UNIVERSITY OF LONDON

GOLDSMITHS COLLEGE

B. Sc. Examination 2006

CREATIVE COMPUTING

IS51012A (CC112)
CREATIVE COMPUTING 1

Duration: 3 hours

Date and time:

There are six questions in this paper. You should answer no more than FOUR questions. Full marks will be awarded for complete answers to a total of FOUR questions. Each question carries 25 marks. The marks for each part of a question are indicated at the end of the part in [.] brackets.

There are 100 marks available on this paper.

This is a practical exam, you should enter all answers that require code into a the *Processing* sketch named by question number, part and sub-part. For example, Q5_b_ii.pde for Question 5 part b sub-part ii. Save your answer to your EXAM SUBMISSION FOLDER. Save Early, Save Often. You are entirely responsible for making sure your answers are saved in the correct location.

Q1
a) Briefly explain what each of the following *Processing* commands does:

i) noLoop() **[2]**

ii) screen(a,b) **[2]**

iii) framerate(F) [**2]**

b) John Maeda's "Design By Numbers" (DBN) language was the precursor to *processing*. It uses different conventions for the Y-axis and colour scale than processing. The origin **O** is in the bottom left of the screen and white is 0 and black is 100. Write down an expression that calculates each of the following:

i) convert a DBN Y-coordinate (yD) to a processing Y-coordinate (yP); assume a DBN screen height of 100. **[2]**

ii) convert a DBN colour value cD to a processing colour value cY. **[2]**

c) Write a *processing* method for each of the following.

i) drawSquare(…){…}draw a filled square. This method should take three arguments **x, y** and **len.** There should be the x and y coordinate of the *centre of the square,* the length of the sides, and the method returns no value. **[3]**

ii) dashedLine (…){…} draw a dashed line. This method should take five arguments, **x1, y1, x2, y2** and **len,** the x and y coordinates of the two end points of the line and the length of the dashes. **[6]**

d) Write a complete processing program to make a chess board consisting of 8 x 8 squares. Each square should have length 64 pixels. **[6]**

Q2

a) Give a brief explanation for each of the following:

i. A biomorph **[1]**

ii. Genotype array **[1]**

iii. Single step selection **[1]**

iv. Cumulative selection **[1]**

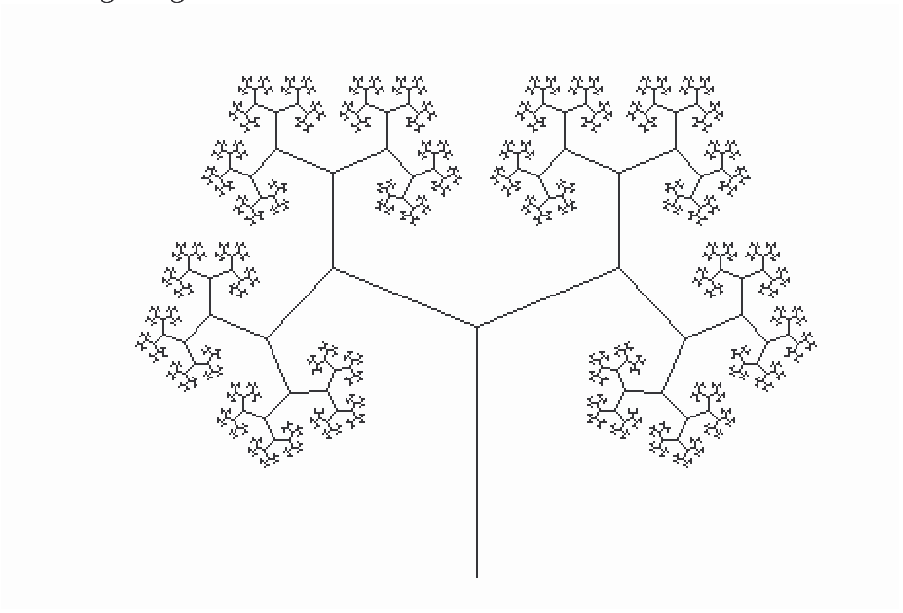
v. For an alphabet of 27 letters, write down a Processing expression for the probability of generating the 28-character sentence “IF MUSIC BE THE FOOD OF LOVE” using single-step selection. **[1]**

b) In this question you are requested to write a processing sketch to draw a biomorph from a genome.

