

Skript zur Vorlesung
Managing and Mining Multiplayer Online Games
im Sommersemester 2012

Kapitel 1: Computerspiele

Skript © 2012 Matthias Schubert

http://www.dbs.informatik.uni-muenchen.de/cms/VO_Managing_Massive_Multiplayer_Online_Games

Organisatorisches

- **Aktuelles**

- Vorlesung: Mittwoch, 13.00 - 16.00 Uhr Raum 151 (Oettingenstr. 67)
- Übung: Do, 14.00 - 16.00 Uhr Raum U127 (Oettingenstr. 67)

- **Anmeldung für die Klausur auf der Homepage unter**

http://www.dbs.informatik.uni-muenchen.de/cms/VO_Managing_Massive_Multiplayer_Online_Games

- **Klausur: Der Stoff der Klausur wird in der Vorlesung
und in den Übungen besprochen.
(Das Skript ist lediglich eine Lernhilfe)**

Was sind Computerspiele ?

Computerspiel:

„An interactive experience that provides the players with an increasingly challenging sequence of patterns which he or she learns and eventually masters.“[Ralph Koster: A Theory of Fun for Game Design, Phoenix, AZ, Paraglyph 2004]

⇒ Interactive: Interaktion mit dem Computer oder Mitspieler

⇒ Experience:

Wahrnehmen und Erleben des Inhalts spielen eine wichtige Rolle

⇒ Challenge, Learning and Mastering:

Ein gutes Spiel ist herausfordernd, aber erlaubt eine Verbesserung bis hin zur Meisterschaft .

3

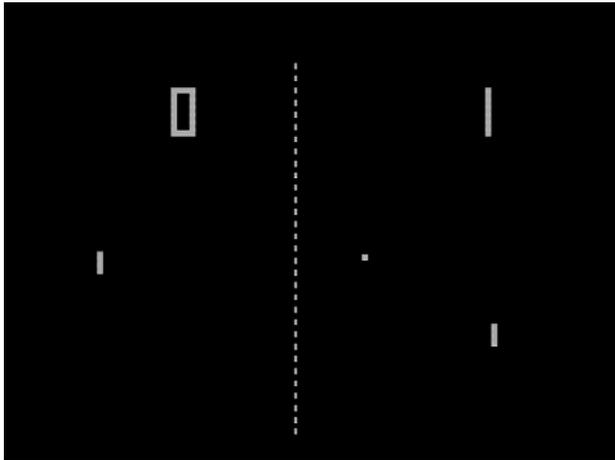
Verwandte Arten von Software

- *Real-Time Simulationen*: Modellierung und Darstellung einer realen physikalischen Umgebung und ihrer dynamischen Entwicklung. Verkehrssimulationen oder Simulation von Schwarmverhalten
- *Geo-Informationssysteme*: Verwaltung von räumlichen Informationen (2D/3D Karten) und der sich darin bewegenden Objekte z.B. Autos.
- *Agentensysteme*: Systeme der Zustand über eine Menge von unabhängig agierenden Einzelentitäten abhängt.
- *Virtuelle Realität*: Rendering von Sound und 3D Umgebungen

4

Die obligatorische Pong Folie

erstes kommerziell erfolgreiches Computerspiel: PONG (Atari 1972)



- Pong ist ein einfaches Tennisspiel(Geschicklichkeitsspiel)
- es wurden eigene Kontroller dafür gebaut
- wurde in den folgenden Jahren oft kopiert und modifiziert

5

Soziale Aspekte von Computerspielen

- 68% der Amerikanischen Haushalte konsumieren Videospiele
- Der Durchschnittsspieler ist 35 und spielt seit 12 Jahren.
- Der Durchschnittskäufer ist 39 Jahre alt.
- 40% der Spieler sind Frauen. Frauen, die älter als 18 sind haben mit 34% einen größeren Anteil als Jungen unter 18. (18%)
- In 2009, spielten 25% der Amerikaner über 50.
- Eine Verbindung zwischen Videospiele und Gewaltbereitschaft ist bis heute nicht wissenschaftlich belegt.
- Es gibt mehrere Untersuchungen, die ein negative Korrelation zwischen der Verbreitung von Computerspielen und der Anzahl von Gewaltverbrechen belegen.

Quelle : <https://depts.washington.edu/critgame/wordpress/2010/04/fyi-video-game-statistics-by-the-entertainment-software-association/>

6

Einnahmequellen und Geschäftsmodelle

Verkauf

- einmalige Preis, den der Kunde zahlen muss die Software zu erheben.
- Verkauf über Geschäfte (Boxed Games) oder auch online
- für Offline-Spiele geeignet
- hohes Einstiegsrisiko für Spieler / bei nicht Gefallen wird Kaufpreis nicht rückerstattet
- Gefahr von Raubkopien
- Marketing und Werbung für Erfolg wichtig

Beispiele: Warcraft, Call of Duty, Resident Evil, Gran Turismo, Grand Theft Auto, Civilization,...



