

Waypoints

An easy to use yet powerful waypoint system for Unity.

V 1.0.1

- Incomplete documentation parts will be improved over time.
- Get the most up to date documentation by [clicking here](#).
- Remember you can hover over fields in the “Inspector” window in Unity’s editor to read tooltip explanations of each field.
- If you have any questions or need assistance email support at intuitivegamingsolutions@gmail.com.

Table of Contents

1. [Table Of Contents](#)
2. [How to: Import the Package into a Unity Project](#)
3. [Getting Started](#)
 - 3.a. [Setting Up Your First PathMove](#)
4. [Scene View Path Visualizations](#)
 - 4.a. [Visualization Breakdown](#)
5. [Path Modes](#)
 - 5.a. [Stop At End](#)
 - The mover will simply stop at the end of the path.
 - 5.b. [Closed](#)
 - The mover will continuously move through the path in a closed loop.
 - 5.c. [Ping Pong](#)
 - The mover will ping pong to and from the end of the path.
6. [PathMove Components](#)
 - 6.a. [Shared Settings](#)
 - 6.b. [TransformPathMove](#)
 - 6.c. [RigidbodyPathMove](#)
 - 6.d. [NavAgentPathMove \(coming soon\)](#)
 - 6.e. [ForcePathMove \(coming soon\)](#)
7. [Rotation Modes](#)
 - 7.a. [None](#) - no rotation.
 - 7.b. [Forward To Point](#) - faces the forward vector of the currently activated waypoint.
 - 7.c. [Copy Waypoint](#) - copies the rotation of the currently activated waypoint.
 - 7.d. [Interpolate Waypoint](#) - interpolates between the current and next waypoint rotations.
8. [Interpolation Modes](#)
 - 8.a. [Linear](#) - interpolate linearly between points.
 - 8.b. [BSpline](#) - use a b-spline to interpolate between points.

- 8.c. [CatmullRom](#) - use a catmull rom spline to interpolate between points.
- 9. [PathMove Editor Breakdown](#)
 - 9.a. [PathMove Editor Settings](#)
 - 9.b. [PathMove Editor Buttons](#)
- 9. [Waypoint Editor & Gizmo Settings](#)
- 10. [Screenshots](#)
 - 10.a. [The 'PathMove Editor Settings' Window](#)
 - 10.b. [The 'PathMove Editor' Window](#)
- 11. [FAQ](#)

NOTE: See 'API Reference.pdf' if you are looking for source code documentation.

How to: Import the Package into a Unity Project

There are 2 ways to import the package.

- a. (Recommended) Using the Unity Editor 'Package Manager'.
 - i. Open the Windows→Package Manager using the Unity editor toolbar.
 - ii. In the upper-left corner of the Package Manager window select 'Packages: My Assets'.
 - iii. Search for "Waypoints" in the list or use the search bar in the window.
 - iv. Select the asset in the package manager, select 'Download'.
 - v. After the package has finished downloading click 'Import' to import it into the project.
- b. Importing Waypoints.untypackage
 - i. Using the Unity Editor's toolbar select Assets→Import Package
 - ii. In the file explorer that opens navigate to Waypoints.untypackage
 - iii. Double click the package and import it.

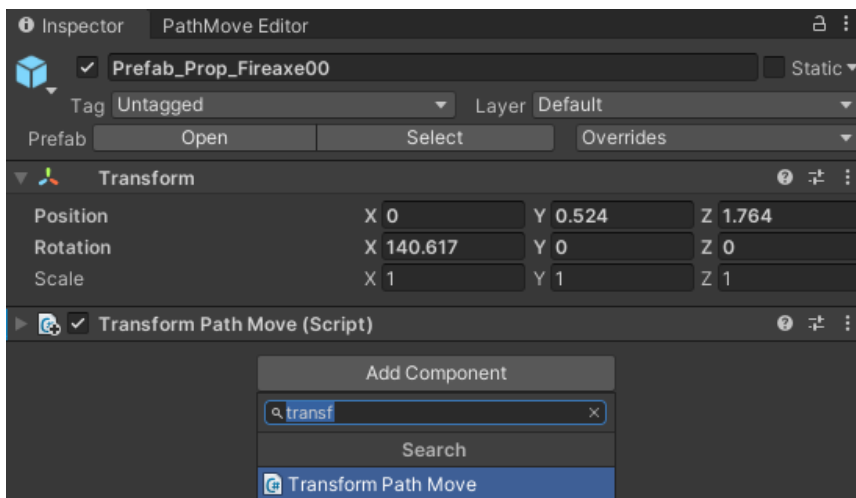
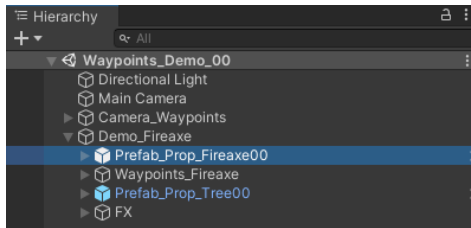
Getting Started

