UNITY ASSET

Simple Matching Pair Game

Overview

This asset allows developers to quickly integrate the core mechanics of a two of a kind matching pair game into pre-existing or new Unity projects, with minimal, if not any code.

Example Scene

- 1. Example Pair Game Portrait
- 2. Example Pair Game Landscape

Setup Video

- https://youtu.be/dKgG8NHQwKA

Tutorial

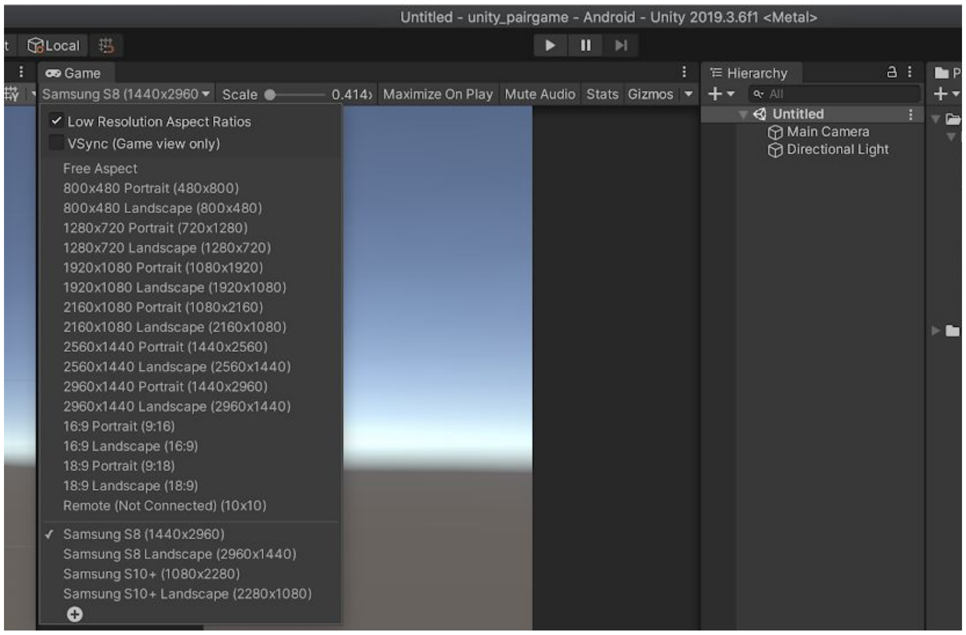
This section covers over how to add change/config/add more grid types/layouts

Setup Screen resolution

The examples of this asset have been developed for mobile portrait resolutions/orientations however, can easily be modified to fit landscape orientations, simply by modifying the grid to have more columns than rows.

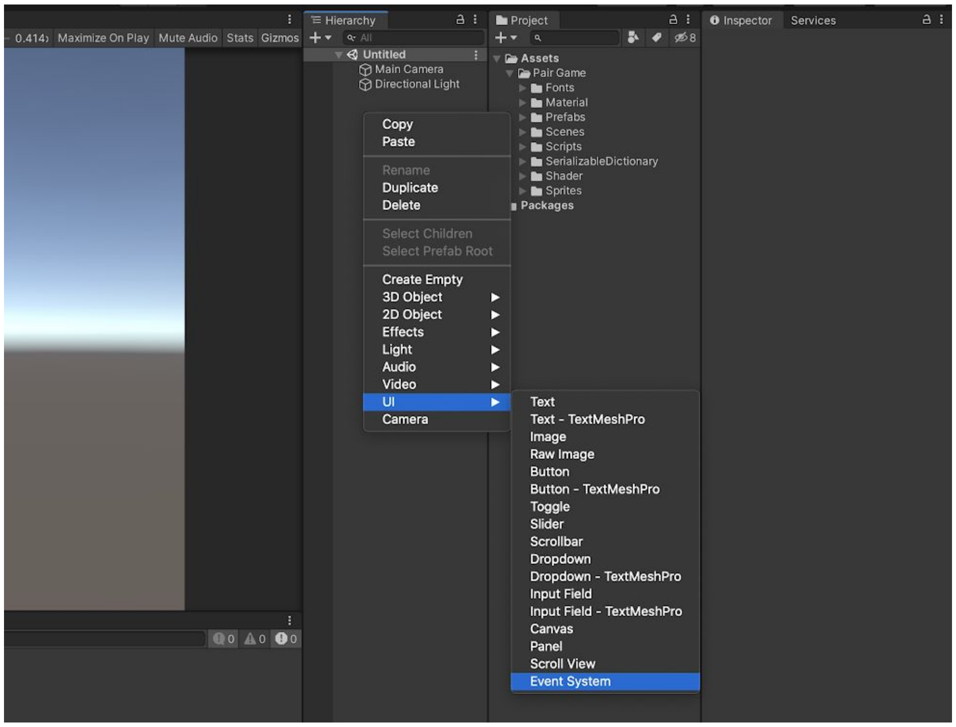
For the purpose of this tutorial, we're going to assume that you want to develop your game for mobile portrait resolution/orientation.

