



Dossier IHM : Jeu du mémory

IUT de Paris - Département Informatique



Youcef MEDILEH, Daryl MARTIN-DIPP



Table des matières

Introduction	3
Problématique	3
Développement	3
Conception	3
Programmation	3
Explication de l'application	4
Introduction	4
Formulaires	4
FormMenu	4
FormJeu	5
FormOptions	7
FormScores	10
Schéma d'ordonnancement des formulaires	12
Bilan du projet	13
Difficultés rencontrée	13
Apport du projet	13
Pistes d'améliorations	13



Introduction

La création d'un jeu est un processus compliqué qui nécessite souvent une réflexion au préalable sur toute sa structure. Cette structuration est facilitée grâce à la programmation orientée objet (POO) auquel nous avons été introduit brièvement durant le module de SDA (Structures de Données et Algorithmes Fondamentaux) puis initié dans le module de conception (BCO). Nos acquis de la Période C sur le module de programmation orientée objet nous ont grandement aidés, ainsi que le projet sur le jeu de cartes.

Cette nouvelle façon de programmer s'accompagne de tout une philosophie qui lui est propre et que nous avons essayé d'adopter afin d'entamer ce projet.

Certains langages sont intégralement conçus pour être utilisés dans ce sens tels que le C++, le Java ou bien le VB .NET dont nous nous sommes servis dans le cadre de ce projet.

Problématique

Le projet consiste en la création d'un programme représentant un jeu de mémory auxquels les joueurs peuvent jouer en interagissant directement avec le formulaire en cliquant sur les images. Les règles sont simples : le joueur doit retourner le plus de carrés de cartes identiques parmi celles qui sont sur la table pendant le temps imparti.

Développement

L'enjeux de ce projet était donc de concilier la conception et la structuration du programme avec la programmation des algorithmes nécessaire au fonctionnement du jeu.

Conception

Dans le cadre de ce projet, nous devions mettre en exécution nos connaissances acquises en termes de création de formulaire, modules et création de jeux (en BPO).

Programmation

Pour ce projet, l'utilisation du langage VB .NET a été choisie : ainsi nous devions nous familiariser avec ce dernier. Nous devions ainsi comprendre son fonctionnement général :

- Les événements (clics de boutons, keyPress...)
- Les boucles for et for each
- Les structures
- Les modules
- Les timers
- Les ImageList
- Les panels
- Tous les autres contrôles de WindowsForms



Explication de l'application

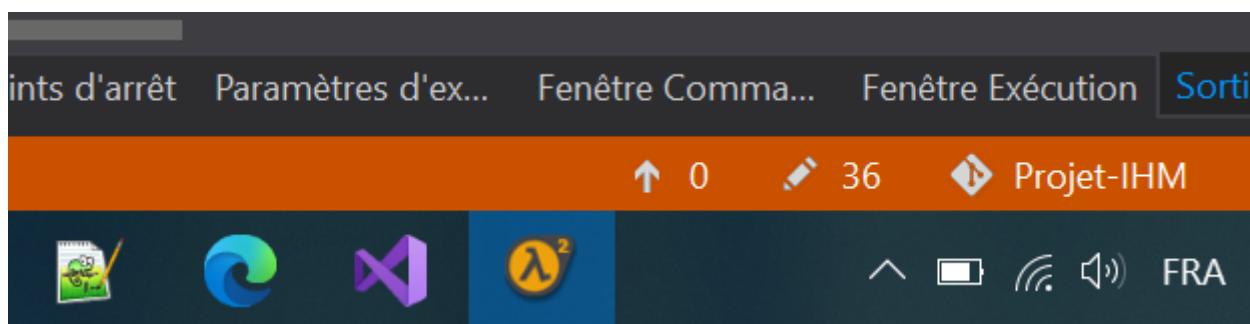
Nous allons vous expliquer ici tout ce qu'il y a à savoir sur notre jeu de mémory.