7

Einnahmequellen und Geschäftsmodelle

Abonnement

- Kunde muss regelmäßig Geld für Account zahlen
- nur für Online-Spiele geeignet
- Account-Verwaltung ist notwendig
- neue Kriminalitätsformen entstehen(z.B. Account-Kidnapping)
- garantiert dem Betreiber regelmäßige Einkünfte
- Content Updates und Kundenzufriedenheit sind sehr wichtig

Beispiele: Ultima online, World of Warcraft, Everquest, Star Wars: The old Republic...



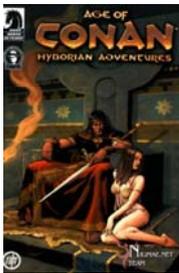
8

Einnahmequellen und Geschäftsmodelle

Micro-Transactions

- Verkauf virtueller Zahlungsmittel, Gegenstände oder Fähigkeiten gegen reales Geld
- geeignet für free-to-play Spiele
- Entwicklungsrisiko am höchsten aber sehr hohe Gewinnspanne
- geringste Einstiegsschwelle für Kunden, geringes Einstiegsrisiko (Spieler investiert erst bei Gefallen Geld)
- birgt die meisten Gefahren durch Betrug, da Spielwirtschaft mit realer Wirtschaft verbunden sein muss

Beispiele: Age of Conan, Herr der Ringe online, Farmville, Travianer, ..



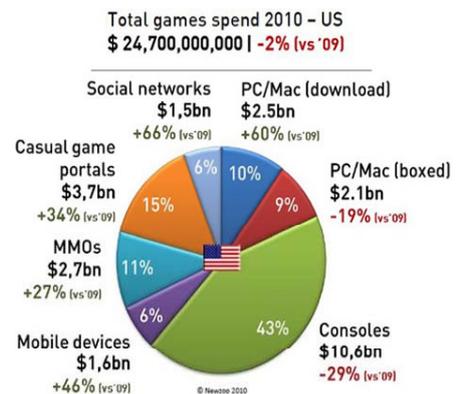
9

Marktentwicklung der letzten Jahre

- Markt für Computerspiele ist einer der größten in der Unterhaltungsbranche (ca. 74 Mrd US-Dollar weltweit)

Zum Vergleich der Markt für Datenbanksysteme hat ungefähr eine Volumen von ca. 20 Mrd US-Dollar

- Boxed Games haben 29% des Marktanteils in 2010 verloren
- MMO, Casual-Game-Portale and Social Network Games hatten einen Zuwachs von 27-66 % in 2010



<http://www.gamesindustry.com/home>

Erfolgsfaktoren

- ansprechende Erfahrungswelt, herausfordernd aber stetige Verbesserung möglich (gutes Spieldesign)
 - gute Servicequalität: Verfügbarkeit, Kundendienst,..
 - ausreichende Spieleranzahl: je mehr Spieler ein Spiel regelmäßig spielen, desto interessanter wird es.
 - Spieldauer: Je mehr Zeit Spieler in ein Spiel investieren desto erfolgreicher ist ein Titel:
 - Soziales Prestige des Spiels steigt bis zu einem gewissen Grad
 - Mehr Verkäufer, längere Abonnement-Dauer, mehr Transaktionen
- Aber es gibt auch negative Folgen des Erfolgs:
- Suchtgefahr: Prestige fällt gewaltig
 - große Community => Lösungen werden schnell publik (
 - Systematischer Betrug konzentriert sich auf erfolgreiche Titel

=> *Langfristige Kontrolle des Spiel-Design und des Spielerverhaltens sind wichtige Aspekte für Spielbetreiber.*

11

Unterscheidungsmerkmale von Computerspielen

- *Anzahl Gesteuerte Spiele-Entitäten:* Einheit, Gruppen, Fraktionen
- *Darstellungsperspektive:* First-Person, Third Person , Vogelperspektive
- *Zeitlicher Ablauf:* Real-Time, Game-Time, rundenbasiert
- *Komplexität der Steuerung:* Simulation vs. Fun-Control, Manual Aim vs. Auto-Target, ...
- *Dimension/Granularität der Spielewelt:* 2D, 3D, fixe Spielfelder vs. freie Bewegung
- *Anzahl der Spieler:* Single-Player, Multi-Player (1-100), Massive-Multiplayer (100-100.000)
- *Entwicklung des Spieleraccounts:* Virtuelle Fähigkeiten, Spieler-Fähigkeiten (RPG vs. Schach)
- *Einfluss des Zufalls:* Würfelspiel vs. Schach

12

Klassische Spiele-Genres

Real-Time Strategie Games (RTS)

- Spiele die Aufbau einer Armee und Sieg über einen Gegner zum Ziel haben.
- Steuerung sehr vieler Einheiten, Vogelperspektive, geringe Granularität der Steuerung, Echtzeit, 2D/3D Welt, Spieleranzahl 1-10
- Beispiele: Dune II, Starcraft, Warcraft, Comman&Conquer, Age of Empires ..

