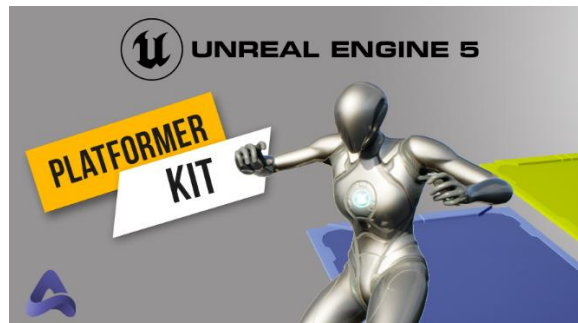


ADS Platformer Kit



Documentation

Version of 30 mai 2024

Table of contents

Foreword	2
Platforms	2
Introduction	2
Moving platform	3
Spline platform	4
Rotating platform	5
Falling platform	6
Alternate platforms	7
Bumping platform	8
Jump	9
Contact	10

Foreword

This pack is a Platformer 3D Platformer Kit. Please note that this is not a template, but simply a set of platforms and an improved jump.

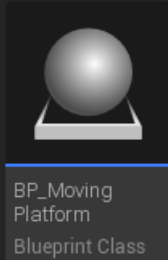
If you don't like the pack or find a problem, please contact me by e-mail (azure.dream.studio@gmail.com) instead of giving a bad rating on the marketplace. If there really is a problem, I'll update the pack quickly.

The platforms

Introduction

Each platform is an Actor Blueprint that can be deposited on the Level, where each instance can be configured via its details panel.

Moving platform





Moves the platform from one location to another.

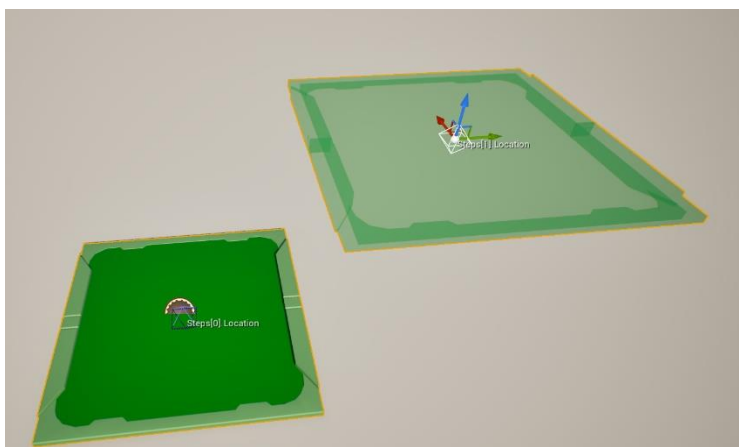
Can move in several steps, each with its own location, speed and pause.

Each step is visualized in the editor and can be positioned using the Gizmo

Instance parameters

Platform Mesh	 SM_Platform	Assigning a Static Mesh to the platform
Platform Materials	2 Array elements	Assigning materials to the platform
Platform Indicator Materials	 M_PlatformIndicator	Assign a material to view the platforms at each step.
Steps	2 Array elements	Allows you to add steps
Index [0]	3 members	Each index is a step.
DelayBeforeMove	2,0	DelayBeforeMove: Wait X seconds before first move
Location	0,0 0,0 0,0	Location: Platform location for this step
Duration	1,0	Duration: Travel time in seconds to reach next location

Use the Gizmo to position a step



Once a step has been added, you can, if you wish, select it on the level and position it directly with the Gizmo displacement tool.


