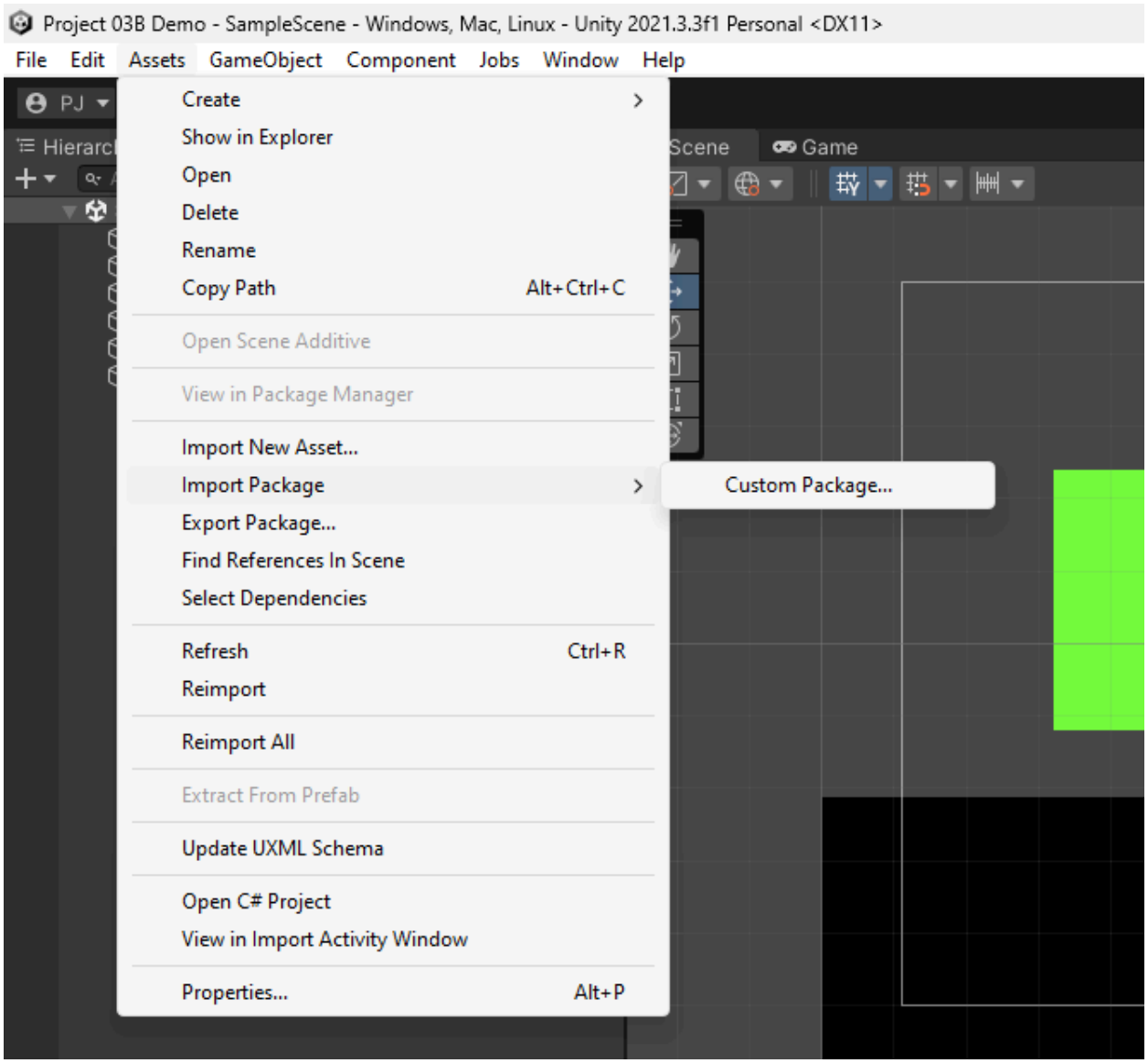
2D Movement Controller and Health Manager Setups

Summary: This package simplifies game development by providing a basic movement controller and a health manager. The latest version contains 2 scripts, MovementController and Health Manager, which can easily be modified and built upon for larger projects.

