Technology
Arts Sciences
TH Köln

Einführung in Game Physics

Simulation für Spiele

Prof. Dr. Wolfgang Konen

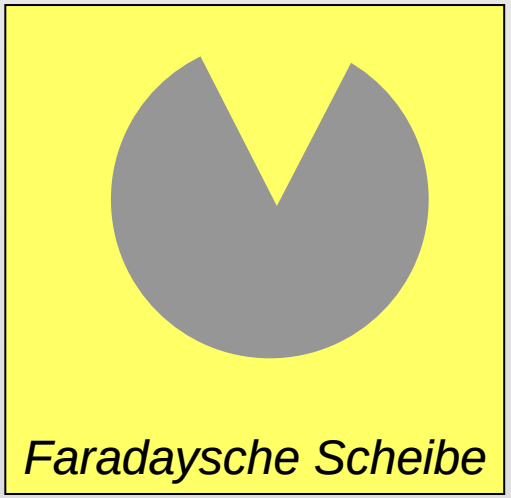
TH Köln

Welche Bedeutung hat die Simulation für Spiele?

Technology Arts Sciences TH Köln

- **Animation vs. Simulation**
- **zwei Wege zur Dynamisierung von Computergrafik**

- **Grundlagen der Bewegtbildwahrnehmung**
  - „Persistence of Vision“
  - 25-50 Hz reichen
  - „natürliche“ Bewegungen erfordern
    - „slow in and slow out“ (Bsp. Auto)
    - eher nichtlineare Bögen als Geraden



Animation

- **im einfachsten Fall: Bereitstellung von zeitlich veränderten Frames. Selbst bei Erzeugung aus „3D-Modell mit Freiheitsgraden“ noch aufwendig:**

	Freiheitsgrade	Frames * Sekunden	Einzustellende Parameter
Starrer Körper	6	30 * 5	900
Virtueller Mensch	200	30 * 60	360.000

[aus: Bender, Brill: Computergrafik]

Animation

Technology Arts Sciences TH Köln

- **Animation bedeutet in der Computergrafik meist: Key-Frames und dazwischen (meist lineare) Interpolation**

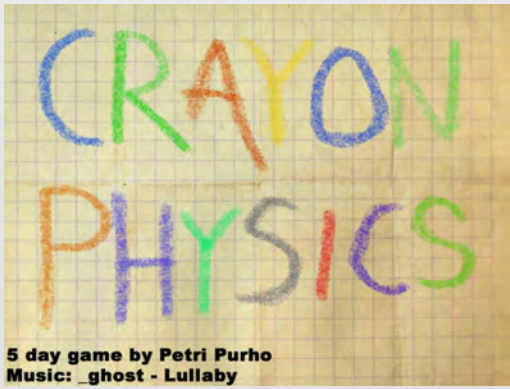
- **Probleme:**
  - linear wirkt oft unnatürlich
  - welche Key-Frames / Interpolationen bei deformierbaren Gegenständen?
  - was ist bei mehreren Objekten, Kollisionen??
  - was ist lineare Interpolation bei Rotationen??
  - bei Gelenkarmen kann es zu unerlaubten Zwischenzuständen kommen
  - was ist bei vielen Teilchen (Rauchpartikel, Fischschwarm)?
  - wie reagiert man auf interaktive Änderung (Spiele!) ?

Simulation

Technology Arts Sciences TH Köln

- **Simulation: Beschreibung der 3D-Welt durch ein dynamisches System**
  - Beispiel: Ball unter Einfluss Schwerkraft, Reflektion an Boden/Wänden

- **Vorteile**
  - unglaubliche Reduktion der einzustellenden Parameter: Man braucht nur Startzustand zu parametrisieren, den Rest rechnet das dynamische System selbst
  - Interaktion jederzeit einbaubar
  - Mehrere Objekte kein Problem, Kollision ebenfalls behandelbar
  - Interpolation der Rotation folgt als emergentes Verhalten starrer Körper



