
Projet IHM : réalisation d'un taquin cylindrique



DI GIACOMO Alessio (104) | VIGNARAJAH Daran (111)

Etudiants en première année de DUT informatique

Introduction

L'objectif de ce projet IHM est de réaliser un puzzle cylindrique codé en Visual Basic sur Visual Studio. L'objectif de ce jeu est le suivant : Un joueur doit résoudre un taquin cylindrique dans un temps défini.

Mais qu'est-ce qu'un taquin ? C'est très simple : créé au 19^{ème} siècle au Etats-Unis, le taquin se présente sous forme d'un jeu solitaire en forme de damier. Le damier est composé de 15 petits carreaux numérotés de 1 à 15 et peuvent glisser dans un cadre prévu pour 16.

L'objectif de ce jeu est de remettre dans l'ordre les 15 chiffres qui étaient préalablement placés au hasard sur le damier.

Un taquin dit cylindrique permet aux carreaux glissants se situant en fin de ligne de revenir en début de ligne.

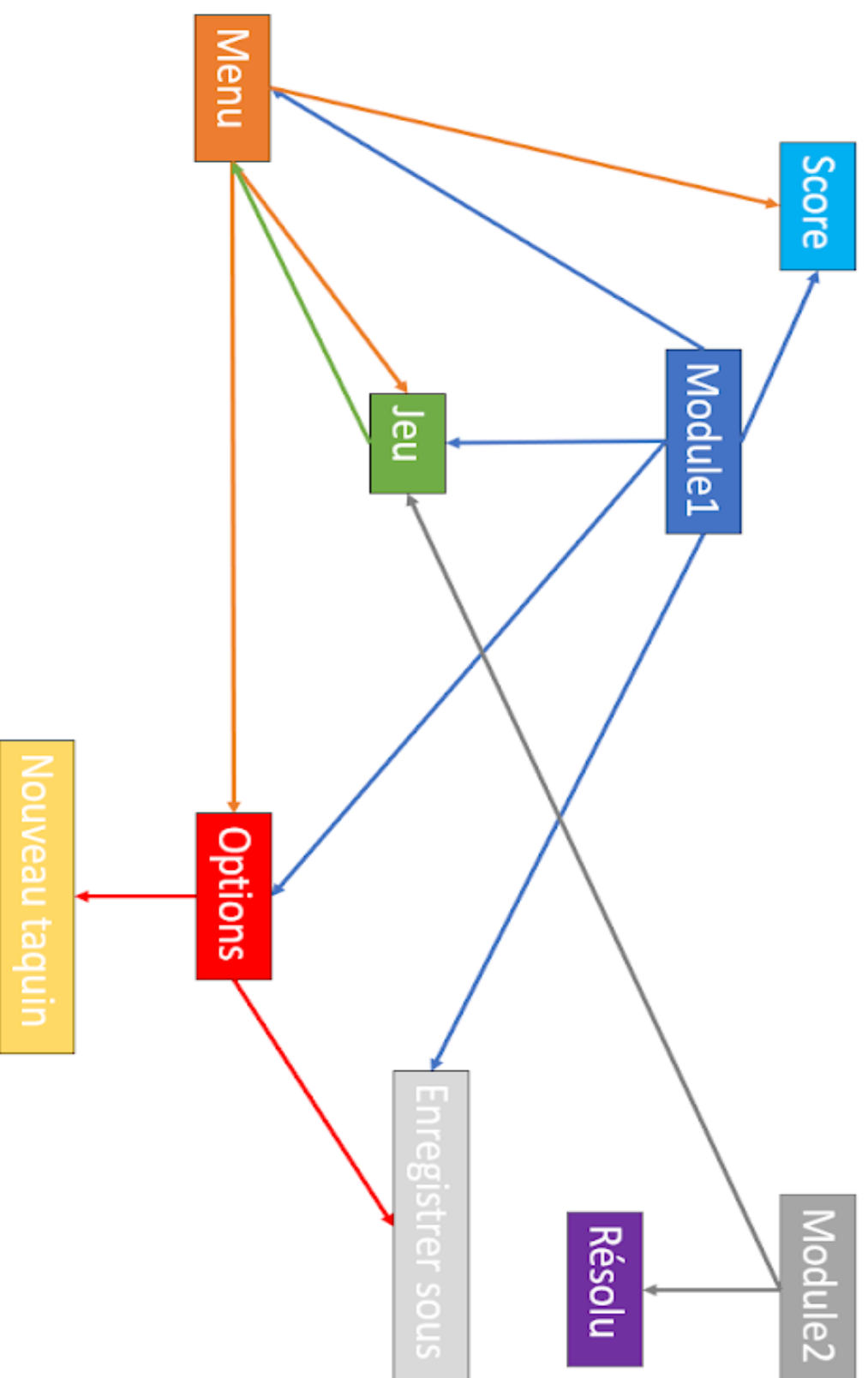
Dans notre application que nous avons codée, les petits carreaux sont représentés par des boutons.

Lors du démarrage de notre application, le joueur lui est proposé une interface où il peut s'inscrire ou bien sélectionner son nom déjà inscrit. Il peut par la suite paramétrer le jeu via un bouton « option » (redimensionnement du taquin, modification du chemin de sauvegarde, réglage du temps...). Lorsque le joueur joue, il peut revenir aux coups précédents mais également demander la solution à travers les boutons proposés. Il peut également abandonner la partie, puis consulter ses scores (en inscrivant son nom dans un texte de saisie et qui va rechercher ce nom) et son classement (il peut également consulter en triant). Enfin il lui est également mis à disposition un bouton permettant de consulter ses caractéristiques complètes.

Fonctionnalités développées :

	Réalisé	A améliorer
Bouton pause	X	
Sauvegarde dans un fichier texte	X	
Généré un Taquin aléatoire mais réalisable	X	
Résoudre automatique le Taquin	X	
Implémentation d'un bouton retour aux coups précédents	X	Créer un bouton recommencé (une ligne et une boucle while aurait suffi dans notre code)
Configuration des paramètres	X	Utilisation de checkBox réduirait le nombre de ligne écrite
Enregistrement automatique des paramètres et des joueurs	X	Création d'une base SQL ou Excel
Plusieurs types de taquin proposé (plus création par l'utilisateur)	X	Permettre de changer la taille en 3 x 3 ou 5 x 5 par exemple
Utilisation d'une musique de fond contrôlable	X	Permettre de choisir la musique
Pouvoir modifier l'esthétique du taquin (en plus du type, la couleur)	X	
Choix d'une difficulté (correspond à une mise en aléatoire du taquin plus ou moins poussé)		
Compteur de coup	X	Aurait pu être une caractéristique supplémentaire du joueur

Schéma d'ordonnancement des formulaires



Explication des formulaires

Nom :
Daran
Créé par Al Alessio

Commencer >

Stop

< Quitter

Permet l'enregistrement du nom du joueur, il a le choix de prendre un nom déjà présent dans la liste (cette liste s'actualise à chaque fois que un joueur est créer ou que l'application se lance en allant les récupérer dans un fichier texte

Permet la lecture et l'arrêt de la musique de fond

Daran

Daran
Alessio

Ordre alphabétique

Meilleur temps

Caractéristiques complètes

0
12

Retour



The 'Options' window contains the following settings:

- Temps:** ☐ Oui, ☐ Non
- Timer :** 60 seconde(s)
- Chemin de sauvegarde:** C:\Users\ALESSIO\Desktop\
- Type de Taquin:** A dropdown menu with 'chiffres', 'lettres', and 'symboles'. A red star is next to the '+' button.
- Couleur du taquin:** A blue button.
- Activation d'un bouton pause:** ☐ Oui, ☐ non
- Affichage d'une barre du temps:** ☐ Oui, ☒ non
- Facile, Difficulté, Difficile:** A slider bar.
- Activation d'un bouton de précédent:** ☐ Oui, ☐ non
- Activation d'un bouton de suivant:** ☐ Oui, ☐ non
- Enregistrer:** A large blue button.

Différentes fonctionnalité de paramétrage pour le confort de l'utilisateur avec un label explicite correspondant, la possibilité notamment de changer le chemin de sauvegarde des fichiers dit de DATA (correspond au paramètre et au caractéristique des joueurs) et changer le type de taquin et sa couleur.

Chemin de sauvegarde choisit par l'utilisateur (il se modifie en fonction de la ou clique l'utilisateur).

