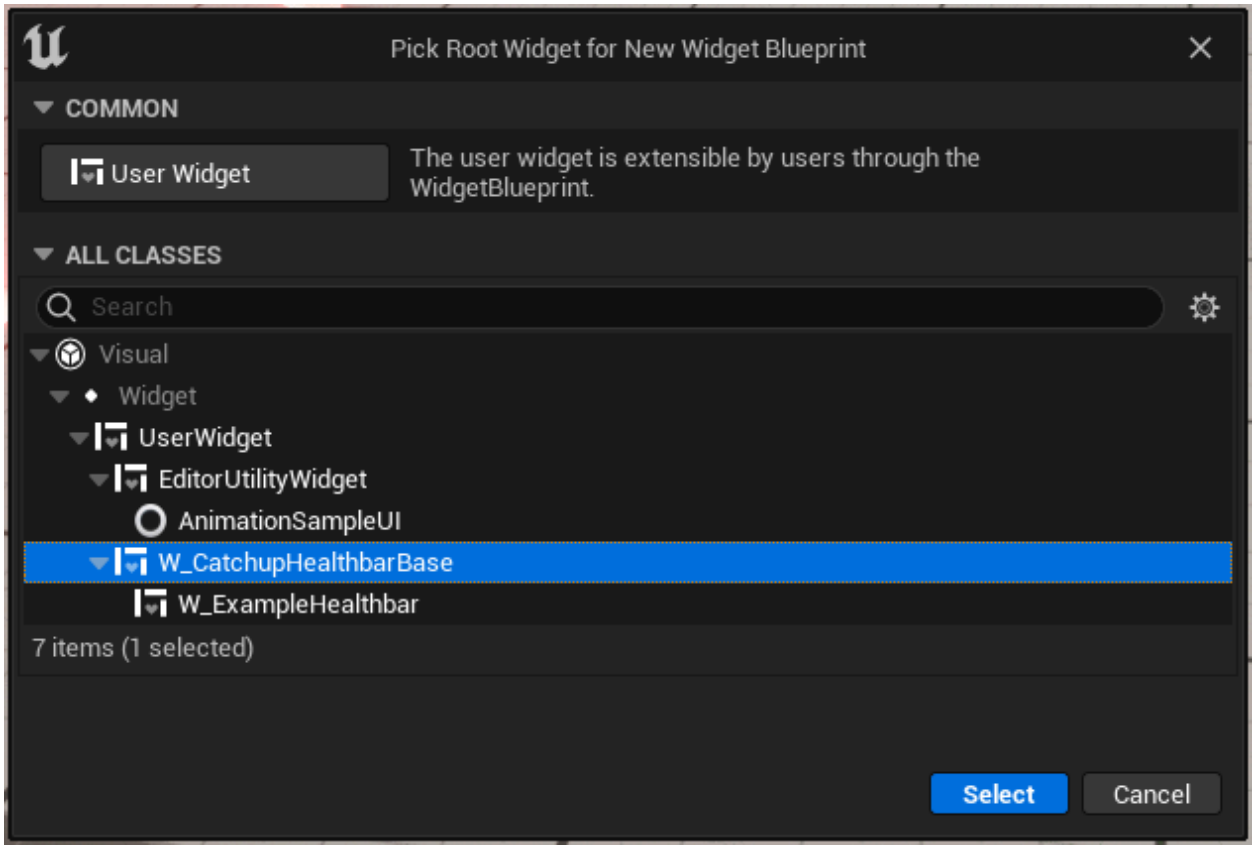
Getting Started

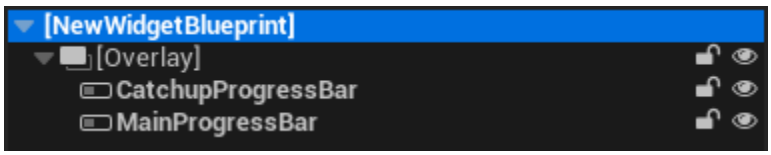
Creating A New Widget With Catchup Functionality

Tip: See `W_ExampleHealthbar` for an example implementation.