Go ahead setup your target resolution - in this example, we are specifically targeting Samsung S8 (1440x2980)


Untitled - unity_pairgame - Android - Unity 2019.3.6f1 <Metal>
Local
Game
Samsung S8 (1440x2960)
Scale
0.414
Maximize On Play
Mute Audio
Stats
Gizmos
Hierarchy
All
Untitled
Main Camera
Directional Light
✓ Low Resolution Aspect Ratios
VSync (Game view only)
Free Aspect
800x480 Portrait (480x800)
800x480 Landscape (800x480)
1280x720 Portrait (720x1280)
1280x720 Landscape (1280x720)
1920x1080 Portrait (1080x1920)
1920x1080 Landscape (1920x1080)
2160x1080 Portrait (1080x2160)
2160x1080 Landscape (2160x1080)
2560x1440 Portrait (1440x2560)
2560x1440 Landscape (2560x1440)
2960x1440 Portrait (1440x2960)
2960x1440 Landscape (2960x1440)
16:9 Portrait (9:16)
16:9 Landscape (16:9)
18:9 Portrait (9:18)
18:9 Landscape (18:9)
Remote (Not Connected) (10x10)
✓ Samsung S8 (1440x2960)
Samsung S8 Landscape (2960x1440)
Samsung S10+ (1080x2280)
Samsung S10+ Landscape (2280x1080)