Introduction

Notre application comporte au total quatre formulaires, tous centrés, ne pouvant pas être redimensionnés, et sans ControlBox.

Pour pouvoir utiliser notre application, il faut que vous ayez au préalable un fichier nommé "save.txt" dans le même dossier que l'exécutable. Sans cela, le programme détecte que le fichier de sauvegarde est manquant, et s'arrêtera tout de suite car il ne pourra plus fonctionner correctement.

Nous avons changé l'icône Windows des formulaires.



Formulaires

FormMenu

Le menu principal permet de lancer plusieurs formulaires :

- FormJeu, qui correspond au jeu de mémory ;
- FormOptions, qui permet à l'utilisateur de choisir ses options ;
- FormScores, qui affiche le récapitulatif des scores.

L'utilisateur peut entrer son nom dans une ComboBox contrôlée, reliée au fichier de sauvegarde, et peut également quitter l'application.



Menu principal

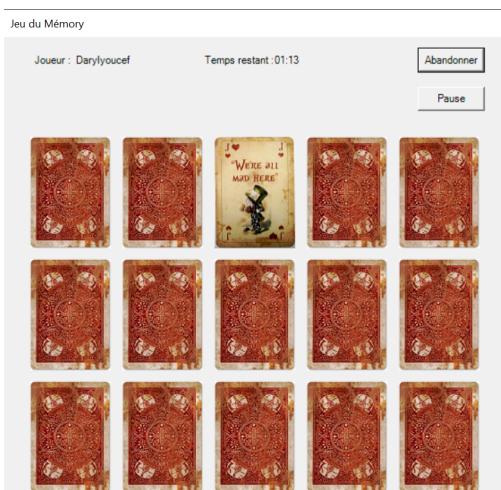


Menu principal



FormJeu

Le formulaire affichant le jeu de mémory comporte le nom du joueur, le temps restant, ainsi que les boutons Abandonner et Pause/Reprendre. Ce dernier change, selon que le jeu soit arrêté ou non.



Le bouton abandonner permet d'annuler la partie encore, sans enregistrer de nouveaux scores.

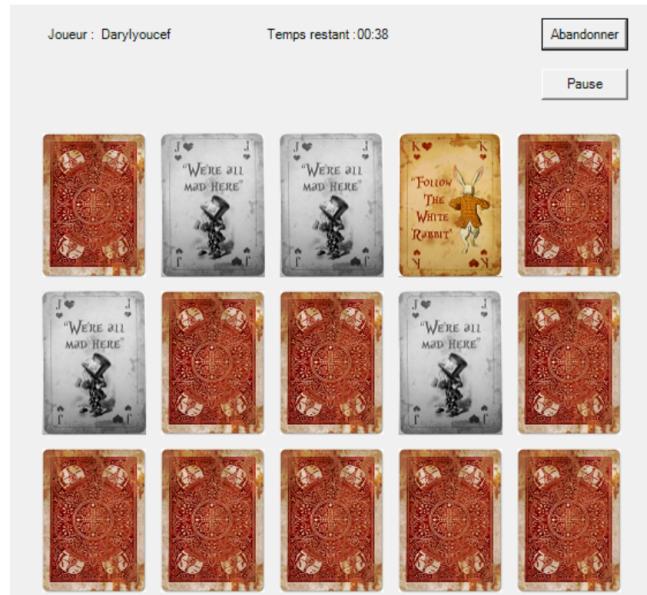
Le timer ne commence que lorsque la première carte est cliquée.

De même, lorsque deux cartes retournées ne correspondent pas, le programme va vous les montrer durant une seconde, puis les retourner.

