HOUR 22

Introduction to UMG: Unreal Motion Graphics

INTRODUCTION

This lecture introduces you to the Unreal Motion Graphics (UMG) UI Designer Editor. UMG is an editor in UE4 used for creating 2D interfaces and HUDs.

LECTURE GOALS AND OUTCOMES

Goals

The goals of this lecture are to

- Learn to create a Widget Blueprint
- Learn about the UMG Editor interface
- Learn about anchors and DPI scaling
- Learn about common widgets
- Learn about Texture settings for use in an interface

Outcomes

By the end of this lecture you will be able to

- Create a Widget Blueprint
- Place a widget on an interface
- Prep a Texture for use in an interface

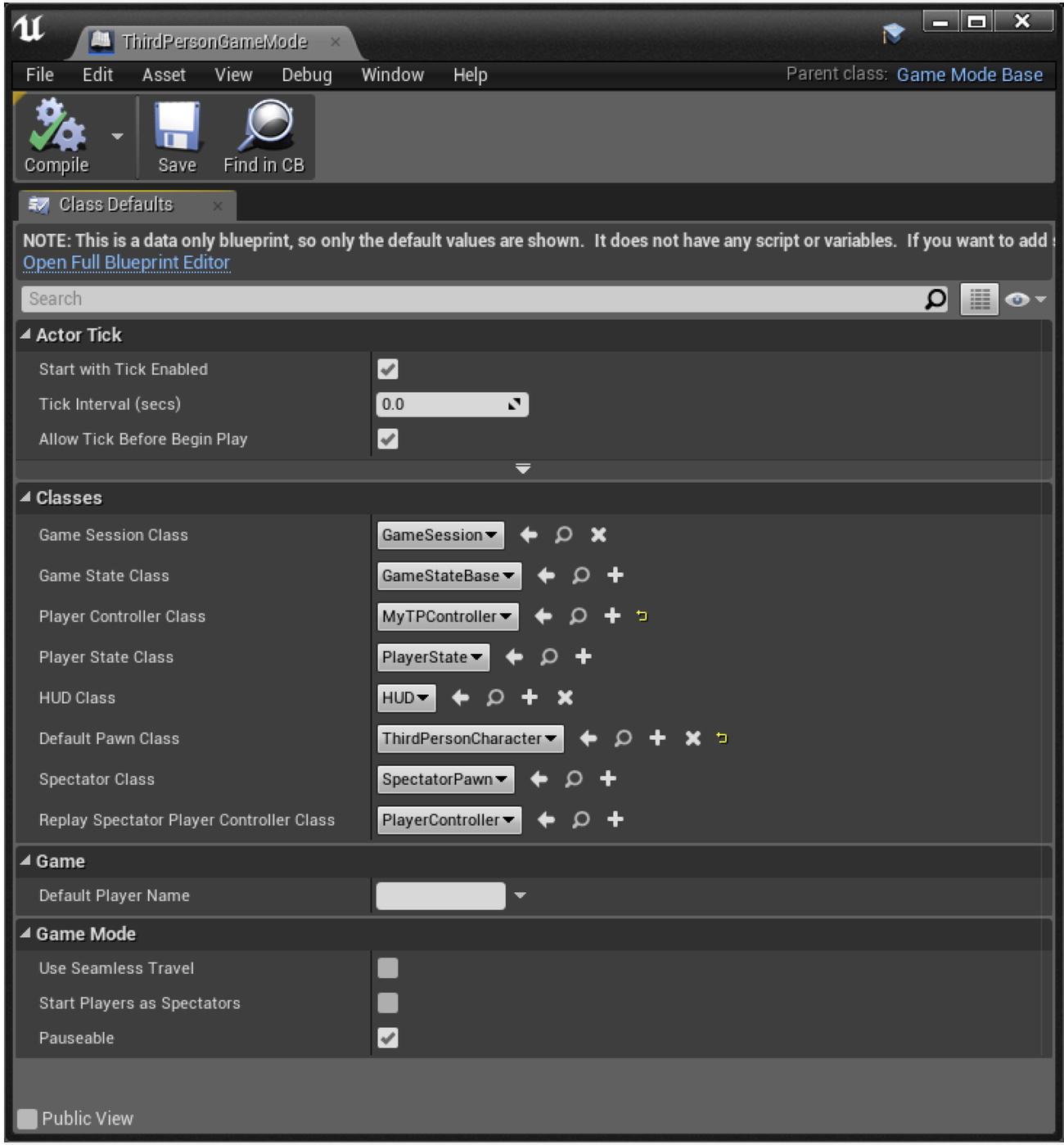
WIDGET BLUEPRINTS

Basic Concepts

HUD CLASS

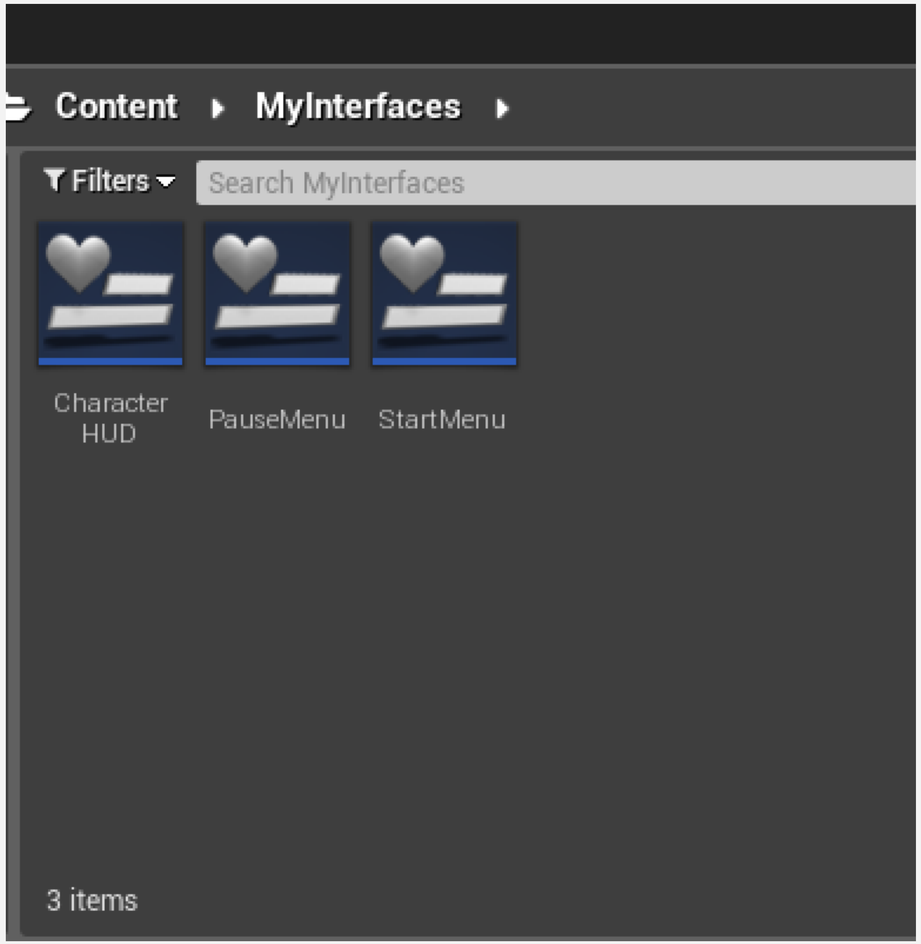
The Gameplay Framework includes a HUD class for creating interfaces using Slate (the legacy HUD system).

Slate is programmer-friendly but not artist- and designer-friendly. Epic has created the UMG Editor, which combines artist-friendly tools for layout with the power of Blueprint.



WIDGET BLUEPRINT ASSET

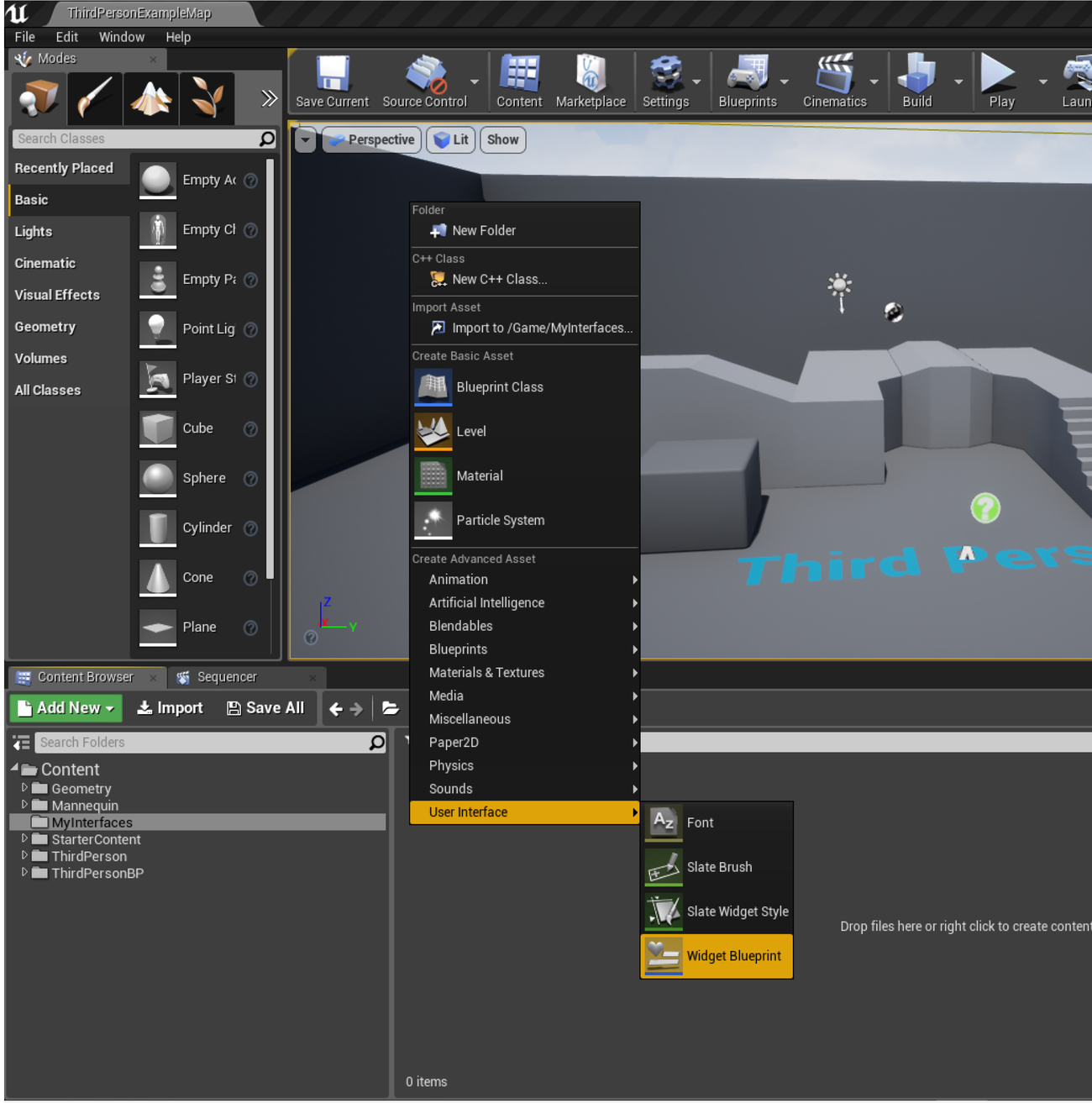
A Widget Blueprint asset is a special Blueprint that is created in the Content Browser and edited in the UMG Editor.



CREATING A WIDGET BLUEPRINT

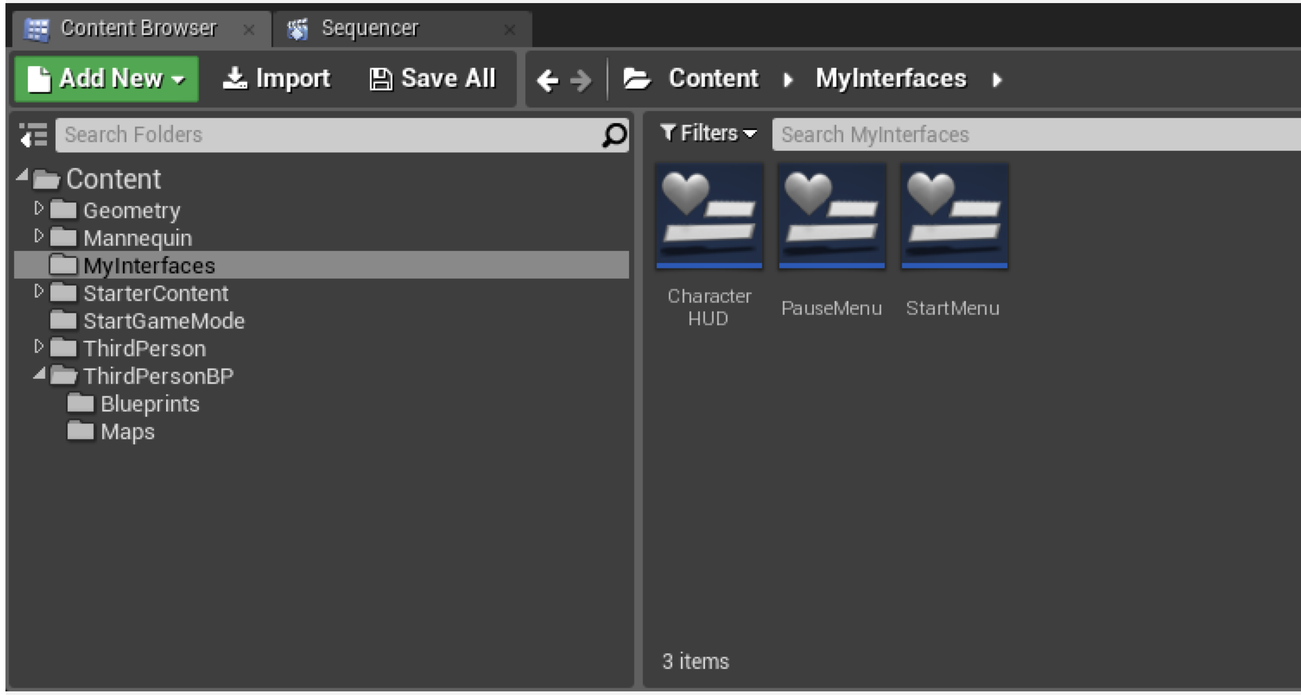
To create a Widget Blueprint, first create a new folder in the Content Browser and give it a name such as “MyInterfaces”.

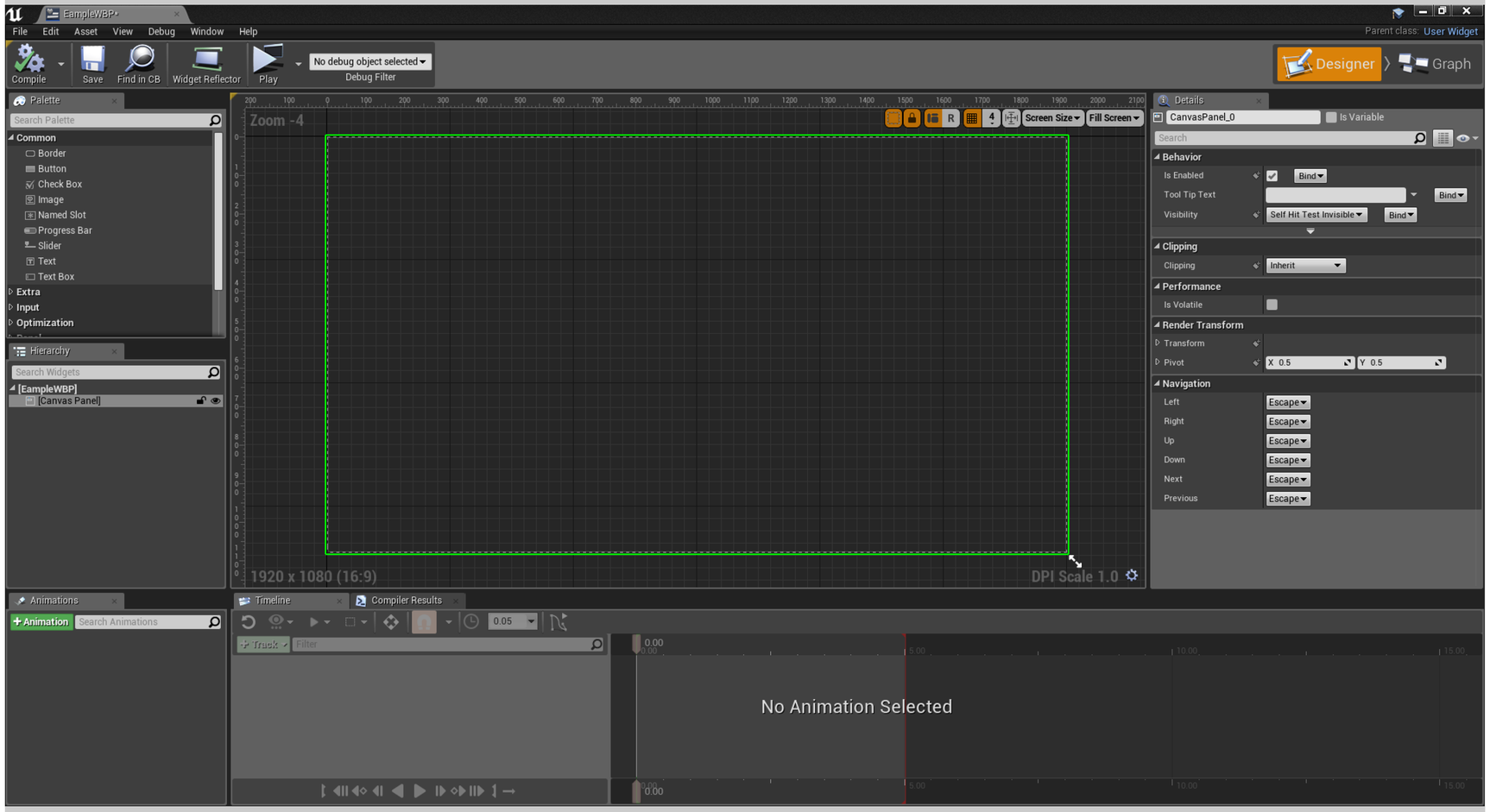
Then right-click in the Content Browser, navigate to User Interface under Create Advanced Asset, and select Widget Blueprint.



