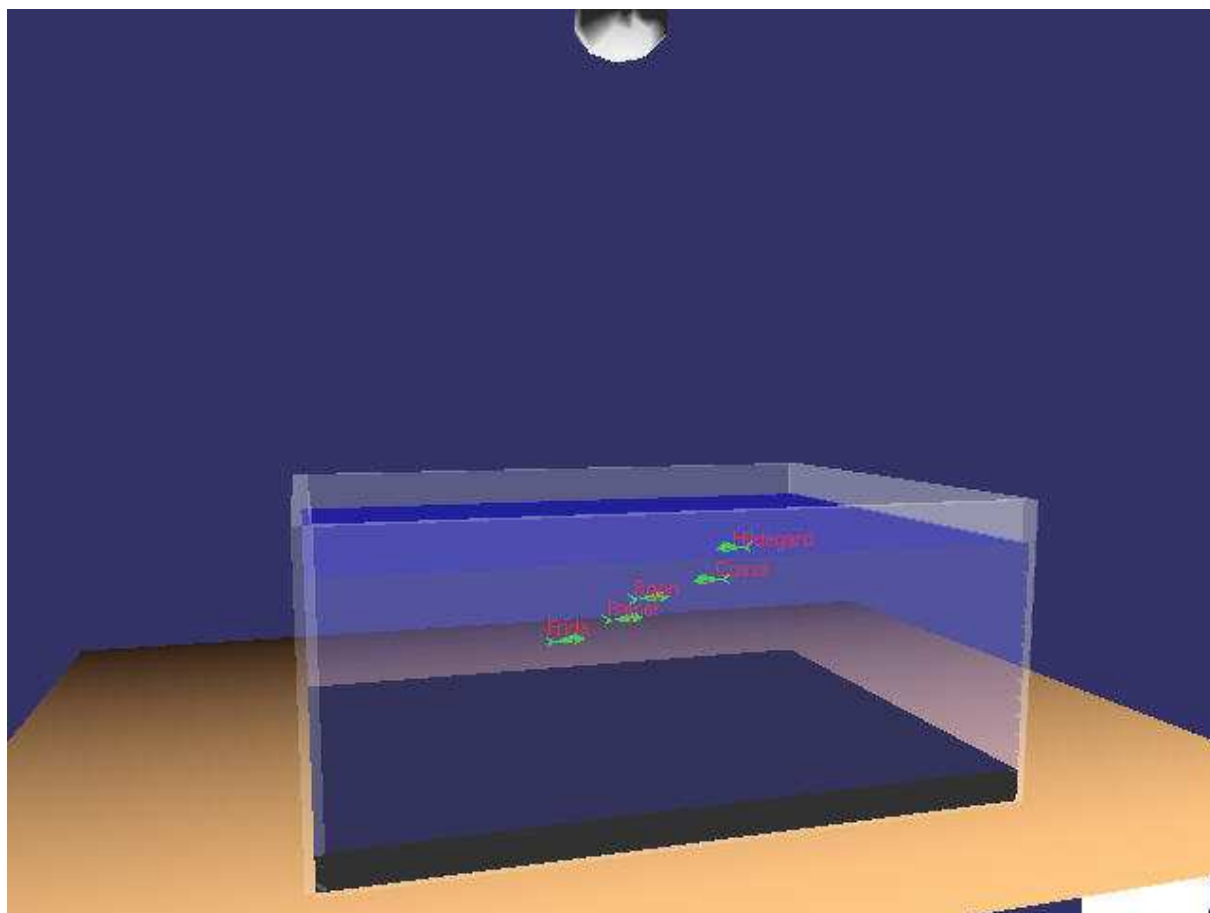


Dokumentation des OpenGL-Projektes



Manuel Hanke
451259

Inhaltsverzeichnis

Beschreibung der Szene	3
Screenshots.....	3
Umsetzung der Regeln	6
Separation.....	6
Cohension.....	6
Alignment.....	6
Experiment Max_Fische	7
Der Hai	7
Hotkeys.....	8
Quellenangabe	9

Beschreibung der Szene

In meinem OpenGL Programm habe ich ein Aquarium auf einem Tisch positioniert in welchem 5 Fische schwimmen. Das Aquarium besteht aus 4 Glaswänden und einer schwarzen Bodenplatte.

