12.5 Game Engine - Camera

Camera

- Camera
  - Default Camera
  - Camera Object
  - Parent Camera to Object
  - Parent to Vertex
  - Object as a Camera
  - Camera Lens Shift
- Camera Editing
  - Embedded Player
  - Standalone Player
  - Stereo
  - Shading
  - Performance
  - Display
- Stereo Camera

Camera

The Game Engine camera is in many ways similar to the Camera in the normal Blender Render system, and is created, parameterized and manipulated in similar ways. However because of its use as a real-time device, the Game Engine camera has a number of additional features - it may be used as not only as a static camera, but also as a moving device with its default characteristics (ie. with its own programmed moves), or it may track another object in the game. Furthermore, any game object may be used as a camera; the view is taken from the object's origin point. Lastly, it may be given special capabilities such as Stereo vision, Dome visualisation etc. which have special relevance to game technology.

When you start the Game Engine, the initial camera view is taken from the latest 3D View. This may be either a selected camera object or the default camera (see below). Thus to start the game with a particular camera, you must select the camera and press `Numpad0` before starting the Game Engine.

Tip

To avoid camera distortion

Always zoom the view in until the camera object fills the entire viewport.

Default Camera

The default camera view is taken from the latest 3D viewport view, at a distance equivalent to the viewer. This means that if the normal 3D view is active the scene does not change when the Game Engine is started.

Camera Object

The Camera object in the Game Engine follows much the same structure as the conventional Blender camera - see *Camera* for details of how to set up, manipulate and select a camera. The following sections show some of the special facilities available in BGE cameras.

Parent Camera to Object

The camera will follow the object. First select the camera and then select the object. Next `Ctrl-P` –> *Make Parent*.

Note that if your object has any rotations then the camera will also have those rotations. To avoid this use Parent to Vertex.

Parent to Vertex

The easiest way to accomplish this is to select your object and `Tab` to *Edit mode*. Now select the vertex and `Tab` back to *Object* mode.

Next, without any objects selected, select the camera and, holding the `Shift` key, select the object. `Tab` into *Edit mode*, and `Ctrl-P` and choose *Make vertex parent*.

Now the camera will follow the object and it will maintain its rotation, while the object rotates.

Object as a Camera

Any object may also become a camera with whatever properties are set for the object.

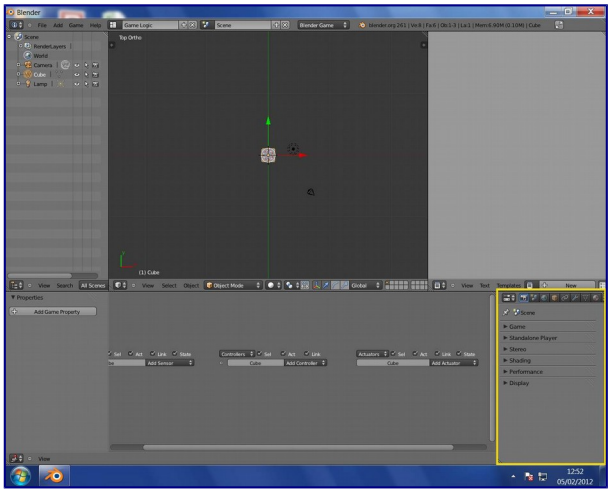
To make an object the camera, in *Object* mode select the object and press `Ctrl-Numpad0` on the numpad.

To reverse it, just select the camera and `Ctrl-Numpad0` again.

Camera Lens Shift

In the Blender interface, there is an option to shift the camera view on the x/y plane of the view. It is comparable to lens shift in video projectors that usually shift the image up along the Y axis. So for example, when you put the beamer on a table it does not project half of the image on the table.

Camera Editing



Camera Properties

The camera (or cameras) used in a Blender game have a wide-ranging effect on the way in which the game is rendered and displayed. Mostly this is controlled using the Properties panel of the camera(s) used in the game.

Tip

Render Engine

Make sure that the render engine is set to Blender Game when attempting to set these controls - otherwise this

description will not tally with what you see!

In the Camera Properties area, there are six panels available, as shown. Each can be expanded or contracted using the usual triangle button. The features in each panel will be described in detail below.

Embedded Player

Game Panel

Start button - Start the Game Engine. Shortcut P.

Standalone Player

Standalone Panel

This panel provides information for the Standalone Game Player which allows games to be run without Blender. See *Standalone Player* for further details.

