
Meeting Report

3 January 2023

Attendees

François GUTHERZ, CTO & Project leader HARFANG®3D
Emmanuel JULIEN, Lead developer HARFANG®3D

Agenda of the day

- Why they create HARFANG®3D
- Overview of the project
- Questions/Answers
 1. Have you already tried to implement F# in FABGen ?
 2. Why do you want to implement F# in FABGen ?
 3. For what purpose did you create HARFANG®3D ?
 4. Do you have an email address or other way of communication ?
 5. Do we have any key points to follow when developing the project ? (syntaxe, performance...)
 6. How will we know if the project is a success?
 7. Who is your audience?
- Open Discussion

Meeting Summary

- Overview of the project :

FABGen was written for the HARfang3D project to bring the C++ engine to languages such as Python, Lua and Go? IT was written as a replacement for SWIG, a very well-known binding generator supporting a lot of target languages. SWIG has

different issues and that's why we create another binding generator. And Your objective is to create a binding for F#.

To create a correct solution, it is essential to have a deep understanding of C++ as well as the target language (such as Python, Lua, or Rust) and its feature set. Cutting corners or taking shortcuts can lead to problems such as a core dump or memory leak in the user's program later on. It's important to approach this task like building Jenga blocks - each piece must be correct from every angle, as you don't know how they will fit together in the user's program.

We giving you this roadmap like a tips to build your implementation :

- Create a mapping of elementary types
- Implement a C API wrapping the c/C++ objects
- i - Improve the integration with the target language

■ Questions/Answers :

- **Have you already tried to implement F# in FABGen ?**

No, it's new for us. We already implemented FABGen in Go, Python and Lua.

- **Why do you want to implement F# in FABGen ?**

We want to implement F# in FABGen to allow non-coding experts to have access to our software. C++ is a very specific language for lambda people so implementing other languages such as Python, F# and Rust allows other experts to easily use the 3D engine.

- **For what purpose did you create Harfang 3D ?**

We created Harfang to address the need for 3D engines in the automotive industry, civil/defense industry.. as these industries have strong technical requirements and need to meet certain safety certifications. Additionally, the 3D engine needs to be embeddable and work with custom hardware, as well as be efficient in terms of power consumption. Harfang may also be designed to meet certain industry standards such as ISO, MISRA, and autosar. Harfang was also created to visualize and manipulate complex data more rapidly.

- **Do you have an email address or other way of communication ?**

contacts: francois.gutherz@harfang3d.com
emmanuel.julien@harfang3d.com

- **Do we have any key points to follow when developing the project ? (syntaxe, performance...)**

One of the key points to respect is that we cannot allocate new memories during the compilation of the code. We cannot use malloc such as "New" in C. Like Harfang is used mainly for industry it is mandatory to deal with this. Project clarification: don't do your F# implementation trying to respect the MISTRA certification. It's nearly impossible

■ Open Discussion

- They create 2 different versions of Harfang 1 which respect the MISTRA certification and one much more generic.

- **Are game engines suitable for the industry ?**

Yes because most of the 3D engines are not private and secure in the long term. Also they don't work 100% offline.

Contrary to video game which has a commercial life of 1 to 3 years, An industrial project must be maintained between 5 to 50 years

- **Why did they build Harfang in C++ ?**

Mainly for performance and portability, as C++ can compile almost anywhere (C++11 or 14, at least).

Next Meeting

It is planned to have a weekly call but it is not yet defined with FRANCK JEANNIN.