Spline platform

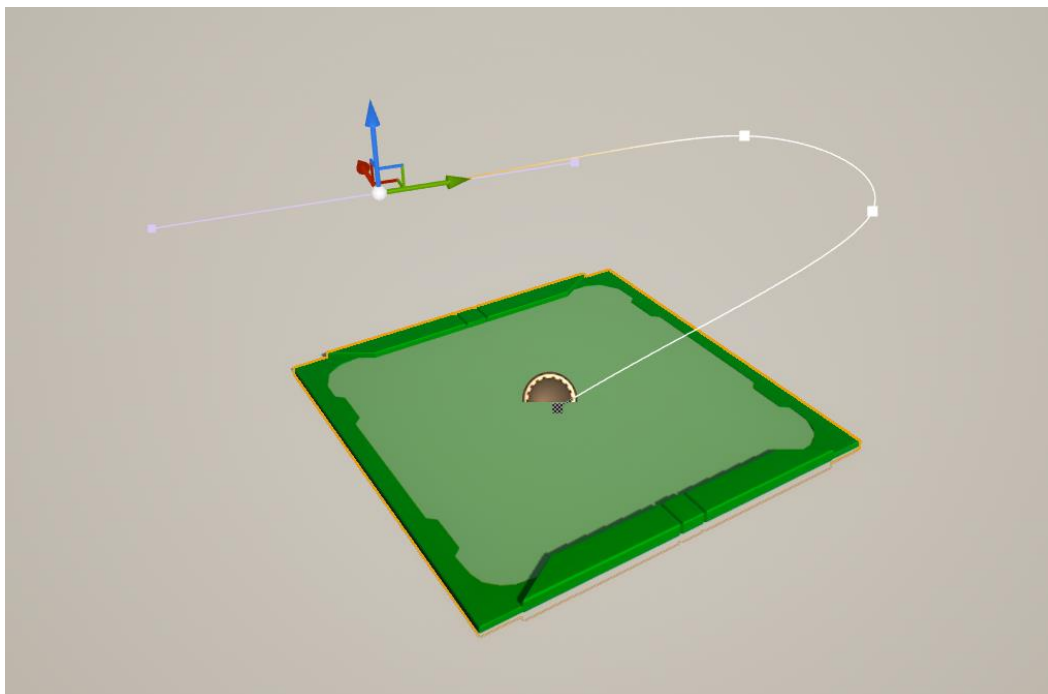


Moves the platform along a Spline. When the platform reaches the end, it moves in the opposite direction.

The Spline is included in the Blueprint. There is therefore no need to place a Spline Actor and assign it

Instance parameters

Platform Mesh	 SM_Platform	Assigning a Static Mesh to the platform
Platform Materials	2 Array elements	Assign materials to the platform
Duration	10,0	Time (in seconds) for the platform to reach the end of the Spline
Delay Before Start	5,0	Time (in seconds) to start movement
Delay Before Reverse	5,0	Time (in seconds) to wait for the platform to make the reverse trip




Rotating platform



Pivoting platform.


The player's camera does not rotate with the player when the player is on the platform

Instance parameters

Mesh	 SM_Platform	Assigning a Static Mesh to the platform
Materials	2 Array elements	Assign materials to the platform
Pause Before Start	0,0	Time (in seconds) to wait before initiating the first rotation
Ping Pong Mode	<input type="checkbox"/>	Activates or deactivates Ping Pong mode. True: once a rotation has been completed, the next rotation will be in the opposite direction. False: all rotations will be in the same direction.
Target Rotation	0,0 0,0 360,0	Define the rotation of the platform
Pause Before Restart	0,0	Time (in seconds) to wait before performing a new rotation once a rotation has been completed.
Speed	50,0	Degrees per second



Falling platform



Platform falls and disappears.

3 possible triggers

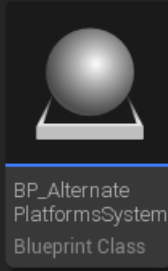
BP_Falling Platform
Blueprint Class

Instance parameters


Falling Type	Leave	Set trigger mode. Leave: Player leaves the platform Delay: X seconds after the player is on the platform Leave + Delay: The combination of the other two
Mesh	SM_Platform	Assigning a Static Mesh to the platform
Materials	2 Array elements	Assign materials to the platform
Delay Before Fall when Delay	0,2	Time (in seconds) to wait before the platform falls. Taken into account if Falling Type is different from Leave.
Delay Before Fall when Leave	0,2	Time (in seconds) to wait before the platform falls. Taken into account if Falling Type is different from Delay.
Delay Before Reappear	5,0	Time (in seconds) to wait before the platform reappears after a fall. 0 = Never reappears.



Alternate platforms




The system to place allows you to manage different groups of platforms, alternating between visible and invisible. A platform's collision will only be active when it is visible.

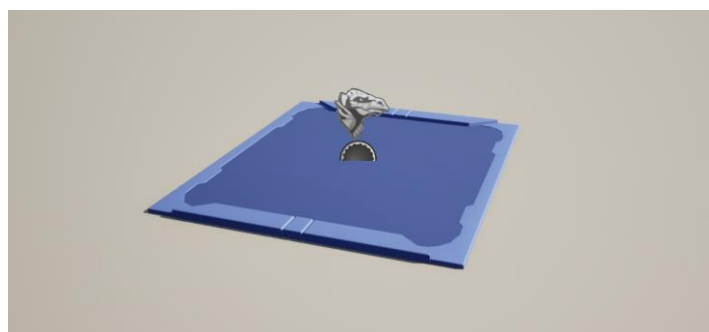


Alternating individual platforms. Do not place directly in the level. The system takes care of placing and configuring them.