Simulation

Schwierigkeiten

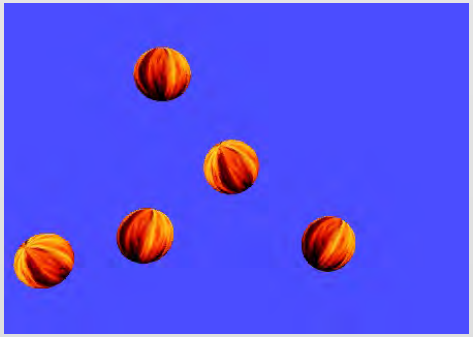
- das dynamische System muss bekannt sein
  - z. B. kann es schwierig sein, das richtige dynamische System für einen menschlichen Körper aufzustellen
  - darin liegt aber zugleich der Nutzen von gut designten Physics Engines
- es kann mathematisch komplex werden
- u.U. hohe Rechenleistung, insbesondere bei zahlreichen Objekten und Kollisionen

Simulation im Beispiel

- **Beispiel: Ball unter Einfluss der Schwerkraft**

$$\ddot{x} = \dot{v} = g \quad \rightarrow \quad \frac{v(t+h) - v(t)}{h} = g$$

$$\dot{x} = v \quad \rightarrow \quad \frac{x(t+h) - x(t)}{h} = v(t)$$



Ausschnitt **balls2D.pde**:

```
double ax=0.0, ay=-9.8;
for (int bn=0;bn < totalball; bn++) {
   ball[bn].sx += ball[bn].vx * DT;
   ball[bn].sy += ball[bn].vy * DT;
   ball[bn].vx += ax * DT;
   ball[bn].vy += ay * DT;
}
```

Hier weggelassen: Reflektion an Boden, Wänden und Decke

Aufgaben der Simulation in Spielen = Aufgaben einer Physics Engine

Technology Arts Sciences TH Köln

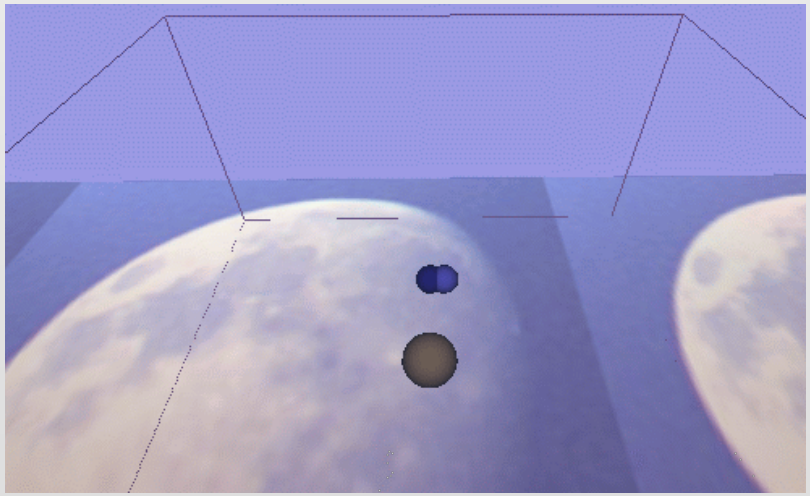
Welche Aufgaben soll eine Physics Engine lösen?

