



## Integration Guide

Version 1.4.0

03/24/24

**magique**  
productions ltd

# Table of Contents

Spawner.....	3
Sky/Weather Systems.....	3
Emerald AI 2024.....	3
Emerald AI Prior to 2024.....	3
Invector FSM AI.....	5
Game Kit Controller AI.....	8
Quest Machine.....	8
Dialogue System.....	12
Pixel Crusher's Save System.....	13
Core GameKit.....	14
Map Magic 2.....	14
Landscape Builder.....	16
Polaris Low Poly Terrain.....	16
Global Snow.....	16
NetCode for GameObjects.....	17
Mirror.....	17
Other Systems.....	18
Artificial Intelligence System.....	19
Support.....	20

# Spawner

This section contains guidance on each integration available for the SoulLink Spawner System

## ***Sky/Weather Systems***

Before using your preferred sky/weather system, go to the Tools/SoulLink/Integration window and ensure that the integration is turned on. No other steps need to be made in order for the integration to work.

## ***Emerald AI 2024***

With the release of Emerald AI 2024, you no longer have to make any code changes to Emerald AI in order to use it with Spawner.

### **Spawnable\_EmeraldAI24.cs**

A custom Spawnable component called Spawnable\_EmeraldAI24 is provided with SoulLink Spawner. Attach this component to all your AI prefabs. Do not use the base Spawnable component.

**NOTE: If you have Auto Add Spawnable Interface checked in SoulLinkSpawner then this component will automatically be added to your prefabs when you reference them in the Spawner interface.**

## ***Emerald AI Prior to 2024***

The following are code changes that should be made in order for Spawner to work correctly with legacy Emerald AI and PoolManager. **Do not make these changes if you are using Emerald AI 2024 or later.**

### **EmeraldAIBehaviors.cs**

Add the following at the top of the AttackState method:

```
// Magique fix  
if (!EmeraldComponent.m_NavMeshAgent.enabled) return;
```

Add the following at the top of the DefaultState method:

```
// Magique fix  
if (EmeraldComponent.m_NavMeshAgent.enabled == false) return;
```

Add the following at the top of the BackupState method:

```
// Magique fix  
if (EmeraldComponent.m_NavMeshAgent.enabled == false) return;
```

### **EmeraldAIInitializer.cs**

Comment out all code in the Start method:

```
void Start()
{
    //Invoke our AI's On Start Event
    // Magique Fix
    //EmeraldComponent.OnStartEvent.Invoke();
}
```

Add the following at the top of the Update method:

```
// Magique Fix
if (EmeraldComponent == null) return;
```

### **EmeraldAIDetection.cs**

Add the following to the top of the SetDetectedTarget method:

```
// Magique Fix
{
    if (!EmeraldComponent.m_NavMeshAgent.enabled || DetectedTarget == null) return;
    var aiSystem = DetectedTarget.GetComponent<EmeraldAISystem>();
    if (aiSystem != null && aiSystem.HitPointTransform == null) return;
}
```

In the FixedUpdate method, put this at the very top:

```
if (EmeraldComponent.m_NavMeshAgent == null) return; // Magique Fix
```

### **EmeraldAISystem.cs**

Comment out the call to Initialize in the Awake method as follows:

```
// Magique Fix
// EmeraldInitializerComponent.Initialize();
```

Add the following at the top of the Update method:

```
// Magique fix
if (m_NavMeshAgent == null) return;
```

Add the following at the top of the Wander method:

```
// Magique fix
if (!m_NavMeshAgent.enabled) return;
```

Add the following to the top of the Damage method:

```
// Magique Fix
if (m_NavMeshAgent.enabled == false) return;
```

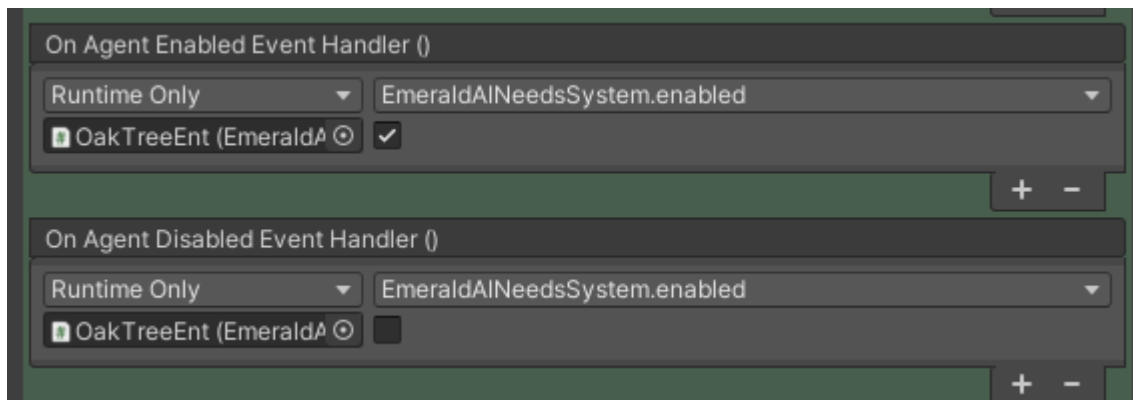
### **Spawnable\_EmeraldAI.cs**

A custom Spawnable component called Spawnable\_EmeraldAI is provided with SoulLink Spawner. Attach this component to all your AI prefabs. Do not use the base Spawnable component.

