

# Computergrafik

## Programmentwurf

über die Theoriephasen des dritten Studienjahrs

an der Fakultät für Technik  
im Studiengang Informatik

an der DHBW Ravensburg  
Campus Friedrichshafen

von

Johannes Brandenburger, Lukas Braun, Henry Schuler

17. November 2022

Bearbeitungszeitraum: 01.10.2022 - 21.11.2022

Kurs: TIT20

Dozent der Hochschule: Prof. Dr. Jürgen Schneider

# Gender Erklärung

Aus Gründen der besseren Lesbarkeit wird in dieser Bachelorarbeit auf die gleichzeitige Verwendung der Sprachformen männlich, weiblich und divers (m/w/d) verzichtet. Sämtliche Formulierungen gelten gleichermaßen für alle Geschlechter.

# Selbstständigkeitserklärung

gemäß Ziffer 1.1.13 der Anlage 1 zu §§ 3, 4 und 5 der Studien- und Prüfungsordnung für die Bachelorstudiengänge im Studienbereich Technik der Dualen Hochschule Baden-Württemberg vom 29.09.2017.

Wir versichern hiermit, dass wir unsere Bachelorarbeit (bzw. Projektarbeit oder Studienarbeit bzw. Hausarbeit) mit dem Thema:

## Computergrafik

selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt haben. Wir versichern zudem, dass die eingereichte elektronische Fassung mit der gedruckten Fassung übereinstimmt.

Friedrichshafen, 17. November 2022

Ort, Datum

Johannes Brandenburger

Friedrichshafen, 17. November 2022

Ort, Datum

Lukas Braun

Friedrichshafen, 17. November 2022

Ort, Datum

Henry Schuler

# Inhaltsverzeichnis

<b>Gendererklärung</b>	<b>II</b>
<b>Selbstständigkeitserklärung</b>	<b>III</b>
<b>Abkürzungsverzeichnis</b>	<b>V</b>
<b>Abbildungsverzeichnis</b>	<b>VI</b>
<b>Tabellenverzeichnis</b>	<b>VII</b>
<b>Listings</b>	<b>VIII</b>
<b>1 Einleitung</b>	<b>1</b>
1.1 Aufgabenstellung . . . . .	1
1.2 Aufbau der Arbeit . . . . .	1
<b>2 Tools</b>	<b>2</b>
<b>3 Designkonzept</b>	<b>3</b>
3.1 Klassenzimmer . . . . .	3
3.2 Flugsimulator . . . . .	6
<b>4 Realisierung</b>	<b>8</b>
4.1 Klassenzimmer . . . . .	8
4.2 Flugsimulator . . . . .	8
<b>5 Installationsanleitung</b>	<b>10</b>
<b>Literatur</b>	<b>A</b>
<b>A Anhang</b>	<b>A</b>

## Abkürzungsverzeichnis

<b>HDR</b> High Dynamic Range . . . . .	6
---	---

## Abbildungsverzeichnis

3.1	Klassenzimmer Skizze . . . . .	3
3.2	Klassenzimmer Entwurf mit Bemaßung . . . . .	4
3.3	Klassenzimmer Entwurf mit Fenster . . . . .	5
3.4	Klassenzimmer Entwurf der Lampen . . . . .	5
5.1	Klassenzimmer Entwurf mit Fenster . . . . .	10

## **Tabellenverzeichnis**

## Listings



# 1 Einleitung

## 1.1 Aufgabenstellung

Die Prüfungsleistung der Vorlesung Computergrafik beinhaltet die Erstellung eines Programm-entwurfs.

Dieser Programmentwurf besteht aus der Erstellung einer animierten 3D-Computergrafik. Hierzu sollen HTML, CSS JavaScript und WebGLv2 verwendet werden. Zur Szenenmodellierung darf außerdem die three.js Bibliothek verwendet werden. Der Programmentwurf muss folgenden Punkte enthalten:

- Szene ist dreidimensional
- Einzelne Objekte in der Szene sind animiert
- Kamera kann sich durch die Szene bewegen
- Mindestens eine Lichtquelle mit Phong-Beleuchtungmodell
- Control Panel zur Steuerung der 3D-Grafik

Das Controlpanel kann auch durch Interaktionen mit der Szene ersetzt werden.

