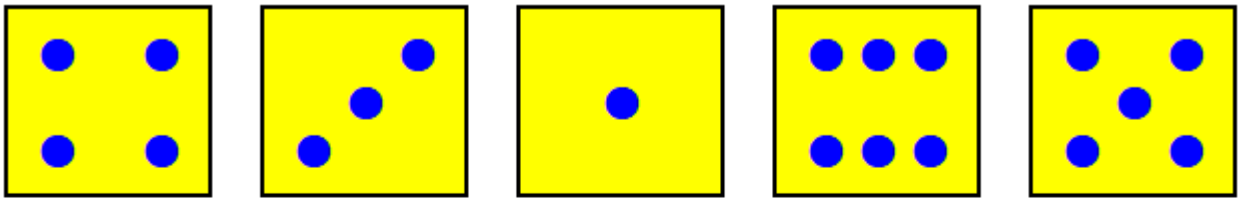
Übungseinheit: # 6

WÜRFELN



Entwicklungsumgebung: Excel 2007

Kursunterlagen erstell von: TECHNISCHES BÜRO MITSCH

Ing. Harald Mitsch
Josefgasse 6
A 3380 Pöchlarn

Ihr IT-Betreuer in Pöchlarn
seit 1990 - 0676/588 09 16
http://www.tb-mitsch.at

letzte Aktualisierung: 11. Jänner 2019

INHALTSVERZEICHNIS

1.) Allgemeines

Bei vielen Spielen benötigt man einen oder mehrere Würfel. In dieser Übungseinheit wollen wir daher das Würfeln simulieren. Zuerst erstellen wir einen einfachen Würfel. Danach erweitern wir das Programm um gleichzeitig 5 Würfel zu steuern. Am Ende versuchen wir noch einen Würfel mit Figuren (Pokerwürfel) zu erstellen. Dafür verwenden wir Shapes (Formen, Grafiken, Bilder, …). Und zu guter letzt würfeln wir noch mit Stäben.

2.) Spielregeln

Dazu gibt es nicht viel zu sagen. Ein Würfel zeigt in der Praxis immer zufällig eine von 6 Seiten. Würfelt man allerdings mit dem Computer, dann kann man auch mehr als 6 Würfelflächen erlauben. Das werden wir beim programmieren zusätzlich berücksichtigen.

3.) Gestalten des Würfels

Zuerst zeichnen wir in einem Tabellenblatt, welches wir logischerweise mit „Wuerfel“ benennen, einen Würfel. Zum Testen legen wir uns auch eine Schaltfläche an, welcher wir später das Makro zuweisen, welches das Würfeln simuliert. Der Entwurf könnte so aussehen:

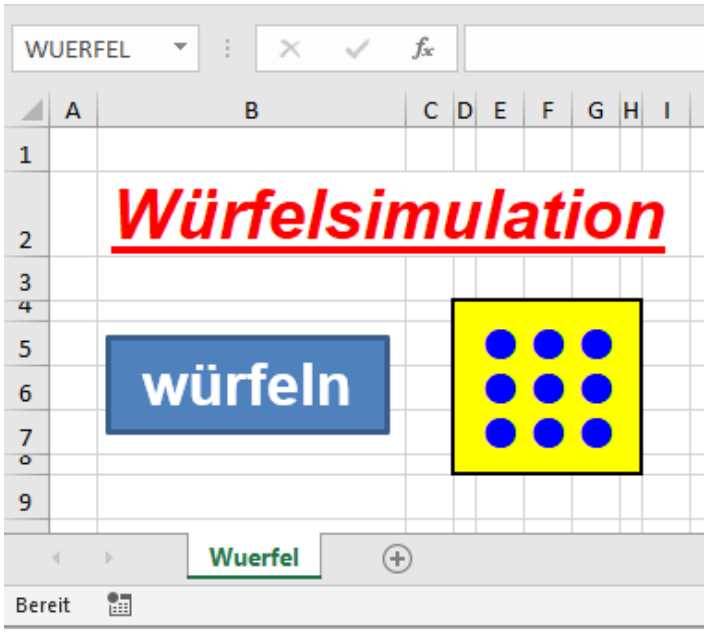
Die Grundfläche des Würfels soll 5x5 Zellen betragen. Diesen Zellenbereich benennen wir mit "WUERFEL" (Namensmanager).

Als Schriftart legen wir "Wingdings" mit der Schriftgröße 20 in Fettschrift fest.

Die Punkte im inneren 3x3 Bereich werden durch ein kleines l (L) (ASC=108) dargestellt.

Spaltenbreite Umrandung: 1
Spaltenbreite für Punkte: 3
Zeilenhöhe Umrandung: 8,5
Zeilenhöhe für Punkte: 18

Hintergrundfarbe und Schriftfarbe für den Bereich "WUERFEL" z.B. mit gelb festlegen, dann die Schriftfarbe für die Punkte mit blau.



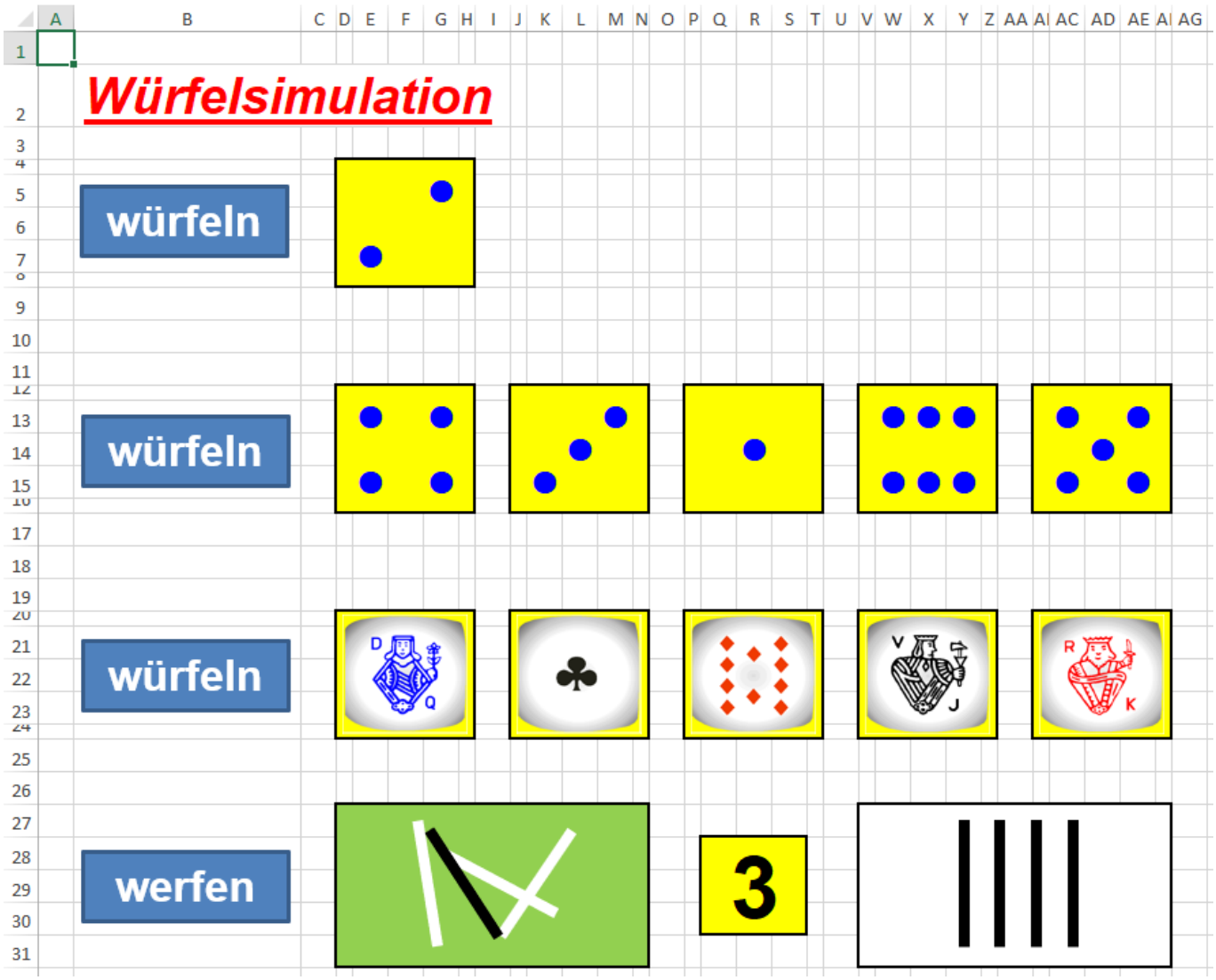
4.) Programmieren der Würfel

4.1.) Vorbereitungen

Zuerst legen wir uns einen neuen Ordner an in welchem wir alle für das Würfeln benötigte Dateien ablegen. In diesen Ordner legen wir uns dann eine neue Excel-Datei an und speichern sie z.B. unter dem Namen: **Wuerfeln.xlsm**. In dieser Datei werden wir das Würfeln simulieren. Am Ende dieser Übung soll ein einziges Modul (z.B.: Wuerfeln) vorhanden sein, welches wir abschließend in die Mustervorlage einkopieren werden, damit die Würfelfunktionen auch für andere Spiele verfügbar sind.

4.2.) Gestaltung des Spielfeldes

Gleich vorab wie das Tabellenblatt am Ende aussehen könnte (nur um einen Eindruck zu bekommen).



4.3.) Programmierung des einfachen Würfels

4.3.1.) Globale Konstanten und Variablen

Zuerst legen wir uns ein neues Modul namens **Wuerfeln** an und definieren folgende Funktione, Konstanten und Variablen:

Zuerst deklarieren wir die Sleep-Funktion, welche wir benötigen um die Drehbewegungen des Würfels optisch schöner gestalten zu können.

```
Option Explicit                                     'Alle Variablen muessen
                                                    'explizit angegeben werden.
'-------------------------------------------------------------------------------/
'-  Deklarierte API-Funktionen fuer 64 Bit Office Versionen                     -/
'-------------------------------------------------------------------------------/
#If VBA7 Then                                       'Fuer Sleep-Funktion 64 Bit Excel:
  Public Declare PtrSafe Sub Sleep Lib "kernel32" (ByVal dwMilliseconds As LongPtr)
'-------------------------------------------------------------------------------/
'-  Deklarierte API-Funktionen fuer 32 Bit Office Versionen                     -/
'-------------------------------------------------------------------------------/
#Else                                               'Fuer Sleep-Funktion 32 Bit Excel:
  Public Declare Sub Sleep Lib "kernel32" (ByVal dwMilliseconds As Long)
#End If
```

Dann definieren wir noch einige Konstanten und Variablen welche wir später benötigen werden.

```
'-------------------------------------------------------------------------------/
'-  Definitionen fuer die Wuerfelsimulation                                     -/
'-------------------------------------------------------------------------------/
Global Const TABELLENBLATT = "Wuerfel"     'Tabellenblatt fuer Spielfeld.
Global Const MAX_AUGEN = 6    'oder 9      'Maximale Anzahl der Wuerfelaugen.
Global Const ASC_AUGE = 108                'Acs-Code fuer Wuerfelauge (108=kleines L).
Global Const MIN_WUERFE = 30               'Minimale Anzahl der Wuerfeldrehungen.
Global Const MAX_WUERFE = 50               'Maximale Anzahl der Wuerfeldrehungen.
Global Const DAUER_WUERFE = 30             'Anzeigedauer nach einer Wuerfeldrehung.

Global oTabellenblattWuerfel   As Object   'Variable fuer Tabellenblatt.

Global Const SCHWARZ = 0                   'Farbe schwarz = RGB(0, 0, 0)
Global Const WEISS = 16777215              'Farbe weiss = RGB(255, 255, 255)
Global Const DUNKELGELB = 62207            'Farbe dunkelgelb = RGB(255, 242, 0)

Global Const FARBE_SW = SCHWARZ            'Farbe fuer schwarze Stabseite.
Global Const FARBE_WS = WEISS              'Farbe fuer weisse Stabseite.
```

4.3.2.) Makros und Funktionen zum Würfeln

Da wir insgesamt drei Würfelsimulationen programmieren wollen, legen wir uns gleich zu Beginn noch drei Makros an, welche wir den drei "würfeln"-Schaltflächen zuordnen. Die Makos erstellen wir im selben Modul gleich nach den globalen Definitionen:

```
Sub Wuerfeln_1()
  Call Wuerfeln1                                    'Mit einem Wuerfel wuerfeln.
End Sub

Sub Wuerfeln_5()
  Call Wuerfeln5                                    'Mit fuenf Wuerfel wuerfeln.
End Sub

Sub Wuerfeln_P()
  Call WuerfelnP                                    'Mit fuenf Pokerwuerfel wuerfeln.
End Sub

Sub Wuerfeln_S()
  Call WuerfelnS                                    'Mit vier Staeben wuerfeln.
End Sub
```

Um das Programm auch kompilieren zu können legen wir uns vorab noch folgende Funktionen an:

```
Public Function Wuerfeln1() As Boolean
On Error GoTo Fehler                          'Bei Fehler zur Fehlermarke.
Wuerfeln1 = True                              'Returnwert = alles OK setzen.

                                              'Fehlerbehandlungsroutine:
Beenden:                                      '========================
  Application.Cursor = xlDefault              'Sanduhr ausblenden.
  Exit Function                               'Funktion abbrechen.
Fehler:                                       'Fehlermarke - Fehlerbehandlung:
  Wuerfeln1 = False                           'Returnwert = Fehler setzen.
  MsgBox Err.Description                      'Fehlermeldung ausgeben.
  Resume Beenden                              'Bei Funktionsende weitermachen.
End Function                                  'Funktionsende.


Public Function Wuerfeln5() As Boolean
On Error GoTo Fehler                          'Bei Fehler zur Fehlermarke.
Wuerfeln5 = True                              'Returnwert = alles OK setzen.

                                              'Fehlerbehandlungsroutine:
Beenden:                                      '========================
  Application.Cursor = xlDefault              'Sanduhr ausblenden.
  Exit Function                               'Funktion abbrechen.
Fehler:                                       'Fehlermarke - Fehlerbehandlung:
  Wuerfeln5 = False                           'Returnwert = Fehler setzen.
  MsgBox Err.Description                      'Fehlermeldung ausgeben.
  Resume Beenden                              'Bei Funktionsende weitermachen.
End Function                                  'Funktionsende.


Public Function WuerfelnP() As Boolean
On Error GoTo Fehler                          'Bei Fehler zur Fehlermarke.
WuerfelnP = True                              'Returnwert = alles OK setzen.

                                              'Fehlerbehandlungsroutine:
Beenden:                                      '========================
  Application.Cursor = xlDefault              'Sanduhr ausblenden.
  Exit Function                               'Funktion abbrechen.
Fehler:                                       'Fehlermarke - Fehlerbehandlung:
  WuerfelnP = False                           'Returnwert = Fehler setzen.
  MsgBox Err.Description                      'Fehlermeldung ausgeben.
  Resume Beenden                              'Bei Funktionsende weitermachen.
End Function                                  'Funktionsende.


Public Function WuerfelnS() As Boolean
On Error GoTo Fehler                          'Bei Fehler zur Fehlermarke.
WuerfelnP = True                              'Returnwert = alles OK setzen.

                                              'Fehlerbehandlungsroutine:
Beenden:                                      '========================
  Application.Cursor = xlDefault              'Sanduhr ausblenden.
  Exit Function                               'Funktion abbrechen.
Fehler:                                       'Fehlermarke - Fehlerbehandlung:
  WuerfelnS = False                           'Returnwert = Fehler setzen.
  MsgBox Err.Description                      'Fehlermeldung ausgeben.
  Resume Beenden                              'Bei Funktionsende weitermachen.
End Function                                  'Funktionsende.
```

4.3.3.) Funktion zur Darstellung einer Seite des Würfels

Zunächst erstellen wir eine Funktion, welche je nach zufällig erzeugtem Wurfergebnis die entsprechenden Augen des Würfels ausgibt. Dabei berücksichtigen wir auch (nur wegen der Optik), daß die Zahlen 2, 3, 6 und 7 in zwei verschiedenen Lagen dargestellt werden können. Welches Bild augezeigt wird erfolgt ebenfalls zufällig.

