

Rapport

Partie Utilisateur:

Le jeu des “Grille d’entiers” à pour but de trouver des sommes importantes dans une grille d’entiers, dans n’importe quelle orientation, afin d’atteindre le score de 100.

Lors de l’exécution du programme, il faudra lui passer trois paramètres, une option d’affichage, le choix de chargement/ création de la grille, ainsi qu’une suite de caractères (nom du fichier sauvegardé contenant une grille ou nom du fichier de sauvegarde potentielle d’une nouvelle grille).

Options d’Affichage:

- **ASCII (a) :**

La partie se déroulera sur le terminal.

A chaque tour, il faut choisir l’une de ces trois options:

- **c : continuer**

dans le cas qui suit, on entre manuellement les coordonnées des cases que l’on souhaite choisir (ligne [espace] colonne). Ces coordonnées doivent être valides (former un segment horizontal vertical ou diagonale dans la grille). Le joueur marque la somme des points et les entiers disparaissent et sont remplacés par les entiers des cases supérieures jusqu’au haut du tableau, où des valeurs nouvelles apparaissent.

- **s : sauvegarder**

La grille, le score, et la valeur maximale attribuée sont sauvegardées dans un fichier portant le nom passé en paramètre lors de l’exécution du programme.

- **q : quitter**

Quitte le jeu.

- **Graphique (g) :**

La partie se déroulera sur une grille dessinée par la bibliothèque graphique .

- Dans ce cas ci, on clique sur deux cases consécutivement pour former un segment. Si ce dernier est valide, Le joueur marque la somme des points et les entiers disparaissent et sont remplacés par les entiers des cases supérieures jusqu'au haut du tableau, ou des valeurs nouvelles apparaissent.

- **Bouton sauvegarde:**

La grille, le score, et la valeur maximale attribuée sont sauvegardées dans un fichier portant le nom passé en paramètre lors de l'exécution du programme.

- **Bouton quitter:**

Quitte le jeu.

Choix de chargement/ création d'une grille:

- **Création d'une nouvelle grille (c) :**

Le programme génère une nouvelle grille de nombres aléatoires avec la taille et la valeur maximale fournies.

- **Lecture d'une grille déjà existante(l) :**

Le programme charge la grille contenue dans le fichier portant le

nom passé en paramètre afin de continuer une partie commencée précédemment.

La commande d'exécution sera donc de cette forme pour un affichage ASCII et la création d'une nouvelle grille, dont le nom sera `new_grid` :

```
>grilleEntiers -ca new_grid
```

Et de celle-ci pour un affichage graphique et la lecture d'une grille pré-existante dont le nom est `old_grid`:

```
>grilleEntiers -gl old_grid
```

Partie Développeur:

Nos structures:

- **typedef struct {}Game:**

Cette structure contient toutes les données essentielles du jeu tel que le tableau NxN et sa taille, ses mêmes informations sont sauvegardées dans le fichier si l'on ouvre le programme en mode création. Elle est nécessaire dans presque toutes les fonctions. La définir dans `my_struct.h` permet de ne pas avoir à rappeler sa définition au début de chaque fichier.

- **typedef struct {}Win :**

Contrairement à Game, cette structure ne contient que des informations graphique (d'où son nom Win pour Window) comme les tailles de fenêtre et de boutons. Ces informations n'étant pas utilisées dans tous les fichiers, elle figure dans `graphique.h`.

La modification (en particulier l'ajout ou la suppression) dans le code de l'une des variables qu'elle contient serait extrêmement fastidieuse si cette structure n'existait pas.

Nos fonctions difficiles:

- modifie_tab:

Le travail de cette fonction se fait en deux temps:

Elle reçoit d'abord, les coordonne de deux points. On ne sais pas dans quel ordre ils ont été sélectionné: peut-être a-t-on cliqué sur la case du bas avant de cliquer sur celle du dessus. Cette fonction commence donc par déterminer les coordonnées de celle au dessus, et de ce celle en dessous. Cela permet de parcourir le tableau dans le bon sens lors des calculs.

Ensuite, on doit déterminer de quel type de segment il s'agit : un segment horizontal, un verticale ou l'un des deux types de diagonal valide. Puis on passe les informations utiles selon chaque cas pour parcourir le tableau, fait le calcul et re-remplir le tableau grâce à notre fonction remplissage.

- coordonnee_graphique:

Cette fonction nous a donné du fil à retordre: comment récupérer 2 point dans le tableau de jeu, un seul lorsqu'on cliquait sur les boutons et ne pas bloquer le jeu?

Il est vite apparu que 2 clics voulaient dire une boucle, de plus, avec les bonnes conditions, cela permet aussi de gérer autant de clics dans le vide que l'on veut. Il a ensuite fallu savoir de quel clic on parlait. le premier ou le deuxième? Il nous aura fallu 3 ou 4 version de cette fonction avant de trouver la réponse.