

WPF Spiele, Simulation und dynamische Systeme

Lehrveranstaltung von Prof. Dr. Wolfgang Konen

Beginn 13.10.2011

Themenvorstellung im Detail

Zeit: Do, 10:00 – 13:30 Uhr

Raum: 3.113

Gliederung / Termine (Änderungen vorbehalten!)

Teil 1: Simulation

P: Projekt, S: Seminar (mit Demonstrationen), V: Vortrag Dozent

(Terminänderungen vorbehalten; s. auch zeitplan-WS11-dienstags.htm) Teil 1: 3 bis max. 4 S/P		Wieviele / Wer
13.10.11	Startup-Veranstaltung, Referate verteilen	Kg
V / Ü	Einführung in die Simulation	Kg
20.10.11	Simulation mit Tabellenkalkulation (mit Übungen)	Kg
V / Ü	Von der Zinseszinsrechnung zur Populationsdynamik. Diskretisierung: Von der DGL zur numerischen Lösung (m. Übungen)	Kg
27.10.11	Analytische Methoden: Linearisierung im Gleichgewichtspunkt	Kg
V / Ü	Konkurrenz der Arten und Bedingungen für Koexistenz (mit Übungen)	Kg
03.11.11		
10.11.11	evtl. noch V-Fortsetzung	Kg
Ergänzende Kapitel:		
V	Numerische Probleme bei der Simulation (Praxisbericht)	
	Simulation von Räuber-Beute-Systemen	

Teil 2: Game Physics

(Terminänderungen vorbehalten; s. auch zeitplan-WS11-dienstags.htm) Teil 2: Einf. OpenGL + 5 weitere S/P		Wieviele / Wer
V	Einführung Game Physics	Kg
24.11.11	Simulation für Spiele	Kg
S	Einführung in OpenGL und JOGL	N.N.

08.12.11		[Prtk: N.N.]
xx.12.11	- kein Vortrag, Referat- und Projektarbeit -	
P/S xx.01.12	3D-Softball-Simulation	N.N. [Prtk: N.N.]
S xx.01.12	Kollisionsdetektion und –antwort "Bouncing Balls"	N.N. [Prtk: N.N.]
P/S xx.01.12	Wieso zeigt der Mond der Erde immer dasselbe "Gesicht"?	N.N. [Prtk: N.N.]
S/P xx.01.12	Fahne/Vorhang simulieren	N.N. [Prtk: N.N.]
S/P xx.01.12	Invertiertes Pendel mit Stichsäge	N.N. [Prtk: N.N.]
S/P xx.01.12	Robocode	N.N. [Prtk: N.N.]
S/P xx.01.12	Game Physics mit Kinect (XBOX)	N.N. [Prtk: N.N.]
xx.01.12	Abschlussdiskussion und Feedback	alle
Ergänzende Kapitel:		
S	Einführung in die Simulation von Partikel- und Schwarmssystemen (BREVE)	
S	Dynamic Engines : ODE (Open Dynamics Engine), Newton Game Physics Engine, ...	
S (V)	(Wieso braucht man bei Game Physics gute Integrationsverfahren?)	
(V)	- kein Vortrag, Referat- und Projektarbeit -	

P: Projekt, S: Seminar (mit Demonstrationen), V: Vortrag Dozent

Startup-Veranstaltung

Typ: Vortrag und Demonstration

1. Einleitung , Beispiele vorstellen
2. Vorstellen der Unterrichtsformen
 1. Vorlesung
 2. Übungen (die alle bearbeiten, z.B. Excel-Blatt erstellen/modifizieren, Simulationsläufe)
 3. Referate der Studenten (am besten mit eingestreuten Übungen)
 4. Projektarbeiten der Studenten (am besten mit eingestreuten Übungen)
3. Bewertungskriterien (ich hoffe noch, auf **Klausur** verzichten zu können, wenn Engagement der Teilnehmer hoch genug. Referate und Projekte führen zu Anrechnungspunkten. Wenn allerdings zu wenig Engagement (bei einigen), dann Klausur)
 1. Jeder sollte Referat oder Projekt machen, das in Form einer ca. 2stündigen Lehrveranstaltung vorgestellt wird
 2. Idealerweise nicht nur Referat, sondern Lehrinheit (Übungen für alle)
 3. Bewertet wird

