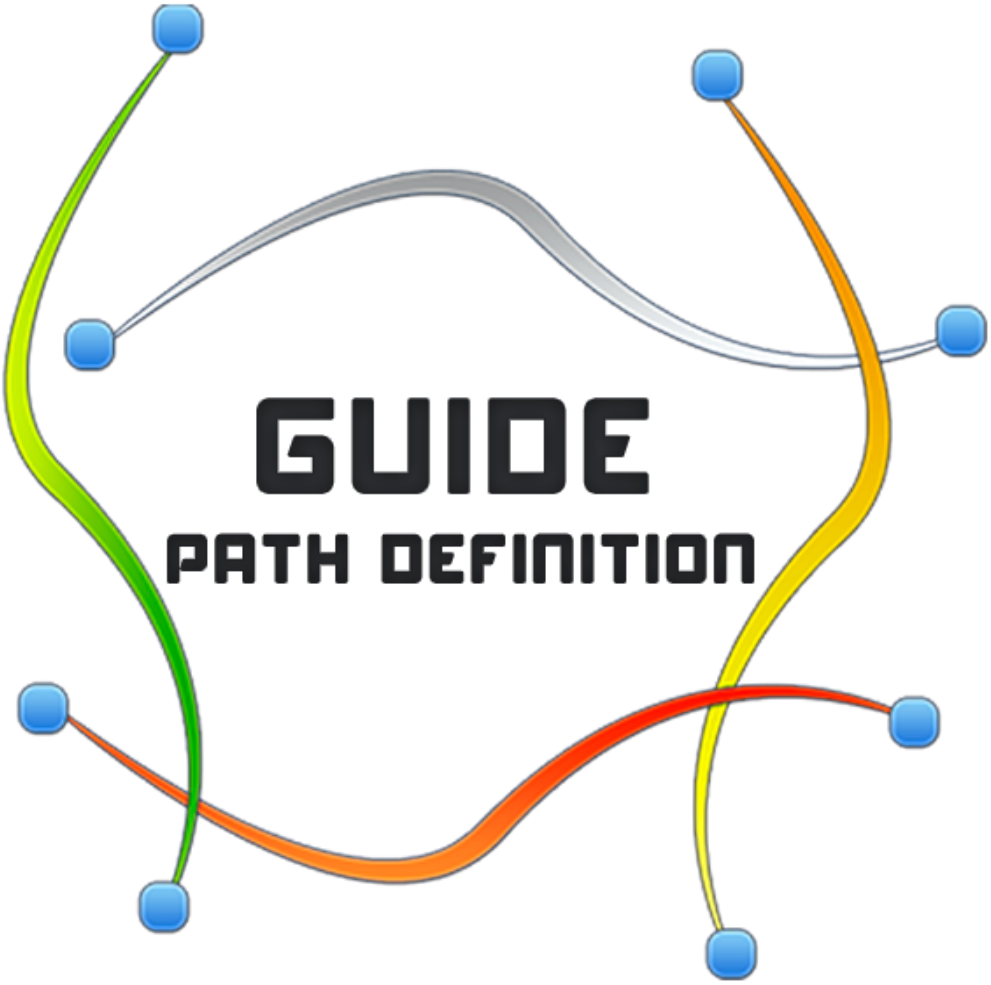


Introduction

First of all thanks for your purchase and I hope you find here what you are looking for.

Sometimes when we're developing our game we want our enemies, player, environment objects such as dogs, monsters, etc. to follow a certain predefined path of any form, or spawn x amount of those objects at once. Well that's what I'm expecting to give you with this Package.

What Wave Spawn Script does?

- Simple and as the name say spawn a wave of a certain object like enemies for example where it's needed.

What Path Definition Script does?

- Define/Create your **own** path with any form: curved, lineal, etc.

What Follow Path Script Does?

- Make your object move on the Path you created and rotate as (**Optional**).

Before we start how to setup everything I recommend you to take a look to the scenes and check how things works and what we can do.

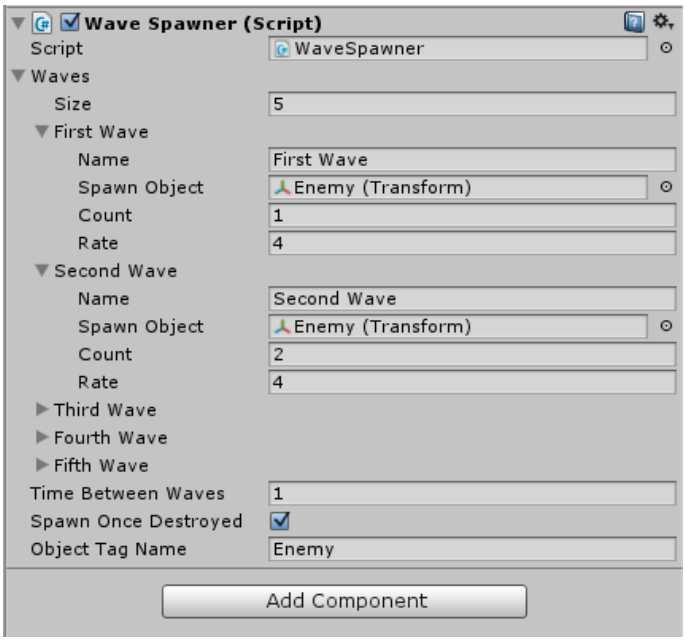
Warning!

