

SANCHEZ Jean-Baptiste

CRAMARÉGEAS Maxence



Le Memory

Table des matières :

Introduction

I- Fonctionnalités du programme

II- Structure du programme

III- Explication de la Grille de jeu

IV- Algorithme de distribution de la grille

Conclusion

Introduction

Dans ce compte-rendu, nous allons vous détailler le projet que nous avons effectué en binôme.

Il faut savoir que ce projet est notre premier projet de programmation. Ce projet consiste à créer un jeu : Le Memory. Plusieurs contraintes s'imposaient à nous, comme le fait d'utiliser la bibliothèque graphique de l'IUT ainsi que le langage C.

Ce projet nous a permis d'utiliser d'une nouvelle manière les connaissances acquises au premier semestre sur le langage C, et de les appliquer via un projet avec un objectif concret : un jeu en utilisant la bibliothèque graphique et nos programmes de TP.

Nous allons donc ici vous présenter en détail, comment nous avons construit et codé notre jeu.

I- Fonctionnalités du programme

Le menu principal est appelé grâce à une fonction menu qui affiche sur l'écran 0 un fond personnalisé avec les boutons :



La page Info présente les règles du jeu. Cette page est accessible en cliquant sur le bouton INFO. Ce clic est géré par la fonction sélection. Pour quitter cette page, il suffit d'appuyer sur une touche puis on relance le jeu :



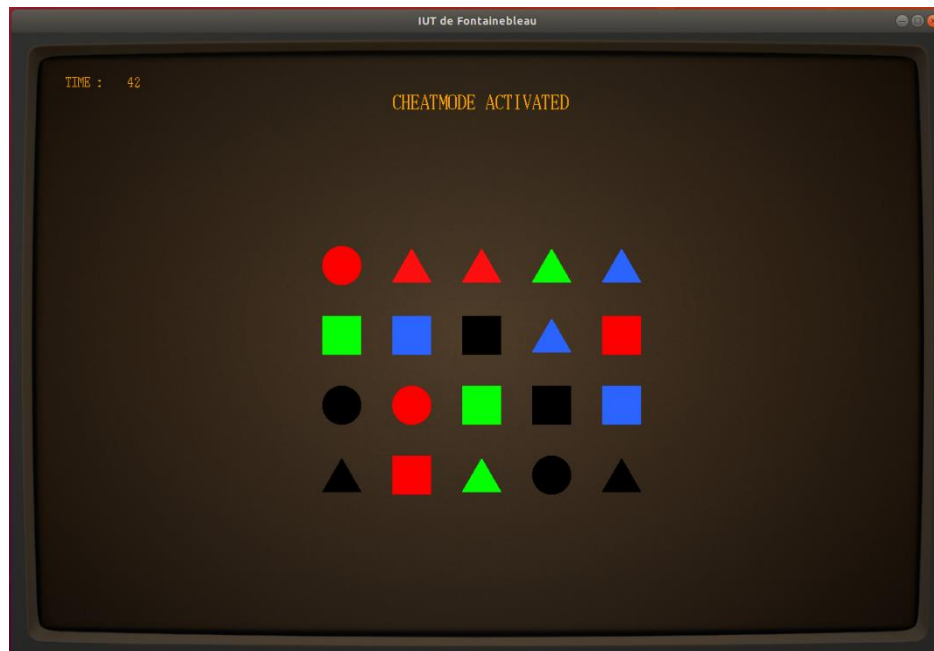
Ici, on a l'écran de sélection de difficulté. Il fait suite au bouton JOUER du menu principal. Ce bouton est géré par la fonction SELECTION :