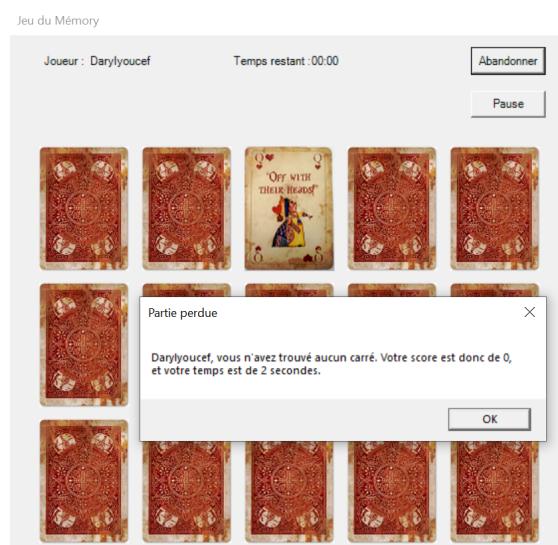
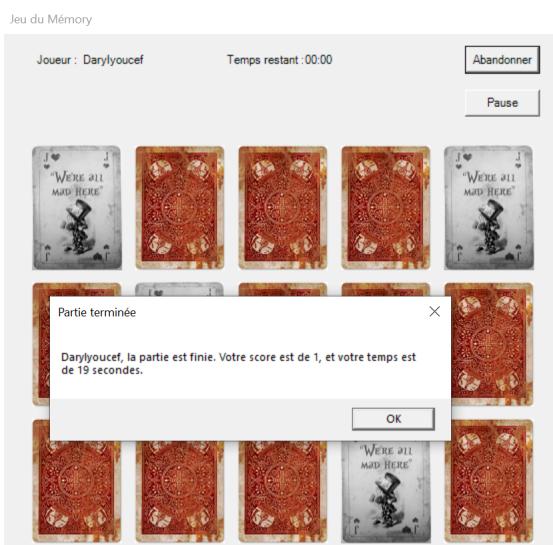
Lorsqu'une série est trouvée entièrement, les cartes vont restées retournées, et vont se mettre en noir et blanc. Le jeu peut alors se poursuivre.



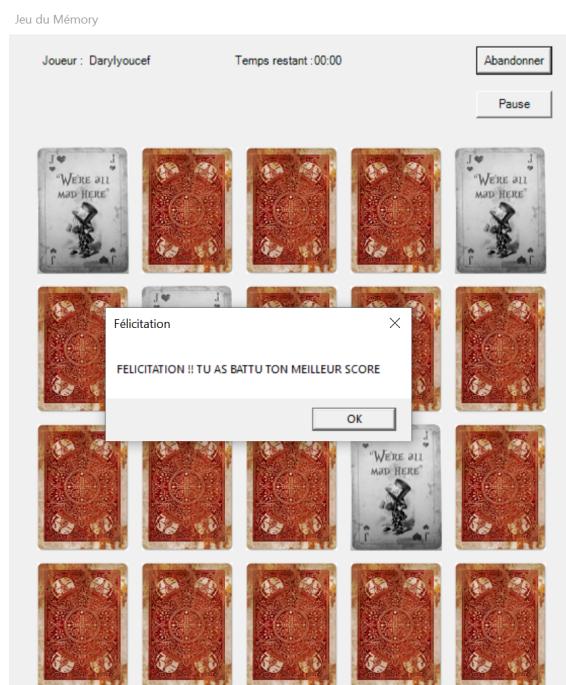
Jeu du Mémory



Une fois la partie finie, deux affichages peuvent apparaître : un si aucune série n'a été découverte, et un autre si au moins une série a été découverte. Dans les deux cas, le programme enregistre le score avec un appel à une fonction stockée dans le module traitementJoueurs.

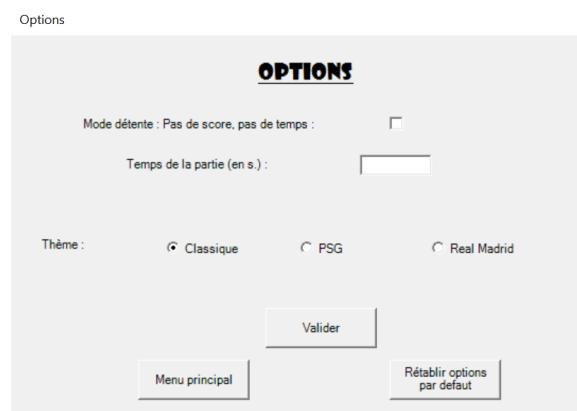


Si vous avez battu votre meilleur score, un message apparaîtra sur votre écran vous le disant.



