

# SAE11\_2022 - Jeu de paires

VALLEJOS  
Andrés

## Table des matières

Introduction.....	2
I - Description des fonctionnalités du programme.....	3
II - Présentation de la structure du programme .....	6
III - Explication des données .....	8
IV - Exposition de l'algorithme .....	10
V - Conclusion .....	13

## Introduction

Le jeu de paires est un jeu auquel on aura le droit de choisir parmi toutes les cartes qu'on nous propose, les deux qu'on veut retourner pour trouver les paires de chacune des cartes sur le plateau !

Le principe est très simple, trouver le double de toutes les cartes sur le plateau de jeu et si cela vous tente, le plus vite possible ! Un chronomètre est disponible sur la page du jeu !

## I - Description des fonctionnalités du programme

1) Quand le jeu est lancé, nous pouvons cliquer sur les cases de jeux **Niveau Facile**, **Niveau Moyen** et **Niveau Difficile** pour pouvoir jouer aux jeux de paires sur des grilles de tailles différentes.



2) Quand l'une des pages de jeux s'affichera, un chronomètre se lancera pour l'affichage du temps en minutes et en secondes depuis le début de la partie.



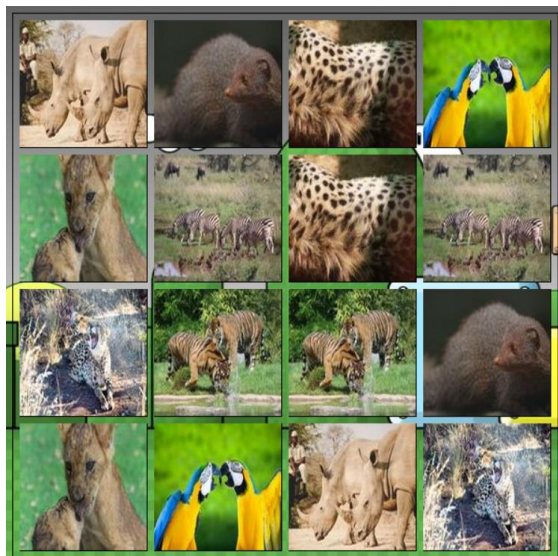
3) Nous pourrons ensuite cliquer sur la carte verso de notre choix pour la retourner et ensuite chercher la paire parmi les autres.



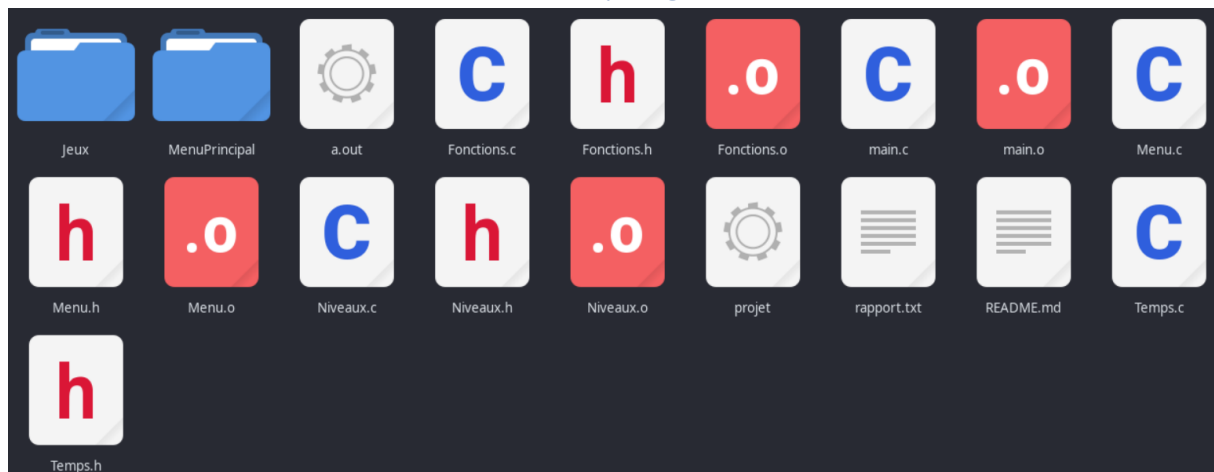
4) Un délai de 1 seconde se lancera après avoir cliqué sur la deuxième carte avant le retour en mode verso si le double de la carte n'a pas été trouvée.



5) Le mode triche sera disponible en appuyant sur la touche **t** de notre clavier pour pouvoir retourner toutes les cartes de notre jeu et interrompre le chronomètre. Nous pourrions rappuyer dessus si nous voulons reprendre la partie.

