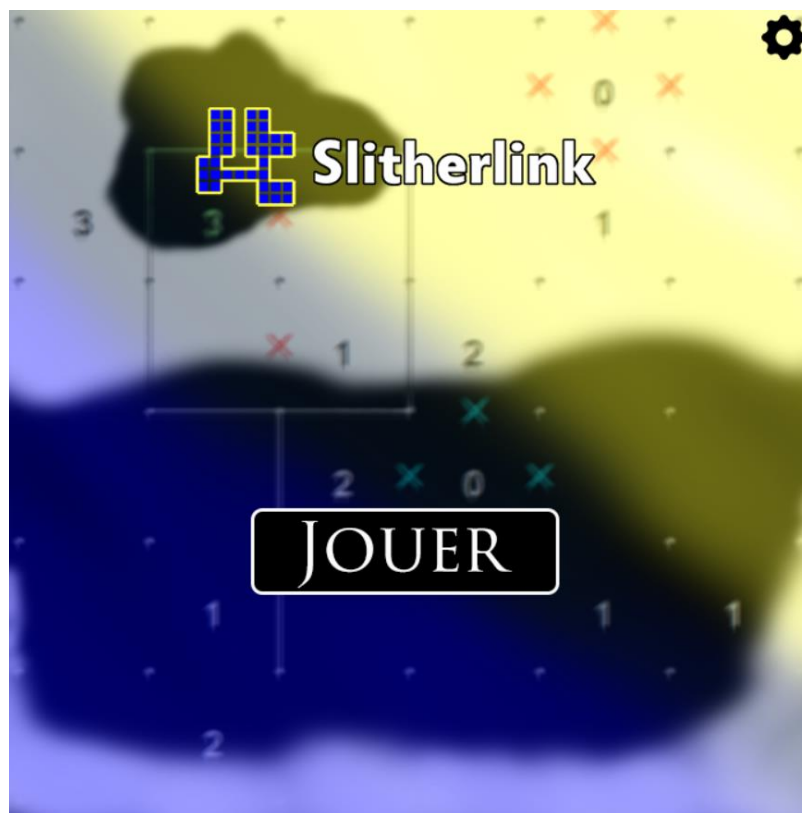


Le SlitherLink



Introduction

L'objectif de mon projet :

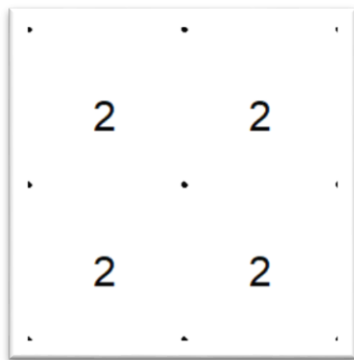
Dès l'annonce du projet nous nous étions mis à chercher comment faire les différentes tâches qui nous étaient demandées. La réflexion était très courte, car les consignes faisaient en sorte que ce projet soit un projet guidé. Un jeu de logique ! ENCORE ?! Ne vous en faites pas, nous adorons !!

Nous n'avions pas seulement pour but de créer un jeu simple. Mais de créer tout un univers, avec des centaines de grilles, pour pouvoir jouer et entrainer un maximum son cerveau.

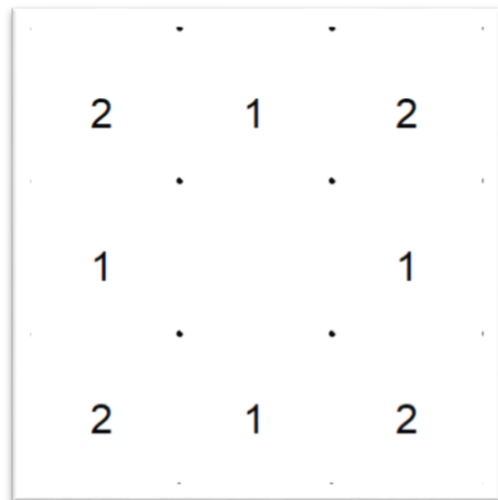
Nous avons aussi créé plusieurs difficultés, que nous retrouvons dans le menu « **Jouer** ». Il existe 4 difficultés... En réalité 3, car une des difficultés est représentée par le niveau trivial qui vous servira, pour apprendre à jouer au jeu. Les 3 autres difficultés sont « Facile », « Moyen » et « Difficile ».

Comme pour tout jeu créé par des étudiants, Nous avons pris soin de rajouter quelque chose d'inutile... UN MENU REGLE... (accessible avec l'engrenage). Pas besoin d'aller dans ce menu, car vous trouverez seulement la définition Wikipédia du jeu SlitherLink.

La taille du plateau de jeu s'adapte. C'est-à-dire que la taille du plateau de la grille est définie selon son nombre de cases.



