CSL7450: Computer Graphics

Rajendra Nagar

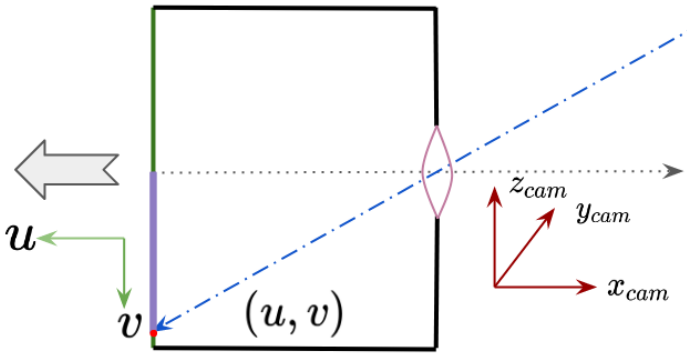
Trimester 2, 2020-2021
Indian Institute of Technology Jodhpur
sites.google.com/iitj.ac.in/csl7450/

December 28, 2020

A Gentle Introduction to OpenGL

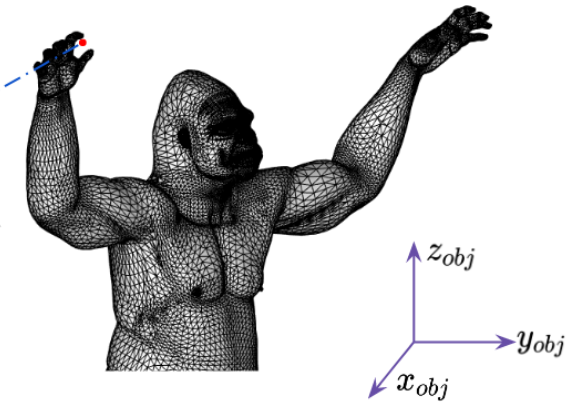
- OpenGL is an API for drawing 2D and 3D graphics.
- A basic library of functions is provided in OpenGL for specifying graphics primitives, attributes, geometric and viewing transformations, describing complex objects with line and polygon approximations, displaying quadrics and B-splines, processing the surface-rendering operations, and many other operations.
- In addition to the OpenGL basic (core) library, there are a number of associated libraries for handling special operations such as OpenGL Utility (GLU) and OpenGL Utility Toolkit(GLUT).

Viewport



Window

Camera Coordinate System



Object Coordinate System

A Gentle Introduction to OpenGL

- **glutInit(&argc, argv)** initializes GLUT .
- **glutInitDisplayMode(unsigned int mode)** specifies whether to use an RGBA or color-index color model. You can also specify whether you want a single- or double-buffered window. Finally, you can use this routine to indicate that you want the window to have an associated depth, stencil, multisampling, and/or accumulation buffer. For example, if you want a window with double buffering, the RGBA color model, and a depth buffer, you might call glutInitDisplayMode(GLUT_DOUBLE I GLUT_RGBA I GLUT_DEPTH).
- **glutInitWindowPosition(int x, int y)** specifies the screen location for the upper-left corner of your window.
- **glutInitWindowSize(int width, int size)** specifies the size, in pixels, of your window.
- **int glutCreateWindow(char *string)** a display window is to be created on the screen with a given caption for the title bar.
- **glutDisplayFunc** we need to specify what the display window is to contain. For this, we create a picture using OpenGL functions and pass the picture definition to the GLUT routine *glutDisplayFunc*, which assigns our picture to the display window.
- **glutMainLoop()** The display window is not yet on the screen. We need one more GLUT function to complete the window-processing operations. After execution of this command, all display windows that we have created, including their graphic content, are now activated.

A Gentle Introduction to OpenGL

glMatrixMode (GL_PROJECTION);
gluOrtho2D (0.0, 500.0, 0.0, 500.0);

- For our first program, we simply display a 2D point.
- To do this, we need to tell OpenGL how we want to "project" our picture onto the display window because generating a two-dimensional picture is treated by OpenGL as a special case of three-dimensional viewing.
- So, although we only want to produce a very simple two-dimensional point, OpenGL processes our picture through the full three-dimensional viewing operations.
- We can set the projection type (mode) and other viewing parameters that we need with these functions.
- This specifies that an orthogonal projection is to be used to map the contents of a two-dimensional rectangular area of world coordinates to the screen , and that the x-coordinate values within this rectangle range from 0.0 to 300.0 with y-coordinate values ranging from 0.0 to 300.0.
- Therefore, the GLU function **gluOrtho2D** defines the coordinate reference frame within the display window to be $(0.0, 0.0)$ at the lower-left corner of the display window and $(300.0, 300.0)$ at the upper-right window corner.
- Since we are only describing a two-dimensional object, the orthogonal projection $\left(\begin{bmatrix} x & y & z \end{bmatrix} \mapsto \begin{bmatrix} x & y & 0 \end{bmatrix}\right)$ has no other effect than to "paste" our picture into the display window.

A Gentle Introduction to OpenGL

- **void glClearColor(r, g, b, w)** Sets the current clearing color for use in clearing color buffers in RGBA mode.
- **void glClear(GLbitfield mask)** Although the *glClearColor* command assigns a color to the display window, it does not put the display window on the screen. To get the assigned window color displayed, we need to invoke this OpenGL function.
- **glColor3f (0.0, 0.4, 0.2)** In addition to setting the background color for the display window, we can choose a variety of color schemes for the objects we want to display in a scene.
- **void glBegin(GLenum mode)** Marks the beginning of a vertex-data list that describes a geometric primitive. The type of primitive is indicated by mode.
- **void glEnd(void)** Marks the end of a vertex-data list.
- **oid glPointSize(GLfloat size)** Sets the width in pixels for rendered points; size must be greater than 0.0 and by default is 1.0.
- **void glFlush(void)** Forces previously issued OpenGL commands to begin execution, thus guaranteeing that they complete in finite time.

A Gentle Introduction to OpenGL

void glVertex[234]{sifd}(TYPE coords);

Specifies a vertex for use in describing a geometric object. We can supply up to four coordinates (x, y, Z, w) for a particular vertex or as few as two (x, y) by selecting the appropriate version of the command. If we use a version that doesn't explicitly specify z or w, z is understood to be 0 and w is understood to be 1. Calls to **glVertex** are effective only between a *glBegin()* and *glEnd()* pair.