

# 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](mailto:support@quantumtekhub.com) for problems with the asset

## Installation

### 1. Package Requirements

Before installation of Quantum UI, 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.

### 2. Install

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

### 3. Demo Scenes

There are 4 menu scenes: Main Menu, Pause Menu, most of the UI Elements, and UI Windows. These can all be found in the Scenes folder.

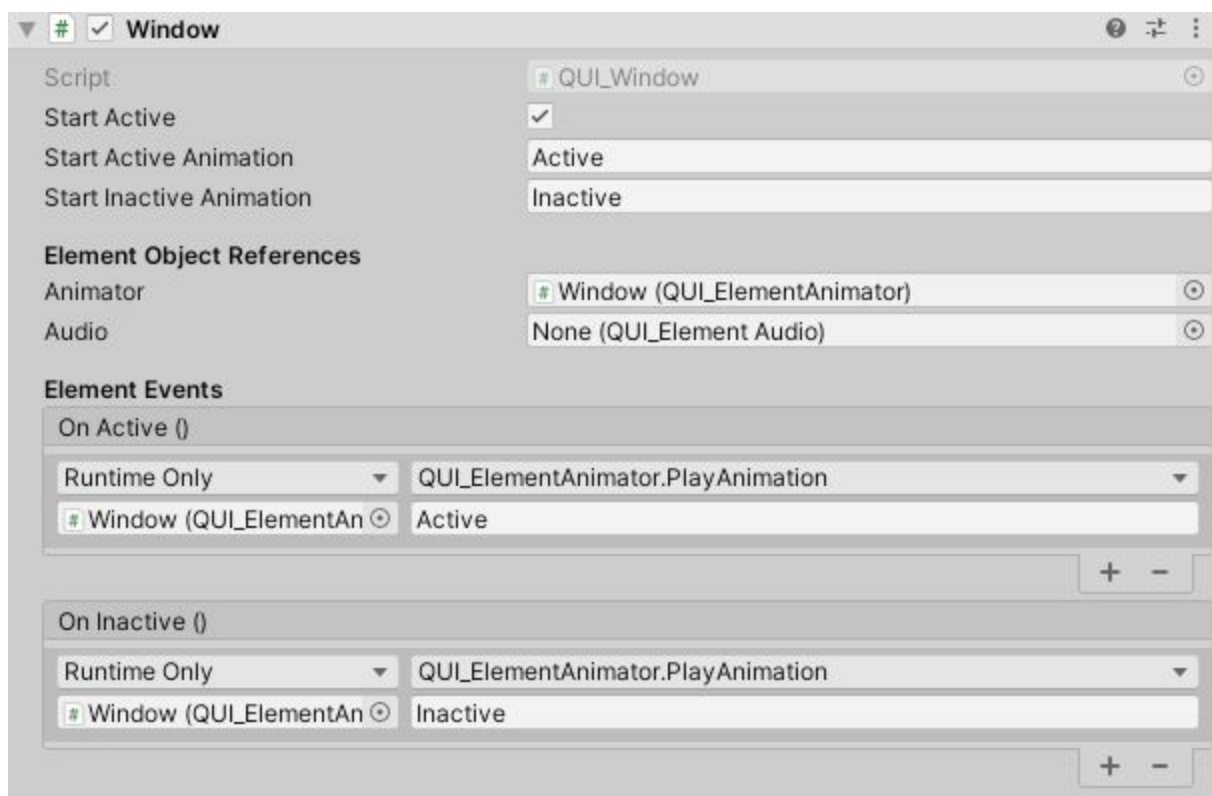
## Getting Started

### Basics

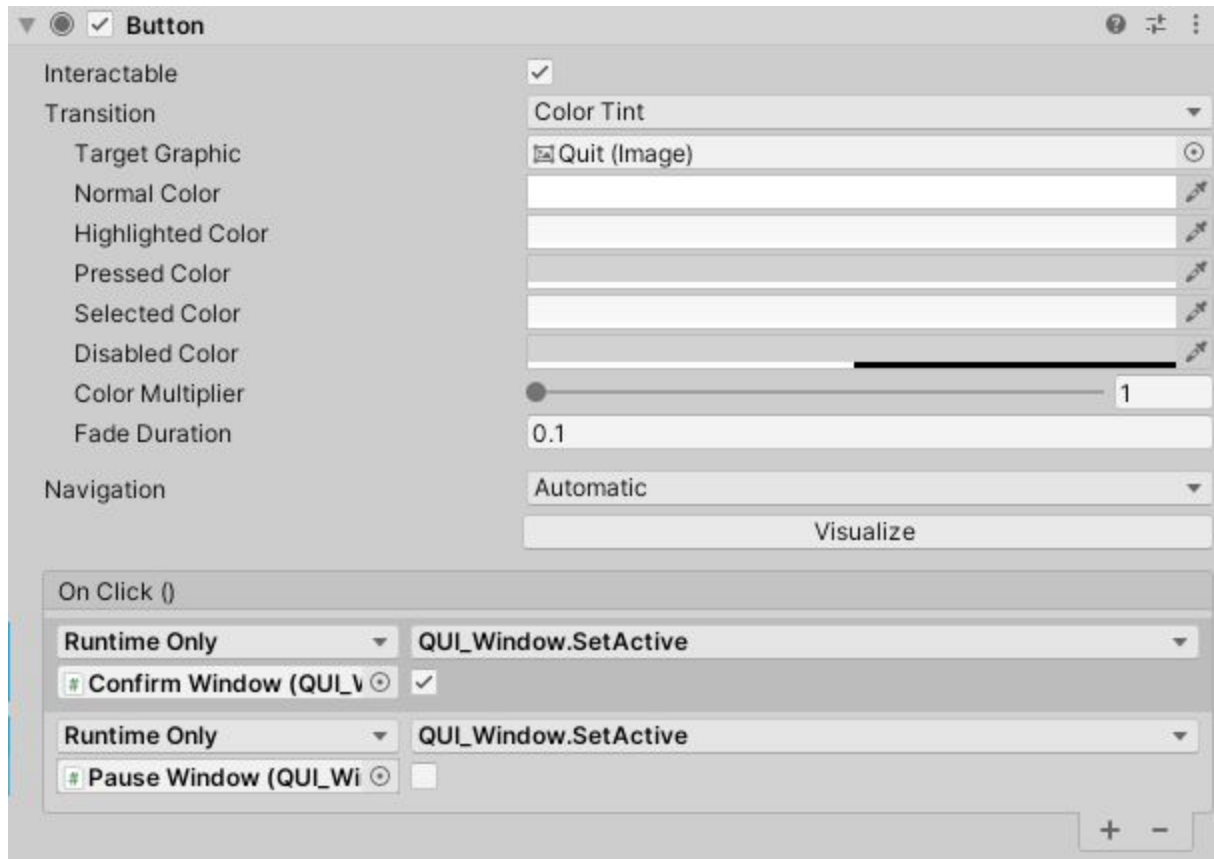
Quantum UI is run off the basis that UI is contained within a menu, which is just a canvas element with the QUI\_Menu

component. In the menu there are windows, tab groups, and UI elements. Windows are just like any normal window in a UI menu. Any UI elements should be placed under the content GameObject. Tab groups are made of tab