16/11/2020

Oscar Prêcheur

MEA5 Polytech Montpellier

FICHE TECHNIQUE

PFE : REPETEUR DE VOL POUR VOILIER

Ce document décrit les paramètres systèmes nécessaires pour la compilation des programmes du projet.

Un mode d’emploi de la suite de logiciels actuelle est aussi présent

**PARTIE 4 : Guide d’utilisation**

Table des matières

Fichier du projet disponible sur github https://github.com/oscarprecheur/PFE.git

# Configuration

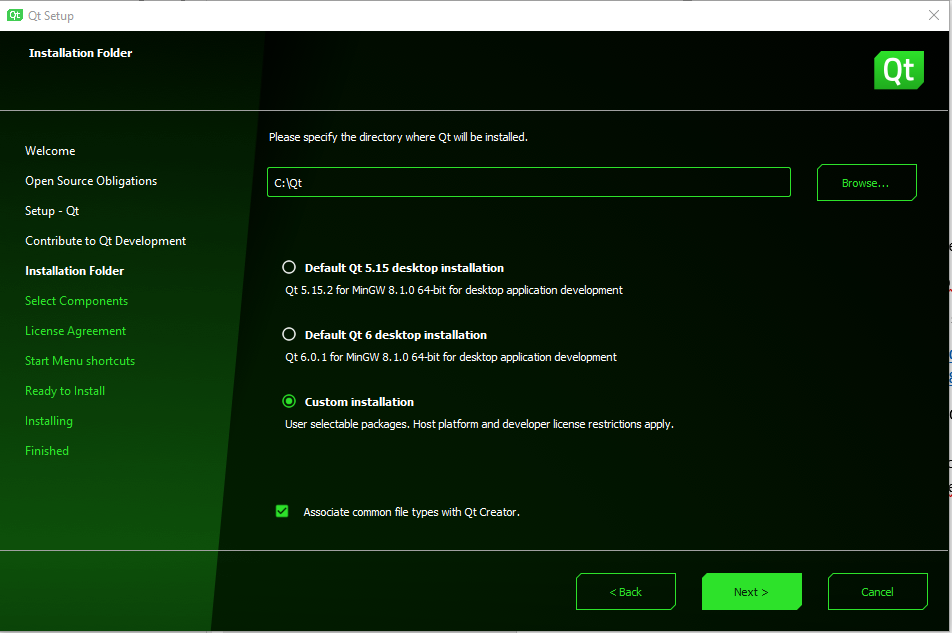
### Installation de Qt et QtCreator

Version de Qt utilisée : Qt 5.15.1 open source

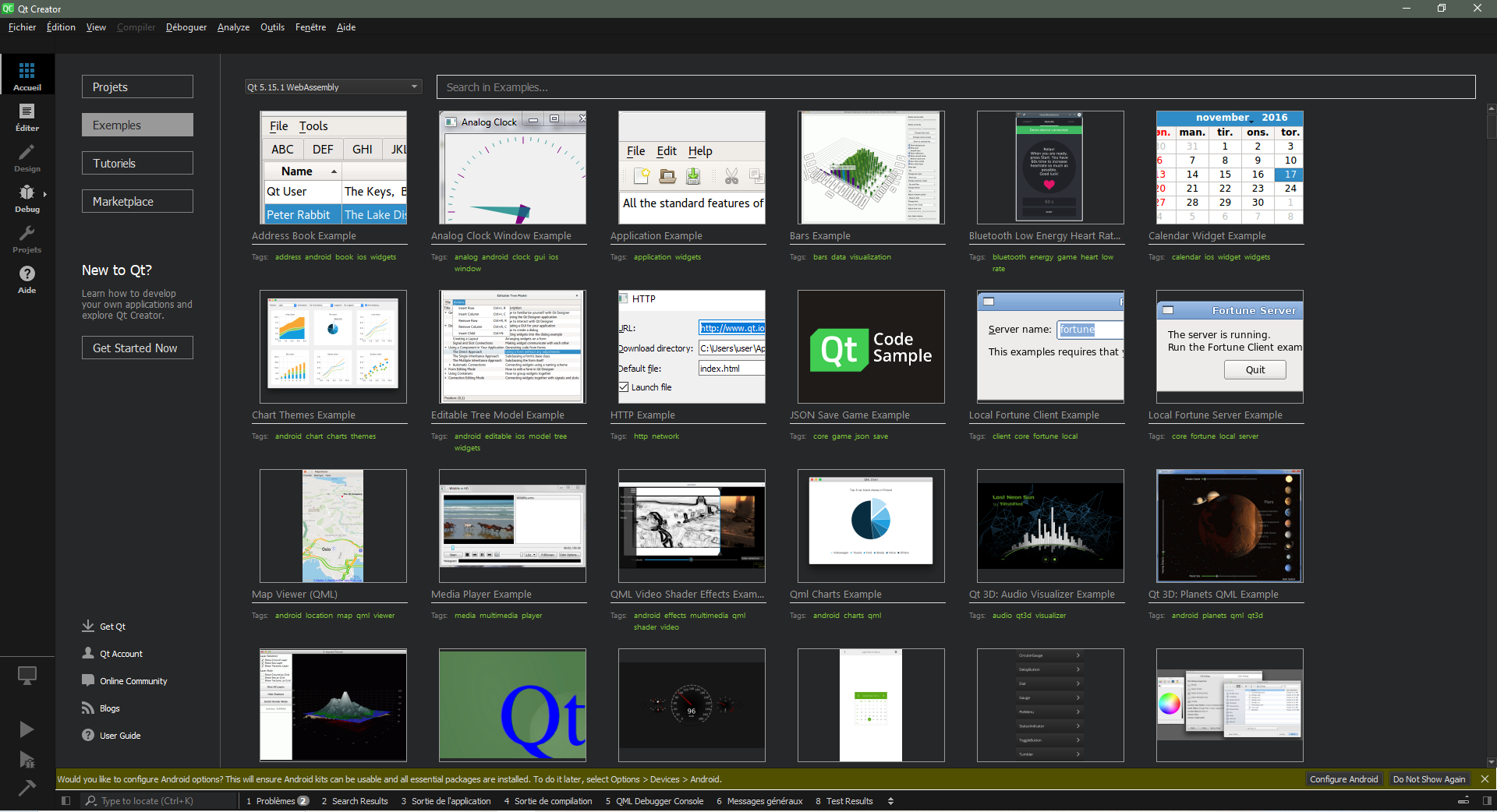
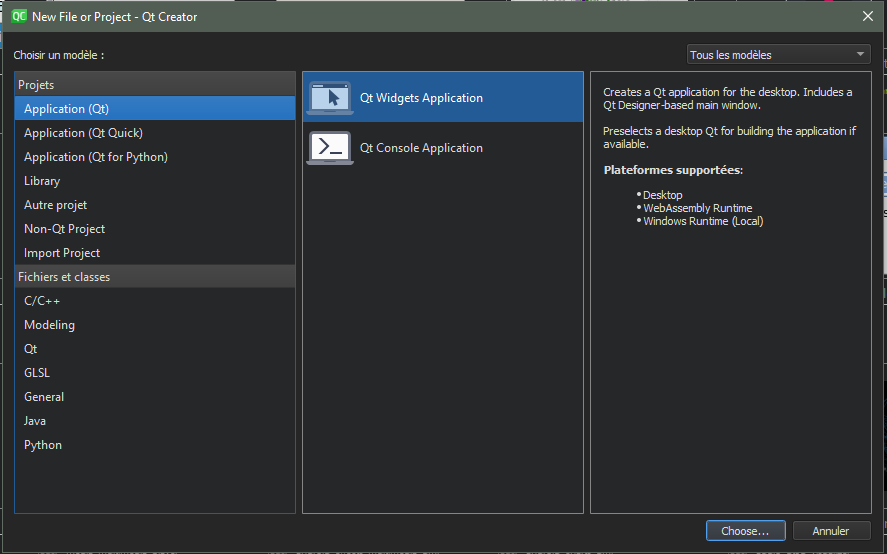