Fullscreen -
: Off - opens standalone game as a new window. On - opens standalone game in full screen.

Resolution

X
: Sets the X size of the viewport for full-screen display.

Y
: Sets the Y size of the viewport for full-screen display.

Quality

Bit Depth
: Number of bits used to represent color of each pixel in full-screen display.

FPS
: Number of frames per second of full-screen display.

Framing

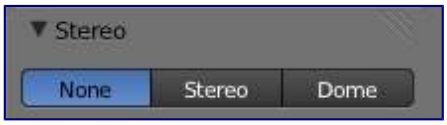
Shows how the display is to be fitted in to the viewport.

Letterbox
: Show the entire viewport in the display window, and fill the remainder with the “bar” color.

Extend
Show the whole display in the viewport, and fill the remainder with bars.
Scale
Scale the display in X and Y to exactly fill the entire viewport.

Bar Color
Select a color to use as the color of bars around the viewport (default black). To use this, select a color mode (RGB, HSV or Hex), then use the color slider and color wheel to choose a bar color.

Stereo



Stereo Panel

Select a stereo mode that will be used to capture stereo images of the game (and also, by implication, that stereo displays will use to render images in the standalone player).

None
Render single images with no stereo.
Stereo
Render dual images for stereo viewing using appropriate equipment. See *Stereo Camera* for full details of available options.
Dome
Provides facilities for an immersive dome environment in which to view the game. See *Dome Camera* for full details of available options.

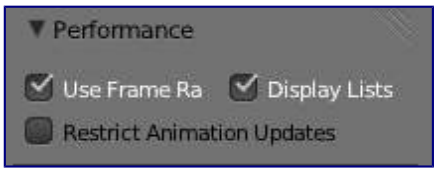
Shading

Shading Panel

Specifies the shading mode to be used in rendering the game. The shading facilities available in Blender for use in *Materials* and *Textures* are essentially the same in the Blender Game Engine. However the constraints of real-time display mean that only some of the facilities are available.

Single Texture
Use single texture facilities.
Multitexture
Use Multitexture shading.
GLSL
Use GLSL shading. GLSL should be used whenever possible for real-time image rendering.

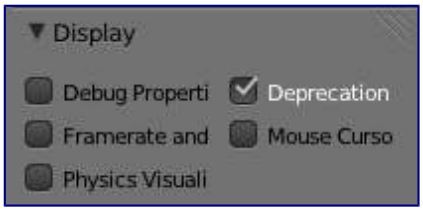
Performance



Performance Panel

Use Frame Rate
Respect the frame rate rather than rendering as many frames as possible.
Display Lists
Use display lists to speed up rendering by keeping geometry on the GPU.
Restrict Animation Updates
Restrict number of animation updates to the animation FPS (this is better for performance but can cause issues with smooth playback).

Display



Display Panel

Gives various display options when running the Game Engine. Under the...

Debug Properties
Show properties marked for debugging while game runs. Note that debug properties to be shown must be requested at source (eg. i-button in state tables). Only available when game is run within Blender - not in standalone player version.
Framerate and Profile
Show framerate and profiling information while game runs. Only available when game is run within Blender - not in standalone player version.
Physics Visualization
Show physics bounds and interactions while game runs (available in both Blender and standalone versions).
Deprecation Warnings
Print warnings when using deprecated features in the python API. Only available when game is run within Blender - not in standalone player version.
Mouse Cursor
Show mouse cursor while game runs (available in both Blender and standalone versions).

Stereo Camera

Stereo Cameras allow you to generate images that appear three dimensional when wearing special glasses. This is achieved by rendering two separate images from cameras that are a small distance apart from each other, simulating how our own eyes see. When viewing a stereo image, one eye is limited to seeing one of the images,

and the other eye sees the second image. Our brain is able to merge these together, making it appear that we are looking at a 3d object rather than a flat image. See Stereoscopy for more information on different stereoscopic viewing methods.

Stereo Settings

Stereo Mode

Set the type of stereo camera to use. Possible modes are detailed below.

Eye Separation
: This value is extremely important. It determines how far apart the two image-capturing cameras are, and thus how “deep” the scene appears. Too small a value and the image appears flat; too high a value can result in headaches and eye strain. The ideal value mimics the separation of the viewer’s two eyes.

Stereo Modes

Specifies the way in which the left-eye image and the right-eye image pixels are put together during rendering. This must be selected according to the type of apparatus available to display the appropriate images to the viewer’s eyes.