CREATING A WIDGET BLUEPRINT

Rename the Widget Blueprint and then double-click on it to open it in the UMG Editor.





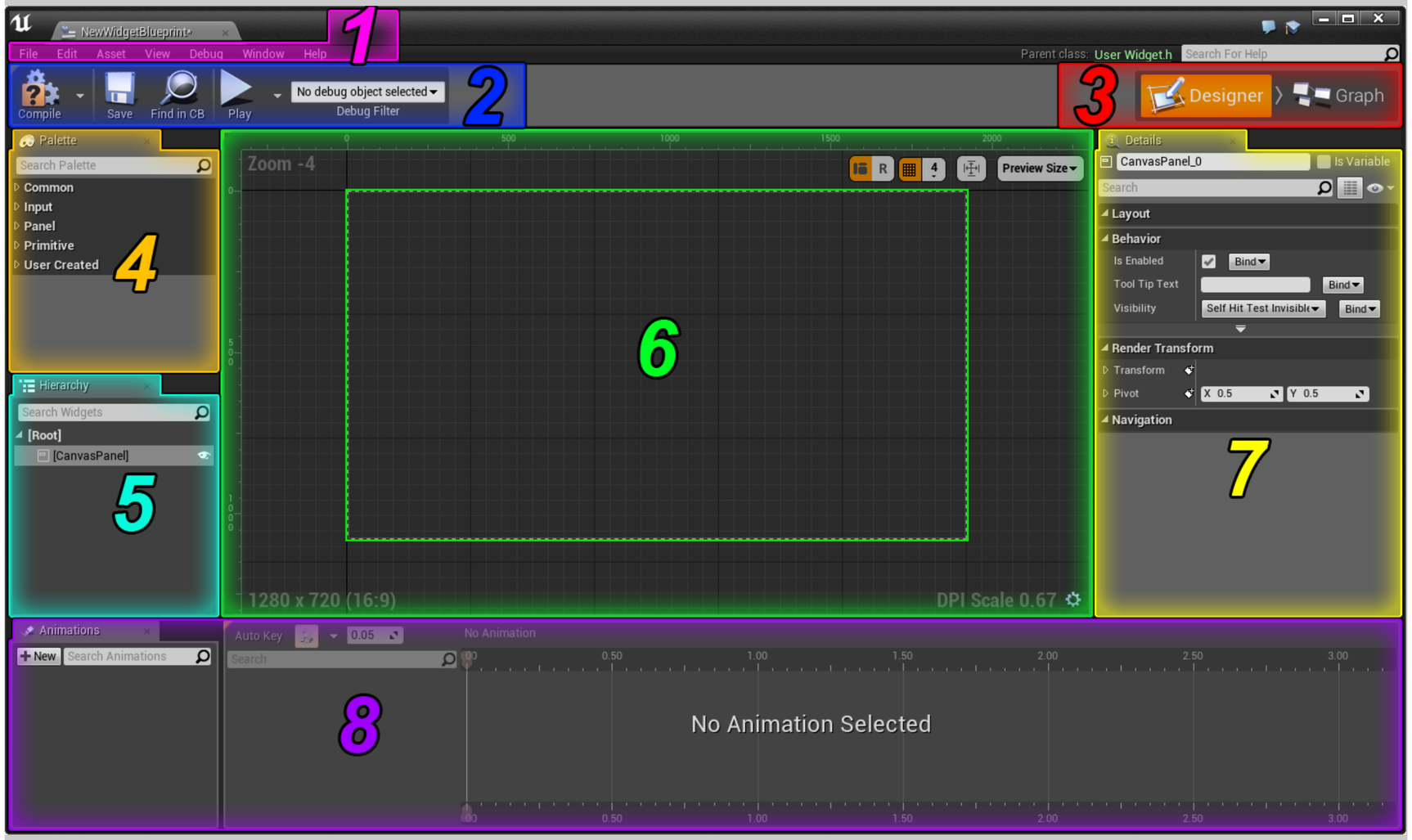
UMG Editor Designer mode

UMG INTERFACE

The UMG Editor has two modes: the Designer mode and the Graph mode.

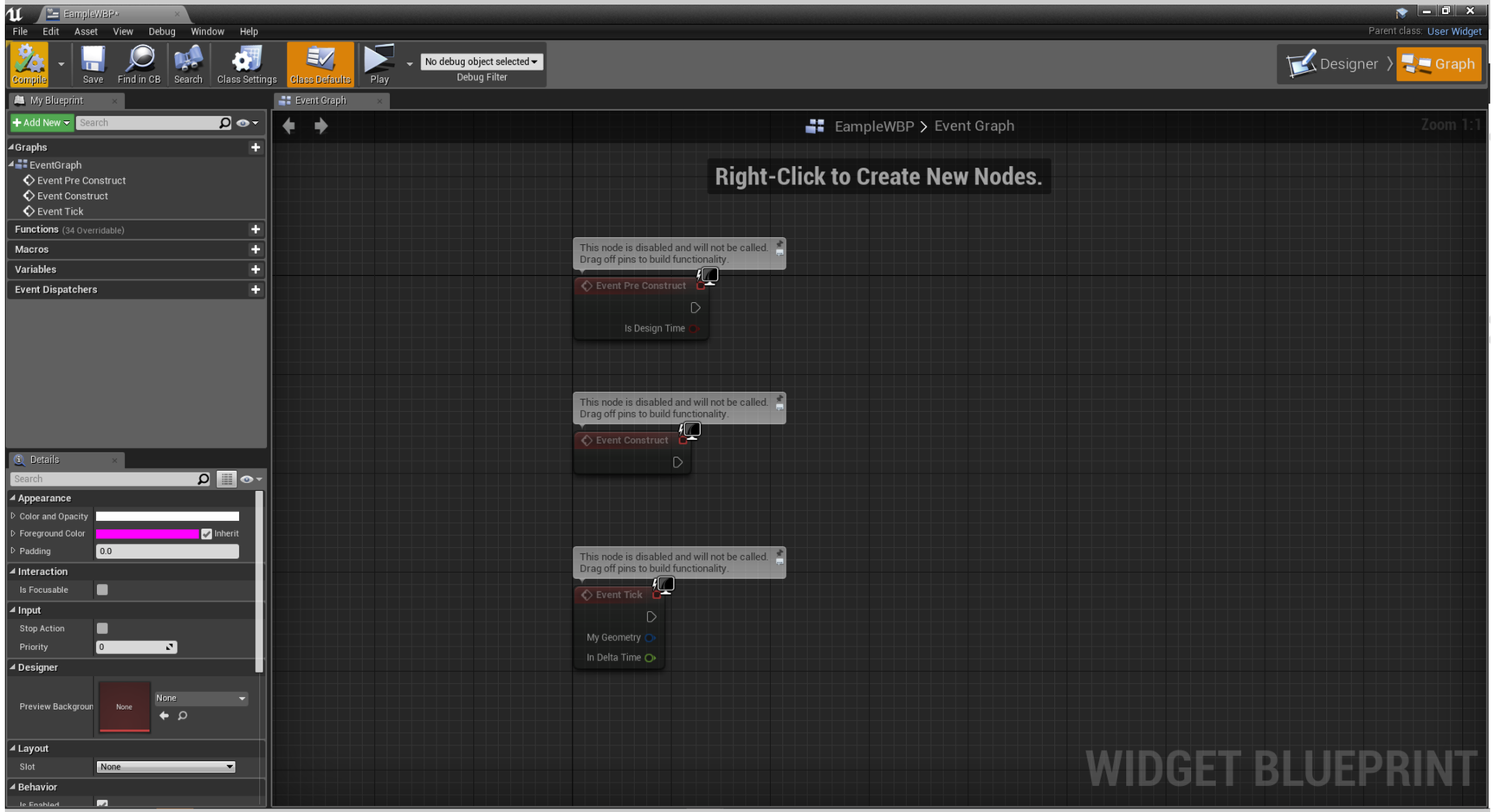
The Designer mode is used to place interface components called widgets, and the Graph mode is a Blueprint Editor used to script interface functionality.

DESIGNER MODE



UMG Editor Designer mode

1. **Menu bar**: The common menu bar.
2. **Toolbar**: Contains a number of commonly used functions for the Blueprint Editor, such as Compile, Save, and Play.
3. **Editor mode**: Switches the UMG Widget Blueprint Editor between Designer and Graph modes.
4. **Palette panel**: Contains the list of widgets that you can drag and drop into the Designer panel. Displays any class inheriting from UWidget.
5. **Hierarchy panel**: Displays the parenting structure of the user widget. You can also drag widgets into this panel.
6. **Designer panel**: Displays the visual representation of the layout. Widgets dragged into this panel can be manipulated.
7. **Details panel**: Displays the properties of the currently selected widget.
8. **Animations panel**: Displays the animation tracks for UMG, which allow you to keyframe animations for your widgets.



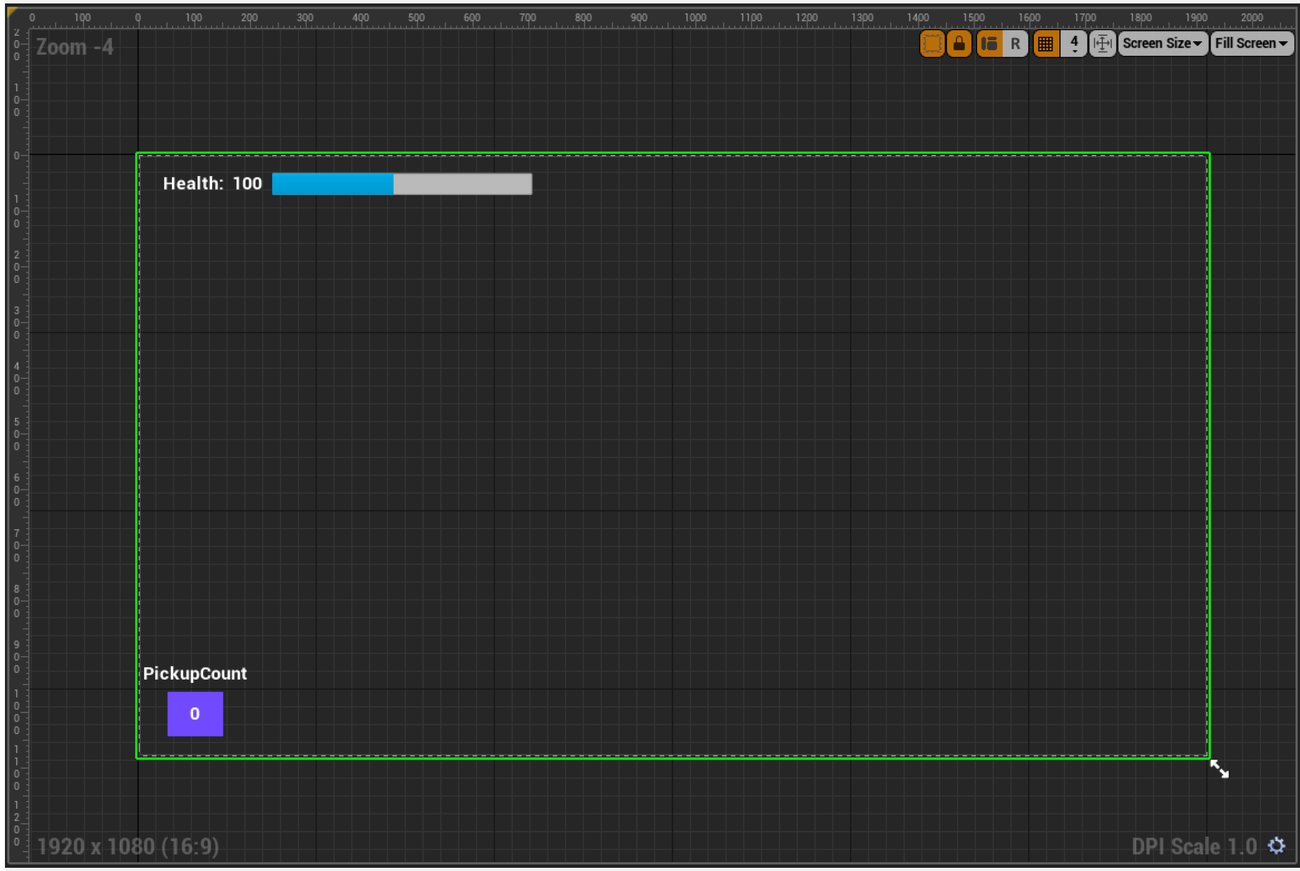
UMG Graph mode

GRAPH MODE

The Graph mode is where you add functionality to your placed widgets. It looks like the standard Blueprint Editor except the Details panel is in the lower left corner by default.

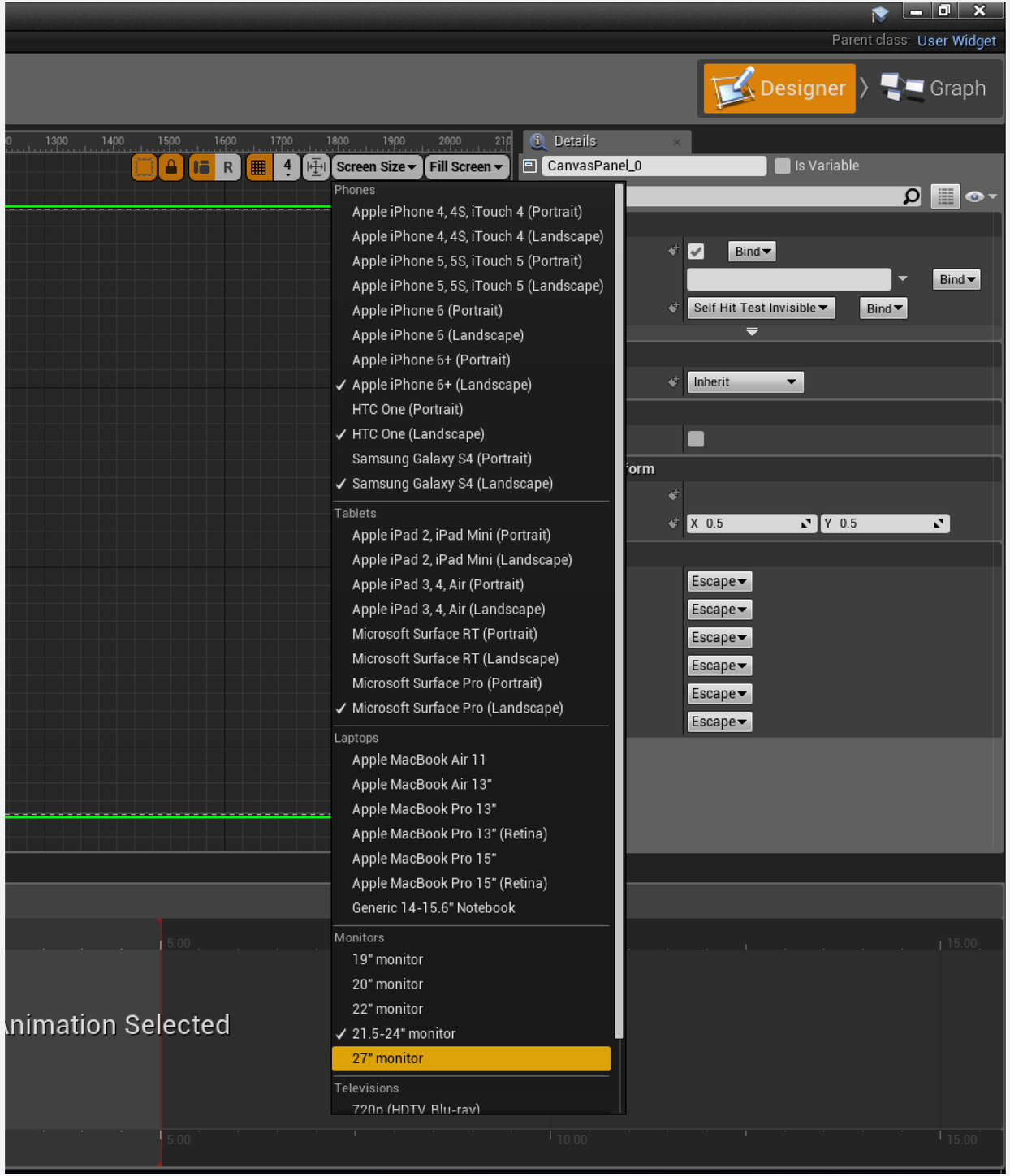
DESIGNER PANEL

The Designer panel represents the screen size and resolution for the interface or HUD you are creating.



