Integrated Path Planning Object (By Mokhtar M. Khorshid)

Version: 1.1 – released on 16 September 2010.

Contents

Features

- Integrates directly with active objects on your frame. Handles basic path planning almost automatically for you.
- Highly efficient. Uses path planning that are optimized for efficiency.
- Flexible. You decide how to use the extension. Either have it automatically handle pathfinding operations for you or manually modify the underlying grid.
- Supports customizable transitions. This means you can have different movement patterns for different units as well as units of different sizes.

Overview

The Integrated Path Planning extensions object aims to efficiently solve most pathfinding problems in games developed in MMF2. Pathfinding problems are those where you have one or more units (also called agents) that you want to navigate through a map to reach some target location while avoiding obstacles on their way. Planning a path means generating a complete plan that would take the agent from its start position all the way to the target position. Typically you want to find an optimal or near optimal path.

This extension was designed to be both efficient and simple to use while remaining reliable. The object also packs some advanced features for power users that allow more complex problems to be solved. In general there are two ways to use the object: direct object manipulation mode and grid mode. You can use both modes simultaneously on the same maps with no problems, there are even some hybrid functionalities.

Direct Object Manipulation Mode

This is the simplest and fastest way to get a pathfinder up and running. All you have to do is tell the extension object which of your active objects are agents and which are obstacles and it will take it from there. You can have the extension automatically orient the active objects in the right direction until their reach destinations at which time a quick condition will let you know that it's there.

This should be the typical mode used for simple situations where your physical world matches the virtual pathfinding world and you don't have a lot of tweaking to do.

Grid Mode

Sometimes, your situation is more delicate and can't be solved automatically by relying on the positions of active objects. For example, if you're using an isometric view or have 3D objects that you can move around. In such cases, you will want to use the grid mode to work with the abstract grid regardless of where your physical objects are.

This mode is intended for advanced users as it takes more effort in setting up correctly, but if you have used any of the previous pathfinding objects (Grid object, Pathfinding object, Advanced Path Finding object, or War Game Map object) then you should already be familiar with the way it works.

Custom Transitions

When enabled in the properties, the custom transitions allow you to control how the agents move. A transitions is composed of a number of conditions and a move. When all the conditions are met, the move is allowed. In this extension the condition is a check that a specified tile is vacant.

When dealing with transitions, we specify tiles using offsets. So (-1, 0) refers to the tile that is one tile to the left of the agent.

Custom transitions can be used to have non-typical moves as well as guard against specific moves. One important application for custom transitions is handling agents of different sizes.

Note that custom transitions when enabled will slow down the searches and prevent advanced heuristics from being used. So only enable custom transitions when you need them. Once enabled in the extension properties, the relevant menus will be accessible in the event editor.

Actions

Add Agent (Agent)
Adds the specified active object to the system as an agent that can navigate through the map.

Remove Agent (Agent)
Removes the specified agent from the system. Do NOT attempt to use the object as an agent after using this action.

Look at next path node (Agent)
Changes the direction of the specified agent to look towards its next node on the path to its destination. The active object representing the agent should already have a movement for this to work.

Add Obstacle (Obstacle, Weight)
Adds the specified active object as an obstacle. The weight parameters should be set to 100. This parameter does not represent an actual weight, and will probably be renamed one day.

When you add an obstacle its current position is marked as occupied on the map, but if the active object moves, the system is not automatically updated to use the new location, however the obstacle objects are tracked internally and will be refreshed whenever you ask them to. For this you need to use the Update Obstacles action.

Remove Obstacle (Obstacle)
Causes the obstacle's active object to no longer be tracked as an obstacle. Note that this has no immediate effect, unless you use the Update Obstacles action, nothing will change.

Update Obstacles ()
If your obstacles moved around the frame and you want to update the system, use this action. It will update the internal obstacles based on the active objects of all the obstacles you added.

Plan Path for Single Agent (Agent)
Will find a path for the specified agent to reach its destination. Make sure that the agent has a destination before using this action by using the Set Destination action.

