



UNREAL
ENGINE

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).

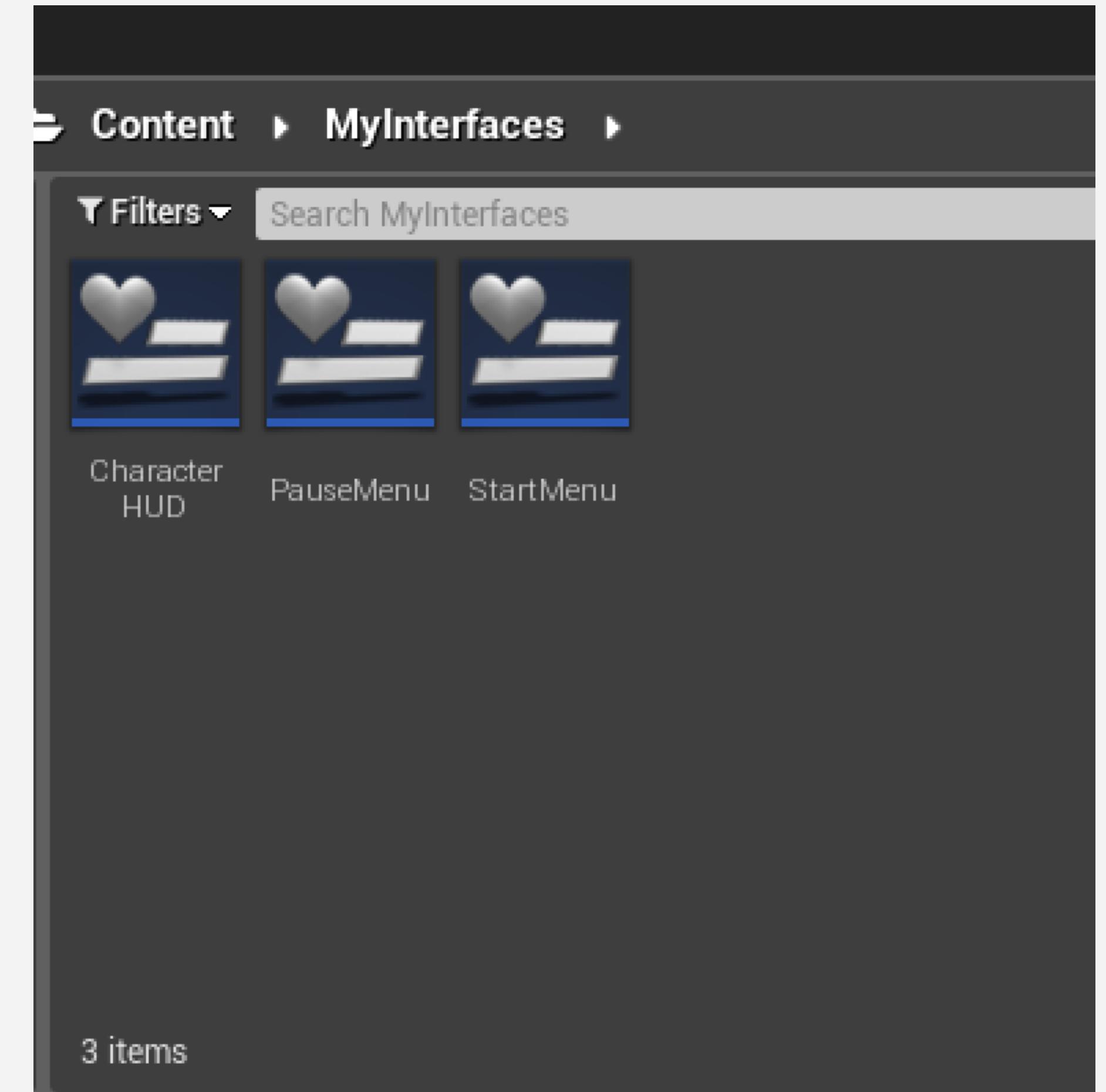
Slate is programmer-friendly but not artist- and designer-friendly. Epic has created the UMG Editor, which combines artist-friendly tools for layout with the power of Blueprint.

The screenshot shows the Unreal Engine Blueprint Editor interface for the 'ThirdPersonGameMode' class. The top menu bar includes File, Edit, Asset, View, Debug, Window, Help, and a note stating 'Parent class: Game Mode Base'. The toolbar features icons for Compile (green checkmark), Save, and Find in CB. A 'Class Defaults' tab is open, containing a note: 'NOTE: This is a data only blueprint, so only the default values are shown. It does not have any script or variables. If you want to add...' and a link to 'Open Full Blueprint Editor'. Below this is a search bar and a tree view with sections for Actor Tick, Classes, Game, and Game Mode. The Actor Tick section shows 'Start with Tick Enabled' checked, 'Tick Interval (secs)' set to 0.0, and 'Allow Tick Before Begin Play' checked. The Classes section lists various game classes with their respective dropdown menus and edit buttons. The Game section contains a 'Default Player Name' input field. The Game Mode section includes settings for 'Use Seamless Travel', 'Start Players as Spectators', and 'Pauseable' (which is checked). At the bottom, there is a 'Public View' button.



WIDGET BLUEPRINT ASSET

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

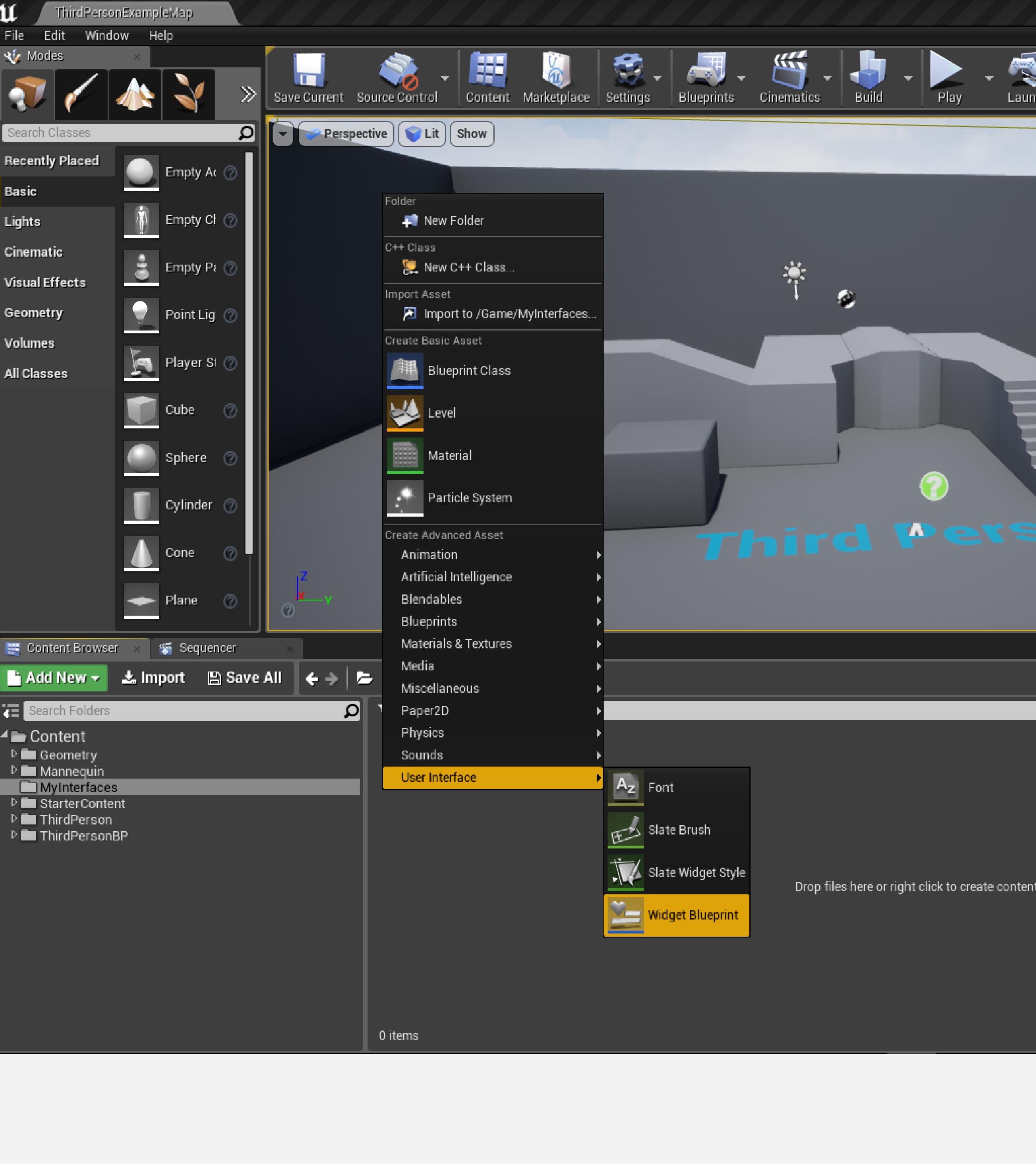




CREATING A WIDGET BLUEPRINT

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

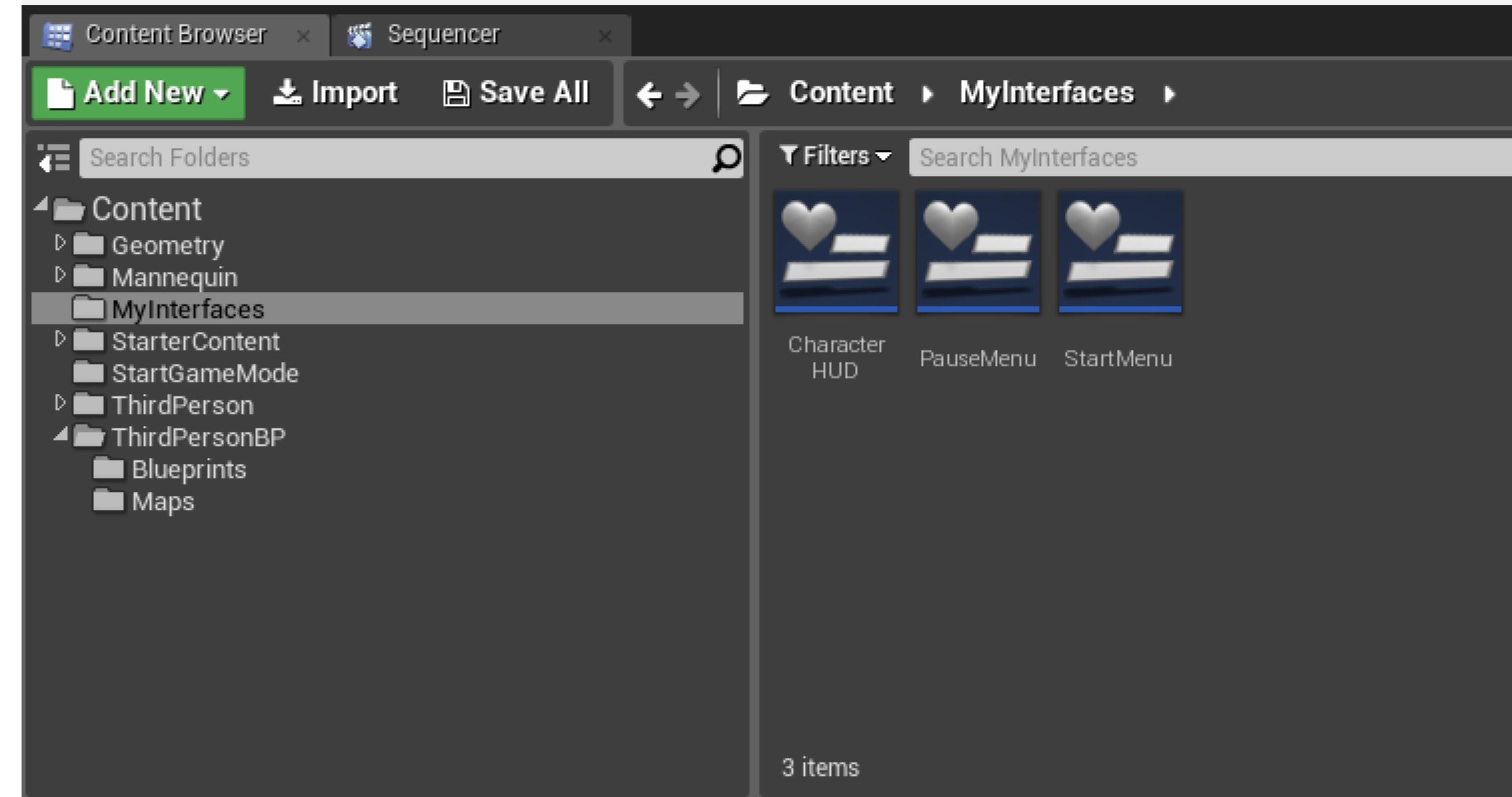
Then right-click in the Content Browser, navigate to User Interface under Create Advanced Asset, and select Widget Blueprint.

