



Dossier SAÉ Mastermind

Développement d'applications avec IHM

Programmation événementielle - Visual Basic .NET

26/05/2023

Rémi LEMAIRE (103)

Antoine LIEVRE (112)

Mouhamed BSIKRI (104)

Eva GENTILHOMME (103)

IUT Paris Rives de Seine (Université Paris Cité)

143 avenue de Versailles
75016 PARIS



IUT de Paris - Rives de Seine
Université Paris Cité

Table des matières

Table des matières	1
Introduction	2
Schéma d'ordonnancement	7
Documentation	8
Conclusion	12

Introduction

Notre application est un jeu vidéo se basant sur le jeu de société Mastermind. Le jeu oppose deux joueurs. L'un choisit, parmi des symboles définis, un ordre spécifique de symboles à deviner. Le second joueur doit deviner ce que le premier joueur a indiqué, dans un temps imparti et avec un nombre de coups maximum.

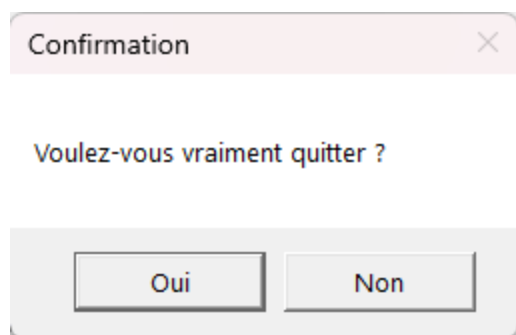
Dans notre application, ce jeu se présente sous la forme de plusieurs formulaires.

Premièrement, l'écran d'accueil :



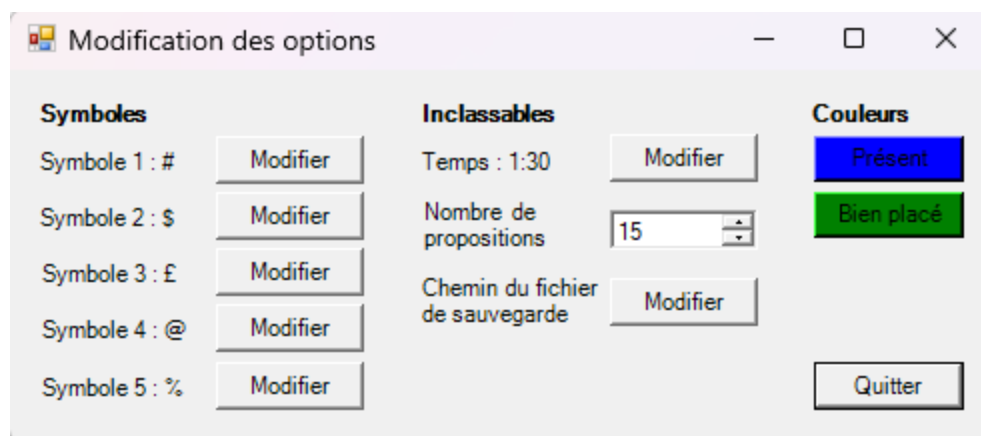
Ce menu propose différentes options. La première option consiste à ce qu'on puisse choisir soit un joueur déjà enregistré, soit de créer un nouveau joueur en rentrant un nouveau nom dans l'espace indiqué.

Le bouton "QUITTER" est simplement le bouton pour fermer l'application. Cette fermeture s'accompagne d'une fenêtre pour confirmer notre décision :



Lors de la fermeture, les joueurs sont enregistrés dans un registre. Le chemin d'accès à ce registre peut être modifié dans le formulaire des options. De plus, ce registre est lu au lancement de l'application afin de mettre en mémoire les joueurs pré-enregistrés.

