Home Values

We have discovered risks to propagating our unsynced paradigm to more components without having proper support for differentiating a default value and a current value. Additionally, It's a big pain point to initialize lots of variables in players rooms, and requires the creator to build additional circuit graphs to reset their rooms to a testable state.

Goals

1. Variables/Components support a home value that can diverge from the current value
2. Players understand the difference between home value, and current value
3. Variable's/Component's current values get set to their default values on Room Load/Room Reset/OnReset
4. Home values can be used reliably when joining the room.
5. It's always clear if the current value diverges from the home value
6. Unlock the UGC team to add unsynced capabilities to more Components
7. Don't cause any major rewrites of CV2 or other systems
8. Don't move too much UX cheese for creators. I.e., try to evolve existing paradigms where possible

How It Works

Terminology

- Default Value (non-player facing terminology)
    - This is the value you get on a fresh object out of the palette (i.e., what we set in C#)
- Home Value
    - This is the value the property will reset to, and what is saved into the room.
- Current Value
    - This is what circuits can touch. The current value of the property. On reset, this value is discarded and reset to the home value. Circuits cannot directly set home values.

Rules

Variable chips/Components need to support the concept of "home value". The rules are:

- Chips/Components have default home values (that's what you get when you spawn a fresh one out of the palette).
- The current value is set to the home value when the chip is created.
- Circuits can change the current value.
- Current and home values can diverge, that's fine.

- Circuits cannot change the home value. Only Maker Pen tools (Move, Rotate, Scale, Config, Recolor, …) can update home values. More on this later.
- The current value is set to the home value on reset/save.
- Home Values are editable via config menu
- When you adjust a property via config menu, you are adjusting both the home value and the current value at the same time.
- When the current value diverges from the home value we visualize it in the config menu
- For variables specifically, we only expose home values to “simple” types. (string, int, float, bool)

Examples

Let’s walk through a few examples of home values following the above rules.

NOTE - For all the following examples, I use a simplified UX visualization than what we really want. At the bottom of the doc are Tan approved visuals that we should adhere to.

Variable - Home Value

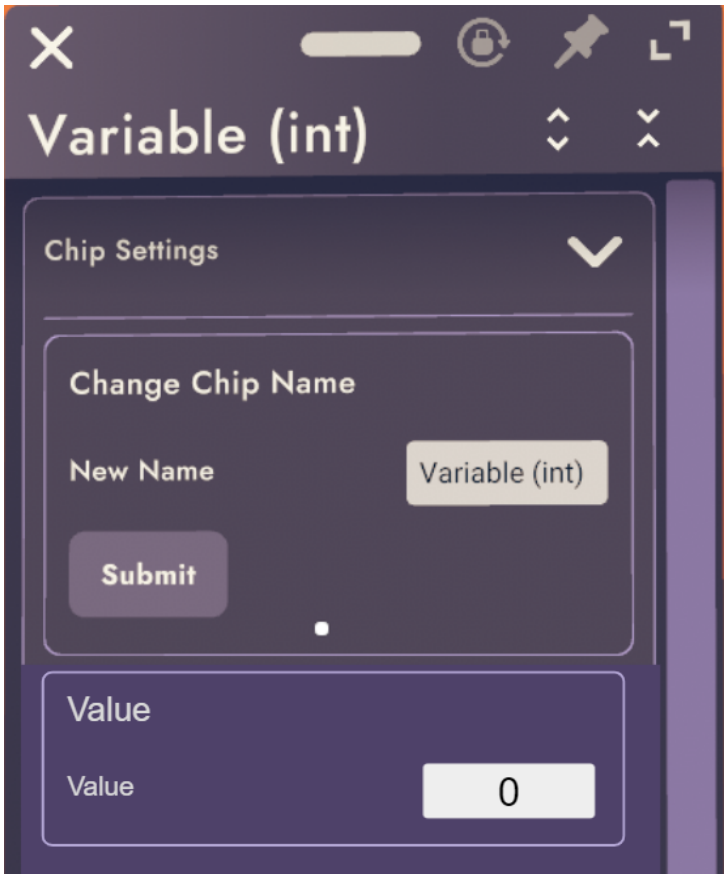
We spawn a brand new int variable out of the palette and open the config menu.

