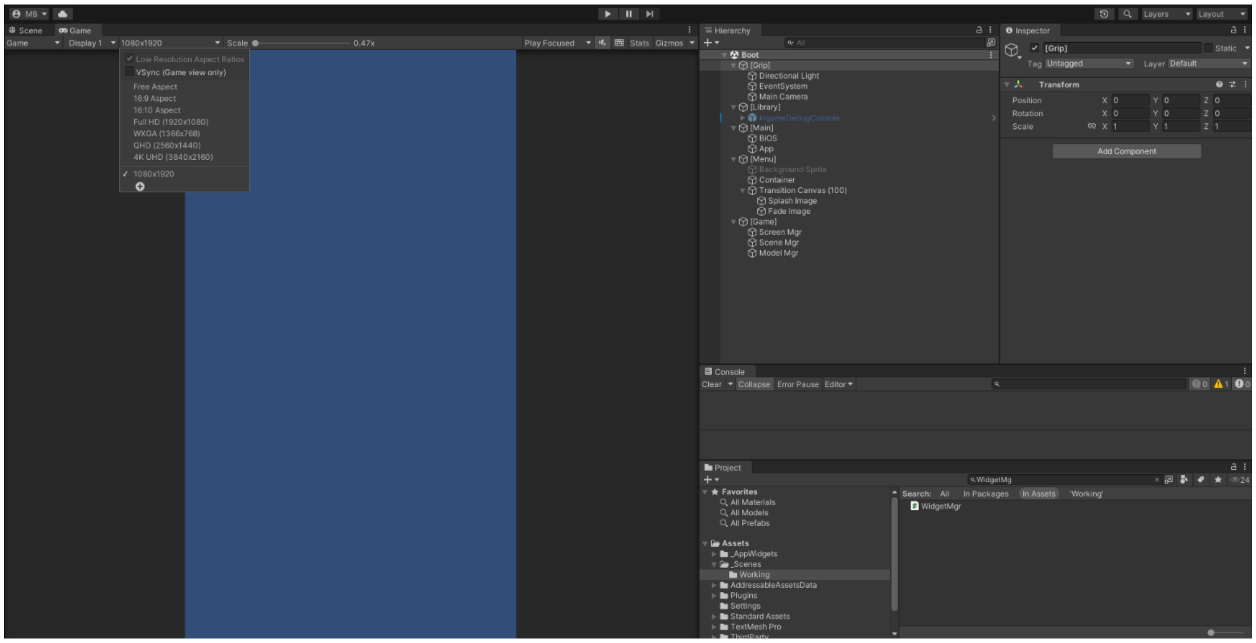
HOLO WERKZ LLC.
WIDGET SERVICES

FOR UNITY DEVELOPERS



Widget Services: Essential Tools for Unity Developers

Widget services are at the heart of a Unity developer's toolkit. They combine key component properties with unique identifiers to streamline the process of connecting user interface components and developing workflows for inter-scene and prefab messaging.

Holo Werkz LLC.

Widget Services

holo.werkz@gmail.com

https://holowerkz.com/

Key Features

- ***Workflow for Sending Data**: Efficiently send data between addressable scenes and addressable prefabs.*
- ***Data Connection Implementation**: Connect user interface components at both edit and runtime.*
- ***Scene and Screen Handlers**: Load and unload addressable scenes and prefabs seamlessly.*
- ***Testing and Automation Interface**: Test and automate using data connections.*

Basic UI Behaviors

Each UI behavior includes a component called a widget, such as Button Widget. The Button Widget handles the heavy lifting by implementing inputs and outputs that connect to UI behavior methods. You can set a text label or receive a click. Messages are sent as objects with GUIDs, decoded by the destination object, and parsed for methodology and data.

System Workflow

1. ***Add Components**: Add a panel, a panel widget, a button, and a button widget.*
2. ***Assign GUIDs**: Assign unique GUIDs to the widgets.*
3. ***Event Handling**: Add a ToWidget component to the button widget to send events to specific widgets by GUID. For example, the button widget can toggle the active state of the panel*

Holo Werkz LLC is pleased to submit this asset for **Widget Services** to support Unity developers in enhancing event communication systems between scenes and instantiated prefabs. We have improved the workflow for adding widgets and setting up events between UI components.

- **Need #1: Methodology to send events between active scenes and additive scenes loaded at run time.**
- **Need #2: Methodology to send events between prefabs instantiated at runtime.**
- **Need #3: Methodology to control addressable scenes and prefab canvas, panels and objects.**

The Widget System is a valuable tool for controlling addressable scenes, prefabs, and widgets, including sending data between scenes and storing objects. Overall, the Widget Services package is committed to improving the developer and user experience through simplicity, interconnectivity, and real-time construction.

