

WPF Spiele, Simulation und dynamische Systeme

Lehrveranstaltung von Prof. Dr. Wolfgang Konen

SS 2017, Beginn 02.05.2017

Themenvorstellung im Detail

Zeit: Di, 14:00 – ca. 17:00 Uhr

Raum: 3.113

Gliederung / Termine (Änderungen vorbehalten!)

Teil 1: Simulation

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

(Terminänderungen vorbehalten)		Wieviele / Wer
02.05.17	Startup-Veranstaltung, Projekte verteilen	Kg
V / Ü	Einführung in die Simulation	Kg
09.05.17	Simulation mit Tabellenkalkulation (mit Übungen)	Kg
V / Ü	Von der Zinseszinsrechnung zur Populationsdynamik. Diskretisierung: Von der DGL zur numerischen Lösung (m. Übungen)	Kg
16.05.17	Analytische Methoden: Linearisierung im Gleichgewichtspunkt	Kg
	dann eigenständige Projektarbeit	
Ergänzende Kapitel:		
V	Numerische Probleme bei der Simulation (Praxisbericht)	
P/S	Simulation von Räuber-Beute-Systemen	
P/S	Gute Integrationsverfahren für Game-Physics	

Teil 2: Game Physics

(Terminänderungen vorbehalten)		Wieviele / Wer
V / Ü	Einführung in Processing	Kg
23.05.17	Tutorial 3D-Simulation in Processing	
V	Einführung Game Physics	Kg
30.05.17	Simulation für Spiele, Ü balls2D + Federkraft	Kg
		Kg
xx.06.17	- kein Vortrag, Referat- und Projektarbeit -	

13.06.17	(evtl.) Milestone-Treffen zu den Projekten	
S/P 27.06.17	Robocode , Teil 1	Jan Schröder, Danny Kürten [Prtk: Bilinsky, Helmedag]
S/P 27.06.17	Kollisionsdetektion und Kollisionsantwort	Jens Bilinsky, Florian Helmedag [Prtk: Suckow, Maclachlan]
S/P 04.07.17	Wieso zeigt der Mond der Erde immer dasselbe "Gesicht"?	Maximilian Suckow, Alasdair Maclachlan [Prtk: Dahmen Friederichs]
S/P 04.07.17	Fahne/Vorhang simulieren	Felix Dahmen, Fabian Friederichs [Prtk: Schröder, Kürten]
S/P 11.07.17	Robocode , Teil 2	Jan Schröder, Danny Kürten [Prtk: Bilinsky, Helmedag]
11.07.17	Abschlussdiskussion und Feedback	alle
S/P 18.07.17	WK Konferenzreise	
Ergänzende Kapitel:		
P/S 16.01.15	„Smoother“ Rotation mit Quaternionen	NN, NN [Prtk: .., ..]
P/S 16.01.15	3D-Softball-Simulation	NN, NN [Prtk: .., ..]
S/P 23.01.15	GVGP: General Video Game Playing	NN, NN [Prtk: .., ..]
S/P 23.01.15	Game Physics mit Kinect 1	NN, NN [Prtk: .., ..]
S/P 23.01.15	Game Physics mit Kinect 2	NN, NN [Prtk: .., ..]
S/P 16.01.14	Invertiertes Pendel mit Stichsäge	N.N. [Prtk: N.N.]
(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. 1,5stündigen 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. 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 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 SS17** beitreten) >> Mailingliste
 2. Bereitstellen der Ausarbeitung, **die zum Vortrag fertig sein muss (!)**: (i) in ILIAS durch Upload unter Gruppe WS14 **UND** (ii) in **Papierform** mitgebracht wird (einmal für mich).

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

Einführung in Processing

Typ: Vortrag Dozent / Praktische Ü am Rechner

ZIEL/TOPICS:

- [Einführung 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](#) und [Meqon Demo](#)
- Abgrenzung, Konzentration auf Physics Engines
- Animation vs. Simulation
- Einfaches GamePhysics-Beispiel: Dynamik in balls.exe ($\frac{1}{2}gt^2$, 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!)