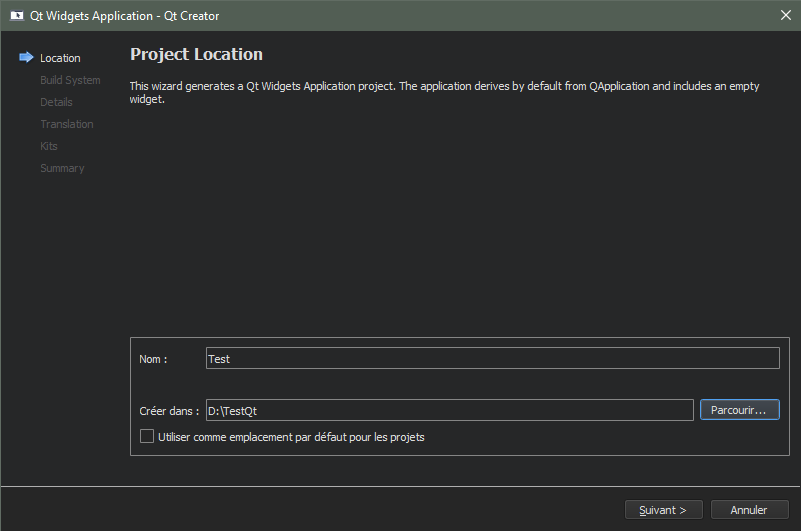
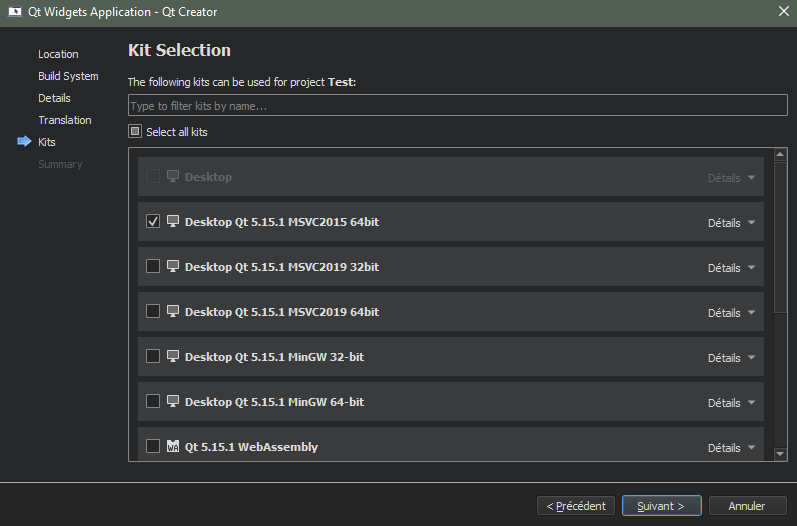
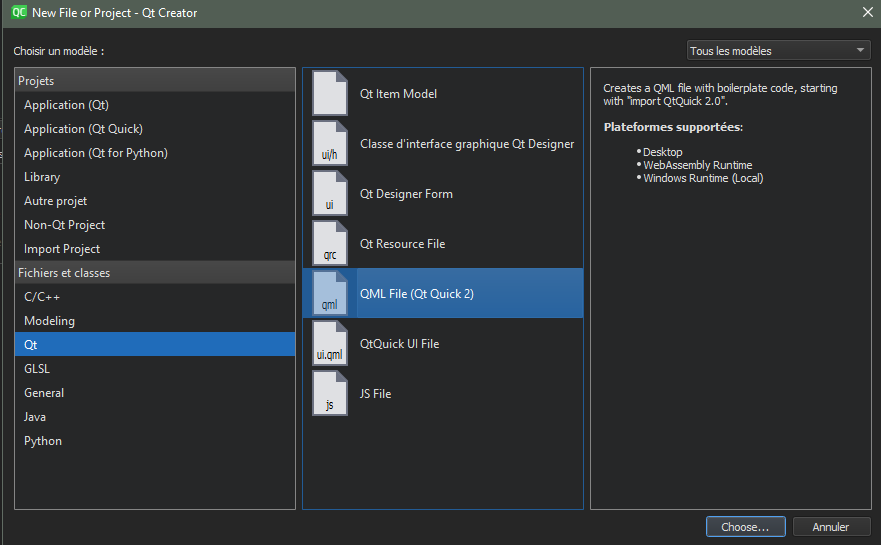
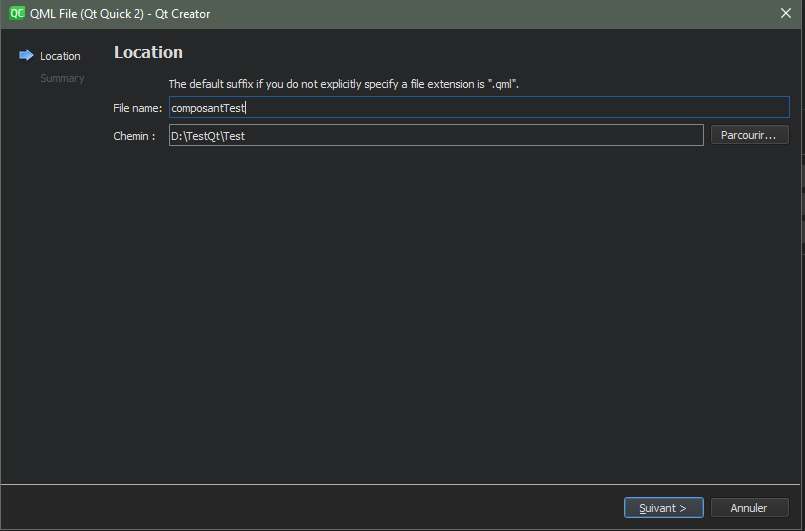
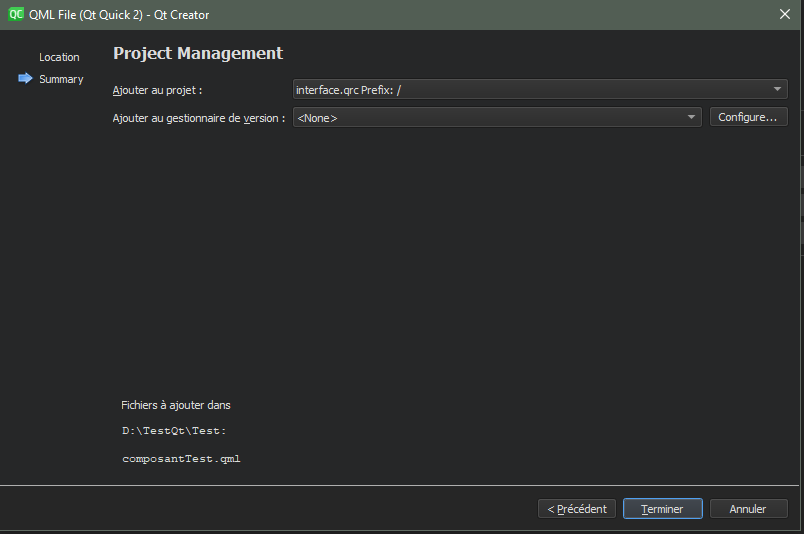
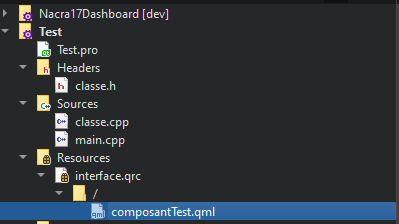
Version de l’IDE utilisé : QtCreator 4.14.0 (community)

