I started with the Brief rolling road, which I changed the name to Endless Runner, where I created an endless game that spawns obstacles, coins and constant running until you hit an obstacle or fall off the level, which will then bring you back to the start. It would be best if you began your journey all over again. The Player starts automatically, running when you press up, left, or right on the keyboard to dodge or jump over obstacles. You also have coins when collected to speed up your movement to make it harder, like games like Subway Surfers when you play for a long time. But through coins instead, I also added distance travel to make you know how far you travelled in the game through this script at the bottom of this paragraph to show how it was done.

7 Scripts allow the game to function appropriately when play has been pressed.

Create a Script which makes the player move. “PlayerMovement” allows the player to move forward when the game starts, and another part will enable you to move left and right on the screen. You can also jump over small obstacles by pressing the up key on the keyboard. You also can know how far you can travel by looking at the right side of the screen, which tells you you’re travelling distance.

The following Script was “CameraFollow”, which allows the camera to keep a certain distance from the player and stop the player from playing the game for two seconds before the fun begins.

The following Script is “Coin”, which allows you, the player, to collect coins partnered with “PlayerMovement” to register that you have collected the coins, which will show on the screen using the Text mechanic.

The “GameManager” Scripts allow me to allow the coins the player has collected to go up, which will show up on the left side of the screen, and the more you collect, the faster you go.

The “GroundSpawn” Script allows the game to spawn the ground, obstacle and coins to generate on the ground where the player will run, so you don’t fall out of the area when playing and be able to collect coins or jump over the obstacle.

The “GroundTile” Script allows the game to destroy the prefabs that appear in the game, so you don’t overload any of the engines you use. It allows you to also work together with “GroundSpawn” and gives it the go-ahead to spawn these prefabs.

The “Obstacle” Script lets you say that you will restart the game from the beginning when the player crashes into the obstacle.

Four Prefabs spawn the Coin, GroundTile, Obstacle and ObstacleTall that will appear in the game, which will be helpful or annoying depending on the situation.

The Things need to make an Endless runner infinite road happen. You right-click in unity, find the empty game object, rename it, and right-click the game object and add a plane (Ground) to it. Click on the game object again and add another game object rename however you think it is suitable and move the Z axis to ten so the ground can match perfectly and a new folder which you should call Prefabs, and then add the first game object you made into the prefabs folder. Delete it from the hierarchy, go to your script folder and create a GroundSpawner script. Then you add a public variable Game Object and call it the name of your prefabs. Go on the hierarchy, add another game object, and call it Ground Spawner. And then, you add the Ground Spawner Script to it. Return to the Ground Spawner script and add void Spawn Tile (spawn the tile and not add anything else.) Add the Instantiate (Allows the ground to generate and where we want to spawn it.) ground Tile, next Spawn Point (Where you want the next floor to follow) Quaternion identity (No rotations), basically referencing the child object of your main Game object. To reference the game object, using (Game Object temp =) in front of the rest of your code will allow you to spawn the ground temporarily.

For the code to work, you need to let the game know what you are referencing by Next Spawn Point (The next ground generating after the main floor) and using temp and telling what position and game object you are referencing so it will appear when needed. Add a number like one. And then, add the transform to get the component and position to get its status/position. We are not telling the engine what to spawn, so go on void start and add ground so it generate-pawn the ground you passed. If you use for (int I = 0; I < length; i++) and spawn Tile () to make the base generate endless on-scene. Add a box collider, make it bigger, and change the Z axis to cover the Ground so the player can’t miss the target. Create a script and call it Ground Tile. Add Ground Spawner and call it ground Spawner with a lowercase g. This script allows you to access functions and variables by giving the engine a reference to what might be your ground and what will be on it later. So, you can destroy the ground later, so it doesn’t crash your game with endless objects. Then add the script to your first Game object.