Blend4Web. User Manual

Release 15.06

Triumph LLC

June 30, 2015

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

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

1.5.2 Experimental Support

- Internet Explorer 11

1.5.3 Mobile Platforms

- Android (on 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
- Device Orientation

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

Features

2.1 General

- effective rendering of 3D scenes of any complexity and size
- data format designed to fit WebGL specifics and optimized for loading
- convenient environment for authoring 3D content, provided by the Blender add-on

2.2 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)
- video textures - video can be loaded and played back, being layered as a texture
- canvas textures - 2D Canvas API can be used to draw on textures

2.3 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

2.4 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

2.5 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

2.6 Particle System

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

2.7 External Scenes Rendering

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

2.8 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
- object outlining

2.9 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)

2.10 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

2.11 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

2.12 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

2.13 Event-Driven Model

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

2.14 Visual Programming

- the NLA Script tool allows to create interactive apps by constructing the logic chains from basic blocks

2.15 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
- module structure of source code
- powerful shader preprocessor with modules and functional blocks (nodes) support
- convenient system to deploy new 3D applications quickly
- support options for a broad range of equipment
- user manual and API documentation
- user interaction - camera, character, actions control

CHAPTER 3

Quick Install

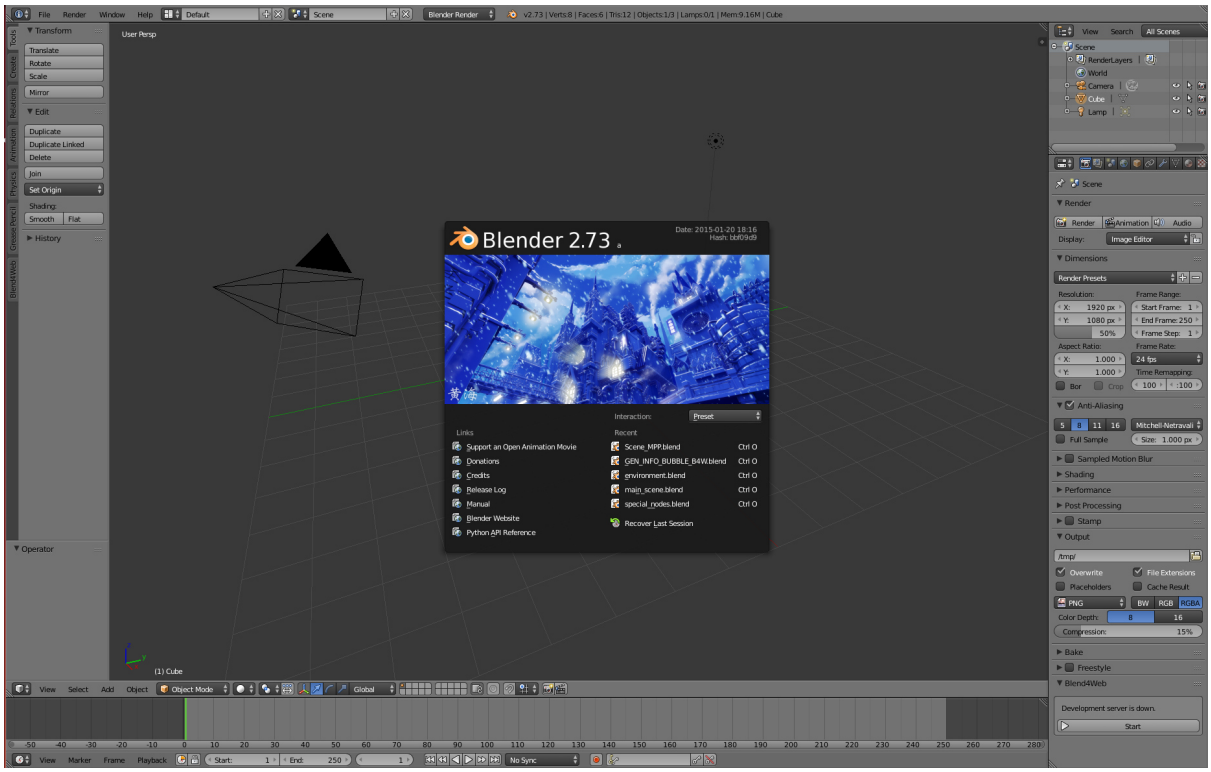
Quick install of the Blend4Web addon suits Blender artists who have no need in full-scale 3D applications development. In this case the main benefit is the opportunity to export a scene into a single HTML file for viewing in WebGL-capable browsers.

For more serious tasks an SDK setup is required.

3.1 Installing Blender

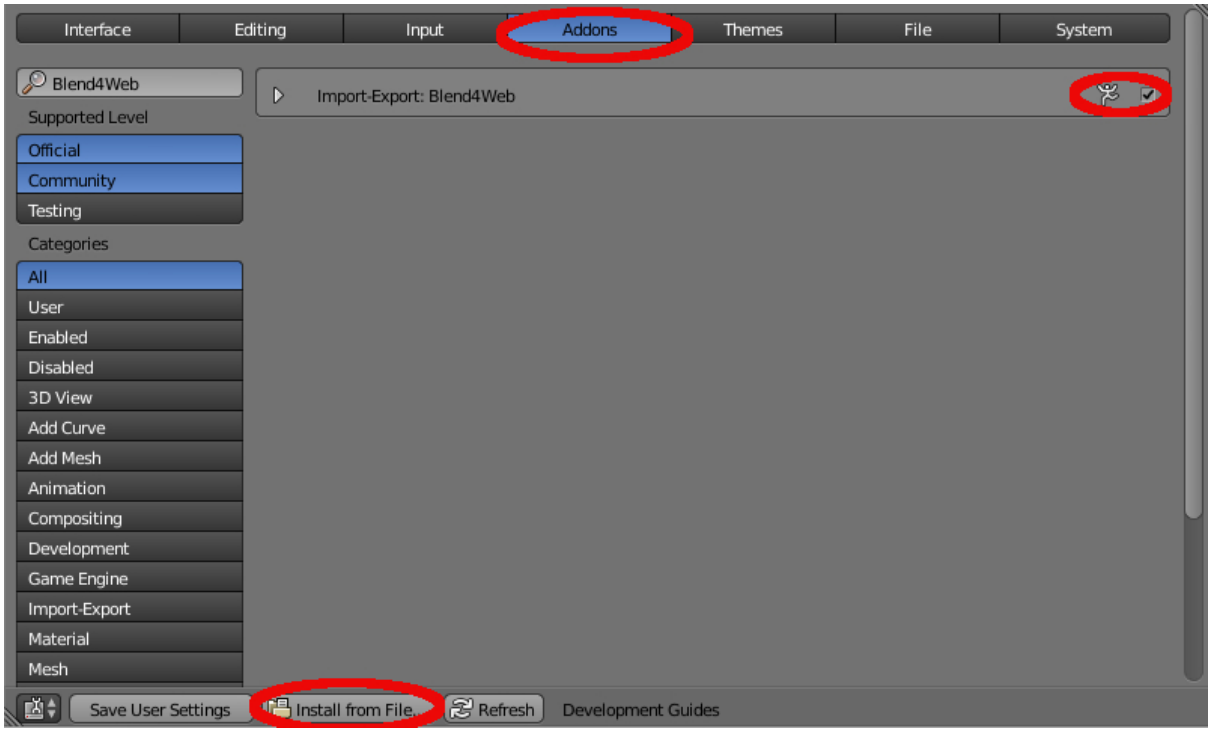
Authoring 3D scenes is carried out directly in Blender which is open source software and is distributed free of charge.

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



3.2 Installing the Addon

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:

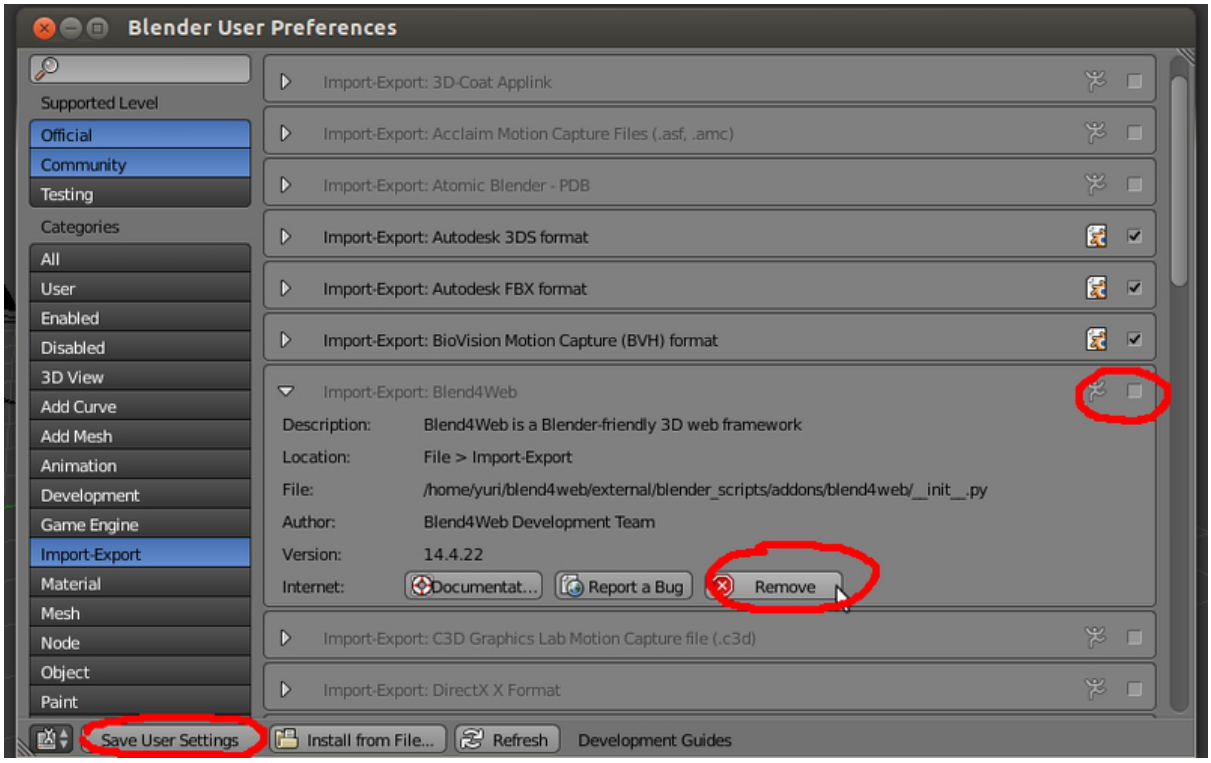
WebGL Browsers Support

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

Installing the SDK

Setting up the development environment suits 3D application developers. To familiarize yourself with the Blend4Web addon quick install can be a better option.

Before installation, please download and install the compatible Blender version, according to this table.

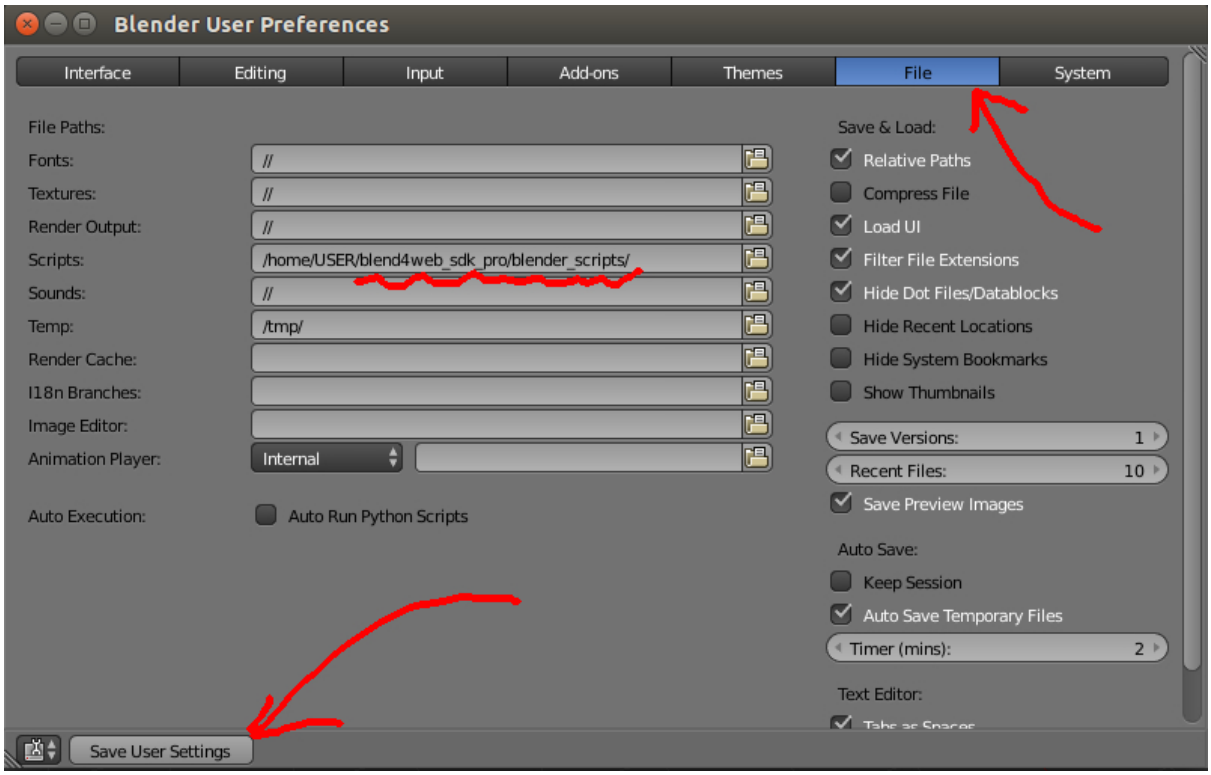
4.1 Unpacking the Archive

Stable versions of the distribution are available as an archive (blend4web_sdk_free_YY_MM.zip – free SDK, blend4web_sdk_pro_YY_MM.zip – commercial SDK). Simply unpack this archive somewhere.

4.2 Setting up the Add-on

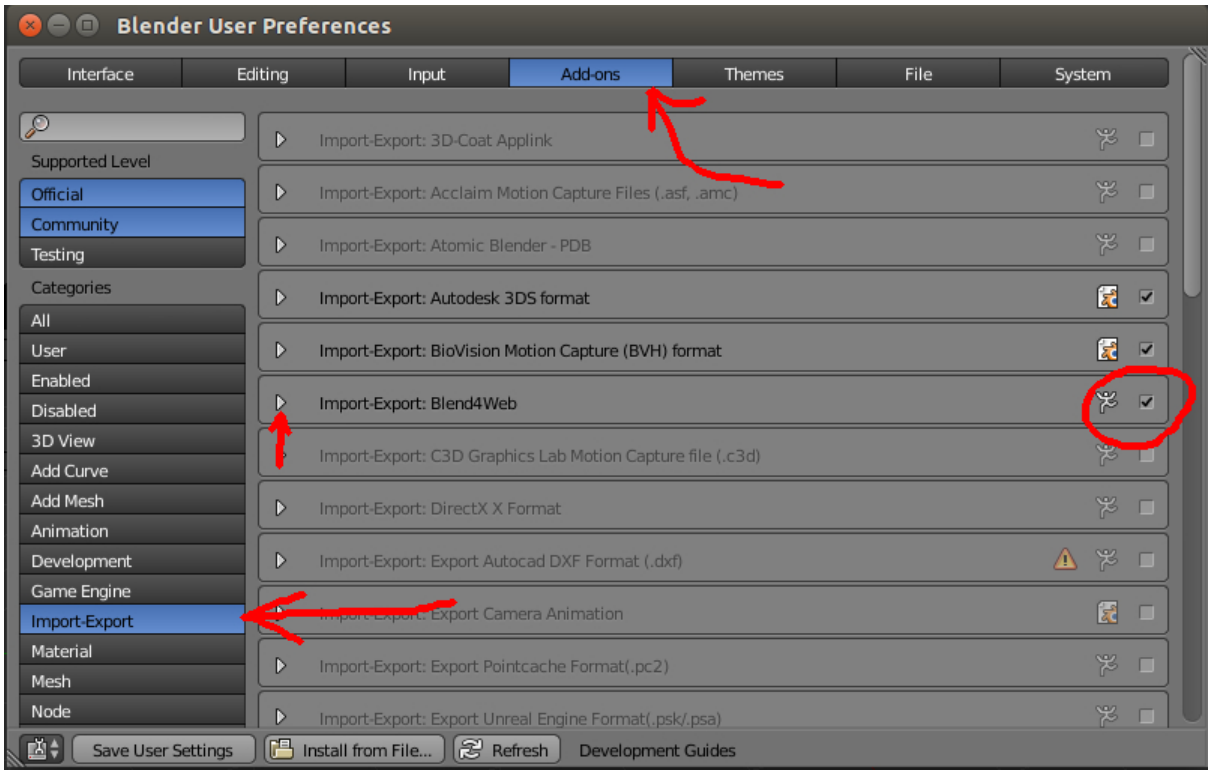
Note: We strongly recommend to remove the addon first if it was originally installed using quick install.

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.

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.



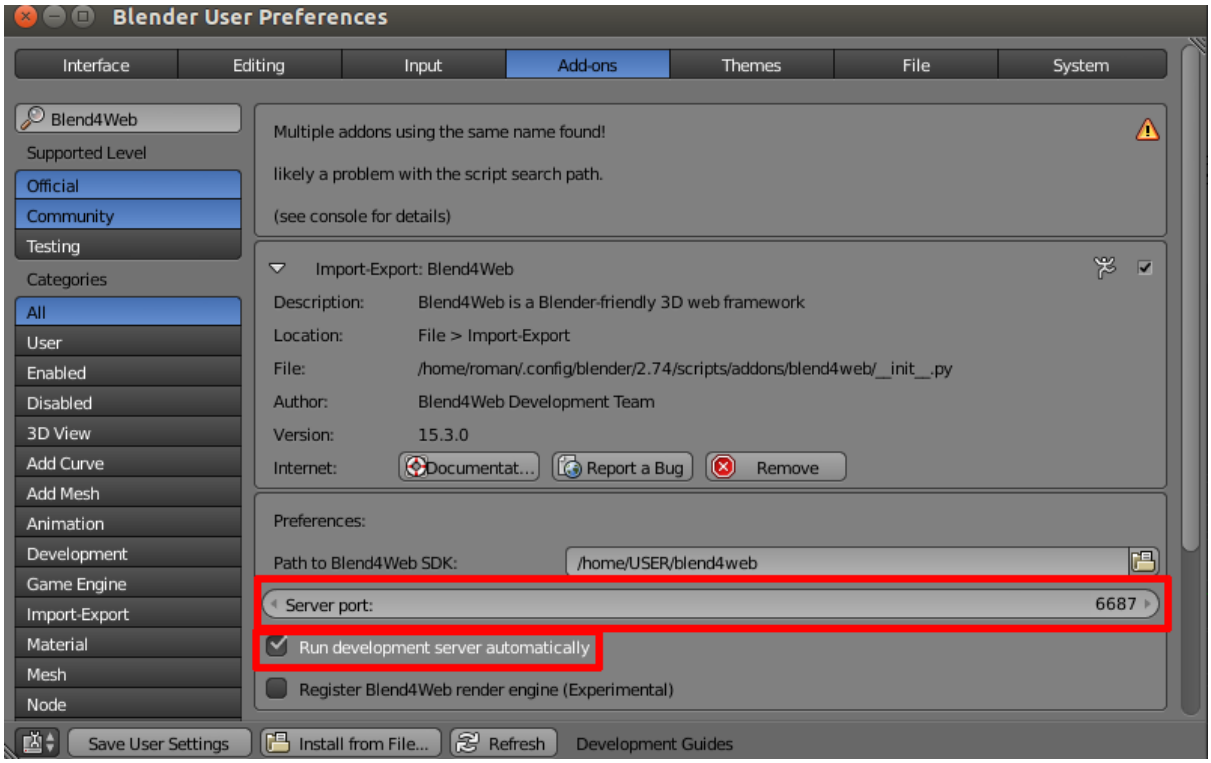
Click Save User Settings. Restarting Blender isn't required.

To check:

In the File > Export menu, the Blend4Web (.json) and Blend4Web (.html) options should appear.

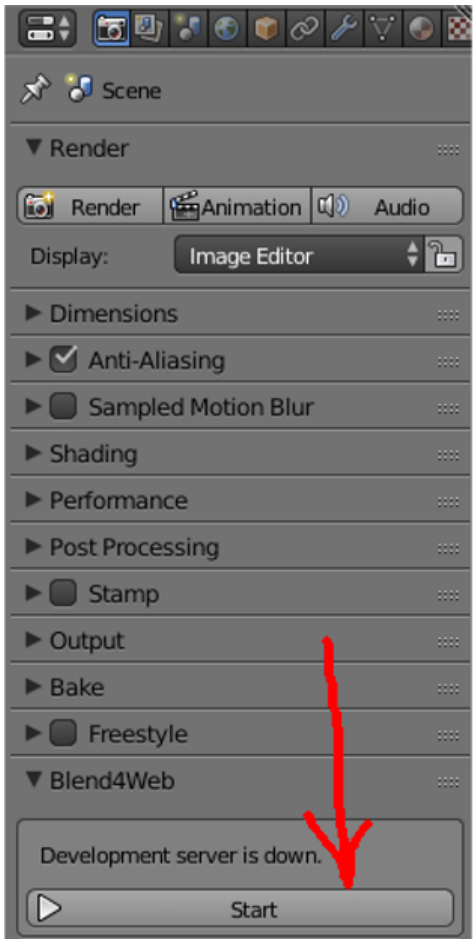
4.3 Local Development Server

Settings for the local development server can be found in File > User Preferences... (hot keys Ctrl-Alt-U). Here you can change the port number to be used to run the development server (6687 by default), and also enable its launching upon Blender startup. To do this, enable Run development server automatically option in the add-on settings.



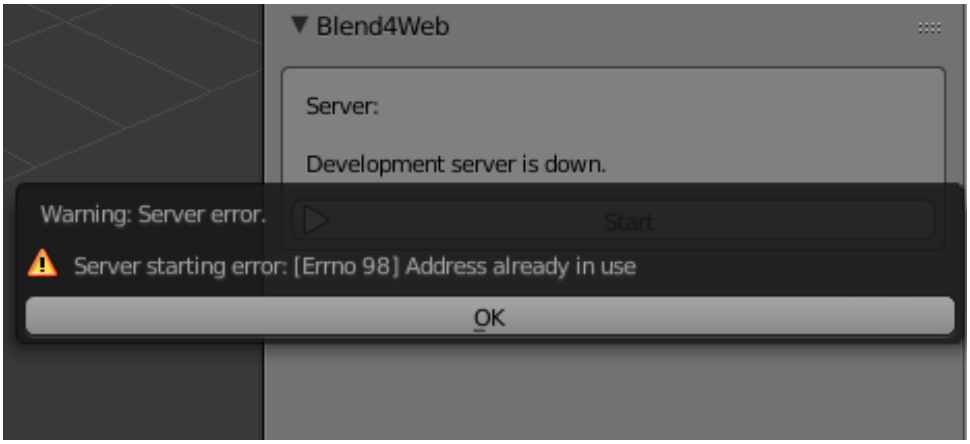
After changing local development server settings in it required to restart Blender.

If you chose not to start the server automatically, you can always do it manually: go to the Render tab and press the Start button on the Blend4Web panel:



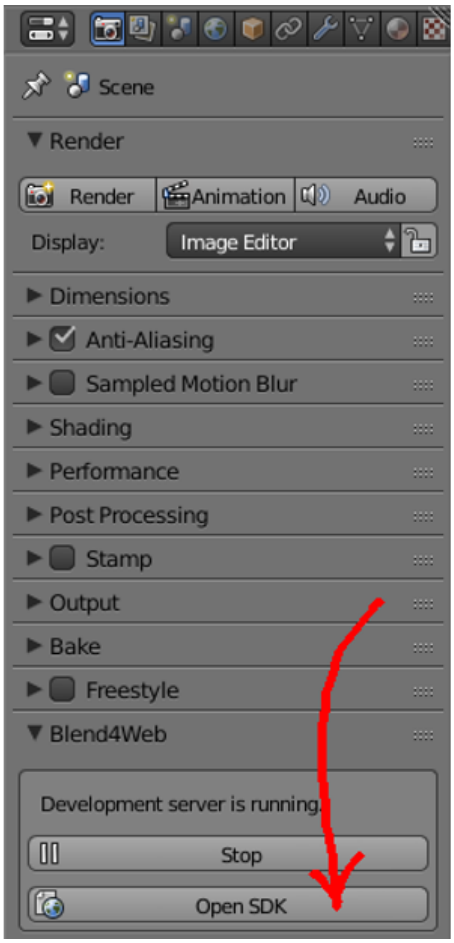
Note: If the path to the Blend4Web SDK is not specified, the local development server cannot be launched. In this case the corresponding message will be displayed instead of the Start button.

If the server is failed to run, an error message will be shown describing the reason:



This error can arise if the server port is already used by some other application.

Press the Open SDK button to open the index web page of the Blend4Web SDK in the browser. This page is available at http://localhost:6687.

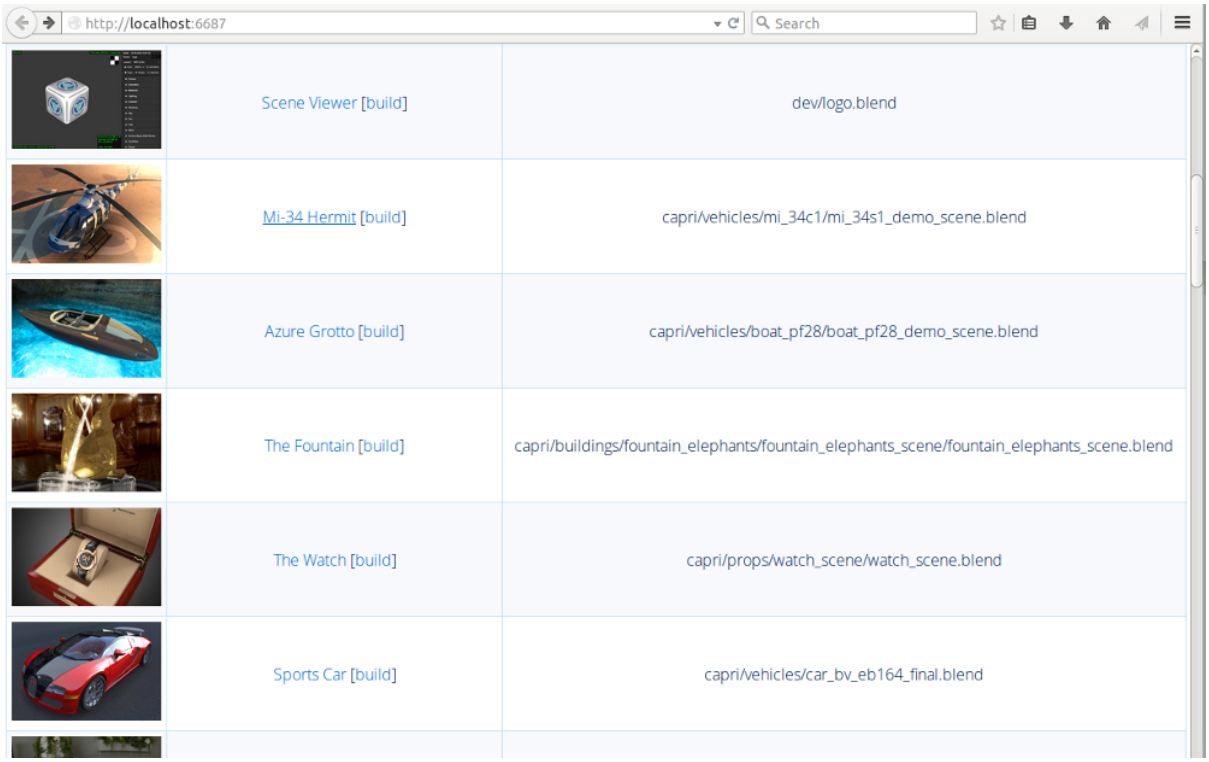


As a result, the default browser for your operating system will be launched.

The server can be stopped by pressing the Stop button. It is also stopped when Blender is closed.

4.4 Running Viewer and Demos

The index page contains links for launching the scene viewer and the demo applications. A WebGL-capable browser is required to run these apps.



Note: If the SDK apps are not displayed correctly, or error messages are shown, follow the instructions in the Problems Upon Startup section.

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 blender/project_name directory.
2. Texture and sound files should be external and located in the 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 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 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.

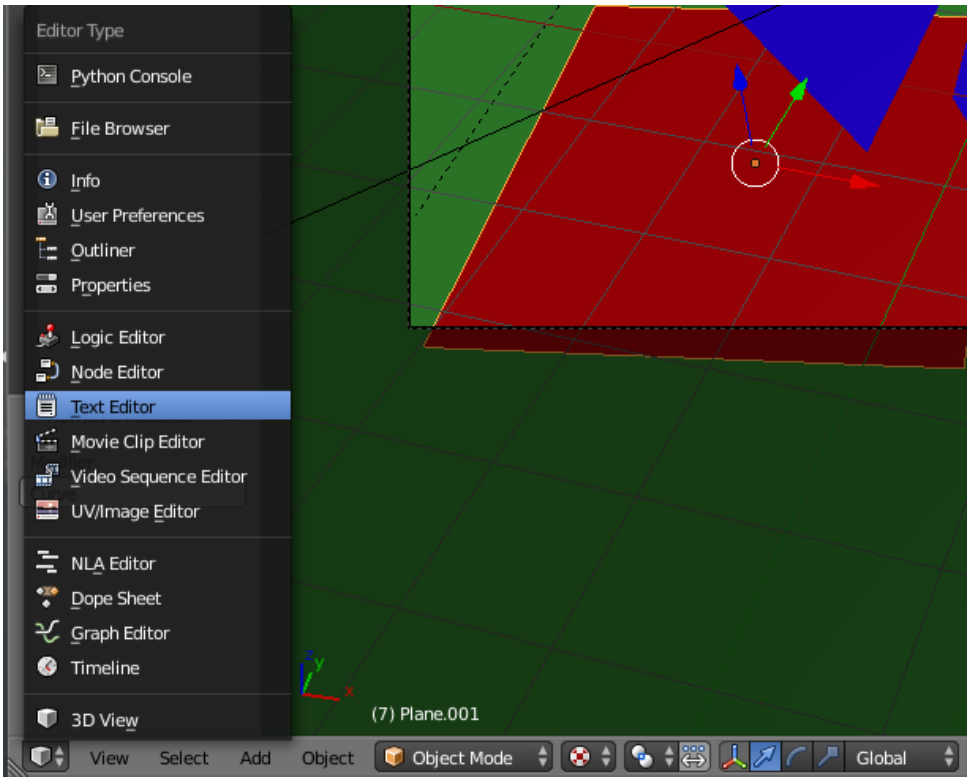
Export options are described in detail in the corresponding section.

5.3 Displaying Scenes in the Viewer

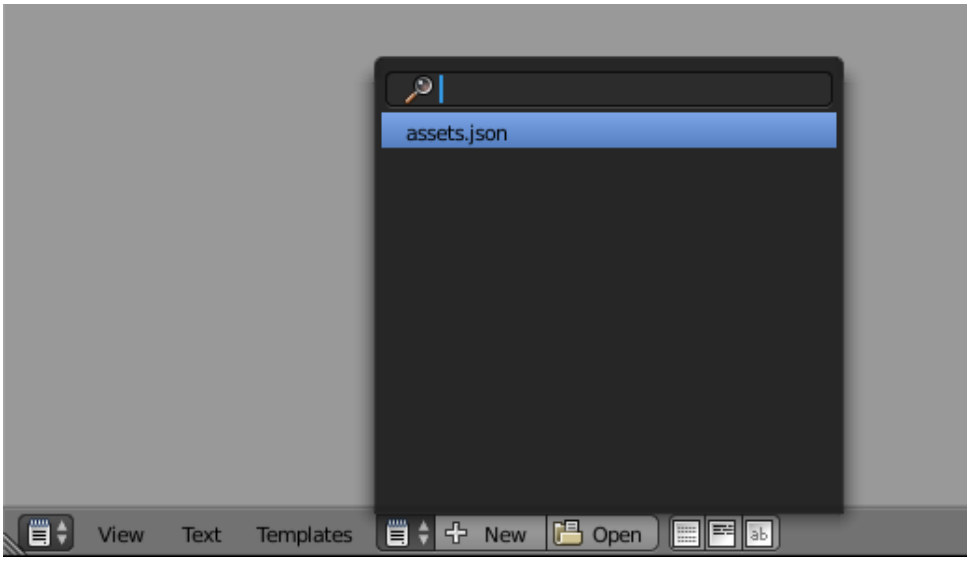
When using the local development server and the Run in Viewer export option, scenes are displayed in the Scene Viewer immediately after the export.

For the long-term storage of the scene in the scene list of the viewer, it's required to manually add the entry to the apps_dev/viewer/assets.json text file. This file is opened automatically in Blender if you are using Blend4Web SDK.

Use the "Text Editor" to modify it:



Then select the assets.json file in the window to edit it:



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.

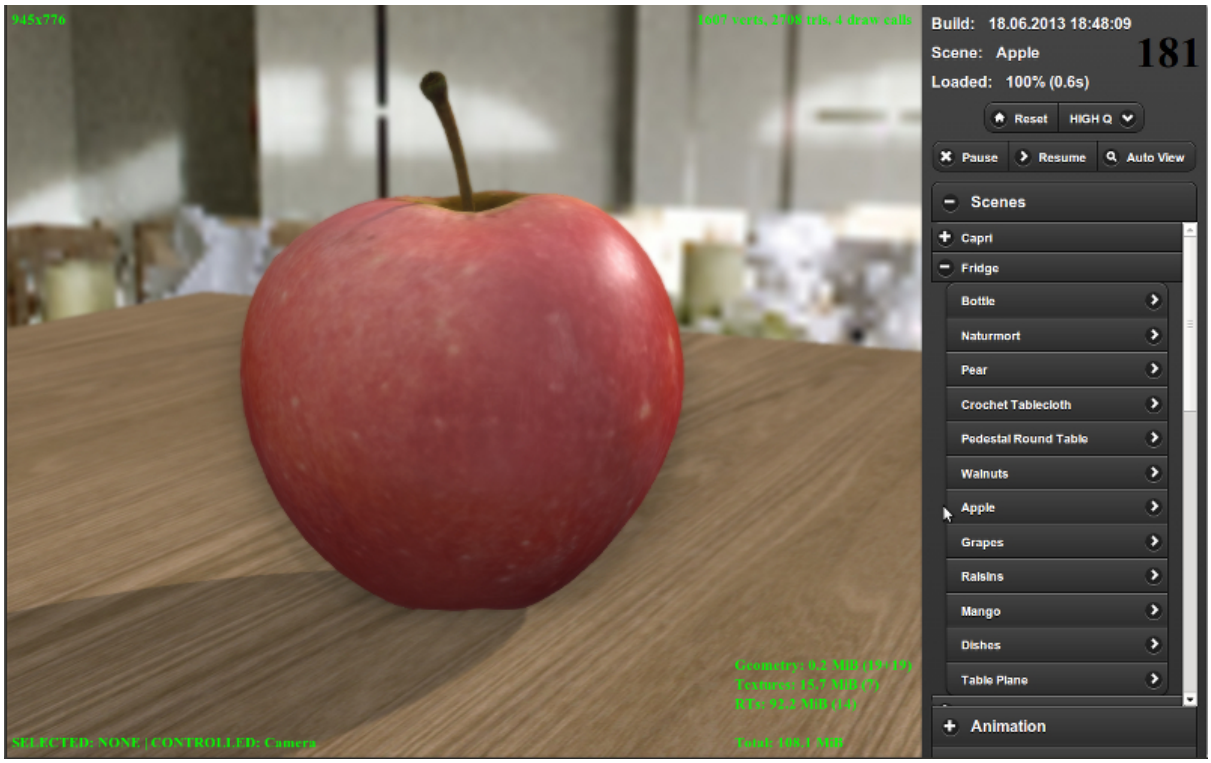
5.3.1 Example

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

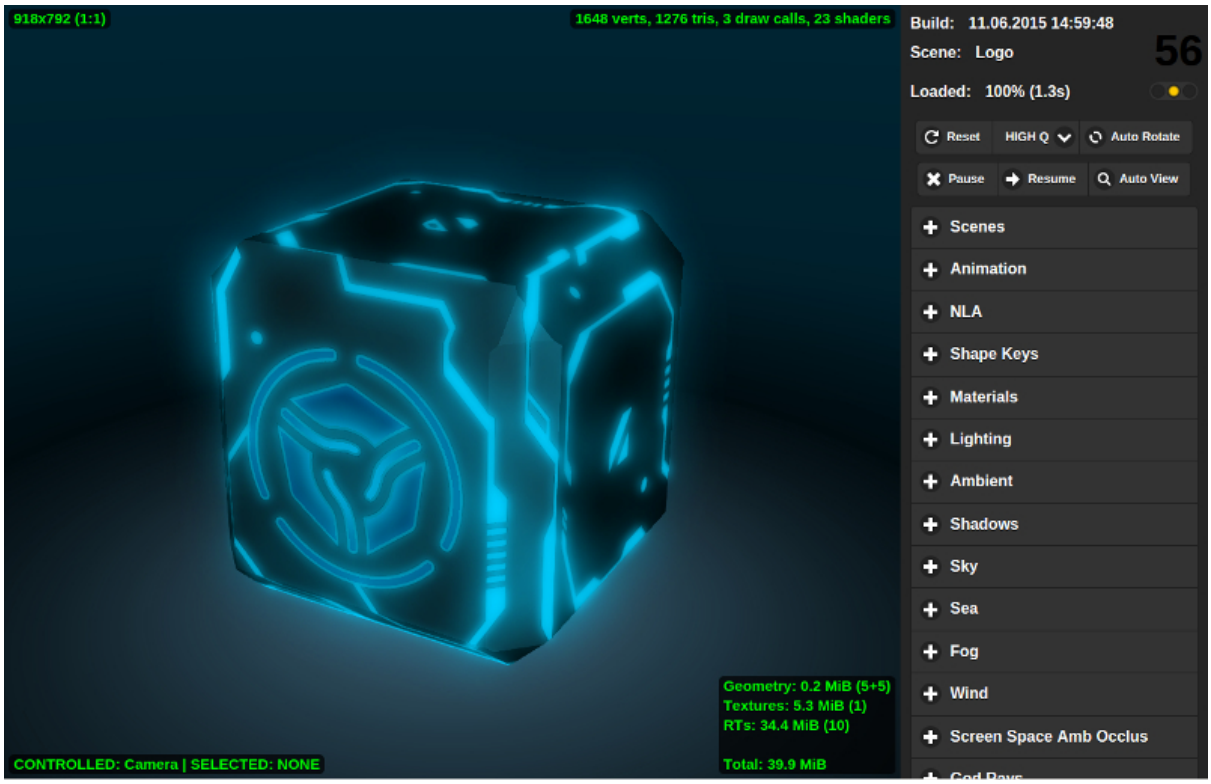
Running The Scenes Viewer.

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).

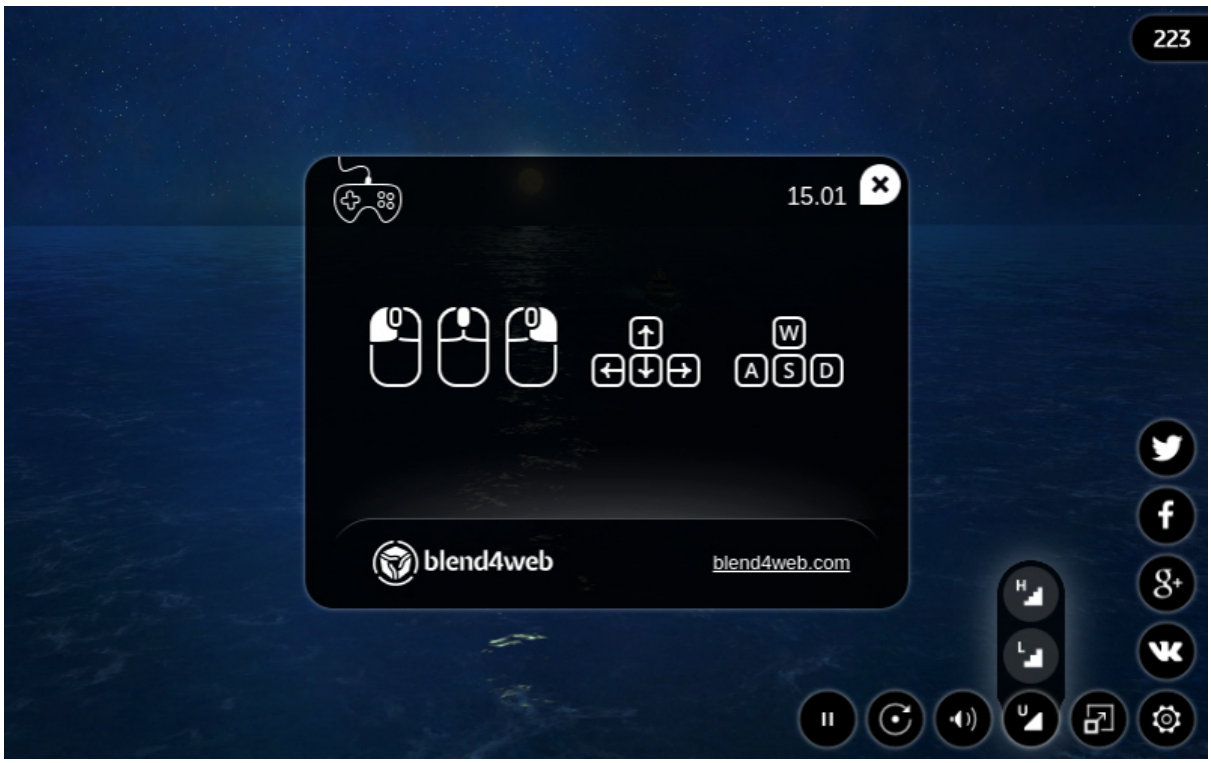
Video memory indicator Is located in the bottom right corner of the rendering area. It displays the amount of video memory used by geometry, textures, render targets, and also the total memory usage.

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

Web Player

The web player is a special application for rendering models and scenes in a demonstration mode.

Desktop version:



Mobile version:

7.1 Use

You can copy the directory containing the web player files, namely deploy/apps/webplayer, from the Blend4Web SDK distribution and deploy it on your web site. You can place the exported scene files on your web site and specify the path to them (absolute or relative) with the load web player parameter.

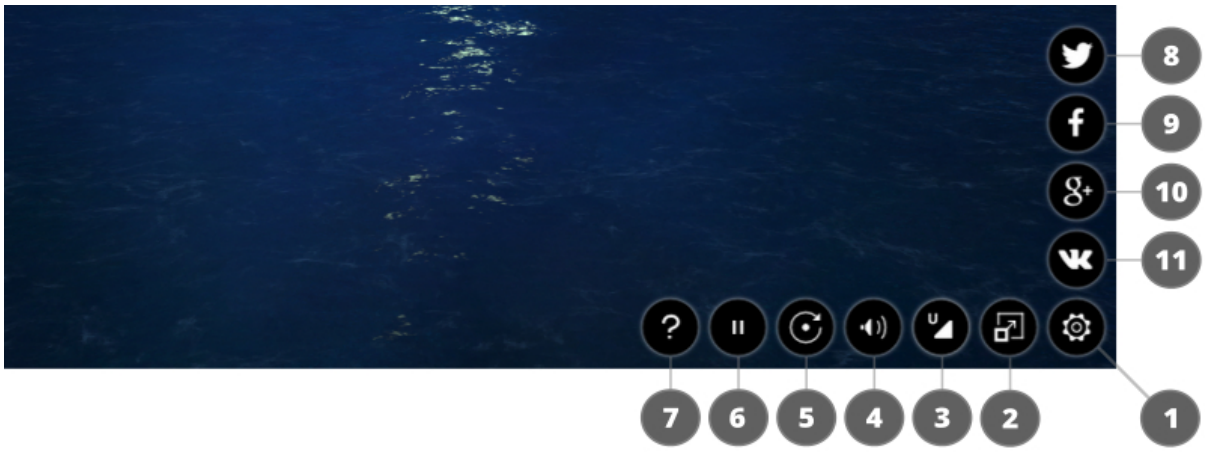
When you export into a single HTML file the web player interface is integrated automatically into it.

7.2 Navigation

The camera (in the Target and Eye modes) is controlled by the mouse with it's button pressed or with the keys: W, A, S, D, R, F (forward, left, back, right, up, down). The numpad keys are also supported.

7.3 Control Panel

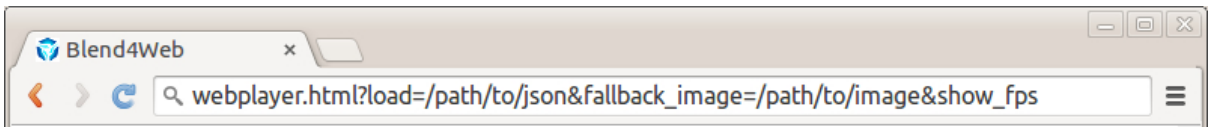
The web player's control panel is shown below.



1. show / hide control panel;
2. fullscreen mode on / off;
3. set the scene quality;
4. sound on / off;
5. camera auto rotation mode on / off;
6. run / stop the engine;
7. open the help window;
8. tweet;
9. share via Facebook;
10. share via Google+;
11. share via VK.

7.4 Attributes

Web player accepts attributes from the browser address line:



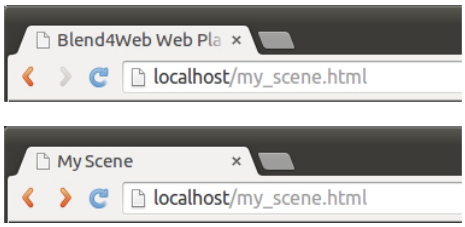
1. the special attribute load is used to load the scene, this attribute contains relative path to a JSON file.
2. in case of a WebGL error the optional fallback_image attribute is used to setup the background image instead of 3D content.

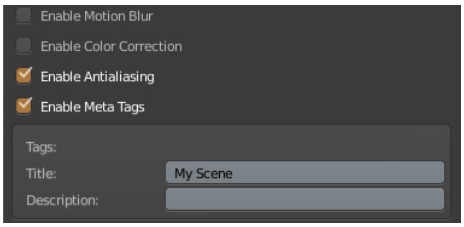
3. in case of a WebGL error the optional fallback_video attribute is used to setup the background video instead of 3D content. Can be used many times to add more video formats.
4. the optional show_fps attribute is used to display the FPS counter in the player's top right corner.
5. optional parameter autorotate is used to enable automatic camera rotation just after the scene loads.
6. the compressed_textures optional parameter is used to enable loading of minified and compressed textures (in DDS format).

Note: If both fallback_image and fallback_video parameters are specified, the fallback_image parameter is used.

7.5 Scene Name as Title

By default the Web Player has the Blend4Web Web Player title. Assigning the meta tag title on the scene in Blender you can change that value to something else.

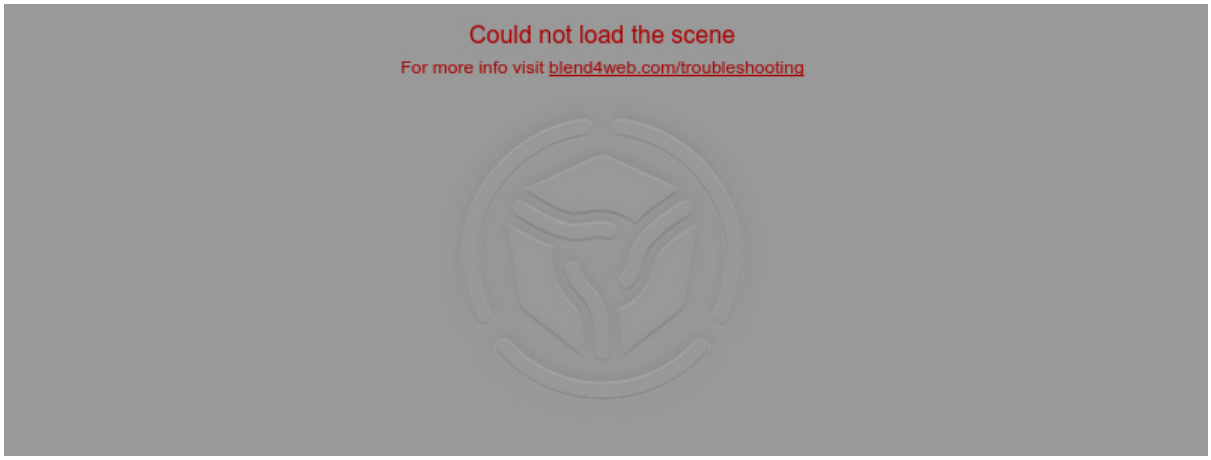




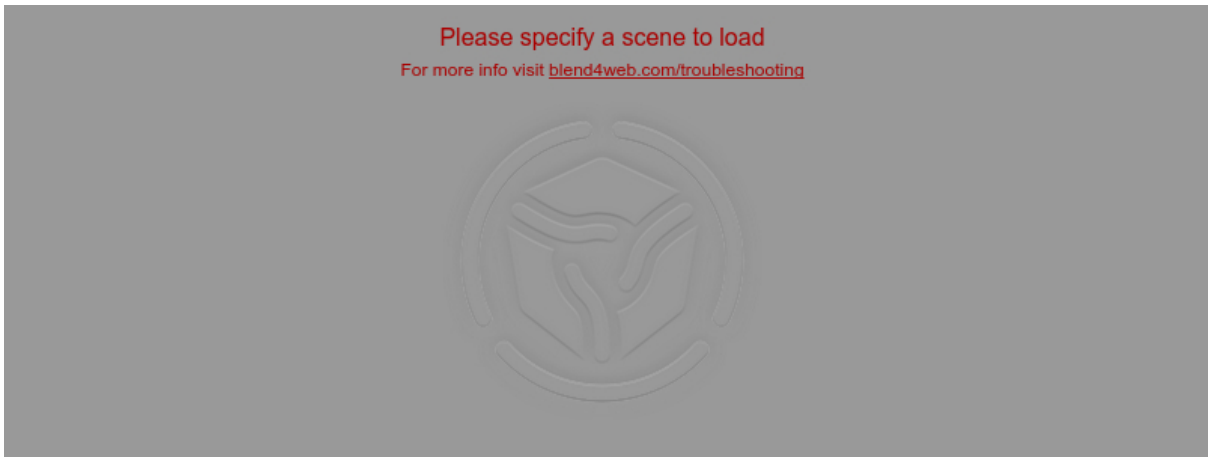
7.6 Scene Errors

If the player is used incorrectly it displays the corresponding errors.

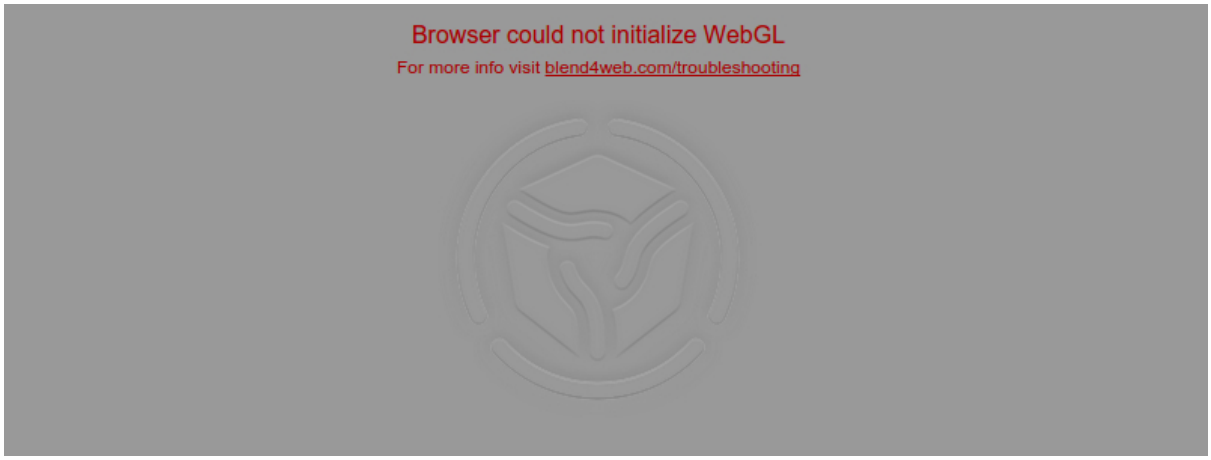
1. the load attribute specifies a wrong path to the JSON file or the file is corrupt;



2. the load attribute is not found or is void;



3. WebGL initialization error.

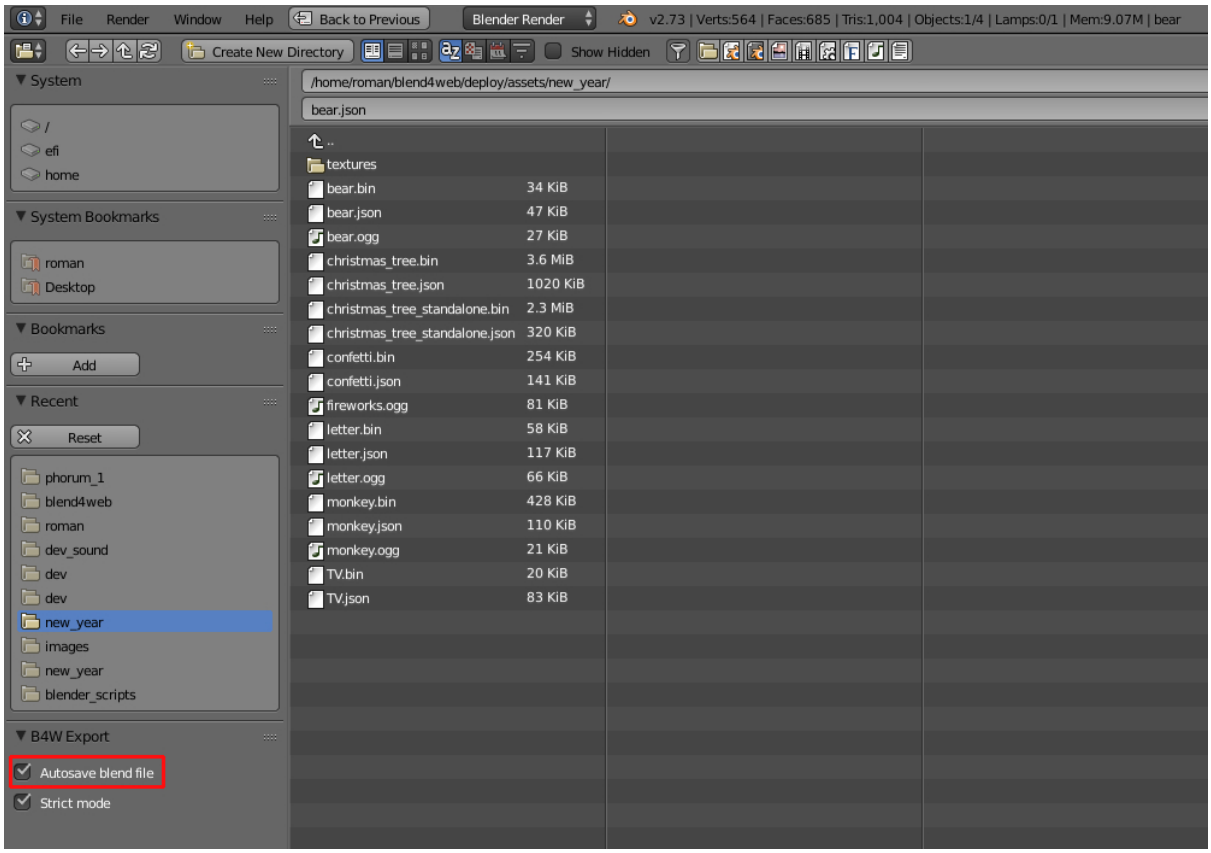


CHAPTER 8

The Add-on

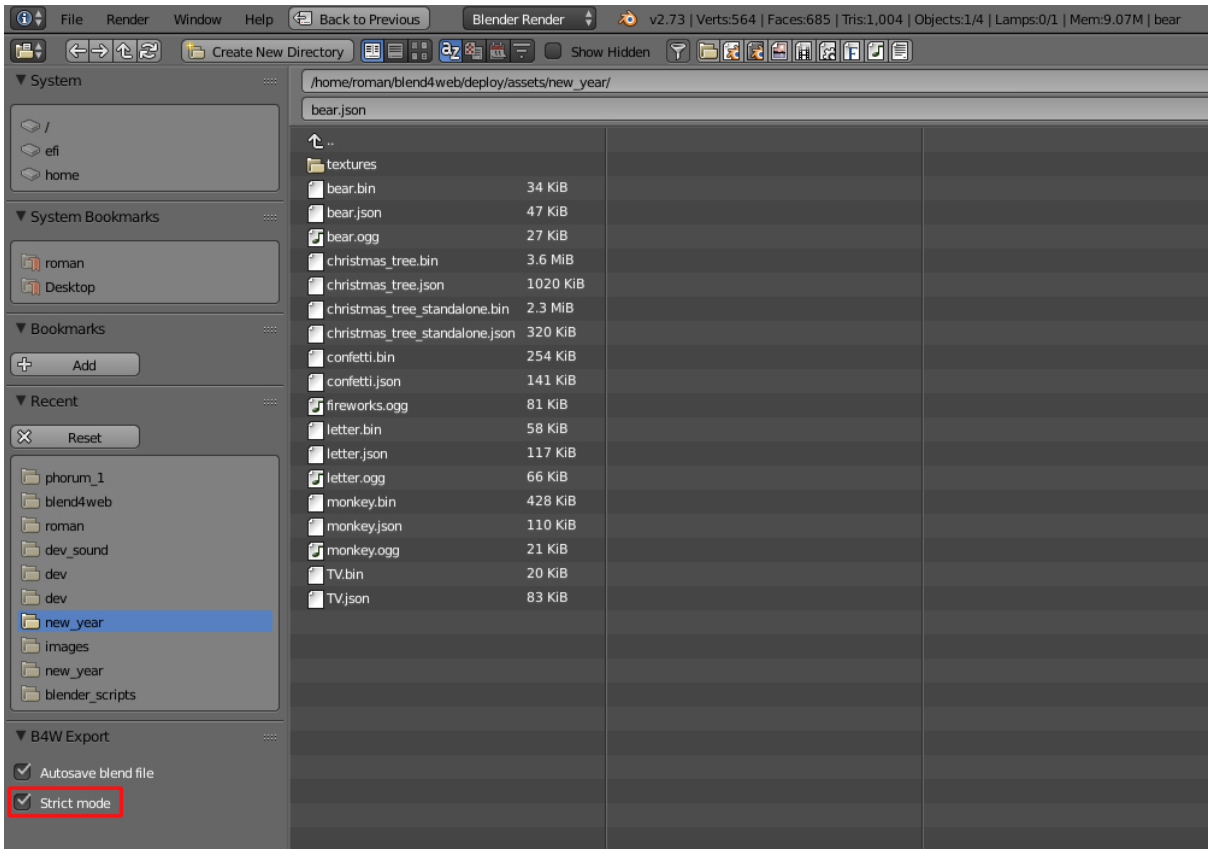
8.1 Export Options

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

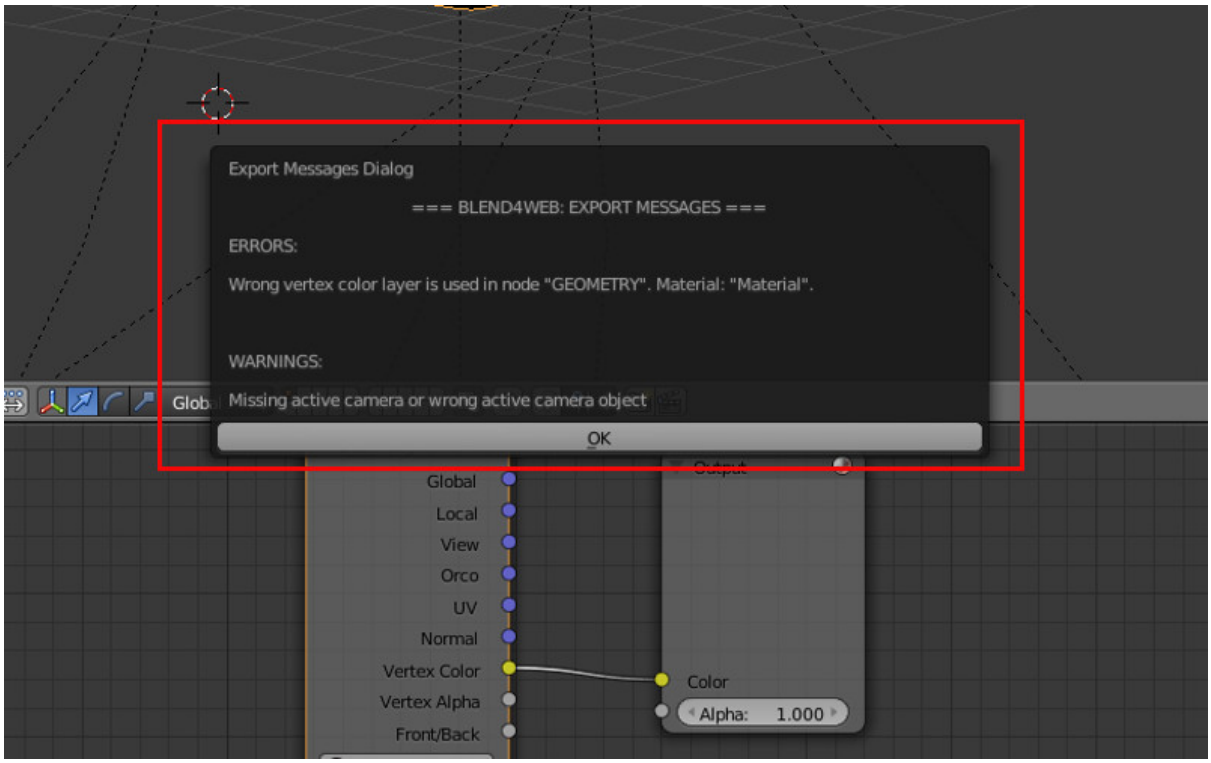


Strict Mode This mode prevents export if there are any errors or messages for users'

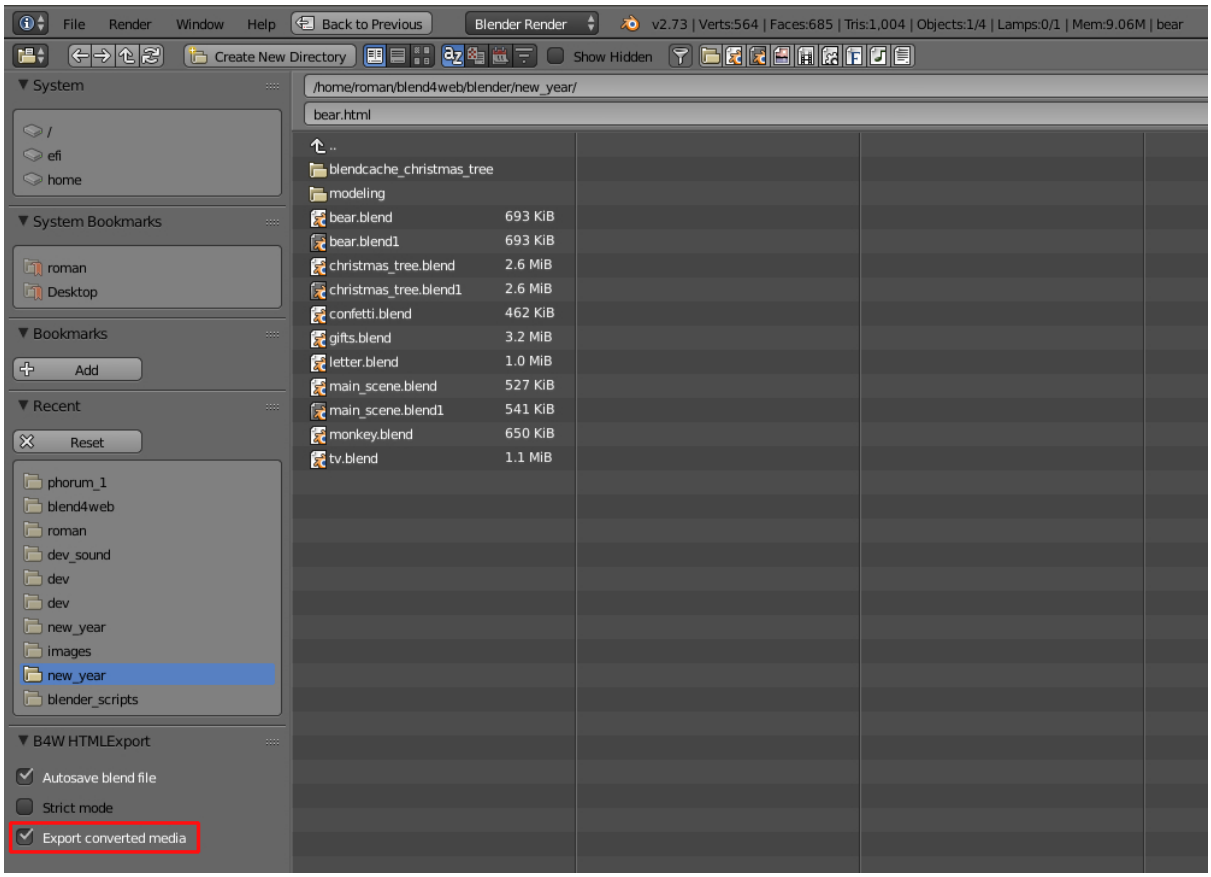
attention. This mode is enabled with the Strict Mode setting in the export menu:



If there are any non-critical errors or messages for users' attention, a dialog window will be show like this:



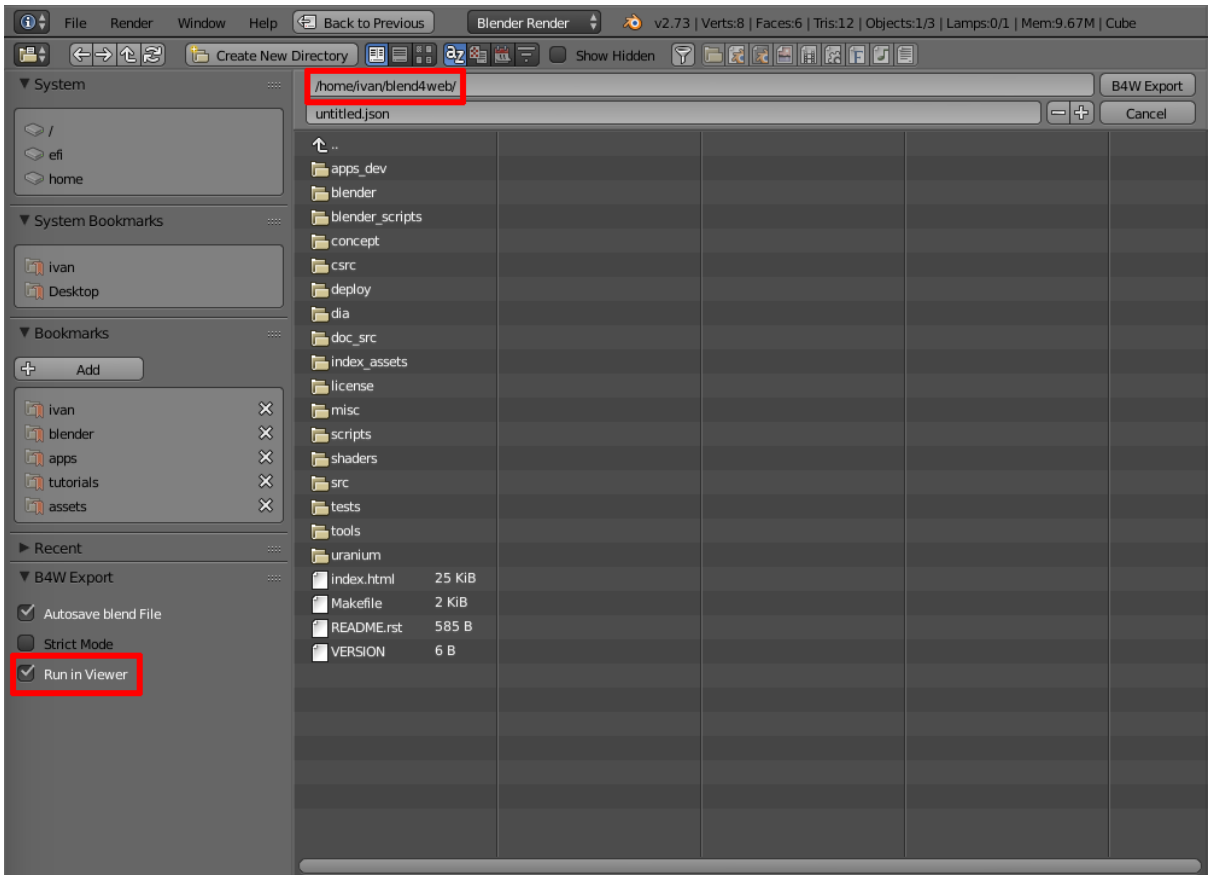
Export Converted Media This option is available for HTML export. When this option is enabled, the converted mediafiles of different formats are written in the HTML file. Using different mediafiles is essential to create cross-browser and cross-platform applications while using HTML export. These files can be created by the converter.



