

---

Subject: Attack\_Target\_Priority, script for AI Bots?  
Posted by [TNaismith](#) on Sun, 12 Sep 2010 07:09:40 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

I've been working around some co-op maps and have been exploring different possibilities for adding bots to maps in connection to map/mission objectives. Browsing through the JFW scripts, and then briefly through others such as KAK, TDA and z, I haven't come across any particular scripts that allow for a dynamic 'attack target priority' system to be set. Let me describe in detail what I mean:

Scenario:

The (blue) bot vehicle has the primary objective of destroying the (red) building. The infantry (light orange) and vehicle (dark orange) are defending the (red) building and trying to stop the (blue) bot vehicle.

How the scripting is setup:

To simulate the (blue) bot vehicle having a primary objective to destroy the (red) building, a "Daves\_Arrow" object is placed on the (red) building, perhaps just a little bit inside the interior walls. Attached to the "Daves\_Arrow" object is some kind of 'Attack\_Target\_Priority' script that says "I am something bots [can 'see' and] will decide to attack based on an assigned priority number and weapon/sight proximity range. I also have a on/off setting that can tell AI Bots to mark me as attack even if there are enemy infantry or vehicles nearby.'. If I am sent a custom, I become deactivated/activated, meaning AI Bots will no longer 'see' me, or shoot me...and vice versa"

For the (blue) bot vehicle, a script that could be named 'Attack\_Target\_Priority\_AI' would be attached, and it would allow the bot to interact with 'Attack\_Target\_Priority' scripts. It would be able to prioritize to fire at any 'Attack\_Target\_Priority' objects nearby, or to fire at nearby enemy infantry or vehicles.

In this scenario, the bot vehicle would fire at the (red) building, but will also fire at the (light orange) infantry and (dark orange) vehicle. It will follow this plan until there are no 'Attack\_Target\_Priority' objects in range, and also no enemy units.

Attack\_Target\_Priority

Here are the possible settings for the 'Attack\_Target\_Priority' script attached to the "Daves\_Arrows" object: Priority\_Number: Variable (Number placed here will tell corresponding AI Bots how high to rank as a priority target to attack compared to any other nearby priority targets and their relative Priority\_Number)

Absolute\_Priority: Boolean (Yes/No. If Yes, then AI Bots will attack target even if there are enemy

infantry or vehicles around. If No, then AI Bots will attack enemy infantry or vehicles nearby, then return to attack the target if area is clear.

Custom\_Enable: Variable (Input the custom message to listen for to activate this script.)

Custom\_Disable: Variable (Input the custom message to listen for to deactivate this script.)

Additional Thoughts: Absolute Priority could be divided into 'Absolute\_Priority\_Infantry' and 'Absolute\_Priority\_Vehicles'. 'Absolute\_Priority\_Air' might be a consideration too.

### Attack\_Target\_Priority\_AI

Here are the possible settings for the 'Attack\_Target\_Priority\_AI' script attached to the (blue) bot vehicle:

ID1: Variable (Input the object ID that has 'Attack\_Target\_Priority' attached to it)

ID2: Variable (Input the object ID that has 'Attack\_Target\_Priority' attached to it)

ID3: Variable (Input the object ID that has 'Attack\_Target\_Priority' attached to it)

ID4: Variable (Input the object ID that has 'Attack\_Target\_Priority' attached to it)

ID5: Variable (Input the object ID that has 'Attack\_Target\_Priority' attached to it)

ID6: Variable (Input the object ID that has 'Attack\_Target\_Priority' attached to it)

ID7: Variable (Input the object ID that has 'Attack\_Target\_Priority' attached to it)

ID8: Variable (Input the object ID that has 'Attack\_Target\_Priority' attached to it)

Priority\_Mode: Boolean (Sets the script's target priority. 0 = Attack\_Target\_Priority scripted objects before Infantry and Vehicles, 1 = Infantry before Attack\_Target\_Priority scripted objects. 2 = Vehicles before Attack\_Target\_Priority objects. 3 = Infantry and Vehicles before Attack\_Target\_Priority scripted objects. Anything else will cause the script to ignore infantry/vehicle distinctions and rely solely on range/health/player/attack\_target\_priority scripted objects for prioritizing. Default is 0. If there are two Attack\_Target\_Priority objects with the same priority numbers, the AI will treat both equal and alternate between attacking both objects.)

Min\_Range: Variable (The minimum range to attack a target. Targets closer than this will be ignored. Default is 0.0.)

MaxRange: Variable (The maximum range to attack a target. Targets further than this will be ignored. Default is 100.0.)