1. Verständlichkeit und Interessantheit des Vortrags (Handouts sind nur Anregung, nicht alles daraus muss gemacht werden, es kann/soll auch anderes eingebaut werden, ich lasse mich gerne von Ihnen überraschen)
2. eingestreute Übungen, Spiele, Simulationen: Wie gut war der Lernerfolg für die Teilnehmer?
3. Ausarbeitung: Form und Inhalt, pünktliche Bereitstellung (s.u)
4. zum Thema **1 Glossarbereich** in ILIAS (s.u.) einbauen
5. Protokoll je einer anderen Lehreinheit (s.u.) (20%)
4. Zur Verankerung der Lernziele schreibt jede Gruppe bei einer anderen Lehreinheit **Protokoll**: Die wichtigsten Facts als Handout, 1-2 Seiten, in ILIAS einzustellen, werden am Ende von allen Teilnehmern Teilnehmern benotet
5. 2 Personen je Thema, je **mind 1 Woche vor** Termin Treffen mit mir: Dann sollte Vortrag schon weitestgehend fertig sein!
6. Organisatorisches:
 1. Jeder, der teilnehmen will, bitte **bei ILIAS anmelden** <http://ilias.fh-koeln.de/start.php> (Fakultät 10, WPF 35 beitreten **UND** darin der **Gruppe WS11** beitreten) >> Mailingliste
 2. Bereitstellen der Ausarbeitung, **die zum Vortrag fertig sein muss (!)**: (i) in ILIAS durch Upload unter Gruppe WS11 **UND** (ii) in **Papierform** mitgebracht wird (einmal für mich).
 3. (Übrigens: Löschen der selbst hochgeladenen Dateien geht in ILIAS über "Administrationsmenüs AN")

Für alle Simulationsprojekte gilt:

- Simulieren Sie nicht einfach drauflos, sondern formulieren Sie vorher Hypothesen, wie Sie erwarten, dass das System sich verhält bei Variation der Parameter. Treffen die Erwartungen ein?
- Wo hat die Simulation Sie "überrascht" in Bezug auf Verhalten insgesamt, in Bezug auf Verhalten bei Variationen?
- Wie robust ist das System, welches sind "kritische Parameter"?
- Notieren Sie Ihre Ergebnisse und berichten Sie im Vortrag darüber!

7. Vorstellen der Kapitel und Themen
 1. Handout
 2. vorläufiger Terminplan
 3. Kommentiertes Literaturverzeichnis
8. Einteilung Referate/Projekte

[Einführung in die Simulation](#)

Typ: Vortrag (s. Link in Überschrift)

- entweder freier Vortrag mit Manuskript oder HTML-Seite am Beamer durchgehen
- Halbkreis am Rechner, Leiter (nach Denkpause) auch
- **Typeneinteilung von Simulationen**

[Simulation mit Tabellenkalkulation](#)

Typ: Vortrag Dozent (s. Link in Überschrift)

- entweder freier Vortrag mit Manuskript oder HTML-Seite am Beamer durchgehen
- Excel-Einführung unbedingt am Rechner
- Bewertung der Tabellenkalku als Simu-Tool: fragend-entwickelnd mit Studenten
- Übungen: Studenten entwickeln einfache Tab.kalkulationen am Rechner

- (s. Ende von [OeSi2.html](#))

Von der Zinseszinsrechnung zur Populationsdynamik und Diskretisierung: Von der DGL zur numerischen Lösung (m. Übung)

Typ: Vortrag Dozent (s. Link in Überschrift)

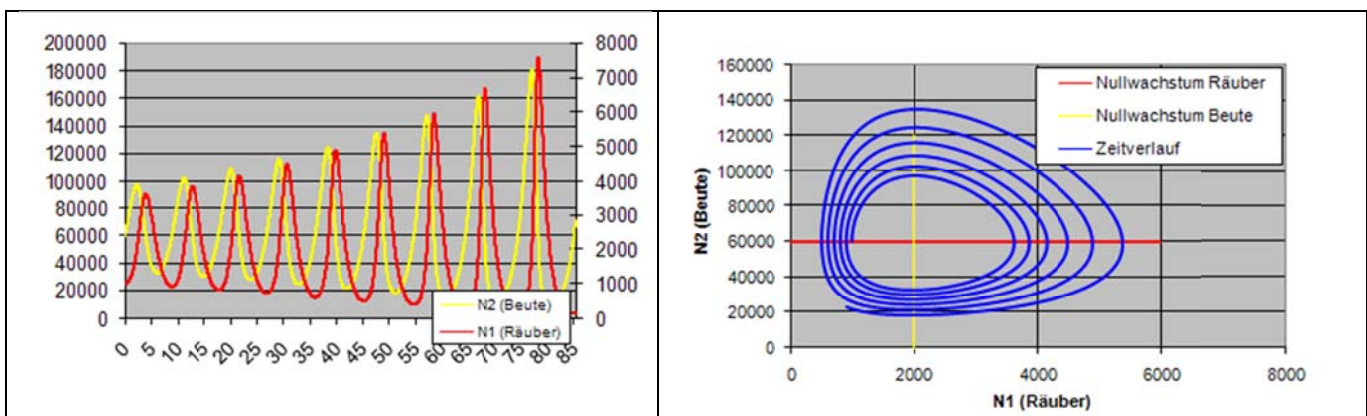
- entweder freier Vortrag mit Manuskript oder HTML-Seite am Beamer durchgehen
- Übungen: (a) Tab.kalku zu bewirtschafteter Population, (b) Tab.kalku zu $\frac{1}{2}gt^2$ -Sprung

