

RAPPORT SAE 1.1

Table des matières :

I) Introduction.....	p.3
II) Description des fonctionnalités.....	p.3
III) Organisation du code.....	p.4
IV) Explication de l'algorithme de remplissage.....	p.5
V) Conclusion.....	p.6

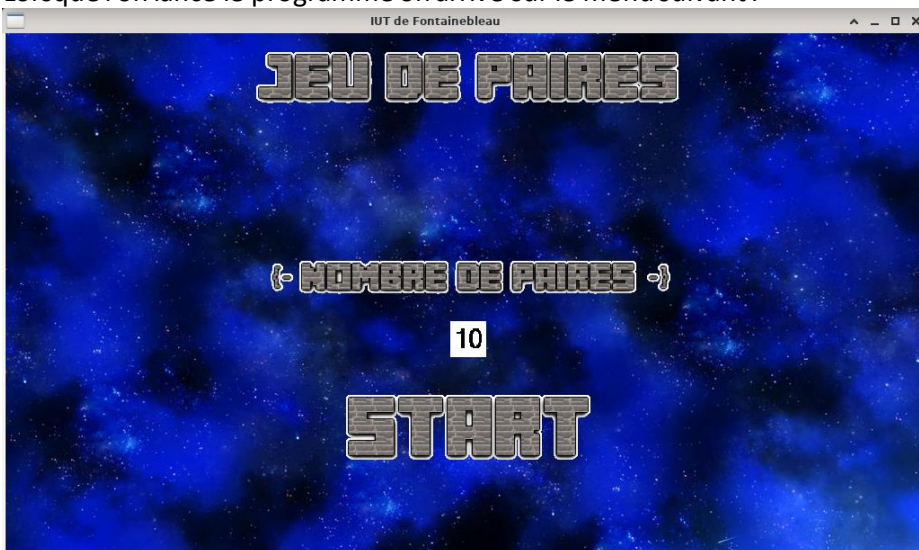
I) Introduction :

Lors de ce projet de développement on a dû créer un jeu de memory (ou de paires) en langage C. Un memory est un jeu où il faut retrouver les paires en les retournant 2 à 2, si les deux cartes sont différentes on les retourne face cachée sinon on les laisse face visible.

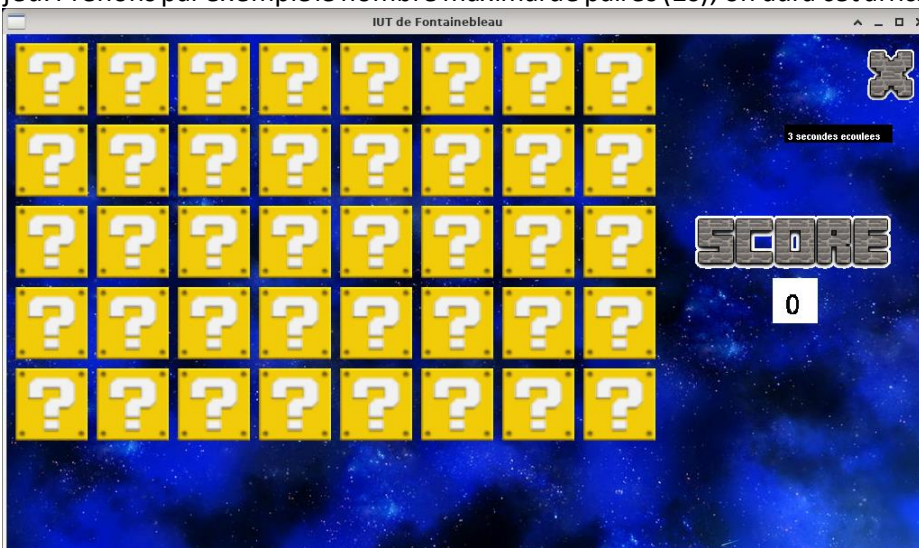
II) Description des fonctionnalités :

Menu :

Lorsque l'on lance le programme on arrive sur le menu suivant :



Si on clique sur les flèches de chaque côté de NOMBRE DE PAIRES, le nombre en dessous est incrémenté ou décrétementé suivant sur quelle flèche on a cliqué. Lorsque l'on appuie sur start on est redirigé sur l'interface de jeu. Prenons par exemple le nombre maximal de paires (20), on aura cet affichage :

