Summary

Introduction

An inventory of card game, but can be changed to of another type.

The inventory contains the script "ResizeGrid", which will adjust the size of the cards according to the mobile.

Along with the inventory has a mini store, where the elements are acquired through percentage.

For the card to be drawn by percentage, a calculation was added where that "increases" the card amount according to the percentage.

About the Element.
Control of The Cards.

GameObject => ControlCards

Script -> ControlOfTheCards

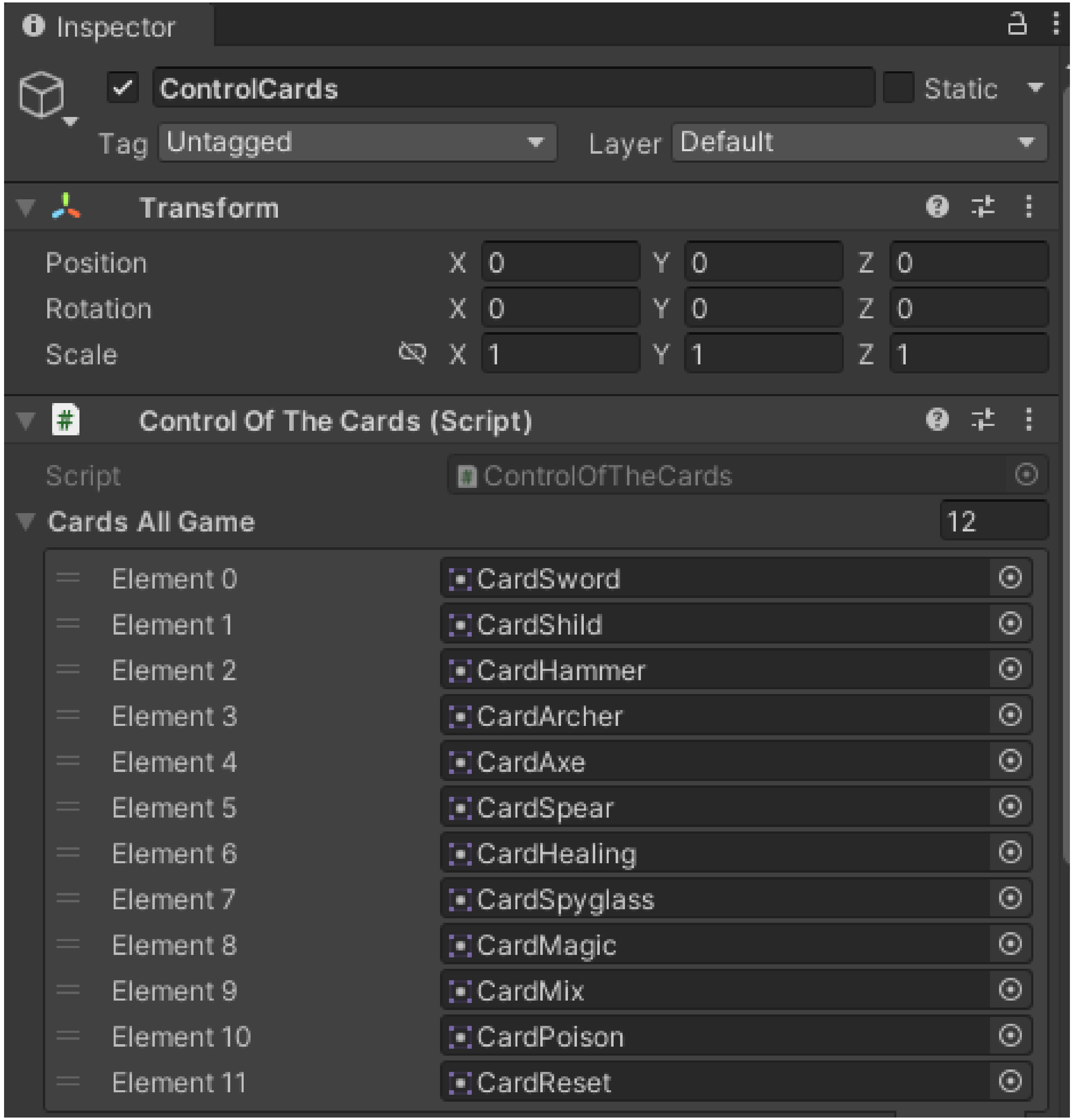
List -> CardsAllGame

How to change the amount of element or the type of element?

To change the quantity just add or remove elements from this list.