([Demo Video - Youtube](#))

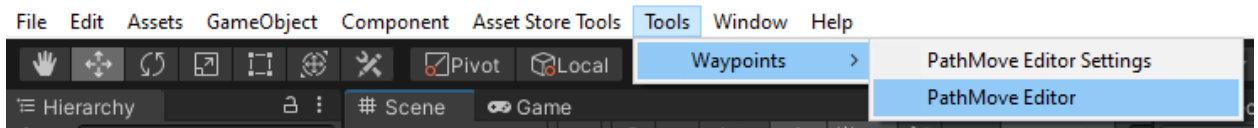
- To get started first simply [import the package from the 'Package Manager'](#).

3.a. Setting Up Your First PathMove

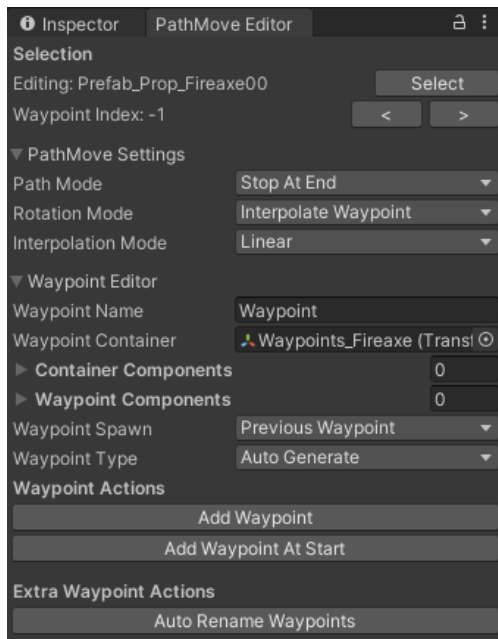
1. After importing the package add the relevant **PathMove** (e.g. **TransformPathMove**) component to the GameObject that you want to move along a path.



2. Note that you can customize almost every setting in the 'Tools → Waypoints → PathMove Editor Settings' window.
3. Next, in the Unity Editor toolbar navigate to 'Tools → Waypoints → PathMove Editor'.



4. You will see the editor open as shown in the screenshot below:



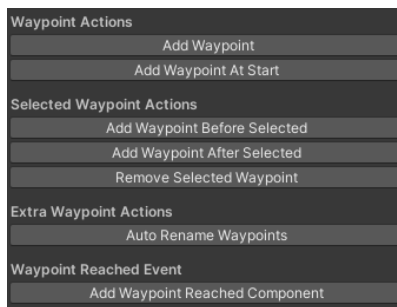
Hovering over any of the settings will open a 'Tooltip' explaining what the setting is.

5. Make sure the GameObject you want to move along the path is still selected in the 'Hierarchy' pane. Its name should appear in the 'Editing: ' part of the 'Selection' section of the editor.
6. Tune your desired settings before adding waypoints.
 - a. **Waypoint Spawn** - controls where the editor spawns new waypoints.
 - i. **Previous Waypoint** - Spawns a waypoint that comes before the selected waypoint, or if no waypoint is selected spawns on the last waypoint, if there are no waypoints spawns on the GameObject that will move along the path.
 - ii. **Mover** - Spawns on the GameObject that will move along the [PathMove](#).
 - b. **Waypoint Type** - controls what kind of waypoint is spawned.
 - i. **Auto Generate** - Generates a 'ghost' waypoint automatically based on the GameObject that will be moved along the path.
 - ii. **Prefab** - Instantiates a prefab instance for each new waypoint specified by the 'Waypoint Prefab' setting that appears while in this mode.
 - iii. **Empty** - Create an empty GameObject for each waypoint.

