

# **Projet de programmation structurée**

## **Jeu du Kakuha**

**Yanrui Guo**

**Yiyang Wang**

**Groupe 2**

**MIAGE-IM**

**2017/2018**

## Sommaire

1 / Introduction.....	- 2 -
2 / Variables utilisées.....	- 3 -
3 / L'interface et le dialogue .....	- 3 -
4 / Le code de l'interface .....	- 10 -
5 / Le noyau fonctionnel.....	- 17 -
6 / Conclusion .....	- 22 -

# 1 / Introduction

Ce projet consiste à la réalisation d'une application Visual Basic permettant à mettre en œuvre le jeu du Kakuha. Ce jeu, d'origine africaine, se joue à deux personnes. Le matériel du jeu est composé de deux rangées de 6 trous dans lesquels 4 graines sont placées au départ. Chaque joueur se positionne de part et d'autre du jeu. Les joueurs jouent à tour de rôle. Jouer un coup consiste, pour un joueur, à choisir un trou de sa rangée, prendre en main les graines du trou en question et les semer une à une dans les trous suivants (dans le sens inverse des aiguilles d'une montre) jusqu'à l'avant dernière. Ensuite : - il met de côté la dernière graine, - il prend en main les graines du trou suivant, - et les sème une à une jusqu'à l'avant dernière. Le joueur répète ses semis jusqu'à ce qu'un trou où il aurait prendre des graines soit vide. Il passe alors la main à l'autre joueur. Le jeu cesse lorsqu'il devient évident qu'aucune prise n'est plus possible. Le vainqueur est celui qui possède le plus grand nombre de graines.

On fait les hypothèses sur la condition d'arrêt du jeu et le fonctionnement du jeu.

-->la condition d'arrêt du jeu (JeuTermine):

On suppose que le joueur dont il a plus la moitié des graines gagne, autrement dire, le vainqueur occupe au moins 24 graines (le nombre total des graines est 48).

-->le fonctionnement du jeu:

C'est-à-dire le joueur joue étape par étape jusqu'à un moment dont il faut changer le joueur. Ou on affiche directement le résultat final selon le coup pour un joueur, par exemple, un joueur du bas qui choisit son trou 4.

Ce qui l'on utilise dans ce projet c'est le jeu étape par étape. Dans la dernière partie de notre rapport, on met une idée pour réaliser l'autre possibilité.

On concept le début du jeu :

on ajoute un peu de liberté du jeu en permettant aux joueurs d'entrer leur propre nom. Bien sûr, ils peuvent aussi jouer sans changer le nom du joueur dans le jeu.

