Table of Contents

Quick-start

If you’re looking for a quick implementation of this quest menu into your game (ie, your game already has the data layer implemented, but just looking to present the data in this quest screen), then create an instance of the WS_Widget and add it to your viewport.

For a robust data-driven MVVM approach to implementing this menu, please read on.

What is MVVM

MVVM stands for Model-View-ViewModel, a software design pattern used in building user interfaces.

MVVM helps keep the code organized by separating the data (Model) from how it's displayed (View), and the View Model helps in coordinating the communication between them.

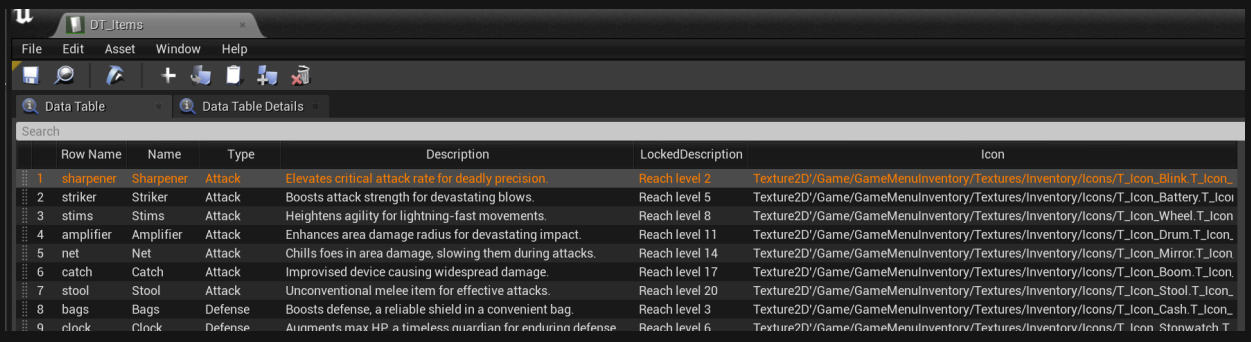
Model

In this package, the DataTables act as the data (Model).

They are located in **`GameMenuInventory/DemoAssets/Data/DT_*.`**

DT_Items

This is the items data. Each row represents one unique item.
It is defined by an ID (row name), name, type (item category), item description, a locked description, and an item icon.



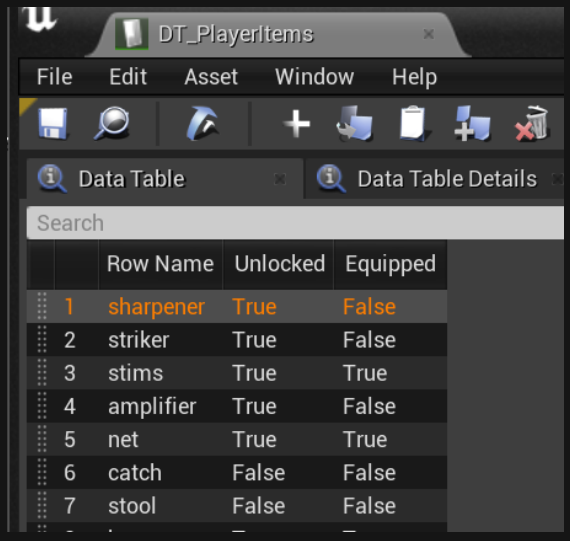
	Row Name	Name	Type	Description	LockedDescription	Icon
1	sharpener	Sharpener	Attack	Elevates critical attack rate for deadly precision.	Reach level 2	Texture2D'/Game/GameMenuInventory/Textures/Inventory/Icons/T_Icon_Blink.T_Icon_
2	striker	Striker	Attack	Boosts attack strength for devastating blows.	Reach level 5	Texture2D'/Game/GameMenuInventory/Textures/Inventory/Icons/T_Icon_Battery.T_Icon
3	stims	Stims	Attack	Heightens agility for lightning-fast movements.	Reach level 8	Texture2D'/Game/GameMenuInventory/Textures/Inventory/Icons/T_Icon_Wheel.T_Icon
4	amplifier	Amplifier	Attack	Enhances area damage radius for devastating impact.	Reach level 11	Texture2D'/Game/GameMenuInventory/Textures/Inventory/Icons/T_Icon_Drum.T_Icon_
5	net	Net	Attack	Chills foes in area damage, slowing them during attacks.	Reach level 14	Texture2D'/Game/GameMenuInventory/Textures/Inventory/Icons/T_Icon_Mirror.T_Icon
6	catch	Catch	Attack	Improvised device causing widespread damage.	Reach level 17	Texture2D'/Game/GameMenuInventory/Textures/Inventory/Icons/T_Icon_Boom.T_Icon_
7	stool	Stool	Attack	Unconventional melee item for effective attacks.	Reach level 20	Texture2D'/Game/GameMenuInventory/Textures/Inventory/Icons/T_Icon_Stool.T_Icon_
8	bags	Bags	Defense	Boosts defense, a reliable shield in a convenient bag.	Reach level 3	Texture2D'/Game/GameMenuInventory/Textures/Inventory/Icons/T_Icon_Cash.T_Icon_
9	clock	Clock	Defense	Augments max HP, a timeless guardian for enduring defense.	Reach level 6	Texture2D'/Game/GameMenuInventory/Textures/Inventory/Icons/T_Icon_Stopwatch.T