Le bouton "Modification des options" amène vers un nouveau formulaire, permettant au joueur de modifier chaque aspect de la partie, il se présente de la façon suivante :

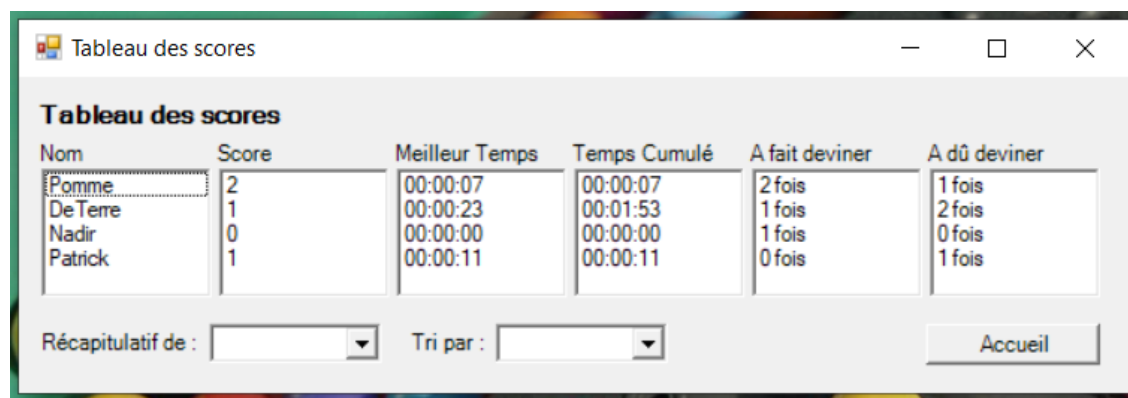


Chaque contrôle appelle une fenêtre demandant plus de précision, à l'exception du nombre de propositions maximum qui est modifié en permanence.

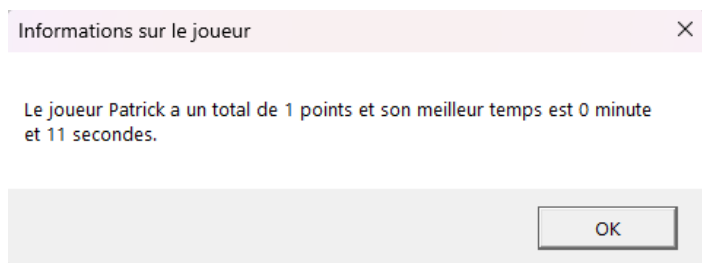
Le bouton "Tableau des scores" du formulaire d'accueil amène vers un autre formulaire, présentant les joueurs enregistrés, avec leur nom, score, meilleur temps, temps cumulé, nombre de fois en tant que joueur 1 et nombre de fois en tant que joueur 2.

Ce formulaire dispose aussi d'un bouton permettant de revenir au formulaire d'accueil.

Il se présente de la façon suivante :



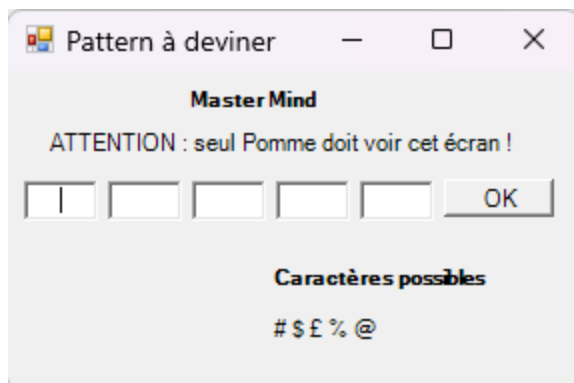
Il y a également la possibilité de trier l'affichage des joueurs selon trois critères distincts : leur meilleur temps, leurs scores et l'ordre alphabétique. Il est également possible d'avoir un récapitulatif pour un joueur donné, qui se présente de cette façon :



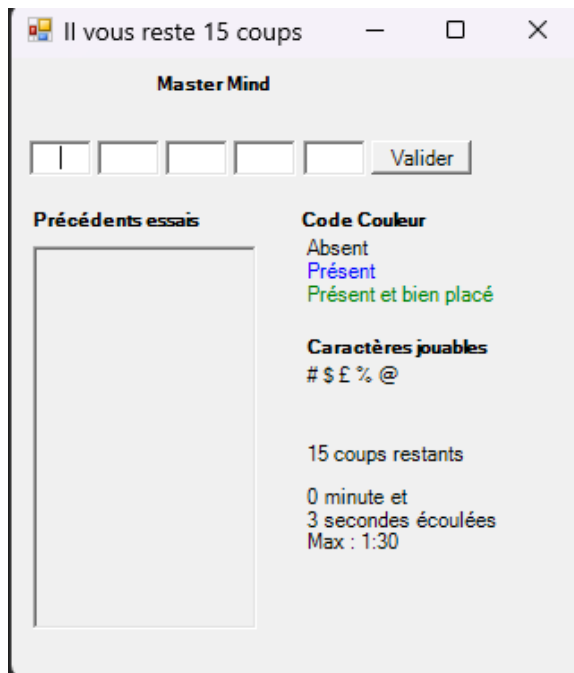
Le dernier bouton de l'accueil, le plus important, est le bouton "JOUER", qui permet de lancer une partie entre le joueur 1 et le joueur 2.

