

Computergrafik 2 Laborübung
Sommersemester 2008



Arabian Fights
Dritte Abgabe

Thomas Mühlbacher	532	0625075	e0625075@student.tuwien.ac.at
Clemens Arbesser	532	0625176	e0625176@student.tuwien.ac.at

In diesem Dokument geben wir einen Überblick über unser Spiel, beschreiben die wesentlichen Features und geben dem Spieler einige Hinweise. Auf technische Details werden wir nur sehr beschränkt eingehen, da diese zu einem großen Teil bereits in der Dokumentation zur zweiten Abgabe behandelt wurden. Aus diesem Grund legen wir dieser Abgabe auch eine Kopie der Dokumentation zur zweiten Abgabe bei.¹ Außerdem präsentieren wir eine Liste der von uns implementierten Effekte.

Arabian Fights

In „Arabian Fights“ fliegt der Spieler auf einem Teppich durch die Welt und bekämpft böse Kreaturen. Dabei steht ein Feuerball-Zauber als Waffe von Beginn an zur Verfügung. Dieser wird durch Besiegen von Gegnern noch stärker, sodass auch härtere Kontrahenten bezwungen werden können. Im Laufe des Spiels findet der Held Wunderlampen (Teapots), die ihm zusätzliche Kräfte verleihen. Ziel des Spiels ist es, zu Überleben und alle Kreaturen zu besiegen.

Die Steuerung

Die Steuerung orientiert sich an Klassikern des First-Person-Shooter – Genres, wobei wir jedoch einen Schwenk der Kamera nach links/rechts unterbinden, da dies die Orientierung behindern und spielerisch keinen Vorteil bringen würde.

Der Hauptangriff des Spielers wird mit einem Klick auf den linken Mausbutton ausgelöst, während der rechte Mausbutton für den Spezialangriff reserviert ist. Letzteren muss man jedoch erst aktivieren, indem man den im Level versteckten Teapot findet und „einsammelt“, d.h. durch ihn hindurchfliegt. Danach kann durch ein Betätigen des rechten Mausbuttons ein vernichtender Feuersturm ausgelöst werden, jedoch nur drei Mal, um das Balancing des Spiels nicht zu stören.

Das HUD dient dem Überblick über wichtige Informationen, die den Standort von Feinden, den derzeitigen Gesundheitszustand und die Mana-Anzeige beinhalten. Letztere füllt sich mit dem Besiegen von Feinden und ist ausschlaggebend für die Größe und Durchschlagskraft der vom Spieler ausgelösten Zaubers. Details zum HUD und den Zaubern behandeln wir im Abschnitt *HUD* bzw. *Partikeleffekte*.

Die Gegner

Die im Spiel verwendeten Models sind zum Teil selbst modelliert, zum Teil auch aus dem Internet entnommen. Sie wurden von uns mit der Software Maya animiert und dem Spiel angepasst, sowie mit Texturen versehen. Dabei haben wir das FBX – Format zum Speichern der Modelle gewählt, die entsprechende Export-Funktion wird dabei von Maya unterstützt. Zum Importieren haben wir uns an den offiziellen FBX-SDK-Samples orientiert.²

Die von uns erstellte KI folgt einem einfachen State-Modell, daher einem deterministischen Automaten. Die Ziele der KI sind dabei einerseits, den Spieler anzupeilen und zu verfolgen, sollte er in ihre Aktionsreichweite kommen, und anzugreifen, solange sie sich in

¹ Zu finden im Verzeichnis doc\2ArabianFightsDokumentation.pdf

² Download: <http://usa.autodesk.com/adsk/servlet/index?siteID=123112&id=6839916>

Angriffsreichweite befindet. Sollte der Spieler fliehen, wird die KI versuchen, ihn zu verfolgen und den Kampf wieder aufzunehmen. Wird der Abstand jedoch zu groß so lässt die KI von der Verfolgung ab.

Die derzeit implementierten Gegner umfassen:

- Vogel (passiv): Dekorationsobjekt, beziehungsweise auch zum Sammeln von Mana
- Vogel (aggressiv): Greift mit schnellen Feuerbällen an, stellt den Anfangsgegner dar
- Dämon: Greifen mit einem sehr gefährlichen Feuersturm an, der den Spieler innerhalb von Sekunden vom Teppich holen kann.

Frustum – Culling und Szenegraph

Um die Performance zu steigern, indem nicht sichtbares Terrain bzw. nicht sichtbare Gegner / Dekorobjekte nicht zu rendern, verwenden wir ein Kegel – Frustum – Culling, mit dessen Hilfe wir einen Szenegraphen berechnen, der alle derzeit sichtbaren Objekte enthält. Für detailliertere Informationen über das verwendete Culling verweisen wir auf unsere zweite Abgabe.

