Script Dependency Visualizer V1.1
Black Pea Studios

Thank you for purchasing the package.

New in V1.1:

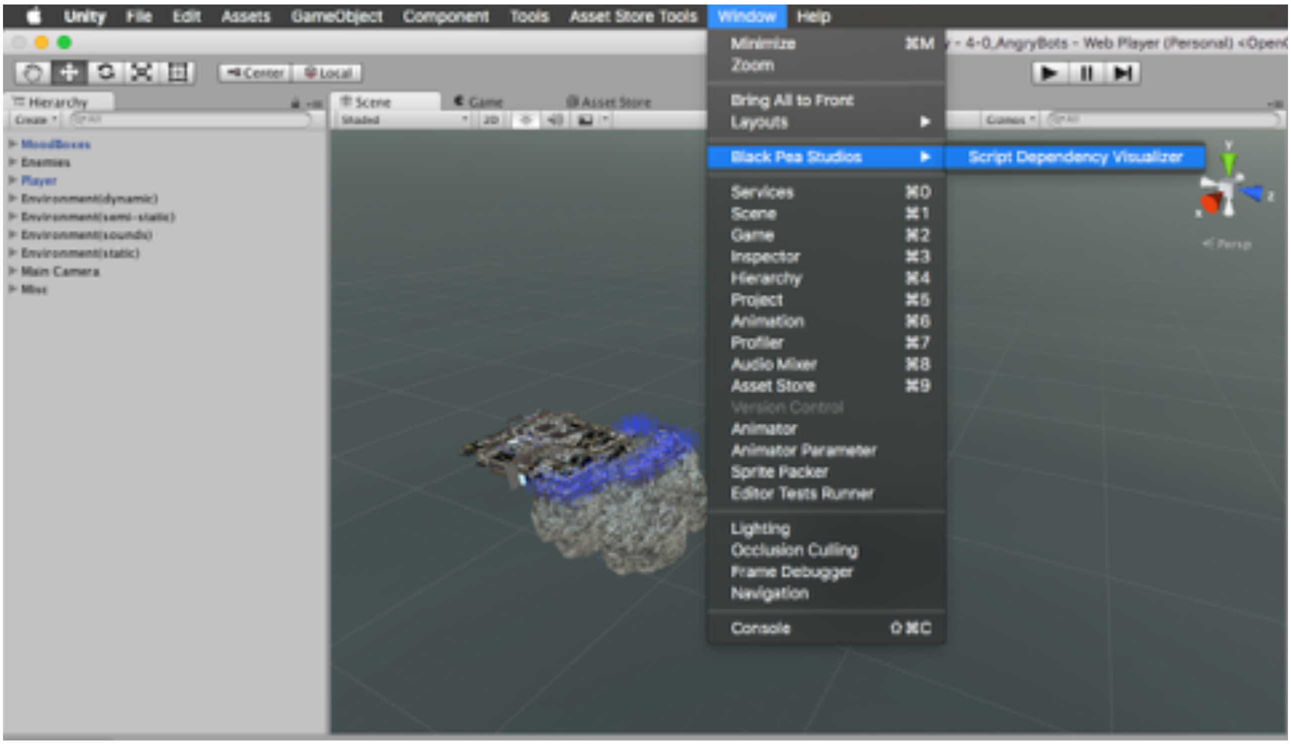
- Save Layout per scene
- Double Click to open Monobehaviours
- Some Adjustments to the Dependency Core
- Minor Layout Changes

Here I will briefly explain how the asset works. If you have any further queries, please visit our website at www.blackpeastudios.co.za or contact us at info@blackpeastudios.co.za.

Please remember to review the asset. It helps a lot in terms of continuing the development of the package.

