

Justin Brauer
QuestController Dev Tool

Link to GitHub Repository: <https://github.com/Dumbass247/QuestController>

Last Verified Unity Version: Unity 2021.3.17f1 Personal <DX11>

Installation:

- Make an empty object in your scene, and attach the QuestManager script to it. Drag the ScriptableObject **QuestJournal** into the Quest Journal field
- Apply the NPCQuest script to any objects in your scene that would give a quest to the player. The last field in the inspector will ask for you to add that object with the QuestManager script to it.
- Make sure you have other objects in the scene that will function as your quest turn-in locations, pick up locations and enemy objects for a kill quest. In this tool, I have a NPCQuestTurnInKill, NPCQuestTurnInFetch, NPCQuestTurnInDelivery, NPCQuestFetchItemGiver, and a simple rat object (for the kill quest target).
 - All of these objects in my scene are placeholders for real game objects
- I would place the editor scripts in a folder labeled Editor for good organization, the location of other files don't specifically need to be placed anywhere, within reason.

Tutorial:

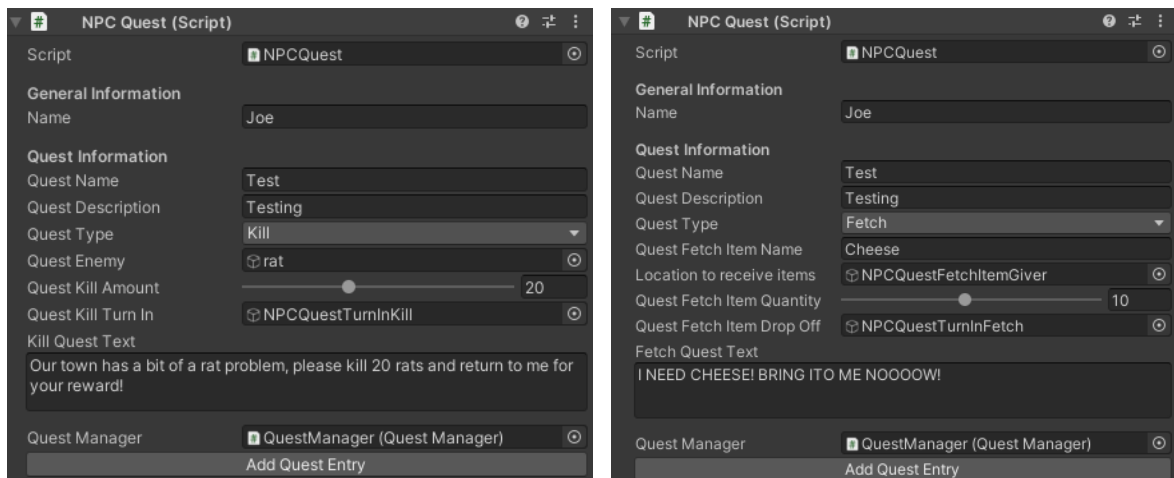
NPCQuest script

- Once that initial set up is done you are ready to use the tool, below are some image guided explanations on how to use it.

Here is my example object that I attached the NPCQuest script to



Inside of that object you can open up the NPCQuest script, you will see some basic fields and a drop down that lets you select between 3 types of quests, Kill, Fetch and Delivery.



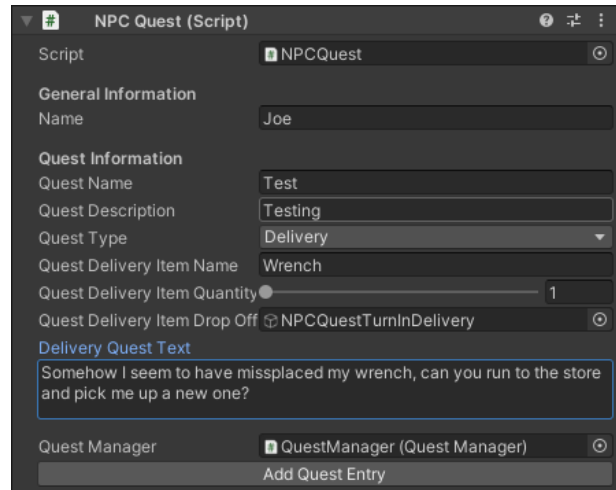
Justin Brauer

QuestController Dev Tool

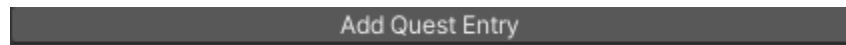
You will see that you can input quite a few different values. Once the quest type is selected you can fill out the corresponding fields.

There are sliders for the quantity of items/kills for the related quest type, and you can drop the game objects into the pick up and drop off fields of which you want to turn in the quests or pick up items.

