

HONETi Unreal GUI Manual v.1.4.0

Demo scene..... 2

Widgets & components..... 3

Windows..... 3

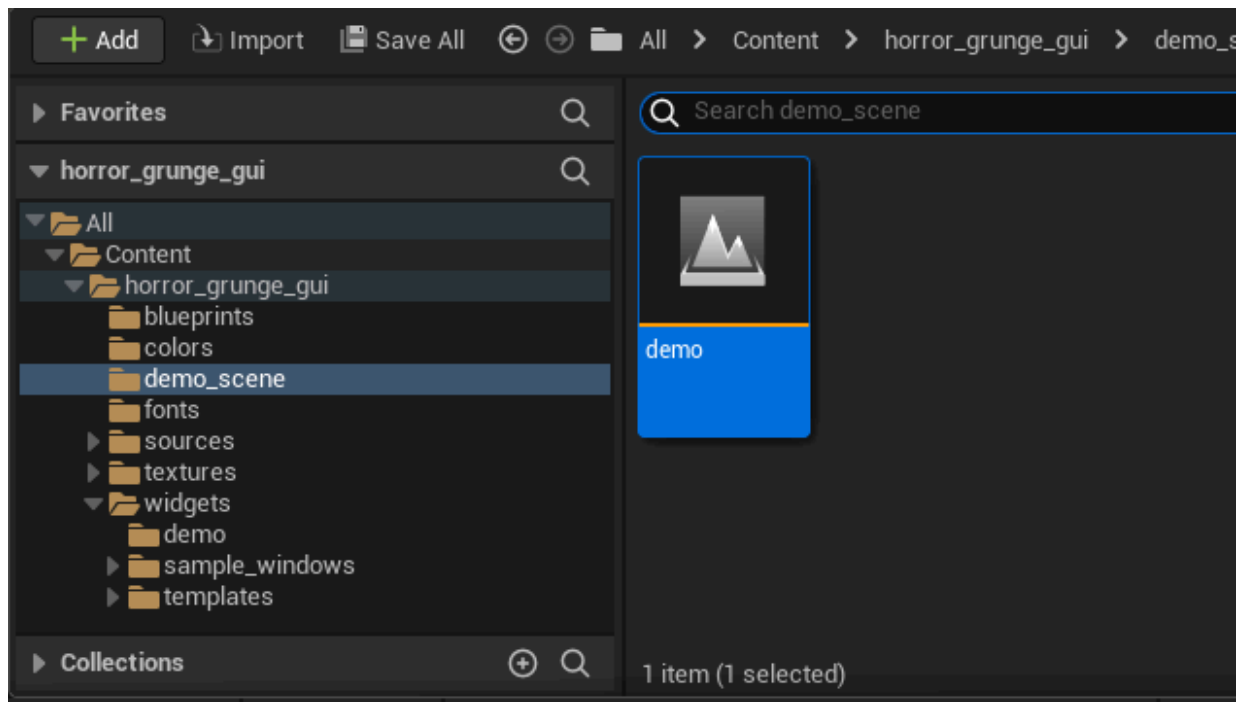
Sources..... 5

Scaling..... 6

How to replace fonts..... 8

Creating a 5-Star Widget in Unreal Engine..... 10

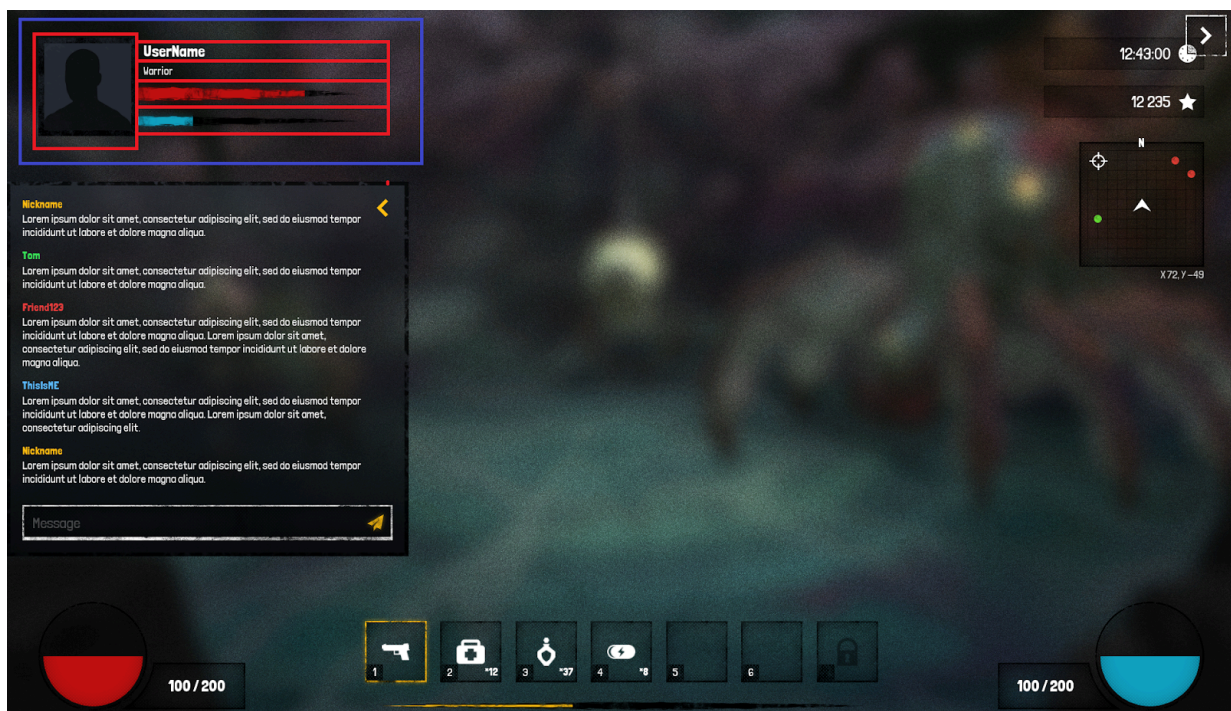
To open the project you need to run the demo inside. This launches the demo gui view and allows you to view the entire project.