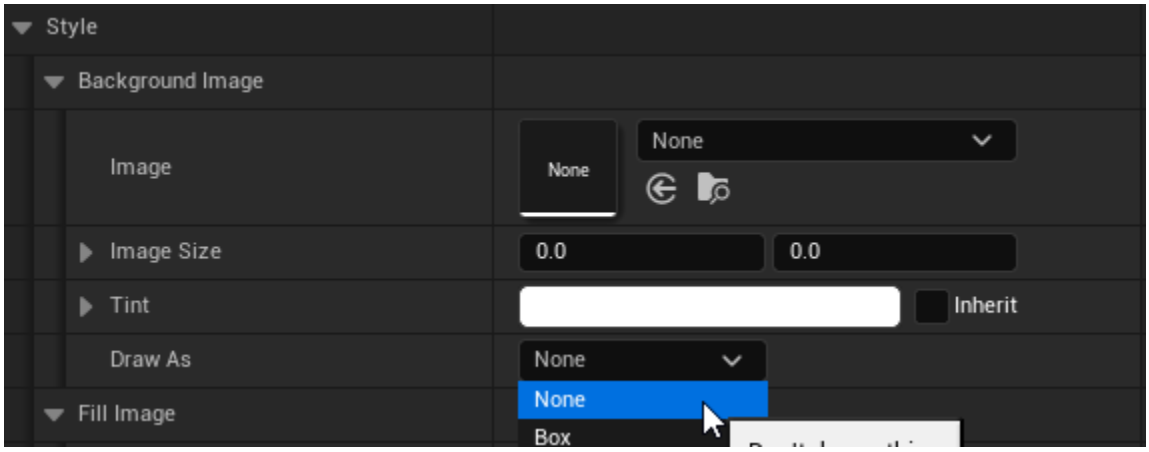
1. Create a new `User Widget` class (User Interface → Widget Blueprint).
2. Select `W_CatchupHealthbarBase` as the parent class for the widget.



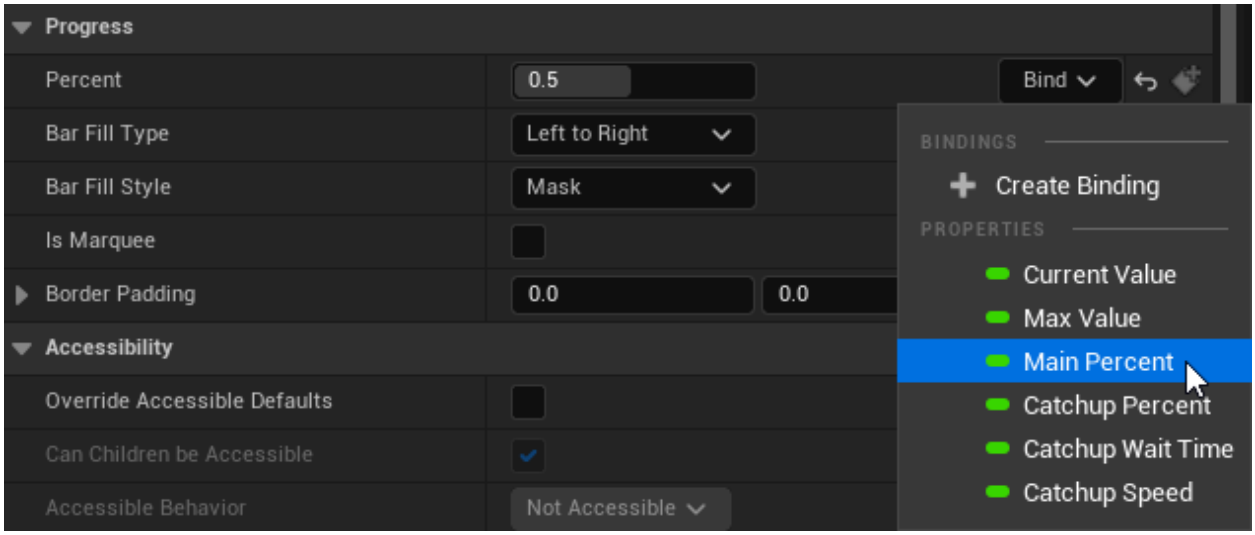
3. Add an `Overlay` widget.
4. Add two `Progress Bar` widgets to the overlay. One will be the main progress bar, and the other will be the catchup progress bar.