The 'Enregistrer sous' window shows a file explorer with the following table:

Nom	Modifié le	Type	Taille
Musiques	22/05/2019 08:36	Dossier de fichiers	
Resolveur	22/05/2019 12:09	Dossier de fichiers	
Types	22/05/2019 08:36	Dossier de fichiers	
data.txt	22/05/2019 22:34	Document texte	0 Ko
types.txt	21/05/2019 09:17	Document texte	1 Ko

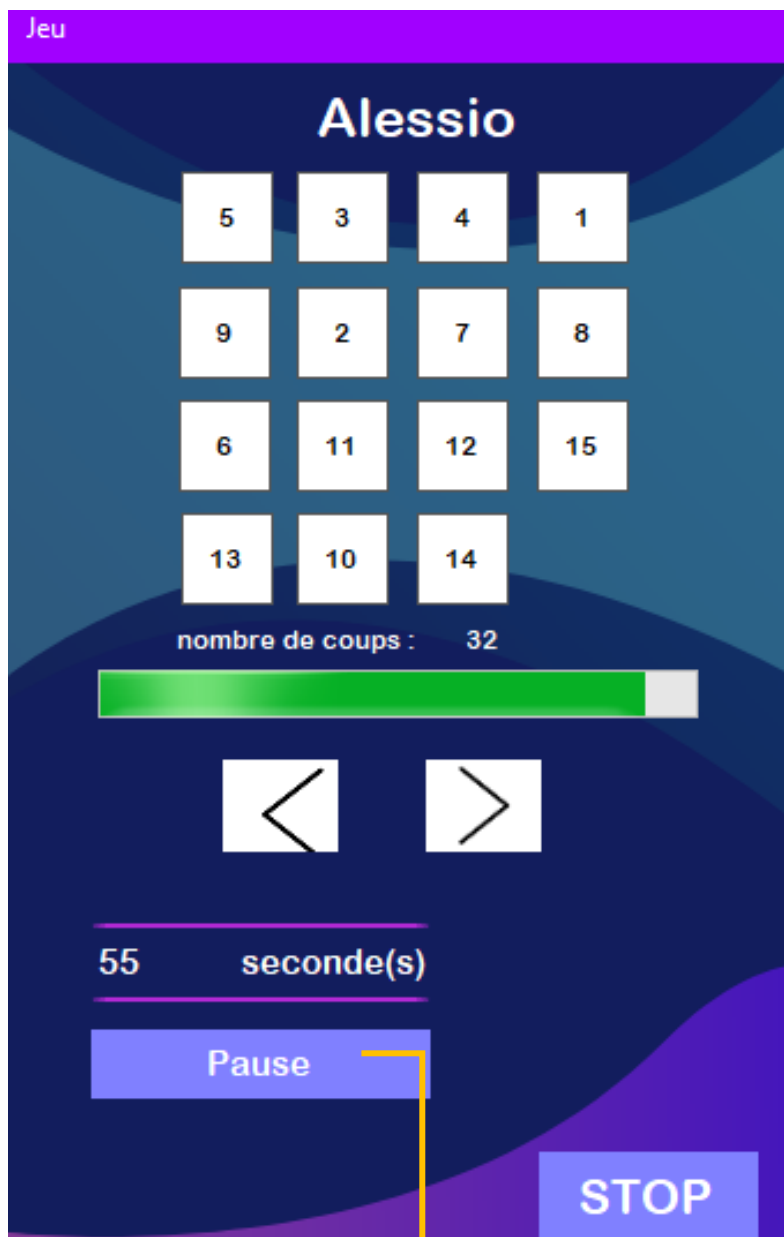
The path 'C:/Users/ALESSIO/Desktop/T' is highlighted in the address bar.

The 'Nouveau taquin' window shows a 4x4 grid of tiles. The bottom row contains a blue circle, a white square, a white square, and a white square. The bottom right corner has a button labeled 'Ajoutez' and a text input field labeled 'Donnez lui un nom'.



Possibilité de créer un nouveau type de Taquin, vous pouvez alors donner un nom personnalisé à votre création.

Lorsque l'utilisateur va appuyer sur le bouton commencé, il pourra alors voir apparaître un Taquin rendu aléatoire mais néanmoins ré solvable qu'il pourra par la suite tenté de résoudre



Un compteur de coups est là pour vous donner un ordre d'idée du nombre de déplacement réaliser.

Un compte à rebours représente ici le temps qui découle, s'ajoute à cela une jauge de progression qui donne plus de visuel au temps.

Ici le bouton pause est présent mais l'utilisateur n'a pas encore cliqué dessus.

Un bouton précédent pour retourner dans les coups précédemment réaliser et un bouton suivant qui signifie résoudre le Taquin automatiquement.

Finalement un bouton stop pour arrêter les souffrances des joueurs et les laisse abandonné

Le joueur clique sur pause, le temps s'arrête ainsi que la jauge du temps mais en contrepartie il ne peut plus jouer



Si le bouton résoudre est cliqué alors ce formulaire apparaît si le joueur le souhaite afin de pouvoir observer les mouvements à effectuer pour résoudre le taquin qu'il avait à résoudre