Example: Horror GUI project.

Demo scene

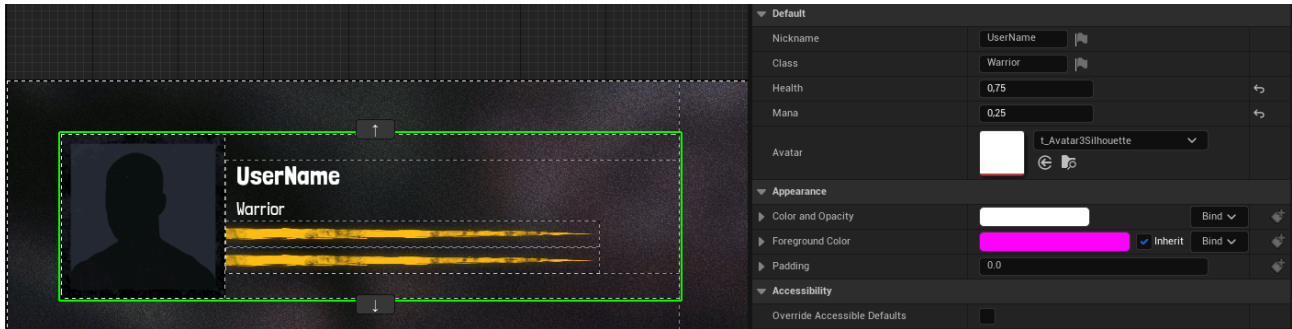
The project consists of widgets and windows that are placed on the demo scene to show them in action.



Example: Horror GUI hud screen.

Widgets & components

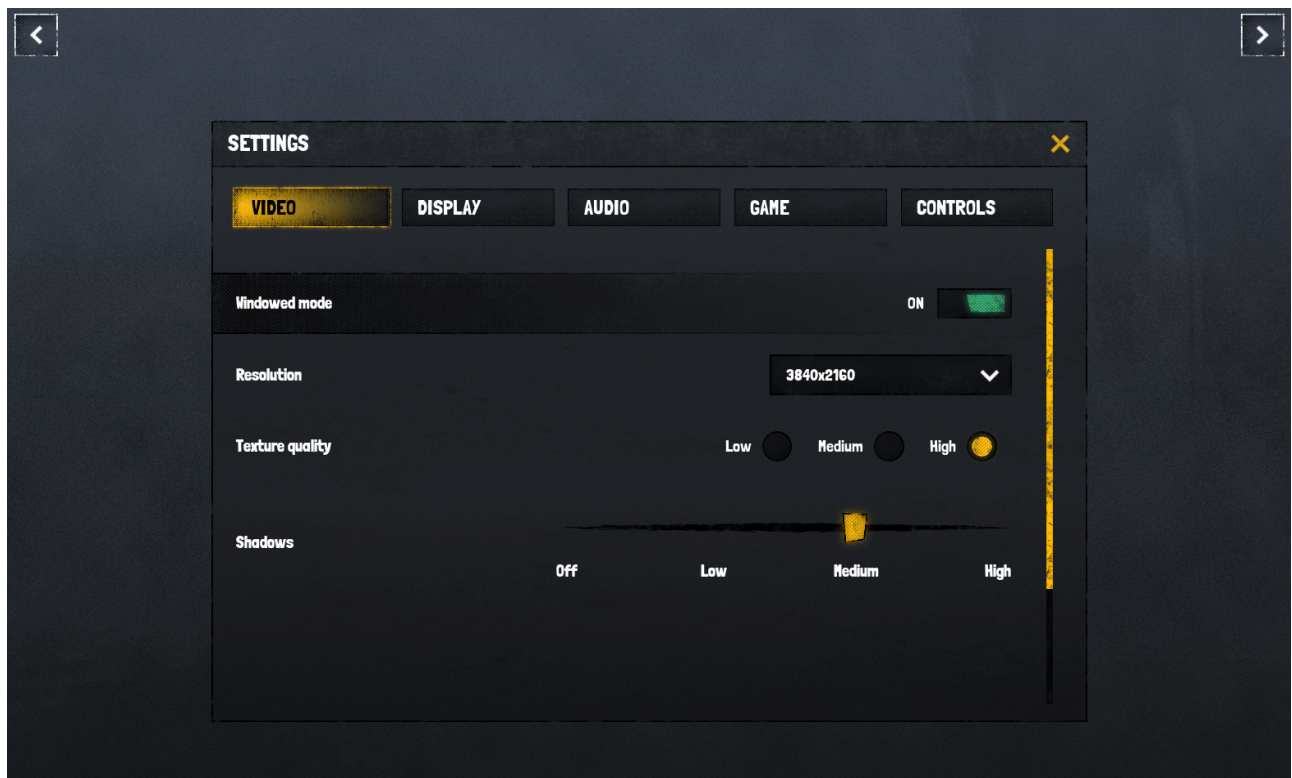
The widget on the Game demo screen is marked with a blue border, inside this widget there are minor window components marked in red. The user status widget consists of 2 text blocks, 2 progress bars and 1 image. By clicking on this widget you can set the listed components in the window:



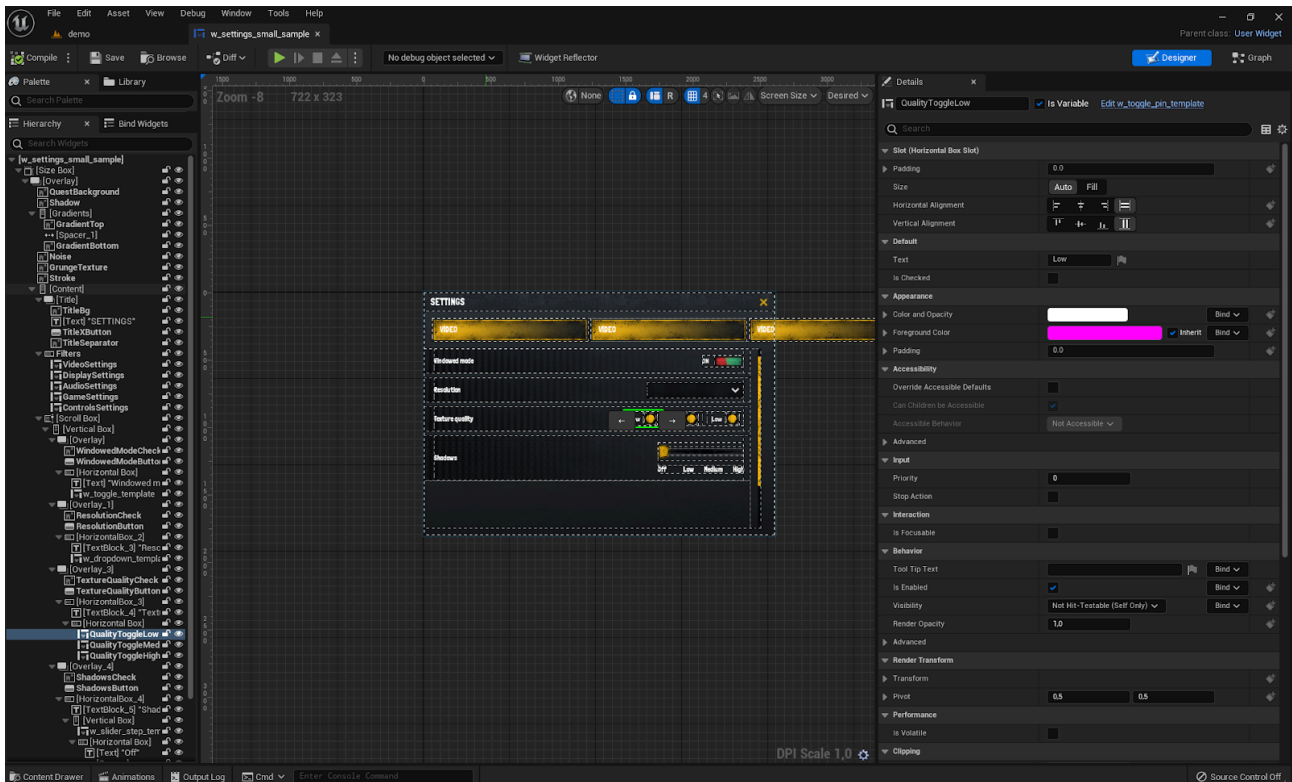
Example: Horror GUI - Widget user status.

Windows

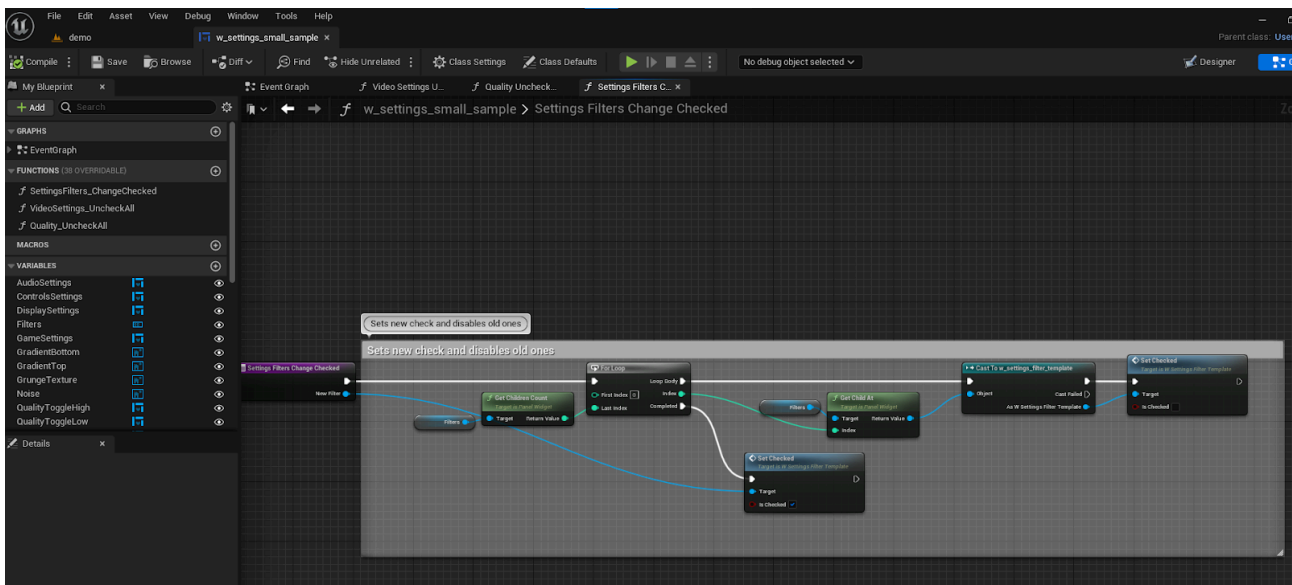
Windows are similar to the widgets presented above, they are larger and can contain many widgets and often have their own independent functions to change the visual appearance of the window, for example the settings window.