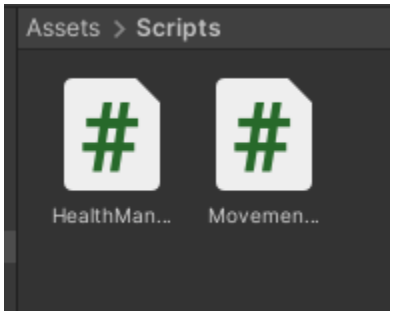
Unity Version: 2021 3.3f1

Installation: PlayerActionKit Repository

The link provided will lead to a GitHub repository containing the package, which can then be imported into a Unity project by clicking **Assets > Import Package > Custom Package**, then locating the package file in the directory downloaded.



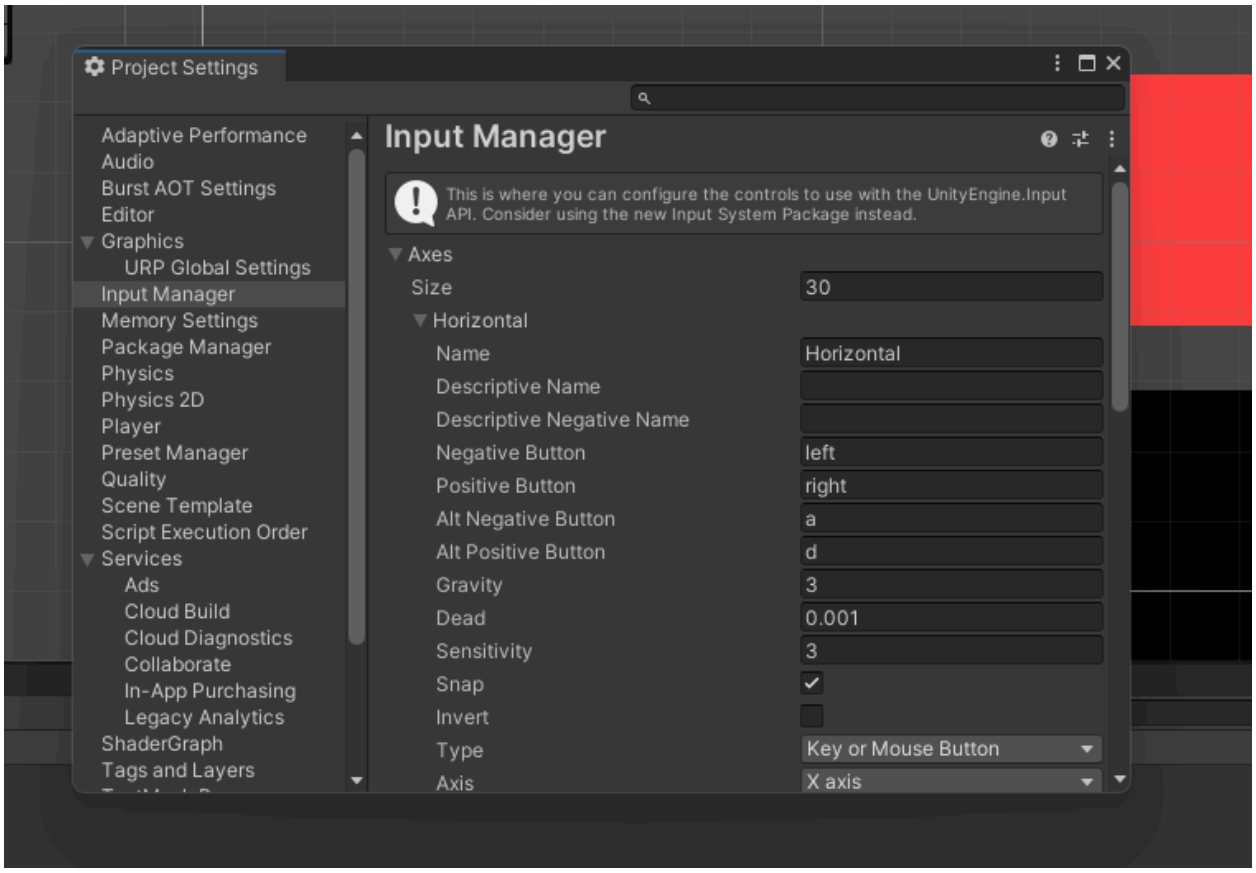
The package will import 2 scripts, both of which can easily be applied to any object in the scene by dragging the script onto the object.



