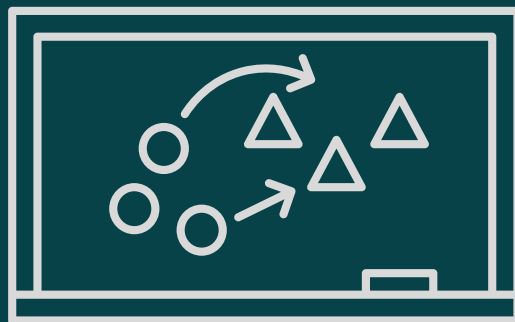




Zakary Melouk 111 , Mehdi Satouri
105, Rania Azmane 101, Geani
Santelli 101

RAPPORT

Logiciel : *Jeu du MasterMind*



START

2022/2023

SALE S2.O1

SOMMAIRE

- 01** INTRODUCTION
 - Formulaire
 - Simulation d'une partie
- 02** SCHÉMA
D'ORDONNANCEMENT
- 03** DOCUMENTATION DES
FONCTIONNALITÉS ET OPTIONS
- 04** CONCLUSION

Introduction

Ce projet consiste à développer une variante du jeu du Mastermind en utilisant Visual Basic .NET. Le Mastermind est un jeu de société inventé par Mordecai Meirowitz dans les années 1970, qui nécessite réflexion et déduction. Il se joue généralement sur un plateau perforé comprenant 10 rangées de quatre trous pouvant accueillir des pions de différentes couleurs. Le nombre de couleurs disponibles est de huit, telles que le rouge, le jaune, le vert, le bleu, l'orange, le blanc, le violet et le fuchsia. Le jeu utilise également des pions blancs et rouges (ou noirs) pour donner des indications à chaque étape.

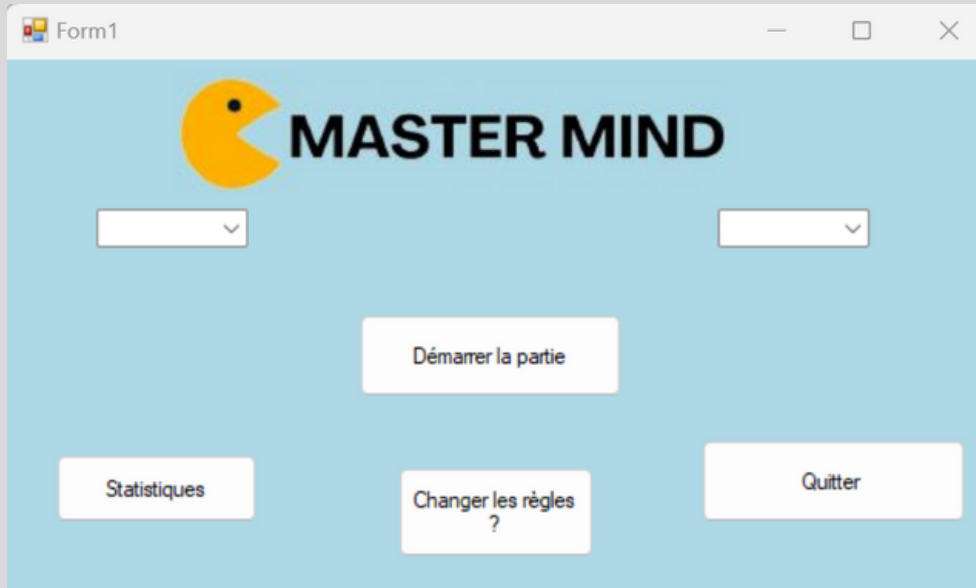
Le principe du jeu consiste en deux joueurs : l'un d'eux choisit une combinaison de caractères, tandis que l'autre joueur tente de deviner cette combinaison. À chaque tour, le joueur propose une combinaison de caractères et la valide, et le jeu fournit des indications sur les caractères bien placés, ainsi que les caractères présents mais mal placés. Le joueur actif utilise ces informations pour affiner sa déduction et proposer une nouvelle combinaison, tout en essayant de se rapprocher le plus possible de la solution.

Ce projet a été réalisé en équipe. L'application se déroulera en plusieurs étapes, commençant par un formulaire d'accueil où les joueurs pourront saisir leur nom. Une fois la partie lancée, le premier joueur choisira une combinaison de caractères que le second joueur devra deviner. Les joueurs auront un nombre limité de propositions et un temps imparti pour atteindre leur objectif.

L'application enregistrera les noms des joueurs, leurs scores, leurs meilleurs temps, ainsi que d'autres statistiques. Un formulaire affichera les statistiques des joueurs, permettant de trier les données par nom, score ou temps. Les options de jeu pourront également être personnalisées, offrant aux joueurs la possibilité de modifier les caractères utilisés, les couleurs utilisées pour les indications, les limites de temps et de propositions, etc.

FORMULAIRE

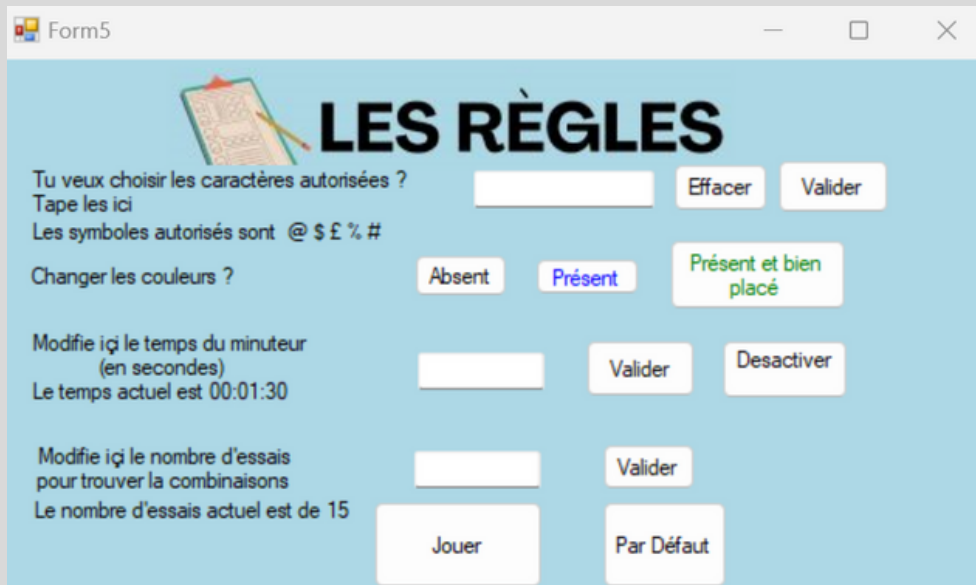
Formulaire d'Accueil



Le formulaire d'accueil de l'application présente les éléments suivants :

