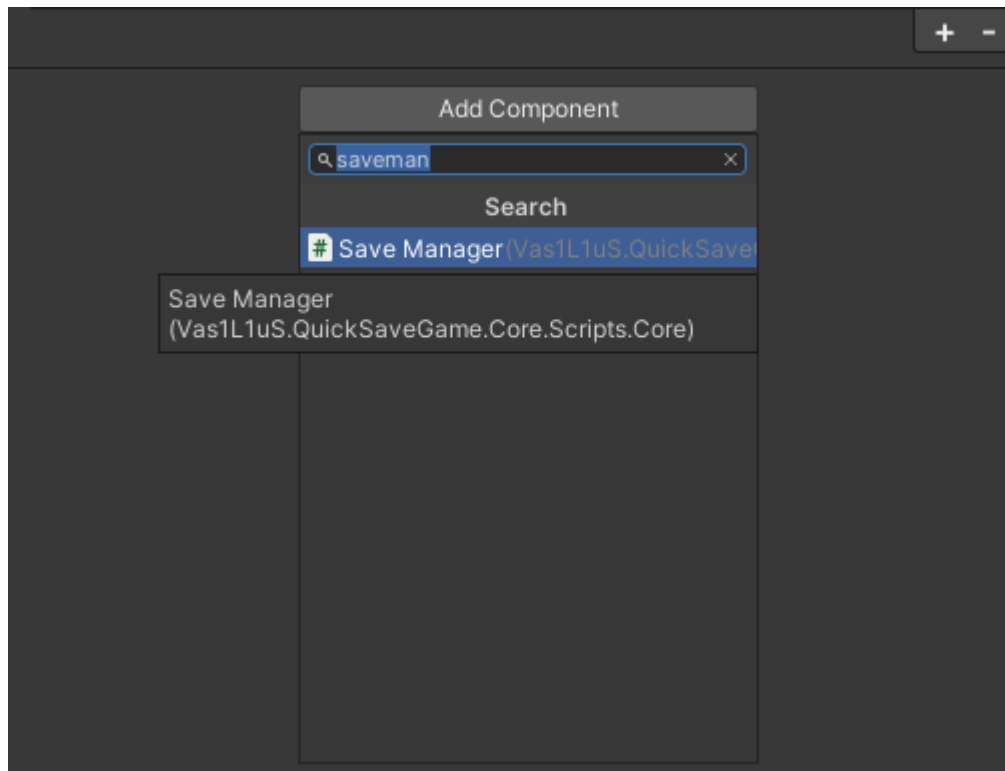
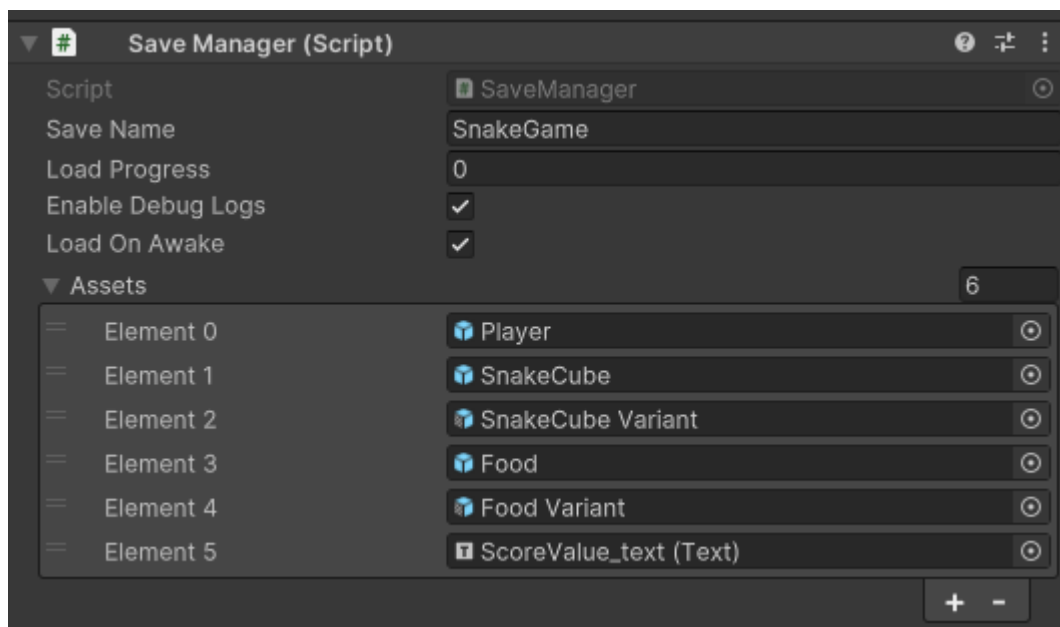


Quick Save Game Documentation



First, create an object on the scene and add the Save Manager component to it.



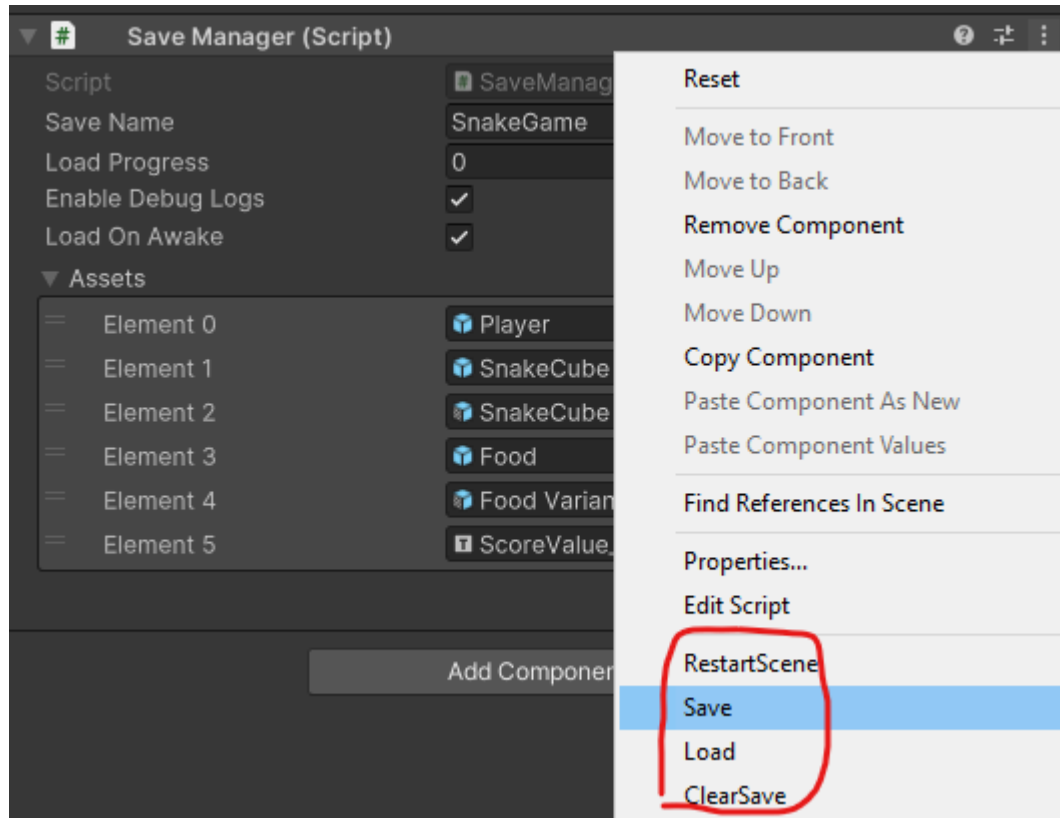
Next, you enter the Save Manager properties in the inspector
-Save Name - name for the current save and load.

-Load Progress - loading progress, you can track the current loading progress through the inspector, relevant for large save files.

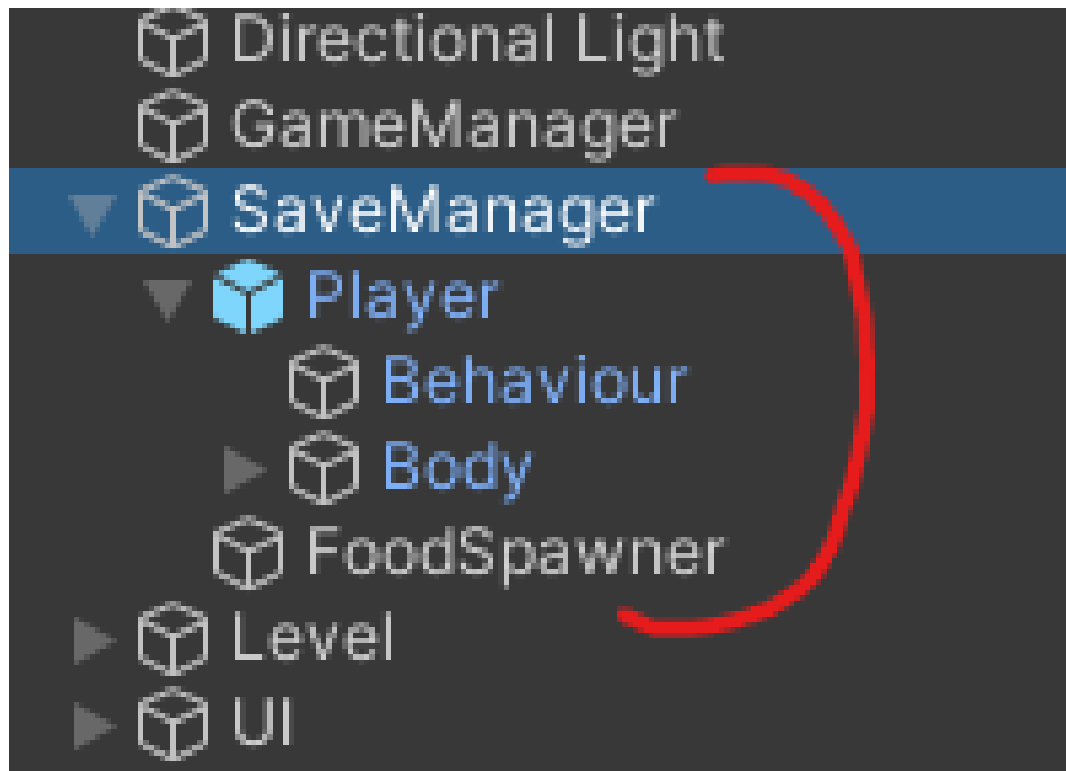
Enable Debug Logs - enables Save Manager logs

Load On Awake - loads the save when waking up

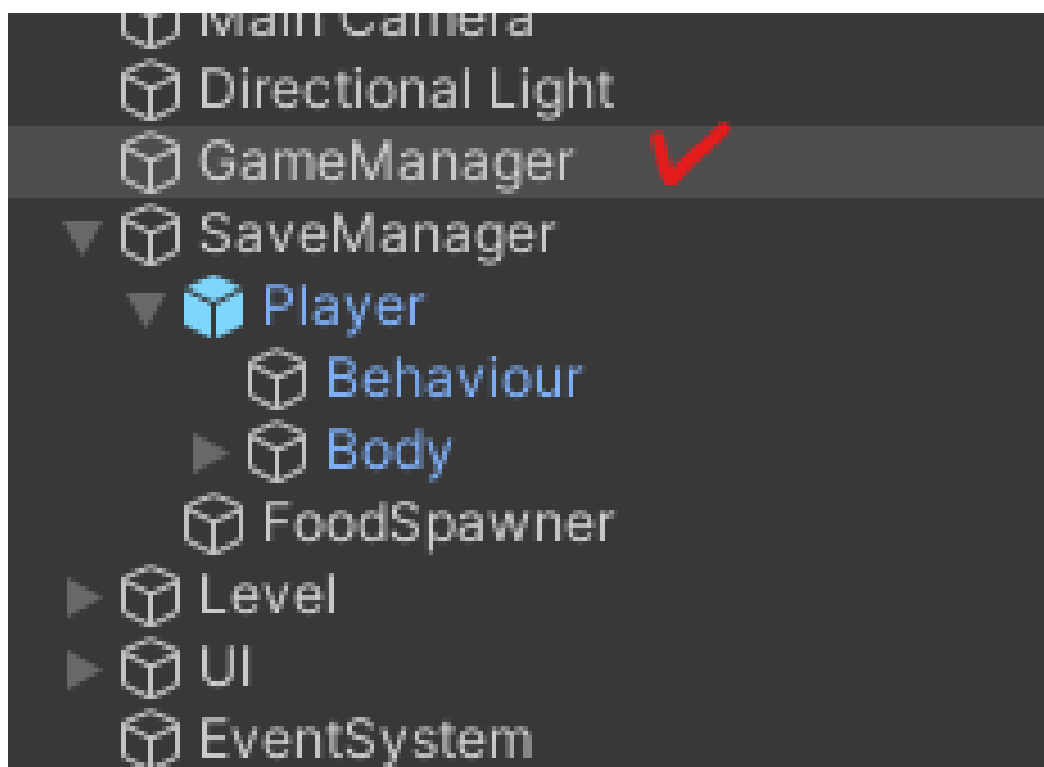
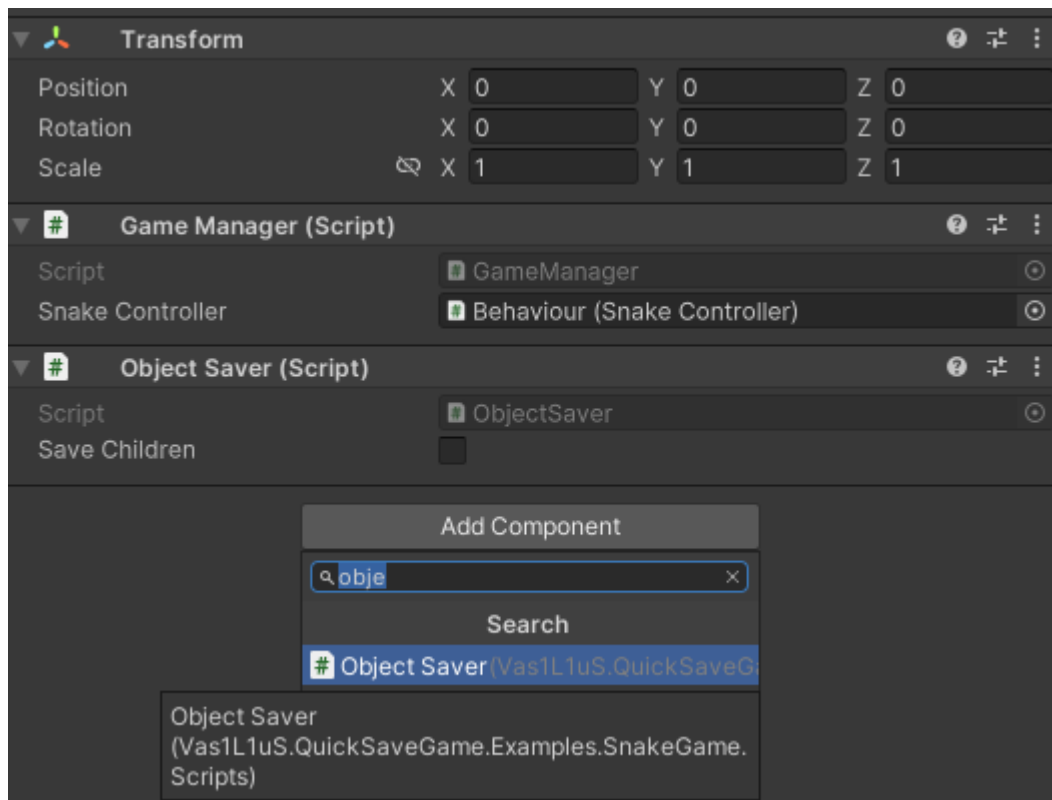
Assets - third-party objects for saving links to them



In the context menu you can call public methods of Save Manager



The objects to be saved must be in a hierarchy within the Save Manager.



If you don't want to put the object inside the Save Manager, then you should add the Object Saver component.

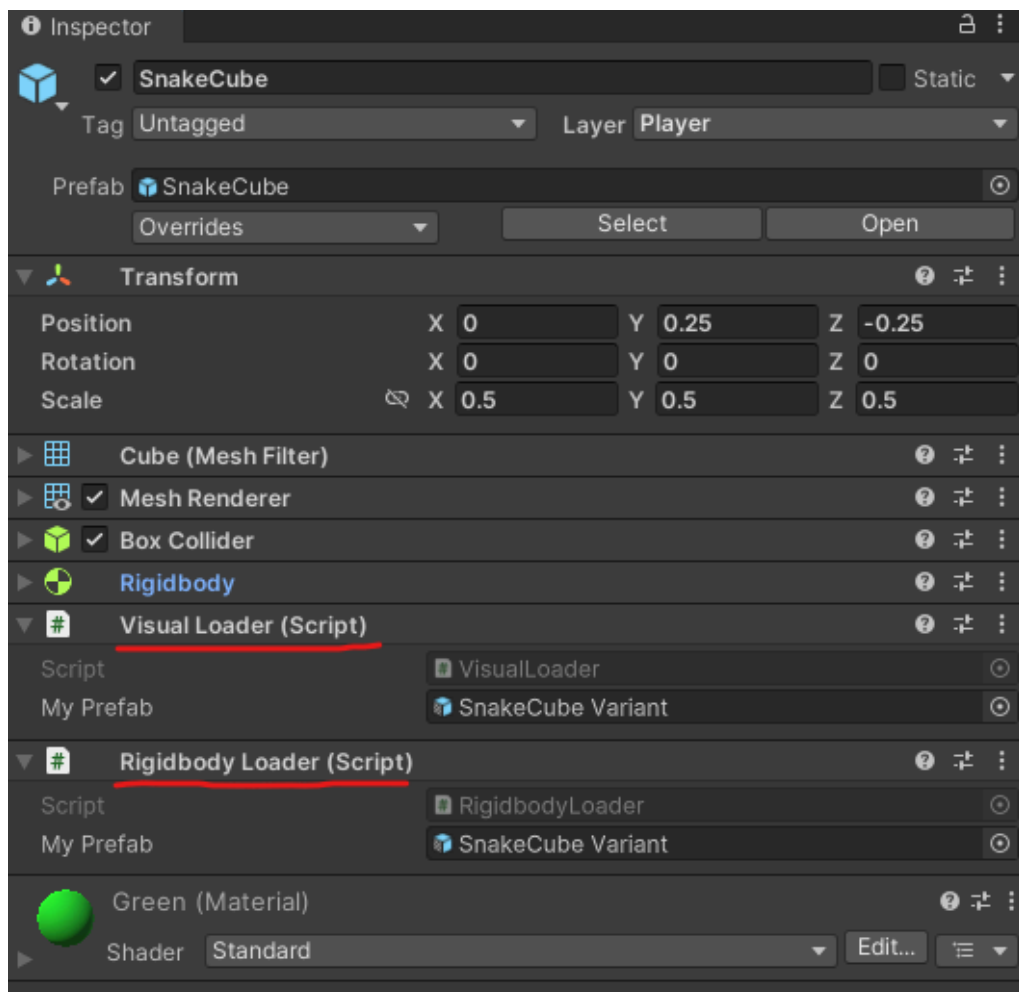
```

[SerializeField] private GameObject _foodPrefab; 🍌 Food
[SerializeField] private Picker _picker; 🔄 Changed in 1 asset
[SerializeField] private Size _mapZone; 🔄 Serializable

[HideInInspector][SerializeField] private PickableItem _currentFood;

```

The save manager saves all serializeField and public fields. You can add the HideInInspector attribute to hide the field from the inspector.



To save Unity components you must add a special component

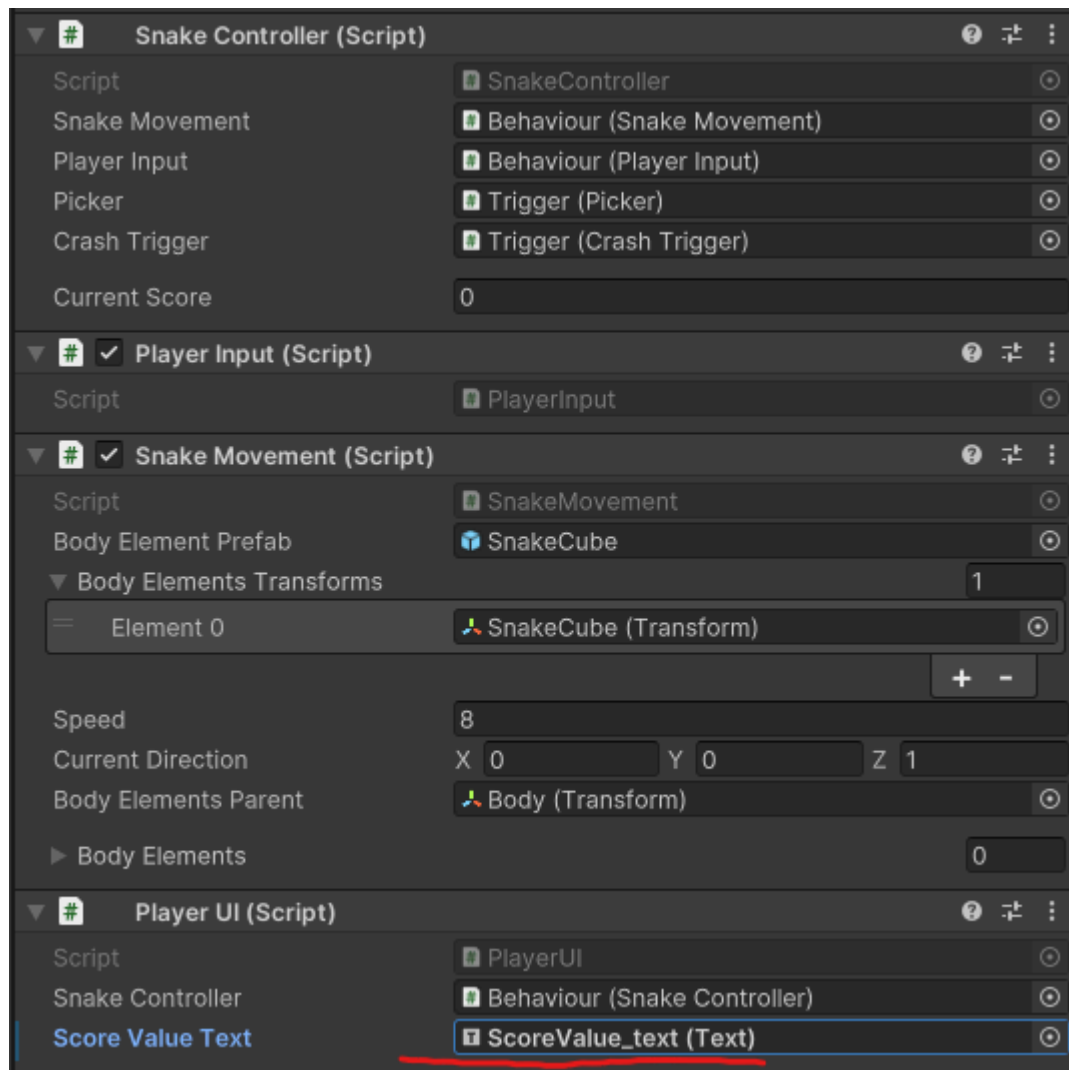
Visual Loader - to load Mesh Filter and Mesh Renderer

