LARRIBE Maëlys 26/01/2021

**Rapport de Projet de C++**

1. **Implémentation Projet**
2. **Diagramme UML**
3. **Manuel d’utilisation**

Tel qu’implémenté, « q » est toujours disponible pour quitter le programme.

Au démarrage il est demandé à l’utilisateur s’il souhaite jouer en multijoueur ou contre l’ordinateur. Pour choisir il doit taper « m » ou « s » respectivement.

Suite à cela, l’utilisateur à l’opportunité de charger une sauvegarde s’il le souhaite (oui : « y », non : « n »).

Une fois la partie lancée, chaque joueur est libre de placer une unité selon son argent disponible (fantassin : « f », archer : « a », catapulte : « c »).

Il a également l’opportunité de passer son tour (« p »).

Une fois la partie terminé le joueur a l’occasion de là sauvegarder s’il l’a interrompu.

1. **Difficultés et améliorations**

De nombreuses difficultés se sont présentés lors de ce projet.

**Switch case impossible :**

Pour une raison que je n’ai pas réussi à déterminé, je me suis retrouvée dans l’incapacité d’effectuer un switch case.

**Parcours du Terrain à revoir :**

Les techniques de parcours du Terrain sont réellement à revoir. Bien que j’aie tenté au maximum d’y éviter la duplication de code, j’aurais vraiment apprécié pouvoir passer plus de temps sur ce projet pour trouver une meilleure méthode (peut-être à base de fonction anonyme).

**Sur vérification :**

Certaines vérifications de pointeur non nul sont effectués à plusieurs endroits dans mon programme, elles ne sont pas nécessairement toutes utiles même si elles protègent le code.

**Historique Terrain :**

J’aurais aimé implémenter une système d’historique dans le Terrain qui stock les actions (achat) effectuer à un instant t. cela aiderais dans l’isolation de bug.

**Super-soldat :**

Pour une adaptabilité plus simple du code, j’aurais préféré prendre le temps de faire une classe à part pour super-soldat.

**Fuites de mémoires :**

Il y a de grande chance pour qu’il y ait des fuites de mémoires.