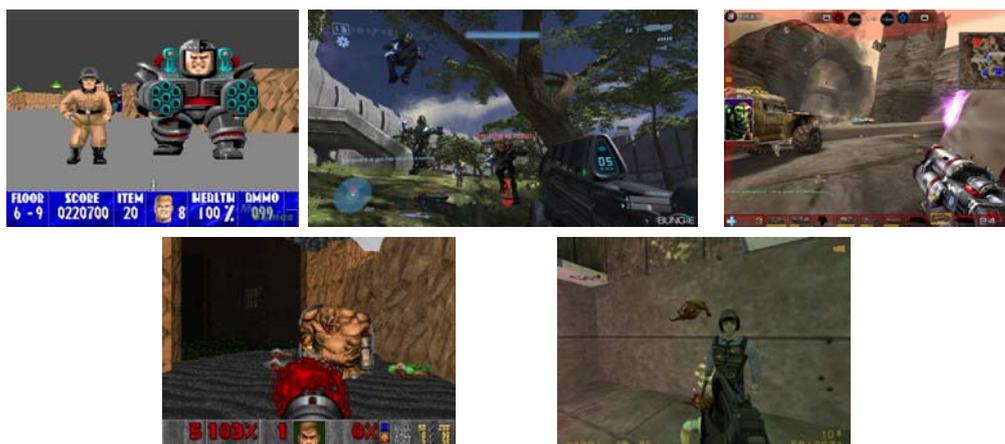


13

Klassische Spiele-Genres

First-Person-Shooter (FPS)

- Simulation von Feuergefechten
- Eine Einheit, Ego-Perspektive, Echtzeit, sehr hohe Steuerungsgranularität, 3D Welt, meist 1-20 Spieler
- *Beispiele:* Wolfenstein 3D, Doom, Unreal, Half-Life, Team Fortress, Halo, Counter Strike,..



14

Klassische Spiele Genres

Massive Multi Player Role Playing Games (MMORPGs)

- Entwicklung und Geschichte eines Spielercharaktärs
- 1 Einheit, Third-Person View, mittlere Granularität der Steuerung, Echtzeit, 3D Welt, Spieler Anzahl sehr hoch (mehrer tausend), Entwicklung des Charaktärs ist zentral,
- Beispiele: Ultima Online, Everquest, World of Warcraft, Star Wars: The old Republic, Age of Conan, Herr der Ringe Online...



15

Klassische Spiele-Genres

Rennspiele

- Steuerung von Fahrzeugen.
- Eine Einheit, Ego und Third Person Perspektive, Echtzeit, sehr hohe bis mittlere Steuerungsgranularität, 2D/ 3D Welt, meist 1-2 Spieler aber auch Multi-Player möglich
- *Beispiele:* Atari's Space Race (1970), Test Drive, Need for Speed, Outrun, Gran Turismo,



16

Klassische Spiele-Genres

Fighting Games

- Spiele die Hand zu Hand Kampf mit Gegnern zu Inhalt haben
- 1 Einheit, Third-Person View, mittlere Granularität der Steuerung, Echtzeit, 2D/3D Welt, multi-player
- Beispiele: Heavyweight Champ(1973), Karate Champ (1984), Street Fighter, Tekken, Mortal Combat



17

Klassische Spiele-Genres

Wirtschaftssimulationen und Rundenbasierte Strategy-Spiele

- Spiele die Aufbau einer Stadt, einer Nation, einer Firma oder eines anderen Verbundes zum Ziel haben
- Übergang zum Strategiespiel bei Konflikten mit Gegnern
- Viele Einheiten, Vogelperspektive, Echtzeit oder rundenbasiert, niedrige Steuerungsgranularität, 2D Welt, 1 bis tausende Spieler
- *Beispiele:* Intopia(1963), M.U.L.E(1983), Civilization, Die Siedler, Sim City, Travianer,...



18

Weitere Genres

- ***Jump and Run***: Steuerung eines Third-Person Charaktärs durch eine 2D/3D Landschaft. (z.B. Super Mario, Jumpman,..)
- ***Sing-, Musik-, Rhythmus-Spiele, Tanz-Spiele***: Zu einem Musikstück passende Eingaben werden mittels des Computers bewertet. (z.B: Sing Star, Rock-Band,..)
- ...