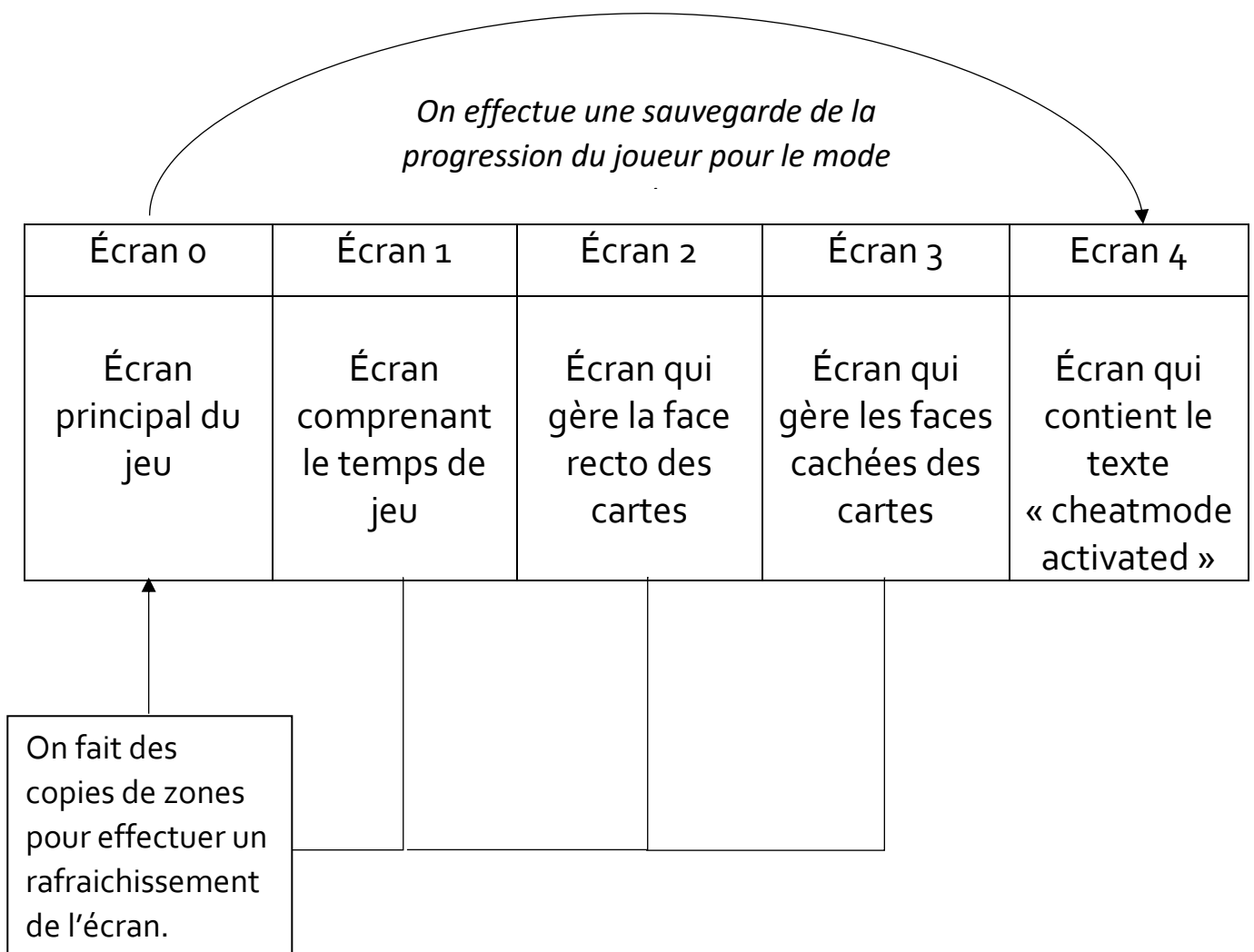
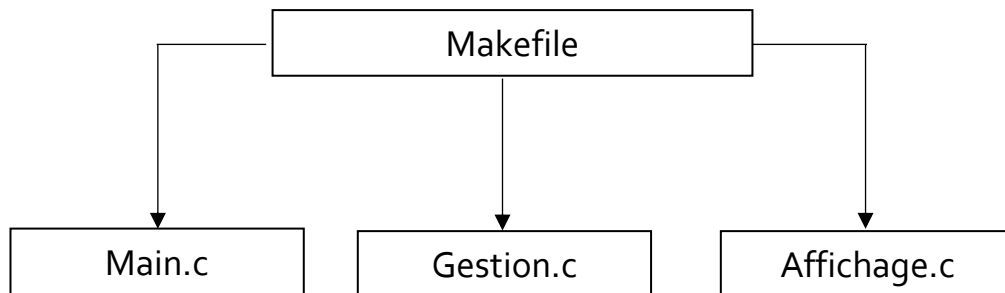
Ici, on a l'écran de jeu (en mode facile) qui nous présente une grille de cartes, faces cachées. On accède à la grille du jeu grâce à la fonction SELECTION qui nous renvoie à une valeur de difficulté qui va déterminer la taille de la grille, de la fenêtre et des marges ? qui vont être appelés avec la fonction Grid. Grâce à la fonction Updatecard on peut retourner deux cartes en cliquant dessus et les comparer afin de déterminer si c'est une paire ou non. Si on a complété notre grille au complet, on accède alors à un écran de victoire grâce à une condition IF dans la boucle WHILE du jeu. Le temps écoulé en seconde est affiché durant la partie en haut à gauche (lui-même est géré par une condition IF dans le WHILE en permettant via un écran virtuel de rafraichir le temps).

