

Hello!

Thank you for downloading the Game Options Settings package. I hope this package takes care of the boring settings menu so you can get back to creating your game. This package features two components, AudioSettings.cs and VisualSettings.cs that are responsible for saving and setting your UI components to the correct player system preferences. All preferences are automatically saved when the menu is closed.

Features:

- Master Volume slider
- Music Volume slider
- SFX Volume slider
- vSync toggle
- Full Screen toggle
- Resolution dropdown
- FPS Cap dropdown
- Graphics dropdown
- Automatic PlayerPrefs saving for all options
- Smart UI initialization
- Auto resolution detect on Full Screen
- Desired Resolution Populator

Setting up your options menu:

In your Canvas create your video and audio menus then attach the scripts to the correct ui components. Reference the UI elements in the scripts, then call the dynamic set functions from the On Value change event of the correct element. The AudioSettings script requires that you set up an audio mixer group and expose its volume parameters. The VisualSettings script requires that you define your target resolutions and define your graphical settings in the Unity Quality Settings menu.

Exposing Audio Mixer Groups:

To use the audio settings component:

- In Unity, click on Window > Audio > Audio Mixer
- In the Audio Mixer window, click the plus button to create a new mixer
- Name the mixer MasterVolume
- Click on the MasterVolume mixer
- In the Inspector window that opens, right click on the word volume, and click "Expose Volume"
- Back in the AudioMixer window, click on Exposed Parameters
- Double click the name and rename it to **MasterVolume**
- In the groups tab, Click on Master
- Click the plus sign and add a group named **SFX**
- Expose the volume for your SFX group and name it **SFXVolume**

- Click the plus sign and add a group named **Music**
- Expose the volume for your Music group and name it **MusicVolume**

NOTE: AudioSources that are SFX need to use the **SFX** audio mixer as their output

NOTE: AudioSources that are music need to use the **Music** audio mixer group as their output

Attaching your Audio Settings Script:

To set up your audio settings component:

- On your audio option settings panel, click Add Component
- Select the AudioSettings component
- Reference the AudioMixer and 3 UI sliders you will use for master, SFX, and music in the script
- Set each slider's minimum value to .0001, and its Max Value to 1
- In the On Value Changes event, click the plus sign
- Drag in the AudioSettings script then select the corresponding dynamic float function
For Example, select SetMasterVolume() for the Master slider

Now when you run the game you can slide the sliders to control your different audio volumes.

On destroying or disabling the AudioSettings.cs script, your values will be automatically saved.

NOTE: you can call the function SaveAllData() to save the settings values any time you see fit.

Attaching your Video Settings Script:

To set up your video settings component:

- On your video option settings panel, click Add Component
- Select the VideoSettings component
- Reference the 2 Toggles and 3 dropdown components inside the script

NOTE: The dropdown components require a TMP_Dropdown component. This is part of the TextMeshPro library

- In each respective UI component, in the On Value Changed field, click the plus sign, reference the VisualSettings script and select the dynamic function that corresponds to the UI element

For Example, for the vSync toggle, select SetVsyncValue(). For the Resolution dropdown, select SetResolutionValue()

- In the Video Settings component, check the Full Screen Resolution Changing if you want to be able to adjust your resolution when the game is running in full screen mode, otherwise the game will select the best resolution dynamically and disable the Resolution dropdown for you
- In the Video Settings component, add Desired Resolutions to reduce the populated list of Resolution options to only the ones you want to support. This component automatically will populate the Native resolution of the display you are using at run time

Now when you run the game you can use the toggles and dropdowns to control your different visual settings. Resolution, FPS Cap, and Graphics dropdown fields are automatically

populated based on your Unity Project Settings. On destroying or disabling the VideoSettings.cs script, your values will be automatically saved.

NOTE: To add or remove Graphic settings, go to Project Settings > Quality > and set up your own graphics levels, they will dynamically populate the Graphics dropdown

NOTE: you can call the function SaveAllData() to save the settings values any time you see fit.