FINAL EXAM - December 2016

CSC 240 01: Computer Graphics

Instructor: Sara Mathieson

- This is a self-scheduled exam to be completed during one of the final exam periods.
- Please write all your work on these pages (front and back is okay, but do not use a blue book or any other pages).
- The exam is closed notes, closed Internet, and closed technology, but you may use two "cheat sheets".
- Your cheat sheets must be hand-written, created by you, 8.5" × 11", and can be double-sided (up to 4 sides total).
- Submit both cheat sheets with your exam.
- Do not discuss the exam with other students and respect the honor code of doing your own work.
- If you are unable to make progress on any part of the exam, tell me what you tried; describe your thought process.

Name	Solutions (sketches)

Part 1	/20	✓
Part 2	/10	✓
Part 3	/15	✓
Part 4	/15	✓
Part 5	/20	✓
Part 6	/20	✓
Total	/100	

Part 1: Short Answer

For the questions below, assume a viewport at $z = -1$, with $x_{\min} = -4$, $x_{\max} = 4$, $y_{\min} = -4$, and $y_{\max} = 4$. Consider two points in the scene: $p_w = (3, 2, -4)$ and $q_w = (3, 2, -2)$.

(a) Using a front *orthographic* projection, which point(s) are visible? Circle one answer.

- p_w only
- (circled) q_w only
- both p_w and q_w
- neither p_w nor q_w

(b) Using a front *perspective* project with a camera at the origin, which point(s) are visible? Circle one answer.

- p_w only
- q_w only
- (circled) both p_w and q_w
- neither p_w nor q_w

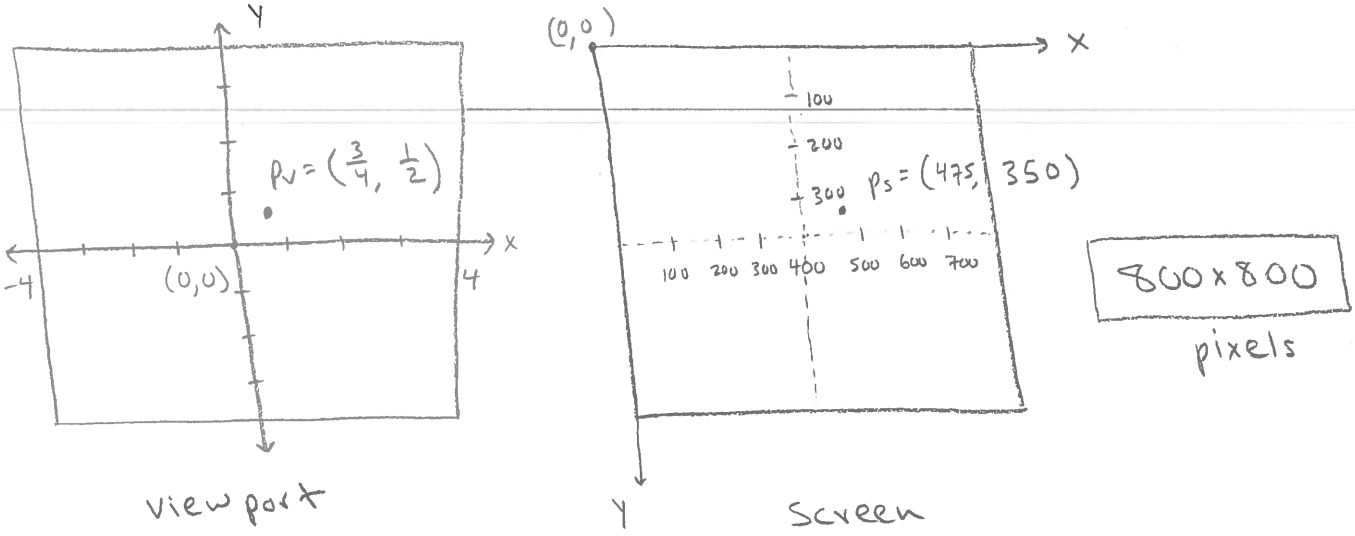
(c) (Vocab) For the point p_w, we can compute its position in different coordinate systems (for a perspective projection). For each set of values below, name the coordinate system.

