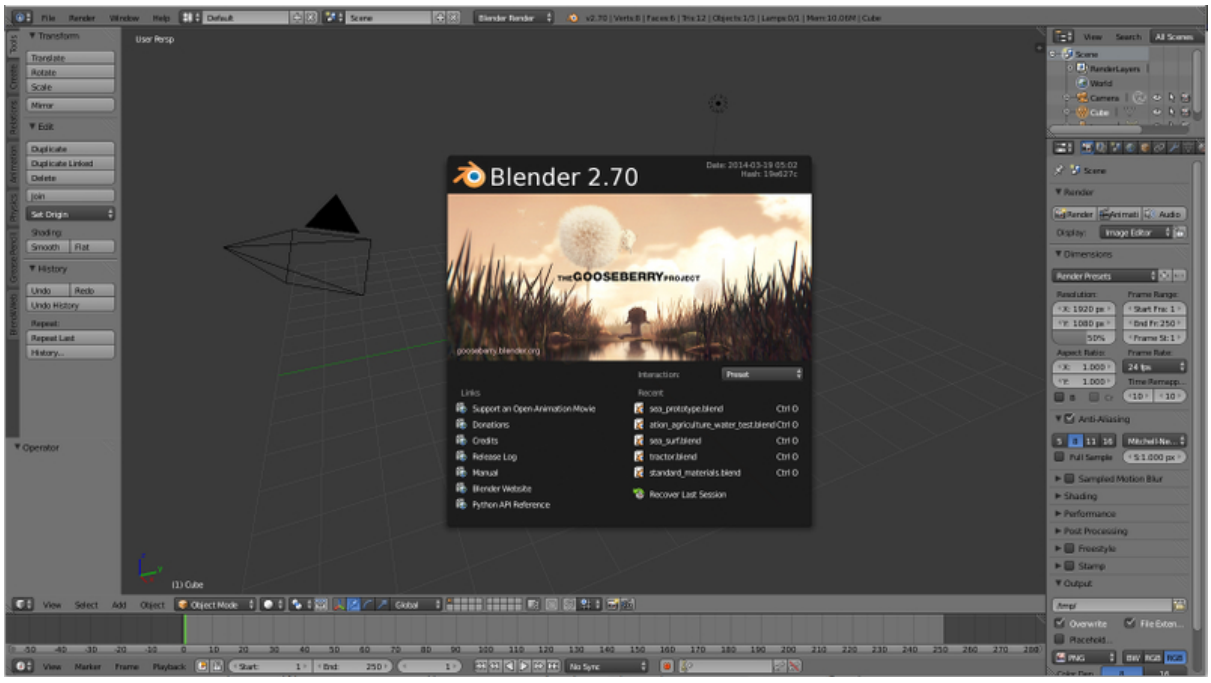
Quick install of the Blend4Web addon suits Blender artists who have no need in full-scale 3D applications development. In this case the main benefit is the opportunity to export a scene into a single HTML file for viewing in WebGL-capable browsers.

For more serious tasks an SDK setup is required.

3.1 Installing Blender

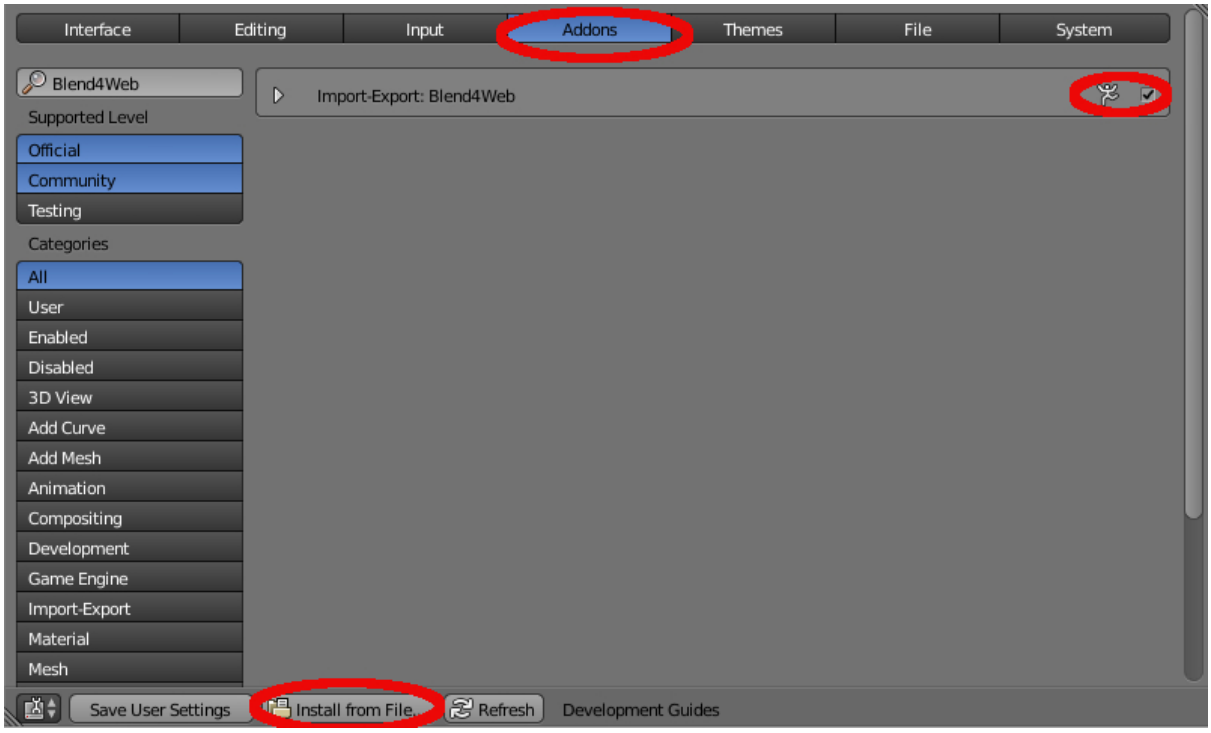
Authoring 3D scenes is carried out directly in Blender which is open source software and is distributed free of charge.

A current stable version of Blender should be used. It can be downloaded from the official site.



3.2 The Engine Addon Install

Run Blender, load the default scene File > New. Open the user preferences File > User Preferences.... Under the Addons tab click Install from File... and then select the zip archive with the addon files. After that turn on the Import-Export: Blend4Web checkbox.



Then click Save User Settings and close the user preferences window.

3.3 Exporting and Viewing Scenes

The created scenes can be exported in HTML format. To do this use the File > Export > Blend4Web (.html) menu option and choose the export filepath. The resulting HTML file can be opened with any browser with WebGL support.

See also:

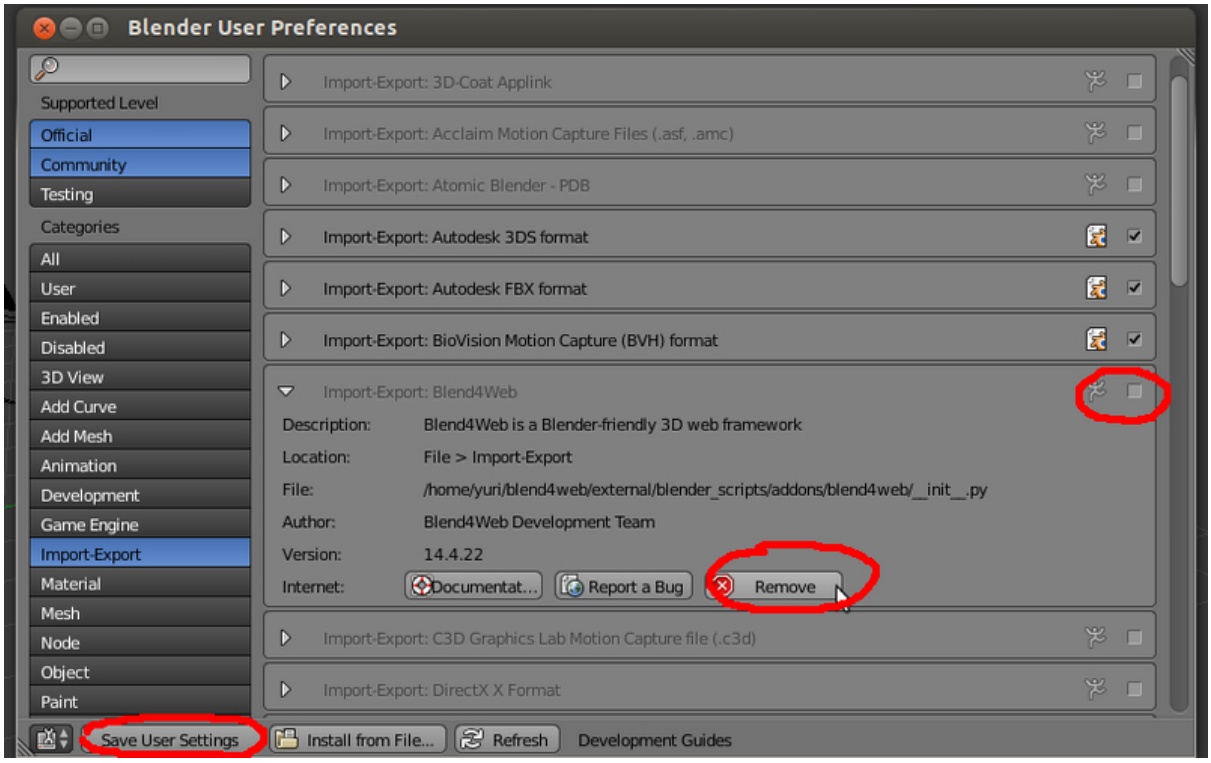
WebGL Browsers Support

3.4 Upgrading the Addon

To upgrade the addon first disable the old version and then remove it.

To disable the addon: run Blender, load the default scene File > New. Open the user preferences File > User Preferences.... Go to the Addons tab and choose the Import-Export category. Disable the Import-Export: Blend4Web checkbox. Then click Save User Settings and restart Blender.

Then to remove the addon open the user preferences window again, expand the Blend4Web info panel and click the Remove button.



CHAPTER 4

Development Environment Setup

This setup option suits 3D application developers. To familiarize yourself with the Blend4Web addon quick install can be a better option.

In order to work you must have the engine distribution, a browser (suitable for local viewing) and Blender (with the addon installed).

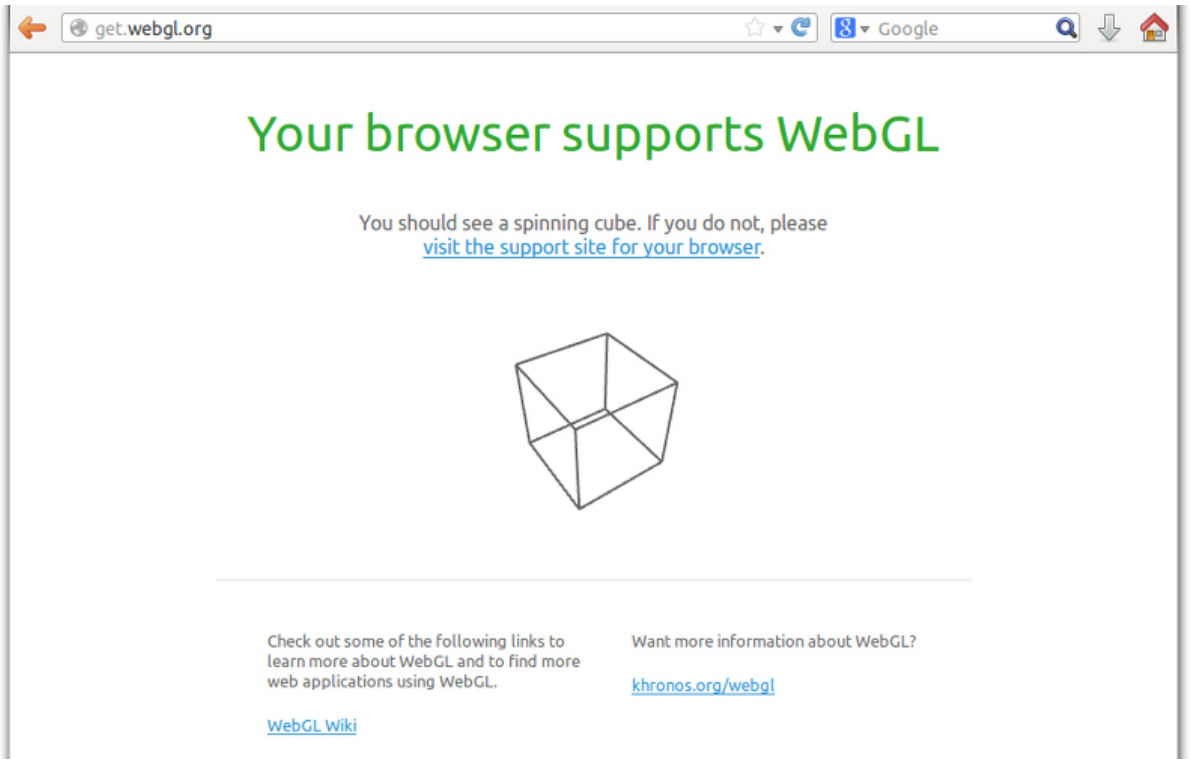
4.1 Distribution Installation

Stable versions of the distribution are available as an archive (blend4web_sdk_free.zip – free SDK, blend4web_sdk_pro.zip – commercial SDK). Simply unpack this archive somewhere.

The distribution consists of the engine's source code, the minified version for applications, the Blender scripts, the source blend-files, the exported scenes, textures and sound files (see detailed repository structure).

4.2 Choosing a Browser

The engine requires a browser with WebGL support. Just to check it open the page http://get.webgl.org/. The green message and rotating cube should appear:



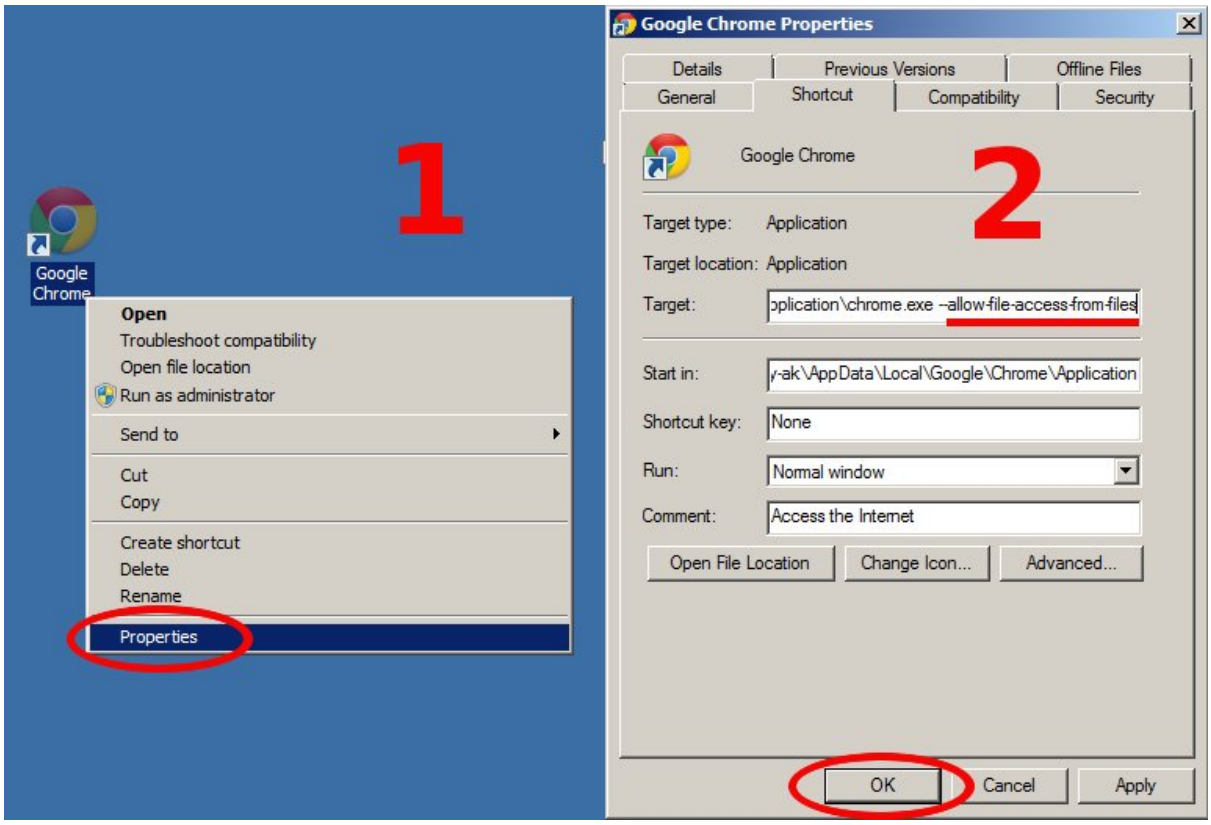
4.3 Setting up the Browser for Loading Local Resources

The engine's renderer is a web application and it works when you view an HTML file in a browser. After initialization the resources (scenes, textures) are loaded. This process is subject to the same-origin policy rule. In particular this rule forbids loading from a local directory. A simple way to bypass this limitation is to set up the browser for loading local resources (recommended). Another way is to use a local web server.

Note: It is recommended to use such browser only for viewing local content because changing the default settings can damage security.

Chrome under Windows:

Right click on the browser shortcut, select the Properties option and add the --allow-file-access-from-files option to the executable filepath (after the space symbol). Click OK.



For convenience create a copy of the shortcut and modify it for local viewing while leaving the original shortcut untouched for normal surfing.

Chrome under OS X:

Open Terminal and run the browser with the following parameter:

> /Applications/Google\ Chrome.app/Contents/MacOS/Google\ Chrome --allow-file-access-from-files

Chrome/Chromium under Linux:

Run the browser with the parameter:

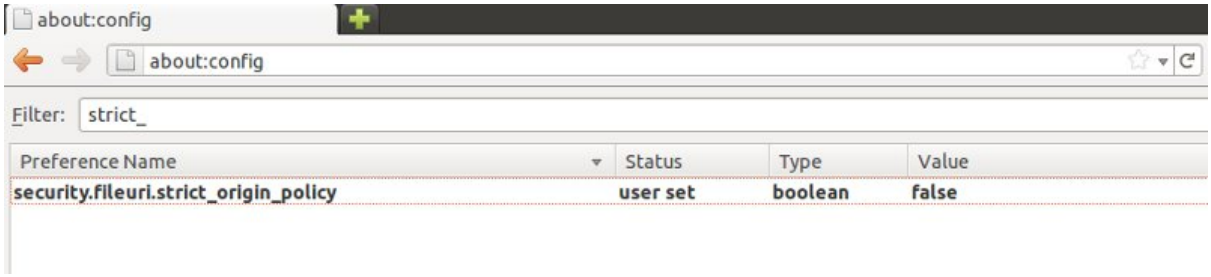
> google-chrome --allow-file-access-from-files

or:

> chromium-browser --allow-file-access-from-files

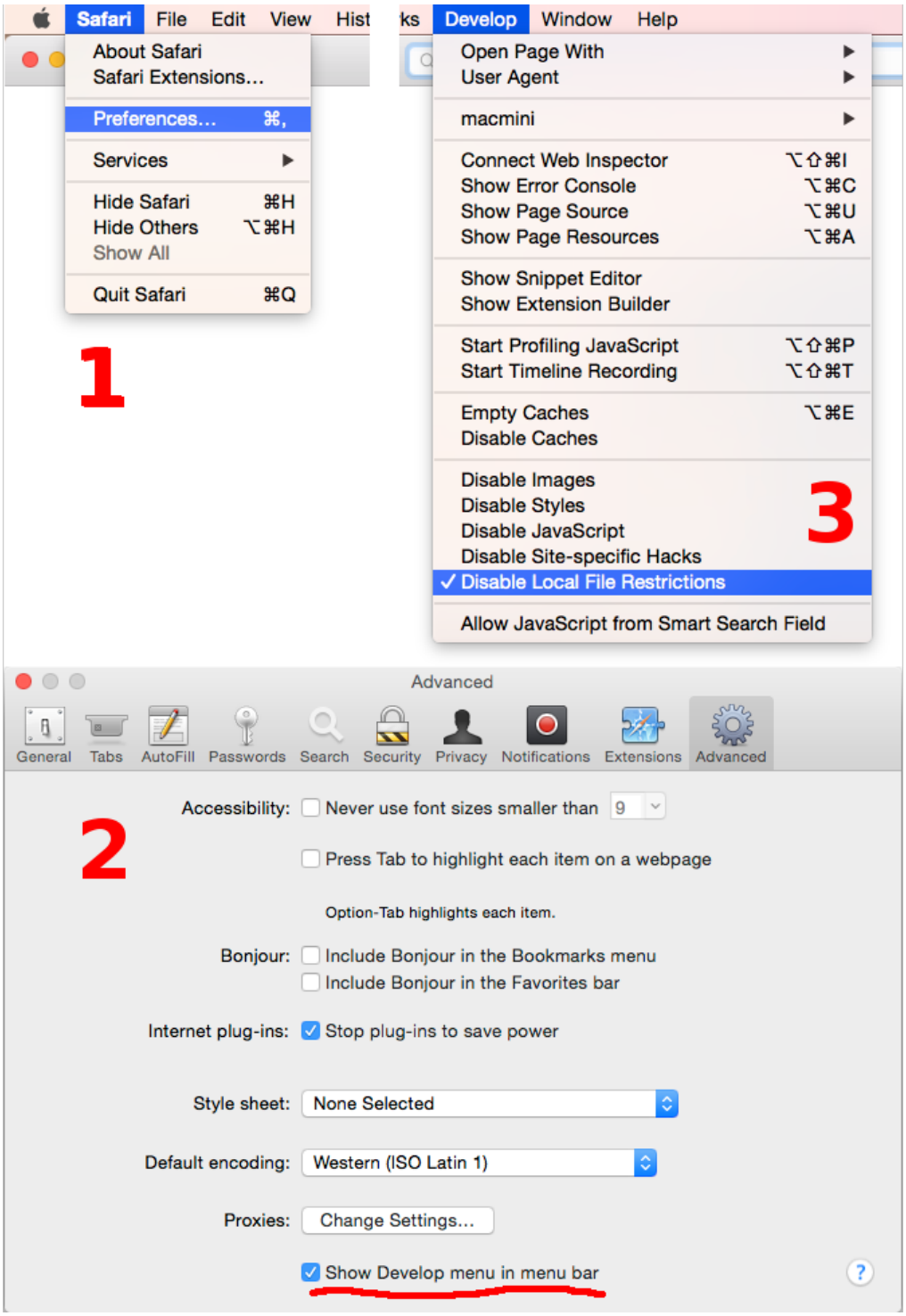
Firefox under Windows/Linux/OS X:

Enter about:config to the browser's address bar, search for the security.fileuri.strict_origin_policy parameter and double-click on it to switch from true to false.



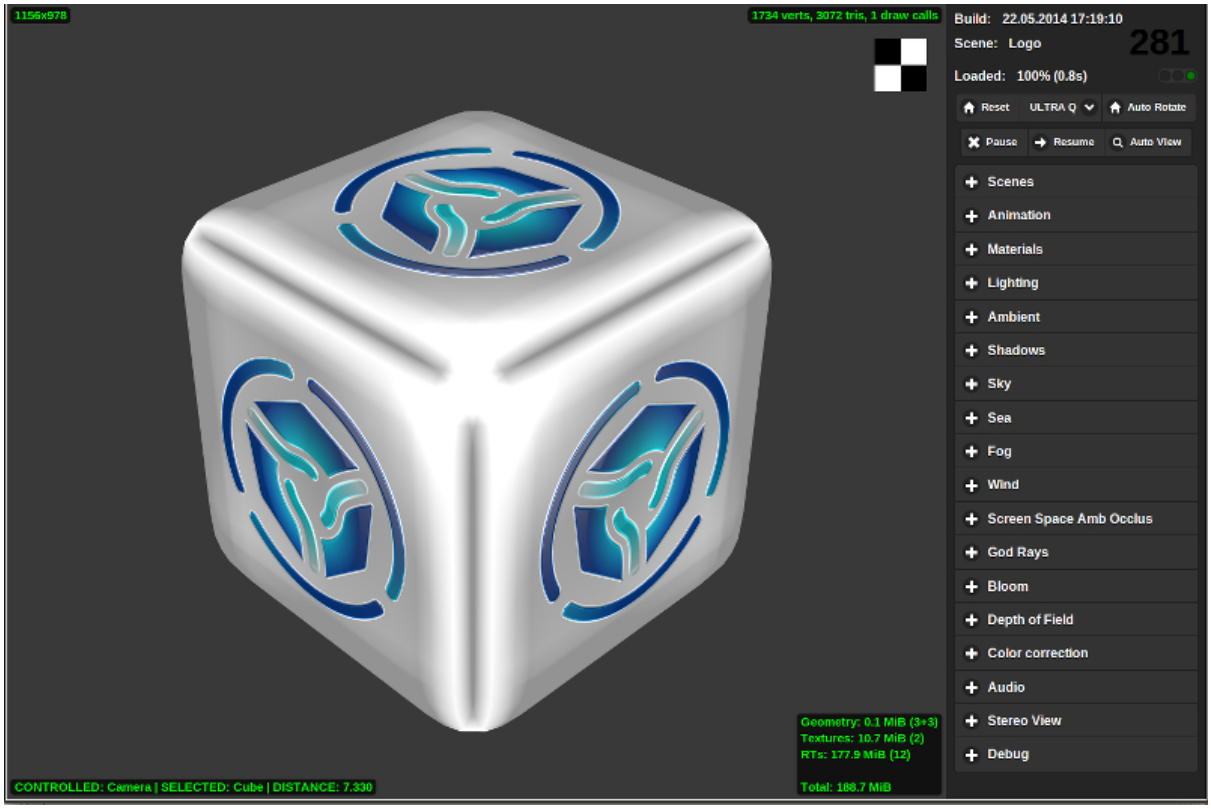
Safari/OS X:

Enable the "Develop" menu in Preferences, and activate the "Disable Local File Restrictions" option.


Safari
File
Edit
View
Hist
About Safari
Safari Extensions...
Preferences... ⌘,
Services
Hide Safari ⌘H
Hide Others ⌥⌘H
Show All
Quit Safari ⌘Q
1
ks
Develop
Window
Help
Open Page With
User Agent
macmini
Connect Web Inspector ⌥⇧⌘I
Show Error Console ⌥⌘C
Show Page Source ⌥⌘U
Show Page Resources ⌥⌘A
Show Snippet Editor
Show Extension Builder
Start Profiling JavaScript ⌥⇧⌘P
Start Timeline Recording ⌥⇧⌘T
Empty Caches ⌥⌘E
Disable Caches
Disable Images
Disable Styles
Disable JavaScript
Disable Site-specific Hacks
✓ Disable Local File Restrictions
3
Allow JavaScript from Smart Search Field
Advanced
General
Tabs
AutoFill
Passwords
Search
Security
Privacy
Notifications
Extensions
Advanced
2
Accessibility:
Never use font sizes smaller than 9
Press Tab to highlight each item on a webpage
Option-Tab highlights each item.
Bonjour:
Include Bonjour in the Bookmarks menu
Include Bonjour in the Favorites bar
Internet plug-ins:
Stop plug-ins to save power
Style sheet:
None Selected
Default encoding:
Western (ISO Latin 1)
Proxies:
Change Settings...
Show Develop menu in menu bar
?

4.4 Running The Scenes Viewer

Open the apps_dev/viewer/viewer_dev.html file in a set up browser. The page with the renderer window and interface elements should be displayed.

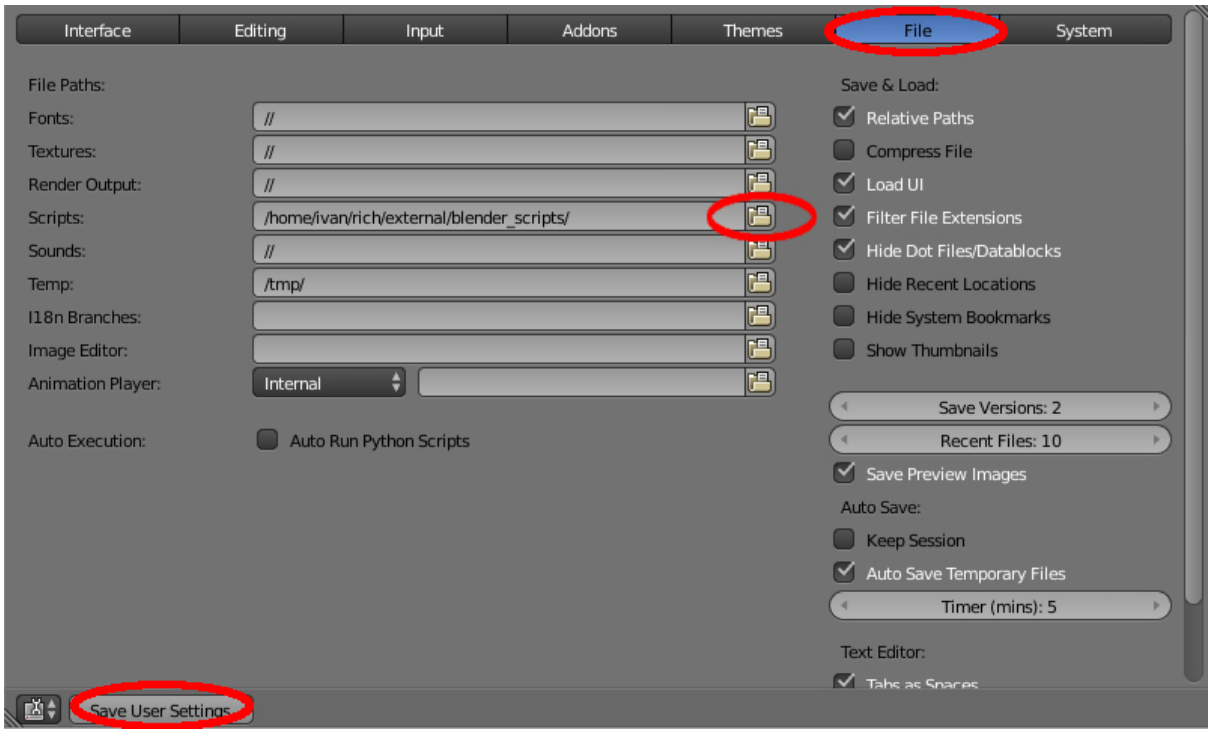


Note: If the page isn't displayed correctly, or an error message is shown, follow the instructions described in the Problems when Running the Renderer section.

4.5 Engine Addon Installation

Note: It is recommended to remove the addon first if it was originally installed using quick install.

Run Blender, load the default scene File > New (hot keys Ctrl-N). Open the user preferences window File > User Preferences... (hot keys Ctrl-Alt-U). Under the File tab in the Scripts field choose the path to the blender_scripts directory.

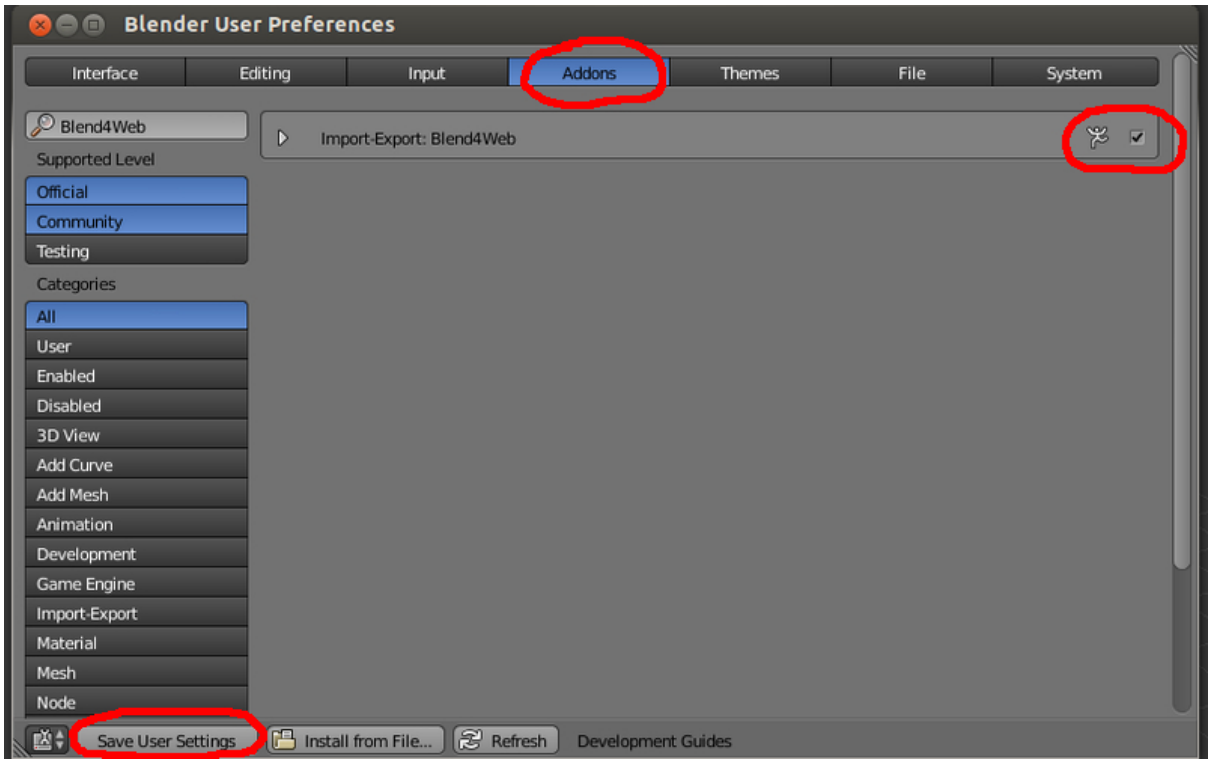


Click Save User Settings and restart Blender.

Note: Instead of this it's possible to copy the scripts directory blender_scripts/addons/blend4web to the already used directory for scripts or even to the installation directory, for example:

C:\Program Files\Blender Foundation\Blender\2.70\scripts\addons\blend4web.

Again load the default scene, open the user preferences window, go to the Addons tab and choose the Import-Export category. Enable the Import-Export: Blend4Web checkbox.



Click Save User Settings. Restarting Blender isn't required.

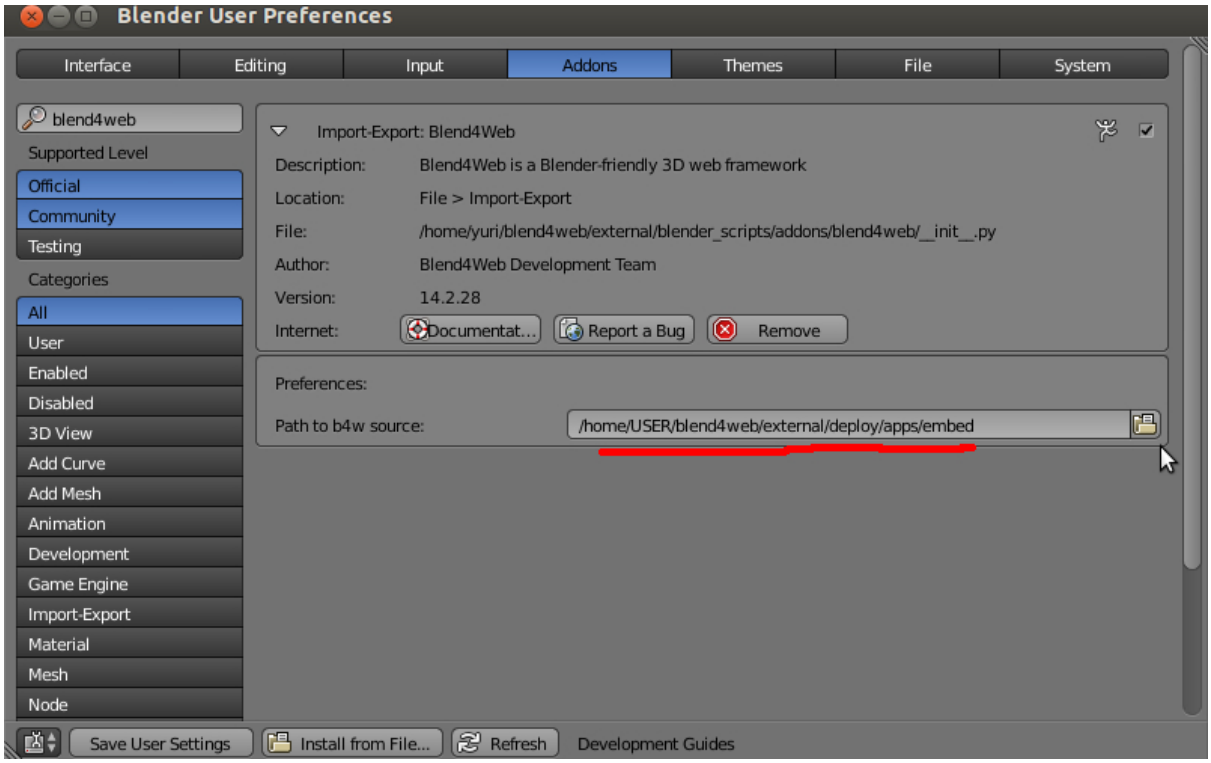
For checking:

In the File > Export menu the Blend4Web (.json) and Blend4Web (.html) options should appear. Also the operators should appear in the search box when searching for "B4W" (hot key SPACE).

4.5.1 Enabling the HTML export option

The Blend4Web (.html) option in the File > Export menu is not active by default contrary to the standalone version of the addon (see Quick Install).

If required (for example for HTML export debugging) this option can be enabled. To do this specify the path to the embed application build of the distribution in the Path to b4w source field. The standard path relative to the engine's root is external/deploy/apps/embed.


Blender User Preferences
Interface
Editing
Input
Addons
Themes
File
System
blend4web
Supported Level
Official
Community
Testing
Categories
All
User
Enabled
Disabled
3D View
Add Curve
Add Mesh
Animation
Development
Game Engine
Import-Export
Material
Mesh
Node
Import-Export: Blend4Web
Description: Blend4Web is a Blender-friendly 3D web framework
Location: File > Import-Export
File: /home/yuri/blend4web/external/blender_scripts/addons/blend4web/__init__.py
Author: Blend4Web Development Team
Version: 14.2.28
Internet: Documentat... Report a Bug Remove
Preferences:
Path to b4w source: /home/USER/blend4web/external/deploy/apps/embed
Save User Settings
Install from File...
Refresh
Development Guides

CHAPTER 5

Workflow

Developing any product is a creative process with many participants who have different skills and experience. However no matter how complex it is and what is the target it's always possible to separate the production stage in which the bulk of assets and source code is authored.

When using Blend4Web the workflow is the following:

1. Preparing a 3D scene in Blender.
2. Exporting the resources in a format suitable for the engine.
3. Running, tweaking and debugging the scene in the viewer.
4. Creating the target application.

5.1 Preparing the Scenes

Besides the usual stages such as modeling, texturing, animation etc a scene should be prepared for working in the engine.

General recommendations:

1. Blend-files should be located in the external/blender/project_name directory.
2. Texture and sound files should be external and located in the external/deploy/assets/project_name directory.
3. Auxiliary files which are not intended for loading into the engine (for example, references), should be located in the external/blender/project_name directory.
4. The file from which export will occur should only contain models required for the application being developed.
5. Object, mesh, material, texture, armature etc should have distinct names (in English). They should not be named "Cube.001", "Material", "Armature".
6. Its possible to link components from other (library) files.

5.2 Exporting Scenes

In order to load scenes authored in Blender into the engine you have to transform them into the format suitable for reading by a browser. At the moment text files with .json extension are used in which exported data structures are saved in the JSON (JavaScript Object Notation) format. This file, in turn, refers to one binary file with a .bin extension containing data arrays of models and to external resources - textures and sound samples.

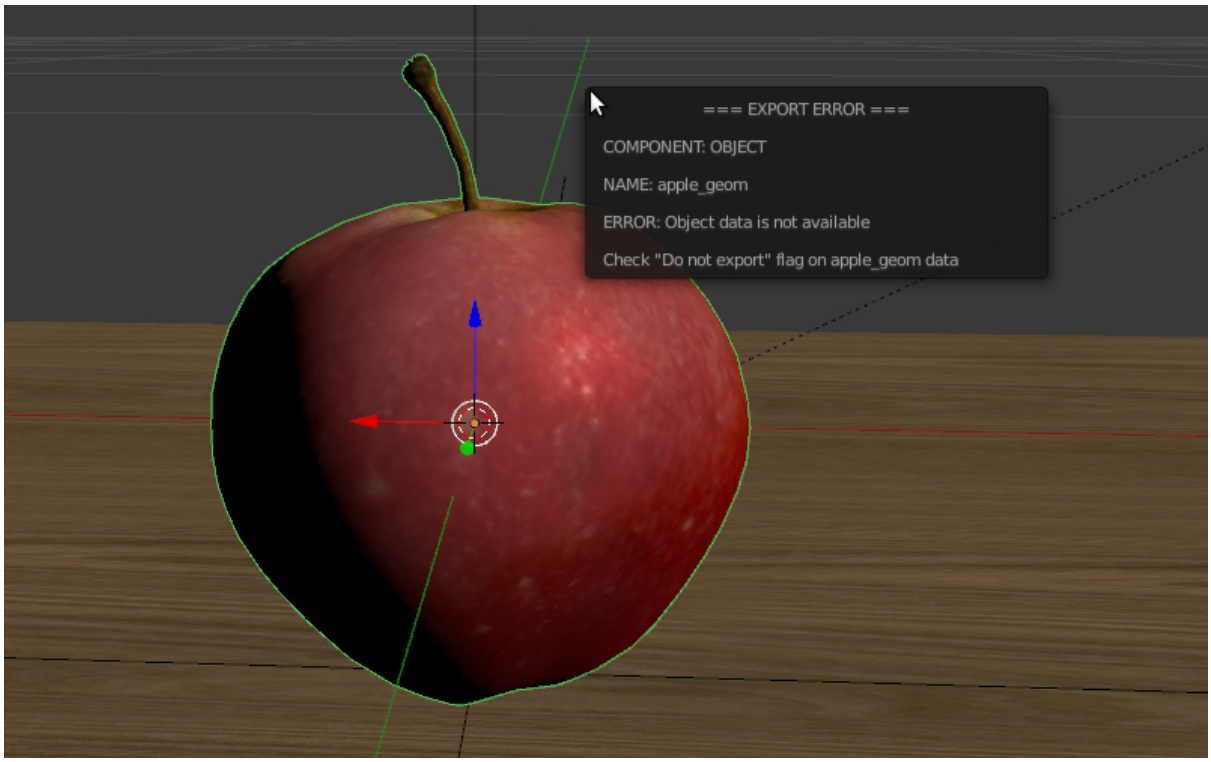
While the .json and .bin files are created upon export, texture and sound files are normally placed by hand (there is an exception though: resources embeded into a .blend-file are placed automatically).

Export can be performed by choosing the Blend4Web (.json) option from the File > Export menu. For quick access search for b4w export (hot key SPACE).

It is recommended to place files intended for export into the directory intended for application deployment, for example external/deploy/assets/project_name.

It's necessary to use relative filepaths for images (normally this is by default). If this is not the case, execute the File > External Data > Make All Paths Relative operator. Using absolute filepaths instead of relative ones may lead to errors when loading .blend and .json files on other computers.

Upon export the scene is checked for Blender features not supported by the engine. In such cases an error message is generated:



The list of possible export errors is specified in the corresponding section.

5.2.1 Export Options

Autosave main file Autosaving the file from which export occurs. Enabled by default. Autosaving happens right after the export to guarantee conformity between the current blend-file and the exported file content. In addition, the relative path to the exported file is saved for convenience.

Filepath Path to the exported file. It is used in the command-line export mode and not used in windowed mode.

5.3 Displaying Scenes in the Viewer

For the scene to appear in the viewer list it's required to manually add an entry to the text file external/deploy/assets/assets.json.

To add a new scene you need to know the category in which it should be displayed. The category normally corresponds to the project name and to the name of the directory where the corresponding files are stored.

5.3.1 Example

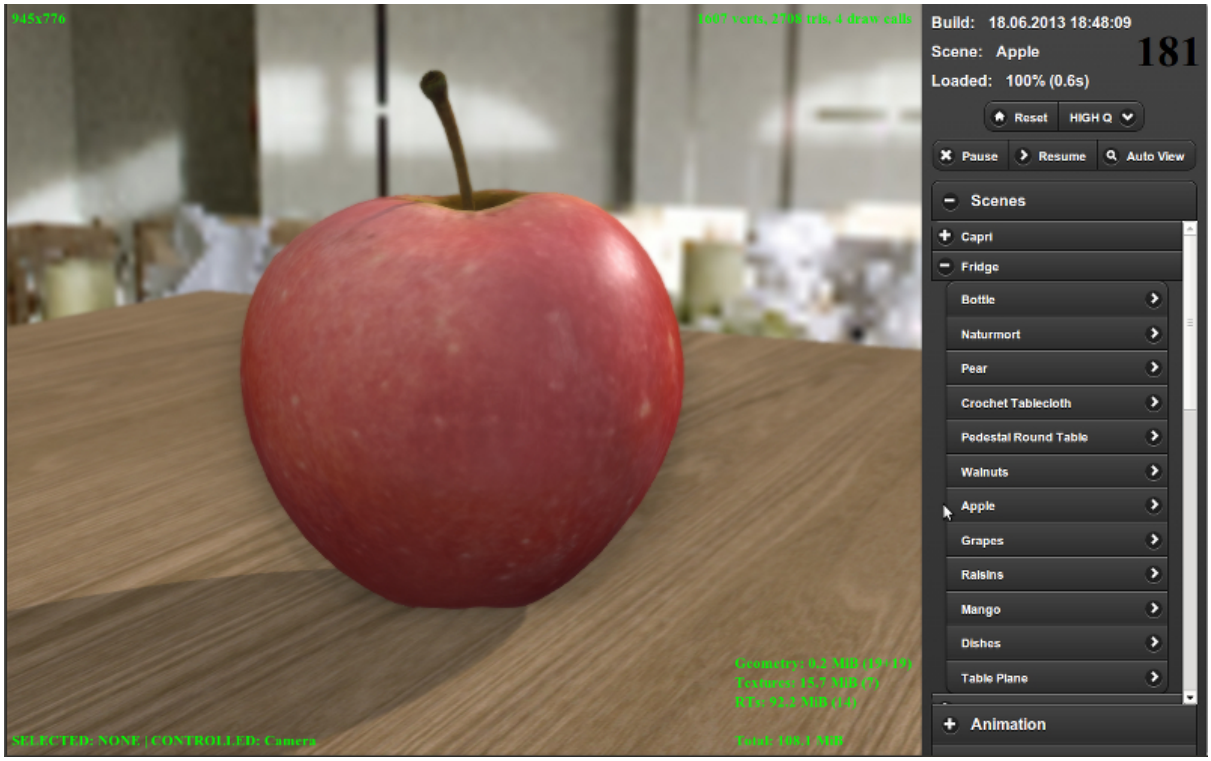
For example below you can see a part of assets.json. In this file there are two projects - "Capri" and "Fridge" each with corresponding scenes:

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

```
            name: "Mango",
            load_file : "fridge/fruits/mango/mango.json"
        }
    ]
}
```

To add a new scene you can copy and paste a similar scene's description to the required category and then edit its name and path to the exported file.