Le fait de cliquer sur ce bouton mène à un formulaire demandant au premier joueur de définir les symboles à faire deviner et leurs positions. Par défaut les symboles pouvant être utilisés sont : "# \$ £ % @", mais ils peuvent être modifiés via le formulaire de modification des options.

Ce formulaire se présente de la façon suivante :



Après que le joueur 1 ait rentré la combinaison secrète, il clique sur le bouton "OK", cette action lance la partie du côté du joueur 2 avec ce nouveau formulaire :

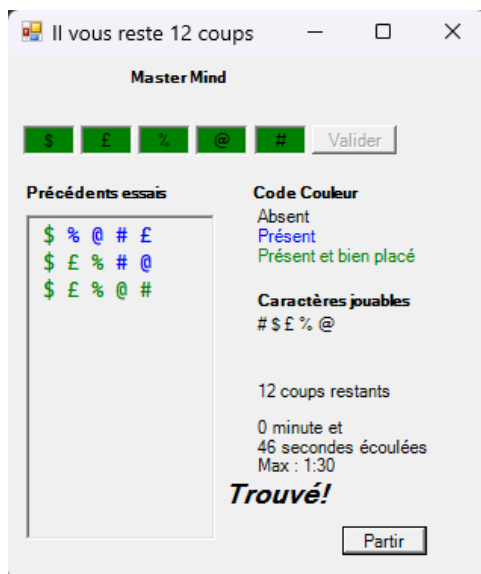



Le joueur 2 doit maintenant deviner la combinaison du joueur 1.

Pour chaque essai, le programme vérifie la combinaison proposée et attribue une couleur à chaque symbole selon un code prédéfini (qui peut également être modifié dans le formulaire des options). De plus, la combinaison colorée est également enregistrée dans un historique.

Pour le test j'ai rentré la combinaison "\$ £ % @ #" avec les symboles par défaut.

Après quelques essais jusqu'à ce que je trouve la bonne combinaison, le formulaire ressemble à ceci :