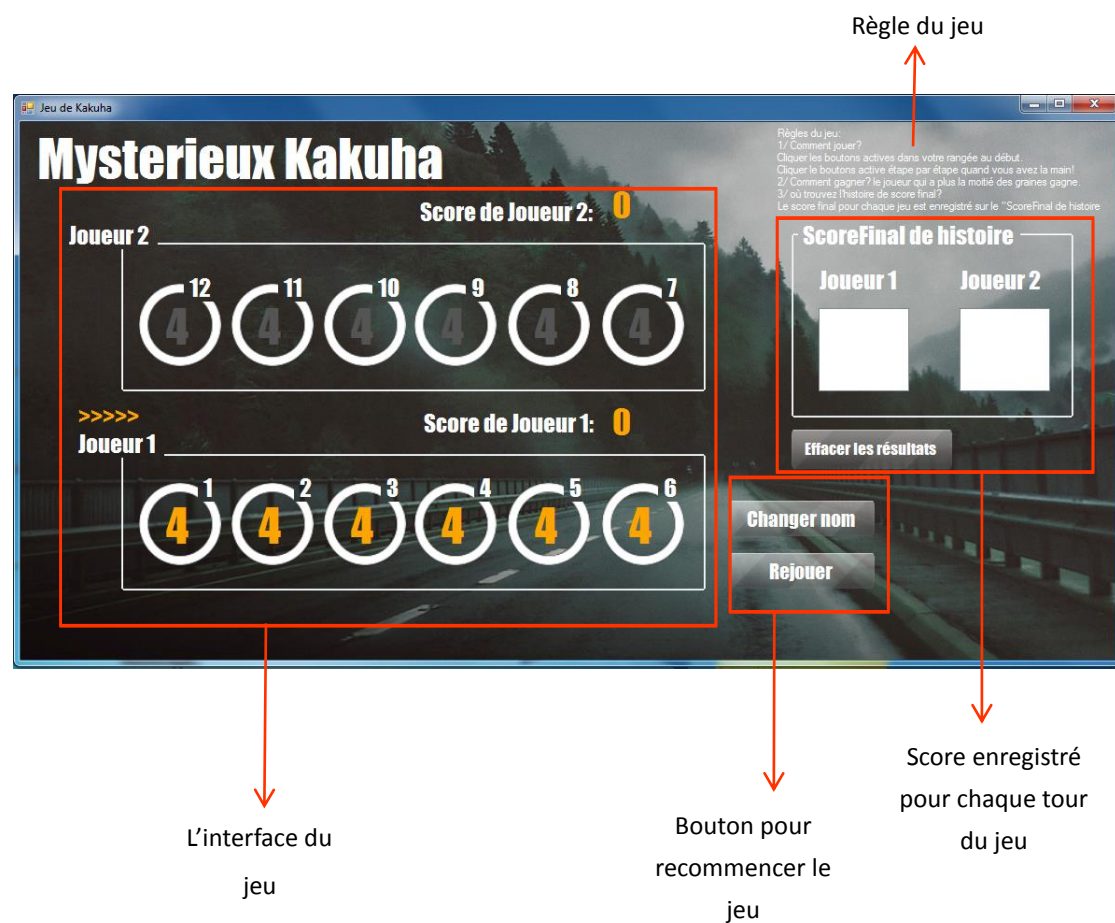
## 2 / Variables utilisées

Les type de données utilisant pour ce projet :

Type	Plage	Utilisation
Table( )	Les variables tableaux	Le tableau à 1 dimension est utilisé pour définir les trous.
Integer	-2 147 483 648 à 2 147 483 64	Tous les chiffres ces qui l'on a utilisé dans ce projet sont entier. le numéro de joueur, le nombre de graine...
Boolean	True ou False (Vrai ou Faux)	le boolean est important pour ce projet car il décider chaque étape du jeu afin d'assurer le fonctionnement du jeu.

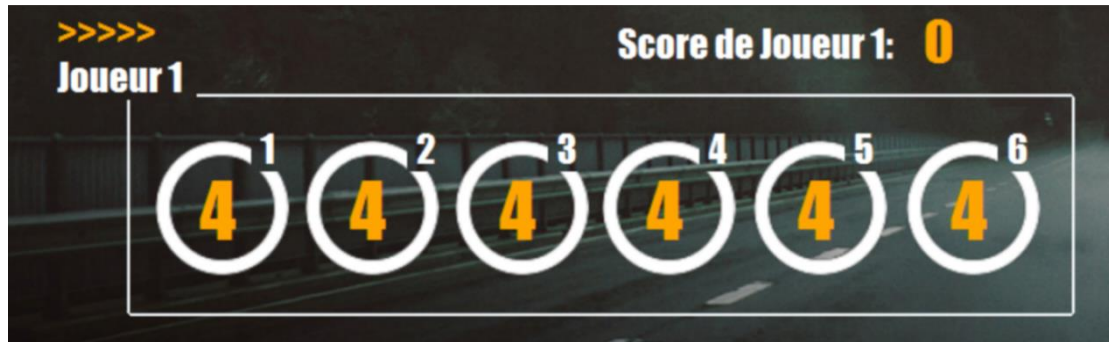
## 3 / L'interface et le dialogue

(1) L'interface (sous la situation initiale) :



La grande partie sur la gauche est l'interface du jeu. Les deux zones sont les rangées, chaque ellipse sous la forme d'un Button est un trou dans lesquels les chiffres affichent le nombre des graines inclus. Et le petit chiffre en haut en droit d'un trou est le numéro du trou.

la rangée du bas pour le joueur 1 comme un exemple:

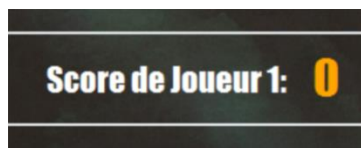


Une rangée est un GroupBox. "Joueur 1" est le texte dans ce GroupBox.



Les flèches au-dessus de "Joueur 1" montrent que c'est le joueur 1 qui a la main.

Changer le joueur, la position des flèches vont changer aussi. (Au-dessus de "Joueur 2 ")



Le score est le nombre de graine gardé par le joueur. On accumuler le score un par un chaque fois le joueur garde une graine.



Le chiffre en couleur orange signifie que ce bouton est activé. A l'inverse, le couleur gris indique l'inactivité. Initialement, quand le joueur 1 a la main, les boutons en bas sont activés et celles en haut sont inactivés.

### Bouton pour recommencer le jeu :

Les deux boutons à côté de l'interface du jeu assurent l'initialisation du jeu.



Le bouton en haut va recommencer dès le début où les deux joueurs tapent leurs noms ensuite initialiser le jeu. C'est-à-dire que deux nouveaux joueurs vont jouer Kakuha.

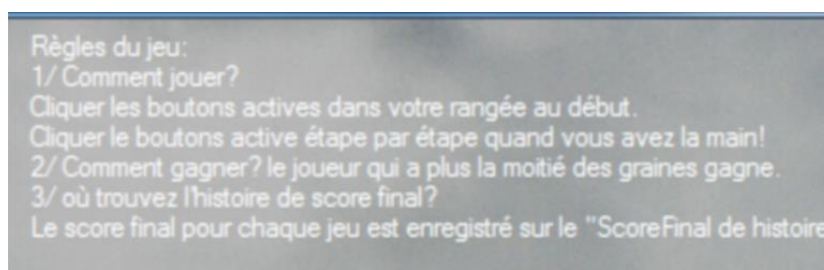
L'autre bouton va initialiser le jeu sauf changer les joueurs participés. Autrement dit, les joueurs toujours restent là.

