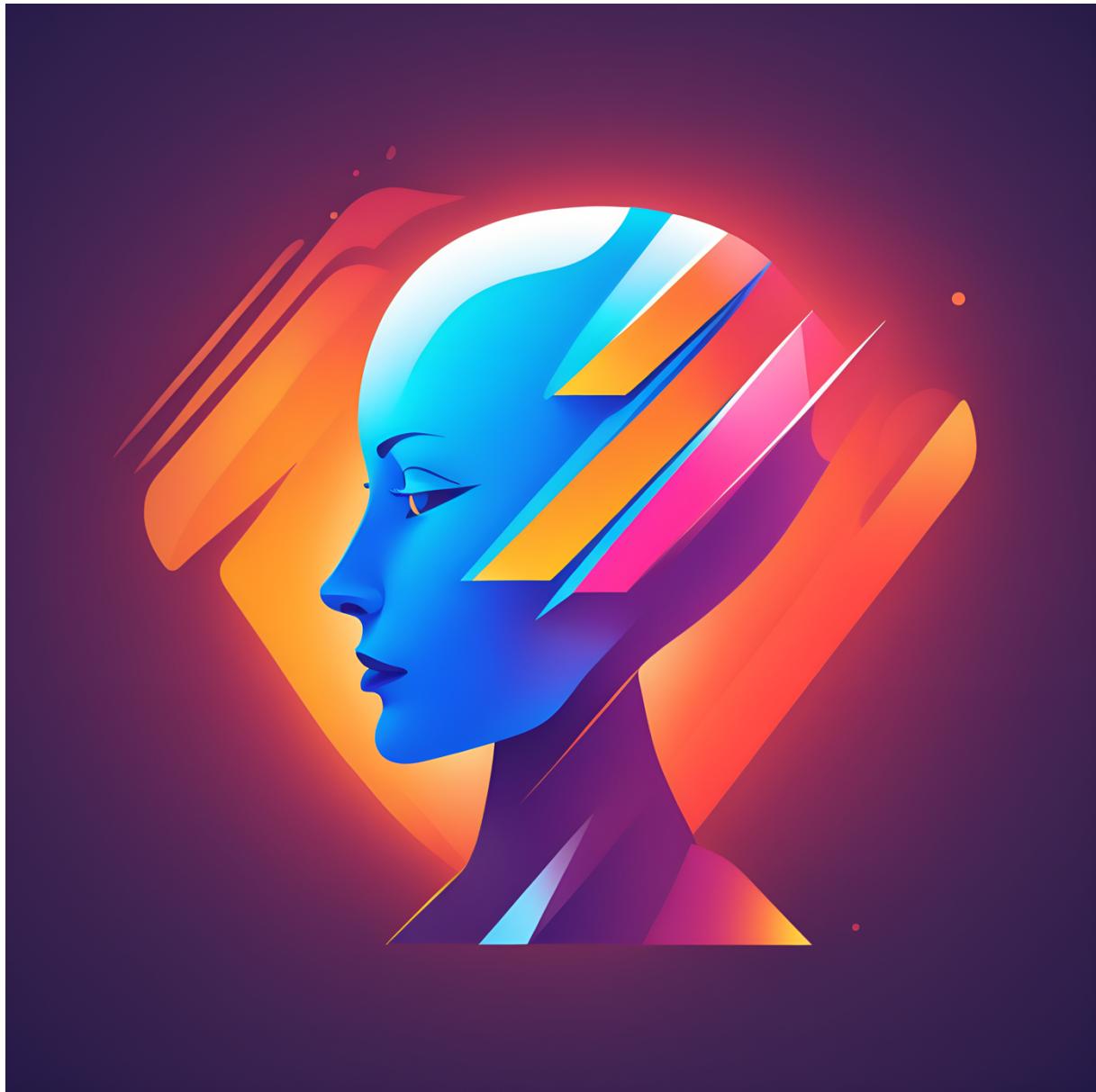


# **DNAIG Dokumentation**



## Installationsanleitung

### 1. Download der portablen Anwendung:

- Gehe auf unsere GitHub Seite:  
<https://github.com/LmaoPewPew/DNAIG/releases/latest>
- Lade das ZIP-Archiv "DNAIG-vx.x.x-windows-portable.zip" herunter

### 2. Entpacken des ZIP-Archivs:

- Navigiere zu dem Ordner, in den du das ZIP-Archiv heruntergeladen hast
- Wähle das ZIP-Archiv mit der rechten Maustaste aus und wähle "Entpacken"

### 3. Navigieren zum Installationsverzeichnis:

- Gehe in den gerade entpackten Ordner, der die Dateien der portablen Anwendung enthält

### 4. Anwendung starten:

- Wähle die Datei "DNAIG.exe" und führe sie aus

### 5. Anwendung schließen:

- Schließe die Anwendung, wenn du sie nicht mehr benötigst

## Hinweise:

Portable Anwendungen benötigen in der Regel keine formelle Installation, und das Deinstallieren ist einfach, indem du den Ordner löschst, den du zuvor erstellt hast. Wenn die Anwendung anpassbare Einstellungen oder Daten speichert, können diese im gleichen Ordner gespeichert werden, um sie portabel zu halten.

## **Bedienungsanleitung**

### **Rechte Leiste:**

#### **1. OBJ-Button:**

Klicke auf den "OBJ"-Button, um eine Objektdatei einzulesen. Wähle die gewünschte Datei aus, und sie wird in die Szene importiert.

#### **2. Kamera-Button:**

Der "Kamera"-Button erlaubt es dir, die Position der Kamera festzulegen. Nach einem Klick kannst du die gewünschte Kameraposition auswählen, um die Szene aus verschiedenen Blickwinkeln zu betrachten.