Run in Viewer Automatically launch the Scene Viewer and add the exported scene to it.

When using the local development server, there is a possibility to open the exported .json scene in the Scene Viewer. To do this, select any path inside the Blend4Web SDK file structure upon export.

A directory inside the SDK should be used for export. If not, this option will not be displayed in the menu. Also it will not be displayed if the local development server is down.



8.2 Initialization Errors

Initialization errors can arise upon installation of the add-on 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 add-on 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.

8.3 Compatibility Errors

During the scene viewing, some compatibility errors may occur if the addon version used to export the scene differs from the Blend4Web engine's version.

Error message	Cause
JSON version is too old relative to B4W engine: VER_OLD, required: VER_NEW. Reexport scene with the latest B4W addon to fix it.	Version of the add-on, with which the scene was exported, is too old: VER_OLD. The engine requires: VER_NEW. The scene will not be loaded. We recommend you to reexport the scene using the latest version of the add-on. We also recommend to use the latest version of the engine.
JSON version is a bit old relative to B4W engine: VER_OLD, required: VER_NEW. Some compatibility issues can occur. Reexport scene with the latest B4W addon to fix it.	Version of the add-on, with which the scene was exported, is a bit old: VER_OLD. The engine requires: VER_NEW. The scene will be loaded as usual, however some errors may occur. We recommend you to reexport the scene using the latest version of the add-on. We also recommend to use the latest version of the engine.
B4W engine version is too old relative to JSON. Can't load the scene. Update your engine version to fix it.	Engine version is too old as compared to version of the add-on with which the scene was exported. The scene will not be loaded. We recommend you to use the latest versions of the engine and the add-on.
B4W engine version is a bit old relative to JSON. Some compatibility issues can occur. Update your engine version to fix it.	Engine version is a bit old as compared to version of the add-on with which the scene was exported. The scene will be loaded as usual, however some errors may occur. We recommend you to use the latest versions of the engine and the add-on.

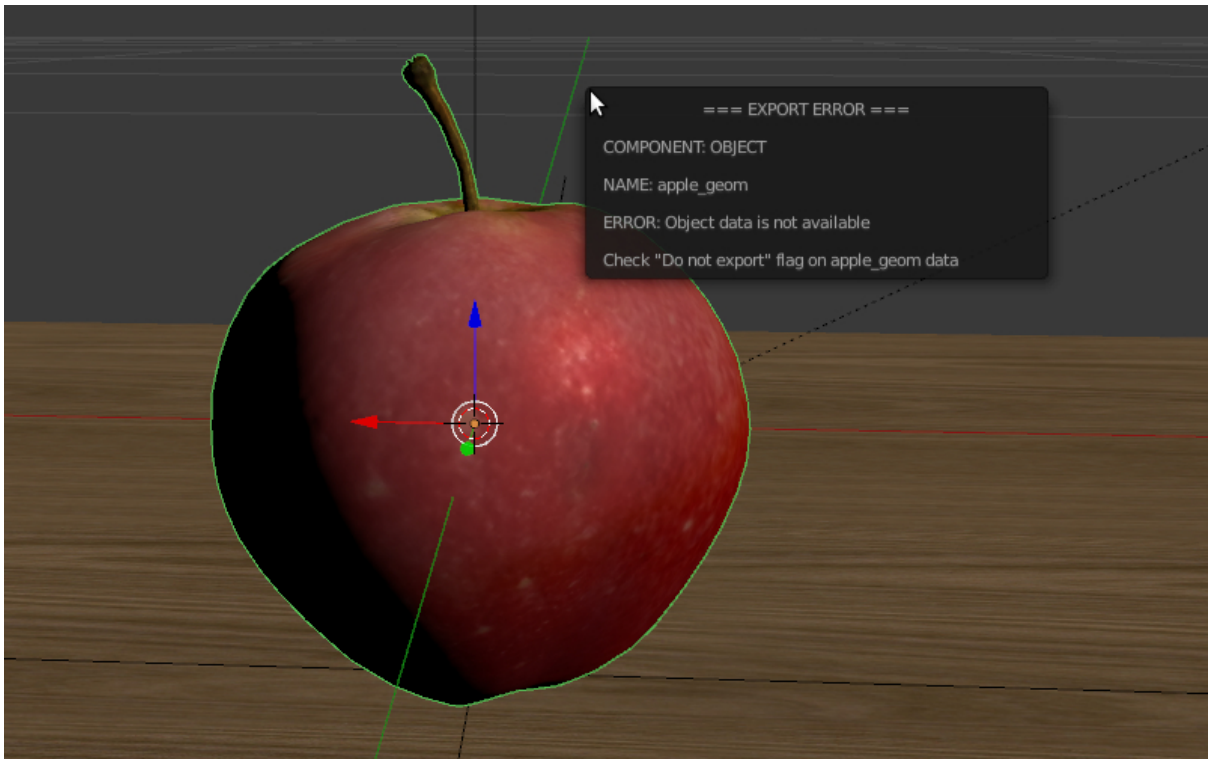
8.4 Critical Export Errors

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.


=== EXPORT ERROR ===
COMPONENT: OBJECT
NAME: apple_geom
ERROR: Object data is not available
Check "Do not export" flag on apple_geom data

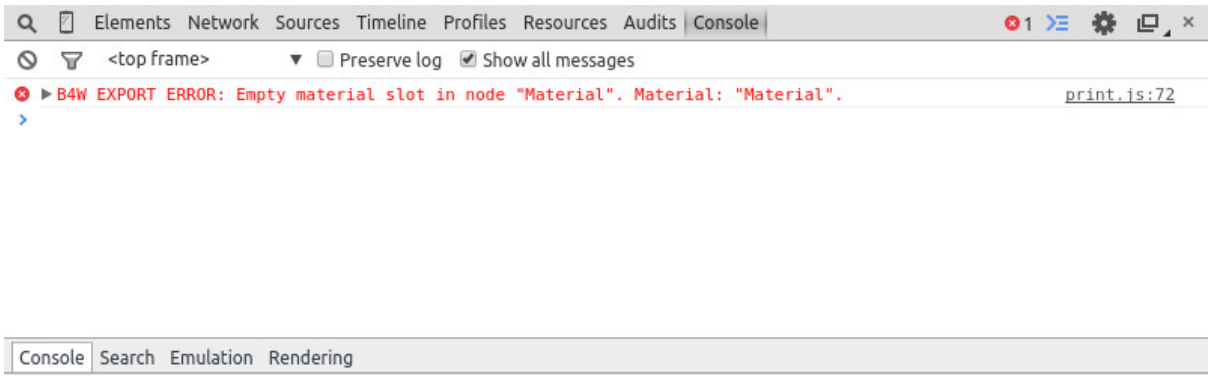
Error message	Cause
Dupli group error; Objects from the GROUP_NAME dupli group on the OBJECT_NAME object cannot be exported	None of the objects in the GROUP_NAME group which were selected for duplication on the OBJECT_NAME object can be exported. Permission to export at least one object of the group, or to remove the duplication of the group is required.
Export to different disk is forbidden	Export to a directory located on a different disk is forbidden
Incompatible objects with a shared mesh; The OBJECT_NAME object has both vertex groups and a shared mesh	Incompatible objects with 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, Apply scale options turned on (because in these cases a full copying of meshes occurs).
Incomplete mesh; Material slot is empty	Material slot is empty.
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.
Loading of resources from different disk is forbidden	Loading of resources from different disk is forbidden.
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 lamp	There should be at least one light source in the scene.
No such file or directory	The file or directory does not exist.
Object constraint has no target	The Target Object property for the object constraint (on the Object Constraints tab) was not set.
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; 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.
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 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.

8.5 Non-Critical Export Errors

In contrast to the above-listed critical export errors, these errors do not prohibit the export, but can make scenes displayed incorrectly. These messages can be viewed in the browser console (opens with F12) when a scene is loaded. The message looks like this:

B4W EXPORT ERROR: Error message


Elements Network Sources Timeline Profiles Resources Audits Console
1
<top frame>
Preserve log
Show all messages
B4W EXPORT ERROR: Empty material slot in node "Material". Material: "Material".
print.js:72
Console Search Emulation Rendering

Error message	Cause
Canvas texture ID NAME already exists. Texture NAME.	This Canvas ID already exists.
Empty canvas texture ID for texture NAME.	Canvas ID is empty.
Empty material slot in node “NAME”. Material: “NAME”.	Empty material slot in “NAME” node.
Exported UV-layer is missing in node “GEOMETRY”. Material: NAME.	The UV layer specified in the GEOMETRY node is not exported (for UV texture coordinates).
Exported UV-layer is missing in texture NAME. [Material: NAME.]	Exported UV-layer is missing in the texture (for UV texture coordinates).
Ignoring LODs after empty LOD for the NAME object.	All LOD objects that follow the empty slot were ignored (in the LOD objects list for the NAME object).
Incomplete mesh NAME; 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; 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.
Incorrect NLA script, falling back to simple sequential NLA.	Incorrect NLA script, falling back to simple sequential NLA.
Invalid link found in node material. Material: “NAME”.	The “NAME” node material contains an incorrect link between nodes.
No image in the NAME texture. [Material: NAME.]	The texture has no image.
No texture for the NAME particle settings texture slot.	No texture in the particle settings’ texture slot.
No texture in the NAME world texture slot.	No texture in the NAME world’s texture slot
No texture in the texture slot. Material: NAME.	There is no texture in the material texture slot.
Node material invalid: “NAME”. 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 “NAME” has the mesh with shape keys. The property “Relative” of mesh has been enabled.	An object named “NAME” has a mesh with shape keys. This mesh has the “Relative” property enabled which is forbidden.

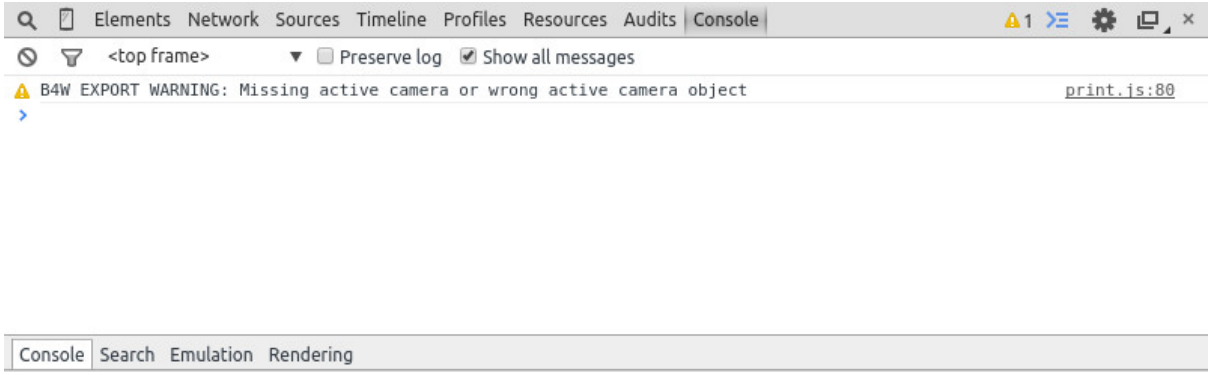
Only 2 UV textures are allowed for a mesh; The mesh has N UVs.	The engine supports up to 2 UV texture layers for each mesh. The number of UV layers for this mesh is N.
Particle system error for “NAME”; The “NAME” vertex color specified in the from field is missing in the last of the “OBJECT_NAME” object’s vertex colors	The NAME vertex color is specified in the from field but it not present in the OBJECT_NAME emitter.
Particle system error for “NAME”; 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 for “NAME”; 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.
The main scene NAME can not be rendered by another scene. Material NAME has been removed.	The main scene NAME can not be rendered by another scene. The material NAME has been deleted.
The main scene NAME can not be rendered by another scene. Texture NAME has been removed.	The main scene NAME can not be rendered by another scene. The texture NAME has been deleted.
The NAME action has decimal frames. Converted to integer.	The NAME action has decimal frames. Converted to integer.
The NAME armature modifier has a proxy object as an armature. Modifier removed.	An armature modifier has a proxy object as an armature.
The NAME armature modifier has no armature object or it is not exported. Modifier removed.	The NAME Armature modifier has no armature object or it is not exported. Modifier removed.
The NAME curve modifier has no curve object. Modifier removed.	The NAME curve modifier has no object. Modifier removed.
The NAME curve modifier has unsupported curve object. Modifier removed.	The NAME curve modifier has unsupported object. Modifier removed.
The NAME object has the NAME armature modifier and a vertex animation. Modifier removed.	The NAME object has both vertex animation and an armature modifier which is not supported. As a result, the modifier has been removed.
The NAME LAMP node has no lamp object. Material: NAME.	Wrong object specified in the NAME LAMP node.
The NAME node is not supported. The NAME material will be rendered without nodes. Material: NAME.	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.
The NAME object has NAME armature modifier which references the wrong group. Modifier removed.	An object should be in the same group as an armature, or both these objects should be explicitly present in the scene.

Using B4W_REFRACTION node NODE_NAME with incorrect type of Alpha Blend. Material: NAME.	A node material with incorrect Alpha Blend property is used. Alpha sort, Alpha blend and Add are allowed when using a “REFRACTION” node.
Wind bending: not all vertex colors exist for “NAME”. Properties were set to default values.	Wind bending parameters setup: all specified vertex color layers should exist.
Wind bending: vertex colors weren’t properly assigned for “NAME”. Properties were set to default values.	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.
Wrong “Height Map” input for the “NAME” B4W_PARALLAX node. Only link from the TEXTURE node with a non-empty texture is allowed.	Wrong data were passed to the “Height Map” input of the NAME B4W_PARALLAX node. Only the output from a non-empty TEXTURE node is allowed.
Wrong texture coordinates type in texture NAME. [Material: NAME.]	The following coordinate types are supported for image textures: UV, Normal and Generated.
Wrong vertex color layer is used in node “GEOMETRY”. [Material: NAME.]	Wrong vertex color layer is used in a “GEOMETRY” node.

8.6 Other Messages

These messages can be viewed in the browser console (opens with F12) when a scene is loaded. The message looks like this:

> B4W EXPORT WARNING: Export message which requires the user’s attention



Error message	Cause
Missing active camera or wrong active camera object	There is no active camera on the scene (Camera property on the Scene tab).
Missing world or wrong active world object	There should be at least one world datablock in the scene.
NAME particle settings has the NAME texture rendering a scene. It has been replaced by the default texture.	The particle settings datablock NAME contains the texture NAME being used for rendering a scene into. This texture has been replaced by a default texture.
The “NAME” camera has unsupported PANORAMIC type. Changed to PERSPECTIVE type.”	Panoramic cameras are not supported. Perspective mode is used instead.

CHAPTER 9

Objects

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

9.1 Types

The engine supports objects of the following types:

- mesh
- camera
- lamp
- empty
- armature
- speaker
- curve
- text
- metaball
- surface

9.2 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.

Merging of static objects - so called batching - is performed automatically when the scene is loaded in order to optimize the number of draw calls. The conversion is performed even

if there is just one object in the scene. The center of the resulting mesh is located in the origin.

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

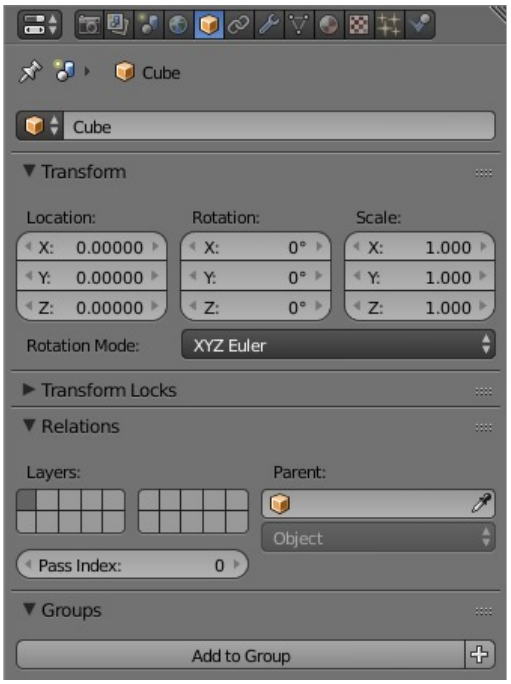
The objects which have animation, physics or a parent, which is a dynamic object, are considered dynamic.

Object movement via API is possible only for dynamic objects. In order to make the movement of the object without dynamic settings possible, it is necessary to activate Force Dynamic Object option in its settings.

9.3 Settings

The following is supported for all types of objects: transform, data reference, parent object, group membership and a set of the Blend4Web's special properties.

9.3.1 Object Tab



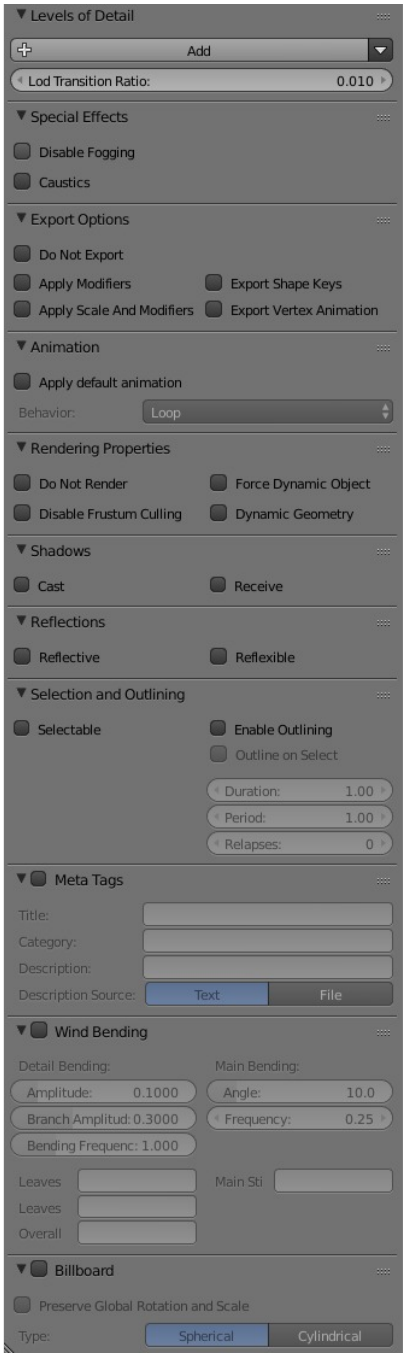
Transform > Location Position coordinates.

Transform > Rotation Rotation angles. For the object rotation all available modes can be used (Rotation Mode). However only Quaternion (WXYZ) and XYZ Euler are supported for object animation.

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

Relations > Parent Reference to the parent object.

Groups Objects' groups to which this object belongs.



Levels of Detail > LOD Transition Ratio Parameter for smoothing the switching between the LOD objects. It defines the additional distance at which the LOD objects are still rendered before getting replaced by the next LOD objects. Assigned for

the main object. Measured in fractions of the object's bounding sphere radius.

Special Effects > Disable Fogging Disable fog for the object.

Special Effects > Caustics The object will render caustics effects from the adjacent water.

Export Options > Do Not Export Do not export this object.

Export Options > Apply Modifiers Apply the object's modifiers upon export. If the SKIN modifier is used we recommend to apply it before the export because it resets vertex color and UV layers which may result in errors.

Export Options > Apply Scale and Modifiers Upon export, apply scale and modifiers for the object.

Export Options > Export Vertex Animation Export previously created and saved vertex animation. Applicable for MESH objects only.

Export Options > Export Shape Keys Export shape keys. Applicable to MESH objects only.

Note: The following properties are mutually exclusive: Apply Modifiers, Apply Scale and Modifiers, Export Vertex Animation and Export Shape Keys.

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

Animation > Animation blending Only for armature objects. Allows blending between skeletal animations.

Animation > Behavior Animation behavior when the last frame is reached: Finish Stop - stop, Finish Reset - stop and go to the zero frame, Loop - repeat forever.

Rendering Properties > Do Not Render Disable object rendering (for example useful for a physics object).

Rendering Properties > Disable Frustum Culling Disable frustum culling optimization.

Rendering Properties > Force Dynamic Object Force the object to become a dynamic object.

Rendering Properties > Dynamic Geometry Allow overriding of the object's geometry through Blend4Web API.

Shadows > Cast The object will cast shadows.

Shadows> Cast Only The object will cast shadows but will remain invisible itself. Becomes available after enabling Shadows > Cast.

Shadows > Receive The object will receive shadows from other adjacent objects.

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

Reflections > Reflexible only The object will be reflected but will remain invisible itself. Becomes available after enabling Reflections > Reflexible.

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

Reflections > Reflection Plane Text field for name of an empty object which defines the reflection plane. Becomes available after enabling Reflections > Reflective.

Selection and Outlining > Selectable Enable object selection with the mouse or another input device.

Selection and Outlining > Enable Outlining Enable outline glow for the object.

Meta Tags Interface for adding meta tags to the object:

Meta Tags > Title Object's title.

Meta Tags > Category Object's category.

Meta Tags > Description Description for the object. Depending on Description Source, this field accepts either description text itself, or the name of a file where this description is contained.

Meta Tags > Description Source Source type for the description: text or text file.

Anchors > Enable Anchor Available for EMPTY objects only. Interface for adding anchors (2D tags) to objects:

Anchors > Type Anchor type. Annotation - the content is obtained from the meta tags assigned to the object and displayed in a collapsible window of standard design. Custom Element - arbitrary HTML element from the current web page is used as anchor. Generic - anchor's position can be detected using anchors module API.

Anchors > Detect Visibility Detect whether the anchor object is overlapped by other objects. Turning this option on decreases performance and should be used only when necessary.

Anchors > Max Width Applicable to annotation anchors. This limits the expanding info window by a predefined value (measured in CSS pixels).

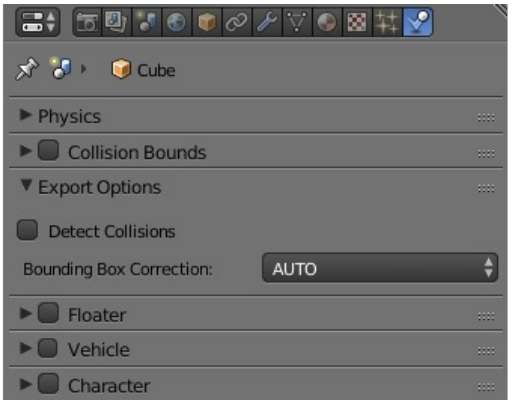
Wind Bending Enable procedural animation under the influence of wind.

Billboard Use the object as a billboard (i.e. automatically orient relative to the camera).

Billboard > Preserve Global Orientation and Scale Take into account rotation and scale of the billboard object (in the world space). The object will be directed toward the camera with its side which is visible when viewing along the Y axis in Blender. Becomes available after enabling the Billboard checkbox.

Billboard > Billboard Type Billboard orientation mode. Spherical (by default) - the object is always oriented with one side toward the camera, regardless of view angle, Cylindrical - similar to Spherical, but rotation is limited to Blender's world Z axis. Becomes available after enabling Billboard.

9.3.2 Physics Tab



Detect Collisions Activate the object's physics.

Floating Make the object floating. The settings for floating objects are described in detail in the physics section.

Vehicle Use the object as part of a vehicle. The vehicle settings are described in detail in the physics section.

Character Use the object for character physics. The character settings are described in detail in the physics section.

9.4 Object Transform API

Note: Make sure that the object you are trying to transform 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).

9.5 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.

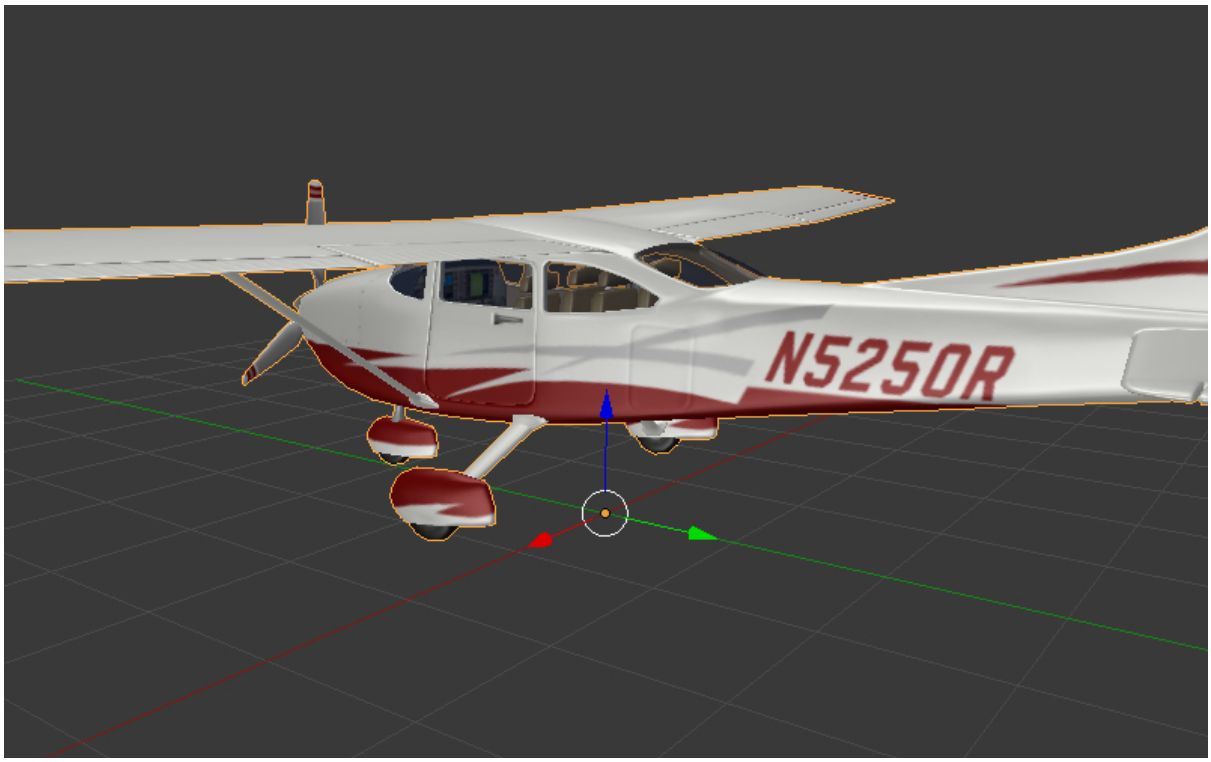
quat.setAxisAngle A quaternion is an alternative presentation of rotation by an arbitrary angle relative to the arbitrary axis (vector). Positive direction of rotation is defined as anticlockwise when viewing from the vector's end. For example the quat.setAxisAngle([1, 0, 0], Math.PI/2, quat) call forms 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.

9.5.1 Quaternion Example

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

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

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

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

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
m_quat.setAxisAngle(AXIS_Y, -Math.PI/3, quat_tmp);
m_transform.get_rotation(aircraft, quat_tmp2);
m_quat.multiply(quat_tmp, quat_tmp2, quat_tmp);
m_transform.set_rotation_v(aircraft, quat_tmp);
```

9.6 Moving via TSR Vectors

It is sometimes convenient to move objects using vectors of the following format:

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

Here T_x, T_y, T_z - the components of the translation vector, S - scale factor, R_x, R_y, R_z, R_w - the components of the quaternion vector. Hence the name of this vector: TSR or TSR-8.

This vector can be operated via tsr module, as well as via set_tsr()/get_tsr() methods of the transform module.

9.7 Copying Objects (Instancing)

It is often required to copy (to make instances of) objects during application work.

Copying objects has its limitations:

- only MESH objects can be copied
- the object should be dynamic (enable Rendering Properties > Force Dynamic Object)
- the source object should belong to the active scene

9.7.1 Making a Simple Copy

In case of simple copying the new object will share the mesh with the original object. Thus, if the the original object's mesh is changed, the copied object's mesh will be changed

too. To make simple copying possible, it's enough to turn on the Blend4Web > Force Dynamic Object setting in the source object's properties.

9.7.2 Making a Deep Copy

In case of deep copying, the new object will have unique properties, namely it will have its own mesh. Thus, if the original object's mesh is changed, the copied object's mesh will not be changed. To make deep copying possible, it is required to enable the Rendering Properties > Force Dynamic Geometry checkbox for the source object.

Copying objects in runtime can be performed with the copy method of the objects.js module. This method requires three arguments: the id of the source object, a unique name for the new object and the boolean value to specify the copy mode (i.e. simple or deep). By default simple copying will be performed.

The newly created object should be added to the scene. This can be performed with the append_object method of the scenes.js module. The new object should be passed to it as an argument.

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

9.7.3 Removing Objects

To remove instanced objects, use the remove_object method of the scenes.js module. Pass the object to it as an argument. Only copied objects can be removed this way.

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

9.8 Object Selection

In order to enable selection of a certain object, it is required to enable the Selectable checkbox on the Selection and Outlining panel.

Note: Make sure that the status on the Scene > Object Outlining panel is set to ON or AUTO.

Object selection is possible programmatically via API, for example, in the scenes.js module there is the pick_object function which selects an object based on canvas 2D coordinates,

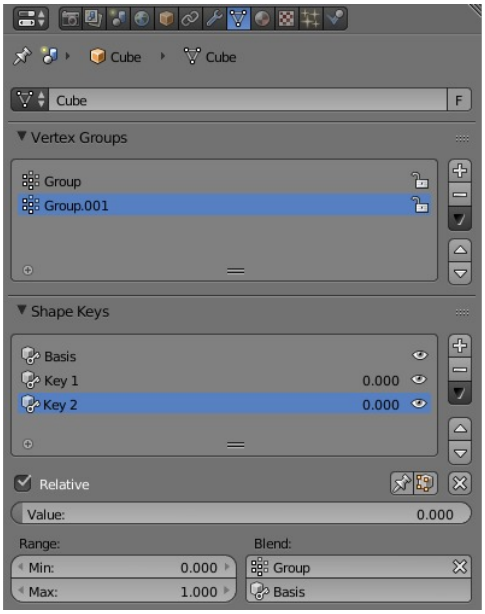
```
// ...
var x = event.clientX;
var y = event.clientY;

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

or using the NLA Script.

If the selectable object has enabled Enable Outlining and Outline on Select checkboxes on the Object > Selection and Outlining panel, then the pick_object function call will activate outline glow animation.

9.9 Morphing

Morph targets can be added using Blender's standard Mesh > Shape keys interface.



The engine supports all shape key options under the "Relative" type.

To set a shape key value, use the apply_shape_key method of the geometry.js module.

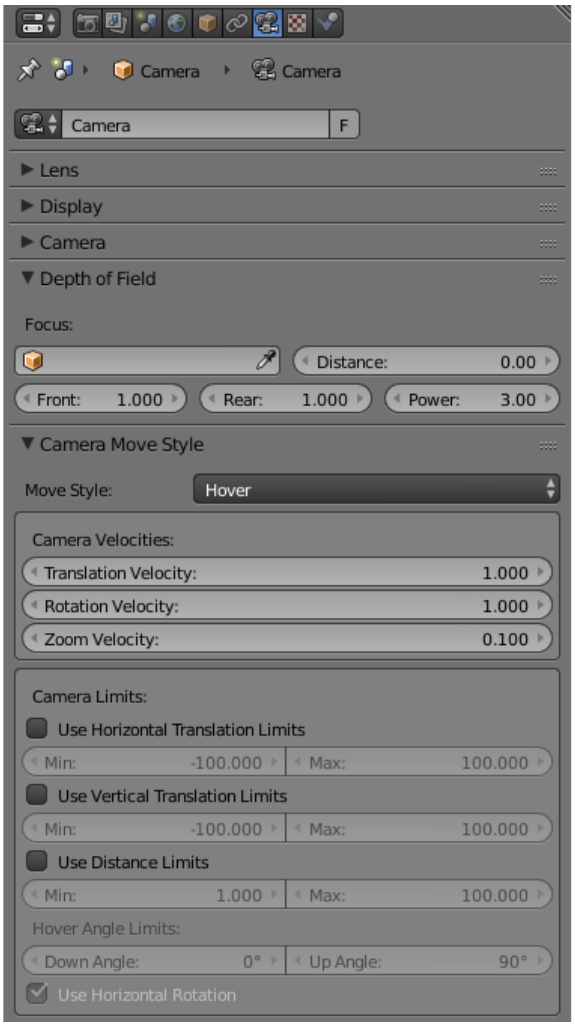
Note: The object must have Export Shape Keys enabled.

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

Camera

10.1 Move Styles and General Settings

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



Camera Move Style > Move Style Camera control mode:

- Target By default the camera is rotating around a fixed point (target). The pivot's position can be changed (see camera panning).
- Eye The Eye mode allows rotation and translation as in first person view.
- Hover In the Hover mode the camera is gliding over the horizontal plane.
- Static In the Static mode the camera can be moved via animation or through API calls.

Camera Move Style > Target Location Available for 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.

Depth of Field Described in the Depth of Field section.

10.2 Camera Movement Velocity

Velocity settings of different type are available for camera movement.

Camera Move Style > Camera Velocities > Translation Velocity Available for the Target, Eye and Hover modes. Defines camera translation velocity.

Camera Move Style > Camera Velocities > Rotation Velocity Available for the Target, Eye and Hover modes. Defines camera rotation velocity.

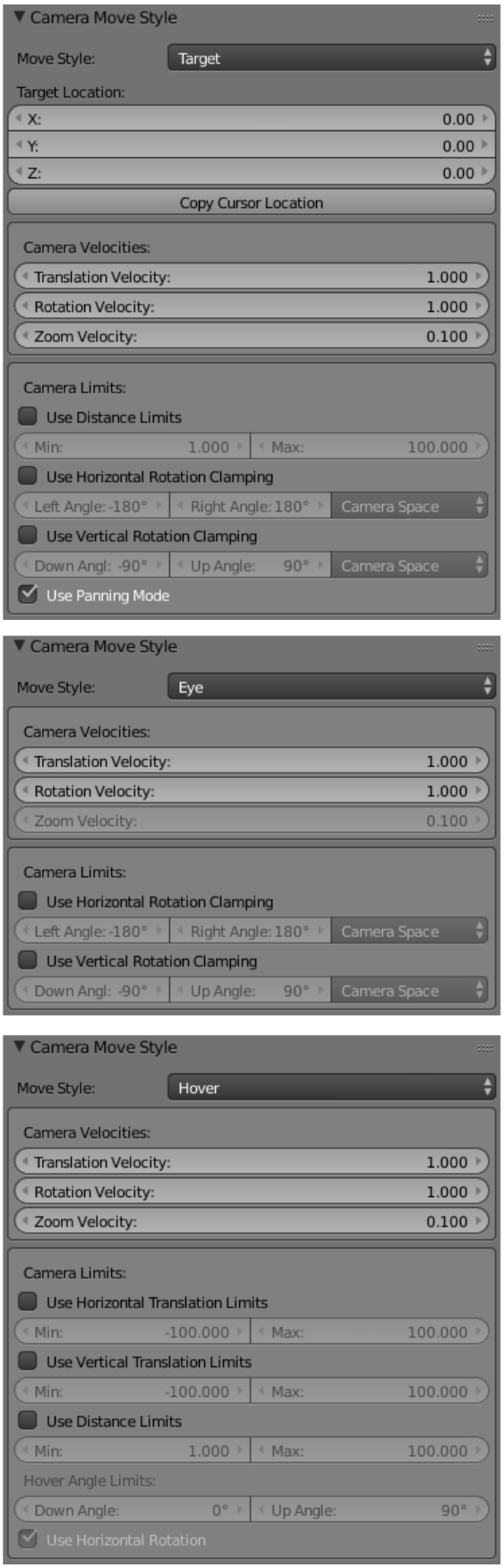
Camera Move Style > Camera Velocities > Zoom Velocity Available for the Target and Hover modes. Defines velocity of approaching of the camera to the pivot point.

Default values for the parameters: Translation Velocity, Rotation Velocity: 1; Zoom Velocity: 0.1.

Allowable values for the parameters: Translation Velocity, Rotation Velocity: $[0, \infty)$; Zoom Velocity: $[0, 1)$.

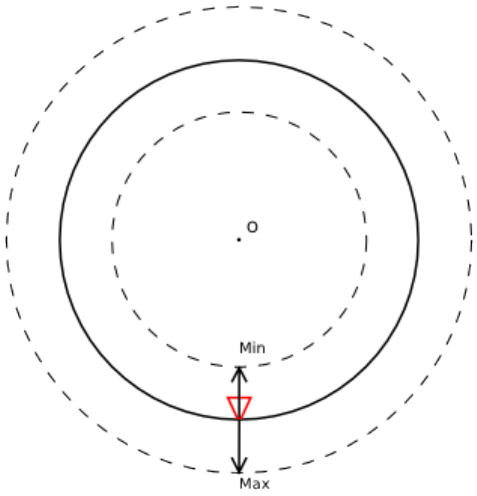
10.3 Limiting the Camera Movement

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

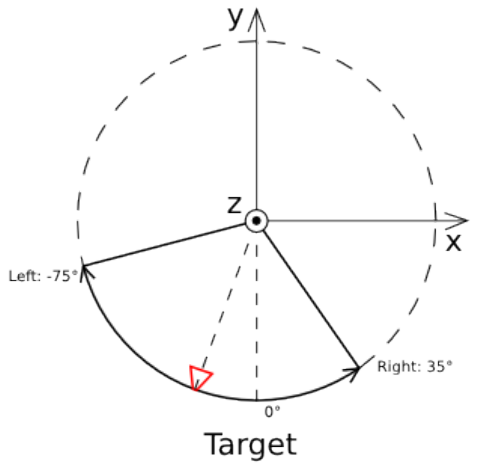

Camera Move Style
Move Style: Target
Target Location:
X: 0.00
Y: 0.00
Z: 0.00
Copy Cursor Location
Camera Velocities:
Translation Velocity: 1.000
Rotation Velocity: 1.000
Zoom Velocity: 0.100
Camera Limits:
Use Distance Limits
Min: 1.000
Max: 100.000
Use Horizontal Rotation Clamping
Left Angle: -180°
Right Angle: 180°
Camera Space
Use Vertical Rotation Clamping
Down Angl: -90°
Up Angle: 90°
Camera Space
Use Panning Mode
Camera Move Style
Move Style: Eye
Camera Velocities:
Translation Velocity: 1.000
Rotation Velocity: 1.000
Zoom Velocity: 0.100
Camera Limits:
Use Horizontal Rotation Clamping
Left Angle: -180°
Right Angle: 180°
Camera Space
Use Vertical Rotation Clamping
Down Angl: -90°
Up Angle: 90°
Camera Space
Camera Move Style
Move Style: Hover
Camera Velocities:
Translation Velocity: 1.000
Rotation Velocity: 1.000
Zoom Velocity: 0.100
Camera Limits:
Use Horizontal Translation Limits
Min: -100.000
Max: 100.000
Use Vertical Translation Limits
Min: -100.000
Max: 100.000
Use Distance Limits
Min: 1.000
Max: 100.000
Hover Angle Limits:
Down Angle: 0°
Up Angle: 90°
Use Horizontal Rotation

Target camera

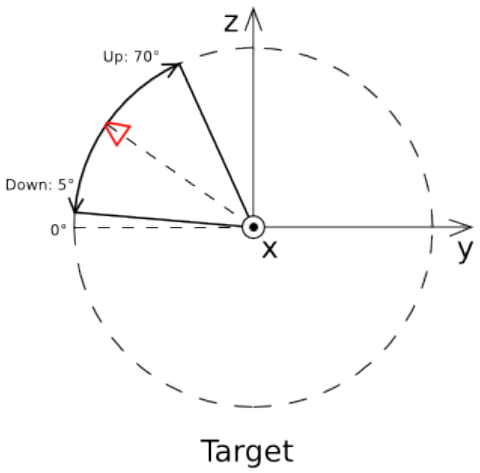
Camera Move Style > Camera Limits > Use Distance Limits Set minimum and maximum distances from the camera to the pivot point. Allowable values: Min ≤ Max. Disabled by default.



Camera Move Style > Camera Limits > Use Horizontal Rotation Clamping Limit the horizontal (around the Z world axis in Blender) camera rotation around the corresponding point. Rotation is performed along the arc of a circle between Left Angle and Right Angle values. The rotation arc corresponds to movement from Left Angle to Right Angle anticlockwise. Disabled by default.



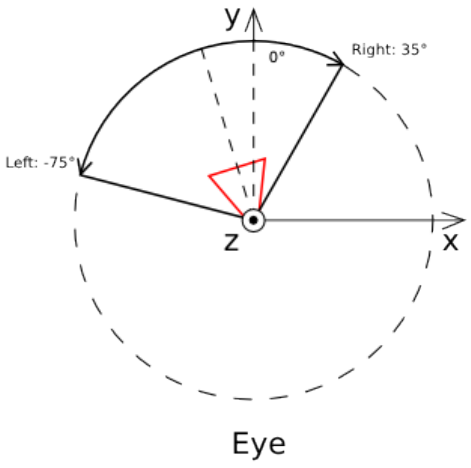
Camera Move Style > Camera Limits > Use Vertical Rotation Clamping Limit the vertical (around the camera's local X axis in Blender) camera rotation around the corresponding point. Rotation is performed along the arc of a circle between Down Angle and Up Angle values. The rotation arc corresponds to movement from Down Angle to Up Angle clockwise. Disabled by default.



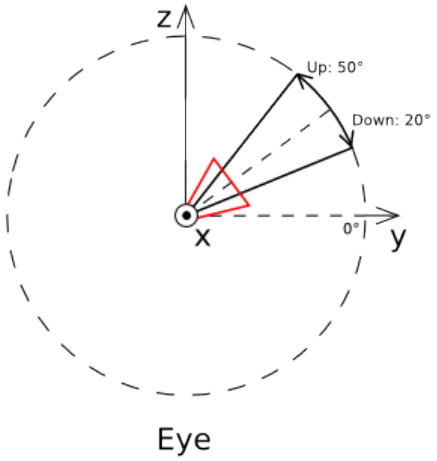
Camera Move Style > Camera Limits > Use panning mode Allow camera panning.

Eye Camera

Blend4Web > Use Horizontal Rotation Clamping Limit the horizontal (around the world Z axis in Blender) camera rotation around its position. Rotation is performed along the arc of a circle between Left Angle and Right Angle values. The rotation arc corresponds to movement from Left Angle to Right Angle clockwise. Disabled by default.



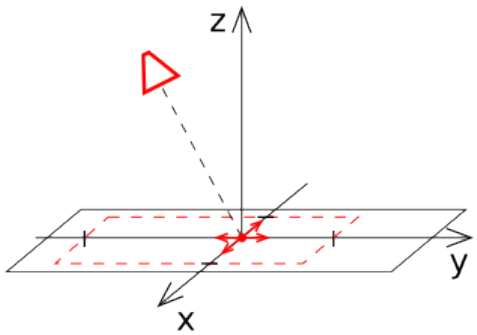
Camera Move Style > Camera Limits > Use Vertical Rotation Clamping Limit the vertical (around the camera's local X axis in Blender) camera rotation around its position. Rotation is performed along the arc of a circle between Down Angle and Up Angle values. The rotation arc corresponds to movement from Down Angle to Up Angle anticlockwise. Disabled by default.



Hover Camera

Camera Move Style > Camera Limits > Use Horizontal Translation Limits Limit movement of the pivot point along the Blender's world X axis. Allowable values: Min ≤ Max. Disabled by default.

Camera Move Style > Camera Limits > Use Vertical Translation Limits Limit movement of the pivot point along the Blender's world Y axis. Allowable values: Min ≤ Max. Disabled by default.

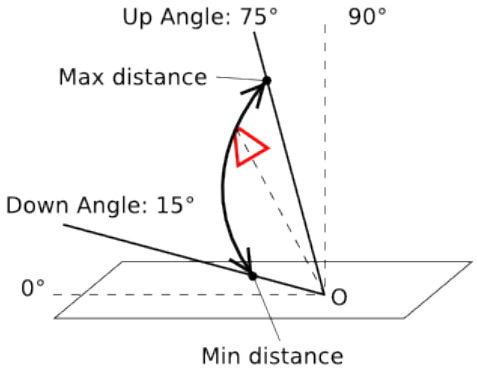


Camera Move Style > Camera Limits > Use Distance Limits Set minimum and maximum distances from the camera to the point of intersection between the camera view direction and the horizontal plane (Blender's world XOY plane by default). Allowable values: Min ≤ Max. If turned off, the camera will freely move along Blender's world Z axis, the pivot point will not be defined, and the Camera Move Style > Camera Limits > Use Horizontal Translation Limits and Camera Move Style > Camera Limits > Use Vertical Translation Limits settings will limit the

position of the camera itself. Disabled by default.

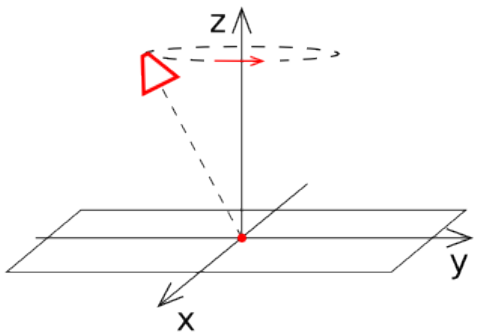
Camera Move Style > Camera Limits > Camera Angle Limits Limit the camera's inclination (the angle between camera view direction and the horizontal plane). Becomes available when the Camera Move Style > Camera Limits > Use Distance Limits option is enabled. Allowable values: $0 \leq$ Down Angle $\leq$ Up Angle ≤ 90. Default values: Down Angle $= 0$, Up Angle $= 90$.

If the Blend4Web > Use Distance Limits setting is turned on, the limits for distance and inclination angle will be applied simultaneously, to define the camera movement path in the vertical plane.



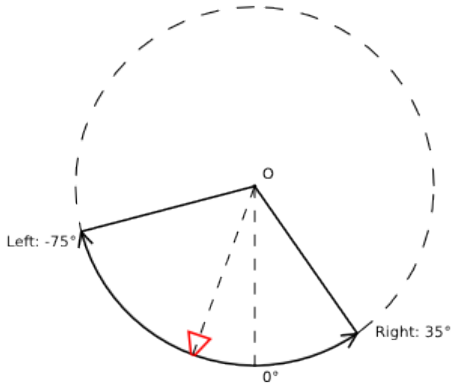
Setting incorrect limits for distance or inclination angle will disable the Camera Move Style > Camera Limits > Use Distance Limits option.

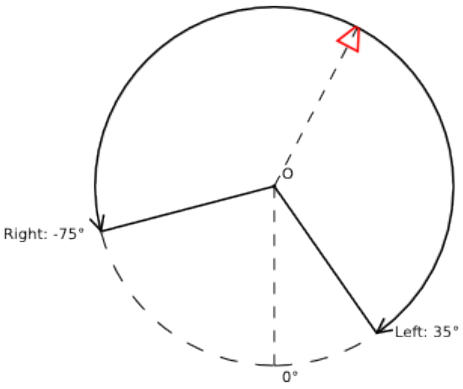
Camera Move Style > Camera Limits > Use Horizontal Rotation Allow camera rotation in Blender's XOY plane around to the pivot point. Becomes available upon turning Camera Move Style > Camera Limits > Use Distance Limits setting on. Enabled by default.



Peculiarities of Limiting Settings

- For EYE/TARGET cameras, interchanging Left/Right or Down/Up values results in movement along the opposite arc of a circle.





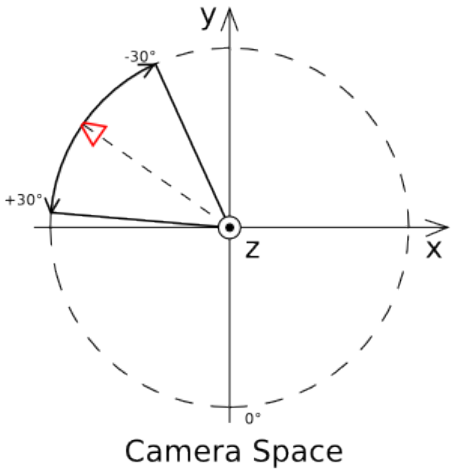
- When limiting the camera's horizontal and vertical rotation, it is possible to choose between the following coordinate spaces:

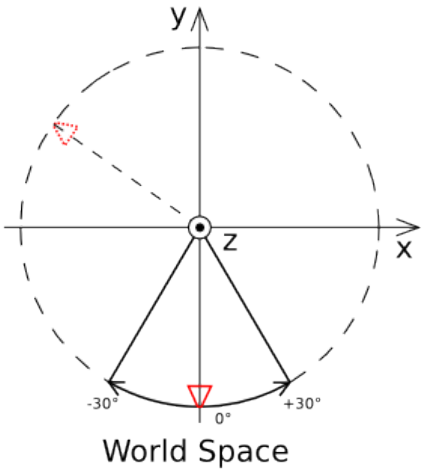
Camera Space All angles are measured relative to the initial camera position and orientation.

World Space Horizontal angles are measured relative to the Y axis in world space; vertical angles are measured relative to the Blender's XOY plane in world space.

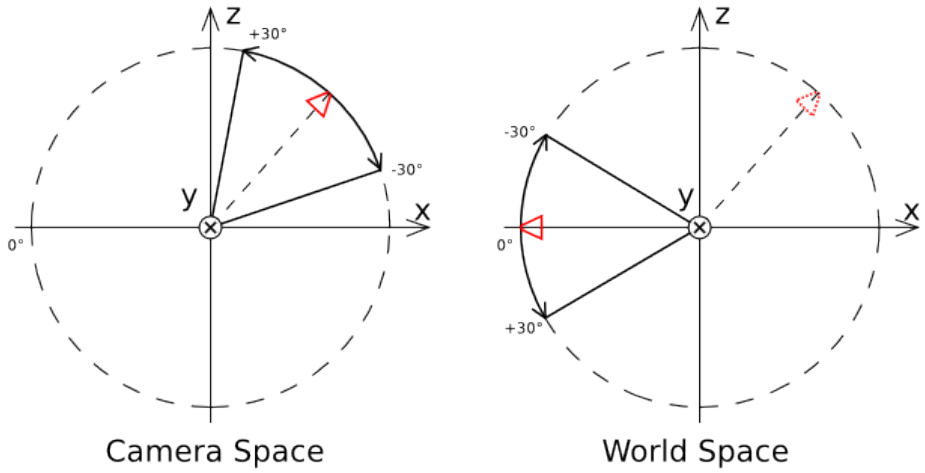
Default value: Camera Space.

Horizontal limits by the example of the TARGET camera:





Vertical limits by the example of the TARGET camera:

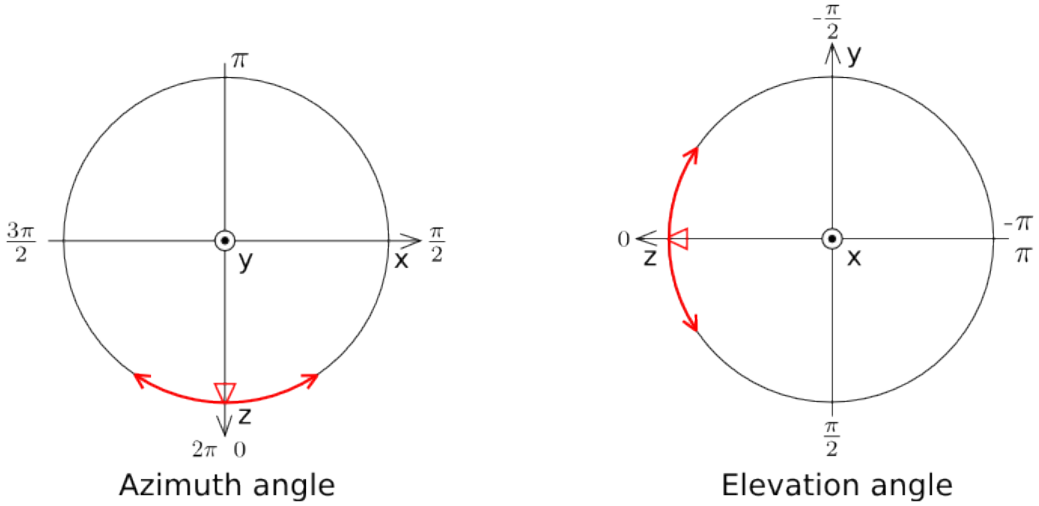


10.4 Camera Controls API

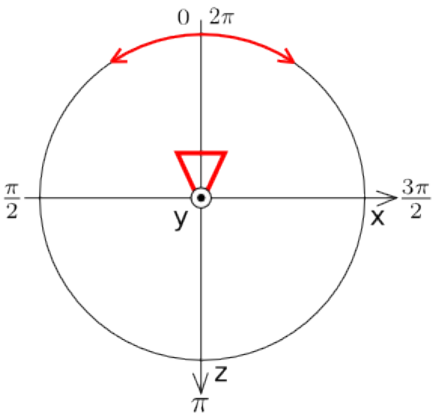
Basic camera controls method are contained in the camera.js module.

When working with camera API (rotation, setting limits), all angles are defined in the engine's coordinate system as follows:

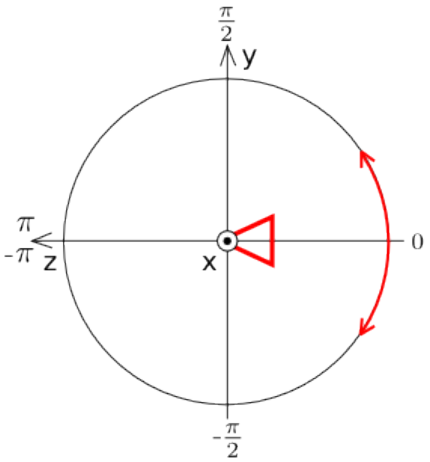
For the TARGET/HOVER camera:



For the EYE camera:



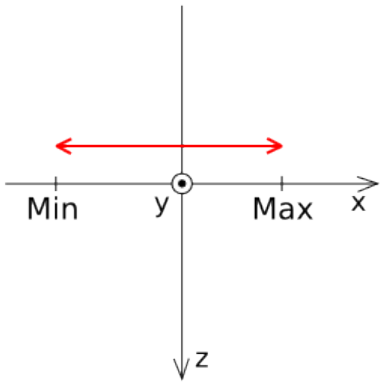
Azimuth angle



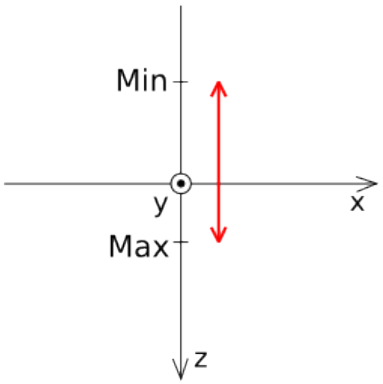
Elevation angle

When setting translation limits for the HOVER camera, the limits' values will correspond to the engine's coordinate axis as follows:

- Horizontal Translation Limits - X axis
- Vertical Translation Limits - Z axis



Horizontal limits



Vertical limits

Changing Movement Style of the Camera

Movement style of the camera (Blend4Web > Move Style option in Blender) can be changed by using the set_camera_move_style() method of the app.js module:

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

If an application does not use the app.js module, this can be achieved by using the set_move_style() method of the camera.js module:

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

Please note that changing movement style of the camera also resets the camera movement limits and the pivot position (for TARGET and HOVER cameras). These parameters should be set again by using the corresponding methods of the camera.js module.

CHAPTER 11

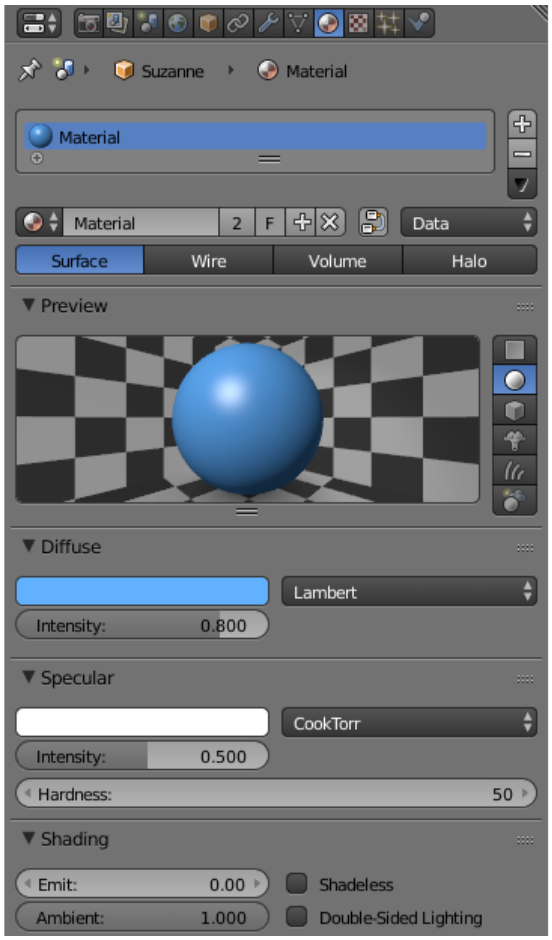
Materials

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.

11.1 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 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 default value is CookTorr.

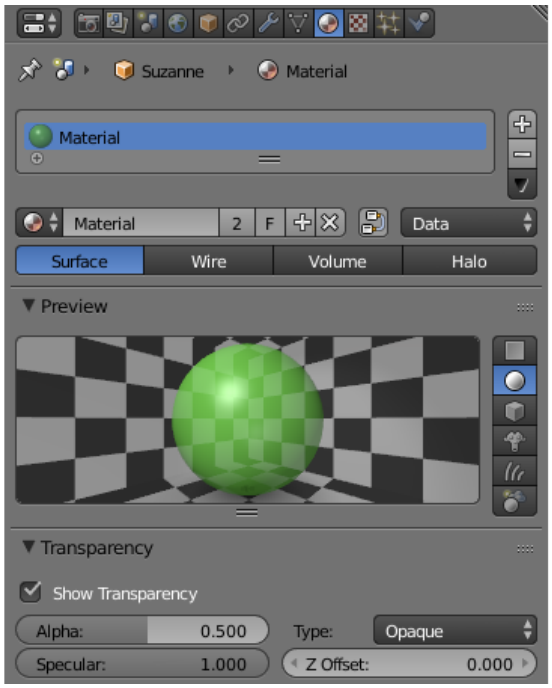
Shading > Emit Emission intensity. The default value is 0.0.