Add Event system

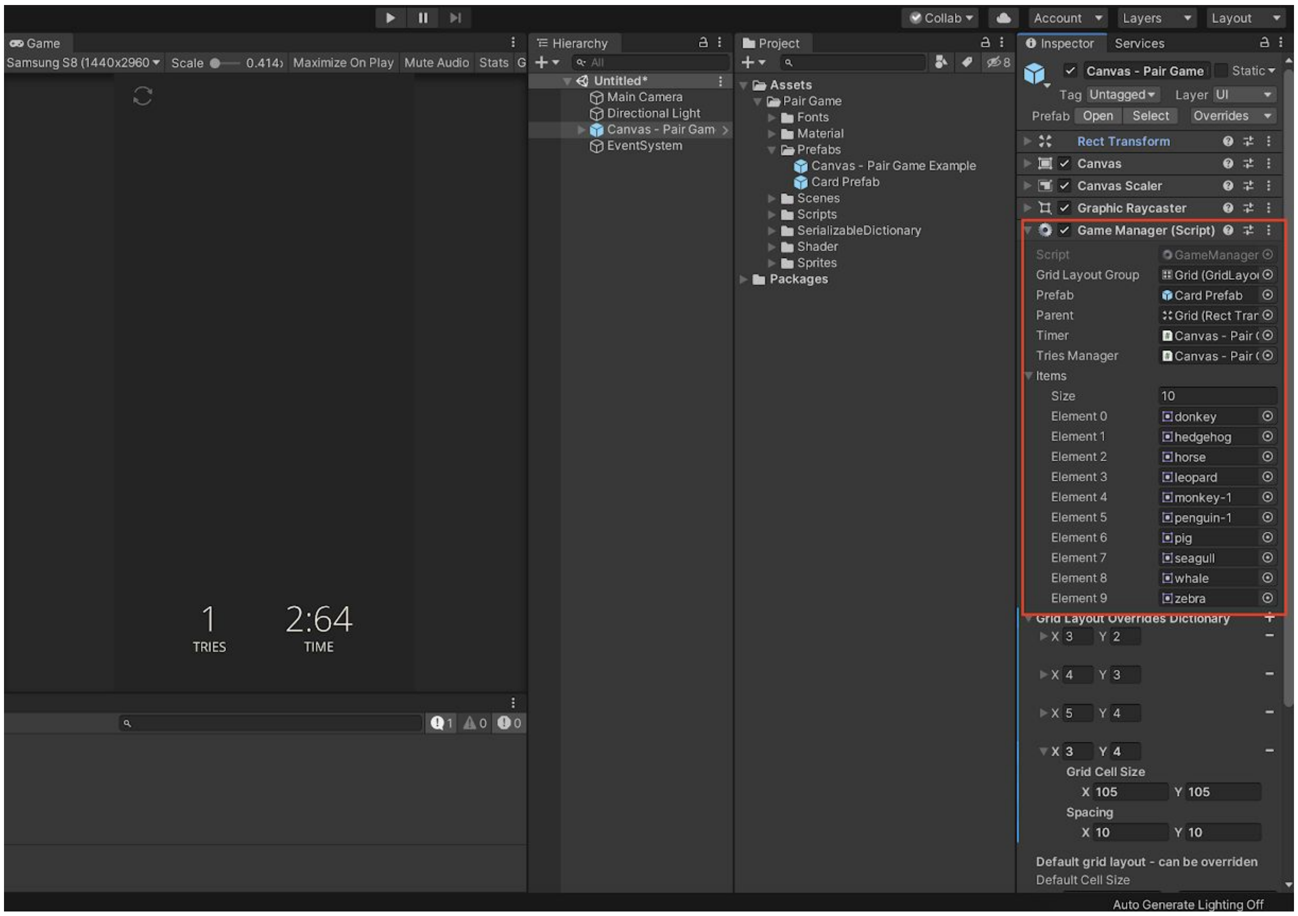
If it doesn’t already exist, add an event system. Doing so by right-clicking in the “hierarchy window” > UI > Event System



Drag and drop Canvas - Pair Game Example into your scene

Select the Canvas - Pair Game Example Prefab and drag and drop in into the hierarchy window/scene.

Attached you will see GameManager.cs(this is where you will do all configuration for the game, such as adding sprites for your cards), TriesManager.cs, TimerManager.cs scripts. They are all pretty self-explanatory however you can find more details towards the bottom of this document.



Canvas - Pair Game — Static
Tag Untagged — Layer UI
Prefab Open Select Overrides

Rect Transform
Canvas
Canvas Scaler
Graphic Raycaster
Game Manager (Script)