**NOTE: If you have Auto Add Spawnable Interface checked in SoulLinkSpawner then this component will automatically be added to your prefabs when you reference them in the Spawner interface.**

### ***NOTE: Regarding the Emerald AI Needs System***

If you have the Emerald AI Needs System attached to your prefab, you need to disable the component and add event handlers on the Spawnable\_EmeraldAI component to enable/disable this system when the agent is enabled/disabled.



## ***Invector FSM AI***

### **Version prior to 1.1.8b**

If you are using Invector FSM AI prior to Version 1.1.8b, do the following:

In vAIMotor, add the following method:

```
public void ResetDeathBehaviour()
{
    _rigidbody.isKinematic = false;
    _capsuleCollider.enabled = true;
    triggerDieBehaviour = false;
}
```

```
    animator.SetBool("isDead", false);
    animatorStateInfos = new vEventSystems.vAnimatorStateInfos(animator);
    animatorStateInfos.RegisterListener();
}
```

and Uncomment the code in `Spawnable_InvectorFSMAI` in the `OnDespawned` method as follows:

```
// Uncomment the following code if you are using Invector FSM AI version prior to 1.1.8b
var aiMotor = GetComponent<vAIMotor>();
if (aiMotor != null)
{
    aiMotor.ResetDeathBehaviour();
}
```

#### Version 1.1.8b or Later

No code changes are required to integrate with 1.1.8b or later. Proceed to the next steps.

#### **HeadTrackSensor Issue**

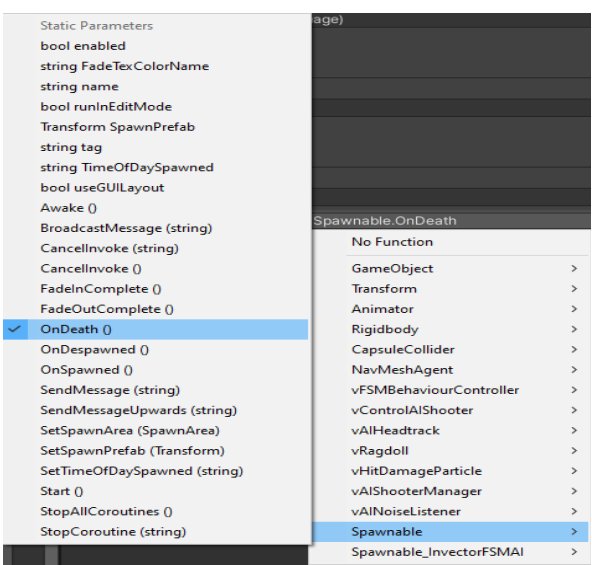
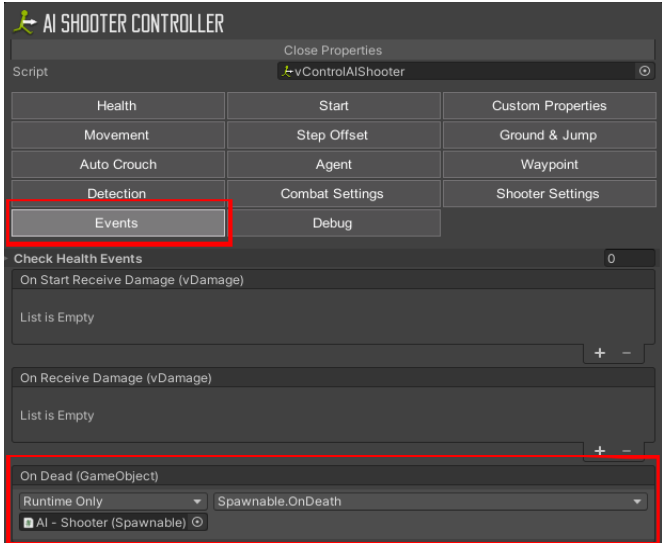
It is a known issue that the `HeadTrackSensor` collider can cause spawns to fail with a slope angle check. To fix this issue, add the `HeadTrack` layer to the `Nav Mesh Ignore Layers` in `Nav Mesh Options`.

#### **Spawnable\_InvectorFSMAI.cs**

A custom `Spawnable` component called `Spawnable_InvectorFSMAI` is provided with `SoulLink Spawner`. Add this component to your `Invector AI` prefabs. If you have `Auto Add Spawnable Interface` checked in `SoulLinkSpawner` then this component will automatically be added to your prefabs when you reference them in the `Spawner` interface.

#### **Add OnDead Event Handler**

On your `Invector AI` component, you need to add an event handler for the `OnDead` event. Reference the `Spawnable` component attached to your `AI` and select the `OnDeath` method.



## Game Kit Controller AI

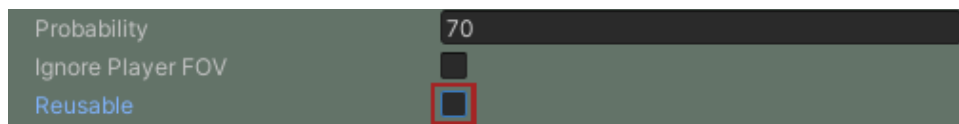
To use GKC AI integration, first go to Tools/SoulLink/Integrations and enable Game Kit Controller.

