

Important Information

Support and Full Documentation

- Full documentation and examples can be found at [our website](#)
- Support for this asset can be found at [our Github](#)
- Contact us at support@quantumtekhub.com for problems with the asset

Installation

1. Package Requirements

Before installation of Quantum Quests, you need to have the Text Mesh Pro package from the Package Manager (Under Window > Package Manager) installed. After installation of Text Mesh Pro, you need to import the files necessary for Text Mesh Pro from Window > Text Mesh Pro > Import TMP Essential Resources.

However, you don't need Text Mesh Pro if you don't import the demo folder.

2. Install

After you meet the package requirements you can install Quantum Quests.

3. Demo Scenes

There is one demo scene: the Quest Demo, located under the Demo folder. The Quest Demo showcases the quest system, with completing a quest comprised of multiple tasks. The demo consists of one quest, with a main task, and two tasks that can be used to complete the main task of getting tomatoes.

Getting Started

Basics

Quantum Quests is a quest system that can have quests with many tasks in it, with a lot of customization. It uses a node editor to make it easier to create a quest tree.

Quantum Quests is made of quests. Quests can be long and complicated, or short and simple, depending on how many tasks are in it. Quests have a name, the name of the NPC that gave the quest, and a description. They also have a status (NotGiven default, Inactive, Active, Completed, Failed), a list of tasks, and a list of the first tasks in the quest. Each task has a name, description, max progress (such as 3 for 3 apples), list of the IDs of the next tasks, and whether the task is optional or not for completing the quest.

To create a quest, go to Create > Quantum Tek > Quantum Quests > Quest. To edit a quest, first Create > Quantum Tek > Quantum Quests > Node Database. This file should be saved under an Editor folder, as it is only needed in the Unity editor. Drag the quest (scriptable object) into the Data DB field of the node database. Select the node database, then go to Window > Quantum Quests > Quest Editor.

Quests have tasks that lead to other tasks. A quest is complete when all of its tasks are complete. Quests are store in a QQ_QuestSO object, and can be put in the list of quests in a QQ_QuestDB object.

Quest Node

Bob Wants Tomatoes	
Name	Bob Wants Tomatoes
NPC Name	Bob
Description	Bob is really hungry for some

Quests need a name (“Bob Wants Tomatoes”), but can also have the name of the quest giver/NPC (“Bob”), and can have a description (“Bob is really hungry for some tomatoes.”). It can connect to multiple tasks, which could be represented as the first tasks. This is a connection between the first output knob (top right) of the quest node, and the first input knob (top left) of the task node.

Task Node

Get Tomatoes for Bob	
Name	Get Tomatoes for Bob
Description	You need to get Bob some to
Max Progress	3
Optional	<input type="checkbox"/>

Tasks need a name (“Get Tomatoes for Bob”), max progress (3), and whether or not they are optional (false). They can also have a description (“You need to get Bob some tomatoes.”). The first input knob (top left) on a task can connect to the first output knob (top right) of a quest or a task.

Using the Editor

Drag nodes by clicking and dragging the mouse. Pan the editor using the bottom and right scrollbars. Right click outside of a node to add a new node. Right click on a node to detach all

connected nodes, or delete the node. Right click on a knob (the blue squares on either side of the node) to attach a node, or detach nodes connected to that knob. When attaching nodes together, the first selected node will have a connector between the node's knob and the mouse. Left clicking on a knob on another node will connect the two, as long as it is an allowed connection, as highlighted in the paragraph below.

Connection Rules

Quest - Inputs: None. Outputs: First - First on Task.

Task - Inputs: First - First on Quest, First on Task. Outputs: First - First on Task.

Quest handlers need a quest DB to assign quests. Quests can't be used unless they are assigned by name from the handler's QQ_QuestDB object. This can be done with the AssignQuest(string name) with the name of the quest. Once a quest is assigned (such as when an NPC gives the quest), its status becomes Inactive. It should be activated whenever the character chooses to follow the quest. A quest's status can be changed with the CompleteQuest(string name), ActivateQuest(string name), DeactivateQuest(string name), FailQuest(string name) methods. CompleteQuest also sets the Completed variable to true.

Calling the GetQuest(string name) function will return the quest with the given name. A quest is made of many tasks, and each task has progress. This can be increased with the ProgressTask(string questName, string taskName, float amount) function. The CompleteTask and CompleteQuest functions are automatically called by the system if needed when using the ProgressTask function, but they can be manually called.

Script Terminology

Quantum Quests contains custom scripts, with each outlined in the table below.

QQ_QuestHandler - Keeps track of a character's current quests.

QQ_Quest - Represents a quest given by an NPC in the world.

QQ_Task - Represents a part of a quest, such as collect 3 apples.

QQ_QuestSO - A quest in a scriptable object form.

QQ_QuestDB - A list of quests.