In this video is shown how to change the list in a basic way.

https://youtu.be/itXzY2y9q-k



☑ ControlCards □ Static

Tag Untagged
Layer Default

	X	Y	Z
Position	0	0	0
Rotation	0	0	0
Scale	1	1	1

Script ControlOfTheCards

Cards All Game 12

Element 0	CardSword
Element 1	CardShild
Element 2	CardHammer
Element 3	CardArcher
Element 4	CardAxe
Element 5	CardSpear
Element 6	CardHealing
Element 7	CardSpyglass
Element 8	CardMagic
Element 9	CardMix
Element 10	CardPoison
Element 11	CardReset

Percentage of cards. Buy Manager.

GameObject => BuyManager

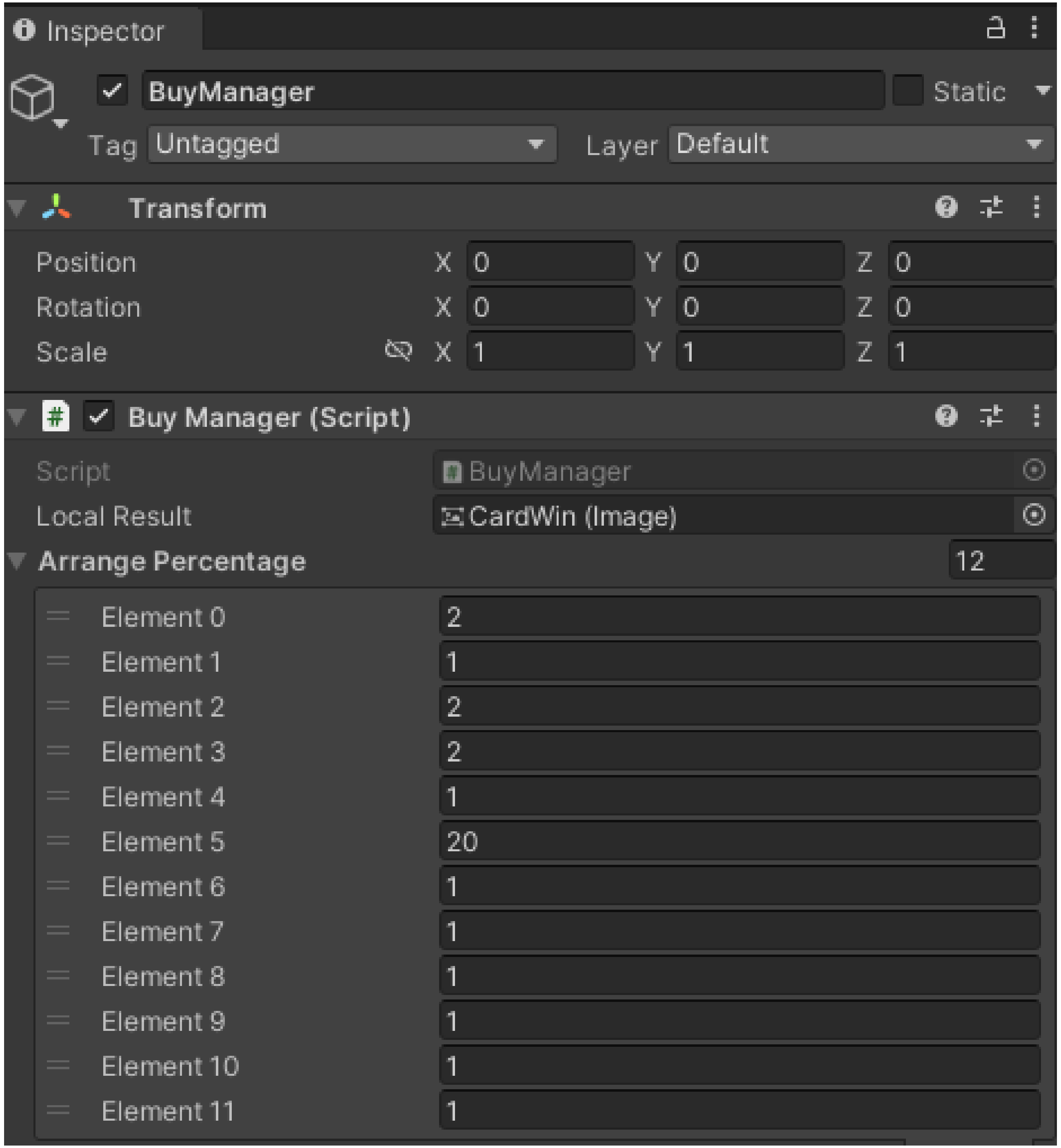
Script -> BuyManager

List -> ArrangePercentage

This list is to define the percentage of each element, the sum of all elements is 100%. If it exceeds or does not reach 100% the value will be divided for all elements.

All other lists are filled automatically.

This list has been set to have the same amount of "CardsAllGame" elements. If elements are missing, it will be filled in with the value 1. If you overtake, the last value will be removed until the same amount is.



