

THIS PAPER IS NOT TO BE REMOVED FROM THE EXAMINATION HALLS

UNIVERSITY OF LONDON

291 0112 ZB

BSc/Diploma Examination
for External Students

CREATIVE COMPUTING

Creative Computing 1: image, sound, motion

Dateline: Friday 15 May 2009 : 10.00 – 1.00 pm

Duration: 3 hours

There are six questions in this paper. Candidates should answer **FOUR** questions. All questions carry equal marks and full marks can be obtained for complete answers to **FOUR** questions.

Questions involving a description or explanation should, wherever possible, be accompanied by an appropriate example.

A hand held calculator may be used when answering questions on this paper but it must not be pre-programmed or able to display graphics, texts or algebraic equations. The make and type of machine must be stated clearly on the front cover of the answer book.

© University of London 2009

UL09/884

PAGE 1 of 10

QUESTION 1 Overview

- (a) (i) What is a *pixel*? For a PNG image in *Processing*, what information is stored for each pixel?

[3 Marks]

- (ii) Given an image in a PNG format, how might you use *Processing* to find out how many pixels there are in that image? Write the fragment of code that you would use to do so.

[3 Marks]

- (iii) For a JPG image in *Processing*, what information is stored for each pixel?

[2 Marks]

- (iv) What is pixellation, and how does it occur? What approaches can be used to lessen the effects of pixellation in an image? Discuss briefly whether or not pixellation is ever a desirable property in an image

[4 Marks]

- (b) (i) In *Processing*, there are three methods for setting the properties of lines: `StrokeWeight()`, `strokeJoin()`, and `strokeCap()`. What effect does each have? Illustrate your answer with diagrammatic examples

[3 Marks]

- (ii) How might you add a new `MODE` to `strokeCap()`? Suggest a useful `MODE` and discuss how that might be implemented

[5 Marks]

- (c) What is recursion? Give an example of a short piece of *Processing* code that contains recursion. Would it be possible to rewrite any piece of recursive code without using recursion? Justify your answer.

[5 Marks]

QUESTION 2 Shape and colour

- (a) (i) What three basic geometrical forms were employed in Itten's Basic Course at the Bauhaus? Briefly explain Itten's rationale for their use.

[4 Marks]

- (ii) Briefly outline the three *Processing* functions that can draw these shapes. Where the *Processing* functions are more general than the corresponding Bauhaus basic shape, state what restrictions must be applied to the parameters so that the Bauhaus basic shape is produced. (E.g. if rhombus was a Bauhaus basic shape (it is not) and *Processing* had a parallelogram function (it does not) then parallelogram vertices would have to be set so the sides were all the same length)

[5 Marks]

- (iii) Describe an artefact designed at the Bauhaus that employs some or all of the basic forms.

[2 Marks]

- (b) Explain the operation of transparency in *Processing*

[3 Marks]

- (c) Write a *Processing* program to set up a 500x500 sketch with a white background and on it draw four circles, each of radius 150, one of each centred on the points (350,350), (150,350), (350,150), (150, 150). Colour one each of the circles red, green, blue and black respectively, each with transparency 50%.

[5 Marks]

- (d) For each of the pixel positions (i), (ii), (iii) below, calculate the (r,g,b) colour value (each value in the range 0...255) and give a name for the colour.

(i) (300, 300)

(ii) (260, 260)

(iii) (250, 250)

[6 Marks]

QUESTION 3 Transformations

- (a) When working with matrix representations of coordinate transformations, a general point (x, y) can be represented in column vector form as

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

Why is it convenient to use a 3-vector rather than a 2-vector?

[1 Marks]

- (b) Derive or directly write down the matrix representations:

(i) \mathbf{R} for rotating a point $\underline{\mathbf{p}}$ by an angle θ clockwise about the origin.

[3 Marks]

(ii) \mathbf{T} for translating a point $\underline{\mathbf{p}}$ by TX and TY in the x - and y -directions respectively.

[2 Marks]

- (c) Given $\theta = \pi/3$, $TX = 3$, $TY = 1$, and

$$\underline{\mathbf{p}} = \begin{pmatrix} 1 \\ 3 \\ 1 \end{pmatrix}$$

find the point $\underline{\mathbf{p}'}$ in each of the cases:

(i) $\underline{\mathbf{p}'} = \mathbf{R} \cdot \mathbf{T} \cdot \underline{\mathbf{p}}$

[3 Marks]

(ii) $\underline{\mathbf{p}'} = \mathbf{T} \cdot \mathbf{R} \cdot \underline{\mathbf{p}}$

[3 Marks]

Give answers in simplified surd form, or, if you use a calculator, give numerical answers to 3 decimal places. For full credit in either case show all stages of your working.

