CS559 Computer Graphics – Spring 2016

Final Exam – Tuesday May 10th 2016

Time: 2 hrs

Name	
University ID	

Part #1	
Part #2	
Part #3	
Part #4	
Part #5	
TOTAL	

1. [$6 \times 4\% = 24\%$] MULTIPLE CHOICE SECTION. Write the letter of the correct answer (or answers) in the space provided. You do not need to give a justification for your answer(s).

   (1) Which of the following curves are *arc-length parameterized*?

   Write the letter of ALL correct answers here: $\boxed{}$

   (a) The curve $\mathcal{C}(t) = (1, t, t^2)$, $t \in [0, 2]$
   (b) The curve $\mathcal{C}(t) = (\cos t, 1, \sin t)$, $t \in [0, 2\pi]$
   (c) The curve (actually, a straight line) $\mathcal{C}(t) = (2^t, 2^t, 2^t)$, $t \in [0, 1]$

   (2) Sometimes when using texture mapping we will have a situation where the size of a display pixel (fragment) where a textured object is drawn will be much **smaller** than the size of the pixels of the texture image that has been mapped to that same object location.
   Which of the following statements is correct about this scenario?

   Write the letter the ONE most correct answer: $\boxed{}$

   (a) If the texture image contained black and white stripes that alternated at a certain frequency, the rendered image could make those stripes appear as if they were alternating at a lower frequency, due to aliasing.
   (b) This is exactly the scenario for which mip-mapping can be most successful in eliminating aliasing artifacts.
   (c) In this scenario, if texture look-ups select only the closest pixel in the texture image, we risk jagged edges appearing on our display. Interpolating between nearby texture pixels (usually via bilinear interpolation) is the typical remedy.

   (3) Sometimes when using texture mapping we will have a situation where the size of a display pixel (fragment) where a textured object is drawn will be much **larger** than the size of the pixels of the texture image that has been mapped to that same object location.
   Which of the following statements is correct about this scenario?

   Write the letter the ONE most correct answer: $\boxed{}$

   (a) This scenario occurs when we have zoomed our camera too close to a textured object.
   (b) This is exactly the scenario for which mip-mapping can be most successful in eliminating aliasing artifacts.
   (c) This is exactly the scenario for which bi-linear interpolation between nearby texture image pixels will be the best way to avoid artifacts.

(4) Can we tell the difference between a curve that is arc-length parameterized and one that is not?

Write the letter of ALL correct answers here:

(a) Yes, a curve that is not arc-length parameterized can be identified by the presence of sharp corners.

(b) If all we are shown is the entire trajectory at once (as opposed to any motion along the curve), plotted as a continuous line, we cannot tell if it is arc-length parameterized or not. This is because any curve can be re-parameterized according to arc-length, without changing its shape.

(c) We can indirectly observe the presence or absence of arc-length parameterization if we are shown the motion of an object along the curve, as the parameter is advanced at a constant rate. An arc-length parameterized curve will result in a motion that always exhibits constant velocity along the curve.

(5) Which of the following statements are true about recursive (backward) ray tracing?

Write the letter of ALL correct answers here:

(a) It cannot handle transparent objects, like glass.

(b) One of its disadvantages is that it makes a relatively crude simplification that all but one ray bounces are fully specular.

(c) It can be implemented in the fragment shader.

(6) Which of the following statements are true about bump mapping?

Write the letter of ALL correct answers here:

(a) It can create surface roughness in the interior of individual triangles by displacing points interior to each triangle outside of the triangle plane.

(b) Its effect on a flat plane goes away if we tilt the bump-mapped plane to the point where it becomes parallel with the viewing direction.

(c) It requires using a texture mapping unit, regardless of whether we use texture mapping for color purposes.

2. [36%] SHORT ANSWER SECTION. Answer the following questions in no more than 1-3 sentences (when required) or by filling in the answer boxes.

   (a) [6%] What are **two** apperance effects that would be very difficult (or impossible) to obtain with *direct illumination* methods, and would require a *global illumination* technique to capture properly? (You may use the WebGL feature set, to the extent described in class, as a representative of the "direct illumination" class of methods).

