using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class Spawner\_Modular : MonoBehaviour

{

public Pathfinding ref\_PathFinding;

public Grid ref\_Grid;

public GameManagerBehavior ref\_GameManager;

public Node spawnNode;

private GameObject spawnObject;

private GameObject spawnWeapon;

public GameObject spawnParticle;

private float SpawnRate, rowInterval, SpawnCount, SpawnType;

private int spawnRowCount, temp\_spawnRowCount;

public bool startSpawning;

private bool coroutineRunning, BstartCoroutine, IsValueSet;

private int currentWave = 0;

public GameObject normalEmu;

public GameObject mechEmu;

public GameObject tankEmu;

public GameObject sniperRifle;

public GameObject shotgun;

public GameObject spammer;

public GameObject runner;

public float[,] spawnValueArray;

// Use this for initialization

void Start()

{

temp\_spawnRowCount = 0;

spawnRowCount = 0;

coroutineRunning = false;

BstartCoroutine = false;

startSpawning = false;

IsValueSet = false;

}

// Update is called once per frame

void Update()

{

if (startSpawning)

{

spawnTimer();

}

}

public void spawnSetup()

{

currentWave = ref\_GameManager.Wave + 1;

for (int i1 = spawnValueArray.GetLength(0) - 1; i1 >= 0; i1--)

{

if (spawnValueArray[i1, 0] == currentWave)

{

temp\_spawnRowCount = i1; //let the row count starts at the correct wave row

spawnRowCount += 1;

ref\_GameManager.enemySpawnCount += Mathf.RoundToInt(spawnValueArray[i1, 2]);

}

}

IsValueSet = false;

spawnRowCount += temp\_spawnRowCount; //so that the "temp\_spawnRowCount < spawnRowCount" will work correctly

}

void spawnTimer()

{

if (coroutineRunning == false && temp\_spawnRowCount < spawnRowCount) //while there are spawn row left

{

if (!IsValueSet) //so that rowInterval will only be set once

{

SpawnRate = spawnValueArray[temp\_spawnRowCount, 1];

SpawnCount = spawnValueArray[temp\_spawnRowCount, 2];

SpawnType = spawnValueArray[temp\_spawnRowCount, 3];

rowInterval = spawnValueArray[temp\_spawnRowCount, 4];

switch (Mathf.RoundToInt(SpawnType))

{

case 1:

//spawnObject = sniperEmu;

spawnObject = normalEmu;

spawnWeapon = sniperRifle;

break;

case 2:

//spawnObject = spamEmu;

spawnObject = normalEmu;

spawnWeapon = spammer;

break;

case 3:

//spawnObject = bombEmu;

spawnObject = normalEmu;

break;

case 4:

//spawnObject = fastEmu;

spawnObject = normalEmu;

spawnWeapon = runner;

break;

case 5:

//spawnObject = sniperMech;

spawnObject = mechEmu;

spawnWeapon = sniperRifle;

break;

case 6:

//spawnObject = spamMech;

spawnObject = mechEmu;

spawnWeapon = spammer;

break;

case 7:

//spawnObject = fastMech;

spawnObject = mechEmu;

spawnWeapon = runner;

break;

case 8:

//spawnObject = sniperTank;

spawnObject = tankEmu;

spawnWeapon = sniperRifle;

break;

case 9:

//spawnObject = spamTank;

spawnObject = tankEmu;

spawnWeapon = spammer;

break;

case 10:

//spawnObject = spamTank;

spawnObject = normalEmu;

spawnWeapon = shotgun;

break;

case 11:

//spawnObject = spamTank;

spawnObject = mechEmu;

spawnWeapon = shotgun;

break;

case 12:

//spawnObject = spamTank;

spawnObject = tankEmu;

spawnWeapon = shotgun;

break;

default:

break;

}

IsValueSet = true;

}

rowInterval -= Time.deltaTime; //wait until row spawn timer is done

if(rowInterval < 0)

{

temp\_spawnRowCount += 1;

BstartCoroutine = true;

}

}

else if (coroutineRunning == false && temp\_spawnRowCount == spawnRowCount) //run out of spawn row

{

spawnRowCount = 0;

startSpawning = false;

}

if (BstartCoroutine == true) //fire the coroutine once

{

IEnumerator spawnCoroutine = spawn(SpawnRate, SpawnCount);

StartCoroutine(spawnCoroutine);

}

}

IEnumerator spawn(float temp\_SpawnRate, float temp\_SpawnCount)

{

BstartCoroutine = false;

coroutineRunning = true;

for (int i = 0; i < temp\_SpawnCount; i++)

{

List<Node> temp\_FinalPath = ref\_PathFinding.returnPath(spawnNode);

var temp\_Emu = Instantiate(spawnObject, spawnNode.position, Quaternion.identity);

var temp\_Weapon = Instantiate(spawnWeapon, temp\_Emu.transform.Find("GunPosition").transform.position , Quaternion.identity);

temp\_Weapon.name = "Weapon";

temp\_Weapon.transform.parent = temp\_Emu.transform;

Instantiate(spawnParticle, spawnNode.position, Quaternion.identity);

if (temp\_Emu.GetComponent<MoveEnemy>() != null)

{

temp\_Emu.GetComponent<MoveEnemy>().waypoints = temp\_FinalPath;

}

if (i == temp\_SpawnCount - 1)

{

coroutineRunning = false;

IsValueSet = false; //set the values again

}

yield return new WaitForSeconds(temp\_SpawnRate);

}

}

}