Projet transverse

TLWL

Ahmed Sabry – Thibault Courtois - Boris Lours-Riou – Adrien Dat - Hugo Cordier

# Table des matières

[Table des matières 1](#_Toc514959107)

[Univers 2](#_Toc514959108)

[La base de données et les statistiques 2](#_Toc514959109)

[Fonctionnement du combat et intégration de l’IA 3](#_Toc514959110)

[Interface Programme 3](#_Toc514959111)

[Python et Scikit learn 4](#_Toc514959112)

# Univers

A la base notre projet était un jeu de rôles complet. Et donc nous avons commencé par réfléchir à l’univers dans lequel nous voulions développer l’histoire et le gameplay. Nous avons choisi l’héroïque fantaisie qui est un genre très populaire dans ce genre de jeu, les jeux de rôle sur papier sont notre modèle de base. Donc nous pourrons trouver des personnages tels que des magiciens, des monstres et des dragons, cela fait partie de l’essence du jeu.

Hélas nous avons décidé de changer notre projet pour que celui-ci colle plus avec le thème des IA. En effet l’implémentation d’une IA pour raconter une histoire nous a parus être une trop peu pertinente pour être poursuivie. Nous avons alors choisi de nous concentrer sur l’un des noyaux du JDR, le combat. Notre combat implique une équipe de trois héros qui se retrouve face à un monstre. Chaque héros possède une classe différente pour varier les plaisirs et auront des paternes d’attaques pré-enregistrer pour simuler les choix d’un vrai joueur. Cependant notre monstre qui fera office de boss sera contrôler par l’IA, et cette IA devra être capable se défendre jusqu’à être capable de tuer les trois héros. Notre combat s’achèvera lors de la mort du boss ou de la mort de toute l’équipe de héros.

# La base de données et les statistiques

Comme nous l’avons dit tout à l’heure chaque héros possède une classe différente, cela n’est pas qu’une question de cosmétique parce que chacun de ces héros aura des capacités différentes en fonction la fonction que nous avons choisie de donner à chaque classe.

* Un Guerrier qui possède une très grande robustesse et beaucoup de points de vie qui sera capable d’encaisser les attaques du boss sans broncher.
* Un Sorcier qui possède une faible résistance mais possède une attaque correcte et donne la capacité à l’équipe d’utiliser des sorts de soin.
* Un Assassin qui ne possède pas une grande résistance mais qui devrait délivrer des coups puissants.
* Le Boss qui possède un très grand nombre de points de vie ce qui donnera du fil à retordre à notre équipe de héros.

Maintenant les classes et leur fonction décider nous allons devoir leur donner des statistiques. A la manière d’une véritable fiche de personnage dans un jeu de rôles standard. Nous avons choisi de décrire nos personnages dans un grand panel de statistique qui seront toutes utiles lors du combat.

* Une image contenant capture d’écran

  Description générée avec un niveau de confiance très élevéLes points de vies
* L’armure
* Le taux de chance de coup critique
* Le taux de chance de blocage
* La statique d’attaque (pour calculer les dégâts)
* Les attaques (plusieurs par personnages)
  + Dégât propre de l’attaque utilisée
  + Le temps de chargement entre chaque utilisation de l’attaque (cool down)

Cette fiche de statistique est stockée dans notre programme en tant que classes qui décrit le personnage qui lui est attribué.

# Fonctionnement du combat et intégration de l’IA

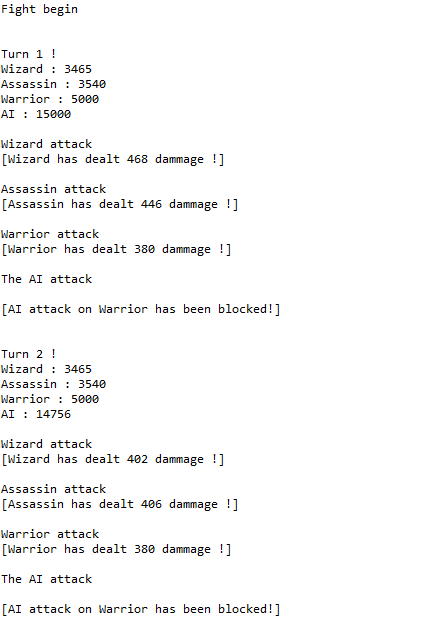
Nos combats fonctionneront de la manière suivante : Chaque joueur sera appelé à attaquer chacun son tour afin de garder et à ce moment un algorithme spécial se chargera de déterminer les dégâts de l’attaque, sa précision et la chance que l’ennemi a de bloquer cette attaque. Tout cet algorithme met en relation les statiques rentrées au préalable par notre équipe afin de rendre le combat le plus équilibré possible. Cet algorithme teste donc des conditions pour chaque personnage lors de son tour qu’il n’est pas mort ou bien qu’il n’attaque pas alors que ses points de vie sont déjà inférieurs à 0 et qu’il est supposé mort.

Il est nécessaire de préciser (ou de repréciser) que nos 3 héros ont un paterne d’attaque préenregistrer afin de simplifier l’intégration de l’IA au programme.

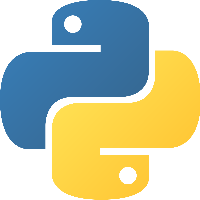
L’IA justement, notre boss est donc une machine de réflexion et d’analyse afin de déduire la meilleure attaque à faire et sur quelle cible. L’IA fonctionne de la façon suivante : le boss tel un enfant attaque de façon plus ou moins aléatoire au début puis au fur et à mesure que le combat se poursuit il va être capable de dire que certaines attaques sont plus efficaces sur certains héros plutôt que d’autres et décidera donc de choisir l’attaque en fonction de sa cible. Tout en gardant en mémoire les dégâts moyens de chaque attaque en fonction des coups critiques et des colons, il va pouvoir dessiner des actions types efficaces. L’IA conserve les données qu’elle a apprises dans un fichier que nous allons utiliser pour les interpréter et dessiner des graphes qui seront significatifs des attaques utilisées par l’IA. De ces graphes l’IA va pouvoir déduire les meilleurs choix d’attaques et donc devenir de plus en plus difficile à battre.

Notre IA est conçue de telle façon à ce qu’elle soit imbattable par nos héros actuels mais que lors d’un vrai combat elle soit un adversaire redoutable.

# Interface Programme

Notre IU est des plus simplifiée. En effet étant inspiré des JDR papiers, notre projet ne délivre à l’utilisateur-joueur que les conséquences de chacune des actions des héros et du boss par tour ainsi que les cibles choisies accompagnés des points de vie de chaque personnage encore dans le jeu. On recommence l’opération à chaque tour jusqu’à la fin du combat.

# Python et Scikit learn

Le choix du langage dans ce projet informatique est très important. Nous avons décidé d’utiliser le Python qui est très répandu dans l’analyse de données et le machine learning, tout ce dont nous avons besoin. Le python est un langage de haut niveau et permet des simplifications dans la syntaxe comparée au langage c et c++, notamment la suppression du « ; » à la fin de chaque instruction qui soit dit en passant peu parfois nous jouer des tours dans l’écriture du code.

Cependant la raison principale de notre migration sur python est la présence de la bibliothèque « Scikit learn ». La raison de la présence de cette bibliothèque est qu’il s’agit d’un formidable outil de gestion de données en ce qui concerne le machine learning. Cette bibliothèque met à disposition tout un tas de fonctions qui nous permettent de manipuler des données afin de tirer des conclusions comme nous le faisons pour notre projet.

# Conclusion

(en préparation par thibault (normalement))