Form7

Damier : 4 lignes, 4 colonnes
Solution en 42 mouvements

1 2 3 4
5 7 12 8
11 6 # 13
9 10 15 14

OUEST

1 2 3 4
5 7 12 8
11 # 6 13
9 10 15 14

SUD

1 2 3 4
5 7 12 8
11 10 6 13
9 # 15 14

OUEST

1 2 3 4
5 7 12 8
11 10 6 13
9 15 14


NORD

1 2 3 4
5 7 12 8
10 6 13
11 9 15 14

OUEST

1 2 3 4
5 7 12 8
13 10 6 #
11 9 15 14

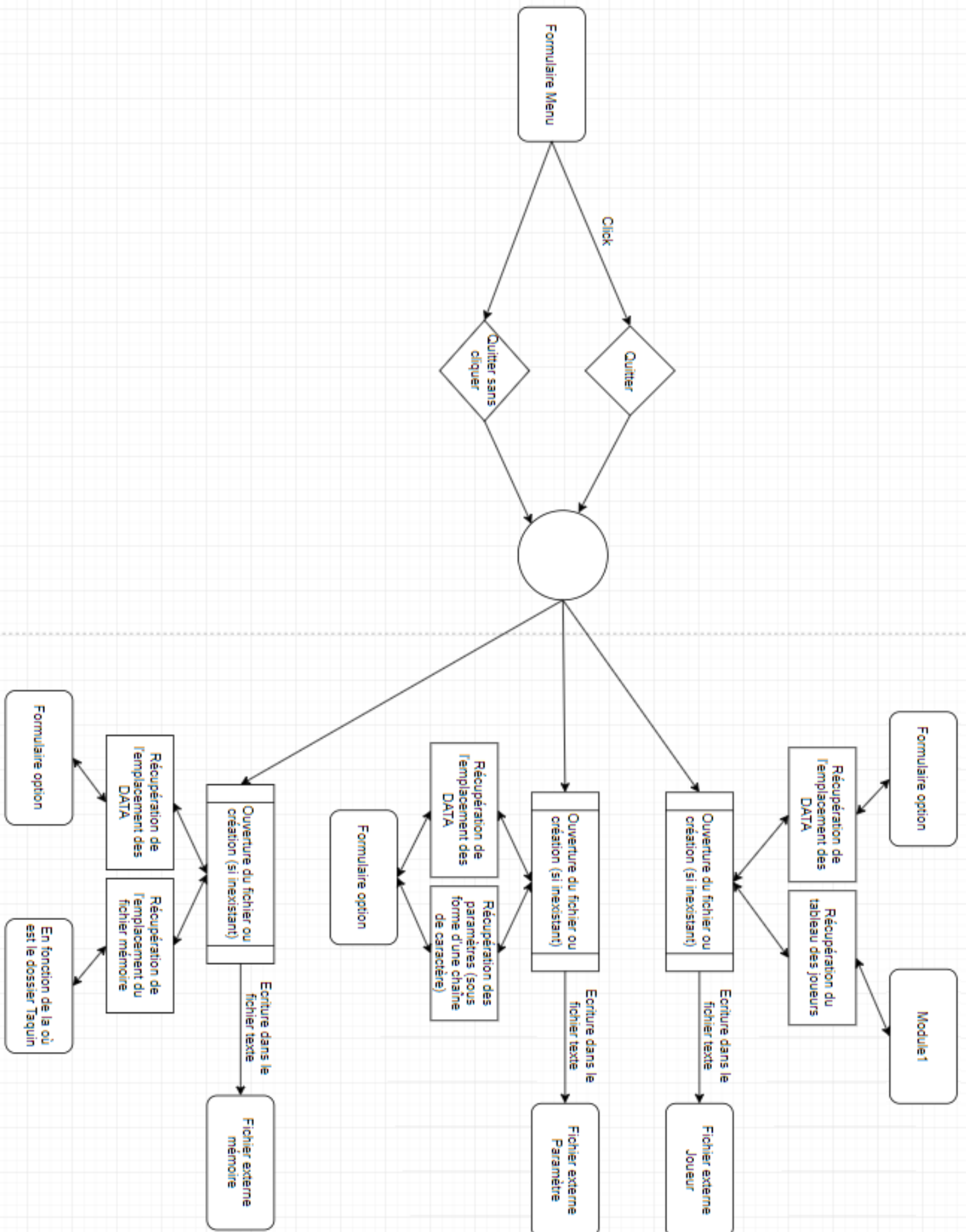
SUD

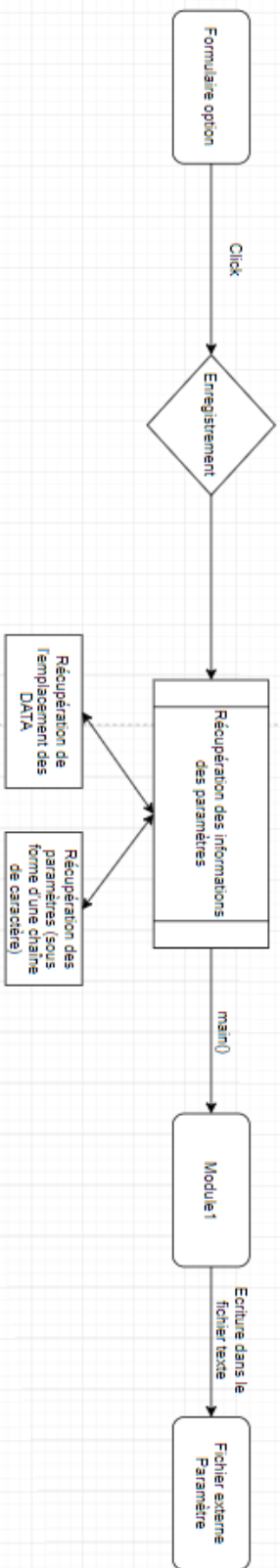
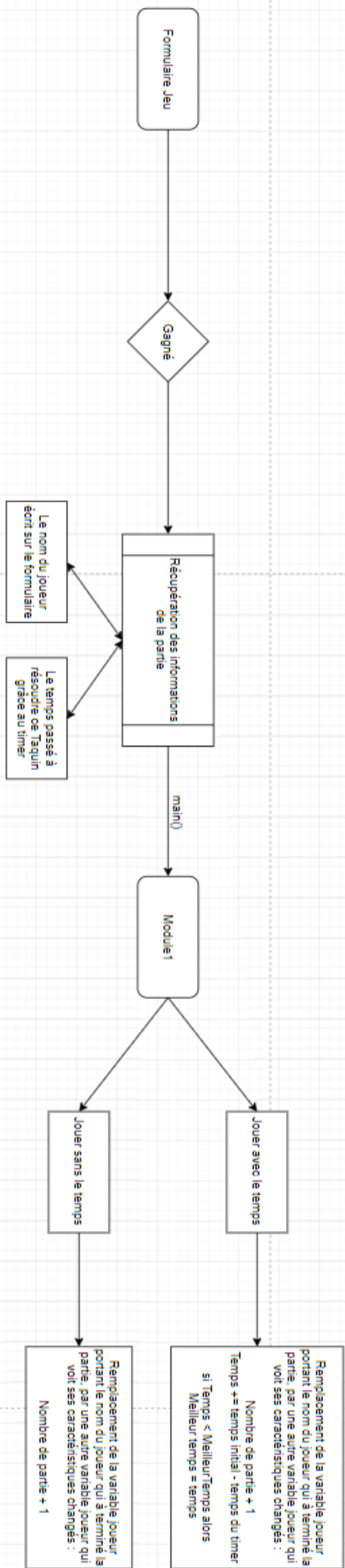


Retour

Affichage des mouvements menant
à la solution à partir du taquin initial
(Grâce au Taquin.exe)

Explication des enregistrements





Code des formulaires

Formulaire menu :

```
Public Class Form_menu

    Dim MusiqueActive As Boolean = True
    Private Sub Form_menu_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        Me.Text = "Taquin"
        label_nom.Text = "Nom :"
        label_presentation.Text = "Cr     par Alessio Di Giacomo (104) et Daran Vignarajah (111)"
        button_commencer.Text = "Commencer >"
        button_quitter.Text = "< Quitter"
        combobox_nom.Focus()
        Module1.RecupererData()
        form_option.RecupererTypes()
        Module1.RecupererParam()
        Module1.RecupererFichier()
        Module1.JouerMusique()
    End Sub

    Private Sub Combobox_nom_TextChanged(sender As Object, e As EventArgs) Handles combobox_nom.TextChanged
        If (combobox_nom.Text.Length > 0) Then
            button_commencer.Enabled = True
        Else
            button_commencer.Enabled = False
        End If
    End Sub

    Private Sub Combobox_nom_KeyPress(sender As Object, e As KeyPressEventArgs) Handles combobox_nom.KeyPress

        If (combobox_nom.Text.Length > 10) Then
            MsgBox("Taille limite atteinte")
        Else
            If (Char.IsLetter(e.KeyChar) = False And Asc(e.KeyChar) <> 32 And Asc(e.KeyChar) <> 8) Then
                e.Handled = True
            Else
                combobox_nom.DroppedDown = True
                combobox_nom.SelectedItem = combobox_nom.FindString(combobox_nom.Text)
            End If
        End If

    End Sub

    Private Sub Combobox_nom_LostFocus(sender As Object, e As EventArgs) Handles combobox_nom.LostFocus
        combobox_nom.Text = StrConv(combobox_nom.Text, VbStrConv.ProperCase)
    End Sub

    Private Sub Button_commencer_Click(sender As Object, e As EventArgs) Handles button_commencer.Click
        Module1.Main(combobox_nom.Text)
        Me.Hide()
        Form_jeu.Show()
    End Sub

End Class
```

```

Private Sub Button_quitter_Click(sender As Object, e As EventArgs) Handles
button_quitter.Click
    If (MsgBox("Etes-vous sûr ?", MsgBoxStyle.YesNo) = MsgBoxResult.Yes) Then
        If (MsgBox("Certains ?", MsgBoxStyle.YesNo) = MsgBoxResult.Yes) Then
            Me.Close()
        End If
    End If
End Sub

Private Sub Button_score_Click(sender As Object, e As EventArgs) Handles
button_score.Click
    Form_score.Show()
End Sub
Public Sub Ajout(s As String)
    combobox_nom.Items.Add(s)
End Sub

Private Sub Button_option_Click(sender As Object, e As EventArgs) Handles
button_option.Click
    form_option.Show()
End Sub

Private Sub Form_menu_FormClosing(sender As Object, e As EventArgs) Handles
MyBase.FormClosing
    Dim s As String = form_option.GetPram()
    Module1.EnregistrerFichier()
    Module1.EnregistrerParam(s)
    Module1.EnregistrerData()
End Sub

Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    MusiqueActive = Not MusiqueActive
    If (MusiqueActive) Then
        Module1.JouerMusique()
        Button1.Text = "STOP"
    Else
        Module1.StopMusique()
        Button1.Text = "PLAY"
    End If
End Sub

End Class

```

Formulaire Jeu :

```

Public Class Form_jeu

    Dim NbCoups As Integer = 0
    Dim tab As New List(Of Object)
    Dim caseVide As New Button
    Dim taquin(3, 3) As Button
    Dim solution As New List(Of String)
    Dim mémoire As New List(Of Button(,))

    Dim commencer As Boolean = False
    Dim joueurAvecTemps As Boolean = True

```

```

Private Sub Form_jeu_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    Me.Text = "Jeu"
    Me.ControlBox = False
    Label1.Text = NbCoups
    button_pause.Text = "Pause"
    If (Form_menu.combobox_nom.Text.CompareTo("") = 0) Then
        label_nom.Text = "pas_de_nom"
    Else
        label_nom.Text = Form_menu.combobox_nom.Text
    End If

    If (joueurAvecTemps = True) Then
        Timer1.Enabled = True
        Timer2.Enabled = True
        Timer1.Interval = Module1.GetTemps() * 1000
        Timer2.Interval = 1000
        label_temps.Text = Module1.GetTemps()
        ProgressBar1.Maximum = Module1.GetTemps
        ProgressBar1.Value = ProgressBar1.Maximum
    Else
        Timer1.Enabled = False
        label_temps.Visible = False
        label_seconde.Visible = False
        ProgressBar1.Visible = False
        ProgressBar1.Enabled = False
    End If

    tab = Init(tab)
    Dim liste As New List(Of Button)

    For i As Integer = Panel1.Controls.Count - 1 To 0 Step -1
        Panel1.Controls.Item(i).Text = tab(0)
        If (Panel1.Controls.Item(i).Text.CompareTo("X") = 0) Then
            liste.Add(caseVide)
        Else
            liste.Add(Panel1.Controls.Item(i))
        End If
        Panel1.Controls.Item(i).BackColor = form_option.ColorDialog1.Color
        tab.RemoveAt(0)
    Next
    For i As Integer = 0 To 3
        For j As Integer = 0 To 3
            If (liste.Item(0).Equals(caseVide)) Then
                taquin(i, j) = caseVide
            Else
                taquin(i, j) = liste.Item(0)
            End If
            liste.RemoveAt(0)
        Next
    Next

    Supprimer()

    Module2.Chambouler()

    tab = Init(tab)
    Module2.Main(tab)
    solution = Module2.GetSolution()

    mémoire.Add(taquin.Clone)

    If (form_option.BoutonPause) Then