Damit wir die Funktion für verschiedene Würfel verwenden können, übergeben wir einen Parameter, der die Position des Würfels im Tabellenblatt angibt. Das Ergebnis speichern wir unsichtbar (gleiche Schrift- und Hintergrundfarbe) in der linken oberen Ecke der Würfelposition. Zusätzlich retournieren wir den Zahlenwert des Wurfes in der Funktion:

```
Public Function WuerfelDrehen(stBereich As String) As Integer
'*************************************************************************/
'*      INTERNES UNTERPROGRAMM - INTERNE FUNKTION                        */
'*************************************************************************/
'*----------------------------------------------------------------------*/
'*                                                                       */
'*      FUNKTION:       WuerfelDrehen                                    */
'*                                                                       */
'*      BESCHREIBUNG:   Simuliert das Wuerfeln fuer einen einzigen Wuerfel. */
'*                                                                       */
'*      RETURN          Integer ..... 1 - 6 (9) = gewuerfelte Zahl       */
'*                                    0 = Fehler in Funktion             */
'*                                                                       */
'*      PARAMETER:      String ..... Bereich wo sich Wuerfel befindet    */
'*                                                                       */
'*----------------------------------------------------------------------*/
Dim i                   As Integer                  'Zufaellige Augenzahl des Wuerfel
Dim j                   As Integer                  'Zufaellige Art der Darstellung.
Dim iZeile              As Long                     'Aktuelle Zeilennummer.
Dim iSpalte             As Integer                  'Aktuelle Spaltennummer.

On Error GoTo Fehler                                'Bei Fehler zur Fehlermarke.

iZeile = Range(stBereich).Row + 1                   'Zeile fuer Wuerfel auslesen.
iSpalte = Range(stBereich).Column + 1               'Spalte fuer Wuerfel auslesen.
i = Int(MAX_AUGEN * Rnd() + 1)                      'Zufaellige Augenzahl.
j = Int(2 * Rnd() + 1)                              'Zufaellige Darstellung.

Select Case i                                       'Je nach Augenzahl den
entsprechenden
  Case 1                                            'Wuerfel darstellen.
    oTabellenblattWuerfel.Cells(iZeile + 0, iSpalte + 0) = ""
    oTabellenblattWuerfel.Cells(iZeile + 0, iSpalte + 1) = ""
    oTabellenblattWuerfel.Cells(iZeile + 0, iSpalte + 2) = ""
    oTabellenblattWuerfel.Cells(iZeile + 1, iSpalte + 0) = ""
    oTabellenblattWuerfel.Cells(iZeile + 1, iSpalte + 1) = Chr(ASC_AUGE)
    oTabellenblattWuerfel.Cells(iZeile + 1, iSpalte + 2) = ""
    oTabellenblattWuerfel.Cells(iZeile + 2, iSpalte + 0) = ""
    oTabellenblattWuerfel.Cells(iZeile + 2, iSpalte + 1) = ""
    oTabellenblattWuerfel.Cells(iZeile + 2, iSpalte + 2) = ""

  Case 2
    If j = 1 Then                                   'Je nach Darstellungsart:
      oTabellenblattWuerfel.Cells(iZeile + 0, iSpalte + 0) = ""
      oTabellenblattWuerfel.Cells(iZeile + 0, iSpalte + 1) = ""
      oTabellenblattWuerfel.Cells(iZeile + 0, iSpalte + 2) = Chr(ASC_AUGE)
      oTabellenblattWuerfel.Cells(iZeile + 1, iSpalte + 0) = ""
```

```
      oTabellenblattWuerfel.Cells(iZeile + 1, iSpalte + 1) = ""
      oTabellenblattWuerfel.Cells(iZeile + 1, iSpalte + 2) = ""
      oTabellenblattWuerfel.Cells(iZeile + 2, iSpalte + 0) = Chr(ASC_AUGE)
      oTabellenblattWuerfel.Cells(iZeile + 2, iSpalte + 1) = ""
      oTabellenblattWuerfel.Cells(iZeile + 2, iSpalte + 2) = ""
    Else
      oTabellenblattWuerfel.Cells(iZeile + 0, iSpalte + 0) = Chr(ASC_AUGE)
      oTabellenblattWuerfel.Cells(iZeile + 0, iSpalte + 1) = ""
      oTabellenblattWuerfel.Cells(iZeile + 0, iSpalte + 2) = ""
      oTabellenblattWuerfel.Cells(iZeile + 1, iSpalte + 0) = ""
      oTabellenblattWuerfel.Cells(iZeile + 1, iSpalte + 1) = ""
      oTabellenblattWuerfel.Cells(iZeile + 1, iSpalte + 2) = ""
      oTabellenblattWuerfel.Cells(iZeile + 2, iSpalte + 0) = ""
      oTabellenblattWuerfel.Cells(iZeile + 2, iSpalte + 1) = ""
      oTabellenblattWuerfel.Cells(iZeile + 2, iSpalte + 2) = Chr(ASC_AUGE)
    End If

  Case 3
    If j = 1 Then                                   'Je nach Darstellungsart:
      oTabellenblattWuerfel.Cells(iZeile + 0, iSpalte + 0) = Chr(ASC_AUGE)
      oTabellenblattWuerfel.Cells(iZeile + 0, iSpalte + 1) = ""
      oTabellenblattWuerfel.Cells(iZeile + 0, iSpalte + 2) = ""
      oTabellenblattWuerfel.Cells(iZeile + 1, iSpalte + 0) = ""
      oTabellenblattWuerfel.Cells(iZeile + 1, iSpalte + 1) = Chr(ASC_AUGE)
      oTabellenblattWuerfel.Cells(iZeile + 1, iSpalte + 2) = ""
      oTabellenblattWuerfel.Cells(iZeile + 2, iSpalte + 0) = ""
      oTabellenblattWuerfel.Cells(iZeile + 2, iSpalte + 1) = ""
      oTabellenblattWuerfel.Cells(iZeile + 2, iSpalte + 2) = Chr(ASC_AUGE)
    Else
      oTabellenblattWuerfel.Cells(iZeile + 0, iSpalte + 0) = ""
      oTabellenblattWuerfel.Cells(iZeile + 0, iSpalte + 1) = ""
      oTabellenblattWuerfel.Cells(iZeile + 0, iSpalte + 2) = Chr(ASC_AUGE)
      oTabellenblattWuerfel.Cells(iZeile + 1, iSpalte + 0) = ""
      oTabellenblattWuerfel.Cells(iZeile + 1, iSpalte + 1) = Chr(ASC_AUGE)
      oTabellenblattWuerfel.Cells(iZeile + 1, iSpalte + 2) = ""
      oTabellenblattWuerfel.Cells(iZeile + 2, iSpalte + 0) = Chr(ASC_AUGE)
      oTabellenblattWuerfel.Cells(iZeile + 2, iSpalte + 1) = ""
      oTabellenblattWuerfel.Cells(iZeile + 2, iSpalte + 2) = ""
    End If

  Case 4
    oTabellenblattWuerfel.Cells(iZeile + 0, iSpalte + 0) = Chr(ASC_AUGE)
    oTabellenblattWuerfel.Cells(iZeile + 0, iSpalte + 1) = ""
    oTabellenblattWuerfel.Cells(iZeile + 0, iSpalte + 2) = Chr(ASC_AUGE)
    oTabellenblattWuerfel.Cells(iZeile + 1, iSpalte + 0) = ""
    oTabellenblattWuerfel.Cells(iZeile + 1, iSpalte + 1) = ""
    oTabellenblattWuerfel.Cells(iZeile + 1, iSpalte + 2) = ""
    oTabellenblattWuerfel.Cells(iZeile + 2, iSpalte + 0) = Chr(ASC_AUGE)
    oTabellenblattWuerfel.Cells(iZeile + 2, iSpalte + 1) = ""
    oTabellenblattWuerfel.Cells(iZeile + 2, iSpalte + 2) = Chr(ASC_AUGE)

  Case 5
    oTabellenblattWuerfel.Cells(iZeile + 0, iSpalte + 0) = Chr(ASC_AUGE)
    oTabellenblattWuerfel.Cells(iZeile + 0, iSpalte + 1) = ""
    oTabellenblattWuerfel.Cells(iZeile + 0, iSpalte + 2) = Chr(ASC_AUGE)
    oTabellenblattWuerfel.Cells(iZeile + 1, iSpalte + 0) = ""
    oTabellenblattWuerfel.Cells(iZeile + 1, iSpalte + 1) = Chr(ASC_AUGE)
    oTabellenblattWuerfel.Cells(iZeile + 1, iSpalte + 2) = ""
    oTabellenblattWuerfel.Cells(iZeile + 2, iSpalte + 0) = Chr(ASC_AUGE)
    oTabellenblattWuerfel.Cells(iZeile + 2, iSpalte + 1) = ""
```

```
    oTabellenblattWuerfel.Cells(iZeile + 2, iSpalte + 2) = Chr(ASC_AUGE)

  Case 6
    If j = 1 Then                                   'Je nach Darstellungsart:
      oTabellenblattWuerfel.Cells(iZeile + 0, iSpalte + 0) = Chr(ASC_AUGE)
      oTabellenblattWuerfel.Cells(iZeile + 0, iSpalte + 1) = Chr(ASC_AUGE)
      oTabellenblattWuerfel.Cells(iZeile + 0, iSpalte + 2) = Chr(ASC_AUGE)
      oTabellenblattWuerfel.Cells(iZeile + 1, iSpalte + 0) = ""
      oTabellenblattWuerfel.Cells(iZeile + 1, iSpalte + 1) = ""
      oTabellenblattWuerfel.Cells(iZeile + 1, iSpalte + 2) = ""
      oTabellenblattWuerfel.Cells(iZeile + 2, iSpalte + 0) = Chr(ASC_AUGE)
      oTabellenblattWuerfel.Cells(iZeile + 2, iSpalte + 1) = Chr(ASC_AUGE)
      oTabellenblattWuerfel.Cells(iZeile + 2, iSpalte + 2) = Chr(ASC_AUGE)
    Else
      oTabellenblattWuerfel.Cells(iZeile + 0, iSpalte + 0) = Chr(ASC_AUGE)
      oTabellenblattWuerfel.Cells(iZeile + 0, iSpalte + 1) = ""
      oTabellenblattWuerfel.Cells(iZeile + 0, iSpalte + 2) = Chr(ASC_AUGE)
      oTabellenblattWuerfel.Cells(iZeile + 1, iSpalte + 0) = Chr(ASC_AUGE)
      oTabellenblattWuerfel.Cells(iZeile + 1, iSpalte + 1) = ""
      oTabellenblattWuerfel.Cells(iZeile + 1, iSpalte + 2) = Chr(ASC_AUGE)
      oTabellenblattWuerfel.Cells(iZeile + 2, iSpalte + 0) = Chr(ASC_AUGE)
      oTabellenblattWuerfel.Cells(iZeile + 2, iSpalte + 1) = ""
      oTabellenblattWuerfel.Cells(iZeile + 2, iSpalte + 2) = Chr(ASC_AUGE)
   End If


  Case 7
    If j = 1 Then                                   'Je nach Darstellungsart:
      oTabellenblattWuerfel.Cells(iZeile + 0, iSpalte + 0) = Chr(ASC_AUGE)
      oTabellenblattWuerfel.Cells(iZeile + 0, iSpalte + 1) = Chr(ASC_AUGE)
      oTabellenblattWuerfel.Cells(iZeile + 0, iSpalte + 2) = Chr(ASC_AUGE)
      oTabellenblattWuerfel.Cells(iZeile + 1, iSpalte + 0) = ""
      oTabellenblattWuerfel.Cells(iZeile + 1, iSpalte + 1) = Chr(ASC_AUGE)
      oTabellenblattWuerfel.Cells(iZeile + 1, iSpalte + 2) = ""
      oTabellenblattWuerfel.Cells(iZeile + 2, iSpalte + 0) = Chr(ASC_AUGE)
      oTabellenblattWuerfel.Cells(iZeile + 2, iSpalte + 1) = Chr(ASC_AUGE)
      oTabellenblattWuerfel.Cells(iZeile + 2, iSpalte + 2) = Chr(ASC_AUGE)
    Else
      oTabellenblattWuerfel.Cells(iZeile + 0, iSpalte + 0) = Chr(ASC_AUGE)
      oTabellenblattWuerfel.Cells(iZeile + 0, iSpalte + 1) = ""
      oTabellenblattWuerfel.Cells(iZeile + 0, iSpalte + 2) = Chr(ASC_AUGE)
      oTabellenblattWuerfel.Cells(iZeile + 1, iSpalte + 0) = Chr(ASC_AUGE)
      oTabellenblattWuerfel.Cells(iZeile + 1, iSpalte + 1) = Chr(ASC_AUGE)
      oTabellenblattWuerfel.Cells(iZeile + 1, iSpalte + 2) = Chr(ASC_AUGE)
      oTabellenblattWuerfel.Cells(iZeile + 2, iSpalte + 0) = Chr(ASC_AUGE)
      oTabellenblattWuerfel.Cells(iZeile + 2, iSpalte + 1) = ""
      oTabellenblattWuerfel.Cells(iZeile + 2, iSpalte + 2) = Chr(ASC_AUGE)
   End If

  Case 8
    oTabellenblattWuerfel.Cells(iZeile + 0, iSpalte + 0) = Chr(ASC_AUGE)
    oTabellenblattWuerfel.Cells(iZeile + 0, iSpalte + 1) = Chr(ASC_AUGE)
    oTabellenblattWuerfel.Cells(iZeile + 0, iSpalte + 2) = Chr(ASC_AUGE)
    oTabellenblattWuerfel.Cells(iZeile + 1, iSpalte + 0) = Chr(ASC_AUGE)
    oTabellenblattWuerfel.Cells(iZeile + 1, iSpalte + 1) = ""
    oTabellenblattWuerfel.Cells(iZeile + 1, iSpalte + 2) = Chr(ASC_AUGE)
    oTabellenblattWuerfel.Cells(iZeile + 2, iSpalte + 0) = Chr(ASC_AUGE)
    oTabellenblattWuerfel.Cells(iZeile + 2, iSpalte + 1) = Chr(ASC_AUGE)
    oTabellenblattWuerfel.Cells(iZeile + 2, iSpalte + 2) = Chr(ASC_AUGE)
```

```
  Case 9
    oTabellenblattWuerfel.Cells(iZeile + 0, iSpalte + 0) = Chr(ASC_AUGE)
    oTabellenblattWuerfel.Cells(iZeile + 0, iSpalte + 1) = Chr(ASC_AUGE)
    oTabellenblattWuerfel.Cells(iZeile + 0, iSpalte + 2) = Chr(ASC_AUGE)
    oTabellenblattWuerfel.Cells(iZeile + 1, iSpalte + 0) = Chr(ASC_AUGE)
    oTabellenblattWuerfel.Cells(iZeile + 1, iSpalte + 1) = Chr(ASC_AUGE)
    oTabellenblattWuerfel.Cells(iZeile + 1, iSpalte + 2) = Chr(ASC_AUGE)
    oTabellenblattWuerfel.Cells(iZeile + 2, iSpalte + 0) = Chr(ASC_AUGE)
    oTabellenblattWuerfel.Cells(iZeile + 2, iSpalte + 1) = Chr(ASC_AUGE)
    oTabellenblattWuerfel.Cells(iZeile + 2, iSpalte + 2) = Chr(ASC_AUGE)
End Select

oTabellenblattWuerfel.Cells(iZeile - 1, iSpalte - 1) = i 'Wurfergebis abspeichern
WuerfelDrehen = i                                        'und als Rueckgabewert festlegen.

                                                         'Fehlerbehandlungsroutine:
Beenden:                                                 '=========================
  Application.Cursor = xlDefault                         'Sanduhr ausblenden.
  Exit Function                                          'Funktion abbrechen.

Fehler:                                                  'Fehlermarke - Fehlerbehandlung:
  WuerfelDrehen = 0                                      'Returnwert = Fehler setzen.
  MsgBox Err.Description                                 'Fehlermeldung ausgeben.
  Resume Beenden                                         'Bei Funktionsende weitermachen.
  Resume                                                 'Nur fuer Testzwecke hinterlegt.
End Function                                             'Funktionsende.
```