Der Fischkörper, samt beweglicher Heckflosse, ist grün gefärbt. Für die Seitenflossen habe ich Rot gewählt, damit man diese besser erkennen kann. Über jedem Fisch wird sein Namen in einem roten Schriftzug angezeigt. Den Fisch „Rainer“ kann man mithilfe der Tasten „+“ und „-“, leuchten lassen. Dazu muss allerdings vorher das Licht ausgeschaltet werden, da man es sonst nicht bemerkt..

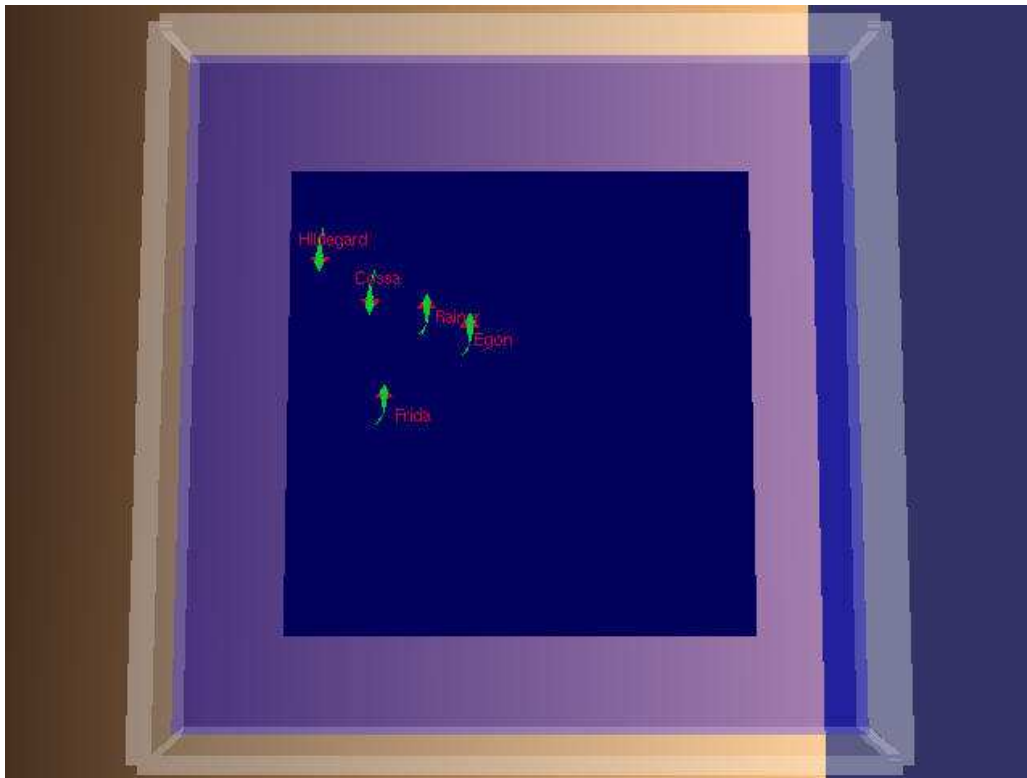
Über dem Aquarium befindet sich eine Kugel, welche den Tisch und das Aquarium beleuchtet. Sie lässt sich über die Taste „L“ aus und wieder anschalten.

Des Weiteren kann man durch Drücken der Taste „H“ einen Hai ins Aquarium lassen. Dieser hat aus Zeitgründen leider kein richtiges Jagdverhalten mehr bekommen, frisst jedoch alle Fische die ihm in den Weg kommen. Durch Drücken der Taste „R“ kann man wieder alle 5 Fische auf ihre Anfangspositionen zurücksetzen.

