

Bibliotheken, welche wir benutzen wollen

Niklas von Hirschfeld

2024-01-30

Contents

LWJGL	2
GLFW	2
OpenGL	2
OpenAL	3
GSON	3
Dear Im GUI	3
Java	4
io	4
nio	4
util	4

LWJGL

- Lizenz: BSD-3-Clause
- Link: <https://www.lwjgll.org/>

“Leight Weight Java Game Library” bietet Zugang zu nativen APIs, welche bei der Entwicklung von Grafik- (OpenGL, Vulkan), Audio- (OpenAL) und Parallel-Computing-Anwendungen (OpenCL) notwendig sind. Es besteht aus sogenannten “Bindings” für die jeweiligen Anwendungen. Die ursprünglichen APIs sind in diesem Fall nicht in Java geschrieben und werden so zu sagen übersetzt.

Der Quellcode ist unter der BSD-3-Clause Lizenz offen verfügbar und wird von einer Kollektive und Freiwilligen aktiv erweitert und aktualisiert.

Zu dem ist LWJGL modular aufgebaut. Man kann das benutzen, was man braucht. Bei der Entwicklung unseres Spieles brauchen wir folgende Module:

GLFW

- Lizenz: Zlib license
- Link: <https://www.glfw.org/>
- Entwickelt in: c

GLFW (Graphics Library Framework) ist eine Multi-Plattform Bibliothek für OpenGL, OpenGL ES und Vulkan. Sie erlaubt es über eine simple API Fenster, Kontexte, Oberflächen, Input und Events zu Managen. Über diese Bibliothek werden wir in erster Linie das Fenster an sich und den Input verarbeiten.

OpenGL

- Lizenz: Eigene, erlaubt es die Software zu verändern und zu verteilen (Mit Bedingungen)
- Link: <https://www.khronos.org/opengl/>
- Entwickelt in: c

OpenGL® ist die am weitesten verbreitete 2D- und 3D-Grafik-API der Branche, die Tausende von Anwendungen auf einer Vielzahl von Computerplattformen ermöglicht. Sie ist unabhängig von Fenstersystemen und Betriebssystemen sowie netzwerktransparent. OpenGL ermöglicht es Entwicklern von Software für PC-, Workstation- und Supercomputing-Hardware, hochleistungsfähige und visuell ansprechende Grafiksoftwareanwendungen für Märkte wie CAD, Inhaltserstellung, Energie, Unterhaltung, Spieleentwicklung, Fertigung, Medizin und virtuelle Realität zu erstellen. OpenGL stellt alle Funktionen der neuesten Grafikkarte zur Verfügung.

Über OpenGL können wir direkt mit der GPU kommunizieren, Texturen speichern und mit eigenen Shadern beeinflussen, wie die Grafikkarte etwas verarbeitet. Gerade diese Shader, welche in einer eigenen Sprache geschrieben sind, erweitern die Möglichkeiten, auch enorm für Pixelgrafik. So können wir zum Beispiel dynamisches Licht gestalten und vieles mehr. Essenziell ist unter anderem die Möglichkeit “Vertex” Buffer direkt an die GPU zu senden und so optimal wie möglich die Oberfläche zu gestalten.

OpenGL wird unter anderem aktive weiterentwickelt von: Google, Nvidia, Apple, AMD, Huawei, Intel, ...

OpenAL

- Lizenz: GNU LIBRARY GENERAL PUBLIC LICENSE
- Link: Aktuellster Fork: <https://github.com/kcat/openal-soft>
- Entwickelt in: c++

OpenAL ist eine plattformübergreifende API für Audio, die hauptsächlich für die Wiedergabe von 3D-Sound und räumlichen Audioeffekten verwendet wird. Es ermöglicht Spieleentwicklern, realistischen Klang in ihren Spielen zu erzeugen, indem sie Klangeffekte basierend auf der Position und Bewegung von Objekten im Spielumfeld platzieren können. OpenAL bietet eine einfache Möglichkeit, komplexe Audioeffekte zu erzeugen und bietet Unterstützung für verschiedene Audioformate und -geräte.

Wir werden uns in erster Linie darauf konzentrieren, dass wir überhaupt Sound haben und auch dafür ist OpenAL ideal.

GSON

- Entwickelt von: Google
- Lizenz: Apache-2.0 license
- Link: <https://github.com/google/gson>
- Entwickelt in: Java

Gson ist eine Java-Bibliothek, die verwendet werden kann, um Java-Objekte in ihre JSON-Darstellung zu konvertieren und wieder zurück. Dies ermöglicht es uns Level, Szenen, Entitäten und Items extern zu speichern und dynamisch zu Laden. Wir planen auch den Spielstand und viele andere Daten so zu Speichern und zugänglich zu machen. Da wir uns bemühen unser Spiel flexibel zu gestalten nimmt uns GSON sehr viel Arbeit und Mühe ab.

Dear Im GUI

- Lizenz: MIT license
- Link: <https://github.com/ocornut/imgui>
- Entwickelt in: C++

Dear ImGui ist eine leichte Bibliothek für grafische Benutzeroberflächen in C++. Sie gibt optimierte Vertex-Buffer aus, die jederzeit in die 3D-Pipeline-fähigen Anwendungen gerendert werden können. Sie ist schnell, portabel, Renderer-unabhängig und in sich geschlossen (keine externen Abhängigkeiten).

Diese Bibliothek ermöglicht es uns und auch später ihnen Level individuell zu gestalten und zu formen. Es ermöglicht uns während der Laufzeit Variablen zu verändern und so ein optimales Spielerlebnis für den Spielenden vorzubereiten.

Java

Dies sind Java eigenen Pakete die wir verwenden.

Weiter Informationen zu diesen Paketen finden Sie unter: <https://docs.oracle.com/en/java/javase/21/docs/api/index.html>

io

- File: um zu checken, ob ein Datei existiert
- IOException: Scheitern von Lese- und Schreiboperationen während der Laufzeit signalisieren

nio

Definiert Puffer, die Container für Daten sind.

- ByteBuffer
- FloatBuffer
- IntBuffer
- files.Files
- file.Paths

util

- ArrayList
- Collections
- HashMap
- List
- Map
- Vector