THIS PAPER IS NOT TO BE REMOVED FROM THE EXAMINATION HALLS

UNIVERSITY OF LONDON **CO1112 ZA**

BSc/Diploma Examination

CREATIVE COMPUTING AND COMBINED DEGREE SCHEME

Creative Computing 1: image, sound and motion

Date and Time: Wednesday 18 May 2016 : 14.30–17.30

Duration: 3 hours

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

Only your first **FOUR** answers, in the order that they appear in your answer book, will be marked.

Each question is worth 25 marks.

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, text or algebraic equations. The make and type of machine must be stated clearly on the front cover of the answer book.

Question 1 General — multiple choice, true and false, and completion of information

(a) Multiple choice

For each question, write down the letter corresponding to the correct answer. If you need to do working, do so on the page and then cross it out once you have written down the letter of the answer you have chosen.

I Conway's game of life utilises a general 2-state, 2-dimensional cellular automaton. For such a general automaton, how many immediate neighbours are there for any cell?

A. 2
B. 4
C. 8
D. 9
E. 16
F. none of the above

[2]

II For a general 2-state, 2-dimensional cellular automaton, how many possible patterns are there for any cell and its immediate neighbours?

A. 8
B. 9
C. 16
D. 256
E. 512
F. none of the above

[3]

III For a general 2-state, 2-dimensional cellular automaton, how many possible rules are there?

A. 3
B. 8
C. 9
D. 2^8
E. 512
F. none of the above

[3]

IV Consider the following cellular automaton, where the x's are live cells and the o's are dead cells, in the context of Conway's Game of Life:

```
o o o o o
o o x x o
o x o o x
o x o o x
o o x x o
```

Conway's Game of Life can be expressed as follows:

```
For a live cell:
    Each cell with one or no neighbors dies, as if by solitude.
    Each cell with four or more neighbors dies, as if by
          overpopulation.
    Each cell with two or three neighbors survives.
A dead cell with three live neighbours becomes live.
```

Which of the following shows the next (first) iteration of the automaton?

A.
```
o o o o o
o o x x o
o o x o x
o o x o x
o o x x o
```

B.
```
o o o o o
o x x x x
o x o o x
o x o o x
o x x x x
```

C.
```
o o x x o
o x o o x
o x o o x
o o x x o
o o x x o
```

D.
```
o o o o o
o o x x o
o x o o x
o x o o x
o o x x o
```
[4]

V For the same original automaton, which of the following shows the third iteration (i.e two further iterations after the one above)?

A.
```
o o o o o
o x x x x
o x x x x
o x x x x
o x x x x
```

B.
```
o o o o o
o o x x o
o x o o x
o x o o x
o o x x o
```

C.
```
o o o o o
o o o x o
o o o x x
o o o x x
o o o x o
```

D.
```
o x o o x
o o x x o
o x o o x
o x o o x
o o x x o
```

E.

```
o o o o o
o x x o o
x o o x x
x o o x x
o x x o o
```

[4]

VI For the multiplication of the two matrices shown below, which is the correct answer?

$$\begin{pmatrix} 4 & 3 & 3 \\ 14 & 2 & 5 \\ 6 & 5 & 7 \end{pmatrix} \cdot \begin{pmatrix} 8 \\ 4 \\ 2 \end{pmatrix}$$

[4]

A. $\begin{pmatrix} 32 \\ 12 \\ 6 \end{pmatrix}$

B. $\begin{pmatrix} 32 & 12 & 6 \\ 112 & 8 & 10 \\ 48 & 20 & 14 \end{pmatrix}$

C. $\begin{pmatrix} 32 & 12 & 6 \end{pmatrix}$

D. $\begin{pmatrix} 50 \\ 130 \\ 82 \end{pmatrix}$

E. $\begin{pmatrix} 84 \\ 120 \\ 72 \end{pmatrix}$

(b) For each of the following sentences, listed below, say whether it is True or False. You need to explicitly do this for each sentence. You should list the sentence number, and then either "True" or "False".

I The Roman numeral system does not have a number for zero.

II The artist Uccello invented the perspective machine, which helped in three-dimensional drawing.

III Xenakis used a model of the behaviour of molecules within gases as the basis of a musical composition.

IV John Cage and Merce Cunningham collaborated using algorithmic approaches to combine poetry and painting.

V The focus on Constructivism meant that there was no place for fine art within the Bauhaus philosophy or teaching.

