

Rapport Final — Info 2A

Arthur PIARD

7 Mai 2018



Table des matières

1	Introduction	2
2	Cahier des Charges	3
3	Diagramme et description des classes	4
3.1	Diagramme	4
3.2	Description des classes	4
4	Manuel Utilisateur	7
5	Essais de l'application	9
6	Conclusion	11

1 Introduction

Il s'agit de modéliser ce jeu, en respectant les règles du Jeu de L' Oie, créer un système de tour par tour, modéliser les dés qui permettent de se déplacer dans le jeu. Il faut créer un tapis de jeu, un tableau de case contenant 64 cases, travailler sur l'affichage. En effet, en Modélisation Objet puisque l'on n'utilise pas d'interface graphique il faut créer un affichage clair et concis afin d'être bien compris par les utilisateurs. Il faut aussi utiliser les liens d'héritages ou d'utilisations entre les classes, avoir la possibilité de choisir le nombre de joueur, donner un nom aux joueurs, ajouter ou supprimer des joueurs. Simuler les lancers de dés, créer des actions sur les cases spécifiques (case piège, oie, téléportation...).

Règles basique du Jeu :

- Si un joueur fait 6, il doit se rendre sur la case 12. (ici, il s'agit d'une case de téléportation dans le jeu de base je crois qu'il s'agit d'une case oie).
- Si lors de la partie, le joueur tombe sur une oie, il avance de nouveau du nombre de points réalisés.
- Le joueur qui tombe sur la case 19 correspondant à un hôtel devra passer son tour durant 2 tours (Ici, l'hôtel est considéré comme une case attente).
- Le joueur qui tombe sur la case 31 correspondant au puits attendra qu'un autre joueur arrive au même numéro et prendra sa place (ici il s'agit une case piège, toute fois la règle reste la même le joueur dans cette case reste bloqué jusqu'à ce qu'un autre joueur arrive dans cette case).
- Celui qui tombe sur la case 42 correspondant au labyrinthe retournera obligatoirement à la case 30.
- Qui ira en 52 correspondant la prison attendra qu'un autre joueur vienne au même numéro pour repartir (même commentaire que pour la case 31).
- Le joueur qui va sur la case 58 correspondant à la case Tête de mort recommencera la partie depuis le début (ici, il s'agit une case de téléportation, la règle reste la même).
- Celui qui est rejoint par un autre joueur sur la même case devra se rendre sur la case où l'autre joueur se situait avant de jouer.

Pour gagner une partie du jeu de l'Oie, il faut être le premier à arriver sur la dernière case numéro 63 mais avec l'obligation d'être pile sur cette case.

Dans le cas où le joueur fait un score aux dés supérieur au nombre de case le séparant de la victoire, il devra reculer d'autant de cases supplémentaires

2 Cahier des Charges

Au lancement du jeu, le programme affiche un message de bienvenue, ensuite il est demandé à combien de joueurs nous souhaitons jouer enfin il faut entrer le nom des joueurs. Le jeu se décompose en tours, chaque joueur est dispensé de jouer les dés manuellement mais un algorithme le fait pour lui. Un affichage permet d'avoir les résultats des dés de chaque joueur, si un joueur tombe sur une case spéciale le jeu crée un affichage correspondant à la case en question. Si deux joueurs sont sur la même case les joueurs sont échangés et un affichage informe les joueurs de l'échange. Comme dit précédemment l'affichage se fait tour par tour, pour passer au tour suivant, l'utilisateur doit appuyer sur Entrée, l'affichage du tour suivant apparaît. Sinon pour quitter, l'utilisateur doit appuyer sur N puis Entrée. Le jeu continue jusqu'à ce qu'il y ait un vainqueur ou bien si le nombre de tour maximal est atteint (défini dans le Main()). A la fin d'une partie il est possible de refaire une partie soit en supprimant un joueur (si le nombre de joueurs est strictement supérieur à un) soit en ajoutant un joueur (si le nombre de joueurs est strictement inférieur à cinq). Si le nombre de joueur atteint un (respectivement cinq) joueur(s), alors le programme nous propose de lancer une partie avec deux (respectivement quatre) joueurs ou bien alors de quitter le jeu. Il sera possible de fixer un nombre de tours de jeu, de quitter le jeu à n'importe quel moment grâce à un contrôle de la saisie de l'utilisateur. Il sera possible de créer le jeu manuellement avec des cases de n'importe quel type dans une certaine méthode.

