

Il programma scolastico di laboratorio di Sistemi ,riguardante la programmazione in Visual Basic , continua con la creazione dell'applicazione:

Tris

Il problema

Si vuole creare una applicazione VB 6.0 che consenta di giocare a tris con avversario il computer.

Analisi preliminare

Struttura del gioco

- Il Tris è un gioco da tavolo che richiede la partecipazione di due giocatori.
- All'inizio di ogni partita si decide ,casualmente , chi dei partecipanti deve iniziare per primo.
- Stabilito l'ordine di gioco ,i giocatori scelgono un segno tra: X e O .
- Durante il proprio turno i giocatori devono effettuare una mossa , cioè , inserire il proprio segno nell'apposita griglia 3x3 ,qui riprodotta:

- Effettuato il segno , il giocatore corrente finisce il turno e cede il gioco all'avversario.
- Il gioco finisce quando un giocatore è riuscito a mettere tre segni di fila , cioè ha fatto Tris, oppure quando tutte le celle sono già state occupate.

I vincoli e le funzionalità individuate

Unico **vincolo** del programma sarà l'**intelligenza strategica del computer** .

Nel programma da creare ,oltre alla funzionalità obbligatoria **gestione gioco** , verranno implementate le seguenti funzionalità aggiuntive: **score** , per registrare su disco i risultati delle ultime cinque partite, **cambia immagine** , per cambiare l'immagine di sfondo del form principale , **informazioni su tris** , per visualizzare le informazioni sull'applicazione tris.exe . Il motivo dell'inserimento di tali complicanze è spiegabile dall'intento di imparare la gestione dei file in ambiente VB e di acquisire nuove conoscenze.

Dati di Input

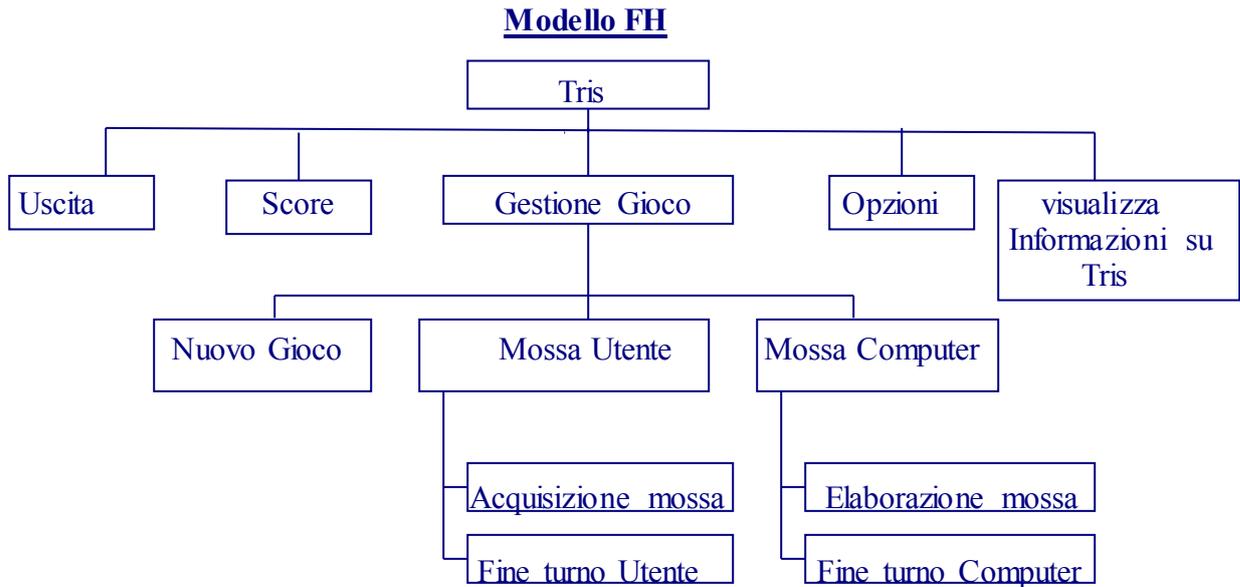
Mossa Utente (l'indice di una cella della tabella del tris)

Dati di Output

Stato del Gioco (Gioco On - vittoria giocatore 1 - vittoria giocatore 2 - pareggio) e **tabella grafica del tris**, indicante ,in **real-time**, le mosse effettuate dai giocatori ad ogni loro turno.

