POLYTECH PARIS-UPMC

UPMC SORBONNE UNIVERSITÉS

Projet C++

Enseignant : Cecile BRAUNSTEIN, Roselyne CHOTIN AVOT

MOUAOUED Kerane
CHERFOUH Amine
EISE4

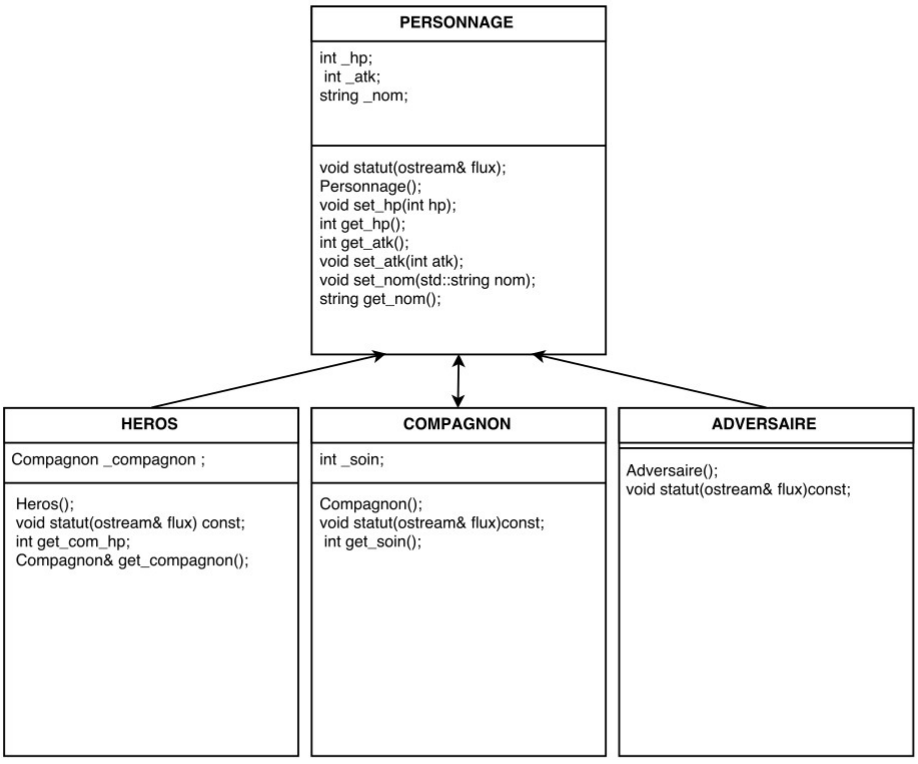
Introduction :

Dans le cadre du module langage c++, nous devions réaliser un projet avec pour thème :
« Élections, piège à cons ».
Pour cela nous avons réfléchi à l'idée de créer un jeu qui à travers différents défis désigneront un Roi. Cependant le jeu est truffé de piège à vous de vous méfier.

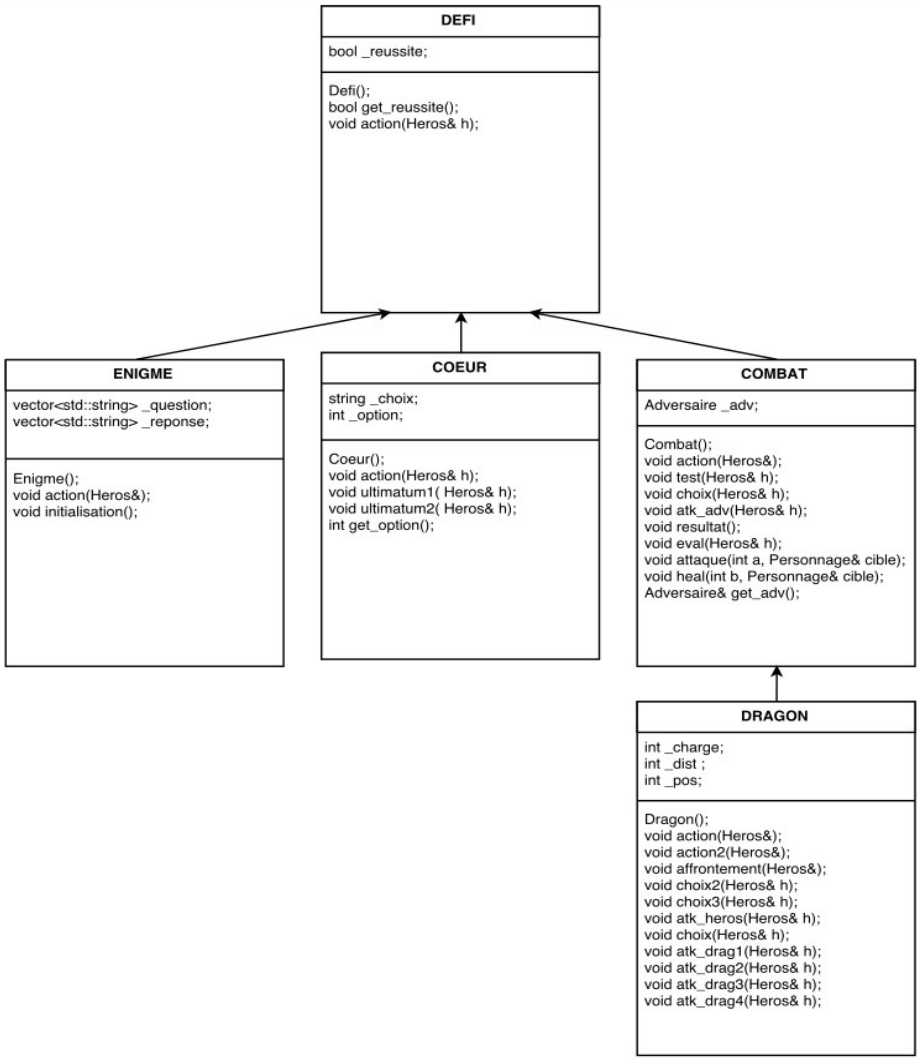
Descriptions de l'application développée :