☑ BuyManager □ Static

Tag Untagged Layer Default

Position X 0 Y 0 Z 0

Rotation X 0 Y 0 Z 0

Scale X 1 Y 1 Z 1

Script BuyManager

Local Result CardWin (Image)

Arrange Percentage 12

Element 0	2
Element 1	1
Element 2	2
Element 3	2
Element 4	1
Element 5	20
Element 6	1
Element 7	1
Element 8	1
Element 9	1
Element 10	1
Element 11	1

Card Size.
Resize Grid.

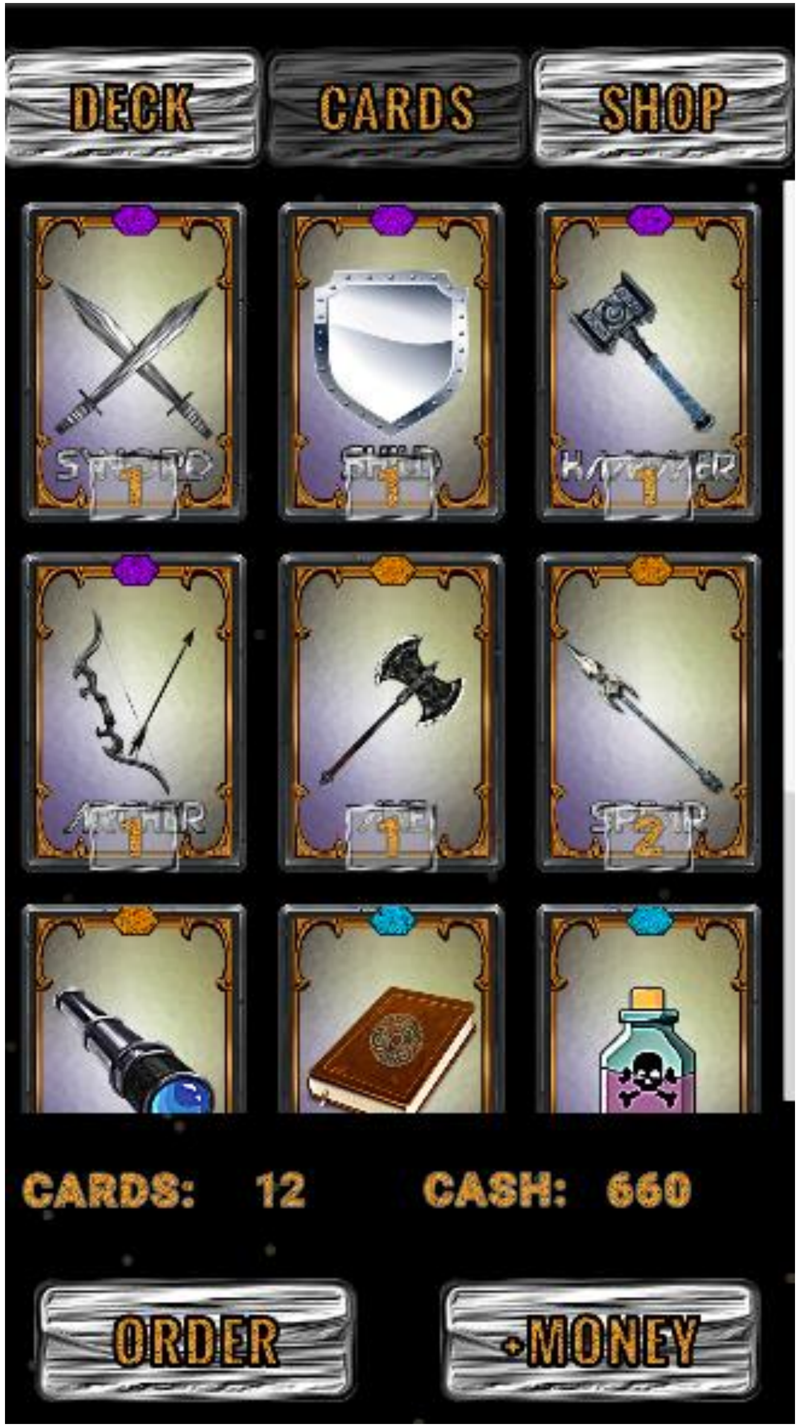
GameObject => Canvas > ScrollView > Viewport > Content

Script => ResizeGrid

This script has the function of changing the size of the elements within the grid.

The first variable, "_numberOfElements", defines the amount of elements per line (Horizontal).

The second variable, "_elementHeight", sets the size of the element in vertical.



Card Prefab. Part 1 Inventory Card.