Anaglyph
: One frame is displayed with both images color encoded with red-blue filters. This mode only requires glasses with color filters, there are no special requirements for the display screen and GPU.

Quad Buffer
: Uses double buffering with a buffer for each eye, totaling four buffers (Left Front, Left Back, Right Front and Right Back), allowing to swap the buffers for both eyes in sync. See Quad Buffering for more information.

Side by Side
: Lines are displayed one after the other, so providing the two images in two frames side by side.

Above-Below
: Frames are displayed one after the other, so providing the two images in two frames, one above the other.

Interlaced
: One frame is displayed with the two images on alternate lines of the display.

Vinterlaced
: One frame is displayed with both images displayed on alternate columns of the display. This works with some ‘autostereo displays’.

3D Tv Top-Bottom
: One frame displays the left image above and the right image below. The images are squashed vertically to fit. This mode is designed for passive 3D TV.

Dome Camera

This feature allows artists to visualize their interactive projects within an immersive dome environment. In order to make it an extensible tool, we are supporting Fulldome, Truncated domes (front and rear), Planetariums and domes with spherical mirrors.

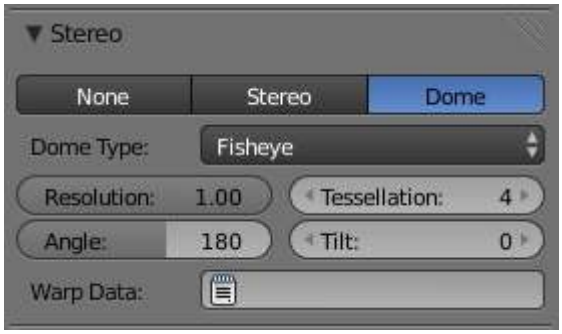
The Dome camera uses a multipass texture algorithm as developed by Paul Bourke and was implemented by Dalai Felinto with sponsorship from **SAT -** Society for Arts and Technology within the **SAT Metalab** immersion research program, that involves rendering the scene 4 times and placing the subsequent images onto a mesh designed especially such that the result, when viewed with an orthographic camera, is a fisheye projection.

Note

Remember to use Blender in **fullscreen mode** to get the maximum out of your projector.

To accomplish that launch Blender with the command-line argument -W. Also to get away of the top menu on Blender try to join all windows (buttons, 3dview, text, ...) in a single one. Otherwise if you only maximize it (Ctrl+Up) you can’t get the whole screen free to run your game (the top bar menu takes about 20 pixels).

Dome Camera Settings



Dome Type
: This menu allows you to select which type of dome camera to use. They are outlined below, along with their respective settings.

- Fisheye Mode
- Front-Truncated Dome Mode
- Rear-Truncated Dome Mode
- Cube Map Mode
- Spherical Panoramic Mode

Available camera settings change depending on the selected Dome Type:

Resolution
: Sets the resolution of the Buffer. Decreasing this value increases speed, but decreases quality.

Tesselation
: 4 is the default. This is the tesselation level of the mesh. (Not available in Cube Map mode).

Angle
: Sets the field of view of the dome in degrees, from 90 to 250. (Available in Fisheye and Truncated modes).

Tilt

Set the camera rotation in the horizontal axis. Available in Fisheye and Truncated modes).

Warp Data Mesh

Use a custom warp mesh data file.

Fisheye Mode

An Orthogonal Fisheye view from 90º to 250º degrees.

- From 90º to 180º we are using 4 renders.
- From 181º to 250º we are using 5 renders.

