

Table des matières

I. Journal de bord	1
07/11/2022	1
11/11/2022	1
12/11/2022	1
13/11/2022	1
14/11/2022	2
15/11/2022	2
20/22/2022	2
27/11/2022	2
26/11/2022	2
28/11/2022	3
II. L'architecture du code	4
intPlus	4
III. Détails de chaque fonction	5
Menus	5
Jeu	5
IntPlus	5
Cartes	5
Main	6
IV. Consigne	7
- Description du sujet	7
- Fonctionnalité du programme	7
- Structure du programme	7
- Explication des valeurs type de la grille de jeu	8
- Algorithme remplissant la grille	9
V. Conclusion	11
Edouard.S	11
Alexis.W	11

I. Journal de bord

07/11/2022

Nous avons créé une fonction qui génère une liste composée de paire de nombre disposé aléatoirement comme dans l'exemple ci dessous.

0,6,4,4,3,0,2,3,1,2,5,1,5

Cette fonction était destinée à être le générateur qui place les cartes aléatoirement sur le terrain.

11/11/2022

Nous nous sommes rendus compte que chaque carte avait besoin de quelques caractéristiques comme:

- leurs coordonnées sur le terrain
- l'état dans lequel est la carte (verso, paire trouvé ou carte choisis)
- le chemin d'accès vers l'image qui correspond à la carte.
- Un numéro qui est propre à la paire de carte

La génération d'une simple liste de nombre ne suffisait donc pas. Nous avons donc opté pour une génération d'une liste de structure (carte). Ainsi chaque élément de la liste contenait toutes les caractéristiques nécessaire d'une carte

Cette nouvelle génération à été divisé en 3 fonctions:

- La première générait une liste contenant toutes les paires de cartes différentes possibles (le même nombre que d'image de cartes différentes)
- La seconde sélectionnait parmi cette liste, des paires de cartes aléatoires qui seront utilisées sur la grille de jeu.
- La dernière fonction, mélangeait les cartes de la listes précédentes et calculait les coordonnées de chaques cartes

12/11/2022

Pour que le joueur puisse interagir avec les cartes, nous avons ajouté une fonction qui prend en argument la position de la souris et qui renvoie la carte sur laquelle la la position de la souris se situe.

Nous avons cherché des images sur internet qui seras notre premier design de jeu

13/11/2022

Nous avons commencez à utiliser l'interface graphique pour ajouter 4 fonctions:

- la première gère toute la partie menu. elle commence par afficher tout ce qui compose le menu. puis gère les interaction avec la souris

- la deuxième, gère toutes la partie jeu. l'affichage de tout ce qui compose l'écran de jeu, le calcul du temps qui s'écoule, le retournement des cartes et le mode triche.
- la troisième, permettait l'affichage du chronomètre sur l'écran de jeu en fonction du temps calculé dans "jeu".
- La dernière est le main qui articule tout le projet et qui le rend fonctionnel.

14/11/2022

La fonction jeu était une très grosse fonction de presque 100 lignes. Nous avons donc décidé de la décomposer en plusieurs fonctions.

Nous avons commencé ce découpage par une fonction qui gère le retournement des cartes et une fonction triche qui change, si nécessaire, l'état du jeu (mode triche ou mode normal)

15/11/2022

Le découpage continue avec une fonction qui gère le temps.

2 bug ont été corrigé:

- Le premier était qu'il était possible de cliquer et de trouver des paires en mode triche. (On pouvait alors finir une grille en 1 centième de secondes)
- Le second permettait l'interaction avec des cartes déjà trouvées.

20/22/2022

Le code à été diviser en plusieurs fichier lié par un makefile

27/11/2022

Une fonctions "bouton dynamique" et une structure "Bouton" on été ajouté au script afin d'avoir des bouton dynamique qui réagissent lorsque la souris passe dessus (en s'éclaircissant par exemple)

26/11/2022

Nous avons modifié le script pour créer des animations lors du retournements des cartes. Malheureusement, cette animation était trop coûteuse et l'ordinateur ne pouvait pas afficher des centaines d'images en un instant très court. Donc cette idée à été abandonnée.

Vous pouvez avoir un aperçu de l'animation du [retournement](#) lorsque le mode triche était activé (il restait quelques bug à corriger)

28/11/2022

Une variable booléen à été ajouté, et qui, si elle est activé, masque les cases déjà trouvées par l'utilisateur. Malheureusement le fait de sauvegarder des images et de piocher dedans pendant le script n'est pas très optimisé, et activé cette variable provoque un manque de performance au début du jeu et à chaque découverte de paire.