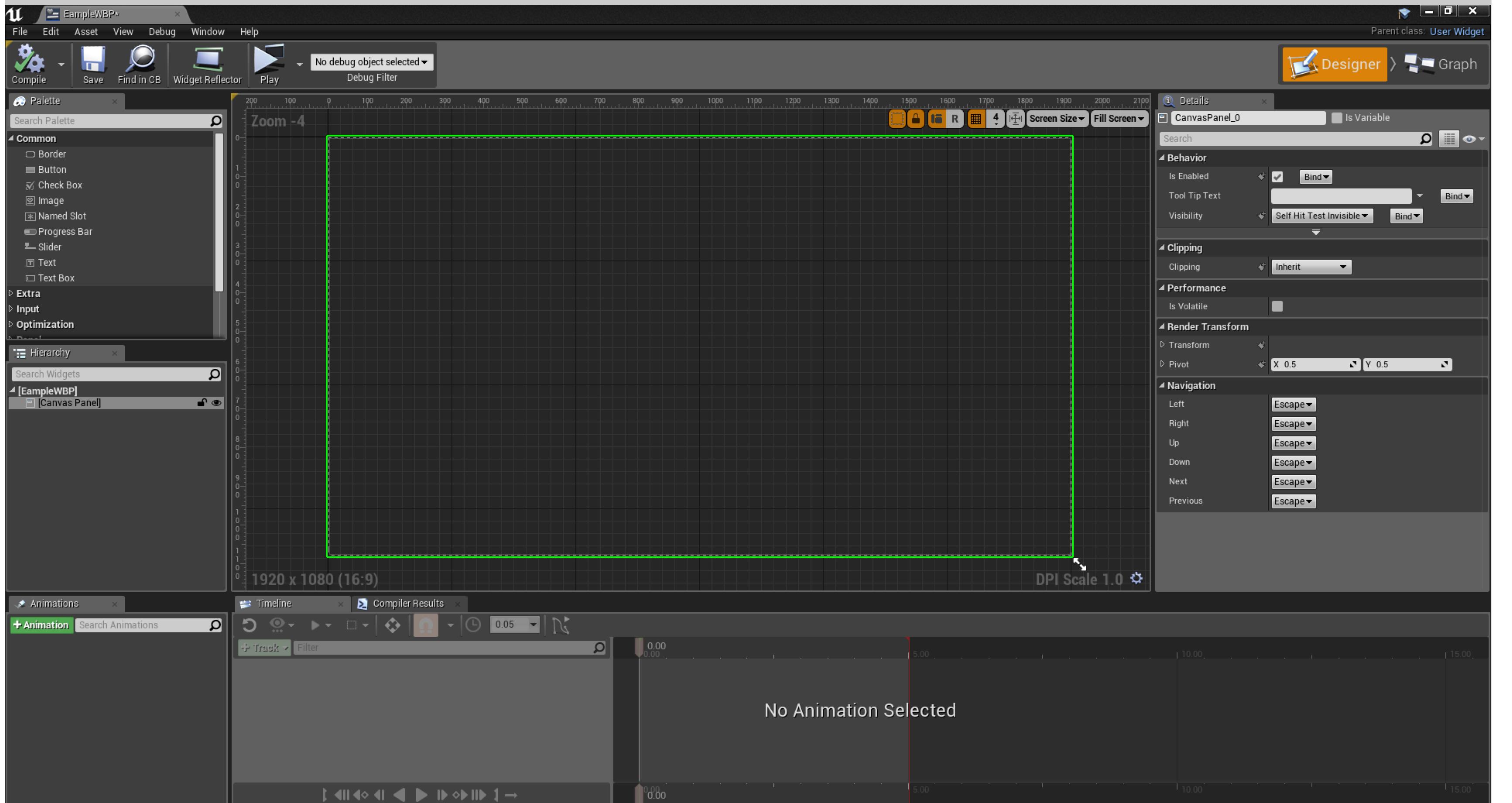




CREATING A WIDGET BLUEPRINT

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





UMG Editor Designer mode

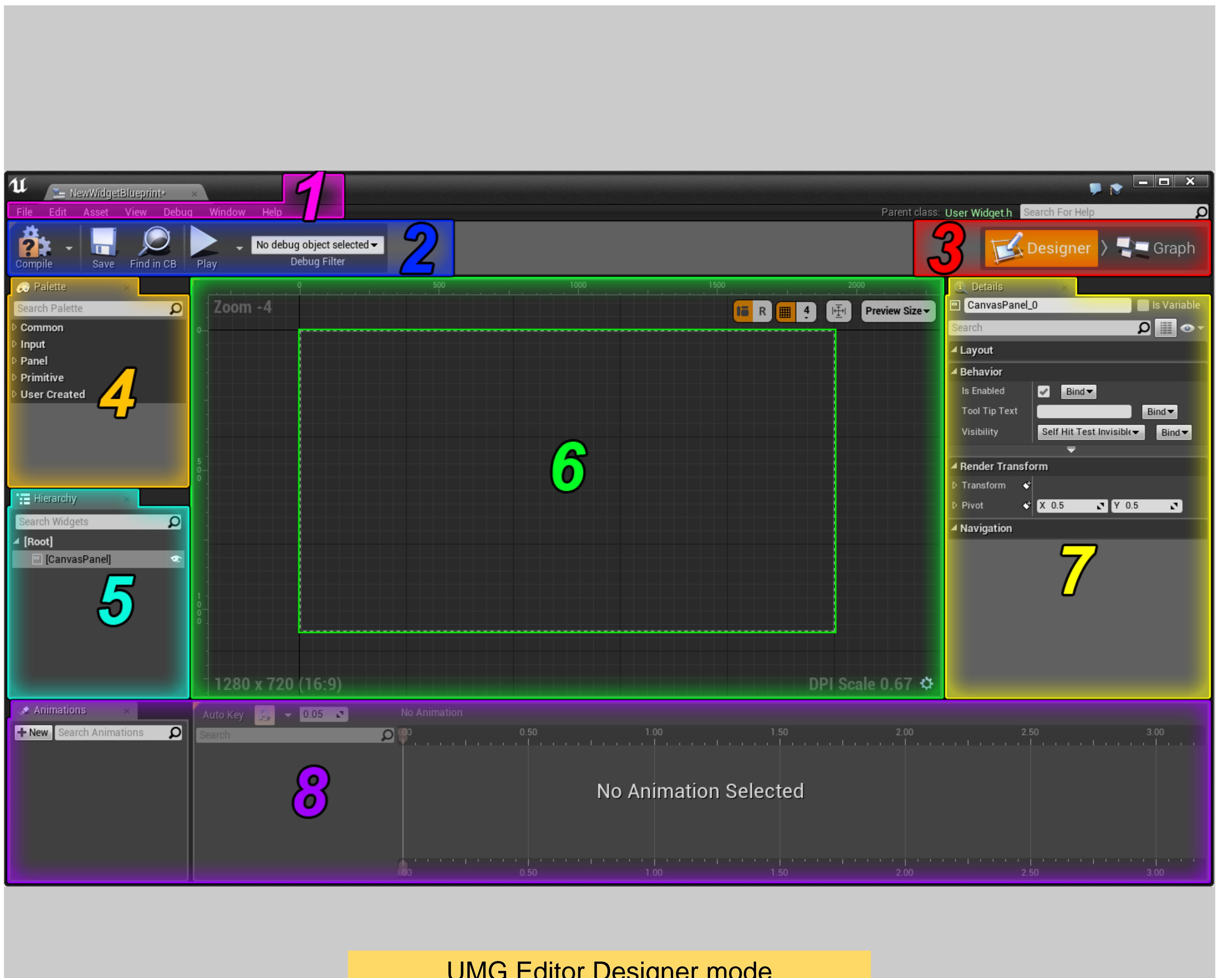
UMG INTERFACE

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.

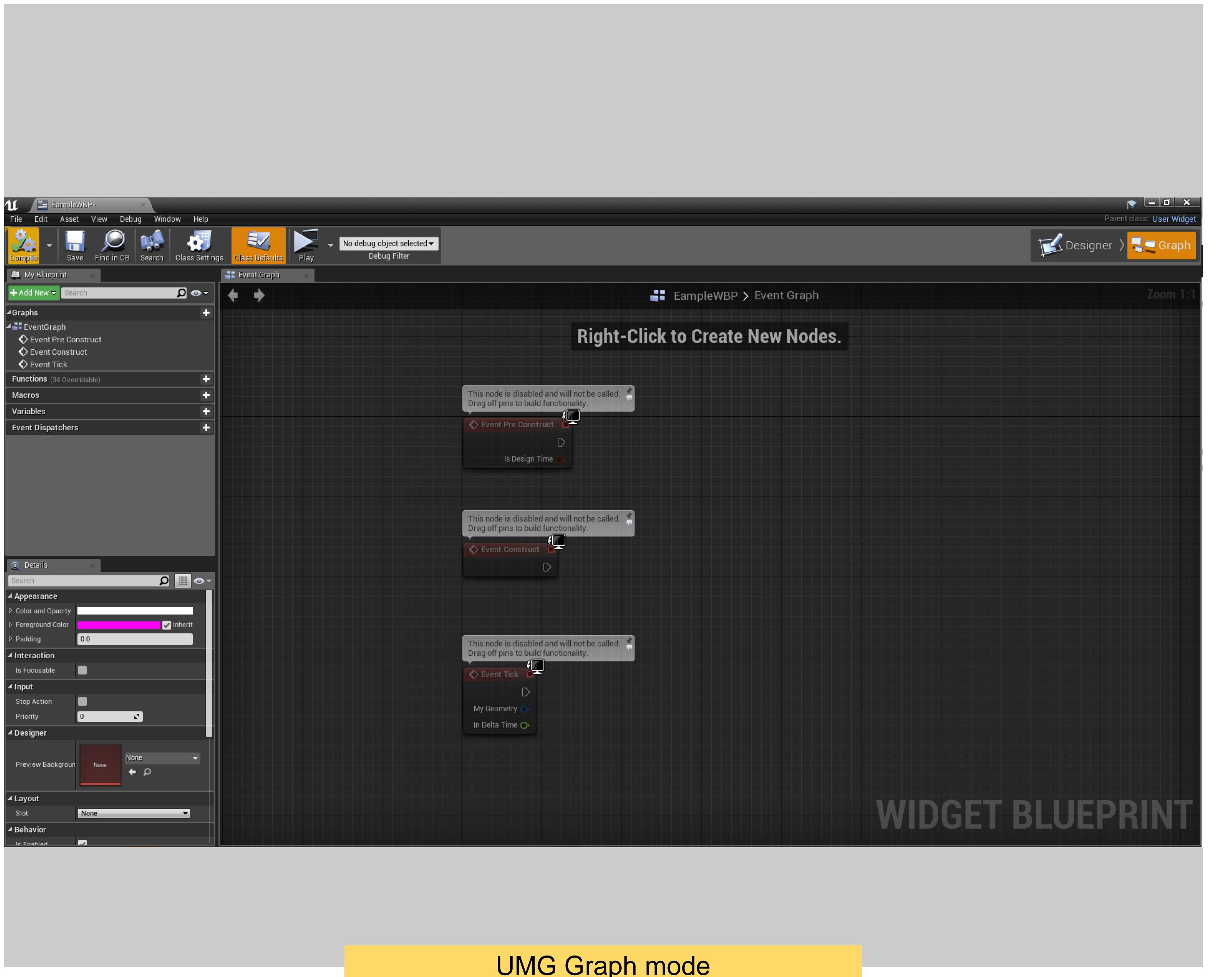


DESIGNER MODE



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.





GRAPH MODE

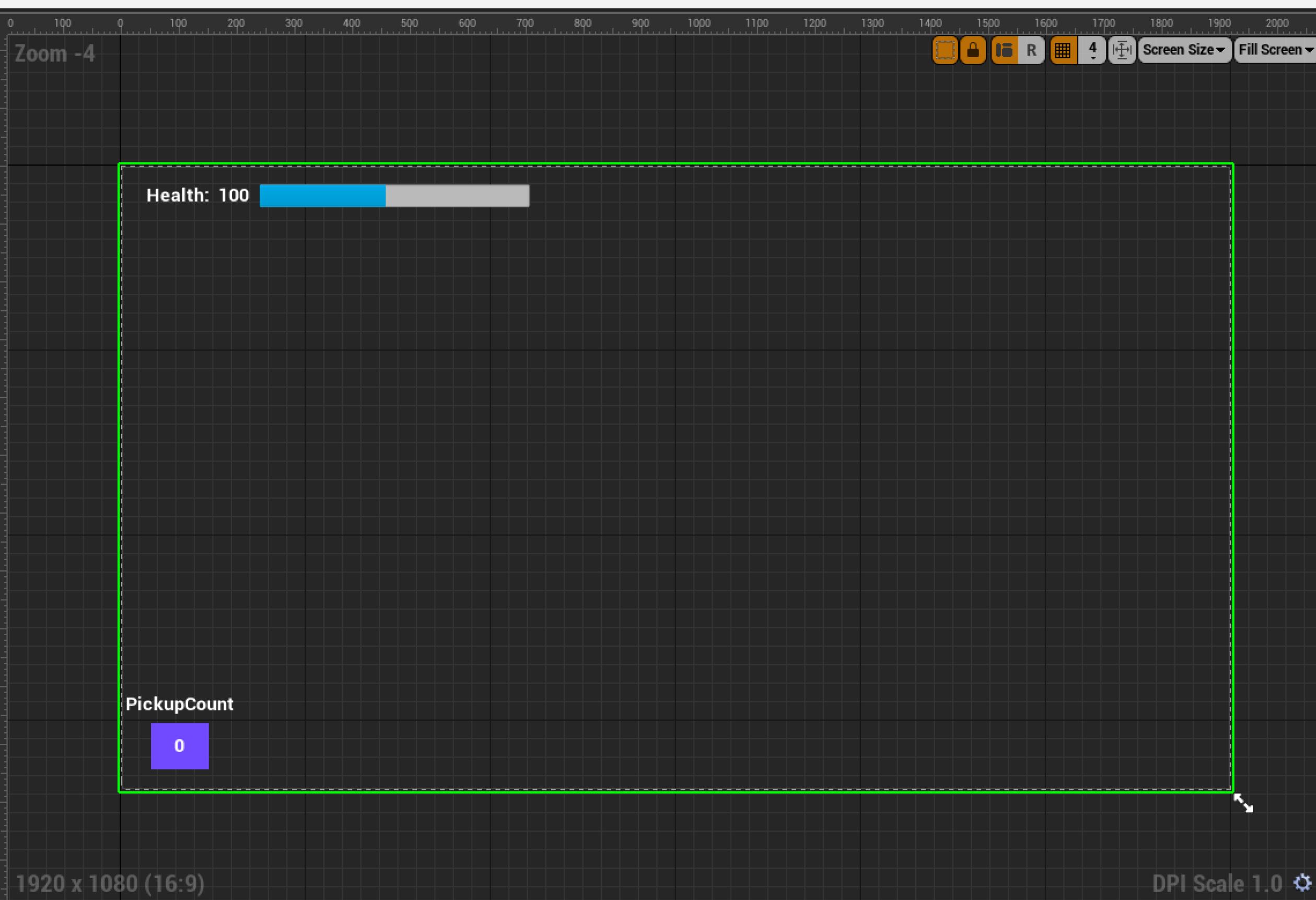
The Graph mode is where you add functionality to your placed widgets. It looks like the standard Blueprint Editor except the Details panel is in the lower left corner by default.





