# ARQUITETURA DE COMPUTADORES

Avaliação Prática  
Arduino

# Giovanni Carmona Prudencio

#include <TVout.h>

#include <fontALL.h>

// Consts

const int ScreenTTX = 120;

const int ScreenTTY = 90;

const int TTX = 120;

const int TTY = 81;

const int BALLTX = 4;

const int BALLTY = 4;

const int TRACKETX = 2;

const int TRACKETY = 25;

const int Tdelay = 30;

const int SoundPIN = 2;

const int PauseLEDPIN = 13;

const int InGameLEDPIN = 12;

const int PowerLEDPIN = 10;

// Facilitator Components

typedef struct

{

    int X, Y;

    int SizeOfX, SizeOfY;

} XY;

enum EBallMovimentType{ UP\_RIGHT, DOWN\_RIGHT, DOWN\_LEFT, UP\_LEFT };

volatile bool Pause = 0;

int BallMovimentType;

int MovimentFactorX;

int MovimentFactorY;

int P1Lifes;

int P2Lifes;

int RoundWinner;

XY Ball;

XY P1Racket;

XY P2Racket;

TVout TV;

/\* --------------------------------------------------------------------------- \*/

void MakeObj ( XY Obj , int Erase = 0 )

{

    int i,j;

    for ( i = Obj.X ; i < ( Obj.X + Obj.SizeOfX ) ; i++ )

        for ( j = Obj.Y ; j < ( Obj.Y + Obj.SizeOfY ) ; j++ )

            if (Erase)

                TV.set\_pixel (i,j, BLACK );

            else

                TV.set\_pixel (i,j, WHITE );

}

void MakeRoundWinner ( int pWinner )

{

    RoundWinner = pWinner;

    MakeObj ( Ball , 1 );

}

void SetPause ()

{

    Pause = !Pause;

}

void CheckPause()

{

    bool InternalPause = Pause;

    if (InternalPause)

    {

        digitalWrite(PauseLEDPIN , HIGH);

        digitalWrite(InGameLEDPIN, LOW);

        TV.select\_font(font8x8);

        while (InternalPause)

        {

            TV.clear\_screen();

            TV.delay(500);

            TV.print(( TTX / 2 ) - 25 ,( TTY / 2 ) - 5,"PAUSE");

            TV.delay(500);

            InternalPause = Pause;

        }

        TV.clear\_screen();

        DrawHeader ();

        digitalWrite(PauseLEDPIN , LOW);

        digitalWrite(InGameLEDPIN, HIGH);

    }

}

void Beep ( int pSoundType = 0 )

{

    digitalWrite(SoundPIN, HIGH);

    TV.delay((pSoundType%2));

    digitalWrite(SoundPIN, LOW);

}

void SetBallFirstPosition ()

{

    Ball.X = ( ( TTX / 2 ) - ( BALLTX / 2 ) );

    Ball.Y = ( ( TTY / 2 ) - ( BALLTY / 2 ) );

    MakeObj ( Ball );

}

void SetBallMoviment ()

{

    int i ,j;

    XY xBall;

    RoundWinner = 0;

    xBall.X       = Ball.X;

    xBall.Y       = Ball.Y;

    xBall.SizeOfX = Ball.SizeOfX;

    xBall.SizeOfY = Ball.SizeOfY;

    switch (BallMovimentType)

    {

        case UP\_RIGHT:

            xBall.Y -= MovimentFactorY; // Up

            xBall.X += MovimentFactorX; // Right

            if ( ( xBall.Y ) <= 0 )

            {

                BallMovimentType = DOWN\_RIGHT;

                Beep (1);

            }

            else if ( ( xBall.X + xBall.SizeOfX ) >= P2Racket.X )

            {

                if ( ( ( ( xBall.Y + xBall.SizeOfY ) >= P2Racket.Y ) ) && ( ( xBall.Y ) <= ( P2Racket.Y + P2Racket.SizeOfY ) ) )

                {

                    BallMovimentType = UP\_LEFT;

                    Beep ();

                }

                else

                {

                    MakeRoundWinner (1);

                    exit;