$p_w = (3, 2, -4)$ ____world____ coordinates

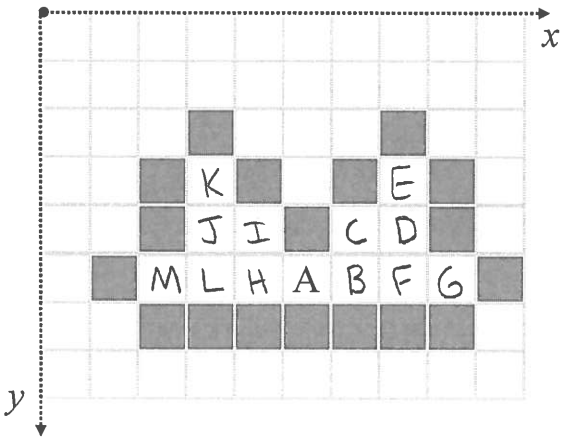
$p_v = (\frac{3}{4}, \frac{1}{2})$ ____viewport____ coordinates

$p_s = (475, 350)$ ____screen____ coordinates

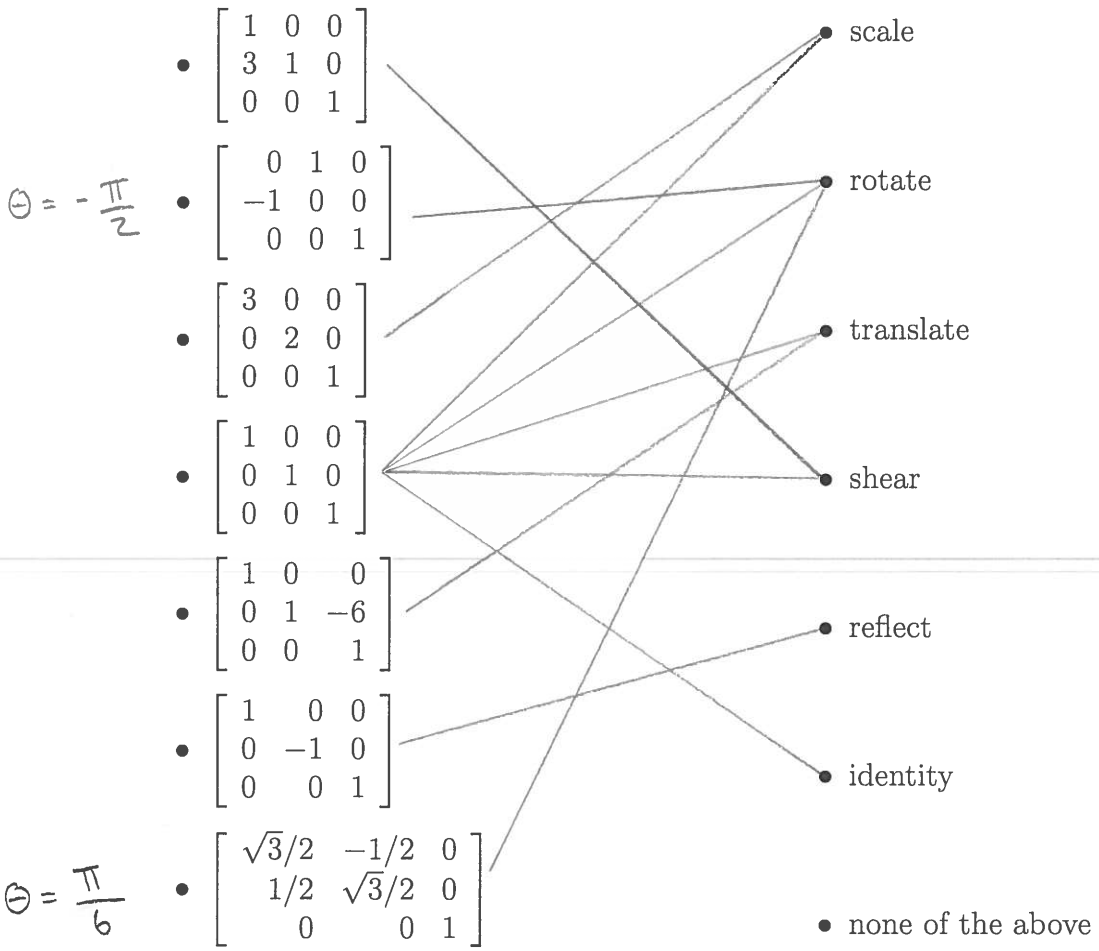
(d) Given the above values, what is the size of the screen in pixels? Draw both the viewport and the screen, labeling the origin and the point in each case.



