

KINGS

Card Swiping Decision Game Asset



V 1.55

Thank you for purchasing this asset!

If you encounter any errors / bugs, want to suggest new features/improvements or if anything is unclear (after you have read this documentation ;) do not hesitate to contact us:

support@km-games.com

**If you like our asset and want to support us,
please leave a review at the Unity Asset Store for us. Thanks!**

Content

Getting started.....	4
How to Play.....	4
Modifying the Game	4
EventScript.....	5
Card Stack (in Scripts)	9
Swipe Script (in Scripts)	10
Value Definitions (in Project folder scripts)	11
Value Script (in Values).....	12
Value Manager (in Values).....	13
Value Dependent Enable Script (in Values)	13
Achievements (in Scripts).....	14
Game Logger (in GameLogger).....	15
Value Dependent Game Logs.....	15
Global Message Event System.....	16
Score Calculator	17
Kings Level Up (in Score Calculator).....	18
Country Name Generator (in Values).....	19
High Scores (in MenuCanvas → HighscorePanel).....	20
Value Dependent Conditional Image (King Image).....	20
SecurePlayerPrefs (in scripts Project folder)	21
Translation Manager (in Scripts)	21
Unity Ads	22
Add Value to Value.....	23
Set Value	23
Change Value over time	24
Value change preview	24
Google Play /iOS Leaderboard and Achievements.....	25
Text Mesh Pro Support	27
Value Dependent Images.....	27
Value Dependent Events	28
Value Math	28
Value Dependent Text	29
Keyboard to Event (in Scripts).....	29
Kings Cards Import/Export (ImEx)	30
Card Styles	31
First Start Panel.....	32
Typewriter Text Effect	32

Reset Player Prefs (in Scripts)	32
Restart Game Button	33
Music Player	33
Background changer / Persistent Images	33
Support for Hyper Card asset.....	34
Support for Shadero Sprite asset.....	34
Unity Video Player.....	34
Google Admob support	35
Unity IAP support	36
Change Value script	37
Make Changes script.....	37
Inventory	38
Dictionary.....	40
Variables / Text Replacement.....	41
Quests.....	42
Timeline	44
Conditional Action List.....	45
Result Cheat (in Scripts)	46
Delayed Events	46
Update 1.55: Safe Area Helper (by Crystal Pug)	46
Update 1.55: Orientation Handler	47
Update 1.55: Ratio Handler	47

Getting started

- create a new Unity Project (3D) and import the Kings Game Asset
- open the „Game“-Scene in „Kings“-Project folder and press Play
- the game should now run
- this asset was built and tested with Unity 2018.3.7f1 and 2019.1.1f1, if any error occurs, we recommend to try it with these versions of Unity
- read this documentation to get an overview of the features and how to use them
- a detailed getting started tutorial can be found [here](#)

How to Play

- in the game example you are the king (or queen) and have to make decisions by swiping left or right, to try to keep the 4 “factions”: Army, People, Religion and Money in Balance.
- if one faction reaches 0 you have lost and the game is over
- if you swipe up the game menu will open, you can see your “secondary stats”, these stats are randomly generated for every new game and will affect some of your decisions.
- in the game menu you will also find the achievements, highscore and settings menu, to close the game menu you simply have to swipe down

Modifying the Game

- we recommend you to edit the existing **Game** scene, this scene has been made to show all features of this asset and is kept as simple as possible for easy editing / re-skinning
- edit the Values in the **ValueDefinition** Script to your needs
- create your own cards, here we also recommend to duplicate one of the existing prefabs in cards folder, modify the look to your liking and use this one then as your base card for creating new ones

You can find below a tutorial video on how to get started with Kings:



Please note: the part with editing the design of the card is obsolete in this video, please check this [written tutorial](#) for editing the design of the card in latest version.

EventScript

Event Script (Script)

Script

EventScript

Text Fields

Title Text

Question Text

Answer Left

Answer Right

Answer Up

Answer Down

Additional Texts

Size

Is Drawable

Is High Priority Card

Card Propability

Max Draws

Redraw Block Cnt

Conditions

Size

Swipe Type

Additional Choices

Results

Result Left

Modifiers

Value Changes

Size

Element 0

Element 1

Extras

Size

Follow Up Card

Result Right

Modifiers

Value Changes

Size

Element 0

Extras

Size

Follow Up Card

Change Value On Card Despawn

Size

Element 0

Change Extras On Card Despawn

Size

Target Type

Item Change

On Card Spawn ()

List is Empty

On Card Despawn ()

List is Empty

On Swipe Left ()

List is Empty

On Swipe Right ()

List is Empty

Army

The Army wants better Weapons

Sure

Your Weapons are good enough

0

☒

☐

1

100

5

0

Left Right

☐

Simple

2

Add

Money

-20

Add

Army

10

0

None (Game Object)

Simple

1

Add

Army

-30

0

None (Game Object)

1

Add

Years

1

1

Item

Sword (InventoryItem)

Add

1

On Card Spawn ()

List is Empty

On Card Despawn ()

List is Empty

On Swipe Left ()

List is Empty

On Swipe Right ()

List is Empty

This script is required by any card. It will let you setup the conditions and results of each card.

The **Text Fields** allow you to enter the Title, Questions and Answer for the card.

Is Drawable determines if the card can be randomly drawn from the CardStack. This should be always enabled, except for FollowUpCards.

Is High Priority Card will make sure the card is always drawn when the condition of the card is met.

Card Probability let's you allow to increase/decrease the probability that this card is drawn (when the condition is met). This doesn't affect High Priority Cards.

Max Draws let's you set up the value of how often this card can be drawn per game.

Redraw Block Cnt gives you the option to block a card for x draws after it has been drawn

Conditions let's you specify when this card will be drawn. You can select as many conditions as you want. If you select no condition the card can be drawn any time (when the Sub Stack Condition of the Card Stack Group is met)

Swipe Type let's you choose between **Left/Right** swipe and **Four Direction** swipe. This now allows you to swipe the card in all directions. (Example for four direction swipe: FourWaySwipe_SampleCard)

Please note: the four direction swipe cards require a different animator (SampleCard_4Dir)

Additional Choices if you want to have buttons with additional choices on your cards you can enables this checkbox. (example: MultichoiceCard)

Results

Results are separated in two parts, **Result Left** and **Result Right** depending of the swipe direction, or if **Four Direction** is select: Result Left, Result Right, Result Up and Result Down.

There are 4 different Results types:

Simple:

increases / decreases the selected values

In this example left result will decrease Money value by -20 and increase Army Value by 10 and will add 1 item (Sword) to the inventory. Right result will decrease Army value by -30.

The screenshot shows the 'Simple' result configuration. It is divided into 'Result Left' and 'Result Right' sections. 'Result Left' has a size of 2, with Element 0 adding -20 to Money and Element 1 adding 10 to Army. It also has an 'Extras' section with a size of 1, where a 'Sword (InventoryItem)' is added. 'Result Right' has a size of 1, with Element 0 adding -30 to Army. Both results have a 'Follow Up Card' set to 'None (Game Object)'.

Conditional:

depending on the player stats, the result is divided in two more parts: true and false

In this example the condition is Look value 40-100, if this is met, the Modifiers True will apply, increasing the Marriage value by 100 and Army, People, Religion and Money by 10, also a Follow Up Card "FollowUp_MarriedSuccess1" is linked, this Card will be drawn directly after this one. If the condition is not met, e.g. Player has Look Value of 30, the Modifier False will be applied, decreasing the Army, People and Religion Values by 10, also the Follow Up Card "FollowUp_MarriedFail1" will be drawn.

The screenshot shows the 'Conditional' result configuration. It has a size of 1 and a condition 'Look' with a range of 40 to 100. 'Modifiers True' have a size of 5, including adding 100 to Marriage and 10 to Army, People, Religion, and Money. A 'Follow Up Card' 'FollowUp_MarriedSuccess1' is linked. 'Modifiers False' have a size of 3, including subtracting 10 from Army, People, and Religion. A 'Follow Up Card' 'FollowUp_MarriedFail1' is linked.

Random Conditions:

This is similar to the Conditional type, except the condition values are not fixed, but randomly generated for each card draw.

In this example the random condition is Luck value, a random value between 1-100 will be generated, e.g. 57, if the Luck value of the Player is above this value the Modifiers True will apply, if it is below this value the Modifiers False will apply.

The screenshot shows the 'Random Conditions' result configuration. It has a size of 1 and a condition 'Luck' with a range of 1 to 100. 'Modifiers True' have a size of 4, including adding 10 to Army, People, Religion, and Money. 'Modifiers False' have a size of 4, including subtracting 20 from Army, People, Religion, and Money. Both have a 'Follow Up Card' set to 'None (Game Object)'.

Random:

one random element will be picked from a list of Value Changes

In this example one of the 3 Value Changes elements will be drawn, this means one of the 3 values Army, People or Religion will be increased by 20.

The screenshot shows the 'Random' result configuration. It has a size of 4 and is divided into three 'Element' sections. Each element has a size of 1 and a 'Value Change' of 20 for one of the stats: Army, People, or Religion. All elements have a 'Follow Up Card' set to 'None (Game Object)'.

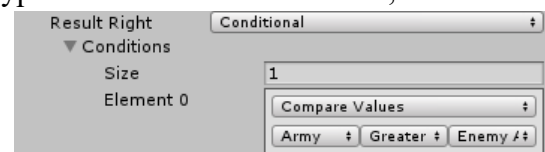
Change Value on Card Despawn this is used if you want to change a value, regardless of which direction the player swiped, for example to increase the value Year by 1.

On Card Spawn Event, can be used to trigger a Unity Event each time the card is spawned. You can use this for example if you want to trigger an achievement when this card is drawn.

On Card Despawn Event, can be used to trigger a Unity Event each time the card despawns. This is the recommend way to add Game logs, Achievement points etc...

Since **On Card Spawn** can be done multiple times, if the player quits the game and the active card has a **On Card Spawn** Event, which already triggered since the card is spawned, it will trigger again when the player resumes the game, because the card stack will automatically draw the last active card.

You have the option for conditions to choose between **Standard** and **Compare Values**. **Compare Values** allows you, as the name suggest, to compare two types of values with each other, instead like **Normal** with a fixed (range of) values. In this below example (card: War_firstAttack) your army value is compared with the enemy army value.

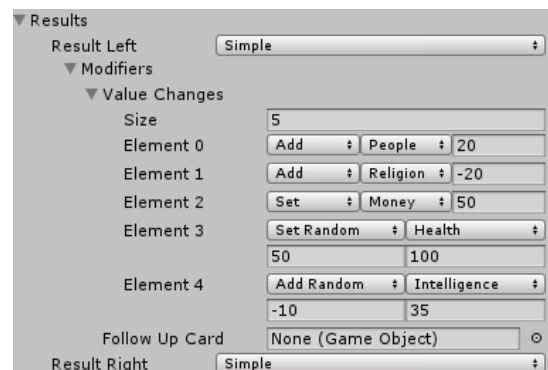


NEW in Update 1.44:

You have now the option to directly **SET** a value in the Event script to a fixed number.

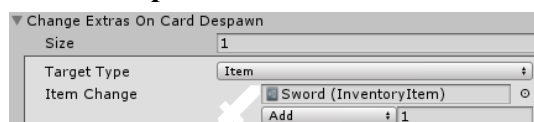
Also new is the option to **ADD RANDOM** or **SET RANDOM** value.

The **ADD** function is the standard method for adding or subtracting a value.



With Update 1.51 the functions **ADD RANDOM INT** and **SET RANDOM INT** has been added, they create whole numbers without fractions, e.g. 7 or 11 instead of 7,415 or 11,0531.

NEW in Update 1.50:



Extras & Change Extras On Card Despawn: Allows the newly added features like [Items](#), [Quests](#), [Timeline](#) and [Dictionary](#) to be changed from the EventScript. For convenience the [GameLogger](#) function can now also be changed directly from the EventScript.

Card Stack (in Scripts)

This script manages which card will be drawn. All cards need to be linked into this script.

If you check the **Verbose** toggle you can display additional information in Play Mode which can be useful for debugging.

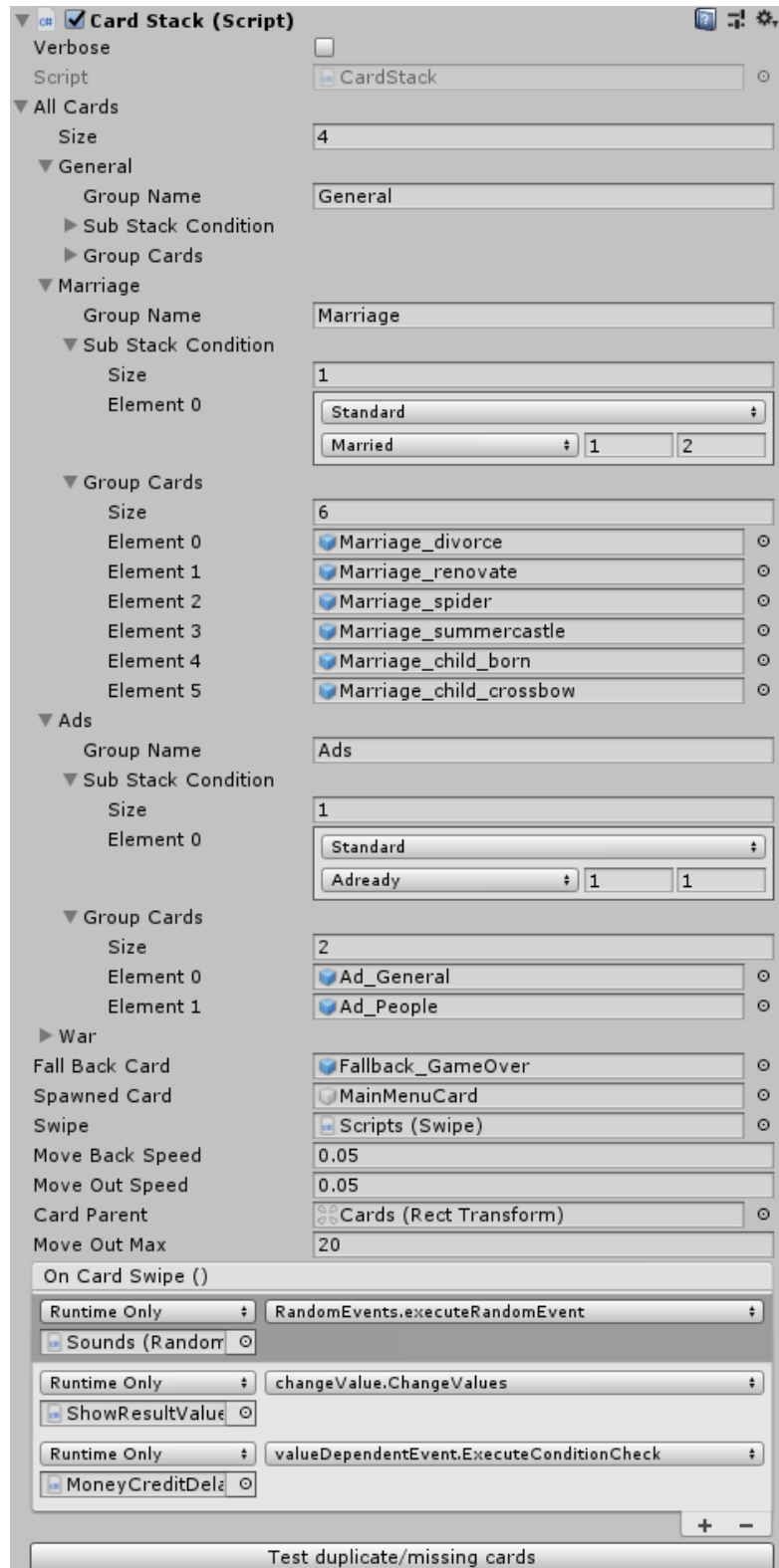
All Cards Size enter the number of how many different groups you want. In the above example we have 4 groups: General, Marriage, Ads, War.