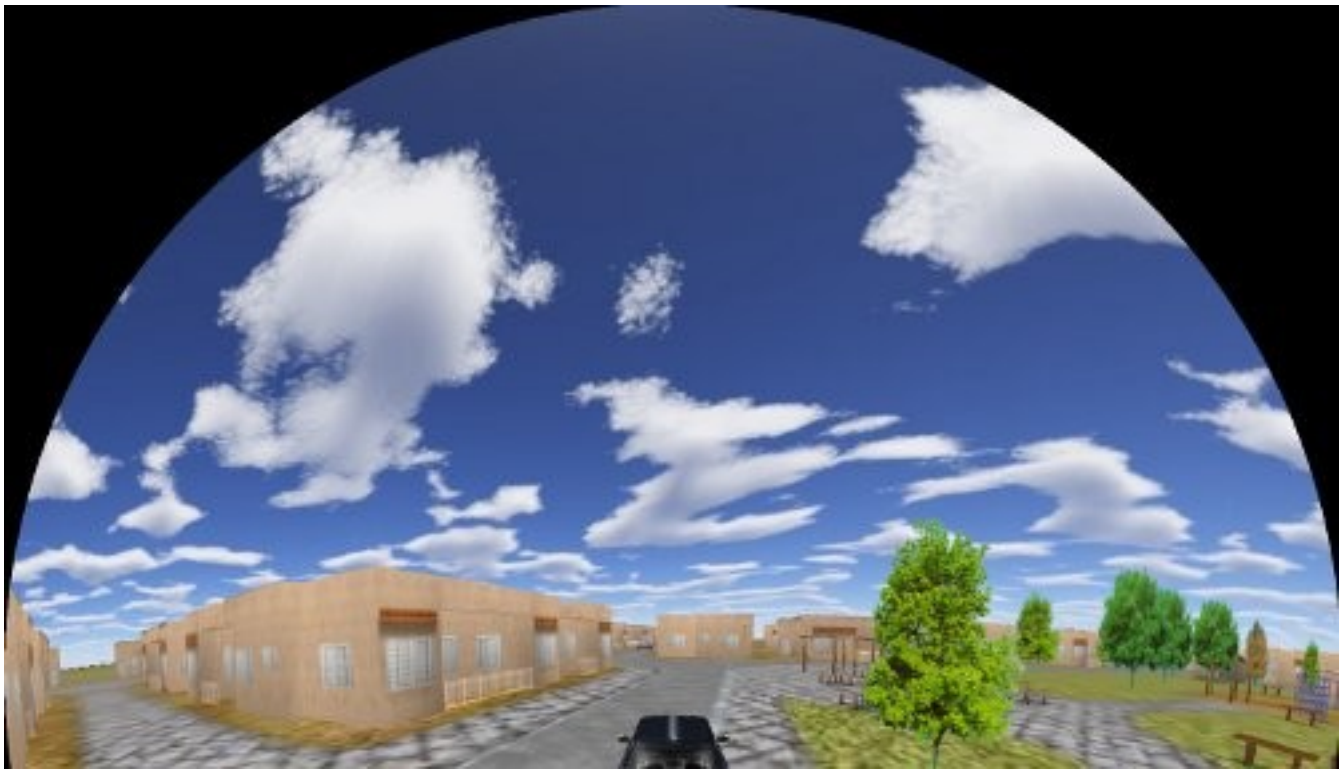
Fisheye Mode

Front-Truncated Dome Mode

Designed for truncated domes, this mode aligns the fisheye image with the top of the window while touching

the sides.

- The Field of view goes from 90º to 250º degrees.
- From 90º to 180º we are using 4 renders.
- From 181º to 250º we are using 5 renders.

