USER INTERFACES

Relaying Information to the Player

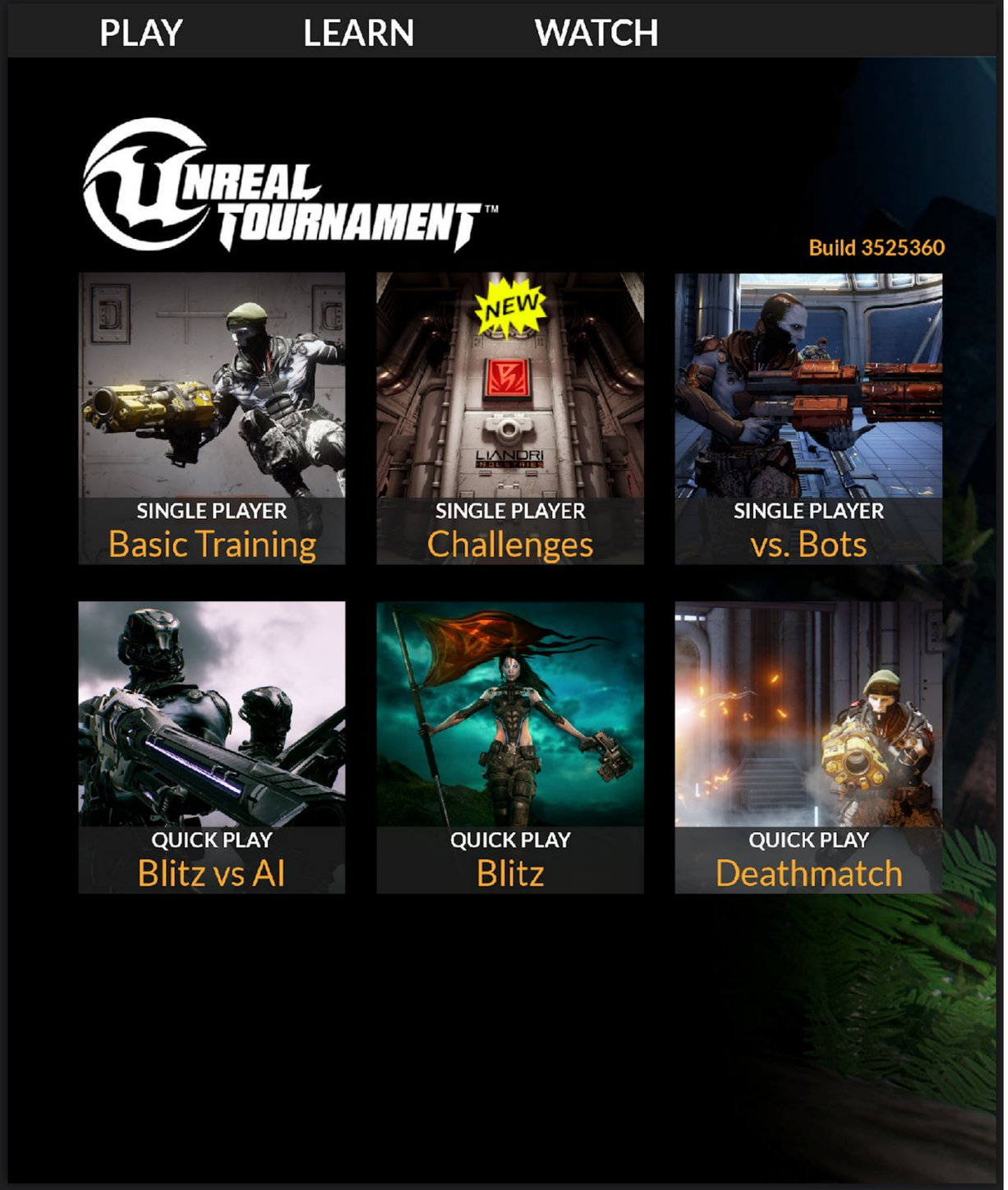
UI 101

The Basics

WHAT IS A USER INTERFACE?

The User Interface of a game is the set of methods and features that allow the player to interact with the game, and the game to provide information to the player. It can include a range of mechanics within the gameplay, input from controllers, menus, and HUDs and utilizes 2D and 3D art, sound, and physical feedback via controllers (rumble, haptic, force feedback).

More often than not, when people discuss User Interface, they are referring to either menus or HUDs. While these are important to most games, remember that other methods exist and in some cases can offer an easier or clearer user experience, or a more immersive experience. A recent trend in game design is to integrate menus and HUDs into the game world and story, a method called Diegetic Interfaces.



HUDs

A heads-up display, or HUD, is a common method of displaying important game information to players.

Menus

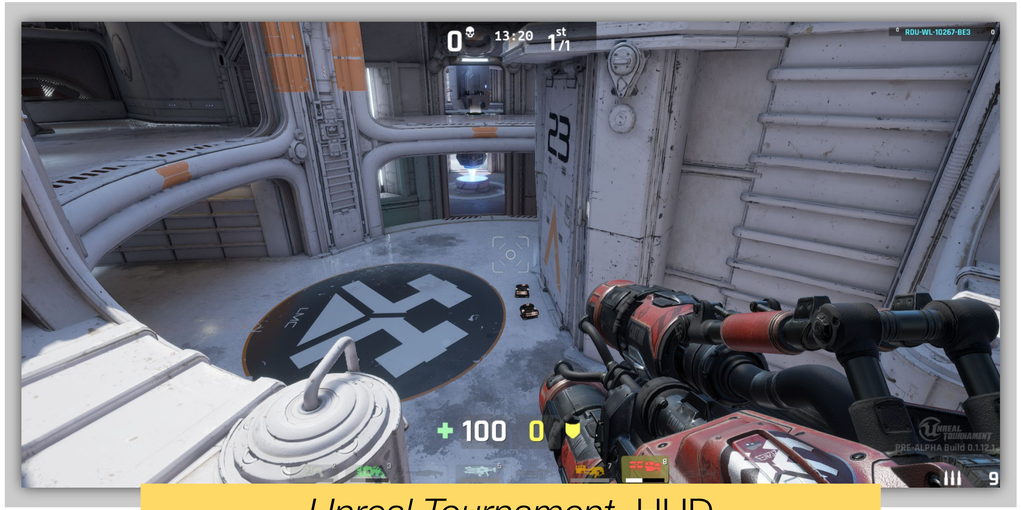
Menus are interactive interfaces that commonly allow players to move between different game states or set various configuration options.