A successfully added scene should appear in the scenes' list of the viewer in the required category.



5.4 Application Development

At this stage an application is created. Logic for scene loading and user interaction is written using JavaScript. The application developer notes are given in the corresponding section.

CHAPTER 6

Scene Viewer

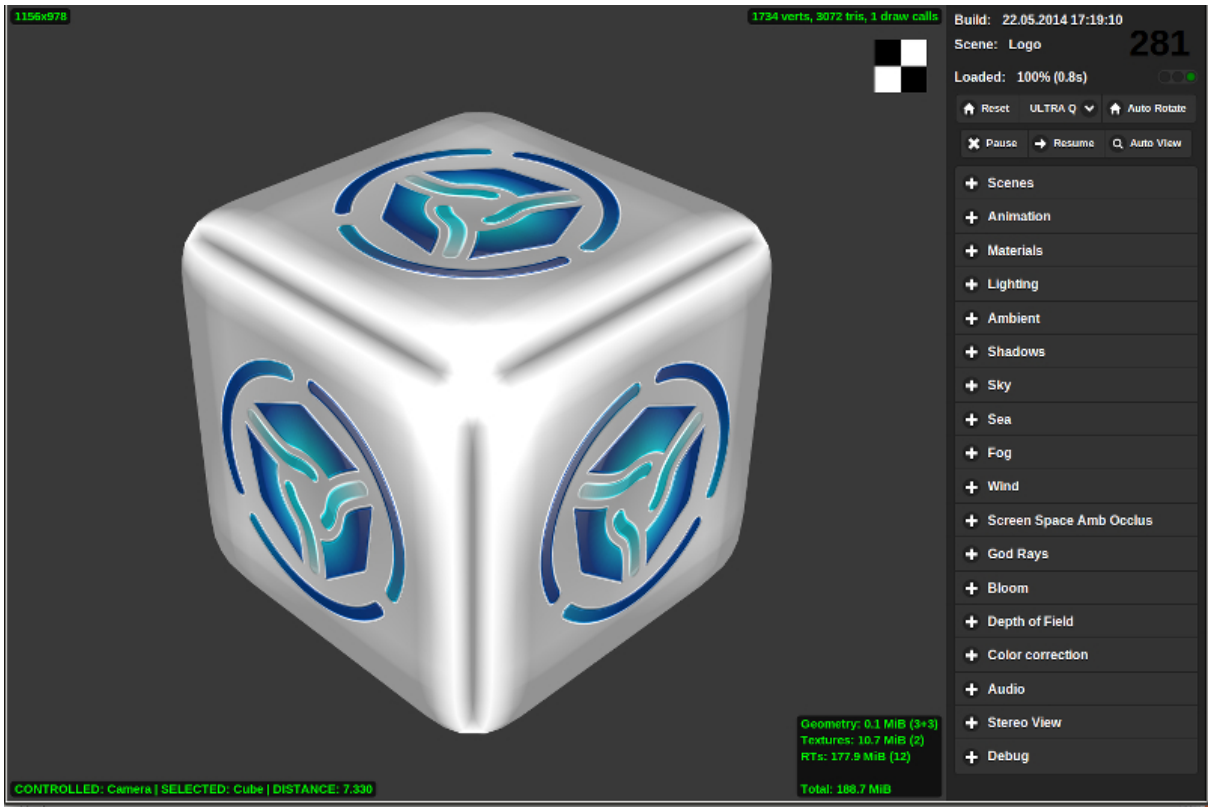
Running The Scenes Viewer.

6.1 Navigation

To control the camera hold down a mouse button and move the mouse. Also control can be performed using the W, A, S, D, R, F keys: forward, left, back, right, up, down. Arrows and numpad keys can be used as well. In the Target camera mode it's possible to focus on the selected object using the Z or .(dot) keys.

6.2 The Side Panel

The side panel consists of three areas: the information board, basic control buttons and the list of drop-down panels with additional control elements differentiated by functionality.



6.2.1 Control elements list in top-to-bottom order

Build The engine build date and time. In the developer version this shows the page load time.

Scene Loaded scene name from the assets.json file. Path to the file pops-up on mouse hover.

Loaded Loading progress and time.

Reset This button deletes the saved name of the last viewed scene and reloads the page back to display the default scene.

LOW Q - HIGH Q - ULTRA Q Drop-down menu for choosing the performance profile of the engine.

See also:

Quality Profiles

Pause Pause rendering.

Resume Resume rendering.

Auto View Activate the automatic scene switching mode; the delay between views is 1 second.

Scenes A double-level list of the categories and scenes from the assets.json file.

Animation Animation controls. When viewing animated models it's possible: to select an object and switch its animation with a drop-down menu, switch cyclic animation mode, stop and resume animation, set the required frame (the animation should be stopped).

Materials Material properties setup. A material can be selected using the drop-down menu. At the moment only a limited range of properties is supported.

Lighting Direct lighting parameters setup. A light source can be selected using the drop-down menu. Changing color and intensity is supported. Daytime and sun lighting parameters can also be tweaked on this panel.

Ambient Ambient lighting parameters setup. Changing the colors and intensity of a hemispheric ambient model is supported.

Shadows Shadow parameters setup, including shadow cascades and shadow edges softening parameters.

Sky Dynamic sky parameters setup such as color, sun light scattering parameters etc.

Sea Water rendering parameters setup, including color transitions by depth and by shore distance, foam and subsurface scattering parameters, waves dynamics etc.

Fog Fog parameters setup, including density and color.

Wind Wind parameters setup, including direction and strength.

Screen Space Amb Occlus Ambient occlusion parameters setup.

God Rays Crepuscular rays effect parameters setup.

Bloom Bright light effect parameters setup.

Depth of Field Depth of field effect parameters setup.

Color correction Color correction parameters setup, including brightness, contrast, exposure and saturation.

Anti-aliasing Selecting the anti-aliasing method.

Audio There is a mixing mode switch on the panel. After it is enabled the mixer interface becomes visible (only for scenes with sound sources).

Stereo View There is a stereoscopy mode switch on the panel.

Debug This panel contains a range of debugging tools, including the wireframe mode and the postprocessing stages viewer switches.

6.3 Indicators

Frames per second counter This is located in the top right corner. It displays the averaged and rounded value for the last 1.5 seconds.

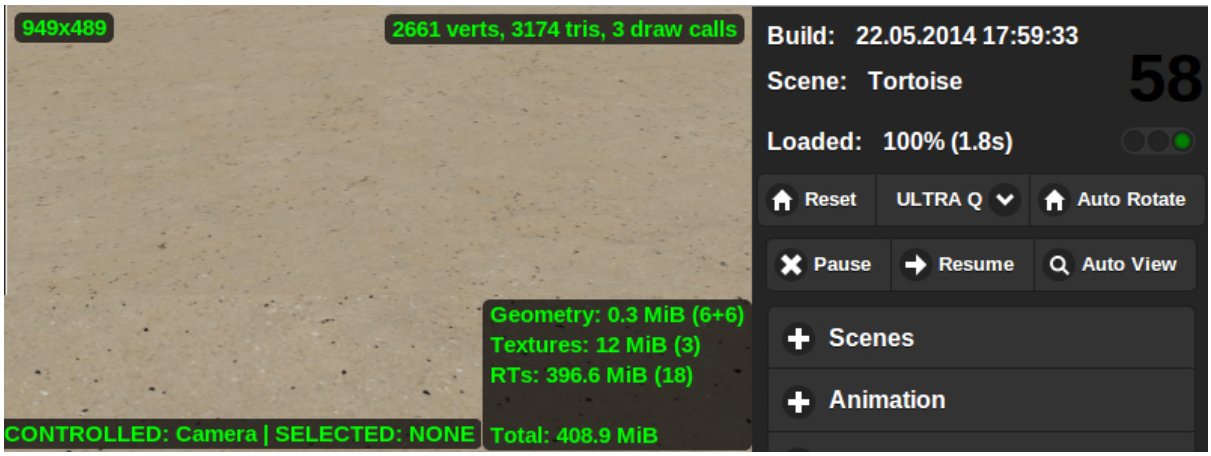
Viewport dimensions This is located in the top left corner. It displays the viewport dimensions in pixels.

Selected object and controlled object This is located in the left bottom corner. It displays the names of selected and controlled objects. Object selection can be performed with the mouse. To control the object directly (normally for physics debugging) press the Q key and click on the object. The object movement is performed with the W, A, S, D keys. To exit the control mode press the Q key and click on an empty space. The indicator also displays the distance to the selected object in Blender units (meters equivalent).

Scene complexity indicator Is located in the top right corner of the rendering area. It displays the number of vertices, triangles and WebGL calls on the main rendering scene (i.e. shadow rendering calls are not included, for example).

Video memory indicator Is located in the bottom right corner of the rendering area. It displays the amount of video memory used by geometry, textures, render targets, and also the total memory usage.

Scene load errors indicator Is located under the FPS counter. Shows errors and warnings which occured during scene load. Red light means errors, yellow - warnings and green means that the scene was loaded successfully.



CHAPTER 7

The Addon

7.1 Initialization Errors

Initialization errors can arise upon installation of the addon or when a scene is opened in Blender. In this case a dialog window with the error description is showed.

Error message	Cause
Blend4Web initialization error! Addon is not compatible with the PLATFORM platform.	The Blend4Web addon is not compatible with the PLATFORM platform.
Warning: Blender version mismatch. Blender VER_REQUIRED is recommended for the Blend4Web addon. Current version is VER_CURRENT.	Warning about possible incompatibility with the current Blender version. It is recommended to use VER_REQUIRED Blender version. The current version is VER_CURRENT.

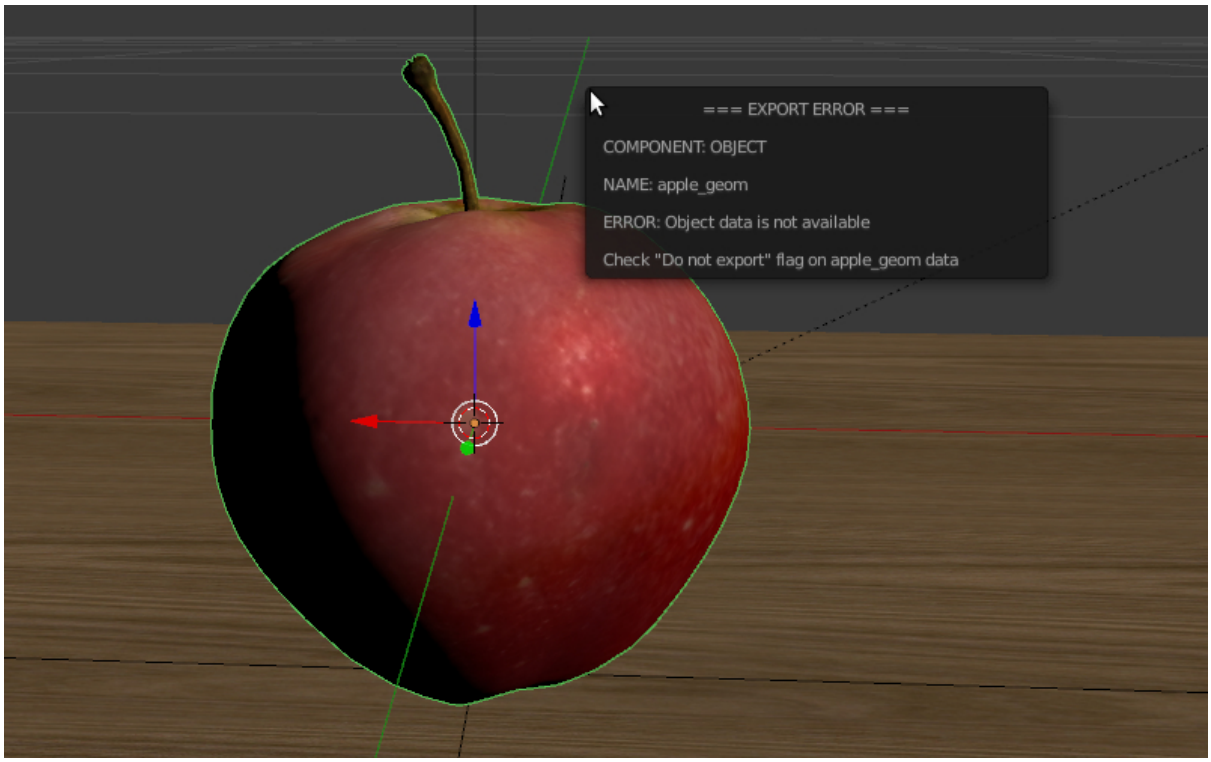
7.2 Export Errors

In case of export errors a BLEND4WEB EXPORT ERROR dialog box describing of the problem appears:

COMPONENT - type of component (object, mesh, material, texture etc) that has caused the export error.

NAME - component name.

ERROR - short description of the occured problem.



Error message	Cause
Dupli group error; Objects from the GROUP_NAME dupli group on the OBJECT_NAME object cannot be exported	None of the objects in the GROUP_NAME group which were selected for duplication on the OBJECT_NAME object can be exported. Permission to export at least one object of the group, or to remove the duplication of the group is required.
Incompatible meshes; Check MESH_NAME1 and MESH_NAME2 UV Maps/ Vertex colors	There are two meshes with the same material. Such meshes are merged automatically into one by the engine with the purpose of optimization, and so they should have the same number of UV Maps and Vertex Colors.
Incompatible objects with a shared mesh; The OBJECT_NAME object has both vertex groups and a shared mesh	Export of an object with both a shared mesh and vertex groups is not allowed. Exceptions: export is possible if an object has the Apply modifiers, Export vertex animation, Export edited normals options turned on (because in these cases a full copying of meshes occurs).
Incomplete mesh; Dynamic grass vertex colors required by material settings	The Dynamic grass size and/or Dynamic grass color options are used by the special terrain material but the mesh has no vertex colors with such names.
Incomplete mesh; Material slot is empty	Material slot is empty.
Incomplete mesh; No UV in mesh with UV-textured material	In the material of the mesh there are textures with texture coordinates type UV, but the mesh lacks UV map layers.
Incomplete mesh; Material settings require vertex colors	The Vertex Color Paint option is enabled for the mesh material, but the mesh has no vertex color layers.
Incomplete vehicle. The NAME vehicle doesn't have any chassis or hull	The modelled NAME vehicle is not complete as it should contain a Chassis or a Hull element.
Incomplete vehicle. The NAME vehicle requires at least one bob	The modelled NAME vehicle is not complete as it should contain at least one Bob element.
Incomplete vehicle. The NAME vehicle requires at least one wheel	The modelled NAME vehicle is not complete as it should contain at least one Wheel element.

Incorrect mesh; Wrong group indices	The mesh has vertices assigned to the non-existing vertex group.
Incorrect vertex animation; Object has no vertex animation	The object's vertex animation export option is on, but there is no vertex animation.
Incorrect vertex animation; Unbaked "ANIM_NAME" vertex animation	Vertex animation export is turned on for the mesh, but the ANIM_NAME animation doesn't have any frames.
The material has a normal map but doesn't have any material nodes	The node material uses Normal Mapping, but has no Material node.
The mesh has a UV map but has no exported material	The mesh has a UV map layer but has no material for export.
The mesh has a vertex color layer but has no exported material	The mesh has a vertex color layer but has no material for export.
Missing active camera	There is no active camera on the scene (Camera property on the Scene tab).
Missing lamp	There should be at least one light source in the scene.
Missing world	There should be at least one world datablock in the scene.
No image	The texture has no image.
No such file or directory	The file or directory does not exist.
No texture in texture slot	There is no texture in the material texture slot.
Node material invalid; Check sockets compatibility: FROM_NODE with TO_NODE	Node material error: the input and output types of the link between the FROM_NODE and TO_NODE nodes should match.
Object constraint has no target	The Target Object property for the object constraint (on the Object Constraints tab) was not set.
Object data not available; Check the "Do not export" flag on the OBJECT_NAME data	The object data is not available. This error appears particularly when the Do not export property is set under the Object Data tab for the object being exported.
Object-parent relation is not supported; Clear the parent's inverse transform	When using parenting it's required to reset transform for the child object using the Object > Parent > Clear Parent Inverse operator (Alt-P).

Only 2 UV textures are allowed for a mesh; The mesh has N UVs	The engine supports up to 2 UV texture layers for each mesh. The number of UV layers for this mesh is N.
Particle system error; Dupli group isn't specified	Particle system error: no group is selected as a particle.
Particle system error; Dupli object isn't specified	Particle system error: no object is selected as a particle.
Particle system error; Dupli object OBJECT_NAME doesn't export	The OBJECT_NAME object which is selected as a particle can not be exported (the Do not export checkbox is set).
Particle system error; The GROUP_NAME dupli group contains no valid object for export	The GROUP_NAME dupli group which is selected as a particle contains no valid object for export. Either such objects have the Do not export checkbox enabled or the types of the objects are unsuitable. Supported object types: MESH.
Particle system error; The "NAME" vertex color specified in the from field is missing in the last of the "OBJECT_NAME" object's vertex colors	The NAME vertex color is specified in the from field but it not present in the OBJECT_NAME emitter.
Particle system error; The "NAME" vertex color specified in the "to field is missing in the "OBJECT_NAME" object ("GROUP_NAME" dupli group)	The NAME vertex color is specified in the to field but it is not present in the OBJECT_NAME object of the GROUP_NAME group which is selected as a particle.
Particle system error; The "NAME" vertex color specified in the to field is missing in the list of the "OBJECT_NAME" object's vertex colors	The NAME vertex color is specified in the to field but it is not present in the OBJECT_NAME object which is selected as a particle.
Particle system error; Wrong dupli object type TYPE_NAME	An object of unsuitable type is selected for the particle. Supported types: MESH.
Permission denied	No access rights to the current directory.
Wind bending: vertex colors weren't properly assigned	Wind bending parameters setup: it's required to specify the names of either all vertex color layers (Main stiffness (A), Leaves stiffness (R), Leaves phase (G), Overall stiffness (B)), or of the main one only (Main stiffness (A)), or of none of them.
Wind bending: not all vertex colors exist	Wind bending parameters setup: all specified vertex color layers should exist.

Wrong edited normals count; It doesn't match with the mesh vertices count	The number of edited normals does not match the number of the mesh vertices. Execute Clean Up or Save in the B4W Vertex Normals Editor panel.
Wrong overridden bounding box; Check the mesh's bounding box values	Wrong dimensions are specified when overriding the mesh's BoundingBox: minimum value is greater than maximum value for at least one of the dimensions.
Wrong texture coordinates type	For textures with images the following coordinate types are supported: UV, Normal.
Wrong vertex animation vertices count; It doesn't match with the mesh vertices count for "ANIM_NAME"	Vertex animation export is enabled but the number of vertices in the baked ANIM_NAME animation frames does not match the mesh vertices number. Possible solution is to "re-bake" the animation.

7.3 Warnings About Export Errors

In contrast to the above-listed export errors, these errors don't prohibit the export, but can make scenes displayed incorrectly. To view these warnings, open the browser console when a scene is loaded. The message looks like this:

EXPORT WARNING: error message

Error message	Cause
The NAME node is not supported. The NAME material will be rendered without nodes.	The engine does not support the node with the this name, and so the node material will be turned off. Often this happens when Cycles nodes are used.

CHAPTER 8

Objects

Objects are intended to position components of different types (meshes, cameras, lamps etc) in a 3D scene space.

8.1 Types

The engine supports objects of the following types:

- mesh
- camera
- lamp
- empty
- curve
- armature
- speaker
- force field

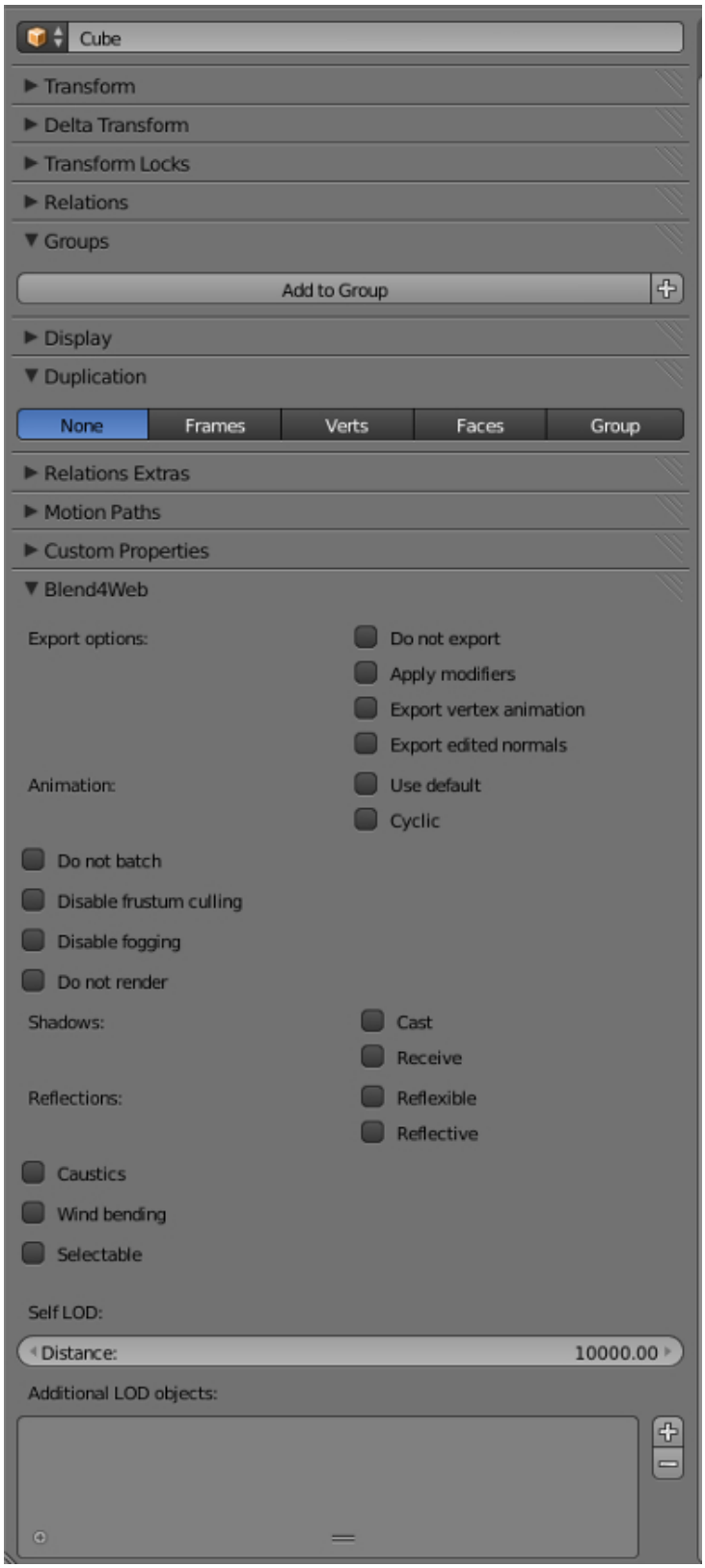
8.2 Settings

The following is supported for all types of objects: transform, data reference, parent object, group membership and a range of the engine's special properties.

Transform > Location Position coordinates.

Transform > Rotation Rotation angles. The XYZ Euler mode should be used (set by default).

Transform > Scale Scaling. All 3 components (x, y, z) should be the same. Scaling for the physics objects is not supported.


Cube
Transform
Delta Transform
Transform Locks
Relations
Groups
Add to Group
Display
Duplication
None
Frames
Verts
Faces
Group
Relations Extras
Motion Paths
Custom Properties
Blend4Web
Export options:
Do not export
Apply modifiers
Export vertex animation
Export edited normals
Animation:
Use default
Cyclic
Do not batch
Disable frustum culling
Disable fogging
Do not render
Shadows:
Cast
Receive
Reflections:
Reflexible
Reflective
Caustics
Wind bending
Selectable
Self LOD:
Distance:
10000.00
Additional LOD objects:

Object Data (tab) Reference to the datablock which is specific for the objects of different types.

Relation > Parent Reference to the parent object.

Blend4Web > Do not export Do not export.

Blend4Web > Apply modifiers Apply the object's modifiers upon export.

Blend4Web > Export vertex animation Export previously created and saved vertex animation.

Blend4Web > Export edited normals Export previously edited and saved normals.

Blend4Web > Animation > Use default Upon loading into the engine start playback of the animation assigned to the object.

Blend4Web > Animation > Cyclic Cyclically repeat the animation assigned to the object. Animation cycling is also applied to particle systems and speakers (if they are present)

Blend4Web > Detect collisions Activate the object's physics.

Blend4Web > Character Enable the object as a physics character.

Blend4Web > Vehicle part Enable the object as a part of a vehicle.

Blend4Web > Do not batch Force the object to become a dynamic object.

Blend4Web > Disable frustum culling Disable frustum culling optimization.

Blend4Web > Disable fogging Disable fog for the object.

Blend4Web > Do not render Disable object rendering (for example useful for a physics object).

Blend4Web > Shadows: Cast и Blend4Web > Shadows: Receive Cast and receive shadows respectively. Can be enabled simultaneously.

Blend4Web > Reflections: Reflexible When enabled the object is reflected in the dynamic mirror surfaces.

Blend4Web > Reflections: Reflective When enabled the object surface reflects other objects.

Blend4Web > Reflections: Reflection plane Text field for the name of the empty object defining the reflection plane.

Blend4Web > Wind bending Enable procedural animation under the influence of wind.

Blend4Web > Self LOD > Distance Distance from the camera at which the object is no longer rendered.

Deprecated since version 14.06: implemented in the standard Levels of Detail Blender's tool.

Blend4Web > Additional LOD objects The interface for adding low-poly objects used for switching the levels of detail.

Deprecated since version 14.06: implemented in the standard Levels of Detail Blender's tool.

8.3 Static and Dynamic Objects

All MESH objects can be divided into static and dynamic objects.

Static objects are objects, the meshes of which can be merged together if they have the same material.

Dynamic objects are objects, the meshes of which cannot be combined with each other.

Static objects are merged in order to optimize the number of draw calls. Dynamic objects are needed to make the movement of a separate object possible.

Among objects of the other type the dynamic ones are CAMERA and ARMATURE. All other objects are static.

The objects which have animation, physics or a parent, which is a dynamic object, are considered dynamic by default. When object movement is required but is not apparent from its settings, then its nesessary to enable the Blend4Web > Do not batch checkbox.

8.4 Camera

The camera settings are specified in the Properties panel under the Object Data tab.

Blend4Web > Move style – camera control mode. By default the camera is in Static mode and this can be changed only through the API call. In the Target mode the camera is rotating around a fixed point. The Eye mode allows rotation and translation as in first person view.

Blend4Web > Target location – available in the Target mode. This is the position of the camera pivot point. The Copy Cursor Location button copies the current 3D cursor position into this value.

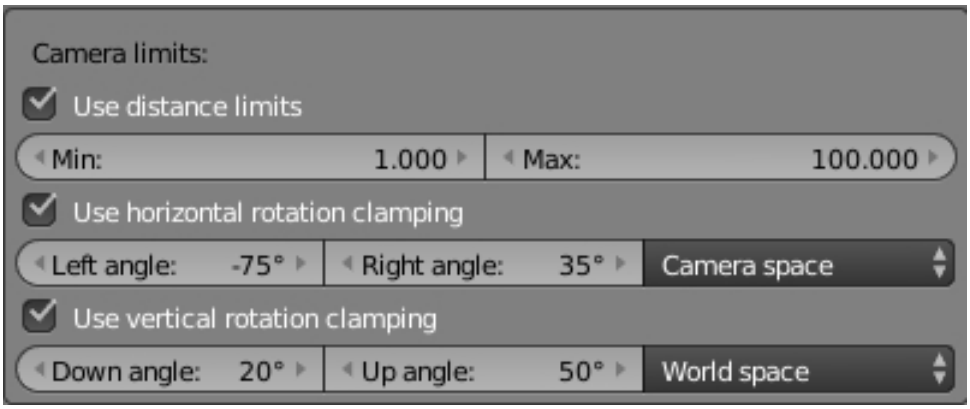
Blend4Web > DOF front distance – described in the Postprocessing Effects section.

Blend4Web > DOF rear distance – described in the Postprocessing Effects section.

Blend4Web > DOF power – described in the Postprocessing Effects section.

8.4.1 Limiting the camera movement

There are several settings for the camera which limit its movement this way or another. They are grouped as Camera limits.

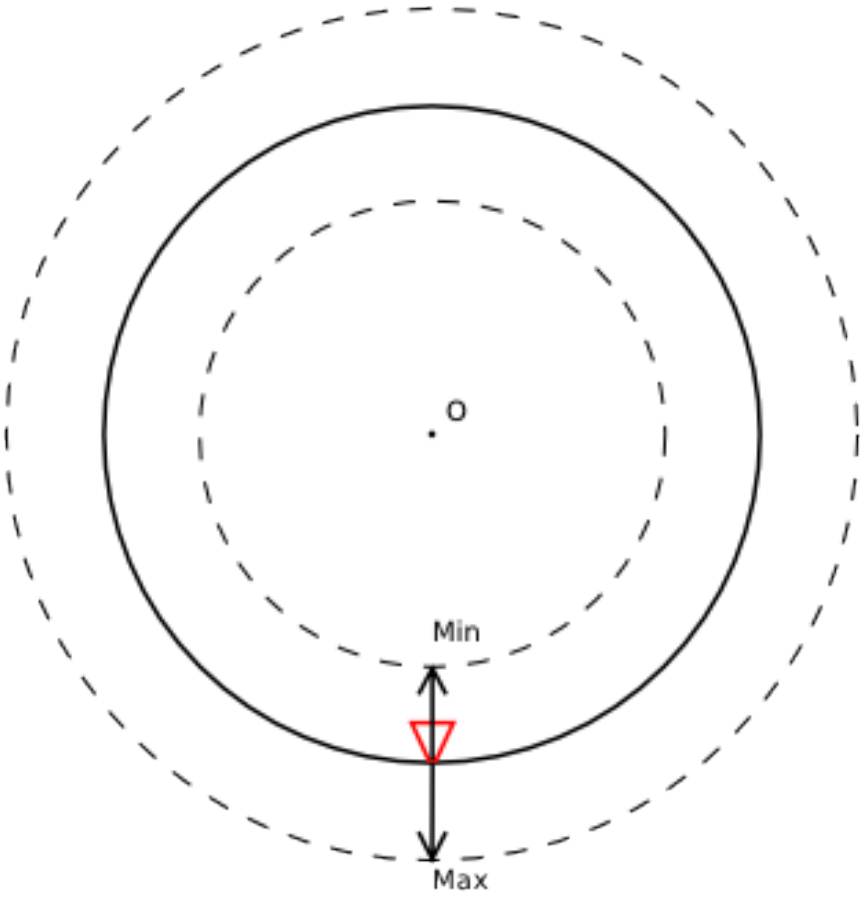


Types of limits

Blend4Web > Use distance limits – available in the Target mode. This defines the minimum and maximum distance from the camera to the pivot point.

Interval variants:

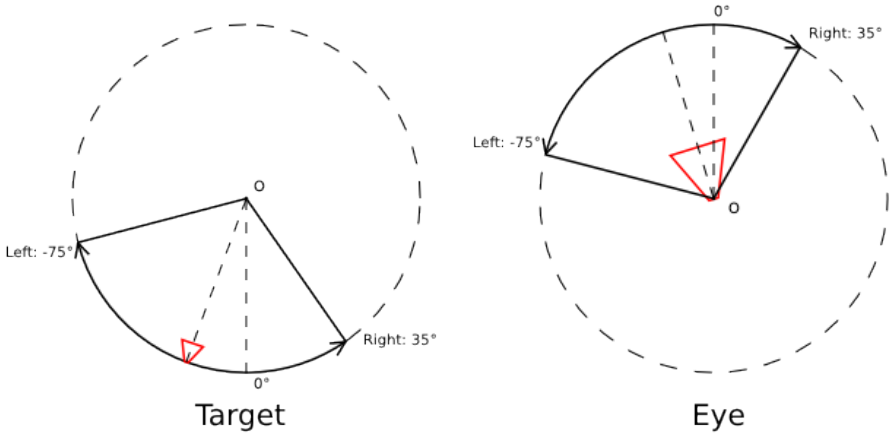
- Max > Min - the distance from the camera to the object is limited by the interval [Min, Max]
- Max = Min - the camera is fixed on a certain height above the object
- Max < Min - no movement limits



Default values: Min = 1, Max = 100.

Blend4Web > Use horizontal rotation clamping – available in the Target and Eye modes. Limits the camera's horizontal rotation relative to the pivot point (in the Target mode) or relative to its position (in the Eye mode).

The direction from Left to Right is considered positive and matches the counterclockwise direction for the Target mode, and the clockwise direction for the Eye mode:



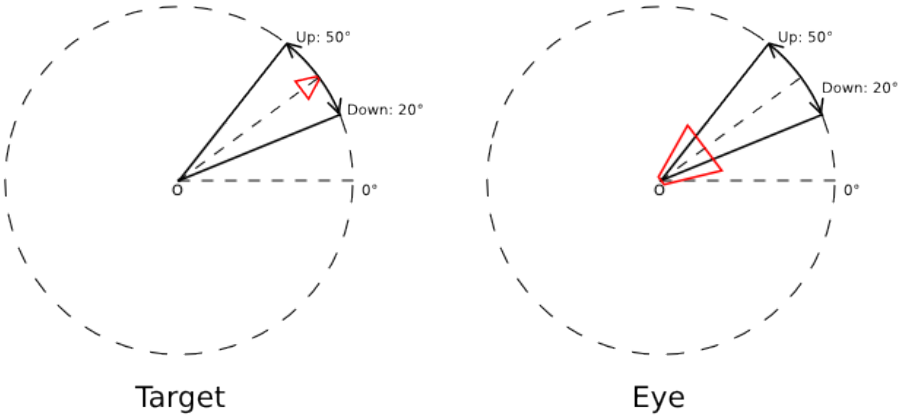
Default values: Left = -180, Right = 180.

Blend4Web > Use vertical rotation clamping – available in the Target and Eye modes. Limits the camera's vertical rotation relative to the pivot point (in the Target mode) or relative to its position (in the Eye mode).

The direction from Down to Up is considered positive:

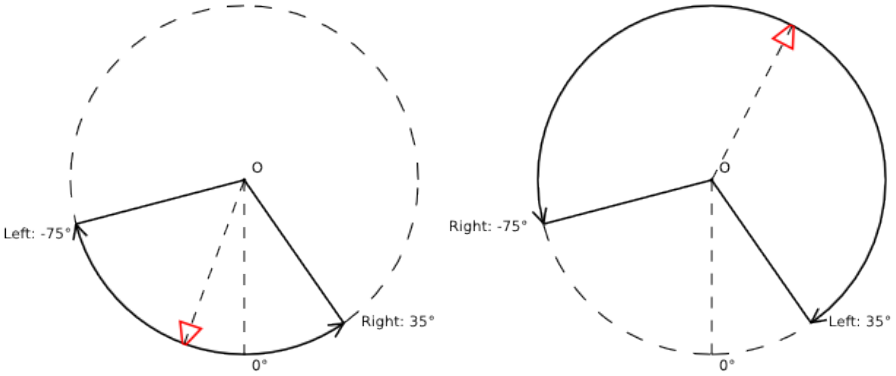
If the Use horizontal rotation clamping checkbox is enabled, vertical rotation will be limited at least to the interval [-90, 90].

Default values: Down = -90, Up = 90.



Possible rotation limits pitfalls

- Swapping the Left/Right or Down/Up values swaps the rotations arcs.



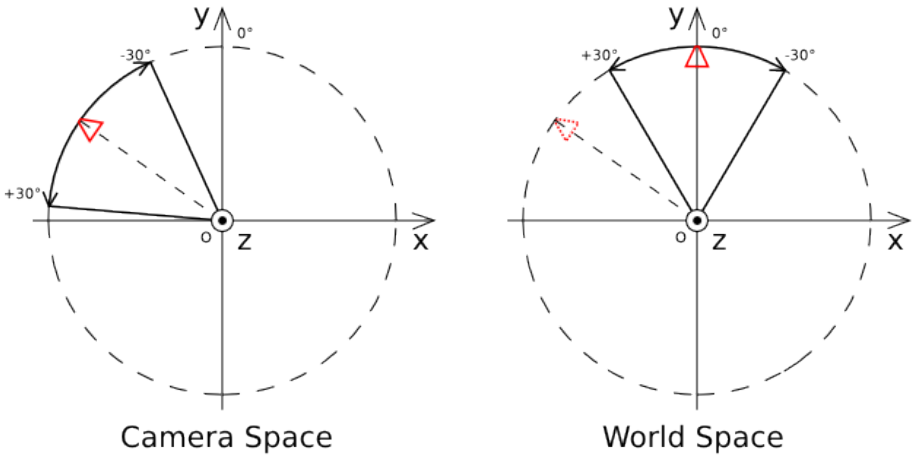
- Left = Right, Up = Down - fixes the camera horizontally and vertically (respectively).

Rotation angles origin

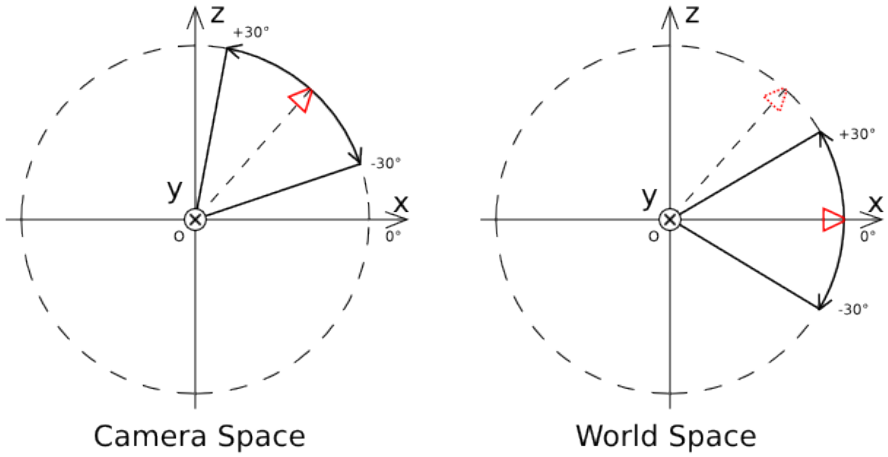
You can choose the space of coordinates for horizontal and vertical rotation limits:

- Camera space - all angles are measured relative to the initial camera position and orientation.
- World space - horizontal angles are measured starting from the Y axis in the world space, while vertical angles - relative to the horizontal plane XOY in the world space.

For horizontal limits:



For vertical limits:



The coordinate axes labelled on the pictures match the Blender's world coordinate axes. By default the Camera space option is selected.

CHAPTER 9

Textures

9.1 Texture Types

The Type drop-down menu (for selecting texture type) is located under the Textures tab. The engine supports the following texture types:

1. Image or Movie
    - diffuse map
    - specular map, this can also be packed into the alpha channel of a diffuse texture
    - normal map
    - height map; this must be packed into the alpha channel of a normal map; it is used for relief surfaces visualization (parallax mapping).
    - transparency map (alpha map) - is used separately only for water rendering at low quality mode; in a generic material this is contained in the alpha channel of a diffuse texture
    - stencil map
2. Environment Map
    - mirror map
    - skydome texture
3. None
    - applied to the Blender's default scene cube. A gray texture is generated in the engine. It is also used for rendering a scene to texture.
4. Blend, gradient
    - is used in particle systems
5. Voronoi procedural texture type
    - is used for water rendering to setup the caustics effect

9.2 Generic Settings

Dimensions Bitmap dimensions for image textures (image width and height in pixels) should be a 2^N number, i.e. 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096 px. Using textures with other dimensions (so-called NPOT) is supported but is not recommended. Dimensions should be at least 4 pixels for the correct texture compression. Normally square images are used (e.g. 512 x 512 px), however rectangular ones can be used too (e.g. 4 x 128 px). Using images bigger than 2048 px is not recommended.

Image Mapping > Extension Texture coordinates interpretation mode (Wrap Mode in WebGL). This is available for Image or Movie texture type. In case of Repeat value the engine sets the REPEAT mode for the texture. In this case the integer part of the texture coordinates is ignored and the fractional part is used. In all other cases (for example Extend) the engine sets the CLAMP_TO_EDGE mode. In this case the texture coordinates are limited by the [0, 1] segment. The default value is Repeat.

Mapping > Coordinates Texture coordinates type. Supported types are UV (use UV map), Normal (use direction at the camera; available only for diffuse maps; used for the creation of material capture, matcap). The default value is Generated (!).

Mapping > Offset Unsupported.

Mapping > Size Scaling the UV map along respective axes. The default values are 1.0.

Blend4Web > Do not export Do not export the texture. Disabled by default.

Blend4Web > Anisotropic Filtering Anisotropic filtering factor for the individual texture. It has priority over the similar parameter for the scene. The default value is DEFAULT (i.e. use the scene settings).

Blend4Web > UV translation velocity Animation speed of texture coordinates along respective axes. The default values are 0.0.

Blend4Web > Water Foam The foam texture. Used by the water rendering material.

9.3 Diffuse Map

A diffuse map is used for specifying scattered light distribution (the Lambert model).

9.3.1 Activation

Enable the Diffuse > Color checkbox on the Textures > Influence panel.

9.3.2 Additional settings

Influence > Diffuse > Color Influence of the texture on the diffuse color. The default value is 1.0.

Influence > Blend The type of the interaction with the material color (Material > Diffuse > Color), or with the vertex color if the Vertex Color Paint checkbox is enabled. The following types are supported: Mix (mixes with the color), Multiply (multiplies by the color). The default value is Mix.

9.4 Specular Map

The specular map is used for specifying the reflected light color distribution (the Phong model).

9.4.1 Activation

Enable the Specular > Color checkbox on the Textures > Influence panel (the Specular > Intensity checkbox is not supported).

9.4.2 Additional settings

Influence > Specular > Color The influence of the texture on the reflected light color. The default value is 1.0.

Influence > Blend The type of interaction with the reflected light color of the material (Material > Specular > Color). Mix (mixes with the color) is the only supported type. The default value is Mix.

The specular map can be packed to the alpha channel of a diffuse texture for optimization purposes. In such case it is required for the texture to enable the Diffuse > Color and Specular > Color checkboxes simultaneously. The color range is limited by gray tints.

9.5 Normal Map

A normal map is used for specifying the distribution of surface normals (perpendiculars) with the purpose of the relief detalization. The information about the normals should be stored in the texture space of coordinates. Normal maps baked in the object space of coordinates are not supported.

9.5.1 Activation

Enable the Geometry > Normal checkbox on the Textures > Influence panel.

9.5.2 Additional settings

Influence > Geometry > Normal Normal map influence on the resulting normals calculation. The default value is 1.0.

9.6 Height Map. Parallax Mapping

A height map contains information about the distribution of relative relief heights. The higher the surface level is, the brighter is its color. A height map combined with a normal map is required for the implementation of relief surface effect (parallax mapping). A height map should be present in the alpha channel of a normal map.

9.6.1 Activation

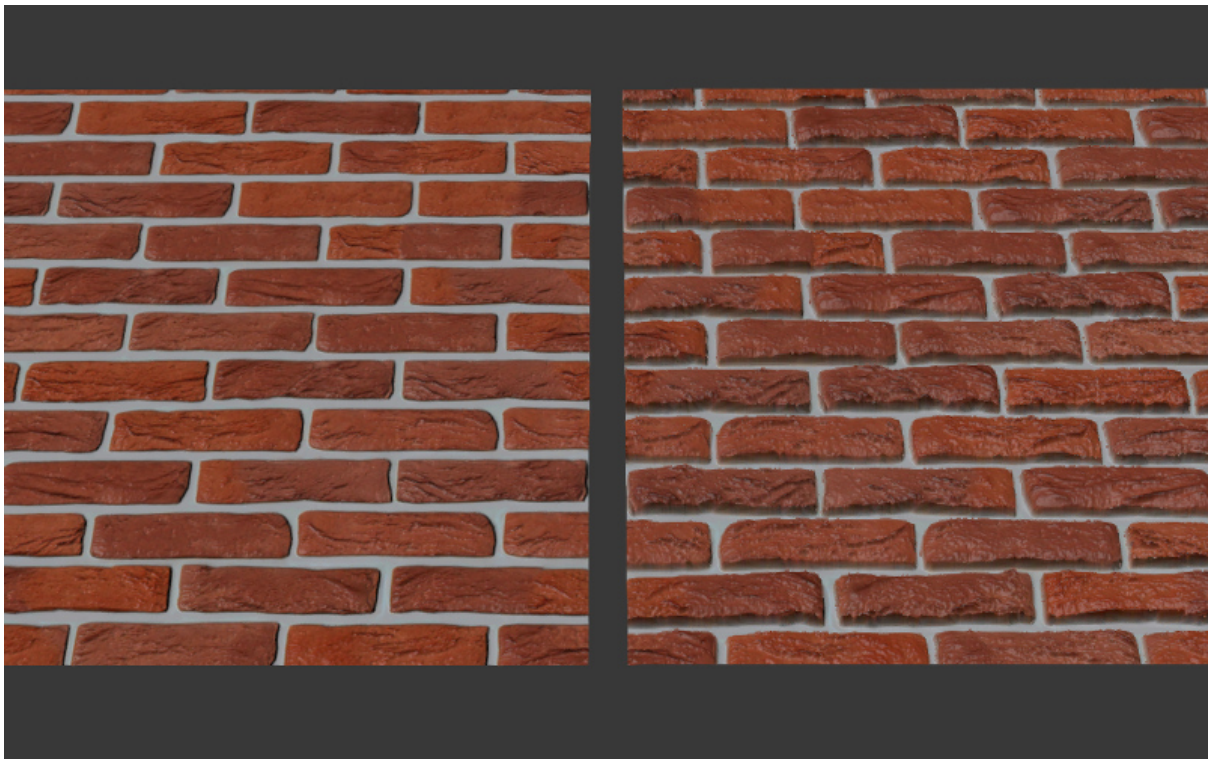
For the normal map enable the Parallax checkbox on the Textures > Blend4Web panel in addition to the Geometry > Normal checkbox on the Textures > Influence panel.

9.6.2 Additional settings

Blend4Web > Parallax Scale Influence factor for the relief surface effect. The default value is 0.03.

Blend4Web > Parallax Steps The number of iterations for the relief surface calculations. Bigger value leads to better quality but is more computationaly expensive.

Blend4Web > Parallax LOD distance Distance at which the parallax effect is observed.



