Lab 5 - Virtual Trackball

You may work in pairs on this assignment. To receive credit, demonstrate your completed program to me by March 20th.

In this lab, you'll learn about an intuitive camera control called a virtual trackball. A virtual trackball takes 2D mouse or touch movements and converts them into a 3D rotation that mimics a trackball. The same grid and cube from Lab 4 have been provided. The UI elements from Lab 4 have been removed. It is your job to implement a virtual trackball to rotate our view of the scene (or more appropriately rotate our scene in front of our camera).

Part 1 - Virtual trackball surface equations

The first part is to implement the equations necessary to convert a 2D mouse click to a 3D point on the virtual trackball surface. Fill in the pointOnVirtualTrackball method so we can use it in the next parts. See the Object Mouse Trackball link for the equations to use.

Part 2 - Construct a rotation matrix

When the user clicks and drags on the screen, we need to use that information to construct a rotation matrix that will change our view. Start by converting each click or move event using the pointOnVirtualTrackball method from Part 1. Keep track of the previous event's location in addition to the current one. Take those two vectors and solve for the angle between them using the dot product. Then solve for a unit vector perpendicular to the two vectors to find the axis of rotation. Use the angle and axis values to construct a rotation matrix using glm::rotate. Keep track of all your rotations by continuously multiplying your rotations together. Create a trackball matrix variable that is initialized to the identity matrix. After each rotation, multiply it into your trackball matrix.

Part 3 - Transform our view

Rather than think of our view changing, think of the scene rotating in front of us. Adjust your vertex shader to accept another matrix for your trackball transformation and multiply it in your chain of transformations in the correct order.

Things to notice

When you click and drag the scene should rotate in the same direction. If it seems to be rotating opposite, you may be crossing your vectors in reverse order.

Recommended Reading

Object Mouse Trackball