CSC 471 Midterm 1 – Fall 2017 - Section 1

Name:____

READ ME FIRST

- Work individually! You may use a calculator.
- Don't spend too much time on any one problem.
- You will have 60 minutes to finish the exam. 4:10pm to 5:10pm
- **Be neat** - I can only grade what I can read.
- Show how you got your answers! The decimal answer is worth 0 points. The process for finding the answer is what I want to see.
- **When in doubt, write down your assumptions.**
- Feel free to use the backs of pages to do work, just make sure it's clear where I can find your answer.

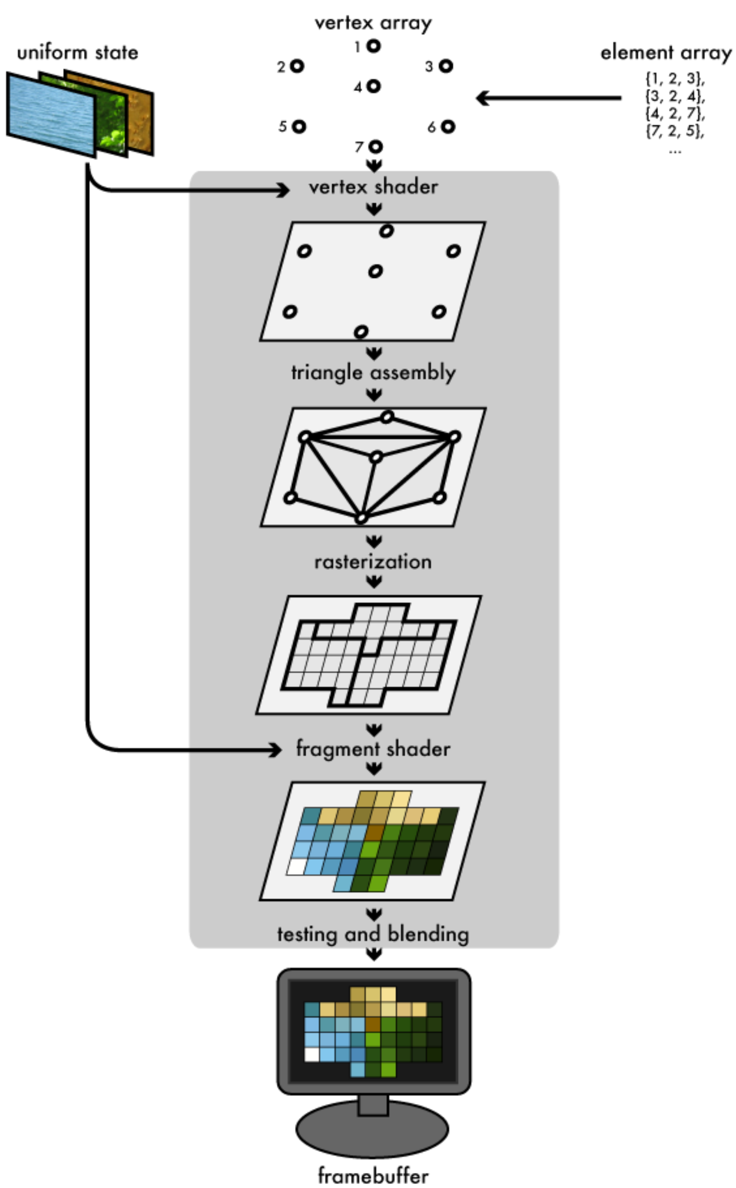
1	15 pts	Short answer	
2	10 pts	Vectors	
extra credit	2 pts		
3	30 pts	2D transform matrices	
4	15 pts	Transforms	
5	10 pts	Rasterization	
	80 pts	Grand total	

(1) Short answer/ true & false questions (20 pts)

(a) (1 pt) In a very general sense, the GPU can be viewed as a SIMD machine that allows a program to run the same 'vertex shader' program on multiple different vertices in parallel and then run a 'fragment shader' program on multiple fragments in parallel, thus speeding up the process of rendering computer graphics

True **False**

(b-f) Refer to the following figure and fill in the missing information – short answers (2 pts each):