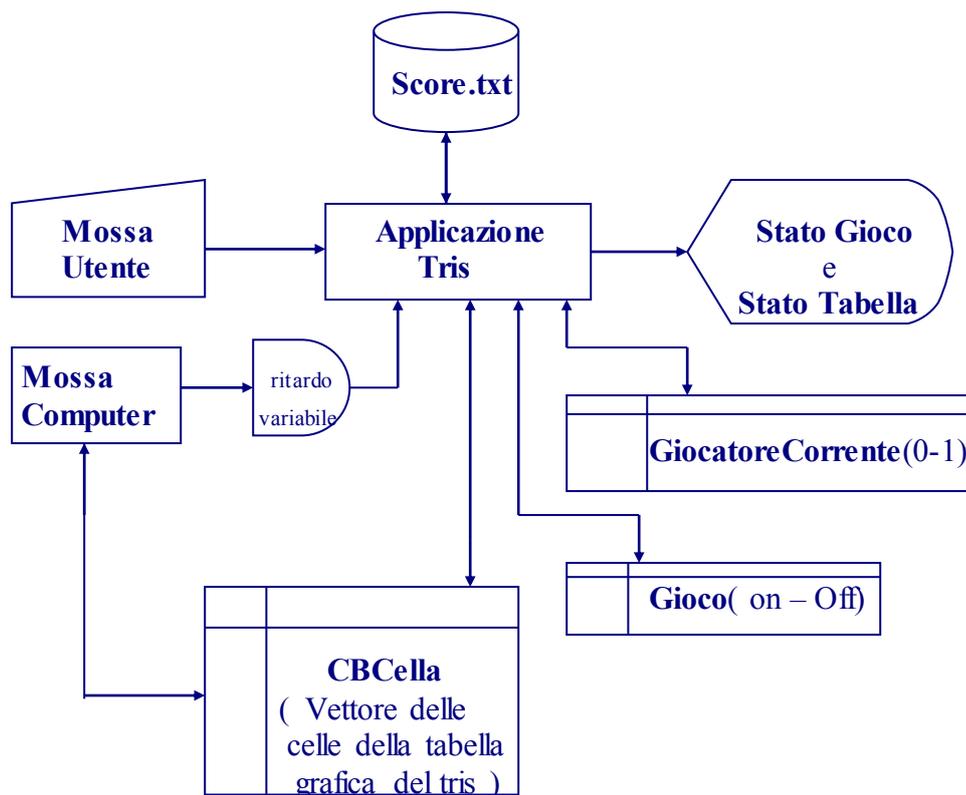
Analisi del problema

L'applicazione da creare sarà basata sul funzionigramma seguente.



E utilizzerà le seguenti risorse di sistema:

System resources chart



in cui:

- **CBCella**, di classe CommandBotton, rappresenterà la tabella grafica del tris.
- **Mossa Utente**, l'indice di una cella della tabella del tris, sarà un intero con valore pari all'index del vettore CBCella, acquisito durante il verificarsi dell'evento clic su un controllo CBCella.

- **Mossa Computer** sarà una procedura ,che verrà richiamata dall'applicazione tris, che si occuperà di gestire il gioco del computer , cioè , di elaborare una mossa ed eseguirla. Prima di inserire il segno nella tabella ,però, viene effettuato un *ritardo di lunghezza variabile* , dipendente dal numero di controlli che sono stati effettuati su CBCella per elaborare la mossa.
- **GiocatoreCorrente** e **Gioco** , sono due variabili Boolean che verranno utilizzati per la gestione del gioco.
- **Score.txt** è un file di testo situato nella stessa directory del file binario tris , che verrà utilizzato per implementare la funzionalità score.

La progettazione

Le specifiche tecniche per la creazione dei moduli del progetto.

L'applicazione ,da realizzare in ambiente VB 6.0 su un singolo *progetto di avvio* ,chiamato Tris, sarà formata da quattro **form**: un form principale , che sarà *l'oggetto di avvio*, chiamato **frmTris**; e tre form secondari chiamati: **frmAbout** , **frmScore** , **frmOpzioni** che verranno richiamati dal menù del form principale; e dal modulo **mdlTris** , contenente le variabili globali.

Nel form principale verrà riprodotta la tabella del tris utilizzando i controlli CommandButton , come precedentemente annunciato,segundo lo schema a pag. 1.

Tali controlli apparterranno ad un vettore di dimensione fissa pari a 9 , chiamato CBCella dato che ogni oggetto del vettore rappresenterà una cella della tabella del tris.

