

Etude de l'outil de production de documentation Doxygen

DENIZOT Hugo
ISIMA
Université Clermont Auvergne
Année Universitaire 2022 - 2023

Enseignant : David HILL
Date du rendu : 17 Avril 2023

ISIMA

1 Rue de la Chebarde - CS 60032 - 63178 Aubière
Tel : 04 73 40 50 00
Site web : isima.fr

Sommaire

1	Introduction	2
2	Présentation de l'outil	2
3	L'outil dans notre projet	3
3.1	Commenter le code	3
3.2	Appliquer doxygen sur le code	3
3.2.1	Wizard	4
3.2.2	Expert	4
3.2.3	Run	4
3.3	Le rendu de notre documentation	4
3.3.1	Présentation globale	4
3.3.2	Onglets de classes	5
3.3.3	Présentation d'une classe	5
4	Conclusion	6

Table des figures

1	Fenêtre logiciel Doxygen	2
2	Exemple de commentaires	3
3	Configuration de notre partie dot	4
4	Screenshot de notre hierarchie	5
5	Screenshot des méthodes de la classe map	5
6	Screenshot de la classe map	6

1 Introduction

Dans le cadre du cours d'Architecture Logicielle et Qualité, nous avons eu l'opportunité de choisir et d'étudier un outil de développement logiciel. Durant ce rapport nous étudierons ce logiciel et montrerons la place qu'il a eu dans notre projet et comment je l'ai implémenté.

Pour ma part, j'ai choisi de présenter le logiciel de production de documentation **doxygen**, qui est, je trouve, un logiciel intéressant et très utile permettant de créer des documentations détaillées, permettant de mieux comprendre un code que l'on propose.

2 Présentation de l'outil

Doxygen est un outil de production de documentation dont la première version est sortie en 1997. Il permet de créer une documentation à partir d'un code si celui-ci est commenté avec des balises particulières.

Ce logiciel, qui s'installe aussi bien sur Windows, MacOS ou Linux, permet de documenter des programmes écrits dans plusieurs langages tel que le C, le C++ (Le langage qui nous intéressera dans cette présentation) et le Java par exemple. La documentation proposée peut être fourni sous plusieurs langages tel que le HTML, LaTeX ou PDF. Doxygen permet aussi de générer des graphiques ou diagrammes, tel qu'un diagramme de classes par exemple si on lui demande.

Il peut s'exécuter en ligne de commande mais possède aussi une interface graphique comme on peut le voir dans la figure 1, où l'on doit renseigner la localisation de notre code ainsi que son nom, et où l'on peut paramétrer plusieurs choses tel que le choix du langage, le type de sortie, si l'on veut des graphiques ou non.

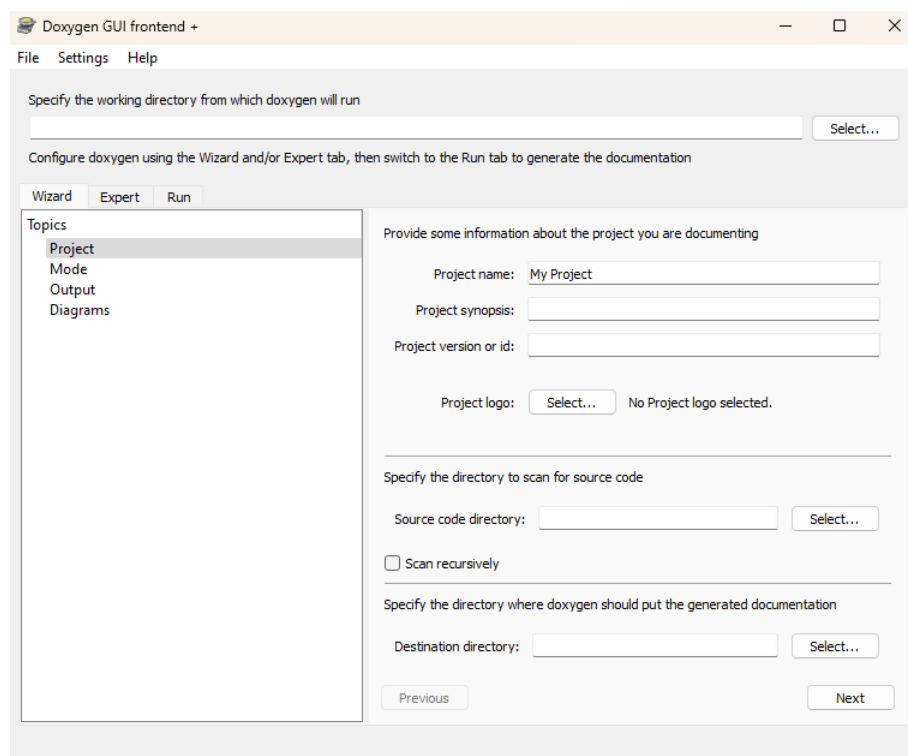


FIGURE 1 – Fenêtre logiciel Doxygen

3 L'outil dans notre projet

J'ai donc choisi d'intégrer cet outil dans notre projet car j'ai trouvé que c'était quelque chose d'important pour pouvoir présenter notre projet ainsi que sa construction.

