Simple Waypoint System Documentation

V5.2+

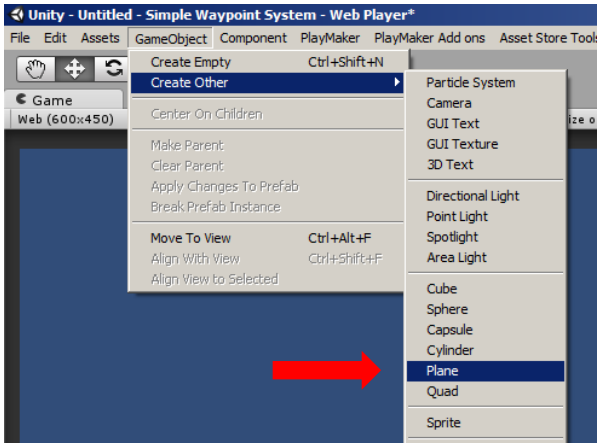
Thank you for buying Simple Waypoint System!
Your support is greatly appreciated.

Scripting Reference

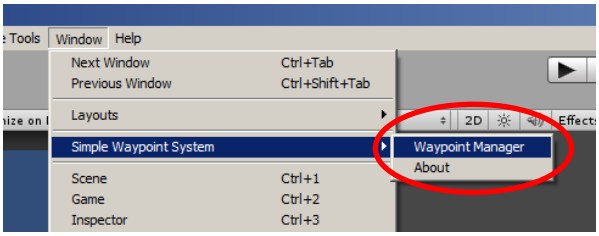
www.rebound-games.com/docs/sws

Quick Start

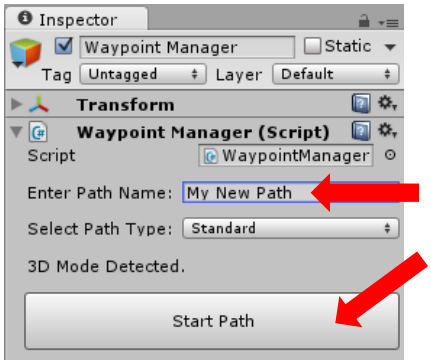
We will start by creating a new path in 3D space. When placing waypoints in 3D, you will need to have colliders in the scene, which act as background objects for the raycast.



