**פרוייקט ביסודות מדעי המחשב באמצעות מחלקת MTP – אופיר הופמן י3**

**PONG**

**מחלקת הכדור**

class Ball

{

private int x;

private int y;

private int maxX = 79;

private int maxY = 26;

private char ch = 'O';

// Directions

enum Direction { UpRight, DownLeft, DownRight, UpLeft};

private Direction direction;

//Constructor

public Ball(int direction)

{

this.direction = (Direction)direction;

this.x = 40;

this.y = 12;

}

public int GetY()

{

return this.y;

}

public int GetX()

{

return this.x;

}

public void Draw()

{

Console.SetCursorPosition(this.x, this.y);

Console.BackgroundColor = ConsoleColor.Blue;

Console.Write(ch);

}

public void Undraw()

{

Console.SetCursorPosition(this.x, this.y);

Console.BackgroundColor = ConsoleColor.Black;

Console.ForegroundColor = ConsoleColor.Black;

Console.Write(" ");

}

// change direction for when hits the borders up or down

public void ChangeDirectionY()

{

this.direction = (Direction)(((int)this.direction + 2) % 4);

}

// change direction for when hits the left or right borders

public void ChangeDirectionX()

{

if ((int)this.direction == 0)

this.direction = (Direction)3;

else if ((int)this.direction == 1)

this.direction = (Direction)2;

else if ((int)this.direction == 2)

this.direction = (Direction)1;

else if ((int)this.direction == 3)

this.direction = (Direction)0;

}

// Move ball

public bool Move(bool b)

{

// parameter if ball about to hit a tile

bool hitTile = b;

if (this.direction == Direction.UpRight)

{

//if about to hit a tile

if (hitTile == true)

{

this.ChangeDirectionX();

Move(false);

return true;

}

// if ball not about to hit anything

else if ((this.y - 1 >= 0) && (this.x + 1 <= maxX))

{

Undraw();

this.y--;

this.x++;

Draw();

return true;

}

//if about to hit up or down borders

else if (this.y - 1 < 0)

{

this.ChangeDirectionY();

Move(false);

return true;

}

else

return false;

}

else if (this.direction == Direction.UpLeft)

{

if (hitTile == true)

{

ChangeDirectionX();

Move(false);

return true;

}

else if ((this.y - 1 >= 0) && (this.x - 1 >= 5))

{

Undraw();

this.y--;

this.x--;

Draw();

return true;

}

else if (this.y - 1 < 0)

{

this.ChangeDirectionY();

Move(false);

return true;

}

else

return false;

}

else if (this.direction == Direction.DownRight)

{

if (hitTile == true)

{

this.ChangeDirectionX();

Move(false);

return true;

}

else if ((this.y+1 <= maxY) && (this.x < maxX))

{

Undraw();

this.x++;

this.y++;

Draw();

return true;

}

else if (this.y + 1 > maxY)

{

this.ChangeDirectionY();

Move(false);

return true;

}

else

return false;

}

else

{

if (hitTile == true)

{

this.ChangeDirectionX();

Move(false);

return true;

}

else if ((this.y + 1 <= maxY) && (this.x - 1 >= 5))

{

Undraw();

this.x--;

this.y++;

Draw();

return true;

}

else if (this.y + 1 > maxY)

{

this.ChangeDirectionY();

Move(false);

return true;

}

else

return false;

}

}

}

**מחלקת המחבטים**

class Tiles

{

private int side; //1-right, 2-left

private int x;

private int y;

private int height;

private int maxY = 25;

public Tiles(int side)

{

this.side = side;

this.height = 7;

this.y = 12;

if(this.side == 1)

{

this.x = 80;

}

else if(this.side == 2)

{

this.x = 5;

}

}

public void Draw()

{

int GibuiY = this.y-3;

Console.BackgroundColor = ConsoleColor.White;

Console.ForegroundColor = ConsoleColor.White;

for (int h = 1; h <= this.height; h++)

{

Console.SetCursorPosition(this.x, GibuiY);

Console.Write("█");

GibuiY++;

}

}

public void Undraw()

