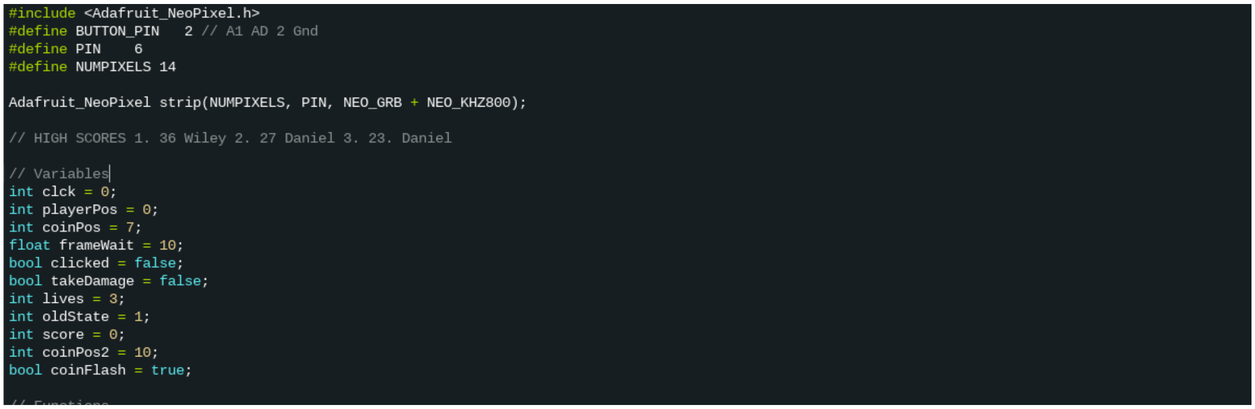
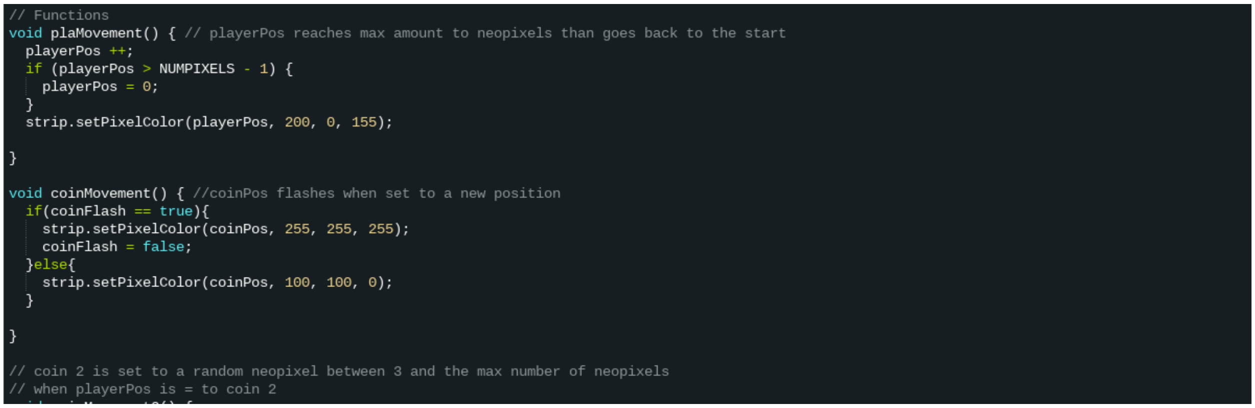
```
#include <Adafruit_NeoPixel.h>
#define BUTTON_PIN   2 // A1 AD 2 Gnd
#define PIN    6
#define NUMPIXELS 14

Adafruit_NeoPixel strip(NUMPIXELS, PIN, NEO_GRB + NEO_KHZ800);

// HIGH SCORES 1. 36 Wiley 2. 27 Daniel 3. 23. Daniel

// Variables
int clck = 0;
int playerPos = 0;
int coinPos = 7;
float frameWait = 10;
bool clicked = false;
bool takeDamage = false;
int lives = 3;
int oldState = 1;
int score = 0;
int coinPos2 = 10;
bool coinFlash = true;

// Functions
void plaMovement() { // playerPos reaches max amount to neopixels than goes back to the start
  playerPos ++;
  if (playerPos > NUMPIXELS - 1) {
    playerPos = 0;
  }
  strip.setPixelColor(playerPos, 200, 0, 155);

}

void coinMovement() { //coinPos flashes when set to a new position
  if(coinFlash == true){
    strip.setPixelColor(coinPos, 255, 255, 255);
    coinFlash = false;
  }else{
    strip.setPixelColor(coinPos, 100, 100, 0);
  }
```

```
}

// coin 2 is set to a random neopixel between 3 and the max number of neopixels
// when playerPos is = to coin 2
void coinMovement2() {
  strip.setPixelColor(coinPos2, 100, 100, 0);
  if (coinPos2 == playerPos) {
    coinPos2 = random(3, NUMPIXELS);
  }
}


// if playerPos is equal to the coin 1 position than +1 to the score
// playerPos speed increases by .2 each time the button is pressed -
