Feuille de TP n°2 :

Interface WIMP
Découverte d'un outil de génération d'interfaces graphiques associé à CodeBlocks

OBJECTIF PEDAGOGIQUE

1.- Découvrir une interface WIMP par l'analyse de ses composants.

2.- Découvrir le générateur d'interfaces graphiques wxSmith associé à CodeBlocks pour produire un premier programme C++ à interface graphique WIMP.

TRAVAUX A RENDRE

- Document de synthèse à déposer sur le WebCampus au plus tard le jeudi 31/03/2016.

RESSOURCES A VOTRE DISPOSITION POUR REALISER LE TP

~ Cette feuille de TP n°2.

DIRECTIVES PARTICULIERES A CETTE SEANCE

Dans votre répertoire de travail habituel du dossier `M2105-IHMs` situé sur votre lecteur réseau (\\iguzki, lecteur Z) :

- créez un répertoire nommé **`tp2`**,
- copiez dans **`tp2`** tous les fichiers qui ont été mis à votre disposition

---

A.- Découverte d'une interface WIMP par l'analyse de ses composants
Découverte du Générateur d'interfaces graphiques wxSmith associé à CodeBlocks et création d'un premier programme C++ à interface graphique.

1.- Programme zero.exe : premier programme C++ à interface graphique

(a) Dans votre répertoire `tp2` (lecteur Z sur \\iguzki), **créez** un projet CodeBlock intitulé **zero** avec les options ci-dessous :

- Utiliser le template wxWidgets (Figure 1)
- Parmi les versions de wxWidgets proposées, sélectionner la version 2.8.x (Figure 2)
- Créer le projet dans votre répertoire tp2, et nommez-le zero (Figure 3)
- Choisir le générateur d'interface GUI graphiques wxSmith, Application de type FRAMED (Figure 4)
- Préciser si nécessaire le répertoire d'installation de la bibliothèque wxWidgets (Figure 5)
- Utiliser le compilateur par défaut Gnu-gcc, sélectionner le mode **DEBUG uniquement** (Figure 6)
- Production de bibliothèques **STATIQUES**, activer le codage des caractères en mode **UNICODE** (meilleure portabilités des applications), et cocher la demande d'options de configuration avancées (Figure 7)
- Application sera en mode **GUI** (et non en mode CONSOLE) et
- Associer les bibliothèques **_WXDEBUG_** et **Debug wxWidgets** au compilateur Gnu-gcc (**Figure 8**) **Attention**, cette option est décochée par défaut…..
- Projet `zero` créé, ouverture de la fenêtre de l'application créé avec l'éditeur d'interfaces graphiques de wxSmith intégré dans CodeBlocs (Figure 9)