Plan Paths for All Agents ()
As the name implies, this action will plan paths for all agents in the system to their specified destinations. Make sure that the agents have already been assigned destinations using the Set Destination action.

Set Destination (Agent, X, Y)
Sets the destination/target of the specified agent to (X, Y). The coordinates are pixel positions on the frame (not on the internal grid).

Jump to Next Path Node (Agent)

This action teleports the agent to the next node of its path. The extension automatically detects when an agent has reached a path node and adjusts the remaining nodes. A path node is basically a tile on the internal grid, whose size is specified by the properties of the extension.

Clear Existing Profile Transitions ()

Removes all transitions for the currently active profile. Note that without transitions agents will be unable to move at all. Use this action when you want to overwrite the current profile with new transitions.

Add Single Condition (Delta X, Delta Y)

Adds a condition to the current transition in construction. Conditions are created first then used up to create a transition. The condition mainly checks that the tile at position (Agent X + Delta X, Agent Y + Delta Y) is vacant.

Add OR condition (Delta X1, Delta Y1, Delta X2, Delta Y2)

This is similar to Add Single Condition except that the condition is satisfied if either of the two positions (Agent X + Delta X1, Agent Y + Delta Y1) or (Agent X + Delta X2, Agent Y + Delta Y2) is vacant. This kind of condition is useful when you want to allow diagonal moves for agents but prevent them from squeezing through two diagonally touching obstacles, in which case you need at least one adjacent tile to be vacant to make the diagonal move.

Add Transition to System (Delta X, Delta Y)

Adds a possible transition (move) that is guarded by all the conditions specified up to this point since the last call to this action. The transition is a move that takes the agent to position (Agent X + Delta X, Agent Y + Delta Y). This action will also clear all the conditions in preparation of creating a new transition.

Set Current Profile (Profile ID)

Switches to the specified transitions profile, creating it if it doesn't exist. All subsequent searches will use this profile unless the agents have their own overriding profiles.

Assign Overriding Profile (Agent, Profile ID)

If your agents have different ways of moving around the map, then you would use this action to override the default transitions profile for individual agents. This action will set the specified agent's profile to the specified profile. As long as an agent has an overriding profile, it will always use that profile regardless of the currently selected profile selected with Set Current Profile.

Copy Profile (Source Profile ID, Target Profile ID)

Copies all the transitions of the source profile over to the target profile, replacing its current set of transitions.

Set Tile Size (Tile Size, Resize Grid Flag)

Sets the size of the individual tiles in the grid. The tiles are always squared and the supplied size is in pixels.

If the Resize Grid parameter is set to 1, then the grid will be resized as well to cover the whole frame with the new tile size. Note that in this case it becomes a very expensive operation and reconstructs the map invalidating everything on it. Use this only at the beginning of the frame while setting up the extension.

Set Grid Size (Width, Height)

Resizes the internal grid to the specified dimensions in tiles (NOT pixels). This is a very expensive operation and invalidates everything on the grid. Use it only at the beginning of the frame while setting up the extension.

Set Tile Weight (X, Y, Weight)

Sets the cost of the specified grid tile to the specified weight. You should supply 0 if the tile is vacant or 100 if it is an obstacle. Note that this is not a true cost which means the extension will not favour low costs over higher costs if you supply any other values in the middle. The meaning of weight is different here and it defines the "type" of the tile, but we are using the terms "weight" and "cost" here for simplicity.

Set Zone Weight (X1, X2, Y1, Y2, Weight)

Identical to Set Tile Weight but works for a whole rectangular zone between (X1, X2) and (Y1, Y2) inclusive instead of a single tile.

Add Agent to Group (Agent, Group ID)

Adds the specified agent to the group identified by the supplied group ID. This will remove the agent from its previous group, if any.

Remove Agent from Group (Agent)

Removes the agent from its current group.

Plan Paths for Group (Group ID)

Plans paths for all the agents belonging to the group to their individual destinations. In the "T0" builds of this extension object, the agents will not consider collisions with one another within the group.