Cases : 2x2



Cases : 3x3

Guide d'utilisateur :

Lorsque vous lancez le programme une fenêtre va apparaître (voir l'image en première page), une fois sur cette page que l'on va appeler **le menu principal**, trois choix s'offriront à vous.

Le premier, faire un clic gauche de la souris sur la croix rouge en haut à droite du menu principal ce qui fermera le jeu. (Pas très pratique si vous voulez jouer)



Le second, vous pouvez faire un clic gauche de la souris sur l'engrenage en haut à droite du menu ce qui vous amènera dans le menu d'information, une page où il est écrit la définition Wikipédia du jeu pour comprendre en quoi consiste ce jeu et comment y jouer.

Pour revenir au menu principal vous pouvez faire un clic gauche de la souris sur la flèche en haut à gauche du menu ou alors vous pouvez quitter le jeu en faisant un clic gauche de la souris sur la croix en haut à droite du menu d'information.



Le troisième (le plus important), vous pouvez faire un clic gauche de la souris sur le rectangle « jouer », ce qui vous amènera sur le menu paramètre qui est utilisé pour choisir la difficulté de jeu. Une fois dans ce menu vous pouvez quitter ce menu à l'aide de la flèche (toujours en haut à gauche cela ne change pas) et vous pouvez aussi fermer le jeu, toujours à l'aide de la croix. Mais bon, tout cela on le sait maintenant, nous ce qu'on veut c'est JOUER !!!



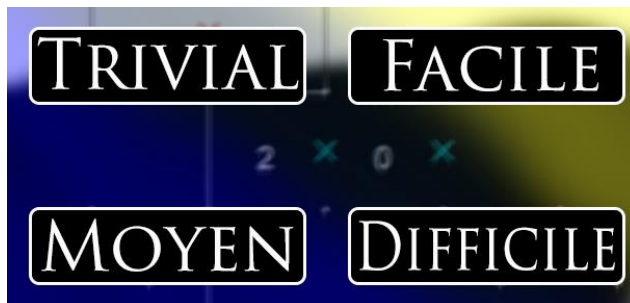
Alors pour jouer c'est très simple une fois dans ce menu paramètre vous aurez le choix entre 3 difficultés et le mode « Trivial » qui n'est ni plus ni moins le mode pour apprendre le jeu.

Si vous ne connaissez pas du tout Slitherlink nous vous conseillons de commencer par le mode Trivial qui vous permettra de comprendre le jeu.

La différence entre les trois niveaux de difficultés est la taille des grilles.

Mais comment jouer ?

Et bien rien de plus simple, une fois votre difficulté choisie (Facile, Moyen ou Difficile).



Il vous suffit de tracer des segments pour former une boucle qui va satisfaire tous les chiffres écrits sur la grille.

Comment fait-on pour tracer les segments et que veut dire satisfaire les chiffres ?

En vérité c'est très simple, pour tracer les segments, il vous suffit de faire un clic gauche de la souris entre deux points cela tracera un segment rejoignant ces deux mêmes points (pas de segment en diagonale, seulement horizontales ou verticales bien sûr).

Pour effacer un segment que vous avez fait, il vous suffit juste à refaire un clic gauche de la souris sur ce même segment.

Bon maintenant vous savez tracer des segments et les effacés mais vous n'êtes pas plus renseignés sur cette histoire de « satisfaire les chiffres ».

« Satisfaire les chiffres » veut tout simplement dire que si le chiffre est un « 1 » il faut qu'il y ait un seul segment autour de ce « 1 », si c'est un « 2 » alors il lui en faudra deux et si c'est un « 3 » je pense que vous avez compris qu'il en faudra trois.

Super maintenant vous savez parfaitement jouer au jeu, mais comment savoir si vous avez **gagné** ou **perdu** ?

Si vous avez gagné le jeu le détectera automatiquement. Il s'affichera donc un message « **Vous avez gagné** » sur votre fenêtre de jeu. (Comme les choses sont bien faites.) Mais si vous avez perdu.... Et bien comme vous pouvez effacer vos traits vous ne pouvez pas vraiment perdre, si ce n'est que perdre la boule après n'avoir trouvé aucunes solutions en 4 heures de jeux.

Petit BONUS, en faisant un clic droit entre deux points cela affichera non pas un segment mais une croix rouge, cette croix rouge vous servira comme rappel, pour vous dire qu'à cet endroit je ne dois pas mettre de segments.

De plus il y a aussi le mode « sauvegarde », en appuyant sur la touche « S » de votre clavier vous pouvez activer le mode « sauvegarde ».

A partir du moment où le mode « sauvegarde » est activé les segments se tracent en bleu et si vous réappuyez sur la touche « S » de votre clavier alors tous les segments en bleu disparaîtront. Ce mode sert à tester des solutions.

Voilà, désormais vous savez jouer à Slitherlink, alors bon jeu !

