Human Centered Multimedia
Institute of Computer Science
Universität Augsburg University

Einführung in die Spieleprogrammierung

Überblick und Organisatorisches

Kontakt

Vorlseung/Übung: Simon Flutura

http://www.informatik.uni-augsburg.de/lehrstuehle/hcm/staff/flutura/

Tobias Huber

http://www.informatik.uni-augsburg.de/lehrstuehle/hcm/staff/huber/

Übung: Florian Lingenfelser

http://www.informatik.uni-augsburg.de/lehrstuehle/hcm/staff/lingenfelser/

Überblick

- Inhalte der Vorlesung
- Übungsablauf
- Notengebung
- GIT, Wiki

Vorlesung

- Vorlesung findet montags um 15:45 Uhr statt
- Fokus auf Forschung
- Schwerpunkt dreht sich um **KI** und Eingabemethoden

Inhalte der Vorlesung

- GameDesign und Storytelling
  - Story Vs. Interaction
  - Simluation Vs. GamePlay
  - Dramaturgie und Erzähltechniken
  - Spieler und Entwickler
- Animation
  - mit Joints, Meshes und Morphtargets
  - Traditionelle Animationsprinzipien
  - Keyframing und Interpolation
  - Von Motion Caputering bis physikbasierten Animationen
  - Inverse und direkte Kinematik
  - Kodierungssysteme und Markupsprachen

Inhalte der Vorlesung

- KI
  - Verhaltenssteuerung
  - Planung und Strategien
  - Wegsuche und Pfadplanung
  - Navigation
  - Crowd Simulation
  - Kognitive Architekturen
  - Emotion und Persönlichkeit
  - Storytelling und Dialoge
  - KI Middleware

Inhalte der Vorlesung

- Benutzerschnittstellen
  - Ein-/Ausgabegeräte und Software
  - Benutzeroberflächen
  - Richtlinien und Kriterien zur Usability
  - Tastatur/Maus und Gamepad/Joystick/Lenkrad
  - Spracheeingabe
  - Soziale Signalverarbeitung SSI, Move, Wiimote
  - Datenhandschuh, Tanzmatte, Computer Vision mit Kameras, Eye Tracker, …

Inhalte der Vorlesung

- Grafik
  - Evolution von 2D zu 3D
  - Stereoskopisches 3D und VR/AR
  - DirectX, OpenGL und mehr
  - Grafik-Pipeline von Modelldaten mit Transformation, Beleuchtung, … über Projektion und Rasterung zum Bildschirm
  - Perspektiven, Sichtbarkeit und Level of Detail
  - Shaderprogrammierung und andere Effekte
- Netzwerkkommunikation
  - Server-Client Architekturen
  - Sockets

Inhalte der Vorlesung

- Physik
  - Physik SDKs und Hardware
  - Rigid und Soft Bodies
  - Kinematik und Gravitation
  - Kollisionserkennung und –auswirkungen
  - Constraints und deren Auflösung
  - Wasser-/Feuer-/Rauch-…-Simulation über Partikeleffekte
- Sound
  - Musik
  - Soundeffekte
  - Text to Speech
  - Mono/Stereo/Raumklang und 3D-Sound
  - Hardware und Software

Ablauf

- In jeder Vorlesung: Präsentation eines Spiels, das euch beeindruckt hat und von dem ihr glaubt, dass die Wenigsten im Kurs es kennen
- Übungsaufgaben (paarweise, 1-3 wöchig)

Spielevorstellung

- Jeder muss am Anfang einer Vorlesung einen Spiel vorstellen (10 min)
  - Warum/Wie hat er euch beeindruckt?
  - Was ist das Besondere/Innovative daran?
  - Was ist neu?
  - Screenshots oder Videos
- Folien spätestens um 10:00 Uhr am Tag eures Vortrages an uns per Mail schicken (falls Feedback erwünscht mindestens einen Tag früher)

Voraussetzungen

- Rechner mit OpenGL 2.0-fähigen Grafikkarte
  - alle halbwegs aktuellen Grafikkarten, aber auch Onboard-Grafikkarten reichen oft aus
- Kenntnisse in C# zwingend nötig (kein Programmierkurs)
- Eigenständiges Bearbeiten der Übungsaufgaben

Übung

- Ausgabe der Übungsblätter mittwochs
- Abgabe (1-3 wöchig) Dienstag Abend 23:59 Uhr
- Aufgabenblatt wird in 2er Gruppen bearbeitet
- Jedes Aufgabenblatt wird bewertet
- Ausgewählte Abgaben werden in den Übungen vorgestellt
- Abgabe in GIT

Notengebung

- Übungsaufgaben (60%)
  - Vollständige Funktionalität
  - Fehlerfreiheit
  - Umsetzung
  - Programm-Struktur und Dokumentation (Code und Wiki)
  - Kreativität und gute Ideen werden besonders belohnt!
- Spielepräsentation (10%)
  - Qualität und Verständlichkeit
  - Zeitmanagement
- Einzelabnahme am Semesterende mit kurzer Befragung zu den eigenen Lösungen (30%)
  - Mehr Kontrolle der bereits erbrachten Leistung als Wissensprüfung

Git, Forum und Wiki

- Jede Gruppe bekommt einen eigenen GIT Zugang
- Abgabe der Übungsaufgaben über das GIT
- Zugangsdaten bekommt jeder per Mail
- Fragen und Antworten bitte nur über das Digicampus-Forum oder in den Übungen

Git, Forum und Wiki

- Die Zugangsdaten für das GIT gelten auch für das Wiki
- Dort Dokumentation der eigenen Übungsaufgaben, z.B.:
  - Systemstruktur
  - Was macht mein Programm?
  - Wie kann ich es starten und bedienen?
  - Welche Design-Entscheidungen wurden getroffen und wieso?
  - Was ist besonders an meiner Umsetzung?
  - …

Git Regeln

- Im GIT bitte nur das einchecken was tatsächlich zum kompilieren benötigt wird!
- Also keine temporären oder rechnerbezogenen Dateien, und insbesondere Binärdateien vermeiden, z.B. .ncb, .sdf, .suo
- Diese Dateien sollten in TortoiseGIT am besten auf die „ignore“-Liste gesetzt werden:
  - Dazu einfach die Datei mit der rechten Maustaste anklicken. Im sich öffnenden Kontextmenu dann „TortoiseGIT/Add to ignore list“ auswählen. Dabei kann man entscheiden, ob nur die eine Datei oder alle Dateien mit dieser Dateiendung ignoriert werden sollen.
  - Gleiches funktioniert auch mit kompletten Ordnern.
  - In der GameEngine sollten grundsätzlich die Ordner „Build“, „include“, „ipch“ und „lib“ auf die „ignore“-Liste gesetzt werden. Außerdem praktisch alle Dateien im „bin“-Verzeichnis.

Termine

- Vorlesung:
  - Montag, 15:45 – 17:15 Uhr (Raum 1054 N)
- Übung:
  - Donnerstag, 9:00 – 12:30 (Raum 2026 N)
  - 1. Übung am 02.05

Gruppeneinteilung