University of California San Diego
Department of Computer Science
CSE167: Introduction to Computer Graphics
Fall Quarter 2016
Midterm Examination #2
Thursday, November 10th, 2016

Name:________________________________

Your answers must include all steps of your derivations, or points will be deducted.

This is closed book exam. You may not use electronic devices, notes, textbooks or other written materials.

Good luck!

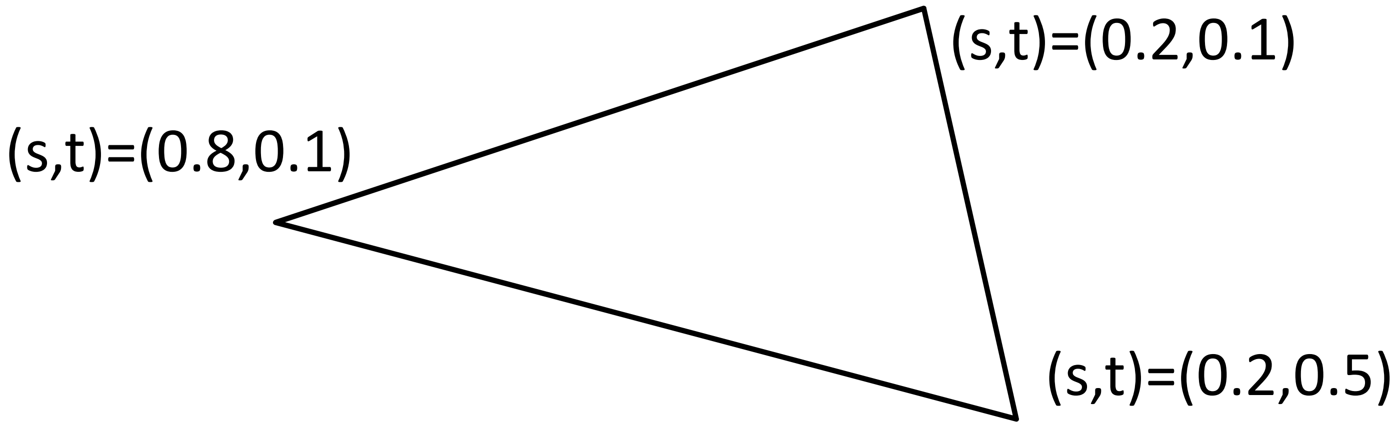
Do not write below this line

Exercise	Max.	Points
1	10	
2	10	
3	10	
4	10	
5	10	
6	10	
7	10	
8	10	
Total	80	

1) Texture Mapping (10 Points)

Part 1: Texture Coordinates

Given a triangle with associated texture coordinates (s,t) for its vertices

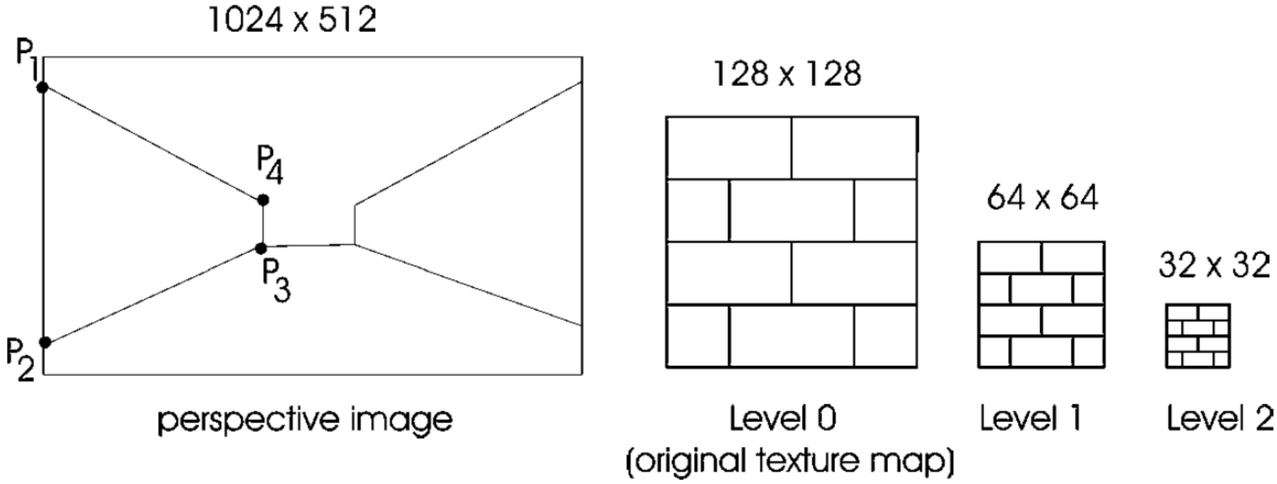


and a simple texture with its corner coordinates given:

a) Sketch **into the triangle above** what it will look like after texture mapping. (4 points)