Pour le téléchargement de Qt et Qt Creator, se rendre sur ce lien :

<https://www.qt.io/download-qt-installer?hsCtaTracking=99d9dd4f-5681-48d2-b096-470725510d34%7C074ddad0-fdef-4e53-8aa8-5e8a876d6ab4>

* Lancer l’installateur online (ATTENTION cet installateur ne fonctionne que sur des machines 64 bits )
* Se connecter à un compte Qt ou en créer un
* A la rubrique **Installation folders** selectionner indiquer le repertoire d’installation et choisir **custom installation**
* A la rubrique **Select Components**  seectionner
  + Additionnal Librairies/Qt/**Qt Image Format**
  + Qt/**Qt5.15.1**
  + Qt/Developer and Designer Tools/
    - **Qt Creatore 4.14.0 CDB Debugger Support**
    - **Debbuging Tools for Windows**
    - **Qt Creator 4.14.0 Debug Symbols**
    - **Qt Creator 4.14.0 Plugin Development**
    - **CMake 3.19.2 64-bit**
    - **Ninja 1.10.0**
* Select jusqu’à la fin

### Utilisation de Qt Creator

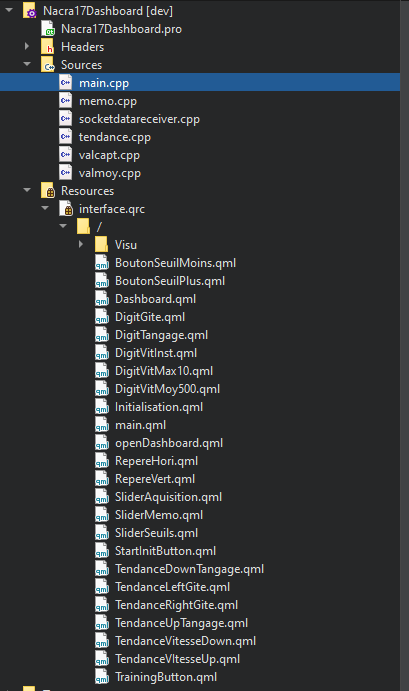
* Lancer Qt Creator, on se retrouve sur cette page : 
* Pour créer un nouveau projet :
  + Fichier/Nouveau Fichier ou Projet …
  + Choisir Application(Qt) puis Qt Widgets Application
  + Choisir un repertoire pour le projet 
  + Build Système : laisser qmake
  + Detail : remplacer MainWindow par le nom de votre 1ere classe, mettez base classe en QWidget et décocher Generate form
  + Translation : ne rien faire
  + Kits choisir Desktop Qt 5.15.1 MSVC2015 64bit
  + Suivant jusqu’à la fin
* Pour ouvrir un projet existant
  + Fichier/Ouvrir un fichier ou projet..
  + Choisir le fichier .pro
* Pour ajouter une classe C++ à un projet :
  + Fichier/Nouveau Fichier ou Projet …
  + C/C++ / C++ Class
* Pour ajouter un composant QML à un projet :
  + Fichier/Nouveau Fichier ou Projet …
  + Si la ressource n’existe pas
    - Qt/ Qt Ressource File
    - Nomer le fichier ressource
    - Puis creez les composant QML
  + Qt / QML File (Qt Quick 2) 
  + Choisir le nom du composant 
  + A la fenetre project manager : A Ajouter au projet choisir le nom du fichier ressource du projet (.qrc) ici c’est interface.qrc
  + On se retrouve avec un projet de ce type 

Dans Header et Source on va retrouver tous les programmes C++ et les classes

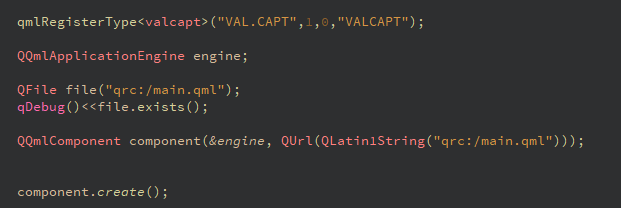
Dans Ressoruce/Interface/qrc on va retrouver tout les composants QML

### Appel d’un composant QML dans le main.cpp

On prend l’exemple du projet dashboard :



On se rend dans le program main.cpp

Voici la suite de ligne de code permettant de créer un composant QML

Ligne 1 : On créer un registre peremttant de lier les Q\_PROPERTY d’un programme C++ à un programme QML , on va pouvoir utiliser des méthode d’un classe C++ dans le programme QML ici on prend la classe « valcapt » dans un registre qu’on va nommer VAL.CAPT 1.0, il sera appelé dans le programme QML en tant que composant nommé VALCAPT

Ligne 2 : Instanciation du moteur d’application

Ligne 3 et 4 : permettent de déterminer si le fichier QML à lier existe (« qrc:/main.qml »)

