

Campus Karte

Entwicklungsprojekt WS22

Domenic Wolf

Jens Burgdorf

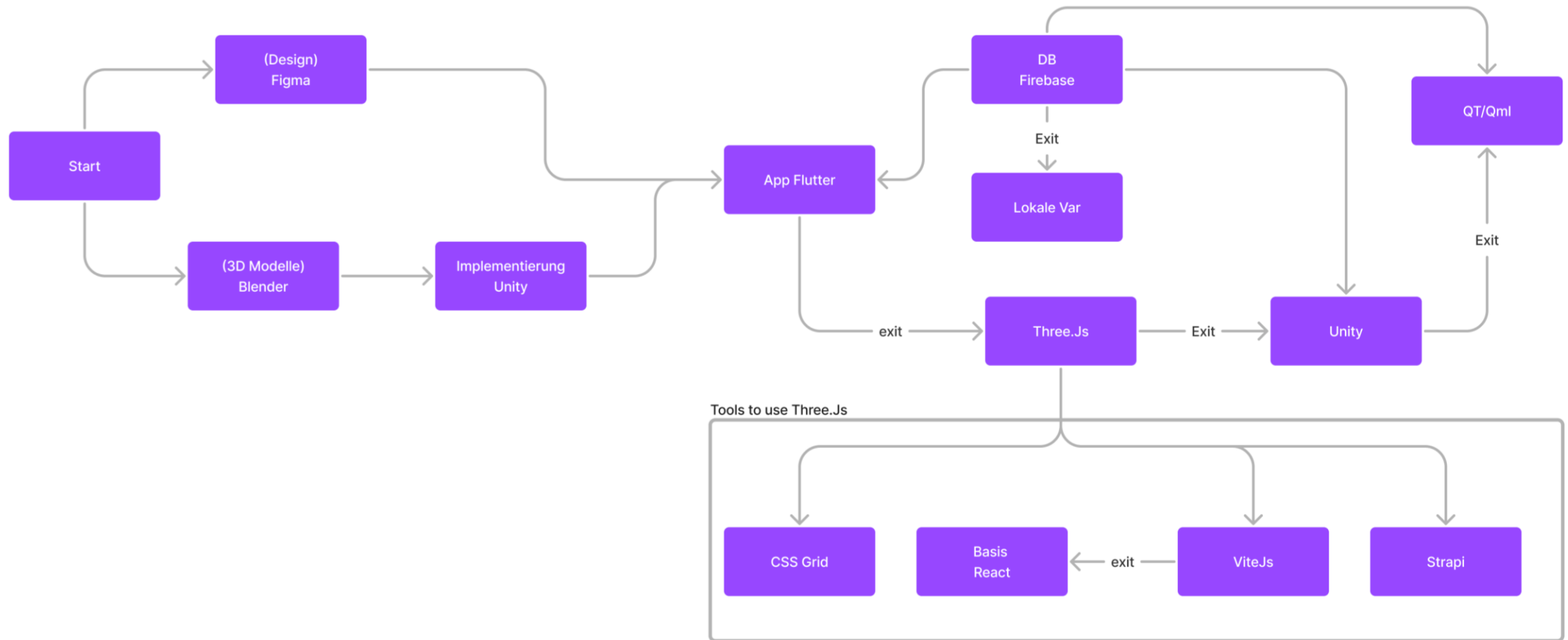
Anastasia Chouliaras

Audit 1 + 2

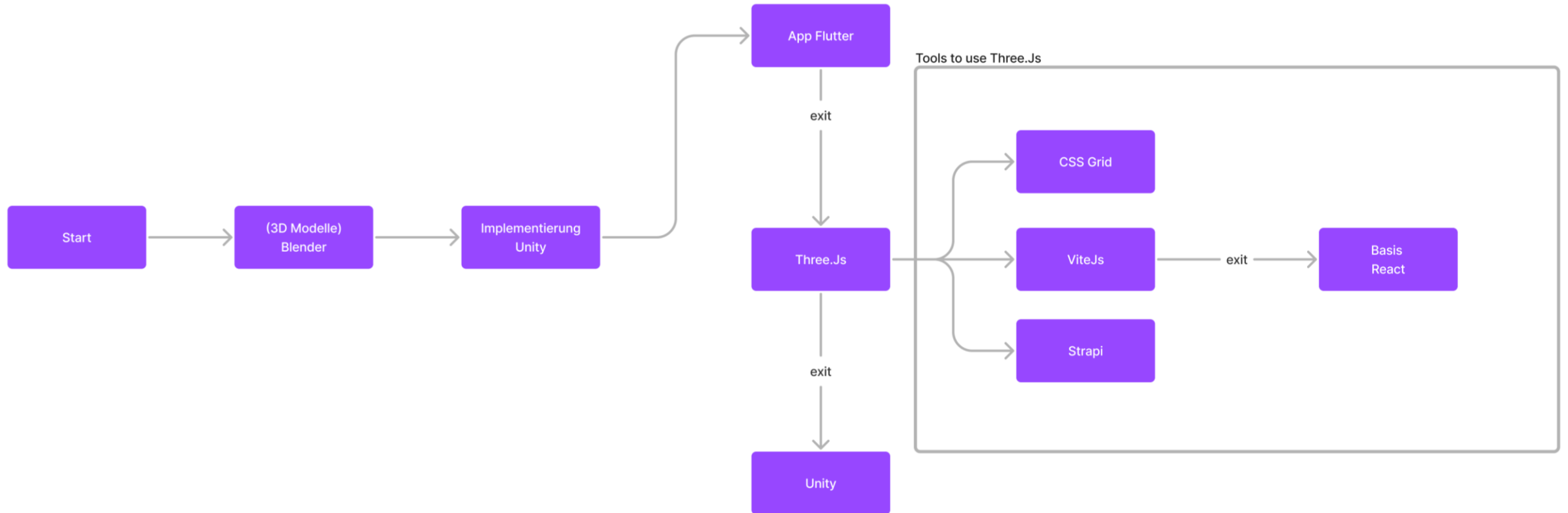
Campus Karte - Entwicklungsprojekt WS22



Vorgehen



Proof of Concepts



Plattformabhängige Erfolgskriterien der PoCs

Bisherige Artefakte:

- Plattformsentscheidung mit Vor- und Nachteilen
- Einzelne Kriterien für die verschiedenen Anwendungen/PoCs

Probleme der Darstellung:

- Keine eindeutige Zuordnung von Plattform und PoCs
- Keine einheitlichen Vergleichskriterien

Web oder App? Vor- und Nachteile

	Pro	Contra
Web-Anwendung	<ul style="list-style-type: none">• OS unabhängig --> mehr User werden erreicht• Kann gut integriert und verlinkt werden	<ul style="list-style-type: none">• keine Offline Nutzung möglich• Aufruf durch URL nicht intuitiv• unterschiedliche Bedienung bei desktop und mobil Version• Einarbeitung in ThreeJS (oder andere Software) notwendig --> komplexer
App	<ul style="list-style-type: none">• Personalisierbar (Favoriten, Stundenplan)• Offline Verwendbar• Entwicklung mit Unity vermutlich einfacher -> größerer Umfang & Fortschritt	<ul style="list-style-type: none">• OS abhängig• auf lange Zeit redundant• Hürde der Nutzung durch notwendigen Download

Flutter

Success	Breake Point
Einfache Verbindung mit Unity	GitHub Problems
Funktion im Web	Probleme bei der 3D <u>umsetzung</u>
