

–Projet– Programmation événementielle

Visual Basic .NET

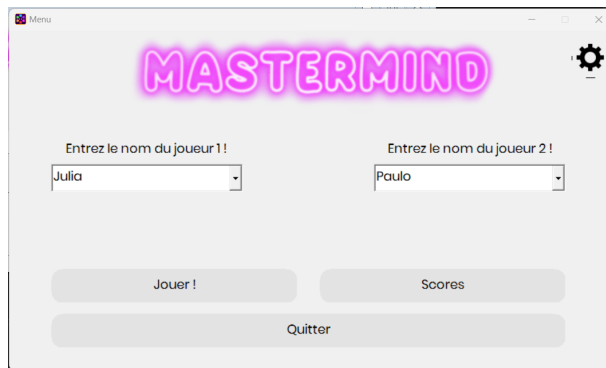
MASTERMIND

Table des matières

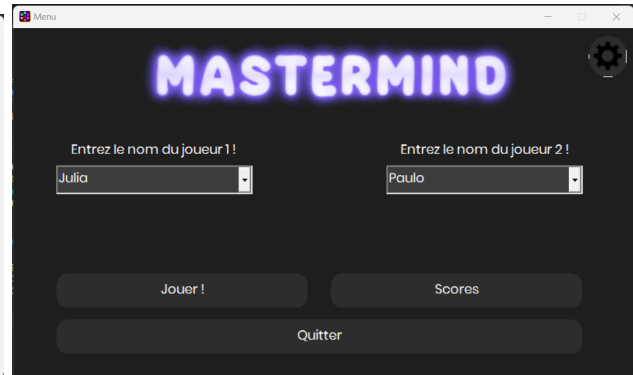
Introduction du projet	3
Schéma d'ordonnancement des formulaires	7
Fonctionnalités et options	9
Bilan du projet	12

Introduction du projet

Ce projet consiste à implémenter le jeu du Mastermind en programmation événementielle (langage Visual Basic avec le framework .NET)

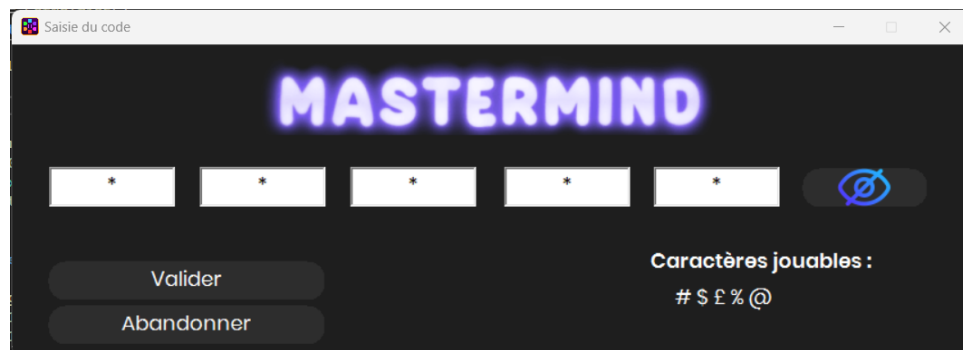


Affichage par défaut

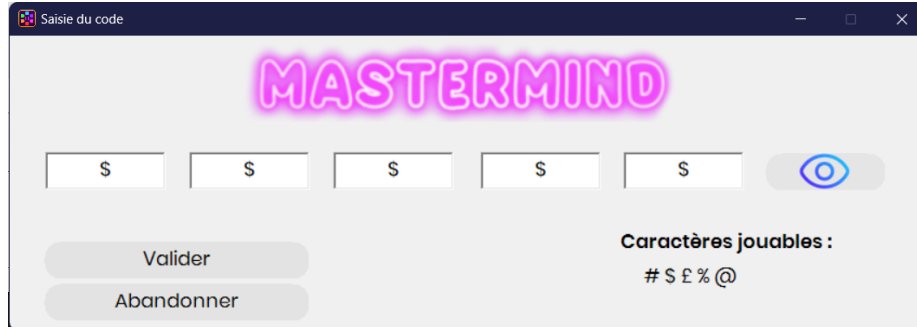


Thème sombre

Le Mastermind est un jeu de société pour deux joueurs dont le but est de trouver un code. Il se présente généralement sous la forme d'un plateau perforé de 10 rangées de quatre trous pouvant accueillir des pions de 8 couleurs différentes.