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.

The screenshot shows the Unreal Engine Designer panel interface. At the top right, there are tabs for "Designer" and "Graph". Below the tabs, the "Details" tab is active, showing the "CanvasPanel_0" component. The "Screen Size" dropdown is set to "Fill Screen". The "Animation Selected" dropdown is set to "5.00". In the center, there is a grid workspace with a green dashed line indicating the current screen size. On the right side, the "Details" panel lists various screen sizes categorized by device type:

- Phones**:
 - Apple iPhone 4, 4S, iPod Touch 4 (Portrait)
 - Apple iPhone 4, 4S, iPod Touch 4 (Landscape)
 - Apple iPhone 5, 5S, iPod Touch 5 (Portrait)
 - Apple iPhone 5, 5S, iPod Touch 5 (Landscape)
 - Apple iPhone 6 (Portrait)
 - Apple iPhone 6 (Landscape)
 - Apple iPhone 6+ (Portrait)
 - ✓ Apple iPhone 6+ (Landscape)
 - HTC One (Portrait)
 - ✓ HTC One (Landscape)
 - Samsung Galaxy S4 (Portrait)
 - ✓ Samsung Galaxy S4 (Landscape)
- Tablets**:
 - Apple iPad 2, iPad Mini (Portrait)
 - Apple iPad 2, iPad Mini (Landscape)
 - Apple iPad 3, 4, Air (Portrait)
 - Apple iPad 3, 4, Air (Landscape)
 - Microsoft Surface RT (Portrait)
 - Microsoft Surface RT (Landscape)
 - Microsoft Surface Pro (Portrait)
 - ✓ Microsoft Surface Pro (Landscape)
- Laptops**:
 - Apple MacBook Air 11
 - Apple MacBook Air 13"
 - Apple MacBook Pro 13"
 - Apple MacBook Pro 13" (Retina)
 - Apple MacBook Pro 15"
 - Apple MacBook Pro 15" (Retina)
 - Generic 14-15.6" Notebook
- Monitors**:
 - 19" monitor
 - 20" monitor
 - 22" monitor
 - ✓ 21.5-24" monitor
 - 27" monitor
- Televisions**:
 - 720n (HDTV Blu-ray)

The "27" monitor" option is currently selected, highlighted with a yellow background. The bottom right of the panel shows coordinate values X: 0.5 and Y: 0.5.



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.

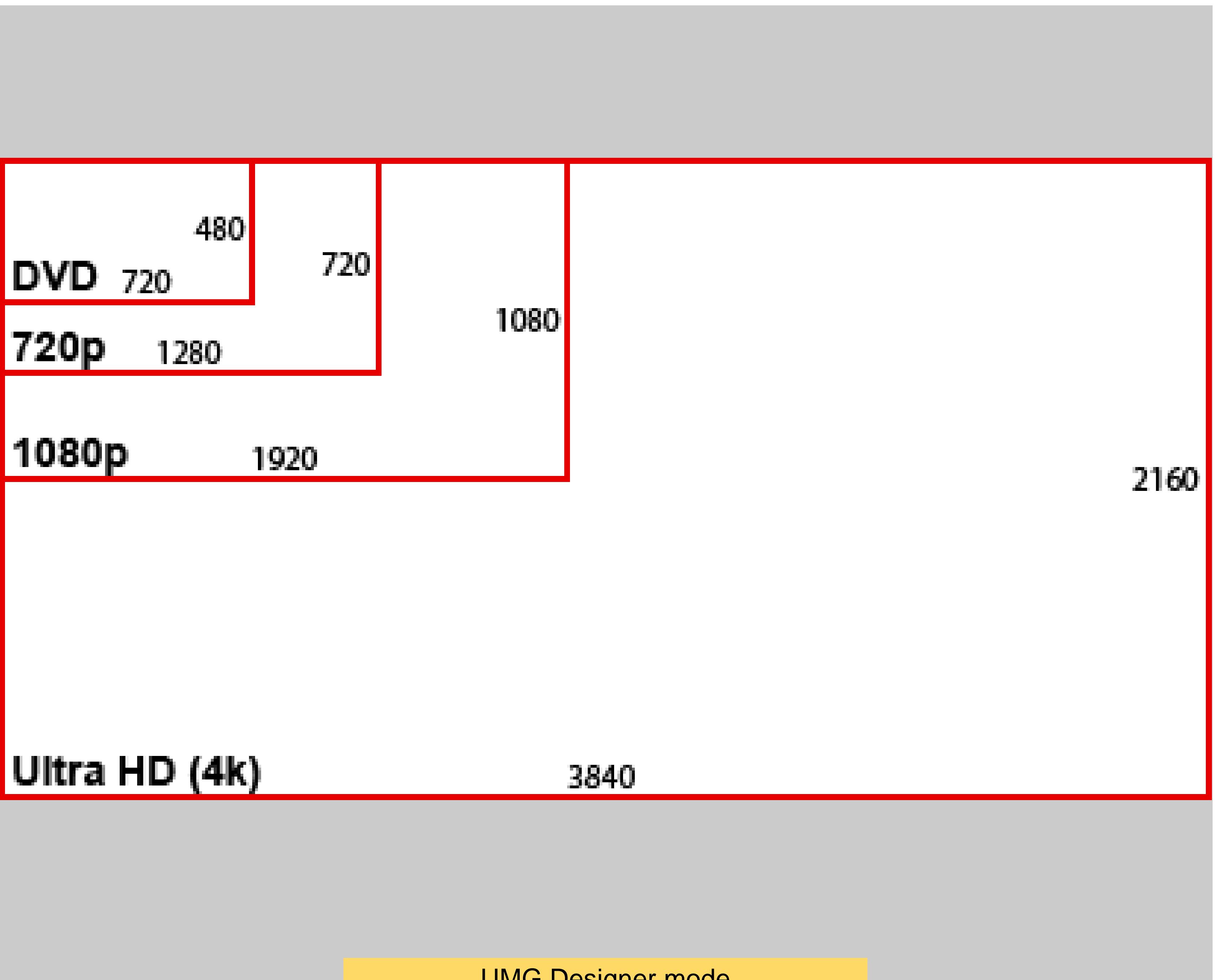
The screenshot shows the Unreal Engine Designer interface with the following details:

- Top Bar:** Designer > Graph
- Properties Panel:** Shows the "CanvasPanel_0" component with the "Screen Size" dropdown set to "Fill Screen".
- Content Area:** A grid-based workspace with a green horizontal line at the top.
- Details Panel:** Displays a list of supported devices categorized by screen size:
 - Phones:** Apple iPhone 4, 4S, iTouch 4 (Portrait), Apple iPhone 4, 4S, iTouch 4 (Landscape), Apple iPhone 5, 5S, iTouch 5 (Portrait), Apple iPhone 5, 5S, iTouch 5 (Landscape), Apple iPhone 6 (Portrait), Apple iPhone 6 (Landscape), Apple iPhone 6+ (Portrait), **✓ Apple iPhone 6+ (Landscape)**, HTC One (Portrait), **✓ HTC One (Landscape)**, Samsung Galaxy S4 (Portrait), **✓ Samsung Galaxy S4 (Landscape)**.
 - Tablets:** Apple iPad 2, iPad Mini (Portrait), Apple iPad 2, iPad Mini (Landscape), Apple iPad 3, 4, Air (Portrait), Apple iPad 3, 4, Air (Landscape), Microsoft Surface RT (Portrait), Microsoft Surface RT (Landscape), Microsoft Surface Pro (Portrait), **✓ Microsoft Surface Pro (Landscape)**.
 - Laptops:** Apple MacBook Air 11, Apple MacBook Air 13", Apple MacBook Pro 13", Apple MacBook Pro 13" (Retina), Apple MacBook Pro 15", Apple MacBook Pro 15" (Retina), Generic 14-15.6" Notebook.
 - Monitors:** 19" monitor, 20" monitor, 22" monitor, **✓ 21.5-24" monitor**, **27" monitor**.
 - Televisions:** 720n (HDTV Blu-ray).
- Right Side:** A vertical toolbar with various icons and settings.



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.



ASPECT RATIOS

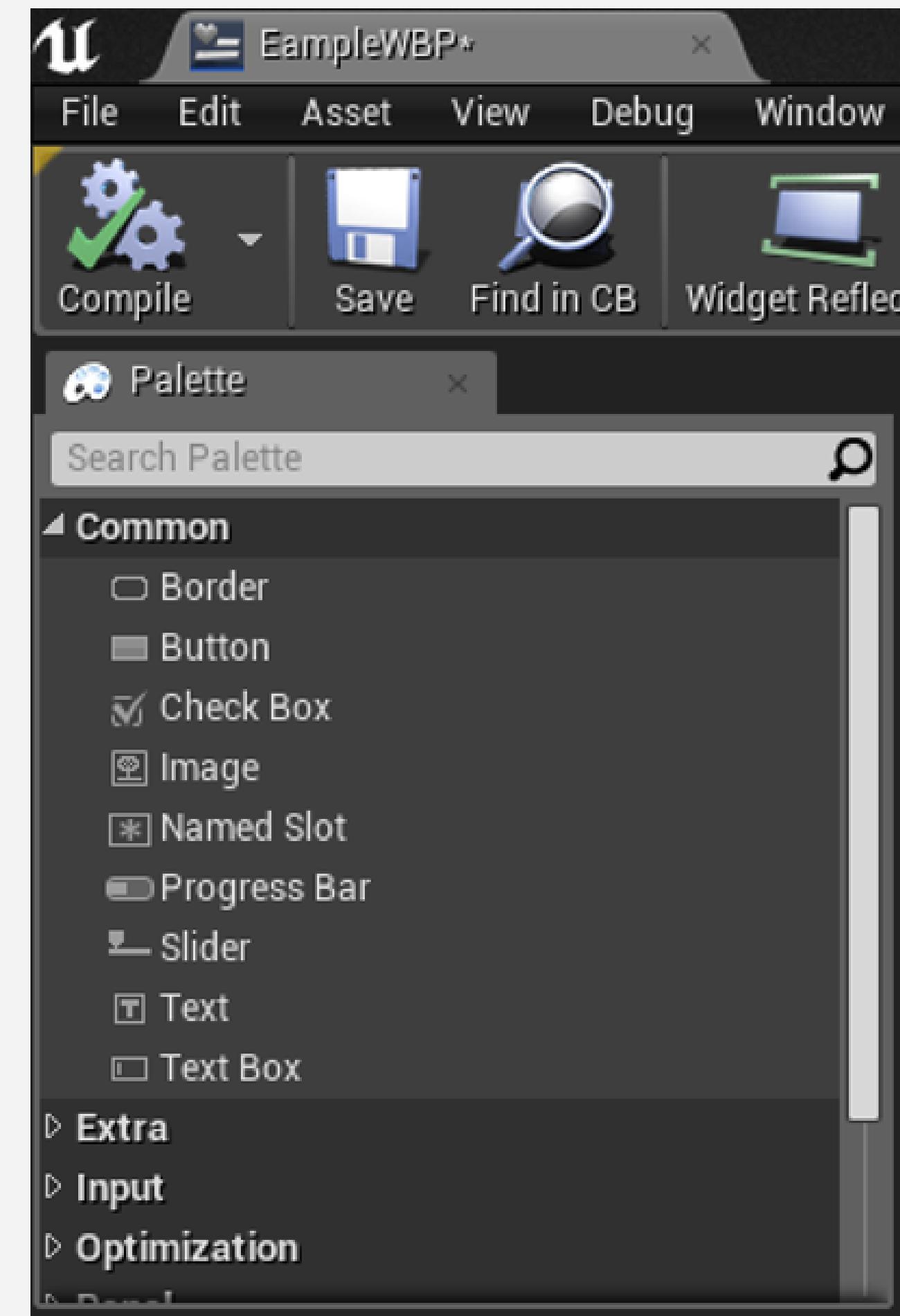
Screen aspect ratios refer to the relationship of horizontal (x) pixels to vertical (y) pixels. For example, if your monitor has a 1920x1080 resolution, its aspect ratio is 16:9. For every 16 pixels across, there are 9 pixels down.





