Last Name: ______________________ First Name: ____________ Cornell NetID: ________

CS 4620 Final, December 14, 2018

This 150-minute exam has 5 questions worth a total of 100 points. Use the back of the pages if you need more space.

Academic Integrity is expected of all students of Cornell University at all times, whether in the presence or absence of members of the faculty. Understanding this, I declare I shall not give, use, or receive unauthorized aid in this examination.

Signature: ______________________________________ Date ____________

1. [20 points] **Overview**

(a) True or False? No explanations are required.

1. ____ In a manifold triangle mesh, vertices A and B can be used together in at most two faces.
2. ____ In a manifold triangle mesh, if vertices A and B are used together in two distinct faces, they must always appear in the same order.
3. ____ When checking if a ray intersects some geometry, the ray direction must be normalized in order to get correct results.
4. ____ When interpolating rotations using quaternions, the resulting motion takes the shortest path between two orientations.
5. ____ To transform normals properly under a rigid motion, you must compute the inverse transpose of the matrix used to transform positions.
6. ____ To transform a direction by an affine transformation matrix in homogeneous coordinates, you multiply the matrix by the direction vector with a 1 in the fourth coordinate, then divide the resulting vector by its fourth coordinate.
7. ____ To transform a direction by an affine transformation matrix in homogeneous coordinates, you multiply the matrix by the direction vector with a 0 in the fourth coordinate, then divide the resulting vector by its fourth coordinate.
8. ____ In order to avoid visible bands in smooth areas of images, we need smaller quantization steps for intensity in the darker tones than in the lighter tones.
9. ____ If the basis functions of a spline are all positive, then the spline must stay inside the convex hull of the control points.
10. ____ If the barycentric coordinates of a point are all positive, then the point must be in the interior of the triangle.

(b) Short answer:

1. What type of GLSL variable, must have matching names in vertex and fragment shader code? What is it used for?

2. When sampling a texture, we usually define the texture image to span the unit square $[0, 1] \times [0, 1]$. What are two common ways of dealing with the case where the texture coordinates are outside that square?

3. In ray-triangle intersection, we often compute barycentric coordinates. How are these useful in the intersection operation, and what other purpose do they serve?

(c) Matching: for each of the following, write the number of the sentence fragment that correctly completes the statement. No explanations are required. The matching is one-to-one.

1. Ray-sphere intersection... ____
2. Ray-triangle intersection... ____
3. A ray tracing renderer... ____
4. A fragment shader... ____
5. A vertex shader... ____
6. A rasterization renderer... ____
7. A cubic B-spline... ____
8. A Catmull-Rom spline... ____
9. A Hermite spline... ____
10. Euler angle representation of rotations... ____
11. Linear quaternion interpolation... ____
12. Spherical linear interpolation... ____

(a) ...interpolates all its control points.
(b) ...involves solving a 3x3 linear system.
(c) ...processes objects one at a time.
(d) ...has problems with gimbal lock.
(e) ...interpolates at a constant speed between endpoints.
(f) ...runs before rasterization.
(g) ...runs after rasterizaion.
(h) ...involves solving a quadratic equation.
(i) ...interpolates at a non-constant speed between endpoints.
(j) ...maintains C^2 continuity between segments.
(k) ...is defined from the positions and tangents at its endpoints.
(l) ...processes pixels one at a time.

2. [20 points] **Perspective**

Consider the following viewing matrix and perspective projection matrix that might be used in OpenGL.

$$M_v = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad M_p = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 3 & 16 \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

A point $\mathbf{p}$ in world space transforms to the point $M_p M_v \mathbf{p}$ in NDC. We use the convention that in NDC, the near plane is $z = 1$ and the far plane is $z = -1$.

Hint: it is helpful to know the inverse matrices. They are (up to a scale factor, to keep the numbers simple):

$$M_v^{-1} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad M_p^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & \frac{1}{16} & \frac{3}{16} \end{bmatrix}$$

(a) What are the viewpoint and view direction (the direction the camera is looking)?

(b) What are the points in eye space that correspond to the NDC points $(0, 0, 1)$ and $(0, 0, -1)$?

(c) What are the near and far distances of the camera?