Part 2: Mipmapping

Suppose we have a brick wall that forms the left-hand wall of a corridor in a maze game, as shown in the image below, and it is defined (in world coordinates) by points P_1, P_2, P_3, P_4. Assume that the brick wall is to be 16 bricks high and 200 bricks long.



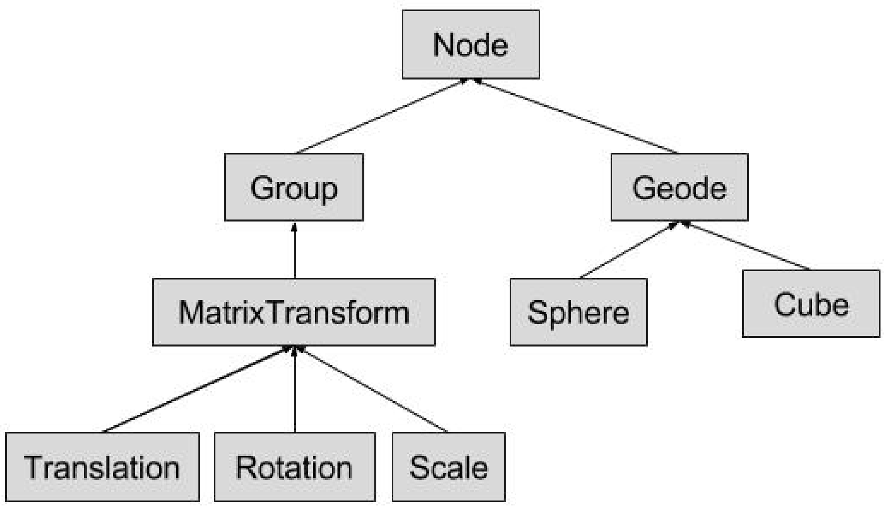
b) Using the height of the brick wall as seen in the image, estimate (with derivation) how many texels of the original texture map (no mipmapping) a screen pixel represents, both at near points on the wall, i.e., on the edge P_1P_2, and at distant points on the wall, i.e., on the edge P_3P_4. (3 points)

c) In the perspective image above, sketch approximately what regions of the wall on the left (defined by points P1, P2, P3 and P4) will use each of the mipmap textures on the right (use nearest-mipmap interpolation, not trilinear mipmapping). (3 points)

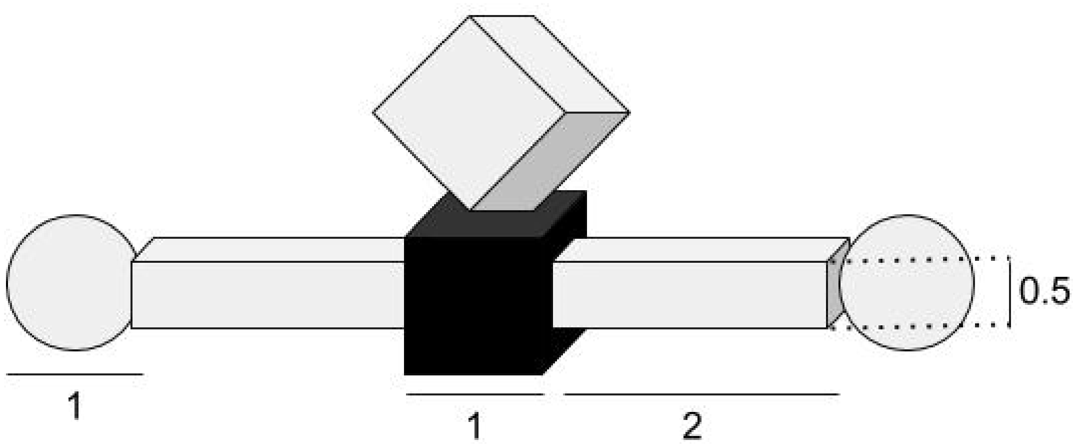
2) Scene Graph (10 points)

