Midterm

This is a preview of the published version of the quiz

Started: May 19 at 8:49am

Quiz Instructions

Question 1 — 2 pts

We've seen numerous coordinate systems in class to this point. Recall *eye coordinates*, which had two properties that were emphasized:

1.
2.

Question 2 — 7 pts

For the following shader, identify the category of each shader variable:

Vertex Shader

```
in  vec4 aPosition;
in  vec3 aNormal;
in  vec4 aColor;

out vec4 vColor;

uniform mat4 P;
uniform mat4 MV;
uniform mat3 N;

void main()
{
  const vec4 blue = vec4(0.0, 0.0, 1.0, 1.0);

  vec3 nHat = N * normalize(aNormal);
  vec3 eyeDir = vec3(0.5, 0.5, 1.0);

  vColor = aColor + max(dot(nHat, eyeDir), 0.0) * blue;
  gl_Position = P * MV * aPosition;
}
```

aPosition	[Choose]
vColor	[Choose]
P	[Choose]
aColor	[Choose]
aNormal	[Choose]
blue	[Choose]

gl_Position [Choose]

Question 3

3 pts

Consider a triangle with one of the primary colors (i.e., red, green, and blue) at each of its vertices. Will the color white appear inside of the triangle? Justify your answer.

Edit View Insert Format Tools Table 100%

12pt Paragraph

p 0 words

Question 4

8 pts

Consider the following pair of vertex and fragment shaders that are used to draw a triangle similar to the one described in the previous question. In the vertex shader, the value `gl_VertexID` is a value indicating which vertex is being processed (e.g., the

first vertex processed will have a `gl_VertexID` of zero, the second vertex a value of one, and so on).

Vertex Shader

```
in  vec4 aPosition;
out vec4 vColor;

void main()
{
  vec4 color = vec4(0.0, 0.0, 0.0, 1.0);

  // set the respective component based on the
  // vertex id to 1.0 in the color
  color[gl_VertexID] = 1.0;

  vColor = color;
  gl_Position = aPosition;
}
```

Fragment Shader

```
in  vec4 vColor;
out vec4 fColor;

void main()
{
  const vec4 white = vec4(1.0);

  const float almostZero = 0.001;
  bool anyValueAlmostZero =
    vColor.r < almostZero ||
    vColor.g < almostZero ||
    vColor.b < almostZero;

  fColor = anyValueAlmostZero ? white : vColor;
}
```

Describe what the rendered triangle looks like on the screen, and why? (Hint: think about how a color's represented, and interpolated)