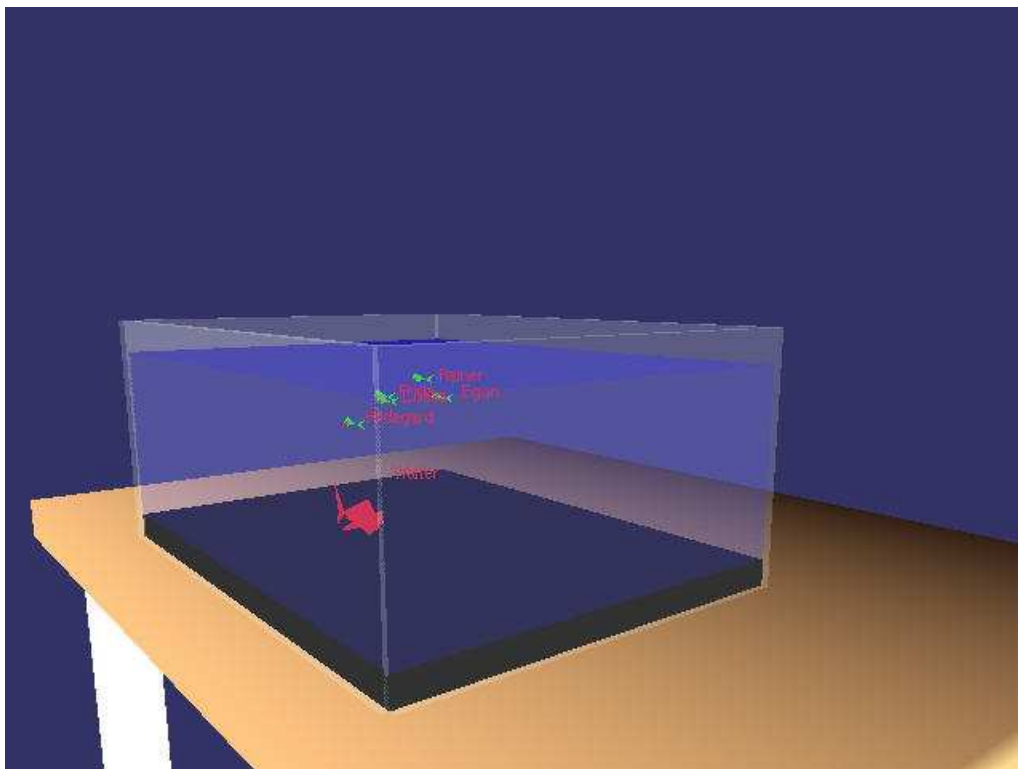
Screenshots



(Ansicht von vorne mit ausgeschaltetem Licht und leuchtendem Fisch)



(Ansicht aus der Vogelperspektive)



(Ansicht von schräg hinten mit Hai)

Umsetzung der Regeln:

Separation:

Die Regel Separation habe ich umgesetzt indem ich die Liste der Fische in jeder Runde durchlaufe und für jeden Fisch überprüfe ob sich ein weiterer Fisch entweder auf der X, Y oder der Z-Koordinate nähert, bzw. ob schon ein anderer Fisch zu nah ist. Ist dies der Fall wird überprüft auf welcher der 3 Achsen dies der Fall ist.

Ist schließlich herausgefunden von wo die Kollision droht wird bei beiden Fischen die Schwimmgeschwindigkeit der Kollisionsachse verringert bzw. erhöht wenn die Kollisionsgefahr von der anderen Richtung droht.

Droht keine Gefahr der Kollision soll sich der Fisch der Schwarmgeschwindigkeit für alle 3 Achsen angleichen.

Bei der Kollision mit der Aquariumswand habe ich mit den statischen Werten des Aquariums gearbeitet. Kommt eine Wand dem Fisch zunahe dreht sich dieser um 180° und schwimmt in die Gegenrichtung weiter.

Cohension:

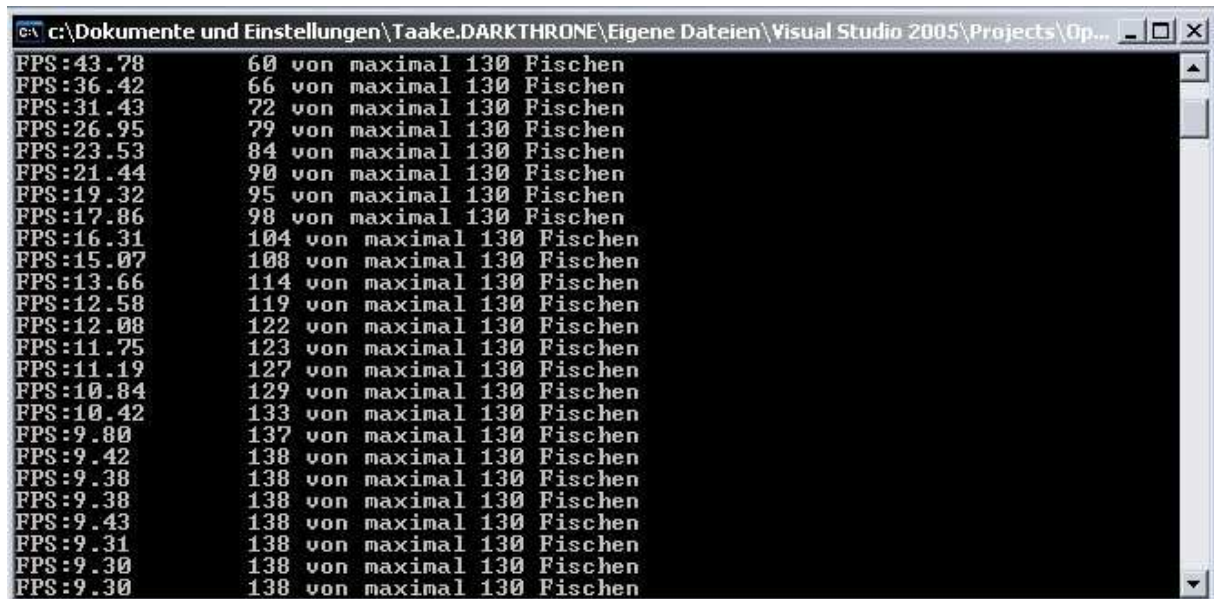
Die Regel Cohension habe ich umgesetzt indem ich auch hier wieder die Liste der Fische durchgehe und bei jedem Fisch überprüfe wo sich der Fisch im Bezug auf den Mittelpunkt des Schwarmes befindet. Befindet er sich auf der jeweiligen Achse zu weit im positiven, dann wird der Delta Wert der Achse verringert, sollte er sich zu weit im negativen befinden so wird er erhöht.

Alignment:

Da der große Teil meiner entwürfe für diese Realisierung gescheitert waren habe ich mich entschlossen lediglich die Ausrichtung in X-Richtung zu machen. Dort dreht sich der Fisch wieder wenn er an eine Aquariumswand stößt.

Experiment Max_Fische:

Die FPS-Rate fällt in meinem Aquarium unter 10, sobald sich mehr als 130 Fische in meinem Aquarium befinden. Wie man auf folgendem Screenshot gut erkennen kann:



```
c:\Dokumente und Einstellungen\Taake.DARKTHRONE\Eigene Dateien\Visual Studio 2005\Projects\Op...
FPS:43.78      60 von maximal 130 Fischen
FPS:36.42      66 von maximal 130 Fischen
FPS:31.43      72 von maximal 130 Fischen
FPS:26.95      79 von maximal 130 Fischen
FPS:23.53      84 von maximal 130 Fischen
FPS:21.44      90 von maximal 130 Fischen
FPS:19.32      95 von maximal 130 Fischen
FPS:17.86      98 von maximal 130 Fischen
FPS:16.31     104 von maximal 130 Fischen
FPS:15.07     108 von maximal 130 Fischen
FPS:13.66     114 von maximal 130 Fischen
FPS:12.58     119 von maximal 130 Fischen
FPS:12.08     122 von maximal 130 Fischen
FPS:11.75     123 von maximal 130 Fischen
FPS:11.19     127 von maximal 130 Fischen
FPS:10.84     129 von maximal 130 Fischen
FPS:10.42     133 von maximal 130 Fischen
FPS:9.80      137 von maximal 130 Fischen
FPS:9.42      138 von maximal 130 Fischen
FPS:9.38      138 von maximal 130 Fischen
FPS:9.38      138 von maximal 130 Fischen
FPS:9.43      138 von maximal 130 Fischen
FPS:9.31      138 von maximal 130 Fischen
FPS:9.30      138 von maximal 130 Fischen
FPS:9.30      138 von maximal 130 Fischen
```

