

# Black Magic

by Philipp Gürtler, Tobias Fink

## Controls

- Kamerasteuerung
  - Die Kamera kann verschoben werden, indem man auf die Pfeiltasten drückt, oder mit der Maus an den Rand fährt.
  - Mit gedrückter mittlerer Maustaste + Mausbewegung wird die Kamera rotiert.
  - Mit dem Mousrad kann man rein- und rauszoomen.
  - Mit Pos1/Home wird die Kamera auf den Ursprung zurückgesetzt.
  - Mit Space wird die Kamera auf den Spieler zentriert (kann man auch halten).
- Spielersteuerung
  - Mit einem Rechtsklick auf das Spielfeld bewegt sich der Spieler dorthin.
  - Mit 'Q' schießt der Spieler einen Feuerball in die Richtung des Mauszeigers.
- Debug
  - F2 - Zeigt an, mit wievielen FPS das Spiel läuft, und wieviele Objekte gezeichnet werden.
  - F3 - Schaltet Wireframe-Modus ein/aus
  - F4 - Schaltet Textur-Sampling-Qualität um (Nearest Neighbor/Bilinear)
  - F5 - Schaltet Mip Mapping-Qualität um (Off/Nearest Neighbor/Linear)
  - F8 - Schaltet View-Frustum-Culling ein/aus
  - Alle obigen Tasten schreiben auch eine entsprechende Meldung in die Konsole.
  - G - Der Spieler wird in die Richtung des Mauszeigers gestoßen.
  - B - Schaltet Bounding-Boxes ein/aus (für Frustum-Culling, teilweise auch Collision)
  - K - Leitet den Boss-Kampf ein

## Implementation

### *Gameplay:*

Im Spiel geht es darum, die feindlichen Gegner mit Feuerbällen abzuschießen um zu überleben. Trifft man einen Gegner, wird er weggestoßen und verliert Lebenspunkte (HP). Kommen einem die Gegner zu nahe, wird man weggestoßen und verliert HP. Derzeit spawnen am Anfang einige Gegner, jedoch passiert noch nichts wenn man sie besiegt. Durch Drücken von 'K' erscheint ein Boss-Gegner, der aber noch in Arbeit ist. Außerdem gibt es es ein paar Models als statische Objekte, bei denen man Shadow-Mapping und Vertex-Skinning gut sieht.

Mit dem unebenen Untergrund gibt es auch 3D-Gameplay. Man kann auf kleinen Unebenheiten raufgehen oder auch vom Boden auf die Lava fallen, wo es einem dann die Gegner wieder schwer machen hinaufzukommen, da es nur bestimmte Stellen gibt, wo das möglich ist. Derzeit nimmt man noch keinen Schaden durchs stehen auf der Lava, das ist aber in Planung.

### *View-Frustum-Culling*

View-Frustum-Culling wurde mithilfe von Bounding-Boxen über jedes Objekt realisiert (B drücken zum anzeigen). Mit F2 sieht man, wie die Objekt-Anzahl geringer wird, wenn man sich mit der Kamera von der Szene entfernt.

### *Experimenting with OpenGL:*

Die genannten Features wurden eingebaut und können auch mit den entsprechenden Tasten ein/aus geschaltet werden.

## **Beleuchtung**

Die Scene wird generell durch ambient- und directional-light, sowie zusätzlich durch ein pointlight und ein spotlight beleuchtet. Das pointlight befindet sich im Bereich der Lava und die Intensität wurde so eingestellt, dass die Felsen schön vom Rot der Lava beleuchtet werden. Das spotlight befindet sich leicht nach links unten versetzt über der Mitte der Felsenplattform und füllt diese mit gelblichem Licht. Es wird auch nur dieses für die Schattenberechnung berücksichtigt.

## **Features**

Physikeffekte mittels bullet. Spriteanimationen für Spells.

## **Effekte**

- **Normal Mapping:** Die Tangentberechnungen wurden mit Hilfe von AssImp durchgeführt. Die normal map wurde auf Basis der selbst zusammengestellten Textur für die Map mit Hilfe von einem NVIDIA Plugin für Photoshop (<https://developer.nvidia.com/nvidia-texture-tools-adobe-photoshop>) erstellt und auf die übliche Weise in den Shader geladen. Ressourcen:  
[http://www.antongerdelan.net/opengl/normal\\_mapping.html](http://www.antongerdelan.net/opengl/normal_mapping.html),  
<http://fabiensanglard.net/bumpMapping/index.php>,  
[http://www.swiftless.com/tutorials/glsl/8\\_bump\\_mapping.html](http://www.swiftless.com/tutorials/glsl/8_bump_mapping.html)
- **Spotlights:** Ressourcen:  
<http://www.lighthouse3d.com/tutorials/glsl-tutorial/spot-light-per-pixel/>,  
<http://ogldev.atspace.co.uk/www/tutorial21/tutorial21.html>
- **Shadow Mapping:** Ressourcen:  
[https://lva.cg.tuwien.ac.at/cgue/wiki/lib/exe/fetch.php?media=students:cgue13\\_shadowmapping.pdf](https://lva.cg.tuwien.ac.at/cgue/wiki/lib/exe/fetch.php?media=students:cgue13_shadowmapping.pdf),  
<http://research.ncl.ac.uk/game/mastersdegree/graphicsforgames/shadowmapping/Tutorial%2014%20-%20Shadow%20Mapping.pdf>,  
<http://www.opengl-tutorial.org/intermediate-tutorials/tutorial-16-shadow-mapping/>
- **GPU Vertex skinning:** Aufgrund dessen, da die Models die wir verwenden, keine Struktur von Animationen haben, sondern nur über gewisse Framesequenzen definiert sind greifen wir mittels Zeitintervallen auf eine gewisse Animationssequenz zu.  
Ressourcen:

<http://ogldev.atspace.co.uk/www/tutorial38/tutorial38.html>

## **Models**

Unsere Models wurden zum Großteil von diversen Webseiten, die diese kostenlos zur Verfügung stellen genommen. Die Map wurde mit Blender von uns erstellt. Texturen wurden mit Photoshop erstellt.

## **Verwendete Tools:**

- Blender zum erstellen der Map

## **Zusätzliche Libraries**

Es wurden folgende externe Libraries verwendet:

- glm 0.9.4.3 (<http://glm.g-truc.net/>)
- assimp 3.0.1270 (<http://assimp.sourceforge.net/>)
- glfw 2.7.8 (<http://www.glfw.org/>)
- glew 1.9.0 (<http://glew.sourceforge.net/>)
- DevIL 1.7.8 (<http://openil.sourceforge.net/>)
- bullet 2.81 (<http://bulletphysics.org>)