

Projet C++

-

Man VS Undead
2020-2021

BENNIS Ismaïl
TRAN Alexia

Mme. BRAUNSTEIN

Sommaire

Introduction	3
I - Description de l'application développée	3
II - Diagramme UML de l'application	4
III - Procédure d'installation et d'exécution du code	4
IV - Les parties de l'implémentation dont vous êtes les plus fières.	5
V - Piste d'amélioration	5
Conclusion	5

Introduction

Lors de ce projet, nous avons pour objectif de faire de la programmation orientée objet en codant en C++. Nous devons créer 8 classes hiérarchisées sur 3 niveaux. Le thème imposé est “Le monde d’après”.

I - Description de l’application développée.

Pour répondre aux contraintes, nous nous plaçons dans un monde post-apocalyptique où une mystérieuse maladie a fait des ravages. Celle-ci ronge le cerveau des humains jusqu’à l’apparition d’un comportement animal. Dans ce monde, le seul divertissement disponible fait penser aux combats de gladiateurs d’antan. En effet, le joueur est placé dans une arène et doit abattre tous les zombies qui s’approchent de lui.

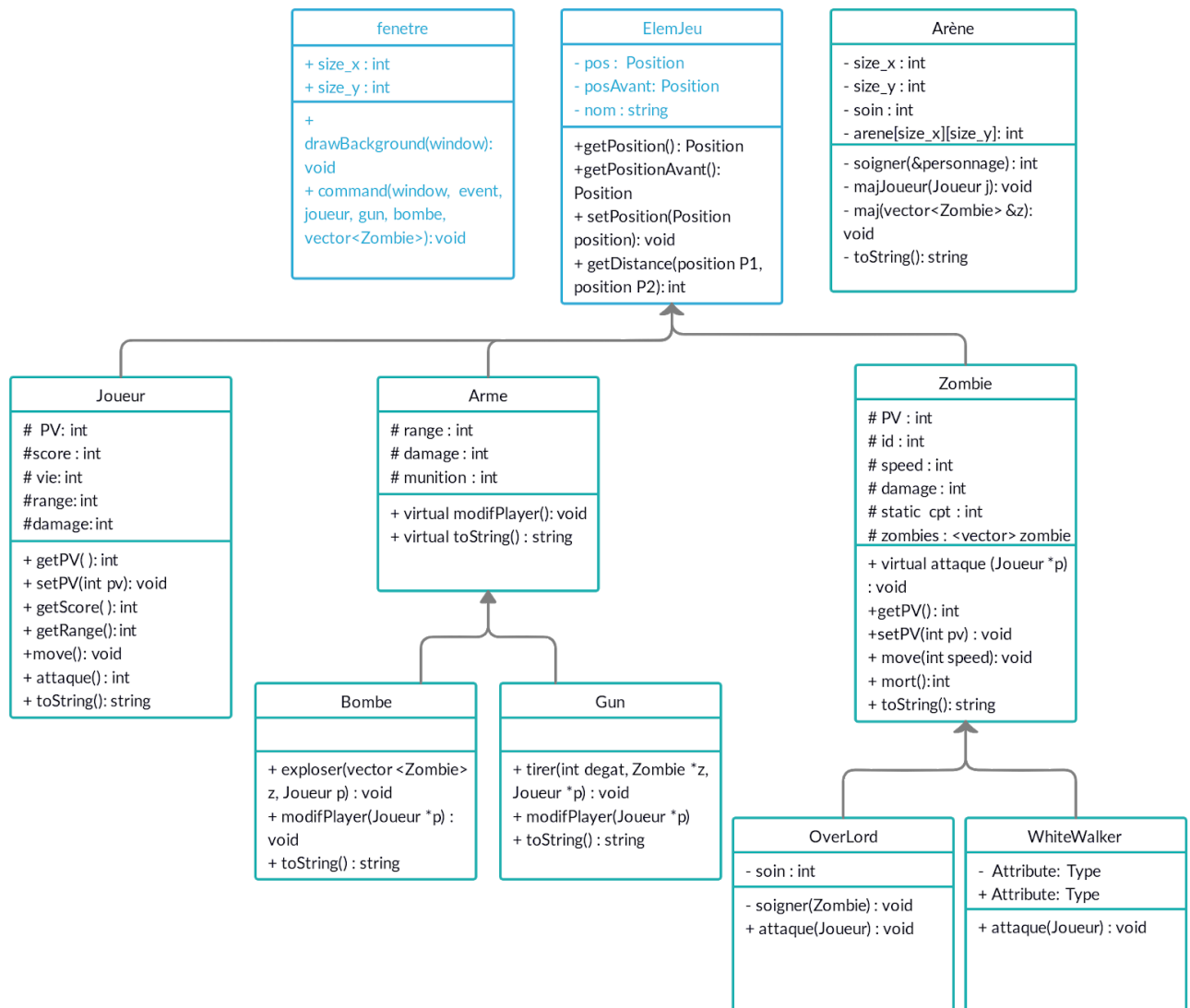
Au démarrage, le joueur possède une arme. Il spawn sur la map avec plusieurs zombies. Le joueur doit utiliser les touches Z, Q, S, D pour se déplacer dans les 4 directions. Les zombies sont placés de manière aléatoire, et puisqu’ils n’ont plus les capacités intellectuelles pour réfléchir, ils se déplacent aussi de manière aléatoire.

