Comme il ne sera pas possible de tout faire dans le temps imparti, nous avons décidé de commencer par le minimum pour pouvoir jouer une partie. Une fois cela réalisé, nous pourrons l’enrichir avec de nouvelles fonctionnalités et nouvelles règles de jeu.

**Ce qui nous a semblé prioritaire pour jouer :**

Afficher à la console un plateau de jeu

Saisir le nombre de joueurs et leur nom

Lancer les dés

Déplacer un joueur

Afficher les informations d’une case

Acheter une propriété et qu’on ne puisse pas acheter la propriété de quelqu’un d’autre

Changer de joueur

**Idée de fonctionnalité à implémenter si temps extra** : Case casino, tu tombes dessus, tu lance les dés et tu peux multiplier ta mise.

**Nous avons ensuite réfléchi aux structures à définir :**

Plateau : tableau de t\_case

**t\_case :**

int numeroDeCase

tableau de Joueurs (les joueurs qui sont sur cette case)

char ou int typeDeCase : propriété, départ, prison, carte chance

int proprietaire = numeroDeJoueur du joueur qui possède la propriété (-1 si la propriété n’appartient à personne, -2 si ça n’est pas une propriété)

int prixAchat (0 si ça n’est pas une propriété). Prix achat affiché si pas de propriétaire

**t\_joueur :**

Numéro de couleur // facultatif

int numeroDeJoueur (plus facile à manipuler que le nom)

char\* nomDuJoueur

tableau de Propriétés (les propriétés que le joueur a achetées)

int numeroDeCase (la case sur laquelle se trouve le joueur)

int soldeBancaire

peut-être aussi int resultatLancerDés

**t\_partie :**

int numeroDeLaPartie

int nombreDeJoueurs

tableau de Joueurs (les joueurs qui jouent cette partie)

**Nous avons listé les traitements :**

Afficher menu principal (et saisir le choix de l’utilisateur)

Saisir le nbre de joueurs

Saisir le nom des joueurs

Afficher les informations d’une case

Afficher le menu d’un tour de jeu :

Nom du joueur

1. Lancer le dé

2. Déplacer le joueur vers une case

3. Afficher les informations d’une case

4. Afficher les informations du joueur

5. Acheter

6. Tirer une carte

7. Recevoir un montant

8. Payer un montant

9. Passer au joueur suivant

10. Arrêter la partie

**Déroulement d’un tour de jeu plus précisément :**

-Lancer les dés

-Déplacer le joueur

-Si Atterrit sur la propriété d’un joueur, ou un impôt, (ou est passé par la case départ), payer son due

-Sinon, propriété libre, le joueur veut-il l’acheter ?

-Sinon, case spéciale (chance, communauté ?) : tirer carte et faire action

**Tous les traitements sont dans des fichiers .c différents.**

Les différents fichier .c sont les suivants :

-menus

-traitement

-affichage

-

**Tous les structures et prototypes des fonctions sont dans un fichier header**

**Nous avons mis en place un environnement pour nous permettre de travailler facilement à plusieurs sur le même projet :**

-Trello pour le suivi du projet (définir les actions à faire, les priorités, suivre l’état d’avancement, identifier les points de blocage)

-GitHub pour cloner le projet, fusionner nos modifications locales dans un répertoire centralisé,

-GitHub Desktop pour récupérer les modifications faites par les autres et envoyer nos propres modifications sur GitHub pour qu’elles soient fusionnées

**Déroulement d’une partie :**

Je connais le nombre de joueur, J’ai mon plateau et mon tableau de joueur

Boucle while(nombreDeJoueur != 1 || isGameStandBy) //la partie se déroule jusqu’à ce qu’il n’y ait plus de joueur ou qu’on décide de l’arrêter

**Déplacer le joueur :**

Si on lance les dés : on le déplace de la valeur des dés : position finale = position initiale + valeurDes

S’il atterrit sur la case « Aller en prison » ; on le déplace en case indice 9(10éme case du tableau)

et le booleen isPlayerinPrison = true, déclenchant des fonctions spéciales lors de son tour.

Si on tire une case qui nous fait nous déplacer : on le déplace de cette case.

Supprimer joueur :

Si joueur est éliminé, alors on passe son tour de force.

Soit on affiche dans le menu tour de jeu (seulement l’option 3 pour terminer le tour) soit on arrive a forcer le passage au joueur au tour de suivant.

Conditions pour que joueur soit éliminé :

Soldebancaire négatif ET rien à hypothéquer

Si soldebancaire négatif alors proposer au joueur d’hypothéquer

**15/12/2021 : Rajout d’un boolen** isPlayerInPrison dans la structure de joueur, pour faire des actions différentes lorsque le joueur est en prison.

Blinder saisie entier / caractère (fonction valider la saisie)

Scanf une string

La comparer à une chaine [1] [2] etc

Peut-être même la comparer a tout un tableau.

Si ma string n’est pas égale à une valeur du tableau alors refaire la saisie