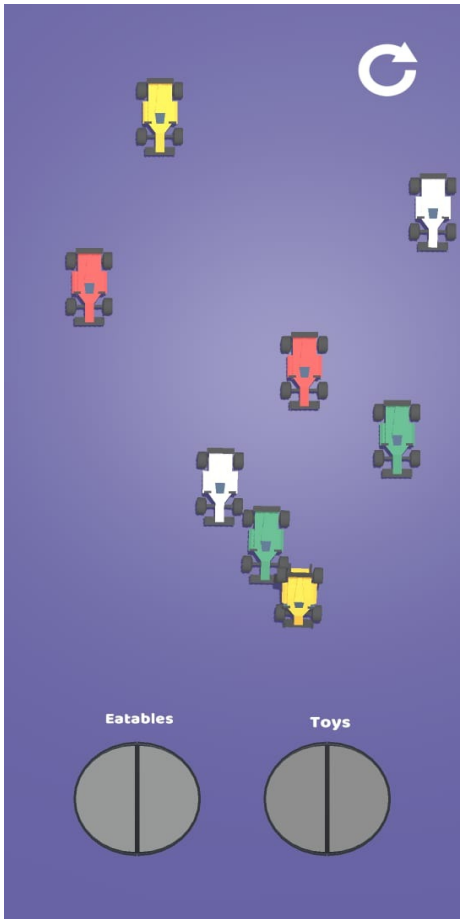
MATCH 3D

(Interview Test, taken by Leyla, Fauna Entertainment)

This project is made with Unity Engine **2019.4.30 (LTS)**, coded using language C#, IDE **JetBrain Rider 2020.3** and **Blender v2.9** for basic model creation.



Design Pattern Used:

- Singleton
- Observer -> To Manage Game State Transitions

Unity Techniques Used:

- Scriptable Object -> To Easily Create New Objects (Matchable/Pickable) and maintain

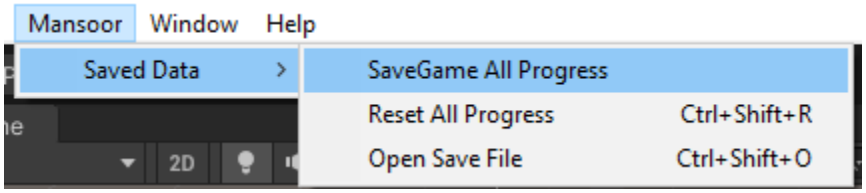
existing objects.

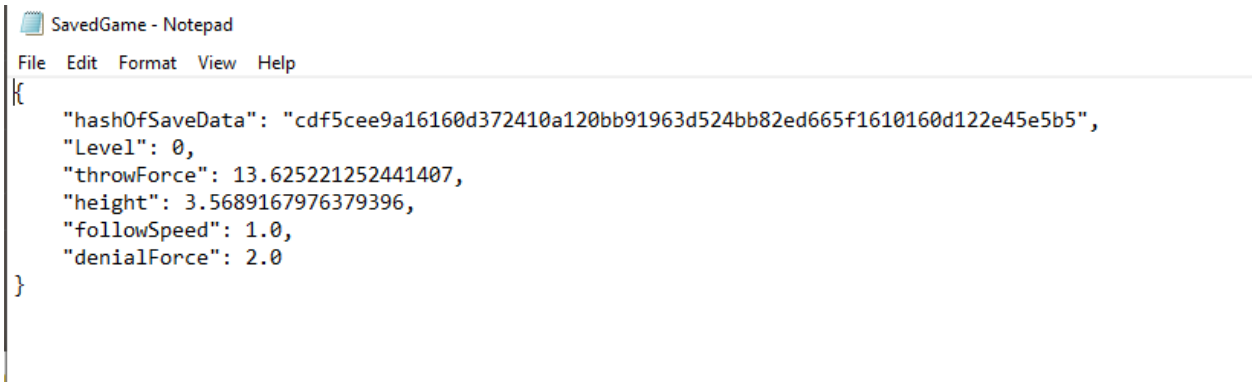
Explanation of Scripts:

- Level Manager.cs: I use Level Manager for handling the enabling or spawning of levels based on the progress we made. Level Manager is also subscribed with Game Progression Manager in order to respond when state is changed. I.e when level is completed. It will respond to save level progress & execute analytics commands.