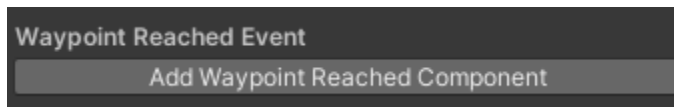
CONSIDERATIONS:

1. *Note that this setting will automatically change to 'Prefab' mode with the reference automatically set when you start editing an existing path that uses prefabs.*
 2. *Similarly, if you start editing an existing path that does not use prefabs it will automatically revert back to 'Auto Generate' mode.*
- c. **Waypoint Container** - (Optional, but recommended) A reference to the GameObject that
 - d. **Container Components** - An array of class references (components) that should be automatically added to the 'Waypoint Container' GameObject.

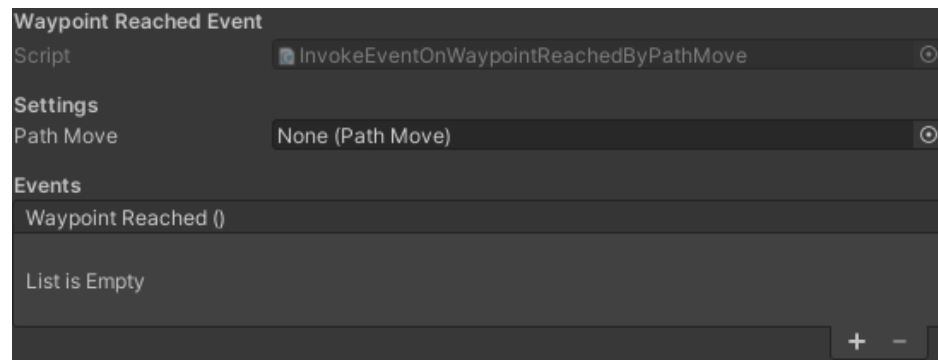
- i. Note that the 'Validate Container Components' button can be clicked to automatically add any missing components to existing waypoints. The button will only appear while this array is *not empty*.
 - e. **Waypoint Components** - An array of class references (components) that should be automatically added to each new waypoint.
 - i. Note that the 'Validate Waypoint Components' button can be clicked to automatically add any missing components to existing waypoints. The button will only appear while this array is *not empty*.
7. Now simply click 'Add Waypoint' to add a new waypoint at the end of the path. Below is a screenshot showing all possible 'Add Waypoint' button variations – *note that some only appear while they are relevant*.



8. (Optional) Add a 'Waypoint Reached Event' component by clicking the button.



- After clicking the 'Add Waypoint Reached Event' button you can modify the event directly in the editor without having to go back to the 'Inspector'.

