# **Exercice : Un joueur achète un terrain**

## Modélisation UML des objets :

Une image contenant texte, capture d’écran, Police, diagramme

Description générée automatiquement

## Mise en contexte :

Au départ :

* Tous les terrains sont à vendre
* La Banque a tous les terrains qui sont à vendre.
* Les joueurs ont une liste vide de terrains achetés.

## Banque

Hérite de Joueur

La méthode privé \_charger\_terrains() va lire le fichier csv contenant les données de terrains, et crée une liste d’instances de la classe Terrain, un pour chaque ligne dans le fichier.

Cette méthode est appelée une fois lors de la création de l’instance Banque.

## Terrain

Les informations pour crée un terrain viennent du fichier « terrains\_original.csv ». À noter que ce ne sont pas toutes les valeurs dans le fichier qui sont utilisé.

La liste de prix\_location correspond au 5 dernière valeurs du csv

La valeur de « prix\_passage\_actuelle » provient des valeurs dans prix\_location, est commence équivalente à la première valeur.

La méthode **batir** fait passer la valeur de « prix\_passage\_actuelle » à la prochaine dans « prix\_location »

## Voici le pseudo-code pour qu’un joueur puisse acheter un terrain

Vérifier si le « propriétaire » a bel et bien le terrain qu’on veut acheter

Si oui

Vérifier si on a assez d’argent

Si oui

Retirer le terrain de la liste des terrains du propriétaire

Retirer le cash du joueur

Ajouter le cash dans le montant\_cash du propriétaire

Ajouter le terrain de la liste des terrains du joueur

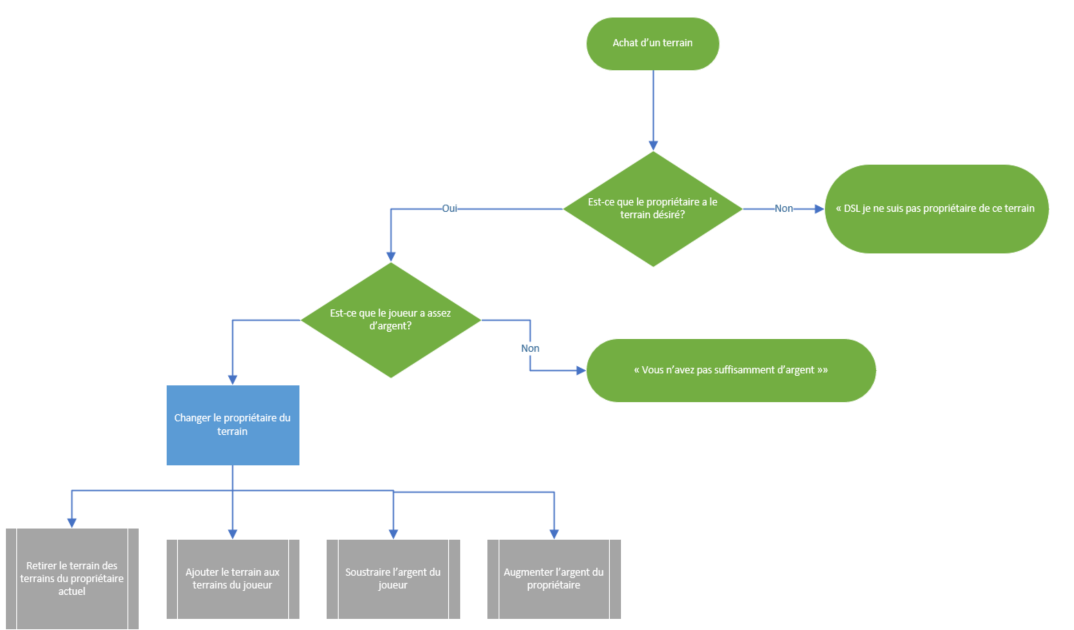
Si non

Écrire un message « Vous n’avez pas assez d’argent pour acheter le terrain X »

Si non

Écrire un message « Désolé, je ne suis pas propriétaire de ce terrain »

## Voici le diagramme de flux pour l’achat d’un terrain



## Création des différentes classes dans Python.

Avant de pouvoir faire le code de l’achat d’un terrain par un joueur, il faut qu’il y ait des terrains, un propriétaire qui les a et au moins un joueur pour acheter un terrain.

À partir du fichier ‘exercice\_Monopoly.py’ et à l’aide du diagramme de flux donné, complétez le code python pour créer les objets. Ce fichier comprend tous les objets mais ils ne sont pas complètement créés.

## Instanciez les objets des différentes classes.

Instanciez 3 terrains qui sont à vendre:

* ‘Avenue Kentucky’, rouge, 220000
* ‘Avenue Pennsylvanie’, vert, 320000
* ‘Promenade’, bleu, 400000

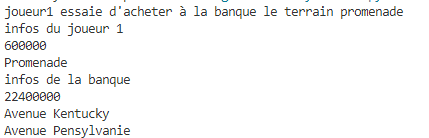
Instanciez la banque, avec 22 millions de cash, et la liste de ces 3 terrains car la banque au départ du jeu possède tous les terrains

Instanciez deux joueurs qui ont 1 million chacun et une liste vide de terrains.

## Testez qu’un joueur peut bel et bien acheter un terrain

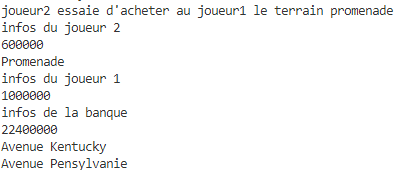
Test no 1 :

* Pour le joueur1, appelez la méthode Acheter() pour l’achat du terrain ‘Promenade’ à la banque
* Vérifiez que le montant\_cash du joueur1 a diminué et est maintenant de 600000
* Vérifiez que la liste des terrains du joueur 1 contient maintenant le terrain ‘Promenade’
* Vérifiez que le montant\_cash de la banque a augmenté et est maintenant de 22400000
* Vérifiez que la liste des terrains de la banque ne contient plus le terrain ‘Promenade’



Test no 2 :

* Pour le joueur2, appelez la méthode Acheter() pour l’achat du terrain ‘Promenade’ au joueur1
* Vérifiez que le montant\_cash du joueur2 a diminué et est maintenant de 600000
* Vérifiez que la liste des terrains du joueur 2 contient maintenant le terrain ‘Promenade’
* Vérifiez que le montant\_cash du joueur1 a augmenté et est maintenant de 1000000
* Vérifiez que les infos de la banque sont inchangées



Test no 3 :

* Pour le joueur2, appelez la méthode Acheter() pour l’achat du terrain ‘Promenade’ au joueur1

(Comme le joueur1 a déjà vendu son terrain Promenade, il ne l’a plus, d’où le message …)

* Vérifiez que les infos du joueur1, du joueur2 et de la banque sont inchangées

