Damien MADEIRA
Félix LEGRELLE

Groupe 4

Memory

Table des matières

INTRODUCTION :

Description du projet :

Le projet est la construction d'un jeu de memory en langage C dans le cadre de notre premier semestre de première année de DUT Informatique.

Le jeu memory est un jeu de carte qui se compose de paires portant des illustrations identiques. L'ensemble des cartes est mélangé, puis étalé face cachée. Vous retournez une première carte puis lorsque vous en retournerez une deuxième deux cas sont possibles :

- Les deux cartes sont identiques, elles restent découvertes jusqu'à la fin de la partie.

- Les deux cartes sont différentes, elles se retourne après un court instant.

Le but du jeu est bien évidemment de trouver toutes les paires le plus rapidement possible.

Les objectifs à réaliser :

Nous voulions que notre projet soit très soigneux, réaliste et amusant. C'est pourquoi nous avons décidé de nous appliquer sur son interface ainsi que sa jouabilité.

Pour nous le jeu devait être esthétique tout en restant simple à comprendre et à utiliser.

Nous voulions aussi que le code soit le plus claire et le plus compréhensible possible. C'est pour cela que nous avons cherché à l'épurer au maximum.

Explication du projet :

Nous avons voulu créer quelque chose d'original et qui se démarquent des projets qui ont été réalisés par nos camarades.

I) Déroulement du projet

Le projet s'est réalisé en 3 grandes étapes :

- Première étape : Création de l'algorithme principale qui nous permet de remplir de façon aléatoire le tableau

- Deuxième étape : Créer tout le contenu visuel du jeu, les différentes pages du memory, les logo des cartes, et tous les petits détails qui sont présents sur l'écran d'un téléphone.

- Troisième et dernière étape : Écrire le code principal ainsi que les fonctions, puis corriger les bugs qui persistaient.

Notre objectif était de finir ce projet pour le vendredi 21 décembre 2018 à 8h00.
Nous avons travaillé régulièrement sur ce projet. Cela ne nous a pas paru contraignant. Nous avons fait ce travail comme un jeu et un défi.

Damien : création de l'algorithme de remplissage.

Félix : création de la charte graphique du jeu.

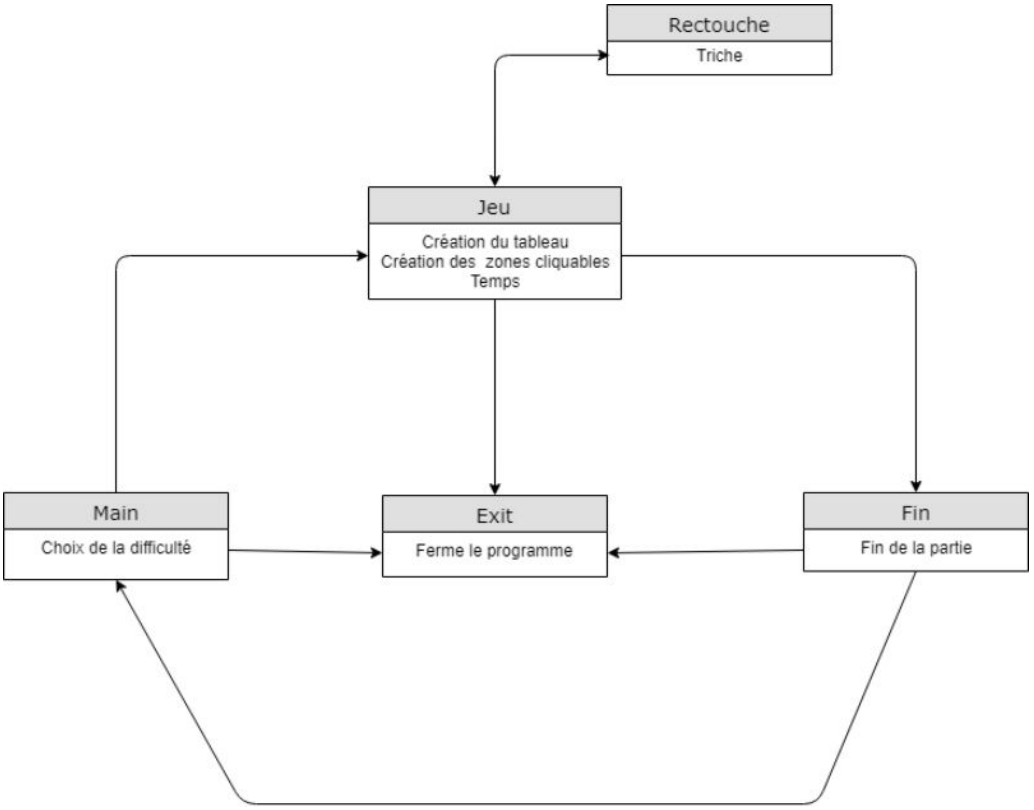
Le groupe entier : création des fonctions et du dossier final.

