07/06/2021 ATTIG Léo – SAUTIER Sébastien – TDF

RAPPORT

But du jeu :

Slitherlink est un jeu de type casse-tête, se jouant sur une grille de nombres et d'inconnues, à la manière d'un Sudoku ou d'un démineur. Le but du jeu est de tracer les côtés des cases en respectant les règles suivantes :
•Un numéro dans une case indique le nombre de côtés de la case qui doivent être tracés : ni plus, ni moins
•Une case vide est une absence d'information, le joueur pourra tracer autant de côtés qu'il le souhaite
• L'ensemble des côtés tracés doit former une unique boucle fermée.

Déroulement du jeu :

Le joueur trace des segments en fonction des nombre qui sont affichés ou des croix s'il pense qu'il n'y a pas de segment à placer à cette endroit. Si le joueur arrive à résoudre la grille alors il gagne.

Guide utilisateur :

Lorsqu'on lance le jeu vous pouvez sélectionner une grille ou bien rentrer la votre.
Pour jouer il suffit de faire un clique gauche si vous pensez qu'un segment à ça place ou un clique droit pour afficher un croix si le joueur pense qu'il ne doit pas avoir de segment à cette place. Si le joueur réussi à résoudre le grille alors il gagne. Chaque chiffre correspond au nombre de segment qu'il doit avoir autour de lui, la figure que formera les segments doit être fermé. Si vous gagnez alors s'affiche un écran de victoire sinon ne se passe rien et vous pouvez continuer à chercher une solution.

Les fonctions utilisées :

La fonction *trace_segment* permet de tracer un segment sur le plateau, elle prends en paramètre le dictionnaire « etat » et le tuple « segment ». Si le segment n'est pas dans le dictionnaire état alors 1 est égale à la clé de se segment, si il est dans le dictionnaire on appelle la fonction efface_segment.

La fonction *interdi_segment* permet d'interdire un segment sur le plateau, elle prend en parametre le dictionnaire « etat » et le tuple « segment ». Si le segment n'appartient au dictionnaire « etat » alors celui ci prends la valeur -1 dans le dictonnaire. Sinon on appelle la fonction efface_segment.

La fonction *efface_segment* permet d'effacer un segment du plateau, elle prend en parametre le dictionnaire « etat » et le tuple « segment ». Tout simplement si on l'appelle alors elle regarde si le segment est dans le dictionnaire « etat » et si c'est le cas elle le supprime.

La fonction *efface_TOUTsegment* permet d'effacer tout les segments, elle prends en paramètre le dictionnaire « etat ». La fonction crée un nouveau dictionnaire « etat » vide.

La fonction *gestion_clique* permet de récupérer les coordonnées d'un clique si le joueur clique sur un segment alors il crée le segment ou le retire s'il y en a déjà un, la fonction prends en paramètre « ev » et « tev », « marge » et « tailleCase » qui sont des entiers et le dictionnaire « etat ». On définit x et y puis calcul dx et dy en fonction de x et y. Si on utilise le clic gauche et que « dx-round(dx) » puis « dy-round(dy) » soient dans un certain intervalle alors on trace le segment. Même choses pour le clic droit sauf qu'on interdit le segment grâce à la fonction interdi_segment.

La fonction *recup_segment* renvoie la liste des segments adjacent à sommet dont l'état est « typeSeg ». Si il est égal à 1 alors il est tracé, si il est égal à -1 c'est interdit et si il ne contient pas de valeur il est égale à None alors le segment est vierge

La fonction *statut_case* indique la case est satisfaite si il reste des segments à mettre ou si il y en a trop ! On regarde tout les segments autour d'une case si ils sont dans le dictionnaire alors on ajoute 1 au nombre de segment et on regarde si il est égal supérieur ou inférieur à se nombre.

