

INFO-F209 - *Capitali\$t*

Hugo Callebaut Nil Fernandez Lojo Arnaud Leponce

Année académique 2022-2023

Description

Capitali\$t est un jeu de plateau dans lequel plusieurs joueurs s'affrontent en **achetant des propriétés, construisant des maisons** et en **faisant payer des loyers** aux adversaires passant par leurs propriétés. Un **capital de départ est alloué** à chaque joueur et il existe plusieurs moyens de gagner ou perdre de l'argent durant la partie. **Lorsqu'un joueur ne peut plus payer son dû il est éliminé**. Lorsqu'il ne reste **plus qu'un joueur, ce joueur est le vainqueur**. Pareillement, si la **durée fixée de la partie est écoulée, le joueur le plus riche est le vainqueur**.

NOTE : Toute **ressemblance avec le jeu Monopoly** est bien entendu purement fortuite.

Déroulement d'une partie

Le jeu se joue au tour par tour sur un plateau sous forme d'**un seul chemin de 40 cases**, formant un carré. Au début de la partie, **l'ordre dans lequel les participants joueront est déterminé et restera fixe** durant toute la partie. Les joueurs **lanceront deux dés** au début de leur tour et **avanceront du nombre de cases indiqué par la somme des deux dés**. En fonction de la case sur laquelle le joueur s'arrête, plusieurs actions seront possibles selon les règles données sur le document de règles disponible sur l'UV.

Modes de jeu

Nous vous demanderons de créer plusieurs modes de jeu, le premier étant **la version classique** de *Capitali\$t* et la seconde étant **la version rapide** (les versions sont détaillées dans les règles).

Création d'une partie

Lorsqu'un **joueur crée une partie**, il doit configurer (au moins) les paramètres suivants :

- le **nombre maximal de joueurs**;
- le **mode de jeu**;
- le **capital de départ** des joueurs;
- le **temps maximal autorisé par tour**;
- la **durée maximale d'une partie**.

Fonctionnalités du jeu

Tout joueur doit être capable de :

- **créer un compte** associé à un **pseudonyme** et **un mot de passe**;
- **se connecter** à son compte;
- **gérer une liste d'amis**;
- **configurer** et **créer une partie**;
- **rejoindre une partie** déjà créée;
- **inviter** un ou plusieurs **amis à sa partie** créée;
- **discuter** avec ses amis;
- **consulter un classement des joueurs existants** selon une ou plusieurs mesure(s) au choix;

- jouer au jeu *Capitalist* en suivant les règles du jeu sans laisser la possibilité aux utilisateurs de tricher. En d'autres termes, le programme ne doit pas permettre aux joueurs de faire des actions qui violent les règles du jeu.

Consignes

Le projet se déroulera en 3 phases. Pour la **phase 1** de ce projet, il vous est demandé de penser en profondeur l'architecture du programme que vous allez réaliser sous forme de diagrammes UML et d'un squelette de code contenant un ensemble de classes (avec méthodes et attributs) que vous implémenterez dans les phases ultérieures.

Une première version de votre *Software Requirement Document* (SRD) sera demandée un peu avant la fin de la première phase. Nous vous donnerons alors un feedback qui vous permettra d'améliorer votre document avant la remise finale qui servira d'évaluation pour cette première phase.

Les phases 2 et 3 seront destinées à l'implémentation en C++, sur Linux, de votre jeu. Durant la **phase 2** vous devrez implémenter le jeu dans sa version terminal en suivant votre préparation rédigée dans le SRD. Lors de la **phase 3**, vous devrez alors modifier votre code pour ajouter une interface graphique tout en maintenant la compatibilité avec la version terminale.

Étant donné l'ampleur du projet, il vous sera demandé de bien structurer votre code. Par exemple, les paramètres par défaut du jeu (temps de partie, prix des terrains, etc) doivent se trouver dans un fichier externe ou sous la forme d'une base de données.

1 Contacts

Vous êtes invités à contacter en priorité l'assistant qui a été assigné à votre groupe.

- Hugo Callebaut (P.N3.207) : hugo.callebaut@ulb.be
- Nil Fernandez Lojo (Teams) : nil.fernandez.lojo@ulb.be
- Arnaud Leponce (P.N8.213) : arnaud.leponce@ulb.be