Example: Horror GUI - Settings window.



Example: Horror GUI - The settings window inside the editor.



Example: Horror GUI - Toggle filter function inside settings window.

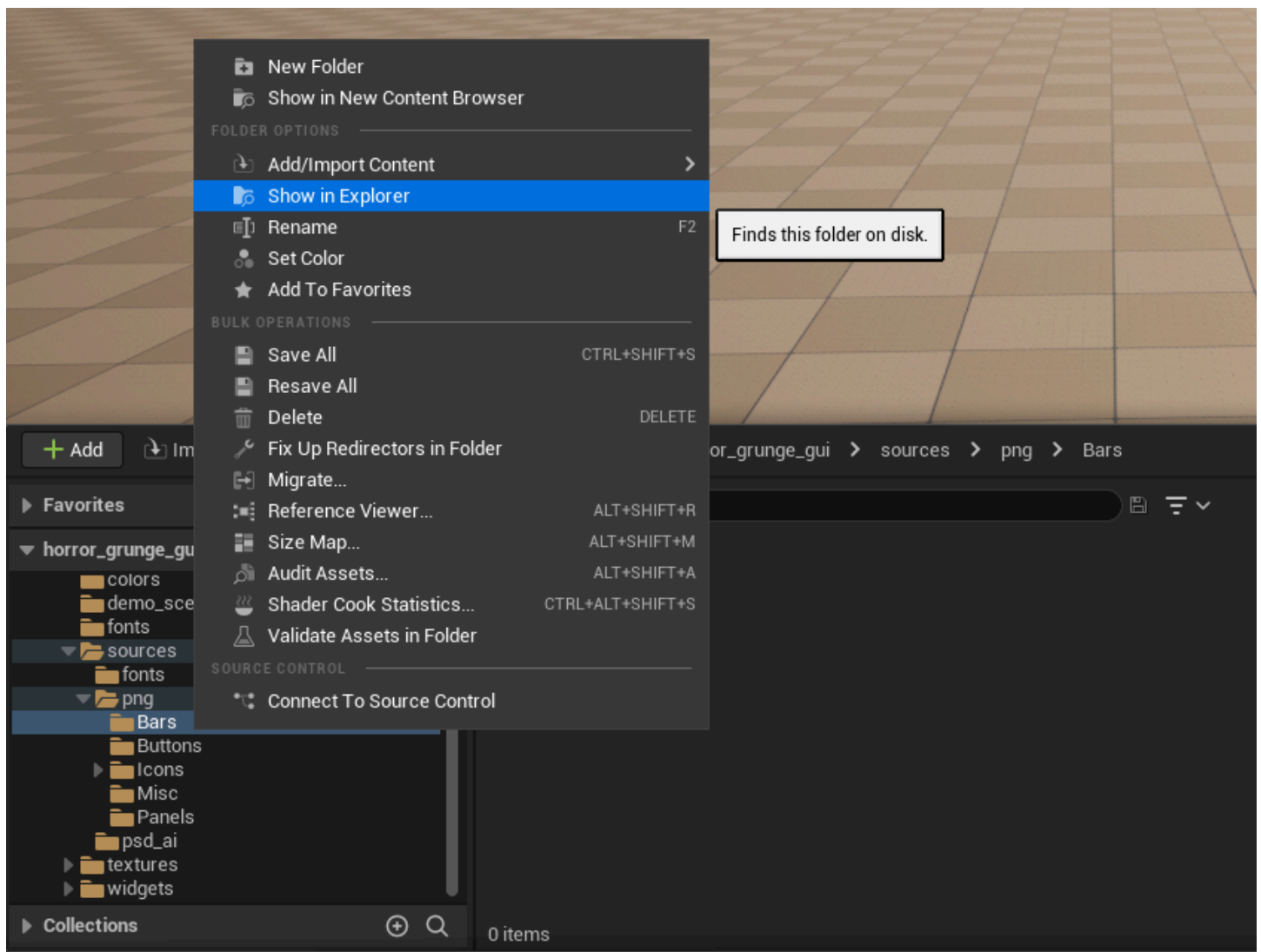
So the hierarchy inside the project is as follows:

1. **The demo** consists of windows, widgets and sometimes basic components such as images or text shown to the user.
2. **Window** consists of widgets and basic components can be used in demo to be shown to the user.
3. **Widget** consists of basic components that can have assigned values by the user.

