



# INTRO TO OPENGL (4)

Prof. Dr. Mohamed Taher El-Sonni  
Eng. Mohamed el Shehaby

# GLOBAL VARIABLES

```
#include <GL/glut.h>
```

```
#include <stdio.h>
```

```
int physWidth = 600, physHeight = 600; // in pixels
```

```
int logWidth = 100, logHeight = 100; // logical units
```

```
int centerX = logWidth/2, centerY = logHeight/2;
```

```
int mouseX = centerX, mouseY = centerY;
```

- Physical width and height of the window

- Logical width and height of the window

- Logical mouse location with respect to the logical window (200,100)

# MAIN()

```
int main( int argc, char ** argv)
{
    glutInit( &argc, argv);
    glutInitDisplayMode( GLUT_DOUBLE |
                        GLUT_RGB);
    glutInitWindowPosition( 100, 100);
    glutInitWindowSize( physWidth, physHeight);
    glutCreateWindow( "problem demonstration");
    init();
    glutDisplayFunc( myDisplay);

    glutMainLoop();
}
```

```
void init()
{
    glClearColor( 0.0, 0.0, 1.0, 1.0); // COLOR
                                BACKGROUND

    glMatrixMode( GL_PROJECTION);
    gluOrtho2D( 0.0, logWidth, 0.0, logHeight);
}

void myDisplay()
{
    glClear( GL_COLOR_BUFFER_BIT);

    // .....

    glutSwapBuffers();
}
```

# DRAWING TEXT

```
void printSome(char *str,int x,int y) {  
    glColor3f (0, 1.0, 0.0);  
    glRasterPos2d(x,y);  
    for (int i=0;i<strlen(str);i++)  
        glutBitmapCharacter(GLUT_BITMAP_HELVETICA_12,str[i]);  
    glFlush();  
}
```

Indicates the logical location (x,y) where the text is to be drawn

# DRAWING TEXT

```
void printSome(char *str,int x,int y) {  
    glColor3f (0, 1.0, 0.0);  
    glRasterPos2d(x,y);  
    for (int i=0;i<strlen(str);i++)  
        glutBitmapCharacter(GLUT_BITMAP_HELVETICA_12,str[i]);  
  
    glFlush();  
}
```



**Draw a character str[i] with a certain font**

# MAIN()

```
int main( int argc, char ** argv)
{
    glutInit( &argc, argv);
    glutInitDisplayMode( GLUT_DOUBLE |
                        GLUT_RGB);
    glutInitWindowPosition( 100, 100);
    glutInitWindowSize( physWidth, physHeight);
    glutCreateWindow( "problem demonstration");
    init();
    glutDisplayFunc( myDisplay);
    glutMouseFunc(mouseClick);
    glutMainLoop();
}
```

```
void init()
{
    glClearColor( 0.0, 0.0, 1.0, 1.0); // COLOR
    BACKGROUND

    glMatrixMode( GL_PROJECTION);
    gluOrtho2D( 0.0, logWidth, 0.0, logHeight);
}

void myDisplay()
{
    glClear( GL_COLOR_BUFFER_BIT);
    printSome("Hello",10,20)
    glutSwapBuffers();
}
```

# MOUSE CALLBACK FUNCTION

(x,y) is the physical location of the mouse in pixels (w=600,h=400)

```
void mouseClicked(int btn, int state, int x, int y)
{
    if(btn==GLUT_LEFT_BUTTON && state==GLUT_DOWN) {
        mouseX = x; mouseX=0.5+1.0*mouseX*logWidth/physWidth;
        mouseY = physHeight- y; mouseY=0.5+1.0*mouseY*logHeight/physHeight;
    }
    if(btn==GLUT_RIGHT_BUTTON && state==GLUT_DOWN) {
        exit(1); // To Exit the Program
    }
    glutPostRedisplay();
}
```

# MOUSE CALLBACK FUNCTION

```
void mouseClicked(int btn, int state, int x, int y)
{
    if(btn==GLUT_LEFT_BUTTON && state==GLUT_DOWN) {
        mouseX = x; mouseX=0.5+1.0*mouseX*logWidth/physWidth;
        mouseY = physHeight- y; mouseY=0.5+1.0*mouseY*logHeight/physHeight;
    }
    if(btn==GLUT_RIGHT_BUTTON && state==GLUT_DOWN) {
        exit(1); // To Exit the Program
    }
    glutPostRedisplay();
}
```

1.0 multiplied to guarantee floating point computation  
+0.5 for rounding



# MOUSE CALLBACK FUNCTION

```
void mouseClicked(int btn, int state, int x, int y)
{
    if(btn==GLUT_LEFT_BUTTON && state==GLUT_DOWN) {
        mouseX = x; mouseX=0.5+1.0*mouseX*logWidth/physWidth;
        mouseY = physHeight- y; mouseY=0.5+1.0*mouseY*logHeight/physHeight;
    }
    if(btn==GLUT_RIGHT_BUTTON && state==GLUT_DOWN) {
        exit(1); // To Exit the Program
    }
    glutPostRedisplay();
}
```

Logical y =0 means bottom while  
physical y=0 means top.

# MAIN()

```
int main( int argc, char ** argv)
{
    glutInit( &argc, argv);
    glutInitDisplayMode( GLUT_DOUBLE |
                        GLUT_RGB);
    glutInitWindowPosition( 100, 100);
    glutInitWindowSize( physWidth, physHeight);
    glutCreateWindow( "problem demonstration");
    init();
    glutDisplayFunc( myDisplay);
    glutMouseFunc(mouseClick);
    glutMainLoop();
}
```

```
void init()
{
    glClearColor( 0.0, 0.0, 1.0, 1.0); // COLOR
    BACKGROUND

    glMatrixMode( GL_PROJECTION);
    gluOrtho2D( 0.0, logWidth, 0.0, logHeight);
}
void myDisplay()
{
    glClear( GL_COLOR_BUFFER_BIT);
    printSome("Hello",mouseX,mouseY)
    glutSwapBuffers();
}
```



**THANKS**