

## IDMT 2020

### Assignment 1

(Version: 20200914)

**This assessment is 10% of your total IDMT module grade.**

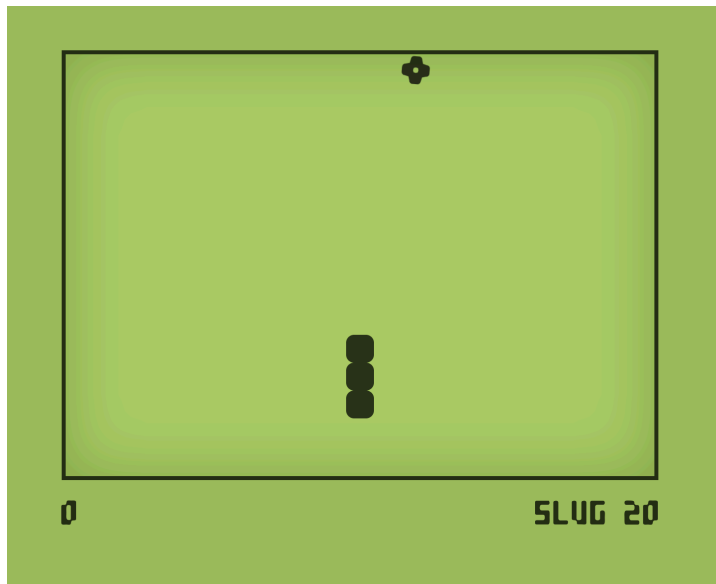
The purpose of this assignment is to help you become familiar with using Processing to create digital media interfaces. You will use Processing to create a simple game.

#### How the game is played: Snake

Example game:

<https://playsnake.org>

(Keys for this example version are: left: A; right: D; up: W; down: S)



Snake starts at 3 units long and is always moving forward.  
Each time it eats an apple it gets one unit longer, and points increase.  
Use keys to change direction of snake: left, right, up, down.  
Snake dies when it hits the wall or runs into itself.

## Instructions

You should write 4 Processing sketches – one for each of the steps below. Each step builds on the previous one, so make copies of your code as you move from one step to the next.

### Step 1: Make a snake 3 units long (sketch name: Step1)

Use an **array** to store the co-ordinates of the snake's 3 units. *Hint:* put the co-ordinates of the snake's head in array location 0, and the co-ordinates of the other 2 units of the snake (the tail) in array locations 1 and 2. Array location 2 is then the co-ordinates of the end of the tail of the snake.

*Hint:* You will need to store which direction the snake is travelling.

Draw the snake on the screen.

Move the snake forward (in its direction of travel) repeatedly. *Hint:* think of the array of snake units as a list of co-ordinates in *time* – when the snake moves forward the co-ordinates at array location 2 are forgotten, the co-ordinates at array location 1 are moved to location 2, and the co-ordinates at array location 0 are moved to location 1, and the new start co-ordinates of the snake are put into array location 0.

### Step 2: Add direction control (sketch name: Step2)

Use keyPressed() to detect presses of keys on the keyboard.

Use key presses to change direction of snake movement.

You should be able to guide the snake around the screen.

### Step 3: Add collision detection (sketch name: Step3)

Check whether snake hits the boundary – game over.

### Step 4: Add points (sketch name: Step4)

Add points to the game and display current points on the screen.

Add an apple at random location on the screen.

Check whether snake hits an apple – when it does the points increase, the apple disappears, and new apple appears on the screen.

## Marking Scheme

You receive the following % of your overall IDMT module grade.

Step	Sketch	Description
Step 1	0%: Not working at all 1%: Partially working 2%: Fully working	0%: No description or a list of comments from the code 1%: Unclear description 2%: Clear description
Step 2	0%: Not working at all 0.5%: Partially working 1%: Fully working	0%: No description or a list of comments from the code 0.5%: Unclear description 1%: Clear description
Step 3	0%: Not working at all 0.5%: Partially working 1%: Fully working	0%: No description or a list of comments from the code 0.5%: Unclear description 1%: Clear description
Step 4	0%: Not working at all 0.5%: Partially working 1%: Fully working	0%: No description or a list of comments from the code 0.5%: Unclear description 1%: Clear description

## Submission

**Your submission should consist of the following:**

- 4 Processing sketch folders containing the code for each step above. Your code should include clear comments to explain what each line of the code does. Make sure to credit any code that you have used in your sketches.
- Short report (PDF, max 2 sides of A4) describing how your code works for each step above. There should be an explanation for each step – explain **what the important variables are, why you use those variables, and how the code achieves the objectives of each step i.e.** how it works and why you wrote it that way. Make sure to credit any code that you have used in your sketches and explain how your code is different. You should include a section on your reflections about your code and how it works. This report should be written in plain English and should not just be a summary of the comments in your code.
- Name your **files with your initials and surname and student number**
- Make a .zip or .tar.gz file of all your code and report and submit it via QMplus
- Make sure the filename of the file you submit to QMPlus includes your **initials, surname, and student number**

Remember to **include your name and QMUL email** in both your code and report