

Mesh editor for Construct

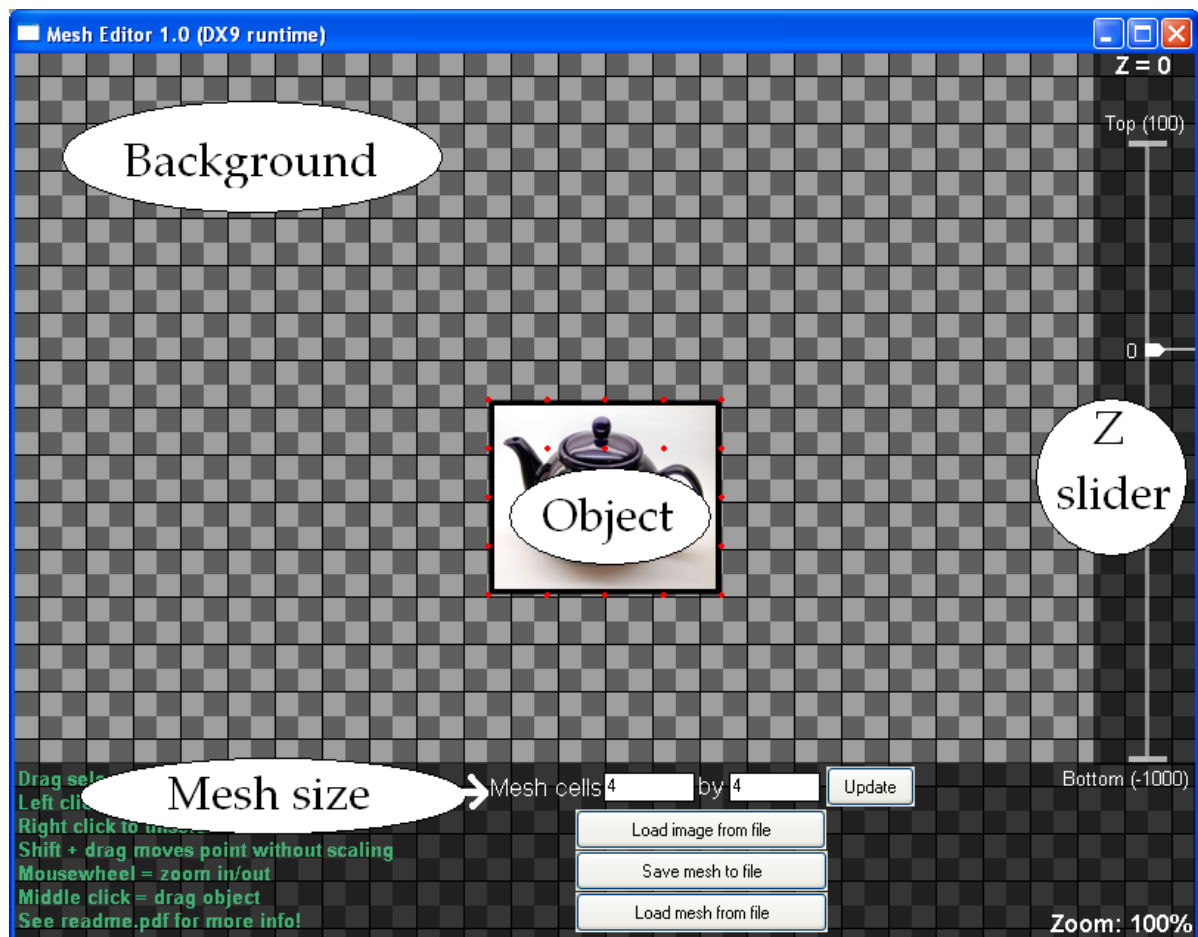
Version 1.0 (9th April 2009)

Overview

The Mesh Editor allows you to visually design meshes for the 'Mesh Distort' feature of sprites in Construct. You can create a custom number of *vertices* (points) in a grid over the object, then distort it by dragging these points. You can also move them on the Z axis, allowing a simple kind of 3D modelling.