Item Information

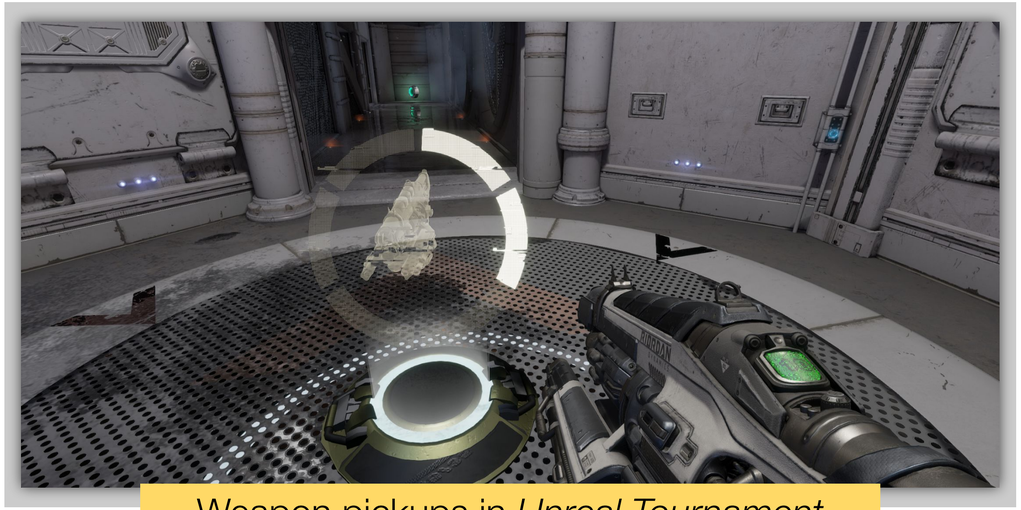
Assets within the world often need to convey information back to the player (for example, how long until the item respawns).

Interactive World Items

Systems such as locks, computer terminals, and conversation dialogue require separate user interface elements to work within the world. These are often driven through the same tools used to create screen user interfaces.



Unreal Tournament HUD



Weapon pickups in *Unreal Tournament*

Screen UI

User Interface elements are typically projected directly to the player's viewport or screen as 2D elements. In the above example, we can see a game HUD with available weapons, health, armor, and other gameplay information.

World UI

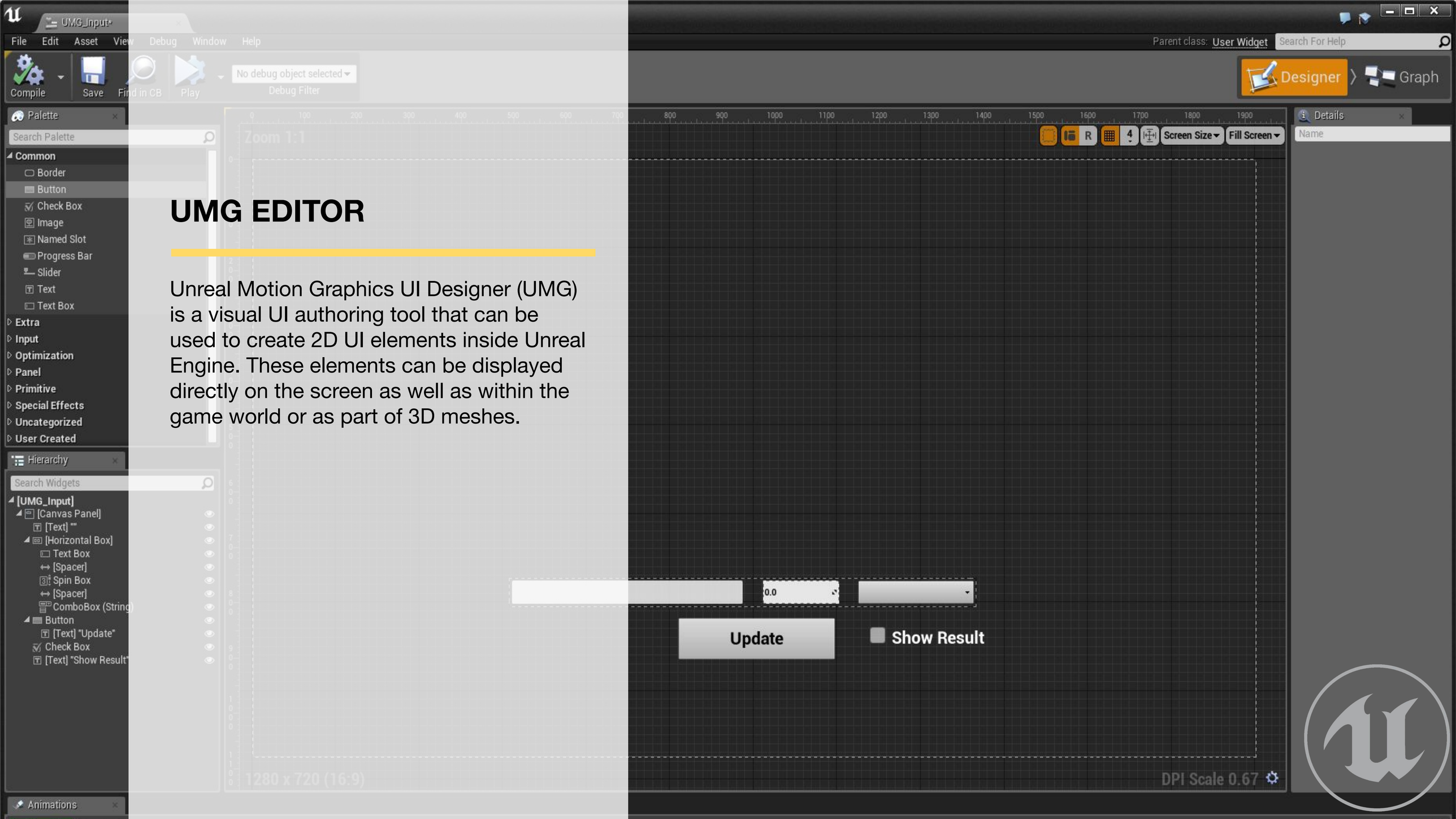
Sometimes UI elements are built into the world. In this example, the time until the weapon respawns is displayed as a UI element on the weapon factory.

