

# Rapport projet c++ : Elections, piège à cons ?

Jules Chassetuillier, Mohamed Legheraba

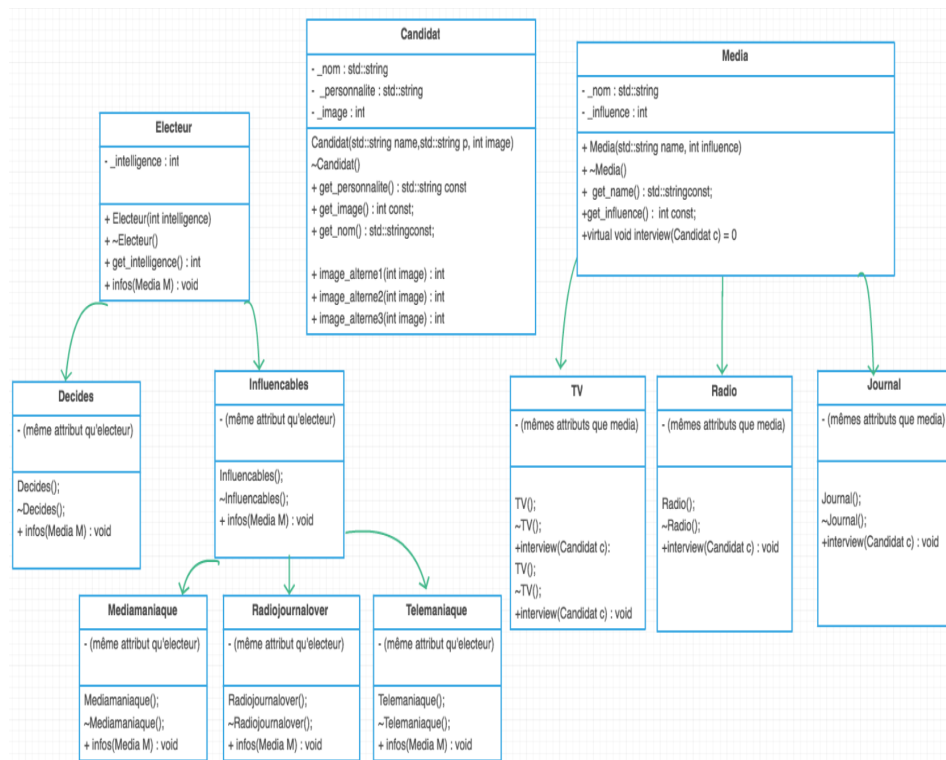
23 janvier 2017



## 1 Introduction

Nous avons décidé, pour ce projet, de mettre en oeuvre une simulation d'élections présidentielles, avec comme acteurs de ces élections les électeurs, un certain nombre de candidats, et des médias. Chacun de ces acteurs peut influencer de différentes manières sur les autres, et ainsi changer la donne pour l'issue de l'élection. Dans ce rapport, nous allons décrire les différentes parties de notre implémentation en c++, en présentant un diagramme UML représentant l'agencement de nos différentes classes, ainsi que certaines parties de notre implémentation.

## 2 Diagramme UML



Nous avons donc implémenté onze classes.

Nous trois classes de départ sont Electeur, Candidat et Media. Electeur a comme attribut une intelligence (entier compris entre 0 et 10).

Plus cet entier est grand, plus l'électeur est intelligent (logique), et cela détermine en partie son vote.

Candidat a comme attribut un nom, une image (entier comprise entre 0 et 10) et une personnalité (gentil, neutre ou salopard). Son image et sa personnalité jouent sur le nombre de votes qu'il recevra lors de l'élection.

Media a comme attributs un nom et une influence (entier compris entre 0 et 10). Les médias peuvent interviewer les candidats, et à l'issue de ces interview l'image du candidat change (peut être améliorée ou diminuer). Les médias diffusent également des informations qui modifient l'intelligence des électeurs. Plus

un électeur est con, plus il a de chances de voter pour un candidat salopard ayant une bonne image. Plus l'électeur est intelligent, plus il a de chances d'aller vers un candidat gentil, et ce indépendamment de son image. Si deux candidats sont gentils, il ira vers celui qui a la meilleure image, à condition que cette image soit issue d'une intervention d'un media de type journal.

Media a comme attribut une influence (entier compris entre 0 et 10)

\_Electeur est une classe virtuelle qui a 2 sous classes : Decide et Influençable  
Decide regroupe les électeurs non influençables par les médias

Influçable est une classe virtuelle qui comporte 3 sous classes : Mediamaniaque, Telemaniaque, et RadioJournallover

Mediamaniaque est un électeur influençable par les 3 types de médias existants

Telemaniaque est un électeur influençable uniquement par la TV

RadioJournallover est un électeur influençable par la radio et les journaux

\_Media est une classe virtuelle qui a 3 sous classes : TV, Radio, et Journal

### 3 La SDL

Après avoir fait notre Simulateur, on s'est dit que ça serait plus joli avec une interface graphique. Nous avons décidé d'utiliser la SDL 1.0 pour faire notre interface, notamment car elle est installée sur les machines de l'école. Nous utilisons également SDL/image pour charger des images autre que en .bmp et SDL/ttf pour afficher du texte à l'écran.

Notre affichage graphique est très rudimentaire. Nous avons un écran avec un titre, une image de fond et un bouton. Quand on appuie sur le bouton cela lance la fonction run du simulateur et on affiche le nom du Candidat qui a gagné l'élection avec le pourcentage de voix qu'il a obtenu.

Le bouton Run! ne marche pas tout de suite, il faut attendre quelques secondes.

### 4 Conclusion

Nous n'avons pas eu le temps de finir notre programme mais la base est là.

Il nous faudrait ajouter plus d'options pour les candidats et les médias.

De plus pour l'instant l'affichage graphique ne sert qu'à afficher le vainqueur et à appuyer sur le bouton run.

On aimerait que l'affichage serve à tout faire. Choisir le nombre d'Electeurs, leurs caractéristiques. De même pour les médias et les Candidats. On voudrait aussi ajouter des options sur le simulateur.

Nous n'avons pas vraiment eu le temps de faire cela car la bibliothèque SDL n'est pas simple à comprendre et car on a mis beaucoup de temps à déboguer la version basique du simulateur.