4.3.4.) Funktion zum Simulieren eines Würfels

Erweitern wir dazu unsere zuvor angelegte Funktion: Wuerfeln1

```
Public Function Wuerfeln1() As Boolean
'*************************************************************************/
'*      INTERNES UNTERPROGRAMM - INTERNE FUNKTION                        */
'*************************************************************************/
'*----------------------------------------------------------------------*/
'*                                                                      */
'*      FUNKTION:       Wuerfeln1                                       */
'*                                                                      */
'*      BESCHREIBUNG:   Simuliert das Wuerfeln fuer einen einzigen Wuerfel. */
'*                                                                      */
'*      RETURN          Boolean ..... True = Funktion in Ordnung        */
'*                                    False = Fehler in Funktion        */
'*      PARAMETER:      keine                                           */
'*                                                                      */
'*----------------------------------------------------------------------*/
Dim i                   As Integer                  'Allgemeine Zaehlvariable.
Dim iWuerfe             As Integer                  'Anzahl der Wuerfeldrehungen.

On Error GoTo Fehler                                'Bei Fehler zur Fehlermarke.
Wuerfeln1 = True                                    'Returnwert = alles OK setzen.

ThisWorkbook.Activate                               'Aktuelle Excel-Datei festlegen.
Set oTabellenblattWuerfel = Sheets(TABELLENBLATT)   'Akt. Tabellenblatt festlegen.
oTabellenblattWuerfel.Select                        'Tabellenblatt auswaehlen.
Range("A1").Select                                  'Zelle A1 selektieren.
Randomize (Timer)                                   'Zufallsgenerator initialisieren.
```

```
iWuerfe = Int(MIN_WUERFE * Rnd() + (MAX_WUERFE - MIN_WUERFE)) 'Zufaellige Anzahl der
For i = 1 To iWuerfe                                          'Drehungen festlegen und den
  Call WuerfelDrehen("WUERFEL")                               'Wuerfel einmal drehen lassen.
  Sleep (DAUER_WUERFE)                                        'Warten auf naechste Drehung.
Next i                                                        'Weiter zur naechsten Drehung.

                                                              'Fehlerbehandlungsroutine:
Beenden:                                                      '=========================
  Application.Cursor = xlDefault                              'Sanduhr ausblenden.
  Exit Function                                               'Funktion abbrechen.

Fehler:                                                       'Fehlermarke - Fehlerbehandlung:
  Wuerfeln1 = False                                           'Returnwert = Fehler setzen.
  MsgBox Err.Description                                      'Fehlermeldung ausgeben.
  Resume Beenden                                              'Bei Funktionsende weitermachen.
  Resume                                                      'Nur fuer Testzwecke hinterlegt.
End Function                                                  'Funktionsende.
```

Zuerst initialisieren wir den Zufallsgenerator (bei richtigen Spielen sollte das bereits beim Öffnen der Datei erfolgen). Danach legen wir die Anzahl der Würfeldrehungen per Zufallsgenerator fest (hier zwischen 30 und 50 Drehungen) und rufen dann für jede Drehung des Würfels unsere Funktion zum Darstellen der Augenzahl des Würfels auf. Danach warten wir eine bestimmte Zeit (damit der Spieler den aktuellen Wurf sieht) und lassen anschließend den Würfel weiterdrehen.

Sieht ja schon ganz gut aus – nicht wahr?

4.3.5.) Funktion zum gleichzeitigen Simulieren mehrerer Würfel

Dazu erweitern wir wieder unsere zuvor angelegte Funktion: Wuerfeln5 (für 5 Würfel)

```
Public Function Wuerfeln5() As Boolean
'*************************************************************************/
'*      INTERNES UNTERPROGRAMM - INTERNE FUNKTION                        */
'*************************************************************************/
'*-----------------------------------------------------------------------*/
'*                                                                       */
'*      FUNKTION:       Wuerfeln5                                        */
'*                                                                       */
'*      BESCHREIBUNG:   Simuliert das gleichzeitige Wuerfeln mit fuenf   */
'*                      Wuerfeln.                                        */
'*                                                                       */
'*      RETURN          Boolean .......... True = Funktion in Ordnung    */
'*                                         False = Fehler in Funktion    */
'*      PARAMETER:      keine                                            */
'*                                                                       */
'*-----------------------------------------------------------------------*/
Dim i                   As Integer                  'Allgemeine Zaehlvariable.
Dim iWuerfe1            As Integer                  'Anzahl Drehungen Wuerfel 1.
Dim iWuerfe2            As Integer                  'Anzahl Drehungen Wuerfel 2.
Dim iWuerfe3            As Integer                  'Anzahl Drehungen Wuerfel 3.
Dim iWuerfe4            As Integer                  'Anzahl Drehungen Wuerfel 4.
Dim iWuerfe5            As Integer                  'Anzahl Drehungen Wuerfel 5.

On Error GoTo Fehler                                'Bei Fehler zur Fehlermarke.
Wuerfeln5 = True                                    'Returnwert = alles OK setzen.
```

```
ThisWorkbook.Activate                                   'Aktuelle Excel-Datei festlegen.
Set oTabellenblattWuerfel = Sheets(TABELLENBLATT)       'Akt. Tabellenblatt festlegen.
oTabellenblattWuerfel.Select                            'Tabellenblatt auswaehlen.
Range("A1").Select                                      'Zelle A1 selektieren.
Randomize (Timer)                                       'Zufallsgenerator initialisieren.

iWuerfe1 = Int(MIN_WUERFE * Rnd() + (MAX_WUERFE - MIN_WUERFE)) 'Zufaellige Anzahl der
iWuerfe2 = Int(MIN_WUERFE * Rnd() + (MAX_WUERFE - MIN_WUERFE)) 'Drehungen fuer jeden
iWuerfe3 = Int(MIN_WUERFE * Rnd() + (MAX_WUERFE - MIN_WUERFE)) 'der fuenf Wuerfeln
iWuerfe4 = Int(MIN_WUERFE * Rnd() + (MAX_WUERFE - MIN_WUERFE)) 'einzeln festlegen.
iWuerfe5 = Int(MIN_WUERFE * Rnd() + (MAX_WUERFE - MIN_WUERFE))
For i = 0 To MAX_WUERFE                                 'Fuer alle moeglichen Drehungen:
  If i < iWuerfe1 Then Call WuerfelDrehen("WUERFEL1")   'Wuerfel 1 einmal drehen lassen
  If i < iWuerfe2 Then Call WuerfelDrehen("WUERFEL2")   'Wuerfel 2 einmal drehen lassen
  If i < iWuerfe3 Then Call WuerfelDrehen("WUERFEL3")   'Wuerfel 3 einmal drehen lassen
  If i < iWuerfe4 Then Call WuerfelDrehen("WUERFEL4")   'Wuerfel 4 einmal drehen lassen
  If i < iWuerfe5 Then Call WuerfelDrehen("WUERFEL5")   'Wuerfel 5 einmal drehen lassen
  Sleep (DAUER_WUERFE)                                  'Warten auf naechste Drehung.
Next i                                                  'Weiter zur naechsten Drehung.


                                                        'Fehlerbehandlungsroutine:
Beenden:                                                '=========================
  Application.Cursor = xlDefault                        'Sanduhr ausblenden.
  Exit Function                                         'Funktion abbrechen.
Fehler:                                                 'Fehlermarke - Fehlerbehandlung:
  Wuerfeln5 = False                                     'Returnwert = Fehler setzen.
  MsgBox Err.Description                                'Fehlermeldung ausgeben.
  Resume Beenden                                        'Bei Funktionsende weitermachen.
  Resume                                                'Nur fuer Testzwecke hinterlegt.
End Function                                            'Funktionsende.
```

Zuerst initialisieren wir wieder den Zufallsgenerator. Danach legen wir die Anzahl der Würfeldrehungen für jeden einzelnen Würfel separat fest (per Zufallsgenerator). Dann durchlaufen wir eine Schleife, welche die maximale Anzahl der Drehungen durchläuft. Soll ein Würfel noch einmal gedreht werden, so rufen wir für den entsprechenden Würfel unsere Funktion zum Darstellen der Augenzahl des Würfels auf. Danach warten wir wieder eine bestimmte Zeit lang, damit der Spieler den aktuellen Wurf betrachten kann.

Sieht jetzt schon besser aus – nicht wahr?

4.3.6.) Funktion zum Würfel mit fünf Pokerwürfel

Sollten Sie mit Shape-Objekten noch nicht vertraut sein, dann lesen Sie sich zuerst das nächste Kapitel durch. Hier wird alles was wir zum Programmieren mit Shaps benötigen abgehandelt.

Grundsätzliche Überlegungen zum Würfeln mit Objekten: Zuerst erstellen wir uns in einem beliebigen Bildbearbeitungsprogramm für jede Würfelseite eine Grafik. Die 6 Grafiken importieren wir ins Excel. Um die einzelnen Shapes über den Programmcode besser ansprechen zu können vergeben wir über das Namensfeld für jede Grafik einen sprechenden Namen. Dann positionieren wir alle Grafiken für einen Würfel übereinander (d.h. jede Grafik befindet sich auf derselben Top- und Left-Koordinate). Somit ist nur mehr eine Würfelseite sichtbar – welche ist egal.

Das Würfeln erfolgt per Zufallsgenerator, welcher eine der 6 Seiten bestimmt. Die Grafik für diese Seite des Würfels bringen wir dann in den Vordergrund – damit ist diese Würfelseite jetzt sichtbar. Mit dem Befehl **Do Events** verschaffen wir dem Programm zusätzliche Rechenzeit, damit die Darstellung am Bildschirm aktualisiert wird und der Anwender das Drehen des Würfels mitverfolgen kann.