Iterate Over All Agents ()

Starts a loop over all agents in the system. Use the On Agent condition to handle the iteration along with the Get Current Agent ID expression.

Iterate Over Agent's Path (Agent)

Starts a loop over the nodes in the specified agent's path nodes. Use the On Path Node condition to handle the iteration along with the Get Current Path Node X/Y expressions.

Iterate Over Quick Path (Start X, Start Y, Destination X, Destination Y)

Finds a path between (Start X, Start Y) and (Destination X, Destination Y) and iterates over the nodes in this path. Use the On Path Node condition to handle the iteration along with the Get Current Path Node X/Y expressions.

Conditions

Is Agent at Destination (Agent)
Returns true while the specified agent is at its destination.

Is Agent NOT at Destination (Agent)
Returns true while the specified agent is NOT at its destination. This condition is there due to a bug in the way MMF handles the negation of conditions with object parameters.

Agent Wants to Move (Agent)
Returns true if the agent has a destination, a path, and wants to move.

Agent Wants to Stop (Agent)
Returns true if the agent has nowhere to go.

Can Agent Reach Its Destination (Agent)
This can only be used after planning a path for the agent. It returns true if the agent has a complete path to its destination and fails if it has no way to reach its destination.

Is Object an Agent (Object)
Returns true if the specified object is an agent in the system.

Is Object NOT an Agent (Object)
Returns true if the specified object is NOT an agent in the system. This condition is there due to a bug in the way MMF handles the negation of conditions with object parameters.

On Agent ()
Triggered once for each iteration of loops that loop over agents. Use the Get Current Agent ID to get the current iteration's agent.

On Path Node ()
Triggered once for each iteration of loops that loop over path nodes. Use the Get Current Path Node X/Y expressions to get the position of the current iteration's node.

Expressions

Get Distance to Goal (Fixed value of Agent)
Returns the number of nodes on the specified agent's path. This is NOT the physical distance on the screen in pixels.

Get Agent Current X/Y Tile (Fixed value of Agent)
Returns the current tile's X or Y coordinate of the grid tile that the specified agent is occupying. Note that these are grid coordinates not pixel coordinates.

Get Agent Target X/Y Tile (Fixed value of Agent)
Returns the current tile's X or Y coordinate of the grid tile that the specified agent is about to move it (i.e. the next node on its path). Note that these are grid coordinates not pixel coordinates.

Get Tile Size ()
Returns the size of the grid tiles in pixels. Note that all tiles are squared and of equal sizes.

Get Grid Width ()
Returns the grid width in tiles.

Get Grid Height ()
Returns the grid height in tiles.

Get Tile Weight (X, Y)
Returns the weight of the specified tile. A value of 0 means the tile is vacant/traversable while a value of 100 means its occupied by an obstacle.

Get Last Quick Path Time ()
Returns the number of milliseconds spent on finding the last quick path.

Get Current Iteration Agent ID ()
Returns the ID (fixed value) of the agent involved in the current iteration of loops over agents.

Get Current Path Node X/Y ()
Returns the X or Y coordinate of the node involved in current iteration of loops over path nodes.

Known Issues

- Do not combine individual-object actions (such as "Add Agent" or "Set Destination") with aggregate actions like "Plan Paths for All Agents" since MMF will execute a single individual-object action along with the aggregate action once and then repeat the individual-object actions only for the remaining ones. For example, if you use set destination then plan paths, then only one of the agents will have a destination by the time plan paths is called.

Credits

Designed and Developed by Mokhtar M. Khorshid for Clickteam.

This extension includes code adapted from the open source HOG2 system by Dr. Nathan Sturtevant.

Special thanks to Ross and Dynasoft for their help with direct object manipulations.

Thanks to Pharanygitis for creating the logo and extension icon.

Support

Depending on how this extension object is distributed you can request support in a number of ways:

- Ask questions or request features on Clickteam's official forums.
- If you have access to the issue tracker, report bugs there.
- If you don't get a response in time use this link.