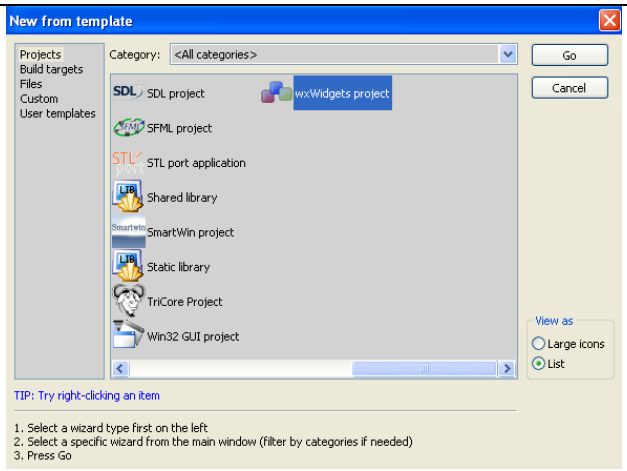

Figure 1 : Création d'un projet avec wxWidgets

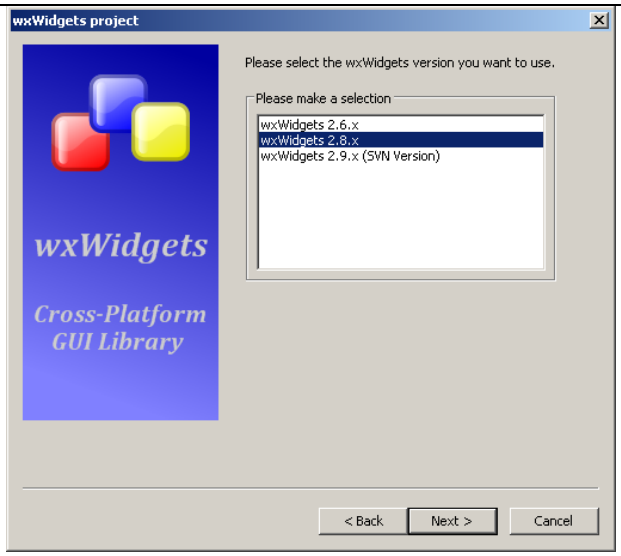

Figure 2 : Choix de le version de wxWidgets

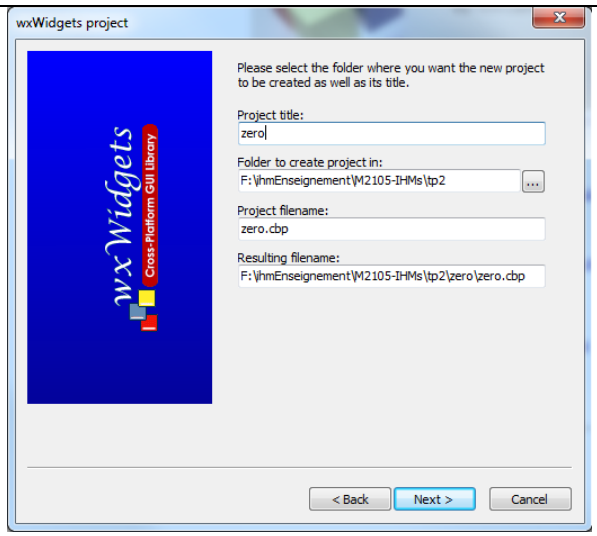

Figure 3 : Projet zero, dans répertoire tp2

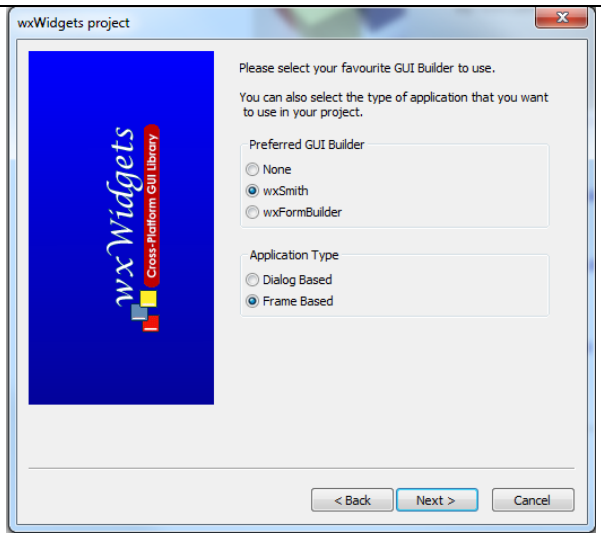

Figure 4 : Utilisation du générateur d'interfaces graphiques wxSmith

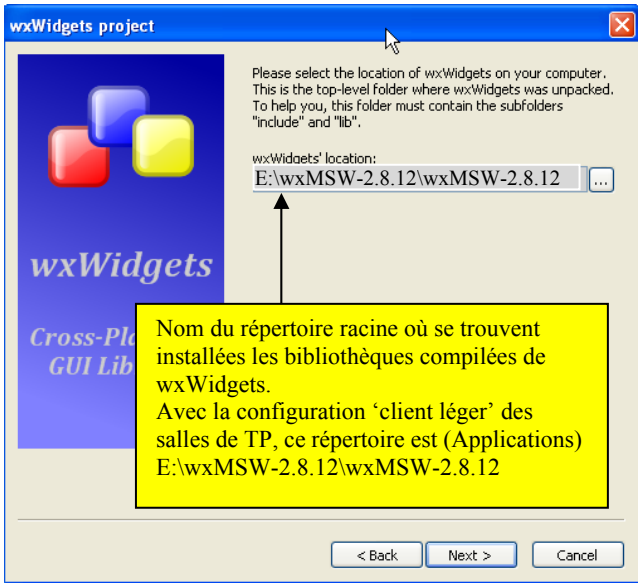