Movement Controller:

The movement controller offers the most basic movements: the ability to move horizontally with the ‘A’ and ‘D’ keys and jump with the space bar.

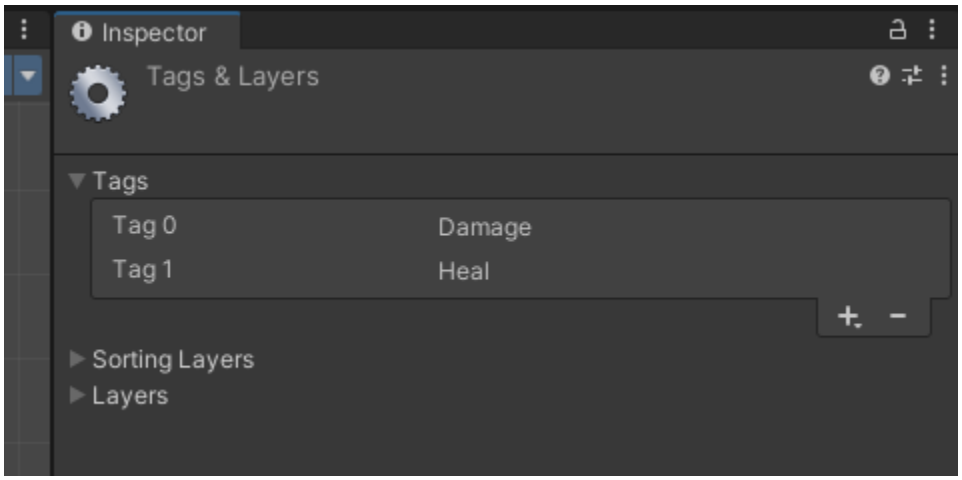
The movement controller requires both **Rigid Body 2D** and a **collider** that suits the object to function as intended, and once the script is applied, you’ll have to adjust both the **movement speed** and the **jump power**, as both values are set to 0 on default. It is recommended to adjust the values along with the **rigid body** and the **input project settings**.



Health Manager:

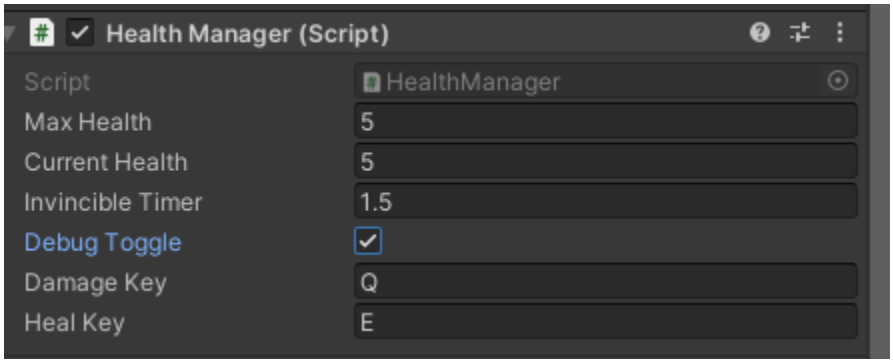
The health manager provides a simple function that allows an object to either take or heal damage. The current health can also be adjusted through the built-in debugging tool.

The health manager requires two custom tags called “**Damage**” and “**Heal**” by default, though these can be changed as needed. It will also need a non-trigger collider to detect the damage and heal objects.



The player will take damage when it comes into contact with another object with a collider that has the custom tag: **Damage** and will heal damage when it collides with an object with the tag **Heal**.

With the debug toggle enabled, the user will be able to adjust the current health by either pressing **Q(damage)** or **E(heal)** by default.



After taking damage, the object with the health manager attached will go into invincible mode, indicated by its blinking visuals, not allowing itself to take damage for a predetermined amount of time. This timer can be adjusted through the variable: **invincible timer**.

Once the object’s current health becomes zero, the default death function will trigger and delete the object. This function can easily be built upon to create a unique death function or game over screen.