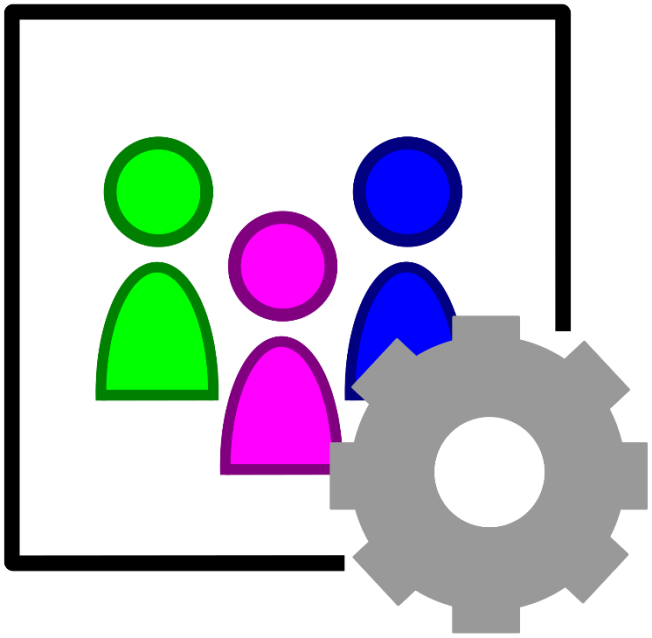


Multipurpose Spawner System

by Daimon Creative ©

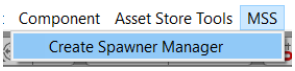
ver. 1.1

Content

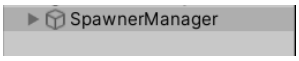
1. What is the Multipurpose Spawner System (MSS)?

The Multipurpose Spawner System is a small tool for Unity which allows you to create and configure spawners through a single manager. MSS was specially designed for beginners and non-programmers or developers who just want to save time. It provides a userfriendly UI by giving the developer as many options as possible.

2. Getting started

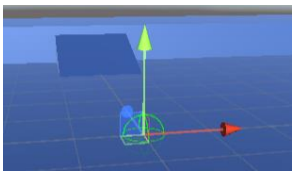


To get started with MSS you first need to create a spawner manager. You can do this by clicking on “MSS/Create Spawner Manager” in the upper menu bar.

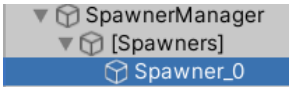


A new GameObject called “SpawnerManager” should now appear in the hierarchy.

To create a spawner, simply select the SpawnerManager-GameObject and click “Create Spawner” in the inspector.



You will notice a green wired sphere appearing in the scene. This shows the position of the spawner and can be moved in the scene to set the spawn position.

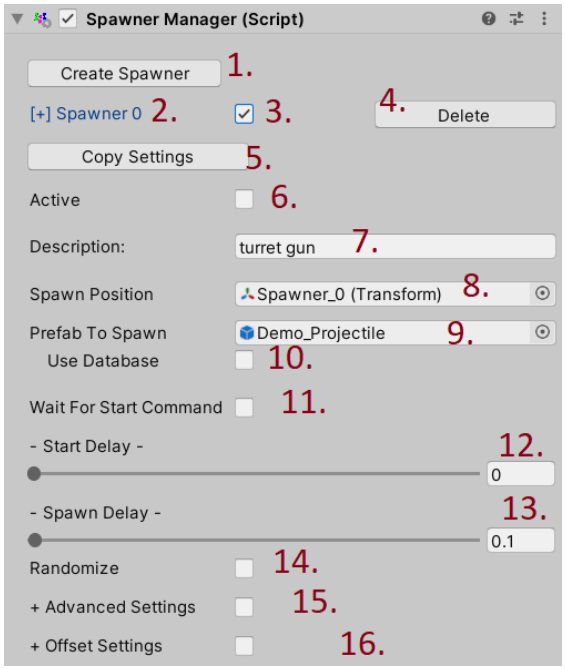


To select a spawner simply open the SpawnerManager and the Spawners-Container-Object in the hierarchy. There you will find all spawners listed with their numbers. Select them to move them around.

3. Inspector view – Spawner Manager

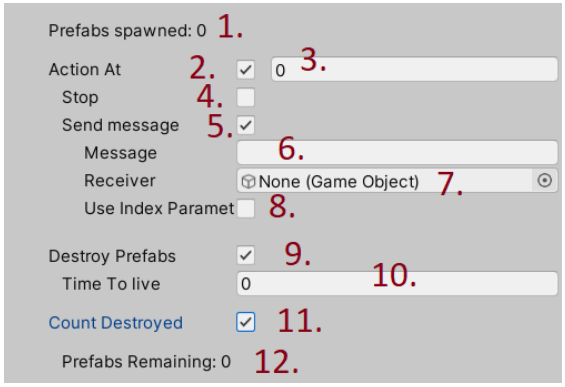
Basic Settings

1. Create new spawner
2. Spawner name
3. Show/Hide spawner settings
4. Delete spawner
5. Copy spawner settings to clipboard
6. Set spawner active/inactive
7. Use this field to add a description to your spawner
8. Empty GameObject operating as Spawn-position
9. Drag and drop prefab here (select spawner tag here when using database)
10. Check if you want the spawner to use prefab database
11. If enabled, the spawner will only start spawning when another script calls StartSpawner()
12. Sets the delay before first spawn
13. Sets the delay between every spawn executed after start delay
14. If enabled, randomize spawn delay
15. Show advanced settings
16. Show offset settings



