TP Générateur de documentation de code
Module : Développement logiciel 2 suite : Outils de suivi des modifications

1. Sommaire

2. Objectifs

- Apprendre à commenter le code source de façon structurée et systématique
- Apprendre à utiliser un générateur de documentation de code (Doxygen)

3. Introduction

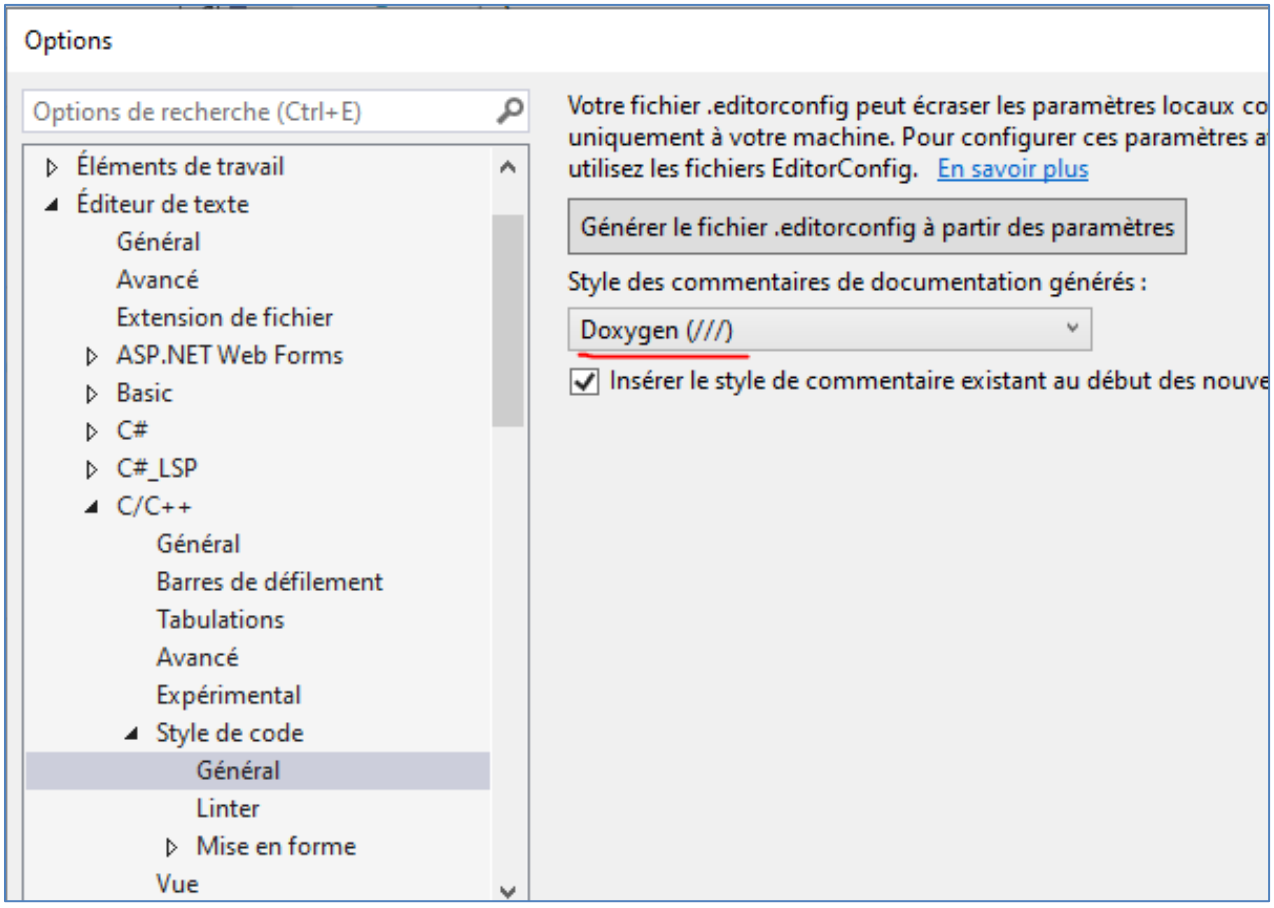
Dans ce TP, vous allez appliquer la procédure de création de la documentation sur la solution « carre » du « TP POO carres animes ».

« TP*NomDuTP* » *sera à remplacer par « carre »*

Cette solution est sur votre GitHub dans un dépôt « carre », vous ajouterez les commits supplémentaires pour le documentation Doxygen.

