using System.Collections;

using System.Collections.Generic;

using UnityEngine;

//using static terainGenScript;

public class floor1GenScript : MonoBehaviour

{

//list lengt

//de hoeveelheid floer en muur tiles die in de lijst zitten

int f;

int w;

//object lists

//hier worden de objecten in geladen

public List<GameObject> floorTiles;

public List<GameObject> wallTiles;

public List<GameObject> floorTraps;

public List<GameObject> wallTraps;

public List<GameObject> enemie;

//GenGrid nummer dingen

//de breedte en hoogte van een block

[SerializeField]

public float tileHeight = 16;

[SerializeField]

public float tileWidth = 16;

//GenRoom nummer dingen

[SerializeField]

private bool genRooms = false;

[SerializeField]

//hoe vaak de code een kamer probeert te genereren

private int roomGenTrys = 10;

//de X en Z coordinaten van de kamer die wordt gegenereerd

private float roomZ;

private float roomX;

//de X en Z coordinaten van de kamer die daarvoor was gegenereerd

private float roomXold = 4;

private float roomZold = 4;

//startTile word niet meer gebruikt

//private bool startTile = false;

//private int RoomNummer = 0;

//hoe moeilijk de kamers zijn

[SerializeField]

private int difficulty = 0;

//GenGrid nummer dingen

private int[,] grid;

private pathfindingScript PathfindingScript;

// Start is called before the first frame update

void Awake()

{

if (genRooms == true)

{

//GenRoom functie

GenRoom();

}

}

public void GenRoom()

{

//laad een object

PathfindingScript = GameObject.FindObjectOfType<pathfindingScript>();

//laad objecten uit folders en doet ze in een lijst

floorTiles = new List<GameObject>(Resources.LoadAll<GameObject>("FloorTiles"));

wallTiles = new List<GameObject>(Resources.LoadAll<GameObject>("WallTiles"));

floorTraps = new List<GameObject>(Resources.LoadAll<GameObject>("FloorTraps"));

wallTraps = new List<GameObject>(Resources.LoadAll<GameObject>("WallTraps"));

enemie = new List<GameObject>(Resources.LoadAll<GameObject>("Enemies"));

f = floorTiles.Count;

w = wallTiles.Count;

//room types

//dit is een array met de locatie van muren en en vloeren in een kamer

int[,,] roomTypeVariation = new int[,,]

{

{

//room 0 wordt niet gebruikt

{1+f, 1+f, 1+f, 1+f, 1+f, 1+f, 1+f, 1 , 1 , 1+f, 1+f, 1+f, 1+f, 1+f, 1+f, 1+f}, //1

{1+f, 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1+f}, //2

{1+f, 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1+f}, //3

{1+f, 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1+f}, //4

{1+f, 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1+f}, //5

{1+f, 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1+f}, //6

{1+f, 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1+f}, //7

{1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 }, //8

{1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 }, //9

{1+f, 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1+f}, //10

{1+f, 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1+f}, //11

{1+f, 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1+f}, //12

{1+f, 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1+f}, //13

{1+f, 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1+f}, //14

{1+f, 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1+f}, //15

{1+f, 1+f, 1+f, 1+f, 1+f, +f , 1+f, 1 , 1 , 1+f, 1+f, 1+f, 1+f, 1+f, 1+f, 1+f} //16

},

//room 1 (spawn room)

{

{1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 }, //1

{1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 }, //2

{1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 }, //3

{1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 }, //4

{1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 }, //5

{1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 }, //6

{1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 }, //7

{1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 }, //8

{1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 }, //9

{1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 }, //10

{1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 }, //11

{1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 }, //12

{1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 }, //13

{1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 }, //14

{1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 }, //15

{1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 } //16

},

//room 2

{

{1+f, 1+f, 1+f, 1+f, 1+f, 1+f, 1+f, 1 , 1 , 1+f, 1+f, 1+f, 1+f, 1+f, 1+f, 1+f}, //1

{1+f, 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1+f}, //2

{1+f, 1 , 1+f, 1+f, 1+f, 1+f, 1+f, 1 , 1 , 1+f, 1+f, 1+f, 1+f, 1+f, 1 , 1+f}, //3