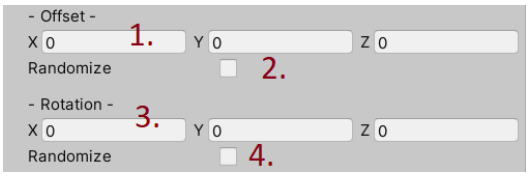
Advanced Settings

1. Counter showing prefabs spawned
2. Execute action whe counter reached…
3. …value
4. Action: Stop spawner
5. Action: Send message
6. Message to send as string
7. GameObject to send message to
8. Send parameter (spawner number) when sending message
9. Destroy prefabs after spawning
10. Time to live in seconds
11. Count destroyed prefabs that have been instantiated by this spawner
12. Counter showing how many prefabs still are in the scene



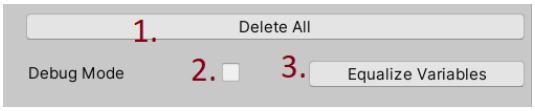
Offset Settings

1. Offset in Vector3
2. Randomize offset
3. Rotation in Vector3
   (will be multiplied with current rotation)
4. Randomize Rotation



Debug/Other Settings

1. Delete all spawners
2. If enabled Unity's console will print out every spawn executed by MSS with relevant data



Symbols/Icons in inspector

+ or [+] - show/hide settings

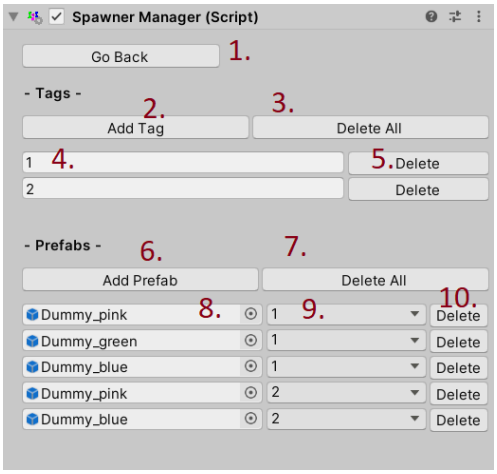
Note: Variables/settings that are marked with these symbols are still active when hidden.

4. Inspector view – Prefabs Database

If you want your spawners to instantiate different prefabs, for example different types of enemies you need to check the "use database" option.

Click on edit to switch to database.

1. Go back to general settings
2. Add new tag
3. Delete all tags
4. Enter tag name here
5. Delete this tag
6. Add a new prefab to database
7. Delete all prefabs from database
8. Drag and drop your prefab here
9. Select a tag to assign it to the current prefab
10. Delete this prefab



Note: Tags created in the database are not equal to Unity's default tags. They are just variables helping you to categorize your prefabs. Later on we can set the database tag in our spawner settings, to let the spawner pick the right prefabs.

5. Manually start and stop spawner

Sometimes it might be useful to manually start and stop single spawners. For these cases the SpawnerManager comes along with two pre-defined functions.

Public void StartSpawner(int index){…} -> enables the spawner with the given number

Public void StopSpawner(int index){…} -> disables the spawner with the given number

You can simply call them up by sending a message or call the function directly. Don't forget to give away an index number to allow the manager to select the right spawner.

Example 1 – via SendMessage:

```
public GameObject manager;

void Start(){
    manager.SendMessage("StartSpawner",1);
}
```

Note: Keep in mind that if you want to stop a specific spawner you can use the "Action at/Stop"-function in the "Advanced Settings"-section in the inspector.

Example 2 – via direct Access

```
public SpawnerManager manager;

void Start(){
    manager.StartSpawner(2);
}
```

Exception! – When calling a function via SendMessage with 0 as parameter there may occur an error saying that there is no parameter given. To avoid that you have to set the option RequireReceiver in your sendMessage-command to tell your script to handle the value 0 like a variable.

6. Example: Endless spawner

Now that we know the basics of MSS, we want to create an endless spawner for training purposes. We already created a spawner in chapter 2 “Getting started”.

We want the spawner to instantiate a simple object which will be destroyed after two seconds and to loop this task with a delay of two seconds. You can use the demo spheres in the package for this purpose.

Navigate to the folder with the demo prefabs first (MSS/Demo/Prefabs/Sphere/…). Choose a prefab you like and drag it into the “Prefab To Spawn” field in the Spawner Manager. Then continue by entering the delay in the “Spawn Delay” field right next to the slider – in this case we enter the value 2.0.

Now check the box “Advanced Settings” to see more options. Go to the box “Destroy Prefabs” and check it too. In the appearing field “Time to live” enter the value 2.

That’s it! Hit play to test the spawner!

7. Demo Package

The demo package contains a mini tower-defense game, that shows you some of the possibilities of MSS. It also includes some simple scripts which can be used freely. You can find the demo scene under “MSS/Demo/Demo.unity”. If you don’t want the demo package to be installed in your current project, simply uncheck the folder “Demo” in the import settings.

Questions, Feedback and News


www.instagram.com/daimon_creative


www.youtube.com/channel/UCW05ecZj9EPKDQWcyj7bJ_Q


daimon.creative@gmail.com

I apologize for my bad English, but I hope you understood the basics of MSS and are now able to use it for your own projects. Feel free to contact me if you have any questions! I wish you a nice and productive day!

Thank you for choosing Multipurpose Spawner System!