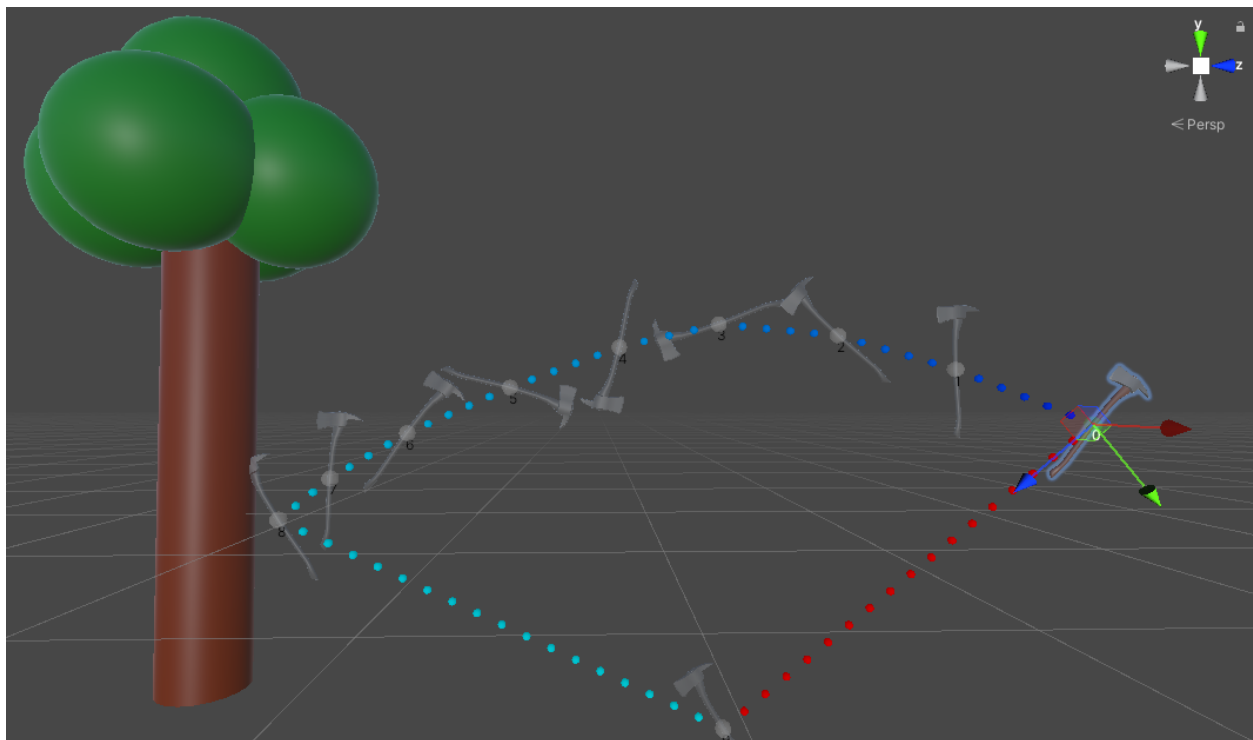


9. Tune the 'PathMove Settings' to your liking before clicking 'Play'.
- a. For more information about the path visualization [read here](#).

Scene View Path Visualizations

4.a. Visualization Breakdown

- Paths are drawn in the scene view for any selected **PathMover** component.
- Here is a breakdown of the below screenshot visualization:
 - The dark blue waypoints represent the start of the path.
 - The teal waypoints represent the end of the path.
 - The red waypoints represent the

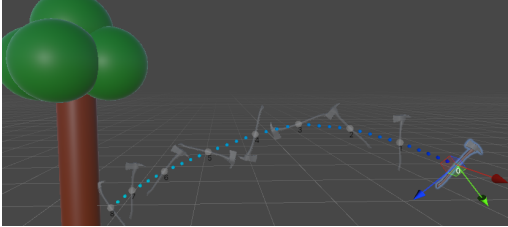


A screenshot of a PathMove path visualization in the scene view using v1.0.0

Path Modes

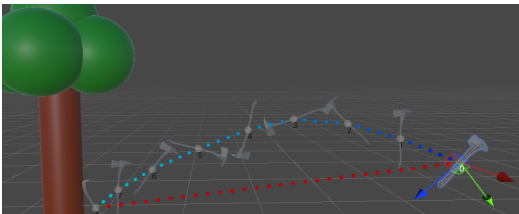
5.a. Stop At End

- The mover will simply stop at the end of the path.



5.b. Closed

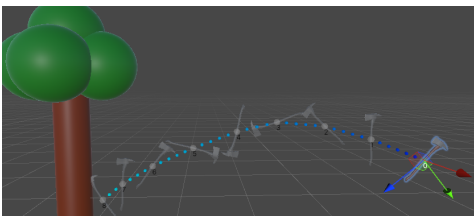
- The mover will continuously move through the path in a closed loop.



- The red line indicates the 'closed loop' line, at the end of the path the mover will move from the teal line back to the start of the path.

5.c. Ping Pong

- The mover will ping pong to and from the end of the path.
- When moving in the forward direction the mover will move from the first dark blue waypoint to the teal waypoints and then back again repeatedly.



PathMove Components

6.a. Shared Settings

- **Path Mode** - The '[Path Mode](#)' to use to follow the configured path.
- **Rotation Mode** - The '[Rotation Mode](#)' to use
- **Interpolation Mode** - The '[Interpolation Mode](#)' to use.

