NATIONAL UNIVERSITY OF SINGAPORE
SCHOOL OF COMPUTING

EXAMINATION FOR
Semester 1, 2011/2012

CS3241 - COMPUTER GRAPHICS

Nov 2011 Time Allowed: 2 Hours

INSTRUCTION TO CANDIDATES

1. This is an OPEN book examination.
2. This examination paper contains **SIX (6)** questions and comprises **SEVEN(7)** printed pages.
3. Answer *ALL* questions within the spaces provided in this booklet.
4. You are allowed to use the back of the paper but please remember to state "P.T.O."
5. *Cross out any draft* or otherwise we will mark the poorer answers.
6. No calculator should be needed in this exam. You can leave your answer in surd form, namely, you can write $\sqrt{2}$ instead of 1.4142...
7. **In the programming questions, you can assume some basic normal vector arithmetic functions are provided for the 2D or 3D vector class `Vector2D (Vector3D)`, even a function `normalize(Vector2D)` to compute a normalized vector.**
8. Please write your matriculation number below, but NOT your name.

TIDINESS COUNTS!

We will deduct marks if your writing is too messy.

MATRICULATION NUMBER:_____

(this portion is for examiner's use only)

Question	Max. Marks	Score	Check
Q1	10		
Q2	8		
Q3	6		
Q4	13		
Q5	8		
Q6	5		
Total	50		

Question 1 [10 marks]

You are given 10 statements below. State if each statement is True (T) or False (F) by writing your answer in the box provided. A correct answer will give you 1 mark. An empty answer will give you zero, but ***a wrong answer will result in -1 mark***. (The lowest mark for this whole question is ZERO)

☐ We only need screen coordinates to perform Phong shading.

☐ Clipping by the viewing frustum is only for speeding up. That means, we can just perform the perspective projection without clipping by the viewing frustum and the render scene will look normal.

☐ In both 2D and 3D, if we want to perform several consecutive translations (and translations only), it can be consolidated and replaced by a single translation matrix.

☐ In both 2D and 3D, if we want to perform several consecutive rotations (and rotations only) , it can be consolidated and replaced by a single rotation matrix.

☐ In scanline conversion algorithm, if all the vertices are within 1 pixel, we just need to color that pixel for a correct and fast result. (This question is only about SCA, ignore any other factors such as z-buffer)

☐ If a floor is modeled by a square with four corners (a single polygon), we can never see the round shape of a spotlight even though the spotlight is shining on the floor in correct settings (e.g. direction and position, etc) by Gouraud shading.

☐ These type of Wall Street Journal pictures on the newspaper (an example on the right) are done by anti-aliasing.

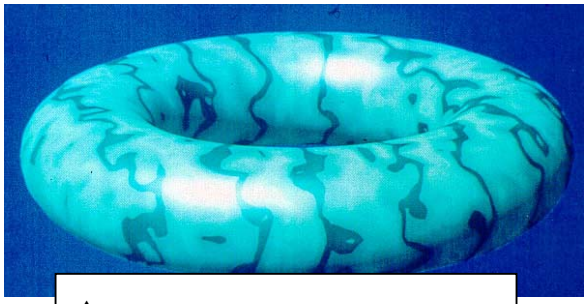
☐ When we see the reflections on objects in the real time graphics in our game consoles (e.g. Playstation 3, Xbox, etc), they are all done by real time ray-tracing.

☐ We can use texture mapping to change a horse into a zebra. (Of course, in computer graphics rendering, not in real life)

☐ If two curves are C1 continuous, they will always be C0 continuous.

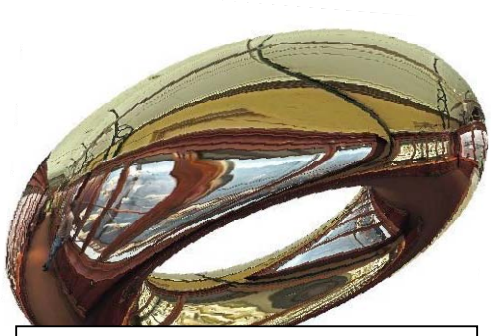
Question 2 [8 marks]

Describe the best techniques taught in our course to draw the following scenes. For each picture, give ONE most important and unique technique.

