Exam One

CS 184: Foundations of Computer Graphics
Spring 2026
Prof. James O'Brien

Student Name:_____ **Student ID:**_____

Instructions — ***Read them carefully!***

Please read these instructions carefully before beginning the exam.

No clarifications will be given for the exam. Do not discuss the exam until solutions are released.

Allowed Resources

This is an open-book, open-notes exam. You may consult textbooks, notes, and other reference materials.

You must write all answers yourself. No other person may help you with the exam. You should not copy answers from the internet.

Use of AI Tools

You may use AI tools (such as an LLM ChatBot) ***only*** *for the following limited purposes:*

• Locating reference materials

• Asking general reference questions about concepts

• Asking the AI whether your answer appears correct and complete

You may ***not*** *use AI to:*

• Write any part of an answer

• Rewrite, edit, or improve your answer

• Suggest how your answer should be corrected

If you choose to use AI, you do so entirely at your own risk. An AI stating that an answer is correct does not guarantee that it is actually correct!

Time Limit

You should spend no more than 4 hours completing this exam.

Please fill in the you start time: _____ and end time: _____

Students with official extra-time accommodations may use their approved time. If you do so, note next to the end time above that you used your time accommodation. Do ***not*** *include any medical details or other private information.*

Writing Your Answers

Please:

- *Write legibly (if you handwrite or draw parts of the exam)*
- *Be clear and concise*
- *Show your reasoning where appropriate.*
- *If you make any assumptions to answer a problem, note what that they are on the test.*

Submitting the Exam

You may complete the exam in either of the following ways:

1. *Fill out the PDF digitally, or*
2. *Print the exam, write your answers, and scan the completed document*

If you wish to use a drawing tool or software to create diagrams, that is permitted.

Honor Code and Academic Integrity Certification

The Honor Code is the commitment of students individually and as a community to uphold honesty and integrity in all academic work.

By signing below, I certify that:

- *All work submitted in my name for this exam is entirely my own work.*
- *I have not received help from another person while completing this exam.*
- *I have used only the resources permitted in the exam instructions.*
- *Any use of AI tools followed the restrictions described in the instructions.*
- *I will inform course staff if I become aware of any Honor Code violations.*

I understand that making a false statement on this declaration is a violation of the Honor Code and may result in disciplinary action under the University Code of Conduct.

Your Signature: _____

Date: _____

Student ID: _____

1. Please fill in each of the blanks with an appropriate answer. If multiple possible answers exist, write the best one. Note any assumptions next to the question, separate from your answer. *2 points each blank*

1. True or False: A Z-buffer stores the scene depth at each pixel as an 8-bit integer. False
2. True of False: The BRDF describes how much light coming in from one incoming direction goes out in another outgoing direction. True
3. True of False: Smeagol's law describes how reflective materials behave. False
4. True of False: The exponent in the Phong shading model controls how bright a material appears. False
5. When representing Points / locations in 3D using homogenized coordinates, the fourth coordinate (i.e. "w") will be non-zero.
6. A directional light can be thought of as a point light source located "at infinity".
7. True or False: Flat shading is named after the French computer graphics researcher Jon-Paul Flat. False
8. True or False: Translating and then scaling is the same as scaling and then translating. False
9. Radiance is measured in units of watts per square meter per steradian.
10. True or False: Rotation is a linear transformation. True
11. A Bezier curve is always enclosed by the convex hull of its control points.
12. The rows and columns of an arbitrary rotation matrix are always orthonormal (orthognal also ok).
13. True of False: Matrix multiplication is associative. True
14. True of False: Homogenized coordinates are needed to allow rotation to be expressed as matrix multiplication. False

15. Of the various methods discussed in class for representing rotations, the method of ___Euler angles___ is least appropriate for interpolation due to singularities and poor behavior which include gimbal lock.

16. In ray tracing, sending multiple rays through a given pixel and averaging the results is called ___supersampling___, and it is used most often for ___antialiasing___.

Also ok to answer antialiasing is used to remove jaggies, or something similar.

17. The implicit formula for a sphere is ___(x-c)^2 - r^2 = 0___.

18. True or False: BSP Trees can be used to accelerate ray intersection tests for complex scenes. ___True___

19. True of False: Under orthographic projection straight lines will always appears as straight lines in an image. ___True___

20. True of False: Under orthographic projection spheres will appears as circles in an image. ___True___

21. A ___Shadow map___ stores depth/distance values and can be used for shadow computation when rendering a scene

22. The ___SVD (polar decomp also ok)___ breaks a matrix A into A = U S Transpose(V) where U and V are orthonormal and S is diagonal.