Sources

Png, psd, fonts and ai files can be found (via system files browser) inside "sources" directory.

Source files are not visible inside the UE editor!

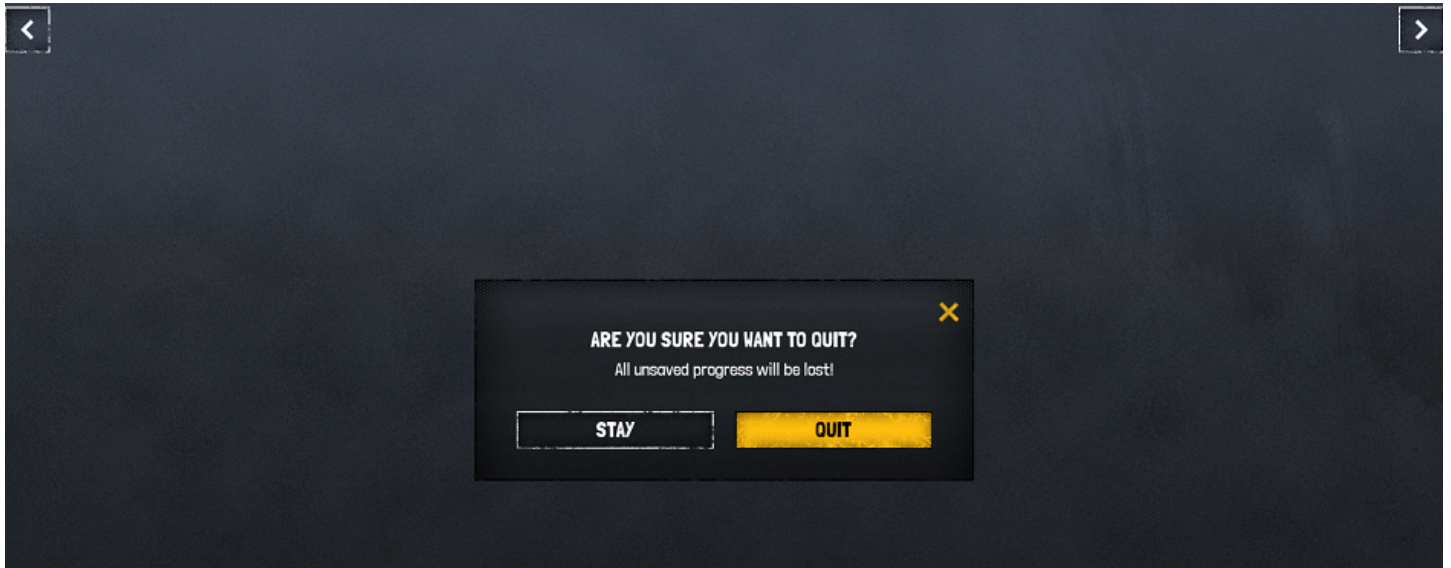


Example: Horror GUI

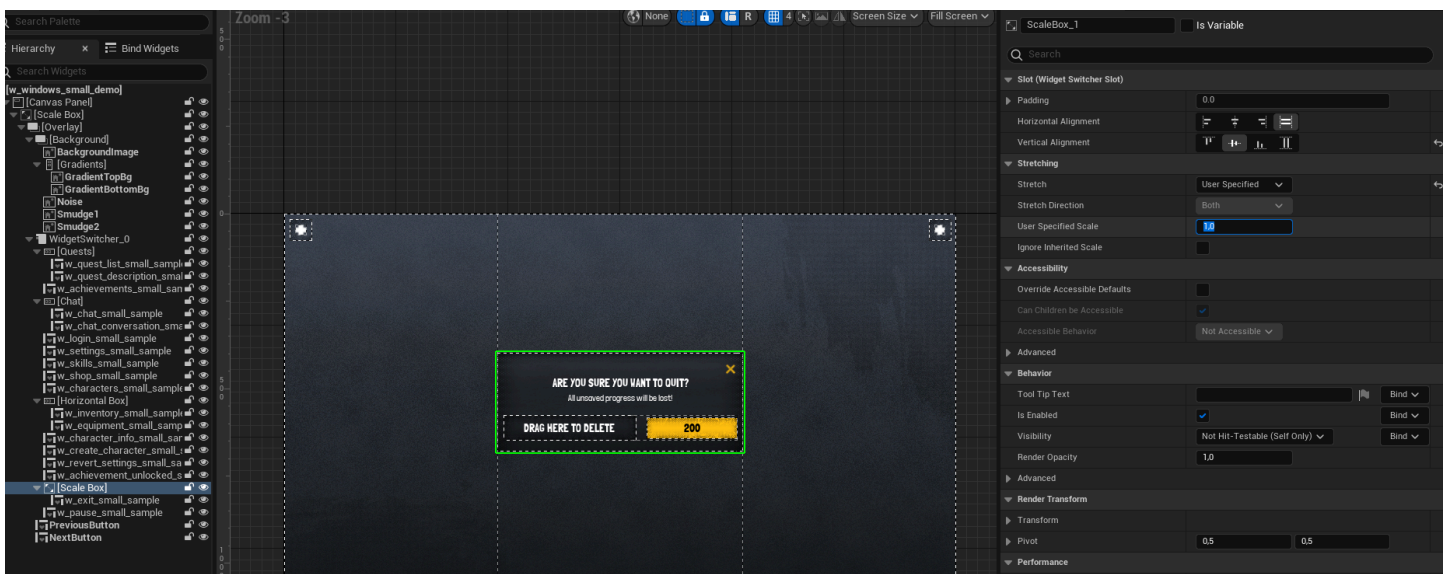
In some cases, the "Source" directory is one level higher than the "Content" directory. Browse it with the system file browser.

