Display\_Name..

Computer Programming Using Kivy - **OpenGL 1 - Subclassing a 3D Engine**

**GOAL: Use a 3D Engine to load a 3D mesh (OBJ file) of your choice.**

OpenGL is an open standard for graphics rendering. Since it is an open standard, any programming language (such as python), framework (such as Kivy), or platform (such as various smart phone operating systems) can use it. It is open, as in the specification is public and anyone can use it for free, and a GL as in a graphics library. OpenGL only defines ways for graphics to be sent to the video card and displayed, so the programmer is responsible to load 3D objects and place the vertices in 3D space using their arrangement and their current location, rotation, and scale—but thankfully, expertmm has created a 3d engine based on nskrypnik’s work and Kivy’s OBJ file loader so that you don’t have to write all that geometry for each vertex of a mesh (for simple object loading without game engine you can use ddd from Kivy garden).

Find a 3D model (must be in OBJ format for Kivy):

* File Explorer, This PC, Resources, Meshes. For KivyGlops you’ll need an OBJ file, and you’ll need the entire filename to put in your code. You can tell what type of file it is, and the full filename, if you right-click the 3D Object file, then click Properties, then the Details tab (the first field, name, will have the full filename including extension such as something.obj)
* Pick any file, select both the obj file & any files with the same name or subfolder (for example, something.obj may require other files such as something.mtl and texture.png in order to have the correct color and texture), right-click them, click Copy
* Go to File Explorer, This PC, your home drive (T:\*username*, H:), File, New Folder, & name it opengl
* Double-click your opengl folder, then hit Ctrl V (or press Alt key, the click Edit, Paste)

Copy the example program, OBJ loader, and shader to your opengl folder:

* File Explorer, This PC, Resources, R:\Classes\ComputerProgramming\Examples\KivyGlops
* Hit Ctrl A or Edit, Select All , so that all of the files are highlighted
* File Explorer, This PC, home drive (such as T:\*username* or H:), then double-click your opengl folder
* Press Ctrl V (or click Edit, Paste)

Edit your code to use the model you chose:

* Open Geany
* File, Open, your home drive, opengl, then choose kivyglopsexample.py
* File, Save As, rename the file opengl1.py -- Before editing, try the example code: Click Execute – when you are sure it is working (shows example scene or black screen), close it.
* Find or implement the load\_glops method (in the KivyGlops subclass such as MainScene)--remove the line that says pass if any—load your obj file in the load\_glops method (using code below), but change the ".obj" file in quotes in the example below to instead use whatever model file you chose in quotes (including .obj):



Execute the program, and you should be able to move the mouse to look around and see your object!

You can use the “asdw” keys to walk around.

|  |
| --- |
| TIPS:  \* If says appended glop in the black console window for each object in the obj file, it worked.  \* If says “Cannot upload more than 65535 indices,” try simpler model (or in Blender, apply a Decimate modifier using Blender then File, Export, Wavefront OBJ, your home drive such as H: or your username on left, then add “decimated” to the filename in the second box, then change your code to use the name of the new file you just saved).  **Missing Texture Issues**:  \* If you created the OBJ file from exporting from a 3d program yourself, make sure you export to the same folder as the texture file, and make sure the texture file is a png or jpg file.  \* If you get an error that says “Cannot seek in this data source” in “pygame.image.load” make sure you copy any image files that are used as textures for your 3d mesh to your opengl folder.  \* If you still get the error or the object is all white, open Notepad++, open the mtl file, and change any absolute paths to relative ones (remove the backslash and anything before it in the path (don’t remove map commands such as map\_Kd)  \* If you have multiple textures, you’ll have to bake them first (for Blender see <http://wiki.blender.org/index.php/Doc:2.6/Manual/Render/Bake> , then Image, Save As Image) then export again as Wavefront OBJ.  **Holes in Object**: If there are triangle-shaped holes in the object, make sure you have an updated version of KivyGlops, otherwise you can open the object in a 3D program and apply Triangulate (or decimate or other feature that makes all faces *tris*) to the object then Export it as Wavefront OBJ to your opengl folder with a new name. |