**Grab A Crab Bonus Game (Design Doc)**

**Written By: Adam Reed**



**Table of Contents:**

**Page 1:** *Table of Contents*

**Page 2:** *Game Specs*

**Page 3:** *Interface Implementation*

**Page 4:** *Creating A New Sequence Button*

**Page 5:** *Creating An Override Sequence*

**Design Specs:**

**Development Engine:** *Unity 2020.3.14*

**Programming Language:** *C#*

**Development Platform:** *PC, Mac & Linux Standalone*

**Game Design/Graphic Design:** *Powerhouse Gaming*

**Developer/Programmer:** *Adam Reed*

**Date Completed:** *August 25th, 2021*

This is a small “Pull Tab” style bonus game that consists of 2 scenes.

* *Main\_Menu*
* *GrabACrab\_Bonus\_Game*

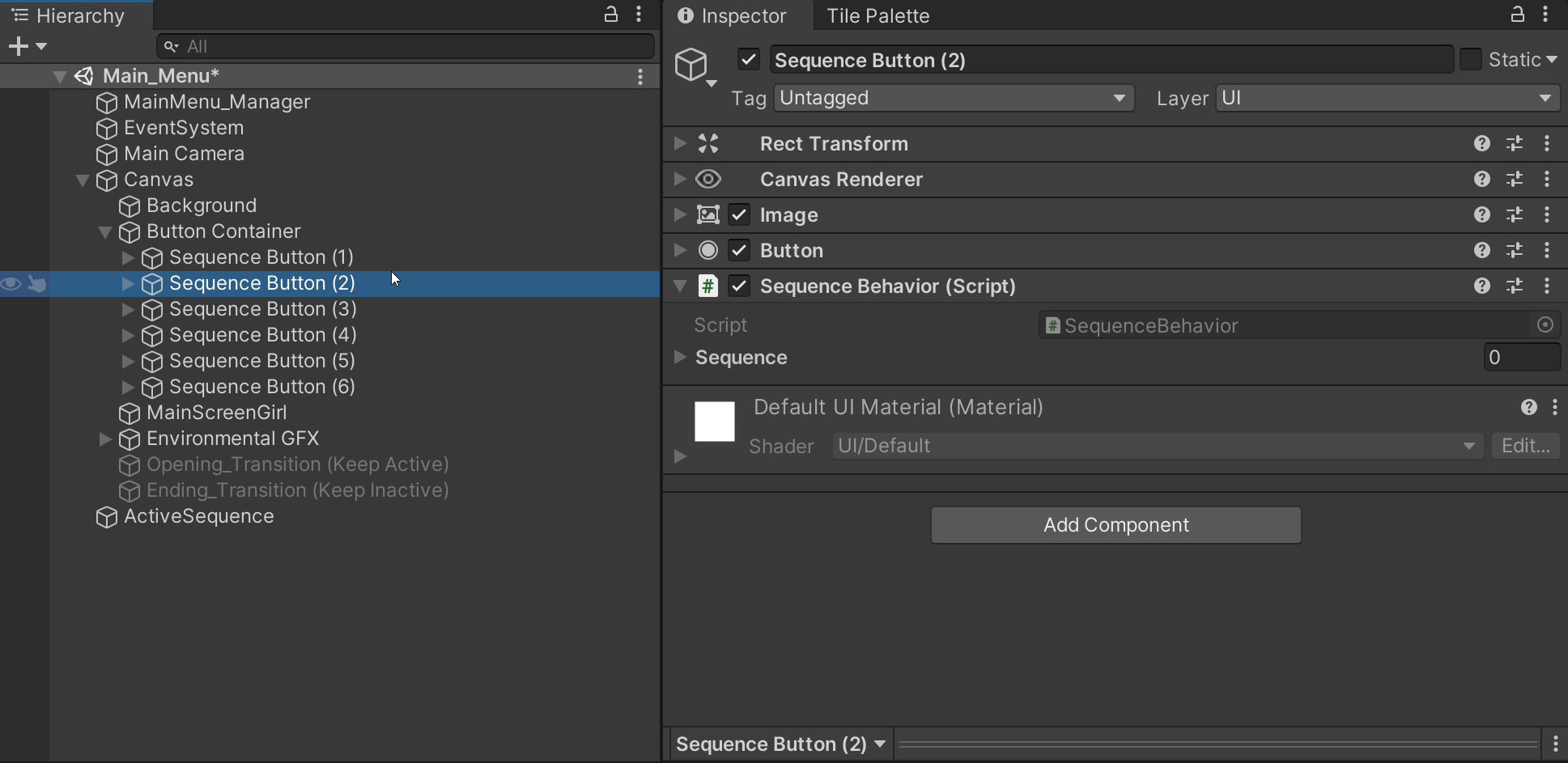
It is built in Unity with a full 2D graphical interface and is designed to fit any screen size/resolution. (i.e 16:9, 720p, 1080p, 4k, etc.)