UMG

Unreal Motion Graphics UI Designer

UMG EDITOR

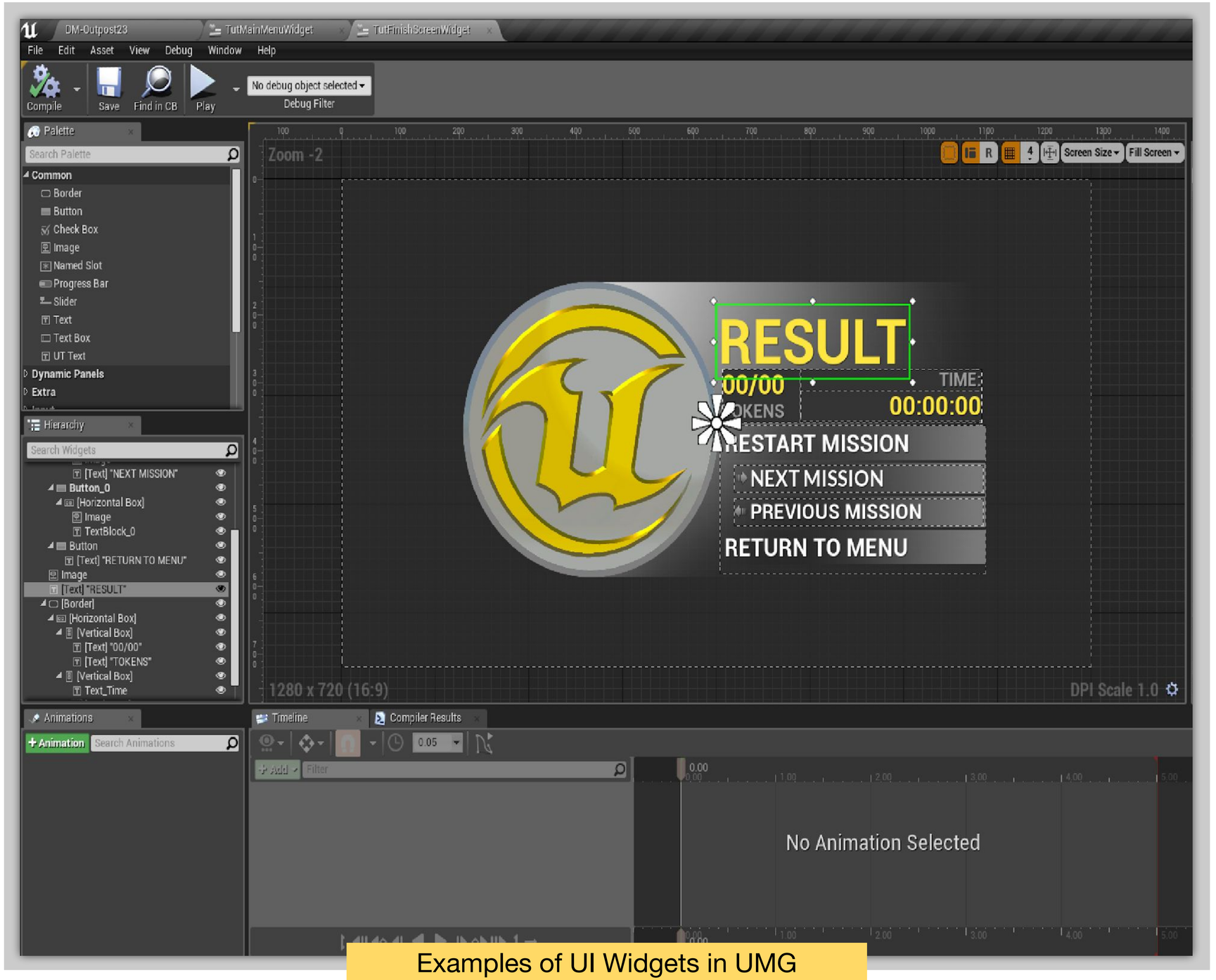
Unreal Motion Graphics UI Designer (UMG) is a visual UI authoring tool that can be used to create 2D UI elements inside Unreal Engine. These elements can be displayed directly on the screen as well as within the game world or as part of 3D meshes.



UMG is the entry point for UI development in Unreal Engine.

The main building blocks of UMG are Widgets. A Widget can be added to another Widget, which can then be added to a third Widget, creating a hierarchical structure. This method of creation is known as composition. When Widget A is added to Widget B, Widget B is now referred to as the *parent* of Widget A, which is referred to as the *child* of Widget B. The parent Widget now can control a range of properties of the child Widget, including where it is rendered, how big it is, and whether or not it is visible.

Not all Widgets can contain other Widgets. To be able to contain another Widget, a Widget must have a *Slot*. A Widget can have a set number of Slots, or allow any number of Slots. A Slot may allow the child Widget to specify some properties about how it would like to be displayed. What those properties are and how they are used are up to the parent Widget, however.



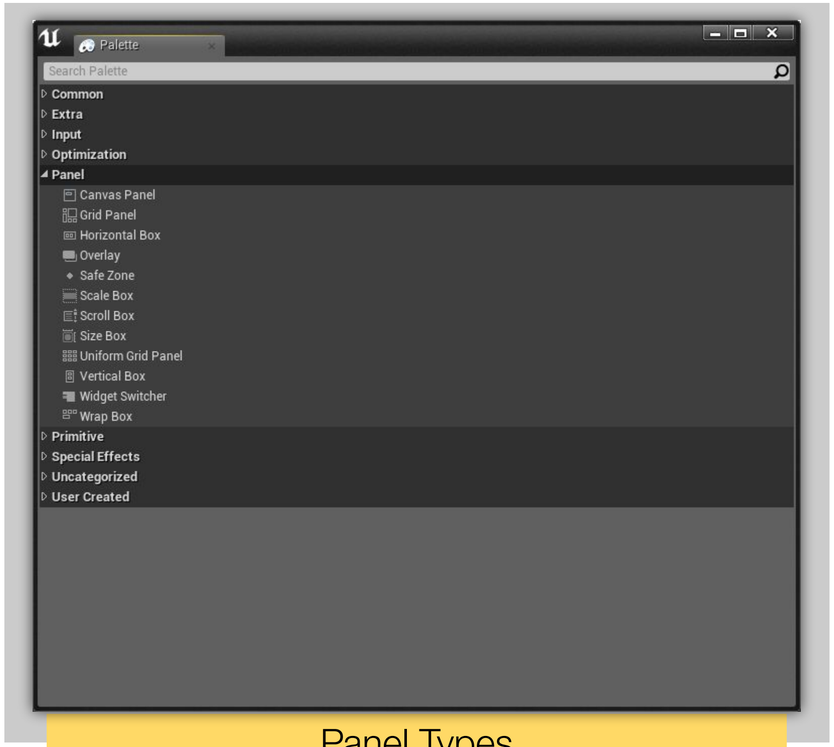
Examples of UI Widgets in UMG

KEY WIDGET TYPES

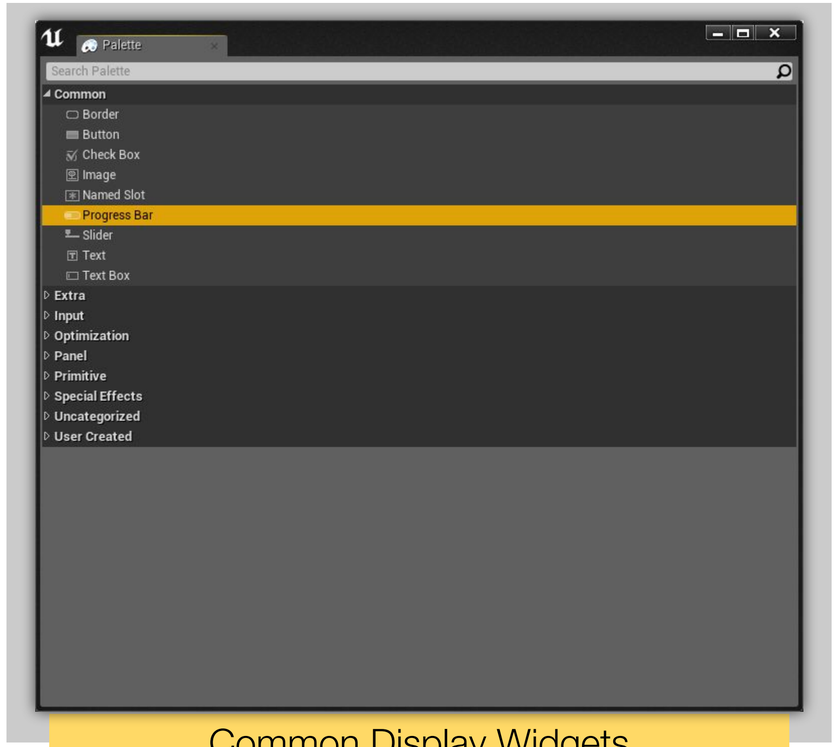
There are many types of pre-built Widgets within UMG.