DT_PlayerItems

This data holds the player state of each item.

For the sake of a demo, this data is stored locally here in a data table. In a published game this is usually managed by a player management layer where this data is stored in a remote data store or a save file.

It is defined by an ID (row name), unlocked state, and equipped state.



View

Widgets in the `GameMenuInventory/Widgets` folder act as the display (View). They bind to change events in the view models and update their own displays correspondingly. They also send events to the view models when they receive player input.

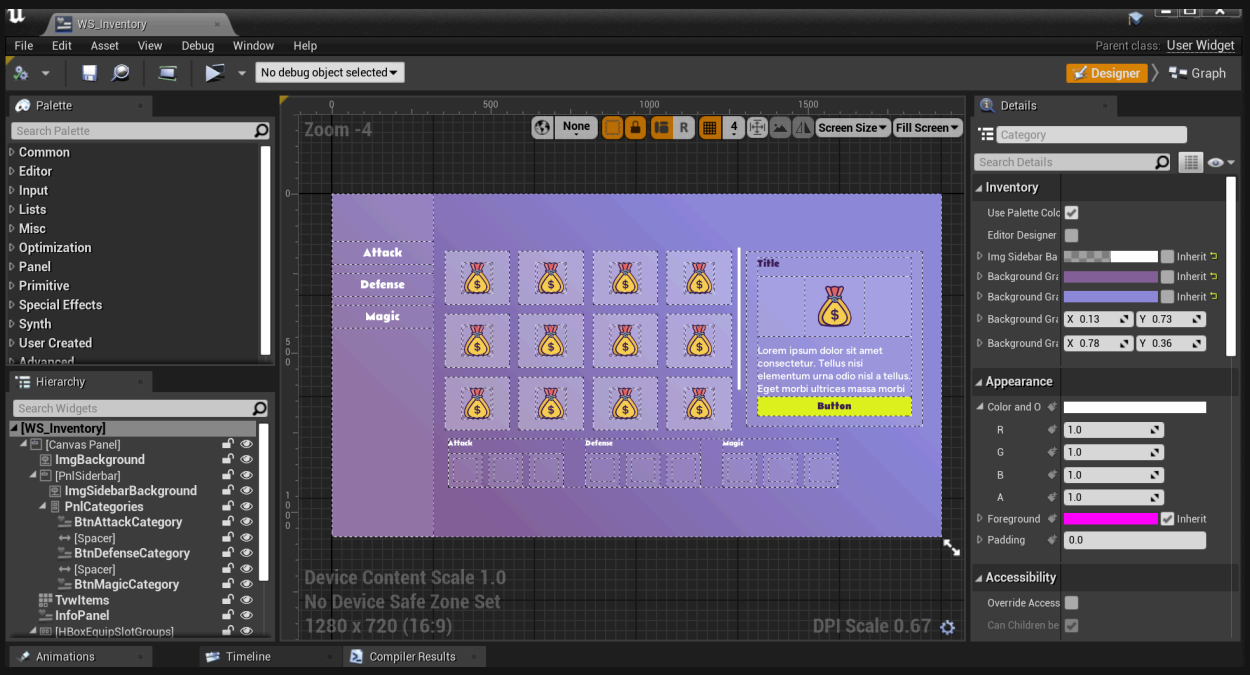
Top-level widget screens have a `WS_` prefix.