Figure 5 : Répertoire d'installation de la bibliothèque wxWidgets

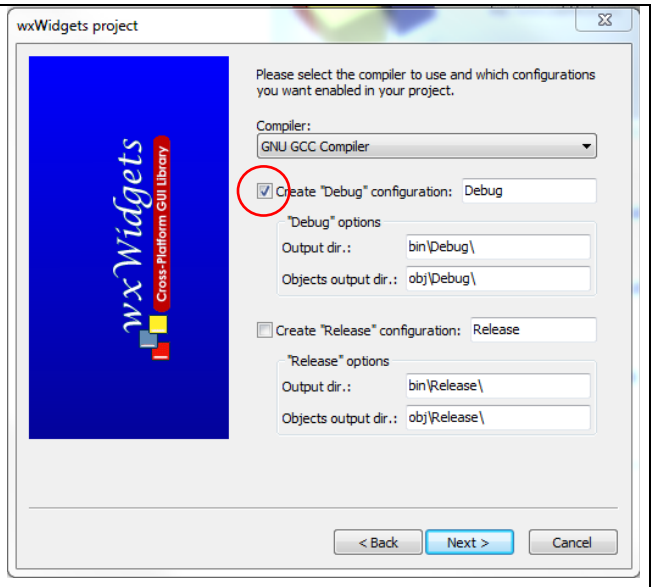

Figure 6 : Choix du compilateur et de la configuration DEBUG

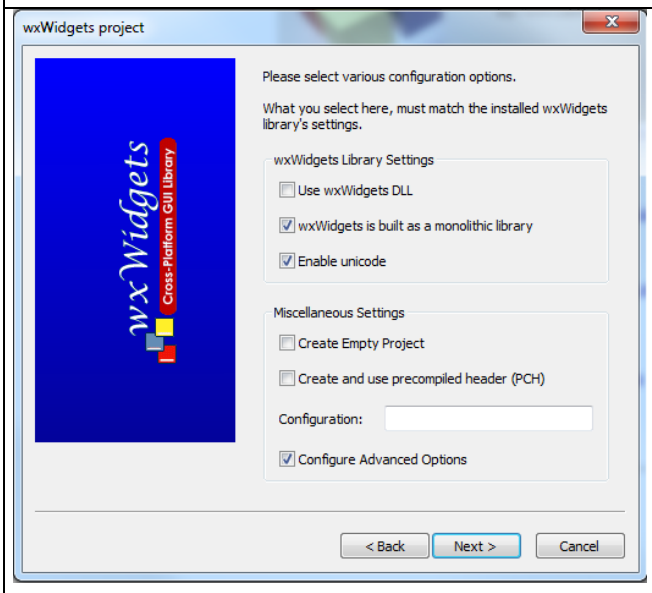

Figure 7 : Options de configuration (1 bibliothèque STATIQUE, Unicode, ...)

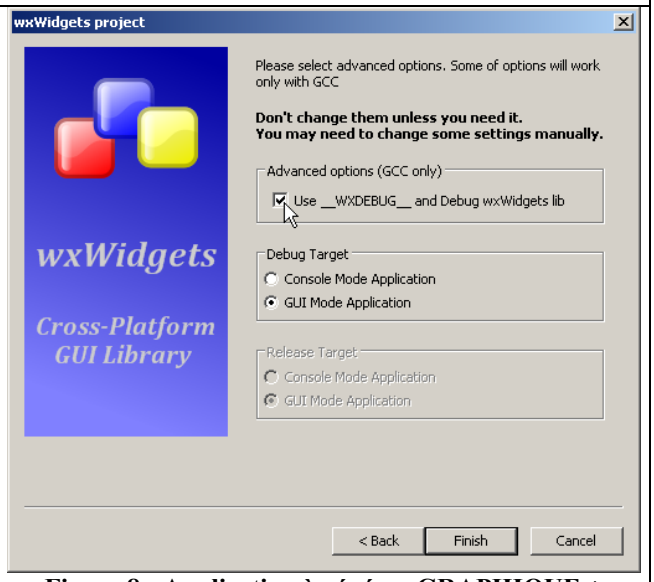

Figure 8 : Application à générer GRAPHIQUE + utilisation de bibliothèques supplémentaires !

