PathToggler Demo Level
for
Unreal Tournament
Info Notes

Intro Explanations:

Target use = Spectating ONE Bot, not for playing, not for servers unless it's one for working/testing.

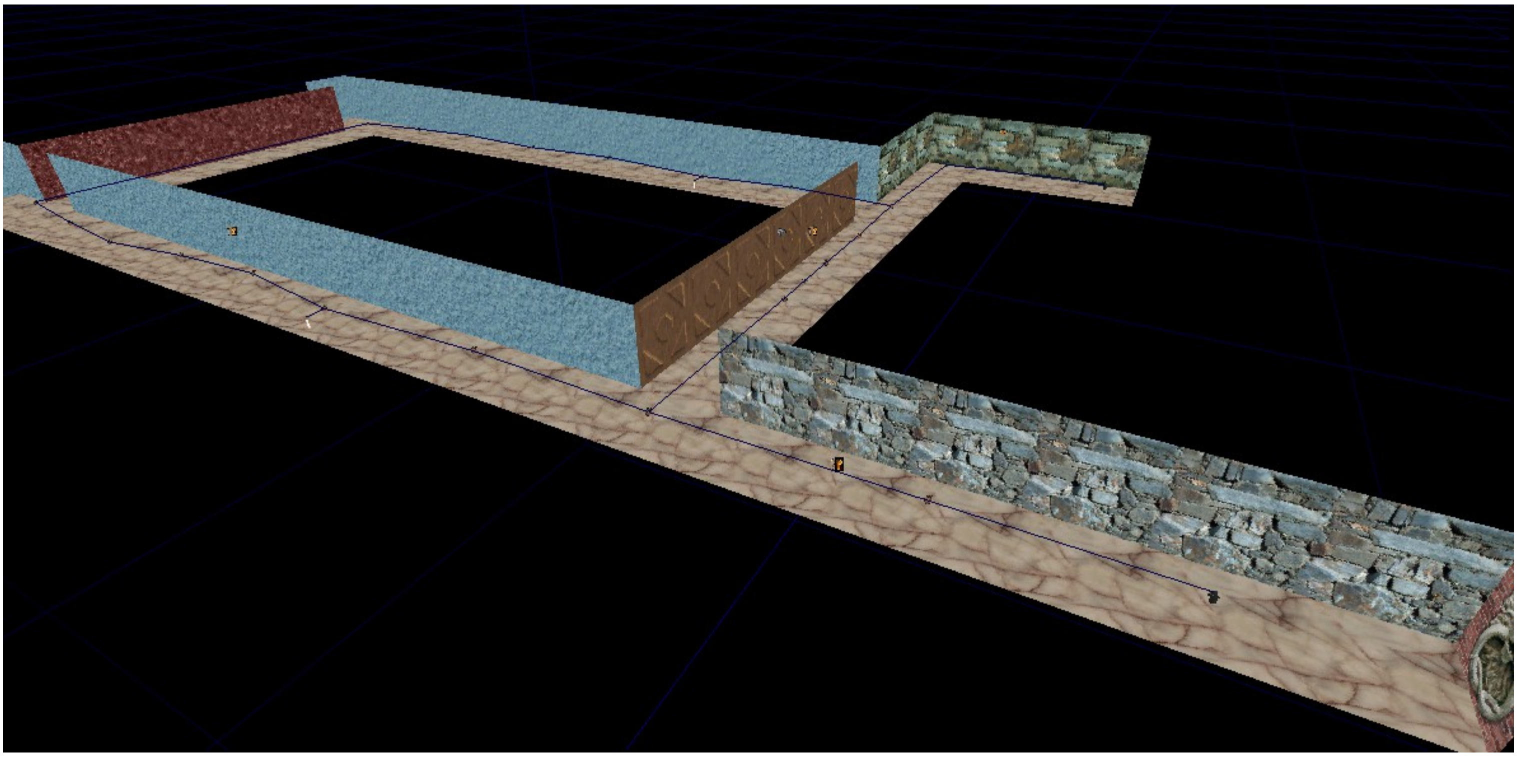
Audience = Curious mappers which want to see what's the deal in a flexible navigation network. If you aren't one of those, save you time by stopping to read this document and remove this stuff because it might be useless for you.