All'interno del form verranno inseriti ,inoltre: **la barra dei menù** e gli oggetti **ImgX** e **ImgO** di classe Image . Mentre questi ultimi serviranno per contenere l'immagine bitmap dei segni grafici X e O , da visualizzare sulla superficie dei bottoni, la barra dei menù consentirà all'utente di:

Giocare una nuova partita , *uscire dall'applicazione* , *visualizzare frmScore* , contenente i risultati delle ultime cinque partite giocate, *visualizzare frmOpzioni* , contenente due bottoni che permettono di cancellare i risultati archiviati sul file score e cambiare lo sfondo di *frmTris* , *visualizzare frmAbout* , contenente le informazioni su l'applicazione.

Nel form **frmTris** verranno inseriti anche degli oggetti di importanza secondaria, quali: **tmGame** , di classe **timer**, che conterà i secondi trascorsi dall'inizio di una nuova partita, e **ProgressBar1**, di classe **ProgressBar**, che visualizzerà il tempo di elaborazione di ciascuna mossa del computer.

Per semplificare l'applicazione, ho deciso di fissare i segni dei giocatori: l'utente avrà segno X ed il computer segno O.

Funzione: Gestione gioco

Il funzionamento della funzionalità più importante dell'applicazione ,gestione gioco, sarà il seguente:

- Quando viene caricata l'applicazione , oppure seleziona la voce *nuova partita* dal menù , verrà richiamata la **Sub Nuova_partita**, che: cancellerà le modifiche applicate a **CBcella** nelle precedenti partite e assegnerà alla var. **gioco true** e alla var. **giocatoreCorrente** un valore casuale **true** o **false** , indicante con **true** il gioco utente e con **false** il gioco computer. Se , a termine, il valore di tale var. sarà **false** verrà richiamata la **Sub GiocoComputer**.
- Nella **Sub GiocoComputer** verrà: elaborata ed effettuata una mossa , richiamata la procedura **ControlloFinePartita**, e assegnato alla var. **giocatoreCorrente true** ,per far sì che il gioco passi all'utente.
- Nella **Sub ControlloFinePartita**, se nella tabella non esistono celle vuote o se è stato effettuato un tris , si assegnerà alla var. **gioco false**.
- Nella procedura di risposta all'evento click su **CBcella(index)** ,se **gioco** e **giocatoreCorrente** sono **true**, verranno: acquisite le mosse dell'utente , verificando se l'utente ha cliccato su una cella già contrassegnata, eseguite, settando le proprietà di **CBcella(index)** di **tab** e di **picture** con ,rispettivamente, "X" e **ImgX.picture** , e richiamata la **Sub ControlloFinePartita** e , se **gioco = true** ,anche la **Sub GiocoComputer**.

Funzione: Elaborazione mossa di Gestione Gioco

L'elaborazione della mossa Computer , nella Sub **GiocoComputer** , sarà risolta in questo modo: Inizialmente si tenta di applicare una **mossa strategica**, se ciò non è possibile si applica una **mossa casuale**. La mossa strategica consiste in una mossa che ostacola un possibile futuro tris dell'avversario. Ad esempio se la tabella del tris contiene:

X	X	
	O	

GiocoComputer eseguirà la mossa :

X	X	O
	O	

Ma ,anche, in una mossa offensiva, ad esempio se la tabella del tris contiene:

X	X	O
	O	

GiocoComputer eseguirà la mossa :

X	X	O
	O	
O		

Ricerca dell'algoritmo che simula un'intelligenza artificiale del computer

L'algoritmo elaborato , come possiamo vedere nella codifica a pag. 8, consiste:

- in una serie di cicli for che contano il numero delle O e delle celle vuote presenti: nelle celle di ogni riga ,per verificare un tris se è possibile fare un tris orizzontale , nelle celle di ogni colonna ,per verificare un tris verticale, e nelle celle della diagonale principale e secondaria,per verificare un tris obliquo; eseguendo la mossa strategica ,se il numero delle O è uguale a due e c'è una cella vuota, nell'index della cella vuota.
- in una serie di cicli for che contano il numero delle X e delle celle vuote presenti: nelle celle di ogni riga ,per verificare un possibile tris orizzontale dell'utente , nelle celle di ogni riga colonna ,per verificare un tris verticale, e nelle celle della diagonale principale e secondaria,per verificare un tris obliquo; eseguendo la mossa strategica ,se il numero delle X è uguale a due e c'è una cella vuota, nell'index della cella vuota.