                }

            }

        break;

        case DOWN\_RIGHT:

            xBall.Y += MovimentFactorY; // Down

            xBall.X += MovimentFactorX; // Right

            if ( ( xBall.Y + xBall.SizeOfY ) >= TTY )

            {

                BallMovimentType = UP\_RIGHT;

                Beep (1);

            }

            else if ( ( xBall.X + xBall.SizeOfX ) >= P2Racket.X )

            {

                if ( ( ( ( xBall.Y + xBall.SizeOfY ) >= P2Racket.Y ) ) && ( ( xBall.Y ) <= ( P2Racket.Y + P2Racket.SizeOfY ) ) )

                {

                    BallMovimentType = DOWN\_LEFT;

                    Beep ();

                }

                else

                {

                    MakeRoundWinner (1);

                    exit;

                }

            }

        break;

        case DOWN\_LEFT:

            xBall.Y += MovimentFactorY; // Down

            xBall.X -= MovimentFactorX; // Left

            if ( ( xBall.Y + xBall.SizeOfY ) >= TTY )

            {

                BallMovimentType = UP\_LEFT;

                Beep (1);

            }

            else if ( xBall.X <= ( P1Racket.X + P1Racket.SizeOfX ) )

            {

                if ( ( ( ( xBall.Y + xBall.SizeOfY ) >= P1Racket.Y ) ) && ( ( xBall.Y ) <= ( P1Racket.Y + P1Racket.SizeOfY ) ) )

                {

                    BallMovimentType = DOWN\_RIGHT;

                    Beep ();

                }

                else

                {

                    MakeRoundWinner (2);

                    exit;

                }

            }

        break;

        case UP\_LEFT:

            xBall.Y -= MovimentFactorY; // Up

            xBall.X -= MovimentFactorX; // Left

            if ( ( xBall.Y ) <= 0 )

            {

                BallMovimentType = DOWN\_LEFT;

                Beep (1);

            }

            else if ( xBall.X <= ( P1Racket.X + P1Racket.SizeOfX ) )

            {

                if ( ( ( ( xBall.Y + xBall.SizeOfY ) >= P1Racket.Y ) ) && ( ( xBall.Y ) <= ( P1Racket.Y + P1Racket.SizeOfY ) ) )

                {

                    BallMovimentType = UP\_RIGHT;

                    Beep ();

                }

                else

                {

                    MakeRoundWinner (2);

                    exit;

                }

            }

        break;

    }

    MakeObj ( Ball , 1 );

    Ball.X          = xBall.X;

    Ball.Y          = xBall.Y;

    Ball.SizeOfX = xBall.SizeOfX;

    Ball.SizeOfY = xBall.SizeOfY;

    MakeObj ( Ball );

    TV.delay ( Tdelay );

}

void SetRacketsFirstPosition ()

{

    P1Racket.X = 0;

    P1Racket.Y = ( ( TTY / 2 ) - ( TRACKETY / 2 ) );

    MakeObj ( P1Racket );

    P2Racket.X = TTX - TRACKETX;

    P2Racket.Y = ( ( TTY / 2 ) - ( TRACKETY / 2 ) );

    MakeObj ( P2Racket );

}

int ReadRacket ( int pPin )

{

    int ReadValue;

    ReadValue = analogRead( pPin );

    return map(ReadValue, 0, 1023, 0, ( TTY - TRACKETY ));

}

void SetRacketMoviment ()

{

    MakeObj ( P1Racket, 1 );

    P1Racket.Y = ReadRacket ( A0 );

    MakeObj ( P1Racket );

    MakeObj ( P2Racket, 1 );

    P2Racket.Y = ReadRacket ( A1 );

    MakeObj ( P2Racket );

}

void DrawWinner ( int pWinner )

{

    digitalWrite(InGameLEDPIN, LOW);

    TV.select\_font(font8x8);

    for ( int i = 0 ; i < 5 ; i++ )

    {

        TV.clear\_screen();

        TV.delay(500);

        TV.println(( TTX / 2 ) - 30 ,( TTY / 2 ) - 15,"Winner");

        TV.print(( TTX / 2 ) - 35 ,( TTY / 2 ) - 5,"Player ");

        TV.print(pWinner);

        TV.delay(500);

    }

    BeginGame();

}

void CheckWinner ()

