



JEU DE PAIRES

SAE 1.1



Sommaire

I – Introduction.....	2
II – Fonctionnalité.....	2
III – Structure	6
IV – Donnée.....	7
V – Algorithme	8
VI – Conclusion.....	8

I – Introduction

Le but de ce projet est de confectionner un Jeu de Paires, ce jeu a pour but trouver une paire de carte en les retournant une à une. Le jeu continue tant que toutes les paires n'ont pas été retrouvées, pour compliquer le jeu les cartes ne resteront tournées seulement 1 seconde le joueur devra donc aller à la fois rapidement et user bien de sa mémoire pour retenir l'emplacement des cartes vues. Ce jeu prendra donc fin une fois toutes les paires trouvées

II – Fonctionnalité

Le jeu doit être conçu pour fonctionner avec **3 niveaux différents**, pour y accéder nous avons donc conçu l'interface graphique du menu ayant 4 boutons fonctionnels :

1. Le bouton "**Easy**" qui nous emmène sur une grille de jeu en **2x2** et qui utilise donc **2 paires** d'images (stocker dans le fichier "pp")
2. Le "**Medium**" qui nous affiche ensuite la grille en **4x4** qui utilise qui utilise **8 paires** d'images
3. Le "**Hard**" qui est le plus dur des 3 avec une grille de **8x8** qui utilise lui **32 paires d'images**
4. Et le bouton "**Quitter**" qui nous fait sortir du jeu

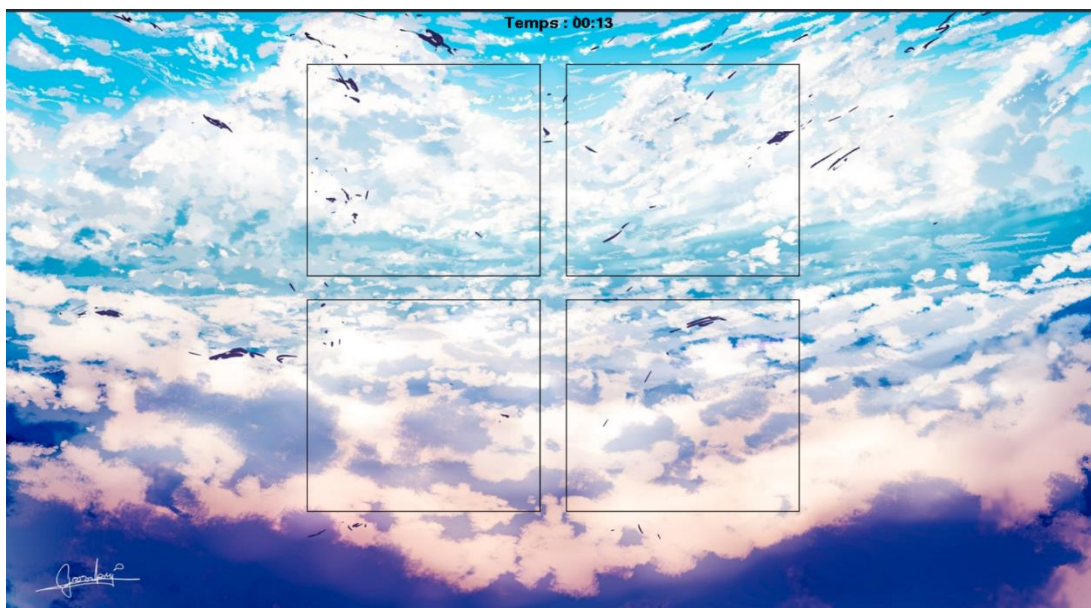
Ces images seront prises de manière aléatoire dans le fichier "pp" pour ensuite être utilisées dans le jeu