FormOptions

Ce formulaire permet de changer les options du jeu de mémory, grâce à un CheckBox, un TextBox, un RadioButton et des Buttons.



- Tout d'abord, le mode détente désactive le timer, afin que le joueur ne soit aucunement stressé. Lorsque la partie est en mode détente, les labels correspondant au timer ne sont donc pas affichés, et les résultats ne sont pas enregistrés. Le joueur a même un message personnalisé en fin de partie.



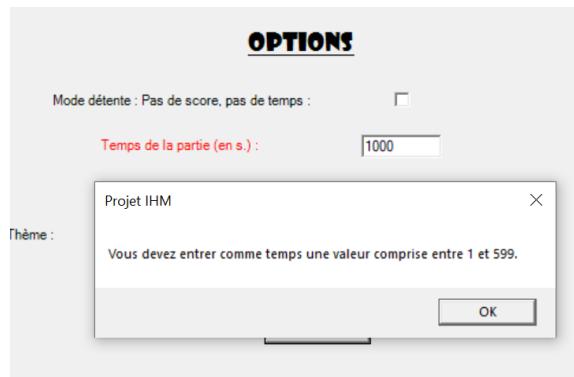
Jeu du Mémory



- Ensuite, l'utilisateur peut choisir le temps de la partie, selon ses envies. Le temps entré ne peut pas dépasser 10 minutes ; nous avons estimé que cela est une durée excessivement élevée pour une partie de mémory, qui perdrat alors de son challenge.
- Lorsque l'utilisateur a choisi le mode détente, la TextBox se grise, et son contenu s'efface.

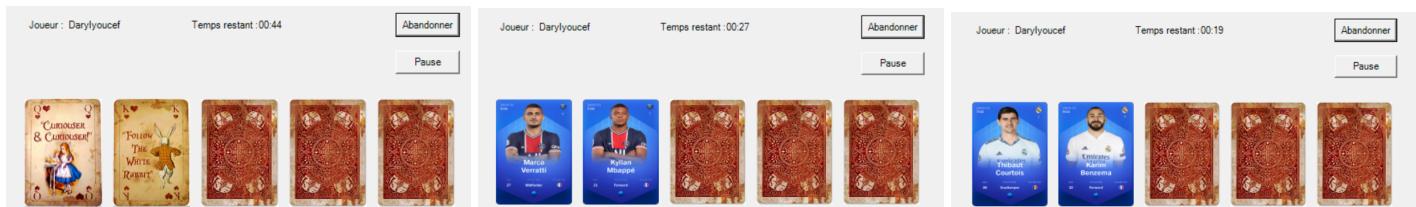


Si l'utilisateur n'entre pas une bonne valeur, un message rappel les consignes, et met l'écriture en rouge. De plus, seuls des chiffres peuvent être entrés dans la TextBox.