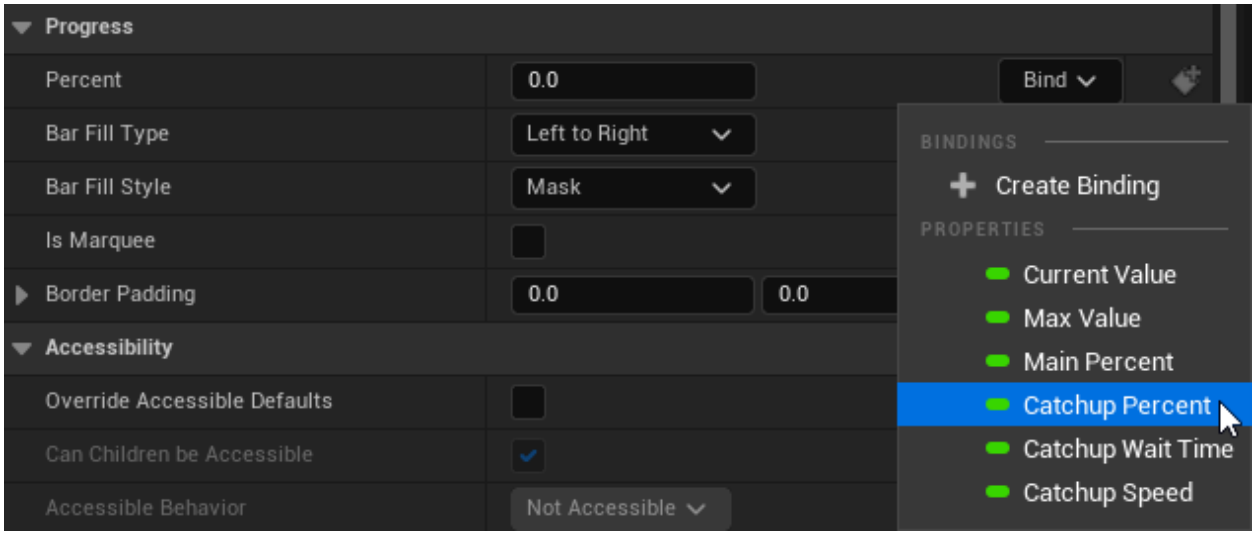
5. Select the main progress bar and ensure `Background Image` is set to draw as None.



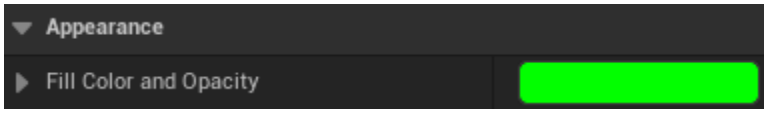
6. Bind the percent for the main progress bar to `Main Percent`.



7. Bind the percent for the catchup progress bar to `Catchup Percent`.

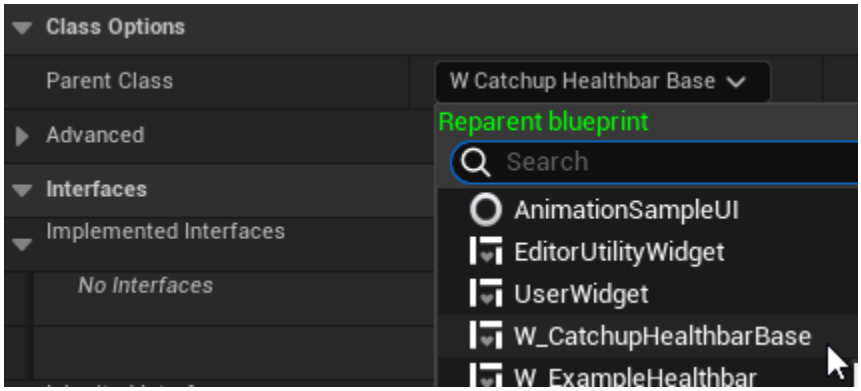


8. Ensure the catchup progress bar has a different fill color than the main progress bar so they can be distinguished from each other.