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

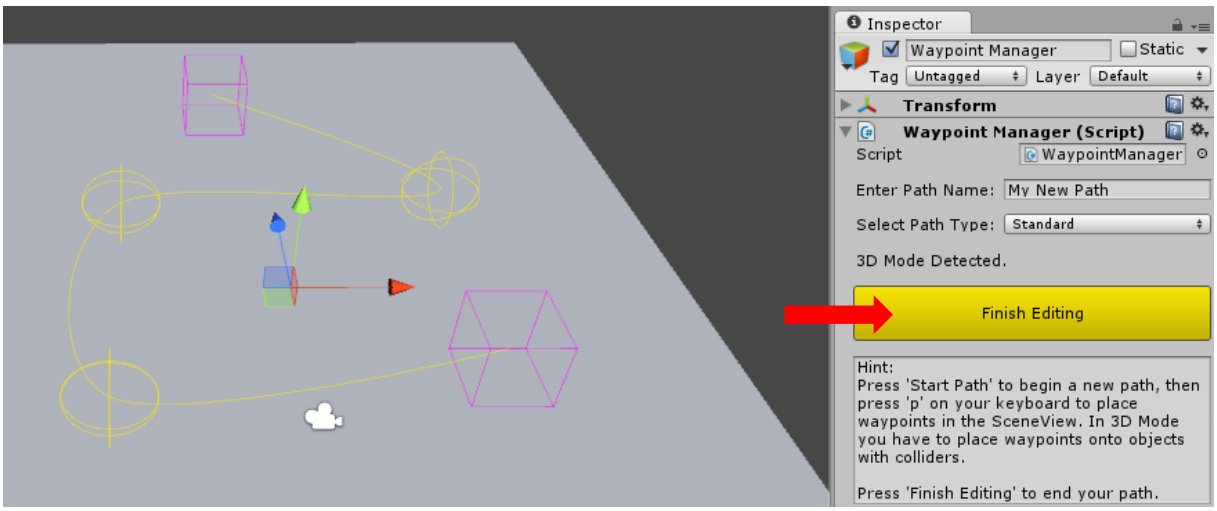


Next, **add the Waypoint Manager** to your scene by using our editor menu.



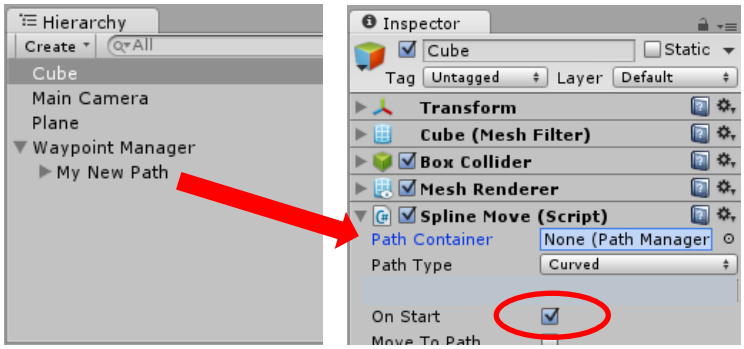
With the Waypoint Manager selected, **enter the name** of your path, leave the path type at "Standard" for now and **press "Start Path"**. The button will turn yellow, indicating we're in active placement mode.

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.



Note: Do not click on other objects in the scene view while creating a path. This would lose focus on the Waypoint Manager, destroy the current path and deactivate placement mode.

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"**.

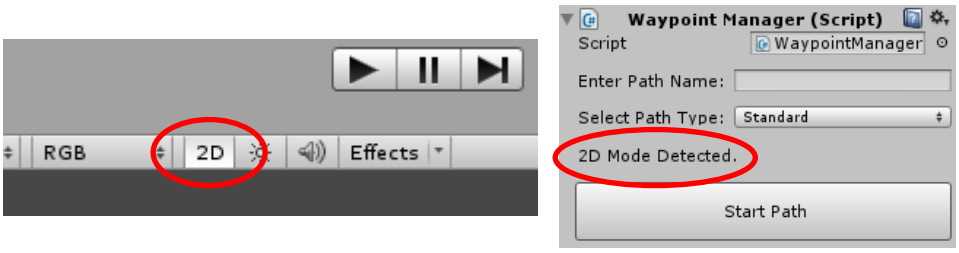


Press play to see the result!

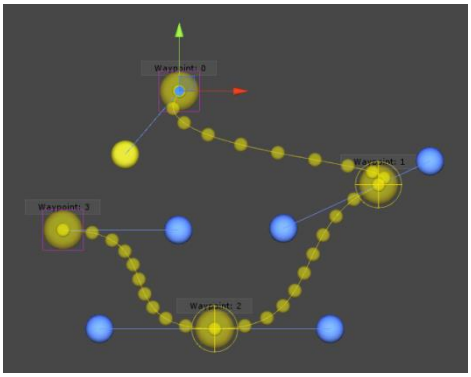
This concludes our introduction to Simple Waypoint System. Please read further to learn more about scripts and techniques when creating paths as well as about movement scripts and advanced walker settings.

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.

