

THIS PAPER IS NOT TO BE REMOVED FROM THE EXAMINATION HALLS

UNIVERSITY OF LONDON

291 0112 ZA

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

UL09/883

PAGE 1 of 10

QUESTION 1 Overview

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

[3 Marks]

- (ii) Given an image in a JPEG 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) 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*, what is the difference between `quad()` and `rect()`?
(ii) Describe how you could obtain the effect of `rect()` using only `quad()`. Illustrate with an example.

[5 Marks]

- (c) (i) The unit of measure used for audio sample rates is the decibel (dB). For CD quality audio, which actual sample rate is most commonly used?
(ii) What is the impact if a higher sample rate is used?
(iii) What sample rate (you may give a range) would be acceptable for telephony?

[5 Marks]

- (d) 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 (150,150), (350,150), (150,350), (350, 350). 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) (250, 250)

(ii) (200, 200)

(iii) (240, 240)

[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 $\theta = \pi/6$, $TX = 2$, $TY = 1$, and

$$\underline{\mathbf{p}} = \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix}$$

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 *Processing* code that utilises the *Sonia* library to sound a note varying continually between an upper A and lower A while plotting sample positions of the associated sine curve.

```

import pitaru.sonia_v2_9 *;
int bufSize = 2048;
float SR = 44100;
float oneHertz = IWO_PI/SR;
float freqHigh = 880*oneHertz;
float freqLow = 220*oneHertz;
float amp = 0.5;
float freqSet= freqHigh;
int IncOrDec=-10;
void setup(){
  size(512,512);
  Sonia.start(this,44100);
  LiveOutput.start(bufSize, bufSize*2);
  LiveOutput.startStream();
  stroke(255);
  strokeWeight(4);
  frameRate(20);
}
void draw(){
  background(0);
  for(int n = 0; n < min(width, bufSize); n++){
    point(n, height*(0.5 - 0.5 * LiveOutput.data[n]));
  }
  //E - point at which to insert code for solution to (e)(ii)
  freqSet=freqSet+IncOrDec*oneHertz;
  if(freqSet < freqLow){IncOrDec = -IncOrDec;}//Reverse frequency change direction
  if(freqSet > freqHigh){IncOrDec = -IncOrDec;} //when go outside bound
}
void liveOutputEvent(){
  for (int n = 0; n < bufSize; n++){
    LiveOutput.data[n] = amp * sin (freqSet * n );
  }
}
}

```

- (i) Explain why the sound would have intermittent high-pitched pips interspersed with the note, instead of being a smooth rendering of the current note.

[3 Marks]

- (ii) Modify the program, by inserting code at the point E indicated, to enable mouse control of `IncOrDec` and `amp` as follows, to operate when the mouse button is pressed:

`IncOrDec` varies from 0 when the mouse is at the left edge of the window to 20 when at the right edge of the window

`amp` varies from 0 when the mouse is at the bottom edge of the window to 0.5 when at the top of the window.

Note: there is no need to write out the whole program and so only the added code need be written down in your answer book

[7 Marks]

QUESTION 4 3D Graphics

(a) What is texture mapping? Explain its importance in graphics.

[4 Marks]

(b) What two main functions does *Processing* make available for use in creating textured surfaces? Describe briefly how they are used.

[6 Marks]

(c) Consider the code below, which implements texture mapping. The figure which follows shows the output when the sketch is executed

(i) Show how you would modify the code so that it texture maps something other than the photograph. Be clear about exactly what changes you would make.