- Deux ComboBox pour saisir les noms des joueurs, avec suggestions de joueurs déjà enregistrés.
- Un bouton "Démarrer la partie" pour démarrer le jeu.
- Un bouton "Quitter " pour quitter l'application avec une boîte de dialogue de confirmation.
- Un bouton "Statistiques" pour visualiser les statistiques des joueurs.
- Un bouton "Changer les règles ?" pour modifier les règles comme l'utilisateur le souhaite.

Formulaire – Définition des règles

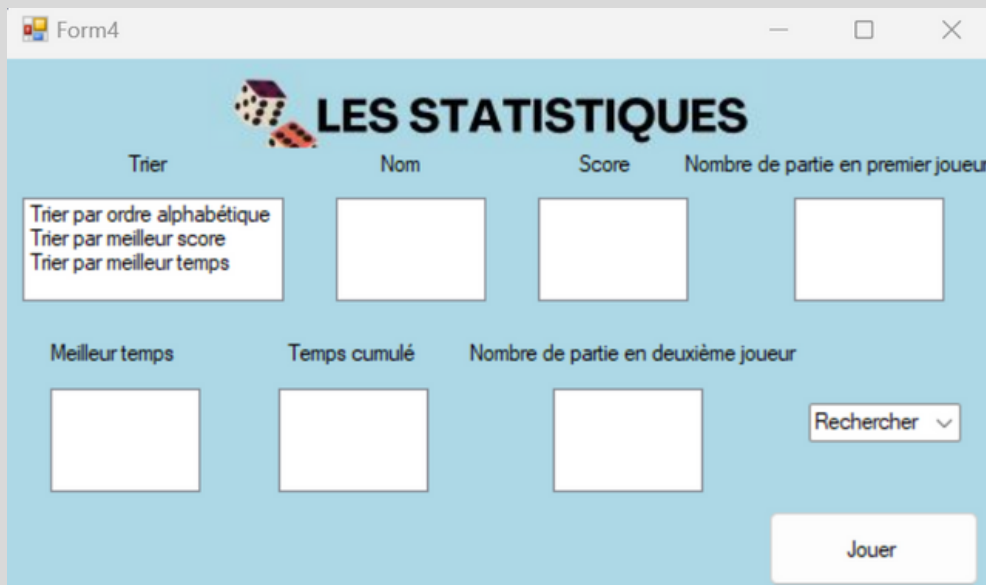


Le formulaire de définition des règles présente les éléments suivants :

- La liste des caractères utilisables pour former la combinaison à deviner.
- Les couleurs utilisées pour indiquer le résultat d'un coup au deuxième joueur.
- La possibilité de modifier ou désactiver la limite de temps pour que le deuxième joueur devine la combinaison.
- La modification de la limite de propositions pour le deuxième joueur.

FORMULAIRE

Formulaire – Statistiques des joueurs déjà enregistrés



Le formulaire des statistiques présente les éléments suivants :

- Une ListBox pour trier les joueurs par nom, score et meilleur temps.
- Une ListBox avec les scores.
- Une ListBox avec les nombre de partie en premier joueur.
- Une ListBox avec les nombre de partie en deuxième joueur.
- Une ListBox avec les meilleur temps
- Une ListBox avec les temps cumulé
- Une ComboBox pour rechercher le nom d'un joueur et afficher ses statistiques avec une MsgBox.
- Un bouton pour revenir au formualire d'accueil

Formulaire – Sélection des caractères à deviner par le joueur 1



Le formulaire de sélection de la combinaison à deviner présente les éléments suivants :

- Cinq TextBox pour saisir un caractère dans chacune d'entre elle.
- Un bouton pour valider la combinaison et passer au formulaire suivant.
- Un label qui affiche les caractères autorisés.