```

```

        button_pause.Visible = True
    Else
        button_pause.Visible = False
    End If

    If (form_option.Progress) Then
        ProgressBar1.Visible = True
    Else
        ProgressBar1.Visible = False
    End If

    If (form_option.BoutonS) Then
        Panel2.Visible = True
        button_precedent.Enabled = False
    Else
        Panel2.Visible = False
    End If

    commencer = True

End Sub

Private Sub Supprimer()
    For Each b As Button In Panel1.Controls
        If (b.Text.CompareTo("X") = 0) Then
            caseVide.Location = b.Location
            Panel1.Controls.Remove(b)
        End If
    Next
End Sub

Private Function Init(tab As List(Of Object))
    Dim fichier As Integer = 0
    Dim nomFichier As String = ""

    fichier = FreeFile()
    nomFichier = form_option.default + "\Data\Types\" + form_option.typeActuel +
".txt"

    FileOpen(fichier, nomFichier, OpenMode.Input)
    While EOF(fichier) = False
        tab.Add(LineInput(fichier))
    End While
    FileClose(fichier)

    Return tab
End Function

Private Sub Timer(sender As Object, e As EventArgs) Handles Timer1.Tick
    Me.Close()
    Timer1.Enabled = False
    If (MsgBox("Perdu !", MsgBoxStyle.OkOnly) = MsgBoxResult.Ok) Then
        Form_menu.Show()
        Form_menu.combobox_nom.ResetText()
    End If
End Sub

Private Sub Affichage_Compte_A_Rebour(sender As Object, e As EventArgs) Handles
Timer2.Tick
    label_temps.Text -= 1
    ProgressBar1.Value -= 1
End Sub

```

```

Private Sub Button_abandonner_Click(sender As Object, e As EventArgs) Handles
button_abandonner.Click
    If (MsgBox("Voulez-vous vraiment abandonner ?", MsgBoxStyle.YesNo) =
MsgBoxResult.Yes) Then
        Me.Close()
        Form_menu.Show()
    End If
End Sub

Private Sub Button_Click(sender As Object, e As EventArgs) _
Handles Button2.Click, Button3.Click, Button4.Click, Button5.Click,
Button6.Click, Button7.Click, Button8.Click,
Button9.Click, Button10.Click, Button11.Click, Button12.Click, Button13.Click,
Button14.Click, Button15.Click,
Button16.Click, Button17.Click

    If (DéplacementPossible(sender)) Then
        Dim var As New Button With {.Location = sender.location}
        sender.location = caseVide.Location
        caseVide.Location = var.Location

        For i As Integer = 0 To 3
            For j As Integer = 0 To 3
                If (taquin.GetValue(i, j).Equals(sender)) Then
                    taquin.SetValue(caseVide, i, j)
                ElseIf (taquin.GetValue(i, j).Equals(caseVide)) Then
                    taquin.SetValue(sender, i, j)
                End If
            Next
        Next

        mémoire.Add(taquin.Clone)
        AfficherCoups()
        button_precedent.Enabled = True
    End If

    If (commencer = True) Then
        If (EstGagné()) Then
            Timer1.Stop()
            Timer2.Stop()
            MsgBox("Vous avez gagné !!!", MsgBoxStyle.OkOnly)
            Form_menu.Show()
            Module1.Enregistrer()
            Me.Close()
        End If
    End If
End Sub

Private Sub AfficherCoups()
    NbCoups += 1
    Label1.Text = NbCoups
End Sub

Private Function DéplacementPossible(b As Button)
    For i As Integer = 0 To 3
        For j As Integer = 0 To 3
            If (taquin.GetValue(i, j).Equals(b)) Then
                If (i = 0) Then
                    If (j = 0) Then
                        Return taquin.GetValue(i + 1, j).Equals(caseVide) Or
taquin.GetValue(i, j + 1).Equals(caseVide) Or taquin.GetValue(i, 3).Equals(caseVide)
                    ElseIf (j = 3) Then

```

```

        Return taquin.GetValue(i + 1, j).Equals(caseVide) Or
taquin.GetValue(i, j - 1).Equals(caseVide) Or taquin.GetValue(i, 0).Equals(caseVide)
    Else
        Return taquin.GetValue(i + 1, j).Equals(caseVide) Or
taquin.GetValue(i, j + 1).Equals(caseVide) Or taquin.GetValue(i, j -
1).Equals(caseVide)
    End If
    ElseIf (i = 3) Then
        If (j = 0) Then
            Return taquin.GetValue(i - 1, j).Equals(caseVide) Or
taquin.GetValue(i, j + 1).Equals(caseVide) Or taquin.GetValue(i, 3).Equals(caseVide)
        ElseIf (j = 3) Then
            Return taquin.GetValue(i - 1, j).Equals(caseVide) Or
taquin.GetValue(i, j - 1).Equals(caseVide) Or taquin.GetValue(i, 0).Equals(caseVide)
        Else
            Return taquin.GetValue(i - 1, j).Equals(caseVide) Or
taquin.GetValue(i, j + 1).Equals(caseVide) Or taquin.GetValue(i, j -
1).Equals(caseVide)
        End If
    ElseIf (j = 0) Then
        Return taquin.GetValue(i + 1, j).Equals(caseVide) Or
taquin.GetValue(i, j + 1).Equals(caseVide) Or taquin.GetValue(i - 1,
j).Equals(caseVide) Or taquin.GetValue(i, 3).Equals(caseVide)
    ElseIf (j = 3) Then
        Return taquin.GetValue(i - 1, j).Equals(caseVide) Or
taquin.GetValue(i, j - 1).Equals(caseVide) Or taquin.GetValue(i + 1,
j).Equals(caseVide) Or taquin.GetValue(i, 0).Equals(caseVide)
    Else
        Return taquin.GetValue(i + 1, j).Equals(caseVide) Or
taquin.GetValue(i - 1, j).Equals(caseVide) Or
        taquin.GetValue(i, j + 1).Equals(caseVide) Or
taquin.GetValue(i, j - 1).Equals(caseVide)
    End If
End If
Next
Next
#Disable Warning BC42105 ' La fonction ne renvoie pas de valeur sur tous les chemins
de code
End Function
#Enable Warning BC42105 ' La fonction ne renvoie pas de valeur sur tous les chemins de
code

Private Function EstGagné()
    tab.RemoveRange(0, tab.Count)
    tab = Init(tab)
    Dim b As Boolean = False
    For i As Integer = 0 To 3
        For j As Integer = 0 To 3
            If (tab(0).ToString.CompareTo("X") = 0) Then
                If (taquin(i, j).Equals(caseVide)) Then
                    b = True
                Else
                    Return False
                End If
            Else
                If (taquin(i, j).Text.CompareTo(tab(0).ToString) = 0) Then
                    b = True
                Else
                    Return False
                End If
            End If
        End If
    End If
End Function

```



```

        tab.RemoveAt(0)
    Next
Next
Return b
End Function

Private Sub Button_pause_Click(sender As Object, e As EventArgs) Handles
button_pause.Click
    Timer1.Enabled = Not Timer1.Enabled
    Timer2.Enabled = Not Timer2.Enabled
    JouerPossible(Timer1.Enabled)
    If (button_pause.Text.CompareTo("Pause") = 0) Then
        button_pause.Text = "Continuer"
    Else
        button_pause.Text = "Pause"
    End If
End Sub

Private Sub JouerPossible(b As Boolean)
    If (b = False) Then
        For Each bouton As Button In Panel1.Controls
            bouton.Enabled = b
            bouton.BackColor = Color.DarkSlateGray
        Next
    Else
        For Each bouton As Button In Panel1.Controls
            bouton.Enabled = b
            bouton.BackColor = Color.WhiteSmoke
        Next
    End If
End Sub

Public Function GetTemps()
    Return joueurAvecTemps
End Function

Public Sub SetTemps(b As Boolean)
    joueurAvecTemps = b
End Sub

Private Sub Button_precedent_Click(sender As Object, e As EventArgs) Handles
button_precedent.Click
    mémoire.RemoveAt(mémoire.Count - 1)
    If (mémoire.Count = 1) Then
        MsgBox("final")
    End If
    If (mémoire.Count = 0) Then
        button_precedent.Enabled = False
    Else
        Dim var As New Button
        For i As Integer = 0 To 3
            For j As Integer = 0 To 3
                If (mémoire(mémoire.Count - 1)(i, j).Equals(taquin(i, j)) = False)
Then
                    var.Location = taquin(i, j).Location
                    For x As Integer = 0 To 3
                        For y As Integer = 0 To 3
                            If (mémoire(mémoire.Count - 1)(i, j).Equals(caseVide))
Then
                                taquin(i, j).Location = caseVide.Location
                                caseVide.Location = var.Location
                                taquin = mémoire(mémoire.Count - 1)

