TP logiciel 1 : Découverte de l'environnement QT Creator

Durée: 4h

Objectifs:

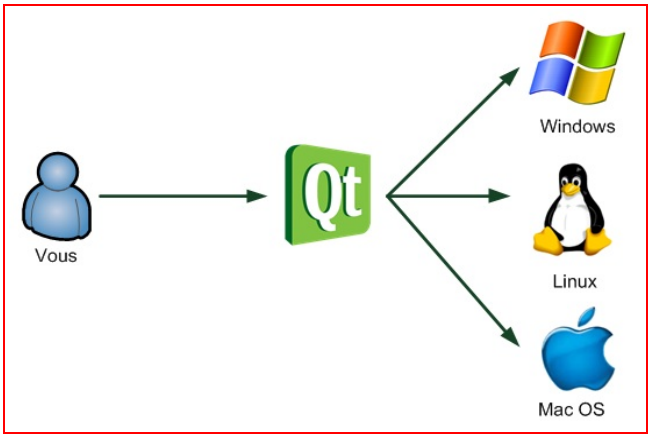
- Utiliser l'environnement de développement QT Creator
- S'initier à un outil de développement pour Windows.
- Utiliser Qt Creator en mode console
- Produire un algorithme et le coder en langage C++;
- Mise en oeuvre d'un "*main*",
- Déclaration d'une classe, déclaration de variables et notion d'appel de fonctions C++.

But: Développer 5 exercices en C++, mode console dans l'environnement IDE Qt Creator.

Présentation de l'environnement de développement QT :

Qt est une bibliothèque multi-plateforme pour créer des GUI (programme sous forme de fenêtre).

Qt est en fait... bien plus qu'une bibliothèque. C'est un ensemble de bibliothèques. Le tout est tellement énorme qu'on parle d'ailleurs plutôt de framework : cela signifie que vous avez à votre disposition un ensemble d'outils pour développer vos programmes plus efficacement.



Qt est donc constituée d'un ensemble de bibliothèques, appelées "modules". On peut y trouver entre autres ces fonctionnalités :

- **Module GUI** : c'est toute la partie création de fenêtres. Nous nous concentrerons surtout sur le module GUI dans ce cours.
- **Module OpenGL** : Qt peut ouvrir une fenêtre contenant de la 3D gérée par OpenGL.
- **Module de dessin** : pour tous ceux qui voudraient dessiner dans leur fenêtre (en 2D), le module de dessin est très complet !
- **Module réseau** : Qt fournit une batterie d'outils pour accéder au réseau, que ce soit pour créer un logiciel de Chat, un client FTP, un client Bittorent, un lecteur de flux RSS...
- **Module SVG** : possibilité de créer des images et animations vectorielles, à la manière de Flash.
- **Module de script** : Qt supporte le Javascript (ou ECMAScript), que vous pouvez réutiliser dans vos applications pour ajouter des fonctionnalités, sous forme de plugins par exemple.
- **Module XML** : pour ceux qui connaissent le XML, c'est un moyen très pratique d'échanger des données avec des fichiers formés à l'aide de balises, un peu comme le XHTML.
- **Module SQL** : permet un accès aux bases de données (MySQL, Oracle, PostgreSQL...).

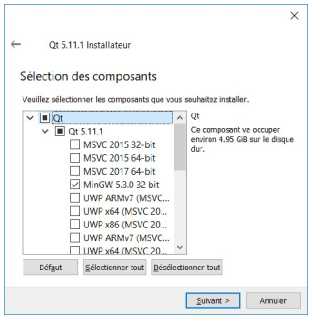
Installation de Qt : Commencez par installer Qt , version proposée : 5.11.1.exe
Fichier : qt-opensource-windows-x86-5.11.1.exe

ou télécharger la version de Qt sur le site: **https://download.qt.io/archive/qt/5.11/5.11.1/**

version de Qt 5..... pour Windows 32-bit (MinGW 5.3.0, OpenGL).

L'installation sous Windows se présente sous la forme d'un assistant d'installation classique.