- (d) Explain under what conditions for θ , TX and TY does rotation followed by translation have the same effect as translation followed by rotation? Explain your answer.

[3 Marks]

(question continues on next page)

- (e) Consider the following fragment of *Processing* code, where the output screen dimensions are 512 x 512.

```
import processing.opengl.*;
size(800,600,OPENGL);
noFill();
background(127);
stroke(255,255,0);
triangle(200,200,400,500,0,500);
rotate(PI/2);
scale(2.0);
stroke(0,255,255);
ellipse(200,-100,75,150);
println(width + ' ' + height);
```

- (i) State the colours used for drawing each shape.

[2 Marks]

- (ii) Sketch a boundary line for the output screen and within it sketch the objects drawn by *Processing*. Exact positions are not needed, just the relative positions, sizes and appearances.

[4 Marks]

- (iii) If the line

```
triangle(200,200,400,500,0,500);
```

is replaced by

```
rotate(-PI/4);
triangle(200,200,400,500,0,500);
rotate(-PI/4);
```

sketch the new appearance of the output screen.

[4 Marks]

QUESTION 4 3D drawing and movement

(a) What is OpenGL, and how is it used in *Processing*?

[6 Marks]

(b) Consider the code on the next page:

(i) There are minor syntactical errors in the code. Say what they are and what the correct code would be

[2 Marks]

(ii) Describe what the code is doing. You should describe its overall functionality, rather than what each line of code does

[5 Marks]

(iii) Modify the code so that when the left mouse button is pressed, the image stops moving, and when the right mouse button is pressed, the image moves again. You don't need to write down the whole program; only the lines you would add; where they would go; and any lines you might remove

[7 Marks]

(iv) What would the effect be if you were to remove the code

```
popMatrix();  
pushMatrix();
```

at the place marked with // code here?

[5 Marks]

(question continues on next page)

```

import processing.opengl *;
float theta=0;
float dItheta=TWO_PI/360;
float radius=200;
// Lights, Camera, Action
void setup(){
  size(800,600,OPENGL);
  noStroke;
  camera(width/2,height/2,width,width/2,height/2,0,0,-1,0)
  lights();
  fill(255);
}
void draw(){
  background(255);
  float x,z;
  x=radius*cos(theta);
  z=radius*sin(theta);
  directionalLight(255, 255, 255, 0, 0, -1);
  pushMatrix();
  translate(width/2+x,height/2,z);
  sphere(50);
// code here
  popMatrix();
  pushMatrix();
  translate(width/2-x,height/2,-z);
  sphere(50);
  popMatrix();
  pushMatrix();
  x=radius*cos(theta+PI/2);
  z=radius*sin(theta+PI/2);
  translate(width/2,height/2+x,-z);
  sphere(50);
  popMatrix();
  theta=(theta+dItheta)%TWO_PI;
}

```

QUESTION 5 Data and Image

- (a) What is compression and why is it important in the context of sound and image? What is the difference between lossy and lossless compression? Give examples of formats of each; you may use audio or picture as the basis. When is it appropriate to use each kind of compression?

[6 Marks]

- (b) Explain how *Processing* gets information into and out of programs or code. Your answer should describe in overview the different ways of getting or reading data (such as keyboard, file, mouse, etc) and presenting or writing data (such as to file, screen, etc.). Discuss any limitations and/or benefits, especially with comparison to other programming languages or systems that you have used.

[6 Marks]

- (c) The `PImage` class in *Processing* has a number of methods that are available. Describe two of them, and explain what they do and how they are useful in processing data that is in the form of images.

[4 Marks]

- (d) Write a *Processing* sketch that will take two images that are stored in two files, `backg.jpg` and `front.png`, and makes a new image that has superimposed one onto the other. Draw pictures of your two images (they can be whatever you choose, and the drawings need only be representational) and then show what the new image will look like. Explain very briefly what approach to superimposing you have taken.

[6 Marks]

- (e) Discuss ways in which the above approach could be used to create a visual piece that has artistic impact, and describe what form of impact would be achieved.