// - when the positions are equal
// coinPos is set to a random neopixel between 3 and the max -
// - of neopixels
// coinPos flashes when position is changed
// if the button is not pressed when playerPos and coinPos  is equal -
// - then you lose a life
void checkClick() {
  if (clicked == true) {
    if (playerPos == coinPos) {
      score++;
      frameWait -= 0.2;
      coinPos = random(3, NUMPIXELS);
      coinFlash = true;
    } else {
      takeDamage = true;
    }
  }
}


void checkDamage() {
  if (takeDamage == true) {
    takeDamage = false;
    lives--; // each time you press the button when playerPos and coinPos are not equal you lose 1
life
    if (lives > 0) { // players loses life but doens't die
```

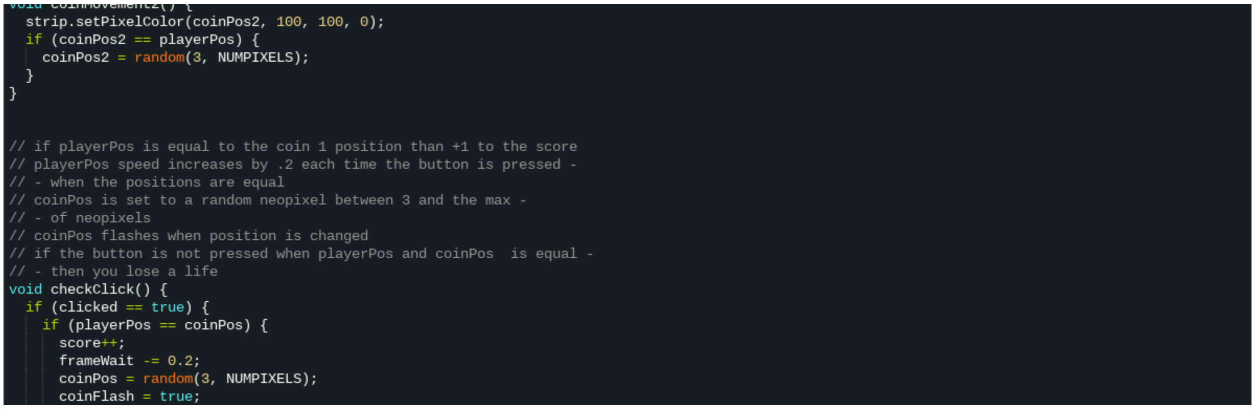
```
      strip.clear();
      strip.setPixelColor(playerPos - 1, 255, 0, 0);
      strip.show();
      playerPos = 0; // when player loses a life playerPos resets to neopixel 1
      delay(1000);
    } else { // player dies
      Serial.println(score); // prints the end score in the monitor

      frameWait = 10; // original speed of playerPos
      lives = 3; // amount of lives at the start
      playerPos = 0; // player position
      score = 0; // starting score
      strip.clear();

      for(int i = 0; i < NUMPIXELS; i++){
        strip.setPixelColor(i, 155, 0, 0);
      }

      strip.show();

      delay(5000);
    }
  }