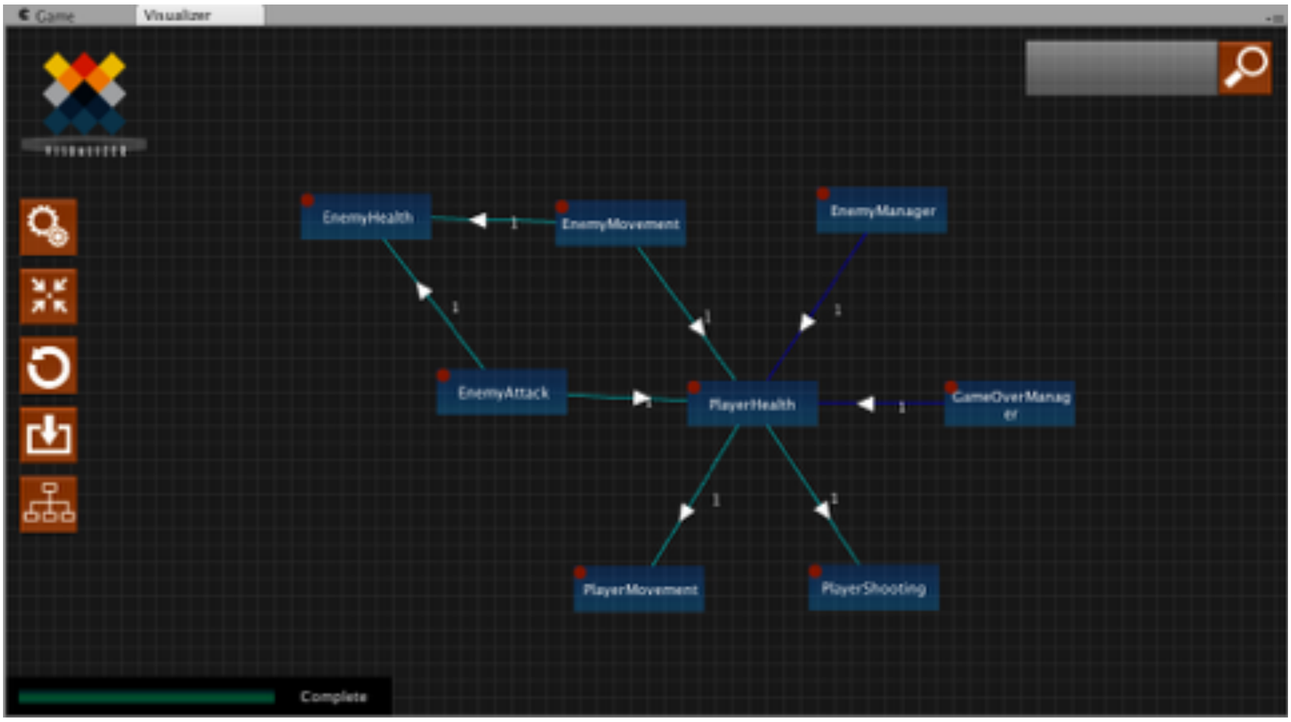
A) Setup

There is no setup required for this asset. Simply import the package into your project and open the scene you wish to inspect.
NB: If there are no MonoBehaviours in your current scene, the visualiser will have nothing to show.
Now just go to Window/Black Pea Studios/Script Dependency Visualizer as seen in the image below.



B) What am I looking at?

Now you should see a window appear that looks something like this:



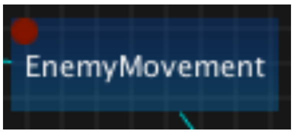
What you are seeing here is Unity's Survival Shooter's main scene.

What this asset does is it looks for all MonoBehaviours in the scene and prefabs in your assets folder.
It then inspects these using Reflection to find out which other scripts they reference and draws edges to those nodes.

There are three types of edges:

1) Pure Dependencies
    1) Light Blue -> Private Instance Field
    2) Dark Blue -> Public Instance Field
    3) Pale Yellow -> Private Static Field
    4) Yellow -> Public Static Field
2) Composite Dependencies
    1) Brown -> Multiple Pure Dependencies
3) Two-way Dependencies
    1) Red -> With arrows pointing to both

Nodes that have the red dot in its top left corner, as shown in the image below, can be double-clicked to open it in MonoDevelop.