4. Configuration de Visual Studio pour les commentaires Doxygen

⇨ Dans le menu « Outils » / « Options », sélectionner le style de commentaire Doxygen (///) :



5. Formalisation des commentaires pour Doxygen et recommandations.

Doxygen est conçu pour documenter les classes et les fonctions dans les fichiers d'entête.

Les commentaires seront reconnus par Doxygen s'ils commencent pas /// suivis par un tag (balise).

Visual Studio va vous aider à commenter du code déjà écrit. Si vous tapez un triple slash (///) sur la ligne au-dessus d'une variable, de l'entête d'une fonction, d'une classe, Visual Studio va automatiquement compléter avec des tags (balises) adaptés.

5.1.Les éléments à commenter

⇨ Les fichiers sources d'entête (.h)

En première ligne ajouter :

```
/// @file NomDuFichier.h
/// @brief Programme principal du TP…
/// @details Ce programme utilise …
/// @author Serge Delbosc
/// @version 0.1 - Visual Studio 2019
/// @date 09/09/2024
```

⇨ Les attributs

Exemple :

```
/// @brief Variable de test
int maVariable = 10;
```

⇨ Les méthodes

Spécifier si le paramètre est une entrée[in], une sortie[out] ou les deux[in/out].

Exemple :

```
/// @brief Fonction de division
/// @param[in] pDivid Dividende
/// @param[in] pDivis Diviseur
/// @param[out] pQuot Quotient
/// @return False si division pas 0
bool Division( int*pDivid, int* pDivis, int*pQuot)
{
	bool divZero = true;
	if (pDivis)
		*pQuot = *pDivid / *pDivis;
	else divZero = false;
	return divZero;
}
```

⇨ Les classes

Les commentaires seront placés dans le fichier d'entête au-dessus de l'entête de la classe et aussi pour chaque attributs et méthodes.

Exemple :

```
/// @brief Classe de base abstraite
class CBase
{
public:
	/// @brief Un attribut de la classe de CBase
	int attrubutB;
	/// @brief Méthode virtuelle pure de la classe CBase
	/// @return Une valeur de retour
	virtual int MethodeBVirtuel() = 0;
	void MethodeB();
};
```

5.2.Autres tags

@example, @note, @pre, @post, @bug, @warning, @attention, @remark, @copyright, …
Voir explication : https://www.doxygen.nl/manual/commands.html