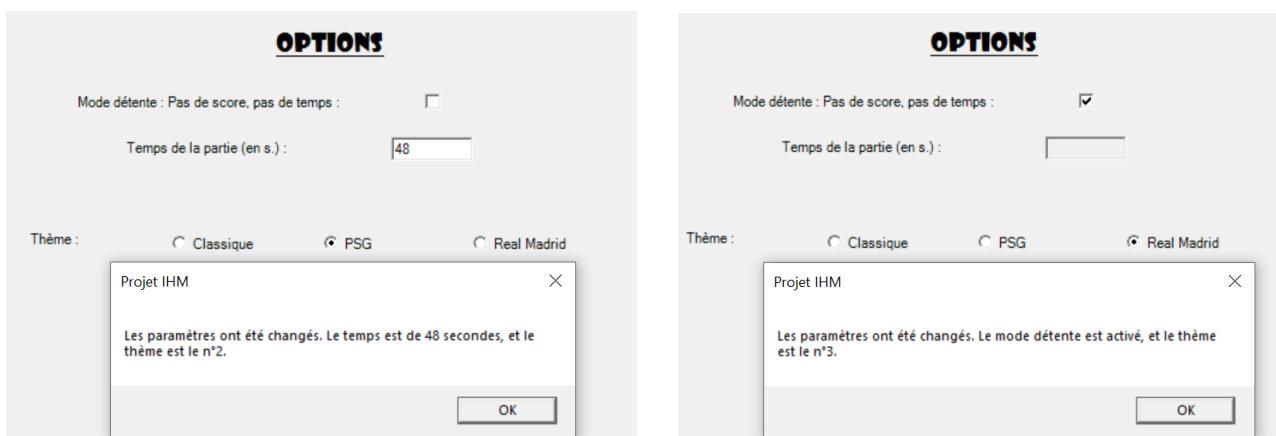


- Enfin, l'utilisateur peut modifier le thème, entre trois disponibles : Classique, PSG ou Real Madrid.



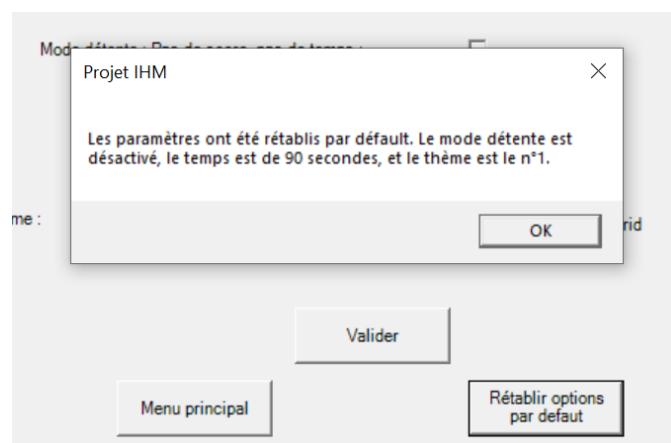
Trois boutons sont ensuite disponibles sur le formulaire.

Valider permet de valider les choix. C'est également ici que va être vérifié la consigne de la TextBox du temps. De plus, un message vous récapitulera les nouvelles options.



Le bouton “Rétablir options par défaut” permet de mettre les options de bases : un temps de 1min30, et les cartes classiques.

Un message va également vous les indiquer.



Enfin, le bouton “Menu principal” permet de retourner au menu sans sauvegarder. Les deux autres boutons vous y amènent aussi, mais enregistrent les nouvelles options.