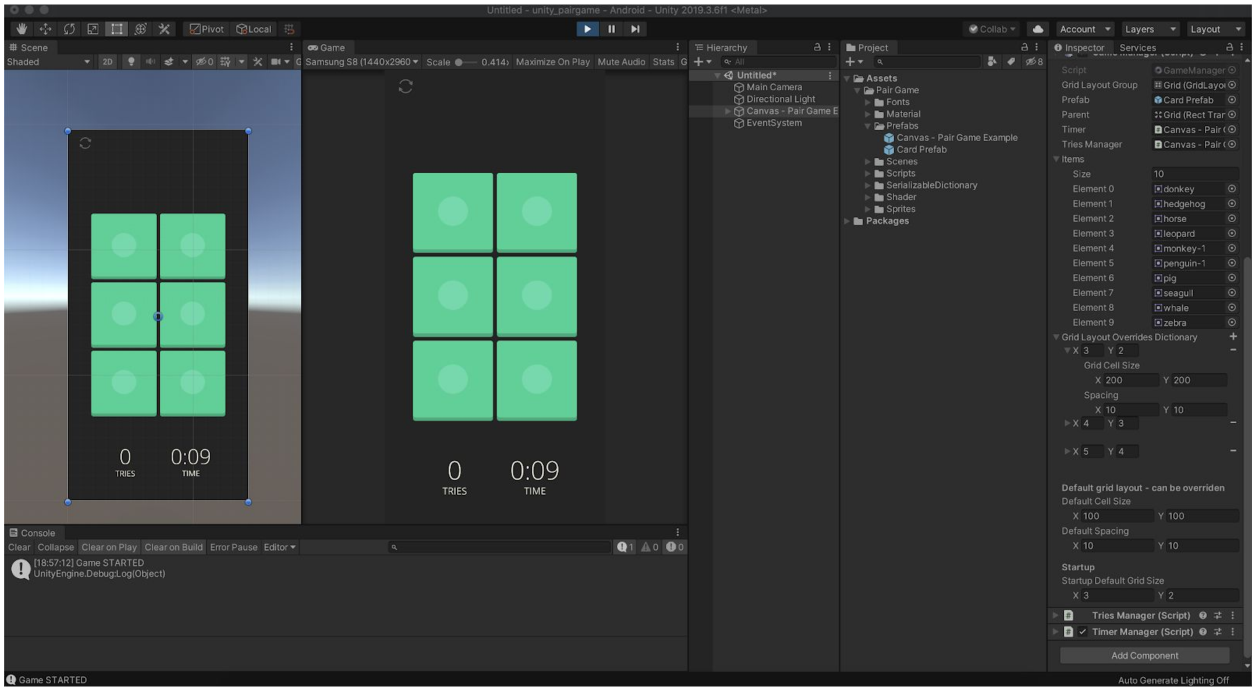
Script	GameManager
Grid Layout Group	Grid (GridLayo
Prefab	Card Prefab
Parent	Grid (Rect Trar
Timer	Canvas - Pair (
Tries Manager	Canvas - Pair (
Items	
Size	10
Element 0	donkey
Element 1	hedgehog
Element 2	horse
Element 3	leopard
Element 4	monkey-1
Element 5	penguin-1
Element 6	pig
Element 7	seagull
Element 8	whale
Element 9	zebra

Grid Layout Overrides Dictionary
X 3 Y 2
X 4 Y 3
X 5 Y 4
X 3 Y 4
Grid Cell Size
X 105 Y 105
Spacing
X 10 Y 10