Sending Data Between Addressable Scenes and Prefabs

1. Define and build addressable assets.
2. Load and unload addressable assets.
3. Send data between scenes and to loaded prefabs.
4. Use unique identifiers for data table keying.

Connecting UI Components at Edit and Runtime

1. Add widget components and list them by behavior type.
2. Automatically assign GUIDs and definitions in the editor or at runtime.
3. Send data to widgets and save data between sessions.
4. Allow runtime-created widgets to send and filter object data.

Scene and Screen Handlers

1. Use addressable scene handlers to load builds independently at runtime.
2. Define panels to load and switch at runtime using screen handlers.

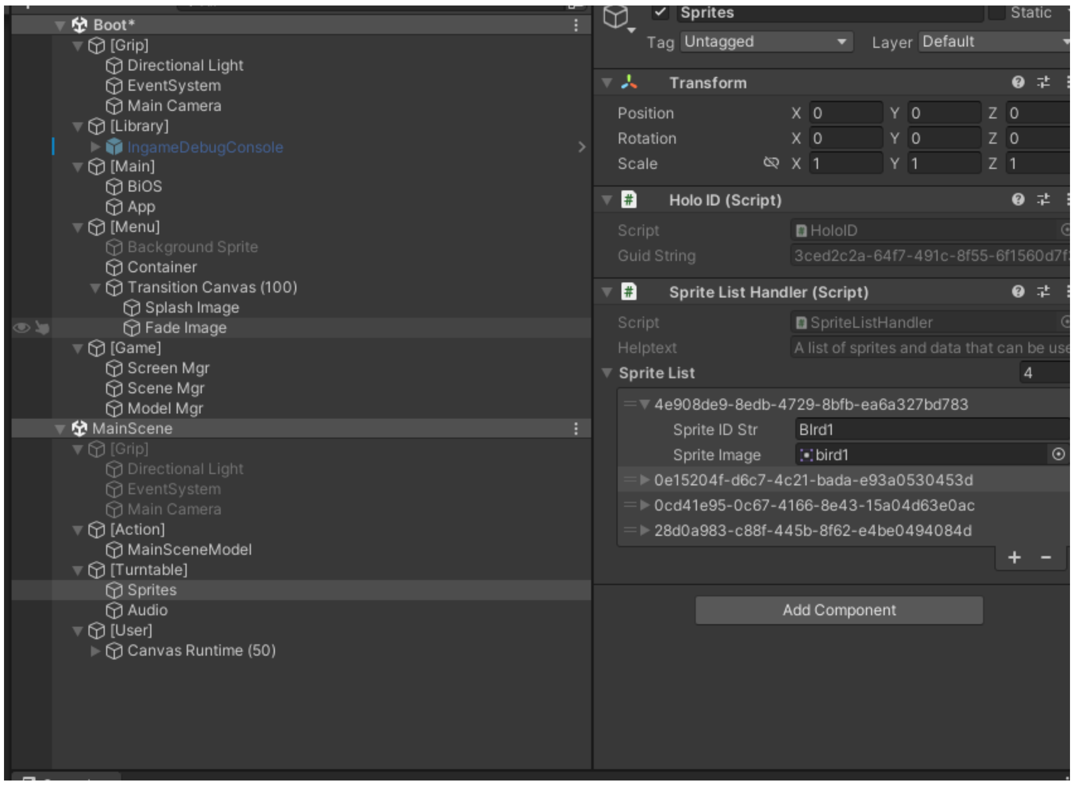
Interface for Testing and Automation

1. Use mock data to simulate user experiences.
2. Automate the triggering of events for testing.

Top Behaviors and Widgets

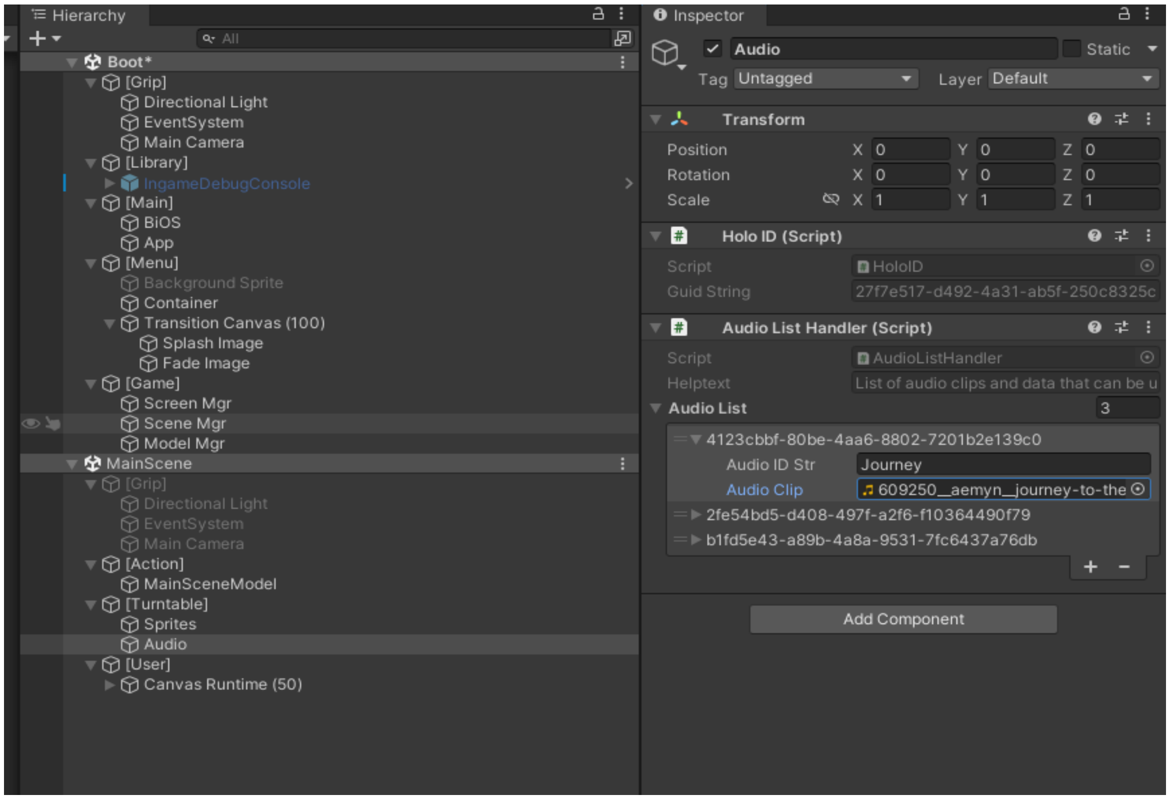
- **Root Standard**: Implements the main static interface handles to non-static MonoBehaviours.
- **Root Widget**: Processes widget lists for the editor and runtime. Manages event listeners as widgets are enabled and disabled. Includes methods to add the correct widget based on components on the Game Object.
- **Main Bus**: Creates a registry for app widgets and passes event triggers and data to the identified widget.
- **Main Signal**: Creates a non-volatile repository to store objects by GUID. Widgets can pass data to other widgets using a combination of the widget's GUID and the data object's GUID in the signal repository.