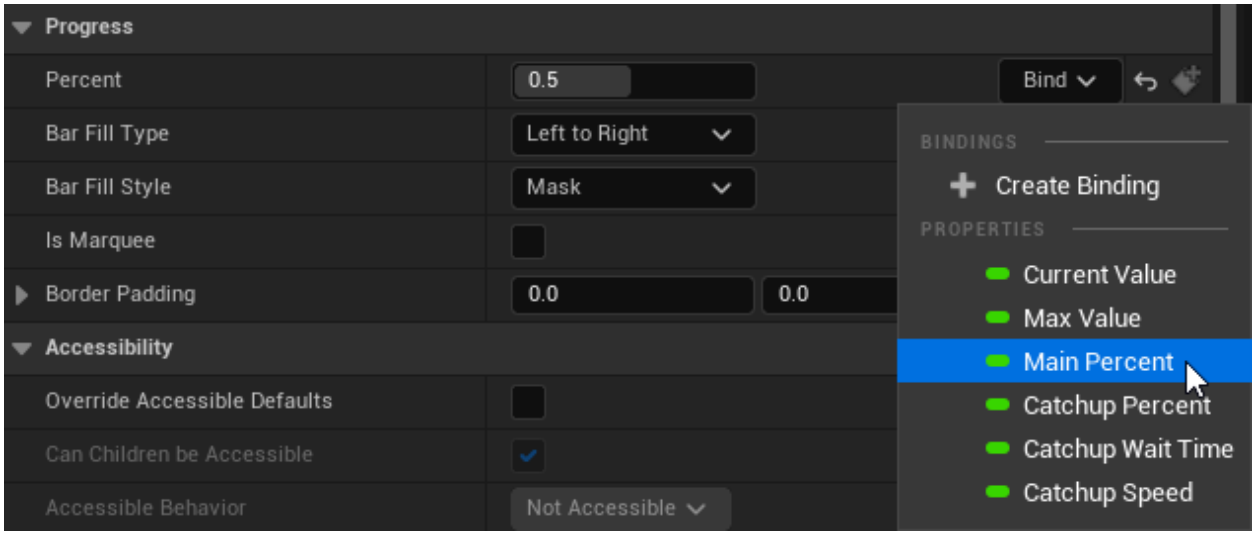


Adding Catchup Functionality To An Existing Widget

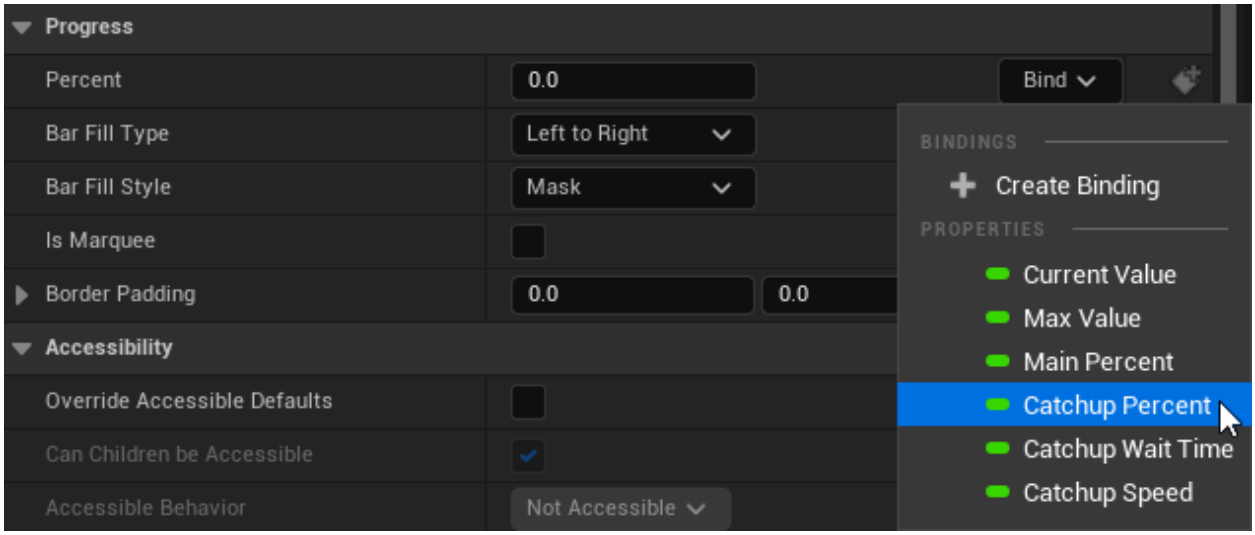
1. Open your widget and click Class Settings.
2. Change `Parent Class` to `W_CatchupHealthbarBase`.



