

Assignment 3

CMSC 427, Fall 2001

Due: 11:00am Tuesday, November 6, 2001

(a) Read in the 3D models `plane.dat` and `land.dat`. They are available at the web-site www.cs.umd.edu/class/fall2001/cmsc427/assg3/ in the following format: (5)

- Line 1 contains the number of vertices N in the model.
- The lines 2 through $N + 1$ describe the N vertices, one vertex per line:

vertex-ID _{i} x_i y_i z_i r_i g_i b_i

- Line $N + 2$ contains the number of triangles M in the model.
- The lines $N + 3$ through $N + M + 2$ describe the M triangles, one triangle per line:

face-ID _{j} vertex-ID _{$j1$} vertex-ID _{$j2$} vertex-ID _{$j3$}

(b) Set up the viewing system with the following parameters:

- The viewer is at $(0, 1, 1)$, looking at $(0, 1, 0)$. The viewer's up vector is defined by $(0, 1, 0)$.
- The field of view is 90° , the window's aspect ratio is 1.0. Set the near plane to be 1 and the far plane to be 800.

Next, rotate the plane 270° along X-axis, followed by 90° along Y-axis. Then scale the plane by a factor of 15 along X , Y , and Z axes. Then translate the plane to $(0, 20, -135)$.

For the terrain, first translate it to $(-0.5, -0.5, -0.2)$, then scale it by $(600, 600, 100)$ along the X , Y , and Z axes. Then rotate it 270° along X-axis, followed by 90° along Y-axis.

Since you are now going to need depth per pixel in your frame buffer be sure to have an additional flag of `GLUT_DEPTH` in `glutInitDisplayMode()`. You should now be able to see the static plane above a terrain in your view window. (5)

(c) In this part you have to implement the plane doing flips as it flies. First implement the plane flying in a circle in 3D that is parallel to the XZ plane and has its center at $(0, 20, -85)$ and radius 50. (4)

Next implement the plane flipping (rotation about the direction in which it is flying) when the user hits the 'f' key. The plane should continue slowly flipping for one complete cycle. (3)

(d) In this part you have to implement the walking around on the terrain as well as pointing the gun in different directions in 3D.

Translation: Let us say that the user presses the left mouse button and drags it by $(\Delta x, \Delta y)$ before releasing it. The speed of translation should be proportional to $(\Delta x, \Delta y)$ and it should continue till the user presses the left mouse button again. Note that the viewer is translating in the XZ plane. (4)

Rotation: Use the four arrow keys to rotate the gun along θ and ϕ . In this assignment the user's view is identical to the gun (it is as if the user is looking through the gun's target finder). (4)

(e) Implement the firing of missiles in 3D when the user presses the key 's'. Assume each missile is a cylinder (you may use `gluSphere()`) and just have it fly in the direction the gun was pointing when it was fired. OpenGL will take care of things such as the cylinder should appear smaller as it goes farther from the viewer. (5)