Funzione: Score

La funzione Score , come precedentemente annunciato, utilizzerà il file score.txt ,situato nella stessa directory del file eseguibile da creare , per archiviare i risultati delle ultime cinque partite effettuate. Per realizzarla si è pensato di :

- leggere il contenuto del file score.txt ,nella procedura di risposta all'evento caricamento dell'oggetto di avvio del progetto , cioè la Sub Form_Load dell'oggetto frmTris , costituito da cinque righe contenenti:due interi, rappresentanti il numero di partite effettuate e di partite vinte in totale ,e una stringa, rappresentante la data e l'ora della fine del ultima partita giocata; e memorizzarlo nelle variabili globali del progetto ,che saranno situate nel modulo mdlTris.mdl .
- Scrivere nel file score.txt , nella procedura di risposta all'evento scaricamento dell'oggetto di avvio del progetto, cioè la Sub Form_Unload dell'oggetto frmTris , i risultati delle ultime cinque partite memorizzate nelle variabili globali del progetto.
- Aggiornare alla fine di ogni partita le variabili globali.
- Aggiornare le proprietà degli oggetti label del form frmScore, che si occuperanno di visualizzare la classifica score ,con il valore delle var. globali ,ad ogni caricamento del form che le contiene o alla fine di ogni partita.

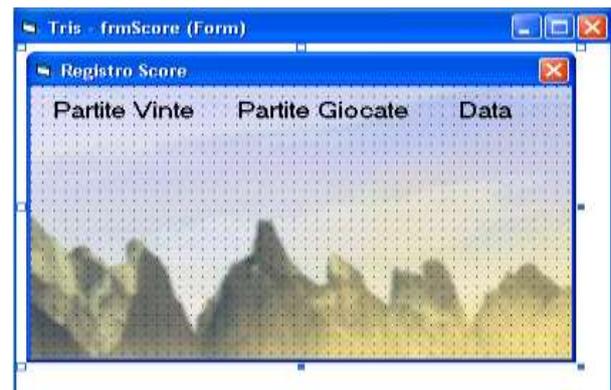
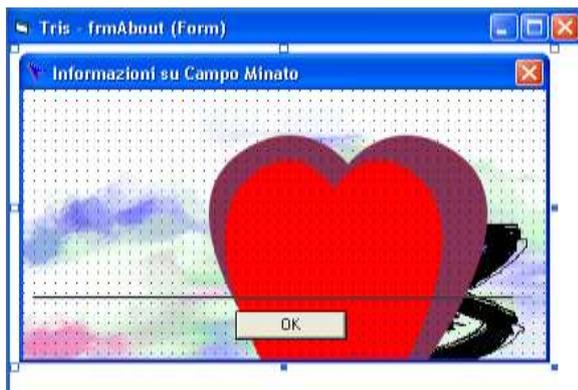
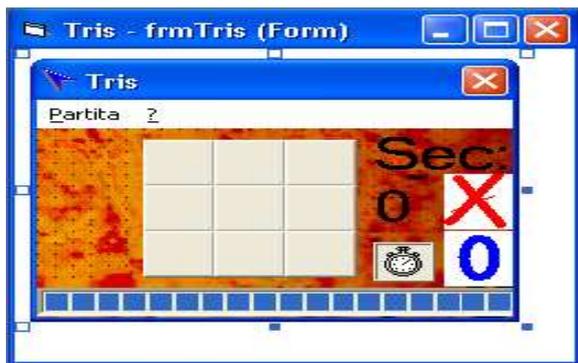
La realizzazione dell' applicazione

In basso viene presentato l'elenco completo degli oggetti inseriti nel progetto; individuati seguendo le specifiche tecniche precedentemente esposte.

Tris (Tris.vbp)

- **form** frmTris (frmTris.frm)
 - **barra dei menù**
 - **Image** ImgX
 - **Image** ImgO
 - **Timer** TmGame
 - **CommandButton** CBCella
 - **label** LabSec
 - **label** LabNunSec
 - **ProgressBar** ProgressBar1
- **form** frmAbout (frmAbout.frm)
 - **label** lbITitle
 - **label** lbIVersion
 - **label** lbIDescription
 - **CommandButton** cmdOK
 - **line** Line1
- **form** frmOpzioni (frmOpzioni.frm)
 - **CommandButton** cmdScore
 - **CommandButton** cmdImage
 - **CommandButton** cmdChiudi
 - **CommonDialog** CommonDialog1
- **form** frmScore (frmScore.frm)
 - **label** lbVociScore
 - **label** lbPartiteVinte(0 to 5)
 - **label** lbPartiteGiocate (0 to 5)
 - **label** lbData(o to 5)
- **modulo** mdlTris(mdlTris.bas)

