

CARD MAKER



TCG - Card Maker

Documentation

1.0.0

18-03-2025



| | |
|------------------------|-----------|
| Overview | 2 |
| Getting Started | 2 |
| Your Saved Cards | 3 |
| CardSO Object | 4 |
| Settings Object | 5 |
| CardEditorPanel | 6 |
| Special Words Object | 8 |
| Customization | 9 |
| CardSO | 9 |
| Traits | 10 |
| Other Scripts | 12 |
| ImageScroller | 12 |
| DeckCardView | 12 |
| TabButton, TabGroup | 12 |
| SpecialWordsPanel | 12 |
| TraitsPanel | 12 |
| DeckPanel | 12 |

Overview

TCG - Card Maker simplifies card creation for your card game with an intuitive UI, instant previews, and easy customization: design, edit, and integrate cards effortlessly

Getting Started

The best way to learn how to work with the asset is investigating the CardMaker scene included.

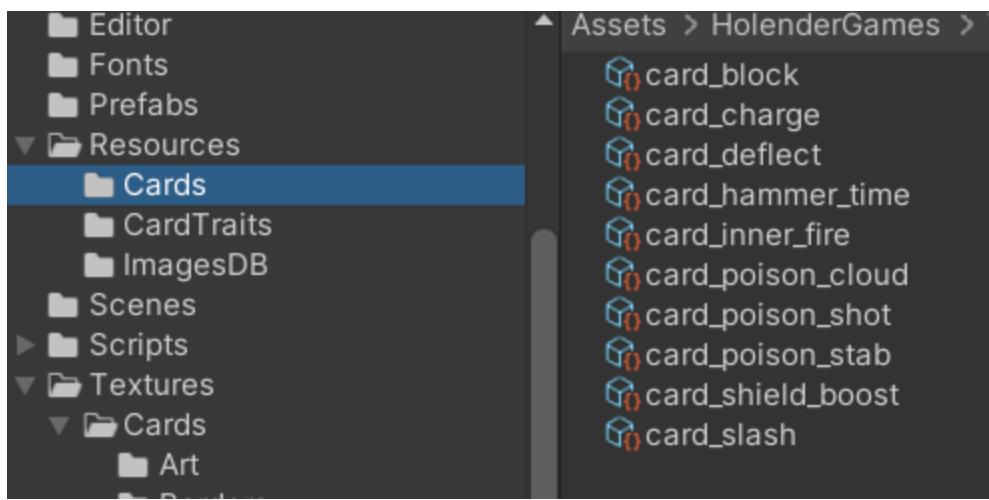
The main components and concepts of CardMaker are:

- **CardSO** - the main scriptable object for storing the cards data of the cards you creating in the tool.
- **Settings** - the main scriptable object to define settings for the CardMaker.
- **CardViewer**: The UI prefab that responsible for viewing the card data. You can modify it with your own art style and text placements
- **CardsDB** - a scriptable object which hold your deck of cards (auto managed)
- **Traits**: special attributes you can apply to a card. You can create your own by adding to the TraitType enum, and then by creating new scriptable objects of traits.
- **TraitsDB**: a scriptable object that hold a list of all available traits.
- **SpecialWords** - a list of words that are special terms in your card game (Damage, HP, Draw, Discard, HP, etc..) this words will be highlighted in the card's description.
- **ImagesDB** - a scriptable object that contains list of images to be used in the tool.
There is 2 being used: border images and art images.

Your Saved Cards

The cards you create and save will be saved under "Resources/Cards"

And also be automatically added to your deck file at "Resources/CardsDB.asset"

