

Rapport de Conception

Médéric HURIER
mederic.hurier@etu.univ-lorraine.fr

29 novembre 2012



Année 2012-2013

Table des matières

1	Analyse du besoin	3
1.1	Expression	3
1.2	Cas d'utilisation	3
1.3	Scénarios nominaux	4
2	Conception	6
2.1	Modèle du domaine	6
2.2	Classes participantes	8
2.3	Interactions	10
2.3.1	Initialisation	10
2.3.2	Jouer le tour	12
2.3.3	Acheter	14
2.3.4	Vendre une maison	15
2.3.5	Vendre un hôtel	15
2.3.6	Vendre un titre	16
2.3.7	Passer au joueur suivant	17
2.3.8	Quitter la partie	18
2.3.9	Finalisation	19
3	Prototype	20
3.1	Environnement de développement	20
3.2	Utilisation	21
3.3	Captures d'écran	22

1 Analyse du besoin

1.1 Expression

L'objectif du projet est la modélisation d'un logiciel permettant à plusieurs joueurs de disputer une partie de Monopoly sur un même terminal en mode texte. Monopoly est le jeu où l'on vend achète ou loue des propriétés de manière profitable, afin que les joueurs puissent s'enrichir (le plus riche d'entre eux étant déclaré vainqueur).

Les joueurs partent de la case "Départ" et déplacent leur pion sur le plateau en fonction du résultat des dés. Lorsqu'un joueur parvient sur la case d'une propriété à vendre, il peut l'acheter à la banque. Si il ne désire pas le faire, la propriété est proposée aux autres joueurs et ira au plus offrant.

Lorsqu'un joueur s'arrête sur la propriété d'un autre joueur, il doit lui verser un loyer. La construction de maisons ou d'hôtels augmente considérablement le montant des loyers. Il est donc sage de construire sur ses terrains. Les cases chance et communauté permettent de tirer une carte dont il faut suivre les instructions. il arrive aussi que l'on se retrouve en Prison. Le but du jeu est d'être le dernier joueur à rester en jeu, c'est à dire le dernier n'ayant pas fait faillite.

1.2 Cas d'utilisation

Le déroulement du jeu peut être séparé en un traitement automatique et un traitement interactif.

Le traitement automatique du Monopoly est basé sur des règles de jeu qui s'appliquent systématiquement et sans que le joueur puisse intervenir. L'utilisateur constate simplement le déroulement des actions qui sont effectués par le système.

Le traitement interactif donne au joueur la possibilité d'acheter et de gérer ses biens lorsque c'est son tour. Il n'est pas limité en nombre d'interaction, mais son choix peut être réduit en fonction de sa situation. Chacune de ces interactions peut être vue comme un service rendu par le système à l'utilisateur.

Le diagramme ci-dessous représente la partie interactive du jeu.

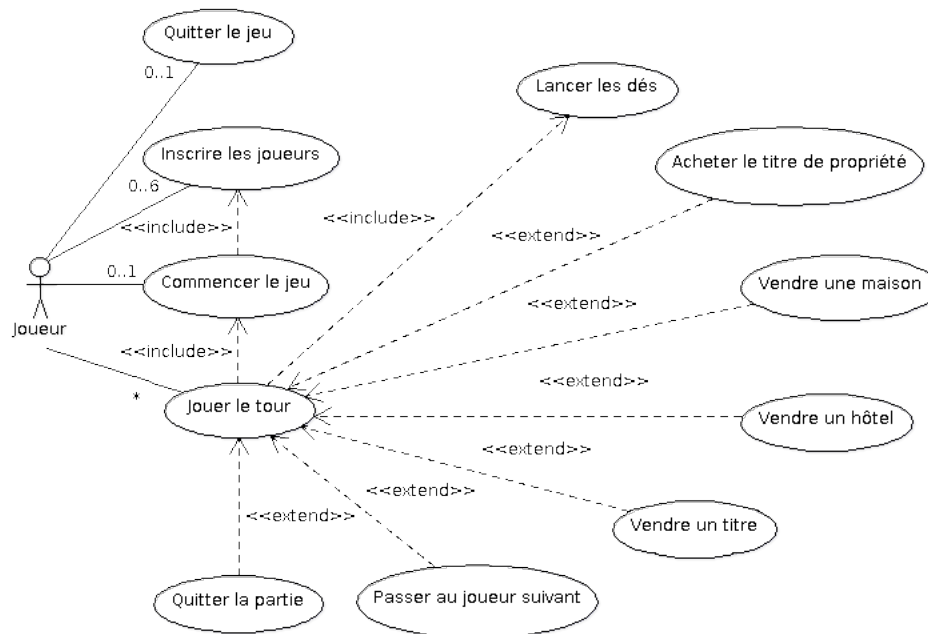


FIG. 1 – Diagramme des cas d'utilisation

Au lancement du logiciel, le propriétaire du jeu doit inscrire entre 2 à 6 joueurs pour commencer la partie. Tant qu'il n'a pas débuter une nouvelle partie, il peut décider de quitter le jeu.

Une fois la partie commencée, les joueurs vont jouer leur tour jusqu'à ce qu'il n'en reste plus qu'un. A chaque tour, le joueur doit obligatoirement lancer les dés pour se déplacer sur le plateau. Il peut ensuite choisir parmi une série d'action pour gérer ces biens, et termine forcément son tour en passant la main au joueur suivant ou en quittant la partie.

1.3 Scénarios nominaux

inscrire les joueurs

objectif ajouter un joueur (nom et couleur de pion) en vue de lancer une nouvelle partie.

acteurs propriétaire du jeu

préconditions pas plus de 6 joueurs ont déjà été inscrits

conditions le nom et la couleur du pion du joueur doivent être unique

commencer le jeu

objectif démarrer une nouvelle partie avec les joueurs inscrits jusqu'à ce qu'un gagnant ait été désigné.

acteurs propriétaire du jeu

préconditions au moins 2 joueurs sont inscrits au jeu

conditions aucune

quitter le jeu

objectif fermer le logiciel si le propriétaire de jeu ne souhaite plus jouer au Monopoly.

acteurs propriétaire du jeu

préconditions aucune

conditions aucune

jouer le tour

objectif permettre au joueur dont c'est le tour d'acheter et de vendre et d'obtenir des informations concernant sa situation.