Lastly you can input any text you wish the NPC to share about the quest in the Quest Text field at the bottom.



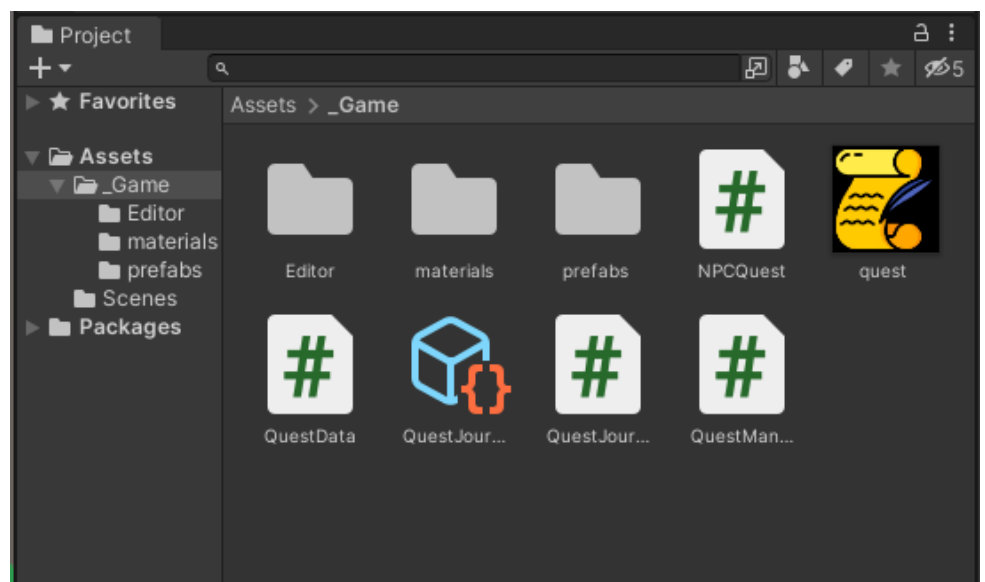
At the bottom of the Inspector for NPCQuest you will see the Add Quest Entry button,



this will move the stored values over into the QuestJournal scriptable object in the assets folder. This button is a placeholder for accepting the quest via interacting with the NPC in a game.

- **QuestJournal ScriptableObject**

Now that those steps are completed and you have migrated the quests to the quest journal you will be able to see them placed there accordingly.



If you click on the QuestJournal ScriptableObject it will open up and show you the quests that you have added