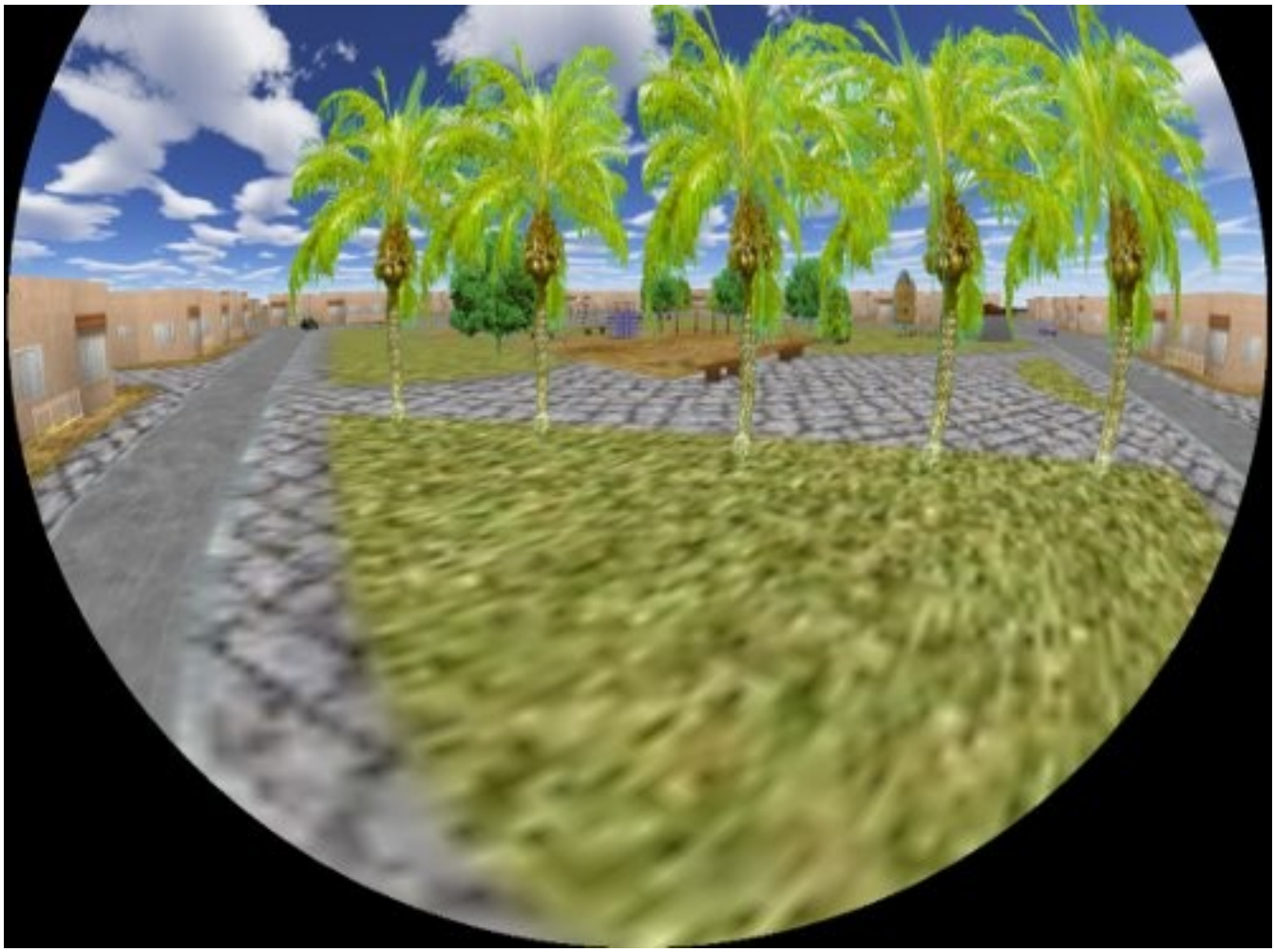


Front Truncated Dome Mode

Rear-Truncated Dome Mode

Designed for truncated domes, this mode aligns the fisheye image with the bottom of the window while touching the sides.

- The Field of view goes from 90º to 250º degrees.
- From 90º to 180º we are using 4 renders.
- From 181º to 250º we are using 5 renders.