In **Sub Stack Condition** you can choose a condition which is applied for this group and all cards in it, in this example the “Marriage” group has condition “Married Value 1-1” this means that the cards in this group can only be drawn if the player is married (The Value is 0 if the player is not married). For the “General” group you should select a value that is always there with a range that will always be met, e.g. 0-100. Update 1.20: You can now have more than one Sub Stack Condition.

Fall Back Card is used only when there is no more available card to draw (no card meets the current conditions). This should not happen and is mainly here only for preventing errors.

Spawned Card is the card, that is already in the scene at the beginning of the game. In the demo scene it is the MainMenuCard, located in GameCanvas → Cards

On Card Swipe lets you trigger an Event every time a card is swiped out of the screen.



Swipe Script (in Scripts)

This script handles all swipe gestures.

Swiped Detection Limit_UD lets you set the value of how long (distance) a swipe has to be, to be recognized as a swipe for up or down swipes.

Swipe Detection Limit_LR is the same just for left and right swipes.

There are several Events for each Swipe directions, also for Swipe Preview, when Player has not fully moved card in one direction.

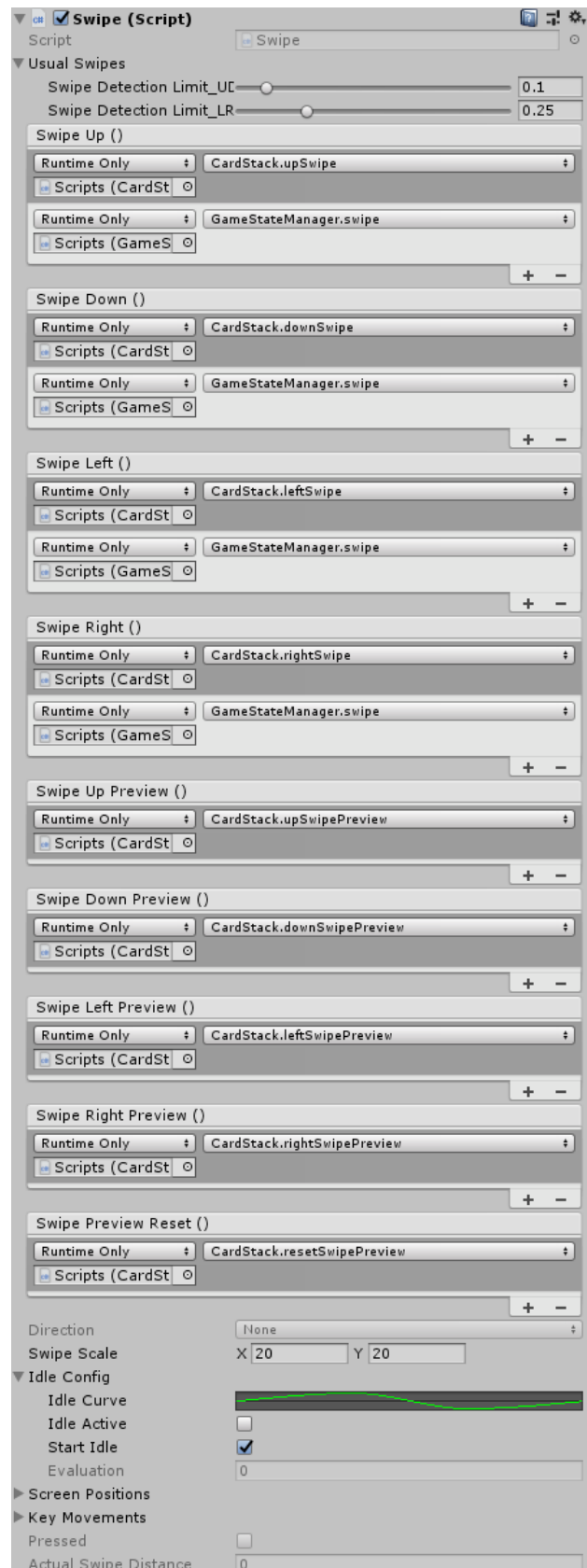
Swipe Scale lets you scale the factor of your swipes for the Animator BlendTree to swipe the cards to the left or right.

Idle Config

Gives you the option to idle swipe a card. It will automatically swipe the card left and right until a user interaction happens. The **Idle Curve** allows to adjust the movement of the card with an animation curve. It can be setup to always idle swipe the first card by checking the **Start Idle** box or triggered manually by calling the event **Swipe.SetIdleSwipeActive**.

Key Movements

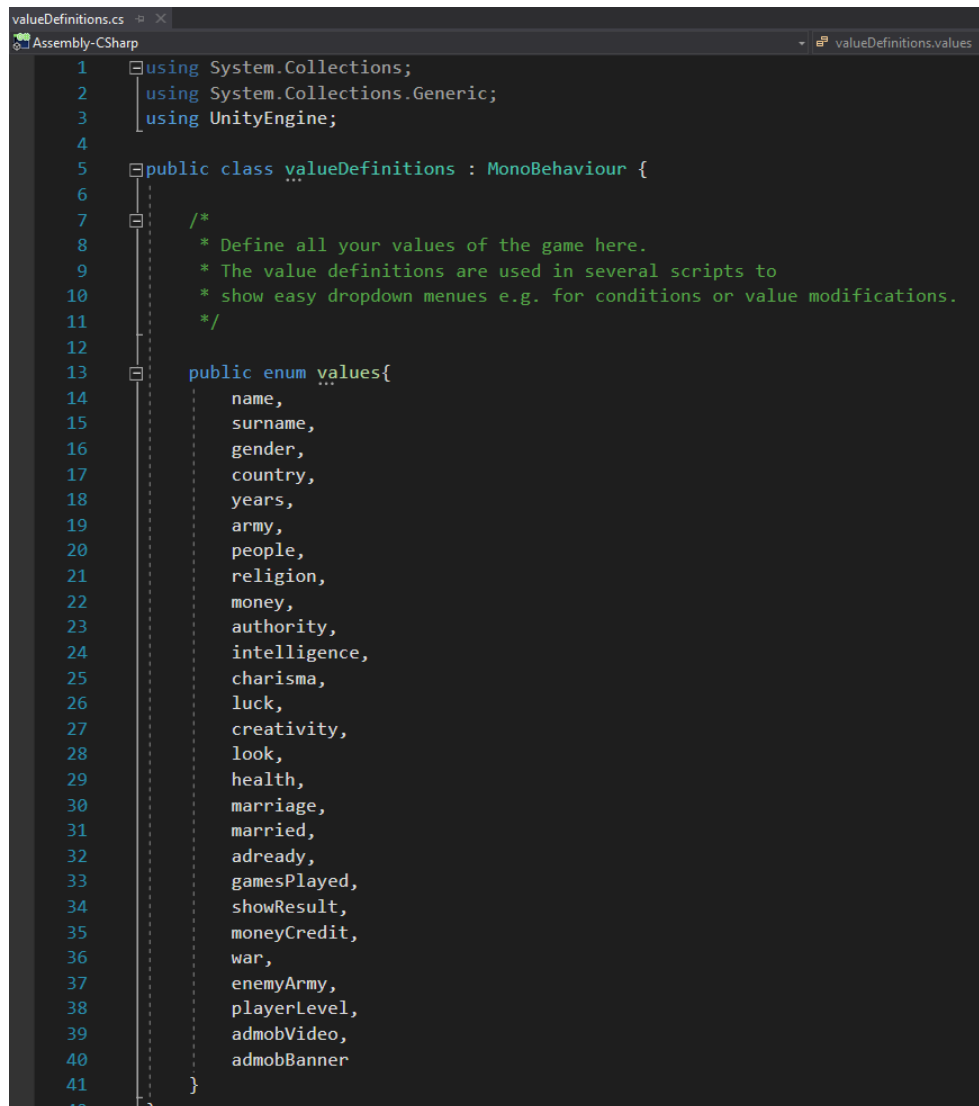
This allows to control the card movement with keyboard instead of mouse/touch swipe input.



Value Definitions (in Project folder scripts)

This Script is not in the Hierarchy window of the game project.

You have to open it from the Project folder **Kings\scripts\valueDefinitions.cs**



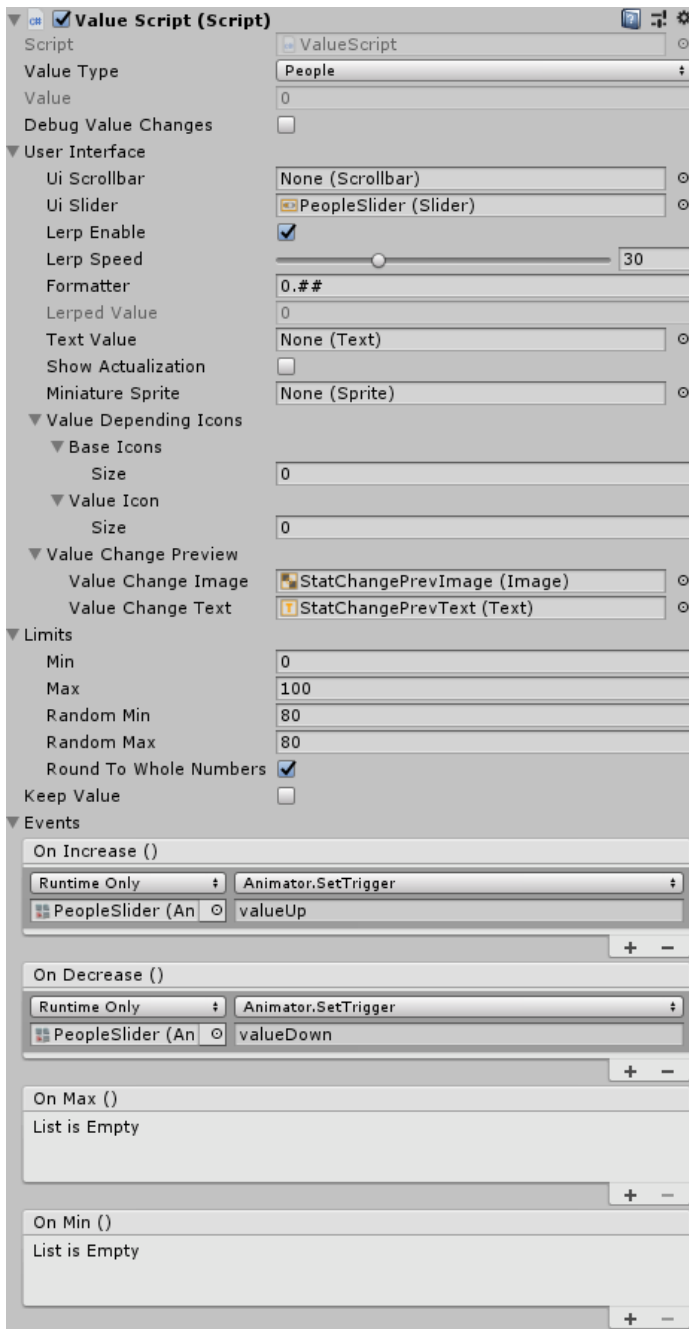
```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class valueDefinitions : MonoBehaviour {
6
7     /*
8      * Define all your values of the game here.
9      * The value definitions are used in several scripts to
10     * show easy dropdown menus e.g. for conditions or value modifications.
11     */
12
13     public enum values{
14         name,
15         surname,
16         gender,
17         country,
18         years,
19         army,
20         people,
21         religion,
22         money,
23         authority,
24         intelligence,
25         charisma,
26         luck,
27         creativity,
28         look,
29         health,
30         marriage,
31         married,
32         adready,
33         gamesPlayed,
34         showResult,
35         moneyCredit,
36         war,
37         enemyArmy,
38         playerLevel,
39         admobVideo,
40         admobBanner
41     }
42 }
```

Every Value needs to be listed in this script, you need to open this script and add/edit the values here. These values will be used in many other scripts like EventScript, where you can simply select them from a drop-down menu.

PLEASE NOTE: If you remove a value from this list (remove line) it will reorder all elements and mess up all existing cards, since these values are addressed by their line number.

When you start your game project the first thing for you to do should be creating your list with all values needed. Adding additional values at the bottom or editing/renaming existing values can be done without a problem.

Value Script (in Values)



For every value you will also need a Value Script. In the example Game scene, they are attached each on an empty GameObject under **Values** in the Hierarchy window.

In **Value Type**, you simply choose from the drop-down menu for which value this Value Script is.

If you want to display this value with a slider you can link it in **UI Slider**, if you want to display it as a text value, you can link it in **Text Value**, of course you can also do both or none if you don't want the value to be shown.

If you enable the **Show Actualization Toggle** a popup window (UpdatedStatsPanel) will appear every time this value changes. This is used if you want to show the player that a value of a secondary stat (StatsPanel, when you swipe up) has changed. You should also then link in a **Miniature Sprite** which will be displayed then.

Value Change Preview lets you add an image and or text to display if and how the value will change depending on the user decision.

Limits let you set the **Min** and **Max** range for the value, in this example from 0-100, this means this value cannot get lower than 0 or higher than 100.

The **Random Min Max** values are the values which are generated on every new game start, in this example the range is from 80-80, this means the value can only be 80. If you choose for example 10-50, it will generate a random value between 10 and 50.

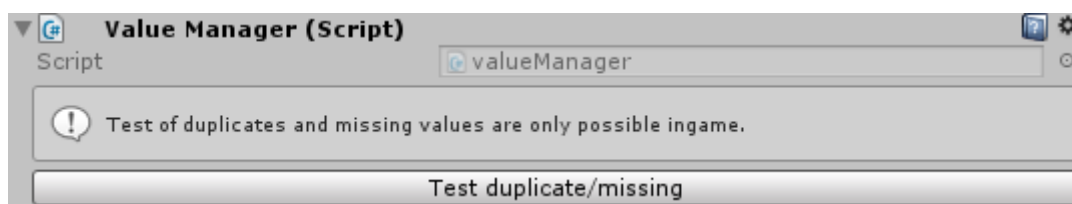
New in Update 1.50: Check if you want to create whole numbers, e.g. 25 instead of 24,987.

The Events for **On Increase, On Decrease, On Min, On Max**, can be used for example if you want to animate the stats display.

Keep Value allows persistent values which are not reset on a new game, this can be useful if you want to keep the years value always growing.

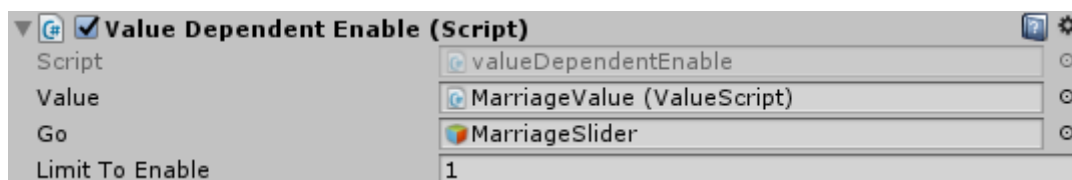
Value Depending Icons gives you the option to change the icon depending on it's value. In the example scene this has been done for the “People” value.

Value Manager (in Values)



When in Play Mode you can run a **Test for duplicate/missing values**. For example, if you set up a value in the Value Definitions Script but did not create a GameObject with the Value Script for it, or if you created two GameObjects for the same value.

Value Dependent Enable Script (in Values)



With this script you can display a specific GameObject (Slider, Text Value, etc.) only when the Value Limit is reached. In this example the Marriage Slider will only be displayed when the Marriage Value is above 1.