Nun zur Programmierung: Dazu erweitern wir zuerst die zuvor angelegte Funktion: WuerfelnP. Hier zuerst einmal der vollständige Programmcode.

```
Public Function WuerfelnP() As Boolean
'*************************************************************************/
'*      INTERNES UNTERPROGRAMM - INTERNE FUNKTION                        */
'*************************************************************************/
'*----------------------------------------------------------------------*/
'*                                                                      */
'*      FUNKTION:       WuerfelnP                                       */
'*                                                                      */
'*      BESCHREIBUNG:   Simuliert das Wuerfeln mit fuenf Pokerwuerfeln,  */
'*                      wobei die einzelnen Wuerfelseiten fuer alle fuenf */
'*                      Wuerfel als Shape importiert wurden und Namen wie: */
'*                      W1_1 (fuer Wuerfel 1, Seite 1)                  */
'*                      W1_2 (fuer Wuerfel 1, Seite 2)                  */
'*                      ............................. usw.  bis         */
'*                      W5_6 (fuer Wuerfel 5, Seite 6)                  */
'*                      den Shapes zugewiesen wurden (ueber Namensbereich). */
'*                                                                      */
'*      RETURN          Boolean .......... True = Funktion in Ordnung   */
'*                                         False = Fehler in Funktion   */
'*      PARAMETER:      keine                                           */
'*                                                                      */
'*----------------------------------------------------------------------*/
Dim i                   As Integer                  'Allgemeine Zaehlvariable.
Dim j                   As Integer                  'Allgemeine Zaehlvariable.

Dim iWuerfe1            As Integer                  'Anzahl Drehungen Wuerfel 1.
Dim iWuerfe2            As Integer                  'Anzahl Drehungen Wuerfel 2.
Dim iWuerfe3            As Integer                  'Anzahl Drehungen Wuerfel 3.
Dim iWuerfe4            As Integer                  'Anzahl Drehungen Wuerfel 4.
Dim iWuerfe5            As Integer                  'Anzahl Drehungen Wuerfel 5.
Dim stShape             As String                   'Name eines Shapes.

On Error GoTo Fehler                                'Bei Fehler zur Fehlermarke.
WuerfelnP = True                                    'Returnwert = alles OK setzen.

ThisWorkbook.Activate                               'Aktuelle Excel-Datei festlegen.
Set oTabellenblattWuerfel = Sheets(TABELLENBLATT)   'Akt. Tabellenblatt festlegen.
oTabellenblattWuerfel.Select                        'Tabellenblatt auswaehlen.
Range("A1").Select                                  'Zelle A1 selektieren.
Randomize (Timer)                                   'Zufallsgenerator initialisieren.

iWuerfe1 = Int(MIN_WUERFE * Rnd() + (MAX_WUERFE - MIN_WUERFE)) 'Zufaellige Anzahl der
iWuerfe2 = Int(MIN_WUERFE * Rnd() + (MAX_WUERFE - MIN_WUERFE)) 'Drehungen fuer jeden
iWuerfe3 = Int(MIN_WUERFE * Rnd() + (MAX_WUERFE - MIN_WUERFE)) 'der fuenf Wuerfeln
iWuerfe4 = Int(MIN_WUERFE * Rnd() + (MAX_WUERFE - MIN_WUERFE)) 'einzeln festlegen.
iWuerfe5 = Int(MIN_WUERFE * Rnd() + (MAX_WUERFE - MIN_WUERFE))
For i = 0 To MAX_WUERFE                             'Fuer alle moeglichen Drehungen:
  If i < iWuerfe1 Then                              'Wuerfel 1 noch weiter drehen:
    j = Int(MAX_AUGEN * Rnd() + 1)                  'Anzahl Wuerfelaugen bestimmen.
    stShape = "W1_" & Trim(Str(j))                  'Shape-Namen zusammenbasteln
    oTabellenblattWuerfel.Shapes(stShape).ZOrder (msoBringToFront) 'und das Shape
```

```
    oTabellenblattWuerfel.Cells(Range("WUERFEL_P1").Row, Range("WUERFEL_P1").Column)
= j
  End If                                              'in den Vordergrund bringen.

  If i < iWuerfe2 Then                                'Wuerfel 2 noch weiter drehen:
    j = Int(MAX_AUGEN * Rnd() + 1)                    'Anzahl Wuerfelaugen bestimmen.
    stShape = "W2_" & Trim(Str(j))                    'Shape-Namen zusammenbasteln
    oTabellenblattWuerfel.Shapes(stShape).ZOrder (msoBringToFront) 'und das Shape
    oTabellenblattWuerfel.Cells(Range("WUERFEL_P2").Row, Range("WUERFEL_P2").Column)
= j
  End If                                              'in den Vordergrund bringen.

  If i < iWuerfe3 Then                                'Wuerfel 3 noch weiter drehen:
    j = Int(MAX_AUGEN * Rnd() + 1)                    'Anzahl Wuerfelaugen bestimmen.
    stShape = "W3_" & Trim(Str(j))                    'Shape-Namen zusammenbasteln
    oTabellenblattWuerfel.Shapes(stShape).ZOrder (msoBringToFront) 'und das Shape
    oTabellenblattWuerfel.Cells(Range("WUERFEL_P3").Row, Range("WUERFEL_P3").Column)
= j
  End If                                              'in den Vordergrund bringen.

  If i < iWuerfe4 Then                                'Wuerfel 4 noch weiter drehen:
    j = Int(MAX_AUGEN * Rnd() + 1)                    'Anzahl Wuerfelaugen bestimmen.
    stShape = "W4_" & Trim(Str(j))                    'Shape-Namen zusammenbasteln
    oTabellenblattWuerfel.Shapes(stShape).ZOrder (msoBringToFront) 'und das Shape
    oTabellenblattWuerfel.Cells(Range("WUERFEL_P4").Row, Range("WUERFEL_P4").Column)
= j
  End If                                              'in den Vordergrund bringen.

  If i < iWuerfe5 Then                                'Wuerfel 5 noch weiter drehen:
    j = Int(MAX_AUGEN * Rnd() + 1)                    'Anzahl Wuerfelaugen bestimmen.
    stShape = "W5_" & Trim(Str(j))                    'Shape-Namen zusammenbasteln
    oTabellenblattWuerfel.Shapes(stShape).ZOrder (msoBringToFront) 'und das Shape
    oTabellenblattWuerfel.Cells(Range("WUERFEL_P5").Row, Range("WUERFEL_P5").Column)
= j
  End If                                              'in den Vordergrund bringen.

  DoEvents                                            'Ansicht aktualisieren lassen.
  Sleep (DAUER_WUERFE)                                'Warten auf naechste Drehung.
Next i

                                                      'Fehlerbehandlungsroutine:
Beenden:                                              '=========================
  Application.Cursor = xlDefault                      'Sanduhr ausblenden.
  Exit Function                                       'Funktion abbrechen.
Fehler:                                               'Fehlermarke - Fehlerbehandlung:
  WuerfelnP = False                                   'Returnwert = Fehler setzen.
  MsgBox Err.Description                              'Fehlermeldung ausgeben.
  Resume Beenden                                      'Bei Funktionsende weitermachen.
  Resume                                              'Nur fuer Testzwecke hinterlegt.
End Function                                          'Funktionsende.
```

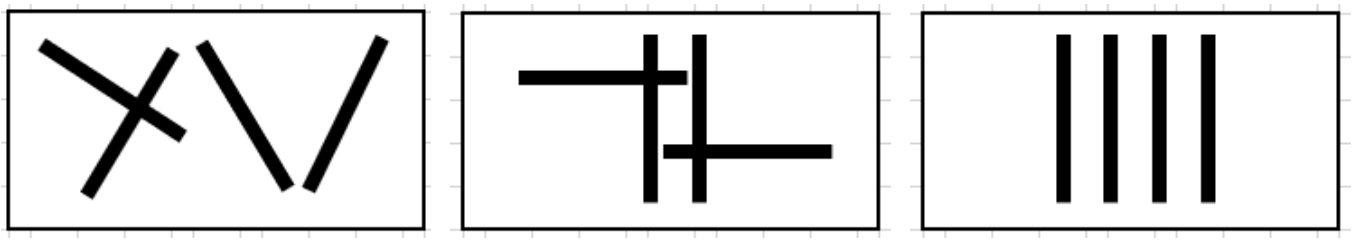
Zuerst initialisieren wir wieder den Zufallsgenerator. Danach legen wir die Anzahl der Würfeldrehungen für jeden einzelnen Würfel separat fest (per Zufallsgenerator). Dann durchlaufen wir eine Schleife, welche die maximale Anzahl der Drehungen durchläuft. Soll ein Würfel noch einmal gedreht werden, so wird die entsprechende Würfelseite in den Vordergrund gebracht. Danach verschaffen wir dem Programm noch etwas Rechenzeit – um die Anzeige am Bildschirm zu aktualisieren, und warten wieder eine bestimmte Zeit lang, damit der Spieler den aktuellen Wurf betrachten kann.

Sieht jetzt wirklich optimal aus – nicht wahr?

4.3.7.) Funktion zum Würfel mit vier Stäben

Grundsätzliche Überlegungen zum Würfeln mit Stäben: Ein Stab wird durch eine Linie (Shape) dargestellt welchem wir über das Namensfeld einen eindeutigen Namen zuweisen. Jeder Stab ist auf der einen Seite weiß und auf der anderen Seite schwarz. Um die Stabseite besser sehen zu können wählen wir eine ziemlich dicke Linienstärke. Per Zufall wird bestimmt, welche Stabseite nach oben zeigt (weiß oder schwarz). Um das Fallen und Drehen der Stäbe zu simulieren verändern wir zufällig den Drehwinkel der Stablinie.

Zuerst erstellen wir uns eine Linie (aus dem Excel-Menü: Formen). Dann legen wir eine Farbe (schwarz), die Linienstärke (Form formatieren: Stärke = 6 Pt.), die Breite (2,5 cm) und den Drehwinkel (0°) fest. Anschließend kopieren wird das Linien-Shape 3 Mal und benennen die vier Stäbe über das Namensfeld zum Beispiel mit: LINIE1, LINIE2, LINIE3 und LINIE4 (nach der Eingabe des Namens nicht vergessen die ENTER-Taste zu drücken – erst dann erfolgt die Namenszuweisung!). Dann legen wir uns im Tabellenblatt einen Bereich fest, in welchem die Stäbe geworfen werden sollen und verschieben die vier Linien in diesen Bereich. Dabei ist zu beachten, daß die Top-Position für alle Stäbe in etwa gleich ist. Drehlt man die Linien um 90 bzw. 180 Grad, dann sollte die waagrechte Linie nicht aus dem Wurfbereich hinausragen. Die Abstände zwischen den senkrechten Stäben sollten in etwas auch gleich groß sein:



Das Werfen der Stäbe erfolgt per Zufallsgenerator, welcher die Farbe und den Drehwinkel für alle Stäbe unterschiedlich zuordnet und am Bildschirm ausgibt. Die gewürfelten Punkte ergeben sich aus der Anzahl der weißen Stäbe – also 1 bis 4. Wirft man hingegen 4 schwarze Stäbe, dann ist das Wurfergebnis eine Sechs. Mit dem Befehl **Do Events** verschaffen wir dem Programm zusätzliche Rechenzeit, damit die Darstellung am Bildschirm aktualisiert wird und der Anwender das Fallen der Stäbe mitverfolgen kann.

Nun zur Programmierung: Dazu erweitern wir zuerst die zuvor angelegte Funktion: WuerfelnS. Hier zuerst einmal der vollständige Programmcode:

```
Public Function WuerfelnS() As Boolean
'*************************************************************************/
'*      INTERNES UNTERPROGRAMM - INTERNE FUNKTION                        */
'*************************************************************************/
'*----------------------------------------------------------------------*/
'*                                                                       */
'*      FUNKTION:       WuerfelnS                                        */
'*                                                                       */
'*      BESCHREIBUNG:   Simuliert das Wuerfeln mit vier Staeben, wobei eine */
'*                      Stabseite weiss und die andere schwarz ist. Jeder   */
'*                      Stab ist ein Shape (Linie) welche diese Namen haben:*/
'*                      LINIE1 (fuer den 1. Stab)                        */
'*                      LINIE2 (fuer den 2. Stab)                        */
'*                      LINIE3 (fuer den 3. Stab)                        */
'*                      LINIE4 (fuer den 4. Stab)                        */
'*                                                                       */
'*      RETURN          Boolean ..... True = Funktion in Ordnung         */
'*                                    False = Fehler in Funktion         */
```

```
'*       PARAMETER:      keine                                          */
'*                                                                      */
'*----------------------------------------------------------------------*/
Dim i                   As Integer                  'Allgemeine Zaehlvariable.
Dim iWinkel             As Integer                  'Allgemeine Zaehlvariable.
Dim lFarbe              As Long                     'Allgemeine Zaehlvariable.
Dim iPunkte             As Integer                  'Allgemeine Zaehlvariable.
Dim iWuerfe1            As Integer                  'Anzahl Drehungen Stab 1.
Dim iWuerfe2            As Integer                  'Anzahl Drehungen Stab 2.
Dim iWuerfe3            As Integer                  'Anzahl Drehungen Stab 3.
Dim iWuerfe4            As Integer                  'Anzahl Drehungen Stab 4.

On Error GoTo Fehler                                'Bei Fehler zur Fehlermarke.
WuerfelnS = True                                    'Returnwert = alles OK setzen.

ThisWorkbook.Activate                               'Aktuelle Excel-Datei festlegen.
Set oTabellenblattWuerfel = Sheets(TABELLENBLATT)   'Akt. Tabellenblatt festlegen.
oTabellenblattWuerfel.Select                        'Tabellenblatt auswaehlen.
Range("A1").Select                                  'Zelle A1 selektieren.
Randomize (Timer)                                   'Zufallsgenerator initialisieren.

oTabellenblattWuerfel.Cells(28, 17) = ""            'Wurfergebnis ruecksetzen.
iWuerfe1 = Int(MIN_WUERFE * Rnd() + (MAX_WUERFE - MIN_WUERFE)) 'Zufaellige Anzahl der
iWuerfe2 = Int(MIN_WUERFE * Rnd() + (MAX_WUERFE - MIN_WUERFE)) 'Drehungen fuer jeden
iWuerfe3 = Int(MIN_WUERFE * Rnd() + (MAX_WUERFE - MIN_WUERFE)) 'der vier Staebe
iWuerfe4 = Int(MIN_WUERFE * Rnd() + (MAX_WUERFE - MIN_WUERFE)) 'einzeln festlegen.

For i = 0 To MAX_WUERFE                             'Fuer alle moeglichen Drehungen:
  If i < iWuerfe1 Then                              'Stab 1 soll noch weiter drehen:
    If Int(2 * Rnd() + 1) = 1 Then lFarbe = FARBE_SW Else lFarbe = FARBE_WS
    If Int(2 * Rnd() + 1) = 1 Then iWinkel = Int(180 * Rnd() + 1) Else iWinkel =
Int(180 * Rnd() + 1) * (-1)
    oTabellenblattWuerfel.Shapes("LINIE1").Line.ForeColor.RGB = lFarbe 'Stabfarbe und
    oTabellenblattWuerfel.Shapes("LINIE1").Rotation = iWinkel 'Drehwinkel zuweisen.
  End If

  If i < iWuerfe2 Then                              'Stab 2 soll noch weiter drehen:
    If Int(2 * Rnd() + 1) = 1 Then lFarbe = FARBE_SW Else lFarbe = FARBE_WS
    If Int(2 * Rnd() + 1) = 1 Then iWinkel = Int(180 * Rnd() + 1) Else iWinkel =
Int(180 * Rnd() + 1) * (-1)
    oTabellenblattWuerfel.Shapes("LINIE2").Line.ForeColor.RGB = lFarbe 'Stabfarbe und
    oTabellenblattWuerfel.Shapes("LINIE2").Rotation = iWinkel 'Drehwinkel zuweisen.
  End If

  If i < iWuerfe3 Then                              'Stab 3 soll noch weiter drehen:
    If Int(2 * Rnd() + 1) = 1 Then lFarbe = FARBE_SW Else lFarbe = FARBE_WS
    If Int(2 * Rnd() + 1) = 1 Then iWinkel = Int(180 * Rnd() + 1) Else iWinkel =
Int(180 * Rnd() + 1) * (-1)
    oTabellenblattWuerfel.Shapes("LINIE3").Line.ForeColor.RGB = lFarbe 'Stabfarbe und
    oTabellenblattWuerfel.Shapes("LINIE3").Rotation = iWinkel 'Drehwinkel zuweisen.
  End If

  If i < iWuerfe4 Then                              'Stab 4 soll noch weiter drehen:
    If Int(2 * Rnd() + 1) = 1 Then lFarbe = FARBE_SW Else lFarbe = FARBE_WS
    If Int(2 * Rnd() + 1) = 1 Then iWinkel = Int(180 * Rnd() + 1) Else iWinkel =
Int(180 * Rnd() + 1) * (-1)
    oTabellenblattWuerfel.Shapes("LINIE4").Line.ForeColor.RGB = lFarbe 'Stabfarbe und
    oTabellenblattWuerfel.Shapes("LINIE4").Rotation = iWinkel 'Drehwinkel zuweisen.
  End If
```

```
  DoEvents                                          'Ansicht aktualisieren lassen.
  Sleep (DAUER_WUERFE / 2)                          'Warten auf naechste Drehung.
Next i

iPunkte = 0                                         'Wurfergebnis ruecksetzen und
If oTabellenblattWuerfel.Shapes("LINIE1").Line.ForeColor.RGB = FARBE_WS Then iPunkte
= iPunkte + 1
If oTabellenblattWuerfel.Shapes("LINIE2").Line.ForeColor.RGB = FARBE_WS Then iPunkte
= iPunkte + 1
If oTabellenblattWuerfel.Shapes("LINIE3").Line.ForeColor.RGB = FARBE_WS Then iPunkte
= iPunkte + 1
If oTabellenblattWuerfel.Shapes("LINIE4").Line.ForeColor.RGB = FARBE_WS Then iPunkte
= iPunkte + 1
If iPunkte = 0 Then iPunkte = 6                     'Punktezahl ermitteln.
oTabellenblattWuerfel.Cells(28, 17) = iPunkte       'Wurfergebnis ausgeben.


                                                    'Fehlerbehandlungsroutine:
Beenden:                                            '=========================
  Application.Cursor = xlDefault                    'Sanduhr ausblenden.
  Exit Function                                     'Funktion abbrechen.

Fehler:                                             'Fehlermarke - Fehlerbehandlung:
  WuerfelnS = False                                 'Returnwert = Fehler setzen.
  MsgBox Err.Description                            'Fehlermeldung ausgeben.
  Resume Beenden                                    'Bei Funktionsende weitermachen.
  Resume                                            'Nur fuer Testzwecke hinterlegt.
End Function                                        'Funktionsende.
```