<table>
<tr><td rowspan="5">
</td><td>(b) Describe what data is typically transferred from the CPU to the GPU for graphics applications:</td></tr>
<tr><td>(c) Describe what computational task a vertex shader typically does?</td></tr>
<tr><td>(d) Describe in your own words, what happens during rasterization:</td></tr>
<tr><td>(e) Describe what computational task a fragment shader typically does?</td></tr>
<tr><td>f) What are the value ranges of the geometry data at the end of the graphics pipeline, i.e. {x, y}?</td></tr>
</table>

(g) (4 pts) Assume you're making a game in which the player must sneak into a palace. One of the palace guards is standing at {-1, 5}. The guard hears a noise and looks towards a shrub at {3, 2}. The guard has a 180° field of view and will be able to see anything in front or directly to the side of its current view, but can't see behind itself.

If the player is standing out in the open at {1, 10}, can they be seen by the guard?

Show your work with math. As a hint, recall that for two positions **a** and **b**, the expression **b - a** gives you the vector from **a** to **b**. A diagram may be helpful, but you should use the value of some mathematical expression to answer the question.

(2) Vectors (10 pts)

Given the following vectors: $\mathbf{v}^T = [\ 4, 2, 3\ ]$ and $\mathbf{u}^T = [\ 6, -1, -3\ ]$ Compute**:**

(1) **(2 pts) v + u =**

(2) **(2 pts) v · u =**

(3) **(2 pts)** If **w = v – u,** What is the length of the vector **w**?

(4) **(4 pts)** Write the normalized form of **w** <u>(from the part 3)</u> (i.e. write **w** as a unit length vector).

(5) **(2 pt extra credit):** draw the vector -1***w** (accurately depicting length (ratio) and direction) as some part of a creature (make it clear which part of the creature is the vector) – you may define the units (i.e. inches, feet, etc.)

(3) 2D transform matrices (30 pts)

Given the following 2D transform matrices:

$$m_0 = \begin{bmatrix} 1 & 0 & -3 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} m_1 = \begin{bmatrix} 0.707 & -0.707 & 0 \\ 0.707 & 0.707 & 0 \\ 0 & 0 & 1 \end{bmatrix} m_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} m_3 = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

a) Name what type of 2D transformation is associated with each matrix and say something about the magnitude of the transform for x and y, or angle. **(4 pts total)**

m0:

m1:

m2:

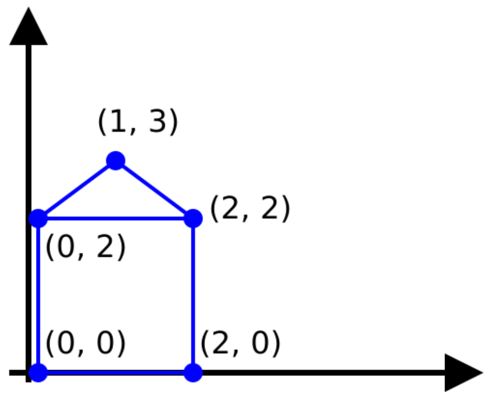
m3:

b) If these are 2D transforms, why are they 3x3 matrices? (Write 1-2 sentences) **(2 pts)**

(c) **Carefully compute** m0*m1 (that is, write out the composite matrix) (4 pts):

(d) **(13 pts total)**
(4 pts) **Draw** the result of applying the composite matrix (from part (c) – i.e. m0*m1) to the following figure (draw the entire house transformed and include the x and y axes for reference).

(3 pts each) **Include coordinate labels for your completed drawing for the updated points {0, 2}, {1, 3} and {2, 2}** (Be careful about how you represent the 2D points as vectors of length 3.

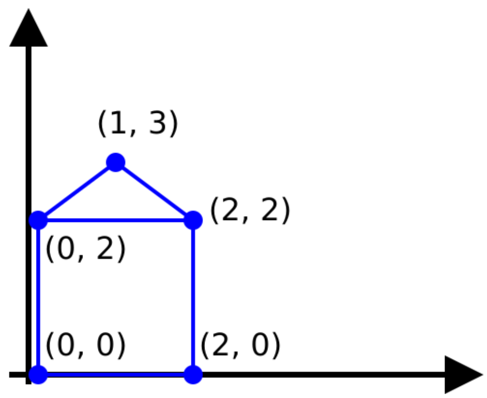


(3 pts) {0, 2} :

(3 pts) {1, 3} :

(3 pts) {2, 2} :

