

# PROJET C++

## Sujet : "Elections piège à cons" ?

### Description

Notre projet est un jeu avec interface graphique développée sous Qt sous la forme d'un pacman mettant en scène un candidat à l'élection présidentielle devant récupérer des votes contre ses opposants, tout en évitant la prison.

Le programme présente :

- 12 classes
- 3 niveaux de hiérarchie
- 3 classes virtuelles (ComposantM, Objet et Personnage )
- 1 surcharge d'opérateur=
- UML : voir Annexe 1 ou fichier "class diagram.png" sur umbrello

### Installation nécessaire

Il faut installer la dernière version de Qt pour pouvoir compiler le programme.

Le makefile est fourni.

A noter que sous l'environnement Linux de l'école, il est possible de lancer directement l'exécutable fourni si l'installation n'est pas envisageable.

Une fois le programme exécuté, une fenêtre devrait s'ouvrir. La navigation s'y fait grâce aux flèches du clavier.

## Points positifs

A la première séance, nous avons plutôt bien déterminé les limites de notre projet. Nous ne nous sommes pas éloignés beaucoup du diagramme initial.

Dans l'ensemble, les classes sont cohérentes : leurs attributs et leurs méthodes sont bien réparties pour le fonctionnement du jeu.

Nous avons utilisé la bibliothèque Qt pour l'interface graphique. Cette bibliothèque s'est avérée facile à utiliser et le rendu est intéressant pour une présentation basique comme la nôtre. Cette base pourra être réutilisée pour d'autres applications en 2D.

## Points à améliorer

Pour le moment, les fantômes se déplacent de façon aléatoire. Une IA pourrait être mise en place pour rendre le comportement des fantômes autour du joueur plus intéressant.

L'utilisateur doit décider de chaque déplacement de son personnage. Un gameplay différent pourrait être implémenté pour plus de confort : le personnage s'avancerait dans la même direction tant qu'il ne rencontre pas un mur, ou tant que l'utilisateur n'a pas décidé d'une nouvelle direction.

Les fuites mémoires du programme sont dues à une compréhension limitée du fonctionnement de la bibliothèque Qt. Le programme est donc améliorable sur ce point.

Le labyrinthe est généré aléatoirement. Le cas des cases isolées (une case autre que obstacle entourée d'obstacles) n'est pas géré. Un algorithme de création de labyrinthe plus performant pourrait être implémenté.

## Annexe 1 : Diagramme de classe

