

# Cultistlike v2.2.0

grav2ity

May 1, 2025

## Contents

<b>1</b>	<b>Setup</b>	<b>3</b>
<b>2</b>	<b>Scriptable Objects</b>	<b>3</b>
2.1	Act . . . . .	4
2.1.1	Token . . . . .	4
2.1.2	Initial . . . . .	4
2.1.3	Tests . . . . .	4
2.1.4	And . . . . .	4
2.1.5	Or . . . . .	4
2.1.6	Fragments . . . . .	4
2.1.7	On Complete . . . . .	4
2.1.8	Furthermore . . . . .	4
2.1.9	Ignore Global Slots . . . . .	5
2.1.10	Slots . . . . .	5
2.1.11	Alt Acts . . . . .	5
2.1.12	Next Acts . . . . .	5
2.1.13	Spawned Acts . . . . .	6
2.1.14	On Spawn . . . . .	6
2.2	Aspect / Card . . . . .	6
2.2.1	Color . . . . .	6
2.2.2	Hidden . . . . .	6
2.2.3	Fragments . . . . .	6
2.2.4	Rules . . . . .	6
2.2.5	Slots . . . . .	6
2.2.6	Decay To (Card only) . . . . .	6
2.2.7	Lifetime (Card only) . . . . .	7
2.3	Deck . . . . .	7

2.3.1	Fragments . . . . .	7
2.3.2	Default Fragment . . . . .	7
2.3.3	Shuffle . . . . .	7
2.3.4	Replenish . . . . .	7
2.4	Slot . . . . .	7
2.4.1	Fragments . . . . .	7
2.4.2	Token . . . . .	7
2.4.3	Unique . . . . .	7
2.4.4	All Tokens . . . . .	7
2.4.5	All Acts . . . . .	8
2.4.6	Spawn Tests . . . . .	8
2.4.7	Spawn Rule . . . . .	8
2.4.8	Required . . . . .	8
2.4.9	Essential . . . . .	8
2.4.10	Forbidden . . . . .	8
2.4.11	Card Test . . . . .	8
2.4.12	Card Rule . . . . .	8
2.4.13	Accept All . . . . .	8
2.4.14	Grab . . . . .	9
2.4.15	Card Lock . . . . .	9
2.5	Rule . . . . .	9
2.5.1	Tests . . . . .	9
2.5.2	And . . . . .	9
2.5.3	Or . . . . .	9
2.5.4	Modifiers . . . . .	9
2.5.5	Furthermore . . . . .	9
2.6	Token . . . . .	10
2.6.1	Label . . . . .	10
2.6.2	Description . . . . .	10
2.6.3	Color . . . . .	10
2.6.4	Slot . . . . .	10
2.6.5	Dissolve . . . . .	10
2.6.6	Unique . . . . .	10
<b>3</b>	<b>Tests</b>	<b>10</b>
3.1	Card Test . . . . .	11
3.2	Location . . . . .	11
3.3	Op . . . . .	11
3.3.1	Random Challenge . . . . .	11
3.3.2	Random Clash . . . . .	11

<b>4</b>	<b>Modifiers</b>	<b>12</b>
4.1	Act Modifiers . . . . .	12
4.1.1	Adjust . . . . .	12
4.1.2	Grab . . . . .	12
4.2	Card Modifiers . . . . .	12
4.2.1	Fragment Additive . . . . .	13
4.2.2	Transform . . . . .	13
4.2.3	Decay . . . . .	13
4.3	Table Modifiers . . . . .	13
4.3.1	Spawn Act . . . . .	13
4.3.2	Spawn Token . . . . .	13
4.4	Path Modifiers . . . . .	13
4.4.1	Force Act . . . . .	13
4.4.2	Set Callback . . . . .	13
4.4.3	Callback . . . . .	14
4.4.4	Game Over . . . . .	14
4.5	Deck Modifiers . . . . .	14
<b>5</b>	<b>Matched Cards</b>	<b>14</b>
<b>6</b>	<b>Text Options</b>	<b>14</b>
<b>7</b>	<b>Changing Card Size</b>	<b>15</b>

## 1 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/Demos/

## 2 Scriptable Objects

Game logic is driven by Scriptable Objects. You can create new Objects with the right click Create menu. Objects need to be placed somewhere under Resources folder.