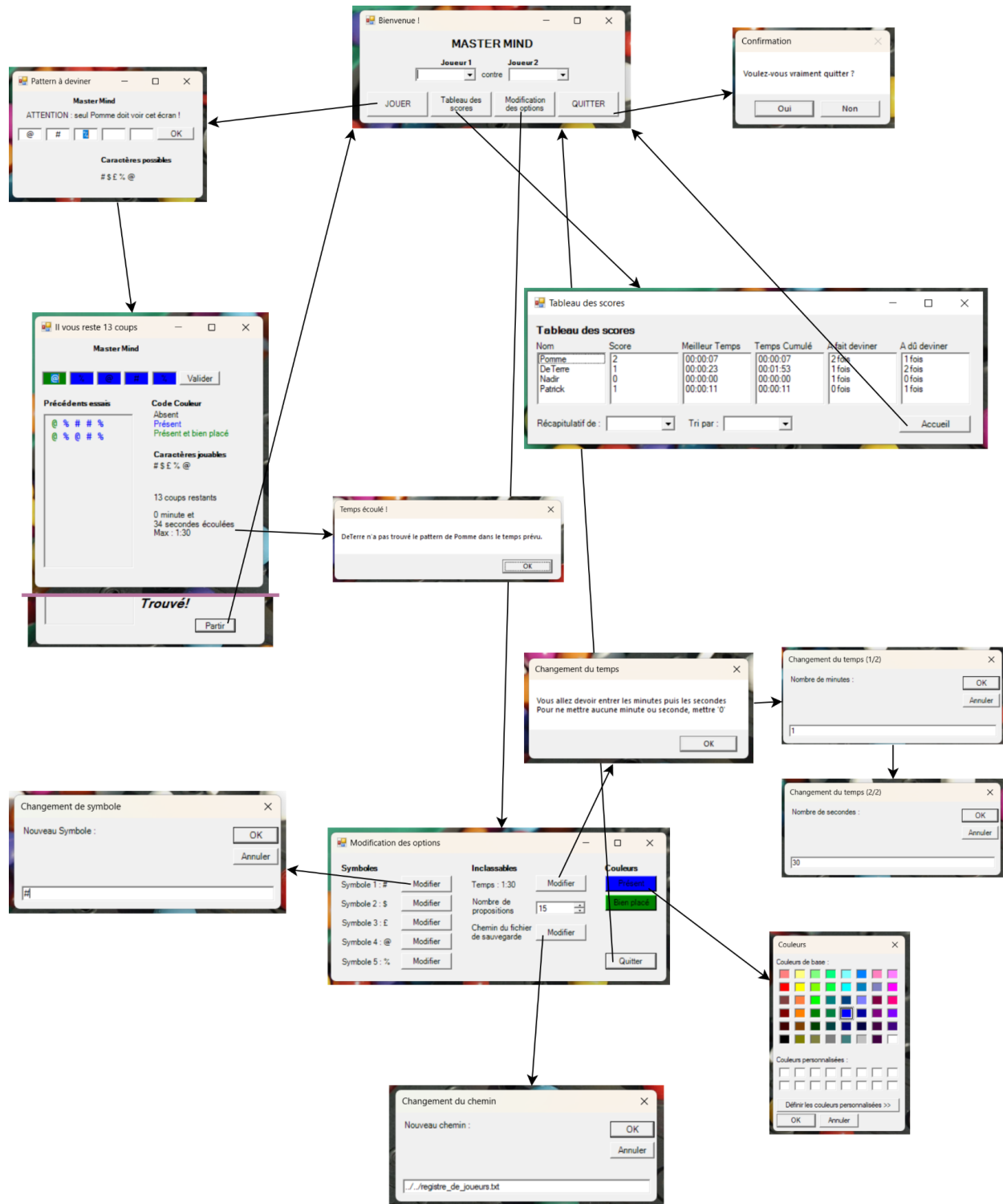


Après la victoire, la partie s'arrête et le bouton "Partir" apparaît et qui ramène au menu d'accueil. Si il y a défaite du joueur 2, une fenêtre indique que le joueur 1 n'a pas deviné le pattern du joueur 2 et ramène à l'accueil.

La partie s'arrête également si le temps est écoulé, ou si le deuxième joueur a utilisé tous les coups auquel il avait droit sans trouver la combinaison.

Schéma d'ordonnancement

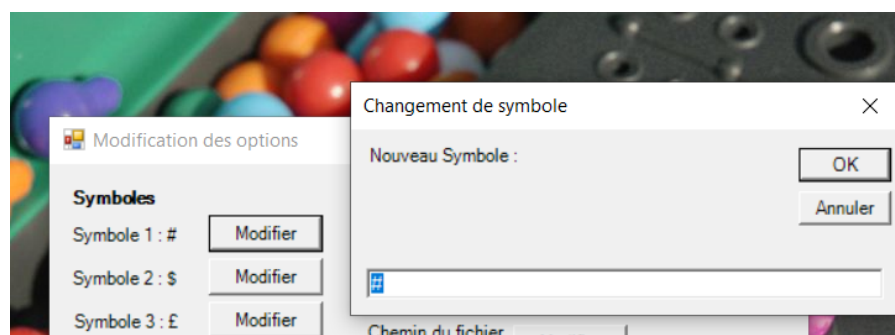
Voici le schéma d'ordonnancement de notre projet :



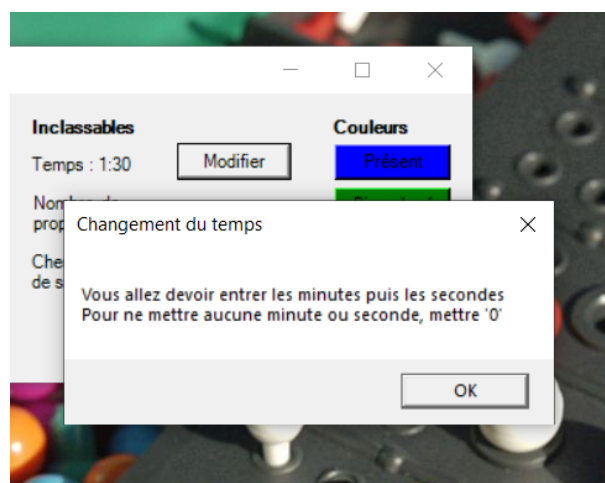
Documentation

La majorité des options que nous avons implémentées sont accessibles depuis le formulaire de modification des options.