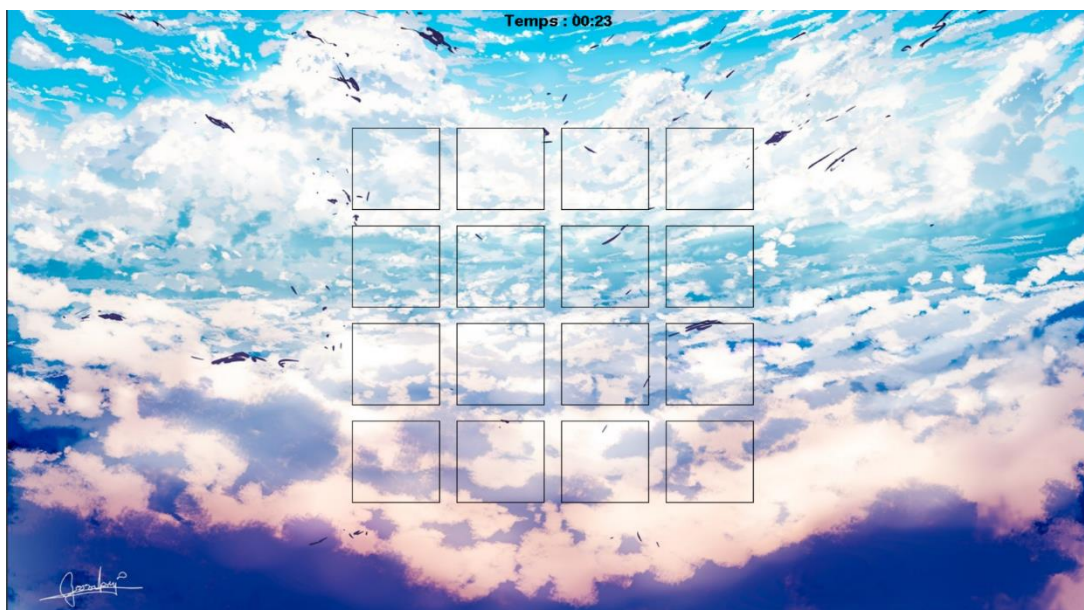
Une fois le jeu lancé, l'utilisateur doit être capable juste en appuyant sur la touche " T " d'afficher toutes les cartes retourner pour pouvoir tricher tout cela sans rajouter du temps au chronomètre. Une fois le jeu totalement terminer le programme appellera le fichier " Fin " pour afficher l'écran de fin de jeu

Description : Ici l'interface Graphique est liée à "Easy" "Medium" et "Hard" grâce à la fonction "easy.h" "medium.h" et "hard.h", cela donne comme effet que quand nous cliquons sur une des trois difficultés proposer, le jeu nous emmène automatiquement vers la grille adapter tout en ayant le "Timer" à l'endroit voulue

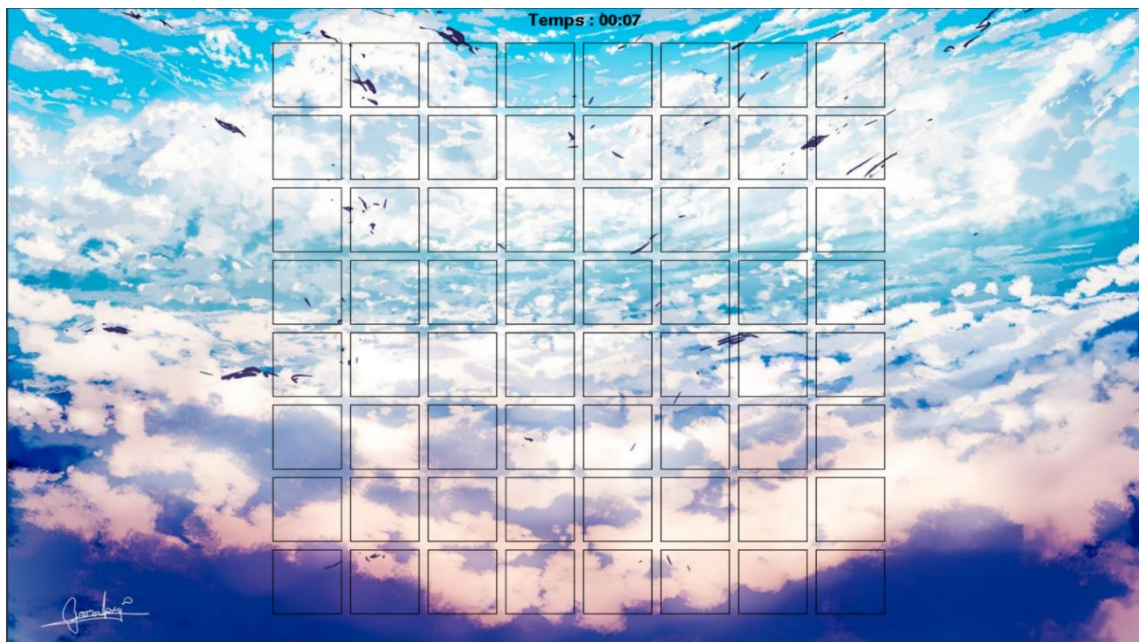
Easy :