Some Widgets are designed to display information or take information from players, while others are used to help format and control the layout and positions of Widgets.

- Panels
- Display Widgets
- Input Widgets
- User Widgets



Panel Types



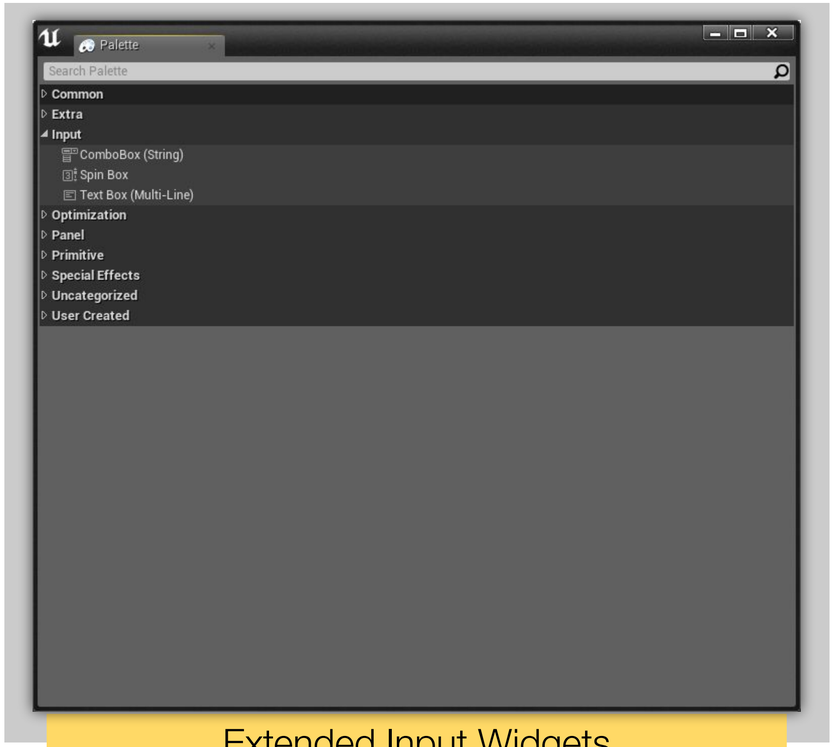
Common Display Widgets

Panels

Panels control the layout and placement of other Widgets.

Display Widgets

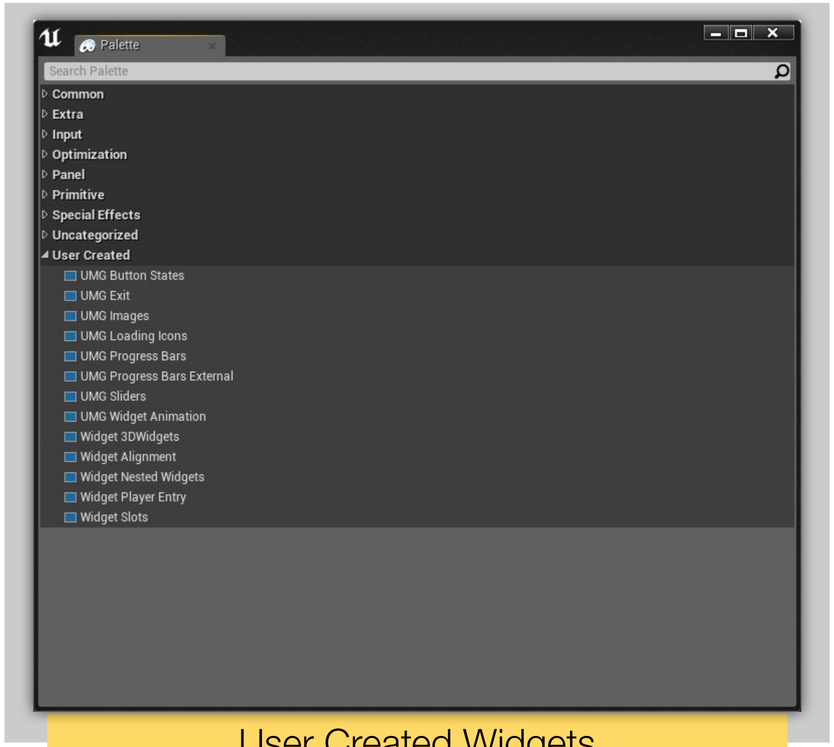
Display Widgets, such as Text and Images, display information to the player. There are a range of pre-built Display Widgets to help with more complex ways of displaying information, such as a Progress Bar.