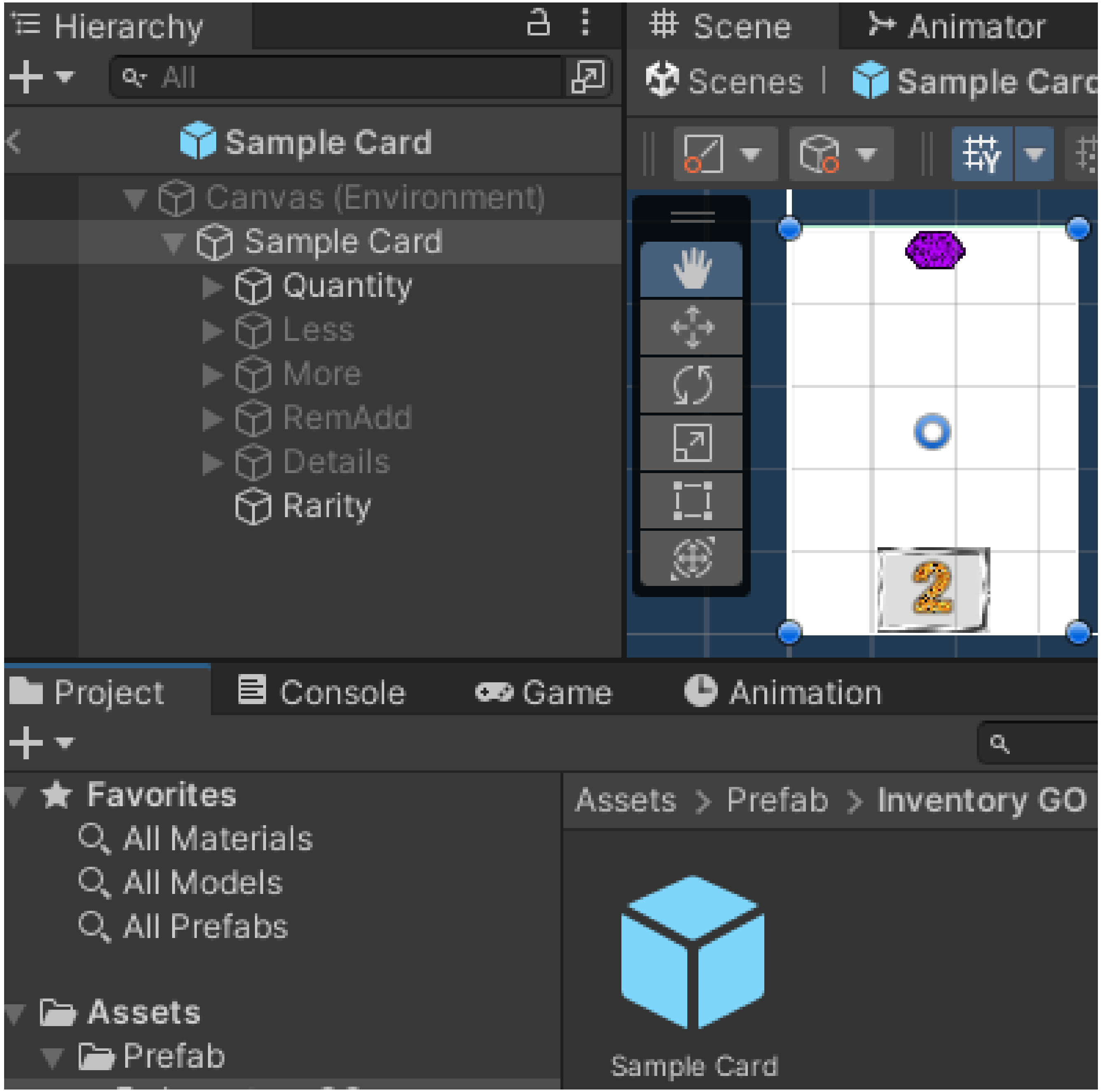
Inside the => Prefab > Inventory GO folder

Script => InventoryCard

Prefab is to represent the purchased elements. Inside the prefab has the child objects that have its functionalities.

Quantity: Shows how many cards you have in the open topic. Different being the topic of the deck and all cards purchased

Less/More: Decreases and Increases the number of cards in the deck. The limit set in the script is 5 or the amount of cards purchased.



