Pong Game

Nume : Unguroiu Victor

Grupa : 237/2

In this Arduino project, I'll be creating a version of the Pong game. You'll use the right push button to move the paddle up and the left push button to move it down, ensuring the ball doesn't get past your paddle. This version of the game is single-player against the computer and doesn't include a scoreboard.

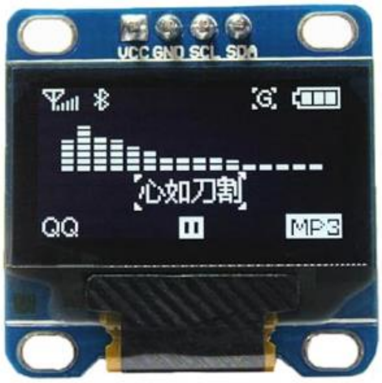
For this project you need following things :

1x Arduino uno : https://store.arduino.cc/products/arduinounorev3?queryID=826717a7c95881418581f7c757c386bf

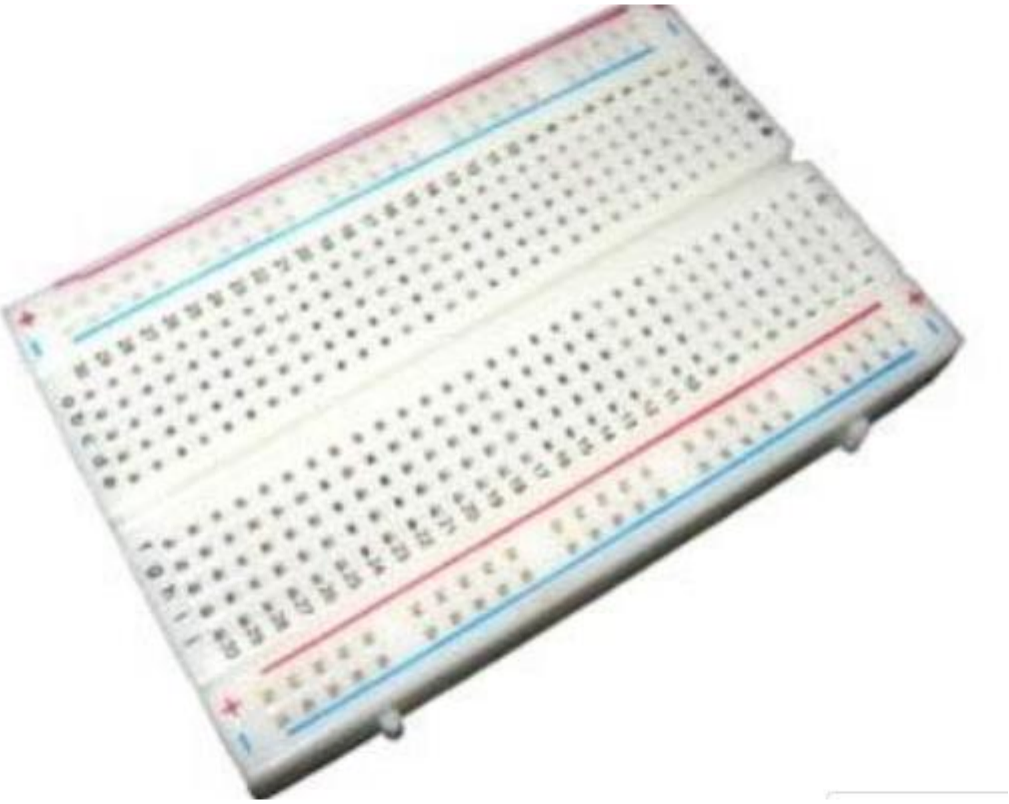
2x push buttons

1x oled display 0.96 ssd1306 i2c :

