```
#include <Adafruit_NeoPixel.h>

int cycleStrip = 5; // Assigns the strip of 15 Neopixels to Pin 5 on the Lilypad

Adafruit_NeoPixel strip = Adafruit_NeoPixel(15, cycleStrip, NEO_GRB + NEO_KHZ800);

int cycleHand = 9; // Assigns the Neopixels on the hand-shapes to Pin 9 on the Lilypad

Adafruit_NeoPixel striphand = Adafruit_NeoPixel(6, cycleHand, NEO_GRB + NEO_KHZ800);

int gameSwitch = 6; // Assigns the switch to Pin 6 on the Lilypad
int scoreKeeper = 7;

int Hand0 = 10; // Assigns the lower-left hand-shape to Pin 10 on the Lilypad.
int Hand1 = A2; // Assigns the middle-left hand-shape to Pin A2 on the Lilypad.
int Hand2 = A5; // Assigns the top hand-shape to Pin A5 on the Lilypad.
int Hand3 = A4; // Assigns the middle-right hand-shape to Pin A4 on the Lilypad.
int Hand4 = A3; // Assigns the lower-right hand-shape to Pin A3 on the Lilypad.
int Hand5 = 11; // Assigns the lower-middle hand-shape to Pin 11 on the Lilypad.

void setup() {
  strip.begin();
  strip.show();
  striphand.begin();
  striphand.show(); // Initialize all Neopixels to 'off'
  pinMode(gameSwitch,INPUT_PULLUP); // I think this defines how the pin is going to "listen" for something to happen on the switch.
  Serial.begin(57600);

  pinMode(Hand0, INPUT_PULLUP); // I think this defines how the pins are going to "listen" for something to happen on the lower-left hand-shape.
  pinMode(Hand1, INPUT_PULLUP); // I think this defines how the pins are going to "listen" for something to happen on the middle-left hand-shape.
  pinMode(Hand2, INPUT_PULLUP); // I think this defines how the pins are going to "listen" for something to happen on the top hand-shape.
  pinMode(Hand3, INPUT_PULLUP); // I think this defines how the pins are going to "listen" for something to happen on the middle-right hand-shape.
  pinMode(Hand4, INPUT_PULLUP); // I think this defines how the pins are going to "listen" for something to happen on the lower-right hand-shape.
  pinMode(Hand5, INPUT_PULLUP); // I think this defines how the pins are going to "listen" for something to happen on the lower-middle hand-shape.
}

void loop() {
      gameStart (); // Runs the gameStart function defined below
      scoreKeeper = 7; // Sets the beginning score to 7, which is in the exact center of the strip of 15 Neopixels
      showScore(scoreKeeper); // Runs the showScore function defined below, using the parameter of scoreKeeper
      delay(1000); // Pauses for one second
      if(!digitalRead(gameSwitch)){ // Checks to see if the switch is turned on
            while((scoreKeeper>0) && (scoreKeeper<14)){ // If the switch IS on, this checks to see if the current score is 0 (a loss) or 15 (a win).
                  handflash(); // Runs the handflash function, which is defined below.
                  scoreKeeper = scoreKeeper + play_round(); // Assigns the running score to go up or down by 1 depending upon what value the play_round function returns.
                  showScore(scoreKeeper); // Runs the showScore function (defined below), causing the new score to be displayed on the Neopixel strip.
                  Serial.println(scoreKeeper); // A programming help that allows someone to observe what the score is after each round.
            }


      }
}

void showScore(int score){ // This function will just display a score by lighting one of the Neopixels on the strip.
      for (int x = 0; x<=15; x++){ // This loop will run through all possible values of the player's score, starting at 0 and going up to 15 by increments of 1.
            if (x==score){ // If and when the loop actually matches the current score, it will execute the next line, otherwise it'll skip to the "else" below.
                  strip.setPixelColor(x,   120, 120, 0); // This is a medium brightness yellow color, activated on the current score's Neopixel rank.
            }else{
                  strip.setPixelColor(x,   0, 0, 0); // This is an unlit Neopixel.
            }
      }
      strip.show(); // This causes the Neopixel commands in the above function to execute.
      delay(200); // This pauses things for a fifth of a second.


