

Item Creation & Modifying Player Inventory

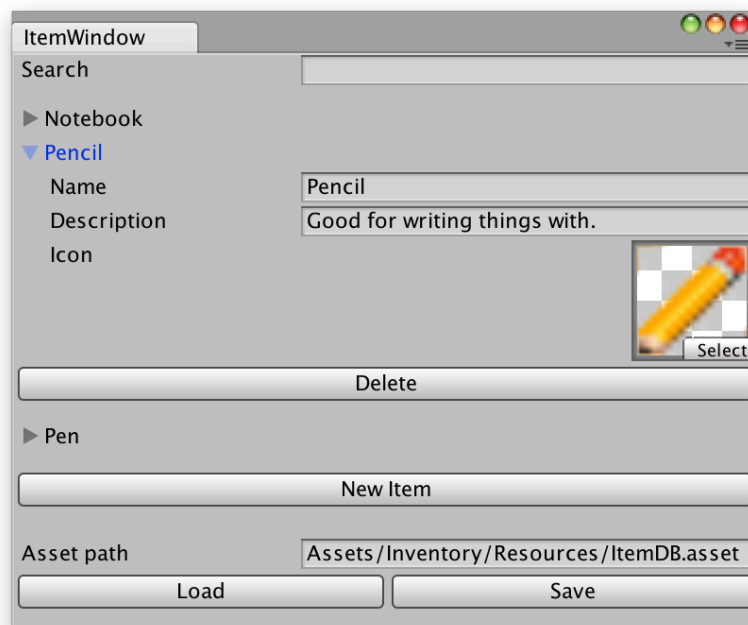
by William Chilcote

Contents

1. [Creating Items](#)
2. [Modifying Player Inventory](#)

Creating Items

At runtime, a list of all items in the game is generated. To add items to the game's item database, entries can be made to **ItemDB** assets using the custom editor. Only one ItemDB asset can be used at a time. The asset must be stored in the top level of a "**Resources**" directory, and be named "**ItemDB.asset**". The custom editor can be found under Window > Item Database. The editor includes a button to create a new item, a field to search items by name, buttons regarding the saving and loading of the **ItemDB** asset, and a list of expandable information for each existing item. The information for each item includes text fields for the item's name and description, and a sprite field for the item's thumbnail image. The description text is displayed at the bottom of the inventory UI when the item is highlighted by the player. The thumbnail image should be a 16x16 pixel PNG. The asset must be saved before exiting the Unity Editor or compiling in order for any changes to be retained.



Modifying Player Inventory

The player's inventory is essentially a sublist of the list of all items in the game, with additional information representing the amount of each item owned. The inventory can be modified easily through the use of four simple methods of type `void`. As there is only one player, these methods are static, which eliminates the need to reference the singleton instance of the `Inventory` class.

`Add` takes a string as its first parameter, referring to the name of the item that is being added to the player's inventory. This function increments the number of that item that the player has by one. There is also an overloaded `Add` method with a second parameter - an integer that represents the amount of the given item that is being added to the player's inventory.

Similar functionality is present for the `Remove` method. An important distinction however, is that the count of any item in the player's inventory will never be less than zero.

A fifth method, `Count`, takes a single parameter - a string representing the name of an item. `Count` returns an integer equal to the amount of the given item that the player has in their inventory.