How does the engine architecture work?

Essentially, this is a WYSIWYG (What-You-See-Is-What-You-Get) website builder tool, but specifically for tabletop games. To design the system, it’s not just static components that don’t interact with each other, but components with linked data, the **model**, that needs to be updated in real-time across all users.

Components reflect certain parts of the model. They can be used to change the model as well.

Consider a Text Input component and its possible content.

|  |
| --- |
| Text Input |
| * *Variable Path* * Label * Placeholder * Popover block * Shuffle button |

The purpose of a Text Input component is to let a user enter some text, and have this text be updated in the model. Shuffle button fetches from a list and chooses a random value.

When the variable changes, other variables that are computed from that variable are updated.

Variables are chained together like nodes, with some variables being *equations* of other variables put together.

Sequence of Events

User makes a change in a component. The component tells the engine to change the value at *Variable Path*. All equations that depend on that variable path get updated recursively. Equation dependencies are automatically updated when the equation is edited.

Not all variables or equations need to have a component. For instance, it’s probably going to be very common to create Boolean “flags” that just mark whether something is on or off.

Equation Syntax

* {var:strength} + {var:agility} - {var:weight}
* Consider using Perchance syntax?
* Full Perchance integration?
* GUI editor for editing equations. Highlight variables. Preview possible errors (dependency not found, etc)