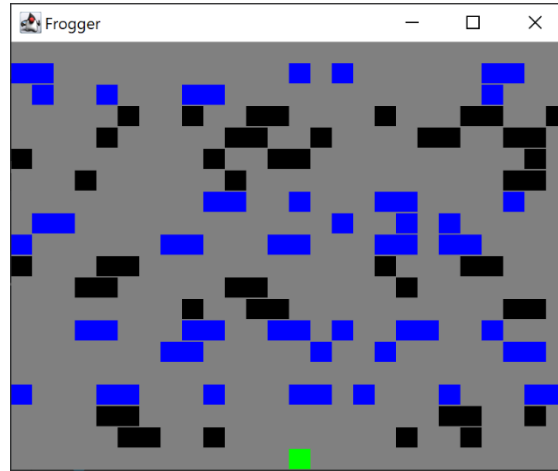


Compte rendu du projet Frogger



Avancement du projet

Parties 1 et 2 fonctionnelles ;

Problème d'affichage dans la partie 3 : l'environnement est censé être infini, mais lorsqu'on bouge la grenouille la « fenêtre de jeu » ne bouge pas en même temps tel que dans le [InfiniteFroggerExample.jar](#).

Le timer dans la partie 4 non fonctionnel tel que demandé, il s'affiche dans le terminal et non sur l'écran de fin ;

Ajout de fonctionnalités : nouvelle partie de jeu si on clique sur la touche « enter » et quitter le jeu si on clique sur « escape ».

Problèmes rencontrés

Partie 1

Liaison de git à IntelliJ compliquée au début mais s'est soldée par le fonctionnement du push et du commit.

Liaison du projet au GivenEnvironment, réussi grâce à l'aide du professeur.

Partie 2

Problèmes d'affichage lié à des méthodes « update () » dysfonctionnelles et à une incompréhension de quelles méthodes affichaient le jeu. Nous avons tout d'abord exploré la globalité du projet afin d'identifier le rôle de chaque méthode fournie et nous avons demandé l'aide d'un camarade en ce qui concerne les méthodes que nous ne comprenions pas. Nous avons fait des tests sur les méthodes update jusqu'à comprendre ce qu'il fallait finalement y mettre. A cause de cela nous avons perdu énormément de temps.

L'affichage des routes se faisait de manière aléatoire et n'affichait jamais toutes les routes à chaque exécution du projet à cause d'erreurs dans la méthode update.

Certaines voitures ne bougeaient pas après avoir affiché le jeu, on a donc cherché où était le problème et c'était l'initialisation de la vitesse qui n'allait pas.

Problèmes mineurs lors de la création des méthodes testWin, l'écran final de gain de partie ne voulait pas s'afficher et après une étude approfondie le problème venait d'une mauvaise condition de isWinningPosition.

Partie 3

Le jeu est bien infini mais l'affichage ne se fait pas comme il faut : lors de l'avancée de la grenouille, la fenêtre ne la suit pas. Elle disparaît donc après la 19ème route. Nous pensons qu'il faut créer une méthode dans EnvInf qui pour chaque position de la grenouille modifiera les lanes qui seront affichés dans la fenêtre en utilisant l'affichage des limites et en créant un nouveau constructeur.

Partie 4

Le timer ne s'affiche pas là où il faut, nous n'avons pas su lier le nouveau constructeur de game à main : il s'avère que même avec une initialisation de game.tm (= string constructeur de game) dans la fonction main, la valeur de ce constructeur n'est plus la même à la sortie de la fonction main, elle n'a pas été enregistrée et nous ne savons comment lier les deux variables. Nous avons donc fait afficher le timer dans le terminal.

Le Timer ne s'arrêtant pas après la perte du jeu malgré l'utilisation de timer1.stop(), nous avons créé une ArrayList de strings dans laquelle nous ajoutons les valeurs du timer à partir de la perte du jeu, la première valeur de la liste est donc le temps de jeu avant de perdre. Le timer n'est pas compatible avec un restart du jeu via la touche « entrer ».

Complément

Nous avons décidé de permettre à l'utilisateur de relancer le jeu si la touche « enter » ou « espace » sont appuyées et aussi lui permettre de quitter le jeu en enclenchant la touche « escape ». Pour cela, dans la classe FroggerGraphic nous avons rajouté plusieurs cas dans la méthode « keyPressed ». Pour les touches « enter » et « espace », lorsque le jeu est fini (gagné ou perdu) et que l'on enclenche celles-ci, alors on relance le jeu avec startGameScreen (pour l'affichage) et game.play(). La méthode play() a été ajoutée dans Game et permet de lancer véritablement une partie (elle crée une grenouille et celle-ci est incluse dans le panel, c'est-à-dire graphic, avec graphic.setFrog()) et qui réinitialise le score à 0. D'autre part nous avons ajouté un attribut de type booléen isRunning dans la classe FroggerGraphic (qui vaut false initialement) qui nous permet de savoir si le jeu tourne ou non (true dans startGameScreen et false dans endGameScreen). Et pour « escape », on envoie au frame un message signalant la fermeture de la fenêtre.

Conclusion

Nous avons perdu énormément de temps au démarrage du projet et à réparer des erreurs d'inattention, nous ne sommes donc en mesure de rendre un projet complet et parfaitement fonctionnel. Mais ce projet aura été très bénéfique pour nous entrainer à coder et réviser pour l'examen. Il nous a également permis de mieux prendre en main IntelliJ.