

## Projet C++

### “Élections piège à cons”



Janvier 2017

## Table des matières

1) Procédure d'installation .....	3
2) Description de l'application développée.....	3
3) UML.....	6
4) Les parties de l'implémentation dont nous sommes fiers .....	7

## 1) Procédure d'installation

Pour ce projet nous avons utilisé Qt Creator, deux options se présentent à vous pour pouvoir lancer notre programme :

- Soit vous n'installez pas Qt Creator, et dans ce, cas vous pouvez essayer de compiler vous même notre projet en tapant :

**make**

- Puis exécutez directement en tapant : (Si jamais la compilation avec make ne fonctionne pas, nous avons fait parvenir l'exécutable déjà compilé)

**./Election**

- Soit vous décidez d'installer Qt Creator, une fois Qt Creator installé, vous pouvez compiler notre projet en tapant :

**qmake -project -> qmake -> make -> ./Election**

(S'il y a une erreur dans ui\_Mainwindow, mettre la ligne 120 en commentaire)

## 2) Description de l'application développée

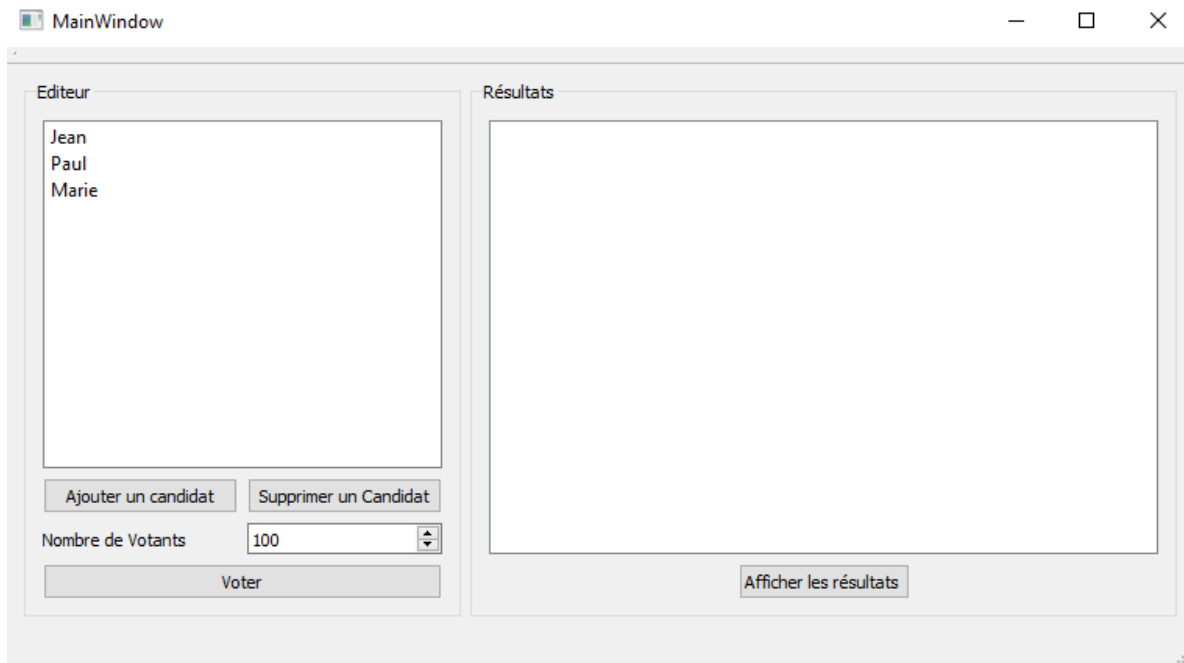
Nous nous sommes inspiré de la vidéo suivante pour notre projet :

