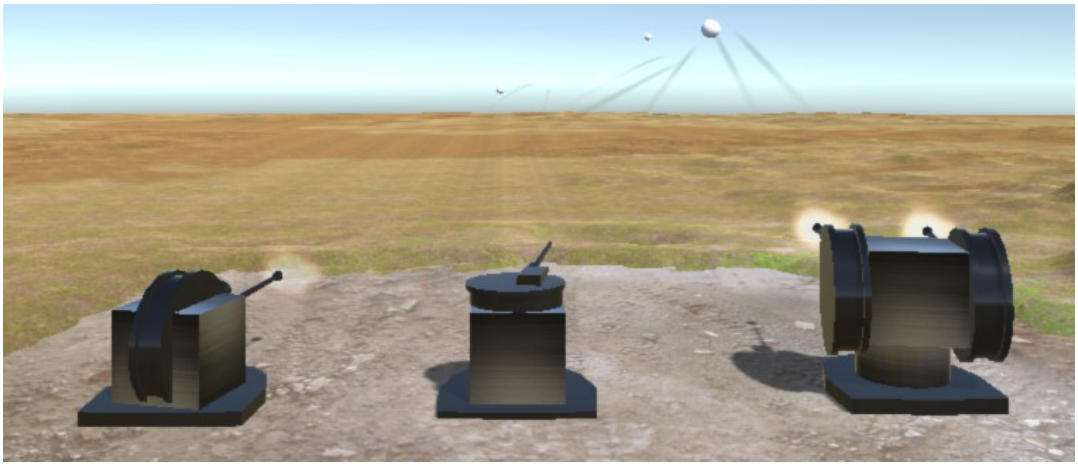
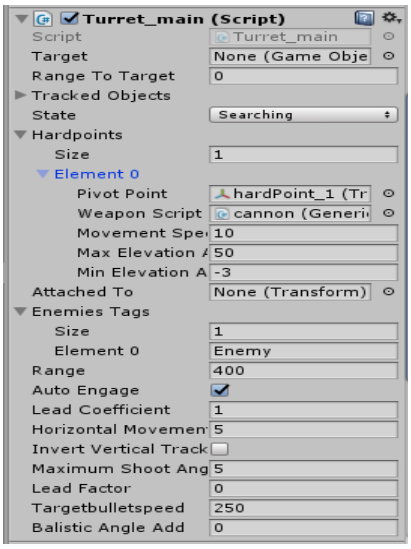
uSim AI Turrets system v1.0b



Introduction: AI turret system is a fast and very simple out of the box automatic defense system. It's object oriented programming in C# allows great degree of flexibility for mostly any project needs. It can be attached to a vehicle or to be static. It can be hanging or laydown. Shooting solver can resolve targeting under any relative movement/state. It can predict using rigidbodies velocity and shoot on lead to be able to hit fast moving targets.
Code driven articluation and movement allows use of any model desired and mounting types. Range and articulation angles can be set as well as rotational speeds.
The pack includes a balistic based ammo class to be used, but any prefab can be assigned to be instantiated by the cannons. Rate of fire can be adjusted.
The turret can also be manned by the user and recieve player input. Included in the package an example to toggle manual control on and off and a simple mouse targeting system.
Dynamic automatic target acquisition based in tag is used to feed the turret targeting system.
Have automatic defense systems for your games done in a second.
Use of state machine and coroutines makes the turret very low on resourse needs, ideal for anti air defenses simulation or tower defense games.
The development is open to sugestions and request on what it might need added, so please feel free to contact via mail at gabriel.campitelli@gmail.com

Setup: After importing the package, go into uSim AI Turret/prefabs folder and trow any type of turret steup into your scene. Place it in the position you want it to be. If it's attached to a moving object, then assign it to the variable “attachedTo” in Turret_main script inspector.

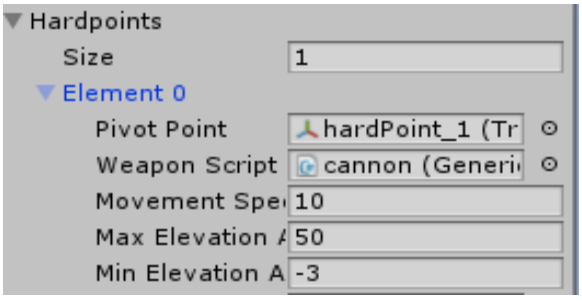


Target acquisition: Look for “Enemies tags” string array in Turret_main. This string correspond to the tag(s) that the turret will use to identify an enemy.
Also is possible to use layer's name as search criteria. Look for “Enemies layers” string array to add layers.
If your project can't use either aprouch you will have to manually feed the targets using

Turret_main.AddTarget(GameObject target);

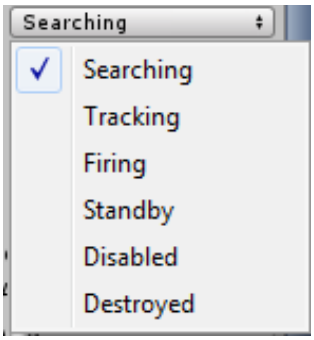
in all turret instances.

