3D Graphics Programming

T163 - Game Programming

Fall 2023

Week 3

Advanced Animation

Fall 2023

Animation Sequence

```
float angle = glutGet(GLUT_ELAPSED_TIME) / 1000.0 * 45; // 45° per second
transformObject(x, Z_AXIS, angle, glm::vec3(0,0,0));
glDrawArrays(GL_LINE_LOOP, 0, 4);
```

Input

❖ We use GLUT to get keyboard events

1. We bind the glut keyboard functions to local functions
2. We use our local functions to check which button event was invoked

Input

- Bind the functions (1)
- In the main function:

```
glutKeyboardFunc(KeyDown);
```

```
glutKeyboardUpFunc(KeyUp);
```

Input

- Define the local functions (2)

```
void KeyDown(unsigned char key, int x, int y)
{
   switch(key) {
        case 'w':
            // call a function
            break;
        case 's':
            // call a function
            break;
        default:
            break;
    }
}
```

FPS

FPS

Animation

- Problem is each hardware is different
  - We can't speed up the slow
  - But, we can slow down the fast

- Always aim to target 30 FPS
  - 33.33... milliseconds/frame

- If you want to target 60 FPS
  - 16.66... milliseconds/frame

Animation

- So we need to make sure the draw function is called at least every 33.33… milliseconds

- The glut timer function is similar to the invoke function in Unity

```
glutTimerFunc(time_ms, callback, timerID);
```

Animation

- We won't need to call the idle function anymore

```
void idle(){
    glutPostRedisplay();
}
```

```
void idle(){

}
```

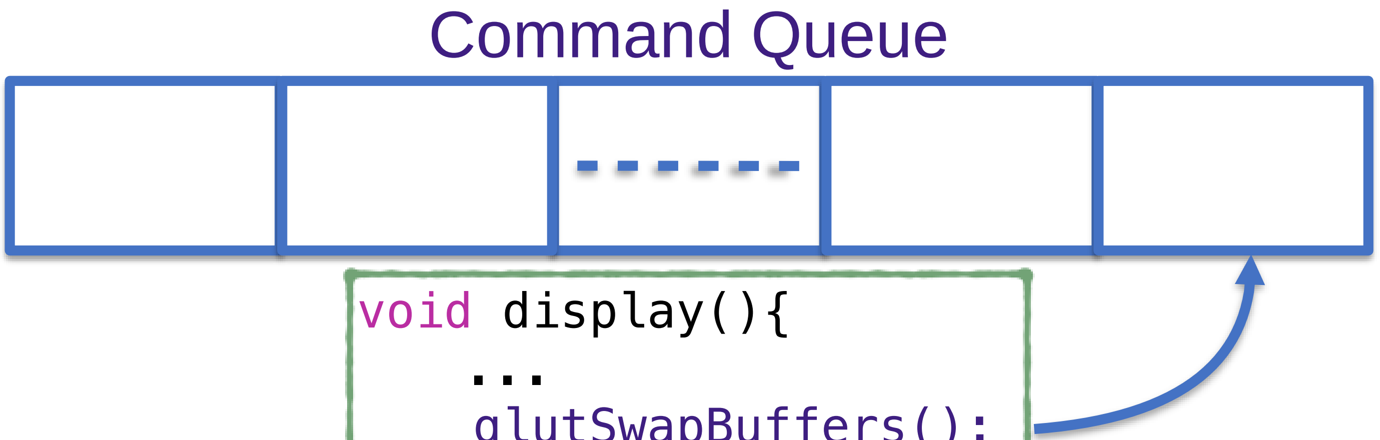
Animation

❖ In the main function:

```
glutTimerFunc(33, Timer, 0);
```

```
void Timer(int id){
    glutPostRedisplay();
    glutTimerFunc(33, Timer, 0);
}
```