6.b. TransformPathMove

- Kinematically moves a Transform along a path.

6.c. RigidbodyPathMove

- Kinematically moves a Rigidbody along a path while updating relevant values such as Rigidbody.velocity.

6.d. NavAgentPathMove *(coming soon)*

- Moves a NavMeshAgent along a path using the component.

6.e. ForcePathMove *(coming soon)*

- Moves a Rigidbody along a path using physical forces.

Rotation Modes

7.a. None

- No rotation

7.b. Forward To Point

- Faces the forward vector of the currently activated waypoint.

7.c. Copy Waypoint

- Copies the rotation of the currently activated waypoint.

7.d. Interpolate Waypoint

- Interpolates between the current and next waypoint rotations.

Interpolation Modes

8.a. Linear

- Interpolates linearly between waypoints.

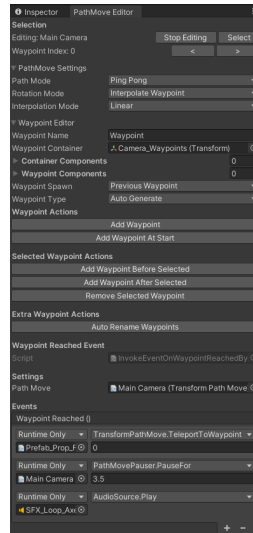
8.a. BSpline

- Interpolates using a b-spline between waypoints.

8.a. CatmullRom

- Interpolates using a catmull-rom spline between waypoints.

PathMove Editor Breakdown



See [Screenshot @ 10.b](#)

9.a. PathMove Editor Settings

- b. **Waypoint Spawn** - controls where the editor spawns new waypoints.
 - i. **Previous Waypoint** - Spawns a waypoint that comes before the selected waypoint, or if no waypoint is selected spawns on the last waypoint, if there are no waypoints spawns on the GameObject that will move along the path.
 - ii. **Mover** - Spawns on the GameObject that will move along the [PathMove](#).
- c. **Waypoint Type** - controls what kind of waypoint is spawned.
 - i. **Auto Generate** - Generates a 'ghost' waypoint automatically based on the GameObject that will be moved along the path.
 - ii. **Prefab** - Instantiates a prefab instance for each new waypoint specified by the 'Waypoint Prefab' setting that appears while in this mode.
 - iii. **Empty** - Create an empty GameObject for each waypoint.

CONSIDERATIONS:

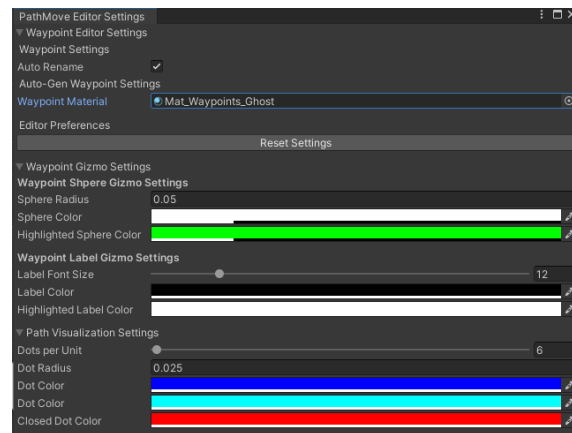
1. *Note that this setting will automatically change to 'Prefab' mode with the reference automatically set when you start editing an existing path that uses prefabs.*
 2. *Similarly, if you start editing an existing path that does not use prefabs it will automatically revert back to 'Auto Generate' mode.*
- d. **Waypoint Container** - (Optional, but recommended) A reference to the GameObject that will hold the waypoints.
 - e. **Container Components** - An array of class references (components) that should be automatically added to the 'Waypoint Container' GameObject.
 - i. Note that the 'Validate Container Components' button can be clicked to automatically add any missing components to existing waypoints. The button will only appear while this array is *not empty*.

- f. **Waypoint Components** - An array of class references (components) that should be automatically added to each new waypoint.
 - i. Note that the 'Validate Waypoint Components' button can be clicked to automatically add any missing components to existing waypoints. The button will only appear while this array is *not empty*.