#### **3. Glühbirnen-Button:**

Der "Glühbirnen"-Button öffnet ein Menü zur Einstellung der Belichtung. Hier kannst du Helligkeit, Position und Lichtvariante für die Szene anpassen.

### **Obere Leiste:**

#### **1. File:**

- **Save As:**

- Durch Auswahl von "Save As" kannst du den Speicherpfad für dein Projekt festlegen und verschiedene Dateiformate auswählen, um deine Arbeit zu speichern.

- **Export:**

- Nutze die "Export"-Option, um das aktuelle Objekt als Bild zu exportieren.

- **Import:**

- Die "Import"-Funktion ermöglicht es dir, verschiedene Dateien in die Anwendung zu importieren.

#### **2. Edit:**

##### **Theme-Auswahl:**

- Unter "Edit" kannst du das Erscheinungsbild der Anwendung anpassen. Wähle zwischen Darkmode und Lightmode.

#### **3. Help:**

Hier findest du nützliche Produktinformationen und Hilfestellungen, die dir bei der Verwendung der Anwendung behilflich sein können.

## Untere Leiste:

### 1. Anzeige von Objekten und Einstellungen:

- In der unteren Leiste werden die ausgewählten Objekte sowie Kamera- und Lichteinstellungen angezeigt.

### 2. Render-Button:

- Klicke auf den "Render"-Button, um die aktuellen Einstellungen für das zu renderndes Objekt zu übernehmen und die Renderung zu starten.

## **Technische- und Architekturdokumentation:**

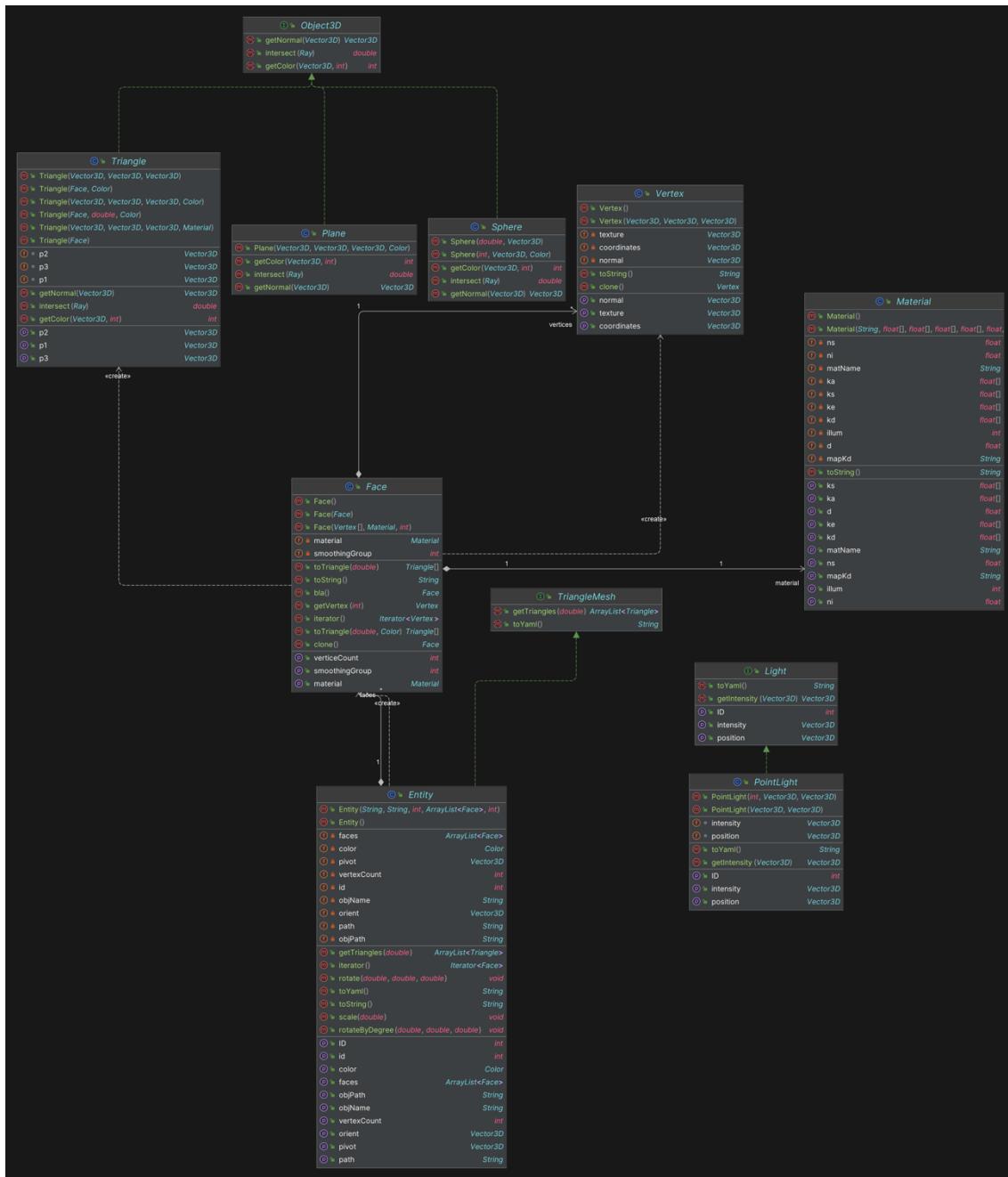
Das Raytracing Projekt ist modular aufgebaut und fördert die Trennung von Verantwortlichkeiten. Die Benutzeroberfläche (UI) in JavaFX ermöglicht die Interaktion, während die Kern-Rendering-Engine das Herzstück bildet und für Lichtberechnungen, 3D-Objektinteraktion und Bildgenerierung verantwortlich ist. Separate Module definieren die Struktur von 3D-Objekten, Einstellungen und Eigenschaften, wobei ein Vorschau-Fenster Echtzeitansichten der gerenderten Szene bietet.