```

```

Exit Sub
End If
Next
Next
End If
Next
End If
End Sub
Public Function GetTaquin()
Return taquin
End Function

Private Sub Button_suivant_Click(sender As Object, e As EventArgs) Handles
button_suivant.Click
While solution.Count > 0
Effectuer_mvt(solution.Item(0))
solution.RemoveAt(0)
End While
If (MsgBox("Voulez-vous voir les étapes de la solution", MsgBoxStyle.YesNo) =
MsgBoxResult.Yes) Then
Form7.Show()
Else
Form_menu.Show()
End If
End Sub

Private Sub Effectuer_mvt(s As String)
If (s.CompareTo("NORD") = 0) Then
For i As Integer = 0 To 3
For j As Integer = 0 To 3
If (taquin(i, j).Equals(caseVide)) Then
taquin(i - 1, j).PerformClick()
Exit Sub
End If
Next
Next
ElseIf (s.CompareTo("EST") = 0) Then
For i As Integer = 0 To 3
For j As Integer = 0 To 3
If (taquin(i, j).Equals(caseVide)) Then
If (j = 3) Then
taquin(i, 0).PerformClick()
Else
taquin(i, j + 1).PerformClick()
End If
Exit Sub
End If
Next
Next
ElseIf (s.CompareTo("OUEST") = 0) Then
For i As Integer = 0 To 3
For j As Integer = 0 To 3
If (taquin(i, j).Equals(caseVide)) Then
If (j = 0) Then
taquin(i, 3).PerformClick()
Else
taquin(i, j - 1).PerformClick()
End If
Exit Sub
End If
Next
Next

```

```

        Next
    ElseIf (s.CompareTo("SUD") = 0) Then
        For i As Integer = 0 To 3
            For j As Integer = 0 To 3
                If (taquin(i, j).Equals(caseVide)) Then
                    taquin(i + 1, j).PerformClick()
                    Exit Sub
                End If
            Next
        Next
    End If
End Sub

Public Function GetCaseVide()
    Return caseVide
End Function

```

End Class

Formulaire Scores :

Public Class Form_score

```

    Dim tableau As List(Of Joueur) = Module1.GetTableauJoueur
    Private Sub Form_score_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        Me.Text = "Score"

        button_retour.Text = "Retour"
        button_ordrealpha.Text = "Ordre alphabétique"
        button_meilleurtemps.Text = "Meilleur temps"
        button_caractéristiques.Enabled = False
        button_caractéristiques.Text = "Caractéristiques complètes"

        For i As Integer = 0 To Module1.GetSize() - 1
            listbox_nom.Items.Add(tableau.Item(i).nom)
            listbox_temps.Items.Add(tableau.Item(i).MeilleurTemps)
        Next

        listbox_temps.Enabled = False

    End Sub
    Private Sub Button_retour_Click(sender As Object, e As EventArgs) Handles
button_retour.Click
        Me.Close()
        Form_menu.Show()
    End Sub

    Private Sub Button_ordrealpha_Click(sender As Object, e As EventArgs) Handles
button_ordrealpha.Click
        Dim var As String
        Dim var2 As Integer
        For i As Integer = 0 To listbox_nom.Items.Count - 2
            For j As Integer = 0 To listbox_nom.Items.Count - 2
                If (LCase(listbox_nom.Items(j)) > LCase(listbox_nom.Items(j + 1)))
Then
                    var = listbox_nom.Items(j)
                    listbox_nom.Items(j) = listbox_nom.Items(j + 1)
                    listbox_nom.Items(j + 1) = var
                    var2 = listbox_temps.Items(j)
                    listbox_temps.Items(j) = listbox_temps.Items(j + 1)

```

```

        listbox_temps.Items(j + 1) = var2
    End If
Next
Next
End Sub

Private Sub Button_meilleurtemps_Click(sender As Object, e As EventArgs) Handles
button_meilleurtemps.Click
    Dim var As String
    Dim var2 As Integer
    For i As Integer = 0 To listbox_temps.Items.Count - 2
        For j As Integer = 0 To listbox_temps.Items.Count - 2
            If (listbox_temps.Items(j) > listbox_temps.Items(j + 1)) Then
                var = listbox_nom.Items(j)
                listbox_nom.Items(j) = listbox_nom.Items(j + 1)
                listbox_nom.Items(j + 1) = var
                var2 = listbox_temps.Items(j)
                listbox_temps.Items(j) = listbox_temps.Items(j + 1)
                listbox_temps.Items(j + 1) = var2
            End If
        Next
    Next
End Sub

Private Sub Comboboxes_nom_KeyPress(sender As Object, e As KeyPressEventArgs)
Handles combobox_nom.KeyPress
    If (Asc(e.KeyChar) = 13) Then
        combobox_nom.Text = StrConv(combobox_nom.Text, VbStrConv.ProperCase)
        For i As Integer = 0 To listbox_nom.Items.Count - 1
            If (combobox_nom.Text.CompareTo(listbox_nom.Items(i)) = 0) Then
                listbox_nom.SetSelected(i, True)
            End If
        Next
    End If
End Sub

Private Sub Listbox_nom_SelectedIndexChanged(sender As Object, e As EventArgs)
Handles listbox_nom.SelectedIndexChanged
    combobox_nom.Text = listbox_nom.Items(listbox_nom.SelectedIndex)
    button_caractéristiques.Enabled = True
End Sub

Private Sub Button_caractéristiques_Click(sender As Object, e As EventArgs)
Handles button_caractéristiques.Click
    Dim nb As Integer = 0
    Dim t As Integer = 0
    For i As Integer = 0 To tableau.Count - 1
        If
(tableau.Item(i).nom.CompareTo(listbox_nom.Items(listbox_nom.SelectedIndex)) = 0) Then
            nb = tableau.Item(i).NbParties
            t = tableau.Item(i).Temps
        End If
    Next

    MsgBox("Nom : " & listbox_nom.Items(listbox_nom.SelectedIndex) & Chr(10) _
        & "Meilleur temps : " & listbox_temps.Items(listbox_temps.SelectedIndex)
& Chr(10) _
        & "Nombre de partie(s) jouées : " & nb & Chr(10) _
        & "Temps passé à jouer : " & t)
End Sub

End Class

```

Formulaire Options :

```
Imports System.IO
Public Class form_option

    Dim bouton As Boolean = True
    Public default As String = GetChemin()
    Public Types As New List(Of String)
    Public typeActuel As String = ""

    Private Function GetChemin()
        Dim s As String = Path.GetFullPath("TaquinVB")
        s = s.Replace("WindowsApp1\WindowsApp1\WindowsApp1\bin\Debug\TaquinVB", "")
        Return s.ToString
    End Function

    Private Sub Form_option_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        Me.Text = "Options"
        txtbox_temps.Text = Module1.GetTemps()
        combobox_chemin.Items.Add(Module1.chemin)
        combobox_chemin.Text = Module1.chemin
        For Each s As String In Types
            listbox_type.Items.Add(s)
        Next
        listbox_type.SetSelected(listbox_type.FindStringExact(typeActuel), True)
    End Sub

    Private Sub Rb_non1_CheckedChanged(sender As Object, e As EventArgs) Handles
rb_non1.CheckedChanged
        If (rb_non1.Checked = True) Then
            label_timer.Enabled = False
            label_seconde.Enabled = False
            txtbox_temps.Clear()
            txtbox_temps.Enabled = False
            Form_jeu.SetTemps(False)
            rb_oui2.Checked = False
            rb_non2.Checked = False
            rb_oui2.Enabled = False
            rb_non2.Enabled = False
            RadioButton1.Checked = False
            RadioButton2.Checked = False
            RadioButton1.Enabled = False
            RadioButton2.Enabled = False
        Else
            label_timer.Enabled = True
            label_seconde.Enabled = True
            txtbox_temps.Text = Module1.GetTemps()
            txtbox_temps.Enabled = True
            Form_jeu.SetTemps(True)
            rb_oui2.Enabled = True
            rb_oui2.Checked = True
            rb_non2.Enabled = True
            RadioButton1.Enabled = True
            RadioButton1.Checked = True
            RadioButton2.Enabled = True
        End If
    End Sub
```

```

Public Function BoutonPause()
    Return rb_oui2.Checked
End Function

Public Function Progress()
    Return RadioButton2.Checked
End Function
Public Function TempsTimer()
    Return Integer.Parse(txtbox_temps.Text)
End Function

Public Function BoutonS()
    Return rb_oui3.Checked
End Function

Private Sub Button_enregistrer_Click(sender As Object, e As EventArgs) Handles
button_enregistrer.Click
    typeActuel = listbox_type.Items(listbox_type.SelectedIndex).ToString
    chemin = combobox_chemin.Text
    Dim s As String = GetPram()
    Module1.EnregistrerParam(s)
    Me.Hide()
End Sub

Private Sub Txtbox_KeyPress(sender As Object, e As KeyPressEventArgs) Handles
txtbox_temps.KeyPress
    If (Char.IsNumber(e.KeyChar) = False And Asc(e.KeyChar) <> 8) Then
        e.Handled = True
    End If
End Sub

Private Sub Txtbox_temps_(sender As Object, e As EventArgs) Handles
txtbox_temps.LostFocus
    If (txtbox_temps.Text.Length = 0) Then
        txtbox_temps.Text = Module1.GetTemps
        MsgBox("Temps par défaut appliqué")
    ElseIf (Integer.Parse(txtbox_temps.Text) <= 0) Then
        MsgBox("Un chiffre positif et non nul est demandé !")
        txtbox_temps.ResetText()
    Else
        Module1.SetTemps(Integer.Parse(txtbox_temps.Text))
    End If
End Sub

Public Function GetPram()
    Dim chaineS As String = ""
    Dim i As Integer = 0
    If (txtbox_temps.Text.Length = 0) Then
        i = 60
    Else
        i = Integer.Parse(txtbox_temps.Text)
    End If

    If (rb_oui1.Checked = True) Then
        chaineS += "AvecTemps_" & i.ToString & vbCrLf
    Else
        chaineS += "SansTemps" & vbCrLf
    End If
    If (rb_oui2.Checked = True) Then
        chaineS += "AvecBouton" & vbCrLf
    Else
        chaineS += "SansBouton" & vbCrLf
    End If