Analytische Methoden. Linearisierung im Gleichgewichtspunkt

Typ: Vortrag Dozent (s. Link in Überschrift)

- entweder freier Vortrag mit Manuskript oder HTML-Seite am Beamer durchgehen
- **Vorlage Grams ist sehr kurz, evtl. ausführlicher** >> z.B. auf Beispiel Konkurrenz.xls (s. OeSi4.html) eingehen

Simulation von Räuber-Beute-Systemen



Typ: Seminar+Projekt (Lehrveranstaltung durch Studenten)

ZIEL: Durch Simulation Verständnis entwickeln für

- Nachhaltigkeit bei Nutzung von Ökosystemen
- die Effekte vernetzter (gekoppelter) Systeme

AUFGABEN:

- Durcharbeiten der Literatur, insbes. Handout [Struggle.html](#) und darin genannte Literatur
- Simulationsblatt LotkaVolterra.xls erstellen
- Lehrveranstaltung konzipieren
 1. Vortragsteil
 2. Demonstrationsteil (Simulationen vorführen)
 3. Übungsteil: Konzeption (sinnvoller und machbarer!) Übungen für die anderen Teilnehmer (z.B. gezielt einzelne Elemente im Excel-Blatt weglassen, gemeinsam mit Teilnehmern lösen)
- Es sollten zumindest einige der im Handout genannten Übungen 1.-8. bearbeitet werden (!), entweder durch die Vortragenden oder als Übung für alle Teilnehmer
- Es sollten über das Handout hinausgehende interessante Informationen aus der Literatur eingearbeitet werden.

MATERIALIEN:

- Handout [Struggle.html](#) und darin genannte Literatur

Einführung Game Physics

Typ: Vortrag Dozent

ZIEL/TOPICS:

- Als Aufhänger: **Crayon Physics**. Ein kleines [Spiel](#) mit großem WOW-Effekt, das Game Physics interaktiv erfahrbar macht
- Verschiedene Spiele-Engines vorstellen, s. [gamePhysics.ppt](#) und [Meqon Demo](#)
- Abgrenzung, Konzentration auf Physics Engines
- Animation vs. Simulation
- Einfaches GamePhysics-Beispiel: Dynamik in balls.exe ($1/2 gt^2$, Kraftgleichungen)
- Basics zu Geometrie einer 3D-Szene, Kamerageometrie
- Beispiele bringen: Billard, StarBall (Hinweis auf OpenSource), Softball, Pendulum
- Roter Faden für die folgenden Vorträge

Einführung in OpenGL und GLUT mit JOGL

Typ: Projekt/Referat

ZIEL:

- Grundkenntnisse zu OpenGL und GLUT vermitteln, damit WPF-Teilnehmer dies nachfolgend benutzen können (Quickstart).
- OpenGL-Geometrie verstehen
- KEINE umfassende Darstellung aller OpenGL-Möglichkeiten (mehrere 100 Funktionen!)

AUFGABEN:

- Einordnung OpenGL vs. GLUT. JOGL. Installationsbedingungen.
- Grundaufbau eines GLUT-Programms, erstes Beispiel.
- Einarbeiten in die JOGL-Arbeit [Dietzel07]. Erstellung einer Easy-to-use-Anleitung für „Gerüst“, die andere WPF-Teilnehmer benutzen können.
- Analyse der Funktionalität in [Dietzel07]: Was ist gut, was könnte man noch besser machen?
- Handling des HeavyWeightListeners verbessern: Maussteuerung / Resize auch im Fenster-Modus, auch im DebugModus.
- Übungen für die Teilnehmer konzipieren (überschaubare Änderungen an vorhandenem Programmgerüst)

OPTIONEN

- Wie simuliert man Licht / Shading / Reflektanzeigenschaften (Tutoriumsbsp)
- Wie simuliert man Textur auf einer 3D-Oberfläche?