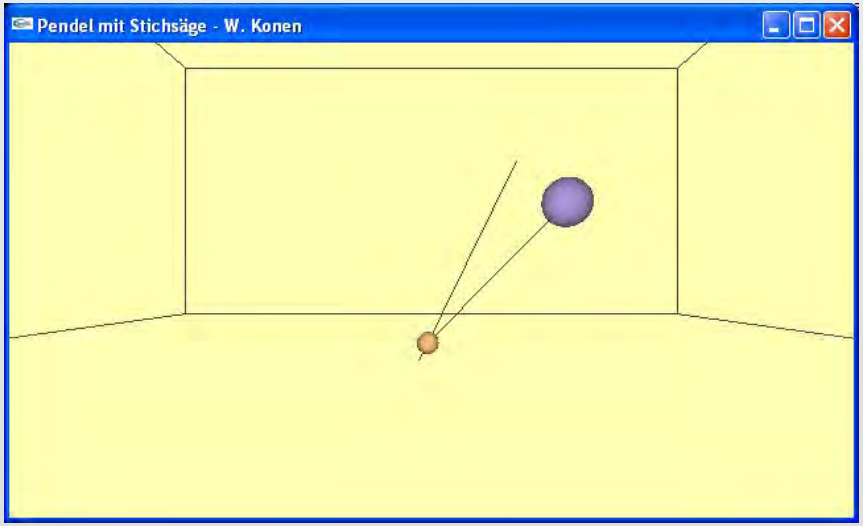
1. **Bewegung gemäß Kraftgleichungen**
2. **Kollision**
3. **Simulation komplexer (nicht-starrer) Körper**
4. **Komplexe Oberflächen (Wasser, Vorhang, Pelz)**
5. **Partikelsysteme**
6. **Reaktion auf Interaktion**

Aufgaben der Simulation in Spielen
1. Bewegung gemäß Kraftgleichungen

Technology
Arts Sciences
TH Köln

1. **Gravitation (Mond)**
2. **Feder, andere äußere Kräfte (Stichsäge)**



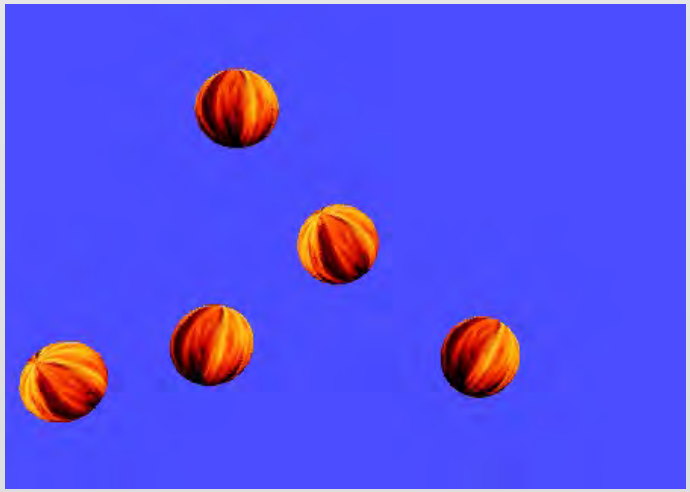


Aufgaben der Simulation in Spielen
2. Kollisionen

Technology
Arts Sciences
TH Köln

Wieso sind Kollisionen kompliziert?

- **Kollisionsgeometrie bei komplexen Objekten**
- **aufwendig bei vielen Objekten >> $O(n^2)$**

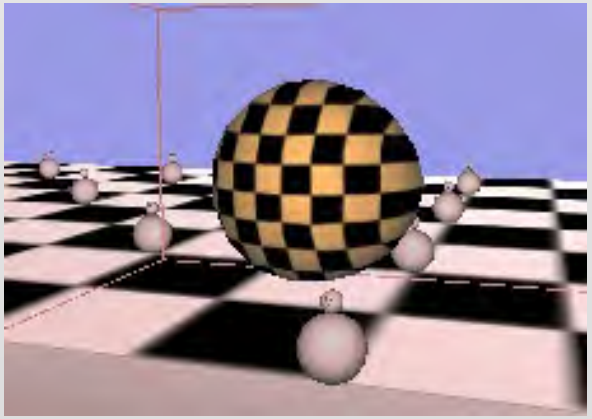


Wir beschränken uns hier auf einfache Verfahren und Objekte, deren genaue Kollisions-beschreibung ist schon kompliziert genug

