HÁSKÓLINN Í REYKJAVÍK
REYKJAVIK UNIVERSITY

Computer Graphics
T- 511 – TGRA

Final Exam

Teacher: Kári Halldórsson

Date: 12. November, 2015

Time: 14:00

Helping materials: non-programmable calculator

Name: _____

Kt.: _____

Answers can be given in English and/or Icelandic.

1. **(10%) Transformations and matrices**
   You have access to the same matrix and graphics classes as we built together this semester, that is a BoxGraphic class with a drawSolidCube that sends vertices into the pipeline for a cube of the size 1x1x1, centered in (0,0,0), and a ModelMatrix class with the basic transformation operations, matrix stack operations and shader manipulation operations.
   These classes have been created and initialized elsewhere. You don't need to remember the exact names of functions, just the idea behind each one you need to call, and what parameters they take.

   What you want to draw is the following scene:

   A box of size 2x2x2, centered in (9,5,-2) and under it a big flat floor which is 10x10 in the x-z plane but only 0.8 thick. Its *corner* should be in (0,0,0) and its top edge level with the base x-z plane.

   a) (5%) Write the code that would draw the scene described. Imagine you're already inside the display function and lights, colors & cameras have already been set.

   b) (5%) Show the values that are in the shader's model matrix when the first box is drawn.

2. (10%) Cohen-Sutherland Clipping

A clipping window has the following geometry:

Window(left, right, bottom, top) = (200, 600, 100, 400)

A line with the following end points is drawn in the world:

P1: (240, 480)
P2: (140, 300)

Show how the Cohen-Sutherland clipping algorithm will clip these lines and what their final endpoints, if any, are. Show the coordinate values of P1 and P2 after each pass of the algorithm.

3. **(10%) Transparency**
   How could one render a transparent object in OpenGL?
   - Where in the OpenGL pipeline would this affect the calculations (and very briefly how)?
   - What would one specifically have to consider, as a programmer using OpenGL, when rendering a transparent object, in order for the effect to appear as intended?

4. (20%) Shaders

On the following two pages there are shaders for use in an OpenGL pipeline, a *vertex shader* and a *fragment shader*. You can add both *uniform* and *varying* variables to these shaders above their main() function definition and you can add any number of variables and code inside the main functions.

The following types can be used:

float:	a floating point number
vec4:	a 4 coordinate vector, used for positions, directions and colors
mat4:	a 4x4 matrix

The following GLSL functions can be used:

float dot(vec4 v1, vec4 v2):	Returns the dot product of two vectors
vec4 normalize (vec4 v):	Returns the normalized vector
float length(vec4 v):	Returns the length of the vector
vec4 cross(vec4 v1, vec4 v2):	Returns the cross product of two vectors
float max(float a, float b):	Returns the higher value
float min(float a, float b):	Returns the lower value
float pow(float num, float exp):	Returns *num* to the power of *exp*

+, -, *, / are done component-wise on *vec4* but as matrix multiplications when one or both sides are *mat4*.

Variables are already defined for the attributes *a_position* and *a_normal* for each vertex that will be sent through the pipeline.

The current shaders both have the same initial definitions at this point. **Strike out those that are unnecessary** in each shader and add whatever definitions you need for each of the following problems.

a) **(5%)** Add variables and calculations needed to transform the position attribute (vertices) to global coordinates, eye coordinates and clip coordinates and set the value for the built-in output variable *gl_Position*.

b) **(10%)** Add variables and calculations needed for the shaders to do per-fragment lighting. Include ambient, diffuse and specular lighting for a single light source. Set the final color value to the built-in output variable *gl_FragColor*.

c) **(5%)** Fog is a color that is mixed with the final fragment color based on distance from the camera. Anything closer than u_startFog is 100% the fragment color, anything further away than u_endFog is 100% the fog color. Anything in between is a weighted average of the two based on the proportion of the distance in the startFog to endFog range.
Add variables and calculations to the shaders to implement this functionality.

Vertex shader:

```
attribute vec3 a_position;
attribute vec3 a_normal;




uniform vec4 u_lightPosition;
uniform vec4 u_lightColor;


uniform vec4 u_materialAmbient;
uniform vec4 u_materialDiffuse;
uniform vec4 u_materialSpecular;
uniform float u_materialShininess;







void main()
{

}
```

Fragment shader:

```
attribute vec3 a_position;
attribute vec3 a_normal;




uniform vec4 u_lightPosition;
uniform vec4 u_lightColor;


uniform vec4 u_materialAmbient;
uniform vec4 u_materialDiffuse;
uniform vec4 u_materialSpecular;
uniform float u_materialShininess;







void main()
{









}
```

5. Camera Transformations (30%)

a) **(15%)**
A camera is set up to be positioned in (0,8,4)
looking at the point (0,3,-1).
It has an up vector (0,0,-1).
Find the point of origin and vectors for the camera's coordinate frame.

b) **(5%)** Set up the values in a matrix that represents this position and orientation of a camera.
Which matrix in your shader should be set to these values?

c) **(10%)** The camera should have a field of view of 75°, an aspect ratio of 16:9, a near plane at 3 and a far plane at 25. Find the exact values for a matrix that calculates this camera.
Which matrix in your shader should be set to these values?

6. **(10%) Rasterization**

Three vertices of a triangle have been sent through the OpenGL pipeline. They have the following pixel positions as well as values for the varying variable v_d:

P1: position = (3,12) – v_d = 14
P2: position = (11,10) – v_d = 19
P3: position = (9,6) – v_d = 11

What will the fragment shader value of v_d be set to at pixel (8,8)?

7. (10%) Vector intersections and reflections

A line has end points (3,8) and (7,6).
A particle starts at (4,2) and travels along in the direction (1,3).

a) (7%) In which point does the path of the particle cross the line?

b) (3%) If the particle is made to bounce off the line, what will it's new direction vector be?

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

A: Don't be absurd. You'd be killed!
B: I'm a good swimmer.
A: The fall alone would kill you.
B: It would hurt. I'm not saying it wouldn't. Tell you the truth, I'm a lot more concerned about that water being so cold.
A: How cold?
B: Freezing. Maybe a couple degrees over. You ever, uh, you ever been to Wisconsin?
A: What?
B: Well, they have some of the coldest winters around. I grew up there, near Chippewa Falls. I remember when I was a kid, me and my father, we went ice fishing out on Lake Wissota. Ice fishing is, you know, where you...
A: I know what ice fishing is!
B: Sorry. You just seem like, you know, kind of an indoor girl. Anyway, I, uh, I fell through some thin ice; and I'm telling you, water that cold, like right down there. It hits you like a thousand knives stabbing you all over your body. You can't breathe. You can't think. At least, not about anything but the pain. Which is why I'm not looking forward to jumping in there after you.