The winnings in this game are predetermined using sequences that are set and decided using the main menu’s button interface. No matter which crab is selected or whether the player waits and lets one be selected at random, the outcome will remain the same.

Custom sequence overrides can be created and applied causing the override sequence to play no matter which button is selected.

**Interface Implementation:**

When you open the game, you will be presented with a “Landing Page/Scene” titled Main\_Menu. Within this scene you will see a series of buttons that each contain a separate iteration of the attached script titled “SequenceBehavior” which in addition to being a MonoBehavior, derives from the “ISequence” interface script. The Sequence Behavior script allows the developer to pull from the interface and establish a base sequence array within the SequenceBehavior script all from within the Unity Editor.



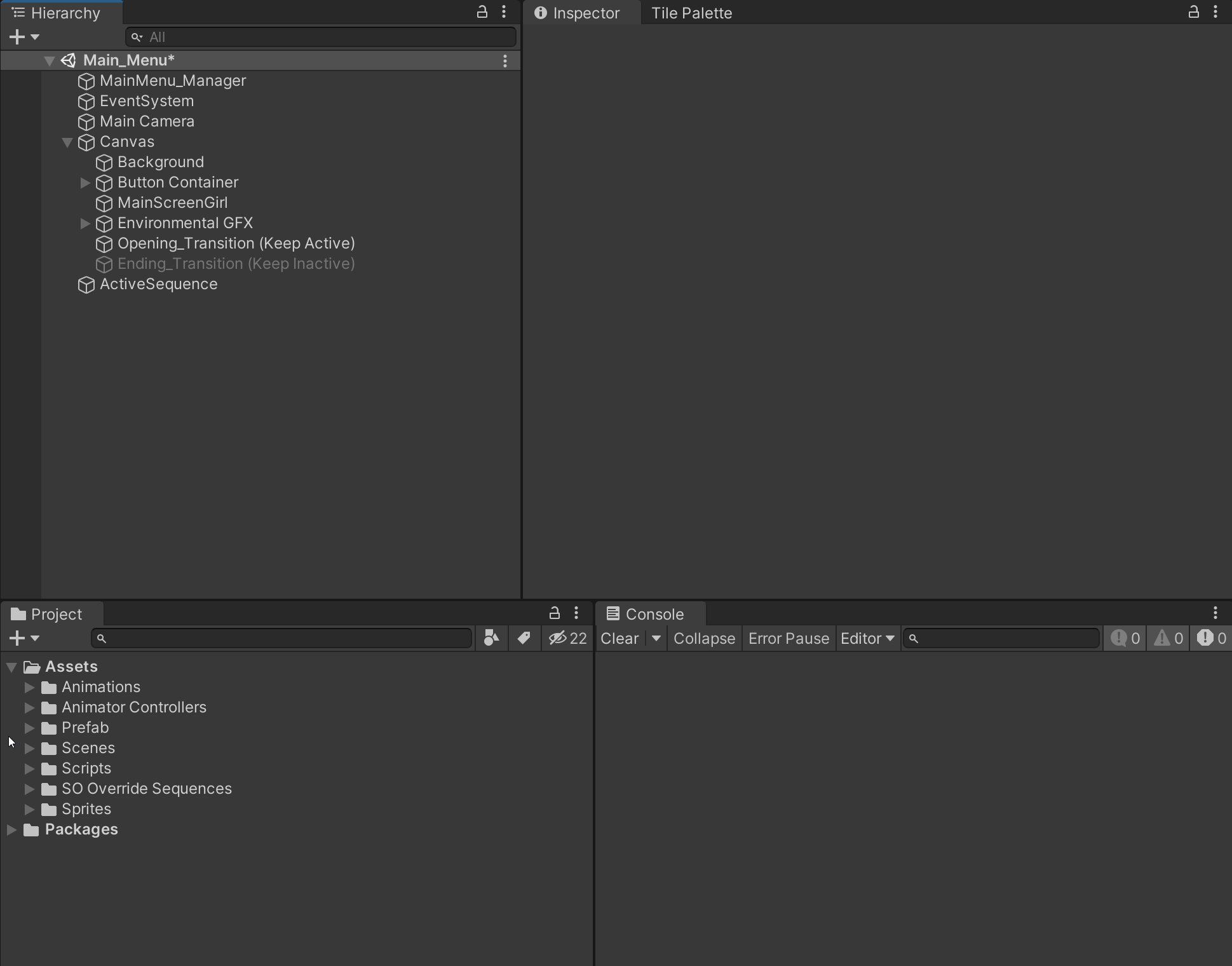
Since “interfaces” do not have the ability to store implementations and cannot contain fields I utilized an object (“Active Sequence”) within the hierarchy to store the sequence data that is pulled from the Sequence Behavior script. The Active Sequence script contains the “DontDestroyOnLoad” method which prevents it as well as the object that contains it from being destroyed upon scene transitions.