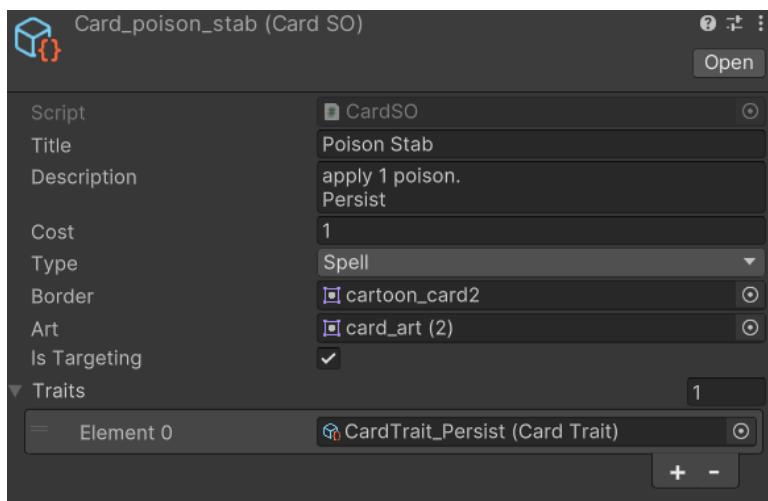


CardSO Object

This is edited and customized in the tool itself. No need to manually edit.

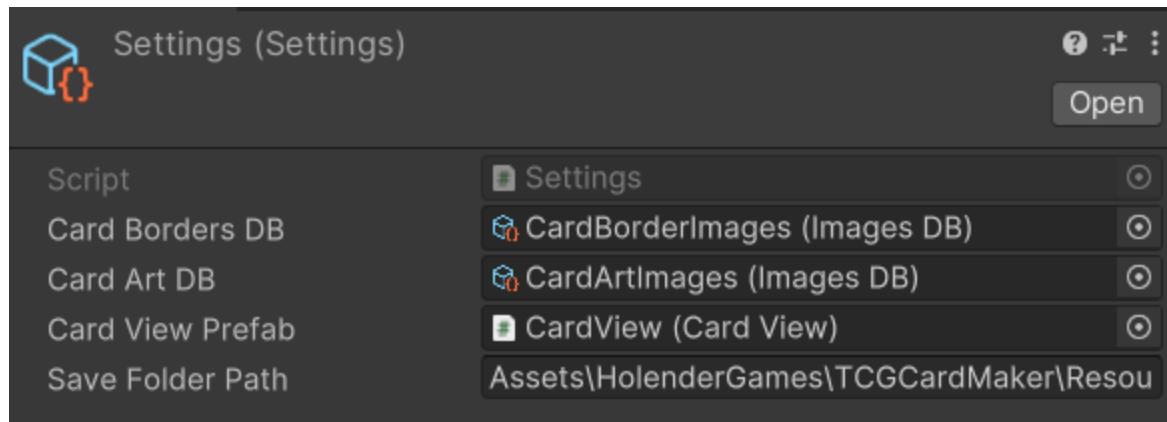
The card object will be created using the tool by saving to file in the main panel.

- Title: name of the card
- Description
- Cost: the cost to play the card (mana/gold/etc)
- Type: a category for cards type (spell/power/attack)
- Border: the image for the border of the card
- Art: the image for the specific card art
- IsTargeting: is this card targets an object in the game, or just being cast globally.
- Traits: list of traits (consume, destroy, etc..)



Settings Object

- Define here the object of your Card Border images DB
- Define here the object of your Card Art images DB
- The prefab to use to view your card
- The folder to save all resources (no need to change)