3.1 Commenter le code

Pour commencer il a donc fallu commenter le code avec un style bien spécial comme nous pouvons le voir dans la figure 2. Dans notre cas, il y a 3 choses que nous avons commentées : les classes, leurs attributs, et leurs méthodes.

```
/**
 * @brief Classe qui représente la map.
 *
 * Cette classe représente la map. Celle ci ne sera instancié qu'une seule
 * fois durant le programme
 */
class Map
{
public :
    int nbpersonne; ///
```

FIGURE 2 – Exemple de commentaires

Pour les classes, il suffit de mettre un commentaire de cette forme avant `/**@brief */`, le `@brief` va correspondre à une courte description de la classe.

Pour les méthodes, les commentaires sont les mêmes mais avec d'autres paramètres en plus. Pour chaque paramètre de la méthode il faut rajouter `@param` avec le nom du paramètre et ce qu'il fait. Il y a aussi le `@return` qui sert à indiquer ce que retourne la méthode.

Enfin il ne reste que les attributs, et pour cela rien de plus simple, on rajoute `///<` après un attribut avec un commentaire pour l'expliquer.

3.2 Appliquer doxygen sur le code

Une fois que chaque classe est documentée on peut passer sur l'utilisation de doxygen en elle même, qui peut être utilisé soit en ligne de commande soit avec l'interface fenêtrée comme montrée précédemment. Ici nous ne parlerons que de la méthode avec l'interface car c'est la façon la plus simple.

3.2.1 Wizard

Vous allez donc pouvoir commencer par la partie Wizard. Dans cette partie il vous faudra choisir par exemple le nom de votre projet, et l'endroit où vous voulez que celui-ci s'enregistre.

Ensuite vous allez pouvoir choisir le langage et quels documents sélectionnés, puis quel type de sortie vous voulez (HTML ou LaTeX ou les deux) ainsi que la forme de celle-ci.

Enfin, vous pouvez choisir quel type de diagrammes vous souhaitez et la façon dont vous voulez les générer (avec GraphViz par exemple).

3.2.2 Expert

Cette partie permet de toucher un peu plus d'options du logiciel. Si vous ne maîtrisez pas totalement le logiciel vous n'avez pas forcément besoin d'y toucher mais cela peut permettre de personnaliser un peu plus sa documentation. De mon côté j'ai modifié les éléments de dot afin d'avoir des diagrammes de classes directement intégrés à notre documentation. Une partie de ma configuration est visible dans la figure 3.