La disposizione dei controlli sui rispettivi **form** sarà la seguente:



Impostazione delle proprietà del progetto Tris

Tipo Progetto	Exe standard
Oggetto di avvio	frmTris
Titolo	Tris
Icona	frmTris
Numero Versione Principale	1
Numero Versione Secondario	4
Numero Versione Revisione	0
Comments	Applicazione dimostrativa delle potenzialità di Visual Basic. Ideazione , progettazione e sviluppo di: >- Caruso Davide -<

Scrittura del codice del Modulo:mdlTris

'spazio per la dichiarazione delle variabili globali del progetto

Public partiteVinte(0 To 4), partiteGiocate(0 To 4), data(0 To 4)

Impostazione delle proprietà degli oggetti del form: frmTris

Timer sarà impostato con **Enabled** = False e **Interval** = 500.

L'oggetto **CBCella** in alto a destra della finestra avrà indice 0 ; tutti gli altri avranno un indice che sarà: maggiore di uno rispetto a quello stessa riga e della colonna precedente , e maggiore di 3 rispetto a quello stessa colonna e della riga precedente. Ed inoltre , tutti gli oggetti CBCella avranno la proprietà **MaskColor** settata con il colore &H00FFFFFF&

Gli oggetti **ImgX** e **ImgO** saranno impostati con **visibile False** e **picture** : ImgX.bmp e ImgO.bmp , che appariranno sulla superficie dei bottoni , grazie alle modifica sull'attributo **MaskColor** , senza lo sfondo bianco.

Impostazione del Menù

Caption	Name	Index	Enabled	Shortcut
&Partita	mnuPartita		True	
...&Nuova Partita	mnuPartitaItem	0	True	
...&Opzioni	mnuPartitaItem	1	True	
...&Score	mnuPartitaItem	2	True	
...-	mnuPartitaItem	3	True	
...&Esci	mnuPartitaItem	4	True	Ctrl+C
&?	mnuHelp		True	
...Guida in linea	mnuhelpItem	0	False	F1
...-	mnuhelpItem	1	True	
...mnuhelpItem	mnuhelpItem	2	True	

Scrittura del codice

'Area Generale

Dim giocatoreCorrente As Boolean *'false = pc , true = Utente*

Dim gioco As Boolean *'false = gameOver , true = gameOn*

Dim sec As Double *'conterà il numero di secondi trascorsi dall'inizio di una partita*

Dim parGiocate *'conterà il numero di partite giocate*

Dim parVinte *'conterà il numero di partite vinte*

'Procedura di risposta all'evento: caricamento dell'oggetto di avvio frmTris

```
Private Sub Form_Load()  
    Randomize 'setta il seme del generatore di numeri pseudocasuali  
    sec = 0 'azzeri i secondi  
    leggiScore 'vengono inizializzate le var globali di mdlTris , parGiocate  
    ' e parVinte  
    NuovaPartita 'vengono inizializzati: CBCella , gioco e giocatoreCorrente  
    ' per una nuova partita  
End Sub
```

'Procedura di risposta all'evento:click su CBCella

```
Private Sub CBCella_Click(Index As Integer)  
    If gioco And giocatoreCorrente And CBCella(Index).Tag = "" Then 'verifica validità mossa  
        CBCella(Index).Tag = "X" 'scrittura segno X  
        CBCella(Index).Picture = ImgX.Picture  
        ContolloFinePartita ("X") ' Controllo Fine Partita per tris utente o fine celle  
        GiocoComputer 'il gioco passa al computer  
    End If  
End Sub
```

'Procedura per il ripristino di: CBCella , gioco e giocatoreCorrente ,per una nuova partita

```
Private Sub NuovaPartita()  
    frmTris.Caption = "Tris"  
    TmGame = True  
    sec = 0  
    For i = 0 To 8  
        CBCella(i).Tag = ""  
        CBCella(i).Picture = LoadPicture() 'viene cancellata la picture di CBCella(i)  
    Next  
    gioco = True  
    giocatoreCorrente = Int(Rnd + 0.5) 'giocatoreCorrente sarà false se 0<= RND < 0.5 e  
    true se 0.5<= RND<1  
    If (Not giocatoreCorrente) Then GiocoComputer 'se giocatoreCorrente= false inizia il computer  
End Sub 'altrimenti gioca l'utente
```

