

## Rapport Projet Frog IPO de Thomas Combeau

Tout d'abord, j'ai fait le projet seul car je n'ai trouvé personne à cause du peu de communication à cause du confinement malgré discord qui n'était finalement, pas très actif. J'ai complété le projet jusqu'au début de la partie 4 que je n'ai pas réussi à finir, sur le sujet du Timer (premier sujet de la quatrième partie). Mes parties d'avant contiennent quelques problèmes que je vais développer plus bas après avoir évoqués les problèmes rencontrés que j'ai résolu.

Globalement dans la partie 1 faite en classe de TP, je n'ai pas eu de difficulté particulière qui mérite qu'on s'y attarde donc je ne vais pas plus l'évoquer dans ce rapport.

Pour la partie 2, j'ai pris beaucoup de temps pour la réaliser (pratiquement deux semaines). Dans cette partie, j'ai codé les méthodes de l'interface d'IEnvironment, compléter les constructeurs des classes environment, Lane, Car et après j'ai fait des premiers tests avec du code dans le update() d'environnement. Sauf que j'ai eu un null pointer exception car les classes environment, Lane et Car n'étaient pas reliés dans le même Game. C'est-à-dire que chacun avait son attribut Game puis s'initialisaient dans leurs constructeurs respectifs mais ils ne se reliaient pas avec ce même Game. Ma première idée a été de compléter la liste de Lane de environment dans son update(). Et dans le update je rajoutais des Car dans la liste cars avec la méthode mayAddCar(). J'ai remarqué alors qu'enfin les trois classes étaient reliées mais quand j'ai regardé les listes de cars de chaque Lane, elles revenaient à une taille 0 après chaque update() de la classe environment. C'était parce que j'initialisais les lanes dans ce update() donc avec chaque update() elles se recréaient. Alors j'ai ensuite trouvé la solution de mettre cette initialisation dans le constructeur de la classe environment et cela marchait car il n'y avait plus qu'une initialisation et dans le update() j'avais juste la mise à jour de l'état de chaque lane.

Ensuite, j'ai eu un problème sur l'apparition de mes Cars car c'était soit elles « clignotaient » sur place et avançaient toutes les 3 secondes environ (car c'était la valeur donnée dans le main), ou soit j'avais des « bandes » qui s'avançaient ensemble toutes les 3 secondes aussi. J'ai alors, au fil de plusieurs essais, organisé le update() de Lane en créant des méthodes comme sensCirculation() et en gérant le timer avec l'attribut speed de Lane. J'ai laissé d'autres méthodes que j'avais créées pour des tests mais que je n'ai pas gardé ensuite dans le update() de Lane.

Ce sont les principales erreurs dont il m'a fallu du temps pour trouver les solutions pour cette partie 2.

Pour la partie 3, j'ai d'abord eu beaucoup de difficulté pour imaginer la conception. J'ai toujours eu l'habitude des créations de classe guidée mais comme on code des choses de plus en plus abstraites, je dois réussir à bien imaginer les conceptions des classes qui les composent.

J'ai donc décidé de compléter les interfaces en reliant FrogInf à l'interface IFrog et en reliant EnvInf à IEnvironment. J'ai donc dû mettre les méthodes que j'ai ajoutées dans l'interface correspondante (l'en-tête) et dans les classes Frog et environment aussi car elles implantent elles aussi les méthodes de l'interface correspondante.

En ce qui concerne les erreurs dont j'ai trouvé la solution dans cette partie, j'ai d'abord eu du mal à trouver comment déplacer l'écran. Après plusieurs idées j'ai essayé de créer des méthodes qui décalent l'ensemble des lanes dans la classe EnvInf puis de les appliquer dans la méthode move(direction) de FrogInf. C'est-à-dire que pour la direction up, au lieu d'un déplacement de grenouille, c'est l'ensemble des lanes qui se décalent d'une case vers le bas. Pour la direction down, c'est l'inverse, l'ensemble des lanes se décalent vers le haut. J'ai alors rencontré un autre problème qui est que mes Cars qui étaient déjà affichés ne suivaient pas leur lanes respectives quand elles devaient monter ou descendre. J'ai donc créé des méthodes dans Lane et Car pour que donc les Cars suivent leurs lanes respectives, et puis je les ai ajoutés dans le move de FrogInf selon Up et Down.

Pour la partie 4, je n'ai réussi qu'à ajouter un chrono dans le mode pas infini du jeu et dans le cas de victoire seulement. Je n'ai pas réussi pour le reste des cas car le chrono continuait d'augmenter même après que la partie soit perdue. Pour la méthode utilisée, j'ai ajouté un attribut chrono (un entier) dans Game que j'ai initialisé à 0 dans le constructeur de Game et à chaque fois qu'il traversait la méthode testWin() de game il augmentait de 1 à tous les update() de game donc normalement toutes les 0,1 seconde. J'ai ensuite divisé par 10 ce nombre et je l'ai affiché pour me donner un résultat en seconde.

Pour les erreurs rencontrées que je n'ai pas su corriger, il y a dans, la partie deux et trois, souvent deux voitures, voir plus, qui apparaissent coller, je n'ai pas trouvé l'erreur car normalement ce n'est pas censé arriver avec la méthode mayAddCar() qui ajoute une voiture seulement si la première case et l'avant-première case sont safe, c'est-à-dire sans voiture.

Et aussi parfois, j'ai ma grenouille qui ne meurt pas en rencontrant une voiture mais ça arrive aléatoirement quelques fois donc je n'ai pas trouvé la source du problème.