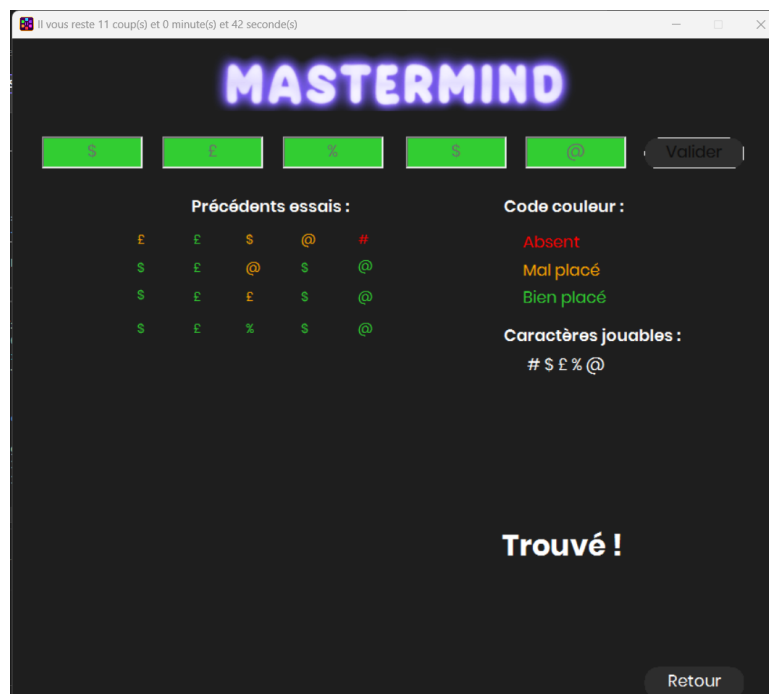


Thème sombre, par défaut le code est caché



Mode clair, en appuyant (maintenu) sur l'icône, le code devient visible

Il y a également des pions utilisés pour donner des indications à chaque étape du jeu. Il existe de nombreuses variantes, la nôtre offrant un code à 5 chiffres avec au maximum 15 essais et 1 minute 30 pour le deviner. A chaque tour, le joueur doit se servir de pions pour remplir une rangée selon l'idée qu'il se fait des pions dissimulés. Une fois les pions placés, l'autre joueur indique le nombre de pions de la bonne couleur bien placés, ainsi que le nombre de pions de la bonne couleur mais mal placés. Le joueur gagne cette manche s'il donne la bonne combinaison de pions sur la dernière rangée ou avant.

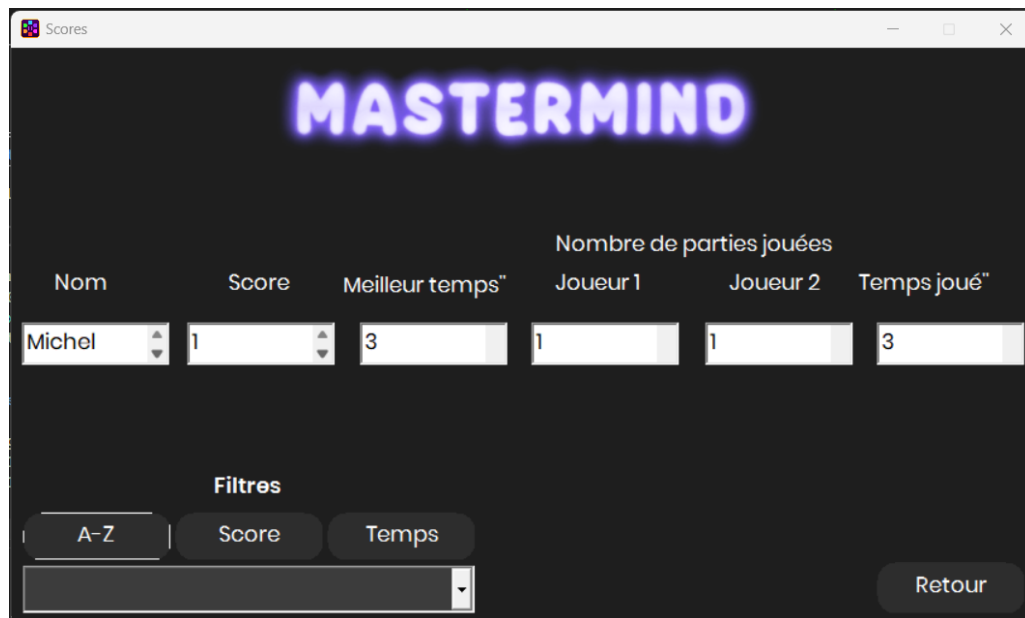


Mode sombre avec les couleurs par défaut



Mode clair, les couleurs ont été modifiées dans les paramètres.

