WPF Spiele, Simulation und dynamische Systeme

Lehrveranstaltung von Prof. Dr. Wolfgang Konen
WS 2024, Beginn 16.10.2024

Themenvorstellung im Detail

Zeit: Mi, 09:00 bis 12:00 od. 12:30 Uhr Raum: 3.113

(Vertiefung zu info_neu.html, das eine allgemeine Einführung in dieses WPF gibt)

Gliederung / Termine (Änderungen vorbehalten!)

Teil 1: Simulation

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

(Terminänderungen vorbehalten)		Wieviele / Wer
16.10.24	Startup-Veranstaltung, Projekte verteilen	Kg
V / Ü 23.10.24	Einführung in die Simulation	Kg
	Simulation mit Tabellenkalkulation (mit Übungen)	Kg
V / Ü 30.10.24	Von der Zinseszinsrechnung zur Populationsdynamik. Diskretisierung: Von der DGL zur numerischen Lösung (m. Übungen)	Kg
	Analytische Methoden: Linearisierung im Gleichgewichtspunkt	Kg

Teil 2: Game Physics

(Terminänderungen vorbehalten)		Wieviele / Wer
V / Ü 06.11.24	Einführung in Processing Tutorial 3D-Simulation in Processing	Kg
V 13.11.24	Einführung Game Physics	Kg
	Simulation für Spiele, Ü balls2D + Federkraft	Kg
		Kg
xx.11.24	- kein Vortrag, Referat- und Projektarbeit -	
11.12.24	Milestone-Treffen zu den Projekten 1) – 4) (Raum 3.113 oder ZOOM)	

S/P 18.12.24	1) Räuber-Beute Simulation 2) Fahne / Vorhang	Akkus, Siemens, Heidenreich *Cremer, Merzouk, *Tubies
31.12.24	**Deadline Abgabe ChatGPT-Diskurs**	jeder Studierende einzeln
S/P 08.01.25	**Blockwoche** Milestone-ZOOM zu den Projekten 5) – 6)	
S/P 15.01.25	3) Kollisionsdetektion 4) 3D-Softball-Simulation	*Sorg, *Foerst Dag, van Tuijl
S/P 22.01.25	5) Kollisionsdetektion 6) Game Physics mit Kinect 2	*Kurilin, *Pivovar *Jehle, *Steinbach, Wegner
29.01.25	Abschlussdiskussion und Feedback	alle
Projektthemen (zu Teil1 + Teil 2):		
S/P	Kollisionsdetektion	~~(Klesen)~~, *Sorg, *Foerst
S/P	Kollisionsdetektion	*Kurilin, *Pivovar
P/S	3D-Softball-Simulation	Dag, van Tuijl
P/S	Fahne / Vorhang	*Cremer, Merzouk, *Tubies
S/P	Räuber-Beute Simulation	Akkus, Siemens, Heidenreich
S/P	Game Physics mit Kinect 2	*Jehle, *Steinbach, Wegner
S/P	~~Game Physics mit Kinect 2~~	~~(Colette, Gonzalez, Trinh)~~
S/P	Invertiertes Pendel mit Stichsäge	NN, NN
P/S	„Smoothe“ Rotation mit Quaternionen	NN, NN
P/S	Mond-Erde	NN, NN
P/S	Robocode	NN, NN
S/P	Integrationsverfahren für Game Physics	NN, NN
S/P	~~Haare simulieren~~	(~~Heidenreich~~)

