

Kollisionskontrolle und Counter

Übung 5

Einführung in die Programmierung

Über dieses Übungsblatt

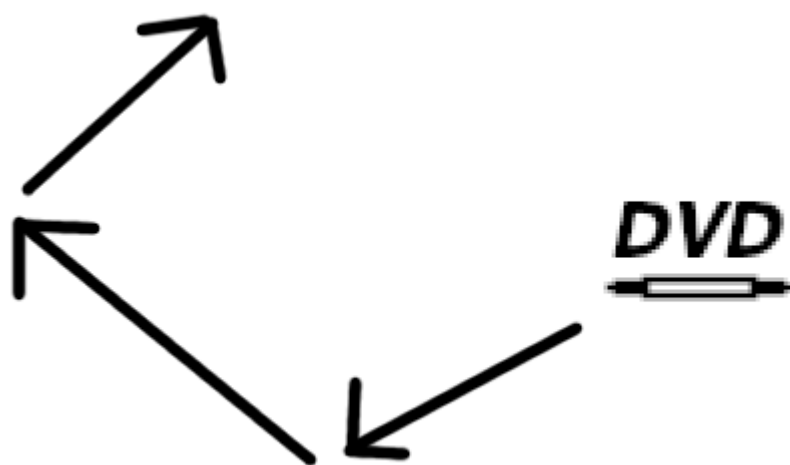
In der ersten Hälfte des Übungsblattes geht es darum, für einen einen sich bewegendenden Agenten Kollisionskontrollen (nicht in Wände reinlaufen etc.) durchzuführen und auf Ereignisse zu reagieren. Dies ist die Grundlage für viele Spiele. In der zweiten Hälfte geht es darum, den Umgang mit Python Listen erneut zu üben und damit Dinge in Schleifen zu zählen (counter).

Aufgabe Dvd Screensaver

Letzte Änderung: 06. July 2023, 13:26 Uhr

28 Punkte – [im Detail](#)

Ansicht:  | 



Dieser Bildschirmschoner war in den 2000-er Jahren Kult

In dieser Ausgabe möchten wir den Kult-Bildschirmschoner alter DVD-Lesegeräte mit Hilfe von Basic-IO nachprogrammieren. Sie werden dafür die Funktionen `read_image`, `clear_image` und `draw_image` benötigen; machen Sie sich damit vertraut. Die Aufgabe wird in mehreren Ausbaustufen gestellt, **nur die letzte Ausbaustufe muss abgegeben werden**. Ein animiertes gif der Lösung liegt im LMS bereit.

1) Lesen Sie das in LMS bereitgestellte DVD-Logo ein mit `load_image` ein und zeigen Sie es mit `draw_image` in der Mitte ihres Bildschirms an.

2) Bewegen Sie das DVD-Logo in einer Animation stetig nach Rechts. Verwenden Sie dafür die `sleep` Funktion des Python-Internen `time` Moduls und die `clear_image` Funktion von Basic-IO. Für eine 60 Herz Anzeige, verwenden Sie einen Sleep-Timer von 0.016 Sekunden.

3) Wenn sich die Koordinaten des DVD-Logos bis auf 25 Pixeln dem rechten Rand nähert, soll die Bewegungsrichtung des Logos sich ändern damit es wieder nach Links zurück läuft.

Das DVD-Logo soll nun wie im animierten gif gezeigt auch schräg an einem Rand abprallen können.

4a) Parametrisieren Sie die Bewegung des DVD-Logos durch eine Gerade im \mathbb{R}^2 , also $\begin{pmatrix} p_1 \\ p_2 \end{pmatrix} + \lambda \cdot \begin{pmatrix} d_1 \\ d_2 \end{pmatrix}$, wobei p der Aufpunkt und d

der Richtungsvektor der Gerade sind. Verändern Sie die anfängliche Bewegungsrichtung des DVD-Logos in eine diagonale Richtung (der genau Winkel ist egal).