```

```

End If
If (rb_oui3.Checked = True) Then
    chaineS += "AvecBoutonS" & vbCrLf
Else
    chaineS += "SansBoutonS" & vbCrLf
End If
If (RadioButton2.Checked = True) Then
    chaineS += "AvecBarreTemps" & vbCrLf
Else
    chaineS += "SansBarreTemps" & vbCrLf
End If
Dim s As String = ColorDialog1.Color.ToString
s = s.Replace(s.First, "")
s = s.Replace(s.Last, "")
s = s.Remove(0, 6)
chaineS += s & vbCrLf
chaineS += typeActuel & vbCrLf
chaineS += "Difficult " + TrackBar1.Value.ToString
Return chaineS
End Function

Public Function EstUnType(s As String)
    For Each chaine As String In Types
        If (chaine.CompareTo(s) = 0) Then
            Return True
        End If
    Next
    Return False
End Function

Public Sub RecupererTypes()
    Dim fichier As Integer = 0
    Dim nomFichier As String = ""
    fichier = FreeFile()
    nomFichier = default + "\Data\types.txt"
    FileOpen(fichier, nomFichier, OpenMode.Input)
    While EOF(fichier) = False
        Types.Add(LineInput(fichier))
    End While
    FileClose(fichier)
End Sub

Private Sub Button_points_Click(sender As Object, e As EventArgs) Handles
button_points.Click
    Form_enregistrersous.Show()
End Sub

Private Sub Button_plus_Click(sender As Object, e As EventArgs) Handles
button_plus.Click
    form_nvtaquin.Show()
End Sub

Private Sub Listbox_type_SelectedIndexChanged(sender As Object, e As EventArgs)
Handles listbox_type.SelectedIndexChanged
    typeActuel = listbox_type.SelectedItem()
End Sub

Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    If (ColorDialog1.ShowDialog() = DialogResult.OK) Then
        For Each b As Button In Form_jeu.Panel1.Controls
            b.BackColor = ColorDialog1.Color
        Next
    End If
End Sub

```

```
End Sub

End Class
```

Formulaire explorateur de fichier :

```
Imports System.ComponentModel

Public Class Form_enregistrersous
    Private Sub Form5_Load(sender As Object, e As EventArgs) Handles Me.Load
        Me.Text = "Enregistrer sous"
        Button1.Text = "Enregistrer"
        button_precedent.Text = "<"
        button_suivant.Text = ">"
        txtbox_chemin.Text = WebBrowser1.Url.ToString
        txtbox_chemin.Enabled = False
    End Sub

    Private Sub Button_precedent_Click(sender As Object, e As EventArgs) Handles
button_precedent.Click
        WebBrowser1.GoBack()
        MAJ()
    End Sub

    Private Sub Button_suivant_Click(sender As Object, e As EventArgs) Handles
button_suivant.Click
        WebBrowser1.GoForward()
        MAJ()
    End Sub

    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        Me.Close()
        form_option.combobox_chemin.Text = Module1.chemin
    End Sub

    Private Sub MAJ()
        Dim chaine As String = WebBrowser1.Url.ToString
        txtbox_chemin.Text = chaine.Replace("file:///", "")
    End Sub

    Private Sub Form_enregistrersous_FormClosing(sender As Object, e As
CancelEventArgs) Handles Me.FormClosing
        Module1.chemin = txtbox_chemin.Text
    End Sub

    Private Sub WebBrowser1_DocumentCompleted(sender As Object, e As
WebBrowserDocumentCompletedEventArgs) Handles WebBrowser1.DocumentCompleted
        MAJ()
    End Sub
End Class
```


Formulaire création de nouveau taquin :

```
Imports System.ComponentModel
```

```
Public Class form_nvtaquin
```

```
    Dim listeTexte As New List(Of String)
```

```
    Private Sub Form6_Load(sender As Object, e As EventArgs) Handles MyBase.Load
```

```
        Me.Text = "Nouveau taquin"
```

```
        Label1.Text = "Nouvelle configuration de taquin"
```

```
        For Each t As TextBox In Panel1.Controls
```

```
            t.TextAlign = HorizontalAlignment.Center
```

```
        Next
```

```
        Panel1.Controls.Item(0).Focus()
```

```
    End Sub
```

```
    Private Sub Button_ajoutez_Click(sender As Object, e As EventArgs) Handles  
button_ajoutez.Click
```

```
        If (txtbox_nom.Text.Length = 0) Then
```

```
            MsgBox("N'oubliez pas de lui donner un nom")
```

```
        ElseIf (CaseRestante() = False) Then
```

```
            listeTexte = GetTexte(listeTexte)
```

```
            CréerNouveauxType()
```

```
            MsgBox("Nouveau type créé vous pouvez à présent essayer de le résoudre")
```

```
            form_option.listbox_type.Items.Add(txtbox_nom.Text)
```

```
            Me.Close()
```

```
        End If
```

```
    End Sub
```

```
    Private Sub Txtbox_nom_KeyPress(sender As Object, e As KeyPressEventArgs) Handles  
txtbox_nom.KeyPress
```

```
        If (Asc(e.KeyChar) = 34 Or Asc(e.KeyChar) = 47 Or Asc(e.KeyChar) = 92) Then
```

```
            MsgBox("N'utilisez pas ce symbole s'il vous plait")
```

```
            txtbox_nom.ResetText()
```

```
        End If
```

```
    End Sub
```

```
    Private Function CaseRestante()
```

```
        Dim b As Boolean = True
```

```
        For Each t As TextBox In Panel1.Controls
```

```
            If (t.Text.Length = 0) Then
```

```
                b = True
```

```
            Else
```

```
                b = False
```

```
            End If
```

```
        Next
```

```
        Return b
```

```
    End Function
```

```
    Private Function GetTexte(liste As List(Of String))
```

```
        For Each t As TextBox In Panel1.Controls
```

```
            liste.Add(t.Text)
```

```
        Next
```

```
        liste.Reverse()
```

```
        Return liste
```

```
    End Function
```

```
    Private Sub CréerNouveauxType()
```

```
        Dim fichier As Integer = 0
```

```
        Dim nomFichier As String = ""
```

```
        fichier = FreeFile()
```

```

        '
        nomFichier = form_option.default + "Types\" + txtbox_nom.Text + ".txt"
        FileOpen(fichier, nomFichier, OpenMode.Output)
        For Each s As String In listeTexte
            Print(fichier, s + vbCrLf)
        Next
        FileClose(fichier)
    End Sub

    Private Sub Form_nvtaquin_FormClosing(sender As Object, e As CancelEventArgs)
Handles Me.FormClosing
        Dim fichier As Integer = 0
        Dim nomFichier As String = ""
        fichier = FreeFile()
        '
        nomFichier = form_option.default + "types.txt"
        form_option.Types.Add(txtbox_nom.Text)
        FileOpen(fichier, nomFichier, OpenMode.Output)
        For Each s As String In form_option.Types
            Print(fichier, s + vbCrLf)
        Next
        FileClose(fichier)
    End Sub

    Private Sub TextBox1_LostFocus(sender As Object, e As EventArgs) _
Handles TextBox1.LostFocus, TextBox2.LostFocus, TextBox3.LostFocus,
TextBox4.LostFocus, TextBox5.LostFocus,
        TextBox6.LostFocus, TextBox7.LostFocus, TextBox8.LostFocus,
        TextBox9.LostFocus, TextBox10.LostFocus, TextBox11.LostFocus, TextBox12.LostFocus,
        TextBox13.LostFocus,
        TextBox14.LostFocus, TextBox15.LostFocus, TextBox16.LostFocus

        For Each t As TextBox In Panel1.Controls
            If (sender.Equals(t) = False And sender.text.CompareTo("") <> 0) Then
                If (sender.text.CompareTo(t.Text) = 0) Then
                    sender.resetText()
                    MsgBox("Cette case existe déjà")
                    Exit Sub
                End If
            End If
        End For
    Next

End Sub

End Class

