

## **Compte rendu:**

### **I- Nom des membres du binôme :**

Aurelien MANGIALOMINI 15 heures  
Simon HANNETEL environ 15 heures

### **II- Introduction**

Le but de ce TP est de coder l'igél aergern, un jeu de stratégie ainsi que certaines extensions. L'avantage de le coder est qu'on peut facilement inventer de nouvelles règles et implémenter pleins de modules pour personnaliser et améliorer le jeu. En changeant quelques variables globales, on pourrait aussi modifier l'affichage et la taille du plateau en fonction de nos préférences, jouer à beaucoup ou peu de joueurs..

### **III- Les fichiers**

Le code est séparé en trois grandes catégories:

-> les .h permettant d'accéder aux prototypes des fonctions facilement, assez pratique pour le débogage et pour utiliser le code de son binôme.

-> les .c contenant le code des fonctions

-> les tests qui permettent de vérifier que tout fonctionne comme prévu.

Le fichier test.c du répertoire test contient le main qui exécute simplement toutes les fonctions tests puis lance une partie. On a fait le choix de commenter les tests dans la version finale du projet mais on pourra les décommenter pour vérifier qu'ils fonctionnent.

Nous allons maintenant décrire les différents modules implémentés(.h et .c) et leur utilités:

affiche\_plateau: fonction d'affichage d'un plateau, prenant un plateau d'une partie en cours et affichant tout les détails comme sur l'énoncé du TP

game: fonction qui lance une partie, en enchaînant des rounds jusqu'à que la condition de fin de partie soit finalement remplie.

globales: un fichier regroupant toutes les variables globales du projet, afin de simplifier la modification des paramètres du jeu comme les extensions choisies ou la taille du plateau.

pile : une implémentation de la structure de pile à l'aide de tableau, nécessaire au stockage des hérissons dans les cases.

plateau: un fichier permettant de gérer la création et libération d'un plateau, la structure de plateau retenue est simplement une matrice de cases, le reste des informations étant contenues dans les variables globales.

rounds\_utils: toutes les fonctions utilitaires lorsqu'on exécute un round, savoir si un joueur a gagné, savoir si un mouvement est autorisé, bouger un hérisson. On a aussi la fonction void flush\_line() qui permet lorsqu'on utilise la fonction getchar pour récupérer les informations rentrés par le joueur de s'assurer que le tampon est vide et donc qu'on ne prenne pas des caractères pour les sauts de lignes ou de fin de ligne par exemple.

rounds: gestion du déroulé d'un round notamment la communication entre le joueur et le programme pour qu'un joueur dont c'est le tour puisse effectuer ses actions.

Enfin, 3 extensions ont été implémentées, qu'on peut activer ou désactiver dans globales:

->doping\_variant: le premier hérisson arrivé est éliminé

->tube variant, on peut accéder à la première ligne depuis la dernière et de la dernière ligne depuis la première

->anarchohedgehog : on peut faire bouger verticalement les hérissons de tous les autres joueurs et pas seulement les siens

## Synthèse:

Le projet était intéressant pour se familiariser avec github. On a aussi pu découvrir d'à quel point il était important de modulariser le projet et de déclarer tout dans des .h pour mieux se coordonner. On a aussi appris de nos erreurs de débutant ; il faut faire attention lorsqu'on compile et qu'on fait des push qu'on a bien fetch la dernière version et que tout nos anciens .o ont bien été supprimés par exemple.

## Bibliographie:

<https://rogerdudler.github.io/git-guide/> : bonne aide pour se souvenir des fonctions clés de github.