Lektion 5: Turtle-Geometrie im Koordinatensystem

Bearbeitet von Kristel Jenkel, Britta Schreiber, Angela Klewinghaus und Karoline Selbach

Zu Beginn einer Prozedur steht die Turtle in der Mitte des Bildschirms. Dieser Punkt ist daher unter allen Punkten des Bildschirms hervorgehoben: Er ist Ursprung eines Koordinatensystems. Die X-Achse verläuft wie üblich horizontal, die Y-Achse vertikal. Die Koordinaten von Punkten auf dem Bildschirm werden in Turtle-Schritten angegeben, d. h. die X- und Y-Koordinaten der Punkte liegen im Bereich zwischen – 500 und + 500.

Für dieses Koordinatensystem gibt es in LOGO bestimmte Befehle. Bereits in Lektion 1 lernten wir den Befehl *home* kennen: Die Turtle geht von der augenblicklichen Position in den Ursprung des Koordinatensystems (mit Blick in Richtung der positiven Y-Achse).

Es folgen Befehle, mit deren Hilfe man Ort und Blickrichtung der Turtle herausfinden kann:

xcor - Der Rechner gibt an, welche X-Koordinate der Punkt hat, in dem sich die Turtle augenblicklich befindet. (Wenn man nur *xcor* eingibt, kommt die „Fehlermeldung“ „*You don't say what to do with ...*“.)

ycor - Der Rechner gibt an, welche Y-Koordinate der Punkt hat, in dem sich die Turtle augenblicklich befindet.

setx { x-Koordinate } - Die Turtle wird horizontal (parallel zur X-Achse) bis zu einem Punkt mit der angegebenen X-Koordinate bewegt.

Beispiel: *setx 100*
Die Turtle bewegt sich vom augenblicklichen Punkt zum Punkt (100 ; b).

setx { x-Koordinate } - Die Turtle wird vertikal (parallel zur Y-Achse) bis zu einem Punkt mit der angegebenen Y-Koordinate bewegt.

Beispiel: *sety -200*
Die Turtle bewegt sich vom augenblicklichen Punkt zum Punkt (a ; -200).

setxy { x-Koordinate y-Koordinate } - Die Turtle wird zu dem Punkt mit den angegebenen X- und Y-Koordinaten bewegt.

Beispiel: *setxy 50 –100*
Die Turtle bewegt sich vom augenblicklichen Punkt zum Punkt (50 ; -100).

setpos [{ x-Koordinate y-Koordinate } *]* - Die Turtle wird zu dem Punkt mit den angegebenen X- und Y-Koordinaten bewegt.

Beispiel: *setpos [50 –100]*
Die Turtle bewegt sich vom augenblicklichen Punkt zum Punkt (50 ; -100). Man beachte die eckigen Klammern!

pos - Der Rechner gibt an, welche X-Koordinate und welche Y-Koordinate der Punkt hat, in dem sich die Turtle augenblicklich befindet. Die Ausgabe erfolgt im Listen-Format, d. h. mit eckigen Klammern.

heading - Der Rechner gibt an, welcher Winkel zwischen der momentanen Blickrichtung der Turtle und der positiven Y-Achse vorliegt (gemessen im Uhrzeigersinn).

setheading { Winkel } – Die Turtle dreht sich im Uhrzeigersinn so, dass zwischen der positiven Y-Achse und der Blickrichtung der eingegebene Winkel vorliegt. Bei 0° blickt die Turtle nach oben, bei 90° nach rechts, bei –90° oder bei 270° nach links, bei 180° nach unten.

Beispiel: *setheading 30* – Die Turtle dreht sich so, dass zwischen der Richtung der Y-Achse und der Blickrichtung ein Winkel von 30° vorliegt – egal, in welche Richtung die Turtle vorher geblickt hat.

towards [{ x-Koordinate y-Koordinate } *]* - Der Rechner gibt an, welcher Winkel zwischen der Richtung der positiven Y-Achse und der Verbindungslinie von der augenblicklichen Turtleposition und dem Punkt (a ; b) vorliegt.