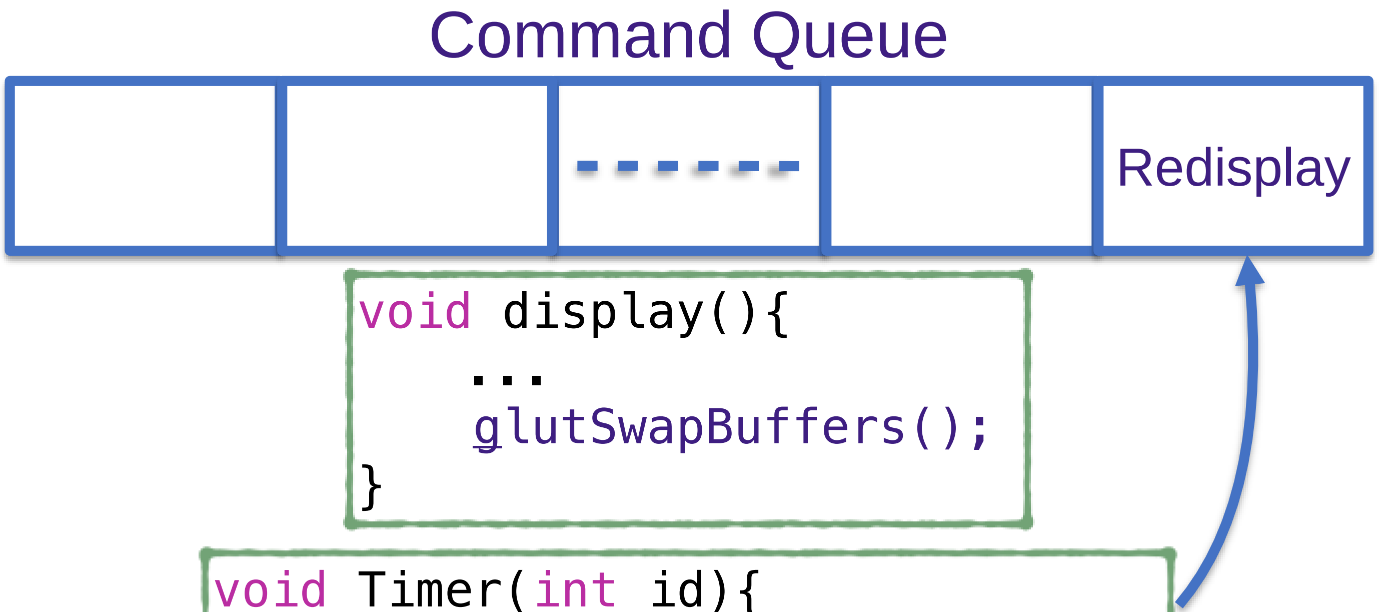
Animation



```
void display(){
    ...
    glutSwapBuffers();
}
```

```
void Timer(int id){
    glutPostRedisplay();
    glutTimerFunc(15, Timer, 0);
}
```

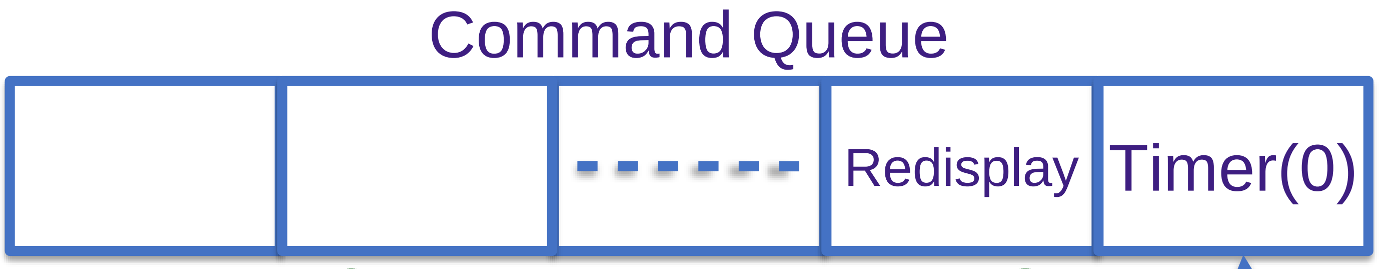
Animation



```
void display(){
    ...
    glutSwapBuffers();
}
```

```
void Timer(int id){
    glutPostRedisplay();
    glutTimerFunc(15, Timer, 0);
}
```