Extended Input Widgets

Input Widgets

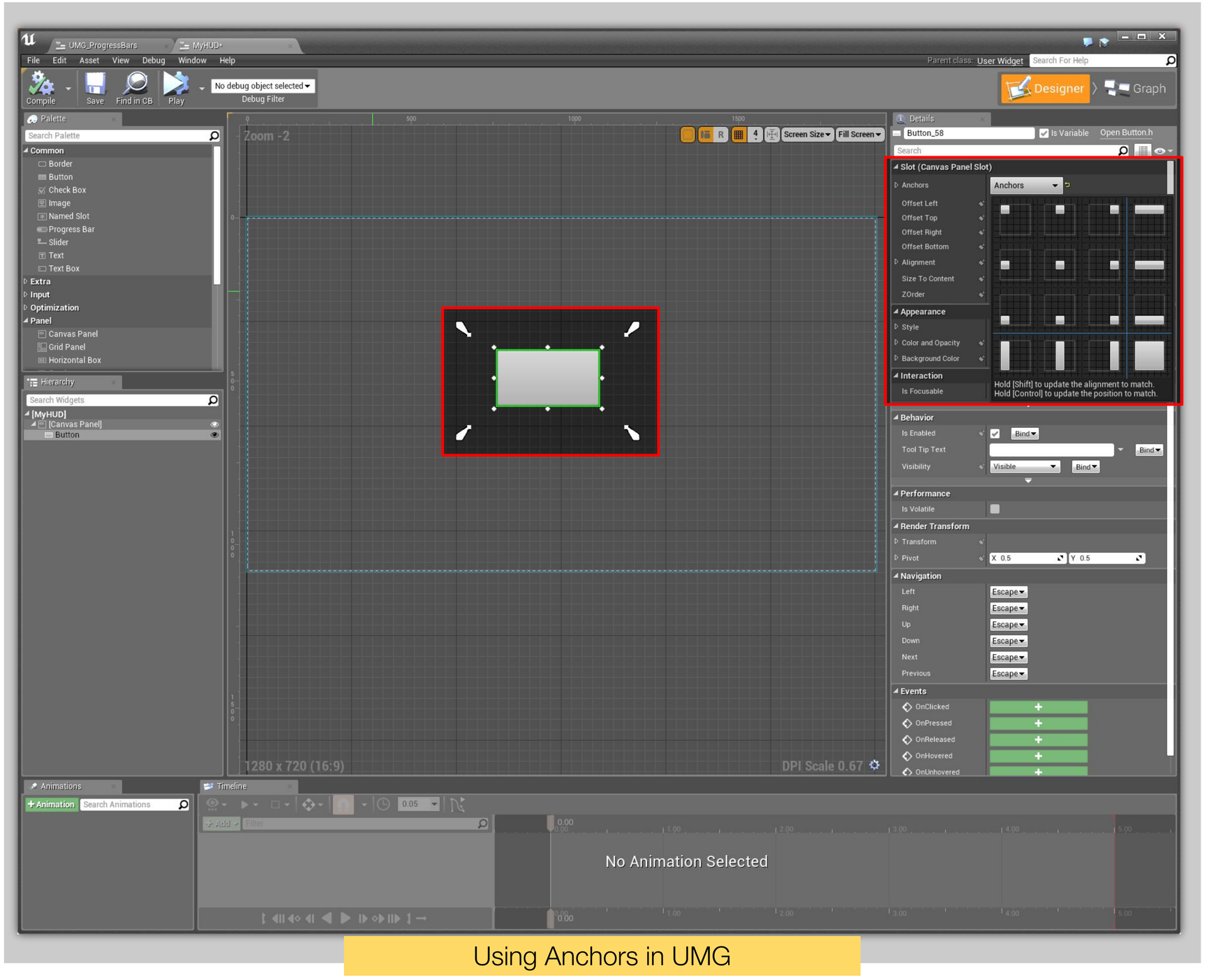
Input Widgets, such as Buttons, Text Boxes, and Combo Boxes, take user input through the keyboard and mouse.



User Created Widgets

User Widgets

User Widgets are Widgets made within UMG, which can be reused many times and placed within other User Widgets.

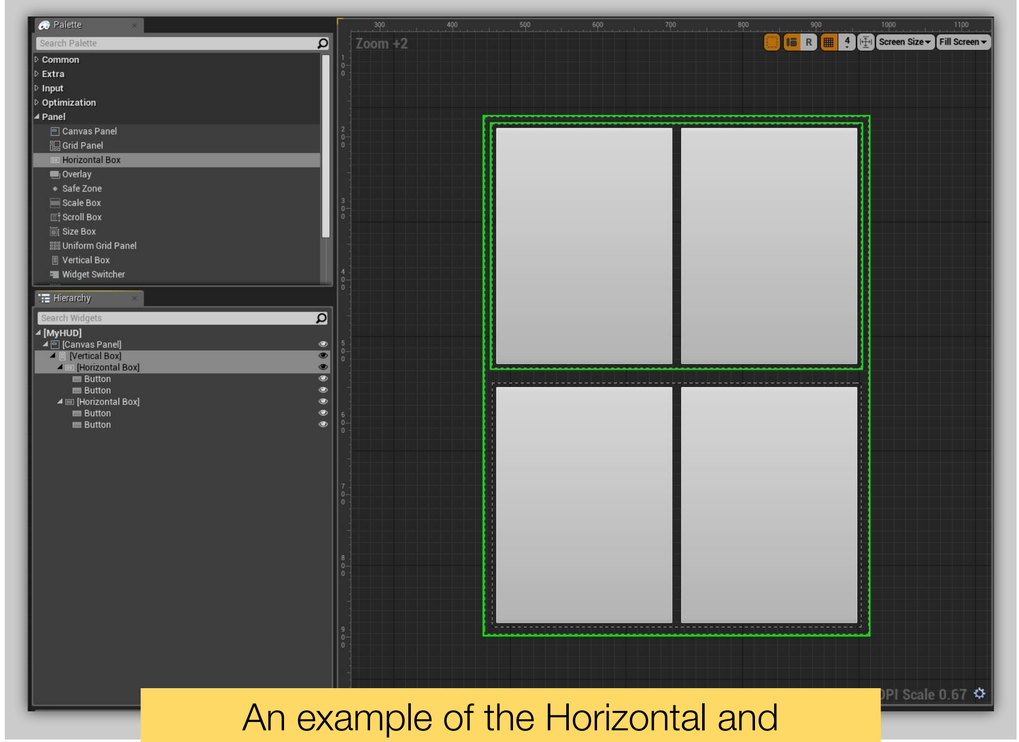


Using Anchors in UMG

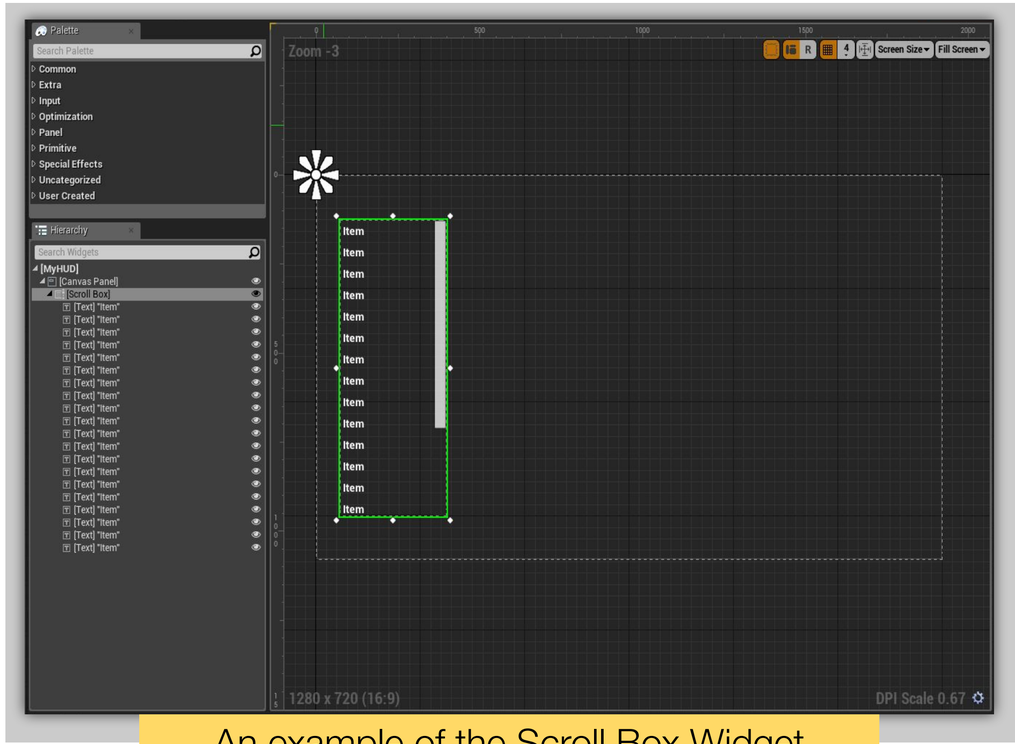
ANCHORS

Anchors are used to define a UI Widget's desired location on a Canvas Panel and maintains that position with varying screen sizes. Anchors are normalized where Min(0,0) and Max(0,0) is the upper left corner and Min(1,1) and Max(1,1) is the bottom right corner.

