Optische Täuschungen – Weitere Übungen

Bei den folgenden beiden Bildern wird die optische Täuschung durch eine Nachbildwirkung auf der Netzhaut erzeugt.

Es handelt sich hier um sog. „Hermann-Gitter“.

Bild 1:

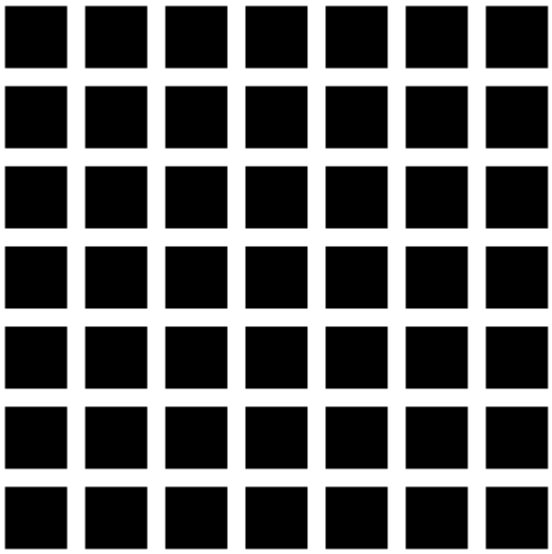

Bild: Screenshot von Ausführung des Programms „hermann_gitter_1“ (Eisenmann)

1. *Beschreibe, wie das Bild entsteht und was man zusätzlich sieht.*
2. *Speichere ein neues Programm unter „hermann_gitter_1“ ab.*
3. *Schreibe zunächst den Rahmen der setup-Methode, gib die Größe des Fensters (500x500) an und lasse den Hintergrund weiß „färben“.*
4. *Deklariere zwei globale Variablen: eine für die Seitenlänge der Quadrate und eine für den Abstand zwischen den Quadraten.*
5. *Schreibe eine Methode quadrate(), die die optische Täuschung zeichnet. Nutze dabei eine verschachtelte Zählschleife.*

Wenn du Hilfe brauchst, nutze die ausgelegten Hilfekarten.

Wenn du fertig bist, vergleiche dein Ergebnis mit der ausliegenden Lösung. Hole dir dann das nächste Blatt mit Bild 2.

Bild 2:

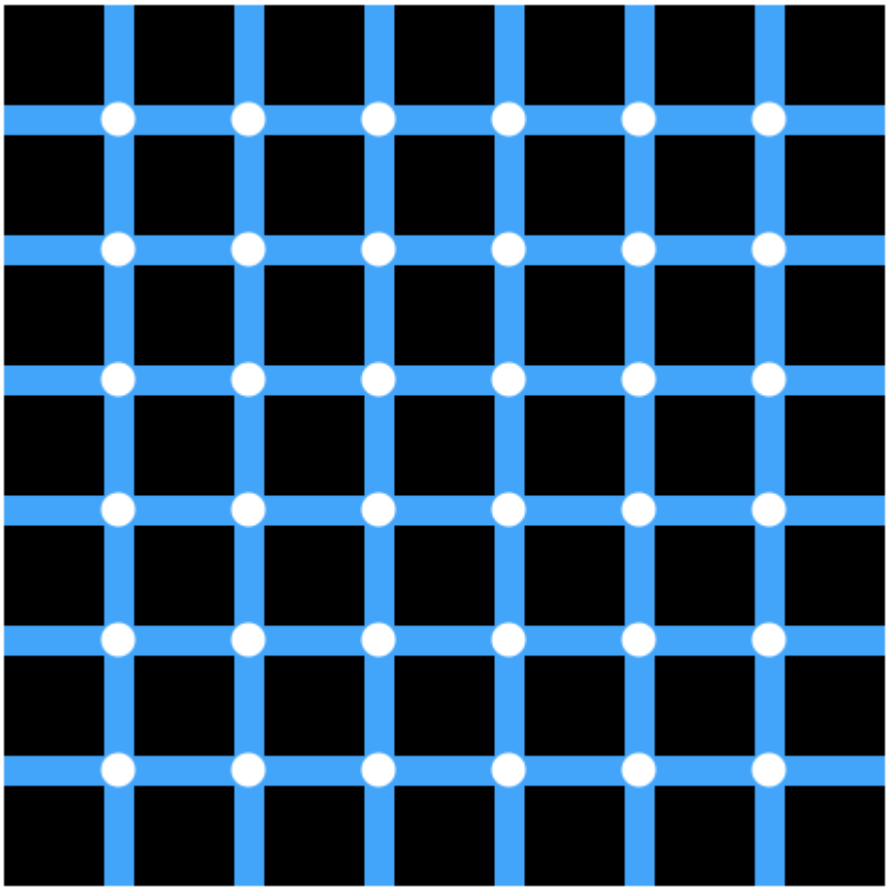

Bild: Screenshot von Ausführung des Programms „hermann_gitter_2“ (Eisenmann)

1. *Beschreibe, wie das Bild entsteht und was man zusätzlich sieht.*
2. *Speichere ein neues Programm unter „hermann_gitter_2“ ab.*
3. *Schreibe zunächst den Rahmen der setup-Methode, gib die Größe des Fensters (500x500) an und lasse den Hintergrund hellblau „färben“.*
4. *Deklariere wieder sinnvolle globale Variablen.*
5. *Schreibe eine Methode, die die optische Täuschung zeichnet. Nutze dazu verschachtelte Zählschleifen. Du kannst hier auch zwei Methoden schreiben: eine für die schwarzen Quadrate und eine für die weißen Kreise.*

Wenn du Hilfe brauchst, nutze die ausgelegten Hilfekarten.

Wenn du fertig bist, vergleiche dein Ergebnis mit der ausliegenden Lösung.

HILFEKARTEN ZU BILD 1

Tipp 1

Da der Hintergrund weiß ist, müssen nur die schwarzen Quadrate gezeichnet werden.

Überlege dir zunächst, wie du **eine** Reihe der Quadrate zeichnen kannst.

Tipp 2

In einer Reihe werden sieben gleiche Quadrate gezeichnet. Du kannst hier eine Zählschleife verwenden.

Versuche, die Zählschleife für die erste Reihe der Quadrate zu schreiben.

Tipp 3

Hat es geklappt? Dann schreibe jetzt die Zählschleife für die zweite Reihe. (Nächster Tipp: **6**)

Es hat nicht geklappt? Dann lies hier weiter:

Die Veränderung ist nur die Position der linken oberen Ecke der Quadrate.

Und hier bleibt die y-Koordinate gleich, nur die x-Koordinate verändert sich um die Seitenlänge eines Quadrates und den Abstand zwischen zwei Quadraten.

Tipp 4

y ist ein fester Wert, z.B. 30

x berechnet sich durch den Startwert (z.B. auch 30) + i * (seitenlaenge + abstand)

Ersetze dabei „seitenlaenge“ und „abstand“ durch den Namen deiner Variablen, die du eingeführt hast.

Tipp 5

```
for (int i=0; i<=6; i++) {
  rect(30 + i*(s + a), 30, s, s);
}
```

(s ist hier die Variable für die Seitenlänge, a für den Abstand)

Schreibe jetzt die Zählschleife für die zweite Reihe.

Tipp 6

Hast du es geschafft? Dann schreibe die Zählschleife für die dritte Reihe und überlege dir dabei, wie man alle Reihen mit einer weiteren Zählschleife programmieren könnte. (Nächster Tipp: **9**)

Hat es noch nicht geklappt? Dann lies hier weiter:

Was verändert sich in der zweiten Reihe im Vergleich zur ersten?

Was bleibt gleich?

Tipp 7

Es bleibt alles gleich bis auf die y-Koordinate.

Hier wird zum Startwert (30) einmal die Seitenlänge und einmal der Abstand addiert.

Tipp 8

Die Schleife für die zweite Reihe sieht folgendermaßen aus:

```
for (int i=0; i<=6; i++) {
  rect(30 + i*(s + a), 30 + s + a, s, s);
}
```

Schreibe jetzt die Zählschleife für die dritte Reihe und überlege dir dabei, wie man alle Reihen in einer weiteren Zählschleife programmieren könnte.

Tipp 9

Mögliche Lösung für die dritte Reihe:

```
for (int i=0; i<=6; i++) {
  rect(30 + i*(s + a), 30 + 2* (s + a), s, s);
}
```

Tipp 10

Hast du herausgefunden, was sich von Reihe zu Reihe ändert und wie du das mit einer weiteren Zählschleife realisieren kannst? - Dann vergleiche dein Ergebnis mit der Lösung.