Zuerst initialisieren wir wieder den Zufallsgenerator. Danach legen wir die Anzahl der Drehungen für jeden einzelnen Stab fest (per Zufallsgenerator). Dann durchlaufen wir eine Schleife, welche die maximale Anzahl der Drehungen durchläuft. Soll ein Stab noch einmal gedreht werden, so wird die Farbe und der Drehwinkel (-180° bis +180°) bestimmt und dem Shape zugewiesen. Danach verschaffen wir dem Programm noch etwas Rechenzeit – um die Anzeige am Bildschirm zu aktualisieren, und warten wieder eine bestimmte Zeit lang, damit der Spieler das Werfen der Stäbe betrachten kann. Am Ende der Funktion berechnen wir noch die geworfene Punkteanzahl und geben sie im Tabellenblatt aus.

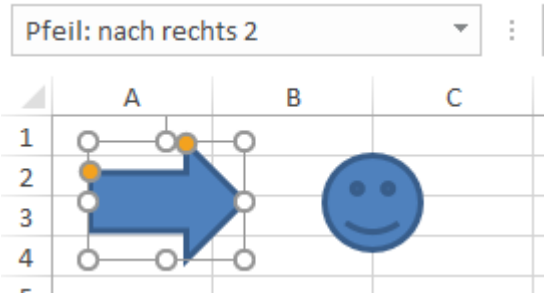
Sieht ziemlich interessant aus – nicht wahr?

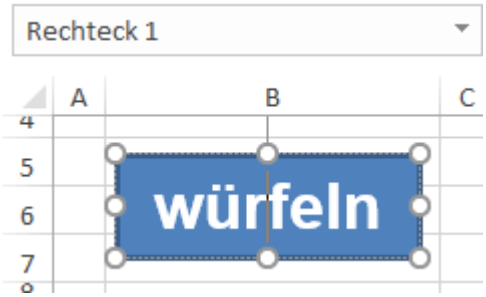
5.) Arbeiten mit Shapes

Was sind eigentlich Shapes? Aus dem englischen übersetzt bedeutet es so viel wie: Formen. Excel hat einen Menüpunkt **Formen** aus dem man grafische Objekte wie Rechtecke, Kreise, Linien, Pfeile, etc. auswählen und in eine Excel-Datei einfügen kann – und natürlich ist jede eingefügte Form auch ein Shape. Aber Excel bezeichnet auch importierte Bilder, Steuerelemente, Piktogramme (Symbole), Kommentare und Diagramme als Shape. Daher kann man für Excel sagen, daß jedes Objekt, welches irgendwie mit einer grafischen Dartsellung zu tun hat, als Shape bezeichnet wird – oder genauer gesamt: alle Objekte die sich auf der Zeichnungsebene befinden oder alles was nicht Inhalt einer Zelle ist.

Beim Einfügen eines Shapes wird automatisch ein Name zugewiesen, z.B. Smiley1, Pfeil: nach rechts 2, Grafik 2, Verbinder: gewinkelt 4, … Wie sich die Nummer am Ende generiert ist auch mir noch ein Rätsel. Daher sollte man für jedes benötigte Shape einen eindeutigen, für sich sprechenden Namen, selbst festlegen.

Für die Benennung von Shapes gibt es in Excel nur eine einzige Möglichkeit: nämlich die Eingabe der eigenen Benennung in das Namensfeld und abschließend die ENTER-Taste (RETURN- oder auch Bestätigungs-Taste) zu drücken. Das Namensfeld befindet sich immer links oberhalb der Zeile mit den Spaltenbuchstaben. In neueren Excel-Versionen läß sich dieses Feld vergrößern, indem man bei gedrückter Maustaste auf die drei Punkte rechts neben dem Namensfeld, das Feld in die Breite zieht (war bei älteren Versionen nicht möglich, daher meine damalige Empfehlung nur solche Bezeichnungen zu verwenden die vollständig angezeigt werden). Im Namensfeld wird jeweils der Name des aktuell selektierten Shapes angezeigt; hier steht im Namensfeld etwa: Pfeil: nach rechts 2 bzw. Rechteck 1:





Über das Namensfeld können keine doppelten Bezeichnungen eingegeben werden. Jedoch über den Programmcode – was absolut nicht zu empfehlen ist, da man dann jeweils nur auf ein zufälliges Shape mit dem gewünschten Namen zugreifen kann. Hier wird nur ein Shape auf Top-Position 100 verschoben.

```
For i = 1 To ActiveSheet.Shapes.Count
  ActiveSheet.Shapes(i).Name = "xxx"
Next i
ActiveSheet.Shapes("xxx").Top = 100
```

Möglich ist diese doppelte Namensvergabe, da jedes Shape zusätzlich eine eindeutige ID-Nummer zugewiesen bekommt. Die ID-Nummer kann jedoch nicht über den Programmcode geändert werden sondern wird automatisch von Excel verwaltet.

6.) Schlußbemerkungen

Wenn Sie alle hier besprochenen Funktionen in einem einzigen Modul hinterlegt haben, dann können Sie dieses Modul exportieren und in anderen Spielen, wo Würfeln benötigt werden, wieder importieren. Sie müssen dann nur mehr die Makros in das entsprechende Makro-Modul verschieben und die globalen Difinitionen ins Modul: Globales.

Sie können hier aber auch weiter Würfel entwerfen. So gibt es Würfel die eine von 6 Farben bestimmen, einen Würfel der 12 Flächen besitzt oder Würfel mit verschiedenen Symbolen. Was fällt mir noch ein: Multiplikator-Würfel, …

So das wär´s dann wieder Mal – viel Spaß beim Gestalten Eurer eigenen Würfel.