II. L'architecture du code

Le code est divisée en plusieurs fichiers :

intPlus

Le fichier intPlus qui ajoute 2 fonction pour manipuler des entiers:

La première permet de compter le nombre de chiffres qui composent un entier (cf)

III. Détails de chaque fonction

Menus

- La structure bouton
- La fonction bouton dynamique
- La fonction affiche bouton
- La fonction sauver bouton
- La fonction menu
- La fonction pause
- La fonction paramètre
- La fonction jeu

Jeu

- La fonction affiche temps
- La fonction calcul temps
- La fonction calcul triche
- La fonction retourne cartes
- La fonction calcul clic

IntPlus

- La fonction len int
- La fonction int to string

Cartes

- La structure carte paquet
Cette structure symbolise une carte ou une paire de cartes qui se trouve dans un paquet de cartes.
Chaque carte contient donc un numéro qui correspond à l'image située sur son verso. Et l'occurrence de la carte dans le paquet. Ce nombre varie donc de 0 à 2. Car il n'y a que deux cartes qui contiennent la même image
- La structure carte table
Cette structure symbolise une carte posée sur un plateau de jeu. Elle contient donc deux coordonnées (x,y),
Un état (0 pour une carte qui n'a pas été sélectionné, 1 pour une carte retourné par l'utilisateur ou 2 si la paire a été trouvé),
Une id qui correspond au numéro de l'image (l'id est unique à chaque paire de cartes),

Et, un chemin d'accès vers l'image que contient la carte

- La fonction fabrique carte

La fonction prend en arguments une liste de bibliothèque d'images, le nombre de bibliothèques qui se trouve dans la liste et une adresse qui contiendra le nombre de cartes disponible dans la totalité des bibliothèques.

Une bibliothèque est une structure "carte paquet". Elle contient dans l'attribut carte, le nombre de cartes contenu par la bibliothèque. Dans l'attribut occurrence la bibliothèque contient 0 ou 2 pour indiquer si les paires de la bibliothèque sont disponibles ou non.

(L'utilisateur pourra choisir d'utiliser une bibliothèque d'images ou non)

Avec ces données, la fonction parcourt dans un premier temps la liste des bibliothèques et ajoute ou non dans la variable (donné en argument) le nombre de cartes situé dans la bibliothèque parcourus, afin de connaître à la fin, le nombre de cartes disponibles.

La fonction (fonction à optimiser pour ne faire qu'un tour)

- La fonction choix carte
- La fonction mélange carte

Main

- La fonction main

IV. Consigne

- Description du sujet

Bienvenue dans notre jeu de mémoire!

Le but est simple. Vous choisissez une grille parmi celles qui vous sont proposées dans le menu. Une fois fait, des cartes apparaissent, face cachée, sur votre écran. Il suffit de cliquer sur l'une d'entre elles pour la retourner et voir l'image qui s'y trouve derrière.

Attention à partir de maintenant un chrono est lancé alors dépêchez vous de lire la suite et de gagner !

Retournez maintenant une seconde carte.

Si les deux images que vous avez trouvé sont identiques, félicitation vous venez de trouver une paire de cartes ! Sinon, essayez de retenir les images qui se trouvent sur ces cartes car elles vont se retrouver automatiquement face cachée. mais surtout, dépêchez vous d'essayer de cliquer sur d'autres cartes ! Le chrono tourne ! Le but est de trouver toutes les paires le plus rapidement possible.

Bonne chance !

- Fonctionnalité du programme

Le programme contient des fonctionnalités supplémentaires. Si vous vous ennuyez déjà sur notre petit jeu, vous pouvez aller dans les paramètres et modifier l'apparence du jeu.

Vous pouvez également choisir, si nos 3 configurations ne vous plaisent pas, le nombre de cartes avec lequel vous allez jouer.

Si vous avez l'esprit de compétition, vous pouvez consulter et essayer d'entrer dans les top10 des parties terminées dans les meilleurs délais (pour une même grille)

- Structure du programme

Le programme qui fait fonctionner notre jeu est divisé en plusieurs fichiers :
boutons.c, menus.c, main.c, cartes.c, jeu.c.

Pour commencer, cartes.c contient toutes les 3 fonctions de génération de cartes (voir plus bas). Elle contient également les structures qui représentent les cartes et les banques d'images de cartes.

Ensuite, bouton.c contient les fonctions qui gèrent l'affichage des boutons. Cela leur permet par exemple d'être dynamiques lorsqu'on passe la souris dessus.

Le fichier menus.c quant à lui, gère le fonctionnement mais surtout l'affichage des menus. Comme les paramètres, l'accueil, ou encore l'affichage de la grille de jeu.

Il existe également jeu.c qui permet de gérer tout le déroulement d'une partie. Que se soit du retournement des cartes au mode de triche, en passant par l'affichage du chronomètre.

Le fichier qui articule les autres c'est main.c
C'est lui qui exécute les premières instructions du programme.

- **Explication des valeurs type de la grille de jeu**

Lorsque vous faites une partie, il y a plein de données que vous ne voyez pas. Voici donc, si vous vous plongez dans notre code, quelques explications.

Commençons avec les booléens.

Vous pouvez constater qu'il existe une variable triche qui contient 1 si le joueur active le mode triche et 0 lorsqu'il le désactive.

La variable ilatriche contient 0 si le joueur n'a pas utilisé le mode triche de la partie et 1 dans le cas contraire.