<https://www.youtube.com/watch?v=ZoGH7d51bvc&t=59s>

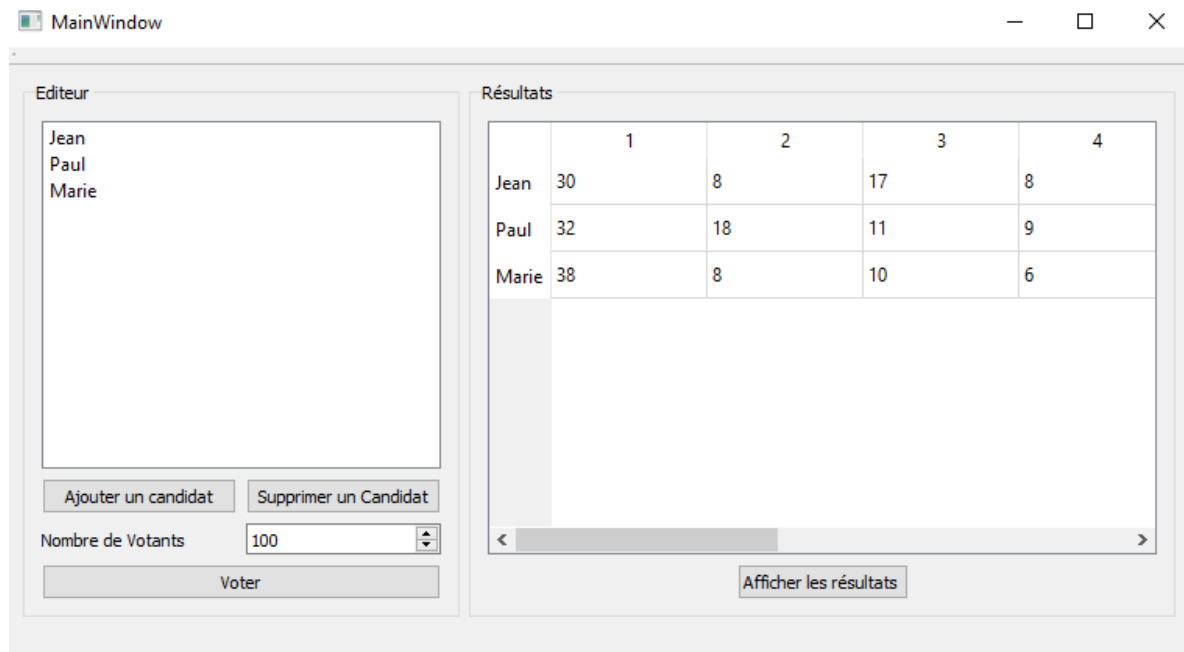
Ainsi, nous avons implémenté un système de vote aléatoire basé sur le vote du "Jugement majoritaire".

Le **jugement majoritaire** est un mode de scrutin proposé par deux chercheurs français, Rida Laraki et Michel Balinski. C'est une forme de vote par valeurs. Avec ce mode de scrutin, les électeurs donnent pour chacun des candidats une mention "Excellent", "Très bien", "Bien", "Assez bien", "Passable", "Insuffisant" ou "À rejeter". Pour chaque candidat, on calcule alors la mention majoritaire définie comme la médiane de ses mentions, c'est la mention qui est telle qu'au moins 50 % des électeurs la trouve valable. Le candidat élu est celui qui obtient la meilleure mention majoritaire. Chaque électeur attribue une note aux candidats en jugeant de leur capacité à occuper la fonction (l'échelle des notes a peu d'influence). Un électeur peut noter identiquement plusieurs candidats. Le candidat élu est celui dont la médiane des notes est la plus élevée.