23. True or False: Rotation about an arbitrary axis in 3D requires three separate transformation matrices. ___False___

24. The cross product of the tangent vectors of a parametric surface generally can be used to computed the surface ___normal / perpendicular___ vector.

25. True or False: Systems like OpenGL typically convert everything to simple quadrilaterals before rendering. ___False___

26. Finding the intersection of a ray with a sphere requires solving a ___quadratic___ equation.

27. Finding the intersection of a ray with a plane requires solving a ___system of linear equations___.

Also ok to say something like 3 cross-product tests

28. True of False: Radiance remains constant along parabolic arcs in free space. False

29. True of False: The implicit representation of a given geometric shape is unique. False

30. True of False: Any set six of non-intersecting polygons can be sorted in front-to-back order. False

31. True of False: Shining an ultraviolet light on scorpions induces a chemical response that causes them to glow green and become agitated. False

32. True of False: C^1 continuity does not always imply G^1 continuity. True

33. True of False: G^1 continuity does not always imply C^1 continuity. True

34. True of False: Bump-mapping will not change an object's silhouette. True

35. True of False: Lambert's law describes how specular materials behave. False

36. True of False: Any curve represented using the cubic Hermite basis could also be represented using regular cubic polynomials. True

37. True of False: Computer graphics has applications in the area of medicine. True

38. True of False: Modern flat-panel displays use arrays of micro-CRTs. False

39. True of False: The barycentric coordinates of a point with respect to a given triangle can be used to determine if the projection of the point perpendicular to the triangle's plane is inside or outside the triangle. True

40. True of False: Rodrigues' Rotation Formula uses complex numbers to represent rotations in 3D space. False

41. True of False: The light received energy from a surface decreases linearly with distance. False

2. **If you have two *unit* vectors, A and B, then we can write the dot and cross products as A . B and A x B, respectively. Simplify the following expressions:** *9 points*

(B x A) x (A x B) = [0, 0, 0] (just 0 is also ok)

A x (B x (A x B) = -(A . B) * (A x B)

B x (A x (B x (B x (A x B)))) = - ||B||^2 * (A x B)

3. **You have a sphere centered at [3,3,3] with radius 4, and a ray from [10,10,10] in the direction [-1,-1,-1]. Write the implicit equation for the sphere, the parametric equation for the ray, and compute the coordinates of the intersection points.** *Be neat and clear!* *15 points*

Sphere equation: (x - [3,3,3]) ^2 - 16 = 0 (ok to use 4^2)

Ray equation: x = [10,10,10] + t [-1, -1, -1] (ok, to normalize the direction)

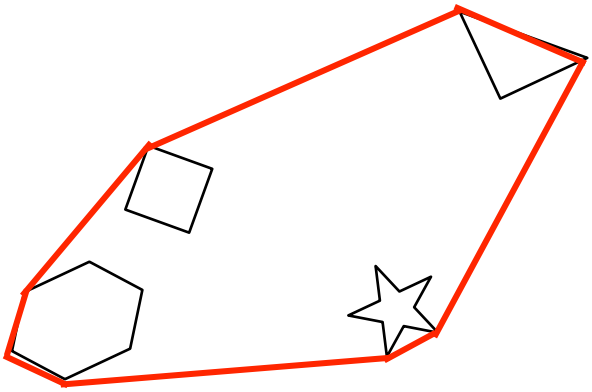
Intersections at =
(5.3094, 5.3094, 5.3094)
(0.6906, 0.6906, 0.6906)

also ok:
(3 + 4/sqrt(3), 3 + 4/sqrt(3), 3 + 4/sqrt(3))
(3 - 4/sqrt(3), 3 - 4/sqrt(3), 3 - 4/sqrt(3))

