2D Platforms Pack

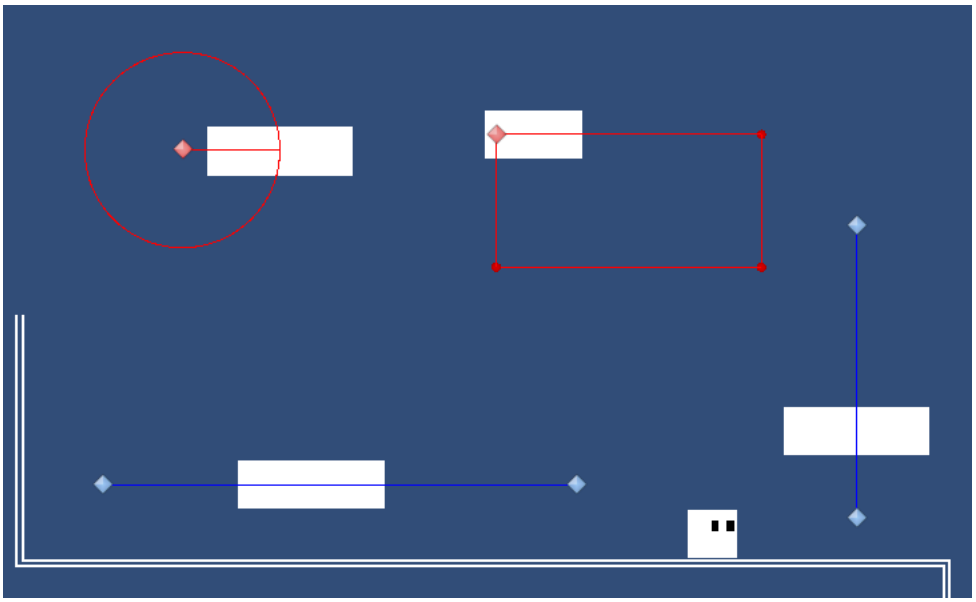
User Guide

By Sicong Nie

Inhoud

Introduction

This asset is a package of pre-made platforms, specifically designed for developers of 2D platform games using Unity. It offers a variety of platform types commonly found in 2D platformers, enabling users to add these platforms to the game and design complex levels.



The purpose of this asset is to assist developers by allowing them to efficiently add or modify the platforms in their 2D platformer without having to spend time creating the platforms themselves.

Package structure

Installing the package is simple; you can import it using the package manager or download it from the website and import it into your project. But before importing the package, you need to first verify the names of folders and scripts to prevent conflicts with existing ones in your project:

Editor folder

This folder houses editor scripts responsible for specific functionalities. It includes the following scripts:

- `FallingPlatformEditor.cs`
- `MovingPlatformPatrolEditor.cs`

Prefabs folder

All pre-made platform prefabs are stored in this folder:

Prefabs of falling platform

- Falling block
- Falling platform
- Falling platform_List

Prefabs of moving platform

- Moving platform_Box
- Moving platform_Circular
- Moving platform_Horizontal
- Moving platform_Patrol
- Moving platform_Vertical

Prefabs of other platforms

- ConveyorBelt platform
- Jumping platform
- Switch platform
- Temporary platform

Prefab of the player

- Player

Scripts folder

Contains all game scripts categorized by platform type:

Scripts of falling platform

- `FallingPlatform.cs`
- `FallingPlatformGenerator.cs`
- `FallingPlatformRespawn.cs`

Scripts of moving platform

- `MovingPlatform_Vertical.cs`
- `MovingPlatform_Patrol.cs`
- `MovingPlatform_Horizontal.cs`
- `MovingPlatform_Circular.cs`
- `MovingPlatform_Box.cs`

Scripts of other platforms

- `ConveyorBelt.cs`
- `JumpingPlatform.cs`
- `SwitchPlatform.cs`
- `TemporaryPlatform.cs`

Script of player

- `PlayerMovement.cs`

How each type platform works

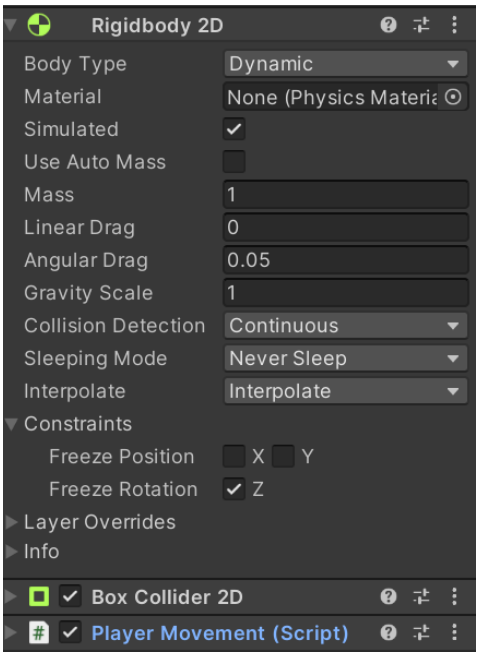
Player Movement

Before delving into the usage of each prefab, it's crucial to verify the mechanics of player movement in your game.