- We don't get responsible for any compilation error if the default script is changed.

Contact me if need help or have a problem.

How to Setup our Wave Spawner

To get started open the Scene called "**WaveSpawnerDemo**" as you can see there's only 1 object in the scene. Well that's all we need actually, click the Object "**WaveSpawner**" and on Inspector you will see the script component "**WaveSpawner.cs**" like this:



I will explain each field in the same order the above image shows.

[**Fields**]

Size: How many waves we are going to spawn.

Name: Name of the Wave we are currently spawning.

Spawn Object: What Game Object we want to spawn.

Count: How many Game Objects we want to spawn on the current wave to go to the next wave.

Rate: How fast we want them to spawn.

Time Between Waves: How much time will wait to start spawning next wave.

Spawn Once Destroyed: Start counting time once all object of the current wave are destroyed.

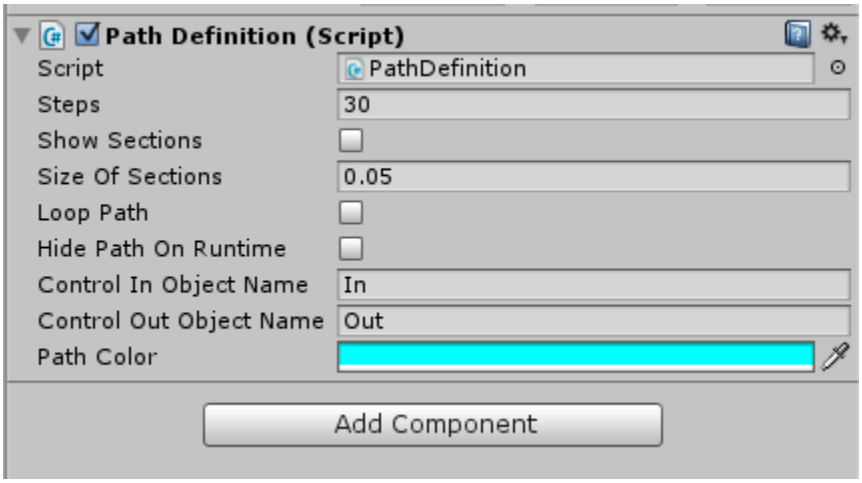
Object Tag Name: Tag name of the object we are spawning.

As you can see on the images we just need to add this script component to our Spawner object and setup all the setting according to our needs.

How to Setup our own Path

Compatibility: This works for 3D and 2D projects.

To get started open the Scene called "**PathDefinitionDemo**" You will see some objects but for now select the Object called "**Path Example**" or "**Loop Path Example**". Notice those two objects has the Script Component "**Path Definition**" and it should looks like this:



I will explain each field in the same order the above image shows.

[**Fields**]

Steps: How many steps the object need to reach next main point.

Show Sections: Show the sections of the path.

Size of Sections: Size of each section.

Loop Path: Make the Path infinite Cycle.

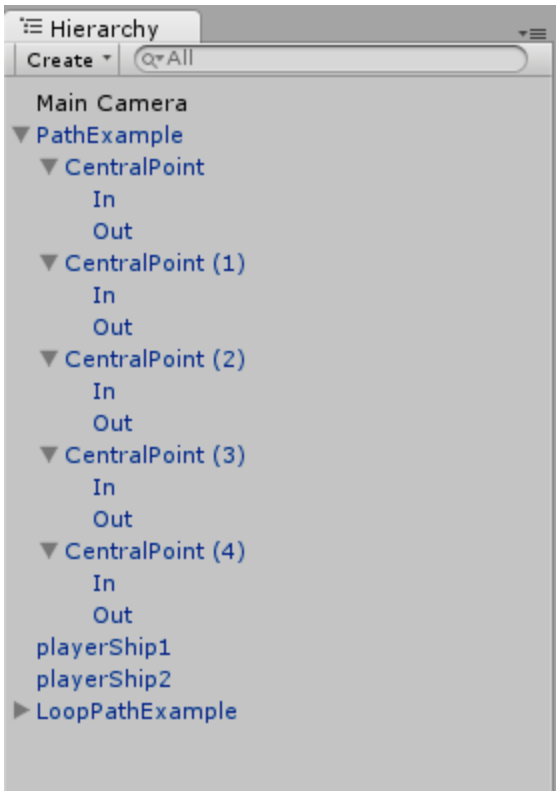
Hide Path On Runtime: Hide the Gizmos and sprites (if any) of the Path before game start.

