# Widget Class Inventory

## Creating Classes

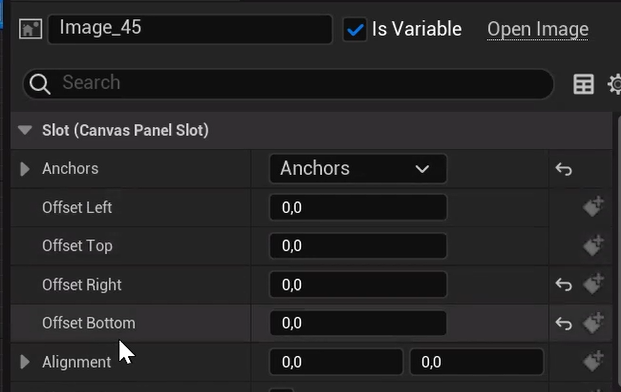
Create a new C++ Class > UserWidget (Inventory)

Create a Blueprint inheriting from WBP\_Inventory

## Editing Blueprints

Drag a **Uniform** **Grid** **Panel** to the Hierarchy. Containers:

* **Canvas**: can use Z-order, worse performance, with Anchors and offsets

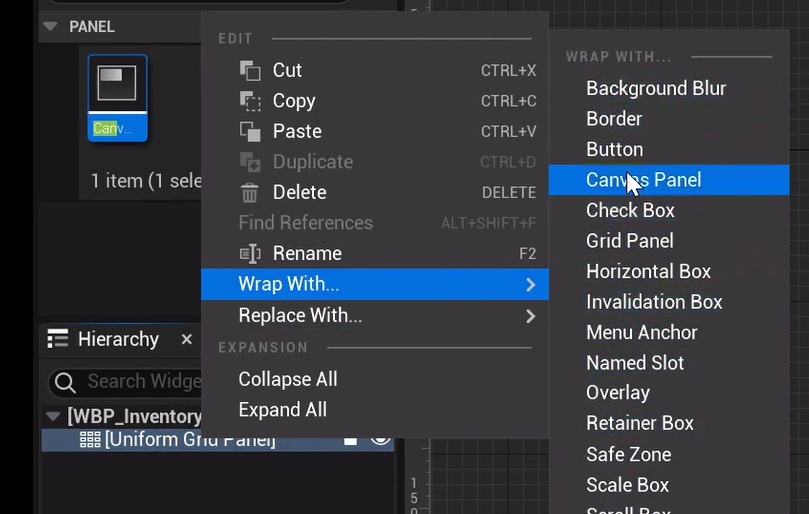


* **Overlay**: cant use Z-order, better performance, with padding



Once the Uniform Grid Panel is created, we can’t add a Canvas as a parent

But we can Right Click the Grid Panel > **Wrap With** > Canvas Panel:

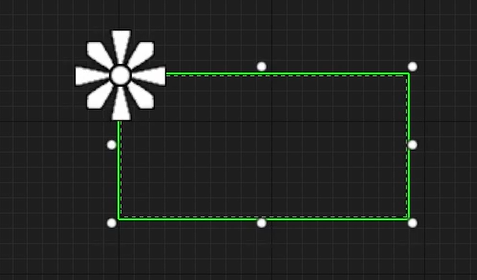


**Add** **an** **Image** to the Canvas.

To **adjust** the image to the whole canvas:

* we put **Offset to 0**
* Anchor
* Opacity 20%

We create another child Canvas.



* Adjust it to center:
  + **Anchor**.
  + **Alignment** to 0.5 and 0.5: center the canvas

### Hierarchy

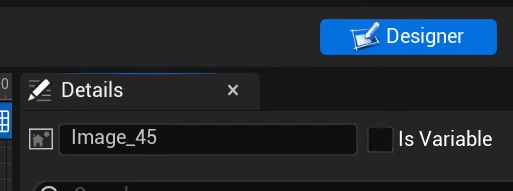


* **Size To Content (Uniform Grid Panel)** : size depends on the child elements

### IS VARIABLE

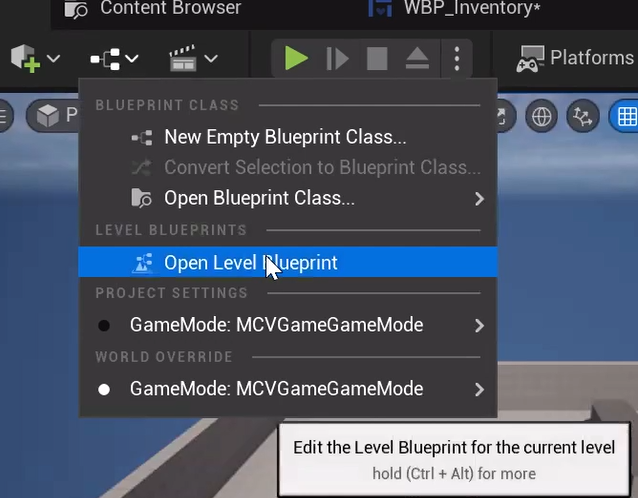
I want the images aren’t variable (they are black screen with 80% opacity).

So in the blueprint we must not edit them 🡪 **Is Variable [ ]**



## Adding the Widget

Open the Level Blueprint:



## Reparent Class

We change the parent class so we inherit from our C++ Class (Inventory) Instead of the User Widget

## C++ Inventory.h

* NumRows and NumColumns with UPROPERTY and
  + EditAnywhere: to change from Editor
  + BlueprintReadOnly to don’t allow change size dynamically
* UUniformGridPanel is mandatory so we add

meta =(BindWidget)

So the result is:

