1 Implement formation motion as a separate (scripted) component

A formation needs different paths than a regular unit. Formations can go over small obstacles, and let their units walk around it. While formations should avoid to walk right next to big obstacles, which causes half of their . This proposal aims to fullfill the special formation movement needs.

1.1 The aim of formation pathfinding

A formation is basically a rectangle, with a depth and a width. Not all terrain under a formation should be passable, but our aim is to keep as much terrain passable, so most units in formations can walk without bumping into obstructions.

Next to that, turning formations is also a difficult case. The turn should be taken slowly to not let units run around like mads.

The formation path should however make sure that there's at least one way all units in the formation can use.

1.2 Getting the path

The formation should get a long-range path from the pathfinder. This happens by calling the ComputePathASync method of the pathfinder, and listening to the PathResult message. Both need to be made available for JS, which isn't hard (a path can easily be encoded as a simple array of waypoints).

A path will ony contain the waypoints where an entity has to take a turn.

1.2.1 Generalisation

After the path is received, it needs to be generalised to get rid of unneeded waypoints. If waypoints are closer to eash other than the formation size (probably best expressed as the minimum of the depth and width), then the average position of both waypoints should taken (unless one of the two was an endpoint, then the endpoint is taken). This can be recursively executed until all waypoints are at least the formation size away.

This way of generalisation is cheap, makes sure there's still a valid path nearby (if the original path was valid), and makes sure turns don't happen very often.

So an ideal formation path consists of long straight lines, and a few turns.

1.3 Following the path

1.3.1 Following a straight line

When following a straight line, the overall direction is kept and the formation keeps turned towards that direction. For each turn along the line, three possible positions are compared:

1. For the walking distance of one turn forward along the direction

2. For the walking distance of one turn sideways along the direction $+45^\circ$
3. For the walking distance of one turn sideways along the direction -45°

For both outer two positions, the distance to the original (generalised) path is compared. If the distance is bigger than the formation width, than the position is ignored.

For each of the remaining positions, the pathfinder is requested to do an obstructed-tile check in the formation rectangle. This check is similar to the foundation-placement check. But here it should return the number of obstructed tiles under the formation. If possible, the pathfinder should also take units (that don't belong to our formation) into account. The position with the least obstructed tiles, will be the target for the next turn. If two positions are equal, the path closest to the original path will be taken. If it's one of the sideway positions, then the formation as a whole will not turn, but units will turn in their position.

1.3.2 Detecting a corner

On every turn, you also need to detect whether you're starting a corner. This can be done with a simple projection of the formation position onto the straight line, and comparing if the projected point is still between the waypoints.

1.3.3 Taking a corner

When taking a corner, the centre of the formation will quite often not be on the corner point, but shifted a bit. The formation should turn as a block around the waypoint of the generalised path. Thus if the centre isn't at the waypoint, it should turn in an arc around it, rotating bit by bit.

The rotation speed should depend on the run speed of units, and the distance they have to cover (so it depends on how far from the centre it turns). When rotating, the formation shouldn't go forward.

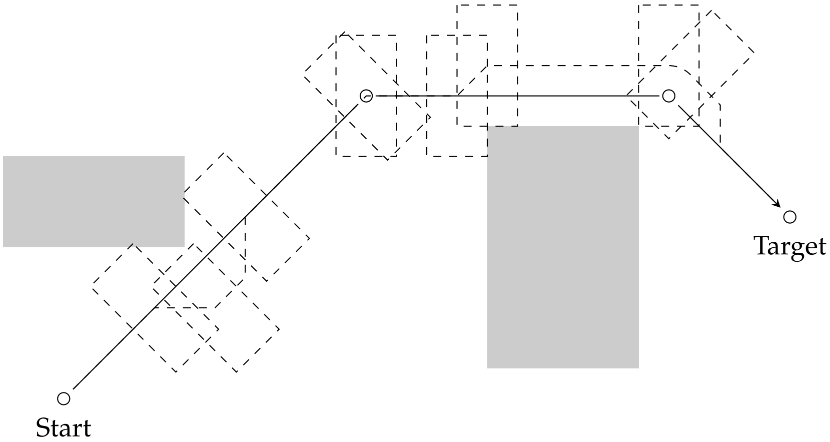
1.3.4 Finalising the path

When the distance to the last waypoint is smaller than the distance to the straight line you're following, then you're finalising your path. From that point onwards, you shouldn't go in a straight line to the last waypoint (while keeping the formation still oriented to the path direction), and stop comparing obstructions.

1.4 Schematic drawing

The image shows a path result from the long-range pathfinder (the solid line) passing by obstacles (the grey rectangles), together with the actual path a formation will follow (the dashed path), and how the formation will be rotated (the dashed rectangles).

You can clearly see that the path isn't optimal (you can clearly see the path deviating from the optimal path), but this isn't neccesary, as long as there's a good path nearby.


Start
Target