(e) (Recursion) For our usual flood fill algorithm with direction order: north, east, south, west, show the order that pixels are filled in for the shape below. Assume the initial pixel colored is A, and continue labeling with B, C, etc.



(f) (2D transformation matrices) For each matrix on the left below, draw a line between it and its type(s) on the right.



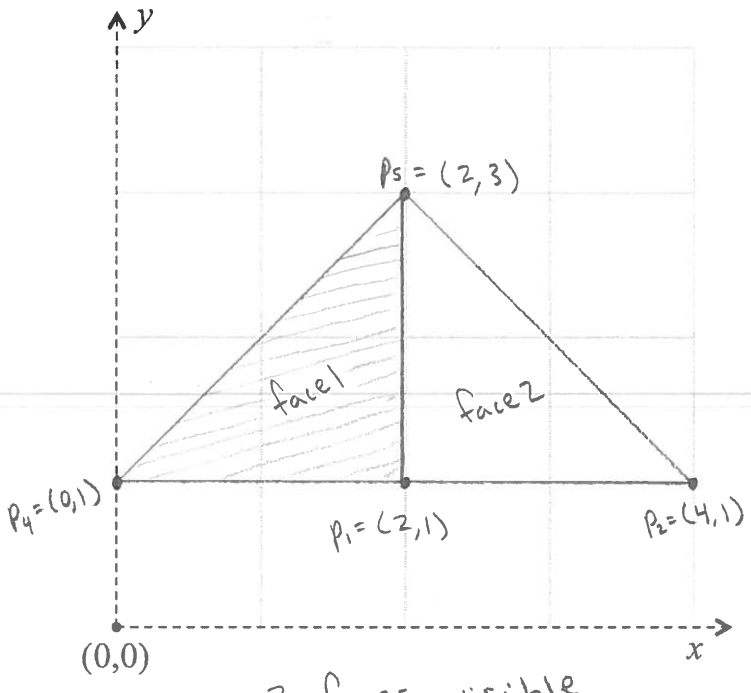
Part 2: Projection

You are given the following 5 vertices of a pyramid in world space, a camera at the origin, and a viewport at $z = -1$.

world coordinates	orthographic projection	perspective projection
p_1 = (2,1,-1)	(2,1)	(2,1)
p_2 = (4,1,-3)	(4,1)	$(\frac{4}{3}, \frac{1}{3})$
p_3 = (2,1,-5)	(2,1)	$(\frac{2}{5}, \frac{1}{5})$
p_4 = (0,1,-3)	(0,1)	$(0, \frac{1}{3})$
p_5 = (2,3,-3)	(2,3)	$(\frac{2}{3}, 1)$