	Performance Probleme
	Responsive Probleme
	Firebase Probleme

Three.js

Success	Breake Point
Schnelles aufsetzen	Probleme bei der 3D Umsetzung
Keine Installation Probleme	Performance Probleme
Responsive auf allen Bildschirm Größen	Responsive Probleme
	Firebase Probleme

Unity

Success	Breake Point
App Prototyp für iOS & Android	Probleme bei der 3D Umsetzung
	Responsive Probleme

Plattformabhängige Erfolgskriterien der PoCs

Neue Erfolgskriterien der PoCs:

- Sind vereinheitlicht
- Berücksichtigen die Plattform
- Haben klare Fail und Exit- Kriterien bekommen

Flutter	Three.JS	Unity
Einrichten der Entwicklungsumgebung	Einrichten der Entwicklungsumgebung	Einrichten der Entwicklungsumgebung
Anwendung im Web ausführbar (Chrome)	Anwendung im Web ausführbar (Chrome)	Die Anwendung ist auf einem Android Device ausführbar
Ein 3D Objekt lässt sich einfügen	Ein 3D Objekt lässt sich einfügen	Ein 3D Objekt lässt sich einfügen
Eine Kamera lässt sich bewegen	Eine Kamera lässt sich bewegen	Eine Kamera lässt sich bewegen
Das Arbeiten mit Flutter hatte ein angemessenes Tempo	Das Arbeiten mit Three.JS hatte ein angemessenes Tempo	Das Arbeiten mit Unity hatte ein angemessenes Tempo
Die Anwendung ist auf einem IOS Device ausführbar		
Die Anwendung ist auf einem Android Device ausführbar		
Eine Kamera lässt sich auf allen getesteten Devices bewegen		