Through my research, it became evident that the manner of player movement in 2D platforms significantly impacts platform mechanics. There are four scripting methods commonly used for player movement in 2D platformers:

1. transform.Translate
2. Rigidbody2D.velocity
3. Rigidbody2D.MovePosition
4. Rigidbody2D.AddForce

This package contains a prefabs of controllable character. The main function of it is to test the platforms. When programmer player movement for this prefab, I've chosen to use **Rigidbody2D.velocity**.

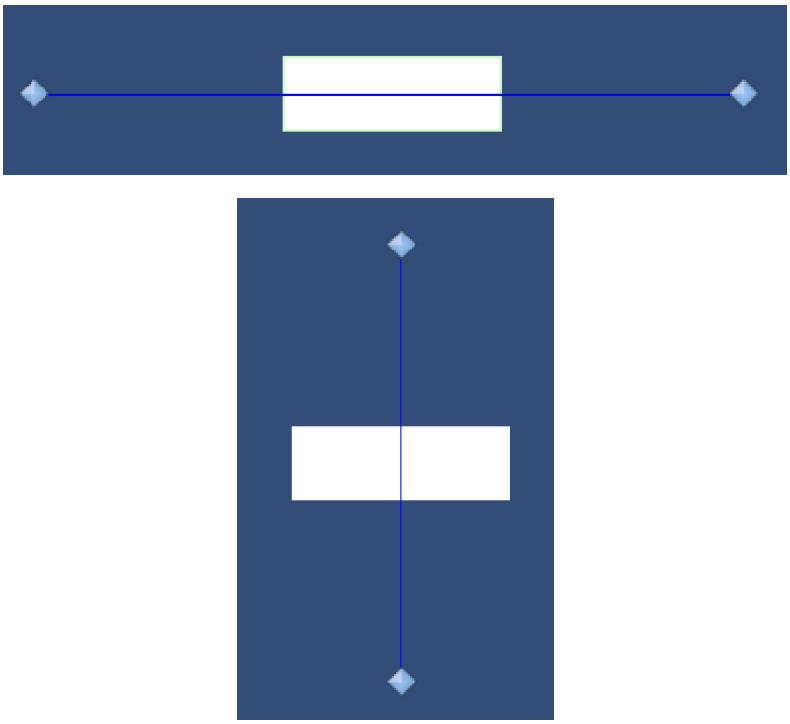


So it means that all prefabs in this package are designed specifically for **Rigidbody2D.velocity**. Therefore, ensure that the main character in your project has a rigibody2D with **Dynamic** bodyType and uses **Rigidbody2D.velocity** for the movement. Otherwise the prefabs of the platforms would not work correctly!

Moving Platforms

There multiple types of moving platforms in this package.

Moving Platform Horizontal and Moving Platform Vertical



These prefabs represent two fundamental types of moving platforms commonly found in 2D platformers. Their function is to move horizontally or vertically. While they may appear similar, there are slight differences in the scripts that distinguish them from each other.

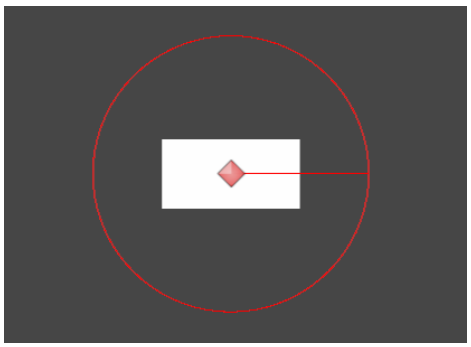
Using the values in the scripts associated with each prefab, you can adjust the following settings:

- **Edges:** Adjust the edges of these platforms to determine the moving length.
- **Move Speed:** Change the speed of the platform (default speed is set to 2f).
- **Start From Right (Horizontal):** If set to true, the platform will move right initially when the game starts; otherwise, it will move left.
- **Start From Down (Vertical):** If set to true, the platform will move downward initially when the game starts; otherwise, it will move upward.
- **Show Line:** When true, a path of the moving platform will be visible; otherwise, it will be hidden.
- **Wait Mode:** When enabled, the platform will pause briefly upon reaching the edges; otherwise, it will immediately move to the other edge.
- **Wait Time:** Set the duration for which the platform will stay at the edges when the wait mode is activated.