Default Value - All int variables spawn out of the palette with the value 0.

Home Value - Knowing the above, the current home value is also 0.

Current value - We haven’t changed the value via circuits yet, so the current value matches our home value.

Let’s try changing the home value.



To change the home value we edit the “Value” field in the config menu. The only way to change the home value is via the config menu.

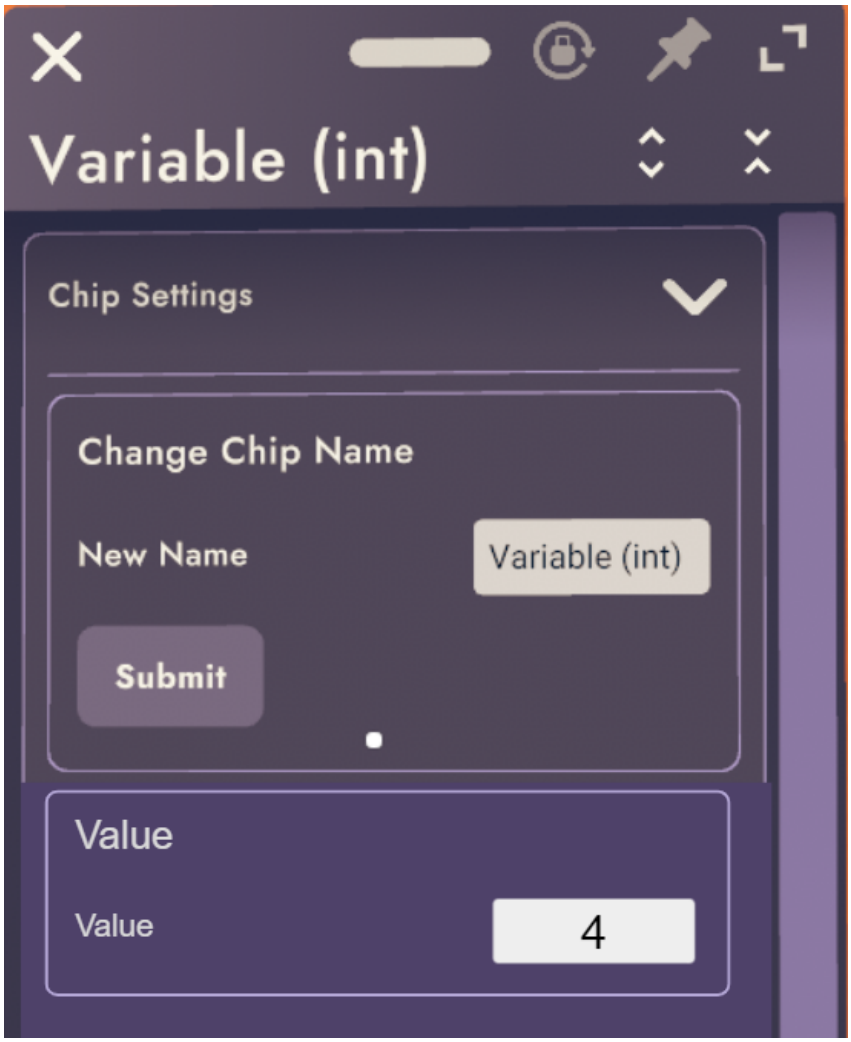
We just changed the home value on this int to 4!

Default value - **This did not change.** If we spawn another int variable, it’s default value will still be 0.

Home value - We changed this from 0 to 4.

