

## Assignment 3

### How To Run:

Cd to the directory that contains the source files, and launch main.py with python

```
cd /path/to/3D
python main.py
```

### Prerequisites

The GUI automatically loads the monkey.txt model upon start from ./data/monkey.txt. Ensure this model stays present and in that directory

### Implementation

The 3D image generator was implemented with a GUI that allows maximum customization of the scene before the image is generated.

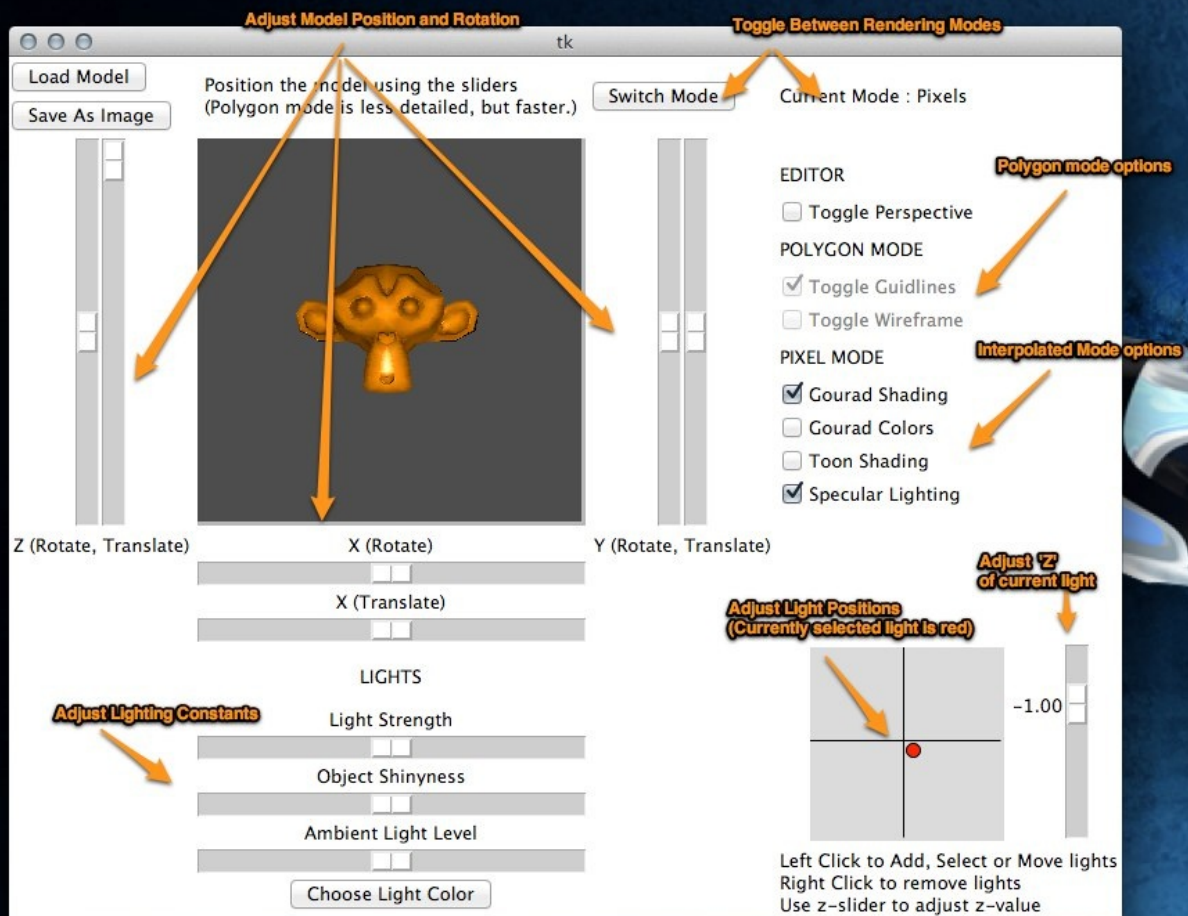
#### Key GUI Features

- Allows you to load models
- Allows you to export images
- Has two visual modes (Fast polygon mode, and high-quality interpolated mode.)
- Provides extensive controls to manipulate the scene, including:
  - X,Y and Z Rotation and Translation
  - Allows you to create, remove and adjust of lights, light color, intensity and ambient light levels.
  - Allows you to adjust the reflectivity of the models surface.
