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

All coordinate systems are right-handed.

Good luck!

Do not write below this line

<table>
<thead>
<tr>
<th>Question</th>
<th>Max.</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>80</strong></td>
<td></td>
</tr>
</tbody>
</table>
1. Projection (10 Points)

The goal of projection is to transform three-dimensional object coordinates \((x, y, z)\) into two-dimensional screen coordinates \((x', y')\).

a. For perspective projection, assume the center of projection is at \((0, 0, 0)\) and the image plane is at \(z=d\). What are the formulas for calculating \((x', y')\) in terms of \((x, y, z)\) and \(d\)? (3 Points)

b. Draw a side view diagram to illustrate what happens to the projected image if we move an object closer to the image plane. Will the image become larger or smaller? (2 Points)

c. For parallel projection, assume the image plane is at \(z=d\). What are the formulas for calculating \((x', y')\) in terms of \((x, y, z)\) and \(d\)? (3 Points)

d. Draw a side view diagram to illustrate what happens to the projected image if we move an object closer to the image plane. Will the image become larger or smaller? (2 Points)
2. Vertex Transformation (10 Points)

Fill in the Complete Vertex Transformation flow chart below (between the dashed lines) with the correct names for matrices (square boxes) and coordinate systems (flat boxes). You can save time by drawing arrows for where the respective elements (left and right of the dashed lines) go.
3. Backface Culling (10 Points)

Two triangles are defined as follows, with indices defined as (0, 2, 1) and (0, 2, 3), in that order. Assuming that the two triangles are on the same plane, please list which triangles you could see with the following OpenGL settings with all of the possible correct visualization options. Assuming that the camera can be moved around freely in all three dimensions. Frustum culling is not considered in this question.

As a reminder, CCW means Counter-Clockwise whereas CW means Clockwise. The OpenGL command \texttt{glFrontFace} specifies the orientation of front-facing polygons. The default value is \texttt{GL\_CCW} for counter-clockwise.

![Diagram of triangles]

a. With the following settings in the code: (2 points)
\begin{verbatim}
    glEnable(GL_CULL_FACE);
    glCullFace(GL_BACK);
    glFrontFace(GL_CCW);
    glFrontFace(GL_CCW);
\end{verbatim}
We could see these views: ________________________________

b. With the following settings in the code: (2 points)
\begin{verbatim}
    glEnable(GL_CULL_FACE);
    glCullFace(GL_FRONT);
    glFrontFace(GL_CCW);
    glFrontFace(GL_CCW);
\end{verbatim}
We could see these views: ________________________________
c. With the following settings in the code: (2 points)
   glEnable(GL_CULL_FACE);
   glCullFace(GL_BACK);
   glFrontFace(GL_CW);
   We could see these views: ________________________________

d. With the following settings in the code: (2 points)
   glEnable(GL_CULL_FACE);
   glCullFace(GL_FRONT);
   glFrontFace(GL_CCW);
   We could see these views: ________________________________

e. With the following settings in the code: (2 points)
   glEnable(GL_CULL_FACE);
   glCullFace(GL_BACK);
   glFrontFace(GL_CW);
   glDisable(GL_CULL_FACE);
   We could see these views: ________________________________

For easier reference, here a copy of the image from the previous page:
4. Skyboxes (10 Points)

```glsl
#version 330 core

in vec3 FragPos;
in vec3 Normal;

uniform vec3 cameraPos;
uniform samplerCube skybox;

out vec4 color;

void main()
{
    vec3 I = normalize(FragPos - cameraPos);
    vec3 R = reflect(I, normalize(Normal));
    // Placeholder for question 2
    color = vec4(texture(skybox, R).rgb, 1.0);
}
```

a. Which part of the code of homework project 4 would you see the code above? Be specific. (2 points)

b. Which of the following lines of code would affect the sampling result if was is inserted in the position of the commented placeholder? Briefly explain what you would see (be specific) if this is used in homework project 4 with the camera position provided in the starter code? (2 points)

1. I = -R
2. R = -R
3. Normal = -Normal
c. Based on the following image of a scene with a skybox in top-down view, specify what color you would see on the selected point, as asked below. Assume that face culling is off, and all sides of the cube are within the camera’s frustum.

![Skybox Diagram](image)

1. What color would you see from the camera location in **point A**? Also, draw the path of the light from skybox wall to camera. (2 points)

2. What color would you see from the camera location in **point B**? Also, draw the path of the light from skybox wall to camera. (2 points)

d. When deciding on the size of a skybox (i.e., the width of its walls), what are the limitations for its size to work correctly? Name one limitation which affects the lower end of the size range (“needs to be big enough to…”), and one that affects the upper end of the size range (“needs to be small enough to…”). (2 points)
5. Bezier curves (10 Points)

Suppose a Bézier curve $X(u)$ is defined by the following four control points in the $xy$-plane: $P_0 = (-2, 0)$, $P_1 = (-2, 4)$, $P_2 = (2, 4)$ and $P_3 = (2, 0)$. Answer the following questions:

a. What is the degree of $X(u)$? (1 point)

b. Geometrically compute $X(0.25)$ with de Casteljau’s algorithm: sketch the control points and add all additional points the de Casteljau algorithm produces, until you get $X(0.25)$. What are the $x/y$ coordinates for $X(0.25)$? (6 points)

c. Divide the curve at $X(0.25)$ into two separate Bezier curves with the same values for $X$, and list the control points of each curve segment in correct order (3 points).
6. Continuity (10 Points)

a. Explain what it means to be C0 continuous? Draw an example of a curve that is C0 continuous but NOT C1 continuous (2 points)

b. Explain what it means to be C1 continuous? Draw an example of a curve that is C1 continuous (2 points)

c. Given the following equation for a Bezier curve: \( Q = (3-t^2) p_0 + (2t - 2t^2) p_1 + t^2 p_2 \)
If we wanted to connect another curve (R) to the end of curve Q, then in terms of p0, p1, and p2, what needs to be true about curve R to have the combined curve be C0 continuous? (3 points)

d. Let the equation for R be \( R = (3-t^2) p_2 + (2t - 2t^2) p_3 + t^2 p_4 \). Will the combined curve described in part c be C1 continuous? Explain why. (3 points)
7. 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.

a. Using the height of the brick wall as seen in the image, calculate (approximately) 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)

b. In the perspective image above, sketch approximately which regions of the wall on the left (defined by points $P_1, P_2, P_3$ and $P_4$) will use each of the mipmap textures on the right (use nearest-mipmap interpolation). (3 points)

c. Explain the difference between nearest-mipmap interpolation and trilinear mipmapping. (2 points)

d. How does Anisotropic Filtering differ from regular mipmapping? (2 points)
8. Procedural Terrain (10 Points)

a. Midpoint Displacement Algorithm (4 Points)

Given the initial line below, with end points A and B, draw the first two steps of the midpoint displacement algorithm below it, using offset parameters of your choice. Highlight all end points of your new lines.

Step 0:

```
A  B
```

Step 1:

```
A  B
```

Step 2:

```
A  B
```

b. Diamond Square Algorithm (6 Points)

In the 5x5 array of terrain nodes below, write in each node the number of the step in which its height is calculated in the Diamond Square Algorithm. The initial Step 0 is already filled in.

```
0   0   0   0   0
0   0   0   0   0
0   0   0   0   0
0   0   0   0   0
0   0   0   0   0
```