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.

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

11.2 Transparency



11.2.1 Types

Transparency implementation type can be selected in the Transparency menu on the Properties > Material panel.

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).

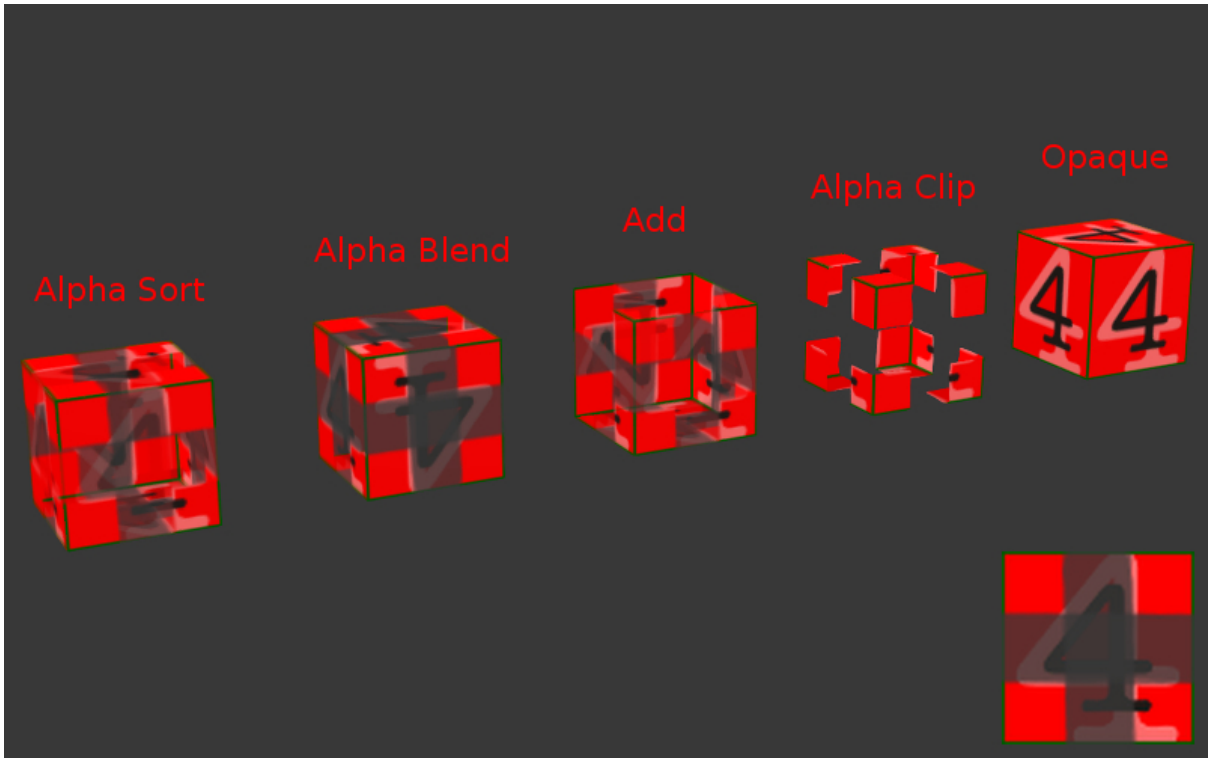
Alpha Blend Transparent with a gradient. The sorting of triangles is not performed. It is recommended to use this feature for unclosed transparent geometry (water surface, decals).

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.



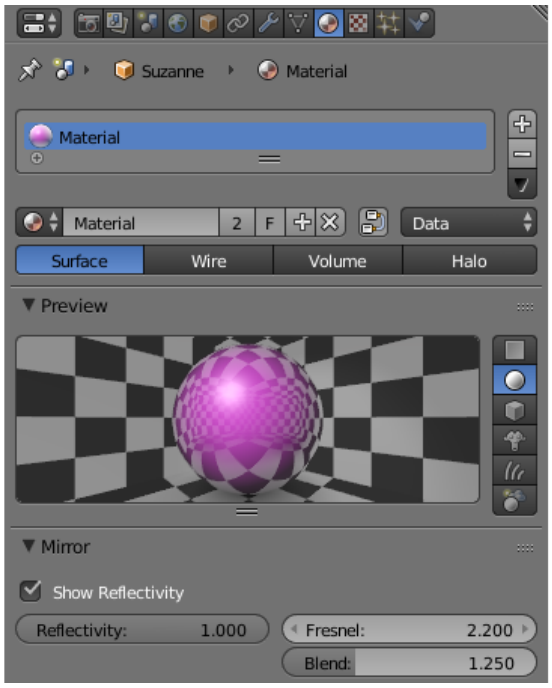
11.2.2 Additional settings

Transparency > Show 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.

Transparency > 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.

11.3 Reflection



11.3.1 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

11.3.2 Dynamic reflection

A surface reflects the selected objects in their current position. The engine supports planar and spherical reflections

Activation

1. Check Reflections setting on the Render > Reflections and Refractions panel.
2. For reflective objects enable the Reflective option on the Object > Reflections panel.
    - For planar reflections, set the Object > Reflections > Type property to Plane. After that, add an empty object to be used as a reflection plane by executing for example Add > Empty > Single Arrow. Rename it for convenience and specify its name in the Reflection plane field of the reflective object.

- For cube-mapped reflections, set the Object > Reflections > Type property to Cube.

3. For the needed materials of the reflective objects, set the Material > Mirror > Reflectivity value.
   - Mirror > Reflectivity > Show Reflectivity is required for displaying reflections on objects in Blender. The engine ignores this option.
4. For the reflexible objects, enable the Reflexible checkbox on the Object > Reflections 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

11.3.3 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.

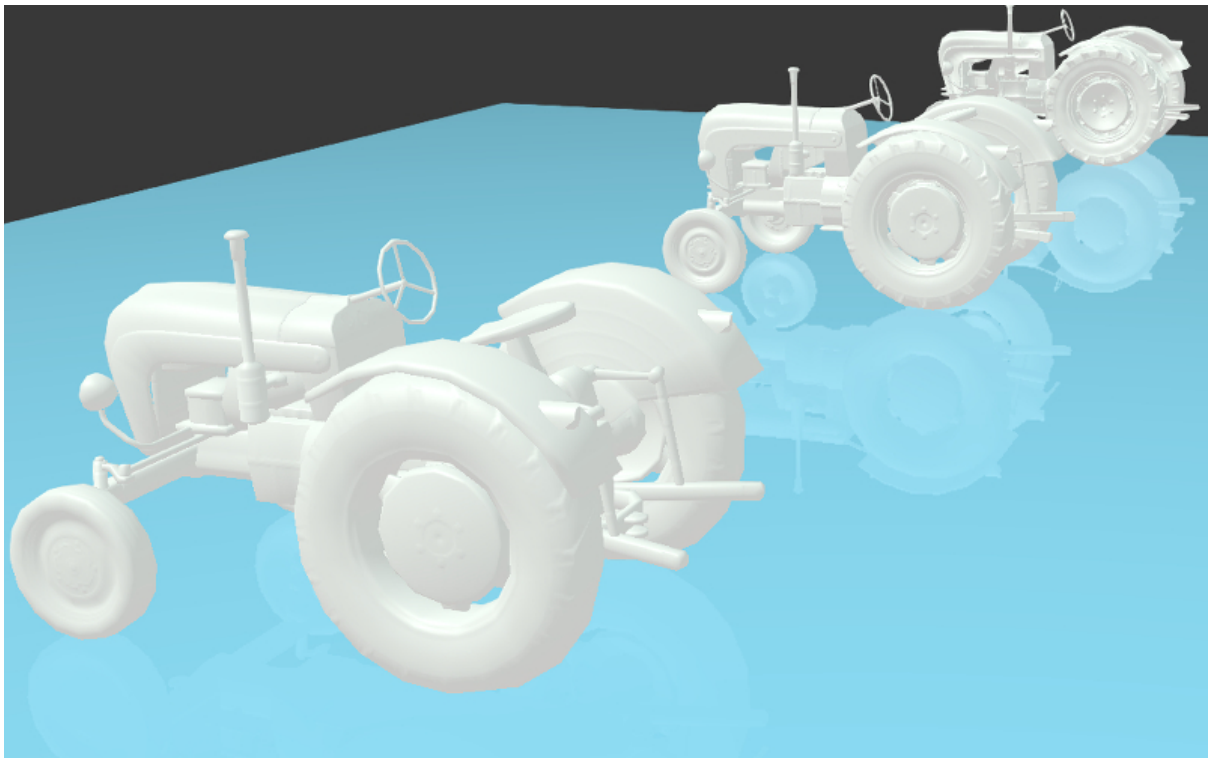
Settings

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

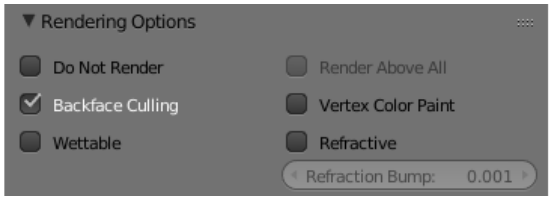
Material > Mirror > Fresnel Fresnel power for reflection. This is the N exponent in the Schlick's formula. In Blender it is limited to values from 0 to 5. If this parameter is equal to zero the Fresnel effect is not observed and the full reflection at all

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.

Material > 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.



11.4 Rendering Properties



Material > Rendering Options > Do not Render Disable rendering of this object.

Material > Rendering Options > Backface Culling When enabled, poligons' back faces are not rendered by the engine. Enabled by default.

Material > Rendering Options > Wettable Water wetting effect is activated for the material.

See also:

Water

Material > Rendering Options > Render above all Material is rendered on top of all scene objects. Transparency type with a gradient is requred (Add, Alpha Blend or Alpha Sort).

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

Material > Rendering Options > Refractive Make object refractive. Perturbation factor can be set with the option Refraction Bump on the Refraction Settings panel. The default value is 0.001.

Note: In order to use this effect, select ON or AUTO on the Render > Reflections and Refractions > Refractions panel. The object must have Alpha Blend transparency type.

See also:

Transparency

11.5 Engine Specific Parameters

Material > Water Special material for water rendering.

Material > Terrain dynamic grass Material is used for grass rendering.

Material > Collision A special material for collision geometry.

See also:

Physics

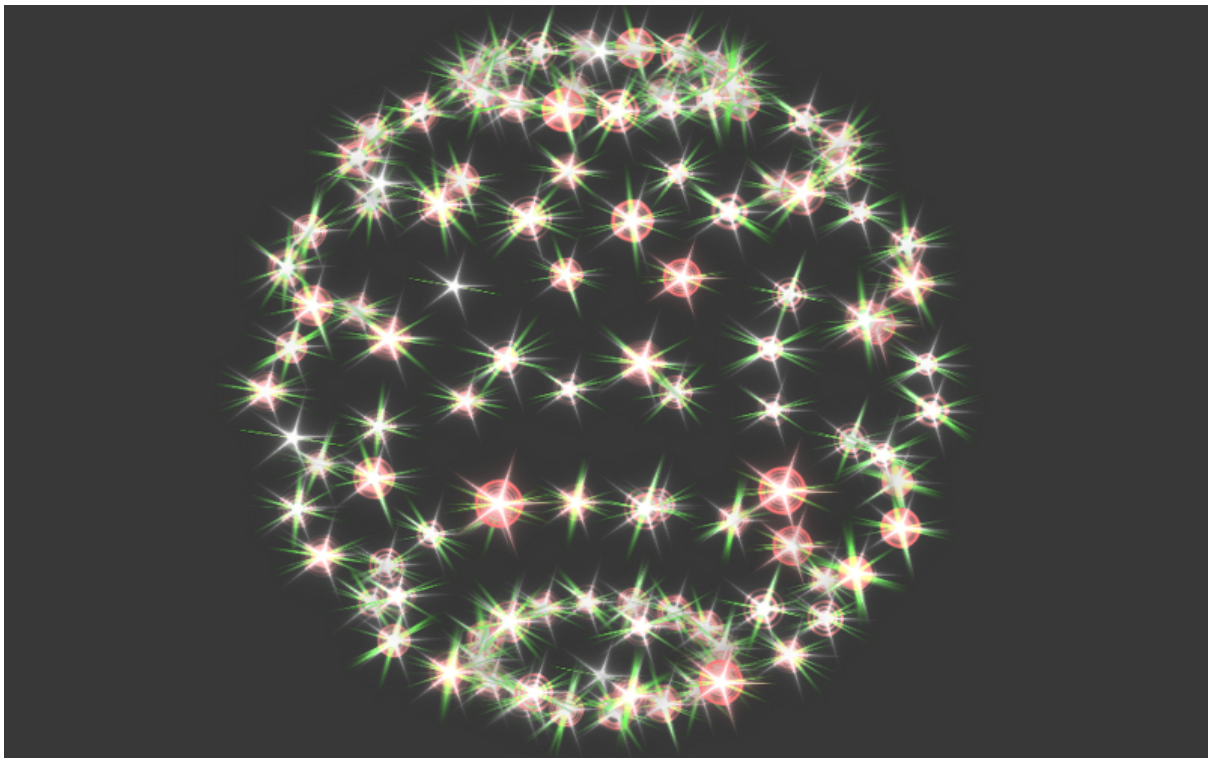
Material > Export Options > Do Not Export Material is not to be exported.

11.6 Halo Materials

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

11.6.1 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).



11.6.2 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 > 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 > 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.

Halo > Special: Stars Enables the starry sky rendering mode. The mesh is fixed relative to the camera. For the Sun lamp it is also required to enable the Lamp > Dynamic Intensity checkbox. Applications should set up the hours of darkness via API.

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

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

CHAPTER 12

Textures

Textures are hand-made or procedurally generated images that can be applied to the model surfaces to add more detail. As a rule, the image pixels are assigned to the 3D surface points using texture mapping. For this reason they sometimes are referred to as maps.

Usually the textures are placed into material texture slots. They can be also used for particle systems parametrization and for creating the skydome.

12.1 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 visualization of relief surfaces (parallax mapping).
    - stencil map
    - video texture
2. Environment Map
    - mirror map
    - skydome texture
    - used for implementation of an environment lighting method
3. None

- applied to the Blender's default scene cube. It is also used for rendering a scene to texture and for rendering canvas textures.

4. Blend, gradient
   - is used in particle systems
5. Voronoi procedural texture type
   - is used for water rendering to setup the caustics effect

12.2 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) and Generated. The default value is Generated.

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

Export Options > Do Not Export Do not export the texture.

Export Options > Disable Compression Disable texture compression for this texture. Used in cases when texture compression deteriorates the image quality.For example it's recommended to disable compression for mask textures used to mix different parts of materials.

Export Options > Shore Distance Map is used in outdoor rendering

Export Options > 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).

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

Note: Texture compression is disabled for textures used as normal maps.

12.3 Diffuse Map

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

12.3.1 Activation

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

12.3.2 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.

12.4 Specular Map

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

12.4.1 Activation

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

12.4.2 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.

12.5 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.

12.5.1 Activation

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

12.5.2 Additional Settings

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

12.6 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.

12.6.1 Activation

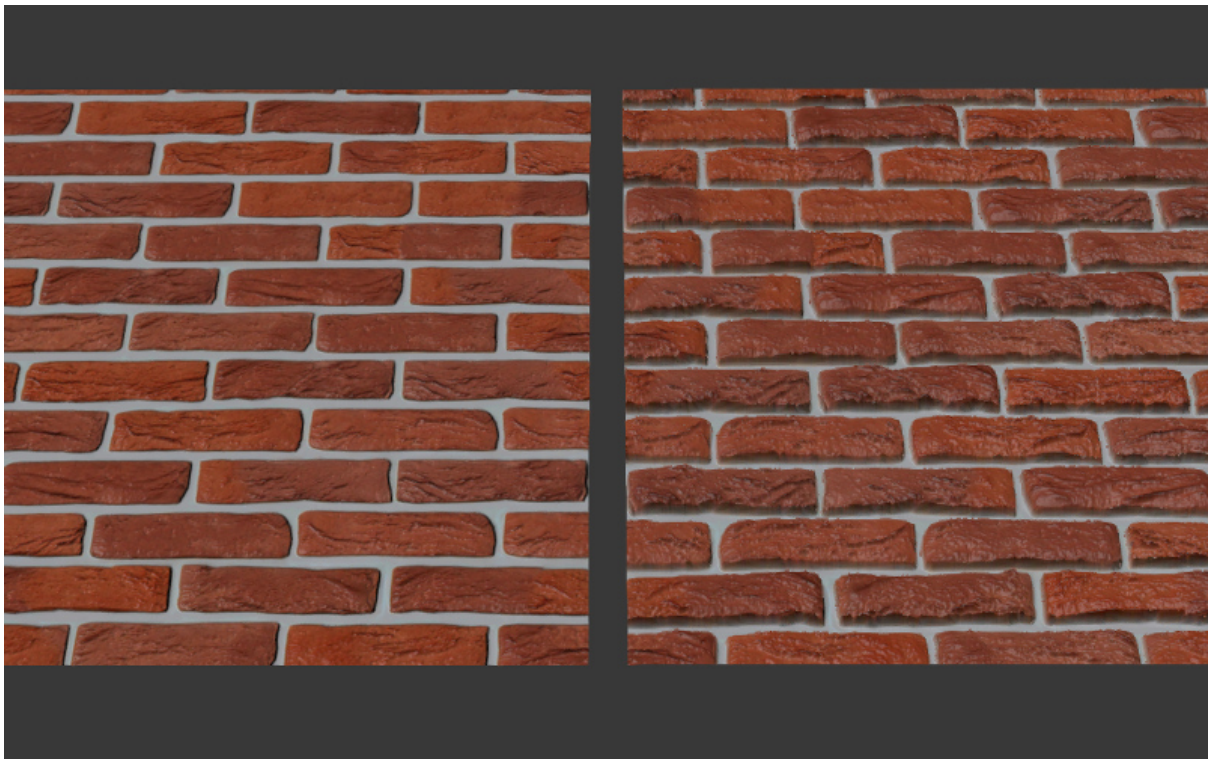
For the normal map enable the Parallax panel in addition to the Geometry > Normal checkbox.

12.6.2 Additional Settings

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

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

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



12.7 Stencil Map

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

12.7.1 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.

12.7.2 Additional Settings

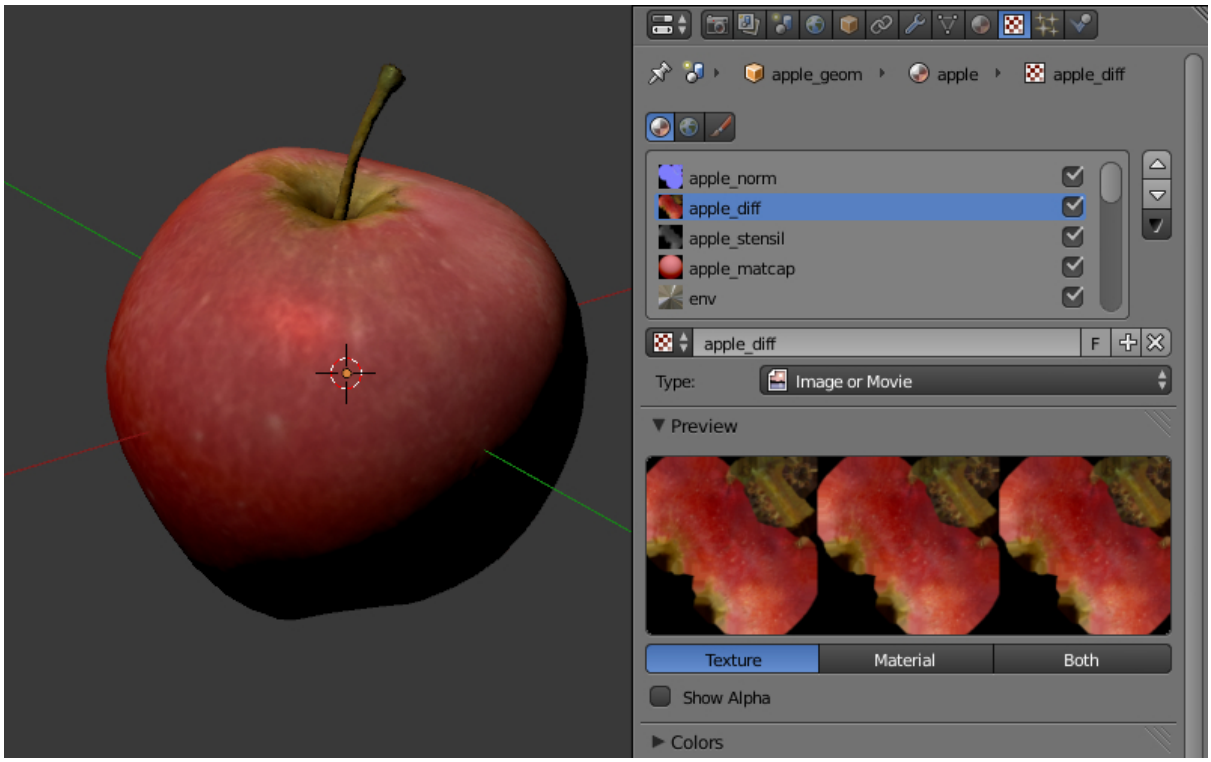
In the case of generic materials one of the mixed diffuse textures can have the Normal ("matcap") texture coordinates type.

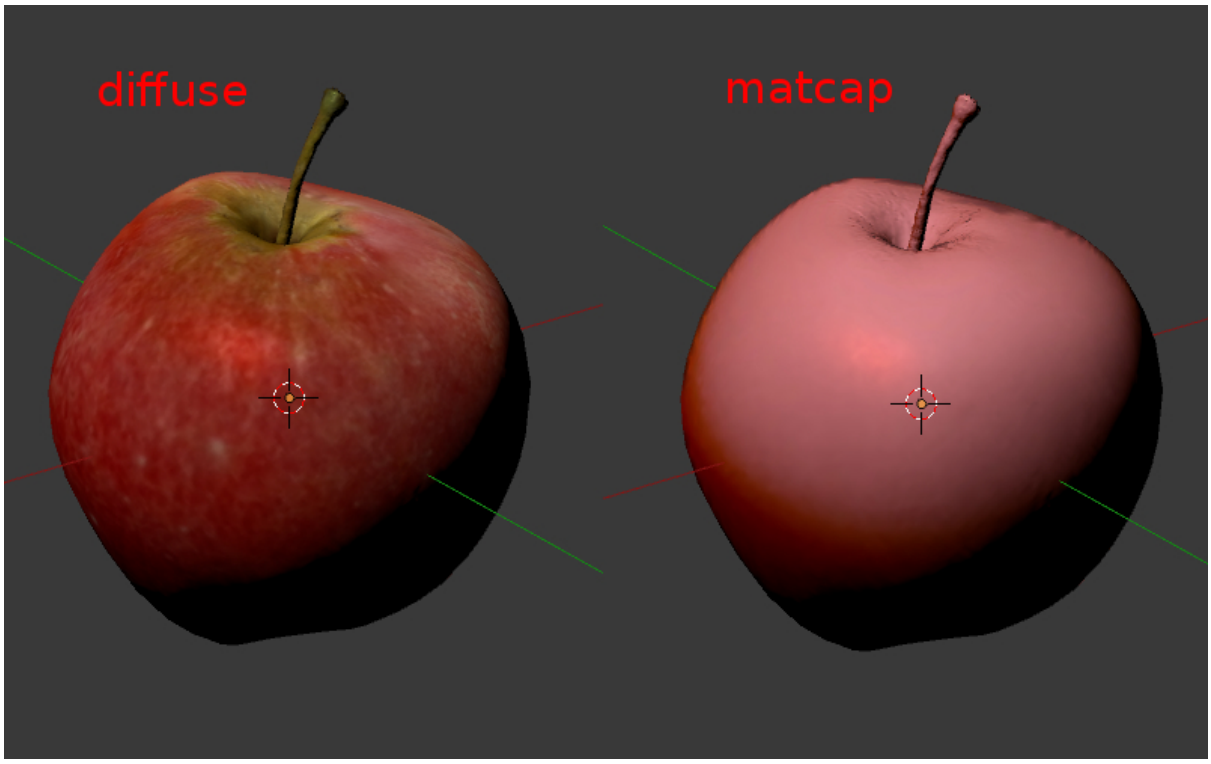
12.7.3 Limitations

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.

12.7.4 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.





12.8 Video Textures

A video can be used as a texture if Image or Movie texture type is selected. Also, if the Scene > NLA is enabled the video textures playback is possible.

Note: Video textures support playing back just video tracks. Audio tracks should be played back by using a SPEAKER object.

12.8.1 Supported formats (containers):

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

We recommend to use WebM as a basic format. It is an open standard supported by the majority of browsers and offers good picture quality.

Note: Files saved in mp4 and ogg formats have different extensions for audio and video data: .mp4 and .ogg extensions are used for sounds, .m4v and .ogv - for video.

Converting resources between different formats is described in the corresponding section.

12.8.2 Setting up the Texture

The following settings are available for video textures on the Texture > Image panel:

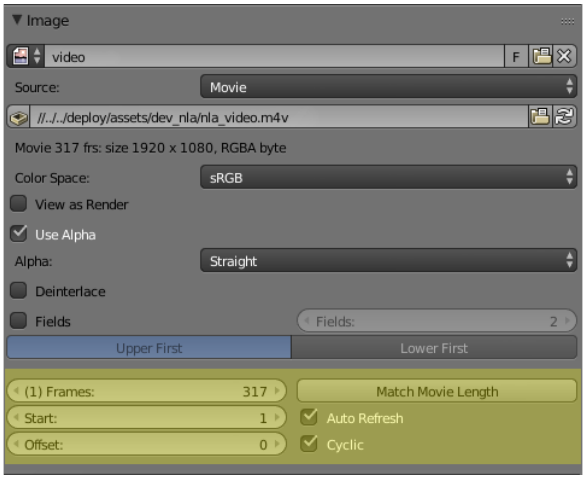
Image > Frames Length of the played fragment in frames.

Image > Start Video playback delay (in frames), applicable for non-linear animation (Scene > NLA option).

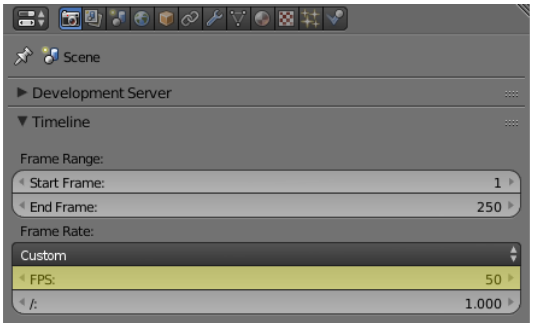
Image > Offset The number of the frame from which the video playback starts.

Image > Auto Refresh Play back the video immediately after loading is complete. Ignored for non-linear animation - other parameters are applicable for video playback.

Image > Cyclic Start video playback afresh each time it finishes.



The video playback rate can be increased. To do this set Scene > Dimensions > Frame rate to a value which is different from the FPS value for the videos. Video playback rate is increased proportionally to the ratio of the scene's FPS and the video's FPS.



12.8.3 Specifics of Mobile Devices

The peculiarities for mobile devices are as follows:

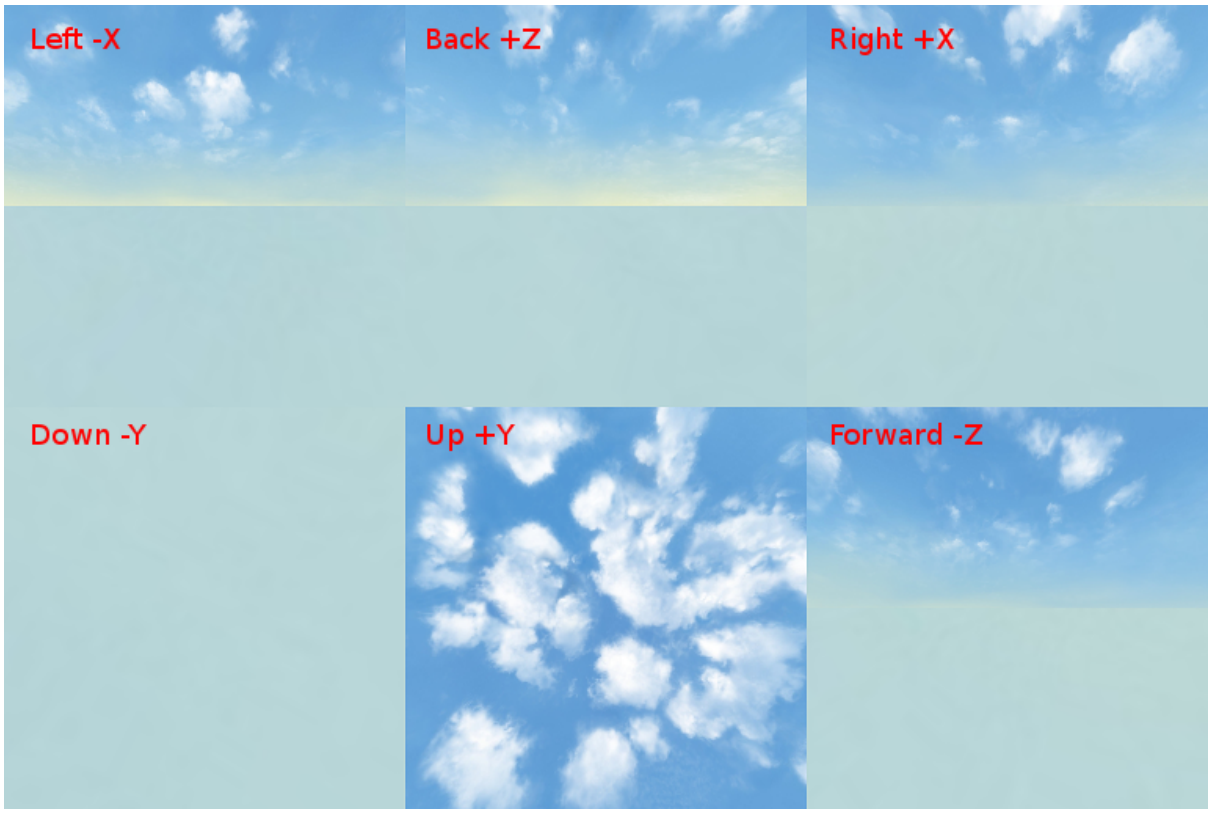
1. Normal operation of video textures on iPhones is not possible because these devices play back videos via the standard iOS video player. For these devices you need to convert your videos to special .seq format by using our converter.
2. some devices only support playing back only one video file.
3. stable operation is not garanteed if the Offset value is not zero.
4. not all devices support changing the video playback rate.
5. iPad and iPhone do not provide the possibility to control the audio volume for video, and so the audio track should be removed from the video before the file is added to Blender.

12.9 Environment Map

An environment map can be used as a mirror map, as a static sky texture (skydome) and also for implementation of an environment lighting method.

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).

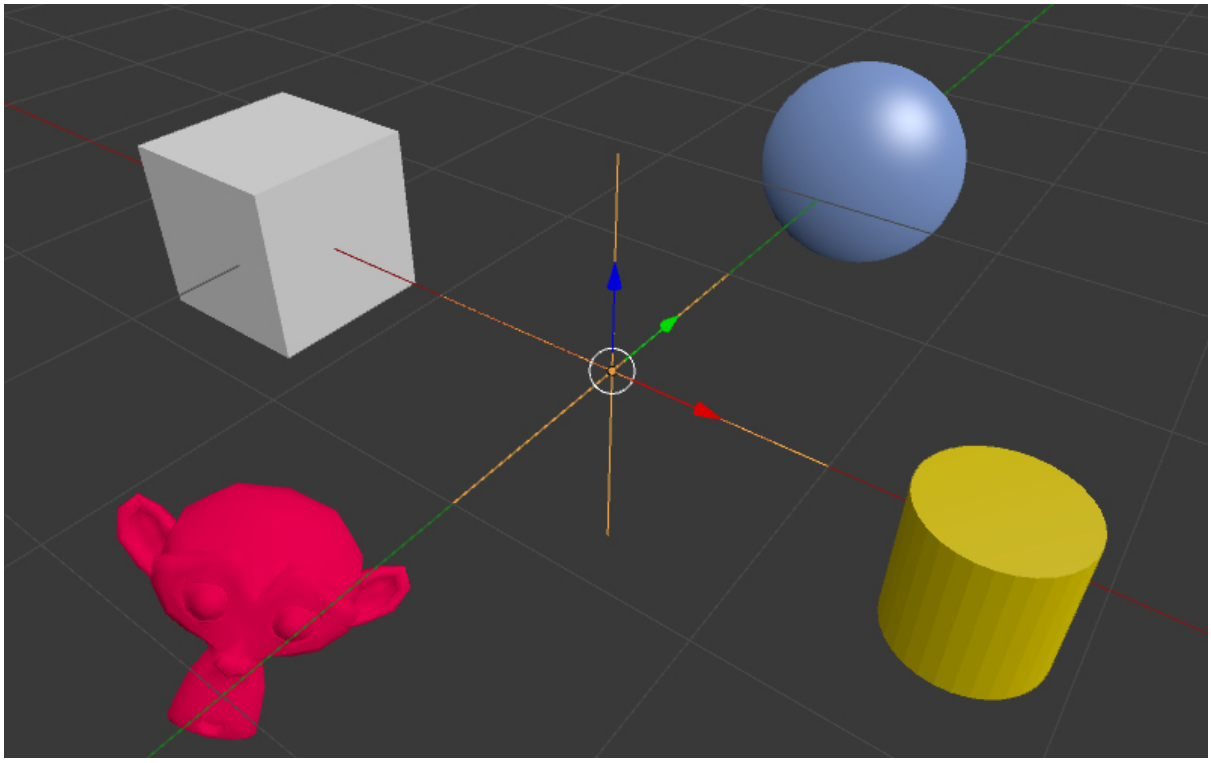
It is recommended to use the lossless format (PNG) in order to avoid seams.

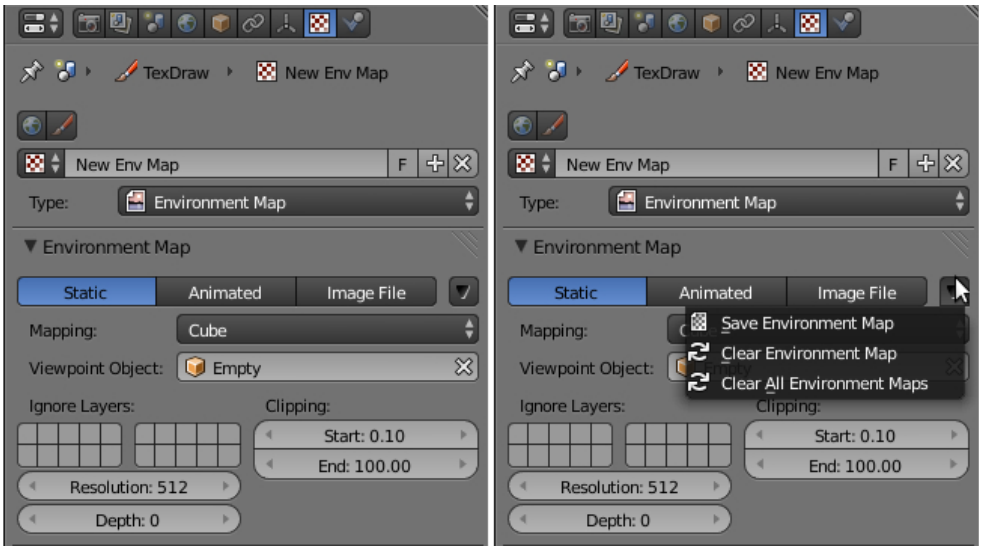


12.9.1 Making Environment Maps

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.





12.10 Mirror Map

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

12.10.1 Activation

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

12.10.2 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.

12.11 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 implement one of the environment lighting methods.

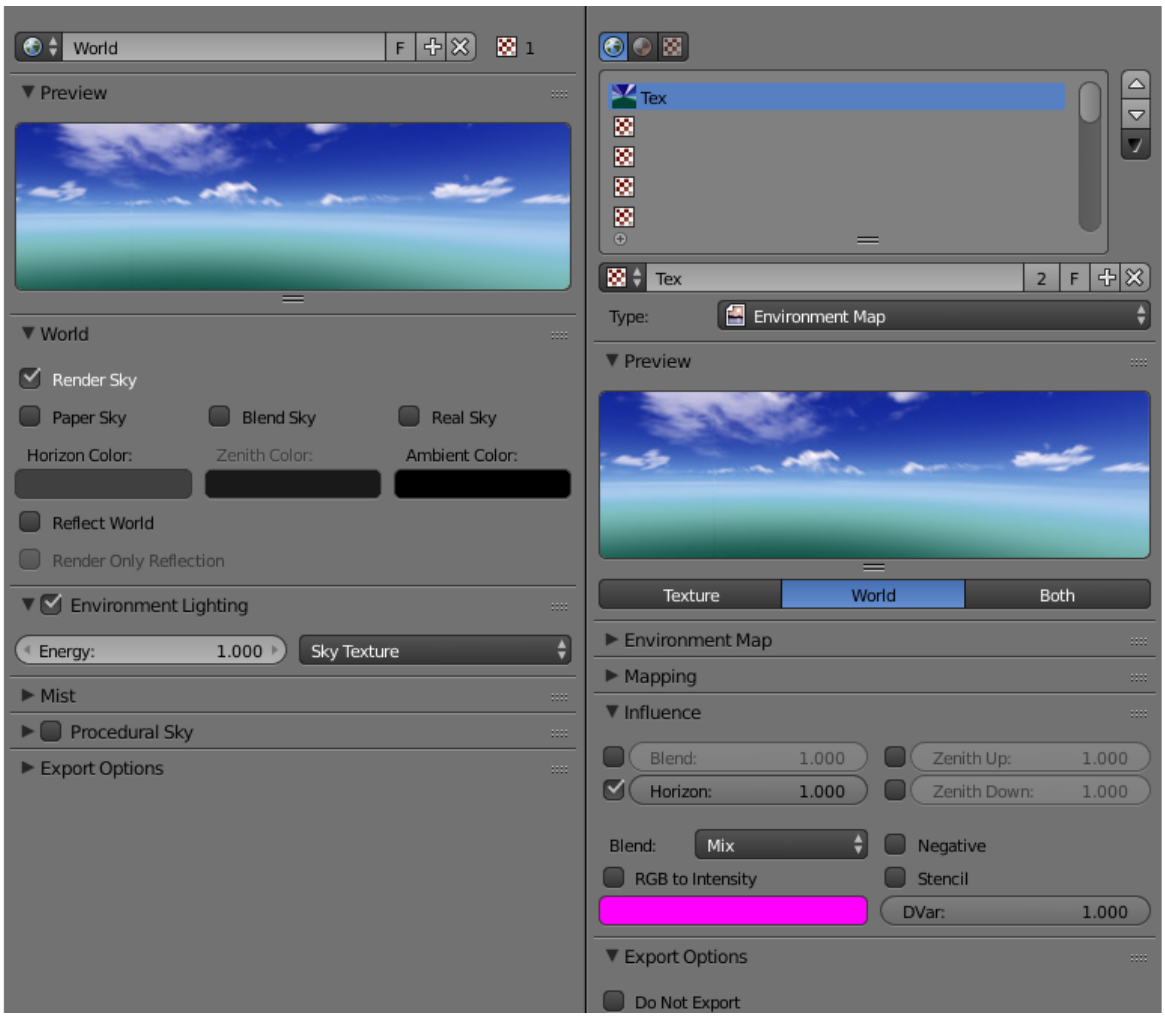
12.11.1 Activation

Create a world texture of Environment Map type. Select the Export Options > Sky Texture Usage > SKYDOME option. Enable World > Render Sky under the World tab.

Note: The behavior of the texture is intentionally made as close as possible to the Blender internal render. That's why the texture may not be displayed upon its default settings. In order to make the texture visible, enable the Influence > Horizon checkbox on its panel and set the Horizon value to 1.0.

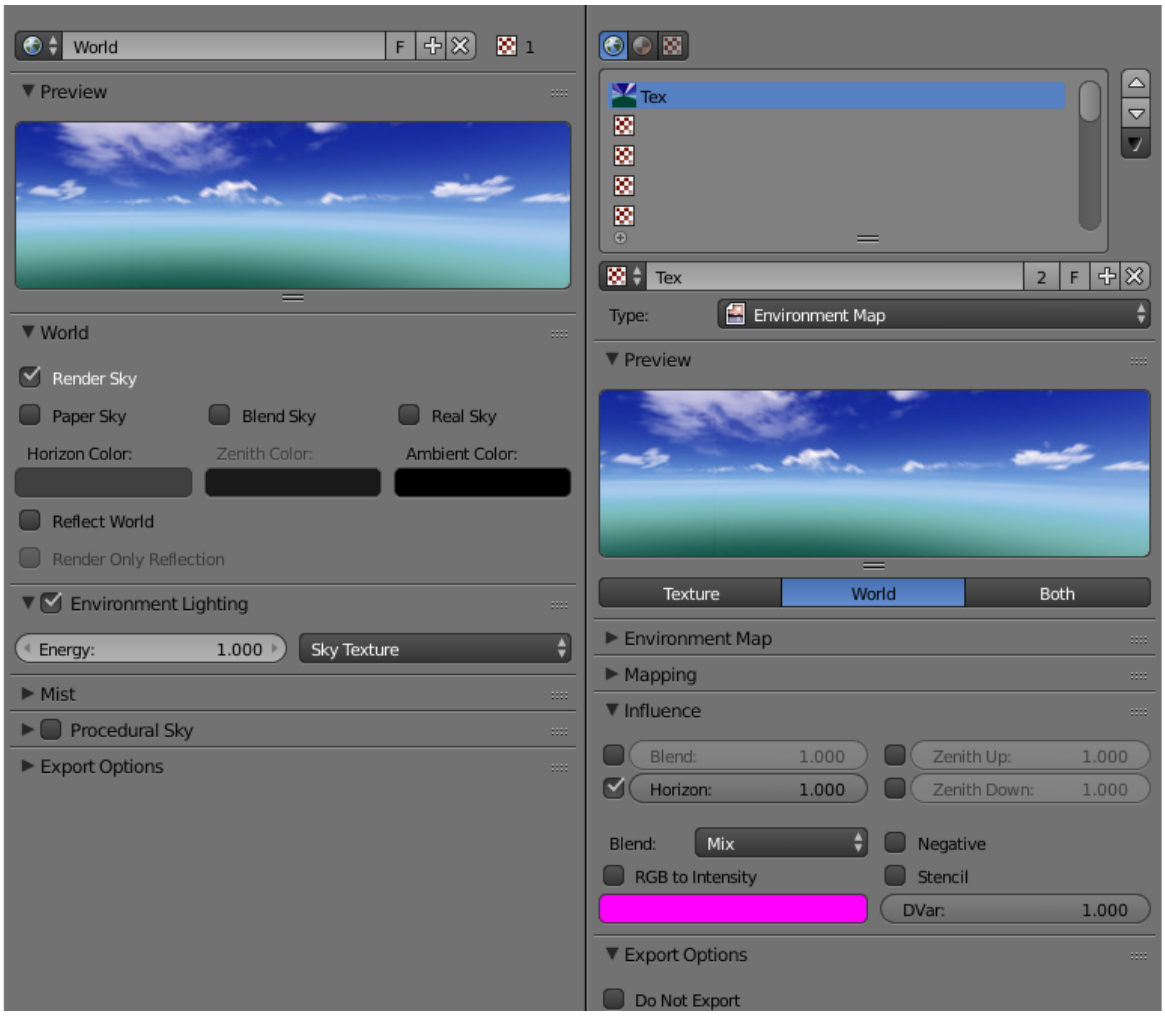
Note: To imitate environment lighting you can select the Export Options > Sky Texture Usage > ENVIRONMENT_LIGHTING option. Also, you should select the corresponding option in the world settings: Environment Lighting > Sky Texture.

To use the world texture both for skydome and for environment lighting, select Export Options > Sky Texture Usage > BOTH.



12.11.2 Additional Settings

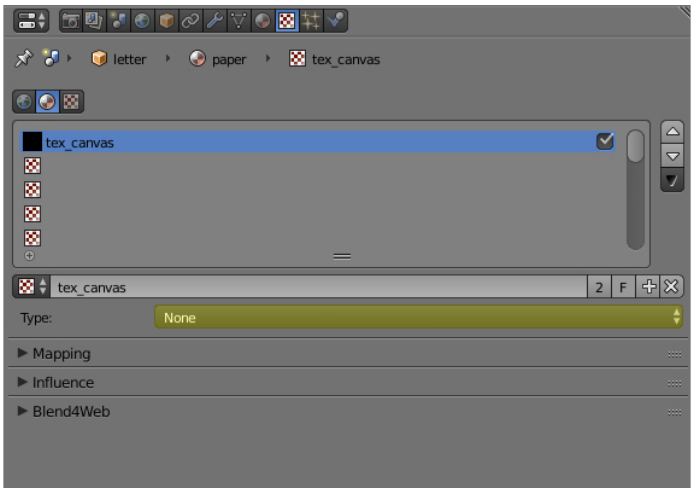
The engine also supports parameters from the world texture's Influence panel which are used for sky rendering. Mixing of the world texture with color depends on the World > Horizon Color and World > Zenith Color parameters, as well as on the Paper Sky, Blend Sky and Real Sky options. All mixing options are supported (Mix, Add, Multiply etc).



Note: The Influence panel parameters only affect the sky rendering. They do not affect environment lighting by any means.

12.12 Special Texture Types

In order to use such textures, select None type under the Textures tab.



On the Textures > Export Options panel, you can set up properties for these textures:

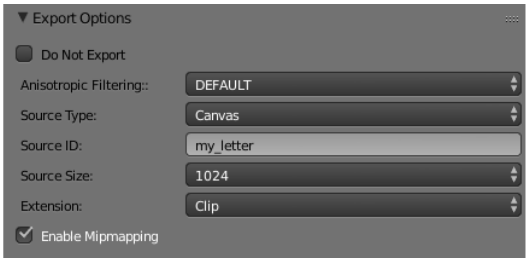
Export Options > Source Type Select texture type: Scene - for rendering a 3D scene into the texture, Canvas - for using <canvas> HTML element and None - for indicating of its absence.

Export Options > Source ID The name of the scene which will be rendered into the texture (for Scene), or ID of the <canvas> HTML element (for Canvas).

Export Options > Source Size Texture resolution.

Export Options > Extension Texture coordinates interpretation mode. Default is Repeat.

Export Options > Enable Mipmapping Enable mipmapping for the Canvas texture. Enabled by default.

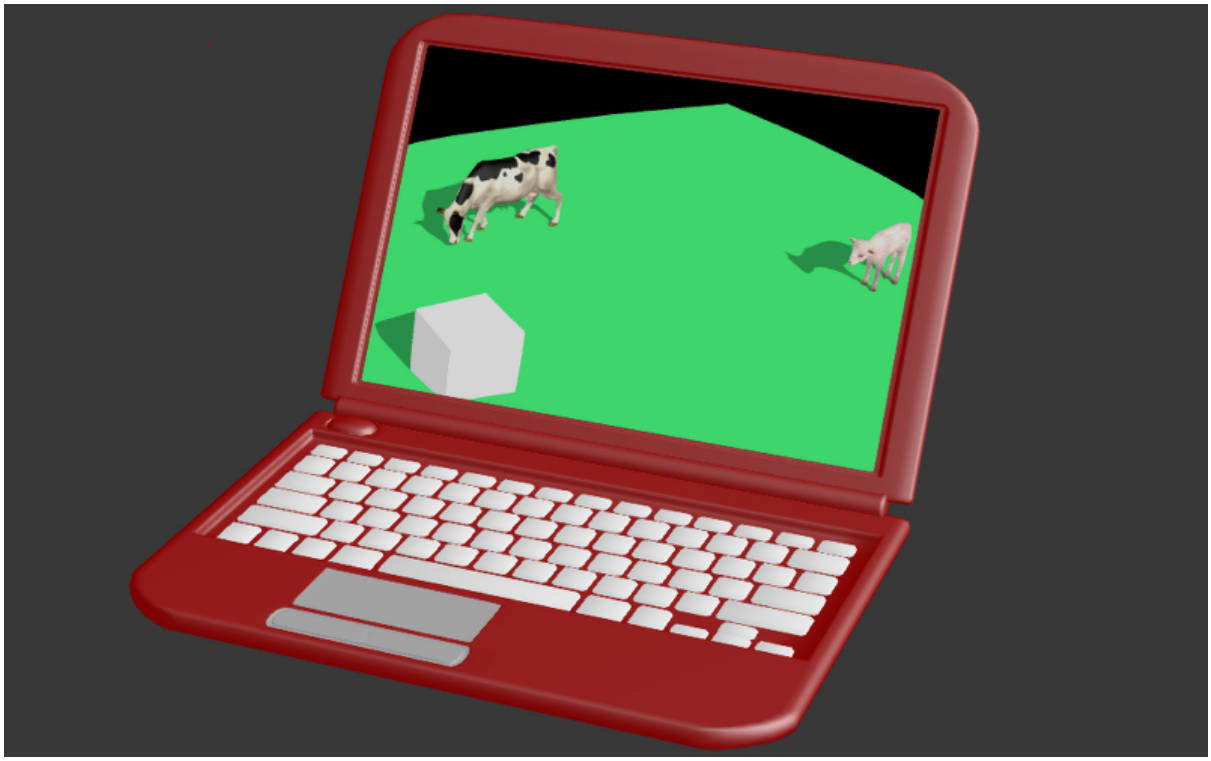


12.12.1 3D scene

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

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 select the Scene type in the Export Options > Source Type menu. Specify the name of the source scene in the Export Options > Source ID field. Set the texture size in the Export Options > Source Size field (in pixels).



The engine also supports the cyclic rendering of scenes to each other.

Note: A project should contain at least one scene which is not rendered by any other scenes.

12.12.2 Canvas

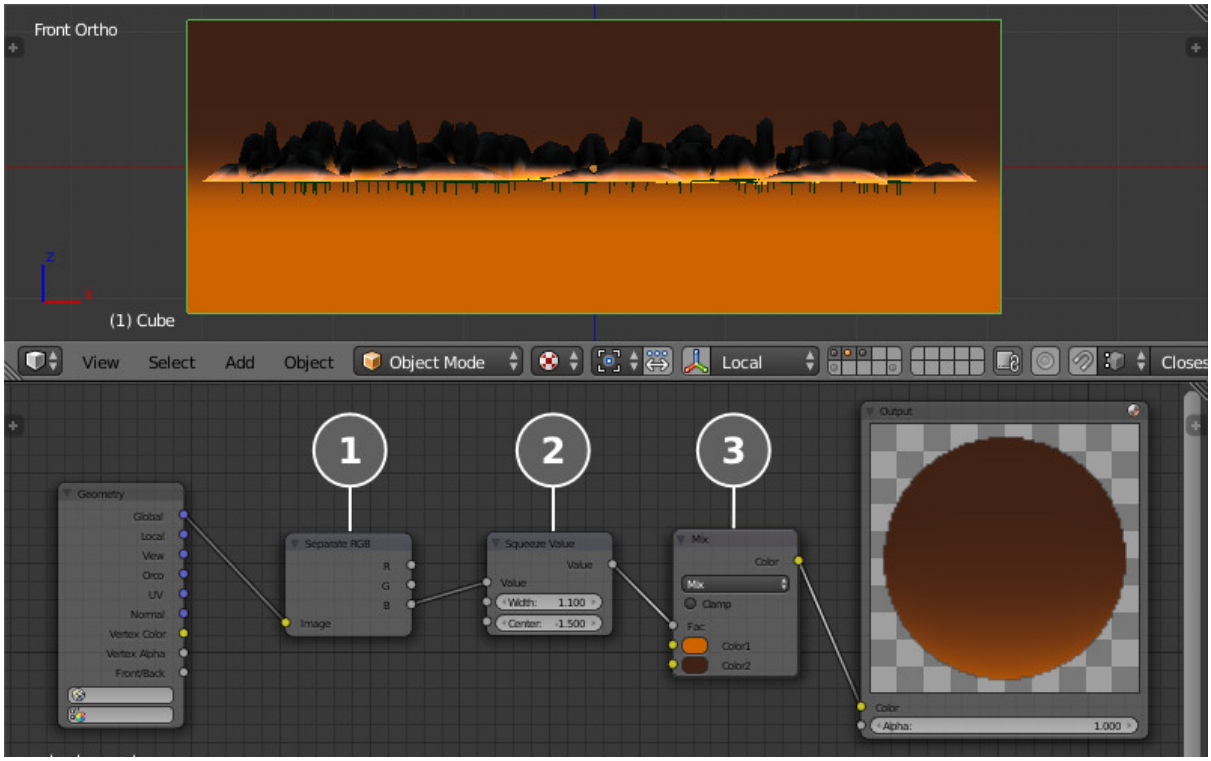
A <canvas> HTML element can be used as a texture. It can be modified via API.

Set the None type for the texture of the target object on the main scene, and select the Canvas type in the Export Options > Source Type menu. Specify the id of the canvas HTML element in the Export Options > Source ID field. Set the texture size in the Export Options > Source Size field (in pixels).

CHAPTER 13

Node Materials

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



13.1 Standard Nodes

All Blender functions are supported except the following cases:

- Geometry - The Vertex Alpha output is not supported
- Lamp data - The Shadow output is not supported.
- Material, Extended Material - no more than one node per material is allowed; the Refl, Ambient, SpecTra inputs are not supported; the AO output is not supported.
- RGB Curves, Vector Curves, ColorRamp and Cycles-nodes have limited support.

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.

13.2 Engine Specific Nodes

Engine-specific nodes extend functionality of the standard nodes to support extra features. These nodes are created as node groups (Node groups or Node tree) with specially determined names and input formats. For convenience, all special nodes are added to a blend file when it is opened.

13.2.1 B4W_LINEAR_TO_SRGB and B4W_SRGB_TO_LINEAR

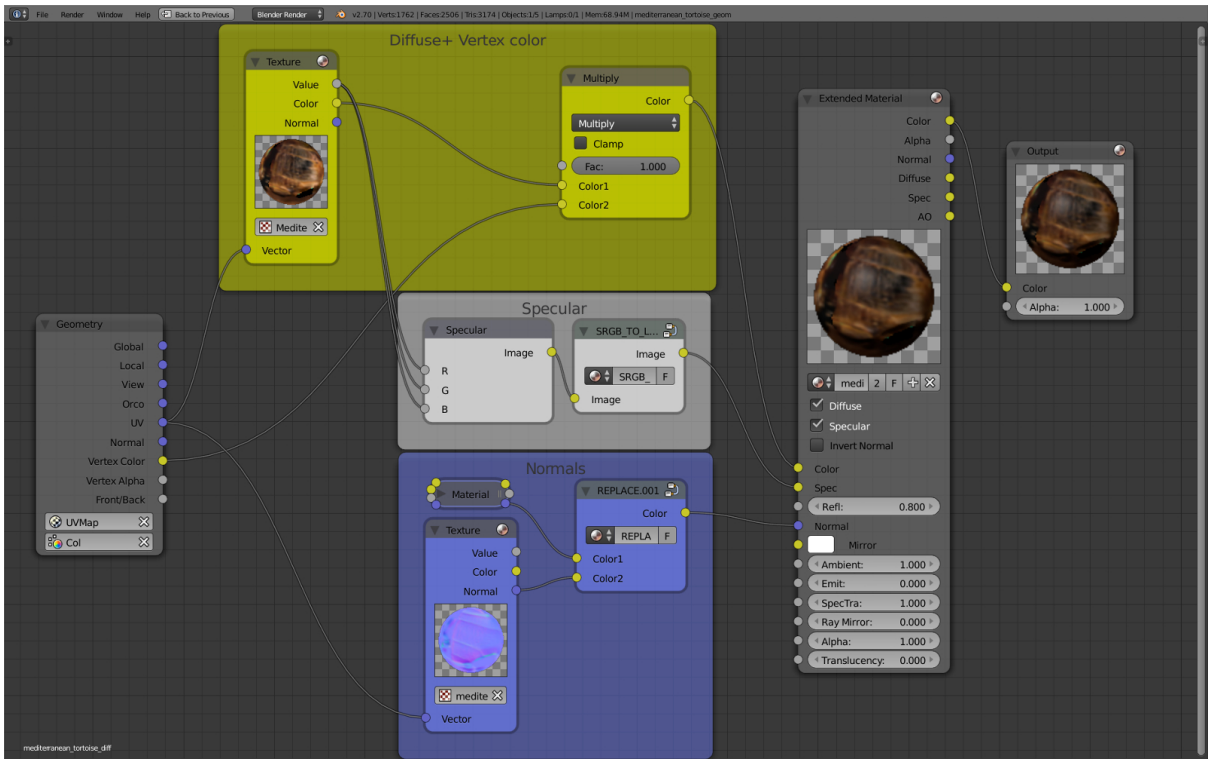
Color correction from linear color space to sRGB space and back. This functionality is declared deprecated since version 15.04 - use Blender's native Gamma node instead.

See also:

Gamma in Node Materials

13.2.2 B4W_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.



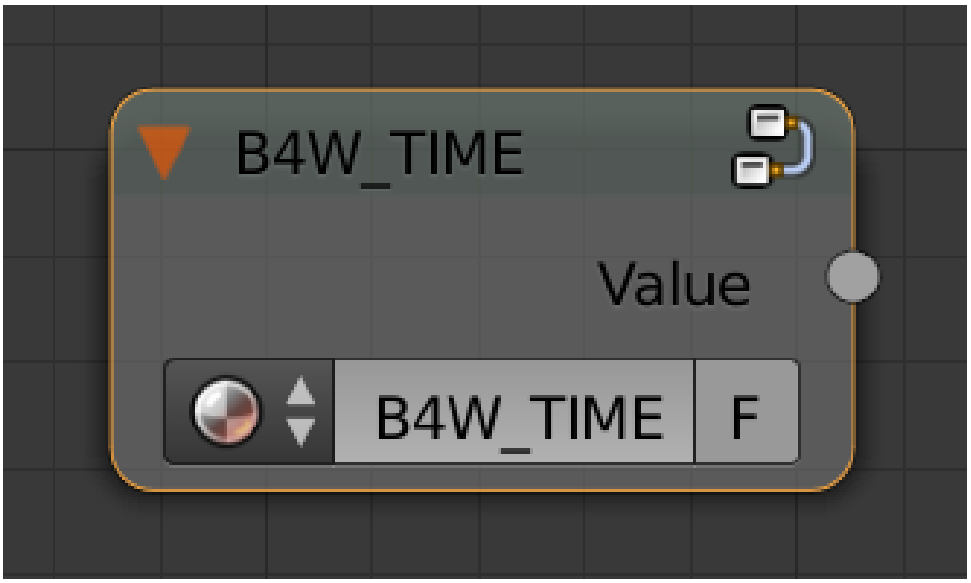
As a rule it is used for normal mapping. Blender's node materials do not support a tangent space of coordinates. Therefore the only possible method to display normal maps in the viewport correctly is their usage inside the Material nodes.

13.2.3 B4W_CLAMP

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

13.2.4 B4W_TIME

Provides the timeline counting from the engine start (in seconds). Can be used for animating any parameters in node materials, such as UV coordinates, mixing factors, transparency etc.



See also:

Animation of Value and RGB Nodes

13.2.5 B4W_VECTOR_VIEW

The node transforms a vector into the camera's space of coordinates. Transformation is necessary because the engine defines most of vectors in the world space of coordinates. If normal vector is being transformed by this node it should be used only for effects and not for connecting to the output of the Material or Extended Material nodes.

13.2.6 B4W_PARALLAX

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

Input parameters

UV Source texture coordinates

Height Map 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.

13.2.7 B4W_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.

Note: This node can work incorrectly, if the mesh normals were edited.

13.2.8 B4W_REFRACTION

Get the refracted image behind object.

Input parameters

Normal Normal map for adding perturbations.

Refraction bump Value of perturbation strength.

The default value is 0.001.

Output parameters

Color Rendered texture behind object with perturbations.

Note: It's neccessary to enable the Render refraction option from the Scene > Blend4Web panel. The object's transparency type must be set to Alpha Blend.

See also:

Transparency

13.2.9 B4W_LEVELS_OF_QUALITY

This node sets output color value depending on the quality profile.

See also:

Quality Profiles

Input parameters

HIGH Node links this parameter to Color parameter in case of high and maximum quality usage.

LOW Node links this parameter to Color parameter in case of low quality usage.

Output parameters

Color Output color.

Note: HIGH input parameter is used for rendering in the Blender viewport.

13.2.10 B4W_SMOOTHSTEP

Perform smooth interpolation between two input values based on first value.

Input parameters

Value Value which determines interpolation smoothness.

Edge0 First interpolation value.

Edge1 Second interpolation value.

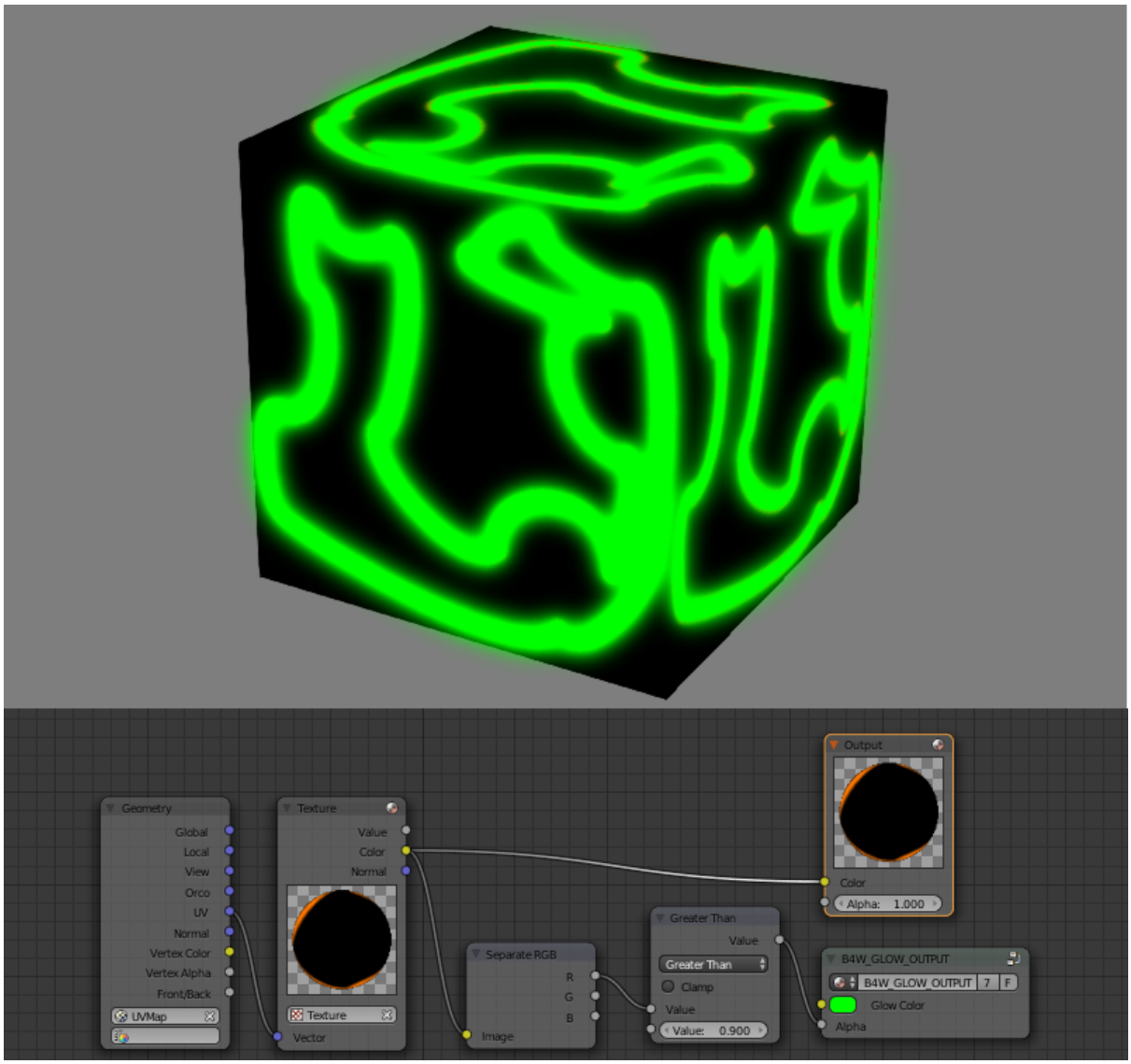
Output parameters

Value Interpolated value.

