



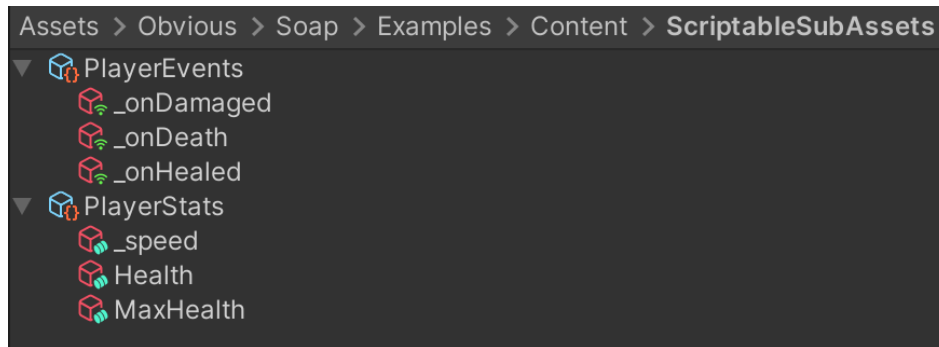
7. Scriptables SubAssets

Table of Content

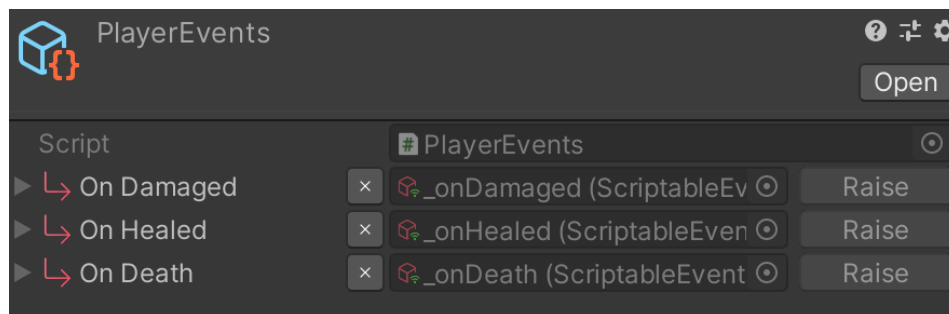
Table of Content.....	1
Set up.....	2
Direct Reference.....	3
Indirect Reference.....	4

Set up

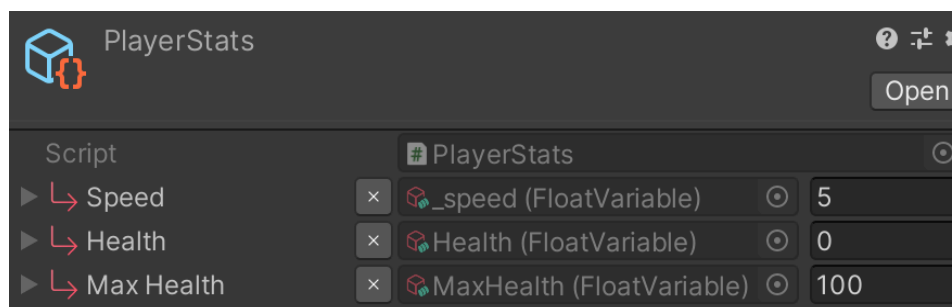
This scene is similar to the 1_ScriptableVariables_Example_Scene, with the difference that we are using sub-assets. In the project window, there are two different scriptable objects wrappers: PlayerEvents and PlayerStats.



PlayerEvents contains all the events related to the player as sub-assets (OnDamaged, OnHealed, OnDeath).

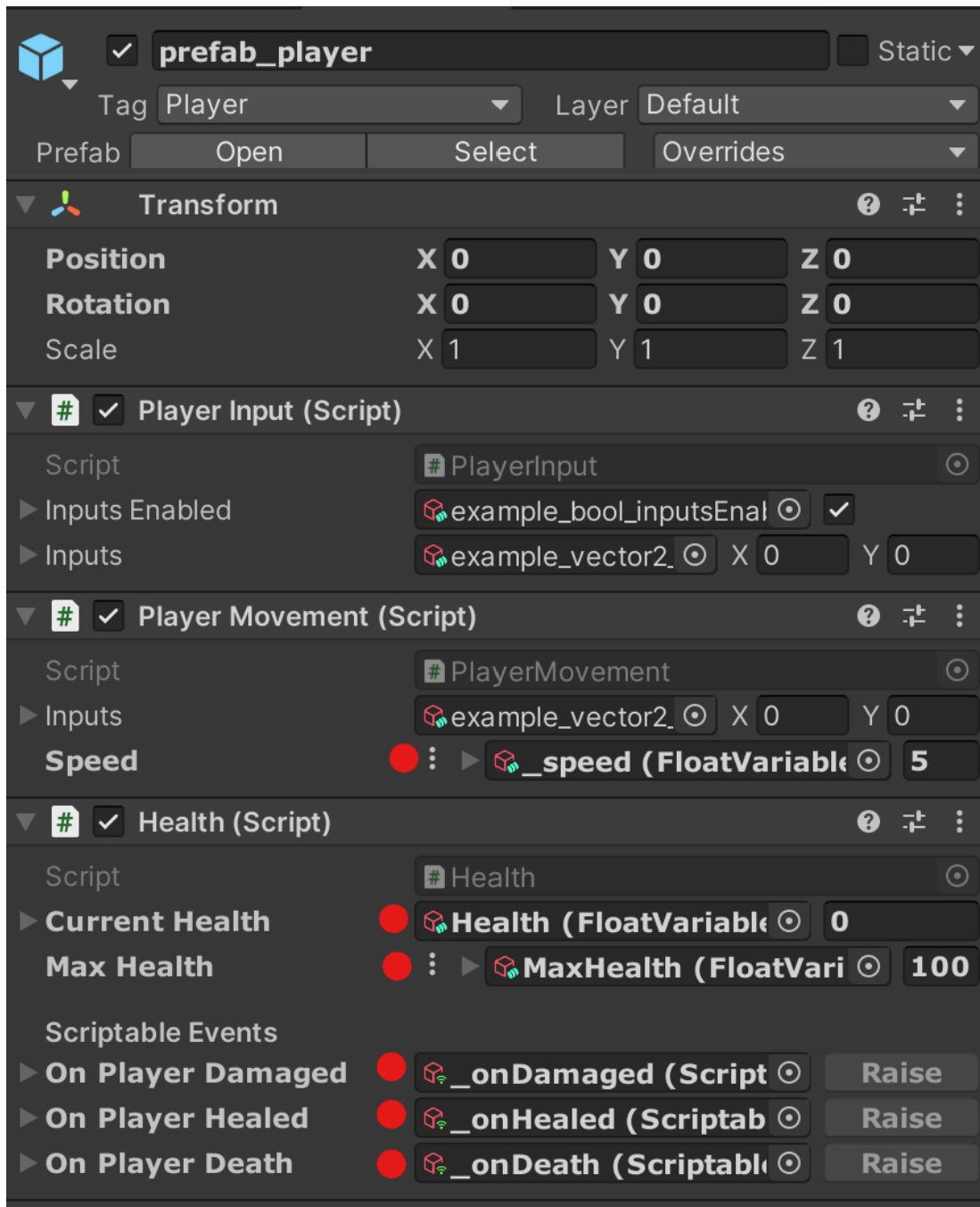


PlayerStats contains all stats related to the player as sub-assets (speed, health, max health).



Direct Reference

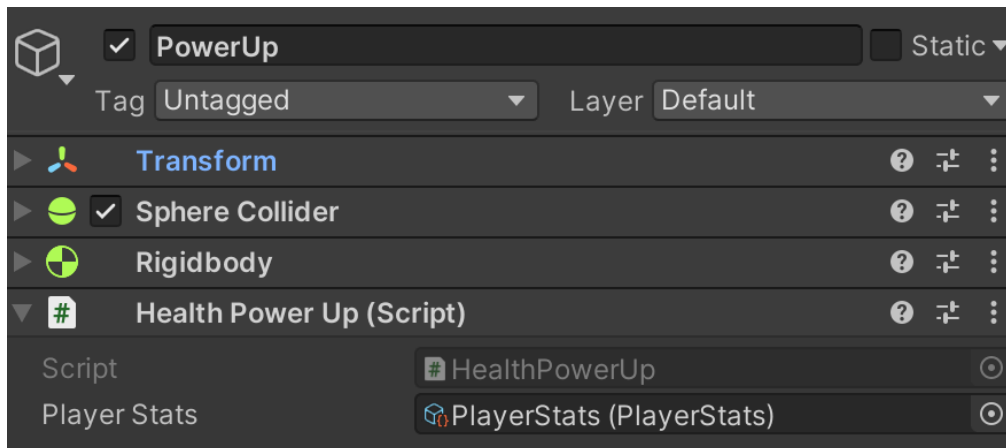
If you select the `prefab_player` in the hierarchy, you can see in the inspector that we referenced the sub assets directly in the components of the player:



Also, we referenced the sub-assets directly in the UI Canvas elements (HealthBar, vignette effects, etc...)

Indirect Reference

If you select PowerUp GameObject in the hierarchy, you can see that it has a HealthPowerUp component attached to it.



This component references the scriptable object wrapper (PlayerStats), as opposed to directly referencing the Health ScriptableVariable. If you check the code for this component, you see how to access the Health indirectly:

```
public class HealthPowerUp : MonoBehaviour
{
    [SerializeField] private PlayerStats _playerStats;

    private void OnTriggerEnter(Collider other)
    {
        _playerStats.Health.Add(30);
    }
}
```

This can be useful. For example, if you need to access multiple sub-assets of the same wrapper in a class. If you press Play and move the player into the power up, it will heal the player.