Nous avons donc tenté de reproduire ce système de vote sur Qt Creator :

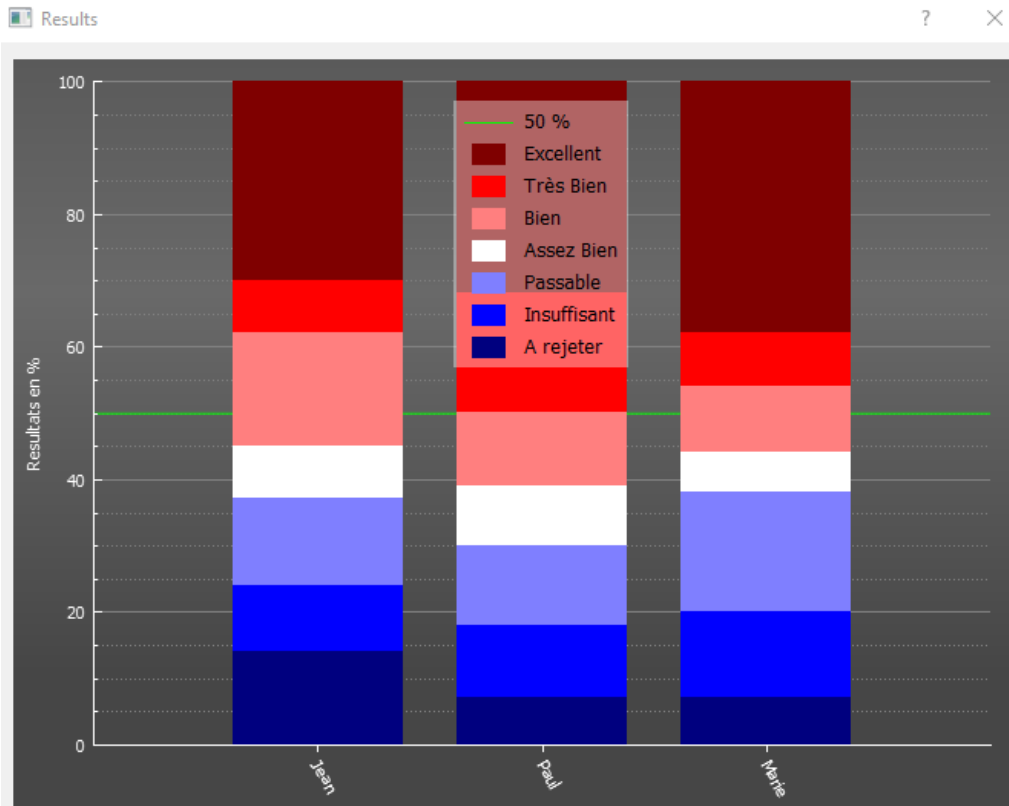


Sur cette fenêtre, vous pouvez “Ajouter un candidat”, “Supprimer un candidat”, changer leur nom et modifier le nombre de votants. Après avoir initialiser le système de vote, appuyez sur “Voter”.