Note: For the correct interpolation input ‘’Value” had to be between ‘’Edge0” and ‘’Edge1’‘.

13.2.11 B4W_GLOW_OUTPUT

Applies the Glow effect to the node material. Besides the B4W_GLOW_OUTPUT node, the node material should have the Output.



Input parameters

Glow Color Glow color.

Factor Glow ratio. Factor $\in [0, 1]$.

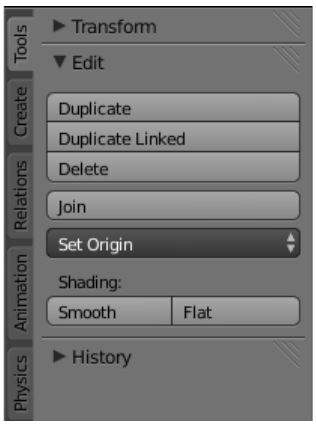
- Factor $= 0$ - no glow.
- Factor $\in (0, 1]$ - there is a glow, colored with Glow Color.

CHAPTER 14

Lighting, Shadows and Background

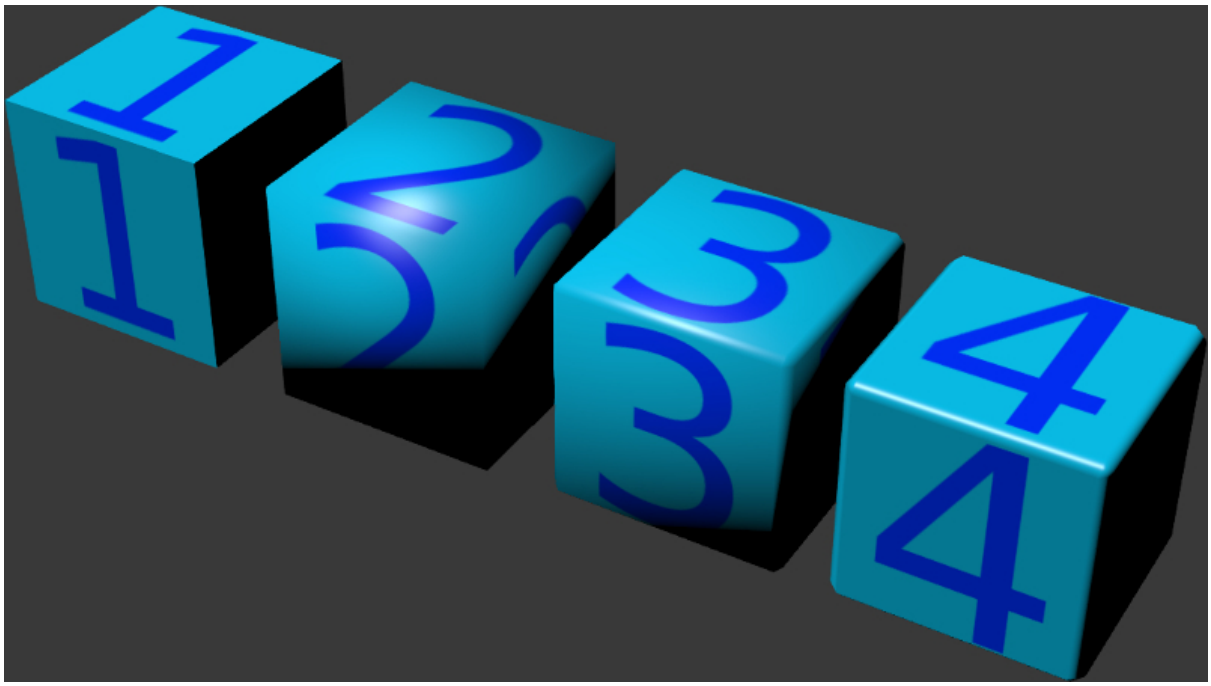
14.1 Shading Types

Lighting (shading) depends on the direction of normal vectors. The standard Blender's shading types are supported: Shading: Flat (face normals are used), Shading: Smooth (interpolated vertex normals are used) and their combinations.



If the required effect is impossible to achieve with the standard tools, you can use the normals editor.

The result of applying different shading types and using the normals editor:



1. Flat Shading
2. Smooth Shading
3. Smooth Shading + bevel
4. Smooth Shading + bevel + editing normals

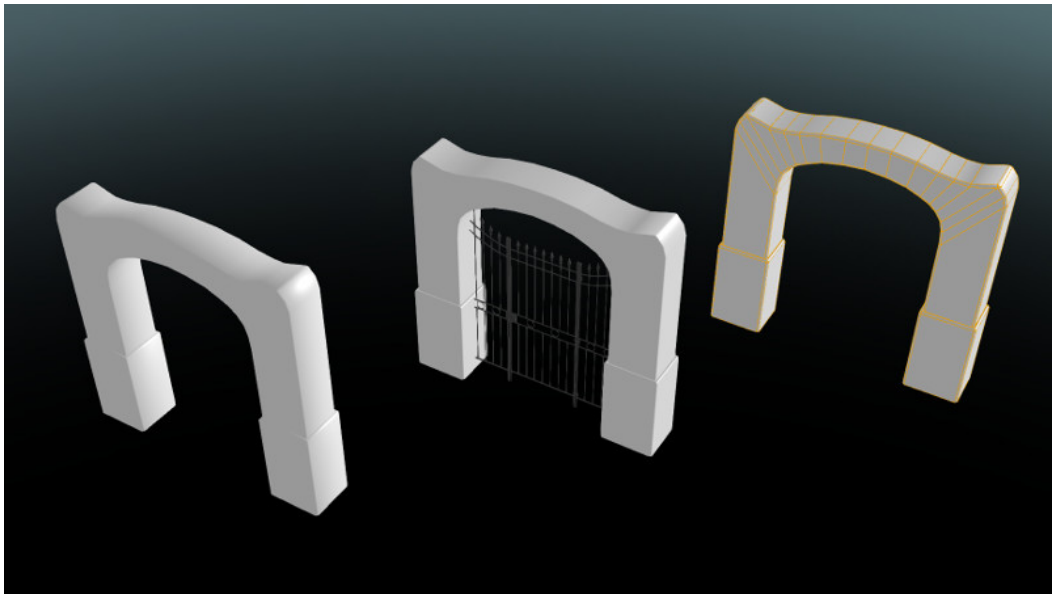
14.2 Normal Editor

Editing the selected normals is a pretty easy and effective way to customize 3d model shading without complicating its geometry.

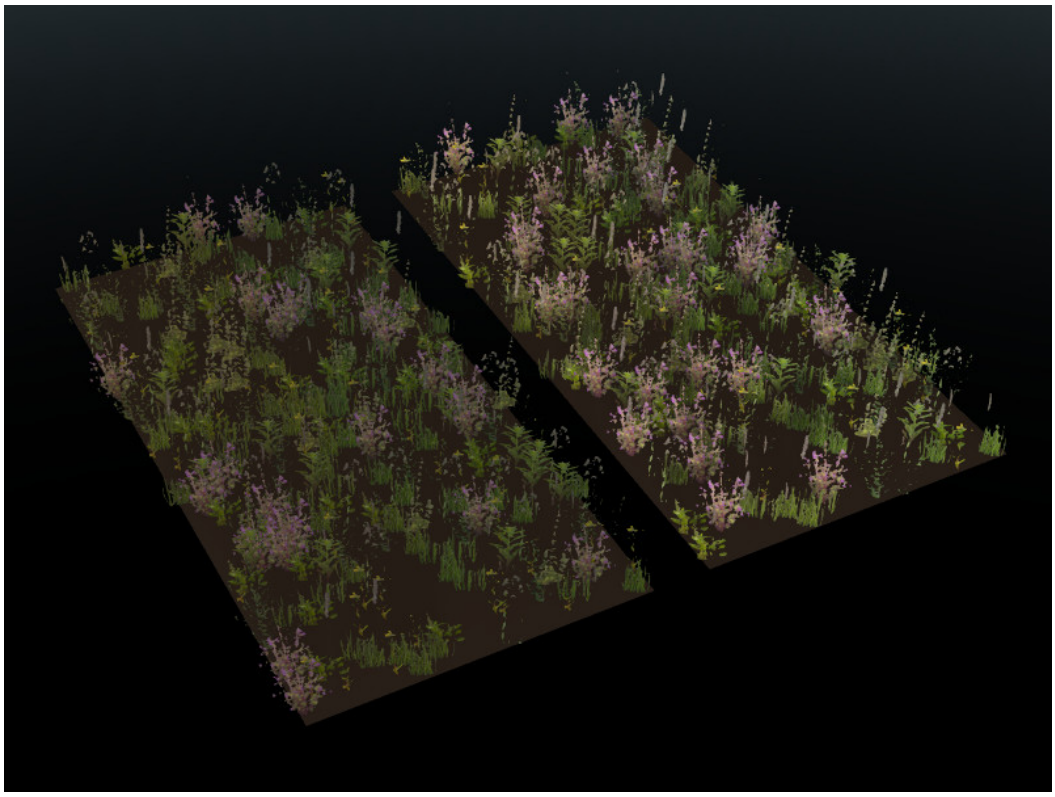
In some cases, using the normal editor you may achieve the effect which is similar to the result of using normal maps. At the same time, editing normals is preferred because it is more computationally effective and consumes less video memory.

Normal editor workflow example:

Simple geometry shading is to the left, a tree with edited normals is to the right.



To the left - gates with common shading; at the center - gates with edited normals; to the right - gates geometry with a wireframe.



To the left - common grass geometry shading; to the right - grass with edited normals.

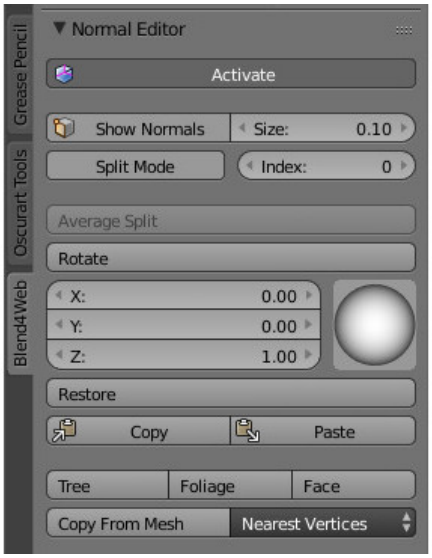
To the left - glasses with common shading; to the right - geometry with edited normals shading.

14.2.1 Main features of the normal editor:

1. native Blender storage is used as a container for edited vertices normals directions (it appeared in Blender 2.74);
2. normals editing and shading visualization are processed in Edit Mode now;
3. all changes are being saved automatically;
4. selected vertex normal rotation can be performed directly in the Viewport window with Shift+Ctrl+R hotkey, similary to other rotation operations in Blender;
5. edited normals are being exported automatically.

14.2.2 Interface of the Vertex Normal Editor

The interface of the normal editor is located on the tools panel Blend4Web > Normal Editor. The Shading: Smooth mode should be enabled and Activate button should be pressed or Auto Smooth flag should be enabled in the mesh settings, before starting to work with the editor.



14.2.3 Activate

Description

The new Activate button turns on vertex normal editing mode.

Usage

You can just press Activate button while in Edit Mode and start editing vertex normals. As it is active, object shading and its export would be processed taking edited vertex normals into account. In other words, after making some changes, this button should be left active if you want to see the changes in the Blender Viewport and in the Blend4Web engine.

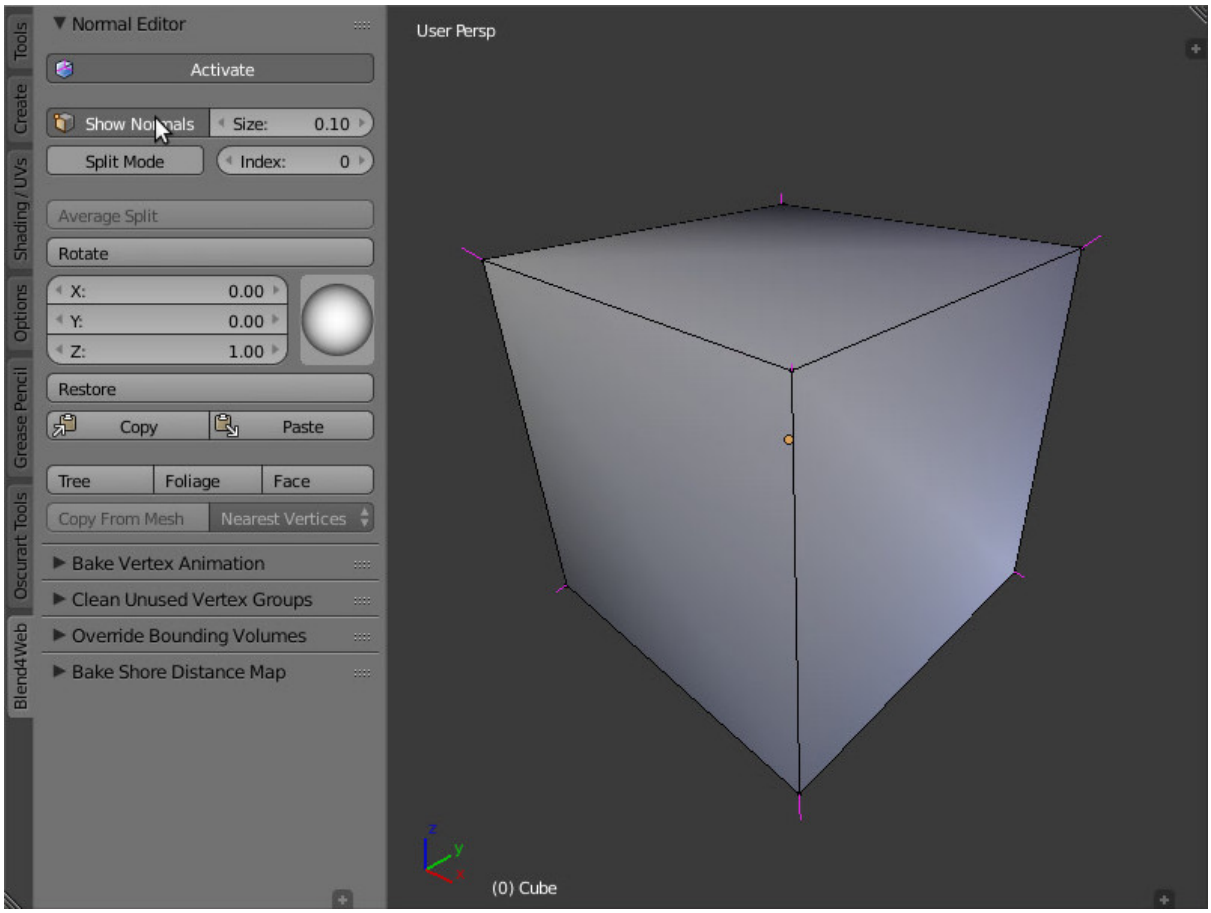
14.2.4 Show Normals

Description

This button actually duplicates the original Blender's button. It turns on displaying the normals in the viewport, while the Size field allows you to set their length.

Usage

You just need to push the Show Normals button either on the normal editor panel or on the Blender's right panel in the Mesh Display section. You can also set the convenient length of normals by tweaking the Size.



14.2.5 Rotation

Description

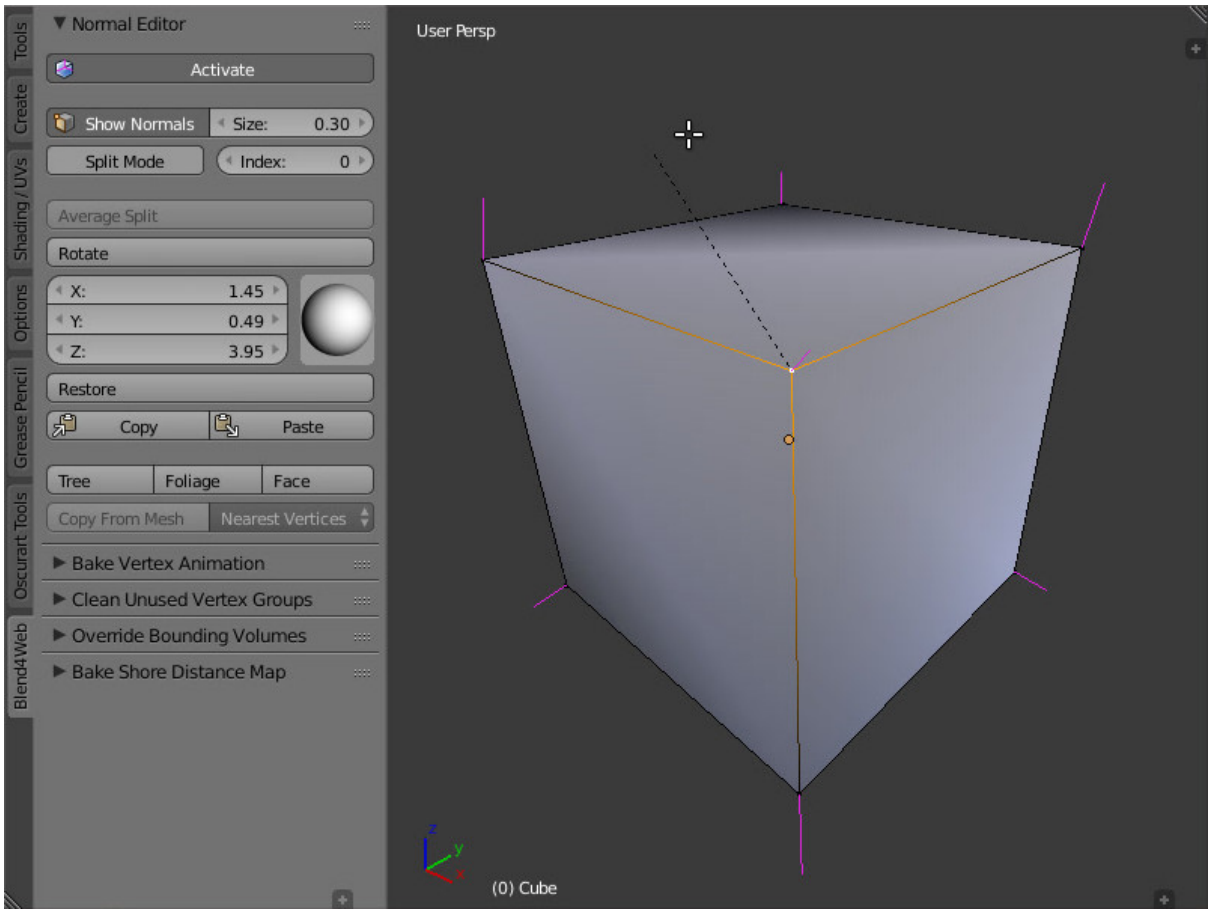
Using these instruments you can change direction of the normals.

The Rotate function is also available through the Shift+Ctrl+R hotkeys, which allows rotation of vertex normals similarly to Blender.

Usage

Select one or more vertices that you want to edit and then rotate their normals using the visual sphere or specify the direction as numerical values.

The Rotate button provides a more convenient way to manipulate normals. The rotation is performed in the screen space. Nevertheless, as with any other rotations in Blender, you can isolate the desired axis during rotation (by typing X, Y or Z).



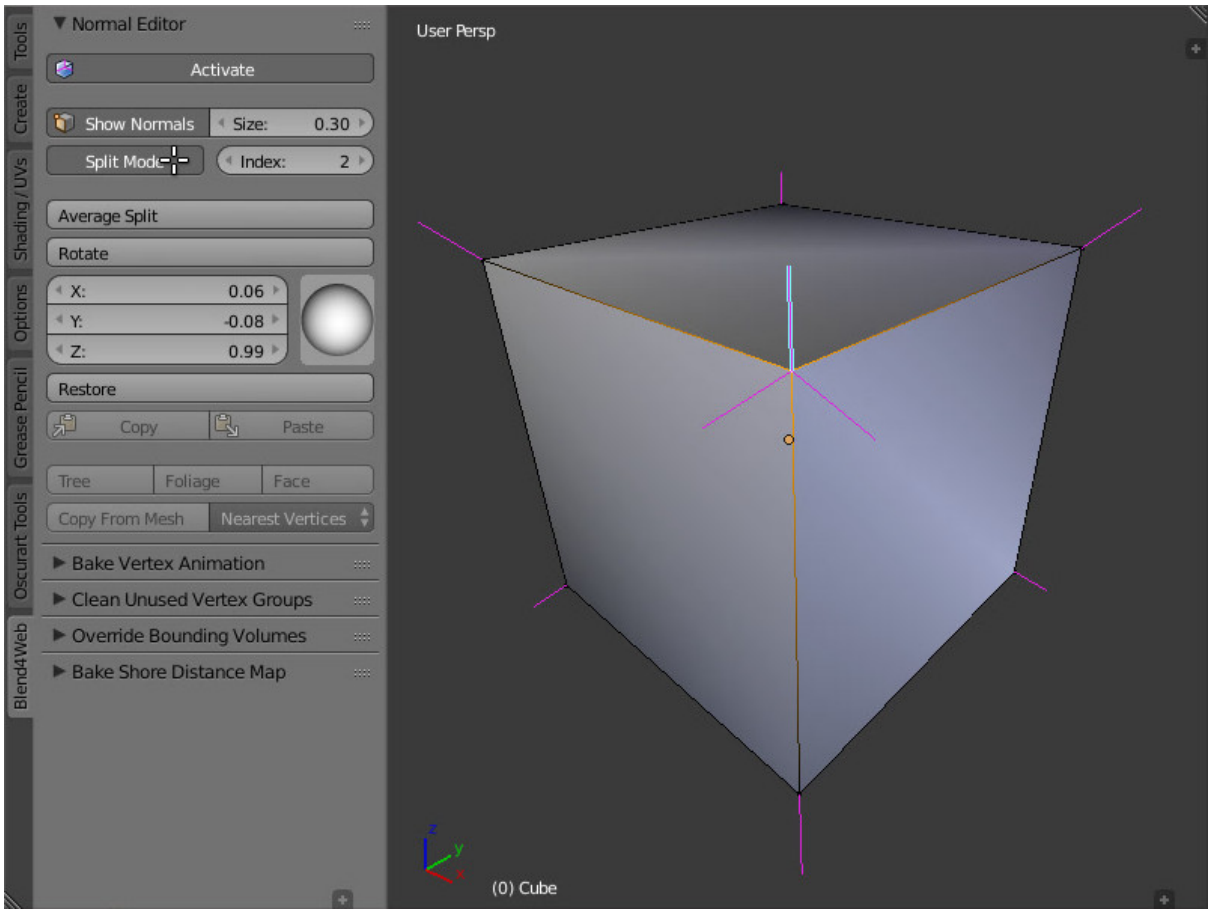
14.2.6 Split Normals

Description

The Split Normals mode allows to edit vertex normals separately for each face that form the vertex. Index allows you to navigate between the split normals.

Использование

Turn on the Split Normals mode, select the vertex and change the direction of its normals. Firstly, the normal which has the zero index in the queue will be modified. Then, by switching between indexes you can go to the next normal of this vertex and edit it, then to the next and so on.



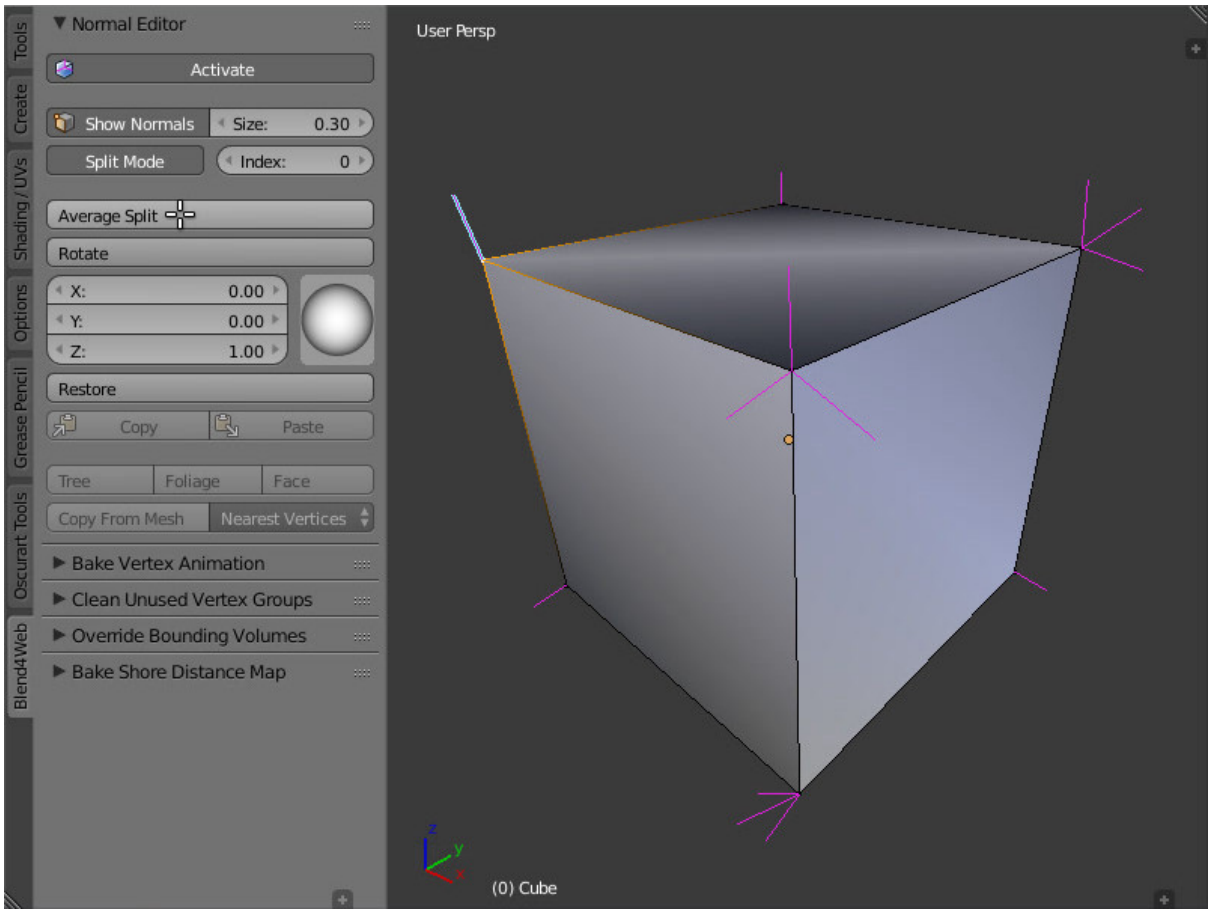
14.2.7 Average Split

Description

The Average Split button averages the direction of the vertex normals which was modified in the Split Normals mode.

Usage

To combine several split vertex normals into one, in order to obtain the average direction of these normals, you just need to select the desired vertex and press the Average Split button.



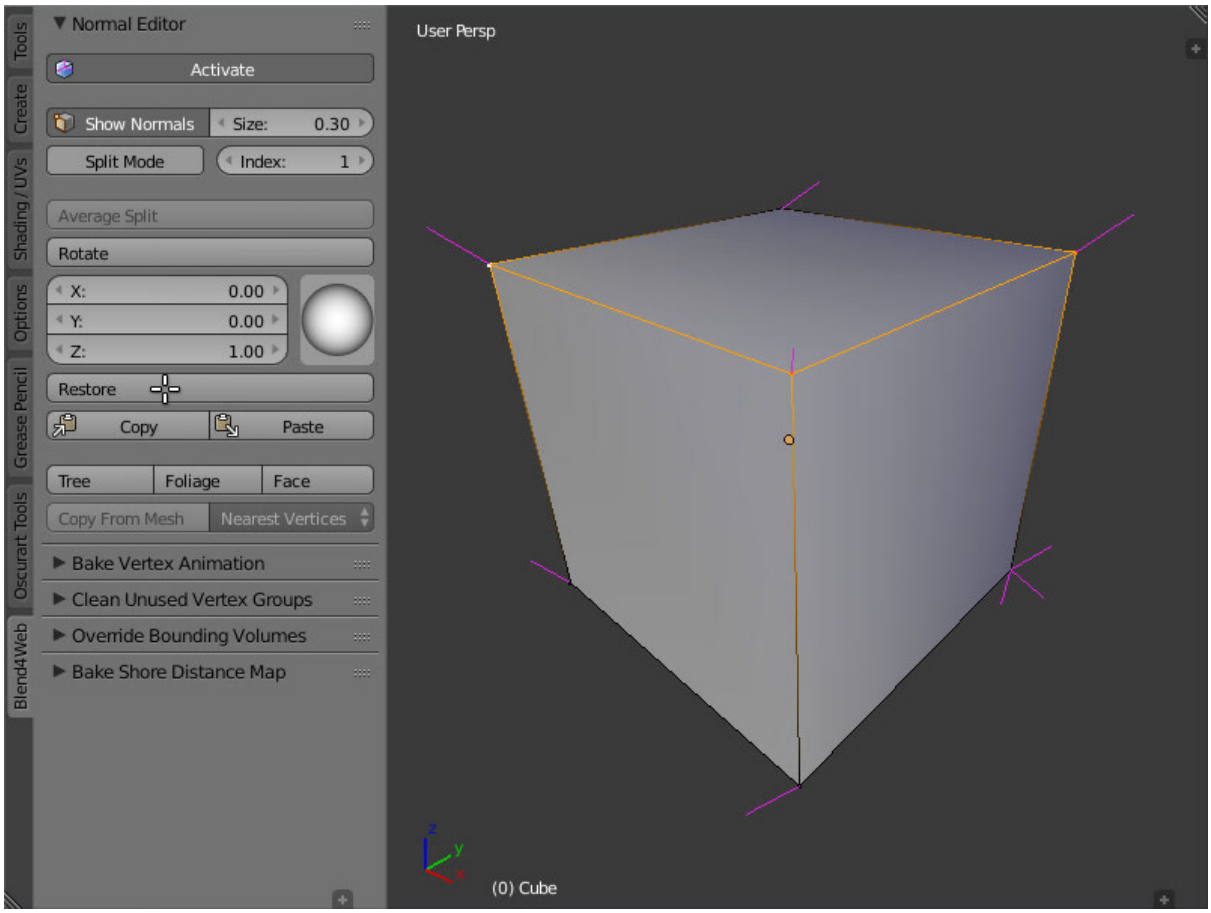
14.2.8 Restore

Description

The Restore button restores the original direction of normals for the selected vertices.

Usage

In order to restore the normal's direction to its original (which is calculated on the basis of face normals), you need to select the desired vertices and click the Restore button.



14.2.9 Tree, Foliage и Face

Description

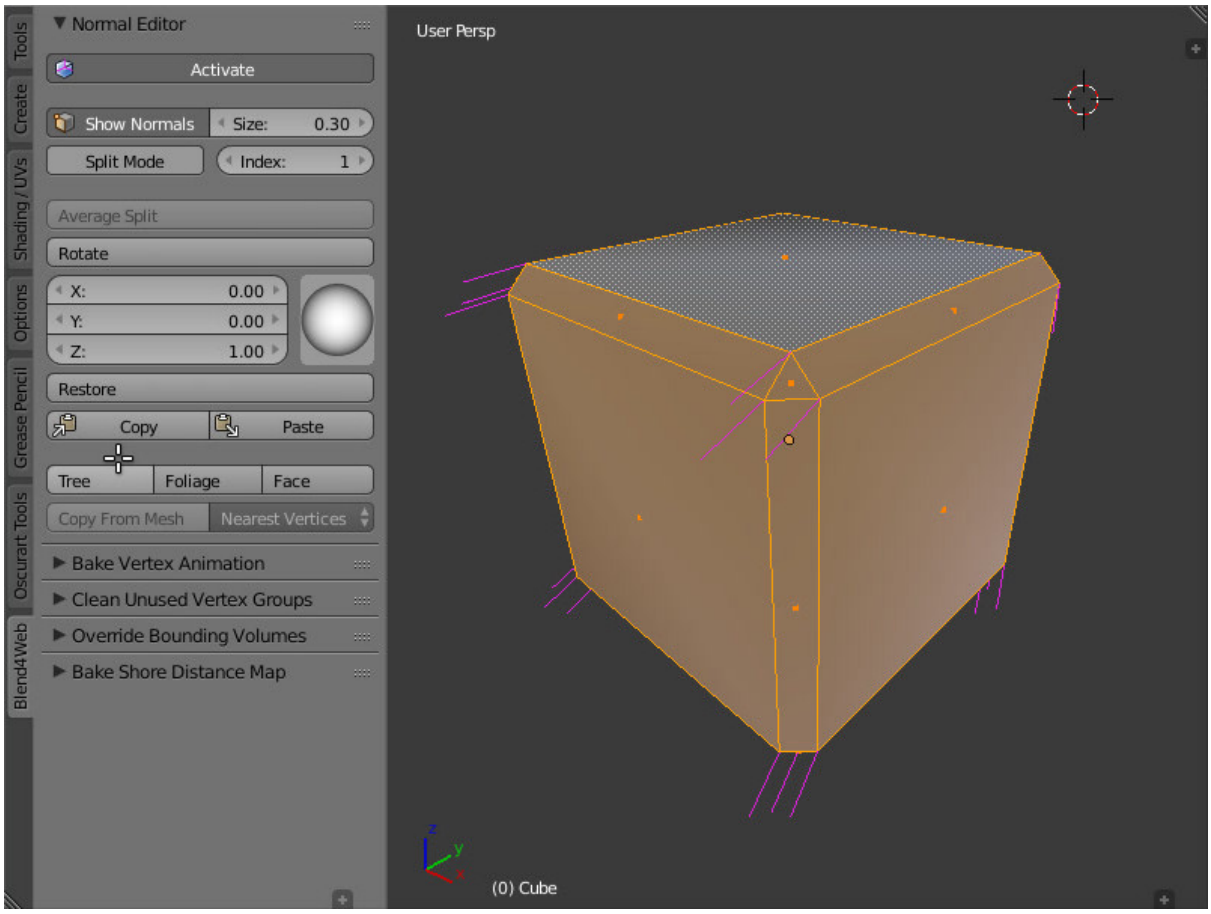
The Tree button directs the normals of the selected vertices away from the 3D cursor. The Foliage button directs the normals of the vertices strictly up. The Face button directs the normals of the selected face parallel to the normal of this face.

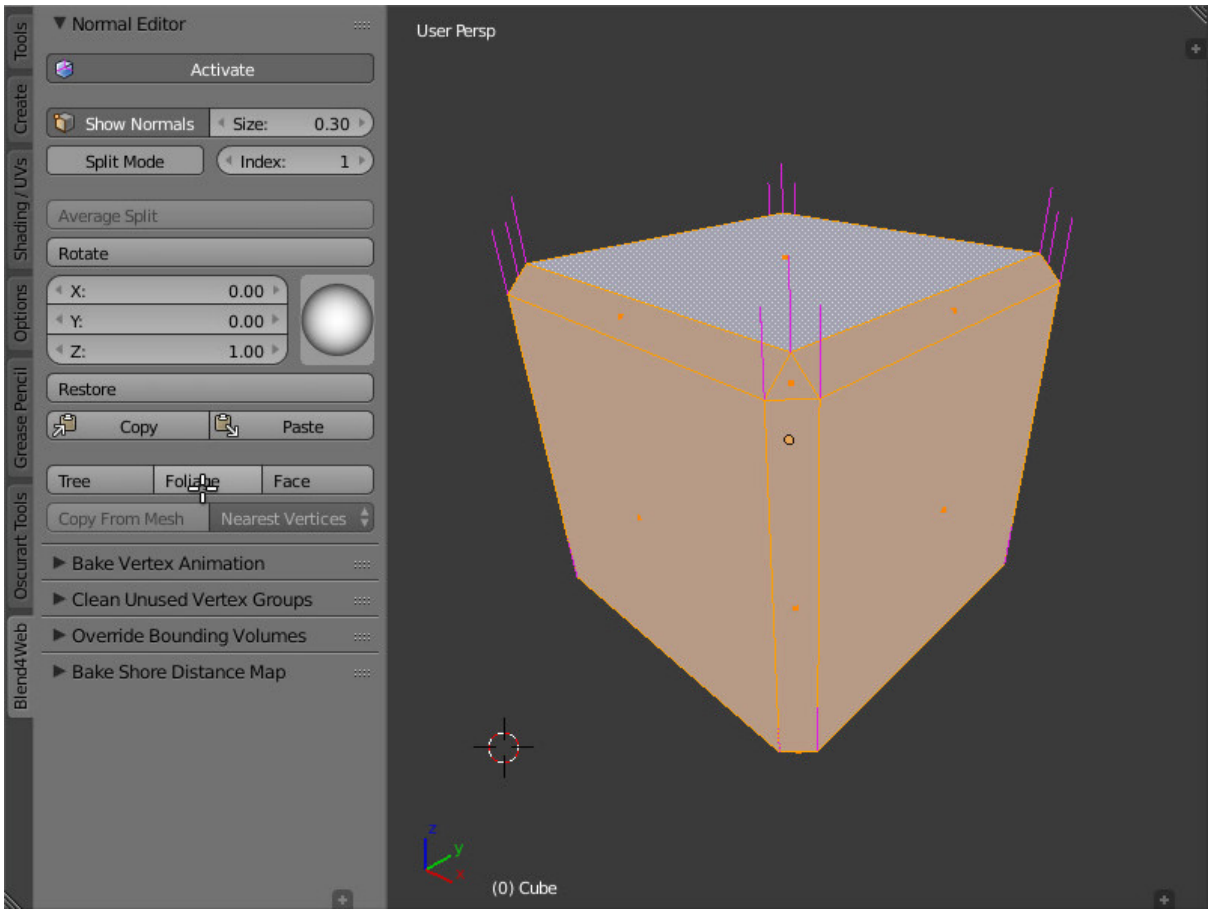
Использование

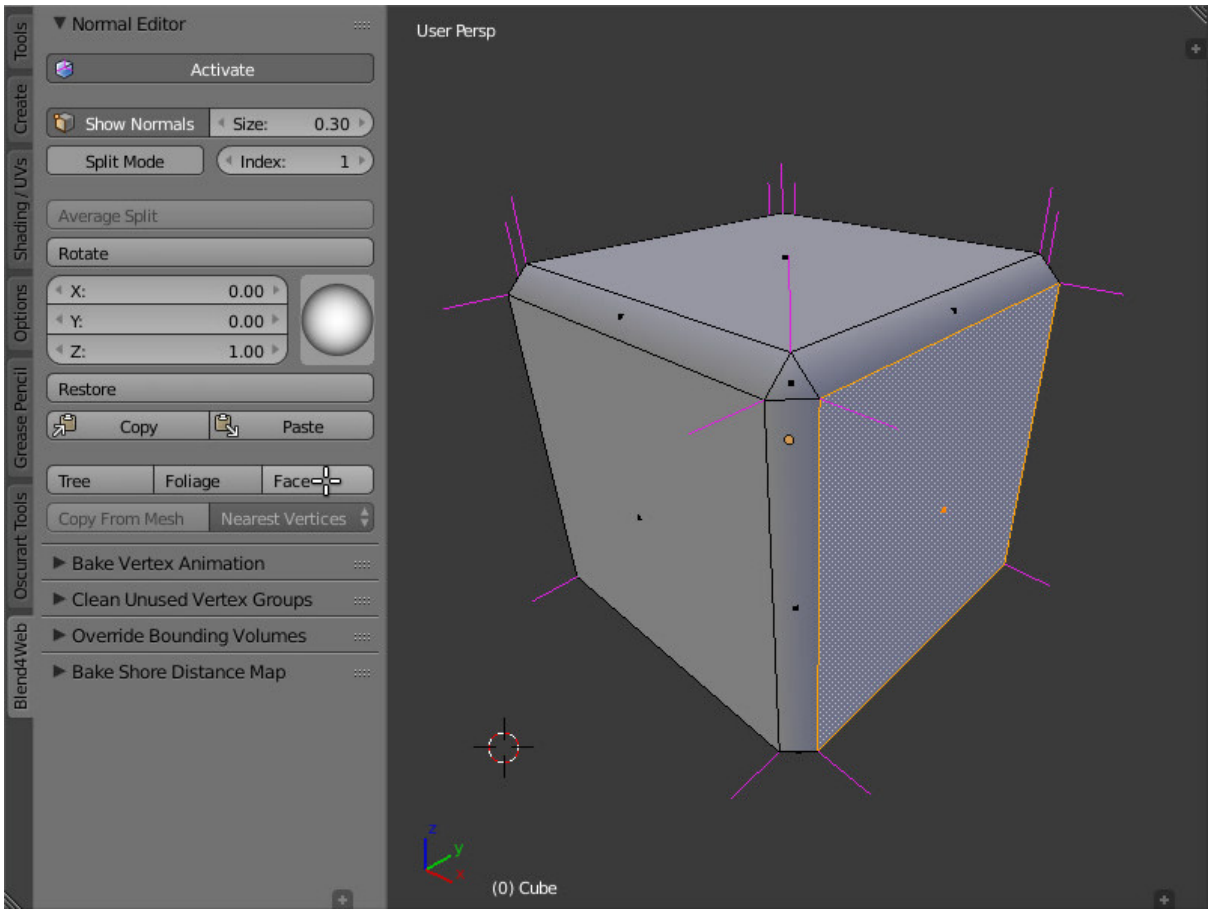
In order to use the Tree function, select the desired vertices and place the 3D cursor in the desired position. Then click the Tree button so all the selected vertices will turn their direction away from the cursor, as if they were shot from one point.

The Foliage function is very easy to use: just select the vertices and press the button, so the function will direct their normals straight up.

In order to direct the normals parallel to the face normal, just select the desired face and click the Face button. The normals of the vertices which form the face will be directed parallel to the face normal. This function works only with one selected face at a time.


Normal Editor
Activate
Show Normals
Size: 0.30
Split Mode
Index: 1
Average Split
Rotate
X: 0.00
Y: 0.00
Z: 1.00
Restore
Copy
Paste
Tree
Foliage
Face
Copy From Mesh
Nearest Vertices
Bake Vertex Animation
Clean Unused Vertex Groups
Override Bounding Volumes
Bake Shore Distance Map
Tools
Create
Shading / UVs
Options
Grease Pencil
Oscurart Tools
Blend4Web
User Persp
(0) Cube


Normal Editor
Activate
Show Normals
Size: 0.30
Split Mode
Index: 1
Average Split
Rotate
X: 0.00
Y: 0.00
Z: 1.00
Restore
Copy
Paste
Tree
Face
Copy From Mesh
Nearest Vertices
Bake Vertex Animation
Clean Unused Vertex Groups
Override Bounding Volumes
Bake Shore Distance Map
Tools
Create
Shading / UVs
Options
Grease Pencil
Oscurart Tools
Blend4Web
User Persp
(0) Cube



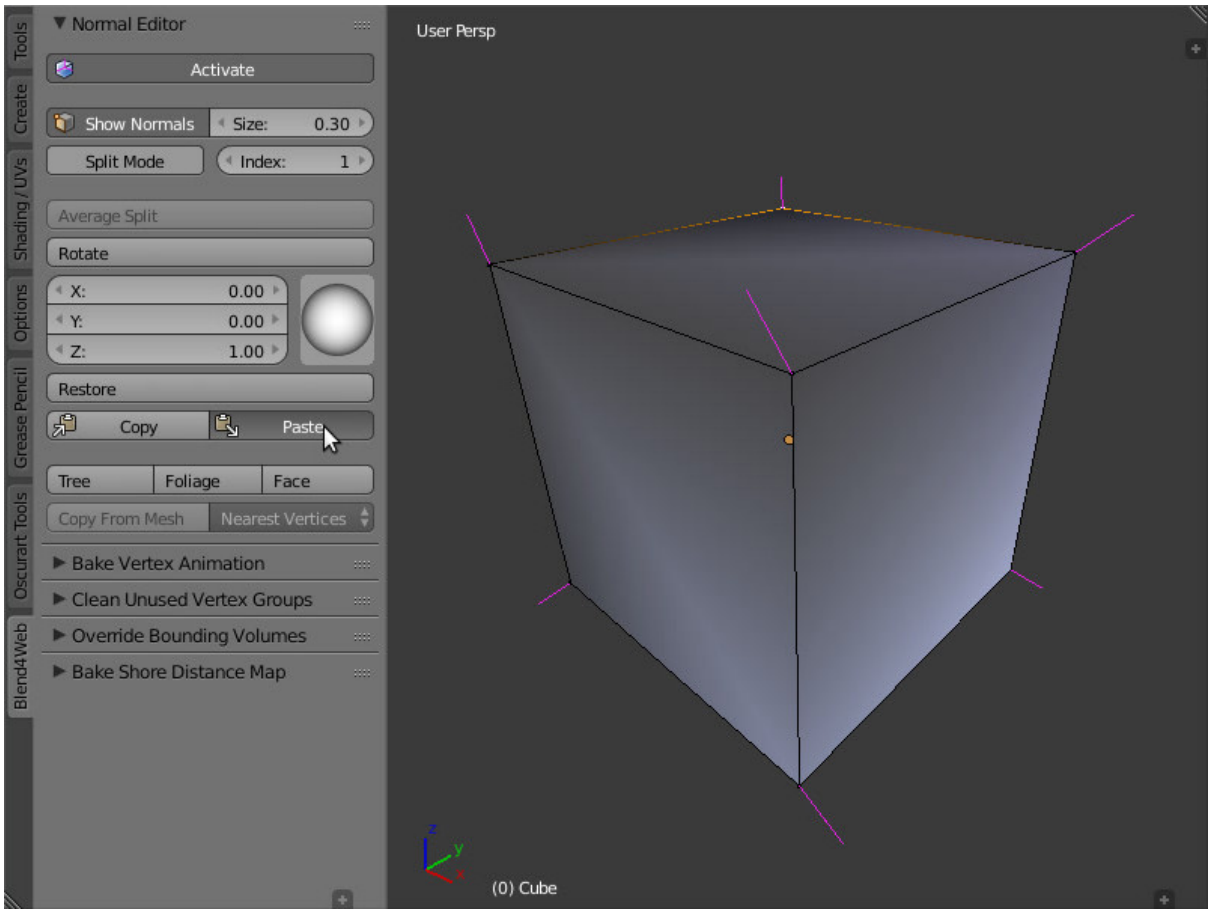
14.2.10 Copy/Paste

Description

Copies the normal direction from one vertex to another.

Usage

Select the vertex you want to copy from and click the Copy button. Then, select the vertex you want to copy to and click the Paste button. You may copy information from one selected vertex to many different vertices. The buttons are not active in the Split Mode and do not copy data of split vertices.



14.2.11 Copy From Mesh

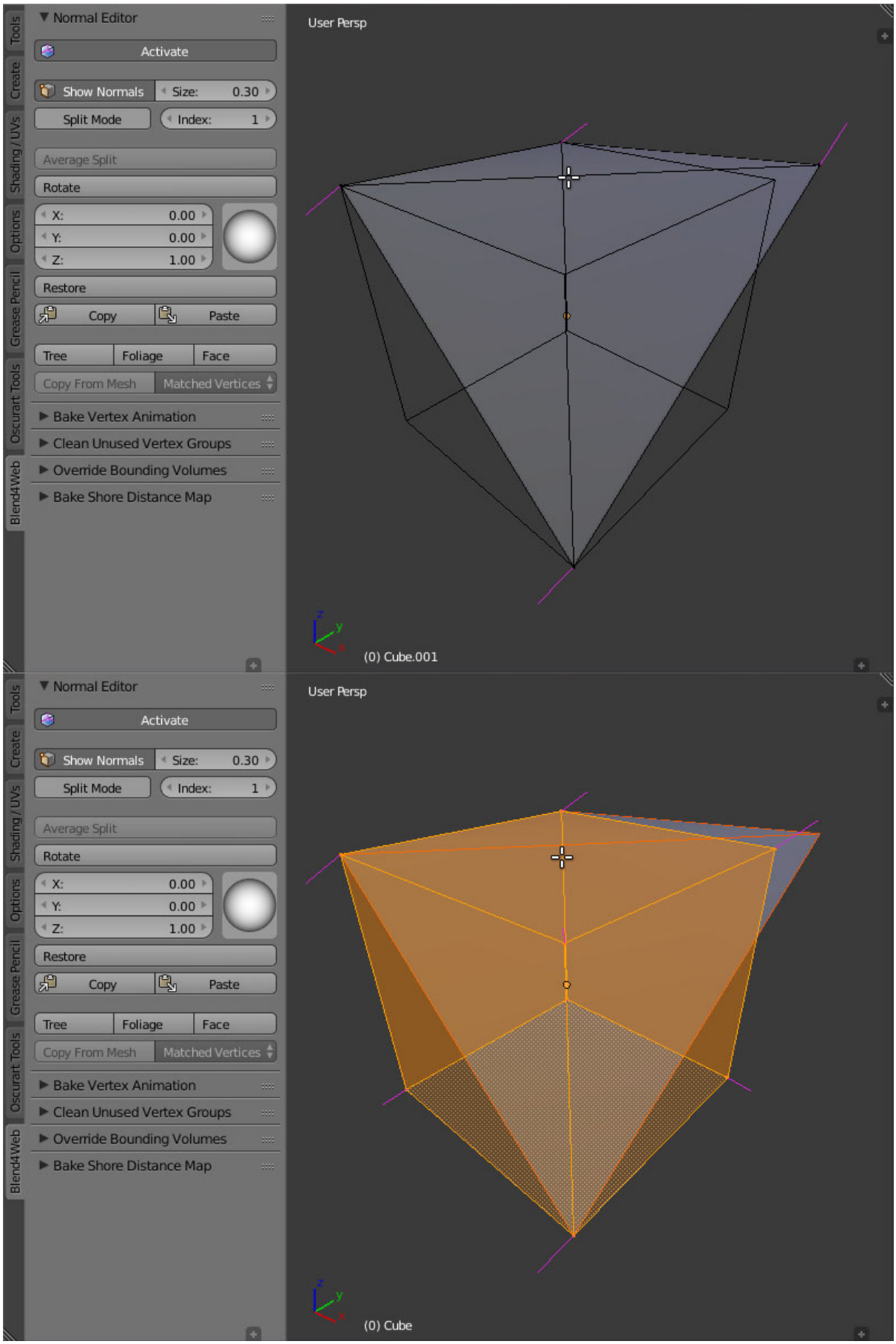
Description

This function allows you to copy the normals from one object to another. There are two modes: Matched Vertices and Nearest Vertices.

The Matched Vertices mode copies the normals from the vertices of one object to another object's vertices only if they have the same coordinates; the other vertices are ignored. The Nearest Vertices mode copies the normals of the nearest vertices of the source object.

Usage

This function works in Blender's object mode. At first, you need to select the object you need to copy from, then the object you need to copy to. It is also necessary to select the target object's vertices to which normals are copied.


Tools
Create
Shading / UVs
Options
Grease Pencil
Oscurart Tools
Blend4Web
Normal Editor
Activate
Show Normals
Size: 0.30
Split Mode
Index: 1
Average Split
Rotate
X: 0.00
Y: 0.00
Z: 1.00
Restore
Copy
Paste
Tree
Foliage
Face
Copy From Mesh
Matched Vertices
Bake Vertex Animation
Clean Unused Vertex Groups
Override Bounding Volumes
Bake Shore Distance Map
User Persp
(0) Cube.001
Normal Editor
Activate
Show Normals
Size: 0.30
Split Mode
Index: 1
Average Split
Rotate
X: 0.00
Y: 0.00
Z: 1.00
Restore
Copy
Paste
Tree
Foliage
Face
Copy From Mesh
Matched Vertices
Bake Vertex Animation
Clean Unused Vertex Groups
Override Bounding Volumes
Bake Shore Distance Map
User Persp
(0) Cube

14.3 Lighting with Light Sources

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

14.3.1 Light source types

The following light source types are supported:

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

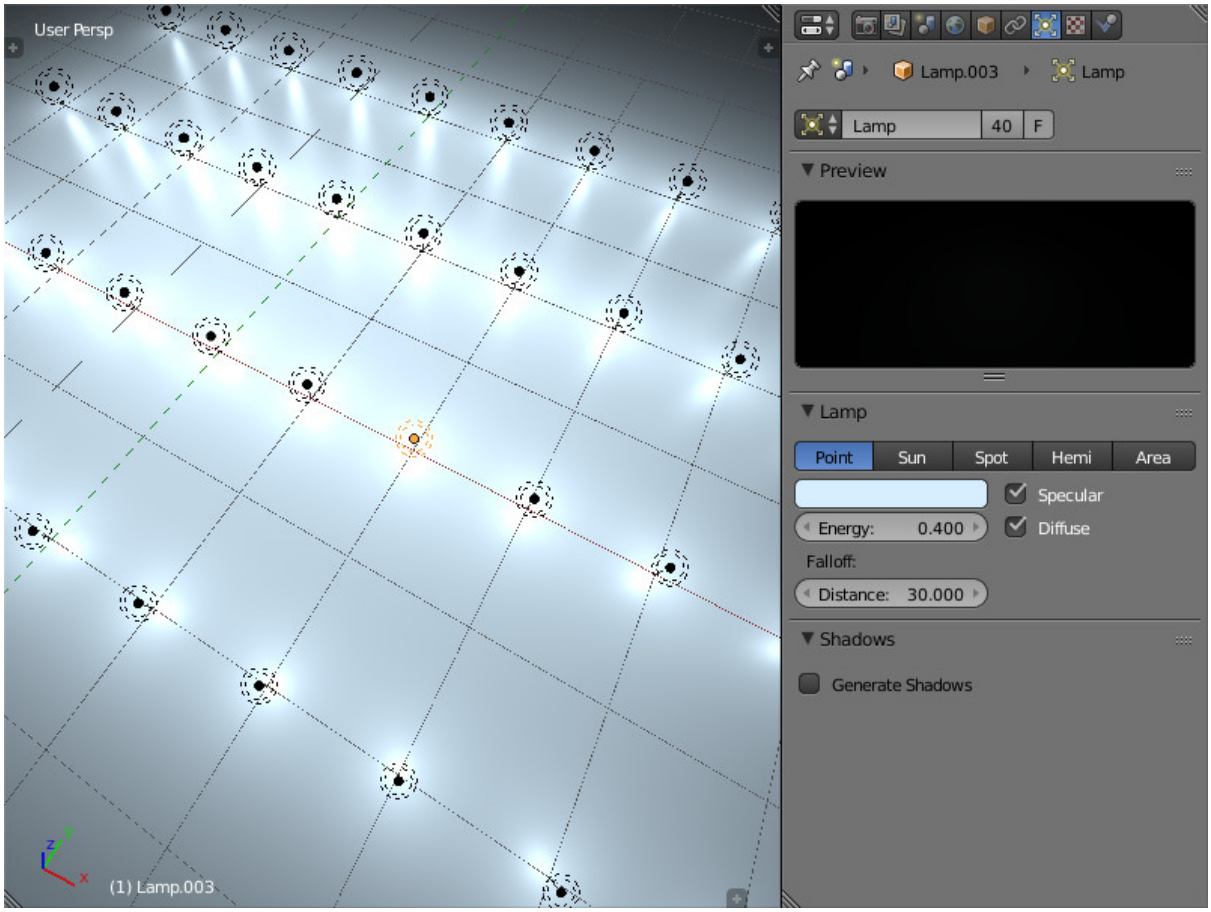
Sun Light propagates from an infinite plane in one direction without attenuation.

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

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

14.3.2 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.

Specular Create specular highlights. Enabled by default.

Diffuse Do diffuse shading. Enabled by default.

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.

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

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

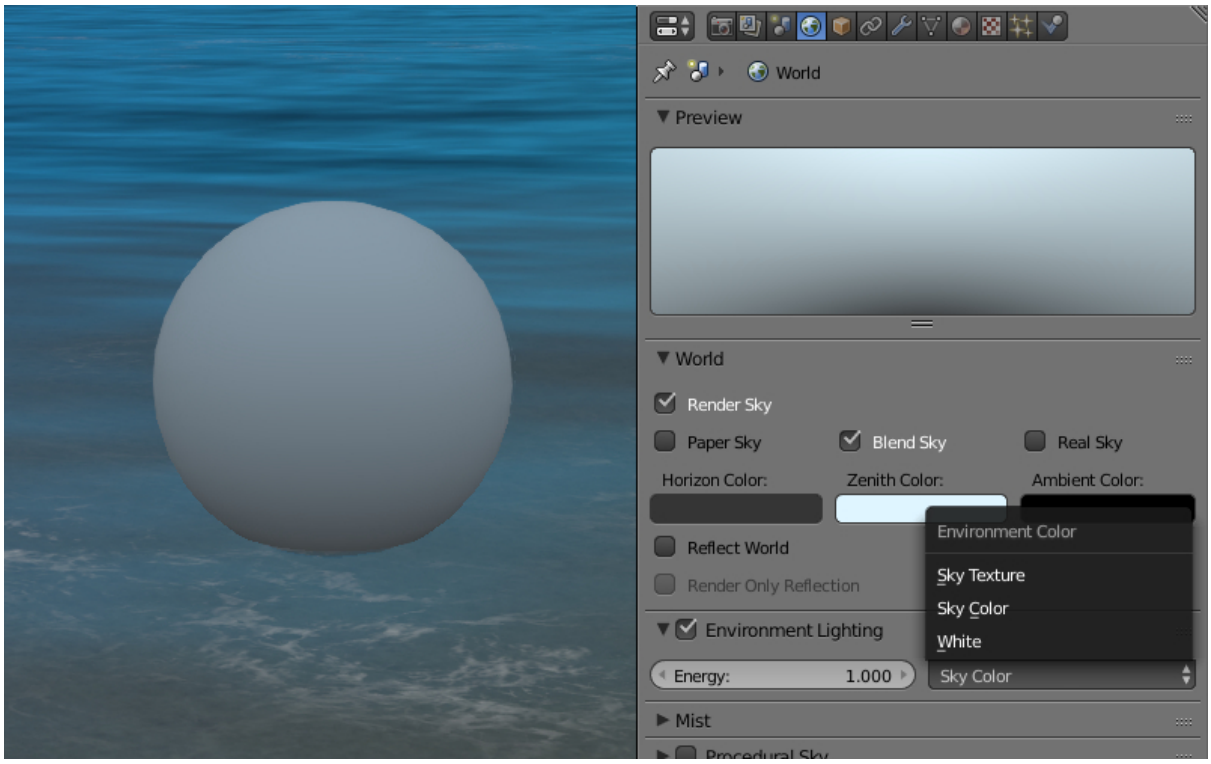
14.4 Environment Lighting (Ambient)

The engine supports 3 methods of the environment lighting simulation.

1. Flat white lighting.
2. Hemispherical lighting model in which horizon and zenith colors should be specified. As a result objects are filled with a gradient between these two colors depending on the direction of normals.
3. Lighting using an environment map - so called image-based lighting.

14.4.1 Activation

Enable the Environment Lighting checkbox on the World tab.



14.4.2 Setup

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

World > Environment Lighting > Environment Color Selection of the environment lighting simulation method: White - flat white lighting, Sky Color - hemisperical model, Sky Texture - lighting using an environment map. The default value is White.

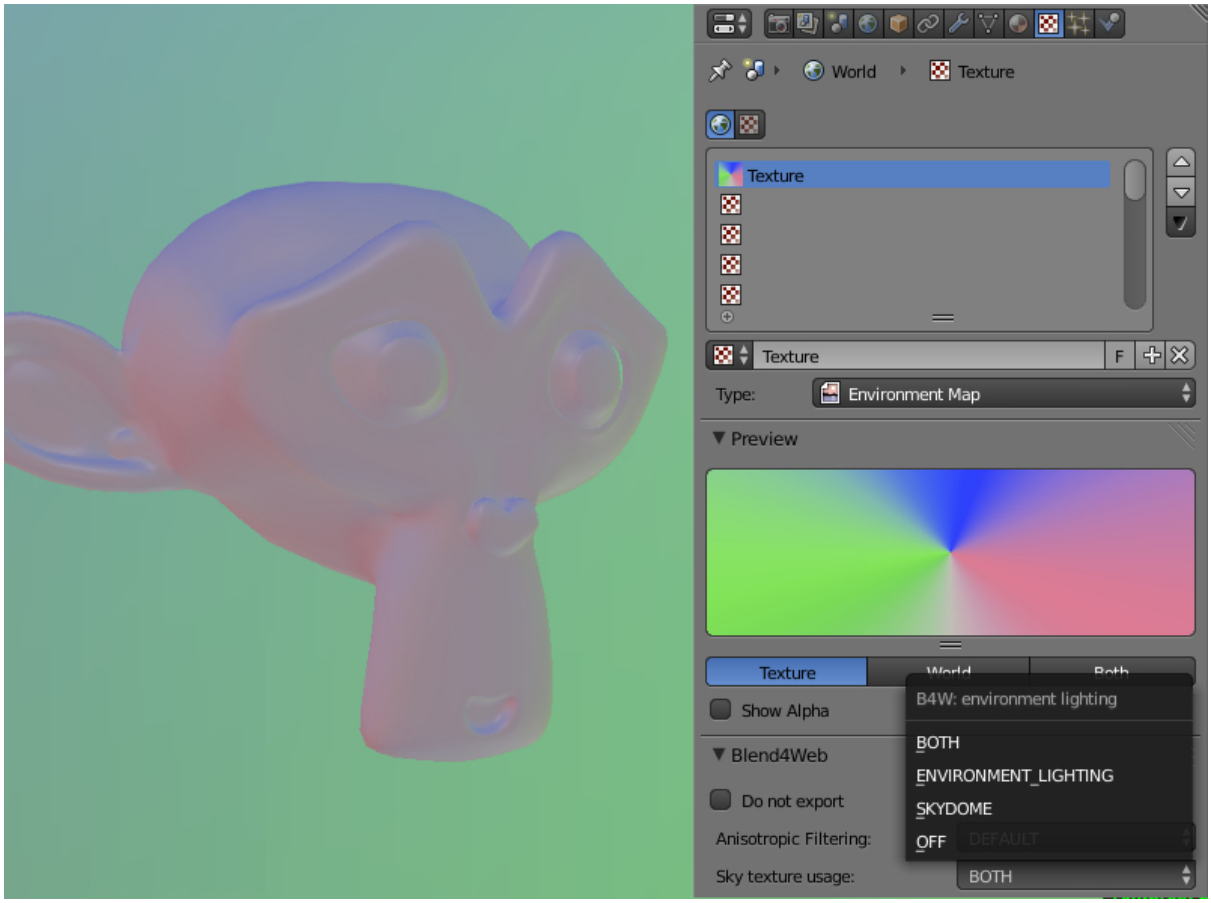
World > Horizon Color and World > Zenith Color If the hemisperical model (Sky Color) is selected the horizon and zenith colors can be specified by means of the World > Horizon Color and World > Zenith Color color pickers. It is recommended to activate the World > Blend Sky option for better color selection.

