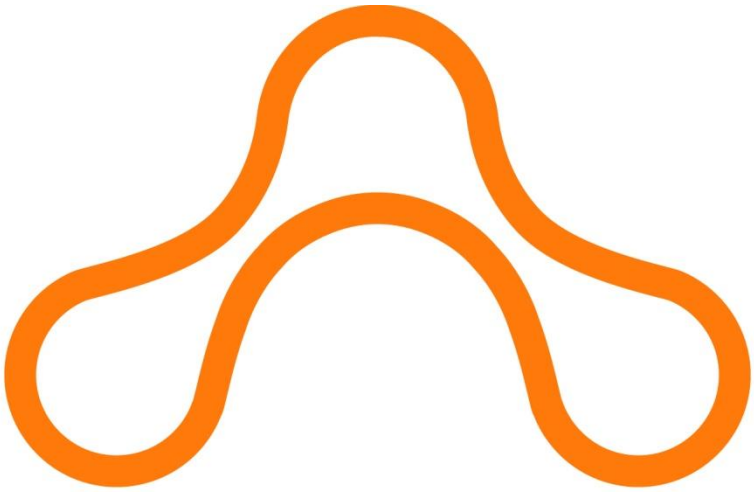


APEX PATH

User Manual

Contents

Version History

Version 1.0	Initial Version
Version 2.0	Updated for Apex Path 2.0
Version 2.1	Corrected dead links + info on Upgrading.

Quick Start

To get started quickly with Apex Path you can follow these quick instructions:

Import Apex Path into your Scene

1. Open Unity
2. If you do not already have a scene, create a new scene: *File -> New Scene* or press CTRL+N
3. Import the Apex Path package into your scene: *Assets->Import Package->Custom Package->Apex Path*.

Use Apex Path inside an existing Scene

1. Add the Component Apex -> *Quick Starts -> Navigating Unit with Selection* to the gameobject you want to navigate around the scene.
2. Add the following Layers “Terrain”, “Blocks”, and “Units”
3. Select the terrain gameobject the agents shall navigate on, and set the layer to “Terrain”
4. Select the GameWorld object, and locate the Layer Mapping component in the Inspector
5. Set the following three properties:
    a. “Static Obstacle Layer” = “Blocks”
    b. “Terrain Layer” = “Terrain”
    c. “Unit Layer” = “Units”

You are now ready to test the scene

1. Press play
2. Hold down left mouse button and select the unit
3. Right click to set the unit destination
4. Hold shift and right click to set waypoints

Upgrading

When upgrading Apex Path from one version to the next there will often be changes that require updates to the scenes and prefabs in your project.

For this purpose we provide the Upgrade tool available from the Tools -> Apex – Upgrade menu.

Please note that you may need to fix any compile errors before the tool can run.

Should you still encounter issues after upgrading, you should do a clean install.

- Remove Apex Path (with the possible exception of the settings asset in the Editor/Data folder).

- Reimport the new version.

User Manual

Introduction

Apex Path is an easy and fast solution for pathfinding in games using the Unity3d engine. Apex Path provides a solid pathfinding solution for any genre and any platform. It includes dynamic pathfinding, steering in changing game worlds.

The user manual is an introduction to the main concepts of Apex Path. Apex Path ships with many Examples, Tutorials and basic prefabs to get you started right away. On this website you can find the video tutorials and contained in the package you will find the full API documentation.

If you need to get started quickly, please visit the Quick Start guide.

If you have any further questions do not hesitate to contact Apex Game Tools at support@apexgametools.com. We want to make it easy to create games and we appreciate any feedback.

How Apex Path Works

Main Concepts

Apex Path is a multi-threaded path finding engine based on the A* path finding algorithm (basic A* and Jump Point Search). You can find plenty of information about the A* algorithm from books, academic articles, and on Wikipedia. We won't delve with the mathematics behind efficient pathfinding or the computational challenges of creating efficient pathfinding libraries in this manual. Instead, we will provide a general introduction to the main concept of pathfinding and how Apex Path deals with pathfinding to make great games.

Path finding allows characters – or agents – in your game to find their way around obstacles, up and down ledges, through buildings and mazes. Depending on the size and complexity of the game world, scene or level, pathfinding can be both complex and tasking. In addition, several factors add to the complexity of robust and well-performing pathfinding. Especially dynamic and changing factors on the game level can be difficult to manage when it comes to pathfinding. Many agents on the scene, doors opening and closing, platforms shifting, or walls and other obstacles being destroyed or changing place.

