ARIFETTE Nassim
AUBRY Pierre

RAPPORT FROGGER – IPO

Pour tenter de bien comprendre le code qui nous a été donné, nous avons décidé, chacun de notre côté, de faire la première partie. Celle-ci n'a pas posé de problème et nous l'avions tous les deux finis sans difficulté lors de la toute première séance de TP.
Cela a permis à chacun de bien comprendre le code donné et ainsi à postériori d'avoir une compréhension plus poussée du jeu.
Nous avons donc choisi d'utiliser la même stratégie pour les parties 2 et 3.
Lors de la partie 3 quelques soucis ont été rencontrés. Lorsqu'on appuyait sur la touche "haut" pour monter sur la ligne suivante, graphiquement les voitures montaient aussi mais elles ne montaient pas réellement, dans le sens où on pouvait mourir alors que graphiquement on était sur une voiture.
Pour régler ce problème nous avons décidé d'afficher les voitures par rapport au score, nous avons utilisé la même méthode pour les cases spéciales.

```
private void addToGraphics(){
   for (int i = 0; i < length; i++) {
       game.getGraphic()
               .add(new Element(leftPosition.absc + i, leftPosition.ord -
(game.mode == GameMode.Infini ? game.score : 0) , name),depth);
   }
```

Pour la partie 4, nous avons décidé de traiter chacun des éléments proposés dans la consigne (Chronomètre, Cases spéciales et Lignes d'eau).
La mise en place du chrono n'a pas posé problème.

Pour l'ajout de cases spéciales, nous avions tout d'abord décidé de faire une classe *CaseSpeciale* puis d'avoir des classes filles Trap, Slide, etc.
Or on s'est rendu compte que c'était peu utile et que de toute façon les effets n'étaient pas gérés dans ces classes.
Nous avons décidé par la suite de factoriser tout ça en utilisant un Enum regroupant tous les éléments que l'on utilise. L'*Enum* est composé d'un attribut *name* qui correspond au nom du fichier qui correspond au sprite de l'élément qu'on ira chercher par la suite dans nos dossiers ((cet aspect est développé dans la partie 5) ; puis d'un index correspondant au z-index, c'est-à-dire à la profondeur .

ARIFETTE Nassim
AUBRY Pierre

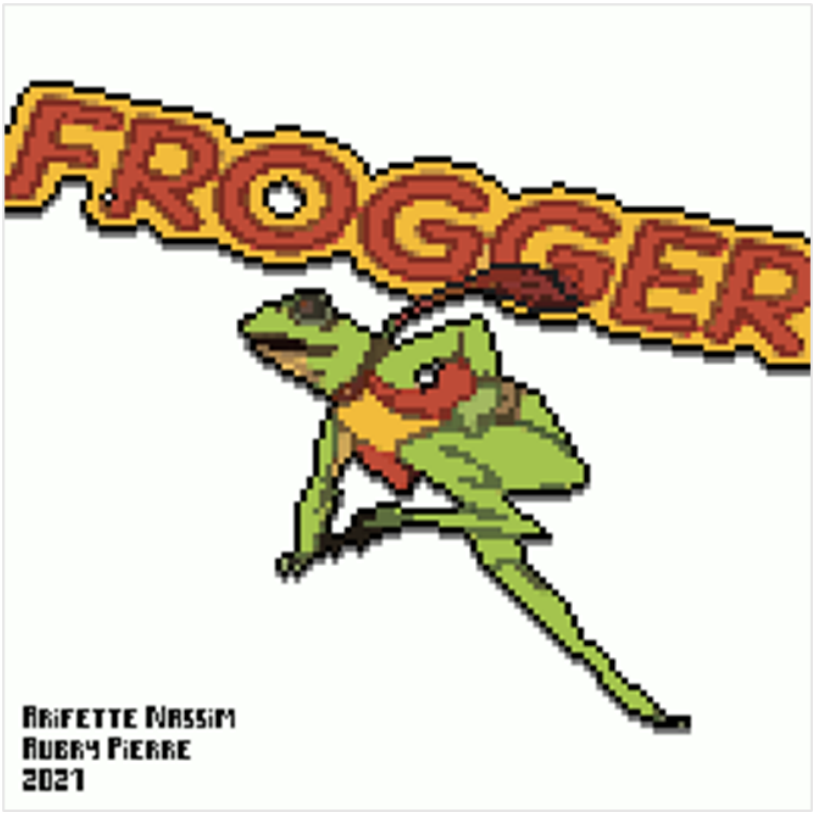
```
NenupharRight("nenupharRight",2),
NenupharLeft("nenupharRight",2),
TurtleLeft("turtleLeft",2),
TurtleRight("turtleRight",2),

FrogPlayer("frogger",3);

private final String name;
private final int zorder;
ElementEnum(String name, int zorder) {
   this.name = name;
   this.zorder = zorder;
}
```