acteurs joueur courant

préconditions ne pas avoir abandonner et ne pas être en faillite

conditions aucune

lances les dés

objectif déterminer le nombre de case dont le pion du joueur va avancer et si il s'agit d'un double pour rejouer.

acteurs joueur courant

préconditions aucune

conditions aucune

acheter

objectif acheter le bien courant, il peut s'agit d'un titre de propriété, d'une maison ou d'un hôtel selon le contexte.

acteurs joueur courant

préconditions le bien courant doit être achetable

conditions le joueur doit avoir assez d'argent et avoir le permi de construire

vendre une maison

objectif échanger une maison à la banque contre sa valeur d'achat

acteurs joueur courant

préconditions le joueur doit disposer d'une maison

conditions aucune

vendre un hôtel

objectif échanger un hôtel à la banque contre sa valeur d'achat

acteurs joueur courant

préconditions le joueur doit disposer d'un hôtel

conditions aucune

vendre un titre

objectif échanger un titre de propriété a un autre joueur.

acteurs joueur courant

préconditions le joueur doit disposer d'une maison

conditions l'acheteur du titre doit accepter la proposition et disposer d'assez d'argent

passer au joueur suivant

objectif passer la main au joueur suivant.

acteurs joueur courant

préconditions aucune

conditions aucune

quitter la partie

objectif retirer le joueur de la partie.

acteurs joueur courant

préconditions aucune

conditions aucune

2 Conception

2.1 Modèle du domaine

Monopoly est avant tout un produit concret composé d'un plateau, de pièces, de pions et de cartes. Pour ce type de problème, le paradigme de la programmation objet permet de représenter facilement cette réalité à l'aide de classes, d'attributs et de méthodes.

Le diagramme de classe ci-dessous schématise le modèle final du jeu de société. C'est couche comprend tous les objets nécessaires à la description d'une situation en cours de partie.

La représentation a été optimisé au cours de l'élaboration. Les versions intermédiaires, que vous pouvez consulter dans l'archive, montrent l'évolution de la représentation au cours des étapes.

Le point central du modèle est la classe `PARTIE`, elle regroupe tous les éléments du jeu dont les joueurs, le banquier et le plateau. Son rôle est de gérer les tours de jeu et d'effectuer plusieurs actions sur le joueur actuel (emprisonnement, abandon ou toucher un salaire).

Le `PLATEAU` de jeu est composé de deux paquets de carte, les cartes chances et les cartes caisses communautaires. Chaque carte décrit une action contraignante pour le joueur lorsque celui ci pioche une carte :

- les cartes `CHANCE` déplacent le joueur d'un certain nombre de case, en avant, en arrière ou sur une case précise.
- les cartes `CAISSE COMMUNAUTAIRE` envoient une amende ou un bonus au joueur qui s'applique immédiatement.

En plus de ces éléments, le plateau comporte une liste de `CASE` avec un nom pour situer le joueur. L'argent des amendes et des impôts est stocké au centre du plateau.

L'utilisateur de l'application est un `JOUEUR` identifié par un nom et une couleur de pion. Il se déplace sur le plateau, gagne et dépense de l'argent et peut abandonner, être emprisonné ou être faillite si il n'a plus assez d'argent.

Le joueur est associé à une classe `LANCE` qui matérialise son dernier lancé de 2 dés.

L'acteur système principal du modèle est le `BANQUIER`. Ce dernier est responsable des titres de propriété, de faire signer des contrats au joueur et de gérer les bâtiments. Un joueur doit forcément passer par lui pour gérer ses biens.

Un `TITRE DE PROPRIÉTÉ` est un élément abstrait descripteur d'un bien. Il se différencie en 3 types : `RUE`, `GARE` et `COMPAGNIE DE DISTRIBUTION`. Chaque objet peut renvoyer un loyer de séjour en fonction des contrats associés au titre.

Le `CONTRAT` lie un joueur et un titre de propriété. Il est géré par la banque, et contient également le nombre de maisons et d'hôtel présent sur le terrain. Si un contrat ne peut être passé, la banque renvoie un objet `CONTRATEXCEPTION`.

2.2 Classes participantes

Le modèle de domaine est étendu par les classes participantes de l'application. Elles contribuent à gérer les événements extérieurs (vues) et le déroulement du jeu (contrôleurs).

Cette logique correspond à une architecture MVC modifiée. Dans un MVC classique, les vues ne connaissent pas les contrôleurs et les communications ne se font que par des fonctions événementielles basées sur le patron de conception "Observateur".

Dans ce logiciel, les interfaces sont rendues interchangeables grâce au mécanisme d'interface objet. La dépendance des vues envers les contrôleurs est forte, mais la dépendance inverse est faible. Ainsi, il est possible de remplacer des interfaces en ligne de commande par des interfaces graphiques ou réseaux.

Ce diagramme de classe décrit les deux couches intervenant dans le processus de jeu.

