

POLYTECH - EISE4

Compte rendu – Projet C++

[Sous-titre du document]

Aminata DIAGNE & Tran Mai VILLAGEOIS

13/01/2017

SOMMAIRE

I.	Introduction.....	2
II.	Description du jeu.....	2
III.	Diagramme de classes.....	4
IV.	Difficultés rencontrées - Satisfactions	5
V.	Conclusion.....	5

I. Introduction

Le jeu que nous proposons se déroule dans un monde fictif « VIDI ».

Dans VIDI, chacun a son importance. Le vote d'un enfant est aussi important que celui d'un adulte. Par contre, certaines règles qui ne conviennent pas au citoyen sont établies. Et le citoyen peut changer cette situation s'il le souhaite.

II. Description du jeu

L'utilisateur incarne un citoyen et selon son âge et les choix qu'il fait, cela aura un impact sur les élections à venir. Il peut même incarner un candidat à ces élections. Mais à VIDI, rien ne se passe comme prévu et le candidat favori n'est pas forcément celui qui gagnera les élections. Par exemple, des règles privilégiant les anciens -car considérés comme plus compétents- sont établies. Seuls les enfants peuvent changer cette situation en faisant appel au Joker qui gonfle leur apport dans les élections.

Première étape : Description de l'utilisateur

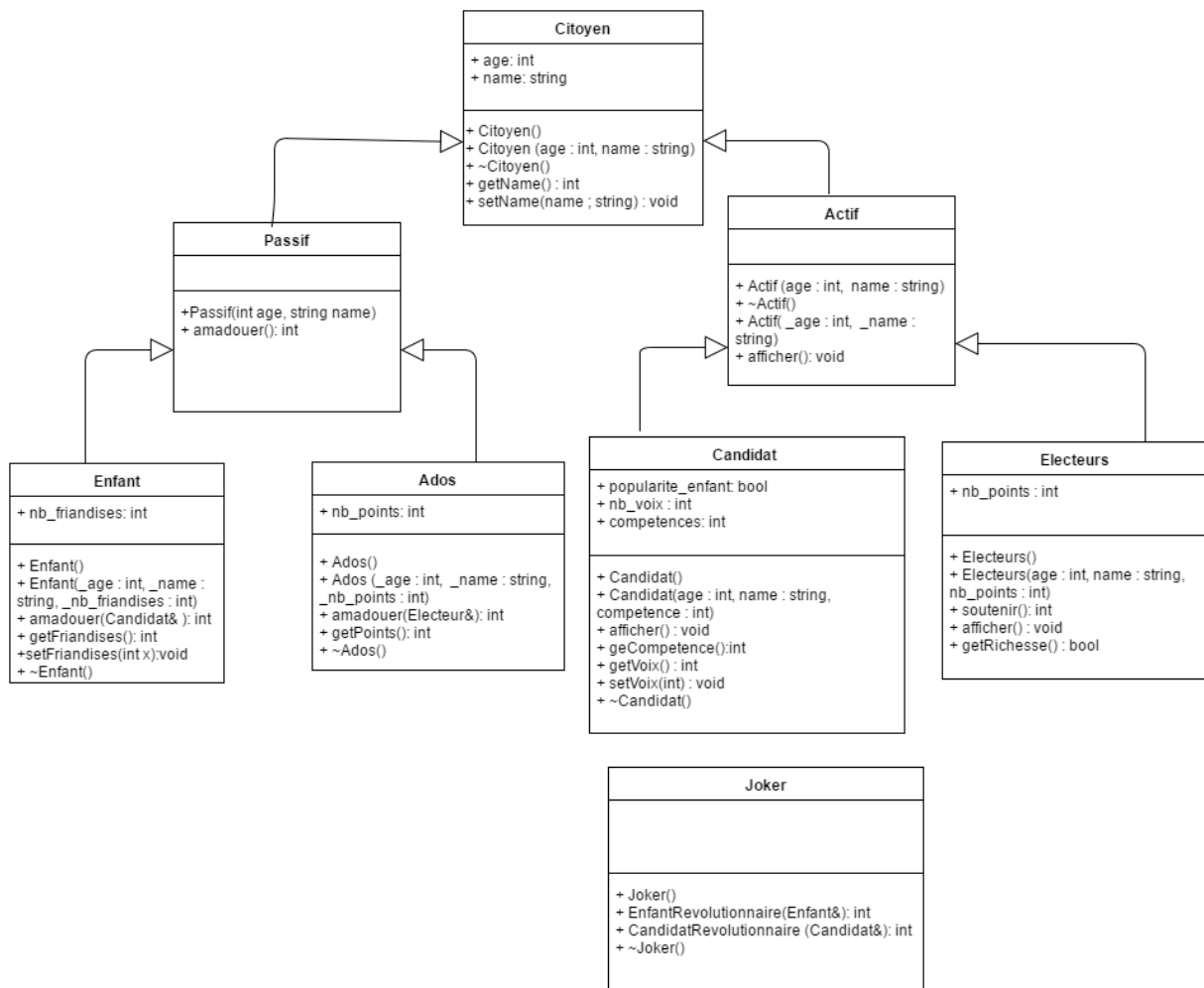
L'utilisateur est au début du jeu un citoyen et renseigne son âge ainsi que son nom.

