This note explains how to debug VaM plugins with breakpoints, step-by-step execution and variable inspection.

The C# interpreter itself could be modified in a way to to provide Soft Debugger capability. It means that upon running the interpreter would open an TCP port and let anything what is connected to the port to debug the program being executed by interpreter

(<u>https://www.mono-project.com/docs/advanced/runtime/docs/soft-debugger/</u>)

To connect and debug I was using dnSpy. Seems like you also can use MonoDevelop or a plugin for VS Code.

So, it short, you need two things:

- 1. dnSpy , which is a free program. You can download it from here ( <u>https://github.com/dnSpy/dnSpy/releases</u> ) or build it from sources.
- 2. Patched mono.dll

Seems like couple of yeas ago patched mono.dll was available for download on dnSpy github, but not anymore. You can still find links to the empty pages where mono.dll was . These days the only thing available is the instruction on how to compile mono.dll yourself.

#### https://github.com/dnSpy/dnSpy-Unity-mono

This tool provides a patcher which constructs source code for mono.dll. The accompanying README.md is quite cryptic and incomplete. More complete version could be found here: <u>https://www.unknowncheats.me/forum/unity/394581-patching-mono.html</u>, it is still hard to follow for my taste. My instruction follows:

So, lets start

- 1. I would assume the working folder is d:/prj and you have more than 10Gb of free space available.
- 2. I assume you have git installed and it is configured to work from any folder
- 3. From the directory d:/prj clone dnSpy's mono patcher: git clone https://github.com/dnSpy/dnSpy-Unity-mono.git
- 4. Also clone mono from Unity's repo: git clone --recurse https://github.com/Unity-Technologies/mono.git
- 5. make copy of umpatcher: copy folder d:\prj\dnSpy-Unity-mono\src\umpatcher to be d:\prj\ umpatcher . Do not compile it in place. Do not modify cloned git repos in any way.
- 6. Open D:\prj\umpatcher\umpatcher.sln with VS2019 and Build Solution . Lets assume you have build Debug version of it and the exe appeared at d:\prj\umpatcher\umpatcher\bin\Debug\net48\ umpatcher.exe
- 7. The original instruction discusses how to determine unity version and its mono version. You may want to do that for other games, but for VaM I would just list known values: Unity is 2018.1.9 and mono version is whatever b4d827005179f19e15114c02b564b448e38e8551 githash takes you to.
- 8. You need to set your git name and git email. Umpatcher seems to do some weird merge and it will fail if name and email are not set. To check if the name is already set, do git config --list and look for user.name and user.email lines. It those are not there, do git config --global user.name "My Name"

git config --global user.email "myemail@example.com"

9. now run umpatcher:

d:\prj\umpatcher\umpatcher\bin\Debug\net48\umpatcher.exe 2018.1.91 ae4c17e9c4fc0c648db0d19be1c028bab1a10dc1 "d:\prj\mono" "d:\prj\dnSpy-Unity-mono" do not use 2018.1.9 for a name. It will fail with an error. umpatcher runs for every version of Unity and checkouts different mono version into each folder. We don't need all that, but this is the way to trigger it to make desired sln file.

#### 10. In case umpatcher fails:

Umpatcher is not an idempotent tool: when run twice it will fail second time. If it fails halfway, you need to revert changes it made. So, before running umpatcher again, run: cd d:/prj/dnSpy-Unity-mono rmdir /s /q unity-2018.1.91 git checkout master git reset --hard b4d827005179f19e15114c02b564b448e38e8551

It is useful to make a .bat file with these lines if you wish to debug umpatcher.

- 11. If umpather finishes succesfully it creates sln files
- 12. Open d:\prj\dnSpy-Unity-mono\dnSpy-Unity-mono-v2018.x.sln with Visual Studio 2019
- 13. Choose Release\_eglib for target and x64 for platfom
- 14. Build !
- 15. Go d:\prj\dnSpy-Unity-mono\builds\Release eglib\unity-2018.1.9\win64 to pick up mono.dll

### VaM

Now copy that mono.dll to VaM/Mono/EmbedRuntime . Do not forget to back up the original file.

Run VaM. Here you may want to check that mono indeed have opened tcp port 55555 at loopback: netstat -a -b -n -p tcp

Run dnSpy

In windows file explorer open VaM\VaM\_Data\Managed and locate Assembly-CSharp.dll . Drag and drop it into the left pane of dnSpy.

