OLLSCOIL NA GAILLIMHE
UNIVERSITY OF GALWAY

Semester 2 Examinations 2022/2023

Course Instance Code(s)	2BCT1, 1OA3
Exam(s)	B.Sc. Degree (Computer Science & Information Technology) International Exchange
Module Code(s)	CT255
Module(s)	Next Generation Technologies II
Paper No.	1
External Examiner(s)	Dr. Ramona Trestian
Internal Examiner(s)	Prof. Michael Madden *Dr. Michael Schukat *Dr. Sam Redfern

Instructions: **Answer three questions in total.**
Answer **one** question from Section A (20 marks)
AND
Answer **both** questions from Section B (10 marks each)

Duration	**2 hours**
No. of Pages	**7**
Discipline(s)	Computer Science
Course Co-ordinator(s)	Dr. Colm O'Riordan

Requirements:

Release in Exam Venue	Yes [x]	No [ ]
MCQ Answersheet	Yes [ ]	No [x]
Handout	None	
Statistical/ Log Tables	None	
Cambridge Tables	None	
Graph Paper	None	
Log Graph Paper	None	
Other Materials	None	
Graphic material in colour	Yes [x] No [ ]	

Section A (Cybersecurity)

Answer any one question from this section

Question 1[20 Marks]

a) Using examples, distinguish between the following GDPR-related terms:

a. Sensitive personal data
b. Data processor
c. Lawfulness
d. Purpose limitation

[4 marks]

b) Assume someone proposes an encryption algorithm based on the Playfair cipher principle that encodes / decodes large integer values (rather than letters). Discuss in some detail how such a solution could be possibly implemented, and highlight potential problems that need to be addressed.

[5 marks]

c) Show how Leslie Lamport's Algorithm can be used by a claimant and a verifier to create and validate one-time passwords.

[5 marks]

d) A Feistel cipher consists of multiple rounds. Outline the inner structure of a single round and, using an example, show how it can can be used for both encoding and decoding.

[6 marks]

Question 2 [20 Marks]

a) Browser cookies fall under GDPR regulations. However, there are two notable exemptions where user consent may not be required. What are the technical terms for these, and when do they apply? Use examples and diagrams to support your answer.

[4 marks]

b) Using an example outline how the Vigenère Cipher can be broken.

[6 marks]

c) Distinguish between passive attacks and active attacks, highlighting typical strategies for both. Use examples to illustrate the latter.

[4 marks]

d) What is a Linear Feedback Shift Register (LFSR) and how can it be used to create pseudorandom bistrings? Use a LFSR design of your own choice to support your answer that shows how the output is generated.

[6 marks]

PTO

Section B

Answer both questions from this section

Question 3 (10 Marks)

Consider the simple Java graphics program presented below (**pages 3-5**). This is the starting point for a 'fireworks' demo.

- It consists of a JFrame-derived class, 'Fireworks', with an embedded class 'Particle'.
- The Fireworks class contains a LinkedList of Particle objects.
- When the user clicks with the mouse, a new Particle is added to the LinkedList, and initialised to be positioned wherever the mouse clicked.
- Animation proceeds at 50 frames per second via a Thread.
- On each frame, each particle is updated via a call to its Update() method.
- Particles disappear after their 'lifetime' counter elapses.
- On each frame, each particle is drawn as a white box which does not move.

Your job is to add the following functionality. Please indicate carefully where your additional code should be added in the program. You do not need to write out any of the code provided below.

a) Change each particle so that it has a distinct randomised colour.

[3 marks]

b) Change each particle so that it has a velocity, i.e. a double for its x-direction speed and another double for its y-direction speed. A new particle should have these numbers randomised. The x-direction value should be in the range [-5, +5] and the y-direction value should be in the range [-8, 0]. On each update, the particle should move according to its velocity.

[4 marks]

c) Add the concept of gravity: on each update, the y-direction value of a particle's velocity should have 0.1 added to it.