La fonction *cree_grille* crée une grille à l'aide d'un fichier txt rempli pour ça. Tout d'abord on ouvre le fichier pour le lire, pour chaque ligne on revient à la ligne si char est égal à « _ » alors son indice est égale à None dans le tableau sinon on ajoute « char » au tableau avec comme indice « i » qui gagne 1 à chaque chiffre/« _ »

La fonction *dessine_plateau* dessiné le plateau en fonction du tableau indices d'une marge et d'une de case spécifique.
Cette fonction recréer un fenêtre avec une taille choisie en fonction du tableau indices. On crée un fenêtre en fonction du tableau d'indice, puis on dessine les points en laissant une marge puis les chiffres. Si l'indice est égale à None alors on passe directement à l'étape suivante puisque ça correspond dans le tableau à un « _ »

La fonction *dessine_segment* permet de dessiner des segments sur le plateau. On calcul les coordonnées du segments que l'on veut tracer si typeSeg = 1 comme vue précédemment alors on trace les segments avec les coordonnées calculé si typeSeg = -1 alors on trace un croix rouge pour dire qu'il est interdit de tracer un segment d'après le joueur.

La fonction *dessine_indices* mets les indices au centres des cases. Tout d'abord on regarde si il y des indices et si la case est complété ou non en appelant la fonction statut_case ou l'on comparait le nombre de segment à l'indice.
La couleur est alors choisit en fonction de si le nombre de segment est supérieur, inférieur ou égal. Et on réécrit les indices avec la nouvelle couleur.

La fonction *check_boucle* permet de regarder si la figure dessiner par les point est une boucle. Si la longueur du dictionnaire « etat » est strictement inférieur à 1 alors on retourne « False » sinon la variable « premier » est égal au premier point de la boucle car on sait qu'il y a au moins 1 élément.La variable « checkedSeg » prends la valeur du premier segment. La variable « checkedPoint » prends la valeur du deuxième point du premier segment. Tant que « CheckedPoint » est différent de « premier » « lstSeg » est égal à ce que retourne la fonction recup_segment. Si la longueur de la liste « lstSeg » est différent de deux alors on renvoie « False ».
On supprime le segment qu'on regardait de la liste et « checkedSeg » est égal au nouveau segment. Puis si le point c'est bien le même point alors « checkedPoint » prends la valeur du point suivant sinon il prends la valeur du premier segment. On retourne « True » si ça forme une boucle

La fonction *deter_VICTOIRE* permet de détecter la victoire d'un joueur. On parcourt le tableau en fonction des indices, puis si l'indice est égal à « None » on passe à l'étape d'après, si la fonction statut_case renvoie quelque choses différent de « True » alors on renvoie « False ».
Si check_boucle est différent de « True » alors on renvoie « False ». Sinon on renvoie « True »

La fonction *ecran_fin* permet d'afficher un écran pour dire que le joueur à résolu le puzzle.

La fonction *commande* permet de créer des commandes comme par exemple si l'on appuie sur la touche « q » le jeu se ferme.

La fonction *click_menu* gère les cliques du menu. Suivant on l'on clique on action sera associé comme l'ouverture du fichier « aide.txt »

Le code principale :

La variable menu est égale à « True » le menu s'affiche et on entre dans un boucle pour attendre un clic sur une grille pour que la partie se lance.
On appelle la fonction cree_grille avec le fichier txt qu'on veut utilisé comme grille et on dessine le plateau.
Ensuite on attend un événement si on utilise les cliques gauches ou cliques droit qui corresponde à la croix et au segment alors on appelle la fonction gestion_clique avec les événements et ensuite hors de la boucle on dessine les segments avec la fonction dessine_segment.
Si deter_VICTOIRE est égale à « True » une fenêtre de victoire s'affiche.

Répartition du travail :

Nous avons dans un premier temps commencer le projets dans les grandes lignes ensembles on écrivant les docstring puis commencer à écrire le codes. Léo à ensuite

commencer le rapport quand Sébastien avançait sur le projet. Nous avons ensuite fait le solveur ensemble car ça représentait une grosse partie du projet

Difficulté rencontré :

On pouvait créer des demie segments sur les bords.