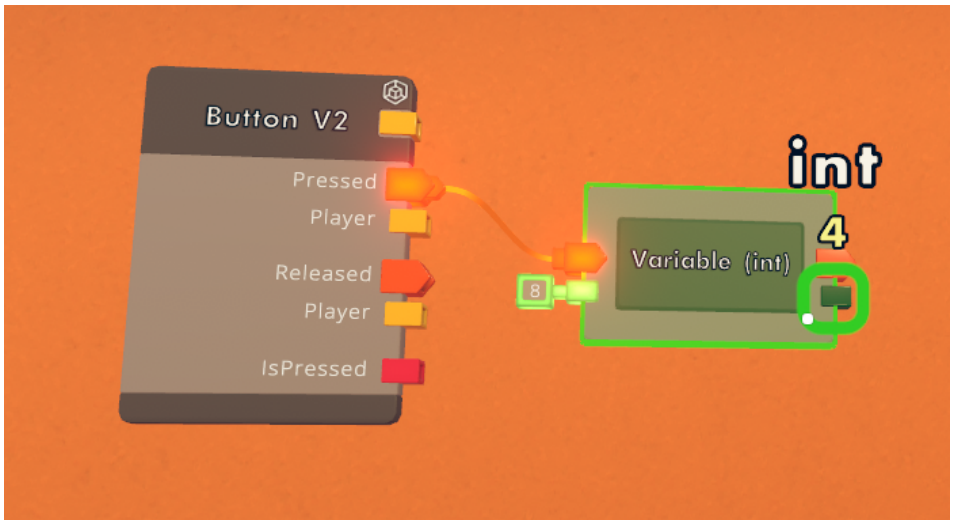
Current value - Looking back at our rules, since we just changed the home value via config, we have also just changed the current value from 0 to 4.

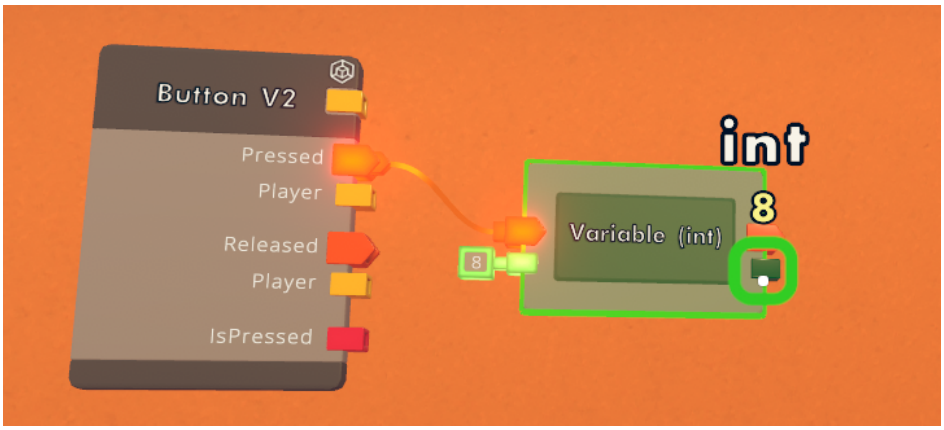
Alright, we’ve successfully changed the home value via config, let’s now look at what happens when the current value diverges from the home value. You might be wondering why that might happen at all, and the answer is CV2.



Variable - Divergence

Let's build a scenario where I change the current value of my int variable.





It worked!

Default value - Once again, **This did not change.** Going forward, I won't mention this one anymore. There simply isn't a way for a user to change this and we don't care about changing it!