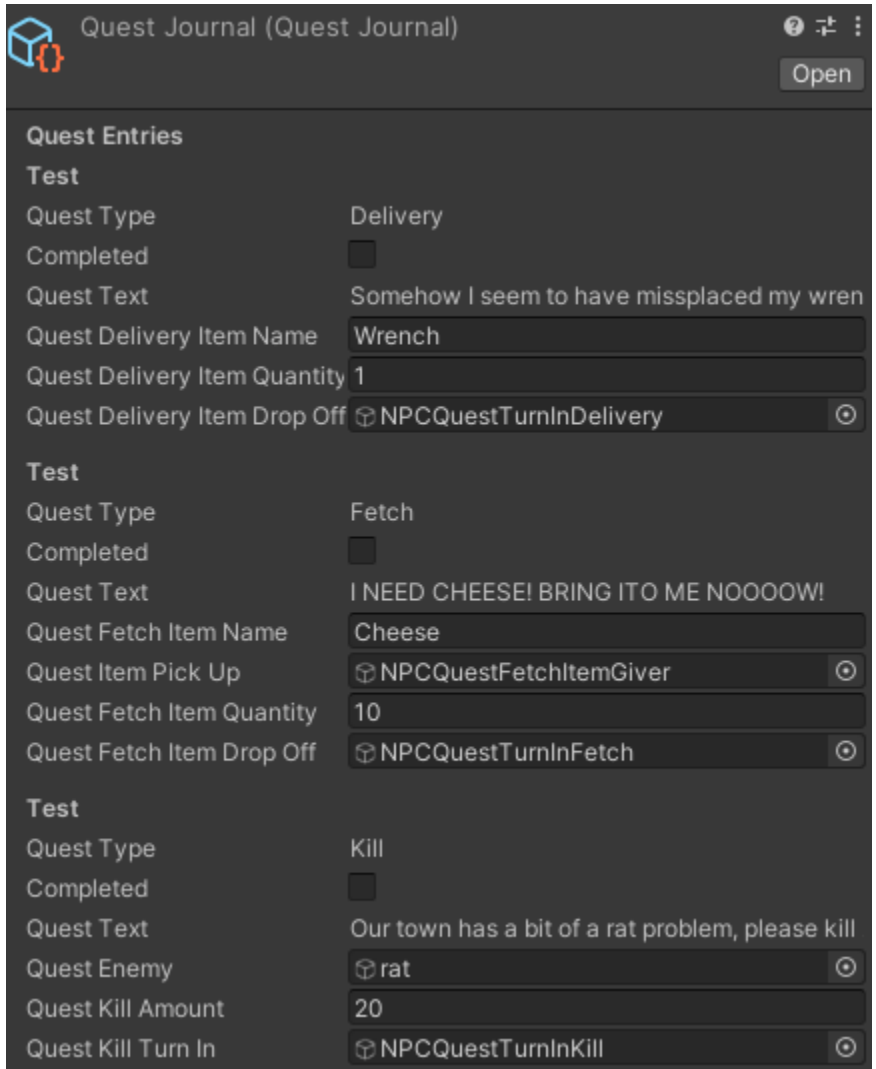
Justin Brauer

QuestController Dev Tool

Here is an example of an empty QuestJournal



And an example QuestJournal with the 3 example quests from early in this tutorial



You will notice that only the corresponding information populates depending on the type of quest. And all of the fields are generated with the correct values.

There is a button that appears under the quest type asking if the quest is completed. This acts as a placeholder again for an interaction system between the player and NPC or Item objects in your scene.

This is left open ended like this to be easily adaptable to whatever inventory system, or interaction system the game developer wishes to implement.

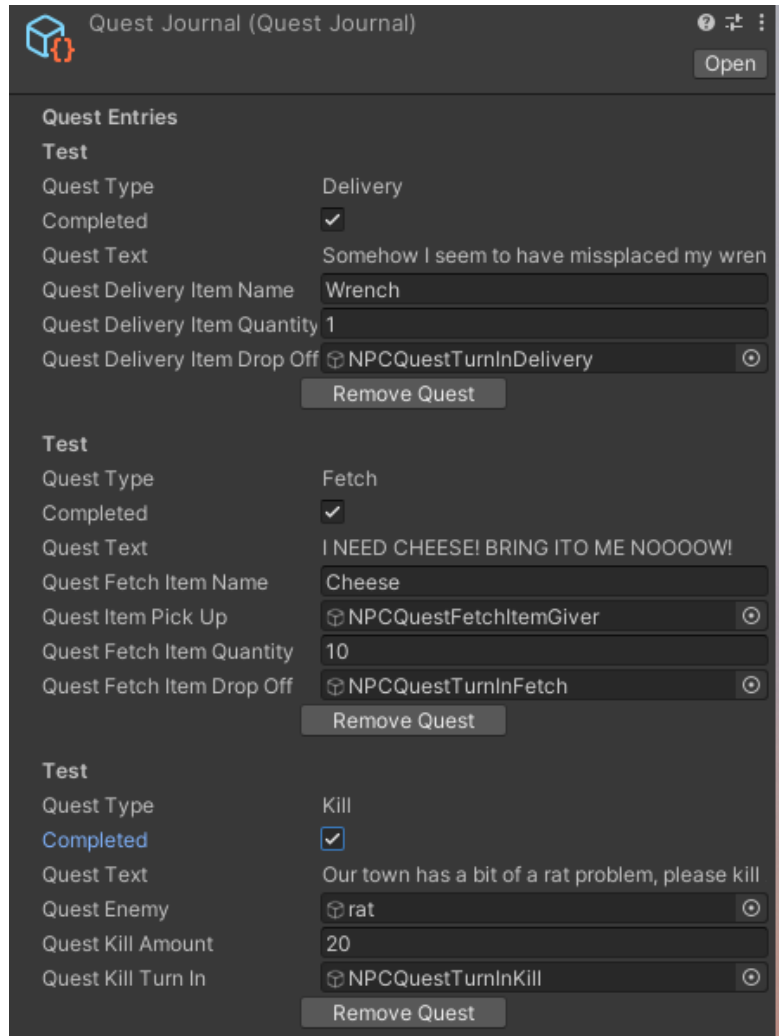
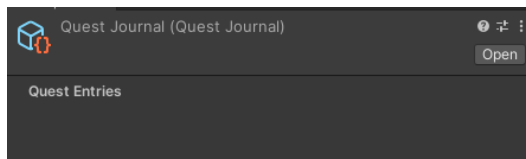
Justin Brauer

QuestController Dev Tool

Once you toggle the quest as complete, you will notice a button appear asking if you'd like to remove the quest from QuestJournal. This is to help declutter the QuestJournal while you develop your game.

Depending on the type of game being developed you could easily adapt this to migrate the information to a completed quests log. It just depends on your plan for your game.

Once clicked the QuestJournal will remove the quest from its log



And that's it for the tutorial! I am super excited to use this myself or see what others are able to do with this tool.