Closed Book and Closed Notes.

You will have the entire exam period (until 9:45am) to complete the exam. If you need more time, we can see if someone else needs the room.

Please write your name on every page!

Write numerical answers in fractional form or use radicals (square root symbols) – we would prefer to see $\frac{\sqrt{3}}{2}$ than .866. You should not need a calculator for this exam.

Unless otherwise noted, assume that everything is a right-handed coordinate system and that angles are measured counter clockwise (i.e. to find the direction of rotation, point your thumb along the axis and curl your fingers).

Please keep your answers concise and readable. Answers that are excessively wordy or illegible will be considered incorrect. If you need more space, use the back of the page, but put a note telling us to look there.

Note: there are some questions at the back of the exam for which there are lots of possible right answers.

There are 10 Questions on this exam (there are 2 questions on page 9)

There are 100 points on the exam:

<table>
<thead>
<tr>
<th>Question</th>
<th>__ / 15</th>
</tr>
</thead>
<tbody>
<tr>
<td>Question</td>
<td>__ / 10</td>
</tr>
<tr>
<td>Question</td>
<td>__ / 15</td>
</tr>
<tr>
<td>Question</td>
<td>__ / 10</td>
</tr>
<tr>
<td>Question</td>
<td>__ / 10</td>
</tr>
<tr>
<td>Question</td>
<td>__ / 8</td>
</tr>
<tr>
<td>Question</td>
<td>__ / 10</td>
</tr>
<tr>
<td>Question</td>
<td>__ / 3</td>
</tr>
<tr>
<td>Question</td>
<td>__ / 12</td>
</tr>
<tr>
<td>Question</td>
<td>__ / 7</td>
</tr>
<tr>
<td>Total:</td>
<td>_____ / 100</td>
</tr>
</tbody>
</table>
Question 1: (15pts total)

To try out the subdivision methods we discussed in class, I wrote a system that takes as input a polygon mesh – it assumes that all of the polygons are flat and convex.

For subdivision schemes that require triangles, all polygons are divided into triangles by adding a point in the center of the of the polygon, and making triangles by connecting adjacent edges – so a pentagon gets divided into 5 triangles, a hexagon into 6 triangles, etc.

The system implements the Catmull-Clark, Loop, and Modified Butterfly subdivision schemes as described in class. It works correctly.

A) For a given input mesh, two of the schemes are likely to give similar looking results (that is that they will be visually similar when you subdivide so much that the polygons are so small that you can’t see them – or, said differently, the limit surfaces will be very similar), while the third is likely to be quite different. Which one will be most different, and why? (4pts)

Modified Butterfly because it is interpolating (and only C1), while the other two schemes are approximating and C2.

B) If the input is a regular triangle mesh with N triangles, how many polygons will there be after 2 rounds of subdivision? (give an answer for each scheme) (6pts)

Loop = N * 4 * 4 = 16 N (each triangle divides into 4 other triangles)
Butterfly = 16N
Catmull-Clark = 3 * 4 * N = 12 N (each triangle becomes 4 quads. Each quad becomes 4 more quads)

C) For all three schemes, there is a special subdivision rule for the boundaries of the mesh. Points on the boundary only depend on other points on the boundary. What is the advantage of such a scheme? (5pts)

It allows two surfaces to be stitched together seamlessly w/C(0) continuity just by sharing their border points

(OK to not say C(0))
Question 2: 10 pts (1,2,2,5)

In this question, we consider resampling an image to make it bigger. We will consider 3 different methods: nearest neighbor interpolation (basically using a box filter), linear interpolation (that is, using a tent filter), and Lanczos interpolation (using a Lanczos filter). The Lanczos filter is a much better approximation to the ideal low-pass filter (sinc) than the linear filter.

I start out with a checkerboard that is light gray and dark gray (the pixel values are 205 and 50). The image is 32x32 pixels, and the checkerboard squares are 4x4.

I resize my checkerboard by a factor of 4 (so the result is 128x128) using each of the three resampling methods (nearest, linear, Lanczos).

<table>
<thead>
<tr>
<th>Original</th>
<th>Sharp</th>
<th>Blurry</th>
<th>Ripples</th>
</tr>
</thead>
</table>

Which method is likely to have given the sharp result? **NN**

Which method is likely to have given the blurry result? **Linear**

Which method is likely to have given the result with the light and dark fringes? **Lanczos**

For this specific case (scaling the checkerboard by 3), the sharp result looks best (since it is a checkerboard with the same two values as the original). More often, we prefer the methods that produced the blury result to the method that produced the sharp one. Explain why and give an example where blurrier is better.

The sharp edges are actually aliasing – we don’t know where the sharp edges are (or that they really are that sharp). For the checkerboard, the aliasing just happens to be consistent and give a nice result.