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. Projektarbeiten / Referate der Studierenden (am besten mit eingestreuten Übungen)
3. Bewertungskriterien
    1. Jede(r) sollte Referat oder Projekt machen, das in Form einer ca. 1stündigen Lehrveranstaltung vorgestellt wird
    2. Idealerweise nicht nur Referat, sondern Lehreinheit (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. Elemente 1.-3. machen 75% der Note aus.
        5. Unabhängig vom Projekt fertigt jeder Studierende einen ChatGPT-Diskurses an (s.u.) (25% der Note)
4. 2 (evtl. 3) Personen je Thema, je **mind. 1 Woche vor** Termin Treffen mit mir: Dann sollte Vortrag schon weitestgehend fertig sein!
5. Organisatiorisches:
    1. Jeder, der teilnehmen will, bitte **bei ILU im WPF SSDS anmelden**: F10, WPF SSDS beitreten **UND** darin der **Gruppe WS2024** beitreten) >> Mailingliste
    2. Bereitstellen der Ausarbeitung, **die zum Vortrag fertig sein muss (!)**: in ILU durch Upload unter Gruppe WS2024. Etwaige Beispielprogramme / Übungen für die Lehreinheit können dort auch hochgeladen werden.
6. **ChatGPT-Diskurs**:
    1. Auch ChatGPT simuliert (manche sagen: halluziniert) seine Antworten. Passt also zum WPF 😊
    2. Eines der wichtigsten Lernziele im Zusammenhang mit ChatGPT: Dass Studierende lernen, die Antworten von ChatGPT zu bewerten und in ihr Wissen einzuordnen.
    3. Als THK-Studierende haben Sie mit **THKI-Lab**: via **https://ki.th-koeln.de** (nur im TH-Netz (VPN)) kostenlosen Zugang zu ChatGPT. **Ihre Aufgabe**:
    4. Verwickeln Sie ChatGPT in eine Diskussion über ein Thema, das im weitesten Sinne zu SSDS gehört. Versuchen Sie Fragen bzw. Aufgaben zu stellen, bei denen die Antworten gut bewertet werden können (Richtigkeit, Vollständigkeit etc.) und die nicht zu einfach zu beantworten sind.
    5. Dokumentieren Sie den Chat-Verlauf. Es muss nicht viel sein, 1-3 Textseiten reichen.
    6. Jetzt kommt Ihre eigentliche Aufgabe: **Bewerten Sie** die von ChatGPT erhaltenen Antworten: Sind sie richtig? D.h. können Sie die Richtigkeit mit anderen Quellen (diese benennen!) verifizieren? Sind sie hinreichend vollständig? Hat ChatGPT nur ‚geschwafelt' oder haben

Sie auch etwas interessantes Neues und Richtiges gelernt? Ordnen Sie die Antworten von ChatGPT so in Ihren Wissenskontext ein.
7. Reichen Sie Ihr Ergebnis (Doku Chat-Verlauf + Ihre Bewertung) als PDF bis zur **Deadline 31.12.2024** ein.
8. Bewertungskriterien für PDF: Originalität der Fragen, Stimmigkeit der Bewertung, Vollständigkeit der Bewertung.

7. **WASP (WAhl-SPezialisierung)**, also WPF I + WPF II:
   1. In der neuen Studienordnung (zumindest INF) gibt es WPFs (5 CP) in Verbindung mit einem größeren Projekt (10 CP).
   2. Ich kann in begrenztem Umfang (≤ **4**) solche 10-CP-Projekte zusätzlich zum WPF betreuen.
   3. Themen, für die eine solche Projekt-Erweiterung inhaltlich sinnvoll ist, sind weiter unten mit * markiert.
   4. Neben dem WPF-Teil (5*30h = 150h) sollte dann jeder am Projekt teilnehmende Studierende noch 10*30h = 300h leisten.
   5. Jede Projektgruppe: Am Anfang Zeitplan und Exposé, regelmäßige Statustreffen mit dem Dozenten, in denen Arbeitsfortschritt vorgestellt wird und Fragen und weitere Arbeiten diskutiert werden.
   6. Abschluss durch Projektdokumentation und evtl. -präsentation.

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!