### Spawnable\_GKCAI.cs

A custom Spawnable component called Spawnable\_GKCAI is provided with SoulLink Spawner. Add this component to your GKC AI prefabs. If you have Auto Add Spawnable Interface checked in SoulLinkSpawner then this component will automatically be added to your prefabs when you reference them in the Spawner interface.

### *Important Notes*

Object Pooling is not currently supported in GKC so this integration provides a workaround for using GKC AI with Spawner. In order to use properly, when you add your spawns to SoulLink Spawner, you must uncheck the Reusable flag.



Do not use SpawnAction\_Fader on Spawnable\_GKCAI because this feature is not compatible with GKC AI.

For best results, ensure that your population caps are high enough so that objects will be used from the object pool for as long as possible. When the pool runs out, new instances will be generated at runtime, which can result in small performance spikes.

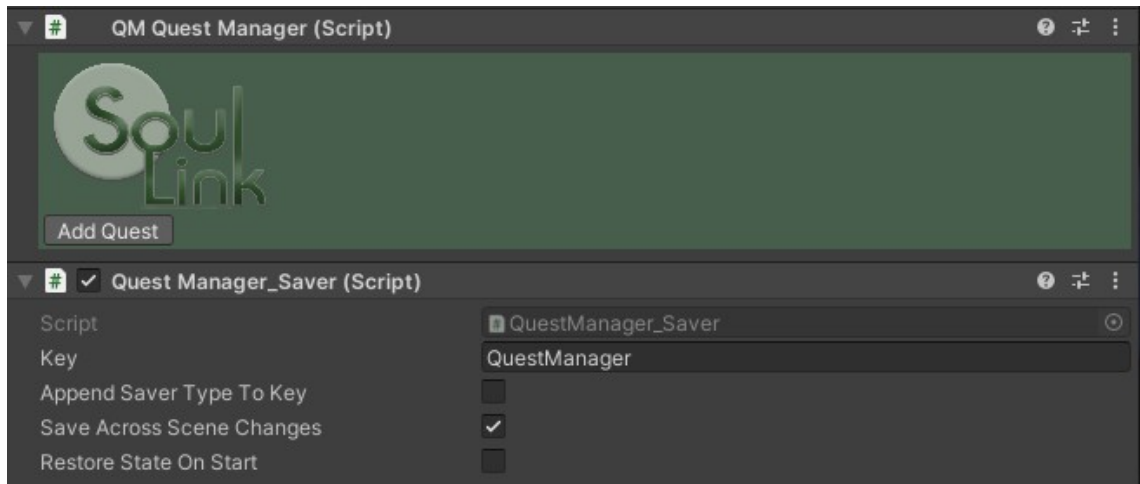
## Quest Machine

To use Quest Machine features with Spawner, first go to Tools/SoulLink/Integrations and enable Quest Machine.

Now, drag the Quest Machine prefab into your scene found in Plugins/Pixel Crushers/Quest Machine/Prefabs. Be sure to edit your Quest Machine database with any quests that you want to use with Spawner.

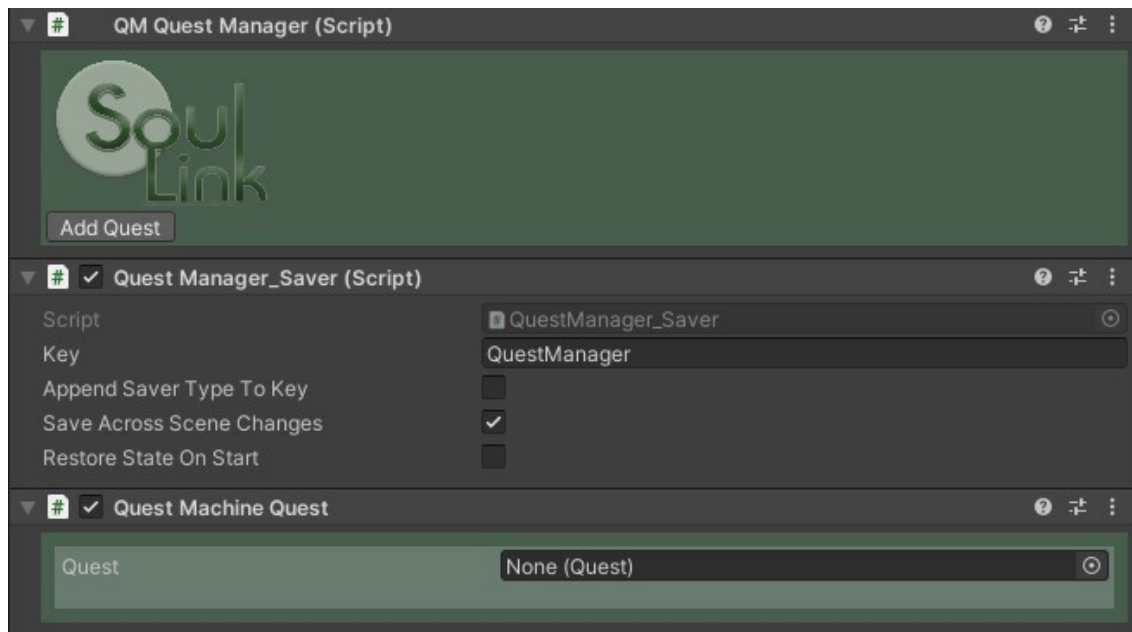
Click on Tools/SoulLink/Spawner/Setup Scene and the system will automatically add a QuestManager object under the SoulLink Global object.