Once you have a Canvas Panel and add other UI Widgets to it, either you can select from a number of preset Anchor positions (typically, selecting one of these should do if you only want to keep your Widget at a specific location) or you can manually set the Anchor position and Min/Max settings as well as applying offsets.



An example of the Horizontal and Vertical Box Widgets



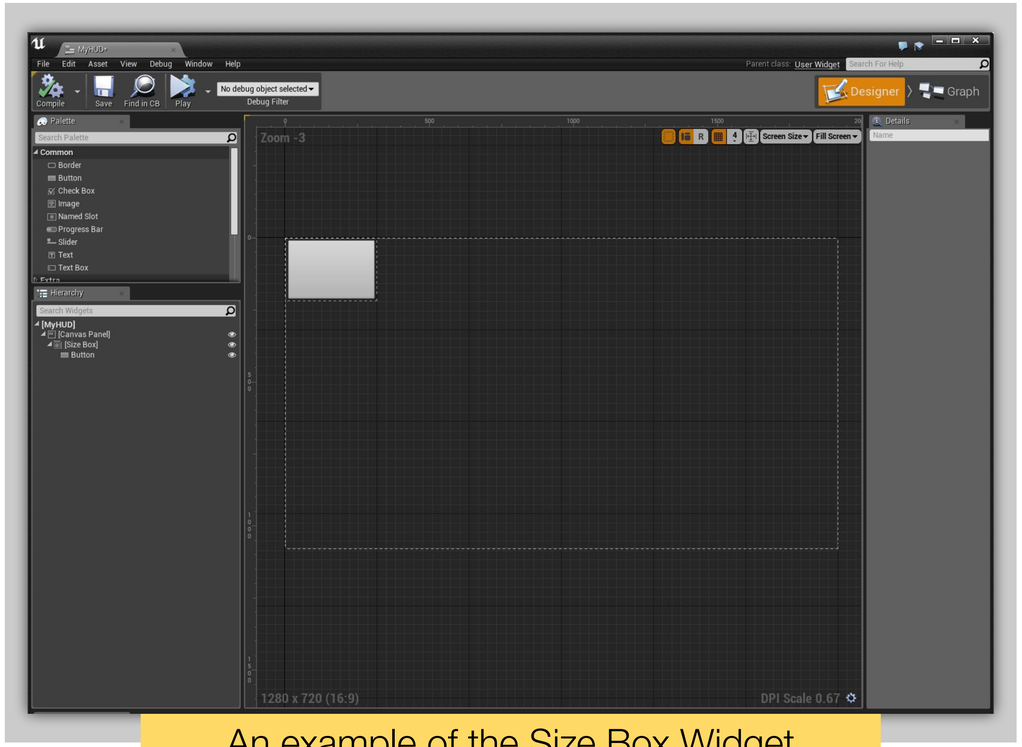
An example of the Scroll Box Widget

Horizontal/Vertical Box

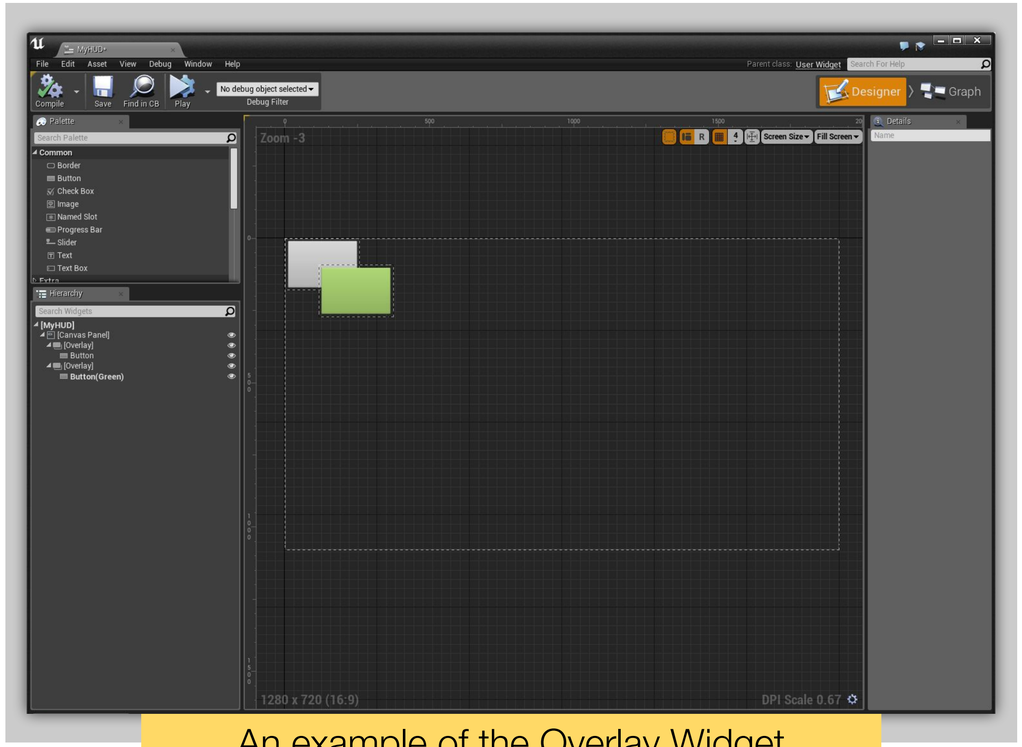
A Horizontal/Vertical Box allows child Widgets to be laid out in a flow horizontally/vertically. The order of the flow will match the order the child Widgets are listed in.

Scroll Box

A Scroll Box is a collection of Widgets than can scroll horizontally or vertically if the space the child Widgets require exceeds the bounds of the scroll box.



