



Rapport de projet

PRÉSENTÉ PAR

Mzyene Assia et
Toussaint Nolan

2022

Table des matières

01

Introduction et cahier des charges

02

Description des fonctionnalités du programme

03

Structure du programme

04

Explication du main

05

Algorithme de la grille

06

Conclusion personnelle

JEU DE PAIRES

Le jeu de paires, aussi connu sous le nom de memory est un jeu qui se joue seul ou à plusieurs et dont le but est de retrouver toutes les paires. On possède un certain nombre de cartes qu'on mélange et que l'on place de sorte à ne pas voir les images. On retourne une première carte puis une deuxième. Si les cartes sont similaires, c'est une paire, sinon on les retourne toutes les deux dans leur position originale. Une fois toutes les paires trouvées, le jeu est terminé. Lorsque l'on joue à plusieurs, la personne qui a le plus de paires gagne.

Le programme devra donc, dans un premier temps, être capable d'afficher une fenêtre dans laquelle des cartes s'afficheront sous forme de grille. Le programme devra aussi être capable de vérifier la position des clics et, si le clic correspond à une carte, il devra être retourné. Si c'est la deuxième carte, on testera si elle forme une paire avec la première.

Dans le cas où elles formeraient une paire, on les laissera afficher à l'écran. Sinon, on les cachera au bout d'une seconde.

Dans un deuxième temps, on affichera le temps écoulé en seconde depuis le début du lancement du jeu. On ajoutera également des tailles de grille variable.

Dans un dernier temps, le programme devra arrêter le temps et afficher toute la grille si le mode tricheur est cliqué.

PRÉSENTATION DES FONCTIONS

Cartes1, Cartes2 et Cartes3 permettent d'afficher les images découvertes aléatoirement sur l'écran grâce à un tableau aléatoire ou affiche uniquement le dos de l'image si elle n'est pas retournée.

CaseCliquee1, CaseCliquee2 et CaseCliquee3 testent chaque case grâce aux coordonnées du clic données en argument et renvoie l'emplacement de cette case.

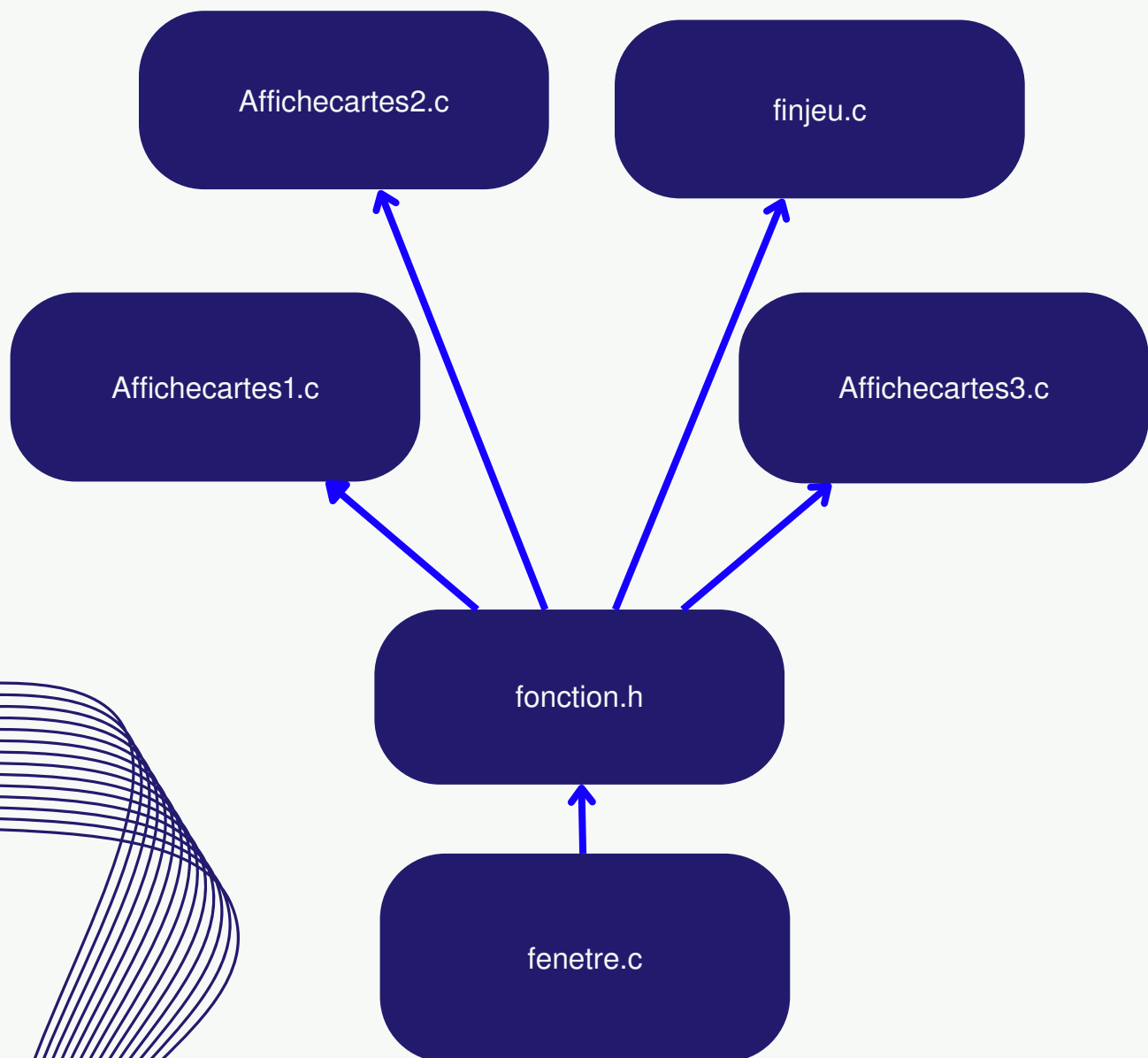
Le programme principal main regroupe les boucles, les conditions et l'affichage graphique des fenêtres.

