

PATHOS+

User Manual

What is PathOS+?

PathOS+ is an end-to-end, lightweight framework for simulating player behavior. PathOS+ agents approximate player navigation in a game's world, and can be viewed in real-time or recorded for later visualization. Agents can also be customized to mimic different player motivations.

PathOS+ is built for Unity, and designed to operate on top of your existing game projects, requiring no instrumentation or modification of game assets or code.

Happy playtesting!

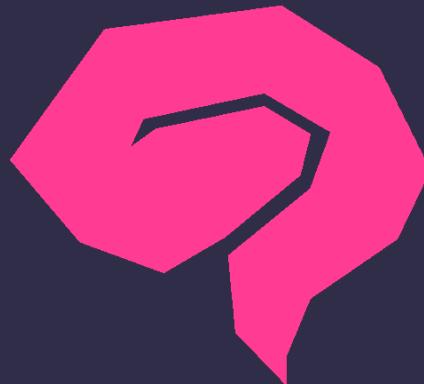


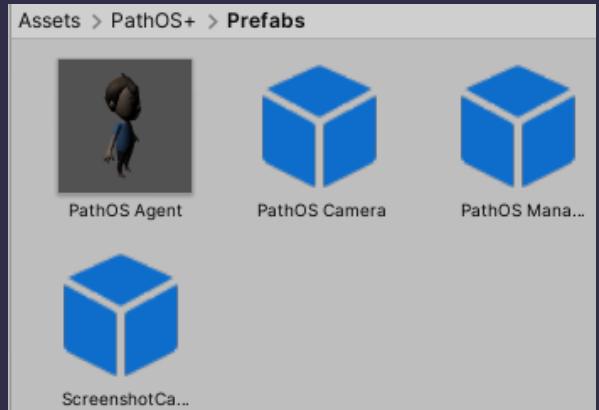
Table of Contents

Quickstart Guide	4
Project Set-Up	5
PathOS+ Window	6
Level Markup	7
Game Entities	8
Runtime Interface	9
Agent Customization	12
Batch Simulation	13
Data Recording and Visualization	14
Expert Evaluation	16
Troubleshooting	17

Quickstart Guide

First Thing's First

Set up the Unity Navmesh if you haven't already from *Window>AI>Navigation*. Make sure you have **PathOSManager** and **PathOSAgent** objects in the scene - prefabs can be found in *PathOS+/Prefabs*. If you wish to take and export screenshots you can also drag and drop the **ScreenshotCamera** object into the scene. Alternatively, you can go to the **Setup tab** of the **PathOS+ Window** (see *PathOS+ Window*) and instantiate the prefabs there.



Level Markup

Use the Level Markup section of the **Manager tab** (see *Level Markup*) to label important or interactive objects in the scene (e.g. enemies, collectibles). Tag any objects that would be indicated on a player's compass or minimap with the "Always Known" flag in the **Manager tab**'s Entity List.

Agent Set-Up

Adjust the motive sliders under the **Agent tab** to reflect the desired profile, or select a profile from the available presets and apply it to the agent (see *Agent Customization*).

Running the Simulation

Hit play to start the simulation and you can watch the agent navigate in real time. Select it in the hierarchy (or the **Agent tab**) to view an on screen overlay showing its targeting logic, mental map, player view, and health bar (see *Runtime Interface*).

Extras

You can run multiple agents automatically as part of a testing batch (see *Batch Simulation*) and record data for later review and visualization (see *Data Recording & Visualization*).

Project Set-Up

Demo Project

The included demo project and prefabs are set up to work “out of the box” - all you’ll have to do is hit the Bake button in Unity’s Navigation panel to re-bake the Navmesh after changing the level layout.

Will my project work with PathOS+?

Hopefully! If your game involves players moving around a 3D world, the answer is probably yes. As long as you can bake a Unity Navmesh for your scene, you can use PathOS+ to test your level designs. However, the tool has its limitations - be sure to read “A Few Caveats” below to see if the framework is a good fit for you.



Navmesh

PathOS+ Agents work with Unity’s Navmesh system for pathfinding. For the tool to work, you’ll need to bake your Navmesh from Window > AI > Navigation. If you’re starting from scratch, make sure that the baked agent settings (height, radius, etc.) match the settings on the Unity NavMeshAgent component of your PathOS+ Agent prefabs and GameObjects.