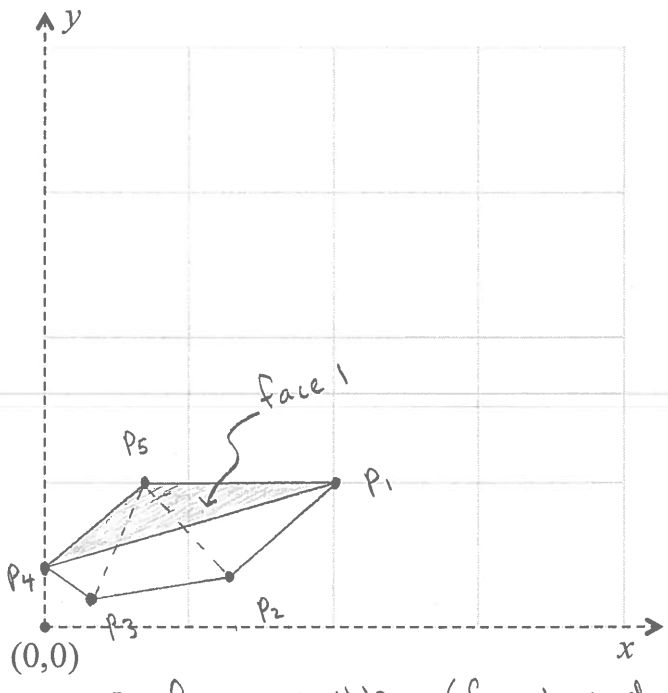
(a) Fill in the table above with the 2D viewport coordinates for each type of projection. Assume the viewport is the same as in Part 1, with $x_{\min} = -4$, $x_{\max} = 4$, $y_{\min} = -4$, and $y_{\max} = 4$.

(b) The coordinates below show one quadrant of the viewport. Draw what the "viewer" would see in each case (feel free to scale the axis, but label with the units you are using). Think about which faces are visible, and use dotted lines to show edges which are not visible.



2 faces visible

Orthographic



2 faces visible (face 1 and bottom face)

Perspective

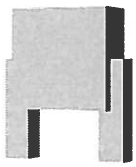
very different!

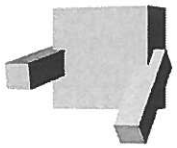
Part 3: Hierarchical Modeling

In this problem we will create a robot with a body and two arms. Part of the code is shown below, and your task is to complete the hierarchical modeling part. There are two movement goals:

1) The entire robot should be able to move around as one piece using keyboard input.
2) Each arm of the robot should be able to rotate around the x-axis, with the "shoulder" as the origin of rotation.

Two views of the robot are shown below, one with no arm rotation, and the other with the arms at different angles (both rotated around the x-axis). The view point is slightly to the side to show the effects, but assume a front view for the code below.





```
// set up cube geometry and material
var geometry = new THREE.BoxGeometry(1,1,1);
var material = new THREE.MeshLambertMaterial( { color: "yellow" } );

// set up mesh objects
var bodyCube = new THREE.Mesh(geometry, material);
bodyCube.scale.set(2,2,1);
var leftArm = new THREE.Mesh(geometry, material);
leftArm.scale.set(0.5,2,0.5);
var rightArm = new THREE.Mesh(geometry, material);
rightArm.scale.set(0.5,2,0.5);

// set up containers
body = new THREE.Object3D();
rightArmContainer = new THREE.Object3D();
leftArmContainer = new THREE.Object3D();

// TODO complete
```

```
body.add( bodyCube )
body.add( rightArmContainer )
body.add( leftArmContainer )
```

} add all three pieces to the body container

```
rightArmContainer.add( rightArm )
leftArmContainer.add( leftArm )
```

} add arm mesh objects to each arm container

```
// END complete
scene.add(body);
```

(continued on next page)

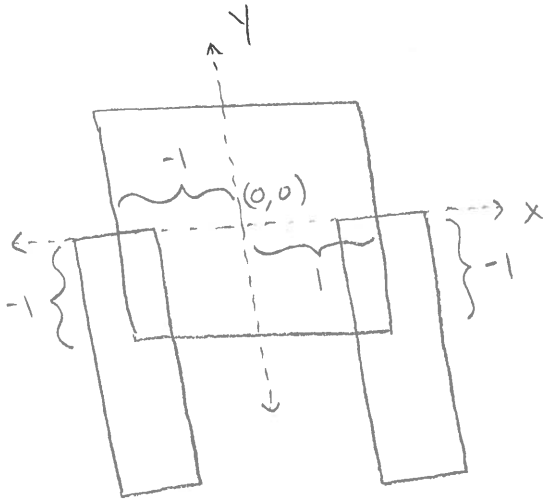
(a) Complete the code on the previous page by adding the appropriate objects to the appropriate containers, and moving any objects so that the rotations will be correct (use this page if you need more space). Keep in mind that the center of the `bodyCube` is at the origin. Syntax does not need to be perfect.

