Pong recreation

PONG Remastered

Introducere

Proiectul ales consta in reproducerea jocului Pong prin intermediul a 2 joystickuri pentru controale , a unui buzzer pentru a semnaliza marcarea unui punct.

Descriere generală

Jucatorii sunt reprezentati prin 2 linii trase pe orizontala pe ecran, aproape lipite de margini Mingea este reprezentata de un pixel de culoare alba care se misca pe ecran Fiecare jucator isi controleaza linia prin cate un joystick La inscrierea unui punct, buzzerul scoate un zgomot scurt

Hardware Design

*Arduino UNO *breadboard * 2 joystickuri * ecarn OLED * buzzer 
arduino_pong.pdf

Software Design

#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 16, 2); *Set the LCD address to 0x27 for a 16×2 display*
——————————————PINS——————————————————

const int Player_1_moveButton = A0; *Joystick Pin for Player 1 const int Player_2_moveButton = A2;* Joystick Pin for Player 2 const int ScoreBuzzerPin = 9;

*——————————————CLASSES—————————————— class Paddle { public: int Paddle_X; int Paddle_Y; int Paddle_Length; Paddle(int x, int y, int length) { Paddle_X = x; Paddle_Y = y; Paddle_Length = length; } void PrintPaddle() { for (int i = 0; i < Paddle_Length; i++) { lcd.setCursor(Paddle_X, Paddle_Y + i); lcd.write(byte(0)); } } void ClearPaddle() { for (int i = 0; i < Paddle_Length; i++) { lcd.setCursor(Paddle_X, Paddle_Y + i); lcd.print(" "); } } void MovePaddleUp() { if (Paddle_Y > 0) { ClearPaddle(); Paddle_Y–; PrintPaddle(); } } void MovePaddleDown() { if (Paddle_Y < 1) { ClearPaddle(); Paddle_Y++; PrintPaddle(); } } }; class Ball { public: int Ball_X; int Ball_Y; int Ball_Velocity_X; int Ball_Velocity_Y; Ball(int x, int y, int velocityX, int velocityY) { Ball_X = x; Ball_Y = y; Ball_Velocity_X = velocityX; Ball_Velocity_Y = velocityY; } void PrintBall() { lcd.setCursor(Ball_X, Ball_Y); lcd.write(byte(1)); } void ClearBall() { lcd.setCursor(Ball_X, Ball_Y); lcd.print(" "); } void moveDiagonal() { ClearBall(); Ball_X += Ball_Velocity_X; Ball_Y += Ball_Velocity_Y; PrintBall(); if (Ball_Y == 0 || Ball_Y == 1) { Ball_Velocity_Y *= -1; } } };*
——————————————OBJECTS——————————————————

Paddle player1(0, 0, 2); *Creating Object for Player 1 Paddle Paddle player2(0, 1, 2);* Creating Object for Player 2 Paddle Ball ball(7, 1, 1, 1); *Creating Object for Ball*
——————————————SETUP———————————————————

void setup() {

```
lcd.init();              // Initialize the LCD
lcd.backlight();       // Turn on the backlight
```

```
pinMode(Player_1_moveButton, INPUT);
pinMode(Player_2_moveButton, INPUT);
pinMode(ScoreBuzzerPin, OUTPUT);
```

```
player1.PrintPaddle();
player2.PrintPaddle();
ball.PrintBall();
```

```
ShowStartingScreen();
```

}

——————————————SCOREBOARD—————————————— void PrintScoreboard(int player1Score, int player2Score) { lcd.clear(); lcd.setCursor(0, 0); lcd.print("P1: "); lcd.print(player1Score); lcd.setCursor(6, 0); lcd.print("P2: "); lcd.print(player2Score); delay(500); Display the scoreboard for 0.5 seconds

```
lcd.clear();
```

}

——————————————SCREENS—————————————— void ShowStartingScreen() {

lcd.clear(); lcd.setCursor(3, 0); lcd.print(“Pong Game”); delay(2000); lcd.clear(); } void ShowGameOverScreen() { lcd.clear(); lcd.setCursor(4, 0); lcd.print(“Game Over”); delay(2000); lcd.clear(); } void PlayScoreSound() { tone(ScoreBuzzerPin, 1000, 200); Play a sound for 200ms at a frequency of 1000Hz

```
delay(200);  // Wait for 200ms to avoid overlapping sounds

}
```

————————————————LOOP———————————————— void loop() { Player 1 controls

```
if (analogRead(Player_1_moveButton) < 500)
{
  player1.MovePaddleUp();
}
else if (analogRead(Player_1_moveButton) > 1500)
{
  player1.MovePaddleDown();
}
```

```
// Player 2 controls
if (analogRead(Player_2_moveButton) < 500)
{
  player2.MovePaddleUp();
}
else if (analogRead(Player_2_moveButton) > 1500)
{
  player2.MovePaddleDown();
}
```

```
// Ball movement
ball.moveDiagonal();
```

```
// Check if a point is scored
static int player1Score = 0;
static int player2Score = 0;
```

```
if (ball.Ball_X == 15)  // Replace with actual condition for player 1
scoring
{
  player1Score++;
  PrintScoreboard(player1Score, player2Score);
```

```
  if (player1Score >= 8 || player2Score >= 8)
  {

    ShowGameOverScreen();
    player1Score = 0;
    player2Score = 0;
    ball.ClearBall();
```

```
    ball = Ball(7, 1, 1, 1);  // Reset the ball position
    ball.PrintBall();
    ShowStartingScreen();
  }
  else
  {
    PlayScoreSound();
    ball.ClearBall();
    ball = Ball(7, 1, 1, 1);  // Reset the ball position
    ball.PrintBall();
  }
}
else if (ball.Ball_X == 0)  // Replace with actual condition for player 2
scoring
{
  player2Score++;
  PrintScoreboard(player1Score, player2Score);
```

```
  if (player1Score >= 8 || player2Score >= 8)
  {
    PlayScoreSound();
    ShowGameOverScreen();
    player1Score = 0;
    player2Score = 0;
    ball.ClearBall();
    ball = Ball(7, 1, 1, 1);  // Reset the ball position
    ball.PrintBall();
    ShowStartingScreen();
  }
  else
  {
    ball.ClearBall();
    ball = Ball(7, 1, 1, 1);  // Reset the ball position
    ball.PrintBall();
  }
}
```

```
// Delay for smoother gameplay
delay(150);
```

```
}
```

Rezultate Obţinute

WIP

Concluzii

Download

O arhivă (sau mai multe dacă este cazul) cu fişierele obţinute în urma realizării proiectului: surse, scheme, etc. Un fişier README, un ChangeLog, un script de compilare şi copiere automată pe uC crează întotdeauna o impresie bună 😉.

Fişierele se încarcă pe wiki folosind facilitatea **Add Images or other files**. Namespace-ul în care se încarcă fişierele este de tipul **:pm:prj20??:c?** sau **:pm:prj20??:c?:nume_student** (dacă este cazul). **Exemplu:** Dumitru Alin, 331CC → **:pm:prj2009:cc:dumitru_alin**.

Jurnal

WIP

Bibliografie/Resurse

WIP

Export to PDF