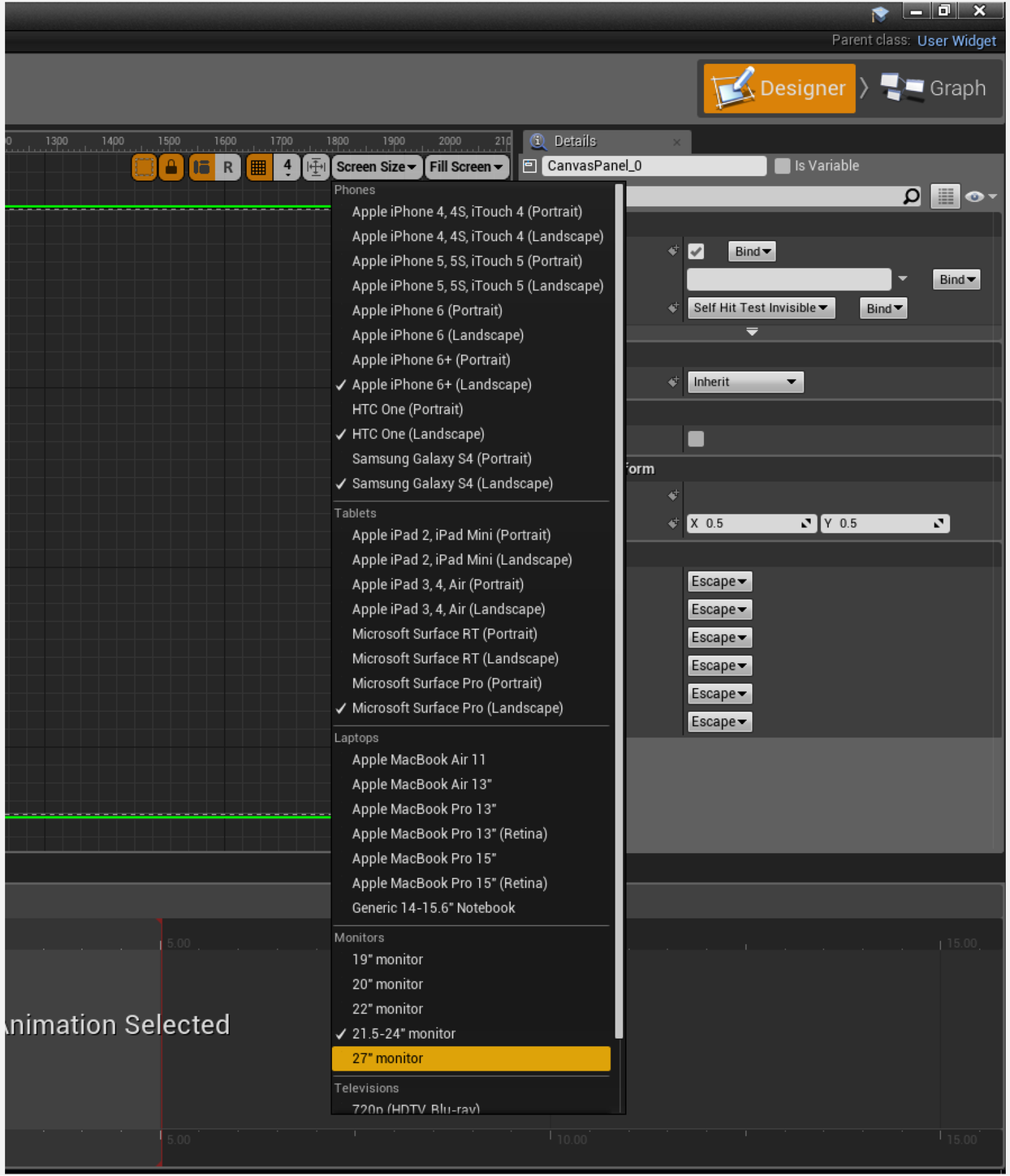
DESIGNER PANEL

In the upper right-hand corner of the panel are display preferences, snap setting, and Screen Size presets.



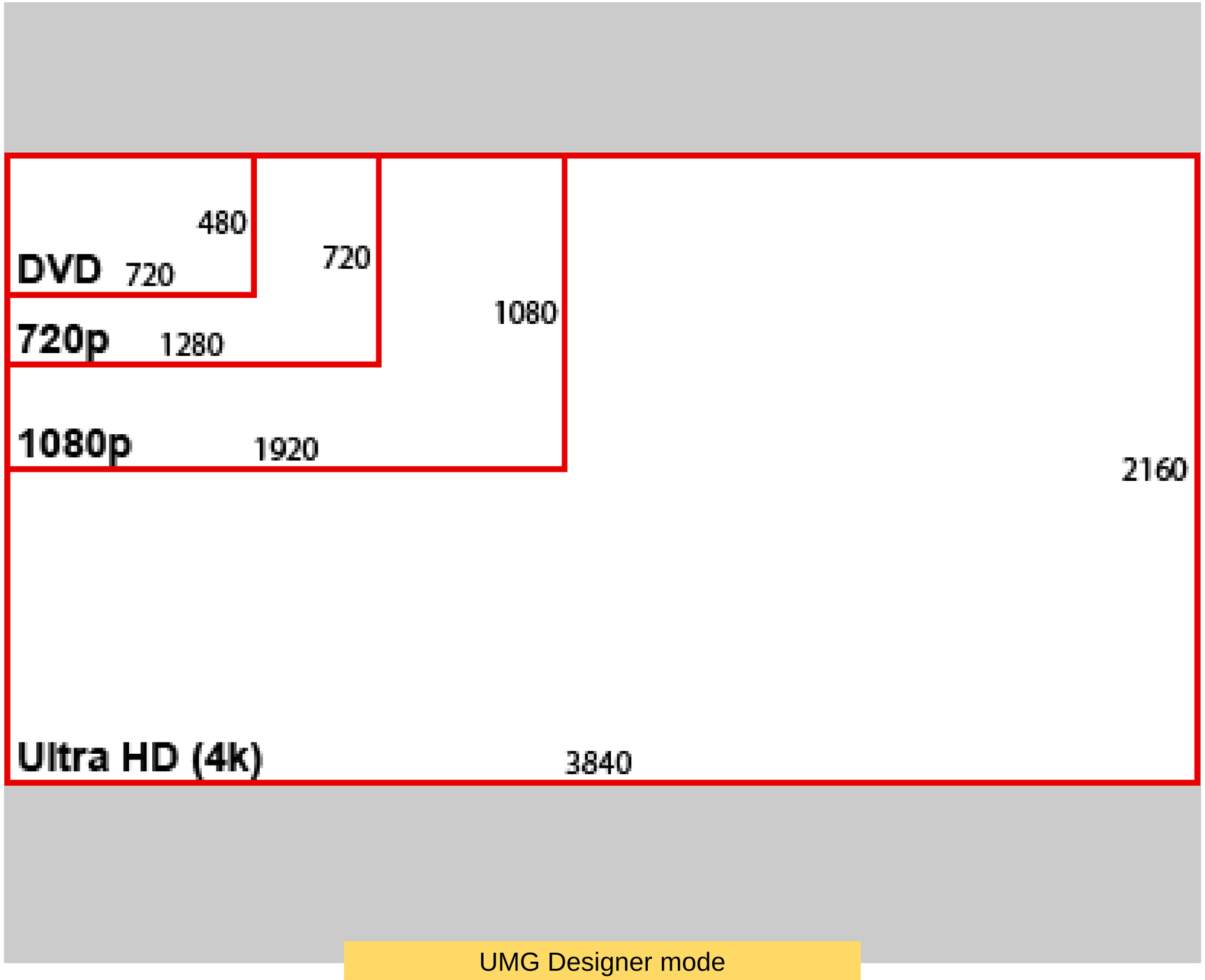
INTERFACE RESOLUTION

When designing interfaces and HUDs, the first thing you need to do is set the aspect ratio and resolution you are designing for.



Setting the screen size is not the same as setting the resolution for the project.

While 3D rendering aspects of a game can scale to whatever screen size the game is being displayed on, HUDs and user interfaces are 2D and are organized based on screen aspect ratios and resolutions. If you are building user interfaces and HUDs for a game that is going to run on different PCs with different monitors or mobile devices, you may have no idea what aspect ratio or screen resolution will be used by your players. UMG uses anchors and DPI scaling to deal with this dilemma.



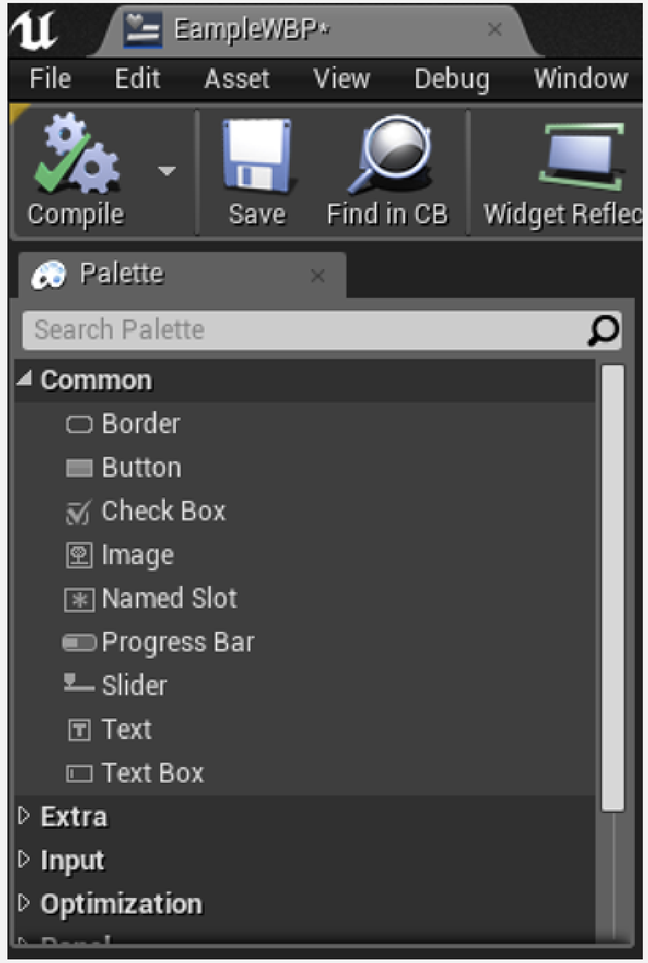
UMG Designer mode

ASPECT RATIOS

Screen aspect ratios refer to the relationship of horizontal (*x*) pixels to vertical (*y*) pixels. For example, if your monitor has a 1920x1080 resolution, its aspect ratio is 16:9. For every 16 pixels across, there are 9 pixels down.

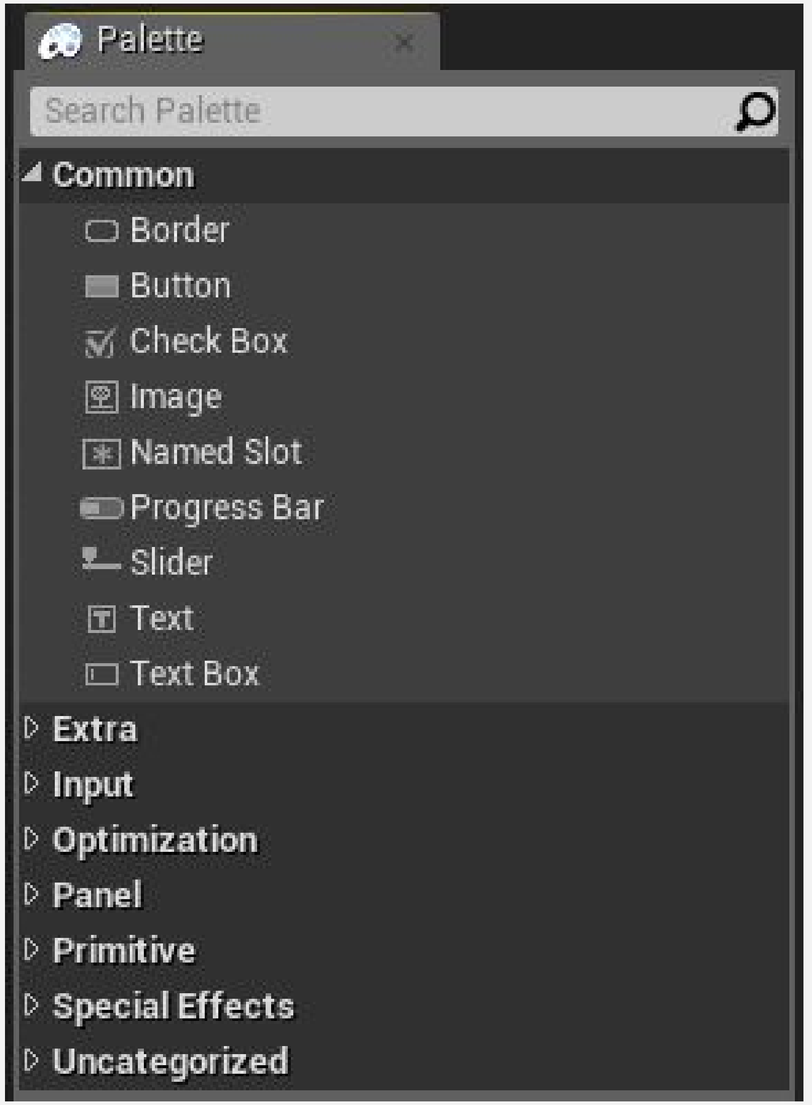
PALETTE PANEL

The Palette panel contains a collection of widgets organized by category. Widgets are interface components with defined functionality that are placed in the Designer panel.



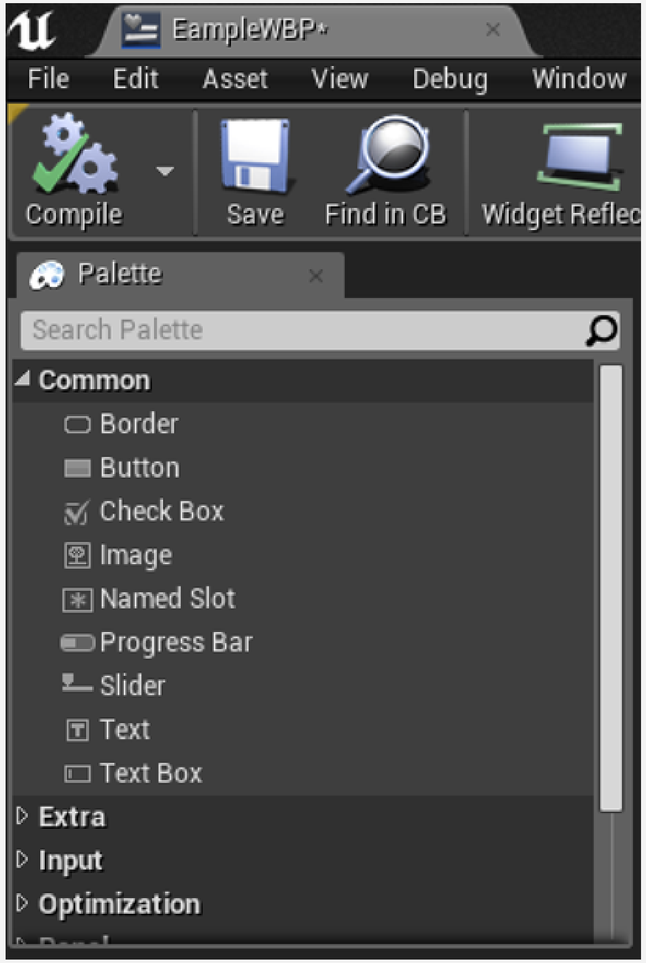
PALETTE PANEL

Each category of widgets contains different widget types that you can drag and drop into the Designer panel. By mixing and matching these widget types, you can lay out the look of your UI on the Designer tab and add the functionality to your widgets through the settings in the Details panel for each widget and via the Graph tab.



