

Rapport première itération TOB

April 2023

Leo Flamencourt, Téo Piseni, Frederic Camail ,Cylia Yangour, Alexis
Courboin, Emile Devos.

Contents

1	Présentation générale du projet	3
2	Application des méthodes agiles	3
3	Explication des fonctionnalités et choix d'implantation	3
4	Diagramme de classe (UML)	4

1 Présentation générale du projet

L'objectif de ce projet est d'appliquer les notions de Technologie Objet vues en cours grâce à la mise en oeuvre notre vision du jeu du Monopoly, Oligopol7, qui se joue directement depuis un ordinateur. Il permettra à l'utilisateur de jouer contre d'autres utilisateurs en multijoueurs sur un même écran, au sein d'un terrain 2D isométrique et une interface graphique vivante, surprenante et dynamique. Il mettra notamment en scène la ville de Toulouse et fera références à certains lieux emblématiques du centre ville et de l'ENSEEIH.

2 Application des méthodes agiles

En suivant les notions vues en méthode agiles, nous avons pris le choix de rendre, pour cette première itération, une version très simplifiée de notre Monopoly, sans interface graphique. Le but étant de se concentrer d'abord sur le "fond" du projet, en répartissant les fonctionnalités sur différentes classes et implémentassions élémentaires, afin de poser les bases et la structure initiale du jeu.

Nous avons donc réparti le travail suivant les fonctionnalités ; chaque personne a géré un aspect particulier (joueur, plateau, arbitre) en créant les classes et le méthode qui lui était nécessaire si possible les tests associés.

3 Explication des fonctionnalités et choix d'implantation

Voici les fonctionnalités implémentées pendant l'itération 1 (avec leur classe associée).

1. Gestion des tours : (arbitre)
 - Lancer le dé (joueur)
 - Avancer le joueur (joueur)
 - Action de la case (caseFonctionnelle)
 - Achat maison si libre et joueur veut (caseLibre)
 - Ne rien faire si veut pas acheter (caseLibre)
 - Payer joueur si maison appartient à autre joueur (caseHotel)
 - Ne rien faire si la maison nous appartient (caseHotel)
 - Fin du tour (arbitre)
2. Changer de joueur automatiquement (arbitre)
3. Arrêter jeu si un joueur n'a plus d'argent (banquerouteException)
4. Afficher les règles (jouer)
5. Lancer la partie (jouer)

6. Afficher les joueurs (plateau)
7. Afficher le plateau (plateau)
 - Afficher une case (caseGraphique)
 - Afficher l'appartenance d'un hotel (caseHotel)
 - Afficher qu'une case est libre (caseLibre)

4 Diagramme de classe (UML)

Ci-dessous notre diagramme de classe UML (premier "brouillon" puis UML final) :