login : rostand.snir@gmail.com Mdp: QtJR2020



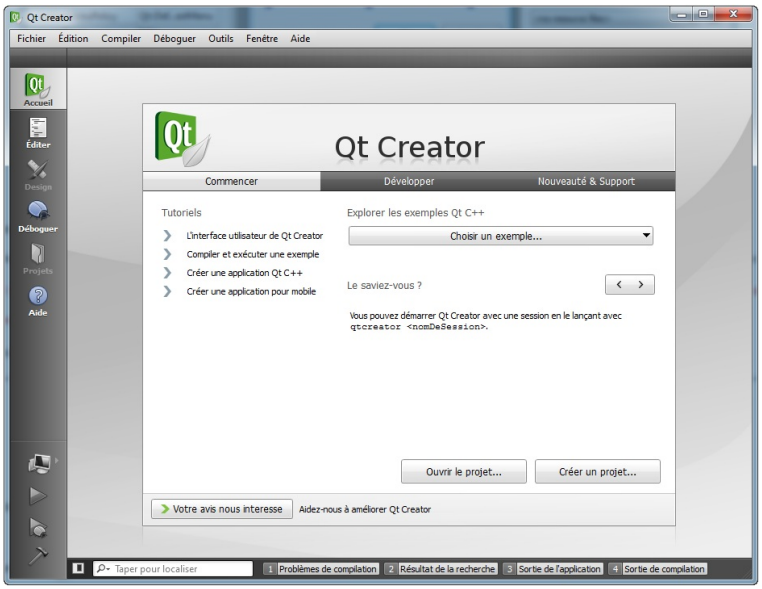
N'oubliez pas de sélectionner l'outil de compilation.
Développez l'onglet : Qt 5.11.1
et cochez l'onglet MinGw 5.3.0 32bits

QT Creator :

Bien qu'il soit possible de développer en C++ avec Qt en utilisant notre IDE (environnement de développement intégré) Il est préférable d'utiliser l'IDE Qt Creator que vous venez d'installer. Il est particulièrement optimisé pour développer avec Qt. En effet, c'est un programme tout-en-un qui comprend entre autres :

- Un **IDE pour développer en C++**, optimisé pour compiler des projets utilisant Qt (pas de configuration fastidieuse)
- Un **éditeur de fenêtres**, qui permet de dessiner facilement le contenu de ses interfaces à la souris
- Une **documentation** in-dis-pen-sable pour tout savoir sur Qt.

Qt Creator lorsque vous le lancez pour la première fois :



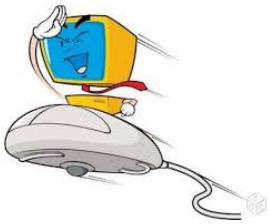
Qt Creator est l'IDE idéal pour programmer des projets Qt.

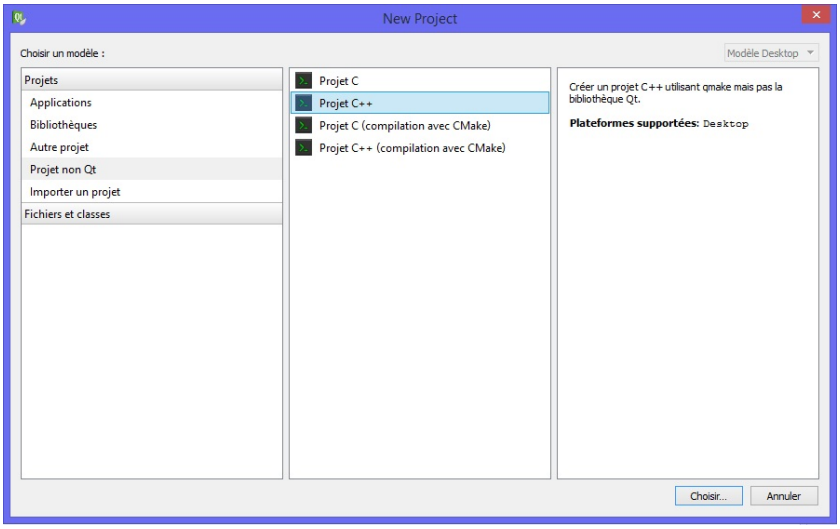
Travail demandé:

1) Création d'un projet C++

Créer un nouveau projet en allant dans le menu **Développer**, puis cliquer sur **Créer un Projet**. On vous propose plusieurs choix selon le type de projet que vous souhaitez créer : application graphique pour ordinateur, application pour mobile, etc. Cependant, pour débuter, il est préférable de commencer avec un projet simple.

- Cliquer sur **Projet non Qt** puis **Projet C++** et enfin sur **choisir**.