## 1.2 Aufbau der Arbeit

Im folgenden werden zunächst die verwendeten Hilfsmittel erläutert, im Anschluss wird ein Konzept für die 3D-Szenze erarbeitet und in verschiedenen Diagrammen dargestellt. Abschließend wird das finale Produkt dargestellt und eine Installationanleitung zur Verfügung gestellt.

## 2 Tools

Wie in der Einleitung erwähnt werden für den Programmentwurf die Programmiersprachen HTML, CSS, JavaScript und WebGLv2 verwendet. Die 3D Szenenmodellierung wird mit der three.js Bibliothek realisiert. Zum erstellen der 3D Modelle wird Blender verwendet. Diese können anschließend als glTF (GL Transmission Format) in three.js importiert werden.

Zur Entwicklung des Sourcecodes wird der Editor Visual Studio Code verwendet. Zusätzlich wird eine Live Server Extension verwendet, diese startet einen Webserver mit der aktuellen Website. Der Sourcecode wird in einem Git Repository auf Github verwaltet.

Um die Szene zu entwickeln und Entwürfe grafisch darzustellen wird Microsoft Visio verwendet. Händische Zeichnungen werden mit Microsoft OneNote oder GoodNotes abhängig vom Teammitglied erstellt, da GoodNotes nur auf Apple Geräten verfügbar ist.

Alle verwendeten Hilfsmittel werden in der folgenden Auflistung dargestellt:

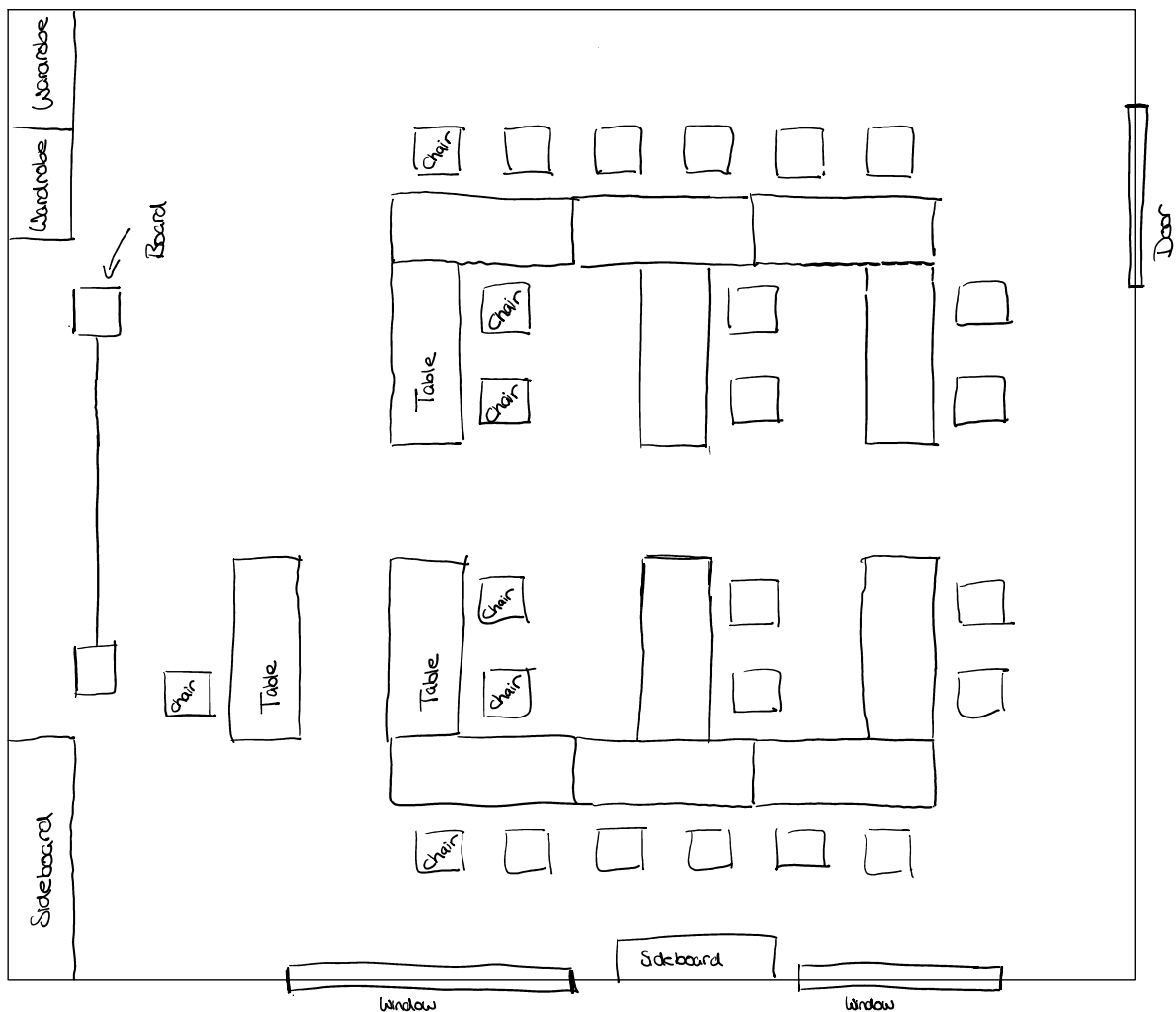
- Visual Studio Code mit Live Server Extension
- Blender
- Github
- Microsoft Visio
- Microsoft OneNote
- GoodNotes