- **Model View Data**: Creates a base view controller to connect model and data classes based on a template. Using polymorphism, any simple item can be stacked on the view and retrieved by derived models and items.
- **Widget Manager**: Handles the dictionaries for widgets registered for activation and data events.
- **Canvas Widget**: Manages screen orientation by activating the left/right or top/bottom cells in the view. It has a widget trigger list that it can send on enable or disable.
- **Panel Widget**: Derived from the Object Widget, it allows the panel to create slide animations to the left, right, top, or bottom.
- **Object Widget**: Extends the App Widget base type with a type and definition. It can be used as a utility widget for attaching Game Objects.

- **Sprite List Handler**: Maintains a dictionary of textures that can be accessed for images.



```
Sprites                                   Static
Tag Untagged            Layer Default
Transform
Position    X 0    Y 0    Z 0
Rotation    X 0    Y 0    Z 0
Scale       X 1    Y 1    Z 1
Holo ID (Script)
Script          HoloID
Guid String     3ced2c2a-64f7-491c-8f55-6f1560d7f
Sprite List Handler (Script)
Script          SpriteListHandler
Helptext        A list of sprites and data that can be use
Sprite List                               4
  4e908de9-8edb-4729-8bfb-ea6a327bd783
    Sprite ID Str   BIrd1
    Sprite Image    bird1
  0e15204f-d6c7-4c21-bada-e93a0530453d
  0cd41e95-0c67-4166-8e43-15a04d63e0ac
  28d0a983-c88f-445b-8f62-e4be0494084d
```