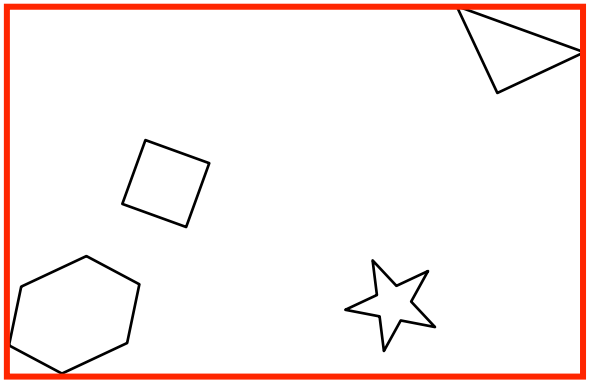
4. **Write out a 3x3 transformation matrix that will rotate the vectors [√1/2, √1/2, 0], [0, 0, 1], and [√1/2, -√1/2, 0] to align with the X, Y, Z axes respectively.** *10 points*

```
s  s  0
0  0  1
s -s  0
```

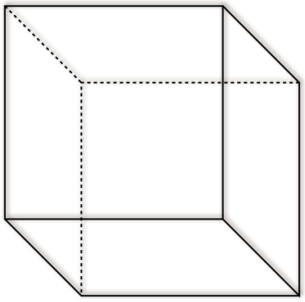
5. Draw the single *convex hull* that encloses all four shapes shown: *16 points*



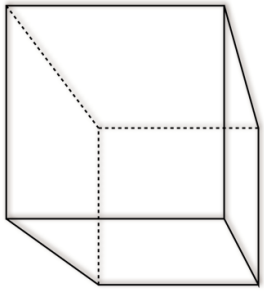
6. Draw the single *axis-aligned bounding box* that encloses all four shapes shown: *12 points*



7. **One of the diagrams below shows a cube under orthographic projection, the other under perspective projection. Label which is which.** *4 points*



ortho



perspective

8. **Write down plausible RGB values for the following materials:** *18 points*

Note: there are multiple correct answers.

Glossy Metallic Magenta
Kd = [255 , 0 , 255]
Ks = [255 , 0 , 255]

Glossy Plastic Yellow
Kd = [255 , 255 , 0]
Ks = [255 , 255 , 255]

Dull/Flat Cyan
Kd = [0 , 255 , 255]
Ks = [0 , 0 , 0]

9. **If shading a point at the origin with normal [0, 1, 0] and Kd=[0.2, 0.4, 0.6], where the light is located at [0, 4, 3] with intensity 20 (white), and the eye located at [27, 91, 17], compute the RGB value of the <u>diffuse</u> lighting term.** *15 points*

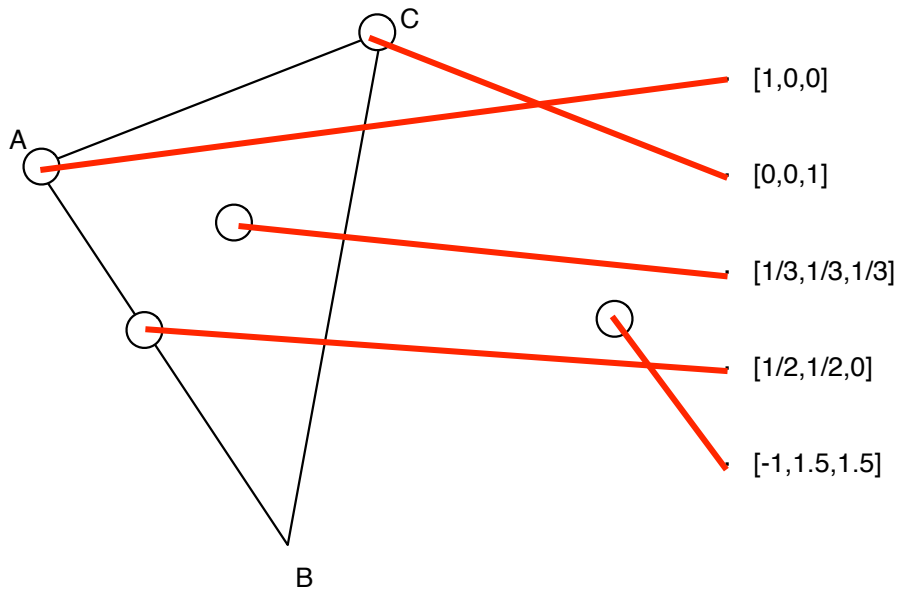
Note: It is ok / optional to include a 1/r^2 falloff

[3.2 , 6.4 , 9.6]

10. Circle the 3D homogenized matrix that would scale objects by 2x. *4 points*

a) $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$ **b)** $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix}$ **c)** $\begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix}$ **d)** $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1/2 \end{bmatrix}$