i) Assuming a 2D canvas, write a method to draw the trunk and two branches, symmetric around the trunk, of a tree. Your tree method must be usable by the following draw method:

```
void draw(){
    float treeHeight=100;
    float angle= PI-PI/1.61;
    float scaleFactor=1.61;
    int nLevels=10;
    translate(width/2,3*height/4);
    tree(treeHeight,angle,scaleFactor,nLevels);
}
```

and the resulting image should look like this:



Your method should accept four arguments, **treeHeight,angle,scaleFactor,nLevels;** namely the height of the tree, the angle of the branches, the shrink factor for the length of branches between successive levels and the number of levels of branches.

```
    void tree(float height, float angle, float scaleFactor,
int nLevels){
     {
          line(0,0,0,-height);
          if(n==0)
               return;
          // YOUR CODE HERE [5]
```

ii) For the biomorph in part (i) declare a global genotype array with space reserved for a parent and 9 progeny, each with three genes. The genes will control the level of branching, the angle of branching and the rate of shrink/growth of the branches. **[5]**

iii) Write a method called **`reproduction()`** that makes copies of the parent genes in part (ii) but with a mutation rate of 0.333. You must choose a suitable mutator expression for each of the three genes. **[5]**

```
     void reproduction(){
          // your code here
     }
```

iv) Now use the your tree method in part (i) with the genome array of part (ii) to draw the parent biomorph and all nine progeny (with mutated gene values) on the screen. **[5]**

Q3.

a) For each of the following affine transformation provide a short description of the effect on a set of points in homogeneous coordinates.

   i. Identity Matrix **[2]**

   ii. Scaling Matrix with ScaleX=0.9 and ScaleY =0.75 **[2]**

   iii. Translation Matrix width Dx=100 and Dy=250 **[2]**

   iv. Rotation Matrix for a rotation by PI/6 radians. Around which point is the rotation centred? **[4]**

b) Write processing code to declare and define a 2D array representing the four points of the square with its top left point at (100,50) and edge length 200. **[4]**

c) Write a processing method called printMatrix that takes as an argument a 2D float array and prints it to the processing text area formatted as a matrix. **[5]**

```
void printMatrix(float[][] M){
// Your code here
}
```

d) Write a processing method called matrixMultiply that accepts a 2D point matrix and a 2D affine transformation matrix both in homogeneous coordinates represented by 2D arrays, and multiplies them returning the result as a new 2D array. **[6]**

```
float[][] R=matrixMultiply(float[][] P, float[][]M){
           // your code here
  }
```

Q4

a) Wolfram Cellular Automata

   i. How many unique 3-adjacent *pixel states* are possible in a Wolfram 3-bit cellular automata system in two colours (black and white). (Show your working). **[2]**

   ii. Make a truth table using 0s and 1s of the cell patterns and their corresponding output states for RULE 22 (the 3-bit Exclusive OR rule). **[3]**

```
 x-1 x  x+1    out
--------------------
0 0 0    0
```

   iii. How many patterns can be generated by a Wolfram 3-adjacent pixel cellular automata in two colours (0,1). (Show your working). **[2]**