A Few Caveats

Tool Usage. PathOS+ is a tool primarily for simulating navigation, not complete gameplay. Agents’ navigation is dependent on the contextual information you provide - which GameObjects are collectibles, enemies, goals, and so on - but while there is some simulation of things like combat, agents don’t fully interact with your game’s mechanics.

Level Layout. Agents navigate in 3D, but their spatial logic is planar. This means PathOS+ works best when your level is mostly laid out at a uniform altitude with a defined ground plane - or when you can test your level in separate, mostly flat sections. PathOS+ will not work properly for levels with vertical layering.

Visibility. Whether or not agents can “see” game objects is based on Unity’s physics system, so anything you want to occlude visibility should have colliders attached. Visibility is calculated approximately, so don’t be surprised if what you see through the player’s POV camera doesn’t match up exactly with the agent’s logic.

PathOS+ Window

This window was created for the purpose of localizing every tool you'll need to set up levels and run tests. In order to open it, just navigate to *Window>PathOS+*. Here is a breakdown of the different tabs and what they each do.



SETUP.

In order to function properly, the window needs a reference to the agent/manager you're using in the scene. These references (along with the reference to a screenshot camera if you added one) can be set up in this tab.

Agent.

This is where all the values for the agent can be set (i.e. their personality) (See *Agent Personality*).

Resource Values.

The resource values can be edited in this tab. This includes how much damage each tier of enemy does, along with how much health the different potion types restore. (As an aside, how much damage the agent takes depends on their level of experience. If it's higher, they'll take less damage).

Batching.

This tab can be used to run tests with multiple agents at once (See *Batch Simulation*).

Profiles.

Agent personality profiles can be created and modified here.

Manager.

This is where all the general setup for the level can be set (such as the entities in the level) (See *Level Markup*).

Visualization.

This is where agent playthroughs can be saved and recorded, and where they can later be loaded in as things like heatmaps.

Expert Evaluation.

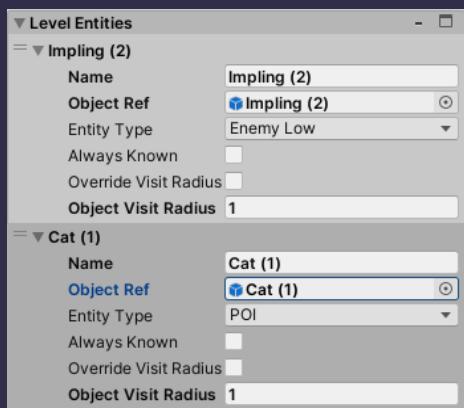
This tab is where expert evaluations can be recorded (see *Expert Evaluation*).

Level Markup

For agents to navigate in the context of your game, they need to understand which objects in the level can be interacted with, and what purpose they serve. To tag GameObjects, use the *Level Markup* section of the [Manager tab](#).

Makeup Brush

In the *Level Markup* section of the [Manager tab](#), you can click on one of the entity types to activate tagging for that type (your cursor will change). In the scene, click on objects to tag them with the selected entity type. You can also use this mode to change the tag on already labeled objects, or clear tags from labeled objects.

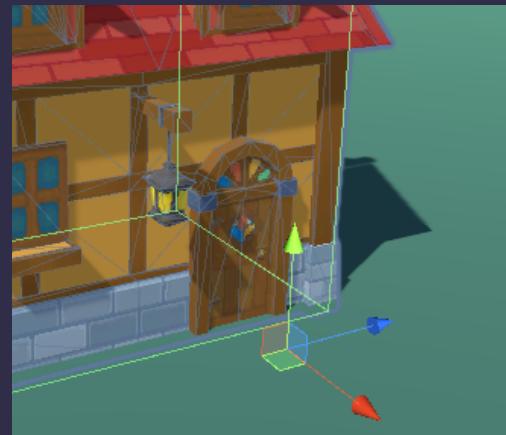


Entity List

You can also edit tags via the *Level Entity List* section in the [Manager tab](#). You can add and remove tags using the '+/-' buttons at the bottom of the list. Here, you can also change the GameObjects referenced by each tag, or set an object as “always known” (mimicking the effect of entities which would be indicated on a player’s minimap or compass).

