Targeting Indicators vs 2.0

Go to
http://www.avariceonline.com/targeting
For documentation, demo and video

Summary :

This package includes all the components to present a rich targeting system for your unity game.

Design Features

- Collider based detection and selection of targets.
- Runtime coloring of indicators based on arbitrary 'code'.
- Minimap and 'Heads up' Indicators included.
- 3d Quad's are used, this enables full support with Unity's animation framework.
- Support for stealth, and stealth areas.

Code Features

- Clean C# coding with comments and html documentation.
- Separation of Indicators (visuals) from targets (model)
- Object pooling included

Initial Setup

This release includes a simplified setup process.

1) Import Package

First thing to do I import the package. You will need to content of the “***Common***” directory. This directory includes common helpers and utilities that I move between projects. This includes the object pool. You will need the ***Targeting*** directory, this includes everything else. You may also want to import the **“*TargetingDemo*”,** this includes the demo scene.

2) Script Execution Order

The system has some script dependencies. Please set the script execution orders.

ObjectPoolService	-1000
TargetService	-500
Target	+500

3) Layers

In order to get indicators to show up on different cameras, we use layers. Please define the following.

- Layer 8 “Hud”
  - Visible only to the Hud Camera.
  - “Heads Up” indicators should use to this layer
- Layer 9 “Map”
  - Visible only to the Map Camera.
  - Map indicators should use to this layer
- Layer 10 “Terrain”
  - Visible only to the Main and Map Cameras.
  - The terrain should use to this layer
- Layer 11 “Target Scanners”
  - Layer for the Target Detector script
  - Set the physics layer to interact with “Targets”
- Layer 12 “Targets”
  - Layer for the target objects
  - Set the physics layer to interact with “Target Scanners”

3) Tags

In order to correctly position heads up indicators, the indicators need access to the HudCamera. Please make sure you have a HudCamera and include the HudCamera tag.

- HudCamera

4) Physics Setup

Targets uses the Unity3d physics system to facilitate interactions. This means that the Targets and the Target Scanners need to have rigid-bodies and collides attached. If these objects move manually (not using the physic system) be sure to set them as IsKinematic.

- The collider attached to the target should match the size of the sprite or mesh. You may want to make the target component a child object to the game play components, but, this is not required.
- The TargetScanner script (child to the player controller) uses a collider to detect the incoming targets. You may re-size the collider to grow / shrink the area of detection.