

Praktikumsversuch 3: Weltraum-Simulation

Hardwarenahe Programmierung · Wintersemester 2024/25 · Prof. Dr. Peter Gerwinski

Aufgabe: Schreiben Sie ein C-Programm, das die Bahnen von beliebig vielen Massenpunkten unter Einfluß der Gravitation simuliert und in bewegter Grafik darstellt.

- Zwei Massenpunkte („Himmelskörper“) mit den Massen m_i und m_j an den Orten \vec{r}_i und \vec{r}_j ziehen einander an. Diese Kraft heißt Gravitation. Sie hat den Betrag:

$$|\vec{F}_{ij}| = \frac{m_j \cdot m_i \cdot G}{|\vec{r}_j - \vec{r}_i|^2} \quad (1)$$

Hierbei ist G eine Konstante (Gravitationskonstante).

- Die auf einen Himmelskörper wirkende Gesamtkraft \vec{F}_i ergibt sich als Summe der von allen anderen Himmelskörpern herrührenden Gravitationskräfte:

$$\vec{F}_i = \sum_{j=0, j \neq i}^{N-1} \vec{F}_{ij} \quad (2)$$

- Die Gravitationskraft beschleunigt jeden Himmelskörper gemäß:

$$\vec{F}_i = m_i \cdot \vec{a}_i \quad (3)$$

- Beispiel: Wir betrachten zwei Himmelskörper. Einer davon („Zentralgestirn“) ruht im Zentrum ($\vec{r}_0 = 0$, $\vec{v}_0 = 0$) und hat eine wesentlich größere Masse als der andere („Satellit“, $m_1 \ll m_0$). Mit geeignetem Anfangsort \vec{r}_1 und geeigneter Anfangsgeschwindigkeit \vec{v}_1 beschreibt dann der Satellit eine elliptische Umlaufbahn um das Zentralgestirn.

- Wir rechnen in zwei Dimensionen x und y .
- Für die Zerlegung einer Kraft \vec{F}_{ij} in x - und y -Komponenten benötigen Sie nur die Grundrechenarten und die Wurzelfunktion, jedoch insbesondere *keine* trigonometrischen Funktionen:

$$\vec{F}_{ij} = |\vec{F}_{ij}| \cdot \frac{\vec{r}_j - \vec{r}_i}{|\vec{r}_j - \vec{r}_i|} \quad (4)$$

- Die Wurzelfunktion `sqrt()` finden Sie in der Mathematik-Bibliothek. Um diese zu nutzen,

verwenden Sie `#include <math.h>` im Quelltext, und geben Sie beim `gcc`-Aufruf `-lm` mit an.

- Speichern Sie die verschiedenen Körper nicht als separate Variable, sondern in einem Array. Ihr Programm wird dadurch nicht nur flexibler, sondern auch übersichtlicher und weniger fehleranfällig.

- Für die Simulation betrachten wir das System in kurzen Zeitintervallen dt und berechnen die Änderungen des Ortes $\vec{r}_i = (x_i, y_i)$ und der Geschwindigkeit $\vec{v}_i = (v_{xi}, v_{yi})$ jedes Himmelskörpers mit Hilfe des expliziten Eulerschen Polyzugverfahrens.

(Wer möchte, darf natürlich auch andere Verfahren anwenden, beispielsweise das klassische Runge-Kutta-Verfahren 4. Ordnung.)

- Für eine derartige Simulation einschließlich ihrer Darstellung als bewegte Grafik können Sie sich von dem Beispiel-Programm `pendulum-1.c` inspirieren lassen.

- In einer `GTK+-drawing_area` liegt der Nullpunkt der Zeichnung oben links, eine Längeneinheit entspricht einem Pixel, und die y -Koordinate wächst nach unten. Es empfiehlt sich, die Koordinaten so umzurechnen, daß der Nullpunkt in der Mitte der Zeichnung liegt, die Längeneinheit Ihrem persönlichen Geschmack entspricht und die y -Koordinate nach oben wächst.

- Beispiel-Szenarien für 3 oder mehr Körper:
 - Planet mit Mond umkreist Sonne
 - Sonne mit mehreren Planeten, die sich gegenseitig beeinflussen
 - zwei Sonnen umkreisen sich gegenseitig, Planet kreist drumherum
 - Raumsonde besucht nacheinander mehrere Planeten

GTK unter GNU/Linux

- Installieren Sie das Entwickler-Paket für GTK-3, z. B. unter Debian GNU/Linux und darauf basierenden Systemen:

```
apt-get install libgtk-3-dev
```

GTK unter MacOS X

- Zuerst den Paketmanager „Homebrew“ installieren:

```
ruby -e "$(curl -fsSL https://raw.githubusercontent.com/
Homebrew/install/2020ws/install) "
```

(Die URL nicht trennen, sondern in einer Zeile schreiben.)

- Danach mit Hilfe von `brew` GTK installieren:

```
brew install gtk+3
```

GTK unter Microsoft Windows: Cygwin

- Im Cygwin-Setup-Programm die Pakete `libgtk3-devel`, `xorg-server` und `dbus-x11` installieren.
- Bevor Sie GTK-Programme starten können, müssen Sie einmalig den X-Server starten:

```
XWin -multiwindow &
```

- Bevor Sie GTK-Programme starten können, müssen Sie einmalig pro Shell-Sitzung die `DISPLAY`-Variable setzen:

```
export DISPLAY=:0.0
```

GTK unter Microsoft Windows: MinGW

- Installieren Sie `MSYS2` (Installer).
- Installieren Sie von der MinGW-Shell aus GTK und weitere Entwicklungswerkzeuge:

```
pacman -S mingw-w64-x86_64-gcc
pacman -S mingw-w64-x86_64-gtk3
pacman -S mingw-w64-x86_64-pkg-config
```

- Falls nötig, ergänzen Sie in der MinGW-Shell den Pfad zu den neu installierten Programmen:

```
export PATH=$PATH:/mingw64/bin
```

GTK unter Microsoft Windows: WSL

- Installieren Sie die GTK-Entwicklungswerkzeuge in ihrer Linux-Umgebung (siehe oben).
- Installieren Sie einen X-Server für Ihr MS-Windows-System, z. B. `Xming`.
- Starten Sie Ihren X-Server mit `XLaunch` und wählen Sie *no access control* aus.
- Bevor Sie GTK-Programme starten können, müssen Sie einmalig pro Shell-Sitzung die `DISPLAY`-Variable setzen:

```
export DISPLAY=$(cat /etc/resolv.conf \
| grep nameserver | awk '{print $2}'):0
```

(Der Backslash ist ein Zeilen-Verlängerungszeichen. Wenn Sie den gesamten Befehl in einer Zeile eingeben, entfällt der Backslash.)

Viel Erfolg!

Stand: 12. Dezember 2024

Copyright © 2014, 2015, 2018-2022, 2024 Peter Gerwinski
Lizenz: CC BY-SA (Version 4.0) oder GNU GPL (Version 3 oder höher)

Sie können diese Praktikumsunterlagen einschließlich \LaTeX -Quelltext und Beispielprogramm herunterladen unter: <https://gitlab.cvh-server.de/pgerwinski/hp>