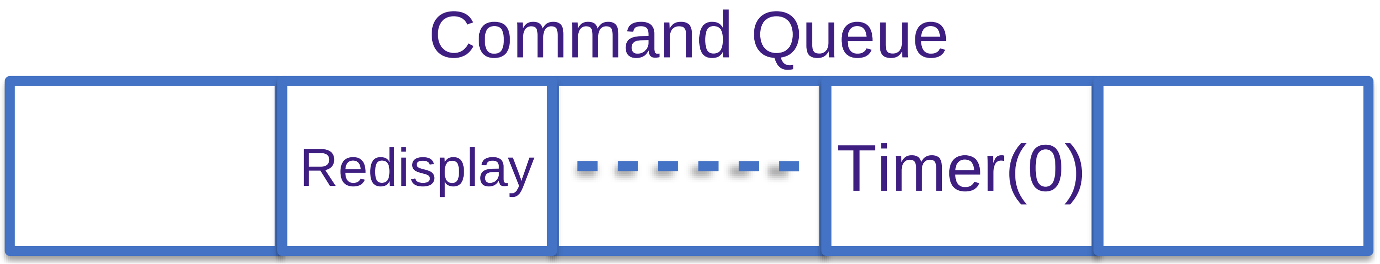
Animation



```
void display(){
    ...
    glutSwapBuffers();
}
```

```
void Timer(int id){
    glutPostRedisplay();
    glutTimerFunc(15, Timer, 0);
}
```

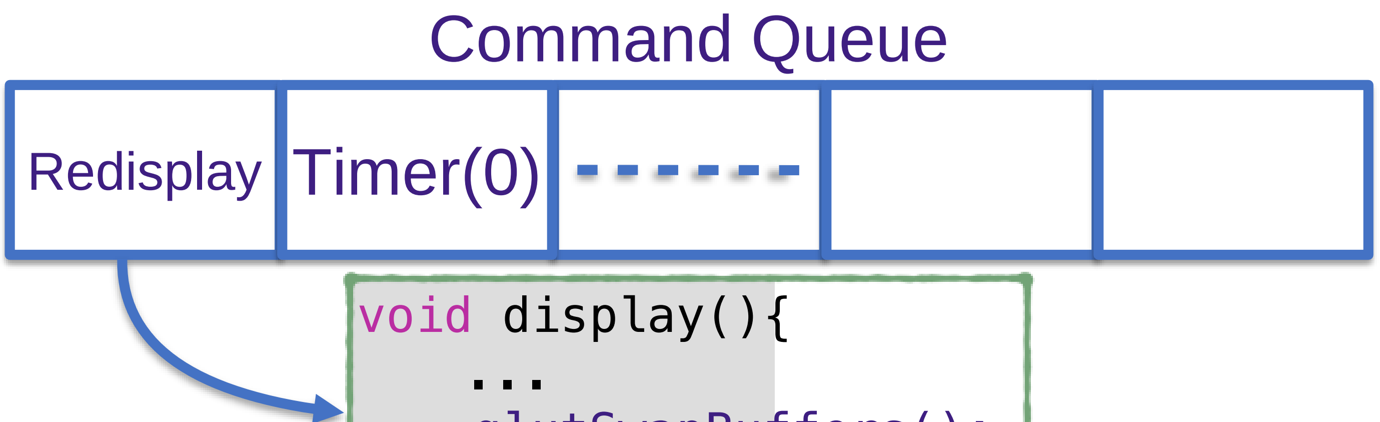
Animation



```
void display(){
    ...
    glutSwapBuffers();
}
```

```
void Timer(int id){
    glutPostRedisplay();
    glutTimerFunc(15, Timer, 0);
}
```