Das Projekt beinhaltet die Entwicklung einer 3D-Szenenbeschreibungsumgebung die Darstellung von Objekten, Lichtquellen und Kameras ermöglicht. Die Anwendung soll verschiedene geometrische Formen rendern und Materialien wie diffuse, spiegelnde und transparente Oberflächen unterstützen, einschließlich der Anwendung von Texturen, Lichtmodelle und realistische Darstellungen von Schatten, Reflexionen und Lichtbrechung in transparenten Materialien. Die Implementierung umfasst auch Anti-Aliasing-Techniken für eine verbesserte Bildqualität und die Möglichkeit zur Bilddatei in verschiedenen Formaten wie PNG und JPEG. Die Programmierung erfolgt in Java, unter Verwendung geeigneter Bibliotheken für Vektor- und Matrizenoperationen. Die Implementierung zielt auf Effizienz ab, einschließlich Unterstützung für Multithreading zur Beschleunigung der Berechnungen. Die Teststrategie umfasst Unit-Tests, Integrationstests und Systemtests.

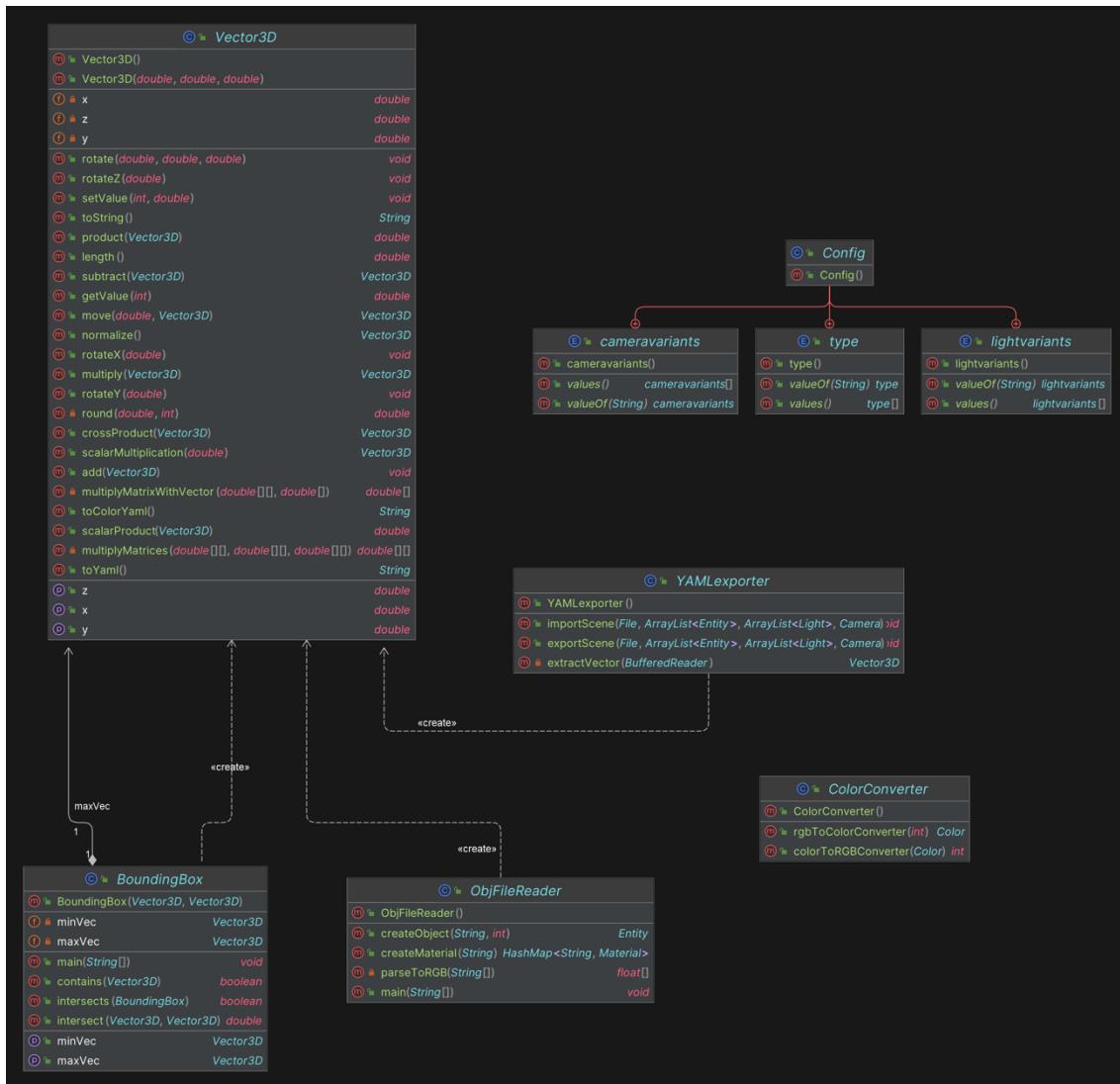
## **Gang Of Four Pattern, welche verwendet wurden:**

- Factory Method Pattern: Unser Objekt-Importer konstruiert, basierend auf den Input ein Entity Objekt aus verschiedenen Klassen wie Vertices, Faces, Material und Vektoren zusammen
- Observer Pattern: Das Preview-Fenster und die GUI verfolgen das Observer Pattern, da sich beide Objekte je nach Event anpassen (Löschen, hinzufügen von Objekten)
- Singleton: Für das Ausgabefenster des gerenderten Bildes, der CustomScene und der Kamera wird das Singleton verfolgt, da für all diese Objekte während der Laufzeit des Programmes nur eine Instanz existieren soll