Auswertung - Flutter

- Einrichtung hat sehr lange gedauert
- Nutzung und Programmierung der Kamera unklar
- IOS Geräte für uns nicht Testbar
- Trotz Dokumentation sehr schwierige Programmierung mit Dart

→ Nur ein Kriterium wird klar erfüllt, der PoC/Rapid Prototype mit Flutter war NICHT erfolgreich

Kriterium	Fail-Kriterium	Exit-Kriterium	Fallback-Kriterium	Auswertung
Einrichten der Entwicklungsumgebung	Einrichtung kann nicht bei allen Teammitgliedern bis zum 22.11.23 abgeschlossen werden	Die Einrichtung verläuft bei allen Teammitgliedern unproblematisch	Ausweichen auf den zweiten Proof of Concept mit Three.js	Die Entwicklungsumgebung konnte eingerichtet werden. Es traten jedoch bei unterschiedlichen Teammitgliedern unterschiedliche Probleme auf.
Anwendung im Web ausführbar (Chrome)	Anwendung kann nicht in einem Browser ausgeführt werden	Anwendung kann in einem Browser ausgeführt werden	Ausweichen auf den zweiten Proof of Concept mit Three.js	Nach der Einrichtung ließ sich das Basic-Project direkt problemlos kompilieren.
Ein 3D Objekt lässt sich einfügen	Es lässt sich kein 3D Objekt rendern	Es lässt sich ein 3D Objekt rendern	Ausweichen auf den zweiten Proof of Concept mit Three.js	Der Import der benötigten Packages war erst nach vielen Versuchen erfolgreich. Die Packages sind zusätzlich alle noch in einer frühen Entwicklungsphase.
Eine Kamera lässt sich bewegen	Die Kamera kann nicht gezielt kontrolliert/ bewegt werden	Die Kamera kann kontrolliert/ bewegt werden	Ausweichen auf den zweiten Proof of Concept mit Three.js	Nach Import des 3D Objekts ist eine Kamera vorhanden, bzw. lässt sich bewegen, jedoch ohne eigene Implementierung. Nicht direkt nachvollziehbar.
Das Arbeiten mit Flutter hatte ein angemessenes Tempo	Die Arbeit mit Flutter hat zu starken zeitlichen Einbußen geführt	Bei der Arbeit mit Flutter konnte der PoC mit angemessenem Aufwand durchgeführt werden.	Ausweichen auf den zweiten Proof of Concept mit Three.js	Das Implementieren der 3D-Funktionalität war mit einem hohen Zeitaufwand verbunden. Zudem gab es nur wenig Dokumentation und
Die Anwendung ist auf einem IOS Device ausführbar	Die Anwendung ist nicht auf einem IOS Device ausführbar	Die Anwendung ist auf einem IOS Device ausführbar	Die Unterstützung des Devices ist vorerst nicht mehr im Projektumfang	Das Testen für Apple-Geräte ist nur auf Mac-OS Geräten möglich.
Die Anwendung ist auf einem Android Device ausführbar	Die Anwendung ist nicht auf einem Android Device ausführbar	Die Anwendung ist auf einem Android Device ausführbar	Die Unterstützung des Devices ist vorerst nicht mehr im Projektumfang	Der Android Studio Emulator konnte die App ausführen, allerdings nicht bei allen Teammitgliedern.
Eine Kamera lässt sich auf allen getesteten Devices bewegen	Die Kamera kann nicht auf allen getesteten Devices kontrolliert/ bewegt werden	Die Kamera kann auf allen getesteten Devices kontrolliert/ bewegt werden	Ausweichen auf den zweiten Proof of Concept mit Three.js	Nach Import des 3D Objekts ist eine Kamera vorhanden, bzw. lässt sich bewegen, jedoch ohne eigene Implementierung. Nicht direkt nachvollziehbar.

Auswertung – Three.js

- Einrichtung war problemlos und sehr kurz
- JavaScript ist gut Dokumentiert und bereits etabliert

→ Alle Kriterien werden klar erfüllt, der PoC/Rapid Prototype mit Three.js war erfolgreich