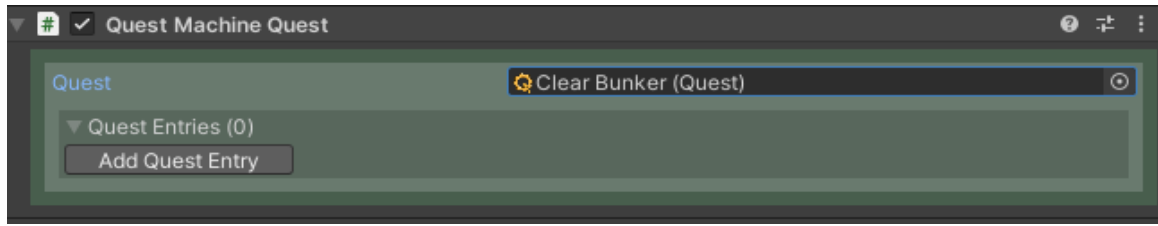
The object will automatically have a saver component added to it if you've added save system integration from the Tools menu.

Now, for each quest you want to track with SoulLink, click the Add Quest button. A Quest Machine Quest object will be added as follows:

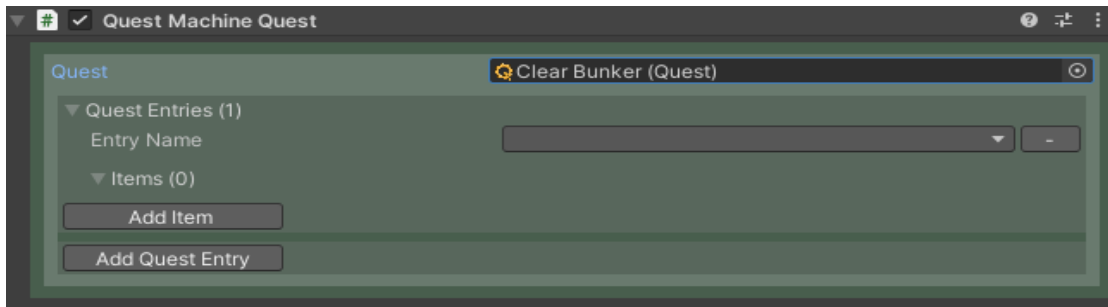


### Quest

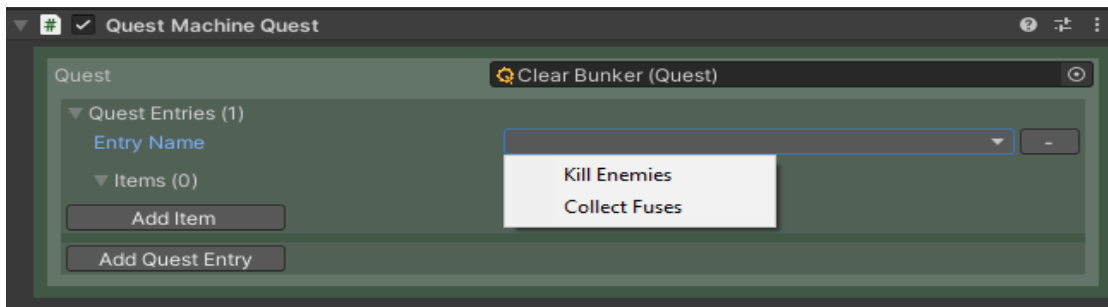
Drag a Quest Machine Quest object into this field.



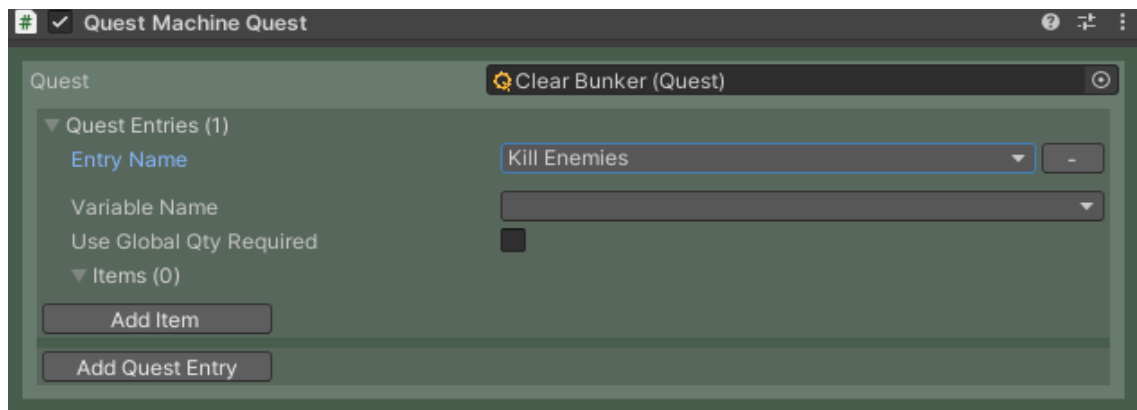
Click Add Quest Entry to add a quest entry.