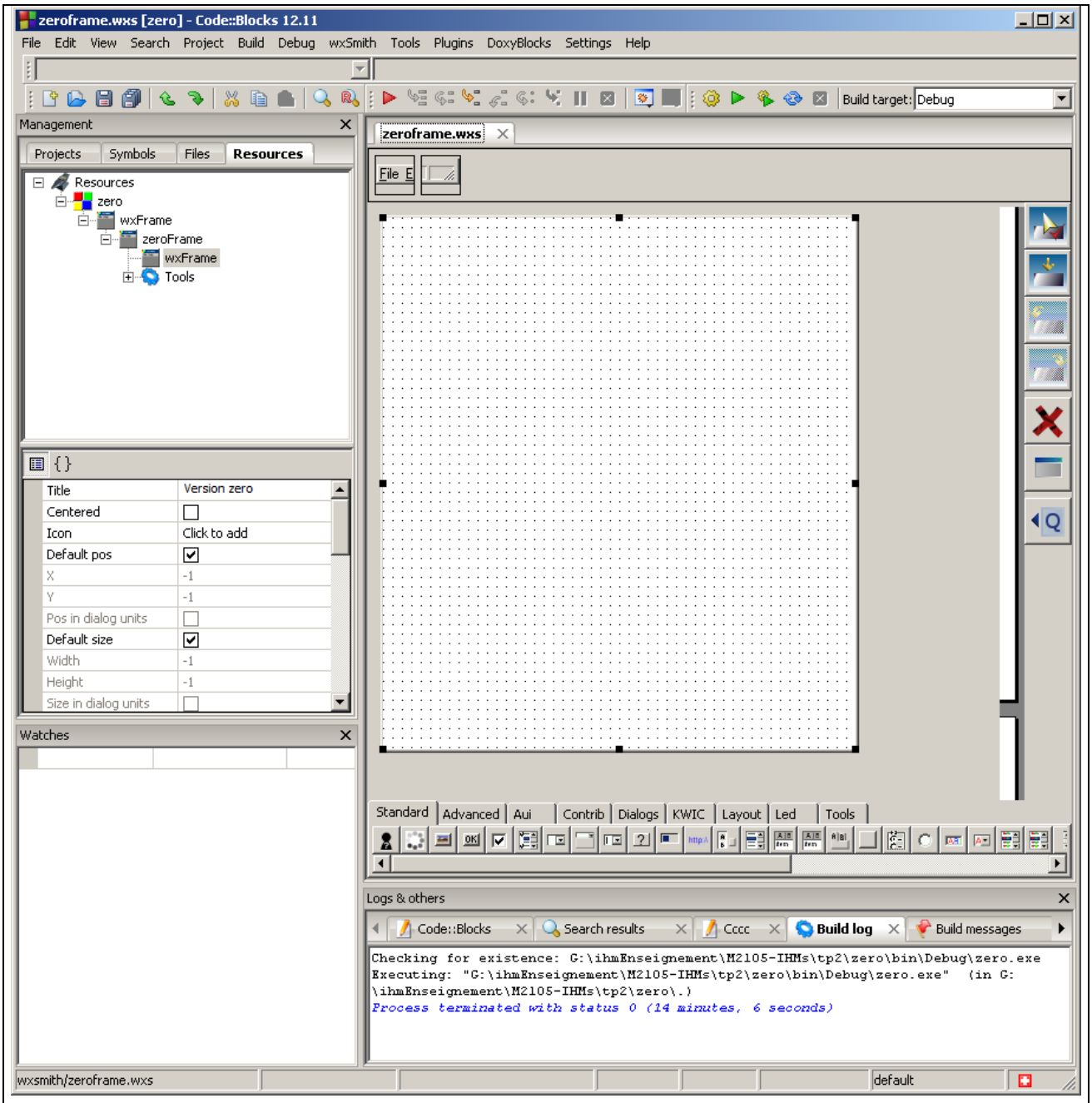

Figure 9 : Fenêtre de l'application zero vue avec wxSmith

(b) Compiler et exécuter le programme généré

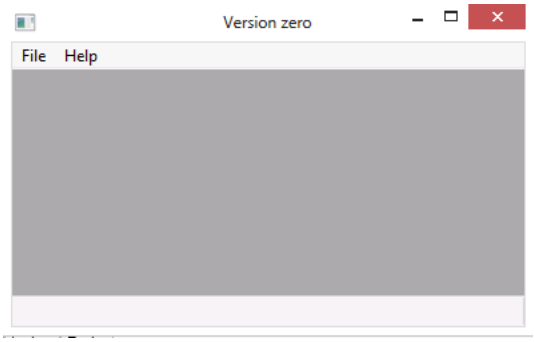

Figure 10 : Programme en cours d'exécution

(c) Analyser le programme en cours d'exécution

- Lister les composants graphiques qu'il contient
- Décrire son comportement (ce qu'il peut 'faire') par défaut
- Quel est le titre de la fenêtre ?

(d) Analyser le contenu du répertoire zero

- Comment s'appelle le programme exécutable ? (le .exe) Où se trouve-t-il ?
- Qu'est le fichier zeroframe.wxs ? Où se trouve-t-il ?
- Qu'est le fichier resource.rc ?
- Lister les fichiers C++ présents dans ce répertoire. Avez-vous une première idée de leur rôle ? (ce sera expliqué plus tard)

(e) Découverte de l'interface de l'éditeur d'interfaces graphiques wxSmith

Dans l'onglet Resources

Développer le menu **Tools**

Nommer et expliquer le rôle de chacun des éléments numérotés (parfois avec doublons) de la **Figure 11** et de la **Figure 12**.