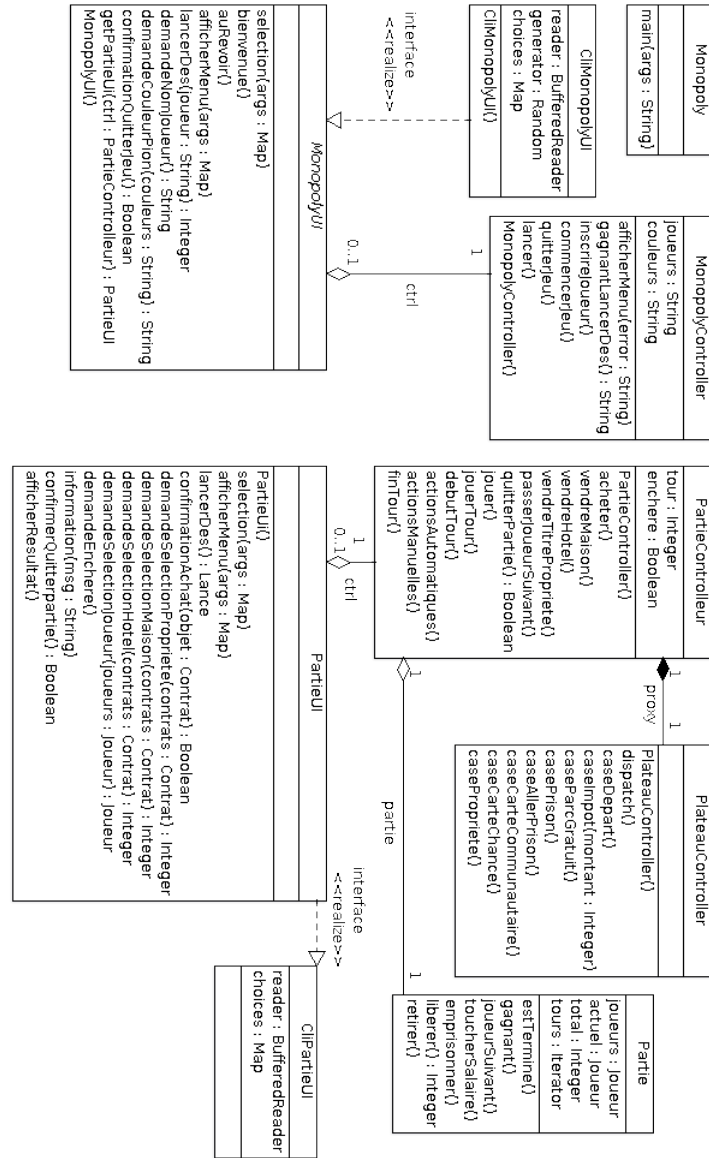


FIG. 3 – Classes participantes

Les couches se divisent en deux modules logiques, une pour la gestion du lancement du jeu et l'autre pour la gestion des parties.

Au lancement du logiciel, la classe MONOPOLYCONTROLLER gère l'inscription des joueurs, le lancement d'une nouvelle partie ou l'action de quitter le jeu. Elle communique avec la classe CLIMONOPOLYUI via l'interface MONOPOLYUI qui peut afficher un menu, renvoyer le nom d'un joueur ou la couleur d'un pion. Avant de débuter une partie, une partie de lancés de dés détermine quel est le joueur qui va commencer.

Une fois la partie lancée, la classe PARTIECONTROLLER prend le relais. Son rôle est de gérer toutes les

interactions avec le joueur (acheter, vendre, passer et quitter) et de gérer les tours de jeu. Elle communique avec la classe `CLIPARTIEUI` via l'interface `PARTIEUI` qui renvoie les informations nécessaires au jeu. Ces classes utilisent la classe modèle `PARTIE` pour accéder à la description intégrale du jeu.

La classe `PLATEAUCONTROLLER` a pour seul but de gérer les actions sur les cases. La méthode `dispatch()` est la seule méthode publique disponible, elle se charge d'appeler l'action de la case correspondante selon la position du joueur courant.

Cette modélisation évite d'implémenter la logique de l'application au niveau des couches modèles (les cases par exemple). L'échange avec la classe `PARTIECONTROLLER` peut être vu comme un proxy, un patron de conception qui relaie une partie des traitements à son contrôleur pair.

2.3 Interactions

2.3.1 Initialisation

Le diagramme de classe suivant schématise l'initialisation de l'application. Il va du lancement du logiciel jusqu'au premier tour du joueur.

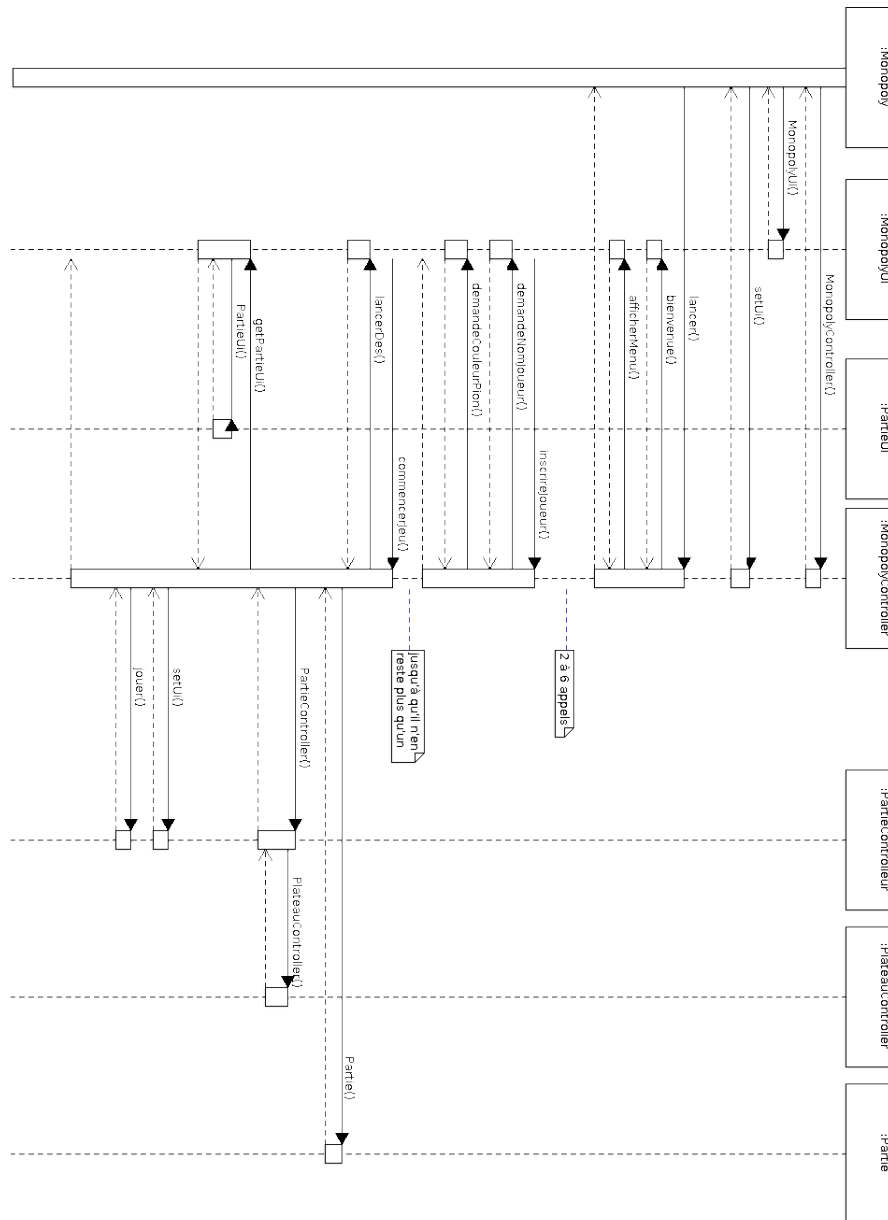


FIG. 4 – Initialisation

La première action du "main" de l'application est de créer un contrôleur de jeu MONOPOLYCONTROLLER et de lui associer son interface graphique, CLIMONOPOLYUI. Puis, le programme lance le contrôleur en appelant la méthode "lancer" qui affiche un message de bienvenue et le menu.

