

Programmieraufgabe

Bearbeitungszeitraum: 10.06.15 -- 10.07.15

Als Upload in Moodle

Prof. Dr. Detlef Krömker

Professur für Graphische Datenverarbeitung

Thorsten Gättinger, M.Sc.
Dipl.-Inf. Alexander Wolodkin

Ein einfaches 3D Spiel

Σ 20

In diesem Aufgabenblatt soll ein kleines Spiel in 3D mit lwjgl3 erstellt werden. Dieses Aufgabenblatt ist in 2er Gruppen zu bearbeiten.

Bei dem Spiel handelt es sich um eine Variante von Pong: Curveball

<http://www.albinoblacksheep.com/games/curveball>

Das Spiel besteht aus:

- Einer n -eckigen Röhre, die an ihren Enden offen ist, in die der Spieler schaut
- Einem Ball
- Zwei Schlägern. Der Schläger des Spielers befindet sich im Vordergrund und ist halbtransparent. Der hintere Schläger wird von der KI gesteuert.

Aufgabe 1: Texturierte Levelobjekte (6 Punkte)

a) Generieren Sie zunächst die Levelobjekte und stellen Sie diese dar. Sie können entweder die Geometrien aus den vorherigen Aufgaben weiterverwenden oder auch externe Objekte benutzen¹. Achten Sie auf korrekte Normalen, vor allem in der Röhre. Es genügt sich bei der Röhre auf einen Quader zu beschränken. Die Objekte müssen nicht als TriangleStrip gezeichnet werden. Eine Angabe von je 3 Vertices pro Polygon mit *glDrawArrays()* ist ausreichend. (3 Punkte)

b) Die benutzten Objekte sollen texturiert werden. Dabei haben Ball, Röhre und Schläger jeweils eine eigene Textur. Es dürfen keine monochrome Texturen ohne erkennbare Struktur verwendet werden.

Um Texturen in ein für OpenGL geeignetes Format zu laden können Sie aus dem Beispielpogramm die Funktion *loadPNGTexture()* verwenden. Diese Funktion ermöglicht es eine PNG-Datei als Textur einzulesen.

Um die Textur auf die Objekte zu mappen ist es nötig alle Vertices um eine zweidimensionale Texturkoordinate zu erweitern. Dies erfolgt analog zu den Vertexattributen Farbe und Normale. Die Textur selbst ist von (0,0) bis (1,1) definiert, unabhängig von Größe und Seitenverhältnis. Dementsprechend dürfen die Objekte auch nur Texturkoordinaten aus diesem Bereich nutzen. Für jedes Polygon wird anhand seiner drei Vertices und deren Texturkoordinaten das entsprechende Teilstück der Textur dargestellt.

Neben dem Laden der Texturen und der Angabe der Texturkoordinaten im Objekt müssen diese noch im Shader verarbeitet werden. Auch hier können Sie sich am Beispielpogramm orientieren. (3 Punkte)

¹Quelle angeben und Copyright beachten!

Aufgabe 2: Spiellogik (4 Punkte)

Implementieren Sie die Ballbewegung. Der Ball wird bei Spielstart in Richtung des Gegners bewegt. Ab jetzt bewegt sich der Ball in seiner Richtung weiter, bis eine Kollision mit der Röhre oder einem Schläger auftritt, oder der Ball verlässt den Spielbereich (= passiert den Schläger). Es genügt den Ball in einer geraden Flugbahn zu bewegen, die gekrümmten Flugbahnen und das Abprallverhalten in Abhängigkeit der Rotation des Balls wie im Original Curveball können Sie vernachlässigen. Bei Kollisionen mit der Röhre entspricht der Ausgangswinkel dem Eingangswinkel. Bei einem Eingangswinkel von 45° liegt dementsprechend ein Rechter Winkel zwischen Eingangs- und Ausgangswinkel. Kollisionen mit dem Schläger verhalten sich im Mittelpunkt des Schlägers ebenso. Je weiter der Aufttrittspunkt am Rand des Schlägers liegt, desto mehr wird die Flugbahn des abprallenden Balls in Richtung dieses Randes gehen.

Implementieren Sie desweiteren eine einfache KI, die Ihre Bälle zurückspielt. Es genügt, wenn sich der Schläger immer in Richtung des Balls gewegt. Es muss möglich sein, dass Sie gegen die KI gewinnen können. Dies kann auch erreicht werden, indem sich die Schwierigkeit schon innerhalb eines Levels schrittweise erhöht.

Um den Spieler zu fordern soll es mehrere Levels geben, die fortlaufend schwieriger werden. Die Schwierigkeit in einem Level kann beispielsweise durch die Geschwindigkeit des Balls, die Stärke der KI oder der Größe des Schlägers abhängen. Hier genügt es eine der genannten Beeinflussungen umzusetzen, oder einer eigenen Idee. Sehen Sie wie im Original Curveball mehrere Versuche vor, bis Ihr Spiel beendet ist und Sie wieder bei Level 1 anfangen. Ebenso wird automatisch das Level gewechselt, wenn die KI besiegt wurde.