8. Vorstellen der Kapitel und Themen
   1. Handout
   2. vorläufiger Terminplan
   3. Kommentiertes Literaturverzeichnis
9. 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 ½gt²-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

Einführung in Processing

Typ: Vortrag Dozent / Praktische Ü am Rechner
ZIEL/TOPICS:
- Einfuhrung Processing.pdf.lnk nach A. Rennertz
- alle: Basic Tutorials von www.processing.org durcharbeiten
  - Coordinate System and Shapes (Drawing)
  - Color
  - 2D-Transformations
- bestimmte Übungen realisieren

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
- Abgrenzung, Konzentration auf Physics Engines
- Animation vs. Simulation
- Einfaches GamePhysics-Beispiel: Dynamik in balls.exe (1/2 gt², Kraftgleichungen)
- optional: Ü zu Federkraft + Mouse (2D), die auf Ecke von Quader wirkt.
- ~~Beispiele bringen: Billard, StarBall (Hinweis auf OpenSource), Softball, Pendulum~~
- Roter Faden für die folgenden Vorträge

Tutorial 3D-Simulation in Processing

Typ: Vortrag Dozent / Praktische Ü am Rechner
ZIEL:
- Grundkenntnisse zu Processing vertiefen.
- 3D-Geometrie (OpenGL, Processing) verstehen
- KEINE umfassende Darstellung aller OpenGL-Möglichkeiten (mehrere 100 Funktionen!)

AUFGA BEN:
- Typisierung 3D-Renderer (OpenGL, JOGL, Processing).
- Basics zu Geometrie einer 3D-Szene, Kamerageometrie, Koordinatensysteme
- Kamera und Kamerasteuerung (`camera` vs. `PeasyCam`)
- Maussteuerung für 3D-Szene: am Beispiel Federkraft überlegen, wie das intuitiv machbar ist.
- 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:
- Materialien auf www.processing.org und dort genannte Bücher, Links
- [**Bohnacker+09**] Hartmut Bohnacker Benedikt Groß, Julia Laub: *„Generative Gestaltung - Entwerfen. Programmieren. Visualisieren“*, Verlag Schmidt, 2009. Schön gestaltetes Buch, viele Abbildungen und Beispiele, mehrfach in Bib CGM vorhanden.
- [**ReasFry07**] Casey Reas, Ben Fry: *„Processing: A Programming Handbook for Visual Designers and Artists”*, MIT Press, 2007. PDF im ILU zum WPF - Ordner "Materialien", *nur WPF-interne Nutzung, nicht weitergeben*. „Extension 2: 3D“ ab S. 551 bringt Beispiele und weiterführende Materialien zu 3D (in Processing).
- www.generative-gestaltung.de: Webseite zum Buch „Generative Gestaltung“ [Bohnacker+GG09], enthält alle Codebeispiele zum Buch als ZIP.
- 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
- Materialien auf www.opengl.org

Projektthemen

(jeweils von 2er- oder 3er-Teams zu bearbeiten)

[Projektthemen mit Stern *THEMA sind WASP-geeignet unter Einbeziehung der jeweiligen WASP-Erweiterung (oder anderer Erweiterungsvorschläge von Ihnen, die mit dem Betreuer abgesprochen werden)]

*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 balls2D.pde in Processing. 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
- Weiterführende Fragen:
  - Wie skaliert das Verhalten mit der Anzahl der Bälle?
  - Wie verhält sich die Simulation bei zunehmend inelastischem Stoß? Bleiben die Bälle bei inelastischem Stoß immer aneinander kleben?
  - Baut man die Kollisionsantwort in balls_V0.zip ein, so zeigt sich unrealistisches Kollisionsverhalten. Versuchen Sie herauszubekommen, woran das liegt.
  - optional: erweitern auf 3D-Bälle in einer „confining box“

- **WASP-geeignete Erweiterung**:
  - Bau eines (einfachen) Billard-Demonstrators **in 3D**: Szene aus verschiedenen Blickwinkeln betrachtbar, aber Bälle rollen auf 2D-Tisch. Geeignete Interaktions-Varianten entwickeln, um Ball 1 auf Ball 2 zu schießen (viele Möglichkeiten sind denkbar), …
  - Erweiterung auf andere Objekte als Bälle, die kollidieren

MATERIALIEN:

- balls2D.pde.zip (Processing-Variante WK, ohne Kollision)
- [Christ14] D. Christopoulos, J. Molofee: "Collision Detection", NeHe Tutorial http://nehe.gamedev.net/ , Lesson 30, 2014.
- [Bourg02] David M. Bourg: Physics for Game Developers, O'Reilly, 2002. Chapter "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.

