

UNIVERSITÉ LIBRE DE BRUXELLES



ECOLE
POLYTECHNIQUE
DE BRUXELLES

COMPLÉMENTS D'ALGORITHMIQUE ET DE PROGRAMATION
INFO-H304

Rapport de projet - ArnianLeTriCozor

Auteurs :

Andrew **Karam**
Nicolas **Lambilliotte**
Arthur **Vanderbreetstraeten**

Professeurs :

Jérémie **Roland**
Robert **Vanden Eynde**

16 Décembre 2018

1 Partie programmation

La base du programme est faite de quatre fichiers.

Dans la classe *gameobject* sont définis l'ensemble des objets du jeu, unités comme terrains. Nous avons considéré les bâtiments comme des terrains, permettant d'y appliquer les mêmes fonctions (*isAccessible*, *getDefense*,...)

La classe *map* se charge de la construction de la map et de la définition de chaque *mapObject*. *Game* est une classe qui reprend toutes les actions et modalités intervenant dans le jeu. L'âme du gameplay se trouve dans cette dernière classe.

Enfin, la fonction *main* fait d'abord appel à une fenêtre d'introduction, où on peut choisir le type de partie que l'on souhaite jouer. Cette dernière va appeler la classe *Mainwindow*, dans laquelle le jeu se passe.

2 Partie algorithmique : IA

L'intelligence artificielle que nous avons conçue se base sur un algorithme glouton. Celle-ci va dans chaque situation parcourir l'ensemble des possibilités.

Notre IA se comporte de la manière suivante :

L'IA parcourt les cases accessibles par chacune de ses unités, elle décide de la ville qui est la plus optimale pour se déplacer dans sa direction. Une fois qu'une unité décide de capturer une ville, celle-ci n'est plus accessible pour les autres unités, elles cherchent donc une autre ville proche capturable. Son objectif est à la fois de capturer l'ensemble des bâtiments et d'éliminer les ennemis. L'IA se comportera différemment en fonction des actions du joueur adverse.

Par défaut, elle va créer des unités infanterie ou bazooka. Ces dernières vont à chaque tour aller capturer la ville la plus proche. Si l'adversaire crée des unités, l'IA produira un autre type d'unité, qui aura pour objectif d'aller attaquer l'adversaire le plus proche. Si plusieurs d'entre eux sont accessibles, elle n'attaquera pas forcément le plus proche mais décidera d'attaquer celui qu'elle endommagera le plus.

Ainsi, notre IA combine la capture de bâtiments avec l'élimination des ennemis.

3 Interface graphique

Le design de l'interface graphique a principalement été programmé dans la classe *mainwindow*. En dimensionnant et en plaçant correctement la map et la *mainwindow*, nous avons créé de l'espace afin de pouvoir ajouter des informations et boutons nécessaires à l'amélioration du gameplay.

Ces ajouts sont de différents types :

- **Labels** : Ce sont de simples éléments de texte ajoutés permanents ou non dans la *Mainwindow*. Exemple, *Orange star*, *Blue Moon*, ...
- **Simple boutons** : Permettent d'ouvrir une fenêtre donnant des indications (*Show commands*, *show Rules*, ...)
- **Boutons reliés au gameplay** : Ces boutons induisent un changement dans le jeu (*Change Turn*)

De plus, lorsqu'on effectue un click droit sur une case de la map, les différentes informations de cette dernière apparaissent à l'écran (si ces dernières sont disponibles).