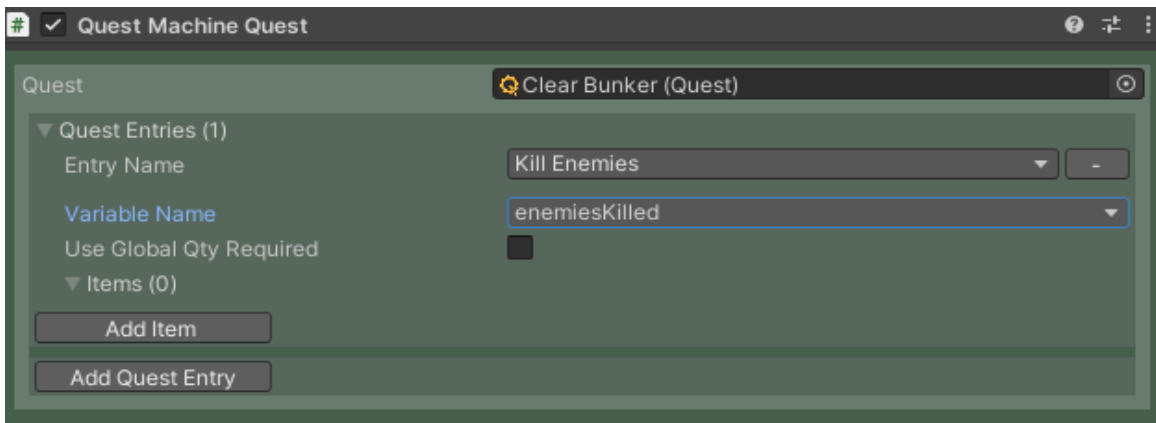
Click on the Entry Name drop-down list and you should see a list of available entries.



Select an entry from the drop-down list.



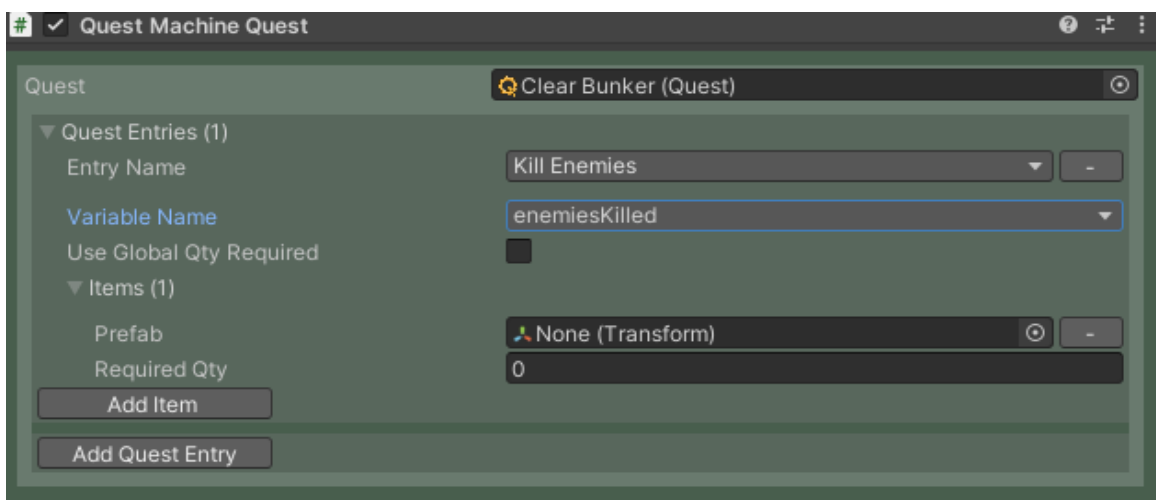
Select the variable name from the drop-down list.



### **Use Global Qty Required**

Check this box if you wish to track the item quantities collectively instead of by the value defined by the Quest Machine variable name.

Now click Add Item.