*„Smoothe" Rotation mit Quaternionen

Typ: Projekt

In der Computergraphik stößt man oft auf folgendes Problem: Ein Objekt (Teekanne, Würfel, …) in 3D liegt in zwei verschiedenen Rotationen A und B vor. Nun möchte man dazwischen interpolieren, d.h. das Objekt auf möglichst „smoothem" Pfad von A in B überführen. Das ist bei Rotationen leider viel schwieriger als bei Translationen! (Warum das so ist, sollen Sie in diesem Projekt auch herausfinden.)

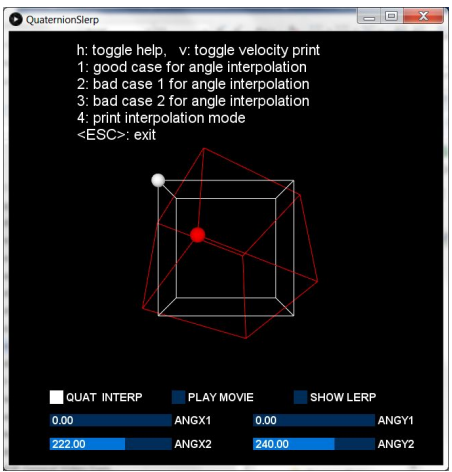
Die Rettung naht in Gestalt der Quaternionen. Diese sind zwar zuerst etwas kompliziert (sie wurden als sogenannte hyperkomplexen Zahlen (Zahlenmenge H) vom irischen Mathematiker Sir William Rowan Hamilton hat sie im 19.Jh. erfunden). Sie machen aber nachher vieles einfacher, besonders bei der Rotation.

ZIELE:

- Grundlegendes Verständnis Quaternionen + Rotation mit Quaternionen
- Wieso ist die Interpolation bei Drehungen schwierig?
- optional: Vergleich verschiedener Ansätze zur Interpolation

AUFGABEN:

- Ein 3D-Objekt soll von einer Anfangsposition A in eine Endposition E gedreht werden. Dies ist in Spielen oftmals eine Anforderung. Sie als Gestalter der Game Physics Engine sollen die Zwischenpositionen berechnen. Diese Interpolation soll möglichst „smooth" und möglichst einfach von A nach E führen.
- Warm-Up: Grundlegendes zu Rotationsmatrix und Drehwinkel erläutern.
- Zeigen Sie, dass eine „smoothe" Interpolation über Rotationsmatrizen oder Drehwinkel oft nicht möglich bzw. problematisch ist.
- Quaternionen, Rotation mit Quaternionen und SLERP (sphärische lineare Interpolation) erklären.



- Erläuterung des gewählten Lösungsansatzes, etwaiger alternativer Ansätze
- Demonstration
- Lessons Learned
- **WASP-geeignete Erweiterung**: Bau eines umfangreichen Demonstrators, der verschiedene Rotationsansätze (ROT, LERP, SLERP, NLERP, …) vergleicht und wissenschaftlich untersucht
  - Zum Beispiel sollte die Tabelle bei [Blow04] mit den Eigenschaften (commutative, constant

velocity, torque-minimal) durch den Demonstrator ‚erfahrbar' gemacht werden.
- Eigenschaften wie „constant velocity" (von was?) wissenschaftlich fundiert messen.
- optional: erweitern auf mehrfache Interpolation: von A nach B … nach E