PALETTE PANEL

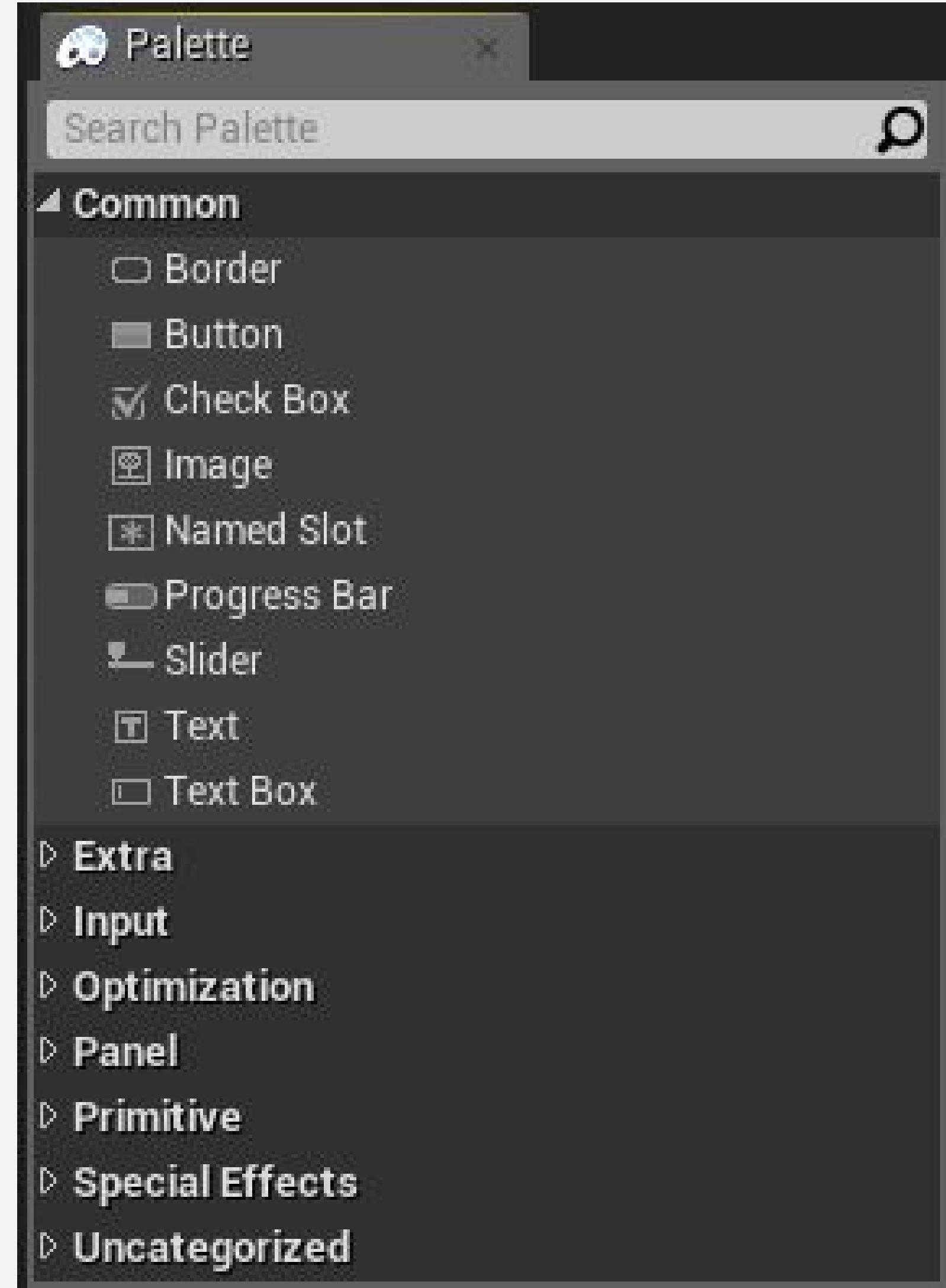
The Palette panel contains a collection of widgets organized by category. Widgets are interface components with defined functionality that are placed in the Designer panel.





PALETTE PANEL

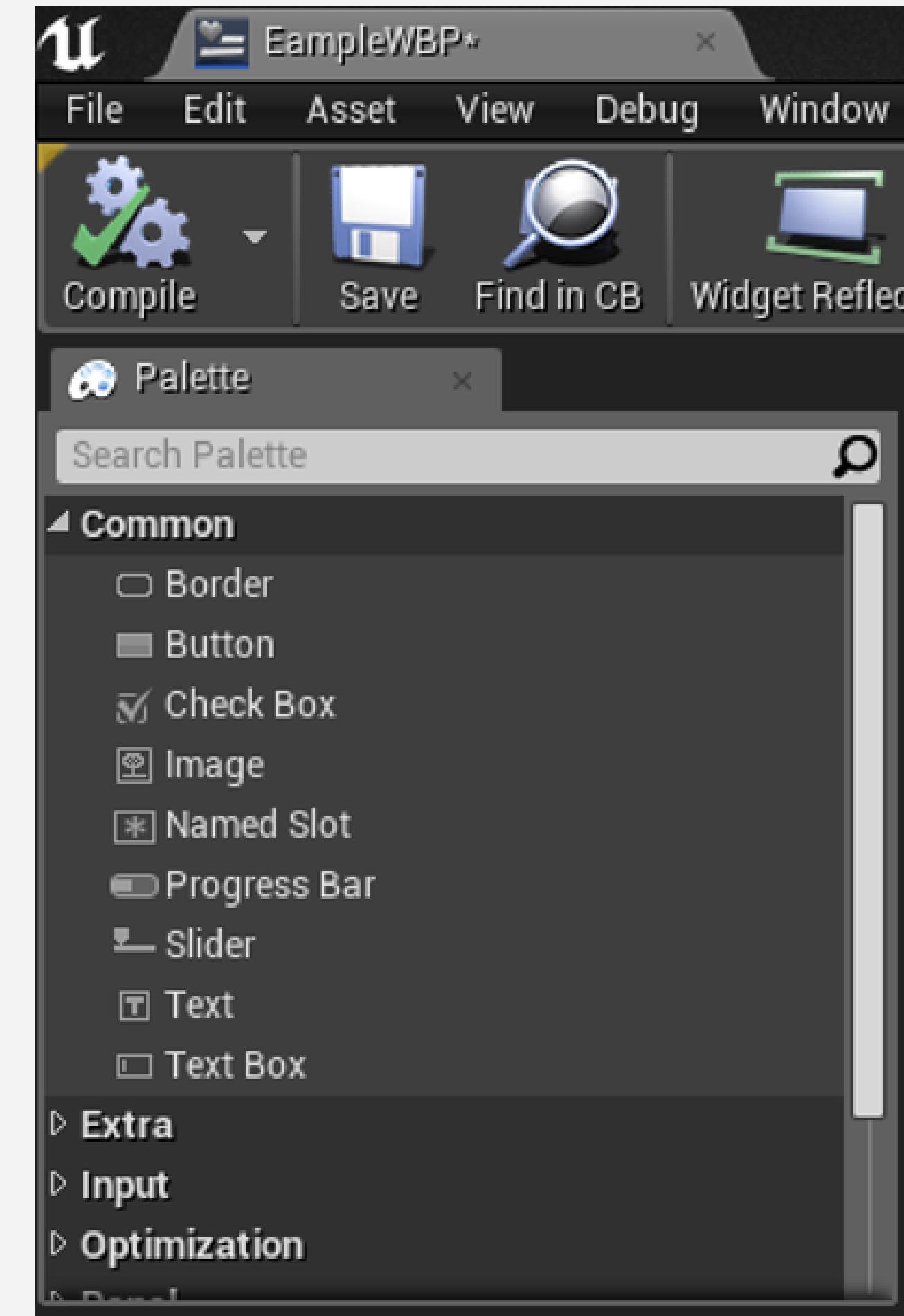
Each category of widgets contains different widget types that you can drag and drop into the Designer panel. By mixing and matching these widget types, you can lay out the look of your UI on the Designer tab and add the functionality to your widgets through the settings in the Details panel for each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





WIDGETS

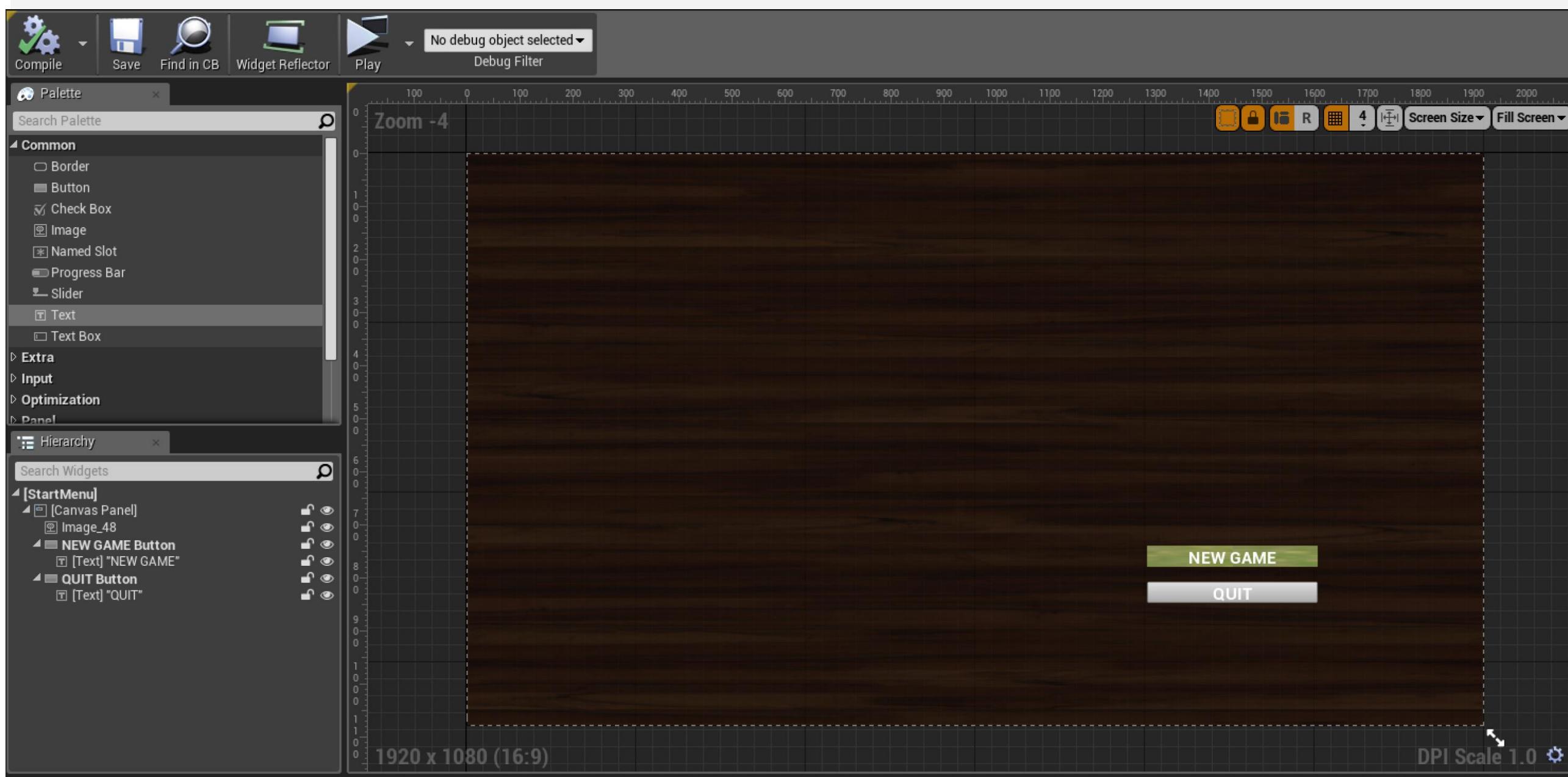
Widget types range from simple Image widgets used to place images (Textures or Materials) to Text widgets used to place text, and Progress Bar widgets used to display player stats or Level load bars. There are also formatting widgets used to organize other widgets placed in the interface.