- Game Progression Manager.cs: A based structure created in observer pattern to help other scripts respond respectively when state is changed.

- SaveData.cs: Game values are necessary to store. This script is responsible for storing, loading, deleting values in a **JSON** format which is **encrypted** with a hash key. Works on all platforms.





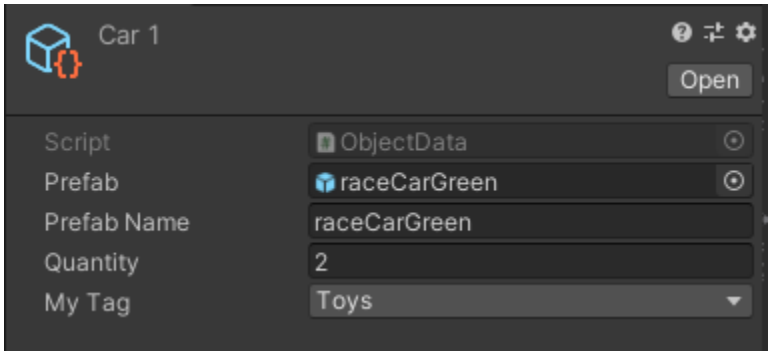
- **PlaneScreenSize.cs :**

```
// dynamic screen size adjustment for surface
var height :double = Camera.main.orthographicSize * 2.0;
var width :double  = height * Screen.width        / Screen.height;
transform.localScale = new Vector3( x: (float) width,  y: (float) height,  z: 0.1f);
```

This script automatically resizes the base surface of our game to any mobile screen & creates colliders around the borders.

-InputSystem.cs: The main script in the game which interacts with user's input and gives output. It allows to move object anywhere in screen.

-ObjectData.cs (Scriptable Object): It's responsibilities is to create scriptable objects of the items we needed in game to match**. Containing Prefab, (auto assigned) name, (Validated) Even Numbers, A Custom Tag.**

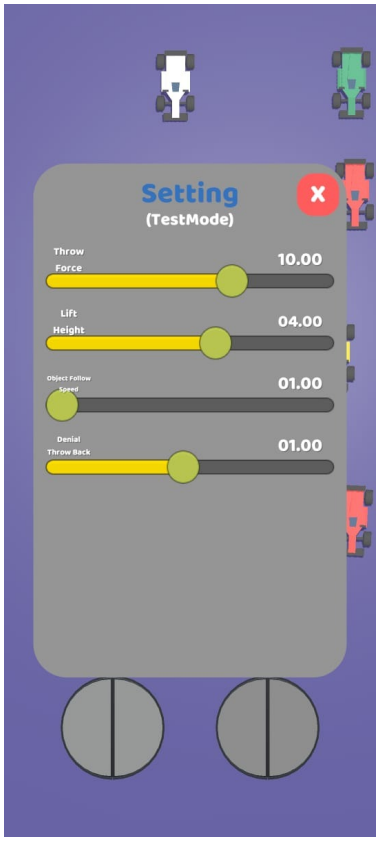


- SliderBinder: One of the important role of this script in game is to change the settings with ease without spending extra time.

```
/// <summary>
/// Initialize slider
/// </summary>
/// <param name="name">name of slider</param>
/// <param name="initValue">initial value</param>
/// <param name="range">min max value of slider</param>
/// <param name="changeValue">Action when value change occur</param>
public void Init(string name, float initValue, Vector2 range, Action<float> changeValue)  4 usages
{

    slider = GetComponentInChildren<Slider>();
    this.changeValue = changeValue;
    nameText.text = name;
    valueText.text = initValue.ToString(format: "F1");
    slider.minValue = range.x;
    slider.maxValue = range.y;
    slider.value = initValue;

}
```



The purpose of this script is to bind slider easily & efficiently for the **QA testers or the Developer** itself to **make changes and test in real time**

There are other scripts i.e **MaterialColorChange** which helps to change color without creating a material, in renderer stage. It is one of the best **solutions** to **prototype Level Design** much **faster.**

Other Utilities:

- Text Mesh pro: To write Text in a 3D world.

- Procedural UI Image: To Create **Custom Images** without downloading or making new one outside of unity engine. We can make it directly in Unity.

-Surge: To Tween some movements & rotation.