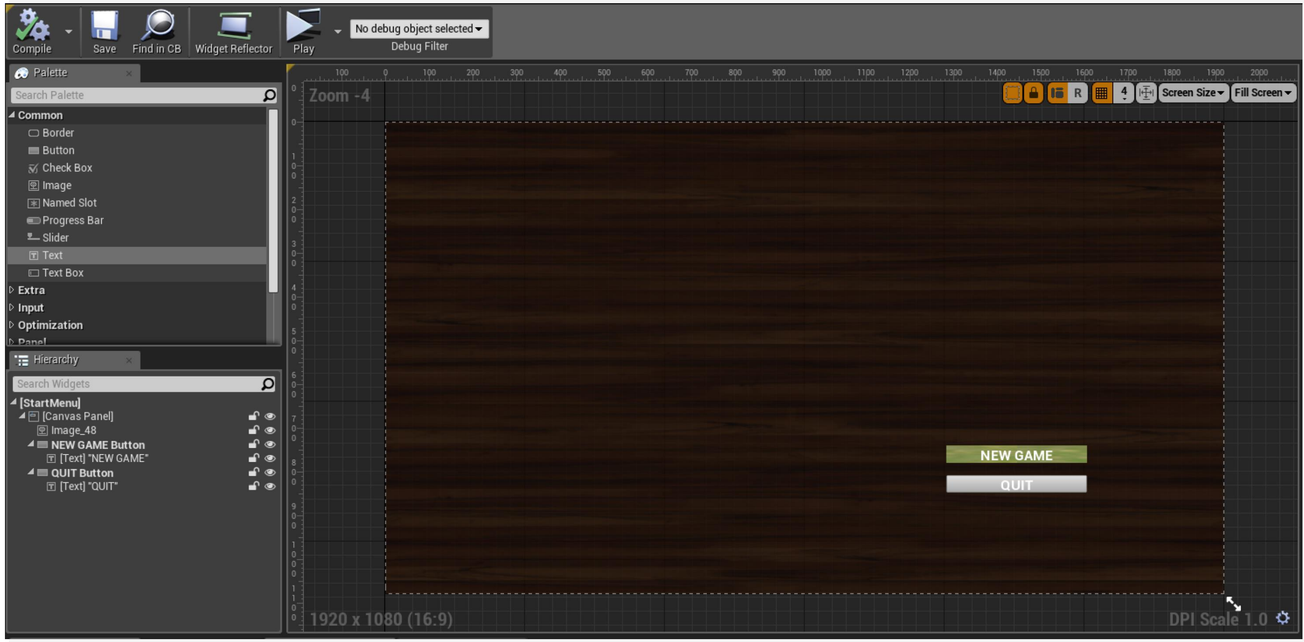
WIDGETS

Widget types range from simple Image widgets used to place images (Textures or Materials) to Text widgets used to place text, and Progress Bar widgets used to display player stats or Level load bars. There are also formatting widgets used to organize other widgets placed in the interface.



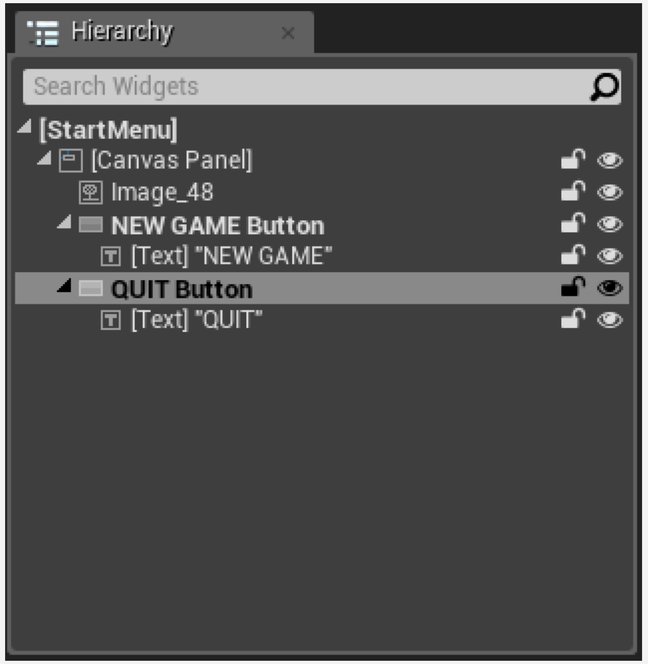
HIERARCHY PANEL

The Hierarchy panel shows all the widgets that have been placed in the Designer panel in list format.



HIERARCHY PANEL

The order in which widgets are added to the layout determines their order in the Hierarchy panel and ultimately the order in which they are displayed.

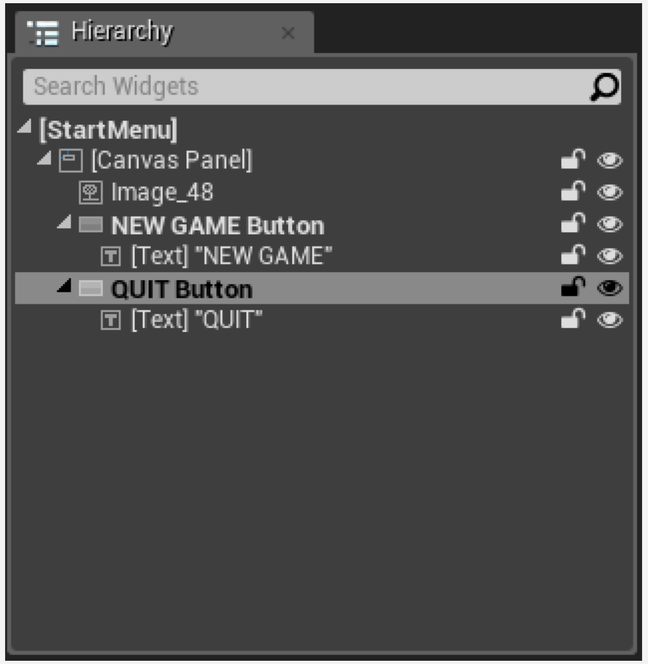


HIERARCHY PANEL

For example, if you use an Image widget for a background image in your interface, you would want it to be at the top of the hierarchy.

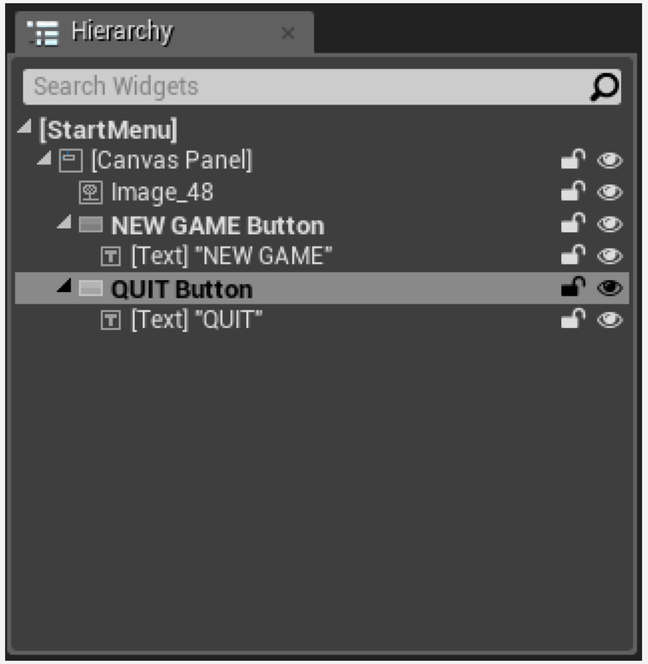
HIERARCHY PANEL

Widgets can be reordered simply by clicking on a widget in the list and dragging it to a new location in the Hierarchy panel.



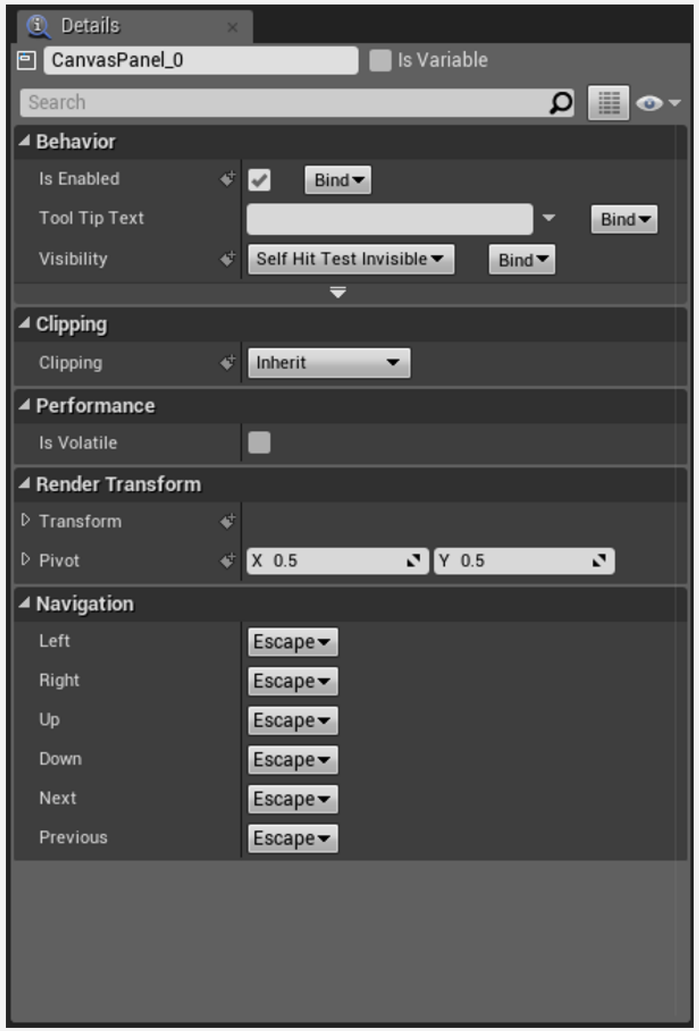
CANVAS PANEL

By default, at the top of the Hierarchy panel is the Canvas Panel widget. It is a designer-friendly panel widget that allows other widgets to be laid out at arbitrary locations, anchored, and z-ordered.



DETAILS PANEL

The UMG Details panel displays the properties of the currently selected widget. While each widget has properties that are unique to its functionality, there are some common properties found on every widget placed, such as the properties found in the Slot tab.



SLOT TAB PROPERTIES

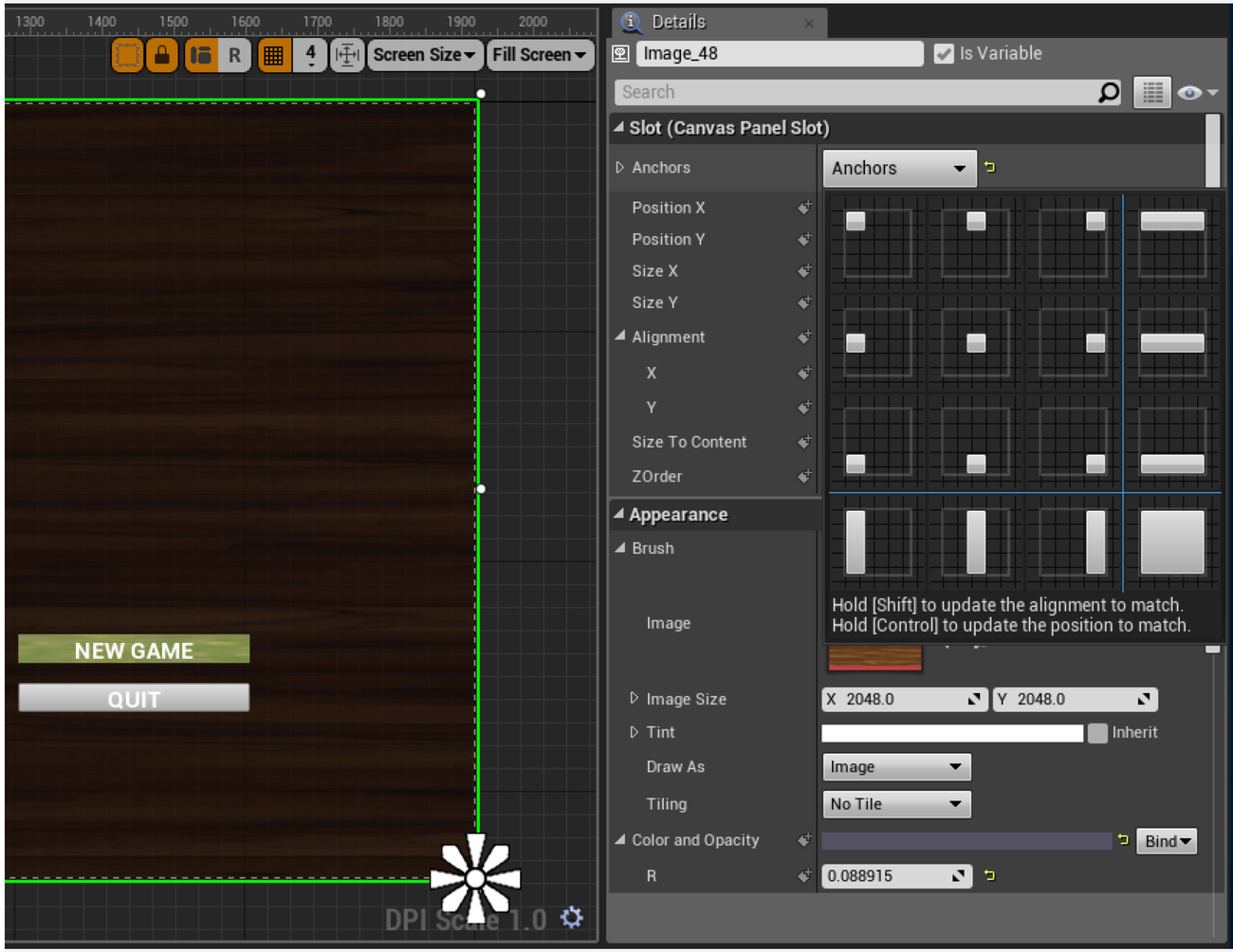
The Slot tab properties contain the position, size, and anchor properties of the selected widget. It also has a ZOrder property that allows you override the draw order defined by a widget’s order in the Hierarchy panel.

ANCHORS

Anchors are used to define a UI widget’s desired location on a Canvas panel and maintain that position with varying screen sizes. Anchors are normalized.

ANCHORS

While a widget’s anchor can be positioned manually, there are common presets that allow you to quickly place them.

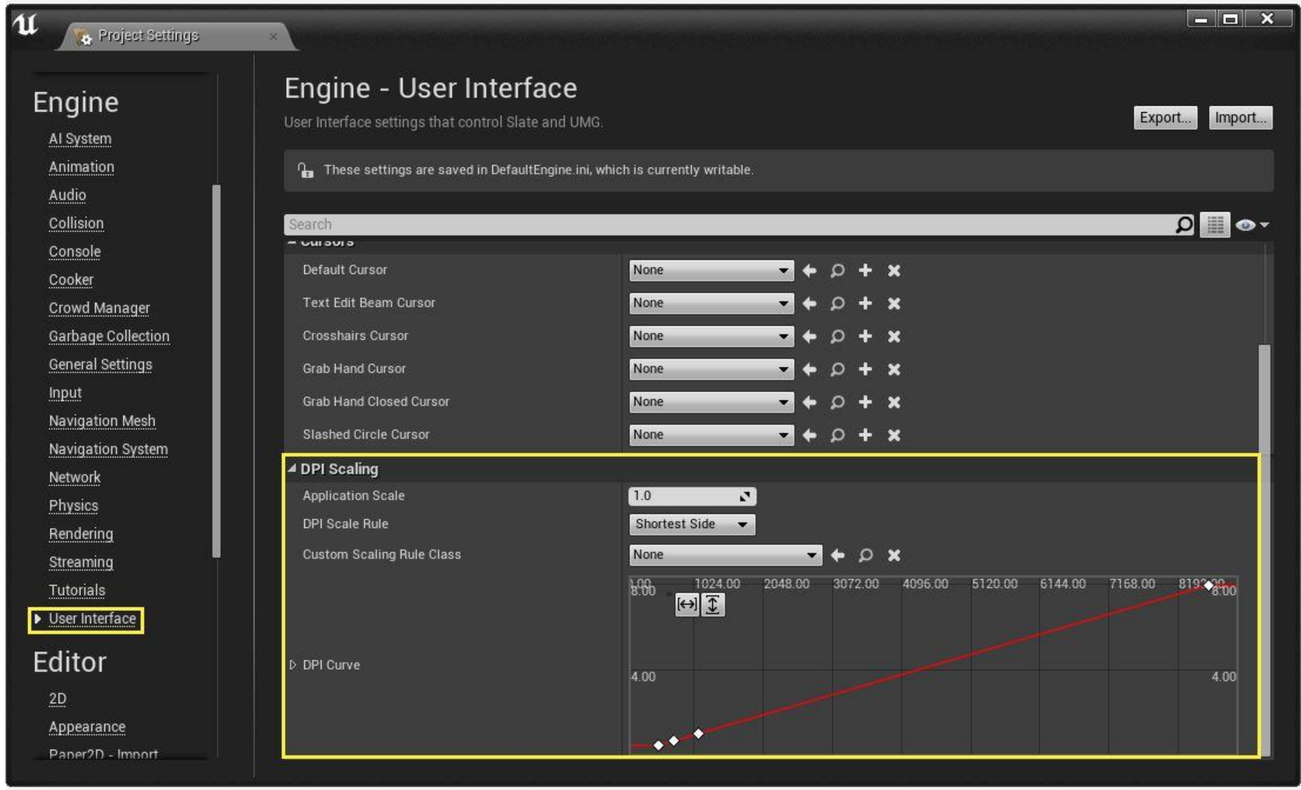


ANCHORS

Because anchors use a normalized value, their position is determined by percentage and not by a set pixel coordinate. This means that if the aspect ratio or resolution of the screen changes, the widget will adjust accordingly and maintain its relative position on screen. Anchors work directly with DPI scaling.