Scaling

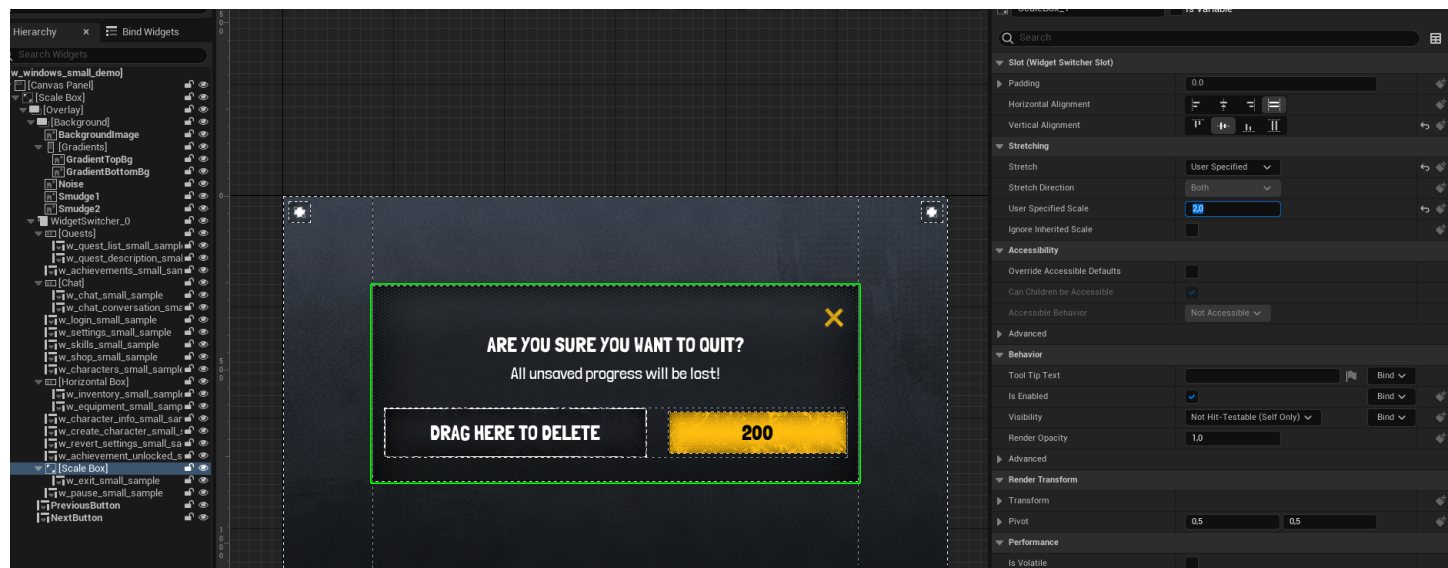
It is possible to scale windows using a scale box. You need to wrap the targeted window with a scale box on the demo scene and set the desired values.