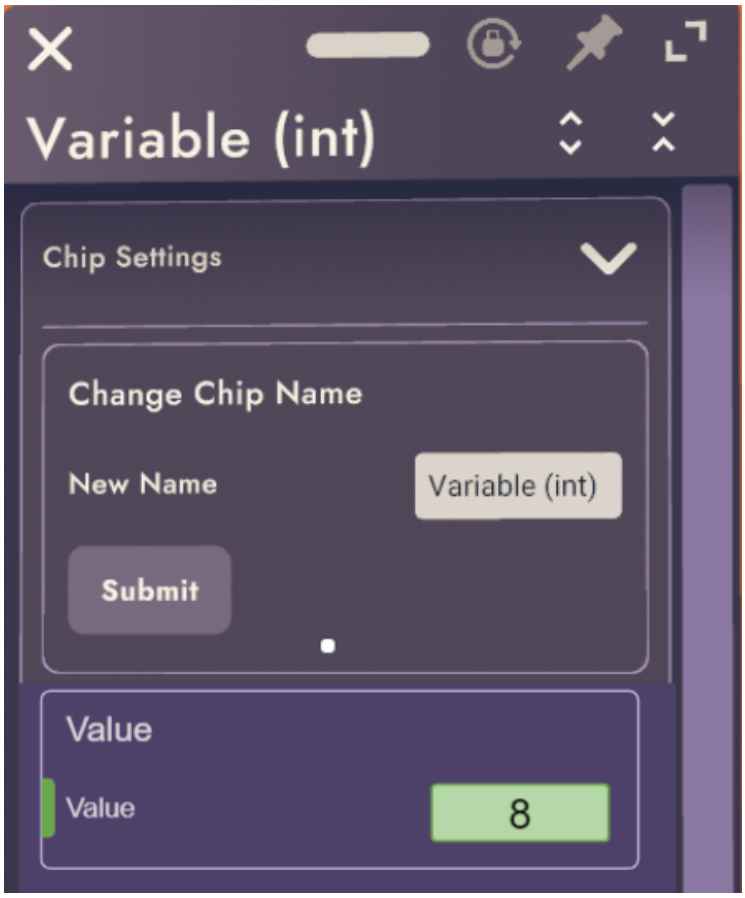
Home value - **This did not change.** Going back to our rules, you can only change a home value by changing the home value field in the config menu. This will be important in just a second.

Current value - **We changed this one!** The current value of our variable is "8"

Let's take a look back at the config menu now for our variable.

You can see on the config menu the field where the current value has diverged from the home value now has the following:

- Green line preceding the value property
- Green outline on the field itself
- Green background on the field



Important things

- The home value remained unchanged. You can only change the home value via the config menu! Under the hood, the home value is still 4. We can get back to this state by resetting the room or loading back into the room.
- If we change this field manually via config menu to “10” it would set the current value and home value to “10” and the divergence visual indicators go away.

Variable - Reset

Our variable’s current value and home value have diverged. The point of having the home value is that on reset our current value reverts back to the home value.

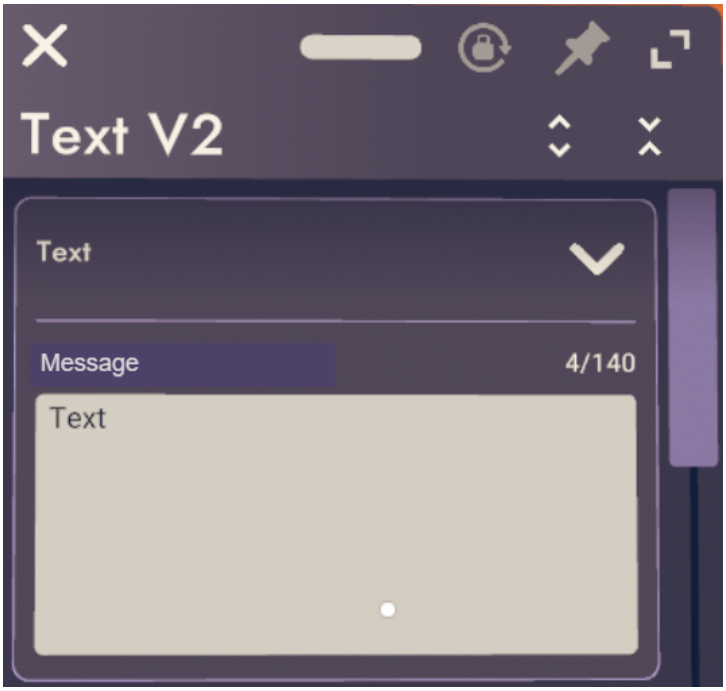
In our current state, if do any of the following:

- Run the room reset event from the palette
- Run the reset room chip
- Run the reset object chip pointed at this text component
- (NEW) Press the reset button in an Variable chip/Component’s config menu. More on this one at the bottom of the doc.
- Save the room

The current value would get set back to what we set it to way back, “4”.

Text - Home Value

We spawn a brand new text component out of the palette and open the config menu.



