namespace ConsoleMinigame

{

internal class Program

{

protected static int origRow;

protected static int origCol;

protected static void WriteAt(string s, int x, int y)

{

try

{

Console.SetCursorPosition(origCol + x, origRow + y);

Console.Write(s);

}

catch (ArgumentOutOfRangeException e)

{

//Console.Clear();

Console.WriteLine("YOUR OUT OF BOUNDS HACKER >:(");

}

}

static void Main(string[] args)

{

//before sarting the game equivilent of void start

int size = 10;

int[,] grid = new int[size, size];

Player p1 = new Player();

GameBuilder(p1);

BuildBorder(grid, size);

//Void update : runs every frame

while (true)

{

Movement\_input(grid, p1);

}

}

static void GameBuilder(object instance)

{

Player player = (Player)instance;

int Y\_off = player.Y\_off;

int X\_off = player.X\_off;

int y\_size = 10;

int x\_size = y\_size \* 2;

//WriteAt("██", j, i);

for (int i = 0; i < y\_size; i++)

{

if (i == 0 || i == y\_size - 1)

{

for (int j = 0; j < x\_size; j += 2)

{

WriteAt("[]", j + X\_off, i + Y\_off);

}

}

else

{

for (int j = 0; j < x\_size; j += 2)

{

if (j == 0 || j == x\_size - 2)

{

WriteAt("[]", j + X\_off, i + Y\_off);

}

}

}

}

}

static void Movement\_input(int[,] grid, object instance)

{

Player player = (Player)instance;

var input = Console.ReadKey().Key;

int X\_off = player.X\_off;

int y\_off = player.Y\_off;

WriteAt(" ", player.x + X\_off, player.y + y\_off);

grid[player.x / 2, player.y] = 0;

switch (input)

{

case ConsoleKey.W:

if(grid[player.x / 2, player.y--] == 1)

{ player.y++; }

player.y--;

break;

case ConsoleKey.A:

if (grid[(player.x / 2)-2, player.y] == 1)

{ player.x += 2; }

player.x -= 2;

break;

case ConsoleKey.S:

if (grid[player.x / 2, player.y++] == 1)

{ player.y--; }

player.y++;

break;

case ConsoleKey.D:

if (grid[player.x / 2+2, player.y] == 1)

{ player.x -= 2; }

player.x += 2;

break;

}

try

{

grid[player.x / 2, player.y] = 1;

WriteAt("██", player.x + X\_off, player.y + y\_off);

}

catch

{

WriteAt("no", player.x + X\_off, player.y + y\_off);

grid[0, 0] = 1;

}

WriteAt(" ", 0, 0);

}

static void BuildBorder(int[,] grid, int size)

{

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

{

if (i == 0 || i == size - 1)

{

for (int j = 0; j < size; j++)

{

grid[i, j] = 1;

}

}

for (int j = 0; j < size; j++)

{

if (j == 0 || j == size - 1)

{

grid[i, j] = 1;

}

}

}

}

}

}