For the final project, I attempted a low level project (the travel component) by having a car drive down a road to a village made of houses. The camera would start within the car and be able to switch between the car and an aerial view. The camera and car can be moved forward as well.

The skybox is created first. The skybox creation is identical to the skybox creation in the Skybox sample code on Moodle minus the cube object. The main replacement is that the bottom part of the skybox is textured grass instead of sky. The grass texture was taken from an online site (credited in the skybox folder) and turned into a ppm by myself.

The car and village is created by importing models using GLM. The credit for these models can be found in their respective folders (car and house). First, the camera is positioned in the car’s driver’s seat by adjusting the eye and the yaw. These imported models are then run through the code provided in class to put them into the scene, translating the houses slightly so they are farther off in the distance. After that, textures are enabled on both the house and the car and applied to them using OpenGL’s support of texture mapping. However, even though the texture mapping code is present, the texture does not seem to be applying itself as the time of this writing.

To display the car and village, I used a modification of the generic display in the sample code. When creating the models, the location of the car and village has been modified so they are significantly far away, giving the car a chance to “drive up” to the village. The Translate function has a variable (drive) that increases as the user holds the up arrow on the keyboard along with movement of the camera using eye. The hold is discovered using glutSpecialFunc() and glutSpecialUpFunc() to measure the key being pressed and when it’s lifted. With these aspects, it seems like the user is driving the car. The car can also go backwards.

The skybox display is created first theoretically, but there seems to be an issue with the depth buffer. No matter what combination of settings I am using (depth buffer on or off, order, etc.), the skybox overwrites every other model. So either the skybox generates fine, or the models do. This can be toggled by uncommenting “displaySkybox()” in the display function.

The keyboard controls the camera location. By hitting “v,” the eye adjusts the camera to go above the car in a helicam view. By hitting “V,” the camera goes back in the car. There is a simple if check to make sure “v” or “V” being hit twice won’t make the camera change in the wrong location. The user can only hit “V” and see an effect if he or she has hit “v” before, and vice versa. The up arrow key also happens to control the car driving. The idle function will keep the car and camera moving forward as long as the up key is being pressed. You can also drive backwards. I have also left the debugging controls on, which allows you to take a look at the environment. If you are not debugging, please only use the following controls.

* “v”
* “V”
* Up arrow
* Down arrow

The quality of the camera controls has only been guaranteed with these.

An attempt to code a model for the ground was attempted by creating a significantly large polygon as the ground below the car. The code is commented out and not used. The skybox bottom face is grass to make up for this.