FORMULAIRE

Formulaire – Le joueur 2 doit trouver la combinaison

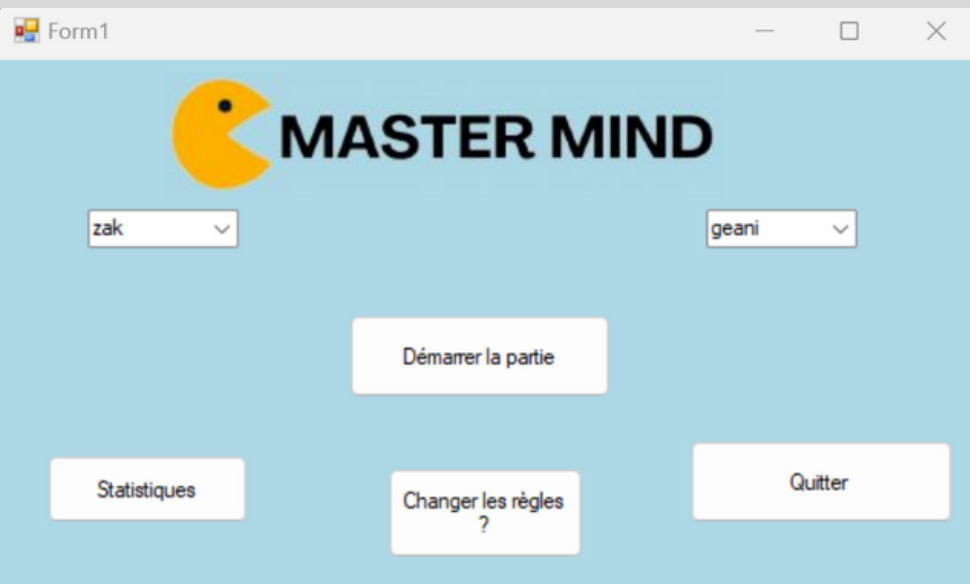


The screenshot shows a Windows application window titled "Form3". The interface has a light blue background. At the top, there are five empty text boxes for entering characters, followed by a "Guess" button. Below this, the interface is divided into two main sections. On the left, under the heading "Precedents essais", there is a large, empty rectangular area for displaying a history of attempts. On the right, under the heading "Code couleur", there are three color-coded labels: "Absent" (black), "Présent" (blue), and "Présent et Bien Placé" (green). Below these, under the heading "Caractères jouables:", there is a label showing the allowed characters "@ \$ £ % #". At the bottom right, there are two labels: "Nombre d'essais restants : 15" and "Le temps restant est de : 00:01:28".

Le formulaire dans lequel le joueur 2 devine la combinaison présente les éléments suivants :

- Cinq TextBox pour saisir un caractère dans chacune d'entre elle.
- Un bouton pour valider la combinaisons.
- Un label qui affiche les caractères autorisés.
- Un label qui affiche le code couleur.
- Une RichtextBox qui affiche l'historique des essais.

SIMULATION D'UNE PARTIE



Form1

MASTER MIND

zak ▼

geani ▼

Démarrer la partie

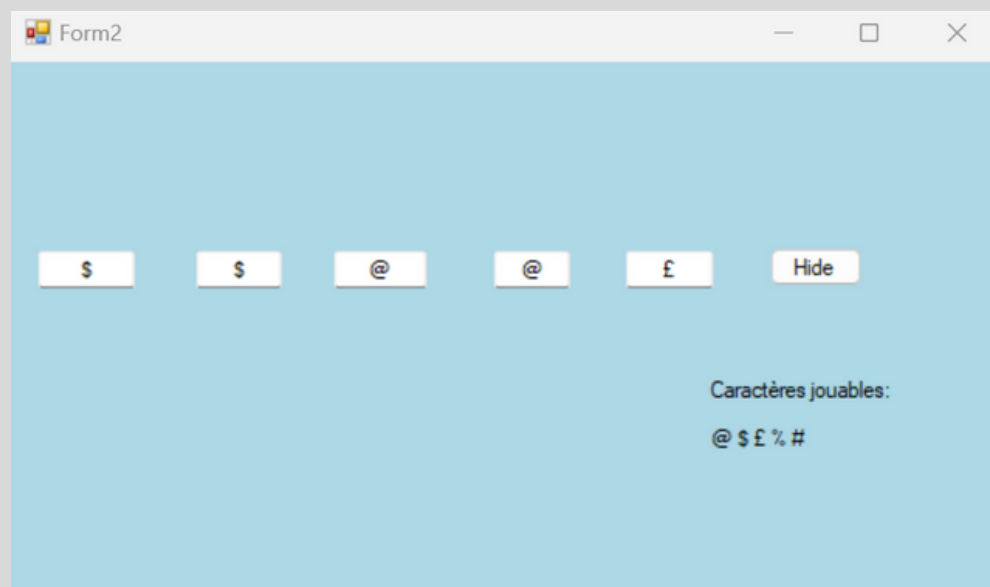
Statistiques

Changer les règles ?

Quitter

On commence par rentrer deux prénoms pour les deux joueurs, on appuie ensuite sur le bouton "Démarrer la partie"

Le premier joueurs rentre la combinaison qu'il souhaite et appuie ensuite sur le bouton "Hide"



Form2

\$ \$ @ @ £ Hide

Caractères jouables:
@ \$ £ % #

SIMULATION D'UNE PARTIE

The screenshot shows the 'Form3' game window. At the top, there are five colored buttons: a white button with '#', a blue button with '@', a blue button with '£', a green button with '@', and a blue button with '\$'. To the right of these buttons is a 'Guess' button. Below the buttons, the interface is divided into three main sections. On the left, under the heading 'Precedents essais', there is a list of previous guesses: '# @ £ @ \$'. In the center, under the heading 'Code couleur', there are three status indicators: 'Absent' (black), 'Présent' (blue), and 'Présent et Bien Placé' (green). On the right, under the heading 'Caractères jouables:', there is a list of playable characters: '@ \$ £ % #'. Below this list, it says 'Nombre d'essais restants : 14' and 'Le temps restant est de : 00:00:56'.

Ici, le troisième tente
une première fois de
trouver la
combinaison

Nous pouvons voir
qu'il a tenté une
deuxième fois de
deviner la
combinaison

The screenshot shows the 'Form3' game window after a second guess. The top buttons are now: a green button with '\$', a blue button with '@', a blue button with '£', a green button with '@', and a blue button with '@'. The 'Guess' button is still present. The 'Precedents essais' section now shows two rows of guesses: '# @ £ @ \$' and '\$ @ £ @ @'. The 'Code couleur' section remains the same. The 'Caractères jouables:' section now shows a list of playable characters: '@ \$ £ % #'. Below this list, it says 'Nombre d'essais restants : 13' and 'Le temps restant est de : 00:00:19'.

SIMULATION D'UNE PARTIE

Form3

\$ \$ @ @ £ Guess

Précédents essais

| | | | | |
|----|----|---|---|----|
| # | @ | £ | @ | \$ |
| \$ | @ | £ | @ | @ |
| \$ | \$ | @ | @ | £ |

Code couleur

Absent

Présent

Présent et Bien Placé

Caractères jouables:

@ \$ £ % #

Trouvé!