Kriterium	Fail-Kriterium	Exit-Kriterium	Fallback-Kriterium	Auswertung
Einrichten der Entwicklungsumgebung	Einrichtung kann nicht bei allen Teammitgliedern bis zum 22.11.23 abgeschlossen werden	Die Einrichtung verläuft bei allen Teammitgliedern unproblematisch	Ausweichen auf den dritten Proof of Concept mit Unity	Die Einrichtung war bei Allen schnell und Problemlos. Für das Ausführen wurde zusätzlich in Visual Studio Code das Plugin "Live Server" genutzt
Anwendung im Web ausführbar (Chrome)	Anwendung kann nicht in einem Browser ausgeführt werden	Anwendung kann in einem Browser ausgeführt werden	Ausweichen auf den dritten Proof of Concept mit Unity	Hat mit dem Plugin problemlos geklappt.
Ein 3D Objekt lässt sich einfügen	Es lässt sich kein 3D Objekt rendern	Es lässt sich ein 3D Objekt rendern	Ausweichen auf den dritten Proof of Concept mit Unity	THREE.js stellt den Obj loader zur Verfügung. Die genaue Funktionalität ist einsehbar.
Eine Kamera lässt sich bewegen	Die Kamera kann nicht gezielt kontrolliert/ bewegt werden	Die Kamera kann kontrolliert/ bewegt werden	Ausweichen auf den dritten Proof of Concept mit Unity	THREE.js stellt OrbitControls zur Verfügung. Die genaue Funktionalität ist einsehbar.
Das Arbeiten mit Three.JS hatte ein angemessenes Tempo	Die Arbeit mit Three.JS hat zu starken zeitlichen Einbußen geführt	Bei der Arbeit mit Three.JS konnten PoC's mit angemessenem Aufwand durchgeführt werden.	Ausweichen auf den dritten Proof of Concept mit Unity	Der zeitliche Aufwand war angemessen (wenige Stunden).

Auswertung – Unity

→ Notwendigkeit für Unity als Fallback-Szenario entfällt

Kriterium	Fail-Kriterium	Exit-Kriterium	Fallback-Kriterium	Auswertung
Einrichten der Entwicklungsumgebung	Einrichtung kann nicht bei allen Teammitgliedern bis zum 22.11.23 abgeschlossen werden	Die Einrichtung verläuft bei allen Teammitgliedern unproblematisch	Projektansatz gescheitert	
Die Anwendung ist auf einem Android Device ausführbar	Die Anwendung ist nicht auf einem Andrid Device ausführbar	Die Anwendung ist auf einem Android Device ausführbar	Projektansatz gescheitert	
Ein 3D Objekt lässt sich einfügen	Es lässt sich kein 3D Objekt rendern	Es lässt sich ein 3D Objekt rendern	Projektansatz gescheitert	
Eine Kamera lässt sich bewegen	Die Kamera kann nicht gezielt kontrolliert/ bewegt werden	Die Kamera kann kontrolliert/ bewegt werden	Projektansatz gescheitert	
Das Arbeiten mit Unity hatte ein angemessenes Tempo	Die Arbeit mit Flutter hat zu starken zeitlichen Einbußen geführt	Bei der Arbeit mit Flutter konnte der PoC mit angemessenem Aufwand durchgeführt werden.	Projektansatz gescheitert	

Risikoanalyse - SWOT

		Opportunities	Threats
Strengths	Internal	<ul style="list-style-type: none"> • Erlernung neuer Programmierkenntnisse • Erstellung eines funktionalen Prototypen, welcher die Grundlage für weitere Projekte bilden kann • Weitere Erfahrung in der Entwicklung und Modellierung eines Systems sammeln • JavaScript ist gut dokumentiert 	<ul style="list-style-type: none"> • Fehleinschätzung der Umsetzbarkeit • Fehlkalkulation der benötigten Arbeitszeit • Ausfälle durch Krankheit • Nicht ausreichende Kommunikation innerhalb des Teams • Kann der User die Gebäude gut erkennen
	External		
	Weaknesses	<ul style="list-style-type: none"> • Grundkenntnisse in OpenGL • Fortgeschrittene Kenntnisse in der Arbeit mit 3D-bezogenen Daten • Kenntnisse verschiedener Programmiersprachen • Unterschiedliche Stärken der Teammitglieder und dadurch mögliche Ergänzung untereinander 	<ul style="list-style-type: none"> • Adaption von OpenGL zu WebGL • Verfeinerung der Kommunikationsfähigkeit innerhalb eines Teams
		<ul style="list-style-type: none"> • Kein reine Konzentration auf EP möglich -> Weitere Module müssen im gleichen Zeitraum bearbeitet werden • keine fortgeschrittenen Programmierkenntnisse in Javascript 	<ul style="list-style-type: none"> • Durchführung von POCs für benötigte Kernfunktionalität
		<ul style="list-style-type: none"> • Vertiefung der Fähigkeiten im Umgang mit Javascript 	<ul style="list-style-type: none"> • Regelmäßig den aktuellen Stand und das weitere Vorgehen im Open Space besprechen • mindestens Wöchentliche treffen um ausreichende Kommunikation sicherzustellen

VERWORFEN

Anforderungen - muss

[F010] Die Anwendung **muss** dem Nutzer die Möglichkeit bieten Start sowie einen Endpunkt einzugeben.

