

Sommaire :

Page 2 - Description du sujet.

Page 3 à Page 6 - Description des fonctionnalités du programme.

Page 7 – Diagramme.

Page 8 - 1. Explication de la grille du jeu.
 2. Explication de son remplissage.

Page 9 - Conclusion personnelle.

INTRODUCTION :



Le but de ce projet était de réaliser un « memory » en langage C avec l'aide de la bibliothèque graphique de l'IUT.

Règle standard du jeu : Le but du jeu de memory est de collecter les paires correspondantes.

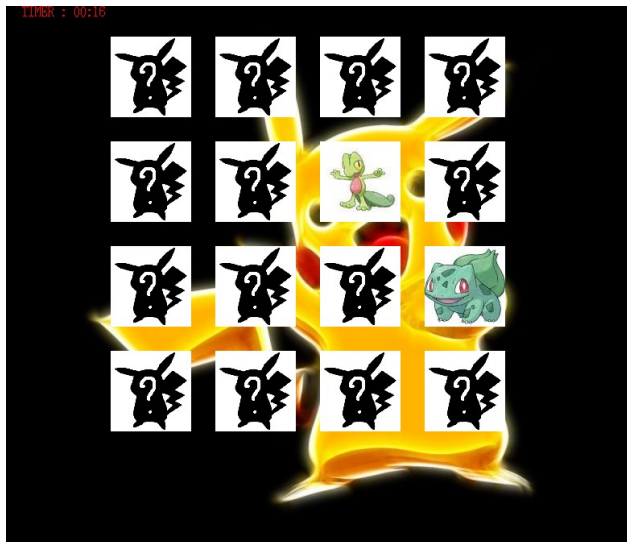
Lors de la conception de celui ci nous avons du y introduire quelques fonctionnalités :

- Les cartes seront retournées par un click de souris.
- Le jeu affichera le temps (en secondes) écoulé depuis le début de la partie.
- On pourra jouer sur des grilles de tailles différentes.
- Le temps de mémorisation d'une paire de cartes découvertes durera une seconde.
- Un mode tricheur est accessible en pressant la touche `t` qui découvre la grille et interrompt le temps jusqu'à ce qu'on appuie sur cette même touche pour reprendre la partie.

Chronomètre

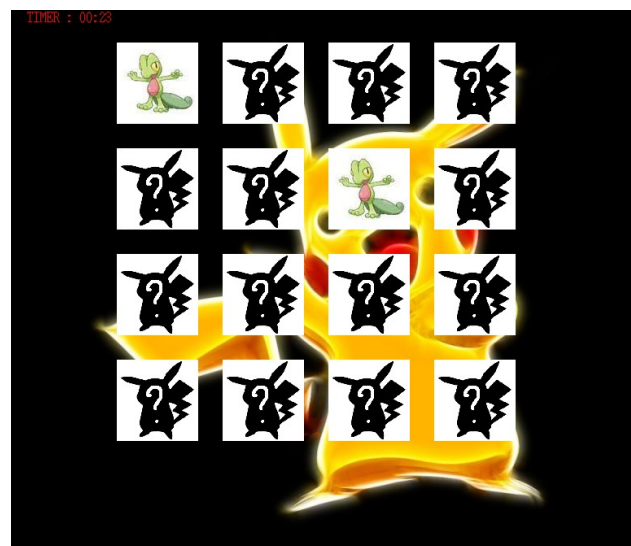
Le chronomètre augmente toutes les secondes. Lorsqu'il passe à 60 secondes il va augmenter la valeur de minute de 1 et ensuite l'afficher. Si le programme est lancé depuis 122 secondes il affichera donc : **02:02**. Il augmente uniquement lorsque le mode triche est désactiver