https://www.amazon.com.au/DIYmall-Serial-128x64-Display-Arduino/dp/B00O2KDQBE

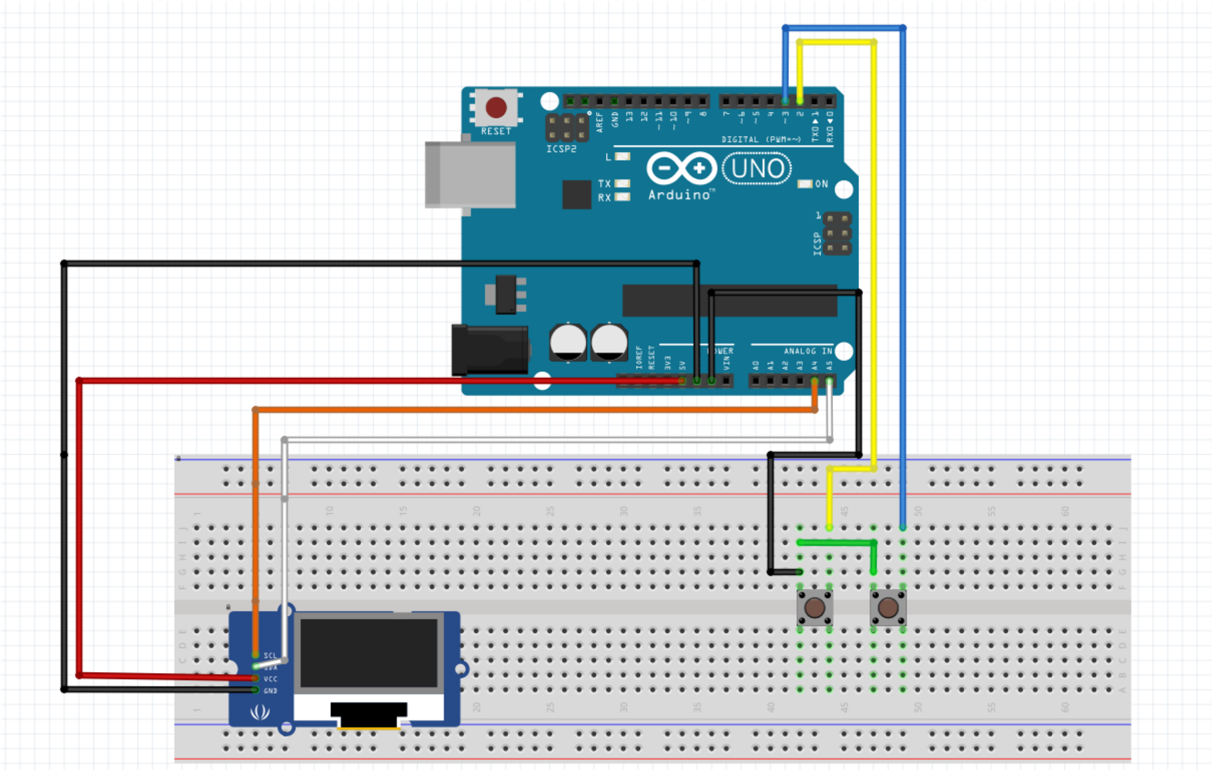


1x breadboard : https://store.arduino.cc/products/breadboard-400-contacts?queryID=80ac52d8f2ef83557f511642ffdbf4e6



Few jumpers : https://store.arduino.cc/products/10-jumper-wires-150mm-male?queryID=c6d1144eb16aa173093e4d6369717384

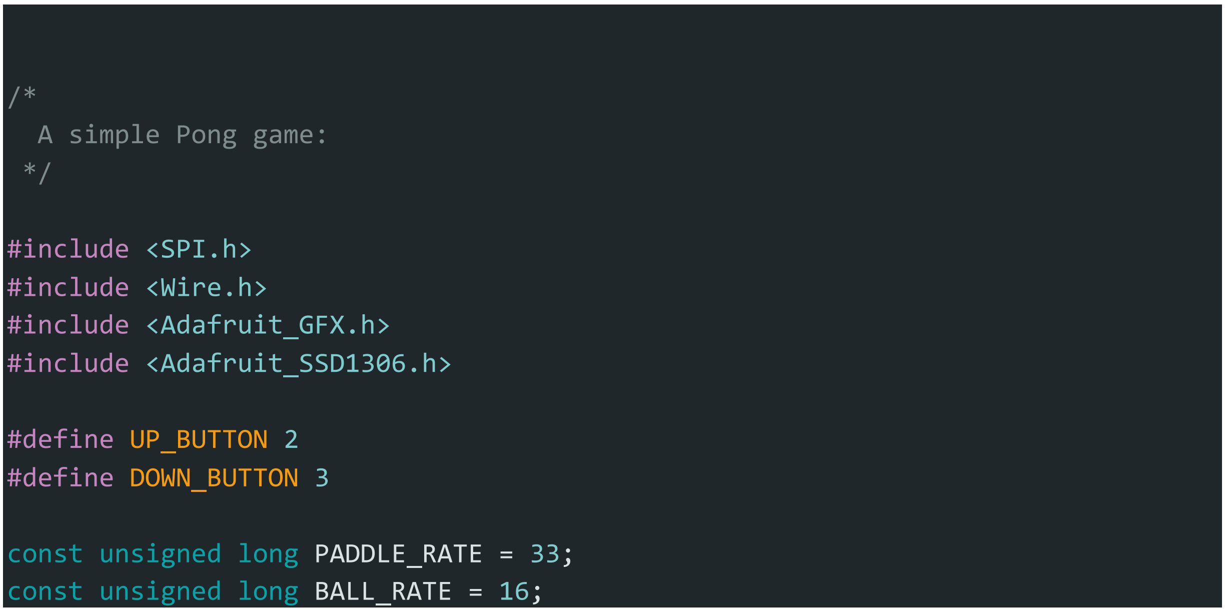
Below is the wiring diagram for how the components should be connected:



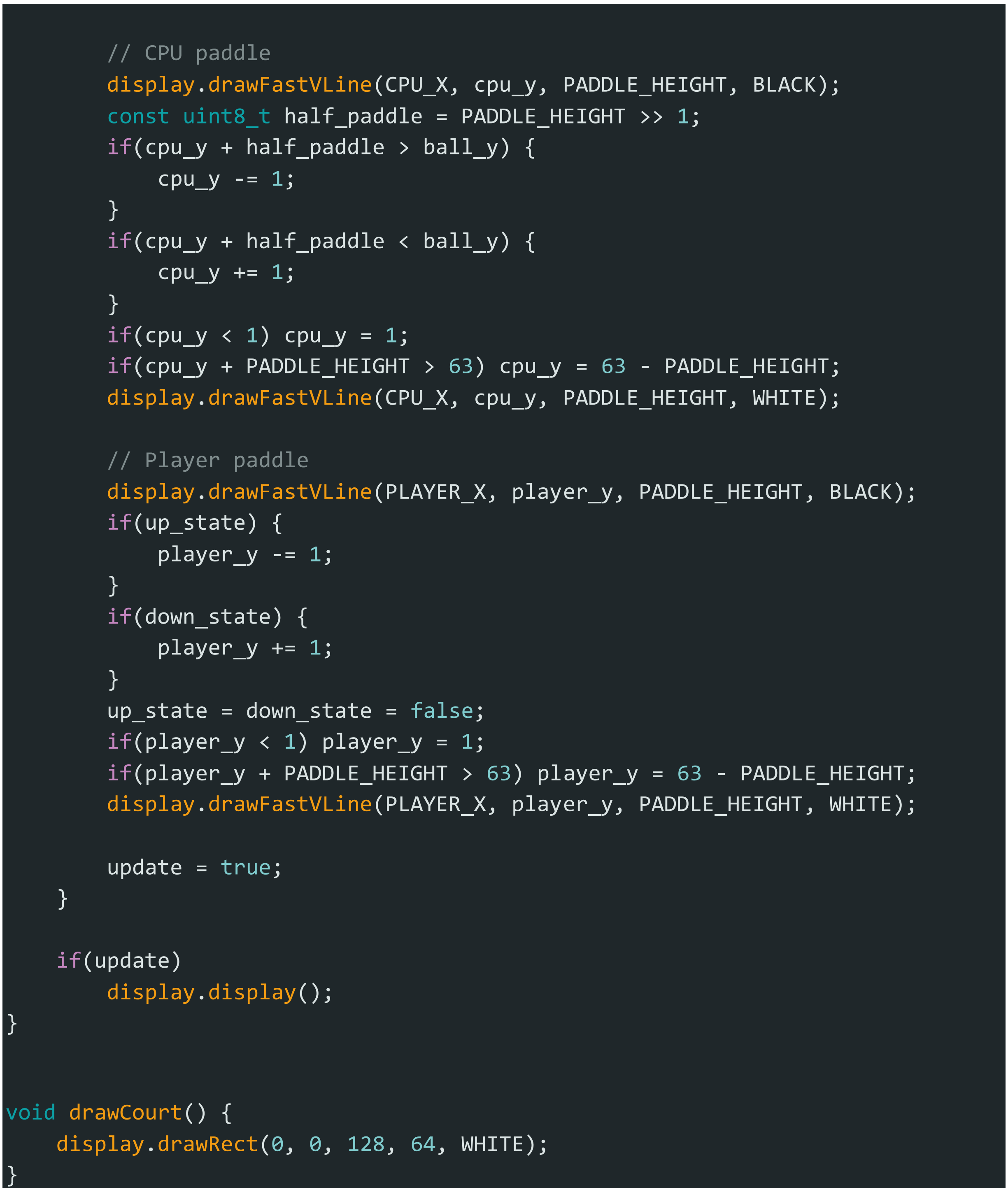
How to play :

Simply upload the code to the Arduino Uno board, and the game will initiate. Your paddle will be positioned on the right side of the screen. Press the left button to move it upward and the right button to move it downward.