[3 Marks]

QUESTION 6 Structure and motion

- (a) What is the Gestalt principle of closure? Give examples of two drawings that contain closure in them. Say what aspect gives the effect in each.

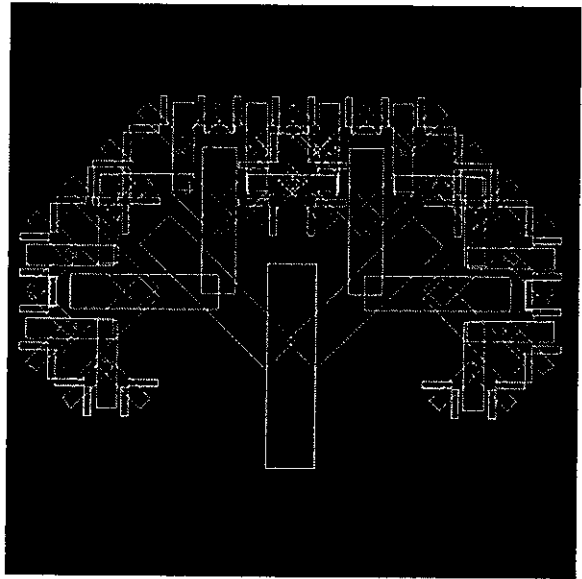
[4 Marks]

- (b) The following code produces the image alongside it.

- (i) What would happen if you changed the first line to be `int Branching = 3;`? Draw the output.
- (ii) What happens if you remove the second method call to `tree()` in the `tree()` method? What happens if you remove both calls to `tree()`?
- (iii) How could you modify the code to produce a tree that branches into 3 branches at each point, rather than 2? Write the code to do that

[8 Marks]

```
int Branching=7;
void setup(){
  size(512,512);
  background(0);
  stroke(255);
  strokeWeight(1);
  rectMode(CENTER);
  noFill();
}
void draw(){
  translate(width/2.,7*height/8.);
  tree(height/4,PI/4,1.41,Branching);
}
// make a tree by recursion
void tree(float sz, float a, float sf, int n){
  if(n==0)
    return; // terminate if n is zero
  rect(0,-sz,n*6,n*26);
  --n; // decrement branching
  pushMatrix(); // save coordinate system
  translate(0,-sz); // move to end of branch
  rotate(a); // rotate by angle
  tree(sz/sf,a,sf,n); // draw a tree
  rotate(-2*a); // rotate other way
  tree(sz/sf,a,sf,n); // draw a tree
  popMatrix(); // restore coordinates
}
```



- (c) Describe two ways in which *Processing* can be used to implement the concept of movement. Discuss each of these in detail, and include a comparison with other ways in your discussion.

[6 Marks]

(question continues on next page)

(d) Consider the *Processing* sketch below, that simulates a paintbrush in 3D, and uses it to create the appearance of a paint textured image on a 2D surface

(i) Add code that will cause the continuous painting to stop, when a key or mousebutton is pressed

(ii) What role is the following code playing?

```
if(random(100)<3){
  dx+=random(1)-0.5;
  dy+=random(1)-0.5;
```

What would happen if the `random()` function was not used?

(iii) Discuss briefly the work of Jackson Pollock in the context of the use of motion to create artworks.

[7 Marks]

```
float x,y,dx,dy;
float s,ds;
int SZ=512;
void setup(){
  size(SZ,SZ,P3D);
  x=SZ/2; y=SZ/2;
  s=1; ds=.002; dx=1; dy=1;
  colorMode(HSB,TWO_PI,1,1);
  background(0,0,0);
  noStroke();
  rectMode(CENTER);
}
void draw(){
  //background(0,0,0);
  fill((x+y)/100,%TWO_PI,1,1);
  translate(x,y);
  scale(s,s);
  for(int k=0;k<20;k++){
    rotate((x+y)/100.);
    rotateX(x/100.);
    rotateY(y/100.);
    rect(0,0,100,50);
  }
  s=s+ds;
  x=x+dx;
  y=y+dy;
  if( s > 2 || s < 0.75 ) ds=-ds;
  if( x <= 0 || x >= width-1 ) dx=-dx;
  if( y <= 0 || y >= height-1 ) dy=-dy;
  if(random(100)<3){
    dx+=random(1)-0.5;
    dy+=random(1)-0.5;
  }
}
```