## 3 Designkonzept

### 3.1 Klassenzimmer

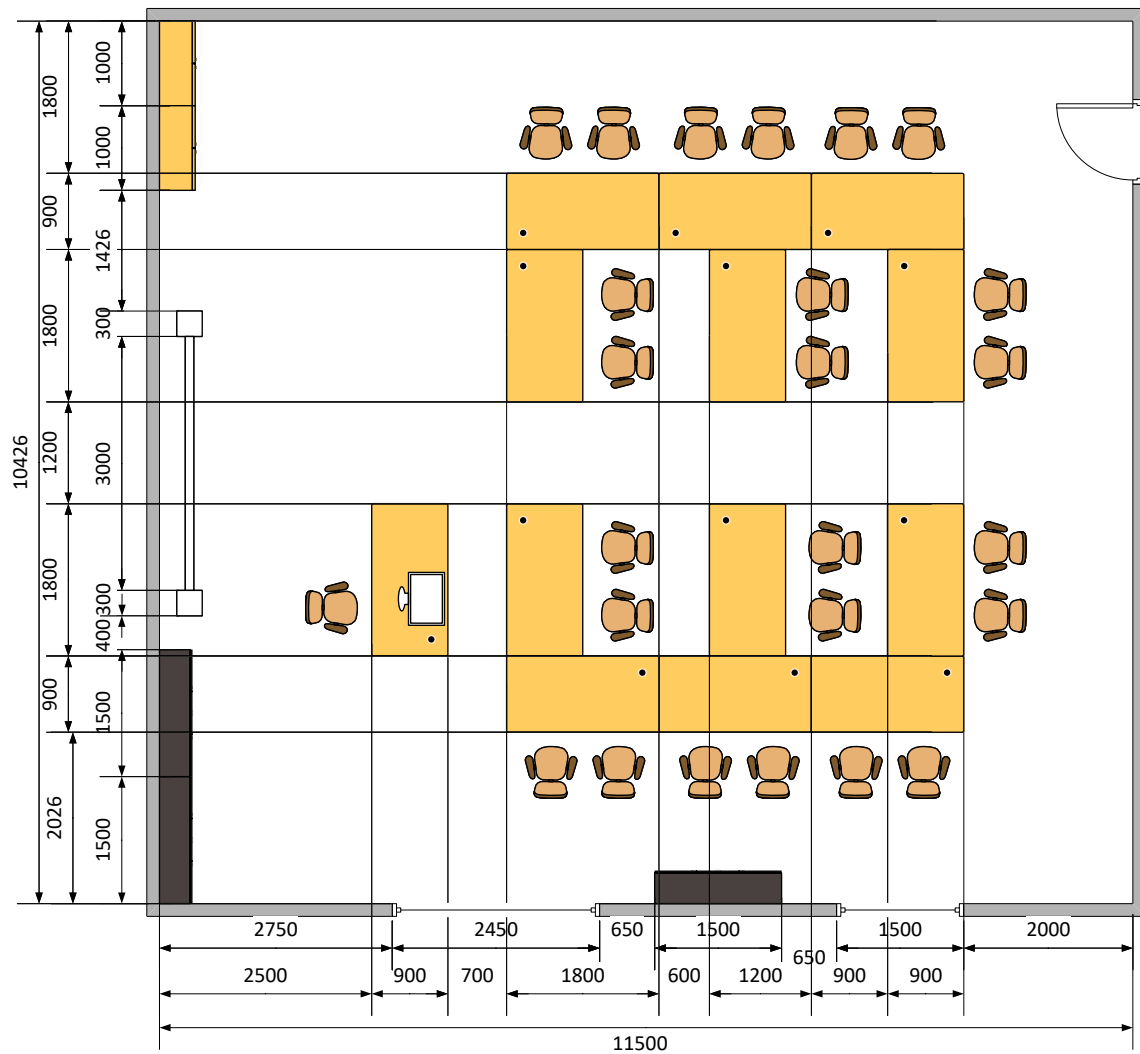
Als grundlegende Idee wurde zunächst ein Vorlesungsaal vorgeschlagen. Um weitere Komponenten aus den Anforderungen an diese Arbeit sinnvoll umzusetzen, wurde die grundlegende Idee überdacht und neu definiert als Klassenzimmer einer Flugschule.

