



FIGURE 1

Software-Projekt Raytracer

Florian Bertscher, Christian Kasper, Lukas Jeckle,
Leon Musliu, Dennis Welsch

Team: RayForge

Winter Semester 2024/25

Technische Hochschule Ulm
Software-Projekt
Prof. Dr. R. Lunde

Table of Contents

1. Kontextanalyse	3
1.1 Einleitung.....	3
1.2 Motivation	3
1.3 Vision	3
1.4 Projektkontext.....	4
2. Anwendungsdomäne	5
3. Domänenmodell.....	6
4. Anforderungsdefinition.....	7
4.1 Funktionale Anforderungen	8
4.2 Nicht funktionale Anforderungen	10
5. Anwendungsfälle.....	12
6. Abläufe	12
7. Benutzerschnittstellen	12
8. Graphische Gestaltung und Nutzungskonzept.....	14
8.1 Hauptbildschirm.....	15
8.2 Einstellungen.....	20
9. Sources	24
9.1 Contents.....	24
9.2 Images	24

1. Kontextanalyse

1.1 Einleitung

In der modernen Computergrafik sind realistische Licht- und Schatteneffekte sowie Reflexionen von zentraler Bedeutung. Ziel dieses Softwareprojekts ist die Entwicklung eines leistungsfähigen Raytracers, der benutzerdefinierte 3D-Szenen verarbeitet und nahezu fotorealistische Bilder erzeugt. Durch die systematische Durchführung des Projekts können die Teammitglieder praktische Erfahrungen in allen Phasen der Softwareentwicklung sammeln. Dabei stehen die Vertiefung der Kenntnisse in Anforderungsmanagement, Design, Implementierung und Test im Vordergrund. Das Projekt fördert zudem Teamarbeit und Kommunikationsfähigkeiten in einem Team von 6-8 Personen. Dieses Lastenheft legt die Anforderungen fest, um die technischen und organisatorischen Ziele erfolgreich zu erreichen.

1.2 Motivation

Nach wochenlanger Planung und strategischer Überlegung ist es unser Ziel, in den Jahren 2024/25 einen leistungsfähigen Raytracer zu entwickeln. Dieses Projekt wird es uns ermöglichen, als Team zusammenzuarbeiten und dabei die individuellen Stärken jedes Mitglieds zu nutzen. Durch den Austausch von Wissen und Erfahrungen werden wir nicht nur unsere Fähigkeiten erweitern, sondern auch den Zusammenhalt im Team stärken. Unser gemeinsames Ziel ist es, einen Raytracer zu schaffen, der die Marktstandards übertrifft und einen echten Mehrwert für unsere Nutzer bietet. Der Fokus liegt dabei weniger auf dem individuellen Erfolg, sondern auf dem kollektiven Ergebnis und dem Spaß, den wir gemeinsam bei der Entwicklung haben.

1.3 Vision

Wie soll das fertige System ungefähr aussehen?

Das fertige System wird ein Raytracer sein, der in der Lage ist, nahezu fotorealistische Darstellungen von Dreiecksszenen zu erzeugen.

➔ Diffuses Raytracing

Welche Features soll es haben?

Die geplante Software für den Raytracer wird eine Vielzahl von leistungsfähigen Features umfassen. Zunächst wird sie die Möglichkeit bieten, Oberflächenmodelle einzulesen, die aus Dreiecken zusammengesetzt sind, und zwar im weit verbreiteten .obj/.mtl-Format, was eine einfache Integration von komplexen 3D-Objekten ermöglicht. Anwender können die Perspektive auf das modellierte Objekt frei wählen und verschiedene Ansichten und Kamerawinkel erkunden. Zudem haben sie die Freiheit, punktuelle Lichtquellen beliebig in der Szene zu platzieren, um realistische Beleuchtungseffekte und Schatten zu erzeugen. Eine integrierte 3D-Vorschau wird es den Benutzern ermöglichen, die Ansichten und Beleuchtungen in Echtzeit zu definieren und zu visualisieren, bevor die endgültige Bildberechnung erfolgt. Die Bildberechnung selbst erfolgt durch die Anwendung eines Raytracing-Algorithmus, der das Lichtverhalten in der Szene realistisch simuliert, einschließlich Reflexionen,

Brechungen und Schatten. Schließlich werden die gerenderten Bilder sowohl auf dem Monitor angezeigt als auch in gängigen Grafikformaten wie PNG oder JPEG gespeichert, sodass sie für die weitere Nutzung oder Bearbeitung bereitstehen. Diese Kombination aus Funktionen wird eine leistungsfähige und benutzerfreundliche Softwarelösung bieten, die den Anforderungen von Entwicklern und Grafikern gerecht wird.

In welchen Szenarien soll es eingesetzt werden?

Unternehmen können den Raytracer nutzen, um Produkte in einer realistischen Umgebung darzustellen, was bei Marketingmaterialien und Online-Shops hilfreich ist, um potenziellen Käufern ein besseres Verständnis der Produkte zu vermitteln.

1.4 Projektkontext

Wie passt das Projekt in die durchführende Organisation?

Ziel der Organisationsform für dieses Projekt ist es, zu erkennen und zu entscheiden, welche verschiedenen Rollen und Fachbereiche innerhalb eines Projektes existieren und zu planen, wie diese am effektivsten zusammenarbeiten können. Dabei treten uns folgende Fragen auf wie: Welches Mitglied aus dem Team kann welche Stärken und Kenntnisse in unser Projekt einbringen und welche Rollen/Charaktere können in unserem Projekt verteilt werden.

Welche Stakeholder sind daran beteiligt?

Man unterscheidet zwischen interne und externe Stakeholder. Hier bei unserem Raytracer sind die internen Stakeholder der Auftragsgeber (Eigentümer) und Mitarbeiter beteiligt. Die externen Stakeholder sind hierbei unsere Endnutzer, die unseren Raytracer nutzen.

Welche möglichen Weiterentwicklungen oder Folgeprojekte sind denkbar?

Eine sinnvolle Weiterentwicklung des Raytracers wäre die Integration von Physically Based Rendering (PBR), um Materialien und Lichtquellen physikalisch korrekter darzustellen und so die Realitätsnähe der gerenderten Szenen zu erhöhen. Zusätzlich könnte der Einsatz von Künstlicher Intelligenz dazu dienen, Renderprozesse zu optimieren und automatisch Licht- und Schatteneffekte anzupassen, was die Benutzerfreundlichkeit verbessert und Renderzeiten verkürzt.