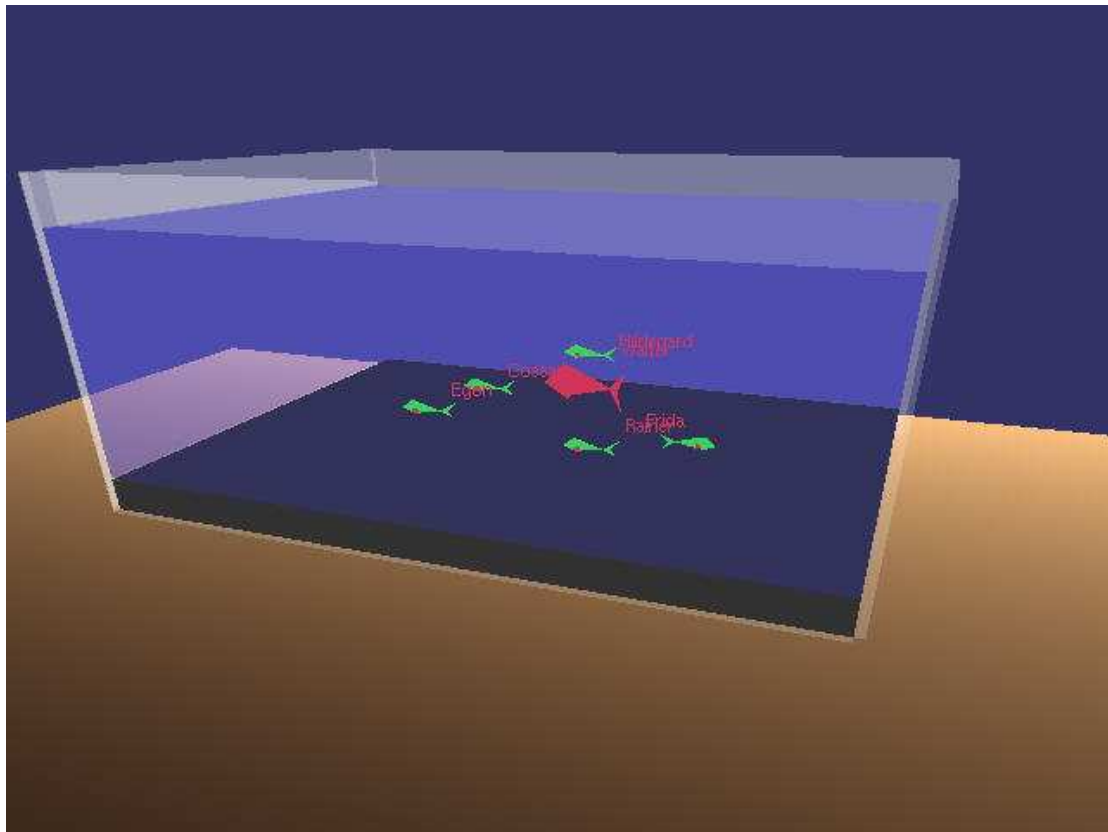
Der Hai:

Ich habe als Bonus einen Hai eingebaut, den man mit der rechten Maustaste aktivieren bzw. deaktivieren kann.

Aus zeitlichen Gründen habe ich es nicht mehr geschafft dem Hai ein realistisches Jagdverhalten beizubringen, deshalb schwimmt er eine konstante Bahn im Aquarium. Sobald ein Fisch ihm vor die Nase schwimmt frisst ihn der Hai und der Fisch verschwindet. Hat der Hai alle Fische gefressen, oder wirkt das Aquarium leer, so kann man über die Taste „R“ die alle Fische wieder zurück auf ihre Startpositionen setzen.

Implementierungstechnisch erbt die Hai Klasse von der Fisch Klasse so dass dem Hai schon das vereinfachte Schwimmverhalten zur Verfügung steht.

Auf der folgenden Seite befindet sich ein Screenshot aus der Vogelperspektive inklusive des Hais.



(Der Hai Walter)

Hotkeys:

Tastatur ,+'	Erhelle die Leuchtkraft des Fisches Rainer
Tastatur ,-'	Verdunkle die Leuchtkraft des Fisches Rainer
Tastatur ,L'	Schalte das Licht an bzw. aus
Tastatur ,R'	Fische werden neu eingesetzt (Reset)
Linke Maustaste	Wechsel der Kameraposition
Rechte Maustaste	Aktiviere bzw. deaktiviere Hai

Quellenangabe:

Codeworx OpenGL Tutorials:

http://www.codeworx.org/opengl_tuts.php

Verwendet bei der Primitiven Klasse und der Transparenz des Aquariums

Lighthouse3d OpenGL GLUT Tutorial:

<http://www.lighthouse3d.com/opengl/glut/>

Verwendet bei dem FPS Zähler, Maus- und Tastatursteuerung

Robsite OpenGL Einführung:

<http://www.robsite.de/daten/tutorials/ogleinfuehrung.pdf>

Verwendet bei der Ansichtstransformation

FH-Wedel OpenGL:

<http://www.fh-wedel.de/~hu/iasem-ws03-04/09/Lichteffekte.htm>

Verwendet beim Spotlight

Die restlichen Programmfunktionen wurden durch die Vorkenntnisse aus der Vorlesung Computergrafik, Programmieren 2 und Programmieren 3 realisiert.