{

    if (RoundWinner)

    {

        switch (RoundWinner)

        {

            case 1:

                P2Lifes--;

            break;

            case 2:

                P1Lifes--;

            break;

        }

        TV.delay (500);

        if (P1Lifes == 0)

            DrawWinner (2);

        else if (P2Lifes == 0)

            DrawWinner (1);

        else

            Start ();

    }

}

void Start ()

{

    TV.clear\_screen();

    DrawHeader ();

    SetRacketsFirstPosition();

    SetBallFirstPosition ();

    TV.delay ( 1000 );

    BallMovimentType = (BallMovimentType + 1)%4;

}

void BeginGame ()

{

    P1Lifes = 5;

    P2Lifes = 5;

    BallMovimentType = -1;

    MovimentFactorX = 2;

    MovimentFactorY = 1;

    DrawWelcome ();

    Start ();

    digitalWrite(InGameLEDPIN, HIGH);

}

void DrawHeader ()

{

    for ( int i = 0; i <= ScreenTTX; i++ )

        TV.set\_pixel (i,( TTY + 1 ), WHITE );

    TV.select\_font(font4x6);

    TV.print( 2 , ( TTY + ( ( ScreenTTY - ( TTY ) ) / 2 ) ) ,"P1 = ");

    for ( int i = 1; i <= P1Lifes; i++ )

        TV.print( "x" );

    TV.print( ( ( ScreenTTX / 2 ) + 2 ) , ( TTY + ( ( ScreenTTY - ( TTY ) ) / 2 ) ) ,"P2 = ");

    for ( int i = 1; i <= P2Lifes; i++ )

        TV.print( "x" );

}

void DrawWelcome ()

{

    TV.select\_font(font8x8);

    TV.clear\_screen();

    TV.println(( TTX / 2 ) - 12 ,( TTY / 2 ) - 5,"Are");

    TV.delay(750);

    TV.clear\_screen();

    TV.println(( TTX / 2 ) - 12 ,( TTY / 2 ) - 5,"You");

    TV.delay(750);

    TV.clear\_screen();

    TV.println(( TTX / 2 ) - 32 ,( TTY / 2 ) - 5,"Ready???");

    TV.delay(2000);

    TV.clear\_screen();

    TV.println(( TTX / 2 ) - 8 ,( TTY / 2 ) - 5,"3");

    TV.delay(1000);

    TV.clear\_screen();

    TV.println(( TTX / 2 ) - 8 ,( TTY / 2 ) - 5,"2");

    TV.delay(1000);

    TV.clear\_screen();

    TV.println(( TTX / 2 ) - 8 ,( TTY / 2 ) - 5,"1");

    TV.delay(1000);

    for (int i = 0; i<5; i++)

    {

        TV.clear\_screen();

        TV.delay(250);

        TV.println(( TTX / 2 ) - 28 ,( TTY / 2 ) - 5,"Pong!!!");

        TV.delay(250);

    }

}

void **setup** ()

{

    TV.begin(NTSC,ScreenTTX,ScreenTTY);

    pinMode (SoundPIN, OUTPUT);

    pinMode (PauseLEDPIN, OUTPUT);

    pinMode (InGameLEDPIN, OUTPUT);

    pinMode (PowerLEDPIN, OUTPUT);

    digitalWrite(PowerLEDPIN, HIGH);

    Ball.SizeOfX     = BALLTX;

    Ball.SizeOfY     = BALLTY;

    P1Racket.SizeOfX = P2Racket.SizeOfX = TRACKETX;

    P1Racket.SizeOfY = P2Racket.SizeOfY = TRACKETY;

    TV.select\_font(font8x8);

    TV.clear\_screen();

    TV.println(( TTX / 2 ) - 36 ,( TTY / 2 ) - 5,"PongDuino");

    TV.delay(2500);

    TV.println(( TTX / 2 ) - 36 ,( TTY / 2 ) + 5,"V 3.0.2.4");

    TV.delay(2500);

    BeginGame ();

    attachInterrupt(1, SetPause, RISING );

}

void **loop** ()

{

    SetRacketMoviment ();

    SetBallMoviment ();

    CheckWinner ();

    CheckPause();

}