- **Audio List Handler**: Maintains a dictionary of audio clips that can be accessed for sound effects.



```
Audio                                     Static
Tag Untagged            Layer Default
Transform
Position    X 0    Y 0    Z 0
Rotation    X 0    Y 0    Z 0
Scale       X 1    Y 1    Z 1
Holo ID (Script)
Script          HoloID
Guid String     27f7e517-d492-4a31-ab5f-250c8325c
Audio List Handler (Script)
Script          AudioListHandler
Helptext        List of audio clips and data that can be u
Audio List                                3
  4123cbbf-80be-4aa6-8802-7201b2e139c0
    Audio ID Str    Journey
    Audio Clip      609250__aemyn__journey-to-the
  2fe54bd5-d408-497f-a2f6-f10364490f79
  b1fd5e43-a89b-4a8a-9531-7fc6437a76db
```

-

UI Behaviors and Widgets

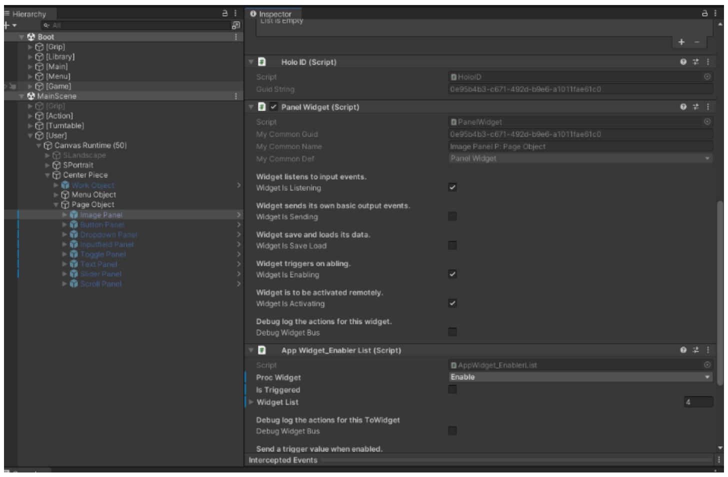
App Widget

- **Base class widget**: Creates the identity, sends triggers to animations, audio, scenes, screens, or other widgets. Handles messages for activation, enabling, showing, or interaction of the widgets.
    - Widget is Listening
    - Widget is Sending
    - Widget Save and Loads
    - Widget triggers on Enabling
    - Widget is remotely Activated
    - Debug Log Actions for this widget

SendMsg Methods

These methods on the base class and the SendMsg methods on derived widget are inputs to the Widgets attached UI class like Image or Button. The SendMsg acts as an output from the UI Component through the Widget.

- **Active State (bool)**: Sets the active state of the object.
- **Active Toggle()**: Toggles the active state of the object.
- **Enabled Object (bool)**: Calls the Enable Object method on the widget.
- **Enabled Toggle()**: Calls the Enable Toggled method on the widget.
- **Interactive Object (bool)**: Sets the interactive state on the Selectable Component.
- **Interactive Toggle()**: Toggles the interactive state on the Selectable Component.
- **Show Object (bool)**: Calls the Show/Hide method on the widget.
- **Show Toggle()**: Toggles the Show/Hide method on the widget.



Holo ID (Script)

Script: HoloID

Guid String: 0e95b4b3-c671-492d-b9e6-a1011fae61c0

Panel Widget (Script)

Script: PanelWidget

My Common Guid: 0e95b4b3-c671-492d-b9e6-a1011fae61c0

My Common Name: Image Panel P: Page Object

My Common Def: Panel Widget

Widget listens to input events.
Widget Is Listening ✓

Widget sends its own basic output events.
Widget Is Sending ☐

Widget save and loads its data.
Widget Is Save Load ☐

Widget triggers on abling.
Widget Is Enabling ✓

Widget is to be activated remotely.
Widget Is Activating ✓

Debug log the actions for this widget.
Debug Widget Bus ☐

App Widget_Enabler List (Script)

Script: AppWidget_EnablerList

Proc Widget: Enable

Is Triggered ☐

Widget List: 4

Debug log the actions for this ToWidget
Debug Widget Bus ☐

Send a trigger value when enabled.

Intercepted Events

The Panel Widget is an Example of the shared properties on the App Widget.

Button Widget

- **OnButtonClick: Send Trigger (string)**: Sends the label of the button on click.
- **SendMsg: Set Label (string)**: Sets the label of the button when triggered.

Dropdown Widget

- **OnValueChanged: Send Value Trigger (int)**: Sends the value of the dropdown selection.
- **OnValueChanged: Send Trigger (string)**: Sends the literal of the dropdown selection.
- **SendMsg: Add Option (string)**: Adds a single string option to the dropdown list.
- **SendMsg: Add Option List (List<string>)**: Adds a list of string options to the dropdown list.
- **SendMsg: Clear Options()**: Clears the options from the dropdown list.
- **SendMsg: Set Value (int)**: Sets the selected value of the dropdown list.

