

Projet Snake

Bases De Données



3. Votre Code

3.1. Implémenter le jeu Snake en mode texte

3.1.1. Classe Game

```
public class Game{
```

Pour écrire le jeu Snake en mode texte, je n'ai pas commencé par écrire la classe Game, mais par écrire la classe Board et la classe Snake. Cela n'a pas été une bonne idée, car je n'avais pas une idée exacte de la direction que je prenais en écrivant les classes Board et Snake. J'ai donc eu beaucoup de mal à rendre cohérentes les deux premières classes une fois que la classe Game commençait à prendre forme.

Dans cette classe Game, on retrouve donc deux fonctions principales : la fonction start et la fonction GameOver, basées sur les fonctions écrites dans les classes Board et Snake, détaillées plus loin dans le compte rendu.

Dans la fonction start(), on trouve l'interaction entre le code et l'utilisateur grâce à l'utilisation de la classe Scanner. En effet, il y est demandé au joueur d'entrer la direction souhaitée, grâce aux quatre touches : q = gauche, x = bas, d = droite, e = haut.

Une fois que l'utilisateur a entré une direction, la fonction start se charge des changements de direction. Elle se charge également de détecter la défaite du joueur, grâce à l'appel de la fonction GameOver définie ensuite. Pour finir, la fonction start s'occupe de faire grandir le serpent s'il détecte une collision entre la tête et une pomme.

La modification de la variable Delete dans la fonction start() est un point intéressant du code. C'est un tableau que je définis dans la classe Snake, initialement à {-1, -1} (coordonnées non présentes dans la grille). Grâce à la fonction getDelete() écrite dans la classe Snake, je permets de récupérer les coordonnées du dernier segment de la queue du serpent dans cette variable, qui devra être supprimé au mouvement suivant.

3.1.2. Plateau du jeu

```
public class Board {
```

Ma classe Board a été la première classe écrite dans mon projet. J'y ai initialisé les valeurs suivantes :

t : tête, w : mur, v : vide, s : corps du serpent et f : nourriture.

Dans un premier temps, je crée une grille grâce à la fonction creationGrid(), qui remplit le tableau double « grid », sous la forme d'un rectangle vide. Les bords de ce rectangle sont représentés par des « w » (mur) et l'intérieur est rempli de « v » (vide).

Il est ensuite question, dans la fonction draw, de traduire le tableau double « grid » en une grille visible par le joueur dans le terminal.

La dernière fonction de ma classe Board est la fonction foodHere, qui est utilisée dans la classe TwoPlayerGame, dans laquelle le joueur affronte l'intelligence artificielle. Cette fonction est appelée lorsque l'on souhaite vérifier s'il se trouve une pomme aux coordonnées passées en paramètre.

3.1.3. Classe Snake

```
public class Snake{
```

Dans cette classe Snake, on retrouve de nombreux setters, permettant aux classes Game, IA_Game et TwoPlayerGame d'obtenir des informations comme la position de la tête du serpent, de son corps ou du dernier segment de sa queue, la valeur booléenne permettant de savoir s'il faut que le serpent grandisse, etc. Une fonction importante de cette classe est la fonction move(), qui gère le mouvement du serpent sur la grille du jeu et son agrandissement si le serpent vient de manger une pomme.

3.1.4 Classe IA_Game

```
public class IA_Game {
```

Dans cette classe, la fonction run(), qui lance le jeu, est très similaire à la fonction start() dans la classe Game.

La partie intéressante se trouve dans les fonctions nextMove() et isSafePosition. La fonction isSafePosition, qui est utilisée dans la fonction nextMove(), sert à évaluer le mouvement qui pourrait être effectué au coup suivant. En effet, la première partie de la fonction nextMove() sert à définir la direction préférée du serpent en fonction de la position de la pomme par rapport à la tête. C'est donc ce que j'avais uniquement écrit initialement dans cette fonction. Mais je me suis rendu compte que la direction optimale pour atteindre la pomme était parfois une direction où se trouvait un mur ou un segment de la queue, ce qui provoquait la mort du serpent. Pour éviter ceci, j'utilise la fonction isSafePosition dans une boucle for, qui parcourt toutes les directions possibles que pourrait prendre le serpent et en sélectionne une sûre.

En effectuant des tests sur ma classe IA_Game, j'ai remarqué que le serpent partait directement vers le bas et entrait dans le mur lorsque la pomme était placée au-dessus de sa position. Pour éviter cela, j'impose un premier mouvement du serpent vers la droite, d'où l'utilisation de l'attribut premierMouvement au début de ma fonction nextMove().

3.1.5. Classe TwoPlayerGame

```
public class TwoPlayerGame {
```

Dans cette dernière classe de mon projet Snake, j'écris un code permettant à un joueur d'affronter l'intelligence artificielle créée précédemment.

Je déclare dans mon constructeur une variable booléenne playerTurn permettant de savoir si c'est au joueur de jouer ou non. Cela me sert à déterminer qui doit effectuer le prochain mouvement.

Pour vérifier si le joueur ou l'IA a mangé une pomme et doit donc grandir, j'utilise la fonction foodHere que j'ai écrite dans la classe Board.

J'ai écrit une méthode updateBoard qui met à jour l'affichage du plateau en fonction de la position actuelle du serpent (qu'il soit le serpent du joueur courant ou de l'IA).

Je mets d'abord à jour la position de la tête sur le plateau de jeu grâce à la fonction board.set : selon le joueur, la tête du serpent est différente.

- La tête du serpent du joueur humain est représentée par 't'.
- La tête du serpent de l'IA est représentée par 'T'.

Je mets ensuite à jour de la même manière le corps du serpent sur la grille.

- Les segments du corps du joueur sont représentés par 's'.
- Les segments du corps de l'IA sont représentés par 'S'.

Enfin, je gère la suppression de la dernière position de la queue. S'il y a un segment à supprimer (si le serpent avance) et que les coordonnées de delete ne sont pas égales à (-1, -1).

Conclusion

Ce projet Snake aura permis de mettre en application les connaissances en Java acquises en deuxième année de prépa intégrée, et dans le module base informatique de ce premier semestre. L'implémentation du jeu Snake en mode texte est le projet le plus conséquent que j'ai eu à réaliser en Java. Il m'a été bénéfique pour apprendre à rendre cohérentes un nombre 'important' de classes. J'ai aussi appris à travailler avec les bases de GitHub, j'en suis très content car j'ai conscience que cet outil est primordial dans la gestion de projets conséquents, qu'ils soient menés par une équipe ou tout seul.