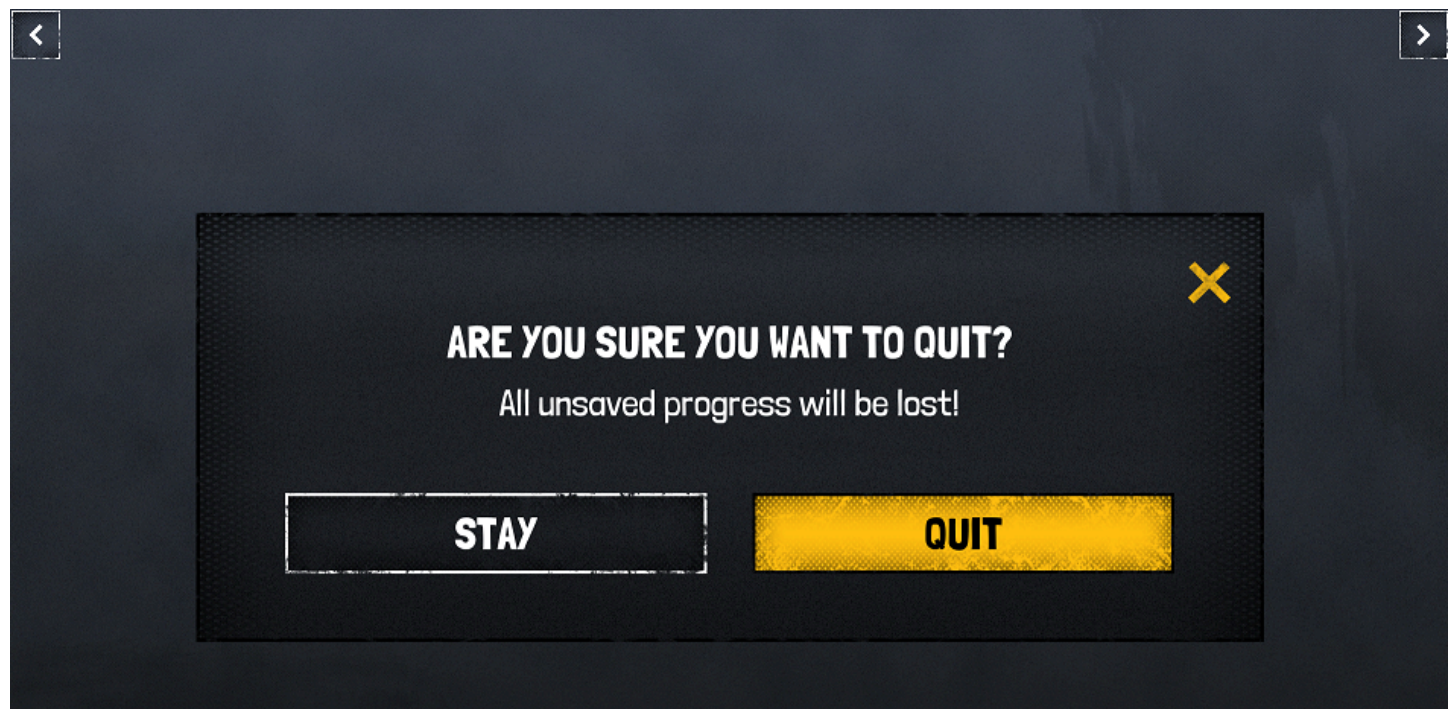
Window in demo scene before



Setting the scale ratio



New scale ratio seen inside editor



The new look on the demo scene.

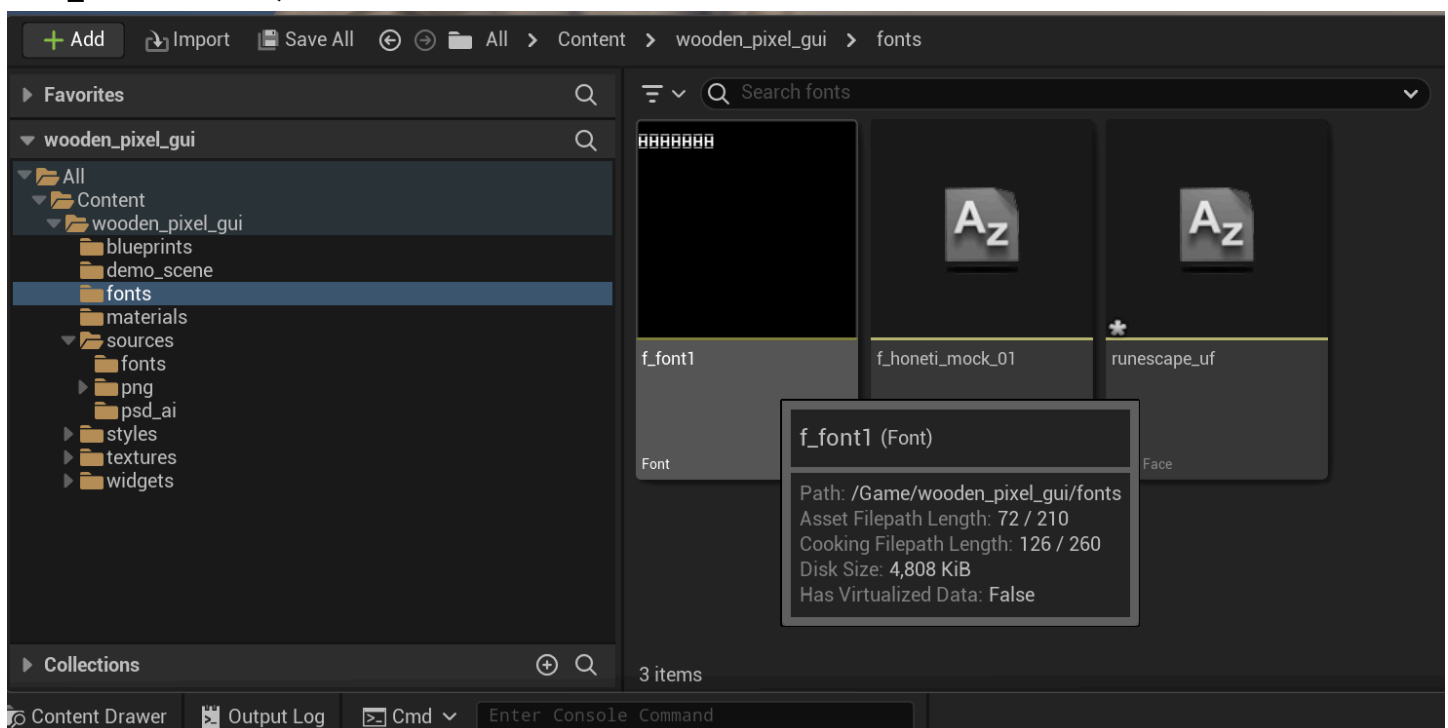
How to replace fonts

Due to EU Marketplace requirements, sometimes, fonts are initially replaced by placeholders and must be installed manually. In this case, you will find detailed instructions on how to do this in the source/fonts/readme.txt file. Below you will find basic instructions on how to replace the font:

1. Import a new font (Font Face):

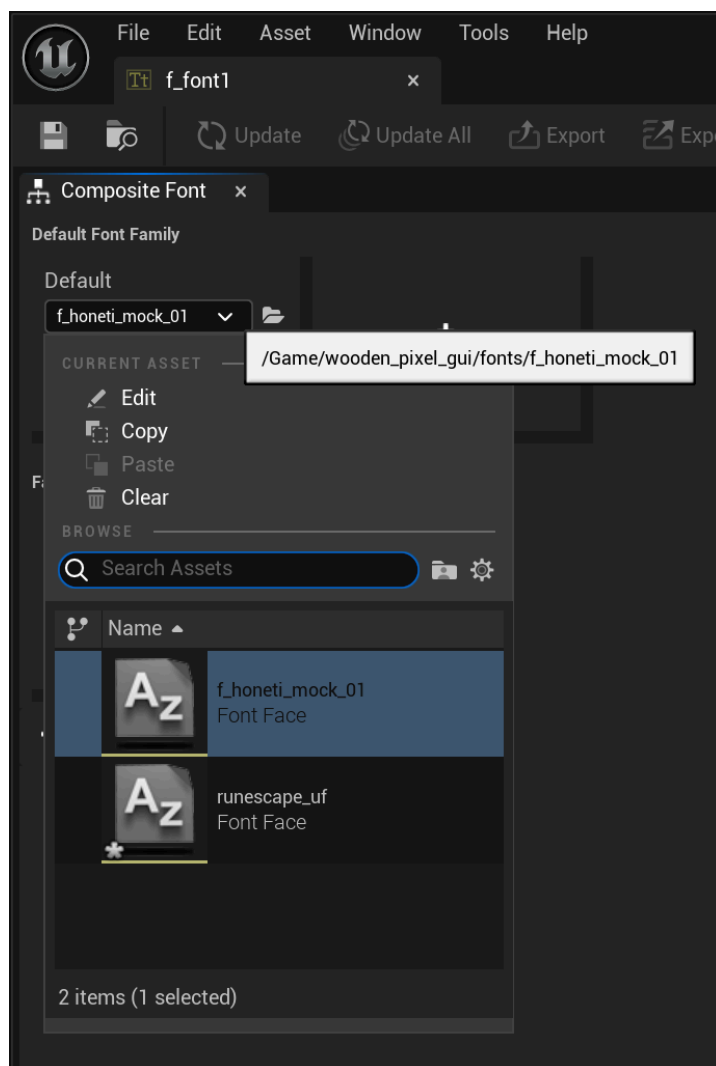
<https://dev.epicgames.com/documentation/en-us/unreal-engine/importing-fonts-in-unreal-engine>

2. Go to the fonts directory in the project and double-click the font file you want to change (ex. f_font1.uasset)

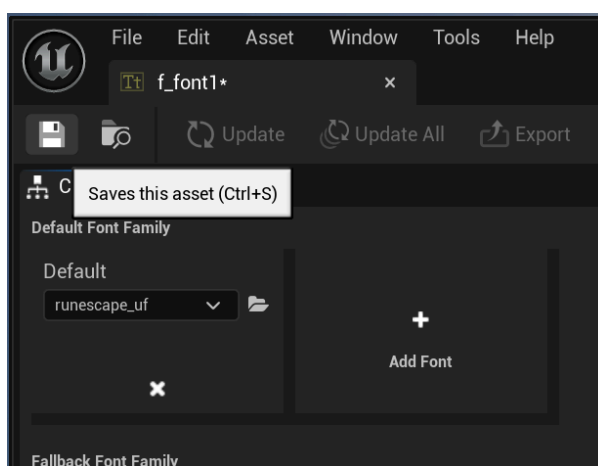


Double-click the font you want to change

3. On the Composite Fonts tab, update the font face by opening the SelectBox and selecting the font you imported in step 2.



4. Make sure to save all changes in font asset.



Saving changes

Creating a 5-Star Widget in Unreal Engine

In this guide, we will show you how to create a five-star rating widget based on an existing three-star widget in Unreal Engine. This process involves duplicating, modifying Blueprint logic, and updating widget content to ensure the new version works correctly.

Steps to Create

1. Duplicate the 3-Star Widget

The best way to create a 5-star widget is to duplicate an existing 3-star widget. This ensures consistency in style and functionality.

1. Locate the existing 3-star widget in the UMG Editor.
2. Right-click on the widget and select Duplicate, then name the new version appropriately (e.g., FiveStarWidget).
3. Open the duplicated widget in the Widget Blueprint Editor to modify its structure.

2. Add Two Additional Stars

Now you need to add the missing stars:

1. In the Designer tab, select an existing star and copy-paste it twice.
2. Ensure that all stars are correctly aligned within the Horizontal Box or another layout container.
3. Adjust the size and spacing as needed.

3. Update the fill_stars Function

The widget requires modifications in Blueprint logic to account for five stars:

1. Open the Graph tab and locate the fill_stars function.
2. Update the loop or logic handling the star fill to accommodate five stars instead of three.
3. Ensure each star updates correctly based on the rating value.

4. Modify the Event Graph

The Event Graph requires some adjustments:

1. In the Event Graph, add Size Boxes for the two additional stars if needed.
2. Configure constraints to ensure all stars maintain a consistent size.
3. Test the widget in the Viewport to confirm the stars react properly to rating updates.

5. Update Function Calls

Next, you need to update related function calls:

1. Replace the previous 3-star widget references with the new 5-star widget in any UI where it's used.
2. Ensure all function calls that reference star count or update behavior are properly assigned.
3. Update the set_color function (if applicable) to handle five stars properly.

Final Notes

- The project does not include a half-star texture, so the system only supports full stars.
- After completing these modifications, test the new widget to ensure it functions correctly across different rating values.