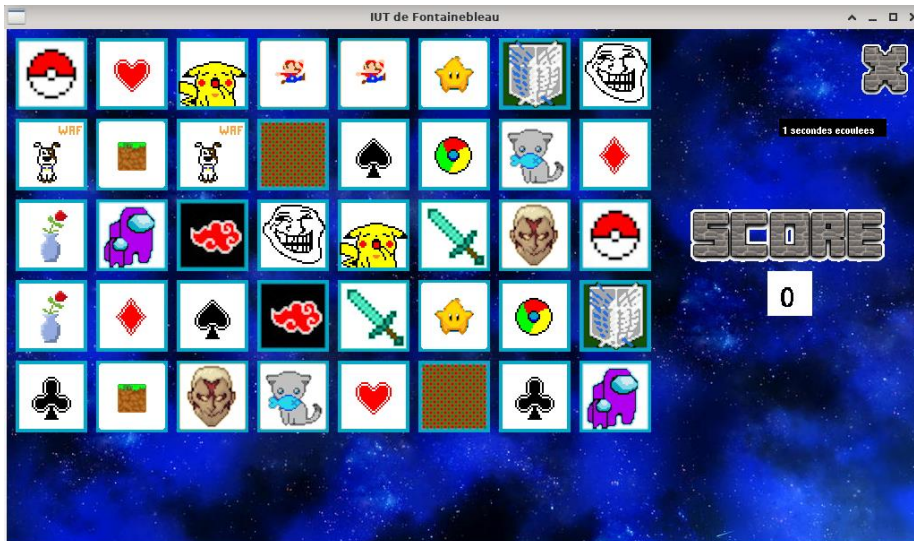


MORIN Lionel
PROAL Mathieu

On peut voir sur la partie droite un affichage du temps qui s'est écoulé et un affichage du score dynamique. On y voit aussi un bouton pour fermer le jeu. Sur la partie gauche on est sur la zone de jeu, les images ? sont cliquables et lorsque l'on clique dessus, cela permet d'afficher l'image correspondant à la position jusqu'à ce que la 2^e image soit cliquée, alors le jeu attend pendant 1s et retourne les images si elles sont différentes, sinon elles restent affichées et ne sont plus cliquables et le jeu reprend.

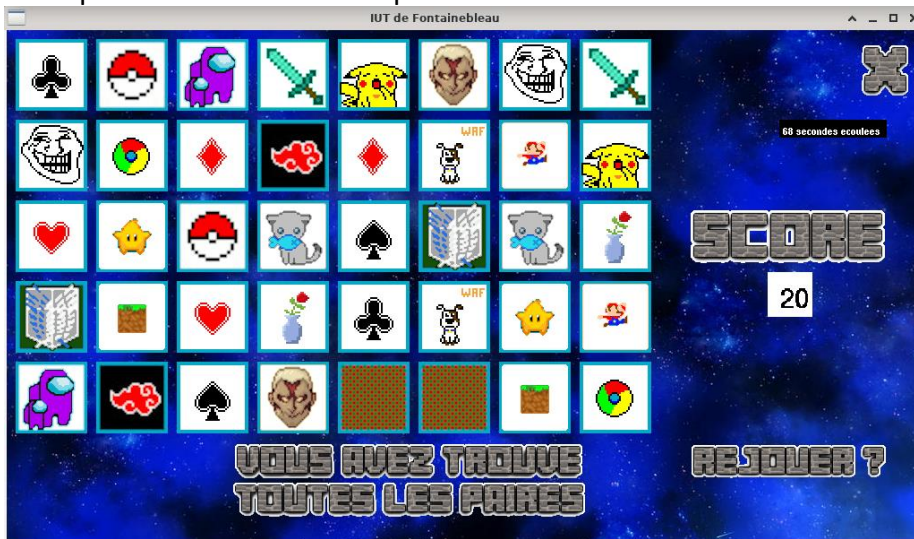
Mode Triche :

Si l'on appuie sur la touche t du clavier, on active le mode tricheur. Il montre alors toutes les images qui n'ont pas été retournées et met en pause le chronomètre. On a alors un affichage qui devrait ressembler à ceci :