MATERIALIEN:

- [Dietzel07] A. Dietzel: [Entwicklung eines JOGL-basierten Frameworks zur Darstellung von Game Physics](#), Diplomarbeit, FH Köln, 2007 ([Abstract](#) / [Paper](#) / [PDF](#)).
- M. Brill: Programmieren mit OpenGL und GLUT, <http://www.vislab.de/cgbuch/intros/glut.pdf>, lokale Version [hier](#).
- Tutorium unter <http://nehe.gamedev.net/>, Realisierung als GLUT-Programme.
- <https://jogl-demos.dev.java.net/>: JOGL-Demos mit Sourcen
- <http://pepijn.fab4.be/software/nehe-java-ports/>: NeHe-JOGL-Port, der auf dem neuen `javax.media.opengl.*`-Stand ist (den wir im WPF verwenden)
- GLUT-Tutorium unter <http://www.lighthouse3d.com/>
- Materialien auf www.opengl.org:

Kollisionsdetektion und Kollisionsantwort "Bouncing Balls"

Typ: Projekt

ZIELE:

- Erläuterung (und Trennung!) der Begriffe, wo liegen die Probleme?
- Kollisionsdetektion und –antwort an einfachem 2D-Beispiel demonstrieren
- optional: Vergleich verschiedener Ansätze zur Kollisionsdetektion

AUFGABEN:

- Ausgangspunkt ist 2D-Programm balls.cpp in GLUT. Ziel ist zunächst die Erläuterung und Demonstration des Basisprogramms, dann dessen Erweiterung um Kollisionsdetektion und Kollisionsantwort.
- Begriffe wie *line of action*, zentraler Stoß erläutern
- Erläuterung des gewählten Lösungsansatzes, etwaiger alternativer Ansätze
- Demonstration
- Lessons Learned
- Übungen / Weiterführende Fragen:
 1. Wie skaliert das Verhalten mit der Anzahl der Bälle?
 2. Wie verhält sich die Simulation bei zunehmend inelastischem Stoß? Bleiben die Bälle bei inelastischem Stoß immer aneinander kleben?
 3. Baut man die Kollisionsantwort in balls_V0.zip ein, so zeigt sich unrealistisches Kollisionsverhalten. Versuchen Sie herauszubekommen, woran das liegt.
 4. optional: BillardGL als Beispiel für ein Spiel mit Kollisionen vorstellen

MATERIALIEN:

- balls_WK.zip (modifizierte Variante WK), balls_V0.zip (Version aus dem Netz)
- Tutorium unter <http://nehe.gamedev.net/>, Lesson 30.
- [Bourg02] David M. Bourg: Physics for Game Developers, O'Reilly, 2002. Kapitel "Implementing Collision Response"
- Artikel zu "Collision Detection" unter <http://en.wikipedia.org> und darin genannte Literatur
- [Treitz04]: N. Treitz: *Leichtes Spiel mit dem Schwerpunkt, Physikalische Unterhaltungen*, Spektrum der Wiss. 8/2004, S. 101.

3D-Softball-Simulation

Typ: Projekt

ZIEL:

- Simulation der Dynamik eines 3D-Softballs, der aus Federn und Massenpunkten besteht und durch inneren Druck eine einstellbare Festigkeit erhält.

AUFGABEN:

- Ausgangspunkt ist ein beispielhaftes 2D-Programm (M. Matyka). Nachvollziehen der Physik im 2D-Beispiel
- Generieren der Ball-Triangulierung (Tessalation), Darstellung des 3D-Objektes in OpenGL
- Einbetten in 3D-Szenerie und Kamerabewegung (s. Gerüst aus OpenGL-Einführung)
- Übertragen der Kraftgleichungen (insbes. Volumenformel) auf 3D
- Erläuterung des gewählten Lösungsansatzes, etwaiger alternativer Ansätze
- Demonstration
- Lessons Learned
- Optional: Erweitern von Euler auf Heun

Diskussionspunkte / Erweiterungen:

- Was ändert sich, wenn die Federkonstante k_s verändert wird (verdoppeln oder halbieren)? Erklärung?

