# 3D Spiel

## Konzept

### Setting

## Technologie

### Plattform

Die Engine soll unabhängig von der unterliegenden Plattform laufen.

Daher wird Java als Programmiersprache und Laufzeitumgebung genutzt.

Als Grafikbibliothek wird JOGL benutzt, das auf OpenGL aufbaut,  
und als Bibliothek für mehrere Plattformen zur Verfügung steht.

Als Audiobibliothek wird OpenAL benutzt, das auch für mehrere Plattformen zur Verfügung steht.

Im Einzelnen werden für die Kopplung folgende Technologien verwendet:

|  |  |
| --- | --- |
| **Kategorie** | **Technologie** |
| Fenstersystem-Anbindung | JOGL NEWT |
| Maussteuerung | JOGL NEWT Events |
| Tastatursteuerung | JOGL NEWT Events |
| Joysticksteuerung | TBD |
| Gamepad Steuerung | TBD |
| Zeitsteuerung | Normale Java System-Timer Anbindung |
| Threadsteuerung | Normale Java Threadsteuerung |
| Datei-ein/ausgabe | Normale Java Ein/Ausgabemethoden |
| Netzwerk ein/ausgabe | Normale Java Ein/Ausgabemethoden |
| Grafikausgabe | JOGL OpenGL |
| Grafikausgabe (Shader) | OpenGL und OpenGL Shading Language |
| Audioausgabe | OpenAL |
| Remote Procedure Calls | Java interne RPC Methodik |

### Benutzeroberfläche (GUI)

Die GUI wird Plattformunabhängig als eigenes Modul erstellt,  
mit Schnittstellen zum Plattformsystem für Mauseingabe und Tastatureingabe, zum Ressourcensystem zum Laden von Ressourcen  
und zum Audio- und Grafiksystem zur visuellen and akustischen Ausgabe. Eine Aktualisierung der GUI erfolgt durch Ereignis-getriggerte Eingabe und zyklischem externem Aufruf einer Aktualisierungsroutine.

### 3D Szenenverwaltung

Die Szenenverwaltung wird unabhängig als eigenes Modul erstellt, mit Schnittstellen zum Plattformsystem für grafische und akustische Ausgabe, und zum Laden von Szenenressourcen. Eventuell wird eine spezielle Rendering-Ebene dazwischen gestellt. Eine Aktualisierung erfolgt durch zyklischen externen Aufruf einer Aktualisierungsroutine.

### Ressourcenverwaltung

Die Ressourcenverwaltung wird unabhängig als eigenes Modul entwickelt und dient zum Laden und Speichern von Ressourcen.

Internationalisierung und das Cachen und wiederverwenden von geladenen Ressourcen soll unterstützt werden.

### Partikelsystem

### Raumaufteilung

### Kollisionserkennung

### Physikberechnung

### Terrainerstellung

### Wegfindung und AI

### Debugging und Logging

### Spiellogik abbilden