A Note on Entity Locations

When agents target and visit game entities, they use the position of the Transform on the tagged GameObject. For large objects with colliders attached (e.g., buildings), a parent or proxy GameObject with the desired “visit location” should be used as the tag reference. The prefabs included in the demo project have already been set up with parent GameObjects with suitable pivot points chosen.



Game Entities

There are fourteen different tags available for level objects during the markup process. Here is a summary of their meaning. Type tags are used by agents to help drive their navigation through the game world.



Optional Goal.

In-game missions or objectives that are optional. (e.g., sidequest marker)



Mandatory Goal.

Objective that must be completed to finish the level. (e.g., main mission marker)



Final Goal.

Objective that would allow the player to complete/exit the level, if applicable.



Collectible.

Item that can be collected in game for achievement value. (e.g., treasure)



Self-Preservation Item (Low).

Item that can be collected to boost player survivability. (e.g., low health potion)



Self-Preservation Item (Med).

Item that can be collected to boost player survivability. (e.g., medium health potion)



Self-Preservation Item (High).

Item that can be collected to boost player survivability. (e.g., high health potion)



Enemy Hazard (Low).

Hazard that could result in a combat encounter if engaged. (e.g., low tier monster)



Enemy Hazard (Med).

Hazard that could result in a combat encounter if engaged. (e.g., medium tier monster)



Enemy Hazard (High).

Hazard that could result in a combat encounter if engaged. (e.g., high tier monster)



Enemy Hazard (Boss).

Boss entity that results in a combat encounter if engaged.



Environment Hazard.

Interactive hazard that agent can take damage from if engaged. (e.g., poison)



Point-of-Interest.

Environment landmark intended to draw in players for exploration. (e.g., setpieces)



NPC.

Non-hostile character that can be interacted with. (e.g., quest giver)



Runtime Interface



Note: During playmode, select the agent in the hierarchy or select the [Agent tab](#) to enable the PathOS Agent UI. Unchecking the “3D Gizmos” option in Unity is recommended.

Controls



Spacebar.

Toggle on-screen legend for mental map and Gizmos.



Click and Drag.

Pan the PathOS World Camera (if present).



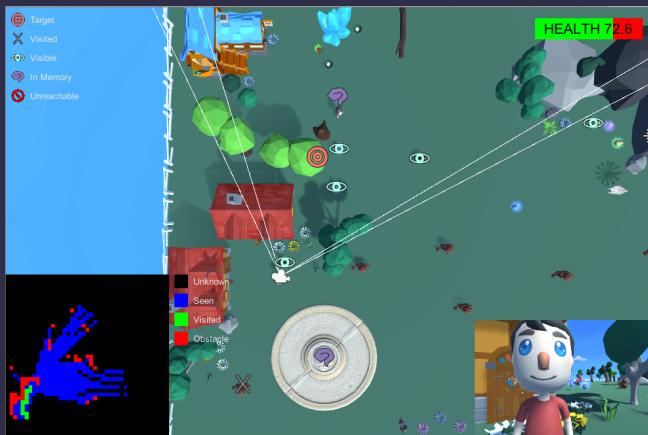
Mouse wheel.

Zoom in/zoom out with the PathOS World Camera (if present).



Right click (in edit mode).

Adds a comment to the expert evaluation tab with gameobject/entity of selected object (if open).



Mental Map

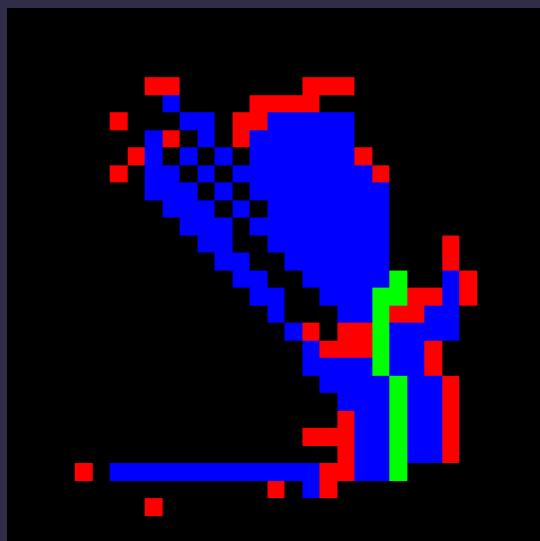
The mental map shows the agent's internal, tile-based representation of the scene's spatial layout. Four colors are used to indicate the state of each tile as perceived by the agent:

Black. Unknown

Blue. Seen as unobstructed (e.g., flat ground)

Red. Seen as obstructed (e.g. wall)

Green. Visited/traversed by the agent





Player View

The player view displays what the agent is currently “seeing” through its player POV camera. This can be used to double-check visibility of game objects outside the approximate system used by agents.