(continued)

leftArm.position.set(-1, -1, 0)

rightArm.position.set(1, -1, 0)

note: these are the meshes, not the containers.



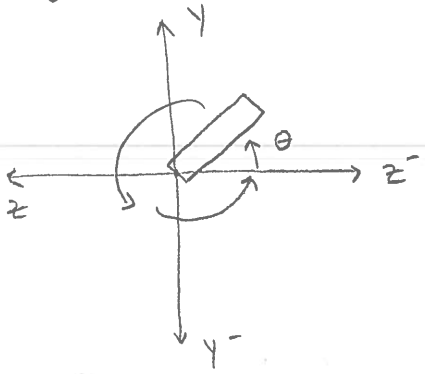
move each arm over 1 (left & right) and down 1

(b) If in the rendering loop I have the code:

```
leftArmContainer.rotation.x += 0.03;
rightArmContainer.rotation.x += 0.06;
```

and I look at the setup from the *right view*, are the arms rotating clockwise or counterclockwise? Explain your answer.

right view



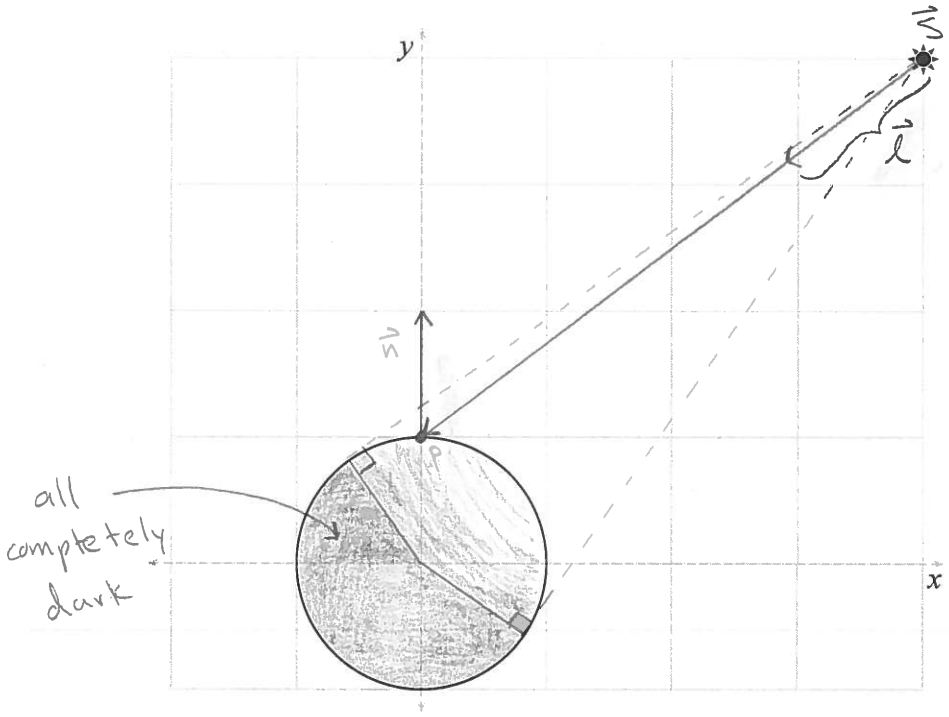
looking down x+

Since the angle is positive and we are looking down the positive x-axis, the arms are rotating

counter clockwise

Part 4: Lighting

Consider the setup below, with a unit sphere at the origin and a light at the point $\vec{s} = (4, 4, 0)$.