permet le déroulement du jeu. Il y a aussi la classe abstraite `JeuDeLoie`, qui est abstraite à cause de la méthode abstraite `initJeu()`. Cette classe a comme attributs un tableau de cases ainsi qu'un nombre de cases. Une classe `JeuNormal` qui est héritée de `JeuDeLoie` permet la construction du tapis de Jeu. Elle crée le tableau, et fixe les cases spéciales grâce à la méthode abstraite de `JeuDeLoie` `initJeu()`.

- Classe Abstraite `Case` : c'est une classe qui est décrite uniquement par des méthodes abstraites, ces méthodes gèrent la majorité du jeu.
- `public abstract Joueur getJoueur();`
- `public abstract void setJoueur(Joueur j);`
- `public abstract void Echange(Joueur j);`
- `public abstract boolean estLibre();`
- `public abstract void passeTour();`
- `public abstract boolean estBloquer();`
- `public abstract int getIndexCase();`
- `public abstract int ActionAFaire(int de);`
- Classe `CaseAttente` extends `Case` : c'est une classe qui étend la classe `Case`, elle reçoit donc toutes les méthodes abstraites qu'elle contient. Elle contient deux constructeurs, l'un prenant en paramètre l'index de la case et le joueur, l'autre prend en paramètre l'index de la case et le nombre de tour à attendre. Elle contient aussi les accesseurs en lecture et en écriture pour le nombre de tour à attendre.
- Classe `CaseBasique` extends `Case` : c'est une classe qui étend la classe `Case`, elle reçoit donc toutes les méthodes abstraites qu'elle contient. Elle contient deux constructeurs, l'un prenant en paramètre l'index de la case et le joueur, l'autre prend en paramètre uniquement l'index de la case. Elle ne contient pas d'accesseurs.
- Classe `CaseDepart` extends `Case` : c'est une classe qui étend la classe `Case`, elle reçoit donc toutes les méthodes abstraites qu'elle contient. Elle contient deux constructeurs, l'un prenant en paramètre l'index de la case et la liste de joueurs, l'autre prend en paramètre uniquement l'index de la case, la liste est instanciée. Elle ne contient pas d'accesseurs.

- Classe CaseOie extends Case : c'est une classe qui étend la classe Case, elle reçoit donc toutes les méthodes abstraites qu'elle contient. Elle contient deux constructeurs, l'un prenant en paramètre l'index de la case et le joueur, l'autre prend en paramètre uniquement l'index de la case, le joueur étant fixé à null. Elle ne contient pas d'accesseurs.
- Classe CasePiege extends Case : c'est une classe qui étend la classe Case, elle reçoit donc toutes les méthodes abstraites qu'elle contient. Elle contient deux constructeurs, l'un prenant en paramètre l'index de la case et le joueur, l'autre prend en paramètre uniquement l'index de la case, le joueur étant fixé à null. Elle ne contient pas d'accesseurs.
- Classe LeJeu : c'est une classe qui a comme attributs une liste de joueurs, une variable jeu de type JeuDeLoie et un booléen qui permet de savoir si le jeu est fini ou non. Elle contient un constructeur prenant en paramètre la variable jeu. Elle a les accesseurs en lecture et en écriture du booléen finJeu, un accesseur en lecture qui permet de récupérer l'instance de JeuDeLoie, une méthode qui permet d'ajouter un joueur à la case départ. Une longue méthode qui s'apparente à une méthode execute() permet le déroulement du jeu.
- Classe Abstraite JeuDeLoie : c'est une classe qui a comme attributs le nombre de case du jeu, et un tableau de Case. Elle a un constructeur qui prend en paramètre le nombre de case, une méthode abstraite initJeu() qui permet la création du tapis de jeu. Elle a les accesseurs en lecture et en écriture du tableau de Case, une méthode de Case qui permet de retourner la case numéro n et enfin un accesseur en lecture pour le nombre de case.
- Classe JeuDeBase extends JeuDeLoie : c'est une classe qui n'a pas d'attributs. Elle hérite de la méthode initJeu() de JeuDeLoie, cette méthode construit le jeu, avec un tableau de 64 cases, créé les cases spéciales ...
- Classe Joueur : c'est une classe qui a comme attributs une case et un nom, elle contient deux constructeurs, l'un prenant en paramètre le nom du joueur et sa case, l'autre prenant en paramètre juste le nom du joueur et fixant sa case à null. Elle a les accesseurs en lecture et en écriture de ses attributs, une méthode lanceDe() qui simule un lancé de dé, ainsi qu'un autre méthode deuxLance() qui simule le lancé de deux dés. Une méthode toString() retourne le nom du joueur.

