Quantum Tek

Menu Template

Overview

Support for this asset can be found at https://github.com/QuantumTekSupport/MenuTemplate/issues, or you can contact us at quantumteks@gmail.com. This asset allows users to easily create menus, either made of multiple windows with however many open at a time, or tab menus with only one window shown, and tabs at the top. There are many prefabs for menus and UI elements. Audio upon button click is supported as well. Settings can be automatically saved using the system setup for it. Lastly, animations are supported for opening/closing windows, along with activating/deactivating tabs.

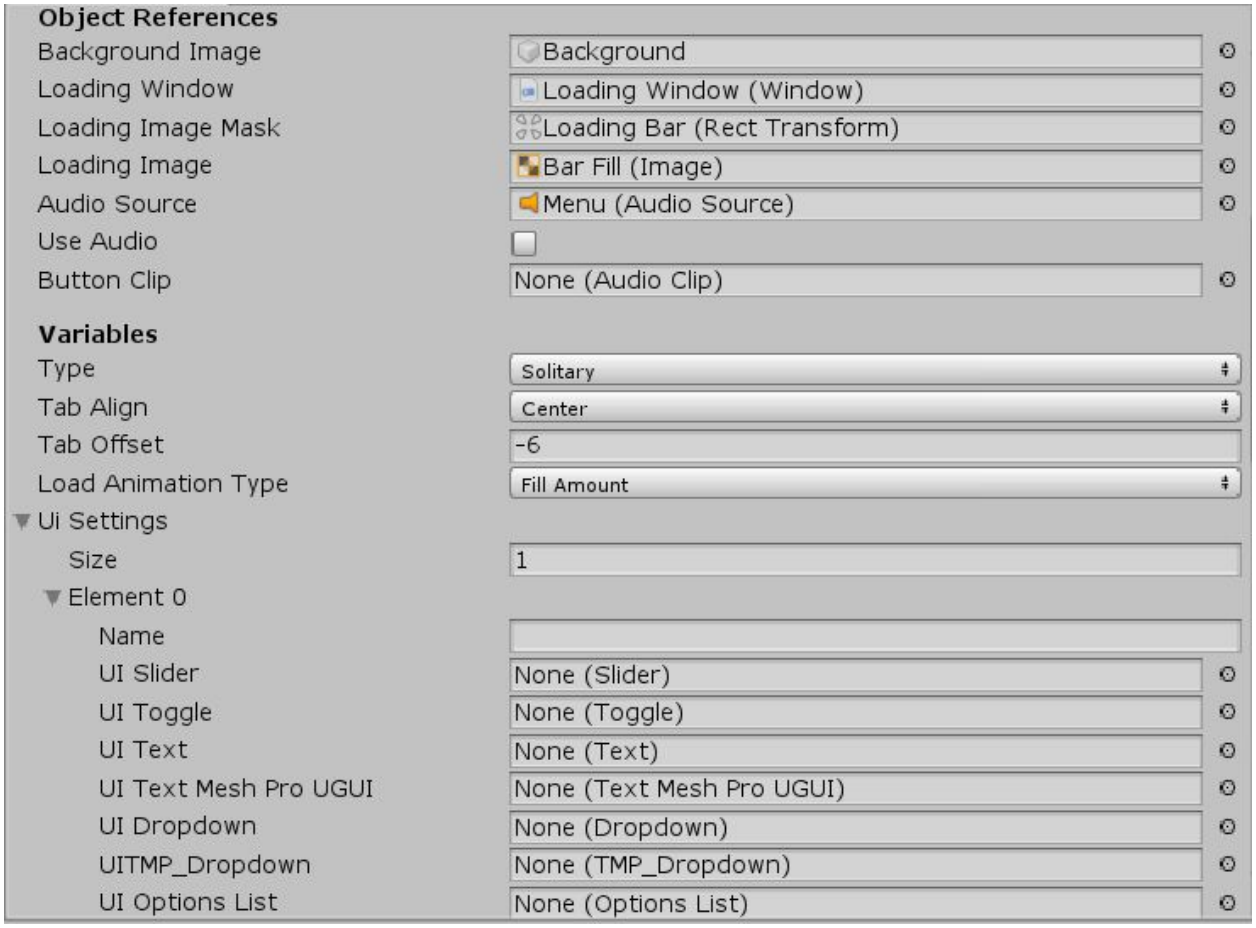
IF SOMETHING isn't working, it might be that TextMesh Pro isn't an installed package in your Unity project. To install it, open the Package Manager (Window/Package Manager), click "TextMesh Pro", and click "Install". When the installation is finished, click "Import TMP Essential Resources" under Window/TextMeshPro. The problem should now be fixed. If not, check that the text components have the TextMeshProUGUI component. Finally you can contact me to help fix the problem.

