

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 Attributes, 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 Attributes.

3. Demo Scenes

There are two demo scenes: the Attribute Demo and the Experience Demo, located under the Demo folder. The Attribute Demo showcases the attribute system, and the Experience Demo shows the experience system.

Getting Started

Basics

Quantum Attributes is comprised of two systems: the Attribute system and the Experience system. The Attribute system handles different character attributes and stats commonly found in an RPG. The Experience system handles character experience levels.

Each character that needs stats should have a `QA_AttributeHandler`, which takes care of that character's attributes. The handler takes in a `QA_AttributeDatabase`, which should be shared amongst characters with the same attributes. Each attribute has a name, minimum value, maximum value, and modifiers.

Modifiers change the value of an attribute, either when added/removed, or over intervals of time. An example could be a character receiving a temporary speed buff that adds speed when applied, but removes the added speed when the modifier ends. The `ModificationInterval` property of a modifier determines how many seconds or frames (based on `TemporarySeconds` or `TemporaryFrames` duration) before the modifier is applied again. Having 0 as the interval means that the temporary modifier will act as a permanent modifier but be removed after a certain period.

When adding an attribute to the handler, call the `AddAttribute` function, inputting the name of the attribute (from the handler's attribute database), along with a starting value. Removing an attribute is as easy as calling `RemoveAttribute` with the name of the attribute.

Modifiers can also be added and removed, using the AddModifier and RemoveModifier functions, taking in the attribute name and modifier name (from the handler's modifier database).

Each character that needs experience, such as after completing a quest, should have a QA_ExperienceHandler. The handler has a level, max level, xp, max xp, max xp increase, xp type and max xp type. Every time the character's xp gets to max, they level up (if available) and their max xp increases. If their xp type is carry, they won't lose excess xp when leveling up. If the type is reset, they lose excess xp. Additionally when leveling up, the max xp can increase in two different ways. It can increase by a percent, or increase by a certain number.

Script Terminology

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

QA_AttributeHandler - Handles any attributes a character has, including decaying and removing them.

QA_Attribute - Represents a character attribute, such as strength, agility, etc. It can also be used to represent health, mana, etc.

QA_AttributeModifier - Represents a modification to a character attribute, such as a permanent strength buff, a temporary poison that decreases health, etc.

QA_AttributeDatabase - Stores a list of attribute templates for characters to use.

QA_AttributeModifierDatabase - Stores a list of attribute modifier templates for characters to use.

QA_ExperienceHandler - Handles character level and experience.