## HOWTO:

To play unzip the KeypadTestUnity zip file and double click KeypadTest. No Extra steps should be needed.

In case a Windows machine is not available the game is also available online at <https://brunocitoni.itch.io/keypadtest> with PASSWORD keypad

## Design Log:

I began by figuring out the architecture I wanted to give the project.

The keypad controller is responsible for listening to UI events such as button presses, handling that information, and notifying the other elements. This is done via C# Actions implementing a basic Observer pattern.

The Keypad manager instantiates the Keypad buttons in runtime as well as assigning them a value and a callback. This callback, which sends the value of each numerical key pressed is sent to the controller who then notify its listeners.

Listening to these changes are the Keypad display, which updates its UI in response to changes in the currently selected code and the Keypad enter button, which waits for the current code to be consisting of 4 digits before becoming interactable.

Upon entering a code with the enter button, the controller checks whether the code is correct, in which case the display shows an UNLOCKED message. If the code is incorrect, a timer is started via the Timer class, during which intractability is removed from the keypad. When the timer is finished, the keypad is restored to its functioning state.

While the same behaviour could have been achieved using fewer scripts and less architectural complexity, I decided to go for a more de-coupled approach. This way the current Keypad prefab I created is then reusable as a self-contained entity that can be expanded in the future.

I was also slightly surprised by the lack of a 0 in the keypad but I stuck to the brief and only implemented 1-9.