

Frogger

Par Fabio et Feddy

Partie 1, 2 et 3

Nous avons abordé le début du projet lors du premier TP relatif au projet. Nous avons pu finir la partie 1 et 2, sans rencontrer de problèmes majeurs, lors de cette séance.

Concernant la partie 2, nous avons répartis les tâches de travail pour cet ordre :

- Fabio fait la classe Environment
- Feddy fait les classes Car et Lane

Néanmoins, à la fin de cette partie 2, nous avons un problème avec les apparitions des voitures qui se “collaient”.

Pour la partie 3, Feddy avait fait le contenu requis. Mais un problème concernant l’affichage des voitures, qui continuait d’afficher toutes les lignes en dehors de l’écran à la dernière ligne en bas. Cela nous a poussé à supprimer le code et refaire avec une implémentation plus légère en lignes de code, pour un débog simplifié en contenu à analyser. Nous avons, à la fin de la nouvelle implémentation (Dépourvue de bug) commencé à nettoyer le code pour préparer les parties 4 et 5.

Partie 4

La répartition de la partie 4 fut la suivante :

- Feddy : 4.3
- Fabio : 4.1 et 4.2

Fabio a choisi d’implémenter une interface pour les cases spéciales, mais aussi de mettre la vérification du mouvement sur une case spécial dans la classe Frog, fonction move. Pour la création des cases spéciales dans le jeu, un nombre aléatoire, avec un switch est utilisé. Le 4.1 était un ajout d’une variable avec getter/setter que nous avons ensuite relié à l’écran de mort, pour le temps de survie de la frog.

Le premier problème majeur s’est posé au 4.3, concernant les rondins :

L’idée de base représentait une nouvelle classe, Rondin, similaire à Car. Mais cette idée fut abandonnée par la complexité de l’implémentation. Cela nécessitait de modifier de grosses parties de codes et compliquer la lecture du code (Plusieurs ArrayList, plusieurs cas, plusieurs fonctions pour créer les voitures, etc...)). Après cette implémentation, non sans bug. Nous avons opté pour une solution plus simple : Accomplir une modification de la classe Car, mettre un boolean isRondin et modifier un peu la fonction de détection pour enfin, appeler move sur la frog pour la déplacer. Cela a été plus compliqué à mettre en pratique mais c’était un bon point de départ pour le 4.3.

Nous avons passé plusieurs heures à chercher un bug avec la détection de la grenouille sur le rondin. Après divers problèmes joyeux comme les morts instantanés, le manque de collision, le rondin qui déplaçait 2 fois la grenouille par frame. Nous avons réussi à fixer le bug qui venait de "inBounds" après diverses manipulations des conditions de la fonction.

Après cela, un ancien bug a refait surface. Celui qui concernait la partie 3, qui nous a aussi fait dépenser du temps pour le réparer.

Partie 5

La partie 5 a été faite en majorité par Fabio.

Nous avons débuté par le 5.3, les images. Après diverses recherches sur des sites comme StackOverflow (😊) pour le chargement des images (Etant donné que nous ne savions pas quel élément/classe nous devions utiliser). Nous sommes passés à l'affichage et divers bug sont apparus, le bug le plus mémorable fut l'apparition des images SEULEMENT lors de la mort. Enfaite, le problème était dans les paramètres de la fonction, avec le ImageObserver.

A ce stade de l'implémentation, seulement la grenouille avait une image, mais nous pouvions nous poser la question de l'implémentation pour les cases. Que faire pour les routes qui ne sont pas des cases ? Et pour les voitures, comment déplacer facilement les images ?

Le chemin suivi était de faire une dérivée de case, utilisant l'interface Sprite, interface que nous avons créée avec des fonctions pour récupérer le sprite, et récupérer la position de l'affichage. Pour les routes, nous avons donc créer une SpriteCase par "case" de route, et les classes pouvant utiliser l'interface Sprite (Comme Car et SpecialCase) utilisent cette dernière.

Nous avons aussi dû mettre un ordre d'affichage, sans lequel nous avions les routes qui s'affichait par-dessus les cases spéciales, la grenouille et les rondins, compliqué pour jouer.

Cela fait, ce fut le tour de l'implémentation du 5.1, mais avant cela, nous avons nettoyé le code.

Donc, pour le 5.1, l'idée était de mettre un boolean pour savoir si la frog est premier ou second joueur et de modifier les moves en fonction, et les checks isAlive pour éviter de finir les parties à la mort/victoire d'une grenouille. Nous avons rencontré aucun problème pour cette partie.

Nous avons ignoré le 5.2 pour la suite du projet, A la fin de tout cela, nous avons fixés les problèmes mineurs, améliorer le code en refactorisant et retirant les classes inutiles (comme Element) et modifié la fonction proposée pour la création de voiture (Pour éviter qu'elles se collent en vérifiant seulement la première case)

Hors du projet, nous avons voulu mettre un menu d'accueil et un écran de fin. Cela marchait assez bien, mais la découverte d'un problème que nous n'avons pas réussi à résoudre, qui était que la

création du jeu n'était pas à la bonne taille, malgré plusieurs tentatives, nous avons abandonnés l'idée et juste garder l'écran de fin.