Tactical Combat Link - (TypeX) Changelog v.1.0.19

==========

Fixed:

(A.I.) Artillery Support: Script error while using T.C.L. without “TCL_Sound.pbo” or script based.
(A.I.) Helicopter: None armed helicopters return to their original position after cargo A.I. unit(s) unload was finished.
(A.I.) Helicopter: Changed A.I. helicopter ranking order to prevent helicopter crew will be switched to A.I. group leader.

Enhanced: (Features)

(A.I.) Artillery: Full re-worked and realistic artillery feature.
(A.I.) Take Cover: Improved A.I. unit(s) cover stance to fit cover object height.
(A.I.) Request Time: Increased time A.I. need to request reinforcement(s).
(A.I.) Static Weapon: A.I. unit(s) assigned to static mortars do fire if enemy(s) are known and do fire for effect.

Tweaked:

TCL_KnowsAbout.sqf: Main enemy detection script loops from 5 second loops to 1 second loops.
Several script and function codes.

Changed:

(A.I.) Artillery Support: Restricted to vehicles kind of “Tank” only.

Added:

(A.I.) Artillery Support: Restricted to vehicles kind of “Tank” only.
(System) A.I. Group Types: Added new A.I. group type “**TCL_Freeze**” which exclude given A.I. group(s) from moving to their enemy(s) position and “**TCL_Default**” which excludes A.I. group(s) from requesting and be requested as reinforcement(s).
Note: Those A.I. group types have been included to prevent mission breaking by any kind of A.I. movement and reinforcement request.
A.I. group(s) using those group types are still using all A.I. combat features.
(System) UserConfig: Added check to prevent error message if no “**UserConfig** > **TCL**” folder is used.
(System) Settings: All settings can be set in the “**Init.sqf**” of the given mission.
If this way is used setting files from given “**UserConfig** > **TCL**” file will be ignored.
(System) Settings: Started adding settings to ”**UserConfig > TCL > TCL_Feature.sqf**” settings.

Warning!

The “**TCL_AI**” and “**TCL_System**” settings have been updated / modified!
Make sure to update them as well if used somewhere in any mission!

New = Red
Changed = Blue

TCL_AI = [**0.1**, 0.15, 3, [3000, 5000, 7000, 10000, 13000], False, **4**, 700, True, **False**, **False**, True, 170, **2**, False];

Tactical Combat Link - (TypeX) Changelog v.1.0.17

==========

Fixed:

(System) Rating: A.I. helicopters wasn't rated correctly.
(System) A.I. Reinforcement Request: Depending on which A.I. side was set to be friendly to another A.I. side (Independent Allegiance) sometimes wrong A.I. reinforcement(s) side was requested.
(System) A.I. Map Marker Debug: Fixed bug where re-spawnde A.I. group(s) failed to get initialized by the A.I. Map Marker Debug.
(System) Script Based: Fixed bug where Script Based initialize used “Sound F.X.” from AddOn Based “TCL_Sound.pbo”.
(A.I.) Regroup: Script syntax error after A.I. unit(s) regrouped caused by deleted A.I. group.
(A.I.) Artillery Support: A.I. artillery did not fire because of wrong friendly within firing range command usage.
(A.I.) Garrison: A.I. which was about to garrison buildings was stopped right after detecting enemy(s) during building garrison.
(A.I.) Reinforcement: A.I. reinforcement request sometimes was executed multiple times for the same A.I. group instead of once only.
(System) A.I. Spawn: Re-wrote of the A.I. spawn initialize system which fixed script errors caused by not properly initialized spawned A.I. unit(s). Tested with over 170 spawned A.I. vs A.I. unit(s) using all features of Tactical Combat Link at the same time.

Optimized: (Code and Syntax)

(A.I.) Visible Enemy Detection: Changed eyePos lineIntersects and terrainIntersectASL to targetKnowledge.
Much faster results and optimized in terms of code performance.
(A.I.) Take Cover: A.I. unit(s) will not use "**nearestTerrainObjects**" command while within cover search range.
Cover objects will be cached once only while given A.I. unit(s) are within the given cover search range which currently is set to 30 meters.
Huge step in terms of code performance and still much more potential when it comes to code optimization.
Cover search function will exit after the first cover object which fully fits all conditions was found which reduces almost every A.I. cover search loop from 10 - 30 to 1 - 10 checked cover objects.
(A.I.) Take Cover: Once more huge code and syntax optimization!
(A.I.) Spawn: Spawned A.I. group(s) wil be initialized right after the last A.I. unit of the spawned A.I. group was created.
(A.I.) Vehicle Check: Changed A.I. unit(s) vehicle check to <**isNull objectParent _unit**>.
(System) Optimization: General code and syntax optimization in many other parts of the mod.

Enhanced: (Features)

(A.I.) House Search: A.I. unit(s) / group(s) spread out to do house search much more often and effectively after their known enemy(s) was killed.
(A.I.) Artillery Support: Added function to prevent A.I. unit(s) / group(s) friendly to the requested artillery move straight into its firing range.

Tweaked:

Changed: A.I. artillery firing range.
Changed: Many 1 second while do loops to 5 seconds.
Changed: A.I. line of sight check from 5 seconds to 30 seconds.
Changed: System initialize to none C.B.A. requirement. Thanks to .kju
Changed: Amount of A.I. unit(s) in vehicle(s) using smoke while unassigning the vehicle(s) to prevent vehicle(s) with huge amount of cargo unit(s) using too many smoke shells.

Added:

(System) Initialize: It is now possible to initialize the A.I. combat system of Tactical Combat Link with each A.I. unit(s). The react to weapon fire and react to suppressed fire will stay restricted to player(s) only.
(System) Setting: Added setting to the “TCL_System” settings to enable / disable the A.I. vs A.I. combat system.
(System) A.I. Group Type: Added new A.I. Group Type to exclude specific A.I. unit(s) / group(s) from using the A.I. vs A.I. combat system initialize by using “TCL_Idle”.
(System) Disable: Added switch to disable initialize of Tactical Combat Link by using <**TCL_Preprocess = False;**> without “< >” from the mission “**Init.sqf**” or any other init field.

Most of those optimizations should be not noticeable but free up some huge space in terms of code reading and execution performance.