NGUYEN Hoang Khuong

BRUN Eliott

# SAé DEV 1.1

Ce projet consiste à créer un jeu de paires, avec un compteur, un mode tricheur et plusieurs grilles de difficultés.

## Fonctions :

Généralité : Les images utilisée pour les cartes sont nommés de 1.png à 24.png pour faciliter l'appel des emplacements des images, exemple :

```
for (ligne=0;ligne<4;ligne++) {
    for (colonne=0;colonne<4;colonne++) {
        if (grille[ligne][colonne]!=0){
            sprintf(str,"./images/%i.png", grille[ligne][colonne]);
            sprite=ChargerSprite(str);
            AfficherSprite(sprite,432+(124*colonne),162+(124*ligne));
            LibererSprite(sprite);
        }
    }
}
```

Le tableau 2D grille ne contient que des entiers.

### CreerGrille() :

Cette fonction est l'une des plus complexes fonctions à coder, on a créé cette fonction pour but de Créer une grille de carte, qui est stocké sous forme de tableau 2D, afin de pouvoir récupérer ce tableau pour le manipuler après il fallait créer une fonction qui renvoie un double pointeur

```
int** CreerGrille(int* tab, int nbligne, int nbcolonne) {
    /*Fonction qui permet de mélanger les cartes de façon aléatoire et les stocker dans un tableau 2D
    Prend en argument un tableau, le nombre de ligne et colonne (nbligne*nbcolonne=longeur du tableau)
    Renvoie un tableau 2D*/
    srand(time(NULL));
    int** grille = calloc(nbligne, sizeof(int*));
    for (int c = 0; c < nbcolonne; c++) {
      grille[c] = calloc(nbcolonne, sizeof(int));
    }
```

On a découvert qu'une fonction ne peut pas renvoyer tableau 2D normale (exemple : int tableau2D[ligne][cologne]; ), on a dû le créer avec des calloc().

Elle prend en argument un tableau tab(ce tableau est rempli des cartes qui ne sont pas mélangées), nombre de ligne et de colonne, ligne*colonne doivent être égale à la longueur du tableau tab.

Ensuite elle va enlever les valeurs de tab aléatoirement une par une et ranger dans la grille 2D.

```
int i;
int image;
int ligne;
int colonne;
int taille =(nbcolonne*nbligne)-1;
for (ligne=0;ligne<nbligne;ligne++) {
    for (colonne=0;colonne<nbcolonne;colonne++) {
        if (taille>=1) {
            image=rand() % taille;
        }
        else {
            image=0;
        }
        grille[ligne][colonne]=tab[image];
        for (i=image;i<taille;i++) {
            tab[i]=tab[i+1];
        }
        taille--;
    }
}
return grille;
```

Pour créer ce tableau, nous avons codé ce morceau de code qui se trouve dans la fonction ChargerFacile() et ChargerMoyen() qui crée un tableau des entiers de 1 à n/2 (dans cet exemple n=16) et chaque entier présent 2 fois dans le tableau.

```
int i;
int f;
int cpt=1;
char str[50];
int tab[16];
int** grille;
for (i=0; i<16; i=i+2) {
    tab[i]=cpt;
    tab[i+1]=cpt;
    cpt+=1;
}
grille=CreerGrille(tab,4,4);
```

PremiereCarte() :

Une fois la grille est affichée et la partie a commencé, le programme appelle cette fonction qui déclenche le chronomètre et attend un clic de souris, si le clic est cliqué sur une carte cachée, alors on tourne la carte et appelle la fonction DeuxiemeCarte().

DeuxiemeCarte() :

Fait exactement la même chose que la fonction PremiereCarte(), sauf qu'elle vérifie aussi si les deux cartes tournées sont identiques, si oui alors les cartes restent dévoilées, et la valeur de cette carte dans la grille devient 0 (comme rien). Sinon les cartes restent affichées pendant 1 seconde.

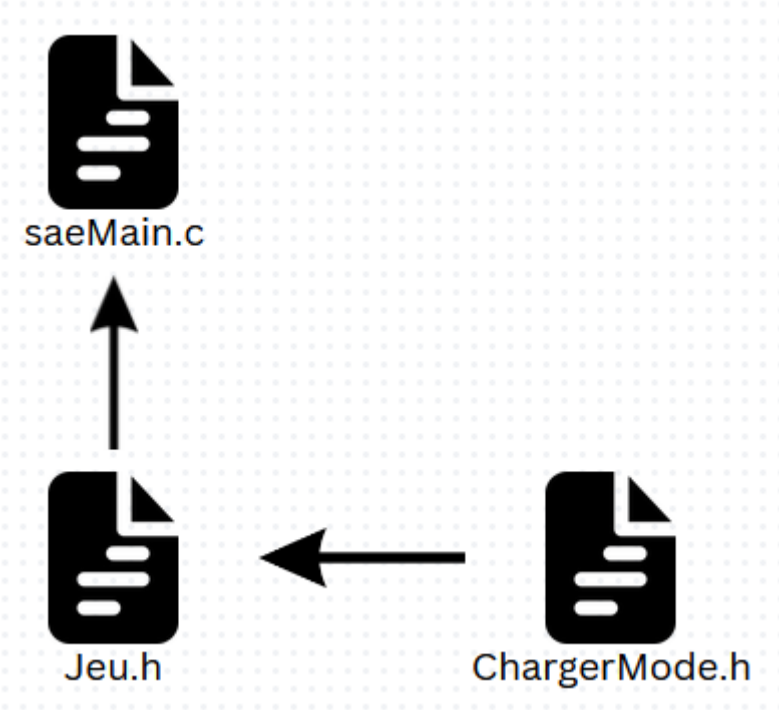
GrilleComplete() :

Une fois que tous les valeurs de la grille devient 0, la grille sera alors considérée comme vide, cette fonction vérifie si tous les valeurs de la grille est vide, renvoie 1 pour oui et 0 pour non.

AfficherGrille() et CacherGrille() :

Deux fonctions utilisées surtout dans le mode tricheur, AfficherGrille() permet d'afficher l'intégralité de la grille et CacherGrille() cache l'intégralité de la grille sauf les cartes qui doivent rester dévoilée.

## Structure du programme :




ChargerMode.h et ChargerMode.c s'occupe de tout ce qui est partie éphémère du programme (grille, tableau, charger une difficulté)

Jeu.h et Jeu.c s'occupe de l'exécution du jeu, (tricher, première carte, deuxième carte, compteur) à l'aide des fonctions de ChargerMode.h

SaeMain.c charge le menu principal.

## Grille:

Cf fonctions : CreerGrille() :

## Conclusions personnelles des auteurs :

Khuong NGUYEN :

Le projet est intéressant, il m'a beaucoup aidé à comprendre l'allocation dynamique, à être familier avec la syntaxe du langage C et comprendre la différence entre C et Python. Cependant, la progression du projet est inconstante dû aux autres projets et contrôle en parallèle. On a investi beaucoup de temps pour créer la fonction maîtresse du programme : CreerGrille(), et étonnamment l'affichage du temps.

Eliott BRUN:

Projet très amusant bien que long et complexe, réalisé avec difficulté à cause d'une incapabilité d'installer une vm fonctionnelle sur mon ordinateur personnel. Ce projet m'as permit de mieux comprendre comment fonctionne le .h et le Makefile que je trouvais au début très contre-intuitifs, avec une difficulté à faire différencier une carte différente de la carte correspondante. Légère perte de temps sur la sélection des images pour le jeu qui nous nous sommes rendu-conte n'étaient pas compatible avec la bibliothèque graphique de l'iut.