3. Bind the percent for the main progress bar to `Main Percent`.



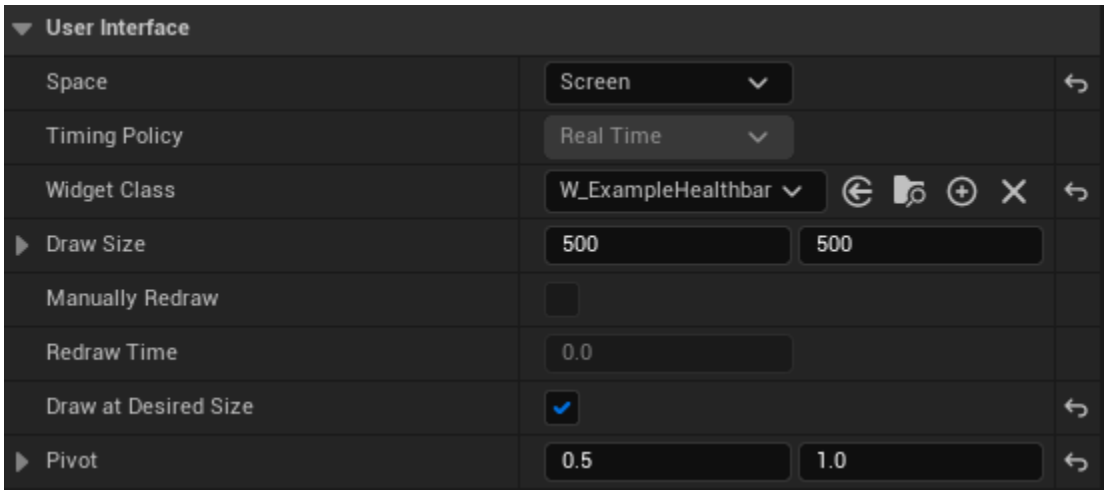
4. Bind the percent for the catchup progress bar to `Catchup Percent`.



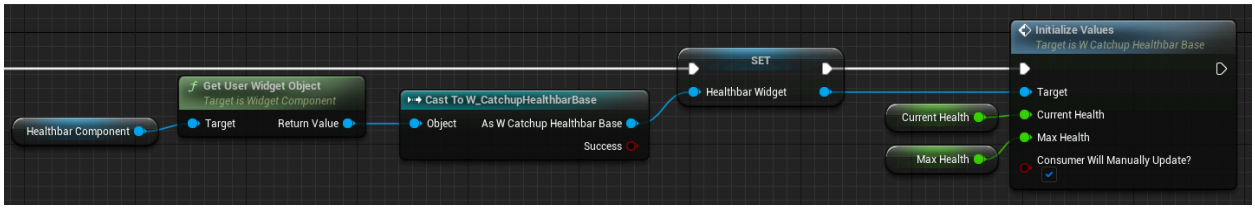
Adding A Healthbar To An Actor

Tip: See `BP_NPC` and `W_ExampleHealthbar` for an example implementation.