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

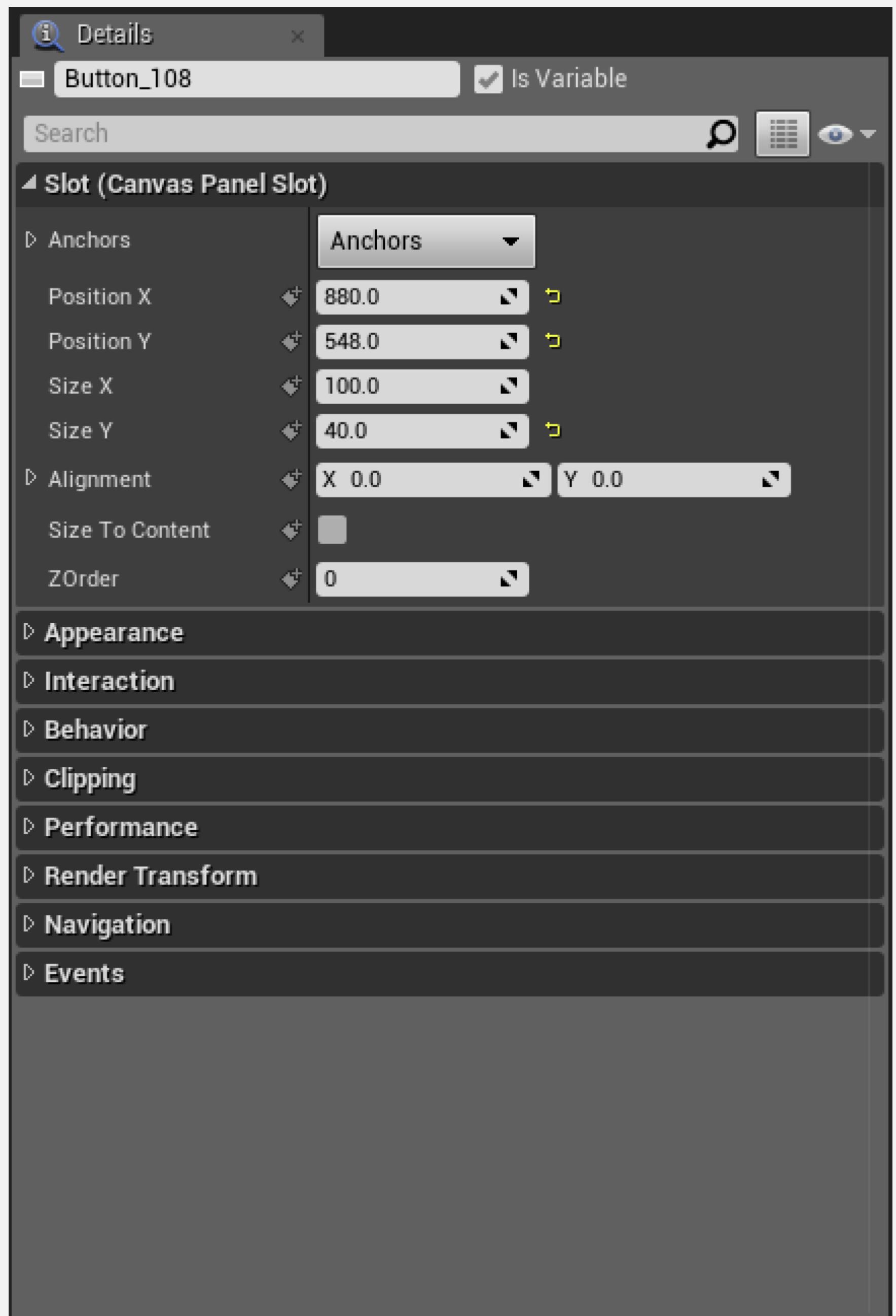
The UMG Details panel displays the properties of the currently selected widget. While each widget has properties that are unique to its functionality, there are some common properties found on every widget placed, such as the properties found in the Slot tab.





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 to 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.

A screenshot of the Unreal Engine's Details panel for a UI component named "Button_108". The panel shows various properties for the button, including its position, size, and anchor settings. The "Slot (Canvas Panel Slot)" section is expanded, displaying the following configuration:

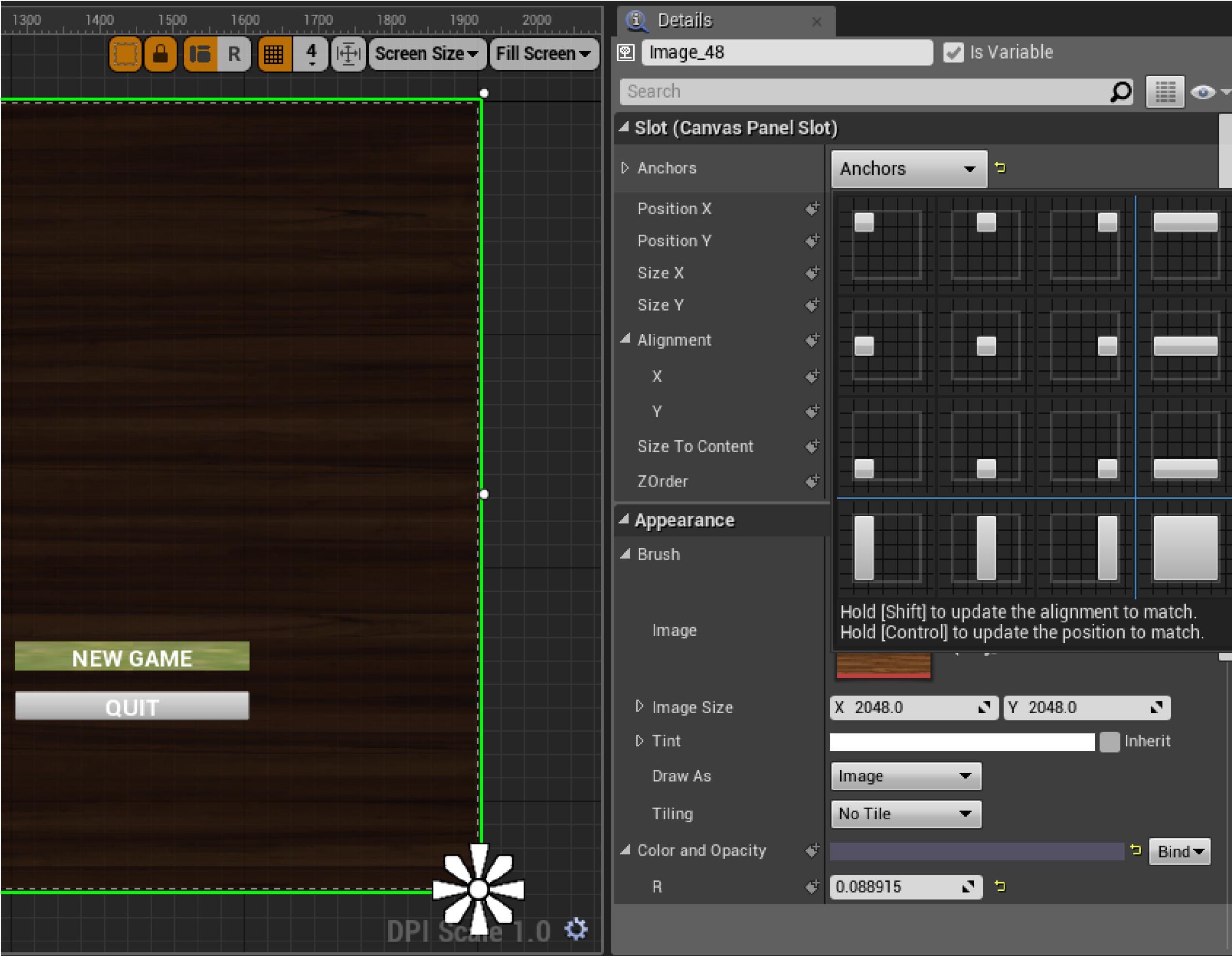
Setting	Value
Position X	880.0
Position Y	548.0
Size X	100.0
Size Y	40.0
Alignment	X: 0.0 Y: 0.0
Size To Content	Enabled (checkbox)
ZOrder	0

The "Is Variable" checkbox is checked. Other collapsed sections include Appearance, Interaction, Behavior, Clipping, Performance, Render Transform, Navigation, and Events.



ANCHORS

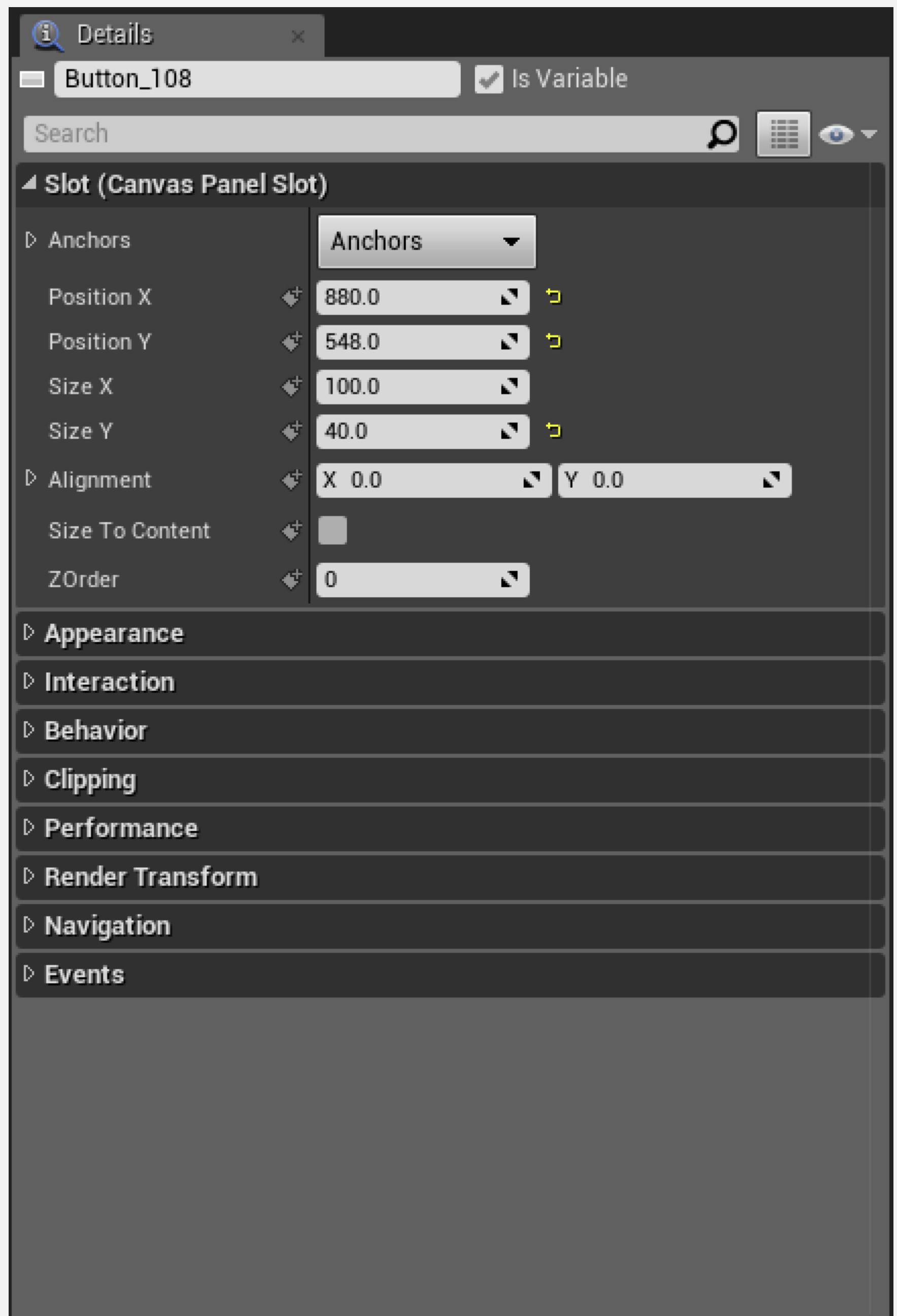
While a widget's anchor can be positioned manually, there are common presets that allow you to quickly place them.





ANCHORS

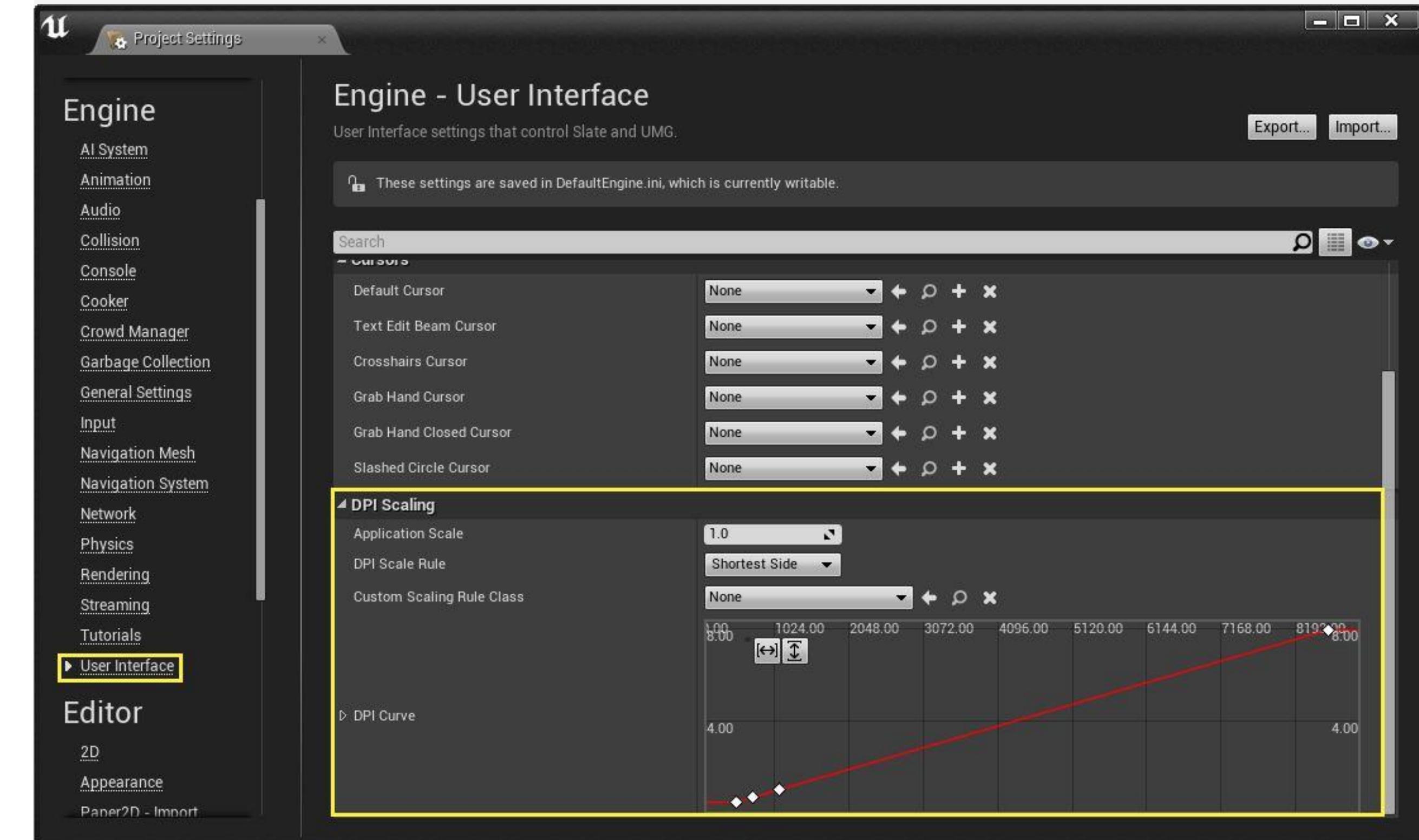
Because anchors use a normalized value, their position is determined by percentage and not by a set pixel coordinate. This means that if the aspect ratio or resolution of the screen changes, the widget will adjust accordingly and maintain its relative position on screen. Anchors work directly with DPI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

The screenshot shows the 'Project Settings' window for the 'Engine - User Interface' section. The left sidebar lists various engine components, and the right pane shows settings for the User Interface. A yellow box highlights the 'DPI Scaling' section, which contains the following configuration:

- Application Scale:** Set to 1.0.
- DPI Scale Rule:** Set to "Shortest Side".
- Custom Scaling Rule Class:** Set to "None".
- DPI Curve:** A graph showing a red curve that scales from approximately 0.5 at 800 pixels to 4.0 at 8192 pixels. The x-axis represents screen width in pixels, and the y-axis represents the scale factor.



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

The screenshot shows the Unreal Engine's Project Settings interface. The left sidebar lists categories like Project, Game, and Engine. In the main content area, under the Engine category, the 'User Interface' section is selected. The 'DPI Scaling' panel is open, showing the 'Application Scale' set to 1.0 and the 'DPI Scale Rule' dropdown set to 'Shortest Side'. A graph titled 'DPI Curve' is present, showing a single data point at coordinates (18.00, 4.00).

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.

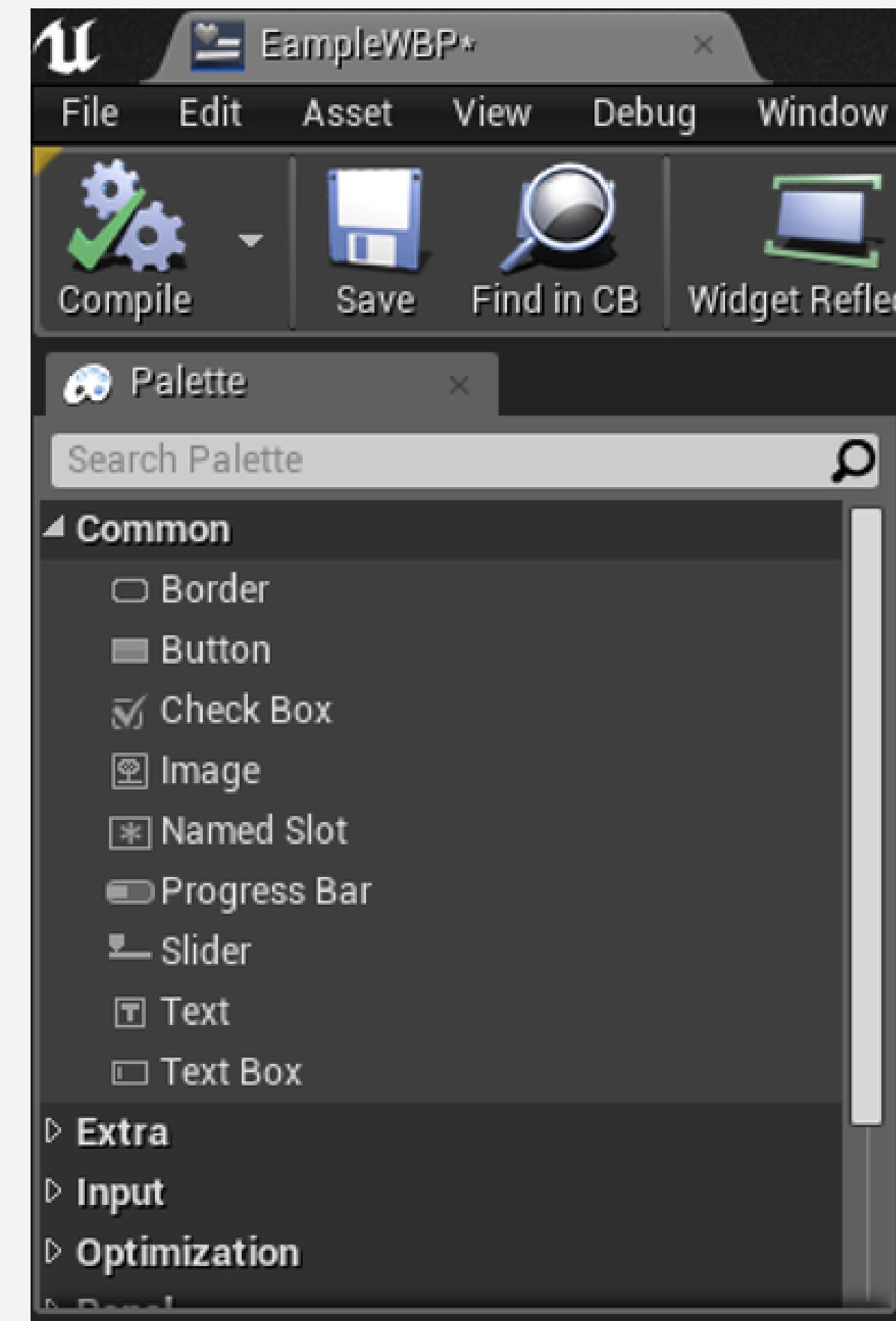




IMAGE WIDGET

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.

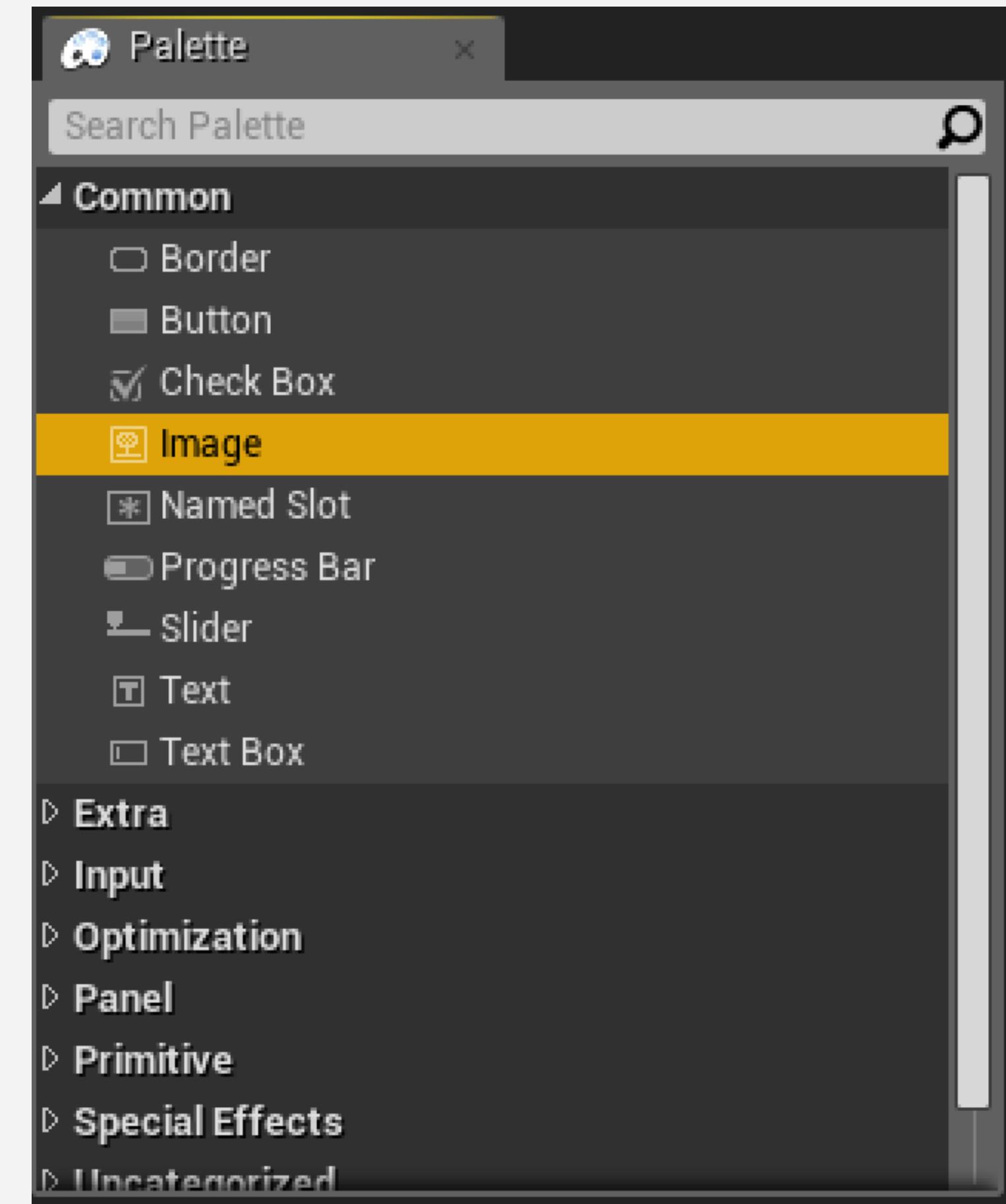




IMAGE WIDGET

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.

The screenshot shows the Unreal Engine Designer interface. At the top, the title bar reads "Parent class: User Widget". Below the title bar, there are tabs for "Designer" (which is selected) and "Graph". The main area displays a dark brown wooden texture in a canvas panel slot. The slot has a green border and a white flower-shaped handle at the bottom center. The "Details" panel on the right contains the following settings:

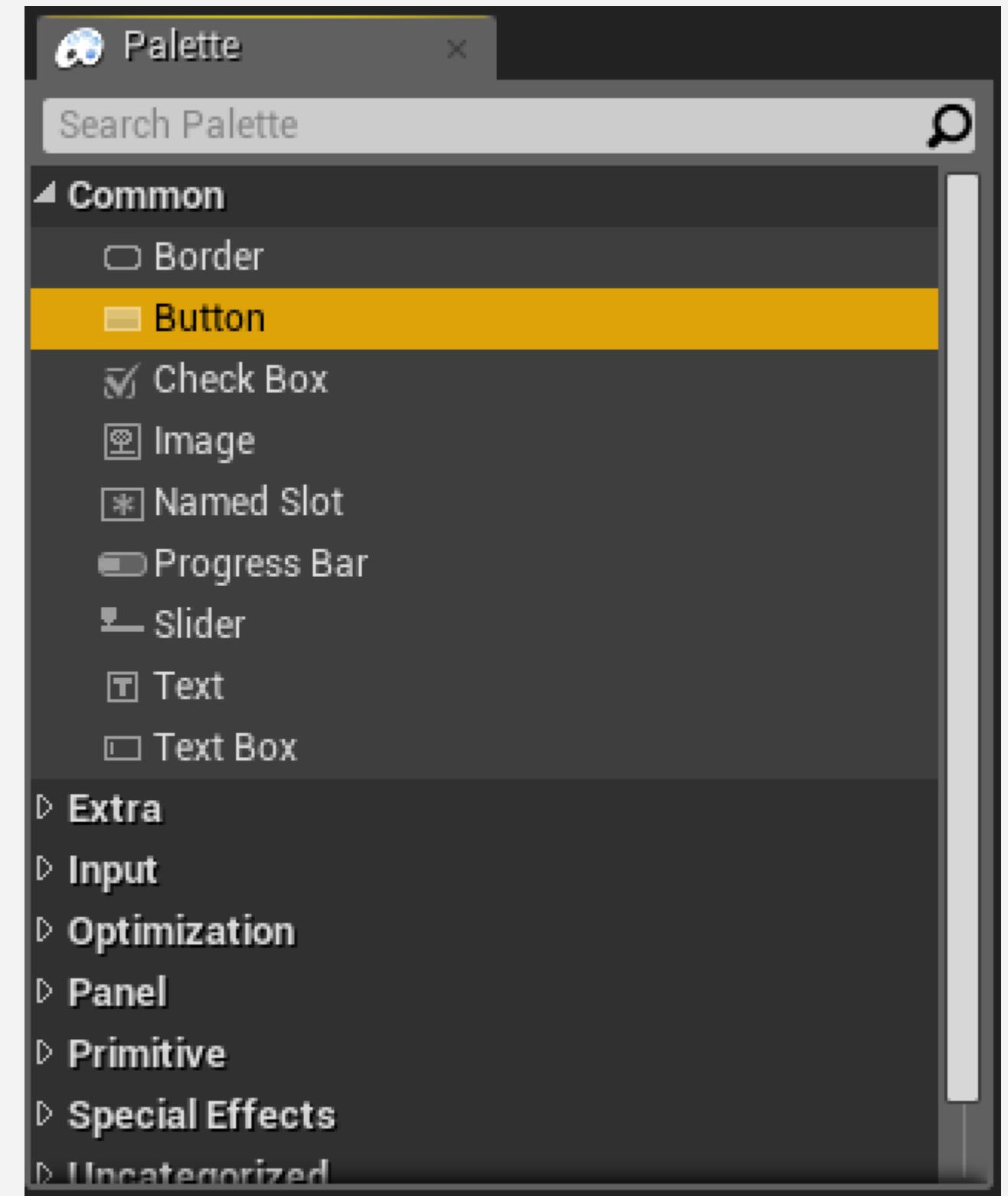
- Slot (Canvas Panel Slot)**
 - Position X: -1920.0
 - Position Y: -1080.0
 - Size X: 1920.0
 - Size Y: 1080.0
 - Alignment: X: 0.0, Y: 0.0
 - Size To Content: checked
 - ZOrder: 0
- Appearance**
 - Brush**: T_Wood_Oak_D (Bind)
 - Image**: T_Wood_Oak_D
 - Image Size**: X: 2048.0, Y: 2048.0
 - Tint**: Inherit
 - Draw As**: Image
 - Tiling**: No Tile
 - Color and Opacity**: R: 0.088915, G: 0.082101, B: 0.130208, A: 1.0 (Bind)

At the bottom of the interface, the text "DPI Scale 1.0" is visible.