5.3.Créer une page principale de la documentation

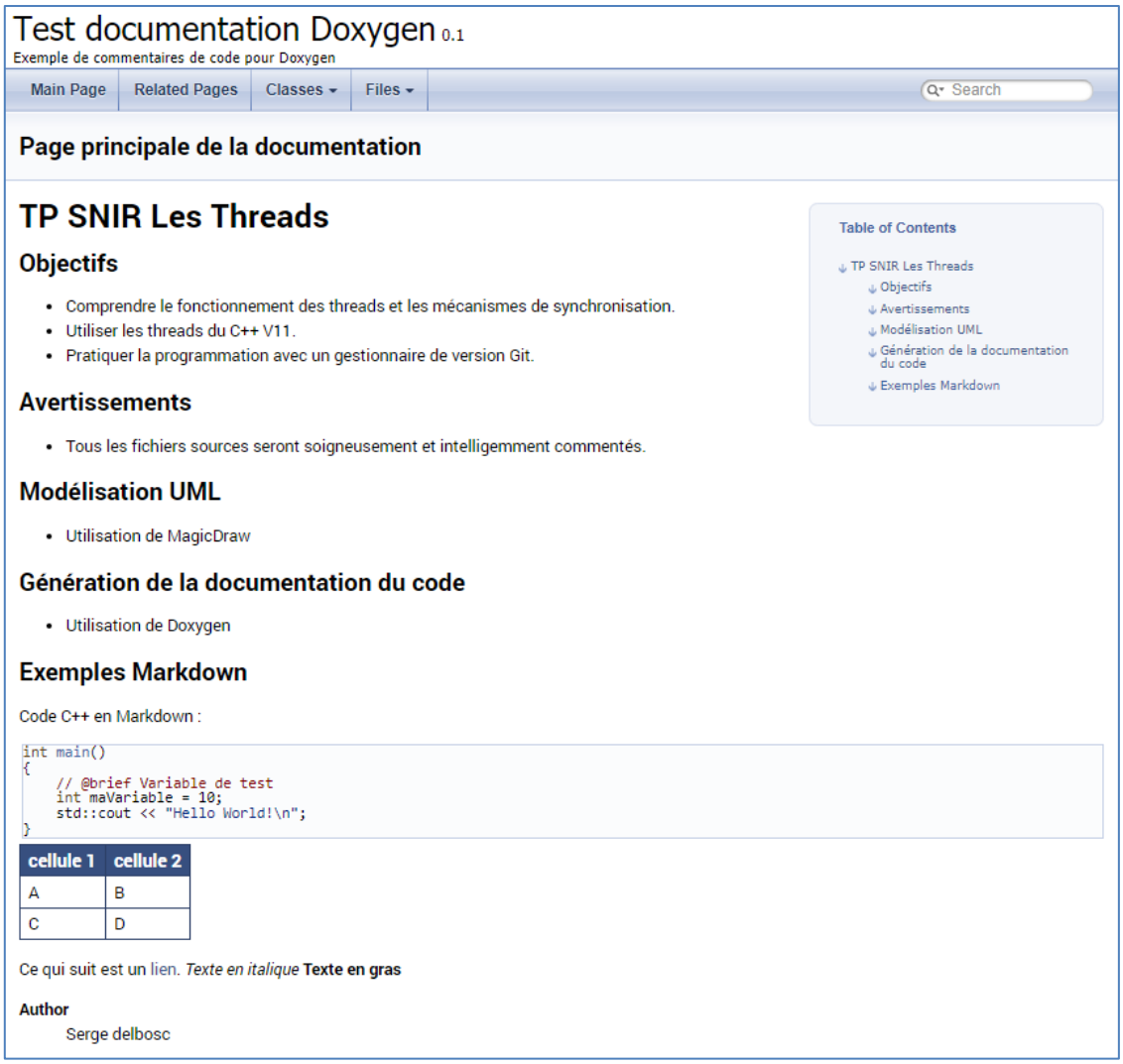
➲ Ajouter un fichier dans la solution de type fichier texte et mettre l'extension .md afin de créer une documentation en Markdown. Commencer la page avec un `\mainpage` suivi du titre de la page principale.

Exemple avec une table des matières [TOC]

````
\mainpage Page principale de la documentation
[TOC]
# TP SNIR Les Threads
## Objectifs
- Comprendre le fonctionnement des threads et les mécanismes de synchronisation.
- Utiliser les threads du C++ V11.
- Pratiquer la programmation avec un gestionnaire de version Git.
## Avertissements
- Tous les fichiers sources seront soigneusement et intelligemment commentés.
## Modélisation UML
- Utilisation de MagicDraw
## Génération de la documentation du code
- Utilisation de Doxygen
## Exemples Markdown
Code C++ en Markdown :
```cpp
int main()
{
    // @brief Variable de test
    int maVariable = 10;
    std::cout << "Hello World!\n";
}
```
|cellule 1|cellule 2|
|--------|--------|
|    A    |    B    |
|    C    |    D    |

Ce qui suit est un [lien](https://example.com/ "titre de lien optionnel").
*Texte en italique*
**Texte en gras**

\author Serge delbosc
\date 2021
\version 0.1Version générée en HTML :
````



6. Génération de la documentation Doxygen

⇨ Les programmes Doxygen sont présents dans 'C:\Program Files\doxygen **SI** ce n'est pas le cas, faites l'installation avec les paramètres par défaut.