(a) We will try to find the lighting at the point on the sphere $\vec{p} = (0, 1, 0)$. Label this point on the diagram above. Draw and label the unit normal vector at this point. What is the unit normal vector $\vec{n}$ at this point?

$\vec{n} = (0, 1, 0)$

(b) Now we will find $\vec{l}$, the unit light vector from the light source to the point $\vec{p}$. First find the direction of the light vector, using the position of the light source and the position of the point. Compute the magnitude. Finally, use the magnitude to compute the unit vector $\vec{l}$.

direction of light vector:

$$\vec{p} - \vec{s} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} - \begin{bmatrix} 4 \\ 4 \\ 0 \end{bmatrix} = \begin{bmatrix} -4 \\ -3 \\ 0 \end{bmatrix}$$

magnitude m

$$= \sqrt{(-4)^2 + (-3)^2 + 0^2}$$

$$= \sqrt{16 + 9}$$

$$= 5$$

$$\Rightarrow \vec{l} = \left(-\frac{4}{5}, -\frac{3}{5}, 0\right)$$

(c) Compute the dot product of the normal vector with the light vector (i.e. $\vec{n} \cdot \vec{l}$). Assuming a white light and that our sphere will be shaded in grayscale, what does this tell us about the color at $\vec{p}$?

$$\vec{n} \cdot \vec{l} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \cdot \begin{bmatrix} -4/5 \\ -3/5 \\ 0 \end{bmatrix} = \boxed{-\frac{3}{5}}$$

the color at $\vec{p}$ will be fairly bright, not the brightest as the dot product is not -1, but still in the light since $\vec{n} \cdot \vec{l} < 0$.

light-medium gray

(d) In the diagram on the previous page, shade the sphere to depict the affects of this light source. It does not have to be exact, but show the lightest part by leaving it unshaded, and clearly show which parts of the sphere are not affected by the light at all.

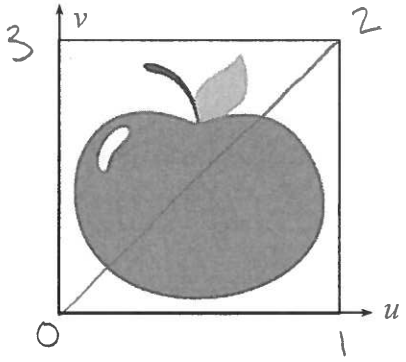
(see previous page)

Part 5: Texture Mapping

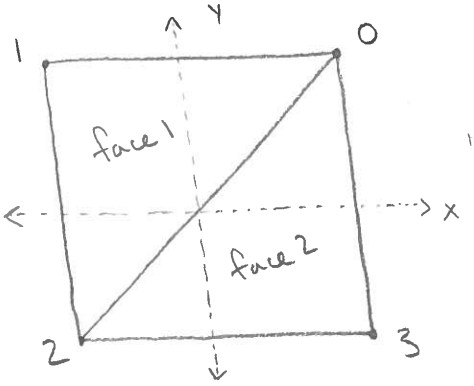
The following section of code is designed to map the apple texture onto a square, so that the square looks like the original texture shown below. For this problem, assume the camera, lighting, texture import, etc have been done correctly so that the square is visible.

```
var squareGeom = new THREE.Geometry();

squareGeom.vertices = [new THREE.Vector3( 1, 1, 0 ),     0
                       new THREE.Vector3( -1, 1, 0 ),    1
                       new THREE.Vector3( -1, -1, 0 ),   2
                       new THREE.Vector3( 1, -1, 0 )];   3

squareGeom.faces = [new THREE.Face3( 2, 0, 1),
                    new THREE.Face3( 2, 0, 3)];

squareGeom.computeFaceNormals();

var uv = [new THREE.Vector2(0, 0),   0
          new THREE.Vector2(1, 0),   1
          new THREE.Vector2(1, 1),   2
          new THREE.Vector2(0, 1)];  3

squareGeom.faceVertexUvs[0].push([uv[2], uv[0], uvcoords[1]]);
squareGeom.faceVertexUvs[0].push([uv[2], uv[3], uvcoords[0]]);

square = new THREE.Mesh( squareGeom, appleMaterial );
```

typo!