Bye

Les symboles sont tous de couleur vertes, le joueur 3 a trouvé la combinaison gagnante, il clique ensuite sur le bouton "Bye", qui va renvoyer sur le formulaire d'accueil

On peut voir sur le formulaire d'accueil que les prénoms sont inversés, nous pouvons démarrer une nouvelle partie en inversant les rôles

Form1

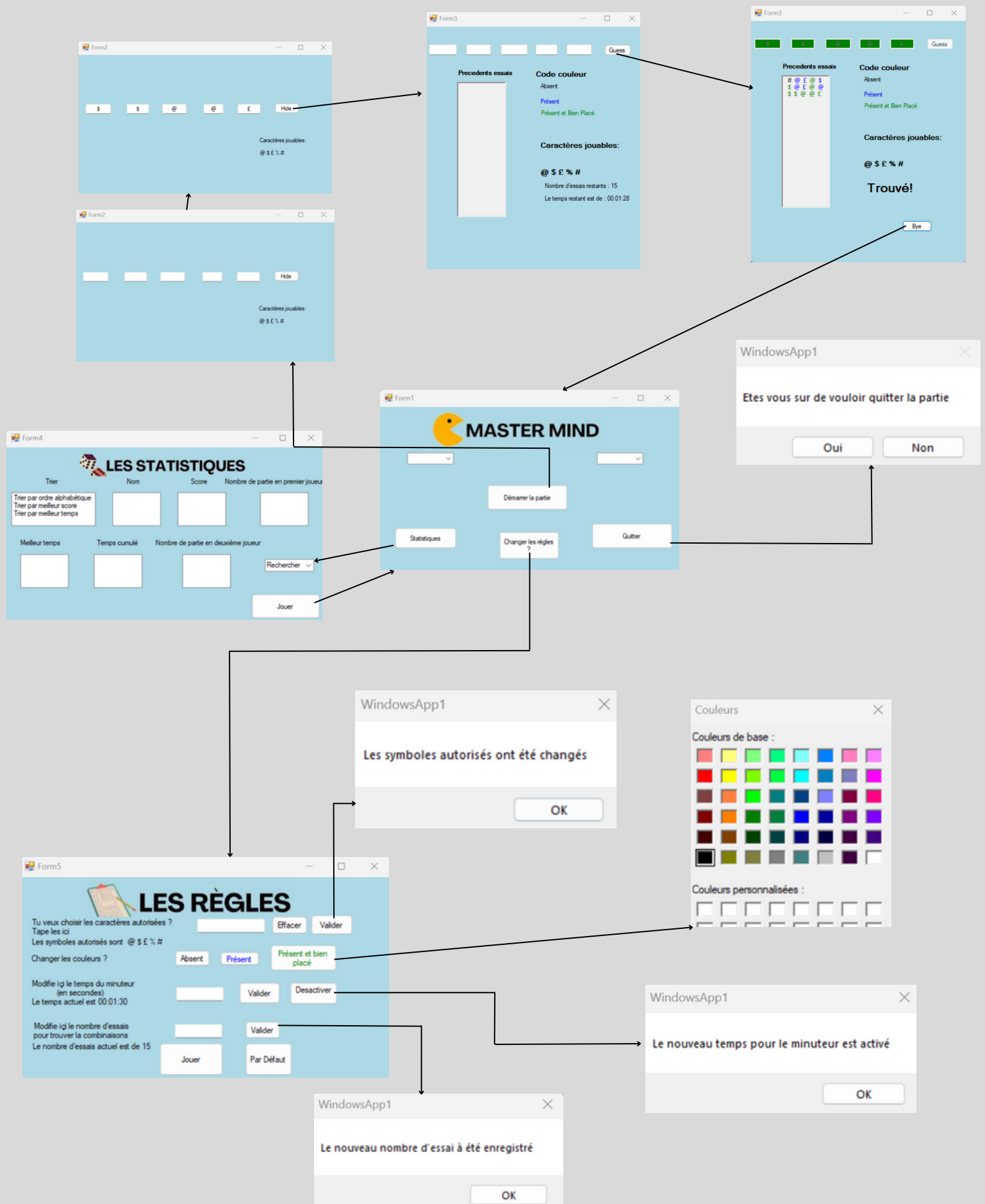
 **MASTER MIND**

geani zak

Démarrer la partie

Statistiques Changer les règles ? Quitter

SCHÉMA D'ORDONNANCEMENT



DOCUMENTATION DES FONCTIONNALITÉS ET OPTIONS

Sur notre projet, nous avons réussi à coder toutes les options demandées. D'abord, notre programme peut afficher les statistiques de chaque joueur grâce au formulaire 4 en récupérant les joueurs stockés dans le module LesJoueurs. Il y a également 3 fonctions du module LesJoueurs qui permettent de trier les joueurs par score, par ordre alphabétique et par meilleur temps. Le formulaire 4 les récupère et s'occupe de l'affichage. Lorsqu'on quitte le jeu, les joueurs sont enregistrés dans un fichier texte grâce à la fonction `Ecriture()` du module LesJoueurs. A son ouverture, les joueurs sont chargés automatiquement dans le jeu grâce à la fonction `Lecture()` également présente dans le même module. Pour améliorer le design, on a mis tous les formulaires en bleu avec le code suivant:

```
BackColor = Color.LightBlue
```

On a également rajouter des images pour les titres des formulaires grâce à des `PictureBox`. Vous pouvez visualiser le résultat sur les captures d'écran des formulaires au début de notre rapport.