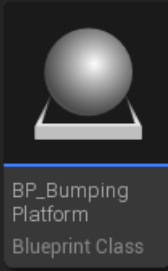
Do not touch.

Instance parameters

Groups	1 Array element	+ -	Add platform groups
bStartVisible	<input checked="" type="checkbox"/>		Defines if the platforms in this group are visible at startup or not
Mesh	 SM_Platform	↶ ↷	Defines the Static Mesh to be used for the platforms in this group
Materials	0 Array element	+ -	Defines the materials to be used by the platforms in this group
Platforms	1 Array element	+ -	Add platforms to this group
Time	2,0		Time (in seconds) during which platforms will remain in their visibility state. Applies to platforms of all groups
Security Time	1,0		Platforms become visible again x seconds before their normal appearance. Allowing a period of safety where all platforms are visible. Applies to platforms of all groups

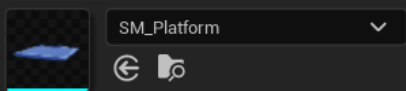



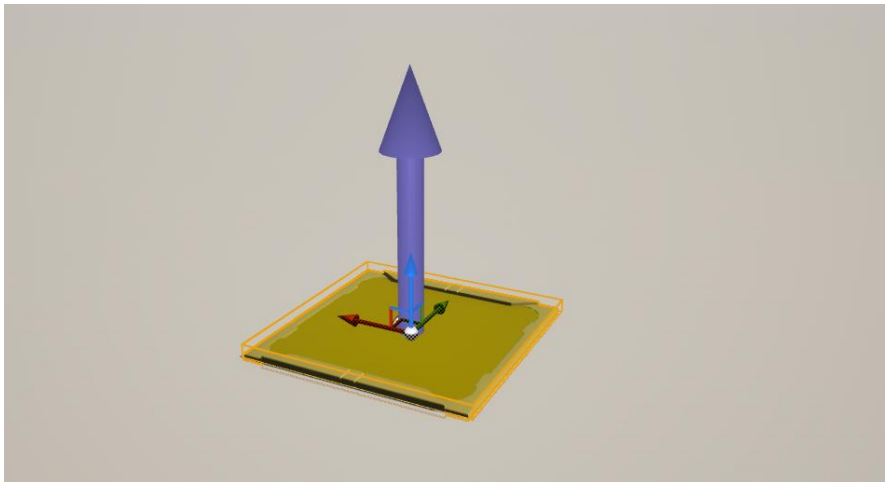
Bumping platform



Automatically launch the player in the direction of the platform's Arrow Component

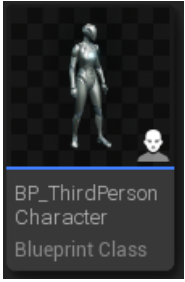
Instance parameters

Mesh		Assigns a Static Mesh to the platform
Materials	2 Array elements 	Assigns materials to the platform
Power	1500,0	Defines the force with which the player will be Launched
Override XY	<input type="checkbox"/>	Defines if this force overrides velocities or adds to the player's velocities
Override Z	<input type="checkbox"/>	



Jump

Jumping is improved over the basic jump provided by the Third Person Template. It's sharper, and the character doesn't hover.



It has been implemented in **BP_ThirdPersonCharacter**.

The entire implementation of this jump is commented on and easily identifiable.

This will enable you to easily implement it in your character via copy/paste.

Player ability management is a complex task. And it's all the more complex when it comes to implementing it in a marketplace pack. The jump ability has been implemented as such to ensure maximum compatibility with your systems.

For example, [ADS Framework](#) contains an ability system where absolutely no abilities are coded in the Player Blueprint. Conversely, many projects implement them directly in the Player Blueprint.

Here, the jump ability has been implemented as such to ensure maximum compatibility with different systems without having to provide a complex system that would increase the price of the pack. This pack is not a template.

Contact

If you have any questions or comments, please don't hesitate to contact me in **English** or **French** by e-mail or private message on my social networks.

azure.dream.studio@gmail.com

Mail

So that I know that your e-mail is not spam, please include **[ADS_PlatformerKit]** at the beginning of the subject line.

Example:

[ADS_PlatformerKit] Question about the bumping platform

X

<https://twitter.com/AzureDreamStud1>

Discord

User name: azuredreamstudio

Server invitation: <https://discord.gg/Vy3TvsjKpQ>

At the time of writing, my discord server has only a few members, and none of them are active. So I don't check the server every day. If the server grows, I'll be much more active, but for the moment I'm less reactive than by e-mail or X.

Please, if you are not satisfied with something, please contact me before giving a bad review.