A la fin du jeu, nous offrons aussi aux joueurs un tableau des scores basé sur leur nombre de points, meilleur temps, nombre de parties en tant que chacun des deux joueurs et leur temps total passé à deviner un code.

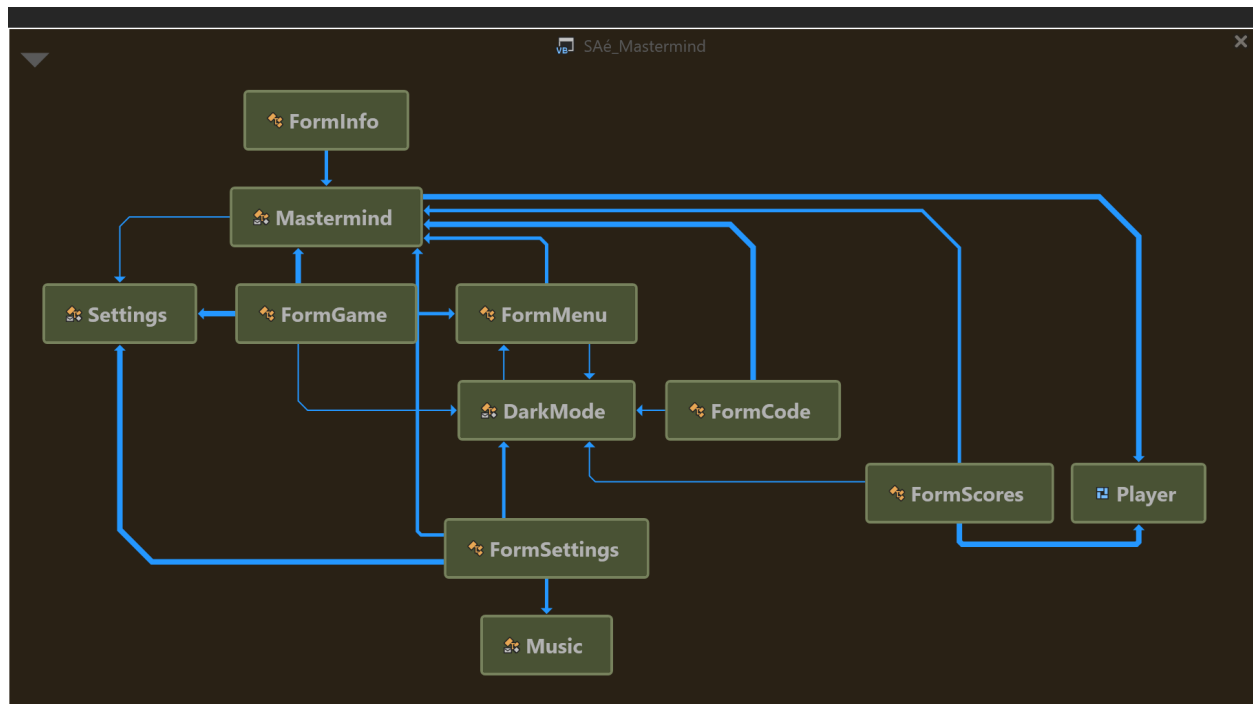


Mode sombre, affichage des statistiques des joueurs par ListBox synchronisées



Mode clair, une MsgBox s'affiche après avoir entré un nom dans la ComboBox

Schéma d'ordonnancement des formulaires



Dans le menu :

- En cliquant sur un bouton *btnStart* (événement *btnStart.Click*) le formulaire vérifie que toutes les conditions pour jouer sont remplies et envoie vers le formulaire *FormCode* afin que le premier joueur saisisse son code.
- En cliquant sur un bouton *btnScores* (événement *btnScores.Click*) le formulaire *FormMenu* redirige vers le formulaire *FormScores* permettant aux joueurs d'accéder au tableau des scores et visualiser leurs statistiques.
- En cliquant sur un bouton *btnSettings* (événement *btnSettings.Click*) le formulaire *FormMenu* redirige vers le formulaire *FormSettings* permettant aux joueurs d'accéder aux paramètres du jeu et le personnaliser.