Nous avons imaginé un scénario se déroulant au moyen âge, afin d'élire un nouveau Roi.
Le jeu consiste à participer à différents défis dans un ordre aléatoire et donc différent à chaque partie :
Un défi d'intelligence
Un défi de cœur
Un défi de force
Et le combat final contre le dragon Smaug.
Pour surmonter ces épreuves vous aurez avec vous un compagnon capable de soigner, à vous d'arriver au bout du donjon sans périr.

Diagramme UML :



Les personnages du jeu possèdent un champ nom, un champ points de vie et un champ point d'attaque. Le héros possède un champ compagnon, un personnage à part entière qui a en plus un champ points de soin. Nous avons surchargé l'opérateur « << » afin d'afficher les informations des personnages à l'écran. Nous n'affichons pas les points d'attaque des adversaires à l'écran, le joueur doit s'y adapter en jouant.



Les défis arrivent aléatoirement dans le jeu mais vous ne pouvez réaliser qu'une seule fois chaque défi par partie. Lors des défis vos choix seront primordiaux pour la suite du jeu. Et à travers vos différents choix chaque partie sera différente des précédentes. Une fois ces différents défis réalisé il vous faudra affronter un défi final afin de mériter le titre de Roi de Bretagne. Ce défi consiste à affronter un dragon, c'est un combat très difficile.

1) Défi d'intelligence

Lors de ce défi vous allez rencontrer un magicien : Merlin l'enchanteur, celui-ci vous posera une question tirée aléatoirement parmi trois énigmes. Pour répondre à cette question vous aurez trois chances, il vous faudra écrire en minuscule, au singulier et sans déterminant. Si vous réussissez ce défi vous obtiendrez une potion qui augmentera vos points de vie de 20 ainsi que ceux de votre compagnon s'il est toujours en vie.

2) Défi du cœur

Lors de ce défi vous allez rencontrer la fée Morgane, celle-ci vous posera un ultimatum qui sera différent selon votre avancement dans le jeu. Cet ultimatum sera déterminant pour la suite du jeu. En effet les ultimatums qui vous seront posés vous ferons gagner ou non l'arme Excalibur. Cette arme permet d'augmenter sensiblement vos points d'attaque pour la suite de l'aventure et en plus permet d'attaquer le dragon à distance en plus lors du défi final. Cependant je vous rappelle que tous vos choix entraine une issue différente.

3) Défi de force

Lors de ce défi vous devrez combattre un adversaire, c'est un combat tour par tour. Le combat ne sera pas le même à chaque partie, en effet cela dépendra encore une fois de vos choix effectués lors des autres défis. Pour combattre vous aurez à chaque tour un choix à faire : Attaquer ou soigner grâce à votre compagnon, s'il a péri vous attaquez de manière automatique. L'adversaire lui attaque

aléatoirement le joueur ou son compagnon pour infliger 15 dégâts. Si vous remportez le combat vous pourrez ramasser sur le cadavre de votre adversaire une potion qui réinitialise vos points de vie ainsi que ceux de votre compagnon, s'il est en vie, au maximum.

