



PROJET 1.1

Le Memory



Alicia FALCON
Groupe 6

Sommaire

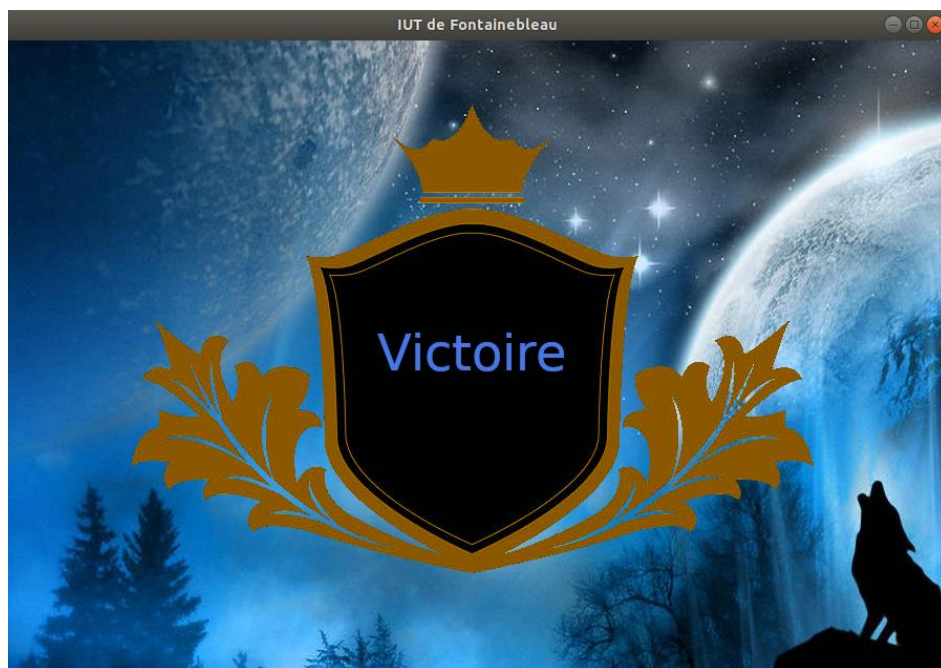
1 Introduction.....	2
1.1 Présentation.....	2
2 Fonctionnalité du programme.....	3
2.1 Fonctionnalité	3
2.2 Captures d'Ecran.....	3
3 Structure du programme.....	5
3.1 Diagramme.....	5
3.2 Explications.....	6
4 Grille de jeu.....	7
4.1 Structure.....	7
5 Algorithme.....	7
5.1 Structure.....	7
6 Conclusion.....	8

Introduction

1.1 Présentation

Le projet demandé consistait à réaliser le jeu de memory qui contient des paires de cartes, où il faut trouver et associer les images identiques. Le principe de ce jeu est donc que quand on retourne deux cartes :

- Si elles sont identiques, elles doivent rester découvertes.
 - Si elles sont différentes, elles doivent se retourner après une seconde.
- Le but de ce jeu est de retourner toutes les cartes en un temps minimum.



Fonctionnalité du programme

2.1 Fonctionnalité

Tout comme les critères imposés, il y a différentes fonctionnalités pour ce jeu :