Control In Object Name: Name of the children objects in the Main Points.

Control Out Object Name: Name of the children objects in the Main Points.

Path Color: Color of the Path we are making.

Now in order to fulfill that component we have to take a look at our Object, expand the game object selected before for example "**Path Example**" and it should have something like this:



- The object called "**Path Example**" is the one who has the script component.
- The object in children called "**Central Point**" is the main point where each line of the path will connect.
- The objects in children called "**In and Out**" of each "**Central Point**" are the one will give curved form to the path.

So on Script Component the fields called "**Control In Object Name**", "**Control Out Object Name**" need to have the same name as the children objects of each "**Central Point**".

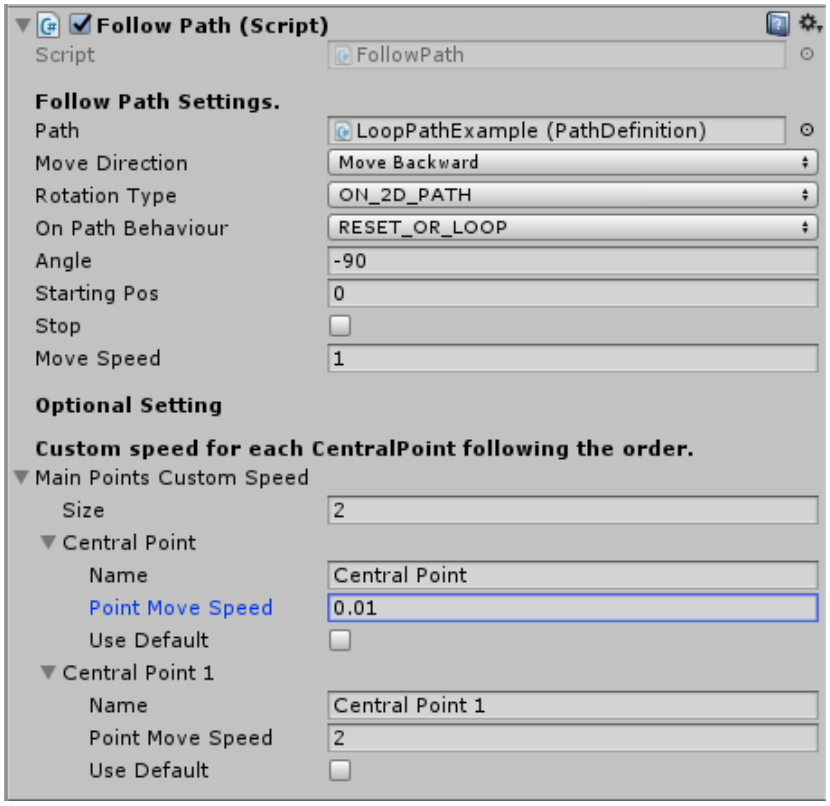
Tip: To make easier the setup use the Path Prefab I give you and duplicate the Central Points and move then wherever you want.

Reset the Curves: Set the children objects "**In & Out**" to (0, 0, 0) position.

To make an object **follow the path** we made, we have to add the Script Component called "**Follow Path".**

- So, select the object you want to follow the path and add the script.

It should look like this:



Object PlayerShip1 Example…

I will explain each field in the same order the above image shows.

[**Fields**]

Path: Drag the Object with the Path Definition Script in order to follow that path.

Move Direction: What direction you want to take on a loop path **Backward** or **Forward**.

Rotation Types: Rotate while running on Path. Perfect for smooth animation.

- **NONE:** Don't rotate while running along the path.
- **ON_2D_PATH:** To run 2D Paths (2DPathDefinitionDemo) Scene as example.
- **ON_3D_PATH:** For 3D-Objects running a 3D Path (3DPathDefinitionDemo) Scene as example.

Angle: Default Angle/Orientation/Rotation of the Sprite or 3D Object before running a path.

Starting Pos: My starting position on path.

Stop: Stop object on his actual position.

Destroy Once Path End: Destroy object once reach the full path. (Won't work if loop field is activated).

Move Speed: How fast the object moves on Path.

Optional Setting

It will assign by Path Definition object order.

Size: How many Main Points will you apply a custom setting.

Element: Each Element has the following attributes.

- **Name:** Name of the element to identify yourself the object.
- **Point Move Speed:** Custom movement speed between this object and the next one.
- **Use Default:** Will use default movement speed assigned on **movespeed field** no matter what**.**

And that's all easy right?

Contact Information

- You can contact me by my e-mail: **guilei4910@yahoo.com** **Name: Guille Bauzá.**
- Don't forget to give us your feedback: Plugin Page.

FAQ's

- **Why does my object slowdown near central points?**
  - ***Please separate the sections between the central points, also using less section help as the object calculate less positions to reach.***