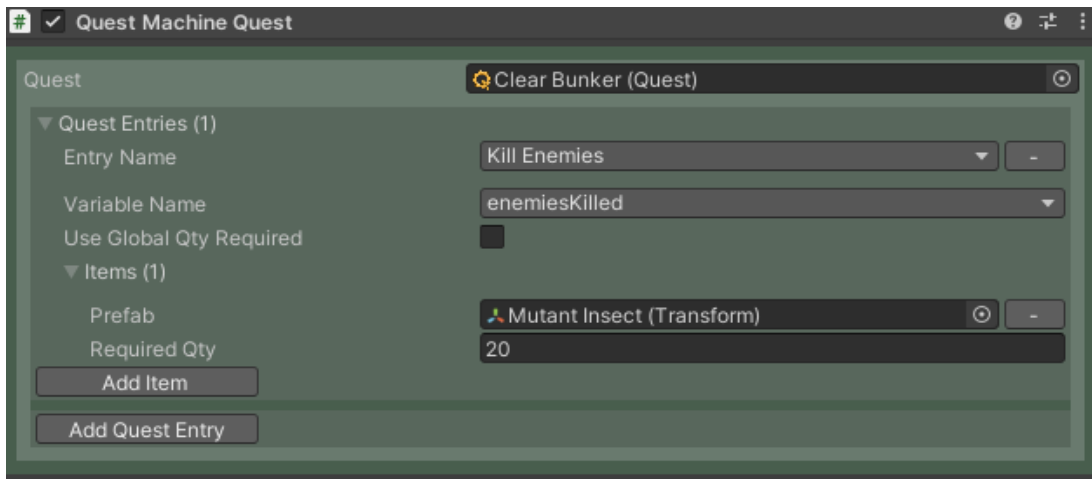


### **Prefab**

Drag a prefab into this field. In this example, we are tracking a number of enemies killed so I want to drag a Spawnable AI prefab into this field that will count towards the total enemies killed when that spawn is defeated.

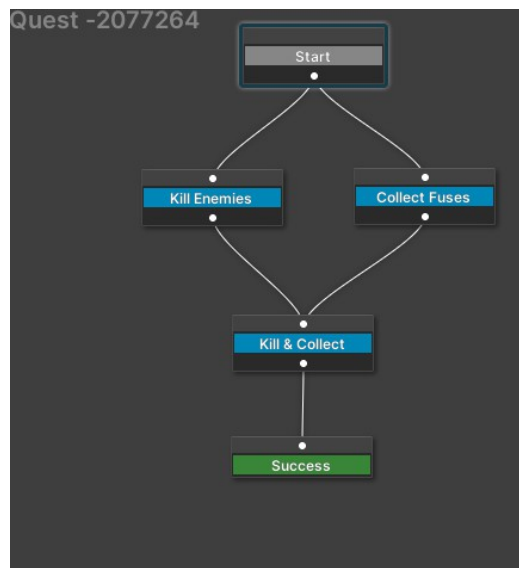
### **Required Qty**

This indicates how many of this AI must be defeated in order to complete this item entry.



In this example, we've added the Mutant Insect prefab with a quantity of 20. So, the player must kill 20 Mutant Insects for this quest entry to be completed.

For the QM Quest example illustrated above, the following is the Quest Machine diagram. Note that our example only showed the Kill Enemies entry. This quest also requires a Collect Fuses entry.



## ***Dialogue System***

To use Dialogue System features with Spawner, first go to Tools/SoulLink/Integrations and enable Dialogue System.

Now, drag the Dialogue Manager prefab into your scene found in Plugins/Pixel Crushers/Dialogue System/Prefabs. Be sure to edit your Dialogue System database with any quests that you want to use with Spawner.

Click on Tools/SoulLink/Spawner/Setup Scene and the system will automatically add a QuestManager object under the SoulLink Global object.

Now, for each quest you want to track with SoulLink, click the Add Quest button. A Dialogue System Quest object will be added

Fill out the Dialogue System Quest component similar to the method described in the Quest Machine integration. There are 2 differences as follows:

### **Quest Name**

The Quest Name field is a drop-down list of available quests defined in Dialogue System. Simply select which quest you are tying this component to.

### **Entry Name**

The entry name must be entered manually and is simply a description of the entry.

## ***Pixel Crusher's Save System***

If you have either Dialogue System or Quest Machine installed and you've set up Pixel Crusher's save system in your scene then you can take advantage of saving and loading spawns and spawn areas for Spawner.

If you have an existing scene using DS or QM then you can update the scene to include the save system integration by going to Tools/SoulLink/Spawner/Setup Save System Integration. This will automatically add the required saver components to your scene objects. You will be asked if you want to regenerate unique Ids. Always says yes unless you are managing your individual saver component key values manually. Each component needs a unique value to save correctly and allowing Spawner to generate or regenerate these is good practice. This can be especially useful if you have SpawnArea components on prefabs and you update the prefab, which can cause the same saver key to be propagated to all the areas in your scene. By re-running this setup option and regenerating the ids, you will fix this issue.

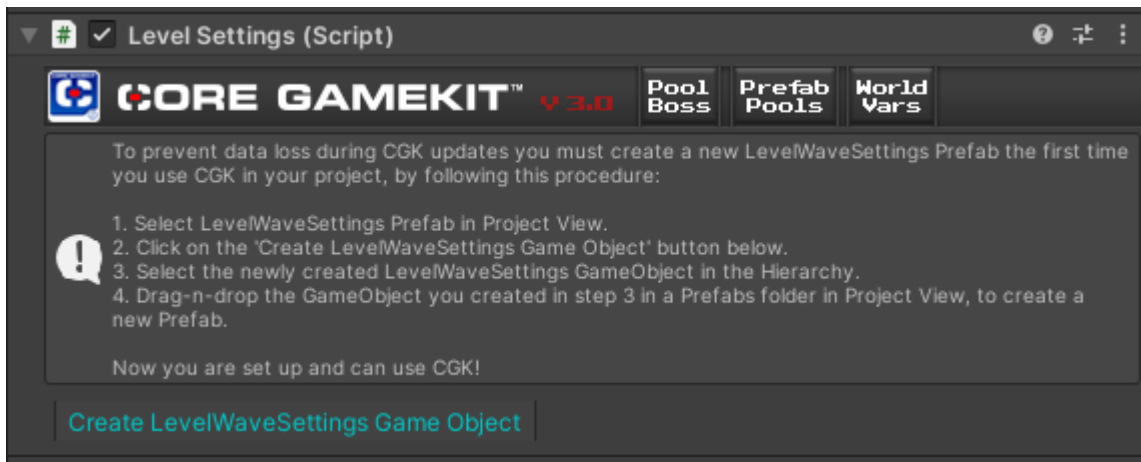
If you are adding on to an existing scene then the saver components will automatically get added to your scene objects as necessary. So, if you add a new SpawnArea, it will automatically add the SpawnArea\_Saver component.