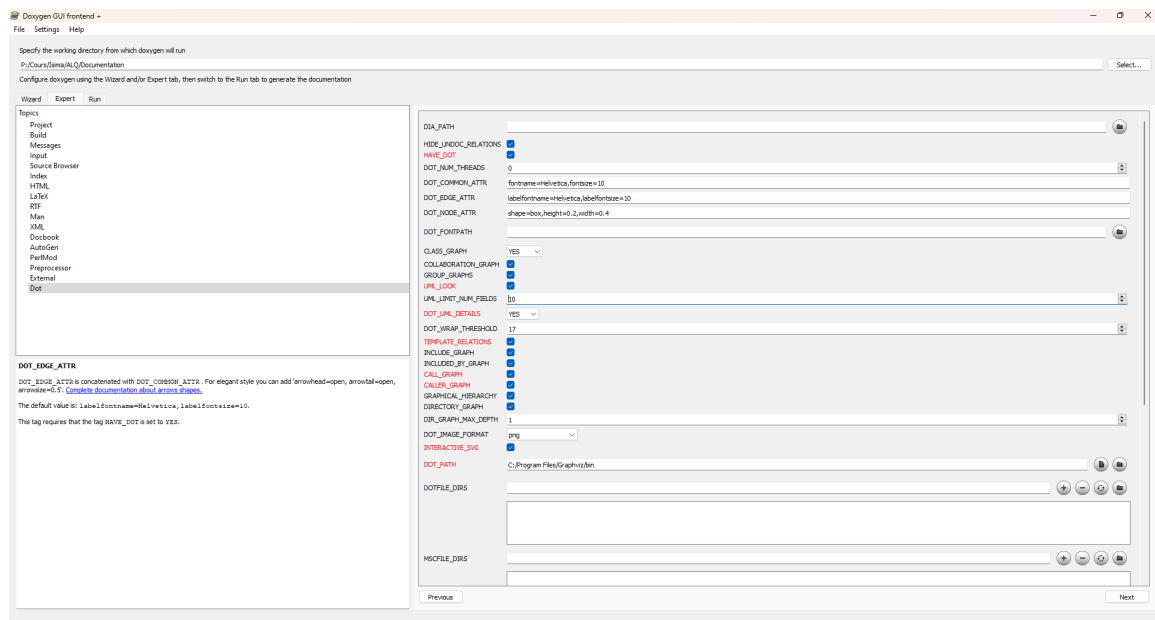


FIGURE 3 – Configuration de notre partie dot

3.2.3 Run

Enfin une fois que vous avez tout configuré, vous pouvez lancer le programme qui vous créera un dossier HTML ou LaTeX ou les deux suivant ce que vous avez demandé.

3.3 Le rendu de notre documentation

3.3.1 Présentation globale

Quand nous lançons la page d'accueil du site, nous tombons sur une page avec deux groupes d'onglets, ceux contenant la documentation des classes et celui contenant les fichiers. Le deuxième permet d'accéder aux fichiers de code, mais je ne vais pas présenter cette partie.

3.3.2 Onglets de classes

Dans les onglets de classes on peut retrouver une liste de toutes les classes, une index afin de retrouver les classes plus vite, une hiérarchie comme sur la figure 4 représentant la hierarchie globale de notre projet, et un dernier onglet repertoriant toutes les méthodes et les attributs des classes disponibles.

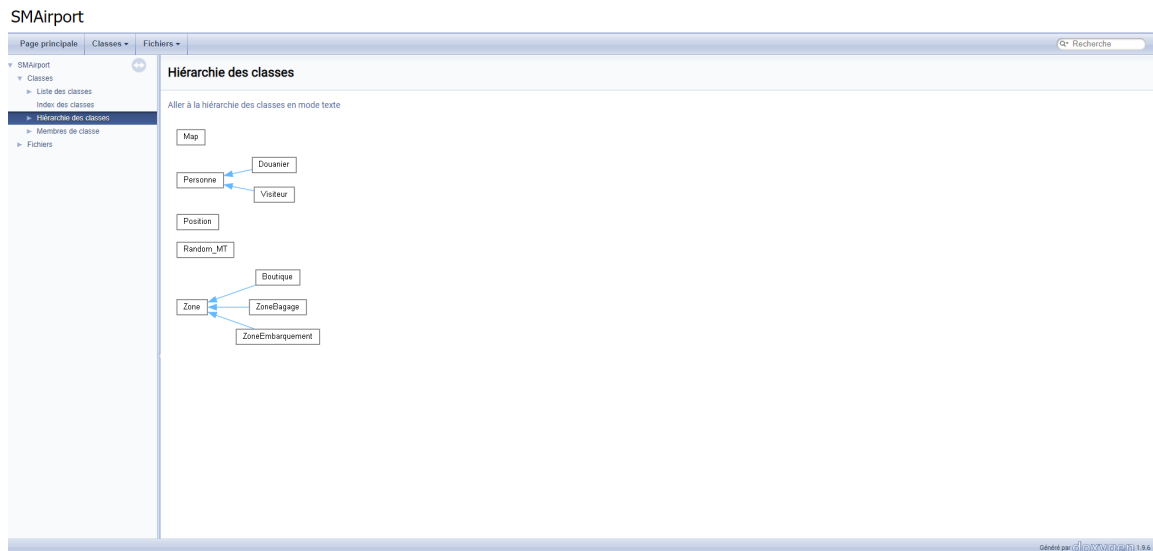


FIGURE 4 – Screenshot de notre hierarchie

3.3.3 Présentation d'une classe

Une classe est constituée d'un diagramme de classe, qui montre toutes les classes qui se rattachent à celle-ci, la liste de ses méthodes avec une petite description et la liste de ses attributs comme sur la figure 5. En descendant plus bas dans la page, on arrive sur une description détaillée de la classe ainsi qu'une description détaillée de chaque méthode comme montrée dans la figure 6.