2. Anwendungsdomäne

Vorlage:

BEGRIFF

BESCHREIBUNG

IST-EIN

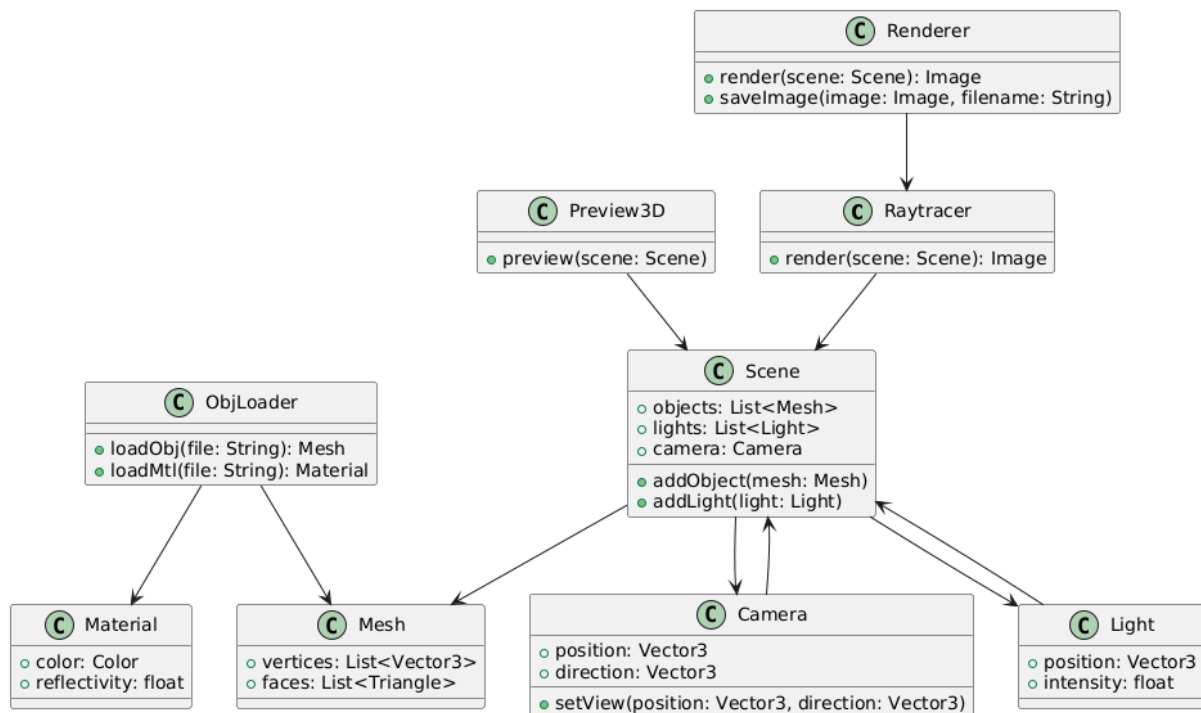
KANN-SEIN

ASPEKT

BEMERKUNG

BEISPIELE

3. Domänenmodell



4. Anforderungsdefinition

Vorlage:

AKTEUR	
BESCHREIBUNG	
ROLLE	

AKTEUR	AUFTRAGGEBER
BESCHREIBUNG	Institut für Softwaretechnik
ROLLE	Der Auftraggeber stellt die Anforderungen bereit, über die das Produkt am Ende verfügen muss. Nach der Entwicklung testet er das Resultat und entscheidet darüber, ob das fertige Produkt die gestellten Anforderungen erfüllt.

AKTEUR	DEVELOPER
BESCHREIBUNG	RayForge (wir)
ROLLE	Sind verantwortlich für das Erstellen des Produkts und müssen dabei auf die Wünsche und Anforderungen des Auftraggebers eingehen.

AKTEUR	CLIENT
BESCHREIBUNG	Die Software des Teilnehmers
ROLLE	Einlesen von aus Dreiecken zusammengesetzten Oberflächenmodellen aus Dateien im .obj/.mtl-Format.

AKTEUR	TEILNEHMER
BESCHREIBUNG	Endnutzer des Projekts
ROLLE	Personenkreis, der die Software nutzt, um photorealistische Darstellungen von Dreiecksszenen zu generieren.

AKTEUR	STACKEHOLDER
BESCHREIBUNG	Prof. Dr. R. Lunde
ROLLE	Stakeholder sind Teilhaber des Projekts. Sie haben außerdem berechtigtes Interesse am Verlauf und dem Ergebnis.

4.1 Funktionale Anforderungen

Vorlage:

ID	FA1
TITEL:	
BESCHREIBUNG	
BEGRÜNDUNG	
ABHÄNGIGKEITEN	

ID	FA1
TITEL:	Einlesen von .obj/.mtl-Dateien
BESCHREIBUNG	Der Raytracer muss in der Lage sein, Oberflächenmodelle, die aus Dreiecken bestehen, aus Dateien im .obj- und .mtl-Format einzulesen.
BEGRÜNDUNG	Dies ist notwendig, um geometrische Objekte für das Rendering bereitzustellen. Die .obj- und .mtl-Formate sind weit verbreitet und ermöglichen eine einfache Integration von 3D-Modellen.
ABHÄNGIGKEITEN	

ID	FA2
TITEL:	Freie Wahl der Sicht auf das Objekt
BESCHREIBUNG	Der Raytracer muss es ermöglichen, die Lage der Projektionsebene, den Bildausschnitt und das Projektionszentrum flexibel zu wählen. Zudem muss ein einheitliches Dateiformat für sichtrelevante Daten definiert werden.
BEGRÜNDUNG	Diese Anforderung ist wichtig, um eine benutzerdefinierte Ansicht des Modells zu ermöglichen. Eine gemeinsame Dateiformatspezifikation fördert die Zusammenarbeit zwischen den Gruppen und stellt sicher, dass alle Sichtdaten konsistent gespeichert werden.
ABHÄNGIGKEITEN	

ID	FA3
TITEL:	3D-Preview für Ansichts- und Beleuchtungsdefinition
BESCHREIBUNG	Der Raytracer muss eine Echtzeit-Visualisierung der darzustellenden Objekte, Kamera und Lichtquellen bieten. Die 3D-Preview soll eine schnelle Reaktion auf Änderungen der Parameter ermöglichen und eine detaillierte Darstellung der Objekte (einschließlich Texturen) unterstützen
BEGRÜNDUNG	Eine Echtzeit-Visualisierung ermöglicht es Benutzern, Anpassungen sofort zu sehen, was die Qualität der Ergebnisse verbessert. Die Möglichkeit, mit Lichtquellen und Kameraeinstellungen zu experimentieren, ist entscheidend für ein effektives Design.
ABHÄNGIGKEITEN	