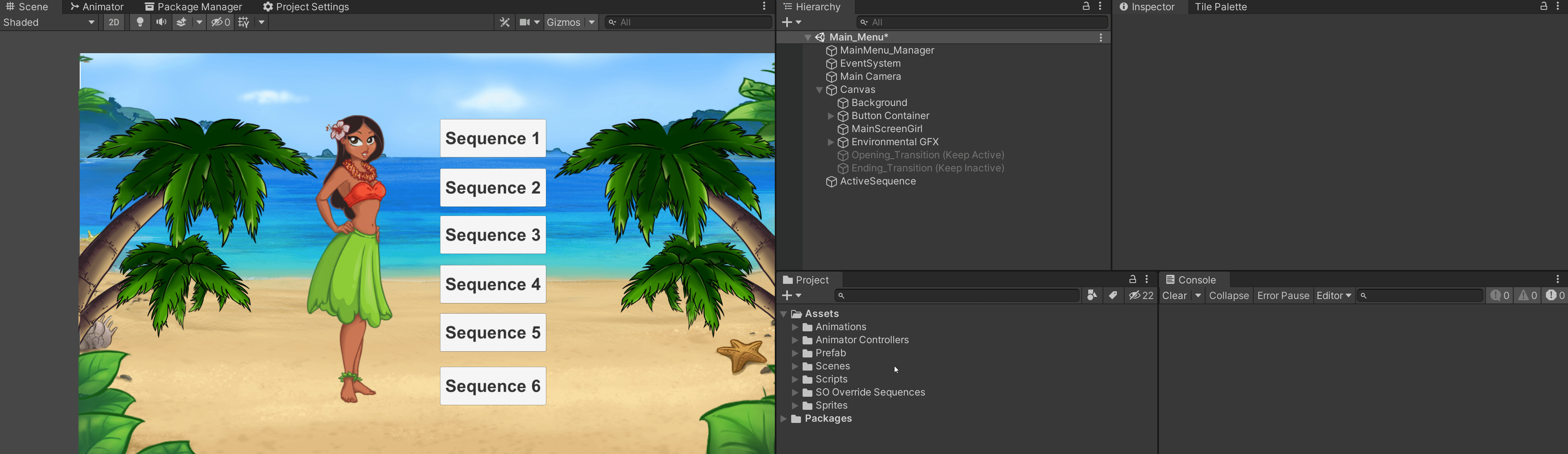
This is what allows the interface sequence data to be carried between multiple scenes.