Achievements (in Scripts)

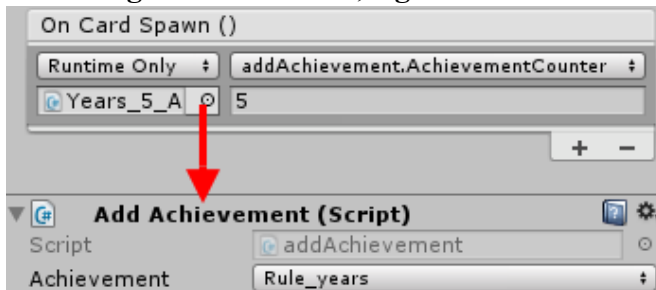
This is a simple Achievement Manager to popup Achievements on special Events. For every Achievement you have you need to set up manually the list in the Achievement Script, similar like the Value Definitions.

```
public enum achievementTyp{  
    marry,  
    rule_years  
}
```

To display the unlocked Achievements in the Achievements Screen you have to create a GameObject (with title, description, image etc.) and link it in **Achievement GameObject**. Two example Achievements are created you can find them in:
MenuCanvas → Panels → AchievementsPanel → Scroll View → Content

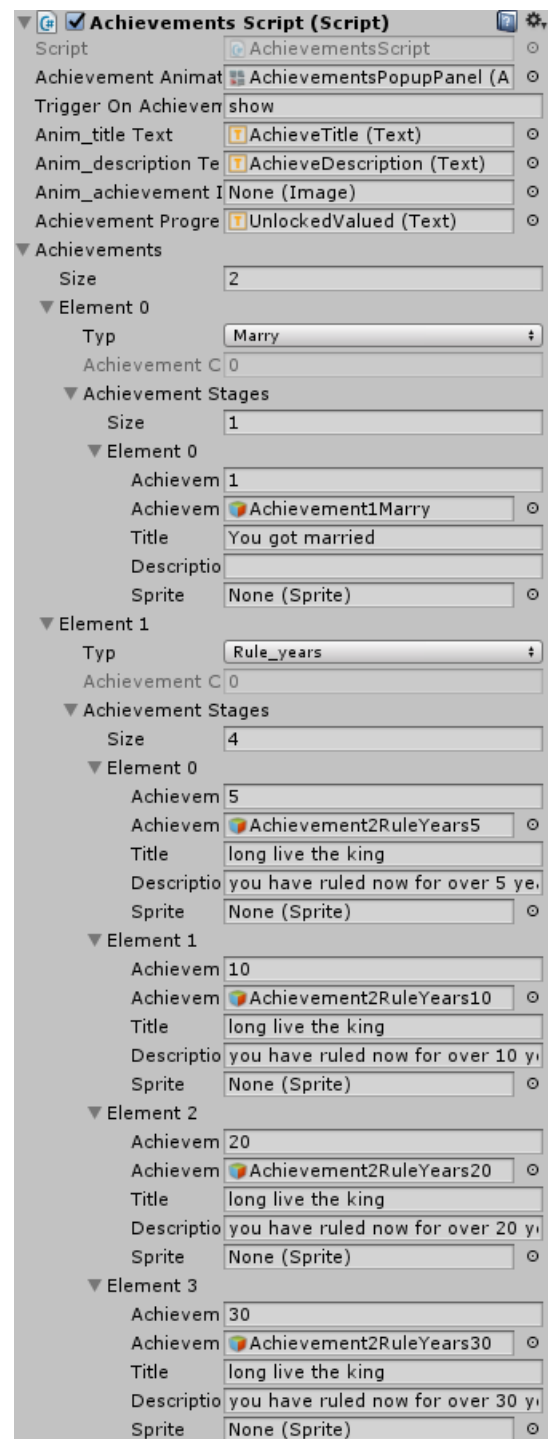
To trigger an Achievement simply add the **Add Achievement Script** to the card which should trigger it, select the appropriate Achievement from the drop-down menu and link it in the On Card Spawn Event of the EventScript and call the function: **addAchievement.add_Achievement**.

“Growing” Achievements, e.g. rule for 5/10/20/30 years.



To use this option, you simple have to call the function **addAchievement.AchievementCounter** instead of **addAchievement.add_Achievement** and enter the appropriate value you have set up for the achievement in the achievement script (right screenshot)

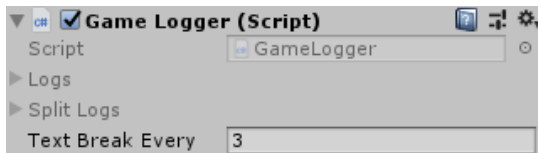
In the achievement script you also have to set up the number of „stages“ the achievement has, in this example it is 4 (5/10/20/30 years). For none growing achievements (like the marriage achievement) you simply have to enter 1.



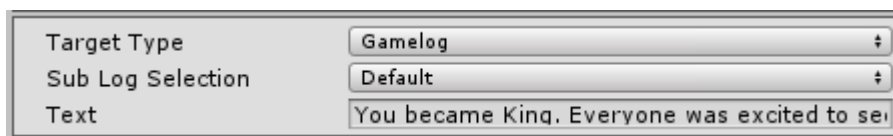
Game Logger (in GameLogger)

The Game Logger consists of two parts, one is the Game Logger itself, the other one is the Value Dependent Game Logs. It allows you to add a text log at the end of the game, which summarizes highlights and actions of the players actions/decisions.

You can choose after how many text blocks a new paragraph should be created with **Text Break Every** for a cleaner look.



Gamelogs can be added from the EventScript in the Extras category:



Sub Logs: allows to split the Game Log, if you want to display them in different categories. Simply select a Sub Log in the “Sub Log Selection” and in the “ShowGameLog” script.

Value Dependent Game Logs

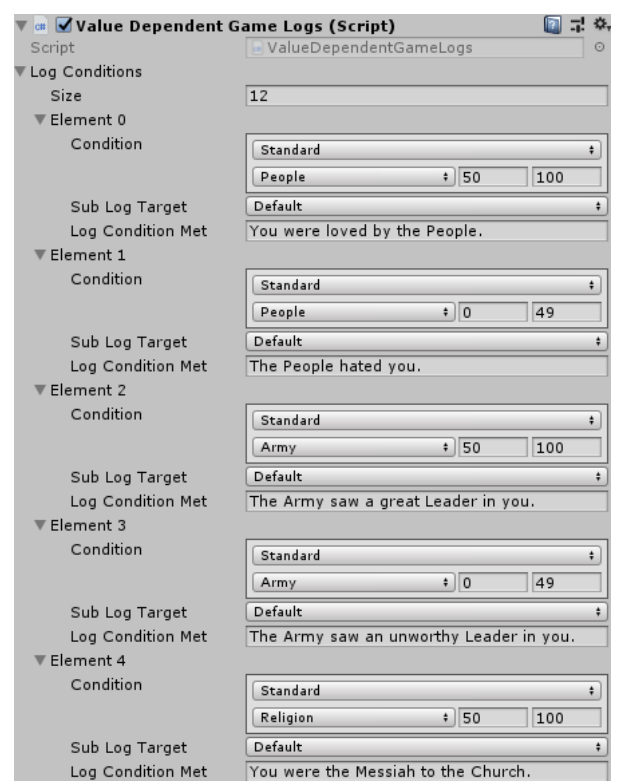
(in GameLogger)

The Value Dependent Game Logs are the other part of the Game Logs. They will be added after the normal Game Logs and are value dependent.

Which means you can select any value of your game and set a range, if the range is met at game over the appropriate text will be added.

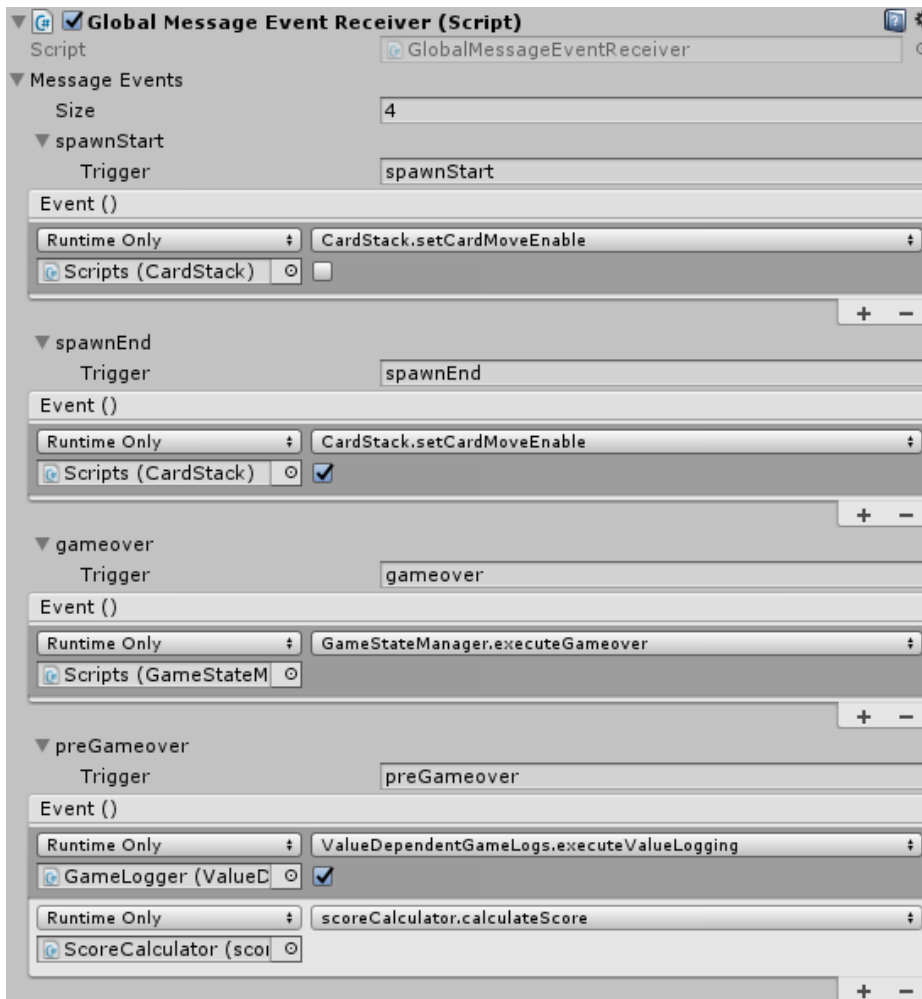
In the example on the right side, there is an element for People 50-100 and another one for People 0-49. This means if the value of People is on Game Over above 50 the Game Logger will display the text: “You were loved by the People”. If it is below 50 it will display the text “The People hated you”.

To Display the Game Log there is a ShowGameLog Script on the GameOver__Log Card.

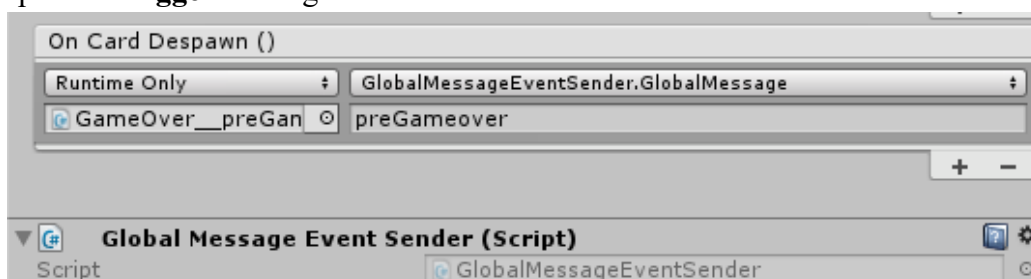


Global Message Event System

The Global Message Event System consists of the 3 scripts: **GlobalMessageEventSender**, **GlobalMessageEventReceiver** and **GlobalMessageEventManager**. It is used to send Unity Events to spawned Prefabs without direct linking.

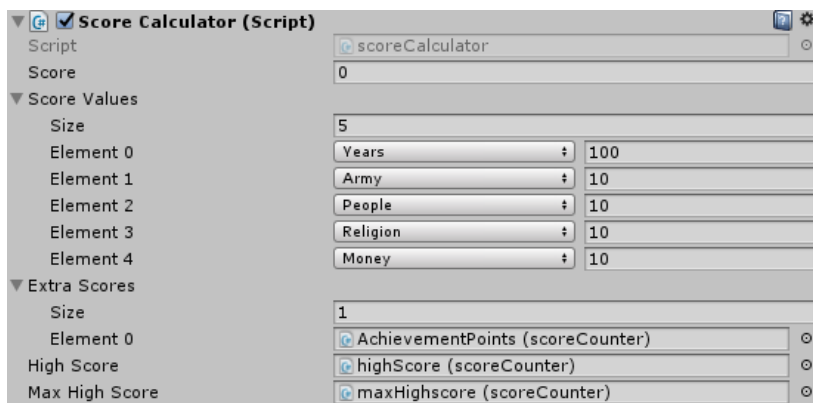


The **GlobalMessageEventReceiver** handles all send messages from every **GlobalMessageEventSender**. To send a GlobalMessage from a prefab (in our case a card) you simply need to add the **GlobalMessageEventSender** script to the card, link it in the **On Card Despawn Event** call the function: **GlobalMessageEventSender.GlobalMessage** and enter the specific **Trigger** message.



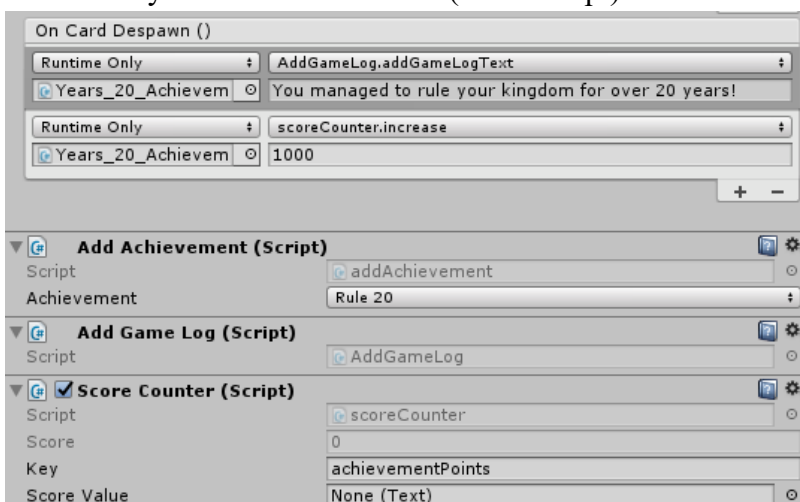
In this example the script is attached to the **GameOver__preGameover** card, when this card despawns the message **preGameover** is sent. The **GlobalMessageEventReceiver** will now run all events for **preGameover**, which is in this case **executeValueLogging** for the **GameLogger** and **calculateScore** for the **ScoreCalculator**.

Score Calculator



The Score Calculator calculates the player score at Game Over which is displayed on the GameOver__Score card. In the game example 5 values are used to generate the Highscore: Years, Army, People, Religion and Money. The number behind each value type is the score multiplier, this means the Years value has a multiplier of 100, e.g. years value is 12 then the score would be 1200 for Years.

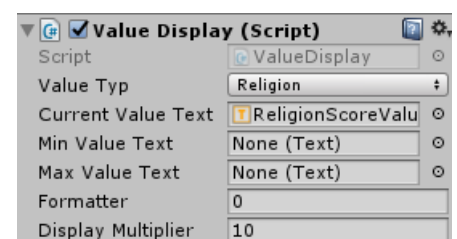
There is also the option to add **Extra Score** if you want to add points to the Highscore which are not value dependent. In this example they are Achievement Points. The achievement points need to be manually added from the cards (EventScript).



This example is from the **Years_20_Achievement** card. The Score Counter script has the key achievementPoints and is linked with the On Card Despawn event, which runs the function: **scoreCounter.increase 1000**, this adds 1000 points to the achievement points.

To display the achievement points or any other score, simply add a score counter script to the object you want it to display (in our example the GameOver__Score card) and link in the text field in Score Value and enter the appropriate key.

To display one of the base values you need to add the script **Value Display**. You can select the value you want to display from the drop-down menu, link in the text field where you want it to display. You can choose between the current value, and min/max value. Current value is for example used on the GameOver__Score card. The min and max values are displayed on the Highscore panel in the game menu, e.g. shortest reign / longest reign.



Kings Level Up (in Score Calculator)

The Kings Level Up script allows based on the Score Calculator to gain experience, which allows the player to level up and unlock new cards or bonuses.