Movement restrictions: Each turret script contains an array hardpoints. This is a sub class that hold the weapons and it's vertical tracking.



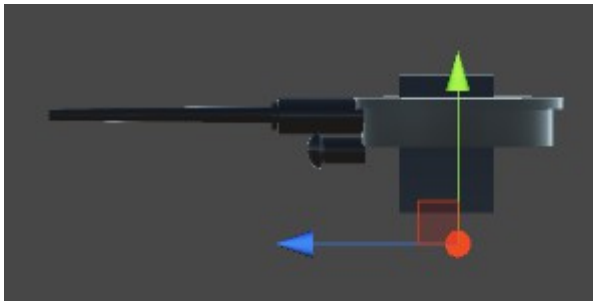
Set the vertical movement speed and the max and min elevations for each hardpoint.
In Turret_main you will find “Horizontal movement” variable for the horizontal speed of the turret main body.

Turret state machine: Here will explain what each state of the turret does.

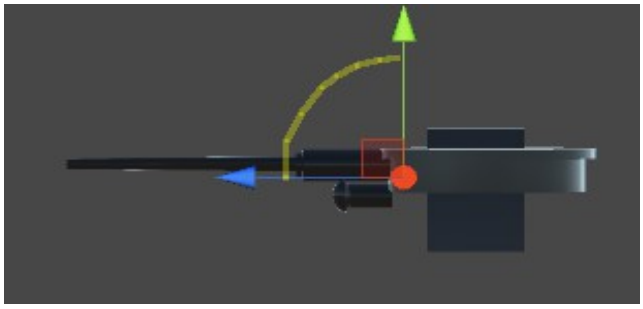


Searching: Puts the dynamic automatic target acquisition at work based in range and enemy tag list.
Tracking: Once a target has been picked the vertical and horizontal tracking is enabled
Firing: Calls the funtion to fire teh weapons if the weapon is ready to shoot.
Standby: Default state, if no targets in range.
Disabled: Does nothing.
Destroyed: Deactivate the turret GameObject if it's enabled.

Articulation: The turret main body rotates on it's base Y(up) axis around the gameobject pivot point.



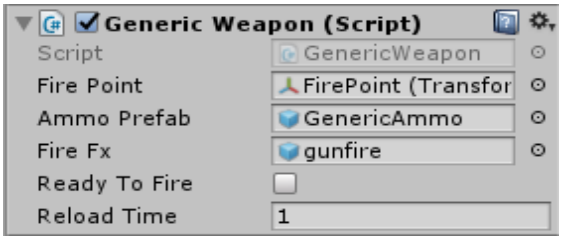
Each hardpoint in the turret is rotated around it X(right) axis around the Hardpoint pivot point.



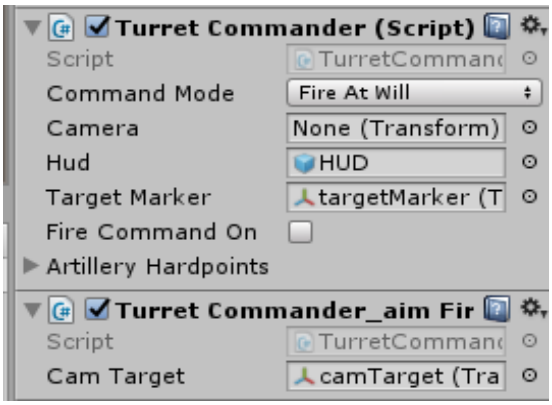
The turret can be customized by just changing the models in the structure.

Fx customization: Fire event instantiates an prefab as fireing Fx. An example is included with particle and sound Fx.
Same for the GenericAmmo script on detonation event. Example included.

Weapon attachment: each hardpoint has one GenericWeapon script attached to it. This script hold values for firing Fx, reload time and ammunition type.



Manual firing (Turret commander) : This class handles enabling and disabling the AI functions. A second script “TurretCommander_aimFiring” is used to direct the camera. The target is set to be in the center of the camera.



When Command mode is set to manual the script enables manual targeting.
A Camera transform needs to be asigned for it to work. Also a target marker transform in turret hierarchy is used. (see example included).
Cam target on the aiming script is the point at wich the camera rotates around based on mouse movement.

Thank you!

developed by Gabriel Campitelli
gabriel.campitelli@gmail.com