'Procedura per il Controllo Fine Partita per fine celle o tris del giocatore segno

```
Private Sub ContolloFinePartita(segno As String)  
    Dim i As Integer, mosse As Integer  
    For i = 0 To 8  
        If CBCella(i).Tag <> "" Then mosse = mosse + 1 'conta delle celle piene  
    Next  
    If mosse = 9 Then frmTris.Caption = "fine partita" 'fine della partita se mosse = 9  
    If ContolloTris(segno) Then  
        frmTris.Caption = "Ha vinto: " & segno & " !"  
        If segno = "X" Then parVinte = parVinte + 1 'incremento parVinte se vi è un tris Utente  
    End If  
    If frmTris.Caption <> "Tris" Then  
        parGiocate = parGiocate + 1  
        lbScore.Caption = "Partite vinte: " & parVinte & " su " & parGiocate  
        gioco = False  
        TmGame = False  
        aggiornaScore 'aggiognerà le delle variabili del modulo  
    End If  
End Sub
```

' Function per il Controllo di un Tris da parte del giocatore segno

Private Function ControlloTris(segno As String) As Boolean

Dim y As Integer

Dim i As Integer

Dim x As Integer

ControlloTris = False

For y = 0 To 2 *'controllo tris orizzontale*

x = 0

For i = 0 To 2

If CBCella(i + (y * 3)).Tag = segno Then x = x + 1

Next

If x = 3 Then ControlloTris = True

Next

For y = 0 To 2 *'controllo tris verticale*

x = 0

For i = 0 To 2

If CBCella(y + (i * 3)).Tag = segno Then x = x + 1

Next

If x = 3 Then ControlloTris = True

Next

x = 0 *'controllo tris diagonale principale*

For i = 0 To 2

If CBCella(i + (i * 3)).Tag = segno Then x = x + 1

Next

If x = 3 Then ControlloTris = True

x = 0 *'controllo tris diagonale secondaria*

For i = 0 To 2

If CBCella(i + (6 - (i * 3))).Tag = segno Then x = x + 1

Next

If x = 3 Then ControlloTris = True

End Function

' Procedura per la gestione del gioco Computer

Private Sub GiocoComputer()

If Not gioco Then Exit Sub *'controllo game over*

If Not strategiaDifensiva Then *'se non si applica nessuna mossa strategica*

Dim a As Integer *'viene effettuata una mossa casuale*

Do

a = Rnd * 8

Loop While (CBCella(a).Tag <> "")

CBCella(a).Tag = "0"

CBCella(a).Picture = Img0.Picture

End If

ControlloFinePartita ("0") *'viene verificato l'eventuale tris o la fine delle celle vuote*

ProgressBar1.Value = 0.1 *'viene azzerato il valore della ProgressBar1*

giocatoreCorrente = True *'il gioco passa all'utente*

Beep *'viene avvisato l'utente con un segnale sonoro*

End Sub

' Procedura per l'elaborazione di una mossa con strategia offensiva o difensiva .

' Questa è la procedura più complessa , e come potete vedere si limita all' effettuazione di un tris, in caso di due celle "O" e una cella vuota allineati , e ad ostacolare l'eventuale tris dell'utente .

Private Function strategiaDifensiva() As Boolean

Dim y As Integer

Dim i As Integer

Dim x As Integer

Dim posto As Integer

Dim paus As Integer

strategiaDifensiva = False

'----- La migliore difesa è l'attacco

'controllo possibile tris orizzontale

For y = 0 To 2

 x = 0

 posto = 0

 For i = 0 To 2

 If CBCella(i + (y * 3)).Tag = "0" Then

 x = x + 1

 Else

 posto = i + (y * 3)

 End If

 Next

 If x = 2 And CBCella(posto).Tag = "" Then GoTo sal

Next

'controllo possibile tris verticale

paus = paus + 1

For y = 0 To 2

 x = 0

 posto = 0

 For i = 0 To 2

 If CBCella(y + (i * 3)).Tag = "0" Then

 x = x + 1

 Else

 posto = y + (i * 3)

 End If

 Next

 If x = 2 And CBCella(posto).Tag = "" Then GoTo sal

Next

'controllo possibile tris diagonale principale

paus = paus + 1

x = 0

posto = 0

For i = 0 To 2

 If CBCella(i + (i * 3)).Tag = "0" Then

 x = x + 1

 Else

 posto = i + (i * 3)