14.4.3 Environment map method

To use an environment map for environment lighting:

1. Enable the Environment Lighting checkbox on the World tab.
2. Select the Environment Lighting > Environment Color > Sky Texture method.
3. Go from the World tab to the Texture tab.
4. Create an environment map, load the corresponding image to it.
5. For the environment map select ENVIRONMENT_LIGHTING or BOTH as the Sky texture usage value on the Blend4Web panel (the BOTH option also enables

using this texture as a skydome texture).



14.5 Shadows

Shadows are exceptionally important for rendering the final picture. They provide the viewer not only with information about the objects' outline but also about their height and relative position, light source position and so on.

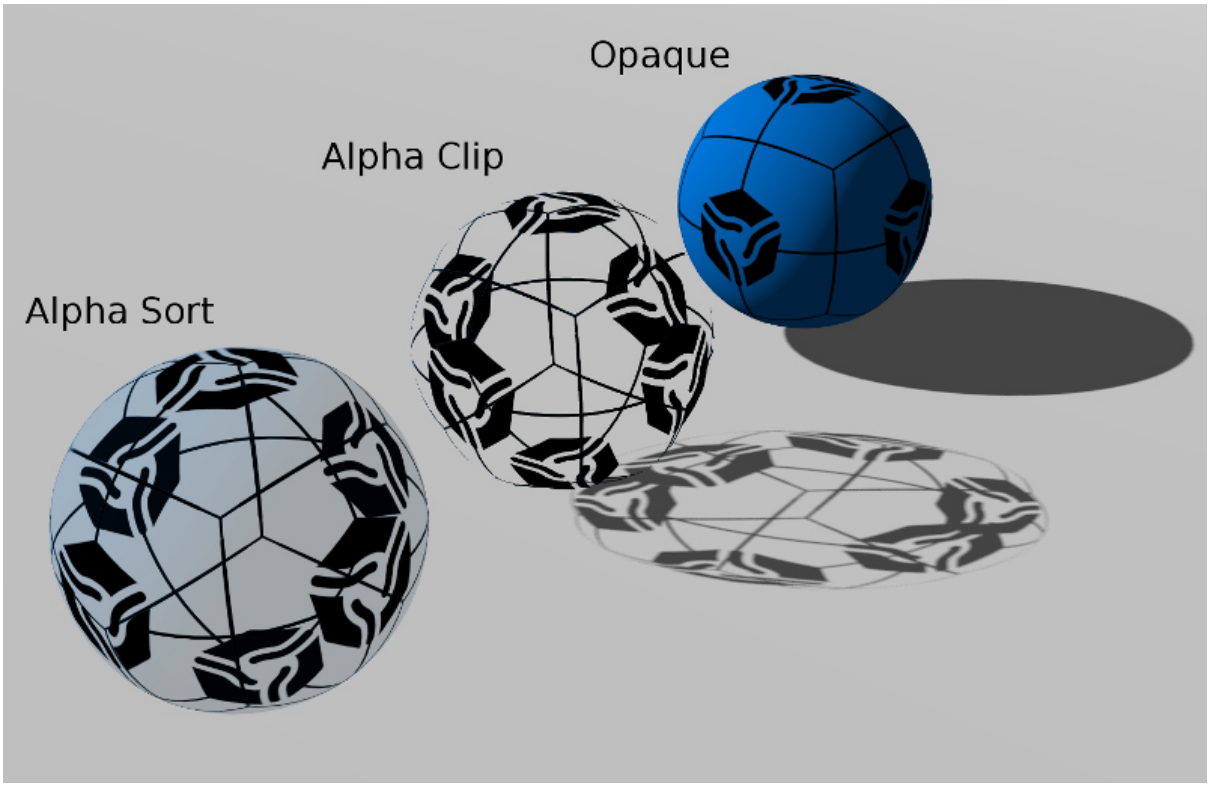
Blend4Web implements the following shadow rendering techniques: cascaded shadow maps (CSM) and softened shadows (PCF).

14.5.1 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.

Note: Objects, which have transparent materials with a gradient, do not cast shadows.

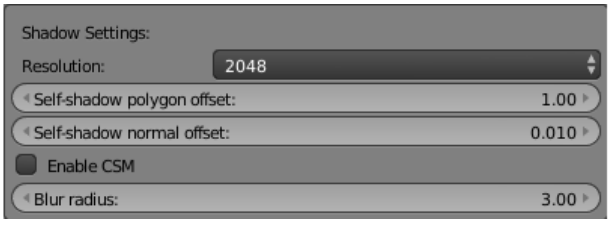


14.5.2 Setup

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 the environment lighting settings.

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



Resolution Shadow map resolution. The default value is 2048x2048px.

Self-shadow polygon offset Coefficient for shifting polygons relative to light source orientation. The default value is 1.

Self-shadow normal offset Coefficient for shifting polygons along their normals. The default value is 0.01.

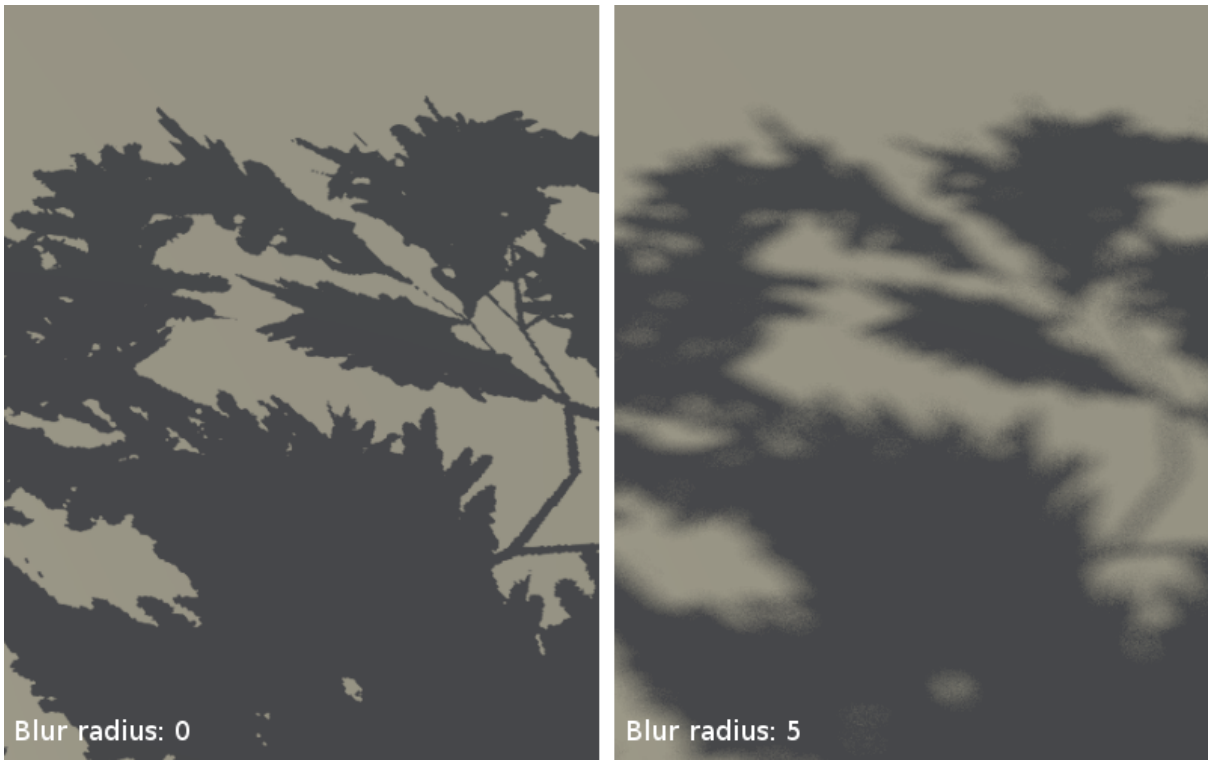
The last two parameters can be used to reduce self-shadowing artifacts. These artifacts appear for the objects that cast and receive shadows at the same time. The Self-shadow polygon offset parameter is more effective for fighting against artifacts in inner areas of poligons while Self-shadow normal offset is better for the boundary areas. Both these parameters lead to shadow distortions so we recommend to set them as low as possible.

Enable CSM Activates the using of cascaded shadows model; reveals additional options. Disabled by default.

This option allows to choose between the following shadow generation models:

- Generic model which uses a single optimized shadow map for the whole scene (Enable CSM is turned off).
- Shadow cascades (Enable CSM is turned on).

Blur radius Blur ratio for setting up softened shadows. The default value is 3. Zero value produces hard shadows.



Softened shadows can improve visual quality and realism. They hide the jugged edges inevitable when using image-based techniques, that is especially noticable for low-resolution shadow maps. The using of softened shadows often allows to decrease resolution without substantial quality loss.

Generic shadows

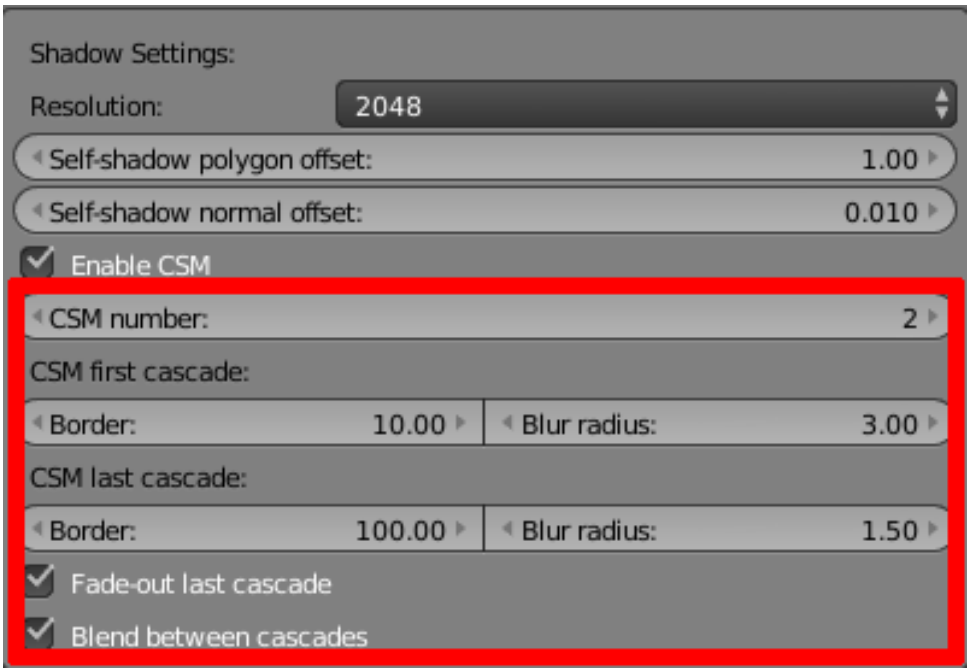
This option suits well smaller scenes with a limited number of objects. Thanks to optimizations applied for such scenes, one can achieve better shadow quality as compared with cascaded shadows. Also this option is simpler and faster for setting up, while using a single shadow map greatly improves the performance.

Shadow cascades

Note: These settings are supported only for Sun light sources. Cascades are turned off for other types of light sources.

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. This option suits well middle-to-large scenes, e.g. game levels.

When enabled the following extended settings are revealed:



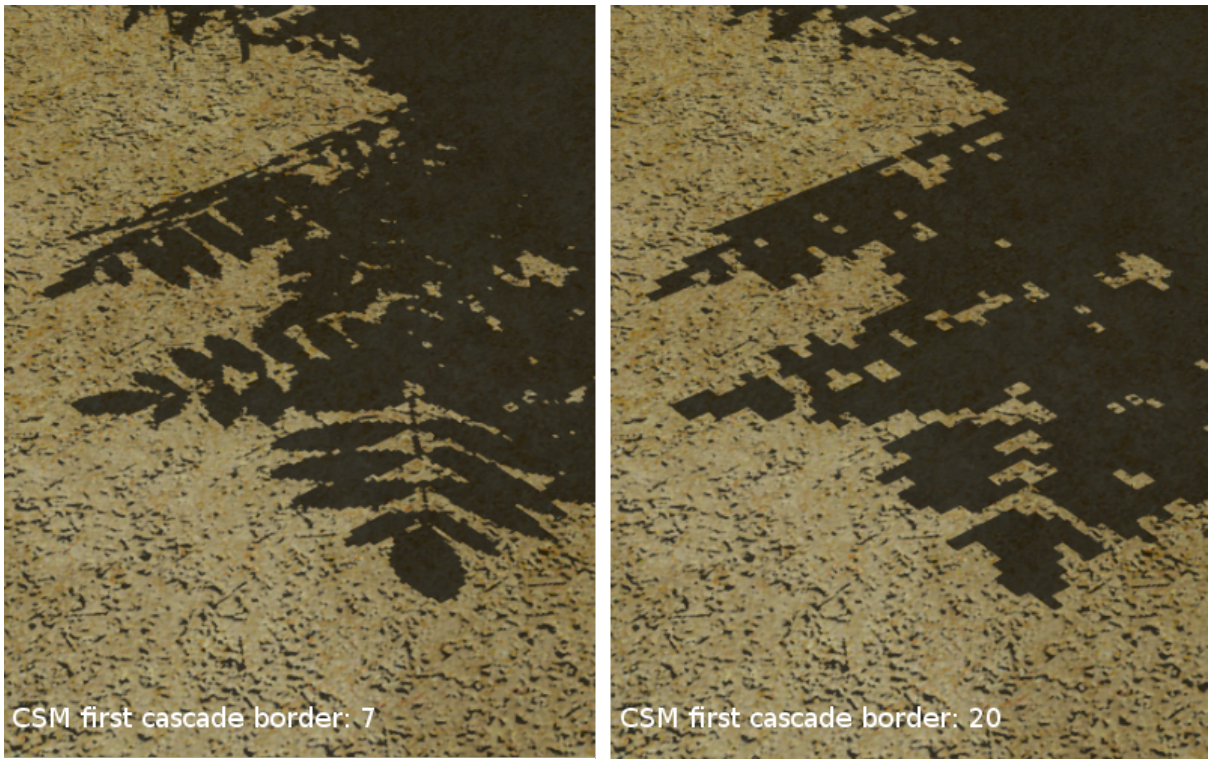
CSM number Number of shadow cascades. From 1 to 4 cascades are supported. The default value is 1.

CSM first cascade border First cascade size. The default value is 10.0.

CSM last cascade border Last cascade size. The default value is 100.0.

The sizes of the intermediate cascades are interpolated between the two above-mentioned parameters.

Note: When setting up the shadows keep in mind that the bigger the cascade size is, the worse and less detailed are the shadows inside it. On the other hand, reducing the CSM first cascade border parameter makes the subsequent less detailed cascades closer to the camera and thus more noticeable. Reducing the CSM last cascade border parameter forces shadows to disappear at more close distance from the camera. However when softened shadows are used the overall quality will improve thanks to blurring at the edges.



CSM first cascade blur radius Blur ratio for the first cascade. The default value is 3. Zero value produces hard shadows.

CSM last cascade blur radius Blur ratio for the last cascade. The default value is 1.5. Zero value produces hard shadows.

The blur radii of the intermediate cascades are interpolated between the two above-mentioned parameters.

Note: We recommend to start setting up the softened shadows with the first cascade (using CSM first cascade blur radius) and then proceed to other cascades (using CSM last cascade blur radius). Often the last cascade may need less blurring than the first one. This may be needed to prevent the shadows on the last cascade being too faded due to low resolution. This also reduces undesirable self-shadowing artifacts.

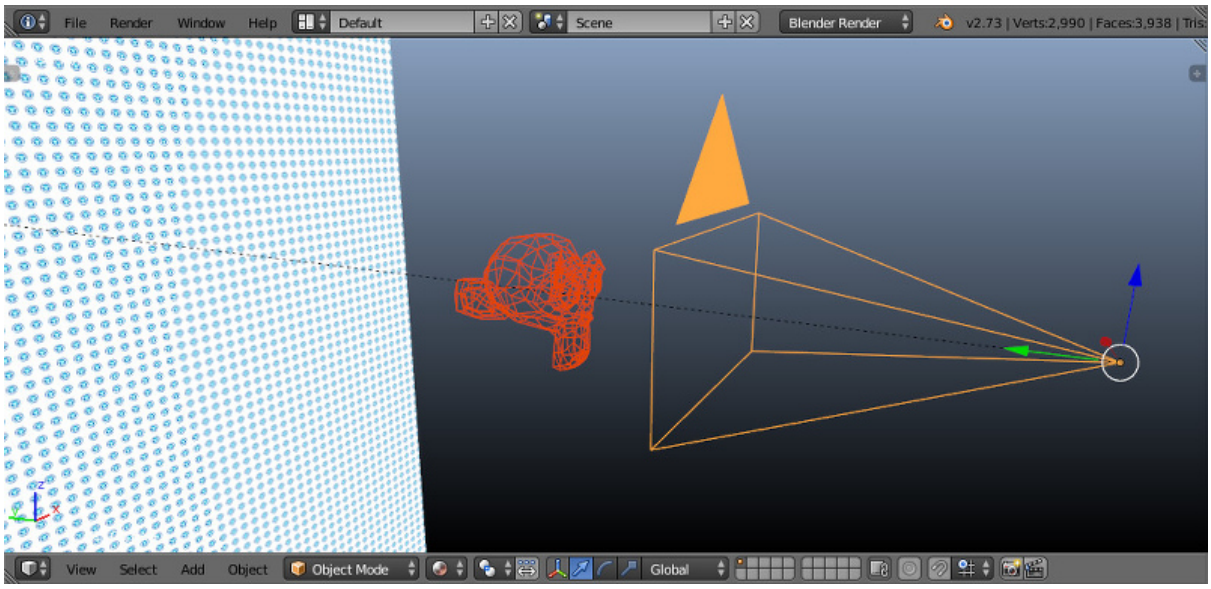
Fade-out last cascade Smooth dying-out of the last cascade. Enabled by default.

Blend between cascades Smoothing the boundaries between the cascades. Enabled by default.

14.6 Background

You can change the background in the following ways:

1. Enable Sky Settings > Render Sky, then set the Horizon Color and the Zenith Color under Blender's World tab.
2. 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.
3. Place a surface with a material and an optional texture in front of the camera. Parent it to the camera. If required, tweak the distance to this surface, starting and ending clipping planes for the camera.



4. Use a skydome.
5. Set up the procedurally generated atmosphere.
6. Set the engine's background_color parameter with the config.set() method. Please note, that Sky Settings > Render Sky under Blender's World tab must be disabled. 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");

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

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

7. 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). Please note, that Sky Settings > Render Sky under Blender's World tab must be disabled. 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 15

Postprocessing Effects

15.1 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.

15.1.1 Activation

Activate the Motion Blur panel on the Render tab.

15.1.2 Additional settings

Factor Effect appearance ratio. The higher this value is the stronger is the 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.

15.2 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.

15.2.1 Activation

1. Select an active camera and go to its settings panel Camera (Object Data).
2. Then two options are available:
   - Select an object to use as the camera's focus in the Focus option 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 same 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.

15.2.2 Additional settings

Front 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.

Rear 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.

Power Blurring ratio. The default value is 3.0.

15.3 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.

15.3.1 Activation

Activate the Ambient Occlusion SSAO panel under the Render tab and set the Render Shadows parameter to AUTO or ON on the Rander > Shadows panel.

15.3.2 Additional settings

Radius Increase The spherical sampling radius multiply factor when transfering from the internal sampling ring to the external one. The default value is 3.0.

Use Hemisphere Use a hemispherical sampling for shading instead of a spherical. Besides it uses different shading law.

Use Blur Depth Test Use edge-preserving blur to SSAO if flag will be turned on. Otherwise it uses blur which averages a 4x4 rectangle around each pixel.

Blur Depth Test Discard Value Influence of depth difference between samples on blur weight. It uses with Use Blur Depth Test activated flag. The default value is 1.0.

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.

15.4 God Rays

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

15.4.1 Activation

Activate the God Rays panel under the Render tab.

15.4.2 Additional settings

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.

15.5 Bloom

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

15.5.1 Activation

Activate the Bloom panel under the Render tab.

15.5.2 Additional settings

Key Bloom intensity.

Blur Bloom blurriness factor.

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

15.6 Outlining

As a result of the outline glow effect, a luminous colored halo will be displayed around the object.

15.6.1 Activation

The outlining is activated programmatically via API. Different animation models can be applied such as constant glow, fading out glow, pulsatory glow and any other. In order to enable the outlining effect on a certain object, make sure that the Render > Object Outlining panel's Enable property is set to ON or AUTO.

15.6.2 Additional settings

On the Object > Selection and Outlining panel:

Enable Outlining Permit using the outline glow effect on this object.

Duration Duration of glow animation, seconds. The default value is 1.

Period Repeat period of glow animation, seconds. The default value is 1.

Relapses The number of iterations of glow animation. If zero, animation is repeated forever. The default value is 0.

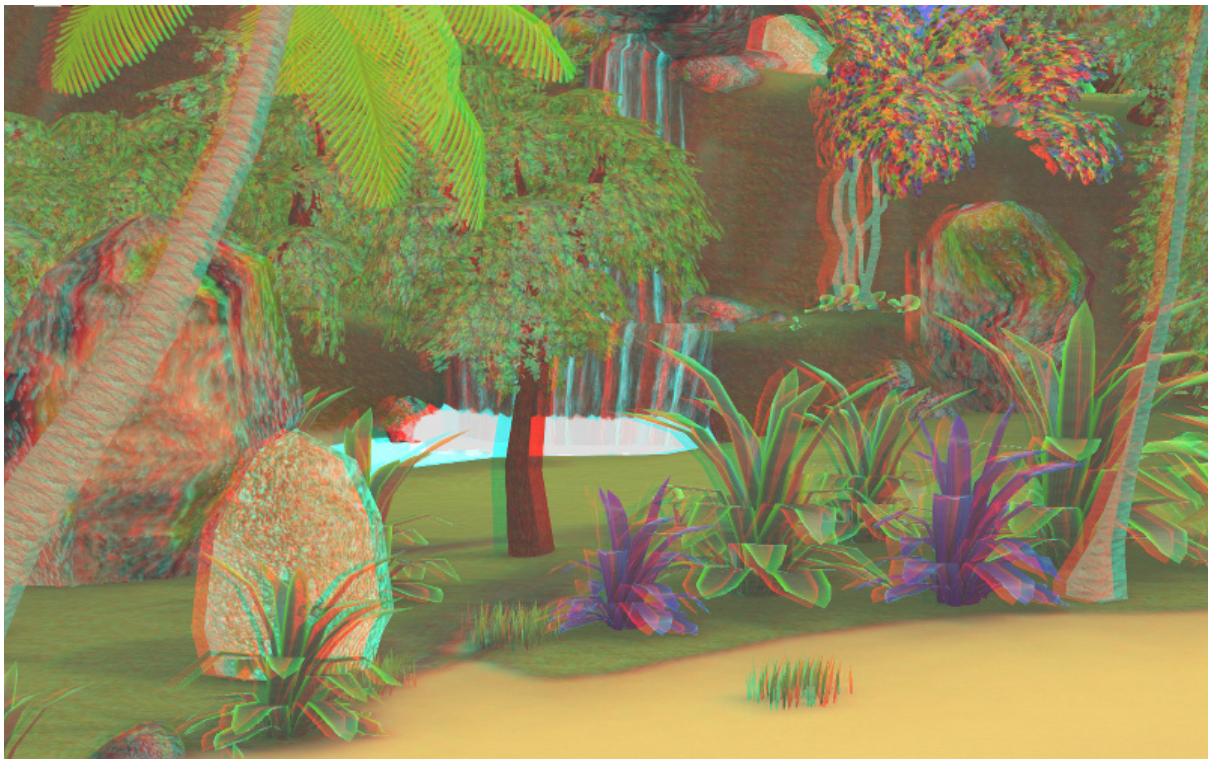
Outline on Select Activate glow animation upon selecting the object. In this case the Selectable option must be enabled. In case of a user-defined glow animation model, this option must be disabled in order to avoid conflict.

On the Render > Object Outlining panel:

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

The Render > Object Outlining settings are taken as default when the glow effect is initiated via API.

15.7 Stereoscopic Rendering (Anaglyph)



15.7.1 Activation

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

15.7.2 Additional settings

No.

15.8 Color Correction

15.8.1 Activation

Activate the Color Correction panel under the Render tab.

15.8.2 Additional settings

Brightness The default value is 0.0.

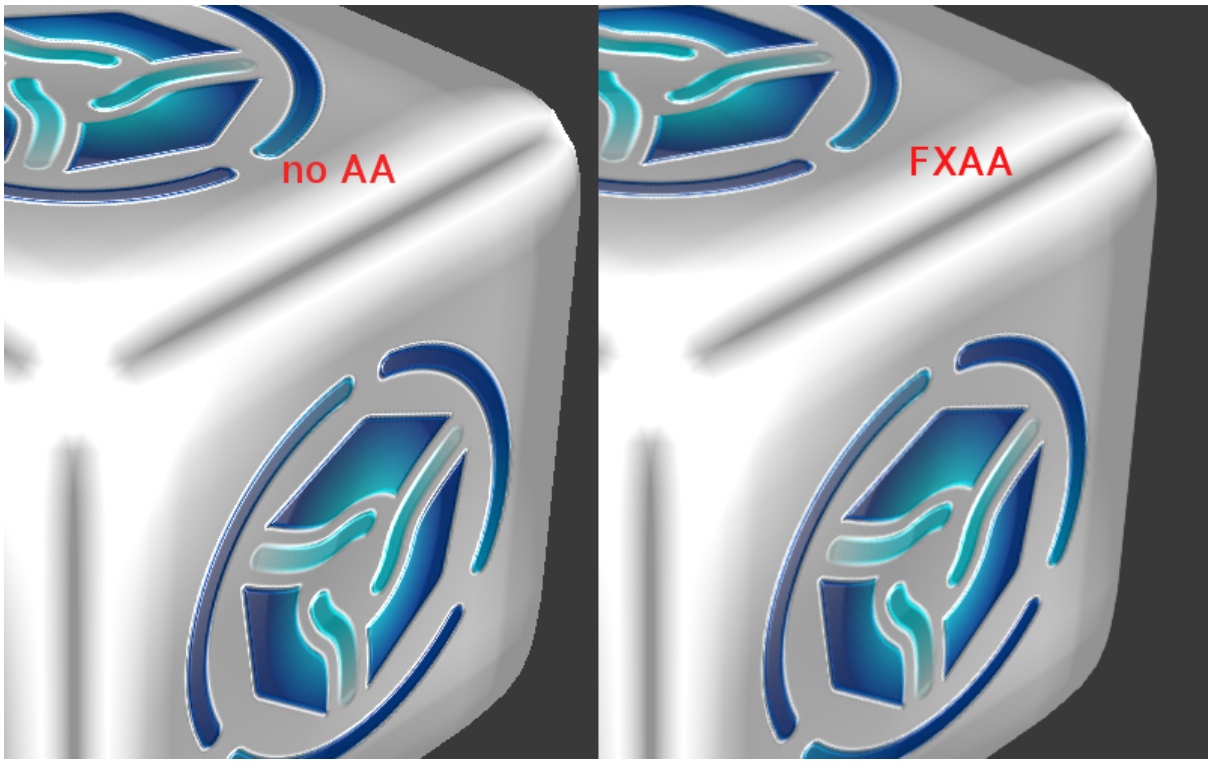
Contrast The default value is 0.0.

Exposure The default value is 1.0.

Saturation The default value is 1.0.

15.9 Anti-Aliasing

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



15.9.1 Activation

Activate the Enable Antialiasing option on the Render > Antialiasing panel.

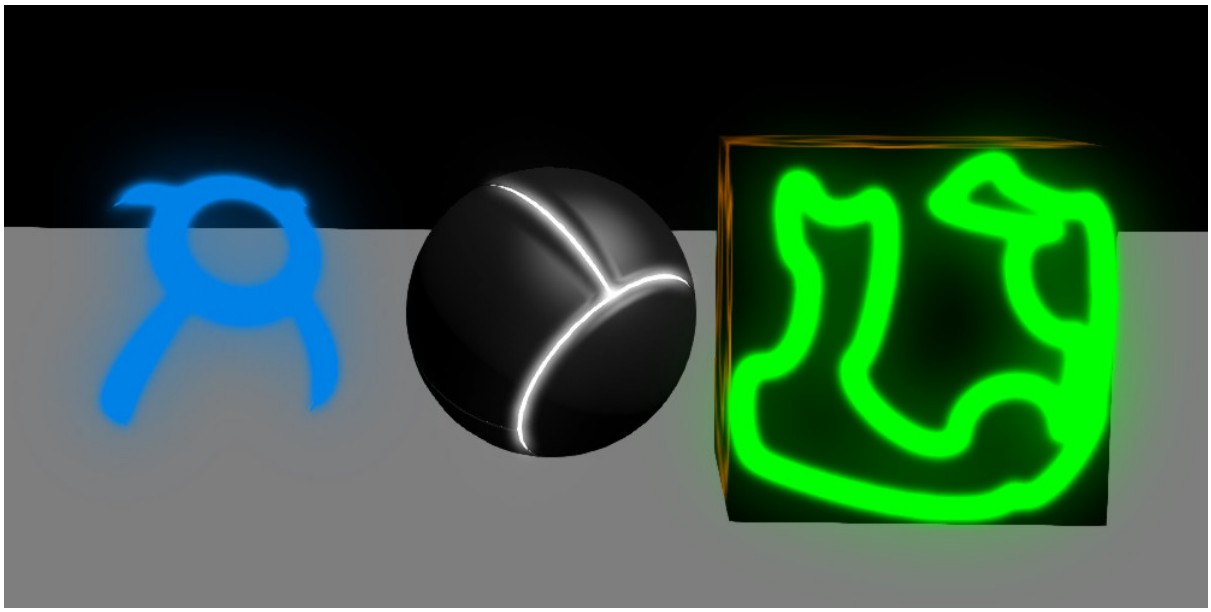
15.9.2 Additional settings

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

15.10 Glow

Effect of halo which is observed around emissive objects due to light scattering in the atmosphere and inside the human eye.



15.10.1 Activation

Add a B4W_GLOW_OUTPUT node to a node material. The Enable Glow Materials option on the Render > Glow Materials panel should be set to ON or AUTO.

15.10.2 Additional settings

Small Mask: Intensity Intensity of glow obtained through the smaller mask. The default value is 2.0.

Small Mask: Width Width of glow obtained through the smaller mask. The default value is 2.0.

Large Mask: Intensity Intensity of glow obtained through the larger mask. The default value is 2.0.

Large Mask: Width Width of glow obtained through the larger mask. The default value is 6.0.

Render Glow Over Transparent Object Render the glow effect over transparent objects.

CHAPTER 16

Particle System. Fluids

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.

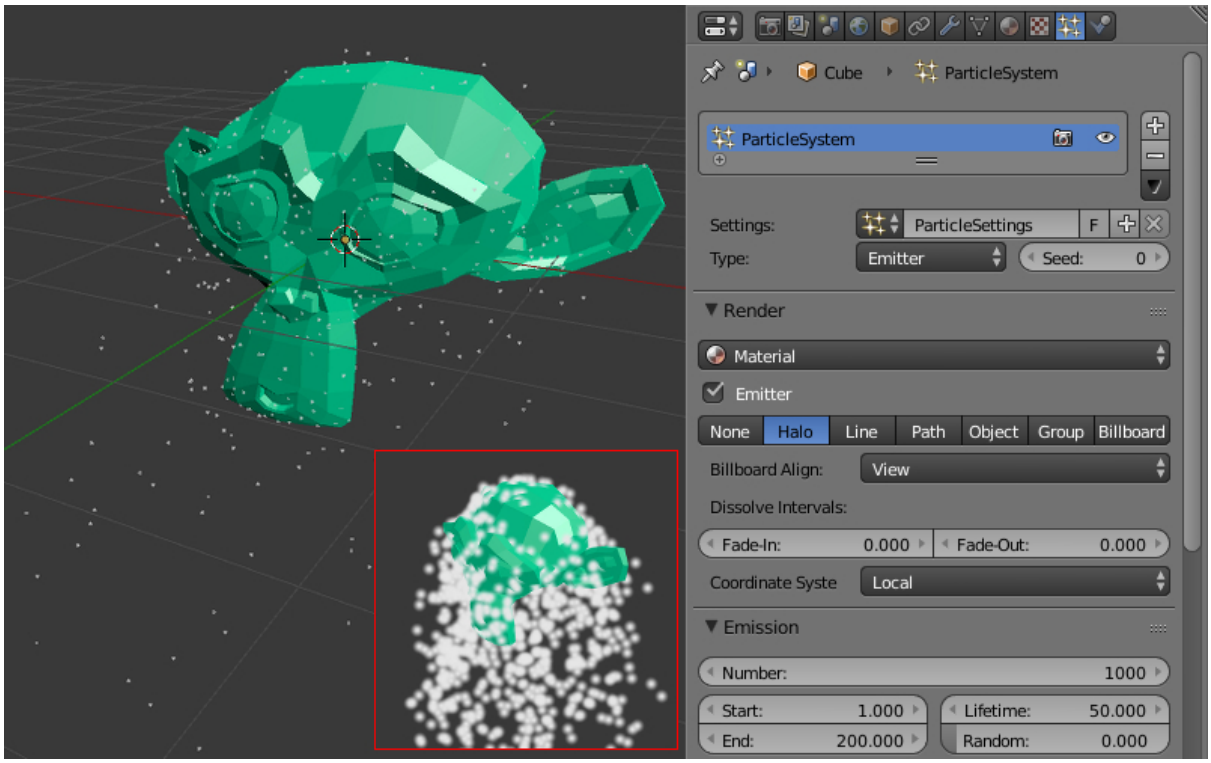
16.1 Use

16.1.1 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 Emission > Cyclic emission option for the particle system.
   - "non-cyclic animation" - enable the Animation > Apply Default Animation option for the emitter.

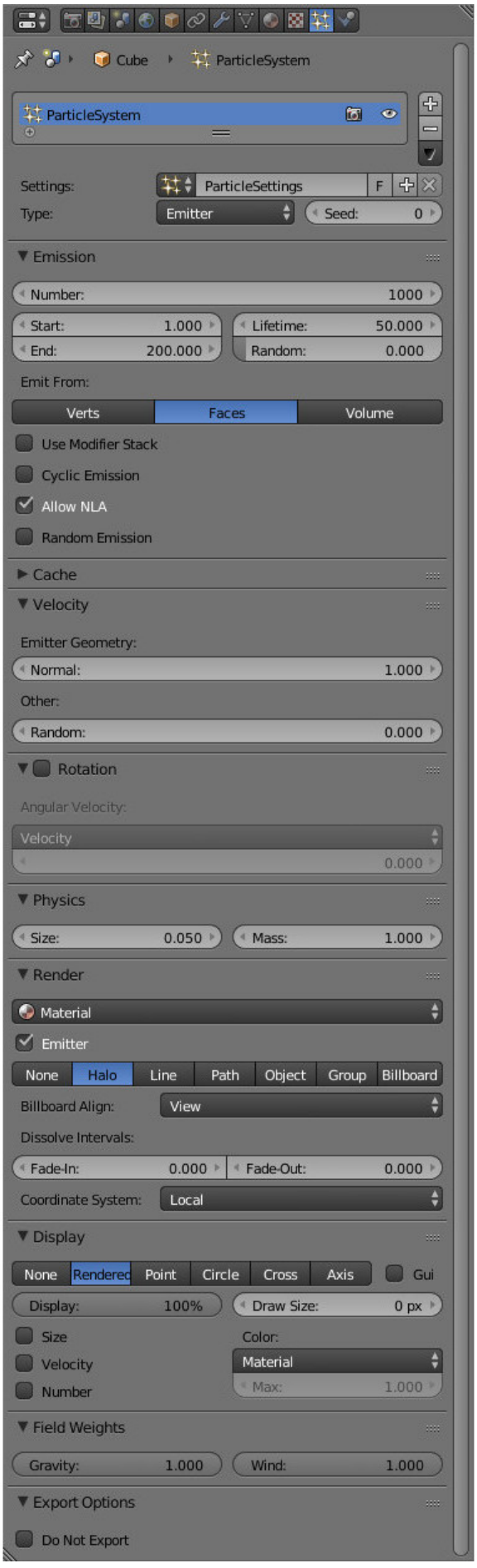
16.1.2 Recommended additional settings

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.



16.2 Setup

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


Cube
ParticleSystem
ParticleSystem
Settings:
ParticleSettings
F
Type:
Emitter
Seed:
0
Emission
Number:
1000
Start:
1.000
End:
200.000
Lifetime:
50.000
Random:
0.000
Emit From:
Verts
Faces
Volume
Use Modifier Stack
Cyclic Emission
Allow NLA
Random Emission
Cache
Velocity
Emitter Geometry:
Normal:
1.000
Other:
Random:
0.000
Rotation
Angular Velocity:
Velocity
0.000
Physics
Size:
0.050
Mass:
1.000
Render
Material
Emitter
None
Halo
Line
Path
Object
Group
Billboard
Billboard Align:
View
Dissolve Intervals:
Fade-In:
0.000
Fade-Out:
0.000
Coordinate System:
Local
Display
None
Rendered
Point
Circle
Cross
Axis
Gui
Display:
100%
Draw Size:
0 px
Size
Velocity
Number
Color:
Material
Max:
1.000
Field Weights
Gravity:
1.000
Wind:
1.000
Export Options
Do Not Export

16.2.1 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.

16.2.2 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.

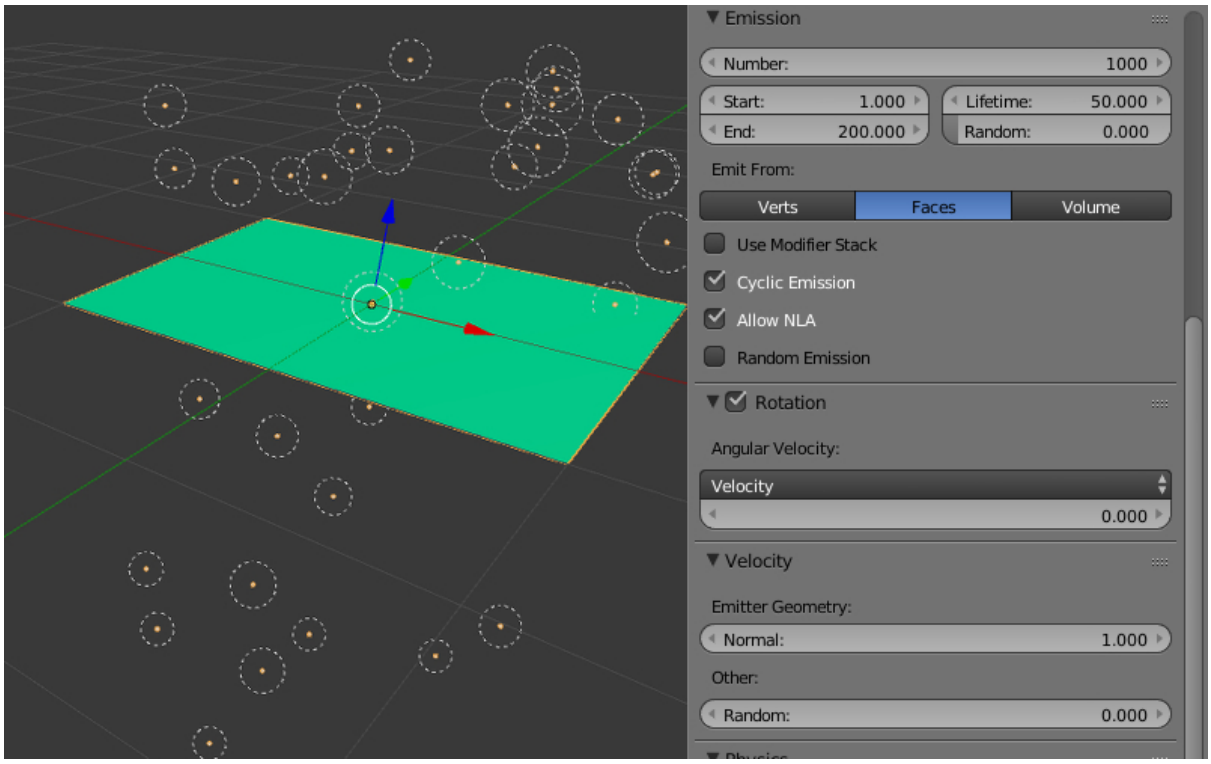
Emission > Lifetime > Random The random factor for the life time. The default value is 0.0.

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

Emission > 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.

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

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.



16.2.3 Direction settings

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.

16.2.4 Rotation settings

Rotation > Angular Velocity > Mode Mode for particle billboards self-rotating. Hair particle system supports all the types of rotation and Emitter supports only Velocity.

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

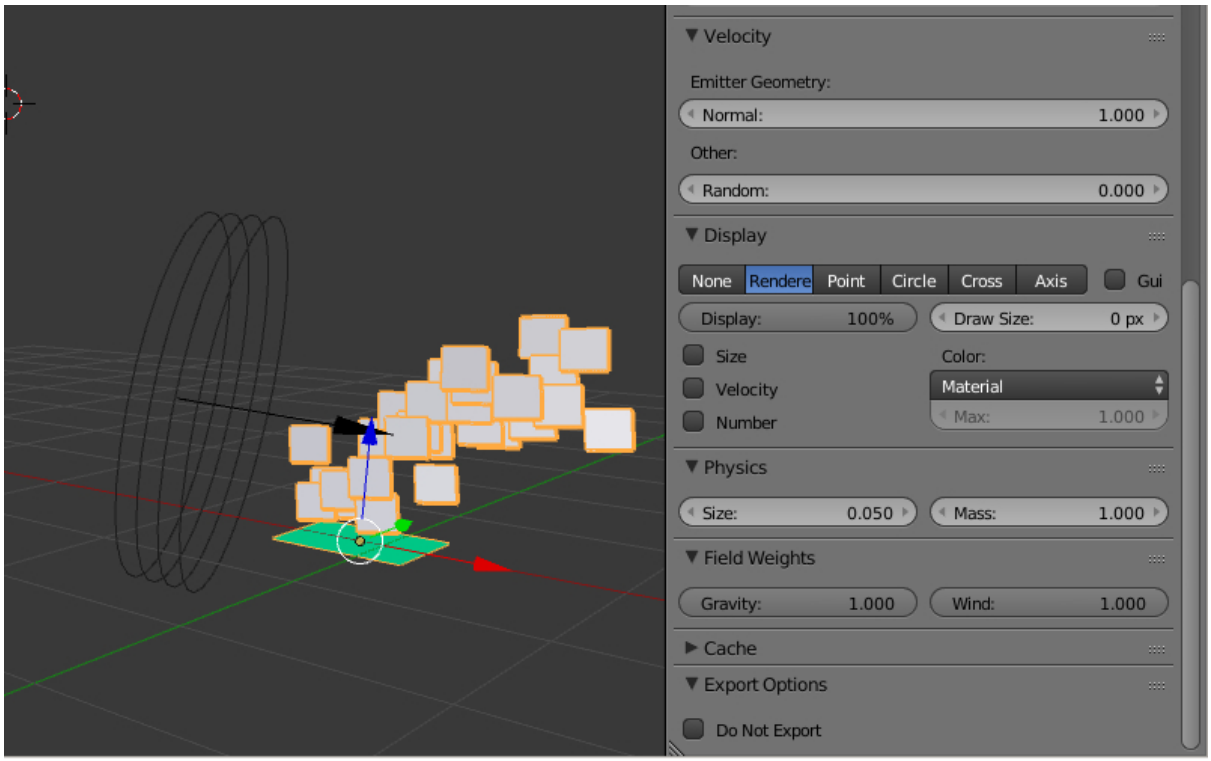
16.2.5 Physics settings

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.



16.2.6 Rendering settings

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.

Render > 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.

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

Render > Coordinate system Coordinate system of emitting particles: Local - use local coordinate system of the emitter object, World - use world coordinate system.

16.2.7 Supported settings for force fields influence

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.

16.2.8 Engine specific settings

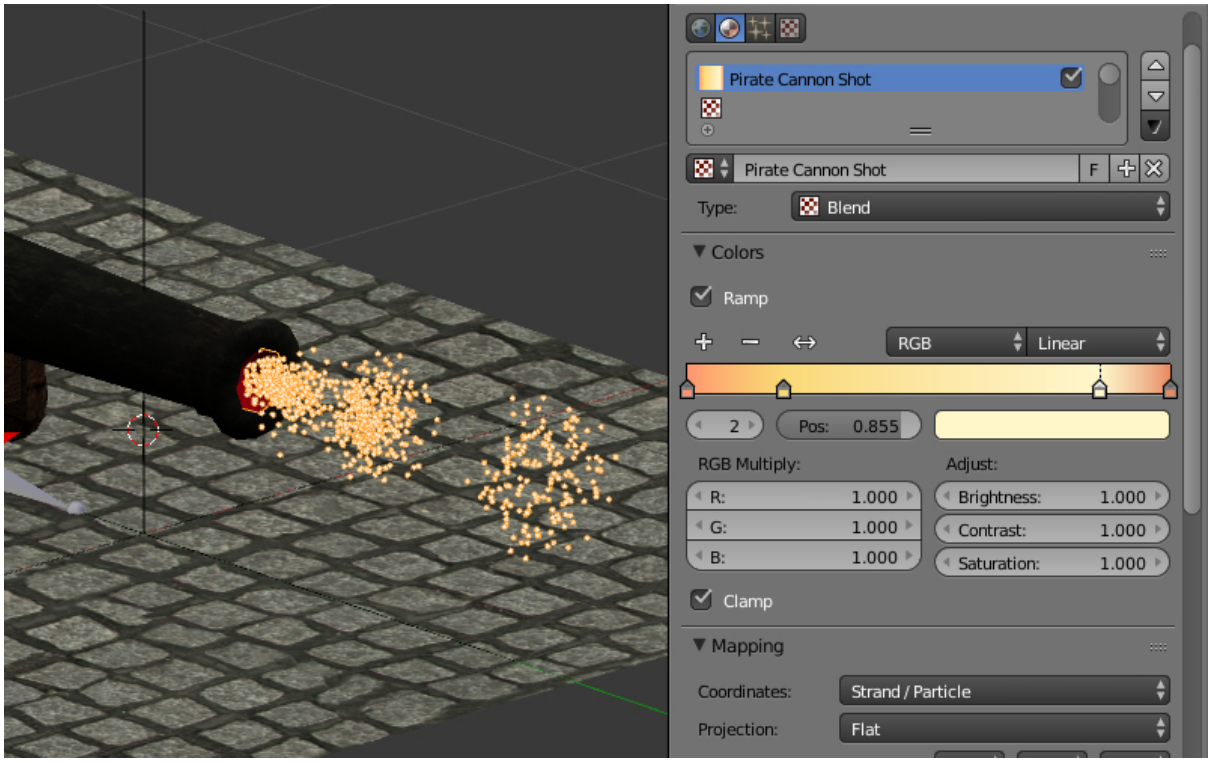
Export Options > Do not export Don't export.

16.3 Textures in Particle Systems

16.3.1 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.

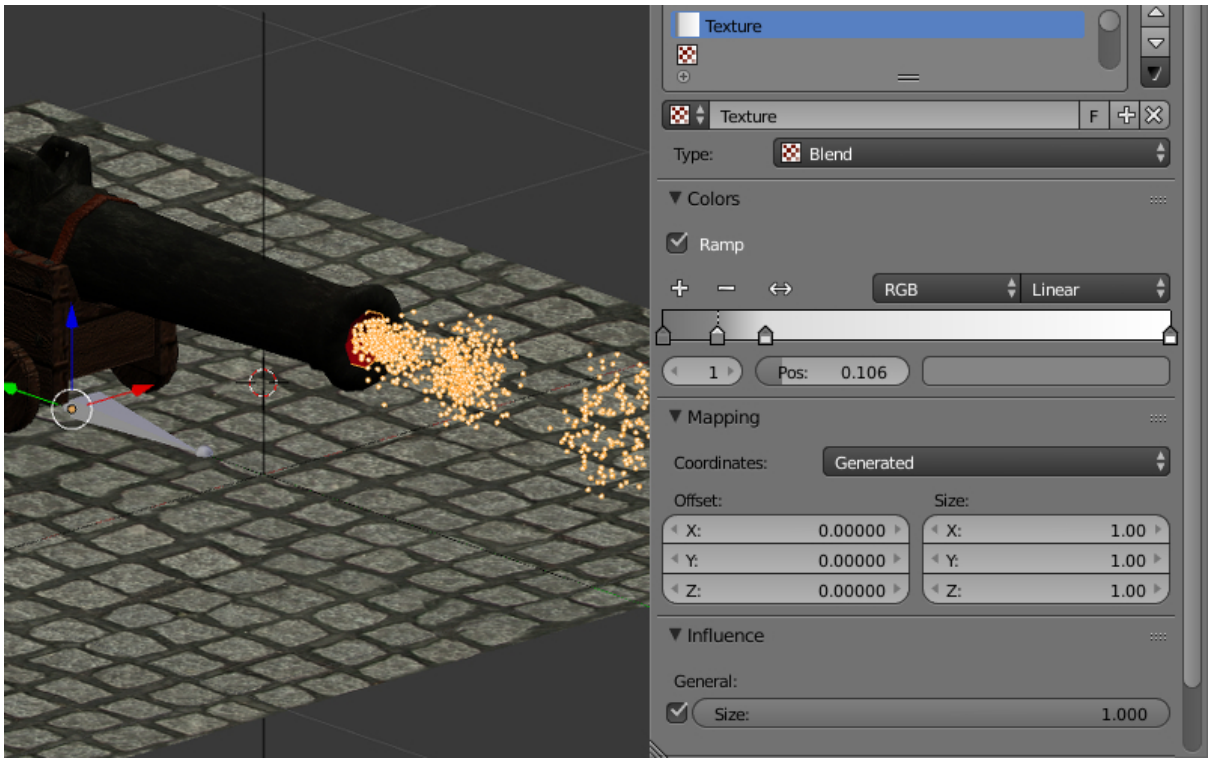


16.3.2 Textures of particle systems

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 17

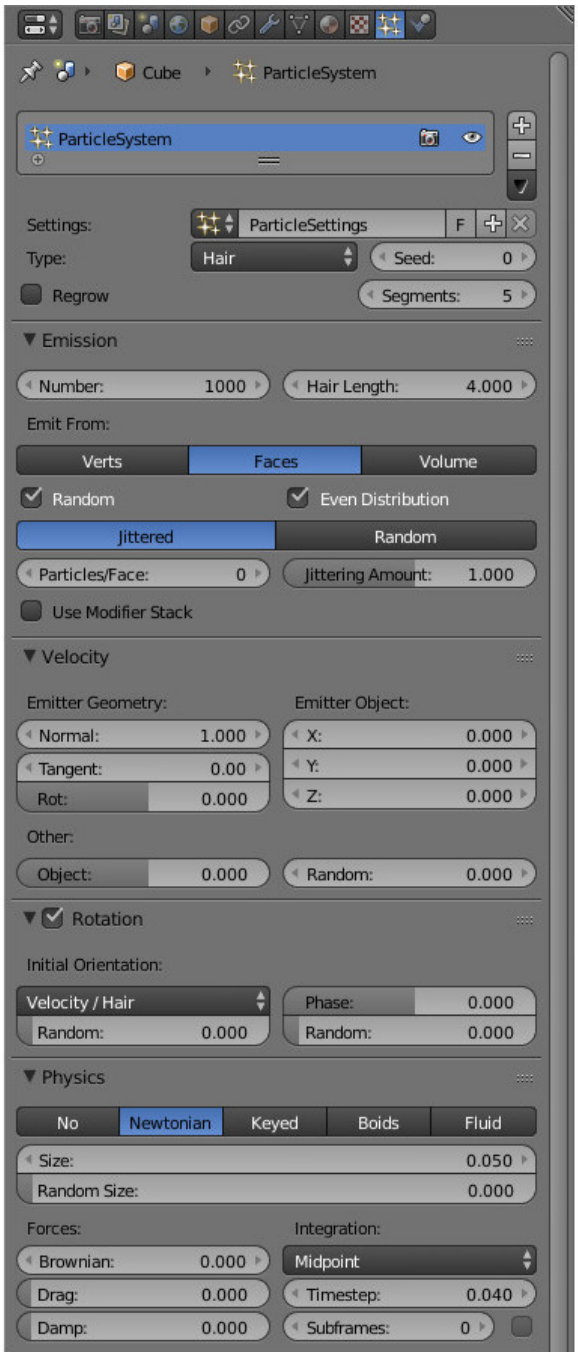
Particle System. Instancing

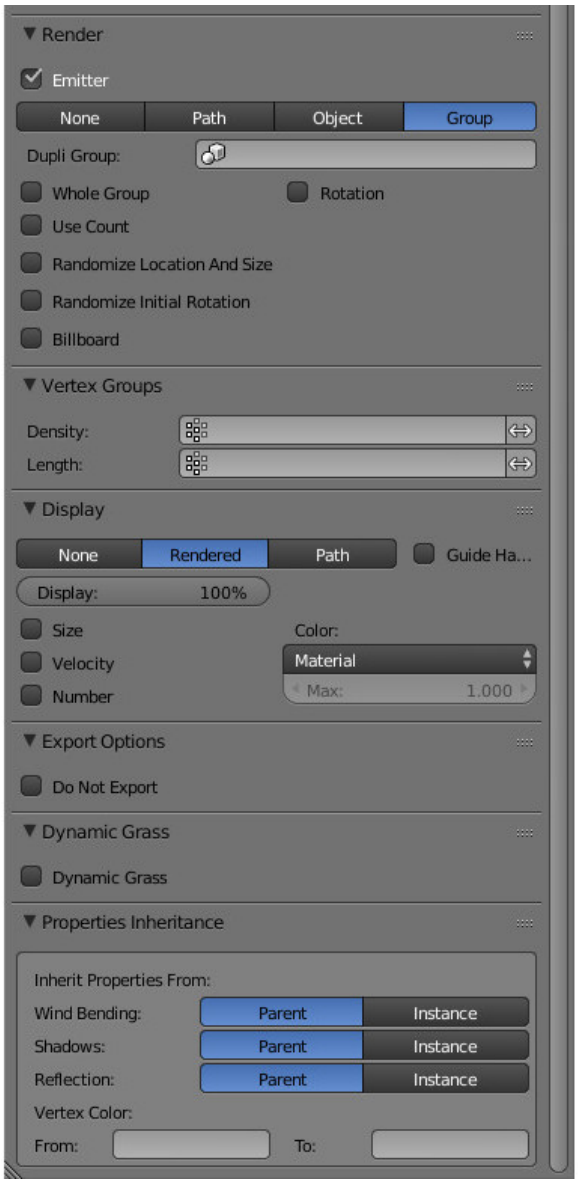
A particle system can be used to create multiple object copies (so called instancing). This technique simplifies scene authoring and also reduces loading time and memory consumption as compared to the using of single objects.

Using particle systems for instancing has some limitations though:

- Movement and animation of objects inside a particle system is not allowed.
- Parenting is not possible among the objects inside a particle system, except for dupli-groups.
- Instancing of non-mesh objects is not possible.

17.1 Particle System Setup



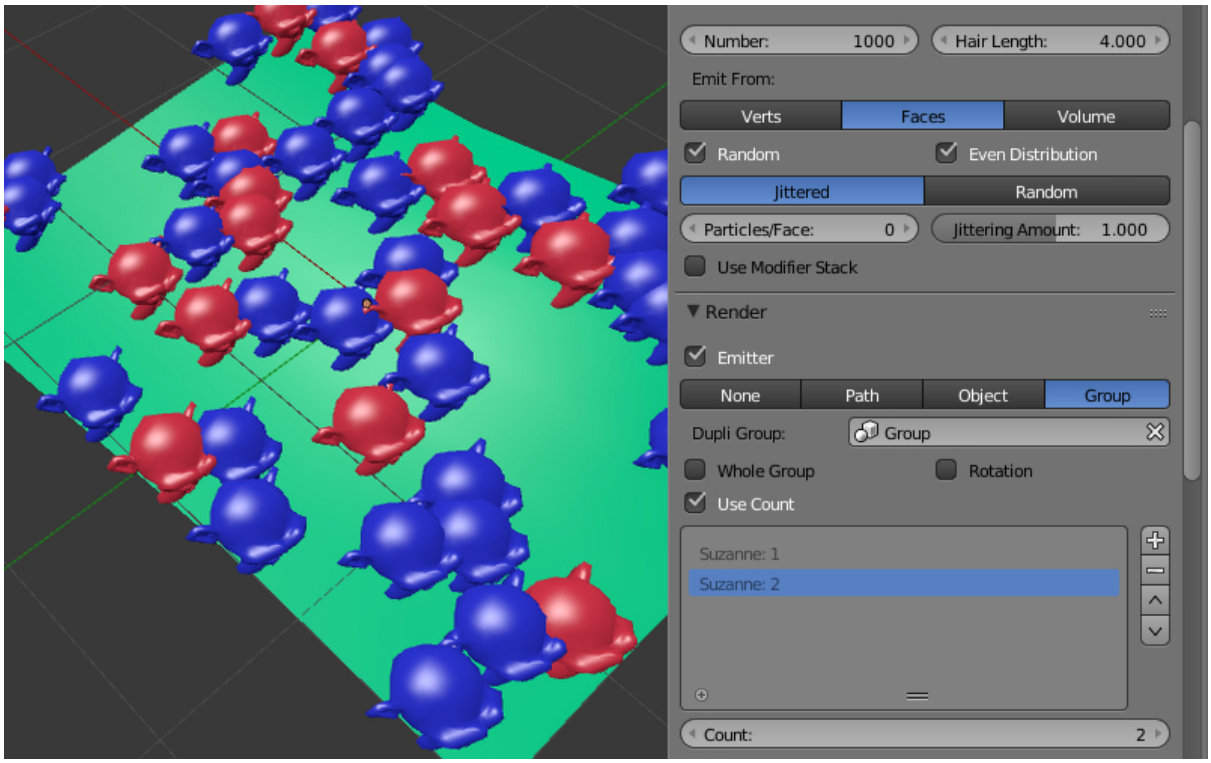


17.1.1 Activation

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. In order to display correct sizes in the viewport, set the Emission > Hair Length and Render > Size parameters to 1.0.



17.1.2 Display settings

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 becomes visible. The engine does not reproduce the exact positions of objects of certain types.

Render > Randomize 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 disabled, the exact coordinates and sizes of the particle objects are exported and used.

Render > Randomize Initial Rotation

This option randomizes rotation of the objects relative to the axis defined by Rotation Type. If enabled, the engine generates random rotation angles for the particle objects. If disabled, the rotation is taken from the Rotation panel.

Render > Rotation Type

An Axis of random object rotation (the property is available when Render > Randomize Initial Rotation is enabled). There are two options:

- 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

Render > Rotation Strength

Coefficient which defines the range of random rotation angles - counting from the direction towards the camera. Available when the Render > Randomize Initial Rotation checkbox is enabled. Examples:

- 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.

Render > Billboard

Enables billboarding for particles. Disabled by default.

Render > Billboard Type

Billboarding type. The option is available when the Render > Billboard option is enabled. Three types are available:

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

The default is Basic.

Render > Jitter Amplitude

Coefficient which defines the particle oscillation amplitude. Available when the Jittered type is selected from the Render > Billboard Type menu. The bigger this parameter is, the bigger is the oscillation amplitude. The default value is 0.

Render > Jitter Frequency

Particle oscillation frequency in hertz. Available when the Jittered type is selected from the Render > Billboard Type menu. The default value is 0.

Render > Billboard Geometry

Billboard rotation type (the option is available when the Render > Billboard checkbox is set). Two types are available:

- 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.

17.1.3 Dynamic grass setup

Dynamic Grass

The option enables the dynamic grass rendering mode. Disabled by default.

Dynamic Grass > Scale Threshold

Minimum size for dynamic grass particles. Smaller particles will not be rendered. The option is available if Dynamic Grass is enabled.

17.1.4 Inheritance settings

Properties Inheritance > 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.

Properties Inheritance > Shadows

Inheriting the shadow settings by particles:

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

The default is Parent.

Properties Inheritance > Reflection

Inheriting the reflection settings by particles:

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

The default is Parent.

Properties Inheritance > Vertex color

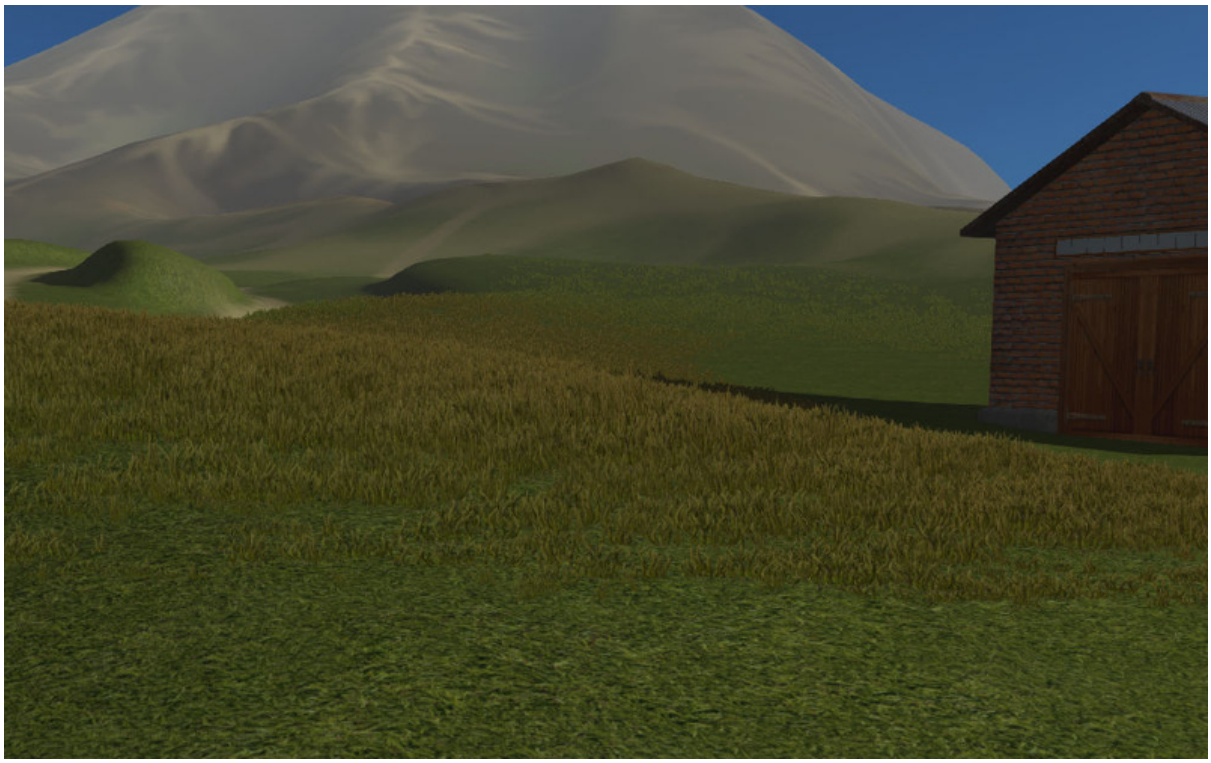
Inheriting the vertex color from the emitter. Contains two fields:

- From - the emitter's existing vertex color name
- To - the particle's existing vertex color name

There is no inheritance by default.

17.2 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 Dynamic Grass option.
2. Enable the Terrain Dynamic Grass option for the supposed landscape material.

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).