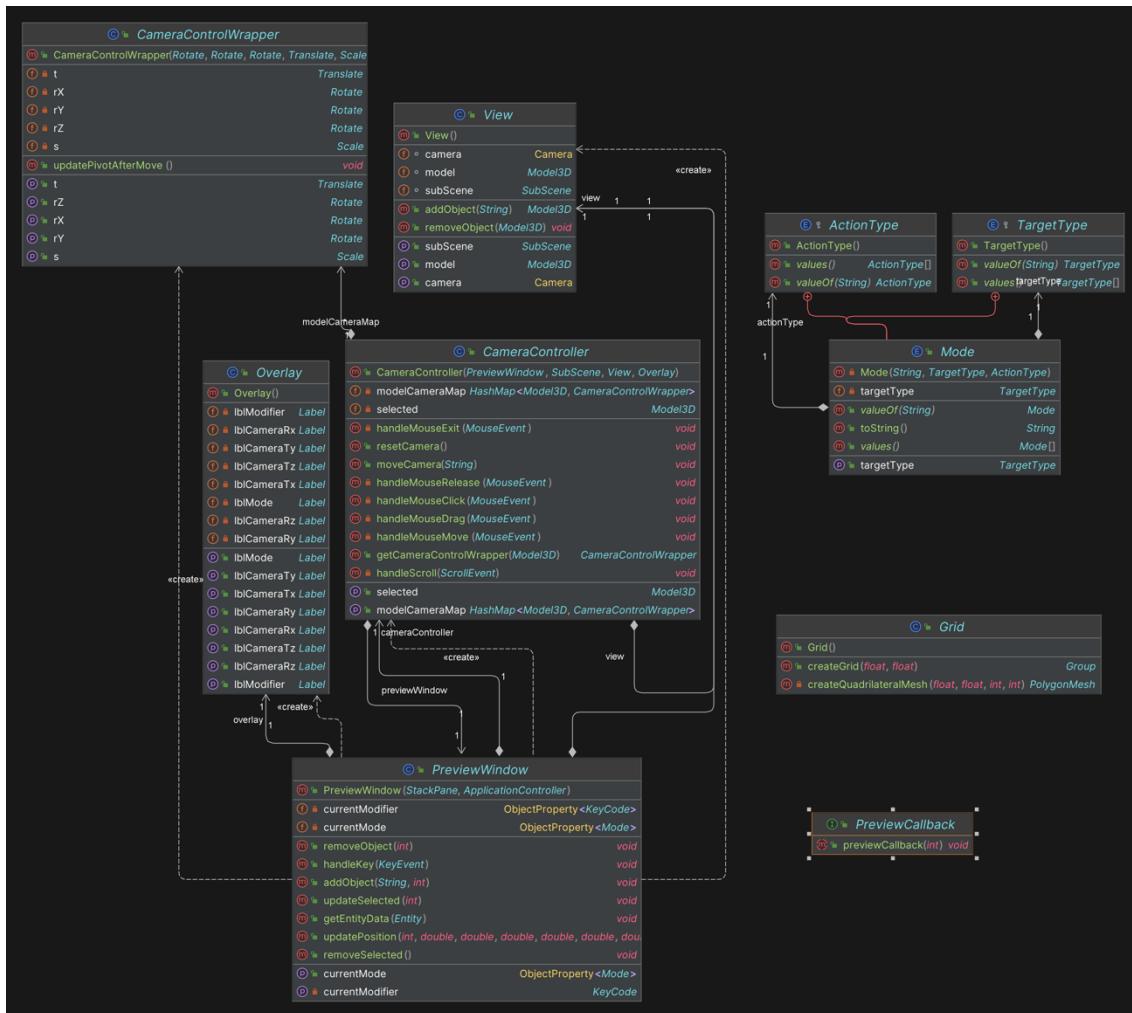
## Object Data (UML-Diagramm):