[5]

Question 2 Creative thinking and computation

(a) Give 3 ways in which a computer, with appropriate software and peripherals, can aid creativity in visual arts. For each of the 3 ways that you list, describe what software and peripherals — if any — might be needed, and for each way, describe very clearly the way in which creativity is aided. Also discuss any limitations in any of these. [10]

(b) Discuss the potential use of computers to aid creativity in the performing arts, such as dance and theatre. [4]

(c) i. What is an algorithm? [2]

ii. Discuss the role of algorithms in creative thinking. [4]

(d) Belgian Surrealist painter Magritte created a piece called "*The treachery of images*". Briefly describe this piece, and discuss the cultural comment that Magritte was trying to make with it. [5]

Question 3 Data, shape and structure

(a) List and describe 2 of the Gestalt laws that relate to how images are perceived. [4]

(b) *Processing* provides the ability to perform both recursion and repetition in code.

i. What is the difference between repetition and recursion? [2]

ii. Consider the following simple *Processing* sketch:

```
1
2     void setup() {
3       size(640, 360);
4       noStroke();
5       noLoop();
6     }
7
8    void draw() {
9       drawCircle(width/2, 280, 3);
10    }
11
12    void drawCircle(int x, int radius, int level) {
13      float tt = 126 * level/4.0;
14      fill(tt);
15      ellipse(x, height/2, radius*2, radius*2);
16      if(level > 1) {
17        level = level - 1;
18        drawCircle(x, radius/2, level);
19    }
20  }
```

Explain briefly what the sketch will look like, either using a diagram or giving a description. [2]

iii. Which part or parts of the code are recursive? [1]

iv. Would it be possible to write the code to produce the same picture, without any use of recursion? Explain your answer. [3]

(c) What is data compression? [1]

(d) Explain why compression is important in the context of digital image. [3]

(e) What is the difference between sampling and compression, in digital audio? In both of these activities, data can be lost. Describe for each, the ways in which this data loss happens, and the implications and tradeoffs of the loss. [9]

Question 4 Motion and interaction

Study the *Processing* sketch below, then answer the questions that follow.

```
1  int x = 0;
2  int y = 0;
3  int d = 30;
4  int dx = 2;
5  int dy = 2;
6  int blocky = 300;
7
8  void setup() {
9    size(600, 600);
10   ????
11 }
12
13 void draw() {
14   background(50, 50, 200);
15
16   // draw ball
17   fill(250, 250, 250);
18   ellipse(x, y, d, d);
19
20   // draw block
21   fill(250, 250, 0);
22   rect(400, blocky, 20, 100);
23
24   if (keyPressed) {
25     if (key == 'a' && blocky > 0) {
26       blocky -= 5;
27     }
28     if (key == 'z' && blocky < 500) {
29       blocky += 5;
30     }
31   }
32
33   x += dx;
34   y += dy;
35 }
```

(a) What line of code should be used at line 10 to set the frame rate of the sketch to 60fps? [2]

(b) Explain the effect of line 14, in the context of this sketch. [3]

(c) Explain the effect of the test "`blocky < 500`" on line 28. Why is the number 500 used here? [2]

(d) In addition to the Boolean variable `keyPressed`, *Processing* also provides another way to detect key presses. What is it, and how is it used? [3]

(e) Give a general description of the overall behaviour of the given sketch. [3]

(f) Improve the sketch by writing code to make the ball bounce off the edges of the screen if it hits any of them. Include comments in your code where appropriate. You do not need to write out the whole new sketch: just give your new code and say where it should be inserted into the existing code. [6]

(g) Improve the sketch by writing code to make the ball bounce off the block if they collide. Include comments in your code where appropriate. You do not need to write out the whole new sketch: just give your new code and say where it should be inserted into the existing code. [6]

Question 5 Generative systems

The *Processing* code shown below implements a simple L-System. Study the code and then answer the questions that follow.

```
1  float  d = 300;
2  float  ang = -HALF_PI;
3  String state = "F-F-F-F";
4  String F_rule = "F+F-F-F+F";
5  int    L = 1;
6
7  void setup() {
8    size(600, 600);
9    background(255);
10   stroke(0);
11   for (int k=0; k<L; k++) {
12     state = substitute(state, F_rule);
13     d /= 3;
14   }
15   noLoop();
16 }
17
18 void draw() {
19   translate(150, 150);
20   rotate(HALF_PI);
21   for (int i=0; i < state.length(); i++)
22     turtle(state.charAt(i));
23 }
24
25 void turtle(char c) {
26   switch(c) {
27   case 'F':
28     line(0, 0, d, 0);
29     translate(d, 0);
30     break;
31   case '-':
32     rotate(ang);
33     break;
34   case '+':
35     rotate(-ang);
36     break;
37   }
38 }
39
40 String substitute(String s, String F) {
41   String s2 = new String();
42   for (int j=0; j < s.length(); j++) {
43     if(s.charAt(j)=='F')
44       s2 = s2 + F;
45     else
```

```
46          s2 = s2 + s.charAt(j);
47      }
48      return s2;
49  }
```