ID	FA4
TITEL:	Bildberechnung durch Anwendung von Raytracing
BESCHREIBUNG	Der Raytracer muss die Bildberechnung durch Strahlverfolgung vom Projektionszentrum in Richtung der Projektionsebene realisieren.
BEGRÜNDUNG	Die Anwendung von Raytracing ermöglicht realistische Bilddarstellungen durch physikalisch basierte Lichtberechnung und effektive Simulation von Reflexion und Brechung. Dies ist entscheidend für die Qualität und Genauigkeit der erzeugten Bilder.
ABHÄNGIGKEITEN	

ID	FA5
TITEL:	Bilddarstellung auf dem Monitor sowie Speicherung in einem gängigen Grafikformat
BESCHREIBUNG	Der Raytracer muss in der Lage sein, die berechneten Bilder auf dem Monitor darzustellen und sie in einem gängigen Grafikformat (z. B. PNG, JPEG) zu speichern.
BEGRÜNDUNG	Diese Anforderung ist entscheidend, um die Benutzererfahrung zu verbessern und die Interoperabilität mit anderen Softwareanwendungen zu gewährleisten. Eine unabhängige Bilddarstellung und -speicherung ist wichtig, um Flexibilität und Zukunftssicherheit zu gewährleisten.
ABHÄNGIGKEITEN	

4.2 Nicht funktionale Anforderungen

Vorlage:

ID	QA1
TITEL:	
BESCHREIBUNG	
BEGRÜNDUNG	

ID	QA1
TITEL:	Robustheit/ Zuverlässigkeit
BESCHREIBUNG	Die Anwendung darf nicht abstürzen. Bei 100 Spielen darf maximal 1 Spiel aufgrund eines Fehlers abgebrochen werden.
BEGRÜNDUNG	Ein möglichst fehlerfreies Spiel ist für eine nutzerfreundliche Erfahrung zwingend notwendig.

ID	QA2
TITEL:	Effizienz der Bildberechnung
BESCHREIBUNG	Die Bildberechnung sollte möglichst schnell erfolgen, wobei der Speicherverbrauch bei der Umsetzung von Beschleunigungsstrategien nicht vernachlässigt werden darf.
BEGRÜNDUNG	Eine effiziente Bildberechnung ist entscheidend für die Benutzererfahrung und die Anwendbarkeit des Raytracers in realistischen Szenarien.

ID	QA3
TITEL:	Wartbarkeit der Software
BESCHREIBUNG	Die Software soll so gestaltet sein, dass sie von einem neuen Team zu einem späteren Zeitpunkt weiterentwickelt werden kann.
BEGRÜNDUNG	Eine wartbare Softwarearchitektur ist entscheidend, um die zukünftige Entwicklung und Erweiterung der Anwendung zu ermöglichen.

ID	QA4
TITEL:	Anwendungssprache, Implementierungssprache und Dokumentationssprache
BESCHREIBUNG	<ul style="list-style-type: none">• Die Benutzerschnittstelle der Anwendung (d.h. aller Komponenten) kann auf Deutsch oder Englisch gestaltet werden.• Die Implementierungssprache (Bezeichner im Source Code und Kommentare) soll jedoch Englisch sein.• Sonstige Dokumente wie Benutzerhandbuch, Testdokumentationen, Projekttagbuch, usw. können auf Deutsch oder Englisch verfasst werden.

ID	QA5
TITEL:	Programmiersprachen und Technologien
BESCHREIBUNG	Als Programmiersprache sollten Java, JavaScript, C# und C++ genutzt werden

ID	QA6
TITEL:	Plattformen
BESCHREIBUNG	Der Benutzer-Client und Editor müssen entweder auf einer aktuellen Linux-Distribution, aktuellen Windows Version oder basierend auf Webtechnologien, in einem aktuellen, standartkonformen Browser Laufen.
BEGRÜNDUNG	Ermöglichen Nutzern der jeweiligen Plattform(en) den Raytracer zu nutzen.

5. Anwendungsfälle

Text

6. Abläufe

Text

7. Benutzerschnittstellen

7.1 3D – Szene JSON

```
{
  "entities": {
    "cube": {
      "uuid": 1,
      "Translation": {
        "position": {
          "x": -5,
          "y": 0,
          "z": 20
        },
        "rotation": {
          "x": 0,
          "y": 90,
          "z": 25
        },
        "scale": {
          "x": 2.5,
          "y": 1,
          "z": 1
        }
      },
      "components": {
        "RenderComponent": {
          "objUUID": 4,
          "matUUID": 5
        }
      }
    },
    "sphere": {
      "uuid": 2,
      "Translation": {
        "position": {
          "x": 5
        }
      },
      "components": {
```

```

    "RenderComponent": {
      "objUUID": 4,
      "matUUID": 5
    }
  },
  "sun": {
    "uuid": 3,
    "Translation": {
      "position": {
        "y": 50
      }
    },
    "components": {
      "LightComponent": {
        "intensity": 0.15,
        "color": {
          "r": 255,
          "g": 255,
          "b": 0
        }
      }
    }
  },
  "resources": {
    "objects": {
      "4": {
        "type": "obj",
        "path": "some/relative/path/to/obj/file/cube"
      },
      "materials": {
        "5": {
          "type": "mat",
          "path": "some/relative/path/to/mat/file/wood"
        }
      }
    }
  }
}

```

In einer 3D-Szene beschreibt das für einen Raytracer verwendete JSON alle wesentlichen Szenenelemente und gliedert sie in zwei Hauptkomponenten: **Entities** und **Resources**.

Die Entities repräsentieren die verschiedenen Objekte in der Szene, wie z.B. 3D-Modelle oder Lichtquellen. Jedes dieser Objekte hat bestimmte Eigenschaften, die seine Platzierung und Ausrichtung im Raum bestimmen. Dazu gehören Position, Rotation und Skalierung, die jeweils die räumliche Anordnung des Objekts bestimmen und beeinflussen, wie es im endgültigen Bild erscheint. Darüber hinaus besitzen diese Objekte Komponenten, die die Darstellungs- und

Materialeigenschaften definieren. Beispielsweise verweist eine Render-Komponente auf ein bestimmtes Material oder eine bestimmte Geometrie, die dem Objekt zugeordnet ist, und Lichtquellen enthalten zusätzlich eine Licht-Komponente, die die Intensität und Farbe des Lichts bestimmt.