If there were smooth edges in the original, or if the pixel edges didn’t map exactly to other pixel edges, we would have had jaggies. If things moved by sub-pixel amounts we would have had crawlies. Either of those look worse than the small amount of blurring (usually).

*(Should say "aliasing" and give some example of why its ugly)*
Question 3: 15 points (5 each)

Please define the following terms concisely – a sentence or two is all that is needed.

Limit Surface –

The surface obtained by applying a subdivision rule infinitely many times.

Shadow Map –

An image taken by a camera at the light source used to determine what the light source can “see” in order to create shadows.

Color Gamut – (in class we just said “Gamut”)

The range of perceived colors that can be produced by a set of primary colors.
Question 4: 10 points (3,3,4)

In the picture below, the curve is made of two connected cubic Bezier segments (one using points 1234 and the other using 4567)

In all parts of this question, we consider the continuity at point 4.

For each of the continuity types below, either say “the curve has it” or describe the minimum number of points that must move to create that continuity, and where those points must go. To describe where a point should go, make a labeled point (with a letter) on the picture above. You’ll need to mark the letters on the picture above yourself.

Example: C(0) continuity: the curve has it

Example: XXX continuity: point 7 moves to X and point 6 moves to Y (where X was drawn)

A) C(1) continuity – move point 5 to A (1 pt for knowing points must move, 1 pt for moving just 5, 1 point for A in right place - 1 square left of 5)

B) G(1) continuity – curve already has it

Note: if you got G and C continuity confused, but knew to move point 5 to A, you get 1 point and 1 point for knowing to move just 1 point.

C) C(2) continuity – move point 5 to A, point 6 to B (1 pt for curve doesn’t have it, 1 pt for 5, 1 point for knowing 6 moves, 1 point for B in some other place)

Note: I realized that we didn’t say how to determine how to match 2nd derivatives, only that 3 points need to match. This is why you don’t need to know where B goes.
Question 5: 10 pts (5,5)

Most humans (i.e. those who are not color blind) are tri-chromats. We have 3 different types of photoreceptors (cones).

Like many kinds of birds, ducks are penta-chromats. They have 5 different types of photoreceptors.

Describe a situation where the duck’s superior color vision might allow it to distinguish colors that we could not.

The duck would be able to distinguish between colors formed by combinations of colors that might be metamers to a creature with fewer cone types. For example, it may be able to distinguish yellow from red+green (which a person could not).

Because we don’t know what the responses that a ducks’ cones have, we can’t really say which colors would be distinct. So any example is fine.

The superior color vision is probably handy in distinguishing different plants (that each reflect different amounts of different colors) – but the key thing to say is that the duck is good at telling combinations of wavelengths apart.

Note: (and this applies to part B too) some people pointed out that the duck’s cones might extend outside the range of “visible light” (for example, they might see ultra-violet). We gave partial credit, because it is an orthogonal issue to the number of cone types. And, technically, ultra-violet and infra-red are not “colors.”

If a duck looks at a rainbow (a real rainbow – created by water droplets acting as a prism spreading the white sunlight into its component colors), does it see more colors than us? Explain your answer.

Hint: a rainbow is made up of a series of pure colors (that is, colors of a single wavelength).

Each spectral color is already distinct to a human since they all create different ratios of LMS response, so we’d see a rainbow just as well as a duck.

(However, the duck would not be tricked into seeing a rainbow from a display of 3 primaries – but that’s part A)
Environment mapping makes some assumptions that allow for the creation of shiny surfaces using standard graphics hardware.

A: Explain what the assumptions are and describe a situation in which Environment mapping works well.

Environment mapping assume that the object is small relative to the distance to the things it reflects. Therefore, things with the same normal “see” the same piece of the environment in the reflection.

A small shiny sphere (or other curved object) in a big room or outdoor environment works well.

Some people said that environment maps only work for static scenes. This is half true – you need to recreate the environment map whenever things in the scene moves, so they are less convenient – but its not a restriction. We gave partial credit for this answer (unless you combined it with some other thing).

B: Describe a situation where the assumptions of Environment mapping are violated, and it would be a poor approximation of a shiny object.

A big flat object, or having things to reflect that are close to the reflecting object, violate these assumptions and look wrong. You don’t want to make a big flat mirror in an indoor scene.
Question 7: (10 points)

When doing MIP-MAPPING, tri-linear interpolation is generally better than bi-linear interpolation. Give some reasons why (describe some things that would look worse with bi-linear).

The biggest problem is that bi-linear would have discrete jumps in the filtering. So, if the texture is applied so that the scaling varies (a wall going off into the distance such that some parts are big and some parts appear small), there will be a visible discontinuity. Or, if the object moves, its textures might all of a sudden get sharper/blurrier.

Another problem is that bi-linear has fewer choices, and therefore might either make the texture blurrer than it needs to be - or not blurry enough (and have aliasing). Trilinear can get closer to having just enough blurring to avoid aliasing.

Either answer is OK.