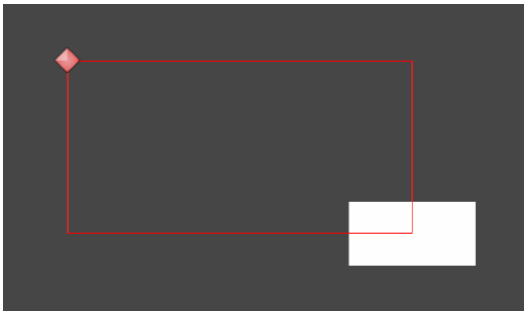
Moving Platform Box and Moving Platform Circular

Both of these moving platforms follow a stable shape as the path during movement.

For the circular platform, you can modify the circle's size using the 'Rotation Radius' value. Increasing the Rotation Radius results in a larger circle for the platform's movement.



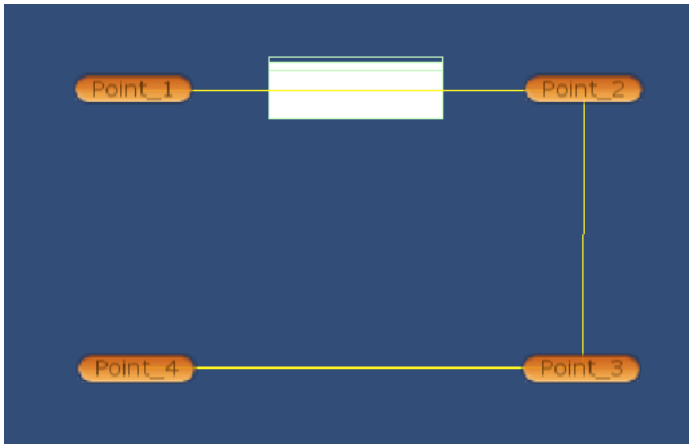
For the box platform, you can adjust its shape simply by relocating the Center's position (identified by the red icon).



Both prefabs feature an option within the script values known as 'isClockwise.' When set to true, the platforms will move clockwise at the start of the game; otherwise, they will move counterclockwise.

Moving Platform Patrol

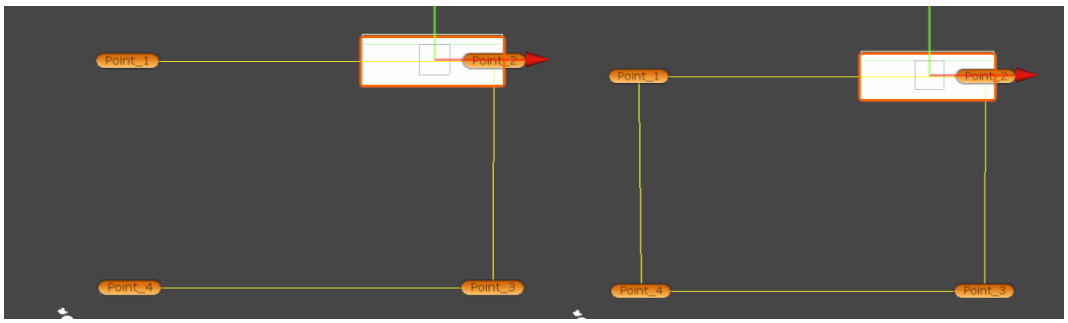
This prefab can move in a customable patrol path.



In its scripts consist the following values:

- **Patrol Points:** This platform operates along a set of designated patrol points. These points define the path and movement sequence for the platform.
- **IsClosed:** Determines the behavior of the platform at the end of its patrol route. When set to true, the platform will move from the last point to the first point, creating a continuous loop.

If set to false, it will return to the previous point, effectively moving back and forth along the patrol route.



There are also two editor buttons:

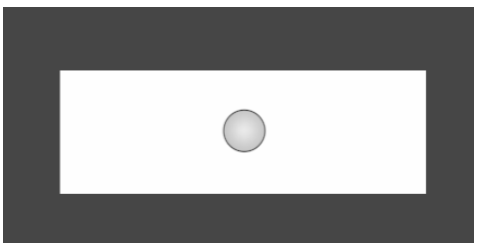
- **Add Point:** This button facilitates the addition of new points to the platform's patrol route. Clicking it will generate a new point, seamlessly integrating it into the existing patrol point list. So when you want to more patrol points, you dont need to manually create new points and add them to the point list. Simply utilize the 'Add Point' button within the editor, and a new point will be generated and automatically added to the point list.
- **Delete Point:** Removes a selected point from the patrol route, offering flexibility in editing the platform's path.