Below we are given a scene graph structure and the model of a legless robot. In the scene graph structure, the class Translation creates a translation matrix, Rotation creates a rotation matrix and Scale creates a scaling matrix. The class “Cube” generates a 1x1x1 cube centered at the origin, and the class “Sphere” generates a sphere with a radius of 0.5. The black cube in the robot model is a 1x1x1 cube centered at the origin, the other cube is also a 1x1x1 cube but it is rotated by 45 degrees about its z axis. The coordinate system is Cartesian with positive x to the right, positive y up and positive z towards the viewer.

The scene graph class hierarchy is:



The robot looks like this:



a) Write pseudo code that will generate the scene graph representing the robot. Minimize the number of nodes in your scene graph (hint: it applies both for Geode and Group subclasses). (7 points)

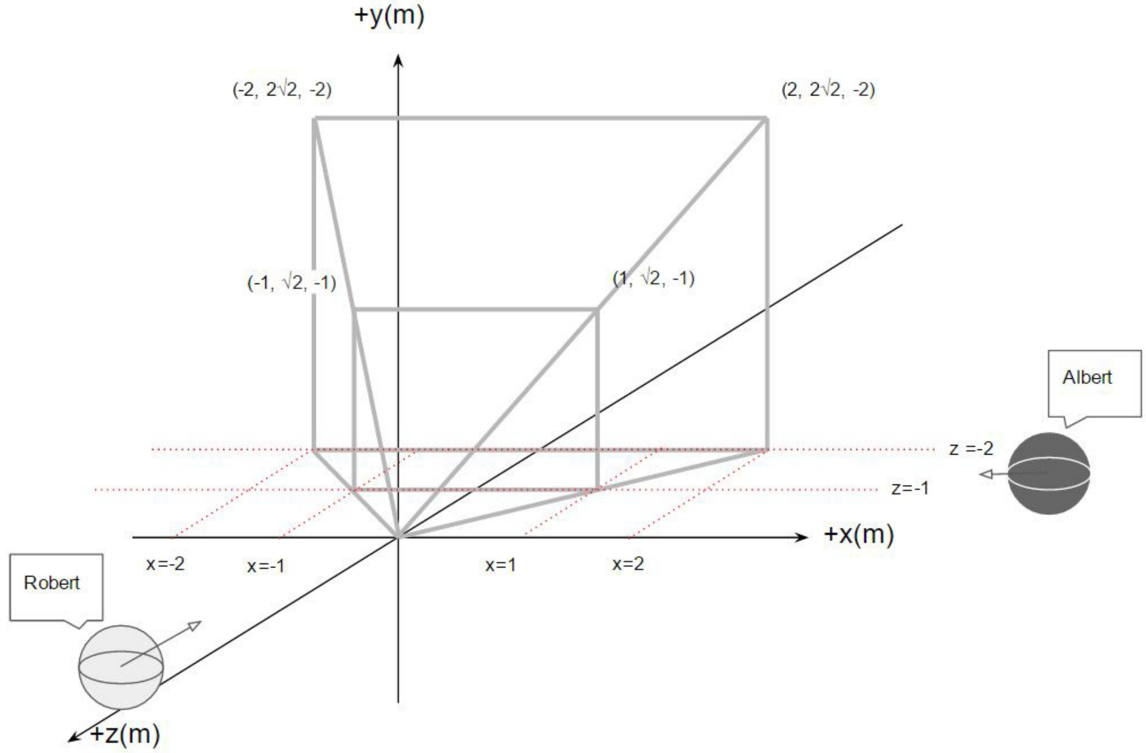
```
robot = new Group()
```

b) Now we want to create a small army of 10 legless robots organized in a 5 x 5 square. We would like to animate the entire army by moving it in the positive x direction. Write the scene graph code (assuming that the previous code has been completed) that will create the army and animate it. (Hint: the robot is originally facing the positive z direction). (3 points)

```
army = new Group()
```

3) View Frustum (10 Points)