9.7 Alpha Map

The separate alpha map is used only for low quality water rendering. In a generic material it can be present in the alpha channel of a diffuse texture.

9.7.1 Activation

For the diffuse texture enable the Diffuse > Alpha checkbox on the Textures > Influence panel in addition to the Diffuse > Color checkbox. For the separate alpha map enable the Diffuse > Alpha checkbox.

9.7.2 Additional settings

Influence > Diffuse > Alpha Unsupported.

Influence > Blend Unsupported.

9.8 Stencil Map

The special purpose texture (colorful or grayscale) contains information about the distribution of other texture surfaces.

9.8.1 Activation

1. In case of node materials a stencil map should be used in the corresponding node structure.
2. In case of generic materials a stencil map should be located in a texture slot between two mixed diffuse textures. A stencil map requires to set both the RGB to Intensity and the Stencil checkboxes on the Textures > Influence panel.

9.8.2 Additional settings

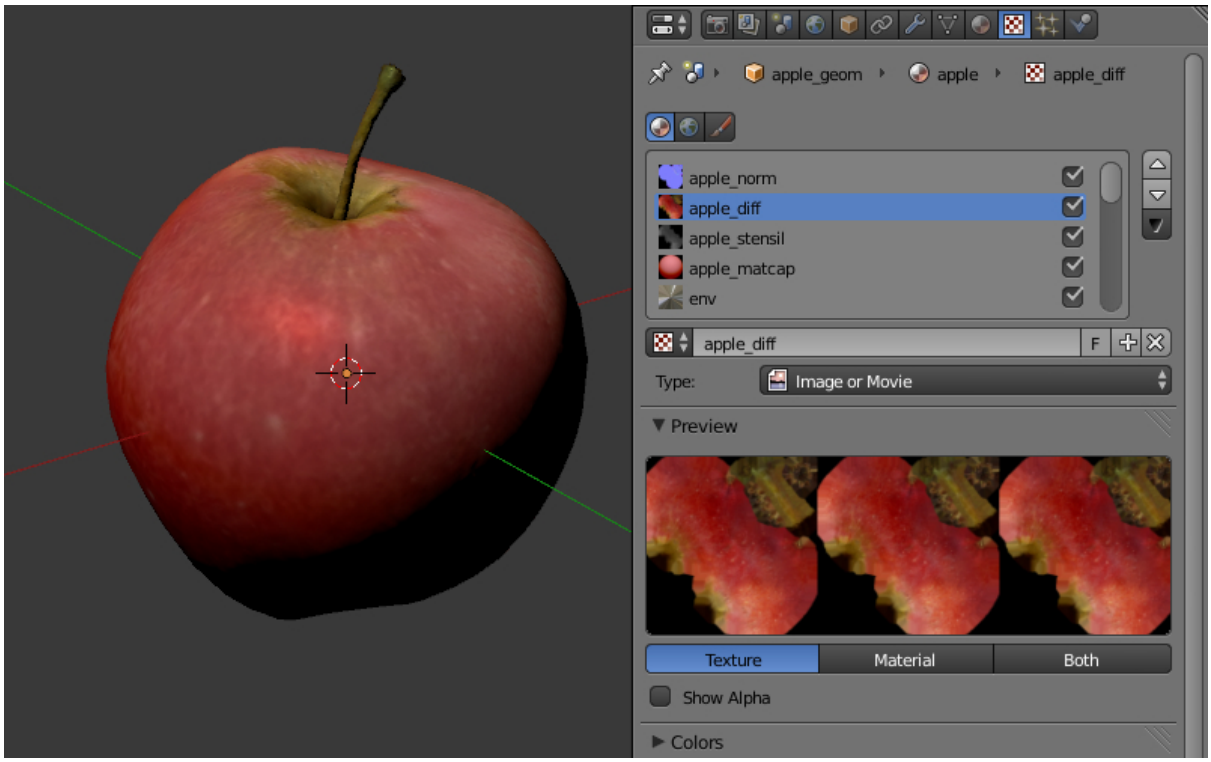
In the case of generic materials one of the mixed diffuse textures can have the Normal ("matcap") texture coordinates type.

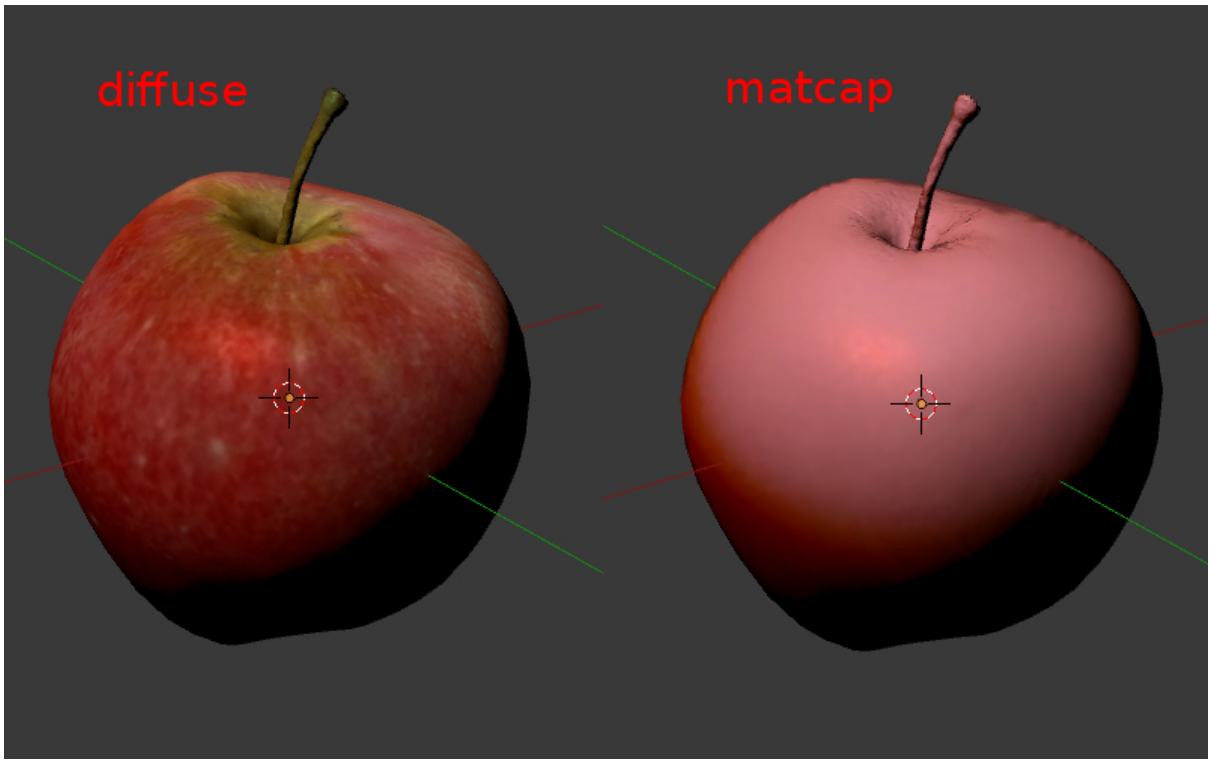
9.8.3 Limitations

In case of generic materials the engine only interprets the red channel of a stencil map. Specular maps or normal maps (if any) are not being mixed. The Mapping > Size setting is extracted from the first texture and is applied to all remaining textures.

9.8.4 Example

The apple model material has the following textures: a normal map, a diffuse texture with a specular map in its alpha channel, a stencil map, a diffuse "matcap" map, an environment map.



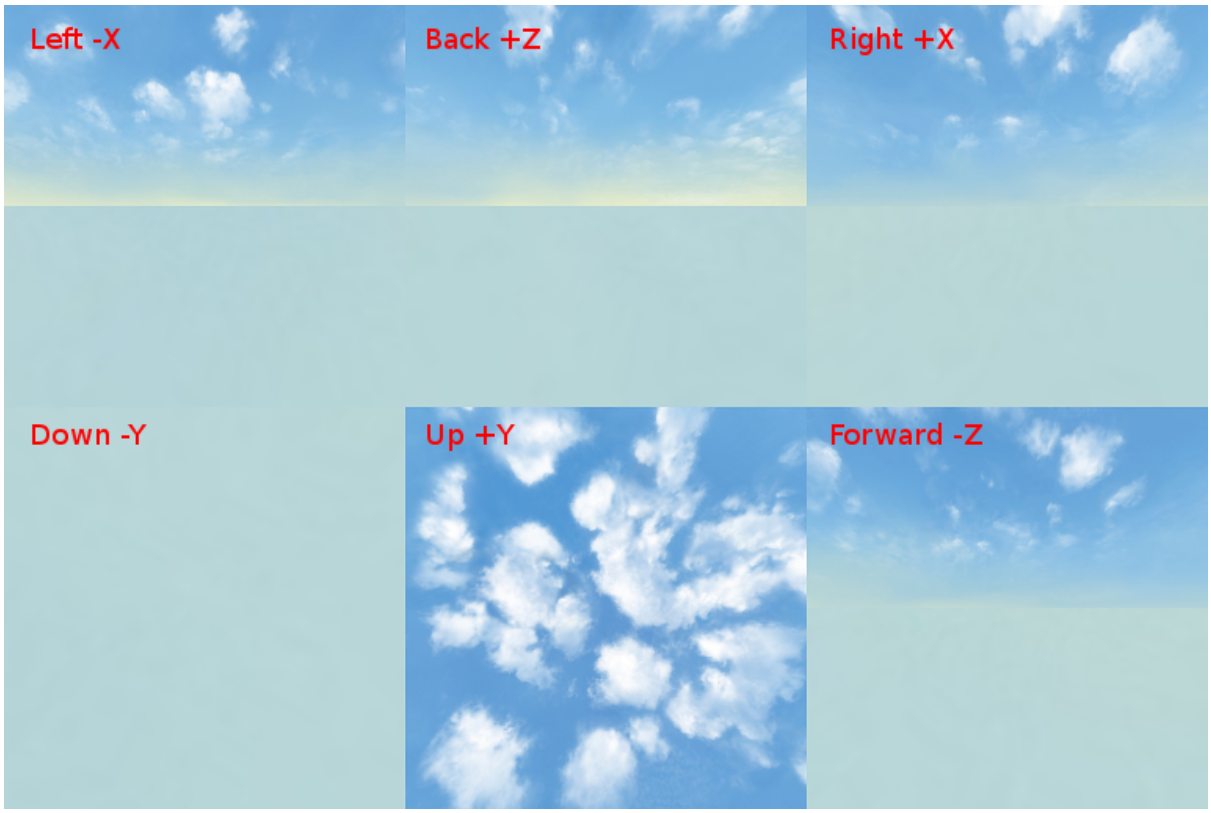


9.9 Environment Map

An environment map is used as a mirror map and as a static sky texture (skydome).

The engine considers it as a cube texture. Environment map bitmaps should contain 6 projected environment images, packed in 2 rows 3 pieces in each (a Blender format). Bitmap dimensions for each image should follow the 2^N rule (512, 1024 etc).

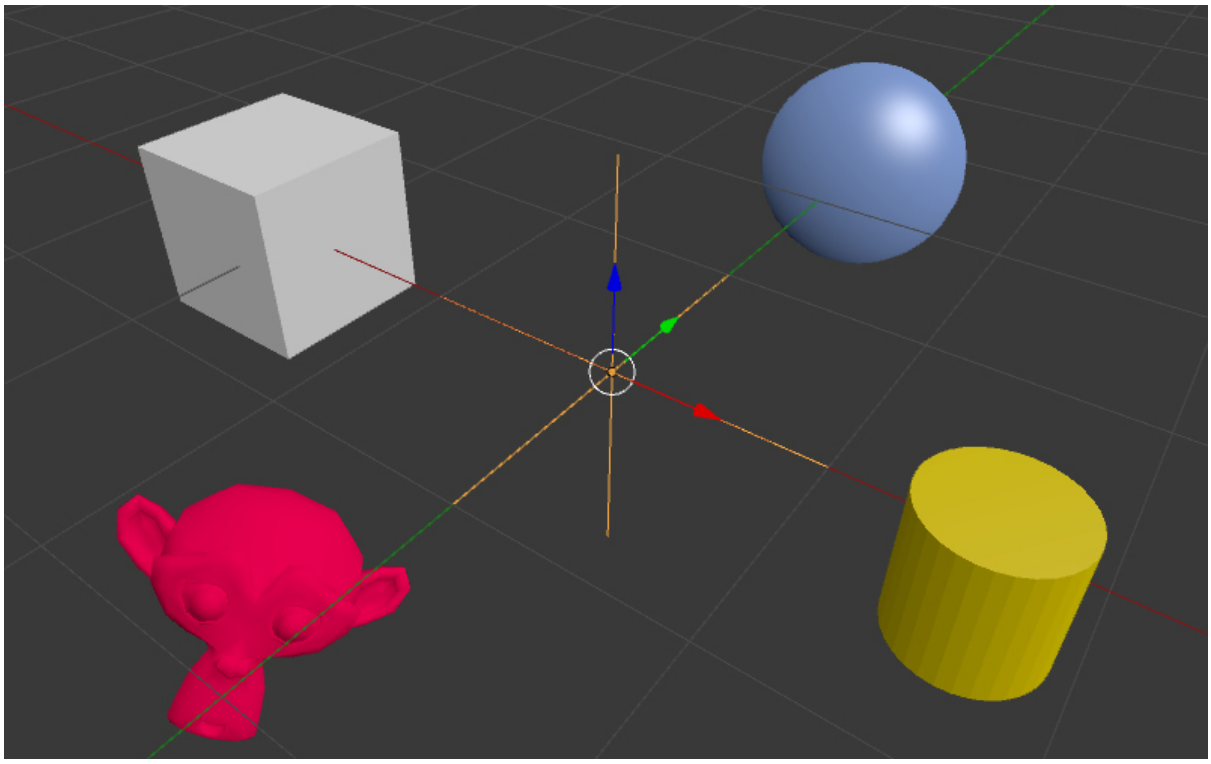
It is recommended to use the lossless format (PNG) in order to avoid seams.

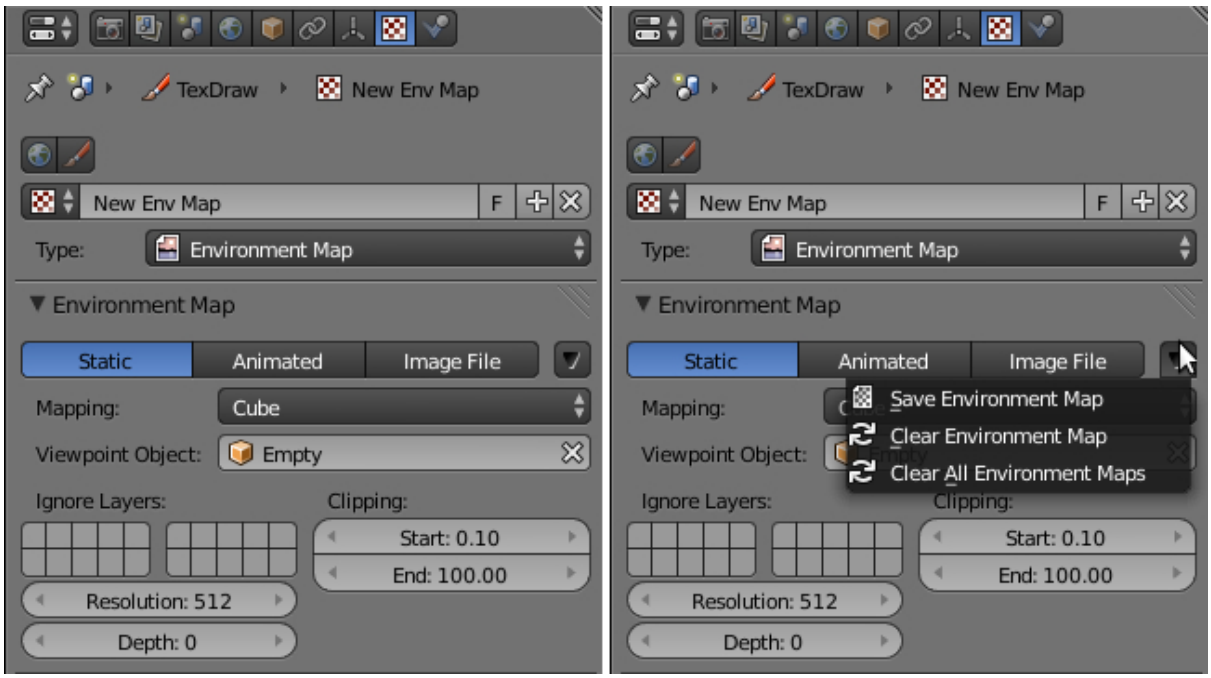


9.9.1 Environment map creation

Blender has an option for baking a scene into an environment map. To do this:

1. Create a scene for baking.
2. Add an empty object in the supposed point of view (Add > Empty).
3. Go to the World tab then to the Textures tab and create a new texture with the Environment Map type.
4. On the Environment Map panel select the Static source, then select the empty object in the Viewport Object field, then set the 2^N dimension (512, 1024 etc).
5. Render the scene by pressing F12 (a camera is required).
6. Save the environment map into a file.





9.10 Mirror Map

A mirror map is used to visualize the surface reflection. This is an environment map.

9.10.1 Activation

Select the Environment Map texture type (Type). Enable the Shading > Mirror checkbox on the Textures > Influence panel.

9.10.2 Additional settings

Influence > Shading > Mirror The degree to which the mirror map affects the reflection. The default value is 1.0.

See also:

Static reflection.

9.11 Skydome

A skydome is used to visualize an infinitely far environment (for example the sky). This is an environment map.

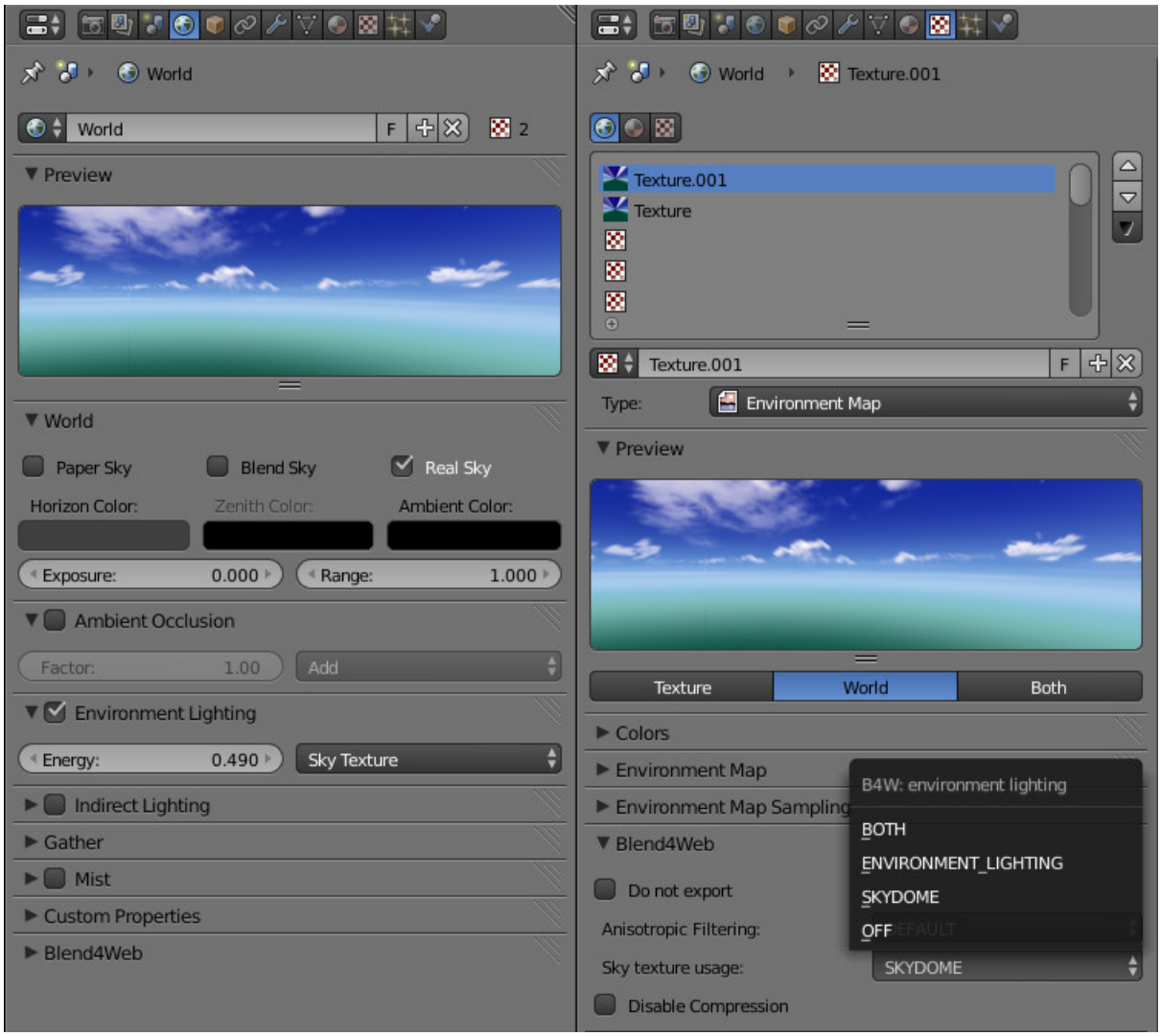
Can be also used to imitate environment lighting for objects.

9.11.1 Activation

Create a world texture with the Environment Map type and select the Blend4Web > Sky texture usage > SKYDOME option.

Note: To imitate environment lighting for objects you can select the Blend4Web > Sky texture usage > ENVIRONMENT_LIGHTING option. To show this property on the object settings, enable the corresponding world option: Environment Lighting > Sky Texture.

To use the world texture both for skydome and for environment lighting, select Blend4Web > Sky texture usage > BOTH.



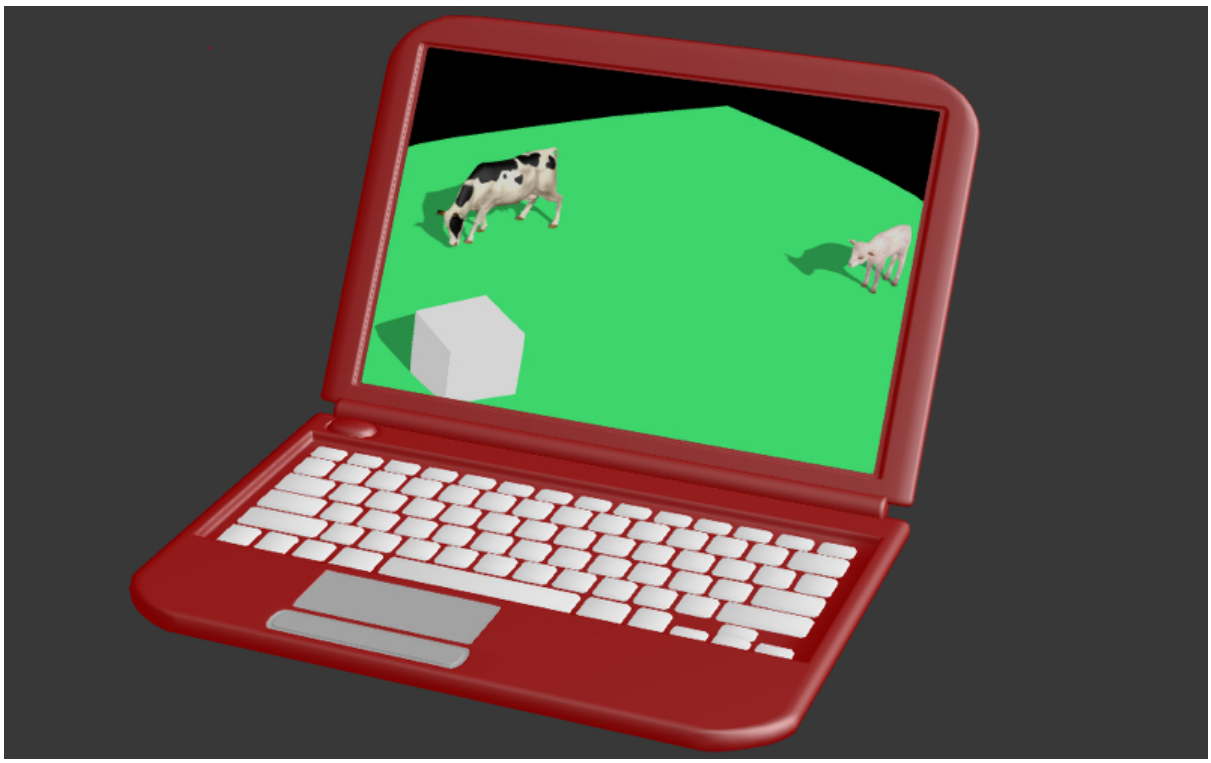
9.12 Render-to-Texture, RTT

A 3D scene's real-time rendered image can be used as a texture by an object from another scene ("main" scene).

9.12.1 Activation

1. Create an additional source scene, rename it for convenience, create a World, add the objects wanted, setup the camera view.
2. Set the None type for a texture of the target object on the main scene, and specify the source scene's name in the Blend4Web > Source scene field. Select the UV in

the Mapping > Coordinates menu. Make sure that the target object mesh has a UV map.



9.12.2 Limitations

Note that there is a bug at the moment which requires both scenes to have at least one shared light source.

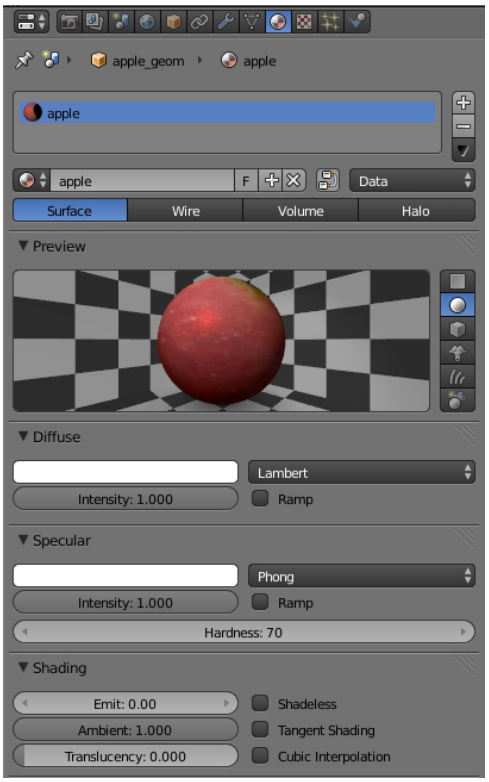
Materials

Materials describe the object surface's response to light and also contain information about its transparency, reflectivity, physical parameters and so on.

Meshes can have one or more materials. In case of multiple materials they can be assigned to different polygons in the Edit Mode. To do this select the needed polygons, select the needed material from the list and click the Assign button.

The following material types are supported: Surface, Halo.

10.1 Lighting Parameters



Diffuse > Color Diffuse light color. The default value is (0.8, 0.8, 0.8). It may interact with the diffuse map color.

Diffuse > Intensity Diffuse light intensity. The default value is 0.8.

Diffuse > Shader Diffuse shading algorithm. The engine supports the following algorithms: Lambert, Oren-Nayar, Fresnel. The default value is Lambert.

Specular > Color Specular light color. The default value is (1.0, 1.0, 1.0). It may interact with the specular map color.

Specular > Intensity Specular light intensity. The default value is 0.5.

Specular > Hardness Exponent in the specular shading calculation formula. The default value is 50. Note that the formula used in the engine differs slightly from the Blender's one.

Specular > Shader Specular shading algorithm. The engine supports the following algorithms: CookTorr, Phong - both are treated as the same, and WardIso. The default value is CookTorr.

Shading > Emit Emission intensity. The default value is 0.0.

Shading > Ambient Ambient influence factor on material. The default value is 1.0.

Shading > Shadeless When enabled, a material doesn't react to light. Disabled by default.

Game Settings > Backface Culling When enabled, poligons' back faces are not rendered by the engine. Enabled by default.

Options > Vertex Color Paint Mesh vertex color is used instead of the material diffuse color when the checkbox is enabled.

10.2 Transparency

10.2.1 Types

Transparency implementation type can be selected in the Alpha Blend menu on the Materials > Game Settings panel (in the Blender Game mode).

The engine supports the following transparency implementation types (sorted in the ascending order by performance):

Alpha Sort Transparent with a gradient. The engine sorts the triangles by camera distance in order to render overlapping transparent surfaces correctly. This operation is computationally expensive. It is recommended to use this feature for closed transparent geometry (bottle, car glass etc).

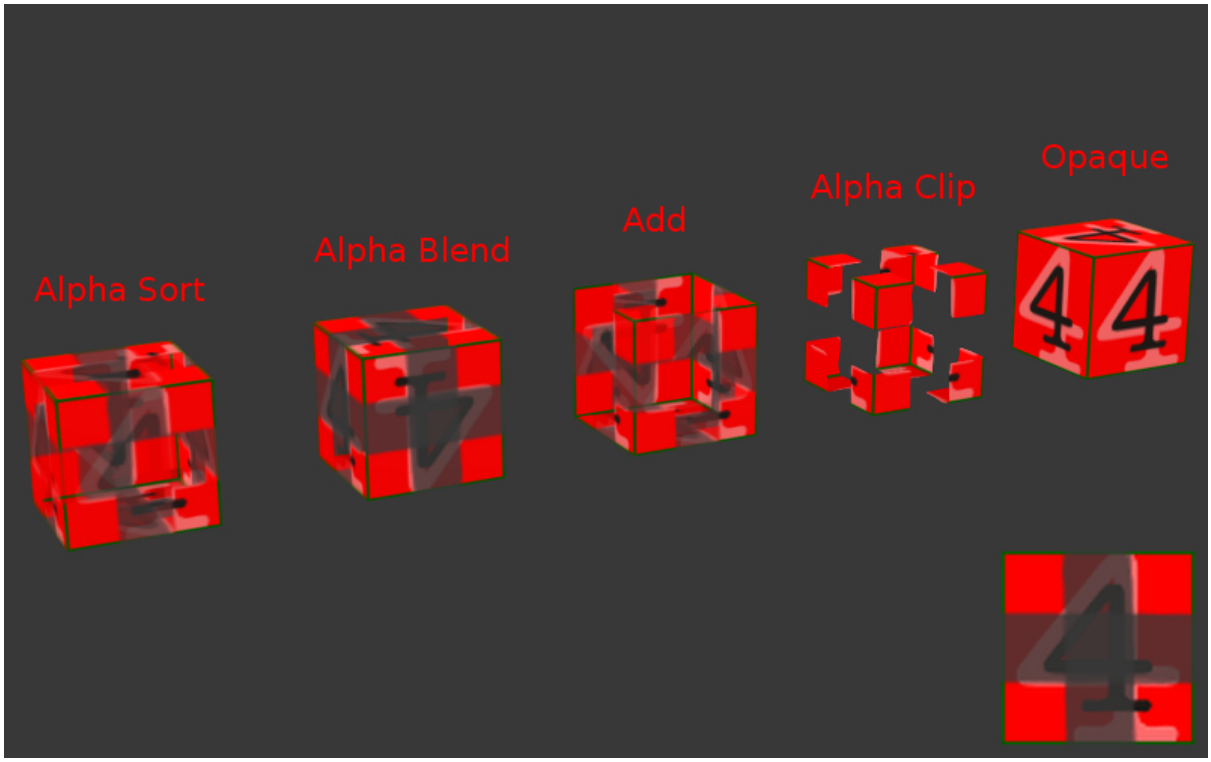
Alpha Blend Transparent with a gradient. The sorting of triangles is not performed. It is recommended to use this feature for unclosed transparent geometry (water surface, decals).

Add Transparent with a gradient. The sorting of triangles is not performed. The engine disables writing to the depth buffer which causes transparent surfaces to be rendered

in arbitrary order. It is recommended to use this feature for effects (particle systems, glowing beams).

Alpha Clip Transparent without a gradient. The engine discards pixels if their alpha is less than 0.5. 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).

Opaque Non-transparent. Alpha is ignored. This is the default value.



10.2.2 Additional settings

Transparency Enabling the transparency checkbox is required for viewing transparent objects in Blender. The engine ignores this option - the Alpha Blend option is used instead.

Transparency > Alpha Material transparency level. The engine ignores this parameter (in contrast to Blender) if there is a diffuse texture - the alpha channel values of a texture are used instead.

Options > Z Offset, depth offset This option explicitly specifies relative positioning order of objects with different materials with the purpose of depth sorting. The option can take both negative and positive values. The more distant the object is the lesser parameter value should be to provide correct rendering. The default value is 0.0.

Transparency > Fresnel Fresnel power for transparency. It is exported but is not used at the moment.

Transparency > Blend Fresnel factor for transparency. It is exported but is not used at the moment.

10.3 Reflection

10.3.1 Static reflection

A surface reflects the same image no matter how the environment changes. For activation simply use the mirror map.

See also:

Fresnel effect for reflection

10.3.2 Dynamic reflection

A surface reflects the selected objects in their current position. Only reflection from a plane is supported.

Activation

1. Enable the Render reflections checkbox on the Scene > Blend4Web panel.
2. Add an empty object as a reflection plane by executing for example Add > Empty > Single Arrow. Rename it for convenience.
3. For reflective objects enable the Reflective checkbox on the Object > Blend4Web panel and specify the empty object name in the Reflection plane field.
4. For the needed materials of the reflective objects enable the Mirror > Reflectivity value.
5. For the reflexible objects enable the checkbox Reflexible on the Object > Blend4Web panel.

Note: It is also recommended to enable the World > Environment Lighting checkbox.

Limitations

Normal maps and shadows are ignored in the reflected image for optimization purposes.

See also:

Fresnel effect for reflection

10.3.3 Fresnel effect for reflection

The Fresnel effect manifests itself as the dependency of the intencity of passing and reflected light on the incidence angle. If the angle of incidence is close to zero (i.e. light falls almost at right angle to the surface) the passing light portion is large and the reflected

light portion is small. On the contrary if the angle of incidence is close to 90 degrees (i.e. light falls almost parallel to the surface) almost all light is reflected.

The engine uses the approximate Schlick's formula:

$R = R_0 + (1 - R_0)(1 - \cos\theta)^N$, where

R - reflection coefficient,

R_0 - reflection coefficient in case of viewing at a right angle to the surface (i.e. when $\theta = 0$),

θ - angle of incidence (which is equal to the angle of reflection under which light enters the camera), it is calculated by the engine in real-time,

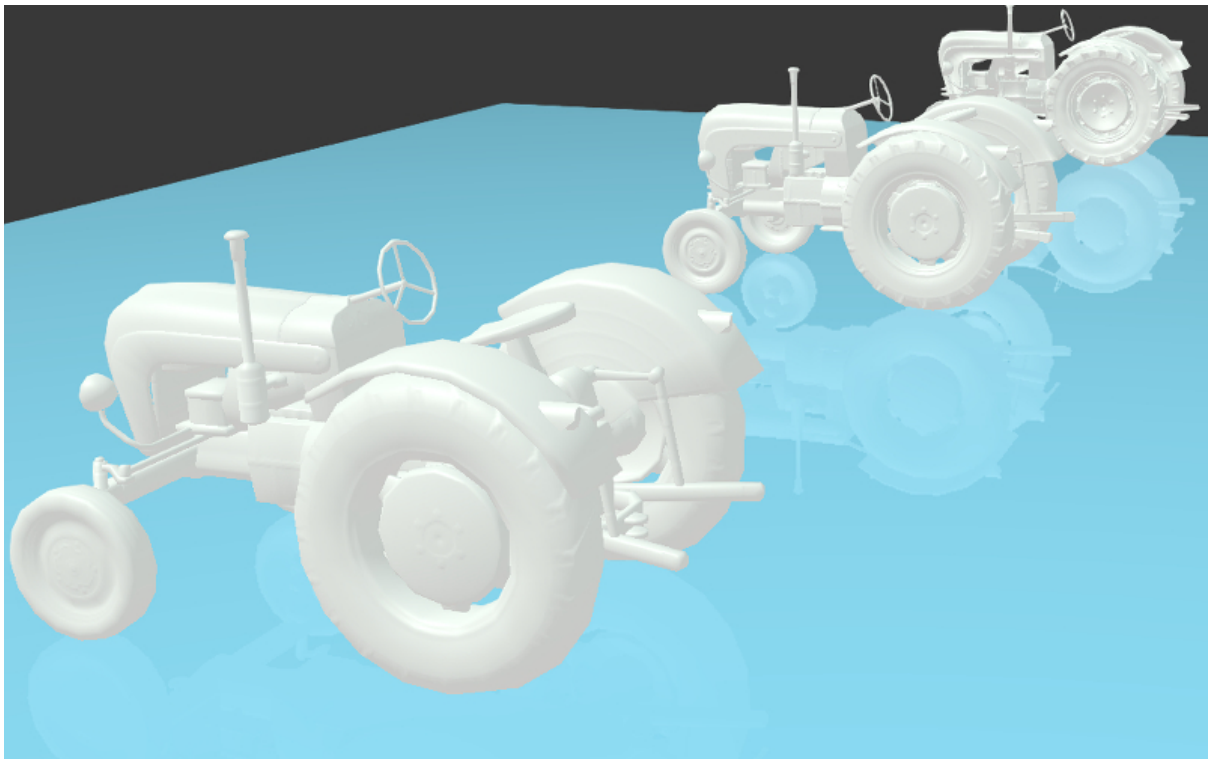
N - exponent.

Settings

Fresnel effect can be set up both for static and dynamic reflection.

Mirror > Fresnel Fresnel power for reflection. This is the N exponent in the Schlick's formula. In Blender it is limited to values from 0 to 5. If this parameter is equal to zero the Fresnel effect is not observed and the full reflection at all angles occurs. If this parameter is greater than zero, the material is less reflective when viewing surfaces at angles which are close to the right angle. The bigger this parameter is the bigger is the angle deviation from the right angle for which the Fresnel effect is observed.

Mirror > Blend Fresnel factor for reflection. It is reduced to R_0 in the Schlick's formula by the following expression: R_0 = 1 - Blend / 5. In Blender it is limited to values from 0 to 5. This parameter defines the Fresnel effect intensity: the bigger the Blend factor is, the more is the Fresnel effect influence. If it is equal to zero the Fresnel effect is not observed.



10.4 Parameters Specific to the Engine

Located on the Blend4Web panel.

Do not export Material is not to be exported.

Special: Water A special material for water rendering.

Special: Collision A special material for collision geometry.

See also:

Physics

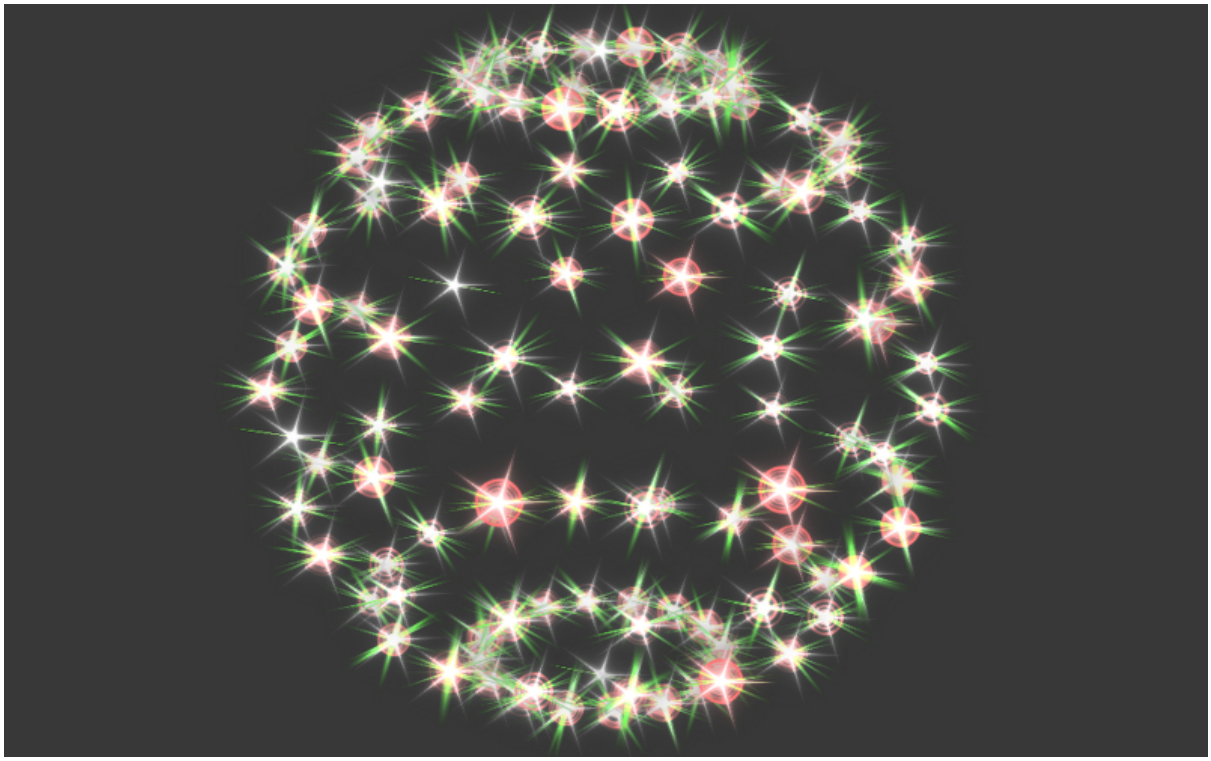
Double-sided Lighting Enables the double-sided lighting mode. This option is useful for non-transparent objects with a single-layered mesh.

10.5 Halo Materials

Halo materials are used in particle systems and in static meshes. Using the halo in static meshes is described below.

10.5.1 Activation

Select the Halo type under the Materials tab. It's also recommended to select the transparency type with a gradient (Add, Alpha Blend or Alpha Sort).



10.5.2 Additional settings

Halo > Alpha Material transparency factor. The default value is 1.0 (non-transparent).

Halo > Color Material color. The default value is (0.8, 0.8, 0.8) (almost white).

Halo > Seed Unused.

Halo > Size Particle size. The default value is 0.5.

Halo > Hardness Exponent for computing the gradient. Affects visible dimensions of particles. The default value is 50.

Halo > Add Unused.

Halo > Rings Use rings. Relative quantity and color can be set up.

Halo > Lines Use lines. Relative quantity and color can be set up.

Halo > Star Tips Use stars. The quantity of edges can be set up.

Blend4Web > Special: Stars Enables the starry sky rendering mode while the mesh is fixed relative to the camera. For the lamp it is also required to enable the Blend4Web > Dynamic intensity checkbox. Applications must set up the hours of darkness via API.

Blend4Web > Blending Height Height range for the fading of stars.

Blend4Web > Stars Minimum Height Minimum height in the object's local space at which stars are visible.

CHAPTER 11

Node Materials

Shader nodes extend significantly the potential of Blender's standard materials by means of presenting shading as a batch of basic transformations.

11.1 Standard Nodes

All Blender functions are supported except the following cases:

- Geometry - the Local, Orco and Vertex Alpha outputs are not supported.
- Material, Extended Material - no more than one node per material is allowed; the Refl, Ambient, SpecTra, Ray Mirror inputs are not supported; the AO output is not supported.
- RGB Curves is not supported.
- Vector Curves is not supported.

In addition a poor performance of some nodes in real-time context should be taken into account. It is not recommended to use the following nodes:

- Hue/Saturation
- MixRGB - the Burn, Dodge, Value, Saturation, Hue, Color types.

It is not recommended to create very complex materials especially if they use many Geometry or Texture nodes.

11.2 Special Nodes

Special nodes extend the standard nodes' functionality in order to support the engine features. Special nodes are created as node groups (Node groups or Node tree) with specially determined names and input formats. All special nodes are gathered in the special_nodes.blend file for convenience.

11.2.1 LINEAR_TO_SRGB and SRGB_TO_LINEAR

Color correction from linear color space to sRGB space and back.

See also:

Gamma Correction in Node Materials

11.2.2 REPLACE

The node replaces the inputs depending on the working environment (i.e. Blender viewport or Blend4Web). When working in Blender the Color1 input is connected to the Color output and the Color2 input is ignored. On the contrary when working in the engine the inputs are interchanged (the Color1 one is ignored and the Color2 one is connected to the output). The node is intended to display one node structure in the viewport and another - in the engine.

As a rule it is used for normal mapping. Blender's node materials do not support a tangent space of coordinates. Therefore the only possible method to display normal maps in the viewport correctly is their usage inside the Material nodes.

11.2.3 CLAMP

The node limits the output value. As a result all the output vector components take values from 0 to 1 inclusive.

11.2.4 TIME

Provides the timeline counting from the engine start (in seconds). May be used for animating any parameters in the node materials.

11.2.5 NORMAL_VIEW

The node transforms a normal into the camera's space of coordinates. Transformation is necessary because the engine defines all normals in the world space of coordinates. A normal should be used only for effects and not for connecting to the output of the Material or Extended Material nodes.

11.2.6 PARALLAX

The node implements the texture coordinates offset using a height map.

Input parameters

UV Source texture coordinates

Height RGBA texture with a height map packed into the alpha channel.

Scale Texture coordinates offset factor

Steps The number of steps for iterative generation of texture coordinates offset. The bigger this value is the better is the final quality.

Lod Distance Maximum distance from camera at which the effect is observed.

Output parameters

UV Resulting texture coordinates which are used as input for the texture nodes.

11.2.7 TRANSLUCENCY

The node implements a translucency effect (with respect to light sources only) for thin objects such as cloth, leaves, paper etc. The effect consists of two parts: 1) brightening of the object side which is opposite to the light source and 2) appearance of a light spot right in the light source place.

Input parameters

Color One-channel texture which defines material heterogeneity - the white color denotes maximum translucency effect while the black color denotes its absence. White color is used by default.

Backside Factor Material color correction coefficient for the side which is opposite to the light source. It describes the color richness effect for the translucent areas.

- Backside Factor < 1 - brightening
- Backside Factor $= 1$ - no correction
- Backside Factor > 1 - darkening

The default value is 1.

Spot Hardness Light spot blurring factor. The bigger this value is the smaller is the spot and the sharper are the spot edges. The default value is 1000.

Spot Intensity Light spot intesity. The bigger this value is the brighter is the light spot. The default value is 1.

Spot Diffuse Factor Material diffuse color influence on the light spot color.

- Spot Diffuse Factor $= 0$ - the light spot has the diffuse color
- Spot Diffuse Factor $= 1$ - the light spot color is white

The default value is 1.

Output parameters

Translucency The output should be connected to the Translucency input of the Extended Material node.

CHAPTER 12

Lighting, Shadows and Background

12.1 Lighting with Light Sources

A scene can have multiple (but not less than one) light sources of different types.

12.1.1 Light source types

The following light source types are supported:

Point Light propagates from one point uniformly to all directions with gradual attenuation.

