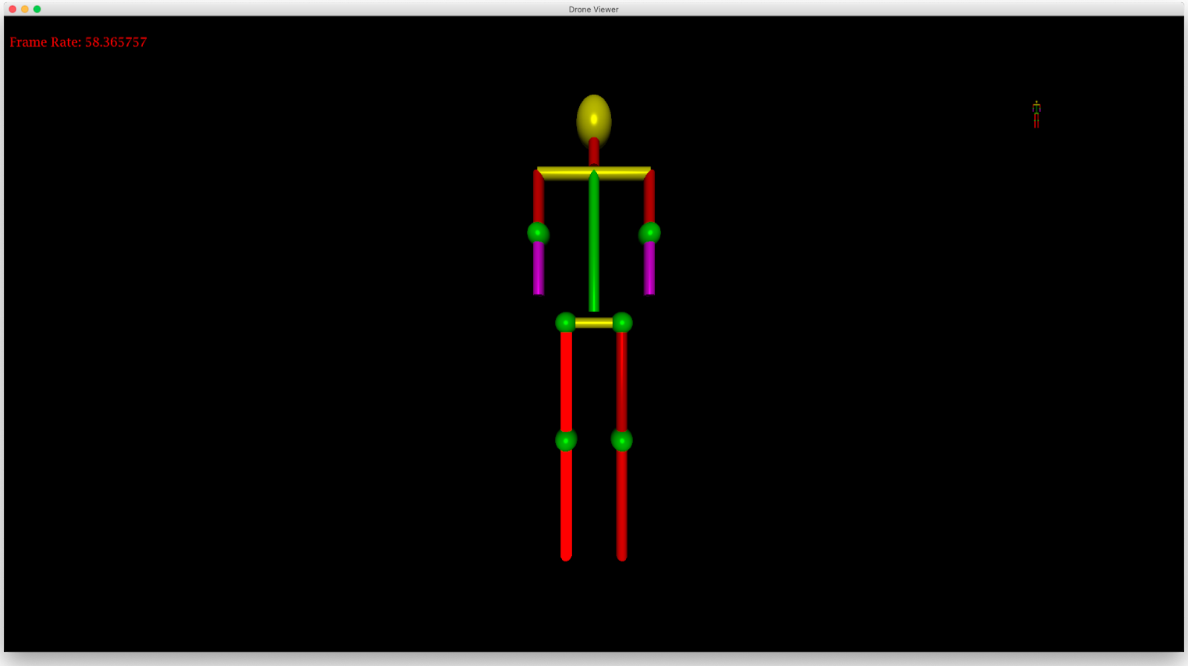
# Assignment 6 User Manual

**Authors:** Benjamin Clauss and Anthony Dominianni

## Overview

This project is a simple scene viewing application with OpenGL. Two primary scenes are provided for viewing - humanoids in the YMCA pose and 2 buildings. There are also other simpler scenes. Any scene can be used provided a respective scene graph file and input configuration (See Usage). There are 2 views displayed. The first is that of a stationary global camera with a position specified by the input configuration. The second is that of a movable drone that can zoom in and out. Whichever camera is toggled will have its view displayed with the other camera's view in the top right corner. The scene also supports lighting (including spotlights) and textures. Like the scene, these are specified in the scene graph file (See Lighting & Textures). Animations are also supported but limited (See Animation).



Frame Rate: 58.365757

# Usage

To use the program, run *DroneViewer*. This program accepts a single command-line argument. The argument is the path of an input configuration text file. These configurations specify the position and orientation of the global camera. The following configurations are provided:

- *configurations/ymca.txt*
- *configurations/buildings.txt*

# Controls

The drone camera is keyboard-controlled, and can be used to interactively fly around in the scene. This drone can also be seen from the global camera's viewpoint.