Table of Contents

Manual

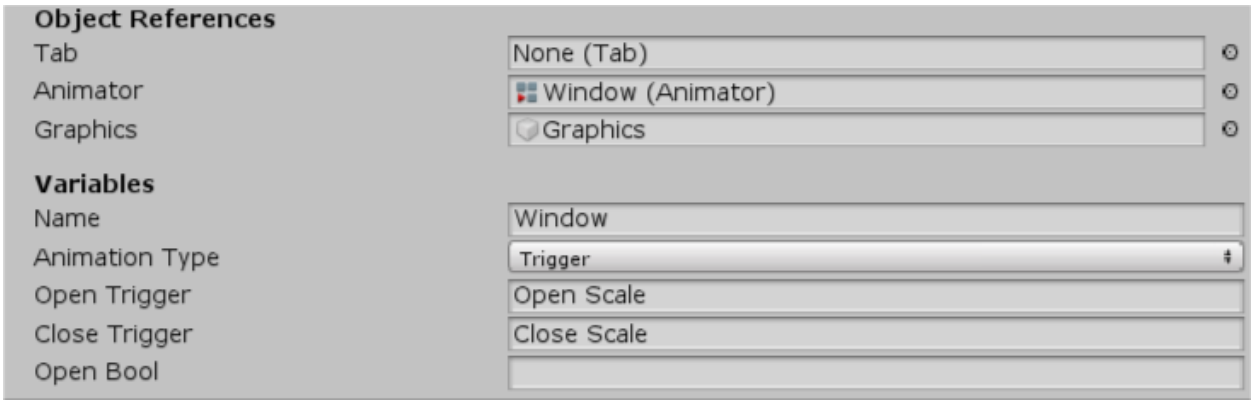
Menu Template

The menu template is made of menus. Menus are made of windows. Windows can have a tab, if they are in a tab menu. If the menu doesn't have tabs, multiple windows can be open at once. However, a tab menu can have normal windows open along with a tab window. This is used for a loading screen and a confirmation window, but can be used for others. Opening/closing a window means that a window is shown/hidden, but a certain animation may trigger, if it is a normal window. If it is a tab window, the window will just disappear. The tab of the current window in a tab menu is selected, or "active", all of the others are "inactive". When tabs change active states, they too can have animations. These can be a Unity animation, color switch, or sprite switch. These behaviours are all dependent upon the variables. This asset uses Unity's new Prefab workflow, so updating the Window prefab, for example, will change windows across all of the menu prefabs. There are four custom scripts, Menu, Window, Tab, and OptionsList. The first three were explained above, but the OptionsList allows for a list of options to pick from, using side buttons to scroll between the options. This is customizable, as it just uses a function to scroll through, and that can be assigned to any of Unity's standard UI elements, such as a Button's on click. Below are pictures of the inspectors for these four scripts, and their explanations.