- Was ändert sich, wenn man die Dämpfungskonstante k_d weglässt? Erklärung?
- Bleibt der Ball auch stabil, wenn man die Anzahl der Punkte erhöht?
- Wie muss man die Druckkraft austarieren, damit der Ball bei mehr Punkten nicht immer "softer" wird?
- Um wieviel größer kann man die Zeitschrittweite DT einstellen, wenn man statt Euler- das Heun-Verfahren anwendet?

MATERIALIEN:

- M. Matyka: How To Implement a Pressure Soft Body Model, maq@panoramix.ift.uni.wroc.pl, March 30, 2004 ([softbody2d.zip](#))
- [softball2d_WK.zip](#) (geringfügig modifizierte Version WK)
- GLUT-Tutorium unter <http://www.lighthouse3d.com/> (Snowman-Beispiel)

Wieso zeigt der Mond der Erde immer dasselbe Gesicht?

Typ: Projekt

ZIEL:

- Es gilt, durch Simulation ein Verständnis dafür zu entwickeln, welche Mechanismen dazu geführt haben, dass der Mond sich gleichzeitig mit seiner Rotation um die Erde **genau 1x** (!!) um seine eigene Achse dreht, und wieso dieses Verhalten nicht "aus dem Takt gerät"

AUFGABEN:

- Literaturrecherche, darauf aufbauend:
- OpenGL-Simulation für die Gravitationsdynamik zweier Körper. Modellieren Sie (zunächst) das äquivalente Einkörperproblem, d.h. halten Sie die Erde im Ursprung des Koordinatensystems fest.
- Für den Mond: entweder ein prototypisches Beispiel, in dem die Mondexzentrizität durch eine "Hantel" (zwei Massenpunkte + Feder) ins Extrem getrieben wird,
- oder, realitätsnäher, Mond als starrer Körper mit Trägheitsmoment eines Rugby-Balls
- optional, noch realitätsnäher, Mond als anfangs noch deformierbarer Körper, der später erstarrt: Ergibt sich eine exzentrische Form als Ergebnis, ergibt sich "Locking" (Einrasten) der Rotationsfrequenz?
- verständliche Erläuterung des gewählten Lösungsansatzes, etwaiger alternativer Ansätze
- Demonstration
- Lessons Learned

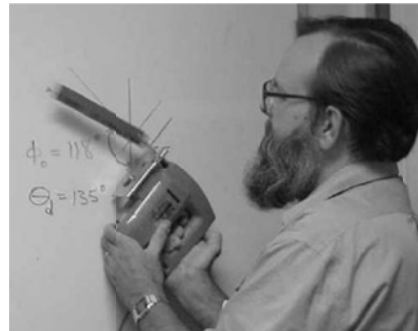
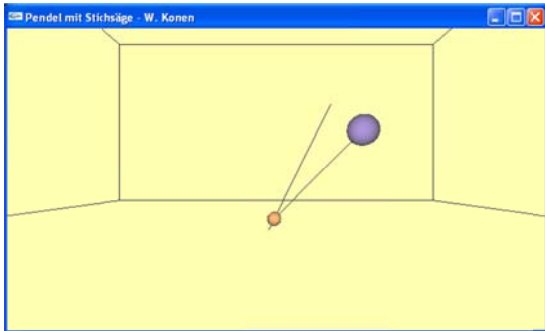
Diskussionspunkte / Erweiterungen:

- Welche anfängliche Tangentialgeschwindigkeit muss man dem Mond mitgeben, damit sich eine (annähernd) kreisförmige Bahn ergibt?
- Wann ergibt sich schneller eine "gebundene" Rotation des Mondes, wenn er gleichsinnig zu seinem Umlaufsinn rotiert oder gegensinnig? Erklärung?
- Wieso kommt es in der Simulation bei zu hoher anfänglicher Mondrotation zu einem Aufschaukeln zu immer schnellerer Rotation, bis der Mond "platzt"? Ist es ein numerischer oder ein physikalischer Effekt?
- **Interaktive** Simulation: Der Anwender soll mit Mausclick einen kurzen Störimpuls (z. B. mittels Federkraft) auf den Mond ausüben können. Was muss man beachten, damit die Mondbahn stabil bleibt, wann wird die Bahn instabil?
- **Bewegte Erde**: In der Realität wird die Erde ja auch durch den Mond mitbewegt. Welche Problematik ergibt sich in der Simulation, wenn wir das auch mitsimulieren? Wie kann man dem begegnen? Hat die Mitbewegung der Erde Einfluss auf die Schnelligkeit der Mondabbremmung?