}



void setup() {
  Serial.begin(9600);
  strip.setBrightness(70);
  pinMode(BUTTON_PIN, INPUT_PULLUP);
  strip.begin();
  strip.clear();
  strip.show();
}

void loop() {
  clck++;

  if (digitalRead(BUTTON_PIN) != oldState) {
    if (digitalRead(BUTTON_PIN) == LOW) {
      clicked = true;
    }
  }


  if (clck >= 20) {
    clck  = 0;
    strip.clear();
    checkClick();
    coinMovement();
    coinMovement2();
    plaMovement();
    checkDamage();

    strip.show();
    clicked = false;
  }
  oldState = digitalRead(BUTTON_PIN);
  delay(frameWait);
}
```



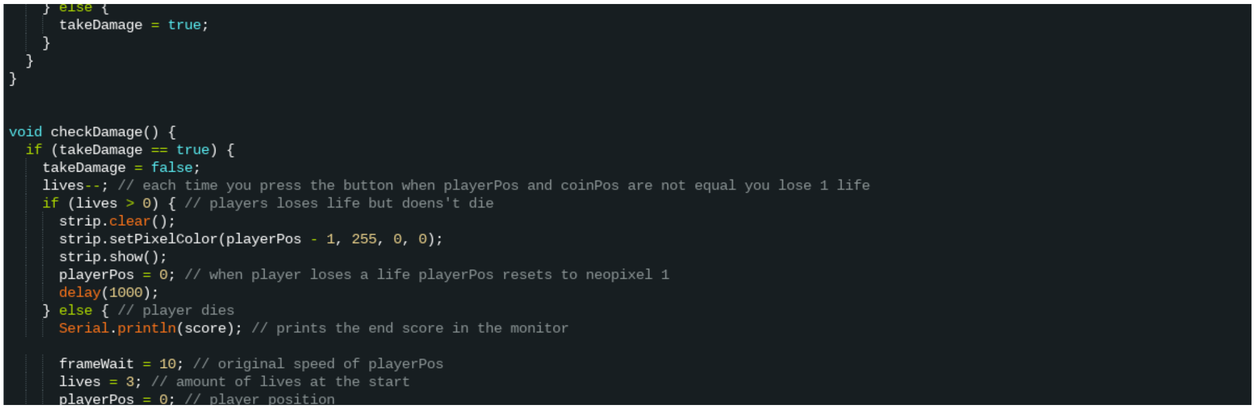
```
#include <Adafruit_NeoPixel.h>
#define BUTTON_PIN   2 // A1 AD 2 Gnd
#define PIN    6
#define NUMPIXELS 14

Adafruit_NeoPixel strip(NUMPIXELS, PIN, NEO_GRB + NEO_KHZ800);

// HIGH SCORES 1. 36 Wiley 2. 27 Daniel 3. 23. Daniel

// Variables
int clck = 0;
int playerPos = 0;
int coinPos = 7;
float frameWait = 10;
bool clicked = false;
bool takeDamage = false;
int lives = 3;
int oldState = 1;
int score = 0;
int coinPos2 = 10;
bool coinFlash = true;

// Functions
```



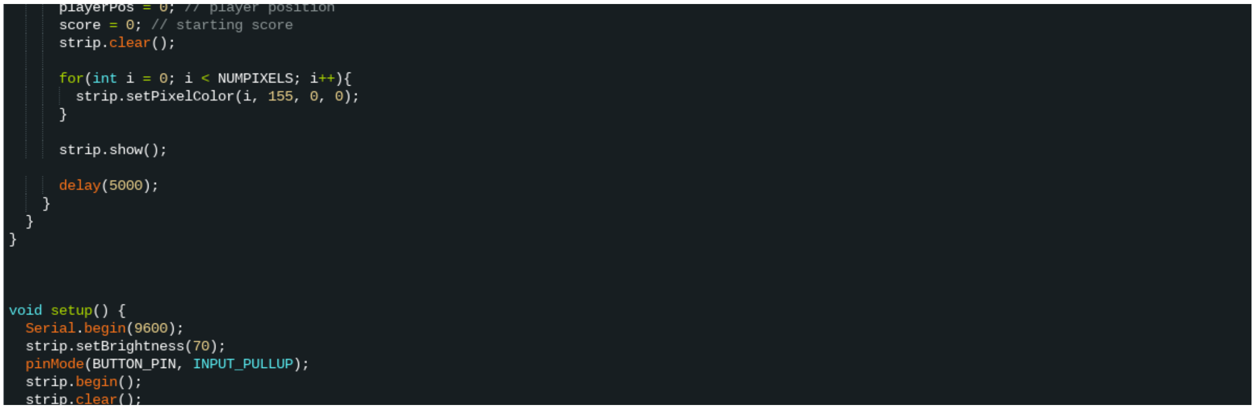
```
// Functions
void plaMovement() { // playerPos reaches max amount to neopixels than goes back to the start
  playerPos ++;
  if (playerPos > NUMPIXELS - 1) {
    playerPos = 0;
  }
  strip.setPixelColor(playerPos, 200, 0, 155);

}

void coinMovement() { //coinPos flashes when set to a new position
