

Tdate – The day after the end

Gameplay:

Wie für einen Egoshooter typisch bewegt man sich bei Tdate aus der Egoperspektive durch einen Level. Hierzu benutzt man wasd zur Bewegung und die Maus um sich umzusehen. Mit der linken Maustaste kann man schießen und sofern man sie im Fadenkreuz hat Gegner eliminieren. Trifft man einen Gegner so blutet er (Partikelsystem) und nach einigen Treffern stirbt der Gegner und verschwindet. Sind alle herkömmlichen Gegner eliminiert erscheint im Level der gesuchte Endgegner. Es gilt diesen zu eliminieren um das Spiel zu gewinnen.

Das Spiel läuft in der derzeitigen Version nur auf sehr sehr leicht. Wer Probleme hat alle Gegner zu finden kann mit F3 in den Wireframe Modus schalten.

Effekte:

Wird ein Gegner getroffen findet ein Blutungseffekt statt. Hierzu wird ein Partikelsystem benutzt. Als Quelle für das how-to diene http://www.codeworx.org/opengl_tut19.php. Die Partikel werden mittel GL_TRIANGLE_STRIP als Quadrate gezeichnet. Damit das wie ein Strahl bzw. explosion aussieht werden ca. 1000 Quadrate gleichzeitig gerendert.

Mit jedem Gegner wird auch ein Partikel Objekt erzeugt. Dieses wird jedes mal wenn der Gegner getroffen wird für eine kurze zeit gerendert.

Weiters werfen alle Gegner einen Schatten auf die Spielwelt, relativ zur Lichtquelle. Dieser wurde mittels Stencil-Buffer implementiert. Als Vorlage diene unter anderem das Shadowdemo auf <http://www.ultimategameprogramming.com>. Da wir uns durch verschiedenste Tutorial und Theorieseiten gearbeitet haben kann keine genaue Angabe über alle Quelle gemacht werden, es sei nur gesagt wir haben uns sehr viel mit Schatten beschäftigt. Nach Berechnung der Shadowmatrix werden die Schatten in 3 Schritten erzeugt, in denen die gesamte Szene 2x gerendert wird. Die Gegner welche die Schatten werfen ein weiteres mal gerendert.

Animierte Objekte:

Animierte Objekte sind in Tdate die Gegner. Während sie sich auf Patrouille befinden wird ein Laufvorgang animiert. Hat ein Gegner ausreichend Treffer kassiert äußert sich das in einer Sterbeanimation. Die Animation wird mittels key-Frames erzeugt. Dabei wird die Bewegung(bzw. Vertex Koordinaten) zwischen zwei key-Frames linear interpoliert. Während der Laufzeit wird jedes Mal ein neuer Vertex-Array erzeugt und mittel Vertex-Arrays gerendert. Dieses wirkt sich vor allem im Display-List modus auf die Framerate aus da jedes mal eine neue Display List erzeugt werden muss.

Experimentieren mit OpenGL / Debugausgaben:

Im Zuge des Punktes Experimentieren mit OpenGL wurden die vorgegebenen Funktionen implementiert. Mit F1 lässt sich die Hilfe ein und ausblenden, diese beinhaltet eine kurze Beschreibung der Aufgabe des Spielers sowie Hinweise zur Steuerung. Mit F2 lässt sich die Framerate ein und ausblenden und mit F3 kann man in den Wireframe Modus schalten. Mit F4 und F5 ist es möglich die Texturqualitäten zu verändern und Mipmapping zu deaktivieren (standardmäßig eingeschaltet). Mit F6 und F7 kann man zwischen den Rendermodi wie Vertex Arrays und Displaylists umschalten, hier kann man deutlich unterschiede in der Framerate sehen. Weitere Veränderungen der Framerate sind durch drücken von F8 zu sehen, was Frustum Culling ein und ausschaltet. Mit F9 kann man die Transparenz ein und schalten.

Steuerung:

Die Steuerung erfolgt wie schon erwähnt über die wasd-Tasten und die Maus. Hier ein Auszug aus der Hilfe:

Vorwärts gehen – w

Rückwärts gehen – s

Links strafen – a
Rechts strafen – d
Schießen – LMB

Loader/Models/Libraries :

Tdate verfügt über einen Worldloader welcher die Leveldaten (Beschaffenheit, Gegnerpositionen) aus einer txt Datei lädt. Weiters wurde ein Modelloader verwendet, Quelle:

<http://www.cs.jcu.edu.au/Subjects/cp2060/2001/lectures/lect183DModels.html>

Es handelt sich um einen Loader für das md2 Format, folglich sind unsere Models auch md2-Models. Der Loader wird nur zum Laden von MD2 Files und der zugehörigen Textur verwendet. Rendern und animieren der Modells wurde allerdings von uns gemacht. Als zusätzliche Library wurde GLUT benutzt.