

CS445 Exam 2 Solutions

Fall 2014

1. (max = 15)	5. (max = 21)
2. (max = 8)	6. (max = 16)
3. (max = 10)	7. (max = 16)
4. (max = 14)	
Final Score: (max=100)	

Please try to write legibly. The instructor's complete failure to decipher what you write will be considered an incorrect answer.

In all of the following problems, assume the following terminology

Scale: $\mathbf{S}(s_x, s_y, s_z)$

Translation: $\mathbf{T}(x, y, z)$

Rotation: $\mathbf{R}_x(\theta), \mathbf{R}_y(\theta), \mathbf{R}_z(\theta)$

Object coordinate system: **OCS**

World coordinate system: **WCS**

Camera coordinate system: **CCS**

Model matrix: **M**

View matrix: **V**

Rotational component of V , i.e. the view rotation matrix: \mathbf{V}_{rot}

Location of the camera expressed in the WCS: **Eye**

A point \mathbf{P} expressed in the OCS will be written as \mathbf{P}_{OCS} . Similarly, points in the WCS or CCS will be written as \mathbf{P}_{WCS} and \mathbf{P}_{CCS} , respectively.

1. (5 pts each, 15 pts total) **Fly-Through Navigation:** Suppose you are flying a plane through a scene with the camera attached to the front of plane (as was done in the lab).
- a) In the lab, how did you update V_{rot} and Eye in order to turn your plane some small angle Θ to the right?

Eye does not change.

$$V_{rot} \rightarrow R_y(\Theta) V_{rot}$$

- b) How did you update V_{rot} and Eye in order to move the camera forward by some small amount α ?

V_{rot} does not change

$$\text{Eye} \rightarrow \text{Eye} - \alpha V_{rot} [2]$$

Where $V_{rot} [2]$ is the third row of V_{rot} and represents n where
 $n =$ the direction of the camera's z axis as expressed in the WCS

- c) How did you calculate the view matrix, V , from Eye and V_{rot} ?

$$V = V_{rot} T(-\text{Eye})$$

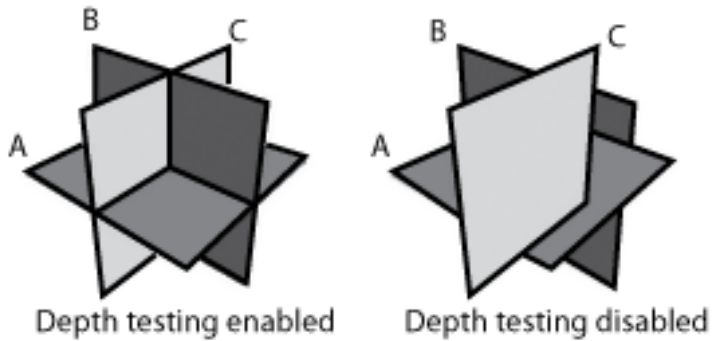
2. (8 pts total) **Blending:** In OpenGL, suppose you render 3 perpendicular planes, A, B, and C. If you have enabled depth testing, you get the picture on the left. If you disable depth testing, you get the picture on the right.

a) (2 pts) Based on the pictures, which plane did OpenGL draw first, second, and last?

First: __B__

Second: __A__

Last: __C__



b) (6 pts) Explain why the images look the way they do. Please use complete sentences.

When depth testing is enabled, the final pixel will be the fragment which is closest to the camera.

When depth testing is disabled, the final pixel will be whichever fragment is calculated last.

3. (5 pts each, 10 pts total) **Gouraud vs Phong Shading:** *Please answer in complete sentences.*
a) What is the difference between Phong and Gouraud Shading?

In the fragment shader, Phong calculates the shading at each fragment based on an interpolated normal.

In the vertex shader, Gouraud calculates the shading at each vertex and interpolates these over the fragments.

- b) Which shading method (Gouraud vs Phong) is generally better and why?

Phong is better. Consider a triangle. Rather than calculating the shading at only 3 points (i.e. at the vertices) and interpolating the results over the many fragments, the shading in Phong is calculated at many points (i.e. the fragments) based on the interpolated normal. This is more likely to capture the specular color which is highly sensitive to small changes in the normal.

4. (14 pts total) **Shaders:** *Please answer in complete sentences.*

a) (3 pts) What is a fragment?