Le propriétaire du jeu est alors en charge d'inscrire les joueurs. Il doit inscrire de 2 à 6 joueurs pour qu'il a puisse commencer une nouvelle partie.

Après l'inscription, les joueurs devront se départager pour voir qui commence. Chaque joueur est invité

à lancer les dés jusqu'à ce qu'il n'en reste plus qu'un en liste.

Le lancement d'une nouvelle partie crée un `PARTIECONTROLLER` et lui associe une interface graphique `CLIPARTIEUI` et un modèle `PARTIE`. Grâce à la méthode `"getPartieUi"`, le contrôleur Monopoly n'a pas besoin de connaître la nature de la classe qu'il associe au contrôleur de partie.

Une fois tous ces éléments initialiser, le contrôleur de Monopoly appelle la méthode `"jouer"` du contrôleur de partie.

2.3.2 Jouer le tour

Une fois la partie lancée, les tours vont se succéder jusqu'à qu'il ne reste plus qu'un joueur.

Le diagramme ci-dessous montre tous les appels qui interviennent à chaque tour. A noter qu'un joueur peut jouer plusieurs tours d'affilé si il fait un double au lancé de dés.

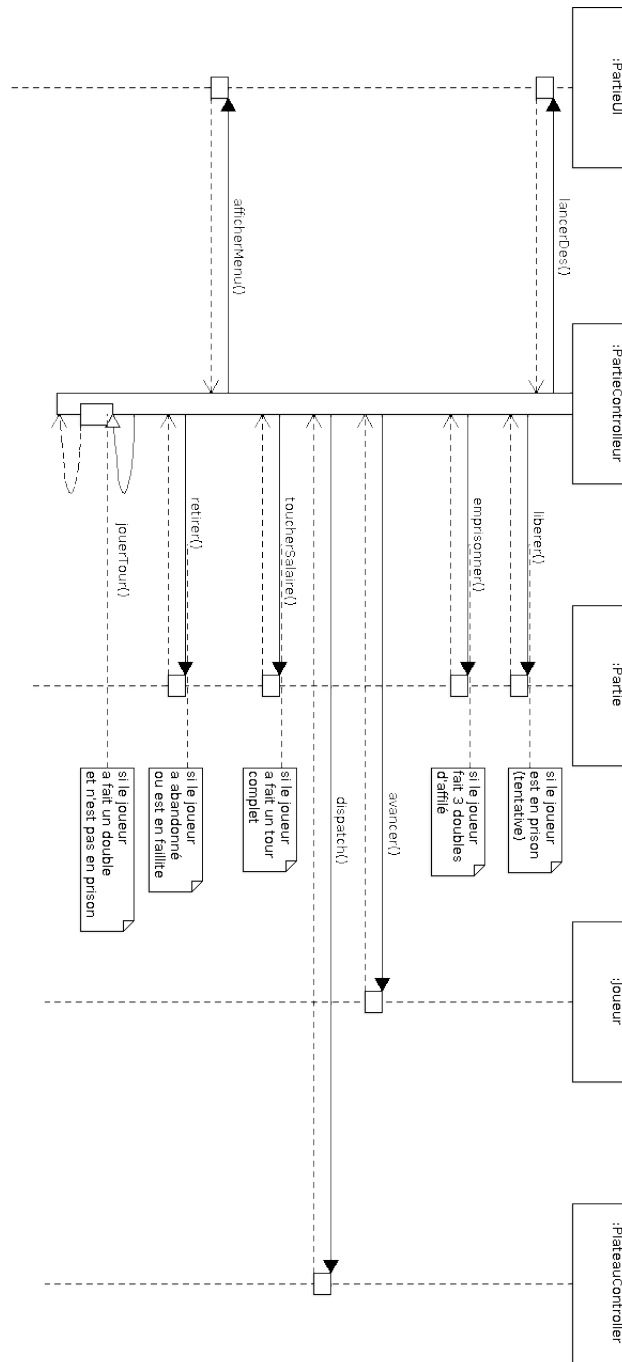


FIG. 5 – Jouer le tour

Chaque tour commence par un lancé de dés qui détermine le nombre de case que le joueur doit avancer. Si il est en prison, il lui est impossible de se déplacer, mais il peut espérer faire un double pour se libérer. A défaut, il sera libérer après 3 tours de prison en payant 5 000 \$ d'amende.

Si c'est le 3e tour d'affilé du joueur, celui ci sera directement envoyé en prison comme le stipule la règle.

Une fois le joueur arrivé sur une case, le `PLATEAUCONTROLLER` entre en jeu pour exécuter le code associé à la case. Pour cela, on appelle la méthode "dispatch" qui se charge de rediriger l'action en fonction de la position du joueur.

Si on détecte que le joueur a fait un tour complet du plateau, le jeu lui verse un salaire de 20 000 \$.

Une fois toutes ces opérations effectuées, le joueur a la possibilité d'effectuer des actions manuelles à partir du menu. Il peut acheter et vendre autant de fois qu'il le désire, mais devra indiquer à la fin de son tour si il se retire de la partie ou passe au joueur suivant.

En fin de tour, on vérifie si il le joueur a le droit à un tour supplémentaire lorsqu'il fait un double.

2.3.3 Acheter

Lorsque le joueur joue son tour, il a la possibilité d'acheter un bien ou de construire un bâtiment si toutes les conditions sont réunies. Les deux opérations de construction et de transfert de titre sont accessibles par la même action, car il est impossible qu'il y ai ambiguïté.