Max Level: value of maximal player level

Xp Costs: set the costs for each level, in this example

Xp Bar: you can link in a slider to display the experience

Xp Bar Fill Speed: adjust the fill rate of the xp bar

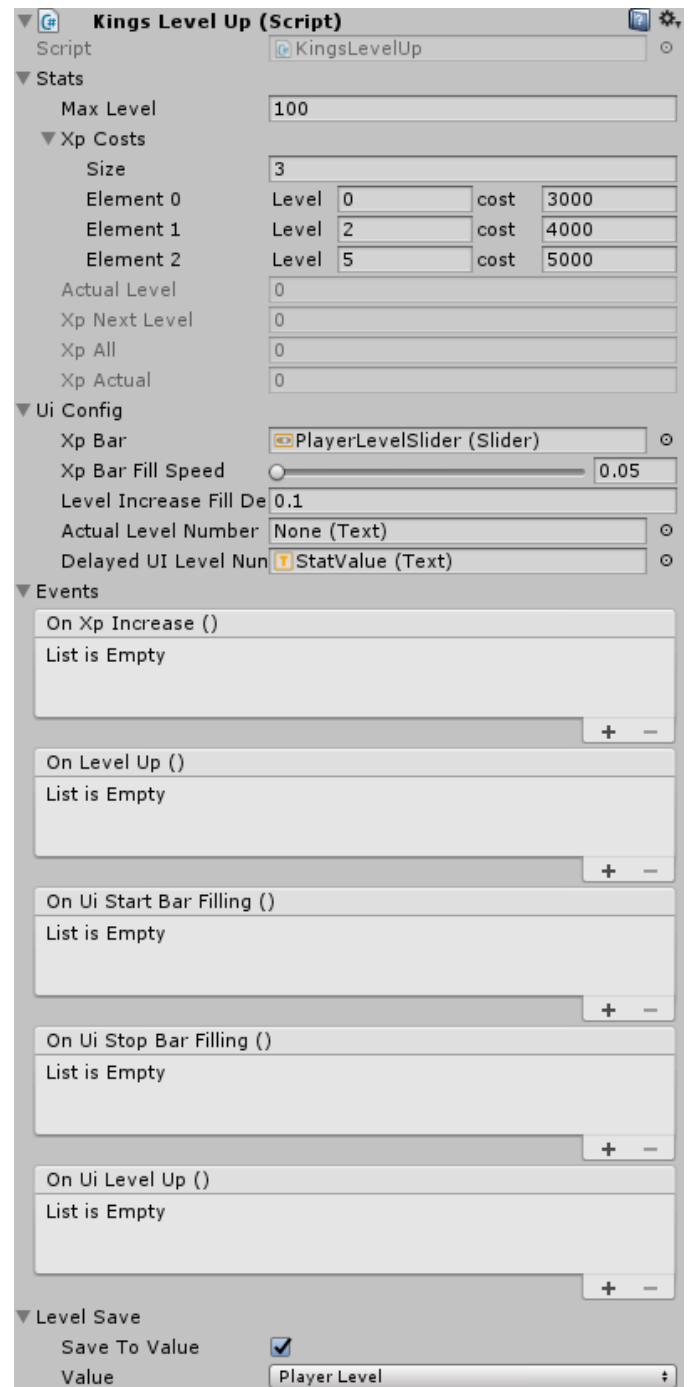
Level Increase Fill Delay: adjust the delay until the xp bar starts to fill again on level up

Actual Level Number: text field to display level value

Delayed Ui Level Number: text field to display level value, with delay matching the fill rate of the xp bar

Events: several triggers for Unity Events are available

Level Save: make sure the “Save To Value” box is checked and that the correct Value for the Player Level is selected



Country Name Generator (in Values)

Script: CountryNameGenerator

Gender: Male

Countries

Size: 3

England

List Entry: England

Name Comb

Size: 6

George	Male
Harry	Male
Charles	Male
Elizabeth	Female
Kate	Female
Diana	Female

Surname

Size: 6

Element 0: the Ruler

Element 1: the Emperor

Element 2: the Mighty

Element 3: the Strong

Element 4: I

Element 5: IV

France

FantasiaLand

Country Text: PlayerCountryText (Text)

Name Text: PlayerNameText (Text)

Country And Name Text: None (Text)

Country And Name Text Format: {0}, {1}

Vs_type_country: Country

Vs_type_given Name: Name

Vs_type_surname: Surname

Vs_type_gender: Gender

The Country Name Generator script is used to generate a combination of Name, Surname, Gender and Country. This should be pretty self-explanatory, but there is one thing to note:

Script: ValueScript

Value Type: Name

Value: 0

Debug Value Changes: ☐

User Interface

Limits

Min: 0

Max: 5

Random Min: 0

Random Max: 5

Events

You should always use the same number of Names and Surnames for each country.

The reason for this is you have to enter in the value script of Name and Surname the Limits/Random min-max, in this example it is from 0-5, this means there are a total of 6 names per country (since it starts counting from 0). If you would now have a country with 8 names, the last two names would never be drawn. If you would set the range number for example from 0-10 and you only have 6 names it would draw the last name more often than the others because for every rolled value above 6 it would draw the last name in the list. But you can of course use 10 values for Name and only 5 for surname, because these values are separated from each other.

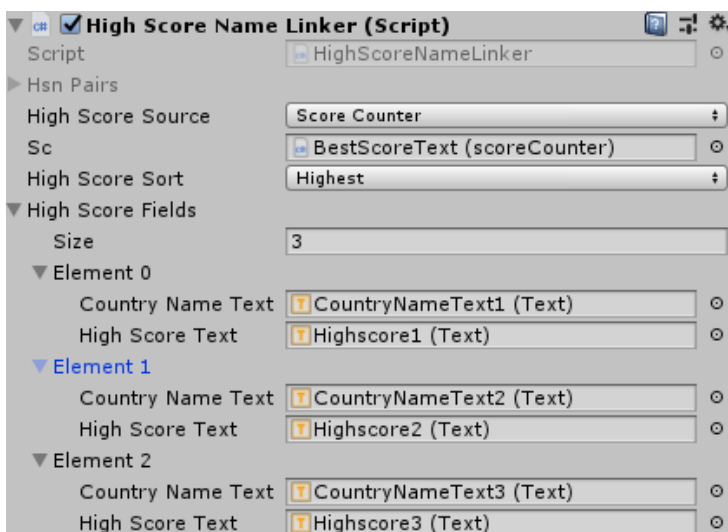
High Scores (in MenuCanvas → HighscorePanel)

The High Score Name Linker script allows you to display several high scores.

High Score Source you can choose between **Score Counter** and **Value Script**. Score Counter will display the values generated by the Score Calculator. When selected Value Script you can display any of your values (year, people, army, etc.).

High Score Sort allows you to choose between Highest, Lowest and Last on Top, which will display the score of the last game.

High Score Fields lets you set the number of entries you want to display.



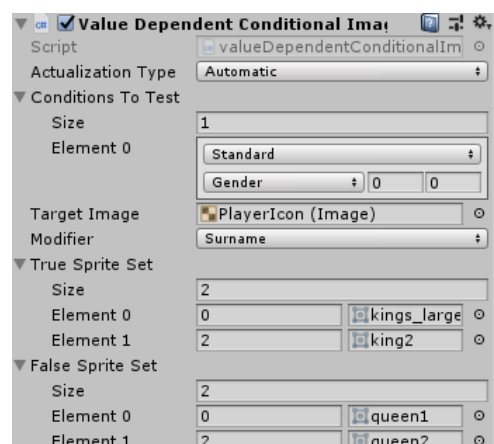
Value Dependent Conditional Image (King Image)

This script is used to display different King / Queen images, depending on the Player selection.

This script checks the “Condition To Test” Gender value:

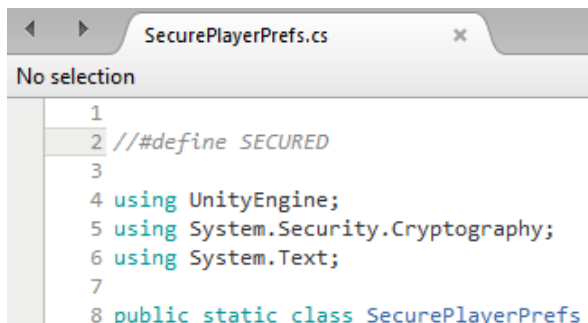
0 = male, 1= female

If 0 then the “True Sprite Set” is selected, it will then determine the icon by the “Modifier” value Surname, otherwise the “False Sprite Set” is selected and the player icon will be chosen from there.



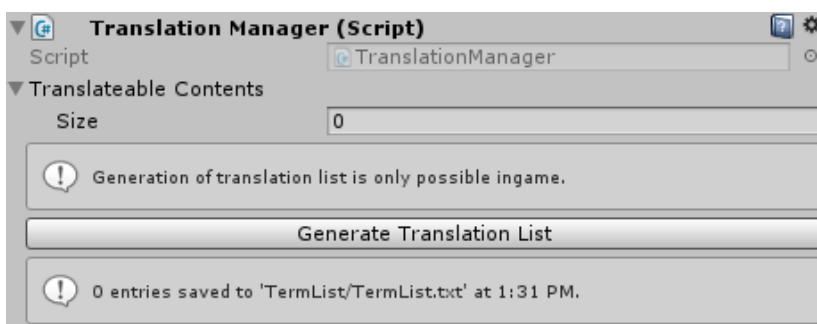
SecurePlayerPrefs (in scripts Project folder)

To prevent manipulation of the score or other values encryption is used. If you want to disable it you can simply open the **SecurePlayerPrefs** script and disable the second line, that it looks like this:



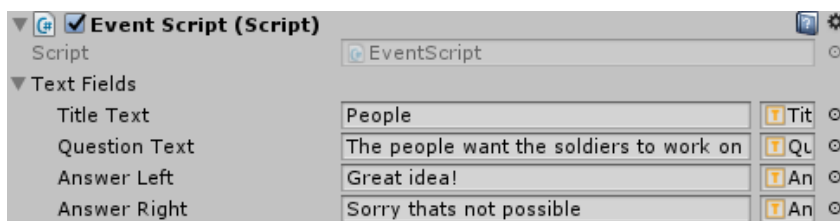
```
1
2 //define SECURED
3
4 using UnityEngine;
5 using System.Security.Cryptography;
6 using System.Text;
7
8 public static class SecurePlayerPrefs
```

Translation Manager (in Scripts)



We have integrated support for the Localization Asset: **I2 Localization**

This means the text you enter in the EventScript of the cards will be used as terms and can be translated with I2 Localization.

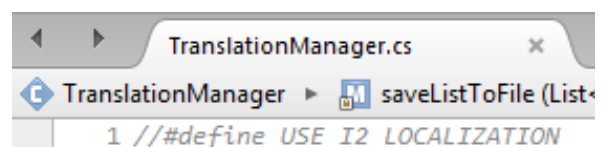


This also works for the Achievements, Country names and Game Logger entries.

With **Generate Translation List** a text file (Kings\TermList.txt) with all the text you have entered in these scripts is generated. You need to be in Play Mode for this. You can then open the text file in Excel (or any similar program) and copy it into the I2 Localization spreadsheet. This way you don't have to enter every term manually, but can import them all at once.

Please note: This does not work for the Game Logs added with the **Add Game Log** script. These have to be added manually to the I2 Localization terms.

To enable translation you need to import the I2 Localization Asset and enable the first line of code in the Translation Manager script.



```
1 //define USE_I2_LOCALIZATION
```

Unity Ads

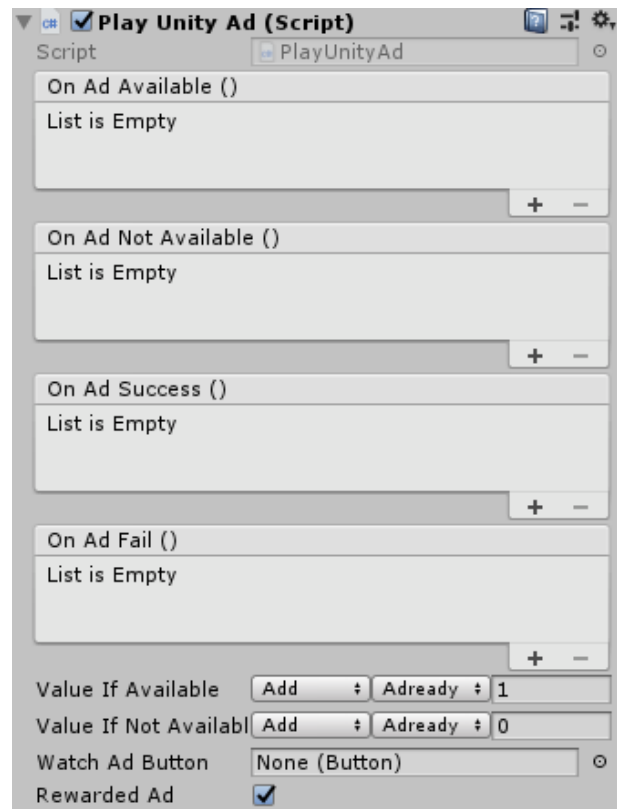
You can start playing Unity Ads simply by calling the function **showAd** of the Play Unity Ad script.

This script gives you also Events for the following cases: Ad available, Ad not available, Ad success and Ad fail.

The value **Adready**. It is set automatically to 1 if an ad is available and to 0 if ads are not available. This allows the Card Stack to only draw Ad-Cards if an Ad is available.

If you uncheck **Rewarded Ad**, Ads can be canceled/skipped by the user.

In our example game two ad card have been created: **Ad_General** (appears one time randomly per game) and **Ad_People** (appears once when People value is 40 or below).



To enable Unity Ads go to the Services Tab (Ctrl+0) and turn Ads ON.

For testing purposes please make sure “Enable test mode” is checked.

Also make sure you have selected Android or iOS as Platform.

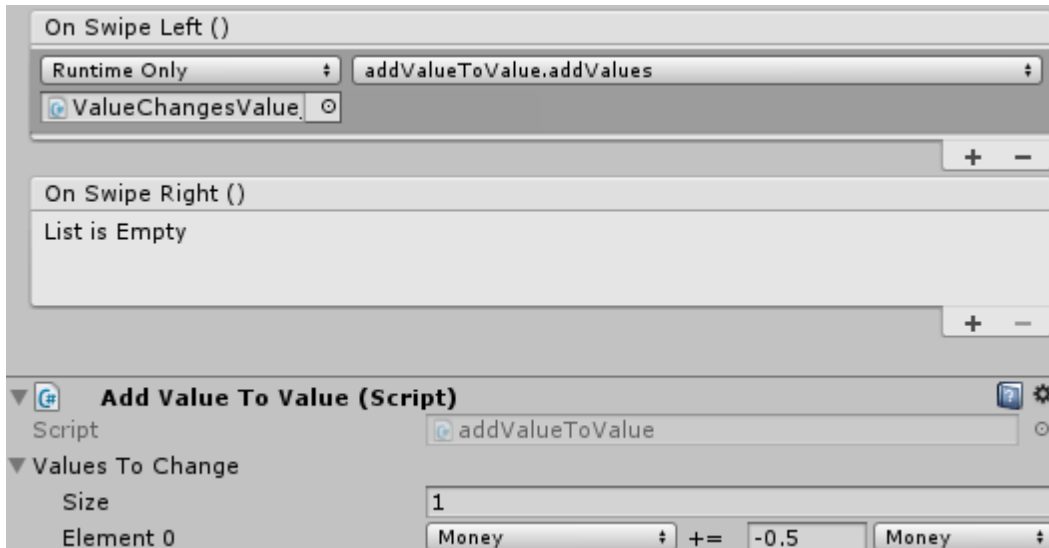
For further information about Unity Ads go here:

<https://docs.unity3d.com/Manual/UnityAds.html>

Add Value to Value

You can directly add/subtract values from each other, with the **Add Value to Value** script.

