**Spike:** 21

**Title:** Control Mapping & Configuration Files

**Author:** Ben Holmes, 103024841

**Goals / deliverables:**

Create a simple application (that uses your framework (SDL2) capture input events (such as key down or up events),  
and map the input to a change.  
You must specifically demonstrate:

1. That input can be mapped to components based on configuration data at run-time, and
2. Be able to change and reload-input mapping configuration without restarting the application.

**Technologies, Tools, and Resources used:**

* Visual Studio 2022
* Word

**Tasks undertaken:**

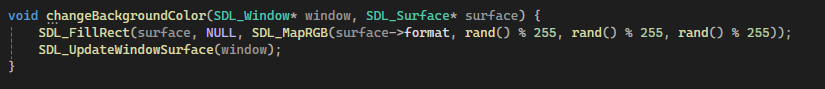
* Copied across spike 17 code
* Adjusted change background to be a function
* Create the key mapping from config file (this will also act as the reset function)
* Create the key change function
* Test
* Add in console output for ease of reading
* Made a very obvious change I cant believe I missed

**What we found out:**

This was a rather simple task in terms of what needed to be done and how long it took me although I did have a painful time with one part

Here are the functions:

This changes the background color randomly



This changes the keys to hardcoded values (this what the changeBindings button does)

A black screen with text

Description automatically generated

The two button keyCodes



The event while loop and key registration

A computer screen shot of a program code

Description automatically generated

The split function for the file input

A screen shot of a computer code

Description automatically generated

A screen shot of a computer program

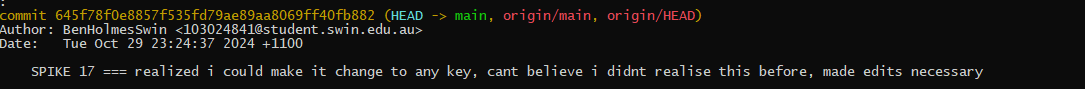
Description automatically generated

And the file input, the painful one and the reason this took me two hours instead of 1. Because of the base encoding of the text file when I would try to read in the tokens[0].data() (the 1 char, “r”) it would not covert over to the int (ASCII) value correctly as it was a const char\* and after searching extensively to figure out how I got the strcpy\_s to transfer it to a char array and then use the first char as the key binding. This does come with the minor problem of potential overloading but if the file structure is different anyway, you are probably going to have problems as it is.

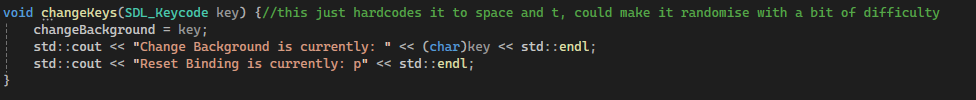
A screen shot of a computer code

Description automatically generated

Cant believe I didn’t realise till after I had done it that I could easily set it to the key that was just pressed. I am dumb



Removed changeBindings and changed the changeKeys function:



Also changed the main loop

A computer screen with text on it

Description automatically generated

It does display the int ascii conversion of the keycode in the console, which is accurate for the standard letter keys, but is wrong for the keypad numbers.

The reset button does not change and you can press at button to rebind the background change to it and then press it again to change color

Also have confirmed that it works the way that James will test on the discord with letter and number keys