CS-315+733 Midterm Exam
October 30, 2024, 14:30 – 15:45, CL112
D. Hepting

This is a closed book exam. You must maintain the confidentiality of your examination; do not provide any opportunity for others to copy any of your work. Electronic devices are NOT permitted during the exam. Please turn off and put away all cell phones and other electronic devices during the exam period.

ANSWER ALL QUESTIONS. All answers must be written on this exam in the space provided. You have 75 minutes to complete the exam. Please plan your answers, favour quality over quantity, do not exceed the space provided, and do your best to write legibly. QUESTIONS ARE ON BOTH SIDES OF THE PAPER. This exam contributes 15 percent towards your final grade. The exam is marked out of 77.

Name (printed): ____

Student Number: ____

Signature: ____

***

Q1. (4 marks) You have a canvas that is 512 pixels wide and 512 pixels high.

- What is the width of the clip space? __________
- What is the height of the clip space? __________
- what proportion of clip space is associated with each of those pixels?

Q2. (6 marks) Describe the relationship between RGB and HSV (Hue Saturation Value) colour models?

Q3. (6 marks) Describe the following concepts and how they are related to one another: continuous, discrete, sampling, and quantization.

Q4. (5 marks) Consider RGB and CMY

- Which is subtractive? __________
- Which is additive? __________
- Which is for screens? __________
- Which is for printers? __________
- Why do colour printers have black ink?

Q5. (3 marks) What is object space? What is image space? In which space does the z-buffer algorithm operate?

Q6. (3 marks) What are triangle strips and why are they important? Are there alternatives?

Q7. (2 marks) Why is the flatten function required in code from the textbook?

Q8. (5 marks) What is the purpose of a z-buffer? Explain how it is used.

Q9. (5 marks) Describe how to specify coordinates for the camera using the “lookAt” method.

Consider the following code fragments in questions Q10-Q12 that follow:

```
<script id="vertex-shader" type="x-shader/x-vertex">
#version 300 es
in vec4 aPosition;

void
main()
{
    gl_Position = aPosition;
}
</script>

<script id="fragment-shader" type="x-shader/x-fragment">
#version 300 es
precision mediump float;
out vec4 fColor;

void
main()
{
    fColor = vec4(1.0, 0.0, 0.0, 1.0);
}
</script>

[…]
const bufferId = gl.createBuffer();
gl.bindBuffer(gl.ARRAY_BUFFER, bufferId);
gl.bufferData(gl.ARRAY_BUFFER, flatten(positions), gl.STATIC_DRAW);
const positionLoc = gl.getAttribLocation(program, 'aPosition');
gl.vertexAttribPointer(positionLoc, 2, gl.FLOAT, false, 0, 0);
gl.enableVertexAttribArray(positionLoc);
[…]
```

Q10. (12 marks) Describe the changes or additions to the vertex shader, the fragment shader, and the JavaScript needed to use to per-vertex colours.

Q11. (6 marks) What is the effect of the following code (as an example) to add data to the `positions` array? Is there a better way to do this? (`currmat` is a 3x3 matrix)

```
positions.push(mult(currmat,vec3(0,0,1)));
```

Q12. (4 marks) Why does this simple vertex shader work properly? What limitations does it have?

Questions Q13-Q15 deal with the following 4x4 matrix.

$$\begin{matrix} 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{matrix}$$

Q13. (4 marks) Describe the effect of this transformation matrix.

Q14. (8 marks) Compute the transformation matrix that centers the above rotation at the point (5,-7,2)

Q15. (4 marks) Verify that your new transformation matrix is correct. How can this be done?