```c
/*
 * simonControl.c
 *
 *  Created on: Jun 11, 2015
 *      Author: Taylor Cowley
 */


#include "simonControl.h"


void simonControl_tick() {
    //This stores the current state of our machine
    static simonControl_st_t currentState = simonControl_init_st;

    //This is the current level.
    static uint16_t currentLevel = SIMONCONTROL_START_LEVEL;

    //This is a generic delay counter for our timeouts
    static int16_t delay = 0;

    //First we do the state actions
    switch(currentState){
    case simonControl_init_st:              //Init everything (like the screen)
        //init stuff
        globals_setSequenceIterationLength(1);  //We start with displaying only one of the sequence.
        simonControl_start_rand_timer();        //start the timer for our create_random_sequence function
        simonControl_display_splash();          //display the splash screen
        break;

    case start_wait_for_touch_st:               //start when they touch
        //do nothing. they need to touch to start
        break;

    case start_wait_for_release_st:             //I lied. start when they release
        //do nothing. they need to let go for their touch to count
        break;

    case flash_sequence_st:                     //show the user the current sequence
        //We do nothing- the flash_sequence state machine needs to run
        break;

    case validate_sequence_st:                  //Let the user try
        //We do nothing- the validate_sequence state machine needs to run
        break;

    case success_st:                            //the user succeeded at this level!
        delay--;            //we are displaying until this times out
        break;

    case touch_for_new_level_st:                //We finished the current level!
        delay--;            //we are displaying until the time runs out
        break;

    case failure_st:                            //the user failed to tap properly
```

```
56          delay--;            //we are displaying until the time runs out
57          break;
58
59      case longest_run_st:                        //we have ended the verify sequence
60          delay--;            //we are displaying until time runs out
61          break;
62
63      default:                                    //This is an error
64          printf("Invalid state in simonControl!");       //display error
65          break;
66
67      }
68
69
70      //Then we do the state change
71      switch(currentState){
72      case simonControl_init_st:          //Init everything (like the screen)
73          currentState = start_wait_for_touch_st;     //only one tick inside init.
74          break;
75
76      case start_wait_for_touch_st:       //start when they touch
77          //as soon as they touch, get the timer for the rand seed and make a new sequence.
78          if(display_isTouched()){                    //we only move on when they touch
79              simonControl_create_random_sequence(currentLevel);  //we make the sequence for
    this level
80              currentState = start_wait_for_release_st;           //and now we wait for the
    button release
81          }
82          break;
83
84      case start_wait_for_release_st:     //I lied. start when they release
85          if(!display_isTouched()){                   //they let go!
86              display_fillScreen(DISPLAY_BLACK);      //blank the splash screen
87              currentState = flash_sequence_st;       //we can continue
88              flashSequence_enable();                 //enable the flashSequence SM!
89          }
90          break;
91
92      case flash_sequence_st:             //show the user the current sequence
93          if(flashSequence_completed()){              //have we completed flashing the sequence?
94              currentState = validate_sequence_st;    //yes! move on to validate sequence
95              display_fillScreen(DISPLAY_BLACK);          //Fill the background
96              flashSequence_disable();                //disable the flashSequence SM!
97              verifySequence_enable();                //enable the validateSequence SM!
98          }
99          break;
100
101     case validate_sequence_st:          //Let the user try
102         if(verifySequence_isComplete()){
103             //Did we fail?
104             if(verifySequence_isTimeOutError() || verifySequence_isUserInputError()){
105                 currentState = failure_st;          //We failed. Go to fail screen
106                 delay = SIMONCONTROL_FAILURE_DELAY; //init the delay for their failure
107                 simonControl_display_failure();     //Show the user their embarassing failure
108             } else {
109                 //if we have succeeded at this level, go to success!
110                 if(globals_getSequenceLength() == globals_getSequenceIterationLength()){
```

```
111                     currentState = success_st;          //Tell the user they beat the level
112                     delay = SIMONCONTROL_SUCCESS_DELAY; //init the delay timer for success
113                     simonControl_display_success();     //tell the user they won
114                 } else {    //if we still have things to go for this level, increment and go!
115                     globals_setSequenceIterationLength(globals_getSequenceIterationLength() +
    1);
116                     currentState = flash_sequence_st;   //and we flash again!
117                     display_fillScreen(DISPLAY_BLACK);          //Fill the background
118                     flashSequence_enable();             //we are entering flashSequence- need
    to enable it.
119                 }
120             }
121             verifySequence_disable();                   //disable the validateSequence SM!
122         }
123 
124         break;
125 
126     case success_st:                    //the user succeeded at this level!
127         if(delay <= 0){                             //if our timer is over
128             currentState = touch_for_new_level_st;  //move to next state
129             delay = SIMONCONTROL_TOUCH_NEW_LEVEL;   //reset the delay timer for our next state
130             simonControl_display_touch_for_new_level();     //and now to display that
131         }
132         break;
133 
134     case touch_for_new_level_st:
135         if(delay <= 0){                             //the user didn't touch in time
136             currentState = longest_run_st;              //so we tell them their score
137             delay = SIMONCONTROL_LONGEST_RUN_DELAY;     //init the delay to tell them their
    longest run
138             simonControl_display_best_run(globals_getSequenceIterationLength());    //Tell
    them their highest score this round
139         } else if (display_isTouched()){                //the user touched it and wants a new
    level!
140             currentState = start_wait_for_touch_st; //we wait for them to release it now.