Um eine erste Vorstellung des Klassenzimmers zu bekommen wurde zunächst eine händische Zeichnung angefertigt. Diese ist in der Abbildung 3.1 dargestellt.

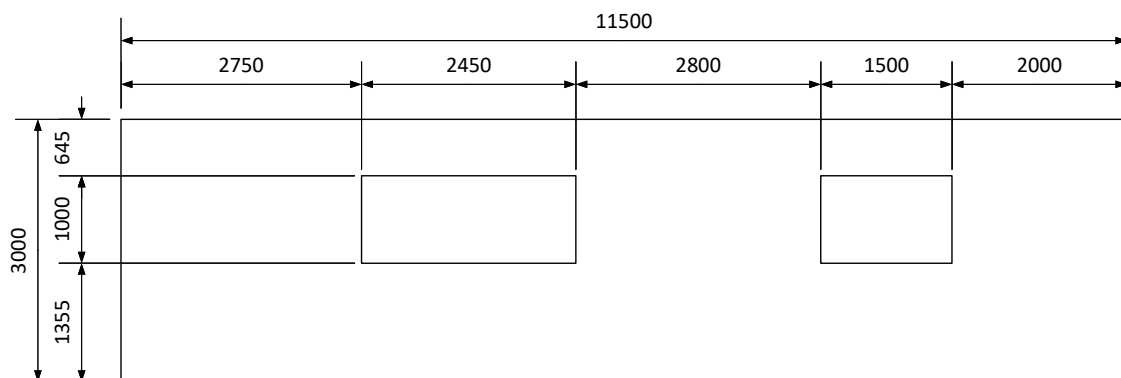


**Abb. 3.1:** Klassenzimmer Skizze

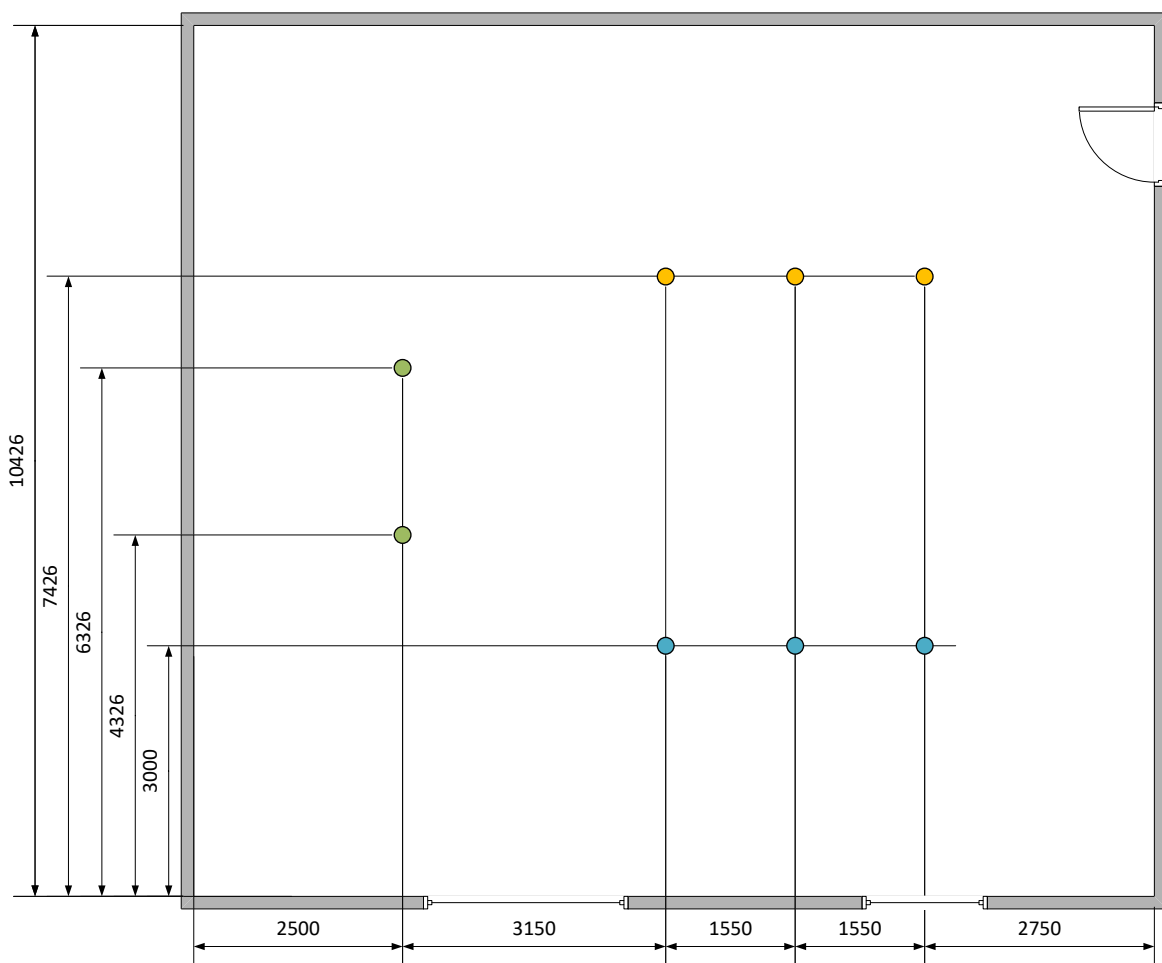
Anschließend wurde der Raum maßstabsgetreu in einem Bauplan gezeichnet um so die Abstände und Maße teilweise zu definieren. Diese Zeichnung ist in der Abbildung 3.2 dargestellt.



Um die Höhen der Fenster zu bestimmen wurde zusätzlich eine Seitenansicht erstellt. Diese ist in der Abbildung 3.3 zu sehen.

**Abb. 3.3:** Klassenzimmer Entwurf mit Fenster

Abschließend wurde eine weitere Zeichnung zur Platzierung der Lampen erstellt. Diese ist in der Abbildung 3.4 dargestellt.

**Abb. 3.4:** Klassenzimmer Entwurf der Lampen

Um die Anforderungen vollständig zu erfüllen müssen Interaktionen mit der 3D Szene möglich sein, diese werden im Folgenden beschrieben.

