MobFarm Basics 2.0 - Getting Started

Get the most up to date information and tutorials at:
mobfarmgames.weebly.com

About

MobFarm Basics are a modular collection of resources providing basic functionality for other MobFarm packages. In general, these resources are for the control of AI units.

Basics include mobility and navigation scripts that allow the AI to control units, scanner and targeting scripts to create simple decision making, and turret and weapon scripts. Also included are a few prefabs of simple units.

Generally, each script component has an abstract base class that allows functionally to be built upon, while providing a common point of access that allows scripts to operate with each other.
For example, MF_AbstractWeapon is the base class for MF_B_Gun. Other scripts will access the base class to use the gun, and this allows the creation and use of different weapon scripts without changing the base class. In this way, most of the components can be swapped out and built upon.

Important!

To use the included scanner and targeting scripts, you'll need to set a few project settings, mainly to coordinate faction tag and/or layer names used.

Turret and Project Set-up

First, you'll need to decide if you want the turrets or units to choose their own targets, or if you'll supply the targets from another source - such as a vehicle that might hold more than one turret, or perhaps from player input.
Included are scripts to scan and choose targets, but of course you may want to extend these or build your own.

Automatic target selection:
The most typical ways sides or factions are tracked, are through tags or layers. SmoothTurret can handle either one.
MF_B_Scanner is the script that scans for targets and stores them in the MF_B_TargetList script, and MF_B_Targeting will then choose a target from that list. We will need to indicate a few things in these scripts that relate to your project.

When using MF_B_Scanner, choose either Tags or Layers from the Faction Method Type menu, corresponding to your method of team differentiation.
Now, you need to supply the tag names or layer names of the factions to be designated as valid targets.

The MF_enum script includes the names, "Side0, Side1, Side2, Side3".
You can change your tag or layers to match these names via the Unity3D menu: Edit > Project Settings > Tags and Layers.
Or, you can change enums to match the tag and layer names of your project by editing this line near the top of the MF_enmus script located at: _MobFarm Basics > Scripts > Abstracts_Statics:

```
public enum FactionType { Side0, Side1, Side2, Side3 };
```

Once you've got your tag or layer names matching, you can begin telling the scanner how to behave.
In the Targetable Factions field, increase the number to match how many different factions should be targeted.
Then, you'll see a matching number of drop down menus where you can choose this scanner's targetable factions.

By default, MF_B_Targeting will choose the closest detected target. You may change this by using the drop-down menu of Priority. Additionally, you may check Keep Current Target, and that target will remain locked until it dies or otherwise drops off the targeting list, such as due to moving out of scanner range.

Make sure the Controller field in MF_B_TurretControl is set to AI_AutoTarget.

Manual target selection:
If you want to manually designate a target:
1. In ST_TurretControl, set the drop-down menu of Controller to AI_AutoTarget.
2. Drag an object into the Target field of the MF_B_TurretControl script component in the inspector.
3. Finally, make sure there is no MF_B_Targeting script active.▯

The turret or unit will now aim or fire at the provided target and will not choose targets on its own.

This enough to get a basic targeting working. The rest of the documentation will cover more advanced features, and describe all the functions and variables in depth.

How to Use

Below is a brief overview on how various components interact. For more detailed descriptions, see the tutorials and script reference.

Movement and Navigation
To make a unit move, it needs a mobility script and a navigation script. The navigation script will make decisions on where to go, and the mobility script will handle how the unit moves. In this way,mobility and navigation scripts can be changed without changing the other.

Scanning and Targeting
If a unit is to react to other units in the scene, it needs a way to see them and make decisions. Scanners will detect and gather information about other units and place that info in a target list. Targeting scripts will then make decisions about elements in the list.
Units can have multiple scanners contributing to a list, and multiple targeting scripts reading the list.

Platforms and Weapons
Once a targeting script has chosen a target, it can then be used to aim a weapon. Platforms hold a weapon, and can be things like a stationary mount or a turret. Each platform is paired with a control element that looks for a target chosen by a targeting script, and operates weapons. The control element will feed the target to the platform, and the platform will handle the specifics on how to engage the target, such as a turret computing an intercept course and turning to the correct position. When a firing solution is found, the control script will operate weapons.

Selecting Units
Also included, are scripts to allow the selection of a unit. This allows a visualization of what units are on its target list, and the ability to manually edit its target list in real-time.