Es gibt noch zahllose weitere Spiele, die nicht unmittelbar in ein Kategorie fallen sondern,

- Kombinationen von bereits genannten Genres darstellen
- vollkommen andere Ideen in ein Spielerlebnis umwandeln

19

Aufbau von Computerspielen

- Moderne Computerspiele können vielschichtige komplexe Softwaresysteme sein, für die große Teams notwendig sind
- je nach Aufbau und Eigenschaften unterscheiden sich Computerspiele in ihrer Komplexität und ihrem Umfang.
- Gerade durch die neuen Plattformen (Android, iPhone/iPad, Browser und Soziale Netzwerke) erfahren einfache Spielprinzipien gerade eine Renaissance
- Selbst einfache Spielprinzipien erfordern im MMO Umfeld einen erheblich höheren Aufwand durch Skalierungsproblematiken und der Notwendigkeit das Spielerverhalten zu analysieren.

20

Aufbau eine Entwicklerteams

- **Software Engineers:** Entwicklung der Game Engine/Game Core und der Tools für Artists and Game Designer
- **Artists:** Zuständig für Content: Konzepte, 3D Modelle, Sound, Animation, Texturen, Lichteffekte, Animation und Motion Capture, Schauspieler...
- **Game Designers:** Zuständig für das Game Play (Interaktion) im der Gaming Experience: Story, Level Design, high Level Ziele, Verteilung der Gegner, Equipment, Spielerfähigkeiten,..
- **Produzenten:** Senior Game Designer, Projektmanager
- **weitere Mitarbeiter:** Marketing, Assistenten, Technisches Personal,...
- **Publishers und Studios:** Marketing, Herstellung, Distribution des Spiels

21

Aufbau von Computerspielen

Faktoren die gegen eine Standardarchitektur sprechen:

- Content steht im Vordergrund
- Unterschiedliche Anforderungen je nach Spiele
- hohe Anforderungen an Performanz
(je weniger Ressourcen je mehr potentielle Spieler)
- Hardware-Ressourcen verändern sich oft sehr schnell

Folge: Architektur ist häufig integriert und spezifisch als modular und wiederverwendbar.

Aber: Spiele sind komplexe Softwaresysteme

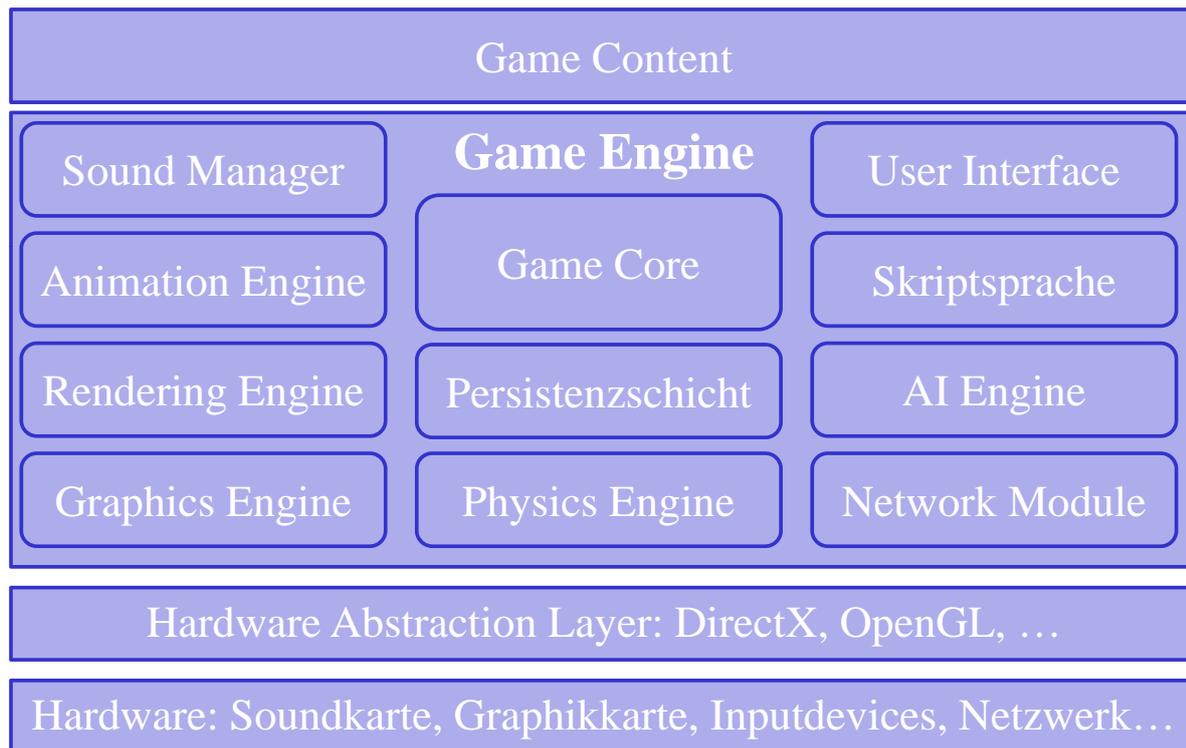
⇒ Verwendung von Standardkomponenten (*Engines*)