En effet, nous avions eu quelques problèmes pour afficher nos éléments puisque lorsqu'ils étaient dessinés ils n'apparaissaient juste pas sur l'écran. En réalité, ils étaient juste en dessous d'autres éléments.
Cela était dû au fait que les éléments étaient dessinés dans l'ordre de création et non par ordre de "profondeur". Ainsi, certaines cases spéciales ou même voiture n'apparaissaient pas aux yeux du joueur.
C'est pourquoi, pour résoudre ce problème, nous avons décidé d'ajouter un attribut *z-order* dans *ElementEnum* et d'ajouter une liste à deux dimensions dans *FroggerGraphics*.
Nous avons alors ajouté les éléments de z-index "0" dans la première case de l'*ArrayList*, puis les éléments de *z-index* "1" dans la deuxième case de l'*Arraylist* et ainsi de suite…
Cela nous a permis de gérer la profondeur de chaque élément, et tout s'est affiché correctement par la suite.

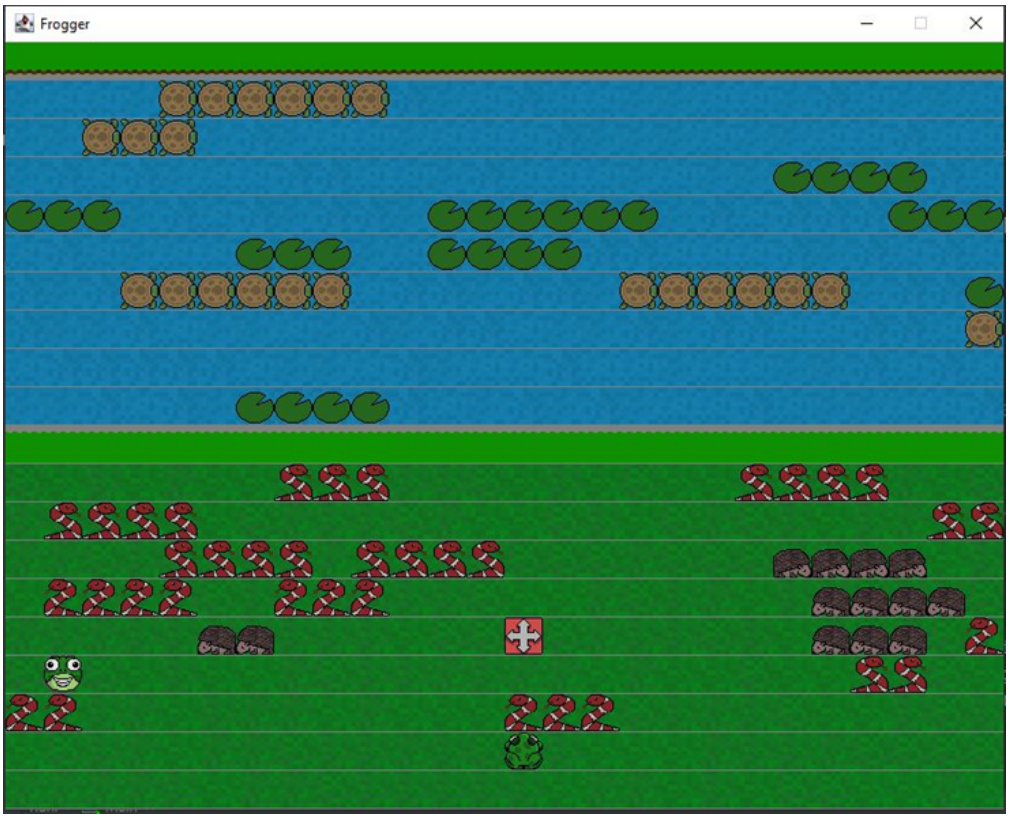
Concernant les lignes d'eau et les rondins, nous avons considéré les rondins comme des cars et juste préciser lors de la création si c'était un rondin ou non, la case *Water* fonctionne comme un piège mais avec un sprite différent.

Pour la partie 5 et l'amélioration graphique nous avons décidé d'attribuer un voire deux sprites par élément.
Cette partie n'a globalement pas posé problème.
Pour l'implémentation nous avons choisis de stocker nos images dans une *static HashMap* que l'on a initialisé au début du programme.
Cette HashMap associe une chaîne de caractère à un *BufferedImage* nous permettant ainsi, lorsqu'on souhaite utiliser une image de ne pas avoir à aller la chercher dans le dossier *ressources* mais directement dans la *HashMap*.
Enfin, nous avons rajouté des effets sonores pour les déplacements de la grenouille et à chaque fois que le joueur perd la partie.

ARIFETTE Nassim
AUBRY Pierre



Ecran de chargement du jeu



Ecran du jeu lors de l'apparition de la grenouille