You can see an example in the **ValueChangesValue** card prefab. In this example, half of the Money value will be deducted from the Money value (on swipe left)

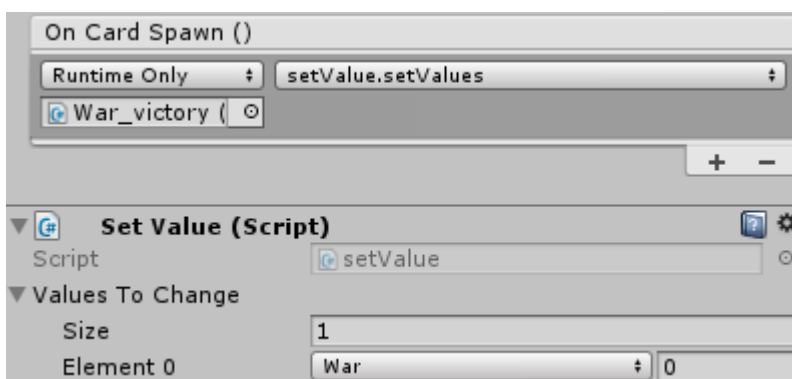


Set Value

The **Set Value** script is obsolete since Update 1.44,
please use now the new [Change Value](#) or [Make Changes](#) script.

The SetValue script allows to set a value directly to a specific number, with the function **setValue.SetValues**

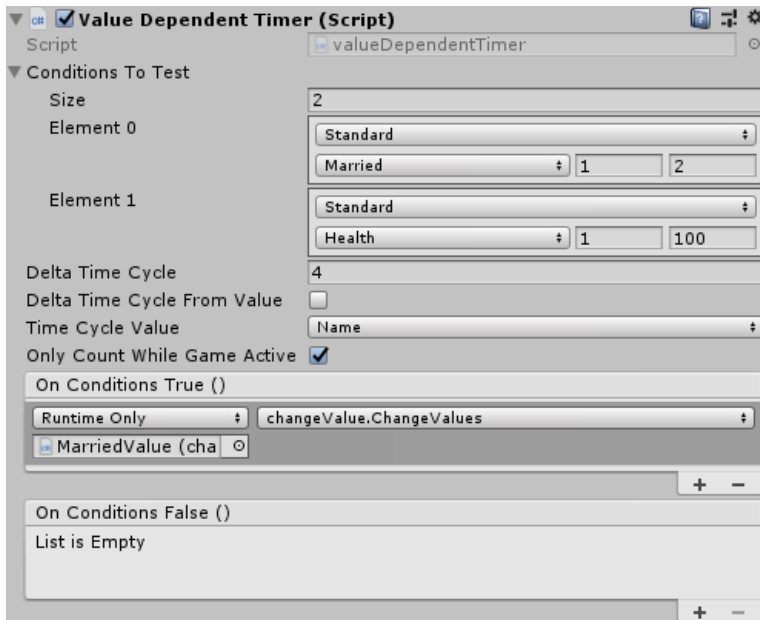
Example: War_Victory sample card, the On Card Spawn event calls the function setValue.SetValues of the Set Value script, which sets the Value of War to 0.



Change Value over time

With the **Value Dependent Timer** script, you can change values over time.

In this example the player will lose 1 health every 4 seconds, as long as he is married and his health is between 1 and 100.



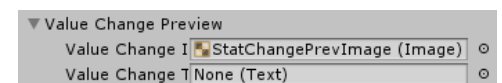
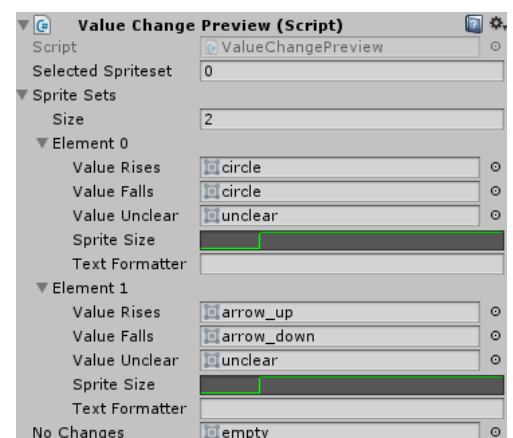
Value change preview

The **Value Change Preview** script (in Values) lets you display an indicator which values will change depending on the player decision.

You have the option to change between Sprite Sets (with the function: **ValueChangePreview.setSpriteSet**) this allows you to e.g. temporarily show the player if the values rises or falls and if you want to you can even display the exact numbers.

In the example scene this is done in Global Messages “show Results” with the **Witch_ShowResults** card.

You have to link in a Preview Image in the Value Script for each Value where you want to show a preview.



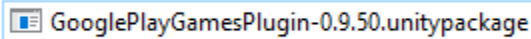
Google Play /iOS Leaderboard and Achievements

1. Download the official Google Play Game Plugin:

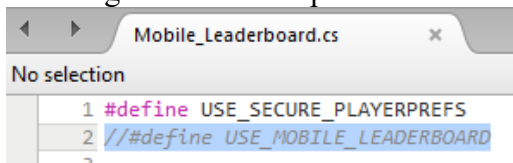
<https://github.com/playgameservices/play-games-plugin-for-unity>

(This asset has been tested with version 0.9.64, if you have any issues, you should try this version first)

2. Install the plugin by double clicking the assetpackage in the “current” folder of the plugin



3. Enable the Mobile_Leaderboard by activating (remove “//”) the following line in the script:



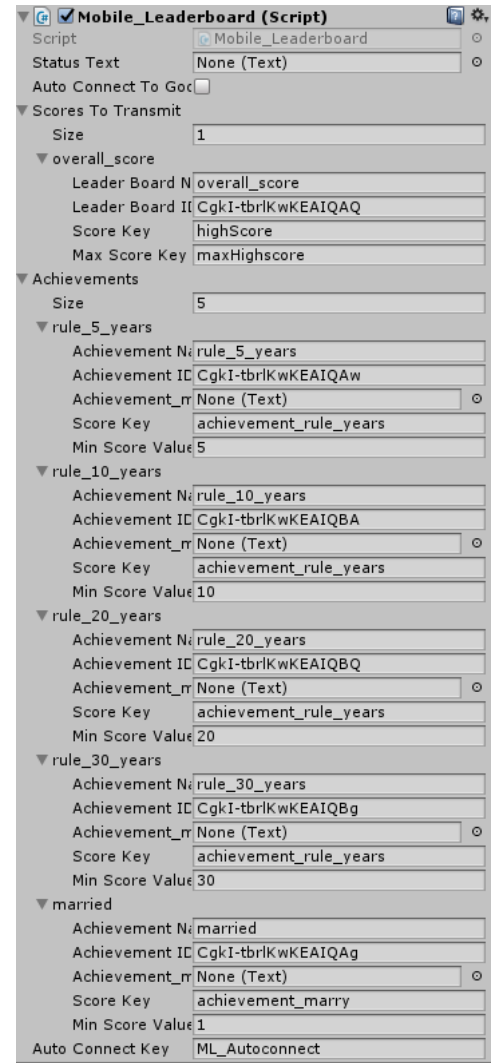
This has to be disabled by default otherwise you would get error messages when the Google Play Games Plugin is not installed.

4. Setup your Achievements and Leaderboard in the Google Play Developer Console according to the instructions:

<https://github.com/playgameservices/play-games-plugin-for-unity>

5. Import the “Resources” into Unity and click on “Setup”.

6. Insert these values into the Mobile Leaderboard like in the two screenshots on the left side and you are ready to go.



Overview of the most important functions:

Open Leaderboard (and transmit score):

Mobile_Leaderboard.UI_call_transmitScoreAndLeaderboard

Open (and transmit) Achievements:

Mobile_Leaderboard.UI_call_computeAchievements

Transmit score (Leaderboard):

Mobile_Leaderboard.std_call_transmitScore

Transmit achievement:

Mobile_Leaderboard.std_call_computeAchievements

With “Auto Connect To Google” you can choose if you want to directly connect on the start of the game or do it manually.

For the Leaderboard the ScoreValue and ScoreKey is generated by the Score Generator. If you want more scores for your Leaderboard you simply need to create more instances of the Score Generator and set it up for the appropriate values.

For the Achievements data is generated by the Achievement script (Scripts). The key is automatically generated by the name you enter, e.g. Achievement name: “marry” means you get the following key: “achievement_marry”

Please note: For iOS a separate plugin is required:

<https://assetstore.unity.com/packages/tools/integration/ios-game-center-plugin-for-unity-14839>

This is an inexpensive plugin that works well. Of course, you can use other plugins, but then you have to make changes on the Mobile Leaderboard script.

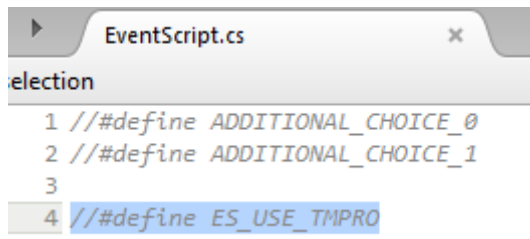
Tip: If you plan on using the Leaderboard for Google Play AND iOS Gamecenter I strongly recommend to setup Google Play first, since Google automatically generates the Achievement and Leaderboard ID's, where Apple let's you choose them, this way you can enter the automatically generated IDs from Google and use the same for Apple. This allows you to have the same configuration for both platforms.



Text Mesh Pro Support

You can use Text Mesh Pro instead of the Standard Unity UI Text.

1. Import the Text Mesh Pro Asset
2. Enable (remove “//”) the following line in the Event Script:

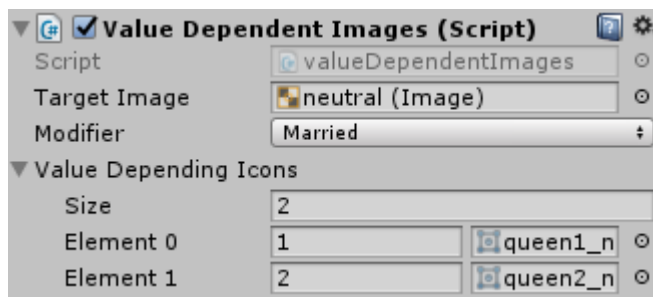


3. You need to replace the UI Text Elements on all cards with the Text Mesh Pro UI Text Elements and link it into the EventScript of each card.

Example card: Kings → cards → _templates → card_template_textmeshpro

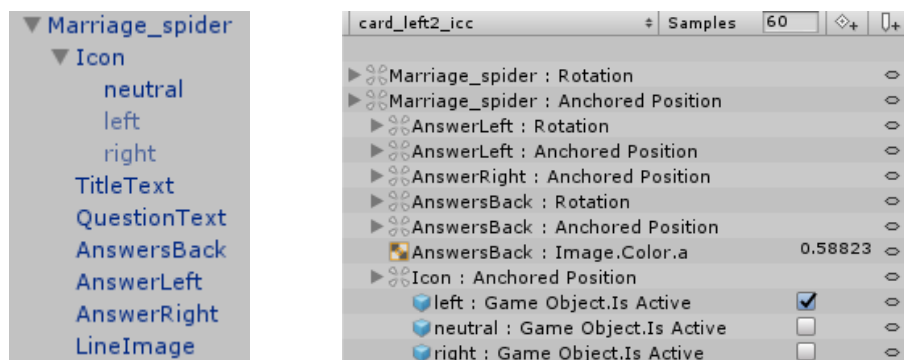
Value Dependent Images

You can now change images depending on a specific value. In the example this has been done with the marriage cards (marry smart/beautiful queen).



When you marry the beautiful one, the “Married” value is set to 1, for the smart one it is set to 2.

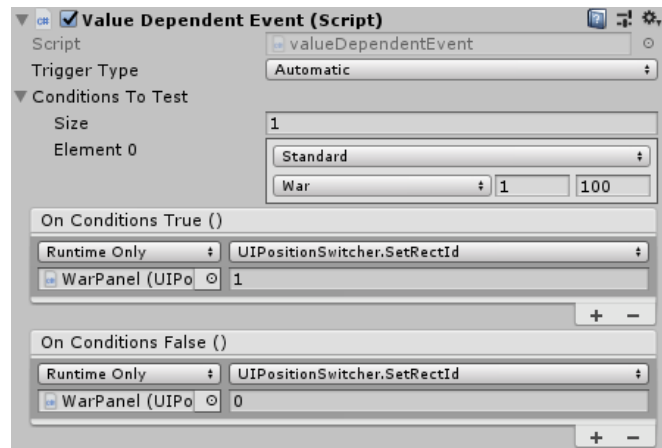
The change of the “mood” images (happy/angry) is simply done with the animator (enable/disable in animation).



Value Dependent Events

With this script, you have the possibility to trigger various Unity Events depending on a specific value.

Trigger Type: Manual or Automatic



Manual: you have to call the function “ExecuteConditionCheck” every time you want to check if the conditions are met

Automatic: script checks automatically on start and every time the value is changed if the condition is met

Conditions to Test: setup the conditions you want to check

On Conditions True: when conditions are met it will trigger these UnityEvents

On Conditions False: when conditions are NOT met it will trigger these UnityEvents

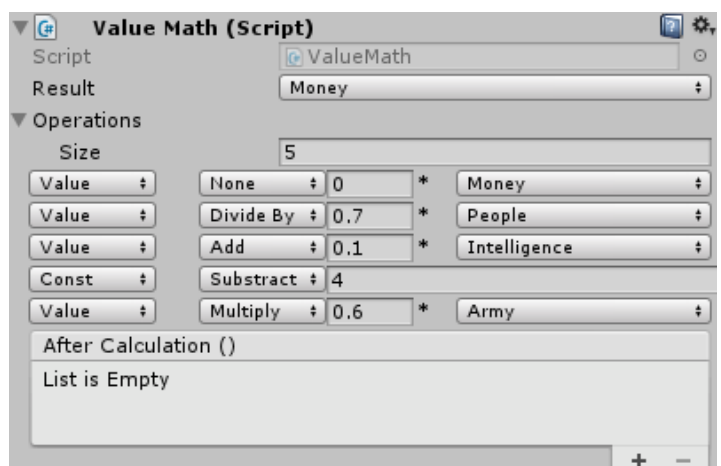
Value Math

The Value Math script allows more advanced value calculations.

The function “**ValueMath.Calculate**” can be accessed from any UnityEvent.

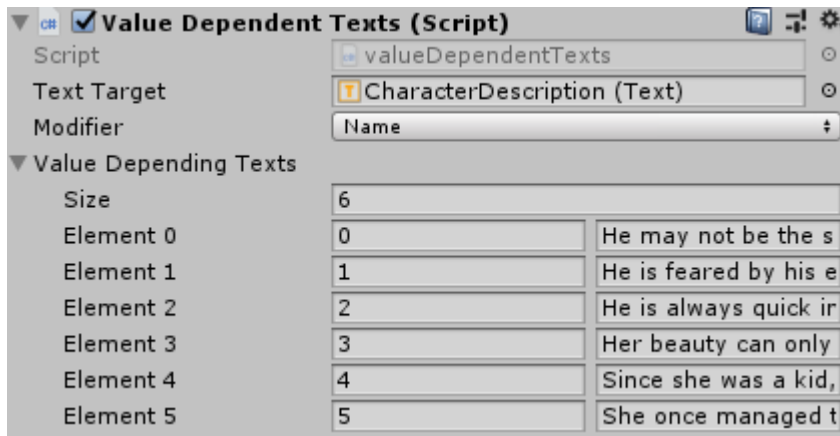
Example: PlayerLevel2_finances sample card

The **After Calculation** Event can be used if an even more advanced value calculation is needed, to access another Value Math script, e.g. $(value1 * value2) + (value3 * value4)$.



Value Dependent Text

Value Depending Text allows you to display different text depending on it's value.