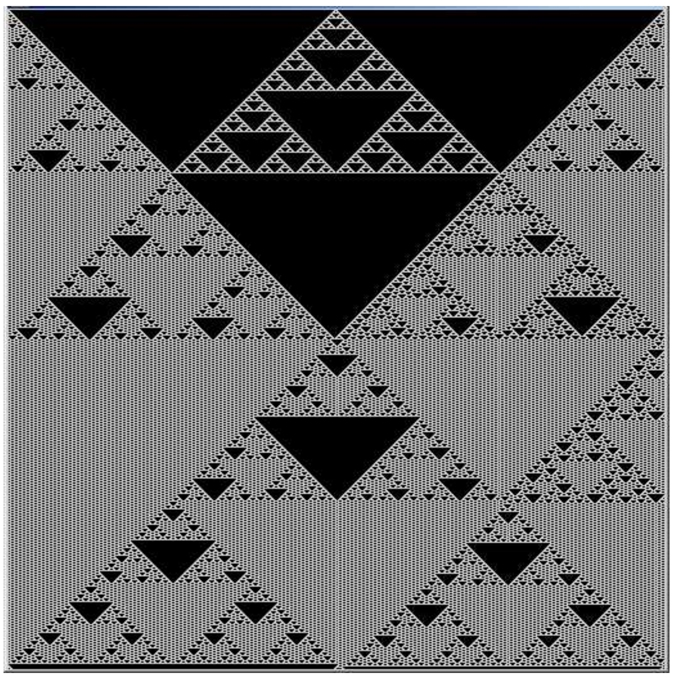
   iv. Complete the following processing sketch to generate a pattern according to RULE 22 assuming a 3-bit Wolfram Cellular Automata: **[5]**

```
int[] state;
int step=0;
void setup(){
 size(512,512);
  state = new int[512];
  state[256]=1;
}

void draw(){
  int[] tmp= new int[state.length];
  // Your code here
}
```

b) Suppose you are given an integer called RULE with a value between 0 and 255. Write a processing **`draw()`** method that will generate the Cellular Automata for the given RULE. You can use either bit-wise operators or conditionals in your answer. **[6]**

c) Write a processing sketch using a suitable initial state array and RULE number to generate a pattern as similar as possible to the following using a screen size of 512 x 512 pixels. (Pixel colour convention: black=0 and white=1): **[7]**



Q5.

a) Briefly explain each of the following processing commands; include in your answer an explanation of the parameters for method calls:

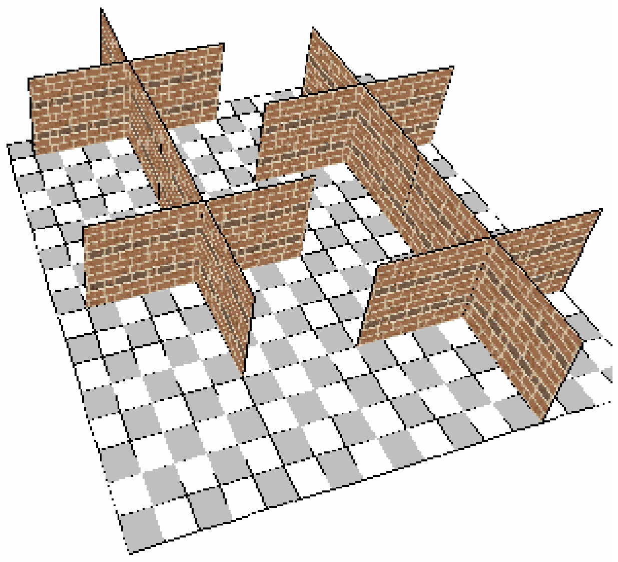
   i. P3D **[2]**

   ii. popMatrix(); **[2]**

   iii. vertex(x,y,u,v); **[2]**

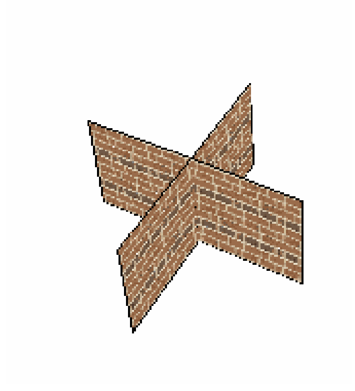
b) Write a processing sketch to draw a rotating earth sphere (radius 100 pixels) and a geo-stationary moon sphere (radius 10 pixels) with distance between the centres of the spheres of 200 pixels. (Geo-stationary means that the moon rotates about the earth at the same speed as the earth). The Earth/Moon system should be centred in the screen. The earth should rotate on its own axis once per minute at a frame rate of at least 10 frames per second. **[5]**

c) Consider the following 3D view of a maze: the checker-board consists of 64 x 64 pixel squares. The screen size is 512 x 512.