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

The screenshot shows the Unreal Engine Editor interface. At the top, there's a toolbar with icons for selection, lock, rotation, and a grid. To the right of the toolbar are buttons for 'Screen Size' (set to 4), 'Fill Screen', and other settings. Below the toolbar is a dark brown textured background. In the center, there are two buttons: a green 'NEW GAME' button with white text and a grey 'QUIT' button below it. A green outline highlights the 'NEW GAME' button, indicating it is the active or selected item. On the right side of the screen is the 'Details' panel, which contains the following sections:

- Search:** A search bar with a magnifying glass icon.
- Appearance:** A section with a heading and several dropdown menus:
 - Style:** A dropdown menu with three items: 'Normal' (highlighted with a red box), 'Hovered', and 'Pressed'.
 - Image:** A preview window showing a grassy texture labeled 'T_Ground_Grass_D'. Below it are dropdowns for 'X' (2048.0) and 'Y' (2048.0) coordinates, and a 'Box' margin set to 0.25.
 - Margin:** A dropdown menu with 'Normal Padding' and 'Pressed Padding' options, both set to '2.0'.
 - Color and Opacity:** A section with color swatches for R (1.0), G (1.0), B (1.0), and A (1.0).
 - Background Color:** A section with color swatches for R (1.0), G (1.0), B (1.0), and A (1.0).
- Interaction:** A section with a 'DPI Scale 1.0' slider.
- Behavior:** A section at the bottom.



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.

The screenshot shows the Unreal Engine Editor interface. At the top, there is a toolbar with various icons and a "Screen Size" dropdown set to "4". Below the toolbar is a grid-based workspace. In the center of the workspace, there is a button labeled "NEW GAME" with a green outline, and below it is another button labeled "QUIT". To the right of the workspace is the "Details" panel for the "NEW GAME Button". The "Appearance" section is expanded, showing the "Normal" style. Under "Normal", the "Image" property is highlighted with a red box. A thumbnail for "T_Ground_Grass_D" is shown, which is a green grass texture. Other properties visible in the "Normal" section include "Image Size" (X: 2048.0, Y: 2048.0), "Tint", "Draw As" (set to "Box"), and "Margin" (set to 0.25). Below the "Normal" section, other states like "Hovered", "Pressed", and "Disabled" are listed. The "Color and Opacity" section shows R, G, B, and A values all set to 1.0. The "Background Color" section also shows R, G, B, and A values all set to 1.0. The "Interaction" and "Behavior" sections are at the bottom of the Details panel.



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.

The screenshot shows the Unreal Engine Editor's Details panel for a Button widget named "NEW GAME Button". The panel is divided into several sections:

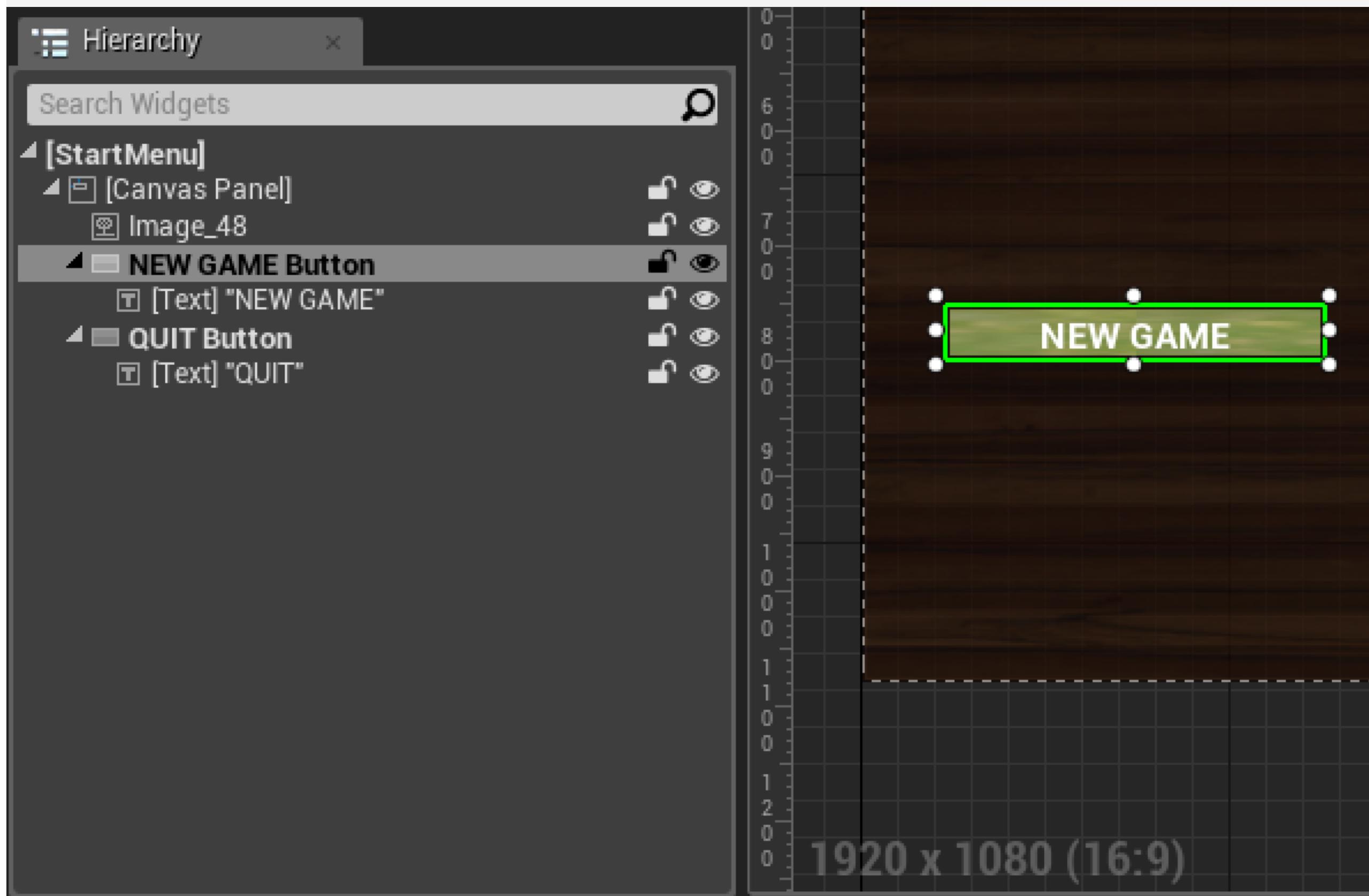
- Appearance**:
 - Style**:
 - Normal**: Shows an image of a green grassy field (T_Ground_Grass_D).
 - Image Size**: Set to X: 2048.0, Y: 2048.0.
 - Tint**: A color bar with "Inherit" checked.
 - Draw As**: Set to "Box".
 - Margin**: Set to 0.25.
 - Hovered**: Shows a dark gray gradient.
 - Pressed**: Shows a light gray gradient.
 - Disabled**: Shows a medium gray gradient.
 - Normal Padding**: Set to 2.0.
 - Pressed Padding**: Set to 2.0, 3.0, 2.0, 1.0.
 - Pressed Sound**: Set to "Collapse01".
 - Hovered Sound**: Set to "None".
- Color and Opacity**:
 - R: 1.0
 - G: 1.0
 - B: 1.0
 - A: 1.0
- Background Color**:
 - R: 1.0
 - G: 1.0
 - B: 1.0
 - A: 1.0
- Interaction**:
 - DPI Scale 1.0
- Behavior**: (This section is partially visible at the bottom)



BUTTON WIDGET

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

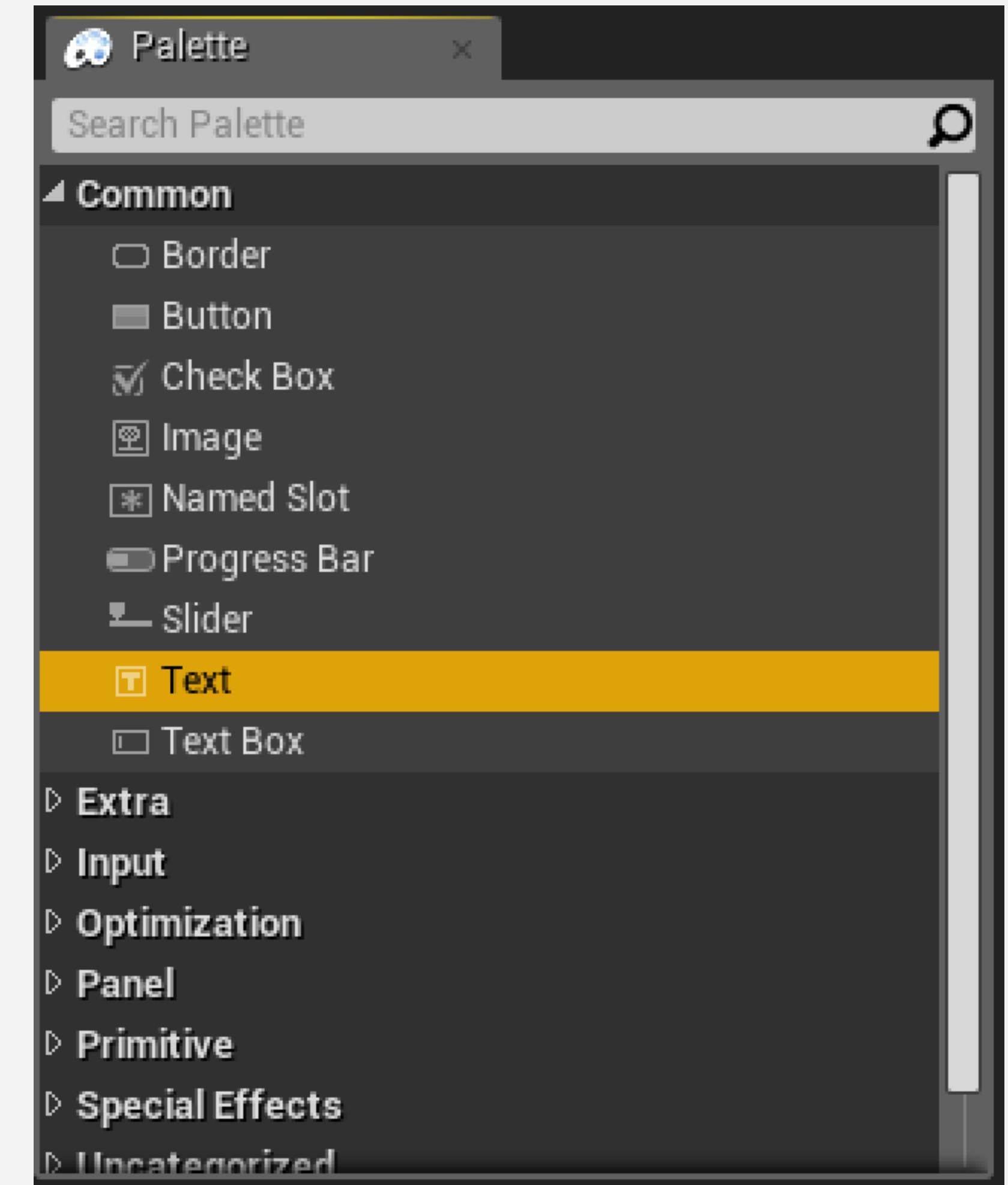
For example, if you drag a Text widget onto the Button, it will snap to the Button and show up as a child of the Button widget in the Hierarchy panel.





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.

The screenshot shows the Details panel for a component named "TextBlock_80". The "Content" section is highlighted with a red border, specifically the "Text" field which contains the value "NEW GAME". Other visible settings include Padding (4.0, 2.0), Horizontal Alignment (center), Vertical Alignment (center), and various Appearance options like Color and Opacity (R: 1.0, G: 1.0, B: 1.0, A: 1.0), Font (Roboto, Bold, Size: 24), and Outline Settings (Shadow Offset X: 1.0, Y: 1.0).



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.