Let's enumerate the default, home, and current values for the "Message" field of the text component.

Default value - All text components spawn out of the palette with the same default value for the message field, "text".

Home value - Knowing the above, that means that the current home value is also "text".

Current value - We haven't changed the value via circuits yet, so the current value matches our home value.

Let's now try changing the home value for the message field to "hi!". To do this, we must edit the value via the config menu.

Random note - the message field is currently called "Message 0" for CV1 reasons I bet. Can we change this to just say "Message"?

Great! We just changed the home value for this text component. Let's enumerate what's changed again.

Default value - **This did not change.** If we spawn another text component, it will still say "text" and not "hi!".

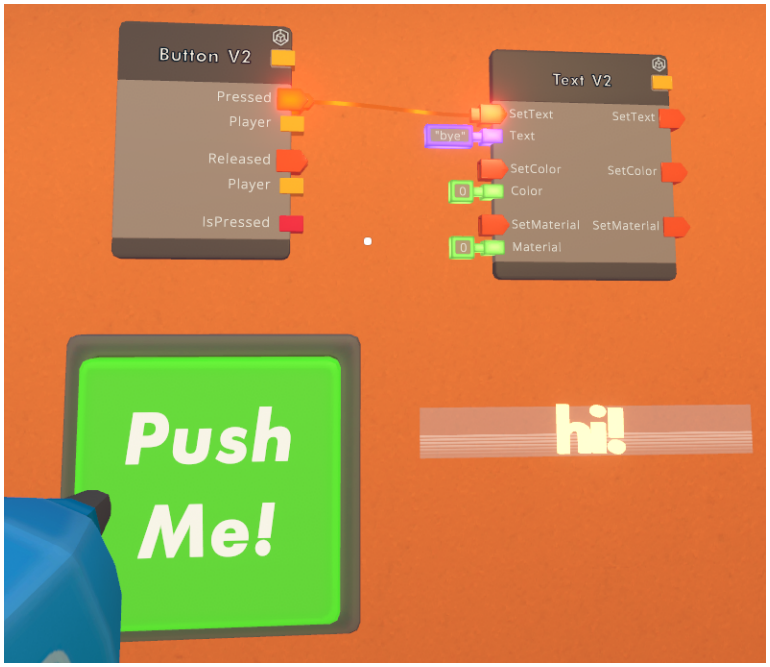
Home value - We changed this from "text" to "hi!". We'll get to what this means exactly in a moment.

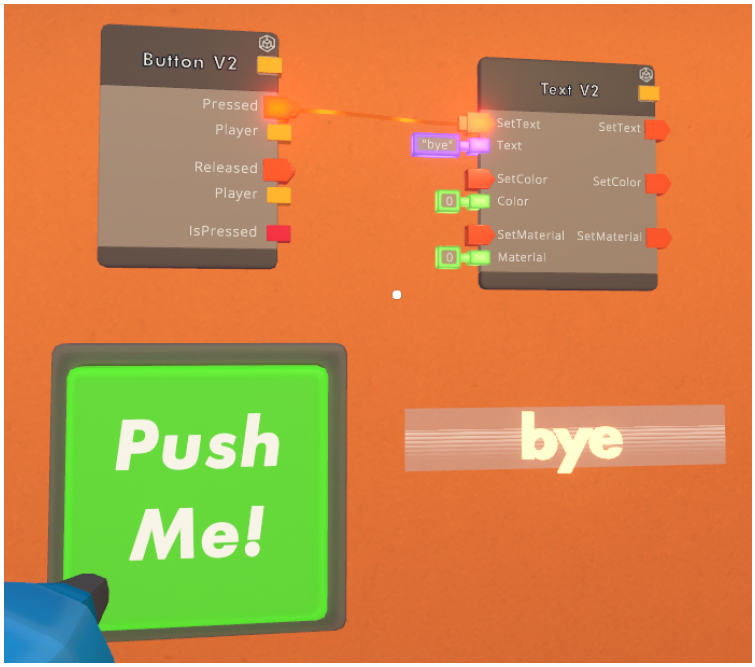
Current value - Looking back at our rules, since we just changed the home value via config, we have also just changed the current value from "text" to "hi!".

Alright, we've successfully changed the home value via config, let's now look at what happens when the current value diverges from the home value.

Text - Divergence

Let’s build a scenario where I use a button to change my “hi!” text to “bye”.





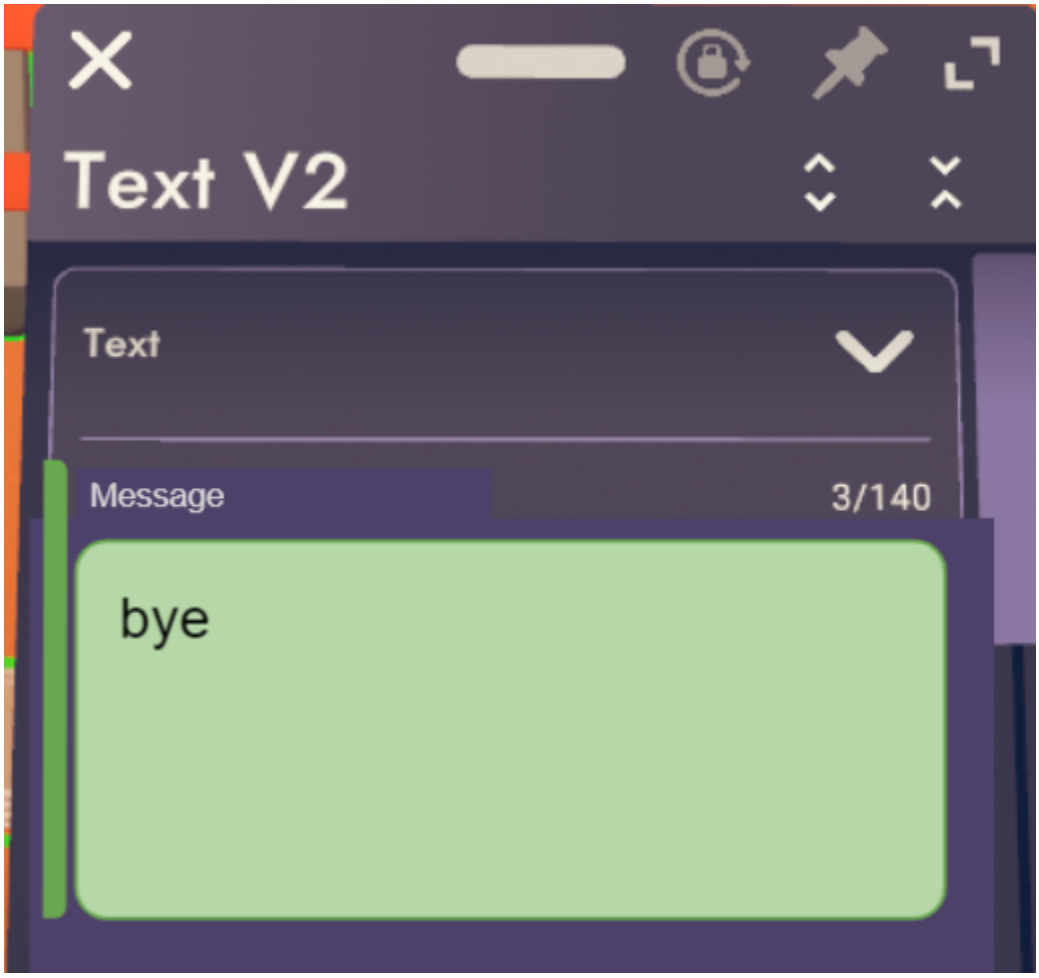
It works! Alright so what changed now?

Default value - Once again, **This did not change.** Going forward, I won’t mention this one anymore. There simply isn’t a way for a user to change this and we don’t care about changing it!

Home value - **This did not change.** Going back to our rules, you can only change a home value by changing the home value field in the config menu. This will be important in just a second.

Current value - **We changed this one!** The current value of the text component is now “bye”.

