

Rendering virtueller Welten mit Shadern

Fachgebiet: Informatik
Betreuer: Urs Künzler
Experte: Peter Matti

Mermelin ist der Name des Spieles, welches das Resultat dieser Bachelor-Arbeit repräsentiert. Es handelt sich um ein Geschicklichkeits- und Rätsel-Spiel. Der Spieler steuert eine Kugel und wendet die vier Elemente der Alchemie auf diese Kugel an, um damit Hindernisse zu überwinden. Die Levels sind Labyrinth-ähnlich aufgebaut, wobei das Ziel darin besteht einen bestimmten Punkt zu erreichen. Das Labyrinth kann aus beliebig vielen Stockwerken bestehen.

Game-Engine

Für die Erstellung dieser Arbeit wurde eine Spieleengine in C++ entwickelt. Diese wurde durch die Verwendung von einzelnen Modulen realisiert, welche jeweils auf freie und verbreitete C++ Bibliotheken aufbauen:

- **Grafik:** OGRE (<http://www.ogre3d.org/>)
- **Physik:** Bullet (<http://bulletphysics.org/>)
- **Input:** OIS (<http://www.wreckedgames.com/>)
- **GUI:** Rocket (<http://librocket.com/>)
- **Audio:** SMFL (<http://www.sfml-dev.org/>)

OGRE nimmt dabei einen besonderen Platz ein, da diese Library weitere Aufgaben der Engine, wie das Laden von Ressourcen (Audio, 3D-Meshes, etc.) oder die Zeitgebung übernimmt. Im weiteren existiert ein Logik-Modul welches keine externe Library verwendet.

Multiplattform

Alle verwendeten Libraries können auf verschiedenen Plattformen genutzt werden. Das Spiel wurde sowohl auf Linux, als auch auf Windows entwickelt und ist auch auf beiden Betriebssystemen lauffähig. Zur Erstellung von entsprechenden Makefiles und Solutions wurde Premake (<http://industriousone.com/premake>) eingesetzt.



Mermelin unter Linux

Assets

Auch für die Erstellung der Assets wurde auf Open-Source gesetzt. Sämtliche Modelle wurden mit Blender (<http://www.blender.org/>) erstellt. Für die Bearbeitung von Grafiken wurde Gimp (<http://www.gimp.org/>) verwendet. Schliesslich wurden die Audio-Dateien (im .ogg Format) mit Audacity (<http://audacity.sourceforge.net/>) bearbeitet.

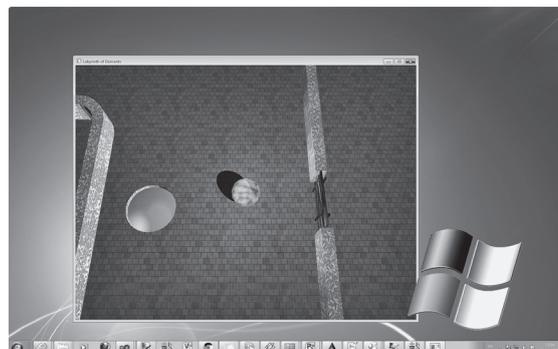
Shader-Entwicklung

Im Speziellen wird in dieser Arbeit demonstriert, wie Shader ein Spiel erweitern und verbessern können. Unter anderem wurden die vier Elemente mit Shadern realisiert. Es wurden Basic- und Intermediate-Versionen der Shader erstellt.

Shader sind Programme, welche direkt auf der Grafikkarte ausgeführt werden. Sie übernehmen Funktionen der Rendering-Pipeline. Sie wurden, in diesem Fall, in der Shader-Sprache GLSL (OpenGL Shading Language) verfasst.

Die folgenden Shader-Arten wurden für Mermelin verwendet:

- **Vertex-Shader:** Verarbeitet Eckpunkte
- **Fragment-Shader:** Verarbeitet Fragmente (Pixel)



Mermelin unter Windows



Eduardo Hahn Paredes
eddy.paredes@gmx.ch



Patrick Joos