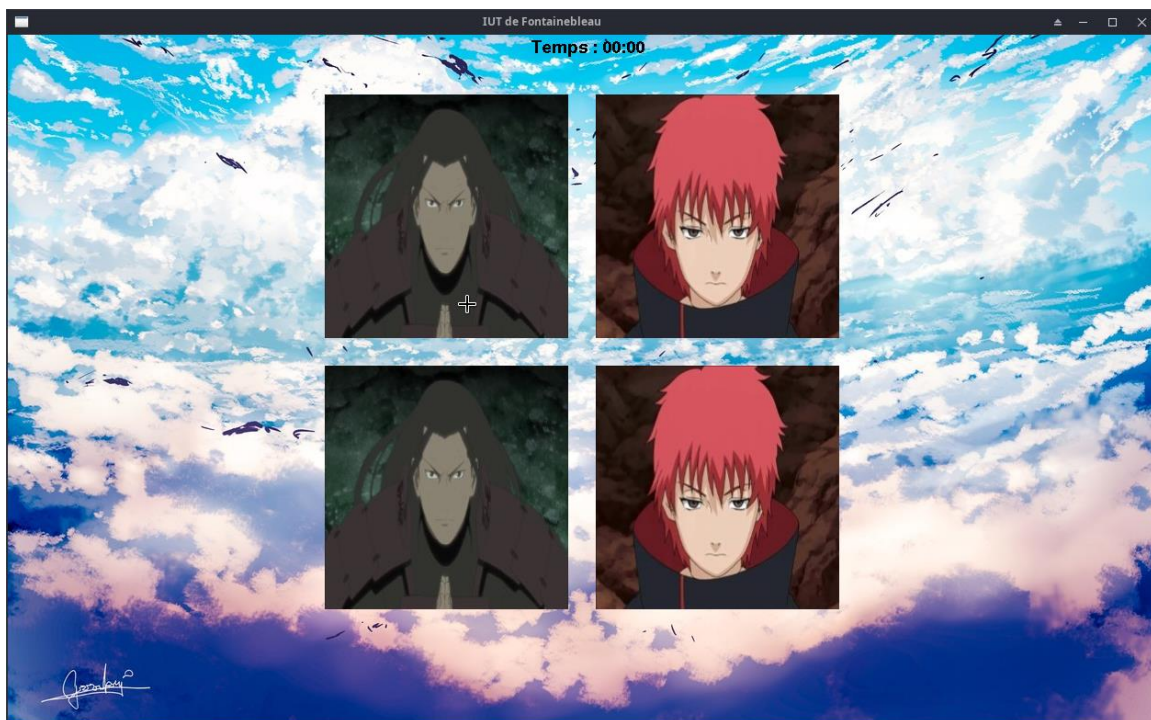
Medium :



Hard :



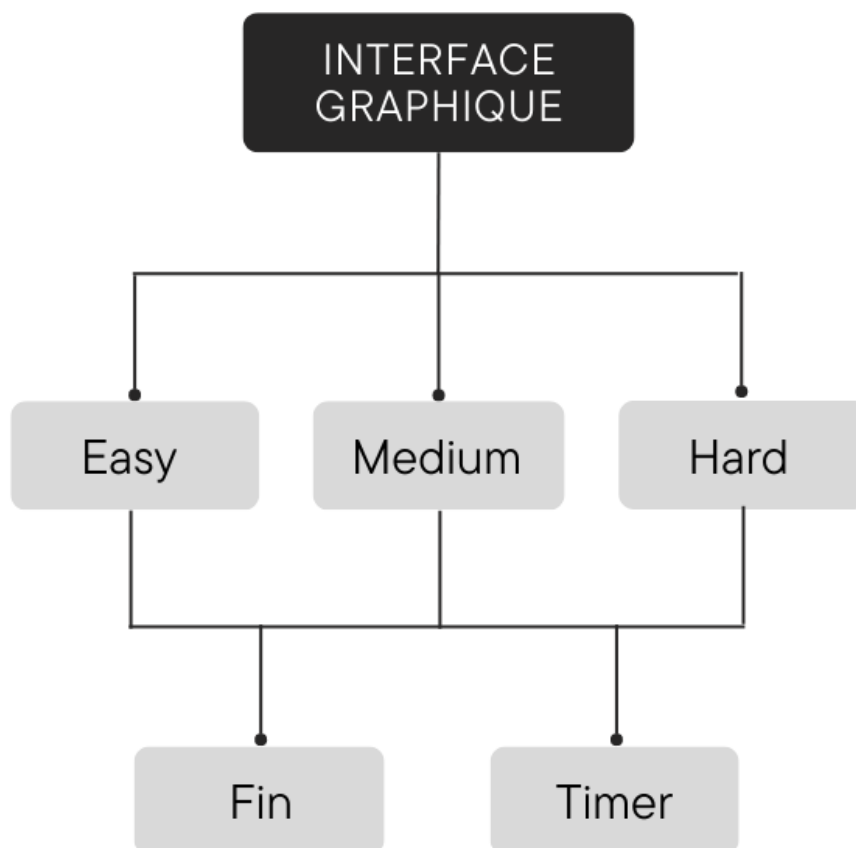
Dans ce jeu nous trouverons également un mode de triche, quand nous appuyons sur ta touche “t” le programme affiche toutes les cartes mais retourné pour mémoriser les emplacements et finir plus vite. Pour que ce mode de triche soit plus efficace il arrêtera également le timer pour avoir le temps le plus petit possible



