

Bedingte Anzeige: Hund am Brunnen

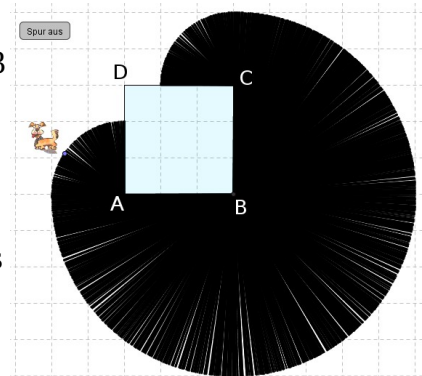
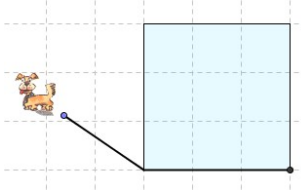


Ein Hund wird an einem quadratischen Brunnen mit Seitenlänge 3 Meter angebunden. Die Leine ist 5 Meter lang. Wie lässt sich der Bereich beschreiben, in dem sich der Hund bewegen kann?

In der Abbildung rechts ist die Fläche bereits zu sehen.

Entscheidend ist, dass die Leine länger ist als das Quadrat. Mittels

bedingter Anzeige lässt sich die Illusion erwecken, dass der Hund tatsächlich an einer biegsamen Leine befestigt ist, obwohl diese mit (starken) Strecken realisiert wurde.



Konstruktion

1. Zeichnen Sie den quadratischen Brunnen mit Seitenlänge 3. Seine Eckpunkte heißen von links unten nach links oben: A, B, C, D.
2. Zeichnen Sie die Kreisbögen, welche den maximalen Bewegungsradius des Hundes beschreiben:

1. Der Befehl des Kreisbogens besitzt die Syntax:

`Kreisbogen[Mittelpunkt, Anfangspunkt, Endpunkt]`.

Ein Viertelkreisbogen mit Radius 2 am linken unteren Eck wird also gezeichnet mit:

`Kreisbogen[A, A + (0,2), A - (2,0)]`

2. Analog lassen sich die beiden anderen Kreisbögen realisieren:

`Kreisbogen[B, B - (5,0), B + (0,5)]`

und

`Kreisbogen[C, C + (0,2), C - (2,0)]`

3. Die Kreisbögen sind noch unverbunden. Damit ein Punkt (unser Hund) sich auf einer durchgängigen Kurve bewegen kann, müssen diese zusammengefügt werden. Dies lässt sich durch Definition einer Liste erreichen. Wenn unsere Kreisbögen die Namen e, f und g haben, definieren wir die Liste „kurve“ mit:

`kurve = {e,f,g}`

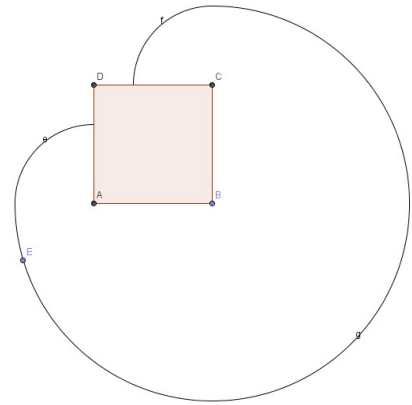
4. Nun definieren wir den Punkt H als beweglich auf dieser Kurve:

`H = Punkt[kurve]`

5. Nun bewegt sich unser Punkt H bereits auf der gewünschten Kurve. Auf der Rückseite finden Sie die Anleitung, wie sich die „Leine“ realisieren lässt.



Ihre Konstruktion sollte zu diesem Zeitpunkt ungefähr so aussehen wie die Abbildung rechts. Nun wollen wir die „Leine“ konstruieren.



6. Zeichnen Sie Strecken von Eckpunkte A, B, C zu H:

`l_1 = Strecke[A, H]`

`l_2 = Strecke[B, H]`

`l_3 = Strecke[C, H]`

Diese werden später den jeweils beweglichen Teil der Leine darstellen.

7. Zeichnen Sie darüber hinaus Strecken zwischen A und B sowie B und C. Diese stellen den Teil der Leine dar, der am Brunnenrand anliegt, wenn sich der Hund auf einem der kleinen Kreisbögen befindet:

`ab = Strecke[A, B]`

`bc = Strecke[B, C]`

8. Nun müssen die Sichtbarkeiten so eingestellt werden, dass sich die Illusion einer biegsamen Leine ergibt. `l_1` und `ab` sind sichtbar, wenn sich der Hund auf dem kleinen Kreisbogen links befindet. Dies lässt sich in Abhängigkeit von den Koordinaten des Punktes A beschreiben. Geben Sie in den Eigenschaften von `l_1` und `ab` unter dem Reiter „Erweitert“ die folgenden logischen Ausdruck ein:

`x(H) < x(A) && y(H) > y(A)`

Damit `l_1` und `ab` sichtbar sind, muss die x-Koordinate von H also kleiner sein als die von A und zugleich die y-Koordinate von H größer als die von A. Das „&&“ ist hierbei ein logisches „Und“. Das „&&“ als „Und“ ist in der Programmierung üblich und erleichtert die Eingabe. Nach Drücken der Eingabetaste wandelt sich der Ausdruck in die „normale“ Schreibweise in der Logik: `(x(H) ≤ x(A)) ∧ (y(H) ≥ y(A))`.

9. Analog gehen Sie für den oberen kleinen Kreisbogen vor. Fügen Sie die folgende Sichtbarkeitsbedingung bei `l_3` und `bc` ein:

`x(H) < x(C) && y(H) > y(C)`

10. Es fehlt noch die Anzeige der Leine, wenn sich der Hund auf dem großen Kreisbogen bewegt. `l_2` soll sichtbar sein, genau dann wenn `l_3` und `l_2` unsichtbar sind. Geben Sie also als Bedingung für die Sichtbarkeit von `l_3` ein:

`! (x(H) < x(A) && y(H) > y(A)) && ! (x(H) < x(C) && y(H) > y(C))`

Das Ausrufezeichen vor den beiden Ausdrücken ist ein logisches „Nicht“. Nach Drücken der Eingabetaste wandelt sich der Ausdruck wiederum in die „normale“ Logik-Schreibweise um.

Erweiterung

- Fügen Sie eine Schaltfläche ein, welche die Spur für die `l_1`, `l_2` und `l_3` ein- und ausschaltet. Definieren Sie dafür einen Wahrheitswert „spur“ und benutzen Sie das Kommando „SetzeSpur“.