2D Random Spawn Point Tutorial

Coding

* *Note that for this tutorial, I am continuing from the 2D Coin Collectable Tutorial, so for this tutorial, you should have a player that can move from a top down angle, and a coin that the player can collect. There should also be a score counter that goes up every time the player gets a coin. Hopefully this clears up any potential confusion as to why I’m skipping creating any steps. If you haven’t done these steps yet, try the 2D Top Down Movement Tutorial and the 2D Coin Collectable Tutorial first.*
* ***Getting Started:***
* First thing’s first; we need to make the coin a prefab. A prefab is basically a template that you can use to create new variations of the same objects. To do this, head to the Hierarchy menu (the menu on the left hand side of the screen, and above the Project menu), and drag the coin into your Project menu (the menu below the scene view and the Hierarchy menu). Once it is dragged in, you should see that your Coin game object in the Hierarchy has changed from white to blue. While you’re here, create a Prefabs folder (right click in the Project menu, select Create, Folder, and rename it Prefabs) and drag the Coin into your new folder, so the menu doesn’t get cluttered.
* Delete the coin from your scene (right click the Coin in your Hierarchy menu and click delete), and create a new script by right clicking in the Project menu and selecting Create, C# Script and rename it “Spawner” by clicking on the text of the folder. Drag this new script into your Scripts folder and in your Hierarchy, create an Empty Game Object by right clicking in the Hierarchy menu, and select Create Empty. Rename this new object “Spawner” in the same way how you renamed your Script, and while your Spawner game object is selected, drag your Spawner script into the Inspector menu (the menu on the right hand side of the screen), and once it is attached, double click on the script to open it in Visual Studio.
* ***Beginning to Code:***
* Let’s start coding! Once you are inside Visual Studio, delete lines 7 to 17 so you are left with this:
* using System.Collections;  
  using System.Collections.Generic;  
  using UnityEngine;  
    
  public class Spawner : MonoBehaviour  
  {  
    
  }
* In between the {}’s, type public GameObject[] coins; Below this, type public transform spawnPoint;
* Below this, type private int rand; (rand is short for Random).
* Below this, type public float startTimeBtwSpawns; (this is a timer that determines how long it takes for a coin to spawn).
* Below this, type private float timeBtwSpawns; Then beneath all of this, type void Start () and then type { and press the enter key to complete the Start function for you. Your code should now look something like this:
* using System.Collections;  
  using System.Collections.Generic;  
  using UnityEngine;  
    
  public class Spawner : MonoBehaviour  
  {  
   public GameObject[] coins;  
   public Transform spawnPoint;  
    
   private int rand;  
    
   public float startTimeBtwSpawns;  
   private float timeBtwSpawns;  
    
   void Start ()  
   {  
    
   }  
  }
* In between the new {}’s you have just created in the Start function, type timeBtwSpawns = startTimeBtwSpawns; and beneath the second } from the bottom (aka the end of the Start function), type void Update () and type { and press enter to create an Update function, similar to how you made the Start function. Your code should now look like this:
* using System.Collections;  
  using System.Collections.Generic;  
  using UnityEngine;  
    
  public class Spawner : MonoBehaviour  
  {  
   public GameObject[] coins;  
   public Transform spawnPoint;  
    
   private int rand;  
    
   public float startTimeBtwSpawns;  
   private float timeBtwSpawns;  
    
   void Start ()  
   {  
   timeBtwSpawns = startTimeBtwSpawns;  
   }  
    
   void Update ()  
   {  
    
   }  
  }
* In your new Update function (aka in between the new {}’s you have just created), type if(timeBtwSpawns <= 0) then press { and enter to create new curly brackets. Then type in between these new {}’s you have created Instantiate(coins[0], spawnPoint.transform.position, Quaternion.identity); Your code should now look like this:
* using System.Collections;  
  using System.Collections.Generic;  
  using UnityEngine;  
    
  public class Spawner : MonoBehaviour  
  {  
   public GameObject[] coins;  
   public Transform spawnPoint;  
    
   private int rand;  
    
   public float startTimeBtwSpawns;  
   private float timeBtwSpawns;  
    
   void Start ()  
   {  
   timeBtwSpawns = startTimeBtwSpawns;  
   }  
    
   void Update ()  
   {  
   if(timeBtwSpawns <= 0)  
   {  
   Instantiate(coins[0], spawnPoint.transform.position, Quaternion.identity);  
   }  
   }  
  }
* Save your code and head back into Unity. If you click on the Inspector menu once your Spawner has been selected, you will notice some changes. From your Prefabs folder, drag your Coin prefab into the Spawner script (this will only work if the Spawner game object is selected, so make sure you do that). You should now see that the Element 0 box should be equal to Coin. (note that from the code before, when you made Instantiate(coins[0]), it means that the spawner will only spawn whatever Element 0 is equal to, which in this case, is your coin). Set the time between spawns (identified by time Btw Spawns) to 1, which means that a coin will spawn every one second.
* Now we need to create a spawn point for our coin to appear from. To do this, head to the Hierarchy and right click. Select Create Empty and rename this object “SpawnPoint.” We need this to be visible in our scene, so to do this, with the Spawn Point object selected, select the grey cube with a down arrow attached to it underneath the Inspector text in the Inspector menu. You will see various icons that you can use. Select whatever one you like, as long as your spawn point is visible. Select the Spawner game object and drag your Spawn Point game object from the Hierarchy to the Spawn Point box in the Inspector menu under your Spawner script.
* Go back into your Spawner script in Visual Studio. Below the third lowest } (aka the } after your Instantiate line), type else and then { and press enter.
* In between the new curly brackets you have made, type timeBtwSpawns -= Time.deltaTime; now head back to your Instantiate line and beneath it (and while still in the Update function) type timeBtwSpawns = startTimeBtwSpawns; (this means that the timer will reset once a coin has spawned). Your code should now look like this:
* using System.Collections;  
  using System.Collections.Generic;  
  using UnityEngine;  
    