```

Formulaire résolution de taquin :

```

Public Class Form7
    Private Sub Form7_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        Module2.RecupererAffichage()
        Me.ControlBox = False
        TextBox1.ScrollBars = ScrollBars.Both
        WebBrowser1.Navigate("https://media.giphy.com/media/7Sj86u4qZH6Uw/giphy.gif")
    End Sub

    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        Form_menu.Show()
        Me.Close()
    End Sub
End Class

```

Code des modules

Module 1 (enregistrement des joueurs et enregistrements des fichiers) :

```
Imports System.IO
Module Module1
    Public Structure Joueur
        Public nom As String
        Public MeilleurTemps As Integer
        Public NbParties As Integer
        Public Temps As Integer
    End Structure

    Dim temps As Integer = 60
    Dim tableau As New List(Of Joueur)
    Public chemin As String = ""
    Public Sub Main(s As String)
        If (Existedeja(s) = False) Then
            Initialisation(s)
        End If
    End Sub

    Private Sub Initialisation(s As String)
        Dim j As New Joueur With {.nom = s, .MeilleurTemps = vbNull, .NbParties = 0,
        .Temps = 0}
        tableau.Add(j)
        Form_menu.Ajout(j.nom)
    End Sub

    Private Sub InitialisationExistant(s As String)
        Dim j As New Joueur

        Dim list As List(Of Char) = s.ToList()

        For i As Integer = 0 To list.Count - 1
            Dim chaine As String = list.Item(i)
            If (chaine.CompareTo("/") <> 0) Then
                j.nom += chaine
            Else
                list.RemoveAt(i)
                Exit For
            End If
            list.RemoveAt(i)
        Next

        Dim m As String = ""
        For i As Integer = 0 To list.Count - 1
            Dim chaine As String = list.Item(i)
            If (chaine.CompareTo("/") <> 0) Then
                m += chaine
            Else
                list.RemoveAt(i)
                Exit For
            End If
            list.RemoveAt(i)
        Next

        j.MeilleurTemps = Integer.Parse(m)
        m = ""

        For i As Integer = 0 To list.Count - 1
```

```

        Dim chaine As String = list.Item(0)
        If (chaine.CompareTo("/") <> 0) Then
            m += chaine
        Else
            list.RemoveAt(0)
            Exit For
        End If
        list.RemoveAt(0)
    Next

    j.Temps = Integer.Parse(m)
    m = ""

    For i As Integer = 0 To list.Count - 1
        Dim chaine As String = list.Item(0)
        If (chaine.CompareTo("/") <> 0) Then
            m += chaine
        Else
            list.RemoveAt(0)
            Exit For
        End If
        list.RemoveAt(0)
    Next
    j.NbParties = m
    tableau.Add(j)
    Form_menu.Ajout(j.nom)
End Sub

Public Sub Enregistrer()
    If (Form_jeu.GetTemps()) Then
        Dim t As Integer = 60 - Form_jeu.label_temps.Text
        Dim j1 As New Joueur
        For i As Integer = 0 To tableau.Count - 1
            If (tableau.Item(i).nom.CompareTo(Form_jeu.label_nom.Text) = 0) Then
                j1 = tableau.Item(i)
                tableau.RemoveAt(i)
                j1.NbParties += 1
                j1.Temps += t
                If (j1.MeilleurTemps < t) Then
                    j1.MeilleurTemps = t
                End If
                tableau.Add(j1)
            End If
        Next
    Else
        Dim j1 As New Joueur
        For i As Integer = 0 To tableau.Count - 1
            If (tableau.Item(i).nom.CompareTo(Form_jeu.label_nom.Text) = 0) Then
                j1 = tableau.Item(i)
                tableau.RemoveAt(i)
                j1.NbParties += 1
                tableau.Add(j1)
            End If
        Next
    End If
End Sub

Private Function Existedeja(s As String)
    For Each j As Joueur In tableau
        If (j.nom.CompareTo(s) = 0) Then
            Return True
        End If
    End For
End Function

```

```

        Next
        Return False
    End Function
    Public Function GetTableauJoueur()
        Return tableau
    End Function
    Public Function GetSize()
        Return tableau.Count
    End Function

    Public Function Première_partie()
        Return tableau.Count = 0
    End Function

    Public Sub SetTemps(i As Integer)
        temps = i
    End Sub
    Public Function GetTemps()
        Return temps
    End Function

    Public Sub EnregistrerFichier()
        Dim fichier As Integer = 0
        Dim s As String = ""
        Dim nomFichier As String = ""
        For i As Integer = 0 To tableau.Count - 1
            s += tableau.Item(i).nom & "/" & tableau.Item(i).MeilleurTemps & "/" _
                & tableau.Item(i).Temps & "/" & tableau.Item(i).NbParties & vbCrLf
        Next
        fichier = FreeFile()
        nomFichier = chemin + "/Joueurs.txt"
        FileOpen(fichier, nomFichier, OpenMode.Output)
        Print(fichier, s)
        FileClose(fichier)
    End Sub

    Public Sub RecupererFichier()
        Dim fichier As Integer = 0
        Dim s As String = ""
        Dim nomFichier As String = ""
        fichier = FreeFile()
        nomFichier = chemin + "/Joueurs.txt"
        FileOpen(fichier, nomFichier, OpenMode.Input)
        While EOF(fichier) = False
            s = LineInput(fichier)
            InitialisationExistant(s)
        End While
        FileClose(fichier)
    End Sub

    Public Sub EnregistrerParam(chaine As String)
        Dim fichier As Integer = 0
        Dim s As String = ""
        Dim nomFichier As String = ""
        fichier = FreeFile()
        nomFichier = chemin + "/Paramètres.txt"
        s = chaine
        FileOpen(fichier, nomFichier, OpenMode.Output)
        Print(fichier, s)
        FileClose(fichier)
    End Sub

```

```

Public Sub RecupererParam()
    Dim fichier As Integer = 0
    Dim s As String = ""
    Dim Chaines As String() = {}
    Dim nomFichier As String = ""
    fichier = FreeFile()
    nomFichier = chemin + "/Paramètres.txt"
    FileOpen(fichier, nomFichier, OpenMode.Input)
    While EOF(fichier) = False
        s = LineInput(fichier)
        If (s.Length <> 0) Then
            If (s.CompareTo("SansTemps") = 0) Then
                form_option.rb_non1.Checked = True
            ElseIf (s.CompareTo("SansBouton") = 0) Then
                form_option.rb_non2.Checked = True
            ElseIf (s.CompareTo("AvecBouton") = 0) Then
                form_option.rb_oui2.Checked = True
            ElseIf (s.CompareTo("SansBoutonS") = 0) Then
                form_option.rb_non3.Checked = True
            ElseIf (s.CompareTo("AvecBoutonS") = 0) Then
                form_option.rb_oui3.Checked = True
            ElseIf (s.CompareTo("AvecBarreTemps") = 0) Then
                form_option.RadioButton2.Checked = True
            ElseIf (s.CompareTo("SansBarreTemps") = 0) Then
                form_option.RadioButton1.Checked = True
            ElseIf (s.Contains("AvecTemps")) Then
                form_option.rb_oui1.Checked = True
                Chaines = s.Split("_")
                If (Chaines.Length = 1) Then
                    temps = 60
                Else
                    temps = Integer.Parse(Chaines(1))
                End If
            ElseIf (s.Contains("Difficulté")) Then
                s = s.Replace("Difficulté_", "")
                Module2.SetDifficulté(Integer.Parse(s))
                form_option.TrackBar1.Minimum = 20
                form_option.TrackBar1.Maximum = 300
                form_option.TrackBar1.Value = Integer.Parse(s)
            ElseIf (form_option.EstUnType(s)) Then
                form_option.typeActuel = s
            Else
                If (s.Contains(",")) Then
                    form_option.ColorDialog1.Color = Color.White
                Else
                    form_option.ColorDialog1.Color = Color.FromName(s)
                End If
            End If
        End If
        form_option.rb_oui1.Checked = True
        form_option.rb_non2.Checked = True
    End While
    FileClose(fichier)
End Sub

Public Sub EnregistrerData()
    Dim fichier As Integer = 0
    Dim nomFichier As String = ""
    Dim s As String = ""
    fichier = FreeFile()
    nomFichier = form_option.defaut + "Data\" + "data.txt"
    s = form_option.combobox_chemin.Text

