**RAPPORT PROJET PROGRAMMATION ORIENTEE OBJET**

**MORESTEL THOMAS**

**ROUSSON GUILLAUME**

**GOURGAUD KEVIN**

Introduction (but du projet) – Plan du rapport

Analyse du projet avec diagramme UML commenté

Technique utilisée (héritage, encapsulation, polymorphisme, exception, …)

Présentation synthétique (fonctionnalité interface)

Estimation temps consacré au projet / étudiant & organisation de travail (répartition des tâches)

Ressources utilisées (outil, livre, tuto)

Conclusion :bilan critique du travail (point fort, point faible) + impression sur aspect de la POO qui ont intérêssé(ou non), qu’on a trouvé difficile.

Introduction :

Le but du projet de l’unité d’enseignement programmation orientée objet était de créer de toutes pièces un wargame dans le langage JAVA avec certaines contraintes parmi lesquelles il devait y avoir des obstacles jonchant la carte, deux armées bien distincts : Les héros et les monstres ainsi que la présence de plusieurs fonctionnalités telles que : La sauvegarde, la restauration de partie, le redémarrage d’une nouvelle partie et un double accès à ces fonctionnalités. D’autres fonctionnalités non-obligatoire étaient suggérées, nous en avons implémenté une bonne partie.

Ressource :

Les textures de types herbes / mer / arbres / rochers sont tous issus des jeux Pokemon. Parmi les musiques que l’on a décidé d’intégrer au jeu nous avons : « Le Départ du Roi » de la Cinescenie ainsi qu’une autre musique trouvée dans une compilation Youtube qui répertoriait beaucoup de musiques style taverne / marché médiéval. Concernant les avatars, ils proviennent tous de ce lien : <https://craftpix.net/freebies/free-fantasy-character-avatar-icons-pixel-art/>

Les couleurs utilisées sont quant à elles issues de l’outil Color Picker de Google.

Les fonctions de sauvegarde et restauration ont été inspirées des nombreux tutoriels existant sur Internet. Nonobstant, tout ces tutoriels utilisaient un simple try tandis que notre environnement de travail nous signalait un « Code Smell » de type blocker (Qui est l’état au-dessus de critical, c’est dire) Celui-ci a été résolu en utilisant un try-with-resource.