windows, which can only have one open at a time. Clicking the different tabs will change the current tab window in the tab group. UI elements are everything else, such as text, sliders, etc.

Most QUI components are based off of QUI\_Element, which can start as active or inactive. This can be on or off, open or close, or anything like that. Elements can trigger UnityEvents to occur when the element is activated/deactivated.



Elements can be activated or deactivated by calling their SetActive function. This can be used to open or close windows on button click.

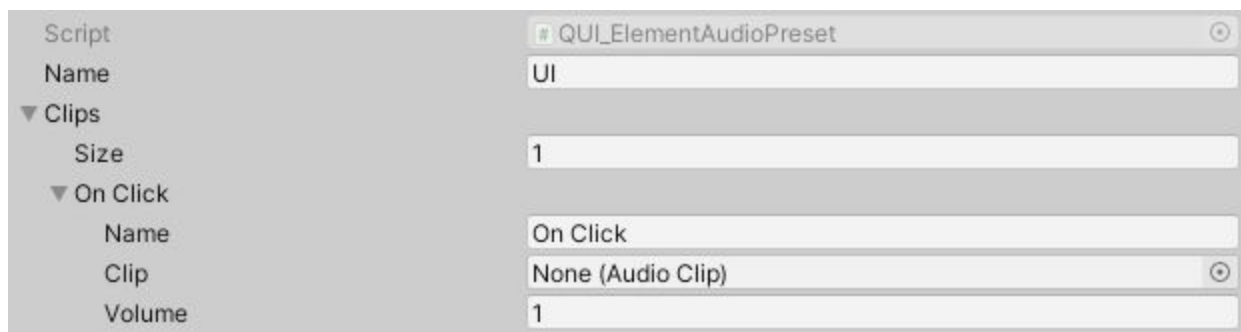


This is especially useful in conjunction with the `QUI_ElementAnimator` or `QUI_ElementAudio` to animate an element (such as a window opening), or trigger audio (such as on button click). The `QUI_ElementAnimator` is used to immediately play the end of an animation on start, depending on if it starts active or not.

