

verifySequence.h

```
1 /* verifySequence.h
2 *
3 *     Author: Taylor Cowley
4 */
5
6 #ifndef VERIFYSEQUENCE_H_
7 #define VERIFYSEQUENCE_H_
8
9 #include <stdint.h>
10 #include <stdbool.h>
11 #include <stdio.h>
12 #include "supportFiles/display.h"
13 #include "simonDisplay.h"
14 #include "buttonHandler.h"
15 #include "globals.h"
16 #include "buttons.h"
17
18 #define VERIFYSEQUENCE_TIMEOUT_SPEED (GLOBALS_VERIFY_TIMEOUT_SPEED)
19
20 enum verifySequence_st_t{
21     verifySequence_init_st, //Init everything (like the screen)
22     wait_for_enable_v,      //we can't do anything unless enabled
23     wait_for_touch,         //wait for the user to touch a button
24     wait_for_release,       //we wait for the touch sensor to cool down
25     analyze_touch,          //time to register the touch
26     end_verify_sequence,    //we have ended the verify sequence
27     wait_for_disable_v      //chill here until disabled
28 };
29
30
31 // State machine will run when enabled.
32 void verifySequence_enable();
33
34 // This is part of the interlock. You disable the state-machine and then enable it again.
35 void verifySequence_disable();
36
37 // Used to detect if there has been a time-out error.
38 bool verifySequence_isTimeOutError();
39
40 // Used to detect if the user tapped the incorrect sequence.
41 bool verifySequence_isUserInputError();
42
43 // Used to detect if the verifySequence state machine has finished verifying.
44 bool verifySequence_isComplete();
45
46 // Standard tick function.
47 void verifySequence_tick();
48
49 // Standard runTest function.
50 void verifySequence_runTest();
51
52 #endif /* VERIFYSEQUENCE_H_ */
53
```