Rapport Modélisation Orientée Objet et UML Projet de Modélisation : A la découverte du Trésor Perdu

Guilmain Loïc Lafage Benjamin Cianamea Mickaël Dulcamara Thomas
CERI
Université d'Avignon, Avignon

8 janvier 2020

1. Abstract

L'objectif de ce projet est de mettre en place une application informatique au moyen d'approches issues du génie logiciel, et en particulier au travers d'une modélisation orientée objet. Vous devrez ainsi définir un planning de travail, avec une estimation de l'organisation de votre temps. Puis vous devrez travailler à la réalisation de différents diagrammes UML (listés dans la suite de ce document). Enfin, vous proposerez une implémentation de votre diagramme de classe permettant de réaliser le logiciel qui vous aura été demandé.

2. Composition de l'équipe

- Guilmain Loïc ⇒ Scrum master
- Lafage Benjamin, Cianamea Mickaël, Dulcamara Thomas⇒ Equipe de développement

3. Modelisation

Lors de la séance du 18/11/2019, nous avons décidé de prendre JAVA comme langage de programmation.

De plus nous avons établi le planning voir l'annexe Planning. Suite a cela nous nous sommes mis sur l'élaboration du diagramme de cas d'utilisation. Durant les semaines suivantes l'équipe a défini les diagramme états-transitions et le diagramme de classe en rapport avec le diagramme de communication.

Les differents diagramme sont accessible en annexe de ce rapport.

4. Conception

4.1. Mise en place

Durant la phase de modelisation plusieur probleme ce sont posés comme le fait que le code sois reutilisable, evolutif ou bien juste maintenable. En effet, le principe des méthodes agiles est de pouvoir modéliser un programme pouvant etre evolutif et reutilisable.

Pour cela nous avons utiliser les cour de genie logiciel et de design pattern, ainsi nous avons mis en place plusieur solution permettant de respecté les principes vu ci-dessus.

Premierement,le premier probleme a était la génération du plateau qui est un tableau de case aléatoire suivant quelque règle (pourcentage de type de case,nombre d'item fixé,etc),nous avons trouvé comme solution de créer une classe Generator avec comme pattern une stratégie car cela nous permettra de pouvoir facilement genérer le plateau ou de changer les regles de generation sans changer la structure meme du plateau.

Deuxièmement,nous nous sommes confronté au probleme de déplacement des differents personnages sur le plateau, il nous fallait ainsi plusieur fonction recherchant et verifant si la future case permettait et accepté le déplacement nous avons donc utilisé la meme méthode que Generator appelé Recherche permettant de rechercher une case en particulier et de verifier si elle accepte le déplacement sans compremettre l'intégrité du personnage ou du plateau.

L'utilisation des patterns nous permaient de pouvoir facilement changer les regles entourant ces cas d'utilisation afin de permettre une evolution ou une maintenance facilité.

4.2. Explication des diagrammes

4.2.1. Diagramme cas d'utilisation

Dans ce diagramme nous avons identifié deux acteurs pour ce programme ainsi le joueur (humain) est un acteur primaire pouvant lancer la partie,charger,sauvegarder ou bien déplacer son Corsaire. Le pirate(ia) acteur secondaire ne peut que lancer un combat ou bien se déplacer. Les Fonctions Creuser et Ramasser objet se font automatiquement apres un déplacement si les conditions son valide.

4.2.2. Diagramme d'activité

Ce diagramme montre les differentes activité possible par un joueur comme défini par le diagramme cas d'utilisation. Par consequent un joueur peut premiere lancer une nouvelle partie, charger une partie ou bien quitter le programme. Nous partirons du principe pur cet exemple qu'il choisie de lancer une nouvelle partie ainsi le programme lance l'initialisation du plateau puis lance le tour ou le joueur pourra choisir entre sauvegarder et se déplacer apres un déplacement on verifie si le joueur a gagné sinon on passe au joueur suivant.

4.2.3. Diagramme de communication

Le plateau est notre objet principale de notre programme, le plateau communique ainsi avec le generator, la recherche, les differents personnages, et les cases.
Le syteme de tour est en communication avec le plateau et les divers personnages encore sur le plateau.

4.2.4. Diagramme de classe

Comme pour le diagramme de communication, le plateau est notre element principale il stocke dans des ArrayList les Corsaires et les Pirates presents sur le plateau et il stocke aussi les cases du plateau dans un tableau de cases. Les Corsaires herite de personnages, les boucaniers et les flibustier heritent de pirate qui herite lui aussi de personages. Les items ont aussi de l'heritage la pell herite de item et la machette,le mousquet et l'armure herite de itemCombat qui herite de item ainsi nous pouvons regroupé les items dans des categories pour facilité les méthodes. Le Plateau possede aussi un objet generator et un objet recherche pour appelé les méthode commune de ces classes.