II) Structure du programme

La fonction la plus utilisée dans ce programme est la fonction "SourisCliquee". Elle se met en route à chaque clique sur une zone cliquable.
Pour savoir si la zone est cliquable ou non, la fonction récupère les coordonnées du clic et si la zone l'est alors elle appelle une autre fonction qui elle va avoir un rôle souhaitée et afficher des images à l'écran.

Ce qui nous parut le plus difficile était de créer une fonction pour que le jeu reconnaisse le clic gauche de la souris puis regarde si les cartes retournées sont identiques. Cette fonction permet donc de retourner les cartes et de regarder grâce aux coordonnées des cartes si les cartes sont identiques ou différentes.
Si elles sont identiques alors la paire trouvée reste retournée jusqu'à la fin de la partie.
Si les deux cartes trouvées sont différentes alors au bout d'une seconde les deux cartes se retournent face cachée.

Diagramme du projet



Comme le montre ce diagramme notre projet était découpé en 4 parties :

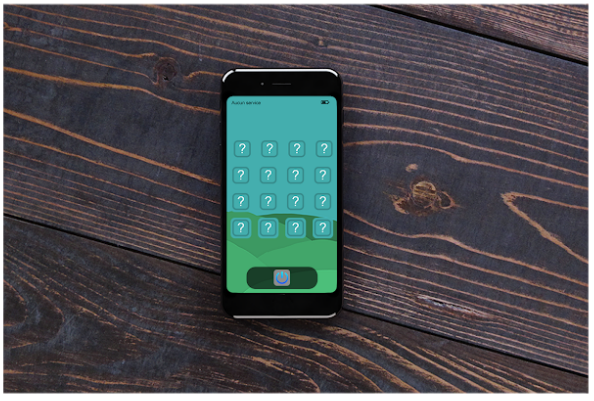
Main : Menu qui propose à l'utilisateur de choisir une difficulté ou de fermer le programme

Jeu : Création d'un tableau avec des paires répartie de façon aléatoire, gestion du temps de jeu, création de zone cliquables avec une valeur assignée, comparaison des valeur suite à 2 cliques et proposition à l'utilisateur de fermer le programme. La fonction "Rectouche" peut se faire appeler.

Rectouche : Lors de l'appuie de la touche "t" de mettre en place un mode tricheur. Il révèle toutes les cartes. Mais lorsque le mode triche est activer le jeu est mit sur pause il lui est donc impossible de cliquer sur les cartes et le temps n'avance plus. Si l'utilisateur appuie de nouveau sur "t" le jeu reprend son cours.

Fin : Lorsque l'utilisateur a fini de jouer il peut choisir de revenir au menu pour recommencer une partie ou quitter le jeu.

Visuel des différents menus du projet



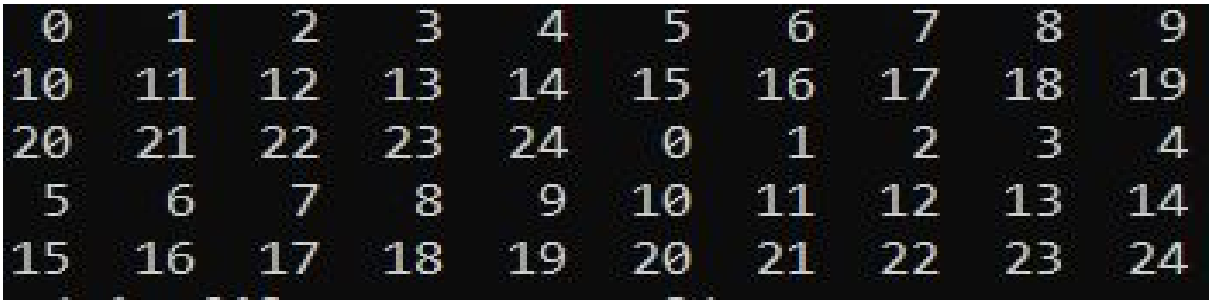
III) Algorithme de remplissage

Nous allons maintenant vous expliquer l'algorithme qui a permit de créer les différentes grilles de jeu.