4) Défi final : combat du dragon

Lors de ce défi final vous aurez à combattre le dragon Smaug. C'est un combat au tour par tour mais qui sera plus difficile que le défi de force. Pour combattre vous aurez à chaque tour un choix à faire : Attaquer, soigner ou esquiver. Les règles sont expliquées dans le jeu avec le texte suivant : « Smaug est un majestueux dragon il peut vous porter un coup de griffe au corps-à-corps, s'envoler dans le ciel ou cracher une terrible fournaise peu importe la distance. Cependant le cas échéant il devra recharger son souffle. Seule la magie d'Excalibur pourrait l'atteindre lorsqu'il est dans le ciel. » . Contrairement au défi de force, l'adversaire effectue plusieurs actions de manière aléatoire : S'il est au sol, à une distance de 0, il peut s'envoler à une distance de 1, attaquer à coup de griffes pour infliger 20 dégâts, cracher du feu pour infliger 30 dégâts. Dans le ciel il peut redescendre ou cracher du feu. Enfin s'il a craché du feu au tour précédent, il ne peut le refaire tant qu'il n'a pas rechargé ses flammes. Le joueur en face en plus d'attaquer ou de se soigner peut décider d'anticiper un coup de griffe ou un souffle : il faudra être stratégique et chanceux.

Avec les différents choix possibles à effectuer durant le jeu, chaque fin de partie sera différente. Afin de respecter le sujet « Élections, piège à cons », réussir à être élu roi est très difficile voire impossible (il faut être extrêmement chanceux), le jeu laisse néanmoins l'illusion que cela est possible. Nous vous laissons jouer pour découvrir pourquoi.

Procédure d'exécution du code :

Pour exécuter le code il faut :

- Utiliser la commande « make » afin de compiler les différents programmes
- Utiliser la commande « make test » afin d'exécuter le code et pouvoir jouer.
- Utiliser la commande « make clean » permet de supprimer tous les fichiers objets.
- Utiliser la commande « make valgrind » permet de voir les erreurs de mémoire.

Le fichier makefile permet d'utiliser toute ces commandes.

-Certains fichiers textes contenant les textes de l'histoire sont joints aux fichiers de code et sont nécessaires au bon déroulement du jeu.

Difficultés rencontrées :

- Nous avons comme contrainte que la taille d'une méthode ne devait pas excéder 30 lignes, cela nous a posé quelques problèmes notamment dans la méthode combat ainsi que la méthode atk_adv. Car nous avions pensé à différentes attaques mais qui nous donnaient énormément de cas à traiter nous avons donc créé plusieurs méthodes (atk_drag1, atk_drag2, etc..) afin de réduire la taille de nos méthodes.

- Nous avons éprouvé des difficultés à rendre le combat final plus interactif. Car lorsque lancelot mourrait, notre personnage ne pouvait plus qu' attaquer Smaug automatiquement comme lors du défi de force. Du coup le combat se déroulait sans qu'on puisse y faire grand-chose, et voir du texte défiler sans interagir n'est pas pertinent pour un jeu. Nous avons donc rajouté 2 choix : esquive de l'attaque griffe acéré de Smaug et parade du souffle de Smaug.

- Nous avons également éprouvé des difficultés à la finalisation du jeu, lorsque l'on devait faire interagir les différentes classes. En effet pour lancer les défis de manières aléatoires à chaque partie, nous les avons mis dans une map de Defi avec une clé int. Cependant il est impossible de mette une classe virtuelle dans un conteneur, cela nous a créés d'innombrables erreurs et nous a fait perdre un temps fou. Il fallait finalement mettre un pointeur sur classe Defi pour que cela compile et que l'on puisse profiter de l'héritage de la classe Defi.

- Les saisies au clavier sur le flux d'entrée à également poser des soucis. En effet au départ nous utilisions cin, dans des strings pour les noms ou les chaînes de caractères, dans des chars pour les choix simples. Cela provoquait des bugs lorsque l'utilisateur saisissait en entrée des espaces : les caractères après l'espace restaient dans le buffer d'entrée et les saisies suivantes prenaient donc en compte ces caractères, ce qui altérait l'expérience de jeu. Nous avons remédié à cela en utilisant la fonction geline(cinq, string), qui saisit toute la ligne entrée. Nous avons donc également changé les chars d'entrée en string.
  Il reste cependant toujours le \n dans le buffer après une geline() ce qui met des passages à la ligne intempestive, mais c'est toujours mieux que de fausser le jeu.