(d) What are the points in world space that correspond to the NDC points $(0, 0, 1)$ and $(0, 0, -1)$?

(e) What are the equations of the near and far planes in world space?

(f) What are the points in eye space that correspond to the NDC points $(0, 1, 1)$ and $(0, -1, 1)$?

(g) What is the vertical field of view of the camera, in degrees?

3. [20 points] **Splines**

The matrix form of the cubic Bézier spline is

$$\mathbf{p}(t) = \begin{bmatrix} 1 & t & t^2 & t^3 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ -3 & 3 & 0 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{p}_0 \\ \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \end{bmatrix}$$

(a) Evaluate this spline for $t = \frac{1}{3}$ and $t = \frac{2}{3}$.

(b) What is the unit vector tangent to this spline at $t = 0$ and $t = 1$, in terms of the control point positions?

We would like to define a cubic spline that is controlled entirely by position: the spline segment passes through the four control points $\mathbf{p}_0$, $\mathbf{p}_1$, $\mathbf{p}_2$, and $\mathbf{p}_3$ at the parameter values $t = 0$, $t = \frac{1}{3}$, $t = \frac{2}{3}$, and $t = 1$, respectively.

(c) Write an expression for the spline matrix of this spline; you may leave it in terms of the inverse of another matrix. Explain your reasoning. Use the ordering $\begin{bmatrix} 1 & t & t^2 & t^3 \end{bmatrix}$.

(d) Does this spline have the convex hull property?

4. [20 points] **Transformation**

(a) Label True/False for each of the 3D homogeneous transformations defined by the matrices below whether the following statements are true about it.

i. It is a rotation (around any axis).
ii. It is a clockwise rotation around the $+x$ axis.
iii. The origin is on its rotation axis or is a scaling center.
iv. The point (1,2,3) is on its rotation axis or is a scaling center.
v. It is an axis-aligned scale (around any point).

$$A_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 5 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix};\ A_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & -2 \\ 0 & 0 & 3 & -6 \\ 0 & 0 & 0 & 1 \end{bmatrix};\ A_3 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix};\ A_4 = \begin{bmatrix} 1 & 0.1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

	i.	ii.	iii.	iv.	v.
A_1					
A_2					
A_3					
A_4					

(b) Suppose we are given a modeling matrix M_o and a viewing matrix M_v. You are writing some code that has a point (x, y, z), a tangent vector (r, s, t), and a normal vector (a, b, c), all represented in object space. How are the eye space and world space coordinates of these three objects computed using these matrices? There are 6 answers, and each will be the product of one or more matrices and one or more 4-vectors.

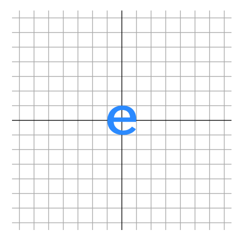
Please write your answers in the table on next page.

Last Name: ______________ First Name: ______________ Cornell NetID: ________

	World Space	Eye Space
Point (x,y,z)		
Tangent vector (r,s,t)		
Normal vector (a,b,c)		

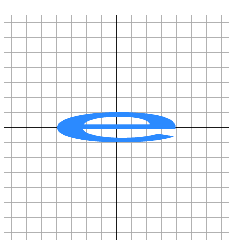
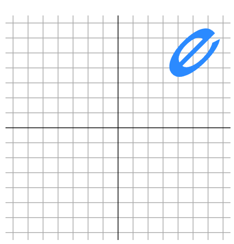
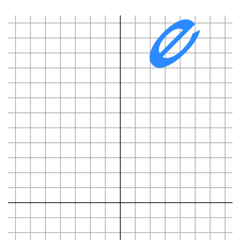
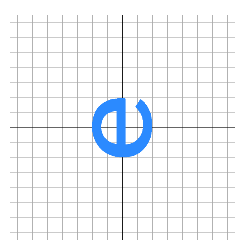
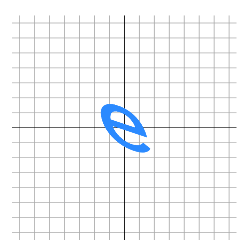
(c) Define the 2D affine transformation matrices:

$$T = \begin{bmatrix} 1 & 0 & 10 \\ 0 & 1 & 10 \\ 0 & 0 & 1 \end{bmatrix};\ S = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix};\ R = \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



Original Image

After being transformed by some product of these matrices, the letter "e" shape below maps to the shapes shown in diagrams i through iv. For each diagram, write the product of matrices below it that corresponds to that result. For instance, the answer for the diagram labeled "example" is SS or S^2. Individual matrices may be inverted too. Note the axis has shifted in diagram ii.