Nous avons choisis d'utiliser des tableaux multidimensionnels car c'est ce qui nous a paru le plus logique et le plus simple pour réaliser le memory étant donné que le memory est composé d'une grille d'un certain nombre de colonnes et d'un certain nombre de lignes variables.Une fois les tableaux initialisés il a fallu dans un premier temps créer des paires de nombres et seulement des paires, nous avons donc vérifier que le tableau était entièrement remplis de paires de nombres.
Pour cela il a fallu créer une condition (avec un if) qui s'effectue lorsque les valeurs dans le tableau sont supérieurs à la moitié de la capacité du tableau. Ici il y a 25 paires (de 0 à 24) pour un tableau comportant 50 cases.

Nous avons extraits les captures d'écrans de notre niveau difficile qui est une grille de taille 5x10.

Nous pouvons voir le tableau de taille 5x10 qui génère les paires de nombres aléatoires, nous remarquons que les paires ne sont pas répartis aléatoirement.

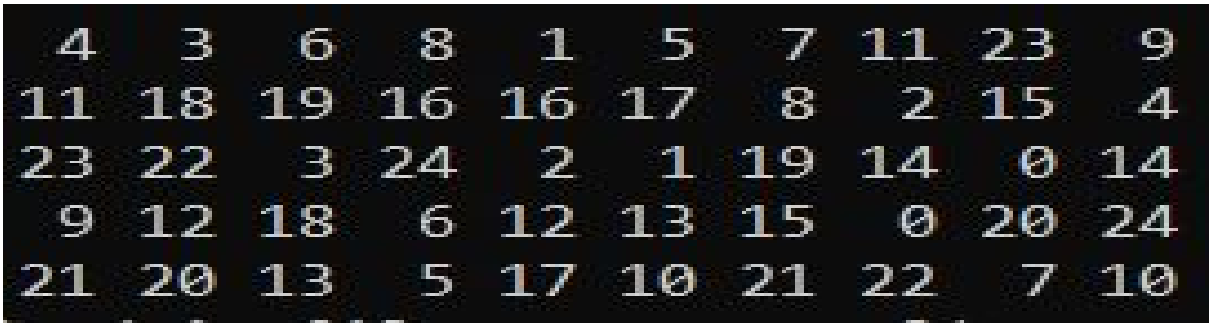


Une fois la première étape validée il a donc fallu mélanger ces paires de nombres aléatoirement. Nous avons choisis de mélanger ces paires en deux étapes:

- mélanger les lignes grâce à une boucle "for"
- mélanger les colonnes grâce à une autre boucle "for"

en mélangeant les cinq lignes et les dix colonnes dans cette exemple nous nous assurons que les paires ne soient plus incrémenter une par une mais bien mélanger dans le désordre.

Nous voyons sur l'image ci-dessous que les paires sont maintenant bien mélangés:



IV) Conclusion personnel

Félix : Le projet est conforme à mes attentes. Je suis satisfait du résultat visuel et du résultat des fonctionnalités du jeu, même si des bugs ont été rencontrés et que certaines de nos attentes ont dû être supprimées.

Plusieurs améliorations auraient été possibles comme :

- Permettre à l'utilisateur un plus grand choix de difficultée
- Un classement fonctionnel
- Animer davantage le jeu
- Améliorer le graphisme
- Permettre à l'utilisateur de personnaliser un niveau de façon complète. Il pourrait choisir le nombre de case et le fond du jeu.
- meilleur intégration du temps

Cette deuxième expérience de "gros" projet m'a permis d'apprendre le travaille de groupe et de découvrir une façon de penser qui a été un peu différente de celle que j'avais l'année dernière en Python. J'ai aimé le fait de mettre en application mes connaissance de façon à créer quelque choses de concret. J'ai pris beaucoup de plaisir à faire ce projet.

Damien : Au début le projet m'a semblé assez compliqué vu que je n'avais jamais fait de gros programme en informatique contrairement à mon camarade Félix. Ce projet m'a appris à m'organiser dans mon code, jusqu'à maintenant nous avons effectués que des petits programmes ou nous pouvons directement codés le programme sans vraiment m'organiser mais avec ce projet nous avons dus nous organiser en plusieurs petites étapes intermédiaires. Ce projet m'a aussi permis d'apprendre à travailler en groupe et de communiquer. Une fois le projet terminé je me suis dit que le projet n'était pas si compliqué que ça. Le fait de pouvoir jouer à son propre jeu nous montre vraiment comment utiliser nos compétences que nous avons acquis jusqu'à là. Au final j'étais très content de faire ce projet car il a permit de valoriser nos connaissances et compétences