Le programme :

L'état d'avancement :

Le projet est fini à environ 80%. A vrai dire, il nous manque seulement le système de solveur à faire. Enfin, nous avons bien entamé le programme permettant de résoudre une grille de SlitherLink. Mais nous avons rencontré plusieurs problèmes. Comme le fait qu'on fasse trop d'appels récursifs.

Améliorations éventuelles :

Ce qui pourrait être amélioré est l'interface graphique pour en faire un beaucoup plus jolie, mais n'étant pas des graphistes professionnels il est difficile pour nous de faire une interface qui ferait plus « professionnelle » cela étant nous restons quand même fières de celle que nous avons pu produire.

Nous avons aussi apporté une amélioration au programme SlitherLink. Cette amélioration est le point de sauvegarde. Grâce à la touche de clavier « S ». Vous pourrez en quelque sorte utiliser un cahier de brouillon pour résoudre la grille.

Démarche de programmation :

En simplifiant comme fonctionne notre programme ?

Dans le programme principale « slitherlink.py » une boucle principale du jeu permet de faire fonctionner le jeu. Cette boucle va appeler toutes les fonctions présentes dans le programme « function.py », pour ensuite faire fonctionner le programme. Grâce aux fonctions, nous initialisons aussi les variables principales...

Les variables principales :

Dans le programme « variable.py » on initialise les variables principales « game », « parametre », « jouer » et « info » en « True » ou « False ».

Lorsqu'elles sont en « True » c'est qu'elles sont actives, par exemple on initialise directement en « True » la variable « game » car elle est utilisée dans le menu principal du jeu tandis que la variable « jouer » qui n'est utilisée seulement lorsqu'on clique sur le rectangle « jouer » elles sont donc en « False » au lancement du jeu.

C'est dans le programme « main.py » que la valeur « True » ou « False » sera changé selon si le programme en a besoin ou non.

De plus on initialise les listes de difficultés « Facile », « Moyen » et « Difficile », chacune de ces listes contiennent plusieurs « grille.txt ». Ces fichiers sont les grilles de jeu au format .txt qui vont par la suite être transformé en véritable grille par les fonctions qui gèrent l'interface graphique.

On initialise aussi « taille_case », « taille_marge » qui vont nous servir pour l’affichage graphique du jeu.

Une des variables (dictionnaire) la plus importante est la variable « Etat ». Celle-ci stocke tous les segments ainsi que s’ils sont tracés, interdits ou vides.

Les fonctions importantes :

Dans le programme « function.py » nous pouvons retrouver toutes les fonctions faisant fonctionner le programme. En premier lieu il y a toutes les fonctions qui servent à créer et faire fonctionner l’interface graphique des menus du jeu.

Ensuite nous avons les fonctions les plus importantes :

- [Les fonctions « clic x » et « clic y »](#)

Ces fonctions servent à gérer les clics gauches, et droits à l’intérieur du jeu. Il en existe deux car, une d’elles est utilisée pour les abscisses et l’autre pour les ordonnées.

- [La fonction « carte creation » et « grille creation »](#)

Ces deux fonctions sont complémentaires. Elles sont utiles au tout début du programme, pour la création et la mise en texte des fichiers textes contenant les grilles du jeu SlitherLink.

- [La fonction « statut case »](#)

Statut_case est super importante car elle est utilisée pour vérifier si chacun des indices est vérifié.

Il y a trois possibilités :

- 1 - La fonction retourne 0. Ce qui veut dire que l’indice est satisfait.
- 2 - La fonction retourne 1. Ce qui se traduit par le fait que l’indice peut encore être satisfait.
- 3 - La fonction retourne -1. Dans ce cas l’indice ne peut plus être satisfait.

Informations complémentaires :

Au cours du développement du jeu SlitherLink, nous n’avons pas rencontré de problème majeur. Jusqu’au moment où nous avons dû commencer à faire le Solveur (la tâche 4).

Dans notre tête la manière donc le solveur devait procéder était très clair, mais à mettre sous forme de code ce ne fut pas la même chose. Nous avons donc fait deux solveurs différents, Un des deux fonctionne à peut près, sauf qu’il nous indique une erreur, car trop de récursivité ont été effectué.

La part du travail a été très bien répartie. S’il faut donner un pourcentage, nous sommes à peut près sûres que chacun a fourni 50%.

Ressenti personnel :

Lorsque nous avons lu l'annonce du projet et vu que nous devions faire un SlitherLink notre première réaction fut, « C'est quoi ce jeu ».

Après nous être documentés nous avons très vite compris le principe de ce jeu et comment il fonctionnait.

Nous avons été globalement contents de pouvoir coder un jeu que nous ne connaissions pas cela nous a fait découvrir un nouveau jeu et nous à fait progresser en programmation.