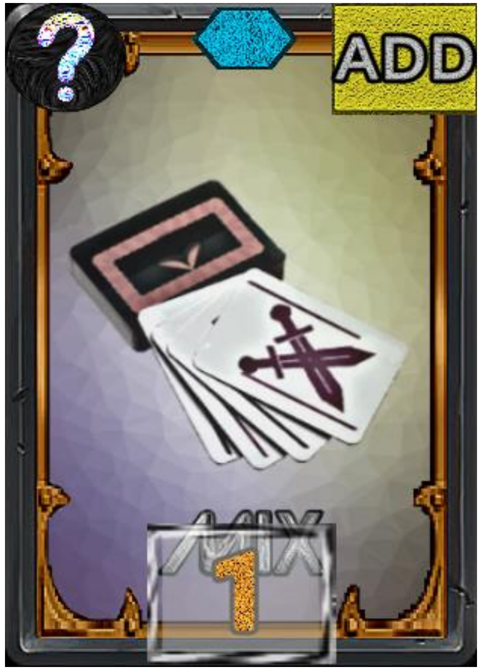
Card Prefab. Part 2
Inventory Card.

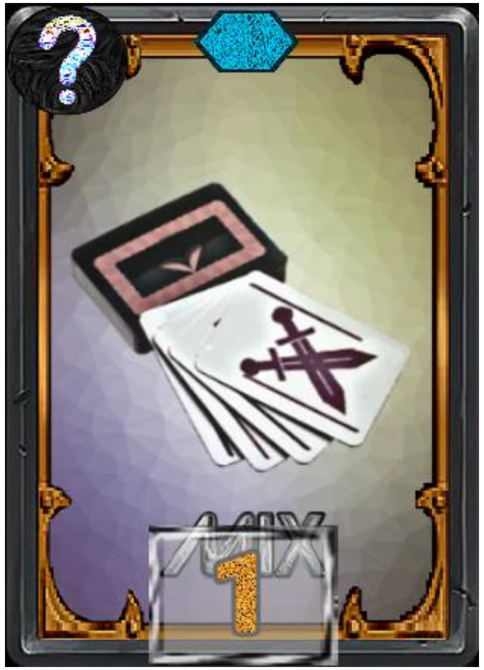
Add/Remove: Add or remove the card from the deck. When you press Add, after purchasing in the store, the add button is disabled. Going in the deck the card will be shown in the instantiated elements with option to remove/decrease/add/details.

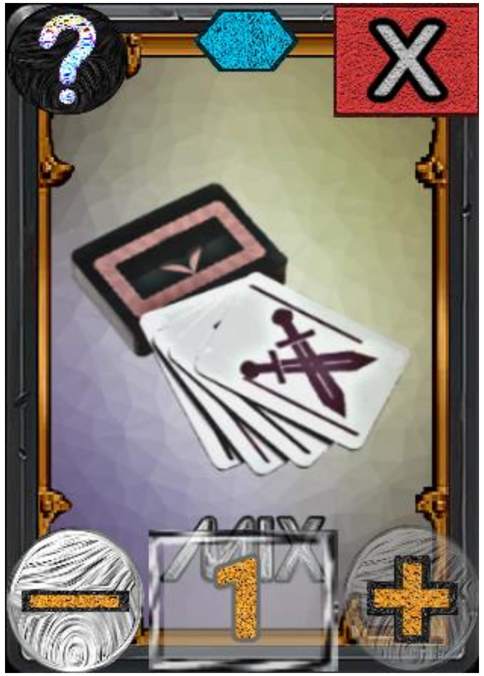
Details: Shows the selected card information.

The detail sequence is the same as the "CardsAllGame" list. If the position of the list changes, modify the order on the "InventoryCard" script switch, "InfoTypeCard" method on line 153

Note about less/more: If you can't make the call function, the button will have your interaction turned off. Example: Press to add card if in your stock is at the limit.