Toolbars like the top navigation bar and the palette switcher toolbar have a `WT_` prefix.

Sub-widgets have a `W_` prefix.

WS_Inventory

This is the main top-level Inventory view.



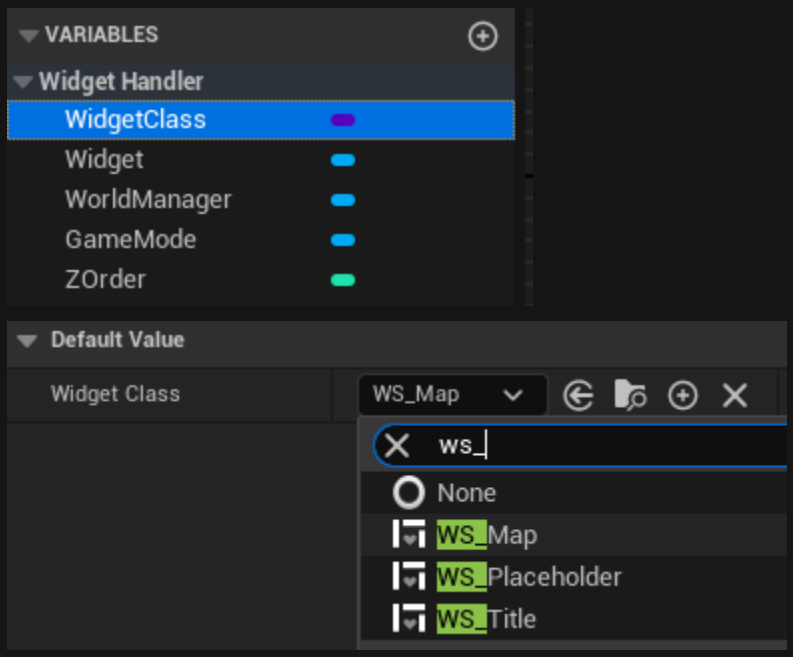
On the left, is a ListView of categories to navigate - this subwidget is **W_InventoryCategoryBtn**.
In the center is a TileView of items to unlock and equip - this subwidget is **W_InventoryItemEntry**.
On the right is a panel displaying the selected item's description and the equip button - this subwidget is **W_InfoPanel**.
On the bottom is a panel displaying all equipped items across all categories - this subwidget is the **W_InventoryEquipSlotGroup**.

View Model

All top-level screen and toolbar view models reside in the `GameMenuInventory/DemoAssets/WidgetHandlers` folder. They have a `WH_` prefix (stands for widget handler).

They share the same name after the prefix. For example, the inventory screen's view model is `WH_Inventory`.

Every widget handler is bound to a top-level screen or toolbar, and it is specified in the widget handler's WidgetClass variable.

