**Simple Waypoint System Documentation**

V5.4+

[Scripting Reference 1](#_bookmark0)

[Quick Start 2](#_bookmark1)

[Path Creation 3](#_bookmark2)

[Path Editing 4](#_bookmark3)

[Movement 5](#_bookmark4)

[Advanced 5](#_bookmark5)

[Example Scenes 6](#_bookmark6)

[Plugins 6](#_bookmark7)

[Contact 6](#_bookmark8)

Thank you for buying Simple Waypoint System!

Your support is greatly appreciated.

# Scripting Reference

[www.rebound-games.com/docs/sws](http://www.rebound-games.com/docs/sws)

**Simple Waypoint System Documentation**

V5.4+

[脚本参考 1](#_bookmark0)

[快速开始 2](#_bookmark1)

[路径创建 3](#_bookmark2)

[路径编辑 4](#_bookmark3)

[移动 5](#_bookmark4)

[高级功能 5](#_bookmark5)

[示例场景 6](#_bookmark6)

[插件 6](#_bookmark7)

[联系 6](#_bookmark8)

Thank you for buying Simple Waypoint System!

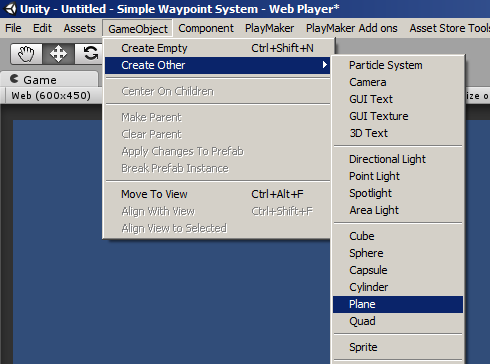
Your support is greatly appreciated.

# Scripting Reference

[www.rebound-games.com/docs/sws](http://www.rebound-games.com/docs/sws)

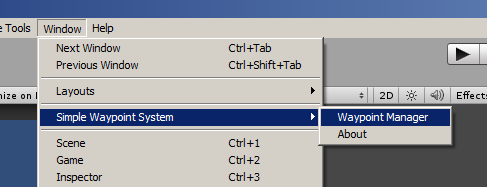
# Quick Start

We will start by creating a new path in 3D space. When placing waypoints in 3D, you will need to have colliders in the scene, which act as background objects for the raycast.

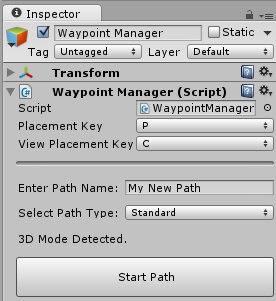


**Open a new scene** and **create a plane**. This object will be used for placing waypoints.

Next, **add the Waypoint Manager** to your scene by using our editor menu.

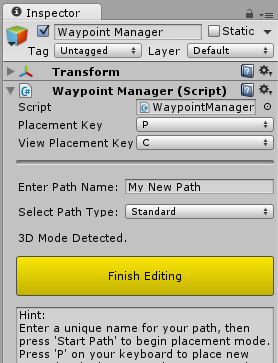
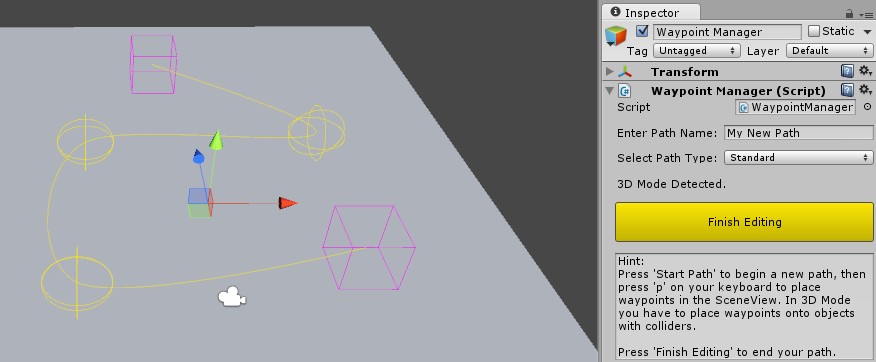


With the Waypoint Manager selected, **enter the name** of your path, leave the path type at “Standard” for now and **press “Start Path”**. The button will turn yellow, indicating we‟re in active placement mode.

