

Evasion

Für die Kommunikation zwischen Server und Client werden TCP Sockets verwendet. Zur Darstellung am Client wird OpenGL mit GLUT, GLU und GLEE verwendet.

Mittels der TCP-Sockets werden so genannte Commands ausgetauscht, auf die von der jeweilig anderen Seite reagiert wird. Diese Commands sind wie folgt aufgebaut:

CommandName(Parameter1,Parameter2, Parameter3,...)

Zum Beispiel sendet der Server zum Client "Delete(55)" was soviel bedeutet wie: "Lösche das Objekt mit der ID 55!".

Die ID wird beim Server durch einfaches Hochzählen erzeugt.

Server

Hauptthread

Hier wird zuerst mittels Konsole eine Konfiguration eingegeben. Danach wird die Welt mit den eingegeben Daten generiert und der Serverthread gestartet. Ab jetzt wird so lange das Programm läuft auf Clients gewartet und für jeden Client ein Clientthread erzeugt.

Serverthread

In diesem Thread werden alle 5 Sekunden 10 + für jeden Spieler 1 Projektil erzeugt, die Richtung Kugeloberfläche fallen, wobei sich das eine zur Position des Spielers beim Erzeugen bewegt. Beim Auftreffen werden diese Projektile dann in ganz normale Obstacles umgewandelt.

Clientthread

Sobald dieser Thread startet wird ihm eine ID mittels eines Commands zugewiesen. Danach werden ihm alle anderen Objekte der Welt übertragen und die Daten des Clients werden den anderen Clients gesendet. Danach wird nur noch auf Commands vom Client gewartet.

Client

Hauptthread

Im Hauptthread wird als erstes OpenGL initialisiert und die Konfiguration ausgelesen, falls sie vorhanden ist. In dieser Konfiguration steht in der ersten Zeile die Adresse und der Port von dem Server zu dem das letzte Mal verbunden wurde. Ist keine Konfiguration vorhanden oder der Server nicht erreichbar kann der Benutzer eine neue Adresse und einen neuen Port eingeben.

Sobald zum Server verbunden wurde wird der Netzwerkthread gestartet.

Von nun an wird alle 30 Millisekunden die Update Methode aufgerufen und parallel dazu in der Display-Methode gerendert.

Display-Methode:

Zuerst wird der jetzige Bildschirminhalt gelöscht und die Modelview- und Projection-Matrix zurückgesetzt. Danach wird die Kamera hinter den Spieler bewegt und die Welt gezeichnet.

Dies funktioniert in dem vom World-Objekt die Draw-Methode aufgerufen wird. Diese zeichnet die Welt und ruft die Draw-Methode von jedem Objekt in der Welt auf. Ganz zum Schluss werden dann Zeichenbuffer ausgetauscht.

Update-Methode:

Hier wird als erstes die Beschleunigung, Geschwindigkeit und Richtung des Spielers mittels Tastaturabfragen aktualisiert:

'W' 'w' oder Pfeil nach oben: Beschleunigen

'S' 's' oder Pfeil nach unten: Rückwärts beschleunigen

'A' 'a' oder Pfeil nach links: Nach links drehen

'D' 'd' oder Pfeil nach rechts: Nach rechts drehen

' ' (Leertaste): Bremsen

Dann wird überprüft ob der Spieler noch HP übrig hat, sollte das nicht der Fall sein wird in der Konsole angezeigt, dass man verloren hat und das Spiel geschlossen.

Danach wird die Update-Methode des World-Objekts aufgerufen, die die Update-Methode jedes Objektes in der Welt aufruft. Beim Spieler wird in dieser Methode die Position mit Hilfe des Velocity-Vectors aktualisiert.

Jetzt wird die Kollision des Spielers berechnet. Dies geschieht mittels einer Spiegelung über die Normale die von dem Kollidierenden Objekt zum Spieler geht.

Zum Schluss wird noch die aktuelle Position des Spielers zum Server gesendet.

Netzwerkthread

Im Netzwerkthread wird auf Commands vom Server gewartet und reagiert. So werden die korrespondierenden Objekte erstellt, geupdatet und gelöscht.

Klassendiagramm:

