Quest Machine

Invector Controller Support

Contents

Chapter 1: Introduction

This manual describes how to use Invector's character controllers with Quest Machine.

Quest Machine's Invector integration provides these features:

- Invector "has items" quest condition.
- Invector "add/remove items" quest action.
- Invector "modify stats" quest action.
- Message listener that sends Message System messages when inventories change.
- Invector reward system for procedurally-generated quests.
- Save Invector stats and inventory using the Pixel Crushers Save System.

Chapter 2: Setup

To enable Invector integration, import these two packages:

- **Plugins ► Pixel Crushers ► Common ► Third Party Support ► Invector Support**
- **Plugins ► Pixel Crushers ► Quest Machine ► Third Party Support ► Invector Support**

If you're using an Invector controller that has an inventory, such as the Shooter Template, define the scripting symbol USE_INVECTOR_INVENTORY. To do this, select Edit > Project Settings > Player and scroll down to *Other Settings*.

Since most users will be using an Invector controller that has an inventory, the example scene is designed for the Invector Shooter Template. However, this integration is designed to work with all Invector controllers.

Scene Setup

To configure your scene:

1. Add the **Quest Machine** prefab.
2. Optionally add a **Save System** component to the Quest Machine GameObject.
3. Add the **Input Device Manager** prefab.
4. For saving and loading, it is generally easiest to inspect the player controller and untick **Grounded > Use Instance**. If you must use an instance, put the quest journal on a scene object instead. Create a new, empty GameObject. Add a a **Quest Journal** component to it.
5. Add a **Quest Machine Pause Player** component to the player controller.
6. Optionally add **Invector Stats Saver** and/or **Invector Inventory Saver** components to the player controller if you want stats and inventory included in saved games. Inspect the Invector inventory prefab (e.g,. Inventory_ShooterMelee) and untick the 'Don't Destroy On Load' checkbox.
7. Add a **Position Saver** if you want to save its position.
8. Use Invector Quest Conditions and Quest Actions in your quests (see below).
9. Add an **Invector Reward System** to any quest generator entities that will offer Invector items as rewards for procedurally-generated quests.

To configure an NPC to use Invector's interaction system like in the example scene:

1. Add a **Quest Giver** component.
2. Add a trigger collider. Set the NPC's **Tag** to *Action* and **Layer** to *Triggers*.
3. Add a **v Trigger Generic Action** component. Configure the **OnPressActionInput()** event to call QuestGiver.StartDialogue, and specify the Quest Journal GameObject.
4. To show action text above the NPC's head, copy a vActionText GameObject from another interactable, reposition it above the NPC, and configure the **OnPlayerEnter()** and **OnPlayerExit()** events to show and hide it.

You may want to configure a quest journal button or hotkey as in the example scene. In the example scene, the JournalButton's OnClick() event calls QuestJournal.ToggleJournalUI.

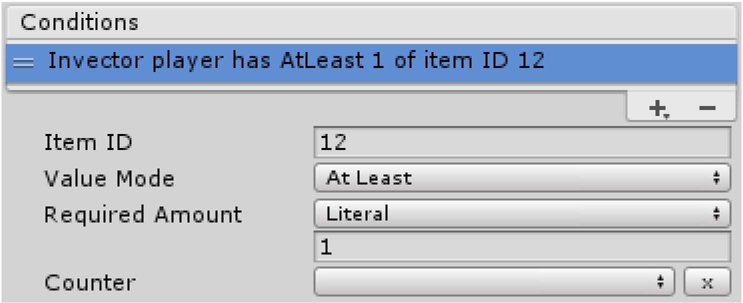
In addition, in the example scene the Quest Machine prefab's Quest Dialogue UI and Quest Journal UI are configured to send the message "Pause Player" and "Unpause Player" when they're opened and closed. This tells the Quest Machine Pause Player component to pause the player and free the mouse cursor. To do this, the UIs have a Message Events component that the OnOpen() and OnClose() events use.

Your own scripts can also call the Quest Machine Pause Player's PausePlayer() and UnpausePlayer() methods any time you want to pause and unpause the player controller.

Reminder: Define the symbol "USE_INVECTOR_INVENTORY" as described above if you plan to use inventory quest conditions or actions.

Chapter 3: Quest Conditions

Invector Has Items Quest Condition


Conditions

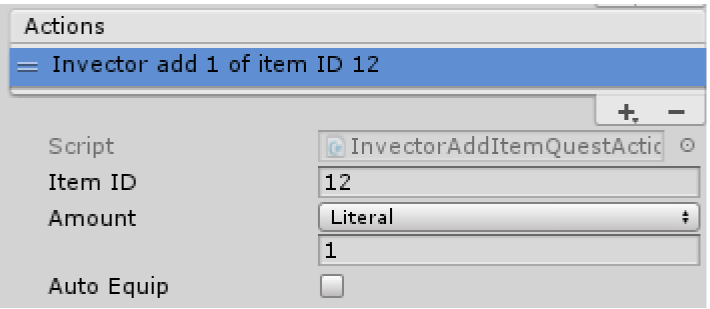
Invector player has AtLeast 1 of item ID 12

Item ID: 12
Value Mode: At Least
Required Amount: Literal
1
Counter:

The **Invector Has Items Quest Condition** checks if the player has a required amount of an item. It can also keep a quest counter updated with the current item count.

Chapter 4: Quest Actions

Invector Add Item Quest Action



Actions

Invector add 1 of item ID 12

Script: InvectorAddItemQuestActic

Item ID: 12

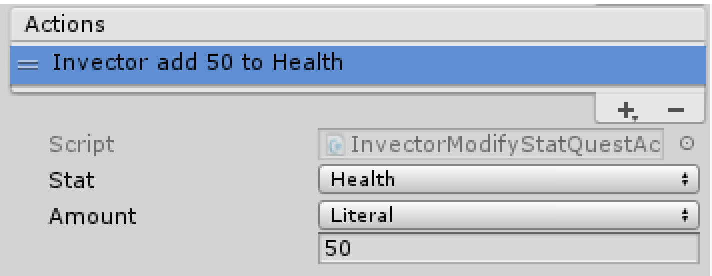
Amount: Literal

1

Auto Equip: □

The **Invector Add Item Quest Action** adds or remove an amount of an item. To remove items, specify a negative amount.

Invector Modify Stats Quest Action



Actions

Invector add 50 to Health

Script: InvectorModifyStatQuestAc

Stat: Health

Amount: Literal

50

The Invector Modify Stats Quest Action modifies a stat value (Health or Max Health).

Chapter 5: Quest Generation

To set up a quest generator NPC to use Invector:

1. Configure the quest generator entity as normal in Quest Machine:
   - Add a Quest Giver component.
   - Add a Quest Generator Entity component.
2. Add an **Invector Reward System** component. Then configure the items that it can offer and the reward system point value for each item.

Final Advice

Pick apart the demo scene to see how it works. If you have any questions, we're here to help! Please visit the forum or email support@pixelcrushers.com any time!