

Introduction to OpenGL

- ❑ OpenGL (**Open Graphics Library**): is a cross-platform application programming interface (API) for rendering 2D and 3D graphics
- ❑ OpenGL was developed by Silicon Graphics Inc. (SGI) in 1992
- ❑ OpenGL library provides functions for specifying graphics **primitives**, attributes, geometric transformations, viewing transformations, and many other operations
- ❑ Other Graphics Packages: Open Inventor, Virtual Reality Modeling Language (VRML), Java 2D, Java 3D, Renderman Interface, ... etc

1

Basic OpenGL Library

- ❑ Function in OpenGL basic library are prefixed with **gl**
 - glBegin, glEnd, glClear, glCopyPixels
- ❑ Constants are written in capital letters and begin with **GL** and underscore (**_**)
 - GL_2D, GL_RGB, GL_LINES, GL_POLYGON
- ❑ OpenGL uses special built-in data types
 - GLByte, GLshort, GLint, GLfloat, GLdouble

2

Related Libraries

- ❑ **OpenGL Utility (GLU)** provides routines for setting up viewing and projection matrices, describing complex objects,...
- ❑ All GLU function names start with the prefix **glu**
- ❑ **OpenGL Utility Toolkit (GLUT)** provides a library function for interacting with any screen windowing system
- ❑ All GLUT function names start with the prefix **glut**

3

```
#include <glut.h>

void init ()
{
    glClearColor (1.0, 1.0, 1.0, 0.0); // set display-window color to white
    glMatrixMode (GL_PROJECTION); // set projection parameters
    gluOrtho2D (0.0, 200.0, 0.0, 150.0);
}

void display ()
{
    glClear (GL_COLOR_BUFFER_BIT); // clear display window
    glColor3f (1.0, 0.0, 0.0); // set line segment color to red
    glBegin (GL_LINES);
        glVertex2i (180, 15); // specify line segment geometry
        glVertex2i (10, 145);
    glEnd ();
    glFlush (); // process all openGL routines as quickly as possible
}

void main (int argc, char** argv)
{
    glutInit (&argc, argv); // initialize GLUT
    glutInitDisplayMode (GLUT_SINGLE | GLUT_RGB); // set display mode
    glutInitWindowPosition (50, 100); // set top left display window position
    glutInitWindowSize (400, 300); // set display window width and height
    glutCreateWindow ("An Example OpenGL Program"); // create display window

    init (); // execute initialization function
    glutDisplayFunc (display); // send graphics to display window
    glutMainLoop (); // display everything and wait
}
```

4

Display Window Management Using GLUT

- ❑ **glutInit (&argc, argv);** initializes GLUT
- ❑ **glutInitDisplayMode (GLUT_SINGLE | GLUT_RGB);** sets display options to single buffering and RGB color. Options are combined with logical or |
- ❑ **glutInitWindowPosition (50, 100);** sets initial location for the top-left corner of display window
- ❑ **glutInitWindowSize (400, 300);** sets initial pixel width and height for display window
- ❑ **glutCreateWindow ("An Example OpenGL Program");** gives a caption for the title bar

5

Display Window Management Using GLUT

- ❑ **glutDisplayFunc (display);** assigns our picture to display window. In this example, the code for generating the picture is in the function display
- ❑ **glutMainLoop ();** This must be the last function in our program. It displays the picture and puts the program into an infinite loop that checks for input from a mouse or keyboard

6

A Complete OpenGL Program

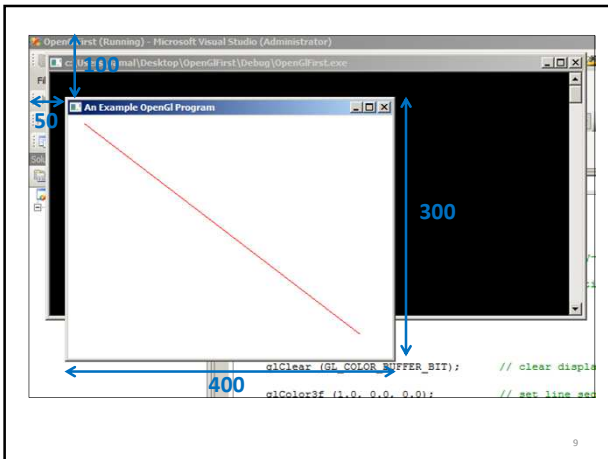
- ❑ `glClearColor (1.0, 1.0, 1.0, 0.0);` using RGB color values, sets the background color for display window to be white
red green blue alpha
- ❑ `glClear (GL_COLOR_BUFFER_BIT);` sets the color buffer to the values indicated in the `glClearColor`
- ❑ `glColor3f (1.0, 0.0, 0.0);` sets object color to be red. Suffix `3f` indicates three RGB color component using floating-point (`f`)

7

A Complete OpenGL Program

- ❑ `glMatrixMode (GL_PROJECTION);` sets the matrix mode to be projection mode
 - ❑ `gluOrtho2D (0.0, 200.0, 0.0, 150.0);` orthogonal projection is used to map the contents of 2D world coordinates to the screen. The x-coordinate ranges from 0.0 to 200.0 and the y-coordinate ranges from 0.0 to 150.0
- ```
glBegin (GL_LINES);
 glVertex2i (180, 15);
 glVertex2i (10, 145);
glEnd ();
```

8



9