Deuxième étape : Participation de l'utilisateur

- S'il a moins de 15 ans, c'est un enfant. Il ne peut pas voter, on lui propose alors deux candidats avec des caractéristiques différentes. Les enfants sont sensibles aux compétences des candidats : plus un candidat est compétent et plus un enfant veut le soutenir en l'amadonnant et lui donnant des friandises. Le nombre de friandises augmente le nombre de voix lors des résultats. Notons que la compétence d'un candidat dépend du nombre de fois qu'il s'est présenté. Et plus il est compétent, plus il a de chances de gagner si tout se déroule dans les normes. Sauf qu'un enfant peut faire appel au Joker qui l'aide à gonfler son score.
- S'il a entre 15 et 35 ans, c'est un adolescent. Il ne peut pas voter mais peut influencer sur le vote des électeurs. On lui propose quatre groupes d'électeurs. Selon son choix, il donnera des points à son groupe d'électeurs qui pourra alors donner plus de voix au candidat soutenu.
- S'il a plus de 35 ans, il peut être un candidat. Selon le nombre de fois où il s'est présenté, il aura plus ou moins d'ancienneté. Vont alors se dérouler les campagnes des enfants et des ados qui vont faire évoluer le nombre de voix.
- S'il a plus de 35 ans et qu'il ne veut pas être candidat, c'est un électeur. Et dans ce cas, il appartient à un groupe avec une idéologie différente et soutient le candidat de son choix avec l'aide des points récoltés auprès des ados qui eux, n'ont pas d'accès direct sur le vote.

III. Diagramme de classes

Trois niveaux de hiérarchie: Citoyen qui regroupe l'ensemble de la nation vidienne. Chaque citoyen a un âge ainsi qu'un nom. Et c'est l'âge qui détermine le niveau d'influence au cours des élections: les enfants et les ados par exemple d'un côté constituant le groupe des passifs et d'un autre côté, les actifs constitués de candidats et d'électeurs.



Pour plus de lisibilité, cliquez sur le lien :

https://www.draw.io/?state=%7B%22ids%22:%5B%220B23YfjoxWq_0cENNM3hUWWFLNzQ%22%5D,%22action%22:%22open%22,%22userId%22:%22112991542661408474183%22%7D#GOB23YfjoxWq_0cENNM3hUWWFLNzQ

IV. Difficultés rencontrées - Satisfactions au cours de la réalisation du projet

Plusieurs problèmes ont été rencontrés principalement les constructeurs et destructeurs, les fonctions amies, l'héritage.

On a beaucoup galéré pour résoudre certaines erreurs mais l'usage des constructeurs/destructeurs dans le cas d'un héritage furent un vrai fardeau. En effet, on a omis de mettre le constructeur par défaut des classes mères. Mais fort heureusement le problème fut résolu. Mais la plus grosse difficulté que nous ayons rencontrée concernait une méthode « amadouer » par exemple que nous avons considérée comme virtuelle. Et l'erreur que nous renvoyait le compilateur (*référence indéfinie vers vtable Actif*) ne nous a pas du tout aidée bien au contraire, elle nous a bien enfoncée. On se concentrait toujours sur les constructeurs/destructeurs en essayant de les redéfinir par diverses manières. Nous sommes restés longtemps dessus ne pouvant pas avancer. Mais en s'intéressant, on ne sait pas comment aux fonctions virtuelles, nous avons pu résoudre notre erreur pour notre plus grande joie. Et ceci nous a également permis de mieux comprendre le mot clé « virtual » mais hélas nous fûmes rattrapés par le temps. Nous avons pu résoudre l'erreur qui nous empêchait d'avancer mais nous n'avons pas pu utiliser une fonction virtuelle ou abstraite en Java. Mais par exemple, dans notre code, si on instanciat un objet Actif(en référence), et qu'on a besoin d'appeler une méthode afficher() par exemple qui a été redéfinie dans chaque classe fille, il va falloir dans ce cas déclarer cette méthode amadouer en virtuelle pure (car non définie dans la classe Actif) pour permettre au compilateur de comprendre quelle méthode de quelle classe utiliser

V. Conclusion

La plus grande difficulté du projet fut de trouver une idée qui respectait les conditions imposées c'est-à-dire le nombre de classes et le niveau de hiérarchie ainsi que la présence d'interactions avec l'utilisateur qui rend le jeu interactif et intéressant à coder en C++.

Il nous aurait fallu davantage de temps pour pouvoir avoir une interface graphique et avoir ainsi un jeu réellement abouti.