AUFGABEN:

- 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 ILIAS zu WPF, *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

„Smoother“ Rotation mit Quaternionen

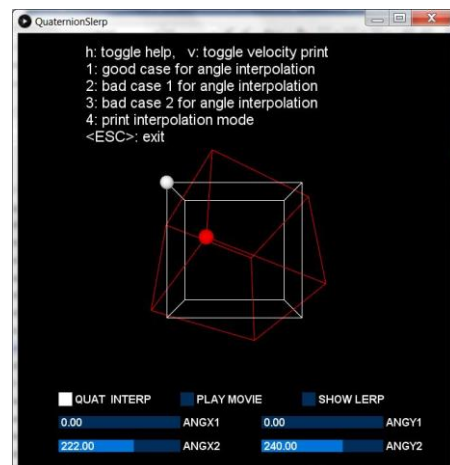
Typ: Projekt

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 „smoother“ 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
- Übungen / Weiterführende Fragen:



- SLERP mit anderen Interpolationen vergleichen (LERP, NLERP)
- Wie ist die Geschwindigkeit einzelner Objektpunkte bei SLERP?
- optional: erweitern auf andere 3D-Objekte
- 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". Last update: Feb'2004. Last access: Feb'2017 ([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](#))

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
- Übungen / 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“
 - optional: [BillardGL](#) als Beispiel für ein Spiel mit Kollisionen vorstellen

MATERIALIEN:

- [balls2D.pde.zip](#) (Processing-Variante WK, ohne Kollision)
- Tutorium unter <http://nehe.gamedev.net/>, Lesson 30.
- [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.
- [Nopper05] T. Nopper et al.: *BillardGL*, <http://www.billardgl.de/>, 2005.

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/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:

- [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 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
- 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 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
- 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 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 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] <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



GVGP: General Video Game Playing – eine Herausforderung für die AI



Typ: Projekt

THEMA:

- Ein NPC in einem Video Game muss entsprechend der Spieltaktik „intelligent“ agieren, um gegen die anderen Spieler bestehen zu können. Das hört sich herausfordernd an. Wie kann ein Computer intelligente Taktiken erlernen? GVGP

geht sogar noch einen Schritt weiter: Eine ganze Klasse (vorher unbekannter) Video-Games (Arcade-Games) soll gemeistert werden. Hierzu wurde auf der Konferenz [CIG'2014](#) eine Competition durchgeführt. Das Java-Framework dazu steht zur Verfügung.

ZIEL:

- GVGP-Competition erläutern, verstehen, eigene Agenten entwickeln
- Ansätze, wie man Agenten entwickeln kann, die in (verschiedenen) Video-Games „intelligent“ agieren.

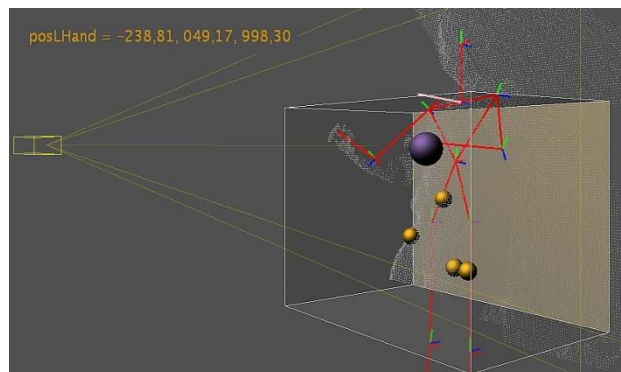
AUFGABEN:

- Einarbeiten in GVGP-Competition-Framework. Verstehen der Beispiel-Agenten.
- Vortrag zu GVGP und „Drumherum“ (insbesondere **VGDL, Video Game Description Language**)
- Optional: Grundlagen **MCTS (Monte Carlo Tree Search)** erläutern
- Eigene GVGP-Controller (Agenten) bauen und ihre Erfolge / Misserfolge analysieren. (Evtl. sich erst einmal nur auf ein Spiel beschränken und ohne Zeitconstraints)
- Überlegen, wie man eine Competition evaluieren und analysieren kann
- Aufgabe für die WPF-Teilnehmer überlegen
- Demonstration
- Lessons Learned

MATERIALIEN:

- <http://gvgai.net>: The General Video Game AI Competition, 2014. Complete Java Framework, explanations, literature.
- [Perez14] Diego Perez, Spyridon Samothrakis and Simon Lucas: **Knowledge-based Fast Evolutionary MCTS for General Video Game Playing**, *IEEE Conference on Computational Intelligence in Games (CIG-2014, Germany)*, p. 68-75. [\[PDF\]](#)
- [Schaul13] Tom Schaul. **A Video Game Description Language for Model-based or Interactive Learning**. *IEEE Conference on Computational Intelligence in Games (CIG-2014, Niagara Falls, Canada)* [\[PDF\]](#)
- [Browne13] Cameron Browne, “Monte Carlo Tree Search”, talk, 2012 http://ccg.doc.gold.ac.uk/teaching/ludic_computing/ludic16.pdf, local copy [here](#) →: good introduction MCTS, including pseudo code and UCT, good Pros and Con

Game Physics mit Kinect 1



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 Lehrsicherheit 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



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=rclrzCyy50U>

- 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

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!

[Kommentiertes Literaturverzeichnis](#)

Ergänzende Themen

Wieso braucht man bei Game Physics gute Integrationsverfahren?