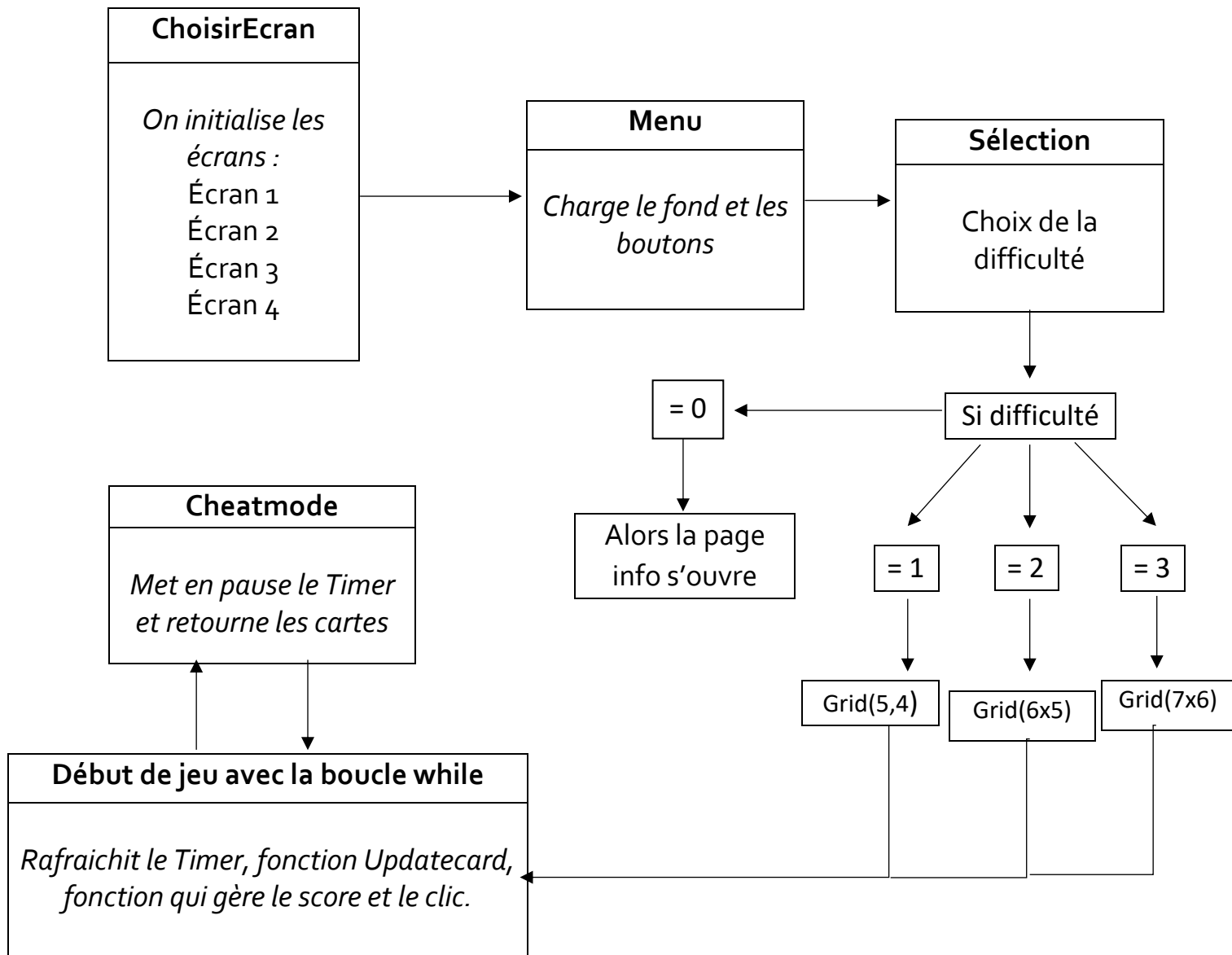


Écran de jeu : lorsque le mode triche est activé, ce mode triche s'active avec la touche « t ». Il arrête le temps et il dévoile l'ensemble des cartes de la grille. la fonction qui est appelée pour exécuter cette tâche est Cheatmode.



II- Structure du programme





III - Explication de la Grille de jeu

Dans cette partie, nous allons vous expliquer, comment nous avons généré une grille de jeu qui respecte les contraintes d'un Memory.

En effet, dans un Memory, les cartes doivent être mélangées avant d'être distribuées.

Nous avons donc décidé de commencer par reproduire ce mélange de cartes.