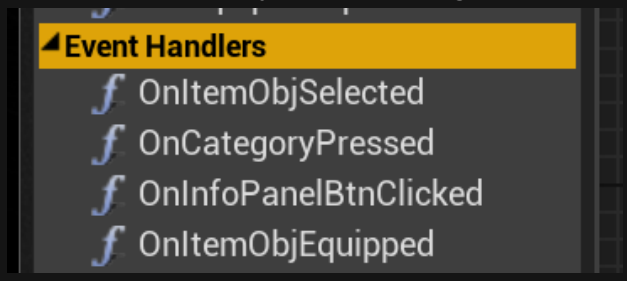


View models serve as the intermediate layer that reads game-specific data and serves it to the display (widgets) through events.

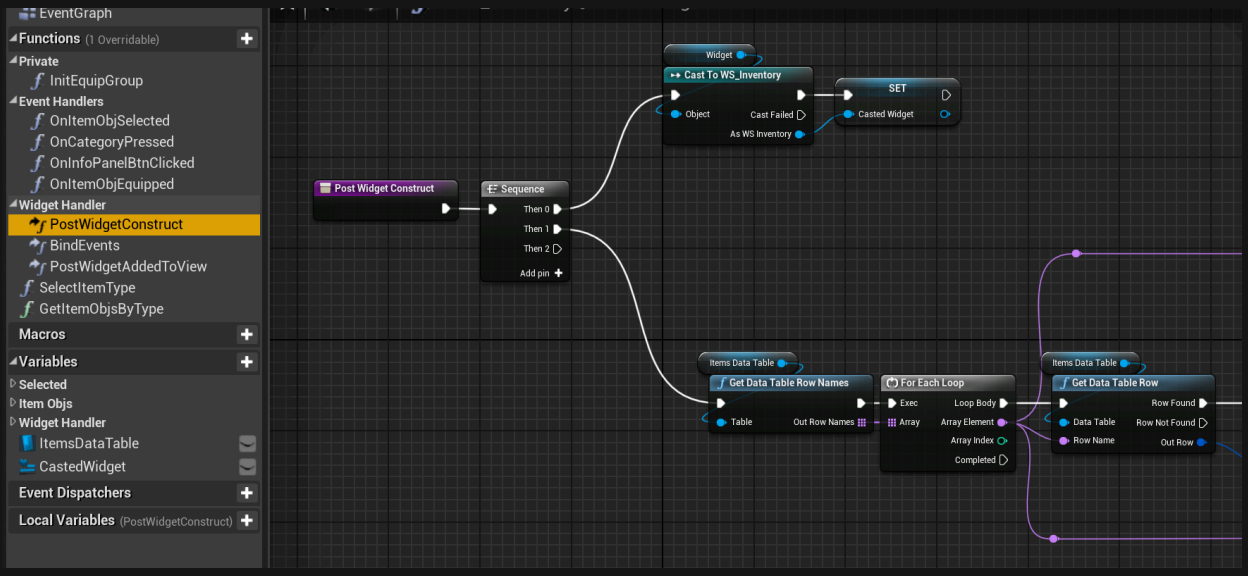
It is best practice for game-specific logic to reside in the view model for portability.

WH_Inventory

This is the main view model.
It binds the inventory screen's navigation events to these functions in this blueprint:



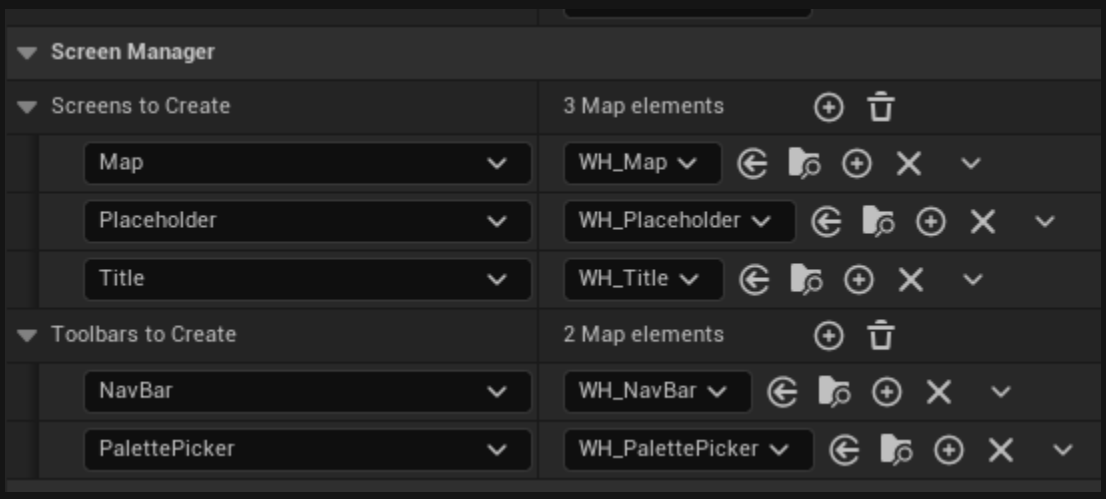
It also initializes the viewmodel with data (model) in the **PostWidgetConstruct** function.



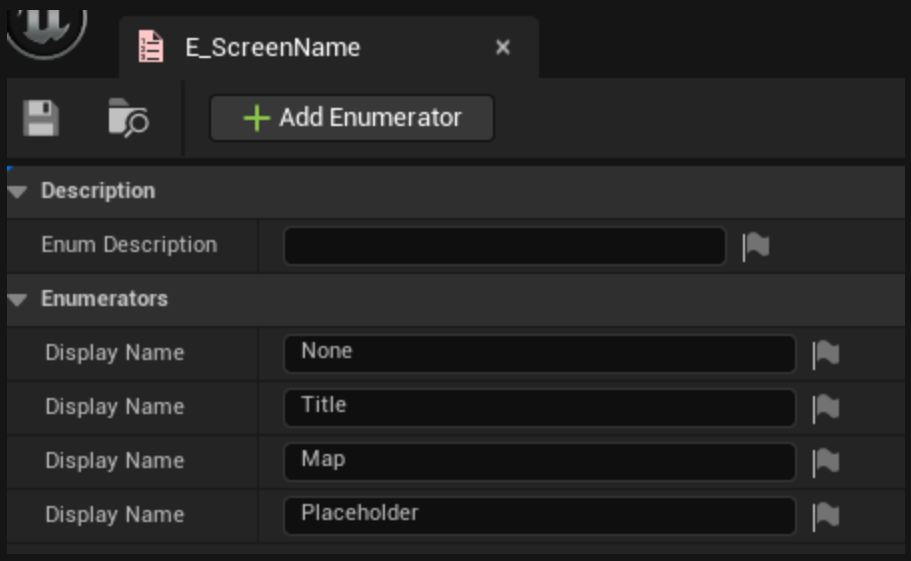
Screen Manager

The ScreenManager is the manager that loads all top-level screens and toolbars.
When the ScreenManager is loaded, it creates the widget handler objects, as well as their bound widget objects.

The list of screens to create can be modified in the instance-editable quest ScreensToCreate. In the demo, this is set in the world instance of ScreenManager in **`LP_DemoQuest`**.