These meshes can then be saved to disk (as .mesh files). You can use the 'Load mesh from file' action to load them in to your game.

Interface



Background

The background consists of grey tiles to show transparency and opacity. There is a black grid at $Z = 0$ for reference.

Object

The sprite object being edited.

Z slider

This slider can be dragged while vertices are selected to change their Z height. The

scale is from 100 (high up, by the camera) through 0 (where the layout lies normally), to -1000 (far away in the distance).

Mesh size

Enter a number of rows and columns for the mesh size and click **Update** to change the number of vertices. Clicking **Update** resets the mesh, undoing your changes.

Performance note

You can do more interesting effect with more vertices, but the more vertices you use, the slower the object is to render. A good rule of thumb is four vertices are used to render ordinary sprites. So if you have 40 vertices, your mesh will probably take the same time to render as ten sprites. Remember it's only rendered if it's on the screen! Off-screen sprites and meshes skip rendering, taking no time at all.

Moving the object and zooming

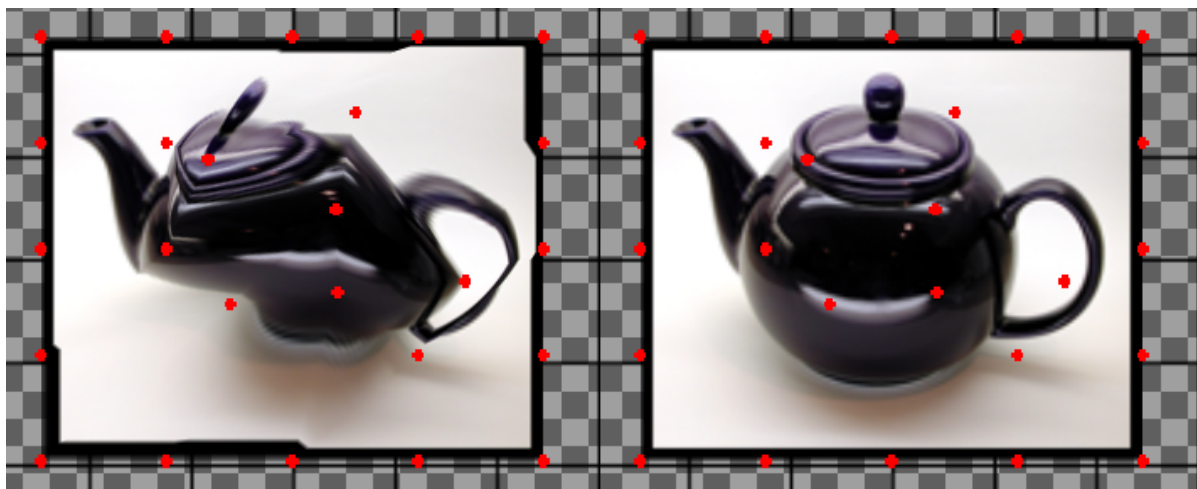
You can hold the middle mouse button to drag around the object being edited. This is especially useful when zooming, or checking how it looks in 3D. The object deliberately lags slightly behind the mouse, which makes the motion look smoother with V-syncing.

Scroll the mouse wheel to zoom in and out. Hit **0** on the num pad to reset zoom to 100%. The zoom level is displayed in the lower right corner.

Selecting and moving vertices

Click on a vertex to select it. Vertices turn **yellow** when they are selected. You can also click and drag to create a selection box and select a large number of vertices. You can hide and show vertices with the **H** key.

Selected vertices can be dragged with the left mouse button to change their position and distort the object. If you drag while holding **shift**, the vertex changes position, but without distorting the object. You can then drag it again without holding shift, to distort it from its current position. This is useful for creating effects not based on a strict grid, especially when elevating points in 3D.



Left: an object with moved vertices, distorting the object.