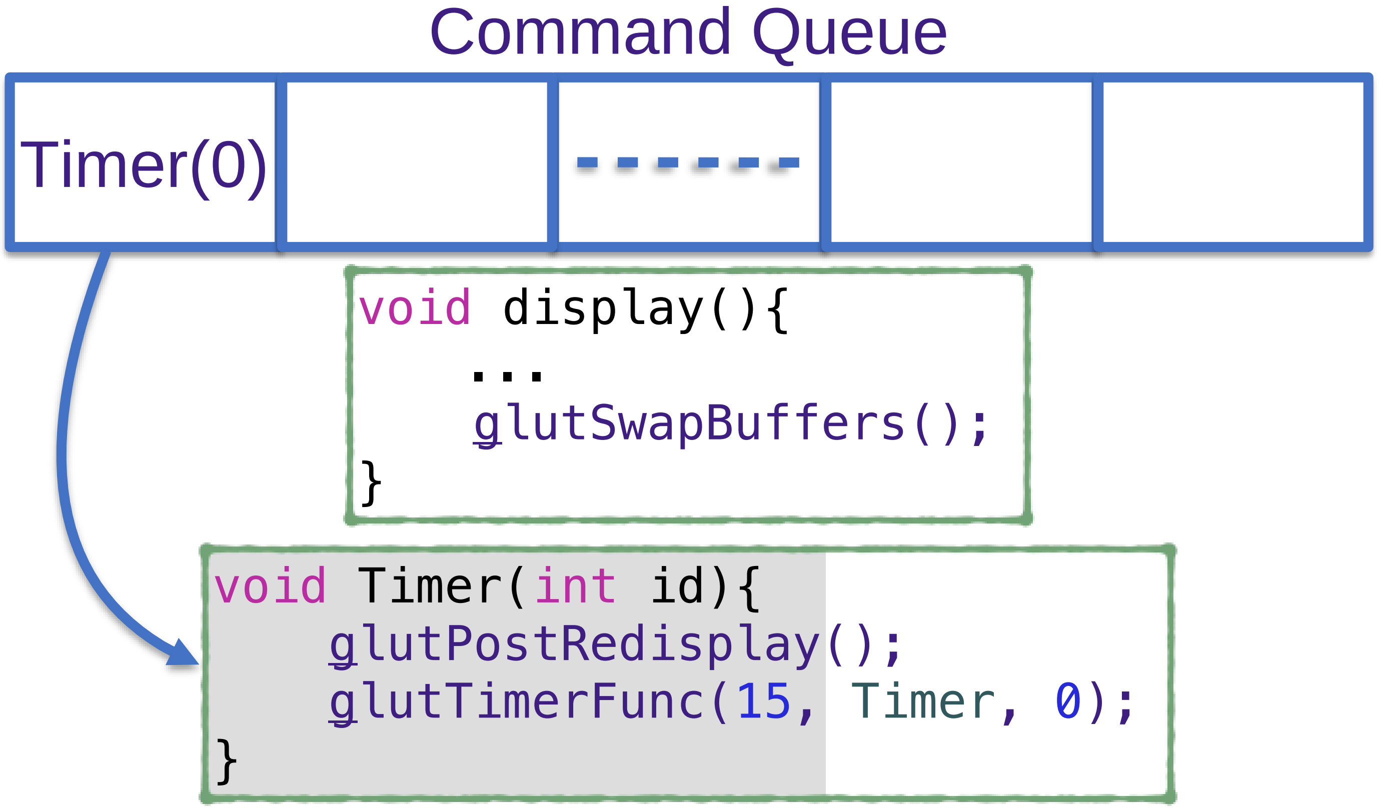
Animation



```
void display(){
    ...
    glutSwapBuffers();
}
```

```
void Timer(int id){
    glutPostRedisplay();
    glutTimerFunc(15, Timer, 0);
}
```