Aufgabe 3: Interaktion (3 Punkte)

Implementieren Sie eine Spielesteuerung mit Maus und Tastatur. Die Bewegung des Schlägers erfolgt über die Maus, kann zusätzlich auch noch über die Tastatur erfolgen. Dokumentieren Sie, wie die Steuerung belegt ist. Folgende Interaktionen müssen mindestens verfügbar sein:

- Starten des Spiels
- Reset
- Pause
- Bewegung des Schlägers

Aufgabe 4: Spielanzeige (2 Punkte)

Geben Sie den aktuellen Punktestand und Status des Spiels graphisch aus. Nutzen Sie dazu Texturen. Sie können sich auf eine feste Anzahl von Leveln beschränken und können somit ausschließlich statische Texturen verwenden.

Aufgabe 5: Weitere Features (5 Punkte)

Aus diesem Bereich sind maximal 5 Punkte anrechenbar. Sie können sich somit frei entscheiden, welche Features Sie umsetzen möchten.

- a) Das Alleinstellungsmerkmal von Curveball gegenüber Pong ist die Beeinflussung von Flugbahn (Krümmung) und Abprallverhalten durch die Schlägerbewegung. Das Verhalten muss nicht physikalisch zu 100% korrekt sein, es muss sich nur richtig anfühlen. Setzen Sie dieses um. (3 Punkte)
- b) Implementieren Sie eine weitere n -eckige Röhre, wobei $n > 4$. Sehen sie eine Möglichkeit vor zwischen diesen zu wechseln. Passen Sie dazu auch die Schnittpunktberechnung an. (3 Punkte)
- c) Implementieren Sie eine n -eckige Röhre für beliebige $n > 4$. Es genügt im Spiel zwischen 3 verschiedenen n wechseln zu können, im Code selbst soll es variabel bleiben. Auch hier muss die Schnittpunktberechnung dementsprechend angepasst werden. (5 Punkte)
- d) Um die Tiefenwahrnehmung zu unterstützen soll die Ballposition in der Tiefe visualisiert werden. (2 Punkte)
- e) Fügen Sie Sounds ein. Es soll mindestens Hintergrundmusik, Töne bei der Kollision und bei Verlust des Balls geben. (2 Punkte)

Aufgabe 6: Abgeben

Laden Sie ihre Abgabe (Quellcode und Dokumentation) als .ZIP-Datei im Moodlekurs hoch. Eine entsprechende Aufgabe ist dafür eingerichtet.

Beachten Sie, dass für die Abgabe eine Dokumentation ihres Programmes zwingend notwendig ist. Sollte diese fehlen, wird die Abgabe als nicht eingereicht gewertet. Die Dokumentation soll dabei folgende Punkte umfassen:

- Planung des Vorgehens (Zeitplan, Arbeitsteilung, Milestones)
- Konzeption des Spieles
- Beschreibung besonderer Funktionen oder Libraries
- Geeignete Testfälle für das Spiel, welche extern gespeichert werden können (pdf, doc, txt)

Der Umfang der Dokumentation, welche als .PDF in der .ZIP-Datei enthalten sein soll, darf nicht mehr als 5 Seiten in Anspruch nehmen. Da es sich hierbei um ein längerfristiges Projekt handelt, ist eine Planung im Vorfeld unvermeidbar. Die Definition von Milestones hilft ein gezieltes Erreichen der gesteckten Ziele zu gewährleisten. Diese Planung ist Teil der Projektdokumentation und entsprechend zu notieren.

Hinweise

- Die Aufgabe soll in Zweier-Gruppen bearbeitet werden. Diese sind den Betreuern mindestens 2 Wochen vor der Abgabe mitzuteilen.
- Zitieren Sie alle Quellen, welche für die Erstellung des Programms benötigt wurden.
- Die Abgabe ist (Quellcode + externe Daten) als gepackte Datei in Moodle hochzuladen. Es muss nur 1 Person den Quellcode hochladen.
- Die Bearbeiter einer Datei sind in einem entsprechenden Header zu vermerken. Es ist darauf zu achten, dass die Arbeit möglichst gleichverteilt ist. Sollten Sie Dateien haben, in denen beide Teilnehmer gearbeitet haben, ist anzugeben wer welche Teile umgesetzt hat.

- Es wird noch je ein Theorieblatt und Programmierblatt geben. Diese Aufgabenblätter werden sich mit der Bearbeitungszeit des Spiels überschneiden. Sie sollten daher versuchen in 2 Wochen den Großteil des Spiels implementiert zu haben.