DPI SCALING

DPI scaling rules define how anchor points should respond if the resolution or aspect ratio changes. For example, if one player has a monitor that has a 4:6 aspect ratio with a resolution of 1024x768 and another player has a monitor that has a 16x9 aspect ratio with a 4K resolution, both need to see the game interface.



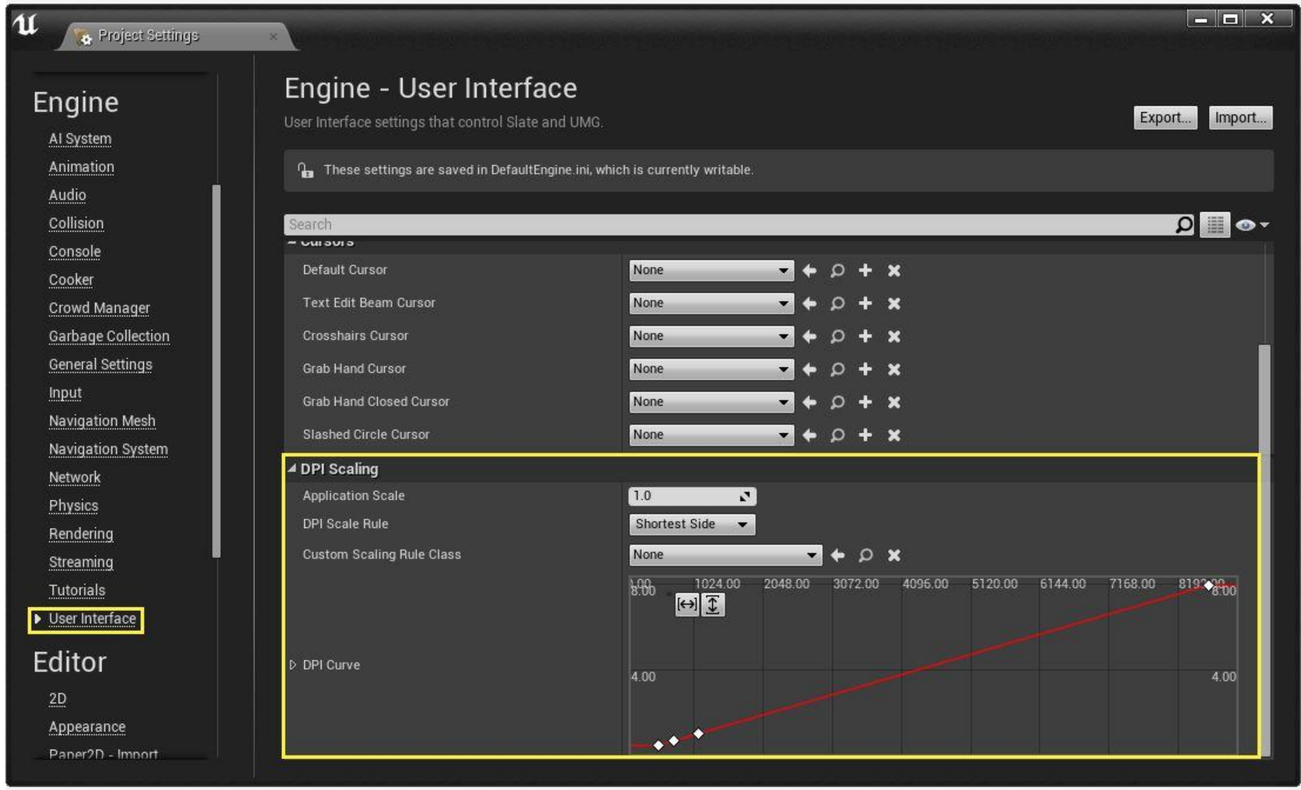
DPI scaling rules define how anchor points should respond if the resolution or aspect ratio changes.

If you did not have DPI scaling rules, you would need to manually create an interface layout for every possible combination of monitor aspect ratio and resolution. This would take a lot of development time as well as increase the project file’s size.

A good rule of thumb is to design your interfaces for the highest resolution and most common aspect. Remember, raster images scale down better than they scale up.

DPI SCALING

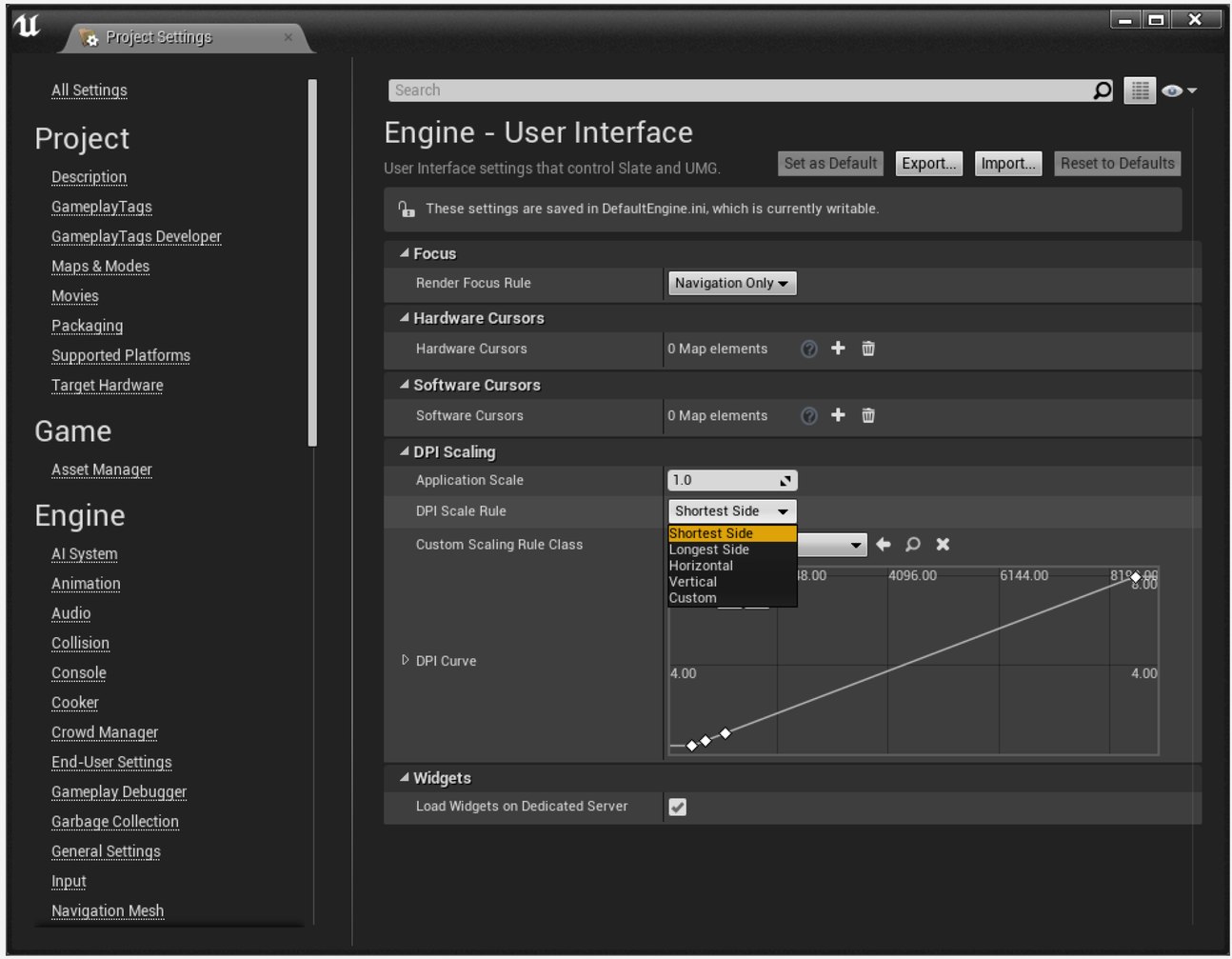
Default DPI scaling rules are applied to every project and can be configured as you see fit in the User Interface section of the Project Settings panel.



DPI SCALING

The DPI Scale Rule property contains the following settings:

- **Shortest Side**: Evaluates the scale curve based on the shortest side of the Viewport (most common setting)
- **Longest Side**: Evaluates the scale curve based on the longest side of the Viewport
- **Horizontal**: Evaluates the scale curve based on the *x* axis of the Viewport
- **Vertical**: Evaluates the scale curve based on the *y* axis of the Viewport



COMMON WIDGETS

Image, Button, Text, Progress Bar

COMMON WIDGETS

In the Palette panel, widgets are organized in categories based on functionality. There are too many to cover here, so we will just focus on the ones needed to make a basic interface.

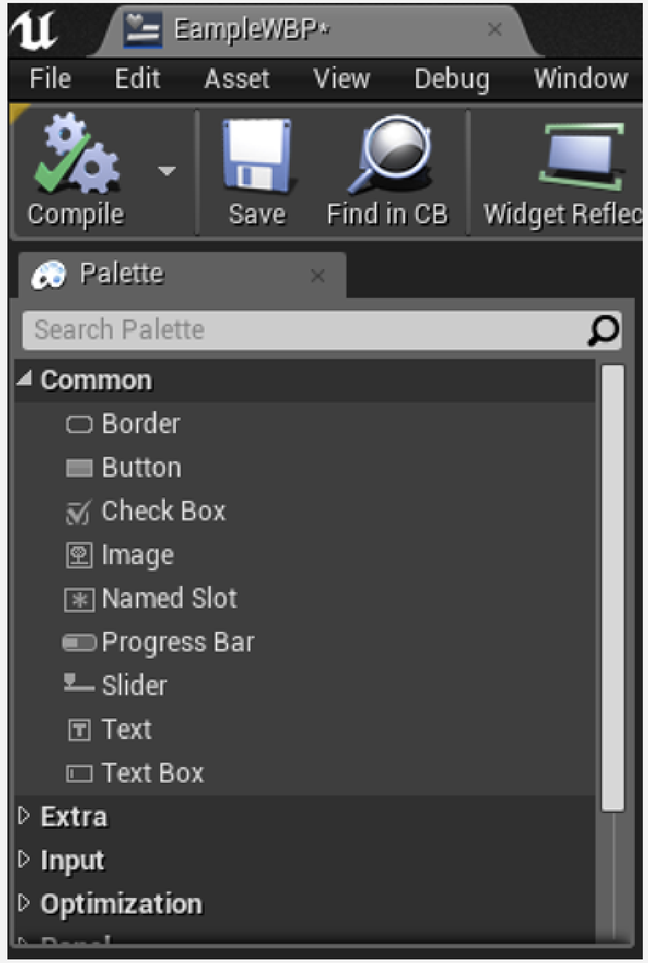


IMAGE WIDGET

The Image widget allows you to display a Slate Brush, Texture, Sprite, or Material in the UI.

To add an image, simply drag a Texture from the Content Browser onto the thumbnail of the Image widget’s property or click on the drop-down and select it from the list.

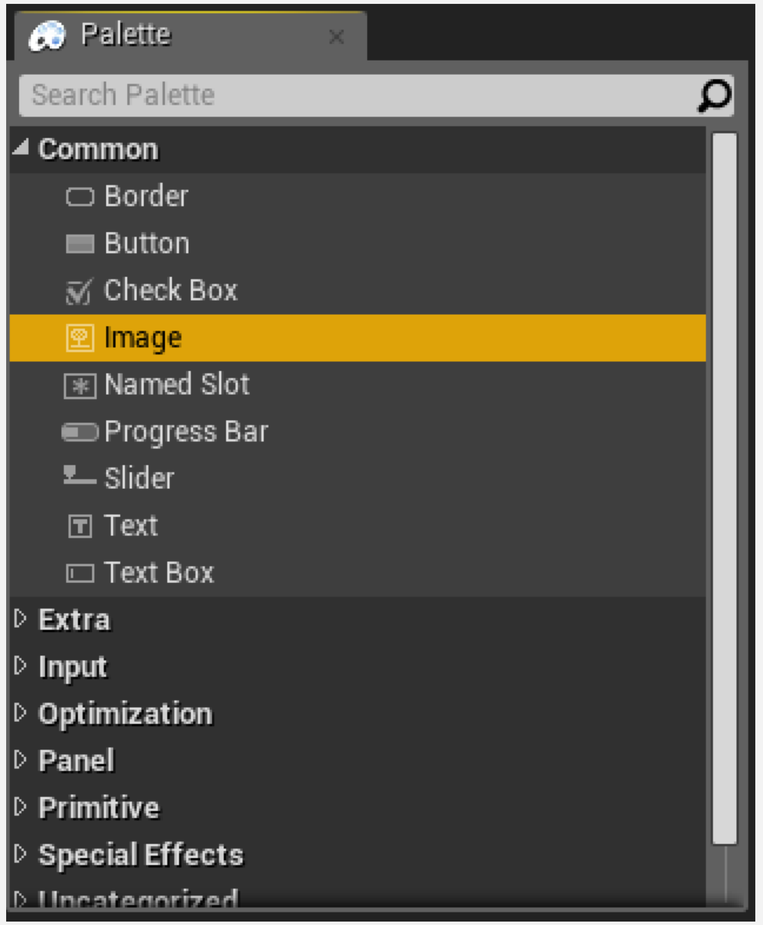
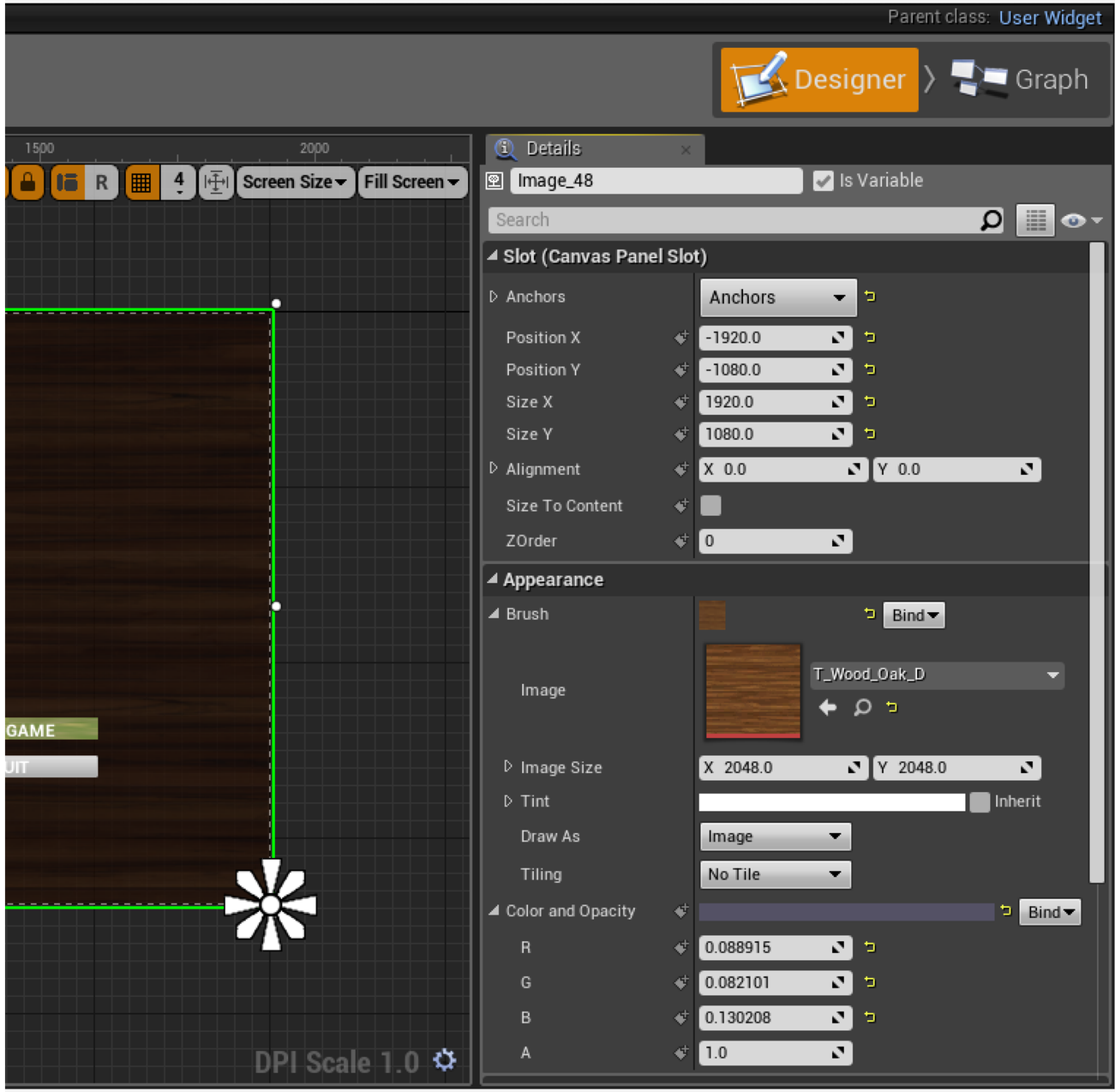