Given a camera at the origin and a view frustum made up of left, right, top, bottom, near, and far planes, as in the picture below:



A robot named Robert is shaped like a sphere of radius 1m. Robert becomes invisible when he is outside the view frustum and the shortest distance from its center to the nearest plane view frustum plane becomes more than its radius. So at the exact moment of intersection, Robert is visible.

a) At t = 0s, Robert is at (0m, 1m, 3m). Robert moves at the rate of (0, 0, -1m) per second. So at t = 1 s, Robert is at (0m, 1m, 2m). When will we see Robert for the first time and when will we see Robert for the last time? (5 points)

b) Robert has a robot friend named Albert. Albert is also shaped like a sphere of radius 1m and is visible under the same conditions as Robert. At $t = 0s$, Albert is at $(9m, 1m, \sqrt{2}m)$. Albert moves at the rate of $(-2m, 0, 0)$ per second. So at $t = 1s$, Albert is at $(7m, 1m, \sqrt{2}m)$. Will Robert and Albert be able to see each other at some point? If so, how long can they see each other? (5 points)

4) Performance Optimization (10 Points)

Briefly describe each of the following rendering performance optimization strategies.

a) 2D impostors for 3D objects (2 points)

b) Adaptive mesh resolution (2 points)

c) Small object culling (2 points)

d) Backface culling (2 points)

e) Degenerate culling (2 points)

5) Parametric Curves (10 Points)

Consider the two points $\mathbf{P_0}$ and $\mathbf{P_1}$ as below. There exists another point **Q** that lies on the straight line connecting $\mathbf{P_0}$ and $\mathbf{P_1}$.

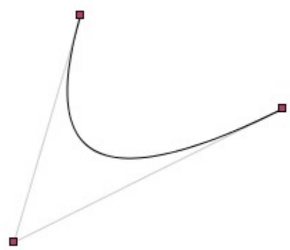
a) Write an equation to represent **Q** in terms of $\mathbf{P_0}$, $\mathbf{P_1}$ and a parameter **t**. (4 points)

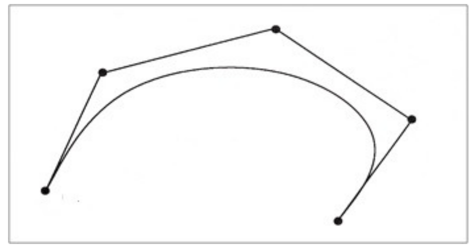
b) What are the two values of **t** at which **Q** would equal $\mathbf{P_0}$ and $\mathbf{P_1}$? (2 points)

c) From the equation above, calculate the tangent at **Q**. Does the tangent vary at every **Q**? (4 points)

6) Bezier Curves (10 Points)

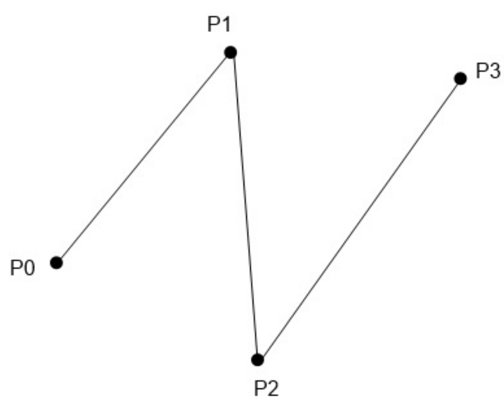
a) Give the order (degree) of each of the following curves: (1 point)