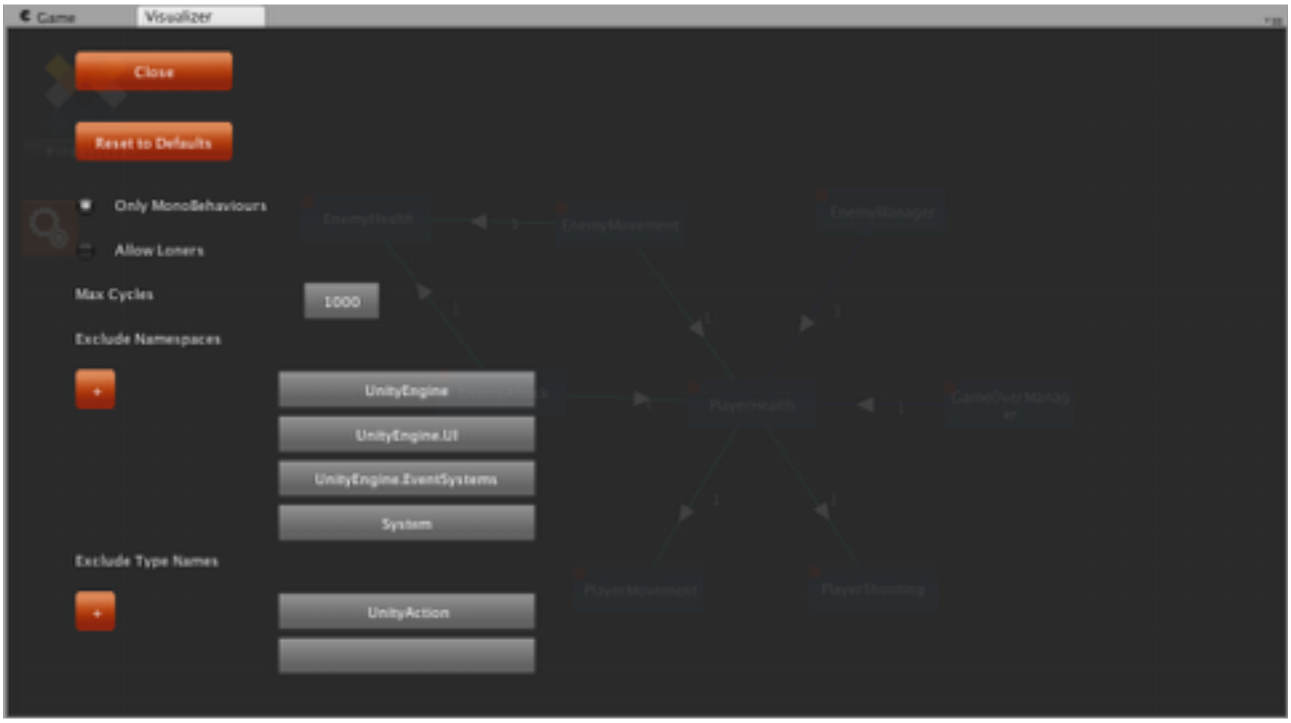
C) Layout

The asset has a move-by-force layout manager that attracts and repulses the nodes to and from each other depending on how they are connected. Although this works nice for smaller projects, with bigger projects that has more than 100 nodes, it can take a while for the process to complete. If you go into settings, you can change the number of cycles that it will go through before it stops forcing the layout. Also note, that it stops either after the max cycles has been reached, or if the graph has reached equilibrium.

From there onwards, you can click and drag the nodes to lay them out as you wish. You also have the option to save the layout if you wish so that you can quickly just open it up to view.

D) Settings

The settings page has a couple of options you can use to configure how the graph is set up.



● Only MonoBehaviours

○ Allow Loners

Max Cycles 1000

Exclude Namespaces

UnityEngine
UnityEngine.UI
UnityEngine.EventSystems
System

Exclude Type Names

UnityAction

Un-ticking “Only MonoBehaviours” will prompt the system to look deeper into the scripts to find Types that does not inherit from MonoBehaviour specifically.

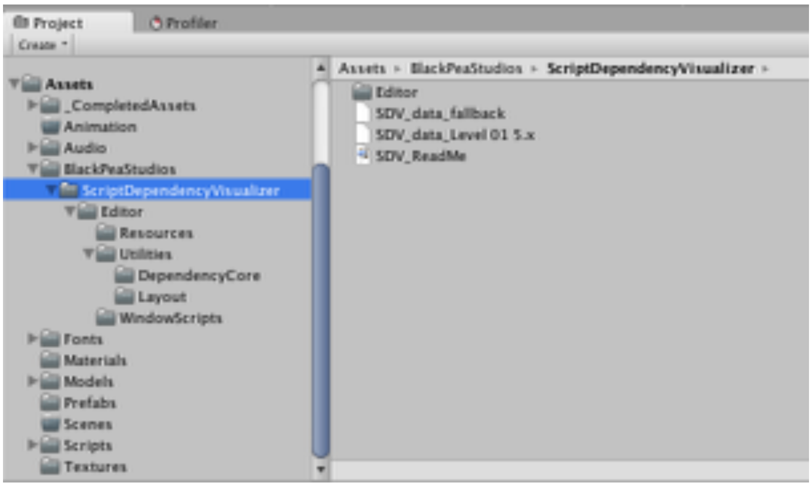
Ticking “Allow Loners” will tell the system to allow Scripts that does not have any relevant dependencies to other Types in the scene. This cleans up the graph nicely in some cases.

You can also set Max Cycles. This is when the move-by-force layout stops trying to reach its equilibrium.

Lastly, you can exclude Namespaces and Types by adding them to the text fields.

E) Saving Layouts

We have implemented saving functionality per scene. The first time a layout is generated for a scene, that layout will be saved to the BlackPeaStudios/ScriptDependencyVisualizer folder under SDV_data_[scene name] as shown here:



If the scene is not saved, the Serializer will complain about this in the console and save that layout in the SDV_data_fallback file.

— — — —

That’s the Script Dependency Visualizer package.

We hope you enjoy the product as much as we enjoyed creating it. We would still like to add some features, so if you have any questions, comments or suggestions, please feel free to visit our website at www.blackpeastudios.co.za or send us a mail at info@blackpeastudios.co.za.

Please remember to review the the package to enable further development and support.

Kind Regards,

Emile Chantler
Black Pea Studios