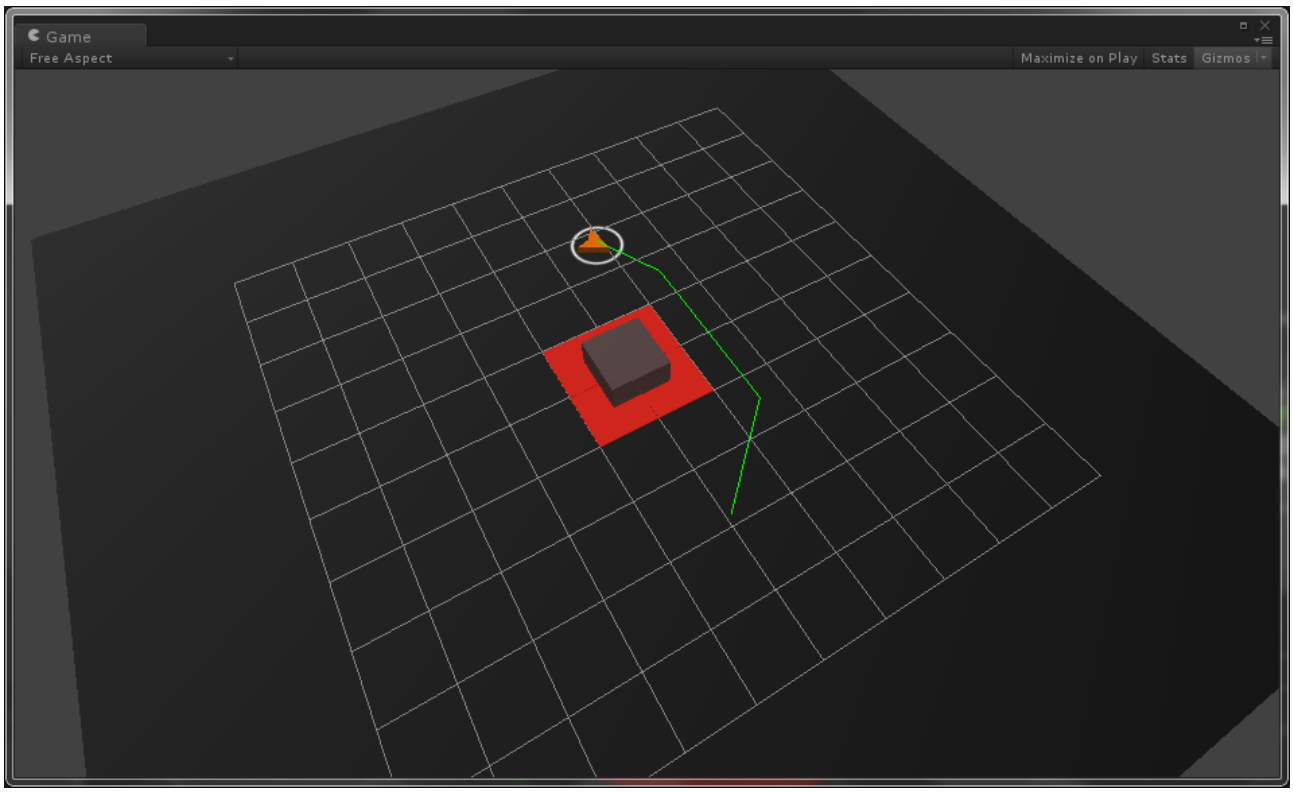
Apex Path is especially designed to manage dynamic levels and have great performance even when many agents are active in the scene. However, first and foremost Apex Path is designed to make it easy for the game designers to handle complex pathfinding scenarios with ease, to allow the game designer to focus on what really matters in the game.

If you are new to pathfinding, the user manual guides you through the main concepts of pathfinding in the following sections.

The main gameobjects you will work with are the agent, ground, path, and obstacle. The following paragraphs outlines from pathfinding in Apex Path works and describes the main components used in the pathfinding engine.

The screenshot shows the agent (in orange) navigating the grid (in white) around the obstacle (in grey). The obstacle blocks part of the grid (in red). The agent has received a path (visualised as a bright green

line). The path takes the agent safely around the obstacle without walking into it and without getting stuck.