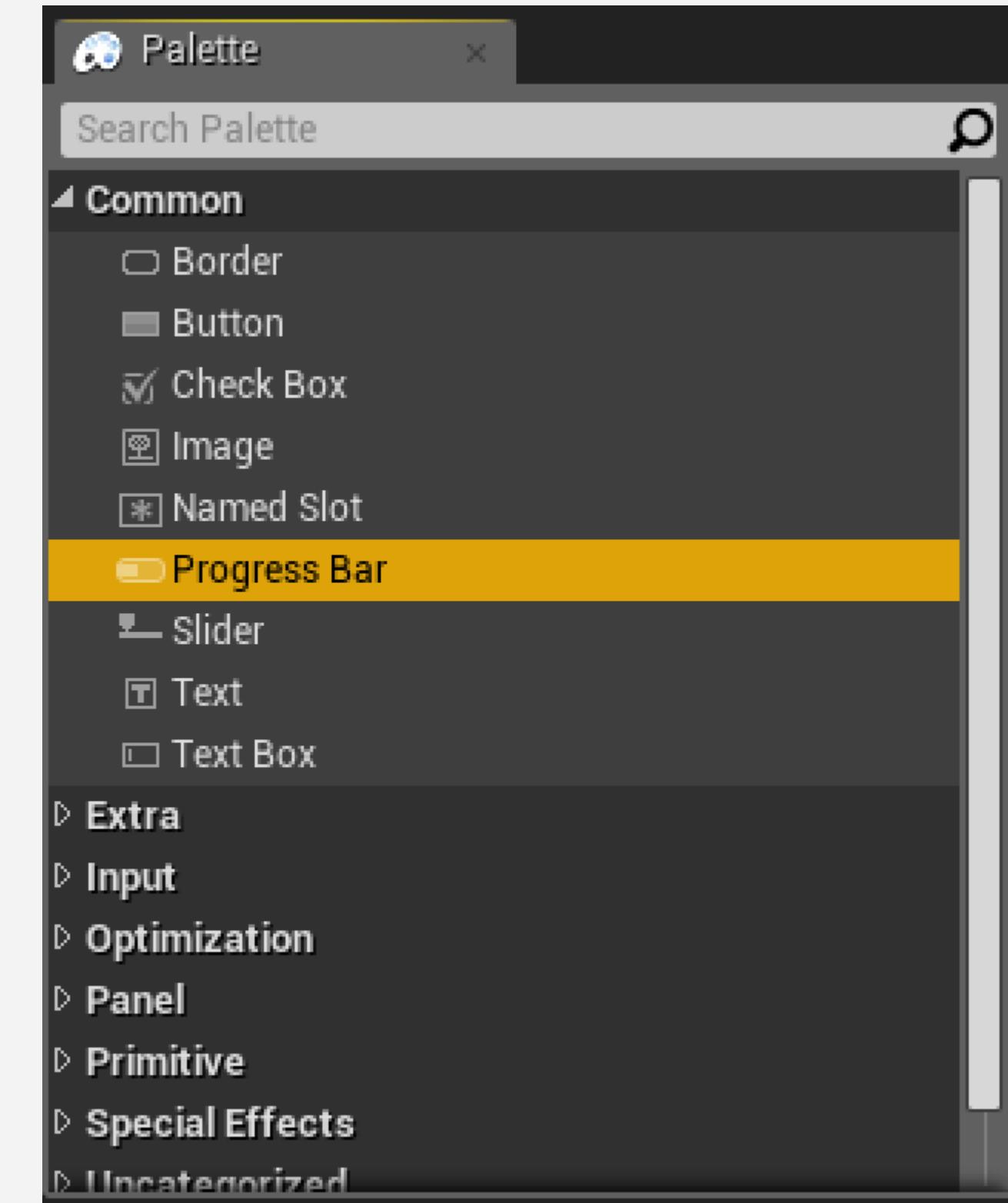
The screenshot shows the Unreal Engine's Details panel for a component named "TextBlock_80". The panel is divided into several sections:

- Slot (Button Slot)**: Contains settings for Padding (4.0, 2.0), Horizontal Alignment (center), and Vertical Alignment (center).
- Content**: Shows the current Text value as "NEW GAME" with a "Bind" button.
- Appearance**: This section is expanded and contains:
 - Color and Opacity**: Shows R, G, B, A values all set to 1.0, with an "Inherit" checkbox and a "Bind" button.
 - Font**: This section is also expanded and contains:
 - Font Family**: Set to "Roboto", highlighted with a red rectangle.
 - Typeface**: Set to "Bold".
 - Size**: Set to 24.
 - Font Material**: Set to "None".
 - Outline Settings**: Shows X and Y offsets both at 1.0.
 - Shadow Offset**: Shows X and Y offsets both at 1.0.
 - Shadow Color**: A color swatch set to black, with a "Bind" button.
 - Min Desired Width**: Set to 0.0.
 - Justification**: Set to center.
 - Margin**: Set to 0.0.
 - Line Height Percentage**: Set to 1.0.



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.

The screenshot shows the Unreal Engine Editor interface with a Progress Bar Widget selected. The top bar displays various icons and a coordinate grid. The main workspace contains a progress bar with a blue fill and a green outline. The Details panel on the right provides configuration options:

- Slot (Canvas Panel Slot)**: Contains a preview of the progress bar.
- Style**:
 - Background Image**: Set to "None".
 - Image**: Set to "None".
 - Image Size**: X: 0.0, Y: 0.0.
 - Tint**: Inherit.
 - Draw As**: Box.
 - Margin**: 0.416667.
 - Fill Image**: A small gray square.
 - Marquee Image**: A small gray square.
- Progress**:
 - Percent**: 0.466667, bound to "GetPercent_0".
 - Bar Fill Type**: Left to Right.
 - Is Marquee**: Off.
 - Border Padding**: X: 0.0, Y: 0.0.
- Appearance**:
 - Fill Color and Opacity**: Bind.
 - R**: 0.0.
 - G**: 0.5.
 - B**: 1.0.
 - A**: 1.0.
- Behavior**, **Clipping**, and **Performance** sections are also present.

DPI Scale 1.0

TEXTURES

Prepping Textures for Interfaces



TEXTURES

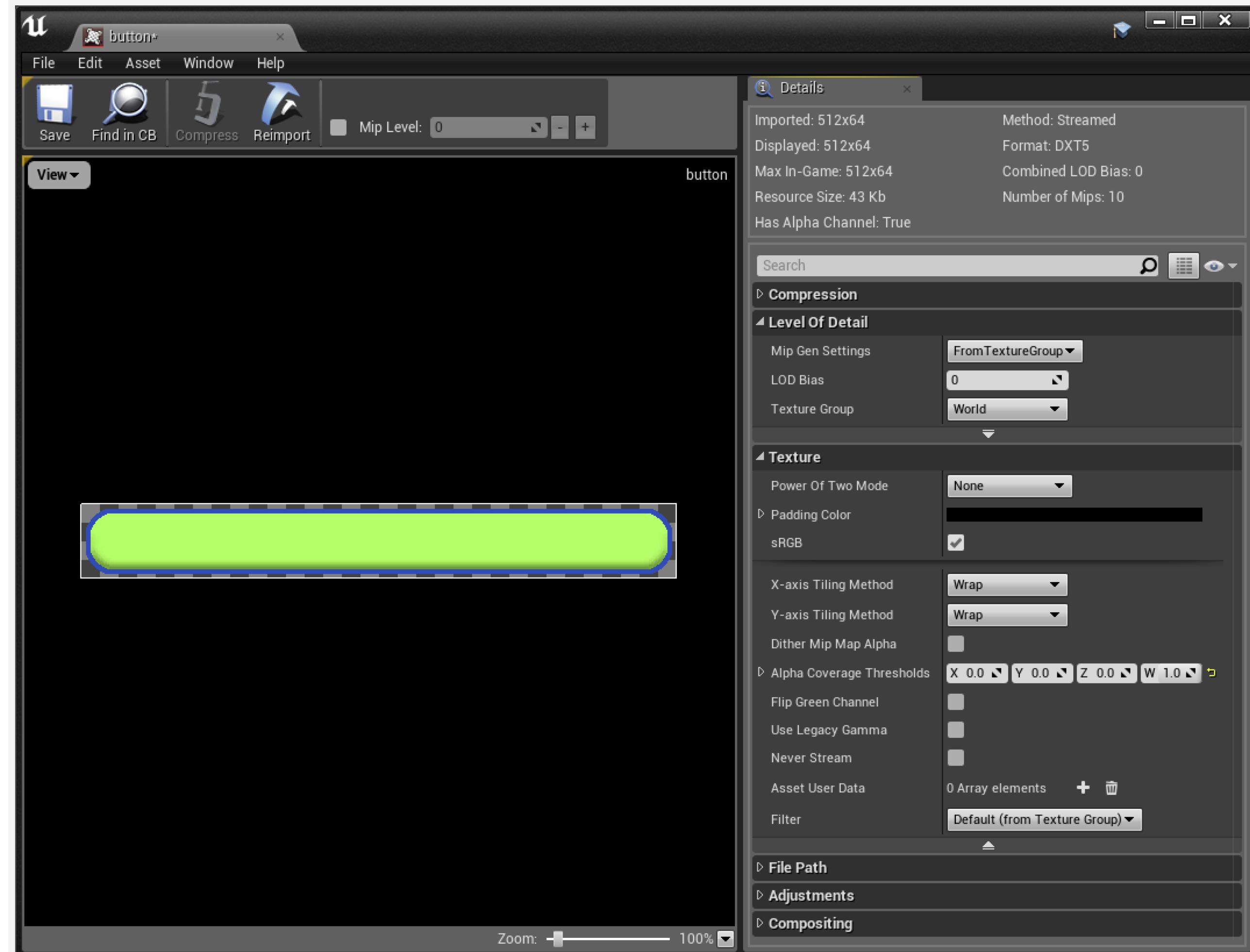
Widgets such as the Button and the Image widget can use Textures or Materials for style settings. When using Textures, there are some basic settings to change.

The screenshot shows the Unreal Engine's Details panel for a 'NEW GAME Button'. The 'Appearance' section is expanded, showing the 'Style' settings. Under 'Normal', the 'Image' is set to 'T_Ground_Grass_D'. Other visible settings include 'Image Size' (X: 2048.0, Y: 2048.0), 'Tint' (Inherit), 'Draw As' (Box), 'Margin' (0.25), and various padding and sound options. Below the 'Appearance' section, 'Color and Opacity' and 'Background Color' sections are expanded, each showing R, G, B, and A values all set to 1.0. At the bottom, 'Interaction' and 'Behavior' sections are partially visible.



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.

The screenshot shows the Unreal Engine Texture Editor's Details panel for a texture asset named "Imported".

General Properties:

- Imported: 512x64
- Displayed: 512x64
- Max In-Game: 512x64
- Resource Size: 43 Kb
- Has Alpha Channel: True
- Method: Streamed
- Format: DXT5
- Combined LOD Bias: 0
- Number of Mips: 10

Compression:

Mip Gen Settings: FromTextureGroup

LOD Bias: 0

Texture Group: World

Texture:

Power Of Two Mode: None

Padding Color: Black

sRGB: Checked

X-axis Tiling Method: Wrap

Y-axis Tiling Method: Wrap

Dither Mip Map Alpha: Off

Alpha Coverage Thresholds: X: 0.0, Y: 0.0, Z: 0.0, W: 1.0

Flip Green Channel: Off

Use Legacy Gamma: Off

Never Stream: Off

Asset User Data: 0 Array elements

Filter: Default (from Texture Group)

File Path:

Adjustments:

Compositing:



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.

The screenshot shows the 'Details' panel in the Unreal Engine Editor. At the top, it displays basic texture information: Imported: 512x64, Displayed: 512x64, Max In-Game: 512x64, Resource Size: 43 Kb, Has Alpha Channel: True. To the right, Method: Streamed, Format: DXT5, Combined LOD Bias: 0, and Number of Mips: 10 are listed. Below this, the 'Compression' tab is selected. Under 'Level Of Detail', the 'Mip Gen Settings' dropdown is set to 'FromTextureGroup'. The 'Texture Group' dropdown is open, showing a list of options: World, WorldNormalMap, WorldSpecular, Character, CharacterNormalMap, CharacterSpecular, Weapon, WeaponNormalMap, WeaponSpecular, Vehicle, VehicleNormalMap, VehicleSpecular, Cinematic, Effects, EffectsNotFiltered, Skybox, UI, Lightmap, RenderTarget, MobileFlattened, ProcBuilding_Face, ProcBuilding_LightMap, Shadowmap, ColorLookupTable, Terrain_Heightmap, Terrain_Weightmap, Bokeh, IESLightProfile, 2D Pixels (unfiltered), and Hierarchical LOD. The 'World' option is currently selected. The bottom of the panel shows numerical fields for X, Y, Z, and W, and a 'Group' dropdown.



TEXTURES

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

The screenshot shows the 'Details' panel in the Unreal Engine Editor. The panel displays various properties for a selected texture:

- Imported:** 512x64
- Displayed:** 512x64
- Max In-Game:** 512x64
- Resource Size:** 43 Kb
- Has Alpha Channel:** True
- Method:** Not Streamed
- Format:** DXT5
- Combined LOD Bias:** 0
- Number of Mips:** 10

Compression: FromTextureGroup

Level Of Detail:

- Mip Gen Settings: FromTextureGroup
- LOD Bias: 0
- Texture Group: World

Texture:

- Power Of Two Mode: None
- Padding Color: Black
- sRGB: Checked
- X-axis Tiling Method: Wrap
- Y-axis Tiling Method: Wrap
- Dither Mip Map Alpha: Off
- Alpha Coverage Thresholds:
 - X: 0.0
 - Y: 0.0
 - Z: 0.0
 - W: 1.0
- Flip Green Channel: Off
- Use Legacy Gamma: Off
- Never Stream: Checked
- Asset User Data: 0 Array elements
- Filter: Default (from Texture Group)

File Path:

Adjustments:

Compositing: