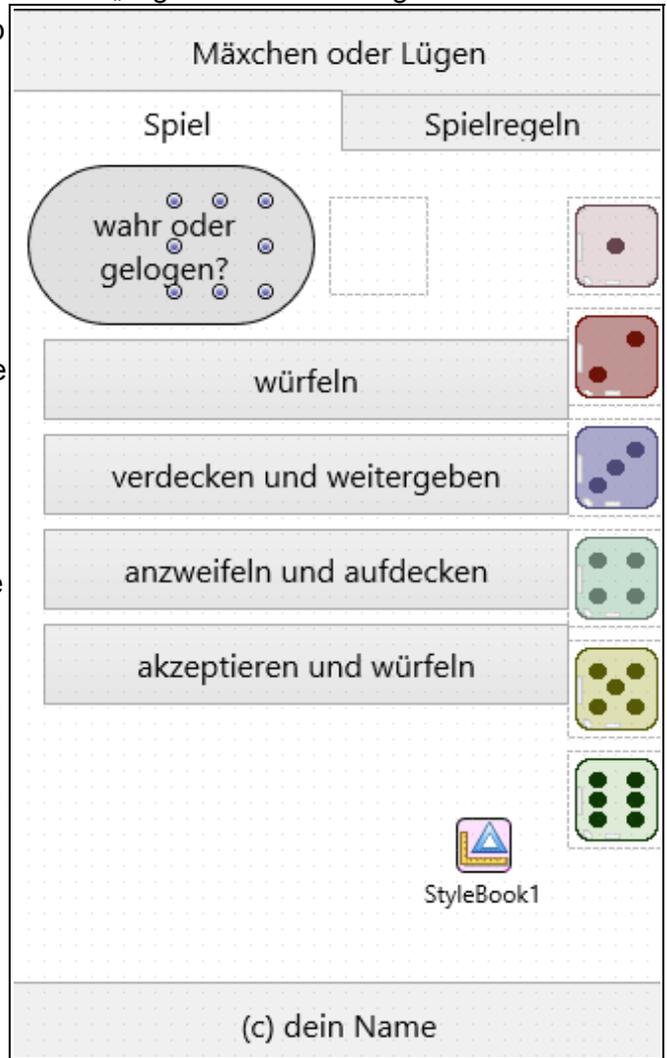


- A
1. Öffne ein neues Projekt in Delphi XE7. Datei-> neu-> geräteübergreifende Anwendung.
 2. Wähle "Leere Anwendung". Ein leeres Fenster erscheint. Links unten im Objektinspektor siehst du einige Eigenschaften des Fensters und kannst sie ändern. Wähle dort unter Height und Width die geeignete Höhe und Breite deines Formularfensters. Lege aus der Tool-Palette rechts unten folgende Objekte auf dem IFormular ab:
 3. Zwei Toolbar-Objekte. Setze bei der unteren Toolbar „align“ auf bottom. Lege auf diesen Toolbar-Objekten jeweils ein Label-Objekt ab und beschrifte es (unter Text im Objektinspektor) mit „Mäxchen oder Lügen“ und deinem Copyright-Vermerk.
 4. Ein TabControl-Element. Klicke doppelt darauf und öffne damit den Eintrags-Editor. Drücke zweimal „Eintrag hinzufügen“ und beschrifte die nun erscheinenden TabItem-Elemente mit „Spiel“ und „Spielregeln“. Setze abschließend „align“ auf client.
 5. Sechs Image-Objekte der Größe 50 x 50 am rechten Rand des Formulars. Erstelle 6 Würfelbilder des Typs „.png“ der Größe 50x50 und lade sie als MultiResBitmap in die entsprechenden Image-Objekte. Setze im MultiResBitmap-Editor „Transparente Farbe“ auf weiß. Damit erreichst du, dass der weiße Rand transparent wird.
 6. Außerdem zwei weitere leere Image-Objekte der Größe 50 x 50 im oberen Bereich des Formulars (in der Abbildung gestrichelt und verdeckt). Ändere ihre Namen in W1 und W2.
 7. Vier Button-Objekte mit der abgebildeten Beschriftung sowie ein RoundRect-Objekt mit einem Label der Beschriftung „wahr oder gelogen“. Um einen individuellen Stil im Aussehen zu erreichen, kann noch ein Stylbook-Objekt hinzugefügt werden.
 8. Gib in der OnShow-Prozedur des Formulars ein:



```

procedure TForm1.FormShow(Sender: TObject);
var i:integer;
begin
  for i:= 1 to 6 do
    tImage(FindComponent('Image'+IntToStr(i))).visible:= false;
  RoundRect1.Visible:=false;
  Button1.Enabled:= true; Button2.Enabled:= false;
  Button3.enabled:= false; Button4.Enabled:=false;
  AbdeckungEndPos:= round(W1.Position.X)-8; // Erfahrungswerte, können
  AbdeckungStartPos:= round(W1.Position.X)-140 // geändert werden
end;