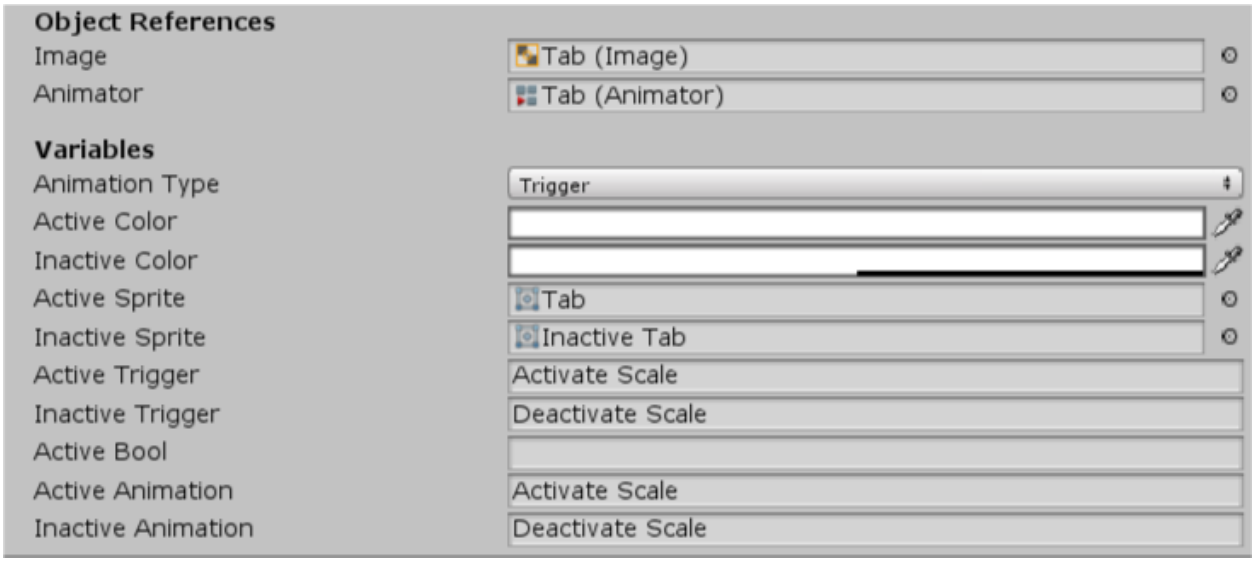


This is the inspector for the Menu script. There are a lot of fields in this. The first is a reference to the background image in the menu. The next three are need for the loading window. One is a reference to the Window itself, and the other two correspond to the loading bar on the window. There is a mask object and the fill bar showing how much of the next scene is loaded. After those three, the next three deal with audio. There is an assigned audio source by default, an option toggle to use audio or not, and an AudioClip to play when a button is pressed. This can be triggered by adding the Menu.PlayAudio() function to a button, or if the button triggers opening/closing a window, loading a scene, or is a part of an OptionsList, it will automatically trigger audio playing. Audio will not play even if these are assigned, if "Use Audio" is turned off. Last but not least are the variables for the menu. These include the type of window (Solitary or Tab, solitary meaning multiple open windows), tab alignment, tab offset on the y axis, the load animation type, and the UI settings. The tabs are automatically aligned at the start of the app, or when clicking the play button in the editor. The tab offset is for adjusting the y position of the tabs on top of the window. The offset is -6 by default to work with the default tab and window graphics. Load animation type is