Anhang A – Vollständiger Programmcode

```
'*****************************************************************************/
'*                                                                           */
'*      PROGRAMM:       Wuerfelsimulation                                    */
'*      PROJEKT:        Spiele programmieren in Excel                        */
'*                                                                           */
'*      MODULNAME:      Wuerfeln                                             */
'*      ARCHIVIERT:     -                                                    */
'*                                                                           */
'*      VERSION:        V  01.00                                             */
'*      VERSIONSDATUM:  11 Jan 2019 12:00:00                                 */
'*                                                                           */
'*      BEARBEITER:     Ing. Harald Mitsch, TBM                              */
'*                                                                           */
'*===========================================================================*/
'*                                                                           */
'*      (C) 2019 by TBM-Technisches Buero Mitsch   Elektrotechn. Planungen   */
'*      Lizenz: FREEWARE                           und ET - Installationen   */
'*      Josefgasse 6, 3380 Poechlarn               EDV - Selfmade Software   */
'*      Tel.:02757 / 46 56 bzw. 0676 /588 09 16    allgemeines Zeichenbüro   */
'*                                                                           */
'*===========================================================================*/
'*                                                                           */
'*      BESCHREIBUNG:   Beinhaltet alle Prozeduren und Funktionen die fuer   */
'*                      Simulationen des Wuerfelns benoetigt werden.         */
'*                                                                           */
'*      NEBENEFFEKTE:   -                                                    */
'*                                                                           */
'*      GRENZEN:        -                                                    */
'*                                                                           */
'*===========================================================================*/
'*                                                                           */
'*      DEKLARIERTE FUNKTIONEN:   Sleep                                      */
'*                                                                           */
'*      EXPORTIERTE FUNKTIONEN:   keine                                      */
'*                                                                           */
'*      IMPORTIERTE FUNKTIONEN:   keine                                      */
'*                                                                           */
'*===========================================================================*/
'*                                                                           */
'*      EXPORTIERTE VARIABLEN:    alles aus Deklarationsbereich              */
'*                                                                           */
'*      IMPORTIERTE VARIABLEN:    keine                                      */
'*                                                                           */
'*===========================================================================*/
'*                                                                           */
'*      INTERNE PROZEDUREN:       Wuerfeln_1                                 */
'*                                Wuerfeln_5                                 */
'*                                Wuerfeln_P                                 */
'*                                Wuerfeln_S                                 */
'*                                                                           */
'*===========================================================================*/
'*                                                                           */
'*      INTERNE FUNKTIONEN:       Wuerfeln1                                  */
'*                                Wuerfeln5                                  */
'*                                WuerfelnP                                  */
'*                                WuerfelnS                                  */
'*                                WuerfelDrehen                              */
'*                                WuerfelPositionieren                       */
'*                                StaebePositionieren                        */
'*                                                                           */
'*===========================================================================*/
'*                                                                           */
'*      AENDERUNGEN: - Modification History:                                 */
'*                                                                           */
```

```
'*      V 01.00     11.01.2019    Mitsch Harald, TBM, Initial Version       */
'*                                                                          */
'****************************************************************************/
Option Explicit                                     'Alle Variablen muessen
                                                    'explizit angegeben werden.

'----------------------------------------------------------------------------/
'-  Deklarierte API-Funktionen fuer 64 Bit Office Versionen                 -/
'----------------------------------------------------------------------------/
#If VBA7 Then                                       'Fuer Sleep-Funktion 64 Bit Excel:
  Public Declare PtrSafe Sub Sleep Lib "kernel32" (ByVal dwMilliseconds As LongPtr)
'----------------------------------------------------------------------------/
'-  Deklarierte API-Funktionen fuer 32 Bit Office Versionen                 -/
'----------------------------------------------------------------------------/
#Else                                               'Fuer Sleep-Funktion 32 Bit Excel:
  Public Declare Sub Sleep Lib "kernel32" (ByVal dwMilliseconds As Long)
#End If


'----------------------------------------------------------------------------/
'-  Definitionen fuer die Wuerfelsimulation                                 -/
'----------------------------------------------------------------------------/
Global Const TABELLENBLATT = "Wuerfel"              'Tabellenblatt fuer Spielfeld.
Global Const MAX_AUGEN = 6  'oder 9                 'Maximale Anzahl der Wuerfelaugen.
Global Const ASC_AUGE = 108                         'Acs-Code fuer Wuerfelauge 108=kleines L.
Global Const MIN_WUERFE = 30                        'Minimale Anzahl der Wuerfeldrehungen.
Global Const MAX_WUERFE = 50                        'Maximale Anzahl der Wuerfeldrehungen.
Global Const DAUER_WUERFE = 30                      'Anzeigedauer nach einer Wuerfeldrehung.

Global oTabellenblattWuerfel        As Object       'Variable fuer Tabellenblatt.

'----------------------------------------------------------------------------/
'-  Definitionen fuer Farben                                                -/
'----------------------------------------------------------------------------/
Global Const SCHWARZ = 0                            'Farbe schwarz = RGB(0, 0, 0)
Global Const WEISS = 16777215                       'Farbe weiss = RGB(255, 255, 255)
Global Const DUNKELGELB = 62207                     'Farbe dunkelgelb = RGB(255, 242, 0)

Global Const FARBE_SW = SCHWARZ                     'Farbe fuer schwarze Stabseite.
Global Const FARBE_WS = WEISS                       'Farbe fuer weisse Stabseite.



'----------------------------------------------------------------------------/
'-  PROZEDUREN                                                              -/
'----------------------------------------------------------------------------/

Sub Wuerfeln_1()
  Call Wuerfeln1                                    'Mit einem Wuerfel wuerfeln.
End Sub


Sub Wuerfeln_5()
  Call Wuerfeln5                                    'Mit fuenf Wuerfel wuerfeln.
End Sub


Sub Wuerfeln_P()
  Call WuerfelnP                                    'Mit fuenf Pokerwuerfel wuerfeln.
End Sub


Sub Wuerfeln_S()
  Call WuerfelnS                                    'Mit vier Staeben wuerfeln.
End Sub
```

```
Public Function Wuerfeln1() As Boolean
'*************************************************************************/
'*      INTERNES UNTERPROGRAMM - INTERNE FUNKTION                        */
'*************************************************************************/
'*-----------------------------------------------------------------------*/
'*                                                                       */
'*      FUNKTION:       Wuerfeln1                                        */
'*                                                                       */
'*      BESCHREIBUNG:   Simuliert das Wuerfeln fuer einen einzigen Wuerfel. */
'*                                                                       */
'*      RETURN          Boolean ..... True = Funktion in Ordnung         */
'*                                    False = Fehler in Funktion         */
'*      PARAMETER:      keine                                            */
'*                                                                       */
'*-----------------------------------------------------------------------*/
Dim i                   As Integer                  'Allgemeine Zaehlvariable.
Dim iWuerfe             As Integer                  'Zufaellige Anzahl der Wuerfeldrehungen.

On Error GoTo Fehler                                'Bei Fehler zur Fehlermarke.
Wuerfeln1 = True                                    'Returnwert = alles OK setzen.

ThisWorkbook.Activate                               'Aktuelle Excel-Datei festlegen.
Set oTabellenblattWuerfel = Sheets(TABELLENBLATT)   'Akt. Tabellenblatt festlegen.
oTabellenblattWuerfel.Select                        'Tabellenblatt auswaehlen.
Range("A1").Select                                  'Zelle A1 selektieren.
Randomize (Timer)                                   'Zufallsgenerator initialisieren.

iWuerfe = Int(MIN_WUERFE * Rnd() + (MAX_WUERFE - MIN_WUERFE)) 'Zufaellige Anzahl der
For i = 1 To iWuerfe                                'Drehungen festlegen und den
  Call WuerfelDrehen("WUERFEL")                     'Wuerfel einmal drehen lassen.
  Sleep (DAUER_WUERFE)                              'Warten auf naechste Drehung.
Next i                                              'Weiter zur naechsten Drehung.


                                                    'Fehlerbehandlungsroutine:
Beenden:                                            '=========================
  Application.Cursor = xlDefault                    'Sanduhr ausblenden.
  Exit Function                                     'Funktion abbrechen.

Fehler:                                             'Fehlermarke - Fehlerbehandlung:
  Wuerfeln1 = False                                 'Returnwert = Fehler setzen.
  MsgBox Err.Description                            'Fehlermeldung ausgeben.
  Resume Beenden                                    'Bei Funktionsende weitermachen.
  Resume                                            'Nur fuer Testzwecke hinterlegt.
End Function                                        'Funktionsende.
```

```
Public Function Wuerfeln5() As Boolean
'*************************************************************************/
'*      INTERNES UNTERPROGRAMM - INTERNE FUNKTION                        */
'*************************************************************************/
'*-----------------------------------------------------------------------*/
'*                                                                       */
'*      FUNKTION:       Wuerfeln5                                        */
'*                                                                       */
'*      BESCHREIBUNG:   Simuliert das gleichzeitige Wuerfeln mit fuenf   */
'*                      Wuerfeln.                                        */
'*                                                                       */
'*      RETURN          Boolean .......... True = Funktion in Ordnung    */
'*                                         False = Fehler in Funktion    */
'*      PARAMETER:      keine                                            */
'*                                                                       */
'*-----------------------------------------------------------------------*/
Dim i                   As Integer                  'Allgemeine Zaehlvariable.
Dim iWuerfe1            As Integer                  'Zufaellige Anzahl Drehungen Wuerfel 1.
Dim iWuerfe2            As Integer                  'Zufaellige Anzahl Drehungen Wuerfel 2.
Dim iWuerfe3            As Integer                  'Zufaellige Anzahl Drehungen Wuerfel 3.
Dim iWuerfe4            As Integer                  'Zufaellige Anzahl Drehungen Wuerfel 4.
Dim iWuerfe5            As Integer                  'Zufaellige Anzahl Drehungen Wuerfel 5.

On Error GoTo Fehler                                'Bei Fehler zur Fehlermarke.
Wuerfeln5 = True                                    'Returnwert = alles OK setzen.

ThisWorkbook.Activate                               'Aktuelle Excel-Datei festlegen.
Set oTabellenblattWuerfel = Sheets(TABELLENBLATT)   'Akt. Tabellenblatt festlegen.
oTabellenblattWuerfel.Select                        'Tabellenblatt auswaehlen.
Range("A1").Select                                  'Zelle A1 selektieren.
Randomize (Timer)                                   'Zufallsgenerator initialisieren.

iWuerfe1 = Int(MIN_WUERFE * Rnd() + (MAX_WUERFE - MIN_WUERFE))   'Zufaellige Anzahl der
iWuerfe2 = Int(MIN_WUERFE * Rnd() + (MAX_WUERFE - MIN_WUERFE))   'Drehungen fuer jeden
iWuerfe3 = Int(MIN_WUERFE * Rnd() + (MAX_WUERFE - MIN_WUERFE))   'der fuenf Wuerfeln
iWuerfe4 = Int(MIN_WUERFE * Rnd() + (MAX_WUERFE - MIN_WUERFE))   'einzeln festlegen.
iWuerfe5 = Int(MIN_WUERFE * Rnd() + (MAX_WUERFE - MIN_WUERFE))
For i = 0 To MAX_WUERFE                             'Fuer alle moeglichen Drehungen:
  If i < iWuerfe1 Then Call WuerfelDrehen("WUERFEL1") 'Wuerfel 1 einmal drehen lassen.
  If i < iWuerfe2 Then Call WuerfelDrehen("WUERFEL2") 'Wuerfel 2 einmal drehen lassen.
  If i < iWuerfe3 Then Call WuerfelDrehen("WUERFEL3") 'Wuerfel 3 einmal drehen lassen.
  If i < iWuerfe4 Then Call WuerfelDrehen("WUERFEL4") 'Wuerfel 4 einmal drehen lassen.
  If i < iWuerfe5 Then Call WuerfelDrehen("WUERFEL5") 'Wuerfel 5 einmal drehen lassen.
  Sleep (DAUER_WUERFE)                              'Warten auf naechste Drehung.
Next i                                              'Weiter zur naechsten Drehung.

                                                    'Fehlerbehandlungsroutine:
Beenden:                                            '=========================
  Application.Cursor = xlDefault                    'Sanduhr ausblenden.
  Exit Function                                     'Funktion abbrechen.

Fehler:                                             'Fehlermarke - Fehlerbehandlung:
  Wuerfeln5 = False                                 'Returnwert = Fehler setzen.
  MsgBox Err.Description                            'Fehlermeldung ausgeben.
  Resume Beenden                                    'Bei Funktionsende weitermachen.
  Resume                                            'Nur fuer Testzwecke hinterlegt.
End Function                                        'Funktionsende.
```

```
Public Function WuerfelnP() As Boolean
'*************************************************************************/
'*      INTERNES UNTERPROGRAMM - INTERNE FUNKTION                        */
'*************************************************************************/
'*-----------------------------------------------------------------------*/
'*                                                                       */
'*      FUNKTION:       WuerfelnP                                        */
'*                                                                       */