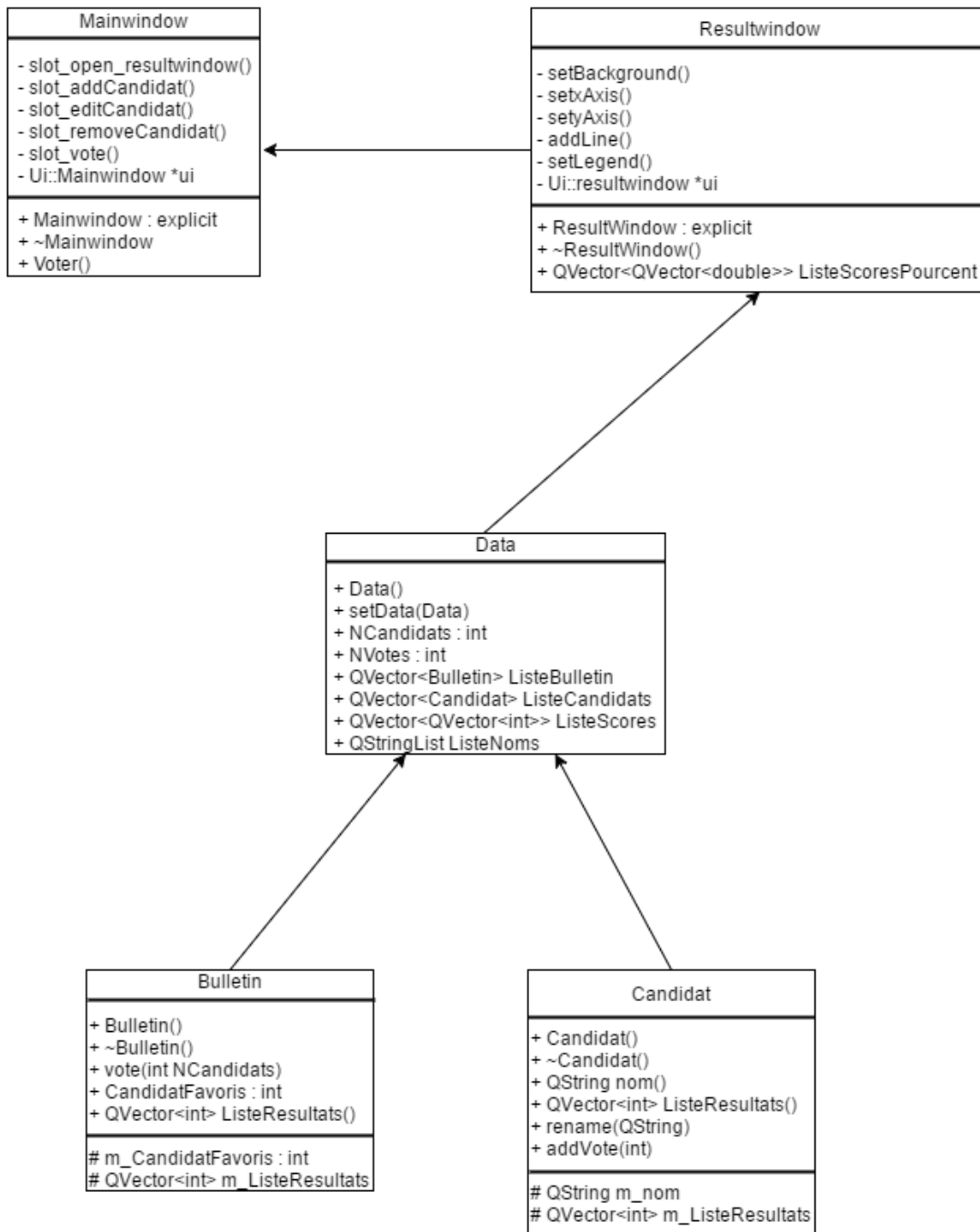


Les résultats s’affichent, ils sont copiés dans un tableau avec le nombre de votants pour chaque mention : 1 correspondant à “Excellent”, 2 à “Très bien”, etc.

Pour mieux se rendre compte du gagnant du vote, appuyez sur “Afficher les résultats”. Sur cet exemple, on voit que le gagnant du vote est Paul.



### 3) UML



#### 4) Les parties de l'implémentation dont nous sommes fiers

Il n'y a pas de partie de l'implémentation dont nous sommes le plus fiers, mais le projet dans sa globalité en est une.

Ce projet a été très enrichissant pour nous, il nous a permis de mettre en pratique les connaissances acquises au cours de l'année en C++ et également de nous familiariser avec Qt Creator, ce qui nous a permis de vous proposer une petite interface graphique.

Ayant été submergés par un nombre important de projets et d'examens ces deux derniers moi, nous n'avons malheureusement pas pu faire tout ce que nous voulions sur ce projet mais sommes tout de même content d'avoir pu vous rendre quelque chose. D'ailleurs, nous tenions à vous présenter nos excuses car nous n'avons donc pas pu respecter toutes les consignes du cahier des charges. En effet, nous n'avons que 5 classes et n'avons pas utilisé de surcharges d'opérateurs ni de fonctions virtuelles.