Additional Thoughts: 'Attack\_Target\_Priority\_AI' could also include parameters for a bot following a waypath; Deciding to stop, hunt, pursue, destroy and fire at any 'Attack\_Target\_Priority' objects (Or enemy infantry/vehicles) while following a waypath, or to continue moving on the waypath while simultaneously firing at 'Attack\_Target\_Priority' objects (or enemy infantry/vehicles) until the bot moves out of weapon/sight range of the said objects and/or enemy infantry and vehicles.

### Bonus Example, An Advanced Scenario:

Given what we know now about how these scripts could work and how they could be setup, here is how the scenario shown above would unfold:

The (blue) bot vehicle (with 'Attack\_Target\_Priority\_AI' script attached) would focus on firing at the two priority = 1 ATTACK TARGET objects (ideally setup as Daves\_Arrows with 'Attack\_Target\_Priority' script attached) until they become deactivated, treating both as equal priority targets to attack. It will also attack any of the defending (light orange) infantry and (dark orange) vehicle(s) that are inside it's weapon/attack range. In addition to the first two ATTACK

TARGET objects (Which have a priority of 1 assigned to each), the (blue) bot vehicle will also proceed to fire at the 3rd ATTACK TARGET object in the top-left corner, with a Priority number of '2'. It will fire at this object less regularly than at the first two ATTACK TARGET objects (with priority of 1), but it will shuffle it's attention around all three ATTACK TARGET objects (unequally) until each one is deactivated (no longer on the AI bot's radar, figuratively speaking), and it will also attack any enemy infantry or vehicles nearby.

Similar Concepts:

- JFW\_Engineer\_Repair ([www.Game-maps.net](http://www.Game-maps.net), update344 documentation)
  - JFW\_Engineer\_Target ([www.Game-maps.net](http://www.Game-maps.net), update344 documentation)
- Quote:JFW\_Engineer\_Target (clone of M03\_Engineer\_Target)

JFW\_Engineer\_Repair (clone of M03\_Engineer\_Repair)  
Repair\_Priority (priority for repairing)

To prevent the problem where when you get out of a vehicle, it is no longer being repaired by the bots, use NH\_SetTeam\_OnLeave, it should work.

Also, in addition, JFW\_Engineer\_Target has been modified to check the Shield Strength (i.e. armour) of the object in addition to the health.

The repair object can be any vehicle or infantry that has a repair gun weapon and wheels/legs/tracks/whatever so it can move.

Also, there is a limit to how far away the repair script will detect a damaged target (since I can't find the sound presets it's using, I can't identify how far that is)

JFW\_Engineer\_Target\_2 (clone of JFW\_Engineer\_Target)

JFW\_Engineer\_Repair\_2 (clone of JFW\_Engineer\_Repair)  
Repair\_Priority (priority for repairing)  
Target\_ID (the actual ID of the object that will get repaired)

This is intended so you can stick e.g. a Daves Arrow such that its "inside" the MCT and the engineer will repair that (and hit the MCT since the MCT is in the way)

- RA\_Base\_Defense\_Simple ([www.sourceforge.net](http://www.sourceforge.net), Readme4 documentation)
- Quote:Script Name: "RA\_Base\_Defense\_Simple"

Description:

A simplified version of a prototype script for a threat assessing Base Defense. It is capable of basic target prioritizing and can adjust its aim when firing at infantry. The script can be set to prioritize infantry, vehicles, or both equally. It will also rate targets based on remaining health, range, and if the target is a player or AI. Players are considered a higher priority than AI. The lower a target's health, the more likely it will be prioritized. The closer a unit is, the higher its priority is.

Script Parameters:

-MinRange: The minimum range to attack a target. Targets closer than this will be ignored. Default is 0.0.

-MaxRange: The maximum range to attack a target. Targets further than this will be ignored. Default is 100.0.

-AntiAir: Sets if the script targets aircraft. 1 = Yes, 0 = No. Default is 0.

-AntiGround: Sets if the script targets ground units. 1 = Yes, 0 = No. Default is 1.

-AdjustAim: Sets if the script will adjust its aim for infantry. 1 = Yes, 0 = No. Default is 0.

-TargetMode: Sets the script's target priority. 0 = Vehicles before Infantry, 1 = Infantry before Vehicles. Anything else will cause script to ignore infantry/vehicle distinctions and rely solely on range/health/player for prioritizing. Default is 0.

Notes:

Adjusting aim for infantry is useful for base defenses that fire "slow" projectiles, like a cannon. Infantry can typically sidestep these kinds of attacks. The adjusted aim targets the ground the soldier is standing on instead of the soldier. This causes the projectile to explode right next to a soldier instead of traveling another 20 or so meters before impacting the ground. For base defenses that fire "fast" projectiles or non-explosive projectiles, like a machine gun, do not adjust aim for soldiers, as the adjustment would likely cause the projectile to miss completely.