An example of the Size Box Widget



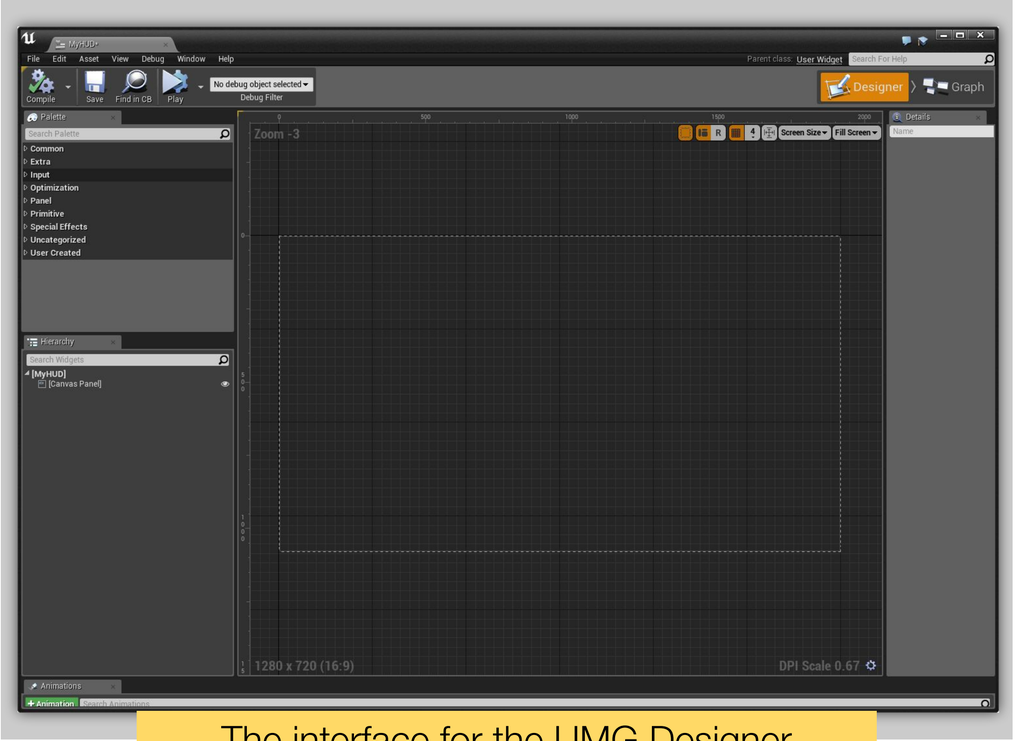
An example of the Overlay Widget

Size Box

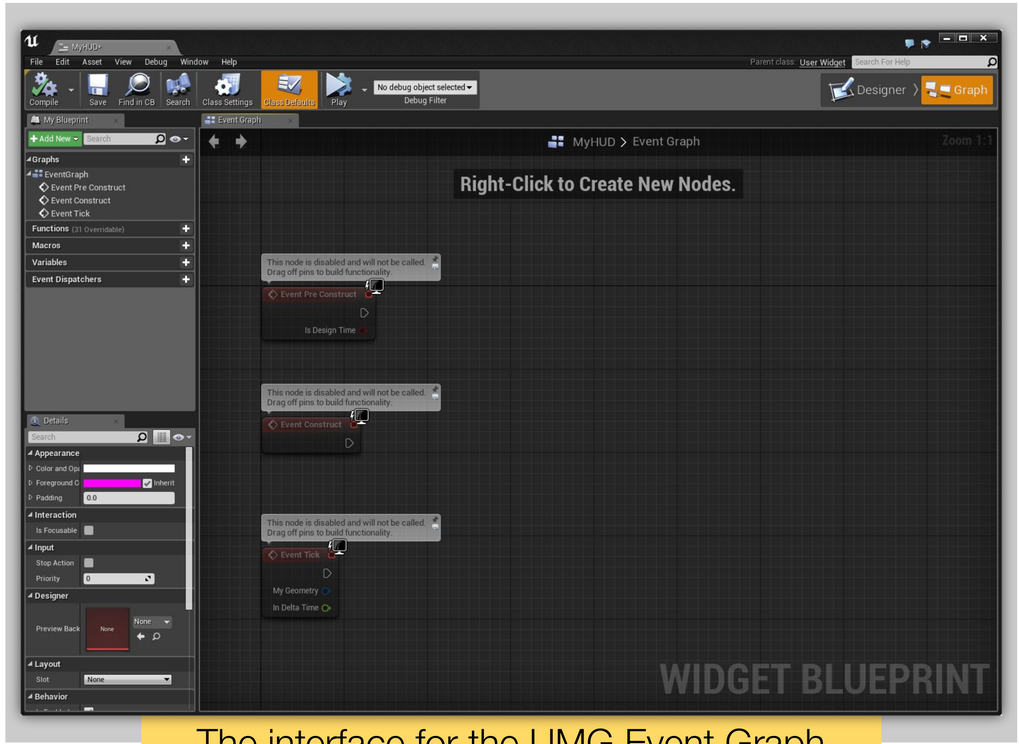
A Size Box contains only a single Slot and forces the contained Widget in that Slot to have a specific size.

Overlay

An Overlay allows child Widgets to be stacked on top of each other and uses a simple flow layout for content on each layer.



The interface for the UMG Designer



The interface for the UMG Event Graph

Designer

The Designer is where Widgets are composited and configured. It provides a “what you see is what you get” (WYSIWYG) editor.

Event Graph

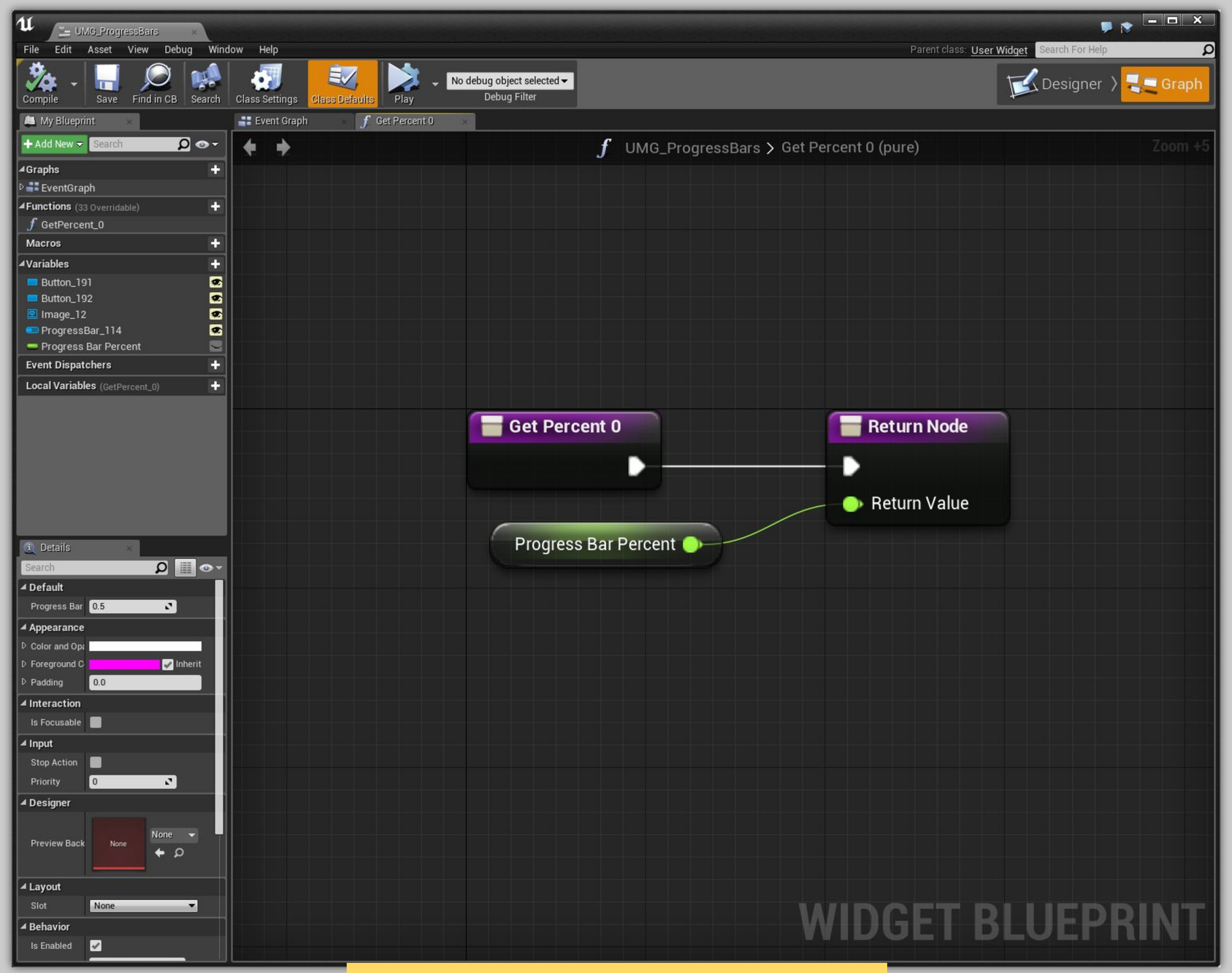
The Event Graph is a Blueprint editor interface for controlling the behavior of Widgets and reacting to input events from players.