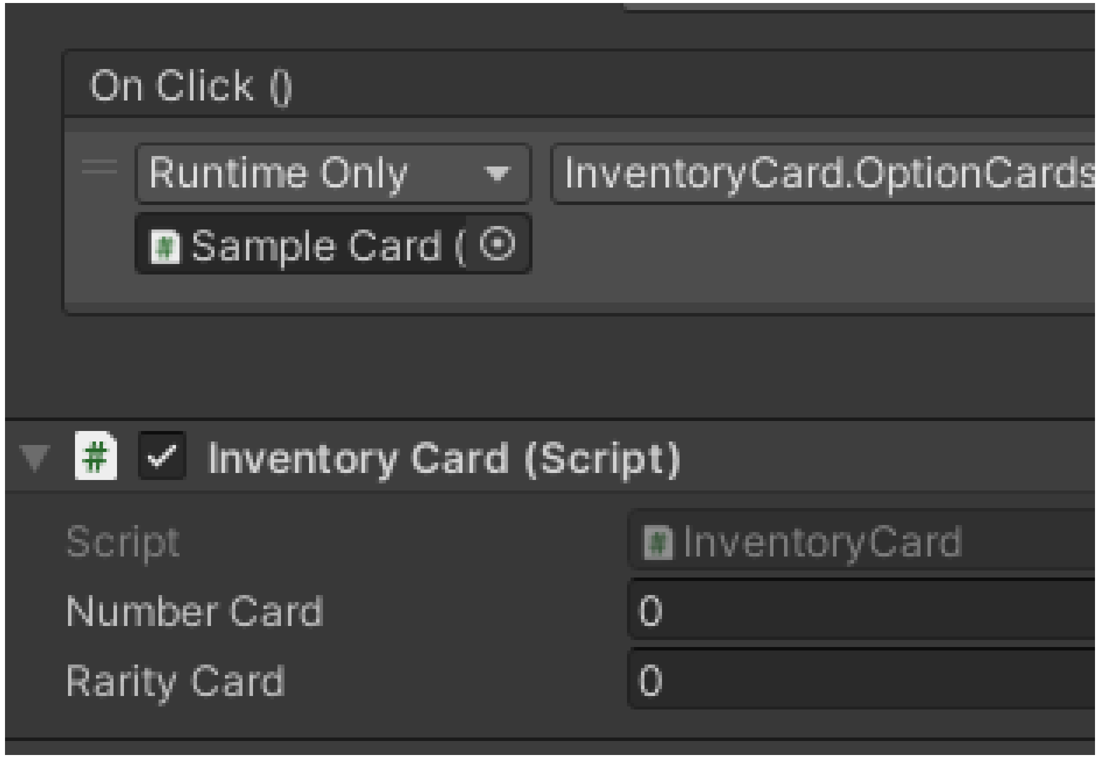


Card Code. Part 1
Script.

All scripts except "WhatScenario" are commented out. In this script "InventoryCard" has two fundamental variables.

NumberCard that defines the card number, is the same index number of the list "CardsAllGame", that number is automatically added according to the list "CardsAllGame".

RarityCard is the rarity of the card, this number is added automatically following the list "RarityOfCards". To turn off automatic rarity mode, simply comment out line 75 to 80.



On Click ()

Runtime Only

InventoryCard.OptionCards

Sample Card (

Inventory Card (Script)