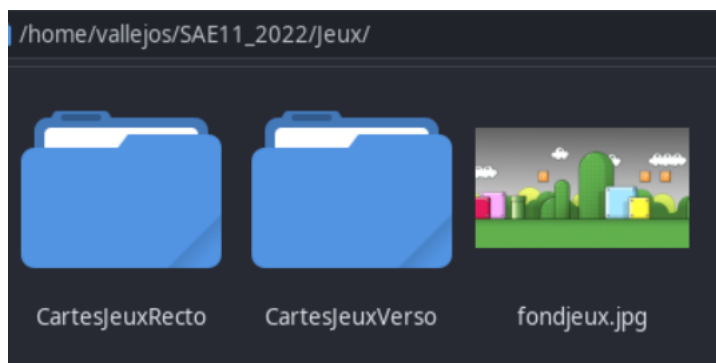


## II - Présentation de la structure du programme

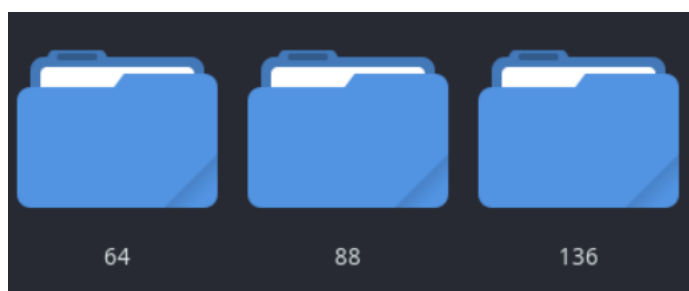


Mon programme est composé de 2 dossiers :

1. Le dossier Jeux,



composée d'une image fondjeux.jpg de 1070x601 pixels.et de deux sous dossiers



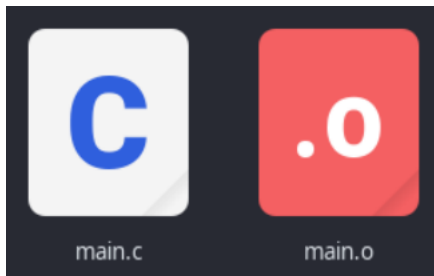
pour l'une comprenant 33 cartes de jeux recto dimensionné à 136x136pixels, 88x88pixels et 64x64 pixels. De me même pour le verso sauf qu'il n'a qu'une carte pour chaque sous dossier 136, 88 et 64.



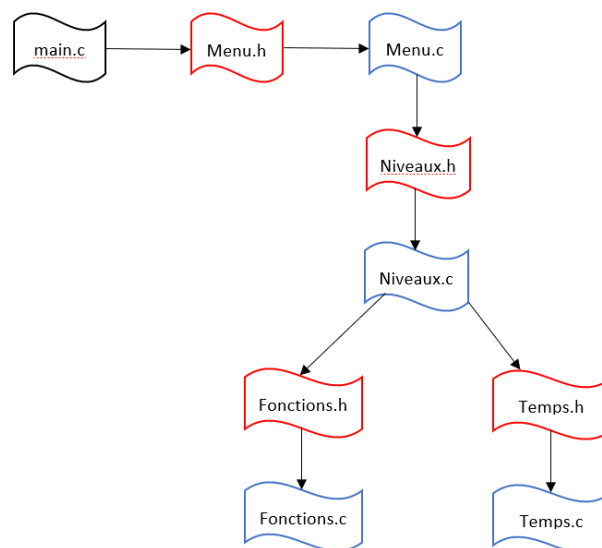
## 2. Le dossier Menu principal,



composée de trois image champi rouge, orange ou vert sous format jpg et une image luigi.jpg en 1070x601 pixels.



Nous pouvons aussi voir dans mon dossier le fichier main.c qui sert à lancer mon jeu.



Et donc voici en globalité comment se compose mon projet avec :

- Le fichier main.c, qui lance le menu
- Le fichier Menu.c qui proposera trois niveaux constitués dans le Niveaux.c
- Les Fichiers Fonctions.c et Temps.c qui contiennent des fonctions utilisées pour générer les niveaux



### III - Explication des données

La grille de jeux est composée de données, valeurs et types qui seront les bases de cette structuration. Grace à trois niveaux qui seront donnés grâce aux valeurs attribués à DIM dont 4, 6 et 8 avec des tableaux de difficultés différentes.