Image Widget

- **SendMsg: Set Value (int)**: Sets the texture of the sprite by indexing the referenced Sprite List.
- **SendMsg: Set Value (Texture2D)**: Sets the texture of the sprite by passing the referenced Texture2D.

Input Field Widget

- **OnValueChanged: Send Trigger (string)**: Sends the string value when changed.
- **OnEndEdit: Send Trigger (string)**: Sends the string value on the end edit event.
- **OnSelected: Send Trigger (string)**: Sends the string value when selected.
- **OnDeselected: Send Trigger (string)**: Sends the string value when deselected.
- **OnSubmit: Send Trigger (string)**: Sends the string value when the submit event is triggered.
- **SendMsg: Set Placeholder (string)**: Sets the input field's placeholder.
- **SendMsg: Set Label (string)**: Sets the input field's text value.

Raw Image Widget

- **SendMsg: Set Value (int)**: Sets the texture of the sprite by indexing the referenced Sprite List.
- **SendMsg: Set Value (Texture2D)**: Sets the texture of the sprite by passing the referenced Texture2D.

Scroll Rect Widget

- Extends the View Controller with a Model and a View. The UI Components on the scroll rect utilize an object pool. Deriving new data objects from the base Simple Item allows for the standard Scroll Rect widget to handle any model and view. This connects the UI Component properties to data in scripts.

  View Controller<TC, TVM> where TC: class where TVM : View Model<TC>

  View Controller<Item Simple Data, Item Simple> viewController.

Slider Widget

- **OnValueChanged: Send Value Trigger (float)**: Sends the value of the slider.
- **SendMsg: Set Value (float)**: Sets the value of the slider.

Text Widget

- **SendMsg: Set Text (string)**: Sets the literal of the text component.

Toggle Widget

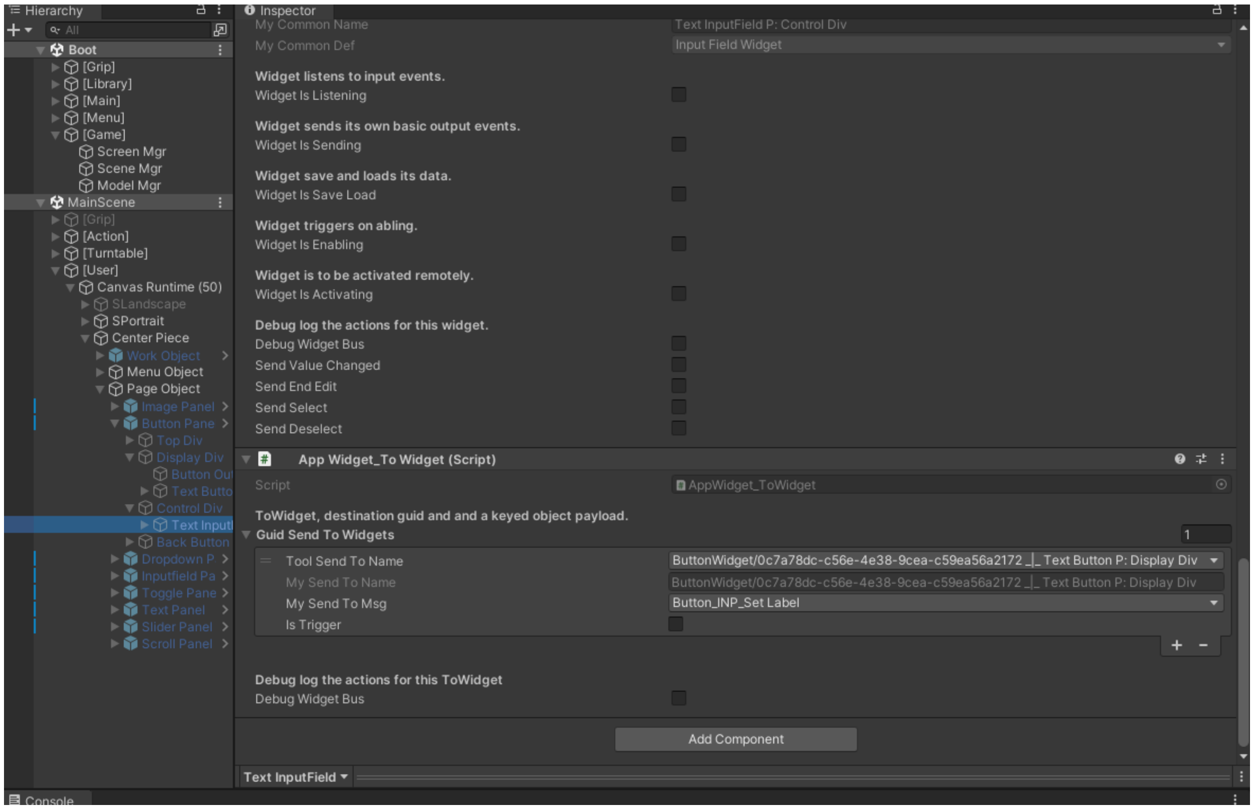
- **OnValueChanged: Send Value Trigger (bool)**: Sends the value of the toggle when switched.
- **SendMsg: Set Is On (bool)**: Sets the “is on” state of the toggle.
- **SendMsg: Switch()**: Toggles the “is on” state of the switch.
- **SendMsg: Set Label (string)**: Sets the literal for the toggle UI text component.