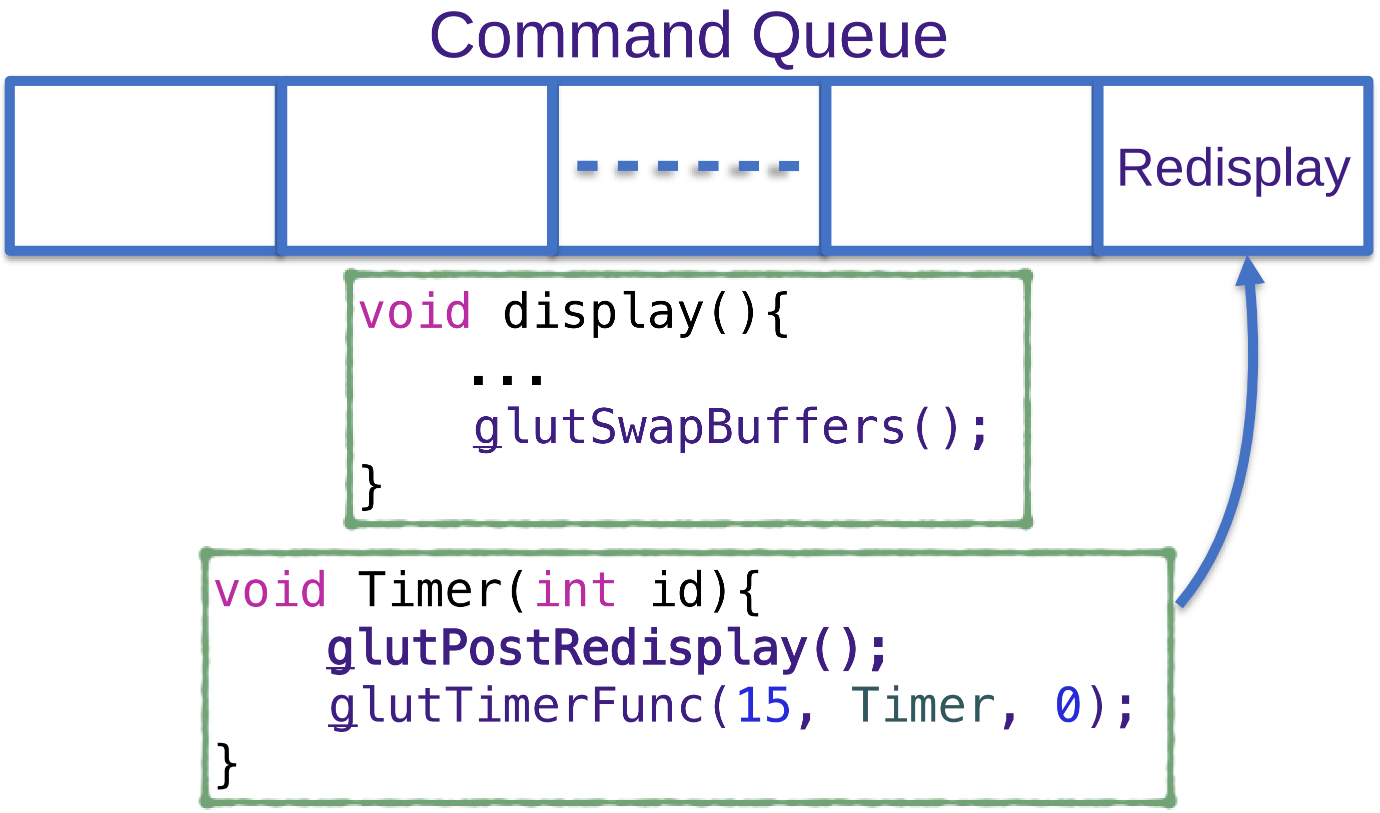
Animation



```
void display(){
    ...
    glutSwapBuffers();
}
```

```
void Timer(int id){
    glutPostRedisplay();
    glutTimerFunc(15, Timer, 0);
}
```

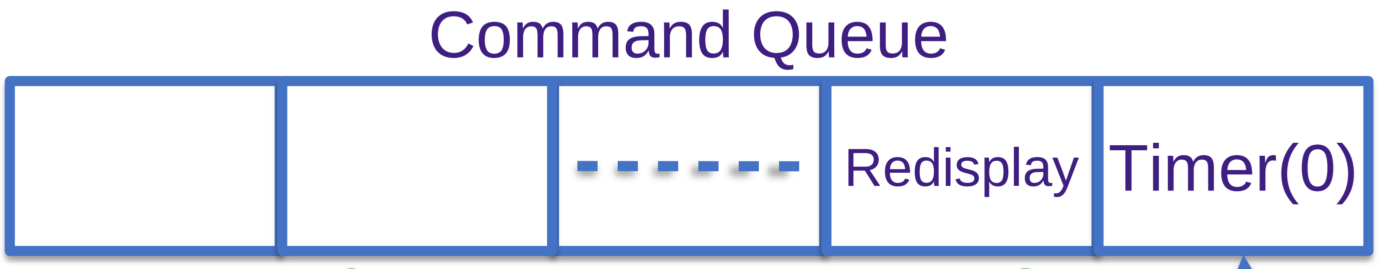
Animation



```
void display(){
    ...
    glutSwapBuffers();
}
```

```
void Timer(int id){
    glutPostRedisplay();
    glutTimerFunc(15, Timer, 0);
}
```

