JOHANNES GUTENBERG UNIVERSITÄT MAINZ

JG|U

Institut für Informatik

Michael Wand
Christian Ali Mehmeti-Göpel

Wintersemester 2024/25

Animieren in Basic-IO

Übung 4

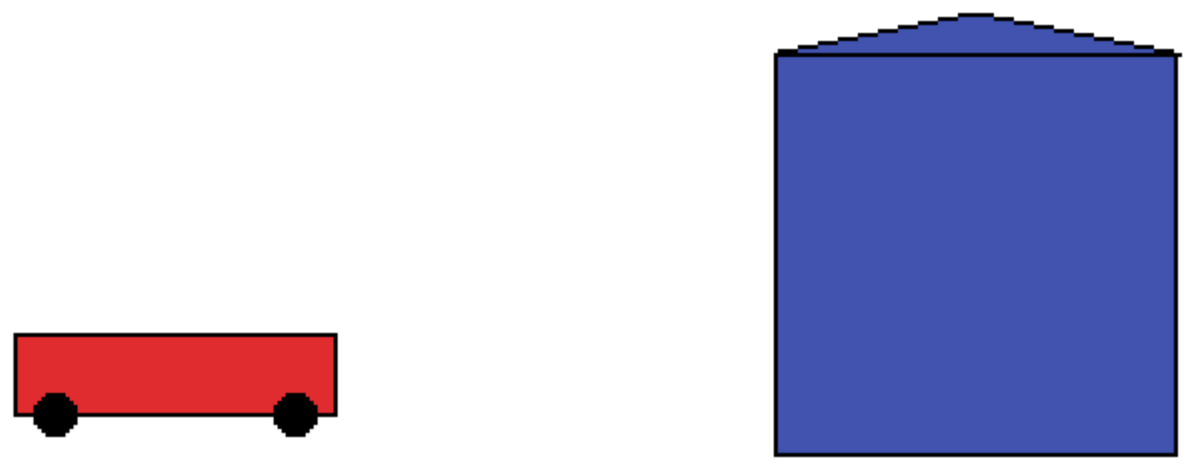
Einführung in die Programmierung

Über dieses Übungsblatt
Auf diesem Übungsblatt geht es darum, etwas komplexere Figuren mit Basic-IO zu zeichnen und zu animieren. Machen Sie sich am besten erneut mit allen Funktionen von Basic-IO vertraut und verwenden Sie die sleep Funktion des Python-Internen `time` Moduls zwischen jeder Zeichenoperation, damit Sie visuell sehen können wie der Programmcode abgearbeitet wird. Das `math`-Modul wird ebenfalls benötigt (für cos, sin) und kann ebenfalls verwendet werden.

Aufgabe Auto

Letzte Änderung: 15. June 2023, 12:38 Uhr
10 Punkte — im Detail

Ansicht: |



So soll das Endergebnis der Aufgabe aussehen.

In dieser Aufgabe möchten wir einfaches Zeichnen mit Basic-IO üben.

1) Zeichen Sie ein Auto wie oben abgebildet am linken Bildschirmrand mit den Befehlen `draw_rectangle` und `draw_circle`.

2) Zeichnen Sie ein Haus wie oben abgebildet etwas weiter Rechts mit dem Befehl `draw_polygon`.

