1 Introduction

Write each program using either Windows or Linux using Code::Bolocks or using the Mac and XCode, zip up all your work and upload the projects to the Assignment #7 page of the MyClasses site for this class.

In these exercises you need to update the DoxyGen documentation. Fully document all new data members and methods to any of the classes, and update the documentation for anything that has changed, including the documentation in the main that gives an overview of the program.

2 Exercise

In this exercise you are to create a plane with a green texture and four towers that have 5 tori and a sphere on top as in the example screen shots below. The tori are to be reduced in size as you move up the tower. The tori are in a blue plastic and the spheres are a duller gray, as can be seen by the specular portions of the surfaces. Each object will be illuminated by some extra red in an animation that starts at the bottom and works its way up the tower. After the sphere is illuminated the animation starts back at the bottom torus. The illumination fades in and out at a default rate of one cycle per second. The S key should increase this speed in a smooth manner to a maximum of one cycle in 0.1 seconds and the A key should decrease this speed in a smooth manner to a minimum of one cycle in 10 seconds. The animation must be controlled by an SFML clock, not the framerate,

As can be seen by the specular highlights there are three light sources, the spherical positions of these lights are (r, θ, ψ) of $(20, 45, 45)$, $(20, 100, -45)$, and $(20, -100, 60)$ in degree measurement. The lights should not have a bounding sphere on them like my examples do nor do any need to be movable.

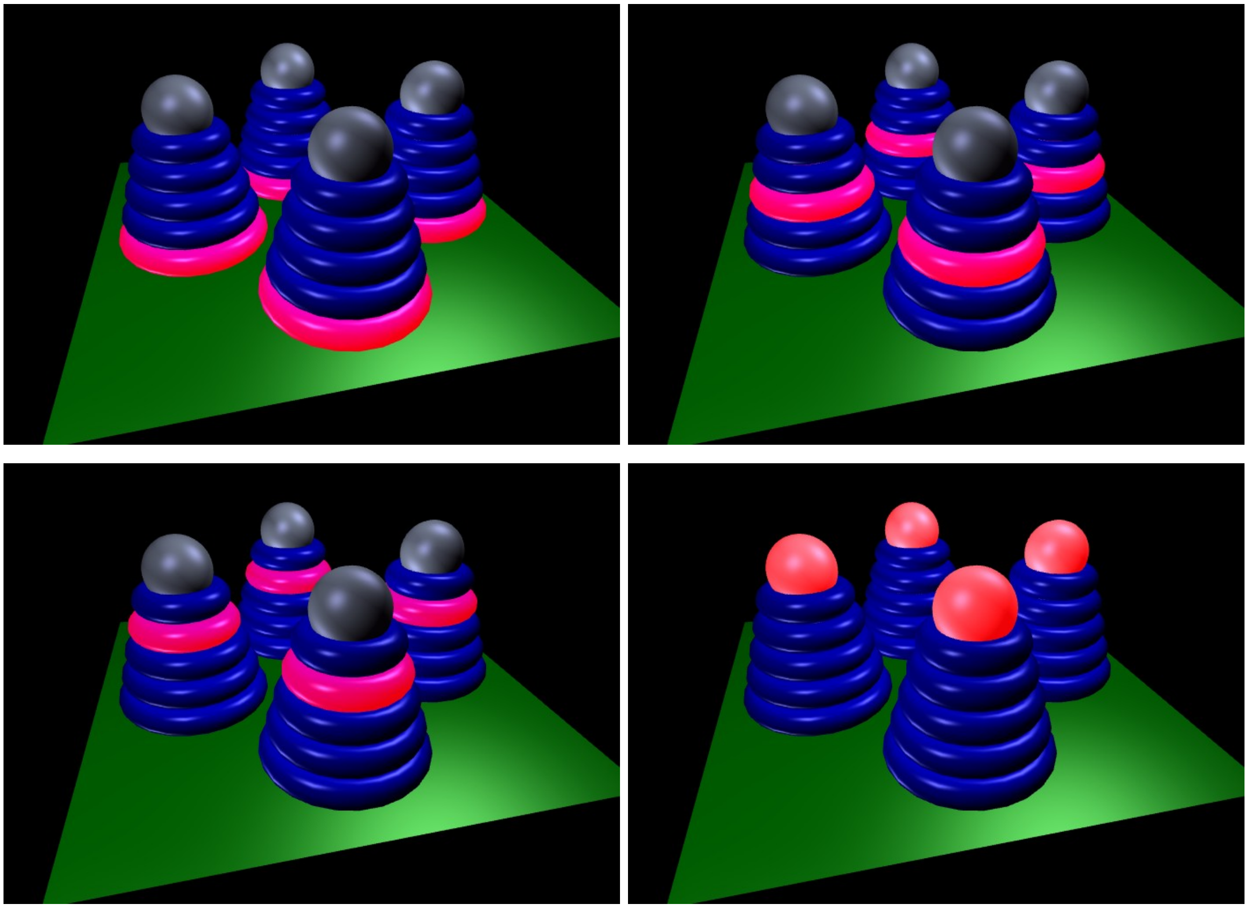
The plane is at the origin and the standard YPR and spherical camera options are expected. You can get by with a single torus and a single sphere object but it will make the programming a little easier to use five torus models and one sphere model. You may do either but you will have at most one sphere and 5 tori loaded to the graphics card and the rest is done with transformations.

User Interface:

- Escape: Ends the program.
- M: Toggles between fill mode and line mode to draw the triangles.
- W: Resets the animation speed to 1 per second.
- S: Increases the animation speed.
- A: Decreases the animation speed.
- F10: Saves a screen shot of the graphics window.
- F11: Turns on the spherical camera.
- F12: Turns on the yaw-pitch-roll camera.
- If the spherical camera is currently selected,
  - Left: Increases the camera's theta value.
  - Right: Decreases the camera's theta value.
  - Up: Increases the camera's psi value.
  - Down: Decreases the camera's psi value.
  - A click and drag with the left mouse button will alter the theta and psi angles of the spherical camera to give the impression of the mouse grabbing and moving the coordinate system.
  - A click and drag with the left mouse button while the Control key is pressed will alter the radius (using the change in the y coordinate) for zooming in and out.

  - Ctrl+Up: Decreases the camera's radius.
  - Ctrl+Down: Increases the camera's radius.

- If the YPR camera is currently selected,
  - Left: Increases the yaw.
  - Right: Decreases the yaw.
  - Up: Increases the pitch.
  - Down: Decreases the pitch.
  - Ctrl+Left: Increases the roll.
  - Ctrl+Right: Decreases the roll.
  - Ctrl+Up: Moves the camera forward.
  - Ctrl+Down: Moves the camera backward.
  - Shift+Left: Moves the camera left.
  - Shift+Right: Moves the camera right.
  - Shift+Up: Moves the camera up.
  - Shift+Down: Moves the camera down.



3 Extra Credit

For a little extra credit add in the user interface of the mouse wheel so that scrolling the mouse wheel will zoom in and out with the spherical camera. It should not have an effect on the YPR camera.