## **2.1 Act**

Initial Acts for a Token are evaluated in Scriptable Object's name alphabetical order.

### **2.1.1 Token**

Limits execution to this Token. Mandatory for initial and spawned Acts.

### **2.1.2 Initial**

First in the Act chain. Can be started by player pressing button.

### **2.1.3 Tests**

All the tests must pass to enter this Act. Matched Cards from Card tests will be available in the On Complete modifiers.

### **2.1.4 And**

All of the And Rules must pass to enter this Act. Modifiers are not applied. Matched Cards are not carried in nor out.

### **2.1.5 Or**

One of the Or Rules must pass to enter this Act. Modifiers are not applied. Matched Cards are not carried in nor out.

### **2.1.6 Fragments**

Fragments are added upon completing Act.

### **2.1.7 On Complete**

Modifiers run upon completing Act. See the Modifiers section.

### **2.1.8 Furthermore**

Rules to be run upon completing Act. Matched Cards are not carried in nor out.

### **2.1.9 Ignore Global Slots**

Only Slots from the list below will attempt to open while Act is running.

### **2.1.10 Slots**

Additional Slots that will attempt to open while Act is running. To see more than one Slot you would have to alter the Act window prefab and add slots to Run Slots list.

### **2.1.11 Alt Acts**

While running it is possible to switch to a different (Alt) Act. This will be triggered by un(sloting) Cards (if any running slots are open).

A list of potential Alt-Acts is considered.

- If Act Rule is set it must pass for the Act to remain in the list.
- If Act Rule is not set Act has Chance probability to remain in the list.
- If there is only one Act in the list Chance is ignored.
- If Random Alt is true this list will be randomly shuffled.

Above points are evaluated just after timer starts.

When user (un)slots Card from the running slots, entrance tests of Acts in the list will be evaluated. First Act in the list that passes its entrance tests will be the Alt-Act and window will display its description. If this switch happens and timer runs out it is as if the original Act never ran. Fragments, On Complete modifiers, Next Acts etc. from the Alt-Act will be used.

### **2.1.12 Next Acts**

Upon completion of the Act a list of potential next Acts is considered.

- If Act Rule is set it must pass for the Act to remain in the list.
- If Act Rule is not set Act has Chance probability to remain in the list.
- If there is only one Act in the list Chance is ignored.
- If Random Next is true this list will be randomly shuffled.

First Act in the potential Next Acts lists that passes its entrance tests will be the Next Act.

### **2.1.13 Spawned Acts**

Upon completion of the Act a list of Spawned Acts is considered.

- If Act Rule is set it must pass for the Act to remain in the list.
- If Act Rule is not set Act has Chance probability to remain in the list.

All Acts in the resulting list will be spawned inside their own Tokens. Make sure the the Acts have a Token field set.

### **2.1.14 On Spawn**

Rule that will be run on spawning this Act in a new Token.

## **2.2 Aspect / Card**

### **2.2.1 Color**

Solid color used when art is not set.

### **2.2.2 Hidden**

Do now show inside UI fragment bar.

### **2.2.3 Fragments**

Initial Fragments.

### **2.2.4 Rules**

Rules will be run on Act completion if Fragment is present.

### **2.2.5 Slots**

Slots that will attempt to open if Fragment is present.

### **2.2.6 Decay To (Card only)**

Whenever Card is created it will automatically Decay to / turn into the specified Card.

### **2.2.7 Lifetime (Card only)**

How long will it take for the Decay to complete.

## **2.3 Deck**

Currently Fragments need to be Cards.

### **2.3.1 Fragments**

Deck content.

### **2.3.2 Default Fragment**

Fragment to draw when Deck is empty.

### **2.3.3 Shuffle**

Randomize deck order.

### **2.3.4 Replenish**

Replenish fragments on exhaustion.

## **2.4 Slot**

### **2.4.1 Fragments**

Fragments will be added to the Act window whenever Card is slotted.

### **2.4.2 Token**

Token in which Slot will attempt to spawn.

