using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Threading.Tasks.Dataflow;

namespace ConsoleMinigame

{

internal class Program

{

ConsoleKeyInfo keyInfo = Console.ReadKey(false);

//class Colectables

//{

// //string[] fruits = { "lemon", "orange", "apple" };

//}

static void Main(string[] args)

{

int n = 15;

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

int x = 3, y = 4;

var input = ConsoleKey.G;

bool fruit\_eaten = true;

BuildBorder(grid, n);

while (true)

{

Thread.Sleep(400);

if (Console.KeyAvailable == true)

{

input = Console.ReadKey().Key;

}

while (fruit\_eaten == true)

{

fruit\_eaten = false;

fruit\_spawn(grid, n);

}

if (input == ConsoleKey.D)

{

grid[y, x] = 0;

x++;

}

if (input == ConsoleKey.S)

{

grid[y, x] = 0;

y++;

}

if (input == ConsoleKey.W)

{

grid[y, x] = 0;

y--;

}

if (input == ConsoleKey.A)

{

grid[y, x] = 0;

x--;

}

grid[y, x] = 2;

Refresh(grid, n);

}

}

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

{

Console.Clear();

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

{

Console.WriteLine();

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

{

if (grid[i, j] == 1 || grid[i, j] == 2)

{

Console.Write("██");

}

else if (grid[i, j] == 0)

{

Console.Write(" ");

}

else if (grid[i, j] == 10)

{

Console.Write("██");

}

}

}

}

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)

{

Console.WriteLine();

grid[i, j] = 1;

}

}

}

}

static void fruit\_spawn(int[,] grid, int size)

{

Random random = new Random();

int x, y = 0;

x = random.Next(1, size- 1);

y = random.Next(1, size - 1);

grid[x, y] = 10;

}

}

}