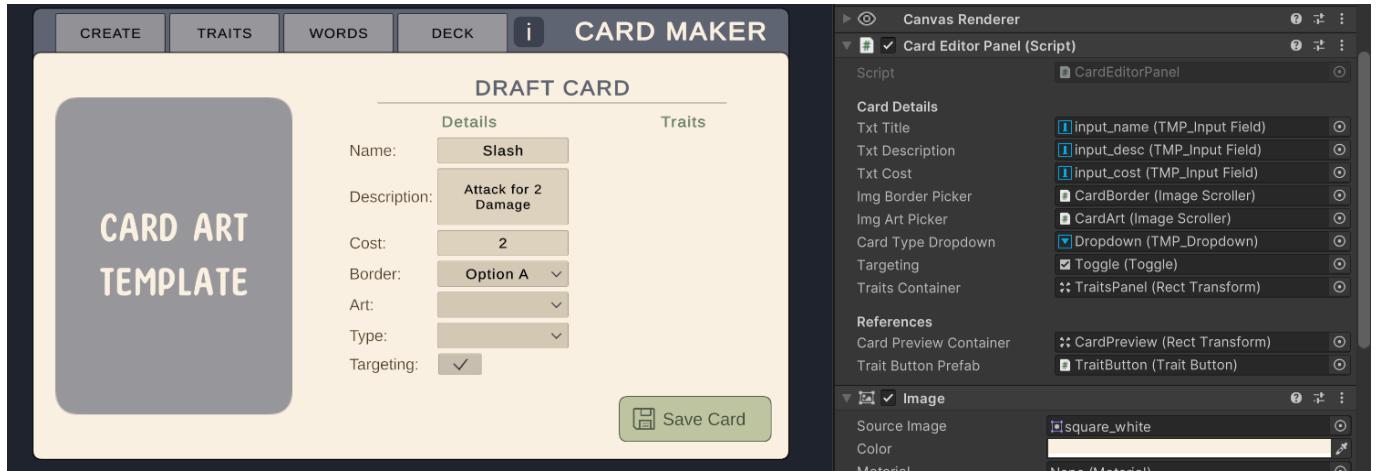


CardEditorPanel

The panel for creating and editing your cards

It holds and manages all the input fields of the user.

To customize it for your special card needs, you can add more fields (copy-paste), like additional cost, addition parameters, etc.



Make sure to update also the script to react to changes in the new fields you add. For example:

```
1 reference
private void UpdateCardType(int cardTypeIdx)
{
    draftCard.Type = (CardType)cardTypeIdx;
    cardView.SetData(draftCard);
}

1 reference
private void UpdateTargeting(bool arg0)
{
    draftCard.IsTargeting = targeting.isOn;
    cardView.SetData(draftCard);
}

1 reference
public void UpdateTitle(String title)
{
    draftCard.Title = title;
    cardView.SetData(draftCard);
}

1 reference
public void UpdateDescription(String desc)
{
    draftCard.Description = desc;
    cardView.SetData(draftCard);
}

1 reference
public void UpdateCost(String cost)
{
    draftCard.Cost = Convert.ToInt32(cost); ;
    cardView.SetData(draftCard);
}
```

Special Words Object

A list of words that are special terms in your card game (Damage, HP, Draw, Discard, HP, etc..) this words will be highlighted in the card's description.

This is edited and customized in the tool itself. No need to manually edit.

The screenshot shows the 'Special Words DB (Special Words)' window in RPG Maker MV. At the top, there is a script icon and the title 'Special Words DB (Special Words)'. To the right are buttons for '?', 'Edit', and 'Open'. Below the title, it says 'Script' and 'SpecialWords'. A counter indicates there are 12 items in the list. A section titled 'Words' is expanded, showing a list of 12 items, each consisting of a name on the left and a color-coded box on the right. The names are: Element 0 (Damage), Element 1 (HP), Element 2 (Poison), Element 3 (Discard), Element 4 (Consume), Element 5 (Persist), Element 6 (Destroy), Element 7 (Power), Element 8 (Spell), Element 9 (Attack), Element 10 (Block), and Element 11 (Heal). At the bottom, there is a 'Special Color' bar with a color swatch and a pencil icon for editing.

| Element | Color |
|------------|---------|
| Element 0 | Damage |
| Element 1 | HP |
| Element 2 | Poison |
| Element 3 | Discard |
| Element 4 | Consume |
| Element 5 | Persist |
| Element 6 | Destroy |
| Element 7 | Power |
| Element 8 | Spell |
| Element 9 | Attack |
| Element 10 | Block |
| Element 11 | Heal |

Customization

CardSO

To customize your card data, you can start by customizing the CardSO scriptable object to be able to hold and save the new data.

Pay attention! Update also the **CreateClone()** method to let the system be able to clone your card for editing purposes.

