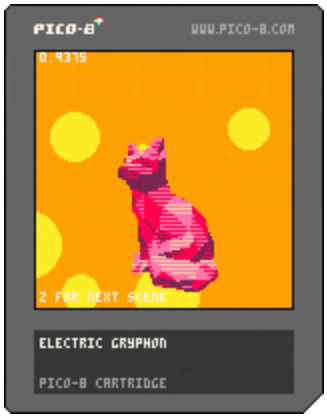
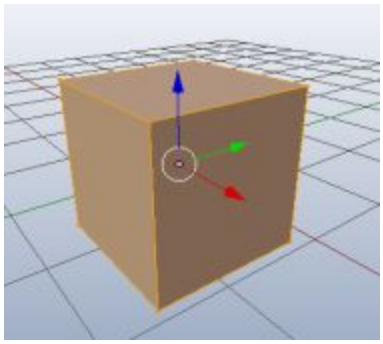
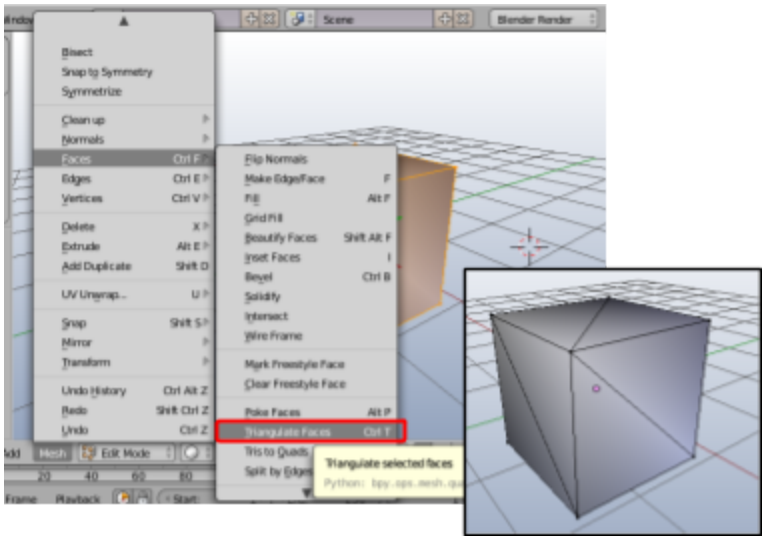
Blender Object Conversion for Gryphon 3D Engine Lib V2

3D Engine File: http://www.lexaloffle.com/bbs/?tid=28077 **Python script:** http://s000.tinyupload.com/index.php?file_id=56902566375777023329 **Python Download:** https://www.python.org/downloads/ (Download version 2.7) **Blender Download:** https://www.blender.org/download/	

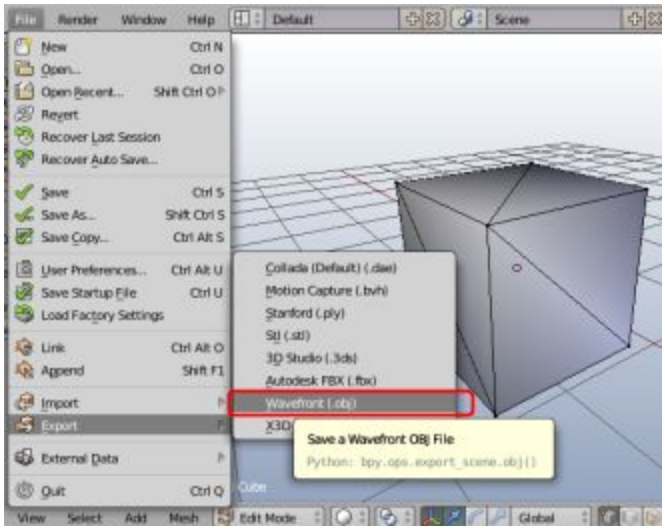
1. Load model in Blender (I'm using Version 2.72b)
2. Enter object edit mode
3. Select All Faces <a>