Default grid layout - can be overriden
Default Cell Size

Hit run

Hit run! By default the asset is setup to spawn a 3x2 grid - let’s change that



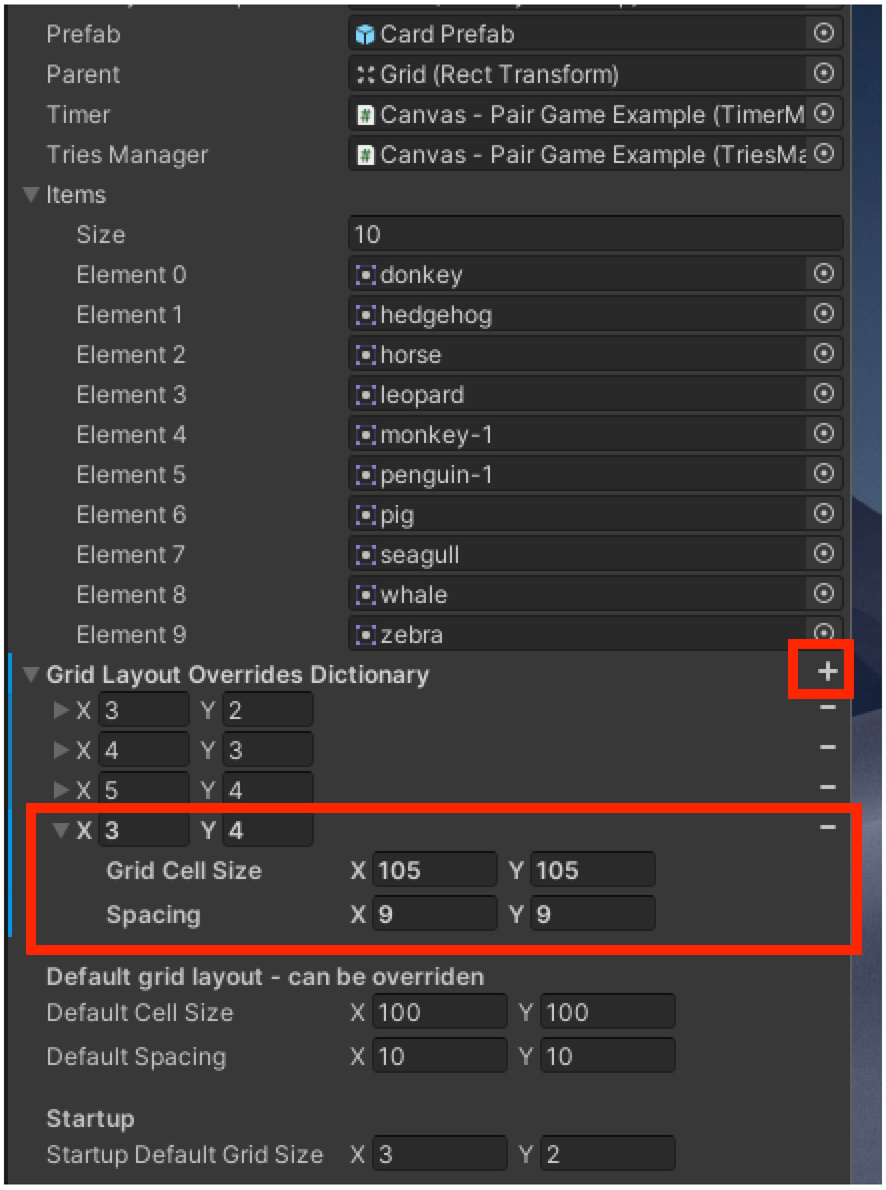
```
Script                     GameManager
Grid Layout Group          Grid (GridLayo
Prefab                     Card Prefab
Parent                     Grid (Rect Trar
Timer                      Canvas - Pair (
Tries Manager              Canvas - Pair (
Items
  Size                     10
  Element 0                donkey
  Element 1                hedgehog
  Element 2                horse
  Element 3                leopard
  Element 4                monkey-1
  Element 5                penguin-1
  Element 6                pig
  Element 7                seagull
  Element 8                whale
  Element 9                zebra
Grid Layout Overrides Dictionary
  X 3   Y 2
    Grid Cell Size
      X 200        Y 200
    Spacing
      X 10         Y 10
  X 4   Y 3

  X 5   Y 4

Default grid layout - can be overriden
Default Cell Size
  X 100          Y 100
Default Spacing
  X 10           Y 10

Startup
Startup Default Grid Size
  X 3            Y 2
Tries Manager (Script)
Timer Manager (Script)
```

```
[18:57:12] Game STARTED
UnityEngine.Debug:Log(Object)
```

```
Game STARTED
```

Adding another grid

- Select the Canvas - Pair Game Example Prefab in the hierarchy window
- Within the inspector, locate GameManager.cs
- Within GameManager locate GridLayout Overrides and selected the + symbol, this will add another override/level
- Expand and change cell size of the newly created item in the dictionary to 105, 105 (this is essentially changes card size within the grid)
- Then change the spacing to 9, 9, which essentially is the size of the space between each card.



