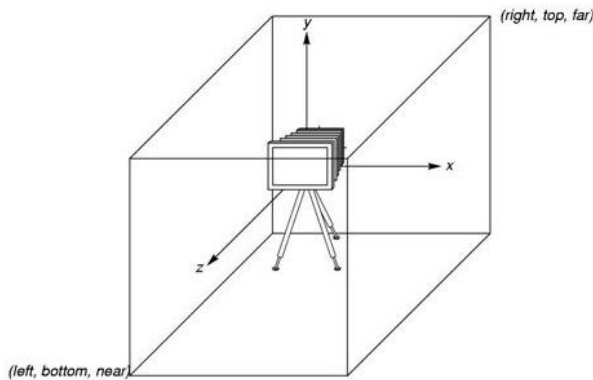


Question 1. (2 marks)

Describe the shape of the default OpenGL view volume.

Ans: The default OpenGL view volume is a cube (box) centred at the origin with side of length 2. The camera is placed at the origin and is looking in the negative z-direction.

**Question 2.** (6 marks)

Describe the components in the pipeline architecture of OpenGL, including a diagram that relates these components in the pipeline.

Ans: Objects passed to the pipeline are processed one at a time in the order they are generated by the application program. All steps can be implemented in hardware on the graphics card.

Vertex Processor: Vertex processor converts object representations from one coordinate system to another using matrix multiplications. Vertex processor also computes vertex colours.

Clipper and Primitive Assembly. Just as a real camera cannot see the whole world, the virtual camera can only see part of the world that is within the view volume. Objects that are outside this volume are clipped out of the scene. Vertices are collected into geometric primitives before clipping.

Rasterizer. For objects that are not clipped out, the appropriate pixels in the frame buffer must be assigned colors. For each primitive, rasterizer produces a set of fragments i.e. potential pixels. They have a location in the frame buffer and have colour, depth, and alpha attributes. Vertex attributes (colour, transparency) are interpolated over the objects by the rasterizer.

Fragment Processor: Fragments are processed to determine the colour of the corresponding pixel in the frame buffer. The colour of a fragment can be determined by texture mapping or by interpolation of vertex colours. Hidden surface removal is done at this stage.



Question 3. (2 marks)

Give two common applications of vertex shaders in OpenGL.

Ans:

(1) Geometric transformations

e.g. change relative location, rotation, and scale of objects/camera, 3D perspective transformation (make far objects smaller)

(2) Moving vertices

e.g. morphing, wave motion and deformation due to physical forces, particle effects - fire, smoke, rain etc

(3) Lighting

e.g. calculate shading colour using light and surface properties, special effects

Question 4. (6 marks)

(a) (3 marks) Explain clearly what the following functions do in OpenGL:

i) `glutInitDisplayMode (GLUT_SINGLE) ;`

Ans: Initializes the display mode using a single buffer as opposed to DOUBLE buffer.

ii) `glutMainLoop () ;`

Ans: Specifies that the program can now enter the (infinite) “event” handling loop.

(b) (3 marks) Briefly describe how the z-buffer algorithm works.

Ans: The z-buffer algorithm determines which objects are visible and should be displayed on the screen. It achieves this by placing the primitives (after vertex processing) in a z-buffer that determines which objects are closer to the camera along the projector (the ray originating from the camera). Objects that are closer will occlude other objects that are far on the projector ray. Z-buffer is also called **depth-buffer** and has the same spatial resolution as the colour buffers.

Question 5. (4 marks)

What is a callback function in the context of interactive programs? Give two examples of callbacks in OpenGL, and explain the purpose of each.

Ans: Callback functions are user-defined functions for handling input events. These functions would automatically be called by the windows manager when the events occur. They must be registered before the program enters the event handling loop.

Examples are (1) mouse callback function and (2) keyboard callback function

END OF PAPER