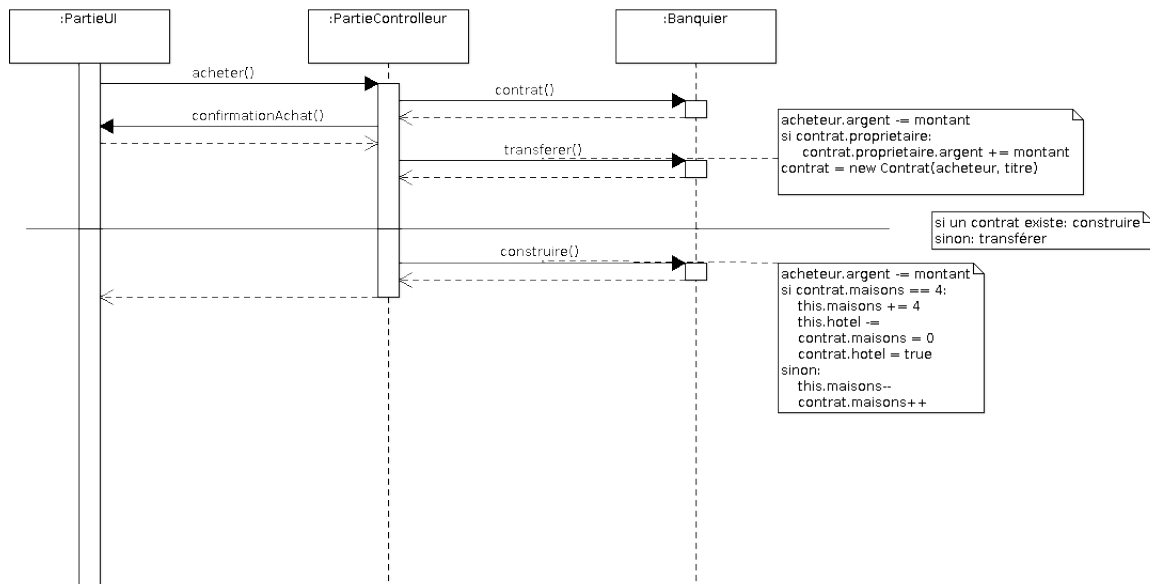


FIG. 6 – Acheter

La vue `PartieUI` appelle la méthode `acheter` du contrôleur de partie. Ce dernier récupère le contrat associé au titre de la case courante pour la suite du traitement.

Avant d'acheter, le joueur doit confirmer graphiquement sa volonté d'achat. Si il répond positivement, il est alors débité et obtient le bien qu'il désire à condition qu'il ai assez d'argent pour cela.

2.3.4 Vendre une maison

En cas de difficulté, le joueur a la possibilité de vendre une de ses maisons à la banque. Il récupère ainsi l'intégralité de l'argent qui lui a servi à acheter la maison.

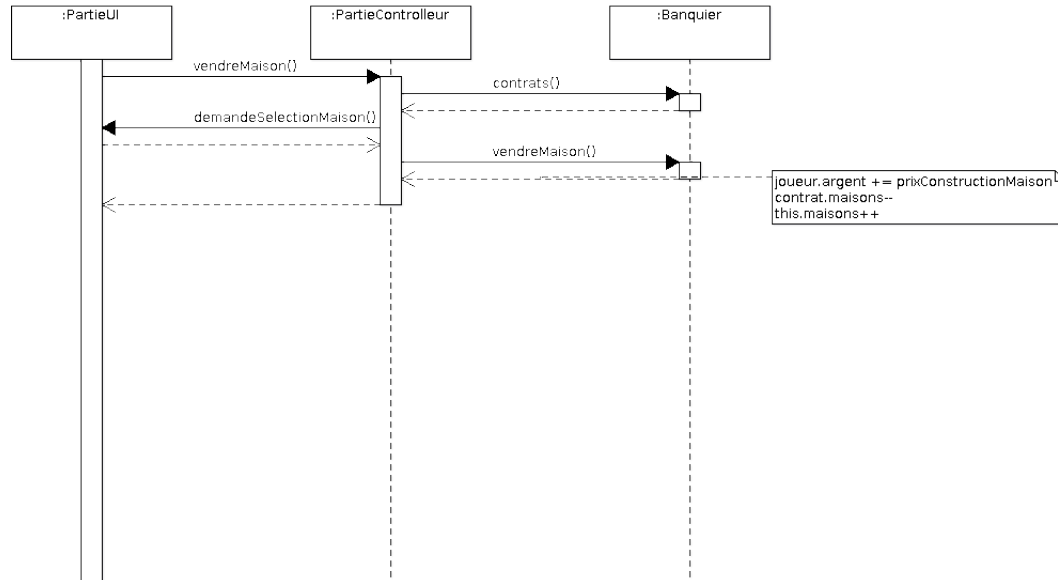


FIG. 7 – Vendre une maison

A l'appel de la méthode "vendreMaison", on demande tous les contrats du joueur à la banque. Ce dernier est alors invité à sélectionner l'une de ses maisons qui sera vendu dans la foulée.

La vente d'une maison rembourse au joueur le prix de construction de la maison, ce qui diminue la valeur du contrat et redonne à la banque une maison à vendre.

2.3.5 Vendre un hôtel

Ce cas est similaire à la vente d'une maison, mais on rembourse au joueur le prix de construction de l'hôtel et des 4 maisons qui ont été présente sur le terrain avant l'hôtel.

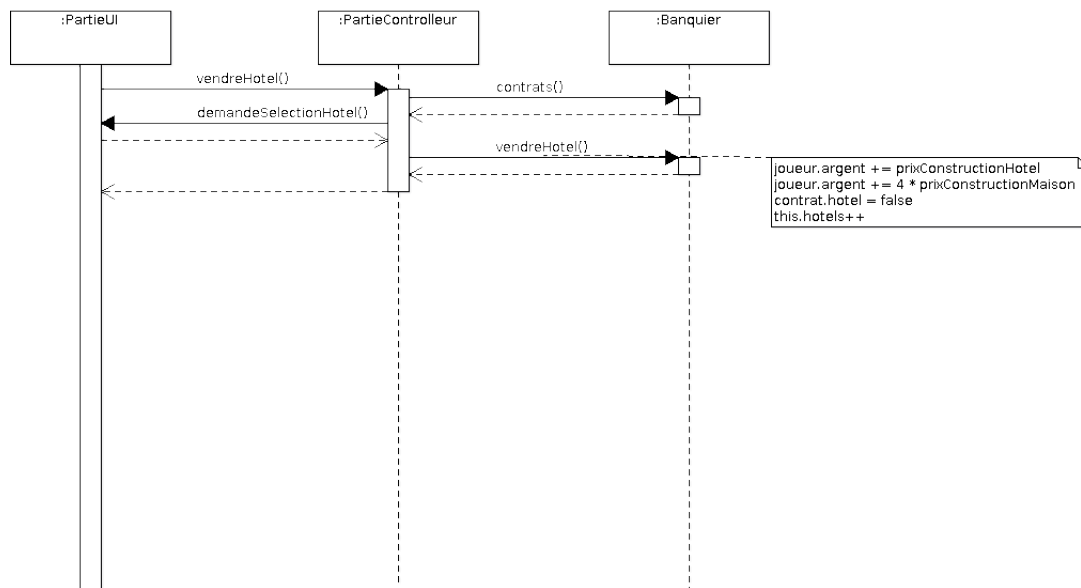


FIG. 8 – Vendre un hôtel

2.3.6 Vendre un titre

La vente de titre est un processus similaire à la vente de maison ou d'hôtel, mais le joueur vend à un autre joueur et non à la banque.