When we talk about the so-called Bot Support and build a smart navigation network capable of diverting the usual shortest routes that a Bot follows, we need not only the known or unknown BlockedPath that UT has in stock, we can use something smarter that is meant to do some things similar to what happens in the UT Assault game but which works in **almost** any circumstance. We say the word **almost** because we have to respect some conditions.

When we want to block/unblock a navigation point we change that ExtraCost with a high/low value - similar to BlockedPath this is blocked and will go unblocked. This value can be changed to zero by an Event on the map when we talk about BlockedPath and the path/route becomes clear at that point. However, there is also the possibility of raising the ExtraCost value of a navigation point and then we get the opposite effect - locking/blocking path.

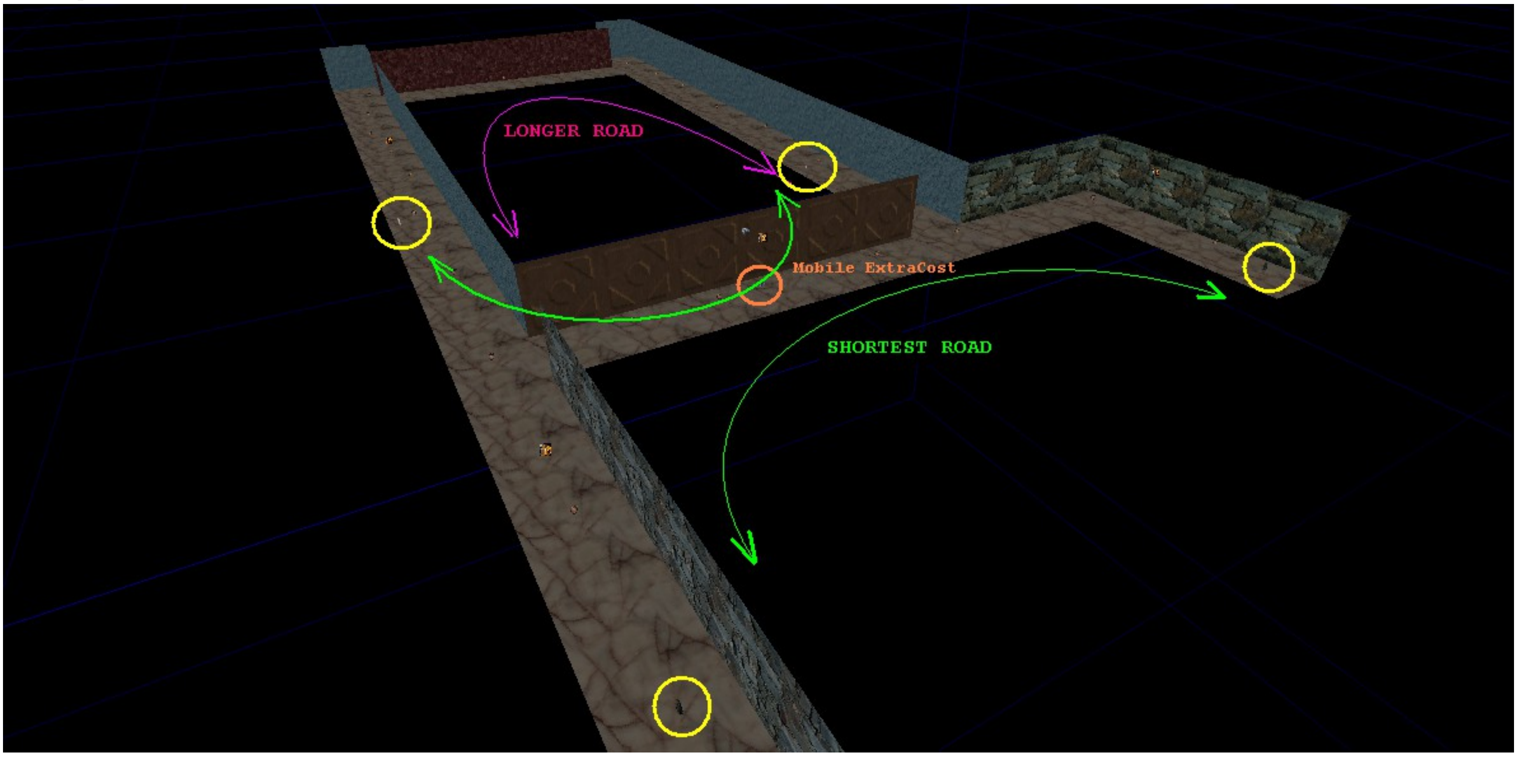
Let's see what to consider when using such flexible navigation points - such as PathToggler which is part of the Demo Map in the current archive. We take care that this navigation point does **not** have **shortcuts** passing over it. If this happens, the directives of our point are simply ignored and the route is still available, the Bot ignoring what we want him to do. In this demonstration map - short, simple, etc. We have navigational routes that go from one point to another without any shortcuts. In this case, if in the middle of the shortest route we act on the ExtraCost value, we block it by forcing the Bot to go on an alternative path.