MATERIALIEN:
- Zum leichteren Einstieg in die Rotation mit Processing steht ein Template QuatTemplate.zip zur Verfügung. Alles mit Quaternions und weiterführende Ansätze müssen Sie allerdings dort noch ergänzen.
- [Shoemake85] Ken Shoemake, *Animating rotation with quaternion curves*, Computer Graphics (Proc. of SIGGRAPH '85) 19, no. 3, pp. 245-254, 1985. (local copy)
- [Shoemake92] Ken Shoemake: Quaternion tutorial, als PDF: www.cs.ucr.edu/~vbz/resources/quatut.pdf
- https://en.wikipedia.org/wiki/Slerp: about SLERP and Quaternions
- [Blow04] Jonathan Blow, Understanding Slerp, Then Not Using It, blog on "Inner Product". **Very good, for game programmers!** Last update: Feb'2004. Last access: Oct'2024 (local copy)
- [DamKoch98] Dam, E., Koch, M., Lillholm, M.: Quaternions, Interpolation and Animation, TR Univ. Copenhagen. A bit mathematical, but interesting for Chapter 4: qualitative comparison of quaternions, Euler angles and matrices. (local copy)

*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 Processing
- 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

WASP-geeignete Erweiterung:
- Verschiedene **Methoden zur Tessalation** (Ringe, Tetraeder, Ikosaeder) implementieren und vergleichen
- 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?
- 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?
- Als Alternative zum **Integrationsverfahren** Euler das Heun-Verfahren implementieren und fundiert testen: Laufzeiten messen (evtl. als Funktion der Punkte). Um wieviel kann man die Zeitschrittweite DT erhöhen (ohne dass der Ball „platzt"), wenn man statt Euler- das Heun-Verfahren anwendet?
- Entwickeln einer **intuitiven Maus-Interaktion** mit dem 3D-Ball: Jeder Klick mit der Maus sollte in einen ‚sinnvollen' 3D-Punkt übersetzt werden. Was sinnvoll ist, hängt von der Orientierung der Kamera ab. Man kann dann durch die Maus auf den Ball einwirken, indem für die Dauer des Maus-Klicks eine Feder zum Ball gespannt wird

und den Ball anzieht.

MATERIALIEN:

- [Matyka] M. Matyka: *How To Implement a Pressure Soft Body Model*, March 30, 2004 (PDF)
- softball2D.pde.zip (korrigierte und auf Processing übertragene Version WK)

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:
- Processing-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 Mondexzentrität 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
- Semi-explizites (auch: semi-implizites) Eulerverfahren erläutern
- 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 Mausklick 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 Mondabbremsung?

MATERIALIEN:

- im WWW unter "Mond gebundene Rotation" recherchieren
- http://en.wikipedia.org/wiki/Semi-implicit_Euler_method
- 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.

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 Developpers*. O'Reilly, 2002. Bringt viele physikalische Grundlagen. Im Kapitel "Particle Systems" interessantes Beispiel zur Simulation einer **Fahne** (Analogon Vorhang).

Robocode

Typ: Projekt
ZIEL:

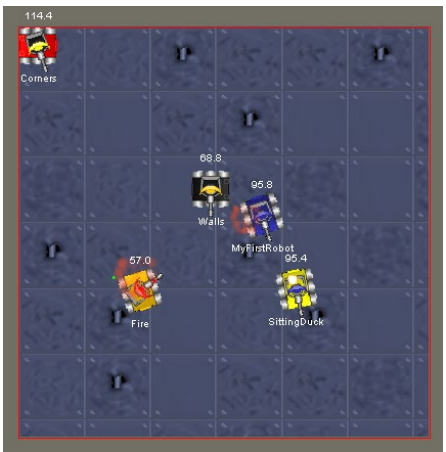
- Eine Robocode-Competition unter den WPF-Teilnehmern (+ Sample-Robots) organisieren
- Verständnis für Multi-Agenten-Spiele entwickeln und wie man sie auswertet