### Score enregistré pour chaque tour du jeu :



Chaque fois que le jeu est terminé, on enregistre le score final du joueur dans le listbox dans lequel le score du vainqueur a une petite étoile à côté. Le bouton permet d'effacer tout dans les deux listbox.

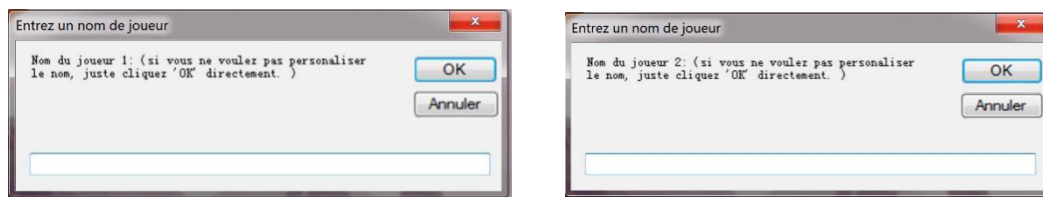
### Règle du jeu :



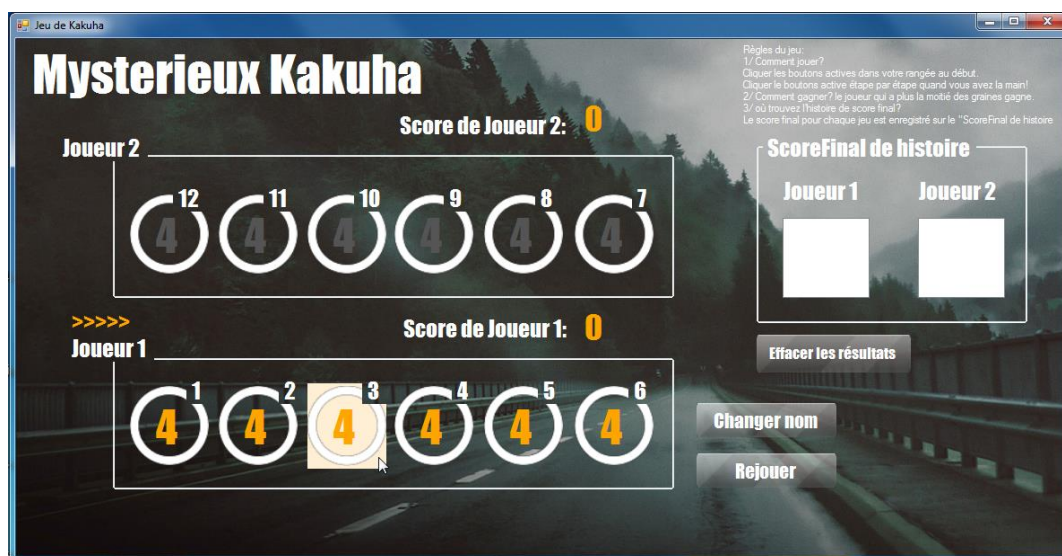
On ajoute un label pour expliquer quelques règles du jeu afin de faire les joueurs de mieux comprendre comment jouer ce jeu.

## (2) Un exemple des processus du jeu :

Au début, il y a deux MessageBox pour demander le nom de chaque joueur. Mais, ce n'est pas obligatoire d'entrer le nom.



Après ça, voilà l'interface entier. On initialise le jeu alors le joueur 1 commence et il peut cliquer n'importe quel bouton activé.