Aufgaben der Simulation in Spielen
3. Komplexe (nicht-starre) Körper

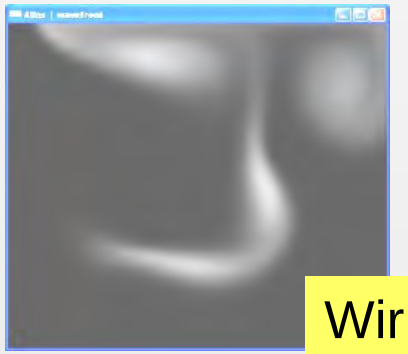
Technology
Arts Sciences
TH Köln

1. **Deformierbare Körper**
   (>> Bsp. Softball, wie man 2D-Dynamik auf 3D verallgemeinert)
2. **Gelenkpuppen, Roboter (>> Bsp. Meqon-Demo)**

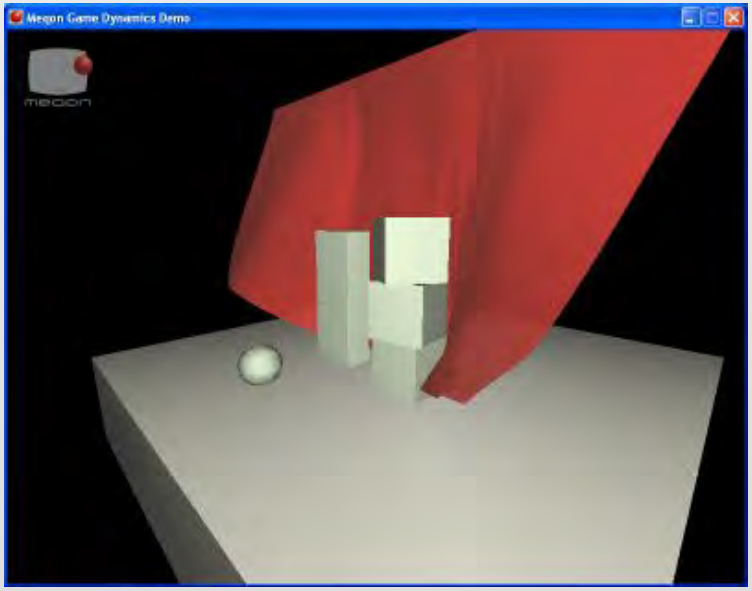


Aufgaben der Simulation in Spielen
4. Komplexe Oberflächen

Technology
Arts Sciences
TH Köln



1. **Wasser , Rauch**
2. **Vorhang, Kleidung**

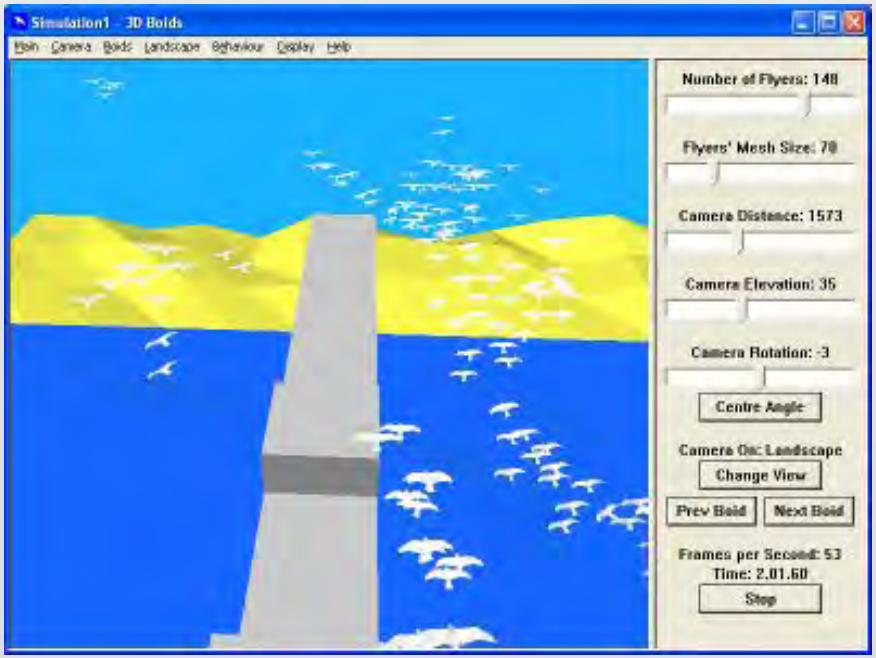


Wir werden dies hier nicht weiter behandeln, da die Mathematik kompliziert ist (partielle Differentialgleichungen, FEM-Verfahren)

Weiterführende Literatur: Baraff, Witkin: „Large steps in cloth simulation“ Comp. Graphics Proc. 1998.