La structure de la grille est faite à l'aide de ces valeurs :

La taille LH\_TABLE de la table de la grille de jeux divisibles par 4, 6 et 8.

```
#define LH_TABLE 576
```

Les espacements de 8 pixels entre les différentes cartes pour éviter l'entassement.

```
#define ESPACEMENT 8
```

La longueur et la hauteur de la taille d'une carte plus l'espacement.

```
#define LH_CARRE LH_TABLE/DIM
```

La longueur et la hauteur de la taille des côtés d'une carte.

```
#define LH_CARTE LH_CARRE-ESPACEMENT
```

Le nombre d'image à disposition par rapport à la grille de jeu choisie.

```
#define NB_IMAGES_UNIQUES DIM*DIM/2
```

Le nombre de cartes dans le dossier images.

```
#define NBCARTES 33
```

Coordonnées x et y du premier petit carré.

```
int Xcarre = 255;
```

```
int Ycarre = 20.5;
```

Coordonnées des carrés ou l'emplacement des cartes seront attribués.

```
int x,y;
```

Indices des numéros de lignes et de colonnes.

```
int ligne;
```

```
int colonne;
```

A l'aide de la grille et de ses coordonnées, j'ai placé à mes conditions les positions voulues pour que chaque carte puisse être disposée dans la table de jeux à l'aide des numéros de ligne, de colonnes et des espacements.

On utilise des tableaux d'entiers représentant les numéros d'images dans ma bibliothèque composée de 33 cartes grâce à nos types ci-dessus.

TabNumImagesUniques, correspond à la moitié d'images totales de la grille de jeu, soit DIM\*DIM/2.

```
int TabNumImagesUniques[NB_IMAGES_UNIQUES];
```

Les trois tableaux suivants sont de dimension DIM\*DIM.

```
int TabNumImages[DIM*DIM];
```

```
int TabNumCartesJouees[DIM*DIM];
```

```
int TabNumCartesGagnees[DIM*DIM];
```

On a deux tableaux d'entiers à deux dimensions correspondant aux lignes et colonnes de la grille de jeu et qui sont associés aux tableaux ci-dessus. Le premier DIM correspond aux nombre de lignes et le deuxième aux nombre de colonnes.

```
int Click[DIM][DIM];
```

```
int Image[DIM][DIM];
```

Et enfin un tableau de char contenant les chemins relatifs des fichiers images de cartes.

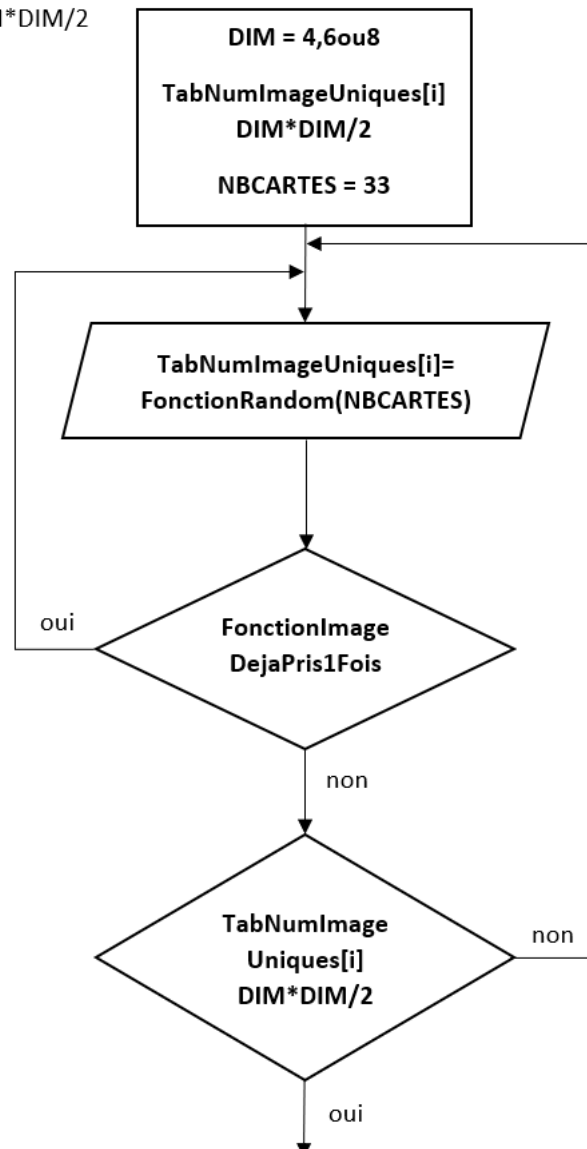
```
char FichiersImage[DIM*DIM+1][300];
```

## IV - Exposition de l'algorithme

Tout d'abord nous allons pouvoir choisir aléatoirement le numéro d'image dans notre bibliothèque d'images composé de 33 cartes dans `TabNumImageUniques[]` de dimension  $DIM * DIM / 2$ .