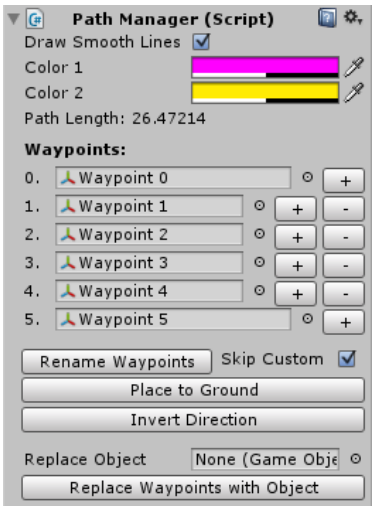


The Waypoint Manager has a drop-down list for selecting path types and comes with another built-in type. **Bezier paths** have additional controls per waypoints, the so-called control points. While these give you more flexibility in defining the shape of the path, movement scripts on bezier paths are 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. You can move waypoints with their handles in the scene view.



In the inspector, each waypoint slot has a button to add and/or remove waypoints at the corresponding index.

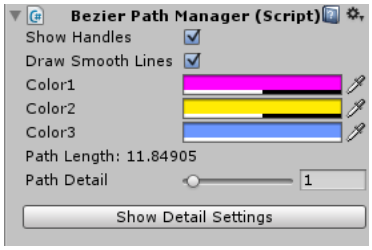
"Rename Waypoints" renames the waypoint game objects in the order they are referenced in the path, so they get added up correctly (0,1,2,3…). You can do this after you gave your waypoints a custom name or skip custom names when renaming them on purpose.

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

"Invert Direction" inverts the currents order of waypoints.

If you assign a replace object and press on the button below it, all waypoints will be replaced with this object.

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.

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.

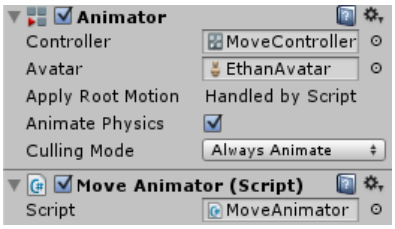
navMove: Uses NavMesh and NavMeshAgents for standard paths.

Please take a look at our Scripting Reference 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.

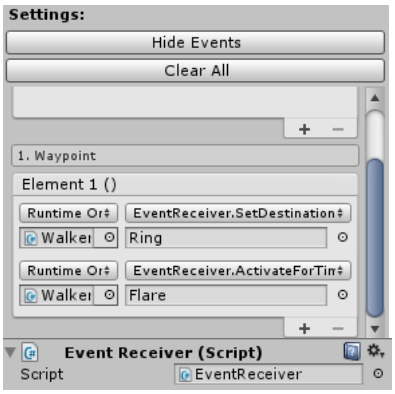
Our controllers have variables and multiple transitions for speed and direction, which are being updated by the Move Animator script as the object moves.



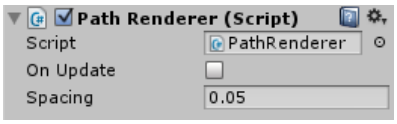
Events are a great functionality to hook up your own methods and behaviors to waypoints. With a path assigned to your movement script, press "Show Events" to unhide event slots per waypoint.

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.

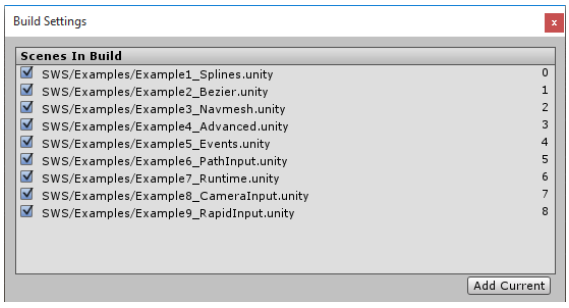


By default, paths in Simple Waypoint System are only drawn in the editor by using gizmos. If you would like them to be drawn at runtime, that's what the **Path Renderer** component is for. When attached to a path, this script will utilize Unity's Line Renderer to draw connections of waypoints, based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



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.

Simple Waypoint System uses the efficient, free tween library DOTween for movement along paths. Please consider donating or buying the Pro version of DOTween at a later point, if you would like to further support its development.

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

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