4 Manuel Utilisateur

- **1ère étape** : lancer le programme, choisir le nombre de joueurs

```
Bienvenu dans le Jeu de L'Oie !  
Choisissez le nombre de joueur puis saisissez  
le nom des joueurs, puis la partie commencera  
Vous pourrez arrêter la partie en cours si vous le voulez.  
Bon jeu à vous !
```

```
A combien voulez-vous jouer? Le chiffre doit être 2,3 ou 4
```

- **2ème étape** : Saisir les noms des joueurs (ici par exemple 3 joueurs)

```
A combien voulez-vous jouer? Le chiffre doit être 2,3 ou 4  
3  
Saisir le nom du joueur 1 :  
Louis  
Saisir le nom du joueur 2 :  
Henry  
Saisir le nom du joueur 3 :  
Mathieu
```

- **3ème étape** : le jeu se lance seul, l’affichage du jeu aussi, vous avez le choix entre appuyer sur la touche ‘N’ pour quitter le jeu ou appuyer sur Entrée pour continuer le jeu et passer au tour 2.

A combien voulez-vous jouer? Le chiffre doit être 2,3 ou 4

3

Saisir le nom du joueur 1 :

Louis

Saisir le nom du joueur 2 :

Henry

Saisir le nom du joueur 3 :

Mathieu

Tour : 1

Le joueur ci-dessous a rejoint une case de téléportation, case numéro : 6 Il est maintenant dans

Louis est dans la case :0. Résultat des dés : 6 , il est donc à la case numéro 12.

Henry est dans la case :0. Résultat des dés : 10 , il est donc à la case numéro 10.

Mathieu est dans la case :0. Résultat des dés : 7 , il est donc à la case numéro 7.

Pour lancer le tour suivant appuyez sur O puis Entrer, pour quitter appuyez sur N puis Entrer

Le jeu continue jusqu'à ce qu'il y ait un gagnant, ou alors si le nombre de tour maximum est atteint. Si il y a gagnant, on peut soit quitter la partie, soit en recommencer une en choisissant si l'on veut ajouter ou alors supprimer un joueur dans les conditions où il doit y avoir au plus quatre joueurs et au moins deux joueurs.


```

Tour : 19

Ce joueur est dans une case piège ! Il doit attendre que quelqu'un rentre dans cette case !
Henry est dans la case :60. Résultat des dés : 3 , il est donc à la case numéro 63.
Henry est dans la case 63 il a donc gagné ! Youpi !

Pour lancer le tour suivant appuyez sur O puis Entrer, pour quitter appuyez sur N puis Entrer

----- Fin du Jeu -----

Voulez-vous rejouer?
Oui
Voulez vous supprimer un joueur ?
Oui

Joueur numéro 3 supprimé
Que 2 joueurs maintenant !

Saisir le nom du joueur 1 :
Louis
Saisir le nom du joueur 2 :
Henry|

```

5 Essais de l'application

Démonstration du Jeu

Dans cette version de l'application il n'y a pas d'affichage au tour par tour, l'affichage est itéré jusqu'à ce qu'il y ait un gagnant ou bien que le nombre de tour maximal soit atteint. Dans notre exemple on voit qu'il y a eu 9 tours car le joueur Testeur1 a été le premier à atteindre la case 63.