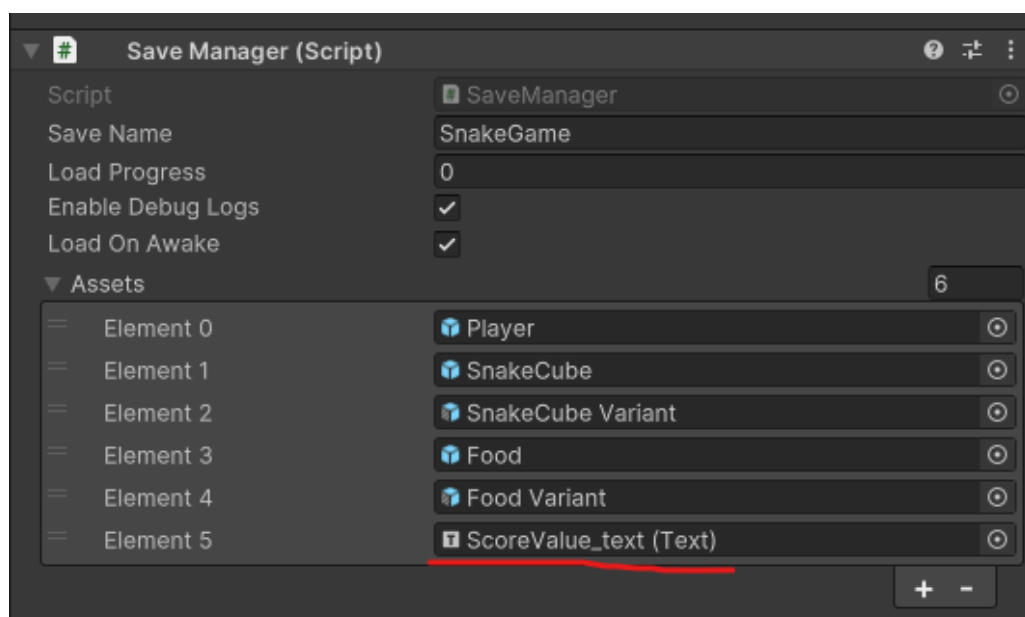
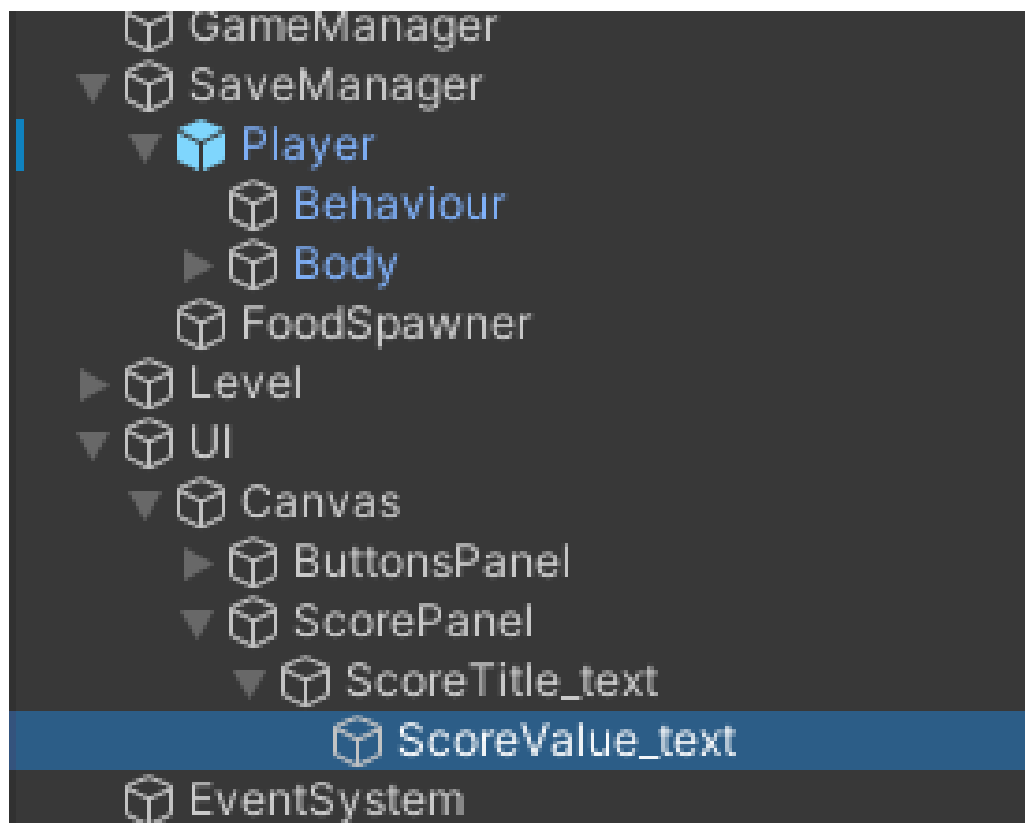
Rigidbody Loader - to load Rigidbody

Collider Loader - to load collider

In them you must place the prefab of the object you want to save. After loading, the component from the prefab will be applied to the object

Don't forget to move your prefab to Assets in Save Manager to save the reference to the object





To save a reference to an object that is not saved, you must add that object to Assets in the Save Manager.

```
private void Awake()
{
    SaveManager.LoadFinished += Init;
}
```

🔥 Frequently called 📄 2 usages 👤 Vasily Karpunin

```
private void Init(bool loaded)
{
    if (_currentFood == null)
    {
        SpawnNewFood();
    }

    SubscribeOnComponents();
}
```

You can monitor the status of Save Manager and perform necessary actions.

```
public static event Action<bool> LoadStarted;
public static event Action<bool> LoadFinished;
public static event Action SaveStarted;
```

All events Save Manager

```
float currentProgress = SaveManager.Instance.LoadProgress;
string saveName = SaveManager.Instance.SaveName;
SaveManager.Instance.Save();
SaveManager.Instance.ClearSave();
SaveManager.Instance.Load();
SaveManager.Instance.RestartScene();
```

All public properties and methods