Nous avons réussi à ajouter plusieurs options supplémentaires. Nous avons décidé de créer un deuxième module pour pouvoir gérer toutes les options sur les règles que nous avons appelé "LesRègles". Il a plusieurs attributs:

```
Private NouvelleCouleurAbsent As Boolean = False
```

```
Private PasDeTemps As Boolean = False
```

```
Private NbEssais As Integer
```

```
Private Symboles As String()
```

```
Private Temps As TimeSpan
```

```
Private CouleurAbsent As Color
```

```
Private CouleurPresent As Color
```

```
Private CouleurBienPlace As Color
```

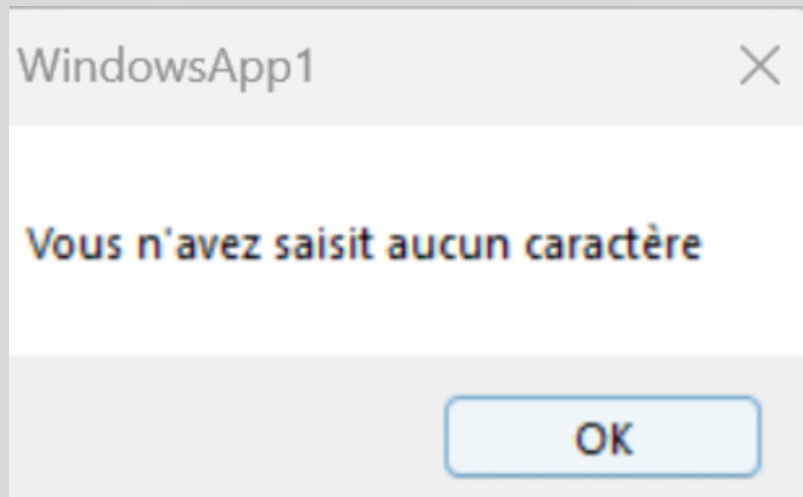
DOCUMENTATION DES FONCTIONNALITÉS ET OPTIONS

Toutes les interactions avec un module qui seront mentionnées porteront sur ce nouveau module. Durant le chargement du premier formulaire, donc au lancement du jeu, la fonction “initialise” de ce module va se lancer pour pouvoir initialiser les règles classiques :

```
Public Sub Initialise()  
    NbEssais = 15  
    Temps = New TimeSpan(0, 1, 30)  
    CouleurBienPlace = Color.Green  
    CouleurPresent = Color.Blue  
    NouvelleCouleurAbsent = False  
    ReDim Symboles(4)  
    Symboles(0) = "@"  
    Symboles(1) = "$"  
    Symboles(2) = "£"  
    Symboles(3) = "%"  
    Symboles(4) = "#"  
End Sub
```

Il y a un formulaire pour pouvoir changer les règles, le formulaire 4, qui s'ouvre depuis un bouton sur le formulaire d'accueil. Dans ce formulaire, on peut changer plusieurs choses. D'abord, les symboles autorisés. Si la personne écrit des symboles dans la première textbox et appuie sur le bouton “Valider”, on va d'abord vérifier si la text box n'est pas vide, sinon ce message s'affiche:

DOCUMENTATION DES FONCTIONNALITÉS ET OPTIONS



Dans le cas contraire, le bouton valider va exécuter cette ligne de code:

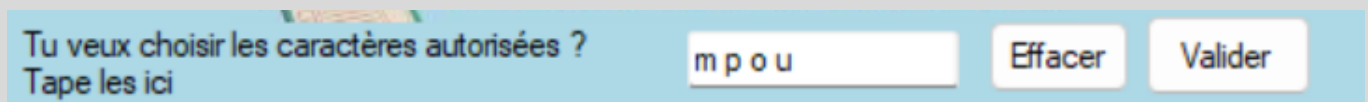
```
Dim NouveauxSymboles As String() = Split(TextBox1.Text,  
" ")  
SetSymboles(NouveauxSymboles)  
MsgBox("Les symboles autorisés ont été changés")
```

Il va appeler la fonction SetNouveauxSymboles du module avec en argument la nouvelle liste qu'il vient de créer. Cette liste va être affecté comme la liste de symboles autorisés dans le module, voici le code de la fonction:

```
Public Sub SetSymboles(Symboles2 As String())  
    Symboles = Symboles2  
End Sub
```

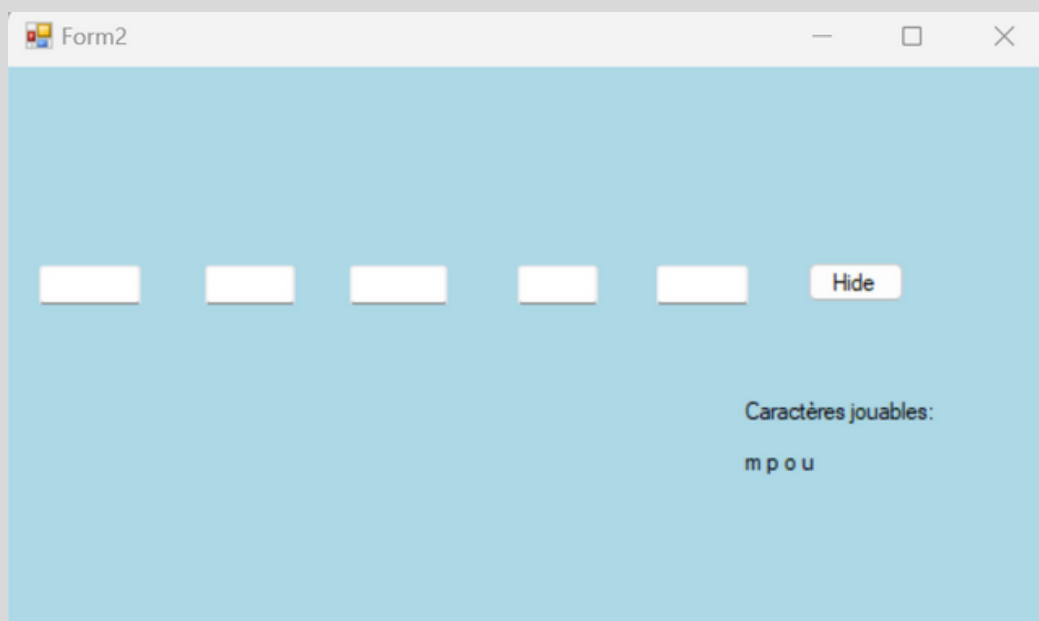
DOCUMENTATION DES FONCTIONNALITÉS ET OPTIONS

