

Grille-Entiers

Table des matières

Partie utilisateurs.....	3
Projet.....	3
Manuel.....	3
Partie développeur.....	4
<i>Structure du programme</i>	4
<i>Les fonctions principales et leurs rôles</i>	4
Organisation.....	5
Répartition et découpage du travail.....	5
Difficultés rencontrées.....	6
Avantages et apprentissages.....	6
Sources.....	6

Partie utilisateurs

Projet

Le projet est un jeu qui se nomme 'Grille d'entiers' il consiste à trouver des sommes dans une grille d'entiers, dans n'importe quelle orientation pour obtenir un score finale qui est égale à 100.

Manuel

On lance le programme depuis la console.

On écrit dans la console «./grilleEntiers» qui est notre programme principale.

Il faut ensuite ajouter les caractéristique de la partie.

On ajoute les arguments «-a» affichage ASCII (exclu l'option suivante), «-g» affichage graphique (exclu l'option précédente), «-c» création d'une grille (exclu l'option suivante), «-l» lecture d'une grille (exclu l'option précédente).

On lance la ligne sur le terminal.

On nous demande alors le nombre de colonne voulu (inférieur à 20) et la valeur max compris dans les cases (ex: si on choisie 25 alors de -25 à 25).

Après avoir sélectionné les caractéristiques de la partie on appuie sur la touche 'entrée'.

On arrive sur une fenêtre avec un quadrillages, remplie de chiffre le but va d'appuyer sur les bonnes cases afin que le jeu se termine.

Tout d'abord on doit respecter certaine règles afin de pouvoir gagner.

La première consiste a obtenir un score finale de 100.

La seconde règle et que la ligne situer entre les deux cases sélectionner doit être en diagonale, à la vertical ou à l'horizontale.

Le joueur possède deux autres boutons à gauche de l'écran qui vont lui permettre de sauvegarder la partie (création d'un nouveau fichier pour une première sauvegarde) et ou de quitter la partie.

La partie ce finie quand le joueur a bien obtenue un score de 100 ou à appuyer sur le bouton «Quitter».

Partie développeur

Structure du programme

Le programme est séparé en 3 parties :

La première partie correspond à la création de notre jeu en mode ASCII (le jeu se joue dans le terminal).

La seconde partie correspond à la création de notre jeu en mode graphique (une fenêtre est créée pour jouer aux jeux).

Il y a dans la troisième partie le programme principale (main) qui exécute toutes nos fonctions.

Les fonctions principales et leurs rôles

void (jeu)

Cette fonction prend en paramètres :

int taille_plat = nombre de colonnes et de lignes.

int tab_val[taille_plat][taille_plat] = un tableau de tableau qui représente notre grille.

int coord[4] = tableaux de longueur 4 contenant les coordonnées de deux points (x1,y1,x2,y2).

int* somme = le score

int* coups, = le nombre de coups joués

int val_max = la valeur max et min prise par une case

char* argv[3] = tableau de longueur 3 contenant les arguments

La fonction nous permet de regarder si l'utilisateur ne clique pas deux fois sur la même case ou si la ligne entre les deux cases sélectionnées est bien une diagonale, une verticale ou une horizontale. Si ce n'est pas le cas on demande à l'utilisateur de rejouer (le coup n'est pas compté). Sinon on additionne le contenu de chaque case, on les remplace par des variables aléatoires et on les échange avec les cases situées au-dessus (les cases tombent si il y en a une qui disparaissent en-dessous).

void (clique coord)

int taille_plat / int tab_val[taille_plat][taille_plat] / int* somme, int coord[4] /
int val_max / int* coups / char* argv[3]

La fonction va transformer les coordonnées du clique, en coordonnées de case (il remplit le paramètre int coord[4]) et de lancer la fonction void(jeu). Si l'utilisateur clique sur le bouton sauvegarde lance la fonction sauvegarde, si il clique sur le bouton clique ferme la fenêtre et redemande un clique si l'utilisateur clique autre part que la grille ou les boutons.

void (creation tab val c)

int taille_plat / int val_max / int tab_val[taille_plat][taille_plat]

Cette fonction permet de créer un tableau de valeur aléatoire (de valeur maximum et minimum, int val_max) correspondant aux paramètres int tab_val[taille_plat][taille_plat].

int main

int argc = la taille de la liste des arguments

char* argv = la liste des arguments

On regarde tout d'abord si les arguments donnés lors du lancement du programme sont reconnus. Le programme principal va ensuite lancer les différents programmes en fonction des arguments choisis.

Organisation

Répartition et découpage du travail

On a créé la plupart des fonctions ensemble, David c'est un peu plus occupé de la version ASCII et Natan de la version graphique et du rapport.

Mais on a tous les deux aidé ce soit dans le programme est dans la rédaction du rapport.

Difficultés rencontrées

Le fait d'utiliser une bibliothèque graphique donner par la fac sur nos ordinateurs personnelle.

Avantages et apprentissages

Ce projet nous a permis d'apprendre à travailler en groupe, de bien se répartir les tâches que chacun doit accomplir, de savoir s'entraider.

On a appris à bien utiliser les adresses et les pointeurs, on a utilisé tout ce dont nous avons vu en cours (boucle, switch, conditions ternaires).

Sources

On a beaucoup utilisé la bibliothèque graphique MLV donnée par les professeurs pour la confection de ce projet.