Agent

The agent is the character in the game that should navigate based on user input or input from the AI in the case of computer controlled characters. When an agent gets an input from the user through the Basic InputReceiver, the agent navigates the game world by moving in a direction towards the goal using the Steerable Unit Component with a certain velocity based on the agent's speed setting in the Humanoid Speed Component. However, the game world can be filled with obstacles such as trees, buildings, and rocks that the agent needs to navigate around. To do this the agent needs pathfinding.

Ground

The Ground (sometimes called the Terrain) is the world geometry that makes up the game world. This could be a dungeon, a forest or a city which the characters of the game travel. As a basis the ground is walkable, meaning that unless the geometry is marked as an obstacle the agent assumes the ground is walkable. There might exceptions to this such as swamps in a forest, a heavily trafficked street inside a city, or an area with magic fire inside a dungeon. The grid Component is responsible for making a discrete grid of cells on top of the world geometry. This grid is used by the Path Service to create paths through the game world that agents can traverse.

Path

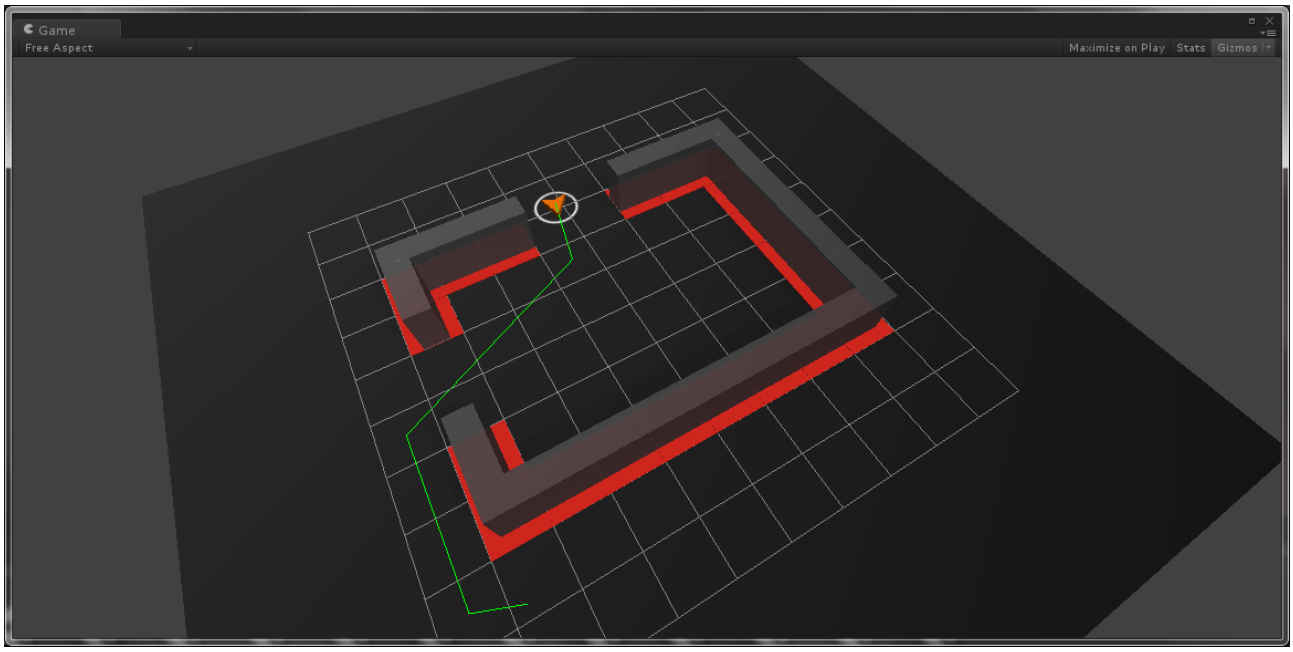
When an agent is instructed to navigate to a certain point in the game world, the agent sends a request to the Path Service component. The Path Service component returns a path to the agent. The path is a list of positions in the game world. By moving towards the next position in the list one-by-one the agent can find its way around obstacles toward the end goal.

Obstacles

Obstacles are geometry in the game world that the agent cannot walk on and must not walk into.

Most obstacles in the game world might be buildings, trees or rocks that are not expected to change position throughout the game. These are called static obstacles and are put in the static obstacle layer in the Layer Mapping component on the GameWorld gameobject in the Hierarchy.

