### SNAKE Game using ARDUINO

### Required Components

* Arduino (e.g., Arduino Uno)
* OLED Display (128x64, SSD1306)
* Push buttons for controls (optional)

### Wiring

* Connect the OLED display to your Arduino:
  + VCC to 5V
  + GND to GND
  + SCL to A5 (SCL)
  + SDA to A4 (SDA)

### Libraries

Make sure to install the following libraries via the Library Manager in the Arduino IDE:

1. **Adafruit SSD1306**
2. **Adafruit GFX**

### C Code for Snake Game

Here’s the Arduino code for a simple Snake game:

cpp

Copy

#include <Wire.h>#include <Adafruit\_GFX.h>#include <Adafruit\_SSD1306.h>#define SCREEN\_WIDTH 128#define SCREEN\_HEIGHT 64#define OLED\_RESET -1Adafruit\_SSD1306 display(SCREEN\_WIDTH, SCREEN\_HEIGHT, &Wire, OLED\_RESET);#define SNAKE\_SIZE 5#define MAX\_LENGTH 100int snakeX[MAX\_LENGTH], snakeY[MAX\_LENGTH];int snakeLength = 1;int foodX, foodY;int direction = 1; // 0: Up, 1: Right, 2: Down, 3: Leftbool gameOver = false;void setup() { display.begin(SSD1306\_I2C\_ADDRESS, OLED\_RESET); display.clearDisplay(); // Initialize snake position snakeX[0] = SCREEN\_WIDTH / 2; snakeY[0] = SCREEN\_HEIGHT / 2; // Generate food generateFood(); // Set the initial game speed delay(200);}

void loop() { if (!gameOver) { display.clearDisplay(); drawSnake(); drawFood(); display.display(); // Handle input if (Serial.available()) { char input = Serial.read(); if (input == 'w' && direction != 2) direction = 0; // Up if (input == 'd' && direction != 3) direction = 1; // Right if (input == 's' && direction != 0) direction = 2; // Down if (input == 'a' && direction != 1) direction = 3; // Left }

moveSnake(); checkCollision(); delay(100); } else { display.setCursor(0, 0); display.setTextSize(1); display.setTextColor(SSD1306\_WHITE); display.println("Game Over!"); display.display(); }

}

void drawSnake() { for (int i = 0; i < snakeLength; i++) { display.fillRect(snakeX[i], snakeY[i], SNAKE\_SIZE, SNAKE\_SIZE, SSD1306\_WHITE); }

}

void drawFood() { display.fillRect(foodX, foodY, SNAKE\_SIZE, SNAKE\_SIZE, SSD1306\_WHITE);}

void moveSnake() { // Move the snake's body for (int i = snakeLength; i > 0; i--) { snakeX[i] = snakeX[i - 1]; snakeY[i] = snakeY[i - 1]; }

// Update the head of the snake if (direction == 0) snakeY[0] -= SNAKE\_SIZE; // Up if (direction == 1) snakeX[0] += SNAKE\_SIZE; // Right if (direction == 2) snakeY[0] += SNAKE\_SIZE; // Down if (direction == 3) snakeX[0] -= SNAKE\_SIZE; // Left // Check if food is eaten if (snakeX[0] == foodX && snakeY[0] == foodY) { snakeLength++; generateFood(); }

}

void generateFood() { foodX = (random(0, (SCREEN\_WIDTH / SNAKE\_SIZE))) \* SNAKE\_SIZE; foodY = (random(0, (SCREEN\_HEIGHT / SNAKE\_SIZE))) \* SNAKE\_SIZE;}

void checkCollision() { // Check wall collisions if (snakeX[0] < 0 || snakeX[0] >= SCREEN\_WIDTH || snakeY[0] < 0 || snakeY[0] >= SCREEN\_HEIGHT) { gameOver = true; }

// Check self-collision for (int i = 1; i < snakeLength; i++) { if (snakeX[0] == snakeX[i] && snakeY[0] == snakeY[i]) { gameOver = true; }

}

}

### How to Play

1. Upload the code to your Arduino.
2. Open the Serial Monitor (set to 9600 baud).
3. Use the following keys to control the snake:
   * w - Up
   * d - Right
   * s - Down
   * a - Left

### Conclusion

This code provides a basic implementation of the Snake game on an OLED display using Arduino. You can expand on this by adding features like score tracking, levels, or improved graphics. Enjoy coding and playing!