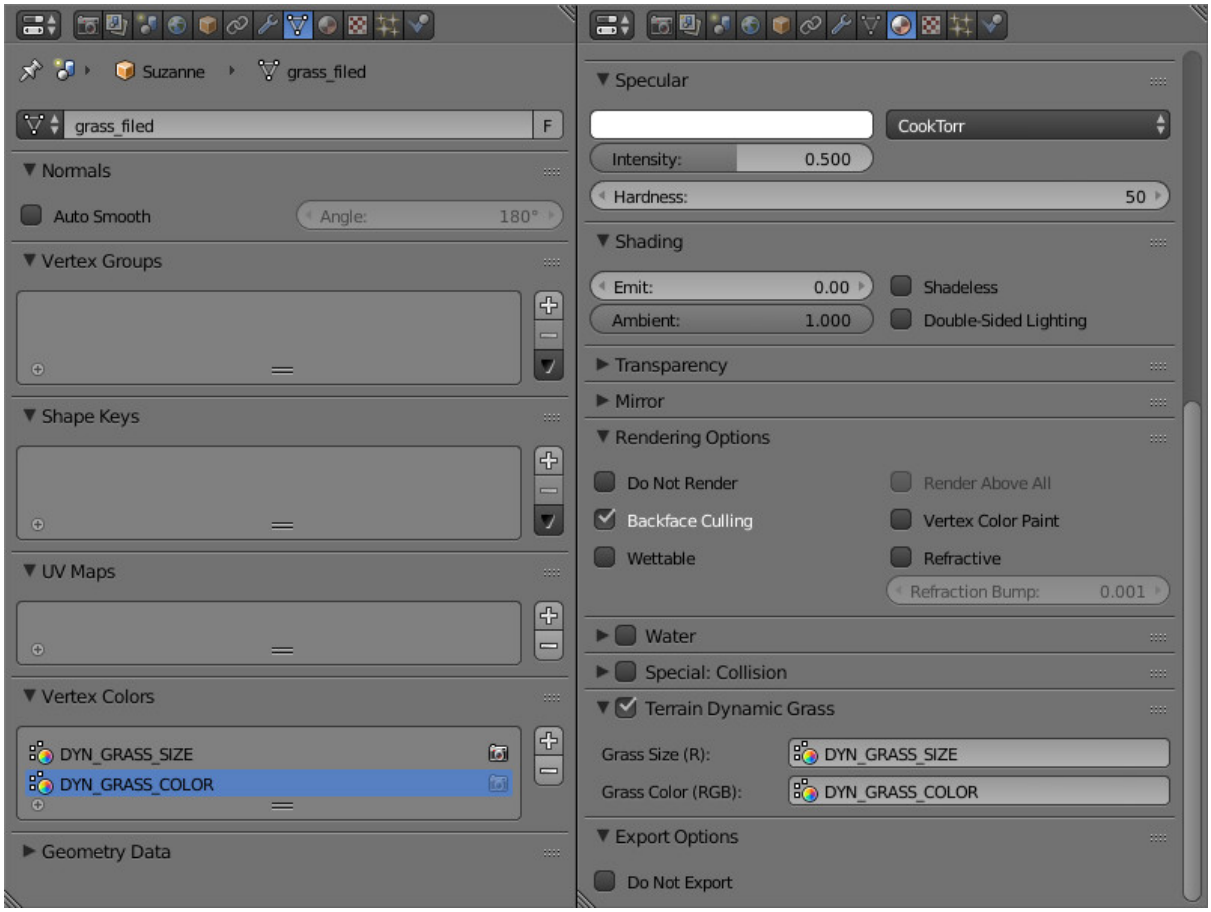
For the landscape's material, the following text fields become active when the Terrain Dynamic Grass option is enabled:

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).



17.3 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 can 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 Initial Random Rotation checkbox is enabled, it is recommended to select the vertical axis for random rotation - Z axis (by using the Rotation Type menu). The Rotation Strength value can be set at will.

2. Billboarding settings

It is recommended to enable billboarding, to set its type as Jittered (by using the Render > Billboard Type menu) and to make it spherical - Spherical (by using the Render

> Billboard Geometry menu). The Render > Jitter Amplitude and Render > Jitter Frequency values can be set at will.

3. Particle position settings

It is recommended to select the Verts value from the Emission > Emit From menu, and to select the emitter's vertex group (in the Vertex Group > Density field) which defines the positions of particles. Note, that the Render > Randomize Location and Size checkbox should be disabled.

4. Wind effect settings

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

5. Vertex color inheritance settings

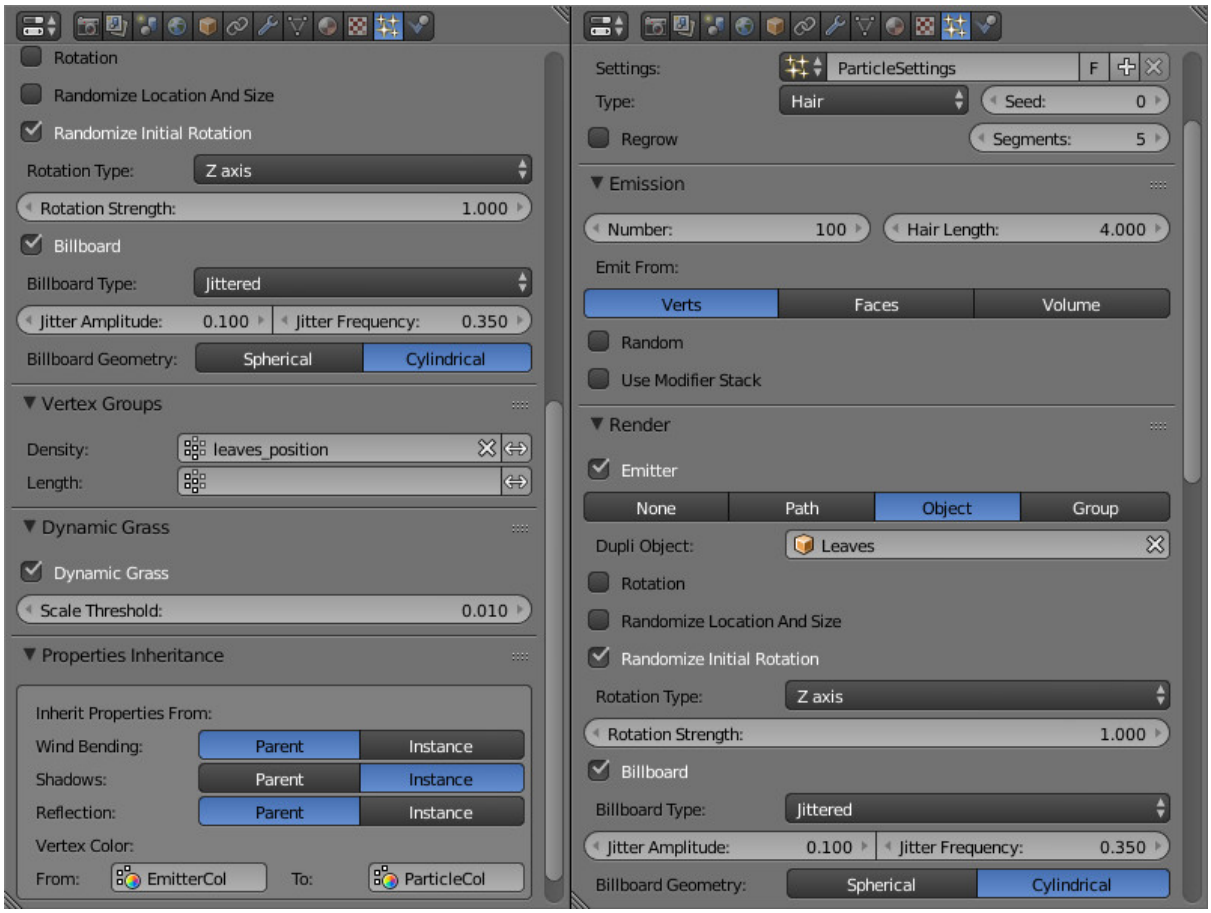
For the emitter's vertex color to be inherited by the particles, it is required to specify both the emitter's vertex color name and the particle's vertex color name in the Properties Inheritance > Vertex Color > From and Properties Inheritance > Vertex Color > To fields respectively. As a result, the color of the emitter's vertex that is closest to the particle (specified in the From field) will be copied and propagated into the particle's To vertex color layer.

The resulting vertex color layer with the name specified in the Properties Inheritance > Vertex Color > To field can be used in the particle's node material for its tinting and for any other effects.

6. Setting up the size of particles via vertex group weights

In order to create dependency between the size of particles and vertex group weights, select the name of the desired vertex group in the Vertex groups > Length field.

The influence can be tweaked by setting weights in the selected vertex group.


Rotation
Randomize Location And Size
Randomize Initial Rotation
Rotation Type: Z axis
Rotation Strength: 1.000
Billboard
Billboard Type: Jittered
Jitter Amplitude: 0.100
Jitter Frequency: 0.350
Billboard Geometry: Spherical Cylindrical
Vertex Groups
Density: leaves_position
Length:
Dynamic Grass
Dynamic Grass
Scale Threshold: 0.010
Properties Inheritance
Inherit Properties From:
Wind Bending: Parent Instance
Shadows: Parent Instance
Reflection: Parent Instance
Vertex Color:
From: EmitterCol
To: ParticleCol
Settings: ParticleSettings F
Type: Hair
Seed: 0
Regrow
Segments: 5
Emission
Number: 100
Hair Length: 4.000
Emit From:
Verts Faces Volume
Random
Use Modifier Stack
Render
Emitter
None Path Object Group
Dupli Object: Leaves
Rotation
Randomize Location And Size
Randomize Initial Rotation
Rotation Type: Z axis
Rotation Strength: 1.000
Billboard
Billboard Type: Jittered
Jitter Amplitude: 0.100
Jitter Frequency: 0.350
Billboard Geometry: Spherical Cylindrical

CHAPTER 18

Animation

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

- Object animation means the transformation of an object as a whole.
- 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.
- Animation of the Value node output in node materials.
- Wind bending - a procedural animation. Described separately.
- Particle emission. Described in the corresponding section.

18.1 Animation Control

There are two ways to control animation in the engine:

1. Automatically, activating the Animation panel and choosing the Behaviour parameter 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.

18.2 Object Animation

The parameters that can be animated are the center coordinates (Location), Rotation and Scale.

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) or XYZ Euler 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 a mesh object is animated it is required to activate the Force Dynamic Object option on the Rendering Propetries panel under the object properties tab.

18.3 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.

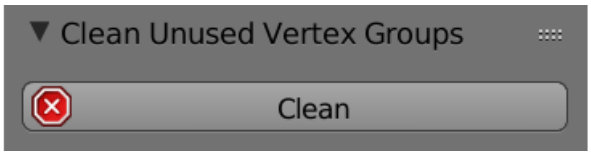
4. When inverse kinematics (IK) or other nontrivial structures are used, an additional step is required to bake the animations (Action datablocks in Blender). Baking can be performed using the Bake Skeletal Animation interface located on the Blend4Web panel.

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

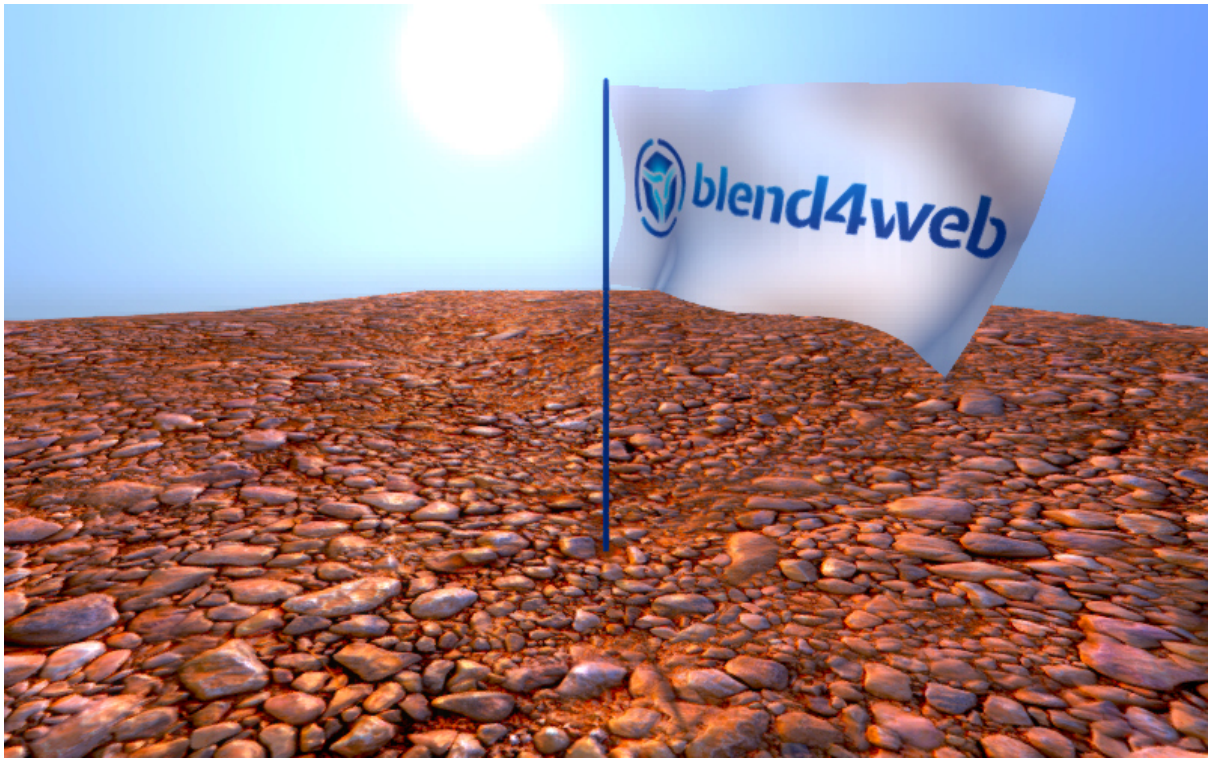
- A window with the list of actions being baked – bake only the actions which are listed, else - bake all the actions possible.
- Name – the current action name from the list of actions being baked.
- Optimize keyframes – optimize the animation keyframes after baking. In case of incorrect results it's recommended to turn this option off.
- 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.

Note: The engine supports up to 4 vertex groups per vertex. If the number of vertex groups exceeds 4 the vertex groups with the most influence are selected. When the scene is loaded the vertex weights are normalized i.e. their sum is reduced to 1.

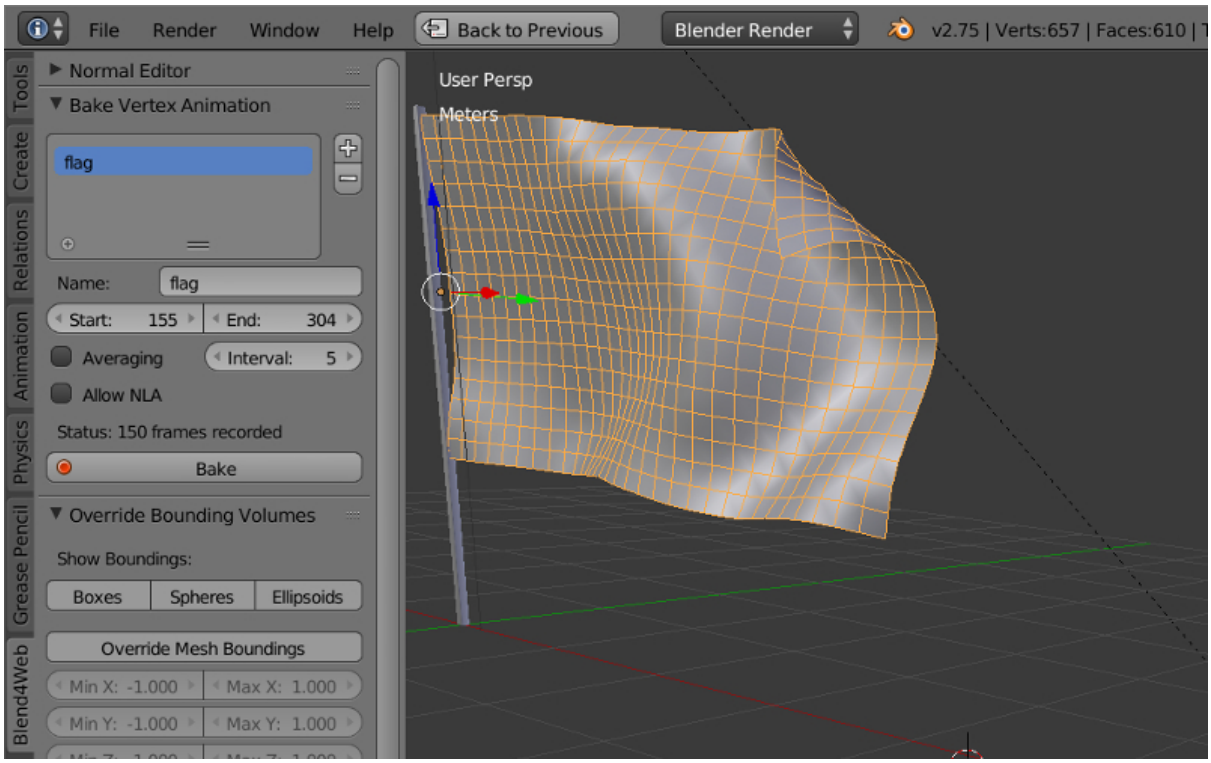
To remove vertex groups which are not used by armature, use button Remove Clean Unused Vertex Groups.



18.4 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 - Bake Vertex Animation - located on the Blend4Web tools panel.



18.5 Audio Source Parametrization

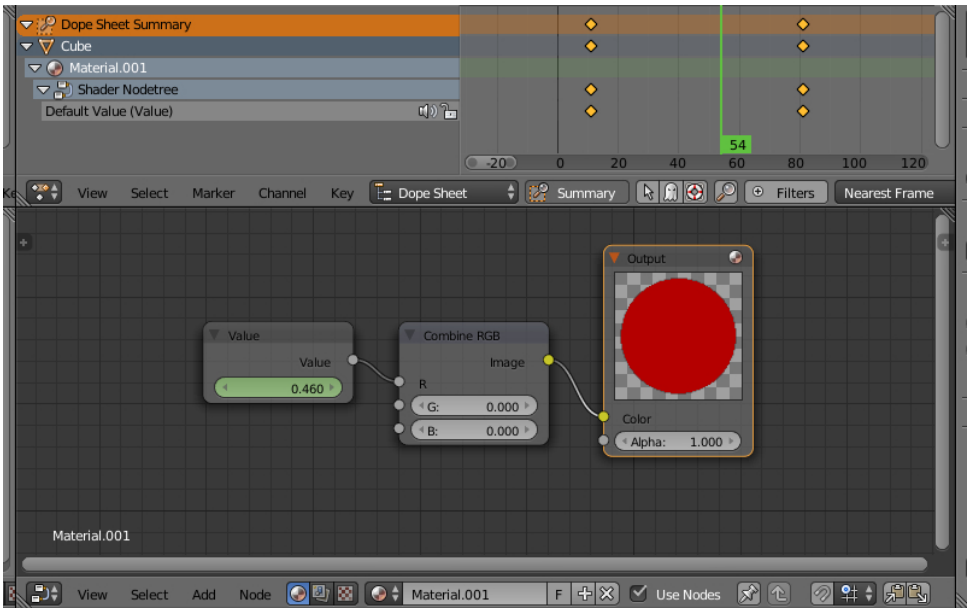
In addition the following animation key types are supported for the speaker objects:

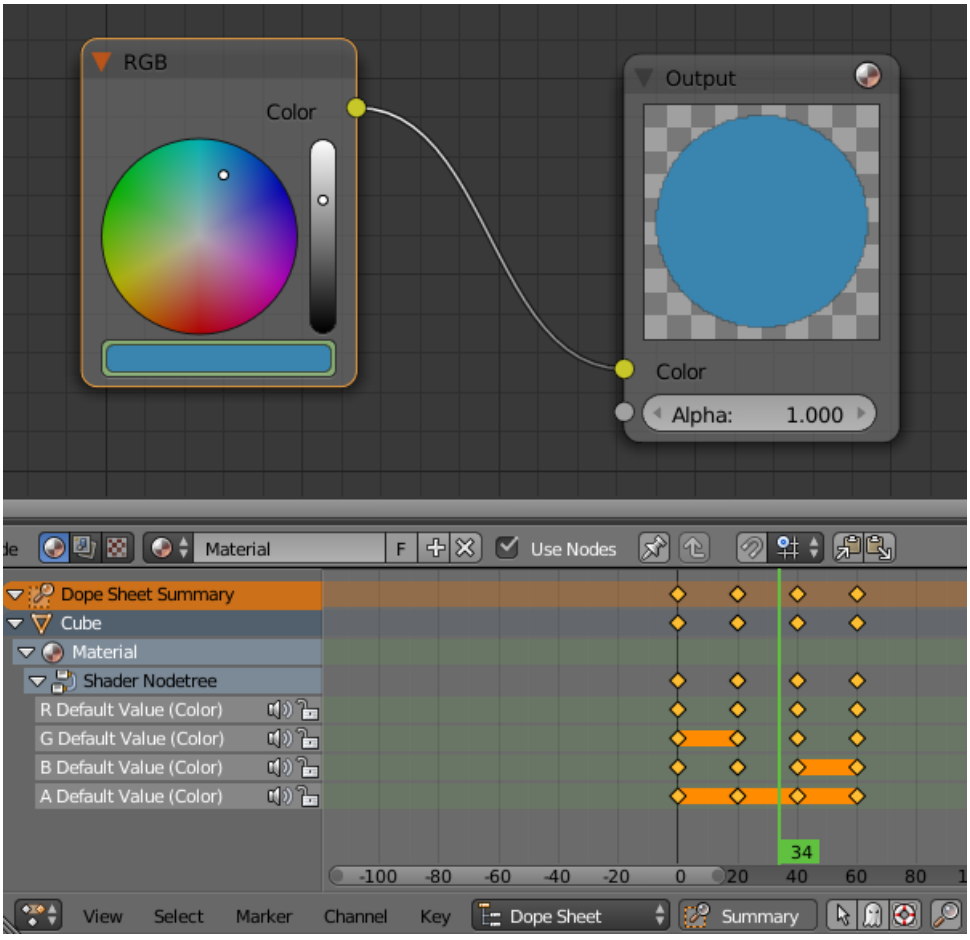
- Volume
- Pitch

Audio sources parametering in essence follows object animation.

18.6 Animation of Value and RGB Nodes

Playback of keyframes inserted on Value and RGB nodes is supported in node materials.


Dope Sheet Summary
Cube
Material.001
Shader Nodetree
Default Value (Value)
54
View Select Marker Channel Key Dope Sheet Summary Filters Nearest Frame
Output
Value
Value
0.460
Combine RGB
Image
R
G: 0.000
B: 0.000
Color
Alpha: 1.000
Material.001
View Select Add Node Material.001 F Use Nodes


RGB
Color
Output
Color
Alpha: 1.000
Material F Use Nodes
Dope Sheet Summary
Cube
Material
Shader Nodetree
R Default Value (Color)
G Default Value (Color)
B Default Value (Color)
A Default Value (Color)
34
-100 -80 -60 -40 -20 0 20 40 60 80
View Select Marker Channel Key Dope Sheet Summary

Note: Animation of numerical and color values on other nodes is not supported.

Can be also used to create tracks in the non-linear animation editor. Multiple animated Value or RGB nodes are supported per single material. Values of these nodes can be also modified via API, by using the set_nodemat_value and set_nodemat_rgb methods of the objects module.

See also:

B4W_TIME

CHAPTER 19

Outdoor Rendering

19.1 Water

19.1.1 Activation

For the supposed water material activate the Water panel option under the Material tab.

19.1.2 Basic Settings

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

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

19.1.3 Waves Dynamics

Ripples on the water are simulated by normal maps with animated UVs (from 0 up to 4 pieces). For normal map the only shared image is used - the textures differs only by the Mapping > Size parameters. The water mesh must have a UV layer.

19.1.4 Surface Wetting

Is carried out automatically. To turn the effect on activate the Wettable option on the needed materials Rendering Options panel.

19.1.5 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.

19.1.6 Shoreline Smoothing

The effect affects the water near the shoreline - it becomes more transparent.

Water > Shore Smoothing Enable smoothing.

Water > Absorb Factor Light absorption coefficient for the water. The higher it is the more transparent is the water.

19.1.7 Color Gradient

For color gradient the water material must have a texture with the Export Options > Shore Distance Map option enabled. This texture can be generated using the script for baking shoreline parameters

Shallow water > Color Shallow water color.

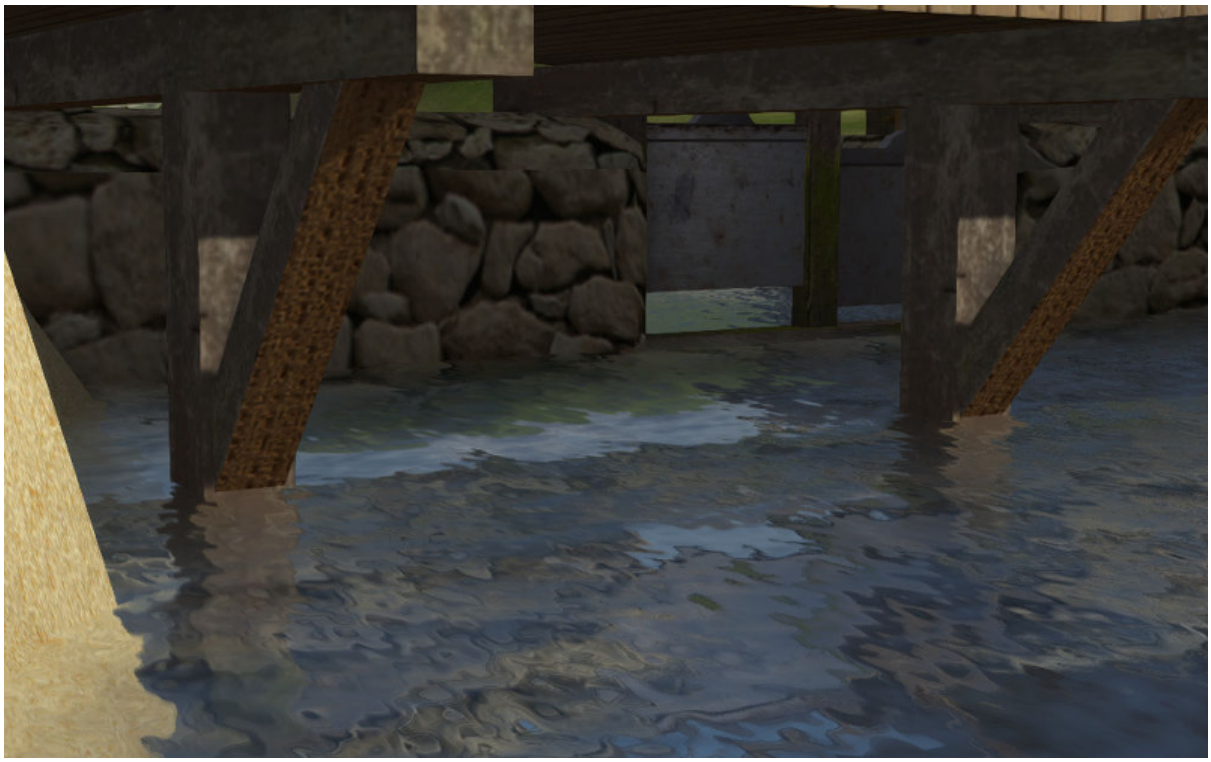
Shallow water > Factor Shallow water color mixing factor.

Shore water > Color Water color just at the shore line.

Shore water > Factor Factor for mixing water color just near the shoreline.

19.1.8 Refraction

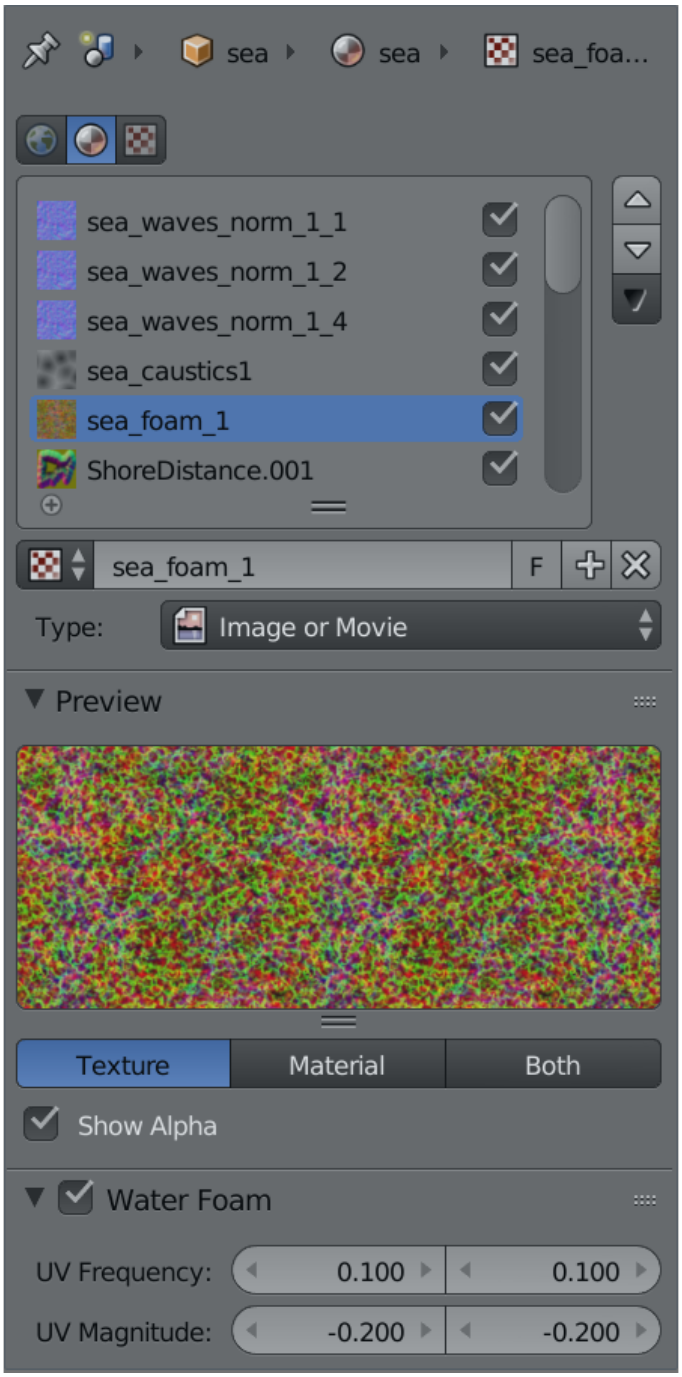
Under the Render tab in the Reflections and Refractions panel set the Refractions option to ON or AUTO.



19.1.9 Foam

Activation

For creating foam add a diffuse texture into the water material slot. Every RGB channel of this image must contain a BW foam texture in it. Then enable the Water Foam panel.



Setting up the Textures

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

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

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

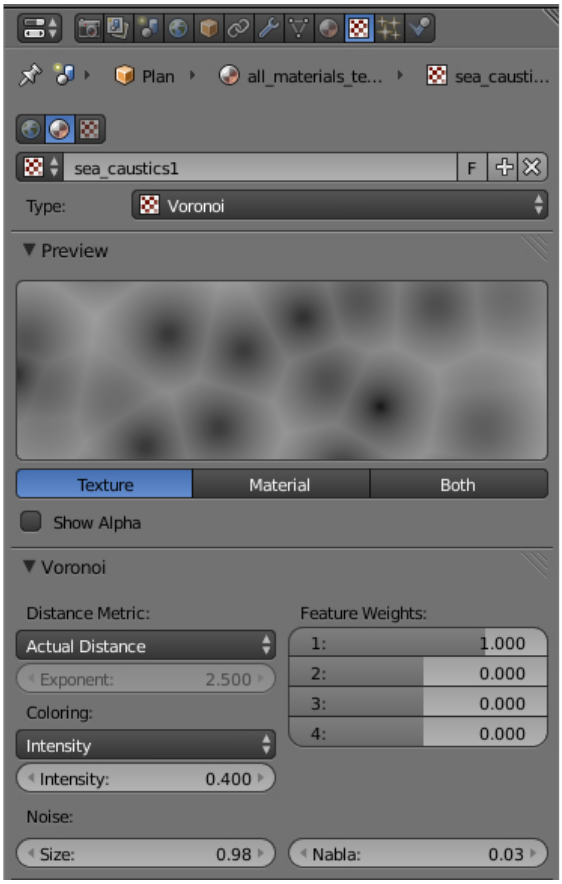
Setting up the Material

Foam > Factor General influence factor for the foam. The default value is 0.5.

19.1.10 Caustics and Chromatic Aberration

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

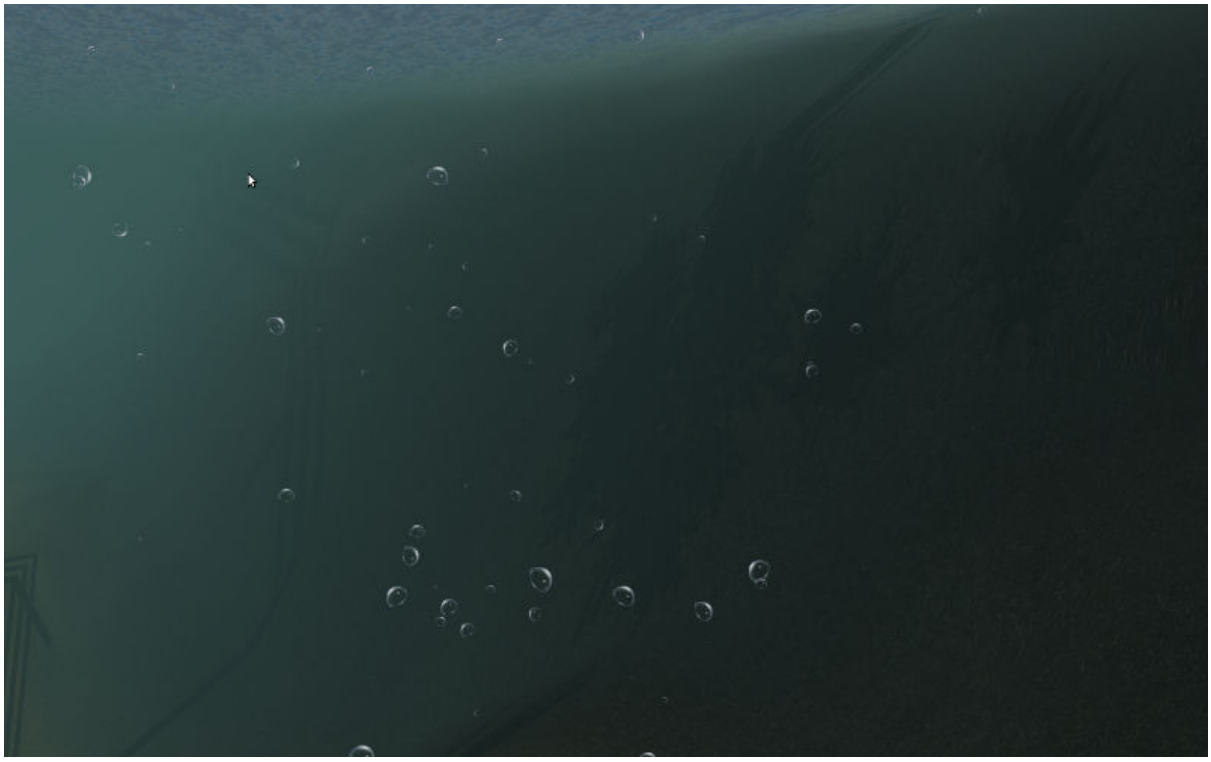
Setting up



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.

19.1.11 Underwater Environment



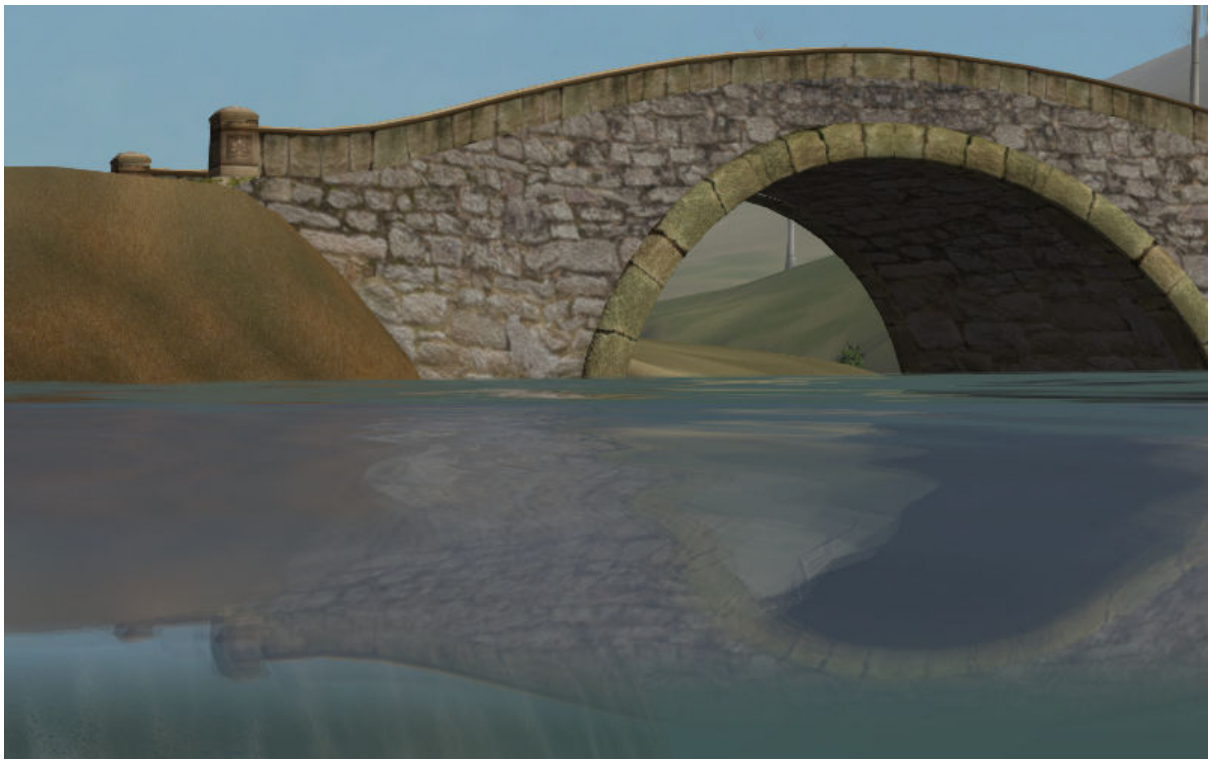
Visibility Settings ("fog")

Underwater Fog > Color Fog color. The default value is (0.4, 0.6, 0.7).

Underwater Fog > Density Exponential factor which affects the density and visibility distance. The default value is 0.06.

The god rays effect settings are also applied.

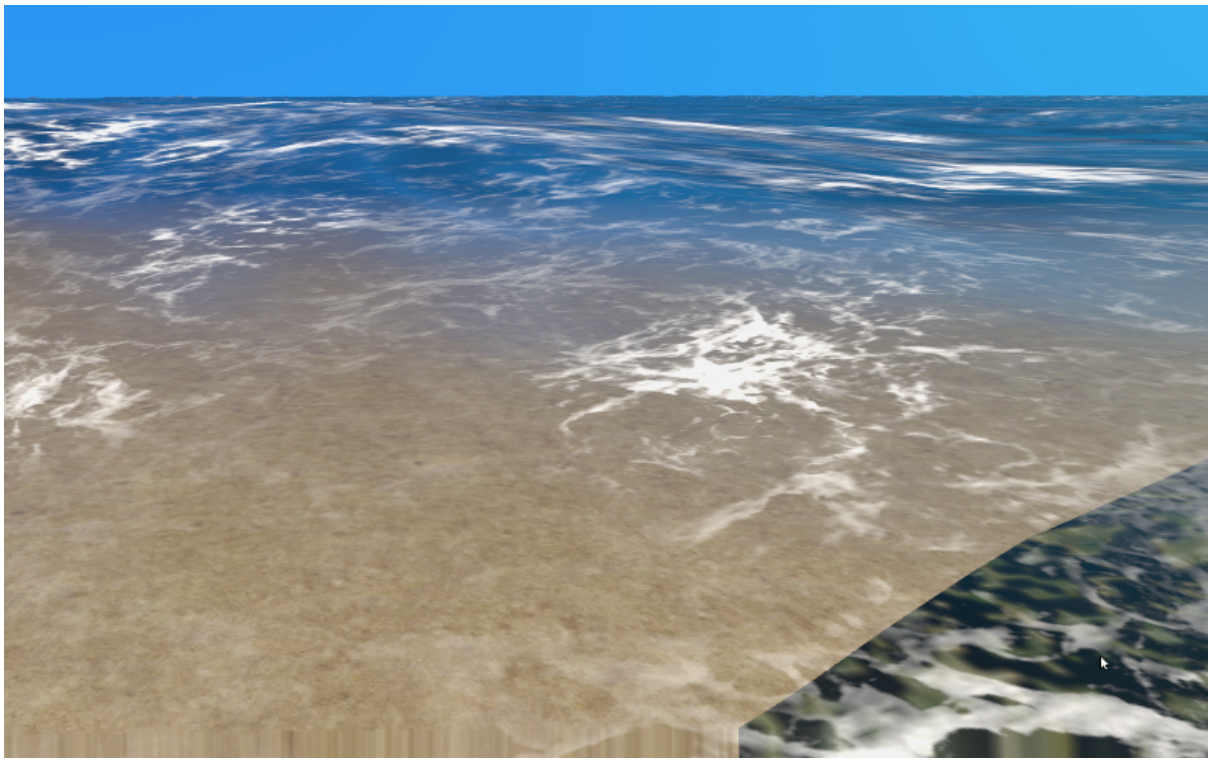
Note: The Rendering Options > Backface culling option must be turned off for the correct water surface rendering.



19.1.12 Volumetric Waves

Activation

To enable procedural waves the Waves option must be turned on.



Setting up

Wave Height Wave height. The default value is 0.0.

Wave Length Wave length. The default value is 10.0.

Noise Dist Scale 0 Size of the first component of the open water waves.

Noise Dist Scale 1 Size of the second component of the open water waves.

Noise Dist Freq 0 Frequency of the first component of the open water waves.

Noise Dist Freq 1 Frequency of the second component of the open water waves.

Min Dir Shore Fac Minimum height decrease coefficient of the shore waves.

Dir Frequency Frequency of the rolling of the shore waves.

Noise Dir Scale Noise size for the shore waves.

Noise Dir Freq Noise frequency for the shore waves.

Min Dir Noise Fac Noise minimum for the shore waves.

Min Dist Fac Minimum coefficient of mixing for open water waves.

Horizontal Factor Coefficient that shows how much the shore waves are shifted in the shoreline direction.

19.1.13 Settings for Surface Generation

Generate Mesh Enable generated surface.

Cascades Number Number of cascades in the generated surface.

Subdivisions Generated mesh subdivisions.

Detailed Distance Maximum distance from camera to the last cascades edge.

Baking Shoreline Data to Texture

On the tools panel (hotkey "T") under the Blend4Web tab open the Bake Shore Distance Map 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 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.

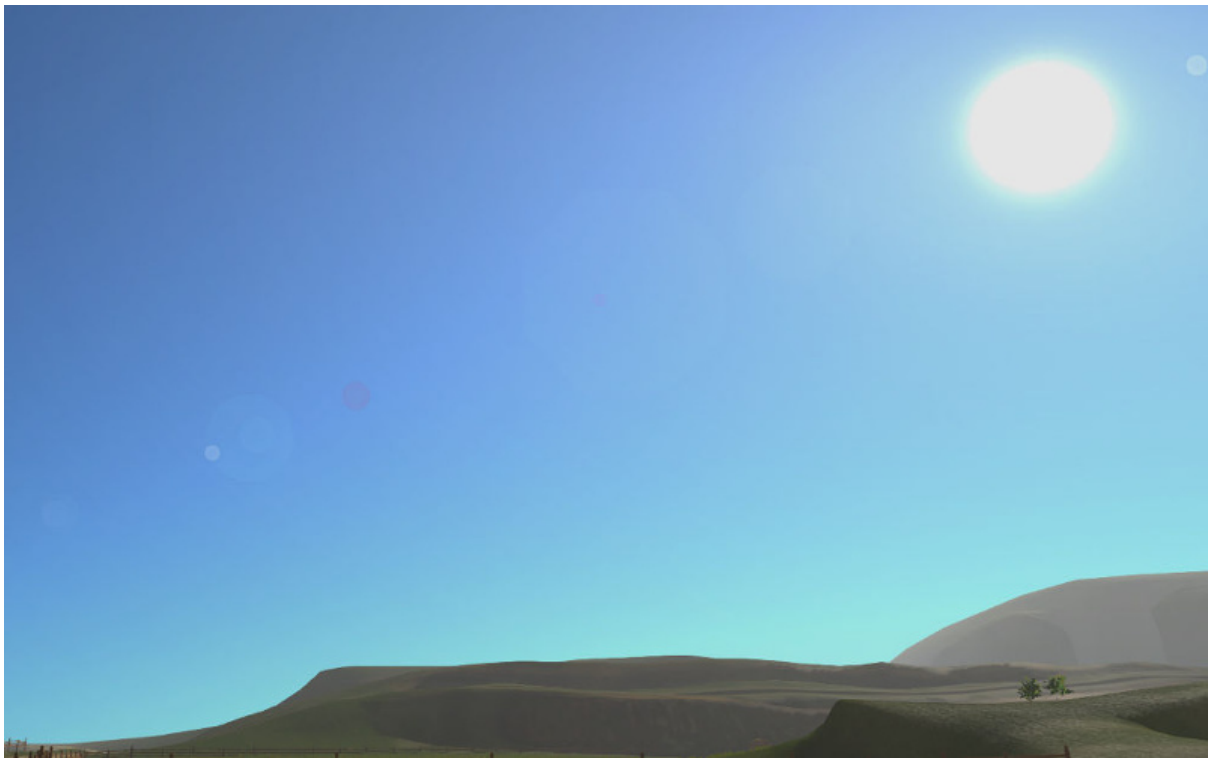
Upon script execution some system properties are saved in the water material. Therefore the scene must be saved after the script has finished working.

19.2 Atmosphere

19.2.1 Scattering

Enable World > Render Sky, then activate Procedural Sky panel under the World tab. Please note, that 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 scattered environment lighting similar to the static skydome texture. To do this, enable the Procedural Sky > Use as Environment Lighting and Environment Lighting > Sky Texture options. If the world texture for environment lighting already exists, it will be replaced.



Supported settings:

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

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

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

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

Procedural Sky > Scatter Strength Light scattering factor. The default value is 0.2.

Procedural Sky > Rayleigh Strength Rayleigh scattering factor. The default value is 0.2.

Procedural Sky > Mie Strength Mie scattering factor. The default value is 0.006.

Procedural Sky > Rayleigh Collection Power Rayleigh scattering exponent. The default value is 0.35.

Procedural Sky > Mie Collection Power Mie scattering exponent. The default value is 0.5.

Procedural Sky > Mie Distribution Mie scattering distribution. The default value is 0.4.

19.2.2 Fog

Can be set up under the World tab.

Mist > Density Exponential factor which affects density and the visibility distance. The default value is 0.0.

Mist > 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.

19.2.3 Time of Day

Enable the Dynamic Intensity options 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.

19.2.4 Stars

Stars setup is described in the Halo Materials section.

19.3 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)

19.3.1 Activation

Add a force field object of the Wind type.

19.3.2 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.

19.3.3 Animation of Grass and Tree Leaves

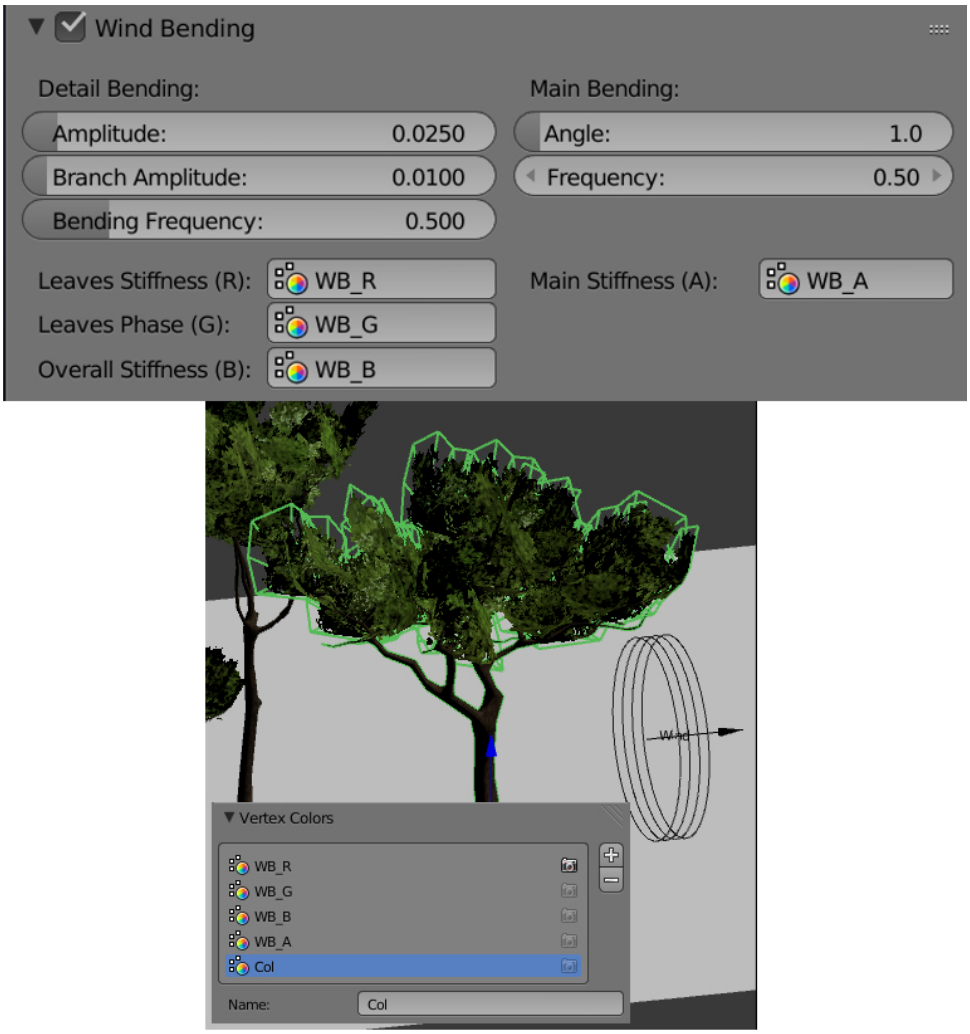
Authoring resources for grass rendering is described in the Grass section.

Activation

Enable the Wind Bending panel for the grass or tree object.object

Setting up

The interface panel becomes visible after turning on the Wind Bending panel.



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 > 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 > Bending Frequency Detail bendind frequency. The default value is 1.0.

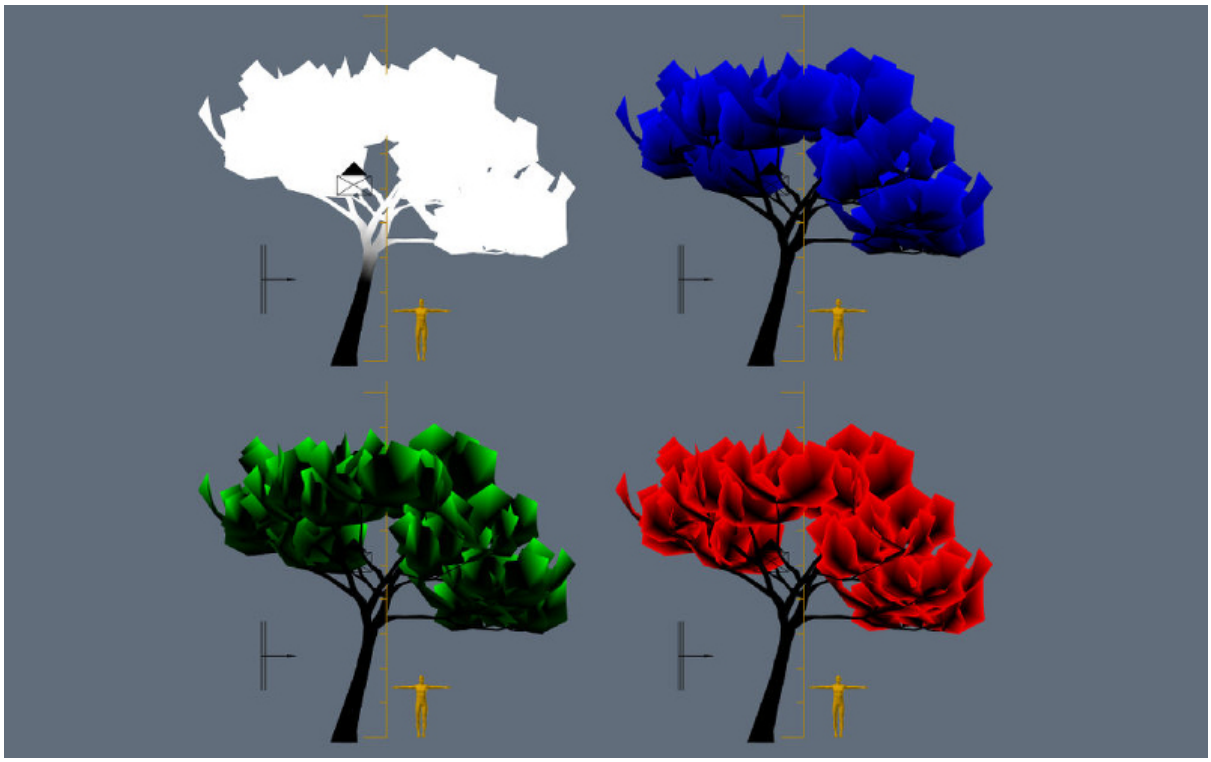
Detail bending > Leaves Stiffness (R) Text field for specifying the name of the vertex color layer which contains the information about the stiffness of leaves. Can be left

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.



CHAPTER 20

Gamma and Alpha

20.1 Gamma Overview

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

Standard (non-HDR) images are always stored in 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.

Overwise, 8 bit of information will not be enough to encode the light intensity. This will result in incorrect rendering of darker tones. For example, dark gradients will look stepped.

Therefore, web browsers, as well as many other programs for viewing and manipulating images, work in non-linear space. However, 3D engines and renderers work in linear space, because it 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.

Undoubtedly, 8 bit of information will not be enough in this case. This can be clearly seen from the table in which approximate illuminance values for some real light sources are shown.

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

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.

20.2 Human Vision and Monitors

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).

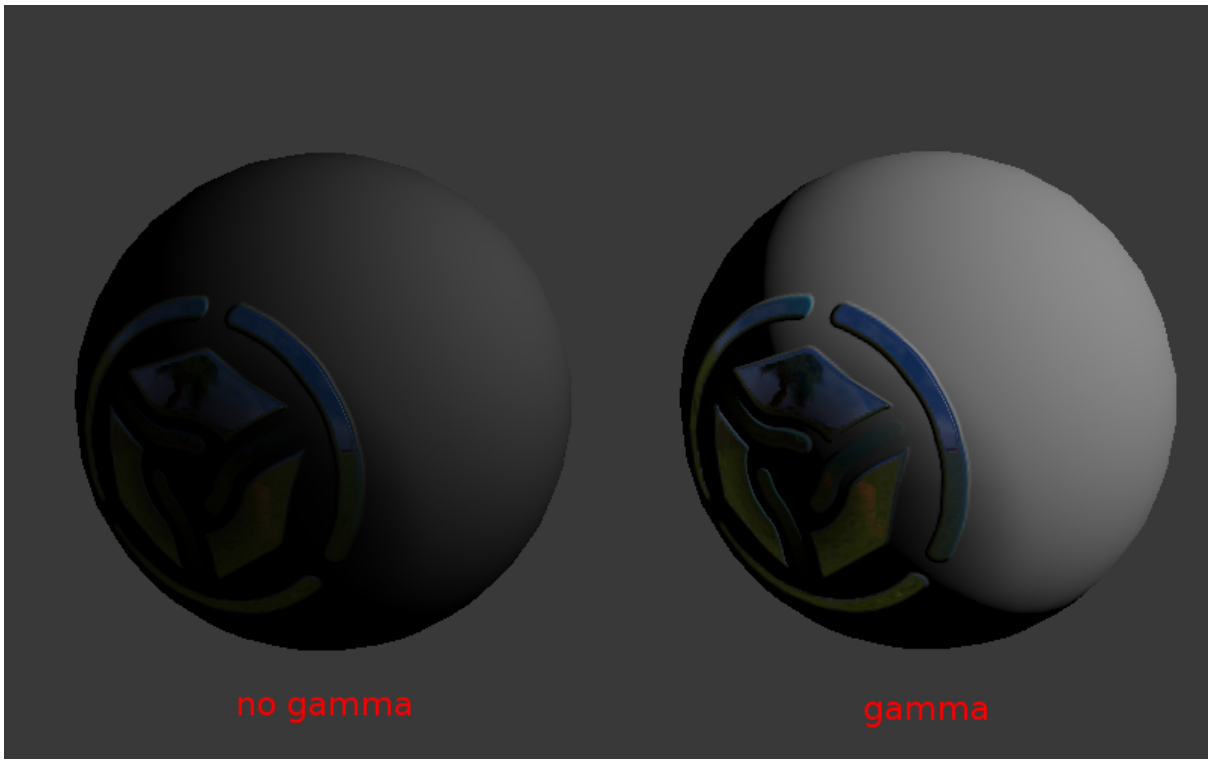
Therefore, the perception characteristics of the human eye are the reason due to which it is possible to pack color channels. At the same time, the technical characteristics of monitors have a secondary significance for gamma correction.

20.3 Gamma Formula

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".



20.4 Gamma in Node Materials

20.4.1 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.

20.4.2 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.

20.4.3 Normal Maps

No transformations are performed for normal maps.

20.4.4 Summary Table

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

See also:

B4W_LINEAR_TO_SRGB and B4W_SRGB_TO_LINEAR

20.5 Alpha Compositing

20.5.1 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.

20.5.2 Implementation

The blending function used in Blend4Web is the following:

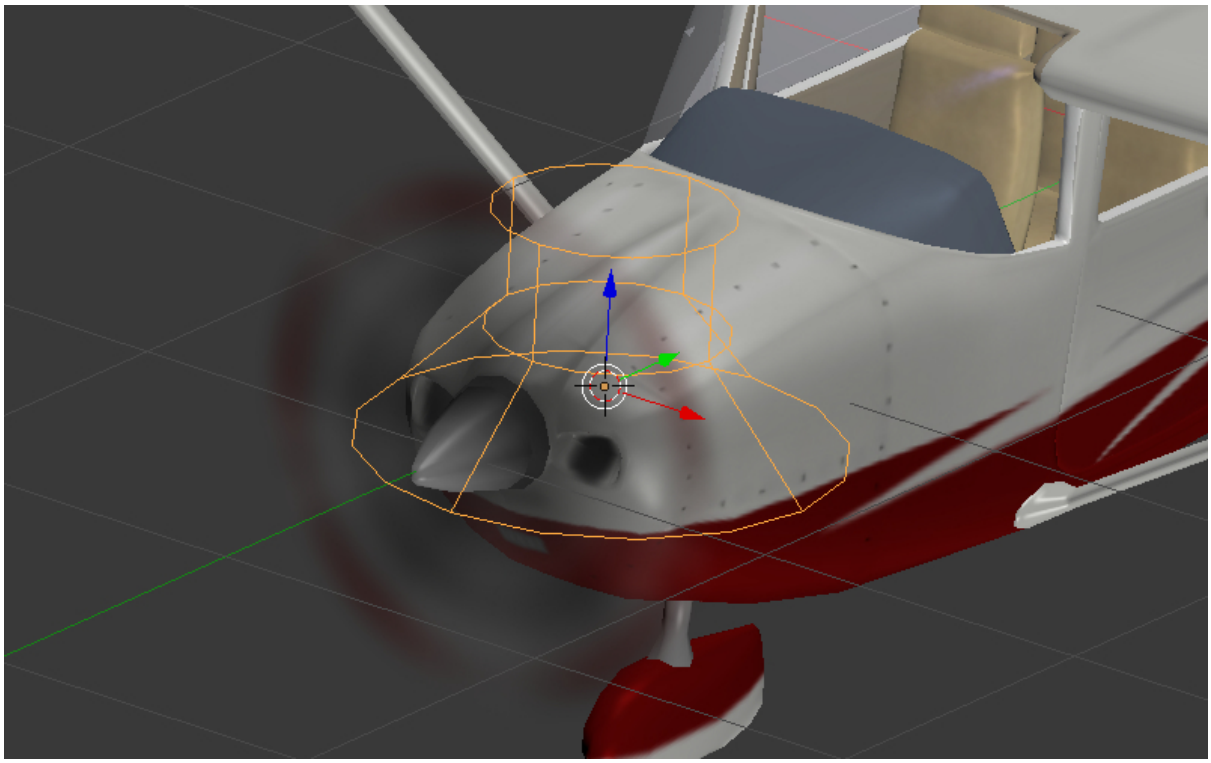
```
gl.blendFunc(gl.ONE, gl.ONE_MINUS_SRC_ALPHA);
```

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 21

Audio System

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



21.1 Audio Source Settings

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

The engine supports all the standard Blender sound parameters and some engine-specific settings.

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.

The following options are available on the Sound panel:

Volume

Speaker volume

Random Volume

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

Pitch

Sound playback velocity.

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.

Disable doppler

Ignore source's frequency shift upon its moving.

Cyclic play

Loop the sound playback.

Delay

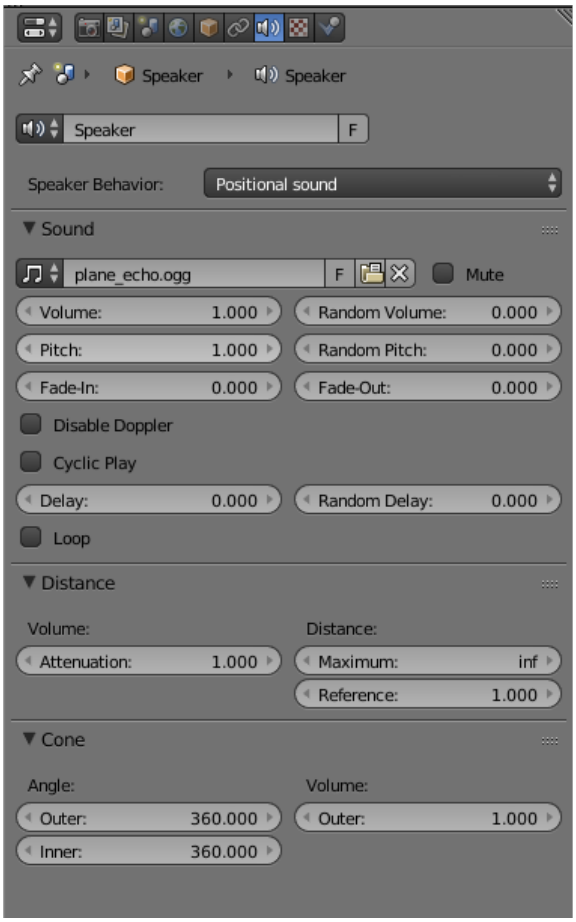
Delay before sound playback starts.

Random Delay

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

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.



21.2 Processing and Decoding

21.2.1 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.