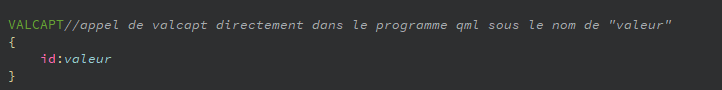
Linge 5 : on instancie le composant QML via QQmlComponent, on le lie au moteur d’application et on donne son URL (« qrc:/main.qml »), qrc:/ permet d’indiquer qu’on va chercher un fichier inclus dans le fichier ressource

Ligne 6 on créer le composant , ici le compoansant main.qml

Pour utiliser les méthode d’une classe c++ dans le QML :

 on import le registre crée dans le main.cpp

On appel le composant correspondant :



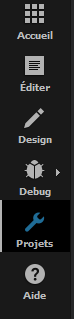
L’id est l’indentifiant du composant. Pour utiliser les méthodes on utilise ce nom de la façon suivante :

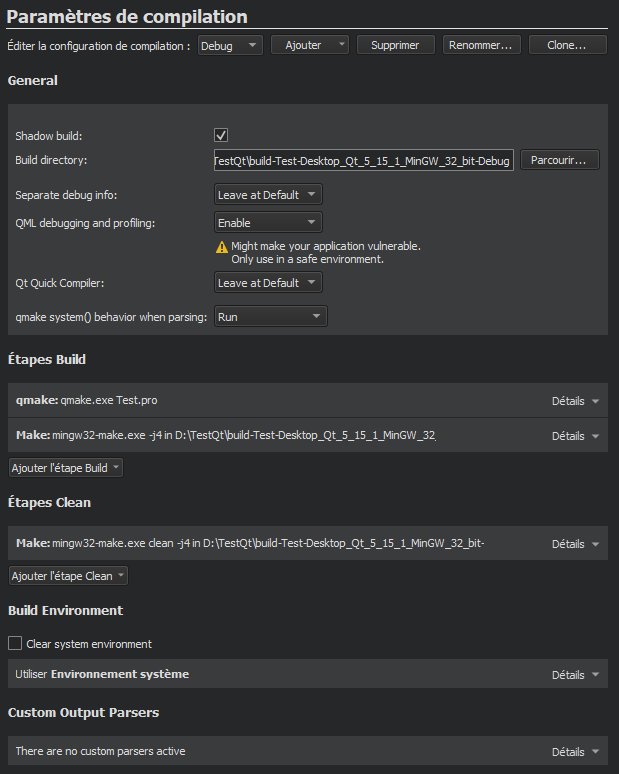


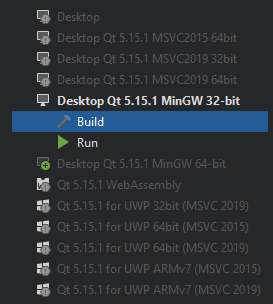
Valeur.<méthode C++>

### Compilation avec QtCreator

* Choix du fichier de build
  + Aller dans Projets



* + Dans éditer la configuration de compialtion choisir **Debug**
  + Choiri le ficiher dans build directory
  + Puis laisser les paramètres existants
  + 
* Dans build & run choisirDesktop Qt5.15.1 MinGW 32-bit ou Desktop Qt5.15.1 MSCV2015 64bit



Pour compiler et build le programme appuyer sur le marteau

Pour compiler et lancer le programme Appuyer sur le bouton play en bas à gauche

Pour un debug le bouton play avec le pictogramme insecte



Toujours exécuter Qmake avant de lancer une compialtion pour cela clique droit sur le repertoir de projet et exécuter Qmake

### Déploiement du projet

Pour déployer un projet Qt , suivre le tuto suivant sur internet :

<https://www.siloged.fr/cours/QTCreator/DeployeruneapplicationQT.html>

### 

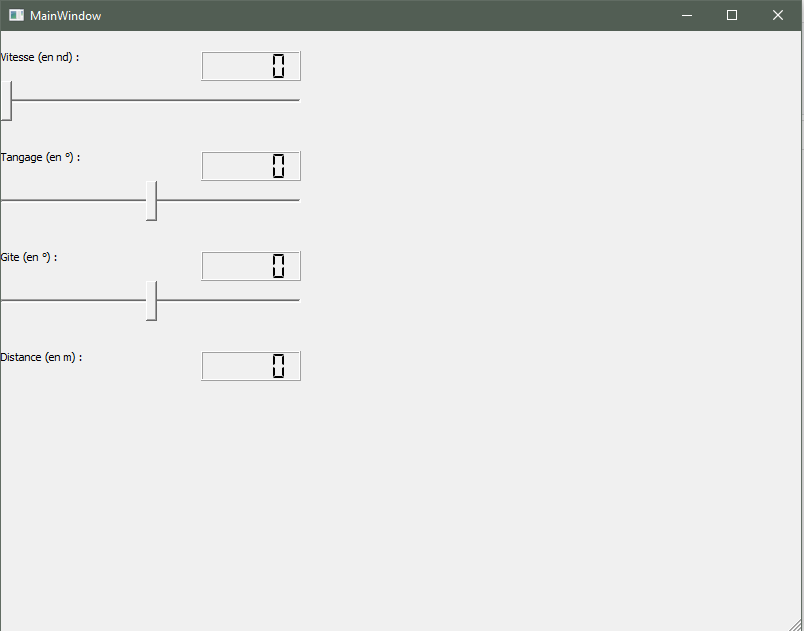
### Modes d’emploi du système

Modes d’emploi de la suite d’applications développées durant le PFE.