[3 marks]

```
import java.awt.Color;
import java.awt.Graphics;
import java.awt.event.MouseEvent;
import java.awt.event.MouseListener;
import java.awt.image.BufferStrategy;
import java.util.*;
import javax.swing.JFrame;

public class Fireworks extends JFrame implements Runnable, MouseListener {

	private LinkedList particlesList = new LinkedList();
	private BufferStrategy strategy;
	private boolean initialised = false;

	// embedded class for handling individual particles
	public class Particle {
		private int x, y; // position
		private int lifetime = 200; // how many frames are left before
destruction
```

```
        // constructor
        public Particle(int x, int y) {
                this.x = x;
                this.y = y;
        }

        // updates the particle for 1 frame of animation
        // returns true if the particle should be destroyed
        public boolean update() {
                lifetime--;
                return (lifetime<=0);
        }

        public void paint(Graphics g) {
                g.setColor(Color.white);
                g.fillRect(x, y, 10, 10);
        }
}

// constructor
public Fireworks() {
        this.setBounds(10, 10, 600, 600);
        this.setVisible(true);
        addMouseListener(this);

        createBufferStrategy(2);
        strategy = getBufferStrategy();

        Thread t = new Thread(this);
        t.start();

        initialised = true;
}

public void mousePressed(MouseEvent e) {
}

public void mouseReleased(MouseEvent e) {
}

public void mouseEntered(MouseEvent e) {
}

public void mouseExited(MouseEvent e) {
}

public void mouseClicked(MouseEvent e) {
        int x = e.getX();
        int y = e.getY();
        Particle p = new Particle(x,y);
        particlesList.add(p);
}

public void run() {
        while(1==1) {
                try {
                        Thread.sleep(20);
                } catch (InterruptedException e) {

                        e.printStackTrace();
                }

                Iterator i = particlesList.iterator();
                while (i.hasNext()) {
                        Particle p = (Particle)i.next();
```

```
                        if (p.update())
                                i.remove();

                }

                this.repaint();
        }
    }

    public void paint(Graphics g) {
        if (initialised == true){
                g = strategy.getDrawGraphics();
                g.setColor(Color.black);
                g.fillRect(0,0,600,600);

                Iterator i = particlesList.iterator();
                while (i.hasNext()) {
                        Particle p = (Particle)i.next();
                        p.paint(g);
                }

                g.dispose();
                strategy.show();
        }

    }


    public static void main(String[] args) {
        Fireworks f = new Fireworks();
    }

}
```

PTO

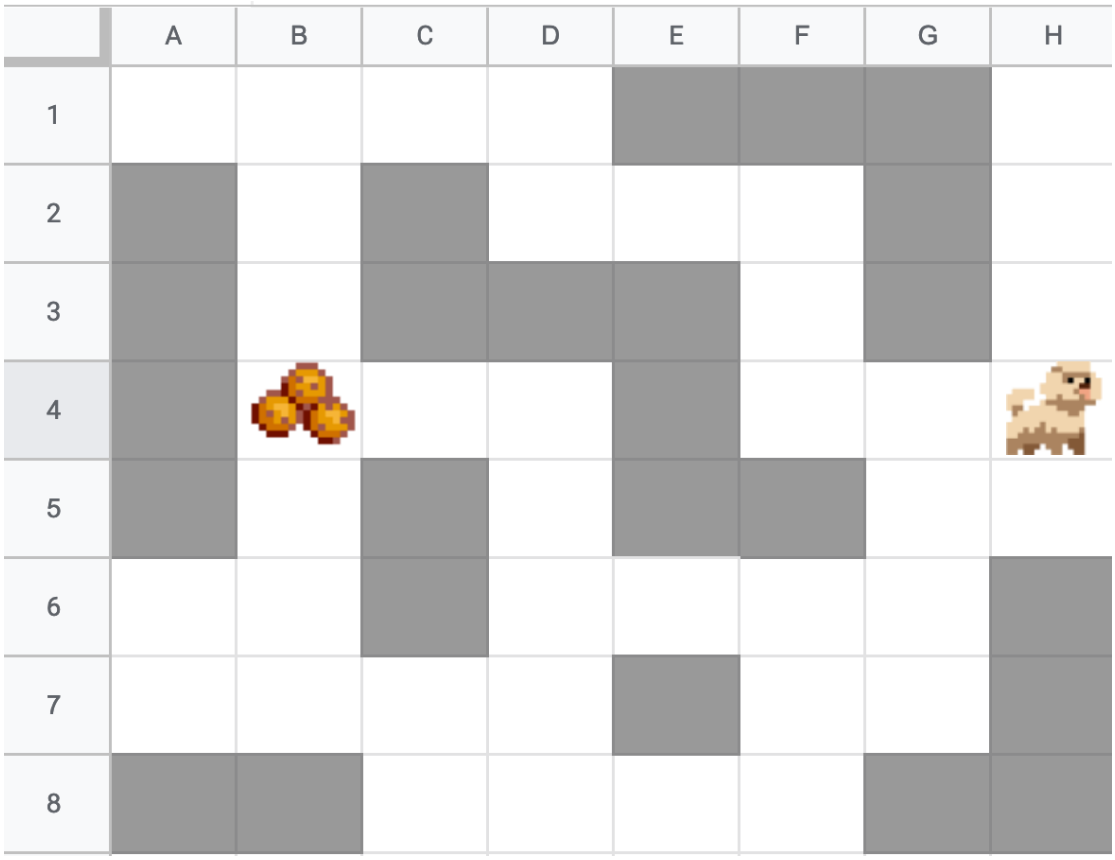
Question 4 (10 Marks)

a) With regard to the A* pathfinding algorithm, discuss the following terms: **node**, **open list**, **parent node**, **expanding a node**, **f, g and h values**

[5 marks]

b) Joey the naughty dog is visiting Sam and needs to get through a maze to steal his cookies (see diagram below). Joey's movements may only be made horizontally or vertically (not diagonally), by a full square at a time. He will use A* pathfinding. The shaded squares represent impassable walls. Annotate and submit the diagram below (**detachable version on final page of this exam paper**) with f, g, h values and parent node indication arrows for each square that will have been considered (i.e. added to the open list) before the correct path is discovered.

[5 marks]



PTO

You may detach and submit this page as part of your answer to Q.4., if you wish

CT255
Student ID _____