

La prima applicazione in visual basic complessa:

Campo Minato

Il problema

Si vuole creare una applicazione VB 6.0 che sia un clone del gioco Campo Minato presente di default in quasi tutte le versioni di Windows.

Analisi preliminare

Nell'applicazione verranno utilizzati due nuovi controlli (rispetto alla esperienza precedente): la **Barra dei menù** ed un controllo importato, di casa Microsoft, chiamato **TextToSpeech**, inserito "sperimentalmente", di mia iniziativa, per cercare di abbellire l'applicazione e provare le funzionalità di tale oggetto.

I **vincoli** del programma saranno i seguenti:

- La tabella del campo minato avrà dimensione fissa 10 x10;
- Alcune funzionalità presenti programma originale, non essenziali per il funzionamento generare dell'applicazione, come, ad esempio, la possibilità di settare alcune impostazioni del gioco (numero mine, suono ON-OFF, segno (?)), oppure di visualizzare la Classifica dei Record, verranno tralasciate.

Dati di Input

Mossa Utente (indice di una cella della tabella del campo minato)

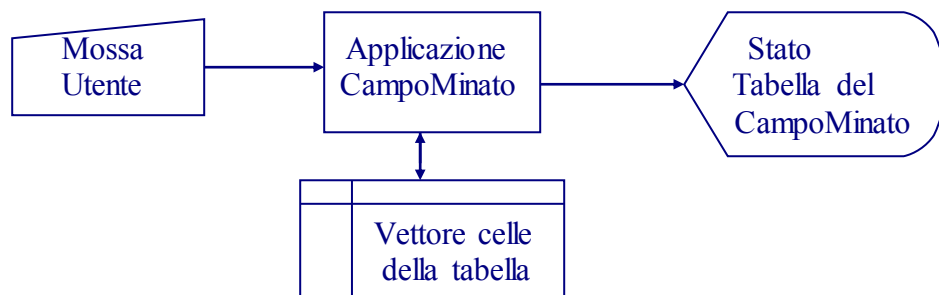
Dati di Output

Stato del Gioco (Gioco On-OFF) e tabella grafica del campo minato, indicante per ogni cella puntata dall'utente, il numero delle mine presenti nelle 8 celle con cui confina la cella puntata, se tale cella puntata non contiene una mina, oppure la soluzione del gioco, se la contiene.

Analisi del problema

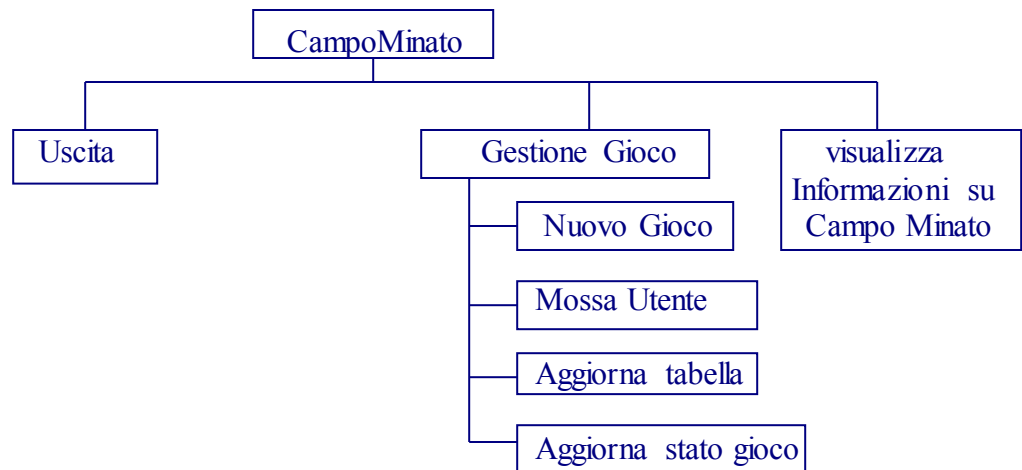
L'applicazione da creare utilizzerà le seguenti risorse di sistema:

System resources chart



E sarà basata sul funzionigramma illustrato nella pagina seguente.

Modello FH

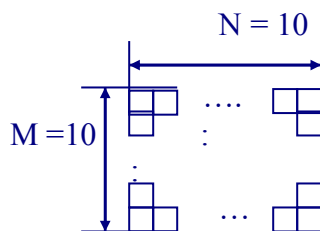


La progettazione

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 Campo Minato, sarà formata da due **form**: un form principale, che sarà l'oggetto di avvio, chiamato **f_Campo_Minato**; e un form secondario, chiamato **frmAbout**, che verrà richiamato dal form principale.