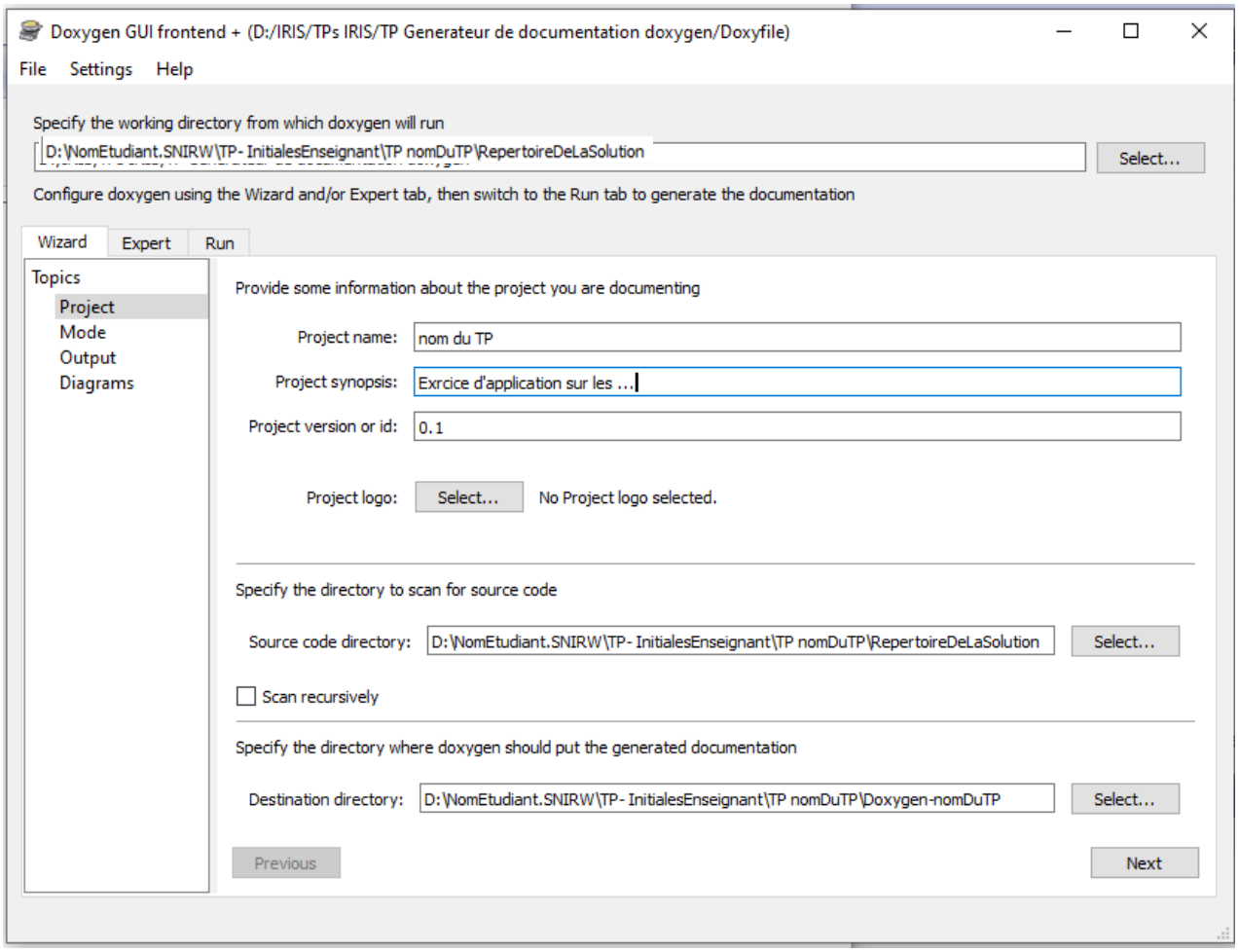
➲ Créer un dossier pour la documentation « Doxygen-TP*NomDuTP* en dehors du répertoire de la solution. Ce Dossier n'est pas dans le dossier de travail de git.

➲ Lancer C:\Program Files\doxygen\bin**doxywizard.exe**

6.1.Configuration du Projet :

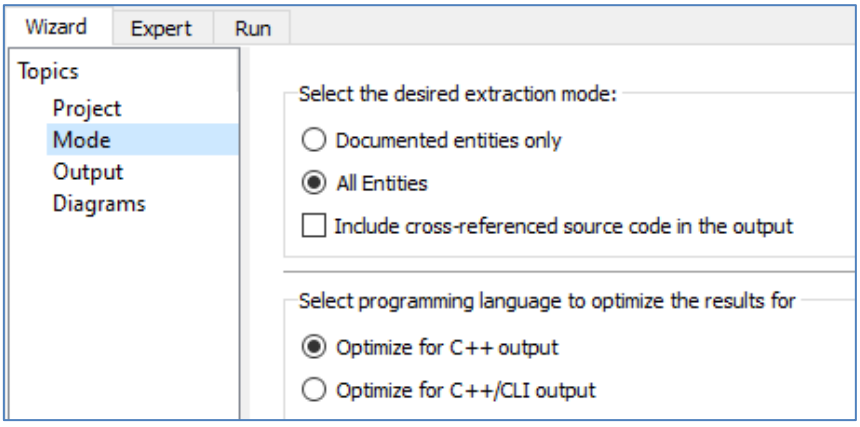
➲ Remplacer « NomEtudiant » « InitialesEnseignant » « nomDuTP » « RepertoireDeLaSolution »
Dans le « RepertoireDeLaSolution », vérifier qu'il y le « .git »

➲ Dans la partie « Provide some information… » Préciser les informations qui figureront en titre sur la documentation.



