# Harfang 3D technical specification

---

---

## Overview

The goal of this project is to update FABGen to add F# bindings To do so we will have to understand what FABGen is and how it works, moreover, to make it work properly we will have to use three languages which are:

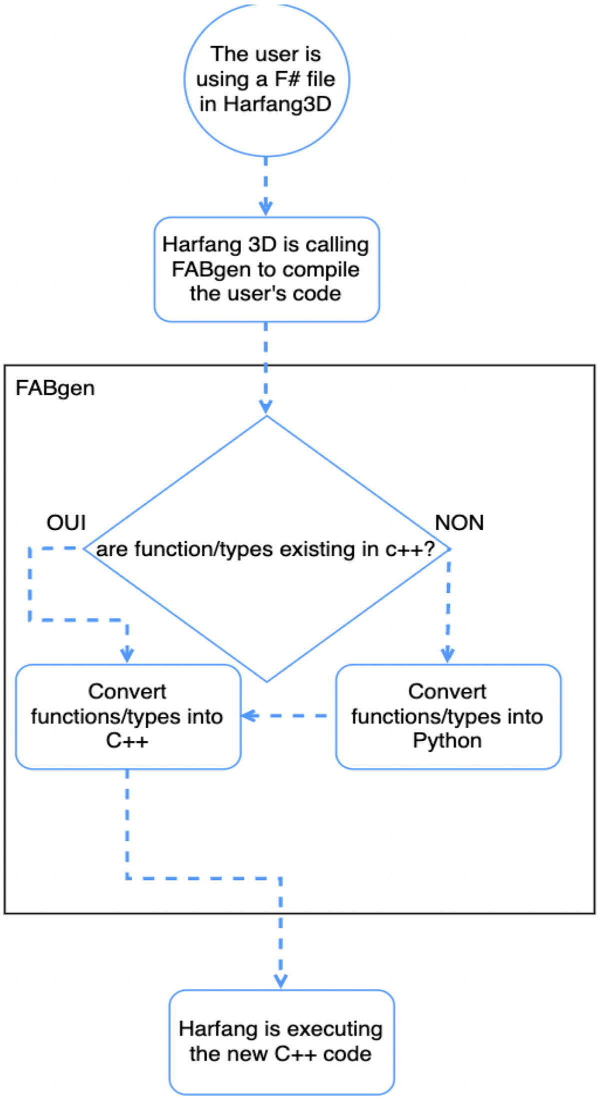
- F#
- C++
- Python

### Why this project?

FABGen is a binding generator whose purpose is to replace SWIG an older binding generator to fit Harfang 3D goals. It is currently working with Python, Lua, and Go. This project aims to add F#.

# Software

## Software workflow


The user is using a F# file in Harfang3D
Harfang 3D is calling FABgen to compile the user's code
FABgen
OUI
NON
are function/types existing in c++?
Convert functions/types into C++
Convert functions/types into Python
Harfang is executing the new C++ code

As you can see above, the user will enter his F# code, then our algorithm will translate the functions in C++, if these functions can't be translated, it will do it in python and then in c++. The remaining code, which is in C++, will finally be send to Harfang 3D.

## File structure

For this project we wanted to use FABgen's original file structure.

Files named "fsharp" and bold files are what we will have to create during the project.

```
├── lang
        ├── __init__.py
        ├── cpython.py
        ├── go.py
        ├── lua.py
        ├── xml.py
        └── fsharp.py
├── lib
        ├── cpython
        │       ├── __init__.py
        │       ├── std.py
        │       └── stl.py
        ├── lua
        │       ├── __init__.py
        │       ├── std.py
        │       └── stl.py
        ├── xml
        │       ├── __init__.py
        │       ├── std.py
        │       └── stl.py
        └── fsharp
                ├── __init__.py
                ├── std.py
                └── stl.py

├── bind.py
├── gen.py
├── license.md
└── readme.md
```

# Risks and constraints

## Risks

This project has few risks, they can be defined as:

The risk of misunderstanding FABGen's functions In the situation where we misunderstood FABgen's function we might produce function that shouldn't be part of our bindings, this will lead to a slowdown of our project and result of being late with our deadlines.

The risk of having a non-organised file structure can lead to bugs or being a problem for updating FABGen with a file structure different than their own.

The risk of having defective tests, this will end up by having a defective product or having a slowdown during the project.

The risk of FABGen's code not being able to run on our machines.

## Constraints

The project has some constraints, which are:

- F# is a statically typed language, so we need to go through the C language to create the binding.
- F# is also different types compared to C++ or Python, to face this problem we can convert types directly to C++ types.

# Testing

Tests are provided by Harfang 3D so we won't have to create them, but we will have to adapt them by changing them to F# tests.

## Footnotes

| word | definition |
|---|---|
| FABGen | FABGen is a script written in Python made to create C++ to allow users to code with multiple languages such as Python, Go, and Lua. |
| Bindings | Bindings refer to the way in which two languages can access and use functionalities of another language. It allows the exchange of data and their functionalities, enabling them to work together. |
| F# | F# is an object-oriented language which means it can associate a block of a program to a concept, F# is widely used to make multiple languages work together. |
| C++ | C++ is a high-level programming language, as F#, C++ is an object-oriented language. |
| Python | Like C++, Python is a high-level language, this language is widely used inside the programming community for his simplicity to use parts of code called libraries which are made by users. |
| Harfang 3D | Harfang 3D is a French company based in Orléans with the objective to create a 3D engine for industrial companies and purposes less explored by his concurrence. Harfang 3D is also the name of their 3D engine. |