Acknowledgments

I would like to openly embrace the fact that this idea isn't foolproof. I do not have experience with coding, programming, writing scripts, C++, or any other special 'technical know-how'. I only use the knowledge and observations made from constantly browsing through scripts while setting up co-op missions + maps in Level Edit, and also from reading documentation about the scripts in 3.4.4 where available. Realizing this, there are probably a lot of possible 'What-ifs' and 'This won't work because...'. I encourage you to point out any flaws or improvements for the idea, and if there is anything similar that already exists for this, please feel free to let me know as well.

Why is this written?

With new maps being made for co-op gameplay over at A Path Beyond, I have found renewed interest in learning to make maps and script co-op missions using the community tools both Renegade and A Path Beyond share in common.

---

Subject: Re: Attack\_Target\_Priority, script for AI Bots?  
Posted by [Sladewill](#) on Sun, 12 Sep 2010 08:18:38 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Sounds interesting.

---

Subject: Re: Attack\_Target\_Priority, script for AI Bots?

Posted by [Spyder](#) on Sun, 12 Sep 2010 08:38:42 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

This would come in handy for a real skirmish map. The "Multiplayer Practice" mode that Westwood designed is just not like the online experience. If there is a possibility to make the scripts you just mentioned above, you might be able to create a more realistic multiplayer experience for several maps, so new players can practice more efficient before joining the online scene.

---

Subject: Re: Attack\_Target\_Priority, script for AI Bots?

Posted by [danpaul88](#) on Mon, 13 Sep 2010 11:28:27 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

My AI scripts already handle targetting priority for AR and APB base defences in our development builds and are designed to be easily extensible to infantry / vehicles... there is in fact an AI for tanks already implemented although it needs some tweaking.

My stuff works by assigning a 'base priority' to each target type (infantry, vtol, light vehicle, heavy vehicle, building) and then adjusting that based on a set of modifiers including distance to target, value of target and targets health status. It then picks the highest priority target to attack and continually re-evaluates the priorities of all available targets.

All the implementation specific classes inherit from the base class which handles the actual priority calculation, so it's easy to re-use it across other scripts. For instance dp88\_AI\_Tesla\_Coil inherits from dp88\_AI\_Turret which in turn inherits from dp88\_customAI (not actually a script since it's partially pure virtual) which itself inherits from PublicScriptImpClass.

dp88\_AI\_Turret is the base class for all turret type AI's and provides virtual functions to check the base power status etc, allowing scripts like dp88\_AI\_Tesla\_Coil to override the base power status with its own implementation which allows the defence to be 'powered' by tesla troopers (this is for the AR mod) OR by the standard base power without rewriting all the targetting and priority calculation code to take it into account.

Example of script parameters in use currently;

Quote:

Priority\_Infantry <- Base priority for attacking infantry (0 = do not attack infantry)

Weapon\_Infantry <- Weapon for attacking infantry (1 = primary, 2 = secondary)

Splash\_Infantry <- Set to 1 to attempt to damage infantry with splash (NOT recommended for high ROF weapons, it will slightly decrease accuracy due to targeting a location rather than the unit)

Priority\_Light\_Vehicle <- Base priority for attacking light vehicles (0 = do not attack light vehicles)

Weapon\_Light\_Vehicle <- Weapon for attacking light vehicles (1 = primary, 2 = secondary)

Priority\_Heavy\_Vehicle <- Base priority for attacking heavy vehicles (0 = do not attack heavy vehicles)

Weapon\_Heavy\_Vehicle <- Weapon for attacking heavy vehicles (1 = primary, 2 = secondary)

Priority\_VTOL <- Base priority for attacking VTOLs (0 = do not attack VTOLs)

Weapon\_VTOL <- Weapon for attacking VTOLs (1 = primary, 2 = secondary)

Min\_Attack\_Range <- Minimum attack range for primary weapon

Max\_Attack\_Range <- Maximum attack range for primary weapon

Min\_Attack\_Range\_Secondary <- Minimum attack range for secondary weapon

Max\_Attack\_Range\_Secondary <- Maximum attack range for secondary weapon

Requires\_Power <- Require power to attack (Checked against base power state)

Distance\_Modifier <- Sets the value for the distance-to-target priority modifier, higher values increase the priority of closer targets (recommendation: higher value for inaccurate weapons, lower for accurate weapons)

Target\_Damage\_Modifier <- Sets the value for the target damage priority modifier, higher values increase the priority of damaged targets (ie: more likely to attempt to finish off a heavily damaged unit instead of a healthy unit)

Target\_Value\_Modifier <- Sets the value for the target value priority modifier, higher values increase the priority of high value targets (ie: purchase cost)

Debug <- Set to 1 to instruct the script to create a text file logging how its calculated priority for different targets. This will allow fine tuning of modifiers and priorities since devs can get an idea of how its working internally.

---