

Damn Dam

Abgabe 3

Walter Huimann
Katrin Lasinger

033 532 0625373
033 532 0726670

e0625373@students.tuwien.ac.at
e0726670@students.tuwien.ac.at

Kurzbeschreibung

Das Spiel ist ein Puzzlespiel, in welchem Löcher in einem Damm gestopft werden müssen. Der Spieler muss dabei verschiedene Gefahren überwinden.

Steuerung

Die Spielfigur wird vom User mit der Standard-Tastatursteuerung WASD bzw. den Cursortasten gesteuert. Dabei kann er sich in 4 Richtungen bewegen (vor, zurück, links oder rechts).

Die Kameraansicht orientiert sich an der Position des Hauptcharakters, welcher immer in der Mitte des Bildschirms gezeigt wird. Sie kann vom User in X-Richtung gedreht werden indem er die linke Maustaste gedrückt hält und die Maus nach links bzw. rechts bewegt. Um den Spieler mit der Steuerung nicht zu verwirren kann die Kamera nur 40° nach links bzw. 40° nach rechts gedreht werden.

Ziel des Spieles

Der User versucht die Spielfigur ins Ziel des jeweiligen Levels zu bewegen. Dieses ist durch eine Zielflagge gekennzeichnet. Hat er das Ziel erreicht, gelangt der Spieler zum nächsten Level.

Implementierung

In unserer Game-Loop werden all unsere GameObjects upgedated und gezeichnet. Zu den GameObjects zählen die Kamera, die Spielfigur (Token), das Spielbrett und Hintergrundobjekte. Das Spielbrett besteht aus einem 2 dimensionalen Array von Quadraten, abhängig vom Level werden bestimmte Quadrate gezeichnet.

Der Token wird bei Userinteraktion bewegt (mit WASD oder Pfeiltasten) und startet eine Animation bei welcher er sich rotierend zur nächsten Position bewegt. Die Kamera folgt der Position des Token und kann mit der Maus, mit gedrückter linker Maustaste, bewegt werden.

Die benötigten Ressourcen werden aus dem ResourceManager geladen (Shader, Texturen, Levels, ...). Die Modelle und Animationen werden mit einem md2-Modelloader geladen. Die Levels werden aus einfachen Textfiles geladen.

Gameplay

Der Spieler steuert die Spielfigur (einen Quader) in vier Richtungen auf dem Spielbrett mit quadratischen Spielfeldern und versucht zum Ziel zu gelangen. Dabei kann er sich nur auf bestimmten Feldern bewegen.

Außerdem kann die Spielfigur an speziellen Feldern ihr Material zwischen Stein und Holz wechseln. Mit Stein kann der Spieler durch Feuer gehen, mit Holz kann er sich über Wasser bewegen. Mit dem jeweils falschen Material verbrennt er oder geht unter. In diesem Fall muss der Spieler das Level neu starten.

Hat der Spieler das Ziel erreicht so gelangt er ins nächste Level.

Der Schwierigkeitsgrad der Levels steigt im Laufe des Spieles.

ACHTUNG: Die Spielfigur muss komplett auf dem Zielfeld bzw. auf einem Wechselfeld stehen. Es reicht nicht wenn die Spielfigur nur teilweise darauf liegt.

Nichttriviale Objekte & animierte Objekte

Die im Spiel vorhandenen Objekte sind md2-Models, welche mit einem Modelloader in das Spiel geladen werden. Dadurch ist es kein Problem nichttriviale Objekte darzustellen.

Zusätzlich können in den md2-Models auch Key-Frame Animationen gespeichert werden welche wir auch abspielen können. Beispiele dafür im Spiel sind der Vogel und die Zielflagge.

Objekte im Spiel:

- Spielstein – durch User bewegt, 2 verschiedene Texturen, animiertes „Umfallen“
- Spielfelder – verschiedene Spielsteine (normales Feld, Ziel mit Flagge, Feuer, Wasser, Wechselfeld)
- Vogel – animiertes Objekt im Hintergrund, fliegt über dem Spielfeld – hat keinen Einfluss auf das Spiel
- Damm – Hintergrundbild, nach innen gewölbte Halbkugel auf der eine Dammtextur zu sehen ist

Transparenz-Effekt

Die „Wasser-Spielfelder“ und „Wechsel-Spielfelder“ werden im Spiel halbtransparent angezeigt, das heißt Objekte welche sich hinter dem Objekt befinden scheinen leicht durch. Dies wurde mit dem Alphakanal der Textur realisiert. Außerdem werden die transparenten Objekte korrekt tiefensortiert nach den nichttransparenten Objekten gezeichnet. Die Transparenz kann durch die Taste F9 während des Spieles ein- und ausgeschaltet werden.

Spezialeffekte

Shadow Maps

Implementierung laut Tutoriums-Folien der CG2-LU:

First-Pass

Berechnung der Entfernung von der Lichtquelle mithilfe eines eigenen Shaders. Dabei wird nur die Position der Vertices verwendet, andere Informationen werden ignoriert. Die Tiefenwerte werden in einer eigenen Textur gespeichert, der Shadow Map.

Second-Pass

Im nächsten Schritt wird die Szene „normal“ gerendert aus Sicht der Kamera. Die Shadow Map wird dabei verwendet um zu bestimmen ob das gezeichnete Pixel im Schatten liegt oder nicht. Liegt es im Schatten wird es nicht mit diffuser Beleuchtung beleuchtet (nur mit ambienter Beleuchtung).

Sonstige Besonderheiten

- Md2-Modelloader
- Md2 Key-Frame Animationen
- Levels können leicht selbst erstellt werden (näheres dazu beim Punkt „Levels erstellen“)

Modellierte Objekte

Einige unsere Modelle haben wir aus dem Internet heruntergeladen (http://telias.free.fr/models_md2_menu.html).

Die restlichen haben wir mit dem 3D-Modeller Milkshape erstellt (<http://chumbalum.swissquake.ch/ms3d/index.html>).

Aufgrund mangelnder Modellierungskünste haben wir hauptsächlich einfache Modelle verwendet.

Experimentieren mit OpenGL

F3 - Wire Frame ein/aus

F4 - Texturqualität verändern: Nearest Neighbor/Bilinear

F9 - Transparenz ein/aus.

Verwendete Libraries

GLFW, GLEW, GLM

Levels erstellen

Im Unterordner levels befindet sich ein File „levels.data“. In diesem können neue Levels hinzugefügt werden. Einfach in einer neuen Zeile den Pfad des Files angeben.
Ein Level-File hat folgende Struktur:

```
xKoordinate1 yKoordinate1 Status1  
xKoordinate2 yKoordinate2 Status2  
...
```

Bsp.:
0 0 1
1 0 1
2 0 1

Folgende Statuszahlen sind möglich:

- 1 ... normales Feld
- 2 ... Feuer
- 3 ... Wasser
- 4 ... Startfeld
- 5 ... Endfeld
- 6 ... Wechselfeld

Es darf nur ein Start- und ein Endfeld geben und es sind nur Koordinaten > 0 möglich. Das Spiel überprüft nicht ob das Level korrekt ist oder gar ob das Ziel erreicht werden kann.

Quellen & Hilfestellungen

Implementierung:

- CG2-Folien
- Informatikforum
- <http://nehe.gamedev.net/>
- <http://tfc.duke.free.fr/coding/md2-specs-en.html>

Modelle:

- http://telias.free.fr/models_md2_menu.html

Texturen

- Total Textures