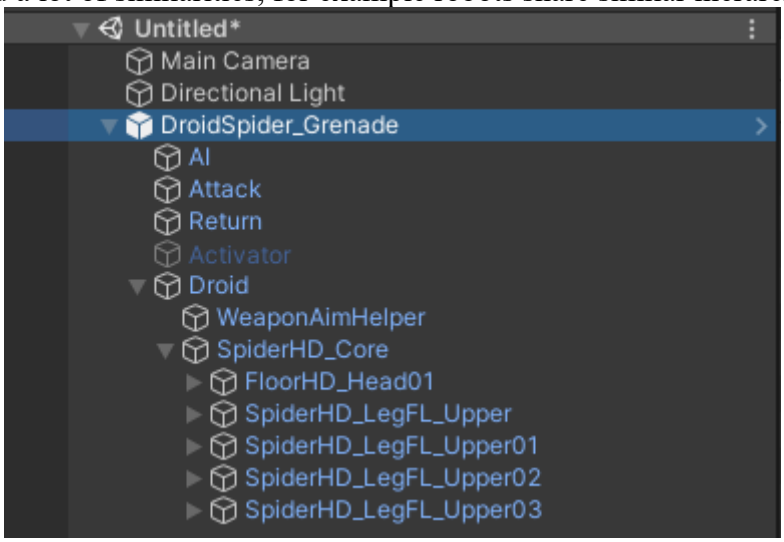
Bad Bots Manual.

Overview.

Bad bots (working title AngryDroids) initially meant to be a C# port of Unity's AngryBots project. They actualy shared a lot of similarities, for example robots share similar hierarchy.

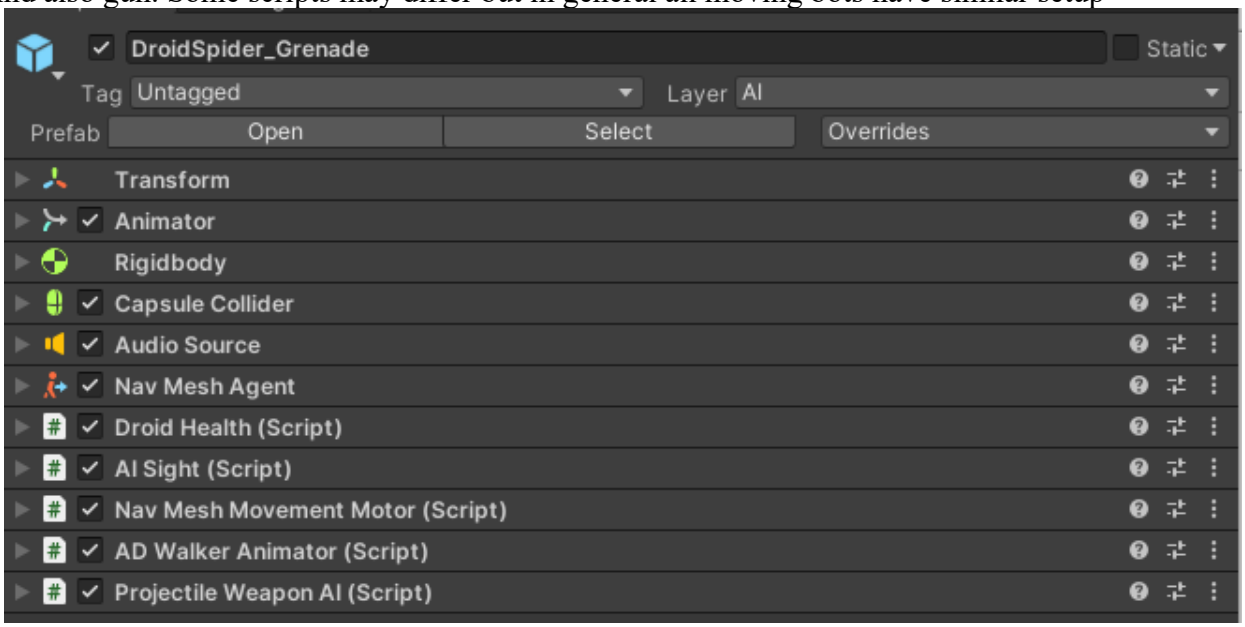


All droids have 4 logical child objects:

- **AI** – responsible for enemy detection
- **Attack** – responsible for attack behaviour
- **Return** – responsible for idle behaviour
- **Activator** – responsible for droid's game object activation/deactivation (effectively disabling droid when for from player and enabling when close)

Hierarchy followed by robot's own animated geometry.

