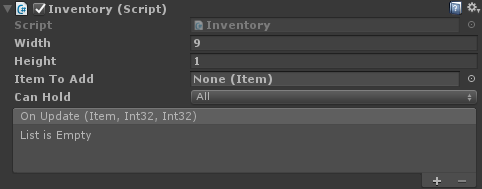
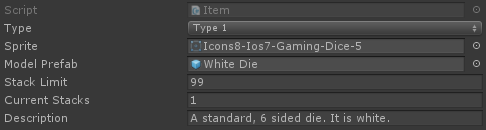
RPGSystem User Guide

Inventory System:

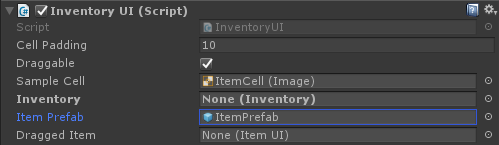
To give an object an Inventory, simply add an Inventory component to the object. Using the inspector you can set the width and height of the inventory, as well as configure which types of items the Inventory can hold:



Items can be created by right clicking in the assets folder, and from the RPGSystem context menu folder, clicking Item. This will create an Item Asset for you to fill in the info describing it:



Inventories can be displayed with the InventoryUI component. Simply add the component to an Image on your canvas:

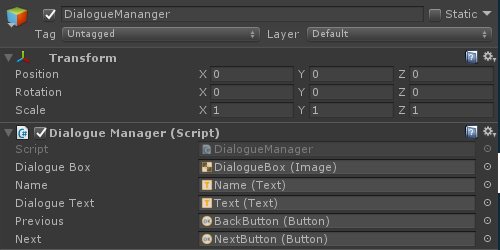


This image will serve as the background for your inventory. Create another Image to be used as the cells for the items and add it as a child to the Background Image. Drag this Image into the InventoryUI component as the Sample Cell property. Once you’ve done that, all you need is a prefab consisting of an image and a text component. The background of this image will be changed during runtime to the Item’s sprite, and the text will be changed to display the number of items in that stack

Dialogue System:

Getting started with the Dialogue System is simple.

Start by creating an Empty Object in your scene and adding a DialogueManager component



If you don’t have one already, add a canvas to your scene.

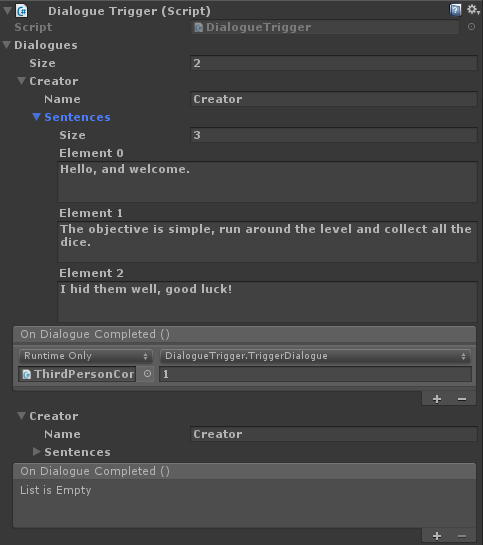
Add an Image to your Canvas, this will be your dialogue box.

Add text elements to display the name of the person talking, and what they’re saying

Add buttons to go forward and backward through the dialogue

Drag these objects into the appropriate slots in the inspector for the DialogueManager.

Now you can display Dialogue in your scene. But you have no Dialogue yet. To add some, add a DialogueTrigger to an object in your scene. Once that’s done, you can start adding Dialogue to it:



Using the On Dialogue Completed event you can trigger new dialogue automatically after one finishes. To start the initial Dialogue, call GetComponent<DialogueTrigger>().TriggerDialogue(), and pass the index of the dialogue in the list to start it. Index defaults to 0 if you don’t pass anything in.

And your done! Dialogue can now be displayed in your scene.