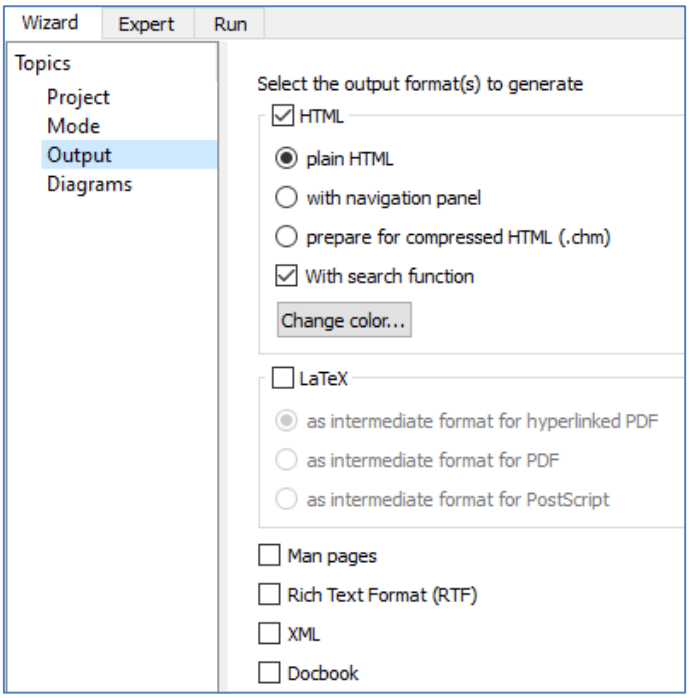
6.2.Configuration Mode

⇨ Cocher « All entities »

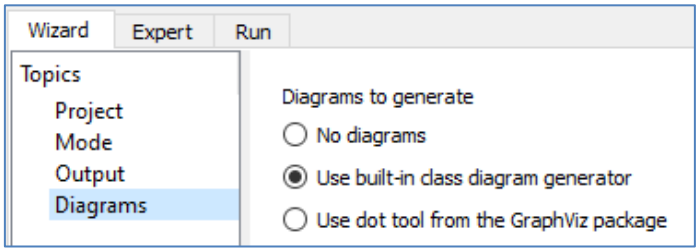


6.3.Configuration Output

⇨ Décocher « latex »