⇒ Geschichtete Architekturen und Modularer Aufbau

⇒ Wiederverwendung des Code für weitere Spiele (*Mods*)

22

Bausteine in der Architektur eines Spiels



23

Hardware Layer und Hardware Abstraction Layer

- **Komponenten der Hardwareplattform**
 - Graphikkarte
 - Soundkarte
 - Eingabegeräte (Keyboard, Maus, Joysticks, Game Pads, Lenkräder, Mikrophone, Kamera,...)
 - Sekundärspeicher
 - Hauptspeicher
 - Processor
- **Device Driver:**
 - Treibersoftware die Hardware vom Betriebssystem aus ansprechbar machen.
 - Heutzutage setzen die meisten Spiele auf einem Betriebssystem auf

24

Hardware Abstraction Layer

- Kapselung der Hardwareeigenschaften bei unterschiedlicher Hardwarekonfiguration (PC, Android)
- Bietet eine einheitliche Schnittstelle zur Hardware
- Angebotene Funktionalitäten erlauben Basisbefehle
- Beispiele:
 - Glide: 3D Graphik SDK, (überholt)
 - DirectX: Microsoft's 3D Graphik SDK
 - OpenGL: weit verbreitetes SDK, implementiert auf vielen Plattformen
 - libgcm+Edge: Graphic Interface der PlayStation 3

25

Graphik Engine

- höhere Zugriffsebene auf Graphikfunktionalitäten
- meist auf eine spezifischen Art der Darstellung zugeschnitten
 - Sprites
 - Isometrisch
 - 3D
- Arbeit mit Modellen auf einer höheren Abstraktionsstufe
 - Sprites
 - Charaktere
 - Solids
- Realisiert kompliziertere Aspekte der Anzeige
 - Mini Maps
 - multiple Sichten
 - Overlays
 - Spezialeffekte

26

Rendering Engine

- Graphic Engine modelliert die Daten nur
- Rendering setzt die Modelle in eine Darstellung auf dem Bildschirm um:
 - Rendering ist abhängig von den Fähigkeiten der Graphikkarte
 - Setzte ebenfalls auf den Hardware Abstraction Layer auf
- Aufgaben der Rendering Engine:
 - low-level Aufbau und Bearbeitung des Szene Graphen
 - visuelle Effekte (Partikel, dynamische Schatten, high dynamic range rendering (HDR Effekte), Spiegeleffekt, Nebel, ..)
 - Darstellung des Front End (GUI, Video, Menüs, Head Up Display (HUD))

27

Physics Engine

- Simuliert die „Regeln“ der Welt
 - Physikalische Grundregeln (Schwerkraft, Bewegungsgesetze, Trägheit..)
 - Kollisionen (häufig: Physics Engine = Collision Engine)
 - Krafteinwirkung auf Objekte (Zerstörung, Zerschneiden, Verformung..)
 - Explosionen
 - Ragdoll Charaktere (z.B. für dynamische Todesanimationen)
- Physik wird zu einem wichtigen Bestandteil mancher Spiele
 - Spiele und Realtime-Simulationen rücken näher aneinander
 - Animationen basieren auf physikalischen Grundlagen
 - Objekt Interaction nutzt Physik (Kollisionen..)

28

Physics Engine

- Verwendung je nach Genre und Perspektive
 - Simulationsspiele: Sehr hoher Detailgrad, umfangreiche Physik Engines
 - Rundenbasierte Strategiespiele: Keine direkte Physikengine, mögliche Züge werden aus abstrakteren Regeln des Spiels abgeleitet.
- einige verfügbare SDKs:
 - Havok
 - Open Dynamics Engines (ODE)
 - Tokamak
 - PhysX (Nvidia, Angeia) Software in Verbindung mit Spezialhardware (Physics Processing Unit (PPU))

29

Animation Engine

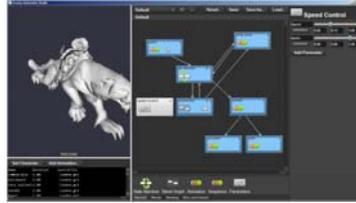
- zuständig für eine natürliche Bewegung der Objekte/Charaktere
- offline Aufgaben:
 - *Motion Capture und Retargeting*: Festhalten von Bewegungsabläufen realer Vorbilder und Übertragung auf andere Computermodelle
 - *Motion Editing und Adaption*: Modifikation aufgenommener Bewegungsabläufe und Übertragung von einem Modell auf ein anderes
- real-time Aufgaben:
 - Animation von Sprites oder Texturen
 - Animation von Graph-Modellen
 - Festkörper- oder Skelettbewegungen

30

Animation Engine

- Packages für die Implementierung von Animation Engines:

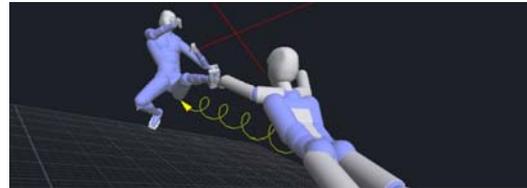
- Granny (weit verbreitet >2000 Spiele)



- Havok Animation (Erweiterung der Havok SDK (siehe Physics Engine))



- Endorphin (für Filme aber auch Spiele)



- Edge (Animation für die PS3)

31

Sound Engine und Audio Manager

- Abspielen und erzeugen von Geräuschen und Musik
- Wichtig für das Spielerlebnis
- Sound Rendering:
 - Abspielen des Sounds in einer bestimmten Simulation
 - Synchronisation von Bild und Ton
 - Abmischen der Geräusche (Lautstärke der Kanäle zueinander)
 - 3D und Entfernungsunterschiede
- Gängige Formate: wave, mp3, midi..
- Sound Manager:
 - XACT for PC and XBOX360
 - EA: SoundRIOT
 - Sony: Scream für PS3
 - IrrKlang, OpenAL, FMOD...

32

Game Core

- Kern der simulierten virtuellen Welt
- Speichert den aktuellen Zustand der Spielwelt
- Kontrolliert den Übergang von einem gültigen Zustand zum nächsten (Game Loop)
- Zeit-Management: echtzeitbasiert(real time), rundenbasiert
- Granularität der Darstellung: Alle für die Regeln des Spiels relevanten Informationen
- Traditionell sehr eng mit den anderen Komponenten verwoben
- Neuere Modelle trennen die Komponenten stärker
- Bei großen Spielwelten:
 - Unterstützung räumlicher Anfragen
 - Aufteilung der Spielwelt
- Mehr im Kapitel 2

33

Network and Multiplayer Module

- Regelt den Datenaustausch über mehrere Rechner hinweg
- Arten von Multiplayer Games:
 - Single-Screen Multiplayer: (z.B. Supermario)
 - eine Sicht auf die Spielwelt
 - Split-Screen Multiplayer: (z.B. Supermario Cart)
 - jeder Spieler hat eigene Sicht auf dem gleichen Bildschirm
 - **Networked Multiplayer**: (z.B. Starcraft, Counter Strike,..)
 - ca. 2-20 Spieler, häufig kein dedizierter Server (dedizierter Client fungiert als Server)
 - geringer Overhead zum Verteilen der Daten (kleine Spielwelten)
 - **Massive Multiplayer Spiele**: (z.B. Eve-Online, World of Warcraft,..)
 - ca. 1000-100.000 Spieler in einer Spielwelt
 - großer Overhead zum Verteilen der Daten (große Spielwelt)
 - aufwendiges Account-Management
 - sehr große Spielwelten

34

Network and Multiplayer Module

- Aufwand je nach Genre: gering bis sehr aufwendig
- Verteilung der Spielfunktionalitäten auf verschiedene Rechner:
 - Synchronisation unterschiedlicher Spielzustände
 - Trennung der Funktionalitäten: Client-Server Modell
 - Alle machen alles und tauschen Daten aus: P2P
- Design von High-Level Protokollen
- User-Authentifizierung
- Beeinflussung der Spielmechanik und des Designs
 - Spielgeschwindigkeit hängt von Latenz-Zeit ab
 - Spielfluss hängt von Serverkapazität ab
- Softwareplattformen für Netzwerkspiele: RakNet, GNE,..
- Mehr im Kapitel 3

35

Persistenz System

- Schicht zur Speicherung von Charakteren und Spielständen
- Speichern von Replays /Spielverläufen
- Bei Online-Spielen:
 - Serveraufgabe
 - Kopplung mit der Account-Verwaltung
 - Absicherung gegen Datenverlust notwendig (Logging und Recovery Lösungen)
 - Effizienzprobleme bei sehr großen Spielständen und Echtzeitspielen
- Mehr im Kapitel 4

36

AI Engine

- Artificial Intelligence (AI) ist generell Teil des Spielinhalts und nicht unbedingt der Game Engine
- Grundfunktionalitäten werden aber mittlerweile über AI Engines gekapselt:
 - Wegewahl in beschränkten Umgebungen:
Suche nach Ein- und Ausgängen, Hindernisse umgehen,..
 - Verhalten und Interaktionen: meist regelbasiert
 - Entscheidungsfindung: ähnlich Expertensystemen
 - Gruppenverhalten: Schwarmbildung, Paniksimulation,..
- Produkte und Packages:
 - AI-implant (Presagis)
 - Kynapse(AutoDesk)
 - DirectAI(Masa)
 - SimBionic
 - ...