4b) Wenn sich das Logo bis auf 25 Pixel einem beliebigen Rand nähert, soll das Logo wie oben gezeigt "schräg abprallen", d.h. der Richtungsvektor der Gerade soll sich ändern so dass Eingangswinkel des alten Richtungsvektors dem Ausgangswinkel des neuen Richtungsvektors entspricht. Dies ist ganz einfach zu realisieren durch einen Vorzeichenwechsel in der richtigen Koordinate des Richtungsvektors.

5) Wenn die Koordinaten des DVD-Logos weniger als 35 Pixel von einer Ecke entfernt sind sagen wir, dass das Logo die "Ecke berührt". Ihr Programm soll jedes mal wenn eine Ecke getroffen wird in der Basic-IO Konsole mit der `print_message`-Methode die Anzahl an Eckenkollisionen ausgeben, die seit Ausführung des Programms aufgetreten sind.

Aufgabe Heavy Metal Umlaute

Letzte Änderung: 06. July 2023, 13:26 Uhr

12 Punkte — [im Detail](#)

Ansicht:  | 

Eine Zeit lang war es bei Rockbands des Subgenres „Heavy Metal“ in Mode, Umlaute auf einige der Buchstaben a, u, o im Bandnamen zu platzieren. So schreibt sich die bekannte Band „Motorhead“ als „Motörhead“. Diesen Prozess möchten wir jetzt automatisieren. Schreiben Sie eine Python-Funktion, die eine Zeichenkette übergeben bekommt und darin nach Buchstaben *"a", "u" oder "o"* sucht. In jedem Wort (getrennt durch Leerzeichen) soll immer der zweite Vokal dieser Art (a, u, o) durch den entsprechenden Umlaut (*"ä", "ü", "ö"*) ersetzt werden. Die Eingabe besteht grundsätzlich nur aus Wörtern (Sequenzen der 26 Buchstaben A-Z des Alphabets, groß oder klein geschrieben), die durch genau ein Leerzeichen getrennt sind; es kommen keine anderen Zeichen vor: kein Zeilenumbruch, Tabulator oder ähnliches und auch keine Umlaute. Wir machen dies in zwei Varianten:

1) Es werden alle Buchstaben zusammen gezählt. Es wird der zweite Buchstabe, der ein a, o oder u ist ersetzt. `auu` wird also zu `äuü`.

Beispiele:

- "Motorhead" → "Motörhead"
- "Heavy Metal Umlaute" → "Heavy Metal Umläute"
- "aaauuu oooo iii kukukuku" → "ääauuu öooo iii kukükuku"
- "Python super Programmiersprache" → "Python super Progrämmiersprache"

2) Jeder Buchstabe wird einzeln gezählt. Es wird das zweite a, das zweite o oder das zweite u ersetzt. `auua` wird also zu `äuüa`

Beispiele:

- "Motorhead" → "Motörhead"
- "Heavy Metal Umlaute" → "Heavy Metal Umläute"
- "aaauuu oooo iii kukukuku" → "ääauuu öooo iii kukükuku"
- "Python super Programmiersprache" → "Python super Programmierspräche"

Tipps:

- Einzelne Buchstaben werden in Python als kurze Zeichenketten geschrieben, z.B. "a", Leerzeichen als " ". Vergleiche mit `==` und `!=` (ungleich) sind wie bei Zahlen möglich.
- Den i-ten Buchstaben von Zeichenkette z (gezählt von 0 an) erhält man mit `z[i]`. Man kann darauf lesend und schreibend (Buchstaben mit `z[i] = ...` ändern) zugreifen.
- Strings sind in Python immutable, das bedeutet dass man einen String nicht in-place modifizieren kann. Es muss also eine neuer String für die Ausgabe aufgebaut werden, z.B. als Liste die am Ende in einen String umgewandelt wird. Sie können dies beispielsweise folgendermaßen realisieren:

```
# create new empty list

out = []

# append stuff
output.append("foo")
output.append("bar")

# make it a string
out = "".join(out)
```

Hinweis: Für diese Aufgabe sind alle Python internen Funktionen erlaubt. Wenn Sie aber ein Anfänger sind, empfehlen wir Ihnen stark, sich auf die Basics zu beschränken.