Droid's root game object has few components attached that control it's movement and behaviour. And also gun. Some scripts may differ but in general all moving bots have similar setup

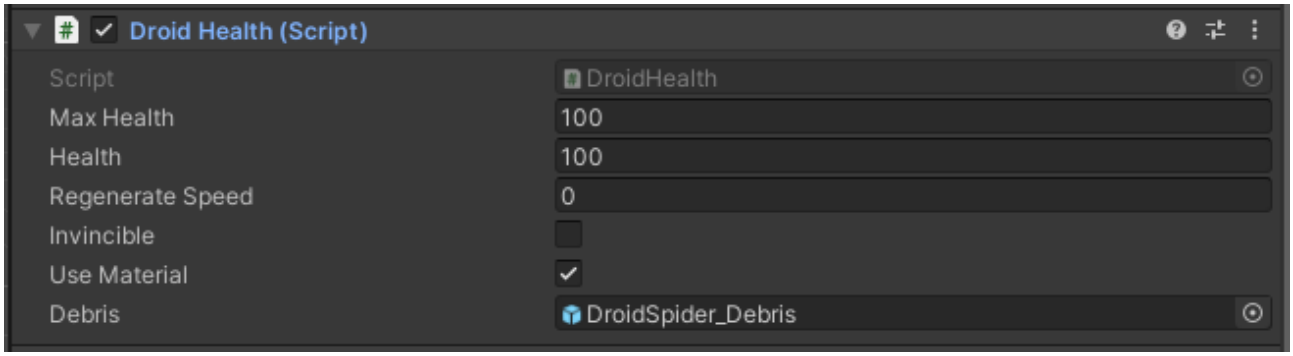


I'll skip unity's standard components as they are well described in reference manual, go see it it's cool.

- **Droid Health.**
- **AI Sight.**
- **Nav Mesh Movement Motor.**
- **AD Animator**

Droid Health.

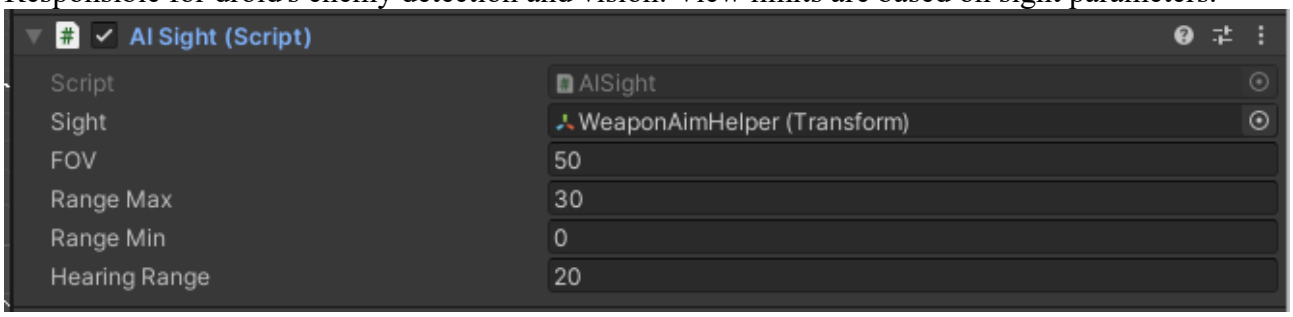
Responsible for droid's HP, manages droid's death sequence



- **Max Health.** Maximum points health can regenerate to.
- **Health.** Current HP droid has. Can be regenerated. Can be bigger then Max health (like heath boost etc), but regeneration is limited by Max Health.
- **Regenerate Speed.** Speed of HP regeneration, if 0 no regeneration is happening.
- **Invincible.** God mod toggle, droid takes no damage.
- **Use material.** If checked robot's damage is highlighted during gameplay
- **Debris.** Game object to spawn after droids death.