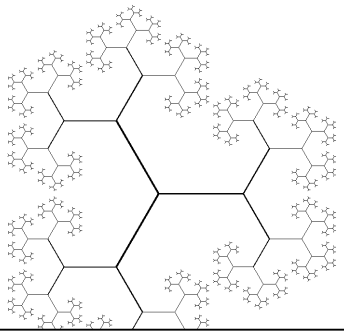


↑

↑



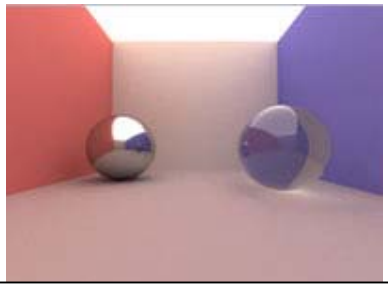
↑



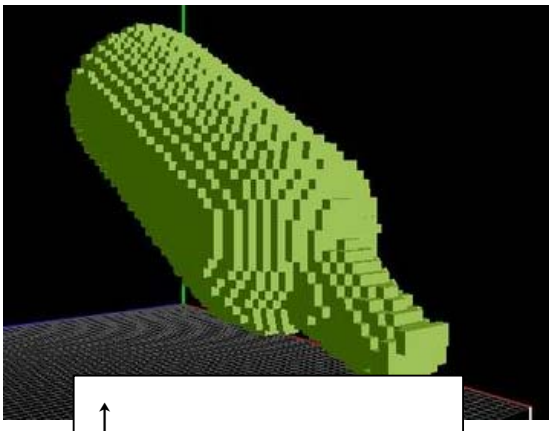
↑

↑

↑



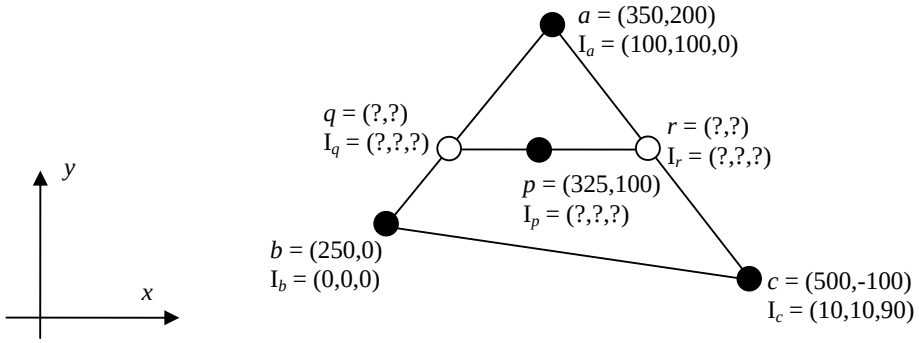
↑



↑

Question 3 [6 marks]

Given the vertices of a triangle *abc* in the screen coordinates and their light intensities I_a, I_b, I_c in RGB format as shown in the diagram below, by using Gouraud shading, calculate the light intensity I_p at the point *p*. Assume that the scan-line conversion is horizontal. (Hint: first calculate the intensities of the two white dots, namely *q* and *r* after finding the ratio of *aq* : *qb* and *ar* : *rc*)



Answer:

Question 4 [3+10 marks]

There is a viewer from the position (-2,-2,1), looking at the direction (1,1,0). Given three objects in the space in the form of paraboloids:

Paraboloid 1: $z = x^2 + y^2 + 2$
Paraboloid 2: $z = x^2 + y^2 - 2y + 1$
Paraboloid 3: $z = x^2 + y^2 - 2x - 2y + 3$

a) Let the ray equation be $r(t) = (x(t), y(t), z(t))$. State $x(t)$, $y(t)$ and $z(t)$.
Answer:

b) Compute the location of intersections of the ray with the three paraboloids and give the values of t if they intersect. You may leave the square root signs in your answer. Finally, which paraboloid does the ray hit first?
Answer:

Question 5 [8 marks]