Edit View Insert Format Tools Table 100%

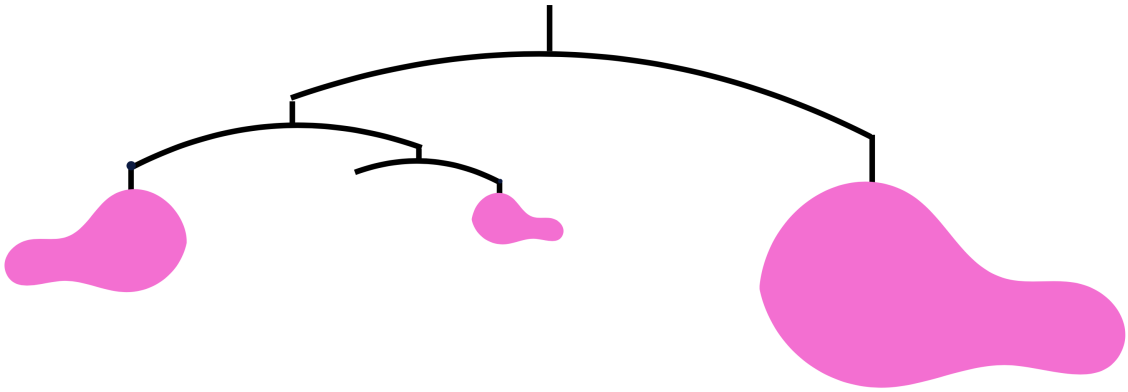
12pt Paragraph

p 0 words

Question 5

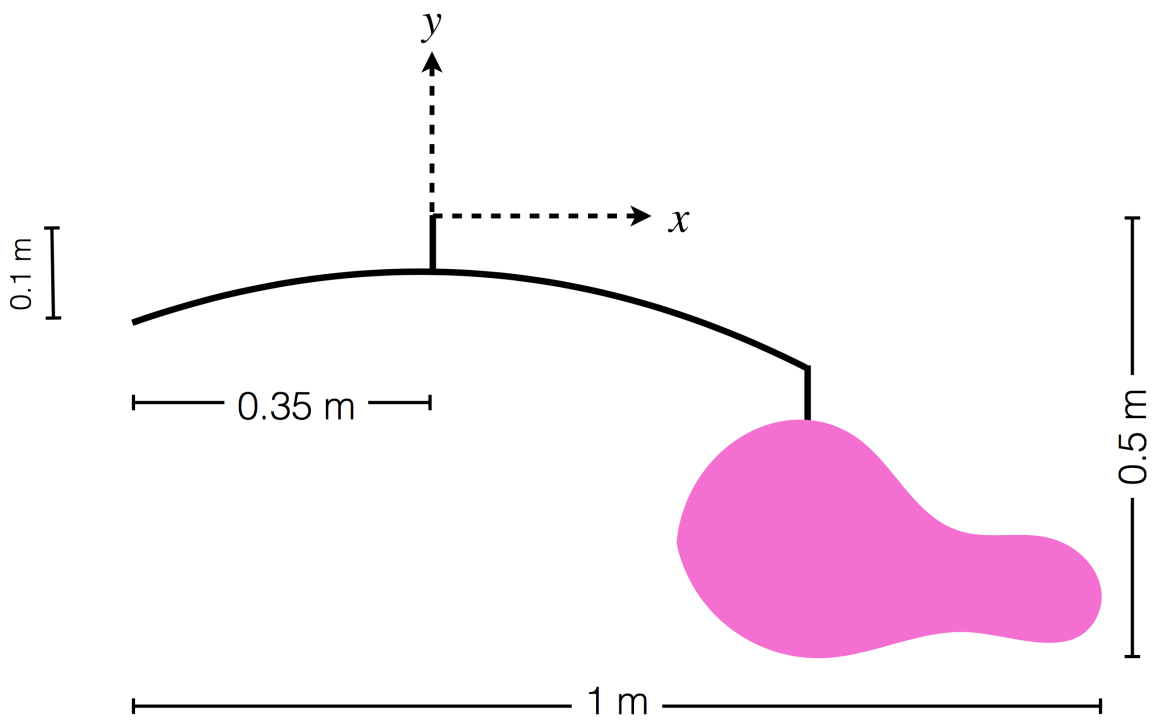
20 pts

As a member of the software team at Spinners, Inc., a producer of high-speed rotating (and potentially body harming) decorative **mobile sculptures (https://en.wikipedia.org/wiki/Mobile_(sculpture))** . The design department has provided you with their latest design, pictured below:



The mobile is composed of three pieces, and for their website, they want an animated version of their model. The CAD (computer-aided design) department provides you with the following geometric model which is drawn using a function named

`drawBlob()`, which renders one of the pieces sized as noted in the following technical specification diagram.



As a top-notch programmer (and someone who can name functions better than `drawBlob()` ... seriously), you decide to use your handy `MatrixStack` class to aid in managing the transformations to render the animated model. As you can see, the mobile is composed of three parts that are differently sized versions of the mobile component. The largest component of the mobile is drawn at its native size (i.e., as shown in the above diagram). The middle-sized component is uniformly 40% the size of the largest component, and the smallest is 20% the size of the largest component.

While the production version of the website is supposed to have an animated version of the mobile with the elements rotating about the mounting point (i.e., the point at the origin in the specification mode), for the prototype, it's sufficient to use fixed angles for the rotations.

Using our handy **MatrixStack.pdf** planning document, please detail the sequence of matrix stack commands (e.g., `load()`, `push()`, `translate()`, etc.). Include considerations for working with the viewing transform, indicate when the model should be rendered, and indicate the state of the matrix stack at each step (using our common notation of **R** for a rotation, **S** for a scaling operation, **T** for a translation, and **V** for the viewing transform) .

Any electronic method of submitting the completed file is acceptable: submitting images of a hand-written page, editing the PDF, etc. If you want to be 100% sure of submission, send an email including your file/images. This is also useful if Canvas is down, and provides a nice timestamp of when you've completed the problem.

Upload Choose a File

Question 6

6 pts

You're helping another developer who's having difficulty with the following draw command:

```
gl.drawArrays(gl.TRIANGLES, 0, 13);
```

How many triangles are rendered with this version of the call? _____

(please enter a numerical value)

The developer then updates the call to the following:

```
gl.drawArrays(gl.TRIANGLE_STRIP, 0, 13);
```

- How many triangles are rendered with the above call? _____ (please enter a numerical value)
- How many times is the vertex shader executed? _____ (please enter a numerical value)
- If some the of the triangles are culled, how many times is the fragment shader executed? _____ (free-form comment)

Question 7

2 pts

When the rasterizer is generating fragments for a geometric primitive, it often uses a pixel location's center to determine if the fragment shader should be executed for that location. For a fragment whose lower-left cornier is at (x, y), the center of the fragment is at $\left(x + \frac{1}{2}, y + \frac{1}{2}\right)$.

Likewise, the edge of a geometric primitive is often represented using our *point-normal form*: $f(\vec{p}) = \vec{p} \cdot \hat{n} + D$. When a positive result (i.e., the sign of $f(\vec{p})$) is found, the rasterizer "emits" the fragment, and the fragment shader is scheduled to run.

Consider the fragment located $(4, 3)$, and a primitive whose edge is specified by the vector $(1, -1)$, and with $D = 1$.

1. Enter the numeric value you used to justify your answer for 1. above.

   (Hint: remember what $\hat{n}$ represents)

2. Should the rasterize emit a fragment for the above situation?

   (enter yes or no)

Question 8

2 pts

3. Briefly justify your answer to Question 7, part 2. above.

Edit View Insert Format Tools Table 100%

12pt Paragraph

p 0 words </>

Not saved Submit Quiz