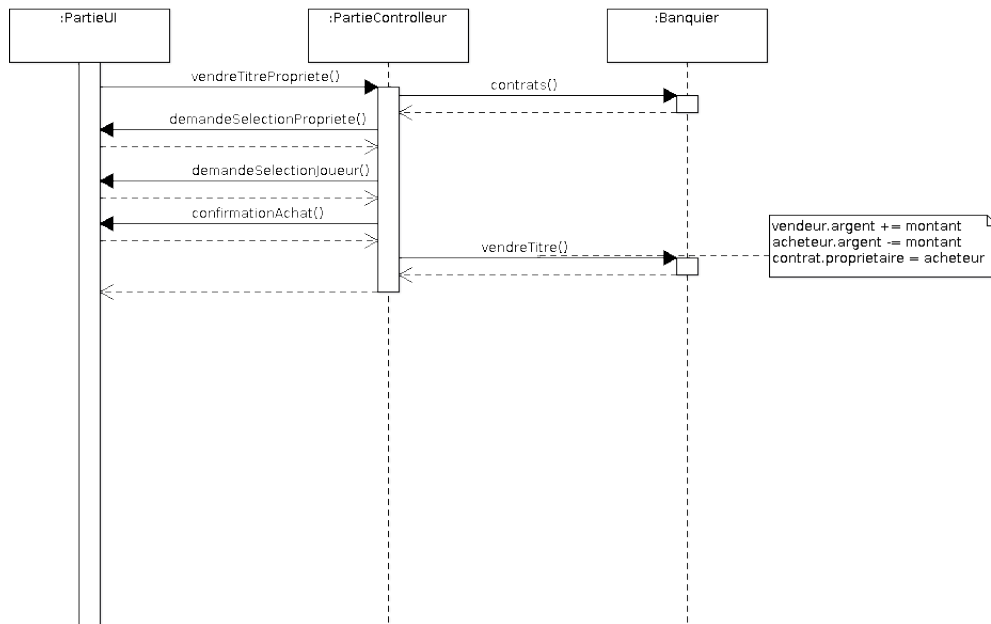


FIG. 9 – Vendre un titre

Dans un premier temps, le joueur est invité à choisir le titre dont il souhaite se séparer. Ensuite, il doit sélectionner le joueur auquel il souhaite le vendre.

Étape importante du processus pour éviter les abus, l'acheteur doit confirmer qu'il est bien disposé à acheter le bien.

Pour terminer, le titre est cédé au nouveau propriétaire si celui-ci a assez d'argent pour l'acheter. L'argent est échangé entre les joueurs (en tenant compte du prix des maisons et des hôtels présents), et le propriétaire du terrain est changé.

2.3.7 Passer au joueur suivant

Au cours de la partie, il est possible qu'un joueur refuse d'acheter un titre de propriété. Dans ce cas, la banque met au enchère le titre à la fin du tour, lorsque le joueur passe la main au joueur suivant.

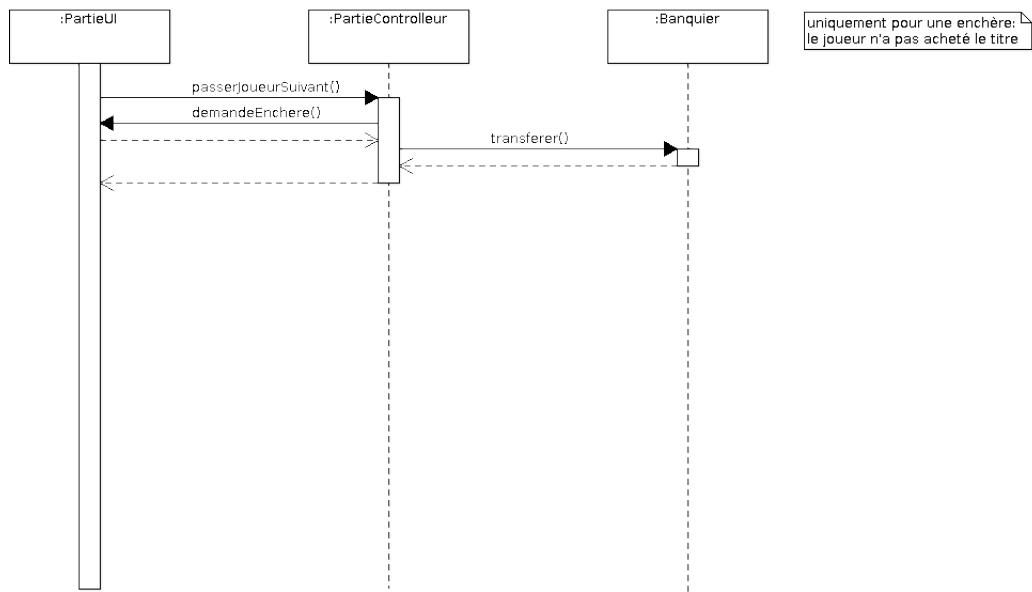


FIG. 10 – Passer au joueur suivant

Pour la demande d’enchère, la vue demande le joueur qui se propose pour acheter le terrain et le montant auquel il l’achète. Il doit être supérieure ou égale au montant du titre.

Si il joueur a assez d’argent, le titre lui est transféré immédiatement.

2.3.8 Quitter la partie

La dernière possibilité offerte au joueur est de quitter la partie. Ses biens sont alors automatiquement saisis par la banque qui les remet en jeu dès le prochain tour.

