﻿using UnityEngine;

using System.Collections;

public class ColumnSpawnScript : MonoBehaviour

{

public GameObject columnPrefab; //the column game object

public int columnPoolSize = 5; //how many columns to keep on standby

public float spawnRate = 3f; //how quickly columns spawn

public float columnMin = -1f; //minimum y value of the column position

public float columnMax = 3.5f; //maximum y value of the column position

GameObject[] columns; //collection of pooled columns

int currentColumn = 0; //index of the current column in the collection

void Start()

{

//initialize the columns collection

columns = new GameObject[columnPoolSize];

//loop through the collection and create the individual columns

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

{

//note that columns will have the exact position and rotation of the prefab asset.

//this is because we did not specify the position and rotation in the

//Instantiate() method call

columns[i] = (GameObject)Instantiate(columnPrefab);

}

//starts our function in charge of spawning the columns in the playable area

StartCoroutine ("SpawnLoop");

}

public void StopSpawn()

{

//stops our spawning function

StopCoroutine("SpawnLoop");

}

//this is a coroutine which manages when columns are spawned

IEnumerator SpawnLoop()

{

//infinite loop: use with caution

while (true)

{

//To spawn a column, get the current spawner position...

Vector3 pos = transform.position;

//...set a random y position...

pos.y = Random.Range(columnMin, columnMax);

//...then set the current column to that position.

columns[currentColumn].transform.position = pos;

//increase the value of currentColumn. If the new size is too big, set it back to zero

if(++currentColumn >= columnPoolSize)

currentColumn = 0;

//leave this coroutine until the proper amount of time has passed

yield return new WaitForSeconds(spawnRate);

}

}

}