### **2.4.3 Unique**

Only one instance of this Slot can be spawned per window.

### **2.4.4 All Tokens**

Attempt to spawn in all Tokens.

#### **2.4.5 All Acts**

Attempt to spawn in all running Acts.

#### **2.4.6 Spawn Tests**

All the Tests must pass for this Slot to spawn.

#### **2.4.7 Spawn Rule**

Rule must pass for this Slot to spawn.

#### **2.4.8 Required**

Card must have at least Count of one of the Required Fragments to be accepted in this Slot.

#### **2.4.9 Essential**

Card must have at least Count for all the Essential Fragments to be accepted in this Slot.

#### **2.4.10 Forbidden**

Card must have less than Count for every Forbidden Fragment to be accepted in this Slot.

#### **2.4.11 Card Test**

Additional tests that must pass for a Card to be accepted. This will not show in the tooltip.

#### **2.4.12 Card Rule**

Additional Rule that must pass for a Card to be accepted. This will not show in the tooltip.

#### **2.4.13 Accept All**

Allows to place all Cards in the Slot.



#### **2.4.14 Grab**

Slot will automatically grab Cards for itself.

#### **2.4.15 Card Lock**

Cannot remove Card from the Slot.

### **2.5 Rule**

Rules are convenience objects that group tests and modifiers.

- To evaluate a rule means to evaluate its tests without executing modifiers.
- To run a rule means to evaluate tests and only execute modifiers if tests passed.
- Rules can be used in many places but often only evaluation will take place.

#### **2.5.1 Tests**

All the tests must pass for this Rule to pass. Matched Cards from Card tests will be available in the Modifiers section.

#### **2.5.2 And**

All of the And Rules must pass for this Rule to pass. Modifiers are not applied. Matched Cards are not carried in nor out.

#### **2.5.3 Or**

One of the Or Rules must pass for this Rules to pass. Modifiers are not applied. Matched Cards are not carried in nor out.

#### **2.5.4 Modifiers**

See the Modifiers section.

#### **2.5.5 Furthermore**

Additional Rules to be run if this Rule passed. Matched Cards are not carried in nor out.

## **2.6 Token**

### **2.6.1 Label**

Label to display when no Act is running.

### **2.6.2 Description**

Description to display when no Act is running.

### **2.6.3 Color**

Solid color used when art is not set.

### **2.6.4 Slot**

First Slot to open for this Token when no Act is running.

### **2.6.5 Dissolve**

Destroy Token after completing last Act.

### **2.6.6 Unique**

Only one Token of this type can be on the table at any given time.

## **3 Tests**

- Tests are arithmetic (in)equalities.
- If Fragment 2 is not set the amount of Fragment 1 is compared against the Constant value.
- If Fragment 2 is set the amount of Fragment 1 is compared against 'Constant \* the amount of Fragment 2'.
- Loc 1 and Loc 2 determine where to count the specified Fragments.
- If Fragment is Aspect all the aspect present in that location will be counted (aspect from Cards + unbound aspect).
- If Fragment is Card amount of that Card type will be counted.

### **3.1 Card Test**

If Card Test option is selected:

- Test is applied on a per Card basis. i.e. it tests for a presence of a Card with Aspect value less/equal/more.
- After evaluating the test all the Cards that passed it become a new Matched Cards list.
- You can slice the MatchedCards list to contain only the n first Cards by setting Fragment 1 to None

### **3.2 Location**

- Scope - count in the current scope. This will be content of the Act window in most cases.
- Matched Cards - count among the Matched Cards. In the topmost test these are always all Cards present in the scope.
- Table - Cards on the table (including moving towards the table) = [everything under the Table Object in scene hierarchy].
- Free - every Card that can be picked up by the player (including Card currently being dragged and Cards in non-locked Slots).
- Anywhere - [everything under the Root object in scene hierarchy].

### **3.3 Op**

#### **3.3.1 Random Challenge**

Each Fragment gives a Constant chance of passing the test. For example with Fragment 1 set to XYZ and Constant set to 10, with 3 XYZ found there will be 30% chance of passing the test.