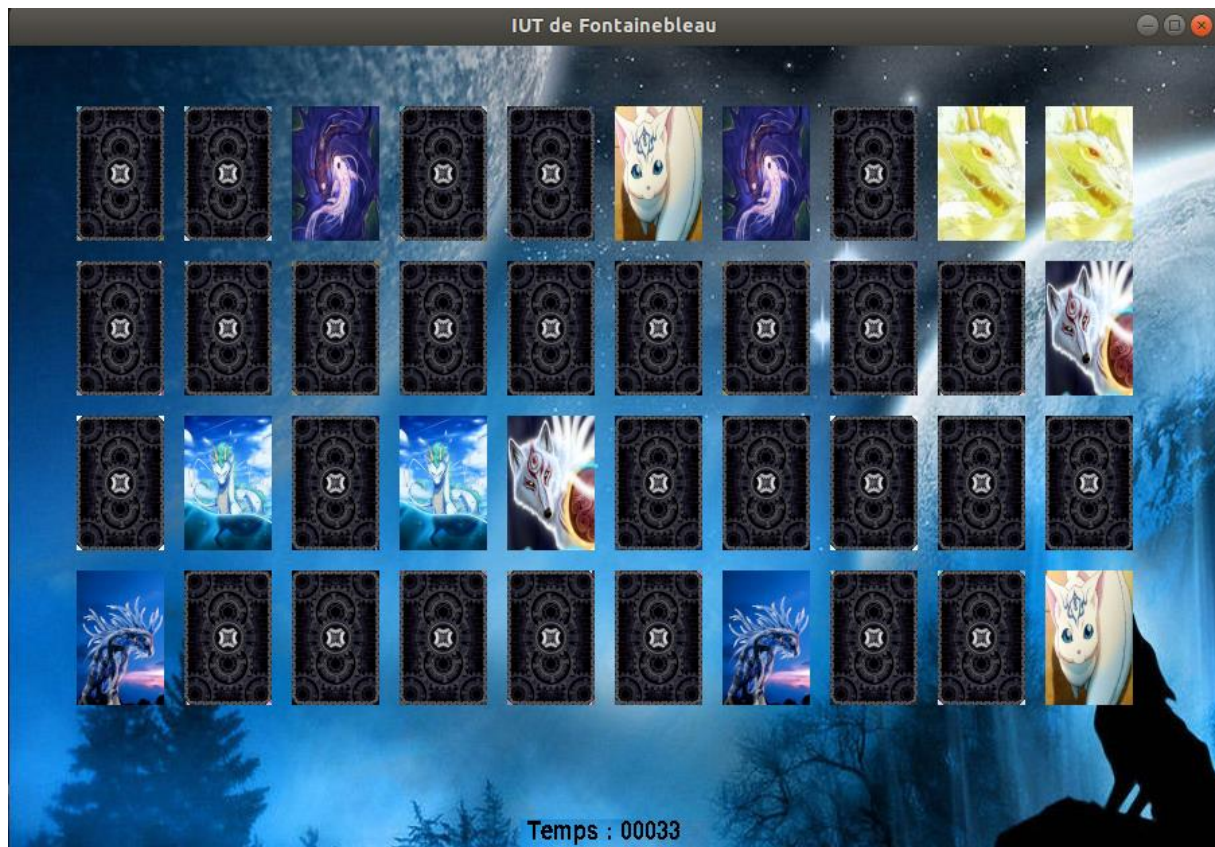
- Les cartes se retournent quand on clique dessus.
- Il y a le temps écoulé depuis le début de la partie qui est affiché.
- On peut choisir des grilles de jeux de différentes tailles.
- Dans le cas où deux cartes différentes sont retournées, elles mettent une seconde avant de se retourner.
- Un mode tricheur est mis en place en appuyant sur la touche "t". Il arrête alors l'affichage du temps et retourne toutes les cartes, jusqu'à ce qu'on rappele sur cette même touche.
- Le mélange des cartes est aléatoire.

2.2 Captures d'Ecran



Ici toutes les paires de cartes sont retournées avec le mode tricheur et le temps est arrêté. On est sur la grille la plus petit 4x4, avec 16 cartes au total.