  - Allows you to toggle Specular lighting, Gourad Shading, Interpolated Colors and Toon Shading
  - Provides you with x,y,z guidelines and a wireframe mode in the Polygon view mode.
  - Allows you to toggle rendering the scene with/without perspective.

#### Limitations

- The Viewer distance used in the Perspective calculation is not adjustable (Allowing this made it too easy to loose the model)
- Wireframe and guidelines are only available in Polygon mode
- Rendering is quite wasteful and not managed terribly efficiently. All possible lighting, shading and coloring values are interpolated on every scene update, regardless of whether they are enabled.

## How to use:



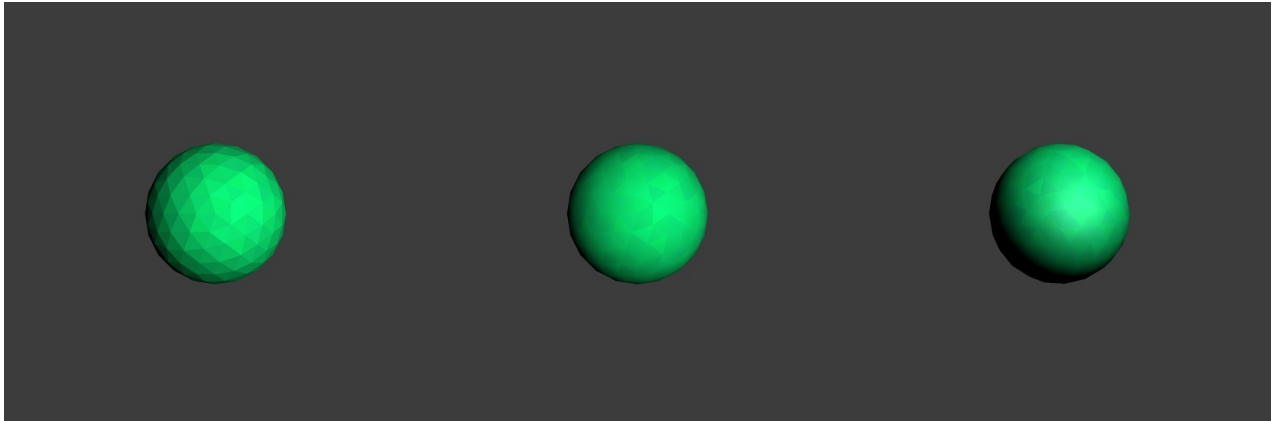
*Key UI Elements*

# Extensions and Results

## Diffuse and Specular Lighting

---

First I implemented Gourad shading and specular lighting, the gourad shading allowed the lighting to appear much smoother, the specular lighting makes the image appear more real.