'*      BESCHREIBUNG:   Simuliert das Wuerfeln mit fuenf Pokerwuerfeln,  */
'*                      wobei die einzelnen Wuerfelseiten fuer alle fuenf*/
'*                      Wuerfel als Shape importiert wurden und Namen wie:*/
'*                      W1_1 (fuer Wuerfel 1, Seite 1)                   */
'*                      W1_2 (fuer Wuerfel 1, Seite 2)                   */
'*                      ............................. usw.  bis          */
'*                      W5_6 (fuer Wuerfel 5, Seite 6)                   */
'*                      den Shapes zugewiesen wurden (ueber Namensbereich).*/
'*                                                                       */
'*      RETURN          Boolean ..... True = Funktion in Ordnung         */
'*                                    False = Fehler in Funktion         */
'*      PARAMETER:      keine                                            */
'*                                                                       */
'*-----------------------------------------------------------------------*/
Dim i                   As Integer                  'Allgemeine Zaehlvariable.
Dim j                   As Integer                  'Allgemeine Zaehlvariable.
Dim iWuerfe1            As Integer                  'Zufaellige Anzahl Drehungen Wuerfel 1.
Dim iWuerfe2            As Integer                  'Zufaellige Anzahl Drehungen Wuerfel 2.
Dim iWuerfe3            As Integer                  'Zufaellige Anzahl Drehungen Wuerfel 3.
Dim iWuerfe4            As Integer                  'Zufaellige Anzahl Drehungen Wuerfel 4.
Dim iWuerfe5            As Integer                  'Zufaellige Anzahl Drehungen Wuerfel 5.
Dim stShape             As String                   'Name eines Shapes.

On Error GoTo Fehler                                'Bei Fehler zur Fehlermarke.
WuerfelnP = True                                    'Returnwert = alles OK setzen.

ThisWorkbook.Activate                               'Aktuelle Excel-Datei festlegen.
Set oTabellenblattWuerfel = Sheets(TABELLENBLATT)   'Akt. Tabellenblatt festlegen.
oTabellenblattWuerfel.Select                        'Tabellenblatt auswaehlen.
Range("A1").Select                                  'Zelle A1 selektieren.
Randomize (Timer)                                   'Zufallsgenerator initialisieren.

iWuerfe1 = Int(MIN_WUERFE * Rnd() + (MAX_WUERFE - MIN_WUERFE))    'Zufaellige Anzahl der
iWuerfe2 = Int(MIN_WUERFE * Rnd() + (MAX_WUERFE - MIN_WUERFE))    'Drehungen fuer jeden
iWuerfe3 = Int(MIN_WUERFE * Rnd() + (MAX_WUERFE - MIN_WUERFE))    'der fuenf Wuerfeln
iWuerfe4 = Int(MIN_WUERFE * Rnd() + (MAX_WUERFE - MIN_WUERFE))    'einzeln festlegen.
iWuerfe5 = Int(MIN_WUERFE * Rnd() + (MAX_WUERFE - MIN_WUERFE))
For i = 0 To MAX_WUERFE                             'Fuer alle moeglichen Drehungen:
  If i < iWuerfe1 Then                              'Wuerfel 1 soll noch weiter drehen:
    j = Int(MAX_AUGEN * Rnd() + 1)                  'Anzahl Wuerfelaugen bestimmen.
    stShape = "W1_" & Trim(Str(j))                  'Shape-Namen zusammenbasteln
    oTabellenblattWuerfel.Shapes(stShape).ZOrder (msoBringToFront) 'und dann das Shape
    oTabellenblattWuerfel.Cells(Range("WUERFEL_P1").Row, Range("WUERFEL_P1").Column) = j
  End If                                            'in den Vordergrund bringen.

  If i < iWuerfe2 Then                              'Wuerfel 2 soll noch weiter drehen:
    j = Int(MAX_AUGEN * Rnd() + 1)                  'Anzahl Wuerfelaugen bestimmen.
    stShape = "W2_" & Trim(Str(j))                  'Shape-Namen zusammenbasteln
    oTabellenblattWuerfel.Shapes(stShape).ZOrder (msoBringToFront) 'und dann das Shape
    oTabellenblattWuerfel.Cells(Range("WUERFEL_P2").Row, Range("WUERFEL_P2").Column) = j
  End If                                            'in den Vordergrund bringen.

  If i < iWuerfe3 Then                              'Wuerfel 3 soll noch weiter drehen:
    j = Int(MAX_AUGEN * Rnd() + 1)                  'Anzahl Wuerfelaugen bestimmen.
    stShape = "W3_" & Trim(Str(j))                  'Shape-Namen zusammenbasteln
    oTabellenblattWuerfel.Shapes(stShape).ZOrder (msoBringToFront) 'und dann das Shape
    oTabellenblattWuerfel.Cells(Range("WUERFEL_P3").Row, Range("WUERFEL_P3").Column) = j
  End If                                            'in den Vordergrund bringen.
```

```
  If i < iWuerfe4 Then                                  'Wuerfel 4 soll noch weiter drehen:
    j = Int(MAX_AUGEN * Rnd() + 1)                      'Anzahl Wuerfelaugen bestimmen.
    stShape = "W4_" & Trim(Str(j))                      'Shape-Namen zusammenbasteln
    oTabellenblattWuerfel.Shapes(stShape).ZOrder (msoBringToFront) 'und dann das Shape
    oTabellenblattWuerfel.Cells(Range("WUERFEL_P4").Row, Range("WUERFEL_P4").Column) = j
  End If                                                'in den Vordergrund bringen.

  If i < iWuerfe5 Then                                  'Wuerfel 5 soll noch weiter drehen:
    j = Int(MAX_AUGEN * Rnd() + 1)                      'Anzahl Wuerfelaugen bestimmen.
    stShape = "W5_" & Trim(Str(j))                      'Shape-Namen zusammenbasteln
    oTabellenblattWuerfel.Shapes(stShape).ZOrder (msoBringToFront) 'und dann das Shape
    oTabellenblattWuerfel.Cells(Range("WUERFEL_P5").Row, Range("WUERFEL_P5").Column) = j
  End If                                                'in den Vordergrund bringen.

  DoEvents                                              'Ansicht aktualisieren lassen.
  Sleep (DAUER_WUERFE)                                  'Warten auf naechste Drehung.
Next i

                                                        'Fehlerbehandlungsroutine:
Beenden:                                                '=========================
  Application.Cursor = xlDefault                        'Sanduhr ausblenden.
  Exit Function                                         'Funktion abbrechen.

Fehler:                                                 'Fehlermarke - Fehlerbehandlung:
  WuerfelnP = False                                     'Returnwert = Fehler setzen.
  MsgBox Err.Description                                'Fehlermeldung ausgeben.
  Resume Beenden                                        'Bei Funktionsende weitermachen.
  Resume                                                'Nur fuer Testzwecke hinterlegt.
End Function                                            'Funktionsende.


Public Function WuerfelnS() As Boolean
'*************************************************************************/
'*      INTERNES UNTERPROGRAMM - INTERNE FUNKTION                        */
'*************************************************************************/
'*-----------------------------------------------------------------------*/
'*                                                                       */
'*      FUNKTION:       WuerfelnS                                        */
'*                                                                       */
'*      BESCHREIBUNG:   Simuliert das Wuerfeln mit vier Staeben, wobei eine */
'*                      Stabseite weiss und die andere schwarz ist. Jeder   */
'*                      Stab ist ein Shape (Linie) welche diese Namen haben:*/
'*                      LINIE1 (fuer den 1. Stab)                        */
'*                      LINIE2 (fuer den 2. Stab)                        */
'*                      LINIE3 (fuer den 3. Stab)                        */
'*                      LINIE4 (fuer den 4. Stab)                        */
'*                                                                       */
'*      RETURN          Boolean ..... True = Funktion in Ordnung         */
'*                                    False = Fehler in Funktion         */
'*      PARAMETER:      keine                                            */
'*                                                                       */
'*-----------------------------------------------------------------------*/
Dim i                   As Integer                  'Allgemeine Zaehlvariable.
Dim iWinkel             As Integer                  'Allgemeine Zaehlvariable.
Dim lFarbe              As Long                     'Allgemeine Zaehlvariable.
Dim iPunkte             As Integer                  'Allgemeine Zaehlvariable.
Dim iWuerfe1            As Integer                  'Zufaellige Anzahl Drehungen Stab 1.
Dim iWuerfe2            As Integer                  'Zufaellige Anzahl Drehungen Stab 2.
Dim iWuerfe3            As Integer                  'Zufaellige Anzahl Drehungen Stab 3.
Dim iWuerfe4            As Integer                  'Zufaellige Anzahl Drehungen Stab 4.

On Error GoTo Fehler                                'Bei Fehler zur Fehlermarke.
WuerfelnS = True                                    'Returnwert = alles OK setzen.

ThisWorkbook.Activate                               'Aktuelle Excel-Datei festlegen.
Set oTabellenblattWuerfel = Sheets(TABELLENBLATT)   'Akt. Tabellenblatt festlegen.
oTabellenblattWuerfel.Select                        'Tabellenblatt auswaehlen.
Range("A1").Select                                  'Zelle A1 selektieren.
```

```
Randomize (Timer)                                   'Zufallsgenerator initialisieren.

oTabellenblattWuerfel.Cells(28, 17) = ""            'Wurfergebnis ruecksetzen.
iWuerfe1 = Int(MIN_WUERFE * Rnd() + (MAX_WUERFE - MIN_WUERFE))  'Zufaellige Anzahl der
iWuerfe2 = Int(MIN_WUERFE * Rnd() + (MAX_WUERFE - MIN_WUERFE))  'Drehungen fuer jeden
iWuerfe3 = Int(MIN_WUERFE * Rnd() + (MAX_WUERFE - MIN_WUERFE))  'der vier Staebe
iWuerfe4 = Int(MIN_WUERFE * Rnd() + (MAX_WUERFE - MIN_WUERFE))  'einzeln festlegen.
For i = 0 To MAX_WUERFE                             'Fuer alle moeglichen Drehungen:
  If i < iWuerfe1 Then                              'Stab 1 soll noch weiter drehen:
    If Int(2 * Rnd() + 1) = 1 Then lFarbe = FARBE_SW Else lFarbe = FARBE_WS
    If Int(2 * Rnd() + 1) = 1 Then iWinkel = Int(180 * Rnd() + 1) Else iWinkel = Int(180 *
Rnd() + 1) * (-1)
    oTabellenblattWuerfel.Shapes("LINIE1").Line.ForeColor.RGB = lFarbe 'Stabfarbe und
    oTabellenblattWuerfel.Shapes("LINIE1").Rotation = iWinkel 'Drehwinkel zuweisen.
  End If

  If i < iWuerfe2 Then                              'Stab 2 soll noch weiter drehen:
    If Int(2 * Rnd() + 1) = 1 Then lFarbe = FARBE_SW Else lFarbe = FARBE_WS
    If Int(2 * Rnd() + 1) = 1 Then iWinkel = Int(180 * Rnd() + 1) Else iWinkel = Int(180 *
Rnd() + 1) * (-1)
    oTabellenblattWuerfel.Shapes("LINIE2").Line.ForeColor.RGB = lFarbe 'Stabfarbe und
    oTabellenblattWuerfel.Shapes("LINIE2").Rotation = iWinkel 'Drehwinkel zuweisen.
  End If

  If i < iWuerfe3 Then                              'Stab 3 soll noch weiter drehen:
    If Int(2 * Rnd() + 1) = 1 Then lFarbe = FARBE_SW Else lFarbe = FARBE_WS
    If Int(2 * Rnd() + 1) = 1 Then iWinkel = Int(180 * Rnd() + 1) Else iWinkel = Int(180 *
Rnd() + 1) * (-1)
    oTabellenblattWuerfel.Shapes("LINIE3").Line.ForeColor.RGB = lFarbe 'Stabfarbe und
    oTabellenblattWuerfel.Shapes("LINIE3").Rotation = iWinkel 'Drehwinkel zuweisen.
  End If

  If i < iWuerfe4 Then                              'Stab 4 soll noch weiter drehen:
    If Int(2 * Rnd() + 1) = 1 Then lFarbe = FARBE_SW Else lFarbe = FARBE_WS
    If Int(2 * Rnd() + 1) = 1 Then iWinkel = Int(180 * Rnd() + 1) Else iWinkel = Int(180 *
Rnd() + 1) * (-1)
    oTabellenblattWuerfel.Shapes("LINIE4").Line.ForeColor.RGB = lFarbe 'Stabfarbe und
    oTabellenblattWuerfel.Shapes("LINIE4").Rotation = iWinkel 'Drehwinkel zuweisen.
  End If

  DoEvents                                          'Ansicht aktualisieren lassen.
  Sleep (DAUER_WUERFE / 2)                          'Warten auf naechste Drehung.
Next i

iPunkte = 0                                         'Wurfergebnis ruecksetzen und
If oTabellenblattWuerfel.Shapes("LINIE1").Line.ForeColor.RGB = FARBE_WS Then iPunkte = iPunkte
+ 1
If oTabellenblattWuerfel.Shapes("LINIE2").Line.ForeColor.RGB = FARBE_WS Then iPunkte = iPunkte
+ 1
If oTabellenblattWuerfel.Shapes("LINIE3").Line.ForeColor.RGB = FARBE_WS Then iPunkte = iPunkte
+ 1
If oTabellenblattWuerfel.Shapes("LINIE4").Line.ForeColor.RGB = FARBE_WS Then iPunkte = iPunkte
+ 1
If iPunkte = 0 Then iPunkte = 6                     'Punktezahl ermitteln.
oTabellenblattWuerfel.Cells(28, 17) = iPunkte       'Wurfergebnis ausgeben.

                                                    'Fehlerbehandlungsroutine:
Beenden:                                            '=========================
  Application.Cursor = xlDefault                    'Sanduhr ausblenden.
  Exit Function                                     'Funktion abbrechen.

Fehler:                                             'Fehlermarke - Fehlerbehandlung:
  WuerfelnS = False                                 'Returnwert = Fehler setzen.
  MsgBox Err.Description                            'Fehlermeldung ausgeben.
  Resume Beenden                                    'Bei Funktionsende weitermachen.
  Resume                                            'Nur fuer Testzwecke hinterlegt.
End Function                                        'Funktionsende.
```

```
Public Function WuerfelDrehen(stBereich As String) As Integer
'*************************************************************************/
'*      INTERNES UNTERPROGRAMM - INTERNE FUNKTION                        */
'*************************************************************************/
'*-----------------------------------------------------------------------*/
'*                                                                       */
'*      FUNKTION:       WuerfelDrehen                                    */
'*                                                                       */
'*      BESCHREIBUNG:   Simuliert das Wuerfeln fuer einen einzigen Wuerfel. */
'*                                                                       */
'*      RETURN          Integer ..... 1 - 6 (9) = gewuerfelte Zahl       */
'*                                    0 = Fehler in Funktion             */
'*                                                                       */
'*      PARAMETER:      String ..... Bereich wo sich Wuerfel befindet    */
'*                                                                       */
'*-----------------------------------------------------------------------*/
Dim i                   As Integer                  'Zufaellige Augenzahl des Wuerfels.
Dim j                   As Integer                  'Zufaellige Art der Darstellung.
Dim iZeile              As Long                     'Aktuelle Zeilennummer.
Dim iSpalte             As Integer                  'Aktuelle Spaltennummer.

On Error GoTo Fehler                                'Bei Fehler zur Fehlermarke.

iZeile = Range(stBereich).Row + 1                   'Zeile fuer Wuerfel auslesen.
iSpalte = Range(stBereich).Column + 1               'Spalte fuer Wuerfel auslesen.
i = Int(MAX_AUGEN * Rnd() + 1)                      'Zufaellige Augenzahl.
j = Int(2 * Rnd() + 1)                              'Zufaellige Darstellung.

Select Case i                                       'Je nach Augenzahl den entsprechenden
  Case 1                                            'Wuerfel darstellen.
    oTabellenblattWuerfel.Cells(iZeile + 0, iSpalte + 0) = ""
    oTabellenblattWuerfel.Cells(iZeile + 0, iSpalte + 1) = ""
    oTabellenblattWuerfel.Cells(iZeile + 0, iSpalte + 2) = ""
    oTabellenblattWuerfel.Cells(iZeile + 1, iSpalte + 0) = ""
    oTabellenblattWuerfel.Cells(iZeile + 1, iSpalte + 1) = Chr(ASC_AUGE)
    oTabellenblattWuerfel.Cells(iZeile + 1, iSpalte + 2) = ""
    oTabellenblattWuerfel.Cells(iZeile + 2, iSpalte + 0) = ""
    oTabellenblattWuerfel.Cells(iZeile + 2, iSpalte + 1) = ""
    oTabellenblattWuerfel.Cells(iZeile + 2, iSpalte + 2) = ""

  Case 2
    If j = 1 Then                                   'Je nach Darstellungsart:
      oTabellenblattWuerfel.Cells(iZeile + 0, iSpalte + 0) = ""
      oTabellenblattWuerfel.Cells(iZeile + 0, iSpalte + 1) = ""
      oTabellenblattWuerfel.Cells(iZeile + 0, iSpalte + 2) = Chr(ASC_AUGE)
      oTabellenblattWuerfel.Cells(iZeile + 1, iSpalte + 0) = ""
      oTabellenblattWuerfel.Cells(iZeile + 1, iSpalte + 1) = ""
      oTabellenblattWuerfel.Cells(iZeile + 1, iSpalte + 2) = ""
      oTabellenblattWuerfel.Cells(iZeile + 2, iSpalte + 0) = Chr(ASC_AUGE)
