

---

# COMPUTER GRAPHICS

## ASSIGNMENT #3

### OPENGL MODELER

---

#### INTRODUCTION

---

In this assignment you will replace your software renderer with an OpenGL version.

#### REQUIREMENTS

---

Your assignment is to use OpenGL for rendering. Depending on the current state of your software, there are two approaches you may follow:

- Change the (internal) implementation of the renderer so it will use OpenGL.
- Discard the renderer and let the scene handle calls to OpenGL directly.

You can choose either option. Here are a few implementation hints you may consider:

- There is an OpenGL demo in Course Material -> Tutorial Slides, that contains basic shaders and can help you start working with OpenGL. To use it, just replace the files in the skeleton with the OpenGL demo files.
- When a model is first created, you should create a vertex buffer object for the model and use the handle to the model. When the model should be drawn, the model will pass only the handle – not the geometry – to the renderer.
- Shader programs can be managed by the scene. For example, the scene might initialize several programs when the application starts (or ask the renderer to do so). Then you can use the handles to the shaders.
- If you discard the renderer, all calls to the renderer should be replaced with calls to OpenGL.
- Some of the states the renderer handled previously may now need to be handled by the scene.

In addition to implementing all existing features (of the first two assignments) in OpenGL, you should also implement the following using shaders:

- Texture mapping. Use texture-coordinates when available, and additionally implement two canonical texture-coordinates (for example, projection on a plane).  
To combine the shading method with texture, use the shading method (flat, gouraud, phong) to compute the illumination at each point, then multiply by the color that is given by the texture.  
To load textures you can use any external image library. We recommend:

- pngLib. Can be found on the course site. Just include the header files and link the libraries.
- Cimg. Found here: <https://cimg.eu/>. This is just a *huge* header file. There is a short explanation on the site and it is very easy to use

You can find meshes with nice texture on the course site. There are also additional meshes here: <https://www.cs.cmu.edu/~kmcrae/Projects/ModelRepository/> (Spot or Bob are recommended) and here are nice textures with normal maps (most require local frame): <https://3dtextures.me/>.

- Normal mapping.
- Environment mapping.
- Toon shading that includes silhouette rendering.
- Color animation – change the color of the model gradually over time, in some smooth manner. For example, you can use HSV representation to change the hue of each color in the model. Create two such color changing effects of your choice and provide a selection between the two in the UI.

To implement the animation, you can re-render the scene automatically every time interval using the functions `glutPostRedisplay` and `glutTimerFunc`.

- Vertex animation – Do something similar to the above with the vertices. Namely, move the vertices gradually over time in a non-uniform way.
- Implement a texture that uses the turbulence function (generated directly using the coordinates in object space, without using UV coordinates). You may implement anything from marble to wood as we learned in the tutorial.

Note that you are not required to implement all the extra features (blur, light bloom, fog effect) with OpenGL.

---

## SUBMISSION

---

Submission is frontal. Before the submission deadline, we will schedule timeslots for you to come and see us. Presentations will last 15-20 minutes, during which you will show us your work and answer our questions.

---

## FINAL NOTES

---

- This is not MATAM – there is no automatic checker. This means that all the features that are to be implemented should be intuitive to the developers with plenty of room for personal interpretation. It also means that the features that you implement should behave quite differently compared to other works. Copying of any kind will not be tolerated!
- DO NOT USE any external code without permission. If you have any doubt, please ask us.
- You are very much encouraged to experiment with your program and add more features to it. Previous experiences show this assignment can be addicting!
- You have three weeks to complete this exercise, this time is more than enough, but you are strongly encouraged to start working on it right away.