

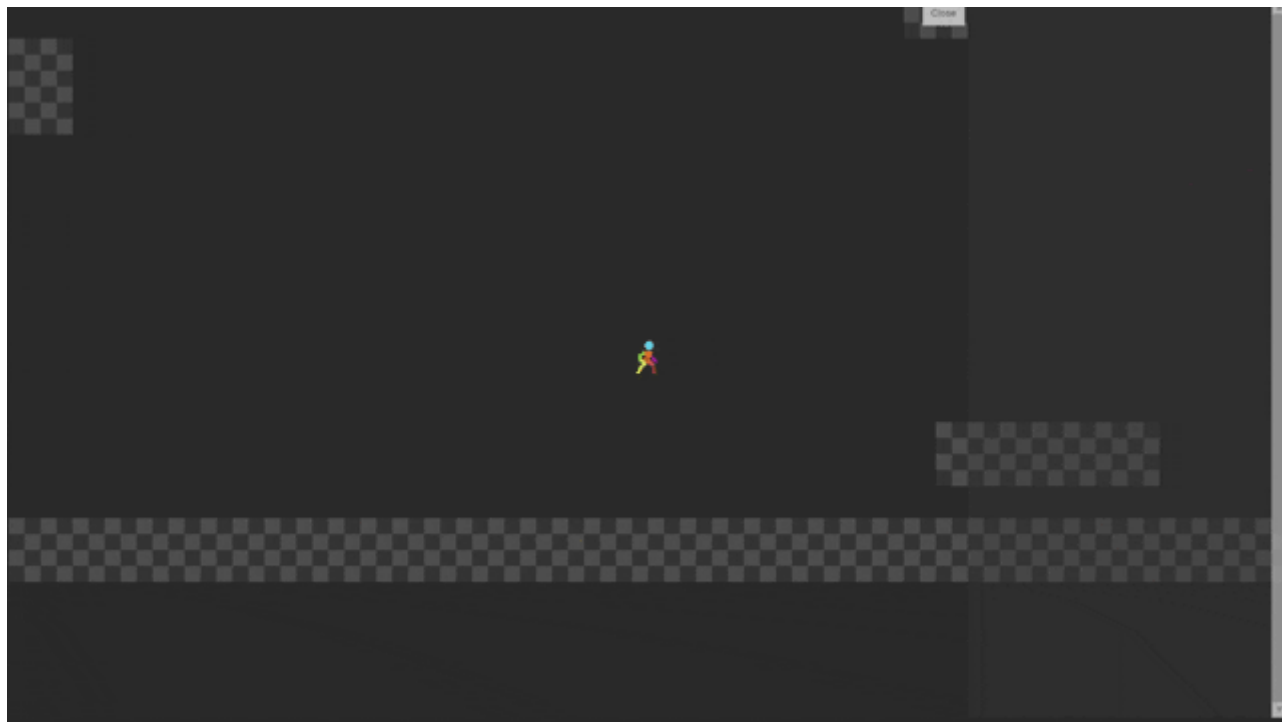
Flexible 2D Platformer Controller - Unity



About

This project is a convenient, customizable controller for your 2D platform game. It's an easily modifiable system for moving your hero. It features an extensive amount of quickly and easily configurable actions. You can use it as a standalone system, supplementing it with other necessary game systems.

[!TIP] You can try the controller in your browser - [Itch.io](https://itch.io)



Installation

[!IMPORTANT]

The controller was created on Unity version **6000.0.38f1 / 11 feb 2025**.

You can download the project and install it through UnityHub, or use the controller as a UPM package for Unity

Installation as a Unity project

1. Download the Unity project archive to your computer using **Download .zip** or **git clone**

```
git clone https://github.com/Vladislav-EG/Flexible2DCharacterControllerForUnity.git
```

2. Open the project using **Unity Hub**

Install as a package

There are two ways to install the package, via URL or from disk

[!NOTE]

When installing **from disk** you will be able to **edit files**, URL installation **does not provide this capability**.

[!TIP] Test scenes can be installed in **Package manager - _PlatformerController2D - Samples - Import**

Installation from disk

1. Download the Unity project archive to your computer using **Download .zip** or **git clone**

```
git clone https://github.com/Vladislav-EG/Flexible2DCharacterControllerForUnity.git
```

2. In Unity **Windows – Package manager - + - install package from disk – pathForDownloadProject\Assets\PlatformerController2D\package.json**

Installation from git URL

[!IMPORTANT]

You need to download not by the classic link that can be copied in Git, but by the **link with the path to package.json**.

1. In Unity – **Windows – Package manager - + - install package from git URL...**

```
https://github.com/Vladislav-EG/Flexible2DCharacterControllerForUnity.git?path=/Assets/PlatformerController2D
```

Usage

When downloading the project in the **Scene** folder or separately importing scenes from **Samples**, you can run any scene and try the controller.

[!TIP] In the **scene hierarchy** or in the **prefabs** folder you can find the character and see its organization.

Controls

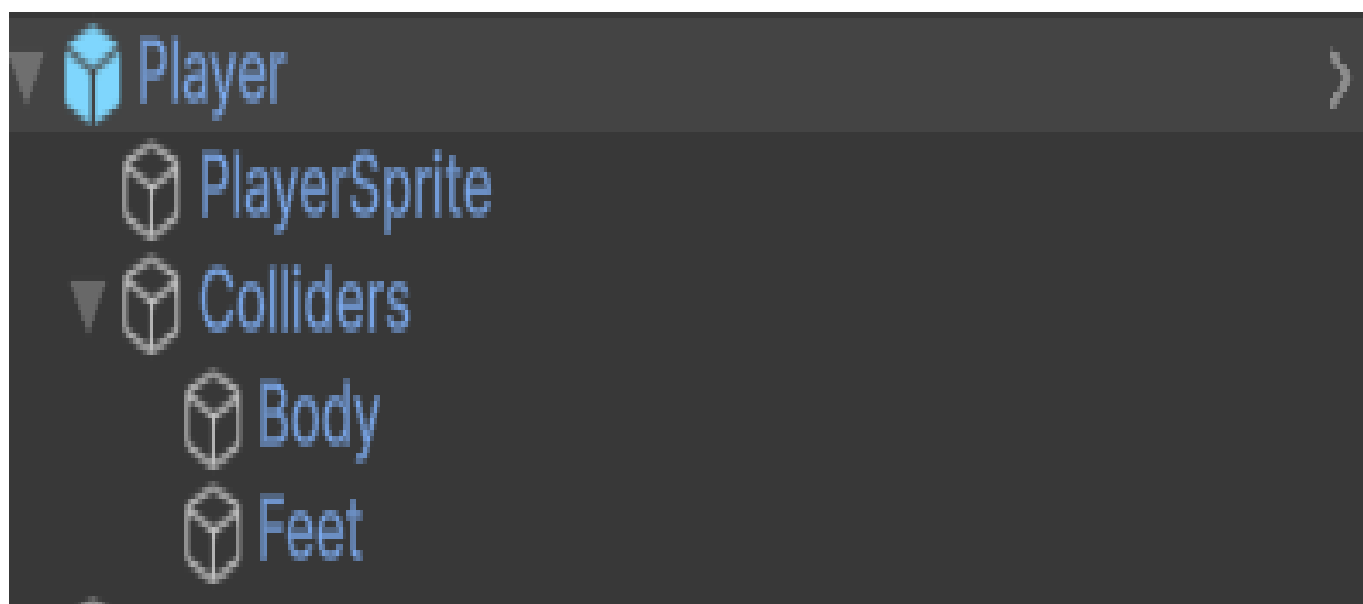


Creating and configuring your own character

1. Create the following object hierarchy with their components in Unity

- Player - (Rigidbody2D, PlayerController, CollisionChecker, Physic Handler 2D, Collider Sprite Resizer, InputReader)
 - PlayerSprite (Sprite Renderer)
 - Colliders (CreateEmpty)
 - Body (Capsule Collider 2D \ Box Collider 2D)
 - Feet (Box Collider 2D)

Example object hierarchy:



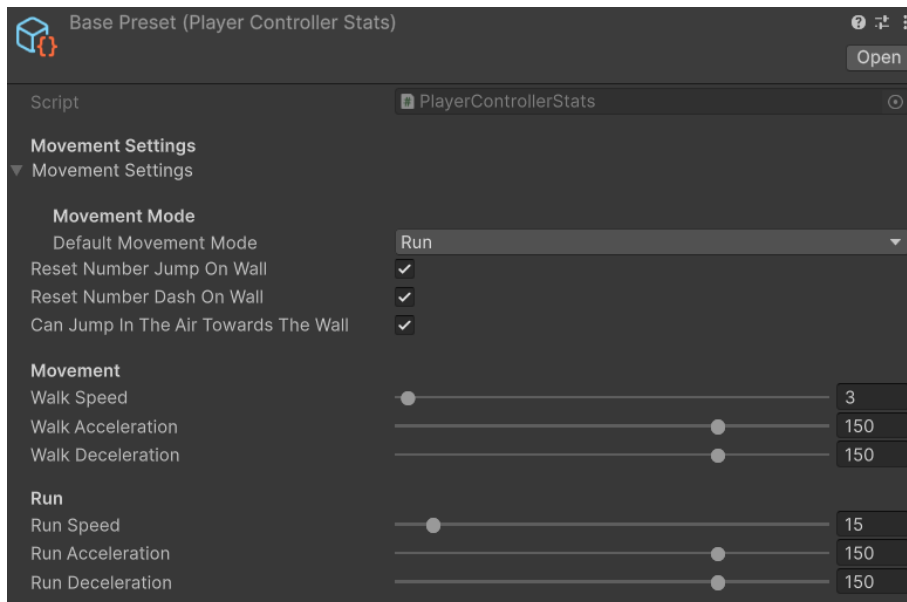
2. Create a settings preset for the character

In Unity **ProjectWindow** - **right-click** - **Create** - **PlayerControllerStats**,
Configure the created scriptable object as you wish.

[!IMPORTANT]

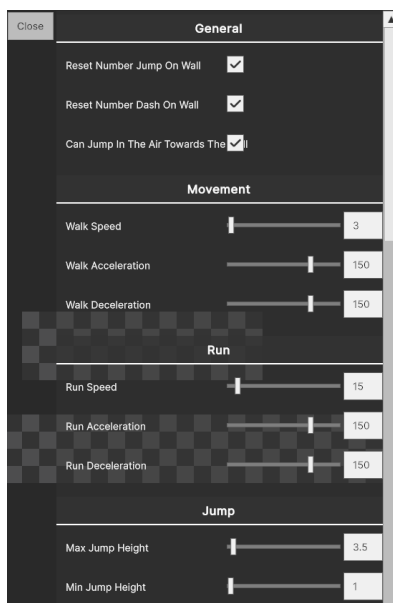
At the bottom there is a **GroundLayer** setting, **create a ground layer** and add it there, also **add this layer to all ground blocks**.

Example of part of the settings for the Player object:



[!TIP]

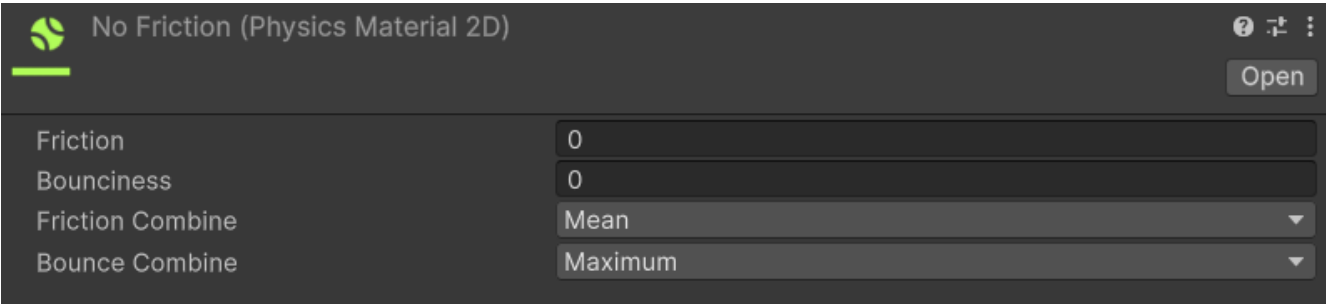
The scene has the ability to configure the preset using UI, when changing values in the UI they will also change in the SO, when disabling **PlayMode** the settings will be saved.



3. Create a physics material

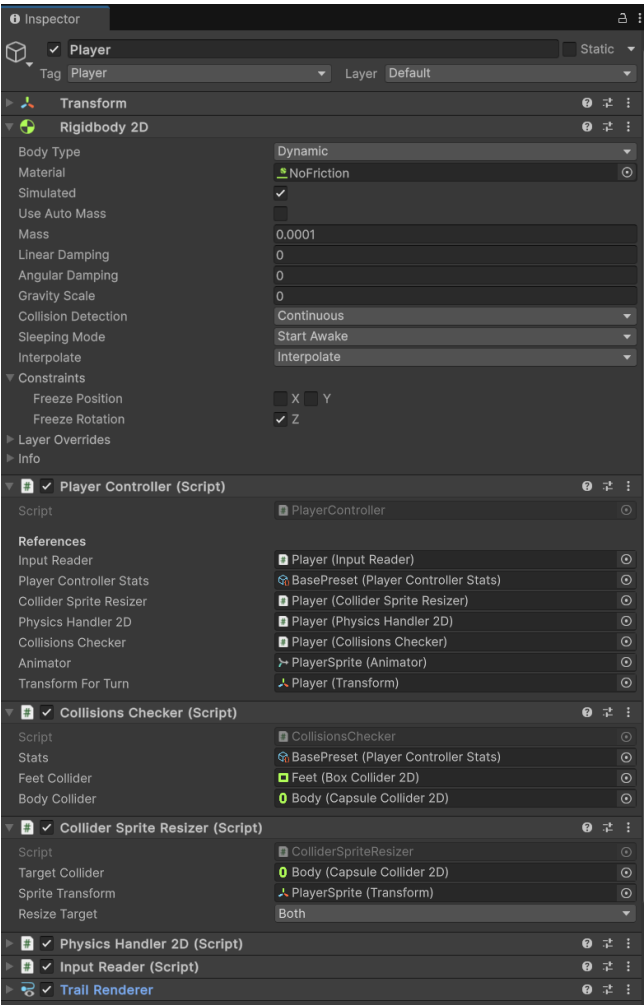
In Unity **ProjectWindow** - **right-click** - **Create** - **PhysicsMaterial**,

Example physics material for Player object:

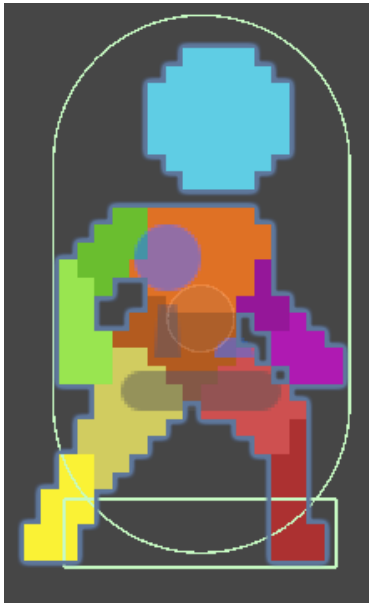


4. Configure all components using the examples below

Example Player object components:

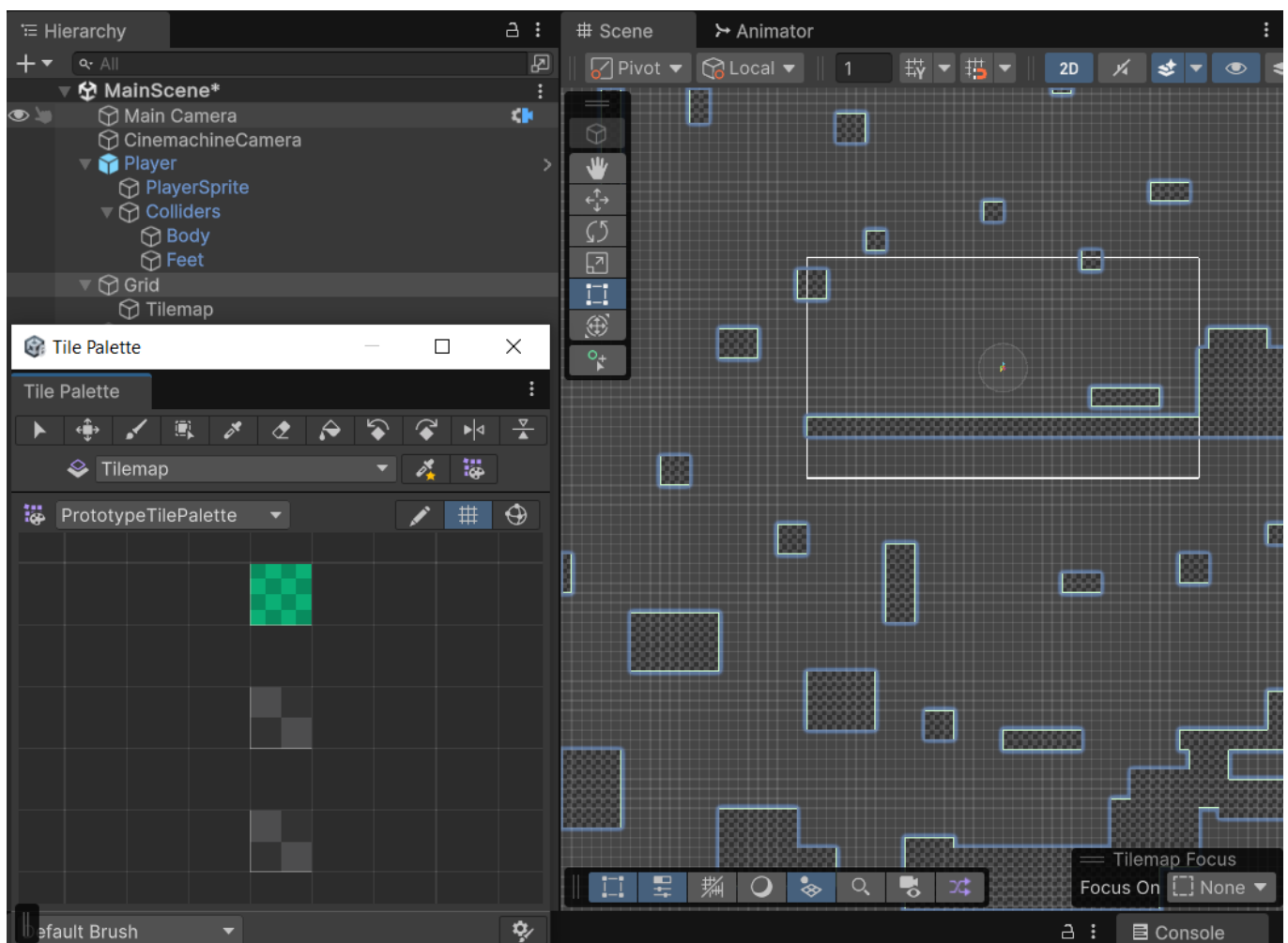


Example Player colliders:



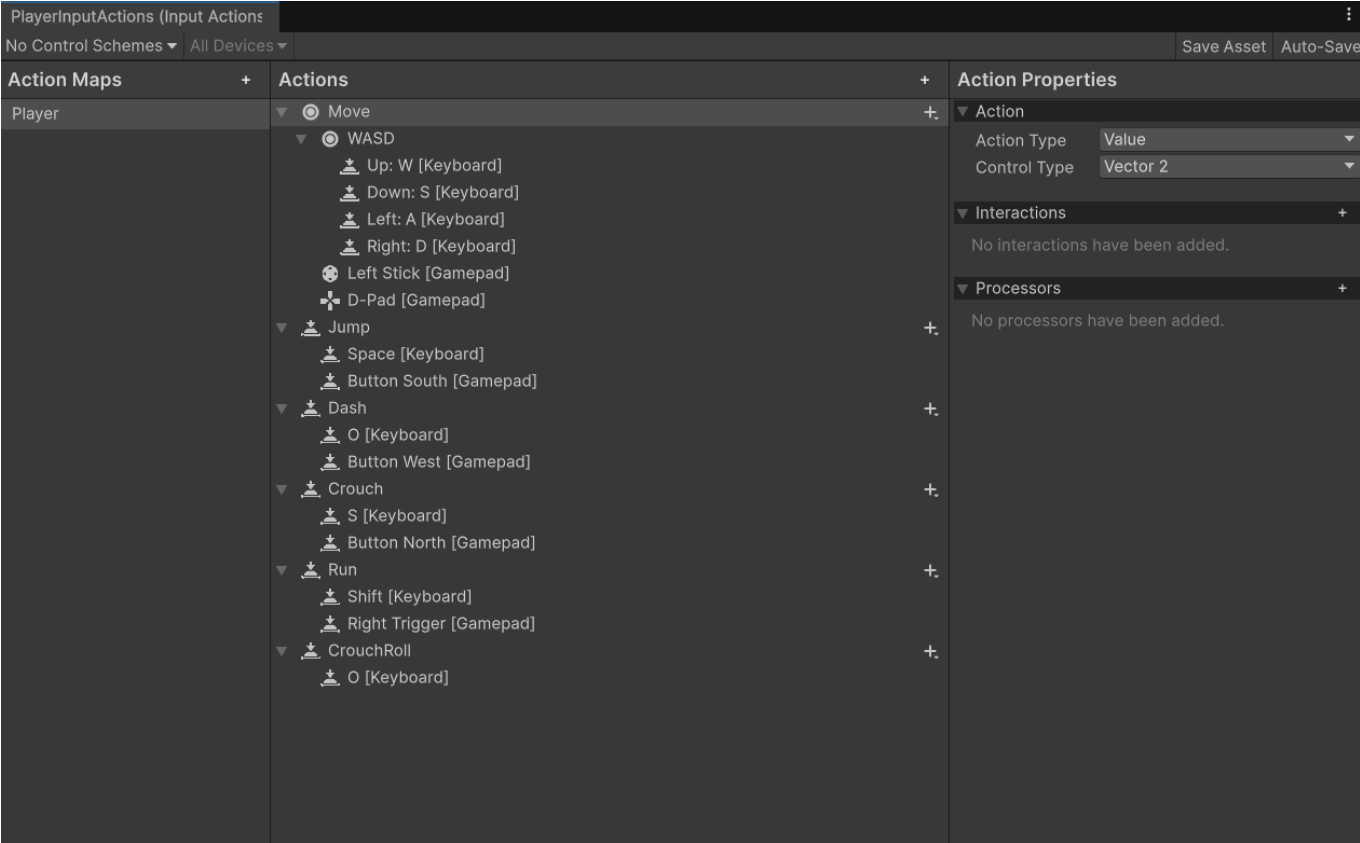
Creating a level

To create a level, drag **GridForScene** from the **Prefabs** folder to the scene, select this object and in the **Scene** window click **Open Tile Palette**, select the needed palette and create the level in the **Scene** window.



Changing key bindings

Go to the folder `Assets\PlatformerController2D\Runtime\Actions` - open the Action file `PlayerInputActions` - in the opened window `change key bindings` - click `Save Assets` on the right.



Character actions and their parameters

All character capabilities and preset configuration parameters can be viewed here - [Character Actions and Preset Configuration Setup](#)

Feedback and Contributions

Enormous effort was put into developing this project, I understand that the architecture, logic, organization in some moments falls short and requires improvement. However, the development process is not yet finished and your help is important for the project.

[!IMPORTANT]

If you have feedback, suggestions for improvement, or you found bugs, please write about it. I would also like to address experienced developers, it would be very nice to hear your feedback about the project architecture, its organization and code. This will help me become better and understand many concepts, I appreciate it!

License

MIT License. Refer to the [LICENSE.md](#) file.

Copyright (c) 2025 Vladislav-EG [Vladislav-EG](#)