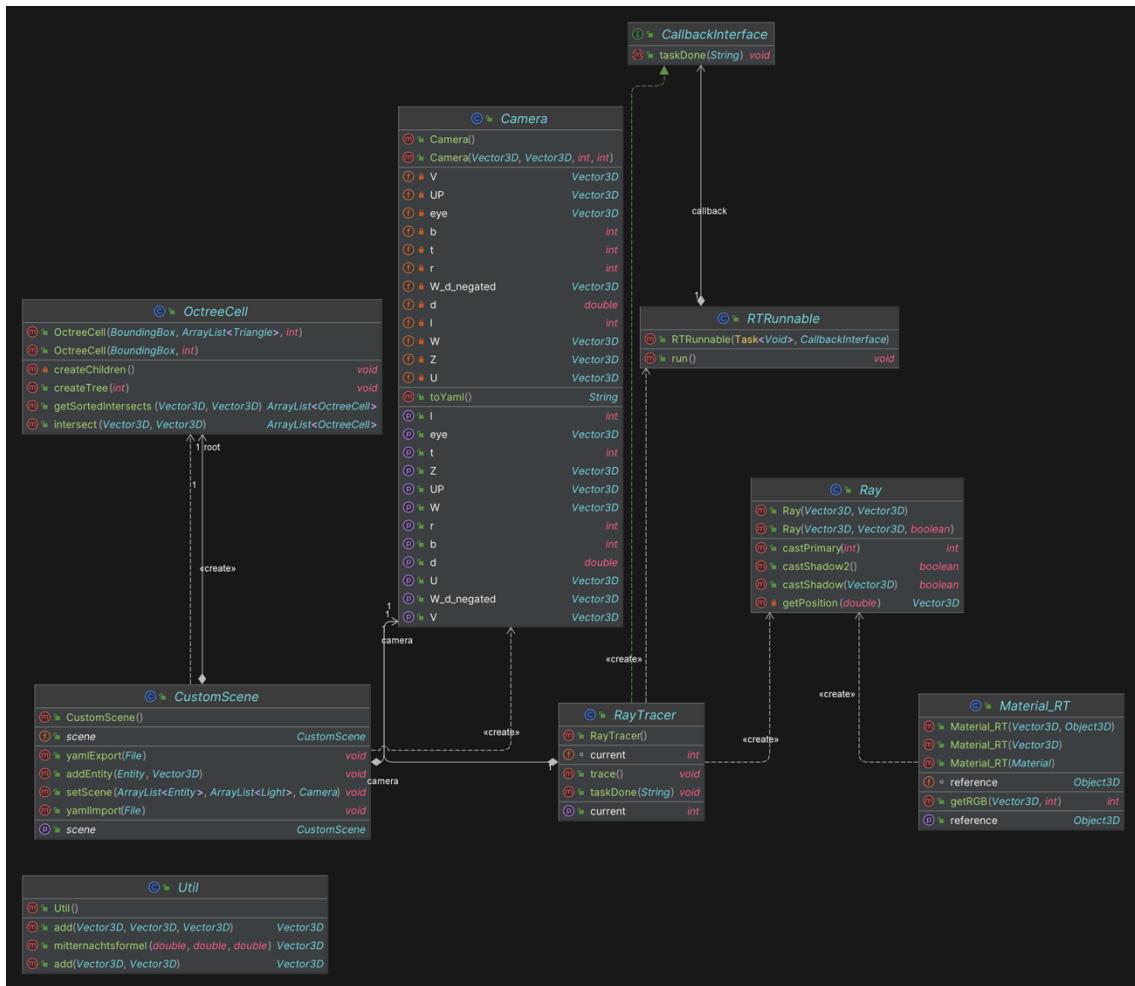
## Utils (UML-Diagramm):



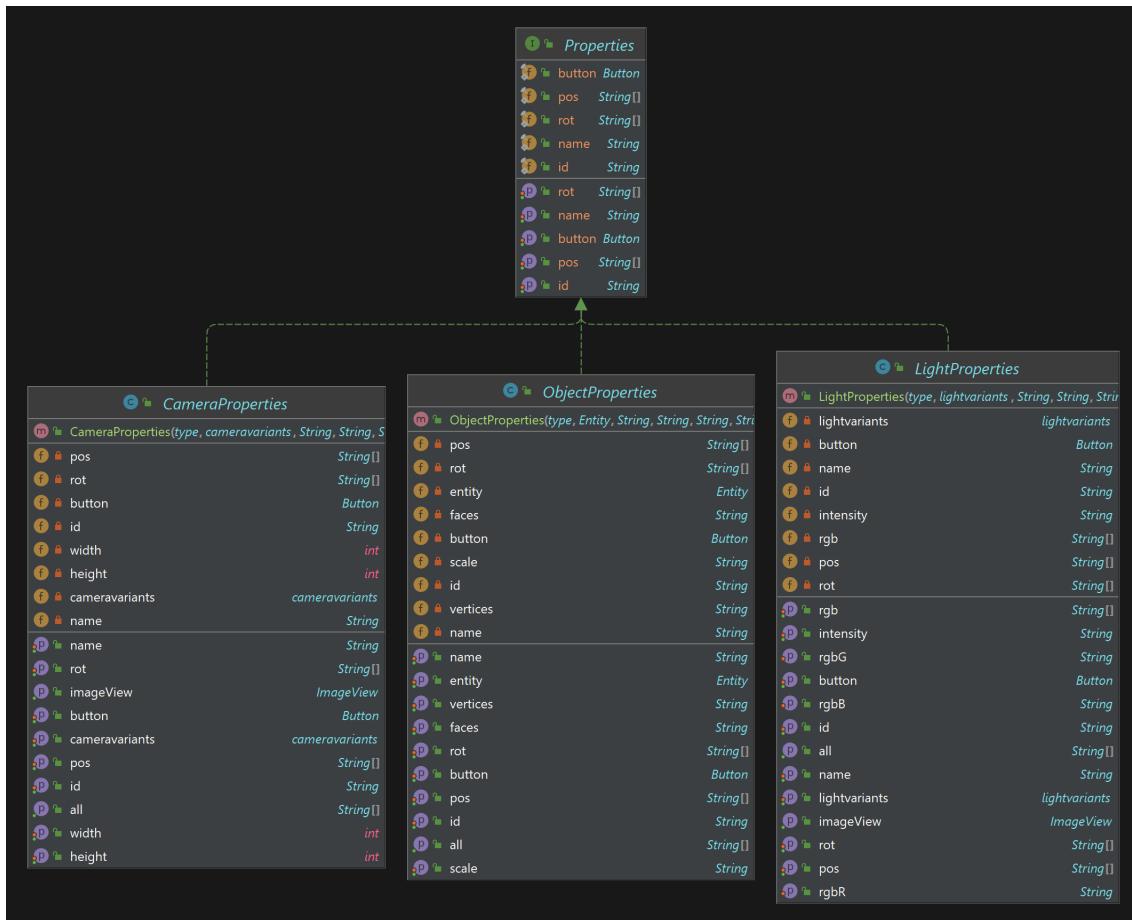
## Preview Renderer (UML-Diagramm):



