Xamarin Studio integration

1. **Reasoning**

Written due to MonoDevelop poor support for C# 6 and general crashes and bugs during development. Speeds up development of C# CryEngine plugins by providing few helper functions, and better debugging.

1. **User guide**

There are few changes from original MonoDevelop plugin. Most important one is new options pane.

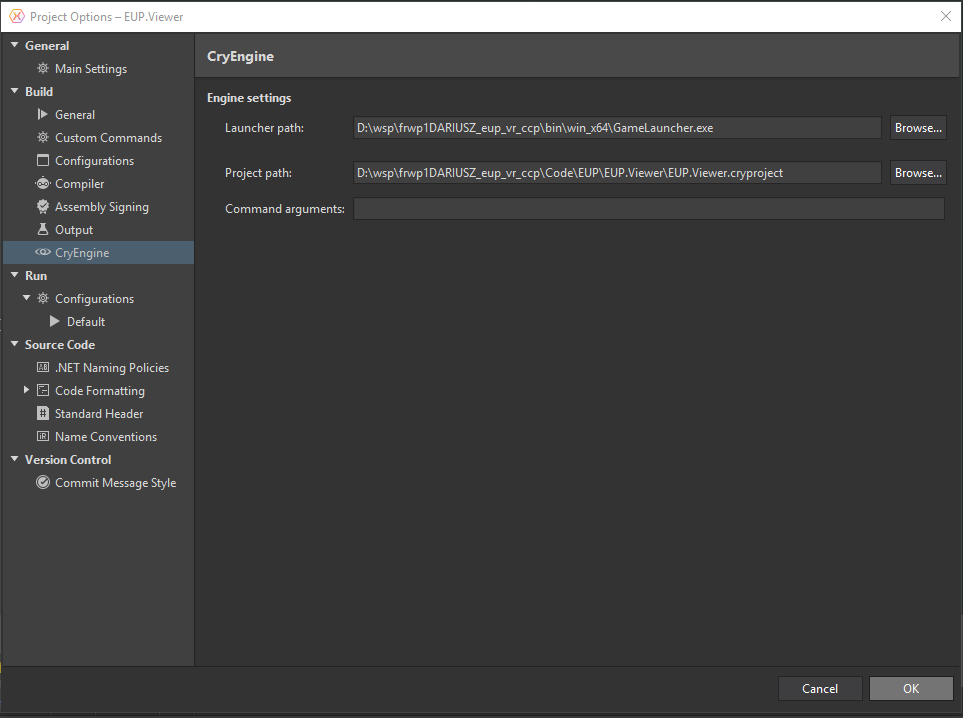


Figure CryEngine options pane

With new version, we only need to pass LauncherPath (our game .exe) and project path (.cryproject). Eventually maybe only project path will be required if we can extract launcher data from it.

Other feature are launch targets, right now it supports normal GameLauncher with VR and non-VR setting (VR one simply sends cvar sys\_vr\_support = 1 to the engine). Also Sandbox has been added so we can debug C# plugin code inside of Sandbox as well.

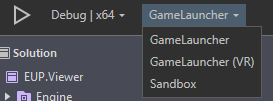


Figure Launch targets

Lastly, from project menu (right-click on project node in solution), we can directly navigate to project folder, engine root folder. We can also compile resources (right now asset path is hardcoded to CCP).

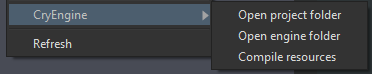


Figure Project utilities

1. **Technical information**

Plugin is written in C# using MonoDevelop Addin maker plugin that has to be installed to open project. Plugin project itself is located in Code/Tools/MonoDevelopAddins/MonoDevelop.CryEngine folder. By default .dll is output to proper plugin location in Tools/XamarinStudio/AddIns/MonoDevelop.CryEngine.

In plugin code folder Properties you will find file “Manifest.addin.xml” which is main configuration point for plugin, and one that is responsible for wiring up extensibility points to their proper implementations.

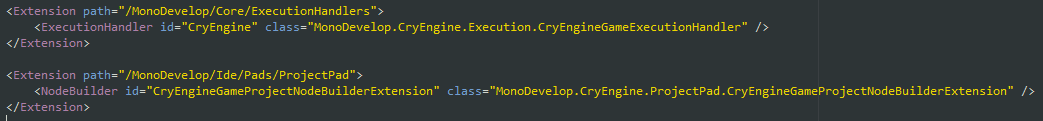


Figure Example of extensibliity points

On the upper picture you can see that we register custom handler for ExecutionHandlers and ProjectPad. That allows to customize those parts of the editor (more info about those points can be found in source code of MonoDevelop on github), by setting full namespace and class name of a type that will implement handlers for requested functionality. You can find those types by simply searching for them in source code.

From debugging side engine behaviour was changed, now XamarinStudio when launching CryEngine project will send random port number (range 17615 – 18000) to it and start debugger session on that port. That allows to debug multiple instances of CryEngine independently, as before engine will only broadcast on single port, leading to unexpected behaviour when launching multiple projects).