Insertion des cliques

Lors du lancement de la partie, toutes les cartes retournées (image avec pikachu noir) ont comme état 0. Cet état qui va être utile lors de la comparaison des cartes, en effet si l'on clique sur une carte, l'état de celle-ci sera de 1, elle restera donc visible. On va aussi prendre la coordonnée de cette carte et son numéro puisque si l'on clique sur une deuxième carte on doit comparer ces deux cartes. Si l'identité des deux cartes ne correspond pas, le programme va attendre une seconde avant de les retourner (leur état revient à 0), par contre si l'identité correspond, on va ensuite vérifier ses coordonnées ce qui permettra d'éviter un double clic sur la même case. Une fois les deux cartes trouvées elles resteront visibles jusqu'à la fin de la partie (leur état restera à 1). On a également mis une condition qui empêche le clic sur une carte avec un état de 1. Elle est donc figée.



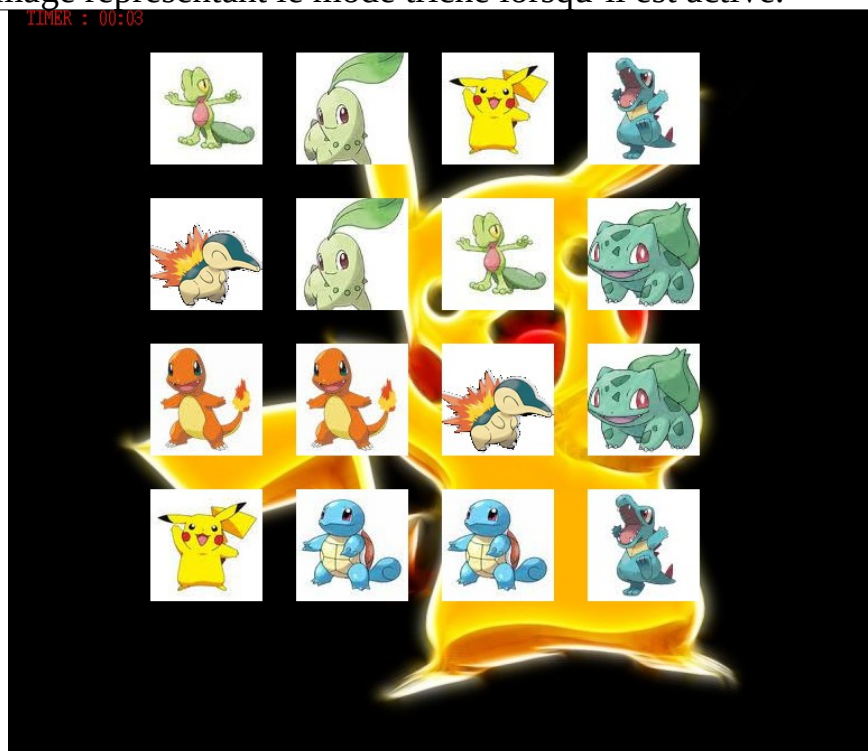
Mode triche

Le mode triche s'active lorsque l'on appuie sur la touche « t ». Il permet d'afficher toutes les cartes. On se sert du numéro que l'on a attribué à une carte. On le récupère en l'insérant dans une chaîne de caractère (« img/'numéroCarte'.png »), celle ci nous permettra donc de pouvoir afficher chaque image à sa position que l'on récupère lors de la création de la grille au début du programme.

Lorsque le mode triche est activé, le chronomètre se met en pause, on ne peut plus cliquer.

Lorsqu'on le désactive, les cartes se remettent face caché. Nous avons donc du créer une condition permettant de ne pas cacher les cartes possédant l'état 1. Puisqu'elles ont déjà été trouvée.

Ci dessous est une image représentant le mode triche lorsqu'il est activé.



Taille de la grille

Nous avons créé un menu qui se lance au début du programme. Celui permettra donc de choisir la taille de la grille. Les choix sont visibles sur l'image en dessous.

Un premier menu s'affiche, lorsque l'on appuie sur Jouer, on arrive au choix de la taille. Si l'on click sur quitter, le programme se quitte.