In dnSpy Debug  $\rightarrow$  Start Debuging...

Select Unity (Connect) for a debug engine. Leave IP address empty and port 55555.

Click ok

Now lets make the first breakpoint. For beginnings lets make it in VaM itself. In the left pane unroll Assembly-CSharp and find default namespace denoted with single dash. It is at the very top of the list. Find SuperController and unroll. In the right pane press Ctrl+F to search and type "KeyCode.P". Look at line "this.gameMode = SuperController.GameMode.Play;" and set breakpoint by clicking on the gray strip at the left of line numbers.

Now switch to VaM windows and press key "P". VaM will freeze.

### Now, this is how it should look like:

mbly Explorer 🛛 🔻 🗙	SuperController $ imes$		
🔺 🏤 SuperController @020 🔺	15994		
Base Type and Interpreter Deviced Type and Interpreter Deviced Type and	15996 if (!this.UIDisabl	ed)	
SuperController():	15997 { 15998 if (Input.GetK	eyDown(KeyCode.E))	
ActivateWorldUI()	15999 {	de - SuperController GameMode Edit:	
AddAtom(Atom, s     AddAtomByPopul	16001 }	Superconcrotten. Gamenoue. Eurcy	
AddAtomByType(	16002 if (Input.GetK 16003 {	eyDown(KeyCode.P))	
AddAtomByType(	16004	ode = SuperController.GameMode.Play;	
AddAtomByTypeF     AddCapyac(Capya	16005 } 16005 if (Input GetV		
AddCanvas(Canva	16007 {	eybown(keycode:hy)	
AddKey() : void @	16008 this.Cycle	<pre>SelectAtomOfType("Person");</pre>	
💁 AddPositionRotati	16010 if (Input.GetK	eyDown(KeyCode.U))	
AddVarPackageRe     AdjustNevinetier	16011 {		
Adjustivavigationr     AlignRigAndContr	16012 this.logg1	eMainHUDMonitor();	
	16014 if (Input.GetKeyDown(KeyCode.T))		
AlignRigFacingSel	16015		
🔍 AllocateOverlappi	Locale		
♥↓ AllocateRaycastHi	Locals	Velue	Tura
ArmAllControlled		Value	SuperController
Assignt II Camera(	ectivel II	SelectedOntions	SuperController Activel II
AtomParentChance	advancedSceneEditDisabled	falce	bool
AtomSubSceneCh	alignRotationOffset	l eft	SuperController AlignBotatio
ଦ୍ଧ Awake() : void @0	allowGrabPlusTriggerHandToggle	true	bool
♥ <sub>★</sub> BuildFilesToMigrat	allowHeadPossessMousePanAndZoom	false	bool
𝔤 BuildMigrationMa	allowPossessSpringAdjustment		bool
	anoth ossessophing/tajasement		bool
Pa BuildObsoleteDire	always IseAlternateHands	talse	0001
♥a BuildObsoleteDire Ø CancelMigrateFile	AlwaysUseAlternateHands	talse	hool
<ul> <li>♥a BuildObsoleteDire</li> <li>♥ CancelMigrateFile</li> <li>♥ CancelRemoveOb</li> </ul>	alwaysUseAlternateHands     assetManagerReady     atomContainerTransform	talse true /SceneAtoms (Unit/Engine Transform))	bool UnityEngine Transform
<ul> <li>♥a BuildObsoleteDire</li> <li>♥ CancelMigrateFile</li> <li>♥ CancelRemoveOb</li> <li>♥a CheckAutoConne</li> </ul>	alwaysUseAlternateHands     assetManagerReady     systematication ContainerTransform     autoErceachaimstionOnSwitchToEditMode	talse true {SceneAtoms (UnityEngine.Transform)}	bool UnityEngine.Transform
<ul> <li>♥a BuildObsoleteDire</li> <li>♥ CancelMigrateFile</li> <li>♥ CancelRemoveOb</li> <li>♥ CheckAutoConne</li> <li>♥ CheckHoldLoad()</li> </ul>	AlwaysUseAlternateHands     AssetManagerReady     AssetManagerReady     AssetManagerReady     AssetManagerReady     Atom ContainerTransform     AutoFreezeAnimationOnSwitchToEditMode     StatutoSimulation	false true {SceneAtoms (UnityEngine.Transform)} false false	bool UnityEngine.Transform bool