Nel form principale verrà riprodotto, utilizzando i controlli CommandButton, il seguente schema:



Tali bottoni apparterranno ad un vettore di bottoni di dimensione $N \times M$, chiamato **CBCella** dato che ogni oggetto del vettore rappresenterà una cella della tabella del campo minato.

All'interno del form verranno inseriti, inoltre: **la barra dei menù** e l'oggetto **ImgMina** di classe Image. Mentre quest'ultimo servirà per contenere l'immagine bitmap delle mine grafiche da visualizzare sulla superficie dei bottoni, la barra dei menù consentirà all'utente di:

Giocare una nuova partita, selezionando una su tre tipi di difficoltà disponibili, che differenziano l'uno dall'altro, per semplicità, soltanto per il numero di mine da nascondere nella griglia, uscire dall'applicazione e visualizzare una nuova form con le informazioni sul programma, la form frmAbout.

Nel form verranno inseriti anche oggetti di importanza secondaria, quali: **voce**, di classe **TextToSpeech**, che convertirà delle stringhe di testo in formato audio (in altre parole, renderà il mio clone di Campo Minato, a differenza del gioco originale, parlante!) e **timer**, di classe **timer**, che conterà i secondi impiegati dal giocatore durante una partita.

Il funzionamento basilare dell'applicazione sarà il seguente:

- Quando viene caricata l'applicazione, oppure seleziona la voce *nuova partita* dal menù, verrà richiamata la **Sub Nuova partita**, che richiederà come parametro un intero, rappresentante il numero delle mine da disporre nella griglia costituita dal vettore **CBcella**. Essa si occuperà di annullare tutte le modifiche applicate alle proprietà degli oggetti **CBcella** nelle partite precedenti e di disporre in esse, casualmente, le mine, cioè, memorizzare nell'attributo **tag** la stringa "mina".

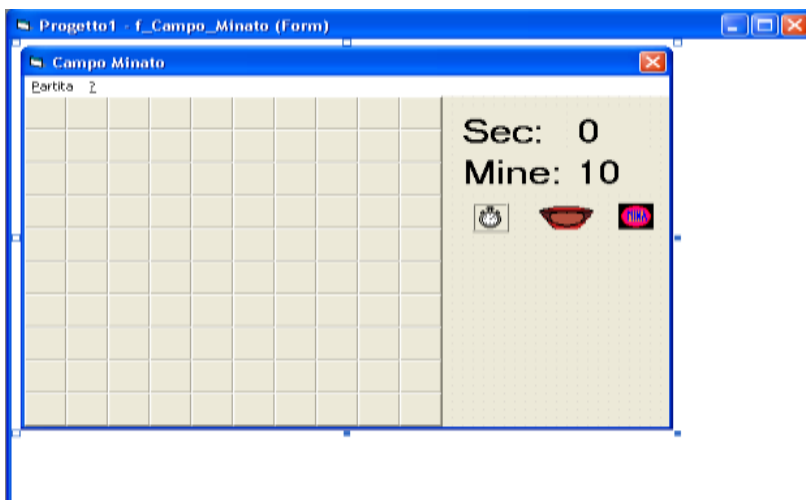
- In stato di gioco attivo , tramite la procedura di risposta all'evento click su **CBcella(index)** , verranno acquisite le mosse dell'utente , aggiornando a seguito di ciascuna mossa: la tabella di bottoni e lo stato del gioco.

La realizzazione dell' applicazione

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

- **CampoMinato**(Campo_Minato.vbp)
 - **form** f_Campo_Minato
 - **barra dei menù**
 - **Image** ImgMina
 - **Timer** Timer
 - **CommandButton** CBCella
 - **TextToSpeech** voce
 - **label** LabSec
 - **label** LabNunSec
 - **label** LabMine
 - **label** LabNumMine
 - **form** frmAbout
 - **label** lbITitle
 - **label** lbIVersion
 - **label** lbIDescription
 - **CommandButton** cmdOK
 - **Image** ImageSfondo
 - **line** Line1

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



'Procedura di risposta all'evento:click su CBcella

```
Private Sub CBcella_Click(Index As Integer)
    If CelleTot = 0 Or CBCella(Index).Tag = "acqua" Then Exit Sub           'verifica GameOver
                                                                              'o se CBcella(Index)è già stata cliccata
                                                                              'verifica se CBcella(Index) 'contiene una mina

    If (CBCella(Index).Tag = "mina") Then
        f_Campo_Minato.Caption = "fine partita"
        Timer.Enabled = False
        soluzione                               'stampa soluzione
        CelleTot = 0                             'CelleTot= game over
        voce.Speak ("Hai perso!")

    Else                                         ' caso: CBCella(Index) non contiene una mina
        CBCella(Index).Tag = "acqua"
        CBCella(Index).Caption = CalcoloMine(Index)
        CelleTot = CelleTot - 1
        If CelleTot = 0 Then                    'controllo vittoria
            f_Campo_Minato.Caption = "You win"
            soluzione
            Timer.Enabled = False              'blocco del il tempo
            voce.Speak ("Congratulazioni , hai vinto!")
            Exit Sub
        End If
    End If
End Sub
```