### 1) Génération des numéros

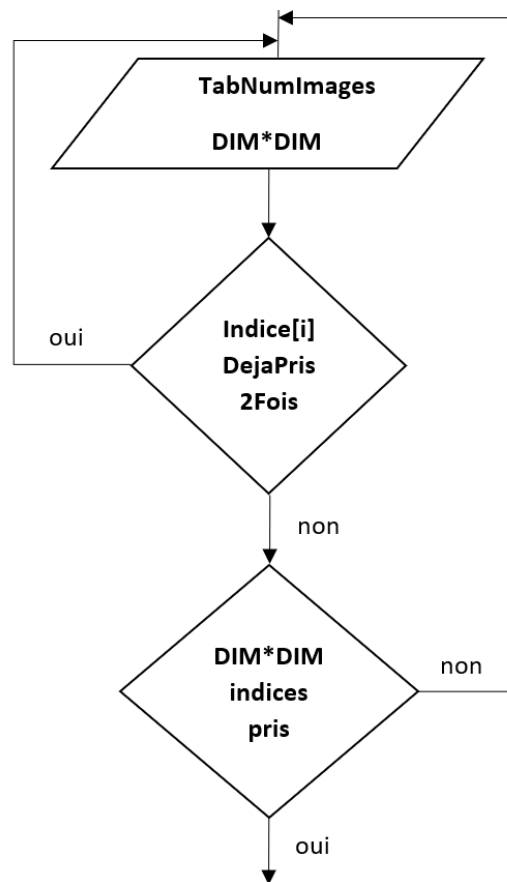
d'images uniques :  $DIM * DIM / 2$



Puis, nous allons générer les paires et choisir de manière aléatoire les indices de `TabNumImages[]`, dimensionné par `DIM*DIM`.

## 2) Attribution des paires

d'indices : `DIM*DIM`

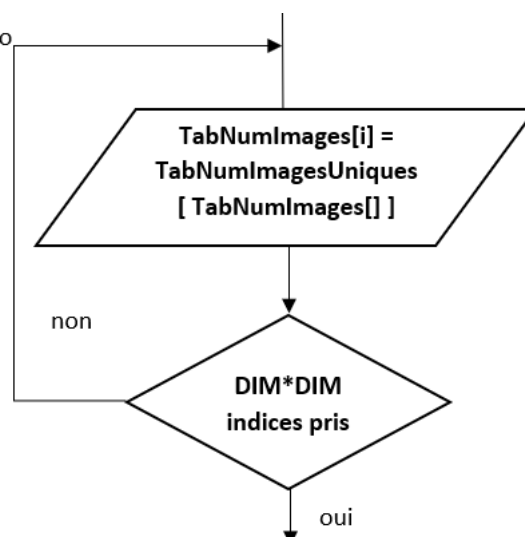


A la suite nous attribuons aux images un numéro unique propre à elle et son double grâce aux indices pris précédemment.

## 3) Attribution de numéro

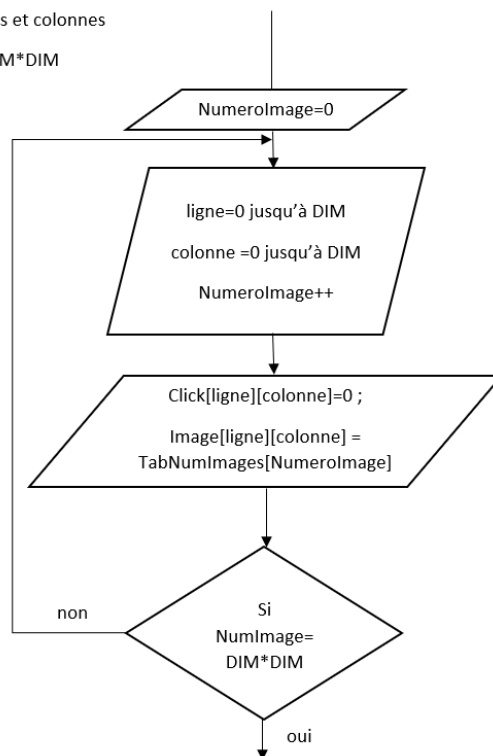
d'image de

`TabNumImages[i]`



Et pour finir le positionnement des cartes sélectionnées juste avant dans les lignes et les colonnes disponibles sur notre table de jeux dimensionné DIM\*DIM.

4) Attributions des Cartes par  
rapport aux lignes et colonnes  
dans les carrés DIM\*DIM



## V - Conclusion

En conclusion, mon jeu Jeux de paires est fonctionnel par rapport à la compilation des fichiers. Les problèmes qui m'ont fait défaut sont :

- le chronomètre
- le make et makefile
- la touche **t**, fonction triche

J'ai essayé au mieux d'organiser mon dossier SAE11\_2022 pour une lisibilité et une compréhension soit-il.