```

```

        FileOpen(fichier, nomFichier, OpenMode.Output)
        Print(fichier, s)
        FileClose(fichier)
    End Sub

    Public Sub RecupererData()
        Dim fichier As Integer = 0
        Dim nomFichier As String = ""
        Dim s As String = ""
        fichier = FreeFile()
        nomFichier = form_option.default + "\Data\data.txt"

        While (File.Exists(nomFichier) = False)
            nomFichier = InputBox("Le chemin précédent à été modifié, veuillez entrer le nouvel emplacement des ressources : ")
        End While

        FileOpen(fichier, nomFichier, OpenMode.Input)
        While EOF(fichier) = False
            s = LineInput(fichier)
        End While
        FileClose(fichier)

        While (Directory.Exists(s) = False)
            s = InputBox("Le chemin précédent à été modifié, veuillez entrer le nouvel emplacement des données : ")
        End While
        chemin = s
    End Sub
    Public Sub JouerMusique()
        My.Computer.Audio.Play(form_option.default + "\Data\Musiques\Lofi.wav")
    End Sub
    Public Sub StopMusique()
        My.Computer.Audio.Stop()
    End Sub

End Module

```

Module 2 (résolveur de taquin) :

```

Imports System.IO
Module Module2

    Private ReadOnly mouvement() As String = {"NORD", "EST", "OUEST", "SUD"}
    Dim solution As New List(Of String)
    Dim chaineText As String = ""
    Dim difficulté As Integer = 0
    Public Sub Main(l As List(Of Object))
        CréerBat()
        CréerIn(l)
        LanceSousProgramme()
    End Sub

    Private Sub LanceSousProgramme()

```

```

        Dim nomFichier As String = Path.GetFullPath("Desktop")
        nomFichier =
nomFichier.Replace("TaquinVB\WindowsApp1\WindowsApp1\WindowsApp1\bin\Debug\Desktop",
"")
        File.Create(nomFichier + "solution.txt").Dispose()

        Dim s As String = nomFichier + "bat.bat"
        Shell(s, AppWinStyle.MinimizedFocus, True)

        Dim fichier As Integer = FreeFile()
        Dim chaine As String = ""

        FileOpen(fichier, nomFichier + "solution.txt", OpenMode.Input)
        While EOF(fichier) = False
            chaine = LineInput(fichier)
            chaineText += chaine + vbCrLf
            If (MouvementExiste(chaine)) Then
                solution.Add(chaine)
            End If
        End While
        FileClose(fichier)

        My.Computer.FileSystem.DeleteFile(nomFichier + "solution.txt")
        My.Computer.FileSystem.DeleteFile(nomFichier + "in.txt")
        My.Computer.FileSystem.DeleteFile(nomFichier + "bat.bat")

    End Sub

    Private Function MouvementExiste(s As String)
        For Each chaine As String In mouvement
            If (s.CompareTo(chaine) = 0) Then
                Return True
            End If
        Next
        Return False
    End Function

    Private Sub CréerIn(l As List(Of Object))
        Dim fichier As Integer = FreeFile()
        Dim nomFichier As String = Path.GetFullPath("Desktop")
        nomFichier =
nomFichier.Replace("TaquinVB\WindowsApp1\WindowsApp1\WindowsApp1\bin\Debug\Desktop",
"")
        FileOpen(fichier, nomFichier + "in.txt", OpenMode.Output)
        Print(fichier, "4 4" & vbCrLf)

        For i As Integer = 0 To 3
            For j As Integer = 0 To 3
                For x As Integer = 0 To l.Count - 1
                    If (Form_jeu.GetTaquin()(i, j).text.CompareTo(l(x).ToString) = 0)
Then
                        Print(fichier, x + 1)
                        Exit For
                    ElseIf (Form_jeu.GetTaquin()(i, j).Equals(Form_jeu.GetCaseVide))
Then
                        Print(fichier, "#")
                        Exit For
                    End If
                Next
            Next
        Next
        Print(fichier, vbCrLf)
    End Sub

```



```

        Next
        FileClose(fichier)

    End Sub

    Private Sub CréerBat()
        Dim fichier As Integer = FreeFile()
        Dim nomFichier As String = Path.GetFullPath("Desktop")
        nomFichier =
nomFichier.Replace("TaquinVB\WindowsApp1\WindowsApp1\WindowsApp1\bin\Debug\Desktop",
"")
        FileOpen(fichier, nomFichier + ".bat", OpenMode.Output)
        Print(fichier, form_option.default + "Data\Resolveur\Taquin.exe" + " < " +
            nomFichier + ".in.txt" + " > " +
            nomFichier + ".solution.txt")
        FileClose(fichier)
    End Sub

    Public Function GetSolution()
        Return solution
    End Function

    Public Sub SetDifficulté(dif As Integer)
        difficulté = dif
    End Sub

    Public Sub Chambouler()
        Dim r As New Random
        For i As Integer = 0 To difficulté
            Dim aléa As Integer = r.Next(3)
            For x As Integer = 0 To 3
                For y As Integer = 0 To 3
                    If (Form_jeu.GetTaquin(x, y).Equals(Form_jeu.GetCaseVide)) Then
                        Possible(x, y, aléa)
                        Exit For
                    End If
                Next
            Next
        Next
    End Sub

    Private Sub Possible(i As Integer, j As Integer, aléa As Integer)
        If (i = 0) Then
            If (j = 0) Then
                If aléa = 0 Then
                    Form_jeu.GetTaquin.GetValue(i + 1, j).performClick()
                ElseIf aléa = 1 Then
                    Form_jeu.GetTaquin.GetValue(i, j + 1).performClick()
                Else
                    Form_jeu.GetTaquin.GetValue(i, 3).performClick()
                End If
            ElseIf (j = 3) Then
                If (aléa = 0) Then
                    Form_jeu.GetTaquin.GetValue(i + 1, j).performClick()
                ElseIf (aléa = 1) Then
                    Form_jeu.GetTaquin.GetValue(i, j - 1).performClick()
                Else
                    Form_jeu.GetTaquin.GetValue(i, 0).performClick()
                End If
            Else
                If (aléa = 0) Then
                    Form_jeu.GetTaquin.GetValue(i + 1, j).performClick()
                ElseIf (aléa = 1) Then

```

```

        Form_jeu.GetTaquin.GetValue(i, j + 1).performClick()
    Else
        Form_jeu.GetTaquin.GetValue(i, j - 1).performClick()
    End If
End If
ElseIf (i = 3) Then
    If (j = 0) Then
        If (aléa = 0) Then
            Form_jeu.GetTaquin.GetValue(i - 1, j).performClick()
        ElseIf (aléa = 1) Then
            Form_jeu.GetTaquin.GetValue(i, j + 1).performClick()
        Else
            Form_jeu.GetTaquin.GetValue(i, 3).performClick()
        End If
    ElseIf (j = 3) Then
        If (aléa = 0) Then
            Form_jeu.GetTaquin.GetValue(i - 1, j).performClick()
        ElseIf (aléa = 1) Then
            Form_jeu.GetTaquin.GetValue(i, j - 1).performClick()
        Else
            Form_jeu.GetTaquin.GetValue(i, 0).performClick()
        End If
    Else
        If (aléa = 0) Then
            Form_jeu.GetTaquin.GetValue(i - 1, j).performClick()
        ElseIf (aléa = 1) Then
            Form_jeu.GetTaquin.GetValue(i, j + 1).performClick()
        Else
            Form_jeu.GetTaquin.GetValue(i, j - 1).performClick()
        End If
    End If
ElseIf (j = 0) Then
    If (aléa = 0) Then
        Form_jeu.GetTaquin.GetValue(i + 1, j).performClick()
    ElseIf (aléa = 1) Then
        Form_jeu.GetTaquin.GetValue(i, j + 1).performClick()
    ElseIf (aléa = 2) Then
        Form_jeu.GetTaquin.GetValue(i - 1, j).performClick()
    Else
        Form_jeu.GetTaquin.GetValue(i, 3).performClick()
    End If
ElseIf (j = 3) Then
    If (aléa = 0) Then
        Form_jeu.GetTaquin.GetValue(i - 1, j).performClick()
    ElseIf (aléa = 1) Then
        Form_jeu.GetTaquin.GetValue(i, j - 1).performClick()
    ElseIf (aléa = 2) Then
        Form_jeu.GetTaquin.GetValue(i + 1, j).performClick()
    Else
        Form_jeu.GetTaquin.GetValue(i, 0).performClick()
    End If
Else
    If (aléa = 0) Then
        Form_jeu.GetTaquin.GetValue(i + 1, j).performClick()
    ElseIf (aléa = 1) Then
        Form_jeu.GetTaquin.GetValue(i - 1, j).performClick()
    ElseIf (aléa = 2) Then
        Form_jeu.GetTaquin.GetValue(i, j + 1).performClick()
    Else
        Form_jeu.GetTaquin.GetValue(i, j - 1).performClick()
    End If
End If

```

```
End Sub
Public Sub RecupererAffichage()
    Form7.TextBox1.Text = chaineText
End Sub

End Module
```