BROCK UNIVERSITY

Final exam
Course: COSC 3P98 Computer Graphics
Date of exam: Tuesday April 26, 2011
Time of exam: 1900-2200

NAME (print): ___________________________________
STUDENT NUMBER: ________________________________

There are 7 questions totaling 154 marks.
A calculator without a memory bank may be used. No other aids are permitted.
Use or possession of unauthorized materials will automatically result in a grade of zero for this examination.
Please answer all questions on the exam paper. Use the back of pages if necessary.
Keep written answers brief and to the point. Write neatly and legibly.
A grade of 40% is required on this exam to pass the course.

<table>
<thead>
<tr>
<th>Question</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>24</td>
</tr>
<tr>
<td>2</td>
<td>20</td>
</tr>
<tr>
<td>3</td>
<td>20</td>
</tr>
<tr>
<td>4</td>
<td>15</td>
</tr>
<tr>
<td>5</td>
<td>20</td>
</tr>
<tr>
<td>6</td>
<td>15</td>
</tr>
<tr>
<td>7</td>
<td>40</td>
</tr>
<tr>
<td>Total:</td>
<td>154</td>
</tr>
</tbody>
</table>
Question 1 [24] Multiple choice: select the best answer to each question
[+2 correct, 0 nothing, -0.5 wrong guess; one free wrong guess without penalty]

1. Computational geometry is:
   a. the study of polygonal mesh surfaces
   b. required when finding specular reflection
   c. finding algorithmic solutions to geometric problems
   d. computationally a pain

2. HSV stands for:
   a. Human Situational Velocity
   b. Hue Saturation Value
   c. Hunter-green Sepia Violet
   d. Hierarchical Screen Video

3. Object definitions started with “glBegin” are normally terminated with:
   a. glEnd
   b. glExit
   c. gluOrtho
   d. glVertex

4. The depth sort algorithm is also known as:
   a. backface culling
   b. depth buffering
   c. illumination
   d. painter’s algorithm

5. Which is a false statement about the “naive” triangulation algorithm:
   a. it requires a second phase that “cleans up” the triangulation.
   b. it creates a minimal-weight triangulation.
   c. it requires a second phase that creates polygon structures from edges
   d. it performs lots of line segment intersection tests.

6. Which is a false statement about OpenGL’s colour map mode:
   a. any colour from 24-bit RGB space may be selectable
   b. only a finite number of colours can be displayed at once.
   c. it is the preferred method for displaying photorealistic shading
   d. it is useful for defining colour palettes in paint programs
**Question 1. (cont)**

7. The midpoint line drawing algorithm:
   a. scan converts lines with no stair casing artifacts
   b. is floating-point intensive
   c. is used to render solid polygons
   d. none of the above

8. In OpenGL, 2D rotation is equivalent to:
   a. rotating around the Z-axis
   b. rotating an object around an arbitrary pivot point
   c. skewing an object in the X and Y directions
   d. a non-affine transformation

9. A procedural texture is:
   a. an algorithmically defined surface geometry
   b. an object-precise illumination model
   c. a Voronoi diagram
   d. Perlin noise

10. A characteristic of vector graphics terminals is:
    a. screen pulsation when scenes become too complex
    b. first example of photorealistic graphics hardware
    c. lovely curved surfaces with minimal aliasing
    d. used in Nintendo DS systems

11. A 3D scanner:
    a. creates a 2D projection from a 3D model
    b. creates a 3D physical model from a 3D mathematical model
    c. creates a 3D virtual reality from a 3D animation
    d. creates a 3D mathematical model from a 3D physical model

12. Ambient lighting is used to:
    a. show surfaces in shadowed areas
    b. create glossy reflections
    c. create mirror reflections
    d. complement John Tesh music during romantic dinners
Question 2 [20] Define and briefly discuss the following terms or OpenGL/GLUT commands:

a) simple polygon

b) clipping

c) scan conversion

d) glutIdleFunc (F)

e) double buffering
Question 3 [20]

(a) [10] Create a compositional homogeneous 2D transformation matrix consisting of these two transformations applied to a vertex in precisely this order: (1) translate(dx, xy) (2) rotate(A). Include notation representing the vertex in your answer.
Question 3 (cont)

(b) [10] Now create an inverse transformation for the compositional transformation matrix in (b) above. Use the same format as you used in (a). Describe and show all the steps in creating this matrix.
Question 4 [15]

Identify the 3 types of lighting effects found in OpenGL’s basic lighting model. Discuss how each effect contributes to the overall lighting of an object. Include equations for the monochrome versions of these effects, and discuss the equations and their parameters.
Question 5 [20]

(a) [4] Define Z-buffering.

(c) [6] Describe how it works (with a diagram).
Question 5 (cont)

(c) [4] Identify 2 sources of potential overhead involved with it.

(d) [6] Describe how to use it in OpenGL programs.
Question 6 [15]  Select one of the following:

(a) Give pseudo-code for the trisection algorithm with cleanup. Discuss what the algorithm steps are doing. Include a diagram to help your discussion. Describe the kind of triangulation obtained at each stage of it. (Note: you do not need to include the algebraic formula used; however, do describe the general calculations required at various steps).

(b) Discuss how OpenGL implements the inversion or “undoing” of sequences of transformations. Mention the kind of data structure it uses, how it is used, and the OpenGL commands for performing inversions. Describe a simple example of when such a transformation might be required during graphics programming.
Question 7 [40]

Write C/OpenGL/GLUT code that does the following. Read the entire question before beginning. Some parts rely on the use of others. You can assume that GLUT callbacks may invoke your functions. Please do not implement anything that is not requested (ie. user commands, lighting, textures, menus, SKYNET, etc.). Use the backs of pages if necessary.

(a) [10] Write some OpenGL code that renders a tetrahedron (see illustration). It should be rendered around the origin (0, 0, 0). Assume that the window has been set up already, and that the code is called via a GLUT callback. The coordinates of the vertices are:

A(0, -10, 10), B(-10, -10, 10), C(10, -10, 10), D(0, 10, 0)

A random colour is given to each vertex of every triangle. You can assume a call to RAND returns a random integer value between 0 and 255.
Question 7 (cont)

(b) [10] Define a C record structure that could be used to define a list of tetrahedrons, that could be used in a particle simulation. Each individual tetrahedron has the following information that should be saved:

- Location of center of tetrahedron in 3D space
- RGB colour of each vertex
- Rotational spin of tetrahedron
- Direction vector
- Speed at which object moving in direction vector
- Uniform scale (size) of tetrahedron
Question 7 (cont)

(c) [20] Write a C function that might be called in a loop in a GLUT callback, such that it updates and renders one single tetrahedron as recorded in the above data structure, in a single frame of an animation loop. Use the record structure defined in (b) in a sensible manner when rendering the tetrahedron. The following updates should be performed as well:

(i) A small gravitational effect is applied to its Y direction.
(ii) It moves at a Speed factor along its direction vector.
(iii) It bounces off the ground (at Y=0).
(iv) It loses some speed when it bounces.
Question 7 (cont)

```glsl
*** glEnd(); ***
```