MATERIALIEN:

- im WWW unter "Mond gebundene Rotation" recherchieren
- www.wonderquest.com/MoonSpin.htm oder seds.lpl.arizona.edu/nineplanets/nineplanets/luna.html)
- A.M. Quetz: *Amalthea* in: Sterne und Weltraum, Nov. 2002, S. 92.
- A.M. Quetz: *Gezeitenreibung* in: Sterne und Weltraum, 2/1998, S. 294.
- A.M. Quetz: *Gezeitenreibung, Teil 2* in: Sterne und Weltraum, 5/1998, S. 492.

Invertiertes Pendel mit Sticksäge



Typ: Projekt

ZIEL:

- In [Pöppe03] steht, wieso ein aufrecht stehendes Pendel allein durch eine Sticksäge – ohne jede "intelligente" Regelung (!) – stabilisiert werden kann.
- Durch Game-Physics-Simulation soll anschaulich demonstriert werden, dass dieser Effekt wirklich gilt

AUFGABEN:

- Durcharbeiten der Literatur, insbes. [Pöppe03] und darin genannte Literatur
- OpenGL-Simulation für invertiertes Pendel: Linie als masselose Stange und je eine Kugel für den Fixationspunkt und die Punktmasse.
- optional: realistischerer Look durch Pendelstange aus GL_QUADS
- Umsetzung der "Game Physics". Am geschicktesten formuliert man die Bewegungsgleichungen im mitbewegten Bezugssystem (Scheinkräfte beachten!). Variable: Auslenkungswinkel bzw. Bogenlänge.
- HINWEIS: am besten zusätzliche Reibungskraft einbauen, damit man sieht, wohin sich die Lösung einschwingt.
- verständliche Erläuterung des gewählten Lösungsansatzes, etwaiger alternativer Ansätze
- Demonstration
- Lessons Learned

Diskussionspunkte / Erweiterungen:

- **Interaktive** Simulation: Der Anwender soll mit Mausclick einen kurzen Störimpuls (z. B. mittels Federkraft) auf die Pendelmasse ausüben können. Was muss man beachten, damit das Pendel stabil bleibt?
- Was passiert, wenn man die Sticksäge schräg hält? >> Simulation erweitern
- Übung für die Teilnehmer: bei schräger Sticksäge, Neigungswinkel β , durch Ausprobieren herausfinden, bei welcher Anfangsneigung $\varphi(\beta)$ das Pendel am schnellsten zur Ruhe kommt. Kann man $\varphi(\beta)$ berechnen?
- Sei $\beta=11^\circ$, $g=9\text{ms}^{-2}$, $L=14\text{m}$, $\omega=72\text{s}^{-1}$. Bei welcher Amplitude (Hub der Sticksäge) bricht das invertierte Pendel zusammen?
- Könnte man durch Regelung der Amplitude oder Frequenz der Sticksäge in Abhängigkeit vom Auslenkungswinkel erreichen, dass das Pendel schneller invertiert zur Ruhe kommt?
- optional: Interaktives Spiel, bei dem der User diese Regelung steuert. Ziel: schnellstmöglich kleine Winkel.
- Hält auch eine horizontale Sticksäge das Pendel fern seiner Ruhelage?

MATERIALIEN:

- [Pöppe03] C. Pöppe, *Die stabilisierende Wirkung der Stichsäge*. *Physikalische Unterhaltungen*, Spektrum der Wiss. 12/2003, S. 110.
- [VanDalen03] G.J. VanDalen, *The Driven Pendulum at Arbitrary Drive Angles*. Juni 2003, www.arxiv.org, physics, 0211047.

Fahne oder Vorhang simulieren

Typ: Projekt

ZIEL:

- In [Bourg02] steht, wie man eine Fahne simulieren könnte.
- Durch Game-Physics-Simulation ausprobieren, wie weit dieser Effekt trägt

AUFGABEN:

- Durcharbeiten der Literatur, insbes. [Bourg02] und darin genannte Literatur
- OpenGL-Simulation für Fahne oder Vorhang entwickeln.
- optional: realistischerer Look durch Texturierung
- optional: Reaktion der Fahne auf Wind oder andere Gegenstände (Bsp. s. Meqon-Demo)
- verständliche Erläuterung des gewählten Lösungsansatzes, etwaiger alternativer Ansätze
- Demonstration
- Lessons Learned



MATERIALIEN:

- [Bourg02] David M. Bourg: *Physics for Game Developers*. O'Reilly, 2002. Bringt viele physikalische Grundlagen. Im Kapitel "Particle Systems" interessantes Beispiel zur Simulation einer **Fahne** (Analogon Vorhang).