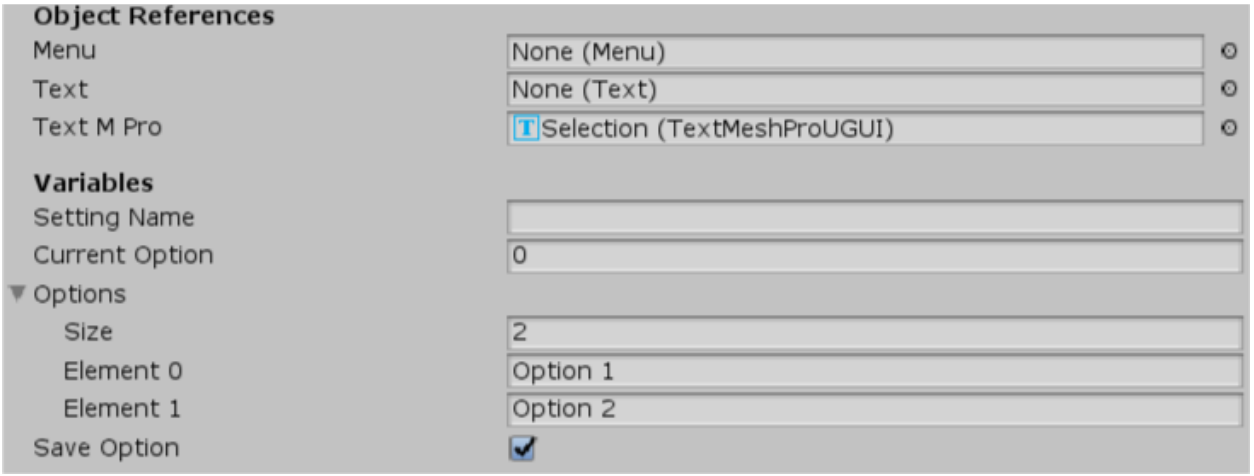
used when playing the loading animation on the loading window. Fill amount won't stretch or shrink the image, because it will use the Fill Amount field on the Image graphic. Width will change the width of the Image graphic, so the image will stretch/shrink. This is useful for sliced and tiled image types. UI settings are for automatically saving and loading settings in a settings/options window. Each UISetting has a name, and a field for each supported type. If the setting controls volume, for example, the slider can be selected in the UI Slider field. The settings are automatically loaded into the UI elements upon play. Settings are saved upon change via the "On Value Changed" property of standard UI elements, or automatically on the OptionsList, covered later.



This is the inspector for the Window script. There is a Tab reference, if the window is a Tab window, but otherwise the field can be empty. The animator field is assigned by default, but not needed unless the animation type is Trigger or Boolean. The Graphics GameObject field is required and assigned by default. It is the Window object's child GameObject that stores all of the graphical components of the window. The name of the window is only used to find the window by name when opening/closing it by the Menu script. The animation type is either None, Trigger, or Boolean. If None is selected, the window will open/close immediately. Trigger uses trigger animator parameters to trigger an animation when opening/closing the window. Boolean does the same, but with a boolean to control the open state. The last three fields relate to these animation types, and are the names of the animator parameters used by the animator.



This is the inspector for the Tab script. Image and Animator are references to those components attached to the Tab GameObject. There doesn't need to be an animator component, but Image is required and assigned by default. Animation type is either None, Color, Sprite, Trigger, or Boolean. If none is selected, nothing will happen when the active state changes. Color will change the color from active to inactive or opposite when the state changes. Sprite behaves similarly, but with a Sprite instead of a color. Trigger and Boolean work the same as with window animations. The rest of the fields have to do with the animation. The colors to switch between, the sprites to switch between, and the animator parameters are all assigned by default. All that needs to be done to see the different ones is switch Trigger to a different type, and click the Unity play button. The final two fields are the names of the animation states when the tab is active or inactive. By default these are "Activate Scale", and "Deactivate Scale", as they are called in the default Tab Animator Controller.



This is the inspector for the OptionsList script. The first field is a reference to the menu the list is in. This is required, and needs to be assigned whenever using a list. The OptionsList isn't automatically in a menu, so this explains the empty field by default. Text and Text M Pro are to show the selected options. This menu template doesn't use Unity's Text element, but supports it in all of the features. Setting name is the name of the UISetting for auto saving in the menu. Auto saving works across menus, so if the setting is the same name and there are two components in different menus, then as long as the components are assigned under the menu's UISettings field under the same name, then the list will appear the same across the two menus. Current option is the index in the list of options that is selected at the start of the program. This will change upon play if there is a saved setting to do so. Save Option is a toggle to save by default when the option is changed. This can be done by manually by code or automatically by checking the toggle on.