[F020] Die Anwendung **muss** dem Nutzer die Möglichkeit bieten den Weg zwischen zwei Räumen im Campus zu zeigen.

[F030] Die Anwendung **muss** dem Nutzer die Möglichkeit bieten Start sowie einen Endpunkt per Mausklick auf die Karte festzulegen.

[F040] Die Anwendung **muss** dem Nutzer die Möglichkeit bieten die Karte zu Drehen und zu bewegen.

Anforderungen – kann und soll

[F050] Die Anwendung **soll** dem Nutzer die Möglichkeit bieten zwischen einer perspektivischen und orthographischen Kamera zu wählen.

[F060] Die Anwendung **soll** dem Nutzer die Möglichkeit bieten diese auf jedem Gerät mit einem aktuellen Webbrowser und einer Internetverbindung aufrufen zu können.

[F070] Die Anwendung **soll** dem Nutzer die Möglichkeit bieten Ausgewählte Räume visuell hervorzuheben.

[F080] Die Anwendung **soll** dem Nutzer die Möglichkeit bieten Informationen zu den Räumen anzuzeigen.

[F090] Die Anwendung **kann** dem Nutzer die Möglichkeit bieten Räume als Favoriten zu speichern und auszuwählen.

[F100] Die Anwendung **kann** dem Nutzer die Möglichkeit bieten die Routengeneration durch URL-Parameter anzugeben.

Risikoanalyse - Formlos

- Für die Routenermittlung muss ein passender Algorithmus gefunden und Implementiert werden [F020]
- Selektion der einzelnen Räume im 3D-Modell [F030]
- Verknüpfen von Informationen mit den einzelnen Räumen [F080]
- Visuell gut erkennbare Darstellung der Geometrie
- Anbindung einer Datenbank für Rauminformationen



Weitere PoCs aus den Anforderungen

Kriterium	Fail-Kriterium	Exit-Kriterium	Fallback-Kriterium	Auswertung
Ermittlung der Route	Eine Ermittlung der kürzestens Route und Weiter-/Ausgabe der nötigen Koordinaten ist nicht	Eine Ermittlung der kürzestens Route und Weiter-/Ausgabe der nötigen Koordinaten ist	Ermittlung einer Route, welche zum Ziel führt, unabhängig von der Optimierung der Länge.	
Selektion von gerenderten 3D-Elementen durch Mausklick	Durch Raycasting an der Position des Mauszeigers kann das gewünschte Objekt nicht zurückgegeben und weiterverwendet werden.	Durch Raycasting an der Position des Mauszeigers kann das gewünschte Objekt zurückgegeben und weiterverwendet werden.	Objekte können nicht interaktiv per Klick selektiert werden. Eine alternative Eingabe über das Interface muss entwickelt werden. Kernfunktionalität gefährdet!	
Visuelle Hervorhebung des selektierten Elements	Das Selektierte Element kann durch Änderung der Materialparameter nicht visuell hervorgehoben werden.	Das Selektierte Element kann durch Änderung der Materialparameter visuell hervorgehoben werden.	Eine andere Möglichkeit zur Hervorhebung muss entwickelt werden.	
Assoziation von Metainformationen zum selektierten Element	Das selektierte Element kann nicht mit entsprechenden Metainformationen assoziiert werden.	Das selektierte Element kann mit entsprechenden Metainformationen assoziiert werden.	Die Funktion kann entfallen, da sie nicht Teil der Kernfunktionalität ist.	
Das Wechseln zwischen zwei Kameras mit unterschiedlicher Perspektive und Bewegungsmöglichkeit	Das Wechseln ist nicht möglich.	Das Wechseln ist möglich.	Verwendung von 2D-Karten anstelle der Orthografischen Vogelperspektive	

Audit 3

Campus Karte - Entwicklungsprojekt WS22

Anforderungen – kann und soll

[F050] Die Anwendung **soll** dem Nutzer die Möglichkeit bieten zwischen einer perspektivischen und orthographischen Kamera zu wählen.

[F055] Die Anwendung **soll** dem Nutzer die Möglichkeit bieten zwischen der Nutzung eines Fahrstuhls oder der Treppen zu wählen.

[F060] Die Anwendung **soll** dem Nutzer die Möglichkeit bieten diese auf jedem Gerät mit einem aktuellen Webbrowser und einer Internetverbindung aufrufen zu können.