' Procedura per il Calcolo del numero delle mine situate nelle celle confinanti con la cella di posto
' index

```
Public Function CalcoloMine(Index As Integer) As Integer
    'dei dintorni di CBcella(index)
    Dim mine As Integer, tmp As Integer, x As Integer
    On Error GoTo err 'Attiva la routine di gestione degli errori. Se si verifica un errore si effettuerà
                                                                ' un salto incondizionato All'etichetta err

    '----- Ricerca delle mine -----
    tmp = Index + 1
    x = Index Mod 10
    If (Not (x = 9)) And (CBCella(tmp).Tag = "mina") Then
        mine = mine + 1
    End If

    tmp = Index - 1
    x = Index Mod 10
    If (Not (x = 0)) And (CBCella(tmp).Tag = "mina") Then
        mine = mine + 1
    End If

    tmp = Index + 10
    If (CBCella(tmp).Tag = "mina") Then
        mine = mine + 1
    End If

    tmp = Index - 10
    If (CBCella(tmp).Tag = "mina") Then
        mine = mine + 1
    End If
```

```

tmp = Index + 11
x = Index Mod 10
If (Not (x = 9)) And (CBCella(tmp).Tag = "mina") Then
    mine = mine + 1
End If

```

```

tmp = Index - 11
x = Index Mod 10
If (Not (x = 0)) And (CBCella(tmp).Tag = "mina") Then
    mine = mine + 1
End If

```

```

tmp = Index + 9
x = Index Mod 10
If (Not (x = 0)) And (CBCella(tmp).Tag = "mina") Then
    mine = mine + 1
End If

```

```

tmp = Index - 9
x = Index Mod 10
If (Not (x = 9)) And (CBCella(tmp).Tag = "mina") Then
    mine = mine + 1
End If

```

CalcoloMine = mine *'Restituisce il valore della var mine*

Exit Function *'esce dalla Function.*

err: *'In caso di errore :indice di CBCella non valido ,si salterà qui.*

tmp = Index

Resume *'Riprende l'esecuzione dalla riga di codice in cui si è verificato l'errore.*

End Function

'Procedura per il ripristino delle celle e l'inserimento delle mine

Private Sub Nuova_Partita(mine As Integer)

Dim i As Integer, a As Long

CelleTot = 100 - mine

Do

CBCella(i).Tag = "" *'vengono ripristinate tutte le celle*

CBCella(i).Picture = LoadPicture()

CBCella(i).Caption = ""

i = i + 1

Loop While i < 100

Do *'vengono inserite casualmente le mine*

a = Rnd * 99

If CBCella(a).Tag <> "mina" Then *'controllo se CBCella(a) contiene già una mina*

CBCella(a).Tag = "mina"

mine = mine - 1

End If

Loop While mine > 0

sec = 0

Timer.Enabled = True *'viene fatto partire il tempo di gioco*

f_Campo_Minato.Caption = "Campo Minato" *'viene settato il titolo del form principale*

voce.Speak ("Isola tutte le mine")

End Sub

```

Private Sub soluzione()
    Dim i As Integer
    i = 0
    Do
        If (CBCella(i).Tag = "mina") Then
            CBCella(i).Picture = ImgMina.Picture
        End If
        i = i + 1
    Loop While i < 100
    mineTot = 0
End Sub

```

*'Procedura di risposta all'evento:Timer ,richiamato ,ricordiamo, ad ogni secondo trascorso
'dall'attivazione*

```

Private Sub Timer_Timer()
    LabNunSec.Caption = sec
    sec = sec + 1
End Sub

```

'Procedure di gestione della barra dei menù

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

```

Private Sub mnuPartitaItem_Click(Index As Integer)
    If Index = 4 Then End
End Sub

```

'Procedura di risposta all'evento:click sul sottomenù di NuovaPartita

```

Private Sub mnuPartitaNuova_PartitaItem_Click(Index As Integer)
    Nuova_Partita ((Index + 1) * 10)
    LabNumMine.Caption = (Index + 1) * 10
    Dim i As Integer
    i = 0
    Do
        mnuPartitaNuova_PartitaItem(i).Checked = False
        i = i + 1
    Loop While i < 3
    mnuPartitaNuova_PartitaItem(Index).Checked = True
End Sub