Ensuite, lors du chargement du formulaire 2, on va récupérer les symboles grâce à la fonction `GetSymboles()` du module, cette fonction va retourner la liste de symboles autorisés. Par exemple, si je tape ces symboles la:



Tu veux choisir les caractères autorisés ?
Tape les ici

Et qu'ensuite j'appuie sur le bouton "Valider". Voici ce que le formulaire 2 va afficher:

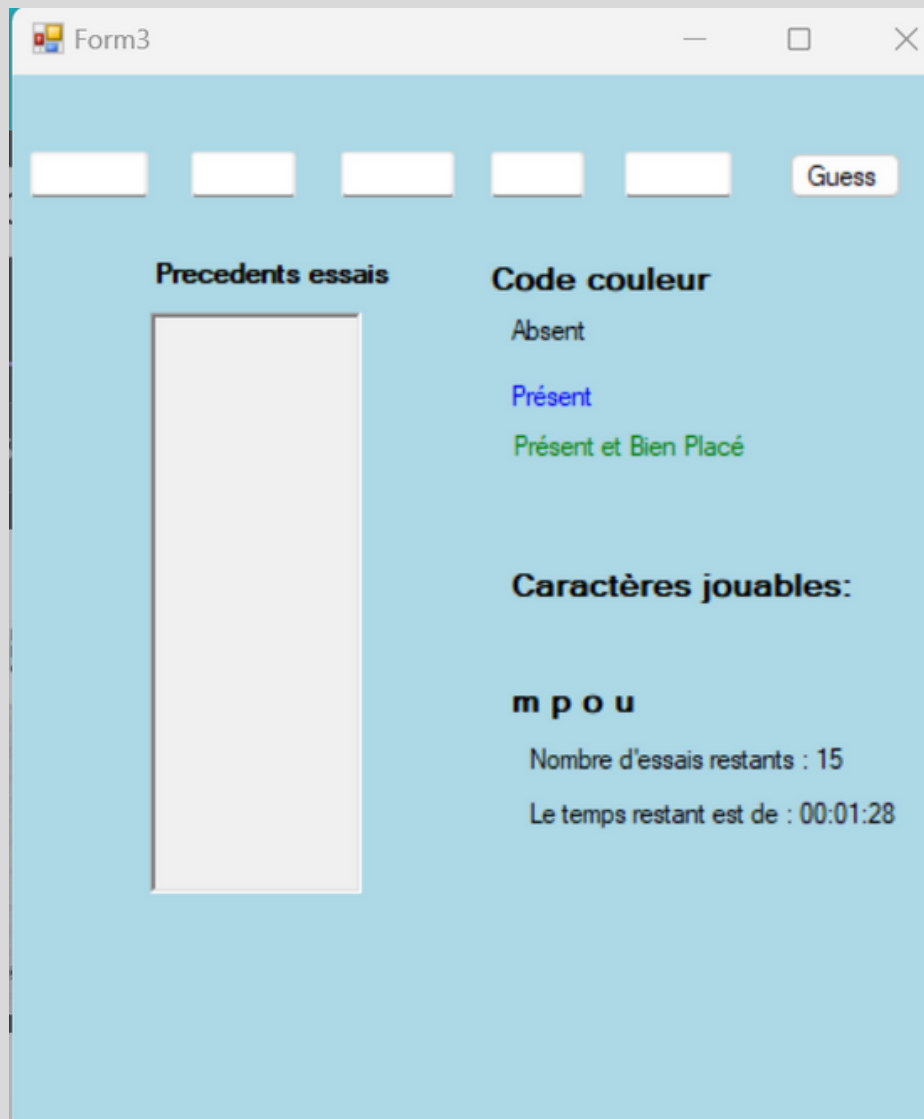


Form2

Caractères jouables:
m p o u

DOCUMENTATION DES FONCTIONNALITES ET OPTIONS

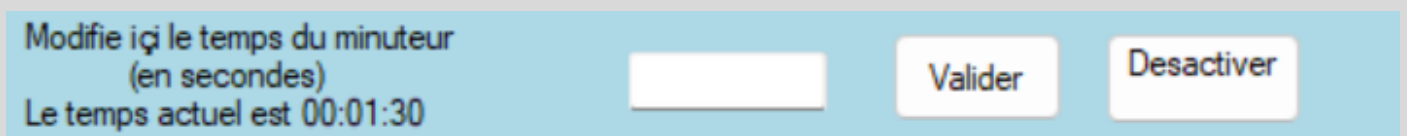
On peut voir en bas à droite que les caractères jouables ont changés. La saisie de tout autre caractère sera bloquée. Ensuite dans le chargement du troisième formulaire, on va également appeler la fonction `GetSymboles()` du module pour récupérer les symboles . En continuant avec les mêmes caractères que dans l'exemple ci-dessus, voici l'affichage du formulaire 3:



On voit en bas à droite que les caractères jouables correspondent aux caractères choisis précédemment. La saisie de tout autre caractère sera bloquée.

DOCUMENTATION DES FONCTIONNALITÉS ET OPTIONS

On peut aussi changer le temps avec une textbox dans le formulaire 5, l'utilisateur doit entrer un temps en secondes puis appuyer sur valider:



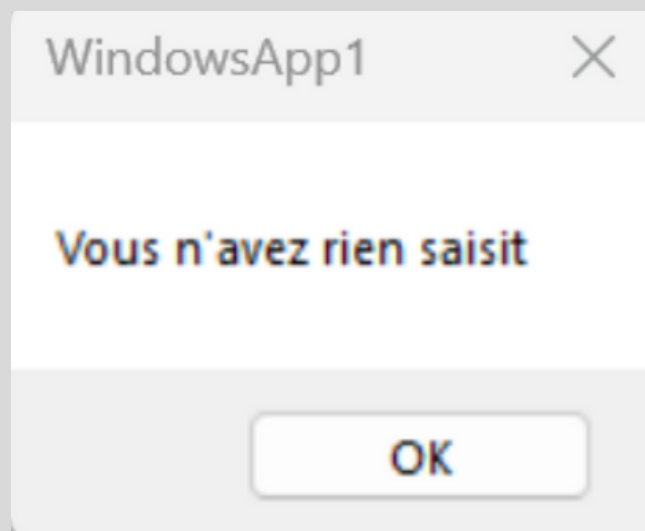
Modifie ici le temps du minuteur
(en secondes)