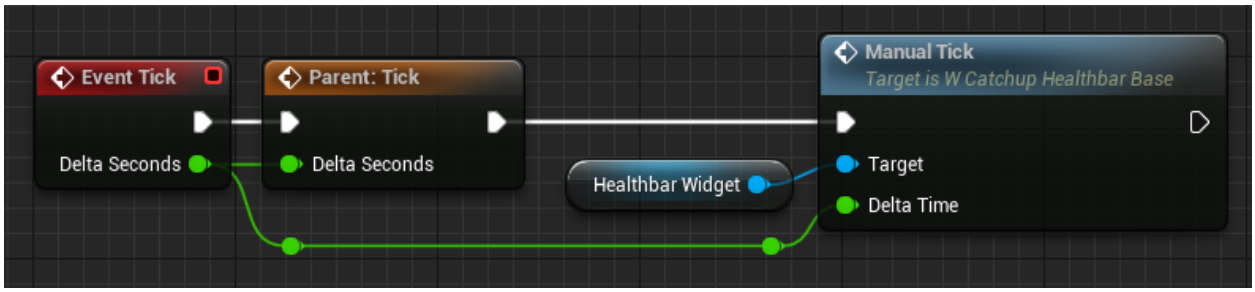
1. Add a `Widget Component` to your actor.
2. Set `Space` to `Screen`, set `Widget Class` to your healthbar widget class, and set `Draw At Desired Size` to `true`.



3. In the `BeginPlay` event, call `Initialize Values`. Make sure to set `Consumer Will Manually Update?` to `true`, because screen space widgets don't always receive tick events (e.g. when they're offscreen) which can lead to odd behavior.



4. In the `Tick` event, call `Manual Tick`. This way, we can be sure the widget always receives tick events.



5. In the `AnyDamage` event (or whenever the current health changes), call `Set Current Value`.



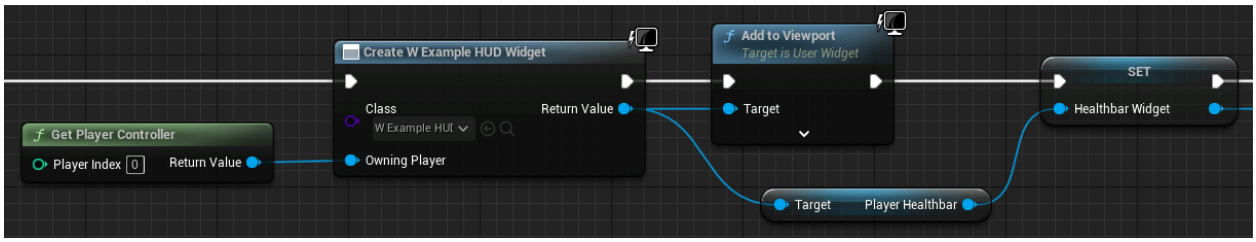
Adding A Healthbar To The Screen

Tip: See `BP_Player` and `W_ExampleHUD` for an example implementation.

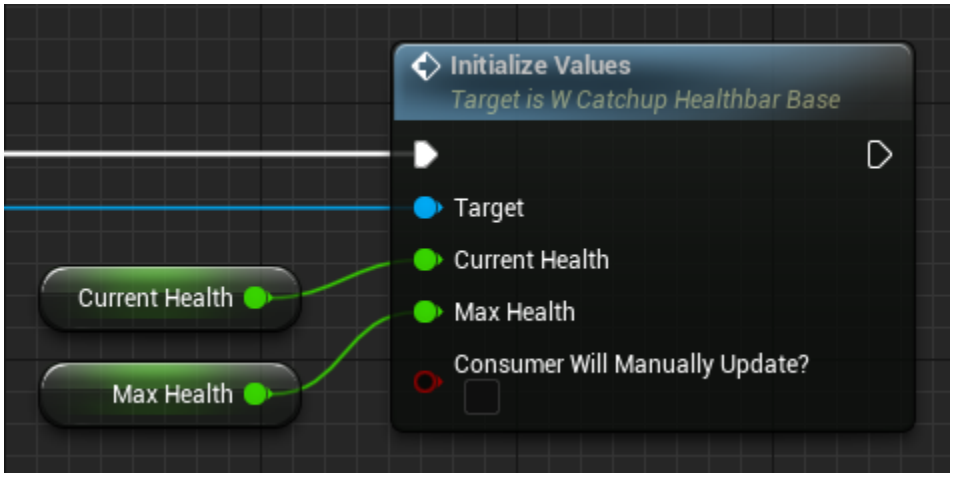
1. Create a new widget or edit an existing one.
2. Add a healthbar widget to your widget.
    a. Tip: If you're using `W_ExampleHealthbar` for this step, set `Hide Healthbar When Full` to `false` so the healthbar is always shown on the screen.
3. Set `Is Variable` to true so the healthbar widget is accessible to Blueprints.