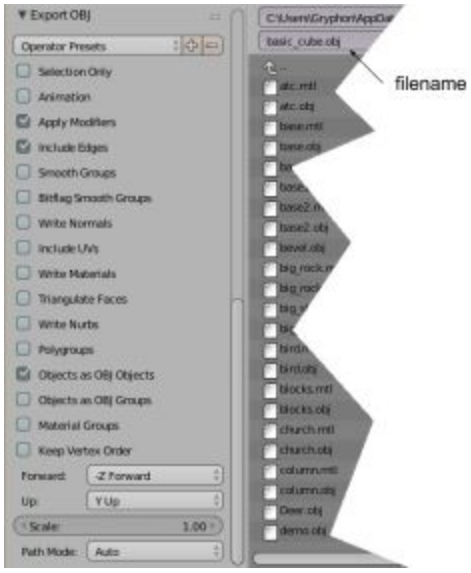
4. Select Mesh->Faces->Triangulate Faces
   Faces need to be triangulated. The engine doesn't handle quads right now.



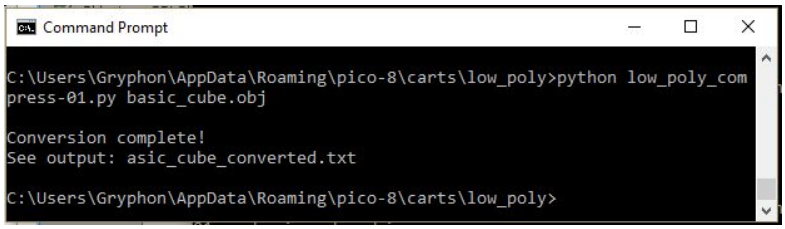
5. Select File->Export->Wavefront (.obj)



6. Set export settings as shown:



7. Place python script in same folder as file.
8. Open a Command Prompt window in that folder.
9. Run: python low_poly_compress-01.py <filename>.obj
   (This is for Python 2.7)



10. File will look like this when opened in a text editor:

basic_cube.converted.txt

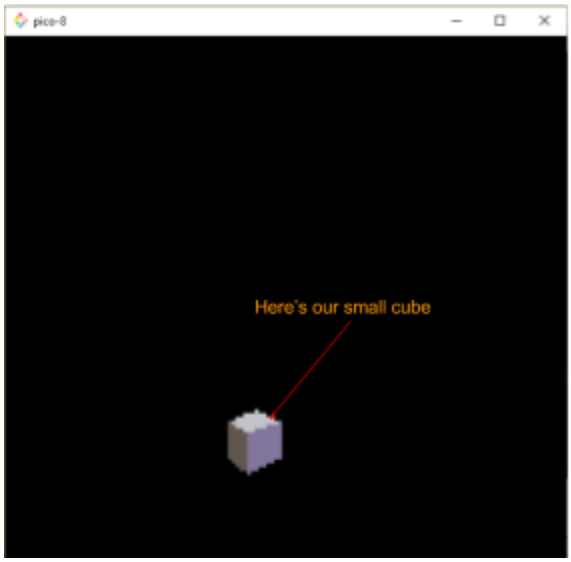
```
model_v="0100ff00ff000100ff000100ff00ff000100ff00ff00ff0001000100ff0000ff0100
0100ff0001000100ff000100ff00"
model_f="020304080706010506020607070804010408010204050806020106030207030704050
0108"
```

11. Copy these two strings to your Pico 8 file.
12. Example main code section:

```
...
----------------------------------end copy------------------------------
----------------------------------electric gryphon's 3d library---------



model_v="0100ff00ff000100ff000100ff00ff000100ff00ff00ff0001000100ff0000ff0100
0100ff0001000100ff000100ff00"
model_f="020304080706010506020607070804010408010204050806020106030207030704050
0108"

function _init()
```

```
	cur_frame=0
	init_3d() --need to call init_3d() to set up player, camera and lights

basic_cube=load_object(read_vector_string(model_v),read_face_string(model_f),
x,0,z,0,0,0,false,k_colorize_static,13)
end


function _update()
	handle_buttons() -- handle default buttons for player-- this can be
overwritten obviously.
	update_player() -- update the player with default movement, stopping at
obstacles
	update_camera() -- update the camera based on player location and
direction
end



function _draw()
	cls()
	update_3d() -- call update_3d() at the end of the _update() function to
transform etc.
	draw_3d() --render objects into triangles, sort the triangles and draw
them onto the screen
end
```

13. Pico-8 output



With the default player position and a 2x2x2 cube, the model looks a bit small, but here it is.