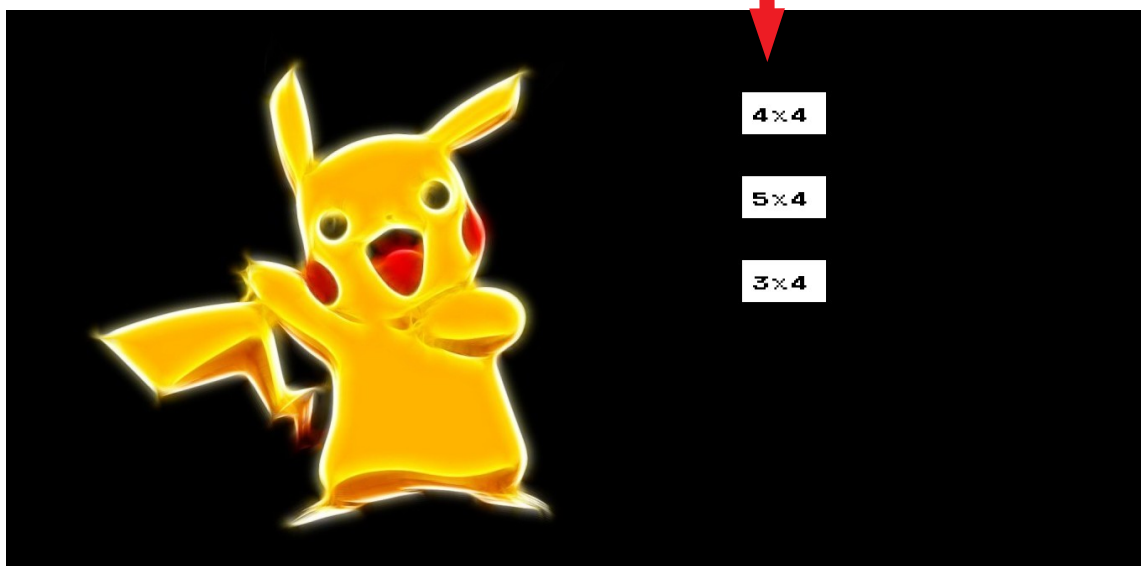
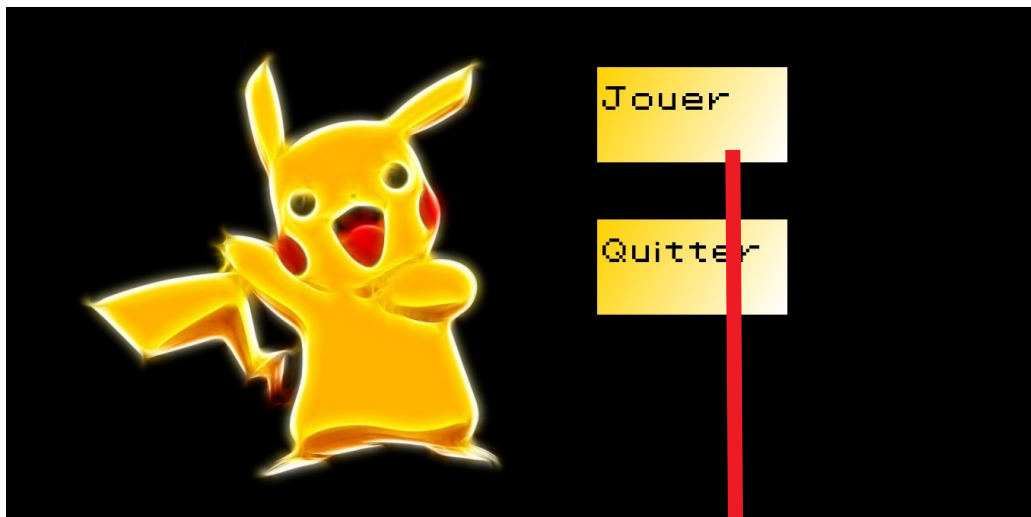