      oTabellenblattWuerfel.Cells(iZeile + 2, iSpalte + 1) = ""
      oTabellenblattWuerfel.Cells(iZeile + 2, iSpalte + 2) = ""
    Else
      oTabellenblattWuerfel.Cells(iZeile + 0, iSpalte + 0) = Chr(ASC_AUGE)
      oTabellenblattWuerfel.Cells(iZeile + 0, iSpalte + 1) = ""
      oTabellenblattWuerfel.Cells(iZeile + 0, iSpalte + 2) = ""
      oTabellenblattWuerfel.Cells(iZeile + 1, iSpalte + 0) = ""
      oTabellenblattWuerfel.Cells(iZeile + 1, iSpalte + 1) = ""
      oTabellenblattWuerfel.Cells(iZeile + 1, iSpalte + 2) = ""
      oTabellenblattWuerfel.Cells(iZeile + 2, iSpalte + 0) = ""
      oTabellenblattWuerfel.Cells(iZeile + 2, iSpalte + 1) = ""
      oTabellenblattWuerfel.Cells(iZeile + 2, iSpalte + 2) = Chr(ASC_AUGE)
    End If

  Case 3
    If j = 1 Then                                   'Je nach Darstellungsart:
      oTabellenblattWuerfel.Cells(iZeile + 0, iSpalte + 0) = Chr(ASC_AUGE)
      oTabellenblattWuerfel.Cells(iZeile + 0, iSpalte + 1) = ""
```

```
      oTabellenblattWuerfel.Cells(iZeile + 0, iSpalte + 2) = ""
      oTabellenblattWuerfel.Cells(iZeile + 1, iSpalte + 0) = ""
      oTabellenblattWuerfel.Cells(iZeile + 1, iSpalte + 1) = Chr(ASC_AUGE)
      oTabellenblattWuerfel.Cells(iZeile + 1, iSpalte + 2) = ""
      oTabellenblattWuerfel.Cells(iZeile + 2, iSpalte + 0) = ""
      oTabellenblattWuerfel.Cells(iZeile + 2, iSpalte + 1) = ""
      oTabellenblattWuerfel.Cells(iZeile + 2, iSpalte + 2) = Chr(ASC_AUGE)
    Else
      oTabellenblattWuerfel.Cells(iZeile + 0, iSpalte + 0) = ""
      oTabellenblattWuerfel.Cells(iZeile + 0, iSpalte + 1) = ""
      oTabellenblattWuerfel.Cells(iZeile + 0, iSpalte + 2) = Chr(ASC_AUGE)
      oTabellenblattWuerfel.Cells(iZeile + 1, iSpalte + 0) = ""
      oTabellenblattWuerfel.Cells(iZeile + 1, iSpalte + 1) = Chr(ASC_AUGE)
      oTabellenblattWuerfel.Cells(iZeile + 1, iSpalte + 2) = ""
      oTabellenblattWuerfel.Cells(iZeile + 2, iSpalte + 0) = Chr(ASC_AUGE)
      oTabellenblattWuerfel.Cells(iZeile + 2, iSpalte + 1) = ""
      oTabellenblattWuerfel.Cells(iZeile + 2, iSpalte + 2) = ""
    End If

  Case 4
    oTabellenblattWuerfel.Cells(iZeile + 0, iSpalte + 0) = Chr(ASC_AUGE)
    oTabellenblattWuerfel.Cells(iZeile + 0, iSpalte + 1) = ""
    oTabellenblattWuerfel.Cells(iZeile + 0, iSpalte + 2) = Chr(ASC_AUGE)
    oTabellenblattWuerfel.Cells(iZeile + 1, iSpalte + 0) = ""
    oTabellenblattWuerfel.Cells(iZeile + 1, iSpalte + 1) = ""
    oTabellenblattWuerfel.Cells(iZeile + 1, iSpalte + 2) = ""
    oTabellenblattWuerfel.Cells(iZeile + 2, iSpalte + 0) = Chr(ASC_AUGE)
    oTabellenblattWuerfel.Cells(iZeile + 2, iSpalte + 1) = ""
    oTabellenblattWuerfel.Cells(iZeile + 2, iSpalte + 2) = Chr(ASC_AUGE)

  Case 5
    oTabellenblattWuerfel.Cells(iZeile + 0, iSpalte + 0) = Chr(ASC_AUGE)
    oTabellenblattWuerfel.Cells(iZeile + 0, iSpalte + 1) = ""
    oTabellenblattWuerfel.Cells(iZeile + 0, iSpalte + 2) = Chr(ASC_AUGE)
    oTabellenblattWuerfel.Cells(iZeile + 1, iSpalte + 0) = ""
    oTabellenblattWuerfel.Cells(iZeile + 1, iSpalte + 1) = Chr(ASC_AUGE)
    oTabellenblattWuerfel.Cells(iZeile + 1, iSpalte + 2) = ""
    oTabellenblattWuerfel.Cells(iZeile + 2, iSpalte + 0) = Chr(ASC_AUGE)
    oTabellenblattWuerfel.Cells(iZeile + 2, iSpalte + 1) = ""
    oTabellenblattWuerfel.Cells(iZeile + 2, iSpalte + 2) = Chr(ASC_AUGE)

  Case 6
    If j = 1 Then                                   'Je nach Darstellungsart:
      oTabellenblattWuerfel.Cells(iZeile + 0, iSpalte + 0) = Chr(ASC_AUGE)
      oTabellenblattWuerfel.Cells(iZeile + 0, iSpalte + 1) = Chr(ASC_AUGE)
      oTabellenblattWuerfel.Cells(iZeile + 0, iSpalte + 2) = Chr(ASC_AUGE)
      oTabellenblattWuerfel.Cells(iZeile + 1, iSpalte + 0) = ""
      oTabellenblattWuerfel.Cells(iZeile + 1, iSpalte + 1) = ""
      oTabellenblattWuerfel.Cells(iZeile + 1, iSpalte + 2) = ""
      oTabellenblattWuerfel.Cells(iZeile + 2, iSpalte + 0) = Chr(ASC_AUGE)
      oTabellenblattWuerfel.Cells(iZeile + 2, iSpalte + 1) = Chr(ASC_AUGE)
      oTabellenblattWuerfel.Cells(iZeile + 2, iSpalte + 2) = Chr(ASC_AUGE)
    Else
      oTabellenblattWuerfel.Cells(iZeile + 0, iSpalte + 0) = Chr(ASC_AUGE)
      oTabellenblattWuerfel.Cells(iZeile + 0, iSpalte + 1) = ""
      oTabellenblattWuerfel.Cells(iZeile + 0, iSpalte + 2) = Chr(ASC_AUGE)
      oTabellenblattWuerfel.Cells(iZeile + 1, iSpalte + 0) = Chr(ASC_AUGE)
      oTabellenblattWuerfel.Cells(iZeile + 1, iSpalte + 1) = ""
      oTabellenblattWuerfel.Cells(iZeile + 1, iSpalte + 2) = Chr(ASC_AUGE)
      oTabellenblattWuerfel.Cells(iZeile + 2, iSpalte + 0) = Chr(ASC_AUGE)
      oTabellenblattWuerfel.Cells(iZeile + 2, iSpalte + 1) = ""
      oTabellenblattWuerfel.Cells(iZeile + 2, iSpalte + 2) = Chr(ASC_AUGE)
   End If

  Case 7
    If j = 1 Then                                   'Je nach Darstellungsart:
      oTabellenblattWuerfel.Cells(iZeile + 0, iSpalte + 0) = Chr(ASC_AUGE)
```

```
      oTabellenblattWuerfel.Cells(iZeile + 0, iSpalte + 1) = Chr(ASC_AUGE)
      oTabellenblattWuerfel.Cells(iZeile + 0, iSpalte + 2) = Chr(ASC_AUGE)
      oTabellenblattWuerfel.Cells(iZeile + 1, iSpalte + 0) = ""
      oTabellenblattWuerfel.Cells(iZeile + 1, iSpalte + 1) = Chr(ASC_AUGE)
      oTabellenblattWuerfel.Cells(iZeile + 1, iSpalte + 2) = ""
      oTabellenblattWuerfel.Cells(iZeile + 2, iSpalte + 0) = Chr(ASC_AUGE)
      oTabellenblattWuerfel.Cells(iZeile + 2, iSpalte + 1) = Chr(ASC_AUGE)
      oTabellenblattWuerfel.Cells(iZeile + 2, iSpalte + 2) = Chr(ASC_AUGE)
    Else
      oTabellenblattWuerfel.Cells(iZeile + 0, iSpalte + 0) = Chr(ASC_AUGE)
      oTabellenblattWuerfel.Cells(iZeile + 0, iSpalte + 1) = ""
      oTabellenblattWuerfel.Cells(iZeile + 0, iSpalte + 2) = Chr(ASC_AUGE)
      oTabellenblattWuerfel.Cells(iZeile + 1, iSpalte + 0) = Chr(ASC_AUGE)
      oTabellenblattWuerfel.Cells(iZeile + 1, iSpalte + 1) = Chr(ASC_AUGE)
      oTabellenblattWuerfel.Cells(iZeile + 1, iSpalte + 2) = Chr(ASC_AUGE)
      oTabellenblattWuerfel.Cells(iZeile + 2, iSpalte + 0) = Chr(ASC_AUGE)
      oTabellenblattWuerfel.Cells(iZeile + 2, iSpalte + 1) = ""
      oTabellenblattWuerfel.Cells(iZeile + 2, iSpalte + 2) = Chr(ASC_AUGE)
   End If

  Case 8
    oTabellenblattWuerfel.Cells(iZeile + 0, iSpalte + 0) = Chr(ASC_AUGE)
    oTabellenblattWuerfel.Cells(iZeile + 0, iSpalte + 1) = Chr(ASC_AUGE)
    oTabellenblattWuerfel.Cells(iZeile + 0, iSpalte + 2) = Chr(ASC_AUGE)
    oTabellenblattWuerfel.Cells(iZeile + 1, iSpalte + 0) = Chr(ASC_AUGE)
    oTabellenblattWuerfel.Cells(iZeile + 1, iSpalte + 1) = ""
    oTabellenblattWuerfel.Cells(iZeile + 1, iSpalte + 2) = Chr(ASC_AUGE)
    oTabellenblattWuerfel.Cells(iZeile + 2, iSpalte + 0) = Chr(ASC_AUGE)
    oTabellenblattWuerfel.Cells(iZeile + 2, iSpalte + 1) = Chr(ASC_AUGE)
    oTabellenblattWuerfel.Cells(iZeile + 2, iSpalte + 2) = Chr(ASC_AUGE)

  Case 9
    oTabellenblattWuerfel.Cells(iZeile + 0, iSpalte + 0) = Chr(ASC_AUGE)
    oTabellenblattWuerfel.Cells(iZeile + 0, iSpalte + 1) = Chr(ASC_AUGE)
    oTabellenblattWuerfel.Cells(iZeile + 0, iSpalte + 2) = Chr(ASC_AUGE)
    oTabellenblattWuerfel.Cells(iZeile + 1, iSpalte + 0) = Chr(ASC_AUGE)
    oTabellenblattWuerfel.Cells(iZeile + 1, iSpalte + 1) = Chr(ASC_AUGE)
    oTabellenblattWuerfel.Cells(iZeile + 1, iSpalte + 2) = Chr(ASC_AUGE)
    oTabellenblattWuerfel.Cells(iZeile + 2, iSpalte + 0) = Chr(ASC_AUGE)
    oTabellenblattWuerfel.Cells(iZeile + 2, iSpalte + 1) = Chr(ASC_AUGE)
    oTabellenblattWuerfel.Cells(iZeile + 2, iSpalte + 2) = Chr(ASC_AUGE)

End Select

oTabellenblattWuerfel.Cells(iZeile - 1, iSpalte - 1) = i 'Wurfergebis abspeichern
WuerfelDrehen = i                                        'und als Rueckgabewert festlegen.


                                                         'Fehlerbehandlungsroutine:
Beenden:                                                 '=========================
  Application.Cursor = xlDefault                         'Sanduhr ausblenden.
  Exit Function                                          'Funktion abbrechen.

Fehler:                                                  'Fehlermarke - Fehlerbehandlung:
  WuerfelDrehen = 0                                      'Returnwert = Fehler setzen.
  MsgBox Err.Description                                 'Fehlermeldung ausgeben.
  Resume Beenden                                         'Bei Funktionsende weitermachen.
  Resume                                                 'Nur fuer Testzwecke hinterlegt.
End Function                                             'Funktionsende.
```

```
Function WuerfelPositionieren()
'*****************************************************************************/
'*      INTERNES UNTERPROGRAMM - INTERNE FUNKTION                            */
'*****************************************************************************/
'*---------------------------------------------------------------------------*/
'*                                                                           */
'*      FUNKTION:       WuerfelPositionieren                                 */
'*                                                                           */
'*      BESCHREIBUNG:   Hilfsprogramm zum Positionieren der Pokerwuerfel.    */
'*                                                                           */
'*      RETURN          nichts                                               */
'*                                                                           */
'*      PARAMETER:      keine                                                */
'*                                                                           */
'*---------------------------------------------------------------------------*/
Dim oben
Dim links1
Dim links2
Dim links3
Dim links4
Dim links5

oben = 322
links1 = 168
links2 = 264
links3 = 360
links4 = 456
links5 = 552

Set oTabellenblattWuerfel = Sheets(TABELLENBLATT)

oTabellenblattWuerfel.Shapes("W1_1").Left = links1
oTabellenblattWuerfel.Shapes("W1_2").Left = links1
oTabellenblattWuerfel.Shapes("W1_3").Left = links1
oTabellenblattWuerfel.Shapes("W1_4").Left = links1
oTabellenblattWuerfel.Shapes("W1_5").Left = links1
oTabellenblattWuerfel.Shapes("W1_6").Left = links1

oTabellenblattWuerfel.Shapes("W2_1").Left = links2
oTabellenblattWuerfel.Shapes("W2_2").Left = links2
oTabellenblattWuerfel.Shapes("W2_3").Left = links2
oTabellenblattWuerfel.Shapes("W2_4").Left = links2
oTabellenblattWuerfel.Shapes("W2_5").Left = links2
oTabellenblattWuerfel.Shapes("W2_6").Left = links2

oTabellenblattWuerfel.Shapes("W3_1").Left = links3
oTabellenblattWuerfel.Shapes("W3_2").Left = links3
oTabellenblattWuerfel.Shapes("W3_3").Left = links3
oTabellenblattWuerfel.Shapes("W3_4").Left = links3
oTabellenblattWuerfel.Shapes("W3_5").Left = links3
oTabellenblattWuerfel.Shapes("W3_6").Left = links3

oTabellenblattWuerfel.Shapes("W4_1").Left = links4
oTabellenblattWuerfel.Shapes("W4_2").Left = links4
oTabellenblattWuerfel.Shapes("W4_3").Left = links4
oTabellenblattWuerfel.Shapes("W4_4").Left = links4
oTabellenblattWuerfel.Shapes("W4_5").Left = links4
oTabellenblattWuerfel.Shapes("W4_6").Left = links4

oTabellenblattWuerfel.Shapes("W5_1").Left = links5
oTabellenblattWuerfel.Shapes("W5_2").Left = links5
oTabellenblattWuerfel.Shapes("W5_3").Left = links5
oTabellenblattWuerfel.Shapes("W5_4").Left = links5
oTabellenblattWuerfel.Shapes("W5_5").Left = links5
oTabellenblattWuerfel.Shapes("W5_6").Left = links5
```

```
oTabellenblattWuerfel.Shapes("W1_1").Top = oben
oTabellenblattWuerfel.Shapes("W1_2").Top = oben
oTabellenblattWuerfel.Shapes("W1_3").Top = oben
oTabellenblattWuerfel.Shapes("W1_4").Top = oben
oTabellenblattWuerfel.Shapes("W1_5").Top = oben
oTabellenblattWuerfel.Shapes("W1_6").Top = oben

oTabellenblattWuerfel.Shapes("W2_1").Top = oben
oTabellenblattWuerfel.Shapes("W2_2").Top = oben
oTabellenblattWuerfel.Shapes("W2_3").Top = oben
oTabellenblattWuerfel.Shapes("W2_4").Top = oben
oTabellenblattWuerfel.Shapes("W2_5").Top = oben
oTabellenblattWuerfel.Shapes("W2_6").Top = oben

oTabellenblattWuerfel.Shapes("W3_1").Top = oben
oTabellenblattWuerfel.Shapes("W3_2").Top = oben
oTabellenblattWuerfel.Shapes("W3_3").Top = oben
oTabellenblattWuerfel.Shapes("W3_4").Top = oben
oTabellenblattWuerfel.Shapes("W3_5").Top = oben
oTabellenblattWuerfel.Shapes("W3_6").Top = oben

oTabellenblattWuerfel.Shapes("W4_1").Top = oben
oTabellenblattWuerfel.Shapes("W4_2").Top = oben
oTabellenblattWuerfel.Shapes("W4_3").Top = oben
oTabellenblattWuerfel.Shapes("W4_4").Top = oben
oTabellenblattWuerfel.Shapes("W4_5").Top = oben
oTabellenblattWuerfel.Shapes("W4_6").Top = oben

oTabellenblattWuerfel.Shapes("W5_1").Top = oben
oTabellenblattWuerfel.Shapes("W5_2").Top = oben
oTabellenblattWuerfel.Shapes("W5_3").Top = oben
oTabellenblattWuerfel.Shapes("W5_4").Top = oben
oTabellenblattWuerfel.Shapes("W5_5").Top = oben
oTabellenblattWuerfel.Shapes("W5_6").Top = oben


oTabellenblattWuerfel.Shapes("W1_6").ZOrder (msoBringToFront)
oTabellenblattWuerfel.Shapes("W2_6").ZOrder (msoBringToFront)
oTabellenblattWuerfel.Shapes("W3_6").ZOrder (msoBringToFront)
oTabellenblattWuerfel.Shapes("W4_6").ZOrder (msoBringToFront)
oTabellenblattWuerfel.Shapes("W5_6").ZOrder (msoBringToFront)
End Function



Function StaebePositionieren()
'*************************************************************************/
'*      INTERNES UNTERPROGRAMM - INTERNE FUNKTION                        */
'*************************************************************************/
'*-----------------------------------------------------------------------*/
'*                                                                       */
'*      FUNKTION:       StaebePositionieren                              */
'*                                                                       */
'*      BESCHREIBUNG:   Hilfsprogramm zum Positionieren der vier Staebe. */
'*                                                                       */
'*      RETURN          nichts                                           */
'*                                                                       */
'*      PARAMETER:      keine                                            */
'*                                                                       */
'*-----------------------------------------------------------------------*/
Dim oben
Dim links1
Dim links2
Dim links3
Dim links4
Dim Laenge
```

```
Dim links11
Dim links22
Dim links33
Dim links44
Dim Winkel

oben = 470
links1 = 180
links2 = 200
links3 = 220
links4 = 240
Laenge = 70

links11 = 480
links22 = 500
links33 = 520
links44 = 540
Winkel = 90

Set oTabellenblattWuerfel = Sheets(TABELLENBLATT)

oTabellenblattWuerfel.Shapes("LINIE1").Rotation = Winkel
oTabellenblattWuerfel.Shapes("LINIE1").Top = oben
oTabellenblattWuerfel.Shapes("LINIE1").Left = links1
oTabellenblattWuerfel.Shapes("LINIE1").Width = Laenge

oTabellenblattWuerfel.Shapes("LINIE2").Rotation = Winkel
oTabellenblattWuerfel.Shapes("LINIE2").Top = oben
oTabellenblattWuerfel.Shapes("LINIE2").Left = links2
oTabellenblattWuerfel.Shapes("LINIE2").Width = Laenge

oTabellenblattWuerfel.Shapes("LINIE3").Rotation = Winkel
oTabellenblattWuerfel.Shapes("LINIE3").Top = oben
oTabellenblattWuerfel.Shapes("LINIE3").Left = links3
oTabellenblattWuerfel.Shapes("LINIE3").Width = Laenge

oTabellenblattWuerfel.Shapes("LINIE4").Rotation = Winkel
oTabellenblattWuerfel.Shapes("LINIE4").Top = oben
oTabellenblattWuerfel.Shapes("LINIE4").Left = links4
oTabellenblattWuerfel.Shapes("LINIE4").Width = Laenge


oTabellenblattWuerfel.Shapes("LINIE11").Rotation = Winkel
oTabellenblattWuerfel.Shapes("LINIE11").Top = oben
oTabellenblattWuerfel.Shapes("LINIE11").Left = links11
oTabellenblattWuerfel.Shapes("LINIE11").Width = Laenge

oTabellenblattWuerfel.Shapes("LINIE22").Rotation = Winkel
oTabellenblattWuerfel.Shapes("LINIE22").Top = oben
oTabellenblattWuerfel.Shapes("LINIE22").Left = links22
oTabellenblattWuerfel.Shapes("LINIE22").Width = Laenge

oTabellenblattWuerfel.Shapes("LINIE33").Rotation = Winkel
oTabellenblattWuerfel.Shapes("LINIE33").Top = oben
oTabellenblattWuerfel.Shapes("LINIE33").Left = links33
oTabellenblattWuerfel.Shapes("LINIE33").Width = Laenge

oTabellenblattWuerfel.Shapes("LINIE44").Rotation = Winkel
oTabellenblattWuerfel.Shapes("LINIE44").Top = oben
oTabellenblattWuerfel.Shapes("LINIE44").Left = links44
oTabellenblattWuerfel.Shapes("LINIE44").Width = Laenge

Application.ScreenUpdating = True
End Function
```