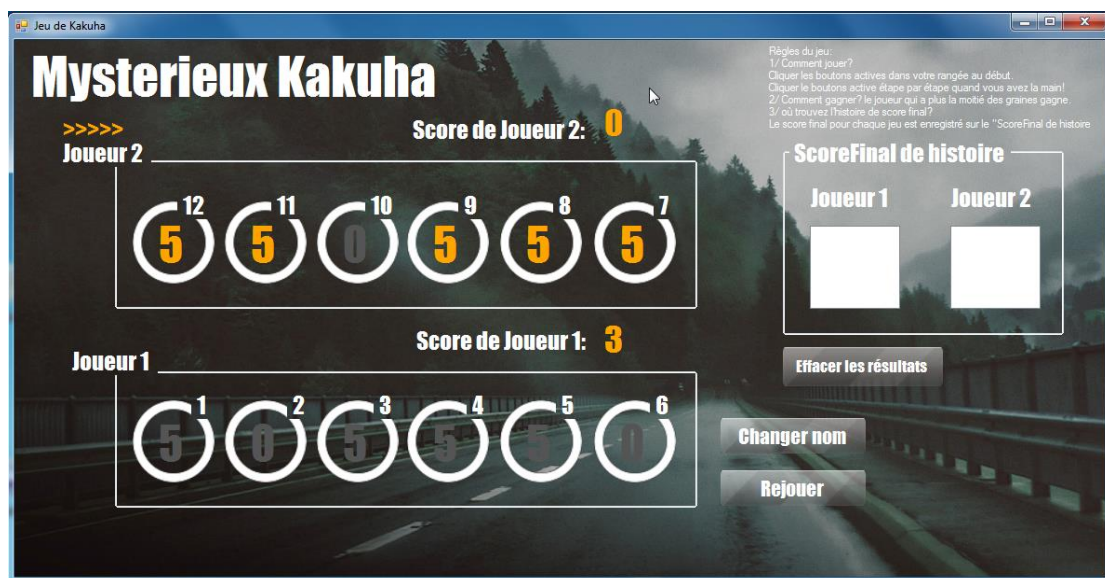


Le joueur choisit un bouton ensuite le clique. Le bouton pour l'étape suivante sera activé et les autres sont inactivés. Le joueur clique le bouton activé jusqu'à le changement de joueur. Et un MessageBox correspondant apparaîtra.

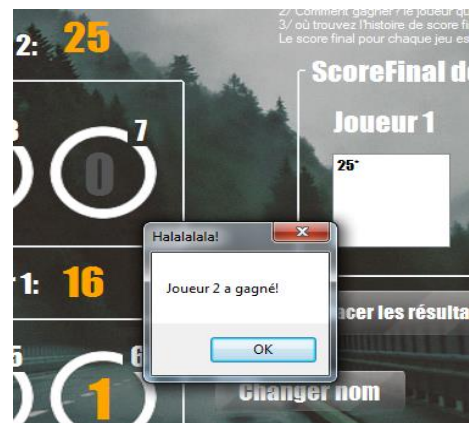




Le joueur est changé donc le joueur 2 a la main. Les boutons disponibles dans la rangée en haut sont activé afin de cliquer par le joueur 2.



Le jeu continue jusqu'à un des joueurs garde plus la moitié des graines alors le jeu est terminé.



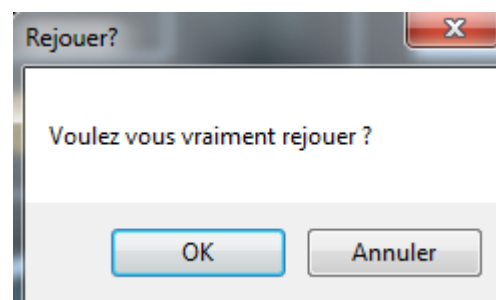


Et on demande les joueurs de changer les joueurs ou rejouer ce jeu.

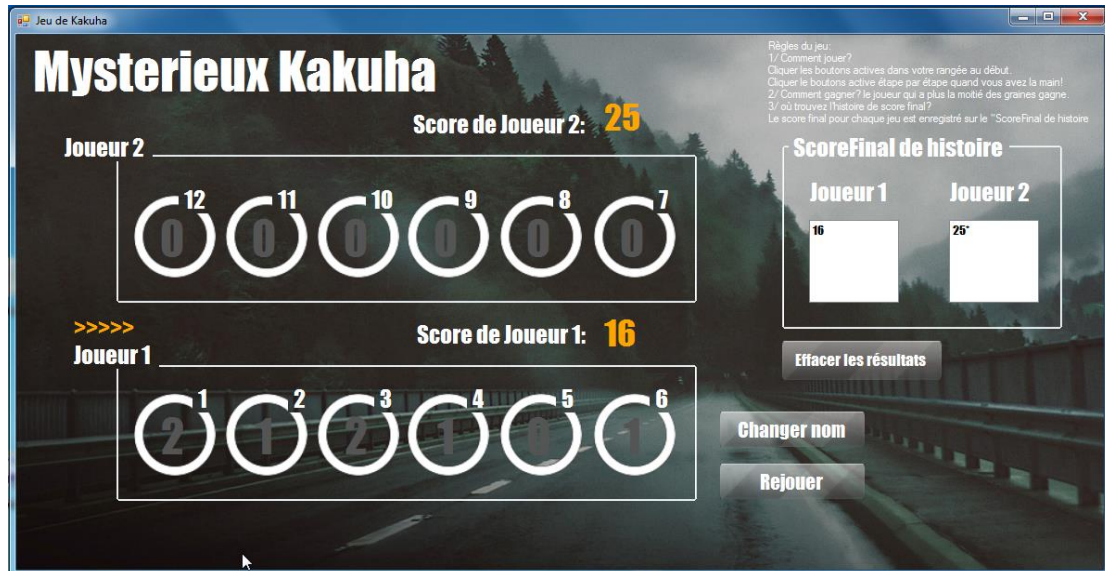


Avant rejouer, la liste de score enregistré va noter les scores.

Et on demande le joueur de confirmer son choix.



Le dernier écran avant le début du jeu suivant.



