



Table of Contents

Section One: Pool Boss Overview	2
Section Two: Miscellaneous Other Scripts	4
Section Three: Code Usage	5
Section Four: Playmaker Integration	5
Section Five: Using JavaScript (UnityScript)	5
Conclusion	5

Pool Boss - Getting Started

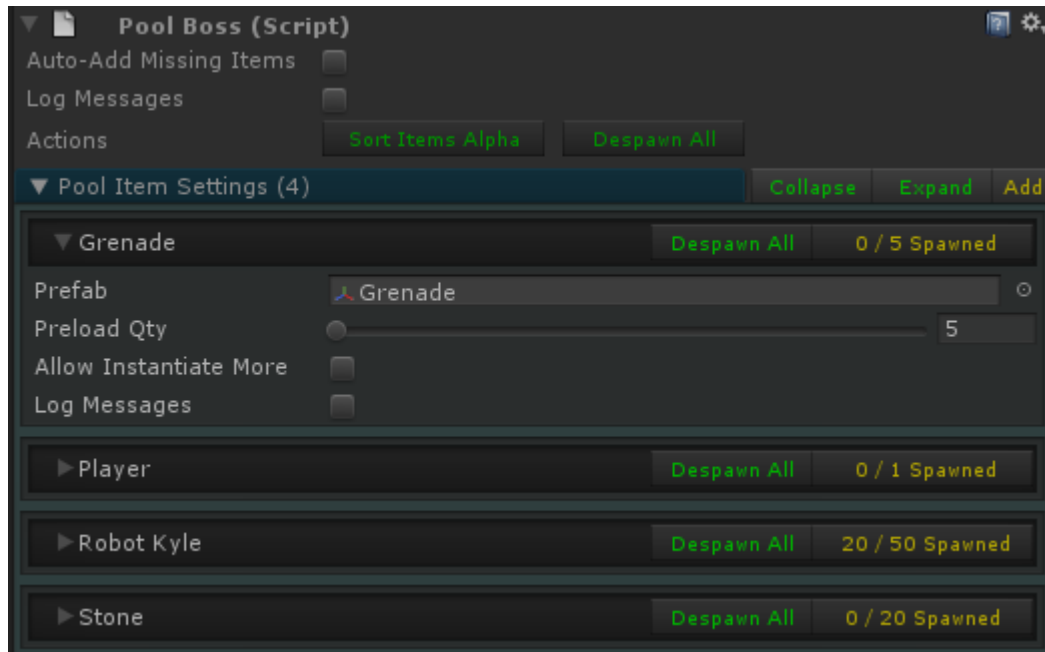
Congratulations on your purchase of Dark Tonic's Pool Boss! You now have a very flexible yet easy to use object pooling system. Pool Boss is part of the popular [Core GameKit](#) plugin by Dark Tonic, which also includes a full combat system, spawners and level setup tools! Check it out if you need any of those features.

To start, find the PoolBoss prefab in Project View under DarkTonic/PoolBoss/Prefabs and select it. Then click the "Create Pool Boss Prefab" button to create one in your Scene.

Note: It's important that you never drag out a copy into your Scene because if you do, the next time you update the plugin you will lose all your pool items. This is a Unity prefab problem. When you drag a prefab into a Scene, the Scene copy is connected to the one in the plugin folder. The button we have provided disconnects your Scene's Pool Boss copy from the source.

Section One: Pool Boss Overview

Pool Boss is a prefab pre-loader / recycler similar to Pool Manager. If you're familiar with Pool Manager, you will have no trouble using Pool Boss. Here's a screen shot.



If you are unfamiliar with the reasons for using a pooling solution, here are the basics. Instantiate and Destroy calls (which are what normally happen if you don't use pooling) can cause stuttering when complex objects are created (and destroyed) during game play, especially on mobile devices. A pooling solution will create everything up front when the Scene starts, and everything not needed at start will begin in a despawned state initially. To set this up, you simply specify which items you want in that pool, and how many of each. At runtime, the pool items are disabled GameObjects when despawned and are enabled when you spawn them. Performance is massively affected and more predictable by a system such as this. The settings are as follows:

1. Top-level controls
 - a. Auto-Add Missing Items - checking this box will actually allow you to "half-pool", meaning that you can check the box and hit play without adding any Pool Items. What will happen is that nothing will be created up front, but Pool Items will be created (Instantiated - bad for performance) as you need them, and can be reused. Also, any prefab that is in the Hierarchy (already in the Scene at startup and doesn't need to spawn) will get despawned in to the pool when it goes away so you can re-use it later in that Scene. If you do not check this box, any prefabs you try to spawn that don't have a matching Pool Item will instead log an error and nothing will be able to spawn without setting up the Pool Items. This is normally only for the prototyping phase, then when you know how many of what you want to pool, you can set them up.