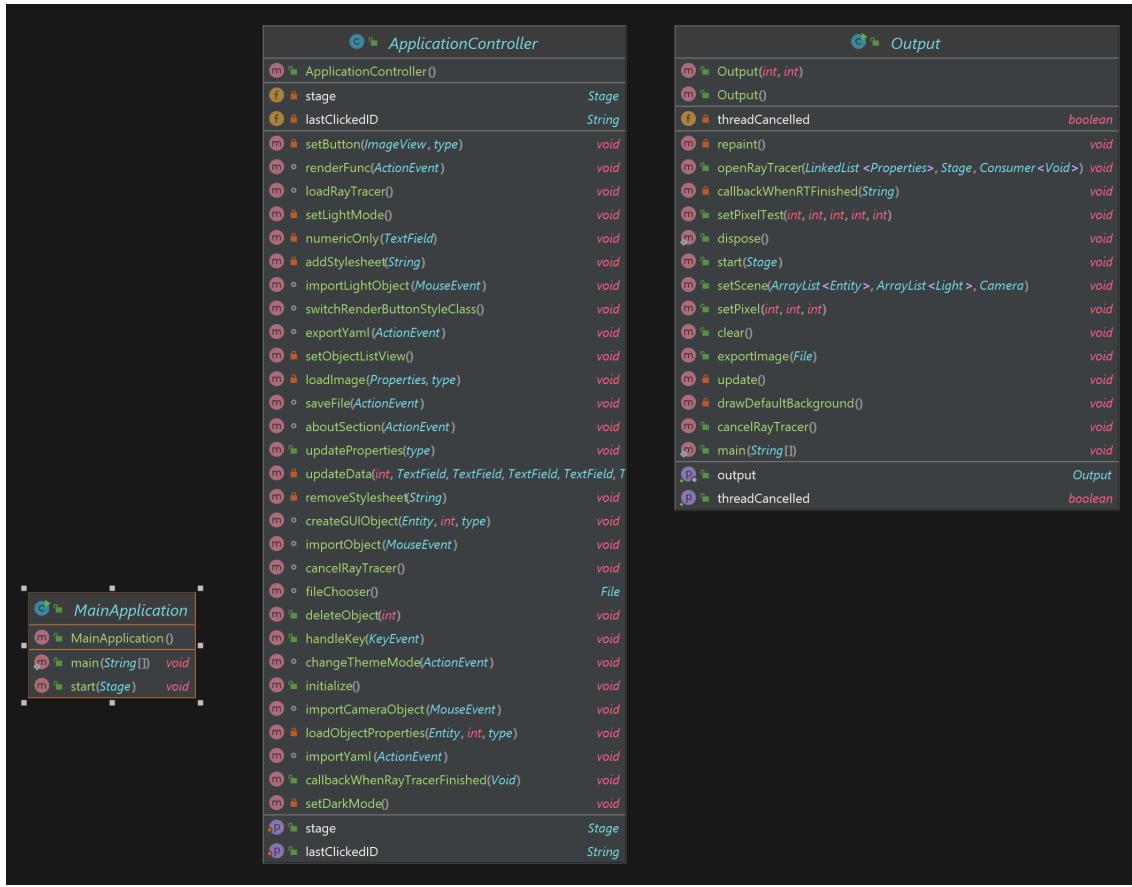
## RayTracer (UML-Diagramm):



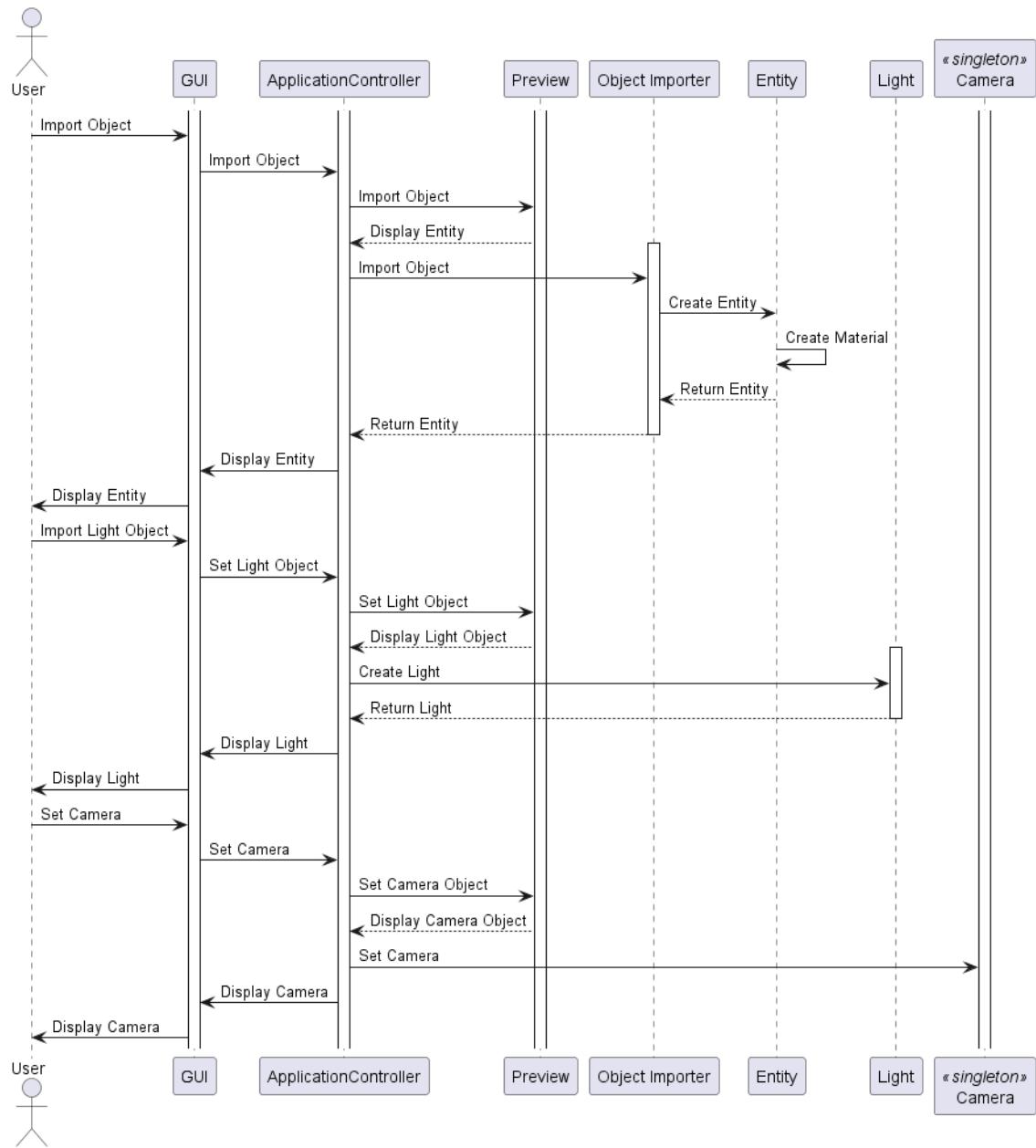
## Properties (UML-Diagramm):