6.4.Configuration Diagrams



6.5.Génération de la documentation

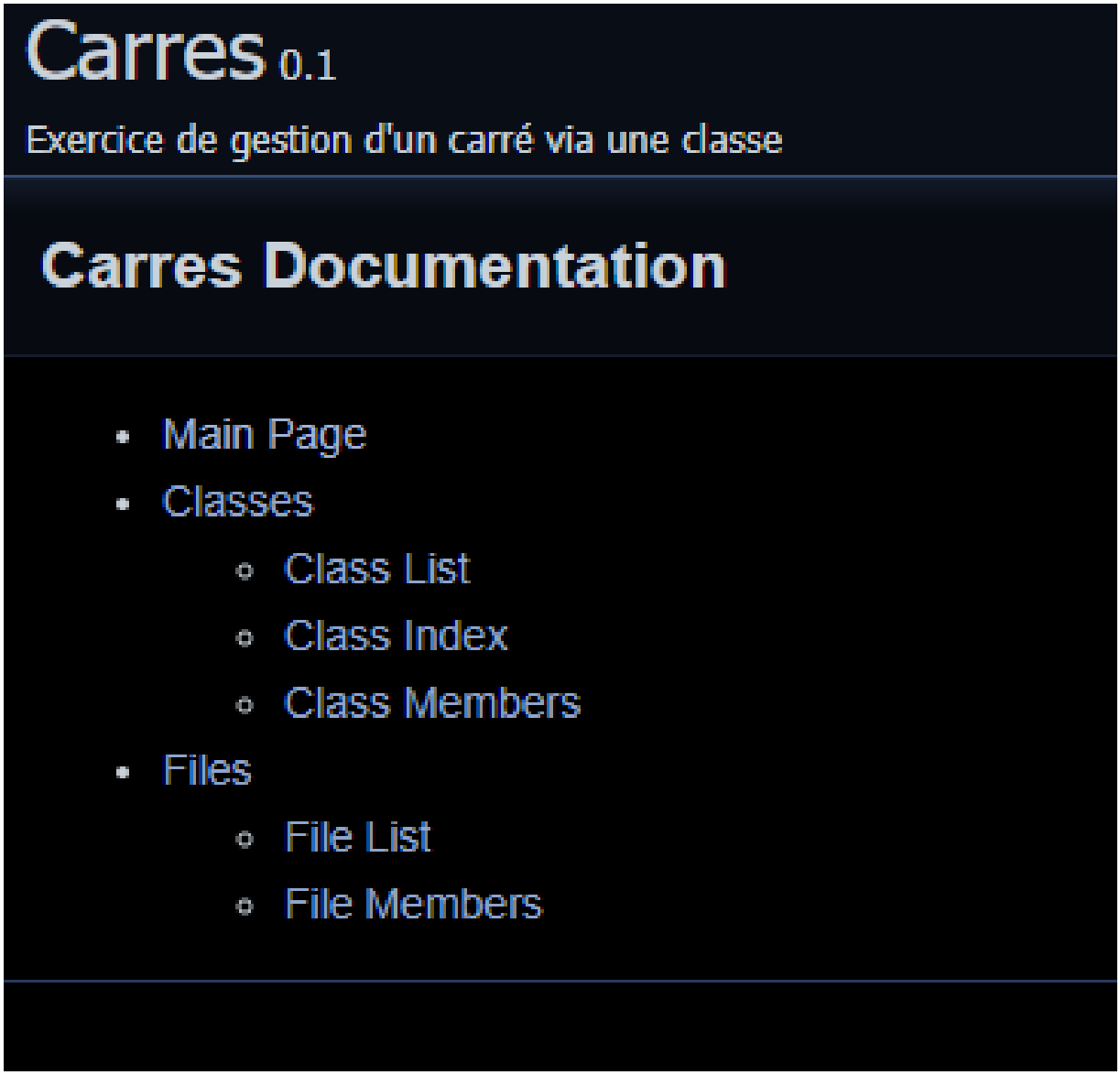
➲ Dans l'onglet Run, cliquer sur « Run doxygen »

6.6.Visualisation de la documentation

➲ Dans l'onglet Run, cliquer sur « Show html output »

➲ Ou bien, ouvrer dans un navigateur D:\NomEtudiant.SNIRW\TP- InitialesEnseignant\TP nomDuTP\Doxygen-nomDuTP \html**\index.html**

➲ Faire une capture d'écran ci-dessous montrant la documentation de la classe :


Carres 0.1
Exercice de gestion d'un carré via une classe
Carres Documentation
Main Page
Classes
Class List
Class Index
Class Members
Files
File List
File Members

6.7.Sauvegarde de la configuration de Doxygen

➲ Sauver dans le répertoire de la solution la configuration de Doxygen «File» «Save as» :
D:\NomEtudiant.SNIRW\TP- InitialesEnseignant\TP nomDuTP\RepertoireDeLaSolution
Le fichier Doxyfile est créé.
Ainsi le fichier de configuration Doxygen est compris dans les fichiers à sauvegarder dans git mais pas la documentation générée.
Bien indiquer le numéro de commit :