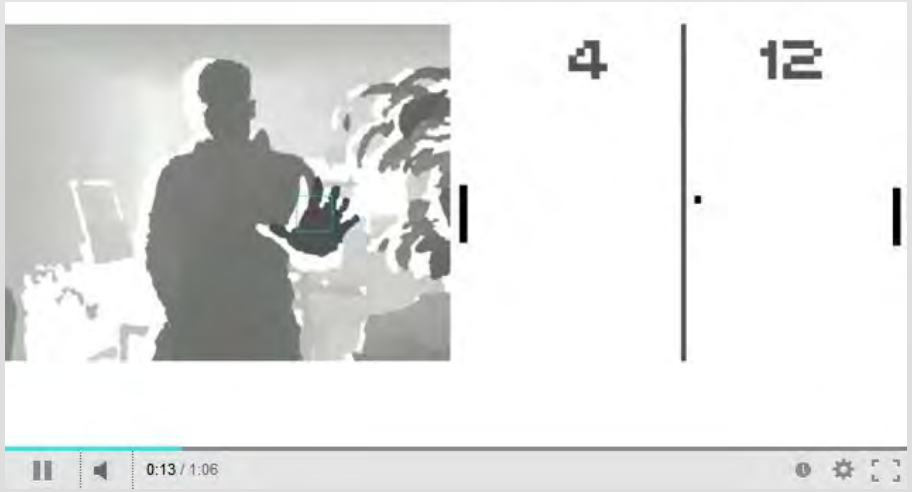
http://www.pixar.com/companyinfo/research/deb/

- **Was ist, wenn die Anzahl der Objekte sehr hoch wird?**
  - Beschreibung vieler Partikel durch statistische Variation der Vorgaben
  - oder durch Vorgabe von Regeln der Interaktion (Boids, Schwärme)



Aufgaben der Simulation in Spielen
6. Reaktion auf Interaktionen

- **... ist eine der Stärken der Simulation**
  (i. Vgl. zu Animation)
- **... denn ein dynamisches System kann jederzeit auf Änderungen reagieren**
  - ein intuitive, *allgemeines* Interface für eine 2D-Maus in einer 3D-Welt zu bauen ist allerdings nicht ganz einfach, erfordert ein wenig Computational Geometry
  - für starre Kameraperspektive sind aber einfache Lösungen möglich
  - Bsp. Game Physics mit Kinect



Sind Spiele Selbstzweck?

Technology Arts Sciences TH Köln

- **Manchmal kann ein Spiel auch wieder in sehr realer Anwendung münden:**
- **Bsp.: Virtueller OP für endoskopische Intervention**
  - mit Kraftrückkopplung durch kleine Roboterarme
  - virtuelle Blutungen über Partikelsysteme (!)