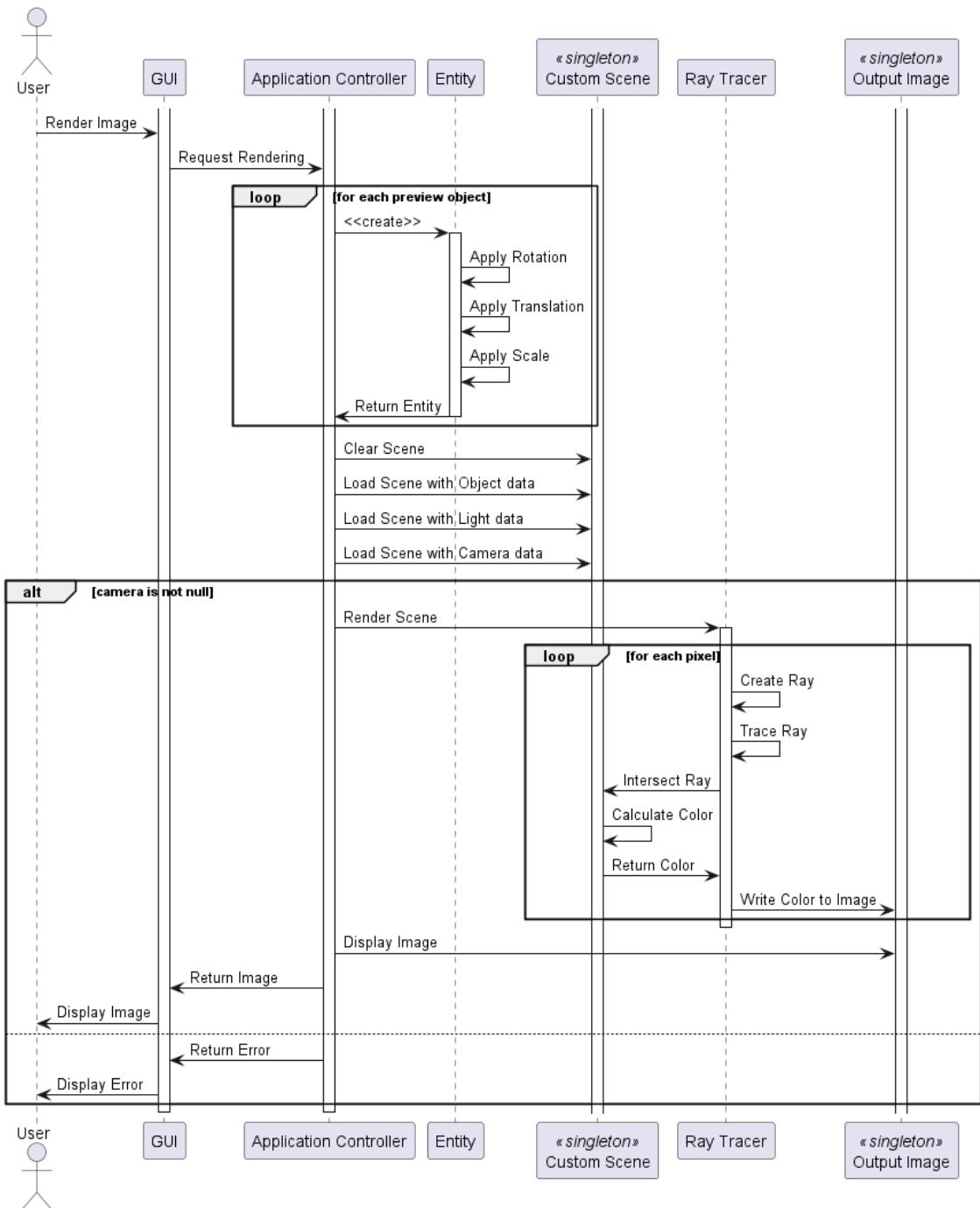
## Main Application (UML-Diagramm):