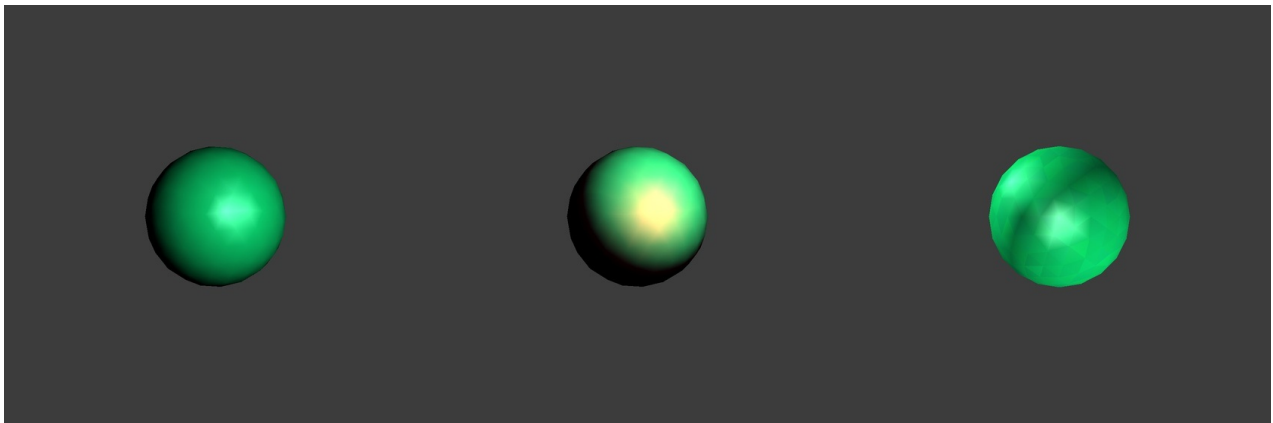
*Base Case*

*Base Case with Gourad  
Shading*

*Base Case with Gourad  
Shading and Specular Lighting*

To make the ball seem even smoother I added an option to also interpolate colors, which makes the different colored polygons of the ball less apparent.

Multiple lights, and coloring of the light source are also added options.



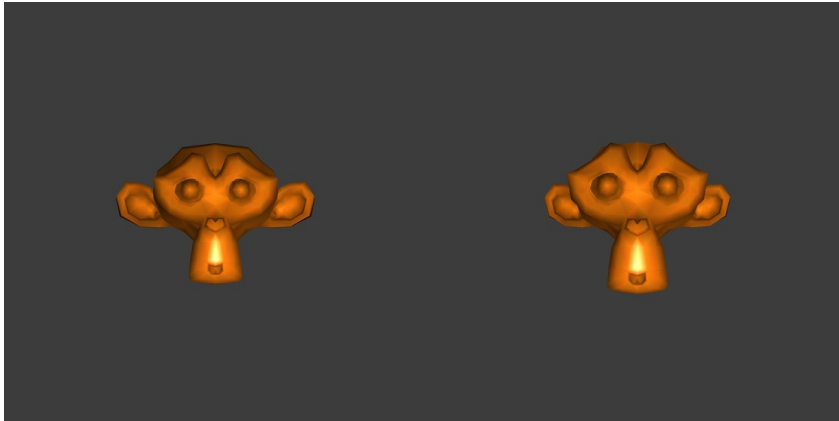
*+Interpolated Coloring*

*+Alternate Lighting Colors*

*+Multiple Lights*

## Perspective

I added an option to render with perspective. This gives a much better sense of depth and distance of an object and means that translating along the z-axis when both the y and x axis are perfectly aligned still provides meaningful differences to the resulting image.



*No Perspective*

*With Perspective*

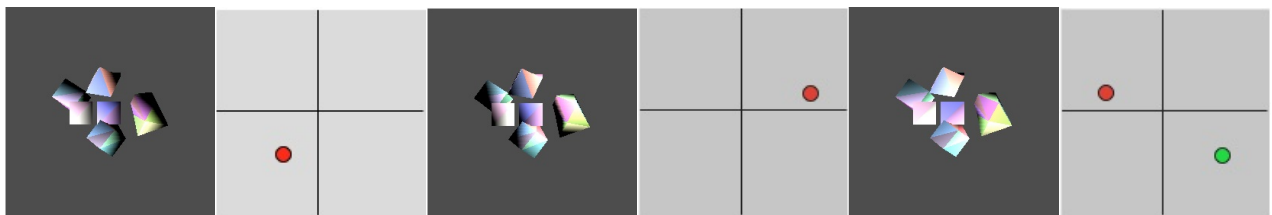


*No Perspective*

*With Perspective*

## Real Time Lighting + Multiple Lights:

The lights in the scene update in real-time as you drag them around, or adjust the z-slider.