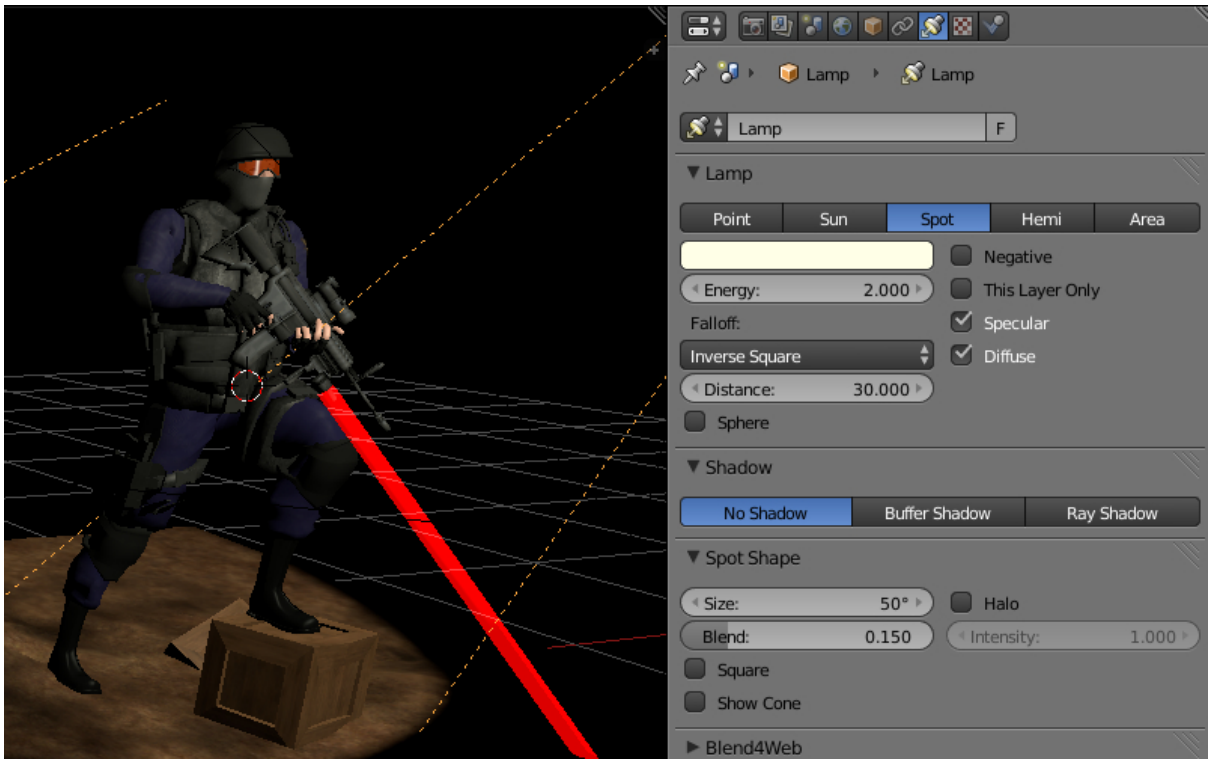
Sun Light propagates from an infinite plane in one direction without attenuation.

Spot Light propagates from one point within the angular limit, with gradual attenuation.

Hemi Hemispherical. Light propagates from an infinite hemisphere without attenuation.

12.1.2 Light source setup

Performed in the Object Data tab when a lamp object is selected.



Color Light color. The default value is (1.0, 1.0, 1.0) (i.e. white).

Energy Radiation intensity. The default value is 1.0.

Falloff Attenuation type. The value is exported but the engine always uses Inverse Square. It is applicable to the Point and Spot light source types. The default value is Inverse Square.

Distance Attenuation parameter. It is applicable to the Point and Spot light source types. The default value is 25.0.

Spot Shape > Size Cone angle in degrees. It is applicable to the Spot light source type. The default value is 45°.

Spot Shape > Blend Parameter for blurring light spot edges. It is applicable to the Spot light source type. The default value is 0.15.

Blend4Web > Do not export Don't export the light source. Disabled by default.

Blend4Web > Generate shadows Use this light source for shadow calculation. Should be used when multiple light sources are present. Disabled by default.

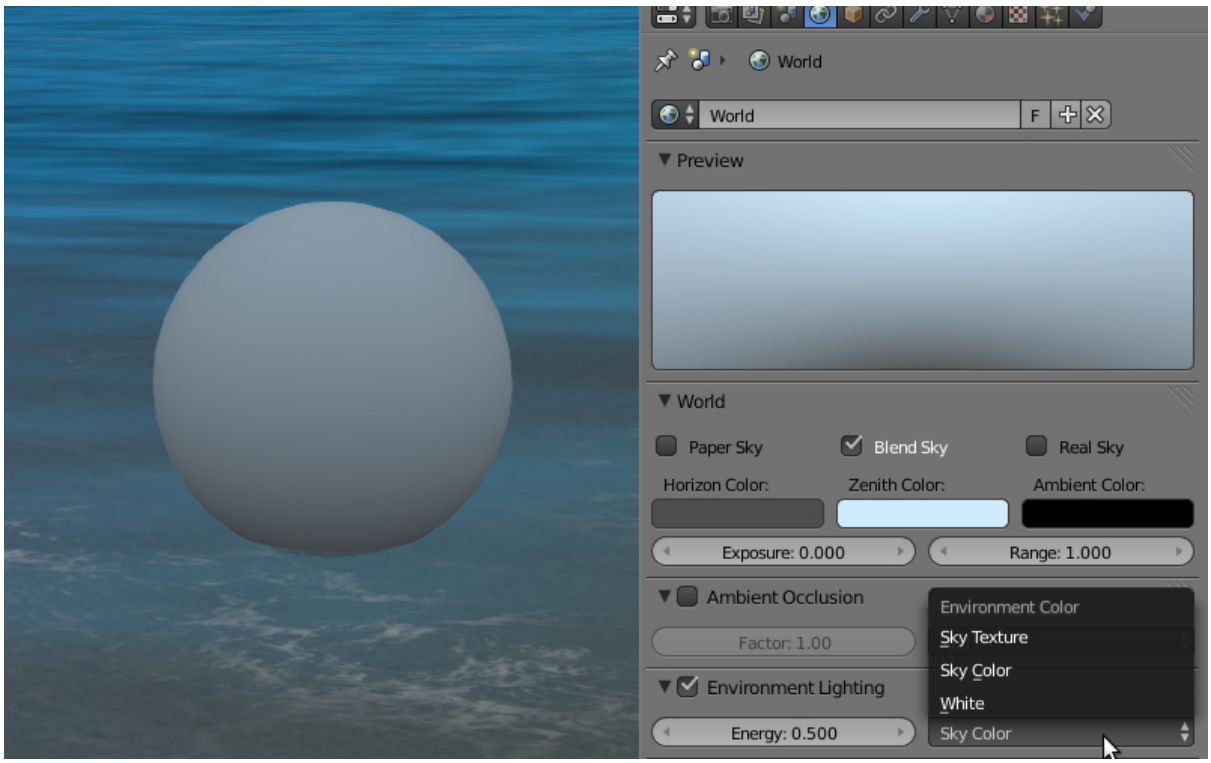
Blend4Web > Dynamic intesity Use this light source for calculating the time of day. Applicable only to the Sun light source type. Disabled by default.

12.2 Ambient Lighting

The engine uses a simple hemispherical lighting model in which horizon and zenith colors should be specified.

12.2.1 Activation

Enable the Environment Lighting checkbox on the World tab.



12.2.2 Setup

Environment Lighting > Energy Ambient lighting intensity. The default value is 1.0.

Environment Lighting > Environment Color Ambient lighting source type - the supported types are White and Sky Color. If the White type is selected, both horizon and zenith colors are white. If the Sky Color type is selected, the horizon and zenith colors can be specified by means of the World > Horizon Color and World > Zenith Color color pickers. The default value is White.

World > Horizon Color and World > Zenith Color Horizon and zenith colors. It is recommended to activate the World > Blend Sky option for better color selection.

12.3 Shadows

12.3.1 Activation

1. Enable the Blend4Web > Shadows Cast checkbox under the Object tab for the objects which cast shadows.
2. Enable the Blend4Web > Shadows Receive checkbox under the Object tab for the objects which receive shadows.
3. Make sure that the Blend4Web > Render shadows checkbox is enabled in the Scene tab.

12.3.2 Setup

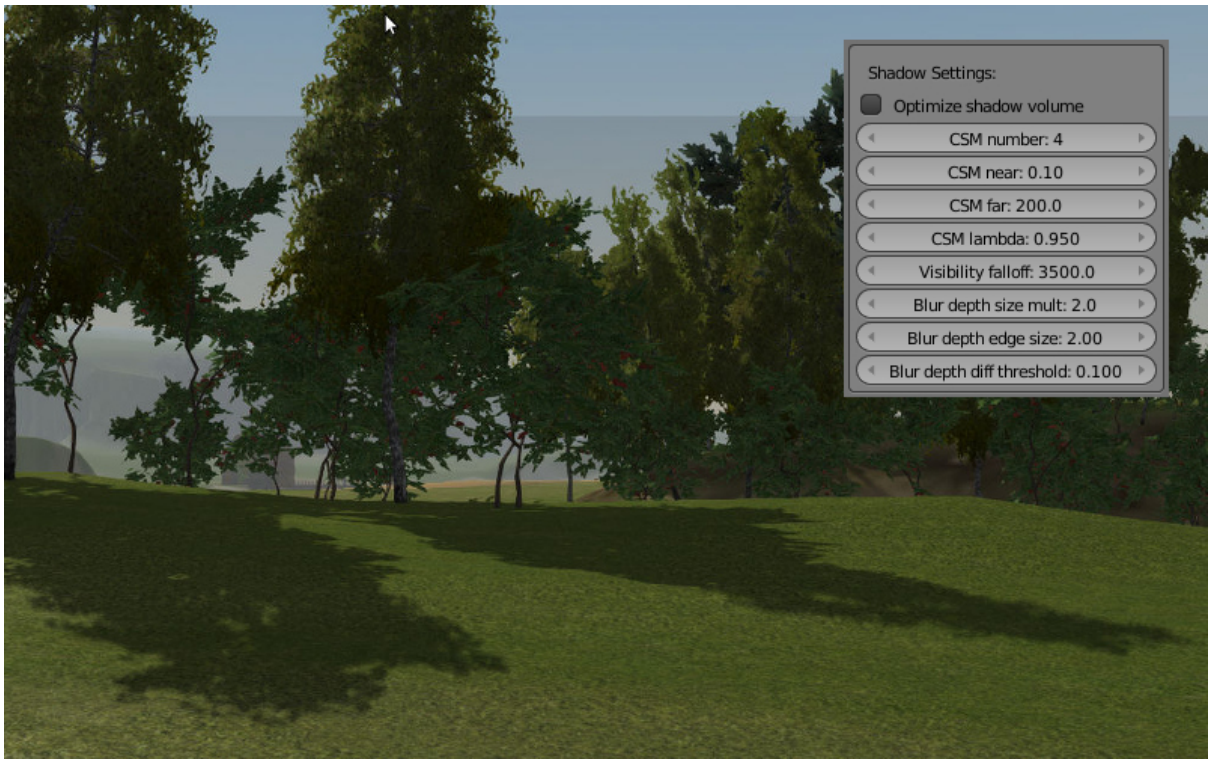
Direction If there are multiple light sources, it is recommended to specify the exact light source which is used for shadow calculations, by enabling the Blend4Web > Generate shadows checkbox under the Object Data tab for the selected lamp object.

Color The shadow color is determined by ambient lighting settings.

The following additional settings are on the Blend4Web > Shadow Settings panel of the World tab:

Cascades

In order to provide acceptable shadow quality and to cover considerable space at the same time it is required to use multiple stages for shadow generation (cascades). Thus the best quality cascades are situated near the observer while the worst quality cascades are in the distance.



CSM number Number of shadow cascades. Supported from 1 to 4 cascades. The default value is 3.

CSM near The nearest boundary of shadow rendering. The default value is 0.1.

CSM far The furthest boundary of shadow rendering. The default value is 100.

CSM lambda The distribution factor of cascade boundaries. The values for the calculated cascade boundaries are displayed in the viewer under the Shadows tab. The default value is 0.875.

Softened shadows

Visibility falloff Factor for the exponential decay of shadow visibility which depends on the distance between the casting and receiving points. It is used to reduce self-shadowing artefacts (i.e. when the castor and the receiver are the same object). The default value is 3500.0.

Blur depth size mult Blur kernel size. Affects the shadow softening ratio. The default value is 1.0.

Blur depth edge size Difference between samples (in texels) for the edge detection algorithm. Decreases haloing by excluding edges from blurring. The default value is 2.0.

Blur depth diff threshold Maximum of depth difference for the edge detection algorithm multiplied by 1000. Decreases haloing by excluding edges from blurring. The default value is 0.1.

12.4 Background

You can change the background in the following ways:

1. Place the whole scene inside a model (e.g. a cube or a sphere) with its normals directed inside, with a material and an optional texture.
2. Use a cube texture.
3. Set up the procedurally generated atmosphere.
4. Set the engine's background_color parameter with the config.set() method. This value is used as argument for the WebGL clearColor() method. For correct results, it's recommended to turn the WebGL context transparency off (the alpha parameter). Such the configuration is used by default in the engine's standard web player.

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

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

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

5. You can use any HTML content behind the canvas element, to which the rendering is performed, as a background. To do this, activate the WebGL context transparency (the alpha parameter). For correct results, it's recommended to set absolutely transparent black background color. Such the configuration is used by default in the scene viewer of Blend4Web SDK.

```
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(...);
```

See also:

Alpha Compositing

CHAPTER 13

Postprocessing Effects

13.1 Motion Blur

The motion blur effect can be used to improve the realism of an interactive scene. It is displayed as picture blurring when the camera or objects move.

13.1.1 Activation

Enable the Enable Motion Blur checkbox on the Scene > Blend4Web panel.

13.1.2 Additional settings

On the World > Blend4Web > Motion blur settings panel:

Motion blur factor Effect appearance ratio. The higher this value is the stronger is the motion blur.

Motion blur decay threshold Blur fade-out ratio. The higher this value is the more distinct is the effect. The default value is 0.01.

13.2 Depth of Field

The depth of field effect (DOF) can be used to accentuate a part of a scene. It is displayed as picture blurring nearer and further from the camera focus.

13.2.1 Activation

1. Select an active camera and go to its settings panel (Object Data).
2. Then two options are available:
    - Select an object to use as the camera's focus in the Focus menu of the Depth of Field panel. In this case moving away or approaching this object will cause a corresponding correction of the camera focus.
    - Set a non-zero value for the Distance on the Depth of Field panel (in Blender units = meters). In this case the camera focus will be located at this distance from the camera and will move together with it.

13.2.2 Additional settings

On the Object Data > Blend4Web panel when the active camera is selected:

DOF front distance The distance from the focus to the nearest plane (relative to the camera) behind which full blurring occurs (in meters). The default value is 1.0.

DOF rear distance The distance from the focus to the furthest plane (relative to the camera) behind which full blurring occurs (in meters). The default value is 1.0.

DOF power Blurring ratio. The default value is 3.0.

13.3 Screen-Space Ambient Occlusion

The screen-space ambient occlusion (SSAO) effect can be used to fake complex light reflections from objects. The basis of this effect is that the space between close objects is less accessible for diffused light and hence is darker.

13.3.1 Activation

Enable the Enable SSAO checkbox on the Scene > Blend4Web panel.

13.3.2 Additional settings

On the World > Blend4Web > SSAO Settings panel:

Radius Increase The spherical sampling radius multiply factor when transfering from the internal sampling ring to the external one. The default value is 1.7.

Dithering Amount Ratio for random noise mixing used for reducing strips. The default value is 0.1.

Gauss Center Mathematical expectation - Gauss distribution parameter for the depth difference between a pixel and a neighbouring sample. The default value is 0.2.

Gauss Width Standard deviation - Gauss distribution parameter for the depth difference between a pixel and a neighbouring sample. The default value is 2.0.

Gauss Width Left Standard deviation in case the depth difference is less than the mathematical expectation. The default value is 0.1.

Influence SSAO appearance factor. The default value is 0.7.

Distance Factor Factor of SSAO decay with distance. The default value is 0.0 (i.e. no decay).

Samples Number of samples (the more samples there are the better is the quality but the poorer is the performance). The default value is 16.

13.4 God Rays

The god rays effect (aka crepuscular rays) simulates well-known natural phenomenon - the shining of illuminated air parts.

13.4.1 Activation

Enable the Enable God Rays checkbox on the Scene > Blend4Web panel.

13.4.2 Additional settings

On the World > Blend4Web > God Rays Settings panel:

God Rays Intensity The effect appearance factor. The default value is 0.7.

Maximum Ray Length Rays length factor. Defines the step between samples of radial blurring. The default value is 1.0.

Steps Per Pass Number of steps per single sample. The default value is 10.0.

13.5 Bloom

Bloom appears when a picture has elements with a very different brightness. A glowing halo is created around the bright details.

13.5.1 Activation

Enable the Enable Bloom checkbox on the Scene > Blend4Web panel.

13.5.2 Additional settings

On the World > Blend4Web > Bloom Settings panel:

Key Bloom intensity.

Blur Bloom blurriness factor.

Edge Luminance The boundary value of an element's relative brightness above which the bloom effect appears.

13.6 Glow

The glow effect consists of illuminating the outline of the certain objects with some color. As a result a glowing halo is created around the object.

13.6.1 Activation

The glow effect can be activated by an application via API. The different models can be applied such as constant glow, fading out glow, pulsatory glow and any other. In order to enable the glow effect on a certain object it's required to enable the Selectable checkbox on the Object > Blend4Web panel.

13.6.2 Additional settings

On the Object > Blend4Web panel:

Glow duration Duration of the glow animation in seconds. The default value is 1.

Glow period Repeat period of the glow animation in seconds. The default value is 1.

Glow relapses The amount of the glow animation iterations. If the amount equals to 0 the animation is repeated forever. The default value is 0.

On the World > Blend4Web panel:

Objects glow color Default effect color for all objects. The defaut value is (1,1,1).

Glow factor When this parameter decreases so does the thickness and the brightness of the halo around the object. The default value is 1.

These settings are taken as default when the glow effect is initiated via API.

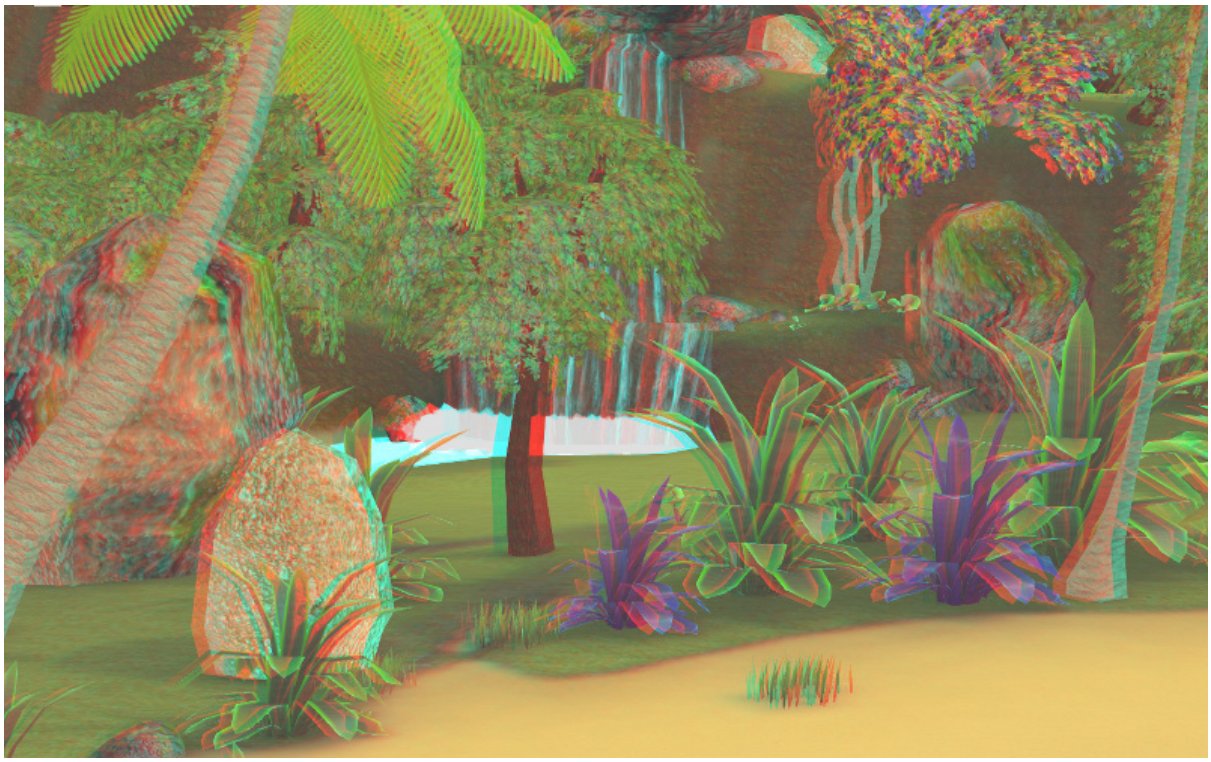
13.7 Stereoscopic Rendering (Anaglyph)

13.7.1 Activation

The stereoscopic rendering mode is intended for viewing the content using special glasses. It is activated by an application via API.

13.7.2 Additional settings

No.



13.8 Color Correction

13.8.1 Activation

Enable the Enable Color Correction checkbox on the Scene > Blend4Web panel.

13.8.2 Additional settings

On the World > Blend4Web > Color Correction Settings panel:

Brightness The default value is 0.0.

Contrast The default value is 0.0.

Exposure The default value is 1.0.

Saturation The default value is 1.0.

13.9 Anti-Aliasing

Anti-aliasing is used to reduce undesirable rendering artefacts (poor pixelization).

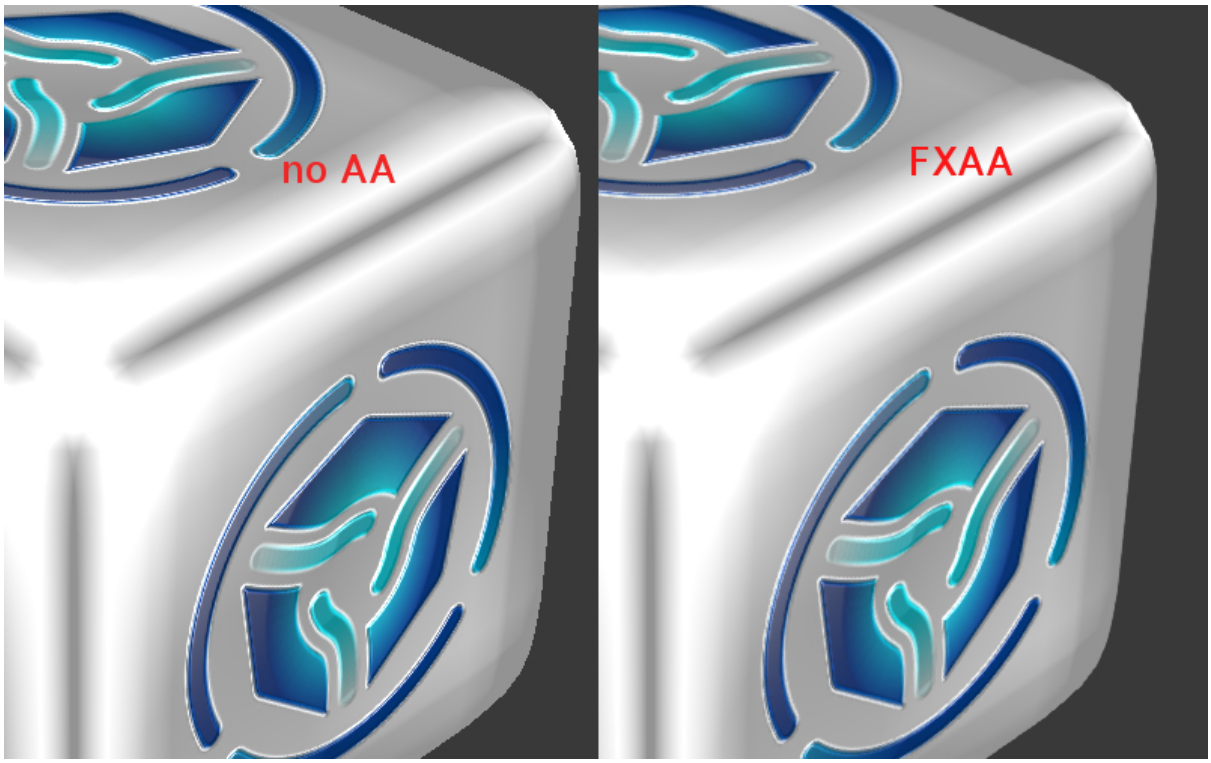
13.9.1 Activation

Enable the Enable Antialiasing on the Scene > Blend4Web panel.

13.9.2 Additional settings

The anti-aliasing method is assigned simultaneously with the selection of the engine performance profile.

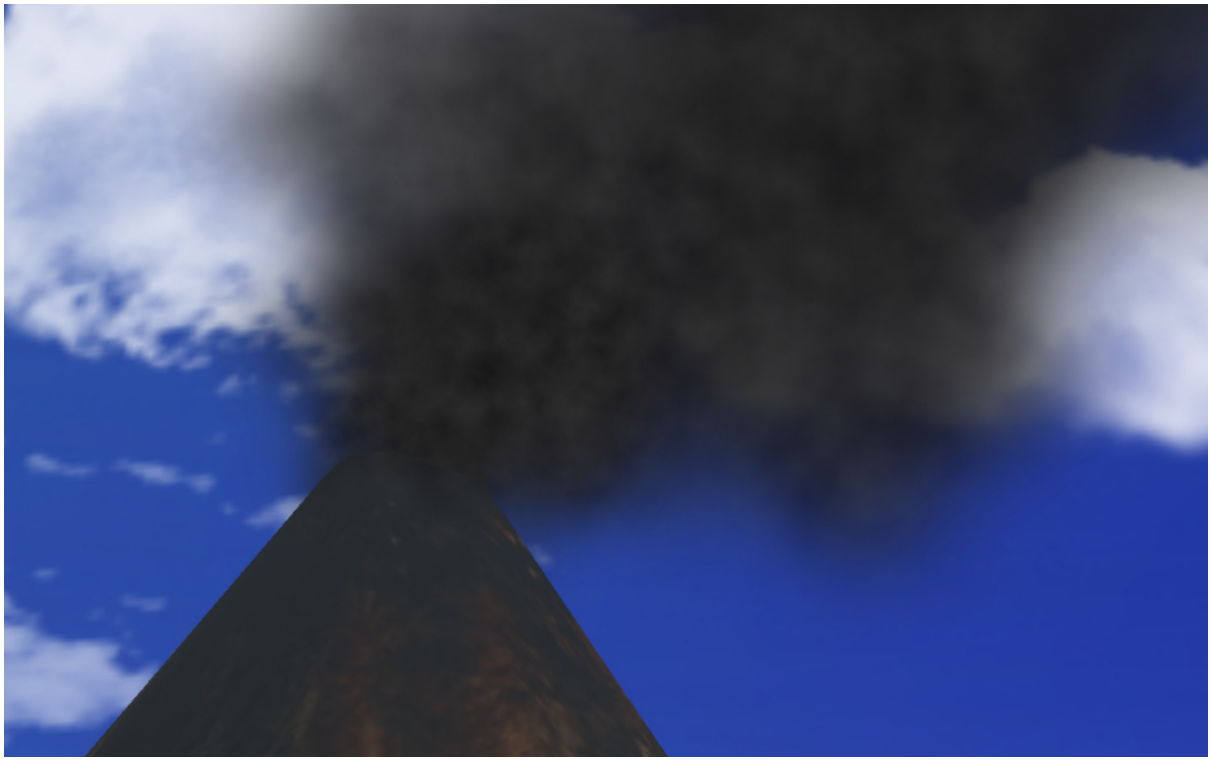
- low quality - anti-aliasing is disabled
- high quality - the anti-aliasing method is FXAA (Fast Approximate Anti-Aliasing) by Nvidia
- maximum quality - the anti-aliasing method is SMAA (Enhanced Subpixel Morphological Anti-Aliasing) by Crytek


no AA
FXAA

CHAPTER 14

Particle System

The particle system is intended to visualize phenomena which are caused by the movement of numerous small objects such as smoke, fire, water splashes and other.



A particle system requires an emitter - an object which defines the location and the direction of the outgoing particles flow.

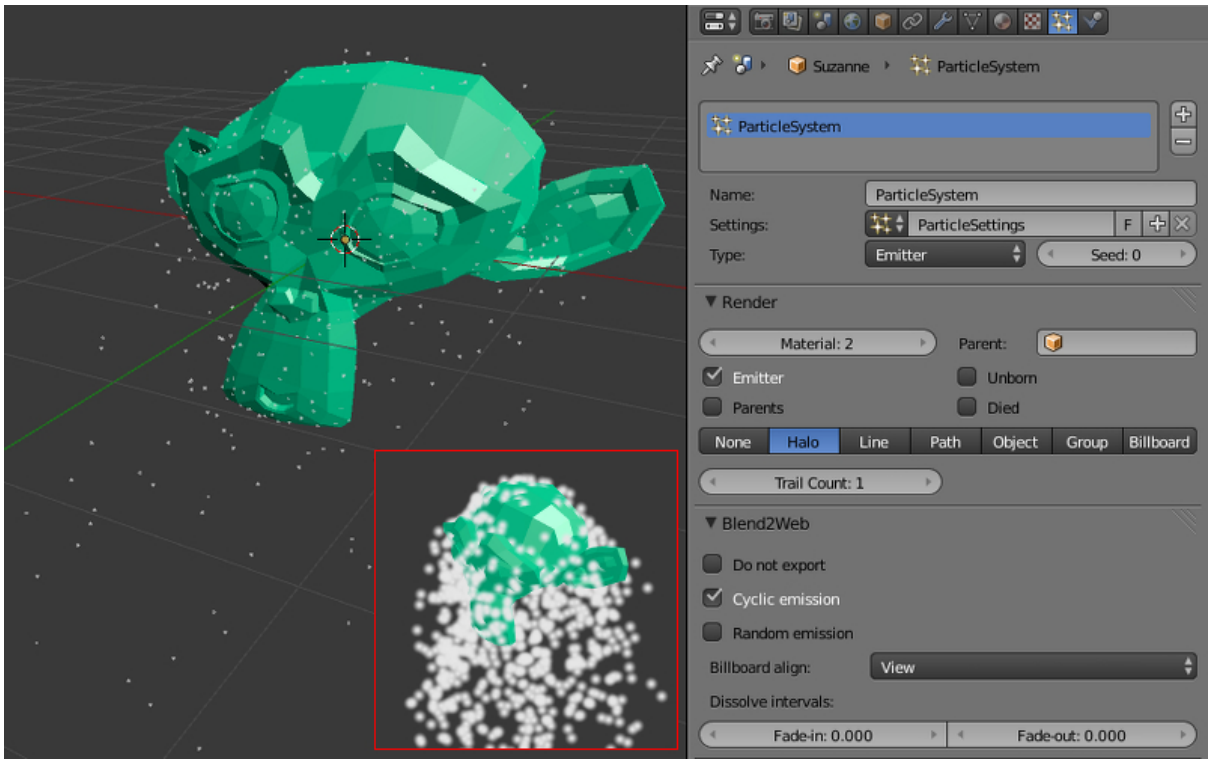
14.1 Use

14.1.1 Necessary steps

1. Add a mesh emitter to the scene.
2. Create a material for particles on the emitter, for example of the Halo type. The Surface material type with a mandatory diffuse texture is also supported.
3. Add a particle system on the emitter.
4. Initiate the engine playback. Two options are available:
    - "cyclic emission" - enable the Blend4Web > Cyclic emission checkbox for the particle system.
    - "non-cyclic animation" - enable the Blend4Web > Animation > Use default checkbox for the emitter.

14.1.2 Recommended additional settings

1. Set the Add transparency type for the particles' material.
2. Disable emitter rendering if needed using the Particles > Render > Emitter checkbox.
3. If an emitter is required on a scene use additional materials for it. In this case select the particles' material in the Particles > Render > Material menu on the particles settings panel.
4. If the Surface material type is used it is required to add a diffuse texture (normally with the alpha channel) to this material. Select UV in the Mapping > Coordinates menu. Make sure that the emitter's mesh has a UV layer.



14.2 Setup

The particle system parameters can be set up under the Particles tab. Multiple particle systems per emitter are supported.

14.2.1 Basic settings

Name Particle system name. The default name is “ParticleSystem”.

Settings Reference to the settings datablock of the particle system. The datablock settings can be shared between different particle systems.

Type Particle system type: Emitter or Hair. Hair particle systems can be used to create numerous copies of an object (so called instancing). The default is Emitter.

Seed Index in the table of random numbers which are used for particle system generation. The default value is 0.

14.2.2 Emission settings

Emission > Number Number of particles. The default value is 1000.

Emission > Start The first frame after which the emission of particles starts. The default value is 1.0.

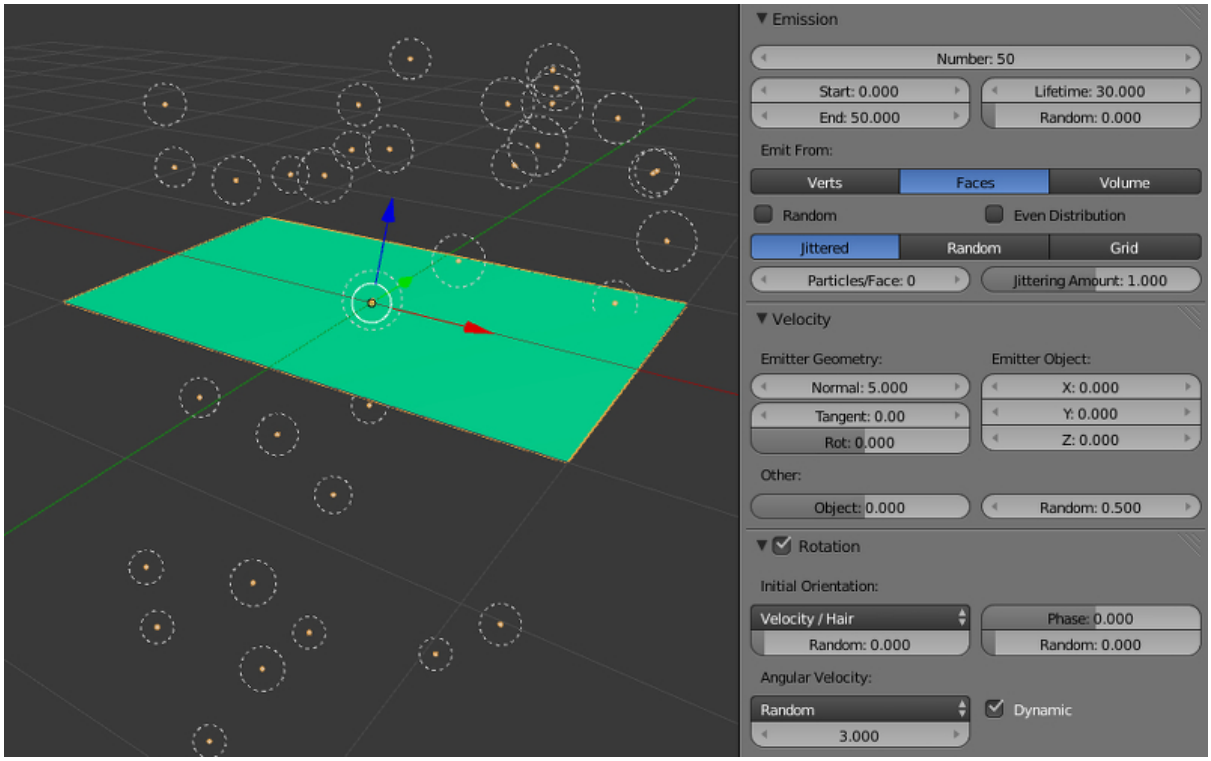
Emission > End The last frame after which the emission of particles ends. The default value is 200.0.

Emission > Lifetime The life time of particles measured in frames. The default value is 50.0.

Emission > Lifetime > Random The random factor for the life time. The default value is 0.0.

Emission > Emit From Emission source type. The following types are supported: Verts (emit from vertices), Faces (emit from polygons). The default is Faces.

Emission > Emit From > Distribution Emission distribution settings: Jittered, Random, Grid. Ignored by the engine. Internally the engine always uses Random distribution. The default is Jittered.



14.2.3 Direction settings

Only the following settings are supported:

Velocity > Emitter Geometry > Normal Factor influencing the emission along the emitter's mesh normals. The default value is 1.0.

Velocity > Other > Random Factor of randomization for emission direction. The default value is 0.0.

14.2.4 Rotation settings

Only the following settings are supported:

Rotation > Angular Velocity > Mode Mode for particle billboards self-rotating. The following modes are supported: Velocity (constant rotation speed), Random (random rotation), None (no rotation). The default is Velocity.

Rotation > Angular Velocity > Factor Factor of rotation velocity for particle billboards. The default value is 0.0.

14.2.5 Physics settings

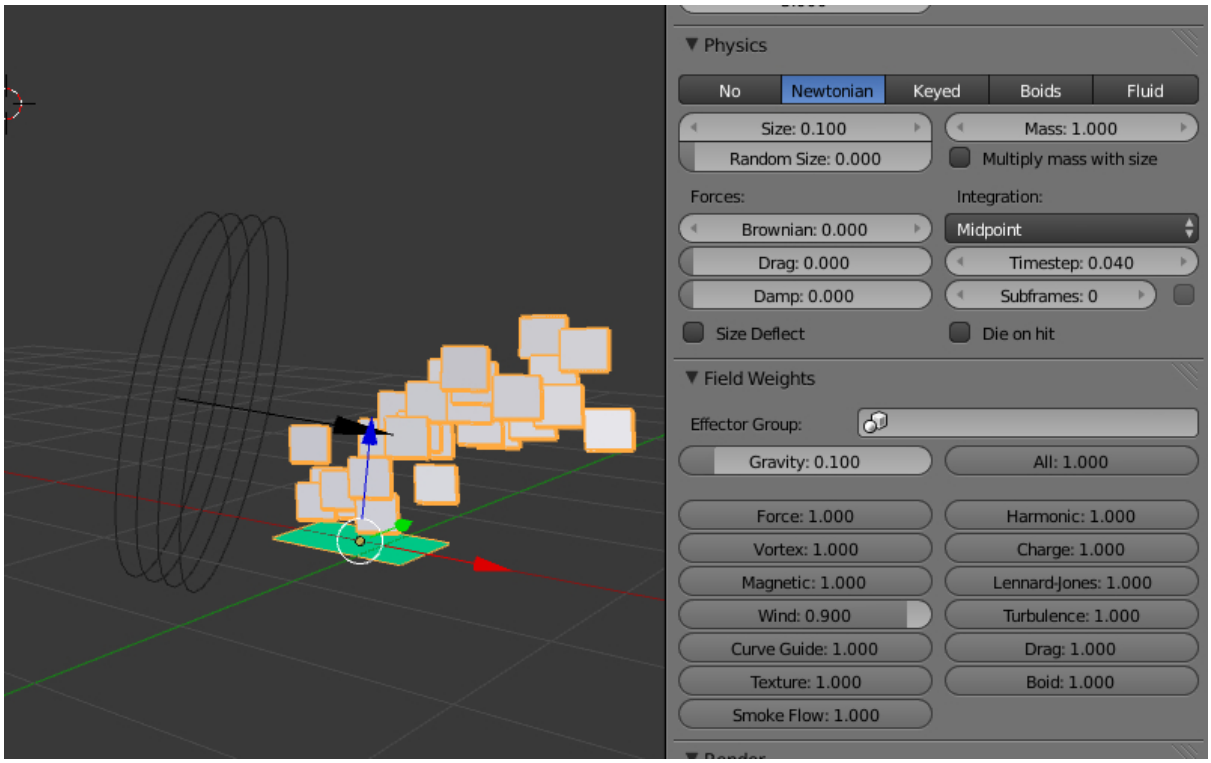
Only the following settings are supported:

Physics > Type Physics calculation type: No, Newtonian, Keyed, Boids, Fluid. Ignored by the engine. Newtonian physics is always used. The default is Newtonian.

Physics > Size Particle size. The default value is 0.05.

Physics > Mass Particle mass. Affects interaction with force fields (such as wind). The default value is 1.0.

Physics > Forces > Brownian Exported but not used by the engine.



14.2.6 Rendering settings

Only the following settings are supported:

Render > Material Menu for selecting the particle's material. Used for referencing to the particle' material in case multiple materials are used by the emitter. The default value is Default Material.

Render > Emitter Enables emitter rendering on the scene. Enabled by default.

Render > Type Particle rendering mode: None, Halo, Line, Path, Object, Group, Billboard. The engine supports the Object and the Group modes which are used for objects and groups instancing respectively. Other modes are ignored. It is recommended to use the Billboard mode for convenient display of billboards. The default is Halo.

14.2.7 Supported settings for force fields influence

Only the following settings are supported:

Field Weights > Gravity Gravity influence factor (Earth's attraction). The default value is 1.0.

Field Weights > Wind Wind influence factor. A Wind force field source should be present (can be added using Add > Force Field). A particle system is also influenced by the wind direction and strength. The default value is 1.0.

14.2.8 Engine specific settings

Blend4Web > Do not export Unsupported.

Blend4Web > Cyclic emission The option enables the cyclic emission mode. It can be used for permanent effects (such as smoke, burning, water splashes). It is recommended to set the Emission > Start value to zero. Disabled by default.

Blend4Web > Random emission The option enables a random emission time for particles. Disabled by default.

Blend4Web > Billboard align The way billboards are oriented: View - follow the camera, XY plane, YZ plane, ZX plane - align to the corresponding plane (in the world coordinate system of Blender). The default is View.

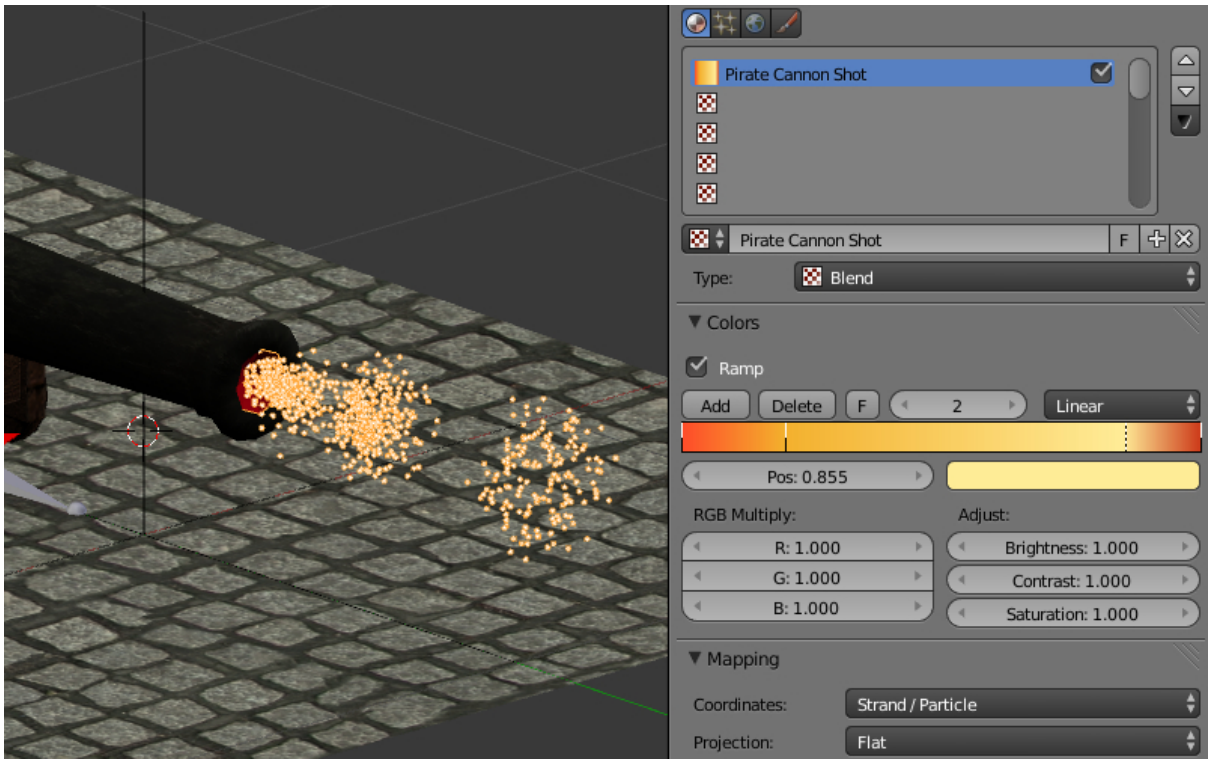
Blend4Web > Dissolve intervals > Fade-in and Fade-out Starting and ending intervals (measured in frames) for gradually increasing and decreasing the particles' transparency.

14.3 Textures in Particle Systems

14.3.1 Textures of the particle's material

For the Surface particle's materials it is required to have a diffuse texture (normally with an alpha-channel). In the Mapping > Coordinates menu choose the UV option. Make sure that the emitter's mesh has a UV layer.

For the Halo particle's materials it is possible to use a Blend texture with a Linear gradient. In the Mapping > Coordinates menu choose the Strand / Particle option. It is required to enable Ramp on a texture. Up to 4 gradient control points are supported.