Input between game and UI must be carefully directed and controlled.

Focus indicates what component of the game is allowed to capture the input from controllers, mice, and keyboards. For example, a mouse is often used in a game to control aim; however, it is also often used to control the cursor in a menu. Doing both at the same time would be confusing and problematic for the player, so Focus allows a developer to explicitly set what should be using the input.

Once UMG is given input focus, it is still necessary to be even more specific as to which Widget can accept the input events. For example, you can have multiple text entry boxes; which one should the letters being typed using the keyboard appear in? This sort of focus can be set by the user—that is, clicking on that Widget using a cursor—and it can also be set programmatically.

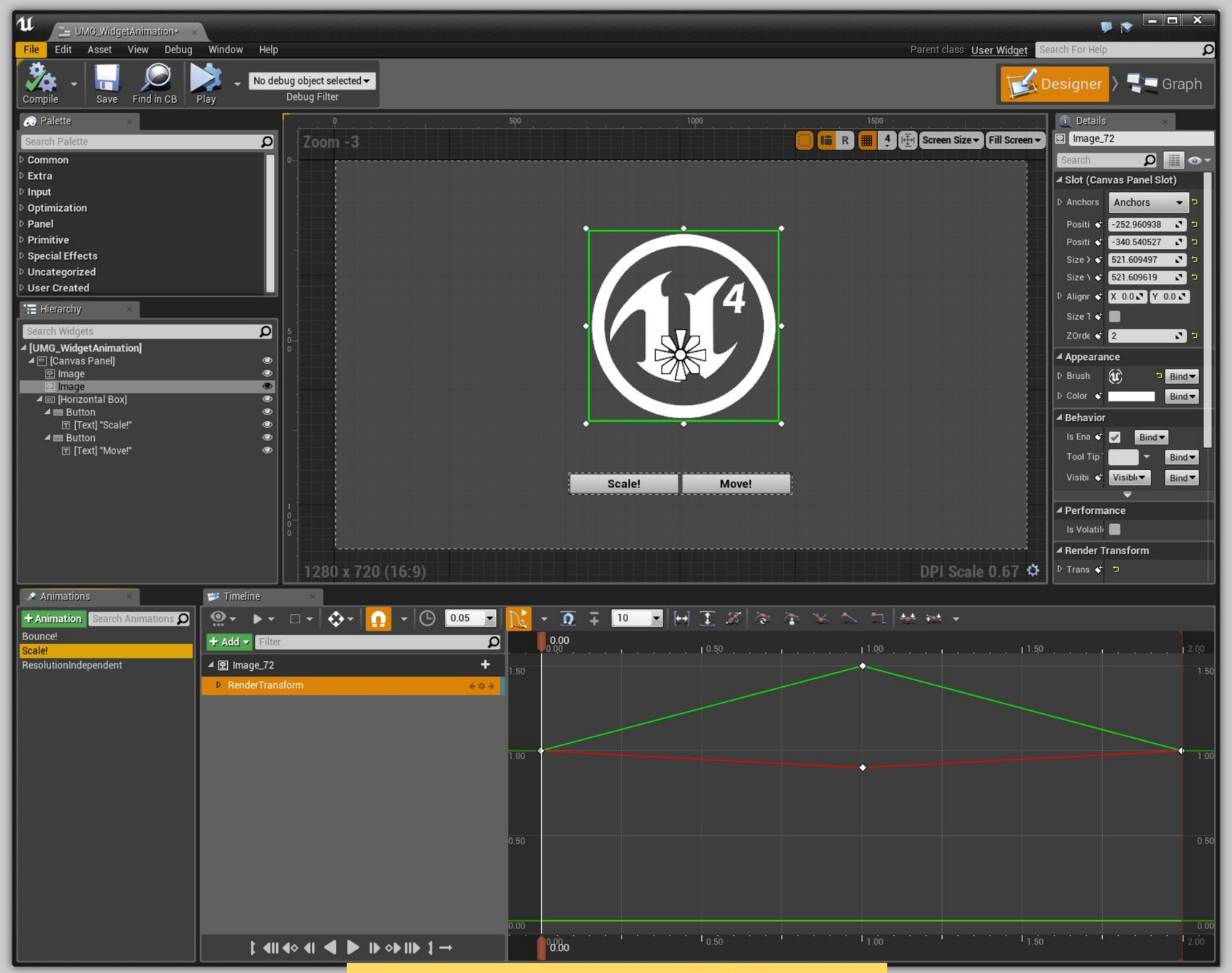
Once a Widget has focus, it can receive input events and define behavior tied to those events using blueprints in the Graph Editor within the UMG Editor.



Property Binding in UMG

BINDING OF PROPERTIES

One of the most useful aspects inside UMG is the ability to bind properties of your Widgets to functions or variables inside the Blueprint. By binding a Widget's property to a function or variable in your Blueprint, anytime that function is called or variable is updated, it will be reflected in the Widget.



Timeline Animation in the UMG Designer

ANIMATION

Located along the bottom of the Widget Blueprint Editor are two windows that allow you to implement and control animations for your UI Widgets.

The first, the Animations window, allows you to create essentially animation tracks, which are used to drive the animation of your Widgets.

The second, the Timeline window, is how an animation is applied to a Widget over time, which is done by placing Keyframes at specified times and then defining how the attached Widget should appear at that Keyframe (this could be size, shape, location, or even coloring options).

CONCLUSION

Questions?