Falling Platforms

There are two types falling platforms in the package.

Falling Platform: Basic

This is a prefab that contains basic functions of the falling platform.



It looks different from an ordinary platform, but has a unique function: when the player steps on it, the platform starts a countdown and falls when the countdown ends. Using values in the prefab association script, you can adjust the following settings:

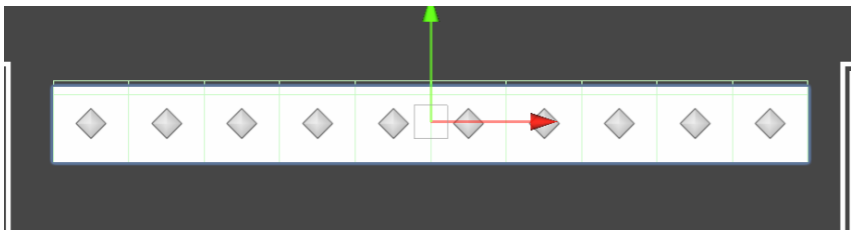
- **Fall Delay:** Determines the countdown before the platform falls.
- **Destroy Delay:** Sets the duration before the platform is destroyed after falling.

Moreover, this platform features a respawn system. Once destroyed, it will respawn after a specified timer at the position of the grey point. Using the script's values linked to the prefab, you can adjust these settings:

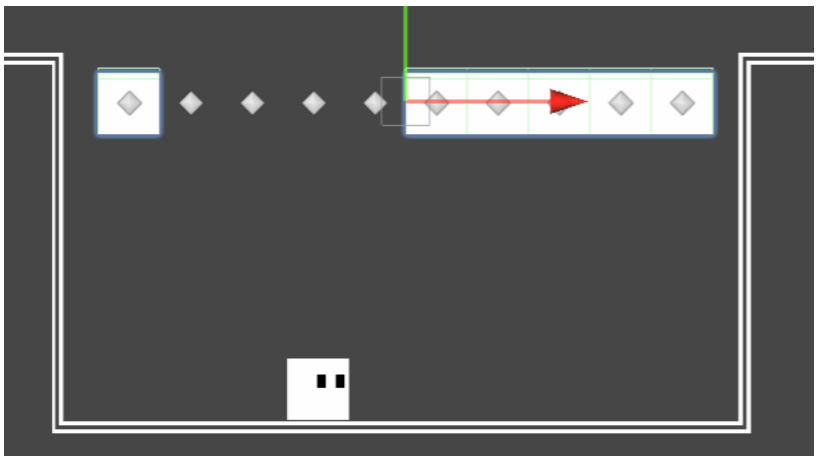
- **Respawn Point:** Defines the respawn position of the platform.
- **Block:** Specifies the default prefab for respawn; you can change it to your preferred object.
- **Respawn Delay:** Sets the time delay before the platform respawns after being destroyed.

Falling Platform: List

Similar to the basic Falling Platform, this prefab serves the same function but within a collective list.



Just like the falling blocks in the Super Mario Bros series, this type of falling platform behaves uniquely: they fall piece by piece. Each block within this prefab functions identically to the basic falling platform—falling, destroying, and respawning. However, the distinguishing feature of this prefab is that each block operates independently.



To achieve this functionality, an additional parent object is included to generate these blocks using the Falling Platform Generator script. By adjusting the settings within the script linked to the prefab, you can customize the settings of these falling blocks:

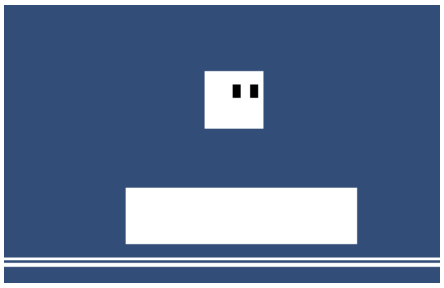
- **Block Prefab:** Specifies the prefab for individual falling blocks.
- **Block Count:** Determines the number of blocks within the list.

Additionally, an editor button is available:

- **Generate Platform:** Initiates the generation of the block list with the specified settings, the list will form a platform.

Jumping Platform

This prefab is designed to enhance the player's jump force. When the player is on this platform, they will automatically bounce when he land on it, until they leave the platform. And when the player lands on this platform and presses the jump button, they will experience a higher jump.