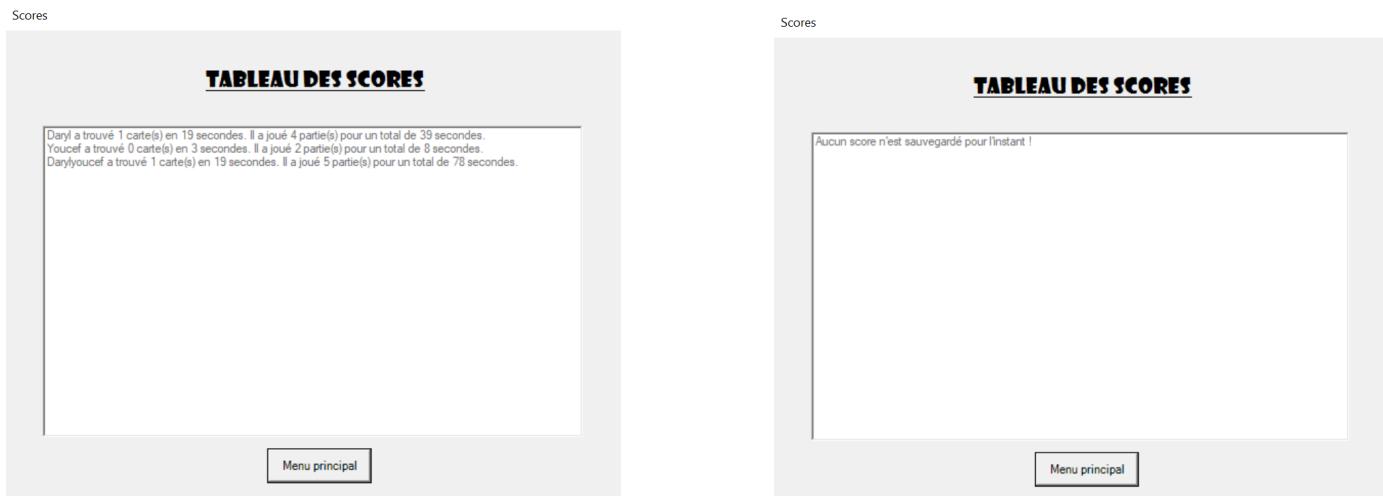


FormScores

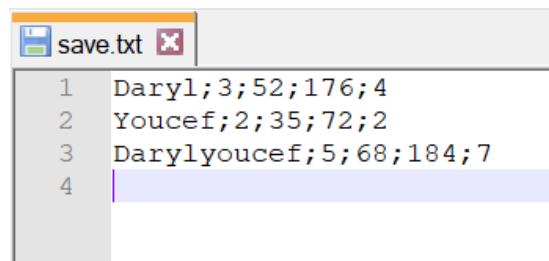
Les scores sont affichés avec une **ListBox**, non touchable par l'utilisateur. Les informations sont chargées depuis un tableau contenant les informations des joueurs. Nous affichons alors :

- Le nom du joueur
- Son meilleur score
- Le temps qu'il a mis pour ce score
- Le nombre de parties jouées
- Le temps total de jeu (en secondes)

Lorsque aucun score n'est enregistré, la liste affiche un autre message.



Le programme enregistre les scores dans le fichier “`save.txt`” sous la forme suivante.



```

save.txt
1 Daryl;3;52;176;4
2 Youcef;2;35;72;2
3 Darylyoucef;5;68;184;7

```

Tout est séparé par des points virgules. La première partie est le nom, la deuxième le score, la troisième le temps mit pour avoir le score, la quatrième le temps cumulé, et la cinquième le nombre de parties.

Par exemple, le record de Daryl est de 3 cartes, trouvées en 52 secondes. Il a un temps cumulé de 176 secondes, en 4 parties. Cette capture d'écran nous le confirme.