Prefab	Card Prefab
Parent	Grid (Rect Transform)
Timer	Canvas - Pair Game Example (TimerM
Tries Manager	Canvas - Pair Game Example (TriesMa
Items	
Size	10
Element 0	donkey
Element 1	hedgehog
Element 2	horse
Element 3	leopard
Element 4	monkey-1
Element 5	penguin-1
Element 6	pig
Element 7	seagull
Element 8	whale
Element 9	zebra

Grid Layout Overrides Dictionary +

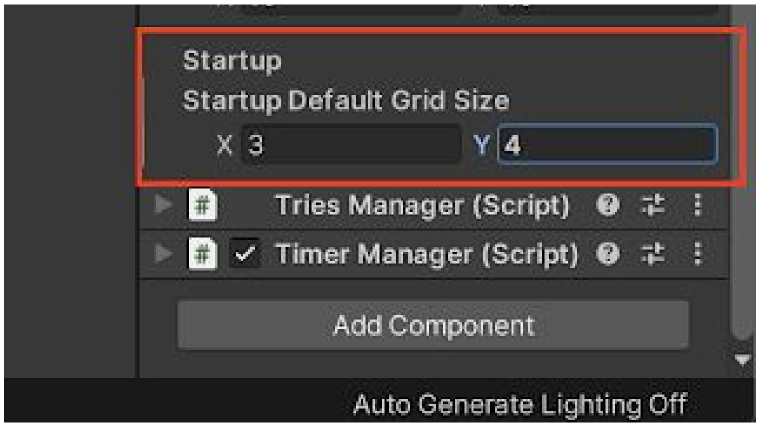
X 3 Y 2 −
X 4 Y 3 −
X 5 Y 4 −
X 3 Y 4 −
Grid Cell Size X 105 Y 105
Spacing X 9 Y 9

Default grid layout - can be overriden
Default Cell Size X 100 Y 100
Default Spacing X 10 Y 10