Simulation der Entfernung einer Gallenblase

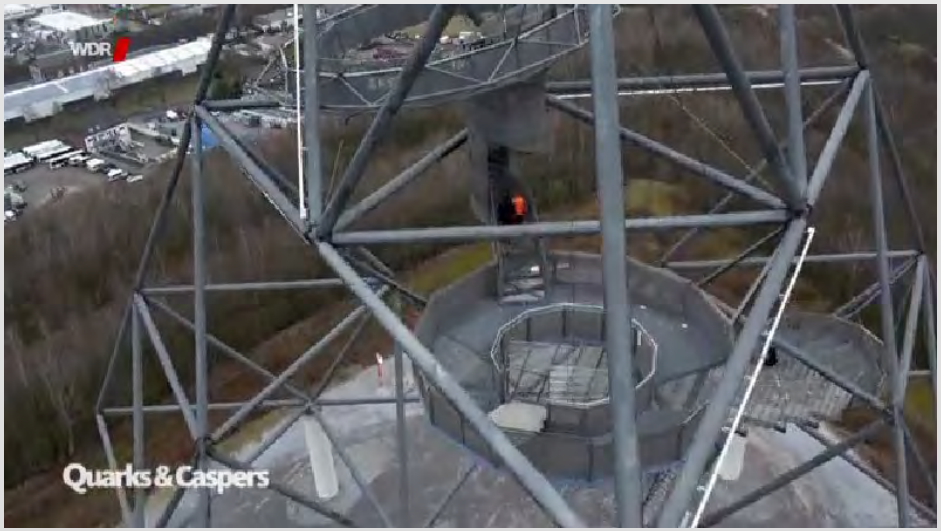
Tim Schröder: *Gallenblase im Simulator*, Spektrum der Wiss. 03/2004, S. 80 oder http://www.wissenschaft-online.de/abo/spektrum/archiv/7033 (ohne Bilder)

Oder 3-min Video, Quarks, VR im Operationssaal, 2018: https://www.youtube.com/watch?v=gH6mqbewPew

VR-Simulation gegen Höhenangst

Technology Arts Sciences TH Köln

- Eine VR-Therapie kann helfen, **Höhenangst** (oder Angst vor Spinnen) zu überwinden:
- 5min-Video, Quarks, VR-Therapie gegen Höhenangst: https://www.youtube.com/watch?v=Co8kj9EgotI



FAZIT: Was sollen Sie bei Simulation in Spielen lernen?

Technology Arts Sciences TH Köln

- **Weites und komplexes Feld, deshalb hier nur Grundlagen**
- **Trotzdem: einfache Dynamiken mit Processing in einfachen Programmen realisierbar**
- **Gerüst für einfache Processing-Programme**
- **Koordinatensysteme**
- **Wie man Kraftgleichungen in Programme umsetzt**
- **Wie man Federkräfte und Scheinkräfte umsetzt**
  (>> geschickte Wahl von Bezugssystemen vereinfacht das Leben)

FAZIT: Was sollen Sie bei Simulation in Spielen lernen?

Technology Arts Sciences TH Köln

- **Was Kollisionsdetektion und Kollisionsantwort unterscheidet**
- **Wie man eine 2D-Dynamik auf 3D verallgemeinert (Bsp. Softball)**

Übung zu Game Physics: „Bouncing Balls“ (1)

Technology Arts Sciences TH Köln

Ü1	**balls2D.pde: Processing-Projekt zum Laufen bringen**
Ändern Sie das Programm so ab, ...	
Ü2	**... dass keine Reflexion mehr an der Decke erfolgt**
Ü3	**... dass 8 Bälle, abwechselnd in der li. und re. Hälfte, erzeugt werden und jeweils nach links bzw. rechts fliegen**
Ü4	**... dass die Bälle in der Mitte an einer weiteren Wand reflektiert werden (Code-Snippet in balls2D.boxDraw() aktivieren)**
Ü5	**... dass eine Luftreibung (c_w=0.01) hinzukommt (*)** $\vec{F}_R = -c_W v^2 \frac{\vec{v}}{\lvert\vec{v}\rvert} = -c_W \lvert\vec{v}\rvert \vec{v}$
Ü6	**… dass bei Key ‚r‘ zwischen ‚random‘ Reflexion am Boden und deterministischer Reflexion gewechselt wird**

