

Projet

But

Il existe de très nombreux jeux de société. Mais en y regardant de près, on peut s'apercevoir qu'il n'en existe que quelques familles bien distinctes. Les jeux de damier sont l'une de ces familles (échecs, dames, go, etc.). Une autre famille est composée des jeux de plateau où l'on doit s'enrichir comme le "Monopoly", "La bonne Paye", "Hôtel", etc.

Le but de ce projet consiste à étudier les jeux de plateau. Il s'agit de concevoir puis d'implémenter un ensemble de classes extensibles qui pourront, dans un deuxième temps, être facilement réutilisable pour réaliser plus facilement et plus rapidement les différentes variantes comme celles citées précédemment. Dans votre cas, nous nous limiterons à implémenter le Monopoly.

Planification

Etape 1 : conception du "modèle" de jeu de simulation. Définir l'ensemble des classes qui permet de gérer une partie. Vous devez trouver les abstractions, les exprimer de la manière la plus abstraite possible (des interfaces) mais également proposer des "factorisations" de code pour que l'implémentation des jeux concrets soit simplifiée quand même (avec des classes abstraites). A chaque fois que vous aurez une idée, vérifiez si elle peut être appliquée dans le Monopoly, mais aussi dans Hôtel, etc. Il s'agit de vérifier que votre proposition est extensible et non spécifique à un jeu donné. Il s'agit de conception donc cette étape consiste à élaborer un ou plusieurs diagrammes de classes. Ces diagrammes seront présentés au moment de la soutenance.

Etape 2 : implémenter ce modèle. Prévoyez toutes les exceptions nécessaires. Utilisez autant de packages que vous voulez pour conserver une architecture cohérente, donc lisible et facilement maintenable. Utilisez des identifiants clairs, et n'hésitez pas à donner des noms longs et significatifs.

Etape 3 : instancier le modèle avec le Monopoly. Vous devez pouvoir tester le déroulement d'une partie en mode "Console". C'est votre programme principal (la fonction main()) qui contient les instructions qui simulent la partie. Il n'est pas nécessaire que l'on puisse jouer en interagissant avec le programme. Toutes les classes de cette instance doivent être placées dans des packages bien identifiés.

Etape 4 : développer une interface graphique permettant de jouer réellement au Monopoly

Evaluation

Les critères pris en compte pour l'évaluation sont :

La validité : votre application doit être organisée comme ci-dessus et implémenter les règles décrites plus loin. Attention : ce qui compte avant tout, c'est un programme qui s'exécute ! Si vous n'avez finalement pas le temps d'implémenter toutes les règles du jeu, tant pis. Mais faites tourner quelque chose !

La robustesse : votre application doit fonctionner correctement (pas de plantage, pas d'incohérence lors de l'exécution, etc.) .

L'extensibilité : le modèle doit être bien conçu pour permettre une instanciation facile. Cette qualité est évaluée en analysant l'organisation de vos classes. Elle ne se voit pas nécessairement à l'exécution.

La lisibilité : le code doit être organisé en packages appropriés, les classes doivent être bien structurées avec les constructeurs qui conviennent. Les getters et les setters, la fonction equals, toString etc. doivent être présents sauf s'il ne les faut pas. Le code doit être bien commenté (javadoc), les identifiants doivent être bien choisis, etc.

L'ergonomie : l'interface graphique doit être jolie (mais ce critère n'est pas primordial).

Règles du Monopoly

Les règles ci-dessous sont extraites des règles présentes sur Wikipédia. Cependant, elles ont parfois été simplifiées pour éviter que vous ne passiez beaucoup de temps sur des détails. Notamment, vous ne gérerez pas les double (lorsque le 2 dés indiquent la même valeur). Ces simplifications réduisent évidemment l'intérêt du jeu, mais ne change rien à la difficulté du projet et à ce qu'il vous permet d'apprendre (Apprendre est quand même l'objectif principal du projet, pas jouer au Monopoly...).