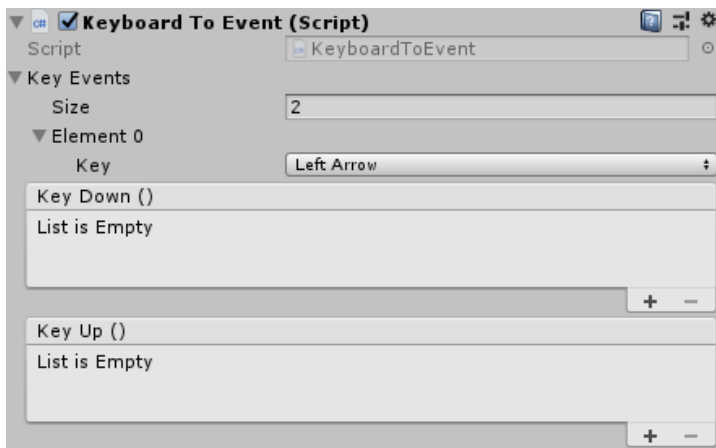


In the example this is used for the CharacterDescription in the PlayerInfoPanel, to display different description text depending on which King/Queen is selected (based on the Name).

Example: MenuCanvas → PlayerInfoPanel → CharacterDescription

Keyboard to Event (in Scripts)

This script allows you to trigger specific events when a key is pressed.



Kings Cards Import/Export (ImEx)

With „Kings ImEx“ it is now possible to export your cards to Excel, edit the text or add new cards and import them again. To open the menu go to the Unity Panel -> Window -> KingsImEx.

GroupName	CardName	StyleName	EventScript titleText	EventScript questionText	EventScript answerLeft	EventScript answerRight	EventScript answerUp	EventScript answerDown	Info
General	_StartCard	cs_King	You are now K	swipe left or ri	i have waited	i will try my be			Style OK.
General	Years_30_Ach	cs_King	30 Years	Congratulation	Great!	Great!			Style OK.
General	Years_20_Ach	cs_King	20 Years	Congratulation	Great!	Great!			Style OK.
General	Army_parade	cs_Army	Army	The army wan	Sure	We don't hav			Style OK.
General	Army_Weapor	cs_Army	Army	The Army wan	Sure	Your Weapons			Style OK.
General	Conditional_Ai	cs_King	ConditionalCai	Choose your s	Try it with you	Charisma, i ch			Style OK.
General	Conditional_M	cs_Marriage	Marriage	Which Princes	The Beautiful	The Smart On			Style OK.
General	Conditional_R	cs_King	Random & Luc	Swipe left to r	Random	I am feeling lu			Style OK.
General	Fallback_Gam	cs_None		This should no					Style OK.
General	FollowUp_Marr	cs_Marriage	Proposal declir	The princess is	Oh no...	Oh no...			Style OK.
General	FollowUp_Marr	cs_Marriage	Proposal declir	The princess is	Oh no...	Oh no...			Style OK.

Export

Directory: choose the directory where the cardlist will be exported

File: choose the filename of the cardlist

Card Stack: add the CardStack script here, only the cards from the cardstack will be exported

Export Cards Button: cards will be exported when pressed, you will see an info panel below

Import

Data File: choose the cardlist you want to import

Style Definitions: choose the „CardStyle_List“ for this field

Select Output Folder: choose the output folder where the cards will be created, if you did edit cards, make sure that it is the same folder as the existing cards, this will update the cards instead of creating new ones

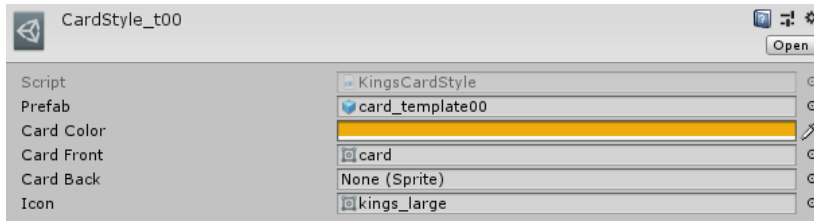
Analyze Import Button: Gives you a preview of the cards that will be imported and also notifies you if there is a problem

Execute Import Button: If there is no problem with the Cardlist, it can be imported by pressing this button

Card Styles

Card Styles lets you easily modify the look of all your cards with just a few clicks. It consists of 3 elements: CardStyle (scriptable object), CardStyle (script) and CardStyle_List.

CardStyle (Scriptable Object)



These let you define the „style“ of the cards. They are located in Kings/cards/_templates.

When you change something on a card style, it will immediately change all cards which have this card style selected.

Prefab: select the card template which will be used when new cards are created with KingsImEx

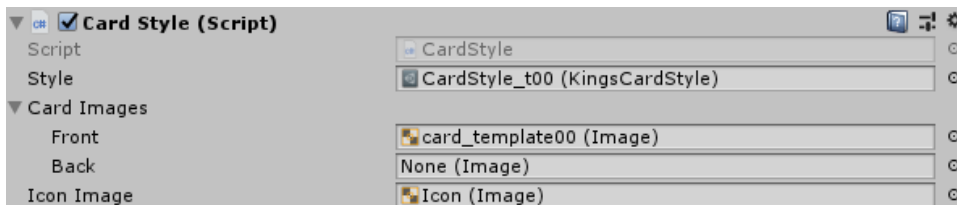
Card Color: choose the color of the card

Card Front: choose the card front image

Card Back: choose the card back image

Icon: choose the Icon of the card

CardStyle (script on every card)



Style: select the style of the card, you can easily swap out styles if you want to change the look

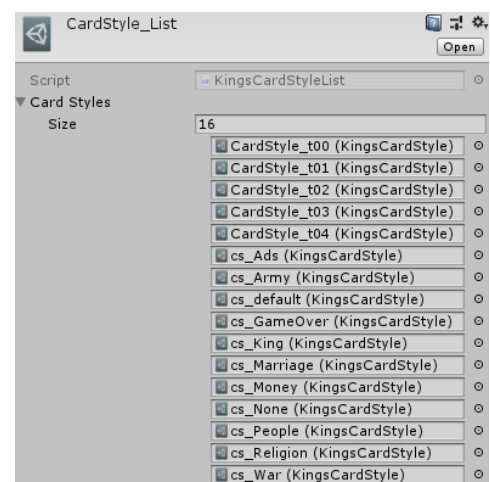
Card Images Front: select the UI Element which display the Image of the front card

Card Images Back: select the UI Element which display the Image of the back card

Icon Image: select the UI Element which display the Image of the card icon

CardStyle List

A list which contains all Card Styles, is required by the KingsImEx script. Is located in Kings/cards/_templates.



First Start Panel

The “First Start Game Object Setter” (in Scripts) will activate a Panel or any other GameObject on the first time you run the game (or after you have reset the game data). Make sure the GameObject is disabled on default.

It can be used for e.g. game instructions, language selection etc.

Typewriter Text Effect

This script allows you to display text with an “Typewriter” effect.

You can find an example for this on the FirstStartPanel or the GameOver_Log card.

It can display a specific text you enter in the “Text” field, or you can send text to it, e.g. by linking it from a Textfield of the EventScript, e.g. QuestionText.



You can cancel the typewriting effect and display the text immediately by calling the function:

Kings.Typewriter.FinishTypewriting

Types per Second: adjust the speed of the writing effect

Start Delay: adjust the delay when the effect will start

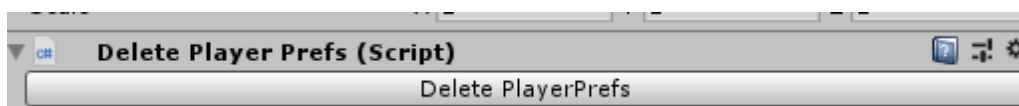
Target Text: link in the target text field where the text will be displayed

On Type Character: triggers an event for each letter typed

On Type Finished: triggers an event when typewriting is finished

Reset Player Prefs (in Scripts)

This script allows you to reset all game data from editor or in game.



In Unity Editor you can click the above button to reset all data.

In game you can call the function “DeletePlayerPrefs.DeleteAll”.

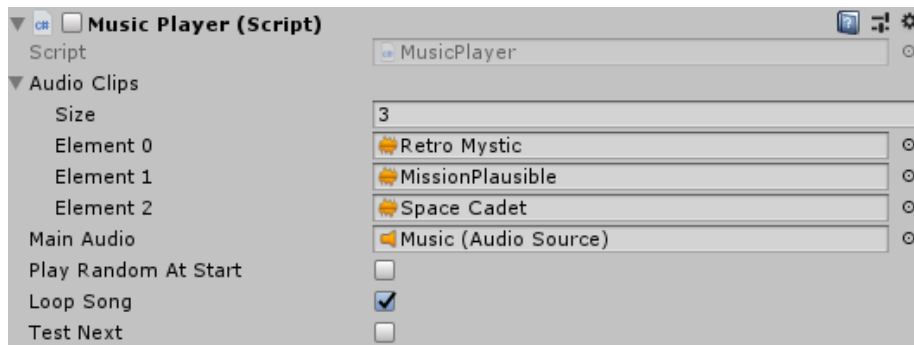
Example: MenuCanvas -> DeleteDataPanel -> ClearButton

Restart Game Button

This allows to end the current game and start a new one.

Example: MenuCanvas -> SettingsPanel -> RestartGameButton

Music Player



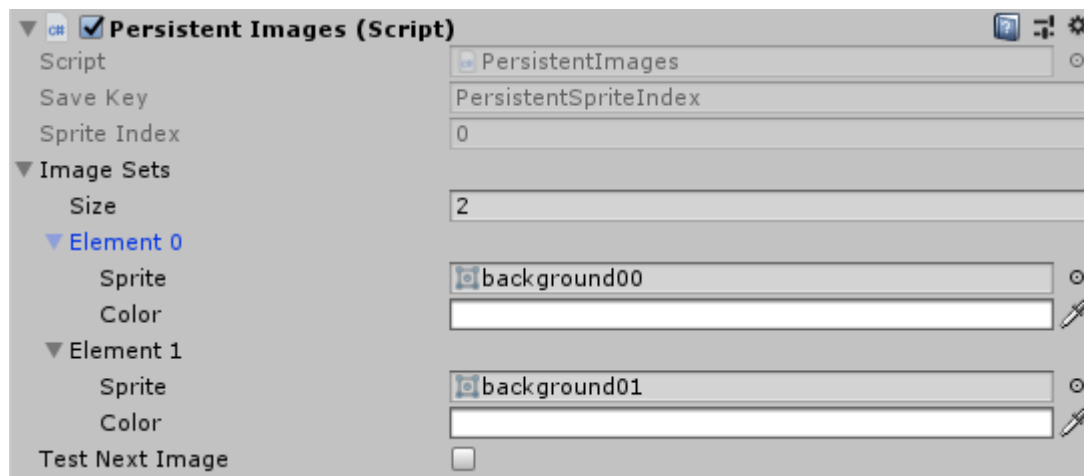
You can now select a specific track by calling the function:

MusicPlayer.setSong

This allows a change of the game theme, example: War

Background changer / Persistent Images

You can now set different images as background during game.



Located in: BackgroundCanvas → Background

To change a background, call the function:

PersistentImages.SetSpriteIndex

This allows a change of the game theme, example: War

Support for Hyper Card asset

With update 1.40 Hyper Card support is now added to the main demo scene. You simply need to import the Hyper Card asset to Kings Project.

A sample card has been created which is located here:

Kings → cards → Misc → SampleCard_Hypercard

Support for Shadero Sprite asset

With update 1.40 Shadero Sprite support is now added to the main demo scene. You simply need to import the Shadero Sprite asset to Kings Project.

A sample card has been created which is located here:

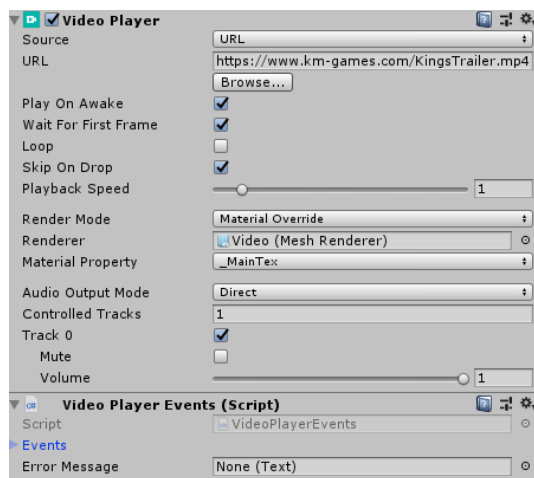
Kings → cards → Misc → SampleCard_Shadero

Unity Video Player

With update 1.43 support for Unity Video Player has been added.

The sample card can be found here: Kings → cards → General → VideoCard

The script “Video Player Events” allows to trigger Unity Events for several events.



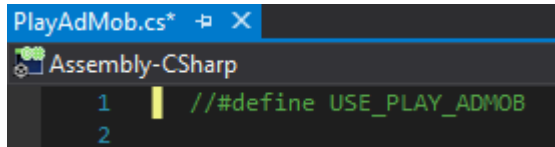
For more detailed information about the Unity Video Player see here:

<https://docs.unity3d.com/ScriptReference/Video.VideoPlayer.html>

Google Admob support

With update 1.43 support for Google Admob is now added.

1. Switch to Android/iOS Platform in Build Settings
2. Download and import the Google Admob plugin:
<https://github.com/googleads/googleads-mobile-plugins/releases/latest>
3. Enable the PlayAdMob script by removing the “//” from the first line:



```
1 // #define USE_PLAY_ADMOB
2
```

4. If you build your project now for Android or iOS it should now display ads

Please note: In Unity Editor Google Admob Ads are not displayed

To create a Admob Account go here: <https://support.google.com/admob/answer/2784575>

To link your App with Admob go here: <https://support.google.com/admob/answer/2773509>

The PlayAdMob script has many options to configure the Ads they way you want.

Ids: Enter your Admob Ad ID

Ad Size: Change the type of the Banner Ad

Ad Position: Change the position of the Banner Ad

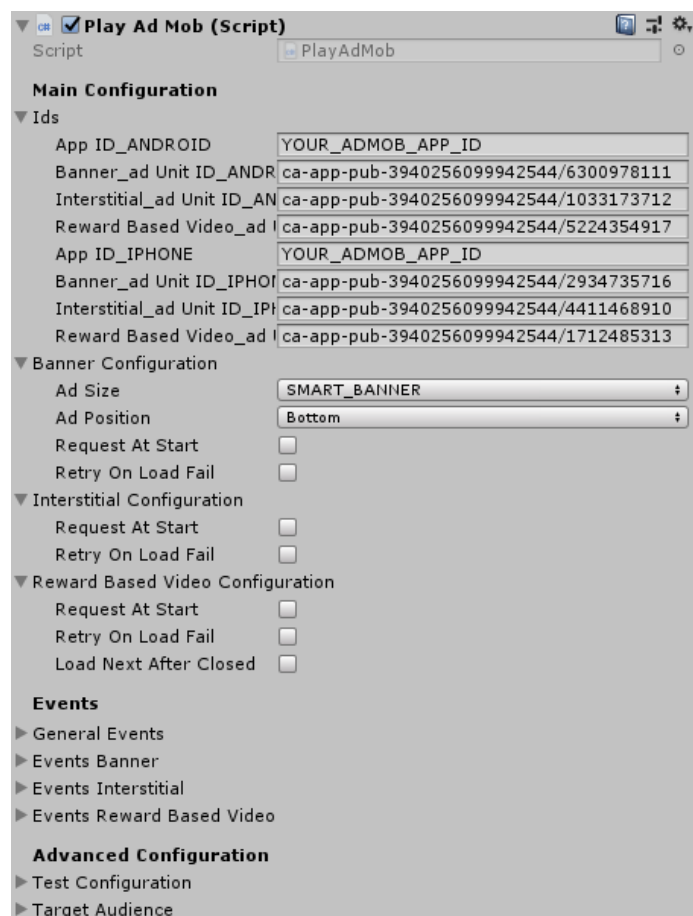
Request at start: Check if you want to load the ad directly on start

Also many Unity Events have been added:
e.g. On Initialization, On Error, On Ad Loaded, On Ad Opened etc.