La variable temps est basé sur le même principe. Lorsque le temps est figé, la variable temps est à 0 sinon, elle est à 1.

La variable cacheTrouve, est fixée à 1 lorsqu'on veut que les paires trouvées disparaissent de la grille et à 0 lorsqu'on veut les laisser face visible. cacheTrouve ne peut se modifier que dans les paramètres.

La variable écran n'est pas présente que dans la phase de jeu. Chaque menu est constitué de cette variable. En jeu, elle est à 1 lorsque vous êtes en cours de partie et à 0 lorsque vous avez terminé.

Pour finir, la variable tour est à 1 lorsque le joueur doit retourner une première carte, et 2 lorsqu'il doit trouver la paire qui correspond.

Il existe aussi d'autres variable qui représente des dimensions :

tCarteX et tCarteY correspondent respectivement à la hauteur, à la longueur d'une carte.

margeX et margeY contiennent l'espace horizontal et vertical entre chaque carte.

nbColonnes et nbLignes correspondent au nombre de colonnes et de lignes de cartes dans la grille de jeu. largeur et hauteur, stockent la taille de la fenêtre.

designCarte et designFonds, contiennent le chemin d'accès vers le fonds de carte et le fonds de Grille choisis par l'utilisateur dans les paramètres.

tempsRetournement contient le temps durant lequel les 2 cartes sélectionnées par le joueur restent visibles. Sur le même principe, dateRetournement contient l'instant où les 2 mêmes cartes doivent être retournées.

durationPartie contient le nombre de microsecondes passées en mode non triche.

Et instantDebut est la durée de la partie présent soustrait à l'instant présent.

La variable numClic contient le numéro de la carte sur lequel le joueur a cliqué (-1 lorsqu'il ne clique pas sur une carte). numCarte1 et numCarte2 contiennent respectivement le numéro de la carte qui a été cliqué au premier et au deuxième tour.

La variable retournement est un peu plus particulière. Lorsque les cartes attendent de se tourner face cachées (car les 2 cartes retournées ne sont pas identiques), le retournement est fixé à 1. Si aucun retournement n'a besoin d'être fait, retournement est mis à 0. En revanche, si les cartes doivent disparaître lorsque

- Algorithme remplissant la grille

Lorsque vous faites une partie, la génération aléatoire des cartes se fait en 3 grandes étapes.

La première est effectuée au lancement du programme ou lorsque le joueur modifie dans les paramètres les bibliothèques de cartes avec lesquelles il souhaite jouer.

A cet instant, une liste est créée. Elle est composée du numéro de toutes les images, inclus dans les bibliothèques choisies par le joueur (ou activé par défaut).

00000	00001	00002	00003	00004	01000	01001	01002	01003	01004
-------	-------	-------	-------	-------	-------	-------	-------	-------	-------

L'étape suivante est effectuée lorsque vous lancez une partie ou cliquez sur rejouer. Lors de cette étape, on connaît le nombre de paires qui seront installées sur la grille. Donc autant de numéros d'images qu'il y aura de paire, sont piochés aléatoirement, (dans la liste précédemment créés) et placés successivement dans une autre liste. Si un numéro d'images a déjà été pris, c'est le suivant qui est sélectionné jusqu'à tomber sur un numéro d'images qui n'a pas encore été pris. Chaque bibliothèque d'images contient plus d'images que de paire dans la plus grande grille possible de créer.

01002	00002	00000	00003	01001
-------	-------	-------	-------	-------

La dernière étape est effectuée directement après l'étape précédente ou lorsque vous recommencez le jeu avec les mêmes cartes.

Des numéros d'images sont successivement choisis dans la liste précédente et placés à la suite d'une nouvelle liste. autant de numéros sont placés qu'il y a de cartes. et si un numéro

de carte à déjà été choisi 2 fois, comme précédemment, on prend successivement le suivant jusqu'à tomber sur un numéro qui n'a pas encore été choisi 2 fois.
 A chaque numéro choisi, on calcule également les coordonnées de la carte qui correspond, le début de son chemin d'accès, son état...

Num	00000	01001	00002	00000	01001	00003	00002	01002	01002	00003
X	20	170	320	470	620	20	170	320	470	620
Y	235	235	235	235	235	385	385	385	385	385
...										

Et en fonction de ces caractéristiques, les cartes sont placées sur la grille

V. Conclusion

Edouard.S

C'était très intéressant de construire un jeu (même si c'est un simple mémoire) car on apprend les petites astuces utilisées pour obtenir ce que l'on veut, notamment en ce qui concerne l'optimisation du jeu.

On remarque que la manière la plus logique de raisonner, pour réaliser une fonctionnalité, est rarement la meilleure ; je pense notamment aux animations des cartes qu'Alexis voulait mettre en place.

Il était aussi intéressant de penser à une interface jolie et intuitive. En tant qu'utilisateur, il est très simple de critiquer les erreurs d'ergonomie alors qu'en créer une de toute pièce est d'une autre paire (:D) de manches.

Alexis.W