## Toon Shading and Light Colors

Just for fun I also added “Toon shading”, which I implemented by:

- forcing the shading to be posterized by rounding all shading values to within 0.3
- Upon rendering calculating all the edges of the resultant image and rendering a black-outline around it.

Aswell as the intensity of the light, color can also be adjusted. This affects both the reflected specular light aswell as the tone of any light vectors. Ambient light is not affected.

The magnitude of how much specular light affects the model is determined by both the light-intensity and the shininess of the model. Both of these values can be adjusted in the GUI.



*Toon Shading*

*Toon Shading*

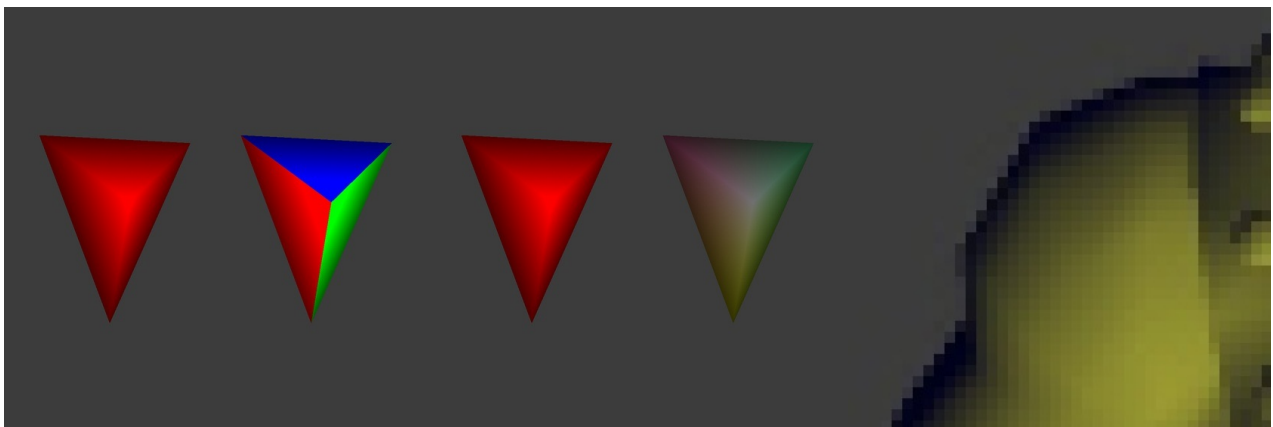
*Toon Shading & Colored Lighting*



*Toon Shading & Colored Lighting*

*Colored Lighting*

*High Intensity Lighting*



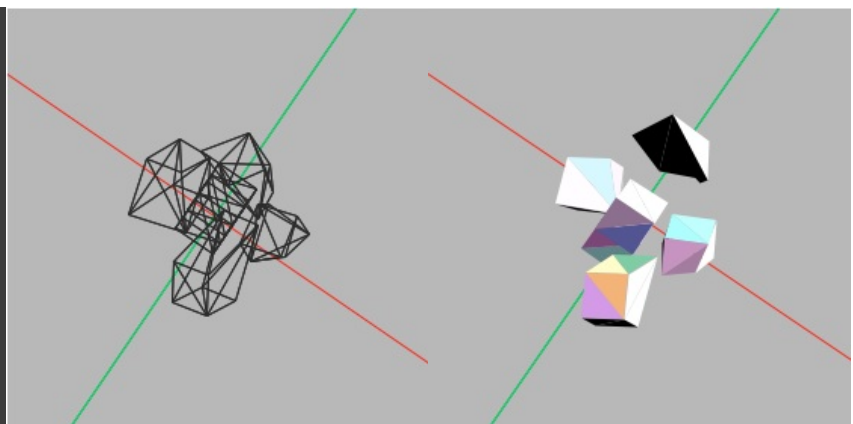
*Plain Coloring*

*Interpolated Coloring*

*Saved images are Anti Aliased*



*Saved images are of Higher Resolution*



*Polygon mode shows guidelines and wireframe*

*Sacrifices nice shading for simple shadows and speed*