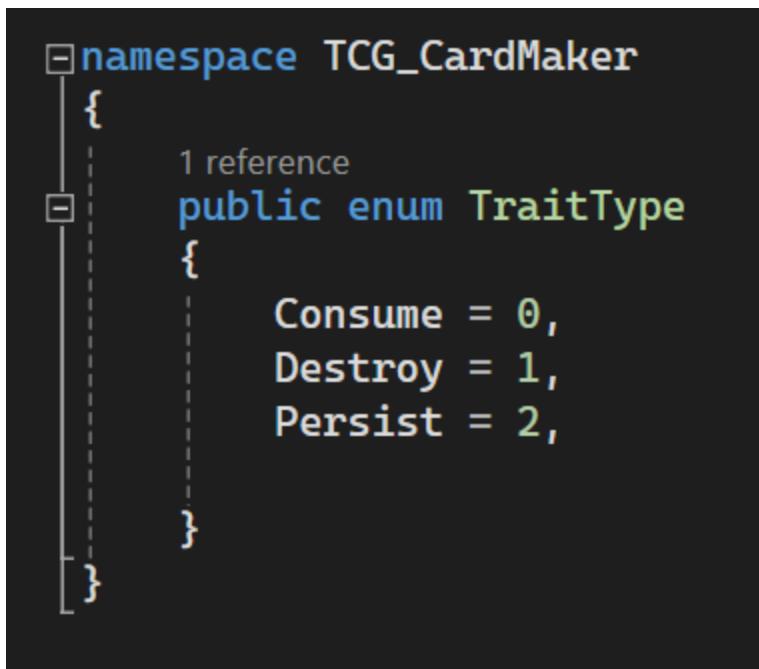
```
[CreateAssetMenu(fileName = "Card", menuName = "Create Card", order = 0)]
➊ Unity Script | 22 references
public class CardSO : ScriptableObject
{
    public string Title = "New Card";
    public string Description = "Describe the card abilities";
    public int Cost=2;
    public CardType Type = CardType.Attack;
    public Sprite Border;
    public Sprite Art;
    public bool IsTargeting;
    public List<CardTrait> Traits = new List<CardTrait>();

    2 references
    public CardSO CreateClone()
    {
        CardSO clone = ScriptableObject.CreateInstance<CardSO>();
        clone.Title = Title;
        clone.Description = Description;
        clone.Cost = Cost;
        clone.Type = Type;
        clone.Border = Border;
        clone.Art = Art;
        clone.IsTargeting = IsTargeting;
        clone.Traits = new List<CardTrait>(Traits);
        return clone;
    }
}
```

Traits

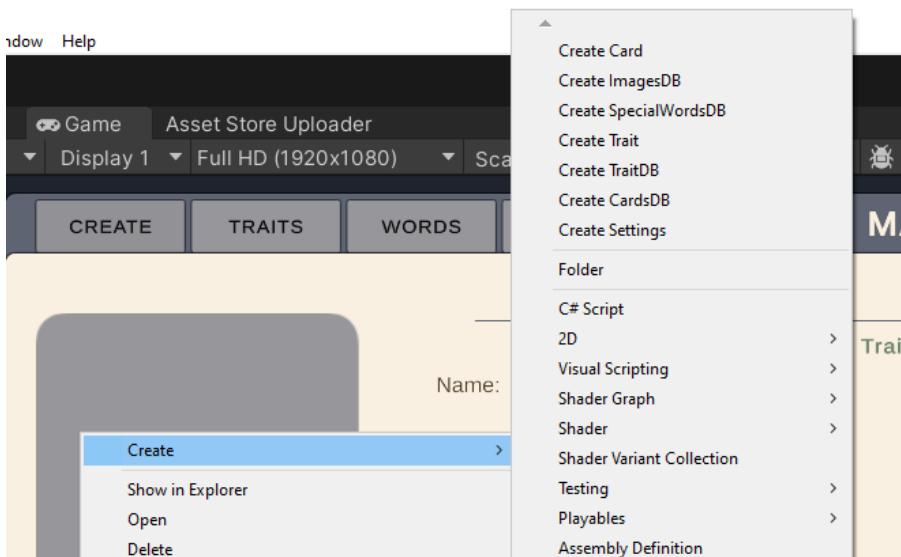
To customize and add new traits:

Step 1: add new type here:

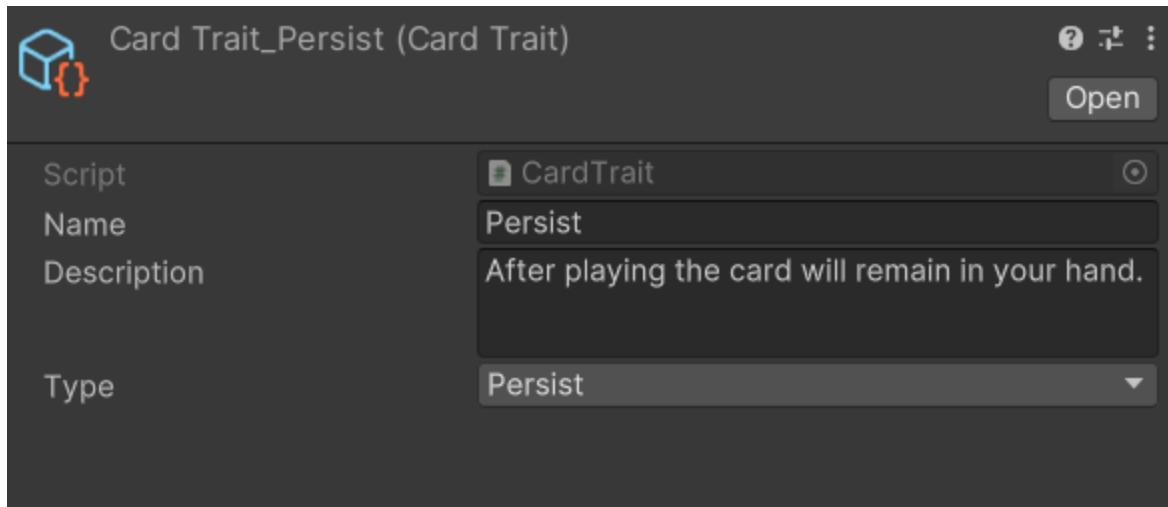


```
namespace TCG_CardMaker
{
    public enum TraitType
    {
        Consume = 0,
        Destroy = 1,
        Persist = 2,
    }
}
```

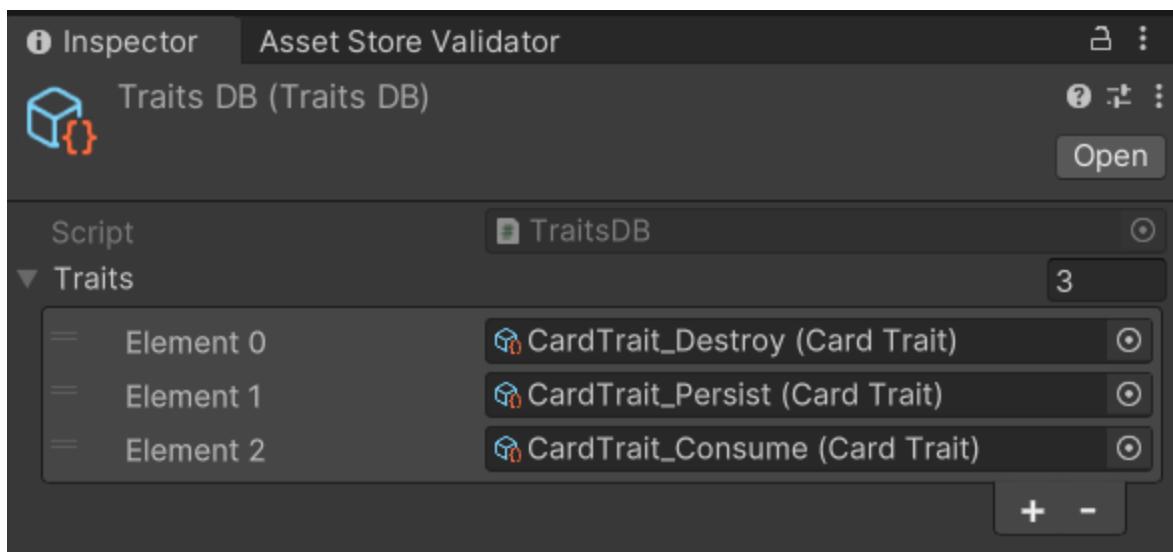
Step 2: create new scriptable object from menu:



Step 3: edit your new trait:



Step 4: add to **TraitsDB**:



Other Scripts

ImageScroller

A dropdown menu to show and select images from a list

DeckCardView

Wraps the CardView object and add a context menu for delete/edit card

TabButton, TabGroup

Handles the tabs and panels

SpecialWordsPanel

Show the special words and allows edit and save.

TraitsPanel

Show the currently configured Traits.

DeckPanel

Shows all the cards created and allows deletion and edit of each card