Bienvenu dans le Jeu de L'Oie version DEMO!
Vous commencez avec 2 joueurs, veuillez donc saisir
le nom des joueurs, puis la partie commencera
Vous pourrez arrêter la partie en cours si vous le voulez.
Bon jeu à vous !

Saisir le nom du joueur 1 :
Testeur 1
Saisir le nom du joueur 2 :
Testeur 2

Tour : 1

La case 9 est une case oie ! Le Joueur ci-dessous est donc à la case numéro 18.
Testeur 1 est dans la case :0. Résultat des dés : 9 , il est donc à la case numéro 18.
Testeur 2 est dans la case :0. Résultat des dés : 8 , il est donc à la case numéro 8.

Tour : 2

Testeur 1 est dans la case :18. Résultat des dés : 7 , il est donc à la case numéro 25.
Testeur 2 est dans la case :8. Résultat des dés : 7 , il est donc à la case numéro 15.

Quelques tours plus tard ...

Tour : 6

Testeur1 est dans la case :43. Résultat des dés : 8 , il est donc à la case numéro 51.
Testeur2 est dans la case :46. Résultat des dés : 7 , il est donc à la case numéro 53.

Tour : 7

Testeur1 est dans la case :51. Résultat des dés : 10 , il est donc à la case numéro 61.
Testeur2 est dans la case :53. Résultat des dés : 9 , il est donc à la case numéro 62.

Tour : 8

Testeur1 est dans la case :61. Résultat des dés : 9 , il est donc à la case numéro 56.
Testeur2 est dans la case :62. Résultat des dés : 4 , il est donc à la case numéro 60.

Tour : 9

Testeur1 est dans la case :56. Résultat des dés : 7 , il est donc à la case numéro 63.
Testeur1 est dans la case 63 il a donc gagné ! Youpi !
BUILD SUCCESSFUL (total time: 10 seconds)

6 Conclusion

Pour la création de ce projet, il a été utilisé une classe Abstraite globale qui est Case. Cette classe contient des méthodes abstraites qui permettent de gérer le jeu dans presque son intégralité. D'autres classes qui étendent Case ont été créées du fait du nombre de case différente du jeu. Ces dernières ont donc accès aux méthodes abstraites de la classe mère. Les méthodes abstraites peuvent être construites suivant les besoins de la classe en question, les méthodes ne vont pas être les même dans chacune des classes. Le code en soi est relativement basique, un affichage simple est généré, on commence avec un nombre de joueur choisi on peut en avoir au plus quatre et au moins deux joueurs. Les consignes générales de ce projet ont été respectées. C'était un projet très ludique dans le sens où le Jeu de L'Oie est un jeu très connu. Il existe des milliers de façon de développer ce jeu et c'est ici que le développement est très intéressant car dans un même sujet il peut exister énormément de façon de faire. Sur Internet il est possible de consulter des programmes simulant ce jeu d'une certaine façon, un autre d'une autre façon, la programmation c'est le principe de créer, de la façon qui nous paraît le plus simple, un programme. La différence entre chaque programme est liée à l'optimisation du code. Ici il est clair qu'une optimisation aurait pu être envisagée, mais par manque de temps je préfère garder une version qui fonctionne et qui respectent les conditions du sujet plutôt que de refaire et optimiser une version qui pourrait finalement ne plus correspondre à ma vision du sujet.