<ul> <li>♥a BuildObsoleteDire</li> <li>♥ CancelMigrateFile</li> <li>♥ CancelRemoveOb</li> <li>♥a CheckAutoConne</li> <li>♥a CheckHoldLoad0</li> <li>♥a CheckIfController</li> </ul>	AlwaysUseAlternateHands     AssetManagerReady     SatureContainerTransform     autoFreezeAnimationOnSwitchToEditMode     autoSimulation	false true {SceneAtoms (UnityEngine.Transform)} false false false	bool UnityEngine.Transform bool bool
<ul> <li>♥a BuildObsoleteDire</li> <li>♥ CancelMigrateFile</li> <li>♥ CancelRemoveOb</li> <li>♥ CheckAutoConne</li> <li>♥ CheckHoldLoad0</li> <li>♥ CheckHoldLoad0</li> <li>♥ CheckIcOntrollerl</li> <li>♥ CheckLoadinglcor</li> </ul>	AlwaysUseAlternateHands     AssetManagerReady     AssetManagerReady     AstomContainerTransform     autoFreezeAnimationOnSwitchToEditMode     fautoSimulation     forowseDisabled     bocsceDisabled	talse true {SceneAtoms (UnityEngine.Transform)} false false false out	bool UnityEngine.Transform bool bool UnityEngine.Transform
<ul> <li>⊕ BuildObsoleteDire</li> <li>⊕ CancelMigrateFile</li> <li>⊕ CancelRemoveOb</li> <li>⊕ CheckAutoConne</li> <li>⊕ CheckHoldLoad0</li> <li>⊕ CheckIcoatinglcor</li> <li>⊕ CheckLoadinglcor</li> <li>⊕ CheckLoadinglcor</li> <li>⊕ CheckLoadinglcor</li> </ul>	AlwaysUseAlternateHands     AssetManagerReady     AssetManagerReady     AdomContainerTransform     AutoFreezeAnimationOnSwitchToEditMode     AutoSimulation     Action Statement And Ceft     AccenterHandLeft	talse true (SceneAtoms (UnityEngine.Transform)) false false false null	bool UnityEngine.Transform bool bool UnityEngine.Transform UnityEngine.Transform
<ul> <li>⊕ BuildObsoleteUre</li> <li>⊕ CancelMigrateFile</li> <li>⊕ CancelRemoveOb</li> <li>⊕ CheckAutoConne</li> <li>⊕ CheckHoldLoad0</li> <li>⊕ CheckIcontrollerl</li> <li>⊕ CheckLoadingIcor</li> <li>⊕ CheckBoadingCor</li> <li>⊕ CheckResumeAuti</li> <li>⊕ CheckResumeAuti</li> </ul>	<ul> <li>JavayayUseAlternateHands</li> <li>JasetManagerReady</li> <li>J. assetManagerReady</li> <li>J. atomContainerTransform</li> <li>JautoFreezeAnimationODSwitchToEditMode</li> <li>JautoFreezeAnimationODSwitchToEditMode</li> <li>JowseDisabled</li> <li>J. centerHandLeft</li> <li>J. acentrelHandLeft</li> </ul>	talse true {SceneAtoms (UnityEngine.Transform)} false false false null null 0	bool UnityEngine.Transform bool bool UnityEngine.Transform UnityEngine.Transform floot
<ul> <li>⊕ BuildObsoleteDire</li> <li>⊕ CancelMigrateFile</li> <li>⊕ CancelRemoveOb</li> <li>⊕ CheckAutoConne</li> <li>⊕ CheckHoldLoad()</li> <li>⊕ CheckIControlleri</li> <li>⊕ CheckLoadinglcor</li> <li>⊕ CheckMessageAn</li> <li>⊕ CheckResumeRen</li> <li>⊕ CheckResumeRen</li> </ul>	<ul> <li>JavayayUseAlternateHands</li> <li>JassetManagerReady</li> <li>J. assetManagerReady</li> <li>J. atomContainerTransform</li> <li>J. autoFreezeAnimationOnSwitchToEditMode</li> <li>J. autoFinulation</li> <li>J. browseDisabled</li> <li>J. centerHandRight</li> <li>J. controllerScale</li> </ul>	talse true (SceneAtoms (UnityEngine.Transform)) false false false null null 0	bool UnityEngine.Transform bool bool UnityEngine.Transform UnityEngine.Transform float