Une fois à portée d’un zombie il peut tirer en appuyant sur la touche N. Lorsque le joueur meurt bientôt et qu’il est désespéré, il peut placer une bombe à ses pieds. Cela détruit tous les zombies à ses alentours mais fait également perdre des points de vie au joueur. Il existe une variété de zombie pouvant soigner ses semblables. Les affichages de ces zombies sont les mêmes, afin que le joueur ne sache pas qui est en train de soigner qui.

Nous n’avons malheureusement pas pu implémenter toutes les fonctions désirées.

II - Diagramme UML de l'application.

Voici le diagramme des classes que nous avons codé :



III - Procédure d'installation et d'exécution du code.

Faites un git clone avec l'adresse :

https://github.com/Segfault9/man_vs_undead.git

À présent vous pouvez compiler le jeu grâce au Makefile (\$ make).

Pour l'exécuter : ./jeu

Nous avons commencé une interface graphique que nous n'avons pas pu finaliser.

Pour voir le debut de l'interface il vous faut tout de même la librairie :

SFML

Pour chercher sa présence sur votre machine, ouvrez un terminal et entrez la ligne suivant:

```
$ dpkg -l | grep sfml
```

Si rien ne s'affiche, entrez la ligne :

```
$ sudo apt-get install libsFML-dev
```

IV - Les parties de l'implémentation dont vous êtes les plus fières.

Toutes les parties de l'implémentation nous rendent fiers. En effet, après de nombreuses heures de débogage sur nos fichiers, chaque objet terminé participe à notre sentiment d'accomplissement. Malgré un code qui ne fonctionne que partiellement, nous en revenons de loin comparé à nos capacités au début de ce projet.

V - Piste d'amélioration

En piste d'amélioration, au niveau du code, nous pouvons mieux factoriser les fonctions dans les classes. En effet, nous avons remarqué que certaines fonctions auraient pu remonter dans la classe mère. Ainsi, peut-être, faire ressortir des fonctions virtuelles.

Certaines classes et fonctions sont incomplètes, notamment la fonction move du joueur, qui aurait dû permettre de tirer et lâcher une bombe. Il manque aussi un élément primordial, l'affichage des points de vie.

Au niveau de l'implémentation graphique, il aurait fallu la terminer, c'est-à-dire placer le joueur et les zombies et pouvoir les déplacer dans la fenêtre.

Conclusion

Pour conclure, nous avons produit un programme qui compile malgré ses défauts. Il y a bien sûr toujours une grande marge de possibles améliorations mais nous avons pu progresser et comprendre des notions de programmation orientée objet.