Script: InventoryCard

Number Card: 0

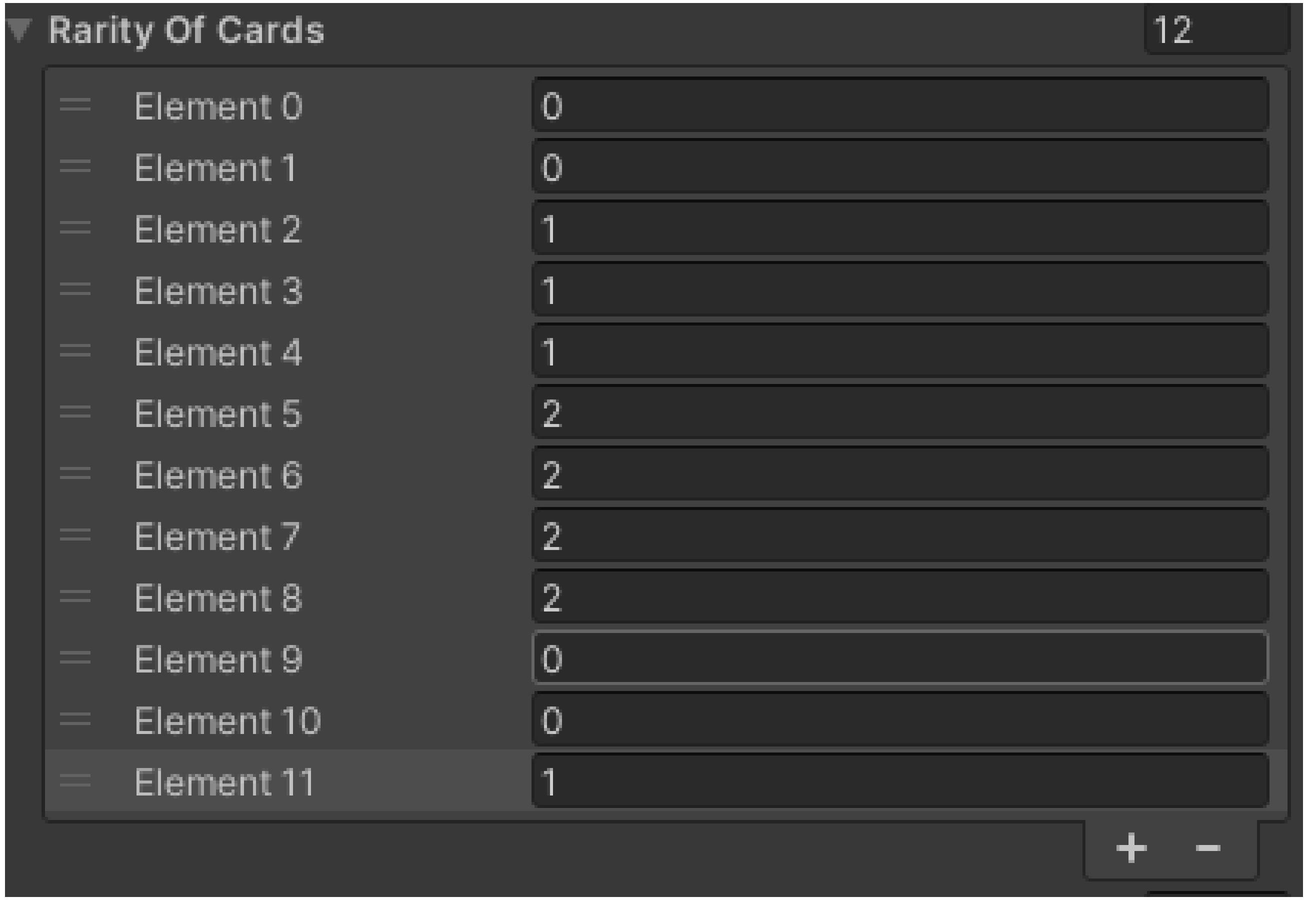
Rarity Card: 0

Card Code. Part 2
Adding manually.

In the "ControlOfTheCards" script, line 82 is specified as manually placing by code.

You can put the rarity number by the Inspector as well, as shown in the image on the right.

In the list displayed in the Inspector, it shows the number of elements next to the name, where you can change to the desired quantity. Below is the "more" and "less" sign where you can add or remove elements.