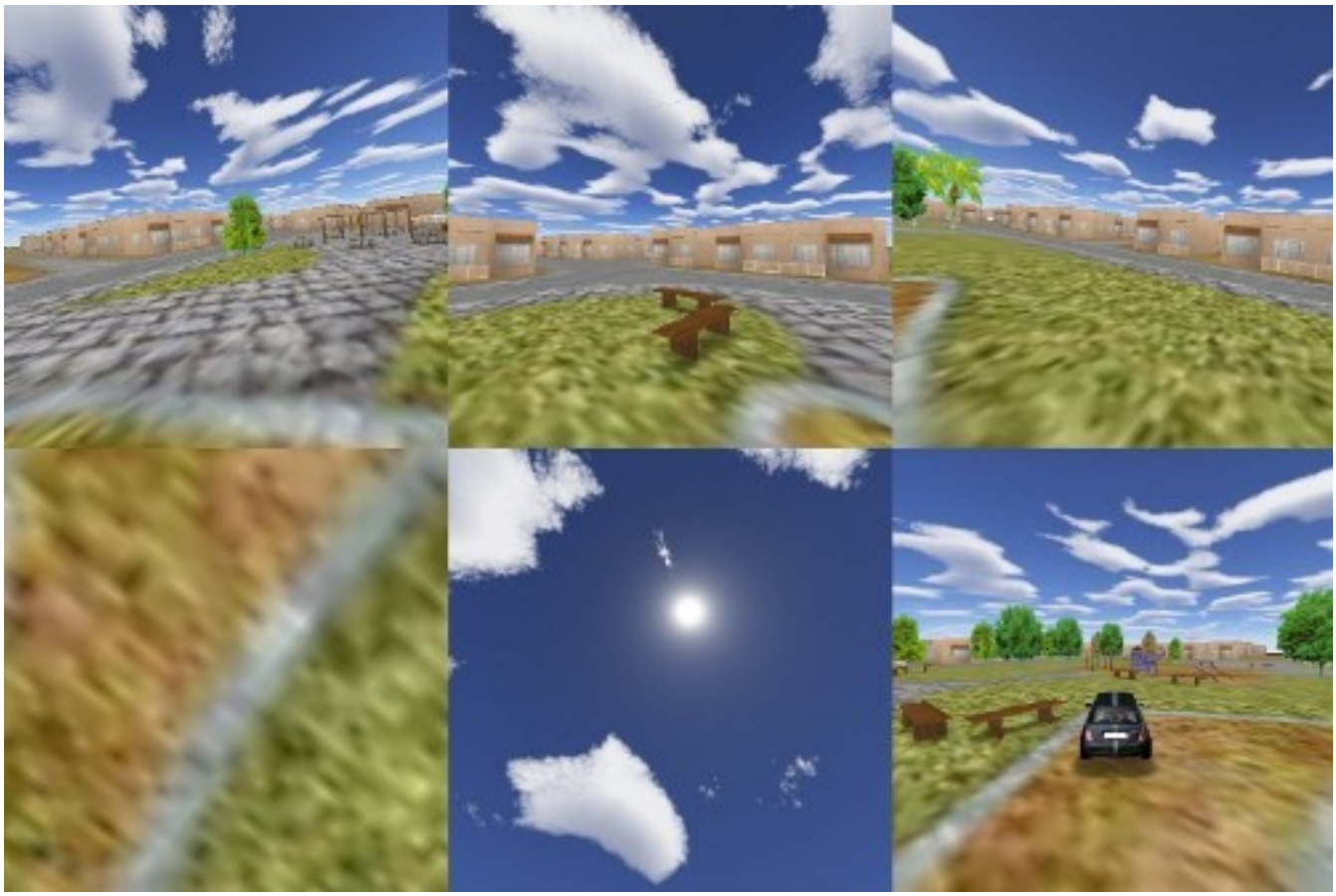


Rear Truncated Dome Mode

Cube Map Mode

Cube Map mode can be used for pre-generate animated images for CubeMaps.

- We are using 6 renders for that. The order of the images follows Blender internal EnvMap file format: - first line: right, back, left - second line: bottom, top, front



Environment Map Mode

Spherical Panoramic Mode

A full spherical panoramic mode.

- We are using 6 cameras here.
- The bottom and top start to get precision with **Definition** set to 5 or more.

Warp Data Mesh

Many projection environments require images that are not simple perspective projections that are the norm for flat screen displays. Examples include geometry correction for cylindrical displays and some new methods of projecting into planetarium domes or upright domes intended for VR.

For more information on the mesh format see Paul Bourke’s article.

In order to produce that images, we are using a specific file format.

File template:

```
mode
width height
n0_x n0_y n0_u n0_v n0_i
n1_x n1_y n1_u n1_v n1_i
n2_x n1_y n2_u n2_v n2_i
n3_x n3_y n3_u n3_v n3_i
(...)
```

First line is the image type the mesh is support to be applied to: **2** = **rectangular**, **1** = **radial** Next line has the mesh dimensions in pixelsRest of the lines are the nodes of the mesh.

Each line is compund of **x y u v i** (x,y) are the normalised screen coordinates(u,v) texture coordinatesi a multiplicative intensity factor

x varies from -screen aspect to screen aspecty varies from -1 to 1u and v vary from 0 to 1i ranges from 0 to 1, if negative don’t draw that mesh node

- You need to create the file and add it to the Text Editor in order to select it as your Warp Mesh data file.
- Open the Text Editor (Window Types/Text Editor).
- Open your mesh data file(ie. myDome.data) in the text editor (Text/Open or Alt O on keyboard).
- Go to Game Framing Settings (Window Types/Buttons Window/Scene)
- Enable Dome Mode.
- Type filename in Warp Data field(ie. myDome.data).

To create your own Warp Meshes an interactive tool called meshmapper is available as part of Paul Bourke’s Warpplayer software package(requires full version).

Example files

Spherical Mirror Dome 4x3, Truncated Dome 4x3, Sample Fullscreen File 4x3, Sample Fullbuffer File 4x3.

Note

Important: the viewport is calculated using the ratio of canvas width by canvas height. Therefore different screen sizes will require different warp mesh files. Also in order to get the correct ratio of your projector you need to use Blender in Fullscreen mode.