Du hast es noch nicht geschafft? Dann schau dir mal die y-Koordinaten an:

30 30 + s + a 30 + 2 * (s + a) 30 + 3 * (s + a), usw.

Wie könnte man das allgemein mit einer neuen Zählvariable j beschreiben?

Tipp 11

Startet j bei 0, dann kann man die y-Koordinate durch 30 + j * (s + a) beschreiben.

Schreibe jetzt die zweite Zählschleife um die erste herum und vergleiche dein Ergebnis mit der Lösung.

Lösung

```
// globale Variablen
int s = 30; // Seitenlaenge Quadrat
int a = 10; // Abstand Quadrate

// Methode, die optische Täuschung zeichnet
void quadrate() {
  fill(0); // Füllfarbe schwarz
  for (int j=0; j<=6; j++) {
    for (int i=0; i<=6; i++) {
      rect(30+i*(s+a), 30+j*(s+a), s, s);
    }
  }
}

// setup-Methode
void setup() {
  size(500, 500);
  background(255); // Hintergrund weiß
  quadrate();
}
```

HILFEKARTEN ZU BILD 2

Tipp 1

Um die schwarzen Quadrate zu zeichnen, kannst du die verschachtelte Zählschleife aus dem Programm zu **Bild 1** nutzen.

Überlege dir, was dann noch fehlt.

Tipp 2

An den Kreuzungspunkten müssen kleine weiße Kreise gezeichnet werden.

Überlege dir, wo die Mittelpunkte liegen und wie groß etwa der Durchmesser sein muss.

Tipp 3

(Variable für die Seitenlänge: s, Variable für den Abstand: a, Variable für Abstand zum Rand: start)

Der x-Koordinate des Mittelpunktes des ersten Kreises ist start + s + a / 2. (Achtung: Das Zeichen / steht für „geteilt durch“)

Die y-Koordinate ist start + s + a / 2.

Der Durchmesser (also Breite und Höhe der Ellipse) ist ein bisschen größer als der Abstand a.

Was verändert sich nach rechts? Was nach unten?

Tipp 4

Nach rechts verändert sich die x-Koordinate um s + a.

Nach unten die y-Koordinate um s + a.

Der Durchmesser bleibt gleich.

Schreibe die Zählschleife für die obere Reihe.

Tipp 5

Eine mögliche Lösung:

```
for (int i=0; i<=5; i++) {
  ellipse(start+s+a/2+i*(s + a), start+s+a/2, a+2, a+2);
}
```

Schreibe jetzt die Zählschleife, die alle Reihen zeichnet.

Tipp 6

Hast du es geschafft? Dann vergleiche dein Ergebnis mit der Lösung.

Hat es noch nicht geklappt?

Dann schreibe zunächst eine Zählschleife für die zweite Reihe, die dritte, … So lange, bis du erkennst, was sich verändert.

Tipp 7

Die Veränderung ist in der y-Koordinate.

Mit einer neuen Zählvariable j, die bei 0 startet, lässt sich die y-Koordinate durch start + s + a/2 + j * (s + a) beschreiben.

Schreibe jetzt die verschachtelte Zählschleife fertig. Vergleiche dann dein Ergebnis mit der Lösung.

Lösung

```
// globale Variablen
int s = 50; // Seitenlaenge Quadrat
int a = 15; // Abstand Quadrate
int start = 30; // Abstand zum Rand

// Methode, die schwarze Quadrate zeichnet
void quadrate() {
  fill(0);
  for (int i=0; i<=6; i++) {
    for (int j=0; j<=6; j++) {
      rect(start+j*(s+a), start+i*(s+a), s, s);
    }
  }
}

// Methode, die weiße Kreise zeichnet
void kreise() {
  fill(255);
  noStroke(); // keine Umrandung
  for (int j=0; j<6; j++) {
    for (int i=0; i<6; i++) {
      ellipse(start+s+a/2+(s+a)*i, start+s+a/2+(s+a)*j, a+2, a+2);
    }
  }
}

// setup-Methode
void setup() {
  size(500, 500);
  background(66, 165, 250);  // Hintergrundfarbe hellblau
  quadrate();
  kreise();
}
```