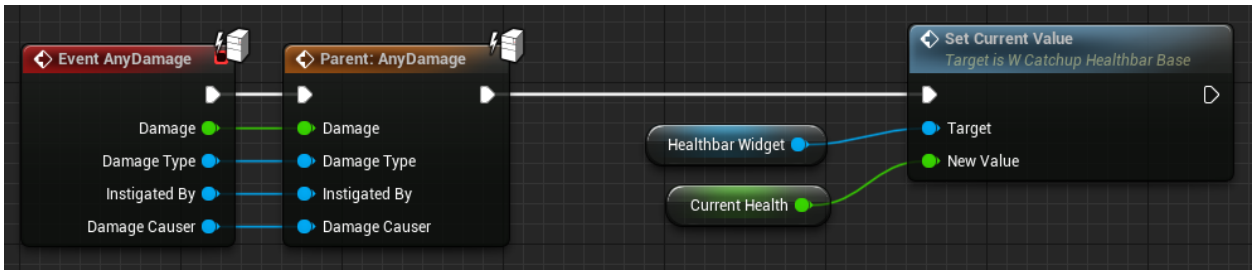
4. In your player blueprint, create your widget and add it to the viewport.



5. In the `BeginPlay` event, call `Initialize Values`. We can leave `Consumer Will Manually Update?` unchecked because the widget will always receive tick events when added to the viewport.



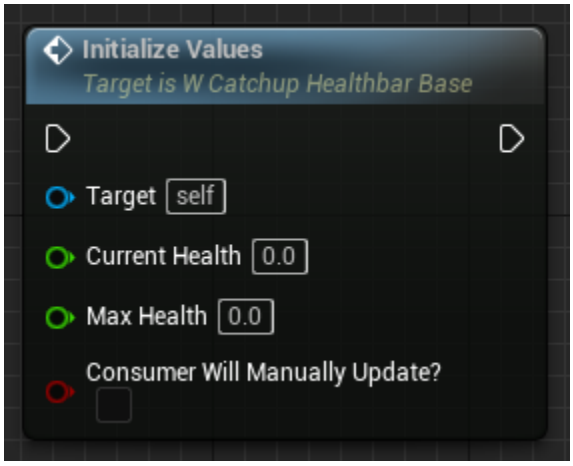
6. In the `AnyDamage` event (or whenever the current health changes), call `Set Current Value`.



