Developers documentation Enigma Suite 2022.0.1

Choices made

Language

I will use C# 10 from Microsoft.

After experimenting with Java, Kotlin, Free Pascal and Delphi I found that C# has the following advantages:

- Excellent support for the GUI, even better than Delphi and Free Pascal. And much better than JavaFX. I did not compare with Jetpack Compose for Desktop (as used by Kotlin) as this is still in its infancy.
- A very good language, in my opinion on the same level as Kotlin and better than Java. Even if I prefer the Pascal syntax of FP and Delphi, I believe that C# does a better job.
- Available functionality like Dependency Injection and Mocking. This is an advantage compared to FP (no DI and mocking) and Delphi (one good library Spring4D, but only one maintainer).
- Documentation. Extensive and very good.
- Integration: a lot of functionality is directly available in the Community Edition of Visual Studio. For Java and Kotlin your need to select many external libraries (sometimes frameworks). YOu can do the same in Visual Studio but it will often not be necessary.
- Long term availability. VS is available in a free edition for a very long time, the community edition for Delphi at this moment for about three years. Free versions of Delphi have been discontinued in the past. Of course, this is only an advantage compared to Delphi as FP, Java and Kotlin are freely available.

Of course, there are also some drawbacks:

- The use of .NET is required. This is not a big problem anymore for installations, as the .NET environment can be included in the installer. But it seriously enlarges the footprint. This advantage is only compared to FP and Delphi. Of course, Java and Kotlin have the same problem.
- Performance could be less than the performance of FP or Delphi because of the use of the .NET environment. I intend to do some benchmarks.

The selection for C# is mainly based on the support for the GUI, the quality of the language and the documentation, and the expected long-term availability.

Unit testing

I will use MS Test.

In Visual Studio the following solutions for unit testing are available.

- xUnit
- nUnit
- MSTest

MSTest is the standard solution from Microsoft. I did not investigate this thoroughly but after a short search on the Internet it appears that nUnit and MSTest are comparable. xUnit has some specifics, like a less understandable syntax. I will use MS Test , mainly because the integration with VS is pretty much guaranteed.