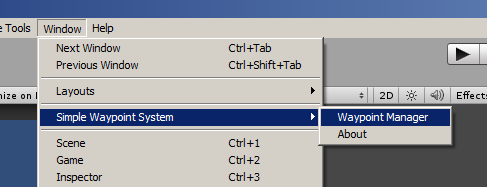
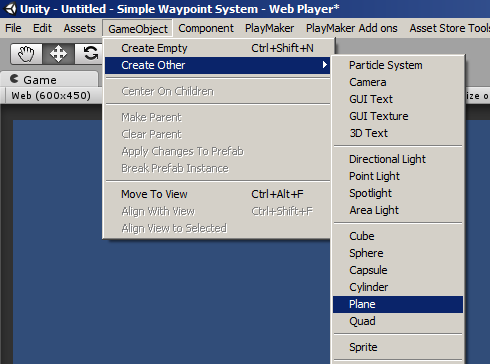


Note: you can customize the „placement key‟ which you would like to use for placing waypoints in the scene, but also the „view placement key‟ which places waypoints at your current scene view camera position (useful for cutscene paths).

**Place waypoints** onto the plane and your mouse position by pressing „P‟ on your keyboard. When you have placed enough waypoints, **press “Finish Editing”** to exit placement mode.



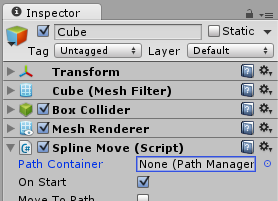
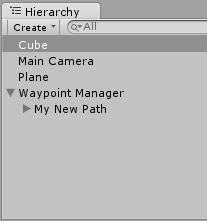
我们将从在3D空间中创建新路径开始。  
在3D中放置航路点时，您将需要在场景中具有碰撞器，该碰撞器充当光线投射的背景对象



打开一个新场景并创建一个飞机。  
该对象将用于放置航点。

接下来，使用我们的编辑器菜单将Waypoint Manager添加到场景中。

We‟ll continue by letting a game object follow this path. **Create a cube** and attach the **movement script “splineMove”** to it. Assign the newly created **path to its container** and **check “On Start”**.



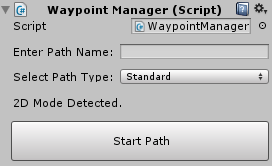
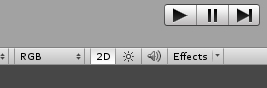
Press play to see the result!

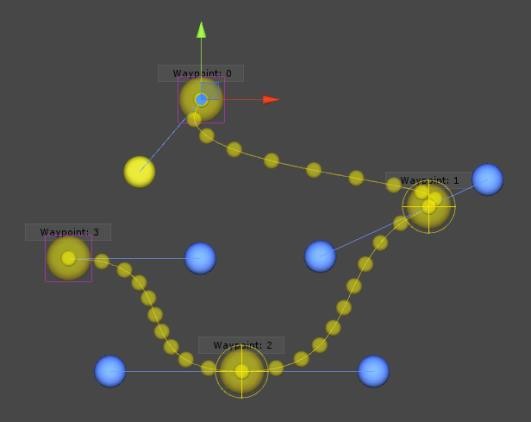


This concludes our introduction to Simple Waypoint System. Please read further to learn more about scripts and techniques when creating paths as well as about movement scripts and advanced walker settings.

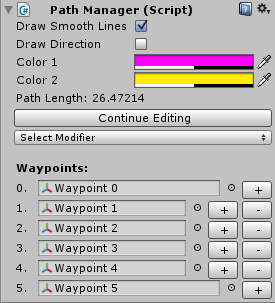
# Path Creation

We‟ve already covered 3D path creation in the introduction. **2D path creation** is just as easy as that. In 2D mode, you won‟t need background objects with colliders, because waypoints will be placed at zero depth by default. To enable 2D placement, switch to 2D mode and let the Waypoint Manager detect the new placement mode for you.



The Waypoint Manager has a drop-down list for selecting path types and comes with another built- in type. **Bezier paths** have additional controls per waypoints, the so-called control points. While these give you more flexibility in defining the shape of the path, movement scripts on bezier paths are not that flexible when it comes down to more advanced movement settings (such as random waypoints). Creating bezier paths works in the same way as described before.

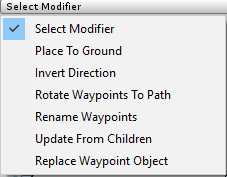
# Path Editing

Standard paths and their **Path Manager component** allow for customization of visual presentations in the editor, such as gizmo connections or colors, and repositioning. In the inspector, each waypoint slot has a button to add and/or remove waypoints at the corresponding index.

The **path modifier dropdown** has several quickly- accessible functions to apply common path changes:

**Place to Ground:** raycasts against colliders beneath waypoints and tries to reposition them on top of hit points.

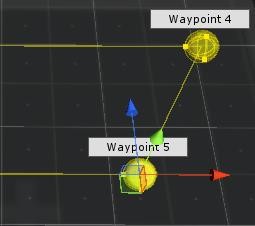
**Invert Direction:** inverts the current order of waypoints.