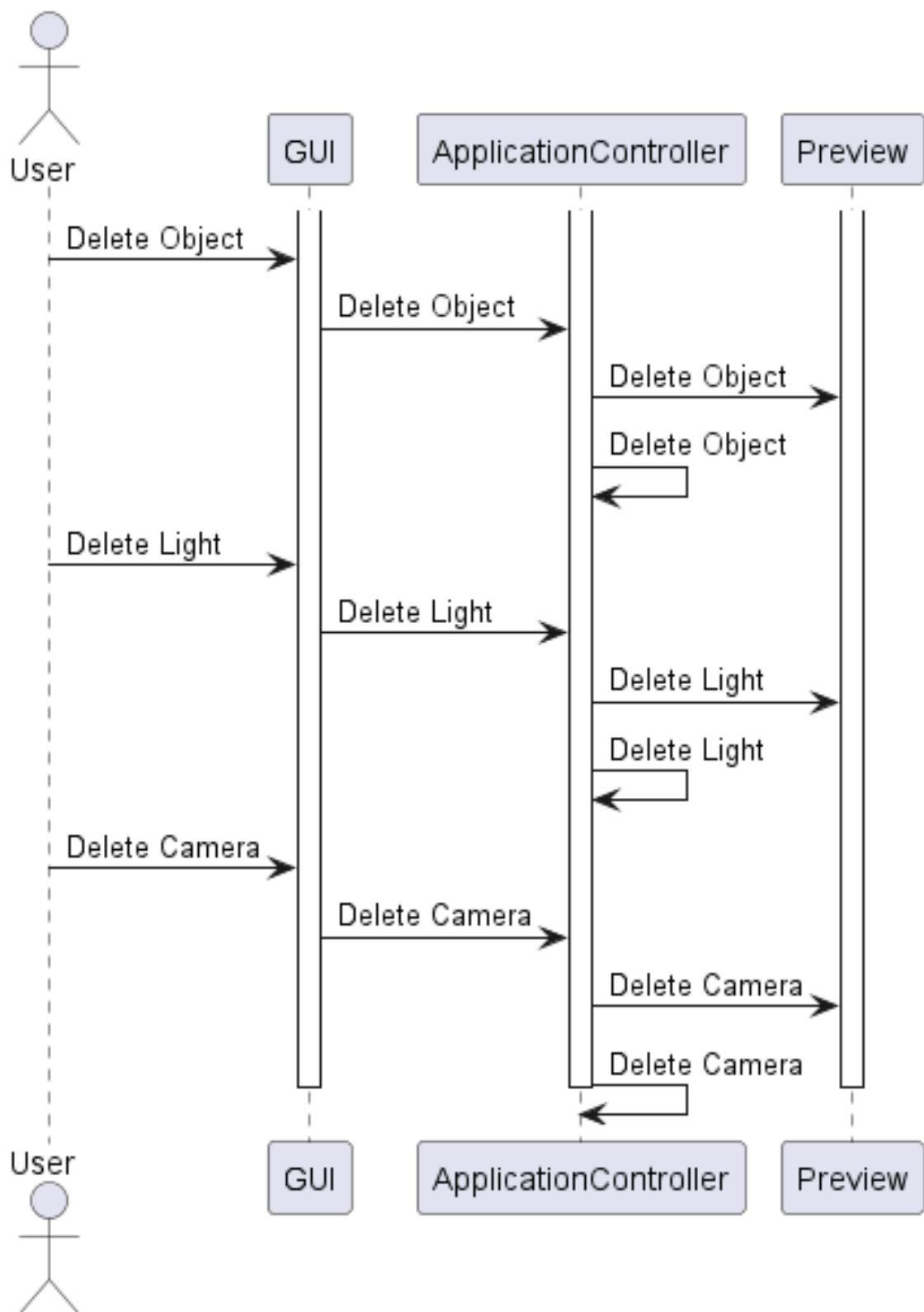
## Import (Sequenzdiagramm):



## Render (Sequenzdiagramm):



Lösch (Sequenzdiagramm):



## Quellennachweise

- Github
- Jira
- Discord
- IntelliJ
- [https://www.youtube.com/playlist?list=PLNmsVeXQZj7rvhHip\\_LTM-J-5wHw7dWnw](https://www.youtube.com/playlist?list=PLNmsVeXQZj7rvhHip_LTM-J-5wHw7dWnw)
- <https://projects.isp.uni-luebeck.de/isp/tutorial/-/blob/ab58f3d450d5e888ea48c540d3bc9cf435f69cd8/javafx/3d/src/main/java/main/Controller3D.java>
- <https://de.wikipedia.org/wiki/YAML>
- [https://en.wikipedia.org/wiki/Wavefront\\_.obj\\_file](https://en.wikipedia.org/wiki/Wavefront_.obj_file)
- <https://bertolami.com/files/octrees.pdf>

## Vorgehensbeschreibung:

Für die Durchführung des Projekts wird das agile Vorgehensmodell SCRUM angewendet. Das Team hatte wöchentliche Sprints durchgeführt, wobei jedes Sprint Review mit Herrn Lunde am Ende eines Sprints als Meilenstein dient. Aufgrund von zeitlichen Einschränkungen wird jedoch jede Woche ein Sprint Review abgehalten, anstatt am Ende eines Meilensteins. Die Kommunikation und Koordination erfolgt über verschiedene Hilfsmittel wie das Product Backlog, Sprint Backlog, Jira, Github, Discord und IntelliJ. Ein wöchentliches Scrum Meeting von etwa einer Stunde wird durchgeführt, um den Fortschritt zu besprechen und eventuelle Hindernisse, Besprechungen, Bugs etc. zu behandeln.

## Rollenverteilung:

Bamberger Dominik:	Entwickler
Kapisiz Yasin:	Entwickler
Miele Luca:	Scrum Master
Neziri Sahit:	Entwickler
Vogt Leon:	Entwickler
Winter Silas:	Entwickler
Yilmaz Alper:	Entwickler
Zero Inel:	Product Owner