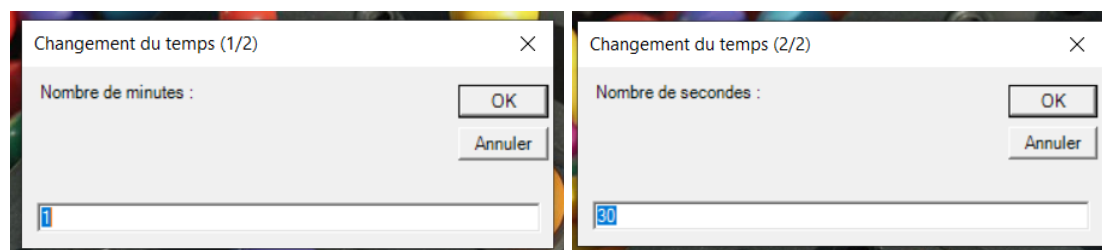
La première est la modification de symbole. Pour chacun des cinq symboles, un bouton permet d'accéder à l'écran de modification :




Il est également possible de modifier le temps alloué au second joueur pour trouver la combinaison :



Il est ensuite demandé de saisir les minutes puis les secondes.

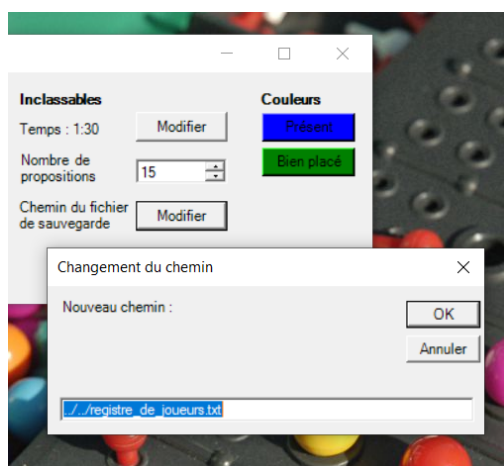


Le nombre maximal de propositions, quant à lui, se modifie directement dans le formulaire.



Nombre de propositions: 15

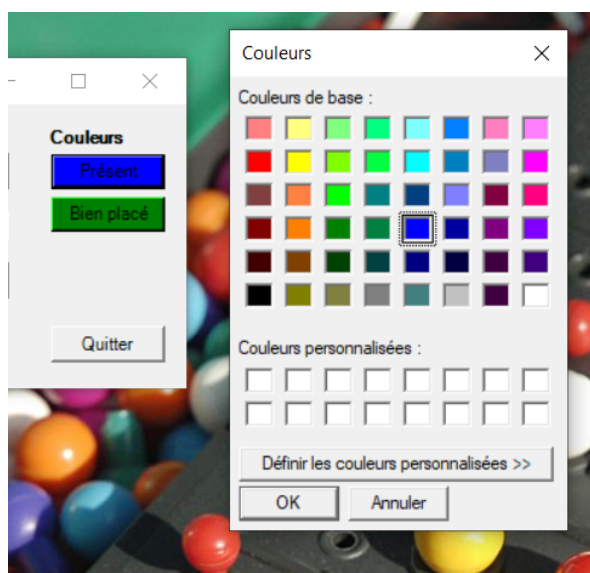
De plus, il est possible de changer le chemin d'accès au fichier de sauvegarde des joueurs



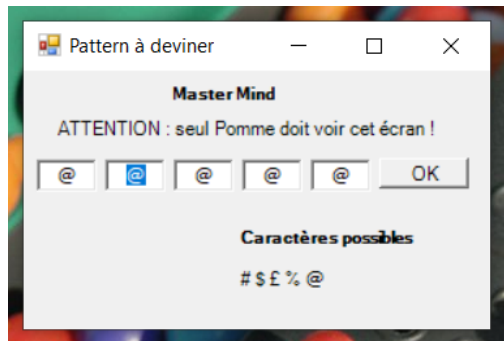
Cette modification implique la sauvegarde du fichier précédemment utilisé, et la lecture du nouveau fichier fourni.

