

buttonHandler.h

```
1 #ifndef BUTTONHANDLER_H_
2 #define BUTTONHANDLER_H_
3 #include <stdint.h>
4 // Get the simon region numbers. See the source code for the region numbering scheme.
5 uint8_t buttonHandler_getRegionNumber();
6
7 // Turn on the state machine. Part of the interlock.
8 void buttonHandler_enable();
9
10 // Turn off the state machine. Part of the interlock.
11 void buttonHandler_disable();
12
13 // The only thing this function does is return a boolean flag set by the buttonHandler
   state machine. To wit:
14 // Once enabled, the buttonHandler state-machine first waits for a touch. Once a touch
   is detected, the
15 // buttonHandler state-machine computes the region-number for the touched area. Next,
   the buttonHandler
16 // state-machine waits until the player removes their finger. At this point, the
   state-machine should
17 // set a bool flag that indicates the the player has removed their finger. Once the
   buttonHandler()
18 // state-machine is disabled, it should clear this flag.
19 // All buttonHandler_releasedDetected() does is return the value of this flag.
20 // As such, the body of this function should only contain a single line of code.
21 bool buttonHandler_releaseDetected();
22
23 // Standard tick function.
24 void buttonHandler_tick();
25
26 // This tests the functionality of the buttonHandler state machine.
27 // buttonHandler_runTest(int16_t touchCount) runs the test until
28 // the user has touched the screen touchCount times. It indicates
29 // that a button was pushed by drawing a large square while
30 // the button is pressed and then erasing the large square and
31 // redrawing the button when the user releases their touch.
32 void buttonHandler_runTest(int16_t touchCount);
33
34 #endif /* BUTTONHANDLER_H_ */
35
```