- "Space" : Toggle between global camera and drone camera
- "up" : Camera moves up
- "down" : Camera moves down
- "right" : Camera moves to the right
- "left" : Camera moves to the left
- "a" : Camera turns left in place
- "d" : Camera turns right in place
- "w" : Camera turns in place
- "s" : Camera turns down in place
- "f" : Camera tilts right in place
- "c" : Camera tilts left in place
- "+" : Camera zooms in
- "-" : Camera zooms out

# Design

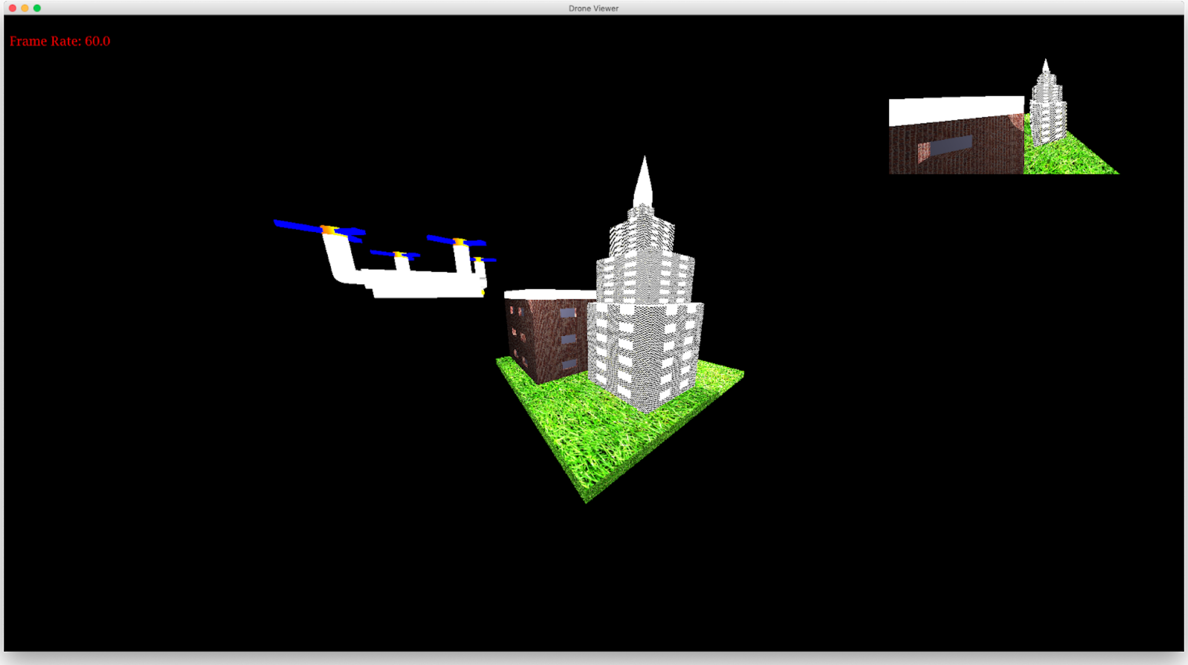
## Lighting and Textures

Lights and textures for a scene are specified in the scene graph file. When this scene graph is loaded, the program will parse the lights and textures and use them in rendering. The included "Assignment 6" document has more information as to how this was implemented.