Beispiel: *towards [100 100]* gibt den Winkel 45°,
towards [100 0] den Winkel 90° zurück.

Einige der Befehle erscheinen vielleicht kompliziert und überflüssig; es gibt aber Situationen, in denen man nur mit Hilfe von schwierigen Rechnungen weiterkäme, wenn man diese Befehle nicht hätte. Wichtig ist auch, dass bei einigen Befehlen die Blickrichtung der Turtle keine Rolle spielt.

Hinzu kommt noch die Möglichkeit, die Befehle zu kombinieren –

Beispiele:
setx xcor + 50 bewirkt, dass die Turtle ausgehend von ihrer jetzigen Lage die X-Koordinate um 50 erhöht.
setheading towards [300 400] bewirkt, dass die Turtle auf den Punkt (300 ; 400) blickt – egal, wohin sie vorher geblickt hatte.

Problem 5.1

Schreibe eine Prozedur, mit deren Hilfe der Buchstabe N im Format: 60 x 100 Turtleschritte auf den Bildschirm gemalt werden kann - egal, wo die Turtle am Anfang steht. Anschließend soll die Turtle 30 Turtleschritte rechts neben dem Buchstaben stehen.

Lösung

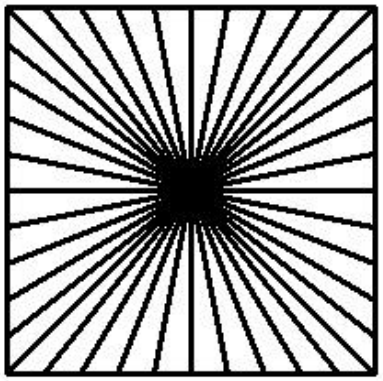
Wenn wir unabhängig von der augenblicklichen Position und Blickrichtung der Turtle zeichnen wollen, gehen wir am besten wie folgt vor:

```
to n
sety ycor+100
setxy xcor+60 ycor-100
sety ycor+100
pu setxy xcor+30 ycor-100 pd
end
```

Statt des ersten bzw. dritten Befehls hätten wir auch *fd 100* schreiben können, wenn die Turtle nach oben geblickt hätte. Den letzten Befehl hätten wir z. B. auch durch *pu bk 100 rt 90 fd 30 lt 90 pd* ersetzen können, da die Turtlespur nicht sichtbar ist. Für den zweiten Befehl gibt es aber keinen Ersatz.

Problem 5.2

Zeichne ein Quadrat der Seitenlänge 200. Betrachte – ausgehend von einem Eckpunkt – alle Punkte auf den Seiten des Quadrats, die den Abstand 20 Einheiten voneinander haben. Schreibe eine Prozedur, mit deren Hilfe je zwei gegenüberliegende Punkte so miteinander verbunden werden, dass eine punktsymmetrische Figur entsteht.



Lösung

Es bietet sich an, die Figur punktsymmetrisch zur Mitte des Bildschirms zu zeichnen. Deshalb springen wir zunächst zum Punkt (-100; -100): *pu setxy -100 -100 pd*

Nun zeichnen wir das umrahmende Quadrat. Danach steht die Turtle in der linken unteren Ecke des Quadrats. Dieser Punkt soll mit dem rechten oberen Eckpunkt des Quadrats verbunden werden. Dazu benutzen wir den Befehl: *setxy -xcor -ycor*

Die Koordinaten des linken unteren und rechten oberen Eckpunktes unterscheiden sich nämlich nur im Vorzeichen. Die Verschiebung um 20 Einheiten nach unten erreichen wir durch *sety ycor - 20*

Danach erfolgt wieder die Verbindung zum Spiegelpunkt mit Hilfe von *setxy -xcor -ycor* und wieder eine Bewegung in vertikaler Richtung, diesmal mit *sety ycor + 20*.

