Project 1 - Transformation with Trackball

CS 1566 — Introduction to Computer Graphics

Check the Due Date on the CourseWeb

The purpose of this project is for you to transform an object in three dimensional space using a mouse.

An Object in 3D Space

First, you need to create an object in 3D space. It can be any object that you like but preferably with multiple (more than 10) faces. Since we are not applying the lighting effect yet, each face should have a unique color. Otherwise, you will not be able to distinguish between faces if they are next to each other. Or at least, make sure that two faces that are next to each other have different color.

Zoom In and Zoom Out

For this project, we will use the scroll wheel of a mouse to scale an object. Scroll one way is an equivalent of enlarge an object in all direction about the origin by the factor of 1.02. Similarly, to shrink the object by the factor of 1/1.02 can be done by scrolling the other way.

Scrolling events are the same as mouse event. So, you need to call the `glutMouseFunc()` function to register your callback function:

```
glutMouseFunc(mouse);
```

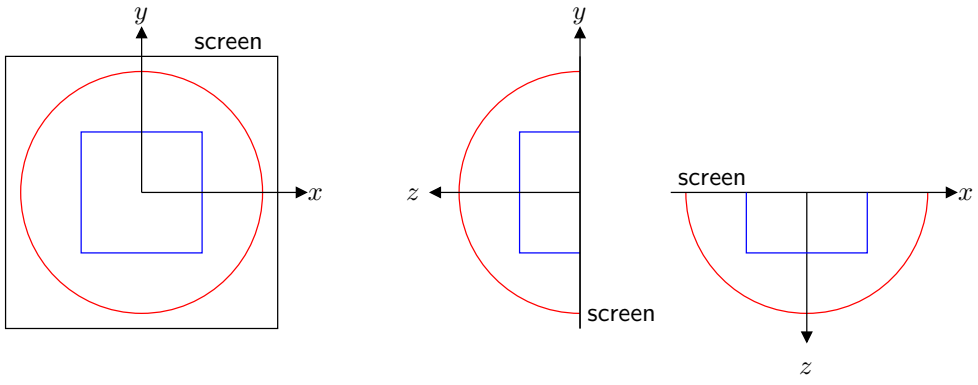
where your `mouse()` function should look like the following:

```
void mouse(int button, int state, int x, int y)
{
   :
}
```

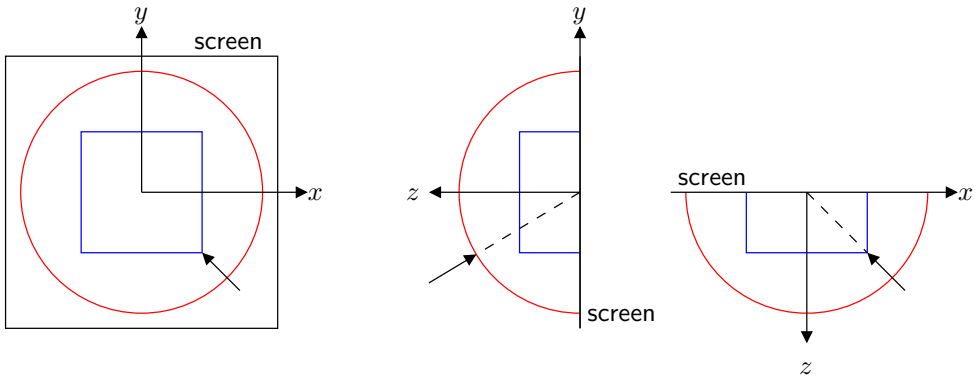
If you scroll up, the `mouse()` function will be call with the variable `button` initialized to 3. Similarly, if you scroll down, the variable `button` will be initialized to 4. Simply apply the scaling matrix and call the `glutPostRedisplay()` function.

Rotation using Trackball

To rotate an object in 3D, simply imagine that your object is located in the middle of a glass ball. This glass ball can be spun in any direction. Now, imagine that half of this glass ball pops out of your screen as shown below:



Note that we a user click a mouse on the screen, you have to imagine that the mouse pointer is a finger that touch the glass and point directly to the origin as shown below:



To rotate the object inside this class ball, user needs to simply move his/her finger while touching the ball. For this project, assume that a user's finger is always point directly to the origin while it is moving. Ideally, a user can twist his/her finger to rotate the glass ball. But since we cannot twist the mouse pointer, we assume that twisting the finger is not allow for this project. We will use left button of a mouse to simulate a user touches the glass ball. If the left button is down, user touches the ball at the current pointer position. If the left button is up, user released his/her finger from the ball. To capture the left button event, we use the same callback as in previous section. The variable `button` will be initialized to `GLUT_LEFT_BUTTON` and the variable `state` will be initialized to either `GLUT_UP` or `GLUT_DOWN`. The variables `x` and `y` will be set to the pointer position. **Note** that the pointer position is the screen position. The top-left corner of the screen is at $(0, 0)$.

If the mouse pointer is moving while the left button is down, it simulates a user turning the glass ball. When the glass ball rotates, it rotates about a vector and the fixed point of rotation is at the origin. Your job is to come up with the vector so that you can apply rotation matrices correctly. A method of calculating this vector will be discussed in class.

To capture mouse motion events, use the `glutMotionFunc()` function as shown below:

```
glutMotionFunc(motion);
```

and the `motion()` function should look like the following:

```
void motion(int x, int y)
{
   :
}
```

The variables `x` and `y` will be set to the current pointer position.

Spinning an Object

One special feature of this glass ball is that it can rotate indefinitely (no friction). If a user touches the glass ball, drags his/her finger, and releases the finger, the glass ball should spin in the same direction of the user's finger indefinitely. **Note** that the zoom-in/zoom-out feature must work while the object is spinning indefinitely.

Submission

The due date of this project is stated on the CourseWeb. Late submissions will not be accepted. Zip all files related to this project into the file named `project1.zip` and submit it via CourseWeb. After the due date, you must demonstrate your project to either TA or me within a week after the due date.