[2 Marks]

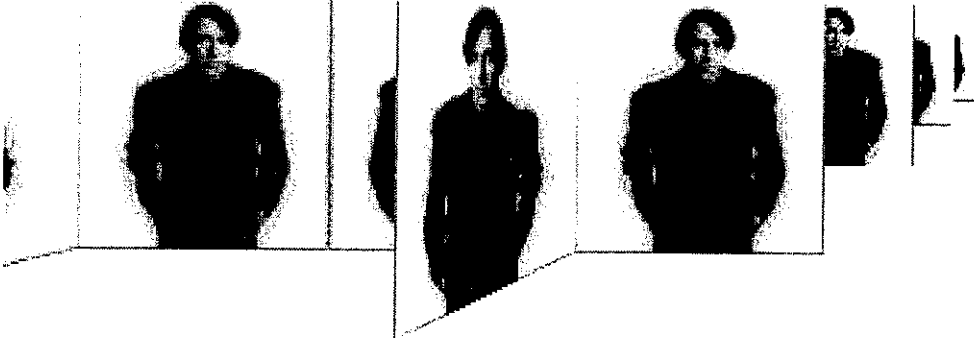
(ii) How could you modify the code to provide something of more visual impact than the original? Be clear about the code modifications, and describe in detail the basis of the impact you are trying to achieve. You should illustrate your answer with code fragments where appropriate

[13 Marks]

```
PImage wallpaper;
float a;
void setup(){
  size(800,400,P3D);
  wallpaper=loadImage("cale.jpg");
  noStroke();
  noLoop();
}
void draw(){
  background(255);
  translate(-width,height/2,-width/2);
  width=width/2;
  maze();translate(width,0,0);
  maze();translate(width,0,0);
  maze();translate(0,0,-width);
  maze();translate(0,0,-width);
  maze();translate(0,0,-width);
  maze();translate(-width,0,0);
  maze();translate(-width,0,0);
  maze();
}
void maze(){
  wall(width/2);rotateY(-PI/2);
  wall(width/2);rotateY(-PI/2);
  wall(width/2);rotateY(-PI/2);
  wall(width/2);rotateY(-PI/2);
}
void wall(int l){
```

(question continues on next page)

```
beginShape();  
texture(wallpaper);  
vertex(0,0,0,0);  
vertex(L,0,L,0);  
vertex(L,L,L,L);  
vertex(0,L,0,L);  
endShape();  
}
```



QUESTION 5 Data and Image

- (a) 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]

- (b) 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]

- (c) Visual art can be used to make social comment. The question that follows examines this.

- (i) Give an example of an artwork (you may describe one that already exists, or one that you might create) that comments on an ethical issue, where a view of right or wrong is being presented. Describe briefly the artwork (you may sketch it if appropriate) and how it is making the comment

[4 Marks]

- (ii) 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 one contained within the other. Draw pictures of your two images (the drawings need only be representational) and then show what the new image will look like

[5 Marks]

- (iii) Discuss ways in which the above approach—of containment, either where one image contains another, or where an image contains a copy of itself, or multiple copies of itself—could be used to create a visual piece that has artistic impact, where it makes a comment about an ethical judgement

[6 Marks]

QUESTION 6 Structure and motion

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

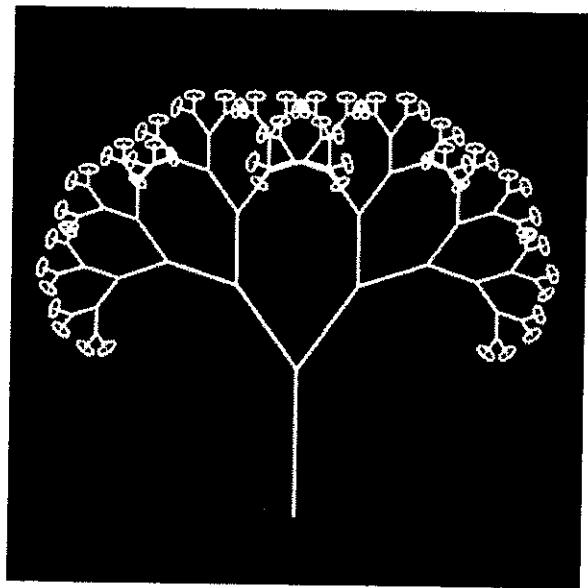
[4 Marks]

- (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(4);
  noFill();
}
void draw(){
  translate(width/2.,7*height/8.);
  tree(height/4,PI/5,1.41,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
  line(0,0,0,-sz); // draw a line
  if(n==1) // if end of tree draw a leaf
    ellipse(0,-sz,16,8);
  --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;
  }
}
```