AI Sight.

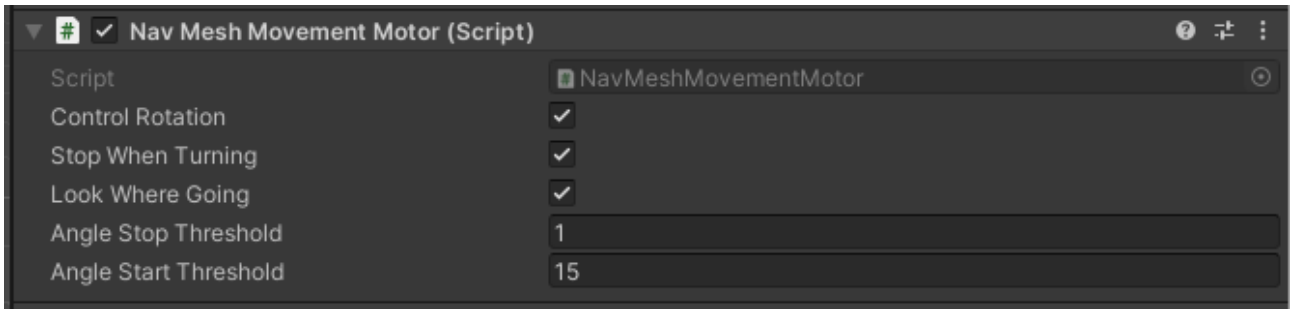
Responsible for droid's enemy detection and vision. View limits are based on sight parameters.



- **Sight.** Transform representing position and direction in what droid is looking at
- **FOV.** Sights field of view
- **Range Max.** Robot's viewing maximum distance
- **Range Min.** Is actually never used, but represents minimum distance droid starts seeing at
- **Hearing Range.** Not used currently, reserved for distance at which robot stops hearing at.

Nav Mesh Movement Motor.

Responsible for droids movement on navigation mesh. Most moving droids are now using it to navigate it's NavMeshAgent around a scene.



- **Control rotation.** If true, script is controlling NavMeshAgent's rotation when moving
- **Stop When Turning.** If Agent shouldn't move when making a turn
- **Look Where Going.** Agent looks along it's movement direction vector.
- **Angle Stop Threshold.** Angle at which Agent starts moving again after turning.
  Have no effect when Stop When Turning is turned off.
- **Angle Start Threshold.** Angle NavMeshAgent starts turning at.
  Have no effect when Stop When Turning is turned off.

AD Walker/Wheels/Hover Animator.