Dans le tableau des scores :

- Lors de la fermeture du formulaire (événement *FormScores.FormClosing*) le formulaire *FormScores* redirige vers le formulaire *FormMenu* permettant aux joueurs de revenir au menu principal.

Lors de la saisie du code :

- Lors de la fermeture du formulaire (événement `FormCode.FormClosing`) le formulaire *FormCode* redirige :
 - Vers le formulaire *FormMenu* permettant aux joueurs de revenir au menu principal si le joueur a décidé de quitter.
 - Vers le formulaire *FormGame* si le joueur a validé, permettant au joueur de jouer une partie.

Dans le jeu :

- Lors de la fermeture du formulaire (événement `FormGame.FormClosing`) le formulaire *FormGame* redirige vers le formulaire *FormMenu* permettant aux joueurs de revenir au menu principal.

Dans les paramètres :

- En cliquant sur un bouton *btnInfo* (événement `btnInfo.Click`) le formulaire *FormSettings* redirige vers le formulaire *FormInfo* affichant les crédits du jeu.
- Lors de la fermeture du formulaire (événement `FormSettings.FormClosing`) le formulaire *FormSettings* redirige vers le formulaire *FormMenu* permettant aux joueurs de revenir au menu principal.

Dans les crédits :

- Lors de la fermeture du formulaire (événement `FormInfo.FormClosing`) le formulaire *FormInfo* redirige vers le formulaire *FormSettings* permettant aux joueurs de revenir aux paramètres.

Fonctionnalités et options

Lors de ce projet nous avons implémenté de nombreuses fonctionnalités.

Dans le menu :

- Saisie du nom des deux joueurs. Il est possible de retrouver le nom des joueurs ayant déjà fait une partie. Les joueurs sont chargés par le module principal qui parcourt un fichier texte de sauvegarde puis fournit aux ComboBox les noms.
 - Les deux joueurs ne peuvent pas avoir le même nom.
- Lancer la partie (autre formulaire), une fois que les deux joueurs auront saisi un nom.
- Quitter l'application, à confirmer après un message
 - L'option par défaut étant de ne pas quitter
- Affichage du tableau des scores dans un second formulaire

Pour choisir le code :

- 5 cases permettant au premier joueur de saisir sa combinaison.
 - Chacune est limitée à un seul des caractères autorisés et passe automatiquement à la suivante après saisie d'un caractère.
 - Le joueur peut utiliser plusieurs fois le même caractère pour composer sa combinaison.
- Un label rappellera au joueur les caractères autorisés pour la saisie. Ces caractères sont stockés dans le premier module et sont récupérés par le label.
- Une fois le code saisi, un bouton permettra de jouer et le code du premier joueur sera gardé en mémoire dans le module principal.

Dans le jeu :

- Jeu fonctionnel
- Le joueur disposera par défaut de 15 coups et 1 min 30 pour deviner le code.
 - Le bandeau de la fenêtre indique au joueur le nombre de propositions restantes ainsi que le temps restant pour deviner la combinaison.
- Cette variante du jeu révèle directement le statut de chaque caractère proposé en modifiant la couleur de chaque cases. Pour cela, le code proposé

par le second joueur est comparé à celui se trouvant dans le module principal.

- Historique des coups précédents : chaque proposition du second joueur sera ajoutée dans un composant graphique afin qu'il puisse les consulter ainsi que leur statut.
- Une fois la partie terminée un bouton permettra de revenir au formulaire d'accueil qui proposera automatiquement la partie suivante. Les rôles seront alors inversés et la partie finie est enregistrée dans un fichier texte de sauvegarde par le premier module qui récupérera les informations de la dernière partie.

