1. What do the contrasting stories of the Brooklyn Bridge and the Quebec Bridge teach us?

Engineering grounded in reality. Good design must follow the laws. If you break the laws in your design then bad things happen. On the other hand, a good design can be around for a very long time.

2. Processing objects 1-by-1 with a graphics pipeline has a major advantage and also a major disadvantage. Name them.

Advantage: Graphics pipeline are more efficient since the objects are processed in parallel.

Disadvantage: Interactions between the objects, where light reflecting from one object makes another one brighter, cannot be modeled.

3. Why must you remove objects that are outside the view volume?

If don’t remove them, the objects behind the camera will be projected into the image.

4. Does a straightline always remain straight after perspective projection? Give a reason; not just yes/no.

Yes.

Prof: Given the line equation in 3D

\[
\begin{pmatrix}
X \\
Y \\
Z
\end{pmatrix} = \begin{pmatrix}
X_0 \\
Y_0 \\
Z_0
\end{pmatrix} + \alpha \begin{pmatrix}
V_x \\
V_y \\
V_z
\end{pmatrix}
\]

(1)

The perspective projection project the 3D coordinate \((X,Y,Z)^T\) to image coordinate \((x,y)^T\):

\[x = f \frac{X}{Z} \text{ - (2)} \text{ and } y = f \frac{Y}{Z} \text{ - (3)} \text{ (f is the focus length)}\]

Combining (1),(2) and (3):

\[x = f \frac{X}{Z} = \frac{X_0 + \alpha V_x}{Z_0 + \alpha V_z} = \frac{X_0}{Z_0} + \frac{\alpha(Z_0 V_x - X_0 V_z)}{Z_0(Z_0 + \alpha V_z)} \text{ and }\]

\[y = f \frac{Y}{Z} = \frac{Y_0 + \alpha V_y}{Z_0 + \alpha V_z} = \frac{Y_0}{Z_0} + \frac{\alpha(Z_0 V_y - Y_0 V_z)}{Z_0(Z_0 + \alpha V_z)} \]
So, \[ \frac{x - X_0}{Z_0} \] \( y - Y_0 \) \[ Z_0 \] = \[ \frac{(Z_0V_x - X_0V_z)}{(Z_0V_y - Y_0V_z)} \]

This is a 2D line equation: \( \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \frac{X_0}{Z_0} \\ \frac{Y_0}{Z_0} \end{pmatrix} + \beta \begin{pmatrix} Z_0V_x - X_0V_z \\ Z_0V_y - Y_0V_z \end{pmatrix} \)

5. Why do we want to make projection into a matrix multiplication?

Multiple projections (perspective projection, orthogonal projection, etc) can be systematically reduced into a single matrix multiplication. Computer hardware is also particularly efficient at matrix computation.

6. Why does OpenGL contain primitives, like triangle fans, that are not necessary (since you could draw the triangles separately)?

Performance, since using triangle fans prevents repeating unnecessary vertices. I reduce the computational cost.

7. Why do we consider color to have three dimensions, such as RGB? There are an infinite number of visible wavelengths.

Because the RGB model is representative of how the human visual system works. We see color as different intensities of red, green and blue. Therefore we can represent all visible colors using this 3 dimensional representation.

8. What is the event loop?

The event loop continuously listens for events to occur. When such an event occurs (input from mouse, keyboard, timer, etc...) the appropriate callback is executed.

9. What does this do: glutDisplayFunc(display)

It registers the display callback.

10. If you don't want to use the entire window for your image, what subroutine do you call?

    glViewport

11. What does this do: glMatrixMode(GL_PROJECTION);

    All subsequent transformations act on a projection matrix

12. Since the mydisplay routine takes no arguments, how do you get data from the main program to it?
Global Variables

13. What rendering technique handles reflections and refractions well?

Ray tracing

14. Use the vector formulation or your intuition to rotate the point (0,1,0) by 60 degrees about the axis (0,1,0).

The point is on the rotation axis, so it remains the same: (0,1,0).

15. What's a disadvantage of standards?

They ignore the differences between hardware. It makes lag the state-of-the-art both in hardware and in users’ requirements.

16. What is an OpenGL fragment?

Potential pixel

17. By now most of you know the advantage of using XOR to write a cursor into the color buffer. So, why don't graphics systems do this all the time? E.g., the system I'm now using doesn't.

XOR cannot draw specific color, since the result color also depends on the current color buffer.

18. What's the normal to the plane 5x+2y+3z=1? Your answer should be normalized.

\[ N = (5, 2, 3)^T \]

Normalize: \[ \sqrt{5^2 + 2^2 + 3^2} = \sqrt{38} \]

Normalized norm = \[ \left( \frac{5}{\sqrt{38}}, \frac{2}{\sqrt{38}}, \frac{3}{\sqrt{38}} \right)^T \]


1) All affine transformations can be represented using matrix multiplication

2) Computer hardware is very fast at computing matrix arithmetic

3) Can represent points at infinity
4) Easy to carry out successive transformations in 3D

20. How is the Z buffer used to make closer objects display in front of farther ones?
The object is processed one by one. And the Z-buffer keeps updating if the new fragment has greater depth than the current depth in the Z-buffer. So, finally the Z-buffer saves the closest fragment and save its color in the corresponding color buffer.

21. Consider this sequence of calls:
   glColor3f(1.,1.,1.);
   glColor3f(0.,1.,0.);
   glVertex3f(1.,1.,1.);
   glVertex3f(2.,2.,2.);
   a. What color is the vertex (1,1,1)?
      Green
   b. What color is the vertex (2,2,2)?
      Green

22. What's the point of using two color buffers, one to draw into, while the other is being displayed?
   If we just use one buffer, when we change the content of the frame buffer during refreshing, maybe only part of the buffer is updated and we may see partially drawn display. Double buffer can handle these by displaying the front buffer and drawing the back buffer.

23. Why are three color buffers even better?
   It can avoid tearing caused by swapping the foreground and background buffers in the middle of refreshing the image and the two buffers have different images. Using 3 buffers, you can display from the 1st buffer, have the 2nd buffer ready to swap with the 1st in the next vertical retrace, and meanwhile compute into the 3rd buffer.

24. Give the 4x4 homogeneous matrix for a 60 degree rotation about the Y axis.
   \[
   \begin{pmatrix}
   \cos(60) & 0 & \sin(60) & 0 \\
   0 & 1 & 0 & 0 \\
   -\sin(60) & 0 & \cos(60) & 0 \\
   0 & 0 & 0 & 1 \\
   \end{pmatrix}
   \]

25. To deal with multiple functions for a similar purpose, C++ and OpenGL take different approaches: C++ use overloading while OpenGL defined a number
of functions which differ from each other in the last letter, for example, glVertex3f, glVertex2i, glVertex3dv all create vertex but the first take different data types as input. What is an advantage of each approach?

C++ approach is easier for programmer, since you don’t need to change the code if the data type changes. OpenGL approach is more efficient. The programmer can have better control of the program, and use suitable data type for specific purpose.

26. How does the following code guarantee the correctness of the resize process?

It makes the aspect ratio of the viewpoint equal to the aspect ratio of the viewing volume.

27. Since the Z buffer looks so useful, why is it not enabled by default?

Z-buffer is not necessary for all the cases (such as 2D case). So, for efficiency, it is not enabled by default.

28. Why do we like to group different physical input devices into the same logical input device?

We just need to care about “what is inputed” in our code. And we don’t need to know “how it is inputed”. So, when the input device is changed, we can use the same code.