37

AI Engine

- Gruppenverhalten kann zentral gesteuert werden:
 - weniger Rechenoverhead
 - leichtere Kooperation der Entitäten
 - unflexible bezüglich Gruppenzusammenstellung
 - in einigen Situationen unrealistisch
- Gruppenverhalten als Agentensystem:
 - Wahrnehmung: Input der Situation
 - Entscheidung: AI (fest oder gelernte Regeln, Zufall)
 - Action: Versuch der Ausführung der Entscheidung
 - Gruppenverhalten ergibt sich aus vielen Einzelhandlungen
 - Kooperatives Verhalten erfordert Kommunikation
 - Rechenaufwand idR höher da jedes Gruppenmitglied einzeln berechnet werden muss
- Mehr in Kapitel 5

38

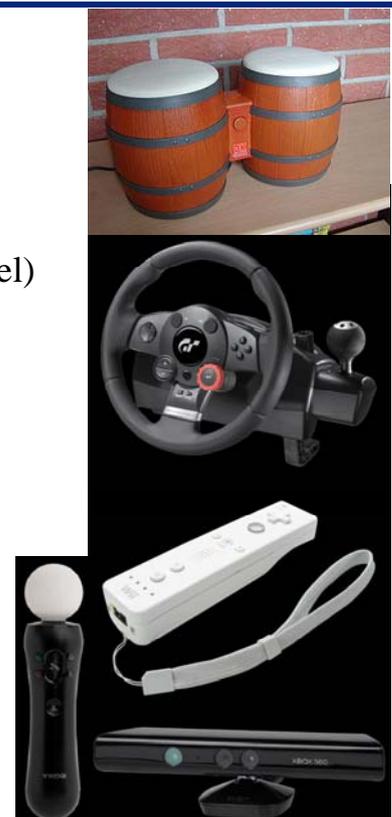
Scripting Engine

- Höhere Programmiersprache mit direktem Zugriff auf die Funktionalitäten der AI-Engine und des Game Cores
- Erleichtert den Designern den Inhalt zu gestalten
- Stellt eine wichtige Schnittstelle zwischen der Game Engine und dem eigentlichen Spielinhalten (Game Content) dar
- Beispiele:
 - LUA (<http://www.lua.org>)
 - GameMonkey (<http://www.somedude.net/gamemonkey>)
 - AngelScript (<http://www.angelcode.com/angelscript>)
- Alternative kann das Spiel auch direkt in der Programmiersprache der Engine entwickelt werden. (häufig effizientere Lösung)

39

User Interface

- Effiziente und intuitive Schnittstellen sind notwendig für ein gelungenes Spielerlebnis:
 - Direkte Reaktion des Spiels auf Eingaben
 - schneller Zugriff auf die notwendigen Fähigkeiten
 - geringe Lernkurve/Learning by doing (Tutorial Level)
- Enge Verknüpfung zur Graphics Engine
- Eingabemethode als Teil des Spielerlebnisses:
 - Eingabe über Spezialcontroller:
Mikrofon, Wii-Gitarre, Lenkräder, DK Bongo,...
 - Eingabe über Bewegungserkennung:
Wii-Controller, X-Box Kinect, PS Move,...



40

Game Content

- Ziele, Aufgaben..
- Spielentitäten: Spielerentitäten (Charakter), Non-Player Entitäten (NPCs, Mobs,..), Umgebungsentitäten(Pflanzen, Felsen..)
- Levels, Karten, Aufgaben,..
- Hintergrundgeschichte
- Videosequenzen
- Balancing: Ausgewogenheit der Einheiten, Karten, des Schwierigkeitsgrades
- Darstellungen: Modelle, Texturen, Animationen, Effekte, Geräusche Musik (Künstlerischer Teil des Spiels)

41

Überblick über die Vorlesung

die Vorlesung gliedert sich in 2 Teile:

1. *Managing Massive Multiplayer Online Games*

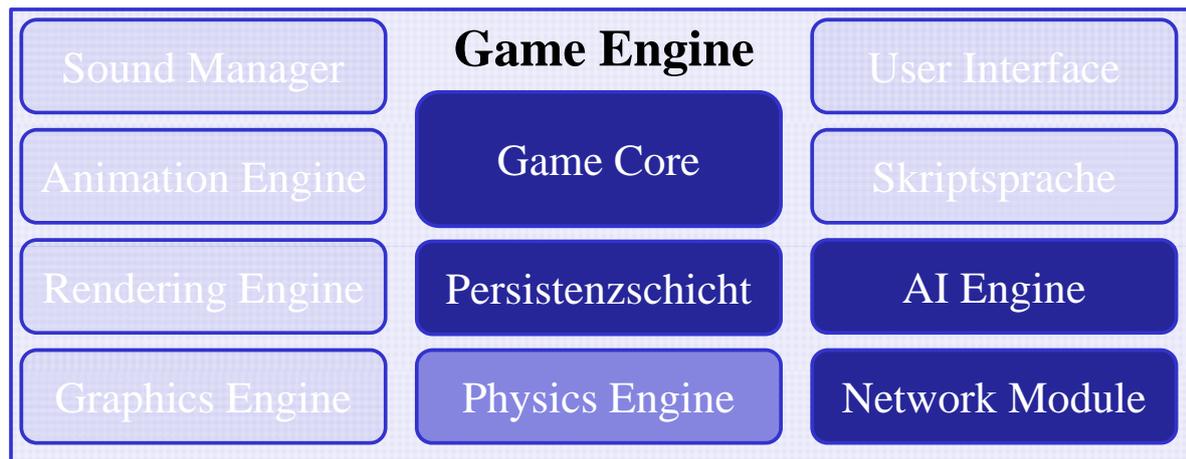
- Untersuchung der Komponenten die für MMOs besonders interessant sind
- Schwerpunkt liegt auf den Game-Server
- Client-Problematiken wie Graphik, Sound, Input/Output werden nicht besprochen