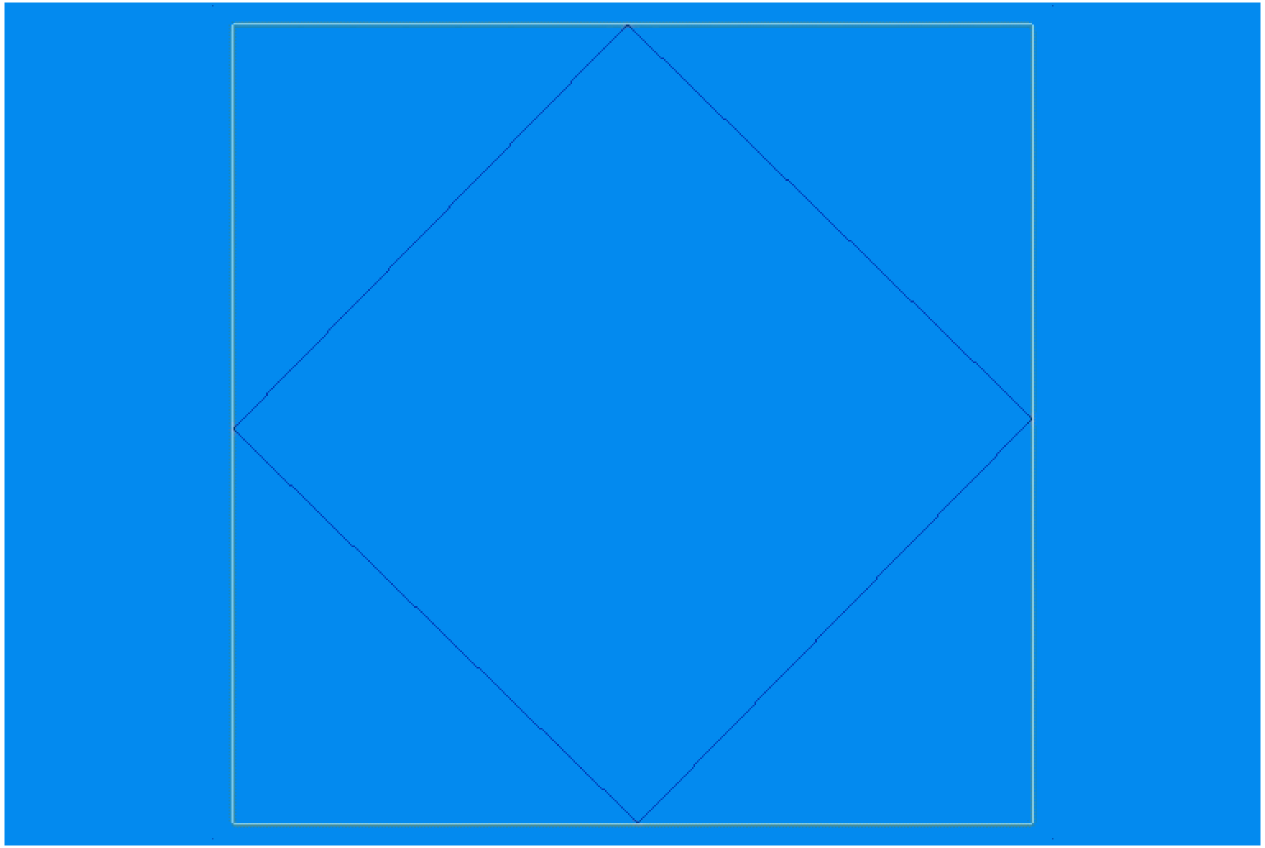
Animation



```
void display(){
    ...
    glutSwapBuffers();
}
```

```
void Timer(int id){
    glutPostRedisplay();
    glutTimerFunc(15, Timer, 0);
}
```