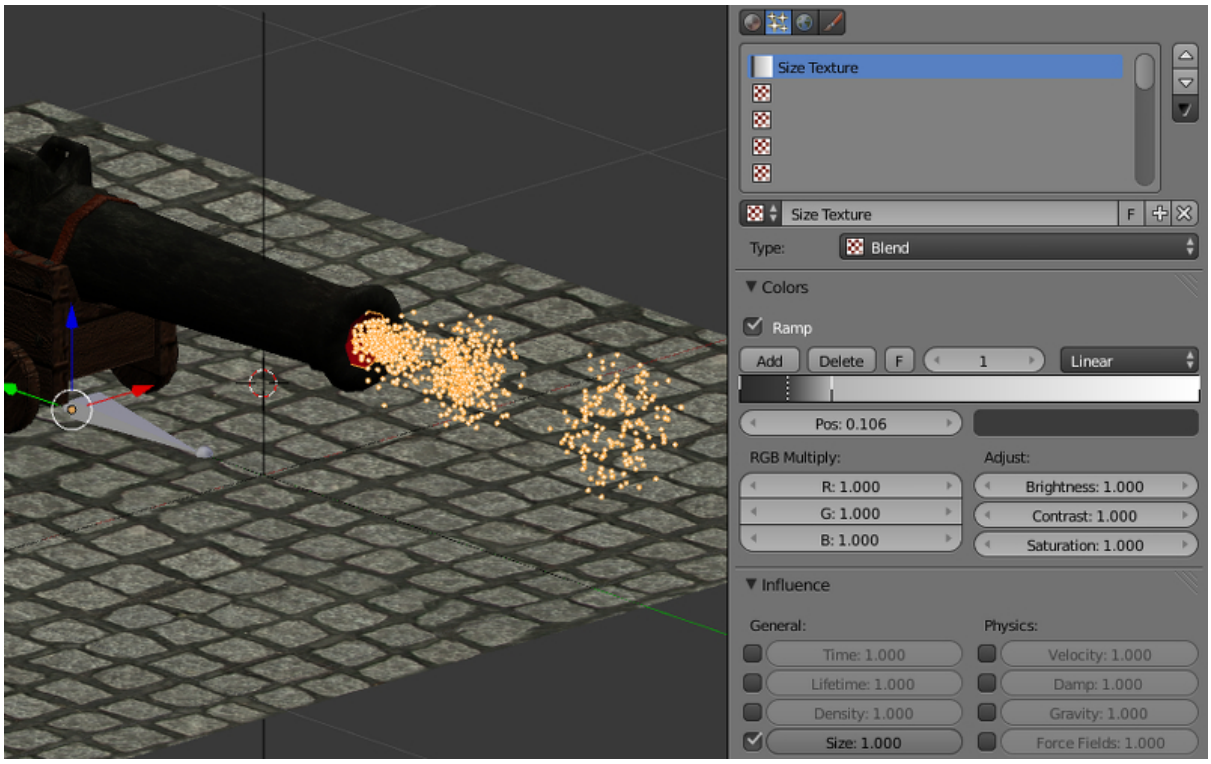


14.3.2 Textures of particle systems

Textures can also be used for setting up the behaviour of particle systems. Unlike textures for particle materials such textures belong to the particle system datablock, not to the material datablock. To create a texture for the particle system it is required to go from the Particles tab to the Textures tab and then to click the New button.

The only supported type of textures is Blend with a Linear gradient. Ramp should be enabled on the texture. Up to 4 gradient control points are supported.

On the Influence panel choose the parameter which is influenced by the texture. At the moment the only supported parameter is Size.



The result of using gradient textures on the particle material and the particle system:

The original model was taken from here

CHAPTER 15

Particle System for Instancing Objects

A particle system can be used to create multiple object copies (so called instancing).

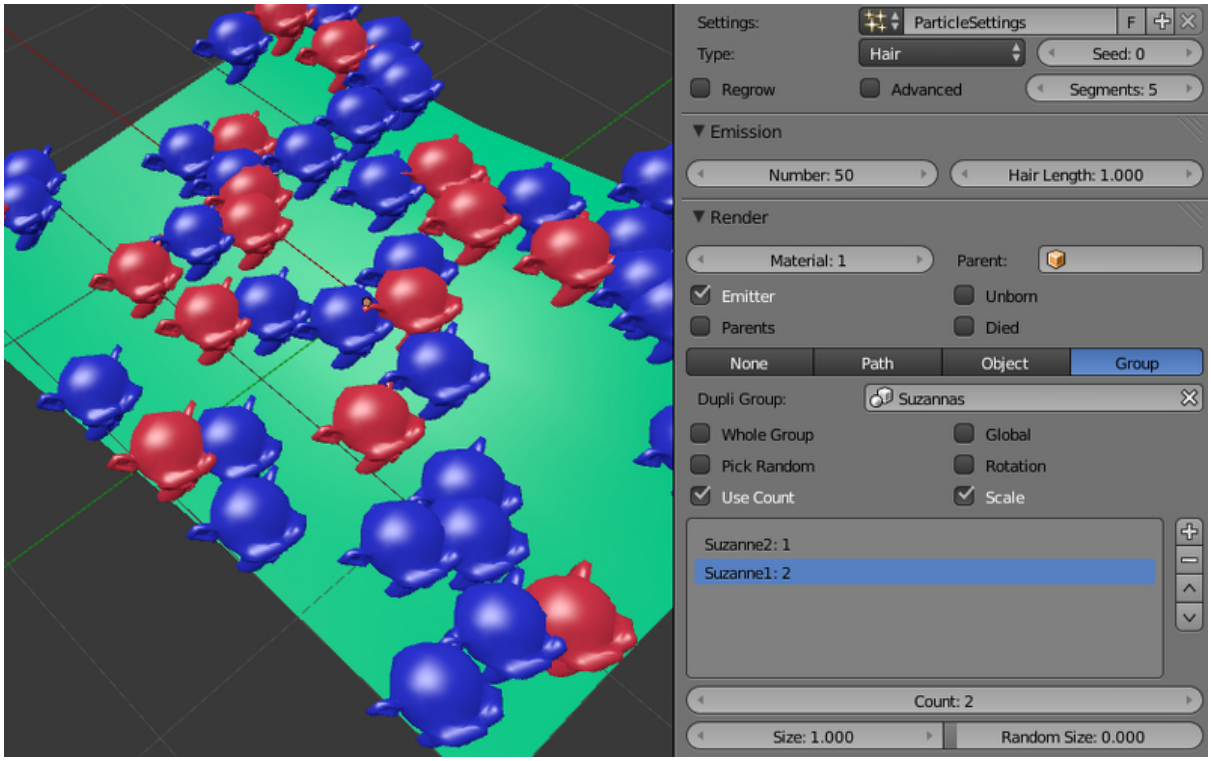
15.1 Particle System Setup

Activation

1. Create a particle system of the Hair type on the emitter.
2. On the Render panel select the Object (or the Group) rendering type.
3. In the Dupli Object field (or in the Dupli Group field) select the object (or the object group) for instancing. Both local and linked objects (or groups) are supported.

Recommended additional settings

1. For correct size displaying set the Emission > Hair Length and Render > Size parameters to 1.0.
2. For correct orientation temporarily enable the Advanced option, activate the Rotation panel and select None in the Initial Orientation menu. Disable the Advanced option. It is also recommended to enable the Render > Rotation option.



Setup

Render > Use Count The option is available for groups of particle objects. When enabled the interface for setting the relative number of objects in a group is displayed. The engine doesn't reproduce the exact location of objects of set types.

Blend4Web > Random location and size The option enables randomization for the location and the size of the objects. If enabled the engine generates random coordinates and size (limited to the ±25% range) for the particle objects. If it is disabled then the current coordinates and sizes of the particle objects are exported and used. Enabled by default.

Blend4Web > Initial random rotation The option enables random rotation of the objects relative to the axis which is defined by the Rotation type parameter. If enabled the engine generates random rotation angles for the particle objects. If disabled the zero rotation angle is used. Enabled by default.

Blend4Web > Rotation type

The axis for randomly turning the object (the option is available when the Blend4Web > Initial

- Z axis - the objects are turned randomly around the vertical Z axis
- Random axis - the objects are turned randomly around a random axis

The default is Z axis

Blend4Web > Rotation strength

The coefficient defining the range of random rotation angles - counting from the direction toward

- Rotation strength = 1 - the angles will lie within the $[-\pi, \pi]$ range
- Rotation strength = 0.5 - the angles will lie within the $[-0.5 \cdot \pi, 0.5 \cdot \pi]$ range
- Rotation strength = 0.1 - the angles will lie within the $[-0.1 \cdot \pi, 0.1 \cdot \pi]$ range

The default value is 1.

Blend4Web > Billboard Enables billboarding for particles. Disabled by default.

Blend4Web > Billboard type

Billboarding type (the option is available when the Blend4Web > Billboard option is set). 3 typ

- Basic - simple one-sided billboarding: particles will be turned with their front side to the observer
- Random - random two-sided billboarding: particles will be more often turned with their front or rear side to the observer and less often with their side; also there will be a small random turn; this model was created specially for instancing grass
- Jittered - one-sided billboarding with particles wavering along the plane which is turned to the observer; this model was created specially for instancing tree leaves

The default is Basic.

Blend4Web > Jitter amplitude The coefficient of the particle oscillation amplitude (the option is available when the Jittered type is selected from the Blend4Web > Billboard type menu). The bigger this parameter is the bigger is the oscillation amplitude. The default value is 0.

Blend4Web > Jitter frequency The particle oscillation frequency in hertz (the option is available when the Jittered type is selected from the Blend4Web > Billboard type menu). The default value is 0.

Blend4Web > Billboard geometry

Billboard rotation type (the option is available when the Blend4Web > Billboard checkbox is se

- Spherical - spherical billboarding i.e. particles are fully oriented to the observer and their rotation is unlimited
- Cylindrical - cylindrical billboarding i.e. particles are rotating only around the vertical Z axis.

The default is Spherical.

Blend4Web > Dynamic Grass The option enables the dynamic grass rendering mode. Disabled by default.

Blend4Web > Wind bending

Inheriting the wind bending settings by the particles:

- Parent - inherited from the emitter
- Instance - inherited from the particle object itself

The default is Parent.

Blend4Web > Shadows

Inheriting the shadow settings by particles:

- Parent - inherited from the emitter
- Instance - inherited from the particle object itself

The default is Parent.

Blend4Web > Reflection

Inheriting the reflection settings by particles:

- Parent - inherited from the emitter
- Instance - inherited from the particle object itself

The default is Parent.

Blend4Web > Vertex color

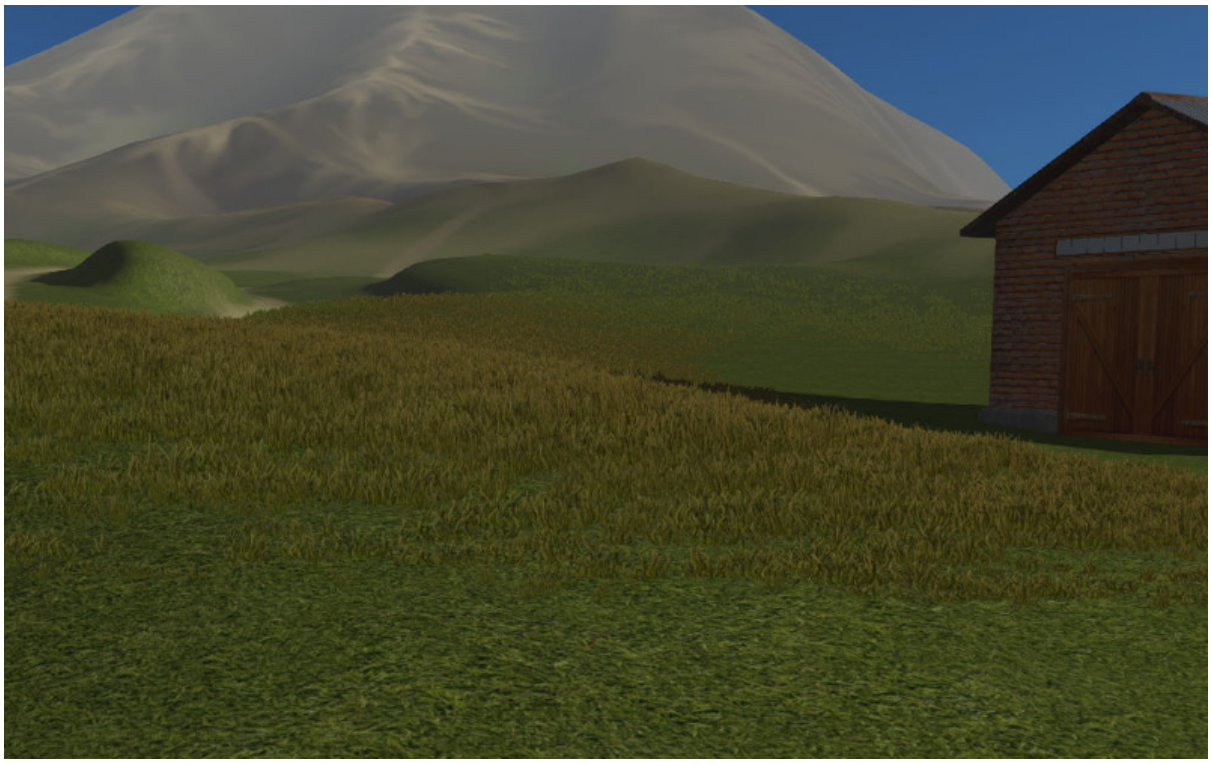
Inheriting the vertex color from the emitter. Contains 2 fields:

- from - the emitter's existing vertex color name
- to - the particle's existing vertex color name

There is no inheritance by default.

15.2 Grass

Instancing of objects can be used for visualizing vast grass. In this case grass is rendered near the camera when it moves through the landscape.



Activation

1. On a separate plane object create a particle system for object instancing. Enable the Blend4Web > Dynamic Grass option.
2. On the supposed landscape material enable the Blend4Web > Terrain dynamic grass option.

Setup

It is recommended to create a few planes (for example 3) with sizes corresponding to the desired grass cascades (e.g. 100, 150 and 250 meters).

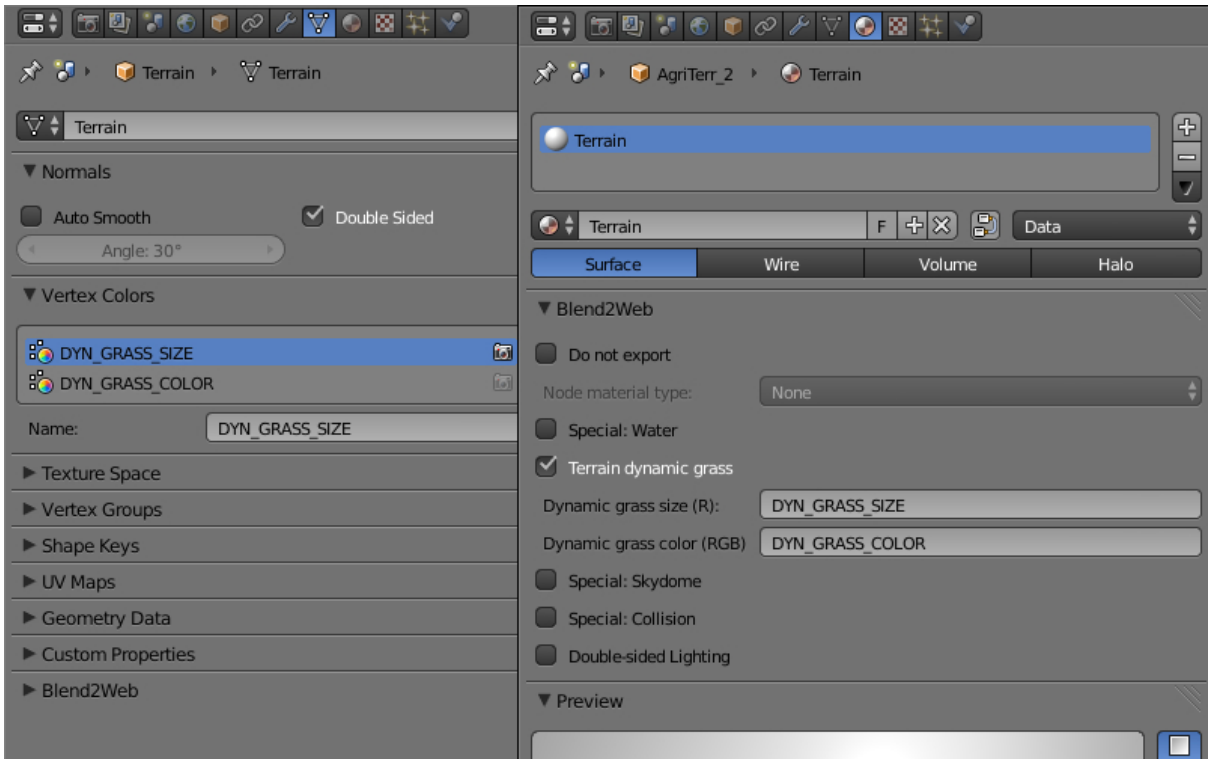
On the landscape material the following text fields get active when the Blend4Web > Terrain dynamic grass option is enabled:

Dynamic grass size (R) Vertex color layer name of the landscape mesh which is intended for modifying the grass size. The size (i.e. height) of the grass is defined by gray tints - the brighter color the is the higher is the grass.

Dynamic grass color (RGB) Name of the landscape mesh's vertex color layer which is intended for grass tinting. The vertex color is multiplied by the grass material color. The Influence > Blend parameter for the grass material's diffuse texture should have the Multiply value.

Vertex color layers with such names should exist in the landscape mesh.

It is also recommended to disable rendering of the emitter (the Render > Emitter option).



15.3 Tree Leaves

Instancing suits the rendering of tree leaves well and allows to get a better level of detail.

Activation

Performed as described in the Particle system setup -> Activation section (see above). In this case the tree is the emitter and the leaves and the small branches are the particles.

Additionally the following operations should be performed for the emitter:

- create a vertex group which includes vertices on which the particles will be placed
- create a vertex color layer for the wind bending parameters of the tree and the leaves
- create a vertex color layer to be inherited by the particles (for example it can be used for tinting the particles)

Setup

1. Random rotation settings

If the Blend4Web > Initial random rotation checkbox is enabled then it is recommended to select the vertical axis for random rotation - Z axis (the Blend4Web > Rotation type menu). The Blend4Web > Rotation strength value can be set too.

2. Billboarding settings

It is recommended to enable billboarding, to set its type as Jittered (the Blend4Web > Billboard type menu) and to make it spherical - Spherical (the Blend4Web > Billboard geometry menu). The Blend4Web > Jitter amplitude and Blend4Web > Jitter frequency values can be set too.

3. Particles location settings

It is recommended to select the Verts value from the Emission > Emit From menu, and to select in the Vertex Group > Density field the emitter's vertex group which defines the placing of particles. Also the Blend4Web > Random location and size checkbox shoud be disabled.

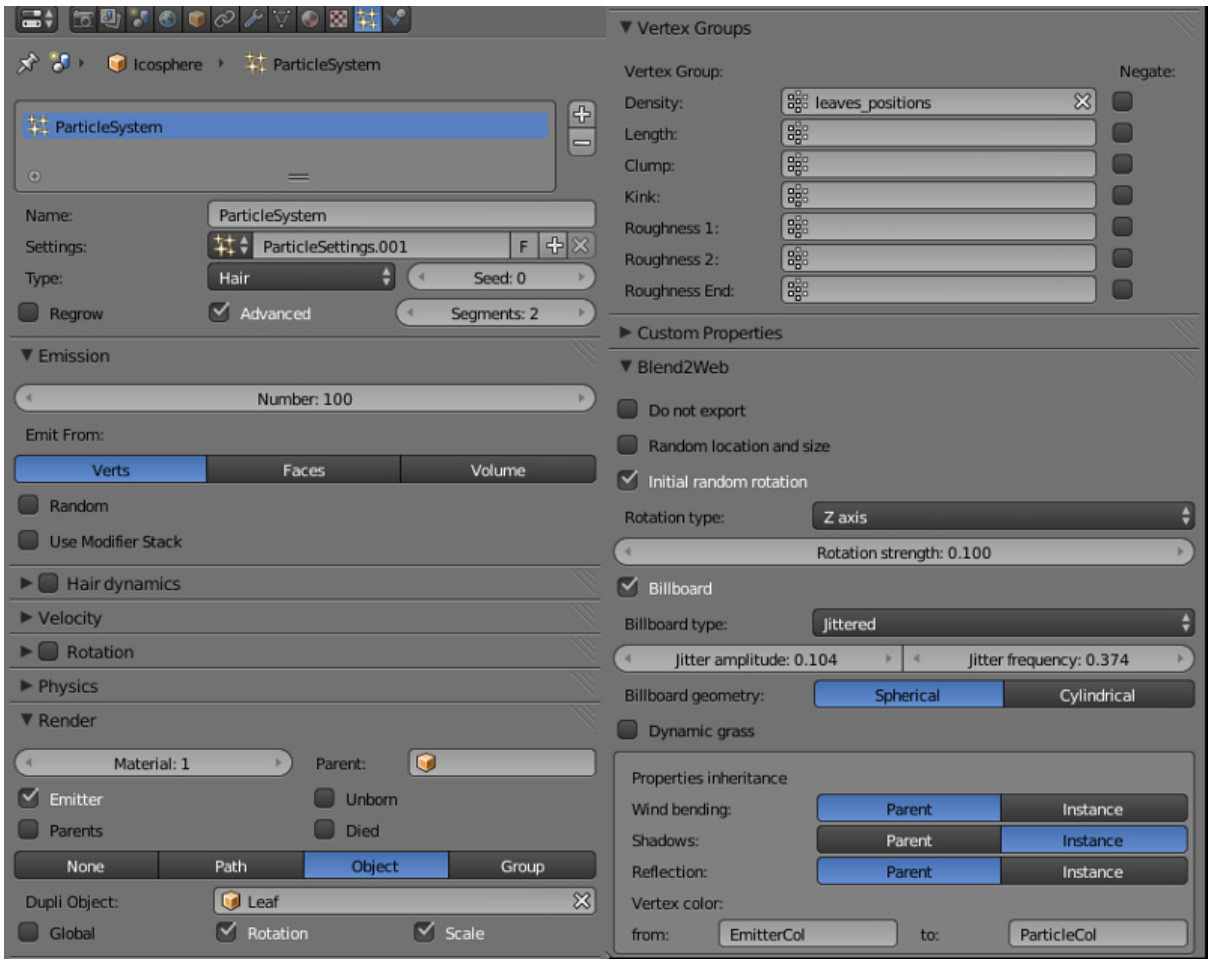
4. Wind Bending settings

It is recommended to enable inheritance settings from the emitter - select the Parent in the Blend4Web > Wind bending menu. Then for the emitter on the Object panel enable the Blend4Web > Wind bending checkbox and setup the bending parameters. For a tree it is enough to specify the Blend4Web > Main Bending > Angle and Blend4Web > Main Bending > Frequency parameters and also a vertex color name for bending in the Blend4Web > Main Bending > Main stiffness field.

5. Vertex color inheritance settings

For the emitter's vertex color to be inherited it is required to specify the emitter's vertex color name and the particle's vertex color name respectively in the Blend4Web > Vertex Color > from and Blend4Web > Vertex Color > to fields. The color of the emitter's vertex that is closest to the particle and specified in the from field will be copied and propagated into the to particle's vertex color layer, as a result.

The resulting vertex color layer with the Blend4Web > Vertex Color > to name may be used in the particle's node material for its tinting or other effects.



CHAPTER 16

Animation

In general animation is changing the object's parameters in time. The engine supports the following types of animation:

- Object animation is the moving an object as a whole. The parameters that can be animated are the center coordinates (Location), the rotation's quaternion (Rotation in the Quaternion(WXYZ) mode) and scaling (Scale).
- Skeletal animation, i.e. object deformation using bones. Animation of a standalone armature object is also supported (for parenting to bones).
- Vertex animation. An object's deformations can be recorded as frames and then reproduced in the engine.
- Audio sources parametrization. Speaker's Volume and Pitch can be animated.
- Wind bending - a procedural animation. Described separately.
- Particle emission. Described in the corresponding section.

16.1 Animation Control

There are two ways to control animation in the engine:

1. Automatically, using the Animation: Use default and Animation: Cyclic checkboxes in the object's properties. In this case an appropriate animation method will be chosen by the engine and the object's animation playback will start just after a scene is loaded. In case of skeletal animation the action which is assigned to the object in the Action Editor window is played by default.
2. In an application via API using the animation module methods.

It's useful to use the Animation interface for tweaking animation. This is covered in the corresponding section.

16.2 Object Animation

Animation keyframes can be added for an object motion in Blender and then reproduced in the engine.

The following keyframe types are supported:

- Location
- Rotation – the Quaternion(WXYZ) mode is required.
- Scale – for correct results the scale factor should be the same along all 3 axes.
- LocRot – a combination of Location and Rotation.
- LocScale – a combination of Location and Scale.
- LocRotScale – a combination of Location, Rotation and Scale.
- RotScale – a combination of Rotation and Scale.

If an object-mesh is animated it is required to enable the Do not batch checkbox under the object properties tab.

16.3 Skinning and Skeletal Animation

For skeletal animation both a mesh object and an armature object are needed. The four steps should be carried out:

1. Create the object's "skeleton" in the armature object.
2. Assign vertex groups in the mesh object and link them to the bones. This can be performed by weight painting for example.
3. Animate the bones in the pose mode of the armature object. The same keyframe types can be used as for the object animation.
4. When inverse kinematics (IK) or other nontrivial structures are used, an additional step is required to bake the animations (Action datablocks in Blender). Baking can be performed using the B4W Animation Bake interface located on the Blend4Web panel.

Baking is performed with the armature object selected. Elements of the B4W Animation Bake interface:

- Clean keyframes – optimize the animation keyframes after baking. In case of incorrect results it's recommended to turn this option off.
- window with the list of actions being baked – bake only the actions which are listed, else - bake all the actions possible.
- Name – the current action name from the list of actions being baked.
- Bake – perform baking. If the process is completed successfully, actions with names of NAME_B4W_BAKED type appear in the scene. These actions can be assigned to the armature object and played back in the engine. It's worth noting that appropriate functioning of such actions in Blender is not garanteed, although the Cons Mute/Cons Unmute interface can help in some cases.
- Cons Mute/Cons Unmute – switch bone constraints off/on. The tool can be used for testing the baked actions.

16.4 Vertex Animation

Allows to record any geometry changes of a mesh object. Note that every vertex animation frame counts as a mesh. It's not recommended to make a long animation for a high-poly mesh, as it can increase the size of the source and exported files significantly and can also slow down the work of the engine.

A special tool is used for baking vertex animation - B4W Vertex Anim Baker - located on the Blend4Web tools panel.

16.5 Audio Source Parametrization

In addition the following animation key types are supported for the speaker objects:

- Volume
- Pitch

Audio sources parametering in essence follows object animation.

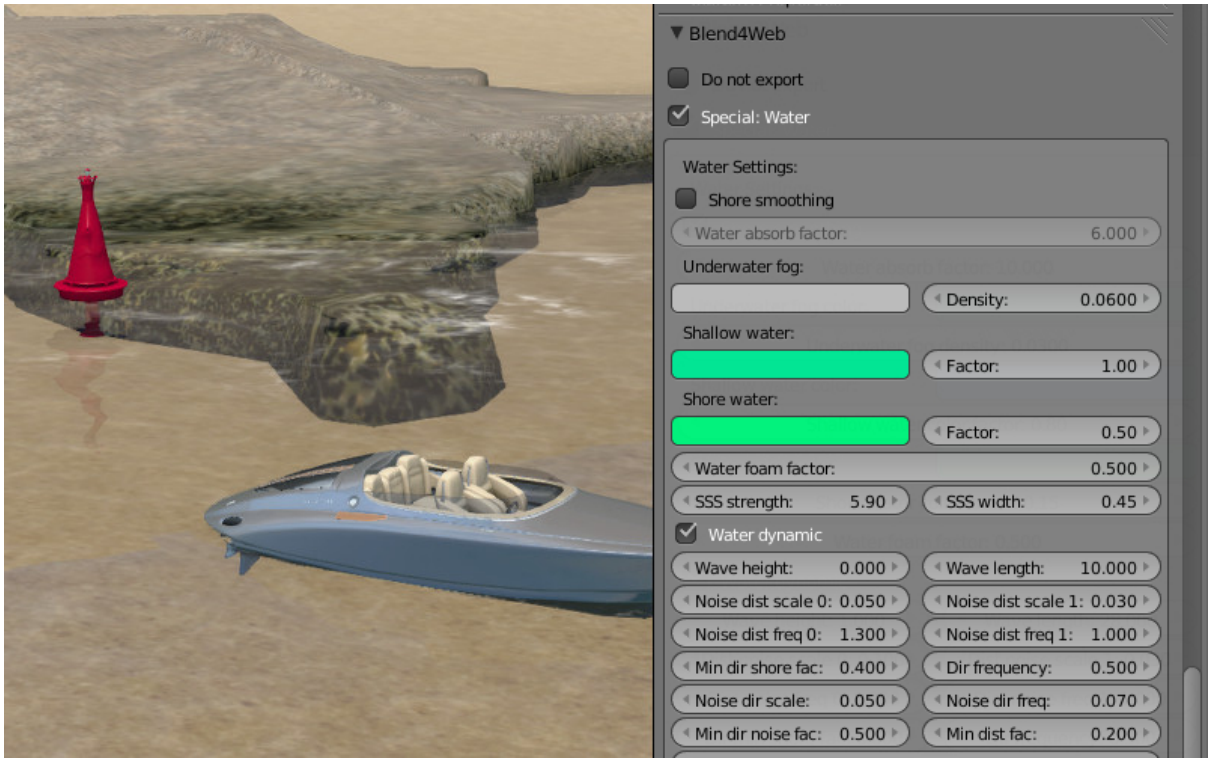
CHAPTER 17

Outdoor Rendering

17.1 Water

17.1.1 Activation

For the supposed water material enable the Blend4Web > Special: Water option under the Material tab.



17.1.2 Basic settings

Transparency It is recommended to enable the gradient transparency Game Settings > Alpha Blend and to tweak the Transparency > Alpha value.

Lighting parameters Lighting parameters for the water material can be set up as described in the Lighting Parameters section.

17.1.3 Waves dynamics

Waves simulation is carried out by normal maps with animated UVs (from 0 up to 4 pieces). For normal textures the only shared image is used - the textures differs only by the Mapping > Size and Blend4Web > UV translation velocity parameters. The water mesh should have a UV layer.

17.1.4 Surface wetting

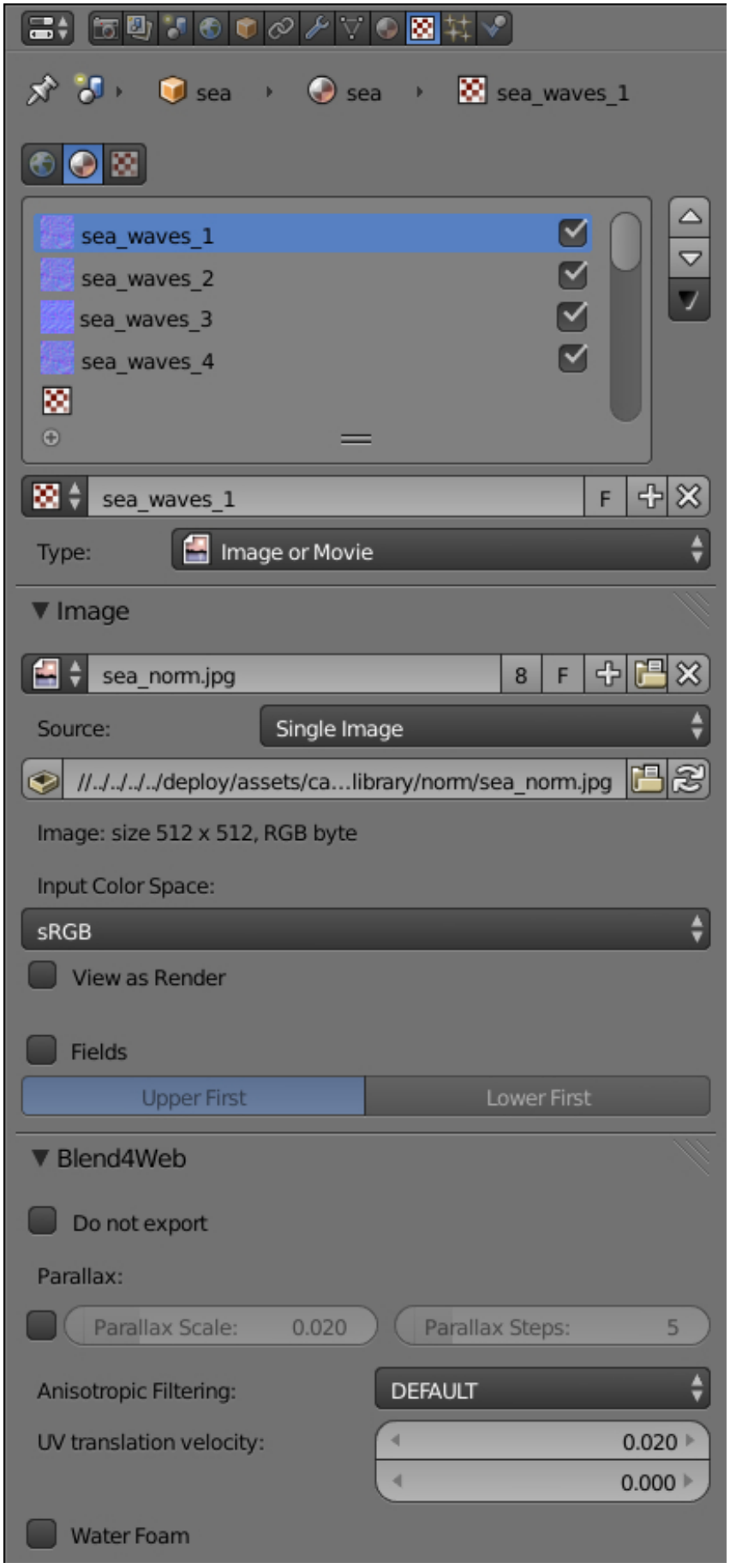
Is carried out automatically. To turn the effect on enable the Wettable checkbox on the needed materials.

17.1.5 Reflection and Fresnel effect

For the water material both static and dynamic reflection is supported as well as the Fresnel effect. See the Reflection section.

17.1.6 Shoreline smoothing

Blend4Web > Water Settings > Shore smoothing Enable smoothing.


sea
sea
sea_waves_1
sea_waves_1
sea_waves_2
sea_waves_3
sea_waves_4
sea_waves_1
F
Type:
Image or Movie
Image
sea_norm.jpg
8
F
Source:
Single Image
//../../../../deploy/assets/ca...library/norm/sea_norm.jpg
Image: size 512 x 512, RGB byte
Input Color Space:
sRGB
View as Render
Fields
Upper First
Lower First
Blend4Web
Do not export
Parallax:
Parallax Scale:
0.020
Parallax Steps:
5
Anisotropic Filtering:
DEFAULT
UV translation velocity:
0.020
0.000
Water Foam

Blend4Web > Water Settings > Water absorb factor Light absorption coefficient for the water. The higher it is the more transparent isthe water.

In the low quality mode an alpha map can be used instead of this option.

17.1.7 Color gradient

For color gradient the water material should have a texture with the Blend4Web > Shore distance map option enabled which can be generated using the script for baking shoreline parameters.

Blend4Web > Water Settings > Shallow water color Shallow water color.

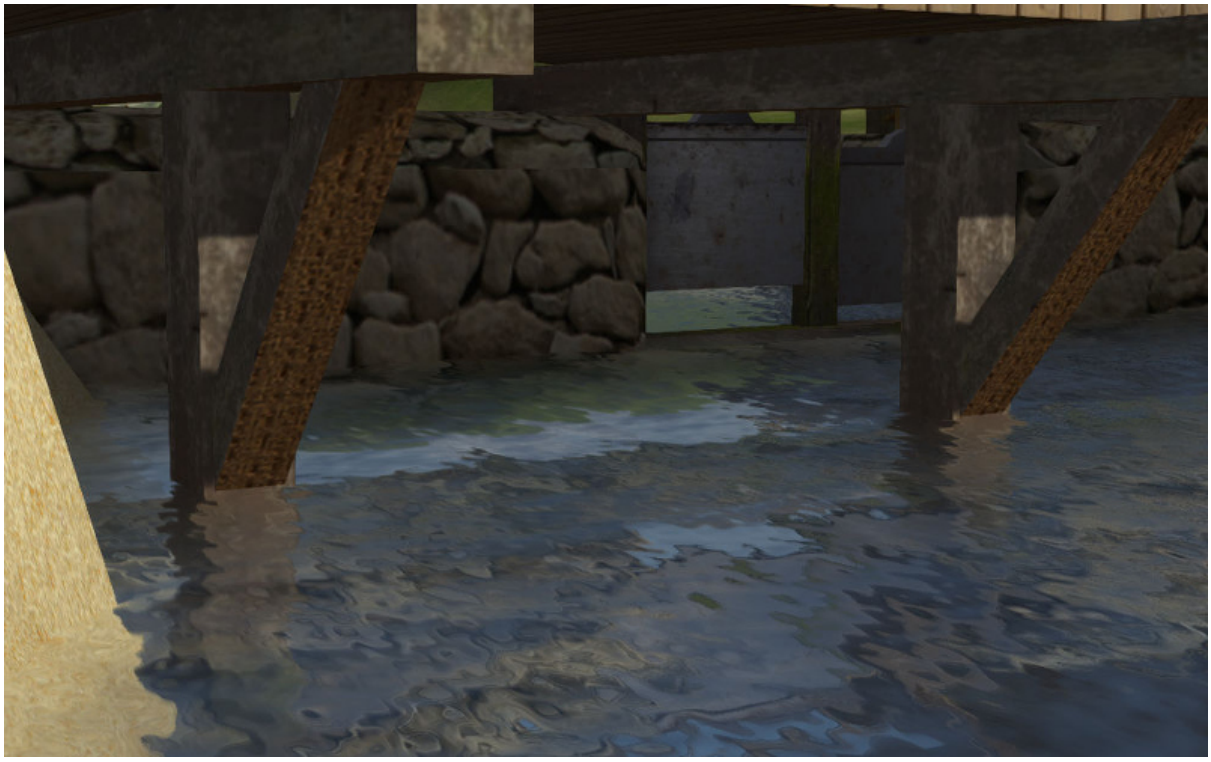
Blend4Web > Water Settings > Shallow water color factor Shallow water color mixing factor.

Blend4Web > Water Settings > Shore water color Water color just at the shore line.

Blend4Web > Water Settings > Shore water color factor Factor for mixing water color just near the shoreline.

17.1.8 Refraction

Under the Scene tab enable the Blend4Web > Render refractions checkbox.



17.1.9 Foam

Activation

For creating foam add two diffuse textures into the water material slots. For these textures enable the Blend4Web > Water Foam option.

Setting up the textures

Influence > Color Texture color influence factor. The default value is 1.0.

Blend4Web > UV Frequency Oscillation frequency of the animated UV coordinates. The default value is (1.0, 1.0).

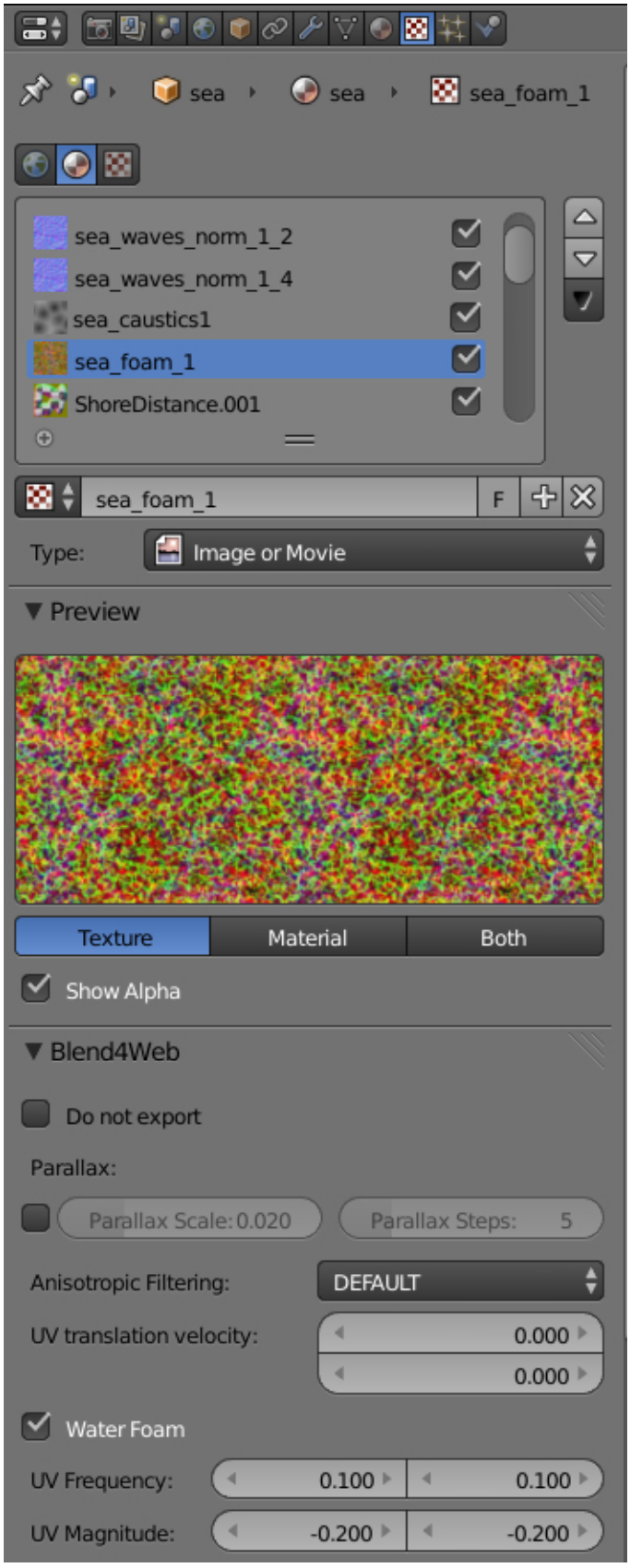
Blend4Web > UV Magnitude Oscillation amplitude of the animated UV coordinates. The default value is (1.0, 1.0).

Setting up the material

Blend4Web > Water Settings > Water foam factor General influence factor for the foam. The default value is 0.5.

17.1.10 Caustics and chromatic aberration

To create the caustics effect add one Voronoi type texture to the water material slot.


sea
sea
sea_foam_1
sea_waves_norm_1_2
sea_waves_norm_1_4
sea_caustics1
sea_foam_1
ShoreDistance.001
sea_foam_1
F
Type:
Image or Movie
Preview
Texture
Material
Both
Show Alpha
Blend4Web
Do not export
Parallax:
Parallax Scale: 0.020
Parallax Steps: 5
Anisotropic Filtering:
DEFAULT
UV translation velocity:
0.000
0.000
Water Foam
UV Frequency:
0.100
0.100
UV Magnitude:
-0.200
-0.200

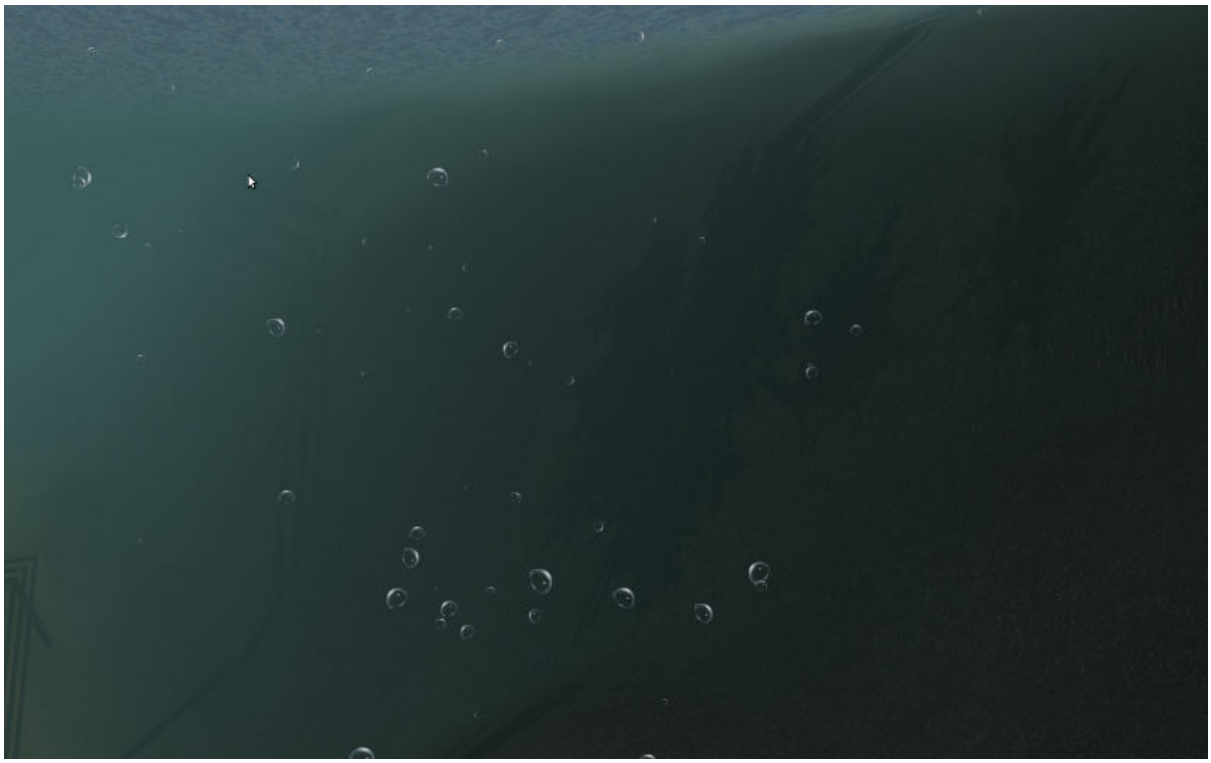
Setting up

Voronoi > Coloring: Intensity Caustics influence factor. The default value is 1.0.

Voronoi > Noise: Size Cell size for the procedural texture. The default value is 0.25.

Blend4Web > UV translation velocity Texture coordinates animation speed. The default value is (0.0, 0.0).

17.1.11 Underwater environment



Visibility settings ("fog")

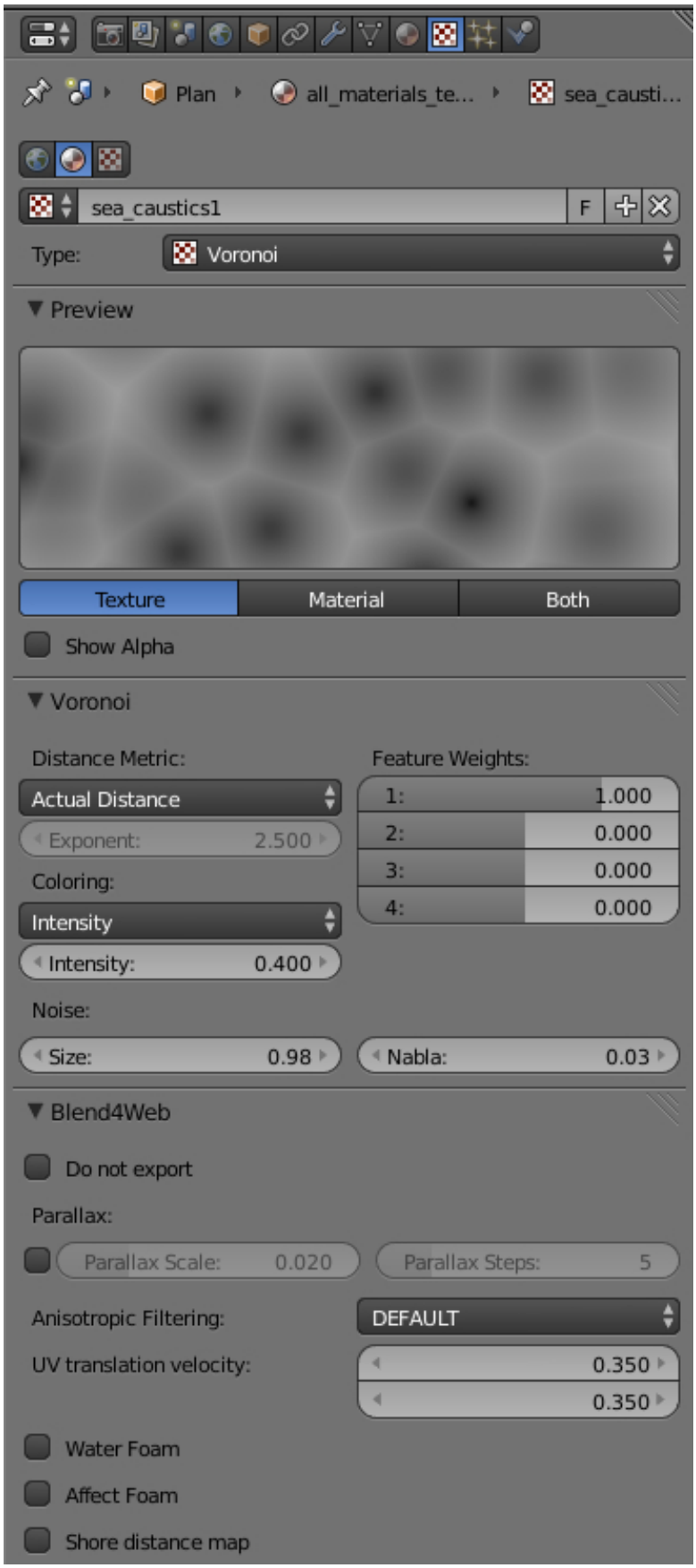
Blend4Web > Water Settings > Underwater fog density Exponential factor which affects the density and visibility distance. The default value is 0.06.