2. Game Analytics

- Vorstellung zu Themenstellung und Aufgabenbereichen
- Einführung in grundlegende Techniken des Data Mining
- Aspekte der Datenbeschaffung und Datenhaltung
- Modellierungsmöglichkeiten für Spieldaten
- Techniken zur Lösung der oben eingeführten Problemstellungen

42

1. Teil der Vorlesung: Managing MMOs



- Untersuchung der dunkel gefärbten Komponenten
- Physics Engine wird in Verbindung mit dem Game Core angesprochen
- 1. Zusätzliches Kapitel (6) beschäftigt sich mit dem Aufbau möglichst betrugssicherer Spiele

43

2. Teil der Vorlesung: Game Analytics

- Erfolgreiches Design eines Computer Spiels hängt stark vom Benutzerverhalten ab:
 - Ist das Spielerlebnis auch nach einiger Zeit noch interessant?
 - Gibt es nicht vorhergesehene Lösungswege, die das Spiel extrem Vereinfachen und dadurch langweilige machen? (Exploits)
 - Passt der Schwierigkeitsgrad zu den Erwartungen meiner Zielgruppe?
 - Verschaffen sich manche Spieler unfaire Vorteile oder umgehen die Regeln ?
 - Wie suche ich passende Gegner in Spieler vs. Spieler Szenarios?
- Die neueren Geschäftsmodelle verstärken den Druck auf den Hersteller ein Dauerhaft interessantes Spielerlebnis zu erschaffen

⇒ **Game Analytics**

Analyse des Spielerverhaltens zur Überprüfung des Einhaltens der Nutzerbedingungen und des Spiel-Designs.

44

2. Teil der Vorlesung: Game Analytics

- Analyse des *Spieler* Verhaltens und des *System* Verhaltens
- Ziele bei der Analyse des Systemverhaltens:
 - Verhalten des Spiels unter Last und realen Bedingungen
 - Adaptive Planung des Hardwarebedarfs aufs der Server-Seite
- Ziele bei der Analyse des Spielerverhaltens:
 - Überprüfung des Game Balancing: Schwierigkeitsgrad, Chancengleichheit, Geschwindigkeit des Spielfortschritts
 - Spielstärkeneinschätzung: Match Making, Spieler Rankings, Dynamische Anpassung des Schwierigkeitsgrades
 - Betrugserkennung: Cheats, Hacks, Exploits, Bots, Account-Kidnapping...
 - Überprüfung der Umsetzung der Design-Ziele:
 - Verwenden Spieler die vorgesehene Strategie?
 - Wie wirken sich Regeländerungen aus?

45

geplante Kapiteleinteilung

1. Multiplayer Online Games

TEIL I: Managing MMOs

2. Der Game Core

3. Netzwerkaspekte

4. Persistenz-System

5. Nonplayer Entitäten

6. Cheat Prevention

TEIL II: Game Analytics

7. Motivation und Aufgaben

8. Data Mining in a Nutshell

9. Telemetry Daten

10. Datenaggregation

11. Sequenzdaten

12. Zeitreihen

13. Räumliche Daten

14. Räumlich-Zeitliche Daten

15. Spielerinteraktionen

46

Literatur

- Jason Gregory: Game Engine Architecture, AK Peters Ltd, MA, 2009
- Nicolas Pronost: Game Engine Programming (<http://www.cs.uu.nl/docs/vakken/mgep/>)
- Critical Gaming Project: (<https://depts.washington.edu/critgame/>)
- Ralph Koster: A Theory of Fun for Game Design, Phoenix, AZ, Paraglyph 2004

47

Dinge die man jetzt kennen sollte

- Definition Computerspiel
- Geschäftsmodelle
- Klassische Genres und Kategorisierungsmerkmale
- Aufbau von Computerspielen
- Komponenten der Spielarchitektur

insbesondere:

- Game Core und Physics Engine
- Netzwerk
- Persistenz System
- AI-Engine
- Persistenz System

48