La fonction temps permet d'afficher le temps en fonction du nombre de secondes données en argument.

La fonction finjeu teste pour chaque case si elle est retournée, si oui, elle renvoie 1.

Decouvrir1, Decouvrir2 et Decouvrir3 font s'arrêter le programme pendant 1 seconde puis retourne les deux cartes cliquées données en argument.

STRUCTURE DU PROGRAMME



s'appuie sur →

EXPLICATION DU MAIN

La construction globale du programme est divisée en différentes parties séparées par des while qui correspondent aux étapes réalisées entre les clics.

L'affichage de l'écran de départ est automatique et unique, il nous emmène directement dans la première boucle désignée par sa position qui affiche la page de menu des niveaux si le bouton start est pressé et nous emmène dans la deuxième boucle.

Celle-ci affiche le fond et la grille si son niveau est sélectionné et nous emmène dans la boucle correspondant au niveau. Chaque niveau est composé de trois boucles while définissant les images à afficher et le temps. La première boucle de chaque niveau teste le premier clic d'une carte à retourner ou l'activation du mode tricheur, ou encore le bouton retour qui renvoie au menu de sélection de niveau.

La deuxième boucle fait de même avec le deuxième clic de carte, le mode tricheur ou le retour. La dernière quand elle teste la validité d'une paire ou la fin du jeu. Si la paire est valide, on retourne simplement à la première boucle du niveau, si elle ne l'est pas, on retourne les deux cartes avec 1 seconde d'attente puis on retourne à la première boucle. Si le jeu est fini, on affiche un écran de remerciements pour avoir aidé un peuple ou récupérer une récompense.

ALGORITHMME DE LA GRILLE

Pour l'affichage de la grille, nous avons utilisé des tableau et des boucles. Tout d'abord, pour le tableau, nous avons créé un tableau dont toutes les valeurs étaient initialisées à 0. Nous avons ensuite parcouru ce tableau de façon aléatoire grâce à la bibliothèque `time.h` et nous avons donné la valeur de l'accumulateur à la case choisi si elle n'avait pas été visitée (si elle était égale à 0). Ainsi, à la fin de la boucle, le tableau était rempli de nombres aléatoire entre 0 et le nombre de cartes divisé par 2.

Ensuite, grâce au tableau aléatoire, on va afficher les images qui ont été indiquées préalablement dans un tableau.

Pour cela, on utilise une double boucle avec `j` étant les lignes et `i` les colonnes, dans lesquelles on teste le visibilité de la carte pour le joueur, donnée en argument grâce au tableau "visibles". Si les cartes sont visibles, l'image est affichée, sinon le dos est affiché, cela évite de voir les cartes pendant la latence entre l'affichage des images et des dos.



CONCLUSION

Grâce à la spécialité NSI, j'avais déjà eu l'occasion de travailler sur des projets similaire mais j'ai trouvé que ce projet était beaucoup plus difficile que ceux sur lequel j'avais pu travailler avant. Le fait que la bibliothèque n'ai pas beaucoup d'exemple et qu'il n'y ait pas d'exemple disponible sur internet fait que j'ai eu beaucoup de difficulté à avancer car il fallait tout comprendre et tester soit même.

Avec Nolan nous avons également manqué d'organisation. Nous avons fini la SAe de SCR assez tardivement, on a donc eu moins de temps pour faire le projet. Mais j'ai pu beaucoup apprendre durant ce projet et j'ai pu revoir des notions que je ne maîtrisais pas totalement en C (comme les pointeurs (même si dans la version rendue nous n'en avons pas utiliser) ainsi que l'utilisation de git). Même si je suis un peu déçue vis-à-vis du programme que nous avons rendu parce que je n'ai pas pu faire tout ce que j'imaginais et j'ai eu du mal presque constamment durant le projet.

ASSIA





CONCLUSION

Comme l'a dit assia précédemment, nous avons prit du retard sur ce projet en raison du projet de SCR. Cependant, je me suis donné les moyens de réussir en travaillant tous les soirs pendant 2 semaines pour avancer sur le projet. Cela m'a énormément aidé à revoir certaines notions comme l'utilisation de GIT, les tableaux, les fonctions et la structure générale d'un programme qui réalise nos besoins.

Pour ma part, j'ai trouvé que la partie graphique était chronophage: la recherche d'image, la modification de taille, l'affichage avec coordonnées qui sont précises et uniques et ainsi de suite. J'ai d'autres part eu beaucoup de mal avec l'affichage du temps, en particulier par rapport à la copie de zones que j'avoue ne pas avoir compris totalement. J'ai également eu un problème avec l'écriture de texte qui ne prend pas les accents.

Cependant, j'ai beaucoup apprécié la programmation des fonctions telles que la détection des clics, la détection de paires trouvées et le retournement des cartes qui m'ont bien fait réfléchir et revoir les mathématiques. J'ai aussi trouvé très intéressant le fait d'utiliser des fonctions externes pour avoir un programme plus clair et de voir la structure du programme avancer avec les boucles et conditions qui forment une base solide pour réaliser presque n'importe quoi à n'importe quel endroit sans difficulté.