  public class Spawner : MonoBehaviour  
  {  
   public GameObject[] coins;  
   public Transform spawnPoint;  
    
   private int rand;  
    
   public float startTimeBtwSpawns;  
   private float timeBtwSpawns;  
    
   void Start ()  
   {  
   timeBtwSpawns = startTimeBtwSpawns;  
   }  
    
   void Update ()  
   {  
   if(timeBtwSpawns <= 0)  
   {  
   Instantiate(coins[0], spawnPoint.transform.position, Quaternion.identity);  
   timeBtwSpawns = startTimeBtwSpawns;  
   }  
    
   else  
   {  
   timeBtwSpawns -= Time.deltaTime;  
   }  
   }  
  }
* Save your code and head back to Unity. If you play your game, you should see multiple coins being spawned (this is also illustrated in the Hierarchy as multiple clones will be spawned there as well). (note that if your coins don’t appear, always make sure that the Z axis is set to 0).
* ***Final Parts:***
* Head back to your Spawner code in Visual Studio. Above your Instantiate line from before, and still in the Update function, type rand = Random.Range (0, coins.Length); Your code should now look like this:
* using System.Collections;  
  using System.Collections.Generic;  
  using UnityEngine;  
    
  public class Spawner : MonoBehaviour  
  {  
   public GameObject[] coins;  
   public Transform spawnPoint;  
    
   private int rand;  
    
   public float startTimeBtwSpawns;  
   private float timeBtwSpawns;  
    
   void Start ()  
   {  
   timeBtwSpawns = startTimeBtwSpawns;  
   }  
    
   void Update ()  
   {  
   if(timeBtwSpawns <= 0)  
   {  
   rand = Random.Range(0, coins.Length);  
   Instantiate(coins[0], spawnPoint.transform.position, Quaternion.identity);  
   timeBtwSpawns = startTimeBtwSpawns;  
   }  
    
   else  
   {  
   timeBtwSpawns -= Time.deltaTime;  
   }  
   }  
  }
* Head back into Unity and select your Spawn Point in either the Hierarchy or the Scene view. Press ctrl/cmd D (cmd is only if you are on Mac) to duplicate the spawn point. You can do this however many times you want. Go back into Visual Studio and check your code for these additions:
* Where it says public Transform spawnPoint; put []’s next to Transform to make it an array (arrays allow you to store multiple objects in a single variable).
* On the Instantiate line in your Update function, add [rand] next to spawnPoint.
* Your code should look like this:
* using System.Collections;  
  using System.Collections.Generic;  
  using UnityEngine;  
    
  public class Spawner : MonoBehaviour  
  {  
   public GameObject[] coins;  
   public Transform[] spawnPoint;  
    
   private int rand;  
    
   public float startTimeBtwSpawns;  
   private float timeBtwSpawns;  
    
   void Start ()  
   {  
   timeBtwSpawns = startTimeBtwSpawns;  
   }  
    
   void Update ()  
   {  
   if(timeBtwSpawns <= 0)  
   {  
   rand = Random.Range(0, coins.Length);  
   Instantiate(coins[0], spawnPoint[rand].transform.position, Quaternion.identity);  
   timeBtwSpawns = startTimeBtwSpawns;  
   }  
    
   else  
   {  
   timeBtwSpawns -= Time.deltaTime;  
   }  
   }  
  }
* Save your code and head back to Unity. Drag all of your Spawn Point objects into the Spawn Point array in your Spawner script in the inspector menu (this should be found underneath Coins). Save your game and head back to your Spawner script in Visual Studio.
* Once there, below private int rand; type private int randPosition; then in between your rand line and your Instantiate line in the Update function, type randPosition = Random.Range(0, spawnPoint.Length); Finally, in your Instantiate line, inside the []’s, replace rand with randPosition.
* Save your code and head back to Unity. When you play your game, the coins should now spawn at random locations. Congratulations, you’ve successfully coded a random spawn point system! :D
* ***Extra:***
* Depending on the type of game you want to create, you can build on this code to make it whatever you want it to be. A quick example is making coins or enemies fall from the spawn points. All you need to do is place your spawn points at the top of the scene view, and add a Rigid Body 2D to the coin prefab. That’s it! The coins should now start to fall! :D
* The final code should look like this:
* using System.Collections;  
  using System.Collections.Generic;  
  using UnityEngine;  
    
  public class Spawner : MonoBehaviour  
  {  
   public GameObject[] coins;  
   public Transform[] spawnPoint;  
    
   private int rand;  
   private int randPosition;  
    
   public float startTimeBtwSpawns;  
   private float timeBtwSpawns;  
    
   void Start ()  
   {  
   timeBtwSpawns = startTimeBtwSpawns;  
   }  
    
   void Update ()  
   {  
   if(timeBtwSpawns <= 0)  
   {  
   rand = Random.Range(0, coins.Length);  
   randPosition = Random.Range(0, spawnPoint.Length);  
   Instantiate(coins[0], spawnPoint[randPosition].transform.position, Quaternion.identity);  
   timeBtwSpawns = startTimeBtwSpawns;  
   }  
    
   else  
   {  
   timeBtwSpawns -= Time.deltaTime;  
   }  
   }  
  }
* The video that inspired this tutorial:
* <https://www.youtube.com/watch?v=yjYSsRiA0do>