[F070] Die Anwendung **soll** dem Nutzer die Möglichkeit bieten Ausgewählte Räume visuell hervorzuheben.

[F080] Die Anwendung **soll** dem Nutzer die Möglichkeit bieten Informationen zu den Räumen anzuzeigen.

Entwicklung der PoCs aus den Anforderungen

Kriterium	Fail-Kriterium	Exit-Kriterium	Fallback-Kriterium	Auswertung
Ermittlung der Route	Eine Ermittlung der kürzesten Route und Weiter-/Ausgabe der nötigen Koordinaten ist nicht möglich	Eine Ermittlung der kürzesten Route und Weiter-/Ausgabe der nötigen Koordinaten ist möglich.	Ermittlung einer Route, welche zum Ziel führt, unabhängig von der Optimierung der Länge.	Ein Abgewandelter A-Star Algorithmus wurde implementiert
Selektion von gerenderten 3D-Elementen durch Mausclick	Durch Raycasting an der Position des Mauszeigers kann das gewünschte Objekt nicht zurückgegeben und weiterverwendet werden.	Durch Raycasting an der Position des Mauszeigers kann das gewünschte Objekt zurückgegeben und weiterverwendet werden.	Objekte können nicht interaktiv per Klick selektiert werden. Eine alternative Eingabe über das Interface muss entwickelt werden. Kernfunktionalität gefährdet!	Ist im Prototypen gelungen.
Visuelle Hervorhebung des selektierten Elements	Das Selektierte Element kann durch Änderung der Materialparameter nicht visuell hervorgehoben werden.	Das Selektierte Element kann durch Änderung der Materialparameter visuell hervorgehoben werden.	Eine andere Möglichkeit zur Hervorhebung muss entwickelt werden.	Ist im Prototypen gelungen.
Assoziation von Metainformationen zum selektierten Element	Das selektierte Element kann nicht mit entsprechenden Metainformationen assoziiert werden.	Das selektierte Element kann mit entsprechenden Metainformationen assoziiert werden.	Die Funktion kann entfallen, da sie nicht Teil der Kernfunktionalität ist.	Ausgabe der Objektinformationen aus Blender ist gelungen.
Das Wechseln zwischen zwei Kameras mit unterschiedlicher Perspektive und Bewegungsmöglichkeit	Das Wechseln ist nicht möglich.	Das Wechseln ist möglich.	Verwendung von 2D-Karten anstelle der Orthografischen Vogelperspektive	Kamera kann durch einen Button gewechselt werden.

Codeinspektion des PoC

Für die bisherigen Artefakte:

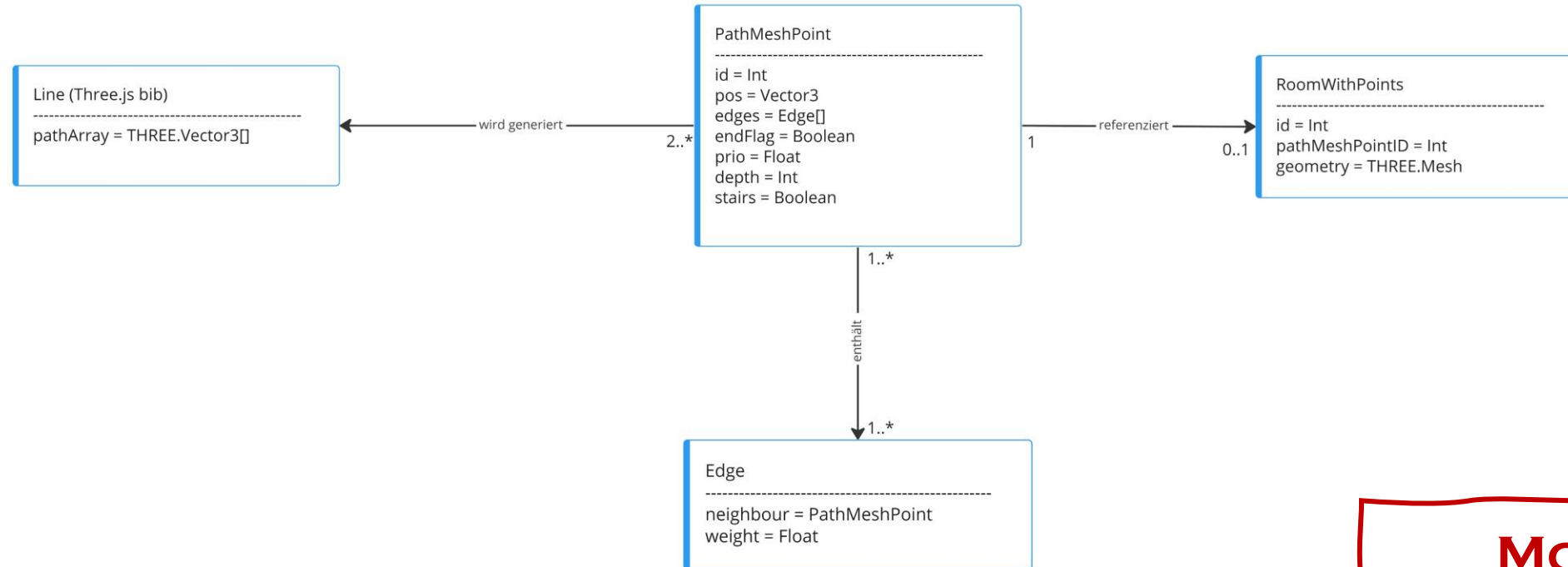
- Austauschen und gemeinsam den Code durchschauen