<ul> <li>⊕ BuildObsoleteDire</li> <li>⊕ CancelMigrateFile</li> <li>⊕ CancelRemoveOb</li> <li>⊕ CheckAutoConne</li> <li>⊕ CheckHoldLoad()</li> <li>⊕ CheckIcoatinglcor</li> <li>⊕ CheckResumeAut</li> <li>⊕ CheckResumeAut</li> <li>⊕ CheckResumeRen</li> <li>⊕ CheckResumeRim</li> <li>⊕ CheckResumeRim</li> </ul>	AlwaysUseAlternateHands     A assetManagerReady     A assetManagerReady     A atomContainerTransform     autoFreezeAnimationOnSwitchToEditMode     f autoSimulation     f autoSimulation     f browseDisabled     P a centerHandLeft     P a centerHandRight     f controllerScale     f creatorKey     f creatorKey	talse true (SceneAtoms (UnityEngine.Transform)) false false false null null 0 "C093426"	bool UnityEngine.Transform bool bool UnityEngine.Transform UnityEngine.Transform float string
<ul> <li>⊕ BuildObsoleteUre</li> <li>⊕ CancelMigrateFile</li> <li>⊕ CancelRemoveOb</li> <li>⊕ CheckHoldLoad0</li> <li>⊕ CheckHoldLoad0</li> <li>⊕ CheckHoldLoad0</li> <li>⊕ CheckHosageAni</li> <li>⊕ CheckResumeAnti</li> <li>⊕ CheckResumeSim</li> <li>⊕ CheckResumeSim</li> <li>⊕ CheckSwapAxip 0:</li> </ul>	<ul> <li>JavayayUseAlternateHands</li> <li>JassetManagerReady</li> <li>JassetManagerReady</li> <li>JautoFreezeAnimationOnSwitchToEditMode</li> <li>JautoSimulation</li> <li>JavayayDisabled</li> <li>JavayayDisabled</li></ul>	talse         true         (SceneAtoms (UnityEngine:Transform))         false         false         false         null         0         "C093426"         null	bool UnityEngine.Transform bool bool UnityEngine.Transform UnityEngine.Transform float string string
<ul> <li>GuiddObsoleteDire</li> <li>CancelMigrateFile</li> <li>CancelRemoveOb</li> <li>CheckHoldLoad0</li> <li>CheckHoldLoad0</li> <li>CheckLoadinglcor</li> <li>CheckLoadinglcor</li> <li>CheckLoadinglcor</li> <li>CheckResumeAuti</li> <li>CheckResumeAuti</li> <li>CheckResumeSim</li> <li>CheckResumeSin</li> <li>CheckResumeSin</li> <li>CheckResumeSin</li> <li>CheckResumeSin</li> <li>CheckResumeSin</li> <li>CheckResumeSin</li> <li>CheckResumeAxis0:</li> <li>CheckResumeAxis0:</li> <li>CheckResumeSin</li> </ul>	JalvaysUseAlternateHands JalvaysUseAlternateHands JasetManagerReady J. assetManagerReady J. autoFreezeAnimationOnSwitchToEditMode JautoSimulation	talse true (SceneAtoms (UnityEngine.Transform)) false false false null 0 "C093426" null null 0	bool UnityEngine.Transform bool bool UnityEngine.Transform UnityEngine.Transform float string string string string
<ul> <li>BuildObsoleteDire</li> <li>CancelMigrateFile</li> <li>CancelRemoveOb</li> <li>CheckHoldLoad0</li> <li>CheckHoldLoad0</li> <li>CheckLoadinglcor</li> <li>CheckLoadinglcor</li> <li>CheckLoadinglcor</li> <li>CheckResumeAuti</li> <li>CheckResumeRen</li> <li>CheckResumeSim</li> <li>CheckResumeSim</li> <li>CheckSwapAxis()</li> <li>ClearAllGrabbedCi</li> <li>ClearAllGrabbedCi</li> <li>ClearAllU()</li> <li>void</li> </ul>	<ul> <li>JavayayUseAlternateHands</li> <li>JavayayUseAlternateHands</li> <li>assetManagerReady</li> <li>JautoFreezeAnimationOnSwitchToEditMode</li> <li>autoSimulation</li> <li>browseDisabled</li> <li>JaccenterHandLeft</li> <li>JaccenterHandRight</li> <li>centerHandRight</li> <li>centerHandRight</li> <li>centerIsaveDir</li> <li>currentSaveDir</li> <li>currentSelectMode</li> </ul>	talse true (SceneAtoms (UnityEngine.Transform)) false false false null null 0 "C093426" null null false fals	bool UnityEngine.Transform bool bool UnityEngine.Transform UnityEngine.Transform float string string string SuperController.SelectMode