## Core GameKit

To use Core GameKit PoolBoss with Spawner, first go to Tools/SoulLink/Integrations and enable Core GameKit. Then do the following in your scene:

### Add Core GameKit level manager settings

Go to Plugins/DarkTonic/CoreGameKit/Prefabs folder and select the LevelWaveSettings prefab. In the Inspector you will see the following:



Click on the Create LevelWaveSettings Game Object button to create an instance in your scene.

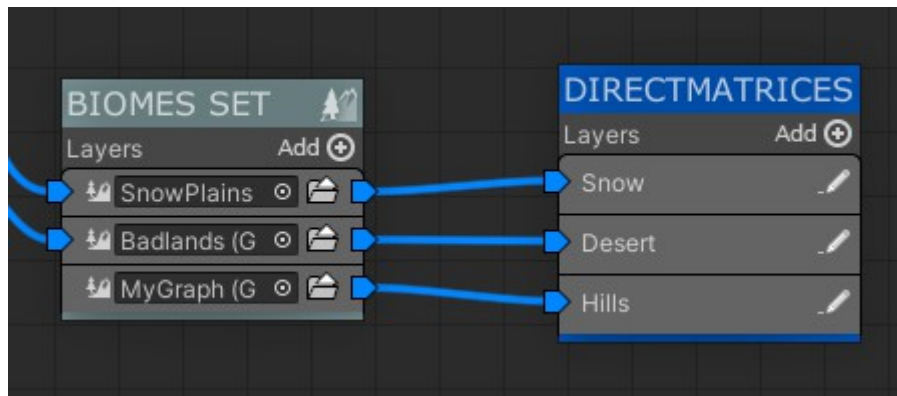
Be sure to activate Core GameKit in the Tools/SoulLink/Integrations window.

## Map Magic 2

If you are also using MapMagic 2, do the following:

### Step 1: Add a MapMagic map with biomes in your scene

Create your MapMagic map with a Biomes Set. From the Biomes Set, export a direct matrices node by using the Add button on the Biomes Set and selecting Map/Output/DirectMatrices. Rename the layers in DIRECTMATRICES as you see fit.



## **Step 2: Prepare Scene Objects**

*Note: This step is only required if you are using infinite terrains and cannot bake your nav mesh manually. If you are not using infinite terrains and are baking your nav mesh manually, you do not need to do anything further in order to use Spawner with MM2.*

In order for Spawner to properly generate nav meshes for the AI at runtime, all objects in the scene that need to be walkable should have the SoulLinkNavMeshSourceTag component attached. If your MapMagic map is placing objects then ensure that the prefabs for those objects have this component attached. Note, this does not apply to the terrain tiles. Terrain tiles in the scene and those added at runtime via MapMagic will automatically have the script attached.

In addition, for Trees to be excluded from navigation so that AI do not walk through them, attach the NavMeshModifierVolume component to your tree prefabs and set to Not Walkable. Ensure that the volume size is a good fit for the tree trunk. Note, however, that for trees to work with this method, they must be placed in the scene as objects with MapMagic and not as Unity terrain trees.

## **Alternative to Biomes**

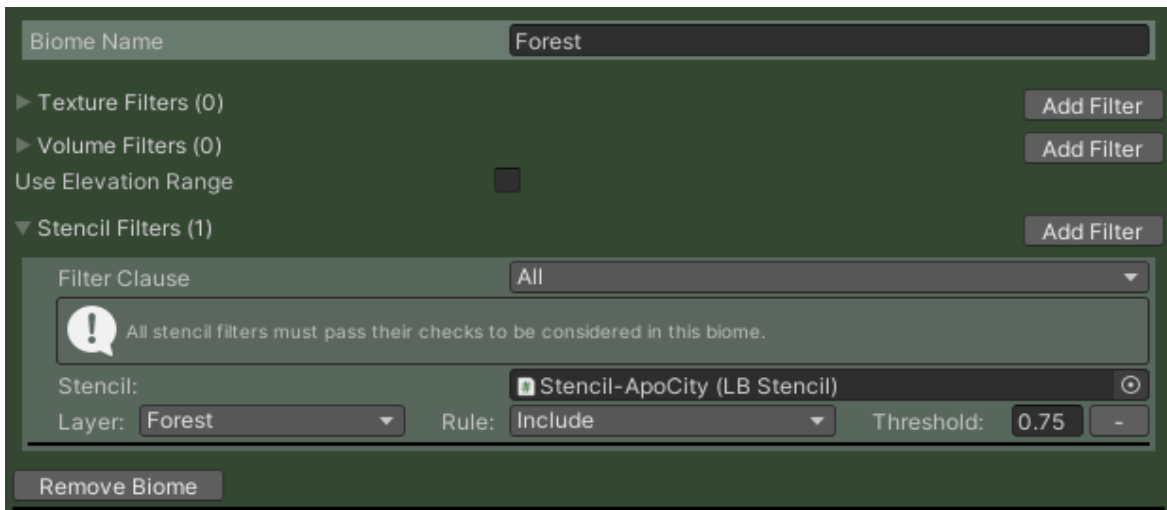
If you are not using Biome Sets with MapMagic 2, you must attach a DirectMatrices node to each texture you wish to be considered a biome. Spawner will recognize them the same it would for Biome Sets.

