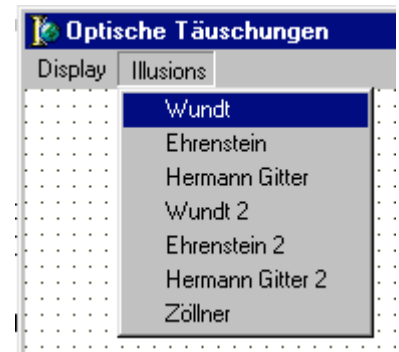


Projekt Optische Täuschungen

Vorarbeiten:

- Kopieren Sie den **Ordner Illusions** vom Server in ihr Arbeitsverzeichnis.
- Öffnen Sie die **Projektdatei illusions.dpr**.



Arbeitsauftrag:

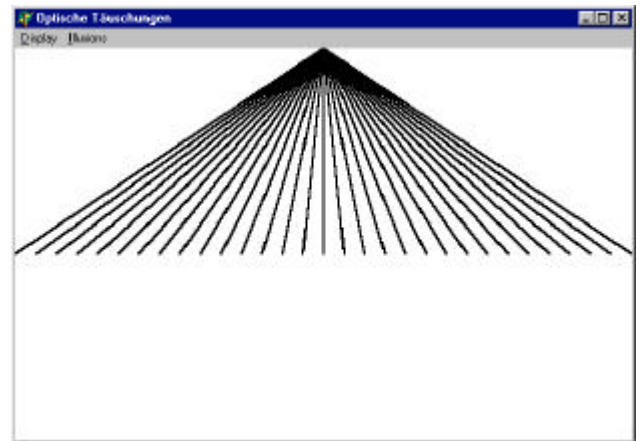
1. Schreiben Sie die Ereignisbehandlungsroutine WundtClick (Doppelklick auf den Menüpunkt Wundt; siehe rechts):

```
procedure TForm1.WundtClick(Sender: TObject);
```

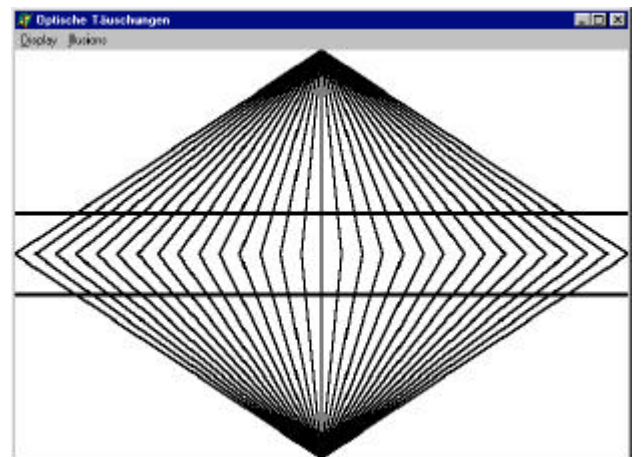
- Initialisieren Sie zuerst die Eigenschaften des Zeichenstiftes (Pen):

```
Canvas.Pen.Width := 2;  
Canvas.Pen.Color := clBlack;
```

- Zeichnen Sie mit Hilfe der Methoden
`Canvas.MoveTo(x,y);`
`Canvas.LineTo(x,y);`
 den rechts abgebildeten Strahlenkranz.
- Schreiben Sie eine **For-To-Do Schleife**, die den abgebildeten Strahlenkranz in der **oberen Hälfte des Fensters** erzeugt.
- Achten Sie auf die Deklaration der Laufvariablen und eine saubere Formatierung.



- Ergänzen Sie die Anweisung des Schleifenkörpers so, daß jeder aufgefächerte Strahl **im Mittelpunkt des unteren Fensterrandes endet** (siehe unten).
- Zeichnen Sie nun die **beiden Parallelen**. Diese sollen
 - mit der Stiftdicke (**Pen.Width**) 4
 - der Farbe (**Pen.Color**) **clRed**
 - **40 Pixel** oberhalb und unterhalb der Fenstermitte erscheinen.



Klammern Sie die einzelnen Canvas-Eigenschaften und Methoden in einer **With Canvas do** Verbundanweisung:

```
With Canvas do  
begin
```

Gratulation !!

2. Schreiben Sie die Ereignisbehandlungsroutine (Doppelklick auf Menüpunkt):

```
procedure TForm1.EhrensteinClick(Sender: TObject);
```

- Initialisieren Sie auch hier die Eigenschaften des Zeichenstiftes (Pen):

```
Canvas.Pen.Width := 3;
Canvas.Pen.Color := clBlack;
Canvas.Brush.Style := bsClear;
```

Damit die einzelnen Kreise sich nicht übermalen, müssen Sie dem Pinsel (Brush) die Stilkonstante **bsClear** zuweisen. Diese selektiert ein transparentes Füllmuster.

Schreiben Sie eine **For-To-Do Schleife**, die das Fenster mit **konzentrischen Kreisen** füllt.

- Verwenden Sie die Canvas-Methode **Ellipse(x-r, y-r, x+r, y+r);** mit geeigneter Wahl der Mittelpunktskoordinaten (300,200). Der **Radius r** soll jeweils um **20 Pixel** anwachsen.
- Achten Sie auf die Deklaration der Laufvariablen und eine saubere Formatierung.



Zeichnen Sie nun das Quadrat mit der Methode **Rectangle(x-r, y-r, x+r, y+r);**

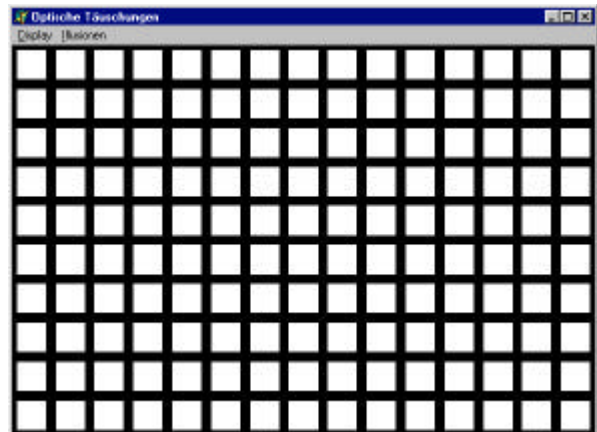
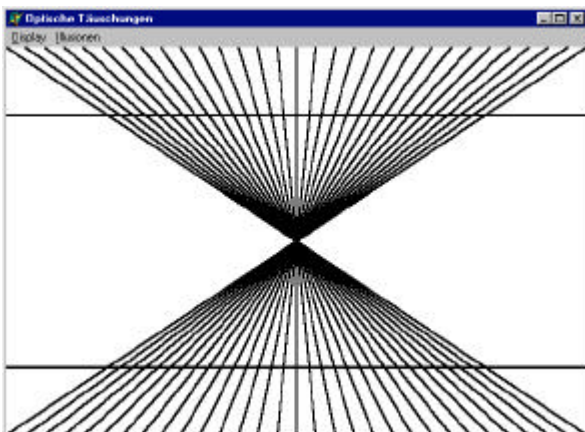
Wählen Sie folgenden **Stifteigenschaften**:

- Stiftdicke (Pen.Width) 4
- Farbe (Pen.Color) Rot (clRed) und eine **Kantenlänge von 300 Pixel**.

Auch diese Folge einzelner Canvas-Eigenschaften und Methoden sollten Sie mit einer **With Canvas do** Verbundanweisung klammern.



3. Realisieren Sie die Variation **Wundt 2 (links)** und das **Hermann Gitter (rechts)**:



4. Schreiben Sie die Ereignisbehandlungsroutine

```
procedure TForm1.HermannGitter2Click(Sender: TObject);
```

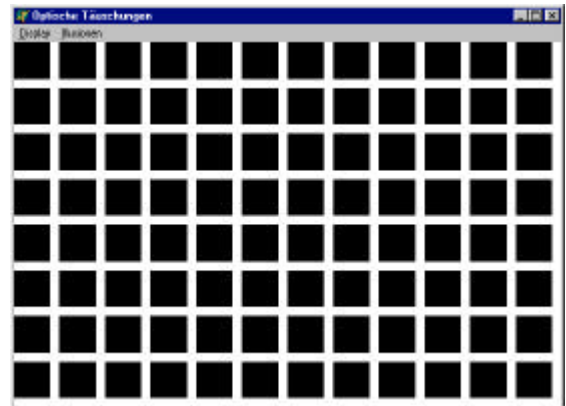
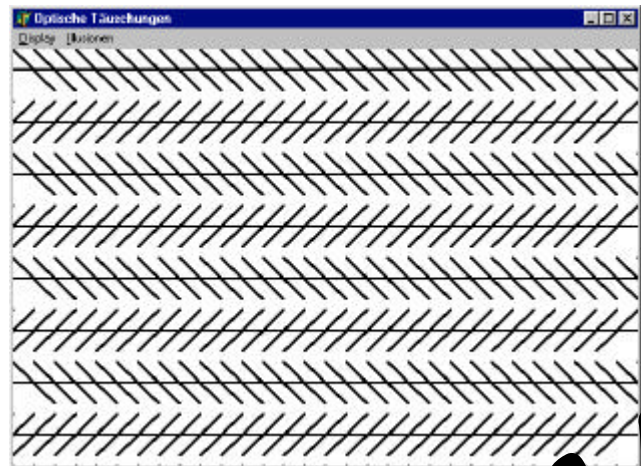
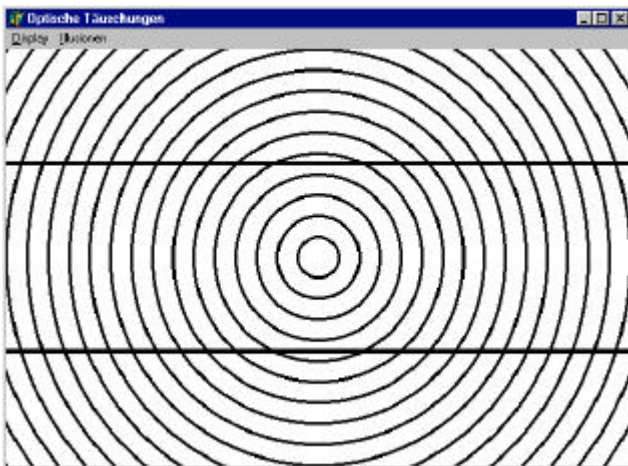
Initialisieren Sie auch hier die Eigenschaften des Zeichenstiftes und des Pinsels:

```
With Canvas do
begin
  Pen.Width := 1;
  Pen.Color := clBlack;
  Brush.Style := bsSolid;
  Brush.Color := clBlack;
end;
```

Damit die einzelnen Quadrate schwarz ausgefüllt werden, müssen Sie den Pinseleigenschaften **Brush.Style** bzw. **Brush.Color** die Füllkonstante **bsSolid** bzw. die Farbkonstante **clBlack** zuweisen.

Das Hermann Gitter besteht aus einem regelmäßigen weißen Gitter auf schwarzem Hintergrund. Da die Farbe des Fensters (Form1) unverändert d.h. weiß bleiben soll, stellen wir das Gitter durch eine **Anordnung schwarzer Quadrate** (8 Zeilen und 12 Spalten) dar.

- Schreiben Sie **zwei ineinander verschachtelte Schleifen**, bei denen die eine Laufvariable die Spaltenzahl, die andere Laufvariable die Zeilenzahl durchläuft.

5. Realisieren Sie die **Variation der Ehrenstein Illusion** und die **Zöllner Illusion** (rechts)

viel Spaß!