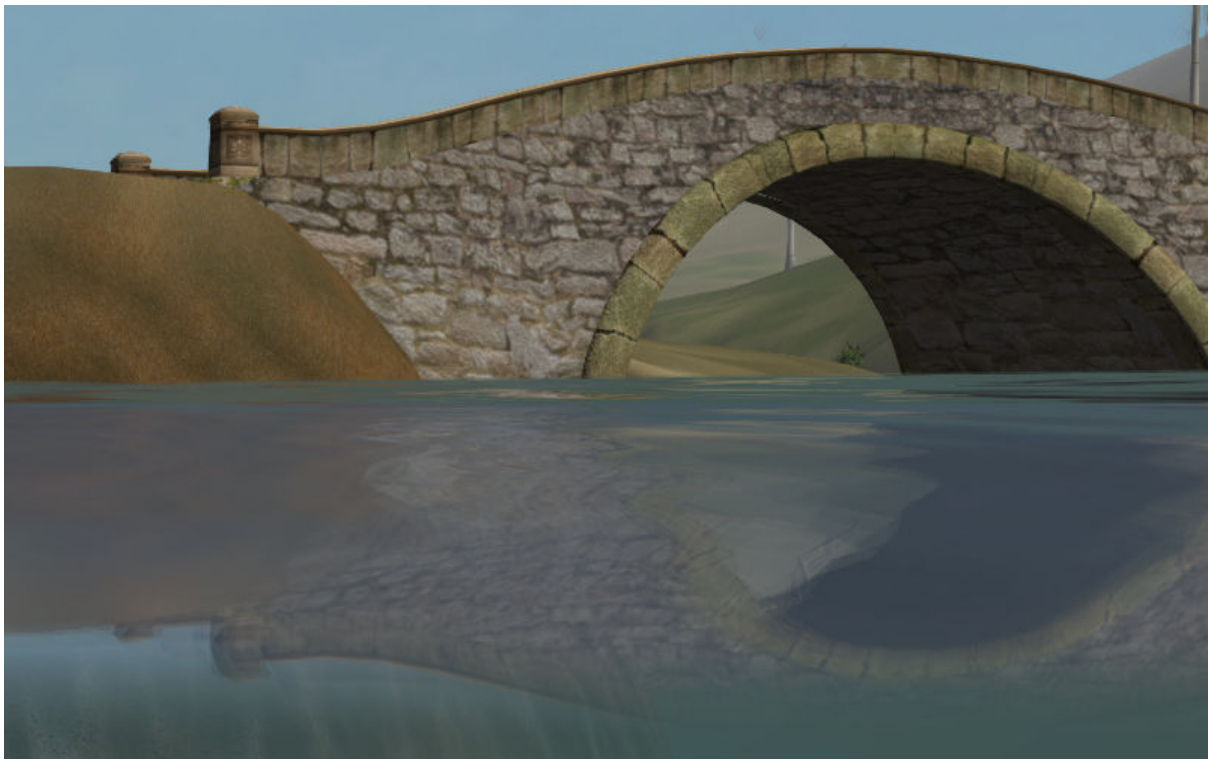
Blend4Web > Water Settings > Underwater fog color Fog color. The default value is (0.5, 0.5, 0.5) (gray).

The god rays effect settings are also applied.

17.1.12 Boundary between environments

Disable the Game Settings > Backface Culling option.


Plan
all_materials_te...
sea_causti...
sea_caustics1
F
Type:
Voronoi
Preview
Texture
Material
Both
Show Alpha
Voronoi
Distance Metric:
Actual Distance
Exponent: 2.500
Coloring:
Intensity
Intensity: 0.400
Feature Weights:
1: 1.000
2: 0.000
3: 0.000
4: 0.000
Noise:
Size: 0.98
Nabla: 0.03
Blend4Web
Do not export
Parallax:
Parallax Scale: 0.020
Parallax Steps: 5
Anisotropic Filtering:
DEFAULT
UV translation velocity:
0.350
0.350
Water Foam
Affect Foam
Shore distance map

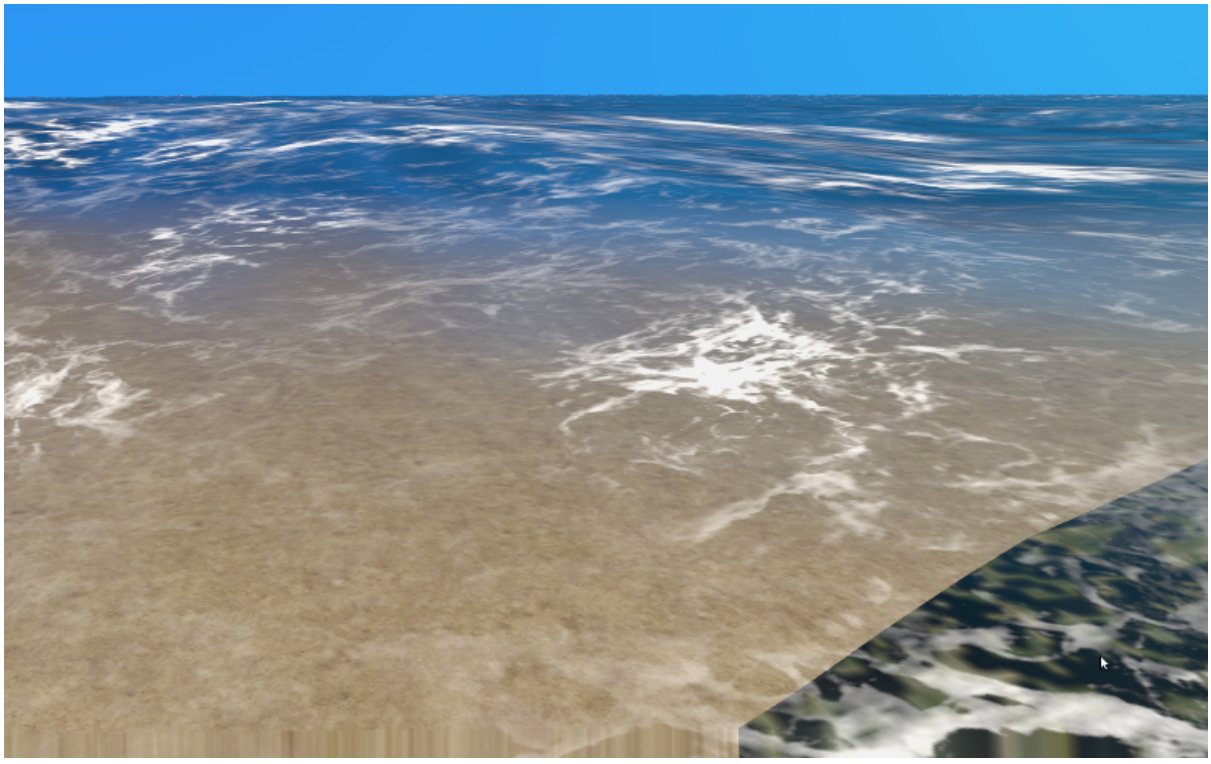


17.1.13 Volumetric waves

Activation

Blend4Web > Water Settings > Water Dynamic

Enable volumetric waves.



Setting up

Blend4Web > Water Settings > Wave height Wave height. The default value is 0.0.

Blend4Web > Water Settings > Wave length Wave length. The default value is 10.0.

Blend4Web > Water Settings > Dist noise scale 0 Size of the first component of the open water waves.

Blend4Web > Water Settings > Dist noise scale 1 Size of the second component of the open water waves.

Blend4Web > Water Settings > Dist noise freq 0 Frequency of the first component of the open water waves.

Blend4Web > Water Settings > Dist noise freq 1 Frequency of the second component of the open water waves.

Blend4Web > Water Settings > Dir min shore fac Minimum height decrease coefficient of the shore waves.

Blend4Web > Water Settings > Dir frequency Frequency of the rolling of the shore waves.

Blend4Web > Water Settings > Dir noise scale Noise size for the shore waves.

Blend4Web > Water Settings > Dir noise freq Noise frequency for the shore waves.

Blend4Web > Water Settings > Dir min noise fac Noise minimum for the shore waves.

Blend4Web > Water Settings > Dist min fac Minimum coefficient of mixing for open water waves.

Blend4Web > Water Settings > Waves horizontal factor Coefficient that shows how much the shore waves are shifted in the shoreline direction.

17.1.14 Generated surface settings

Blend4Web > Water Settings > Generate mesh Enable generated surface.

Blend4Web > Water Settings > Number of cascades Number of cascades in the generated surface.

Blend4Web > Water Settings > Detailed distance Maximum distance from camera to the last cascades edge.

Creating a texture with shoreline parameters.

On the tools panel (hotkey “T”) under the Blend4Web tab open the B4W Shore Distance Baker panel. Set the parameters: maximum distance to shore (Maximum Distance) and the resulting texture size (Texture Size). Select a landscape object (or multiple objects) first, and then - a water object. Click the Bake Shore Distance button.

Depending on the texture size and the number of vertices in the processed meshes the execution time of the script may vary from a fraction of a second up to several minutes. Make sure that the texture named ShoreDistance is created for the water mesh.

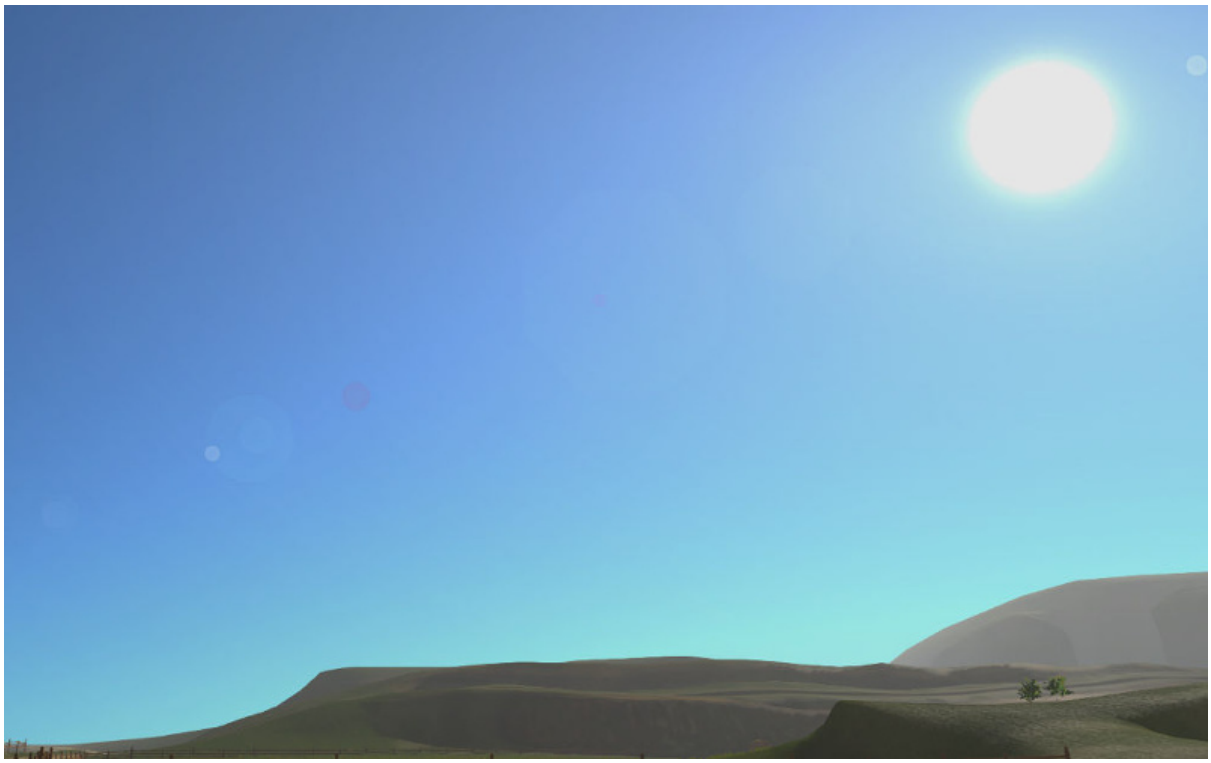
Upon script execution some system properties are saved in the water material. Therefore the scene must be saved after the script has finished working.

17.2 Atmosphere

17.2.1 Scattering

Enable the Sky Settings > Procedural skydome option under the World tab. If a static skydome texture is being used at the same time, it will be replaced.

Note: Also, a procedural sky texture can be used to imitate environment lighting for objects, similar to the static skydome texture. To do this, enable the Sky Settings > Use as environment lighting and Environment Lighting > Sky Texture checkboxes. If the world texture for environmet lighting already exists, it will be replaced.



Supported settings:

Sky Settings > Sky color Base sky color. The default value is (0.087, 0.255, 0.6) (blue).

Sky Settings > Rayleigh brightness Rayleigh scattering brightness (i.e. scattering on small particles). The default value is 3.3.

Sky Settings > Mie brightness Mie scattering brightness (i.e. scattering on large particles). The default value is 0.1.

Sky Settings > Spot brightness Sun spot brightness. The default value is 20.0.

Sky Settings > Scatter strength Light scattering factor. The default value is 0.2.

Sky Settings > Rayleigh strength Rayleigh scattering factor. The default value is 0.2.

Sky Settings > Mie strength Mie scattering factor. The default value is 0.006.

Sky Settings > Rayleigh collection power Rayleigh scattering exponent. The default value is 0.35.

Sky Settings > Mie collection power Mie scattering exponent. The default value is 0.5.

Sky Settings > Mie distribution Mie scattering distribution. The default value is 0.4.

17.2.2 Fog

Can be set up under the World tab.

Blend4Web > Fog Settings > Fog density Exponential factor which affects density and the visibility distance. The default value is 0.0.

Blend4Web > Fog Settings > Fog color Fog color. The default value is (0.5, 0.5, 0.5) (gray).

When a dynamic skydome is used the fog color is defined by the sky color.

17.2.3 Time of day

Enable the Blend4Web > Dynamic intensity checkbox for the lamp.

Time of day can be set by applications via API. Particularly time of day can be set using the Lighting interface of the Scene viewer.

17.2.4 Stars

Stars setup is described in the Halo Materials section.

17.3 Wind

Wind strength and direction affect

- grass and tree leaves animation
- particle system dynamics
- water waves rolling frequency (at the moment only strength is taken into account)

17.3.1 Activation

Add a force field object of the Wind type.

17.3.2 Setting up

Direction Direction can be set by rotating the force field object.

Force Fields > Strength Wind strength. Located under the Physics tab. The default value is 1.0.

17.3.3 Grass and tree leaves animation

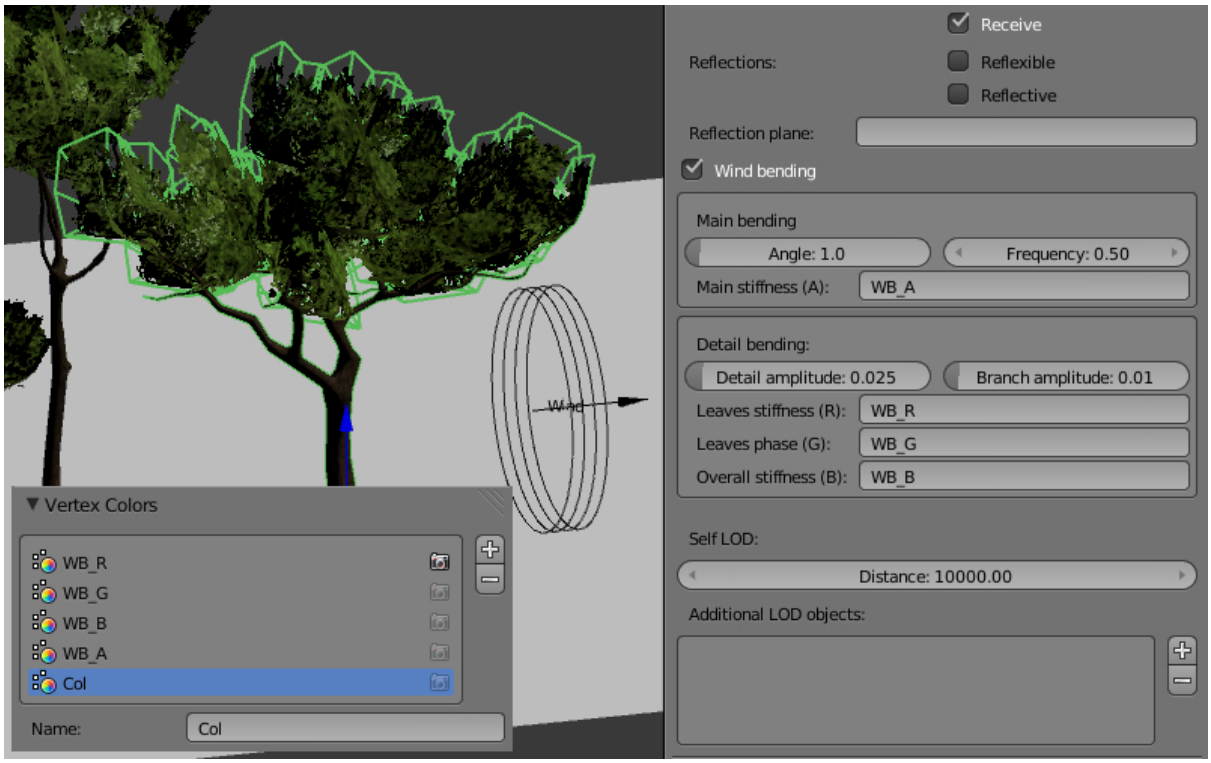
Authoring resources for grass rendering is described in the Grass section.

Activation

Enable the Blend4Web > Wind bending checkbox for the grass or tree object.

Setting up

The interface panel becomes visible after the activation of the Blend4Web > Wind bending checkbox.



Main bending > Angle Angle amplitude of the “main” deviation under the influence of wind (in degrees). The default value is 10.0.

Main bending > Frequency Frequency of the “main” deviation under the influence of wind. The default value is 0.25.

Main bending > Main stiffness (A) Text field for specifying the name of the vertex color layer which contains the information about the stiffness of the “main” deviation. Can be left empty.

Detail bending > Detail amplitude Angle amplitude of the “detail” deviation caused by the influence of wind (in degrees). The default value is 0.1.

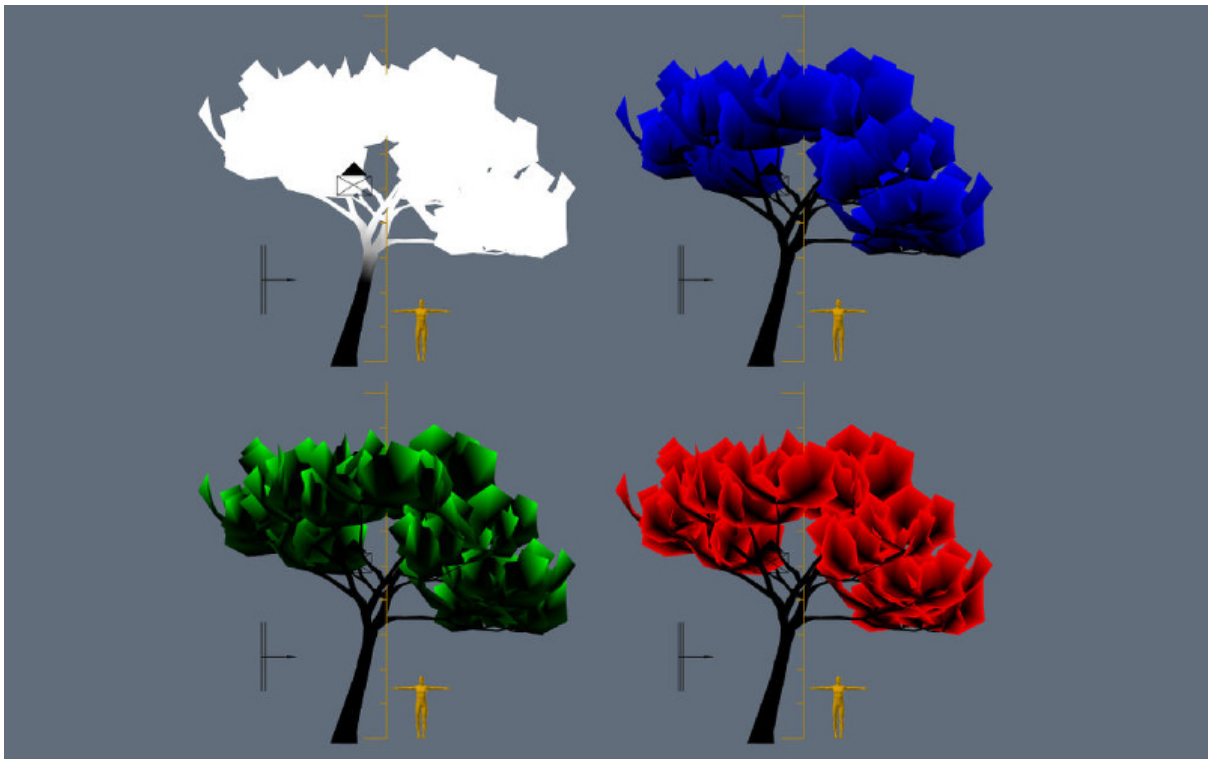
Detail bending > Branch amplitude Angle amplitude of the branch deviation caused by the influence of wind (in degrees). The default value is 0.3.

Detail bending > Leaves stiffness (R) Text field for specifying the name of the vertex color layer which contains the information about the stiffness of leaves. Can be left empty.

Detail bending > Leaves phase (G) Text field for specifying the name of the vertex color layer which contains the information about the phase of leaves deviation. Can be left empty.

Detail bending > Overall stiffness (B) Text field for specifying the name of the vertex color layer which contains the information about the overall stiffness of leaves. Can be left empty.

Vertex color layers should be present in the mesh if their names are specified.

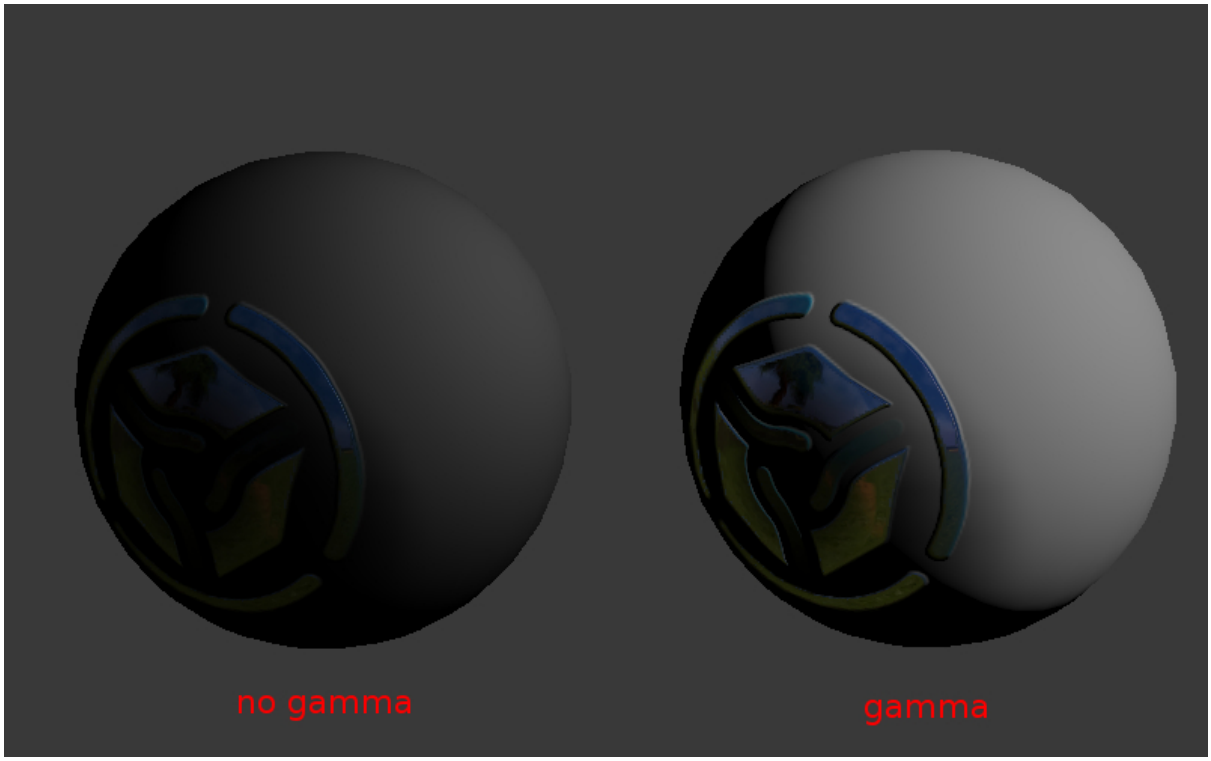


Gamma Correction and Alpha Compositing

18.1 Overview

The essence of gamma correction is packing the image brightness channel into 8 bits of information.

Graphics editors normally work in a non-linear color space where the darker components are encoded using more bits than the brighter ones. That means that a bigger RGB value corresponds to 0.5 of the real light intensity (a physical quantity called illuminance) - in a simplest case this value equals to 0.5 ^ (1/2.2) = 0.73.



Images are always stored in a non-linear space because if they were not, 8 bits of information would not be enough to encode the light intensity and it would result in a wrong rendering of the darker tints.

Web browsers work in a non-linear space.

When the Color Management > Display Device > sRGB option is enabled for a scene, Blender works in a linear space. Material colors and lamp settings correspond to physical values. For texture images (except normal maps) it is required to select the Image > Input Color Space > sRGB option. In this case an automatic image unpacking (sRGB -> Linear) is performed at the time of rendering.

Engines and renderers work in a linear space because this is the only correct way to represent light behavior in the real world. For example, the illuminance from two identical lamps exceeds the illuminance from one lamp exactly by two times.

Some values of illuminance:

Description	Illuminance, lux
Summer noon	17 000
Winter noon	5 000
Dull day	1 000
In a light room	100
Full moon by night	0.2
Moonless light	0.001

18.2 Human Vision, Monitors and Gamma Correction

While the human vision is non-linear (a human recognizes the darker light tints better than the brighter ones), the light coming into the eye still obeys the physical laws (see the lamps example).

In CRT monitors the brightness is dependent non-linearly upon the electric voltage applied to the monitor's input (the voltage itself is determined by the color channel value in the video memory). LCD monitors mimic the same characteristics. Nevertheless the light emitted by such monitors obeys the physical laws. For example the addition of a second light source to a virtual scene should cause the brightness to double (in the perfect case).

Therefore the perception characteristics of the human eye and the technical characteristics of monitors have a secondary significance for gamma correction.

18.3 Gamma

Used in the following simplified formula:

$$V_{out} = V_{in}^{\gamma}$$

$\gamma < 1$ - packing gamma, $\gamma > 1$ - unpacking gamma. In the simplest case 1/2.2 and 2.2 values are used respectively. Hereinafter the "packing" (Linear -> sRGB) and "unpacking" (sRGB -> Linear) terms are used instead of "gamma correction".

18.4 Gamma Correction in Node Materials

18.4.1 Nodes for colouring

Unpacking (sRGB -> Linear) is required when textures and vertex colors are used for colouring (not for masking). The texture node implements unpacking automatically. For vertex colors the unpacking should be performed explicitely with the SRGB_TO_LINEAR special node.

Note that the alpha channel of a texture node is not corrected. Its values are in the linear space.

18.4.2 Nodes for masking

Textures and vertex colors can be used as masks i.e. for mixing colors and for other mathematical operations. In such a case no transformations are required.

In case of a texture however there is a nuance: a texture node implements unpacking automatically. This results in necessity of the additional transformation back to the non-linear space, for which the LINEAR_TO_SRGB node is used.

18.4.3 Normal maps

No transformations are performed for normal maps.

18.4.4 Resulting table of correction in node materials

Use case	Correction
Texture for colouring	implemented automatically in the texture node (the alpha channel is not corrected)
Texture for masking	LINEAR_TO_SRGB
Vertex color for colouring	SRGB_TO_LINEAR
Vertex color for masking	not required
Normal map	not required

18.5 Alpha Compositing

18.5.1 Overview

Physically correct alpha compositing is performed according to the formula [source]:

$$C_o = C_a\alpha_a + C_b\alpha_b(1 - \alpha_a).$$

This formula differs from the classic mix operation (aka convex combination) because it has the α_b multiplier in the second summand. Therefore not only the α_a value of the source pixel should be known for alpha compositing, but also the α_b value of the pixel over which the rendering is performed.

In case of preliminary multiplication of the α values by the color channels (so called premultiplied alpha) the formula becomes as following:

$$C_o = C_a + C_b(1 - \alpha_a).$$

The last formula is used also to calculate the resulting α_o value:

$$\alpha_o = \alpha_a + \alpha_b(1 - \alpha_a).$$

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.

18.5.2 Implementation

The blending function used in Blend4Web is the following:

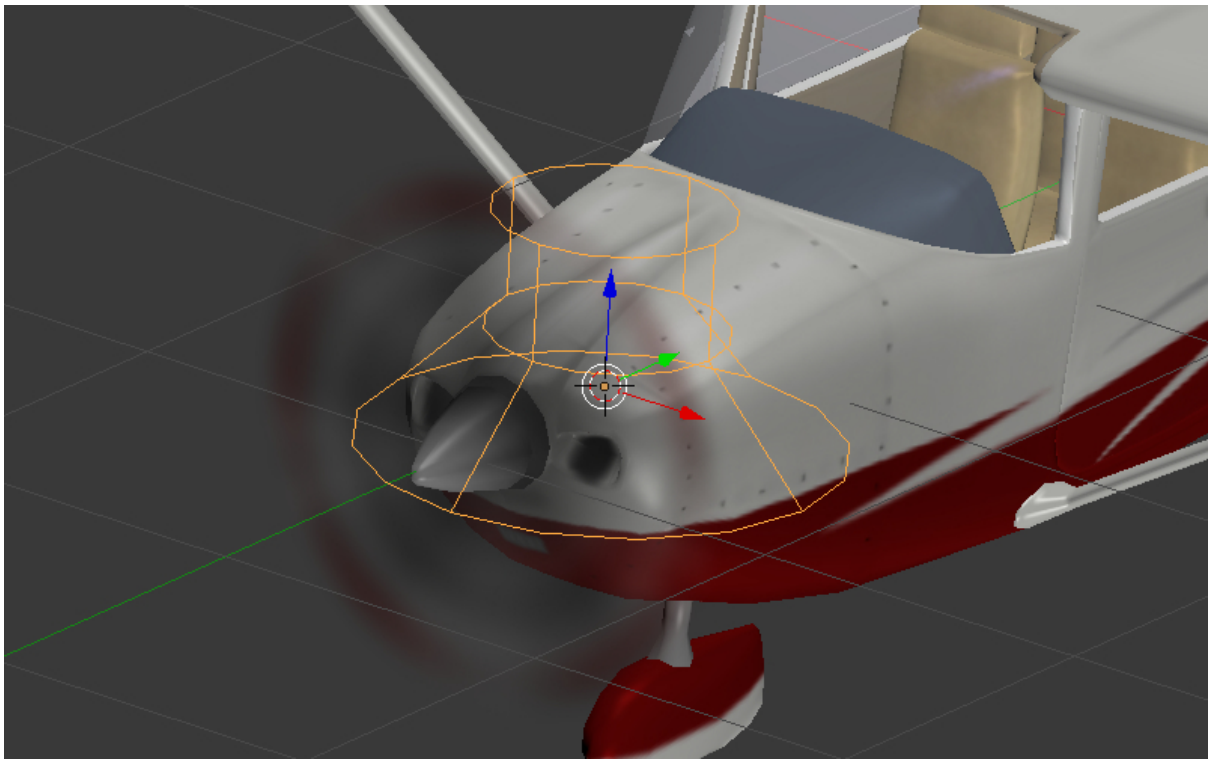
```
gl.blendFunc(gl.ONE, gl.ONE_MINUS_SRC_ALPHA);
```

WebGL context initialization is performed using the premultipliedAlpha = true parameter (that is the default value). Also multiplication of all the color channels by the α value is performed on the output of the shaders.

CHAPTER 19

Audio System

Audio sources are created in Blender. The standard Speaker object is used.



19.1 Audio Source Settings

Speaker parameters can be set up on the Properties panel under the Object Data tab.

The engine supports all the standard parameters. The engine-specific settings are located on the Blend4Web panel.

Speaker behavior The behavior of the audio source.

Positional — high-quality sound with spatial positioning and directivity (conicity). The Web Audio API is used for sound rendering. Playback performance of such sounds is the least and so use them only for short samples.

Background sound — high-quality omnidirectional sound without spatial positioning. The Web Audio API is used for sound rendering. It is more performant but is not effective for music.

Background music — used for music playback. It has maximum performance due to the use of the Audio HTML tag, but has minimum flexibility.

Disable doppler Ignore source's frequency shift upon its moving.

Cyclic play Loop the sound playback.

Delay Delay before sound playback starts.

Random delay Additional delay randomization. The resulting value is calculated according to the formula $Delay_{result} = Delay + Delay_{random} * Random_{[0-1]}$.

Random volume Additional volume randomization. The resulting value is calculated as for the delay.

Random pitch Additional randomization of the sound playback speed. The resulting value is calculated as for the delay.

Fade-in Fade-in time interval.

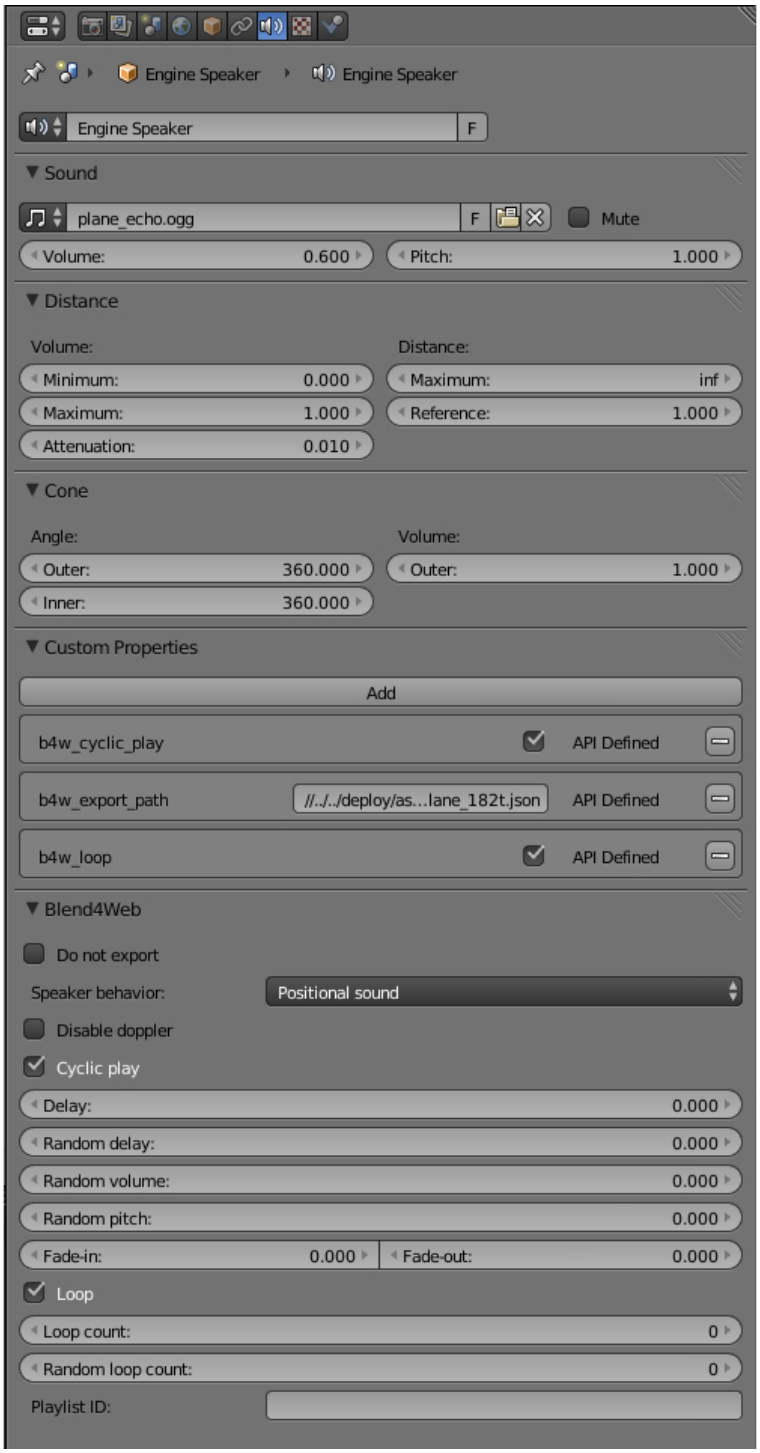
Fade-out Fade-out time interval.

Loop Loop the sound playback. Contrary to the Cyclic play option it guarantees a zero delay upon repeat. The option is available only for sound sources with Positional or Background sound behavior.

Loop count Not implemented.

Random loop count Not implemented.

Playlist ID Not implemented.


Engine Speaker
Engine Speaker
Engine Speaker
F
Sound
plane_echo.ogg
F
Mute
Volume: 0.600
Pitch: 1.000
Distance
Volume:
Minimum: 0.000
Maximum: 1.000
Attenuation: 0.010
Distance:
Maximum: inf
Reference: 1.000
Cone
Angle:
Outer: 360.000
Inner: 360.000
Volume:
Outer: 1.000
Custom Properties
Add
b4w_cyclic_play
API Defined
b4w_export_path
//../../deploy/as...lane_182t.json
API Defined
b4w_loop
API Defined
Blend4Web
Do not export
Speaker behavior:
Positional sound
Disable doppler
Cyclic play
Delay: 0.000
Random delay: 0.000
Random volume: 0.000
Random pitch: 0.000
Fade-in: 0.000
Fade-out: 0.000
Loop
Loop count: 0
Random loop count: 0
Playlist ID:

19.2 Processing and Decoding

19.2.1 Supported formats (containers):

- ogg, Vorbis codec (Chrome, Firefox)
- mp3 (Chrome, Safari)
- mp4, AAC codec (Chrome, Safari)

It is recommended to use ogg as it is an open standard, is widespread in browsers and provides good sound quality. The optimal format in respect to the quality and compatibility is 48kHz/16bit. Single-channel sound (mono) is used to store shot samples while two-channel sound (stereo) is used for music playback.

To support a wider range of platforms, a Python script (scripts/converter.py) for converting the source files into other formats is included into the distribution. Run it with the command:

```
> ./converter.py convert_sounds
```

Converting is performed according to the scheme:

- ogg -> mp4
- mp3 -> ogg
- mp4 -> ogg

Resource converting is performed with quality loss and so the resulting files receive a .lossconv suffix.

CHAPTER 20

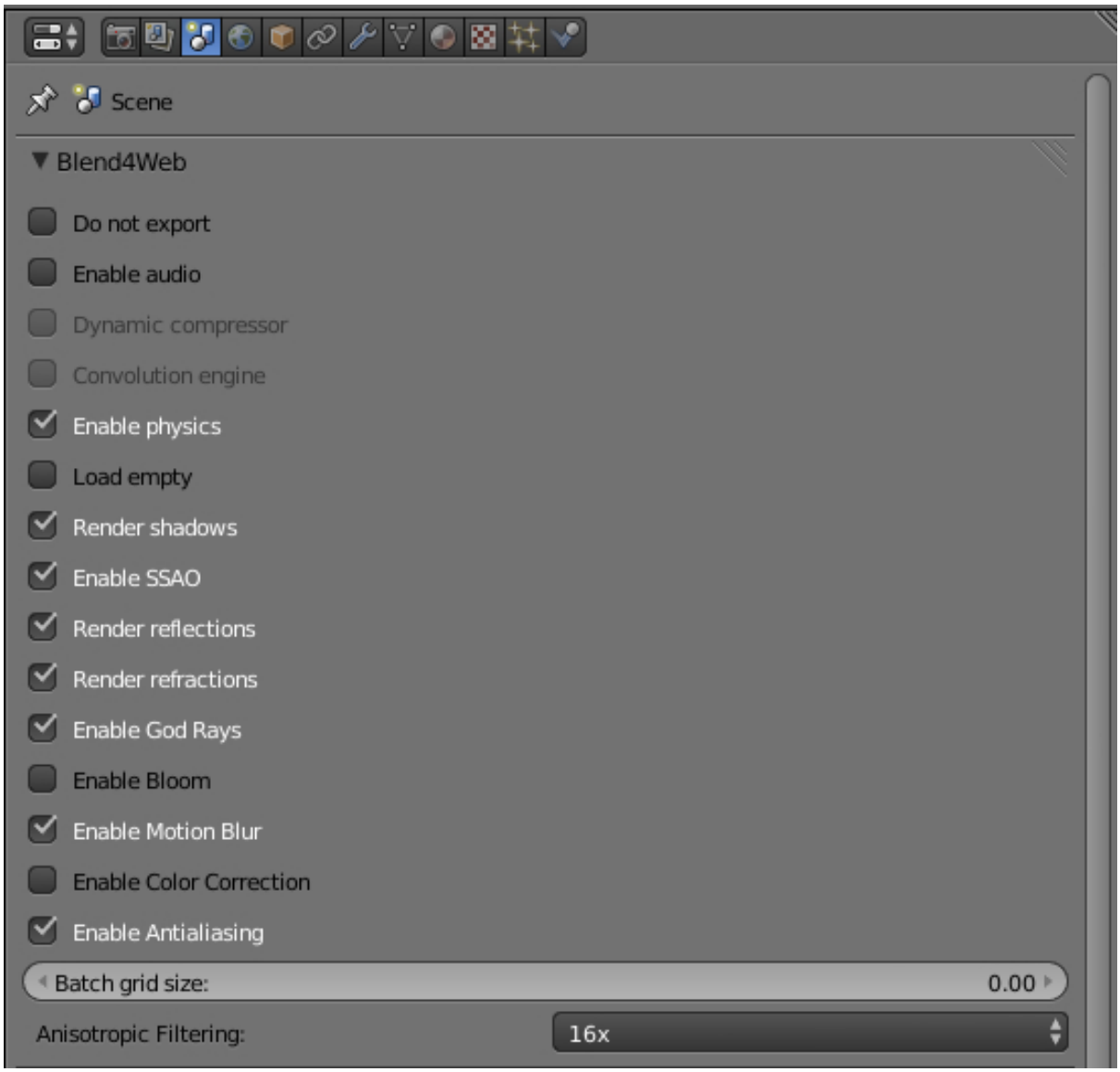
Physics

20.1 Preparing for Use

The physics system is implemented in the uranium.js module and loaded separately from the engine's main code. The uranium.js module itself is a modification of the Bullet physics engine, which is ported to work in browsers. To enable the physics system and to specify the uranium.js module's loading path, applications must use the external API:

```
b4w.config.set("physics_enabled", true);
b4w.config.set("physics_uranium_path", "../../external/deploy/apps/common/uranium.js");
```

To enable physics on the scene use the Enable physics checkbox under the scene tab in Blender.

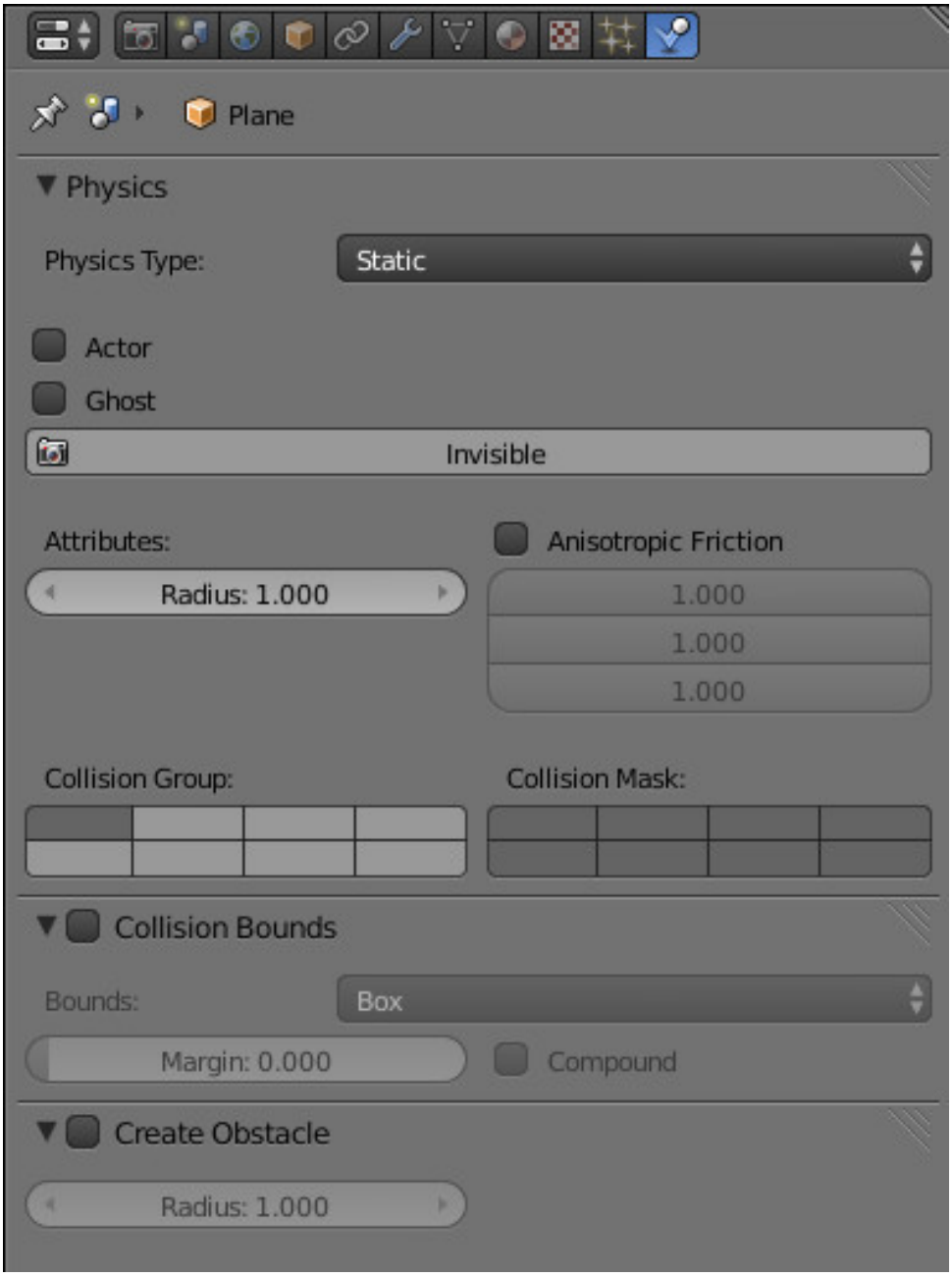


Physical parameters can be set up in the Blender Game mode.