## ***Landscape Builder***

Integration with Landscape Builder is automatic. You only need to ensure that you have at least Landscape Builder version 2.4.1. Older versions do not have full support for SoulLink Spawner.

### **Stencil Biomes**

Integration with landscape Builder allows you to define Biomes by Stencil layers. You may continue to use texture filters as well.

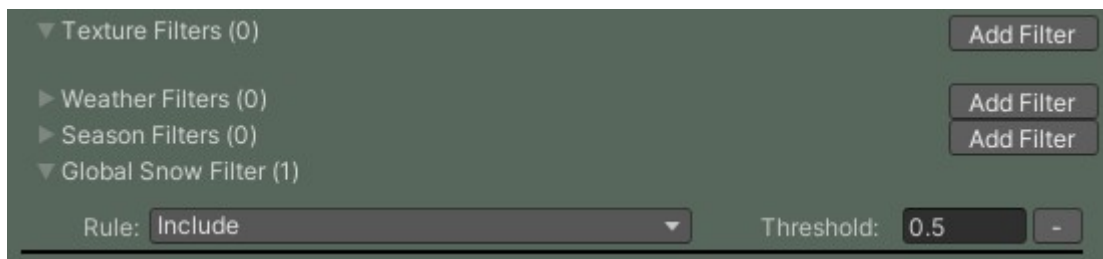


## ***Polaris Low Poly Terrain***

Polaris integration is automatic and will behave the same as Unity terrain. However, the integration requires Splats in order to work. If your terrain does not have splats then you will be notified in a warning so you can make corrections.

## ***Global Snow***

Global Snow, by Kronnect, integration allows you to use snow filters for your spawns:





## **NetCode for GameObjects**

Integration with NetCode for GameObjects provides multiplayer support for Spawner.

First, install NetCode for GameObjects from the Unity Package Manager.

Then, install the NetCodeTesting Unity package found in SoulLink/Integrations/Spawner/NetCode folder.

After installing NetCode and the testing package, turn on the integration through Tools/Spawner/Integrations.



Go to the NetCodeTesting/Scenes folder and open the demo scene. On the NetworkManager component make sure you have a Transport protocol selected and have assigned the sample NetFPSPlayer prefab to the Player Prefab field.

Run the scene and select Server to test a player running on the host system.

To test Client, add the demo scene to your build settings and Build the project. Run the project in the Unity editor as Server and run the built executable as Client.

For more details. Please watch the tutorial video found on this page:

<https://www.magiqueproductions.com/soullink-spawner/>

## **Mirror**

Integration with Mirror provides multiplayer support for Spawner.

First, install Mirror from the Unity Package Manager.

Then, install the MirrorTesting Unity package found in SoulLink/Integrations/Spawner/Mirror folder.

Go to the MirrorTesting/Scenes folder and open the demo scene. On the NetworkManager component make sure you have have assigned the sample NetFPSPlayer prefab to the Player Prefab field.

Run the scene and select Host (Server + Client) to test a player running on the host system.

To test Client, add the demo scene to your build settings and Build the project. Run the project in the Unity editor as Host (Server + Client) and run the built executable as Client.

For more details. Please watch the tutorial video found on this page:

<https://www.magiqueproductions.com/soullink-spawner/>

## ***Other Systems***

Various other systems have been tested and known to work with SoulLink Spawner, but require some guidance to use correctly. Some of these systems include the following:

### World Streamer 2

To work correctly, the navmesh needs to be baked and available in the Game Streamer scene. To do this, have both your work scene and your streamer scene open at the same time when you perform your nav mesh bake. Once completed, save both scene. You can now load just the game streamer scene and have the associated nav mesh information available for spawning.

SECTR Stream  
Behavior Designer

## **Artificial Intelligence System**

This section contains guidance on each integration available for the SoulLink Artificial Intelligence System. This sections is for future expansion when the SoulLink AI system is released.

## Support

Please e-mail support or visit the discord support channel to receive guidance on using these systems if you run into problems.

Support E-Mail

[support@magiqueproductions.com](mailto:support@magiqueproductions.com)

Discord Support

<https://discord.gg/dFMQxEn4kQ>