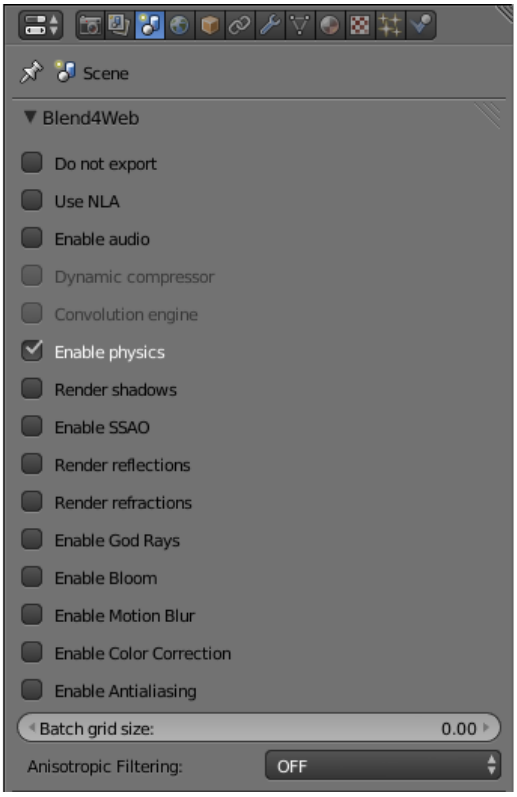
Converting resources between different formats is described in the corresponding section.

CHAPTER 22

Physics

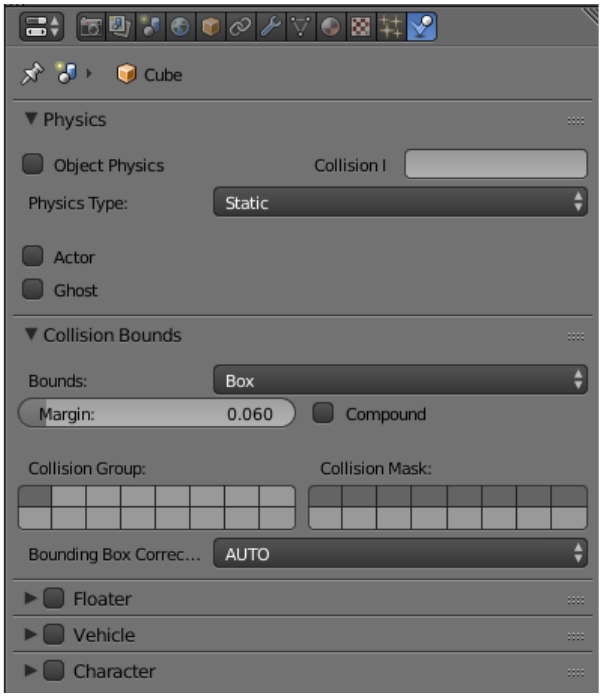
22.1 Preparing for Use

In order to enable physics on the scene, please use the Enable Physics checkbox in the Physics panel under the scene tab in Blender.

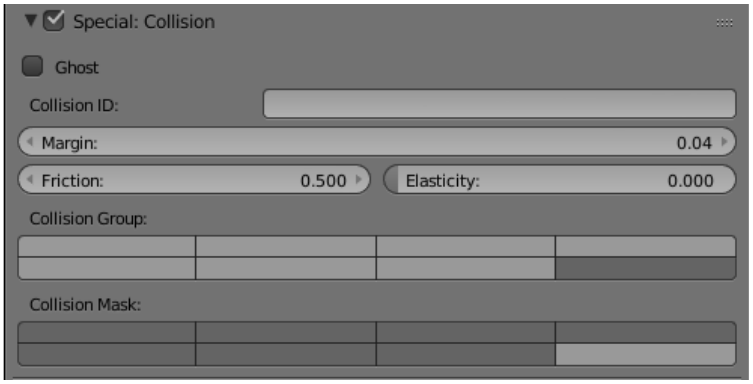


22.2 Static Physics Type

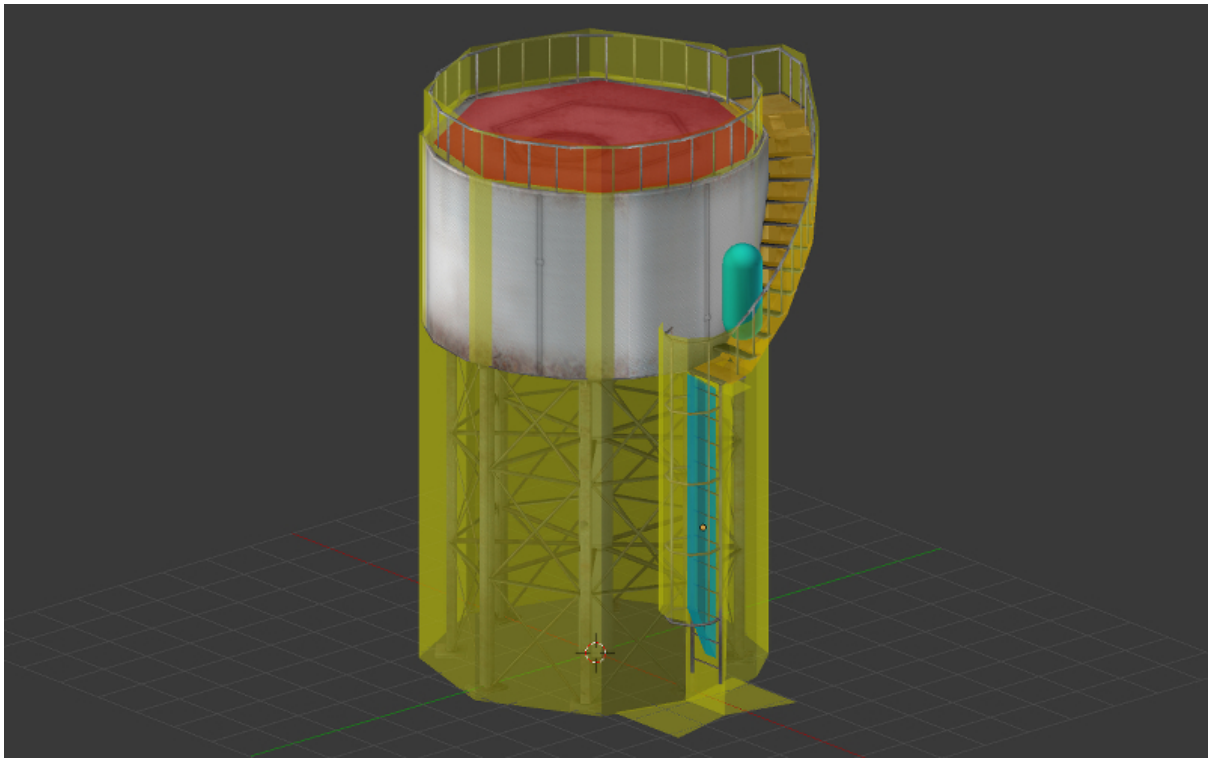
Can be used to limit the movement of other objects, for example to detect collisions with a landscape, walls and so on. In the physics settings of such an object the Static value (set by default) should be selected for the Physics Type option.



One or multiple physics materials can be assigned to a mesh. Under the Material tab the Special: Collision panel must be activated.



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 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.

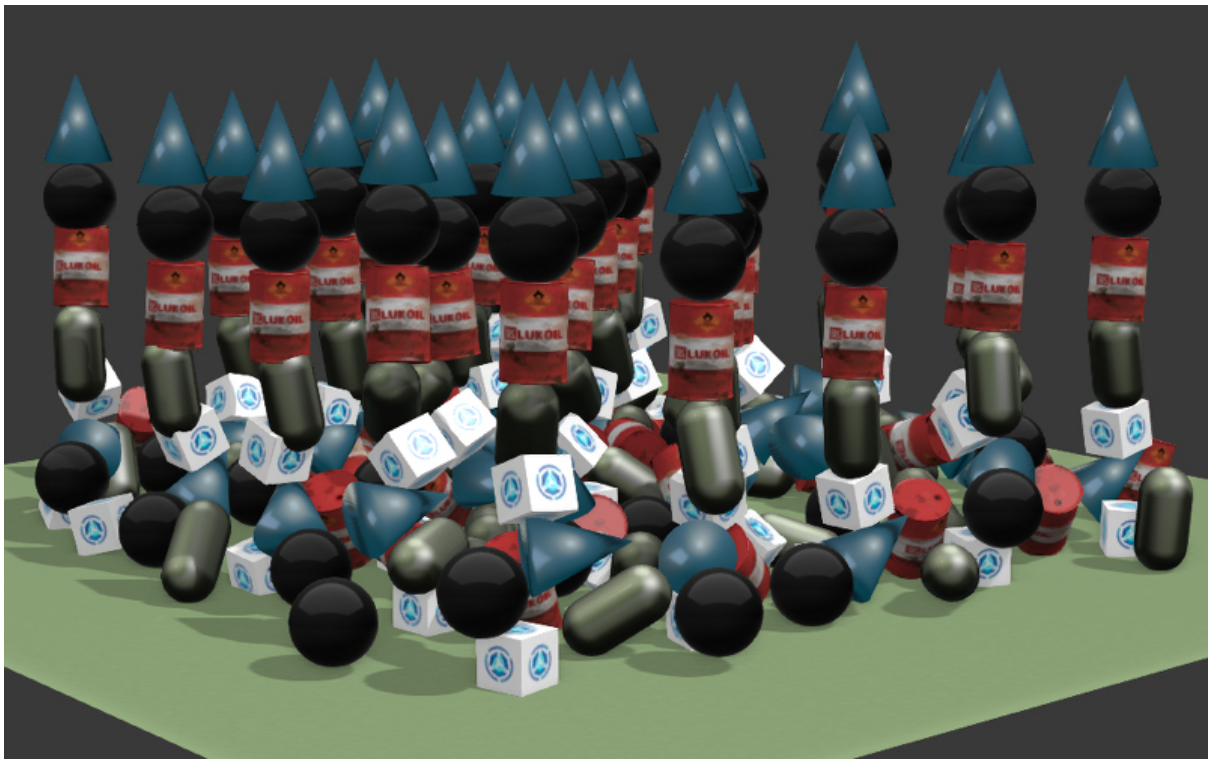
The Margin field allows to customize the width of the zone where mesh reacts on collisions. This option improves physical collisions simulation stability.

Also, there are material physics settings in this panel. The following settings are supported: Friction, Elasticity.

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.

22.3 Dynamic Physics Type

Intended for rigid body movement simulation.

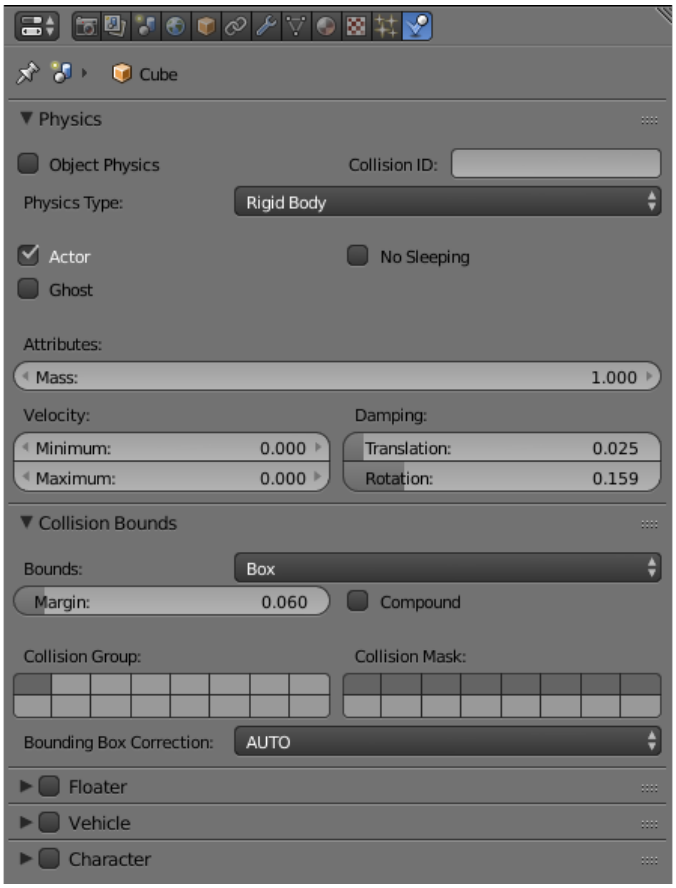


The Object Physics checkbox must be enabled under the object's Physics panel. The Collision ID field is intended for detecting collisions with a specific object (for example, for detecting proximity of a FPS character to different items) and can be left empty.

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. Also, the following physics parameters can be set: 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.

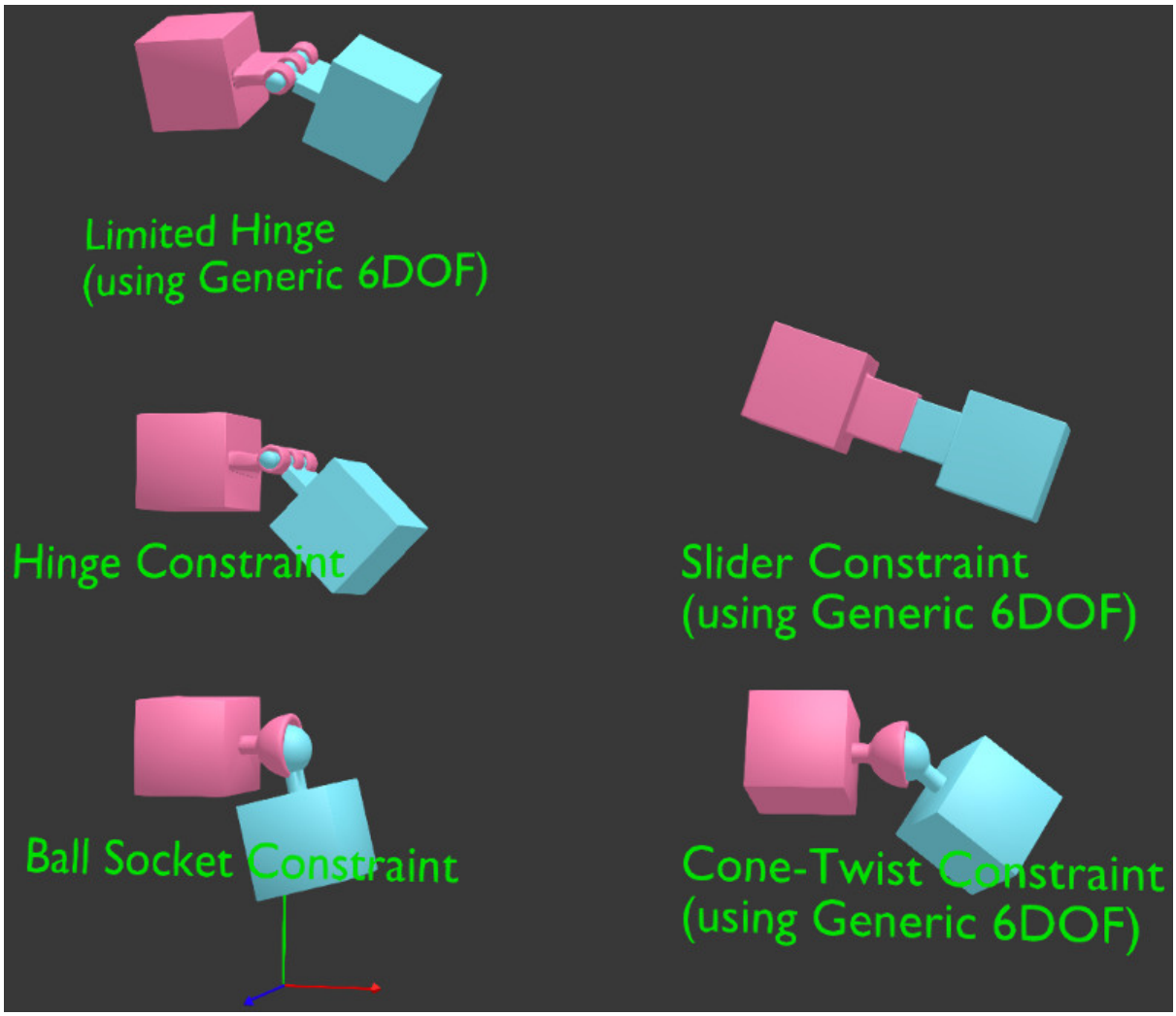


Friction and Elasticity are supported for the material of such an object. When multiple materials are used on a single mesh, the physics parameters are taken from the first of them.

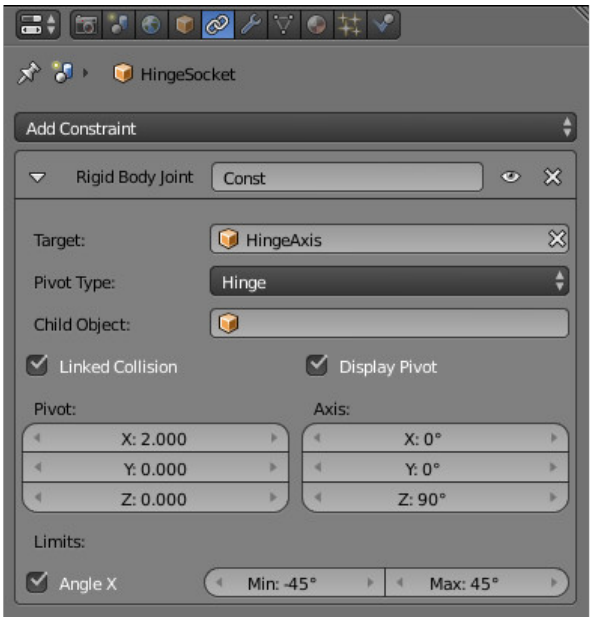
For the camera object the Physics Type = Dynamic parameter must be used, and the Object Physics checkbox must be enabled.

22.4 Constraints

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

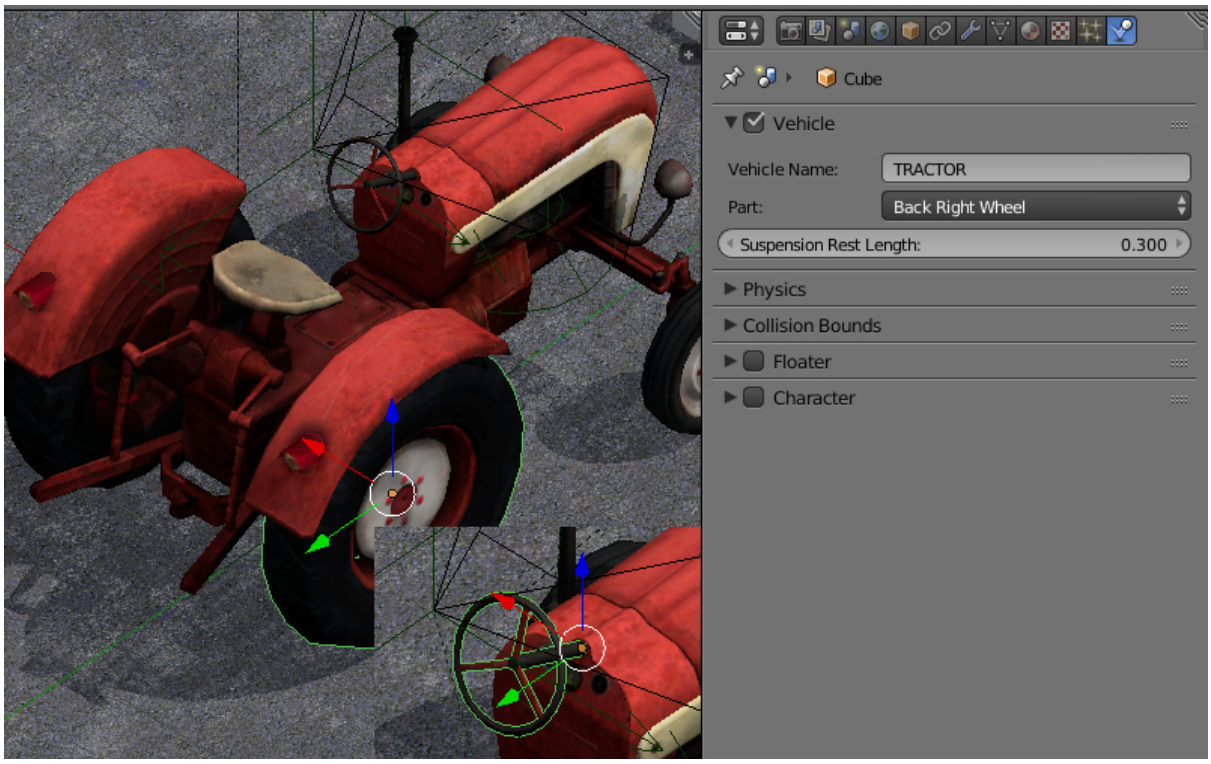


Adding a physical constraint (Rigid Body Joint) to the object can be performed on the Object Constraints panel. The supported types (Pivot Type) are: Ball, Hinge, Cone Twist, Generic 6 DoF. A physical constraint can be added to one of the two interacting objects, while the other object acts as a Target.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



22.5 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 Part, specify the same id in the Vehicle Name field, select the right object type - Chassis, Steering Wheel, Back Right Wheel and so on. The 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.

22.5.1 Chassis Settings

Force Max Maximum driving force of the vehicle.

Brake Max Maximum braking coefficient.

Suspension Compression Damping coefficient for suspension stretching.

Suspension Stiffness Suspension stiffness coefficient.

Suspension Damping Suspension damping coefficient.

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).

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

Max Suspension Travel Cm Maximum suspension travel in centimeters.

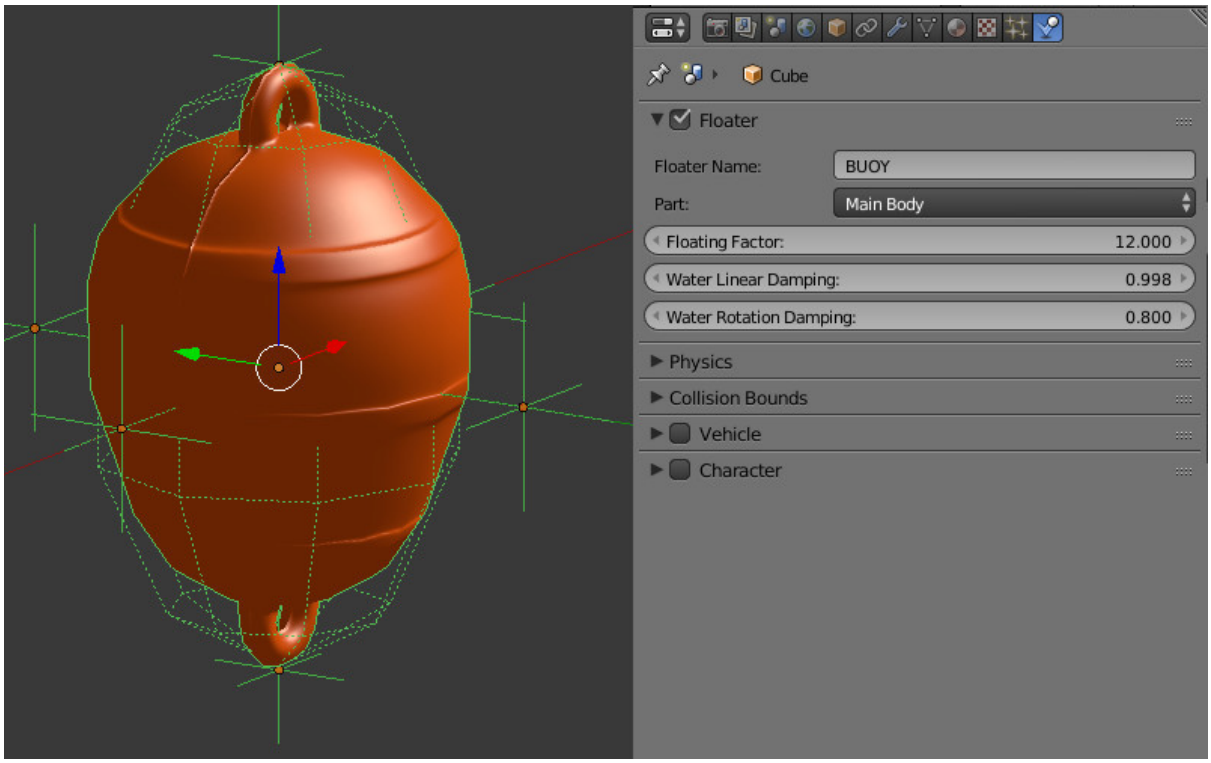
For the Steering Wheel it is necessary to specify the maximum steering angle (Steering Max) and the ratio between the turn of the steering wheel and the turn of the wheels (Steering Ratio). The maximum steering angle value is specified in revolutions. A single revolution equals to 360 degrees. Therefore 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.

22.6 Floating Objects



In order for the object to float on the water surface (an object with the Water material), it is neccessary to enable the Floater panel. 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.

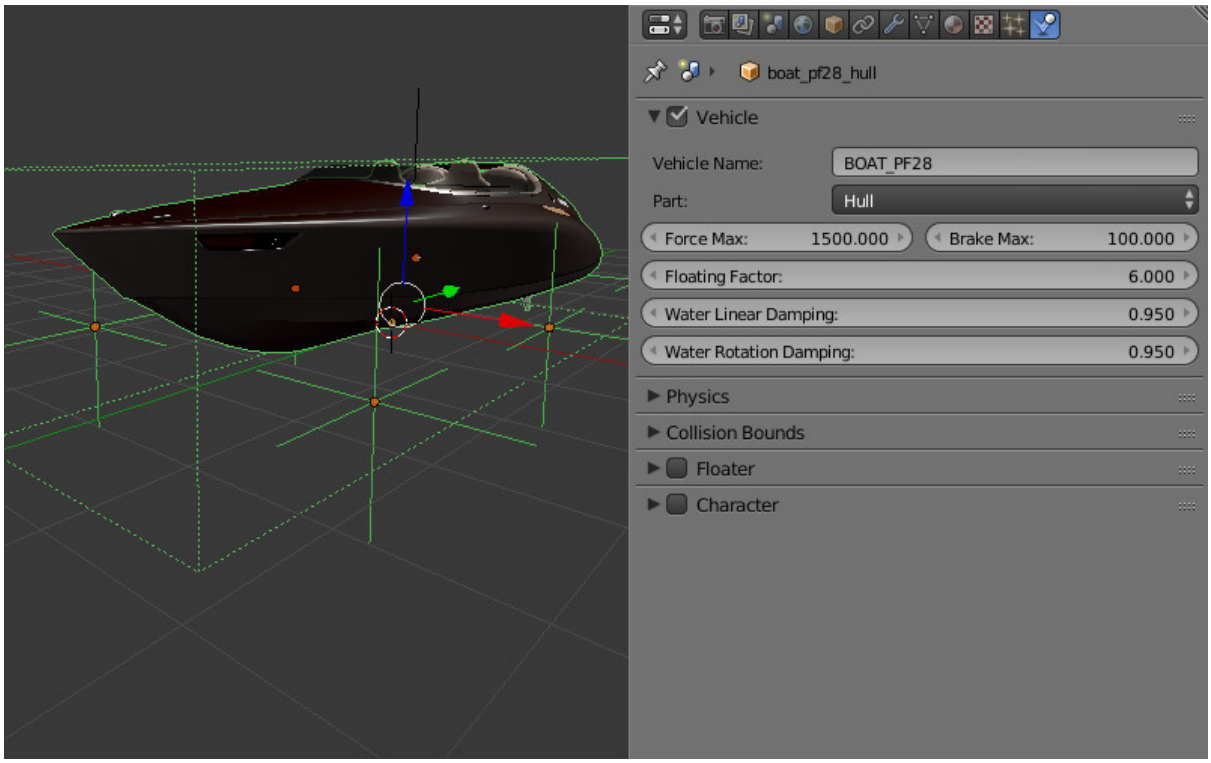
22.6.1 Floating Object Settings

Floating Factor Buoyancy coefficient.

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.

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.

22.7 Floating Vehicles aka Watercrafts



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

22.7.1 Watercraft Settings

Force Max Maximum driving force of the vehicle.

Brake Max Maximum braking coefficient.

Floating Factor Buoyancy coefficient.

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.

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.

22.8 Use in Applications

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. In order to activate the physics system, it is enough to put the uranium.js and uranium.js.mem files in the same directory as the source code of the application.

Another way is to explicitly specify the loading path of the uranium.js module by using the following API method:

```
m_config.set("physics_uranium_path", ".../uranium.js");
```

Note: When applications are developed within the SDK, the path to the physics engine is detected automatically.

If your application does not use physics, we recommend you to turn off the Enable Physics flag in the Physics panel under the scene tab in Blender. It is also possible to forcibly disable loading of the uranium.js module by calling the following method before initialization of the engine:

```
m_config.set("physics_enabled", false);
```

CHAPTER 23

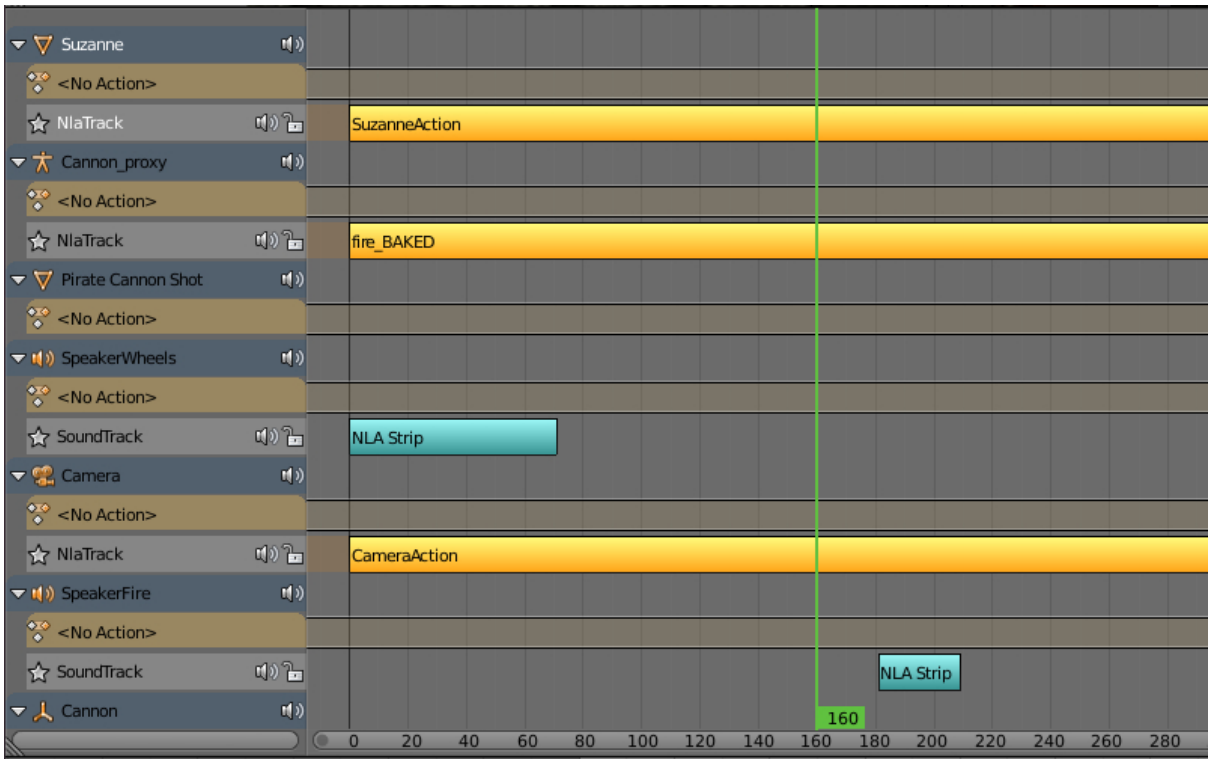
Non-Linear Animation

23.1 NLA Editor

The Blender's non-linear editor lets us set the scene's behavior in a comfortable way. With its help we can implement simple scenarios. This way coding is not needed for simple scenes and 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)



23.1.1 Usage

1. Activate the NLA panel 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.

23.1.2 Additional settings

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

Note: In order to use vertex animation, enable "Allow NLA" option on the vertex animation panel.

23.1.3 Limitations

- A simultanious playback of differnet types of animation for the same object is not supported.

23.2 Visual Programming (NLA Script)

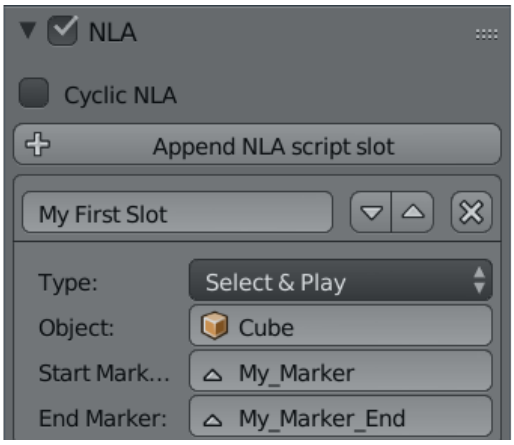
Performed by appending logic slots (NLA Script Slot) under the scene tab in Blender. These slots can extend the scene functionality significantly without any coding.

The slots themselves are logic blocks that are executed from the first to the last sequentially except that this slot explicitly contains a transition (Jump or Conditional Jump). When the end of the list is reached the logic stops to execute, or - if NLA > Cyclic NLA checkbox is enabled - starts again.

Note: Using NLA > Cyclic NLA together with NLA Script does not result in automatic looping of the animation.

For implementing complicated logic there are numeric variables called registers. Each of 8 registers can store a single numeric value. The registers can be used for storing some scene state (e.g. this can be a counter of animation playbacks, character's health points etc).

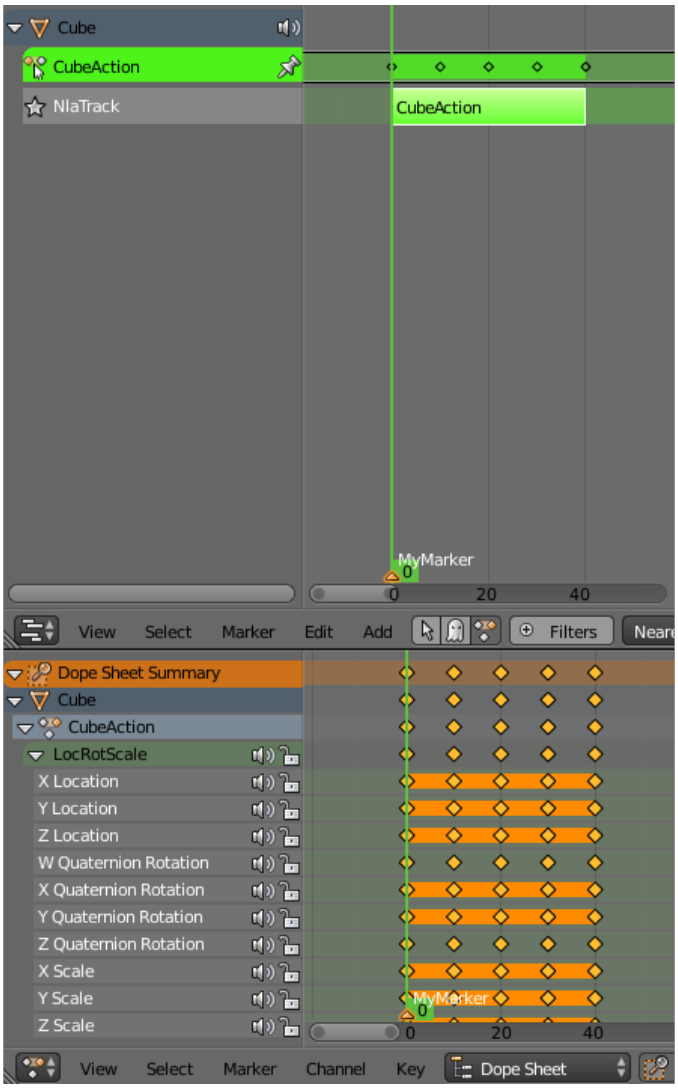
NLA Script usage example:



All possible slots are described below.

23.2.1 Play

Play back the NLA segment starting from the frame to which the marker points. Animation is played back until the next marker is met, or to the end of the timeline. Finally the next slot receives the control.



23.2.2 Select & Play

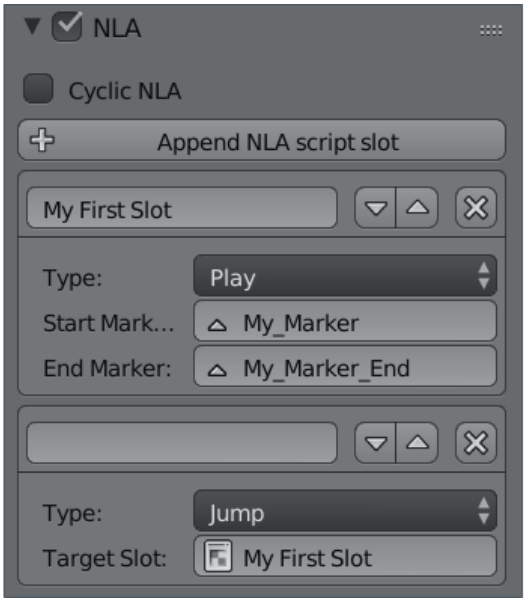
Wait until the user selects an object (on desktops - with a mouse click, on mobile devices - with a touch). If an object that is specified in this slot is selected - start animation similar to Play slot. If any other object is selected - immediately transfer the control to the next slot.

The latter functionality can be used for selecting one of the multiple objects. In this case it is convenient to place Select & Play slots in the stack one by one. The result of the user selection is catched up by one of this sequence slots because the transition without animation is performed instantly.

Note: In order to allow the user to select an object, enable the Selectable checkbox in its settings, similar to implementation of object outlining.

23.2.3 Jump

Go to the specified slot.



23.2.4 Select & Jump

Is similar to the Select & Play slot, except the transition happens instead of animation. This function allows to implement a complicated logic because in this case there is a possibility to identify the user selection results (selecting an object leads to the transition to the named slot which is needn't to be the next in the list).

23.2.5 Conditional Jump

Go to the specified slot if the certain condition is met. The parameters (operands) can also be registers that are activated using the corresponding switches.

23.2.6 Register Store

Save a numeric value in the register.

23.2.7 Math Operation

Perform a math operation and save the result in the register. Any of parameters (operands) can be either a numeric value or a register.

23.2.8 Show Object and Hide Object

This is used to hide and show 3D objects.

23.2.9 Page Redirect

This serves as a page redirect to other web pages.

23.2.10 Page Param

Allows to store any web page parameter in a given numerical register.

23.2.11 Noop

Shortening of "No Operation". When such a slot is processed no actions are performed. This slot is convenient to place together with a Jump or Conditional Jump slot.

23.3 Controlling via API

Non-linear animation playback can be controlled via API methods of the nla.js module.

```
// ...
var m_nla = require("nla");
// ...
m_nla.set_frame(150);
// ...
var frame = m_nla.get_frame();
// ...
m_nla.play();
// ...
m_nla.stop();
// ...
```

Please note, that if the NLA Script is used, the set_frame, play, stop methods are not available.

CHAPTER 24

For Application Developers

24.1 Hello World!

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>
```

This app prints a message and the engine's version in the browser window. Let's look at this example in detail. The engine library (without add-ons) is embedded with the <script src="..."> element. Then, the app waits for the page to load and prints the current version in the browser window. In this example, version is the only used module which has a function with the same name - version(). A more detailed info about the usage of engine's modules and functions can be found in the API documentation.

The compiled engine file b4w.min.js can be copied from the SDK's deploy/apps/common directory and placed in the same directory as the HTML file.

24.2 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.

24.3 Creating Apps Quickly

Creating an app from scratch can be a tedious task, especially for beginners. To address this there is a special add-on for the engine called app:

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

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

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

function load_cb()
```

```
    m_data.load("some_scene.json", loaded_cb);
}

function loaded_cb() {
    m_app.enable_controls();
    m_app.enable_camera_controls();
}

</script>
</head>

<body>
    <div id="container_id" style="width: 350px; height: 200px;"></div>
</body>

</html>
```

In this case the app module will create a <canvas> element inside the container with the specified container_id id. Then it will initialize the engine after the page is loaded and will finally execute the load_cb callback.

Then the some_scene.json scene is loaded similar to the previous example. The only difference is that after the scene is loaded, the control system is initialized and camera movement with keyboard and mouse (or sensor screen) becomes possible.

In case when the app module is used, it is necessary to specify dimensions of the container element. Otherwise the created <canvas> element will have zero dimensions.

24.4 Developing Apps Within the SDK

The SDK includes a script apps_dev/project.py for building applications.

Example:

```
./project.py -a my_app_path -o advanced -b copy -v 15.02 -s my_resources_path
```

- With the "-a" parameter the app's directory can be specified.
- The "-o" parameter (optional) is used in order to specify the optimization level for JavaScript files. Available options: whitespace, simple (default) and advanced.
- With the "-b" parameter (optional) the engine's directory can be specified. Available options: link (default) - the engine file is linked from deploy/apps/common/, copy - the engine file is copied into the directory with the built app, combine - the engine file is merged with the minified script located in the app's root, compile - required engine modules are compiled together with the app code.
- With the "-d" parameter (optional) the directory for the compiled app can be specified.

- With the "-s" parameter (optional) the directory can be specified where the scene resources are located.
- The "-p" parameter (optional) specifies the directory for scene resources, relative to the app root.
- The "-v" parameter (optional) adds version statement to the urls of scripts and styles. This will force the browser to use updated scripts and styles instead of loading them from the browser cache.
- The "-j" parameter (optional) specifies JavaScript files to be excluded from compilation.
- The "-c" parameter (optional) specifies CSS styles to be excluded from compilation.

App structure:

- Single HTML file should be located in the app's root.
- Scripts and styles can be located both in the app's root and in the nested directories.

How the app builder works:

- An app is copied as a whole into deploy/apps/app_name/.
- If the '-d' parameter is used, the app is compiled into the corresponding directory.
- Scripts and styles are compiled relative to the parent directory.

24.5 Non-FullScreen Web Apps

The Canvas element, to which the rendering is performed, can change its position relative to the browser window. This can occur due to some manipulations over the DOM tree, or as a result of page scrolling which is especially relevant for non-fullscreen web applications.

In most cases this will not affect the performance of the app by any means. However, some DOM events related to mouse cursor or touch position may carry incorrect information. This occurs because the coordinates obtained from the corresponding events are measured relative to the origin of the browser window, while the engine works with the coordinate space of the Canvas element itself (its origin is located in the top left corner).

In order to obtain coordinates suitable for use in the engine, you can transform them by using the client_to_canvas_coords method of the container module:

```
var m_cont   = require("container");
var _vec2_tmp = new Float32Array(2);
// . . .
var canvas_xy = m_cont.client_to_canvas_coords(event.clientX, event.clientY, _vec2_tmp);
// . . .
```

In order to obtain coordinates in the Canvas space, the engine should know its position relative to the browser window. However, if this position is subjected to changes during

the work of the app (due to scrolling for example), the Canvas position should be recalculated. To do this automatically, you can set the track_container_position property upon app initialization:

```
exports.init = function() {
    m_app.init({
        // . . .
        track_container_position: true,
        // . . .
    });
    // . . .
}
```

Please note, that this setting can lead to performance degradation in some browsers (such as Firefox) due to frequent DOM tree accesses. If the performance is critical, you can update the Canvas position manually when it is really necessary. To do this, use the force_offsets_updating and update_canvas_offsets methods instead of the track_container_position setting, or even the lower-level set_canvas_offsets method from the container module:

```
var m_cont = require("container");
// . . .
m_cont.force_offsets_updating();
// . . .
m_cont.update_canvas_offsets();
// . . .
m_cont.set_canvas_offsets(offset_left, offset_top);
// . . .
```

24.6 Code Examples

The SDK includes the Code Snippets application which demonstrates how to use the engine's functionality.

Currently, this application contains the following examples:

- Canvas Texture - working with canvas textures
- Camera Animation - procedural camera animation
- Camera Move Styles - changing control modes for the camera
- Custom Anchors - creating custom annotations
- Dynamic Geometry - procedural geometry modification
- Gyro (Mobile Only) - working with mobile devices' gyroscopes
- Instancing - copying scene objects in runtime
- Material API - tweaking material properties and replacing objects' materials
- Morphing - using shape keys

The Code Snippets application is available at SDK/apps_dev/code_snippets/code_snippets_dev.html
It can be also run by using a link in the index.html file located in the SDK root.

24.7 Resource Conversion

Existing browsers do not fully support all possible media formats, so in order to create cross-browser applications and for optimization purposes we need to use a resource converter. The conversion is performed according as follows:

for audio (convert_media):

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

We recommend to use Ogg as the basic format. In this case the only conversion required for cross-browser compatibility will be ogg to mp4. Example of an input file: file_name.ogg, example of an output file: file_name.altconv.mp4.

for video (convert_media):

- webm -> m4v
- m4v -> webm
- ogv -> webm
- webm -> seq
- m4v -> seq
- ogv -> seq

We recommend to use WebM as the basic format. In this case the only conversion required for cross-browser compatibility will be webm to m4v (webm to seq for iPhone). Example of an input file: file_name.webm, example of an output file: file_name.altconv.m4v.

for images (convert_dds):

- png -> dds
- jpg -> dds

Example of an input file: file_name.jpg, example of an output file: file_name.altconv.jpg.dds.

For the purpose of optimizing application performance it's possible to use min50 (halved) and DDS textures. In order to do this, we need to pass the following parameters during the application initialization.

```
exports.init = function() {
    m_app.init({
        // . . .
        assets_dds_available: true,
```

```
        assets_min50_available: true,
        // . . .
    });
    // . . .
}
```

The .seq file format is used for sequential video. This is applied for IE 11 and iPhone because they are currently missing support for standard video formats for textures. Using dds format for images is more optimal compared to other formats.

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 [-d file_path] resize_textures | convert_dds | convert_media
```

With the -d parameter you can specify the path to a directory in which converting will take place.

To exclude some directory from resource conversion, it is enough to put a file named .b4w_no_conv in it. This will not affect conversion in nested directories.

The resize_textures argument is used for decreasing texture resolution for the LOW mode.

The engine can use files which are manually created by a user if they have the following names: file_name.altconv.m4v, file_name.altconv.mp3 and so on. Such files should be placed in the same directory as the mediafiles used in Blender.

You can also use the free and cross-platform application Miro Video Converter to convert mediafiles.

24.8 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();
```

24.9 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.

24.9.1 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 payload. 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.

24.9.2 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
```

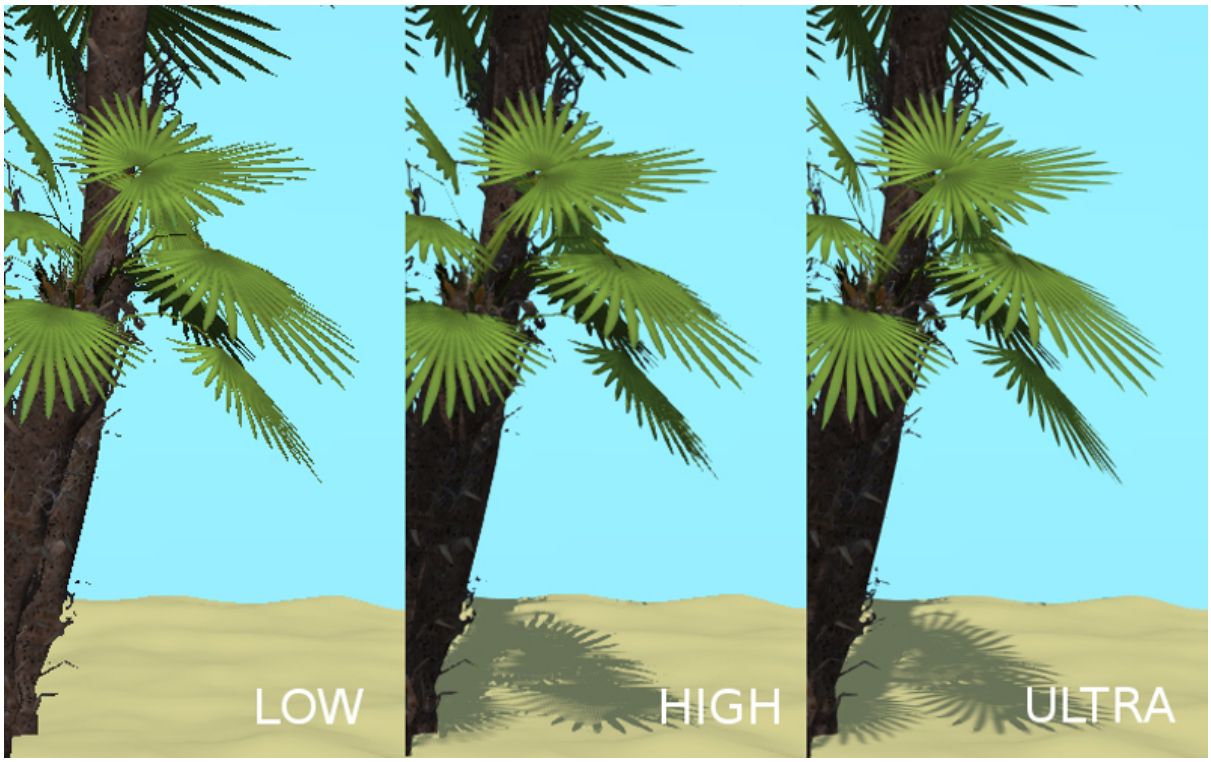
```
m_controls.create_sensor_manifold(stone, "IMPACT", m_ctl.CT_SHOT,
    impact_sens_array, impact_sens_logic, impact_cb);
```

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).

24.10 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 add-on:

```
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
});
```

24.11 SDK File Structure

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

Makefile the file for building all applications from the SDK

project.py script for application developers

viewer the sources files of the Viewer application

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

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

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 the resource directory for deploying on the server (scene source files, compiled applications and documentation)

api_doc API documentation for developers (built automatically, based on the engine's source code)

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

common Compiled engine files. Shared by all applications from SDK (hence the name).

assets downloadable resources: scenes, textures and sounds

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

index.html и index_assets main SDK webpage files

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.py script for building engine code and applications

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

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

reexporter.py script for automatic reexport of all scenes from the SDK

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

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

shaders GLSL shaders of the engine

src main source code of the engine's kernel

addons source code of engine addons

ext source code of the external declarations that form the engine's API

libs source code of the libraries

tools Various tools for building the engine and applications

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

glsl

compiler compiler for the engine's GLSL shaders

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

yuicompressor utility for compressing CSS files

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

VERSION contains the current version of the engine

24.12 Canvas Textures

Such textures can be accessed via the textures module. The workflow is described below.

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

The context can be obtained with the get_canvas_texture_context() function, to which a "canvas_id" identifier specified by the user in Blender is passed. After any operations with the context the update_canvas_texture_context() function should be executed, which performs visualization of the changes of the "canvas_id" element.

24.13 Setting up the Browser for Loading Local Resources

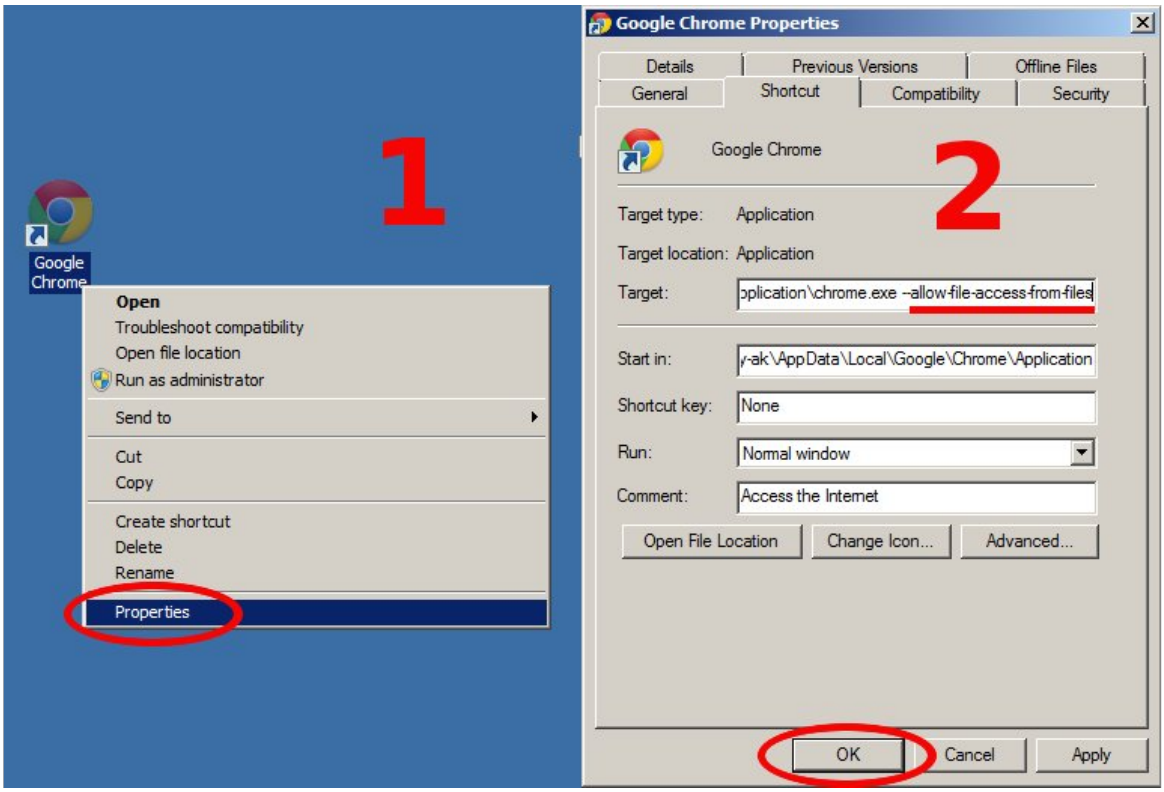
Since version 15.02 the Blend4Web SDK includes the development server to solve the problem of loading local resources. Nevertheless the following instructions can still be useful for developers.

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.

Setting up the browser for loading local resources can be a simple and versatile method to bypass security limitation. It is recommended to use such browsers only for viewing local content.

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/GoogleChrome.app/Contents/MacOS/GoogleChrome

> –allow-file-access-from-files

Chrome/Chromium under Linux:

Run the browser with the parameter:

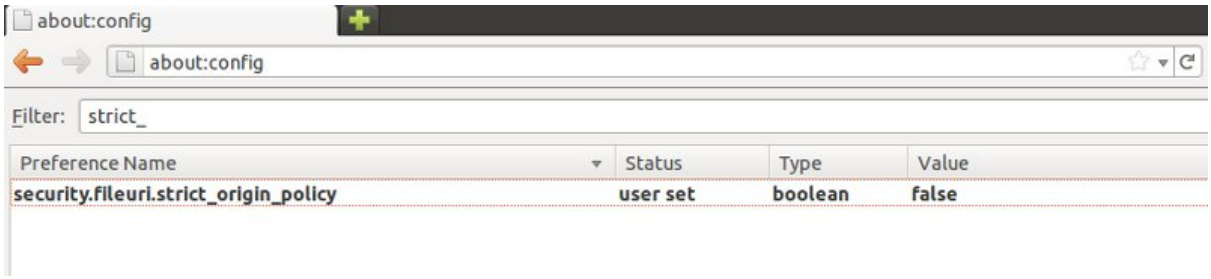
```
> google-chrome --allow-file-access-from-files
```

or:

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

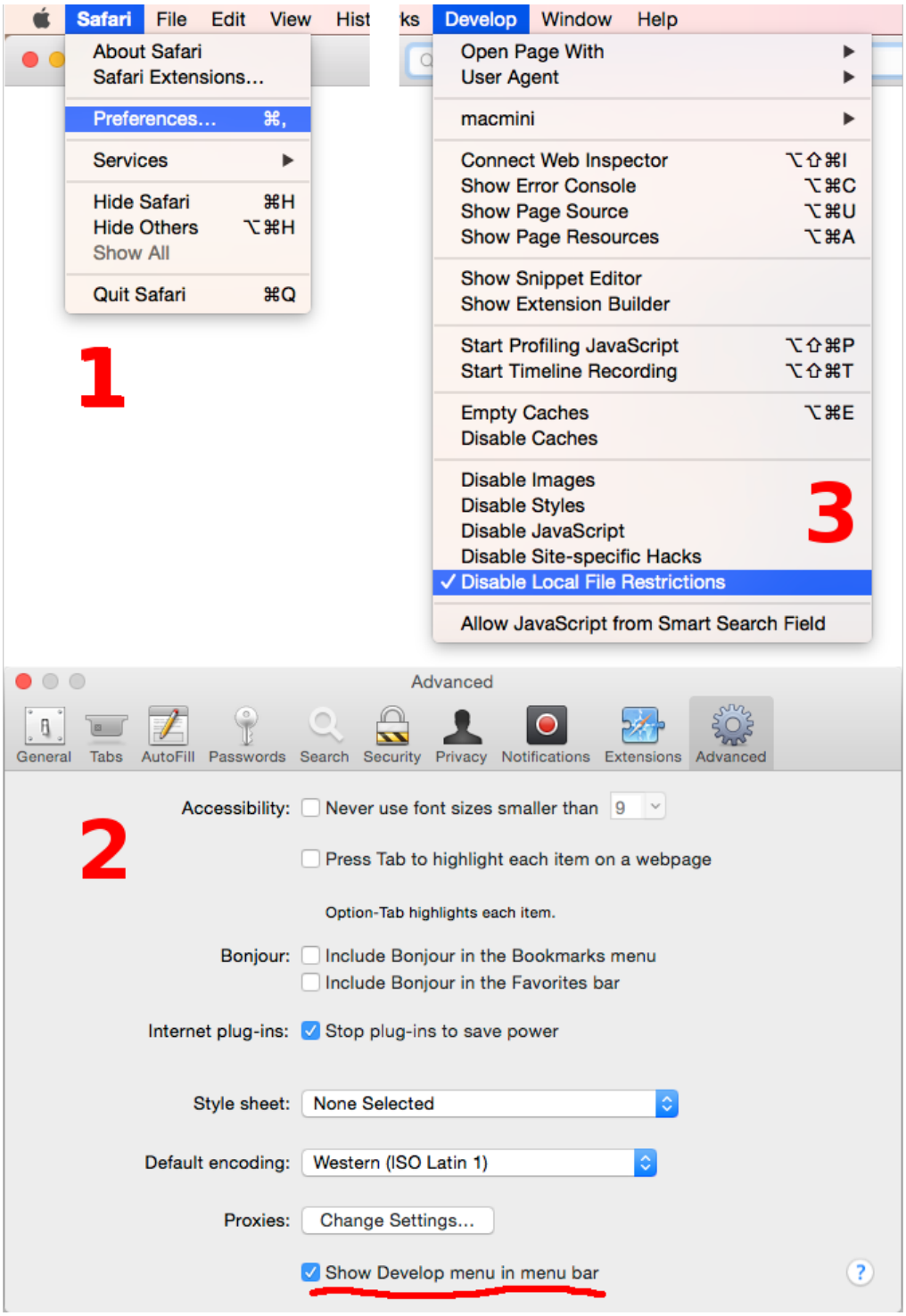
Firefox under Windows/Linux/OS X:

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.



24.14 Running a Local Server

You may prefer running your own web server instead of the development server included in the SDK. An easy way to do this is running the web server from the standard Python library.

Under Windows:

1. Download and install the latest Python distribution from the official site. When installing select the option for adding the executable to the path (Add python.exe to Path).
2. Run Command Prompt.
3. Type the following command in the SDK's root directory:

```
> python -m http.server
```

Under Linux/OS X:

1. If Python is not already present in your distribution, download and install its current version from the official site.
2. Run Terminal.
3. Type the following command in the SDK's root directory:

```
> python3 -m http.server
```

After starting the server, open the index webpage with the SDK links by navigating to http://localhost:8000.

If needed, a port number can be specified by the additional parameter:

```
> python3 -m http.server 8080
```

CHAPTER 25

For Engine Developers

25.1 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).

25.1.1 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;
```

The names of external API methods and properties are written after a point. To avoid obfuscation of fields they must be listed with the @cc_externs tag:

```
exports.FOO_BAR = 123;

exports.foo_bar = function() {

}

/**
 * Set properties.
 * @method module:properties.set_props
 * @param {Object} foo Foo object
 * @cc_externs props_1 props_2
 * @cc_externs props_3 props_4
 */
exports.set_props = function(foo) {

    var bar_1 = foo.props_1;
    var bar_2 = foo.props_2;
    var bar_3 = foo.props_3;
    var bar_4 = foo.props_4;

    ...
}
```

Commenting is in English only. Comment style - JSDoc.

25.2 Building the Engine

Before building please make sure that your system has all required dependencies installed (see the table).

To compile the engine and the applications included into the SDK please execute the following command (in the SDK root):

```
make compile
```

The full building that includes converting the resources (textures, sounds and videos), compilation and converting the docs, can be performed with the following command:

```
make build
```

Building the archives with the distributions:

```
make dist
```

All above mentioned operations can be performed with a single command:

```
make all
```

25.3 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:

- build script: csrc/b4w_bin/build.py
- Blend4Web addon: 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.

25.4 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	shader compilation
Java	default-jre	compiling and obfuscating the engine modules
Emscripten	from EMSDK source code	building Uranium
ImageMagick, GraphicsMagick	imagemagick, graphicsmagick	converting the resources
NVIDIA Texture Tools	libnvtt-bin	converting the resources
NVIDIA Cg Toolkit	nvidia-cg-toolkit	debugging shaders
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 JSDoc source code	building the API documentation
PEG.js	from PEG.js source code	shader preprocessing

25.5 Naming 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

25.6 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.

25.7 Shader Compilation

All shaders used in the engine are processed by a compilator. The compilator performs the following three main procedures:

- validation of the shader code,
- its obfuscation and
- optimization.

In order to run the compilator, execute one of the following commands in the SDK root:

```
> make compile_shaders
> make verify_shaders
```

- make compile_shaders - performs validation, obfuscation, optimization and finally, export of the compiled shaders,
- make verify_shaders - performs only validation, obfuscation and optimization.

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

25.7.1 Validation

The compilator performs the following procedures related to shader code validation:

- reporting about unused variables and functions (dead code),
- checking the syntax of shaders,
- checking the conformance of shaders to the import/export mechanism,
- removing odd or repeatitive tokens: spaces, endlines and semicolons.

25.7.2 Obfuscation

Obfuscation minifies the GLSL code and makes it diffucult to understand it. So far the following procedure is implemented:

- replacing the user-defined identifiers with shorter single-symboled, two-symboled etc names (with support of the import/export mechanism).

25.7.3 Optimization

Optimization constitutes the following procedures:

- removing curly brackets which are not useful in any ways except creating local scopes (this functionality is used for processing node/lamp directives),
- optimization inside functions - creating shared local variables to replace ones originally created by the programmer.

An example of removing unused curly brackets: replacing the following code

```
void function(){
    int a;
    {
        a = 1;
    }
}
```

with this code

```
void function(){
    int a;
    a = 1;
}
```

Low number of temporary local variables is achieved by repetitively using them in different contexts. For example, the following code

```
int function(){
    int a = 1;
    int b = a + 3;
    return b;
}
```

will be replaced with

```
int function(){
    int _int_tmp0 = 1;
    _int_tmp0 = _int_tmp0 + 3;
    return _int_tmp0;
}
```

Note: Local variables for structures and arrays are not optimized this way.

25.7.4 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.

25.7.5 Recommendations and Limitations

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

9. Do not declare variables with such names as node_[NODE_NAME]_var_[IN_OUT_NODE], where NODE_NAME — name of some node, IN_OUT_NODE — name of an input or an output of the node.
10. Repetitive use of #nodes_main, #nodes_global or #lamps_main directives is not permitted inside a single shader.