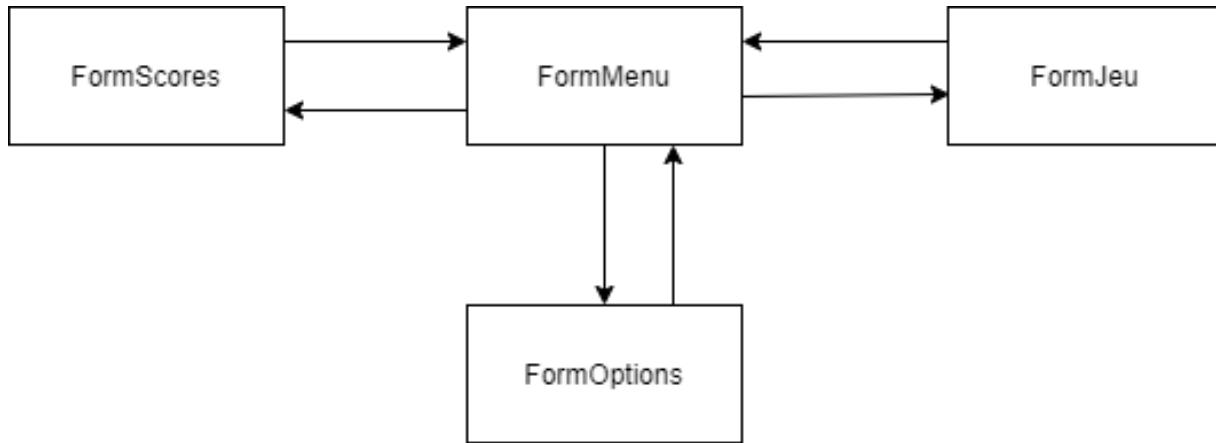
Scores

TABLEAU DES SCORES

Daryl a trouvé 3 carte(s) en 52 secondes. Il a joué 4 partie(s) pour un total de 176 secondes.
Yousef a trouvé 2 carte(s) en 35 secondes. Il a joué 2 partie(s) pour un total de 72 secondes.
Daryloucef a trouvé 5 carte(s) en 68 secondes. Il a joué 7 partie(s) pour un total de 184 secondes.

[Menu principal](#)

Schéma d'ordonnancement des formulaires





Bilan du projet

Difficultés rencontrée

Nous avons rencontré plusieurs problèmes dans la réalisation de ce projet que ce soit problèmes internes ou externes. Parmi les difficultés extérieures, il y a notamment le développement à terme de celui-ci tout en ayant des projets en parallèle (PPP, BPO, GPI, EGO) ainsi que des examens de période D à préparer.

Parmi les difficultés liées au projet lui-même, nous avons rencontré un très grand problème dès le début. Nous devions réfléchir à la structure du programme, à la logique à utiliser pour mélanger les cartes, les différencier parmis la liste des cartes, enregistrer le nombre de carrés trouvés et le temps mis.

La plus grande difficulté que nous avons rencontrée est la sauvegarde dans un fichier texte avec le module. En effet, il était impératif d'avoir un code factorisé et bien structuré pour gérer la sauvegarde. Nous avons eu du mal à gérer la détection du meilleur score : tout d'abord nous avons eu l'idée de remplacer le nombre de cartes ainsi que le temps directement dans le fichier texte, mais il s'avérait difficile de le faire. alors nous avons opté pour une solution plus simple : copier les informations de chaque joueur dans un tableau de "Joueur". De cette façon, l'accès aux données des joueurs est plus rapide et garanti. A la fin de la manipulation, le fichier est mis à jour intégralement.

A la fin, on a rencontré une ultime difficulté : impossible de mettre à jour le tableau des scores et la combobox après que le joueur a terminé de jouer. La seule solution qu'on a trouvé c'est de redémarrer le programme avec Application.Restart().

Apport du projet

Tout au long de ce projet, nous avons appris l'importance de l'organisation.

Nous avons par exemple utilisé le plugin LiveShare sur Visual Studio pour pouvoir collaborer en direct et nous avons aussi utilisé Github pour pouvoir effectuer un versionning de notre code.

De même au niveau des commentaires mais surtout au niveau des fichiers sources : il fallait tout structurer correctement et créer un maximum de procédures et fonctions. Bien entendu, nous avons grandement consolidé nos connaissances dans la structuration du code et acquis des connaissances en VB .NET .

Pistes d'améliorations

En termes d'améliorations, une implémentation de l'IA comme dit précédemment, ainsi que la possibilité de jouer avec d'autres types de pièces comme la dame, fou, cavalier ou pion.

Nous avons aussi pensé à créer une version plus élégante avec des images au lieu des boutons pour aller plus loin dans la customisation.

Une autre piste d'amélioration serait de créer une version multijoueur de ce jeu grâce à nos connaissances acquises en module de réseau et des classes de la bibliothèque standard (les Sockets TCP et UDP), mais la gestion de l'anti-triche est difficile; en effet, nous avons réussi à tricher sur notre propre jeu grâce au logiciel



de modification de mémoire : Cheat Engine. On a changé la valeur du score dans la mémoire. De plus, les sauvegardes sont enregistrés dans un fichier texte donc facilement modifiables : il faudra alors écrire en mode aléatoire.