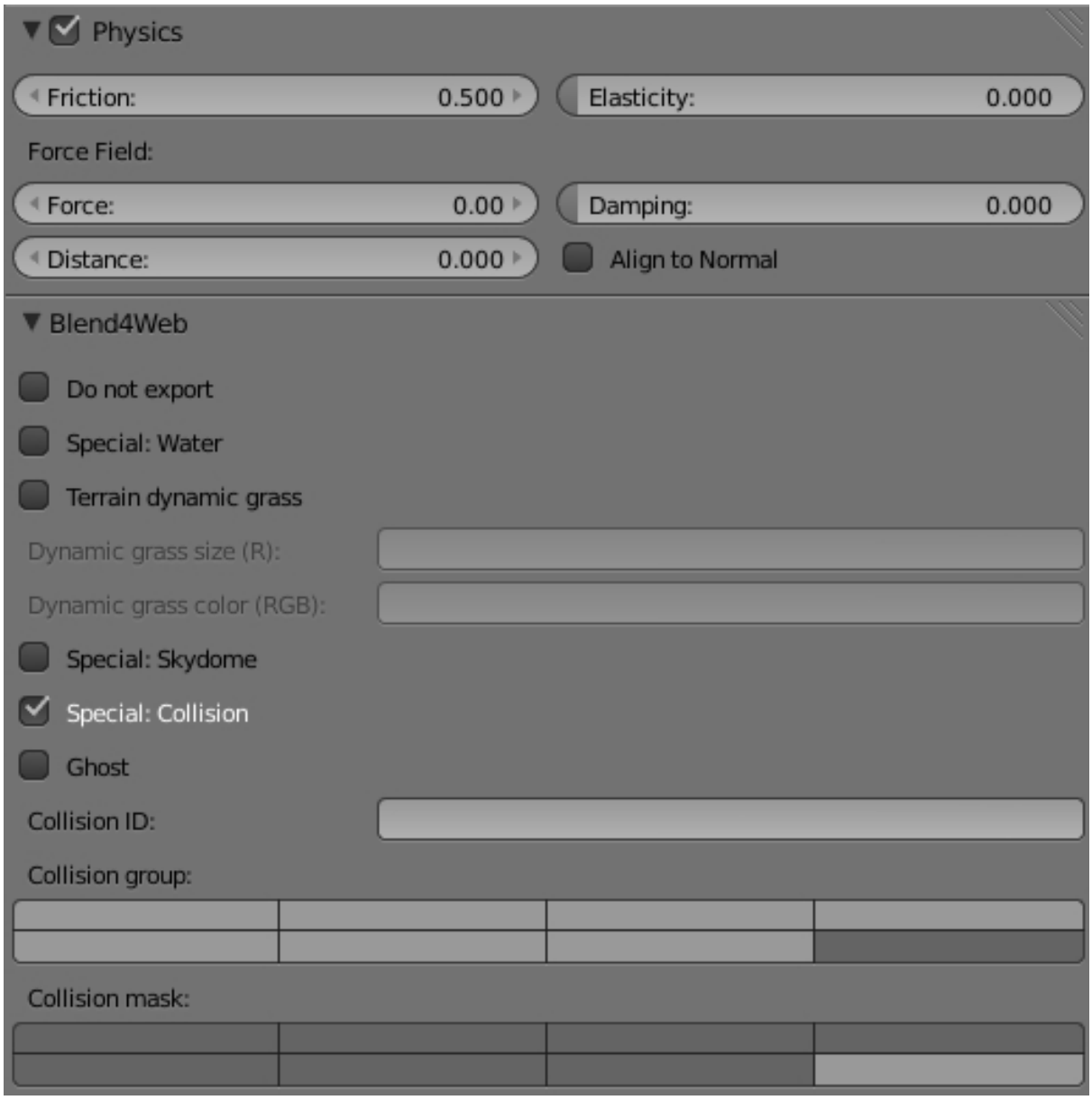
20.2 Static Physics Type

Can be used to limit the movement of other objects, for example to detect collisions with a landscape, walls and so on. In the physics settings of such an object the Static value

(set by default) should be selected for the Physics Type option.

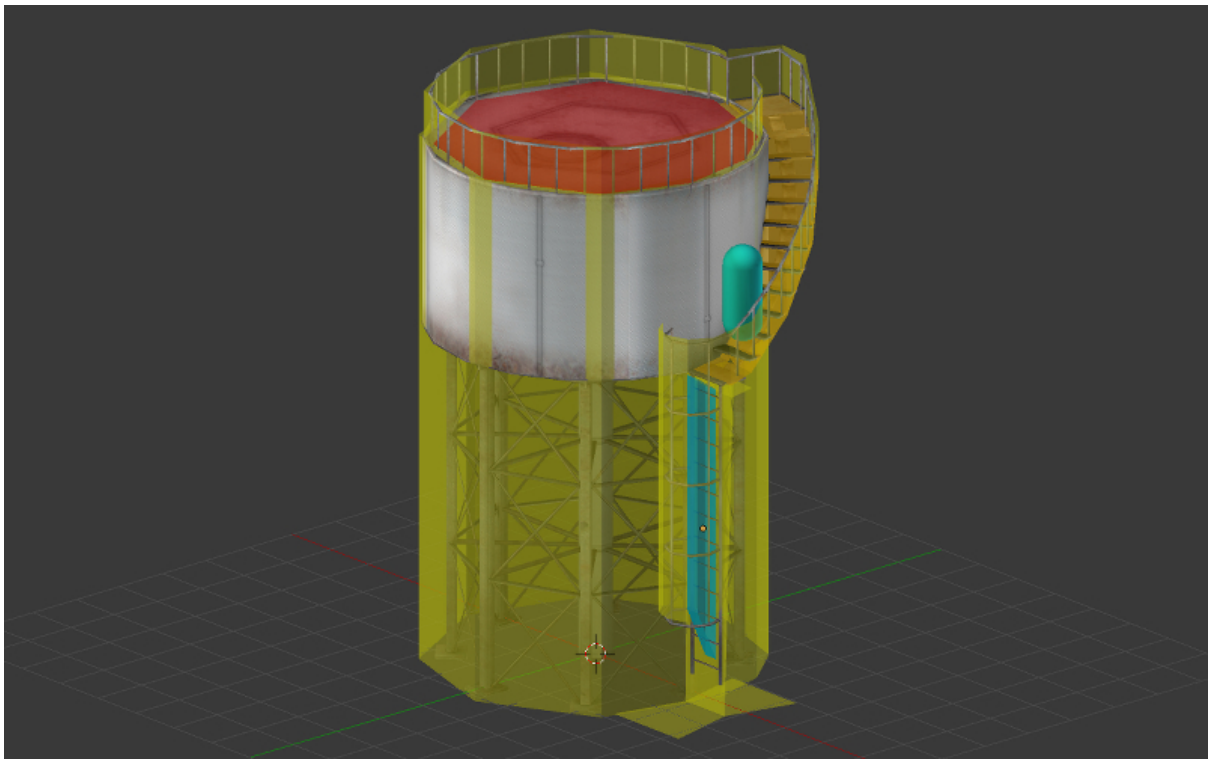


One or multiple physics materials can be assigned to a mesh. Under the Material tab the Special: Collision checkbox must be enabled. The material physics settings are also located on the Physics panel under the Material tab (in the Blender Game mode). The following material physics settings are supported: Friction, Elasticity.



The Collision ID field is intended for detecting collisions with specific materials and can be left empty. An example of Collision ID usage is detecting the landscape surface a character is located on - grass, sand, wooden coating and so on.

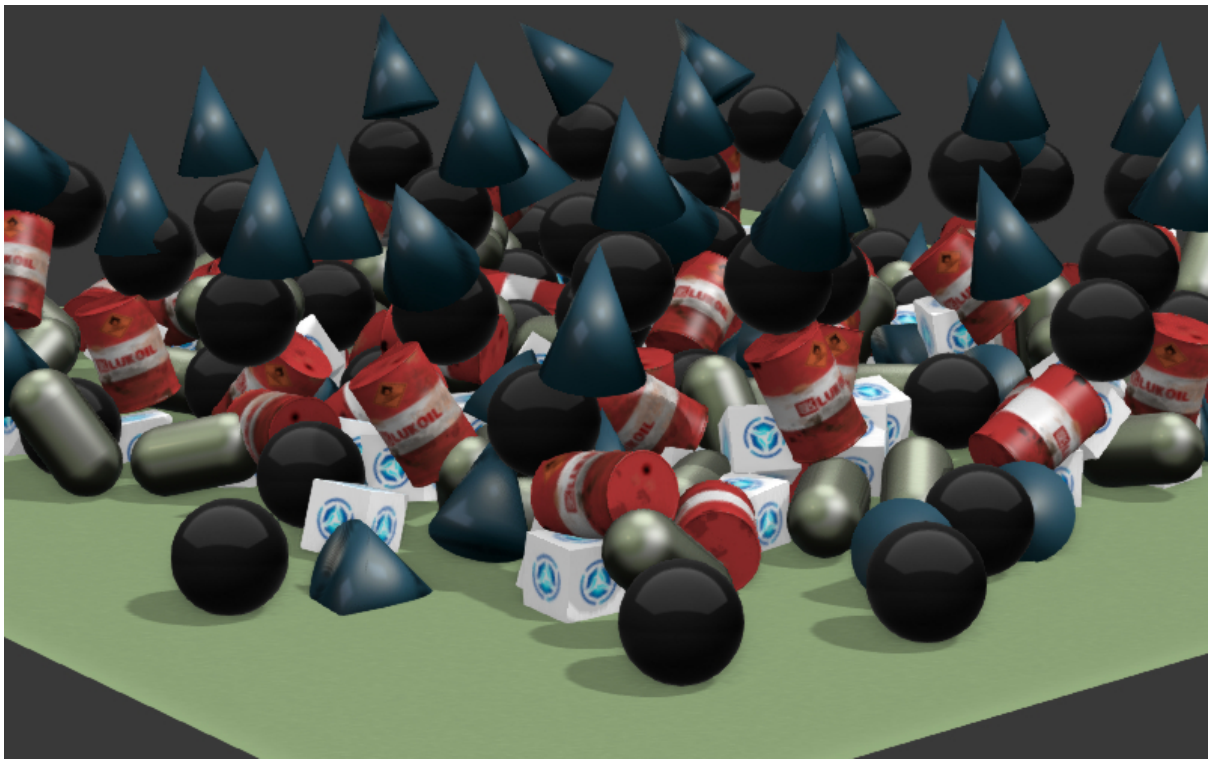
The Ghost option excludes the material from physical interactions but still notifies the application about the contact with it. An example - detecting that the character is located on a vertical ladder.



The Collision group field corresponds to the physics group which the material belongs to. The Collision mask field defines all physics groups with which this material will interact.

20.3 Dynamic Physics Type

Intended for rigid body movement simulation.



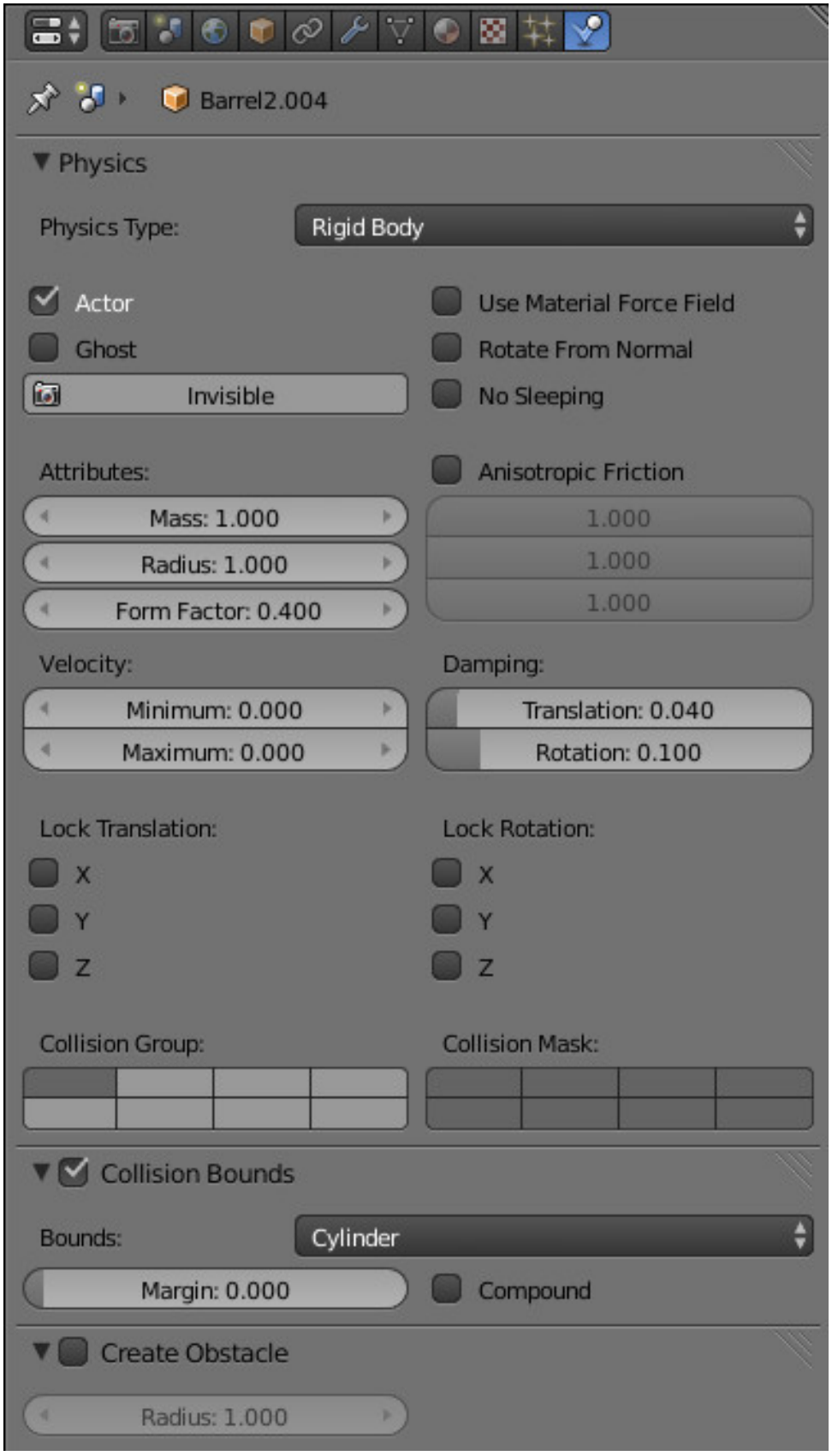
In the physics settings of such an object the Rigid Body (with rotations) or Dynamic (without rotations) values can be selected for the Physics Type option. In the Collision Bounds settings the collider type can be selected - the supported types are: Box, Capsule, Sphere, Cylinder, Cone. Other supported settings are: Mass, Damping - for Translation and Rotation.

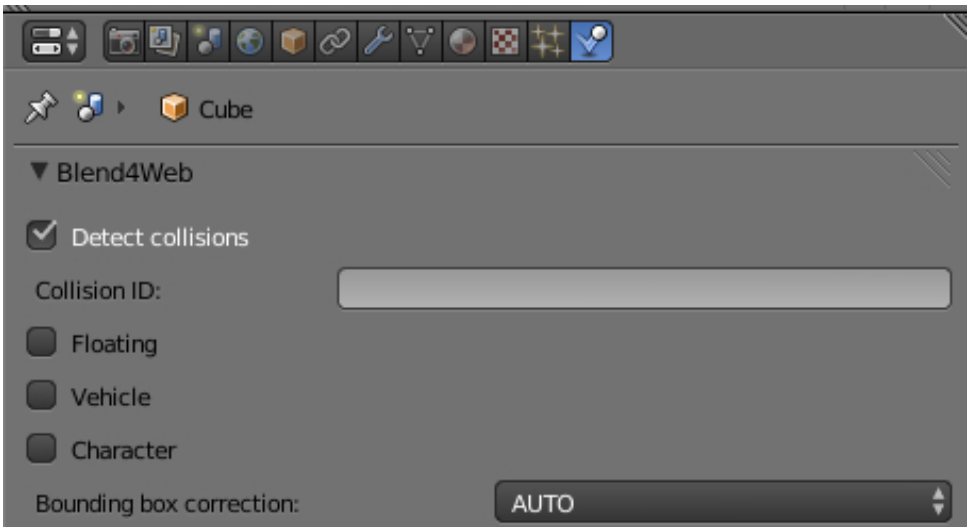
The Collision group field corresponds to the physics group which the object belongs to.

The Collision mask field defines all physics groups with which this object will interact.

The Detect collisions checkbox must be enabled under the object's physics tab. The Collision ID field is intended for detecting collisions with a specific object (for example for detecting proximity of a FPS character to different items) and can be left empty.

Friction and Elasticity are supported for the material of such an object. When multiple materials are used on a single mesh, the physics parameters are taken from the first of


Barrel2.004
Physics
Physics Type:
Rigid Body
Actor
Ghost
Invisible
Use Material Force Field
Rotate From Normal
No Sleeping
Attributes:
Mass: 1.000
Radius: 1.000
Form Factor: 0.400
Anisotropic Friction
1.000
1.000
1.000
Velocity:
Minimum: 0.000
Maximum: 0.000
Damping:
Translation: 0.040
Rotation: 0.100
Lock Translation:
X
Y
Z
Lock Rotation:
X
Y
Z
Collision Group:
Collision Mask:
Collision Bounds
Bounds:
Cylinder
Margin: 0.000
Compound
Create Obstacle
Radius: 1.000

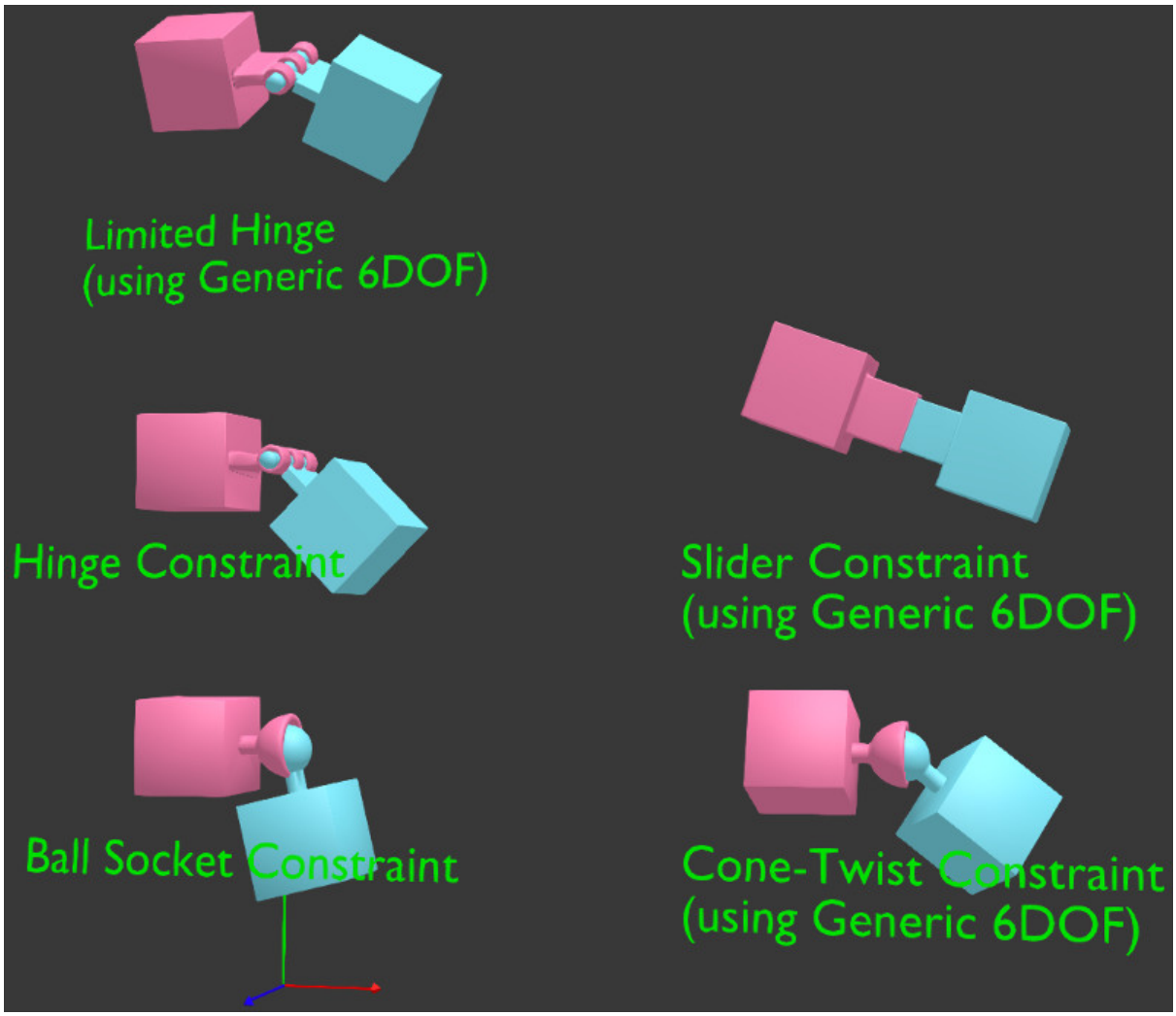


them.

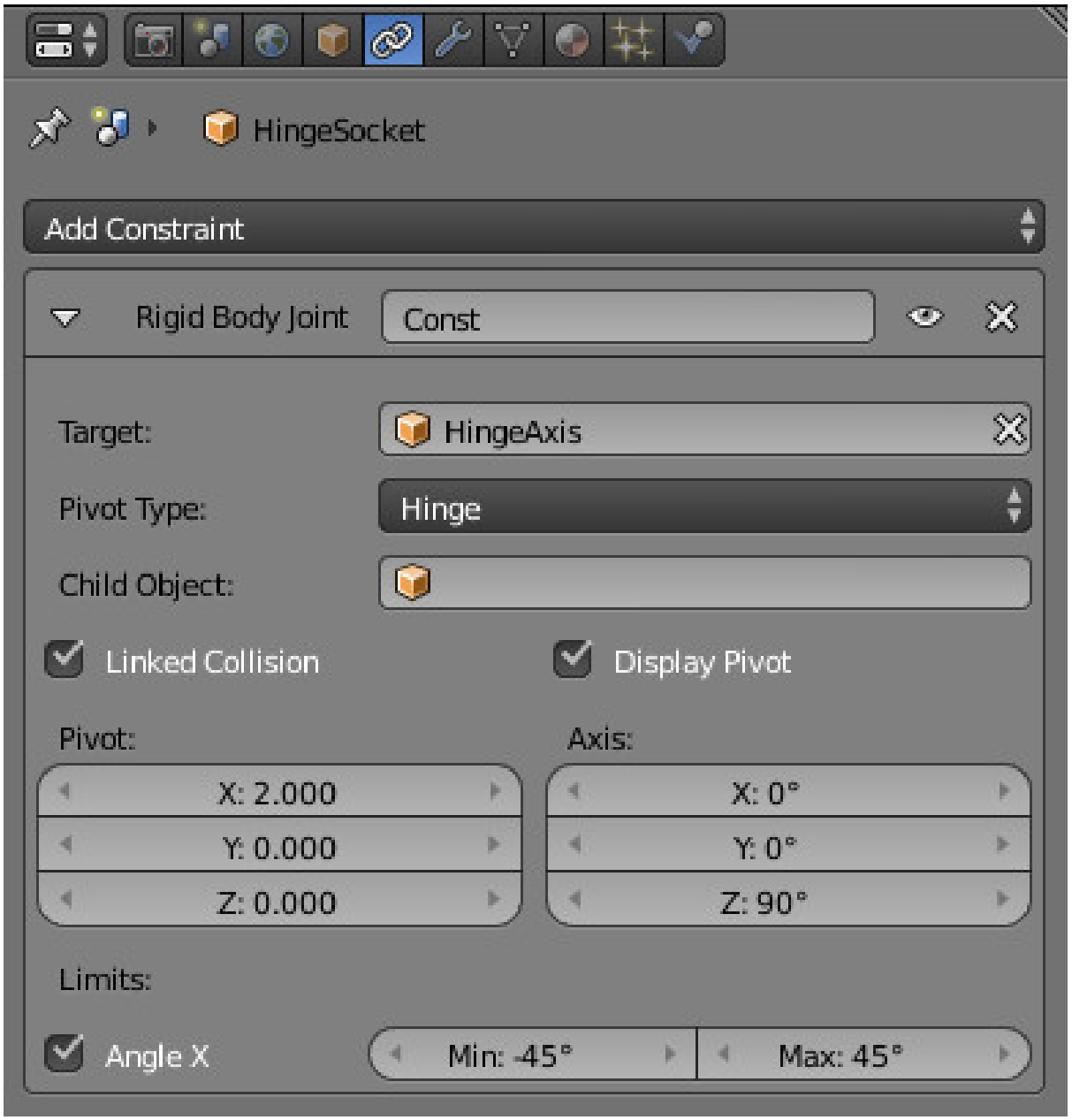
For the camera object the Physics Type = Dynamic parameter must be used, and the Detect collisions checkbox must be enabled.

20.4 Constraints

Physical constraints are used for limiting the objects' degrees of freedom.

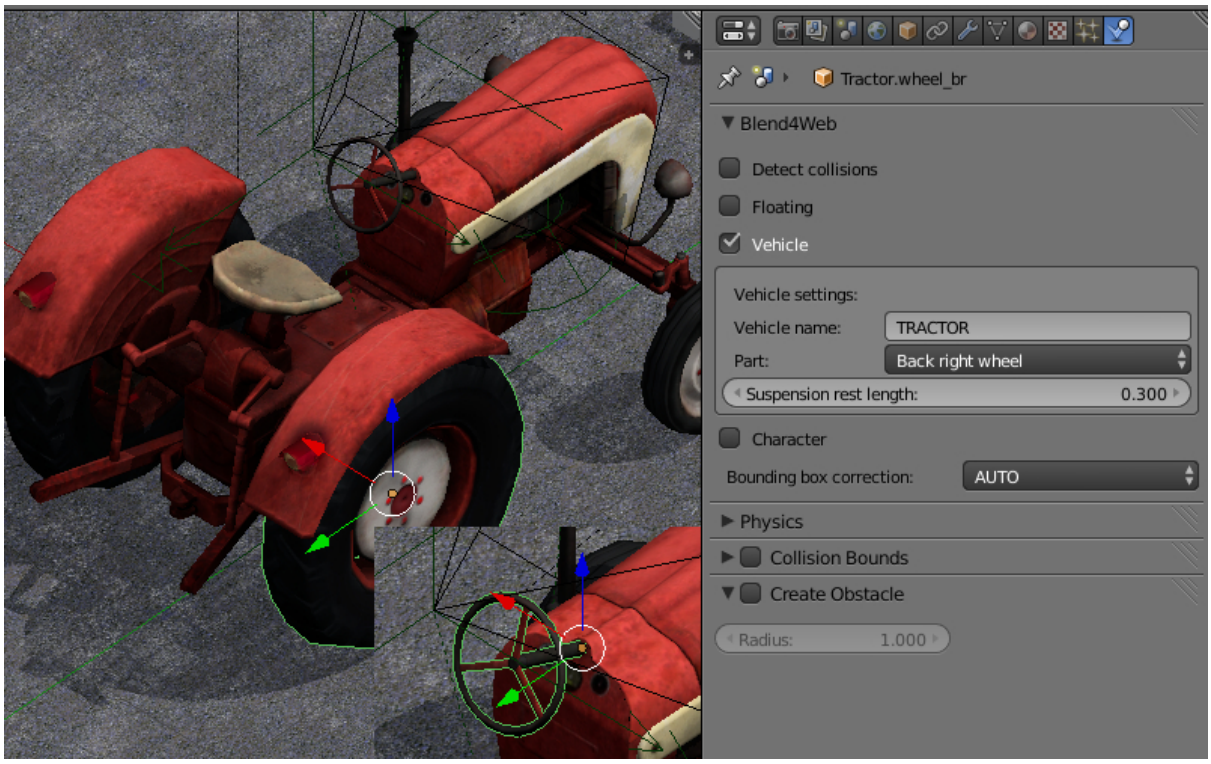


Adding a physical constraint (Rigid Body Joint) to the object can be performed on the Object Constraints panel. The supported types (Pivot Type) are: Ball, Hinge, Cone Twist, Generic 6 DoF. A physical constraint can be added to one of the two interacting objects, while the other object acts as a Target. Both objects can have a static and/or dynamic physics type. In constraints (except Ball) the translation and rotation limits can be set up.



20.5 Wheeled Vehicles

The model of a vehicle must consist of 6 separate objects - a chassis, 4 wheels and a steering wheel. The chassis' mesh center should correspond to the mass center. The centers of the wheels' and the steering wheel's meshes should be located on the rotation axes. The steering wheel should be oriented in the local space of coordinates - X - the rotation axis, Y - to the right and Z - upwards. The object can have any names.



For all 6 objects: select the Vehicle part, specify the same id in the Vehicle name field, select the right object type - Chassis, Steering wheel, Back right wheel and so on. A Suspension rest length setting is also available for the wheels.

It is neccessary to specify a realistic mass for the chassis (because the default value is only 1 kg). To do this go to the physics settings, choose the Rigid Body value for the Physics Type option and specify the required value (for example, 1000 kg) in the Mass field.

20.5.1 Chassis settings parameters

Vehicle Settings > Force max Maximum driving force of the vehicle

Vehicle Settings > Brake max Maximum braking coefficient

Vehicle Settings > Suspension compression Damping coefficient for suspension stretching

Vehicle Settings > Suspension stiffness Suspension stiffness coefficient

Vehicle Settings > Suspension damping Suspension damping coefficient

Vehicle Settings > Wheel friction Friction constant between the wheels and the surface. It should be around 0.8 for realistic vehicles. But it can be increased significantly to achieve a better control (1000 and more).

Vehicle Settings > Roll influence Decreases the wheels' torque decreasing the probability of the vehicle overturning (0 - no torque, 1 - real physics behavior).

Vehicle Settings > Max suspension travel cm Maximum suspension travel in centimeters.

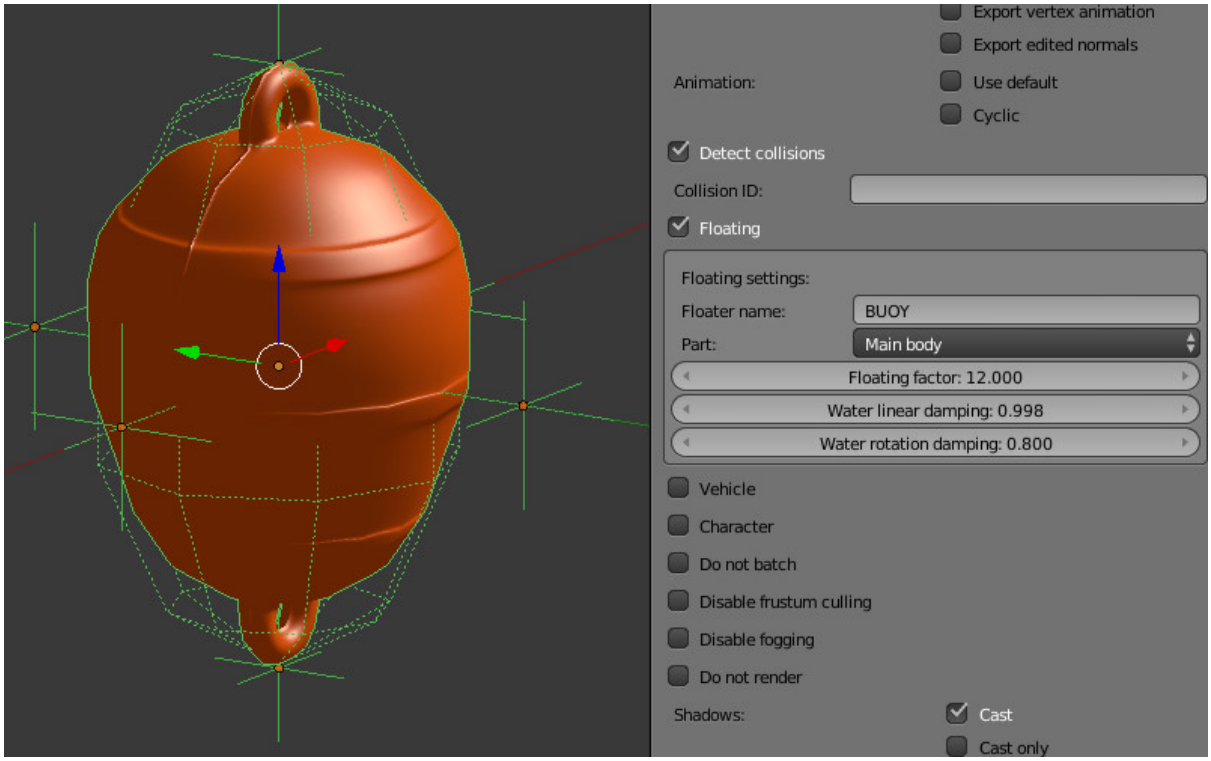
For the Steering wheel it is necessary to specify the maximum steering angle (Steering max) and the ratio between the turn of the steering wheel and the turn of the wheels (Steering ratio). The maximum steering angle value is specified in revolutions. A single revolution equals to 360 degrees. Therefore if Steering max is equal to one and Steering ratio is equal to 10, the maximum turn of the steering wheel will be 360 degrees and the maximum turn of the front wheels will be 36 degrees.

On this stage you can export and load the scene into the engine. We recommend to create a road surface with a physics material. To choose the controlled object press the Q key in the Viewer and select the chassis. Use the W, A, S, D keys as controls.

We can additionally tweak the Damping of Translation and Rotation. This will influence the speed and inertion of the vehicle.

The friction and elasticity of the road surface material do not influence the vehicle's behavior.

20.6 Floating Objects



In order for the object to float on the water surface (an object with the Special water material), it is neccessary to enable the Floating checkbox. There are two types of floating objects: Main body - the floating object itself and Bob - an auxiallary bob-object onto which the buoyancy will be acting. A floating object can have an unlimited number of Bob objects. This can be both meshes or Empty objects.

All objects that are part of the same floating object must have the same name in the Floater name field.

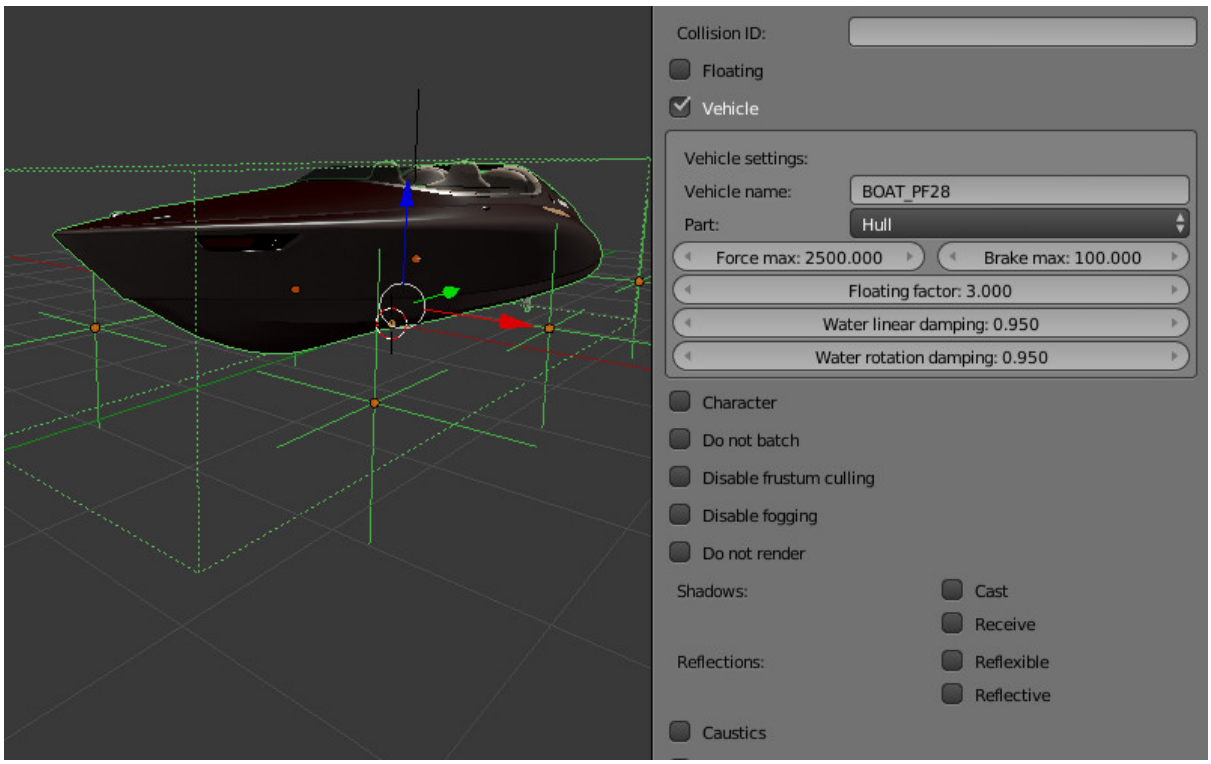
20.6.1 Floating object settings

Floating settings > Floating factor Buoyancy coefficient

Floating settings > Water linear damping Linear velocity damping when the object is on the water surface (or under water). When the object is not in water the physics settings are used.

Floating settings > Water rotation damping Rotation damping when the object is on the water surface (or under water). When the object is not in water the physics settings are used.

20.7 Floating Vehicles aka Watercrafts



Watercrafts use some parameters from the Vehicle settings and all the settings which are similar to Floating settings. It is neccessary to set the Vehicle part type Hull on the main object. Similar to a floating object a watercraft requires auxillary Bob objects.

20.7.1 Watercraft settings

Vehicle Settings > Force max Maximum driving force of the vehicle

Vehicle Settings > Brake max Maximum braking coefficient

Floating settings > Floating factor Buoyancy coefficient

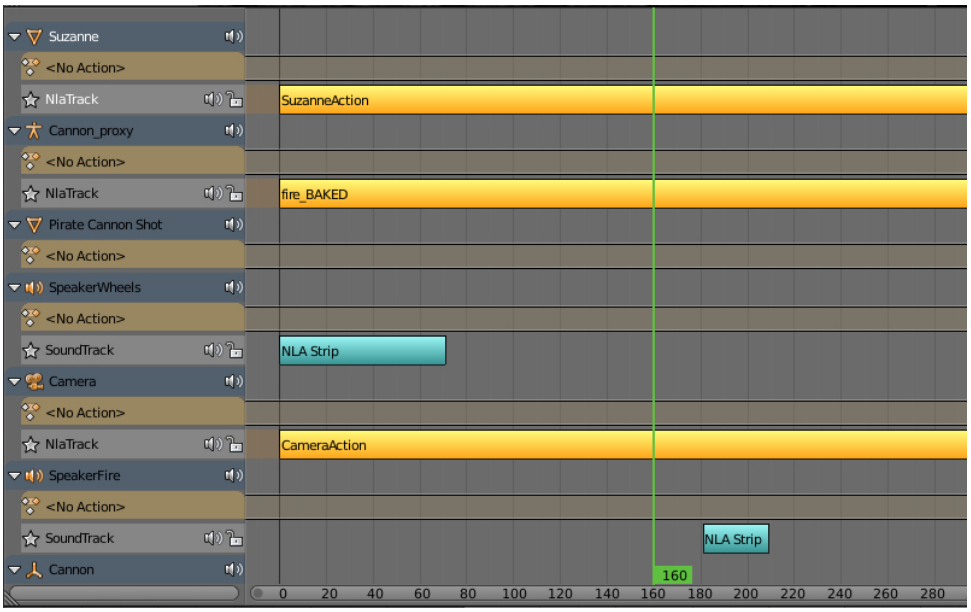
Floating settings > Water linear damping Linear velocity damping when the object is on the water surface (or under water). When the object is not in water the physics settings are used.

Floating settings > Water rotation damping Rotation damping when the object is on the water surface (or under water). When the object is not in water the physics settings are used.

CHAPTER 21

Non-Linear Animation

The Blender's non-linear editor lets us set the scene's behavior in a comfortable way. With its help we can implement a simple scenario which does not require any interference from the user. This way coding is not needed for simple scenes and demo applications.



The engine supports controlling the following entities:

- Any animation the parameters of which can be presented with Actions
- Audio playback
- Particles emission (in the form of a connection with the global timeline)

21.1 Activation

1. Enable the Use NLA checkbox under the Scene tab.
2. In the NLA Editor set up the required behavior for the scene.
3. Choose the animation time interval on the Timeline panel.

21.2 Additional Options

The Blend4Web > Cyclic NLA scene setting activates the cyclic NLA animation mode.

21.3 Limitations

- Vertex animation is not supported.
- Scaling and looping of separate animation strips is not supported.
- A simultanious playback of differnet types of animation for the same object is not supported.

CHAPTER 22

For Application Developers

22.1 A Hello World! App

The simplest Blend4Web app may look like this:

```
<!DOCTYPE html>
<html>
<head>
<script src="b4w.min.js"></script>
<script>
function hello() {
    var m_version = b4w.require("version");
    document.body.innerHTML = "Hello, Blend4Web " + m_version.version() + "!";
}
</script>
</head>

<body onload="hello()"></body>

</html>
```

The engine library is embedded with the <script src="..."> tag. The app waits for the page to load and prints the current version in the browser window.

22.2 Loading a Scene into an App

To load a 3D scene you need:

1. Place a <canvas> element on a page for rendering.
2. Call the m_main.init() function with the created element id to init the WebGL context after the page is loaded.
3. Call the m_data.load() function to load a 3D scene.

```
<!DOCTYPE html>
<html>
<head>
<script src="b4w.min.js"></script>
<script>
function hello() {
    var m_main = b4w.require("main");
    var m_data = b4w.require("data");

    var canvas_elem = document.getElementById("canvas_id");
    m_main.init(canvas_elem);
    m_data.load("some_scene.json");
}
</script>
</head>

<body onload="hello()"><canvas id="canvas_id"></canvas></body>

</html>
```

Note that a real app should include error checking, setting up the engine before initializing and also a basic system for interacting with the user.

22.3 Creating Apps Quickly

Creating an app from scratch can be a tedious task, especially for beginners. To address this there is a special addon for the engine called app.js. The addon can be linked up similar to the main b4w.min.js module and can be accessed via the app namespace:

```
<!DOCTYPE html>
<html>
<head>
<script src="b4w.min.js"></script>
<script src="app.js"></script>
<script>

var m_app = b4w.require("app");
var m_data = b4w.require("data");

m_app.init({
    canvas_container_id: "body_id",
    callback: load_cb
});

function load_cb() {
    m_data.load("some_scene.json");
}

</script>
```

```
</head>

<body id="body_id"></body>

</html>
```

In this case the app module creates a <canvas> element inside the body_id container, inits the engine upon page loading and then calls the load_cb callback.

22.4 Module System

While the engine gives an app programmer an API in the scale of dozens of modules, it occupies a single b4w namespace. To call a module's method import it first with the b4w.require function.

It is possible to register external modules if their names do not collide with already existing modules. A module can be registered with a b4w.register call. Check if a module with some name already exists with a b4w.module_check call.

Example:

```
// check if module exists
if (b4w.module_check("my_module"))
    throw "Failed to register module: my_module";

// register my_module
b4w.register("my_module", function(exports, require) {

    // import module "version"
    var m_version = require("version");

    // export print_build_date() from module "my_module"
    exports.print_build_date = function() {
        // exec function date() from module "version"
        console.log("Engine build date: " + m_version.date());
    }
});

// import module "my_module"
var m_my_module = b4w.require("my_module");

// exec function print_build_date() from module "my_module"
m_my_module.print_build_date();
```

22.5 Moving Objects

Note: Make sure that the object you are trying to move is a dynamic object.

Use the following methods of the transform module to move objects in the engine:

get_translation Get the coordinates of the object's center. The option with a single argument returns a new vector (non-optimized option) while the option with two arguments requires an additional vector to write the result down.

get_rotation Get the object's rotation quaternion. There are two ways of calling this function similar to get_translation.

get_scale Get the object's scale.

set_translation, set_translation_v Move the object's center into the specified location. The first function takes separate coordinates as arguments while the second one takes a three-component vector (Array or Float32Array).

set_rotation, set_rotation_v Set the object's rotation quaternion. The first function takes separate coordinates as arguments while the second one takes a four-component vector (Array or Float32Array).

set_scale Set the object's scale. One corresponds to the object's original state. Values less than one mean scaling down, more than one - scaling up. Note that not all objects can be scaled. Particularly scaling is not allowed for physics objects.

set_rotation_euler, set_rotation_euler_v Set the object's rotation using Euler angles. An intrinsic YZX rotation system is used (that means the angles follow in the YZX order and the origin of coordinates rotates and takes up a new position for every angle).

22.6 Quaternions

Quaternion is a four-component vector used to perform rotating. Quaternions have a number of advantages over other rotation methods such as:

- A quaternion isn't ambiguant and doesn't depend on the rotation order as the Euler angles.
- Quaternion's memory usage is more effective (2-4 times less depending on the matrix used).
- Better computing efficiency than for matrices in case of a series of rotations.
- Numeric stability - compensation for multiplication errors arising from float number inaccuracy.
- Convenient interpolation method.