Entity Gizmos

For all game entities tagged using the markup system, Gizmos are displayed indicating how they factor into the agent’s logic at any given time. The meaning of these gizmos is as follows:



[NO ICON] Entity is not affecting agent logic - it is not visible, remembered, or previously visited.



Currently targeted by the agent. Can be applied to a game entity, or an empty point in the scene (while the agent is exploring).



Entity is contained in the agent’s memory.



Entity has been previously visited.



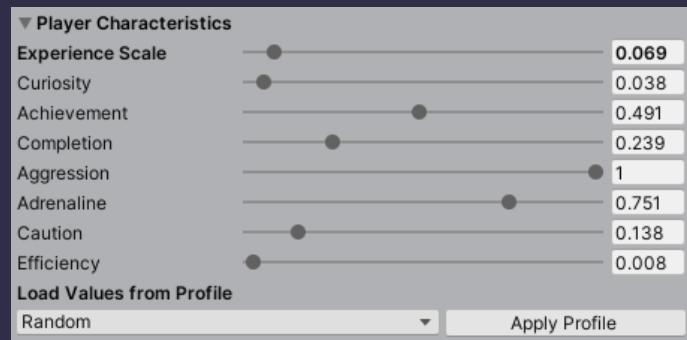
Entity is currently visible to the agent.



Entity has been determined to be unreachable (the agent cannot navigate to it using the Navmesh).

Agent Customization

Agents are governed by their motives, which reflect player motivations and can be customized for each agent. To change an agent's behavior, you can use the *Player Characteristics* section of the **Agent** tab. Here you can tweak individual motives, or apply a custom profile preset (see below).



Motives

There are seven agent motives in addition to the experience scale (e.g., amount of prior game experience). These motives affect the agent's behavior and affect the way it will evaluate tagged entities as potential destinations. These motives are as follows:

- Curiosity.*** The motivation to explore for exploration's sake, and discover all a level has to offer.
- Achievement.*** Wanting to earn achievements, complete game objectives, and rack up a high score.
- Completion.*** The desire to complete every in-game log, find every collectible, and so on.
- Aggression.*** A drive to seek out conflict and combat, dominating the game world.
- Adrenaline.*** Thrill-seeking, not only in combat, but in besting challenges or environmental gauntlets
- Caution.*** Taking care to maximize survivability, avoiding combat and hoarding resources.
- Efficiency.*** Wanting to get through a level as quickly as possible, prioritizing necessary goals.

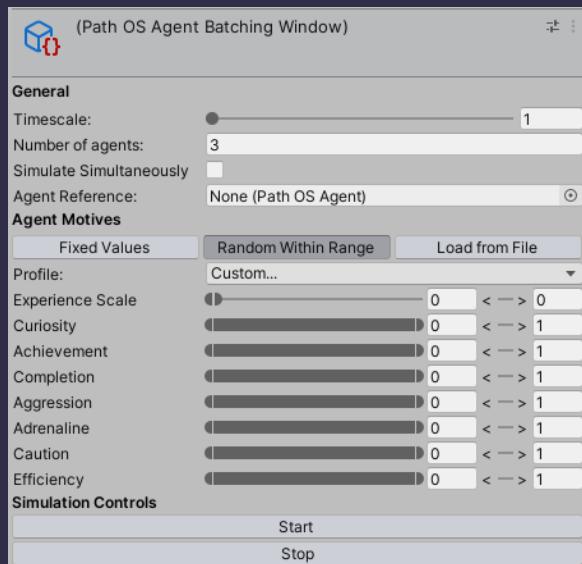
For advanced users, you can view and edit the relationship between these motives and level markup tags in the *Motive Weights* section of the **Manager** tab. Positive weights between a motive and entity tag indicate that an agent with a high value for that motive will be drawn to entities of that type. Negative weights will cause repulsion, and a zero weighting indicates no effect of the chosen motive on the agent's behavior around entities of the chosen type.