{1+f, 1 , 1+f, 2 , 2 , 2 , 1+f, 1 , 1 , 1+f, 2 , 2 , 2 , 1+f, 1 , 1+f}, //4

{1+f, 1 , 1+f, 2 , 2 , 2 , 2 , 1 , 1 , 2 , 2 , 2 , 2 , 1+f, 1 , 1+f}, //5

{1+f, 1 , 1+f, 2 , 2 , 2 , 2 , 1 , 1 , 2 , 2 , 2 , 2 , 1+f, 1 , 1+f}, //6

{1+f, 1 , 1+f, 2 , 2 , 2 , 1+f, 1 , 1 , 1+f, 2 , 2 , 2 , 1+f, 1 , 1+f}, //7

{1 , 1 , 2 , 2 , 2 , 2 , 1+f, 1 , 1 , 1+f, 2 , 2 , 2 , 2 , 1 , 1 }, //8

{1 , 1 , 2 , 2 , 2 , 2 , 1+f, 1 , 1 , 1+f, 2 , 2 , 2 , 2 , 1 , 1 }, //9

{1+f, 1 , 1+f, 2 , 2 , 2 , 1+f, 1 , 1 , 1+f, 2 , 2 , 2 , 1+f, 1 , 1+f}, //10

{1+f, 1 , 1+f, 2 , 2 , 2 , 2 , 1 , 1 , 2 , 2 , 2 , 2 , 1+f, 1 , 1+f}, //11

{1+f, 1 , 1+f, 2 , 2 , 2 , 2 , 1 , 1 , 2 , 2 , 2 , 2 , 1+f, 1 , 1+f}, //12

{1+f, 1 , 1+f, 2 , 2 , 2 , 1+f, 1 , 1 , 1+f, 2 , 2 , 2 , 1+f, 1 , 1+f}, //13

{1+f, 1 , 1+f, 1+f, 1+f, 1+f, 1+f, 1 , 1 , 1+f, 1+f, 1+f, 1+f, 1+f, 1 , 1+f}, //14

{1+f, 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1+f}, //15

{1+f, 1+f, 1+f, 1+f, 1+f, 1+f, 1+f, 1 , 1 , 1+f, 1+f, 1+f, 1+f, 1+f, 1+f, 1+f} //16

},

//room 3

{

{1+f, 1+f, 1+f, 1+f, 1+f, 1 , 1 , 1 , 1 , 1 , 1 , 1+f, 1+f, 1+f, 1+f, 1+f}, //1

{1+f, 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1+f}, //2

{1+f, 1 , 1+f, 1+f, 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1+f}, //3

{1+f, 1 , 1+f, 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1+f}, //4

{1+f, 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1+f}, //5

{1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 }, //6

{1 , 1 , 1 , 1 , 1 , 1 , 1+f, 1+f, 1+f, 1+f, 1 , 1 , 1 , 1 , 1 , 1 }, //7

{1 , 1 , 1 , 1 , 1 , 1 , 1+f, 1+f, 1+f, 1+f, 1 , 1 , 1 , 1 , 1 , 1 }, //8

{1 , 1 , 1 , 1 , 1 , 1 , 1+f, 1+f, 1+f, 1+f, 1 , 1 , 1 , 1 , 1 , 1 }, //9

{1 , 1 , 1 , 1 , 1 , 1 , 1+f, 1+f, 1+f, 1+f, 1 , 1 , 1 , 1 , 1 , 1 }, //10

{1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 }, //11

{1+f, 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1+f}, //12

{1+f, 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1+f, 1 , 1+f}, //13

{1+f, 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1+f, 1+f, 1 , 1+f}, //14

{1+f, 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1+f}, //15

{1+f, 1+f, 1+f, 1+f, 1+f, 1 , 1 , 1 , 1 , 1 , 1 , 1+f, 1+f, 1+f, 1+f, 1+f} //16

}

};

//roomtypes floortraps

//dit is een array met de locaties van traps die op de vloer liggen in een kamer

int[,,] floorTrapsVariation = new int[,,]

