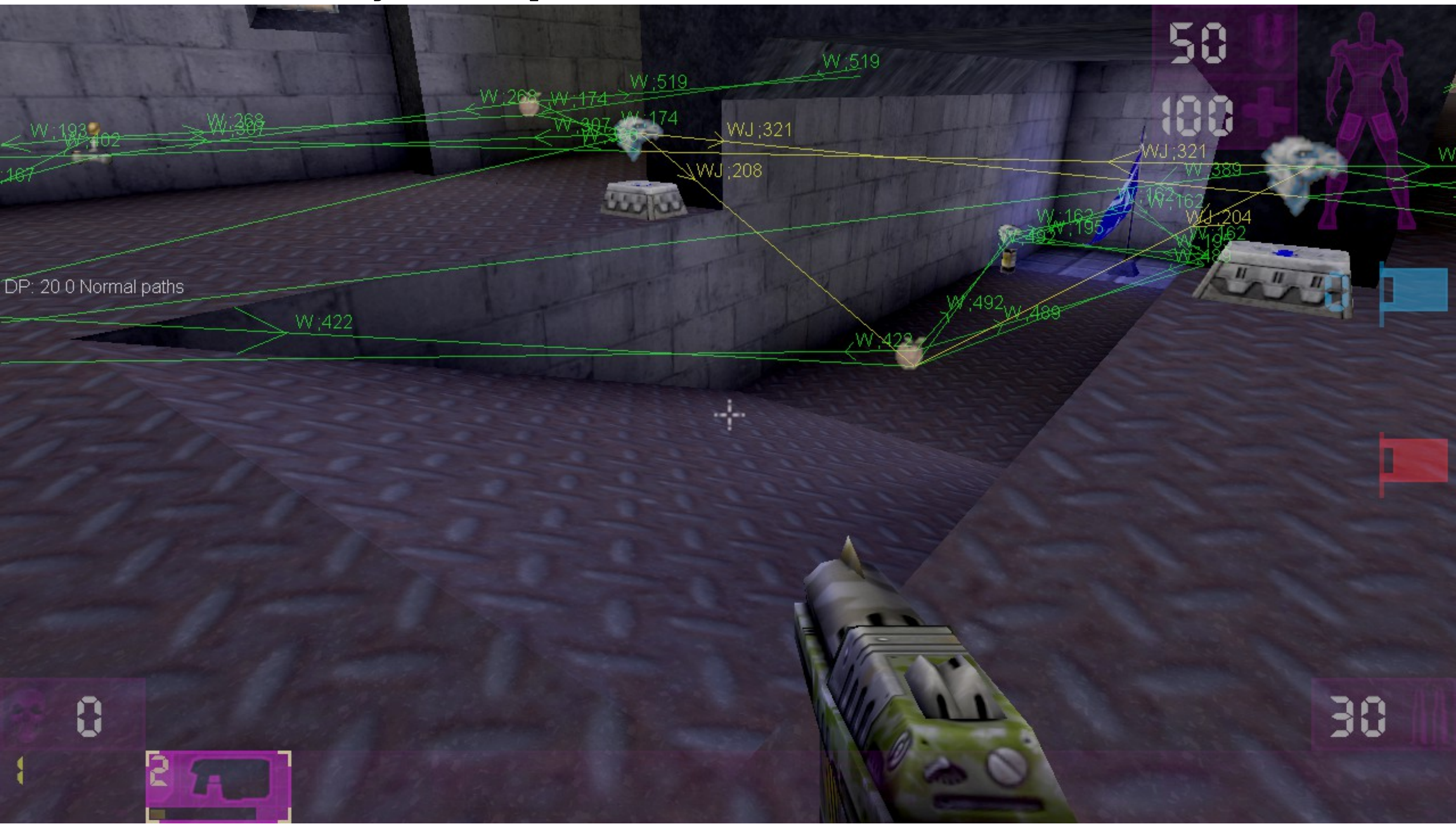
DRAWPATHS

UT'99 Tester mutator
by Buggie

Basic Info: This mutator is used to see the navigation paths used by artificial intelligence, namely the connections between navigation points. As the default property, the mutator draws the navigation pathways normally used but it can show us other details, as well as the dynamics of the navigation network processing when a creature is looking for a way to a particular target. The mutator can be handled using the <MUTATE> command and specifying parameters separated by character <SPACE>.

It starts working this way:



Drawing navigation paths is different from what the original UT Editor shows us, it's a mix between U227 and UT469. We have "X" like drawn paths as in U227 but colors and arrows similar to the patched UT469. It shows ReachFlags shortly - W = Walk J = Jump, etc. and paths distances.

Extra Info: The mutator works using the native function in the UE1 that is called <describespec>.

```
native(519) final function describeSpec(int iSpec, out Actor Start, out Actor End, out int ReachFlags, out int Distance);
```

As a result of using the original function, we will not have data about the size of the creature accepted by a navigation area, but we have enough basic data to tell if we have or not a fracture and therefore a compromised navigation network.

Operation:

In a local game, we choose this mutator from the list of mutators. It is called <DrawPaths>, and we start game alone or with Bots. We can see Paths as Lines drawn by mutator.

Options and Drawing Types (mutate commands in color):

mutate draw 0 - it draws normal paths;

mutate draw 1 - it draws VisNoReachPaths - these are not exactly usable paths;

mutate draw 2 - it draws PrunedPaths - this is the only way to see these;

mutate draw 3 - it draws Radii/Height tubes - I don't think I need this;

mutate draw 4 - it draws bEndPointOnly nodes - not very End Points as they can successfully substitute a PathNode (InventorySpot) - I believe this is an internal solution for creating paths heading to inventories or whatever processing method;

mutate draw 5 - it draws the dynamic internal chain NextOrdered PrevOrdered variables and links - this data has no use in Editor, it's adjusted during run-time;

mutate draw 6 - it draws StartPath and PreviousPath - also junk data, navigation processing works out of these ”chains” without a single problem;

mutate coverage 0 - it removes shown area accepted for entering the navigation network;

mutate coverage 1 - it draws area accepted for entering the navigation network, if pawn it's too far, it won't navigate unless it gets closer in this area - also it depends on how big is the map. Huge maps are not suitable for navigation due to engine's internal limitations.

Setup:

Copying INT and U files in System Folder and picking mutator in a Practice Session of game.

Assuming that user knows how to debug a problematic navigation network a tutorial about pathing it's not part of this document.

Credits goes at Buggie doing this tester mutator and sharing it at UT99.ORG.