The demonstration prototype looks like this:



More Explanations:

The Bot's movement is basically between four points: two HealthVials and two ThigPads. When the Bot has enough life and does not need HealthVials, it will continue to collect ThigPads moving from one item to another. The usual route would have been through the side marked with green, but here we have a "problem" from time to time: The shortest route is blocked somewhere and then the longest one becomes the route used.



When the route is blocked, the navigation point - which is visible here - will have the texture of the STOP traffic sign. When the path is free for access with ExtraCost = 0, the texture changes according to the ROAD PRIORITY sign. When we look at the Bot as a spectator from a position similar to the ScreenShot of the map, we see what the Bot does about the navigation mode when the flexible navigation point changes its appearance and ExtraCost value.

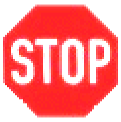
Misc:

Let's see what else might be noted. If we make a copy of the map and in that copy remove the name used by the TAG of the navigation point actor PathToggler, that StochasticTrigger will no longer instigate it through the Event. Also, if we delete it and replace it with a regular PathNode and rebuilding the navigation network, we can see that the Bot will no longer travel on another route, only the shortest one, with this we are demonstrating how PathToggler works and how it can be applied for the development of a smarter navigation network.

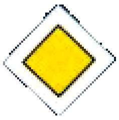
Bot Notes: Route goes blocked in Paths Processing task from Engine. As result, when Bot is moving and it's not processing and his MoverTarget is this PathToggler which in that moment is changing ExtraCost, Bot will still move there passing through the point. When PathToggler goes blocked behind Bot, again, this Bot will keep moving to whatever goal because said point is behind and no longer part of his paths - is history. When Bot goes to the alternate route and PathToggler unlocks paths, if Bot is still finding this way shorter "will change his mind" returning to this shortest route. When Bot goes to shortest route and a few points forward PathToggler increases ExtraCost blocking/locking path, Bot will return from his way changing route, following an alternate solution.

Bot can be spectated here more time and you'll figure how is moving and where is moving if PathToggler is changing the look - at once with ExtraCost.

Blocked means ExtraCost = 100000000 and texture is:



Free paths means ExtraCost = 0 and texture is:



Small Tutorial:

For preventing nasty shortcuts in raw Path Building solution from Editor and not 21th century specific toys you need to keep in mind some basic but important things:

- a PathToggler or other actor accesing a PathNode must have a visual contact with some three (or two) navigation points, but those points should not get connected each-other, only with PathToggler - preventing shortcuting through more or less PrunedPaths;
- if nodes nearby PathsToggler have some Prunedpath[x] shown, if these might be linked each-other I'm not sure if Path Toggling or PathToggler will have effect;
- you can set this also combined with One-Way type paths which are easy to do now days if a MapGarbage builder tool it's used in map's editing process.

Cute Add-ons:

- XC_EditorAdds from XC_Engine_V24 will build paths more easy without or with less shortcuts and then acting over a default PathNode will deliver similar effects to what PathsToggler does.

December 2019 - (before Christmas)