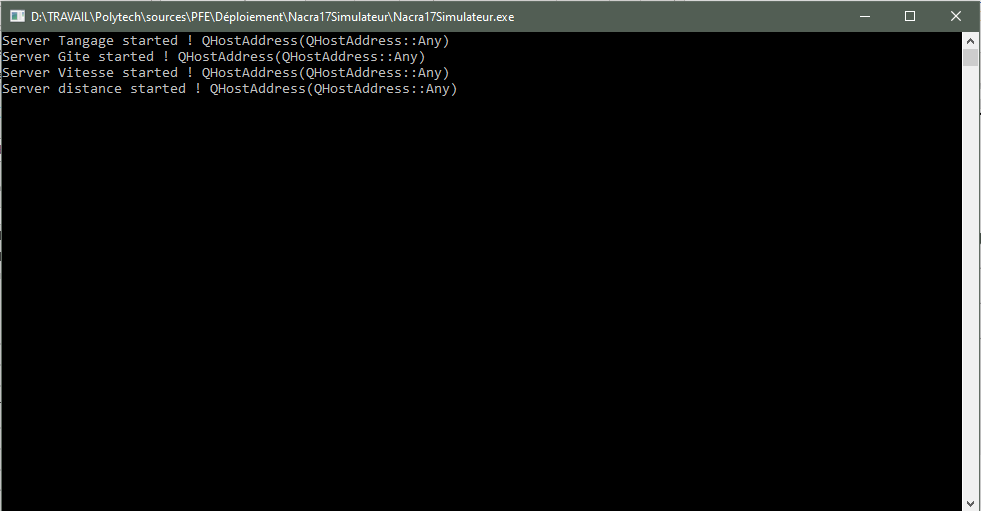
### Utilisation du simulateur

ATTENTION :Pour garantir un bon fonctionnement, est important de toujours lancer Nacra17Simulateur **AVANT** le programme Nacra17Dashboard !

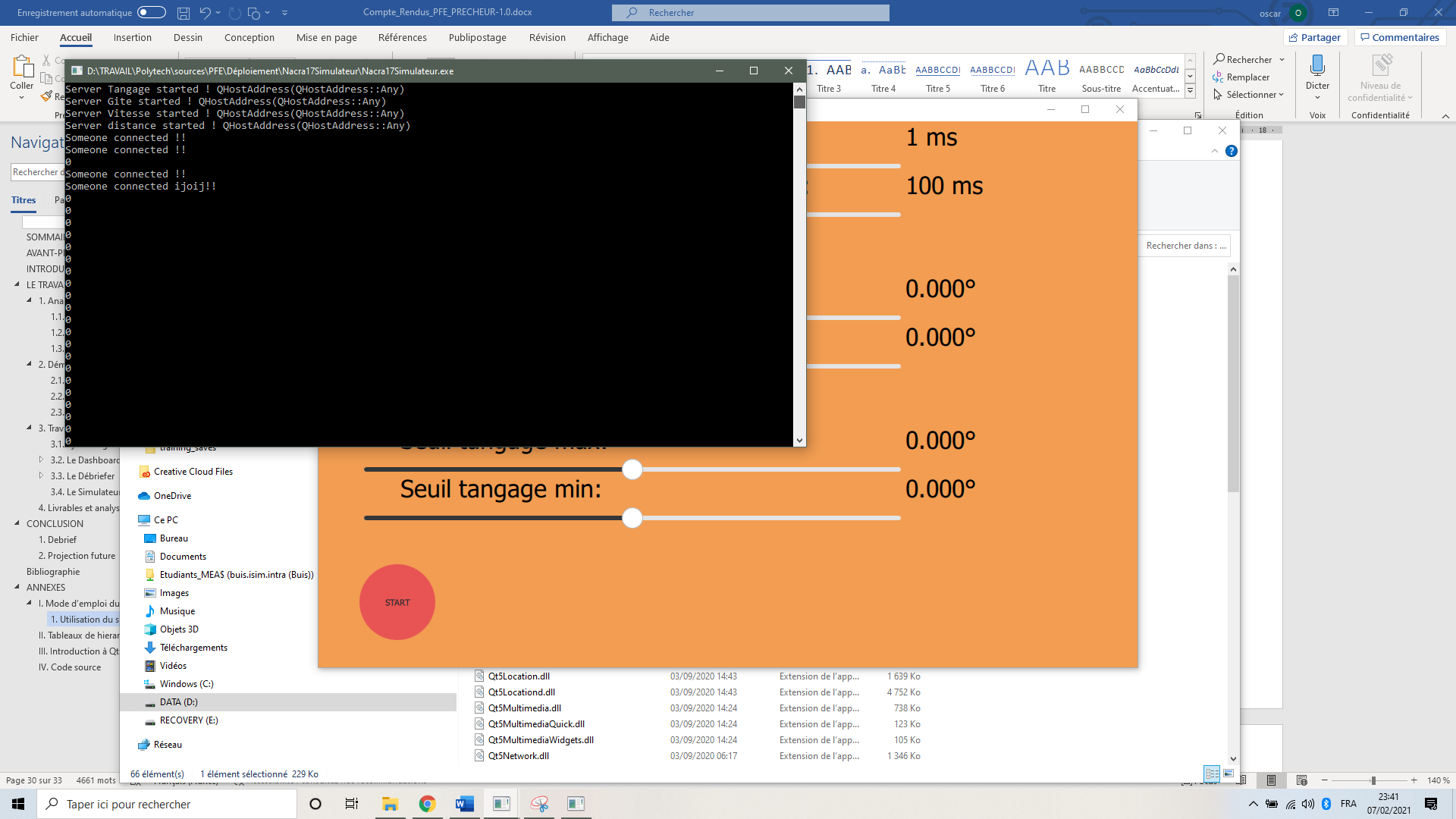
* 1. Se rendre dans le répertoire **\*/sources/PFE/Déploiement/Nacra17Simulateur** présent dans le code source (voir annexe code source).
  2. Lancer l’application **Nacra17Simulateur.exe**
  3. La fenêtre suivante s’affiche :