Codereview der neuen Rapid Prototypes:

- Jeder im Team stellt seinen Code vor, sowie Fragen
- Rückfragen und Kommentare werden von den anderen beiden Teammitgliedern ergänzt

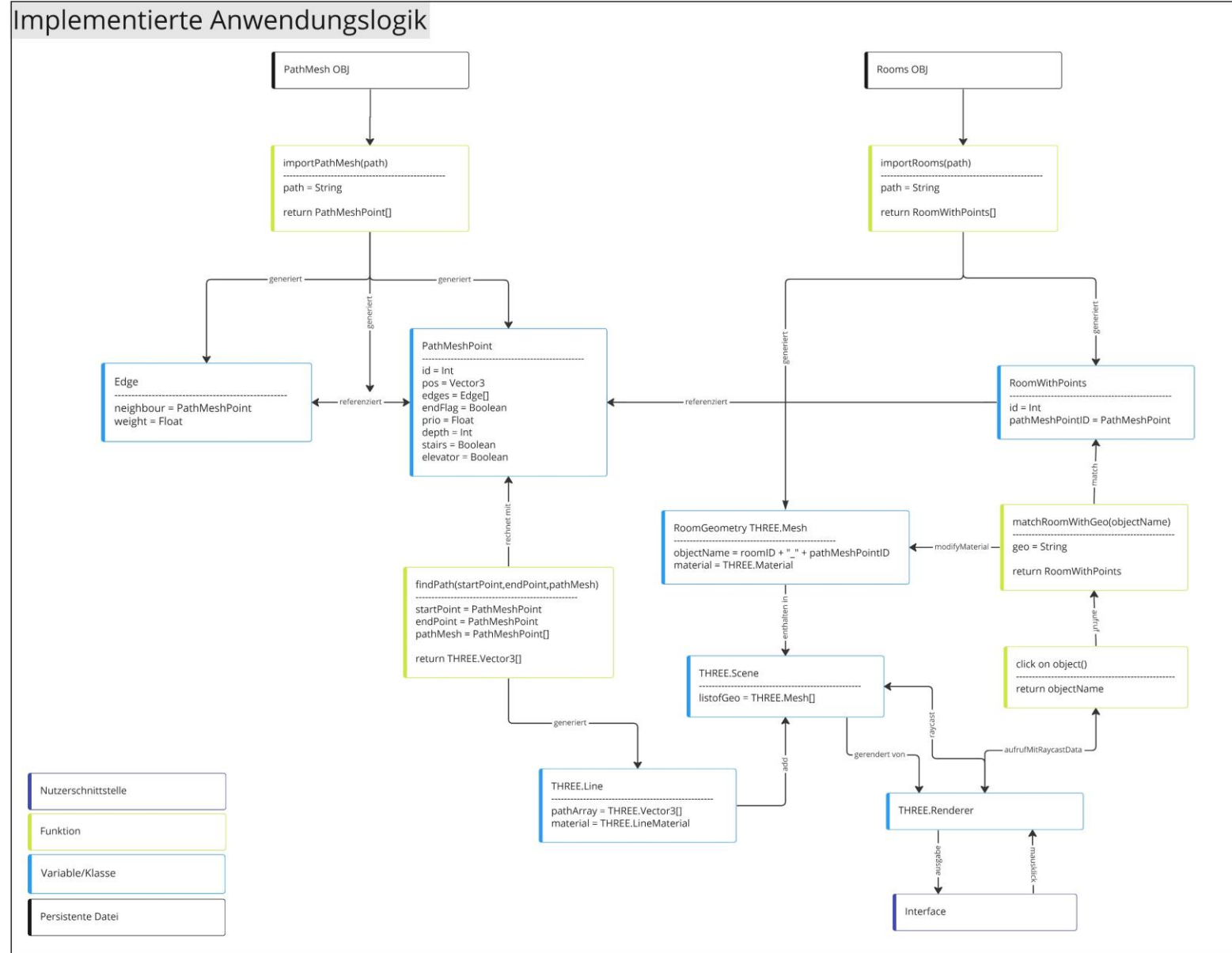
Modellierung der Anwendungslogik - Klassendiagramm