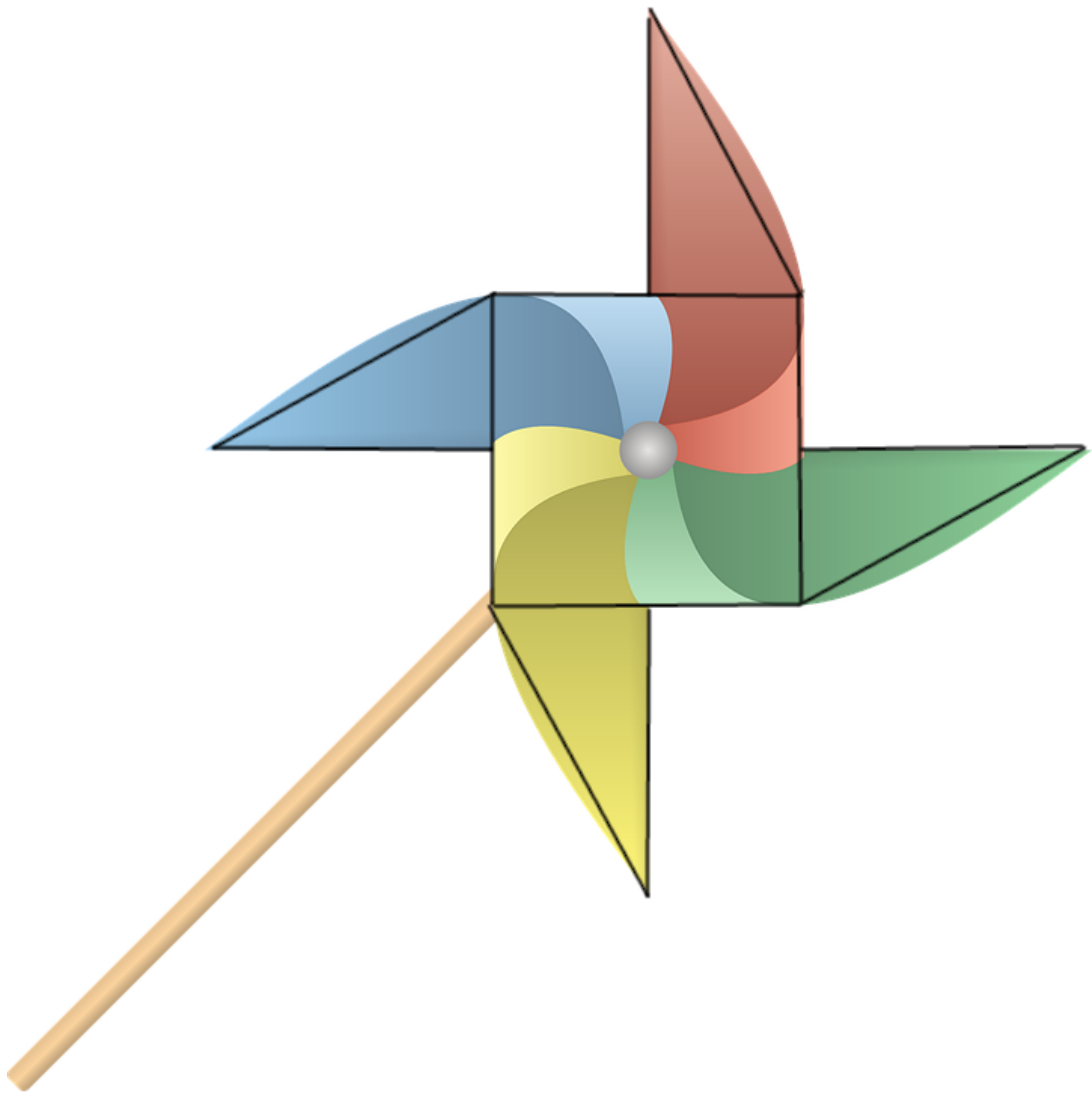
3) Animieren Sie das Auto, dass es nach rechts fährt. Verwenden Sie dafür die sleep Funktion des Python-Internen `time` Moduls und wiederholte `draw` sowie `clear_image` Befehle. Für eine 60 Herz Anzeige, verwenden Sie einen Sleep-Timer von 0.016 Sekunden. Das Auto soll **vor** dem Haus vorbeifahren.

4) Wiederholen sie die letzte Aufgabe, das Auto soll aber nun **hinter** dem Haus vorbeifahren.

Aufgabe Windrad

Letzte Änderung: 06. July 2023, 13:26 Uhr
15 Punkte — im Detail

Ansicht: |



Ein Windrad für Kinder Bildquelle

Das Ziel dieser Aufgabe ist es ein stilisiertes Windrad für Kinder wie oben dargestellt aus einem Rechteck und vier Dreiecken zu zeichnen.

1) Zeigen Sie das Stilisierte Windrad statisch auf dem Bildschirm an.

2) Die Koordinaten eines rotierten Punktes $(x_1, y_1) \in \mathbb{R}^2$ mit Winkel θ um den Mittelpunktist gegeben durch

$$x_1 = (x_0 - x_c)cos(\theta) - (y_0 - y_c)sin(\theta) + x_c$$
$$y_1 = (x_0 - x_c)sin(\theta) + (y_0 - y_c)cos(\theta) + y_c$$

wobei (x_0, y_0) die ursprünglichen Koordinaten des Punktes sind und (x_c, y_c) die Koordinaten des Zentrums der Rotation.

2a) Animieren Sie zunächst das Quadrat in der Mitte so, dass es sich (in gut sichtbarer Geschwindigkeit) um die eigene Achse dreht. Schreiben Sie dazu eine Funktion `draw_rot_rectangle(angle)`, die vom Rotationswinkel abhängt.