A fragment is a “preliminary” pixel. The final color of a given pixel is a function of the many fragments that overlap at that pixel’s location. Each of these overlapping fragments is the result of the rasterization of some triangle that covers that pixel.

b) (3 pts) What is the difference between a uniform variable and attribute variable?

An attribute variable is associated with each vertex. The uniform variable is applied to each object (or draw command)

c) (3 pts) What calculations are typically done in the vertex shader and what are done in the fragment shader?

Typically, the vertex shader applies the model, camera, and projection transformations to each vertex.

The fragment shader does the shading calculation and must output a color.

d) (5 pts) What are at least 5 operations that opengl performs between the vertex and fragment shader?

After the vertex shader and before the fragment shader, opengl does the following:

1. Perspective division
2. Primitive assembly
3. Backface culling
4. Viewport transform
5. Rasterization
6. Clipping

5. (3 pts each. 21 pts total) **Coordinate Transformations, Part 1:** Match each of the items (a-g) with one of choices A-J.

a) $(0,0,0,1)^T_{CCS}$ Ans: ____ C ____

b) $(0,0,0,1)^T_{OCS}$ Ans: ____ G ____

c) $(0,0,0,1)^T_{WCS}$ Ans: ____ E ____

d) $V(0,0,0,1)^T$ Ans: ____ F ____

e) $M(0,0,0,1)^T$ Ans: ____ H ____

f) $MV(0,0,0,1)^T$ Ans: ____ J ____

g) $VM(0,0,0,1)^T$ Ans: ____ I ____

Fill in the above answers with one of the letter choices below

- A) The location of the camera as represented in the OCS.
- B) The location of the camera as represented in the WCS.
- C) The location of the camera as represented in the CCS.
- D) The origin of the world as represented in the OCS.
- E) The origin of the world as represented in the WCS.
- F) The origin of the world as represented in the CCS.
- G) The center of the object as represented in the OCS.
- H) The center of the object as represented in the WCS.
- I) The center of the object as represented in the CCS.
- J) None of the above - The expression has no obvious interpretation.

6. (16 pts total) **Coordinate Transformations, Part 2:** How do you implement the following actions? Your answer should be expressed as the product of one or more transformations (e.g. S, T, R, M, V, V_{rot}) multiplied by P_{OCS} , where P_{OCS} is some point in the **OCS**. Please take your time to read the question carefully – these are tricky!

Example:

A rotation of P_{OCS} by Θ about the OCS x-axis, as expressed in the OCS.

$$Ans = R_x(\Theta) P_{OCS}$$

- a) (3 pts) A rotation of P_{OCS} by Θ about the **OCS** x-axis, as expressed in the CCS.

Ans: _____ $V M R_x(\Theta) P_{OCS}$

- b) (3 pts) A rotation of P_{OCS} by Θ about the **WCS** y-axis, as expressed in the CCS.

Ans: _____ $V R_y(\Theta) M P_{OCS}$

- c) (5 pts) A uniform scale by 2 of P_{OCS} about a point Q_{OCS} , as expressed in the WCS.

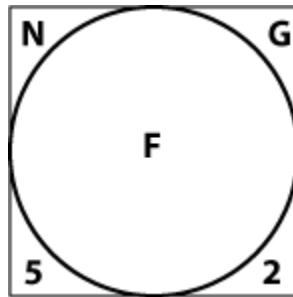
Ans: _____ $M T(Q_{OCS}) S(2) T(-Q_{OCS}) P_{OCS}$

- d) (5 pts) A uniform scale by 2 of P_{OCS} about a point Q_{WCS} , as expressed in the CCS.

Ans: _____ $V T(Q_{WCS}) S(2) T(-Q_{WCS}) M P_{OCS}$

7. (4 pts each, 16 pts total) **Texture Coordinates:** Suppose you want to use the image below as a texture on a square (e.g. a quad modeled as 2 triangles, although this doesn't matter for this problem).

image to be used as a texture object:



For each of the following possible texture coordinates, draw how the texture will look on the square. Assume that the texture wrap parameter is set to repeat in both dimensions.

(1,1) (1,0)

(0,1) (0,0)

(0,2) (2,2)

(0,0) (2,0)

(0,.5) (1,.5)

(0,0) (1,0)

(1,1) (0,1)

(1,0) (0,0)

texture coordinates