```

Füge dazu als globale Variablen unter der Deklaration von Form1 die beiden Variablen AbdeckungStartPos und AbdeckungEndPos als integer-Variablen ein:

```
var
  Form1: TForm1;
  AbdeckungStartPos, AbdeckungEndPos: integer;
```

9. Trage in der OnClick-Methode des ersten Buttons (würfeln) ein:

```
procedure TForm1.Button1Click(Sender: TObject);
var WurfErgebnis, Spalte: integer;  Startzeit, Stoppzeit: tDateTime;
begin
  Startzeit:= Time;
  repeat
    for Spalte:=1 to 2 do begin
      WurfErgebnis:= random(6)+1;
      tImage(FindComponent('W'+IntToStr(Spalte))).MultiResBitmap:=
        tImage(FindComponent('Image'+IntToStr(WurfErgebnis))).MultiResBitmap;
      WEFeld[Spalte]:= WurfErgebnis;
    end;
    sleep(30);
    Application.ProcessMessages;
  until SecondOf(Time-Startzeit)>1;
  Button1.Enabled:= false; Button2.Enabled:= true;
  Button3.enabled:= false; Button4.Enabled:=false;
end;
```

10. Trage in der OnClick-Methode des zweiten Buttons (verdecken und weitergeben) ein:

```
procedure TForm1.Button2Click(Sender: TObject);
var i: integer;
begin
  RoundRect1.Visible:=true;
  with RoundRect1 do begin
    for i:= AbdeckungStartPos to AbdeckungEndPos do begin
      Position.X:= i ;
      if i mod 2 = 0 then
        Application.ProcessMessages;
    end;
    Position.X:= AbdeckungStartPos;
  end;
  Button1.Enabled:= false; Button2.Enabled:= false;
  Button3.enabled:= true; Button4.Enabled:= true;
end;
```

11. Trage in der OnClick-Methode des dritten Buttons (anzweifeln und aufdecken) ein:

```
procedure TForm1.Button3Click(Sender: TObject); // aufdecken
var i: integer;
begin
  with RoundRect1 do begin
    Position.X:= AbdeckungStartPos;
    for i:= AbdeckungEndPos downto AbdeckungStartPos do begin
      Position.X:= i ;
      //eventuell hier: sleep(SleepKonstante);
      if i mod 2 = 0 then
        Application.ProcessMessages;
    end;
  end;
  Button1.Enabled:= true; Button2.Enabled:= false;
  Button3.enabled:= false; Button4.Enabled:= false;
```

```
end;
```

12. Trage in der OnClick-Methode des vierten Buttons (akzeptieren und würfeln) ein:

```
procedure TForm1.Button4Click(Sender: TObject);  
begin  
    RoundRect1.Visible:=false;  
    Button1Click(Sender);  
    Button1.Enabled:= false; Button2.Enabled:= true;  
    Button3.enabled:= false; Button4.Enabled:= false;  
end;
```

13. Nun können abschließend die Symbole erstellt werden und es kann die App unter den entsprechenden Betriebssystemen getestet werden.