{

{

//room 0 wordt niet gebruikt

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //1

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //2

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //3

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //4

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //5

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //6

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //7

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //8

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //9

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //10

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //11

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //12

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //13

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //14

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //15

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 } //16

},

//room 1 (spawn room)

{

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //1

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //2

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //3

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //4

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //5

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //6

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //7

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //8

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //9

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //10

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //11

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //12

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //13

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //14

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //15

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 } //16

},

//room 2

{

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //1

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //2

{0 , 0 , 1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //3

{0 , 0 , 1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //4

{0 , 0 , 1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //5

{0 , 0 , 1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //6

{0 , 0 , 1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //7

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //8

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //9

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //10

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //11

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //12

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //13

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //14

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //15

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 } //16

},

//room 3

{

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //1

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //2

{0 , 0 , 0 , 0 , 1 , 1 , 1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //3

{0 , 0 , 0 , 0 , 1 , 1 , 1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //4

{0 , 0 , 0 , 0 , 1 , 1 , 1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //5

{0 , 0 , 0 , 0 , 1 , 1 , 1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //6

{0 , 0 , 0 , 0 , 1 , 1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //7

{0 , 0 , 0 , 0 , 1 , 1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //8

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //9

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //10

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //11

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //12

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //13

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //14

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //15

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 } //16

}

};

//roomtypes wall traps

//dit is een array met de locaties van traps die op muren komen in een kamer

int[,,] wallTrapsVariation = new int[,,]

{

{

//room 0 wordt niet gebruikt

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //1

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //2

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //3

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //4

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //5

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //6

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //7

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //8

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //9

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //10

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //11

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //12

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //13

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //14

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //15

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 } //16

},

//room 1 (spawn room)

{

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //1

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //2

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //3

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //4

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //5

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //6

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //7

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //8

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //9

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //10

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //11

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //12

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //13

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //14

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //15

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 } //16

},

//room 2

{

{1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //1

{1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //2

{1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //3

{1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //4

{1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //5

{1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //6

{1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //7

{1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //8

{1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //9

{1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //10

{1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //11

{1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //12

{1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //13

{1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //14

{1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //15

{1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 } //16

},

//room 3

{

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //1

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //2

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //3

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //4

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //5

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //6

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //7

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //8

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //9

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //10

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //11

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //12

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //13

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //14

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //15

{1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 } //16

}

};

//dit is een array met de locaties van vijanden in een kamer

int[,,] enemieVariation = new int[,,]

{

{

//room 0 wordt niet gebruikt

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //1

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //2

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //3

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //4

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //5

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //6

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //7

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //8

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //9

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //10

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //11

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //12

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //13

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //14

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //15

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 } //16

},

//room 1 (spawn room)

{

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //1

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //2

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //3

{0 , 0 , 0 , 1 , 0 , 1 , 0 , 1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //4

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //5

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //6

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //7

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //8

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //9

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //10

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //11

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //12

{0 , 0 , 0 , 1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //13

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //14

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //15

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 } //16

},

//room 2

{

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //1

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //2

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //3

{0 , 0 , 0 , 1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 1 , 0 , 0 , 0 }, //4

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //5

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //6

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //7

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //8

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //9

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //10

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //11

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //12

{0 , 0 , 0 , 1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 1 , 0 , 0 , 0 }, //13

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //14

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //15

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 } //16

},

//room 3

{

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //1

{0 , 1 , 1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //2

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //3

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //4

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //5

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //6

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //7

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //8

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //9

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //10

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //11

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //12

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //13

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //14

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 }, //15

{0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 } //16

}

};

int[,,] grid = new int[,,]

{

//room locations

// dit is een array met de locaties van kamers en de soort kamers

{{0, 0, 0, 0 }, {0, 0, 0, 0 }, {0, 0, 0, 0 }, {0, 0, 0, 0 }, {0, 0, 0, 0 }, {0, 0, 0, 0 }, {0, 0, 0, 0 }, {0, 0, 0, 0 }, {0, 0, 0, 0 }, {0, 0, 0, 0 }},//1