PLEASE NOTE:

If your game should crash on start when using Admob you will have to add your Admob ID to the Android manifest.

<https://developers.google.com/admob/unity/start>



Unity IAP support

With update 1.43 support for Unity IAP has been added. It uses the Unity codeless IAP, for detailed information see here: [Unity IAP Documentation](https://docs.unity3d.com/Manual/UnityIAP.html)

1. Go to Window → Package Manager and install the In App Purchasing package
2. Enable Unity IAP in the Services Tab (Ctrl+0). Import the Package by clicking on “Import”. When asked if you would like to run the Unity IAP installer, click “Install Now”.
3. After installation is completed, go to Window → Unity IAP → IAP Catalog and create an IAP item, by entering ID, Title, Description and Price

For our example we use the following:

ID: **remove_ad**

Type: **Non_Consumable**

Title: **Remove Ads**

Description: **Removes Ads from the Game**

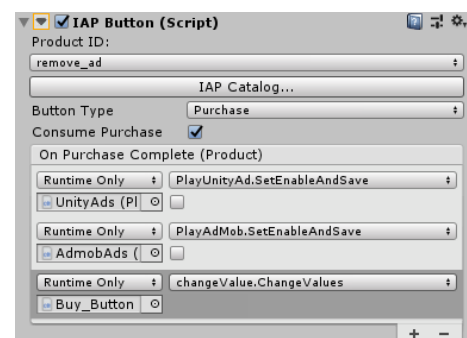
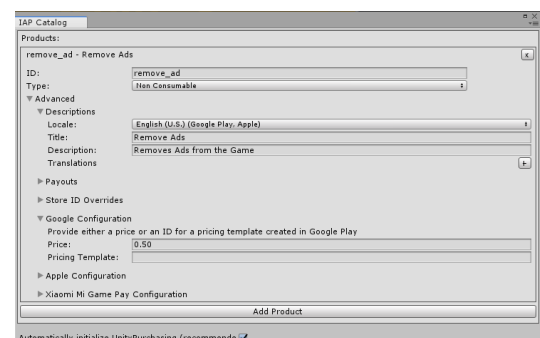
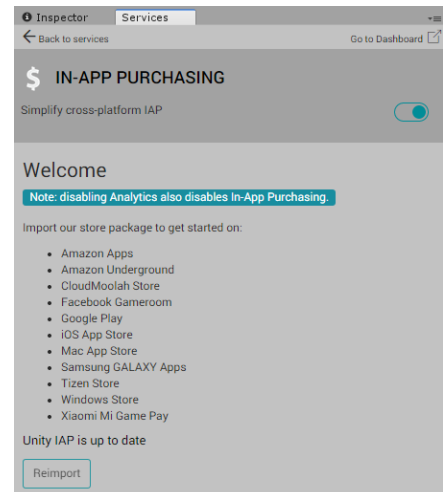
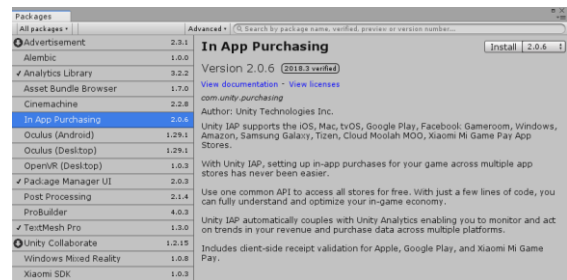
Price: (in Google Configuration) **0.50**

4. In the demo scene an example Button is prepared. It is located in MenuCanvas → SettingsPanel → IAP_Panel → Buy_Button. Select the Button and click on “Add component” and add “IAP Button”, then call the function **SetEnableAndSave** for both the PlayUnityAd and PlayAdMob scripts and the **changeValues** function for the ChangeValue script which is attached on this button.

5. Now select the RestorePurchase_Button, click on “Add component” and add “IAP Button”, then select “Restore” as Button Type.

You can also add a complete new IAP Button by right-clicking in the Hierarchy window → Unity IAP → IAP Button

Please note: After purchasing the “Remove Ad” you may have to restart the App for the change to take effect.



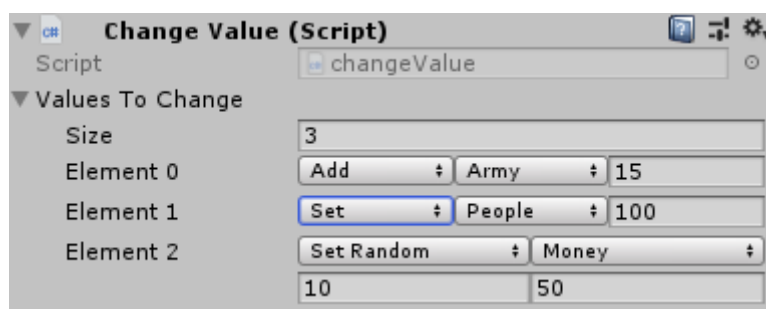
Change Value script

With updated 1.50 we have introduced many new features, like Inventory, Quests, etc., we created a new script “**Make Changes**” (see below) that support all these features, so we recommend to use this instead of the Change Value script.

The Change value script replaces the SetValue and AddValue script, which combines now both scripts into one and offers additional functions.

The Change Value script offers the same options to modify values like the EventScript, but can be used without a card, e.g. if you want to modify values from a game menu with a button.

To trigger the value modifications, you need to call the function **changeValue.ChangeValues**



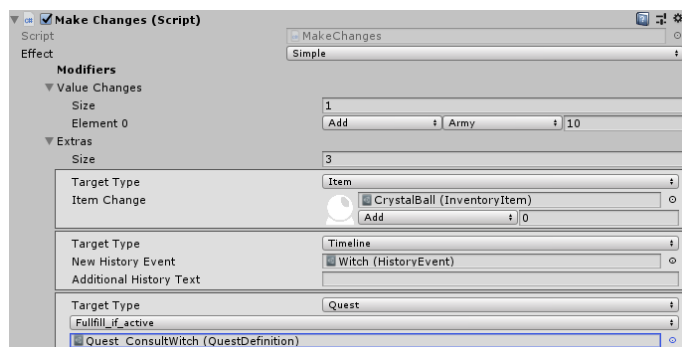
In this example +15 would be added to the Army value, the People value would be set to 100 and the Money value would be randomly set between 10-50.

Make Changes script

The Make Changes script offers the same options as the Change Value script to modify values like the EventScript, but can be used without a card, e.g. if you want to modify values from a game menu with a button. **But it also allows to modify the new features like Items, Dictionary, Quest, Timeline and Gamelog.**

As in the Event Script you have the option to choose between Simple, Conditional, Random Conditions and Random. For detailed information about this, look at the [Event Script](#).

To trigger the value modifications, you need to call the function **MakeChanges.ExecuteEffect**



Inventory

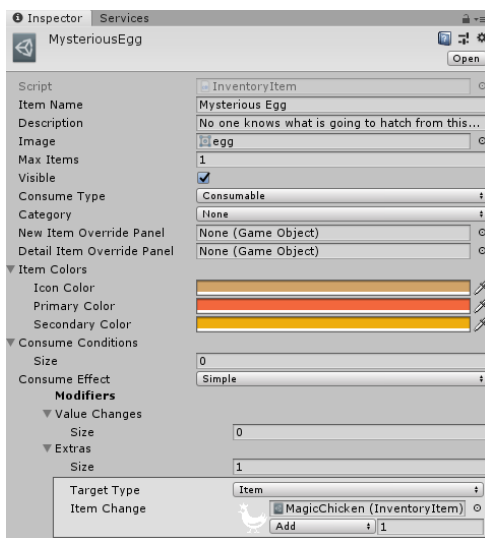
The Inventory allows you the use of Items.

Items are created as Scriptable Objects and are located in:

Kings → addons → Inventory → Resources

All items need to be in this directory!

You can create them by right-clicking → Create → New Inventory Item, or by duplicating an existing one.



Max Items: Limits the number of this item

Visible: If you do not want to display this item, uncheck it

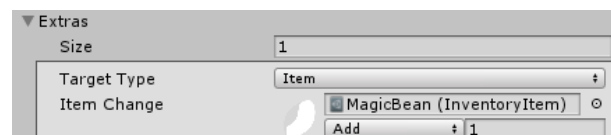
New Item Override Panel: If you want to use a custom prefab for “New Item Popup” you can link it in here

Detail Item Override Panel: If you want to use a custom prefab for “Detail Item View” you can link it in here

Consume Conditions: You can prevent consumption of this item by adding conditions, e.g. if an item increases Army value, you can set condition Army = 0-99, this mean it can be used if value is already 100, since it wouldn’t have any effect.

Consume Effect: Allows “Values” and “Extras” changes on Item consumption

You can **add/remove** or **set** Items in the [EventScript](#) or [MakeChanges](#) script in the “Extras” category.



Example Card: Merchant (in Cards → General)

The **Inventory script** (in Scripts) allows you to configure the display and handling of the items.

Detail View Panel_Prefab: This prefab that will be spawned to display the detailed view of an item.

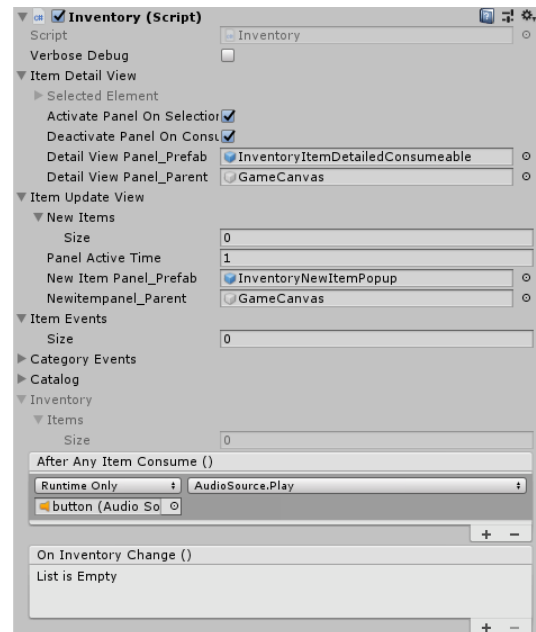
New Item Panel_Prefab: This prefab will be spawned when you receive a new item.

Item Events: You can create specific events on Item changes

Category Events: You can create specific event on Category changes

After any Item Consume: triggers an event when an item is consumed

On Inventory Change: triggers an event when items are added or removed

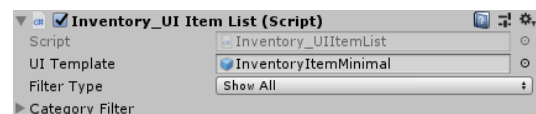


The “Inventory_UIItemList” script is used to display a list of Inventory items.

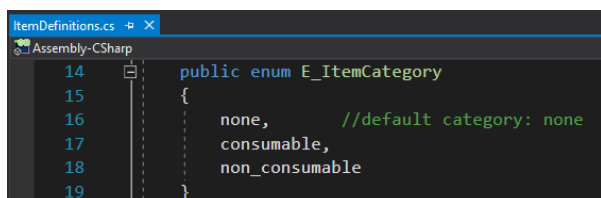
Example: Hierarchy → GameCanvas → Inventory Panel → Inventory List

UI Template: select the prefab template how the items are displayed in this list

Filter Type: allows you to filter items with different categories



You can create Item Categories for displaying/sorting purposes by adding them to the **ItemDefinitions** list, located in Kings → addons → Inventory → Scripts.



If you want to reset the Inventory, call the function: **Inventory.DeleteInventory**

Dictionary

The Dictionary allows you to save any kinds of text.

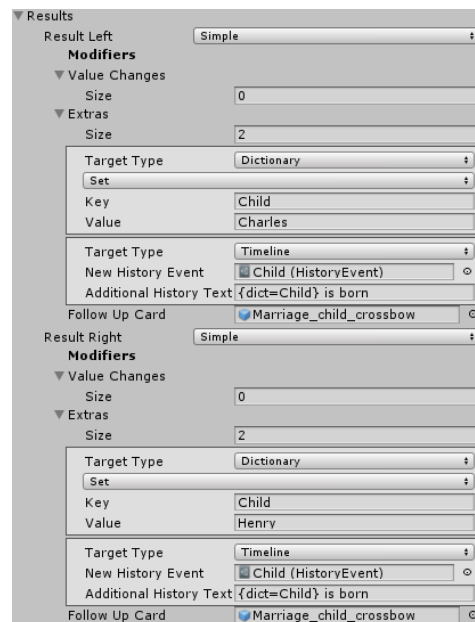
All entries in the Dictionary will be automatically saved and opposite to the "Values" you don't have to create a list/add an object in Hierarchy.

Example Card: Marriage_child_born

In this example the player has to choose a name for his son. If the swipes left, the dictionary entry: "Child=Charles" will be created, if he swipes right it will be "Child=Henry".

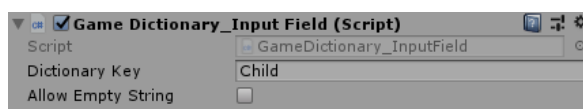
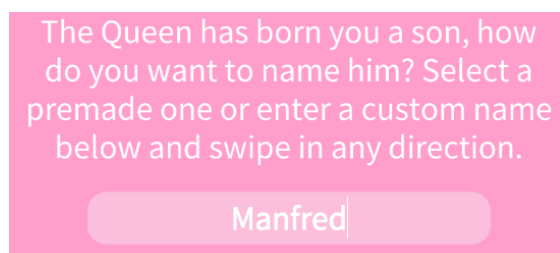
Below the Dictionary entry, there is also a [Timeline](#) Event. It will add the entry: "Charles/Henry is born" to the Timeline depending on the player choice. This is done with help of the Text Variables script, which can replace specific text with custom data, e.g. Dictionary entries, Values etc.

E.g. with the command {dict=Child} the child name the player has chosen (Charles/Henry) will be displayed.



User Input

With the "Game Dictionary_Input Field" script you have the option to let the user create their own entries into the dictionary, e.g. for the below example to name their son.

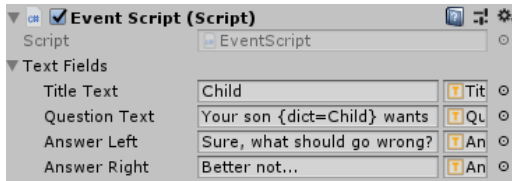


Example Card: Marriage_child_born → InputField

To keep an overview of all Dictionary entries we highly recommend to create a spreadsheet.

	A	B	C	D
1	Keys	values		
2	MagicBean	yes		
3	Child	Charles	Henry	custom

Variables / Text Replacement



Variables allow you to show Values, Dictionary entries and Item data in a Text field.

Examples:

{value=army}	is replaced by the actual value of army, e.g. 75
{value=army,format=0.0}	is replace like before but with formatting, e.g. 75.0
{value=army,result=up}	is replaced by the result for army when swiping up
{item=sword}	is replaced by the number of “swords” item in the inventory
{dict=name}	is replaced by the value for the key 'name'

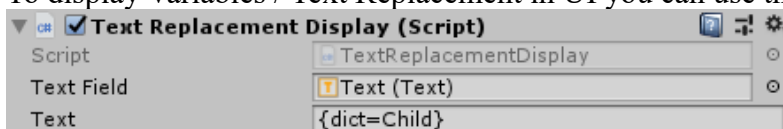
Keywords:

- **Item**
can be followed by format
- **value**
can be followed by preview, format
- **dictionary**
replaces dictionary entry with string, format is not supported
- **change**
similar to result but shows the change of the value, instead of the result
value should be declared first. can have the parameters: **up, down, left, right, add0, add1**
- **result**
similar to change but shows the possible result of the value
value should be declared first. can have the parameters: **up, down, left, right, add0, add1**
- **format**
can be used every time and defines the formatter of the string cast e.g. '0.0', '##' etc.

Please note: Preview function (change/result) does not work for items and external scripts, like “addValuetoValue” script.

Please note: Variables/Text Replacement entries are case sensitive.

