

# **COMP115 Introduction to Computer Science**

Session 1, 2014

## **Assignment One: Swimmy Fish Part One**

**Due: 11:00am, Monday March 24, 2014 (Week Four)**

**Worth: 6%**

### **Learning Goals**

This assignment contributes to the learning outcomes of COMP115 as follows:

- *Apply problem solving skills to develop algorithms that solve small to medium-sized computational problems:* Introductory experience in developing an algorithm to solve a problem from an informal specification of the problem.
- *Design and code implementations of their algorithms in an imperative programming language:* Practical experience in translating a simple algorithm into an equivalent Processing program that uses drawing, animation and variables.
- *Use standard software engineering practices to document, debug and test their programs:* Experience in developing a clear program from a specification, testing the conformance of the program to the specification, and debugging any problems that are detected.

### **Introduction**

This assignment asks you to write a program to draw and animate shapes. You should only need to use material from up to and including Chapter Four of the textbook. The most relevant module is Pixels and Variables.

We strongly suggest that you divide the assignment up into pieces, tackle each piece separately and don't move on until you get each one working.

### **Questions**

If you have questions about the programming for this assignment, ask on the Assignments Forum on iLearn, consult with your tutor or ask Tony Sloane. No extensions will be granted except for serious and unavoidable disruption. Contact Matt Roberts as soon as possible, preferably before the due date, if you have suffered a disruption that affects your ability to submit this assignment.

## What your program needs to do

The aim of this assignment is to start the implementation of a game called “Swimmy Fish”. You will continue the implementation in the other two assignments. This assignment starts things off by setting up the game’s window, then drawing and animating the fish.

The following instructions explain what your program should do. There is also a video in the Assignment One materials on iLearn that gives you some more guidance about how your program should look and behave. Note: the video includes some circle effects that show you when the mouse button is clicked; you do not have to reproduce those effects in your program.

Your program should use a window that is 500 pixels wide and 400 pixels high. The background should be a blue-green colour to simulate an ocean. Your colour should approximate the colour shown in the assignment video as closely as you can.

After the initial window has been drawn, your program should behave as follows:

- Whenever the user clicks the mouse button with the mouse located anywhere in the window, a fish should be drawn. The appearance of your fish should match that shown in the assignment video as closely as possible and the centre of the fish body should be at the location where the mouse button was clicked.
- After a fish is initially drawn it should move slowly across the screen to the right, staying at the same vertical position in the window. When a fish reaches the right-hand edge of the window it should keep going until it disappears, never to be seen again.
- If the mouse button is clicked in the window while a fish is already on the screen, then that fish should disappear. A new fish should be drawn at the mouse click location and then move as described above. In other words, there should be at most one fish on the screen at a time and its vertical position should be given by the most recent mouse click in the window.

## Submission

Before the due date and time, you must submit your Processing program online via the COMP115 iLearn site. You must submit your program as a single Processing source file called **ass1.pde**.

You can find the pde file inside the folder that stores your Processing sketch. Please rename it (if necessary) and submit just that file. Do not submit the whole sketch.

## Marking

80% for the correctness of your code. We will check to see that your program correctly implements the specifications given above.

20% for the quality of your code. We will check to make sure that you have used reasonable names and types for your variables and that your code is presented in a manner that makes it understandable (e.g., good formatting, using comments to explain what your code is doing). Use the sample programs shown in lectures or the textbook as guidance.