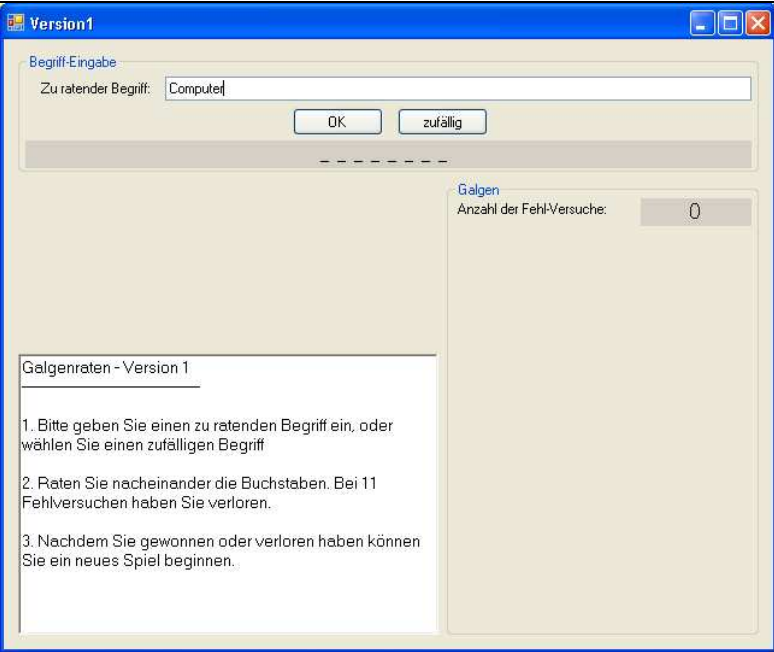



PROJEKT GALGENRATEN

Wir wollen uns an ein etwas größeres Projekt wagen: Das Projekt Galgenraten oder auch Hangman. Bei diesem Spiel geht es darum, einen zuvor festgelegten Begriff buchstabenweise zu erraten. Immer wenn der Buchstabe im Wort enthalten ist, wird dieser angezeigt. Ist der Buchstabe dagegen nicht im Wort enthalten, so wird ein Galgen Schritt für Schritt gezeichnet. Das Spiel endet, wenn das Wort komplett erraten wurde oder der Galgen vollständig gezeichnet ist.

Ich stelle mir das Layout des Projekts wie folgt vor:

<p>Am Anfang kann man den zu ratenden Begriff eingeben. Die Striche für das zu ratende Wort werden automatisch erzeugt.</p> <p>Die Anzahl der Fehlversuche beträgt zu Anfang 0 und der Galgen ist noch nicht gezeichnet.</p> <p>Zusätzlich ist eine kleine Spielregel (links unten) hilfreich für den Anwender des Programms.</p> <p>Bestätigt man das zu ratende Wort mit OK, so beginnt das Spiel.</p>	
<p>Jetzt kann man nacheinander Buchstaben raten. Ist der Buchstabe im zu ratenden Wort enthalten, so wird der Strich durch den zugehörigen Buchstaben ersetzt. Ist der Buchstabe dagegen nicht enthalten, so wird die nächste Stufe des Galgens gezeichnet und die Anzahl der Fehlversuche wird um 1 erhöht.</p> <p>Die bereits geratenen Buchstaben werden ebenfalls angezeigt.</p>	

Aufgabe 1: Entwirf in Visual Basic das Layout des Projekts. Lege jeweils eine Groupbox für die Begriff-Eingabe, für den Rate-Teil und für die Galgen-Ausgabe an. Kümmere dich zunächst nicht um die Galgen-Grafik, du kannst später passende Bilder von meiner Homepage herunterladen.

Fangen wir zunächst mit dem ersten Problem an: Man gibt einen Begriff ein und automatisch wird das zu ratende Wort (_ _ _ _ ...) erstellt.

Wir werden zum einen das vom Benutzer eingegebene Wort, also die Lösung, in einem String speichern. In unserem Fall würde "C O M P U T E R" gespeichert werden.

Zum anderen werden wir das zu ratende Wort (hier: " _ _ _ _ _ _ _ _ ") intern speichern.