{{0, 0, 0, 0 }, {0, 0, 0, 0 }, {0, 0, 0, 0 }, {0, 0, 0, 0 }, {0, 0, 0, 0 }, {0, 0, 0, 0 }, {0, 0, 0, 0 }, {0, 0, 0, 0 }, {0, 0, 0, 0 }, {0, 0, 0, 0 }},//3

{{0, 0, 0, 0 }, {0, 0, 0, 0 }, {0, 0, 0, 0 }, {0, 0, 0, 0 }, {0, 0, 0, 0 }, {0, 0, 0, 0 }, {0, 0, 0, 0 }, {0, 0, 0, 0 }, {0, 0, 0, 0 }, {0, 0, 0, 0 }},//4

{{0, 0, 0, 0 }, {0, 0, 0, 0 }, {0, 0, 0, 0 }, {0, 0, 0, 0 }, {0, 0, 0, 0 }, {0, 0, 0, 0 }, {0, 0, 0, 0 }, {0, 0, 0, 0 }, {0, 0, 0, 0 }, {0, 0, 0, 0 }},//5

{{0, 0, 0, 0 }, {0, 0, 0, 0 }, {0, 0, 0, 0 }, {0, 0, 0, 0 }, {0, 0, 0, 0 }, {0, 0, 0, 0 }, {0, 0, 0, 0 }, {0, 0, 0, 0 }, {0, 0, 0, 0 }, {0, 0, 0, 0 }},//6

{{0, 0, 0, 0 }, {0, 0, 0, 0 }, {0, 0, 0, 0 }, {0, 0, 0, 0 }, {0, 0, 0, 0 }, {0, 0, 0, 0 }, {0, 0, 0, 0 }, {0, 0, 0, 0 }, {0, 0, 0, 0 }, {0, 0, 0, 0 }},//7

{{0, 0, 0, 0 }, {0, 0, 0, 0 }, {0, 0, 0, 0 }, {0, 0, 0, 0 }, {0, 0, 0, 0 }, {0, 0, 0, 0 }, {0, 0, 0, 0 }, {0, 0, 0, 0 }, {0, 0, 0, 0 }, {0, 0, 0, 0 }},//8

{{0, 0, 0, 0 }, {0, 0, 0, 0 }, {0, 0, 0, 0 }, {0, 0, 0, 0 }, {0, 0, 0, 0 }, {0, 0, 0, 0 }, {0, 0, 0, 0 }, {0, 0, 0, 0 }, {0, 0, 0, 0 }, {0, 0, 0, 0 }},//9

{{0, 0, 0, 0 }, {0, 0, 0, 0 }, {0, 0, 0, 0 }, {0, 0, 0, 0 }, {0, 0, 0, 0 }, {0, 0, 0, 0 }, {0, 0, 0, 0 }, {0, 0, 0, 0 }, {0, 0, 0, 0 }, {0, 0, 0, 0 }},//10

{{0, 0, 0, 0 }, {0, 0, 0, 0 }, {0, 0, 0, 0 }, {0, 0, 0, 0 }, {0, 0, 0, 0 }, {0, 0, 0, 0 }, {0, 0, 0, 0 }, {0, 0, 0, 0 }, {0, 0, 0, 0 }, {0, 0, 0, 0 }}//11

};

//room location code

//dit is een for loop die kamers probeert te generen

for (int trys = 0; trys < roomGenTrys; trys++)

{

//bij het genereren van kamers begint de code in het midden van een grid

//daarna bepaalt de float direction of hij zich een vakje verplaats op de X ass of een een vakje op de Z ass

//als hij rand van de grid komt beweegt hij van de rand af

float direction = UnityEngine.Random.Range(-1, 2);

if (direction < 0 && roomX > 2 && roomX < 8)

{

//hier wordt gekozen of hij naar links of naar rechts beweegt

roomX = roomXold + UnityEngine.Random.Range(-1, 2);

roomZ = roomZold;

roomXold = roomX;

}

if (direction < 0 && roomX < 2)