{

int GibuiY = this.y - 3;

Console.BackgroundColor = ConsoleColor.Black;

Console.ForegroundColor = ConsoleColor.Black;

for (int h = 1; h <= this.height; h++)

{

Console.SetCursorPosition(this.x, GibuiY);

Console.Write(" ");

GibuiY++;

}

}

public int GetY()

{

return this.y;

}

public int GetX()

{

return this.x;

}

public void MoveUp()

{

if (this.y-4 >= 0)

{

Undraw();

this.y--;

Draw();

}

}

public void MoveDown()

{

if (this.y+4 <= maxY)

{

Undraw();

this.y++;

Draw();

}

}

}

**המשך למטה**

**Program.cs**

internal class Program

{

public static void SetLevel(int level)

{

if (level == 1)

{

Console.BackgroundColor = ConsoleColor.Blue;

Console.ForegroundColor = ConsoleColor.Black;

Console.SetCursorPosition(70, 10);

Console.WriteLine("Level 1 (easy peasy)");

}

else if (level == 2)

{

Console.BackgroundColor = ConsoleColor.Blue;

Console.ForegroundColor = ConsoleColor.Black;

Console.SetCursorPosition(70, 12);

Console.WriteLine("Level 2 (a bit harder)");

}

else if (level == 3)

{

Console.BackgroundColor = ConsoleColor.Blue;

Console.ForegroundColor = ConsoleColor.Black;

Console.SetCursorPosition(70, 14);

Console.WriteLine("Level 3 (extreme!!!)");

}

}

public static void UnsetLevel(int level)

{

if (level == 1)

{

Console.BackgroundColor = ConsoleColor.Black;

Console.ForegroundColor = ConsoleColor.Blue;

Console.SetCursorPosition(70, 10);

Console.WriteLine("Level 1 (easy peasy)");

}

else if (level == 2)

{

Console.BackgroundColor = ConsoleColor.Black;

Console.ForegroundColor = ConsoleColor.Blue;

Console.SetCursorPosition(70, 12);

Console.WriteLine("Level 2 (a bit harder)");

}

else if (level == 3)

{

Console.BackgroundColor = ConsoleColor.Black;

Console.ForegroundColor = ConsoleColor.Blue;

Console.SetCursorPosition(70, 14);

Console.WriteLine("Level 3 (extreme!!!)");

}

}

public static bool LoseOnPurpose(int probPracentage)

{

Random rnd = new Random();

int loseProb = rnd.Next(1, probPracentage+1);

if (loseProb == 1)

{

return true;

}

return false;

}

static void Main(string[] args)

{

Console.CursorVisible = false;

Console.ForegroundColor = ConsoleColor.Blue;

Console.SetCursorPosition(0, 4);

Thread.Sleep(100);

Console.WriteLine("\r\n██████╗░░█████╗░███╗░░██╗░██████╗░" +

"\r\n██╔══██╗██╔══██╗████╗░██║██╔════╝░" +

"\r\n██████╔╝██║░░██║██╔██╗██║██║░░██╗░" +

"\r\n██╔═══╝░██║░░██║██║╚████║██║░░╚██╗" +

"\r\n██║░░░░░╚█████╔╝██║░╚███║╚██████╔╝" +

"\r\n╚═╝░░░░░░╚════╝░╚═╝░░╚══╝░╚═════╝░");

Console.Beep();

Thread.Sleep(1000);

Console.WriteLine("\r\n██████╗░██╗░░░██╗" +

"\r\n██╔══██╗╚██╗░██╔╝" +

"\r\n██████╦╝░╚████╔╝░" +

"\r\n██╔══██╗░░╚██╔╝░░" +

"\r\n██████╦╝░░░██║░░░" +

"\r\n╚═════╝░░░░╚═╝░░░");

Console.Beep();

Thread.Sleep(1000);