11. This diagram shows a triangle with vertices labeled a, b, and c. Five locations have been indicated with circles. The list of numbers to the right contains triples of numbers representing the barycentric coordinates of these circles. Draw a line connecting each triple with the correct circle. *20 points*



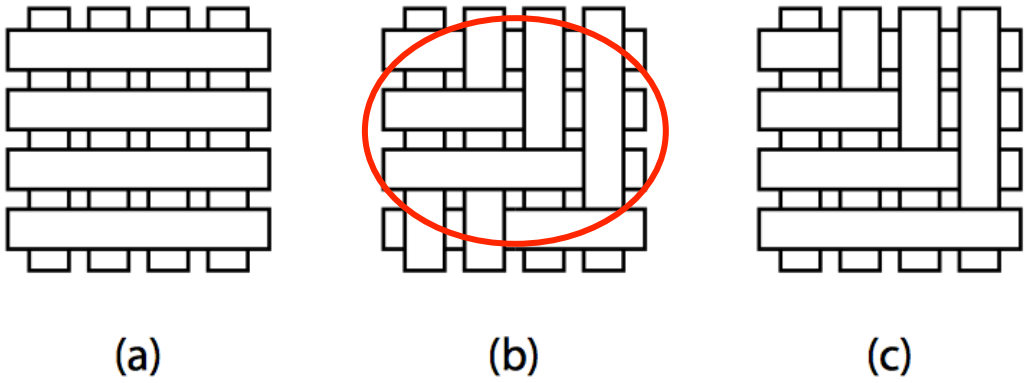
12. Write out the 3x3 matrixes for both a rotation about the Z axis, and for a scale along the Z axis. *4 points*

$$rotz(q) = \begin{bmatrix} \cos(q) & -\sin(q) & 0 \\ \sin(q) & \cos(q) & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad scalez(s) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & s \end{bmatrix}$$

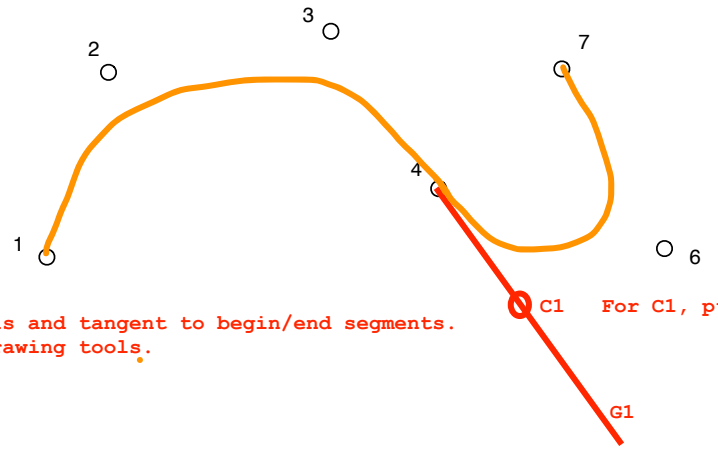
Now write out a 3x3 matrixes for a rotation about the Z axis and for scale along the Z axis. that would result if we did rotations *clockwise* instead of *counter-clockwise*. *8 points*

$$rotz(q) = \begin{bmatrix} \cos(q) & \sin(q) & 0 \\ -\sin(q) & \cos(q) & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad scalez(s) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & s \end{bmatrix}$$

13. Which of the following scenes would cause problems for the Painter's Algorithm? *4 points*



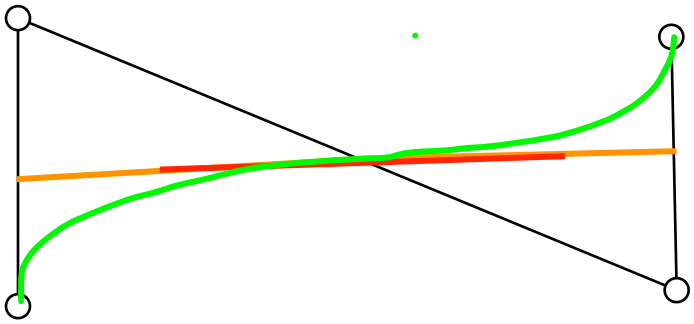
14. The diagram below shows control points for a curve made by joining two cubic Bezier segments. However control point #5 has been removed. Indicate a location where #5 may be placed to achieve C^1 continuity and draw the curve that would result. Also draw a line where #5 may be placed to achieve G^1 continuity. *Make sure your diagram is clear and geometrically reasonable.* *25 points*



Curve should be in convex hulls and tangent to begin/end segments.
Make allowances for limited drawing tools.