IMAGE WIDGET

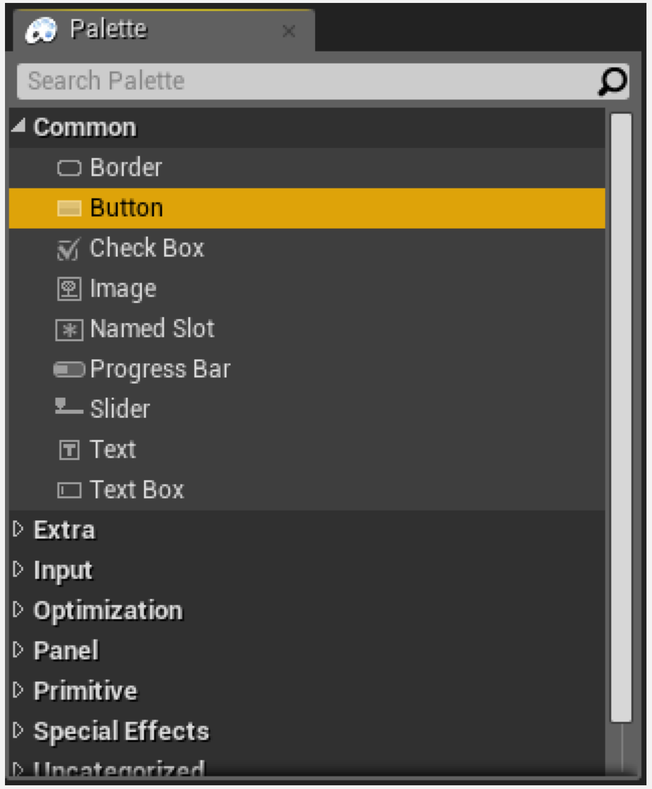
While the Texture you use already has its own size, you can adjust the Image widget's size manually in the Viewport or change it in the Slot category.

In the example on the right, the image is 2048x2048, but the widget is set to 1920x1080.



BUTTON WIDGET

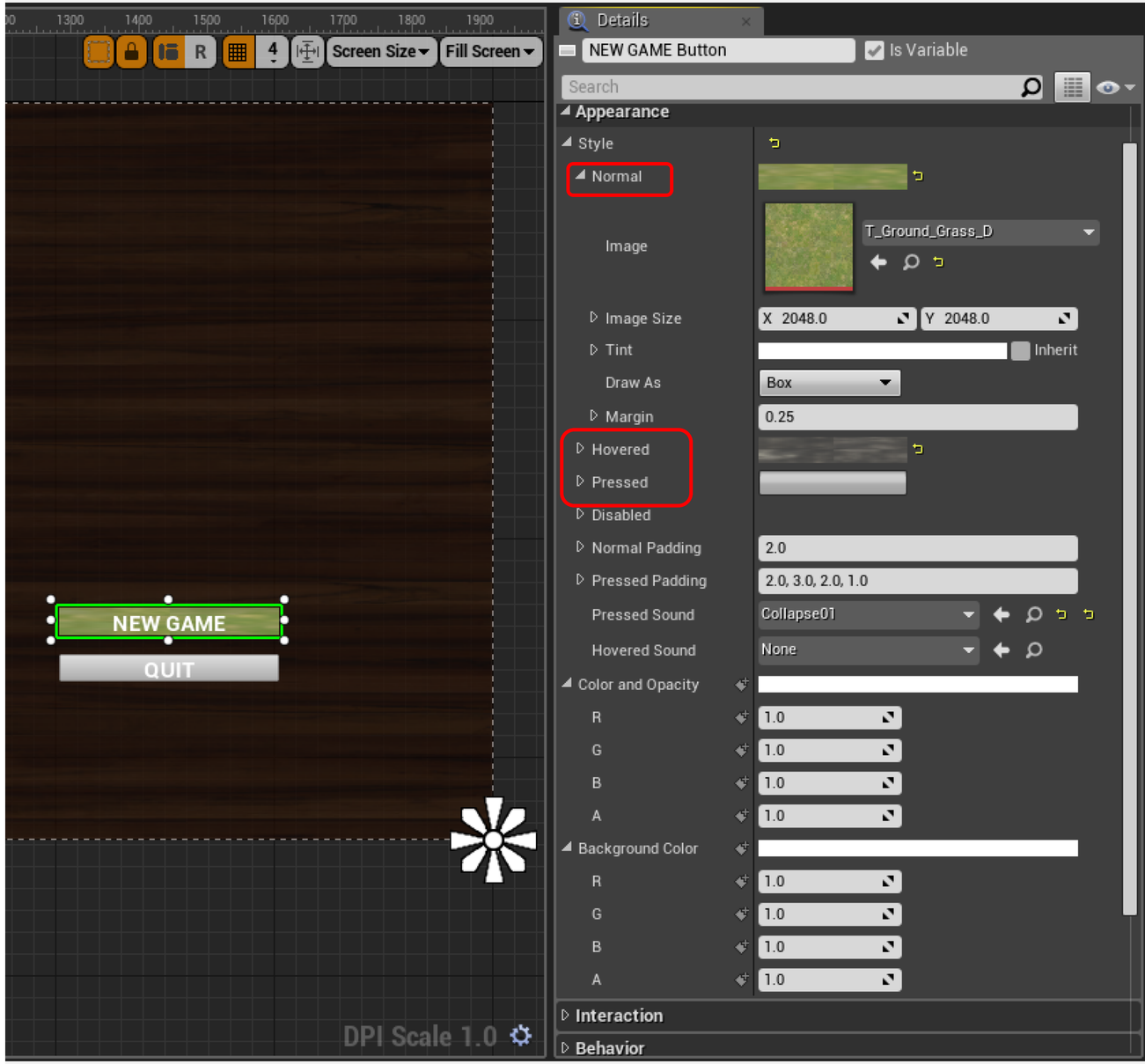
The Button widget is a clickable primitive widget that enables basic interaction, meaning it is already set up to respond to basic mouse interactions.



BUTTON WIDGET

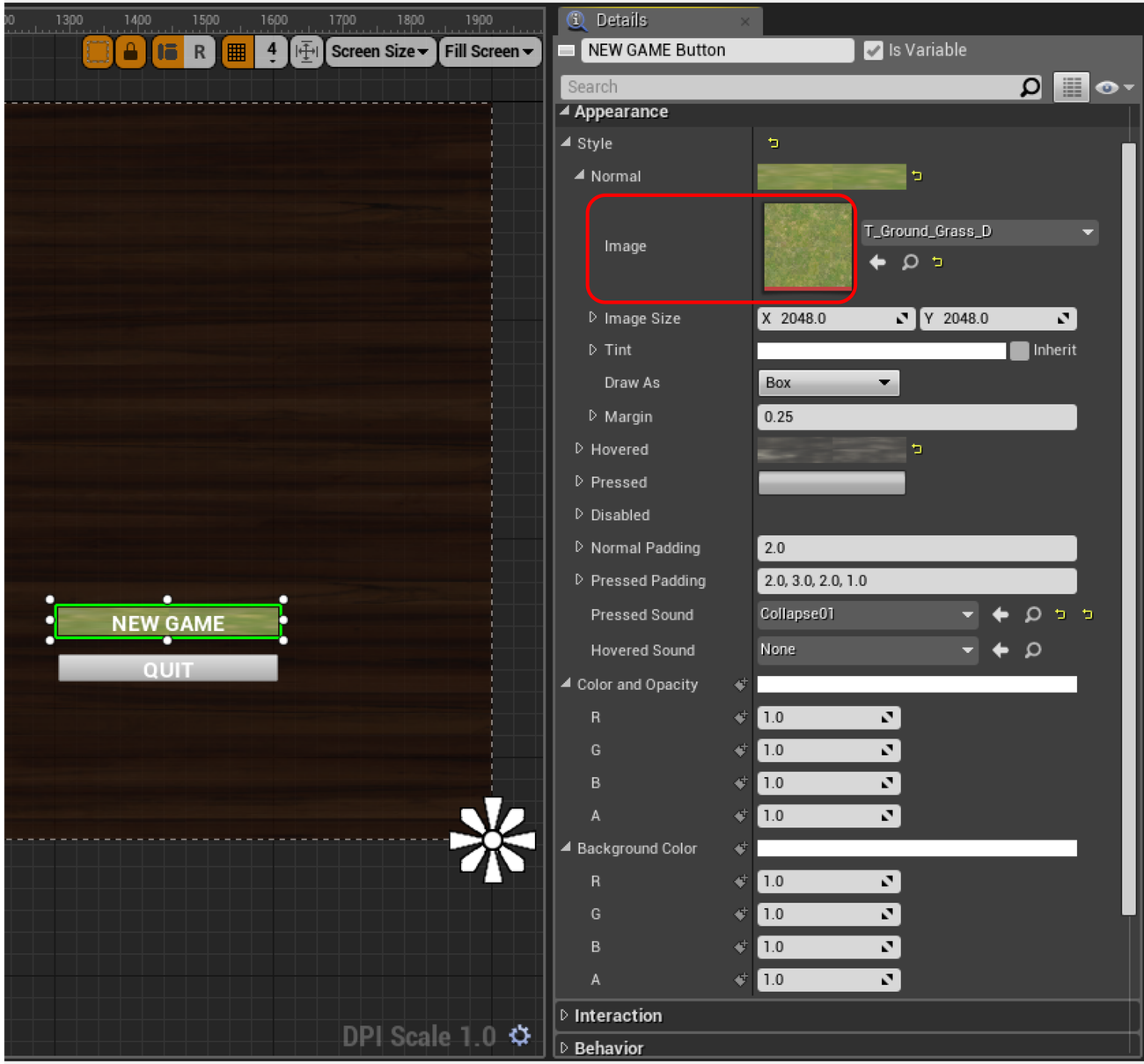
There are three interaction states found under Style in the Appearance category in the Details panel:

- **Normal**: The default state and displayed when the button is not being interacted with
- **Hovered**: Displayed when the mouse cursor is over the button
- **Pressed**: Displayed when the mouse is clicked on the button



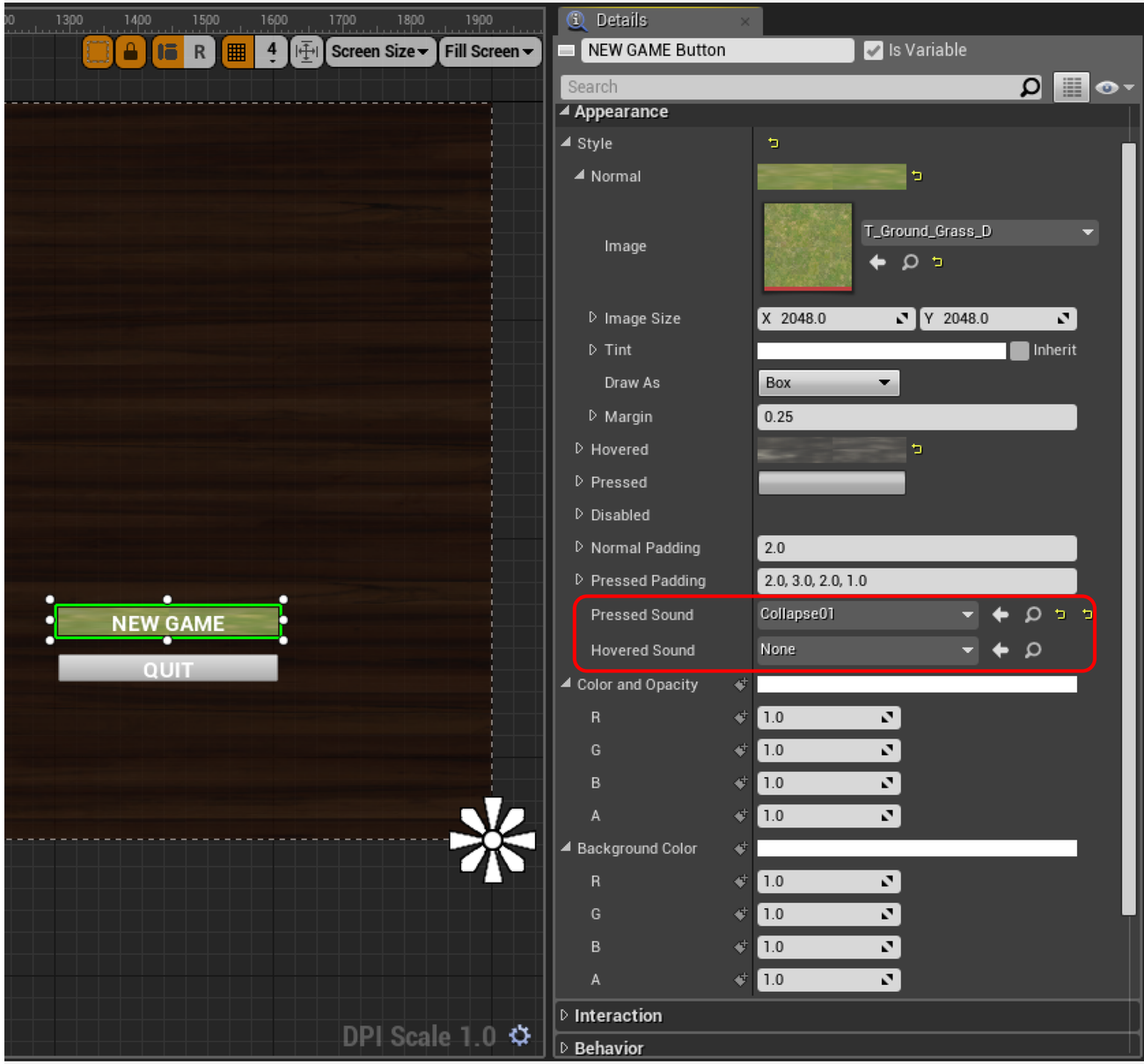
BUTTON WIDGET

To assign a Texture or Material to any of the button’s states, simply drag the asset from the Content Browser onto the state’s Image property thumbnail.



BUTTON WIDGET

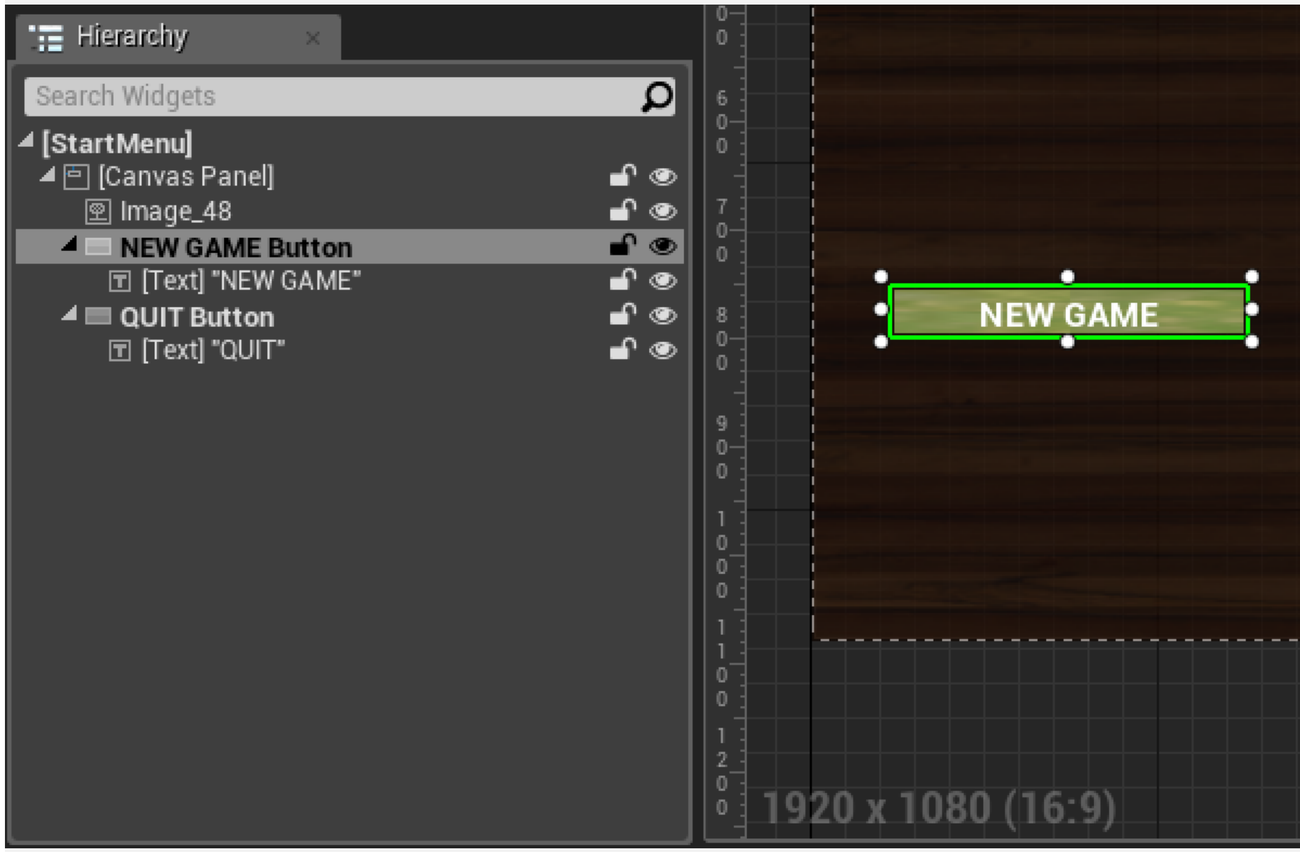
The Button widget also allows you to assign Sound Wave or Sound Cue assets to play when the pressed or hovered states of the Button are activated.