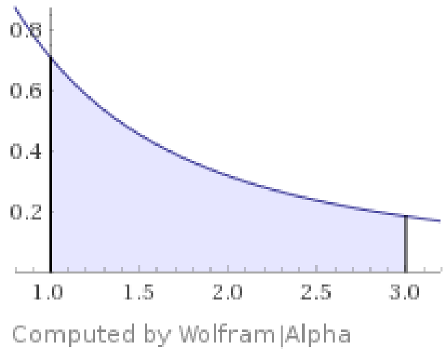
Example	i.	ii.	iii.	iv.
				
SS				

5. [20 points] **Monte Carlo Integration**

   (a) We would like to compute the definite integral

   $$I = \int_1^3 \frac{1}{\sqrt{x^3 + x}}\, dx$$

   with Monte Carlo integration.



   1. Suppose we compute a Monte Carlo estimate of I using a uniform probability density. What is the probability density $p(x)$, and what is the estimator $g(x)$?

   2. To improve our importance sampling, suppose we design a new estimator $g'(x)$ that uses a probability density $q(x)$ that is proportional to $1 - (2/5)x$. What is the probability density $q(x)$ and what is the estimator $g'(x)$?

   (b) We would like to estimate the light reflected from a diffuse surface using Monte Carlo integration. The surface reflectance is R, so the BRDF has the constant value R/π. The shape of the surface is convex, and there are no other objects in the scene. The only illumination comes from a constant environment with radiance L (i.e. the radiance seen in all directions is equal to L).

   Assume you have the following functions available:

   - `Vector3 sampleHemi(Vector3 normal)` — return a unit vector in the hemisphere around `normal` selected randomly according to the probability density $p_{\text{const}}(\omega) = 1/(2\pi)$.
   - `Vector3 sampleHemiCos(Vector3 normal)` — return a unit vector in the hemisphere around `normal` selected randomly according to the probability density $p_{\cos}(\omega) = \cos\theta/\pi$, where θ is the angle between ω and `normal`.
   - `boolean shadow(Vector3 shadingPoint, Vector3 shadowDir)` — return `true` if `shadingPoint` is shadowed in the direction of `shadowDir`.

   [Continued on next page]

1. Write Java-like pseudo-code for the function `reflectedRadiance` specified below. Your code only needs to work in the special case described above (meaning you know the surface reflectance and the incident radiance already, so you don't need to have BSDF or Surface or Scene objects like you would have in a real ray tracer.)

```
/*
 * Return a one-sample Monte Carlo estimate of the reflected radiance
 * from the surface.
 *    viewDir -- the direction in which light is leaving the surface
 *    shadingPoint -- the point on the surface where the light is reflecting
 *    normal -- the normal to the surfae at the shading point
 */
float reflectedRadiance(Vector3 viewDir, Vector3 normal, Vector3 shadingPoint) {



}
```

2. Now suppose:
   - The lighting comes from an environment map that supports only evaluation (not sampling), which can be evaluated by calling the global function `float environmentRadiance(Vector3 dir)`.
   - There may be other obects in the scene.

   Again write Java-like pseudo-code for the function `reflectedRadiance` specified below, continuing to use the same surface and scene as above.

```
/*
 * Return a one-sample Monte Carlo estimate of the reflected radiance
 * from the surface.
 *    viewDir -- the direction in which light is leaving the surface
 *    shadingPoint -- the point on the surface where the light is reflecting
 *    normal -- the normal to the surfae at the shading point
 */
float reflectedRadiance(Vector3 viewDir, Vector3 normal, Vector3 shadingPoint) {



}
```

in both (c) and (d) you can use either sampling strategy—one may, however, lead to a simpler estimator.