Im Klassenzimmer ist es möglich zu laufen, bei einer Kollision mit einem Gegenstand wird die Bewegung angehalten. Das Licht im Klassenzimmer kann durch drei Lichtschalter neben der Tür per Mausklick gesteuert werden. Mit einem Schalter kann jeweils ein Cluster angesteuert werden, diese sind in Abb. 3.4 farblich gekennzeichnet. Außerdem können die Stühle auf- und abgestuhlt werden. Bei einem Blick aus dem Fenster soll die DHBW dargestellt werden, diese wird als High Dynamic Range (HDR) Bild eingebunden. Zusätzlich können die Schränke geöffnet und geschlossen werden und die Tafel hoch bzw. nach unten geschoben werden.

Aus den beschriebenen Animationen und den Plänen aus den Abbildung 3.2, 3.3 und 3.4 ergeben sich alle Komponenten der Szene. Einige Maße werden bereits durch den Plan vorgegeben in einem weiteren Schritt werden diese nun vollständig definiert.

Mit den oben definierten Größen können anschließend die Blender Modelle erstellt werden. Um das Klassenzimmer ansprechender darzustellen werden weitere Elemente aus dem Internet eingefügt, diese werden mit ihren Quellen in der untenstehenden Tabelle aufgezählt.

## 3.2 Flugsimulator

Als Erweiterung der Flugschule, wird ein Flugsimulator-Spiel integriert, welches folgende Funktionen beinhaltet:

Durch einen Klick auf den Monitor in der Flugschule wird auf eine Unterseite weitergeleitet und der Flugsimulator gestartet. Ein Flugzeug-Modell wird in der Mitte der Szene platziert. Dessen Flugrichtung ist mit der Maus veränderbar, wodurch eine Steuerung des Flugzeugs möglich ist. Die Geschwindigkeit des Flugzeugs wird nicht durch den Benutzer verändert, sondern passt sich anhand der Neigung des Flugzeugs an.

Beim Starten des Flugsimulator-Spiels wird ein Countdown gestartet, welcher 60 Sekunden lang dauert. In dieser Zeit kann der Benutzer durch zufällig generierte Ringe fliegen und dabei (je nach Art des Rings) einen oder fünf Punkte sammeln. Zur zusätzlichen Herausforderung werden Hindernisse generiert. Wenn das Flugzeug in eines dieser Hindernisse oder gegen einen der Ringe fliegt oder die Zeit abläuft, wird das Spiel beendet. Der Benutzer bekommt hierbei die erreichte Punktzahl angezeigt und kann das Spiel erneut starten oder zur Flugschule zurückkehren.

Im Hintergrund des Spiels wird ein Ozean und ein Himmel dargestellt. Die Kamera bewegt sich während des Fluges mit dem Flugzeug mit. Die Steuerung kann durch das Drücken der

Taste I invertiert werden, um allen Benutzern das bestmögliche Spielerlebnis zu ermöglichen.

## 4 Realisierung

### 4.1 Klassenzimmer

Anschließend werden diese Modelle in three.js eingebunden und gemäß dem Plan in Abbildung 3.2 platziert. Abschließend müssen die Scheiben in die Fenster eingefügt werden, da dies nicht mit einem Export aus Blender möglich ist.

Nachdem die Szene vollständig erstellt wurde, müssen nachfolgend die definierten Interaktionen hinzugefügt werden. Um sich im Raum zu bewegen, wird ein unsichtbarer Quader hinzugefügt, der die Person darstellt. Dieser Quader enthält auch die Kamera und kann mit den Tasten „W“, „A“, „S“ und „D“ durch den Raum bewegt werden. Um Kollisionen zu erkennen wird vor Ausführung der Bewegung überprüft, ob die Bewegung ausgeführt werden darf. Würde der Quader innerhalb eines anderen Objekts sein, wird die Bewegung nicht ausgeführt.