 End If

Next

If x = 2 And CBCella(posto).Tag = "" Then GoTo sal

'controllo possibile tris diagonale secondaria

paus = paus + 1

x = 0

posto = 0

For i = 0 To 2

 If CBCella(i + (6 - (i * 3))).Tag = "0" Then

 x = x + 1

 Else

 posto = i + (6 - (i * 3))

 End If

Next

If x = 2 And CBCella(posto).Tag = "" Then GoTo sal

paus = paus + 1

'controllo possibile tris avversario orizzontale

For y = 0 To 2

 x = 0

 posto = 0

 For i = 0 To 2

 If CBCella(i + (y * 3)).Tag = "X" Then

 x = x + 1

 Else

 posto = i + (y * 3)

 End If

 Next

 If x = 2 And CBCella(posto).Tag <> "0" Then GoTo sal

Next

paus = paus + 1

'controllo possibile tris avversario verticale

For y = 0 To 2

 x = 0

 posto = 0

 For i = 0 To 2

 If CBCella(y + (i * 3)).Tag = "X" Then

 x = x + 1

 Else

 posto = y + (i * 3)

 End If

 Next

 If x = 2 And CBCella(posto).Tag <> "0" Then GoTo sal

Next

'controllo possibile tris avversario diagonale principale

paus = paus + 1

x = 0

posto = 0

For i = 0 To 2

 If CBCella(i + (i * 3)).Tag = "X" Then

 x = x + 1

 Else

 posto = i + (i * 3)

 End If

Next

If x = 2 And CBCella(posto).Tag <> "0" Then GoTo sal

'Procedura di gestione della barra dei menù

'Procedura di risposta all'evento:click sul menù principale Partita

```
Private Sub mnuParamItem_Click(Index As Integer)
    Select Case Index
        Case 0
            NuovaPartita
        Case 1
            frmOpzioni.Show
        Case 2
            frmScore.Show
        Case 4
            End 'termina il programma
    End Select
End Sub
```

'Procedura di risposta all'evento:click sul menù principale' ?'

```
Private Sub mnuhelpItem_Click(Index As Integer)
    If Index = 2 Then frmAbout.Show 'viene aperta la finestra frmAbout
End Sub
```

'Procedura per l'implementazione della funzionalità score

'Procedura per la lettura del contenuto del file score.txt e l'inizializzazione delle variabili del

'moduloee ,di parGiocate e di parVinte

```
Public Sub leggiScore()
    Open ".\score.txt" For Input As #1 'Apre il file per l'output.
    For i = 0 To 4
        Input #1, partiteGiocate(i), partiteVinte(i), data(i)
    Next
    Input #1, parGiocate, parVinte
    parGiocate = LTrim(parGiocate) 'Consente di rimuovere gli spazi iniziali da una stringa
    parVinte = LTrim(parVinte)
    lbScore.Caption = "Partite Vinte " & parVinte & " su " & parGiocate
    Close #1 'Chiude il file
End Sub
```

'Procedura l'aggiornamento delle variabili globali in modo che i risultati della partita più antica

'vadano sostituiti con i risultati della partita immediatamente più recente

```
Private Sub aggiornaScore()
    Dim i As Integer
    For i = 0 To 3
        partiteGiocate(i) = partiteGiocate(i + 1)
        partiteVinte(i) = partiteVinte(i + 1)
        data(i) = data(i + 1)
    Next
    partiteGiocate(4) = parGiocate
    partiteVinte(4) = parVinte
    data(4) = Date & " " & Time
    lbScore.Caption = "Partite vinte " & parVinte & " su " & parGiocate
    frmScore.Form_Load
End Sub
```

*'Procedura per la scrittura del file score con il contenuto delle variabili del modulo ,di parGiocate
'e di parVinte*

```
Public Sub salvascor(i)
  Open ".\score.txt" For Output As #1 ' Apre il file per l'output.
  For i = 0 To 4
    Write #1, partiteGiocate(i), partiteVinte(i), data(i) ' Scrive il testo nel file.
  Next
  Write #1, parGiocate, parVinte
  'Notare l'uso dell'istruzione Write # anziché Print #  
'per scrivere i dati nei file. L'istruzione Write # garantisce la corretta  
'delimitazione di ogni campo di dati.
  Close #1 ' Chiude il file.
End Sub
```