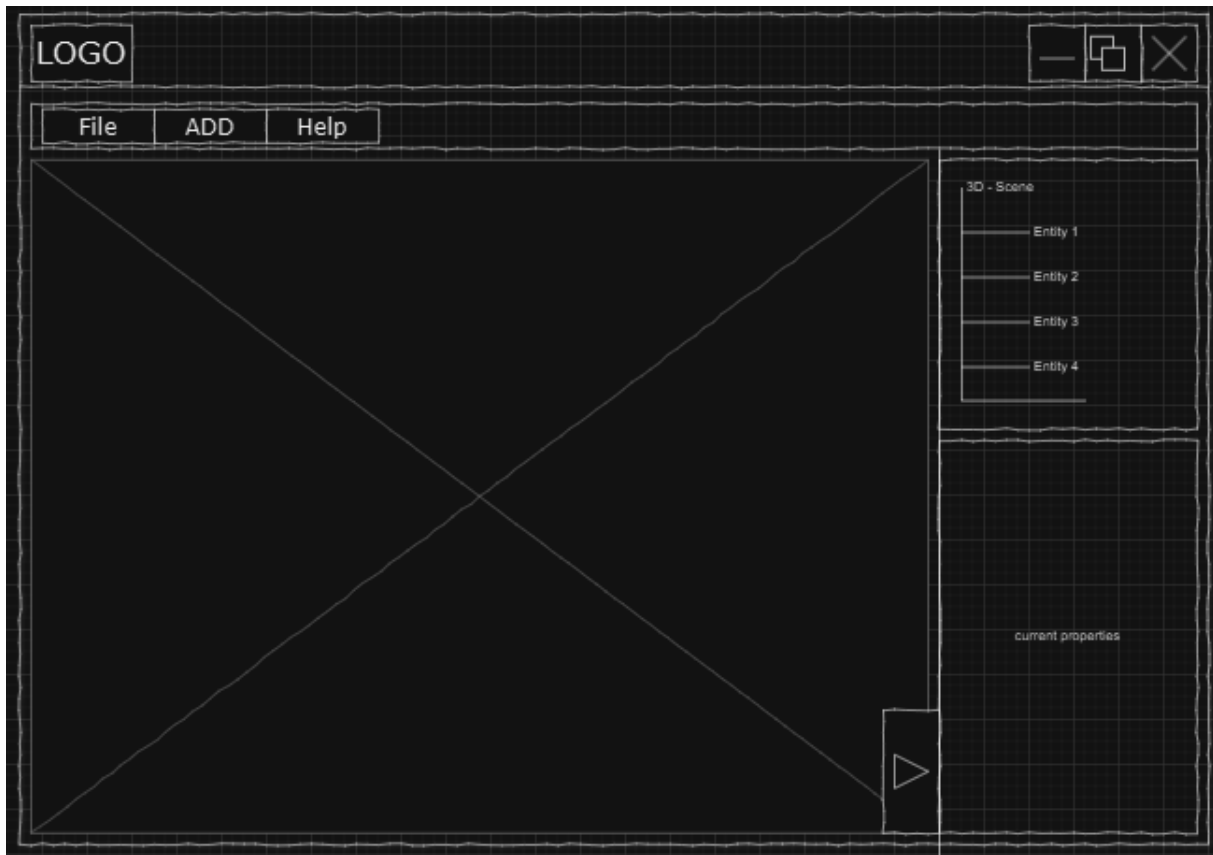
Die Ressourcen verweisen auf interne Dateien mit einem relativen Pfad, die für die Geometrie- und Materialdefinitionen der Objekte in der Szene benötigt werden. Zu den Ressourcen gehören Verweise auf Geometrien wie OBJ-Dateien, die das Modell eines Objekts beschreiben, und auf Materialien, die Texturen oder andere Oberflächeneigenschaften enthalten. Diese Dateien liefern wichtige Informationen über die physikalischen Eigenschaften der Oberflächen, damit der Raytracer das Aussehen der Objekte so realistisch wie möglich berechnen kann.

Zusammengefasst bietet dieses JSON eine strukturierte Datenbasis für die Beschreibung und den Aufbau einer 3D-Szene. Durch die Definition der wichtigsten Eigenschaften der Objekte in der Szene, wie Position, Material und Beleuchtung, ermöglicht es einem Raytracer, diese Informationen zu verwenden, um die Szene korrekt zu beleuchten und zu rendern.

8. Graphische Gestaltung und Nutzungskonzept

Im Folgenden werden einige grafische Gestaltungskonzepte der einzelnen Dialoge durch Mock-Up-Zeichnungen dargestellt. Textuelle Beschreibungen veranschaulichen zudem die Übersicht der Dialoge.

