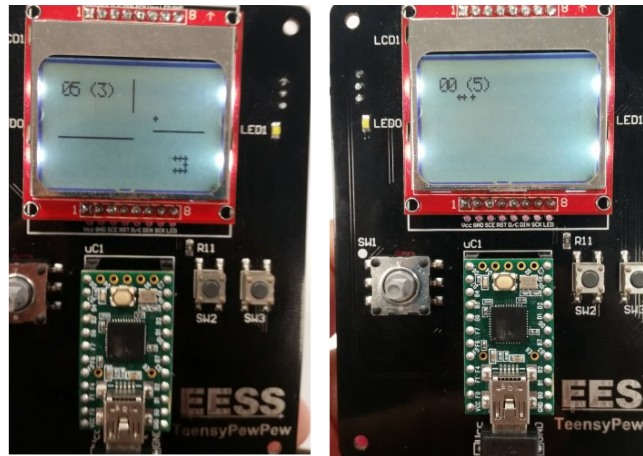


# CAB202 Assignment 2: Snake!!

*Assignment 2 will be marked during your assigned tutorial session in Week 13 (24/10 to 28/10)  
using the files submitted to AMS by 23/10  
Marks: 40 (40% of your final mark)*

For this assignment, you will be writing your own version of the snake game.



**Figure 1. Screenshots of demo game**  
(On the right, the snake with length 2 and the food in front of it with 5 lives.  
On the left, the snake eaten 5 foods and has 3 lives left. )

## Requirements

Your task is to implement a game in which you control a snake (i.e. a train of sprites) using SW1 (i.e. the 5-way switch) on your Teensy. The game starts with the snake at length 2 and with 5 lives. To move it, you use the 5-way switch.

The game should start with a welcome message displaying your name and student number for 2 seconds then the snake appears waiting for you to move SW1 (like in figure.1 right image) and the motion starts in the selected direction of SW1. After the first SW1 move, the snake does not stop and keep moving.

The food appears on a random position within the LCD. Every time the head of the snake (i.e. the first sprite in the train) runs over food, the snake grows by one segment (one sprite) and new food appears somewhere (randomly) on the LCD. The snake has a width of 3 (i.e each sprite in the train is size 3 by 3). The food should also be 3 by 3 in size.

The snake can always warp through the edges of the LCD (e.g exits from the right edge of the LCD and appears again on the left edge. The same applies for the top and bottom edges). However, the snake dies if you try to go backwards or if the snake cross itself.

A set of walls appears when you press SW3. A minimum of three walls should appear as straight lines with one end connected to the edge of the screen (see Figure 1). The walls should be parallel to either the X axis or the Y axis of the screen. The walls have width of 1 pixel and a length in the range between 10 pixels and half the

dimension of the screen (e.g if the wall is parallel to the X axis the length of the wall should be between 10 and 42 pixels). If the snake hits any of the walls, it dies and the player loses a life. Pressing SW2 should hide the walls. When the walls are visible, the score increases by 2 instead of 1.

The score and the lives remaining always appear on the top left corner of the LCD (see figure 1). The food should never appear over the score or over the walls.

When all lives are consumed, the game is over. A "Game over" appears on the LCD.

**WARNING: QUT takes plagiarism very seriously. If your assignment has material which has been copied from other class members, previous assignments, or code from any other source you will be penalised severely. If you cannot explain exactly what your code is doing, it is clearly not your own work, and you should not be submitting it as such!**

## Level 1: Basic functionality [20 marks total]

- Your name and student number appear for 2 seconds before the game start. [MANDATORY] [1 marks]
- Screen setup: a score, remaining lives appear at the top left corner of the screen. [3 marks]
- The snake is drawn with a width of 3 pixels and a length of 2 sprites on the LCD. Food is drawn on the LCD with size 3 by 3 pixels. [3 marks]
- Moving the SW1 switch makes the snake start moving in the selected direction by the switch. [3 marks]
- The score increase by 1 when the head of the snake goes over the food. [2 marks]
- The snake increases in length by 1 sprite every time it run over the food [3 marks]
- Each time the food is consumed, it reappears randomly in a new unoccupied position on the LCD [3 marks]
- The snake loses a life if it crosses itself. [2 marks]

## Level 2: [10 marks total]

- The snake loses a life if it receives a command from the SW1 switch to go in the opposite direction of its current motion [3 marks]
- Every time the snake loses a life, its length is reset to 2 sprites and appears like it did at the beginning of the game waiting for the player to move the SW1 switch. [3 marks]
- If the snake exits from on edge of the LCD, it reenters the screen from the opposite edge. [2 marks]
- The speed of the snake can be controlled by one of the potentiometers. Moving the potentiometer in one direction makes the snake to move faster, moving it in the opposite direction slows down the snake. Speeds should be reasonable so the game is still playable. [2 marks]
- 

## Level 3: [10 marks total]

- Pressing SW3 draw a set of walls on the LCD. A minimum of three walls should appear as straight lines with one end connected to the edge of the screen. The walls should be parallel to either the X axis or the Y axis of the screen. The walls have width of 1 pixel and a length in the range between 10 pixels and half the dimension of screen (e.g if the wall is parallel to the X axis the length of the wall can be between 10 and 42 pixels). [3 marks]
- Pressing SW2 should hide the walls. [2 marks]
- When the walls are visible, consuming food adds 2 points to the score instead of 1 [2 marks]
- The snake dies and loses a life if it crosses any of the walls. [3 marks]

# Marking

The assignment will be out of 40 marks and will be worth 40% of your total grade in this subject. The following should be noted about marking:

- **If your code does not compile, you will get 0 marks for the entire assignment.**
- If the game crashes or locks up during testing, you will receive marks for what has been tested up to that point. No more of your assignment will be marked. We will not debug your code to make it compile or run.
- Your game must be easily playable. If timings, settings, or controls are set in a manner that makes it difficult to play (e.g. not using the key inputs specified, ridiculously fast movement, etc.), you **will receive 0 for the assignment**.
- The assignment will be marked during your scheduled tutorial in week 13. To receive a mark for the assignment, you are required to:
  - attend the tutorial in person,
  - bring your student ID card,
  - demonstrate your program to a tutor,
  - explain details of your implementation, and
  - hand in your Teensy.

**If you fail to do any of these things you will receive a mark of 0 for the assignment.**

# Submission

Your assignment is due on Sunday October 23th, 2016 at 11:59pm.

Submission will be online through the AMS used in the tutorials. You must submit through the submission page (available at <http://bio.mquter.qut.edu.au/cab202/Exercise?TopicNumber=6&ProblemNumber=1>), and follow all of the instructions. The link will be active on Monday 19/09/2016.

When submitting to the AMS, it is your responsibility to make sure that your assignment compiled correctly. In general, AMS will compile your code and return any errors that occur. There are circumstances when very poor code causes the compiler on the server to terminate abnormally, or to hang. You are responsible for ensuring that this does not happen by submitting only valid source code which you have already compiled using the same settings as those used by AMS. As mentioned above, if you do not submit a compiled assignment, you will receive 0 marks. You will have 50 submission attempts for the assignment, so there are no excuses for not resolving any compilation issues.

AMS operates in unattended mode outside regular Monday to Friday office hours. Please note that issues such as software failure, hardware failure, and network congestion problems DO NOT constitute exceptional circumstances for the purpose of Special Consideration. It is in your best interest to complete this assignment early to eliminate the possibility of a last-minute disaster. To ensure success you should aim to submit a substantially complete version of your assignment by the beginning of Week 12.