GIANG Cécile (3530406), ZHANG Shuting (3772752)

COMPTE-RENDU DU MINI-PROJET JAVA (2I002)
- Feeding Shuting -

Pour notre projet, nous avons décidé de créer notre propre jeu, FeedingShuting, basé sur le jeu FeedingFrenzy.

BUT DU JEU :

Le joueur a pour objectif de se nourrir de poissons plus petit que lui. Ce faisant, son score augmente et à partir d'un certain seuil, le niveau ainsi que sa taille augmentent. Mais gare aux poissons plus gros que soi ! S'ils arrivent à manger le joueur, son score diminue, et il perd une vie. Le joueur peut aussi se nourrir de piranha (toxique) ou de perles (non-toxique).

MODALITÉS DE JEU :

Le jeu se déroule dans un tableau (**`Mer`**) d'éléments marins (**`ElemMer`**). Les éléments marins se divisent en deux catégories : les **`Poisson`** et la **`Nourriture`**.

Ici, le joueur est représenté par une instance (unique) de la classe **`Shuting`**, héritée de la classe **`Poisson`**. Il ne peut se nourrir que de poissons plus petits, auquel cas il gagne autant de points que le poisson possède d'énergie, et perd des points ainsi qu'une vie s'il se fait manger par un poisson plus gros. Parmi les classes héritées de **`Poisson`**, l'on compte : **`Shuting`**, **`Nemo`**, **`Tuna`**, **`WhiteShark`** et **`BigWhale`**.
Notons que tout poisson peut chasser, être chassé et se déplacer, mais ces propriétés sont spécifiques à chaque poisson, d'où la création d'interfaces correspondantes (**`Chasser`**, **`EtreChasse`**, **`SeDeplacer`**) et implémentées par **`Poisson`**.
Les proies et les prédateurs du joueur sont renseignés dans deux `ArrayList`, mises à jour à chaque niveau.

Le joueur peut également consommer de la **`Nourriture`**, mais celle-ci se décompose en nourriture **`Toxique`** (**`Piranha`**) qui, lorsqu'elle est consommée, fait perdre une vie et un nombre considérable de points, et **`NonToxique`** (**`Perle`**), qui, elle, lui donne une vie et un grand nombre de points.

Le joueur possède 5 vies en début de jeu. Nous avons choisi de définir son **`Niveau`** dans une classe statique : en effet celle-ci ne possède que deux méthodes, `getNiveau()` et `augmenteNiveau()`, que nous avons choisi de définir statiques puisque ne dépendant d'aucune instance (**`Niveau`** ne peut être instanciée).

Enfin, le plateau de jeu est en fait un tableau d'éléments de mers (**`ElemMer`**) instancié dans la classe **`Mer`**. Cette classe regroupe toutes les méthodes qui construisent le jeu (initialisation du tableau de jeu, mise à jour avec les déplacements -non contrôlés par le joueur- des instances de **`Poisson`**, déplacement par le joueur de l'instance de **`Shuting`**).

La classe instanciant **`Mer`** -et donc nous permettant de jouer- est la classe **`JeuFeedingShuting`**.
Le jeu se joue depuis le terminal, grâce à la commande :

```
javac *.java && java JeuFeedingShuting
```

GIANG Cécile (3530406), ZHANG Shuting (3772752)

COMMANDES DE JEU :

Le titre du jeu s'affiche pendant 2,5 secondes puis les commandes s'affichent. Pour les faire défiler, appuyer sur la touche Entrée. Pour déplacer Shuting :

* En haut: touche 5
* En bas: touche 2
* À gauche: touche 1
* À droite: touche 3

Pour valider un déplacement, appuyer sur la touche Entrée.

TABLEAU RÉCAPITULATIF DES CLASSES :

TABLEAU	-	Dans **`Mer`** : tableau de **`ElemMer`** (tabMer), qui sert de tableau de jeu
ARRAYLIST	-	Dans **`Mer`** : `ArrayList` de proies et de prédateurs de **`Shuting`**
HÉRITAGE	-	* **`Poisson`**, **`Nourriture`** héritent de **`ElemMer`** * **`Shuting`**, **`Nemo`**, **`Tuna`**, **`WhiteShark`**, **`BigWhale`** héritent de **`Poisson`** * **`Toxique`** et **`NonToxique`** héritent de **`Nourriture`** * **`Piranha`** hérite de **`Toxique`** * **`Perle`** hérite de **`NonToxique`**
ABSTRACTION	Méthode	Les méthodes apparaissant dans les interfaces **`Chasser`**, **`etreChasse`** et **`seDeplacer`**
	Classe	Les classes **`Poisson`** et **`Nourriture`**
STATIQUE	Variable	* Dans **`Shuting`** : `score`, `nbVies` * Dans **`Niveau`** : `niveau`
	Méthode	* Dans **`Shuting`** : `getScore()`, `getNbVies()` * Dans toutes les classes filles de **`Poisson`** : `getNbMax()` * Dans **`Niveau`** : `getNiveau()`, `augmenteNiveau()`
	Classe	* La classe **`Niveau`**
INTERFACE	-	Les classes **`Chasser`**, **`etreChasse`**, **`seDeplacer`**
EXCEPTION	-	Dans **`Mer`** : `InterruptedException` et `ArrayIndexOutOfBounds`
AGRÉGATION	-	**`Mer`** est composé de **`ElemMer`**