BUTTON WIDGET

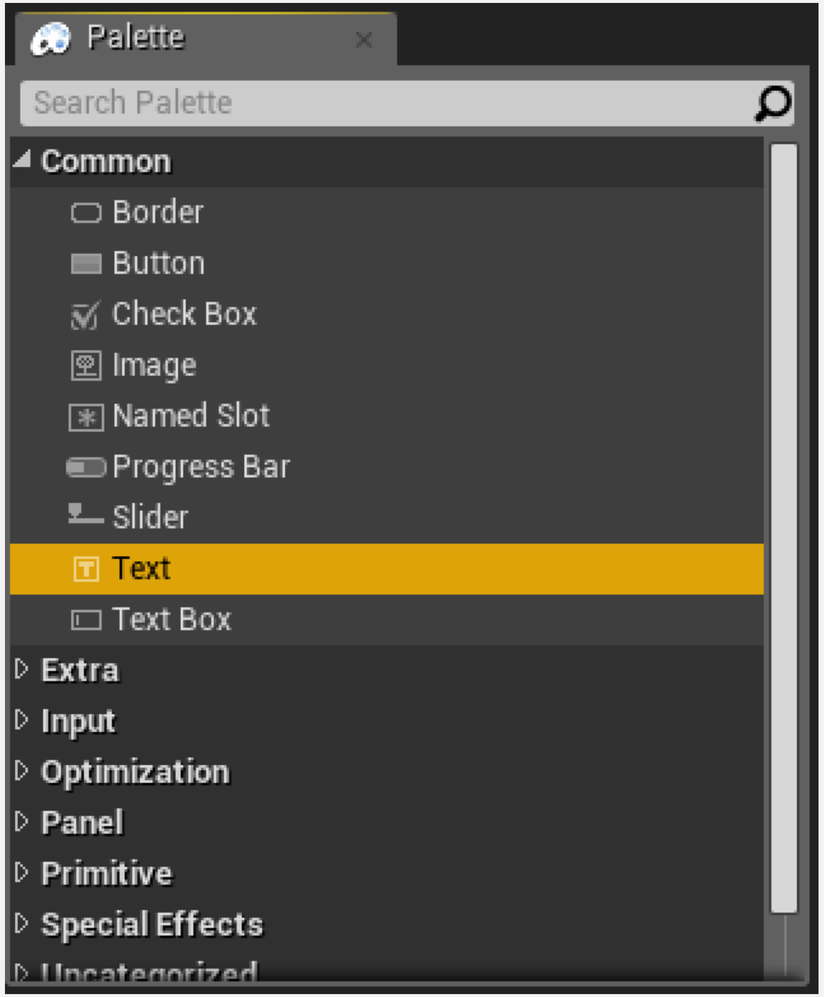
You can place any other widget inside a Button widget to make a more complex and interesting clickable element in your UI.

For example, if you drag a Text widget onto the Button, it will snap to the Button and show up as a child of the Button widget in the Hierarchy panel.



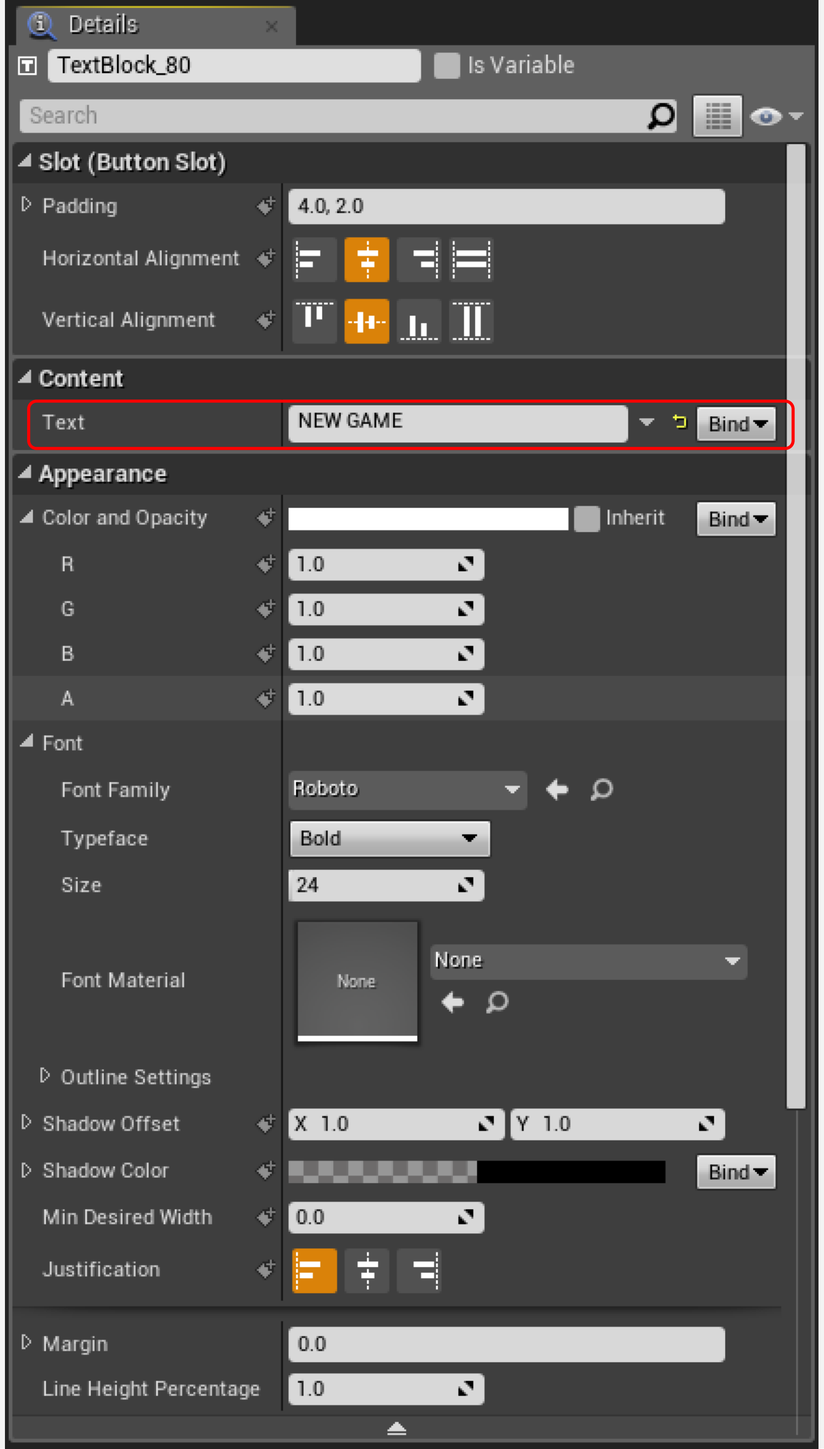
TEXT WIDGET

The Text widget is the basic mechanism for displaying text on screen and can be used for text descriptions of options or other UI elements.



TEXT WIDGET

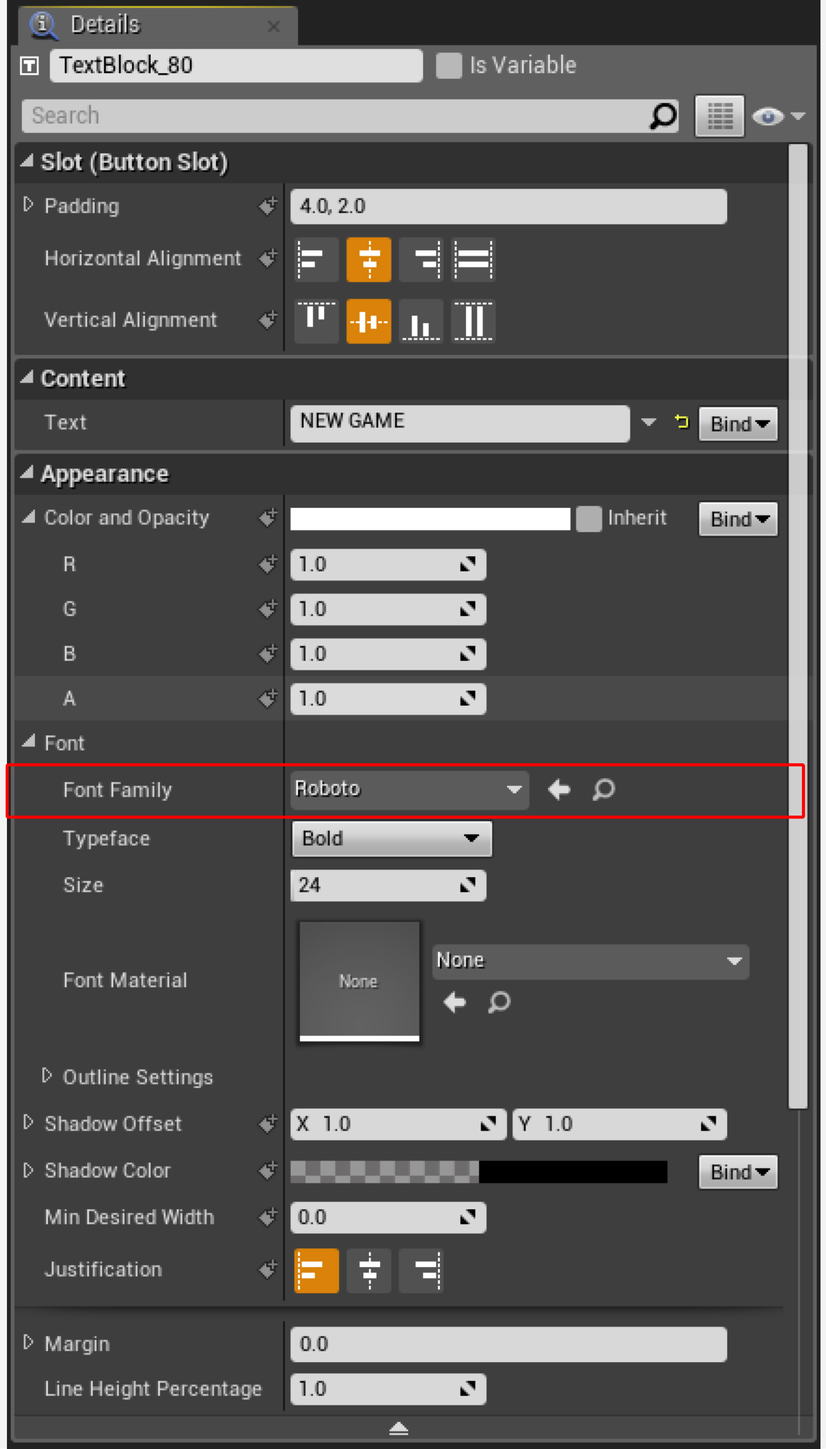
The text that the Text widget displays on screen is set using the Text property under the Content category in the Details panel.



TEXT WIDGET

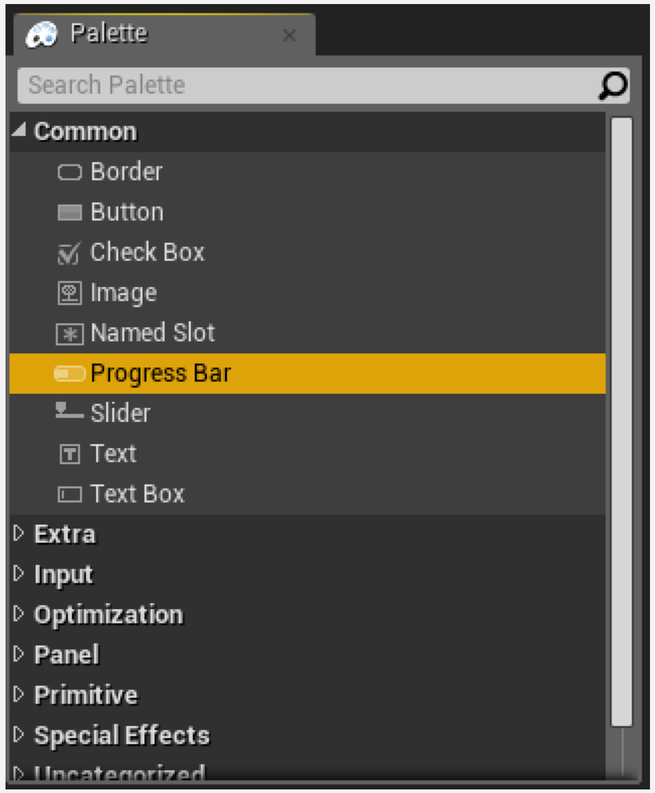
The Text formatting properties are found in the Appearance category.

To change the font used, simply import a font TTY file into the Content Browser. Then click and drag the Font asset from the Content Browser onto the Font Family category in the Appearance category.



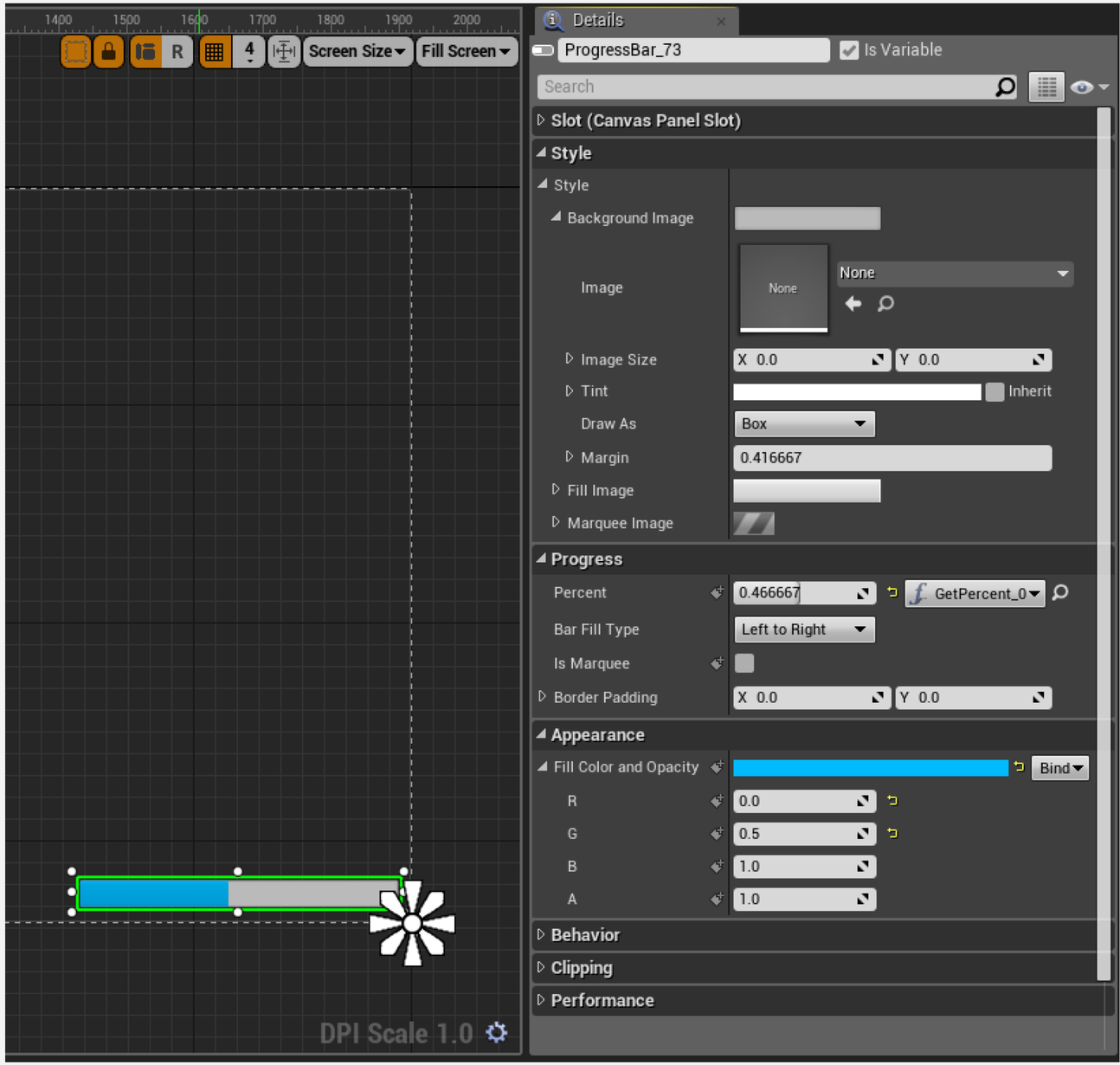
PROGRESS BAR WIDGET

The Progress Bar widget is a simple bar that fills up and can be restyled to fit any number of uses, such as displaying experience, health, points, and Level load times.