Element animators do not need an element to function, but they do need an animator and an animation preset. Animation presets just say what animation to play, or what parameter to set (along with the value for that parameter). The example preset for Window Active State is a good example of this, as it uses a boolean and a value to set, along with two animations for active and inactive.



Element audio is a component built for UI SFX. It requires an audio source and audio preset. Audio presets have a list of audio clips that take an audio file, name, and volume.



There are many other UI elements discussed in the Prefab Terminology section below.

## Prefab Terminology

Quantum UI contains many helpful prefabs for speeding up UI development, outlined in the list below.

**Menu** - Just a canvas with the `QUI_Menu` script attached.

**Window** - The basic window for holding UI elements. The Window Content GameObject contains all UI in the window. To open or close the window, call its SetActive function.

**Tab Group** - Similar to a window, but rather a group of tab windows. Only one tab window can be open at a time. The Tab Group Content GameObject has all the tab windows in the tab group. To open or close the tab group, call its SetActive function. To change the tab window, call the ChangeWindow function.

**Tab Window** - Similar to a window, but with a tab as well as tab window content for UI elements.

**Tab** - A button showing if a tab window is open or not, based on the current tab window in the tab group.

**Background** - A simple tiled image covering the whole screen.

**Scene Transition** - Made to handle loading different scenes either instantly, asynchronously with UI showing, or asynchronously with a loading scene showing. Also controls scene transition animations, with the default being fading the screen.

**Text** - Just a TextMeshPro UGUI component, to allow for changing most similar text across all UI with one prefab.

**Animated Text** - Similar to the Text prefab, but with a QUI\_ElementAnimator.

**Button** - A basic button with a text label.

**Icon Button** - A button using only an icon and no text.

**Icon Button With Background** - An icon button with a sprite background below the icon.

**Toggle** - A basic toggle with a text label.

**Toggle Group** - A group of toggles.

**Switch Toggle** - Similar to toggle, but changes the image sprite and background rather than turning it off or on.

**Switch Toggle With Label** - Switch toggle, but with a text label to the right.

**Icon Toggle** - Similar to a switch toggle, but doesn't look like a switch.

**Icon Toggle With Background** - Icon toggle, but with a background.

**Input Field** - The basic TextMeshPro input field.

**Dropdown** - The basic TextMeshPro dropdown.

**Option List** - Similar to a dropdown, but horizontal and using arrows to change the option selected.

**Bar** - Used for progress bars, health bars, and so on. Can fill with width, height, or Image fillAmount.

**Radial Bar** - A radial version of the bar.

**Slider** - A default slider.

**Slider With Label** - A slider with a text label above.

**Scroll View** - A simple scroll view.

**Loading UI** - A panel with loading text and a progress bar.

**Confirm Window** - A window with confirmation text and yes and no buttons.

**Loading Window** - A window version of the Loading UI prefab.

**Login Window** - A window with a username and password input field, along with a button.

**Pause Window** - A window with resume, settings, and exit buttons.

**Settings Tab Group** - A simple tab group with basic settings and gameplay, audio, and visual tabs.

**Main Menu** - A basic main menu with a play button and settings window.

**Pause Menu** - A Pause Window prefab combined with the Settings Tab Group prefab.

## Script Terminology

Quantum UI contains many custom scripts, with each outlined in the list below.

**QUI\_Menu** - Responsible for managing all of the UI elements in a menu.

**QUI\_Element** - The base for most UI elements in the Quantum UI system. It contains all the basic fields and methods used by each element.

**QUI\_Window** - A basic UI element responsible for holding other UI elements.

**QUI\_TabGroup** - A basic UI element responsible for holding QUI\_TabWindows.

**QUI\_TabWindow** - A basic UI element responsible for holding other UI elements in a tab group.

**QUI\_Tab** - Used for changing which tab window is shown in a tab group.

**QUI\_ElementAnimator** - Used to handle animation of a UI element.

**QUI\_ElementAnimation** - Used for the animation of a UI element.

**QUI\_ElementAnimationPreset** - Gives a set of animations for a UI element.

**QUI\_ElementAudio** - Used to make UI sounds on an element.

**QUI\_ElementAudioClip** - Stores info about an audio clip to be played an element audio source.

**QUI\_ElementAudioPreset** - Gives a set of audio clips for a UI element.

**QUI\_SceneTransition** - Handles smooth transition effects between scenes, along with loading screen effects.

**QUI\_SwitchToggle** - Handles the change of sprites for a toggle UI element on click.

**QUI\_OptionList** - Offers multiple strings to choose from using buttons.

**QUI\_Bar** - Represents any type of UI bar, such as a health or progress bar.