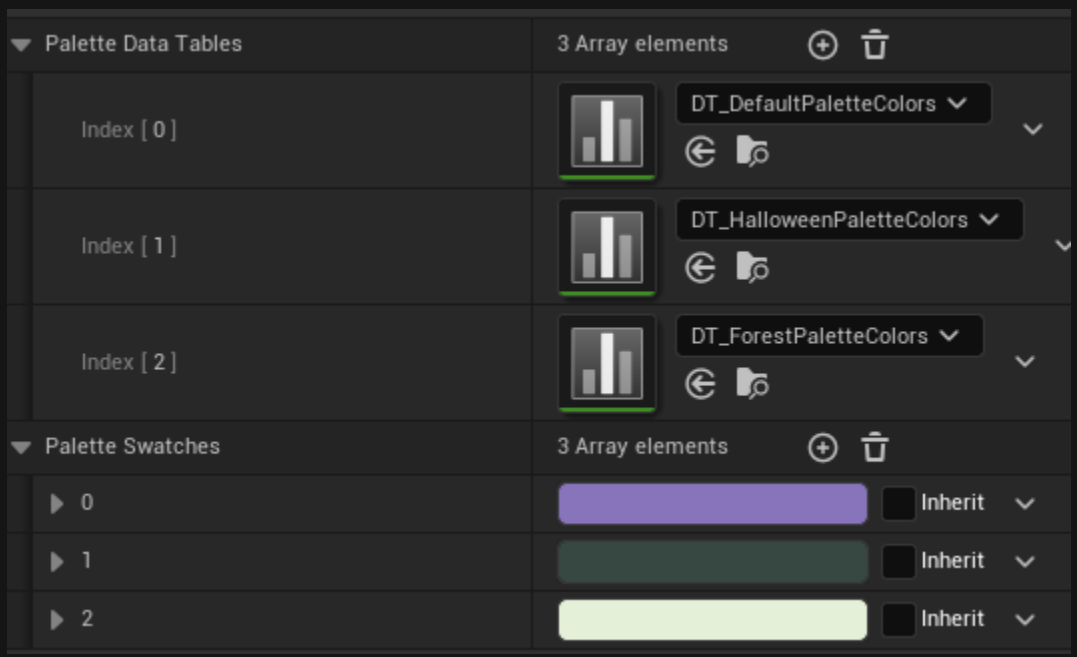
To add a new screen, simply add a new screen type (in the enum **`E_ScreenName`**), and bind it to a widget handler in the ScreenManager instance.



Palette Manager

Swappable palettes are defined in data tables in `GameMenuQuest/Palette/DT_*`.

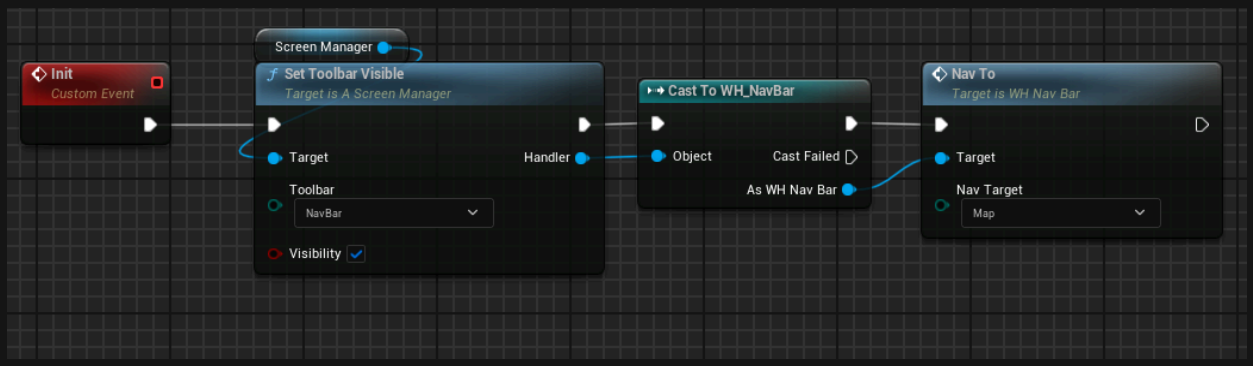
To create a new palette, duplicate the default palette, `DT_DefaultPaletteColors`, customize the colors to your liking, and add it to the PaletteManager world instance's settings.



Adding a corresponding palette swatch color will also add the palette to the palette hotswap toolbar.

Debug Manager

The DebugManager boots the demo into the Quest screen for debug purposes. It is an example of how to use the ScreenManager's NavTo function to open a screen.



Contact

We would love to hear your feedback and questions on the product.
Please leave us your thoughts in the product questions page or reach out to us directly at obertake@gmail.com.

Thank you