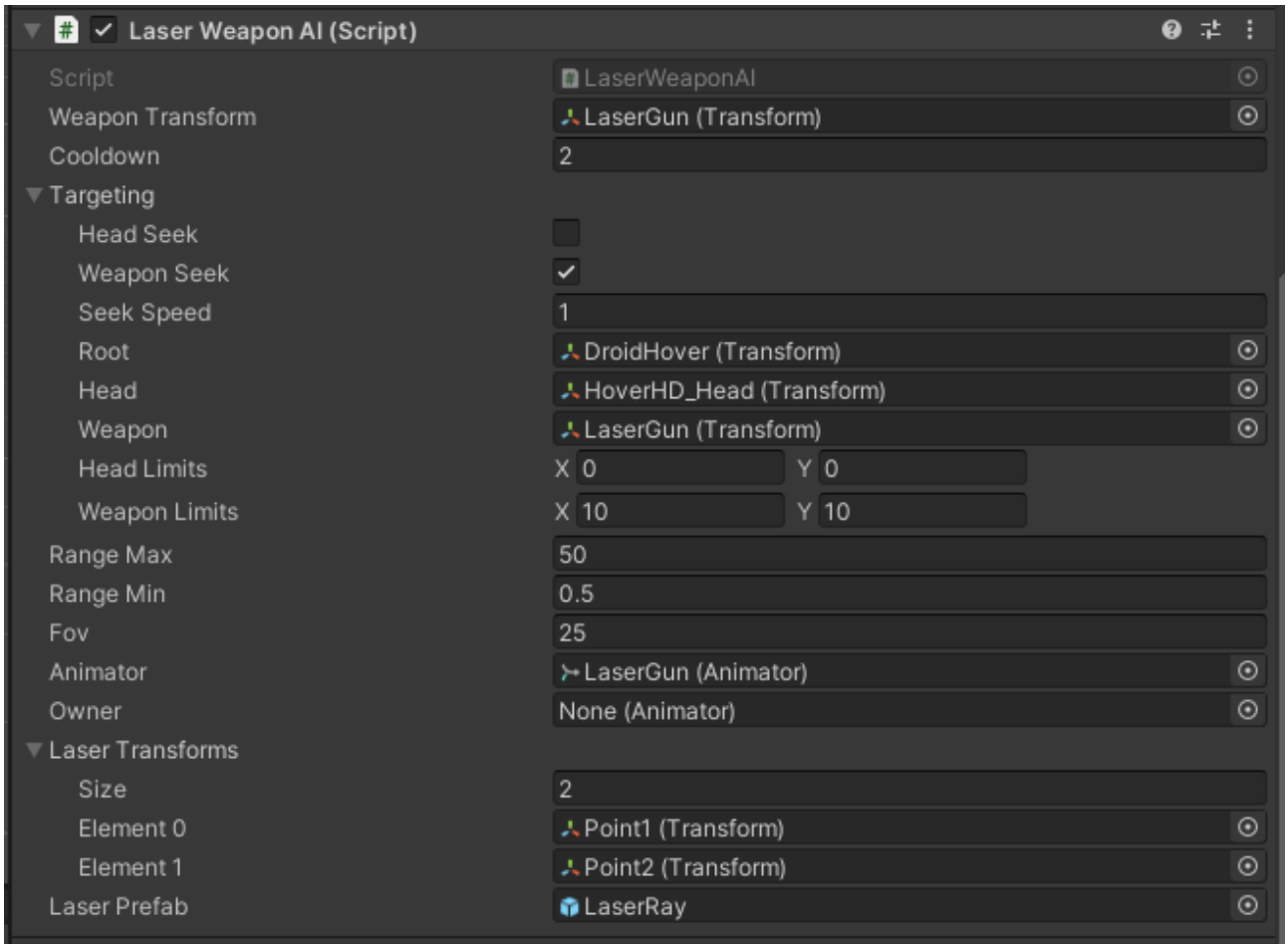
Scripts wiring droids movement with animation, in case of wheels script is controlling their rotation instead of animator.

Weapons.

Weapon components responsible weapons behavior, determining conditions under which weapons can be used (range, FOV, etc.) and is also responsible for weapons targeting.

Laser Weapon AI.

Responsible for Laser and Tesla weapons behaviour.