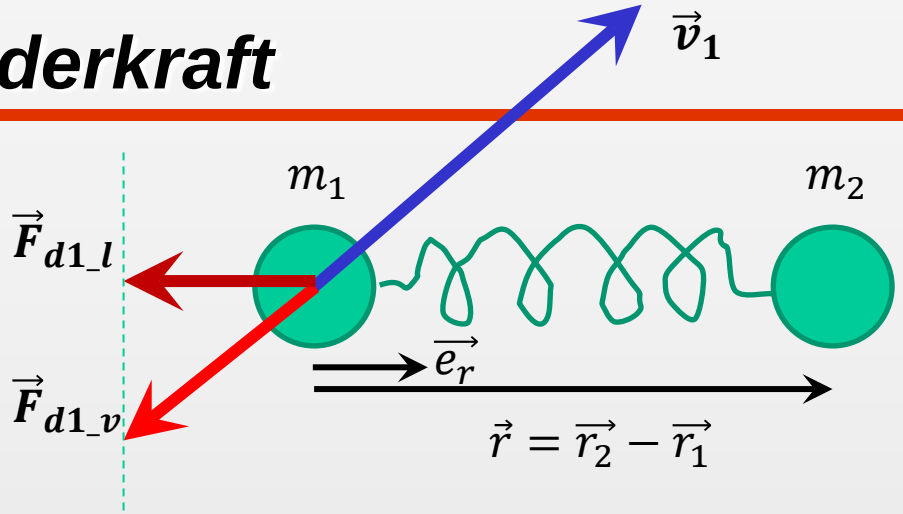
(*) Zufalls-Geschwindigkeit bei Bodenreflektion erhöhen: $\Delta v_x \in [-150,150]$, $v_y \in [-50,-350]$

Übung zu Game Physics: „Bouncing Balls“ (2)

Technology Arts Sciences TH Köln

Zusatzfragen:	
F1	**Gehen Bälle verloren, wenn Ü2 aktiviert wird?**
F2	**Zeitschrittweite so stark erhöhen, bis Artefakte sichtbar werden. Welche Artefakte, ab welcher Schrittweite?**
F3	**Warum verlieren die Bälle auch bei randomFloor=false und OHNE Luftreibung (Ü5) mit der Zeit langsam an Energie?**

Federkraft



$$\vec{e}_r = \frac{\vec{r}_2 - \vec{r}_1}{|\vec{r}_2 - \vec{r}_1|}$$

- **Federparameter: Ruhelänge l, Kraftkonstante k_s**
- **$\vec{F}_1$ zieht auf m_2 zu, wenn $|\vec{r}_2 - \vec{r}_1| > l$, sonst von m_2 weg**

$$\vec{F}_1 = \vec{e}_r k_s (|\vec{r}_2 - \vec{r}_1| - l)$$

- **Dämpfungskraft (Luftreibung) : $\vec{F}_{d1\_v}$ entgegen $\vec{v}_1$**

$$\vec{F}_{d1\_v} = -k_d \vec{v}_1$$

- **Oder (longitudinale Federreibung) entlang $\vec{e}_r$**

$$\vec{F}_{d1\_l} = -\vec{e}_r k_d (\vec{v}_1 \cdot \vec{e}_r)$$

Übung zu Federkraft (1)

Ü1	**Starten Sie mit balls2D.pde und reduzieren Sie (zunächst) auf einen Ball, ohne ‚random floor'.**
Ändern Sie das Programm so ab, ...	
Ü2	**... dass bei Mausklick eine Federkraft (+ Dämpfung) zwischen Maus und Ball wirkt**
Ü3	**... dass die Feder, je nachdem ob sie anziehend oder abstoßend wirkt, als grüne oder rote Linie gezeigt wird**
Ü4	**... dass die Dämpfung k_d über Slider zwischen 0% und 100% k_d einstellbar ist**
Ü5	**... dass die Federstärke k_s über Slider einstellbar ist**
Ü6	**(Optional: Erweitern Sie nun wieder auf mehrere Bälle)**

Übung zu Federkraft (2)

Technology Arts Sciences TH Köln

Zusatzfragen:	
F1	a) Was passiert ohne Dämpfung? b) Was ist besser, Luftreibung oder longitudinale Reibung?
F2	Können Sie die Parameter so einstellen, dass Sie mit minimalen Mausbewegungen den Ball um die Maus rotieren lassen können?
F3	Können Sie Artefakte beobachten, z.B. wenn Schrittweite zu groß? Erklärbar?
F4	Vergleichen Sie (ohne Dämpfung) Euler und semi-explizites Euler-Verfahren. Was ist besser?