

Projet de programmation objet avancée

Aurélien Terrain et Julien Guyot

4 Mai 2019

1 Introduction

Ce projet porte sur la réalisation à partir d'une base d'un jeu de type «kill them all» : Le but est de tuer un ensemble de «gardiens» pour accéder à un trésor.

Durant ce rapport, nous décrirons brièvement la structure du code ainsi que les choix faits, avant de conclure.

Par ailleurs, nous avons généré une documentation basique (voire rudimentaire) via Doxygen, décrivant toutes les fonctions et tous les attributs rajoutés aux diverses classes, et qui est disponible dans les répertoires `docs/html/index.html` et `docs/latex/refman.pdf`

2 Structure du code

Le code que nous avons produit est disponible dans les trois classes suivantes, comportant chacune un fichier `.cc` et `.h`.

2.1 Labyrinthe

Le principal choix fait a été de modifier la structure du tableau `_data[] []` : chaque case peut contenir une valeur comprise entre 0 (`EMPTY`) et 5 (`TREASURE`), indiquant sur la nature de ce qui occupe la case.

Une fonctionnalité que nous avons implémenté a été de permettre que les boîtes soient destructibles pour permettre au joueur de régénérer sa santé.

2.2 Gardien

Les gardiens sont dotés d'un certain nombre de points de vie au début du jeu, et perdent leur précision quand ils se font toucher par une boule de feu. On remarque aussi qu'ils possèdent une perte de précision de base, contrairement au chasseur.

Leur décision est faite en fonction d'un potentiel de protection, calculé par rapport à des paramètres, plus un quota aléatoire. Le garde ne fait pas de décision à chaque appel de `update()`, mais prend une décision, et la suit pendant un temps aléatoire (mais borné).

Les décisions peuvent être d'attaquer (auquel cas le garde se dirige vers le joueur), de défendre (aller vers le trésor) ou explorer (exploration au hasard dans le labyrinthe).

Nous pouvons aussi remarquer que la fonction de mouvement utilisée par les gardien est relativement rustique, et conduit souvent à un amoncellement de gardiens sur une case donnée.

2.3 Chasseur

Le chasseur dispose d'un plus grand nombre de points de vie que les gardiens, et perd moins de précision.

3 Conclusion

Nous avons implémenté un jeu fonctionnel à partir du code donné en cours.