(e) Now, only **draw** the result of applying two transforms: m2*m3 to the same figure (feel free to compute the composite matrix if that helps you, but it is not required). Be sure that your drawing includes a representation of the axes and dimensions to clarify the house's exact final position and scale: **(7 pts)**

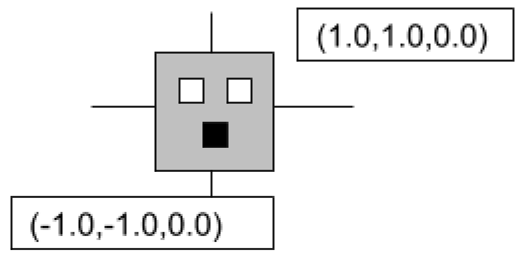


(**4) Transforms (15 pts)**
Assuming you have the following functions:

*mat4 **scale**(float sx, float sy, float sz) {… } : returns a scale matrix*
*mat4 **rotate**(float angle, float ax, float ay, float ax) {…} : returns a rotation matrix by the given angle and axis [ax, ay, az]*
*mat4 **translate**(float tx, float ty, float tz) {…} : returns a translation matrix*
*And assume the **operator *** is defined for matrix multiplication as expected*

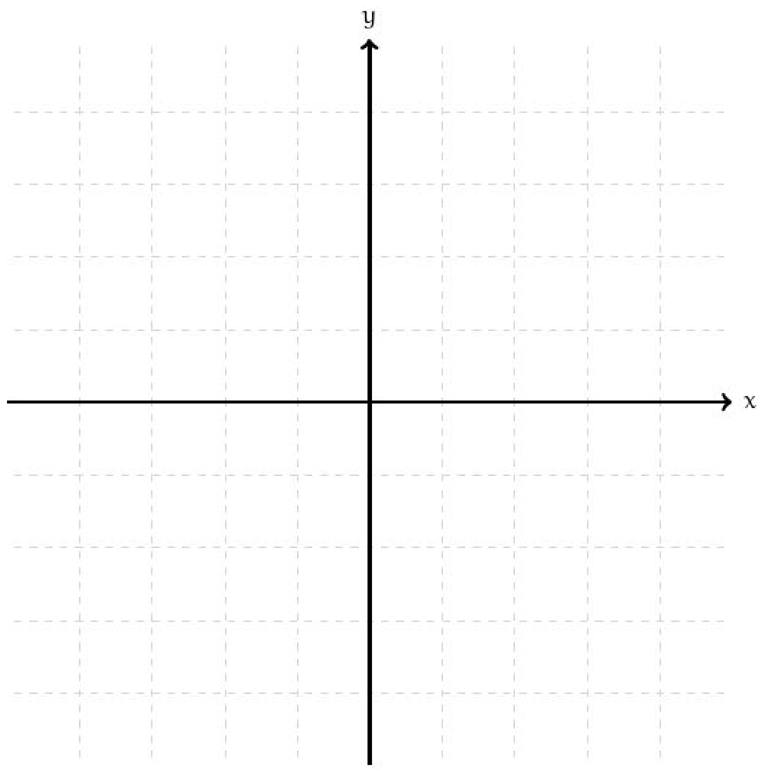
Carefully draw the result of the following OpenGL/GLSL code assuming that the DrawRobotFace() function draws the complete image below (i.e. one grey box with sides of length 2 with three small sub-boxes inside with sides of length 0.5: white eyes and a black mouth). Recall that rotations are specified as counter-clockwise.

Carefully read all the code below before drawing and be sure that it is clear what the final drawing will look like (a mat4 is a GLSL/glm 4x4 matrix – as expected):



