# 

# 

# Rapport final Monopolytech

## Sommaire

1. Revisite de notre modèle UML……………………………………………………………… 3
2. Code source du projet……………………………………………………………………………………. 5
3. Conclusion…………………………………………………………………………………………………………………. 6

## 

## Revisite de notre modèle UML

Beldi à toi de jouer mon pote

Screen du nouvel UML + explication des changements apportés

* Exceptions (exception mère de laquelle les autres héritents) :

L’exception mère est JeuException. Il hérite de RuntimeException.

* Nouvelles méthodes dans joueur

Il s’agit des méthodes :

* Eliminer qui sort un joueur et qui restaure la liste de ses investissements.
* Une fonction resetInvestissement qui restaure les investissement d’un joueur et qui désigne l’état comme le nouveau propriétaire.
* La méthode compenser qui de vend les investissement d’un joueur lorsque celui ci n’a pas assez de liquidité pour payer (soit à la BFP soit à un autre joueur) jusqu’à ce que la somme qui lui manque soit obtenue. Dans le cas où la somme n’est pas obtenue, le joueur est éliminé.
* getCurrentCase et setCurrentCase
* Nouvelle méthodes dans Plateau

generatePlateau qui crée les cases du plateau et attribue à chaque case un rôle (antitrsut, subvention, bfp, investissement)

* Investissement & Configuration Manager

- Investissement manager permet de définir les différents investissements

- Configuration manager met en place la configuration du jeu en lisant les instructions que le client entre au clavier lors de l’exécution du programme

* Partie devient Main -> nouvelles méthodes

On définit ici le joueur en précisant son nom et son type (agressif ou prudent) avec la fonction definePlayer. Le jeu est lancé par la fonction gameLoop.

## Code source du projet

Nous avons déployé notre projet sur un repository commun sur le GitLab de Polytech.

Le fait d’avoir un dépôt distant nous a permis d’apprendre à travailler en équipe et à diviser les tâches afin que le travail de chacun soit complémentaire.

Nous avons également appris à gérer les conflits.

Lien du repository GitLab : <https://gitlab.univ-lille.fr/clement.alba2.etu/monopolytech>

## Conclusion

En conclusion, ce projet nous a permis de mettre en œuvre toutes les notions que nous avons assimilées en cours de Programmation Par Objet.

Nous avons su diviser notre projet en classes et en packages, afin de simuler au plus proche ce qui se passe dans la vie réelle.

Nous pensons que c’est un projet réussi, car nous avons su répondre au cahier des charges. Le jeu fonctionne bien et finit toujours par avoir un gagnant (ce qui était notre problème majeur avant l’écriture de ce rapport).

Les exceptions sont bien gérées et sont “throw” au bon moment et au bon endroit.

Le joueur qui lance la simulation à une expérience utilisateur agréable et simple, et peut décider du nombre de joueurs, de leur comportement ainsi que de la configuration qu’il souhaite appliquer.

Chacun de nous a su apporter ses points forts à l'équipe, et grâce à une bonne utilisation de Git nous avons su les unifier.

Nous pensons que ce projet nous a beaucoup apporté et était d’une grande richesse collaborative.