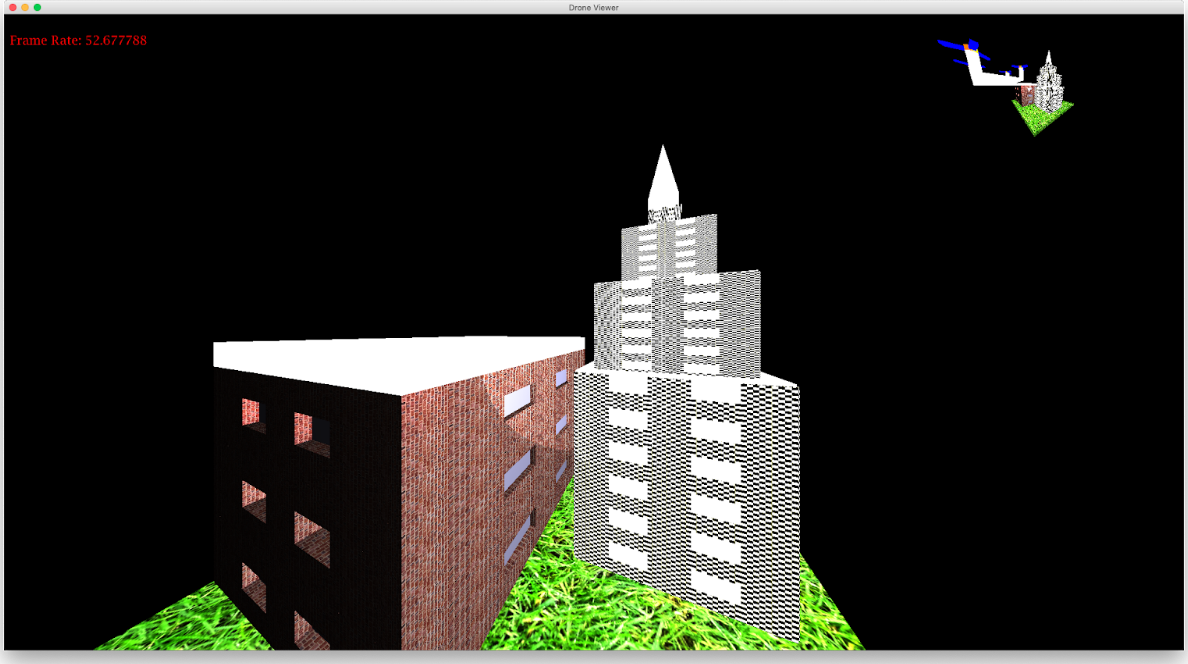
## Animation

Animation is "hardcoded" in to the View class. A map of node names to animations for that node are provided to the scene graph when drawing. To animate things in a scene, simply pass a map of node names and respective animations to the scenegraph.