  if(coinFlash == true){
    strip.setPixelColor(coinPos, 255, 255, 255);
    coinFlash = false;
  }else{
    strip.setPixelColor(coinPos, 100, 100, 0);
  }

}

// coin 2 is set to a random neopixel between 3 and the max number of neopixels
// when playerPos is = to coin 2
```



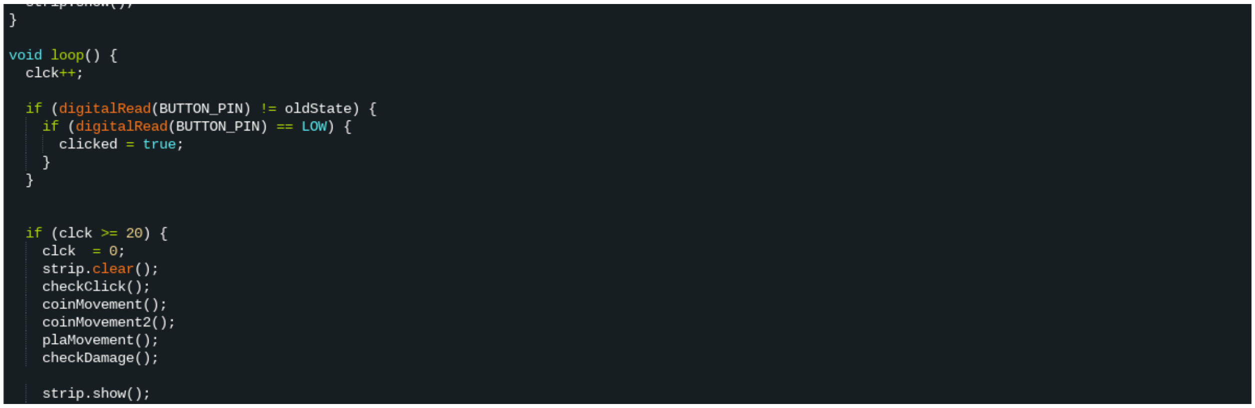
```
void coinMovement2() {
  strip.setPixelColor(coinPos2, 100, 100, 0);
  if (coinPos2 == playerPos) {
    coinPos2 = random(3, NUMPIXELS);
  }
}


// if playerPos is equal to the coin 1 position than +1 to the score
// playerPos speed increases by .2 each time the button is pressed -
// - when the positions are equal
// coinPos is set to a random neopixel between 3 and the max -
// - of neopixels
// coinPos flashes when position is changed
// if the button is not pressed when playerPos and coinPos  is equal -
// - then you lose a life
void checkClick() {
  if (clicked == true) {
    if (playerPos == coinPos) {
      score++;
      frameWait -= 0.2;
      coinPos = random(3, NUMPIXELS);
      coinFlash = true;
```



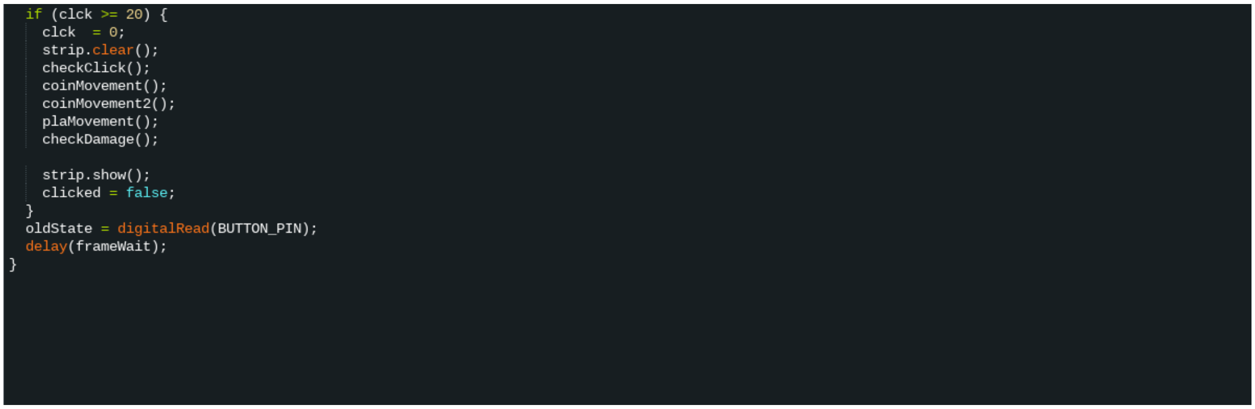
```
    } else {
      takeDamage = true;
    }
  }
}


void checkDamage() {
  if (takeDamage == true) {
    takeDamage = false;
    lives--; // each time you press the button when playerPos and coinPos are not equal you lose 1 life
    if (lives > 0) { // players loses life but doens't die