Utility Behaviors and Widgets

- **Animation Widget**: This widget lerps a rectangle transform off-screen in the left, right, top, or bottom directions.
    - Slide Time
    - Slide Motion
    - Rect Transform
    - Disable View
- **Animator Widget**: Sends the slide time and motion to a Panel GUID and initiates the animation. When called, the animator overrides the default values with those provided.
    - Slide Time
    - Slide Motion
    - List of GUIDs
- **Audio Widget**: Calls the Audio Controller to play the audio clip as the main effect when enabled or added as a one-shot overlay. The widget triggers audio events using a literal lookup of the audio clip name and the audio list.
    - Audio Name
    - List of Audio Handles
    - Play On Enable
    - Play One Shot
- **Fade Widget**: Controls the splash screen display and a black overlay for fading. Supports fading in, out, in-out, and out-in combinations. Requires references to the splash and fade images.
    - Fade Type
    - Fade Time
- **Scene Widget**: Triggered by the Button Widget to control an addressable scene, either loading (on) or unloading (off) it.
    - Scene Name
    - Scene State

- **Screen Widget**: Loads, switches, and unloads prefab addressable assets as UI panels. Managed by an enumeration identifier.
    - Screen Prefab
    - Screen State
        - On Others Change No
        - On Others Change Off
        - Off Others Change No
        - Off Others Change Off
        - Toggle Others Change No

- **To Widget**: Sends output events and data to a list of other widgets by GUID. The send value functions are triggered by UI component events, which are processed by the App Widget to harvest data and send messages.
    - List of Send To Widgets
        - Send To Name (Component Type/GUID/Game Object Name/Parent)
        - Send To Msg (Key message and Data Struct)



My Common Name: Text InputField P: Control Div
My Common Def: Input Field Widget

Widget listens to input events.
Widget Is Listening

Widget sends its own basic output events.
Widget Is Sending

Widget save and loads its data.
Widget Is Save Load

Widget triggers on abling.
Widget Is Enabling

Widget is to be activated remotely.
Widget Is Activating

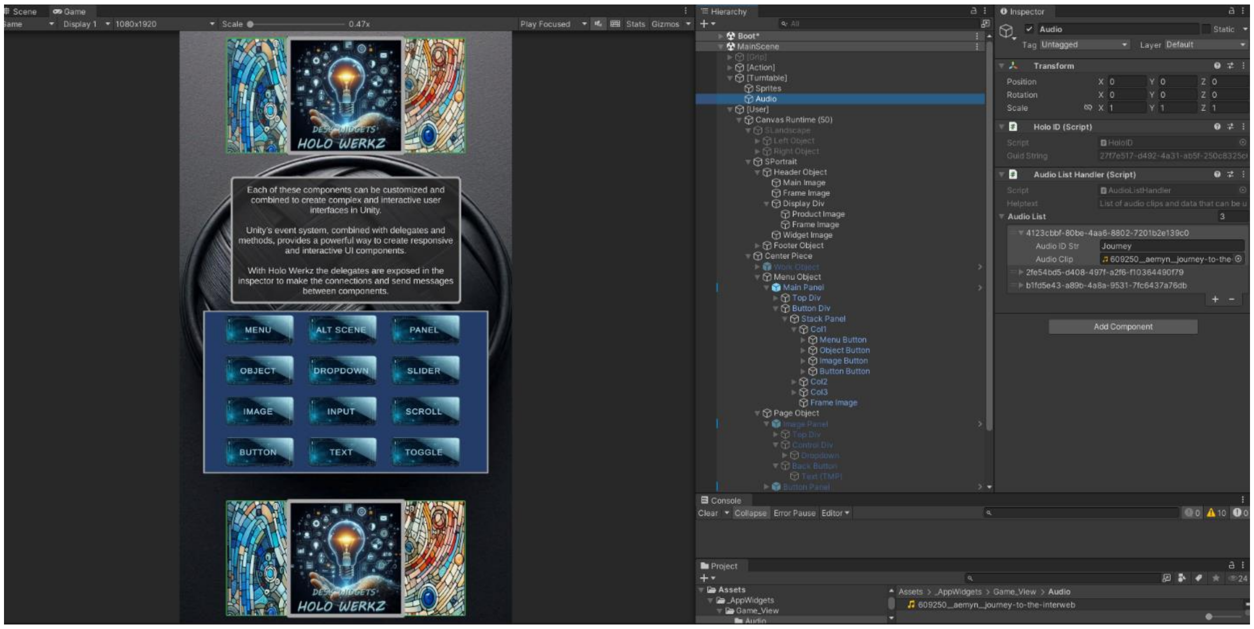
Debug log the actions for this widget.
Debug Widget Bus
Send Value Changed
Send End Edit
Send Select
Send Deselect

