

CS-315+733 Midterm Exam  
October 30, 2024, 08:30 – 09:45, 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 63.

Name (printed): \_\_\_\_\_

Student Number: \_\_\_\_\_

Signature: \_\_\_\_\_

\*\*\*

Q1. (6 marks) Explain vector and bitmap images. Indicate a positive and negative of each.

Q2. (6 marks) Describe a process of creating an image of size 1024x1024 with 24 bit colour. Highlight and explain the concepts of **continuous**, **discrete**, **sampling**, and **quantization**. How many bits are required for such an image?

Q3. (4 marks) WebGL2 is an application programming interface for what purpose? Why?

Q4. (6 marks) What is the purpose of a z-buffer? Explain how it is used. Why not just sort elements from far to near?

Q5. (6 marks) What inputs are required for the “lookAt” method and how are they used to compute camera coordinates?

Q6. (4 marks) Fill in the table cells with the elements of a rotation matrix that takes the X axis to the axis; the Y axis to the -Y axis; and the Z axis to the X axis.

|  |  |  |  |
|--|--|--|--|
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

```
<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 fColour;

void
main()
{
    fColour = 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);
[...]
```

Q7. (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? \_\_\_\_\_

Q8. (4 marks) Why does the simple vertex shader (above) work properly? What limitations does it have?

Q9. Describe the changes or additions to the vertex shader, the fragment shader, and the JavaScript (no coding required) needed to:

a) (6 marks) use to per-vertex colours

b) (6 marks) transform the vertex positions

Q10. (4 marks) What is YCbCr and why is it suited to (lossy or lossless?) compression?

Q11. (6 marks) Describe how you would show mathematical empathy when rotating (see Q6) about a point other than the origin? Use the point (1, -2, 3) as the new center of rotation in your answer.