For the Code i used this Sketch in Arduino IDE



```
/*
  A simple Pong game:
 */

#include <SPI.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

#define UP_BUTTON 2
#define DOWN_BUTTON 3

const unsigned long PADDLE_RATE = 33;
const unsigned long BALL_RATE = 16;
const uint8_t PADDLE_HEIGHT = 24;

#define SCREEN_WIDTH 128 // OLED display width, in pixels
#define SCREEN_HEIGHT 64 // OLED display height, in pixels

// Declaration for an SSD1306 display connected to I2C (SDA, SCL pins)
#define OLED_RESET     4 // Reset pin # (or -1 if sharing Arduino reset pin)
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);

void drawCourt();

uint8_t ball_x = 64, ball_y = 32;
uint8_t ball_dir_x = 1, ball_dir_y = 1;
unsigned long ball_update;

unsigned long paddle_update;
const uint8_t CPU_X = 12;
uint8_t cpu_y = 16;

const uint8_t PLAYER_X = 115;
uint8_t player_y = 16;

void setup() {
    display.begin(SSD1306_SWITCHCAPVCC, 0x3C);


    display.display();
    unsigned long start = millis();

    pinMode(UP_BUTTON, INPUT);
    pinMode(DOWN_BUTTON, INPUT);
    digitalWrite(UP_BUTTON,1);
    digitalWrite(DOWN_BUTTON,1);
    display.clearDisplay();
    drawCourt();

    while(millis() - start < 2000);

    display.display();

    ball_update = millis();
    paddle_update = ball_update;
}

void loop() {
    bool update = false;
    unsigned long time = millis();

    static bool up_state = false;
    static bool down_state = false;

    up_state |= (digitalRead(UP_BUTTON) == LOW);
    down_state |= (digitalRead(DOWN_BUTTON) == LOW);

    if(time > ball_update) {
        uint8_t new_x = ball_x + ball_dir_x;
        uint8_t new_y = ball_y + ball_dir_y;

        // Check if we hit the vertical walls
        if(new_x == 0 || new_x == 127) {
            ball_dir_x = -ball_dir_x;
            new_x += ball_dir_x + ball_dir_x;
        }

        // Check if we hit the horizontal walls.
        if(new_y == 0 || new_y == 63) {
            ball_dir_y = -ball_dir_y;
            new_y += ball_dir_y + ball_dir_y;
        }

        // Check if we hit the CPU paddle
        if(new_x == CPU_X && new_y >= cpu_y && new_y <= cpu_y + PADDLE_HEIGHT)
{
            ball_dir_x = -ball_dir_x;
            new_x += ball_dir_x + ball_dir_x;
        }

        // Check if we hit the player paddle
        if(new_x == PLAYER_X
           && new_y >= player_y
           && new_y <= player_y + PADDLE_HEIGHT)
        {
            ball_dir_x = -ball_dir_x;
            new_x += ball_dir_x + ball_dir_x;
        }

        display.drawPixel(ball_x, ball_y, BLACK);
        display.drawPixel(new_x, new_y, WHITE);
        ball_x = new_x;
        ball_y = new_y;

        ball_update += BALL_RATE;

        update = true;
    }

    if(time > paddle_update) {
        paddle_update += PADDLE_RATE;
```



```
        // CPU paddle
        display.drawFastVLine(CPU_X, cpu_y, PADDLE_HEIGHT, BLACK);
        const uint8_t half_paddle = PADDLE_HEIGHT >> 1;
        if(cpu_y + half_paddle > ball_y) {
            cpu_y -= 1;
        }
        if(cpu_y + half_paddle < ball_y) {
            cpu_y += 1;
        }
        if(cpu_y < 1) cpu_y = 1;
        if(cpu_y + PADDLE_HEIGHT > 63) cpu_y = 63 - PADDLE_HEIGHT;
        display.drawFastVLine(CPU_X, cpu_y, PADDLE_HEIGHT, WHITE);

        // Player paddle
        display.drawFastVLine(PLAYER_X, player_y, PADDLE_HEIGHT, BLACK);
        if(up_state) {
            player_y -= 1;
        }
        if(down_state) {
            player_y += 1;
        }
        up_state = down_state = false;
        if(player_y < 1) player_y = 1;
        if(player_y + PADDLE_HEIGHT > 63) player_y = 63 - PADDLE_HEIGHT;
        display.drawFastVLine(PLAYER_X, player_y, PADDLE_HEIGHT, WHITE);

        update = true;
    }

    if(update)
        display.display();
}


void drawCourt() {
    display.drawRect(0, 0, 128, 64, WHITE);
}
```