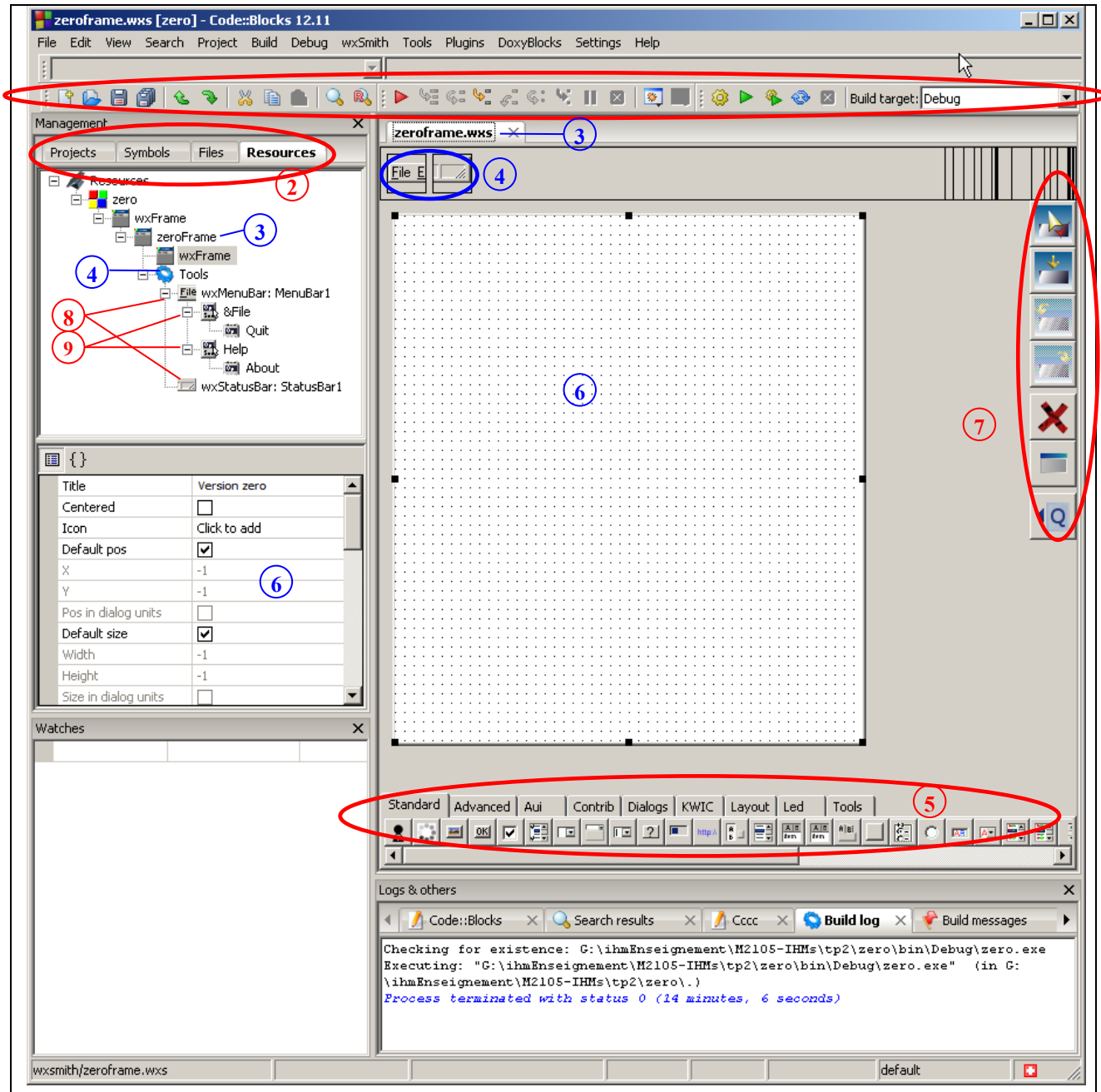

Figure 11 : Nommer et expliquer le rôle des éléments numérotés

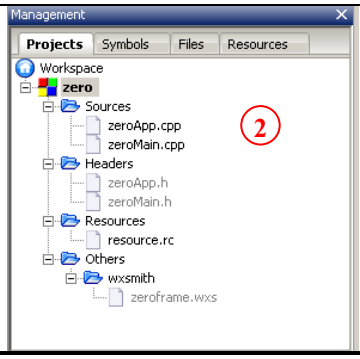

Figure 12 : Onglet Projects de le fenêtre Manager

2.- Programme addition.exe

But du programme (final) :

Calculer et afficher la somme de deux entiers préalablement saisis par l'utilisateur au moyen de deux zones de saisie fournies. La somme est calculée suite au clic sur le bouton **Addition !**

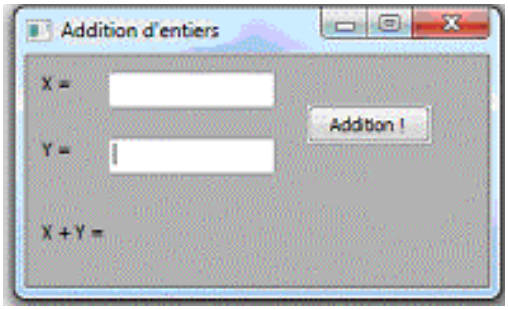

Figure 13 : additionneur simplifié

Dans cette version, le programme ne réalise encore aucune action : pas de calcul, pas de contrôle de cohérence….
L'exercice demandé consiste donc à créer l'interface du programme au moyen de l'éditeur d'interfaces wxSmith.

(a) Dans votre répertoire `tp2`, **créez** un projet CodeBlocks intitulé **addition** avec les mêmes options que pour le programme zero.exe.

(b) Placer les composants nécessaires à ce programme.
- Justifier le choix du type de chaque composant
- Placer les éléments visibles aux coordonnées (colonne, ligne), exprimées en pixels, suivantes :

Hauteur fenêtre: 150 pixels

Largeur fenêtre : 300 pixels

10, 10 50, 10

10, 50 50, 50

10, 100 180, 30

Figure 14 : additionneur simplifié

- Compléter par les éléments manquants.

(c) Nommage des éléments
- Quelle stratégie avez-vous choisie pour nommer les éléments du programme ?
- Comparer votre stratégie à celle de votre voisin de TP. Est-ce la même ? Décrire en quelques mots différentes stratégies possibles.

3.- Synthèse

Complétez votre document de synthèse du TP avec les réponses (correctement rédigées) aux questions suivantes :

- dans le point B.1- (B.1.c B.1.d et B.1.e)
- dans le point B.2- (B.2.b et B.2.c)