(a) When describing substitution systems such as this L-System, the terms *initiator* and *generator* are often used. What initiator(s) and generator(s) are used in this code? [2]

(b) What is the turtle interpretation of the `F` symbol in this code? [3]

(c) What is the content of the `state` variable after one call to the `substitute()` method in line 12? [2]

(d) In the context of this sketch, what is the purpose of line 13 ("d /= 3")? [4]

(e) Draw the output of this sketch. Label your drawing with the co-ordinates of both ends of the left-most line drawn. [5]

(f) The *Blind Watchmaker* algorithm described in the Subject Guide is an example of a *genetic algorithm*.

i. What is a genetic algorithm? Explain its important features. [5]

ii. The *Blind Watchmaker* is an example of a *user-guided* genetic algorithm. What is the difference between a user-guided genetic algorithm and one which uses a fitness function? [2]

iii. Give one advantage, and one disadvantage, of a user-guided genetic algorithm over one that uses a fitness function. [2]

Question 6 3D graphics and effects

Study the *Processing* code shown below and then answer the questions that follow.

```
1  void setup() {
2    size(600, 600, P3D);
3    camera(0, 0, -800, 0, 0, 0, 0, 1, 0);
4    noLoop();
5  }
6
7  void draw() {
8    int n, c;
9    fill(255);
10   box(100);
11   for (int i=0; i<4; i++) {
12     if (i%2 == 0) {
13       n = 2; c = 200;
14     }
15     else {
16       n = 3; c = 100;
17     }
18     drawShapes(90, n, c);
19     rotateZ(HALF_PI);
20   }
21 }
22
23 void drawShapes(int offset, int num, int colour) {
24   pushMatrix();
25   fill(colour);
26   translate(0, offset, 0);
27   int d = 80;
28   for (int i=0; i<num; i++) {
29     box(d);
30     translate(0, d, 0);
31     rotateY(PI/4);
32     d *= 0.75;
33   }
34   popMatrix();
35 }
```

(a) State the colour, size and position of the output generated by line 10 ("`box(100)`"). [3]

(b) In the context of this sketch, explain the effect of lines 24 (`pushMatrix()`) and 34 (`popMatrix()`) in the `drawShape()` method. [4]

(c) Lines 30 and 31 in the code are a call to `translate()` followed by a call to `rotateY()`:

i. In general, when performing a rotation and translation, does the order of applying these two operations matter? Explain your answer. [3]

ii. Would the output of this particular sketch be different if these two lines were swapped around (i.e. if *rotateY()* was called before *translate()*)? Explain your answer. [2]

(d) Draw a rough sketch of the output of this program. Label positions, sizes and colours where appropriate. [5]

(e) What would be the effect on the output of changing the last three parameters passed into the `camera()` method on line 3 from "0, 1, 0" to "1, 0, 0"? [2]

(f) Imagine we have a graphics file called `bricks.jpg` in the sketch's data folder, which contains a 100 by 100 pixel image. We would like to use this graphics file to render the front face of the centre cube in the sketch as a texture mapped surface. We can modify the sketch to accomplish this as follows:

Before line 1, insert

```
PImage bricks;
```

Before line 4, insert

```
bricks = loadImage(''bricks.jpg'');
```

Before line 11, insert

```
translate(0, 0, ?A?);
?B?
texture(?C?);
vertex(-50, -50, 0, 0);
vertex(?D?, ?E?, ?F?, ?G?);
vertex(?H?, ?I?, ?J?, ?K?);
vertex(-50, 50, 0, 100);
endShape();
translate(0, 0, ?L?);
```

In the final block of code shown above, there are twelve places where the real code has been replaced with the markers `?A?` to `?L?`. For each of these twelve markers, write down what the real code should be. For example, if you think the first line should read `translate(0, 0, 20)` instead of `translate(0, 0, ?A?)`, write A=20. [6]

END OF PAPER