Prof. Dr. Günther Greiner, Benjamin Keinert, Matteo Colaianni, Darius Rückert *Erlangen, 11.11.2015*

Kugeln (Aufgabenblatt 2)

In diesem Aufgabenblatt soll das vorhandene Partikelsystem so erweitert werden, dass die Simulation von Kugeln ermöglicht wird. Das beinhaltet vor allem die Erkennung und Behandlung von Kollisionen zwischen den Kugeln. Im vorliegender Betrachtung ist eine Kugel ein Partikel, das zu seiner Masse noch eine räumliche Ausdehnung (definiert durch den Radius) besitzt.

Aufgabe 1 [6 Punkte] (Einfache Kollisionserkennung)

a) Schreiben Sie eine `device` Funktion `collideSpherePlane`, welche in der Lage ist, Kollisionen zwischen einer Kugel und einer Ebene zu erkennen. Eine gute Einführung bietet das Dokument `collision1.pdf` (Verfügbar im `StudON`)

b) Nun soll eine `device` Funktion `collideSphereSphere` entwickelt werden, die Kollisionen zwischen den Kugeln erkennt. In einer ersten vereinfachten Version soll jede Kugel S_i nacheinander jede andere Kugel S_j testen, ob eine Kollision besteht.

c) Da bisher noch keine Kollisionsantwort implementiert wurde, verwenden Sie die Möglichkeit einer visuellen Kollisionskontolle, indem sie z.B. die Farbe der Kugel verändern, sobald eine Kollision detektiert wurde.

Aufgabe 2 (Kinematische Kollisionsantwort)

In dieser Teilaufgabe soll die Kollisionsantwort ohne Integration von Kräften durchgeführt werden. Die dabei entstehende Simulation wird sich nicht physikalisch korrekt, dafür aber *plausibel* verhalten.

a) Lösen Sie die Kollision auf, indem Sie die Position der eindringenden Kugel S_i auf die letzte gültige Position entlang der Bewgungsrichtung setzen.

b) Damit die Simulation plausibel wird, muss das Reflexionsverhalten im Falle einer Kollision berücksichtigt werden. Spiegeln Sie daher den Impuls an der Kollisionsnormalen.

Aufgabe 3 (Dynamische Kollisionsantwort)

Der eben entwickelte Kernel soll nun eine korrekte Kollisionsantwort ermöglichen. Im Falle eine Kollision soll nun ein Kaftstoß erzeugt werden:

$$\Delta\vec{p} = \vec{F}_{total} \cdot \Delta t = (\vec{F}_{spring} + \vec{F}_{dashpot} + \vec{F}_{shear}) \cdot \Delta t$$

a) Implementieren Sie diese Kollisionsantwort sowohl für die Kugel-Ebene- als auch für die Kugel-Kugel-Kollision, wobei sich $\vec{F}_{total}$ aus drei Komponenten zusammensetzt:

- Eine Federkraft, die versucht, die Kugel aus der anderen zu drücken:
  $\vec{F}_{spring} = \lambda_{spring} \cdot (r_i + r_j - \|\vec{c}_i - \vec{c}_j\|) \cdot \vec{n}_{ij}$
- Eine Dämpfungskraft, die in Richtung der Kollisionsnormalen wirkt:
  $\vec{F}_{dashpot} = \lambda_{dashpot} \cdot \vec{v}_{rel}$
- Eine Scherkraft in tangentialer Richtung zur Kollisionsebene:
  $\vec{F}_{shear} = \lambda_{shear} \cdot (\vec{v}_{rel} - \langle\vec{v}_{rel}, \vec{n}_{ij}\rangle \cdot \vec{n}_{ij})$

Die Bezeichnungen bedeuten:

- Kugelmittelpunkt der Kugel i: $\vec{c}_i$
- Relative Geschwindigkeit: $\vec{v}_r el = (\frac{\vec{p}_j}{m_j} - \frac{\vec{p}_i}{m_i})$
- Kollisionsnormale: $n_{ij} = \frac{c_i - cj}{\|c_i - c_j\|}$

b) Die Resultierende Impulsveränderung muss auf beide Kugeln addiert werden. Achten Sie bei der Addition auf Nebenläufigkeit und behandeln Sie diese entsprechend.

Aufgabe 4 (Szene)

Nun soll die Implementierung an einer kleinen Szene getestet werden.

a) Fügen Sie der Szene 5 Ebenen hinzu, so dass eine nach oben geöffnete Schachtel entsteht.

b) Fügen Sie der Szene eine beliebige Anzahl an Kugeln N hinzu und initialisieren Sie die Positionen kollisionsfrei.

Viel Erfolg !

Abgabe: Montag, 23.11.2015, vor 24:00 Uhr.