Enfin, il est possible de changer les couleurs d'indicateur de présence d'un symbole dans la combinaison, ou de présence à la bonne place.

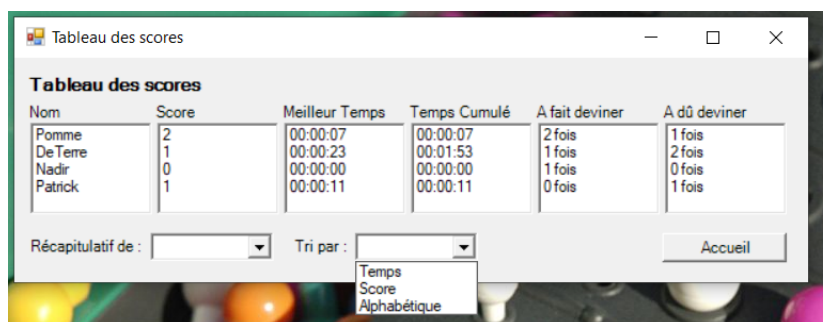
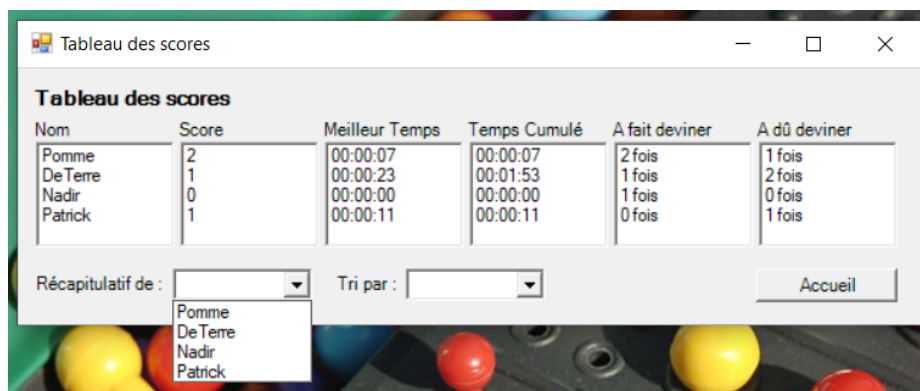
Pour cela un clic sur le bouton correspondant vous amènera sur un écran de sélection de couleur :



Lors de la saisie d'une combinaison, le fait de cliquer sur une case ou d'appuyer sur une touche correspondant à un symbole utilisable la réinitialisera et la remplacera immédiatement par le symbole voulu.




Enfin, sur le formulaire des scores, on peut soit trier les noms affichés de 3 différentes manières, soit obtenir un récapitulatif pour un joueur.



Pour communiquer avec les différents modules, les formulaires font appel aux méthodes (publiques) qui peuvent interagir avec les attributs (privés).

Prenons l'exemple du module de stockage des joueurs.



Une structure (privée) englobe tout d'abord les données qui définissent un joueur (son nom, son score, etc). Dans cette même structure des méthodes publiques sont présentes, permettant de rajouter un point, de modifier le temps, etc.

Hors de la structure se trouvent le tableau des joueurs, le compteur permettant de savoir à quel index ajouter un joueur, et le pas d'extension.

De plus il y a des méthodes, publiques, qui permettent d'interagir avec le tableau des joueurs, comme par exemple l'ajout d'un point à un joueur à partir de son nom, ou encore la récupération du temps minimal d'un joueur à partir de son index.

Les autres modules (stockage des couleurs, du chemin d'accès, etc...) utilisent le même principe de stockage de données (structure privée et méthodes publiques).

Conclusion

Nous sommes tous d'accord pour dire que ce projet a été très enrichissant, en nous permettant de mettre en application ce que nous avons appris en cours, mais aussi pour deux d'entre nous de découvrir un jeu très intéressant à programmer.

Mais personne n'est parfait, et donc notre projet non plus, il pourrait encore et toujours être amélioré, par exemple en stockant les options du jeu dans un fichier texte afin de les conserver entre les lancements du jeu, ou encore une option permettant de retrouver la liste des joueurs qui nous ont battus, afin de les affronter à nouveau.