8.1 Hauptbildschirm

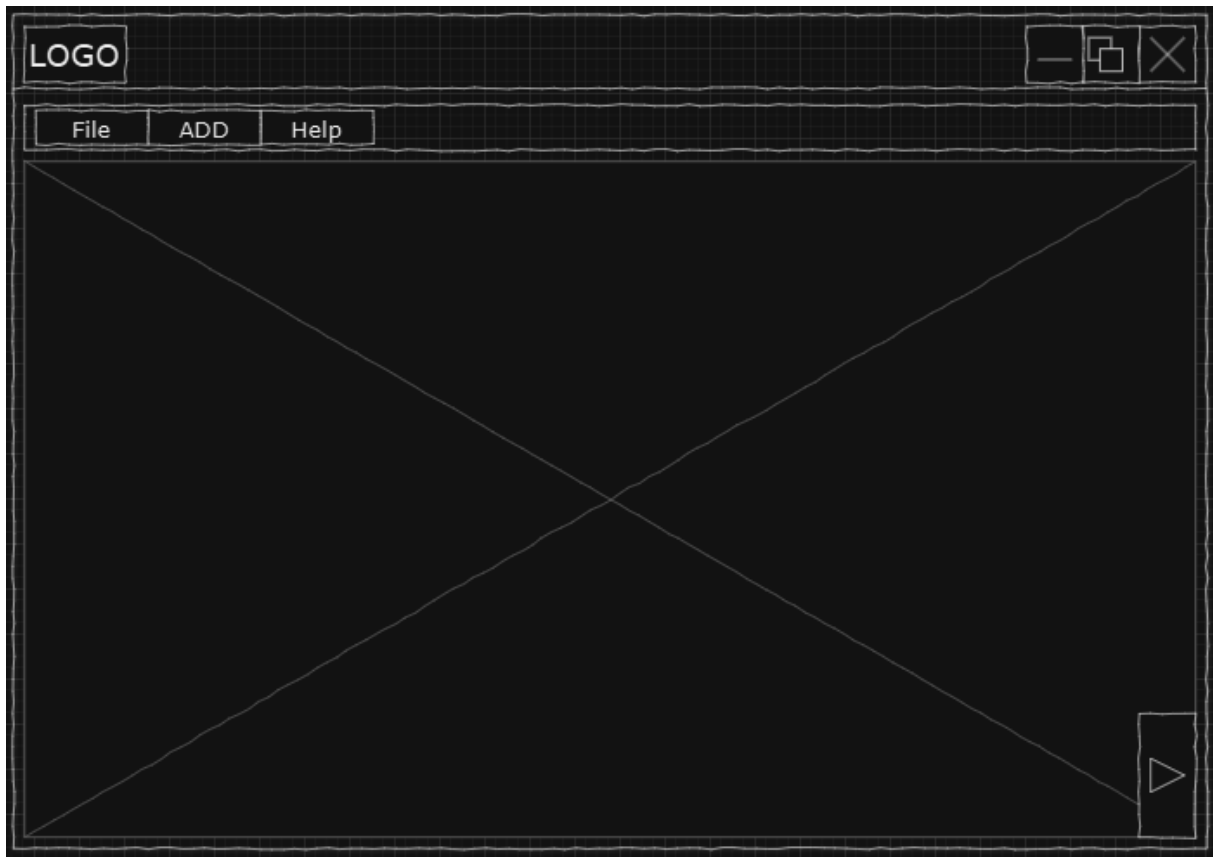


Unser Hauptbildschirm verfügt oben über einen Reiter, über den man auf weitere Informationen, Einstellungen oder funktionale Anforderungen zugreifen kann. Im Menü befinden sich die Einträge „File“ und „Add“, die jeweils als Dropdown-Menüs gestaltet sind und weiter unten detailliert aufgelistet und erklärt werden.

An der rechten Seite finden sich wichtige Informationen zur aktuellen 3D-Szene. Darunter ist ein Entity-Tree platziert, und direkt darunter werden weitere Informationen zur aktuellen 3D-Szene angezeigt, einschließlich Informationen zur eventuell geladenen JSON-3D-Szene. Diese Informationen lassen sich durch einen Button am Rand, der in der Mitte einen Pfeil enthält, einklappen und wieder ausklappen.

Hier sehen Sie eine Mockup-Zeichnung im eingeklappten Informationszustand:

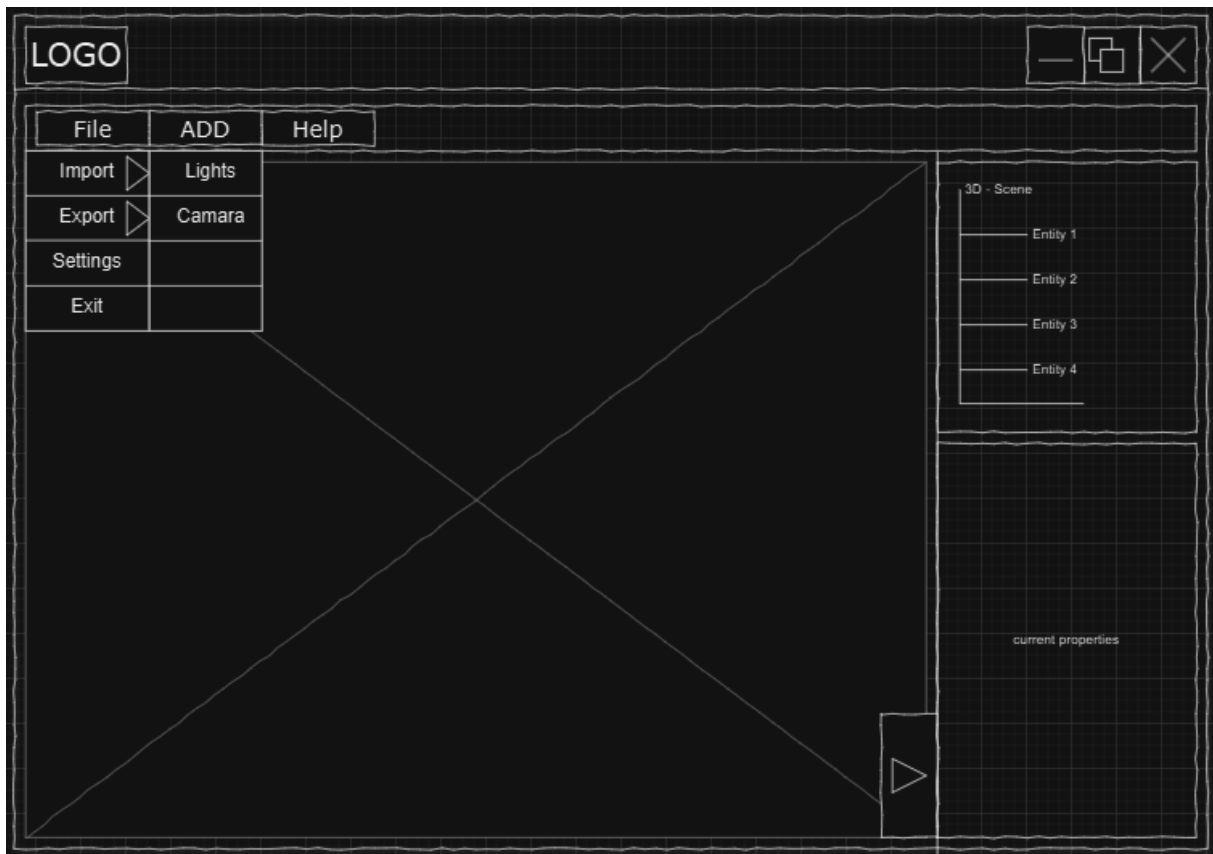
1. Eingeklappten Informationszustand:



Hier sind die Dropdown-Menüs der beiden Reiter im Detail dargestellt: Der Reiter *File* enthält die Optionen Import, Export, Settings und Exit, die den Zugriff auf verschiedene Datei- und Anwendungseinstellungen ermöglichen. Im Reiter *Add* befinden sich die Optionen Lights und Camera, mit denen sich Beleuchtungselemente sowie Kameras in die 3D-Szene hinzufügen lassen.