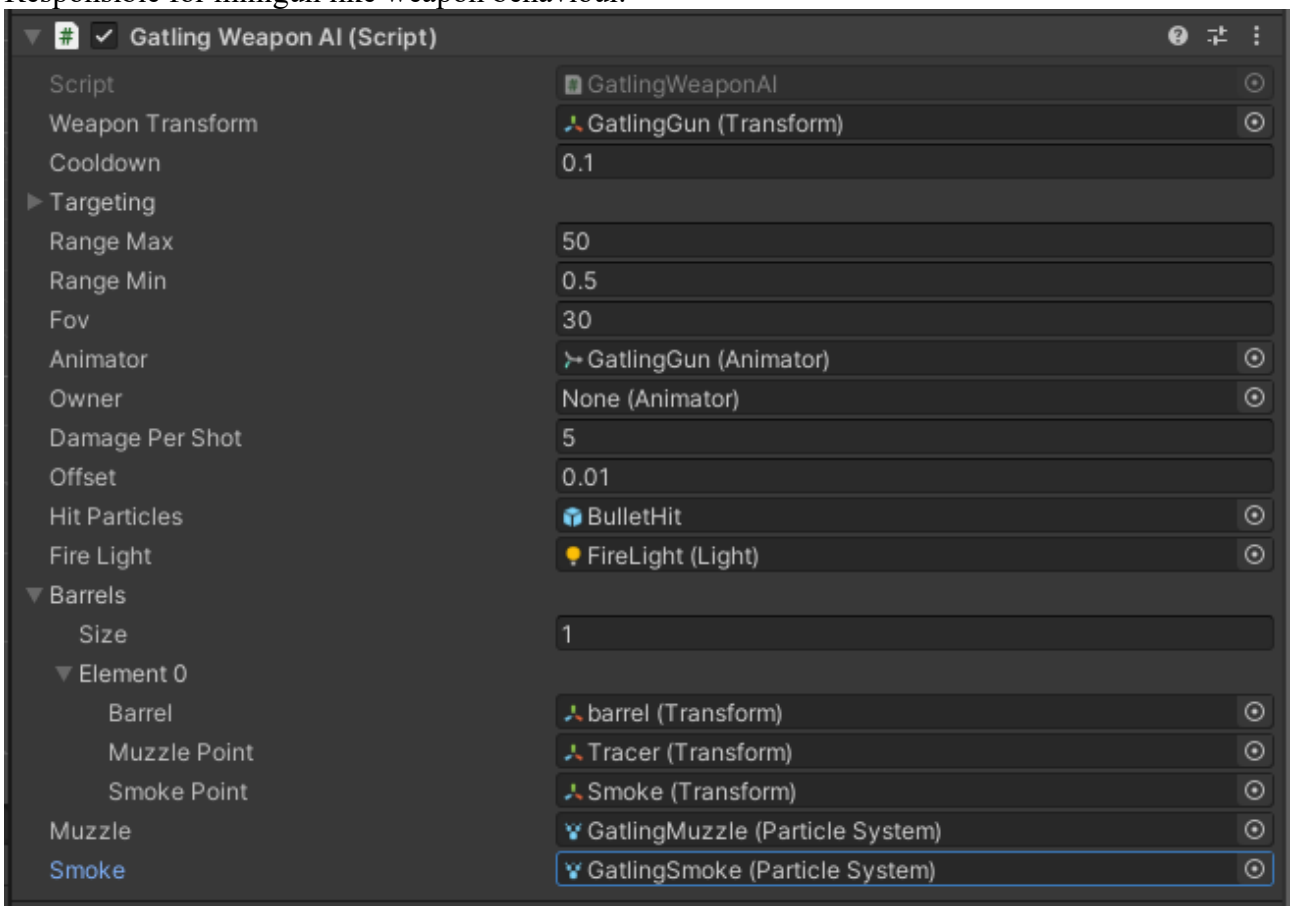


- **Weapon Transform.** Actual weapon game object. Transform is used as a reference of weapons position, direction, raycasts etc.
- **Cooldown.** Time it takes for weapon to get ready for next shot.
- **Targeting.** Weapon/Head aiming parameters that control droid's weapon rotation when aiming at it's target.
  Head seek. If true robot aims rotating it's head transform.
  Weapon seek. If true robot is aiming using it's weapon.
  Seek speed. Speed robot aims with, the more the sharper movement is, lower values for smoother motion. Note: slow speeds are significantly affected by animations.
  Root. Droid's transform used as world/local reference.
  Head/Weapon. Transforms to rotate when aiming
  Limits. Define vertical/horizontal angles limiting Head/Weapon rotation.
- **Range Max / Min, Fov**. Parameters similar to AISight's, defining zone of weapon's vision inside which object is considered valid for shooting.
- **Animator**. Weapons animator triggering small weapon shooting/recoil etc. animations
- **Owner.** Droid's animator to trigger when shooting. Currently is not used until shooting animations are not fixed.

- **Laser Transforms.** Parameters specific to laser type weapons. Points where laser prefabs will be instantiated.
- **Laser Prefab.** Parameters specific to laser type weapons. Laser ray (or any other) prefab to instantiate.

Gatling Weapon AI.

Responsible for minigun like weapon behaviour.