   (b) [6%] Polynomial curves, where each component $x(u), y(u), z(u)$ of the curve formula $\mathcal{C}(u) = (x(u), y(u), z(u))$ is a polynomial expression, are the most important and popular class of parametric curves. Explain one reason why using polynomial expressions can be very advantageous, when designing parametric curves.
   Hint: Alternatives to polynomials would be expressions that include trigonometric, exponential or logarithmic functions, square roots, ratios of expressions, etc. What disadvantages would those have?

   (c) [6%] When we were discussing diffuse and specular lighting, we saw that performing the lighting calculations in the *vertex shader* (instead of the fragment shader) would probably be suboptimal, but maybe acceptable in some cases. Explain why, in the case of texture mapping, it would make almost no sense to attempt to do texture look-ups in the vertex shader, and why such texture calculations should be done in the fragment shader instead.

(d) [18%] There are certain appearance effects that rely on the graphics card's ability to support multiple textures (i.e. having multiple texture mapping units), while other effects are only made possible due to the render-to-texture ability of the modern graphics pipeline. Finally, some effects would be too complex to reproduce using the standard pipeline and shaders, and would be best suited to global illumination techniques. For the scenarios below, mark one of the following options in the boxes provided, to indicate which of these features will be necessary to reproduce them.

MULTI-TEXTURE: Give this answer if the ability to have multiple textures is crucial in order to support this effect. Another way to think about this – select this option if this effect would have been very difficult to achieve using only a single texture mapping unit.

RENDER-TO-TEXTURE: Give this answer if the ability to support rendering-to-texture is crucial in order to achieve this effect.

MR: *Both* multi-texture *and* render-to-texture are needed.

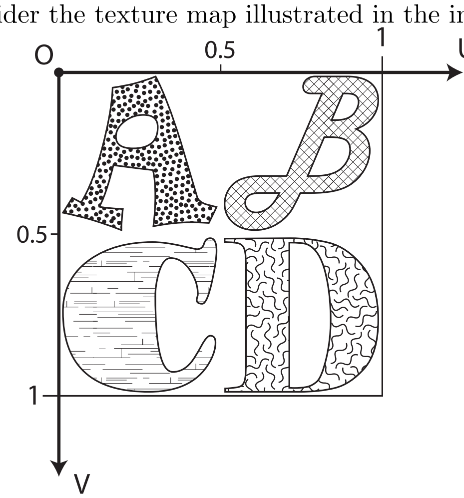
GLOBAL ILLUMINATION: This effect is not well suited to what the standard graphics pipeline (and WebGL) can do, and should instead be replicated using a global illumination technique.

NONE: This effect can be achieved with the standard graphics pipeline, even without using multi-texture or render-to-texture capabilities.

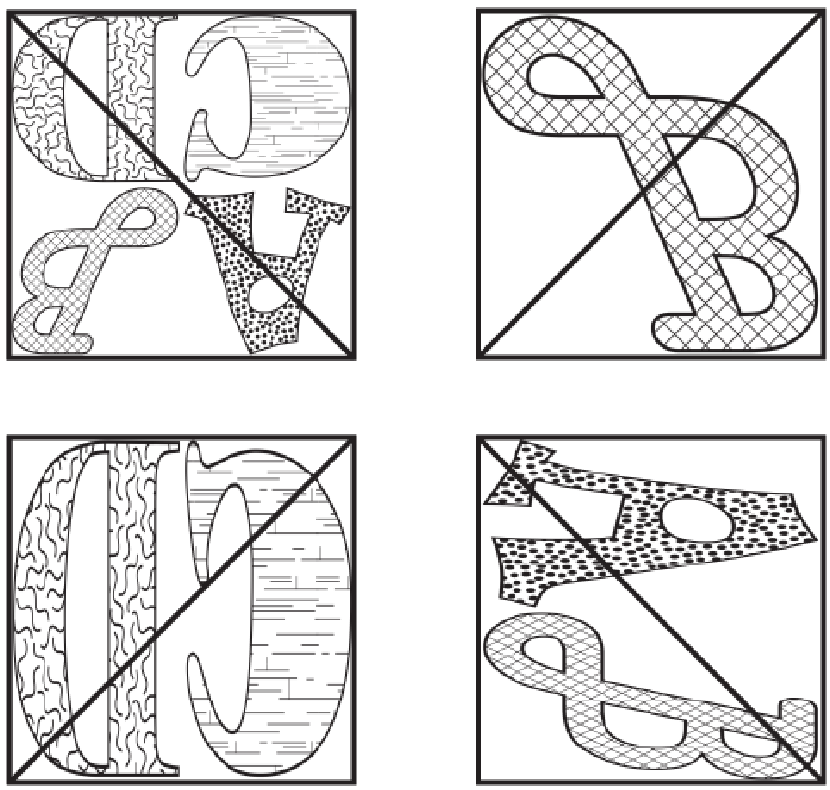
(Write "M","R", "MR", "G", or "N" next to each question below. You don't need to provide an explanation, unless you really feel that your answer comes with significant caveats.)