Damit die Tafel nach oben bzw. nach unten bewegt werden kann, wird die Tafel mit der Maus angewählt und verschoben. Hierzu wird bei einem Mausklick überprüft ob auf die Tafel geklickt wurde und anschließend abhängig von der Maus Bewegung in die Y-Achsen Richtung die Tafel verschoben. Das Auf- und Abstuhlen der Stühle funktioniert ähnlich. Es wird ebenfalls überprüft ob bei einem Mausklick auf einen Stuhl geklickt wurde und anschließend wird dieser auf- oder abgestuhlt. Das Öffnen oder Schließen der Schranktüren funktioniert analog zu den Stühlen. Zusätzlich darf maximal ein Abstand von vier Metern zwischen dem Objekt und dem Personen-Quader sein um die Animation auszuführen.

### 4.2 Flugsimulator

Um das Flugsimulator-Spiel umzusetzen, wird zunächst eine Umgebung benötigt. Diese wird mit Hilfe von vorgefertigten three.js Elementen wie einem Meer und einem Himmel erstellt. Anschließend wird ein Flugzeugmodell im .glb Format eingelesen und der Szene hinzugefügt.

Da die FlyControls von three.js nicht den Ansprüchen des Spiels genügen, wird eine eigene Steuerung für das Bewegen des Flugzeugs implementiert. Dabei wird die Cursorposition mit Abstand zum Mittelpunkt des Bildschirms verglichen und die Flugrichtung (Vektor) entsprechend angepasst. Für diese Anpassung werden die Helper-Funktionen `turnVectorAroundVerticalAxis` und `turnVectorAroundHorizontalAxis` implementiert und in jeder Animationsschleife aufgerufen. Speziell für das Drehen eines Vektors um die horizontale Achse, ist die Mathematik nicht trivial, da zunächst eine Rotationsachse berechnet werden muss, die senkrecht zur Flugrichtung steht und bei jeder Position des Flugzeugs die gleiche Richtung hat.



Eine besondere Herausforderung bei der Flugsteuerung stellen die Überschlge des Flugzeugs dar. Diese mssen vom Programm erkannt werden, um aktiv die Kamera, das Flugzeug und die Steuerung zu invertieren. Um die Geschwindigkeit des Flugzeugs mglichst realistisch zu gestalten, wird diese proportional zur Neigung beim Sinkflug erhht und beim Steigen verringert bis bestimmte Minimal- und Maximalwerte erreicht sind.

Die Ringe werden in einer Schleife als three.js Torus-Objekte erstellt und zufllig in der Szene platziert. Auerdem werden der Szene Hindernisse hinzugefgt, die das Flugzeug nicht berhren darf. Diese werden aus Dodekaeder, Icosahedron, Oktaeder und Tetraeder Elementen erstellt und ebenfalls zufllig in der Szene angeordnet.

Um zu erkennen, ob ein das Flugzeug erfolgreich durch einen Ring geflogen ist oder mit einem Ring oder einem sonstigen Hindernis kollidiert ist, wird Kollisionserkennung bentigt. Dazu wird zunchst das Element mit dem geringsten Abstand zum Flugzeug ermittelt. Von diesem Objekt wird anschlieend eine BoundingBox bzw. ein BoundingSphere erstellt und geprft, ob diese die Flugzeugposition beinhaltet. Bei den Ringen wird zustzlich auf den Abstand zum Mittelpunkt des Torus-Objekts geprft und so determiniert, ob das Flugzeug den Ring durchfliegt, in den Rand des Rings fliegt oder auerhalb am Rand des Rings vorbeifliegt. Die Kollisionserkennung wird in jeder Animationsschleife aufgerufen und die Punktezahl entsprechend angepasst. Bei einem Kollisionsereignis oder beim Ablauf der Zeit, wird die Flug-Animation unterbrochen, die aktuelle Punktezahl in einem „Game Over“-Overlay angezeigt und der User bekommt die Mglichkeit, das Spiel neu zu starten oder zurck zur Flugschule zu gelangen.

## 5 Installationsanleitung

1. Wenn node und npm noch nicht installiert sind, installieren Sie diese von [nodejs.org](https://nodejs.org).
2. Führen Sie `npm run start` aus, um alle Abhängigkeiten zu installieren und den Webserver auf Port 3000 zu starten.
3. Öffnen Sie `http://localhost:3000` in Ihrem Browser.