Once the scene has finished its transition into the “GrabACrab\_Bonus\_Game” scene, the “GameManager” script will search for the “ActiveSequence” script and pull the interface data from it. This then allows the various aspects of the bonus game including features within the UIManager script and the Crabs script to utilize the sequence data and reveal the appropriated winnings in the order of the sequence provided.

**Creating A New Sequence Button:**

Should you want to add a “new” sequence button to the scene, all you have to do is drag and drop the “New Sequence Button” prefab object from project menu (located within “Assets>Prefab”) into the hierarchy. Then simply update the empty sequence within the “SequenceBehavior” script attached to your new button!





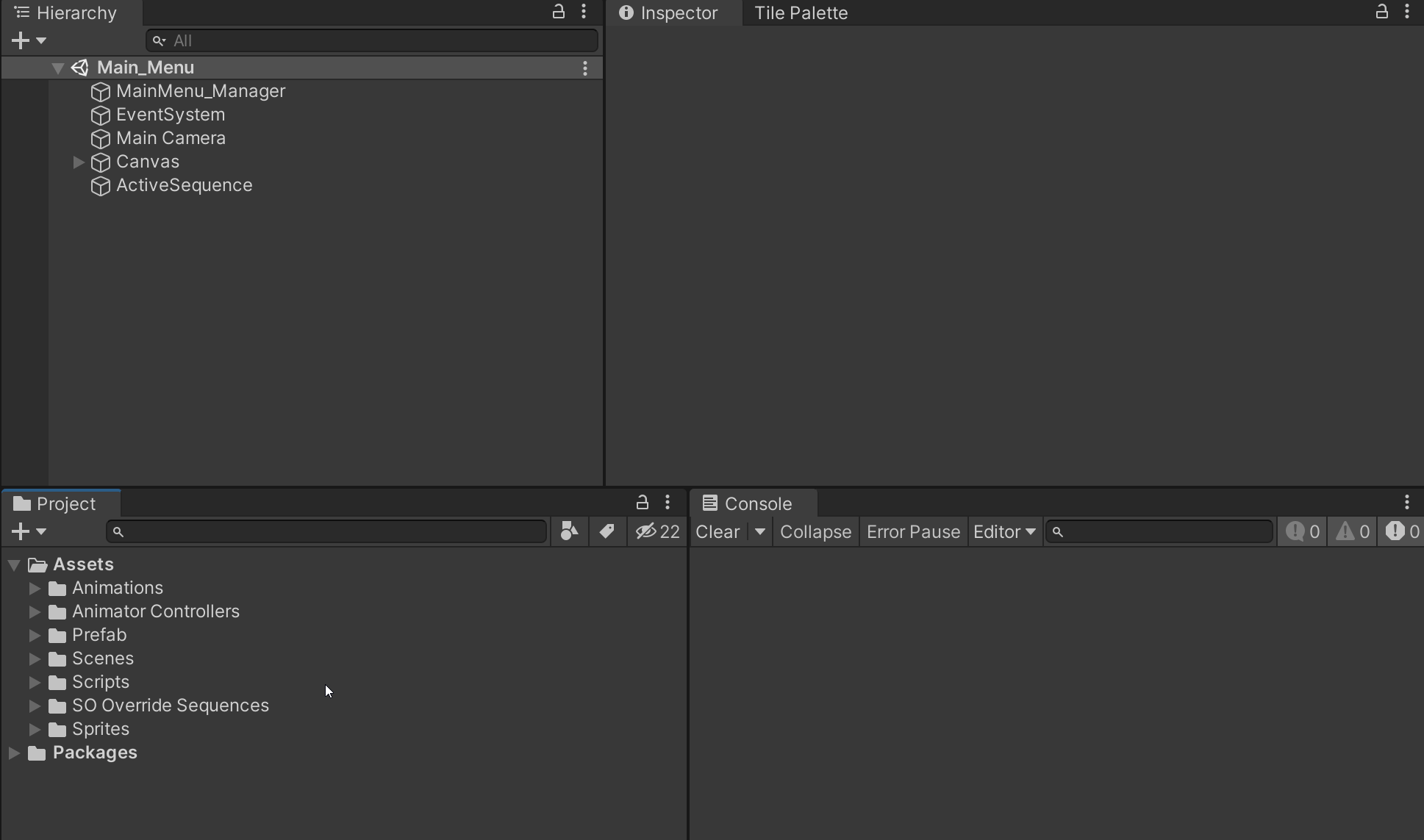
**Creating An Override Sequence:**

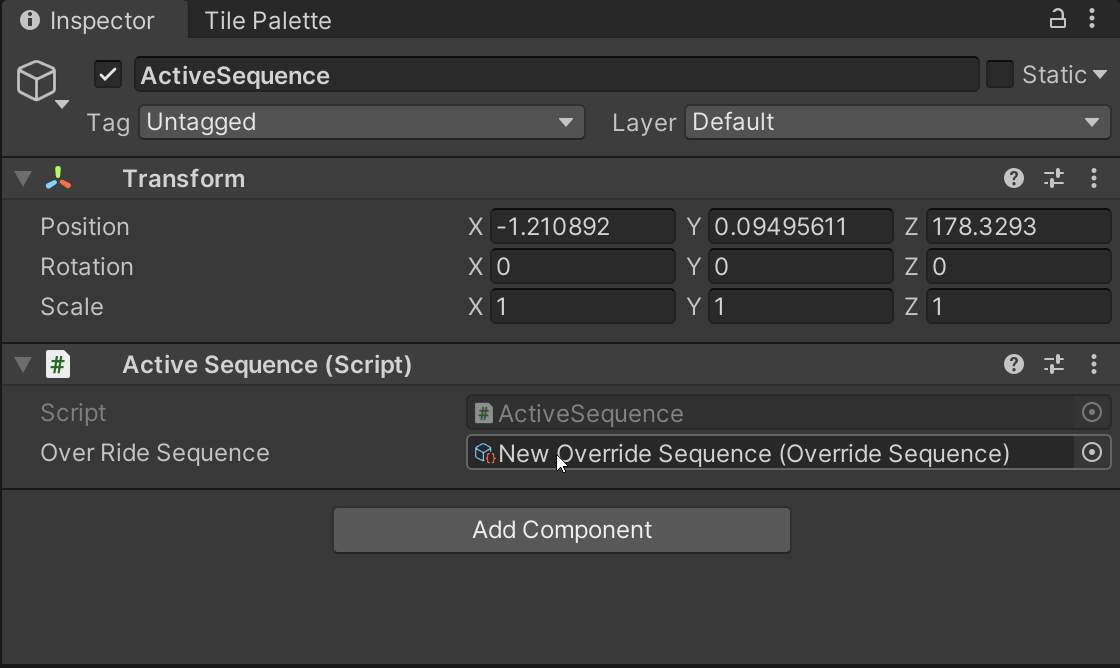
On the other hand, if you are wanting to create an “*arbitrary win sequence*” that can override the current win sequences established within the Sequence Buttons then you can do just that in on a few simple steps.

Simply right click within the project menu and select “Create>Create New Override Sequence”. This will produce a new “Scriptable Object” titled New Override Sequence which contains a float array field that can be custom set in the Unity Inspector.

Lastly, in order to implement this override, just drag and drop your New Override Sequence scriptable object from the project menu, into the “Active Sequence” script’s field titled “Over Ride Sequence”.

This will make it so that no matter which sequence button you press, the override sequence will always be the one that is used.





To remove this override, simply remove the reference to

the scriptable object from the Active Sequence script.