PROGRESS BAR WIDGET

The Progress Bar has both Style and Appearance settings, but the fill amount is determined by the Percent property. The value is also normalized from 0 to 1.

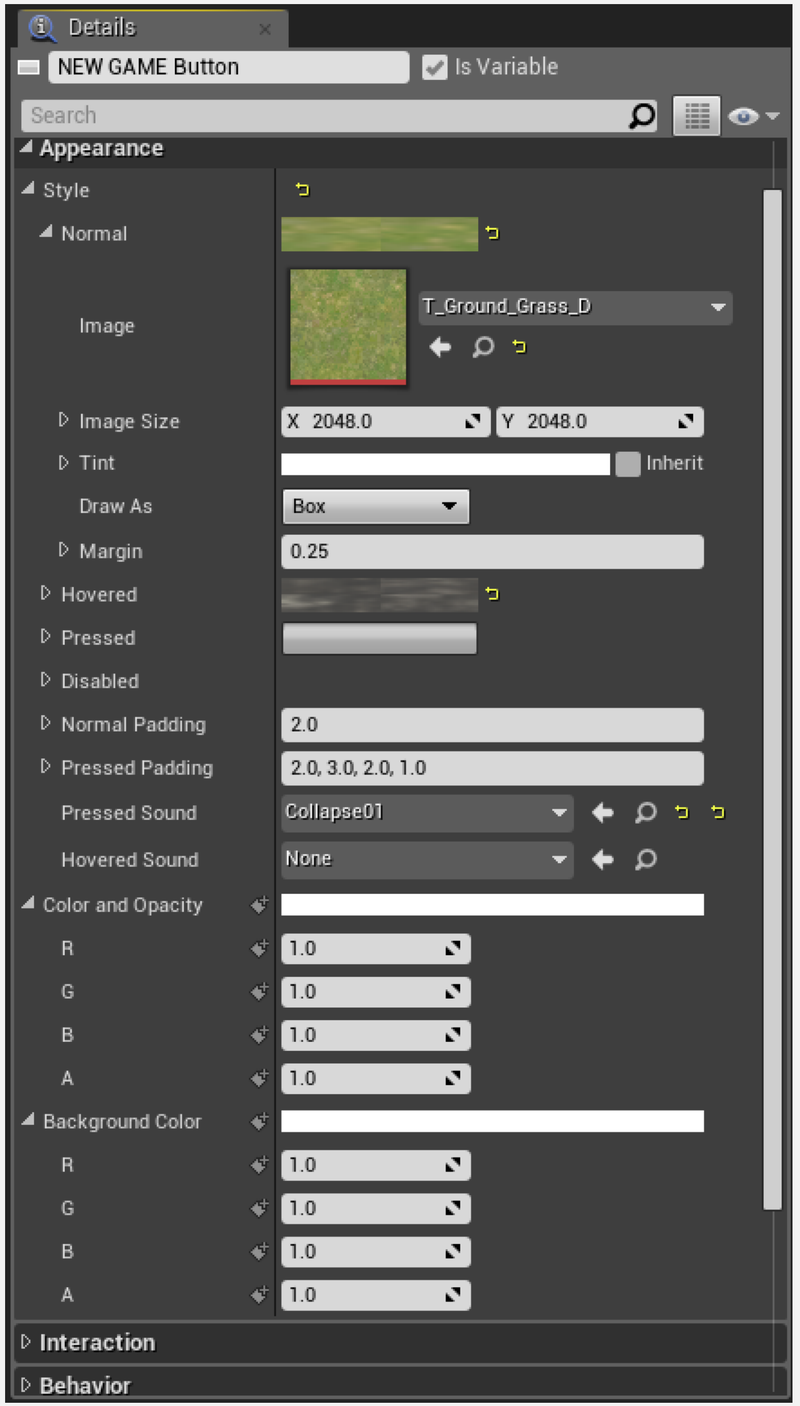


TEXTURES

Prepping Textures for Interfaces

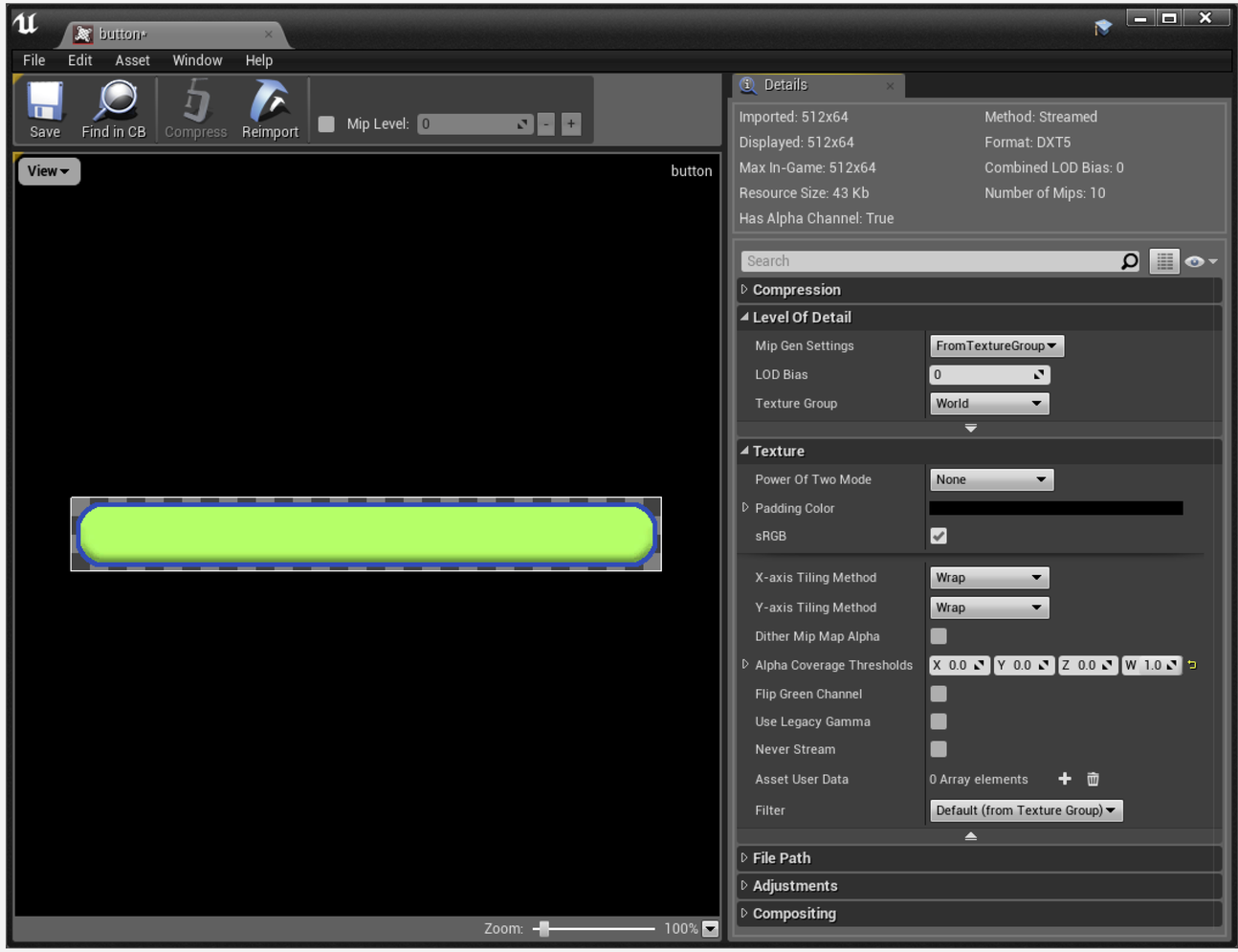
TEXTURES

Widgets such as the Button and the Image widget can use Textures or Materials for style settings. When using Textures, there are some basic settings to change.



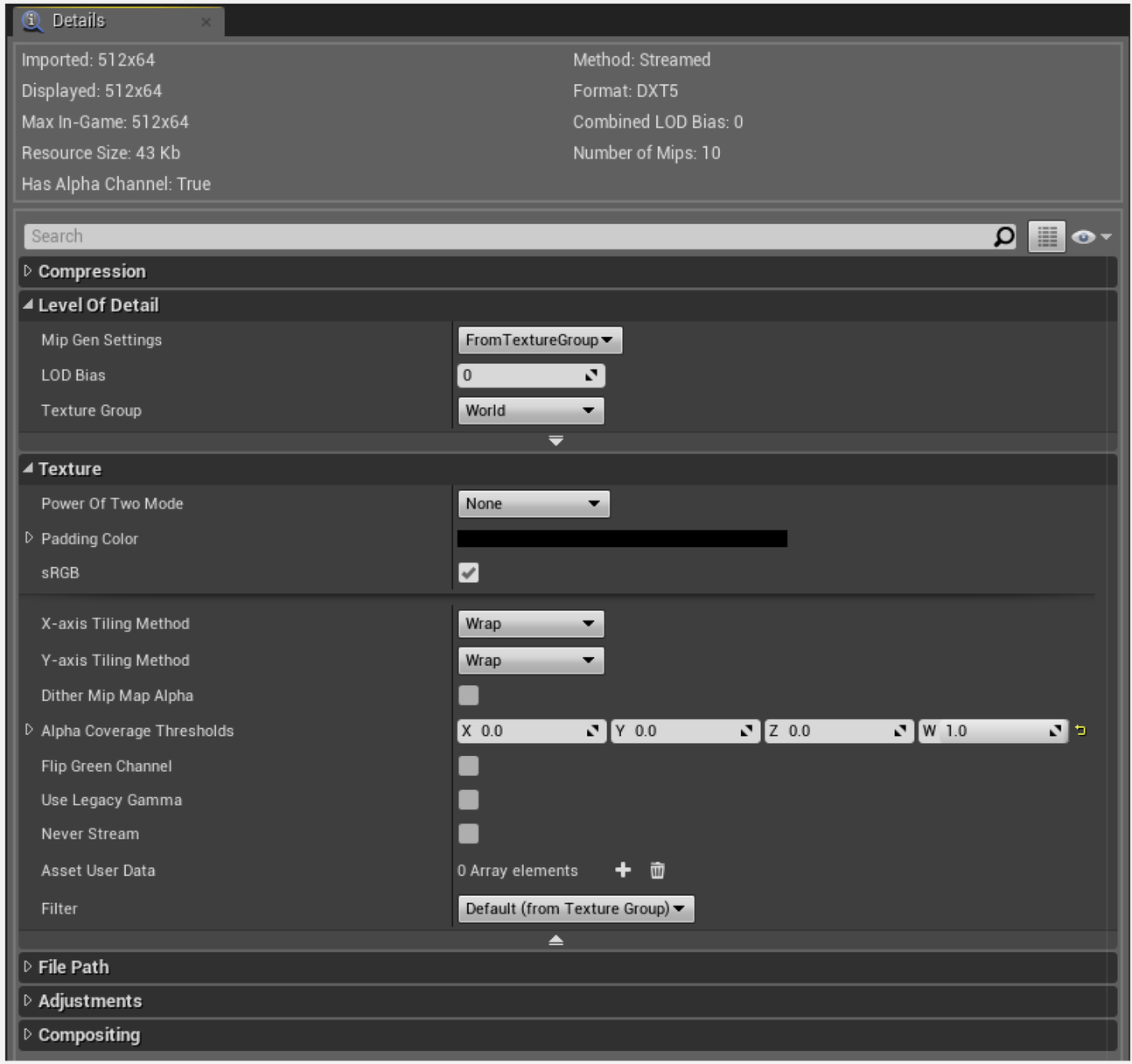
TEXTURES

Unlike Textures used on Materials assigned to Static Meshes, Textures used for user interfaces do not need to have mipmaps, don’t have to be powers of 2, and don’t need to be able to stream.



TEXTURES

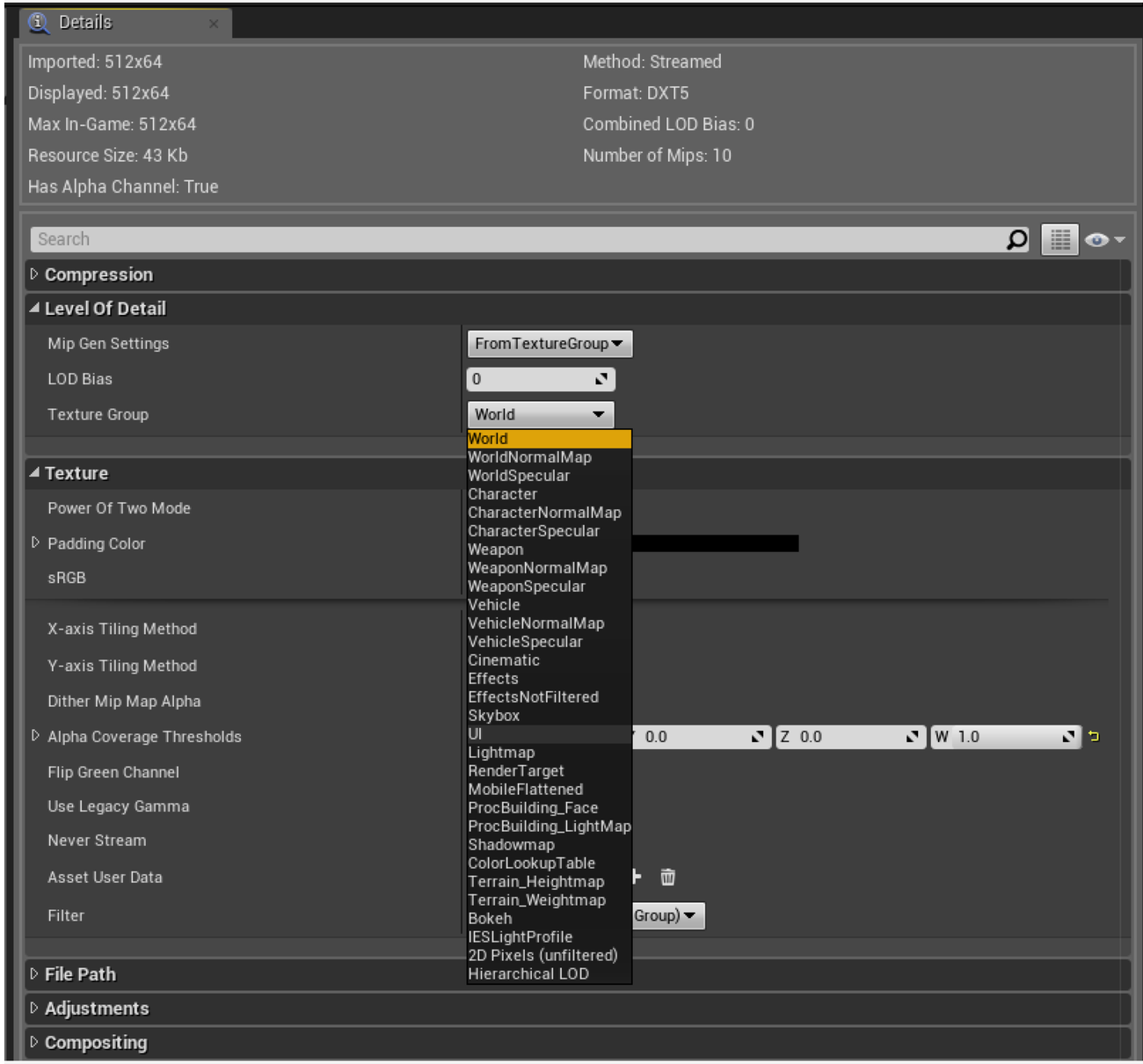
To change the properties of a Texture after it has been imported, double-click on the Texture asset in the Content Browser to open the Texture Editor.



TEXTURES

By default, the Mip Gen settings are determined by the Texture Group assignment.

To change the Texture Group assignment, click on the drop-down next to Texture Group in the Level of Detail tab and select UI from the list.



TEXTURES

Under the Texture category, turn on Never Stream. Now the Texture is ready for use in UMG.