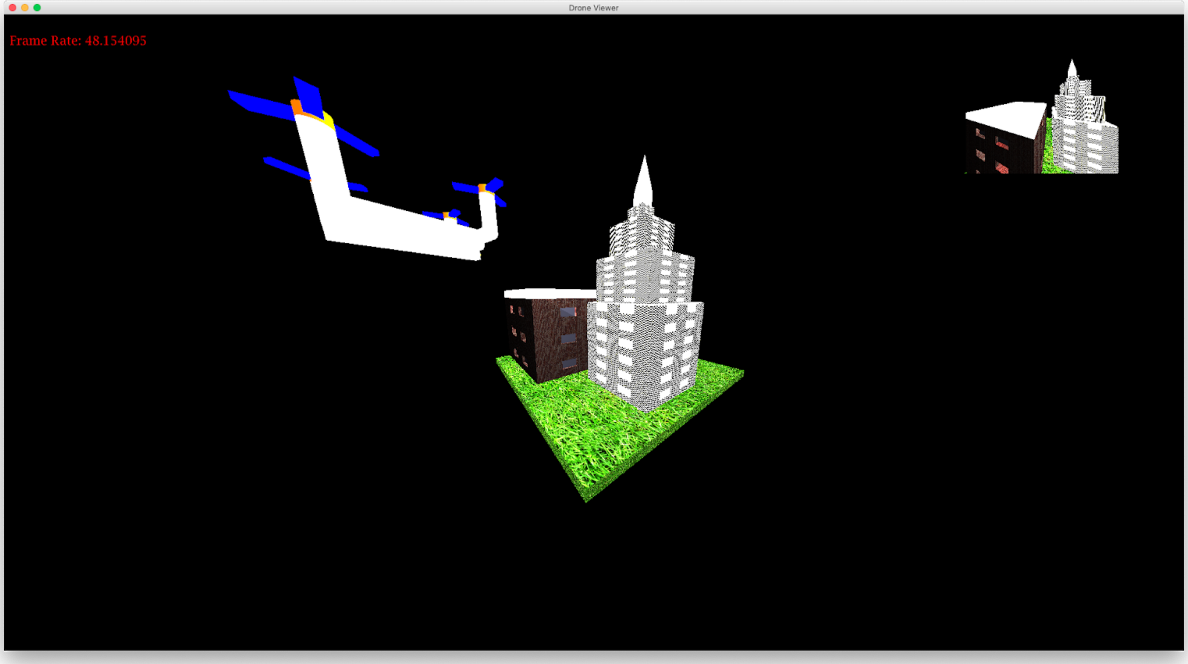
# Images



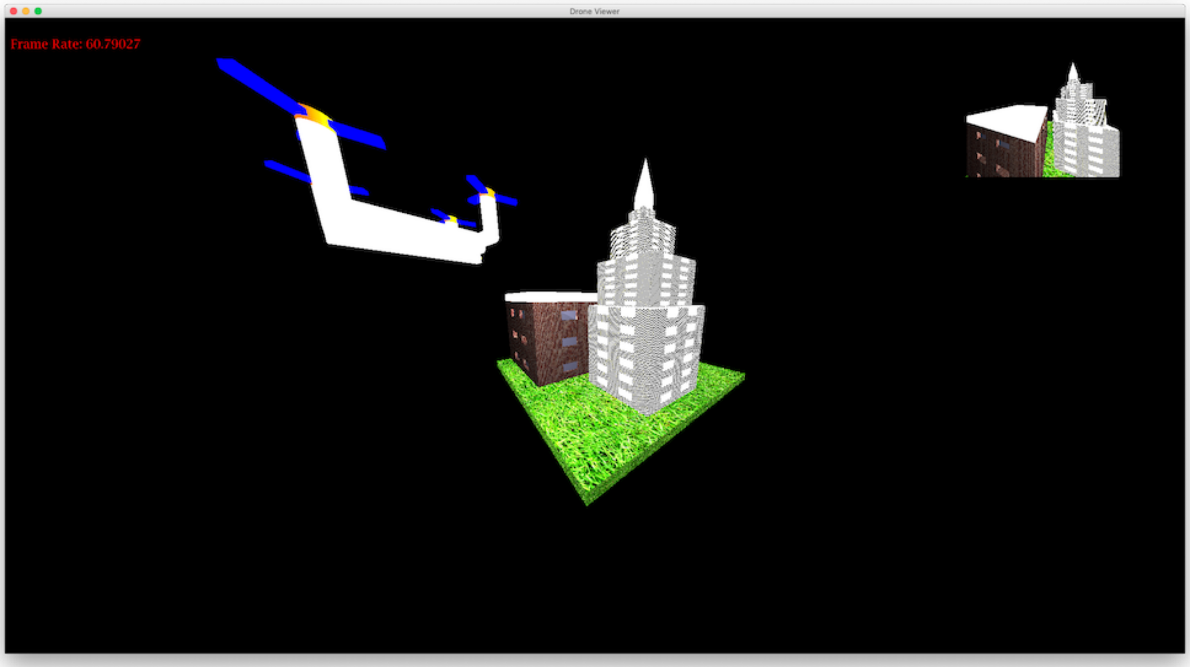
Frame Rate: 60.0



Frame Rate: 52.677788



Frame Rate: 48.154095



Frame Rate: 60.79027

1. The first image is the default view with drone with animated propellers.
2. The second image is the drone view. Note the spotlight effect from the drone.
3. The third image shows the dark effect of a rotating light around the scene.
4. The fourth image shows the light effect of a rotating light around the scene.

## Texture Sources

- http://www.textures4photoshop.com/tex/nature-grass-and-foliage/seamless-grass-texture-free.aspx
- https://www.3dxo.com/textures/5335_bricks_30

## References

- https://learnopengl.com/Lighting/Basic-Lighting
- https://www.tomdalling.com/blog/modern-opengl/08-even-more-lighting-directional-lights-spotlights-multiple-lights/