## 4 / Le code de l'interface

```
Public Class kakuha
    Public button(12) As Button
    Public lblscore(2) As Label

    Public Sub Tbutton()

        'on connecte les buttons et les labels qu'on définit au début avec les buttons
        et labels dans l'interface.
        button(1) = Button1
        button(2) = Button2
        button(3) = Button3
        button(4) = Button4
        button(5) = Button5
        button(6) = Button6
        button(7) = Button7
        button(8) = Button8
        button(9) = Button9
        button(10) = Button10
        button(11) = Button11
        button(12) = Button12

        lblscore(1) = score1
        lblscore(2) = score2

    End Sub

    'Partie InitialiseJeu :
    Public Sub InitialiseJeu()

        'On utilise les noms de utilisateurs pour les labels d'interface.

        nomJoueur1s.Text = nomJoueur1.Text
        nomJoueur2s.Text = nomJoueur2.Text
        lblscoreJ1.Text = "Score de " & nomJoueur1.Text & ":"
        lblscoreJ2.Text = "Score de " & nomJoueur2.Text & ":"

        'Au début, dans chaque trou, il y a 4 graines.
        For i = 1 To 12
            table(i) = 4
            button(i).Text = table(i)
        Next i
    End Sub
End Class
```

```

Next i

'Score de chaque joueur est de 0.
lblscore(1).Text = 0
lblscore(2).Text = 0
score(1) = 0
score(2) = 0

'on suppose que le joueur1 a la main premièrement et on met la flèche avant
le joueur1.
joueur = 1
flechejoueur1.Text = ">>>>"
flechejoueur2.Text = ""

End Sub
Public Sub btninitialise()

    'Par défaut, on suppose que le joueur 1 a la main, et tous les boutons attachés
    au joueur 1 sont disponible, et les autres sont indisponible.
    For i = 1 To 6
        button(i).Enabled = True
    Next
    For i = 7 To 12
        button(i).Enabled = False
    Next

End Sub
Public Sub bloque()

    'Dans cette procédure, on bloque tous les buttons.
    For i = 1 To 12
        button(i).Enabled = False
    Next

End Sub

```

#### Partie form\_load:

```

Private Sub kakuha_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    'Au début le jeu, on utilise les procédures pour initialiser.
    'Au début, on demande les utilisateurs d'entrer leur noms, si ils ne veulent
    pas les entrer,
    'Les noms de utilisateurs sont 'Joueur 1' et 'Joueur 2' par défaut.

```

```

nomJoueur1.Text = InputBox("Nom du joueur 1:", "Entrez un nom de joueur")
If nomJoueur1.Text = "" Then
    nomJoueur1.Text = "Joueur 1"
End If
nomJoueur2.Text = InputBox("Nom du joueur 2:", "Entrez un nom de joueur")
If nomJoueur2.Text = "" Then
    nomJoueur2.Text = "Joueur 2"
End If

Call Tbutton()
Call InitialiseJeu()
Call btninitialise()

```

End Sub

#### Contrôle de bouton:

```

Private Sub Button_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click, Button2.Click, Button3.Click,
    Button4.Click, Button5.Click, Button6.Click, Button7.Click, Button8.Click,
Button9.Click, Button10.Click, Button11.Click, Button12.Click

```

'On définit 'i' comme le septième caractère du nom des buttons.

'Par exemple, les six premiers caractères de 'Button' est 'b'u't't'o'n', et le septième est de '1', et on converti le caractère '1' à byte '1'.

```
Dim i As Integer = CByte(Mid(sender.name, 7))
```

```
Dim g As Integer = Cint(button(i).Text) 'On définit 'g' comme le nombre des
graines dans le trou.
```

Jouer(i) 'Dans un premier pas, on utilise la fonction 'Jouer' pour calculer les évolutions de nombre de sème dans les trous.

```
For j = 1 To 12 'On met à jour les text de button après la fonction 'Jouer'
    button(j).Text = table(j)
```

```
Next
```

lblscore(joueur).Text = CByte(lblscore(joueur).Text) + 1 'On ajoute score de 1 au joueur qui a la main.

```
Call bloque() 'On bloque tous les buttons.
```

If PeutJouer(i) = True Then 'Si le Boolean 'Peutjouer' est vrai, c'est-à-dire le trou suivant n'est pas vide. On peut débloquent le bouton qu'on doit choisir l'étape suivant.

If i + g - 1 < 12 Then 'Condition 1 : La somme du numéro de trou et le nombre de graines dans ce trou est inférieur à 12

If button(i + g).Text <> 0 Then  
button(i + g).Enabled = True  
End If

ElseIf i + g - 1 <= 24 Then 'Condition 2 : Si i + g - 1 > = 12, le coup de jouer va retourner au premier trou.

If button(i + g - 12).Text <> 0 Then  
button(i + g - 12).Enabled = True  
End If

Else

If button(i + g - 24).Text <> 0 Then 'Condition 3: Si i + g - 1 > 24, le coup de jouer va faire 2 retour.

button(i + g - 24).Enabled = True  
End If  
End If

Else 'si peutjouer est false, il faut changer le joueur

MsgBox("C'est le temp de changer le joueur!")