11. The #nodes_main, #nodes_global and #lamps_main directives are recommended to use in the file, containing these shader nodes description, for example, in the same include-file. This is necessary for the correct shader validation.

25.7.6 WebGL Extensions

Compilation may depend on WebGL extensions being used if they somehow influence the shading language. At the moment the following extensions are supported by the compilator:

- OES_standard_derivatives

25.7.7 Compilation Errors

In case of an error the compilator will output the corresponding message in 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'. Varying/uniform or varying/attribute qualifiers combination. File: 'FILE_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. 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 26

Team Work. Using Git

26.1 Overview

In order to organize team work a Git version control system can be used. Using Git has a number of benifits as compared with other ways to collaborate:

- saving the history of changes with the possibility to roll back to prevoius versions
- syncronizing changes between users and automatic merging of changes
- working with large binary files is possible

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 dedicated machine (server). Access to the server can be organized through SSH protocol.

Although there are many GUIs for Git beginners, here the work with the git standard console utility is explained.

26.2 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.

A limited set of Git commands recommended for authoring applications and their graphical resources is described below.

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

```
> cd ~/blend4web
```

26.3 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.

26.4 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")
```

26.5 Before the Commit

26.5.1 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);
+        }
     }
```

26.5.2 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.

26.5.3 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.

26.6 Preparing for Commit

26.6.1 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.

26.6.2 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.

26.7 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.

26.8 Syncing Between Repositories

26.8.1 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    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: blender/landscape_objects/Fallen_tree.blend (...)

Auto-merging blender/landscape_objects/Fallen_tree.blend
CONFLICT (content): Merge conflict in blender/landscape_objects/Fallen_tree.blend
Automatic merge failed; fix conflicts and then commit the result.
```

The steps to be taken at conflicts are described below.

26.8.2 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.

26.9 Resolving Conflicts

26.9.1 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.

26.9.2 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:      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.

26.9.3 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 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 blender/landscape_objects/Fallen_tree.blend
```

Select the local version of the file again (- -ours).

```
> git checkout --ours 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.

26.9.4 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).

26.9.5 Correcting commit

After selecting the required files or editing the changes, add them for commit:

```
> git add 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.

26.10 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
```

26.11 Other Useful Commands

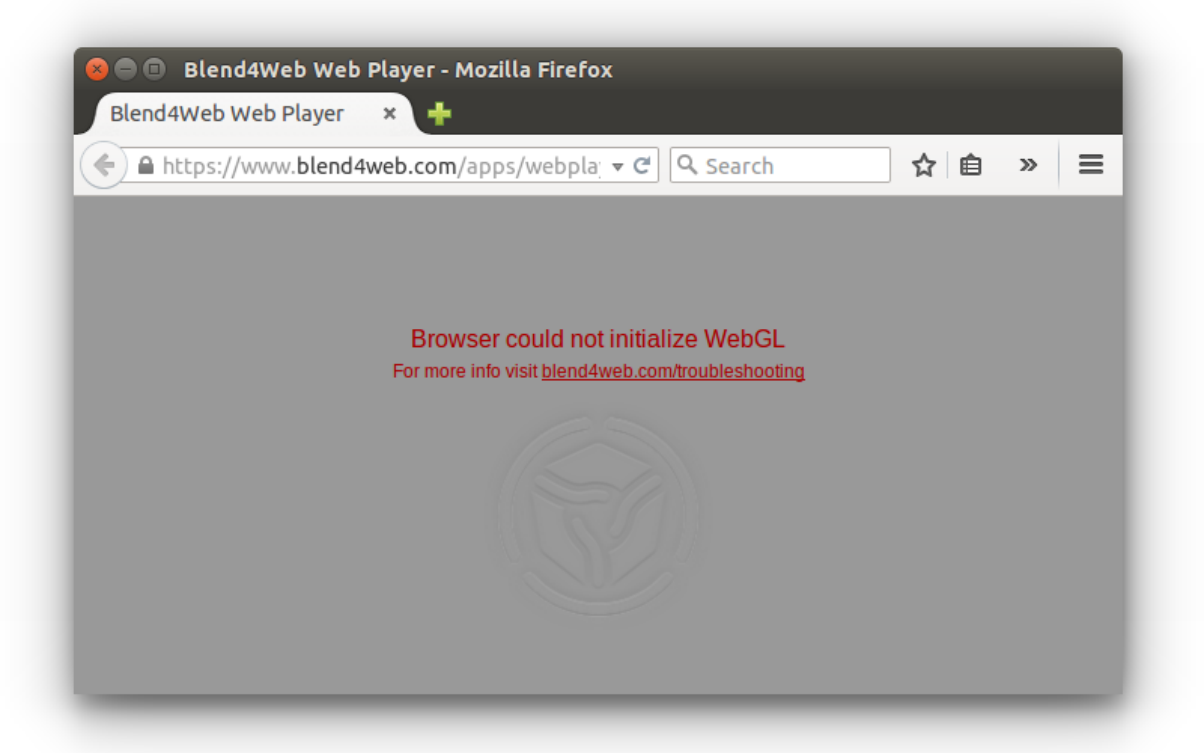
Check the log for January, 2012, show file names without merging commits:

```
> git log --after={2012-01-01} --before={2012-01-31} --name-only --no-merges
```

Problems and Solutions

27.1 Problems Upon Startup

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 or background is shown but the default scene is not rendered. At the same time the http://get.webgl.org/ site and other WebGL applications are working correctly.

Possible causes:

- A local web server is not used or the browser is not set up for loading local resources. See the Setting up the Browser for Loading Local Resources section.

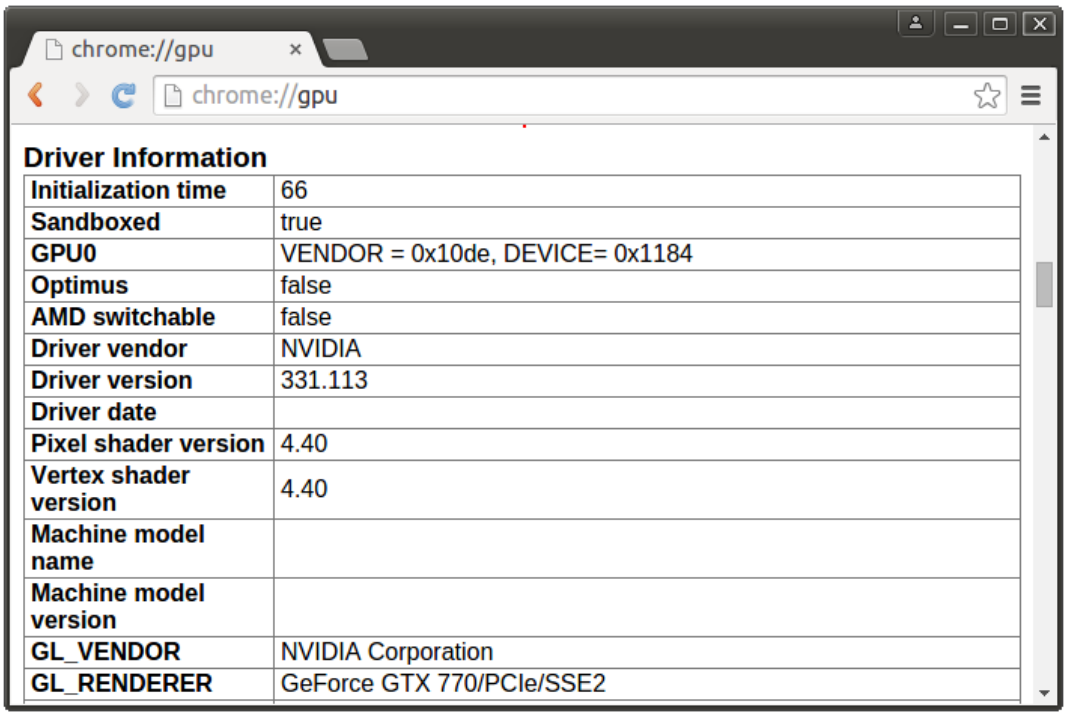
- The engine tries to load resource files which were moved or deleted.
- You are using the old versions of video drivers.
- You are using open source drivers which do not fully support WebGL.

  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.
- You are using an outdated operating system, such as Windows XP.

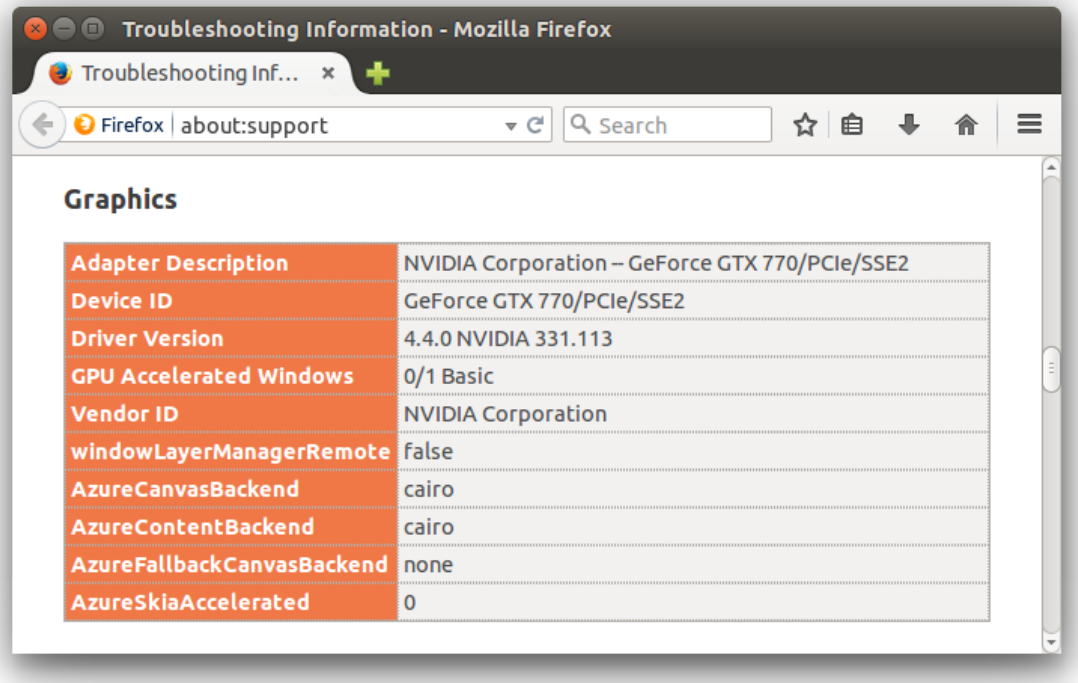
27.2 WebGL Failed to Init

The http://get.webgl.org/ page tells about problems when viewing it in recent Chrome or Firefox. What can I do?

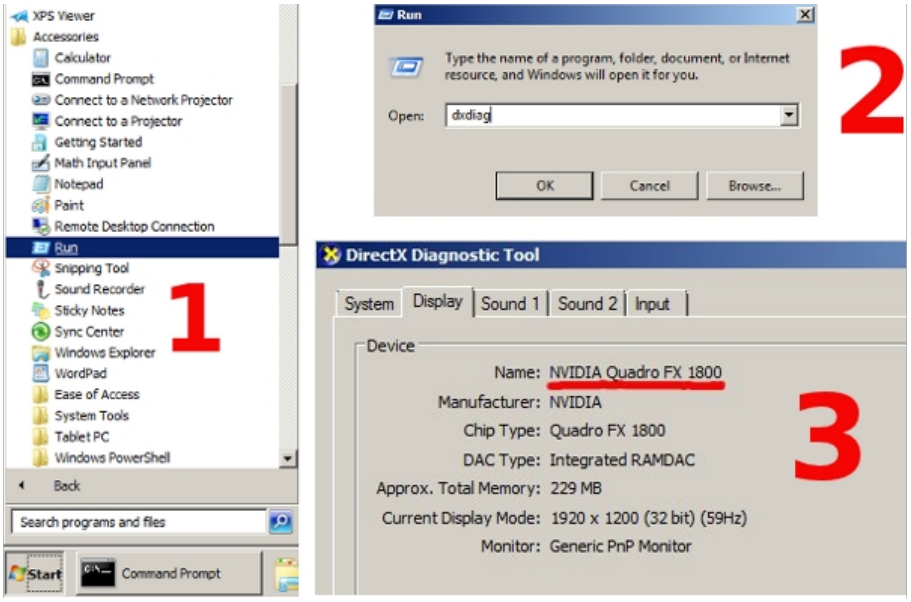
1. Install the latest updates for your system (for MS Windows see the guide). In case of MS Windows install the latest DirectX runtime. Reboot.
2. It is recommended to timely update video card drivers. To detect your video card and its vendor please type about:gpu (or chrome://gpu) to the address bar of Chrome browser...



or Firefox...



For Windows, you can run the DirectX Diagnostic Tool called dxdiag.

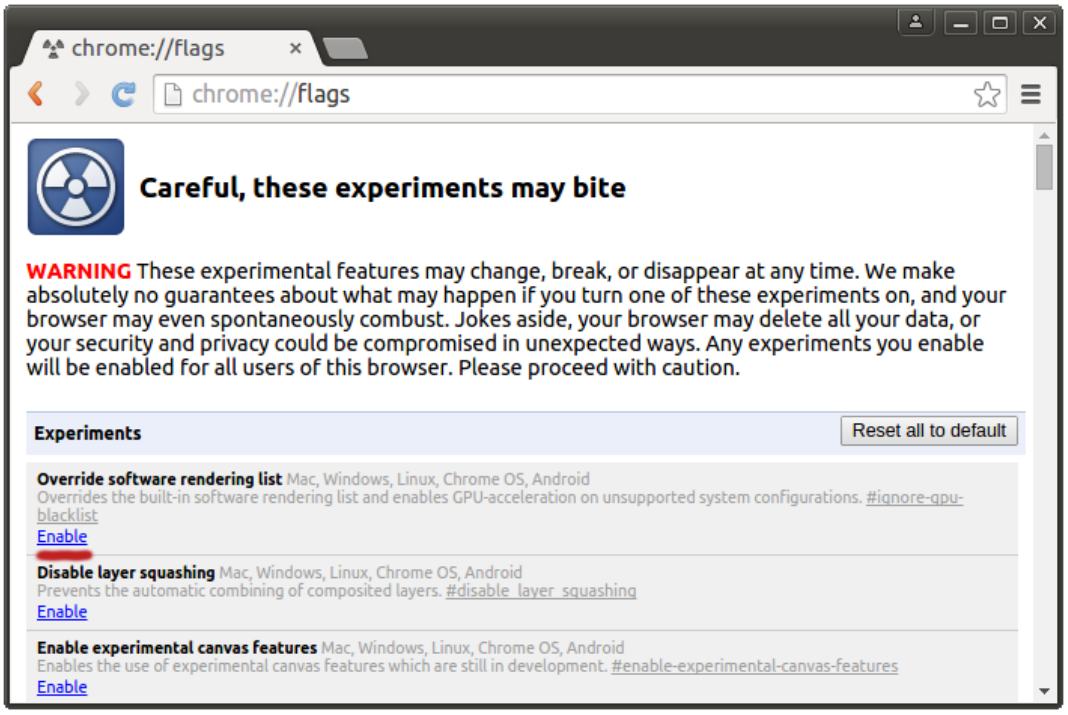


Download the drivers from the corresponding support center (for example Intel, Nvidia, AMD/ATI). Reboot the system after the drivers are installed.

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.

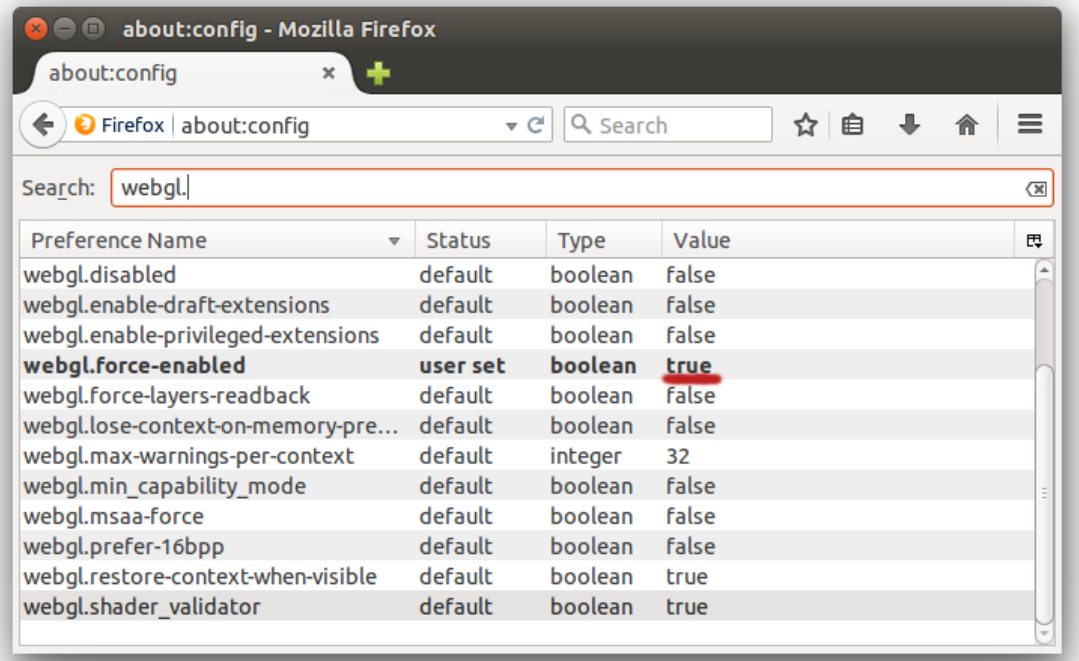
For Chrome:

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.


about:config - Mozilla Firefox
about:config
Firefox | about:config
Search
Search: webgl.
Preference Name
Status
Type
Value
webgl.disabled default boolean false
webgl.enable-draft-extensions default boolean false
webgl.enable-privileged-extensions default boolean false
webgl.force-enabled user set boolean true
webgl.force-layers-readback default boolean false
webgl.lose-context-on-memory-pre... default boolean false
webgl.max-warnings-per-context default integer 32
webgl.min_capability_mode default boolean false
webgl.msaa-force default boolean false
webgl.prefer-16bpp default boolean false
webgl.restore-context-when-visible default boolean true
webgl.shader_validator default boolean true

CHAPTER 28

Release Notes

28.1 v15.06

28.1.1 New Features

- New add-on user interface.

  Add-on interface has been redesigned. It is now activated by the new rendering profile, Blend4Web, which only contains panels and options explicitly supported by the engine. To simplify navigation, the old multi-line Blend4Web panels have been re-grouped into the smaller ones, based on functionality.

  Also, there is a new feature to automatically assign graphic effects required for a scene. In particular, shadows, refraction, Glow and Outline effects acquired the new AUTO property which automatically activates them if some objects or materials located on the scene require such effects.

- New normal editor.

  The normal editor has been substantially upgraded. Now it is fully compatible with the native Blender datablock used to store normals. This new editor has more efficient UI and also allows to edit split normals.

- Support for new material nodes.

  Orco and Local outputs of Geometry node are now supported. There are also some preliminary steps to support RGB Curves, Vector Curves, ColorRamp and Cycles nodes.

- New shading models.

  Minnaert/Toon diffuse shaders and Blinn specular shader have been implemented. Thus, starting from this release the engine supports all shading models of Blender.

- Support for Blender 2.75

- Multiple physics improvements.

Code for collision detection has been rewritten. Now it is possible to determine colliding objects, and also the coordinates and the normal at the collision point.

Improved ray casting API. In particular, one can specify an option to perform an automatic cleanup of the ray test object and also another option to cast a ray through multiple objects. As in the case of collision detection, this new API allows to determine the target object and the position/normal of the hit point. There is also a new possibility to cast rays from point to point in global space, without requirement to specify a source object.

Extended possibilites of Collision and Ray sensors.

Support for deleting physics objects and automatic recalculation of collision/ray tests after physics objects have been added/removed.

- A new tool for reexporting multiple scenes.

  A new Mass Reexporter tool has been added to addon. This tool allows to automatically reexport all scenes from the specified list of directories.

- Possibility to check for updates.

  You can now enable a Check for Updates on Startup option in addon settings to perform automatic checks for the new versions of Blend4Web.

- API to control Motion Blur postprocessing effect.

  New methods to control Motion Blur effect get_mb_params() and set_mb_params() has been added to scenes.js module.

- Support for Timeline markers.

  To extract frame numbers from timeline markers a new method marker_frame() has been implemented in scenes.js module.

- New NLA APIs.

  A new set of methods: set_range(), reset_range(), set_cyclic() and clear_callback() has been added to nla.js module. Also, it's now possible to specify callback in play() method.

- New API to change Canvas resolution.

  To change Canvas resolution it's sufficient to execute method set() from config.js module with the following parameters: canvas_resolution_factor and value, where value is the new resolution of Canvas. This feature is particulary useful for creating high-definition screenshots.

- Support for Vertex Groups -> Length option in Hair particle system.

- New API documentation.

  To document types used by Blend4Web applications we developed a new formal type system. For example, in our old API docs 3D vectors had

Float32Array type. Now they have formal Vec3 type. This solution allows us to formulate more clear and intelligible descriptions for API and, more importantly, helps our users to develop more readable and reliable applications.

- New colors API.

  To work with colors in efficient way two new modules: rgb and rgba have been created. They include APIs to create new color vectors and convert them between different profiles.

- Automatic determination of path to SDK.

  Addon option Blend4Web SDK Directory is filled up automatically, if the addon is located in default blender_scripts directory inside SDK.

- API for correct calculation of Canvas 2D coordinates.

  For proper manipulations with mouse cursor and touchscreen devices the engine requires correct 2D Canvas coordinates.

  Details about calculation and use cases of such coordinates are described in the separate topic.

  To support this feature the following methods have been added to engine's APIs: client_to_canvas_coords(), set_canvas_offsets(), update_canvas_offsets() and force_offsets_updating(). Also, there is a new configuration option: track_container_position.

- API to change smooth factors of the camera's movement.

  To change smooth factors of the camera's movement the following APIs have been implemented: set_plock_smooth_factor() and get_plock_smooth_factor() in mouse.js module (for Pointer Lock mode) and also set_camera_smooth_factor() and get_camera_smooth_factor() in app.js addon (for general use cases).

- New favicon picture has been added to Webplayer.

28.1.2 Changes

- New API spec for collision detection and ray casting.

  Methods append_collision_test() and append_ray_test(), as well as create_collision_sensor() and create_ray_sensor() are now have new spec, which is incompatible with the previous versions of Blend4Web engine. All developers should consider upgrading their applications to match this new behavior.

- Rendering to texture changes.

  It's now possible to render scene into texture cyclically, e.g. when two scenes render one into another. The main constraint here is the requirement to have at least one scene, into which is nothing is being rendered.

- The option Apply Default Animation is now disabled if an object already have an NLA animation attached.

  In cases when the object already have an NLA animation attached, the option Apply Default Animation is disabled to eliminate possible animation conflicts.

- Changed Hemi lamp behavior.

  If the object is being illuminated by the Hemi lamp, the Lambert shading model will allways be applied to it's materials. This is done to match the Blender's behavior.

- Support for exported-to-HTML video textures in Firefox browser.

  Since Firefox 38 an error with video textures exported to base64 was fixed, so it's now possible to use them in such browsers.

- Changed assignment of UV layers to match Blender's behavior.

  Missing from node materials UV layers are determined automatically as it's done in Blender.

- Improved stability of exported to HTML video textures.
- Optimizations of CSM shadows.
- Depth shader optimizations.
- Billboard objects optimizations.
- Configuration option resolution_factor was renamed to render_resolution_factor.
- Improved support for particle emitters which have several materials.

  Distribution of the particles to emitters as well as vertex colors inheritance is being done taking into account each material i.e mesh part.

- Changed Wind Bending inheritance on particle systems.

  If Wind Bending Inheritance property is set to Instance then Wind beinding property for emitter object is not switched of anymore.

- Updated messages about addon/engine version incompatibilities.

  For more info see version errors.

- Remove sensor locks API.

  Unused sensor locks API was removed from controls.js module.

- Fixed behavior of node materials with missing Output node.

28.1.3 Fixes

- Fixes in screenshooter.js addon.

  Fixed an error with impossibilty to take a screenshot.

- Fixed a bug in set_frame() method from nla.js module.

  Fixed set_frame() inaccurancy.

- Improved exported stability.
- Fixed a bug with addon removal/update on Windows.

  Refined binary loader in addon.

- Fixed a bug with shading from SPOT/POINT lamps.
- Fixed incorrect behavior of coordinate calculations in methods get_coords_x() and get_coords_y() from mouse.js addon.
- Fixed calculations of alpha channel in Outline effect.
- Fixed Wind Bending effect error.
- Fixed an error when particle's Scale was not taken into account on particle systems.
- Fixed synchronization error on animated EMITTER particle systems.
- Fixed a bug with shadows on billboard objects.
- Fixed incorrect exporting of Override Mesh Boundings settings.
- Fixed a bug with billboard rendering on iPad.

28.1.4 Known Issues

- Problems with updating of the add-on.

  The addons which have version 15.05 or earlier should be deleted before a newer version of addon can be installed. This is especially important on Windows. After deleting the old version, Blender must be restarted.

- NVIDIA 331 driver in Linux can cause WebGL errors.
- There is a bug with video textures on Chrome 43 for Android.

  Please update your Chrome browser to Beta or wait until the next Chrome update.

28.2 v15.05

28.2.1 New Features

- Glow effect.

  Supported an effect which occurs when the light scatters in the atmosphere and inside of the human eye and looks like a halo around glowing objects.

- The local development server runs automatically.

A new option has been added to the addon settings. This option turns on automatic start of local development server upon opening Blender. With the help of this functional web applications in development can be run without any preparations.

- Cube reflections.

  Apart from plane reflections, there are now cube reflections available. There is a new Reflection Type option available when Reflective flag is set on the object. Setting it to Cube turns this feature on.

- More NLA options supported.

  Added support of Blender's NLA tools: Scale, Muted, Reversed and Repeat. The support of these tools broadens capabilities of interactive scene developers.

  Furthermore, to control NLA through API a new nla.js module was added. This module contains methods like play(), stop(), get_frame(), set_frame() that can play/stop NLA and get/set the current frame. New methods have been involved in implementing control panel interface in the Viewer app.

- Increased rendering possibilities for sky textures.

  Influence parameters for sky texture rendering are now supported. Those parameters are: Blend, Horizon, Zenith Up, Zenith Down,"Blend", Negative, RGB to Intensity, DVar.

- In node materials, the engine now correctly processes connections between inputs and outputs of different types.

  It is now possible to connect outputs of one type with inputs of another type (Non-critical error). Now Blender's native behaviour is supported by the engine.

- For Hair particles the options on the Rotation panel are now supported.

  Now the engine fully supports the state of particles set in Blender. In particular not only location and scale are supported now, but rotation as well.

- Some demos for postprocessing effects demostration were added.

  Examples were prepared for the following effects: Bloom, Depth of Field, God Rays, Motion Blur and SSAO.

- Added a new module container.js.

  DOM tree elements can be added with a specific depth relative to the depth of the canvas element with the help of container.js's insert_to_container() method. This method replaces CSS property z-index because location depth of elements is now determined by their position in the container element.

- Improvements in the physical engine.

Margin property of physical elements and materials is now supported. This option allows for improved stability of object collision simulation. Bullet engine was updated to version 2.83.

- API for changing camera controls mode.

  Added methods of changing movement style of the camera. The example can be found in Code Snippets app in the "Camera Move Styles" section. Also has been added set_hover_pivot() method. This method allows to shift the control point of HOVER camera.

28.2.2 Changes

- Reorganized SDK's scene list.

  All the scenes in the Viewer app has been sorted by groups: App contains finished apps, Demos contains demo-scenes and examples, Tutorial Exports contains tutorials source files.

- Added syntax highlight in Code Snippets app.

  Code Snippets app now has a new design. Also it was optimized for low-resolution screens.

- API controls of video and canvas textures have been changed.

  A new parameter data_id has been added to these textures' control methods. This parameter contains an ID of dynamically loaded scene.

- Handling of animated bone excess has been changed.

  The skeletal animation now just turns off when exceeding the maximum number of bones. It resulted in shader compilation error and unstable application behaviour before.

- Some particle system properties has been renamed and now they are turned off by default.

  In particular, Hair particle system's properties Randomize Location And Size and Randomize Initial Rotation are turned off by default now.

- Doppler effect for speakers is now turned off in some browsers.

  Doppler effect support in Web Audio has been pronounced as deprecated and will be removed in Chrome starting from version 45. Other browsers still support this functionality.

- Changed objects' behavior when both skeletal and vertex animation are applied.

  If an object has both armature modifier and vertex animation applied on it, the armature modifier won't be exported.

- Rendering of particle system procedural animation (Wind Bending effect) has been optimized.

- The main .json and .bin scene files loading error handling has been improved.

- Windows 32 version of Blend4Web addon is now compiled natively.

  This feature improves addons compatibility on such systems.

28.2.3 Fixes

- The error that led to the wrong height of the description element in module "anchors.js" has been fixed.
- Support for Epiphany and other WebKit-based browsers.

  Achieved by fixing the code which works differently in the different JavaScript engines.
- Gestures on Internet Explorer 11 were disabled for Microsoft Windows touch devices.

  Previously, gestures usage (Windows Touch Gestures) was leading to unnecessary HTML-elements scaling and movement on such configurations. It is expected that correct gestures behaviour will be supported in further browser releases.
- Vertex animation with animated armature bake error was fixed.
- The error with rendering billboard objects on the iPad has been fixed.
- Node material's NLA animation applied to several objects was fixed.
- Fixed a bug related to the Motion Blur effect.

28.2.4 Known Issues

- Blender add-on upgrading problems.

  We strongly recommend not installing a new version of an addon without deleting the old one, especially on Windows. After deleting the old version, Blender must be restarted.

28.3 v15.04

28.3.1 New Features

- Deformations by using Shape Keys (Morphing).

  Added support for object's Morph targets (known as Shape Keys in Blender). To apply such keys, use the set_shape_key_value method of the geometry.js module. Simple example of how to use such functionality is given in the Code Snippets app.
- Support for Horizon Color and Zenith Color background settings.

It's now possible to tweak scene background by using the Horizon Color and Zenith Color properties as well as the Paper Sky, Blend Sky and Real Sky options directly from Blender.

- Support for the Gamma node.

  We have implemented the Gamma node back in Blender v2.74. Now this node is finally supported by Blend4Web.

- Various improvements in the Anchors tool.

  It's now possible to limit the pixel size of an annotation. Added support for dynamic loading/unloading of Anchors. Implemented the possibility to hide and show Anchors by using the show()/hide() API functions and/or by the NLA Script tool.

- Shader optimizations.

  Shader compiler improvements. Added the following features: local variables optimizations, brackets removal. Improved the performance of node materials.

- Physics engine optimizations.

  To save the load time, the physics modules are now loaded only when explicitly required. Overall size of the modules has been decreased by 20%.

- Extended tools for physics debugging.

  The new physics_stat() method has been added to the debug module. This method returns physics statistics such as the number of physics objects (separated by type), amount of geometry and other info. It is now also possible to display the number of physics iterations per second aka Physics FPS (activated in the config module).

- The new API method to attach objects to the camera independently from the camera's aspect ratio or the field of view.

  Implemented in the append_stiff_viewport() method of the constraints module.

- The new module to perform transformations: "tsr.js".

  This new module makes it possible to apply a variety of tranformations to objects by using versatile TSR vectors. Each TSR vector combines translation, scale and rotation (hence the name). These vectors may be used instead of matrices as a more convenient and effective way to apply transformations.

- The possibility to exclude any directories from being converted by the resource converter.

  To exclude some directory from being converted by the resource converter, it is enough to place a file named .b4w_no_conv to this directory.

28.3.2 Changes

- The API documentation has changed its appearance. Links for quick access to methods and properties were added.
- Skeletal animation now takes into account the relative translation of an armature and a skinned object.

  Native Blender's behavior is now supported. There is now no need to position an armature and an animated object in the same place and with the same rotation and scale.

- Independent translation, rotation and scale animations are now supported.

  The animation system no more forces keyframes to be present in every channel, which makes it possible to save original values in unused channels or change them with API.

- World background support.

  Background and sky can be enabled with the Sky Settings > Render Sky option under the World tab. It is turned off by default.

- The Uranium physics engine now consists of two modules.

  There are now two physics modules instead of one: uranium.js - the engine's code and uranium.js.mem - the file for memory initialization. Both modules must be placed in the same directory.

- The glow effect together with its related components was renamed to Outline.

  The new name better describes the principle of this effect: highlighting objects' edges.

- Limiting the camera translation using the "append_semi_stiff_cam" method of the "constraints.js" module is now possible only for the "EYE" type.
- Local development server for Blender was replaced by a new one.

  Now, instead of the standard Python SimpleHTTPServer, the Tornado web server is used as the local development server. The new server has greater performance and it also offers more options to disable browser cache.

- Keyboard controls for sliders in the Viewer application were added.

  You can now control sliders with < and > keys.

- Changes in the "update_object_animation" method of the "animation.js" module.

  The optional "force_update" parameter was added. It forces animated objects to be updated even when their animation is not being played back.

- API changes in the mouse.js module.

The enable_mouse_hover_outline and disable_mouse_hover_outline methods were added.

The following methods were declared deprecated and will be removed in the next releases: enable_mouse_hover_glow and disable_mouse_hover_glow.

- API changes in the scenes.js module.

  The outlining_is_enabled, set_outline_intensity, get_outline_intensity, apply_outline_anim, apply_outline_anim_def, clear_outline_anim, set_outline_color and get_outline_color methods were added.

  The following methods were declared deprecated and will be removed in the next releases: set_glow_intensity, get_glow_intensity, apply_glow_anim, apply_glow_anim_def, clear_glow_anim, set_glow_color and get_glow_color.

- Some settings were changed in the Object > Blend4Web panel.

  The Enable Outline option was added to enable using the outline effect on the given object. Also, the Outline on Select option was added to activate glow animation when the object is selected (previously this behavior was defined by the Selectable flag).

- New settings were added to the Scene > Blend4Web panel.

  The Enable Object Outlining option was added to control the overall possibility of outlining. Similarly, the new Enable Object Selectable option controls the overall possibility of objects' selection.

- Now the following object properties: Apply Scale, Apply Modifiers, Export Vertex Animation, Export Edited Normals and Export Shape Keys are mutually exclusive.

- API changes in modules.

  The new is_armature method was added to the API of the util.js module. It checks if the given object is of the ARMATURE type.

  The new get_parent method was added to the API of the constraints.js module. It returns the parent object of the given object.

28.3.3 Fixes

- Fixed a bug in the "anchors.js" module which caused objects' descriptions to disappear.
- Fixed a bug in the Animation Baker script that occured when there were armature objects in hidden layers.
- Fixed the camera's behavior while using "append_semi_stiff_cam" method of the "constraints.js" module.

Fixed correction of the camera's vertical axis relative to the parent object. Also the original camera orientation is now being taken into account. This can require some adjusments of the camera's rotation limits that are passed to this function.

- Fixed a bug with reloading of the playlist when it was empty.
- Fixed the buggy behavior of physical objects that occured after deleting at least one of them from the scene.
- Fixed a bug that occured when there were zero-scaled objects instanced through DupliGroups.
- Fixed a compilation error of the water shader occured on Windows and some mobile devices.
- Fixed a bug that occured when there were duplicates of animation keyframes.
- Actions from different files sharing one name can be now used for NLA animation.
- Fixed duplication of event listeners that occured when the "pointerlock" function was repeatedly called.
- Fixed behavior of the "Alpha sort" transparency type for dynamic objects.
- Fixed an add-on compilation error that occured on Windows without C++ 2010 runtime installed.
- Fixed a bug with rendering of billboard objects on iPad.

28.4 v15.03

28.4.1 New Features

- New tool for adding annotations to 3D objects.

  Now it is possible to assign anchors to empty objects. These anchors can be of three different types: "Annotations" - information from object's meta-tags is used, "Custom Element" - a custom HTML-element from the current web-page can be used as annotation, "Generic" - an invisible anchor with coordinates calculated using anchors.js module API.

- Animation and API methods for Value and RGB nodes in node materials.

  Now it is possible to animate not only Value nodes but also RGB nodes. Also, the corresponding API methods for changing such nodes were added in the objects.js module.

- New "Code Snippets" application.

  This application was created to simplify access to the examples of engine's functionality. It is also possible to look at the examples' scripts. This application can be launched from the index.html file located in the Blend4Web SDK's root directory.

- New control functions for the Glow effect.

  New APIs were added in the scenes module: get_glow_intensity() and get_glow_color().

- Improvements in the Scene Viewer.

  Design of the "Home" button was changed. A new button "All objects selectable" was added. It allows to turn off automatic "Selectable" option assignment for all scene objects. Also, it is now possible to see the total number of shaders on the loaded scene.

- Dynamic copying of scene objects (instancing).

  Now it is possible to dynamically copy and remove scene objects (to create and remove instances).

- Handling errors related to the B4W_PARALLAX Blend4Web-specific node.

  In case of incorrect usage of the B4W_PARALLAX node, an export error warning is generated.

- New options in the applications builder.

  There are now new options in the application builder: -j and -c. They add scripts and styles correspondingly to the exceptions in order to be not compiled.

- Experimental Blend4Web render engine.

  It can be turned on in the addon settings using the "Register Blend4Web render engine (Experimental)" flag. This mode is designed to simplify customization of scene properties. Also, it simplifies the interface by removing unsupported panels. At the moment, it is not possible to edit shader node tree in the Blend4Web render mode.

28.4.2 Changes

- Origin for counting off the camera limits has been changed.

  Setting camera movement limits via API now perfectly corresponds to values measured in the engine's coordinate system. Setting horizontal limits for the TARGET camera in the Blender's world space has been changed. Thus, it may require changeovers for old scenes.

- Documentation update for the camera settings.

- Horizontal and vertical limits of the camera rotation are completely independent from each other.

- Some APIs in the camera.js module were changed.

  APIs of the camera.js module have undergone a number of changes.

New methods were added: is_target_camera, is_eye_camera, is_hover_camera, rotate_camera, rotate_target_camera, rotate_eye_camera, rotate_hover_camera, get_camera_angles, hover_cam_set_translation.

The following methods were declared as deprecated and will be removed in the next releases: rotate_pivot, rotate_hover_cam, rotate, set_eye_params, get_angles, translate_hover_cam_v, set_hover_cam_angle.

The set_ortho_scale and get_ortho_scale methods now print error message when they are applied to the Orthographic camera. The behavior of the get_hover_angle_limits method was also changed. This method now returns angle limits for the HOVER camera in the [down, up] format instead of [up, down] as it was before.

- The lighting system was significantly optimized.

  Many of the lamp props are calculated at the compile time now. Now, there is no 4 lamp restriction is imposed for some mobile devices.

- The HTML layout method was changed for apps using the app module.

  Now, upon initializing an application using the app.js module, the dimensions of the created <canvas> element are completely determined by the size of the container element. Thus, if a element is used as a container, the size of <canvas> will be zero by default since div's default size is zero. You can set correct params for the container with CSS and inline-style. Also, you have to use resize_to_container() method from the app module when the container is changed. The same effect may be achieved if the autoresize option is set upon initializing the application (in the app.init() function). The low level method for changing the element's dimensions with main.resize() function is still supported.

- Now, the enable_controls() function from the app module should be called without any params.

- Deprecated API methods were removed.

  textures.js module : stop_video.

  scenes.js module: add_object, get_screen_scenes, set_light_pos, set_light_direction, set_dir_light_color, get_lights_names, remove_all, check_collision, check_ray_hit, get_appended_objs, get_object_by_empty_name.

  physics.js module: set_character_dist_to_water.

  material.js module: set_batch_param, set_max_bones, max_bones.

  main.js module: set_shaders_dir, set_texture_quality.

  data.js module: get_bpy_world.

  controls.js module: sensor_make_positive, sensor_make_negative.

camera.js module: change_eye_target_dist (deprecated MS_CONTROLS was also removed).

- There are now new conditions to allow changing object's position via API.

  The functions from the transform.js module related to changing object position can be now applied to dynamic objects only.

- You can now use TEXTURE nodes without textures.

  In this case the rendering of the node material completely corresponds to Blender.

- Updated the procedure of compatibility checks for versions of exported files and the engine itself.

  The engine will report about the scene's incompatibility by printing messages in the browser console.

- The "Do Not Batch" property was renamed to "Force Dynamic Object"

  This option instructs the engine that the object must be dynamic regardless of other settings. Now its name is more clear.

28.4.3 Fixes

- Fixed camera autorotate feature of the web player.
- Fixed an error related to the fullscreen mode in the Web Player.
- Fixed an error related to the determination of the camera's angular coordinates in some positions.
- Fixed an error with camera autorotation if the horizontal limits are enabled.
- Fixed an error when Blend4Web-specific nodes were being added multiple times to a .blend file.
- Fixed a bug with replacing materials by using the "inherit_material" function from the "material.js" module.
- Fixed an error occured while rendering reflections on an object which has been changed through the "material.js" module APIs.
- Fixed generation of the debugging wireframe spheres.
- Fixed optimization of the TEXTURE nodes in node materials.
- Fixed "Clamp" option behavior in the MixRGB (Linear Light) node.
- Fixed an export error occured when an object shares its mesh with another object and one of the following flags is set to true: "Apply Scale", "Apply Modifiers", "Export Vertex Animation" or "Export Edited Normals".
- Fixed an error with "Blend4Web > Preserve Global Orientation and Scale" option on some mobile devices.

- Fixed fog rendering error in some versions of Chrome/Firefox under Windows.

28.4.4 Known Issues

- Blender add-on upgrading problems.

  We strongly don't recommend to install the new Blender add-on version without uninstalling the previous version first (especially for Windows).

- Video textures do not work in Firefox for scenes exported as HTML files.

28.5 v15.02

28.5.1 New Features

- The local development server can be run from Blender.

  It's possible now to run local development server when using Blend4Web SDK. This server allows fast access to the Blend4Web SDK content and also make it possible to automatically open exported scenes in the Viewer application.

- Support for Spot lights shadows.

  Shadows for Spot lamps were processed in the same way as for Sun lamps. Now the calculations are performed in the same way as it's done in blender i.e the light scattering is taken into account.

- Added/improved support for "Metaball", "Surface" and "Curve" objects.

  Support for Metaball, Surface and Curve objects was added. Objects of these types are automatically converted into meshes during export. Support for Curve objects in modifiers was preserved.

- Social networks buttons are added to the Web Player.

  These buttons allow placing a link and a description to the 3D scene in one of the four popular social networks.

- Added support for editing the list of Viewer scenes directly in Blender.

  It is now possible to edit assets.json file with a list of Viewer scenes inside Blender. This works only when using Blend4Web SDK.

- Screenshots can now be taken in the Viewer application.

- New fallback_video option is added to the Web Player.

  Now the user can choose a video file to play instead of 3D content on systems without WebGL support. It is possible with the help of the new fallback_video=/path/to/video/ option.

- Improved rendering to texture functionality.

Added support for rendering scenes into several textures at a time. Scenes now can have any nesting level.

- Billboards now can save orientation and scale in world coordinates.

  To use the feature you need to set Blend4Web > Preserve global orientation and scale flag in the object's settings panel.

- Improvements on the main SDK web page.

  It is now possible to find out the version of the SDK and check the system for WebGL compatibility on the main web page.

- Added support for the Clamp flag in MATH and MIX_RGB nodes

  At first this functionality was released in Blender 2.73, and now it's also supported in the engine.

- Considerable imporovements in rendering quality on systems without depth-texture support.

  Supported rendering features on systems without depth-texture support were extended. There are such effects available now: reflections, bloom, glow, motion blur, antialiasing.

- New documentation is added.

  Added documentation for the vec3, vec4, quat, mat3, mat4 modules and for the global namespace b4w. Documentation web pages design was improved.

- Support for several engine instances on the same web page.

  Several engine instances can now work simultaniously, by specifying the namespace on engine's initialization stage.

- Possibility to use SDK on Apple OS X.

  On OS X all SDK functionality including engine and applications building, resource conversation and documentation generation is now available.

- The new set_trans_pivot() method is added to the camera module.

  This function allows setting an arbitrary position of the pivot point and the position of the camera of the Target type.

- A new ‘‘version‘‘ property is added to the ‘‘project.py‘‘ utility.

  This property allows adding a version to the scripts and styles of the compiled application.

28.5.2 Changes

- Now it is possible to add options with the same names via browser address bar.

  A new optional parameter allow_param_array is added to get_url_params() function of app module. It is set to false by

default. Setting this parameter to true leads to a merge of identical functions into a massive, other way the last one will be used.

- Improved ‘‘Background Music‘‘ speaker.

  Now the user can specify parameters of delay and playback time.

- Blend-file now includes special Blend4Web nodes by default.

  Now there's no need to add Blend4Web special nodes into a file. It is available in both SDK and addon versions of Blend4Web.

- Changed export of empty “Mesh” objects.

  Now Mesh objects without polygons are exported as Empty.

- Changed “light.js” module.

  Added function get_light_type; functions get_light_params and set_light_params now get object LAMP instead of objects name. Also the user now can change spot_blend, spot_size and distance properties of the SPOT light source through those functions.

- Improved refractions on LOW quality settings.

  New simplified (without distortion) refraction model is now used when LOW quality is chosen.

- Shader nodes optimization.
- Now automatic camera rotation can be turned off by touching touch screen.

28.5.3 Fixes

- Corrected behavior of the preloader for the Web Player.

  Some artifacts could be visible on B4W logo while opening Web Player.

- Fixed an error in rendering of one-cascaded shadows. The error was related to appearing of a hard non-lighted line on the cascade border.
- Fixed an error related to names collision while linking objects in Blender.
- Fixed an error with optimization of SDK apps.
- Fixed export error of flat shaded meshes on Linux x32.
- Fixed incorrect behavior of Target camera in particular cases.
- Fixed an error when using a shadow map with size exceeding device limits.
- Fixed an error that leads to FPS drop in Firefox 35/36 on Windows when shadows are turned ON.

28.5.4 Known Issues

- Blender add-on upgrading problems.

  We strongly don't recommend to install the new Blender add-on version without uninstalling the previous version first (especially for Windows).

- Video textures do not work in Firefox for scenes exported as HTML files.

28.6 v15.01

28.6.1 New Features

- Support for panning on touchscreen devices.

  Panning is performed by swiping two fingers on the surface of the screen.

- Support for "Text" objects.

  These objects are now automatically converted to meshes during export.

- Extended support for the NLA Script tool.

  Added new logic slots: Show Object and Hide Object used for hiding and showing the obects, Page Redirect - for redirecting to other webpages , Page Param - for storing any webpage parameter in given numerical register. Simplified usage of Select & Jump and Select & Play slots. Now it's not required to specify Selectable property on selectable objects.

- Support for high definition displays (HIDPI, Retina).

  The HIDPI mode allows to achieve considerable improvement of picture quality on devices with high resolution. This mode is activated automatically upon application startup if ULTRA quality profile has been selected. If necessary, high resolution can be turned on for other quality profiles.

- Support orthographic camera scaling.

  An API to change the Orthographic camera scale has been added (Orthographic scale in Blender).

- "autorotate" option has been added to the webplayer.

  The option autorotate is used to turn on the automatic camera rotation as soon as the scene loads.

- Simplied keyboard control mode has been added to function "enable_camera_controls.

  The mode is enabled by passing the optional parameter disable_letter_controls Thus, the keyboard controls with letter keys (WASD and so on) will be turned off.This feature can be used in cases

when you need to use the letter keys for purposes other than moving the camera.

- Support for gyroscope on mobile devices.

  To work with gyroscope on mobile devices the two sensors was implemented. The first sensor allows to operate with current device position compared with the previous one (position delta). It's created by using the function create_gyro_delta_sensor from "controls.js" module. The second sensor returns current device angle and created by create_gyro_angles_sensor from "controls.js" module. It's worth to mention that all angles are given in radians. Also the special addon "gyroscope.js" was created. This addon implements simple camera movements due to device rotation. You can find an example of using such feature in our Viewer application by selecting the "Gyroscope" menu option.

- New ''Do not Render'' property has been added to material settings.

  Enabling the property allows to hide parts of the scene objects which use such material.

- Support for video-textures on IE 11 and iPhone.

  The support is achieved by creating a new video-sequence format, *.seq. For more info check the following topic in documentation.

- Support for "title" tag in Web Player.

  The Web Player's title (shown as web browser header) is now extracted from JSON file of the loaded scene. For more info about this feature check the following topic in documentation.

- Support for meta tags in Blender.

  It's now possible to append meta tag information to scenes and objects in Blender. Possible tags for scenes are "title" and "description". Possible tags for objects are "title", "description" and "category".

- Added support for execution of user-defined functions every frame.

  To help users to create sophisticated application the new function append_loop_cb has been added to "main.js" module. This function allows to execute given callback every frame. This callback has two parameters: time since the application start and time delta between current and previous frame. Both parameters are in seconds. To cancel the callback execution every frame you should remove it by using the remove_loop_cb function from the module "main.js".

- Added support for simple preloader screen animation.

  To create an application with animated preloader pass an option "preloader_fadeout" with the value "true" to the function create_simple_preloader from "preloader.js" module.

- Added support to export converted media files to HTML file.

Now then you export HTML files it's possible to store converted files in them. To do so you need to enable "Export Converted Media" option in the export options.

- Added support for using min50 and dds textures in Web Player.

  To enable this feature pass "compressed_textures" parameter to Web Player.

28.6.2 Changes

- SDK file hierarchy was simplified.

  The external directory was removed, all its content was moved to the upper level - in the root directory of the SDK. The file with the list of the scenes used by the Viewer application is now located in apps_dev/viewer directory.

- Changed camera autorotation behavior (experimental).

  If the camera limits are present, the camera smoothly slows down when approaching the limits, then moves in the opposite direction.

- The usage of ''Special: Collision'' property was changed.

  Earlier, enabling the option automatically resulted in objects' hiding. Now, to do the same thing, you have to specify Do not Render property in the material settings.

- Changed suffix for converted media files.

  Old *.lossconv.* suffix was replaced by *.altconv.*.

- Behavior of "Do not render" object property was changed.

  Now, when the option is activated, an object's physics is not disabled. The object simply becomes invisible.

- Improved the SDK structure.

  Free and Pro SDK now come with the new and more polished examples; old and arid examples were dropped.

28.6.3 Fixes

- When using "Panoramic" camera type in Blender the camera automatically obtains "Perspective" type when exported.
- Fixed bug with "Target" camera dragging in rare cases.
- Minor fixes in the "B4W Anim Baker" addon.
- Fixed issue with sound for scenes with multiple cameras.
- Improved stability of "Timer" sensors in "controls" module.

- Fixed issue when browsing exported HTMLs in IE browser.
- Video texture optimizations. Now the video texture is not updated for suspended video playback.
- Fixed rendering issue in node materials with ‘‘REFRACTION‘‘ node.

28.6.4 Known Issues

- Blender add-on upgrading problems.

  We strongly don’t recommend to install the new Blender add-on version without uninstalling the previous version first (especially for Windows).

- Video textures do not work in Firefox for scenes exported as HTML files.
- Slow and unstable rendering of depth textures in Firefox 35.

  In various scenes the FPS is degraded when shadows are turned on. There is also an incorrect rendering of transparent materials. The issue is reported here and is to be fixed in the future browser updates.

28.7 v14.12

28.7.1 New Features

- Camera velocity settings are now available.

  Now camera movement velocity can be set up, including translation, rotation and zooming. Velocity for all camera types (Target, Hover, Eye) can be tweaked both using Blender UI and through Blend4Web API.

- Mipmapping is now supported for Canvas textures.

  We have added mipmapping support for Canvas textures.

- Full support for the “MAPPING” node.

  Now all Vector type options available for the MAPPING node are supported, including Texture, Point, Vector and Normal.

- Glow on mouse over.

  API in the mouse.js module were created for the effect of outlining the objects under the mouse pointer. Also, for purposes of controlling this effect, enable_mouse_hover_glow() and disable_mouse_hover_glow() methods were added. The objects should have Object > Blend4Web > Selectable checkbox enabled.

- A brand new app building system.

Now, the users can develop their apps right in the SDK thanks to the new project.py utility. This script makes it possible to build the apps together with the engine, to minify JavaScript and CSS files and to export the final apps to be deployed on a server.

28.7.2 Changes

- Removed support for deprecated "UV translation velocity" texture settings.

  We recommend to use node materials instead.

- Removed deprecated "Levels of Detail" user interface.

  This functionality can be used through Blender's standard "Levels of Detail" tool.

- The pointerlock.js add-on was renamed into mouse.js.

- A mouseup event is now fired when the mouse pointer is leaving the app viewport.

  So the problem with broken camera controls is no longer observed.

- Error message about "Clear parent inverse" is no more shown.

  Before, when using parenting, it was required to reset translation, rotation and scale of child objects (Object > Parent > Clear Parent Inverse). Now such transformation is natively supported by the engine.

- "Apply scale" option no longer applies modifiers.

  As before, Apply modifiers should be used in order to apply modifiers.

- Use of normal maps in node materials no longer requires a Material or Extended Material node.

  In some cases (e.g. refraction) normal maps can be used in shadeless materials.

28.7.3 Fixes

- Fixed audio playback error occured when using NLA.

  This happened due to insufficient float number precision.

- Fixed incorrect rendering of light sources on mobile devices.

- Layering shadows is fixed when multiple active light sources are present.

  Now the shadows are calculated like in Blender, that is areas illuminated by other light sources are not darkened.

- Node material rendering error has been fixed.

  The error occured when a MATERIAL node (or MATERIAL_EXT) with a linked (from another .blend file) material was used.

- Animation baker ("B4W Animation Bake" operator) no longer resets an armature pose.

  Now, when using the animation baking tool, the armature pose is being left intact.

- Fixed jerky camera movement upon application startup.
- Fixed error with incorrect determination of the camera's horizontal movement limits.
- Fixed error occured when unused textures were exported.

28.7.4 Known Issues

- Blender add-on upgrading problems.

  We strongly don't recommend to install the new Blender add-on version without uninstalling the previous version first (especially for Windows).

- Video textures do not work in Firefox for scenes exported as HTML files.

28.8 v14.11

28.8.1 New Features

- Video textures support.

  Video textures are now supported for Image or Movie textures.

- Frame rate.

  Frame rate for animation and video textures can now be changed through the Scene > Dimensions > Frame rate option.

- Canvas textures support.

  A canvas HTML element can be now used as a texture. The workflow is described in the manual.

- Camera panning.

  In the mode when the camera is rotating around a single point (Target) the users now have the ability to move the pivot point within the view plane (so called camera panning) while the right or middle mouse buttons are pressed. This function is turned on by default and can be turned off in Blender settings at need.

- New camera control mode - Hover.

  The Hover mode is now available when the camera is gliding over the horizontal plane (including zooming in and out). This camera mode makes

it possible to realize scenarios for a convenient viewing of scenes which are spread in two dimensions (rooms, game levels).

- The SDK now contains a root index.html webpage for simplifying navigation within the distribution.
- The resource converter now has the ability to convert videos.
- We have added a build system which was absent in previous public SDK distributions.
- The export Strict mode is implemented in the add-on.

  Activating Strict mode gives the possibility to display all possible errors and warnings connected with incorrect scene settings. The option is useful for final scene debugging for getting the most correct and optimized resource files.
- Audio playback support for iOS devices.

28.8.2 Changes

- The webplayer's "bg" parameter is renamed to "fallback_image".

  This option also has changed its behavior. If the fallback_image is defined the error message that WebGL is unavailable is not shown any more, instead the user sees just the image.
- If there are no sound sources in the scene the sound mute button is no longer shown in the webplayer.
- Generic materials workflow is now more predictable.
- The "mouse_down" sensor provides the code of the mouse button pressed. This code can be obtained from the payload sensor's parameter.
- Hair particle systems can be now exported significantly faster.

28.8.3 Fixes

- Normal maps now work with Generated and Normal texture coordinates.

  Using UV layers is no more required for normal maps.
- Fixed the problem with the wrong path to the physics engine in the webplayer.

  This error arose when uranium.js was moved out of the directory containing the main HTML file of the webplayer.
- In the add-on we have fixed the problem with packed textures. Export crashed when the "Automatically Pack Into .blend" option was enabled.

28.8.4 Known Issues

- Blender add-on upgrading problems.

  We strongly don't recommend to install the new Blender add-on version without uninstalling the previous version first (especially for Windows).

28.9 v14.10

28.9.1 New Features

- A new Web Player.

  The new minimalistic Web Player design blends perfectly with any 3D scenes. It has a simplified user interface and build-in help. The Web Player works on all devices including mobile ones.

- Improved shadows.

  It's now possible to choose a non-cascaded shadow model, based on a single optimized shadow map. Such model is easier to configure and suits well for relatively small scenes. For more info see the docs.

- Many NLA system improvements.

  It's now possible to create a complex logic using the Conditional Jump, Register Store, Math Operation NLA Script logic blocks and register-stored variables.

  It's now possible to use all types of supported animations in the NLA, including sound playback, vertex animation and particle emission. It's now possible to play different animation types simultaneously.

- Supported objects billboarding.

  The objects received the new set of options, allowing to configure billboarding.

- The "XYZ Euler" mode is supported for animating rotations.

  Object and skeletal animations now support the XYZ Euler mode for rotations.

- Support for the GENERATED texture coordinates.
- Support for Cross-origin resource sharing (CORS).
- Scene export process simplified.

  The range of material export errors are now not blocking the export. Instead this material will be highlighted pink at scene loading. Detailed error descriptions can be found in the manual.

- Added support for the "Do not export" option for particle systems.

- Improved stability on iOS devices.

28.9.2 Changes

- Changed SDK path setting for HTML export.

  It's now needed to set the SDK path setting for HTML export to SDK root directory. Earlier it was required to provide the full path to embed application. Pay attention, the old behavior is not supported anymore.

- Deprecated the "UV translation velocity" option.

  The option will be removed since version 14.12.

- Removed option "Do not export" from the "Object data" panel.
- Removed "Blend4Web > Animation > Cyclic" option from the object properties panel.

  Instead one should use Blend4Web > Animation > Behavior option located in the same place. Scenes with default animations may work incorrectly, so they need behavior property to be set to Cyclic.

- Modified SSAO algorythm realisation.

  The new implementation is much faster and shows better quality. The settings of the algorithm are changed too. For more info see the manual section.

28.9.3 Fixes

- Fixed rendering error for HALO materials.
- Fixed a rendering error when an object with the enabled "Force Dynamic Object" property has a parent object.
- Fixed error with keyboard shortcuts in Blender.

  Fixed error with inability to assign keyboard shortcuts in Blender for export menu items:File->Export->Blend4Web(.json) and File->Export->Blend4Web(.html).

- Fixed crash when loading textures with size exceeding device limits.
- Fixed node material errors resulting in unstable engine behavior.
- Fixed error in node materials that contained complex Node Groups.
- Fixed errors of shaders compilation on devices with mobile graphics Qualcomm Adreno 305.
- Fixed rendering error when using REFRACTION nodes in transparent meterials.
- Fixed an issue in "B4W Vertex Anim Baker" tool when current frame reset was occured after using bake.

28.9.4 Known Issues

- Blender add-on upgrading problems.

  We strongly don't recommend to install the new Blender add-on version without uninstalling the previous version first (especially for Windows).

- Normal maps don't work for Generated texture type.

  It is necessary to use UV mapping for normal maps.

28.10 v14.09

28.10.1 New Features

- ABSOLUTE type support for the MATH node.
- Support for LEVELS_OF_QUALITY special node.

  Allows to control the material's complexity depending on the quality profile which is specified by the user upon engine start.

- Support for SMOOTHSTEP special node.

  Simplifies the creation of some effects in node materials.

- Node groups support.

  Node groups allow the sharing of node blocks between materials.

- The ability to output intermediate rendering results for debugging.

  The rendering result for a certain stage can be now output above the main picture. This can be set up in the config.js module through the debug_subs options.

- The logic for controling Blender's NLA animation using a visual editor has been implemented.

  The NLA Script tool has been added to Blender's interface to allow the implementing of simple scenarios using visual blocks, for example playing an animation in response to the user actions.

- Multiple sensor system improvements.

  It is now permitted to register sensor manifolds globally using a controls module method with no connection to any object. To do this null should be passed into the corresponding API. The sensor logic is processed in a more predictable and robust way according to the sequence in which their manifolds are created. Callbacks of the user action events are now assigned using the register_<inputtype>_events() functions. To these functions it is now possible to pass the prevent_default flag which allows to unblock the standard browser behavior for the corresponding events.

- The Web Player now supports physics.

  Works only in the Web Player version where JSON files are loaded separately. Physics still not supported in the single HTML files.

- Skeletal animation mixing is now supported.

  The animation.js module now contains API for smooth transitions betweeen skeletal animations: get_skel_mix_factor() - for getting the current mixing factor value and set_skel_mix_factor() - for setting it.

- The Value node can now be animated in node materials.

  The functionality is similar to other animation types. Working in NLA is also supported.

- Diffuse and Specular lamp's properties are now supported.

- The possibility to render a transparent object above other objects on the scene.

  Activated with the Render above all checkbox for transparent materials (i.e. not Opaque).

- Scale is applied automatically to the object mesh.

  Activated by enabling the Apply scale checkbox in the object settings.

- High quality profile (including shadows, dynamic reflections and anti-aliasing) has been implemented for iOS.

28.10.2 Changes

- Shadow rendering improved.

  Shadow rendering system is significantly changed: it is now based on the Stable Cascaded Shadow Maps technique. This technique allows to greatly diminish the flickering of shadow edges when the camera moves. Smoothing is implemented between cascades. Also shadows of the last cascade fade out at distancing. Softened shadows are rendered using the Percentage Closer Shadows technique. The shadows' user settings are reworked and simplified. Now its possible to tweak the size of shadow maps, blur ratio and the setting for removing self-shadowing artefacts. The new settings are documented in detail.

- In the Web Player graphics quality settings are now saved independently for each scene.

- The behavior of the app configuration parameters has been changed: physics_uranium_path, smaa_search_texture_path and smaa_area_texture_path.

  These parameters are now calculated automatically depending on the running HTML files location, if they haven't been overriden during the app's initialization.

- Transition is completed to the system of modules which are linked via b4w.require() call.

  This also means that starting form the current version its impossible to call modules in the engine's release version using the old b4w.<module> namespaces. For compatibility purposes the ns_compat.js add-on has been implemeted, the linking of which allows to restore the old behavior.

- The Web Player's control panel can now be hidden.
- Skeletal animation is now applied to armature objects only.

  There is no need to apply skeletal animation to MESH objects. If they are linked to some animated armature, their skinning will be automatic.

- Demos and tutorials are updated according to the newly implemented features.

28.10.3 Fixes

- The preloader didn't diappear in case of a loading error (texture or sound file).
- Lagging during scaling and turning on mobile devices is fixed.
- TARGET-type camera shimmering has been removed for small turnings.
- EYE-type camera controls was fixed for mobile devices.
- The Farm demo controls are improved for Safari browser.
- Errors concerning using the unsupported shading models in node materials are now fixed.
- "Selectable" option now works for the objects without materials.
- There is no longer need to enable "Force Dynamic Object" for the objects that are animated using NLA.
- The particle system error when the object being instanced is parented to another object, has been fixed.

28.10.4 Known Issues

- Blender add-on upgrading problems.

  We strongly don't recommend to install the new Blender add-on version without uninstalling the previous version first (especially for Windows).

- Armature animation mixing doesn't work with some browsers.

  If skeletal animation mixing API brings unexpected errors, it is neccessary to override standard Math.sign function as follows:

```
var m_util  = require("util");
Math.sign = m_util.sign;
```

- genindex
- search

Symbols

A

B

C

D

E

F

G

H

I

L

M