

# Arcane Arts

Submission 2: Final Game

## Features

Das Spiel verfügt über **2 aktive Zauber** (Blitz und Feuer) in der rechten Hand, mit denen man den Monstern Schaden zufügen kann. Diese müssen mit Gedrückthalten der linken Maustaste aufgeladen werden und mit Loslassen werden sie abgeschossen. Zusätzlich haben wir einen **Stein** implementiert, den man werfen kann, um die **Advanced Physics** zu demonstrieren. Die Wurfstärke dieses Steines kann man durch längeres/kürzeres Gedrückthalten der linken Maustaste anpassen. Mit dem Stein kann man **Würfel/Items** von erhöhten Plattformen abwerfen, durch welche man nach dem Einsammeln einen **Health-Boost** bekommt. Zusätzlich zu den **Health-Würfeln** gibt es auch **Mana-Würfel** in der Welt, welche die Mana-Leiste auffüllen. Die Hauptfigur verfügt über eine **Mana- und Lebensanzeige**. In der linken Hand befinden sich **2 passive Zauber**, einer für Licht und einer für Heilungsmagie, die mit der rechten Maustaste aktiviert werden können. Die Aufgabe des Spiels ist es, alle Monster im Level ausfindig zu machen und zu besiegen. Am oberen rechten Bildschirmrand wird einem ebenfalls die Anzahl der im Spiel noch zu besiegenden Monster angezeigt. Hat man das geschafft, ist das Spiel vorbei. Verloren hat man, wenn die Lebensanzeige leer ist.

## Compulsory Gameplay

- ✓ 3D Geometry
- ✓ Playable
- ✓ Advanced Gameplay
- ✓ 60 FPS and Framerate Independence
- ✓ Intuitive Controls
- ✓ Intuitive Camera
- ✓ Illumination Model
- ✓ Textures
- ✓ Moving Objects
- ✓ Documentation
- ✓ Win/Lose Condition
- ✓ Adjustable Parameters (config.yaml)

## Optional Gameplay

- ✓ Collision Detection
- ✓ Advanced Physics
- ✓ Heads-Up Display

## Libraries

- [entt](#) (entity-component-system)
- [physx](#) (physics library)
- [irrklang](#) (sound library)
- [GLFW](#)
- [Glad](#)
- [glm](#) (mathematics library)
- [spdlog](#) (logging library)
- [stb](#) (image loader)
- [tinyobjloader](#) (obj loader)
- [yaml-cpp](#) (YAML parser)
- [ImGui](#) (GUI for debug/develop – **not for the HUD!**)

## Effects

### Lighting: Shadow Map with PCF

Für die Berechnung der Schatten, die vom Mond (Directional Light) ausgehend geworfen werden, verwenden wir eine Shadow Map. Zur Vermeidung von Aliasing setzen wir Percentage-Closer Filtering ein.

### Advanced Modelling: GPU Particle System using Transform Feedback

Das Partikelsystem wird eingesetzt, um zu zeigen, welcher aktive Zauber gerade ausgerüstet ist. Ist ein aktiver Zauber ausgerüstet, werden in der rechten Hand entsprechende Partikel erzeugt.

Zudem wird ein Lagerfeuer in die Welt gesetzt, welches ebenfalls mit Partikeln und einer Lichtquelle versehen ist.

### Texturing: Environment Map

In der Mitte der Spielwelt befindet sich eine reflektierende Ritterstatue. Die Reflexion auf der Oberfläche wird mittels Environment Mapping umgesetzt.

### Animation: Vertex Shader Animation

Die Vertex Shader Animation wird bei den Magiekugeln eingesetzt, um eine Art *Wobble* zu erzeugen und somit die Magiekugeln dynamischer oder *magischer* aussehen zu lassen.

### Shading: Simple Normal Mapping

Das Simple Normal Mapping wird zur realistischeren Darstellung der Umgebung (in unserem Fall des Terrains bzw. des Bodens) verwendet. Im Gegensatz zum *Simple* Normal Mapping haben wir zusätzlich die Transformation zum Tangentenraum implementiert.

### Post Processing: Bloom/Glow

Bloom/Glow verwenden wir in Verbindung mit HDR für die Magiekugeln und das Lagerfeuer.

## Links / Tutorials

### C++ / OpenGL

- [YouTube - The Cherno](#)

### Uniform Buffers

- [lighthouse3d.com](#)

### Textures

- [learnopengl.com](#)

### Shadow-Mapping

- [opengl-tutorial.org](#)
- [learnopengl.com](#)

### Particle System (using Transform Feedback)

- [ogldev.atSPACE.co.uk](#)

### Cubemaps / Skyboxes / Environment Mapping

- [learnopengl.com](#)
- [YouTube - Brian Will](#)

### Normal Mapping

- [learnopengl.com](#)

### HDR

- [learnopengl.com](#)

### Bloom

- [learnopengl.com](#)

## Controls