Startup
Startup Default Grid Size X 3 Y 2

Modify default grid size

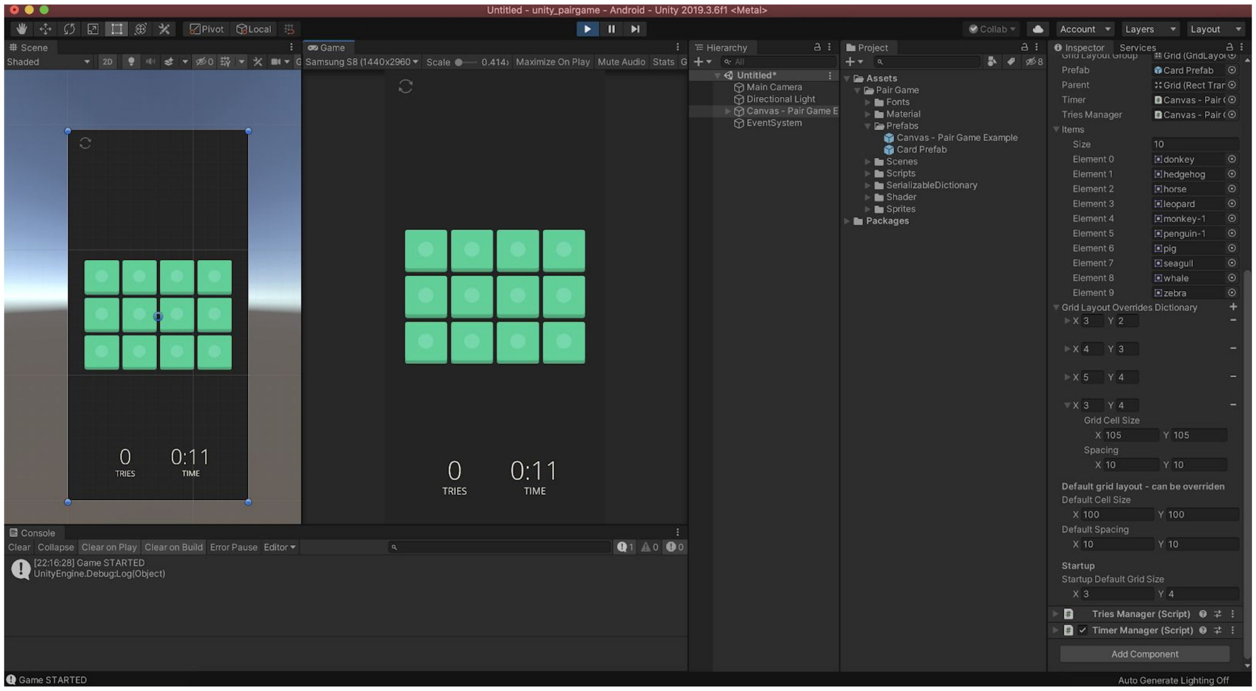
Change the default grid size to 3,4



Startup
Startup Default Grid Size
X 3 Y 4

Hit play

You should now see the default grid 3x4, you just setup. With a cell-size of 105, 105 and spacing of 9,9.



Untitled - unity_pairgame - Android - Unity 2019.3.6f1 <Metal>

Grid Layout Group	Grid (GridLayout
Prefab	Card Prefab
Parent	Grid (Rect Trar
Timer	Canvas - Pair (
Tries Manager	Canvas - Pair (
Items	
Size	10
Element 0	donkey
Element 1	hedgehog
Element 2	horse
Element 3	leopard
Element 4	monkey-1
Element 5	penguin-1
Element 6	pig
Element 7	seagull
Element 8	whale
Element 9	zebra

Grid Layout Overrides Dictionary
X 3 Y 2
X 4 Y 3
X 5 Y 4
X 3 Y 4
Grid Cell Size
X 105 Y 105
Spacing
X 10 Y 10

Default grid layout - can be overriden
Default Cell Size
X 100 Y 100
Default Spacing
X 10 Y 10

Startup
Startup Default Grid Size
X 3 Y 4

```
[22:16:28] Game STARTED
UnityEngine.Debug:Log(Object)
```

That's it! Read the **things to note** section below, to become even more familiar with the asset!

Things to note:

UI sprites shader

The cards require the UI Card Shader, this shader prevents UI from showing if the card is not facing the camera.

Adding/changing sprites

Select Canvas - Pair Game Example Prefab > within the inspector locate GameManager.cs > locate Items list > here you can add and remove sprites that will be used for your game.

Grid Layout Override Dictionary

This dictionary will modify/override the gridlayout. If the setup grid/default grid matches up with a grid defined here. For example if the default grid is 3x2 and there is a key 3x2 within the override dictionary, the gridlayout component would be modified with the values defined in the override dictionary.

Error: Sprite limit

Error Sprite limitation will be thrown if there aren't enough sprites to fill all cards.

Random Sprites

Random Sprites chosen from the GameManager item list are chosen from random for each game, making each game not always the same.

Shuffling

Shuffling of cards make the game more random and hard, this functionality is built within GameManager.cs

Cards.cs

Responsible for hiding and showing/display specified sprites/cards.

Communicating with GameManager externally doing by calling GameManager.Instance then .Method() or .Function()

GameManager

GameManager.cs is the brains/manager of the game, controlling Cards.cs, TriesManager.cs, TimerManager.cs

Change level

You can change the level programmatically by accessing GameManager.Instance.Setup(int _rowCount, int _columnCount)

Restarting the game

You can restart the level by calling GameManager.Instance.Restart()

GameStarted(), CardStarted(), GameOver

You can add additional functionality to GameStarted(), CardsMatched(), GameOver() methods

TriesManager.cs

Responsible for counting the amount of tries a user makes and displays the value using a text component.

TimerManager.cs

Responsible for counting/showing the duration of a session/game.