using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class hubLoadScript : MonoBehaviour

{

//maakt lijsten waar later de blokken in worden geladen

public List<GameObject> floorTiles;

public List<GameObject> wallTiles;

//list lengt

//lengte van de lijsten

int f;

int w;

// Start is called before the first frame update

void Start()

{

//LoadHub functie

LoadHub();

}

public void LoadHub()

{

//de blokken worden uit folders gehaald en in de lijsten geladen

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

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

//stopt de lengte van de lijsten in variabele

f = floorTiles.Count;

w = wallTiles.Count;

//dit is een 2d array van de locaties van de blokken

//alles zonder + f is een vloer alles met + f is een muur

int[,] hubMap = new int[,]

{

{0, 0, 0, 0, 0, 0, 0, 0, 1+f, 1+f, 1+f, 1+f, 1+f, 1+f, 1+f, 1+f, 1+f, 1+f, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}, //1

{0, 0, 0, 0, 0, 0, 0, 0, 1+f, 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1+f, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}, //2

{0, 0, 0, 0, 0, 0, 0, 0, 1+f, 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1+f, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}, //3

{0, 0, 0, 0, 0, 0, 0, 0, 1+f, 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1+f, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}, //4

{0, 0, 0, 0, 0, 0, 0, 0, 1+f, 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1+f, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}, //5

{0, 0, 0, 0, 0, 0, 0, 0, 1+f, 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1+f, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}, //6

{0, 0, 0, 0, 0, 0, 0, 0, 1+f, 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1+f, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}, //7

{0, 0, 0, 0, 0, 0, 0, 0, 1+f, 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1+f, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}, //8

{0, 0, 0, 0, 0, 0, 0, 0, 1+f, 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1+f, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}, //9

{0, 0, 0, 0, 0, 0, 0, 0, 1+f, 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1+f, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}, //10

{0, 0, 0, 0, 0, 0, 0, 0, 1+f, 1+f, 1+f, 1+f, 1 , 1 , 1+f, 1+f, 1+f, 1+f, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}, //11

{0, 0, 0, 0, 0, 0, 0, 0, 1+f, 2 , 2 , 2 , 2 , 2 , 2 , 2 , 2 , 1+f, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}, //12

{0, 0, 0, 0, 0, 0, 0, 0, 1+f, 2 , 2 , 2 , 2 , 2 , 2 , 2 , 2 , 1+f, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}, //13

{0, 0, 0, 0, 0, 0, 0, 0, 1+f, 2 , 2 , 2 , 2 , 2 , 2 , 2 , 2 , 1+f, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}, //14

{0, 0, 0, 0, 0, 0, 0, 0, 1+f, 2 , 2 , 2 , 2 , 2 , 2 , 2 , 2 , 1+f, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}, //15

{0, 0, 0, 0, 0, 0, 0, 0, 1+f, 2 , 2 , 2 , 2 , 2 , 2 , 2 , 2 , 1+f, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}, //16

{0, 0, 0, 0, 0, 0, 0, 0, 1+f, 2 , 2 , 2 , 2 , 2 , 2 , 2 , 2 , 1+f, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}, //17

{0, 0, 0, 0, 0, 0, 0, 0, 1+f, 2 , 2 , 2 , 2 , 2 , 2 , 2 , 2 , 1+f, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}, //18

{0, 0, 0, 0, 0, 0, 0, 0, 1+f, 2 , 2 , 2 , 2 , 2 , 2 , 2 , 2 , 1+f, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}, //19

{0, 0, 0, 0, 0, 0, 0, 0, 1+f, 2 , 2 , 2 , 2 , 2 , 2 , 2 , 2 , 1+f, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}, //20

{0, 0, 0, 0, 0, 0, 0, 0, 1+f, 2 , 2 , 2 , 2 , 2 , 2 , 2 , 2 , 1+f, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}, //21

{0, 0, 0, 0, 0, 0, 0, 0, 1+f, 2 , 2 , 2 , 2 , 2 , 2 , 2 , 2 , 1+f, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}, //22

{0, 0, 0, 0, 0, 0, 0, 0, 1+f, 2 , 2 , 2 , 2 , 2 , 2 , 2 , 2 , 1+f, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}, //23

{0, 0, 0, 0, 0, 0, 0, 0, 1+f, 2 , 2 , 2 , 2 , 2 , 2 , 2 , 2 , 1+f, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}, //24

{0, 0, 0, 0, 0, 0, 0, 0, 1+f, 2 , 2 , 2 , 2 , 2 , 2 , 2 , 2 , 1+f, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}, //25

{0, 0, 0, 0, 0, 0, 0, 0, 1+f, 2 , 2 , 2 , 2 , 2 , 2 , 2 , 2 , 1+f, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}, //26

{0, 0, 0, 0, 0, 0, 0, 0, 1+f, 2 , 2 , 2 , 2 , 2 , 2 , 2 , 2 , 1+f, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}, //27

{0, 0, 0, 0, 0, 0, 0, 0, 1+f, 2 , 2 , 2 , 2 , 2 , 2 , 2 , 2 , 1+f, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}, //28

{0, 0, 0, 0, 0, 0, 0, 0, 1+f, 1+f, 1+f, 1+f, 1+f, 1+f, 1+f, 1+f, 1+f, 1+f, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}, //29

{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}, //30

{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}, //31

{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} //32

};

//dit is een for loop die naar alle locaties in de array kijkt en daar een blok laadt

for (int xInHub = 0; xInHub < 32; xInHub++)

{

for (int zInHub = 0; zInHub < 32; zInHub++)

{

if (hubMap[xInHub, zInHub] > 0 && hubMap[xInHub, zInHub] < f)

{

//laad een vloer blok uit de lijst

Instantiate(floorTiles[hubMap[xInHub, zInHub]], new Vector3((xInHub \* 8)\*2, -8, (zInHub \* 8)\*2), Quaternion.identity);

}

if (hubMap[xInHub, zInHub] > f && hubMap[xInHub, zInHub] < w + f + 1)

{

//laad een muur blok uit de lijst

Instantiate(wallTiles[hubMap[xInHub, zInHub] - f], new Vector3((xInHub \* 8)\*2, 0, (zInHub \* 8)\*2), Quaternion.identity);

}

}

}

}

// Update is called once per frame

void Update()

{

}

}