# Étapes pour insérer un nouveau panneau.

Menu wxSmith, Ajouter wxDialog

Remplir le nom de la classe : par exemple ZoomSpecifique

Fichier d’entête et source seront créés dans le sous répertoire src avec le même nom

Accepter l’intégration dans les sous-projets (debug, release, …)

Fichier .wxs affiché en mode wxSmith

Lui donner un titre puis …

Entrer les différents éléments graphiques, zones de texte, cases à cocher, …

Il est préférable de les entrer dans l’ordre, du haut vers les bas, pour faciliter la lecture directe du fichier wxs (en fait un format xml), mais cela n’a pas d’importance.

Créer dans wxFrame, Outils le sous menu Zoom Spécifique, donner un nom à la variable, par exemple ici : Menu\_ZoomSpecifique

Ajouter dans { } un EVT\_MENU, ajouter un gestionnaire et accepter la proposition de nom (ou la changer). Note, bug C::B car il faut le faire 2 fois pour que ça marche (mais la seconde fois le nom change un peu, alors refuser le nom et choisir le bon ) !

Dans OvniMain.cpp est créé :

void OvniFrame::OnMenu\_ZoomSpecifiqueSelected(wxCommandEvent& event)

Ne pas hésiter à enregistrer les wxs modifiés et/ou .cpp, .h car, en tous cas pour les wxs, l’enregistrement ne semble pas toujours fait automatiquement.

Il faut remplir cette routine par au moins un show() afin d’afficher la boite de dialogue.

Ici, par exemple on va mettre : ZoomSpecifique\_Panel->Show();

Dans OvniMain.h, ajouter en haut un #include ”ZoomSpecifique.h” et ajouter un peu plus bas :

ZoomSpecifique\* ZoomSpecifique\_Panel;

Avec les autres déclarations du même type.

Dans ZoomSpecifique.h, ajouter #include ”OvniMain.h” après les headers introduits par wxSmith et avant la déclaration de classe.

En fin de ZoomSpecifique.h, ajouter OvniFrame\* MAIN; juste avant DECLARE\_EVENT\_TABLE, dans la zone private.

Dans ZoomSpecifique.cpp, ajouter à la fin de ZoomSpecifique::ZoomSpecifique :

this->MAIN = dynamic\_cast<OvniFrame\*>(parent);

Dans OvniMain.cpp créer la classe ZoomSpecifique\_Panel dans OvniFrame::OvniFrame, là où il y a déjà les autres déclarations :

ZoomSpecifique\_Panel = new ZoomSpecifique(this,wxID\_ANY);

Dans interface.h, ajouter en début (avec les autres déclarations du même type) :

class ZoomSpecifique;

Créer dans ZoomSpecifique.wxs un évênement OnClose dans { }. Ajouter dans la routine ainsi créée un Hide() ;

Dans ZoomSpecifique.wxs, associer par exemple un des boutons (OK, Quitter, …). Dans le code créé ajouter :

// Boutton OK <=> OnClose

wxCloseEvent close\_event;

OnClose(close\_event);

Ainsi, un clic sur ce bouton aura le même effet qu’un clic sur la croix de fermeture, en haut à droite.

Attention : il faudra convertir au moins ZoomSpecifique.cpp en utf8, via Editer / Encodage du fichier / UTF-8, surtout si certaines entrées texte ont des accents.

à ce niveau, une regénération complète doit fonctionner et un clic sur le nouveau menu (Zoom Spécifique) doit provoquer l’affichage de la boîte de dialogue. Les boutons Quitter (ou OK) et la croix de fermeture en haut, à droite doivent aussi fonctionner.

Reste à remplir les autres fonctionnalités.

Il peut être utile de créer dans interface (.cpp et .h) un pointeur vers le panel créé afin de faciliter des initialisations par exemple.

Dans interface.h :

ZoomSpecifique \*MZoomSpec= nullptr;

Dans interface.cpp

if (MZoomSpec != nullptr) {

MZoomSpec->SpinCtrl\_LSI->SetValue(lround(m\_gldata.rotx));

MZoomSpec->SpinCtrl\_LAZ->SetValue(lround(m\_gldata.rotz));

}

Dans OvniMain.cpp (OuvrirFichier) :

Element->MZoomSpec= ZoomSpecifique\_Panel;