Le temps actuel est 00:01:30

Valider

Desactiver

Lorsque l'utilisateur appuie sur la validé, il y aura d'abord une vérification sur le contenu de la textbox, si elle est vide, ce message apparaîtra:



DOCUMENTATION DES FONCTIONNALITÉS ET OPTIONS

Dans le cas contraire, c'est exactement le même fonctionnement que pour les nouveaux symboles :

```
MsgBox("Le nouveau temps pour le minuteur est activé")
    Dim temps As TimeSpan = New TimeSpan(0, 0,
    TextBox2.Text)
    SetTemps(temps)
```

On peut également désactiver le minuteur en appuyant sur l'autre bouton "Desactiver le minuteur". Ce bouton va executer ce code:

```
MsgBox("Le minuteur est désactivé")
SetPasDeTemps(True)
```

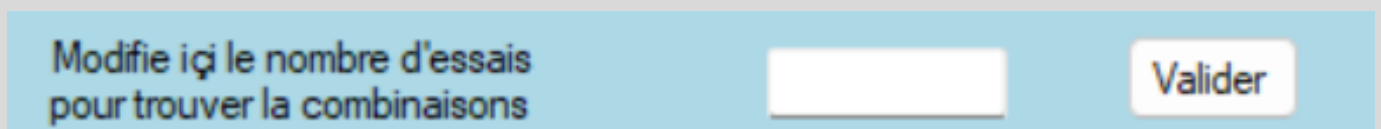
La fonction SetPasDeTemps() du module permet de changer l'état du booléen PasDeTemps du module à true en lui passant True en argument. Ensuite au chargement du troisième formulaire, on va récupérer le temps grâce à la fonction GetTemps du module.

Il faut savoir qu'à chaque seconde, notre minuteur appuie sur un bouton qui incrémente le compteur du temps et qui ensuite le compare au temps limite, s'il est égal au temps limite, il va alors faire perdre la partie au joueur. Cependant, avant de faire tout ça il vérifie s'il le temps n'a pas été désactivé en évaluant la valeur du booléen Pas De Temps à l'aide de la fonction getpasdetemps du module. Si le booléen est à true, il va directement quitter le programme:

```
If (GetPasDeTemps.Equals(True)) Then
    Return
End If
```

DOCUMENTATION DES FONCTIONNALITÉS ET OPTIONS

Il ne va donc pas exécuter la suite qui elle permettait de faire perdre ou non le joueur qui devine la combinaison.
On peut aussi changer le nombre de combinaisons à toujours dans le formulaire 5 :



Modifie ici le nombre d'essais
pour trouver la combinaison

Valider

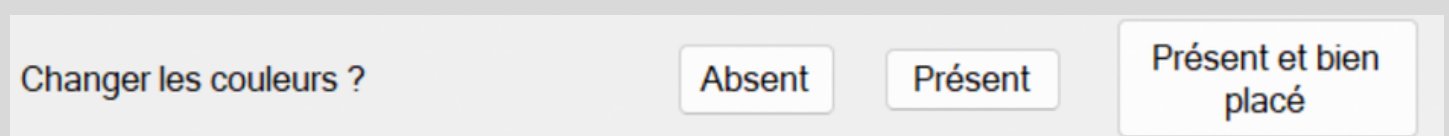
Lorsque l'utilisateur valide, on va d'abord vérifier que la textbox n'est pas vide et ensuite lui afficher un message pour lui confirmer que cela a été validé tout en mettant à jour le nouveau nombre d'essais dans le module :

```
MsgBox("Le nouveau nombre d'essai à été enregistré")  
SetNbEssais(TextBox3.Text)
```

La fonction SetNbEssais du module LesRègles va permettre de changer le nombre contenue dans la variable NbEssais du module.

Ensuite lors du chargement du formulaire 3, on va simplement récupérer cette valeur grâce à la fonction GetNbEssais() du module.

Enfin la dernière option est la modification des couleurs pour l'affichage des caractères absent, présent et présent et bien placé:



Changer les couleurs ?

Absent

Présent

Présent et bien
placé

DOCUMENTATION DES FONCTIONNALITÉS ET OPTIONS

Il y a 3 boutons qui chacun possède sa ColorDialog dans laquelle il va être invité à sélectionner une nouvelle couleur comme ceci:



Pour les boutons “Présenté” et “Présent et bien placé”, les 2 fonctions `SetCouleurBienPlace` et `SetCouleurPresent` du module vont changer la valeur des variables `CouleurPresent` et

`CouleurBienPlace` également présente dans le module en leur attribuant respectivement les couleurs sélectionnées dans leur `ColorDialog`. Pour le bouton absent, il y d’abord la fonction `SetNouvelleCouleurAbsent` qui va elle changer l’état du booleen `NouvelleCouleurAbsent` en le mettant à true puisqu’on va passer l’argument `True` à la fonction comme ceci:

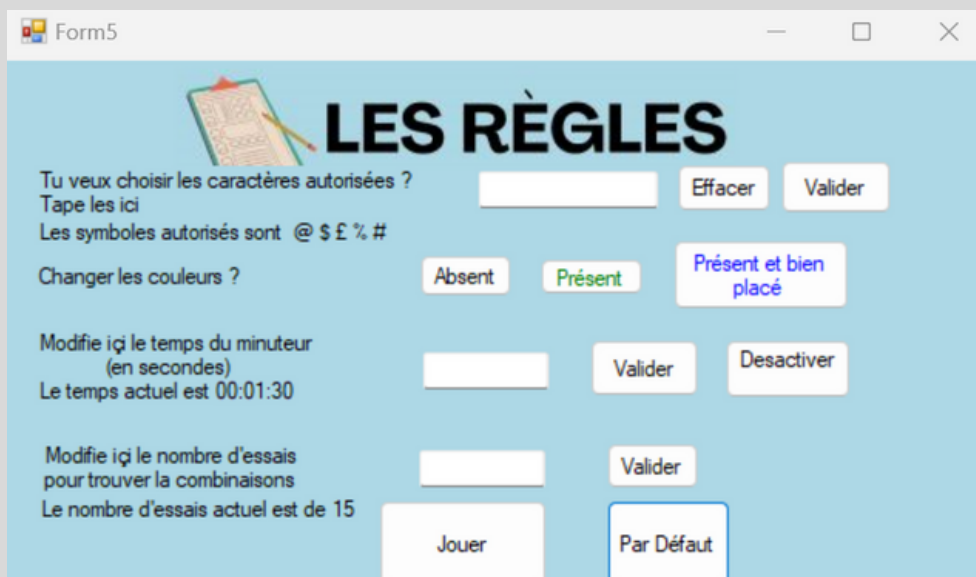
DOCUMENTATION DES FONCTIONNALITÉS ET OPTIONS

```
SetCouleurAbsent(ColorDialog1.Color)  
SetNouvelleCouleurAbsent(True)
```

On va ensuite, comme pour les 2 autres boutons, changer la valeur de la variable `CouleurAbsent` du module grâce à la fonction `SetCouleurAbsent` en lui passant en argument la couleur sélectionnée.

Ensuite, lorsque que sur le troisième formulaire on appuie sur `guess`, on va récupérer les nouvelles couleurs grâce aux trois fonctions du module `LesRègles` : `GetCouleurAbsent`, `GetCouleurPresent` et `GetCouleurBienPlace`. Cependant, pour la couleur “absent”, on va d’abord vérifier l’état du booléen `NouvelleCouleurAbsent` à l’aide de la fonction `GetNouvelleCouleurAbsent`. S’il est à `true`, on récupère la couleur.

SI l’utilisateur veut remettre les règles classiques du jeu, il peut appuyer sur le bouton “Par défaut” situé en bas du formulaire:



The screenshot shows a window titled "Form5" with a light blue background. At the top, there is a clipboard icon and the title "LES RÈGLES". Below this, the form contains several sections for configuring game rules:

- Tu veux choisir les caractères autorisés ?**: A text input field with "Tape les ici" below it. To the right are "Effacer" and "Valider" buttons. Below the input, it says "Les symboles autorisés sont @ \$ £ % #".
- Changer les couleurs ?**: Three buttons labeled "Absent" (white), "Présent" (green), and "Présent et bien placé" (blue).
- Modifie ici le temps du minuteur (en secondes)**: A text input field with "Le temps actuel est 00:01:30" below it. To the right are "Valider" and "Desactiver" buttons.
- Modifie ici le nombre d'essais pour trouver la combinaisons**: A text input field with "Le nombre d'essais actuel est de 15" below it. To the right is a "Valider" button.

At the bottom of the form, there are two large buttons: "Jouer" and "Par Défaut". The "Par Défaut" button is highlighted with a blue border.

DOCUMENTATION DES FONCTIONNALITÉS ET OPTIONS

Lorsque l'utilisateur appuie sur ce bouton, toutes les textbox vont être vidées grâce à l'instruction "clear", les boutons vont reprendre leur couleurs initiale et la fonction initialise va être appelée pour réinitialiser toutes les règles présente dans le module LesRègles.

```
Initialise()  
TextBox1.Clear()  
TextBox2.Clear()  
TextBox3.Clear()  
Button4.ForeColor = Color.Black  
Button5.ForeColor = Color.Green  
Button6.ForeColor = Color.Blue
```

Nous avons aussi décidé d'afficher le nombre d'essais restants à chaque coup dans la partie grâce à un label :

```
Label10.Text = ("Nombre d'essais restant : " &  
NbEssaisAutorisé - Compteur)
```

La variable compteur renvoi au nombre de coup joué par le joueur, donc la soustraction du nombre autorisés et du nombre de coup joué affiche le nombres de coup restant. Le temps restant est aussi affiché avec le même raisonnement :

```
Label11.Text = ("Le temps restant est de :  
"Temps_afficher.ToString)
```

Voici un exemple d'affichage :

DOCUMENTATION DES FONCTIONNALITÉS ET OPTIONS



Ce choix d'implémentation me paraît assez optimisé puisqu'en enregistrant les règles classiques dans des variables, il n'y a qu'à les modifier pour pouvoir changer les règles. Alors que si nous n'avions pas décidé de le faire, on aurait dû créer deux cas de figures: le premier cas avec les règles classiques, et le deuxième avec de nouvelles règles, on aurait donc dû dupliquer du code ce qui n'est pas une bonne idée.

CONCLUSION

Ce projet a été une expérience enrichissante mais aussi très amusante pour chacun d'entre nous.

Contrairement aux autres projets de développement de logiciel qui ont été fait à deux, celui-ci fut plus compliqué, du fait que nous sommes un groupe de 4. Le partage des tâches a été un challenge car nous n'avons pas tous le même niveau. Certains avaient plus de facilité que d'autres. Nous avons de ce fait été prévoyant et commencé le projet au plus tôt, ce qui nous a permis une bonne gestion du temps. Concernant la difficulté du partage du travail, la solution trouvée fut l'entraide et la communication.

Notre quatuor a su travailler de manière efficace et a été motivé par le sujet qui fut inspirant. Le résultat final est donc satisfaisant et témoigne de l'implication de chacun. La mise en commun des quatres travaux a donné lieu à une diversité d'idées ce qui a permis un meilleur programme et une progression plus rapide du projet.

Nous pensons que nous aurions pu améliorer certains points tel que l'esthétique ou la longueur du code. Nous aurions aussi pu mettre en place un moyen pour l'utilisateur de charger sa propre sauvegarde et d'en avoir plusieurs.