If joueur = 1 Then 'On initialise de la disponibilité de buttons après changer le joueur.

For i = 1 To 6  
If button(i).Text <> 0 Then  
button(i).Enabled = True  
End If

Next

flechejoueur1.Text = ">>>>>" 'quand le joueur a la main, il y a une flèche avant le label de nom de joueur.

flechejoueur2.Text = ""

Else

For i = 7 To 12  
If button(i).Text <> 0 Then  
button(i).Enabled = True  
End If



```

Next
flechejoueur1.Text = ""
flechejoueur2.Text = ">>>>"

End If

End If

If JeuTermine(1) Then 'l'affichage de terminer de jeu
    If score1.Text = score2.Text Then
        MsgBox(" Match nul ! ")
        MsgBox(" Cliquez le button 'Rejouer' pour jouer un autre tour !!!
Cliquez le button 'changer nom' pour changer le joueur !!! ")

    Else
        MsgBox(nomJoueur1.Text & " a gagné !")
        MsgBox(" Cliquez le button 'Rejouer' pour jouer un autre tour !!!
Cliquez le button 'changer nom' pour changer le joueur !!! ")
    End If

    scorefinaljoueur1.Items.Add(score(1) & "*") 'On ajoute le score dans la
listbox de histoire, et on met '*' avant le score de joueur qui a gagné.
    scorefinaljoueur2.Items.Add(score(2))
    Call bloque()

ElseIf JeuTermine(2) = True Then

    If score1.Text = score2.Text Then
        MsgBox("Match nul!")
        MsgBox(" Cliquez le button 'Rejouer' pour jouer un autre tour !!!
Cliquez le button 'changer nom' pour changer le joueur !!! ")
    Else
        MsgBox(nomjoueur2.Text & " a gagné! ")
        MsgBox(" Cliquez le button 'Rejouer' pour jouer un autre tour !!!
Cliquez le button 'changer nom' pour changer le joueur !!! ")
    End If

    scorefinaljoueur1.Items.Add(score(1))
    scorefinaljoueur2.Items.Add(score(2) & "*")
    Call bloque()
End If

```

End Sub

```
Private Sub BtnRejouer_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles BtnRejouer.Click
```

'Quand on clique le button de Rejouer, si le joueur clique 'OK', alors on initialise la disponibilité des buttons et les valeur des paramètres.

'C'est pour éviter la situation que le joueur clique 'Rejouer' accidentellement.

```
Dim r As Byte
```

```
r = MsgBox("Voulez vous vraiment rejouer ?", 1, "Rejouer?")
```

```
If r = vbOK Then
```

```
Call InitialiserJeu()
```

```
Call btninitialiser()
```

```
End If
```

```
If r = vbCancel Then
```

```
End If
```

End Sub

```
Private Sub Effacer_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles Effacer.Click
```

'quand on clique le button Effacer, les nombre de listbox d'historique sera effacés.

```
scorefinaljoueur1.Items.Clear()
```

```
scorefinaljoueur2.Items.Clear()
```

End Sub

```
Private Sub btnchangerNom_Click(sender As System.Object, e As System.EventArgs)  
Handles btnchangerNom.Click
```

'Au début, on demande les utilisateurs d'entrer leur noms, si ils ne veulent pas les entrer,

'Les noms de utilisateurs sont 'Joueur 1' et 'Joueur 2' par défaut.

```
nomJoueur1.Text = InputBox("Nom du joueur 1:", "Entrez un nom de joueur")
```

```
If nomJoueur1.Text = "" Then
```

```
nomJoueur1.Text = "Joueur 1"
```

```
End If
```

```
nomJoueur2.Text = InputBox("Nom du joueur 2:", "Entrez un nom de joueur")
```

```
If nomJoueur2.Text = "" Then
```

```
nomJoueur2.Text = "Joueur 2"
```

```
End If
```

```
        Call InitialiseJeu()  
        Call btninitialise()  
    End Sub  
End Class
```

## 5 / Le noyau fonctionnel

(On suppose que le joueur joue étape par étape jusqu'à un moment dont il faut changer le joueur)

```
Module Module1
```

```
    Public table(12) As Integer
```

```
    Public graines As Integer
```

```
    Public joueur As Integer
```

```
    Public fois As Integer 'ça veut dire le fois maximum qu'on peut semer selon le  
    nombre de graines à la main.
```

```
    Public score(2) As Integer
```

```
    Public nonvide As Boolean 'si le trou suivant n'est pas vide, nonvide est vrai.
```

```
Procédure Jouer :
```

```
    Public Sub Jouer(ByVal i As Integer)
```

```
        'par défaut, le joueur 1 peut commencer avance.
```

```
        'i : numéro de trou
```

```
        Dim n As Integer 'indice de fois qu'on peut semer
```

```
        graines = table(i) 'c'est le nombre de graines dans le trou qu'on choisit.
```

```
        fois = graines - 1 'On sait qu'il faut garder la dernier graine, donc on déduit  
une graine au début.
```

```
        table(i) = 0
```

```
        'Au premier cas, on suppose que le numéro du trou dernier qu'on sème (= i  
+ fois) est inférieure ou égale à 12.
```

```
        'On vide le trou qu'on choisit et on sème les graines une à une jusqu'à l'avant  
dernière.
```

```
        If i + fois <= 12 Then
```

```
            For n = 1 To fois
```

```
                table(i + n) = table(i + n) + 1
```

```
            Next
```

```
        ElseIf i + fois <= 24 Then
```

```
            'Quand le numéro du trou dernier qu'on sème est supérieure à 12, il faut  
recommencer à partir du numéro 1.
```