- b. Log Messages - checking this box will tell you when each prefab spawns, despawns, or if Pool Boss needs to Instantiate an extra copy because you are already using all copies of the prefab.
 - c. Sort Items Alpha - clicking this button will sort the Pool Items A-Z according to their name.
 - d. More buttons are available at runtime:
 - i. Despawn All - this will despawn all active Game Objects that are set up in Pool Boss.
- 2. Per Item Settings
 - a. Use arrows to move an item up or down in the list.
 - b. Click the Add button to add a new blank item, then drag the prefab in from Project view.
 - c. Click the Del button to delete a pool item.
 - d. Prefab - drag a prefab here to specify which prefab you want to pool.
 - e. Preload Qty - here you specify the amount of copies of the prefab to create when the Scene starts. They will all begin despawned (inactive in the Hierarchy) and will appear as a child of the PoolBoss prefab
 - f. Allow Instantiate More - this defaults to off. If you check this box, and there are no remaining copies of the prefab despawned, another copy will Instantiate (not good for performance). Leave this checkbox on if you aren't sure how many you need. Later you can adjust the Preload Qty and turn this off.
 - g. Item Limit - this is only visible and used when Allow Instantiate More is checked. This is used to put a limit on the amount of items that can be Instantiated after the Scene begins. This number will default to your Preload Qty.
 - h. Log Messages - checking this box lets you log messages for only the Pool Items you check instead of all Pool Items. Good for troubleshooting.
- 3. Notice the yellow text "20/50 spawned" in the screen shot above. That appears at runtime only to let you know how many are spawned and what the total number of all copies of the Pool Item is (spawned + despawned). This is also great when you have turned on Auto-Add Missing Items. You can play the busiest section of your game, and just before pressing stop, pause the game and click on PoolBoss to take note of the totals so you can create or modify your Pool Items with those numbers in mind so you don't need to count them. Good time-saver! You can also click on this text to select all the spawned items in the Hierarchy.
- 4. Also during runtime, more buttons will appear on each item row. These are the same as the same-named buttons in the top-level section, but apply only to the single prefab configured in that pool item.
 - a. Despawn All
- 5. Code usage - there are events broadcast whenever a prefab is spawned or despawned. Any script on any prefab (or sub-prefab) can perform actions in these event handlers. Just to keep things easier for people migrating from Pool Manager, I have named the events the same as they did.
 - a. OnSpawned - this code fires immediately after the prefab has spawned. Note that when not using pooling solutions, you put a lot of initialization code in the Awake or

Start event. However, you will want to move some or all of that code into the OnSpawned event so that it fires each time the prefab is respawned. This is because the Awake and Start events will not fire again when the prefab is respawned.

- b. OnDespawned - this code fires immediately before the prefab despawns. This should be cleanup code only, and it should not take longer than one frame to execute (starting a CoRoutine is not advised as it will probably not finish). When despawn occurs, the prefab will be re-parented as a child of the PoolBoss prefab under LevelWaveSettings automatically.

Code would look like this (C#):

```
void OnSpawned() {  
    // initialization code.  
}  
void OnDespawned() {  
    // cleanup code.  
}
```

Note: If you are using a pooling solution, never call Instantiate or Destroy! If you destroy a cloned prefab in the pool by code, it will no longer be available the next time Pool Boss needs it. Pool Boss will log an error if this happens so you can find your code that did it.

Note: If you want to spawn Particles or audio items, make sure to attach the Timed Despawner script to those prefabs and set the Life Seconds to the amount of time the audio should play or the amount of time the particle takes to disappear. Then these prefabs will return to the Pool. However, if you need a way to pool audio, we do recommend that you check out our flagship product [Master Audio](#).

Section Two: Code Usage

To use Pool Boss without Playmaker, you will need to use one line of code to spawn and to despawn. The complete API can be [found here](#). Here are the basics in C#.

```
PoolBoss.SpawnInPool(Transform transToSpawn, Vector3 position, Quaternion rotation);
```

This will spawn the prefab you pass in at the position and rotation you provide, with the Pool Boss prefab as its parent.

```
PoolBoss.SpawnOutsidePool(Transform transToSpawn, Vector3 position, Quaternion rotation);
```

This will spawn the prefab you pass in at the position and rotation you provide, with no parent.

```
PoolBoss.Despawn(Transform transToDespawn);
```

This will despawn the prefab you pass in. There are also variants where you use a string instead of a Transform for the name of the item.

Section Three: Miscellaneous Other Scripts

1. Timed Despawned - found under Dark Tonic -> Pool Boss -> Timed Despawner. This script can be used to automatically despawn a prefab after X seconds. This is useful for despawning prefabs with particle systems without writing any code.

Section Four: Playmaker Integration

We have included an optional Playmaker package in the 3rd Party Plugin Integration folder.

- 1) Playmaker - there are several custom actions included in the Playmaker_CustomActions package. These show up under the Script Control category in Playmaker.
 - Pool Boss Despawn
 - Pool Boss Despawn All Prefabs
 - Pool Boss Despawn Prefabs Of Type
 - Pool Boss Item Despawned Count
 - Pool Boss Item Is In Pool
 - Pool Boss Item Spawned Count
 - Pool Boss Item Total Count
 - Pool Boss Prefab Count
 - Pool Boss Spawn

Section Five: Using JavaScript (UnityScript)

You will need to move the Assets/DarkTonic/PoolBoss/Scripts folder into the Assets/Plugins folder so that it will compile in the right order and be accessible from other scripting languages. It's wise to also rename the "Scripts" folder to "PoolBoss" so you know what they are after moving.

Conclusion

That's it for now! We hope you enjoy this plugin as much as we have. Now get making awesome games with this! Dark Tonic will also help you promote your games made with Pool Boss. Email us for details!

Thank you,

-All at Dark Tonic

Pool Boss is part of the complete game system [Core GameKit](#) which includes simple combat, spawners, level setup tools and much more. Check it out if you'd like the entire integrated system!

Make sure to check out our other plugins such as the top-selling Master Audio at <http://u3d.as/content/dark-tonic-inc-/master-audio/3PY>. Support is available by emailing info@darktonic.com. You can also post on the Unity Forum Pool Boss thread [linked here](#). Or use the official Dark Tonic forums here for help: <http://darktonic.freeforums.net>

Tutorial videos of our plugins can be [found here](#).