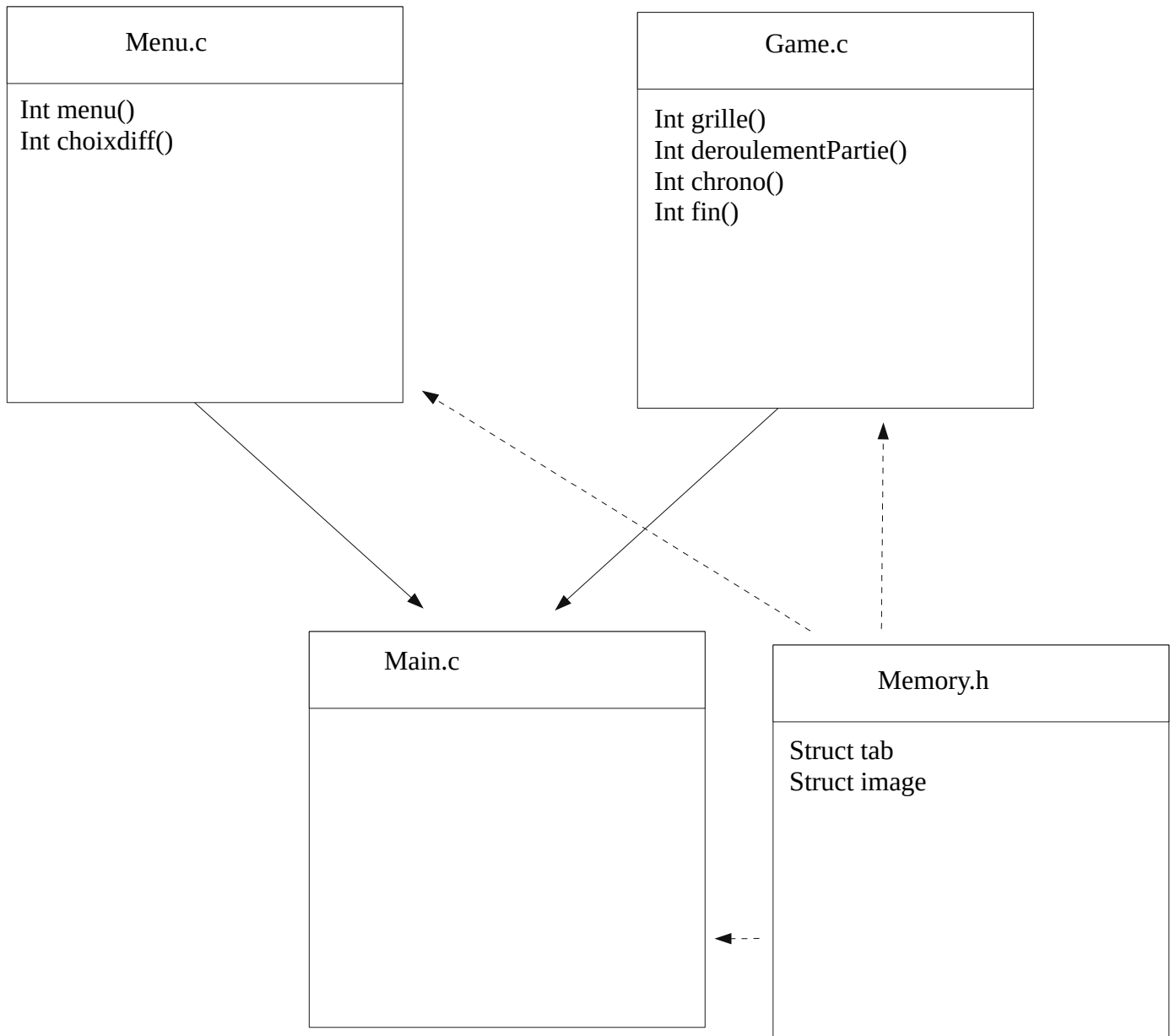