'Par exemple, supposons que  $i = 10$ , graines = 5, fois = 4, donc  $(i + \text{graines} - 1) = 14$  est bien supérieure à 12.

'En effet, on doit semer les graines à trous 11,12 et 1,2

'Donc on divise ce cas en 3 parties.

'Partie 1 : La somme du numéro de trou et le numéro de graines dans le trou est inférieure à 12.

'ici c'est l'évolution de nombre de graine dans le trou 1 et 2, parce qu'on doit passer la rangée de "7-12" à "1-6", autrement dire, la deuxième tour.

'pour recommencer à partir du numéro 1, on doit savoir qu'on doit semer combien de fois au trou ce qui est sur la rangée "1-6"

For n = 1 To i + fois - 12

' (i + fois - 12) est le numéro du trou dernier qu'on sème à la nouvelle rangée. Egalement, ça égale la fois de sème car cette rangée commence à 1. Dans ce exemple, Pour n = 1 à 2

'Dans l'exemple, Table(10) = 0, c-à-d qu'on vide le trou qu'on choisi  
table(n) = table(n) + 1      'On considère n comme la fois et le numéro en même temp. Plus précisément, ici, il reste 2 fois de sème : première fois est dans le trou (1), deuxième fois est dans le trou (2).

Next

'Partie 2 : le coup de jouer va retourner au premier trou.

'ici c'est l'évolution de nombre de graine dans le trou 11 et 12

For n = 1 To 12 - i '(12-i) est pour savoir qu'il y a combien de fois de sème entre trou(i) et trou (12).

table(i + n) = table(i + n) + 1

Next

Else

'Partie 3 : Si  $i + g - 1 > 24$ , le coup de jouer va faire 2 retour.

For n = 1 To 12 - i

table(i + n) = table(i + n) + 1

Next

For n = 1 To 12

table(n) = table(n) + 1

Next

For n = 1 To fois - 12 - (12 - i)

table(n) = table(n) + 1

Next

End If

'On met à jour des nouveaux nombres des graines dans les trous.

score(joueur) = score(joueur) + 1

End Sub

Fonction PeutJouer :

Public Function PeutJouer(ByVal i As Integer) As Boolean

'Si on n'a plus de graines sur la main et va prendre les graines au trou prochaines. Du coup, avant passer l'étape prochaine on vérifie le nombre de graines du trou suivant.

'Si le trou suivant est 0, le joueur s'arrête ici et on change le joueur.

'i est le numéro de trou

'Dans ce fonction on utilise 'Joueur' qu'on le défini au début de module

nonvide = True 'On suppose qu'il existe des graines dans le trou dans lequel il aurait dû prendre des graines.

fois = graines - 1 'On sait qu'il faut garder la dernier graine, donc on déduit une graine au début.

If i + fois < 12 Then 'Cas 1:

If table(i + fois + 1) = 0 Then

nonvide = False

'Le trou suivant est vide

If joueur = 1 Then

'Dans ce cas, si les trous de Joueur 2 est tous vide, on n'a pas besoin de changer le joueur. C'est encore joueur 1 qui a la main.

If table(7) <> 0 Or table(8) <> 0 Or table(9) <> 0 Or table(10) <> 0 Or table(11) <> 0 Or table(12) <> 0 Then

joueur = 3 - joueur 'C'est pour changer le joueur. Par exemple, joueur 1 s'arrête ici, numéro du joueur = 3 - 1 = 2, c'est-à-dire le joueur 2 a la main.

End If



```

Else
    If table(1) <> 0 Or table(2) <> 0 Or table(3) <> 0 Or table(4)
<> 0 Or table(5) <> 0 Or table(6) <> 0 Then

        joueur = 3 - joueur

    End If
End If

End If

ElseIf i + fois < 24 And i + fois >= 12 Then 'Cas 2:
    If table(i + fois + 1 - 12) = 0 Then
        nonvide = False
        If joueur = 1 Then
            If table(7) <> 0 Or table(8) <> 0 Or table(9) <> 0 Or table(10)
<> 0 Or table(11) <> 0 Or table(12) <> 0 Then

                joueur = 3 - joueur
            End If
        Else
            If table(1) <> 0 Or table(2) <> 0 Or table(3) <> 0 Or table(4)
<> 0 Or table(5) <> 0 Or table(6) <> 0 Then

                joueur = 3 - joueur
            End If
        End If
    End If

ElseIf i + fois >= 24 Then 'Cas 3:
    If table(i + fois - 24 + 1) = 0 Then
        nonvide = False
        If joueur = 1 Then
            If table(7) <> 0 Or table(8) <> 0 Or table(9) <> 0 Or table(10)
<> 0 Or table(11) <> 0 Or table(12) <> 0 Then

                joueur = 3 - joueur
            End If
        Else
            If table(1) <> 0 Or table(2) <> 0 Or table(3) <> 0 Or table(4)
<> 0 Or table(5) <> 0 Or table(6) <> 0 Then

                joueur = 3 - joueur