AUFGABEN:

- Kurzeinführung, die den 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: [Shichel05] und [Harper11] studieren und Grundideen in Vortrag erläutern
- Demonstration
- Lessons Learned

MATERIALIEN:

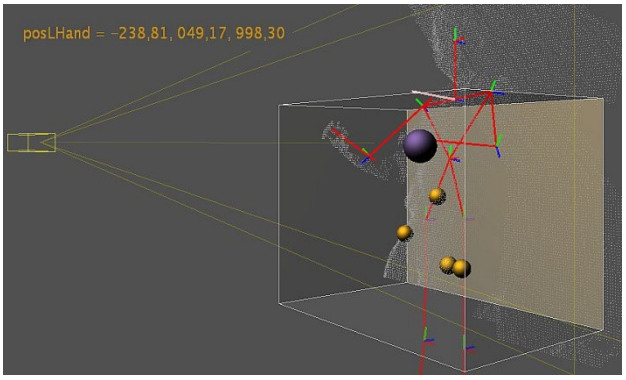
- [Nelson01][Larsen06] [Nielsen10] http://robocode.sourceforge.net/



- [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 (local copy)
- [Harper11] Harper, R: Co-Evolving Robocode Tanks. GECCO '11: Proceedings of the 13th annual conference on Genetic and evolutionary computation, 2011, 1443 (local copy)

Game Physics mit Kinect 1

VORBEHALT: Sie müssen prüfen, ob Kinect noch auf heutiger Computer-Hardware lauffähig gemacht werden kann (!)



Typ: Projekt
ZIELE:

- Einführung in die Kinect Sensoreinheit und SimpleOpenNI
- Skeleton-Programmierung mit SimpleOpenNI verständlich machen
- Möglichkeiten und Grenzen der 3D User-Interaktion erfahrbar machen

AUFGABEN:

- Durcharbeiten der Literatur, insbes. [Rheiner12], Teile aus [Bore12] und darin genannte Literatur
- Begriffe wie OpenNI, SimpleOpenNI, Synapse, PrimeSense usw. erklären
- Grundlagen zum Skeleton Model erklären
- Einfache Beispiele zur Kinect-Programmierung durchgehen und demonstrieren
- Umfassenderes Beispiel mit 3D-GamePhysics-Interaktion:
  - Innerhalb einer 3D-Box befinden sich einige Kugeln, die sich unter dem Einfluss der Schwerkraft bewegen
  - Der (oder die) User können mit den Kugeln interagieren: Bei Annäherung bildet sich eine Feder zwischen User-Hand und Kugel: Die Kugel hängt „am Haken“ und folgt der Hand nach.
  - Wird die Feder zu stark in die Länge gezogen, dann reißt sie wieder.
- Oder ein selbst-gewählter Ansatz von ähnlicher Komplexität
- verständliche Erläuterung des gewählten Lösungsansatzes, etwaiger alternativer Ansätze
- Wünschenswert: Übungen für alle Teilnehmer überlegen, die diese in der Lehreinheit auf Basis Ihres Codes einbauen.
- Demonstration
- Lessons Learned

KINECT-LITERATUR:

- [Bore12] Borenstein, G.: „Making Things See – 3D-Vision with Kinect, Processing, Arduino and MakerBot”, O’Reilly, 2012. Very good intro to Kinect + Processing (sometimes a bit lengthy, but accurate).
  http://makingthingssee.com/ → “See all code examples on Github”
- [Rheiner12] http://code.google.com/p/simple-openni/: **SimpleOpenNI** project page
- http://code.google.com/p/simple-openni/wiki/Installation hat wichtige Installation-Hinweise.