{

roomX = roomXold + 1;

roomZ = roomZold;

roomXold = roomX;

}

if (direction < 0 && roomX > 7)

{

roomX = roomXold - 1;

roomZ = roomZold;

roomXold = roomX;

}

if (direction > 0 && roomZ > 2 && roomZ < 8)

{

//hier wordt gekozen of hij naar boven of naar beneden gaat

roomZ = roomZold + UnityEngine.Random.Range(-1, 2);

roomX = roomXold;

roomZold = roomZ;

}

if (direction > 0 && roomZ < 2)

{

roomZ = roomZold + 1;

roomX = roomXold;

roomZold = roomZ;

}

if (direction > 0 && roomZ > 7)

{

roomZ = roomZold - 1;

roomX = roomXold;

roomZold = roomZ;

}

//room type randomizer

//hier zet hij op de locatie waar hij in het grid staat een kamer en bepaalt hij hoe de kamer eruit ziet hij geeft die locatie

//een nummer voor welke plattegrond er moet worden gebruikt voor de muren, vloeren, trap, en vijanden

grid[System.Convert.ToInt32(roomZ), System.Convert.ToInt32(roomX), 0] = UnityEngine.Random.Range(2, 4);

grid[System.Convert.ToInt32(roomZ), System.Convert.ToInt32(roomX), 1] = UnityEngine.Random.Range(1, 4) + difficulty;

grid[System.Convert.ToInt32(roomZ), System.Convert.ToInt32(roomX), 2] = UnityEngine.Random.Range(1, 4) + difficulty;

grid[System.Convert.ToInt32(roomZ), System.Convert.ToInt32(roomX), 3] = UnityEngine.Random.Range(1, 4) + difficulty;

}

//spawn

// dit zorgt dat de spawn locatie de goeie plattegrond krijgt

grid[4, 4, 0] = 1;

grid[4, 4, 1] = 0;

grid[4, 4, 2] = 0;

grid[4, 4, 3] = 0;

//this fixed a bug

grid[0, 0, 0] = 0;

//room maker

//dit is een for loop die kijkt bij elke positie in de kamer array of daar een kamer moet staan en welke kamer

for (int z = 0; z < grid.GetLength(0) - 1; z++)

{

for (int x = 0; x < grid.GetLength(1) - 1; x++)

{

if (grid[x, z, 0] > 0)

{

//if (startTile == true)

//{

//}

//door and start location

//if (startTile == false)

//{

//startTile = true;

//}

//side walls

//dit kijkt of er naast een kamer geen andere kamer staat en zet muren aan de kant waar geen andere kamer staat

//kijkt naar de Z ass

if (x < 9 && x > 0 && z < 9 && z > 0 && grid[x - 1, z, 0] < 1)

{

for(int zLeft = 0; zLeft < 16; zLeft++)

{

if (roomTypeVariation[grid[x, z, 0], 0, zLeft] > 0 && roomTypeVariation[grid[x, z, 0], 0, zLeft] < f)

{

//laad muren uit de lijst

Instantiate(wallTiles[1], new Vector3((0 \* 8 + x \* 128 - 12)\*2, 0, (zLeft \* -8 + z \* -128 + 1012)\*2), Quaternion.identity);

}

}

}

if (x < 8 && x > 0 && z < 9 && z > 0 && grid[x + 1, z, 0] < 1)

{

for (int zRight = 0; zRight < 16; zRight++)

{

if (roomTypeVariation[grid[x, z, 0], 15, zRight] > 0 && roomTypeVariation[grid[x, z, 0], 15, zRight] < f)

{

//laad muren uit de lijst

Instantiate(wallTiles[1], new Vector3((15 \* 8 + x \* 128 - 12)\*2, 0, (zRight \* -8 + z \* -128 + 1012)\*2), Quaternion.identity);

}

}

}

//kijkt naar de X ass

if (z < 9 && z > 0 && x < 9 && x > 0 && grid[x, z - 1, 0] < 1)

{

for (int xUp = 0; xUp < 16; xUp++)

{

if (roomTypeVariation[grid[x, z, 0], xUp, 0] > 0 && roomTypeVariation[grid[x, z, 0], xUp, 0] < f)