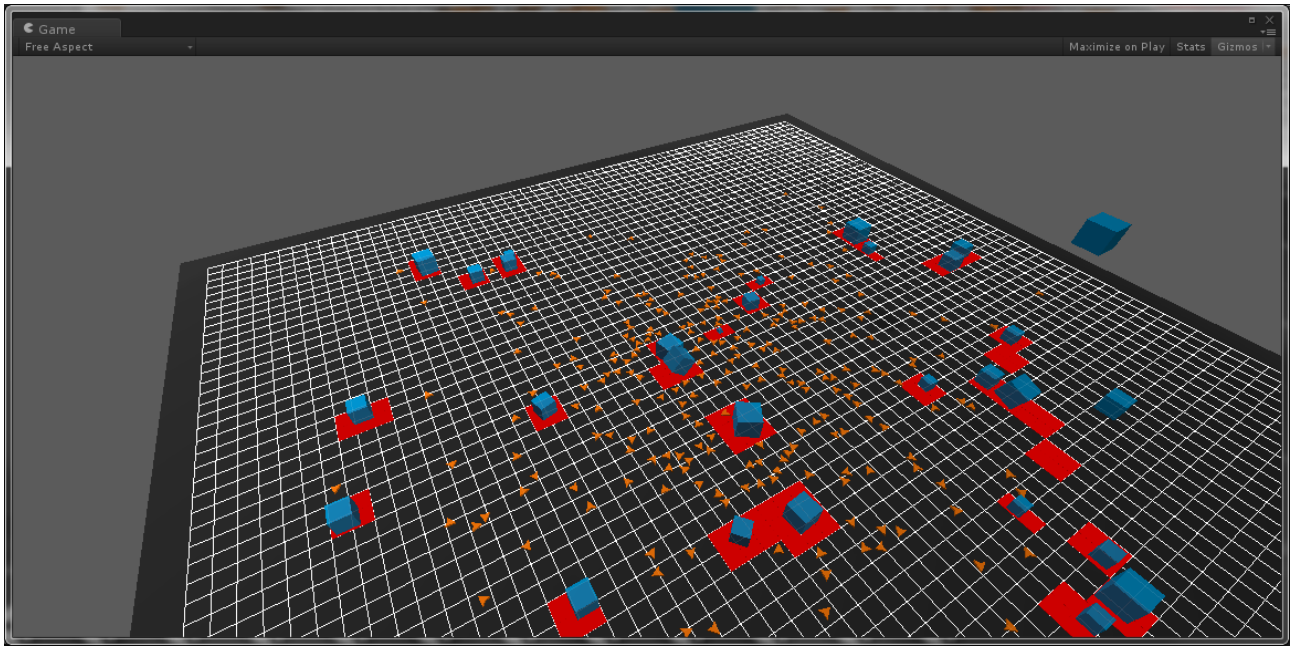
Alternatively all obstacle geometry can be added to the Terrain layer instead and use the height map settings of the grid to determine what is and is not walkable.



The walls (in grey) of the building are blocking (in red) several cells in the grid. The path (visualised by a bright green line) correctly takes the agent (in orange) safely through the building without walking into any walls.

Obstacles that can change their position dynamically through the game are called dynamic obstacles. These obstacles are typically doors, walls coming down, or debris. Dynamic obstacles have the Dynamic Obstacle Component added to them. This ensures that Apex Path takes them into consideration when agents are pathfinding around them.

The agents (in orange) are in a game world with multiple dynamic obstacles (in blue) falling out of the sky. The agents carefully avoid the cells in the grid that are blocked (in red).



Attributes

Apex Path comes with a concept called attributes. Attributes are special tags that can be assigned to units that allow other entities such as dynamic obstacles to interact in different ways with different units. This is a very powerful tool to allow for advanced functionality in the pathfinding.

Example:

Some units in your game might be allowed to walk through certain doors while others are not. This could for example be in cases where certain units have access cards, while other units do not. Set an Attribute on the Unit component as "On Blue Team" and set the Exceptions property on a Dynamic Obstacle to the same Attribute "On Blue Team". The Unit will now ignore the Dynamic Obstacle when it searches for a path.

Apex Path comes with several example scenes that show how to set up and use attributes to create doors that open and close, moving platforms and other dynamic mechanics for your game. Apex Path also includes a specific tutorial on attributes.

Tutorials

The tutorials provide detailed walkthroughs of the main concepts in Apex Path.

Apex Path Tutorials

Examples

The examples illustrate how Apex Path is used in different scenarios.

Apex Path Examples

Glossary

Glossary