Diese Befehlsfolge muss fünfmal wiederholt werden, bis die Turtle sich in der linken oberen Ecke des Quadrats befindet.

Nun müssen Bewegungen auf dem oberen und unteren Quadratrand in horizontaler Richtung vorgenommen werden:
setx xcor - 20 bzw. *setx xcor + 20*

Die gesamte Prozedur lautet daher:

```
to symm
pu  setxy -100 -100  pd
quadrat 200
repeat 5 [ setxy -xcor -ycor  sety ycor-20  setxy -xcor -ycor  sety ycor+20 ]
repeat 5 [ setxy -xcor -ycor  setx xcor-20  setxy -xcor -ycor  setx xcor+20 ]
end
```

Probiere aus ...

a) *Gib die Befehlsfolge repeat 90 [fd 1 rt 2] ein. Was wird gezeichnet? Was kann man mit Hilfe des xcor - Befehls herausfinden?*
Verfahre analog für die Befehlsfolgen: repeat 60 [fd 1 rt 3]; repeat 45 [fd 1 rt 4]; repeat 36 [fd 1 rt 5] ; repeat 30 [fd 1 rt 6]

b) Zeichne ein Quadrat so, dass die beiden Diagonalen sich im Ursprung des Koordinatensystems schneiden. Die Diagonalen des Quadrates unterteilen die Fläche in vier Sektoren. Mit der Befehlsfolge *repeat 1000 [rt 10]* wird die Turtle fortlaufend gedreht. Versuche, mit dem Befehl HALT im rechten Befehlsfeld des Commander die Turtle in einem bestimmten Sektor des Quadrates anzuhalten. Prüfe im Zweifelsfalle die Richtung des Turtles mit Hilfe des Befehls *heading*. Das Spiel wird spannender, wenn die Turtle versteckt wird.

c) Bestimme die Länge der Diagonalen in einem Quadrat der Seitenlänge 100 mit Hilfe einer geeigneten Abfrage.

d) Die Turtle bewege sich in 20 x 5 Schritten vom Punkt (100; 0) auf den Ursprung zu. Lies jeweils ab, unter welchem Winkel sie den Punkt (0; 100) auf der Y-Achse sieht.

e) Von einem Dreieck seien die Länge der Seiten c und a, sowie die Größe des Winkels ß bekannt. Wie kann man mit Hilfe der neuen Befehle die Größe des Winkels β bestimmen? Führe dies konkret aus für c = 200, a = 120 und ß = 75°. Kann man mit Hilfe der Turtle auch die Länge der Seite b herausbekommen?

f) Zeichne ein rechtwinkliges Dreieck, bei dem die beiden kürzeren Seiten die Länge 80 und 100 (120 und 150) Einheiten haben. Benutze geeignete Turtlebefehle, um die beiden anderen Winkel zu bestimmen.

Überlege ...

a) Mit den Befehlen *rt 10* bzw. *lt 10* wird die Turtle gegenüber seiner vorherigen Blickrichtung um 10 Grad gedreht. Mit welchen neu gelernten Befehlen kann man dies auch erreichen?

b) Vom Ursprung aus gehe die Turtle um 100 Schritte unter einem bestimmten Winkel nach oben (nach unten, nach rechts, nach links). Man stelle sich vor, dass die Turtle dort an einer (imaginären) Wand reflektiert wird (d. h. die imaginäre Wand soll parallel zum oberen, unteren, rechten und linken Rand verlaufen). Überlege geeignete Befehle, mit deren Hilfe man aus dem vorliegenden Kurs den neuen Kurs bestimmen kann.