Quaternions have some drawbacks:

- Rotating a vector with a quaternion is more computationally expensive than rotating with a matrix.
- It is difficult to use quaternions for non-rotation transformations (such as perspective or orthogonal projection).

The engine has a number of functions to make it more convenient to work with quaternions:

quat.multiply Quaternion multiplication. Note that left-multiplying A quaternion by B quaternion A*B is a rotation by A. I.e. the object already has some rotation B which we supplement with a new rotation by A.

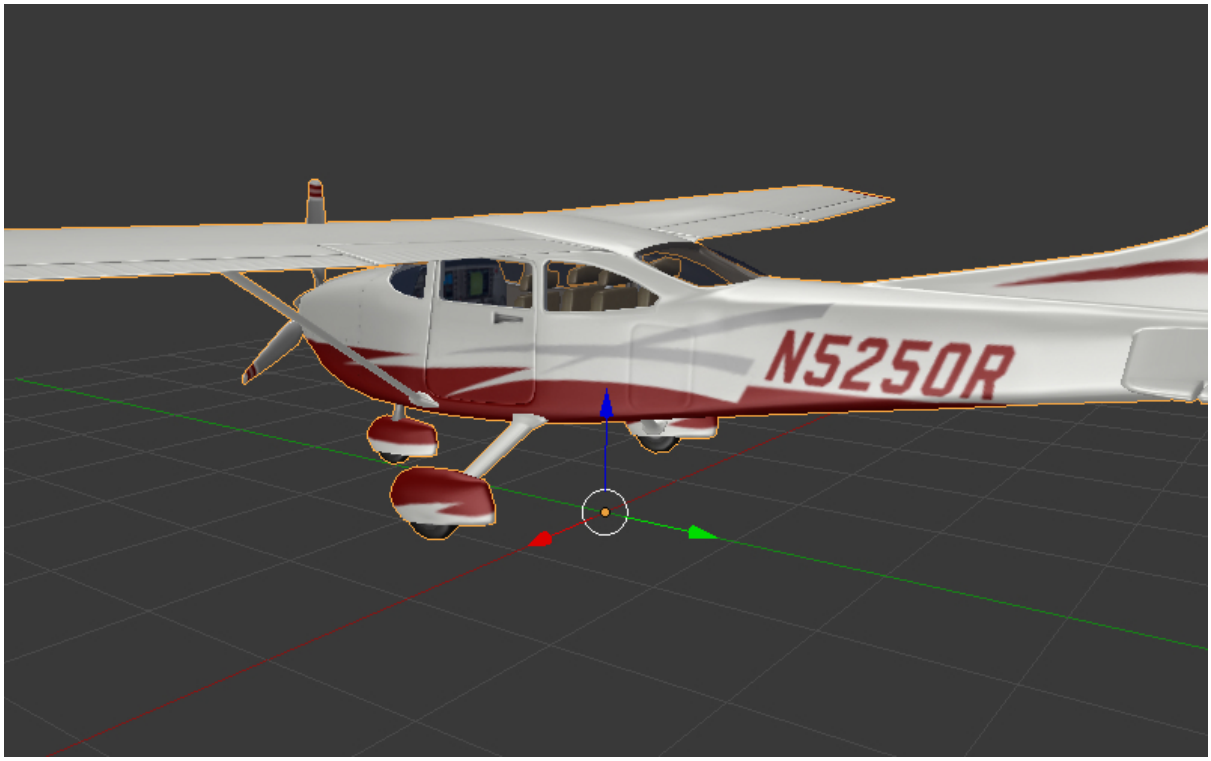
quat.setAxisAngle A quaternion is an alternative presentation of rotation by an arbitrary angle relative to the arbitrary axis (vector). Positive direction of rotation is defined as anticlockwise when viewing from the vector's end. For example the quat.setAxisAngle([1, 0, 0], Math.PI/2, quat) call forms a quaternion which can be used for rotating the object by 90 degrees (anticlockwise if viewing from the X axis' end) relative to the X axis.

quat.slerp Spherical interpolation of quaternions. Used for smoothing the object's rotation and animation.

util.euler_to_quat, util.quat_to_euler. Conversion from Euler angles and back.

22.6.1 Example of working with quaternions

We need to rotate the object by 60 degrees in a horizontal plane to the right. We have a model named "Cessna" in Blender.



Lets save a reference to the object in the aircraft variable:

```
var aircraft = b4w.scenes.get_object_by_name("Cessna");
```

Lets rotate it:

- The orientation of coordinate axes is different in Blender and in the engine. Upon export there will be a transformation [X Y Z] (Blender) -> [X -Z Y] (the engine). Therefore we need to rotate the object relative to the Y axis and not the Z axis.
- A clockwise rotation corresponds to the rotation to the right (i.e. in the negative direction).
- 60 degrees = $\pi/3$ radians.

As a result we get:

```
// compose quaternion
var quat_60_Y_neg = b4w.quat.setAxisAngle([0, 1, 0], -Math.PI/3, b4w.quat.create());

// get old rotation
var quat_old = b4w.transform.get_rotation(aircraft);

// left multiply: quat60_Y_neg * quat_old
var quat_new = b4w.quat.multiply(quat_60_Y_neg, quat_old, b4w.quat.create());

// set new rotation
b4w.transform.set_rotation_v(aircraft, quat_new);
```

The optimized version which does not create new objects:

```
// cache arrays as global vars
var AXIS_Y = new Float32Array([0, 1, 0])
var quat_tmp = new Float32Array(4);
var quat_tmp2 = new Float32Array(4);
...
// rotate
b4w.quat.setAxisAngle(AXIS_Y, -Math.PI/3, quat_tmp);
b4w.transform.get_rotation(aircraft, quat_tmp2);
b4w.quat.multiply(quat_tmp, quat_tmp2, quat_tmp);
b4w.transform.set_rotation_v(aircraft, quat_tmp);
```

22.7 Event-Driven Model

The event-driven model provides a universal interface for describing the 3D scene's change of state. It simpifies the processing of physics events and user actions.

22.7.1 Sensors

The basic unit of the event-driven model is a sensor. A sensor is a programming entity and can only be active (1, one) or inactive (0, zero). Some sensors may carry a pay-

load. For example the ray-tracing sensor (Ray Sensor) provides the relative length of the intersection ray.

Users cannot directly control sensors via the external API. Instead all sensors must be present in one or multiple collections - so called sensor manifolds. A manifold is a logic container associated with a scene object. It generates a response to a defined set of sensor events by executing a callback function. To define the manifold it is required to have the following information (see also the API documenation for decription of the controls.create_sensor_manifold() function):

- An object to carry the manifold (e.g. a thrown object).
- An unique id of the manifold (e.g. "IMPACT").
- A callback execution mode (the options are: CT_CONTINUOUS, CT_LEVEL, CT_SHOT, CT_TRIGGER).
- An array of sensors.
- A logic function to define the combination of the sensor states for which the callback function is executed.
- A callback function.
- An optional parameter to pass into the callback function.

22.7.2 Example

Lets consider the task to insonify the impact of a thrown stone. A distinctive sound should be produced for impacting different media (for example terrain and wall). There are collision meshes with physical materials in the Blender scene, material ids are "TERRAIN" and "WALL". There is also a physical object being thrown in the scene, the object is named "Stone".

Lets define a collision sensor for each medium, by the type of the sound produced.

```
// 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");
```

Add the sensors into an array. Use the OR logic in the logic function. Place the sound processing code in the callback function. Create the sensor manifold with the "IMPACT" id and the CT_SHOT type.

```
// 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);
```

When the "Stone" object collides with any physical material of "TERRAIN" or "WALL", the callback function is executed. Inside this function we get the values of both sensors by their indices in the sensor array (0 - "TERRAIN", 1 - "WALL"). The sensor value = 1 (active) means that the collision happened with the corresponding physical material. As a result the corresponding sound is produced (the code is not shown).

22.8 SDK File Structure

apps_dev source code of the applications (not all applications are available as a free version)

closure-compiler Google Closure compiler, its externs and their generators

csrc source code (in C) of the binary part of the engine exporter and of the other utilities

doc_src source files of the current manual written in reST

external

- blender source files of the Blender scenes (not all scenes are available as a free version)
- blender_scripts exporter and utility scripts for Blender
- deploy
  - api_doc engine's API documentation for developers in HTML format (built automatically based on the engine source code)

apps 3D applications intended for deploying; the directory duplicates apps_dev

assets loading resources: scenes, textures, sounds (not all resources are available as a free version)

assets.json meta data with information about scenes loaded by the Viewer

doc the current user manual in HTML format, built autamatically from doc_src

globals_detect utility code for detecting global variables

tutorials source files for the tutorials

reexporter.py and Makefile Python script and makefile for exporting of all the scenes located in the external/deploy/assets automatically

glsl_utils

compiler compiler for the engine's GLSL shaders

out contains the result of the engine's GLSL shaders compilation

pegjs grammars of the PEG.js parser generator for implementing the GLSL preprocessor, and also the script for generating the parser modules from these grammars

index.html web page with links to 3D applications (not available as a free version)

license files with license texts

Makefile makefile for building the engine, the applications, the documentation and for deploying on a remote server (not available as a free version)

README.rst README file

scripts utility scripts

chrome_debug.sh script which starts Chrome in debugging mode

compile_b4w.sh script for executing Google Closure compiler to minify and obfuscate the engine and applications code

converter.py script which halves the texture dimensions, compresses the textures into the DDS format, converts sound files into mp4 and ogg formats

custom_json_encoder.py fork of the json Python module, sorts the keys in reverse order

gen_glmatrix.sh script for generating the math module based on the source code of glMatrix 2

gpu_shader_analyzer_server.py script starting the local web server which calculates complexity of the shaders

graph.sh SVG generator for the current scene graph, used for debugging rendering

memory.sh script for checking memory (RAM) and video memory (VRAM)

plot.sh debugging information graph builder

remove_alpha_channel.sh script for removing the images alpha channel

report_unused_resources.py script for checking of and reporting about unused resources (images and sounds referenced by the exported files)

screencast.sh script for screen video recording

shaders GLSL shaders of the engine

src main source code of the engine's kernel

addons source code of the addons (not included into the engine's kernel, should be linked separately)

ext source code of the external declarations (forms the engine's API)

third_party code of the third party libraries

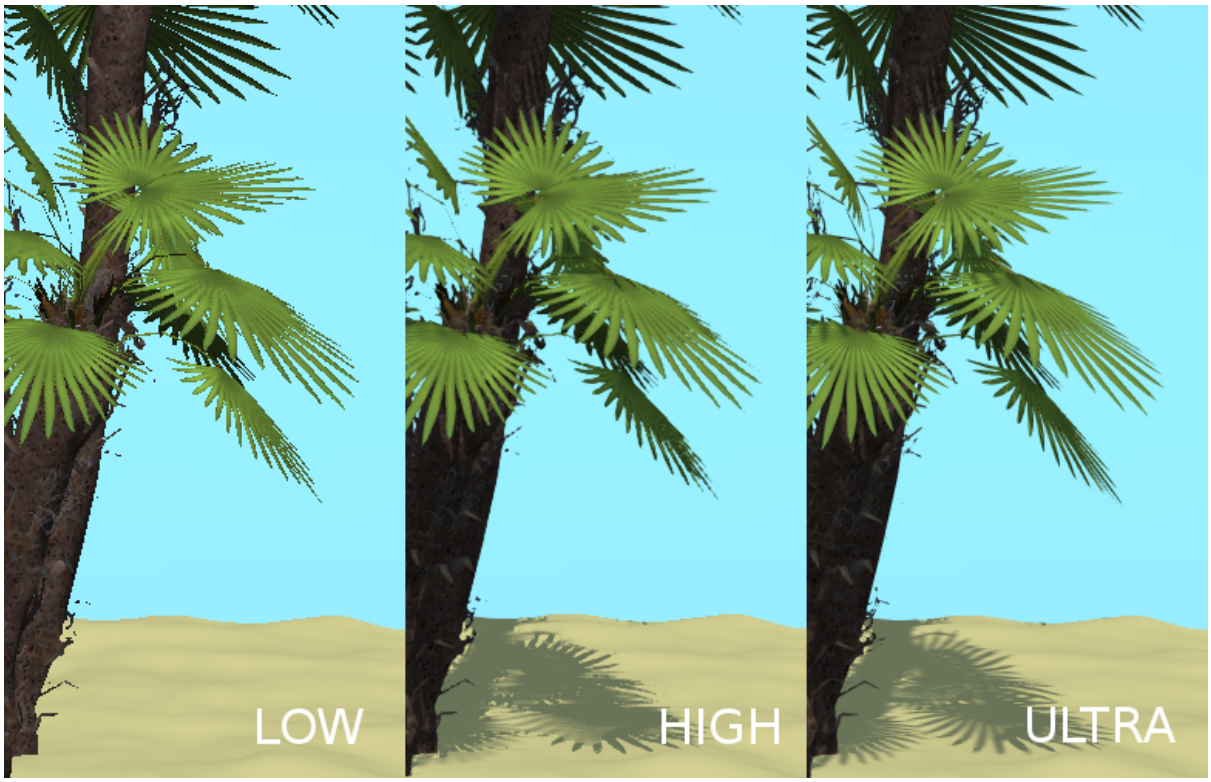
uranium source code and building scripts of the Uranium physics engine (the fork of Bullet)

VERSION contains the current version of the engine

22.9 Quality Profiles

Several quality profiles are implemented in order to support platforms with different functionality.

- low quality (P_LOW) - a range of functions is turned off (such as shadows, dynamic reflection, postprocessing), the size of textures is halved when using a release version, anti-aliasing is disabled
- high quality (P_HIGH) - all features requested by the scene are used, the anti-aliasing method is FXAA
- maximum quality (P_ULTRA) - rendering resolution is doubled, resolution of shadow maps is increased, the anti-aliasing method is SMAA



Switching the quality profiles can be performed in runtime before initialization of the WebGL context. The default profile is 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(...);
```

Application developers can also set the quality parameter upon engine initialization using the app.js addon:

```
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
});
```

CHAPTER 23

For Engine Developers

23.1 Coding Style

This engine uses structural programming. The code is organized in modules. OOP methods are not used, classes are not defined, inheritance is not performed and so on.

The K&R style is used except for the fact that the opening bracket for a compound operator is placed on the same line, for example:

```
function foo_bar() {
    // ...
}

if (a > b) {
    // ...
}
```

4 spaces are used for identation (no tabs allowed).

23.1.1 Examples

The underscore symbol is used in function and variable names:

```
var foo_bar = 123;  // correct
var fooBar = 123;   // wrong
```

All global variables begin with an underscore:

```
var _foo_bar = null;
```

The constants are written in capital letters and never begin with an underscore:

```
var FOO_BAR = 100;
```

To avoid obfuscation the names of external API methods and properties are written as strings:

```
exports["FOO_BAR"] = 123;

exports["foo_bar"] = function() {

}
```

Commenting is in English only. Comment style - JSDoc.

23.2 Building the Addon

Binary Blend4Web addon builds are available for the following platforms: Linux x32/64, OS X x64, Windows x32/64. At the same time users can compile the addon by themselves.

To do this Python 3.x (it's better if it's the same version as in Blender) and a C compiler are required. Under Linux it's enough to install the python3-dev and build-essential packages.

Paths relative to the repository root:

- building script: ./csrc/b4w_bin/build.py
- Blend4Web addon: ./external/blender_scripts/addons/blend4web/

The building process is started in the following way:

```
python3 ./csrc/b4w_bin/build.py
```

As a result of the building you'll get a binary file called:

b4w_bin_[PLATFORM]_[ARCHITECTURE].[STANDARD_EXTENSION],

located in the same directory as the addon. Example: b4w_bin_Linux_64.so. After this the addon is ready to use under this platform.

23.3 Dependencies

In order to effectively develop the engine and the applications a number of third party programs is neccessary (so called dependencies). The majority of these dependencies are part of modern GNU/Linux distributions such as Ubuntu. Under other Unix-like systems (Apple OS X, FreeBSD) installing them from source codes or from other sources is not a problem.

All dependencies are listed in the table below in order of decreasing importance.

Name	Ubuntu 14.04 package	Purpose
Bash	Included by default	interpretator for scripts
Python 3	Included by default	interpretator for scripts
NodeJS	nodejs	compiling shaders
Java	default-jre	compiling and obfuscating the engine modules
LLVM, Clang	llvm, clang	building Uranium
Emscripten	from the source code	building Uranium
ImageMagick, GraphicsMagick	imagemagick, graphicsmagick	converting the resources
NVIDIA Texture Tools	libnvtt-bin	converting the resources
Libav	libav-tools	converting the resources
Gnuplot	gnuplot	debugging
Graphviz	graphviz	debugging
xsel	xsel	debugging
Sphinx	sphinx-doc	building the manual (HTML version)
sphinx-intl	installed with PIP	building the manual (internationalization)
TeX Live	texlive, texlive-latex-extra texlive-lang-cyrillic	building the manual (PDF version)
JSDoc 3	from the source code	building the API documentation

23.4 How to Name Functions and Variables

When creating new functions and variables it is recommended to use the following prefixes and suffixes.

init_ create an abstract object

create_ create a certain object

update_ update the state of an existing object

attach_/detach_ add/remove a temporary object property

append_/remove_ add/remove a temporary property to the already existing properies of the same kind

insert_/pop_ add/remove an array element (accessed by index)

apply_/clear_ operation with flags, binary values or arbitrary parameters

set_/get_ set/get the property/variable value

_tmp global variable - cache in the form of a simple object (array, vector)

_cache global variable - cache in the form of a complex object

23.5 Debugging

Engine debugging is performed with the debug.js module methods.

The structure of the current render graph can be saved in the DOT format using the b4w.debug.scenegraph_to_dot() call, for example, in the browser console. After calling this method save the console's output into the file with the .gv extension. To get the graph in a visual form the graphviz utilities are required. Converting to the SVG format is performed using the command:

```
> dot -Tsvg graph.gv -o graph.svg
```

where graph.gv is the name of the file with the saved graph.

23.6 Shaders

23.6.1 Obfuscator

The shaders used in the engine are converted with an obfuscator. In order to perform the obfuscation it is required to run one of the following commnads in the repository root:

- make compile_shaders - check, obfuscate and export the shaders
- make verify_shaders - only check and obfuscate the shaders

The obfuscator minifies the GLSL code, optimizes it and makes it diffucult to understand it. So far the following procedures are implemented:

- removing excessive space symbols, line brakes and repetitive ";" symbols
- replacing the user ids with shorter single-symboled, two-symboled etc names
- outputting warnings about unused variables and functions (dead code)
- checking the shader syntax
- supporting the import/export mechanism and checking the shaders for the conformance with it

Syntax analysis (parsing) of the shader text is first performed during obfuscation. The corresponding parser is created automatically based on the grammar with the PEG.js generator. Then the shaders are optimized and validated according to the parser data, and after that the shaders are exported in the form of an abstract syntax tree (AST) for direct loading into the engine.

The location of the main files in the repository:

- initial grammar - glsl_utils/pegjs/glsl_parser.pegjs
- parser generation script - glsl_utils/pegjs/gen_nodejs.sh
- parser - glsl_utils/compiler/glsl_parser.js

23.6.2 import/export directives

import/export directives are used to organize, structure and increase the readability of the shader code in the include file. They are specified in the beginning of the file and should look approximately like this:

```
#import u_frame_factor u_quatsb u_quatsa u_transb u_transa a_influence
#import qrot

#export skin
```

The #import directive defines a set of ids which are declared outside the include file but can be accessed from inside it. There is a limitation though: such ids must necessarily be declared somewhere above the place where the include file is linked.

The #export directive defines a set of ids which can be accessed from outside this file. Such ids must necessarily be declared in this file.

Therefore the shader which uses the include file must have all the declarations necessary for import before the place of linking, and can use the exported ids after it.

Ids can be both variable names and function names. If there are no import/export directives it's considered by default that the include file does not use external declarations and does not allow the using of internal ones.

23.6.3 Recommendations and limitations for obfuscation

Because of the following reasons: preprocessing, the need to process multiple shaders and include files and due to the obfuscator's features - its possible to garantee the work of the output code only if a number of rules and limitations are respected relative to the shader source code.

1. In order to describe constants which are defined by the engine at run, it's neccessary to use the #var special directive. For example:

```
#var AU_QUALIFIER uniform
AU_QUALIFIER float a;
```

The syntax here is similar to the #define directive. The point of the #var directive is that the value which it defines allows to parse the initial shader. It's irrelevant what exactly it will be (e.g. 'uniform' or 'attribute' in the above example), because at this level it's unknown anyway. Nevertheless it's better to specify a more or less suitable description and not something arbitrary.

Note: The #var directive is not necessary for constants used not in the shader code but in the preprocessor expressions.

---

2. Using the import/export directives when needed.
3. The built-in functions must not be overloaded - only the user ones.

4. Variables should not be declared with names of the built-in functions, or main (even if it doesn't lead to errors).
5. The #var and #define directives must not be used for replacing single symbols in such operators as: "++", "–", "*=", "/=", "+=", "-=", "==", "<=", ">=", "!=", "&&", "||", "^^".

For example:

```
#var EQUAL =
...
a *EQUAL b;
...
```

6. The usage of the #include directive should not lead to ambiguity during the obfuscation of the include file. This can happen when multiple shaders are included into the same file and the above defined directives (like #var or #define) can have influence on any of them. Also it's better not to use undeclared functions and variables in the include file.
7. Multi-level includes or multiple inclusion of the same include into the same shader is not supported.
8. The shader's malfunction can also be caused by nontrivial using of preprocessing, for example, creating an invalid GLSL code:

```
#if TYPE
void function1() {
#else
void function1(int i) {
#endif
    ...
}
```

23.6.4 WebGL extensions support

The obfuscator's functioning may depend on the WebGL extensions being used, if they somehow influence the shading language. At the moment the following extensions are supported:

- OES_standard_derivatives

23.6.5 Obfuscation errors

In case of an error the obfuscator will output the corresponding message into the console.

Table of possible errors:

Error message	Cause
Error! Ambiguous obfuscation in include file 'FILE_NAME'.	Ambiguous obfuscation in the 'FILE_NAME' include file.
Error! Bad preprocessing collision while obfuscation identifier: 'NAME'.	Error in the FILE_NAME file. Its impossible to obfuscate the NAME variable because of redefining at preprocessing. Redefining the same variable with different qualifiers.
Varying/uniform or varying/attribute qualifiers combination. File: 'FILE_NAME'.	Unacceptable combinations: varying/uniform, varying/attribute.
Error! Extension NAME is unsupported in obfuscator. File: 'FILE_NAME'.	The NAME WebGL extension used in the FILE_NAME file is not supported by the obfuscator.
Error! Include 'FILE_NAME' not found.	The FILE_NAME include file could not be found.
Error! Undeclared TYPE: 'NAME'. File: 'FILE_NAME'.	Error in FILE_NAME file. Undeclared identifier NAME of type TYPE (variable, function, structure etc).
Error! Undeclared TYPE: 'NAME'. Importing data missed. File: 'FILE_NAME'.	Undeclared identifier NAME of type TYPE (variable, function, structure etc). Declaration missing for the identifier required in the FILE_NAME include file according to the #import directive.
Error! Undeclared TYPE: 'NAME'. Possibly exporting needed in include file 'INCLUDE_NAME'. File: 'FILE_NAME'.	Error in FILE_NAME file. Undeclared identifier NAME of type TYPE (variable, function, structure etc). Possibly its export into the INCLUDE_NAME include file should be allowed.
Error! Undeclared TYPE: 'NAME'. Possibly importing needed. File: 'FILE_NAME'.	Undeclared identifier NAME of type TYPE (variable, function, structure etc). Possibly it should be specified as imported in the FILE_NAME include file.
Error! Unused export token 'NAME' in include file 'FILE_NAME'.	Undeclared identifier NAME is allowed for export in the FILE_NAME include file.

Error! Using reserved word in TYPE 'NAME'. File: 'FILE_NAME'.	Error in FILE_NAME file. A reserved id is used for declaring the identifier NAME of type TYPE (variable, function, structure etc).
Error! 'all' extension cannot have BEHAVIOR_TYPE behavior. File: 'FILE_NAME'.	The #extension directive specified for all WebGL extensions in the FILE_NAME file does not support the behavior BEHAVIOR_TYPE.
Syntax Error. ERROR_MESSAGE. File: FILE_NAME, line: LINE_NUMBER, column: COL_NUMBER.	Syntax error in line LINE_NUMBER column COL_NUMBER during parsing the FILE_NAME shader. The initial error description is quoted in the ERROR_MESSAGE. The code listing taken from around the corresponding line is attached to the message (note the peculiarity of pegjs parser which specify the line which is a little bit after the actual error.
Warning! Function 'NAME' is declared in [include]file FILE_NAME, but never used.	An unused function NAME is declared in the FILE_NAME file.
Warning! Include file 'FILE_NAME' not used in any shader, would be omitted!	The FILE_NAME include file is not used in any of the shaders and so it will be excluded from the obfuscated version.
Warning! Unused import token 'NAME' in include file 'FILE_NAME'.	An unused id NAME is imported in the FILE_NAME include file.
Warning! Variable 'NAME' is declared in include file FILE_NAME, but never used.	An unused variable NAME is declared in the FILE_NAME file.

CHAPTER 24

Team Work. Using Git

24.1 Overview

In order to organize team work a git version control system can be used.

The git's main function is to save the history of changes with the possibility to roll back to prevoius versions. Another important function is syncronizing with automatic merging of changes.

Git is a distributed system and every developer or designer has his own local repository (storage). Syncing between the local repositories can be performed via the central ("shared") storage, which can be located on a special machine (server). Access to the server can be organized through SSH protocol.

24.2 Typical Workflow

1. Files can be created, added or deleted during the work process in the local repositories.
2. After a certain logical period of work is finished it is necessary to fix (commit) the changes and/or syncronize with your team mates.
3. Files are prepared for commit i.e. the accounting of changed, new and deleted files and also the resetting of changes.
4. Commit is performed.
5. Local changes are uploaded into the shared storage and become available for the collegues.

There are GUIs for git which provide a user interface for the commands. Nevertheless it's quite convenient to directly enter the git commands.

A limited set of git commands recommended for creating applications and graphical resources is described below.

It's necessary to switch to the repository before executing the commands, e.g.:

```
> cd ~/blend4web
```

24.3 Individual Settings

A new user can set up his name and email using the commands:

```
> git config --global user.name "Ivan Petrov"
> git config --global user.email ipetrov@blend4web.com
```

The set up data will be used in the changelog.

24.4 Checking the Status

It's recommended to check the state of the repository before, in progress and after performing all the operations.

Use this command to check the status:

```
> git status
```

The result of the git status command if all the commits were performed and there are no new files:

```
# On branch master
# Your branch is ahead of 'origin/master' by 2 commits.
#
nothing to commit (working directory clean)
```

Possible result of git status if there are changes. For example the apps_dev/firstperson/firstperson.js and doc_src/git_short_manual.rst files are changed and a new file 123.txt is created:

```
# 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")
```

24.5 Before the Commit

24.5.1 Checking changes (of the text files)

In case of text files it is recommended to view the introduced changes before performing the commit.

Check what was changed in the whole directory:

```
> git diff
```

or in a specific file only:

```
> git diff apps_dev/firstperson/firstperson.js
```

A possible result of the git diff command for a text file:

```
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);
+        }
     }
```

24.5.2 Rolling back files

If the file was changed or deleted but it is necessary to recover it (to the latest commited state) use the command:

```
> git checkout doc_src/git_short_manual.rst
> git checkout 123.txt
```

The introduced changes will be cancelled - this is why this command should be performed with caution.

24.5.3 Unwanted files

If a file is listed in the Untracked files (git status), but version control is not needed for it, it should be deleted or moved beyond the working directory.

24.6 Preparing for Commit

24.6.1 Adding files

If you are happy with the changes, add the needed changed and/or new files for commit.

```
> git add apps_dev/firstperson/firstperson.js
> git add 123.txt
```

Check the status again:

```
> git status
```

Possible result of the git status command after adding some files with the git add command:

```
# 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
#
```

You can see that the apps_dev/firstperson/firstperson.js and 123.txt files were added for commit and the doc_src/git_short_manual.rst file was not added. To make things easier it is recommended to either add such files for commit or cancel their changes with the git checkout command.

24.6.2 Removing files

Some files can be marked as deleted from git after performing the git status command, for example:

```
# 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")
```

In this case if deleting the file should be recorded (i.e. enter the commit), perform the git rm command, for example:

```
> git rm 123.txt
```

If the file was deleted by accident and its necessary to recover it, use the git checkout command.

24.7 Commit

Perform commit with the command:

```
> git commit
```

A text editor window will show up (for example, nano or vim), in which it's nesessary to enter the commit comment in English.

```
  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
```

Save the changes and quit the editor (in nano Ctrl+O, then Ctrl+X; in vim ZZ, or ESC :wq).

After commit it's recommended to recheck the status. Commit is performed correctly if the git status command returns nothing to commit, working directory clean.

24.8 Syncing Between Repositories

24.8.1 From the remote - to the local

After all the commits are performed it's nesessary to load the changes from the remote ("shared") repository to the local one:

```
> git pull
```

Result of the git pull command if there are no changes in the remote repository:

```
Already up-to-date.
```

Result of the git pull command if the remote repository contains changes and syncing was successful:

```
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    external/deploy/assets/location_agriculture/textures/rotonda_02_diff.png
```

If you wish it's possible to look up the changes made by your collegues using the following command:

```
> git diff dbf3877..9f9700c
```

The parameter of this command - in this case dbf3877..9f9700c - shows between which commits exactly the changes were made. This parameter can be conveniently selected in the console in the git pull results and pasted with a mouse click (middle button) where you need.

You can also view the changelog:

```
> git log
```

The git pull command does not always lead to a successful synchronization. The result of git pull when there are conflicts:

```
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 lixer:blend4web
   ff715c2..dbf316a  master     -> origin/master
warning: Cannot merge binary files: external/blender/landscape_objects/Fallen_tree.blend (...)

Auto-merging external/blender/landscape_objects/Fallen_tree.blend
CONFLICT (content): Merge conflict in external/blender/landscape_objects/Fallen_tree.blend
Automatic merge failed; fix conflicts and then commit the result.
```

The steps to be taken at conflicts are described below.

24.8.2 From the local - to the remote

After that the changes should be uploaded from the local repository to the remote ("shared") one to make the changes available for team mates.

```
> git push
```

The result of the git push command if the remote repository already contains all the local changes:

```
Everything up-to-date
```

The result of the git push command if synchronization was successful:

```
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
```

The result of the git push command if synchronization was not successful because the git pull command was not executed first:

```
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.
```

You should execute the git pull command.

The changes uploaded into the central repository can be received by other developers with the git pull command.

24.9 Resolving Conflicts

24.9.1 Overview

Synchronization conflicts occur if both conditions are met

1. the same file was changed both in the local and remote repositories, and
2. automatic merging of the changes didn't occur because the changes are in the same place of the file.

Typical cases:

1. a binary file (texture, blend file) was independently changed by two developers
2. different changes were introduced to the same line of the same text file
3. one developer has changed the file while the other has moved it and so on.

Although synchronization conflicts are normal, if they happen too often it slows down the work. It is recommended to notify your team mates about start of working with the shared binary files, and also to perform synchronization more often. It is nesessary to effectively distribute the work between developers to reduce the number of such shared files. This can be achieved particularly through linking of all the scene's resources from the separate blend files into the master file.

24.9.2 The steps to be taken

It's not recommended to perform any files operations (modifying, deleting) while the repository is in a conflict state.

The first thing to do is to perform the git status command.

```
# 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:      external/blender/landscape_objects/Fallen_tree.blend
#
no changes added to commit (use "git add" and/or "git commit -a")
```

A list of conflicting files can be found in the Unmerged paths section.

The order of the following steps is different for binary and text files.

24.9.3 Binary files

At this stage the conflicting binary files are in the same state as they were in the local repository before the synchronization attempt. The files are fully functional (for example they can be opened by graphics editors).

In case of conflicting binary files it's necessary to sort out (with the team mates or by yourself) which of the files should be left and which should be discarded. Selecting can be performed with the git checkout command.

Select the local version of the file (- -ours). To make sure that it's local you can open it.

```
> git checkout --ours external/blender/landscape_objects/Fallen_tree.blend
```

Select the remote version of the file (- -theirs). To make sure that it's remote you can open it.

```
> git checkout --theirs external/blender/landscape_objects/Fallen_tree.blend
```

Select the local version of the file again (- -ours).

```
> git checkout --ours external/blender/landscape_objects/Fallen_tree.blend
```

Eventually you have to stick to the right version of the file. In case there is a threat of loosing the work you may save the discarded file outside the repository.

24.9.4 Text files

At this stage git introduces both local and remote changes to the conflicting text files, in a special format. Such text files are not workable as a rule.

Example. One developer changed the scene name from "Blue Lizard" to "Green Lizard" in the application file and uploaded the changes into the central repository. Another developer changed "Blue Lizard" to "Red Lizard" in the same line, performed commit and executed the git pull command. As a result this very developer will be responsible for resolving the conflict. The following lines will be present in his version of the application file:

```
<<<<<<< HEAD
            "name": "Red Lizard",
=======
            "name": "Green Lizard",
>>>>>>> 81bf4e2d5610d500ad4d2a2605ee7e61f759f201
```

In case of conflicting text files the following steps can be taken. Files with source code should be edited with or without respect to the changes introduced by both parties. On the other hand, it is easier to reexport the exported scene text files (ending with .json).

24.9.5 Correcting commit

After selecting the required files or editing the changes, add them for commit:

```
> git add external/blender/landscape_objects/Fallen_tree.blend
> git status
```

Possible result of git status command after adding the conflicting files for commit:

```
# 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)
```

Perform commit. It is recommended to leave the default comment:

```
> git commit
> git status

# On branch master
# Your branch is ahead of 'origin/master' by 8 commits.
#
nothing to commit (working directory clean)
```

Conflicts are resolved, the changes from the remote repository are successfully applied in the local repository. Now the changes in the local repository - including the just resolved conflict - can be uploaded to the remote repository with the git push command.

24.10 Tags

Tags are intended for pointing at a certain commit, for example, to specify a stable product version.

View the list of tags:

```
> git tag
```

Create a tag for the release from June 3, 2013, pointing to the commit with a stable product version:

```
> git tag R130603 67bb597f7ed1643ed0220d57e894f28662e614e5
```

Check the commit tag information:

```
> git show --shortstat R130603
```

Roll back to the tag...

```
> git checkout R130603
```

...and return:

> git checkout master

Synchronize the tags with the remote repository:

> git push --tags

Delete the tag (if created by mistake):

> git tag -d R130603

24.11 Other Useful Commands

Check the log for January, 2012, show file names without merging commits:

> git log --after={2012-01-01} --before={2012-01-31} --name-only --no-merges

CHAPTER 25

Problems and Solutions

25.1 Problems when Running the Renderer

1. The "Browser could not initialize WebGL" message is shown.

Follow the instructions listed in the WebGL Failed to Init section.

2. The user interface and background are shown but the default scene is not rendered (a cube with the logo). At the same time the http://get.webgl.org/ site and other WebGL applications are working correctly.

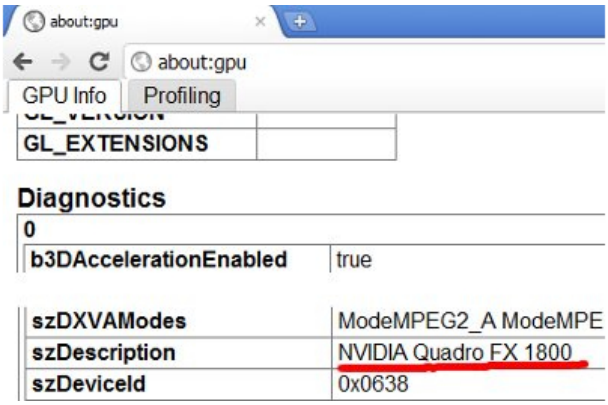
Possible causes:

- The browser is not set up for loading local resources. See the Setting up the Browser for Loading Local Resources section.
- The renderer tries to load resource files which were moved or deleted.
- You are using a combination of browser/drivers/operating system which does not fully support WebGL (examples are: Internet Explorer 11 / MS Windows, open source drivers / Linux).

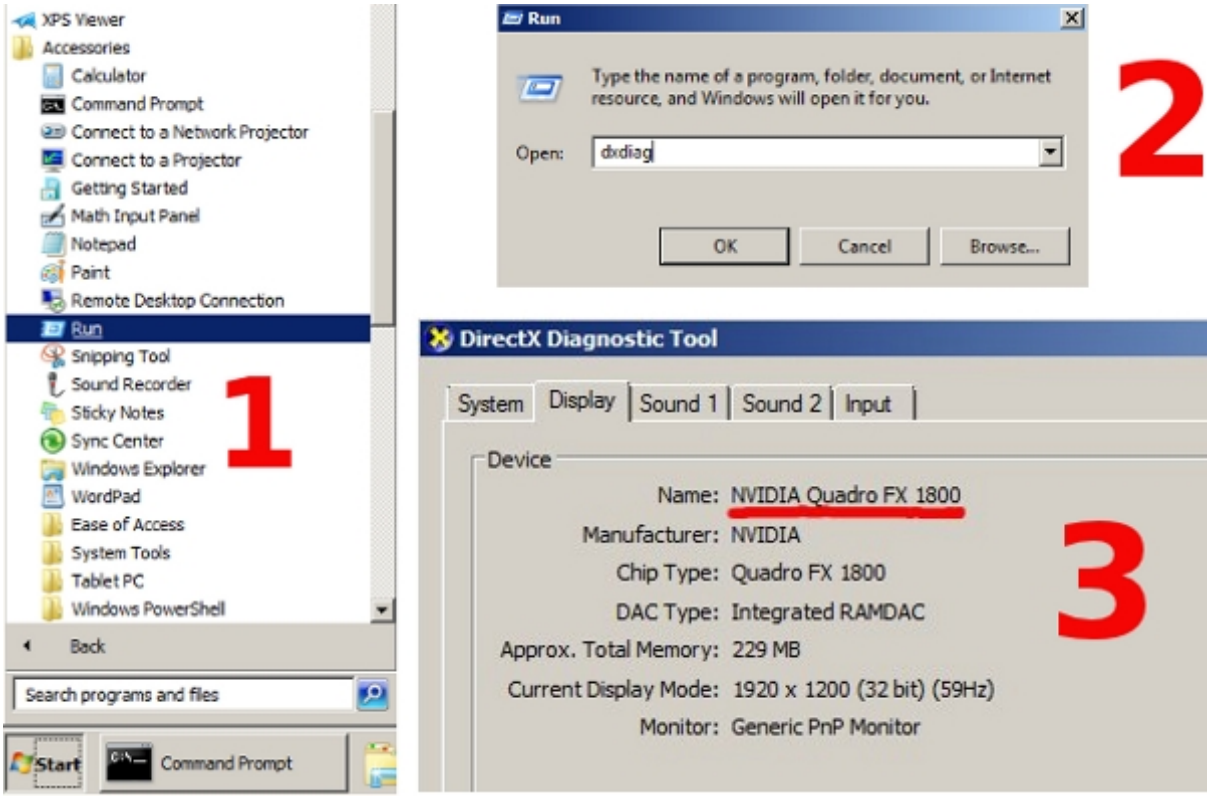
25.2 WebGL Failed to Init

The http://get.webgl.org/ page tells about problems when viewing it in recent Chrome or Firefox. What can I do?

1. Install the latest updates for your system (for MS Windows see the guide). In case of MS Windows install the latest DirectX runtime. Reboot.
2. It is recommended to timely update video card drivers. To detect your video card and its vendor please type about:gpu to the address bar of Chrome browser...



or run the DirectX Diagnostic Tool called dxdiag (MS Windows only).

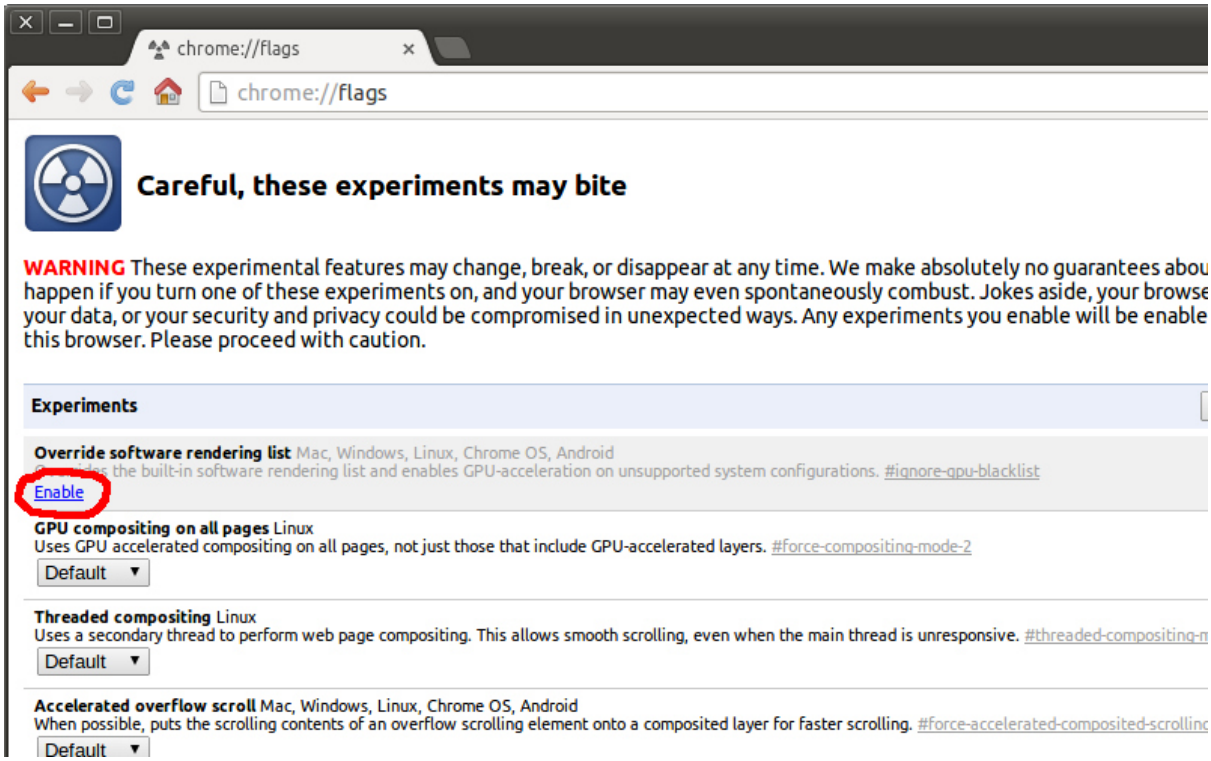


Download the drivers from the corresponding support center (for example Intel, Nvidia, AMD/ATI). Reboot the system after the drivers are installed.

3. If the measures described above did not help to initialize rendering (or there is no possibility to update the system) try to change the browser settings.

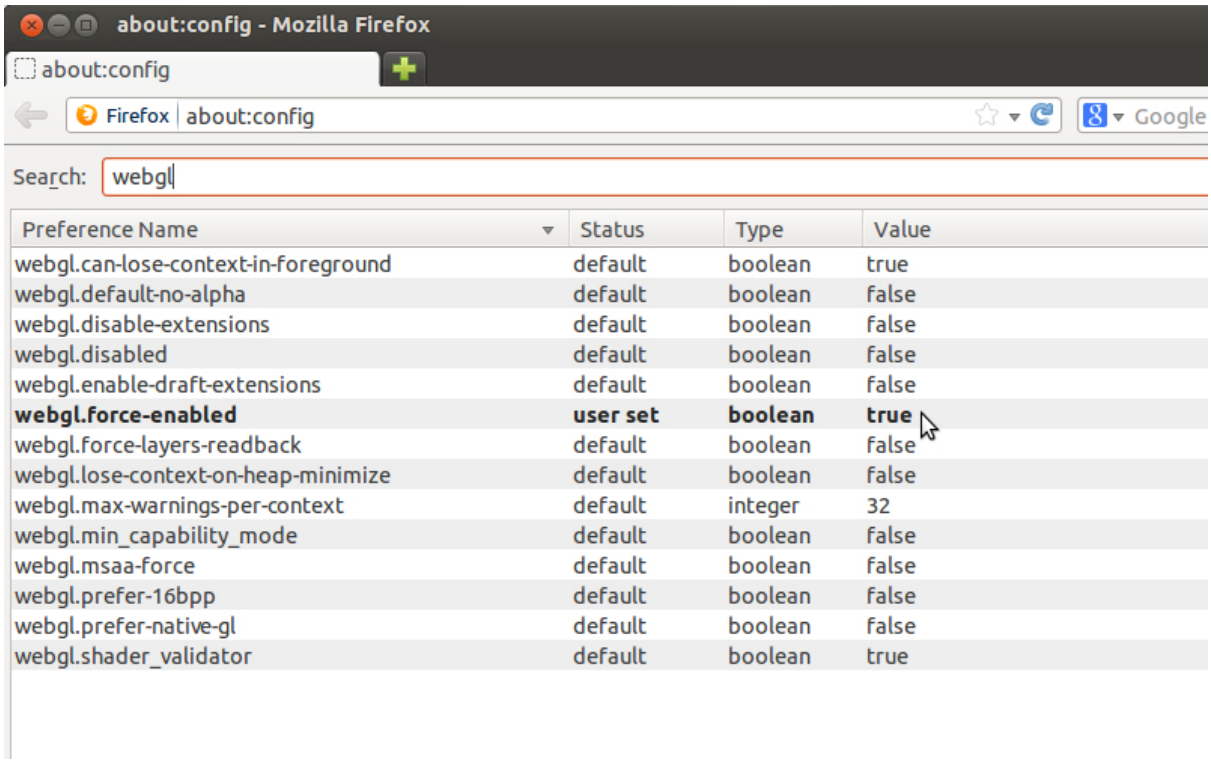
For Chrome:

Enter about:flags (or chrome://flags/) into the browser's address bar, click Enable under the Override software rendering list option and restart the browser.



For Firefox:

Enter about:config into the browser's address bar, search for the webgl.force-enabled parameter and double-click on it to switch from false to true.



Note: For Linux users - due to incomplete OpenGL implementation in open source drivers at the moment it is recommended to use current versions of proprietary drivers for Nvidia and AMD video cards.

25.3 Using a Local Web Server

An easy way for viewing local resources in browsers is running the web server from the standard Python library.

Under MS Windows:

1. Download and install the latest Python distribution from the official site. The current version is 3.4 and by default it will be installed to the Python34 directory on the C drive.
2. Run Command Prompt.
3. Enter the following commands:

   > c:
   > /Python34/python -m http.server

4. Open the http://localhost:8000 page and select the required file for displaying.

Under Linux:

> python -m SimpleHTTPServer

or:

> python3 -m http.server

A port number can be specified by the additional parameter:

> python -m SimpleHTTPServer 8080

- genindex
- search

Symbols

A

B

C

D

E

F

G

H

I

L

M