UI Prefabs

There are two types of UI Prefabs in this asset: menu prefabs and UI element prefabs. The menu prefabs include a large main menu with the buttons on the bottom and the focus on the title and background, a typical main menu with a level select, options menu with gameplay, video, and audio, and a how to button. There are also two pause menu prefabs, one is a normal menu, and the other uses tab windows. Additional prefabs for a basic menu, window, tab menu, and tab window, and tab are also available. Lastly there are multiple UI element prefabs. There is one for a button, a button with an icon, a square button, a slider, a toggle, a window header (using TextMesh Pro), an OptionsList, a grid of buttons, and two windows. These two windows are the loading window prefab and the confirmation window prefab. Every menu comes with a loading window by default to use when loading a scene. This is required for loading a scene by use of the Menu script API for it. Demo scenes for the menu prefabs are available under the asset's directory in the Examples/Scenes folder.

Scripting API

All of the following classes use public variables only if needed, otherwise they are protected to allow for polymorphism (scripts based off of them).

Menu

protected List<Window> windows - The list of windows in this menu.
protected List<Tab> tabs - The list of tabs in this menu.
protected List<Window> activeWindows - The list of active/open windows in this menu.
protected Tab activeTab - The currently active tab in this menu.
protected GameObject backgroundImage - The background image in this menu.
protected Window loadingWindow - The loading window in this menu.
protected RectTransform loadingImageMask - The RectTransform loading image's mask in this menu.
protected Image loadingImage - The loading image in this menu.
protected AudioSource audioSource - The audio source in this menu.
protected bool useAudio - Whether or not to use audio upon button action in this menu.
protected AudioClip buttonClip - The AudioClip played when a button in this menu is clicked.
protected MenuType type - The type of this menu.
protected TabAlign tabAlign - The tab alignment of this menu, if there are tabs.
protected float tabOffset - The tab offset on the y axis.
protected LoadAnimationType loadAnimationType - The load animation type of this menu.
protected List<UISetting> uiSettings - The list of settings to store.

public void PlayAudio() - Triggers the button click sound to be played.
public void SaveSetting(string settingName) - Saves the given setting from certain UI elements. **settingName** - The name to save under.
public float GetFloatSetting(string settingName) - Returns the value of a setting. **settingName** - The name of the setting.
public int GetIntSetting(string settingName) - Returns the value of a setting. **settingName** - The name of the setting.
public string GetStringSetting(string settingName) - Returns the value of a setting. **settingName** - The name of the setting.
public void OpenMenu() - Opens the menu and its currently active window(s).
public void CloseMenu() - Closes the menu and its currently active window(s).
public void HideMenu() - Hides the menu and its currently active window(s).
public void OpenWindow(Window pWindow) - Opens a window if it is in this menu. **pWindow** - A reference to the Window object.
public void OpenWindow(string pName) - Opens a window if it is in this menu. **pName** - The name of the window.

public void CloseWindow(Window pWindow) - Closes a window if it is in this menu. **pWindow** - A reference to the Window object.
public void CloseWindow(string pName) - Closes a window if it is in this menu. **pName** - The name of the window.
public void LoadScene(int pBuildIndex) - Loads a scene while showing a loading screen. **pBuildIndex** - The build index of the scene.
public void LoadScene(int pSceneName) - Loads a scene while showing a loading screen. **pSceneName** - The build index of the scene.

Window

protected Tab tab - The tab on this window.
protected Animator animator - The Animator attached to this window. This is not needed.
protected GameObject graphics - The GameObject holding all of the graphics for this window.
protected new string name - The name of this window.
protected bool active - Whether or not the window is active.
protected bool setup - Whether or not the window was setup.
protected WindowAnimationType animationType - The animation type of this window.
protected string openTrigger - The name of the trigger animator parameter that opens the window. This is not needed unless an animator with trigger parameters is used.
protected string closeTrigger - The name of the trigger animator parameter that closes the window. This is not needed unless an animator with trigger parameters is used.
protected string openBool - The name of the boolean animator parameter that controls the open state of the window. This is not needed unless an animator with a boolean parameter is used.

public void Open() - Opens the window and triggers an animation if there is one.
public void Close() - Closes the window and triggers an animation if there is one.
public void Setup() - Sets up the window. Only for use by the Menu script.