9.b. PathMove Editor Buttons

- **Stop Editing** - Stops editing the [PathMove](#) component that is currently being edited.
 - This button only appears when the [PathMove](#) is not the active selection.
- **Select** - Selects the GameObject that has the [PathMove](#) component that is currently being edited.
- **Validate Container Components** - Automatically adds any missing 'Container Components' to the specified 'Waypoint Container'.
 - This button only appears when there is at least 1 entry in the 'Container Components' array.
- **Validate Waypoint Components** - Automatically adds any missing 'Waypoint Components' to all waypoints registered with the [PathMove](#) that is currently being edited.
 - This button only appears when there is at least 1 entry in the 'Waypoint Components' array.
- **Add Waypoint** - Adds a waypoint at the end of the path.
- **Add Waypoint At Start** - Adds a waypoint at the start of the path.
 - This button only appears when there is at least 1 waypoint registered for the [PathMove](#) being edited.
- **Add Waypoint Before Selected** - Adds a waypoint before the selected waypoint.
 - This button only appears when a waypoint is selected.
- **Add Waypoint After Selected** - Adds a waypoint after the selected waypoint.
 - This button only appears when a waypoint is selected.
- **Auto Rename Waypoints** - Automatically renames all waypoints in the [PathMove](#) currently being edited to be sequential.
 - This button only appears when there is at least 1 waypoint registered for the [PathMove](#) being edited.
- **Add Waypoint Reached Component** - Adds a
 - This button only appears when a waypoint is selected that does not already have one of these components.
 - If the selected waypoint already has a [InvokeEventOnWaypointReachedByPathMove](#) component the Inspector for this component will be rendered in place of the button.

Waypoint Editor & Gizmo Settings

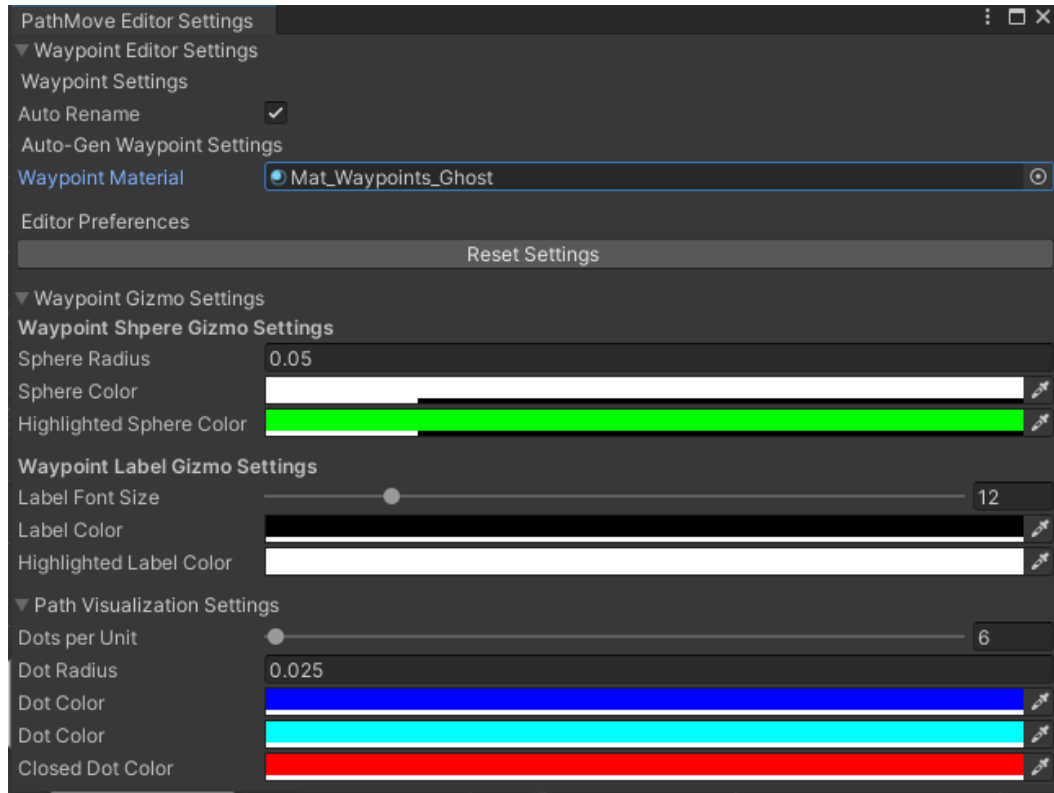


See [Screenshots @ 10.a](#)