(a) Draw a picture of the *geometry* of this square, labeling each vertex with its vertex number as defined in the code. Show the two faces that make up this square.



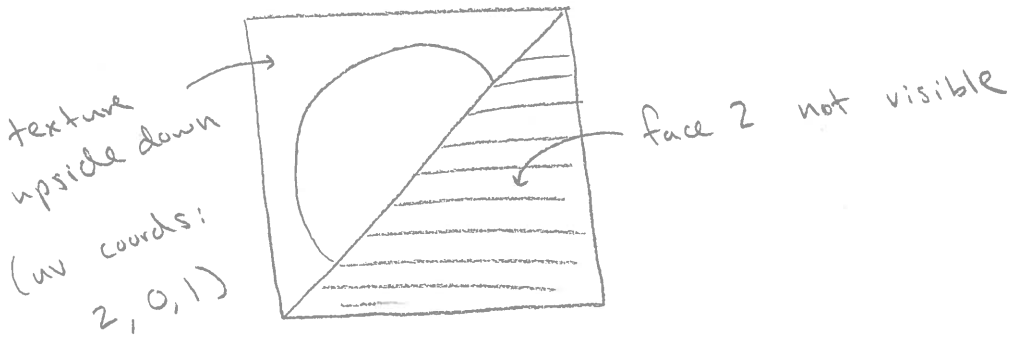
face 1: (2, 0, 1)

face 2: (2, 0, 3)

→ not counterclockwise!

face2 will not be visible since normal points away.

(b) Label the UV points on the apple texture above with their indices. Based on the code above, what does the viewer see? Draw a picture showing the current texture mapping.



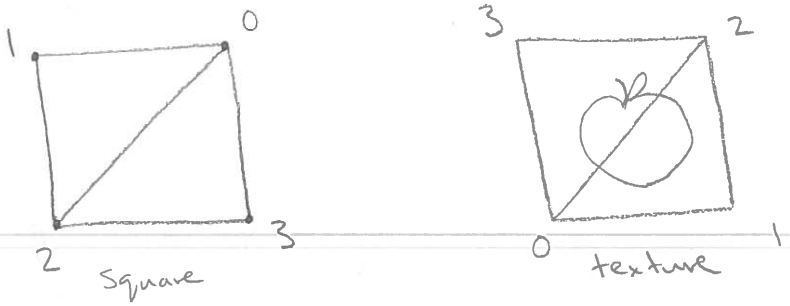
(c) Identify, *explain*, and correct the *two* mistakes in this code, so that the texture mapping looks like the apple above. You can make the corrections directly on the code above if you like.

① face 2 vertex order incorrect, should be:

(2, 3, 0) or (3, 0, 2) or (0, 2, 3)

↑ assume this way for the next mistake

② texture is mapped upside down, should be:



face1: (uv[0], uv[2], uv[3])

face2: (uv[0], uv[1], uv[2])

Part 6: Ray tracing

For this setup, we will have a camera at the origin, defined by the line:

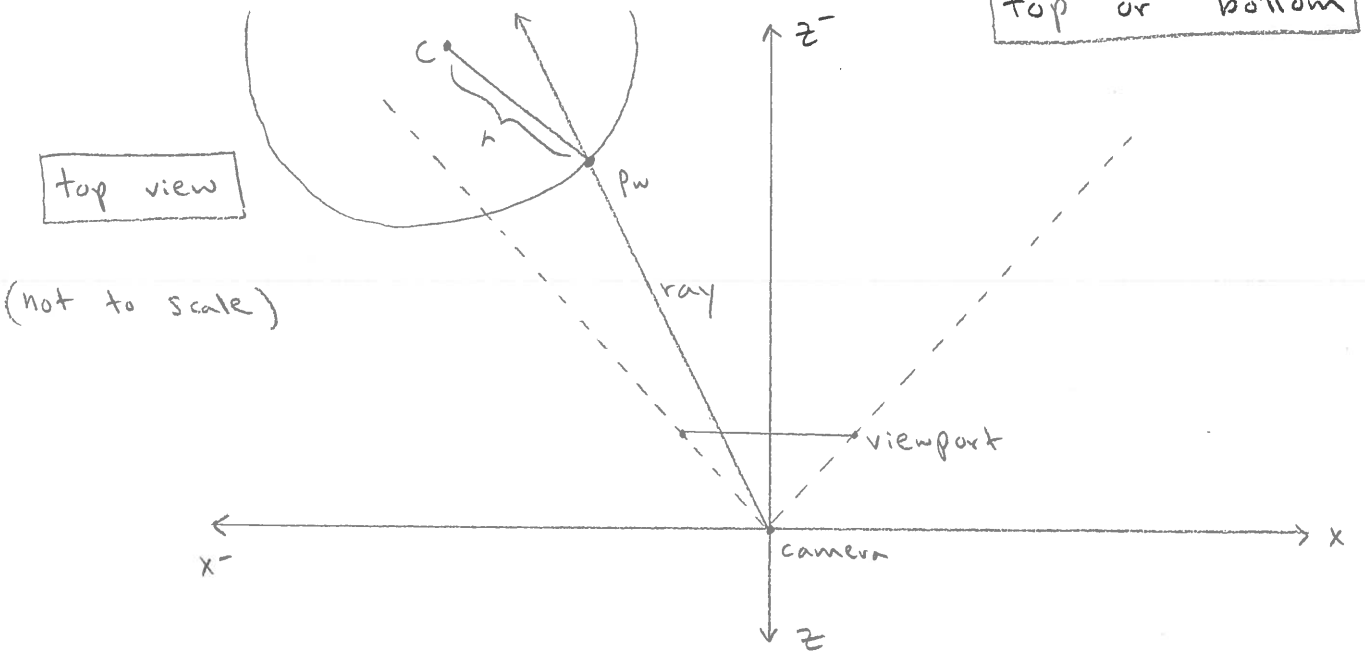
```
var camera = new THREE.PerspectiveCamera( 90, 1, 1, 40 );
```

(a) Consider a ray with unit direction

$$\vec{R}_d = \left(-\frac{5}{13}, 0, -\frac{12}{13}\right).$$

What viewpoint(s) will helpful for visualizing this setup? Draw one such viewpoint, labeling the axes, viewport, camera, and ray in this direction.

top or bottom



(b) This ray intersects a sphere 26 units from the camera. Using the ray equation, determine this intersection point p_w.

$$R(t) = \vec{R}_o + t \cdot \vec{R}_d$$

$$p_w = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + 26 \begin{bmatrix} -5/13 \\ 0 \\ -12/13 \end{bmatrix}$$

$$p_w = \begin{bmatrix} -10 \\ 0 \\ -24 \end{bmatrix}$$

(c) If the center of the sphere is at the point $c = (-18, 0, -30)$, what is the radius of the sphere?

$$r = \| p_w - c \|$$

$$r = \left\| \begin{bmatrix} -10 \\ 0 \\ -24 \end{bmatrix} - \begin{bmatrix} -18 \\ 0 \\ -30 \end{bmatrix} \right\|$$

$$r = \left\| \begin{bmatrix} 8 \\ 0 \\ 6 \end{bmatrix} \right\| = \sqrt{8^2 + 0^2 + 6^2} = \sqrt{64 + 36} = \sqrt{100}$$

$$\Rightarrow \boxed{r = 10}$$

More space (tell me which problem this is for), or draw me a picture.

Thank you for your enthusiasm and hard work in graphics, I really appreciate it. Enjoy the break!