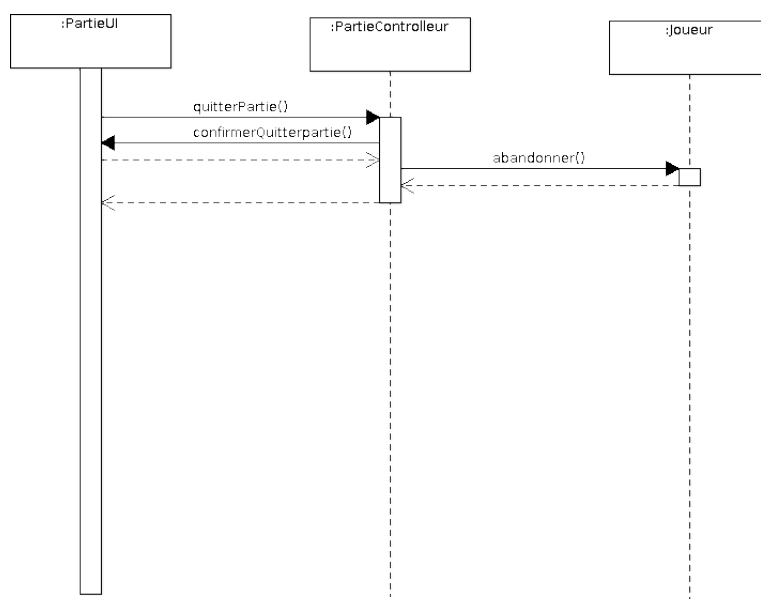


FIG. 11 – Quitter la partie

C'est un action simple où l'on demande au joueur de confirmer son choix avant de le retirer de la partie.

Le joueur est retiré automatiquement à la fin du tour.

2.3.9 Finalisation

La finalisation est une étape importante qui permet de libérer la mémoire et d'effectuer les dernières actions avant que l'application ne ferme. Avec l'environnement Java, la mémoire est automatiquement gérée par le Garbage Collector, nous laissant ainsi la liberté de nous consacrer aux autres traitements.

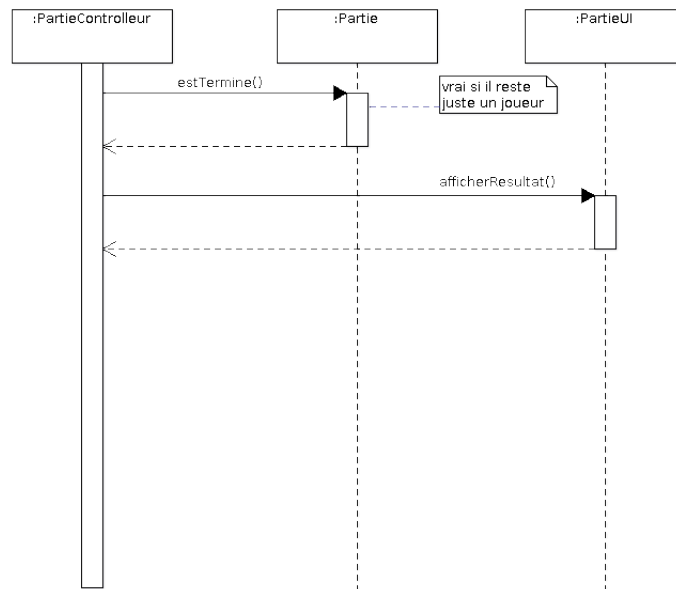


FIG. 12 – Finalisation

Lorsque le contrôleur de partie détecte que la partie est terminée, les résultats sont affichés. Le joueur gagnant est déclaré vainqueur, son nom et l'argent qu'il a recueilli sont affichés avant que l'application ne se termine.

3 Prototype

L'objectif pédagogique du projet Monopoly se focalise sur la bonne compréhension du sujet et la conception exhaustive avant d'entrer en phase de réalisation. Dans ce cadre, il est prématuré de coder le produit sans une réflexion préalable.

Néanmoins, la réalisation d'un prototype en cours de conception permet d'être sûr que notre vision du produit est correcte et fonctionnelle. Il faut pour cela mettre en place un environnement productif qui ne freine pas le processus de pensée.

3.1 Environnement de développement

Pour réaliser le prototype, j'ai choisi d'utiliser la technologie Groovy. Ce langage de programmation apporte beaucoup de raccourci à la programmation Java et permet de coder dynamiquement ou statiquement les types de données. Les traitements sont donc simplifiés, et il est possible à tout moment de revenir à un code source purement Java sans trop d'effort. Le code peut être exécuté dynamiquement, et être compilé en bytecode Java pour la portabilité. Pour plus d'information sur le langage, vous pouvez consulter ce lien : <http://groovy.codehaus.org/>.

Pour coder, j'utilise l'environnement netbeans 7.2 avec le plugins Groovy. Java est en version 7 pour supporter les dernières fonctionnalités.

3.2 Utilisation

L'application dispose d'une interface en mode console simple d'utilisation. A chaque demande d'interaction avec le joueur, le logiciel renvoie une commande "appuyer sur entrée pour lancer les dés", "sélectionner une action", ou "choisissez un bien/joueur de la partie".

Cette capture montre l'écran de démarrage de l'application. Elle comprend un menu à 3 entrées : inscrire le joueur, commencer le jeu et quitter le jeu.

```
=====
BIENVENUE AU JEU DE MONOPOLY DE FREAKMIND
=====

-----

MENU PRINCIPAL:
Taper 2: Inscrire un joueur
Taper 3: Quitter le jeu
>> 2

Entrer le nom du joueur: mederic

Couleurs disponibles: Rouge, Vert, Bleu, Jaune, Magenta, Noir
Entrer la couleur du pion: rouge

-----
Joueurs participants: Mederic(Rouge)

MENU PRINCIPAL:
Taper 2: Inscrire un joueur
Taper 3: Quitter le jeu
>> 2

Entrer le nom du joueur: lina

Couleurs disponibles: Vert, Bleu, Jaune, Magenta, Noir
Entrer la couleur du pion: bleu

-----
Joueurs participants: Mederic(Rouge) , Lina(Bleu)

MENU PRINCIPAL:
Taper 1: Commencer le jeu
Taper 2: Inscrire un joueur
Taper 3: Quitter le jeu
>>
```

FIG. 13 – Menu au lancement du jeu

Dans ce cadre d'utilisation simple, on a inscrit 2 joueurs (Médéric et Lina(ma soeur avec qui j'ai joué avant de concevoir le logiciel)) en entrant un nom et une couleur de pion. Les joueurs participants sont récapitulés à chaque requête.