Robocode

Typ: Projekt

ZIEL:

- Eine Robocode-Competition unter den WPF-Teilnehmern (+ Sample-Robots) organisieren
- Verständnis für Multi-Agenten-Spiele entwickeln und wie man sie auswertet

AUFGABEN:

- Kurzeinführung, die WPF-Teilnehmern die Grundzüge Robocode und die Competition erklärt
- Vortrag zu Robocode und „Drumherum“
- Competition evaluieren, analysieren
- Eigene Robotanks bauen und ihre Erfolge/Misserfolge analysieren
- Ansätze zur automatischen Generierung von Robotanks: [Shichei05] und [Harper11] studieren und Grundideen in Vortrag erläutern
- Demonstration
- Lessons Learned

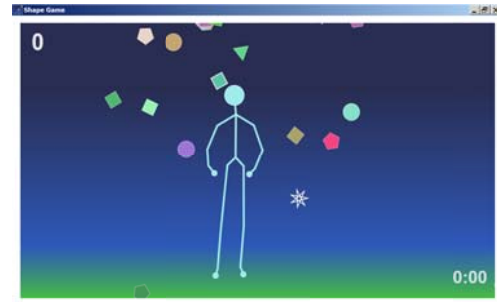
MATERIALIEN:

- <http://robocode.sourceforge.net/>



- [Shichel05] Shichel, Y.; Ziserman, E. & Sipper, M.: [GP-Robocode: Using Genetic Programming to Evolve Robocode Players Proceedings of the 8th European Conference on Genetic Programming](#), 2005, 3447, 143–154
- [Harper11] Harper, R: [Co-Evolving Robocode Tanks](#). GECCO '11: Proceedings of the 13th annual conference on Genetic and evolutionary computation, 2011, 1443

Game Physics mit Kinect (XBOX)



Typ: Projekt

ZIEL:

- Einführung in die Kinect Sensoreinheit und Kinect SDK
- Durch Game-Physics-Simulation ausprobieren, wie weit dieser Effekt trägt

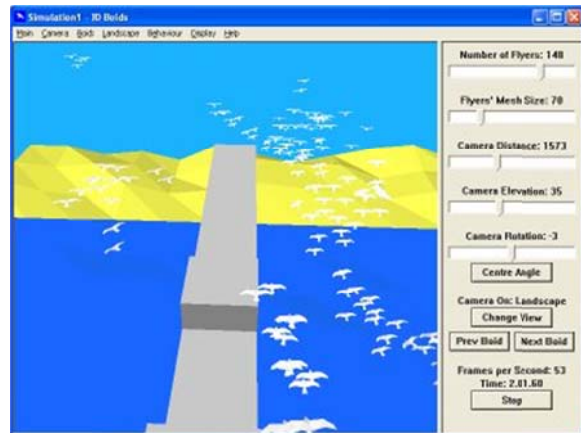
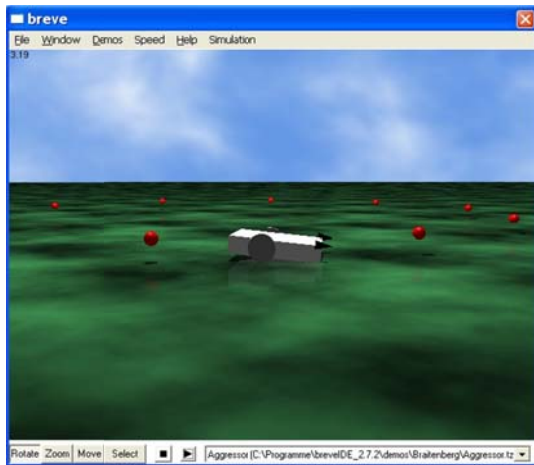
AUFGABEN:

- Durcharbeiten der Literatur, insbes. [TODO] und darin genannte Literatur
- Funktionsprinzipien der Kinect erklären
- An ersten einfachen Beispielen Kinect SDK und Visual C# kennenlernen
- Grundlagen zum Human Body Model in Kinect for Windows SDK Beta recherchieren und erklären
- Beispielanwendung **GameShape**: Game Physics darin erklären und verändern (z.B. Anzahl Objekte kleiner und dann Objekte kollidierbar machen)
- Beschleunigungspotentiale GameShape ausloten
- Sind 2 Spieler möglich?
- ODER auf Basis von GameShape eigenes Minimal-Spiel schreiben (Bälle, die mit der Hand bewegt werden)
- verständliche Erläuterung des gewählten Lösungsansatzes, etwaiger alternativer Ansätze
- Demonstration
- Lessons Learned