To display Variables / Text Replacement in UI you can use the script: **Text Replacement Display**



Quests

Max Nr of Active Quests: set the number of Quests which are active, only active quests can be completed

Auto Refill Active Quests: A new quest will automatically be added if a quest is completed

Quest Fullfillment Popup: the prefab that appears when a quest is completed

Quest Events: allows to trigger Events when a specific quest is completed

Ingame quest test control: gives you several options to test your quests while in Play mode.

In the Example Scene the Active Quests are refilled every new game, with the function: **Quests.FillActiveQuests**, called from the Game State Manager (in Scripts).

Quests are created as Scriptable Objects and are located in:
Kings → addons → Quests → Resources

All quests need to be in this directory!

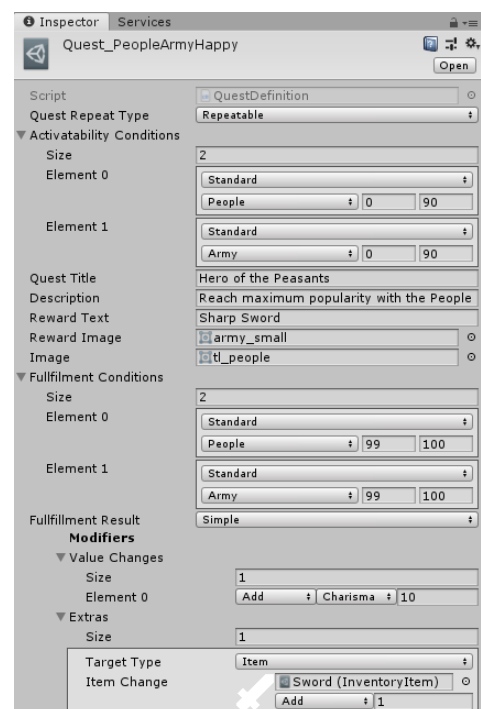
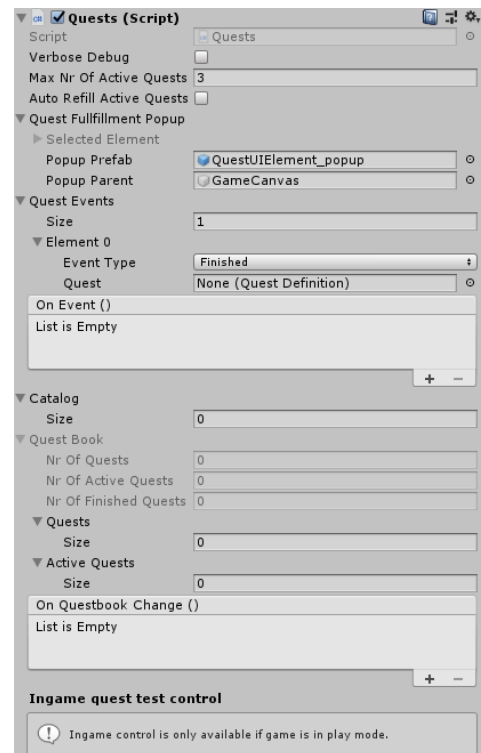
You can create them by right-clicking → Create → New Inventory Item, or by duplicating an existing one.

Quest Repeat Type: you can choose “Repeatable” or “Only Once”, as the name says these can only be completed once.

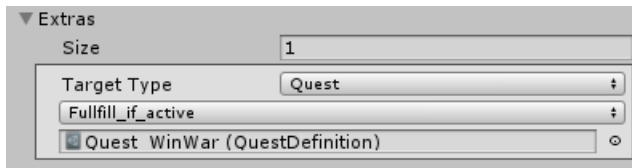
Activatability Conditions: set conditions for quests to allow them to be set active. They are not automatically set to active when conditions are met, Quests are still selected random, but if the conditions are not met, these quests can’t be set active. In this example People and Army value needs to be 90 or below to allow the quest to be set to active.

Fullfilment Conditions: Set the conditions for the quest to be completed. In this example People and Army value have both to be above 99.

Fullfilment Result: Set the value changes / Rewards for when the quest is completed.



Beside the “Fullfilment Conditions” option, quests can also be completed by manually trigger them as completed:



This can be done on the Cards with the EventScript, or with the MakeChanges script. If this method is used, the quest will be completed even when the Fullfilment Conditions are not met.

The “Quest_UIList” script allows to display the quests in a list.

Example: Hierarchy → MenuCanvas → QuestsPanel → ScrollView → Content

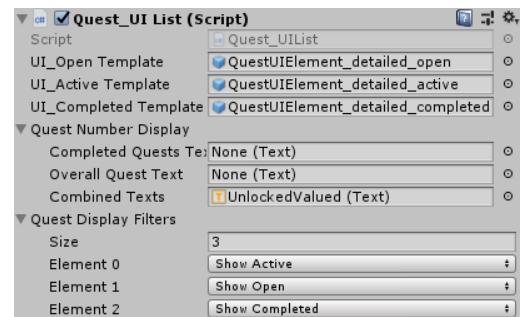
UI_Open Template: prefab for open quests

UI_Active Template: prefab for active quests

UI_Completed Template: prefab for completed quests

→ This allows to separate the quests if you are display them in one list

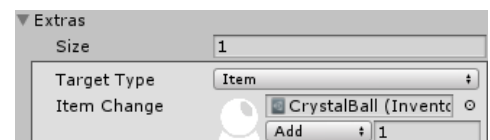
Quest Number Display: display for number of open / completed quests



Quest Display Filters: allows you to choose which quests should be displayed, and in which order. In the above example all quests are displayed in the following order: active quests, open quests, completed quests.

You can **add/remove** or **set** Quests in the [EventScript](#) or MakeChanges script in the “Extras” category.

Example Card: Merchant (in Cards → General)



Timeline

The Timeline feature allow to create a list of History events, to give the player a summary of his actions.

Time Value: select the “Years” value

Start Year: set the start year of the timeline

History events are created as Scriptable Objects and are located in:

Kings → addons → Timeline → Resources

All History events need to be in this directory!

You can create them by right-clicking → Create → New History Event,
or by duplicating an existing one.

The Timeline is displayed with the “Timeline_UIEventList” script.

UI Template: prefab for a History event

Auto Scrollbar Move: check if you want the Timeline to scroll automatically

Display Filler Events: uncheck if you want empty spaces (for each year) between the History events

Nr of Post Fillers: you may need to adjust the number of filler elements if you change the size of the timeline

On Animation End: allows to trigger an event when scrollbar is finished moving

Two examples of different use cases for the Timeline are available:

Scrolling, minimal Timeline: Kings → cards → General → GameOver_Log

Static, detailed Timeline: Kings → cards → General → Timeline

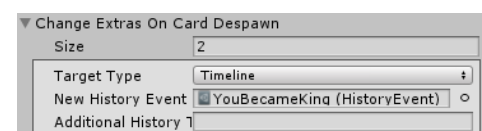
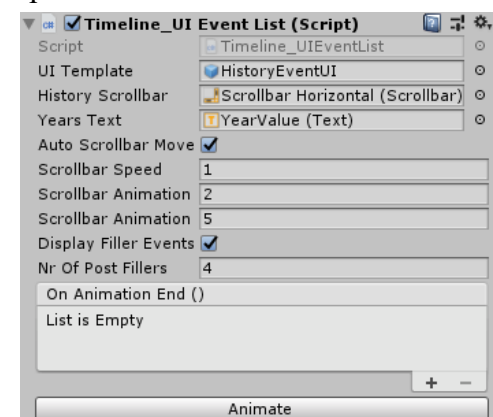
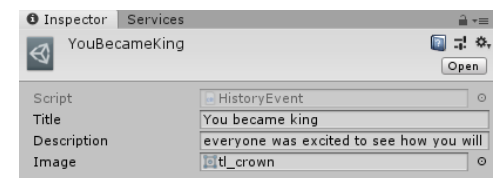
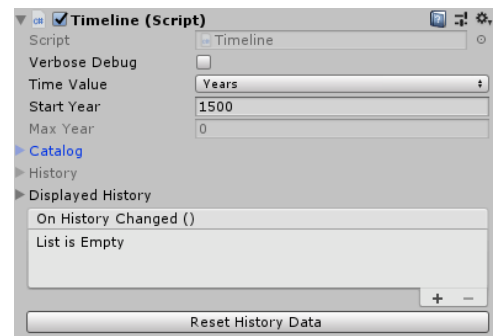
You can **add** Timeline Events in the [EventScript](#) script in the “Extras” category.

The **Additional History Text** gives you the option to add custom text to an History Event.

Example Card: `_StartCard` (in Cards → General)

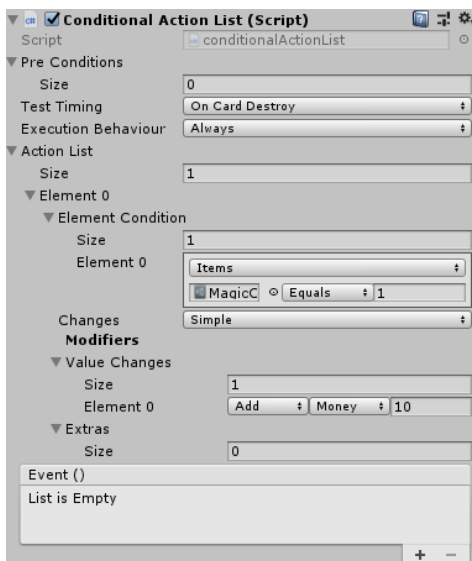
If you want to reset the Timeline, call the function: `Timeline.ResetHistoryData`

Please note: Every card that adds a timeline entry, also needs to add +1 to year.



Conditional Action List

This script allows to trigger Value Changes, Extras and Events when specific conditions are met.



Pre Conditions: Changes are only executed, if all pre conditions are met

Test Timing: lets you set how often the conditions are checked

On Card Destroy: conditions are checked on with every new card

Cyclic: conditions are checked all the time → should only be done in combination with Execution Behavior “On State Change”

Manual: conditions are checked only manually when function: “ExecuteCheck” is called

Execution Behavior: let you set when the value/extra changes are done

Always: Every time a check is done and the conditions are met the changes are executed

onStateChange: Execute if a state changes

Action List Size: allows to create a list of actions, e.g. if you want trigger separate events for a list of value ranges, e.g.

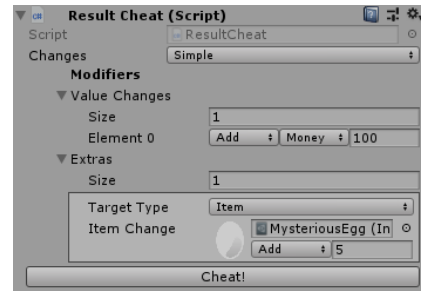
Army value 0-10 = Event1

Army value 11-50 = Event 2

Army value 51-100 is Event3

Result Cheat (in Scripts)

To allow easier testing we have added this script, that lets you change values / items etc. while in Play Mode.



Delayed Events

Delayed Event allows you to delay a specific event or set a countdown for an event.

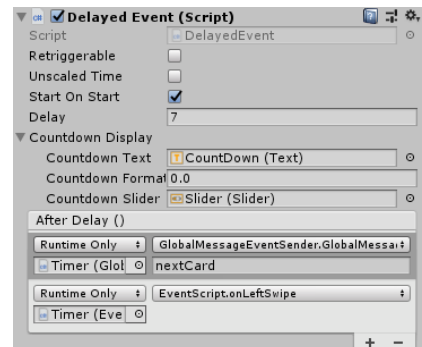
Start on Start: check if you want to start the delay/countdown as soon as the game object with the Delayed Event is spawned

Delay: set the delay in seconds

Countdown Display: Link in a Textfield or and a Slider to display the countdown

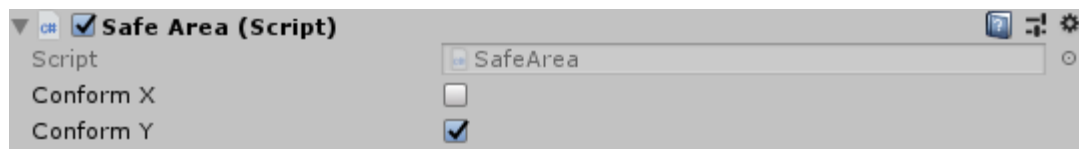
Countdown Format: set the format of the text countdown

Example: Timer card, in Kings → cards → General



Update 1.55: Safe Area Helper (by Crystal Pug)

The Safe Area script helps to adjust the UI to Smartphone Notches / Display Cutouts.



How to use: Simply attach this script to an UI Panel which you want to consider the Notch/Display Cutout and check Conform X / Y, depending if it needs to be adjusted for Landscape/Portrait mode.

To test it in Editor, attached the SafeAreaDemo somewhere in the scene and press key “A” (default setting). After testing, please remove it, since the swipe function may not work correctly on Devices with this script.

Please note: This is a Third-Party asset, developed by Crystal Pug. Many thanks to him for giving permission to include it in the Kings Asset!

Detailed Information about this asset: <https://connect.unity.com/p/updating-your-gui-for-the-iphone-x-and-other-notched-devices>

Asset Store Link: <https://assetstore.unity.com/packages/tools/gui/safe-area-helper-130488>

Update 1.55: Orientation Handler

The Orientation Handler allows to setup the UI that it will work with Landscape and Portrait mode.

Detection Type: You can choose between “Device Orientation” or “By Resolution”.

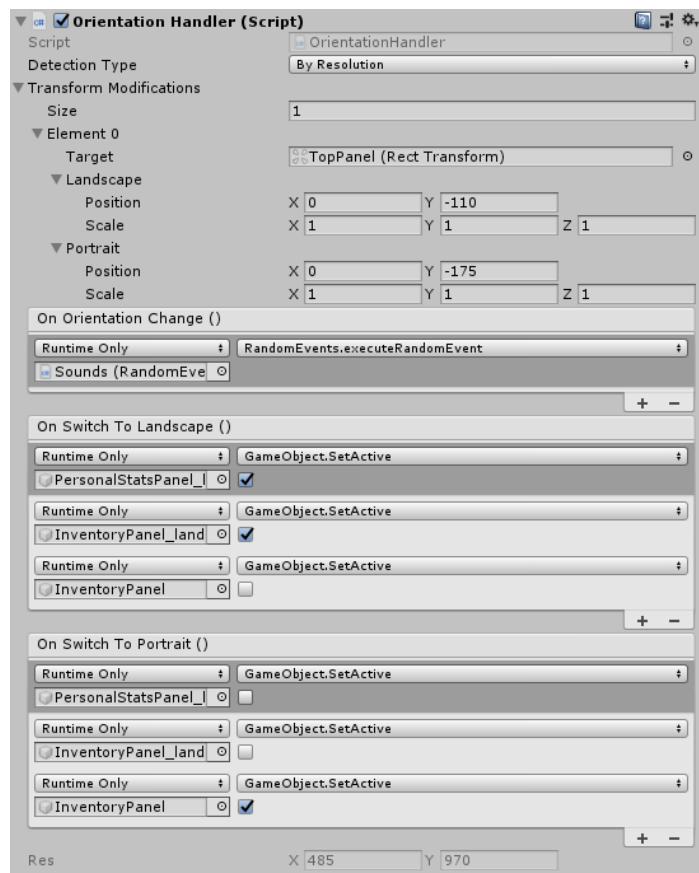
By Resolution has the advantage that it works also on Desktop.

Transform Modifications: Change the Position and Scale of UI Elements depending on Screen Orientation.

On Orientation Change: Triggers Event when orientation is changed

On Switch to Landscape: Triggers Event when orientation is changed to Landscape

On Switch to Portrait: Triggers Event when orientation is changed to Portrait



Update 1.55: Ratio Handler

The Ratio Handler allows UI modification depending on the current Aspect Ratio.

Transform Modifications: Change the Position and Scale of UI Elements depending on Screen Ratio.

On This Ratio Selected: Triggers Event when screen has this ratio

On Orientation Change: Triggers Event when orientation is changed