- ☐ A scene with a skybox.
- ☐ Caustics seen behind a glass full of water.
- ☐ A textured object that also exhibits specular reflection.
- ☐ Object shadows generated via shadow mapping.
- ☐ Applying decal textures to objects.
- ☐ Using dynamic environment maps to allow object surfaces to reflect the image of other objects within the scene.
- ☐ Rainbow effects caused by light refracted through glass.
- ☐ A scene containing a moving, hierarchically modeled object with many solid-colored, diffuse-shaded, parts.
- ☐ An object with a simple texture (for color), and a bump map for faking surface roughness.

3. [13%] Consider the texture map illustrated in the image below:



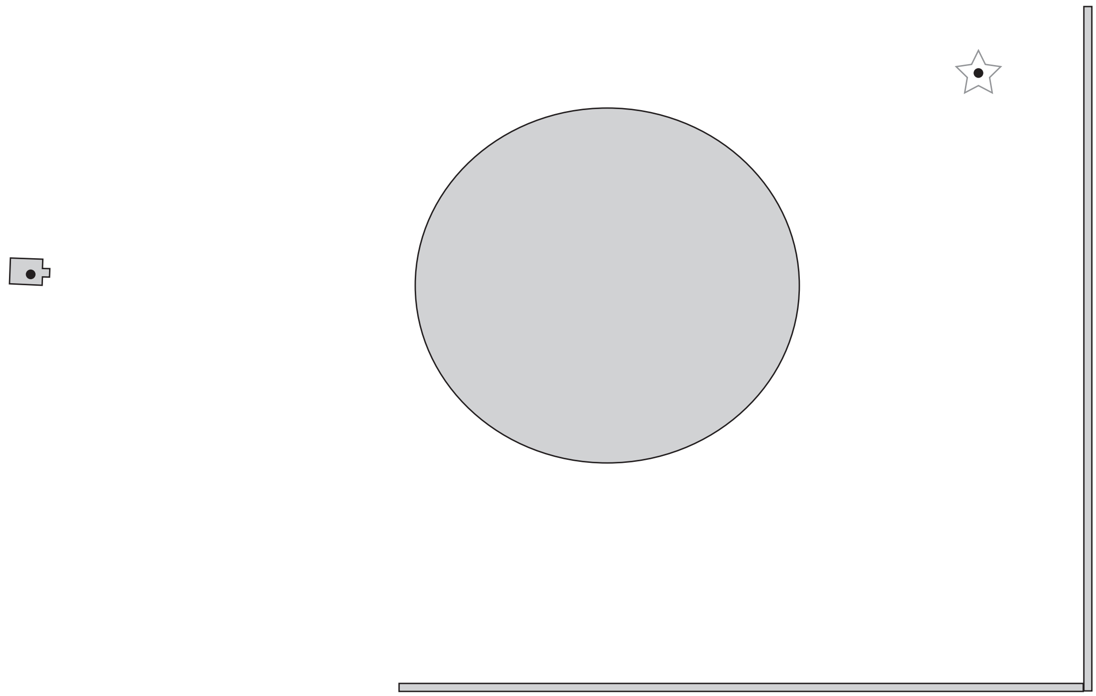
We want to apply this texture to the four square surface patches shown below, each of which has been split into two triangles:



For each vertex of each of the four squares, write the texture coordinates that we need to associate with this vertex in order to produce the desired apperance. Write the coordinates in the form (u, v) next to each vertex. *Note: Don't be confused by the fact that each square is "split" into two triangles. You can simply answer this question on a vertex-by-vertex basis. At the end, you will write 16* (u, v) *pairs of numbers, one next to each vertex of the squares above.*

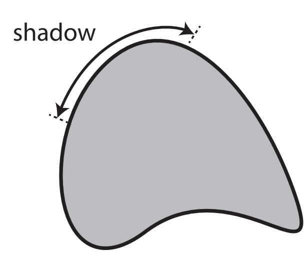
4. [13%] In the recursive ray tracing algorithm, we allow light to travel from the light to the camera via a number of bounces. Remember that, with the exception of the bounce closest to the light source (which can incorporate both diffuse and specular reflection), all other bounces are purely specular (mirror reflections).