141             globals_setSequenceIterationLength(1);  //We start with displaying only one of the
    sequence.
142             simonControl_create_random_sequence(++currentLevel);//make the level harder and
    increment level.
143         }
144         break;
145 
146     case failure_st:                    //the user failed to tap properly
147         if(delay<= 0){
148             currentState = longest_run_st;              //time to move on
149             simonControl_display_best_run(globals_getSequenceIterationLength()-1);  //tell
    them their highest score this round
150             delay = SIMONCONTROL_LONGEST_RUN_DELAY;     //init the delay to tell them their
    longest run
151         }
152         break;
153 
154     case longest_run_st:                //we have ended the verify sequence
155         if(delay<= 0){
156             currentState = simonControl_init_st;        //so we start the game over
157         }
158         break;
```

```
159
160     default:                            //This is an error
161         printf("Invalid state in simonControl!");       //display error
162         break;
163
164     }
165
166 }
167
168
169 //Starts the timer that we use for the rand seed
170 void simonControl_start_rand_timer(){
171     //init timer for rand seed - we don't reset it because we don't care what number is in it
172     intervalTimer_init(SIMONCONTROL_RAND_TIMER);
173     //start the timer!
174     intervalTimer_start(SIMONCONTROL_RAND_TIMER);
175 }
176
177 //Stops the timer that we use for the rand seed
178 void simonControl_stop_rand_timer(){
179     intervalTimer_stop(SIMONCONTROL_RAND_TIMER);
180 }
181
182 //Turns the timer that we use for the rand seed into a random sequence for our game.
183 void simonControl_create_random_sequence(uint16_t length){
184     //Make our sequence. Might as well make it size length
185     uint8_t newSequence[length];
186
187     //get the 64 bit int for the rand timer, MOD it with total possible sequences, and seed
    rand with it
188     srand(intervalTimer_getTotalDuration(SIMONCONTROL_RAND_TIMER));// %
    SIMONCONTROL_TOTAL_POSSIBLE_SEQUENCES);
189
190     //We get a new random number for every item in our array
191     for(uint16_t i = 0; i < length; i++){
192         //We mod it with the number of buttons and store it in our array
193         newSequence[i] = rand() % SIMONCONTROL_NUM_BUTTONS;
194     }
195
196     //We make the global sequence the one we just made
197     globals_setSequence(newSequence, length);
198
199 }
200
201 //This is the starting splash screen
202 void simonControl_display_splash() {
203     display_init();                                 //gotta init the screen!
204       display_fillScreen(DISPLAY_BLACK);            //Fill the background
205       display_setCursor(0, SIMONCONTROL_UPPER_HEIGHT);      //Center the text vertically
206       display_setTextColor(DISPLAY_RED);            //Cyan is a good text color
207       display_setTextSize(SIMONCONTROL_TITLE_SIZE); //We are doing the title first
208       display_println("Simon");                               //display simon
209       display_setCursor(0, SIMONCONTROL_LOWER_HEIGHT);      //Center the text vertically
210       display_setTextSize(SIMONCONTROL_STATUS_TEXT_SIZE);   //We are doing the subtitle next
211       display_println("Touch to start");                     //tell them to touch
212 }
213
```

```
214 //This is the "touch for new level" screen
215 void simonControl_display_touch_for_new_level(){
216       display_fillScreen(DISPLAY_BLACK);            //Fill the background
217       display_setTextColor(DISPLAY_RED);            //Cyan is a good text color
218       display_setCursor(0, SIMONCONTROL_LOWER_HEIGHT);      //Center the text vertically
219       display_setTextSize(SIMONCONTROL_STATUS_TEXT_SIZE);   //We are doing the subtitle next
220       display_println("Touch = new level");             //tell them touch for new level
221
222 }
223
224 //This is the success screen
225 void simonControl_display_success(){
226       display_fillScreen(DISPLAY_BLACK);            //Fill the background
227       display_setTextColor(DISPLAY_RED);            //Cyan is a good text color
228       display_setCursor(0, SIMONCONTROL_LOWER_HEIGHT);      //Center the text vertically
229       display_setTextSize(SIMONCONTROL_STATUS_TEXT_SIZE);   //We are doing the subtitle next
230       display_println("You win!");              //tell them touch for new level
231 }
232
233 //This is the failure screen
234 void simonControl_display_failure(){
235       display_fillScreen(DISPLAY_BLACK);            //Fill the background
236       display_setTextColor(DISPLAY_RED);            //Cyan is a good text color
237       display_setCursor(0, SIMONCONTROL_LOWER_HEIGHT);      //Center the text vertically
238       display_setTextSize(SIMONCONTROL_STATUS_TEXT_SIZE);   //We are doing the subtitle next
239       display_println("You FAIL!");             //tell them touch for new level
240 }
241
242 //This is the best run screen
243 void simonControl_display_best_run(uint16_t best_score){
244       display_fillScreen(DISPLAY_BLACK);            //Fill the background
245       display_setTextColor(DISPLAY_RED);            //Cyan is a good text color
246       display_setCursor(0, SIMONCONTROL_LOWER_HEIGHT);      //Center the text vertically
247       display_setTextSize(SIMONCONTROL_STATUS_TEXT_SIZE);   //We are doing the subtitle next
248
249       char str[SIMONCONTROL_DISPLAY_SCREEN_MAX_LENGTH];     //storage for our string
250       sprintf(str, "Best run: %d", best_score); //make our formatted string
251       display_println(str);             //tell them touch for new level
252 }
253
```