Pour ce faire, nous avons créé un tableau à allocation dynamique, qui sera contenu dans le pointeur « randcard ». Celui-ci, a pour but de stocker l'ensemble des clés de cartes en fonction de la taille de la grille de jeu.

Tout d'abord, nous avons dû introduire nos cartes dans un tableau d'entier comportant tous les chargements de sprites nécessaires (les sprites correspondent à toutes les faces retournées des cartes du jeu). Puis on a dû remplir le tableau à allocation dynamique de zéros en fonction du nombre de paires.

Ensuite, nous avons généré une clé d'identification pour chaque paire de cartes afin de les discriminer pour déterminer lorsque le joueur trouve une paire ou non.

Ces clés d'identification s'insèrent dans notre tableau randcard et prennent une valeur en fonction de l'horloge de l'ordinateur ce qui permet de simuler en quelque sorte le hasard.

IV- Algorithme de distribution de la grille

Précédemment, nous vous avons détaillé la structure des données, ainsi que la façon dont nous avons généré nos paires de cartes afin de les distribuer de manière aléatoire, et de pouvoir discriminer les paires de cartes entre elles pour détecter quand le joueur trouve ou non une paire

Il faut savoir que les paramètres de notre fonction Grid nous permet d'appeler des grilles de tailles différentes selon la difficulté choisie par le joueur. Cela va donc distribuer les cartes faces cachées au centre de notre écran.

Puis nous avons dû placer les cartes mélangées sur la grille.

C'est pour cela que nous nous sommes servis de notre tableau à allocation dynamique randcard pour distribuer les clés d'identification des cartes sur toute la grille. Il était donc nécessaire que notre algorithme puisse parcourir la grille, afin de placer les cartes faces cachées sur l'écran principal, mais aussi sur l'écran 3 dans le but de remettre les cartes faces cachées si le joueur n'a pas trouvé sa paire.

C'est pour cela que nous avons utilisé deux boucles FOR qui nous sert à se déplacer en ligne puis en colonne pour distribuer les cartes sur toutes les places de la grille de jeu.

Conclusion Personnelle

de Jean-Baptiste SANCHEZ

En conclusion, je peux dire que ce projet a été révélateur et très instructif. En effet, je trouve qu'il m'a appris beaucoup de choses et a permis d'exercer ma logique algorithmique ainsi que ma façon de résoudre les problèmes d'une manière concrète et en groupe.

Le fait d'avoir ce côté graphique apporte un tas de nouveaux problèmes à résoudre, forts intéressants. De plus, en voulant faire les choses d'une certaine manière par soucis d'esthétisme cela a créé de nouveaux problèmes qui complexifient l'écriture du jeu. Cela rend particulièrement intéressant la recherche de solutions pour atteindre l'objectif. Ce qui a été déterminant aussi, fut l'organisation des codes sources et des brouillons. Même si on est organisé, on peut vite s'y perdre dans ses brouillons ainsi que dans nos fonctions. Il faut donc savoir gérer ce qu'on fait, et comment on le fait, ainsi que s'aider des outils comme le dépôt GIT créé pour l'occasion.

Finalement, j'ai énormément travaillé sur l'ensemble du programme et j'en suis heureux car j'appréhende dorénavant d'une manière différente les problèmes. Par exemple, même si j'ai compris la notion de pointeurs, j'ai eu un peu de mal au début à bien les gérer entre nos différentes fonctions. Le projet m'a permis d'appréhender les problèmes et certaines notions de diverses manières. Au fil du projet, je me suis senti plus familier avec l'environnement graphique. Pour finir, je suis assez enthousiaste et exalté pour élargir mes connaissances dans le domaine. Cela a attisé mon envie de découvrir

et j'ai pu imaginer quelques possibilités de projet à écrire en langage C.

Notre jeu respecte toutes les contraintes imposées. Même s'il paraît simple j'en suis fier personnellement car il n'a pas été si facile à coder. On pourrait dorénavant s'amuser à créer des nouvelles fonctions pour l'améliorer, comme une fonction de retour au menu pour éviter de devoir quitter le jeu ou encore créer d'autres modes. Mais encore modifier dans un menu OPTION la grille de jeu à la guise du joueur. On pourrait essayer d'alléger au maximum le code aussi par exemple ou de rendre plus poussé l'interface homme/machine.

Enfin je suis heureux d'avoir mené à bien ce projet et d'avoir complété ce qu'il y était demandé. Ça a été vraiment instructif et j'ai bien envie d'améliorer le travail fourni ou bien d'essayer de faire d'autres jeux en poussant le niveau un peu plus loin.