

# F U S I O N

Wolf-Dieter Vogl	0626355	532	w.d.v@gmx.at
Roman Hochstöger	0627154	532	cg2@hochi.at

Das Spiel Fusion ist ein einfaches Echtzeit-Aufbau-Strategiespiel, bei dem es darum geht die eigene Basis vor den Angreifern zu verteidigen.

## GAMEPLAY

### **Tower Defense:**

Das Hauptgebäude (ein Fusionsreaktor) muss mit taktischem platzieren von Verteidigungstürmen vor den angreifenden Panzern verteidigt werden. Dabei steht aber nur eine begrenzte Anzahl an Energie zur Verfügung.

Es gibt 2 Arten von Türmen, die man bauen kann, den Teslaturm und den Laserturm. Ein Teslaturm hat eine höhere Schussfrequenz, dieser hat aber eine wesentlich geringere Reichweite. Bei den Angreifern gibt es auch 2 verschiedenen Panzertypen mit unterschiedlichen Reichweiten.

Beschädigte Gebäude und Türme können auch repariert werden, wobei hier auch wieder für die Reparatur Energie benötigt wird.

Derzeit gibt es 2 Level, wobei man das nächste Level erreicht, wenn man alle Panzer zerstört hat (oder mit der Cheat-Taste <F12>)

### **Energie:**

- Ein Teslaturm benötigt 20 Energiepunkte, ein Laserturm 60
- Ein Schuss kostet 1 Energie.
- Wenn zu wenig Energie vorhanden ist kann kein Turm gebaut werden bzw kein Schuss abgegeben werden.
- Die Energie erhöht sich pro Sekunde um eins und kann maximal 100 betragen.

**TASTENBELEGUNG:****Gameplay:**

<i>B:</i>	Tesla Turm bauen.
<i>N:</i>	Laser Turm bauen.
<i>R :</i>	Gebäude oder Turm reparieren.
<i>X:</i>	Turm entfernen, bzw. bauen abbrechen.
<i>Linke Maustaste:</i>	Turm platzieren, oder Gebäude bzw. Panzer auswählen. Wenn ein Objekt ausgewählt wurde, werden darüber die Lebenspunkte angezeigt.

**Kamera:**

<i>WASD o. Cursortasten:</i>	Schnelle Bewegung über Karte.
<i>Mauscursor am Bildschirmrand:</i>	Langsame Bewegung über Karte.
<i>Rechte Maustaste:</i>	Rotation der Kamera.
<i>Scroll-Rad:</i>	Höhe der Kamera über der Karte ändern.

**Andere Tasten:**

<i>Pause:</i>	pausiert das Spiel.
<i>ALT+ENTER:</i>	Fullscreen.
<i>F1:</i>	Hilfe .
<i>F2:</i>	FPS.
<i>F3:</i>	Wireframe.
<i>F4:</i>	Texturfiltermodus (Nearest Neighbour / Bilinear)
<i>F5:</i>	Mipmap Filtermodus (Aus, Nearest Neighbour, Linear)
<i>F6:</i>	Umschalten zwischen Intermediate, Vertex Arrays und Vertex Buffer Object Modus
<i>F8:</i>	Frustum Culling ein/aus
<i>F9:</i>	Transparenz aus/ein
<i>F10:</i>	Physx Debug Renderer & Bounding Box.
<i>F11:</i>	Anisotropes Filtern (aus, 4x, 8x, 12x, 16x).
<i>F12:</i>	<i>Cheat-Taste</i> - Level wechseln (oder Spielende bei letztem Level)

**OBJEKTE****Panzer:**

Hier. Animation	Der Panzerturm richtet sich in Richtung eines Turmes bzw. des Hauptgebäudes aus bevor er schießt
Bewegung	Der Panzer bewegt sich in Richtung Hauptgebäude
Texturierung + Material	Der Panzer ist vollständig texturiert
Beleuchtung	Der Panzer wird von einer Lichtquelle beleuchtet (Sonne)

**Turm:**

Texturierung + Material	Der Turm ist mit verschiedenen Materialien ausgestattet, teilweise semi-transparent, keine Textur
Beleuchtung	Der Turm wird von einer Lichtquelle beleuchtet (Sonne)

**Hauptgebäude:**

Hier. Animation	Rotation des Stellators
Texturierung + Material	Das Hauptgebäude ist mit verschiedenen Materialien ausgestattet, teilweise transparent, keine Textur
Beleuchtung	Das Hauptgebäude wird von einer Lichtquelle beleuchtet (Sonne)

**Lebensenergieanzeige:**

Texturierung + Material	Die Lebensenergieanzeige besteht aus einem transparenten Material, keine Textur
Beleuchtung	Die Lebensenergieanzeige wird von zwei Lichtquellen beleuchtet (Sonne + eine stationäre Lichtquelle)

**Terrain:**

Texturierung + Material	Das Terrain verwendet 2 überlagerte Texturen, eines für die Farben und eines für die Feinstruktur, Materialien werden keine verwendet. Die Schatten sind in der Textur inkludiert und nicht extra berechnet
-------------------------	---

**Wasser:**

Texturierung + Material	Das Wasser besteht aus 2 überlagerten Texturen, wobei bei einer die Texturkoordinaten verschoben werden, um eine Wasserbewegung zu simulieren. Weiters ist das Wasser teilweise transparent, um die Spiegelung sichtbar zu machen.
-------------------------	--

**MODEL-LOADER**

Der Loader/Paser wurde selbst geschrieben, ein Model wird aus einem obj File und optionalem Material mtl File geladen.

Die Model-Objekte sind in einzelne Bauteile unterteilt, für jedes Bauteil wird ein Vertex Buffer Object angelegt(sofern unterstützt). Das Zeichnen der Bauteile erfolgt mittels glDrawArray.

Jedes Objekt hat mindestens eine Materialeigenschaft sowie Transparenz, Normalvektoren und optional eine Textur.

Die Panzer Objekte sind aus dem Internet. Türme und Gebäude wurden aber selber in Maya modelliert. Die Landschaft wurde in Terragen erzeugt.

## EFFEKTE

- **Planare Spiegelungen:** Die Spiegelungen auf dem Wasser werden erzeugt indem eine planare Spiegelung des Terrains mittels eines Stencile Buffers gerendert wird. <http://www.opengl.org/resources/code/samples/advanced/advanced97/notes/node90.html>
- **Partikelsystem:** Klassisches Partikelsystem bei Explosionen, wenn ein Gebäude oder ein Panzer zerstört werden. <http://nehe.gamedev.net/>
- **Lens Flares:** Wenn die Sonne sich im Blickfeld befindet, sieht man Lens Flare Reflektionen. Diese sind jedoch nicht zu sehen, wenn die Sonne nicht im Frustum ist, oder durch ein Objekt verdeckt wird. <http://nehe.gamedev.net/>

## LIBRARIES

### Empfohlene Libraries:

PhysX, FmodEX, FreeImage, Glew, Imath, PhysX, SDL

### Sonstige Libraries:

- *Font-Manager:* <http://osdl.sourceforge.net/main/documentation/rendering/SDL-openGL-examples.html#ttf>
- *Log4cxx:* <http://logging.apache.org/log4cxx/index.html>
- *Memory Leak and Exception Trace:* <http://www.codeproject.com/KB/applications/leakfinder.aspx>
- *CEGui:* [www.cegui.org.uk](http://www.cegui.org.uk)