Typ: Referat

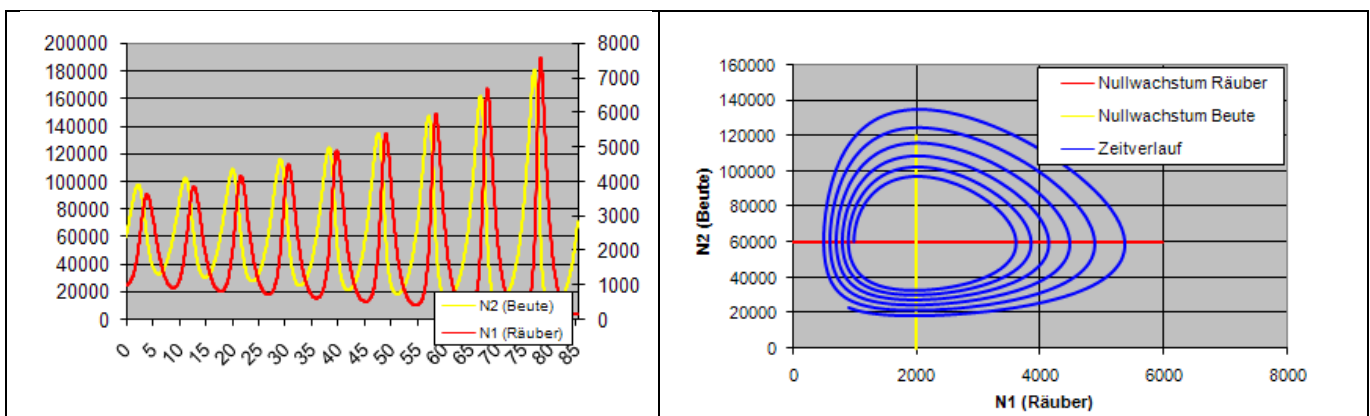
AUFGABENSTELLUNG:

- 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
- 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

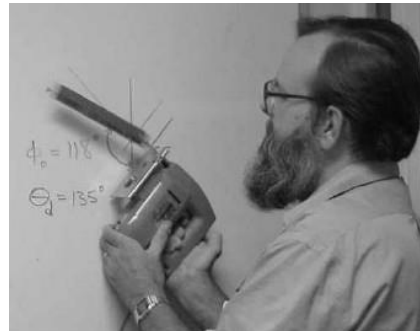
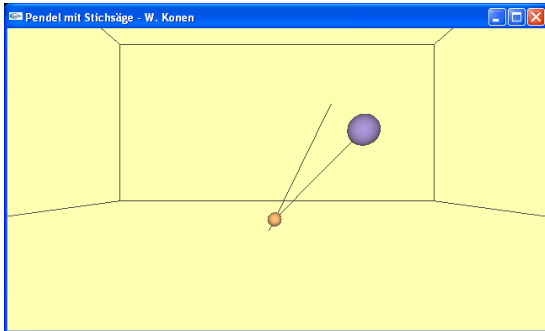
AUFGABEN:

- 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 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 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 $\beta=11^\circ$, $g=9\text{ms}^{-1}$, $L=14\text{m}$, $\omega=72\text{s}^{-1}$. 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.

Fighting Game AI Competition

Typ: Projekt

THEMA:

- Wie bei GVGP, dieses Mal jedoch für ein Fighting Game AI Competition (FGAIC). Auch diese Competition wurde auf der [CIG'2014](http://www.cig2014.com) vorgestellt. Im Gegensatz zu GVGP ist FGAIC jedoch ein fortlaufender Wettbewerb, bei dem auch in 2015 wieder Competitions durchgeführt werden. Dieses Projekt kann als Vorbereitung dafür dienen. Das Java-Framework dazu steht zur Verfügung.

ZIEL:

- FGAI-Competition erläutern, verstehen, wenn möglich eigene Agenten entwickeln

AUFGABEN:

- Einarbeiten in FGAI -Competition-Framework. Verstehen der Beispiel-Agenten.
- Vortrag zu FGAI und „Drumherum“
- Eigene FGAI -Controller (Agenten) bauen und ihre Erfolge / Misserfolge analysieren. (Evtl. ohne Zeitconstraints)
- Überlegen, wie man eine Competition evaluieren und analysieren kann
- Aufgabe für die WPF-Teilnehmer überlegen
- Demonstration
- Lessons Learned

MATERIALIEN:

- <http://www.ice.ci.ritsumei.ac.jp/~ftgaic>: The Fighting Game AI Competition, 2014. Complete Java Framework, explanations, literature.
- weitere Publikationen, die auf der Seite „Links“ von <http://www.ice.ci.ritsumei.ac.jp/~ftgaic> zur Verfügung stehen.



[Lösungen.docx](#)