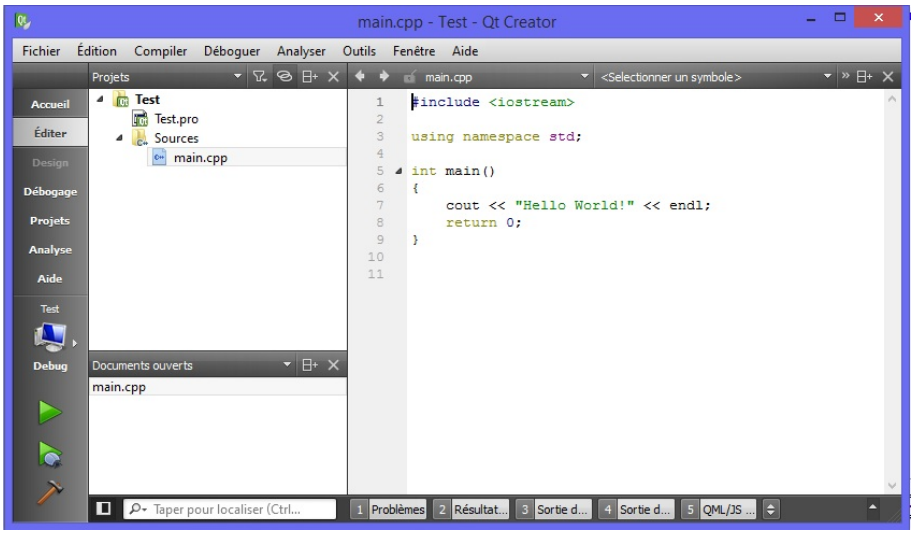


Un assistant s'ouvre alors pour vous demander le nom du projet et l'emplacement où vous souhaitez l'enregistrer :

Définir un répertoire de travail et un nom de projet judicieux pour la sauvegarde.

- Créer un répertoire portant votre ***nom_Prénom*** dans la partition D du disque dur
- Définir un nouveau ***Projet C++*** et sauvegarder le sous le nom "**Nom élève_Tp01**"dans votre répertoire.

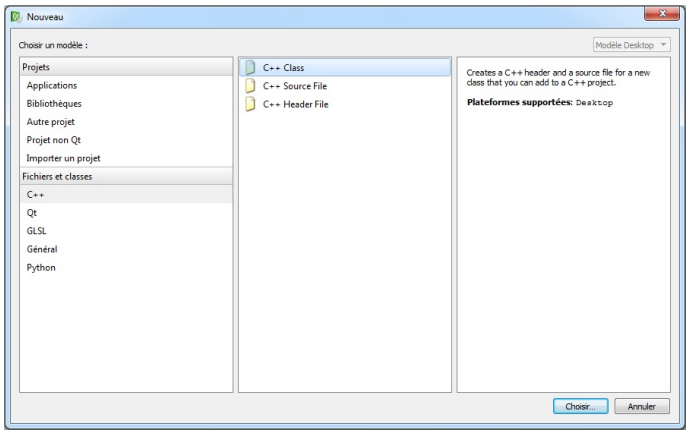
- **Vous obtenez votre programme de base.**



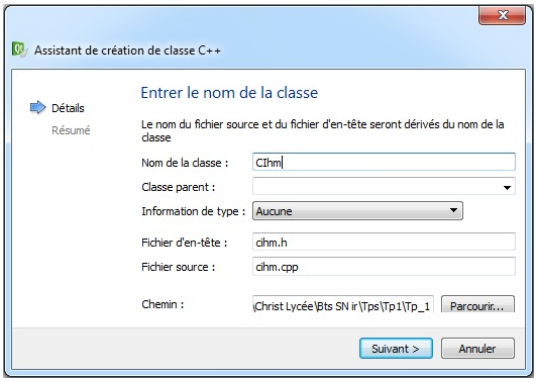
Le projet est constitué de 2 fichiers. Un fichier **.pro**, propre à Qt, qui sert à configurer le projet au moment de la compilation. Qt Creator modifie automatiquement ce fichier en fonction des besoins, et un fichier **main.cpp** qui contiendra le code programme du langage C++.

Comme nous développons en langage C++, nous allons créer une classe **CIhm**

Cliquer sur **Fichier** puis **Nouveau fichier ou projet**... puis choisir **C++** et **C++ Class**



nom de la classe : **CIhm**



- Respecter les majuscules et les minuscules et ne rien mettre dans la case **Classe parent**
- Faire **Suivant >**

2 fichiers ont été créés : **cihm.h** et **cihm.cpp**

- Modifier le fichier entête **cihm.h** et le fichier Source **cihm.cpp** de tel manière à obtenir :

```
#ifndef CIHM_H
#define CIHM_H

#include <iostream>
using namespace std;

class CIhm
{
public:
    CIhm();
    void fonctionPrincipale();
};

#endif // CIHM_H
```

```
#include "cihm.h"

CIhm::CIhm()
{
}

void CIhm::fonctionPrincipale()
{
    cout << "Bonjour Bts SNIR 2021" <<endl;
}
```

puis modifier le fichier Source **main.cpp** tel que:

```
#include "cihm.h"

int main( )
{
    CIhm ihm;
    ihm.fonctionPrincipale( );
    return 0;
}
```