MATERIALIEN:

- <http://research.microsoft.com/en-us/um/redmond/projects/kinectsdk/download.aspx>
- <http://research.microsoft.com/apps/video/default.aspx?id=150286>
-

Simulation von Partikel- und Schwarm- und Agentensystemen



Typ: Referat

ZIEL:

- Begriffe Partikel- und Schwarmssysteme verstehen
- Verständnis für Simulation von Agenten / interagierenden Vehikeln entwickeln

AUFGABENSTELLUNG:

- Überblick über das Gebiet der Partikelsysteme
- Einarbeiten in BREVE, Erklären Grundbegriffe
- evtl. Schwarm-Demo (kurz)
- Braitenberg-Tutorial aufbereiten für Vortrag, hierbei aber auch Elemente aus Braitenberg.tz erklären (!)
- Entwickeln Sie weitere Braitenberg-Vehikel (Ideen finden Sie in [Pfeifer99]!). Z.B. eine Simulation mit mehreren Vehikeln oder Vehikel, die selbst ein Licht tragen, oder ...
- Übungen hierzu

OPTION

- Ausblick PSO (**P**article **S**warm **O**ptimization)

MATERIALIEN:

- s. Abschnitt "Partikel- u. Schwarmssysteme" im [Kommentierten Literaturverzeichnis](#), insbesondere [Klein03], Braitenberg-Tutorial, [Pfeifer99].
- Applet mit schöner Visualisierung PSO in <http://gecco.org.chemie.uni-frankfurt.de/PsoVis/applet.html>

Dynamic Engines: ODE (Open Dynamics Engine), Newton Engine, meqon Game Physics Engine, ...

Typ: Referat / Projekt

ZIEL:

- Überblick ODE, Einsatzbereiche
- Überblick Newton Engine, Einsatzbereiche, Konzept der SDK
- Vor- und Nachteile zu anderen Simulationsumgebungen diskutieren

AUFGABEN:

- Literaturstudium (angegebene Lit. & eigene Recherche)
- Durcharbeiten [Tutorial Newton in ILIAS](#) (Praxisprojekt Blache/Müller), Auswählen von Elementen daraus für Workshop,
 - Prüfen der Einsetzbarkeit auf Rechnern Mathe-Labor

- Kritische Durchsicht der Inhalte (Ist alles richtig? Ist alles für Newcomer im Gebiet verständlich?)
- Eigene Erweiterungen vorschlagen, Übungen für die Teilnehmer konzipieren
- erfordert C++-Kenntnisse und OpenGL unter C++

MATERIALIEN:

- zu ODE: [\[Flintrop03\]](#) und darin genannte Referenzen zu ODE
- www.ode.org
- www.newtondynamics.com
- [eLearning-Tutorial „Newton“ in ILIAS](#)
- www.meqon.com/downloadarea/meqon-demos-1.4.zip

Wieso braucht man bei Game Physics gute Integrationsverfahren?

Typ: Referat

AUFGABENSTELLUNG:

- Motivation: An 2D-Softball o.ä. kann man sehen, wie schnell das Euler-Verfahren zum völligen "Kollabieren" führen kann.
- Problematik erläutern. Zeigen, was passiert. Verbesserungsverfahren vorstellen: Euler, Heun, Runge-Kutta >> vgl. Untersuchungen an OpenGL-Beispiel
- Prinzip der steifen DGL an einfachem Beispiel (Tab.Kalku?)
- Erläuterung des gewählten Lösungsansatzes, etwaiger alternativer Ansätze
- Demonstration
- Lessons Learned

MATERIALIEN:

- [\[Berchtold03\]](#) Kurze Vorstellung von Heun, Euler und Runge-Kutta. Lokale Kopie in [DGL-Heun-Euler.pdf](#).

Für alle Simulationsprojekte gilt:

- Simulieren Sie nicht einfach drauflos, sondern formulieren Sie vorher Hypothesen, wie Sie erwarten, dass das System sich verhält bei Variation der Parameter. Treffen die Erwartungen ein?
- Wo hat die Simulation Sie "überrascht" in Bezug auf Verhalten insgesamt, in Bezug auf Verhalten bei Variationen?
- Wie robust ist das System, welches sind "kritische Parameter"?
- Notieren Sie Ihre Ergebnisse und berichten Sie im Vortrag darüber!

[Kommentiertes Literaturverzeichnis](#)

Ergänzende Themen

[Lösungen.doc](#)