Custom Profiles

To manage agent profiles, go to the **Profiles tab**. From here, you can create and edit profiles, as well as loading or saving files to carry profile sets between projects. Each profile has a range defined for the seven agent motives, as well as experience. When a profile is applied to an agent, values are picked randomly from these ranges



Batch Simulation



To simulate multiple agents automatically, go to the **Batching tab**. From here, you can choose to simulate agents simultaneously. All you have to do is specify a prefab for the system to instantiate for each agent needed, as well as a starting position for agents in world space.

The number of agents active at once is capped at 8. Any number greater than this will be divided into batches of 8 or less at a time, and automatically run in sequence.



Note: You can also adjust the timescale of the simulation to speed things up - note that any time limit set in the **Manager tab** will proceed in real time, however.

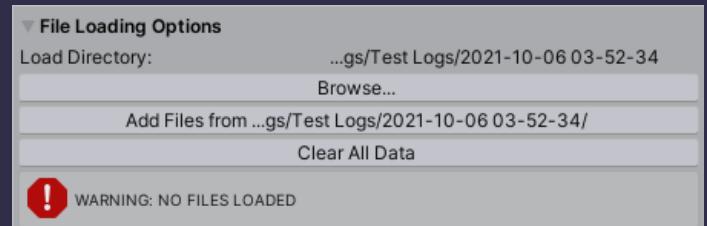
Agent motives will be initialized automatically based on the settings specified - either using fixed values, randomizing them within a range, or loading their values from a file. For range initialization, a custom profile (see above) can be selected to automatically define ranges. For file initialization, a .CSV file should be specified containing values for each of the desired agents.

Data Recording and Visualization

To automatically record logs containing information on agents' navigation and visiting level entities, toggle the “Enable Logging” setting in the [Visualization tab](#). This will record logs both if playmode is triggered manually, or if simulation is handled en masse through the agent batching window.

Loading Data Logs

Load logs through the [Visualization tab](#). Logs are stored as CSV files. To load them, select the directory where logs are located and hit “Add Files from...”.



Display Filters



It's possible to set the timescale of the loaded data. You can also control what data is visible. The “display height” controls at what altitude (in units) visualization elements will be rendered in the scene. You can also choose which agents should be included or excluded from the visualization. The profiles of the agents used to create the data can be viewed by clicking the ellipsis next to each agent name.

Viewing Agent Paths

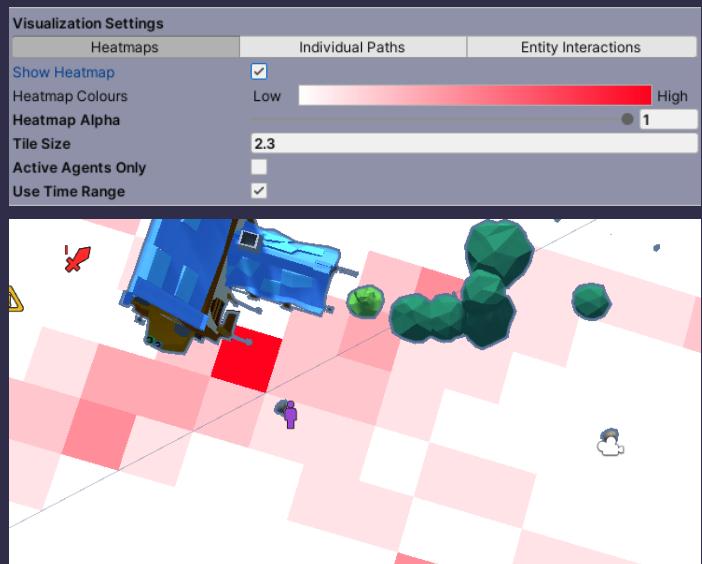
Individual agent paths through the world can be toggled, along with directional arrows for the paths. Enabling “individual interactions” will also show a record of individual agents visiting level entities during their traversal. From here, you can also specify colors for displaying each agent’s trajectory. If you hover over a part of the agent’s path, you can see what time the agent was at that location, and their corresponding health.



Heatmaps

To use the heatmap functionality, make sure that “Show Heatmap” is toggled after you load the files in.

