

Faire une documentation avec Doxygen

Lorsque l'on programme que ce soit dans le cadre d'un TP, d'un projet, d'une SAÉ, d'un stage, ou dans notre travail plus tard on se doit de documenter notre code pour que celui-ci puisse être facilement relu (par nous même dans quelque temps, ou par quelqu'un d'autre) et donc maintenable. Pour documenter un programme en C on peut utiliser Doxygen.

1 Balises de documentation

Pour documenter notre code nous plaçons dans nos fichiers .c et .h des sortes de commentaires écrits d'une façon particulière(entre `/**` et `*/`) pour qu'ensuite grâce à doxygen une documentation html soit générée.

Au début de chaque fichier nous placerons systématiquement une documentation de fichier indiquant le nom du fichier, le ou les auteurs du fichier, la date de création du fichier et le contenu du fichier. Pour réaliser cette documentation de fichier les balises `\file`, `\author`, `\date` `\brief` sont utilisées.

Exemple :

```
/**
 * \file nomFichier.c
 * \brief description courte du fichier
 * \author André Dupond
 * \date 18 mai 2021
 *
 * on peut également écrire ici plusieurs lignes d'une description plus
 * longue pour expliquer un peu plus longuement le contenu du fichier
 * si on juge que cela peut être utile.
 */
```

Puis, pour chaque fonction on va faire juste avant la définition de la fonction une documentation de cette fonction afin que toute personne qui voit juste le prototype de cette fonction et sa documentation soit capable de l'utiliser dans un autre programme. La documentation doit donc permettre d'indiquer les paramètres de la fonction, la valeur retournée et une description brève ou pas de la fonction. Pour ce faire les balises `\param` et `\return` (et `\brief` si on veut) seront utilisées.

Exemples :

```
/**
 * \brief affiche la moyenne du candidat donné
 * \param num numéro du candidat
 * \param moy moyenne du candidat
 */
void affichCandidat(int num, float moy){
    ....
}
```

```
/**
 * \brief calcule le maximum entre deux nombres
 * \param nb1 un premier nombre
 * \param nb2 un deuxième nombre
 * \return la valeur du maximum de ces deux nombres
 */
float maximum(float nb1, float nb2){
    ....
}
```

Concernant la balise `\param`, il est possible d'indiquer si le paramètre est entrant (in), sortant (out) ou entrant/sortant (in, out).

```
/**
 * Charge dans un tableau les nombres lus dans le fichiers
 * \b nombres.txt et en fait la somme
 * \param[in,out] tabNombres tableau des nombres que l'on charge
 * \param[out] somme pointeur sur la somme des nombres lus
 * \param[in] tmax taille physique du tableau
 * \return le nombre de nombres chargés dans le tableau
 */
int fchargeEtSom(float tabNombres[], float *somme, int tmax){
    ....
}
```

Quand plus tard dans le cours on définira des structures ou énumérations, on pourra également les documenter en utilisant les balises `\struct` et `\enum`.

2 Configuration

Il faut générer un fichier nommé Doxyfile. Ceci peut se faire de 2 manières :

1. on lance doxywizard (en tapant la commande doxywizard)
 - Étape 1 : on sélectionne le répertoire de travail.
 - Étape 2 : on remplit les 4 parties (Project, Mode, Output, Diagrams) de l'onglet Wizard.
 - pour la partie Project :
 - on donne un nom au projet (par exemple : TP1exo1, TP2, ...)
 - on sélectionne le répertoire contenant le code source
 - pour la partie Mode
 - on sélectionne le mode d'extraction désiré (seulement les entités documentées)
 - on sélectionne le langage de notre source (C)
 - pour la partie Output
 - on ne sélectionne que html et pas latex
 - pour la partie Diagrams
 - on laisse comme c'est
 - puis on sauve (File, Save ou ctrl+S). On génère alors un fichier nommé Doxyfile dans le répertoire courant.
2. ou alors on génère un fichier Doxyfile en tapant dans le terminal : `doxygen -g`
puis on ouvre le fichier Doxyfile généré et on modifie quelques valeurs :
 - `PROJECT_NAME` = "le nom du TP par exemple entre guillemets"
 - `OPTIMIZE_OUTPUT_FOR_C` = YES
 - `INPUT` = le chemin du répertoire contenant les fichiers source
par exemple : `INPUT = /home/casimon/TPalgo/TP1` ou on laisse vide, si répertoire courant : `INPUT =`
 - `GENERATE_LATEX` = NO

3 Exécution

Pour générer la documentation, il est nécessaire d'exécuter doxygen. On peut le faire directement en tapant dans un terminal : `doxygen`
ou alors si doxywizard est ouvert on peut utiliser l'onglet Run et cliquer sur "Run doxygen" (à condition bien sûr d'avoir à l'instant configuré doxywizard ou ouvert le fichier Doxyfile avec doxywizard).