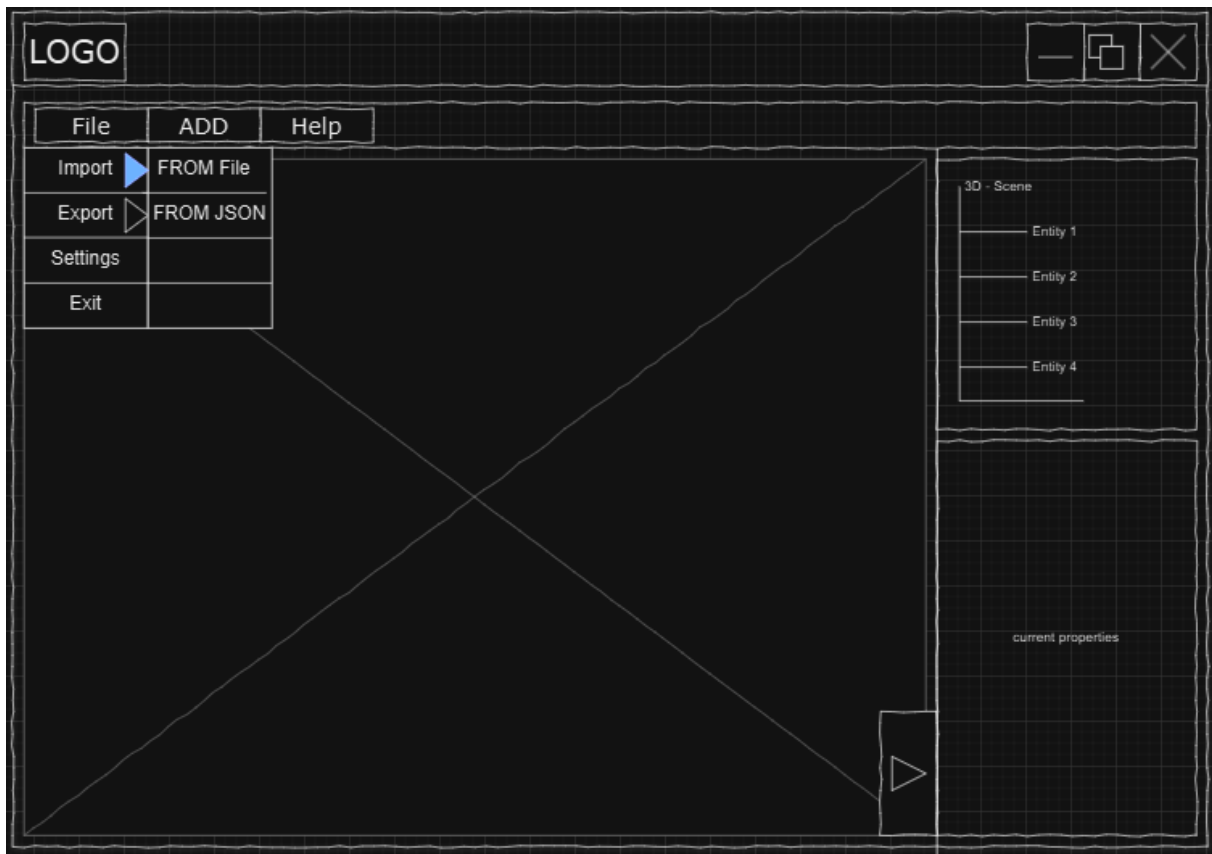
Die Optionen Lights und Camera bieten die Möglichkeit, entweder neue Lichtelemente oder Kameras zu erstellen oder bereits vorhandene Elemente in den Fokus zu nehmen. Sobald ein Element fokussiert ist, kann es mit der Maus bewegt und präzise in der 3D-Szene positioniert werden.

2. Ausgeklappte Reiter „File“ und „Add“:



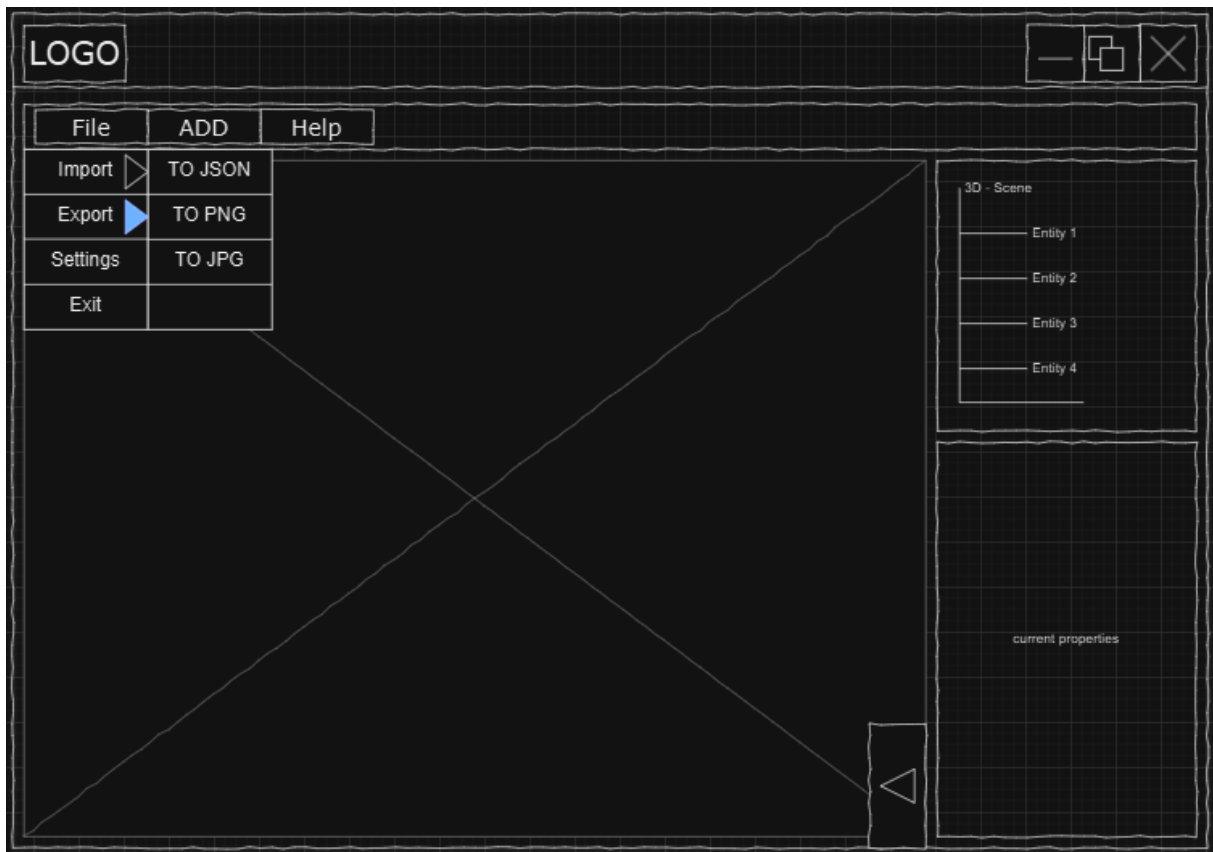
Nun werfen wir einen Blick auf die Optionen, die sich durch Anklicken oder Hover-Effekt beim Menüpunkt „Import“ öffnen. Hier hat der Nutzer die Wahl, entweder eine Datei oder eine 3D-Szene im JSON-Format zu öffnen bzw. zu importieren.

3. Ausgeklappte Import-Option:



Beim Menüpunkt „Export“ stehen nach dem Anklicken oder Hovern mehrere Exportoptionen zur Verfügung. Der Nutzer kann hier entscheiden, in welchem Format die Daten gespeichert werden sollen. Zur Auswahl stehen die Formate JSON, PNG und JPG, wodurch sowohl strukturierte Daten als auch Bilddateien generiert werden können, je nach gewünschtem Verwendungszweck und Zielplattform. Diese flexible Auswahl ermöglicht es, die exportierten Inhalte an spezifische Anforderungen oder Weiterverarbeitungsprogramme anzupassen.