**Rotate Waypoints To Path:** is a useful method for rotating the waypoints to each other, which comes in handy when using waypoint rotation on movement scripts.

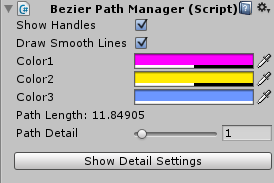
**Rename Waypoints:** renames the waypoint game objects in the order they are referenced in the path, so they get added up correctly (0,1,2,3…). You can do this after you gave your waypoints a custom name or skip custom names when renaming them on purpose.

**Update From Children:** lets you rebuild the path after you‟ve added or removed child objects, which are then taking into account for the path calculation.

**Replace Waypoint Object:** will display a replacement object slot, which you assign with a different game object or prefab, in order to replace all default waypoint game objects with that object.



You can modify waypoints with their **handles in the scene view**. The handle type depends on your selected Unity tool. If you would like to move waypoints around, select Unity‟s **move tool ** and click on a waypoint gizmo to select it. The move handles will show up on the selected waypoint. If you would like to rotate a waypoint instead, select the **rotate tool ** and the rotation handles will show up on the current waypoint selection.

**Bezier Path Manager components** extend standard paths with fine control over the detail of path segments.

**Path detail** defines the number of calculated path points on the whole bezier curve. For individual detail on path segments between waypoints, press **Show Detail Settings** and enable custom detail.

标准路径及其“路径管理器”组件允许自定义编辑器中的视觉表示，例如Gizmo连接或颜色，以及重新定位。  
在检查器中，每个航点插槽都有一个按钮，用于在相应索引处添加和/或删除航点。

路径修改器下拉菜单具有几个可快速访问的功能，以应用常见的路径更改：

* 放置到地面：对航路点下方的碰撞器进行射线广播，并尝试将其重新放置在命中点上方。
* 反转方向：反转航路点的当前顺序。
* 将航点旋转到路径：这是一种将航点彼此旋转的有用方法，在移动脚本上使用航点旋转时非常方便。
* 重命名路标：按照在路径中被引用的顺序重命名路标游戏对象，以便正确添加它们（0、1、2、3…）。  
  您可以为路点指定一个自定义名称，也可以在有意重命名它们时跳过自定义名称。
* 从子对象更新：添加或删除子对象后，您可以重建路径，然后将这些对象考虑到路径计算中。
* 替换航点对象：将显示一个替换对象插槽，您将其分配给其他游戏对象或预制件，以便用该对象替换所有默认的航点游戏对象。  
  您可以在场景视图中使用其控点修改路标。手柄类型取决于您选择的Unity工具。如果要四处移动航路点，请选择Unity的移动工具，然后单击一个航路点小控件以将其选中。移动手柄将显示在选定的航点上。如果您想旋转航路点，请选择旋转工具，旋转手柄将显示在当前航路点选择上。

Bezier路径管理器组件扩展了标准路径，并且可以很好地控制路径段的细节。  
路径细节定义了整个贝塞尔曲线上计算出的路径点的数量。有关航点之间路径段的详细信息，请按显示详细信息设置，然后启用自定义详细信息。

# Movement

Movement on a path takes nothing more than one movement script attached to your object. Assign the desired path to the path container of your movement script, play around with its settings until you are satisfied and you‟re good to go!

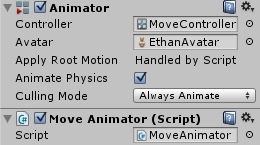
You have the choice between 2 movement scripts, depending on your path and app design. All of them are built on the same basis, but each one has a few different functions to consider.

**splineMove**: Provides linear or curved movement on standard and bezier paths.

**navMove**: Uses NavMesh and NavMeshAgents for standard paths.

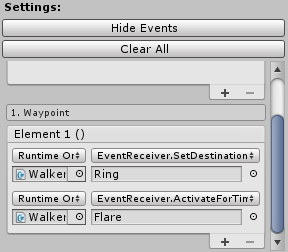
Please take a look at our [Scripting Reference](#_bookmark0) for an overview of public variables and methods.

# Advanced

Simple Waypoint System comes with support for **Mecanim Animator Controllers**. Just attach an Animator and Move Animator component to the moving object and choose one of our controllers (MoveController or MoveController2D) to get started right away.