API Reference

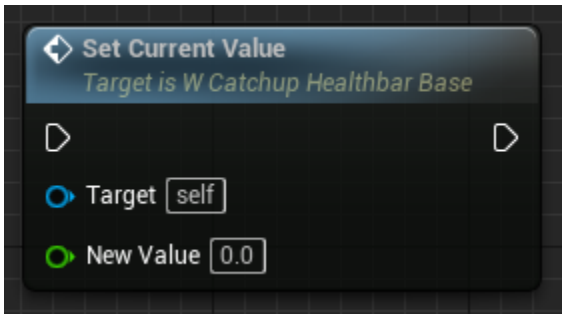
Methods

Initialize Values



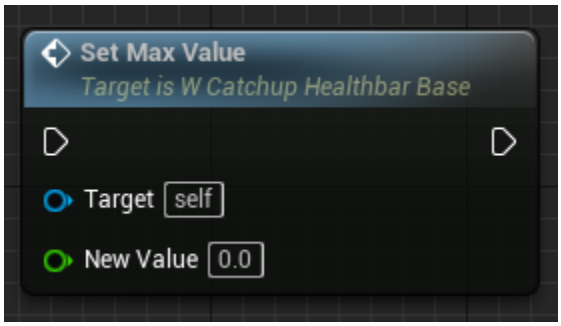
Convenience function. `Current Value` should be the current health, `Max Value` should be the maximum health. `Consumer Will Manually Update?` controls if the widget should handle updating itself or not (`false` = automatically update, `true` = manually update).

Set Current Value



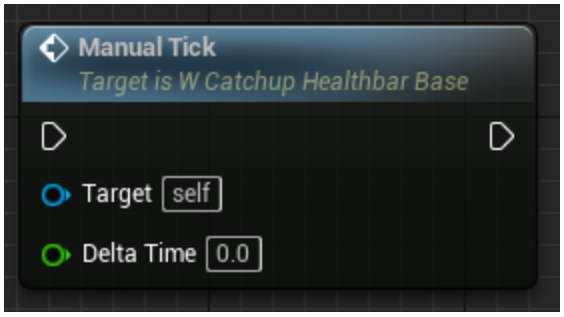
Sets the current healthbar value to `New Value`. Triggers events if the value has changed.

Set Max Value



Sets the maximum healthbar value to `New Value`. Triggers events if the value has changed.

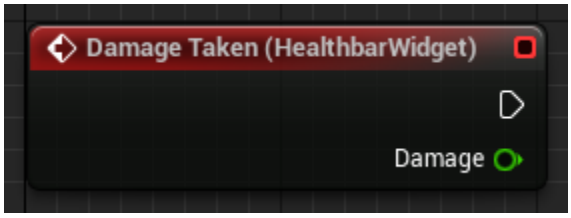
Manual Tick



Updates the catchup healthbar. Should only be called if `Consumer Will Manually Update?` is set to `true`, otherwise the widget itself will call this function unnecessarily. `Delta Time` should be set to the delta time of the calling tick event, or the world delta time.

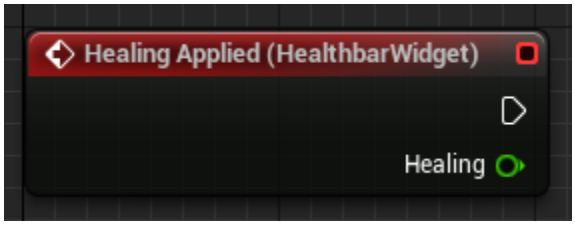
Event Dispatchers

Damage Taken



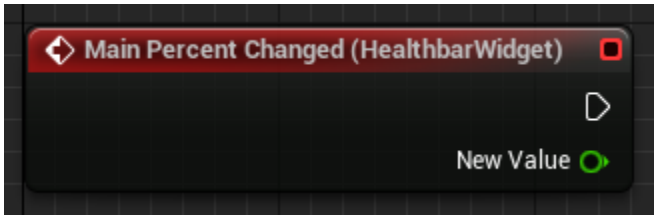
Called when the current healthbar value decreases. `Damage` is the absolute value of the decrease.

Healing Applied



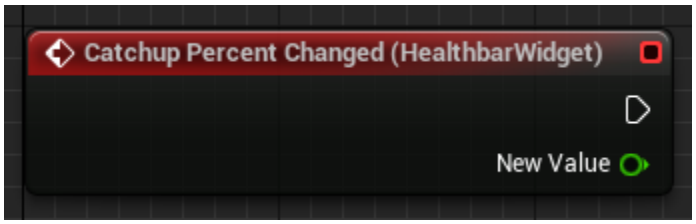
Called when the current healthbar value increases. `Healing` is the absolute value of the increase.

Main Percent Changed



Called when the main healthbar percent (0-1) changes. `New Value` is the percent after the change.

Catchup Percent Changed



Called when the catchup healthbar percent (0-1) changes. `New Value` is the percent after the change.