Console.WriteLine("\r\n░█████╗░██████╗░██╗░░██╗██╗██████╗░  ░░██╗  ██╗" +

"\r\n██╔══██╗██╔══██╗██║░░██║██║██╔══██╗  ░██╔╝  ╚═╝" +

"\r\n██║░░██║██████╔╝███████║██║██████╔╝  ██╔╝░  ░░░" +

"\r\n██║░░██║██╔═══╝░██╔══██║██║██╔══██╗  ╚██╗░  ░░░" +

"\r\n╚█████╔╝██║░░░░░██║░░██║██║██║░░██║  ░╚██╗  ██╗" +

"\r\n░╚════╝░╚═╝░░░░░╚═╝░░╚═╝╚═╝╚═╝░░╚═╝  ░░╚═╝  ╚═╝");

Console.Beep();

Thread.Sleep(1000);

Console.BackgroundColor = ConsoleColor.Blue;

Console.ForegroundColor = ConsoleColor.Black;

Console.SetCursorPosition(70, 10);

Console.WriteLine("Level 1 (easy peasy)");

Console.BackgroundColor = ConsoleColor.Black;

Console.ForegroundColor = ConsoleColor.Blue;

Console.SetCursorPosition(70, 12);

Console.WriteLine("Level 2 (a bit harder)");

Console.SetCursorPosition(70, 14);

Console.WriteLine("Level 3 (extreme!!!)");

bool Choosed = false;

int markedLevel = 1;

while (!Choosed)

{

SetLevel(markedLevel);

ConsoleKeyInfo k = Console.ReadKey();

if (k.Key == ConsoleKey.DownArrow)

{

if (markedLevel != 3)

{

UnsetLevel(markedLevel);

markedLevel++;

SetLevel(markedLevel);

}

}

if (k.Key == ConsoleKey.UpArrow)

{

if (markedLevel != 1)

{

UnsetLevel(markedLevel);

markedLevel--;

SetLevel(markedLevel);

}

}

if (k.Key == ConsoleKey.Enter || k.Key == ConsoleKey.Spacebar)

{

Choosed = true;

Console.BackgroundColor = ConsoleColor.Black;

Console.ForegroundColor = ConsoleColor.White;

}

}

Console.Clear();

Console.SetCursorPosition(0, 27);

// Draw border

for (int i = 1; i <= 80; i++)

{

Console.Write("-");

}

Random rnd = new Random();

// Left Tile - PC

Tiles left\_Tile = new Tiles(2);

left\_Tile.Draw();

// Right Tile - User

Tiles right\_Tile = new Tiles(1);

right\_Tile.Draw();

int randomDirection = rnd.Next(0, 3);

Ball ball = new Ball(randomDirection);

ball.Draw();

bool cont = true;

int side = 0; //1-right, 2-left

// Main Loop

while (cont)

{

if (ball.GetX() > 40)

side = 1;

else

side = 2;

ball.Undraw();

// Move Ball

if (((ball.GetX() < 7) && ((ball.GetY() < left\_Tile.GetY() + 3) && (ball.GetY() > left\_Tile.GetY() - 3))))

ball.Move(true);//true if ball is about to hit the tile

else if (((ball.GetX() > 77) && (ball.GetY() >= right\_Tile.GetY() - 3) && (ball.GetY() <= right\_Tile.GetY() + 3)))

ball.Move(true);

else

ball.Move(false);

//check if ball is out

if (ball.Move(false) == false)

{

cont = false;

}

right\_Tile.Draw();

// move pc's tile according to ball

if (markedLevel == 1)

{

// Make pc lose on purpose according to level

if (LoseOnPurpose(13) == true)

{

if (left\_Tile.GetY() > ball.GetY())

{

left\_Tile.MoveDown();

left\_Tile.MoveDown();

}

else if (left\_Tile.GetY() < ball.GetY())

{

left\_Tile.MoveUp();

left\_Tile.MoveUp();

}

}

else

{

if (left\_Tile.GetY() > ball.GetY())

{

left\_Tile.MoveUp();

left\_Tile.MoveUp();

}

else if (left\_Tile.GetY() < ball.GetY())

{

left\_Tile.MoveDown();

left\_Tile.MoveDown();

}

}

}

else if (markedLevel == 2)

{

if (LoseOnPurpose(16) == true)

{

if (left\_Tile.GetY() > ball.GetY())

{

left\_Tile.MoveDown();

left\_Tile.MoveDown();

}

else if (left\_Tile.GetY() < ball.GetY())

{

left\_Tile.MoveUp();

left\_Tile.MoveUp();

}

}

else

{

if (left\_Tile.GetY() > ball.GetY())

{

left\_Tile.MoveUp();

left\_Tile.MoveUp();

}

else if (left\_Tile.GetY() < ball.GetY())

{

left\_Tile.MoveDown();

left\_Tile.MoveDown();

}

}

}

else if (markedLevel == 3)

{

if (left\_Tile.GetY() > ball.GetY())

{

left\_Tile.MoveUp();

left\_Tile.MoveUp();

}

else if (left\_Tile.GetY() < ball.GetY())

{

left\_Tile.MoveDown();

left\_Tile.MoveDown();

}

}

//move user's tile

if (Console.KeyAvailable)

{

ConsoleKeyInfo k = Console.ReadKey(true);

if (k.Key == ConsoleKey.UpArrow)

{

right\_Tile.MoveUp();

right\_Tile.MoveUp();

}

else if (k.Key == ConsoleKey.DownArrow)

{

right\_Tile.MoveDown();

right\_Tile.MoveDown();

}

}

ball.Draw();

//Set speed according to level

int speed;

if (markedLevel == 1)

{

speed = 170;

}

else if (markedLevel == 2)

{

speed = 150;

}

else

{

speed = 120;

}

Thread.Sleep(speed);

}

Console.BackgroundColor = ConsoleColor.Black;

Console.ForegroundColor = ConsoleColor.Red;

Console.Clear();

Console.SetCursorPosition(0, 4);

// check if winner and print a message

if (side == 1)

{

Console.Write("\r\n██╗░░░██╗░█████╗░██╗░░░██╗  ██╗░░░░░░█████╗░░██████╗████████╗  ██╗░░  ██╗" +

"\r\n╚██╗░██╔╝██╔══██╗██║░░░██║  ██║░░░░░██╔══██╗██╔════╝╚══██╔══╝  ╚██╗░  ╚═╝" +

"\r\n░╚████╔╝░██║░░██║██║░░░██║  ██║░░░░░██║░░██║╚█████╗░░░░██║░░░  ░╚██╗  ░░░" +

"\r\n░░╚██╔╝░░██║░░██║██║░░░██║  ██║░░░░░██║░░██║░╚═══██╗░░░██║░░░  ░██╔╝  ░░░" +

"\r\n░░░██║░░░╚█████╔╝╚██████╔╝  ███████╗╚█████╔╝██████╔╝░░░██║░░░  ██╔╝░  ██╗" +

"\r\n░░░╚═╝░░░░╚════╝░░╚═════╝░  ╚══════╝░╚════╝░╚═════╝░░░░╚═╝░░░  ╚═╝░░  ╚═╝");

}

else if (side == 2)

{

Console.Write("\r\n██╗░░░██╗░█████╗░██╗░░░██╗  ░██╗░░░░░░░██╗░█████╗░███╗░░██╗██╗██╗██╗" +

"\r\n╚██╗░██╔╝██╔══██╗██║░░░██║  ░██║░░██╗░░██║██╔══██╗████╗░██║██║██║██║" +

"\r\n░╚████╔╝░██║░░██║██║░░░██║  ░╚██╗████╗██╔╝██║░░██║██╔██╗██║██║██║██║" +

"\r\n░░╚██╔╝░░██║░░██║██║░░░██║  ░░████╔═████║░██║░░██║██║╚████║╚═╝╚═╝╚═╝" +

"\r\n░░░██║░░░╚█████╔╝╚██████╔╝  ░░╚██╔╝░╚██╔╝░╚█████╔╝██║░╚███║██╗██╗██╗" +

"\r\n░░░╚═╝░░░░╚════╝░░╚═════╝░  ░░░╚═╝░░░╚═╝░░░╚════╝░╚═╝░░╚══╝╚═╝╚═╝╚═╝");

}

Console.ReadKey();

Console.ReadKey();

}

}