Screenshots

10.a. The 'PathMove Editor Settings' Window

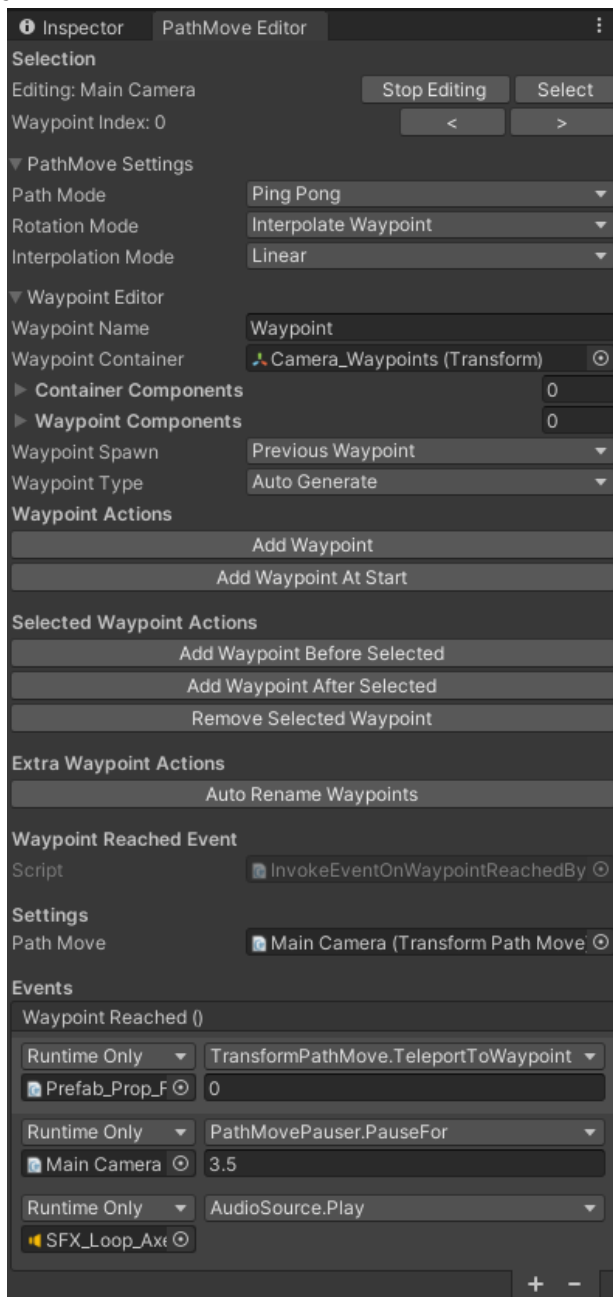
- The 'PathMove Editor Settings' window can be accessed by navigating in the Unity Editor toolbar to 'Tools → Waypoints → PathMove Editor Settings'.
- *Remember to hover over settings in the editor to view their descriptions.*



A screenshot of the 'PathMove Editor Settings' window in v1.0.0

10.b. The 'PathMove Editor' Window

- The 'PathMove Editor Window' allows you to edit the paths of any [PathMove](#) component through a clean and simple visual user interface. This can be accessed by navigating to 'Tools → Waypoints → PathMove Editor'.
- *Remember to hover over settings in the editor to view their descriptions.*
- Keep in mind that **the contents of the editor window will vary depending on what you have selected.**



A screenshot of the 'PathMove Editor' window in v1.0.0 with a waypoint that has an [InvokeEventOnWaypointReachedByPathMove](#) component.

FAQ

(Frequently Asked Questions)

1. How can I make it so waypoint names are automatically numbered without having to press 'Auto Rename Waypoints'?
 - a. Easy! **'Auto renaming'** for waypoints is off by default to avoid disrupting existing work, if you want waypoint GameObject names to be automatically numbered enable this setting in the 'PathMove Editor Settings' window.