- Tester le programme de base en cliquant sur **la flèche verte**

2) Remplacer le code, dans la procédure **fonctionPrincipale( )** de la classe CIhm comme ci dessous.

```
//-------------------------------------------------------------------------
#include <iostream>    /* bibliothèque d'entrees-sorties standard en C++,  la ligne
montre l'utilisation d'une commande du préprocesseur (#include) qui permet d'inclure
un fichier dont le nom est passé en argument. Les symboles < et > indiquent qu'il faut
rechercher ce fichier dans les répertoires connus du compilateur. Le préprocesseur
remplacera cette commande #include par le contenu du fichier iostream utile pour
notre application pour la fonction cout. */

#include <conio.h>
using namespace std;

void CIhm::fonctionPrincipale( )
{
   int a, b, calcul ;  /* déclaration de 3 variables de type entière */
   int u ;
   char v ;
   cout << "BONJOUR\n\n"; /* utilisation d'une fonction-bibliothèque */
   a = 10 ; /* affectation*/
   b = 50 ;  /* affectation */
   u = 65 ;
   v = 'A' ;
   calcul = (a + b)*2 ;  /* affectation et opérateurs   */
   cout << "Voici le resultat : "<< calcul << "\n\n" ;
   cout << "1er affichage de u : " << u << endl ;
   cout << "2ieme affichage de u : " << (char)u <<  "\n\n" ;
   cout << "1er affichage de v: "<< v << endl ;
   cout << "2eme  affichage de v: "<< (int)v << "\n\n";
   cout << "Pour continuer frapper une touche..." << endl <<endl ;
   getch( );  /* fonction saisie d'un caractère clavier */
}
```

- Analysez la production de ce code et commentez le rôle de chaque ligne du programme.

- Modifiez dans votre programme les variables a, b, u et v par des valeurs différentes et analysez l'instruction ***cout*** associée à ses paramètres.

3)

- Ajouter dans la déclaration de la classe **CIhm** (fichier **cihm.h**), une nouvelle procédure tel que : "**void fonctionExercice2( );**"

- Tapez le code de la fonction "**void CIhm::fonctionExercice2( )**"dans le fichier **cihm.cpp** :

```
void CIhm::fonctionExercice2( )
{
  char c;
  c = 66;          /* c est le caractere alphanumerique B */
  cout << (char)c << endl ;
  cout << (int)c << endl ;
  cout << oct << (int)c << endl;
  cout << hex << (int)c << endl;
  cout << dec << (int)c << endl;
  cout << "Pour continuer frapper une touche..." << endl << endl ;
  getch( );  /* Attente d'une saisie clavier */
}
```

- Pour tester le programme, ajouter dans la fonction **main()** (fichier **main.cpp**), l'appel de la procédure: **ihm.fonctionExercice2( )**

- Tester votre programme et donnez le rôle des manipulateurs de flux des objet ***cout*** **.**

- Affecter à la variable ***c*** la valeur **322** au lieu de 66. Commentez les résultats de chaque ligne ayant un ***cout*** .

- Remplacer le type ***char*** de la variable ***c*** par un type ***int***. Commentez les résultats de chaque ligne ayant un ***cout*** .

Pour améliorer la qualité de l'affichage, retapez le programme en utilisant les fonctions suivantes**: *system("cls"); system("pause");*** *(inclure la librairie "**stdlib.h**")*

N'oubliez pas de consulter l'aide de Qt ou d'internet pour ces nouvelles fonctions.

4) Ajouter une nouvelle procédure à la classe **CIhm** : "**void fonctionExercice3( );**"

- Tester le suivant et conclure:

```
void fonctionExercice3( )
{
  char c = 'A';   /* c est le caractere alphanumerique A */
  cout << "decimal = "<< (int)c << "ASCII = "<< c<<endl;
  cout << "Pour continuer frapper une touche..." << endl ;
  getch( );  /* Attente d'une saisie clavier */
}
```

5) Ajouter une nouvelle procédure à la classe **CIhm** "**void fonctionExercice4( );**" et la coder tel que :

- a et b sont des entiers, a = -21430 b = 4782,

- Calculer et afficher a+b, a-b, a*b, a/b, a%b en format décimal, et en soignant l'interface homme/machine.

Remarque: a/b donne le quotient de la division, a%b donne le reste de la division.

6) Ajouter une nouvelle procédure à la classe **CIhm** "**void calculPerimetreSurface( );**" et la coder tel qu'on puisse calculer le périmètre et la surface d'un cercle à partir de son rayon.

- On introduit l'objet ***cin*** pour saisir au clavier l'information du rayon.

- En utilisant l'aide d'internet, établir un algorithme permettant de décrire cette procédure qui permet de saisir le rayon d'un cercle au clavier et qui nous donne à l'écran le périmètre et la surface de ce cercle.

- Coder la procédure à l'image de votre algorithme.

- Améliorer votre programme, on souhaite afficher les résultats avec deux chiffres après la virgule(le point) au maximum.

Annexe: Diagramme de classe du programme Tp1.