Übungen

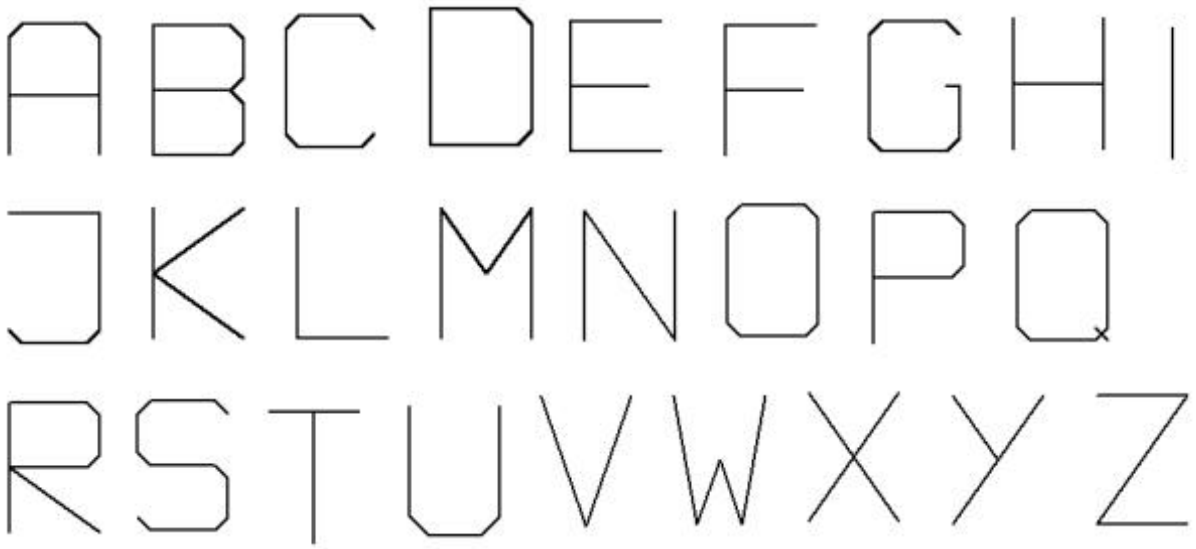
Ü5.1
Schreibe eine Prozedur *sprung :z1 :z2* , mit deren Hilfe man an eine beliebige Stelle des Bildschirms springen kann.

Ü5.2
Schreibe eine Prozedur, mit deren Hilfe man die Buchstaben H, L, M, Z, X auf den Bildschirm malen kann.

Ü5.3

a) Aufgabe als Gruppenarbeit: Schreibe Prozeduren für alle Buchstaben des Alphabets entsprechend dem folgenden Muster.

b) Führe einen Parameter ein, so dass die Schriftgröße verändert werden kann.

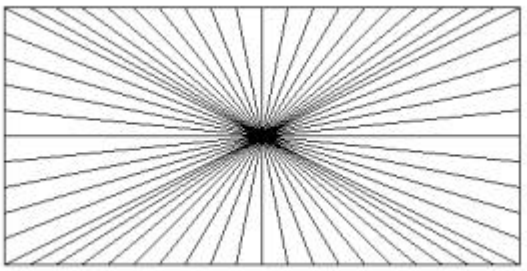


Ü5.4

Verallgemeinere die Prozedur SYMMETRIEBILD aus Problem 5.2:

a) An Stelle eines Quadrats soll ein Rechteck als Rahmen gezeichnet werden.

b) Der Abstand zwischen den Punkten auf dem Rand soll variabel sein. Am einfachsten ist es, die Anzahl der Unterteilungen als Parameter einzuführen. Warum muss diese Anzahl gerade sein?



Ü5.5

Was zeichnet die folgende Prozedur? Überlege, ohne die Befehle am Computer einzugeben.

(1)

```
to proz
repeat 6  [ sety ycor + 30  setx xcor + 50 ]
sety ycor - 30  setxy xcor - 250  ycor-150  setx xcor - 50
end
```

(2)

```
to proz
setxy xcor-40 ycor+80 setxy xcor+40 ycor+50
setxy xcor+40 ycor-50 setxy xcor-40 ycor-80
end
```