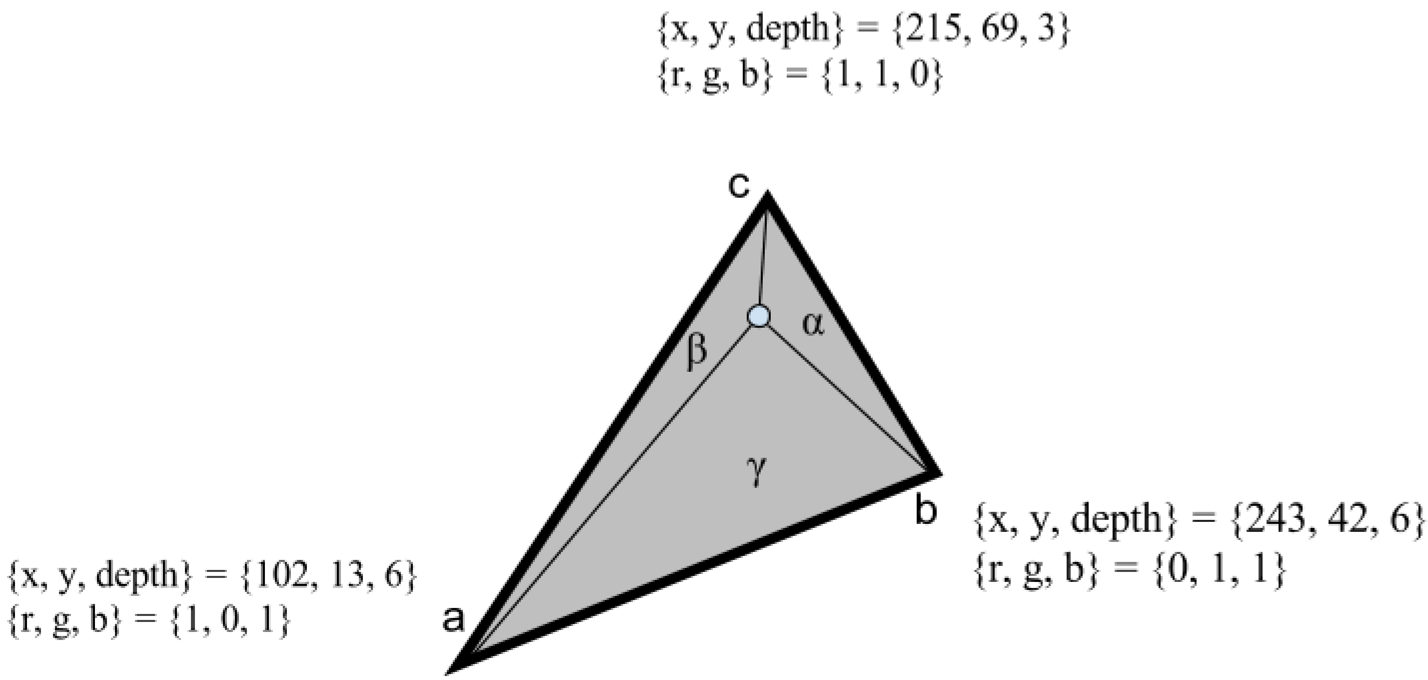

```
/* set up the first matrix */
mat4 Scale1 = scale(2, 1, 1);
mat4 Trans1 = translate(-2, 0, 0);
mat4 Rot1 = rotate(-90, 0, 0, 1);
mat4 Model1 = Trans1 * Rot1 * Scale1;

/* send matrix to the vertex shader */
glUniformMatrix4fv(prog->getUniform("MV"), 1, GL_FALSE, Model1);
/* draw */
DrawRobotFace();

/* Set up the second matrix */
mat4 Scale2 = scale(1, 1, 1);
mat4 Trans2 = translate(1, 1, 0);
mat4 Rot2 = rotate(45, 0, 0, 1);
mat4 Model2 = Trans2 * Rot2 * Scale2;

/* send matrix to the vertex shader */
glUniformMatrix4fv(prog->getUniform("MV"), 1, GL_FALSE, Model2);
/* draw */
DrawRobotFace();
```

Complete your drawing on the next page


y
x

(5) Rasterization (10 pts total):

If you have a triangle converted to window coordinates with the following coordinates, (including depths and colors) – **given the associated Barycentric coordinates** (i.e. do not compute them, use what is given):



(a) **(3 pts)** What are the x y z coordinates for the interpolated point on the triangle, given by following barycentric coordinates?:

$\{x, y, z\} = ?$

$\{\alpha, \beta, \gamma\} = \{0.2, 0.2, 0.6\}$

(b) **(3 pts)** What is the interpolated color?:

(c) **(4 pts)** Assuming the current value stored in the depth buffer/z-buffer for the associated pixel is 5.5, would the frame buffer/color buffer be updated with the new color? **Assuming the z values specified are distances measured from the camera – thus smaller values are closer to the camera.**