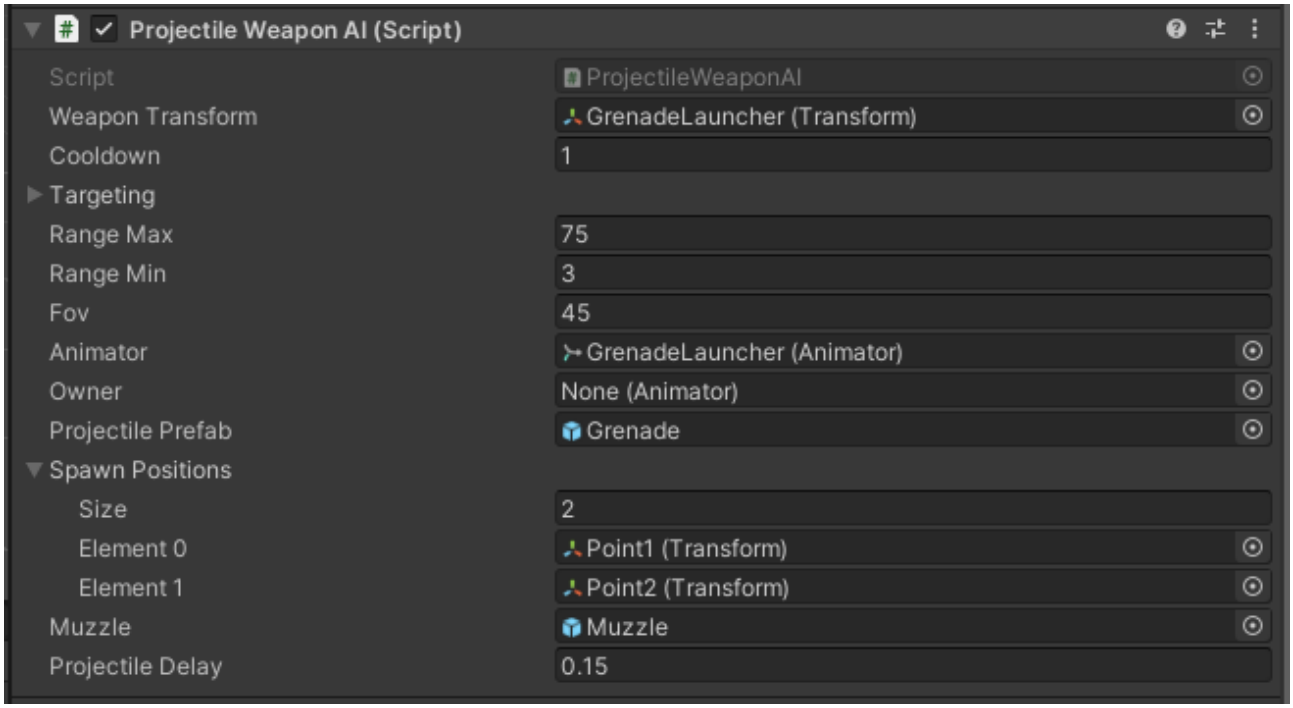


Most parameters are shared between different weapons.
Gatling weapon specific parameters:

- **Damage Per Shot.** Damage one hit does to enemy
- **Offset.** Shot offset to make shooting look more random, not 100% precise.
- **Hit Particles.** Particles to spawn on hit position.
- **Fire Light.** Lighting shots with dynamic lighting
- **Barrels.** Transforms responsible for positioning of particle effects.
- **Muzzle.** Prefab to use as muzzle flash prefab.
- **Smoke.** Prefab to use as barrel smoke.

Projectile Weapon AI.

Responsible for projectile launching weapons, like Grenade and Rocket launchers.