```

```

        End If
    End If
End If
Return nonvide
End Function

Public Function JeuTermine(ByVal joueur As Integer) As Boolean

    If score(joueur) > 24 Or score(1) = score(2) = 24 Then 'Si un joueur garde
plus la moitié des graines, ce joueur a gagné. Si 2 joueurs ont le même nombre de
graines de 24, c'est un match nul.
        JeuTermine = True
    Else
        JeuTermine = False
    End If

    Return JeuTermine

End Function

End Module

```

## 6 / Conclusion

Grâce à ce projet, nous avons compris mieux les cours de programmation. Ce qui a été le plus difficile c'était de penser à toutes les situations possibles pour pouvoir ensuite corriger au mieux s'il y a une erreur.

Pour afficher directement le résultat final selon le coup pour un joueur, nous avons proposé une idée de solution :

```
Public Sub Jouer2(ByVal i As Integer)
    'Jouer2 affiche directement le résultat final d'un coup
    'Le paramètre i correspond au numéro du trou choisi par le joueur qui a la
    main.

    Dim fois As Integer 'pour calculer le score du joueur
    Dim passe As Boolean 'pour déterminer si le joueur peut passer à l'étape
    deuxième
    Dim suivantok As Boolean 'si le nombre de graine dans le trou prochain est
    1, le joueur peut jouer sur le trou prochaine mais pas sur le deuxième trou
    Dim x As Integer 'x= i +2

    'initial
    fois = 0 'le score du jouer est 0.
    graines = table(i) : table(i) = 0 : button(i).Text = table(i)
    'le joueur choisit le trou(i) et prend en main les graines du trou. Du coup, les
    trou(i) = 0.
    passe = True
    suivantok = True

    While suivantok = True

        If graines = 1 Then 'si le joueur choisit trou(i) et le nombre des
        graines au trou(i) est 1.
            fois = fois + 1 : graines = 0
            If table(i + 1) > 0 Then 'avant de passer l'étape suivant, vérifier
            le nombre des graines au trou suivant. si nb de graine>0, c-à-d, nb=1 ou nb> 1
                If table(i + 1) = 1 Then 'si nb de graine = 1, le joueur peut
                jouer sur le trou prochaine mais pas sur le deuxième trou
                    suivantok = True : passe = False
                    graines = table(i + 1) : table(i + 1) = 0 : button(i + 1).Text
                    = table(i + 1) 'le jeu continue, le joueur prend en main les graines du trou(i+1)
                    i = i + 1
```

```

Else 'table(i+1)>1
    passe = True : suivantok = True 'le jeu continue
End If
Else 'table(i+1)=0
    passe = False : suivantok = False 'Aucun grain en main, aucun
grain au trou suivant, alors arrêter. C'est le moment de changer le joueur
End If
Else 'graine>1
    passe = True : suivantok = True
End If

While passe = True 'ça veut dire que le joueur a plusieurs graines (>=2)
et il peut semer au moins un trou.
    If i > 10 Then 'assurer le cycle du trou
        If i = 11 Then
            i = 12
            x = 1 'x=i+2, ici, x=13, c'est le trou 1
        Else
            i = 1
            x = 2
        End If
    Else 'le trou 1.2....10
        i = i + 1 'le numéro du trou prochain
        x = i + 1 'car i est déjà changé
    End If
    table(i) = table(i) + 1 : button(i).Text = table(i) 'dans le processus
de seme

    graines = graines - 1 'semer une fois, nb de graine réduit 1

    ' après chaque étape de semis, vérifier le nb de graine en main
    If graines = 1 Then 'après semer, il reste seulement un grain,
le joueur le garde
        fois = fois + 1 : graines = 0
        If table(x) = 0 Then 'en effet, c'est le trou prochain ensuite
vérifier le deuxième trou
            passe = False 'Aucun grain en main, aucun grain au trou
suivant, alors arrêter. c'est le moment de changer le joueur
        Else 'table(x) >0, le jeu continue
            graines = table(x) : table(x) = 0 : button(x).Text = table(x)
            i = x
            passe = True
        End If

```

```
        Else    'graines en mains > 1, le jeu continue
            passe = True
        End If
    End While

End While

score(joueur).Text = score(joueur).Text + CInt(fois)
'afficher le score sur le form

End Sub
```