```

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

```

Private Sub mnuhelpItem_Click(Index As Integer)
    If Index = 2 Then frmAbout.Show
End Sub

```

Impostazione delle proprietà degli oggetti del form: frmAbout

- **(label)** lblDescription.Caption = "Applicazione dimostrativa delle potenzialità di Visual Basic. Ideazione , progettazione e sviluppo di
>- Caruso Davide -<"
- **(CommandButton)** cmdOK.Caption = "OK"
- **(Image)** ImageSfondo.picture=(17mmagine bitmap che avete visto di sfondo al form a pag.3). Successivamente a tale impostazione applicare l'opzione:Porta in secondo piano.
- **(line)** Line1.BorderColor= &H00000000&

Scrittura del codice

'Procedura di risposta all'evento:click su cmdOK

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

' chiude il form

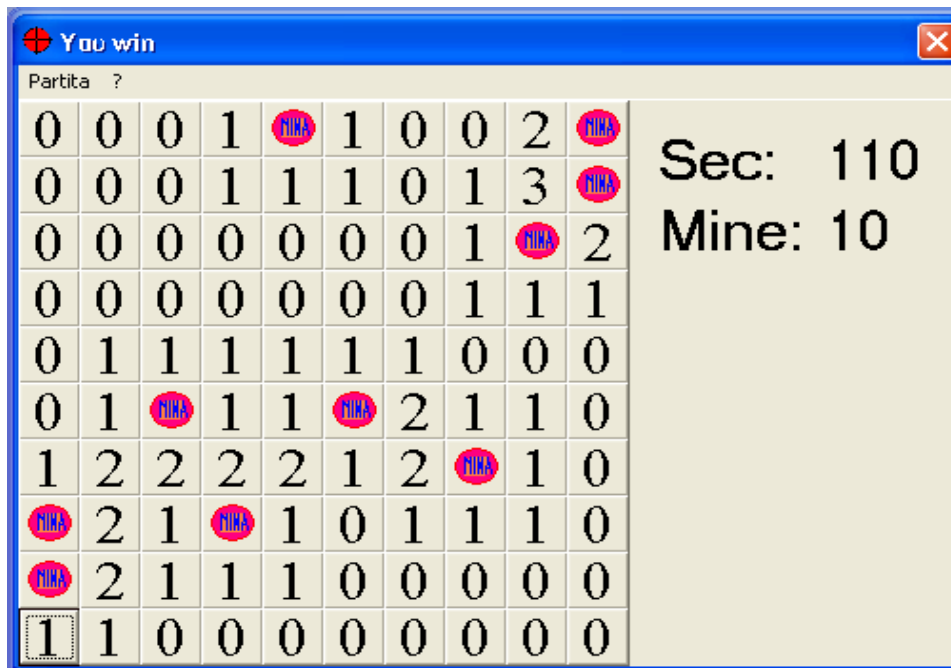
'Procedura di risposta all'evento:Caricamento della form frmAbout

```
Private Sub Form_Load()  
    'Vengono settate le proprietà caption del form in esame,di lblVersion e lblTitle , utilizzando  
    'l'oggetto app che contiene gli attributi del progetto a cui le due form appartengono, decritto  
    'a pag. 4.  
    Me.Caption = "Informazioni su " & App.Title  
    lblVersion.Caption = "Versione " & App.Major & "." & App.Minor & "." & App.Revision  
    lblTitle.Caption = App.Title  
    f_Campo_Minato.voce.Speak ("Caruso Davide presenta :")  
    f_Campo_Minato.voce.Speak ("- Campo Minato versione 1.0")  
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 CampoMinato.exe.

In basso viene riportato lo **screenshot** di una partita di prova.



Commento

L'applicazione creata non presentava alcun errore . Con un maggior tempo a disposizione , sarebbe stato possibile superare i vincoli imposti inizialmente per aggiungere altre funzionalità secondarie. Tutto sommato , però , direi che stato ottenuto un buon risultato.

In questa esperienza ho imparato che :

In Visual Basic è disponibile un'ampia gamma di strumenti per la gestione e la personalizzazione delle applicazioni. Nell'applicazione creata ne ho utilizzati solo alcuni:quelli più basilari ,come i CommandButton o gli Image ,e quelli speciali come i **TextToSpeech**.

Il loro utilizzo è spesso molto semplice da effettuare e abbastanza intuitivo ; in caso contrario è sempre possibile utilizzare la ,ben fatta, guida on line del VB.