App Widget_To Widget (Script)
Script: AppWidget_ToWidget

ToWidget, destination guid and and a keyed object payload.
Guid Send To Widgets: 1
Tool Send To Name: ButtonWidget/0c7a78dc-c56e-4e38-9cea-c59ea56a2172 _|_ Text Button P: Display Div
My Send To Name: ButtonWidget/0c7a78dc-c56e-4e38-9cea-c59ea56a2172 _|_ Text Button P: Display Div
My Send To Msg: Button_INP_Set Label
Is Trigger

Debug log the actions for this ToWidget
Debug Widget Bus

- **Trigger List Widget**: Sends messages to a list of App Widgets.
    - Widget Change State Proc (Activate, Enable, Interact, Show)
    - Is Triggered
    - App Widget List (Game Object References)
- **Enabler List Widget**: Triggered when the widget is enabled or disabled. You can set the trigger to fire for each method and specify the Boolean payload for the trigger.
    - Widget Change State Proc (Activate, Enable, Interact, Show)
    - Is Triggered
    - App Widget List (Game Object References)



Each of these components can be customized and combined to create complex and interactive user interfaces in Unity.

Unity's event system, combined with delegates and methods, provides a powerful way to create responsive and interactive UI components.

With Holo Werkz the delegates are exposed in the inspector to make the connections and send messages between components.

```
Audio                                   Static
Tag Untagged              Layer Default
Transform
Position        X 0      Y 0      Z 0
Rotation        X 0      Y 0      Z 0
Scale           X 1      Y 1      Z 1
Holo ID (Script)
Script          HoloID
Guid String     27f7e517-d492-4a31-ab5f-250c8325c
Audio List Handler (Script)
Script          AudioListHandler
Helptext        List of audio clips and data that can be u
Audio List                                   3
  4123cbbf-80be-4aa6-8802-7201b2e139c0
    Audio ID Str    Journey
    Audio Clip      609250__aemyn__journey-to-the-
  2fe54bd5-d408-497f-a2f6-f10364490f79
  b1fd5e43-a89b-4a8a-9531-7fc6437a76db
```