Dans le tableau des scores :

- Affichage des noms des joueurs ayant joué au moins une partie, ainsi que leur score et statistiques par des ListBox synchronisées. Elles récupèrent ces informations du premier module.
- Les listes peuvent être triées, à la demande de l'utilisateur, par ordre alphabétique selon le nom des joueurs, ainsi que par ordre de meilleur scores ou de meilleurs temps.
- Recherche d'un joueur par son nom, puis d'affichage de ses statistiques dans une MsgBox.
- Toutes ces informations sont stockées dans un fichier (à la fin d'une partie comme mentionné précédemment) puis automatiquement rechargées lors d'un prochain démarrage par le module principal.

Nous avons aussi implémenté des options afin d'enrichir notre application :

Par exemple, nous avons pris en charge la fermeture des formulaire par bouton ou par fermeture de fenêtre (croix rouge) séparément, demandant une confirmation et redirigeant vers le menu principal.

Nous avons ajouté un icône au jeu mais aussi des logos faits par nos soins dans les formulaires. Les boutons sont maintenant arrondis pour des effets esthétiques.

Dans le menu :

- Bouton vers les paramètres (autre formulaire) avec une icône de roue.

Lors de la saisie du code :

- Icône sur le bouton permettant de visualiser le code (appui long plutôt que activé/désactivé) montrant un œil ouvert puis fermé.

Dans les paramètres :

- Modification du nombre d'essai possibles pour deviner le code (de 1 à 15) à l'aide d'une TextBox formatée
 - 2 caractères maximum, implémentation d'un KeyPress et impossibilité d'entrer certains chiffres / nombres
- Modification du temps disponible pour deviner le code (de 10 secondes à 5 minutes) avec une ScrollBar.
 - Il est aussi maintenant désactivable à l'aide d'une CheckBox.
- Modification des couleurs permettant de connaître le placement de chaque symbole lors du jeu à l'aide de plusieurs TextBox reliées à un ColorDialog.
- Création d'un mode sombre au jeu activable par CheckBox, celui-ci modifie aussi le logo du jeu sur les formulaires.
- Tous ces paramètres sont envoyés par un bouton dans un module Settings regroupant les paramètres personnalisés du joueur.
- Ajout d'une page de crédits

Bilan du projet

Ce projet s'est bien passé. Les premiers formulaires ont été faits assez rapidement. Nous avons commencé par faire le design des formulaires, comme ça il ne restait plus qu'à coder.

Au départ nous nous sommes focalisés sur la partie fonctionnelle en vue des recettes lors de notre séance de TP. De ce fait, nous avons un jeu plutôt simple lors des recettes. Nous n'avions pas de tâches désignées, nous avons travaillé en collaboration sur toutes les parties du projet. Bien que le projet était plutôt simple à ce stade là, nous avons tout de même implémenté de petites fonctionnalités ergonomiques au projet comme :

- Passage automatique d'une case à la prochaine lors de la saisie des codes
- Appui maintenu du bouton permettant de révéler le code pour plus de protection
- Affichage de certains éléments comme le nombre d'essais et le temps restant dans le bandeau supérieur de la fenêtre pour ne pas surcharger l'affichage.

A la suite des recettes nous avons commencé à développer des options et amélioré le design des formulaires. Nous avons ainsi modifié nos boutons afin d'arrondir les angles pour un effet plus esthétique, nous avons aussi ajouté un hover sur ceux-ci, modifiant la couleur du bouton lorsqu'il est survolé ou appuyé. Nous avons aussi ajouté des icônes pour effectuer certaines tâches.

Enfin, nous avons développé des options telles que :

- Changement des couleurs donnant des indices sur le statut des codes devinés
- Modification du nombre d'essais maximum du joueur 2
- Modification du temps maximum du joueur 2, voire sa suppression
- Activation d'un thème sombre
- Activation d'une musique

Pour ce qui est des améliorations, il serait bien de pouvoir nettoyer et optimiser le code car certaines fonctions et subs peuvent être assez longs. De même pour la gestion du texte de sauvegarde ; celui-ci pouvant provoquer des erreurs si modifié manuellement. Enfin, d'autres options pourraient être ajoutées afin de rendre le jeu plus attractif, par exemple des langues différentes, des changements de difficulté plus larges ou encore un outil de partage à la fin d'une partie.

En conclusion, ce projet de Mastermind est fonctionnel, regroupe des options de personnalisation pour le joueur et le code est structuré. Le fichier de sauvegarde est

opérationnel. Nous avons apprécié travailler sur ce projet et avons pu en apprendre plus sur la programmation événementielle.