HÁSKÓLINN Í REYKJAVÍK
REYKJAVIK UNIVERSITY

Computer Graphics
T-511-TGRA
Final exam

Teacher: Kári Halldórsson

Date: December 2nd 2013

Time: 14:00 – 17:00

Supporting materials:
Included “cheat sheet”
Non-programmable calculators allowed

Answers can be given in English and/or Icelandic

Name:____

ID: ____

1. **The graphics pipeline (10%)**
   Draw up the main components of the graphics pipeline and label them. Imagine a vertex sent through the pipeline. Describe it's "journey" through the pipeline. Mark each change made to the vertex and it's associated data and label/name the calculations and algorithms it goes through.
   There is no need to describe calculations in detail. Just name/label the data/calculations/algorithms used.

2. **Lighting (10%)**
   Describe per vertex lighting and the combination of different "types" of light. What geometrics are used, what data produced and how is it calculated?

3. **Matrix Transformations (40%)**
   A camera is set up to be positioned in (3,3,3) looking at the point (2,0,-2). It has an up vector (0,0,1).
   If at any point you have no answer at all to one question needed to continue to the next, use unit vectors and/or identity matrices and add clear explanations for what you're doing.

   a) Find the point of origin and vectors for the camera's coordinate frame (15%)

b) Set up the matrix to calculate eye coordinates for the camera (5%)
Which matrix is this?

c) Add to that matrix a translation by (2,5,4) (10%)

d) What are the eye coordinates for the vertex with world coordinates (2,6,4) *(considering the current state after **b** and **c**)*? (10%)

4. **Perspective Projections (10%)**
   Consider the following code:

   ```
   gluPerspective(90.0, 1.25, 1.0, 110.0);
   ```

   What happens in the OpenGL state machine when it is run?
   Which matrix is affected and what are it's exact values afterwards?

5. **Rasterization (10%)**

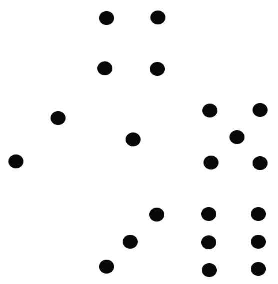
Three vertices of a triangle have been sent through the OpenGL pipeline.
Their pixels and greyscale color values are:
P1: (4,3) - c1: 0.3
P2: (5,8) - c2: 0.7
P3: (1,6) - c3: 0.6

After shading the whole triangle what will the color value be in pixel (3,5)?

6. **Clipping (10%)**
   Consider the window W(l,r,b,t) = (-200, 200, -100, 100).
   A line has two endpoints, P1=(-200, -150) and P2=(300, 100).
   Use the Cohen Sutherland clipping algorithm to clip the line against the window.
   Show the steps the algorithm takes, the data at the beginning and and of each step (calculated values) and how each step is exited. Beyond that exact/specific calculations are not necessary, so long as steps are shown and each result is correct.

```
public class D6 {
    FloatBuffer vertexBuffer;
    FloatBuffer texCoordBuffer;
    Texture texture;

    public D6(Texture tex)
    {
        vertexBuffer = BufferUtils.newFloatBuffer(72);
        vertexBuffer.put(new float[] {
            -0.5f, -0.5f, -0.5f, -0.5f, 0.5f, -0.5f,
            0.5f, -0.5f, -0.5f, 0.5f, 0.5f, -0.5f,
            0.5f, -0.5f, -0.5f, 0.5f, 0.5f, -0.5f,
            0.5f, -0.5f, 0.5f, 0.5f, 0.5f, 0.5f,
            0.5f, -0.5f, 0.5f, 0.5f, 0.5f, 0.5f,
            -0.5f, -0.5f, 0.5f, -0.5f, 0.5f, 0.5f,
            -0.5f, -0.5f, 0.5f, -0.5f, 0.5f, 0.5f,
            -0.5f, -0.5f, -0.5f, -0.5f, 0.5f, -0.5f,
            -0.5f, 0.5f, -0.5f, -0.5f, 0.5f, 0.5f,
            0.5f, 0.5f, -0.5f, 0.5f, 0.5f, 0.5f,
            -0.5f, -0.5f, -0.5f, -0.5f, -0.5f, 0.5f,
            0.5f, -0.5f, -0.5f, 0.5f, -0.5f, 0.5f
        });
        vertexBuffer.rewind();


        texCoordBuffer = BufferUtils.newFloatBuffer(_________);
        texCoordBuffer.put(new float[] {







        });
        texCoordBuffer.rewind();


    }

    public void draw()
    {
        Gdx.gl11.glVertexPointer(3, GL11.GL_FLOAT, 0, vertexBuffer);





        Gdx.gl11.glNormal3f(0.0f, 0.0f, -1.0f);
        Gdx.gl11.glDrawArrays(GL11.GL_TRIANGLE_STRIP, 0, 4);
        Gdx.gl11.glNormal3f(1.0f, 0.0f, 0.0f);
        Gdx.gl11.glDrawArrays(GL11.GL_TRIANGLE_STRIP, 4, 4);
        Gdx.gl11.glNormal3f(0.0f, 0.0f, 1.0f);
        Gdx.gl11.glDrawArrays(GL11.GL_TRIANGLE_STRIP, 8, 4);
        Gdx.gl11.glNormal3f(-1.0f, 0.0f, 0.0f);
        Gdx.gl11.glDrawArrays(GL11.GL_TRIANGLE_STRIP, 12, 4);
        Gdx.gl11.glNormal3f(0.0f, 1.0f, 0.0f);
        Gdx.gl11.glDrawArrays(GL11.GL_TRIANGLE_STRIP, 16, 4);
        Gdx.gl11.glNormal3f(0.0f, -1.0f, 0.0f);
        Gdx.gl11.glDrawArrays(GL11.GL_TRIANGLE_STRIP, 20, 4);


    }
}
```

7. **Graphics programming (10%)**
The class *D6* (previous/opposite page) contains data to draw a 3D cube. Imagine that the texture sent into it's constructor is this image:



a) Add to the class D6 the code needed to use this texture and to map a different numbered face to each of the cube's sides. You can add code between any lines in the code on the opposite page.
Hint: You need to set up the UV coordinates (between 0 and 1) for all the cube's vertices.
(5%)

b) A separate dice rolling engine returns a list of the structure RolledD6:
(5%)

```
Public class coords {
      public float x;
      public float y;
      public float z;
}
public class RolledD6 {
      public float size;
      public coords position;
      public coords rotationAxis;
      public float rotationAngle;
}
```

Finish code that will display each of the rolled dice of the size indicated, at the correct position and rotated by the rotation angle around the vector called rotation axis.

```
Void drawDiceSet {
      List<RolledD6> diceList = DiceRollingEngine.getDice();



      foreach(RolledDice d in diceList) {







      }





}
```

Bonus 3%
In which movie does the following dialog occur?
Who is character A?
Who is character B?

A: I know Kung-fu!
B: Show me.