The Heatmap tab can be used for customization (e.g., color scheme). The “tile size” attribute can be used to adjust the heatmap granularity. If “active agents only” is enabled, only data from agents enabled in the Filtering section will be used. If “use time range” is enabled, only data from the time range specified will be used.



Viewing Agent Interactions

In the Entity Interactions section, you can visualize how many agents visited different level entities in the scene. Each entity is shown as a circle, with scale and color adjusted to reflect the proportion of agents which visited each entity.

If “active agents only” is enabled, only data from agents enabled in the Filtering section will be used. Circles will be scaled according to the number of total agents in the enabled group. If disabled, data from all agent logs loaded will be used and scaled according to the total number of logs loaded.

If “use time range” is enabled, only data from the range in the Filtering section will be used (i.e., entities visited outside this time range are excluded from the visualization).

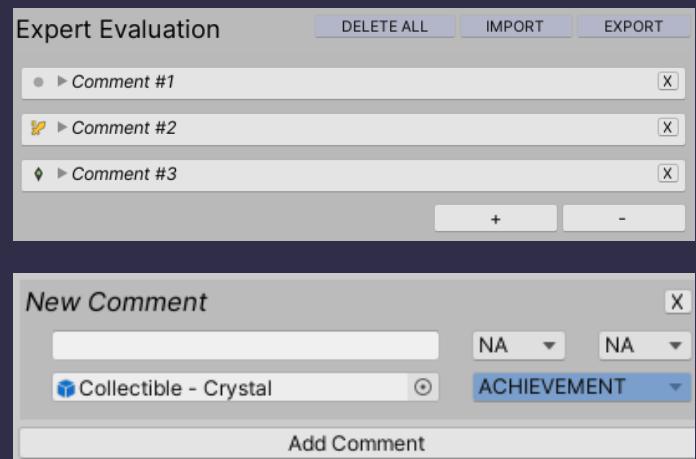
Expert Evaluation

This tab can be used to record your observations as you conduct an expert evaluation.

Adding/Deleting Comments

Comments can be added by clicking the plus button, and can likewise be deleted by pressing the minus button, or the X attached to each comment.

If the PathOS+ manager reference is set up, you can right click an object in the scene. A popup will show up that allows you to add a comment which has the corresponding gameobject and entity name associated with it.



For each comment, there is a textfield where you can write down your thoughts. Each one can be rated as **POS** or **NEG**, and be given a **LOW**, **MED**, or **HIGH** severity. The **GameObject** reference and entity type can be manually set for each comment. You can also choose to leave all these fields blank if you so choose.

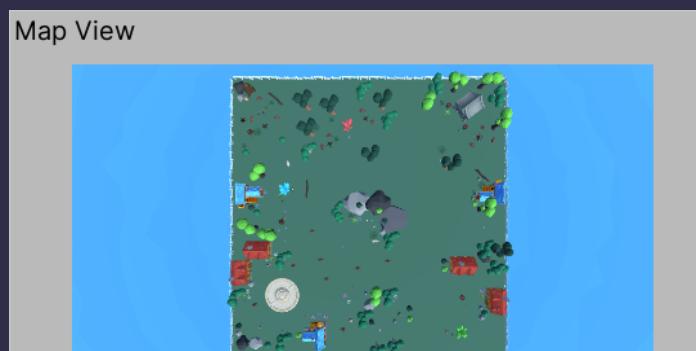
Severity	NONE	HIGH ENEMY	MANDATORY GOAL		
None					
Low	#1	#2			
Med					
High		#3			
#	Description	Severity	Category	GameObject	Object
1	This is a comment.	LOW	NEG	No GameObject	NA
2	This is an imp. The comment has the entity type and gameObject because I right clicked in the scene.	LOW	NEG	No GameObject	NA
3	This is another comment/ representative of my evaluations.	HIGH	POS	No GameObject	NA
					MANDATORY GOAL

Import/Export

Evaluations can be exported or imported as a formatted .CSV file. This file lists all the different comments, and also creates a table that sorts the comments based on their entity type and severity.

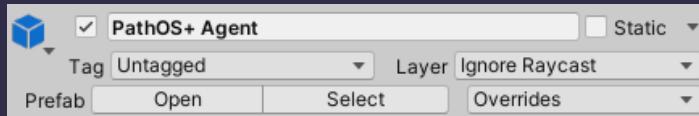
Map View

If the screenshot manager reference is set, it's possible to see a real time birds-eye-view map view of the level. This view will also be exported as a .png alongside the evaluation .CSV file.



