Inference Engine Documentation/Ideas

Integration Team.

Integration of an inference engine in Office Tactics would occur solely in the CorpAIController class. Currently, The Corp AI Controller has a function “FindTargetObjective()” which does the core of the decision making of the NPC characters.

I submit that within the Corp AI Controller class, we implement 3 functions in lieu of “FindTargetObjective()”. Those 3 functions will be the Working Memory, Knowledge Base, and Agenda, functions. The Corp AI Controller would also have 2 container data structures, such as an array, to “pass” the data from each function to the next.

The Working Memory Array would be static and every element would be a specific statistic, that way when it is passed to the knowledge base, that function will understand which element is which statistic.

for example the working memory array would consistently look like:

Array:

[0] = Unit ID

[1] = Current Unit Health (%)

[2] = Current Ability Points

[3] = Current Enemy Unit Health 1(%)

[4] = Current Enemy Unit Health 2(%)

[5] = Unit Potential Damage\*

[6] = Enemy Unit Potential Damage 1

[7] = Enemy Unit Potential Damage 2

[8] = Unit Location (x,y)

[9] = Enemy Unit Location 1

[10] = Enemy Unit Location 2

[11] = Unit Last Move.

The Working Memory could use the built in functions already provided to gather this information and store it.

The knowledge base would use simple “IF-THEN” statements using the Working Memory Array data as the left hand side of the rules. Then provide potential Right hand rules depending on the Working Memory data, and store the decisions that proved true, in another array to have the Agenda function finalize the decision.

The Knowledge base array would have to choose results from a predetermined list of potential decisions, stored as strings.

For Example:

“MoveAway”

“Attack\_MoveAway”

“MoveTowards\_Attack”

“Ability”

The knowledge base decisions need to be intelligent without being too specific. for example,

IF ( Health < 50% && Enemy in attack range)

“MoveAway”

Then store those decisions in the array.

The Agenda would decide what the highest priority decision would be, and execute the decisions by calling the particular “state” in Corp AI Controller. This can be done by using a series of IF statements, having the highest priority result be the first firing IF statement, which would change the current state of the controller if the if-statements prove true.