CAB202 Tutorial 2
Practice with Conditionals

Booleans in C

Booleans (variables which can either have the value of true or false) are not a native type in the C programming language. There are many different ways to introduce boolean like variables into C. For this unit, we will take advantage of the fact we are compiling against **gnu99** standards, and use the **stdbool.h** library. This library allows you to use Booleans (through the **bool** identifier) like they were a native type. The snippet below shows how to include them and use them in basic operation:

```
#include <stdbool.h>

int main() {
	bool my_boolean = true;
	if (my_boolean) {
		// Will do this
	} else if (!my_boolean) {
		// Will not do this
	}
}
```

Using “cab202_timers” from the ZDK

The ZDK includes a series of functions to provide timer-like operation. To use them, you need to include the appropriate header file, and call your desired function (in this tutorial we will only be using the **timer_pause()** function). For example, to pause for 5 seconds:

```
#include “cab202_timers.h”

int main() {
	// Pause for 5 seconds
	timer_pause(5000);
}
```

Non-Assessable Exercises

NOTE: These exercises are not assessable, and the tutor can help you with them. Assessable exercises are in the following section. Do NOT attempt to complete these exercises by reading pixels from the screen.

1. Detect if a character is drawn within a box

The template provided in **question_1.c** on Blackboard is for a game where a player is trying to bomb (represented by an '**x**' character) a home base (represented by ‘**H**’ characters). The edges of the home base go from 25% to

75% of both the screen width and screen height. In the logic of this game, there needs to be a function that evaluates whether the last bomb hit the base.

The demo code provided for this exercise has a game which repeatedly drops bombs at random locations until **has_bomb_hit()** returns true. At the moment, this function always returns false. To make the game work correctly, provide an implementation that returns true when the bomb coordinates hit a part of the base, and false otherwise.

Challenge exercises:

- *Extend the game so that a bomb is dropped every time the space bar is pressed. Then update the behaviour so that the player only has 5 bombs. If all of the bombs are used, and none hit the base, instead present an alternate message telling the player that they lost.*

2. Detect a bullet hit in a space invaders style game

The template provided in **question_2.c** on Blackboard is for a space invaders game, where the bullet fires across the screen until it collides with an object. In this example consider a bullet (represented by a '**-**' character) that starts in the middle of the leftmost column (i.e. column is 0 and row is 50% of screen height) and shoots towards the right. At some point in its trajectory there will be a rock (represented by a '**#**' character).

The demo code provided for this exercise draws a rock at a random position and calls function **shoot_rock()**. Complete this function so that the bullet continues to the right until the rock is adjacent to the bullet (i.e. directly to the left of the rock).

3. Investigate the scope of a variable in the previous solution

In the template used above for question 2, the variable named **rock_x** was used in a number of different places throughout the *.c file. Answer the following questions:

- How many times is the variable declared? Are you allowed to declare variables multiple times? What allows multiple declarations of the variable in this code?
- How come changing the value of the variables in within the **draw_rock()** function doesn't affect its value in **main()**? Make modifications to the code to verify this behaviour.
- Is it valid to declare variables outside functions? What happens if a variable is declared at the top of the source file after the #include directives? Why is declaring variables there often advised against?

Assessed Exercises (AMS)

Complete the assessed exercises via the AMS (available at http://bio.mquter.qut.edu.au/CAB202).