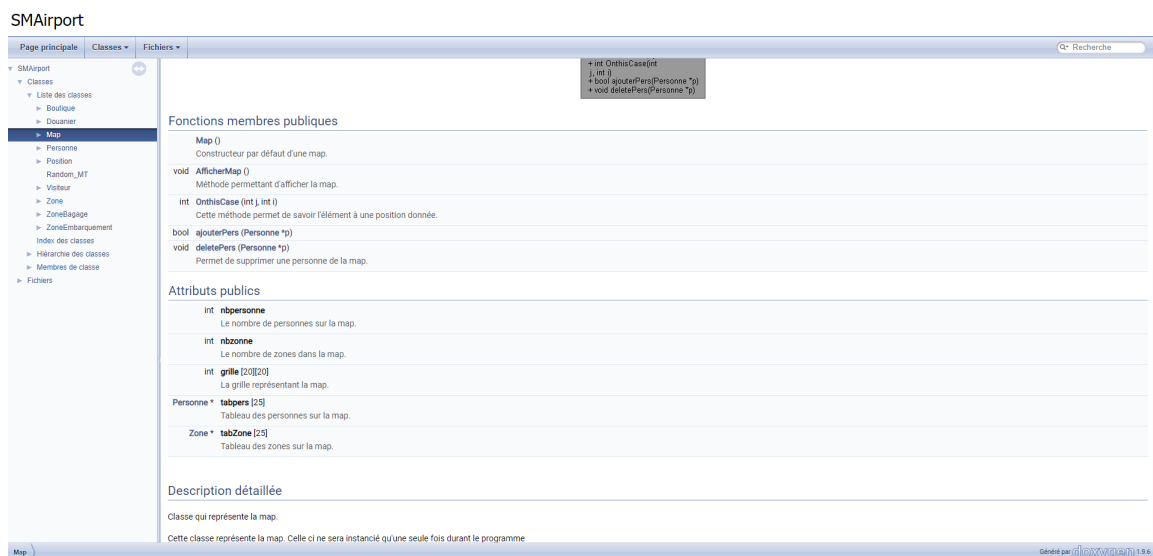


FIGURE 5 – Screenshot des méthodes de la classe map

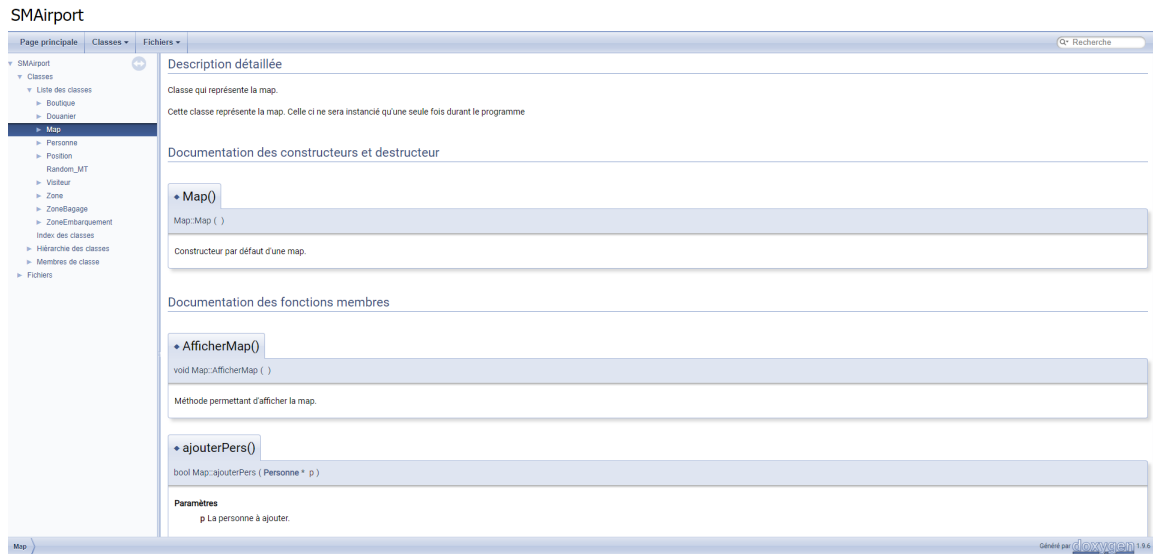


FIGURE 6 – Screenshot de la classe map

On peut donc voir le contenu de la classe map après les commentaires mis en 2. Cela nous permet de documenter très facilement et d'avoir une belle documentation simple à comprendre et détaillée.

4 Conclusion

Nous pouvons donc voir que ce logiciel permet grâce à quelques commentaires et l'application doxygen, de créer rapidement une documentation HTML très simple à comprendre et très détaillée si les commentaires le sont.

Je trouve que ce logiciel est très utile notamment pour finir un projet par exemple afin de le rendre avec une documentation permettant de mieux le comprendre.

La documentation est disponible dans le dossier Documentation du Gitlab.