   The illustration below illustrates a camera (left side), a light source (drawn as a star), and an environment consisting of opaque walls (shaded gray).



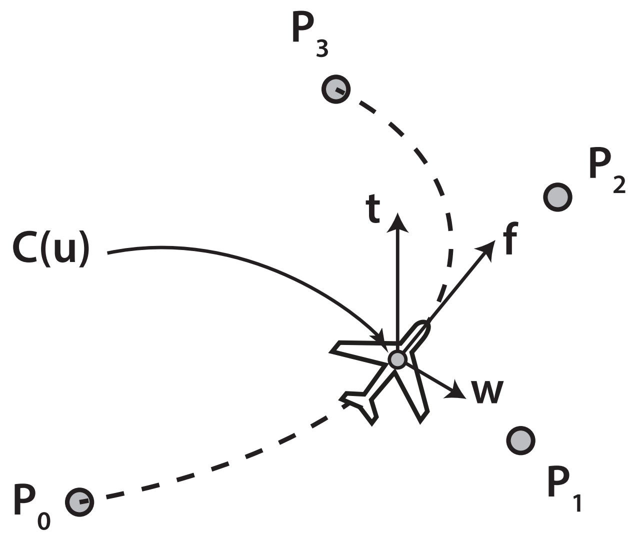
   (a) Assume that we use a recursive ray tracing algorithm that allows rays to sustain a **maximum of two** bounces before reaching the light. Thus, there will be some parts of the object surfaces in this scene that will be rendered as dark (shadowed), even if there is a direct line-of-sight between them and the camera (those surface locations will appear dark, because no ray cast from the camera towards them managed to get to the light with the allowable number of bounces).

   Indicate, along the object surfaces in this scene, which such locations are **visible, yet shadowed** from the camera's viewpoint. Draw arrows to indicate the extent of the shadow regions, as shown below.



   (b) Draw a light path with exactly **two** bounces, that connects the camera to the light source (without shadowing). **Clearly mark any angles along the path that are assumed to be equal.**

5. [14%] A plane is flying (in 3D) along a cubic Bézier curve $\mathcal{C}(u)$ as shown in the illustration below.



Remember that the Bézier curve is described by the expression

$$\mathcal{C}(u) = \begin{bmatrix} 1 & u & u^2 & u^3 \end{bmatrix} \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ -3 & 3 & 0 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{bmatrix}}_{\mathbf{B}} \underbrace{\begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{bmatrix}}_{\mathbf{P}}$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ -3 & 3 & 0 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 8 & 0 & 0 \\ 8 & 8 & 8 \\ 0 & 8 & 8 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 24 & 0 & 0 \\ -24 & 24 & 24 \\ 0 & -16 & -16 \end{bmatrix}$$

where $\mathbf{B}$ is the Bézier basis matrix, and $\mathbf{P}$ is the 4×3 matrix with the coordinates of the control points P_0, P_1, P_2, P_3 as its rows.

In the picture above, the control points have the coordinates $P_0 = [0, 0, 0]$, $P_1 = [8, 0, 0]$, $P_2 = [8, 8, 8]$, $P_3 = [0, 8, 8]$.

(a) Compute the position of the plane $\mathcal{C}(u)$ for parameter value $u = 0.5$.

(b) Compute a vector $\mathbf{f}$ that is *tangential* to the trajectory at $u = 0.5$, as shown in the illustration. The magnitude of the vector is not important, as long as the direction you compute is correct.

(c) For the same parameter value $u = 0.5$, compute a vector $\mathbf{w}$ ("wing" vector) such that (a) it is perpendicular to the direction of motion, and (b) it is perpendicular to the vector $\mathbf{t} = [0, 1, 0]$ (the "up" vector). *Note: It is acceptable to express* $\mathbf{w}$ *using a mathematical expression (e.g. a cross product). It is not essential to carry out the operations to compute the individual coordinates of* $\mathbf{w}$*.*

(Page intentionally left blank if you need more room for calculations on problem 5)