EXAM COVER SHEET

RMIT University Examinations

EXAMINATION DETAILS:

Course code/s: COSC1186/1187
Course name/s: Interactive 3D Graphics and Animation
Date of exam: Tue 7th June 2011
Time of exam: 2:00pm to 3:15pm
Duration of exam: 1:15
Total no. of pages: 3

ALLOWABLE MATERIALS AND INSTRUCTIONS TO CANDIDATES:

1. Write your full name and student number on each examination booklet together with the number of examination books used. Students must not write during reading time.

2. This examination paper contains 4 questions

3. Attempt all parts of all questions. Commence each question on a new page. Carry out the instructions on the front cover of the examination booklet

4. This exam accounts for 25% of the total marks for the course.

5. This exam totals 25 marks.

6. Marks for each question are shown.

7. This exam is closed book.

8. Calculators are not permitted.
Question 1

Give a concise answer (1–2 lines) to each of the following questions:

(a) Give the command line for compiling a program teapot.c which uses OpenGL, GLUT and GLU.

(b) What is the relationship between frame rate and frame time?

(c) Show the difference between glVertex3f and glVertex3fv by giving an example call of each.

(d) Who is Ivan Sutherland?

(e) Draw triangles which allow the values of cos(30°), sin(30°) and sin(45°) to be determined.

(1 mark each = 5 marks)

Question 2

(a) Draw a diagram of the graphics pipeline and discuss bottlenecks which may occur in computer graphics performance.

(b) For a scene which consists of a single sphere, when increasing the tessellation the sphere which part of the pipeline is likely to become the bottleneck? Why?

(c) Discuss the advantage of using quad strip primitives compared to individual quads.

(4+1+1 = 6 marks)

Question 3

A unit square is transformed by performing a translation of 4.0 in x and a rotation of 60° about z, as shown below. A set of world coordinate axes, a set of local coordinate axes and a grid are also drawn.

(a) Assuming functions void drawUnitSquare(), void drawGrid(float length, float step) and void drawAxes(float length) are available, give code which draws the world coordinate axes, the grid, performs transformations and draws the transformed square and the local coordinate axes, as shown in the image on the right.

(b) Give the individual 4×4 transformation matrices and the combined transformation matrix.

(2+3 = 5 marks)
Question 4

The wireframe renderings below show Saturn and its rings as it rotates. The rings are edge on, and so are barely visible. Saturn’s equator is drawn in yellow. Two sets of axes are shown, one for the world coordinate system and one for Saturn. Saturn animates by rotating about its $y$ axis, which is tilted 27° to the world coordinate system $y$ axis.

(a) Write a function void drawSphere(float radius, int slices, int stacks) which draws a sphere using procedurally generated geometry and quad strip primitives. Again assume colour, transformation, polygon mode etc. settings are made outside of the function.

(b) Now write a function void drawSaturn(float theta, int slices, int stacks) to render Saturn as above, including performing the required transformations to tilt and rotate the planet and setting any other important state values. Use the function drawSphere from above, along with the following functions which are available:

void drawCircle(float radius, int slices)
void drawAnnulus(float radius1, float radius2, int slices, int stacks)
void drawAxes(float length)

Saturn’s radius is 60,000 km. The ring’s inner radius is 65,000 km and outer radius is 120,000 km.

(c) Give changes to drawSaturn to control rendering of the sphere (and rings) as wireframe or filled, depending on the value of options[OPT_WIREFRAME], where options is a global boolean array.

(d) Give changes to drawSphere to provide normal vectors, if options[OPT_LIGHTING] is set to true.

(e) Give changes to drawSphere to provide texture coordinates, if options[OPT_TEXTURE] is set to true.

(3+3+1+1+1 = 9 marks)

THE END