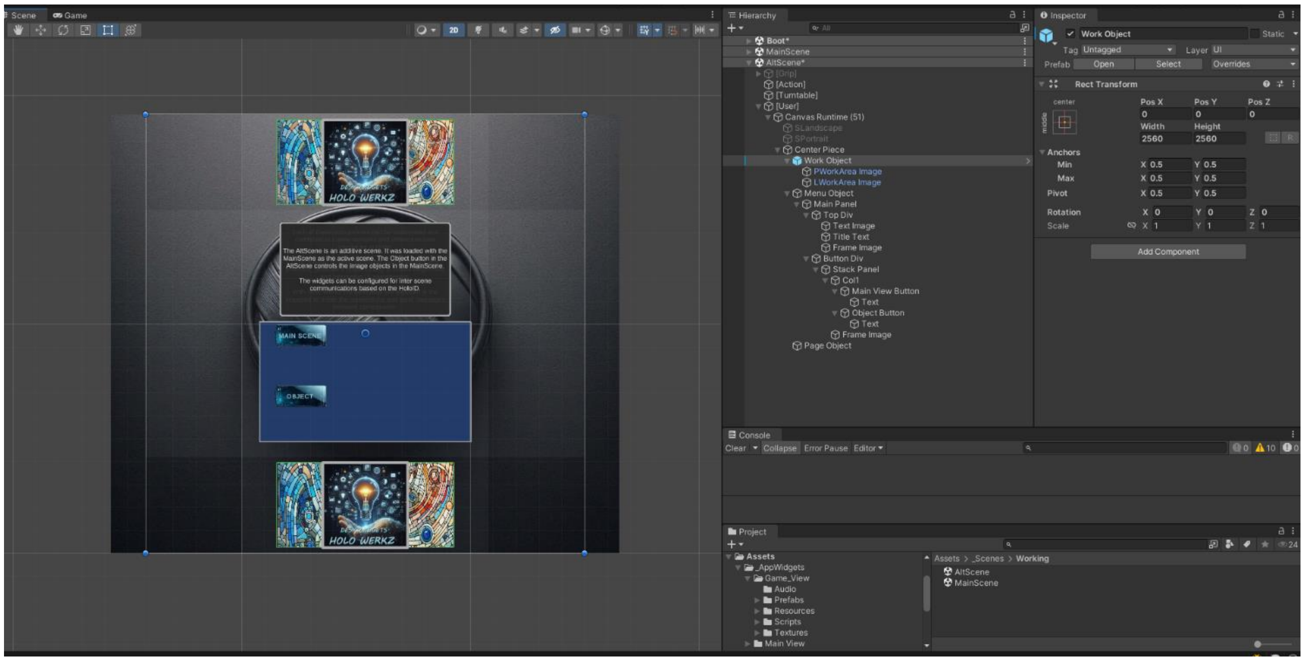
Basic Examples

Example Usage – UI Panel

1. **Create UI Panel**: Add a panel and a Panel Widget.
   - Set the panel to listening.
   - Set the panel to activate.
2. **Create UI Button**: Add a button and a Button Widget to the scene.
   - Add a ToWidget to the button.
   - Add one Guid Sent To Widget property.
   - Set the drop-down value to the Panel object you just created.
     - **Note**: As you add more widgets, this drop-down value will change. The concrete value set by the drop-down is stored below in a read-only label. This read-only value is the GUID being used.
   - Set the enumeration of the message to be sent. In this case, set Widget_Active_State_Trigger.
   - Leave the trigger value unchecked to deactivate the panel when the button is clicked.

Example Usage – Alt Scene

1. **Create Previous Example in Alternate Scene**: Add a second panel and button then set the messages in the alt scene to control objects in the active main scene.



The AltScene is an additive scene. It was loaded with the MainScene as the active scene. The Object button in the AltScene controls the image objects in the MainScene.

The widgets can be configured for inter scene communications based on the HoloID.

Work Object
Tag Untagged Layer UI
Prefab Open Select Overrides

	Pos X	Pos Y	Pos Z
	0	0	0
	Width	Height	
	2560	2560	
Anchors Min	X 0.5	Y 0.5	
Anchors Max	X 0.5	Y 0.5	
Pivot	X 0.5	Y 0.5	
Rotation	X 0	Y 0	Z 0
Scale	X 1	Y 1	Z 1

Example Usage – UI Dropdown Panel

1. **Create UI Panel**: Add a panel and a Panel Widget, then arrange the following components:
   - Add a Text Component and a Text Widget.
   - Add a Dropdown Component and a Dropdown Widget.
   - Add an Input Field Component and an Input Field Widget.
   - Add a Button Component and a Button Widget.
2. **Configure the Button**:
   - Label it “Clear”.
   - Add a ToWidget.
   - On the ToWidget, create one SendToGuid for the Dropdown Widget with a message of Dropdown_INP_Clear_Option.
3. **Configure the Input Field**:
   - Add a ToWidget.
   - On the ToWidget, create one SendToGuid for the Dropdown Widget with a message of Dropdown_INP_Add_Option.
   - On the Input Field Widget, check the box to Send End Edit.
4. **Configure the Dropdown**:
   - On the Dropdown Widget, check the box to listen for input events.
   - Add a ToWidget.
   - On the ToWidget, create one SendToGuid for the Text Widget with a message of Text_INP_Set_Text

CONCLUSION

We look forward to working with Unity Developers and supporting your efforts to improve your skills; working with User Interface Components and more scene objects. We are confident that with this simple framework you will save time creating user experiences in the editor and at runtime.

If you have questions on this documentation, feel free to contact Holo Werkz by email. We will be in touch with you in the next one to three business days.

Thank you for your consideration,

Holo Werkz LLC.

holo.werkz@gmail.com