

# Projet Monopoly

BTS SIO 1ère année

## 1 Introduction

Le Monopoly (litt. « monopole » en anglais) est un jeu de société américain. Le but du jeu consiste à ruiner ses adversaires par des opérations immobilières.<sup>1</sup>

Les règles de ce jeu étant multiples et complexes, nous allons simplifier quelques points afin de rendre la réalisation plus simple (certains points pourront être ajoutés en tant que bonus) :

- l'hypothèque est supprimée
- un joueur a un solde, mais pas de liste de billets; par exemple si un joueur a un solde de 10, on ne se demande pas si c'est un billet de 10, deux de 5, dix de 1, ...
- les gares, les cases chance, caisse de communauté, impôts et compagnie d'eau et d'électricité, ne seront pas traitées
- les cases « Départ », « Prison », « Parc gratuit » et « Aller en prison », seront des cases sans traitement (rien ne se passe)
- un joueur passant par la case « Départ » reçoit 200
- une maison augmente le loyer du terrain de 1.2, un hôtel de 1.4

## 2 Mise en place

Créer les fichiers suivants :

- Case.py
- Terrain.py
- CaseSpeciale.py
- Joueur.py
- Partie.py
- Plateau.py
- main.py
- test.py

## 3 Explication des fichiers

En POO, chaque fichier correspond à une classe. Ainsi le fichier **Case.py** correspond à la classe Case.

Pour utiliser une classe dans un autre fichier, il faut importer le fichier puis préciser quelle classe est utilisée. Exemple :

```
1 import Case
2
3 depart = Case.Case("Depart")
```

Pour éviter d'utiliser le nom du fichier en préfixe, on peut préciser dans l'importation ce que l'on souhaite utiliser.

```
1 from Case import Case
2 # from random import randint
3
4 depart = Case("Depart")
5 # a = randint(0, 4)
```

---

1. Wikipédia : <https://fr.wikipedia.org/wiki/Monopoly>

**Plateau.py** : Un plateau est défini par un tableau de cases.

Méthodes de Plateau :

— avoir\_terrain(i) : Donne le i-ème terrain du plateau.

**Case.py** : Classe générique de toutes les cases du plateau. Chaque case possède au moins un attribut commun : Un nom.

**CaseSpeciale.py** : Les équivalents des cases spéciales dans le « vrai » jeu sont les gares, les cases chance, ... . Une case spéciale est définie par un nom

**Terrain.py** : Un terrain est défini par un nom, une couleur, un nombre de maisons, un nombre d'hôtels, un propriétaire, un coût d'achat, un loyer, un coût pour mettre une maison et un coût pour mettre un hôtel. Ces 4 derniers attributs sont déterminés en fonction de couleur du terrain (voir plus bas).

Méthodes de Terrain :

— est\_achetable() : Indique si le terrain est ou non achetable. Cette méthode

— ameliorer\_terrain() : Améliore un terrain, rajoute d'abord une maison (si le joueur le peut et le veut) ; s'il y a 4 maisons sur le terrain, rajoute un hôtel (si le joueur le peut et le veut) ; s'il y a un hôtel, le terrain ne peut plus être augmenté

**Joueur.py** : Un joueur est défini par un nom, un compte, un tableau de propriétés et une position. Tous les joueurs, démarrent avec 1500, aucune propriété et une position égale à 0.

Méthodes de Joueur :

— tirer\_de() : le joueur tire les dés

— deplacement() : le joueur se déplace sur le plateau

— acheter() : le joueur achète un terrain

— payer() : le joueur est sur une propriété appartenant à quelqu'un, il paye alors le loyer

**Partie.py** : une partie est définie par un tableau de joueurs et un plateau.

Méthodes de Partie :

— choix\_action() : le joueur choisit une action à faire

— deplacement() : le joueur tire le dé et se déplace du nombre de cases correspondant, le système de déplacement est décrit plus bas

— traitement\_post\_deplacement() :

— si la case actuelle est un terrain et qu'il est achetable et que le joueur peut l'acheter, alors on propose le terrain à l'achat

— sinon si la case actuelle est un terrain et qu'il est achetable et que le joueur ne peut pas l'acheter, alors on indique au joueur qu'il ne peut pas l'acheter

— sinon si la case actuelle est un terrain et qu'il appartient à un autre joueur que celui qui est tombé dessus alors le joueur paye le propriétaire

— sinon si la case actuelle n'est pas un terrain (c'est une case que l'on appelle « spéciale »), alors on indique au joueur que cette case n'est pas achetable et on effectue l'éventuelle opération nécessaire (améliorations).

— tour() : le joueur choisit l'action à faire, se déplace et fait l'action post déplacement

— joueur\_faillite() : indique si un joueur est en faillite (attention un joueur ayant un compte à 0 n'est pas en faillite)

— definir\_gagnant() : indique le joueur ayant le plus haut compte en banque

## 4 Grille des prix selon les couleurs

couleur du terrain	marron	bleu	rose	orange	rouge	jaune	vert	violet
coût d'achat	60	100	140	180	220	260	300	350
loyer	40	80	120	160	200	240	280	330
coût construction maison	100	100	150	150	200	200	300	350
coût construction hôtel	500	500	700	700	900	900	950	1000

## 5 Exemple d'exécution du programme

Ceci est un exemple, vous pouvez faire l'affichage que vous souhaitez, il faut que le joueur comprenne ce qu'il peut et ce qu'il doit faire.

```
Tour de Aboudou

Que voulez-vous faire ?
1 - Tirer le dé
2 - Consulter mon compte
3 - Voir sur quelle case je suis
Mon choix : 1

Aboudou tire le dé
Aboudou a fait 1
Aboudou se déplace de 1 case(s)
Aboudou est arrivé sur la case Boulevard de Belleville
Boulevard de Belleville n'a pas de propriétaire
Boulevard de Belleville coûte 60 et Aboudou a 1500
Voulez-vous l'acheter (1/0) ?

Aboudou a acheté Boulevard de Belleville
Aboudou a 1440€
Il possède :
- Boulevard de Belleville
```

## 6 Améliorations

**Il est demandé à chaque groupe de réaliser au moins deux améliorations différentes.**

Liste non exhaustive d'améliorations possibles :

- Implémenter les diverses cases spéciales.
- Implémenter les cartes chances et caisse de communauté.
- Implémenter le système de l'hypothèque.
- Implémenter le système des doubles (Rejouer et prison au bout de trois doubles consécutifs).
- Ajouter une interface graphique.
- etc.