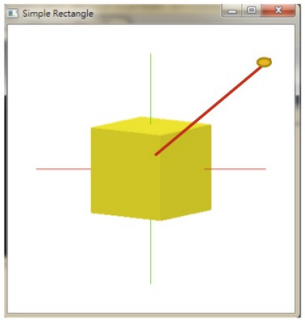
Lab 04
Transformation Matrix

Lab 04 Goal – Transformation matrix

1. Arbitrary Rotation 90%
   - Print the window coordinate (x,y) where your mouse click on (20%)
   - Draw the dot (only the last one) where your mouse click on (40%)
   - Draw the line (arbitrary axis) between the origin and the last dot (10%)
   - Rotate your object along the arbitrary axis (20%)
     - use your own key setting
2. You will still need to be able to do the rotation, translation from previous lab
3. Reset the object to origin 10%
   - use your own key setting



- Write comments in your code about your key setting
- Do not use glRotate, glTranslate in your code
- Turn in your code

Transformation Matrix

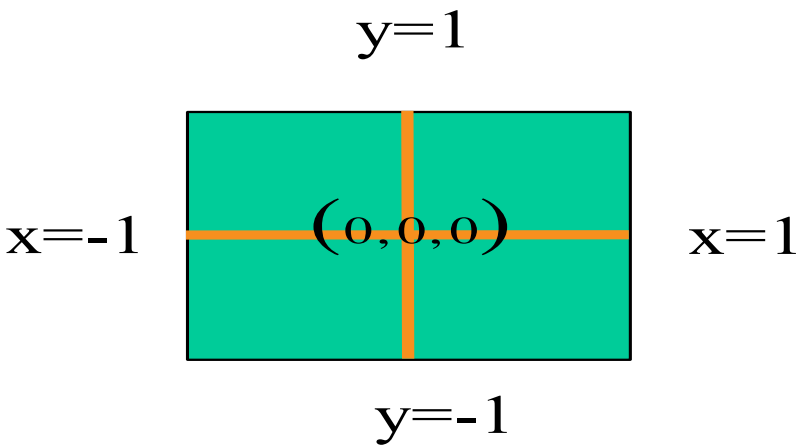
- All modeling transformations are represented as 4x4 matrices
- Identity matrix

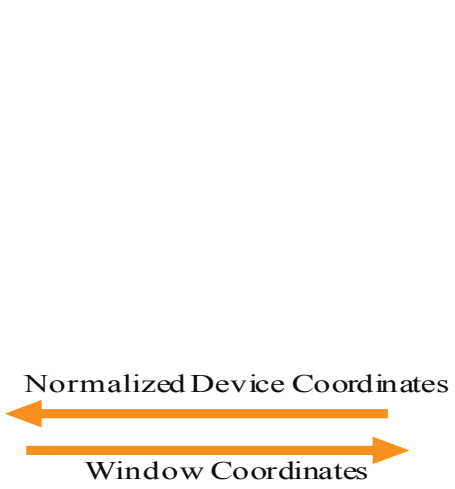
```
GLfloat rotMatrix[] = {
        1.0, 0.0, 0.0, 0.0,
        0.0, 1.0, 0.0, 0.0,
        0.0, 0.0, 1.0, 0.0,
        0.0, 0.0, 0.0, 1.0   };
```

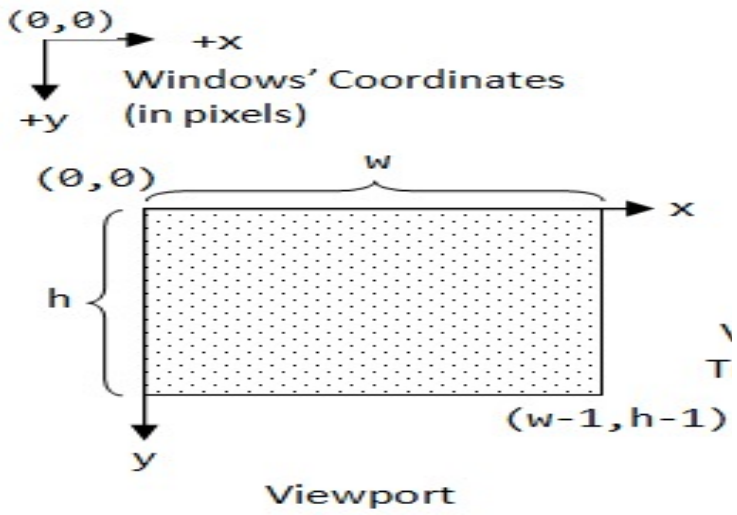
```
rotMatrix[0] = 1;   rotMatrix[4] = 0;    rotMatrix[8] = 0;   rotMatrix[12] = 0;
rotMatrix[1] = 0;   rotMatrix[5] = 1;    rotMatrix[9] = 0;   rotMatrix[13] = 0;
rotMatrix[2] = 0;   rotMatrix[6] = 0;  rotMatrix[10] = 1;   rotMatrix[14] = 0;
rotMatrix[3] = 0;   rotMatrix[7] = 0;  rotMatrix[11] = 0;   rotMatrix[15] = 1;
```

Mouse Click Location

- Click at (win_x, Win_y)
- Convert it to OpenGL's coordinate (x, y)
- Draw the dot







Positioning

- The position in the screen window is usually measured in pixels with the origin at the top-left corner
  - Consequence of refresh done from top to bottom
- OpenGL uses a world coordinate system with origin at the bottom left
  - Must invert y coordinate returned by callback by height of window
  - y = h – y;

The Mouse Callback

```
glutMouseFunc(mymouse)
void mymouse(GLint button, GLint state, GLint x, GLint y)
```

- Returns
  - which button
    - `GLUT_LEFT_BUTTON`
    - `GLUT_MIDDLE_BUTTON`
    - `GLUT_RIGHT_BUTTON`
  - state of that button
    - `GLUT_UP`
    - `GLUT_DOWN`
  - Position in window

My_Mouse() - Code snippe

```
/// Register callback functions
////////////////////////////////////////////////////////////
glutReshapeFunc ( My_Reshape );
glutDisplayFunc ( My_Display );
glutKeyboardFunc( My_Keyboard );
glutMouseFunc   ( My_Mouse );
glutSpecialFunc ( My_SpecialKeys );
glutTimerFunc   (timer_speed, My_Timer, timer_flag);
////////////////////////////////////////////////////////////

/// Initialize OpenGL
```

- Handle the mouse events

```
////////////////////////////////////////////////////////////
// Called by GLUT library when the mouse event is triggered
void My_Mouse(int button, int state, int x, int y)
{
    switch (button)
    {
    case GLUT_LEFT_BUTTON:
        if (state == GLUT_DOWN)
            cout << "Mouse left button down" << endl;
        break;
    case GLUT_MIDDLE_BUTTON:
        if (state == GLUT_DOWN)
            cout << "Mouse middle button down" << endl;
        break;
    case GLUT_RIGHT_BUTTON:
        if (state == GLUT_DOWN)
```

Which button ?

Up or down ?

Screen coordinates