A vector-based display .

1

A pixel-based display .

2

In a graphics pipeline, what is a fragment?

3

What is the fundamental difference between a fixed and a programmable graphics pipeline?

4

Describe OpenGL in one sentence.

5

oversees the development of OpenGL.

6

What kind of matrix does OpenGL use to encode 3D transformations?

7

Why do we use 4x4 transformation matrices in 3D graphics?

8

A pixel-based display approximates shapes.

2

A vector-based display can draw lines, text and points.

1

There are parts of a programmable pipeline that the user must program themselves.

4

A candidate pixel, which or may not end up being displayed.

A loose definition of a fragment is a "potential pixel". The fragment (x,y) will be drawn on the screen only if it meets certain criteria, such as, it's nearer to the viewer than the pixel already recorded (if any) for position (x,y).

3

Khronos oversees the development of OpenGL.

6

An API for doing 3D computer graphics

5

They can encode translations as well as scales and rotations.

8

4×4

You can encode 3D scaling and 3D rotations in a 3x3 matrix, but if you want to include translations too, you need a 4x4.

7

In 2D a rotation is defined with respect to a point. In 3D a rotation is defined with respect to ________.

9

Assuming we represent 3D points using column vector notation, if I take a point P and first transform it by matrix M1, then matrix M2, and finally matrix M3, whats the expression that gives P', the final value of P?

10

Given a matrix A and its inverse B, what is the result P' of applying the composite transformation (A x B) to a point P?

11

What is the effect on a vector V of normalising it?

12

Given two normalised 3D vectors V1 and V2, what gives the vector V3 which is perpendicular to both V1 and V2?

13

What is the purpose of the Z-buffer?

14

________ is the process of splitting a complex polygon into a set of separate convex ones. ________ an odd-shaped polygon guarantees that it can be rendered correctly.

15

What is the the purpose of "viewing" a model in computer graphics?

16

$$P' = M3 \times M2 \times M1 \times P$$

10

In 2D a rotation is defined with respect to a point. In 3D a rotation is defined with respect to a 3D vector.

9

Scaling V so that it has a magnitude of 1, without affecting its direction.

12

$$P' = P$$

Multiplying a matrix by its inverse results in the identity transformation, which transforms a point to itself (i.e., doesn't change it).

11

To almost guarantee that only pixels nearest to the camera are displayed. This is not always true as there can be some problems, such as Z-fighting.

14

$$V3 = V1 \times V2$$

Where × means cross product.

13

To enable the user to control how parts of the model are seen on screen.

16

Tessellation is the process of splitting a complex polygon into a set of separate convex ones. Tessellating an odd-shaped polygon guarantees that it can be rendered correctly.

15

In a simple 2D window-to-viewport view, we have 3 steps: M1 shifts the window to the origin; M2 scales the window to be the same size as the viewport; M3 shifts the viewport to the desired place. The overall view transformation is ________.

17

What best describes "the duality of modelling and viewing"?

18

When defining the coordinate system for the the camera, why do we not create a system with axes Q, V, and F, where Q = normalise(V x F)? (assume V and F are normalised).

19

What are the two main classes of projections?

20

In perspective projection a point (x, y, z) *will project to* (xp, yp, zp) *on a projection plane. By what method can we find* (xp, yp, zp)*?*

21

OpenGL combines a matrix for applying modelling transformations, M, with a matrix V for applying the camera. In which logical order are they applied to coordinates?

22

What is the view volume in parallel projection?

23

What is the main purpose of projection normalization?

24

Moving the model by some transformation is equivalent to applying the inverse transformation to the camera viewing it.

18

In a simple 2D window-to-viewport view, we have 3 steps: M1 shifts the window to the origin; M2 scales the window to be the same size as the viewport; M3 shifts the viewport to the desired place. The overall view transformation is $M3 * M2 * M1$.

17

Orthographic and perspective.

20

Because it would only work if V, which is specified by the user, is orthogonal to F. It may not be.

19

First M and then V.

22

Similar triangles.

21

It allows us to keep the z coordinates, which would otherwise have been set to the z-value of the projection plane.

24

A tetrahedron.

23

What is themselves difference between local and global illumination models?

25

makes materials look dull and matte.

26

Ambient light is an approximation of the .

27

What is the function of the n exponent in the Phong specular term?

28

Does the use of Ks allow us to render metal accurately?

29

In Gouraud shading, the is interpolated along a scan line.

30

In Phong shading, the is interpolated along a scan line.

31

What is Mach banding?

32

Diffuse reflection makes materials look dull and matte.

26

A gobal model takes into account light interactions between objects; a local model doesn't.

25

It models how shiny a surface is. n small means not very shiny; n large means very shiny.

28

Ambient light is an approximation of the overall light level in an environment.

27

In Gouraud shading, the pixel colour is interpolated along a scan line.

30

No, because it doesn't capture the complex interactions between light and metals.

29

The illusion that changes of intensity are greater than they actually are.

32

In Phong shading, the normal vector is interpolated along a scan line.

31

What is the main purpose of mip-mapping?

What is the basic principle behind bump mapping?

To make a surface look bumpy by altering normals during rendering.

To deal with mismatches between texel and pixel resolutions.