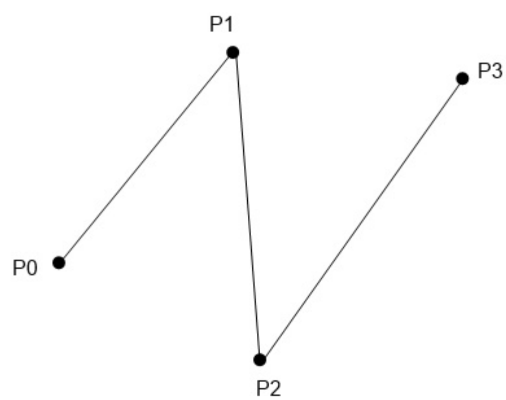


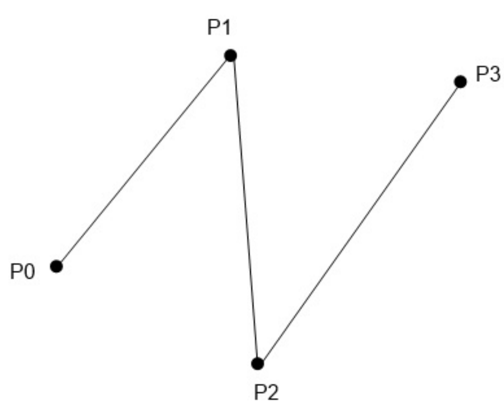


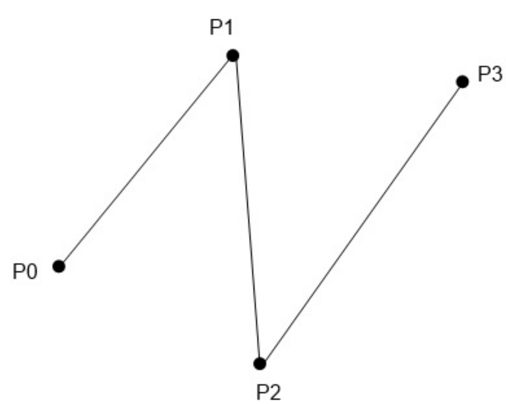
b) If the number of control points that define a curve is *n,* what is the order of the curve in terms of n? (1 point)

c) Given to you is a Bezier Curve's 4 control points P_0, P_1, P_2 and P_3. (Note the order in which they are given!). Use the De Casteljau algorithm to find a point S (approximately) on the curve for values of: **t = 0.2, t = 0.5 and t = 0.75**
You need to use Linear Interpolation to find the points Q_0, Q_1, Q_2, R_0, R_1 and S. **Draw** one diagram for each of the above values of t.
In the fourth diagram, by looking at the points you have obtained, approximate the shape of the curve. (8 points)









7) Surface Patches (10 Points)

Evaluating along a line requires an interpolation between two points. This concept can be extended to two dimensions creating a surface patch. Given the values for p_0, p_1, p_2, and p_3 below, find point $x(\frac{1}{2}, \frac{1}{4})$ following the steps below. x is defined as x(u, v).

$$p_0 = < -2,\ -4,\ 8 >$$
$$p_1 = < -6,\ 28,\ -24 >$$
$$p_2 = < 40,\ -14,\ -4 >$$
$$p_3 = < 32,\ -18,\ -12 >$$

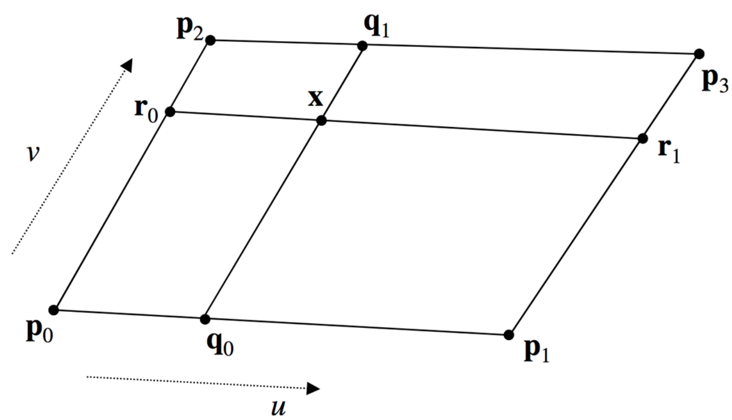


Diagram illustrates the concept of a surface patch, but is not to scale.

a) Find points q_0 and q_1 (6 points):

$$q_0 = < \quad , \quad , \quad >$$

$$q_1 = < \quad , \quad , \quad >$$

b) Find point x (4 points):

$$x = < \quad , \quad , \quad >$$

8) Environment Mapping (10 Points)

a) Which problem does Environment Mapping attempt to solve? (2 points)

b) Which two types of geometries for environment maps did we discuss in class? Name one advantage for each of them that it has over the other. (4 points)

c) With environment mapping, is it easier to render metallic or diffuse objects? Explain why. (2 points)

d) Name two particularly computationally expensive operations which are part of the environment mapping algorithm, which can be done efficiently in a shader. (2 points)