Une fois le jeu commencé, une nouvelle invite s'affiche pour demander aux joueurs de lancer les dés. Le gagnant aura l'honneur de commencer. Puis, on affiche le menu principal de jeu.

```

Mederic~ appuyer sur entrée pour lancer les dés:
Votre lancé: 2 + 1 = 3

Lina~ appuyer sur entrée pour lancer les dés: Votre lancé: 2 + 2 = 4

-----
TOUR N°1: Lina
Appuyer sur entrée pour lancer les dés: Lancé: 6 (5 + 1)

Position: Rue de Vaugirard(6)
Argent: 150000 $

ACTIONS:
Taper 1: Acheter
Taper 5: Passer au joueur suivant
Taper 6: Quitter la partie
>> |

```

FIG. 14 – Menu au tour de jeu

Ici, le gagnant au lancé de dés est Lina qui a obtenu 4. Le menu montre les actions possibles lors de son premier tour : acheter un bien, passer au joueur suivant ou quitter la partie.

La liste complète des actions est :

1. acheter un titre, une maison ou un hôtel (seulement si les conditions sont remplies)
2. vendre une maison (uniquement si le joueur possède une maison)
3. vendre un hôtel (uniquement si le joueur possède un hôtel)
4. vendre un titre (uniquement si le joueur possède un titre)
5. passer immédiatement au joueur suivant
6. quitter la partie

Avant chaque tour, il sera demandé de lancer le dés en appuyant sur la touche entrée.

3.3 Captures d'écran

Les captures ci-dessous montrent plusieurs situations tirées du prototype.

```

-----
TOUR N°6: Lina
Appuyer sur entrée pour lancer les dés: Lancé: 7 (6 + 1)

Propriétés: Rue de Vaugirard
Position: Avenue Henri-Martin(24)
Argent: 108000 $

ACTIONS:
Taper 1: Acheter
Taper 4: Vendre un titre
Taper 5: Passer au joueur suivant
Taper 6: Quitter la partie
>> 5

Nouvelle enchère sur Avenue Henri-Martin à 24000 $
appuyer sur n'importe quelle touche pour passer l'enchère

Liste des joueurs:
0: Lina
1: Mederic

Entrer le numéro du joueur: 0

Vous avez 108000 $
Entrer le prix d'achat: 10 000
Le prix d'achat doit être supérieur au prix actuel

-----
TOUR N°7: Mederic
Appuyer sur entrée pour lancer les dés: Lancé: 10 (6 + 4)

# Parc Gratuit: vous recevez 52000$

Position: Parc Gratuit(20)
Argent: 182000 $

ACTIONS:
Taper 5: Passer au joueur suivant
Taper 6: Quitter la partie
>>

```

FIG. 15 – Enchère sur un titre et gain du parc gratuit

Ce premier cas montre le processus d’enchère. Le joueur Lina n’a pas voulu acheter l’avenue Henri-Martin et la banque doit mettre le titre aux enchères.

Le joueur qui se propose pour acheter se manifeste en entrant son numéro de joueur. Puis, on affiche le montant de l’enchère et l’argent disponible. Il entre ensuite son prix d’achat, et si celui ci est supérieure ou égale au prix du terrain, celui ci lui est transféré.

Dans ce cas précis, le joueur a entrée une somme trop petite pour obtenir le titre.

Dans un second temps, on voit que le joueur Médéric est tombé sur la case parc gratuit. Il reçoit tout l’argent au centre, soit 52 000 €. Les actions automatiques effectuées par le système sont des messages préfixés du symbole ”#”.

```

Propriétés: Place de la Bourse, Avenue de Breteuil
Position: Avenue de Breteuil(31), propriété de Lina
Argent: 62000 $

ACTIONS:
Taper 4: Vendre un titre
Taper 5: Passer au joueur suivant
Taper 6: Quitter la partie
>> 4

Liste de vos titres:
27: Place de la Bourse
31: Avenue de Breteuil

Entrer le numéro du titre de propriété:
27
Liste des joueurs:
0: Mederic

Entrer le numéro du joueur: 0

Êtes vous sûr de vouloir acheter Place de la Bourse à Lina ? (o/N): o

Propriétés: Avenue de Breteuil
Position: Avenue de Breteuil(31), propriété de Lina
Argent: 88000 $

ACTIONS:
Taper 4: Vendre un titre
Taper 5: Passer au joueur suivant
Taper 6: Quitter la partie
>>

```

FIG. 16 – Vente d'un titre

Ce cas de figure montre la vente d'un titre par un joueur. Celui ci désigne un titre et un joueur en se basant sur leurs numéros. Puis, l'acheteur potentiel doit accepter la proposition en appuyant sur la touche "o". On voit que le titre n'apparaît plus dans la liste des propriétés du joueur.

```

-----
TOUR N°14: Lina
Appuyer sur entrée pour lancer les dés: Lancé: 7 (3 + 4)

# Impôt sur le revenu: vous versez 20000$

# Tour complet: vous recevez 20000$

Position: Impôt sur le revenu(4)
Argent: 118000 $

ACTIONS:
Taper 5: Passer au joueur suivant
Taper 6: Quitter la partie
>> 5

-----
TOUR N°15: Mederic
Appuyer sur entrée pour lancer les dés: Lancé: 9 (5 + 4)

# Simple visite

Propriétés: Rue de Vaugirard, Place de la Bourse, Faubourg Saint-Honoré, Avenue de Breteuil, Avenue des Champs-Élysées, Boulevard de Belleville
Position: Prison(10)
Argent: 14000 $

ACTIONS:
Taper 4: Vendre un titre
Taper 5: Passer au joueur suivant
Taper 6: Quitter la partie
>> |

```

FIG. 17 – Impôt après tour complet et simple visite en prison

La dernière capture montre 2 autres situations. Au 14e tour, le joueur tombe sur la case impôt sur le revenu et doit payer 20 000 \$ mais reçoit 20 000 \$ pour avoir fait un tour complet.

Au 15e tour, le joueur tombe sur la case "Prison", et le système affiche qu'il s'agit d'une simple visite.