Notes:

1. For Interactive camera part, use Q, W to control X-direction movement(left and right), E, R to control Y-direction movement(up and down), T, Y to control Z-direction movement(zoom in and out), A, S to control X-direction rotation, D, F to control Y-direction rotation, G, G to control Z-direction rotation.
2. Haven't completed Lambertian Reflection(5 pnts). For extra credit part, I've already added own custom code to the JSON files provided as well as MainWindow::on_actionLoad_Scene_triggered() and the Polygon class, and I've obtained the normal vector for each point using Barycentric_Interpolation and Perspective correction so far, what's left is to use the normal vector to texture the image, which I don't really know how to interpolate the vector for color/texture mapping.

Resources:
09/25 14:00
Search on how to extra RGB value of a pixel in an image:
http://stackoverflow.com/questions/12382301/qt-qimage-how-to-extract-rgb

09/25 14:30
Search on how to handle key press event:
http://stackoverflow.com/questions/12382301/qt-qimage-how-to-extract-rgb

09/25 15:00
Look into some intros to rasterizer:
http://www.scratchapixel.com/lessons/3d-basic-rendering/rasterization-practical-implementation/rasterization-practical-implementation

09/25 18:00
Something about Lambertian Illumination Model:
https://github.com/mattdesl/lwjgl-basics/wiki/ShaderLesson6#IlluminationModel
https://developer.valvesoftware.com/wiki/Constant-Linear-Quadratic_Falloff

09/28 19:00
Office hour. Question about Z-direction interpolation since my program didn't work at first in regard to z direction translation and got image scaling problems. Finally solved because I didn't divide by W.