Universität
des
Saarlandes
FR 6.2 Informatik

Prof. Dr. Kurt Mehlhorn, Dr. Konstantinos Panagiotou **WiSe 2011/2012**

Übungen zu Computational Thinking

http://www.mpi-inf.mpg.de/departments/d1/teaching/ws11/ct/

Blatt 11 Abgabeschluss: 30.1.12 16:00

Regeln: Bis zum Semesterende müssen mindestens 42% der maximal erreichbaren Punkte aller Übungszettel erworben werden.
Programmcode ist elektronisch per E-Mail abzugeben. Zusätzlich müssen die Ausgaben einer exemplarischen Programmausführung mitgeliefert werden.

Bleistiftaufgaben

Aufgabe 1 (*20 Punkte*) Nehmen Sie an, Sie wollen die Bewegung eines Autos simulieren. Über eine numerische Methode soll die Geschwindigkeit des Autos an Hand der Beschleunigungs- und Bremsoperationen bestimmt werden. Sie verfügen über folgende Informationen:

$$\begin{aligned} v(0) &= 0 \\ v(t) &= \int_0^t a(t)\ dt \\ a(t) &= \begin{cases} 5 & t \leq 25 \\ -7 & t > 25 \end{cases} \end{aligned}$$

wobei $a(t) = dv/dt$ die Beschleunigung bezeichnet.
Wir benutzen die Eulermethode mit Schrittweite δ. Wir wollen bestimmen, wann das Auto erstmals eine Geschwindigkeit von 100 überschreitet und wann es wieder eine Geschwindigkeit von 0 hat. Außerdem interessiert uns die Geschwindigkeit zum Zeitpunkt $t = 25 + \delta$.
Welchen Einfluss hat die Wahl der Schrittgröße bei der numerischen Approximation auf die Genauigkeit unserer Antworten? Wie kann man mit den auftretenden Problemen umgehen und welche Annahmen muss man dazu ggf. über die zu approximierende Funktion machen können?

Python

Aufgabe 2 (*10 Punkte*) Wir wollen die Bewegung eines Punktes (x, y) mit $0 < x < 1$ und $0 < y < 1$ im 2D-Raum simulieren. Der Punkt bewegt sich in diskreten Schritten, wir können die Simulation also perfekt ausführen. Die Bewegung folgt den folgenden Regeln:

$$\begin{aligned} x(t+1) &= 4x(t) \cdot (1 - x(t)) \\ y(t+1) &= \begin{cases} x + y & x + y < 1 \\ x + y - 1 & \text{sonst} \end{cases} \end{aligned}$$

Wir können allerdings die Anfangsposition des Punktes nicht ganz genau bestimmen. Unsere Messgeräte haben einen kleinen normalverteilten Messfehler mit Mittelwert $\mu = 0$ und Standardabweichung $\sigma = 0.05$. In Python können sie normalverteile Zufallszahlen mit Hilfe der im Modul `random` definierten Funktion `random.gauss(mu,sigma)` erzeugen.
Untersuchen Sie, welchen Einfluss die Messfehler auf die Vorhersagekraft unserer Simulation haben.

Aufgabe 3 (*30 Punkte*) Auf der Webseite von John Zelle[1] finden Sie eine Python Datei, die es Ihnen erlaubt einfache Graphiken anzuzeigen und zu animieren, sowie dazugehörige Dokumentation.

a) (*10 Punkte*) Schreiben Sie ein Programm, das die Bewegung eines Flummis mit Durchmesser 10 in einem 400×400 Einheiten großen Raums simuliert. Der Flummi startet zentriert an Position $(200, 200)$ mit einer anfänglichen Geschwindigkeit von $v_x(0) = 50$ in x-Richtung und $v_y(0) = 70$ in y-Richtung. Im Raum herrscht normale Erdgravitation mit Gravitationskonstante $g = 9.81$. Jedes Mal wenn der Flummi von einer Wand apprallt, verliert er 10% seiner Geschwindigkeit.

b) (*20 Punkte*) Schreiben Sie eine möglichst realistische Simulation der Bewegung der Planeten unseres Sonnensystems. Die relevanten Daten können Sie der Wikipedia entnehmen. Bei der Darstellung sollten Sie Abstände und Planetengrößen so skalieren, dass man etwas sieht.

Wir nehmen bei allen Aufgaben an, dass die Newtonschen Bewegungsgleichungen korrekt sind.

[1] `http://mcsp.wartburg.edu/zelle/python/`