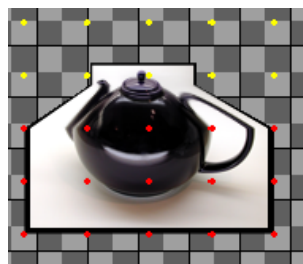
*Right: the same vertices moved while holding **shift**. The vertices move without distorting. They can now be moved again without holding shift to distort from those positions.*

Right click anywhere to unselect all vertices. They will all turn red again.

Using the Z slider

When vertices are selected (showing in yellow), drag the white slider on the Z slider to change their position in 3D. The slider ranges from 100 (at the camera) through 0 (no Z elevation) to -1000 (off in the distance). Nothing can move more distant than -1000, because Construct clips (hides) anything beyond that distance.

Tip: it can be hard to judge how the 3D mesh looks unless you drag the object around the screen with the middle mouse button. Since you'll be viewing it at different angles doing this, you can get a feel for the shape of the object. Zooming in can help you inspect 3D parts of the image closely.



Example of some vertices moved in to the distance with the Z slider.

Don't forget to enable **3D layering** on the object's layer, if you change any of the Z positions. Otherwise, the object may not render correctly at certain angles.

Viewing the mesh wireframe

You can press the following keys to show different aspects of the mesh:

H (toggle vertices)

Toggle displaying the red/yellow vertices. Useful to see the texture unobscured. Vertices are shown by default.

V (toggle depth lines)

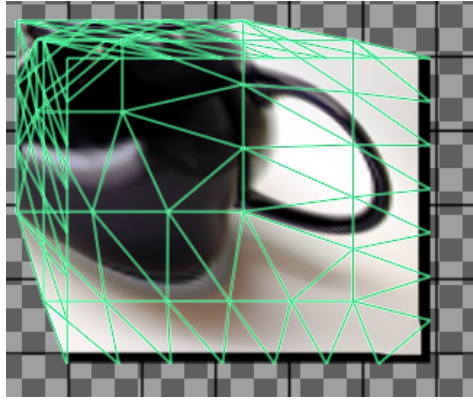
Toggle displaying vertical lines showing the height/depth of vertices. A light blue line is drawn from the vertex to its Z height. Since the vertices are always shown at $Z = 0$, this is useful to link a point on a 3D texture to the vertex if you need to move it. Depth lines are hidden by default.

M (toggle mesh)

Toggle showing the full mesh wireframe with green lines. This is off by default. The mesh shows you exactly how the texture is deformed. It can be useful also to see the individual triangles. This helps, for example, if you want to make sure you aren't moving a vertex to cross over another triangle, which often looks wrong.

T (toggle texture)

The texture is shown by default. You can hide the object image if you just want to see the mesh or vertices alone without a texture. The background turns black with this option since you don't need to see opacity or transparency.



Example of an object with vertices and depth lines hidden, but texture and mesh visible.

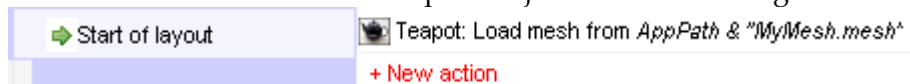
Using the mesh in a game

Since your game probably does not involve distorted teapots, you can load in a custom image by clicking **Load image from file**. Note this resets the vertices to a flat grid.

When you are done, click **Save mesh to file** to open a save dialog. Meshes are saved to **.mesh** files.

You can click **Load mesh from file** to open an old mesh you saved. There are some example meshes in the Mesh Editor folder you can also try loading.

Once your mesh is finished and saved to disk, you can now load it in to your game, using the **Load mesh from file** action in the Sprite object. Your event might look like this:



Remember the **AppPath** expression returns the current application directory. It's a good idea to keep all your game files in the game folder or subfolders.

Construct does not care what the image looks like when saving and loading meshes. It does not save the texture. If you design a mesh on the teapot, then load it on to a piece of terrain, the same mesh is stretched over the new object. For this reason, you probably want to design your meshes with the intended object image loaded!

Note about collisions

As of Construct 0.99, mesh distorted sprites do not respond to collisions differently. They still collide as if they were not mesh distorted at all. This is a known limitation and may be corrected in a future version of Construct.