Let’s see what this looks like in the config menu.



Important things

- The home value field remained unchanged. You can only change the home value via the config menu!
- If we change this field manually to “hello” it would set the current value and home value to “hello” and the lightning bolt would go away.
- Same UX as above for showing how the current value diverged from the home value.
    - Green line preceding the value property
    - Green outline on the field itself
    - Green background on the field

Text - Reset

Our text component's current value and home value have diverged. The point of having the home value is that on reset our current value reverts back to the home value.

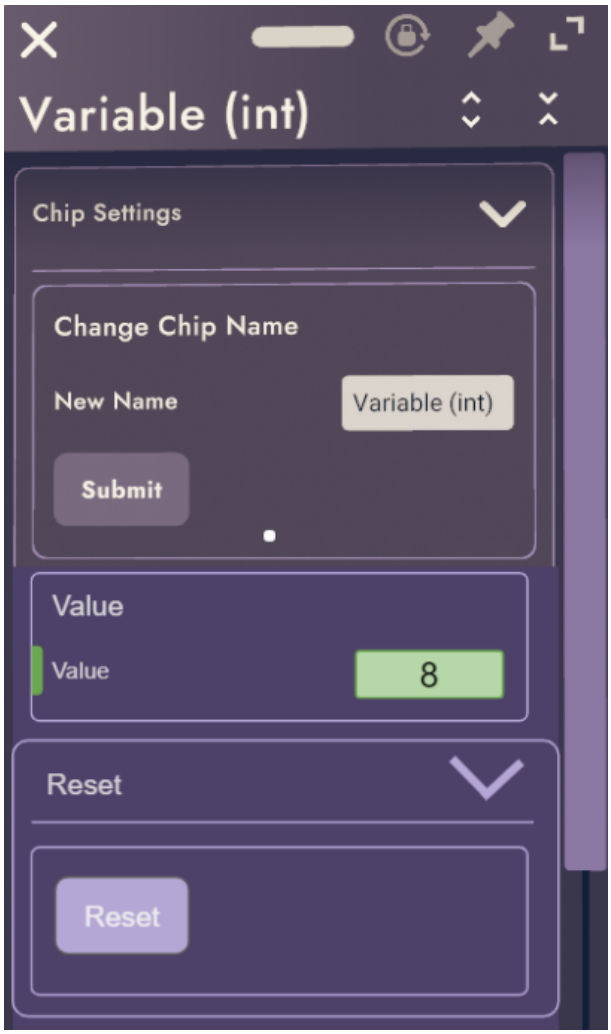
In our current state, if do any of the following:

- Run the room reset event from the palette
- Run the reset room chip
- Run the reset object chip pointed at this text component
- (NEW) Press the reset button in an Variable chip/Component's config menu
- Save the room

The current value would get set back to what we set it to way back, "hi!".

Reset Via Config Menu

Slight tangent to add an object specific Reset to config menus. This would appear on anything that can be Reset. We should add this so its easy to get an object back to its home values without disturbing everyone in the room. This should be the same reset that the room reset paradigm uses, not something new or different. When resetting an object, all of its current values get set to the home values.



Open Questions

- When are these values actually usable? If I load into a room with an int's home value to 4, can I reliably grab that value off the Room Load event? If not, when can I reliably get it?
- Do we like or hate the lightning bolt? I don't have a strong preference. As long as its something that is visible and clear I'll be happy.
- Can we please add the config menu reset with this work.
- Consider the example and tell me if it's possible.
    1. Make a new room
    2. Spawn an int variable chip
    3. Change the home value of the int to 4
    4. Reset the room (before saving)
    5. Expectation - the ints current value is now 4, not 0. Home value is also still 4.

Home Value Audit

The home value functionality needs to be propagated to all of our components + variables. This is a list of things we have in the config menu + chips that can change those values.

- Variables - string, int, float, bool types
- Button V2 - message text
- Piston V2 - max distance/target distance
- Rotator V2 - target rotation
- Trigger Volume V2 - filter by role/by tag
- SFX V2 - volume
- (All things) Tag adds/removals