Impostazione delle proprietà degli oggetti del form: frmAbout

- **(CommandButton)** cmdOK.Caption = "OK"
- **(Image)** frmAbout.picture=(l'immagine bitmap che avete visto di sfondo nel form a pag.3).
- **(line)** Line1.BorderColor= &H00000000&

Scrittura del codice

'Procedura di risposta all'evento:click su cmdOK

```
Private Sub cmdOK_Click()
  Unload Me ' chiude il form
End Sub
```

'Procedura di risposta all'evento:Caricamento della form frmAbout

```
Private Sub Form_Load()
  'Vengono settate le proprietà caption di: frmAbout, lblVersion , lblTitle e  
' lblDescription , utilizzando l'oggetto app che contiene gli attributi del progetto a cui le due  
'form appartengono, descritto  
'a pag. 4.
  Me.Caption = "Informazioni su " & App.Title
  lblVersion.Caption = "Versione " & App.Major & "." & App.Minor & "." & App.Revision
  lblTitle.Caption = App.Title
  lblDescription.Caption = App.Comments
End Sub
```

Impostazione delle proprietà degli oggetti del form: frmScore

- **(label)** lblVociScore.Caption="Partite Vinte Partite Giocate Data"
- **(label)** lblVociScore.BackStyle=0 - trasparente
- **(label)** lblPartiteVinte(0 to 5).BackStyle=0 - trasparente
- **(label)** lblPartiteGiocate (0 to 5).BackStyle=0 - trasparente
- **(label)** lblData(0 to 5).BackStyle=0 - trasparente

Scrittura del codice

'Procedura di risposta all'evento:Caricamento della form frmScore

```
Public Sub Form_Load()
  Dim i As Integer
  For i = 0 To 4
    lblPartiteVinte(i).Caption = partiteVinte(i)
    lblPartiteGiocate(i).Caption = partiteGiocate(i)
    lblData(i).Caption = data(i)
  Next
End Sub
```

Impostazione delle proprietà degli oggetti del form: frmOpzioni

- **CommandButton** cmdScore.caption= "Cancella Registro Score"
- **CommandButton** cmdImage.caption= "cmdImage"
- **CommandButton** cmdChiudi.caption= "cmdChiudi"

Scrittura del codice

'Procedura di risposta all'evento:click su cmdChiudi

```
Private Sub cmdChiudi_Click()  
    Unload Me  
End Sub
```

'Procedura di risposta all'evento:click su cmdImage

```
Private Sub cmdImage_Click()  
    'Imposta CancelError su True.  
    CommonDialog1.CancelError = True  
    'Imposta i flag.  
    CommonDialog1.Flags = cdOFNHideReadOnly  
    'Imposta i filtri.  
    CommonDialog1.Filter = "Tutti i immagine(*.bmp)"  
    'Specifica il filtro predefinito.  
    CommonDialog1.FilterIndex = 2  
    'Visualizza la finestra di dialogo Apri.  
    On Error GoTo noImage          'in caso Annulla effettua un salto incondizionato a  
                                     'noImage  
  
    CommonDialog1.ShowOpen  
    'Visualizza il nome del file selezionato.  
    frmTris.Picture = LoadPicture(CommonDialog1.FileName)  
noImage:  
End Sub
```

'Procedura di risposta all'evento:click su cmdScore

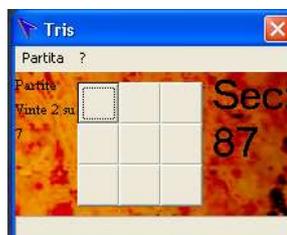
```
Private Sub cmdScore_Click()  
    Dim i As Integer  
    If (MsgBox("Vuoi Cancellare l'archivio score?", 4)) = vbYes Then  
        Open ".score.txt" For Output As #1 'Apri il file per l'output.  
        For i = 0 To 17  
            Write #1, "0"  
        Next  
        Close #1 'Chiude il file.  
        frmTris.leggiScore  
        frmScore.Form_Load  
    End If  
End Sub
```

Creazione dell'applicazione eseguibile

Dopo aver completato il progetto e controllato il funzionamento delle sue componenti, ho finalmente creato il file eseguibile, selezionando dal menù File del VB la voce Crea Tris.exe. In basso viene riportato lo **screenshot** di una partita di prova.

Commento

L'applicazione creata non presentava alcun errore.



Partite Vinte	Partite Giocate	Data
1	3	30/11/2004 16:50:43
1	4	30/12/2004 16:50:51
2	0	30/12/2004 16:50:59
2	6	30/12/2004 16:51:11
2	7	30/12/2004 16:51:17