- Anmerkung: Nach der Installation der SimpleOpenNI Library in Processing findet man unter „File – Examples – Contributed Libraries“ zahlreiche instruktive Beispiele, die sich für einen Start in die Processing-Kinect-Programmierwelt eignen (!)
- SimpleOpenNI is a great tool. If you want to use it to recognize complex gestures have a look at http://code.google.com/p/kineticspace

*Game Physics mit Kinect 2

VORBEHALT: Sie müssen prüfen, ob Kinect noch auf heutiger Computer-Hardware lauffähig gemacht werden kann (!)

Typ: Projekt
ZIEL:

- Gestaltung eines einfachen Games mit Kinect Sensoreinheit
-

AUFGABEN:

- Man findet mit den Schlagworten „Games Kinect Processing“ zahlreiche Beispiele im Netz. Lassen Sie sich von einigen inspirieren, z.B. von diesen YouTube-Links
  - http://www.youtube.com/watch?v=h6xYoOTsAqQ
  - http://www.youtube.com/watch?v=rcrIzCyv50U
- und entwickeln Sie dann Ihr eigenes kleines Spiel auf Basis von Kinect, SimpleOpenNI und Processing.
- Es kann ein 2D- oder ein 3D-Game sein.
- Wenn möglich, Game Physics geeignet einbauen und verständlich erklären.
- Wünschenswert: Übungen für alle Teilnehmer überlegen, die diese in der Lehreinheit auf Basis Ihres Codes einbauen.
- Sind 2 Spieler möglich?
- verständliche Erläuterung des gewählten Lösungsansatzes, etwaiger alternativer Ansätze
- Demonstration
- Lessons Learned
- **WASP-geeignete Erweiterung**:
  - Wenn Kinect-Install lösbar, dann ausführliches Installations-Dokument (evtl. für verschiedene Plattformen, evtl. mit Vor- und Nachteilen)
  - Umfangreicheres Spielprojekt (3D) realisieren oder mehrere Spielprojekte und deren Vergleich. Es sollte auf jeden Fall Game Physics enthalten sein. Die dahinterliegende Dynamik soll anschaulich erklärt und untersucht werden (Stabilität, Parameter für gutes Gelingen, …).
-

MATERIALIEN:

- KINECT-LITERATUR (s.o.)
- Links zu den Schlagworten (s.o.)

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!

Wieso braucht man bei Game Physics gute Integrationsverfahren?

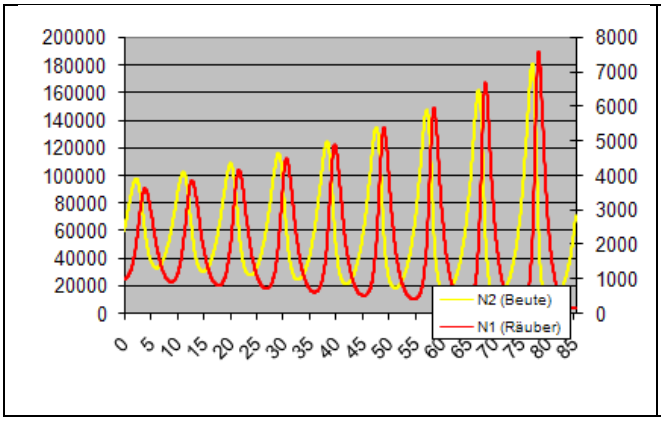
Typ: Referat
AUFGABENSTELLUNG:

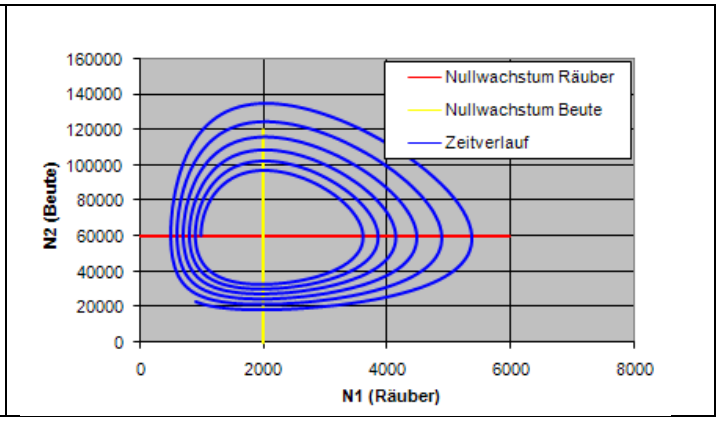
- Motivation: An 2D-Softball o.ä. kann man sehen, wie schnell das Euler-Verfahren zum völligen "Kollabieren" führen kann.
- Problematik erläutern. Zeigen, was passiert. Verbesserungsverfahren vorstellen: Euler, Heun, Runge-Kutta >> vgl. Untersuchungen an Processing-Beispiel
- Prinzip der steifen DGL an einfachem Beispiel (Tabellenkalkulation oder Processing)
- Erläuterung des gewählten Lösungsansatzes, etwaiger alternativer Ansätze
- Planetenbewegung oder Federpendel: Vgl. semi-expliziter Euler mit „normalem“ Euler
- Demonstration
- Lessons Learned

MATERIALIEN:

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

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

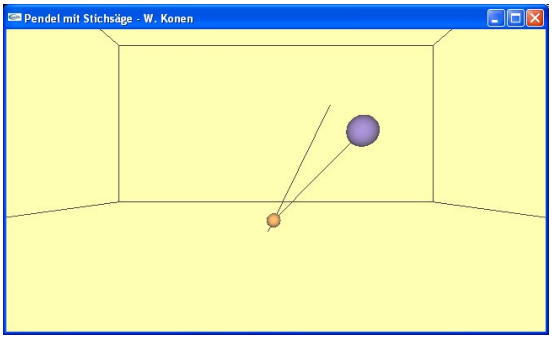
1. Vortragsteil
2. Demonstrationsteil (Simulationen vorführen)
3. Übungsteil: Konzeption (sinnvoller und machbarer!) Übungen für die anderen Teilnehmer (z.B. gezielt einzelne Elemente im Excel-Blatt weglassen, gemeinsam mit Teilnehmern lösen)

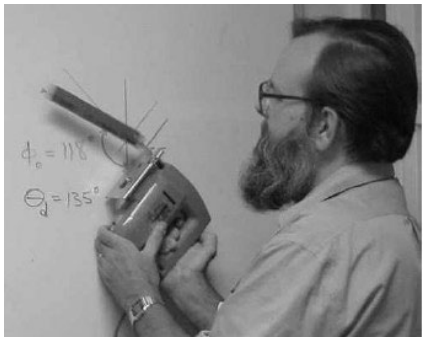
- Es sollten zumindest einige der im Handout genannten Übungen 1.-8. bearbeitet werden (!), entweder durch die Vortragenden oder als Übung für alle Teilnehmer
- Es sollten über das Handout hinausgehende interessante Informationen aus der Literatur eingearbeitet werden.

MATERIALIEN:

- Handout Struggle.html und darin genannte Literatur

Invertiertes Pendel mit Stichsäge





Typ: Projekt

ZIEL:

- In [Pöppe03] steht, wieso ein aufrecht stehendes Pendel allein durch eine Stichsäge – 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
- Processing-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 Mausklick 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 Stichsäge schräg hält? >> Simulation erweitern
- Übung für die Teilnehmer: bei schräger Stichsäge, Neigungswinkel β, durch Ausprobieren herausfinden, bei welcher Anfangsneigung $\varphi(\beta)$ das Pendel am schnellsten zur Ruhe kommt. Kann man $\varphi(\beta)$ berechnen?

- Sei $\beta=11^0$, $g=9ms^{-1}$, $L=14m$, $\omega=72s^{-1}$. Bei welcher Amplitude (Hub der Stichsäge) bricht das invertierte Pendel zusammen?
- Könnte man durch Regelung der Amplitude oder Frequenz der Stichsäge 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 Stichsäge 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.

Kommentiertes Literaturverzeichnis

Lösungen.docx