It offers a single customizable value: **Bounce**. Adjusting this value determines the jump force applied to the player when on this platform. A higher value results in a greater jump height for the player.

Conveyor Platform

this platform functions as a conveyor belt, automatically moving players horizontally in a specific direction at a constant speed. Using the values in the associated prefab script, you can adjust its movement direction with the **'isClockwise'** value and change the platform's speed using the **'MoveSpeed'** value.

Temporary Platform

This prefab introduces a time-limited function. It remains available for a limited duration, after which it either disappears or reappear.

Using values in the prefab association script, you can adjust the following settings:

- **Disappear Duration:** Determines how long the platform remains visible before disappearing.
- **Reappear Duration:** Specifies the duration until the platform reappears after disappearing.

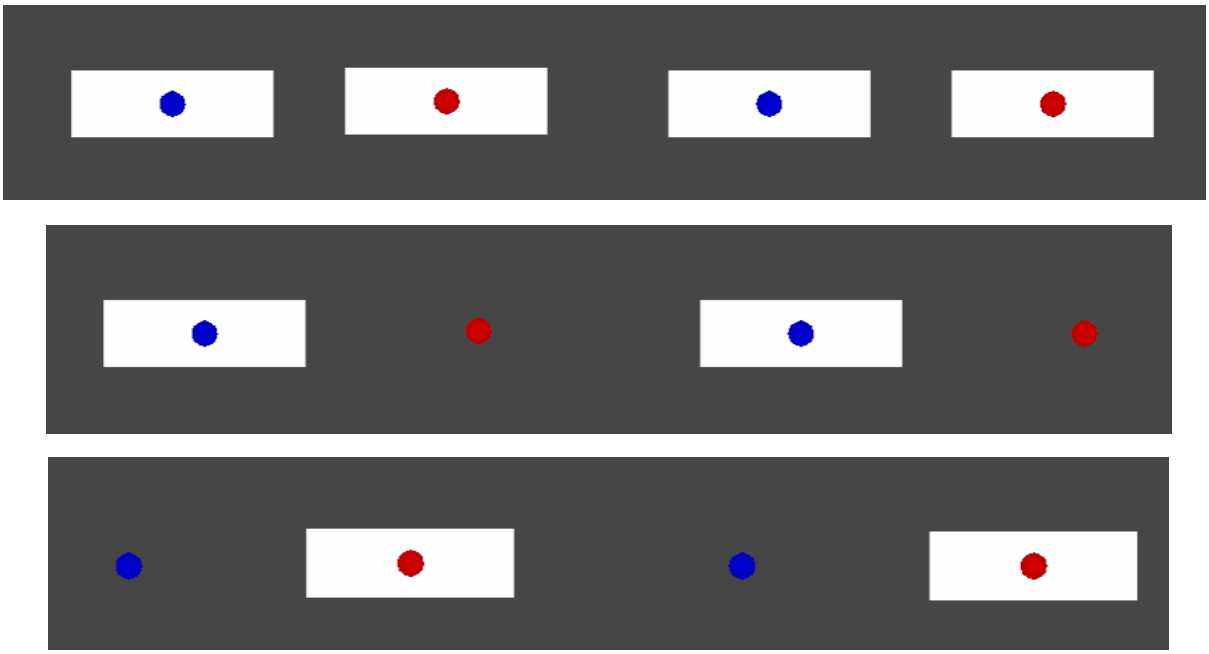
Additionally, this platform features a blink system. As the disappearance duration approaches, the platform begins to blink, serving as a reminder to the player. The following values control this function:

- **Blink Interval:** Sets the time interval between blinks.
- **Begin To Blink:** Determines the time at which the blinking commences.

- **Show Timer:** When enabled, displays the timer on the platform; otherwise, it remains hidden.

Switch Platform

This prefab features two groups of platforms, allowing only one group to be active within the game at any given time. When group 1 is active, the platforms in group 2 become inactive and invisible, and vice versa. The switch between groups is governed by the timers.



Using values in the prefab association script, you can adjust the following settings:

- **Group 1 Timer**: Sets the duration for which group 1 platforms remain active.
- **Group 2 Timer**: Specifies the duration for which group 2 platforms remain active.
- **Show Points:** Displays points for both groups in Gizmos, aiding in distinguishing the platforms of each group.
- **Group 1 first:** Activates the first group initially if set to true; otherwise, activates the second group.
- **Show Timer:** When enabled, displays the timer on the platform; otherwise, it remains hidden.

Additionally, the prefab comes with two platforms in each group by default. You can easily add more platforms by duplicating any of the existing ones.