4. Ausgeklappte Export-Option:



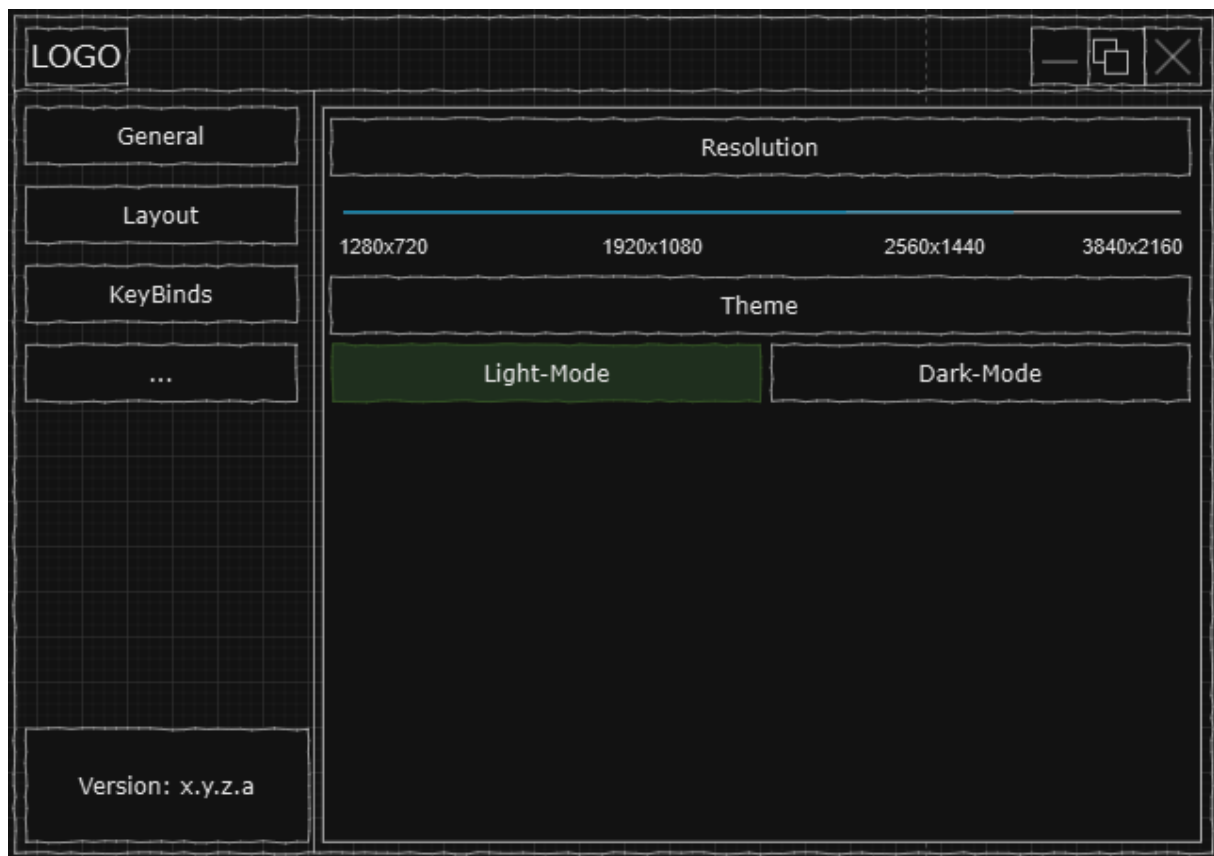
Nun kommen wir zur Option „Settings“. Durch das Öffnen dieser Option wird eine neue Oberfläche angezeigt, die den Hauptbildschirm überlagert und eine eigenständige Struktur aufweist. Diese Struktur enthält verschiedene Einstellungsmöglichkeiten, deren Aufbau und Funktionen später noch im Detail erklärt werden.

8.2 Einstellungen



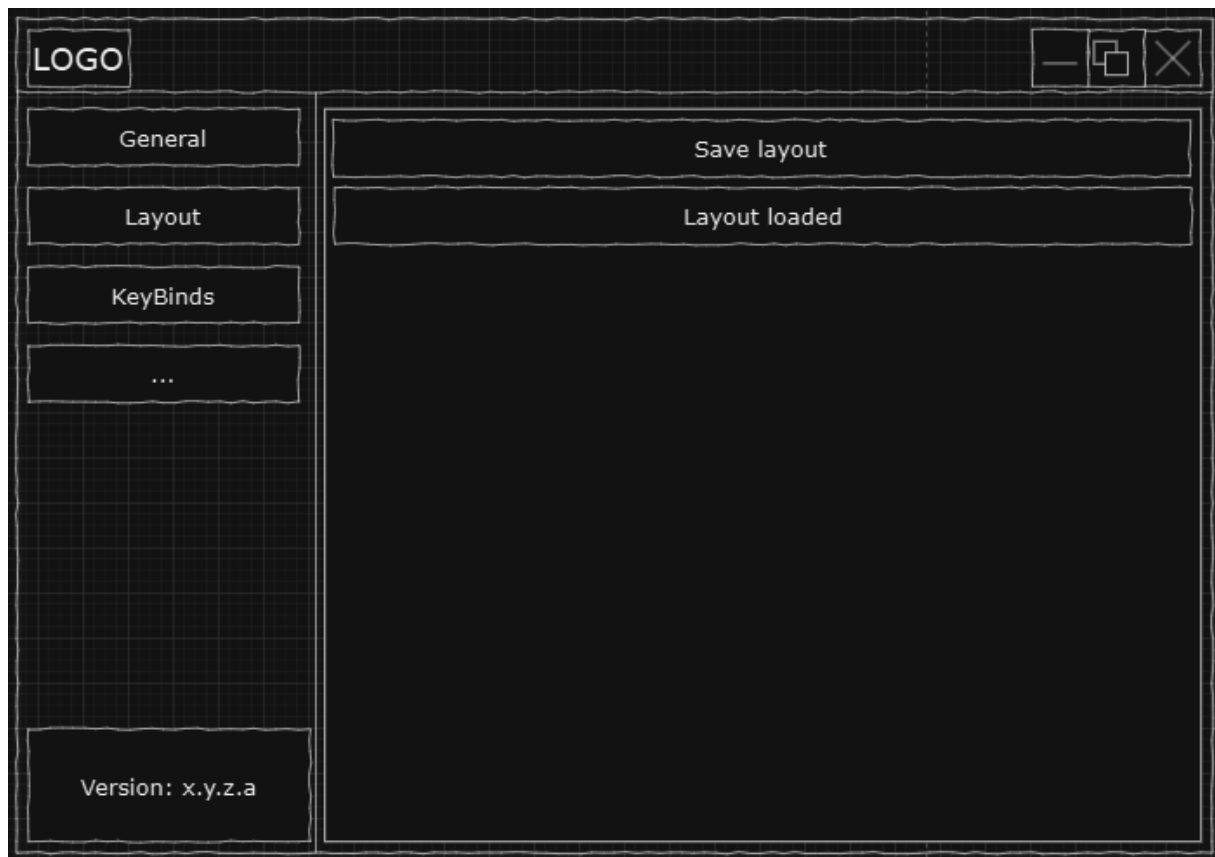
Mein Settings-Fenster weist eine einfache, aber moderne Struktur auf. Auf der linken Seite sind die Optionen wie „General“, „Layout“ und „Keybinds“ angeordnet. Wenn eine dieser Optionen angeklickt wird, werden auf der rechten Seite im großen Kontextfenster alle zugehörigen Eigenschaften und Möglichkeiten angezeigt. Diese übersichtliche Anordnung ermöglicht es den Nutzern, schnell zwischen den verschiedenen Einstellungskategorien zu wechseln und die gewünschten Anpassungen vorzunehmen.