      strip.clear();
      strip.setPixelColor(playerPos - 1, 255, 0, 0);
      strip.show();
      playerPos = 0; // when player loses a life playerPos resets to neopixel 1
      delay(1000);
    } else { // player dies
      Serial.println(score); // prints the end score in the monitor

      frameWait = 10; // original speed of playerPos
      lives = 3; // amount of lives at the start
      playerPos = 0; // player position
```



```
      score = 0; // starting score
      strip.clear();

      for(int i = 0; i < NUMPIXELS; i++){
        strip.setPixelColor(i, 155, 0, 0);
      }

      strip.show();

      delay(5000);
    }
  }
}



void setup() {
  Serial.begin(9600);
  strip.setBrightness(70);
  pinMode(BUTTON_PIN, INPUT_PULLUP);
  strip.begin();
  strip.clear();
```



```
  strip.show();
}

void loop() {
  clck++;

  if (digitalRead(BUTTON_PIN) != oldState) {
    if (digitalRead(BUTTON_PIN) == LOW) {
      clicked = true;
    }
  }


  if (clck >= 20) {
    clck  = 0;
    strip.clear();
    checkClick();
    coinMovement();
    coinMovement2();
    plaMovement();
    checkDamage();

    strip.show();
```



```
  if (clck >= 20) {
    clck  = 0;
    strip.clear();
    checkClick();
    coinMovement();
    coinMovement2();
    plaMovement();
    checkDamage();

    strip.show();
    clicked = false;
  }
  oldState = digitalRead(BUTTON_PIN);
  delay(frameWait);
}
```