Dafür benötigen wir zwei Variablen vom Typ String, welche am Anfang der Klasse `MainForm` angelegt werden. Zudem benötigen wir eine Variable vom Typ Integer für die Anzahl der Fehlversuche:

```
Public Partial Class MainForm
    Dim loesung As String
    Dim geraten As String
    Dim versuche As Integer
```

Immer wenn der Anwender nun das Textfeld für den zu ratenden Begriff verändert soll folgendes passieren:

Zur Erstellung des Strings `loesung` durchlaufen wir alle Buchstaben des Textfeldes und hängen die entsprechenden Großbuchstaben getrennt von einem Leerzeichen hintereinander.

Zur Erstellung des Strings `geraten` fügen wir für jeden Buchstaben des Textfeldes einen Unterstrich gefolgt von einem Leerzeichen (" _ ") an den String `geraten` an.

Aufgabe 2: Klicke doppelt auf das Textfeld zur Begriff-Eingabe. Folgende Methode öffnet sich:

```
Sub TWortTextChanged(ByVal ... Args)

End Sub
```

Um alle Buchstaben des zu ratenden Wortes zu durchlaufen bietet sich die For-Schleife an:

```
Dim wort As String = tWort.Text
For i = 0 To wort.Length-1
    ...
Next
```

Möchtest du auf einen Buchstaben des zu ratenden Wortes zurückgreifen, so hilft dir der Befehl `wort.Chars(i)`. Damit bekommst du den i-ten Buchstaben des Strings `wort`:

```
loesung = loesung + wort.Chars(i) + " "
geraten = geraten + "_ "
```

Zum Schluss muss noch alles in Großbuchstaben gesetzt werden. Dazu bietet sich der folgende Befehl an:

```
loesung = loesung.ToUpper
```

Vervollständige nun die Methode, so dass bei jeder Änderung des zu ratenden Begriffs die Anzeige aktualisiert wird.

Schön wäre es, wenn Leerzeichen nicht als " _ " angezeigt würden. Ändere dein Programm entsprechend.

So, da wäre das erste große Problem doch schon mal geschafft. Kommen wir nun zum nächsten Problem, dem Buchstabenraten.

Wenn der Benutzer das zu ratende Wort mit OK bestätigt sollte die Groupbox für den Rate-Teil sichtbar gemacht werden. Hier kann der Anwender nun Buchstaben eingeben, die dann in dem angezeigten String `geraten` sichtbar werden.

Aufgabe 3: Klicke doppelt auf den Button "raten". Die folgende Methode öffnet sich:

```
Sub BRatenClick(ByVal sender As Object, ByVal e As EventArgs)

End Sub
```

Nun musst du in dieser Methode überprüfen, ob der eingegebene Buchstabe irgendwo im String `loesung` vorkommt. Hierfür hilft dir wieder die For-Schleife.

Innerhalb dieser Schleife muss nun immer dann, wenn der Buchstabe mit einem Buchstaben des Lösungswortes übereinstimmt, in dem String `geraten` der entsprechende Strich durch den Buchstaben ersetzt werden. Dies kannst du wie folgt machen:

```
Dim c As Char
c = tBuchstabe.text.ToUpper.chars(0)

If c = loesung.Chars(i) Then
    geraten = geraten.Substring(0,i) + c + geraten.Substring(i+1)
End If
```

`geraten.Substring(0,i)` bedeutet, dass vom String `geraten` nur die Buchstaben von 0 bis `i-1` genommen werden. Danach wird der Buchstabe `c` und dann der Rest des bisher geratenen Wortes angehängt.

Wenn der Buchstabe allerdings nicht in dem Wort enthalten war, so muss die Anzahl der Fehlversuche erhöht werden (Variable `versuche`). Um festzustellen, ob der Buchstabe nicht enthalten war, hilft dir die Funktion:

```
If loesung.IndexOf(c) = -1 Then
    versuche = versuche + 1
    ...
    'Galgen erweitern ... kommt später
End If
```

`loesung.IndexOf(c) = -1` bedeutet so viel wie: Suche im String `loesung` nach dem Buchstaben `c`. Ist der Buchstabe enthalten, so wird die Position des Buchstabens zurückgegeben. Ist der Buchstabe nicht enthalten, so wird `-1` zurückgegeben. Und genau das interessiert uns an dieser Stelle, da in diesem Fall die Anzahl der Versuche um 1 erhöht werden muss.