Most parameters are shared between different weapons.
Projectile launcher specific parameters:

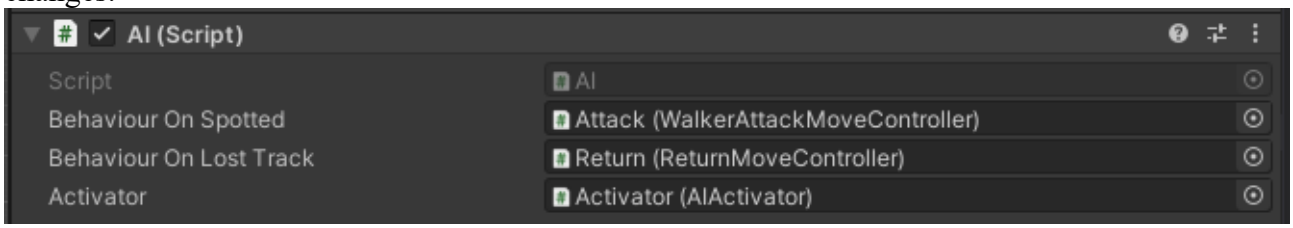
- **Projectile prefab.** Prefab to use as projectile. Can be guided/unguided.
- **Spawn positions.** Transforms representing poins where projectiles are spawned.
- **Muzzle.** Muzzle flash to spawn on weapon shot.
- **Projectile Delay.** A delay after which projectile is considered armed and can blow up. Implemented to avoid rocket exploding inside droid's weapon.

Artificial Intelligence.

Now let's look at how AI is implemented in Bad Bots.

AI.

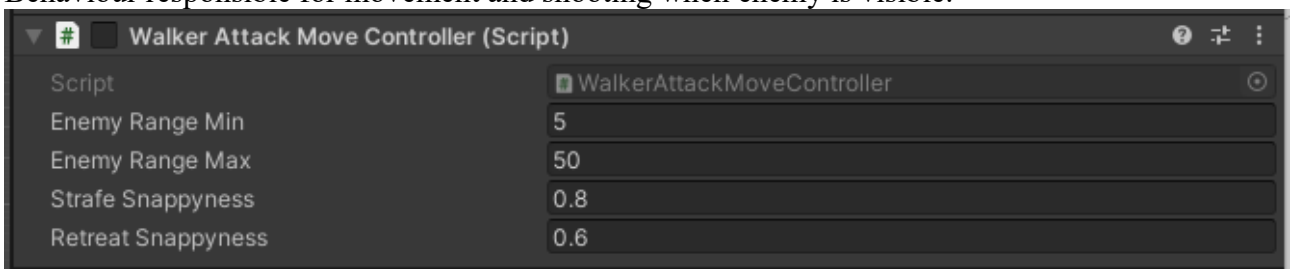
Main ai manager responsible for switching between behaviours as reactions to environment changes.



- **Behaviour On Spotted.** What to do when droid detects enemy.
- **Behaviour On Lost Track.** What to do when enemy escapes.
- **Activator.** Trigger that activates/deactivates droid.

Attack Move Controller.

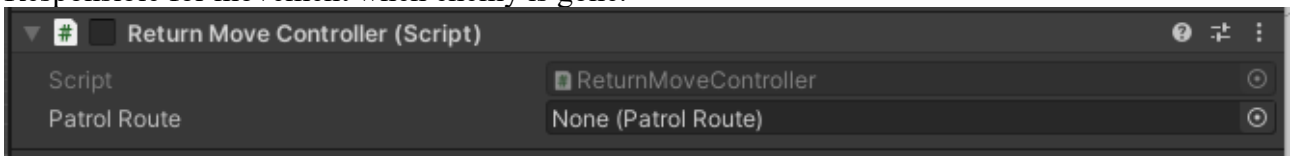
Behaviour responsible for movement and shooting when enemy is visible.



- **Enemy Range Min.** Minimum distance to enemy when attacking.
- **Enemy Range Max.** Maximum distance to enemy when attacking.
- **Strafe Snappyness.** Speed scale when moving sideways during attack when enemy is aiming
- **Retreat Snappyness.** Speed scale when going back if enemy is too close or advancing.

Return Move Controller.

Responsible for movement when enemy is gone.



- **Patrol Route.** If assigned droid will go back patrolling when enemy is gone or will do it at game start

This guide is far from finished and will be updated in next releases. For any questions bugreports simple contact enquiries please drop a mail to defaxer@gmail.com or leave them on this forum thread