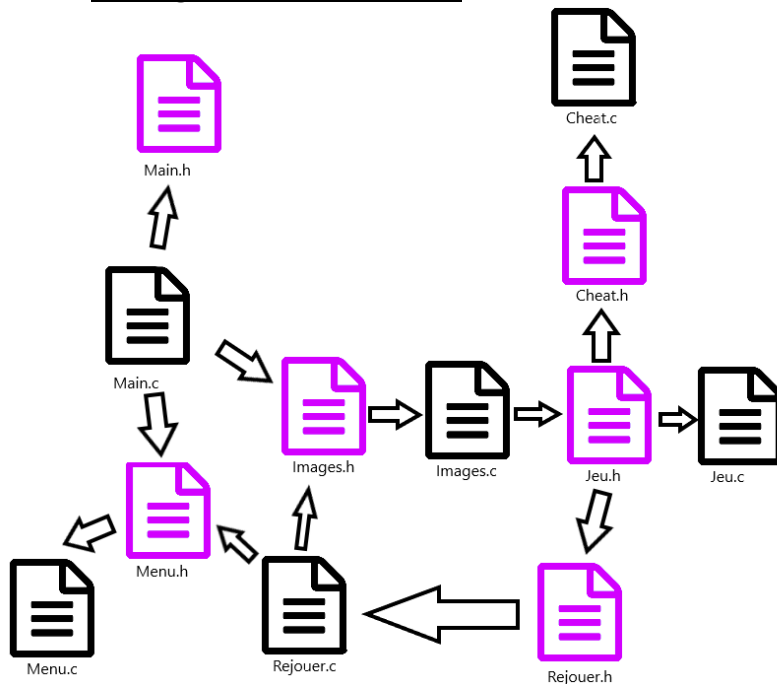


Ecran de fin :

Lorsque l'on a découvert toutes les images, un message nous indiquant la fin du jeu est affiché et un bouton nous demandant si on veut rejouer est affiché, si l'on clique dessus on est redirigé vers le menu et on peut recommencer une partie.



III) Organisation du code :



Lorsque l'on lance le programme on lance le fichier main.c qui va tout d'abord lancer la partie menu.c via le menu.h, grâce à cela on récupère le nombre de paire et on le transmet à images.c qui s'occupe de l'aléatoire et d'afficher les ? à l'écran ensuite on est redirigé vers jeu.c qui s'occupe de toute la partie « en jeu ». Enfin, rejouer.c est utilisé si le joueur appuie sur la touche pour rejouer et relance le menu puis l'aléatoire.

IV) Explication de l'algorithme de remplissage :

Pour l'aléatoire, on prend chaque image 1 par 1 et on tire un numéro au hasard. Ce numéro est l'indice du tableau dans lequel seront rangés les numéros des images dans leur ordre d'apparition à l'écran. On vérifie ensuite si, à l'indice du tableau tiré, la case est occupée ou non. Si oui on refait le tirage de la carte, sinon on place l'indice de la carte (de 1 à 20) dans la case du tableau. On répète tout cela une fois pour avoir des paires dans le tableau.

Pour ensuite afficher les cartes on utilise des boucles for qui correspondent à la taille de nos images et qui comparent par rapport au tableau généré plus tôt.

V) Conclusion :

Lionel : J'ai trouvé ce projet très intéressant à faire car il m'a permis d'approfondir mes connaissances en C, de plus, au départ je voulais partir dans une certaine voie mais on s'est aperçu que ce n'était pas la meilleure et on a changé de direction. J'ai aussi beaucoup appris sur l'utilisation de git et de comment résoudre certaines des erreurs rencontrées comme les problèmes de merge et j'ai beaucoup aimé le fait de pouvoir se partager les tâches en se donnant certaines parties du code à faire.

Mathieu : Personnellement, j'ai énormément apprécié cette SAE car elle m'a permis de comprendre plein de choses que j'avais apprises et que j'appliquais sans les comprendre, comme les .h ou les Makefile par exemple. Ensuite, j'ai toujours aimé créer des petits jeux, d'abord avec Scratch au collège puis Python au lycée, et découvrir le C puis l'appliquer dans la création d'un memory a été très plaisant. Même si nous avons beaucoup pataugé et fait des détours inutiles (comme une fonction de 2800 lignes), c'était amusant de tester toutes nos idées, et incroyablement satisfaisant de réussir à résoudre nos problèmes. Je suis plutôt fier du résultat que nous avons réussi à produire, même si esthétiquement ce n'est pas au top du top. Nous travaillions bien tout le deux, le plus souvent chacun avançait sur quelque chose de son côté, et l'autre venait l'aider en cas de besoin, puis nous réunissions le tout, tandis que d'autres fois nous travaillions ensemble sur une même partie du code assez coriace. Cela nous a montré une manière de travailler en équipe, et c'est rassurant de se dire que si on rate notre coéquipier est là pour nous épauler. En conclusion, c'était une très bonne expérience, et malgré les heures de travail je suis très content de l'avoir fait.