<ul> <li>BuildObsoleteUre</li> <li>CancelMigrateFile</li> <li>CancelRemoveOb</li> <li>CheckAutoConne</li> <li>CheckHoldLoad)</li> <li>CheckIControlleri</li> <li>CheckIControlleri</li> <li>CheckMessageAni</li> <li>CheckResumeAuti</li> <li>CheckResumeRen</li> <li>CheckResumeSim</li> <li>CheckSwapAxis0:</li> <li>ClearAllGrabbedCi</li> <li>ClearAllGrabbedCi</li> <li>ClearAllGrabbedCi</li> <li>ClearFirers0::void</li> <li>ClearFirerS0::void</li> </ul>	JalvaysUseAlternateHands JalvaysUseAlternateHands JavaysUseAlternateHands JavatoFreezeAnimationOnSwitchToEditMode JautoFreezeAnimationOnSwitchToEditMode JautoFreezeAnimationOnSwitchToEditAnim	talse         true         (SceneAtoms (UnityEngine.Transform))         false         false         false         null         0         "C093426"         null         Olff         false	bool UnityEngine.Transform bool bool UnityEngine.Transform UnityEngine.Transform float string string string SuperController.SelectMode bool
<ul> <li>BuildObsoleteUre</li> <li>CancelMigrateFile</li> <li>CancelRemoveOb</li> <li>CheckAutoConne</li> <li>CheckHoldLoad()</li> <li>CheckIControlleri</li> <li>CheckLoadinglcor</li> <li>CheckRessageAni</li> <li>CheckRessageAni</li> <li>CheckRessmeAuti</li> <li>CheckResumeRen</li> <li>CheckResumeSim</li> <li>CheckSwapAxis():</li> <li>ClearAllGrabbedC</li> <li>ClearFireSP void</li> <li>ClearFireSP ovoid</li> <li>ClearFireSP ovoid</li> <li>ClearFireSP ovoid</li> </ul>	JalvaysUseAlternateHands JalvaysUseAlternateHands JassetManagerReady JassetManagerReady JautoFreezeAnimationOnSwitchToEditMode JautoSimulation	talse         true         (SceneAtoms (UnityEngine.Transform))         false         false         null         null         0         "C093426"         null         Off         false         false         false	bool UnityEngine.Transform bool bool UnityEngine.Transform UnityEngine.Transform float string string SuperController.SelectMode bool bool bool bool bool bool bool boo

You can do step-by-step debugging and see local variables.

## Plugins

Plugings are compiled at run time, so there are no place to put breakpoint to.

This would not do anything:

```
if(System.Diagnostics.Debugger.IsAttached) System.Diagnostics.Debugger.Break();
```

And just this would trigger VaM to terminate:

```
System.Diagnostics.Debugger.Break();
```

My idea is to trigger infinite loop by keypress and pause the program. Put this into the beginning of an Update function of a plugin:

```
if (Input.GetKeyDown(KeyCode.F9))
{
         bool stopFlag = true;
         while(stopFlag);
}
```

Drop the plugin .cs file to the left pane of dnSpy. Run that plugin and press F9. VaM will freeze. Switch to dnSpy and pause the execution: From here you can put up breakpoints. Change flag to false to continue running VaM.



# dnSpy

Sometimes dnSpy will be unable to find the main unity thread. Sadly dnSpy does not have a tool to select a thread manually. The solution is to build dnSpy from sources with a little patch:

git clone --recurse https://github.com/dnSpy/dnSpy.git then find dnSpy\Extensions\dnSpy.Debugger\dnSpy.Debugger.DotNet.Mono\Impl\ DbgEngineImpl.Threads.cs

, inside find IsMainThread and replace it with

then build