   i. Write a processing sketch to draw a 2D plan view of the maze (i.e. the view of the maze from above). **[5]**

ii. The file bricks.gif is provided. Write a processing sketch to draw four texture-mapped planes joined at the centre offset by an angle of PI/2, to make one section of wall of the maze. **[4]**



iii. Write a new processing sketch to draw the 3D maze pictured above. (You may ommit the checkered ground plane with no loss of credit). **[5]**

Q6.

a) Briefly describe each of the following *first person* motion controls, give an example of how to implement the motion control in processing.

   i. strafe left **[2]**

   ii. move forward **[2]**

   iii. look up **[2]**

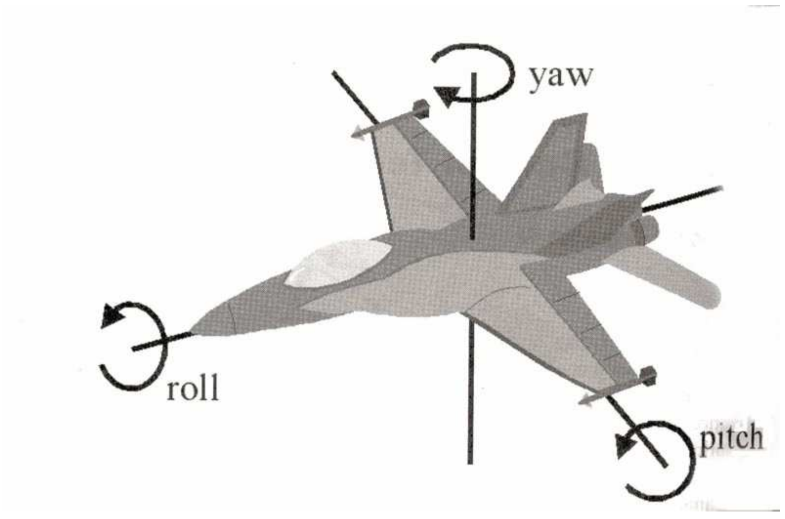
b) Make a processing sketch to implement each of the following first-person motions in a 3D world. Assume that a one-button mouse is available.

   i. Rotate left/right **[3]**

   ii. Rotate up/down **[3]**

   iii. Move forward **[3]**

c) A flight simulation program allows the following motions in addition to thrust and reverse thrust:



the following code is provided, it is a 3D world consisting of a pilot view (as if from a plane) and a runway. The plane is in forward motion.

```
import processing.opengl.*;
float velocity=2;
float z=1000;
void setup(){
size(512,512,OPENGL);
stroke(0);
}


void draw(){
  background(255);
  noFill();
  // DRAW PILOT's VIEW FROM PLANE
  beginCamera();
  camera(0,0,0,0,0,-1000,0,1,0);
  translate(width/2,height-200,z);
  endCamera();

  // DRAW RUNWAY
    pushMatrix();
      translate(width/2,height,0);
      rotateX(-PI/2);
       stroke(0);

      // RUNWAY EDGE
      rect(-100,0,200,10000);

      // RUNWAY GUIDE LINES
      for(int k=0;k<10000;k+=50){
        if(k%100==0)
          rect(-1,k,2,50);
       }
    popMatrix();

  // MOVE PLANE FORWARD
  z-=velocity;
}
```

i. Modify the given code to make the plane *roll* in response to a suitable mouse command **[5]**

ii. Modify the given code to make the plane pitch in response to a suitable mouse command **[5]**