2b) Animieren Sie nun auch die Dreiecke so, dass sie sich mit dem Quadrat um den Mittelpunkt des Quadrats drehen. Schreiben Sie dazu eine Funktion `draw_rot_trinagle(angle, ref_coords)`, die vom Rotationswinkel sowie von Referenzkoordinaten (z.B. die untere linke Ecke oder der Mittelpunkt des Dreiecks) abhängt und verwenden Sie diese vier Mal mit verschiedenen Argumenten.

Aufgabe Hypnobot

Letzte Änderung: 15. June 2023, 12:38 Uhr
15 Punkte — im Detail

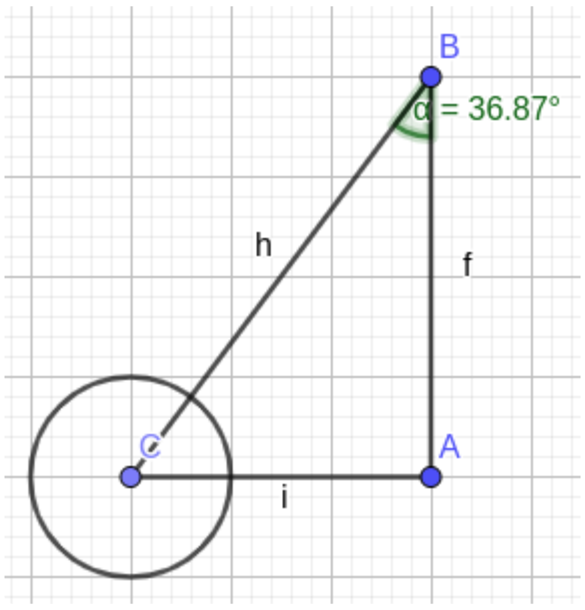
Ansicht: |

Ein schwingendes Pendel und konzentrische Kreise

In dieser Aufgabe möchten wir ein paar regelrecht "Hypnotisierende" Grafiken auf den Bildschirm bringen. Ein animiertes gif des gewünschten Endprodukts der Aufgabe liegt in LMS bereit.

1) Wir beginnen mit den konzentrischen Kreisen am rechten Bildschirmrand. Animieren Sie fünf konzentrische Kreise verschiedener Farbe wie in der Musterlösung abgebildet. Die Radien der Kreise sind initial 20; 40; 60; 80 und 100, schrumpfen aber immer weiter mit der Zeit bis sie Radius 0 erreichen und dann wieder bei ihrem Initialwert starten. Verwenden Sie dazu eine Division mit Rest, in Python verfügar durch den Modulo-Operator `%`. Die Animation soll von einer Zeitvariable t abhängen dessen Arbeitsintervall sensibel gewählt werden muss (die Animation soll nicht zu schnell oder zu langsam sein). Kapseln Sie ihren Programmcode in einer Funktion `draw_concentric_circle(t, circle_center)`, die von der Zeit `t` sowie dem Mittelpunkt der Kreise `circle_center` abhängt.

2) Nun möchten wir ein schwingendes Pendel animieren. Ein einfaches Pendel besteht aus einem Gewicht (Kreis) an einem Faden (Gerade), das hin und her schwingt. Die Länge des Fadens (der Gerade) ist hierbei immer gleich. Der Winkel des Pendels in Abhängigkeit der Zeit ist gegeben durch $\phi(t) = \phi_{max} \cdot sin(\omega t)$ wobei t die Zeit, ω eine Konstante und ϕ_{max} den Winkel der maximalen Auslenkung des Pendels repräsentiert.



2a) In dieser Figur ist B der Aufhängepunkt des Pedels, das Gewicht hängt an Punkt C und h enspricht der Fadenlänge. Berechnen Sie die Seitenlängen f und i in Abhängigkeit von der Fadenlänge h und des Winkels α.

2b) Animieren Sie ein wie in der Musterlösung schwingendes Pendel mit $\phi_{max} \approx pi/2$ und $\omega = 1$. Kapseln Sie ihren Programmcode in einer Funktion `draw_pendulum(angle)`, die vom Winkel `angle` abängt; der Winkel soll aus der gleichen Zeitvariable wie in **1)** berechnet werden. Passen Sie dazu ggf. das Arbeitsintervall der Zeitvariable an.

2c) Kombinieren Sie beide Animationen, so dass der Mittelpunkt der konzentrischen Kreise dem Mittelpunkt des Pendelgewichts (Punkt C) entspricht. Verwenden Sie dazu die in den letzten Aufgabenteilen erstellten Funktionen.