Erweitere nun noch die Methode so, dass die Anzahl der Fehlversuche und das bisher geratene Wort stets auf der Oberfläche aktualisiert wird.

Ergänze die Methode um eine Ausgabe der bisher geratenen Buchstaben. Diese sollen kommasetrennt hintereinander geschrieben werden.

Im Prinzip ist unser Spiel nun spielbereit. Allerdings wäre es schön, wenn man auch tatsächlich einen Galgen sehen könnte, der bei jedem Fehlversuch vervollständigt wird. Zunächst brauchen wir dafür 11 `PictureBoxen`, die wir mit den Galgenbildern besetzen (Eigenschaft `Image`) und anfänglich auf unsichtbar setzen.

Dann machen wir uns eine neue Methode, die je nach Fehlversuchsanzahl (`versuche`) die passende Galgen-`PictureBox` sichtbar macht:

```
Sub galgenAnzeigen
    lVersuche.Text = versuche
    If versuche = 1 Then
        galgen1.Visible = True
    ElseIf versuche = 2 Then
        galgen1.Visible = false
        galgen2.Visible = true
    End If
End Sub
```

```
ElseIf versuche = 3 Then
```

```
...
```

Der Aufruf dieser Methode geschieht dann bei der Aktualisierung der Anzeige.

Aufgabe 4: Erweitere dein Projekt entsprechend, so dass bei jedem Fehlversuch der Galgen erweitert wird.

Nun fehlt noch, dass unser Spiel automatisch endet, sobald das Wort erraten wurde bzw. sobald die Anzahl der Fehlversuche 11 beträgt. Letztes solltest du eigentlich hinbekommen. Um festzustellen, ob das Wort vollständig erraten wurde, hilft uns die Methode `IndexOf(...)`. Ein Wort ist nämlich dann erraten, wenn das Zeichen "_" im String geraten nicht mehr vorkommt. Siehe dir dazu nochmals die Erläuterungen in Aufgabe 3 an.

Aufgabe 5: Erweitere dein Projekt dahingehend, dass das Spiel automatisch neu startet, sobald das zu ratende Wort erraten wurde oder der Galgen vollständig gezeichnet wurde. Der Galgen sollte auf den Ursprungszustand zurückgesetzt werden, die Anzahl der Fehlversuche ebenfalls. Die Groupbox für den Rate-Teil sollte unsichtbar werden und das Textfeld für die Begriff-Eingabe muss sichtbar gemacht werden.

Im Layout hast du vielleicht schon gesehen, dass man als Anwender auch die Möglichkeit haben soll, einen zu ratenden Begriff vom Computer vorgeben zu lassen. Auch das kannst du in deinem Projekt realisieren.

Aufgabe 6: Analysiere die folgende Button-Methode:

```
Sub BZufallClick(ByVal sender As Object, ByVal e As EventArgs)
    dim woerter() as String = {"Hund", "Katze", "Maus", "Auto", _
        "Pferd", "Computer", "Haus", "Ei", "Karl der Grosse", _
        "Gymnasium Odenthal"}
    lEingabe.Visible = False
    tWort.Visible = False

    tWort.Text = woerter(rnd*(woerter.Length-1))

    groupBoxRaten.Visible = True
End Sub
```

Erweitere dein Projekt entsprechend und füge weitere Begriffe ein, die sich der Anwender vom Computer vorgeben lassen kann.

So, nun ist es an der Zeit, dass du dein Projekt nach deinem Geschmack veränderst. Wenn du z. B. einen schöneren Galgen haben möchtest, dann kannst du dich nun daran begeben. Soll dein Projekt die Möglichkeit bieten, dass man das Spiel im Duell spielen kann – auch kein Problem. Schön wäre es auch, wenn der Anwender das ganze Wort auf einmal raten könnte, wenn er meint, die Lösung zu kennen. Im Moment muss er noch alle Buchstaben einzeln raten.

Ich glaube, du hast selber auch noch viele Ideen, die du hier verwirklichen kannst. Viel Spaß!