Commit #10 : Doxygen

Autoévaluation

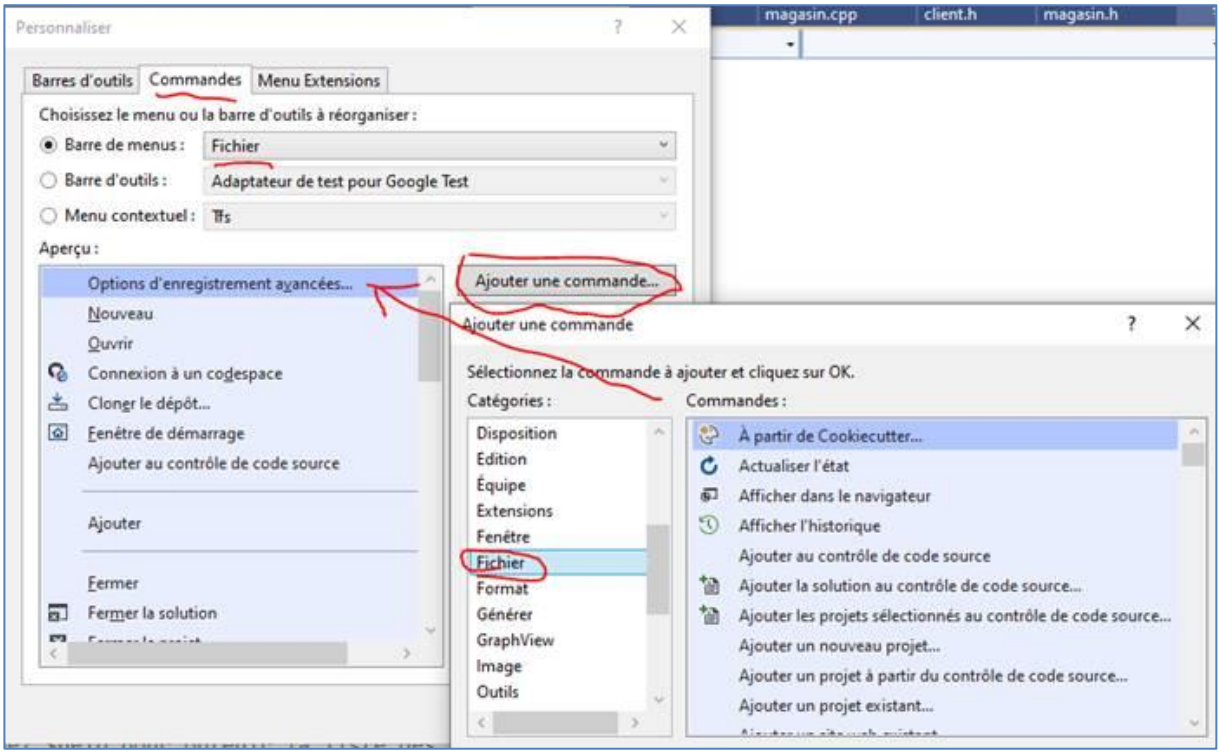
7. Configuration pour l'encodage des caractères accentués

Deux possibilités, soit vous modifier l'encodage des fichiers sources dans VS pour qu'il soit compatible avec les option par défaut de Doxygen, soit vous modifiez l'encodage de Doxygen.
La deuxième solution est préférable.

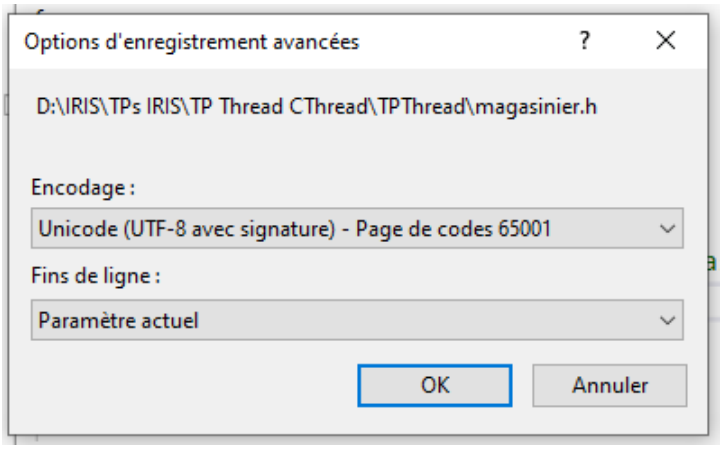
1er Solution : Modification de l'encodage sous VS

Si vous utilisez des commentaires en français avec les accents, il faut enregistrer les sources sous Visual Studio avec l'encodage utf8.

⇨ Pour obtenir cette « Option d'enregistrement » dans le menu Fichier il faut aller dans Outils/Personnalisé/ puis :



⇨ Puis pour chaque fichier source (.cpp, et .h) modifier « Fichier » « Options d'enregistrement avancées » et choisir UTF-8 (idem pour les fichiers Markdown)



2ème Solution : Modification de l'encodage sous Doxygen

⇨ Dans l'onglet « expert » puis topics, puis :
Dans « projet » modifier DOXYFILE_ENCODING avec la valeur ISO-8859-1
Dans « input » modifier INPUT_ENCODING » avec la valeur ISO-8859-1

8. Références

Documentation Doxygen : https://www.doxygen.nl/index.html

9. Notes personnelles

Cadre dans lequel vous noterez les points importants que vous devez retenir du TP et aussi tout ce que vous avez appris en informatique durant le TP.