Déroulement

Le jeu se déroule en tour par tour, avec deux dés ordinaires à 6 faces. Chaque joueur lance les dés, avance son pion sur le parcours, puis selon la case sur laquelle il s'arrête effectue une action correspondante :

- Un *terrain*, une *gare* ou un *service public* n'appartenant à personne : il a alors le droit de l'acheter
- Un *terrain*, une *gare* ou un *service public* lui appartenant : rien ne se passe
- Un *terrain*, une *gare* ou un *service public* appartenant à quelque autre joueur : il lui paye la somme due pour une nuitée passée sur ce terrain.
- Case *Chance* : il pioche une carte Chance. Cette case ne porte pas nécessairement bonne chance : il peut en effet s'agir d'une amende (à payer au Parc Gratuit). Chance (en français) est une adaptation de Chance (en anglais) qui signifie plutôt hasard.
- Case *Caisse de Communauté* : il pioche une carte Caisse de Communauté
- Case *Taxe de luxe* : il en paye le montant au Parc Gratuit.
- Case *Départ* : il gagne 200€, et 400€ s'il s'arrête dessus
- Case *Impôts sur le revenu* : il paye 200€ au Parc Gratuit
- Case *Allez en Prison* : le joueur va directement en prison (en reculant et donc sans passer par la case départ).

- Case *Simple visite* : case neutre
- Case *Prison* : il applique les règles pour en sortir
- Case *Parc gratuit* : tout au long de la partie, les sommes provenant des impôts et amendes sont versées sur cette case. Le joueur qui s'arrête sur la case Parc Gratuit ramasse cette somme. Le contenu Parc Gratuit est remis à 0

Les terrains sont groupés par couleur. Dès qu'un joueur est en possession de l'ensemble des terrains d'une même couleur, il est en mesure d'y construire des maisons et des hôtels, le joueur possède donc un monopole. Il doit construire uniformément : il ne peut y avoir plus d'une maison de différence entre deux terrains de la même couleur (un hôtel correspond à 5 maisons). De plus, on ne peut construire qu'un seul hôtel par terrain.

Lorsque c'est son tour de jouer, un joueur a également la possibilité d'effectuer d'autres actions, quelle que soit la case où il se trouve :

- Il peut vendre ou acheter des maisons des maisons sur des terrains qu'il possède
- Il peut abandonner (il y aura un joueur de moins dans la partie)
- Il peut finir son tour quand toutes ses obligations sont réalisées (payer un loyer, une amende, etc.)

Prison

Au cours de la partie, le joueur peut se retrouver en prison :

- en s'arrêtant sur la case : Allez en prison
- s'il pioche la carte : Allez en prison.

Lorsqu'un joueur est en prison, il pose son pion sur la case Prison. Pendant ce temps, il ne joue pas ; il attend son tour jusqu'à ce qu'il soit libéré.

Le joueur sort de prison :

- s'il détient la carte : Vous êtes libéré de prison
- s'il paye une amende de 50€ au Parc Gratuit

Le joueur ne peut rester plus de trois tours en prison. Le troisième tour, il est obligé de payer.

Fin du jeu

Le vainqueur est le dernier joueur n'ayant pas fait faillite (c'est-à-dire qui n'a pas abandonné) et qui possède de ce fait le monopole complet.

Vous trouverez facilement sur Internet, la liste exacte des terrains (choisissez la version française) et la liste des cartes Chance et Caisse de communauté.

Remarque générale : il y a beaucoup de terrains et de cartes à coder. Mais si votre modèle est effectivement extensible, alors il sera très facile de créer une version de base, par exemple avec uniquement les gares et la carte prison, qui permettent de vérifier que tout marche bien. Ensuite, en fonction du temps qui vous reste, vous complèterez avec d'autres terrains, d'autres actions, etc.

Groupe : le projet se fait par deux. Si vous êtes seuls, parlez-en à votre enseignant qui vous attribuera un groupe.

Soutenance : la soutenance aura lieu le plus tard possible, c'est-à-dire aux environs de la semaine d'examen.