#### **3.3.2 Random Clash**

If there is 5 of Fragment 1 and 5 of Fragment 2 there is 50% chance of passing the test. If there is 10 of Fragment 1 and 90 of Fragment 2 there is 10% chance of passing the test.

## 4 Modifiers

### 4.1 Act Modifiers

Act Modifiers add/remove Fragments to the current Act window's scope.

- Level - amount to be added / removed / grabbed.
- Reference - if Reference is not set Level acts as a value. If Reference is set final Level will equal 'Level \* the amount of Reference'.

#### 4.1.1 Adjust

- If Fragment is Aspect this will add/remove unbound (not bound to a Card) Aspect.
- If Fragment is Card this will add freshly minted Cards (with default fragments) or remove and destroy up to Level existing Cards (with their current fragments).
- If Fragment is special (`__MatchedCards`) this will create Level copies of those Cards (with their current fragments) or remove and destroy them.

#### 4.1.2 Grab

- If Fragment is Aspect this will grab up to Level Cards having that Aspect.
- If Fragment is Card this will grab up to Level Cards of that type.
- If Fragment is special (`__MatchedCards`) this will grab up to Level of those Cards.
- If Level is set to 0 this will grab all matching fragments.

### 4.2 Card Modifiers

Card Modifiers target Cards in current Act window's scope.

- If Target is Aspect this will target all Cards that have that Aspect.
- If Target is Card this will target all Cards of that type.
- If Target is special (`__MatchedCards`) this will target those Cards.

#### **4.2.1 Fragment Additive**

Add/remove Level of Fragments from the Target Cards.

- Fragment should be an Aspect.

#### **4.2.2 Transform**

Transforms Target into new Card type. Target Card's aspects are kept. New Card's aspects are ignored.

- Fragment must be a Card.
- Level must be  $> 0$ .

#### **4.2.3 Decay**

Decays Target. If Fragment is set this will decay Target to Fragment in Level seconds. Otherwise this will use Target Card's decay settings.

- Fragment must be a Card or None.

### **4.3 Table Modifiers**

#### **4.3.1 Spawn Act**

Spawns a new Token and starts Act. Act's On Spawn Rule will be run. Make sure Act's Token field is set.

#### **4.3.2 Spawn Token**

Spawns a new Token via Act's Token field.

### **4.4 Path Modifiers**

#### **4.4.1 Force Act**

Unconditionally runs Act.

#### **4.4.2 Set Callback**

Save an Act to be called on later. This will persist until another Set Callback.

#### 4.4.3 Callback

Run previously set callback Act. This takes precedence over normal Next Acts list. Callback Act's entrance tests must still pass, otherwise it reverts to Next Acts list.

#### 4.4.4 Game Over

Clears the table and spawns Act.

### 4.5 Deck Modifiers

## 5 Matched Cards

For the first (topmost) test the Matched Cards are simply all the Cards inside the scope. For any subsequent test Matched Cards are those that passed Card Tests above it. In this way you can progressively filter Cards.

Inside modifiers you can reference the Matched Cards by choosing special `___MatchedCards` fragment. Matched Cards are only valid within a single Unity Editor Act or Rule view i.e. Matched Cards will not carry in nor out of And/Or/Furthermore Rules.

#### WARNING

Matched Cards are evaluated at the start but modifiers are executed after timer completes. If you are matching Cards outside of the current scope (e.g. on the table) they might have been destroyed by the end of the Act. Inside an Act that takes non-zero time to complete do not run modifiers on Cards matched outside of the current scope.

## 6 Text Options

Inside (End) Text fields:

- `[SOfilename]` will be replaced by the count of that Fragment.
- `[MC]` will be replaced by the label of the first Matched Card.
- `{[SOfilename op number] text}` will be replaced by text if inequality holds and empty string otherwise.

Where

- SOfilename stands for filename of the Fragment Scriptable Object.
- op is one of (>,<==,!=,>,<).
- number is a positive integer.
- text can contain other [] expressions but nested {} are not supported.

## 7 Changing Card Size

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

- Actual physical size of the prefabs (Card, Card Slot, Token)
- ArrayTable component's Cell Size setting. This is a physical size of a single table cell.
- How many cells does one Card (Token) occupy on the table. This is changed via CardViz (TokenViz) 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.