Tab

protected Image image - The Image attached to this tab.
protected Animator animator - The Animator attached to this tab.
protected bool active - Whether or not the window is active.

protected bool setup - Whether or not the window was setup.
protected TabAnimationType animationType - The animation type of this tab.
protected Color activeColor - The color of the tab when active. This is not needed unless a color animation is used.
protected Color inactiveColor - The color of the tab when inactive. This is not needed unless a color animation is used.
protected Sprite activeSprite - The sprite of the tab when active. This is not needed unless a sprite animation is used.
protected Sprite inactiveSprite - The sprite of the tab when inactive. This is not needed unless a sprite animation is used.
protected string activeTrigger - The name of the trigger animator parameter that activates the tab. This is not needed unless an animator with trigger parameters is used.
protected string inactiveTrigger - The name of the trigger animator parameter that deactivates the tab. This is not needed unless an animator with trigger parameters is used.
protected string activeBool - The name of the boolean animator parameter that controls the active state of the tab. This is not needed unless an animator with a boolean parameter is used.
protected string activeAnimation - The name of the activation animation. This is not needed unless an animator is used.
protected string inactiveAnimation - The name of the deactivation animation. This is not needed unless an animator is used.

public void Setup(bool pActive) - Sets up the tab. Only for use by the Menu script.
pActive - Whether or not the tab is active.

OptionsList

protected Menu menu - The Menu this list is a part of. Used for autosaving of the option.
protected Text text - The Text object this list shows. Only required if using Text and not TextMeshPro for the list.
protected TextMeshProUGUI textMPro - The TextMeshPro object this list shows. Only required if using TextMeshPro and not Text for the list.
protected string settingName - The name to save the current option as.
protected int currentOption - The current option.
protected List<string> options - The list of available options.
protected bool saveOption - Whether or not to save the current option.

public void ChangeOption(int direction) - Changes the currently displayed option. **direction** - Which direction to change the option, either 1 (right), or -1 (left).
public string GetOption() - Returns the selected option.

UISetting

Stores a name and a UI element to save the value of.
public string Name - The name of the setting.
public Slider UISlider - A reference to the Slider using this setting, if there is one.
public Toggle UIToggle - A reference to the Toggle using this setting, if there is one.
public Text UIText - A reference to the Text using this setting, if there is one.
public TextMeshProUGUI UITextMeshProUGUI - A reference to the TextMeshProUGUI using this setting, if there is one.
public Dropdown UIDropdown - A reference to the Dropdown using this setting, if there is one.
public TMP_Dropdown UITMP_Dropdown - A reference to the TMP_Dropdown using this setting, if there is one.
public OptionsList UIOptionsList - A reference to the OptionsList using this setting, if there is one.

LoadAnimationType

The type of the load animation.
FillAmount - Changes the image fill amount based on how much the scene is loaded.
Width - Changes the image width based on how much the scene is loaded.

TabAlign

The way to align the tabs at the top of the menu on runtime.
Center - The tabs will be aligned to the center of the menu.
Left - The tabs will be aligned to the left of the menu.
Right - The tabs will be aligned to the right of the menu.

MenuType

The type of a menu.
Solitary - The menu has many windows that can be open at the same time, and open/closed at any time.

Tab - The menu has many windows, but only one can be open at a time, although they can be open/closed by a tab at any time.

WindowAnimationType

The animation type of a window.
None - The window is not animated.
Trigger - The window is animated by a trigger.
Boolean - The window is animated by a boolean.

TabAnimationType

The animation type of a tab.
None - Nothing happens when the tab's active state is changed.
Color - The tab's color changes when the tab's active state is changed.
Sprite - The tab's sprite changes when the tab's active state is changed.
Trigger - An animation is triggered by a trigger animator parameter when the tab's active state is changed.
Boolean - An animation is triggered by a boolean animator parameter when the tab's active state is changed.