For C1, pt5 - pt4 = pt4 - pt3

15. The diagram below is the control polygon for a Bezier curve segment. Draw the curve and show how de Casteljau's algorithm can be used to subdivide the curve into two equal halves. Make sure your drawing is geometrically reasonable and shows correct curve tangents for the the beginning, middle, and end of each segment. 20 points



16. Given three points, A, B and C, write out a parametric equation in two variables, *u and v,* for the triangle formed by these three points. *12 points*

x(u,v) = C + u * (A - C) + v * (B - C)

17. Would the differences between Phong interpolation (interpolating normals) and Gouraud interpolation (interpolating colors) be more noticeable on diffuse or specular surface? Explain why. *20 points*

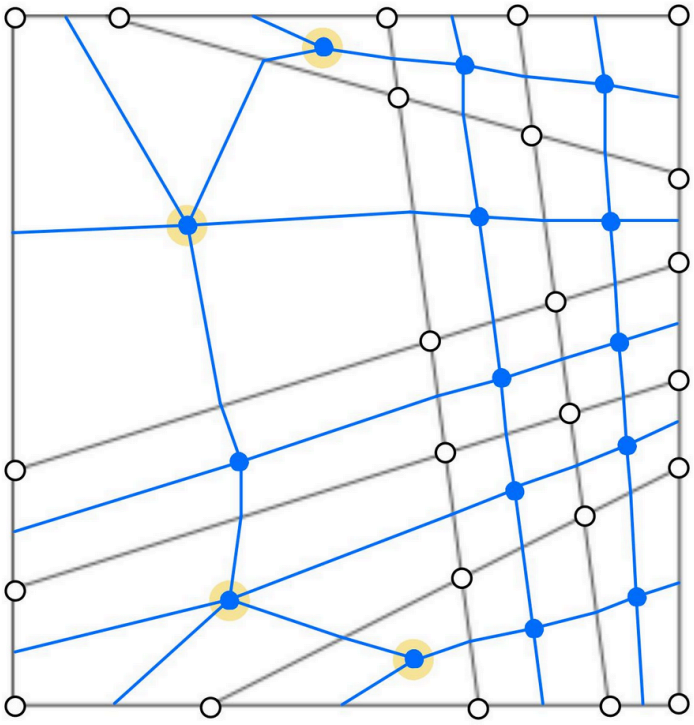
On a specular surface because the specular highlight depends on the normal in a very non-linear way.

18. Given a rotation matrix, how would you determine the axis that it rotates around? *9 points*

The eigenvector with eigenvalue = +1

Also, log of matrix is ok.

19. Here is a piece of mesh. Draw the result of applying one iteration of Catmull-Clark subdivision. Then circle all vertices that are extraordinary (both original and the new ones you added). ***Note that only the topology of your answer is relevant.*** *25 points*



20. Below is example code that returns the <u>first</u> intersection of a ray with any object in a BVH.

```
# Here is example code that returns the FIRST intersection of a ray
# with any object in a BVH.

# node.bv.intersect(ray) returns true if any part of the ray with
# t in [ray.t_min, ray.t_max] overlaps the bounding volume.
#
# node.obj.intersect(ray) returns the FIRST intersection of the ray
# with the object whose t value is in [ray.t_min, ray.t_max],
# or null if there is no such intersection. Nothing other than the
# first in-range t value is returned.

function intersectBVH(node, ray):

    if node == null:
        return null

    if not node.bv.intersect(ray):
        return null

    if node.isLeaf:
        return node.obj.intersect(ray)

    hitLeft  = intersectBVH(node.left, ray)
    hitRight = intersectBVH(node.right, ray)

    if hitLeft == null:
        return hitRight

    if hitRight == null:
        return hitLeft

    if hitLeft.t < hitRight.t:
        return hitLeft
    else:
        return hitRight
```

Write pseudocode for a function that instead returns the <u>last</u> intersection of the ray with any object in the BVH within the range [ray.t_min, ray.t_max].

You may use only:
- **node.bv.intersect(ray)**
- **node.obj.intersect(ray)**
- **the BVH node fields shown in the example code**
- **recursion and local variables**

Assume intersections return a structure containing only the parameter t, and return null if no intersection exists. *20 points*

```
The simple answer is to just reverse the ray.

More complex answers are fine, as long as they would produce the correct result.
```