

Cultistlike v1.0.0

Setup

- Start a 3D project (Built-In Render Pipeline)
- Import DOTween package
- Import Cultistlike package

Please refer to the demo scene for a working setup.
Cultistlike/Demo/Scenes/

Game Logic

You can setup game logic by creating **Act**, **Card** and **Rule** assets from the *Assets/Create* menu. **Acts** are objects that accept **Cards** (drag & drop) and are described by **Rules** of **Card** exchange. Each **Rule** specify Requirements (cards that go in) and Results (cards that go out). To get the gist of the idea it would be helpful to play the demo.

Act

Act Name: this will be used as a caption of the first slot
Text: text displayed when the slotted **Card** does not match any rule or no **Card** is slotted
Rules: rules that can be executed by this **Act**
Consume Rule: **Rule** that automatically grabs matching cards (e.g. resources) from the table and destroys them
On Consume Fail: *UnityEvent* to be run when the *Consume Rule* fails to grab a **Card**

Rule

Time: how many seconds it takes to execute the **Rule**
Requirements: **Cards** required to run the **Rule**. Currently **Card** accepting window has the maximum number of 3 **Card** slots but this will be expanded in the update. **Cards** must be slotted in the exact order.
Name: this will be the caption of the **Card** slot. Name of the first requirement is ignored since the caption of the first slot is set by the **Act** name.
Card: reference to the required **Card**
Aspects: if the above *Card* field is not set any **Card** that has all the listed **Aspects** will be accepted
Results: one result will be randomly chosen from the provided list
Chance: 0-1 chance of this result. If only one result is set this field is ignored.
Cards: awarded upon successful execution of the **Rule**
Extra: extra **Act** spawned upon successful execution of the **Rule**
End Text: text to be displayed upon successful execution of the **Rule**
Start Text: text to be displayed after slotting first required **Card**
End Text: default text to be displayed upon successful execution of the **Rule**. Used if the *End Text* field of the executed **Rule** was empty.

Changing Size of Cards

To change size of **Cards** (and **Acts**) you need to modify three sets of values.

1. Actual physical size of the prefabs (*Act*, *Card*, *Card Slot*)
2. *ArrayTable* component's *Cell Size* setting. This is a physical size of a single table cell.
3. How many cells does one **Card** (**Act**) occupy on the table. This is changed via *CardViz* (*ActViz*) component's *Cell Count* variable with final cell count set to $(1,1) + 2*(x,y)$.

By playing with these values you can create margins between **Cards** on the table or make them overlap. There is currently no mechanism for any Z-sorting of overlapping **Cards**.