Diagramme (structure du programme)



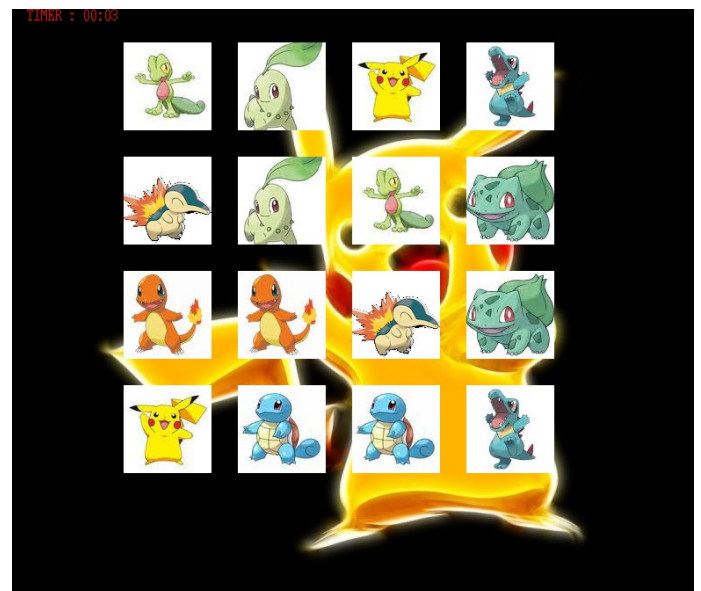
Forme de donnée qui représente la grille :

Pour stocker les valeurs des données de la grille nous avons créé un struct qui nous a permis de créer le type **tab** qui contient en sous-variable **int id** et **int etat**. Pour initialiser une valeur à chaque case de la grille on a donc créé deux variables à deux dimensions qui s'appellent **Image** et **ImageFind** de type **tab**. On a ensuite fait deux boucles permettant de parcourir les tableaux en y attribuant à chaque fois un numéro qui correspond à l'id de la carte et son état.

On a alors créé deux grilles, une correspondante aux cartes de dos (imageFind) et une correspondante aux cartes lorsqu'elles sont visibles (image). On va ensuite afficher la grille possédant les cartes visibles lorsque le mode triche est activé, et donc afficher la grille avec les cartes retournées si celui-ci est désactivé. Cette grille changera au fur et à mesure de la partie puisque lorsque l'on trouve ou clique sur une case, celle-ci verra son état changer (vu plus haut). Cela permettra de sauvegarder la grille au fur et à mesure de la partie et que lorsque l'on utilise le mode triche, ne pas perdre son avancé.



Grille carte de dos → imageFind



Grille initiale, carte visible → image

Conclusion personnelle :

Stéphane : Pour ma part, cela m'a apporté plusieurs compétences et m'a permis de découvrir de nouvelles choses plus concrètes, cela m'a permis d'exécuter un programme avec une table graphique. Cela m'a donné l'envie de continuer à faire des recherches pour faire d'autres jeux, ou autres mini projets. De plus, j'ai penser que c'était un bon exercice et j'ai beaucoup aimer découvrir ce que l'on pouvait faire avec du code en concret.

Gaston : En ce qui me concerne, la réalisation de ce projet m'a plu, puisque c'est la première fois que l'on utilisait une bibliothèque graphique, on travaillait en dehors de la console. On pouvait jouer avec le design, en rendant le jeu comme on le souhaitait. Ce projet fut mon premier « gros » programme et a été enrichissant puisque j'ai pu prendre conscience de certaines mauvaises habitudes, comme ne pas m'organiser avant de me lancer dans le codage, l'importance des fonctions pour rendre claire et simple son programme. Ce projet a donc été bénéfique.