Ainsi que la console suivante :

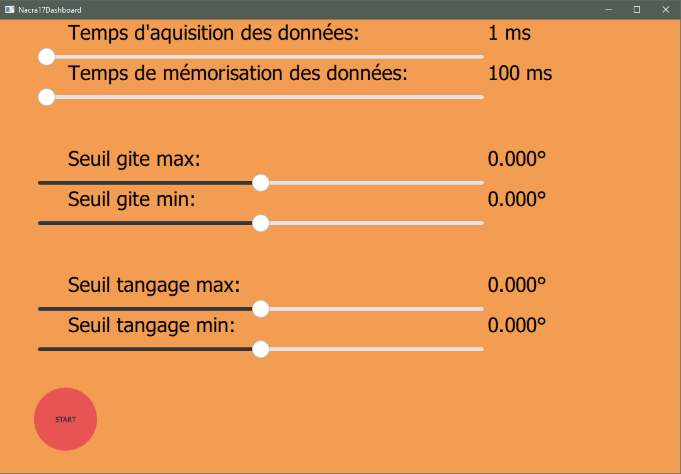


* 1. Le simulateur est en attente du Nacra17Dashboard , lancer l’application Nacra17Dashboard comme détaillé dans la partie 2. .
  2. La console affiche :



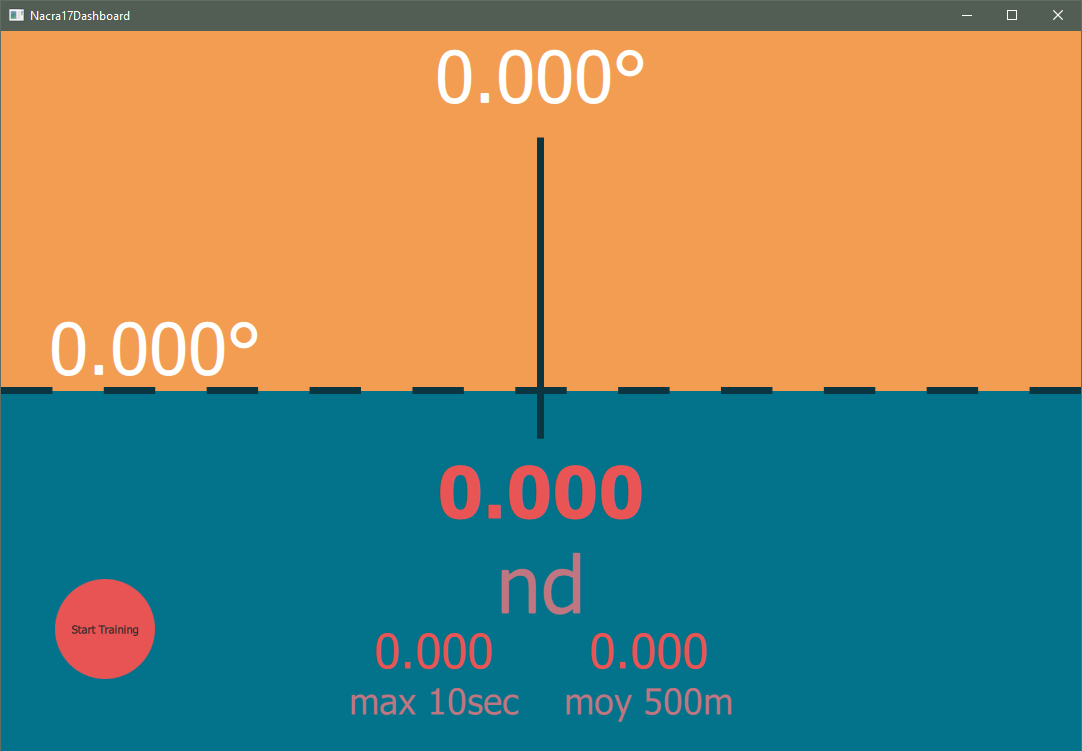
Le simulateur est prêt à fonctionner. Pour changer les valeurs il suffit de faire coulisser les curseurs.