5. General:



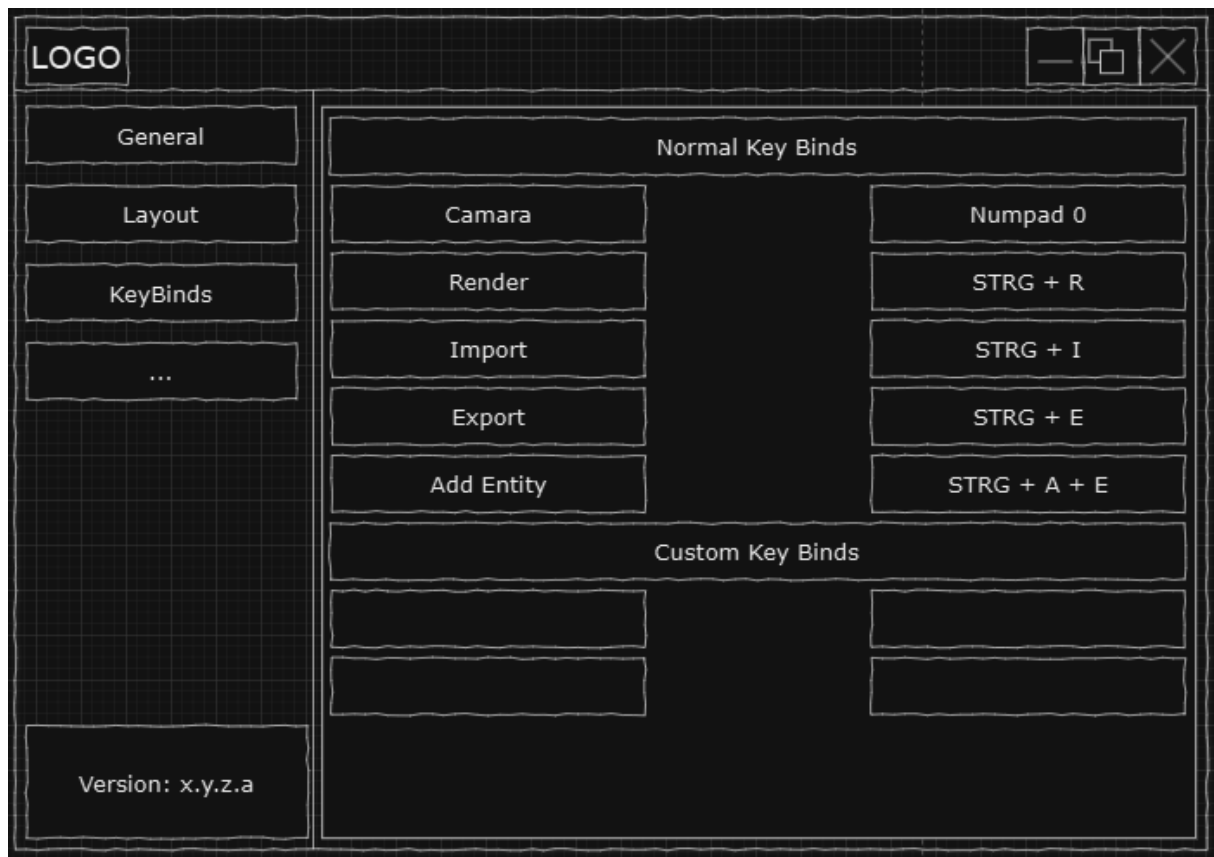
Wenn die Option „General“ ausgewählt wird, erhält der Nutzer die Möglichkeit, die Auflösung bzw. „Resolution“ der Anwendung anzupassen. Außerdem kann gewählt werden, ob die Anwendung im „Light Theme“ oder „Black Theme“ genutzt werden soll. Durch das Anklicken der jeweiligen Theme-Option wird diese hervorgehoben, sodass die aktuelle Einstellung klar angezeigt wird. Diese intuitive Gestaltung erleichtert die Anpassung der Benutzeroberfläche entsprechend den individuellen Vorlieben.

6. Layout:



Wenn die Option „Layout“ ausgewählt wird, hat der Nutzer die Möglichkeit, die Anordnung der NanoGUI-Elemente auf dem Hauptbildschirm zu speichern, da diese frei bewegbar sind. Außerdem steht ein Standard-Layout als Vorschlag zur Verfügung, das zu Beginn geladen wird. Dies ermöglicht es den Nutzern, entweder mit der vorgeschlagenen Anordnung zu arbeiten oder ihre eigene, personalisierte Anordnung zu erstellen und abzuspeichern.

7. Key-Binds:



Bei der Option „Key Binds“ hat der Nutzer die Möglichkeit, bereits vorgeschlagene Tastenkombinationen einzusehen und nach seinen Wünschen zu bearbeiten. Darüber hinaus stehen zusätzliche benutzerdefinierte Slots zur Verfügung, sodass der Nutzer auch eigene Tastenkombinationen festlegen kann, falls er spezielle Anforderungen oder Vorlieben hat. Diese Flexibilität ermöglicht eine individuelle Anpassung der Steuerung und verbessert die Benutzererfahrung erheblich.

9. Sources

9.1 Contents

Content 1: Link

Content 2: Link

Content 3: Link

9.2 Images

Figure 1: https://innosued.de/wp-content/uploads/2019/02/THU_Logo_Standard.jpg

Figure 2: Link

Figure 3: Link