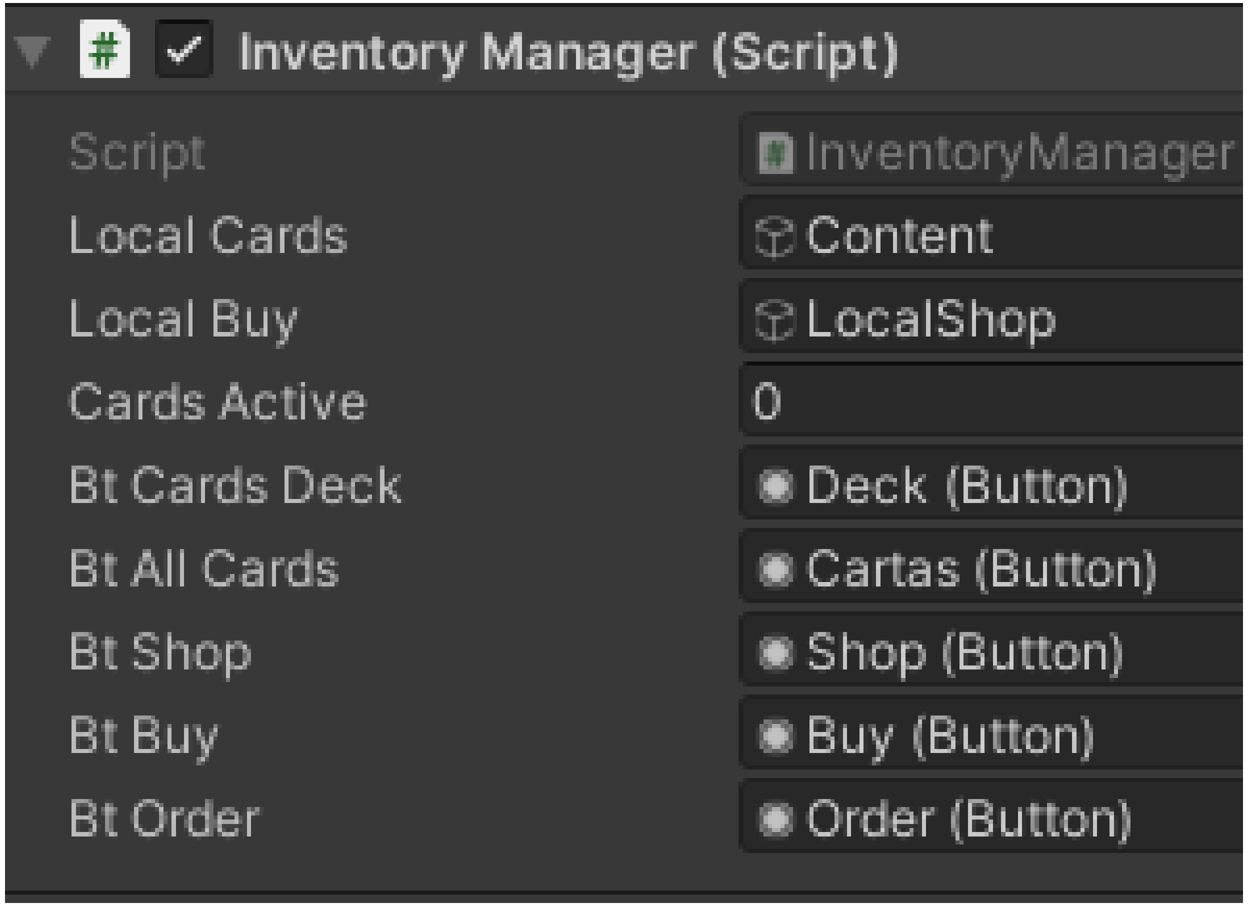
Rarity Of Cards 12

Element 0	0
Element 1	0
Element 2	1
Element 3	1
Element 4	1
Element 5	2
Element 6	2
Element 7	2
Element 8	2
Element 9	0
Element 10	0
Element 11	1

Inventory Manager. Adding manually.

"InventoryManager" is the script that has inventory control. Activation of buttons, Instantiation of cards and change of tab

The two "SpawnCards" methods have the settings for how cards are instantiated (Cards and child objects). By modifying the 84 line method ("SpawnCardsAll"), you change the cards purchased from the store. By modifying the 98 line method ("SpawnCardsDeck"), you change the cards in the deck.

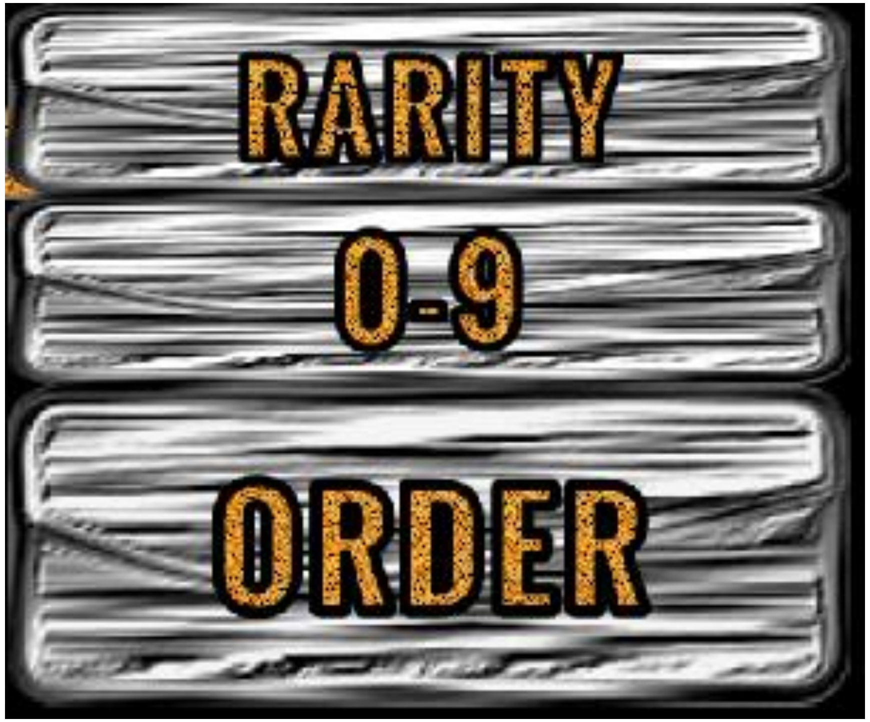


Inventory Manager (Script)

Script	InventoryManager
Local Cards	Content
Local Buy	LocalShop
Cards Active	0
Bt Cards Deck	Deck (Button)
Bt All Cards	Cartas (Button)
Bt Shop	Shop (Button)
Bt Buy	Buy (Button)
Bt Order	Order (Button)

Organize Cards.
Bubble Sort.

The technique used to rearrange the elements is the "Bubble Sort". The script comes with the function of organizing the elements by rarity or amount of repetitions. To do this it copies the information from the elements, deletes the ones that have been instantiated and instantiates again but following the chosen sequence.



What Scenario.
Scenario Control.

This script has been added to make it easier to use the Inventory tool. With this script you can find in which scenario is in the order of "Build Settings", block certain functions of the script to not work if it is not in your scenario.

Let's put an example. Whenever you start the game, one of your script scans for a GameObject with a certain name, but you changed the scenario script and it no longer finds that GameObject, you will find this GameObject only in scenario 4. To narrow this search, you would add the command:
if(WhatScenario.instance.Phase == 2)

As shown in the script "ControlOfTheCards" on line 50.

How to use?
Use the elements in the game.

How can I use the elements I purchase in my game? The "ControlOfTheCards" script, responsible for saving the elements, contains Singleton, which will make it easier to use in other scenarios. Then just add the scenario in Build Setting and after tinkering with inventory, you can go to another scenario that the script "ControlOfTheCards" will follow.