L'Interface Graphique agit à la fois comme le menu et comme celui affiche le jeu qui affiche le jeu grâce aux multiples fonctions auxquelles il est relié. Une fois le niveau terminer le niveau appellera lui-même la fonction " **fin** " pour afficher l'interface de fin et ainsi réellement terminer le jeu et le quitter.

III – Structure

Pour pouvoir programmer le jeu de manière claire nous avons dû structurer le programme de la façon suivantes :



En haut de cette structure nous avons le fichier " InterfaceGraphique " qui affiche le menu mais qui agit aussi comme un relais vers les divers niveaux proposer, il possède aussi les boutons pour se diriger vers ces divers niveaux.

InterfaceGraphique est donc le centre de mon programme. Grace a une fonction, ayant au préalable incluse "Easy ", "Medium", "et " Hard " il vas pouvoir afficher les niveaux en utilisant la fonction "check_hitbox" pour cliquer

sur le bouton et ensuite appeler soit “facile”, “moyen” ou “hard” selon la difficulté choisie pour afficher les niveaux.

Pour effectuer une action sur les boutons nous utiliserons la fonction “hitbox” que nous avons définie au préalable dans “InterfaceGraphique”

Quand le jeu est en cours de fonctionnement, le timer est affiché en haut au centre, il s’affiche grâce à la fonction “timer” qui permet d’afficher le timer et de le faire évoluer au cours de la partie grâce à la fonction “maj_timer” et s’arrête grâce à la fonction “end_timer”.

Une fois que toutes les cartes sont retournées est que le jeu est donc terminé, le niveau va appeler le fichier “fin”, ce fichier créera une fenêtre avec l’interface de fin, elle possèdera un bouton quitter qui va nous servir à arrêter le programme et donc de fermer le jeu mais nos félicitations pour avoir terminé le jeu

IV – Données

Nous avons créé les structures pour les zones que nous avons appelé hitbox et les cartes que nous avons appelé “carte”.

Les zones contiennent 4 informations :

1. PositionX
2. PositionY
3. Hauteur
4. Longueur

Elles seront donc utilisées pour déterminer les zones dans lesquelles nous pouvons cliquer (bouton et carte).

La carte possède quand a-t-elle 4 informations

1. Le numéro de la carte
2. Le chemin vers la carte (file)
3. L’état de la carte
4. Une zone (hitbox)

Le tableau multidimensionnel nommée "tableau_carte" contient quand a-t-elle les carte de la partie. Le tableau contient des structure " carte" qui nous permet de faire des boucles pour parcourir toutes les cartes

V – Algorithme

Pour remplir le tableau de carte , nous effectuons grâce a notre algorithme une selections aléatoire des images numéroté de 1 a 60 , pour crée les structures des cartes nous avons insérer une boucle qui appel la fonction "creation_carte" et qui par la suite les revoie pour les stocker dans le tableau "carte" . Lorsque que la boucle fonctionne , des calcules sont fait afin d'obtenir l'emplacement de la carte en fonvtion des colonne , la taille quand a elle reste fixe . Pour finir , un formatage de la chaîne de caractère est effectuer pour obtenir le chemin d'accès vers la carte , le chemin varie selon le nom donnée au ficgier de la carte

VI – Conclusion

Amir : Pour conclure ce jeu met en pratique toutes les compétences acquises depuis le début de l'année et même plus, cette SAE m'aura certes apporté énormément de stresse mais aussi des connaissances et une manière de travailler en équipe efficace qui m'a permis de fournir un travail à la hauteur de mon espérance.

Mikailcan : Ce projet m'a fait découvrir la difficulté d'un projet, l'importance de se gérer pour éviter les erreurs, nous avons essayé de surmonter les difficultés même si la fatigue s'accumulait. J'espère qu'il vous plaira !