

Lab 6 – Starting Vector class in preparation for the virtual trackball.

This lab will prepare you to start writing your program to rotate an object using the virtual trackball. In addition, you will be developing your 3D vector class, which you will use for the rest of the quarter. Take time to develop and test these functions, as you will be using them in almost every program we write from now on.

- 1) Write a class or struct for 3D vectors (using 3 floats or doubles to represent the direction and magnitude of the vector. I also recommend including another 3 floats to represent the current position of the tail of the vector).
- 2) Write code to create at least 2 vectors, based on the user's mouse location. You may write the vector interface however it makes sense to you – however your lab must allow the user to draw two vectors and the second vector should be drawn as the user drags the mouse around. Note that for this lab both vectors will start at the origin, so when the user creates the first vector for example, it should be drawn starting at the origin and pointing to wherever the user currently has the mouse. I suggest you only create and draw the vector when the left mouse button is down.
- 3) Create code to draw your vectors to do some visual debugging:
 - a. For now we will still be drawing in 2D, so just ignore the z value of your vector for drawing and set them all to 0.
 - b. You can just use a `GL_LINE` to draw the vectors, but add some designating factor (like a point at the tail of the vector), which will distinguish your vectors from lines.
 - c. Compute the length of each vector, which is drawn, and print it to the console, (and be sure to test a reasonable case in order to test the validity of your code.)
- 4) Write a function which computes the dot product of two 3D vectors
 - a. Compute the dot product between the two vectors the user has drawn (and be sure to test a reasonable case, for example, perpendicular, in order to test the validity of your code.)
- 5) Write code to compute the cross product of two 3D vectors. Make sure your cross product code is correct. For example, if you draw two vectors the closely follow the x axis, the vector generated by the cross product should point along the y axis or the negative y axis.