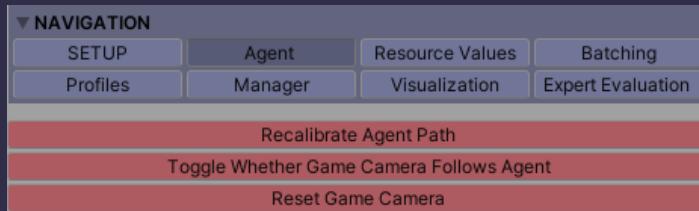
Troubleshooting

1. The simulation starts, then abruptly stops



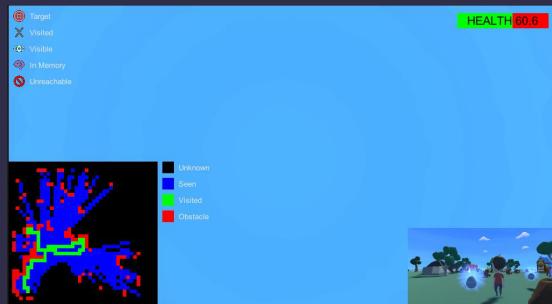
Make sure to have the PathOS+ agent enabled in the inspector.

2. The PathOS+ Agent got stuck



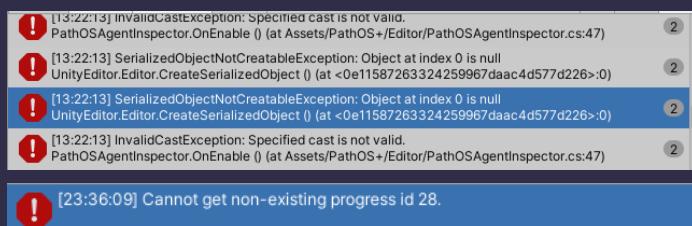
If, during runtime, the PathOS+ agent gets stuck at any point on the map, under the **Agent tab** there should be a button you can click that says “Recalibrate Agent Path”. This gets them unstuck. You can also press “Toggle Whether Game Camera Follows Agent” if you want the camera to continuously follow the play.

3. The game camera isn't in the right spot



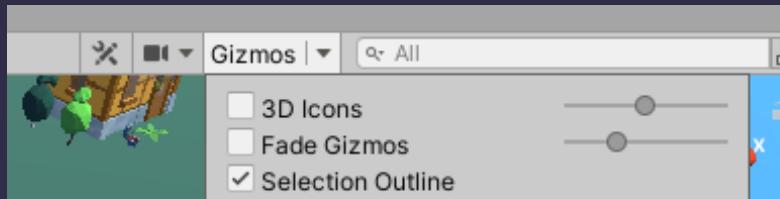
If, during runtime, the camera becomes displaced and you are no longer able to view where the agent is, you can go to the **Agent tab** and select “Reset Game Camera”. This will reset the position of the camera to where the agent is located.

4. I keep getting these error messages



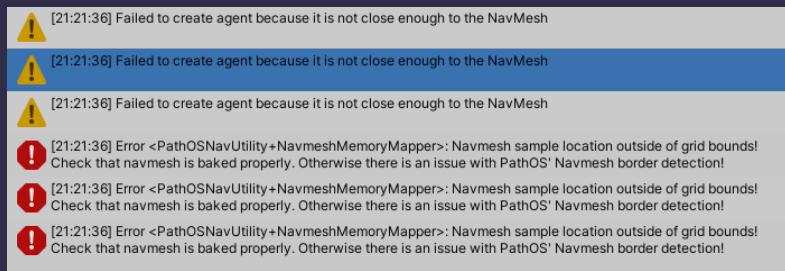
This is a glitch with the Unity engine. It won't affect what you're doing, but if the error messages are bothering you can restart Unity, or clear the messages out of the console.

5. The entity/runtime icons aren't visible



Make sure to have Gizmos enabled, and 3D icons disabled.

6. It doesn't let me run batch tests



If you're getting these errors, it means that the starting position for the batch agents aren't on the navmesh. Look at the position of the PathOS+ agent in the scene, and copy those coordinates to the X, Y, Z input fields in the **Batching Tab**.