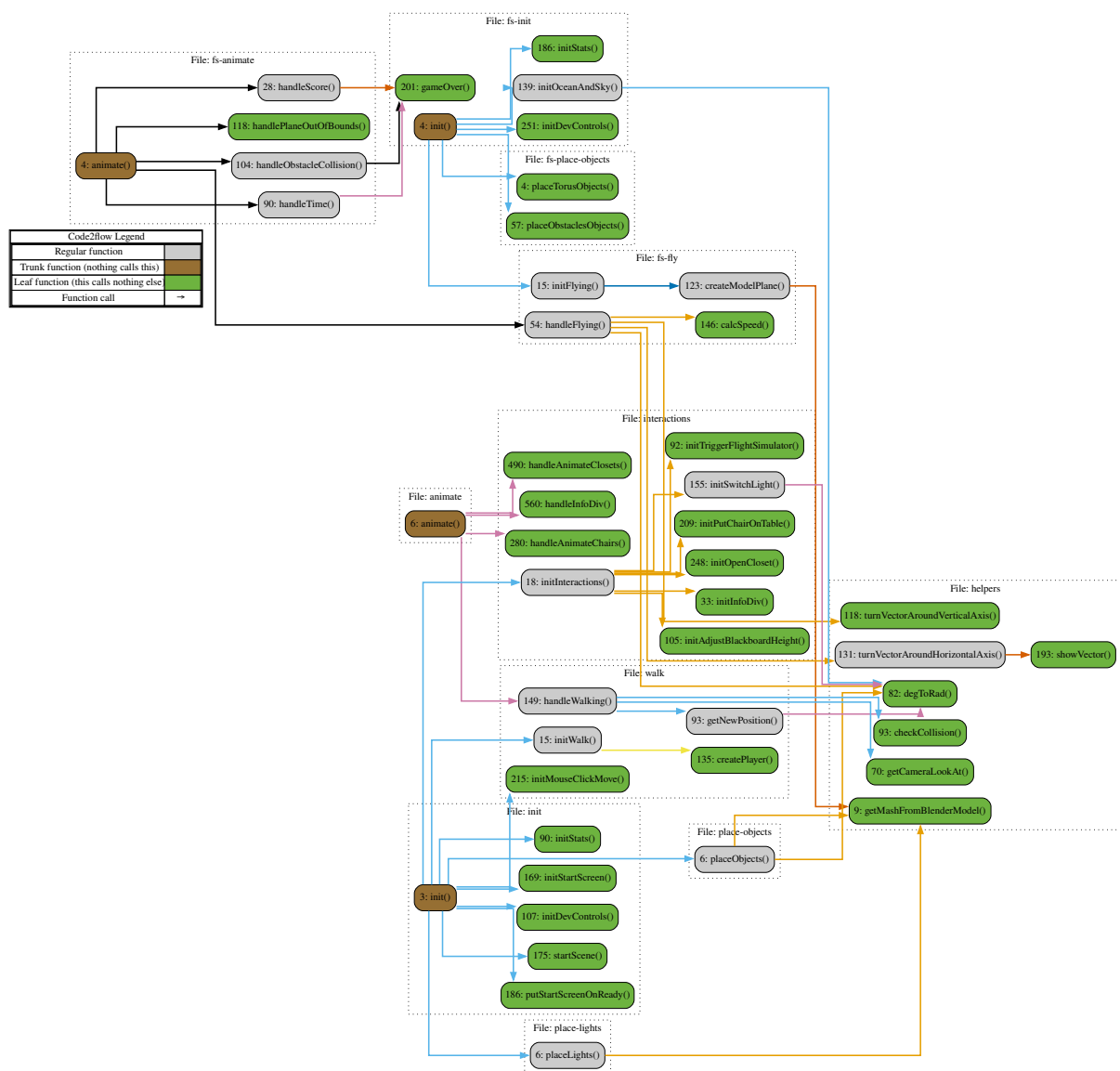


Abb. 5.1: Klassenzimmer Entwurf mit Fenster

## A Anhang