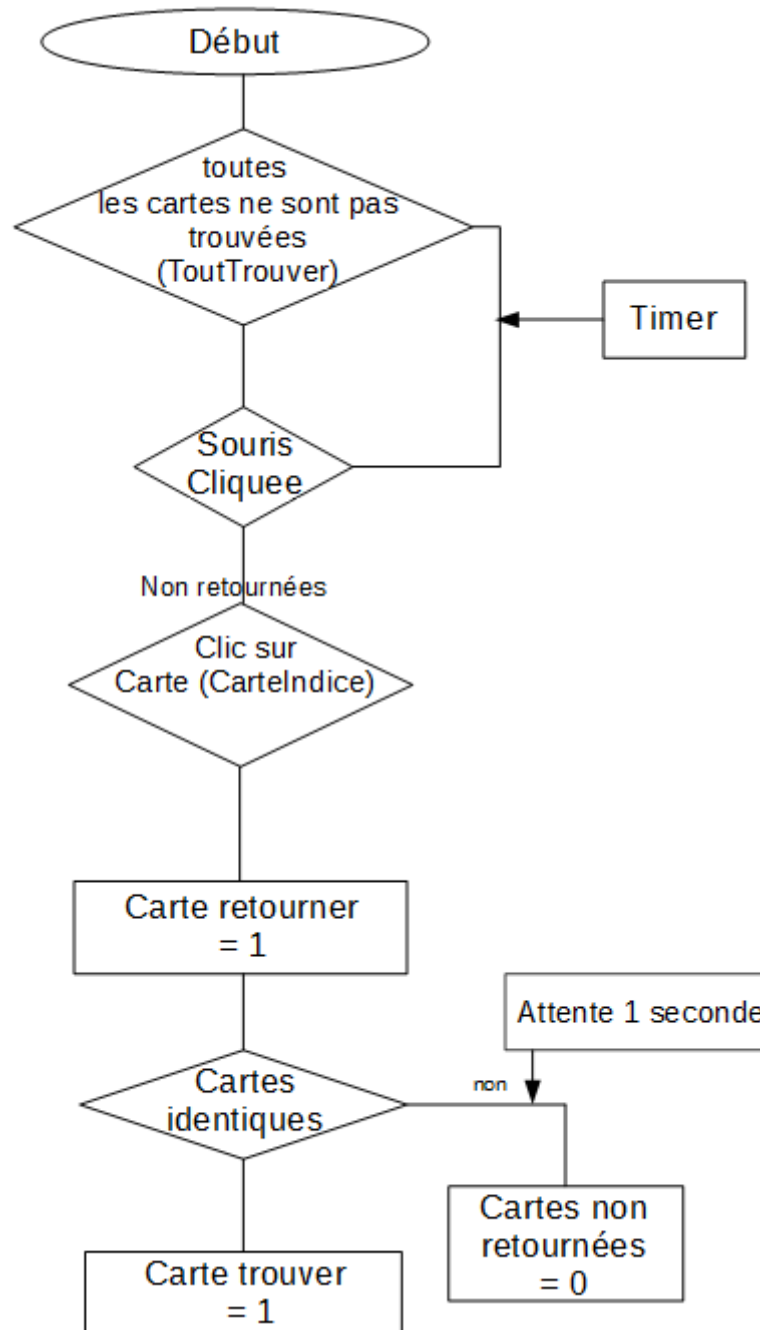
Projet 1.1



Ici, c'est la grille la plus grande 4x10, avec 40 cartes. Il y a des paires de cartes qui ont été trouvées, donc elles restent retournées jusqu'à la fin du jeu. On peut voir aussi que le mélange est aléatoire.

Structure du programme

3.1 Diagramme



3.2 Explications

Mon programme est structuré autour de la boucle while qui s'exécute tant que toutes les paires de cartes n'ont pas été trouvées, grâce à ma fonction ToutTrouver.

Ensuite, quand il y a un clic de la souris, je récupère ses coordonnées, si évidemment la carte n'est pas déjà retournée et la fonction CarteIndice vérifie si le clic était sur l'une des cartes ou non.

La fonction Timer qui affiche le temps s'exécute donc entre les deux conditions, sinon elle s'effectuerait à chaque clic.

Si le clic était bien sur une des cartes, alors la carte prend la valeur retourner = 1 et son image correspondant à son chemin est charger. Lorsque deux cartes ont été cliquées, la fonction CarteIdentique vérifie si les deux chaînes de caractères correspondant aux chemins sont identiques ou non. Si elles sont identiques la valeur de trouver = 1 et les cartes restent découvertes. Sinon, la fonction Attente s'exécute pour qu'une seconde s'écoule entre le moment du clic et le moment où les cartes se retournent, les cartes prennent donc la valeur retourner = 0.

Grille de jeu

4.1 Structure

Pour pouvoir représenter la grille de jeu, j'ai choisie de faire une structure, plus précisément un tableau de structure. Cette structure "Image" contient quatre variables.

La première variable correspond à la position x des cartes.

La deuxième variables correspond à la position y des cartes.

La troisième variables "retourner" sert à savoir si la carte est de dos ou de face : si retourner est égal à 0 c'est qu'elle est de dos, à l'inverse elle est de face si c'est égal à 1.

La quatrième variable "trouver" sert quant à elle à savoir si la carte a été trouvée, la paire de carte pour le coup. Si elle est égale à 0 la carte n'est pas trouvée et au contraire si c'est 1, elle est trouvée.

Et enfin la dernière variable qui est une chaîne de caractères sert pour stocker le nom ou le chemin des images des cartes.

Algorithme

5.1 Structure

A partir de cette structure, j'ai créé un tableau de 40 cases correspondant à chacune de mes cartes qui sont à chaque fois en doublon pour faire la paire, avec la fonction InitTableauImage.

Elle lui passe les valeurs de trouver et retourner à 0 par défaut, comme les cartes sont à l'origine retournées. Ensuite, elle leur affecte une position x et une position y en fonction des lignes et des colonnes.

Sachant que la taille qui est renvoyé par la fonction Niveau correspond aux nombres de colonnes des différentes grilles. Et pour finir elle copie les chaînes de caractères du chemin des images.

Conclusion

Pour conclure, ce projet m'a pris pas mal de temps, c'est d'ailleurs ce fait qui m'a plu aussi. Je sais que j'aurais dû séparer mes fonctions dans plus de fichiers, car ils ne sont pas bien répartis. Du coup, ça déséquilibre avec seulement deux fonctions dans un fichier. De plus, mon programme est loin d'être optimisé pour copier les chaînes de caractères dans le tableau de cartes. Je compte donc amélioré tout ça pour la prochaine fois.