{

//laad muren uit de lijst

Instantiate(wallTiles[1], new Vector3((xUp \* 8 + x \* 128 - 12)\*2, 0, (0 \* -8 + z \* -128 + 1012)\*2), Quaternion.identity);

}

}

}

if (z < 8 && z > 0 && x < 9 && x > 0 && grid[x, z + 1, 0] < 1)

{

for (int xDown = 0; xDown < 16; xDown++)

{

if (roomTypeVariation[grid[x, z, 0], xDown, 15] > 0 && roomTypeVariation[grid[x, z, 0], xDown, 15] < f)

{

//laad muren uit de lijst

Instantiate(wallTiles[1], new Vector3((xDown \* 8 + x \* 128 - 12)\*2, 0, (15 \* -8 + z \* -128 + 1012)\*2), Quaternion.identity);

}

}

}

//room grid

//if (grid[x, z, 0] > 0)

//{

//deze for loop kijkt naar elke locatie in een kamer en kijk of daar een blok moet staan en welk blok

for (int xInRoom = 0; xInRoom < 16; xInRoom++)

{

for (int zInRoom = 0; zInRoom < 16; zInRoom++)

{

//loads floors and floor traps and enemies

//kijkt of het een vloer tile is en laad een vloer

if (roomTypeVariation[grid[x, z, 0], xInRoom, zInRoom] > 0 && roomTypeVariation[grid[x, z, 0], xInRoom, zInRoom] < f)

{

if (startTile == true)

{

//laad een vloer uit de lijst

Instantiate(floorTiles[roomTypeVariation[grid[x, z, 0], xInRoom, zInRoom]], new Vector3((xInRoom \* 8 + x \* 128 - 12)\*2, -8, (zInRoom \* -8 + z \* -128 + 1012)\*2), Quaternion.identity);

//kijkt of er een trap moet staan

if (floorTrapsVariation[grid[x, z, 1], xInRoom, zInRoom] > 0)

{

//laad een trap uit de lijst

Instantiate(floorTraps[floorTrapsVariation[grid[x, z, 1], xInRoom, zInRoom]], new Vector3((xInRoom \* 8 + x \* 128 - 12) \* 2, -8, (zInRoom \* -8 + z \* -128 + 1012) \* 2), Quaternion.identity);

}

//kijkt of er een vijand moet staan

if (enemieVariation[grid[x, z, 3], xInRoom, zInRoom] > 0)

{

//laad een vijand

Instantiate(enemie[enemieVariation[grid[x, z, 3], xInRoom, zInRoom]], new Vector3((xInRoom \* 8 + x \* 128 - 12) \* 2, -3, (zInRoom \* -8 + z \* -128 + 1012) \* 2), enemie[enemieVariation[grid[x, z, 3], xInRoom, zInRoom]].transform.rotation);

}

}

}

//loads walls and wall traps

//kijkt of er een muur moet staan

if (roomTypeVariation[grid[x, z, 0], xInRoom, zInRoom] > f && roomTypeVariation[grid[x, z, 0], xInRoom, zInRoom] < w + f + 1)

{

if (startTile == true)

{

//laad een muur uit de lijst

Instantiate(wallTiles[roomTypeVariation[grid[x, z, 0], xInRoom, zInRoom] - f], new Vector3((xInRoom \* 8 + x \* 128 - 12)\*2, 0, (zInRoom \* -8 + z \* -128 + 1012)\*2), Quaternion.identity);

//kijkt of er een trap moet staan

if (wallTrapsVariation[grid[x, z, 2], xInRoom, zInRoom] > 0)

{

//laad de trap uit de lijst

Instantiate(wallTraps[wallTrapsVariation[grid[x, z, 2], xInRoom, zInRoom]], new Vector3((xInRoom \* 8 + x \* 128 - 12) \* 2, 0, (zInRoom \* -8 + z \* -128 + 1012) \* 2), Quaternion.identity);

}

}

}

}

//}

}

}

}

}

}

// Update is called once per frame

void Update()

{

}

}