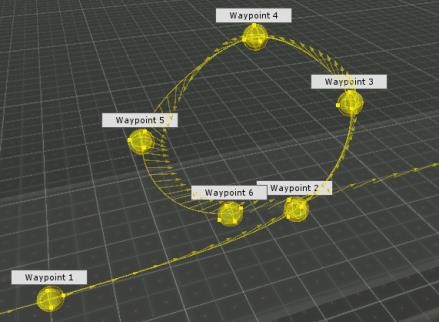
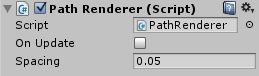
Our controllers have variables and multiple transitions for speed and direction, which are being updated by the Move Animator script as the object moves.

**Events** are a great functionality to hook up your own methods and behaviors to waypoints. With a path assigned to your movement script, press “Show Events” to unhide event slots per waypoint.



Events work exactly like script calls and allow for one argument. You can define your methods in a separate script as usual, then attach it to an object in the scene.

After programming your methods in the receiving script, reference the method name and an optional argument in the event. You could also define multiple events per waypoint. Have a look at the Events example scene for an overview of possible use-cases.

By default, paths in Simple Waypoint System are only drawn in the editor by using gizmos. If you would like them to be drawn at runtime, that‟s what the **Path Renderer** component is for. When attached to a path, this script will utilize Unity‟s Line Renderer to draw connections of waypoints, based on the path type.

**Waypoint Rotation** is an *experimental* feature, allowing you to define custom rotations per waypoint for the moving object. To use it, toggle “Draw Direction” on paths. With the path selected use the rotate tool to rotate path waypoints. On splineMove, set “Path Mode” to “Ignore” and “Waypoint Rotation” to “All”.

splineMove：在标准和贝塞尔曲线路径上提供线性或曲线运动。

navMove：将NavMesh和NavMeshAgents用于标准路径。

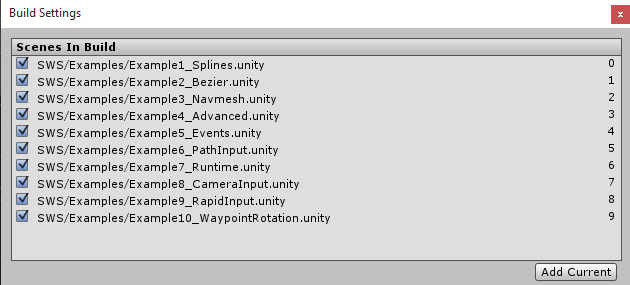
Simple Waypoint System附带Mecanim动画控制器的支持。只需将Animator和Move Animator组件附加到移动对象，然后选择我们的一个控制器（MoveController或MoveController2D）即可立即开始。  
我们的控制器具有变量以及速度和方向的多个转换，当对象移动时，Move Animator脚本会对其进行更新。

事件是一种很棒的功能，可以将您自己的方法和行为连接到航点。为运动脚本分配路径后，按“显示事件”以取消隐藏每个航路点的事件时段。  
事件的工作方式与脚本调用完全相同，并且允许一个参数。您可以像往常一样在单独的脚本中定义方法，然后将其附加到场景中的对象。  
在接收脚本中对方法进行编程后，请在事件中引用方法名称和可选参数。您还可以为每个航路点定义多个事件。请查看“事件”示例场景，以获得可能的用例概述。

默认情况下，仅使用gizmos在编辑器中绘制Simple Waypoint System中的路径。如果您希望在运行时绘制它们，这就是Path Renderer组件的用途。附加到路径后，该脚本将根据路径类型利用Unity的Line Renderer绘制航路点的连接。

航点旋转是一项实验性功能，可让您为移动对象定义每个航点的自定义旋转。要使用它，请在路径上切换“绘制方向”。选择路径后，使用旋转工具旋转路径航路点。在splineMove上，将“路径模式”设置为“忽略”，将“航点旋转”设置为“全部”。

# Example Scenes

Everything we discussed in this documentation is also demonstrated in example scenes, as well as a lot more! Runtime access, message or 2D usage and custom hacking of movement scripts are just a few to name here. For a tour through all examples, please **add our example scenes** at the beginning of your **build settings** and start with the example scene for splines:

 Example1\_Splines

# Plugins

For PlayMaker users, there is a separate package with custom actions and a sample scene included, located in the project panel at SWS > Plugins > PlayMaker.

# Contact

As full source code is provided and every line is well-documented, please feel free to take a look at the scripts and modify them to fit your needs.

If you have any questions, comments, suggestions or other concerns about our product, do not hesitate to contact us. You will find all important links in our About window, located under Window > Simple Waypoint System.



For private questions, you can also email us at [info@rebound-games.com](mailto:info@rebound-games.com)

If you would like to support us on the Unity Asset Store, please write a short review there so other developers can form an opinion. Again, thanks for your support, and good luck with your apps!

## Rebound Games