Robot Army - New Algorithm for squad maintaining and AI handling

# New algorithm main objectives:

* Assembler-centric operations – squad and origin of enemy targeting is from the assembler.
* split up units spawned by player, and units spawned by assembler style buildings
* Individual combinator-control of assembler’s controlled squad. Per-squad size, hunt radius settings.
* Make game start-up settings for some of the defaults, to help before the user gets combinators.
* Patrol stations get a ‘guard radius’ config param and combinator control signal. Squad will respond to short range threats and return to their patrol post.
* Remove all usage of unitgroup API.
* Old “Squad” table is now just created one per assembler, or as needed by proximity when player places units manually.

Features which must work:

1. Droid assembler is able to spawn units, have them maintained in a list, and issue commands on AI tick functions like before the changes.
2. Droid guard station is able to spawn units, have them maintained in a list, and issue commands on AI tick functions like before the changes.
3. Select a squad and issue move commands directly.
4. User is able to place units, which adds to a list of loosely gathered units nearby if any, to form an ad-hoc squad. Squad selection tool should then allow control. No direct/automatic AI for these units.

# For Units Made by Assemblers:

**global.assemblerSquad[assembler.unit\_number]**

Table now used for storing information about the assembler’s squad and associated units.

Has sub-tables for member list, some simple settings/configuration which are default or set by attached combinators.

**global.assemblerAssignment[SpawnedEntity.unit\_number]**

Table used when a unit is spawned by an assembler, to allow the unit’s unique key to look-up the associated assembler immediately. This then allows the rest of the assembler’s units to be accessed/referenced, and the settings etc. Helpful when selecting/commanding the unit/s.

## Spawning/Creation Process:

Unit is spawned, added to the assembler’s unit list. Unit list is checked for ‘alive’ droids and maintained. With another global table, unit’s *unit\_number* is used as a key to refer back to the assembler which made it.

## Automatic AI handling:

The AI is still handled per squad, but the squad is attached to either ‘nothing’ (player,event spawned), or a droid assembler/guard station.

The Squad’s tick handler AI will interrogate for the squad’s ‘home’. Will run AI routine based on the result of that.

# For Units Made by Player Placement:

### Spawning/Creation Process:

Unit is spawned. Does not have assembler reference. Set to wander.

## Automatic AI handling:

The player placed units squads do not have normal AI. They must be controlled with selector tool and hotkeys (later GUI buttons?) or with Unit Control mod tools. Issued commands directly, for short-term missions.

# For Unit Selection/Commands

Selection tool – single or box select. All entities added to player’s unique selection list. Then, the entire squads of any selected entities are added to the selection. Apply selection stickers. The selection list is just simple entity list.

Commands iterate through entity list to issue command.

Deselection causes sticker destruction and nil the selection list.

**Handling some side cases:**

What happens if an assembler is removed/destroyed? What does its squad do?

What happens if a squad fails to get a task done? How do we tell, if there is no unitgroup anymore?

What happens if unit control is active? Combinator/Mod setting to disable automatic AI?

**Additional features:**  
Hotkey for selected squads to ‘retreat’ to their assemblers ( ALT + R)

Hotkey for selected squads to ‘follow player’. ( ALT + F )

Hotkey for selected squads to ‘stop’. ( ALT + S )

Hotkey for selected squads to ‘hunt’. ( ALT + H )