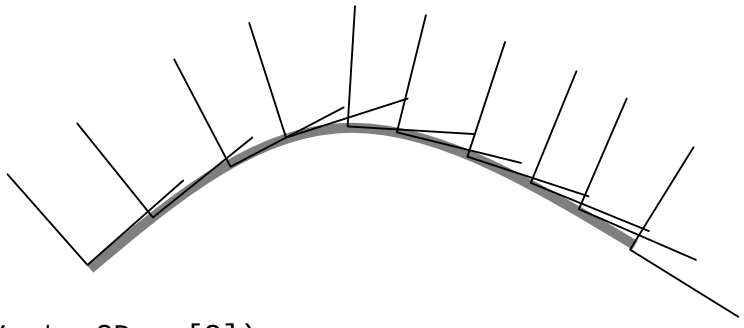
In this question, your code should be close to the real C++ language. For example, you cannot use the summation sign in your code. However, you can assume some basic library such as Vector2D with some arithmetic operations, and assume that OpenGL can take in the Vector2D class, e.g. `Vector2D cp;, glVertex2fv(cp).` You do not have to worry about the color and the line width, the example shown below is just for your understanding.

a) Write pseudo code to compute a point `p(t)` on a quadratic (degree 2) Bezier curve, with control points `cp[i]` (i=0 to 2) in the two dimensional space. (2 marks)

b) Write pseudo code to compute the velocity `p_dash(t,cp[3])` at *t*. (2 marks)

c) Write pseudo code to compute the normal `p_normal(t,cp[3])` at *t*. (Any normal, just need to be perpendicular to the velocity) (2 marks)

d) Write OpenGL code to draw the quadratic Bezier curve with 10 subdivisions as shown below. Also draw the "hairy" unit tangent and normal vectors (namely, the length is 1 unit). (2 marks)

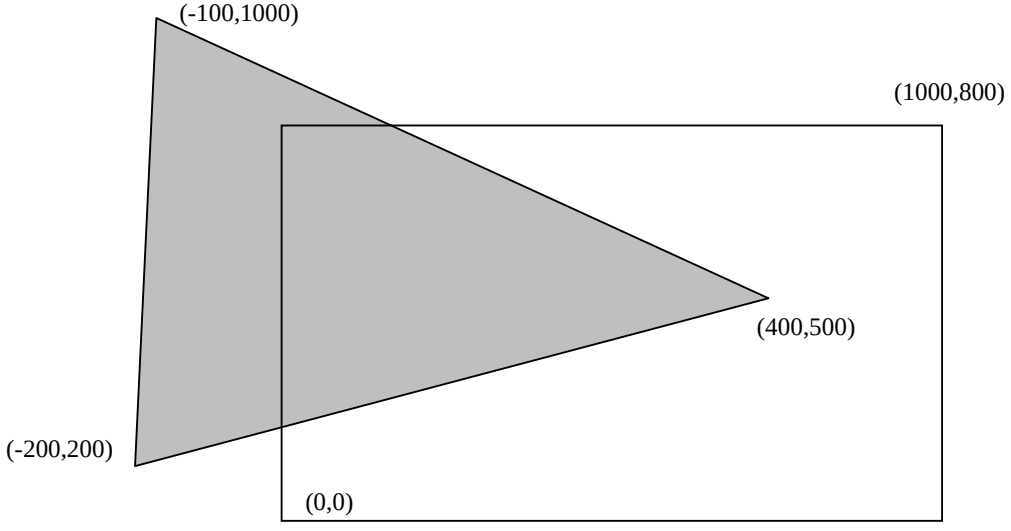


```
Vector2D p(t, Vector2D cp[3])
{




}

Vector2D p_dash(t, Vector2D cp[3])
{




}
```

```
Vector2D p_normal(t, Vector2D cp[3])
{









}

drawHairyBezierCurve(Vector2D cp[3])
{









}
```

Question 6 [5 marks]

Clip the following triangle according to the screen (1000x800). Fill the clipped polygon with black (or blue) color. Draw the new vertices in the diagram and write down their new coordinates.



--- END OF PAPER ---