}




void gameStart () {
      for (int y = 0; y<=15; y++){
            for (int x = 0; x<=15; x++){
                  if (x==y){
                        strip.setPixelColor(x,   120, 120, 0);
                  }else{
                        strip.setPixelColor(x,   0, 0, 0);
                  }
            }
            strip.show();
            delay(200);
      }




      for (int y = 0; y<=6; y++){
            for (int x = 0; x<=6; x++){
                  if (x==y){
                        striphand.setPixelColor(x,   120, 120, 0);
                  }else{
                        striphand.setPixelColor(x,   0, 0, 0);
                  }
            }
            striphand.show();
            delay(200);
      }
}

void handflash(){ // This function cycles one light through all six Neopixels above the hand-shapes as a signal that the next round of the game is about to begin. It runs through them all first in red, then green, then blue.
      int flashdelay = 100;
      for (int y = 0; y<=6; y++){
            for (int x = 0; x<=6; x++){
                  if (x==y){
                        striphand.setPixelColor(x,   255, 0, 0);
                  }else{
                        striphand.setPixelColor(x,   0, 0, 0);
                  }
            }
            striphand.show();
            delay(flashdelay);
      }

      for (int y = 0; y<=6; y++){
            for (int x = 0; x<=6; x++){
                  if (x==y){
                        striphand.setPixelColor(x,   0, 255, 0);
                  }else{
                        striphand.setPixelColor(x,   0, 0, 0);
                  }
            }
            striphand.show();
            delay(flashdelay);
      }

      for (int y = 0; y<=6; y++){
            for (int x = 0; x<=6; x++){
                  if (x==y){
                        striphand.setPixelColor(x,   0, 0, 255);
                  }else{
                        striphand.setPixelColor(x,   0, 0, 0);
                  }
            }
            striphand.show();
            delay(flashdelay);
      }
}




int play_round(){ // I think this just turns off all the lights before the randomly determined on is told to turn on on lines 164 - 171
      for (int i = 0 ; i <= 5 ; i++){
    striphand.setPixelColor(i,   0, 0, 0);
      }

      delay(2000);

  int min_time = 800; // in ms
  unsigned long base_time = millis();  // Track round start time

  // Setup
  randomSeed(millis());    // Seed random function



  // Random from 0 to "this number"-1
  int light    = random(6);      // Random target selection
  //int delta_t  = random(200);    // Random time interval padding to min time
  int delta_t = -(scoreKeeper - 7)*50; // Assigns a change in time interval by adding or subtracting up to 350 milliseconds depending upon how low or high the current score is.


  // Turn light on
  for (int i = 0 ; i <= 5 ; i++){
    if (i==light){
      striphand.setPixelColor(i,   255, 0, 0);
    }else{
      striphand.setPixelColor(i,   0, 0, 0);
    }
  }
  striphand.show(); // Update lights


  // Check for hit or miss
  while( millis() < (base_time + min_time + delta_t) ){
    int read_0 = !digitalRead(Hand0);  // Sample here
    int read_1 = !digitalRead(Hand1);  // Sample here
    int read_2 = !digitalRead(Hand2);  // Sample here
    int read_3 = !digitalRead(Hand3);  // Sample here
    int read_4 = !digitalRead(Hand4);  // Sample here
    int read_5 = !digitalRead(Hand5);  // Sample here


    // Goes through all six cases in the hand-shapes, resulting in one of them being correct, and either that one being closed by the player or not.
    if (read_0 == 1){
      if (light == 0){
        return 1;
      }else{
        return -1;
      }
    }


    if (read_1 == 1){
      if (light == 1){
        return 1;
      }else{
        return -1;
      }
    }


    if (read_2 == 1){
      if (light == 2){
        return 1;
      }else{
        return -1;
      }
    }


    if (read_3 == 1){
      if (light == 3){
        return 1;
      }else{
        return -1;
      }
    }


    if (read_4 == 1){
      if (light == 4){
        return 1;
      }else{
        return -1;
      }
    }



    if (read_5 == 1){
      if (light == 5){
        return 1;
      }else{
        return -1;
      }
    }



  }
  Serial.println("timeout"); // This is saying that if the player fails to hit any hand-space, right or wrong, within the time interval, it counts as a "wrong."
  return -1;

}
```