Terrain

Das Terrain wurde bereits hinlänglich in der zweiten Abgabe behandelt, es wurde von technischer Seite nicht verändert. Lediglich das Level wurde von uns neu erstellt (Software L3DT) und mit Dekorationsobjekten sowie Gegnern befüllt.

Das Heads – Up – Display

Das HUD beinhaltet eine schematische Minimap, die sowohl die Position der Gegner als auch der Dekorationsobjekte anzeigt, und auch den Gesundheits – und Manazustand des Spielers. Ersterer ist selbsterklärend, letzterer ist ein Indikator für die „magische Kraft“ des Spielers und beeinflusst die Größe und Stärke der Angriffszauber. Sie kann durch das Besiegen von Feinden vergrößert werden, wobei schwierigere Gegner einen größeren Manabonus hinterlassen. Außerdem wird, falls der Spieler von einem Gegner getroffen wird, der Bildschirm kurz rot eingefärbt.

Partikeleffekte und Waffen

Alle im Spiel verwendeten Zaubersprüche basieren auf einem relativ einfachen Partikelsystem, das äußerst flexibel ist und somit leicht angepasst werden kann, um unterschiedliche Effekte darzustellen.³ Um einen möglichst realistischen Eindruck zu

³ Zusammenstellung aus verschiedenen Quellen:

Implementierungsdetails: <http://nehe.gamedev.net/data/lessons/lesson.asp?lesson=19>

http://gpwiki.org/index.php/OpenGL_Tutorial_Framework:Particles

Theoretische Grundlagen: http://www.mhl.de/Papers/ParticleSystems/Particle_System_Tutorial.pdf

hinterlassen, werden die Partikel, nachdem sie zum Betrachter gedreht worden sind, mit einer Vielzahl von zufallsgesteuerten Parametern in eine vordefinierte Richtung geschleudert. Dies lässt sich im Spiel gut bei der Benutzung von Zaubersprüchen erkennen, bzw. beim Besiegen eines Gegners oder auch beim Auftreffen von Zaubersprüchen auf dem Terrain. Ausserdem verändern die Partikeleffekte mit steigendem Mana des Spielers ihr Aussehen und sorgen daher für eine größere Abwechslung.

Implementierte Effekte

Effekt	Kann gefunden werden durch:
Dynamisches Level-of-Detail des Terrains	Einschalten des WireFrame - Modus
Partikeleffekte	Aktivieren von Zaubersprüchen Besiegen von Gegnern Aufprall von Zaubern auf dem Terrain
QuadTree mit angepasster Quad-Größe	Einschalten des WireFrame - Modus
Bereits implementiert, jedoch noch nicht aktiviert	Grund
statisches Level-of-Detail	Architektur bereits vorhanden, Modelle müssen noch „downgesampelt“ werden
Vertex Buffer Objects	Bereits implementiert, resultiert aber derzeit in einem zerstörten Terrain

Leitfaden durch das Spiel

Der Spieler startet das Spiel in einem relativ ruhigen Abschnitt des Levels; seine ersten Ziele sollten die überall anzutreffenden Vögel sein, um seinen Manavorrat aufzufüllen. Danach kann der Spieler versuchen, die in den Ecken des Levels anzutreffenden Dämonen anzugreifen, oder aber zuvor noch die versteckte Teekanne zu finden, um den Spezialangriff freizuschalten.

Nachbesserungen für das Spielevent

Bis zum Spieleevent nächsten Freitag möchten wir unser Spiel auf jeden Fall noch um folgende Features erweitern, die sich bisher aufgrund von Prüfungen nicht ausgingen:

- in erster Linie Content, d.h.
 - mehr Gegner (ein paar Models + KI)
 - mehr Landschaftsdekoration
 - mehr Wunderlampen (für Waffen, Heilung, Geschwindigkeitszauber)

- ordentliche Levelgrenze (d.h. das Terrain wird nach allen Seiten erweitert, der Spieler wird aber gezwungen, nach Überschreiten der „Grenzen“ bald umzudrehen. „Deserters will be shot!“)

- Spielgewinn und Spielverlust müssen deutlicher vor Augen geführt werden -> Spieler kann sterben bzw. ein Spielziel erreichen + Belohnung

- statisches Model – Level Of Detail (Modelgeometrie reduzieren)

Weiters hätten wir gerne folgende Features, sofern noch Zeit bleibt:

- fehlende OpenGL-Spielereien, i.e. VBO-Support, Zur-Laufzeit-Ein-und-Ausschalten der restlichen Features

- ein paar Sounds