### Utilisation du dashboard

* 1. Se rendre dans le répertoire **\*/sources/PFE/Déploiement/Nacra17Dashboard** présent dans le code source (voir annexe code source).
  2. Lancer l’application **Nacra17Dashboard.exe**
  3. La fenêtre de paramétrage suivante s’affiche :
* **Temps d’acquisition des données** : intervalle de temps entre 2 lectures des valeurs capteurs (conseillé : minimum)
* **Temps de mémorisation des données**: intervalle de temps entre chaque enregistrement des données en mode entrainement
* **Seuils**: valeurs de déclanchement d’alerte si dépassée

Paramétrer le tableau de bord et appuyer sur start

* 1. La fenêtre suivante s’affiche :



Eau

Ciel

Gite

Bouton start/stop enregistrement de séance

Moyennage vitesse

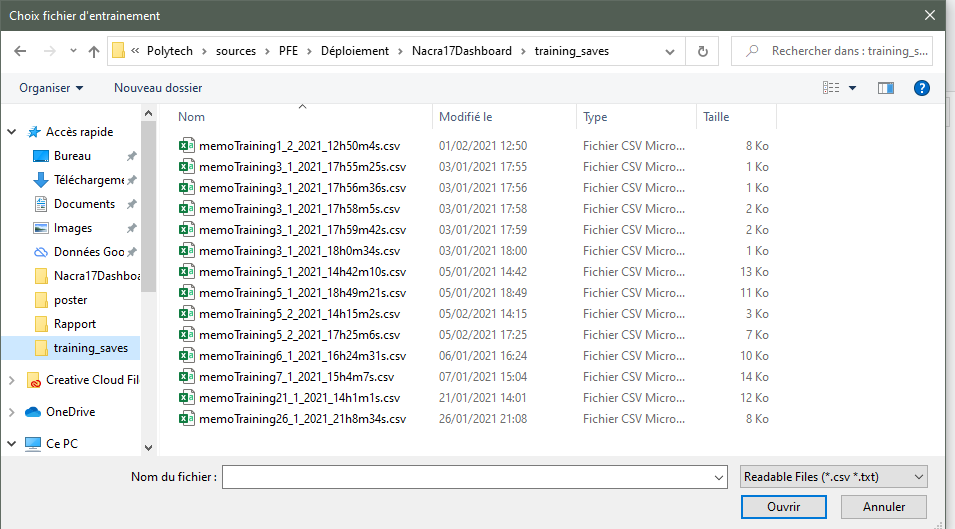
Tangage

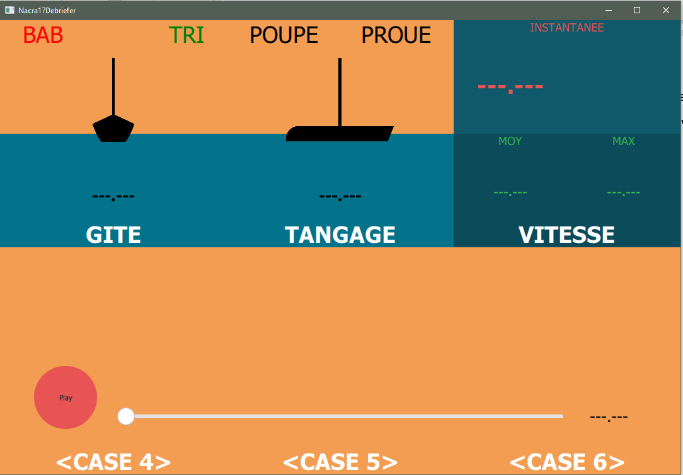
Vitesse GPS en nœud

* 1. Pour enregistrer un entrainement : appuyer sur le bouton « Start Training », pour arrêter l’enregistrer : appuyer à nouveau sur le bouton. Un nouveau démarrage d’entrainement entraine la création d’un nouveau fichier de sauvegarde. Ces fichiers sont situés **\*/sources/PFE/Déploiement/Nacra17Dashboard/Training\_saves**

### Utilisation du débriefer

* 1. Se rendre dans le répertoire **\*/sources/PFE/Déploiement/Nacra17Debriefer** présent dans le code source (voir annexe code source).
  2. Lancer l’application **Nacra17Debriefer.exe**
  3. Une fenêtre de dialogue s’affiche. Se diriger vers le répertoire **\*/sources/PFE/Déploiement/Nacra17Dashboard/Training\_saves** et choisir le fichier .csv correspondant à l’entrainement à étudier (ils sont indiqués par leur date de création)



* 1. Après ouverture du fichier la fenêtre suivante s’affiche : 

Indicateur de temps

Curseur timeline

Bouton Play/Pause

Lancer la lecture avec le bouton ou parcourir l’entrainement à l’aide du curseur de la timeline