Animation



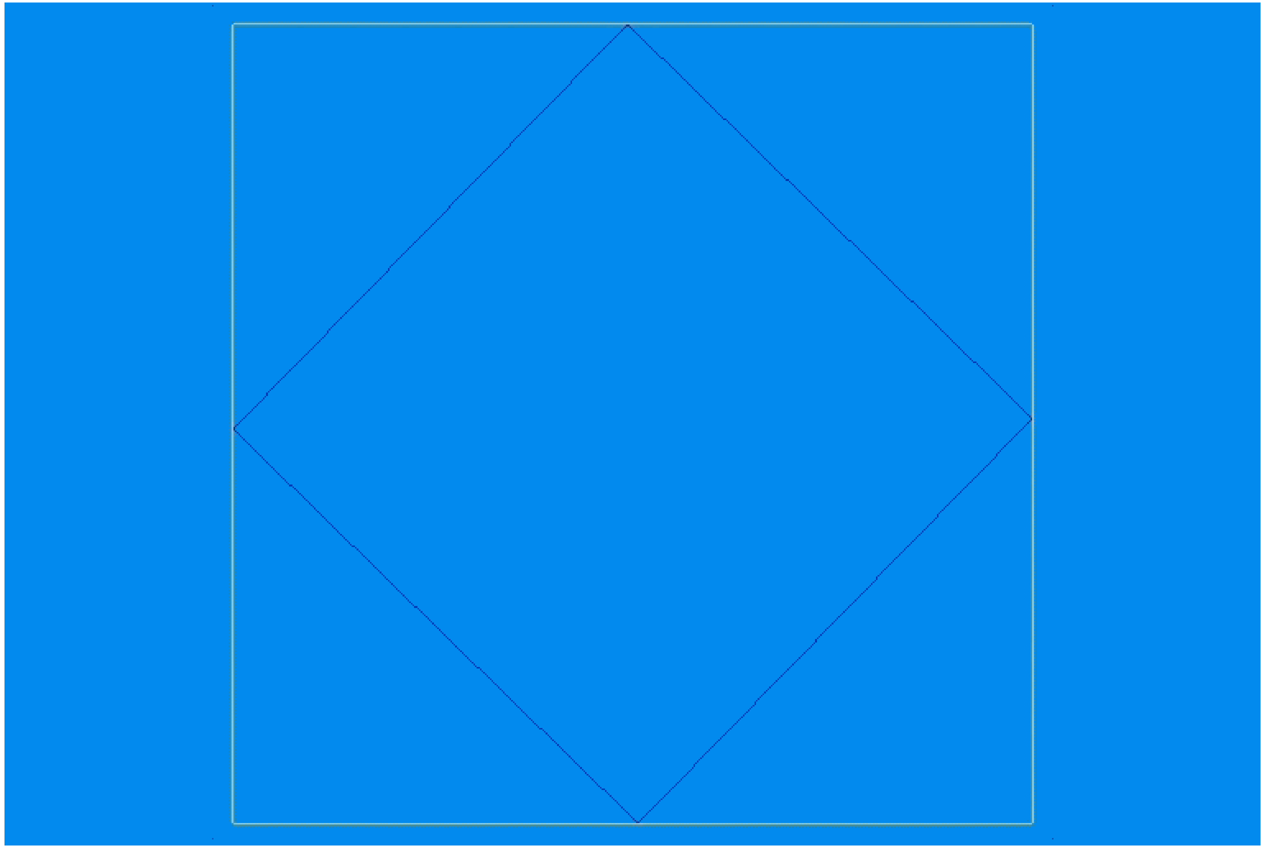
Animation

```
glutGet(GLUT_ELAPSED_TIME)
```

- Returns the elapsed time (in milliseconds) since Glut was initialized

```
float angle = glutGet(GLUT_ELAPSED_TIME) / 1000.0 * 45;  // 45° per second
transformObject(1.0f, Z_AXIS, angle, glm::vec3(0,0,0));
```

Animation



Week 3

Lab Activities

Week 3 Lab

- For the lab, see Hooman's material (with video)
- OpenGL examples covered:
  - 3D cube animation
  - Indexed draws
  - More animations
    - Automatic and semi-automatic/interactive

Week 3

End