Klassendiagramm

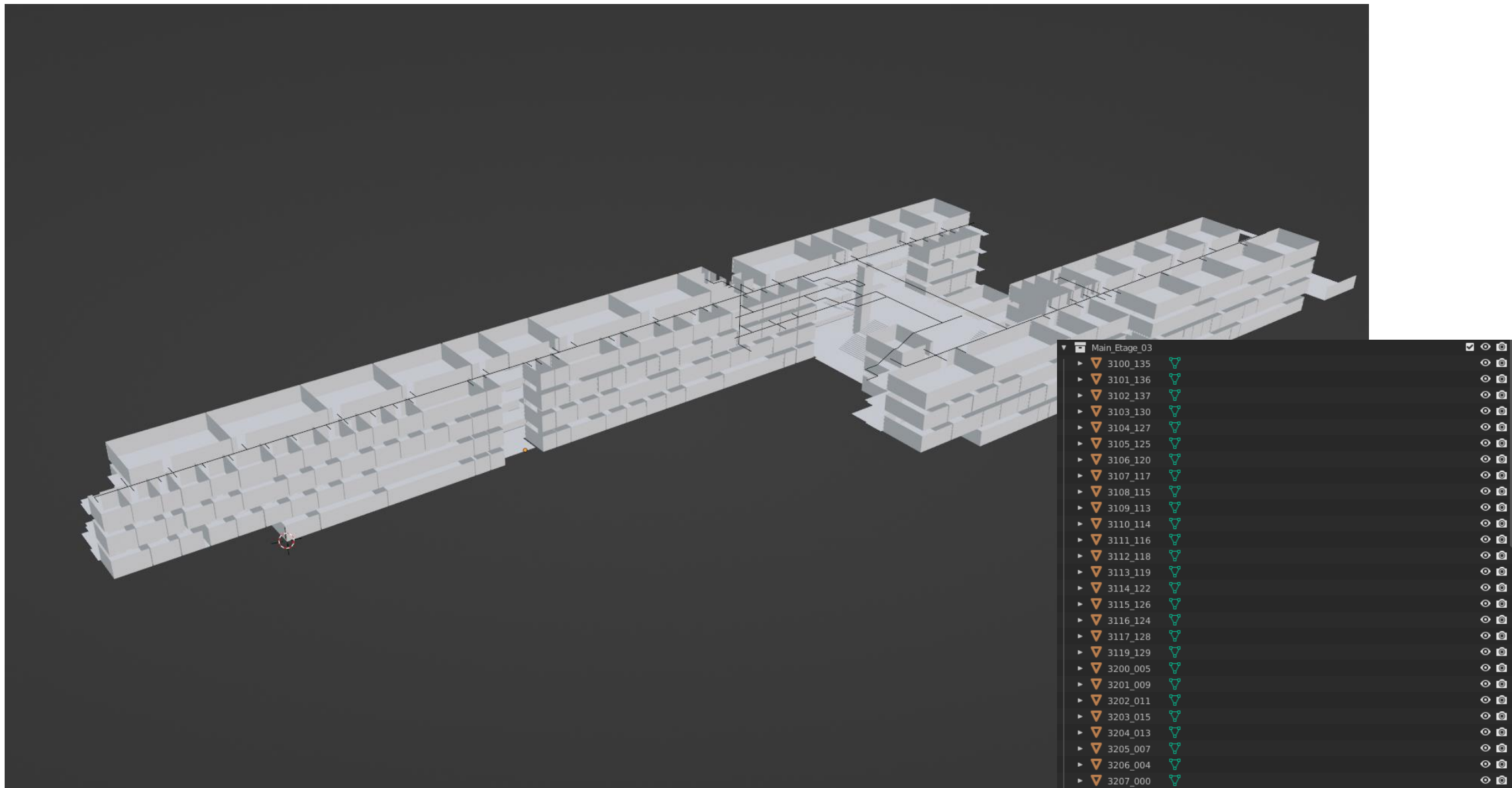


**MODELL
VERWORFEN**

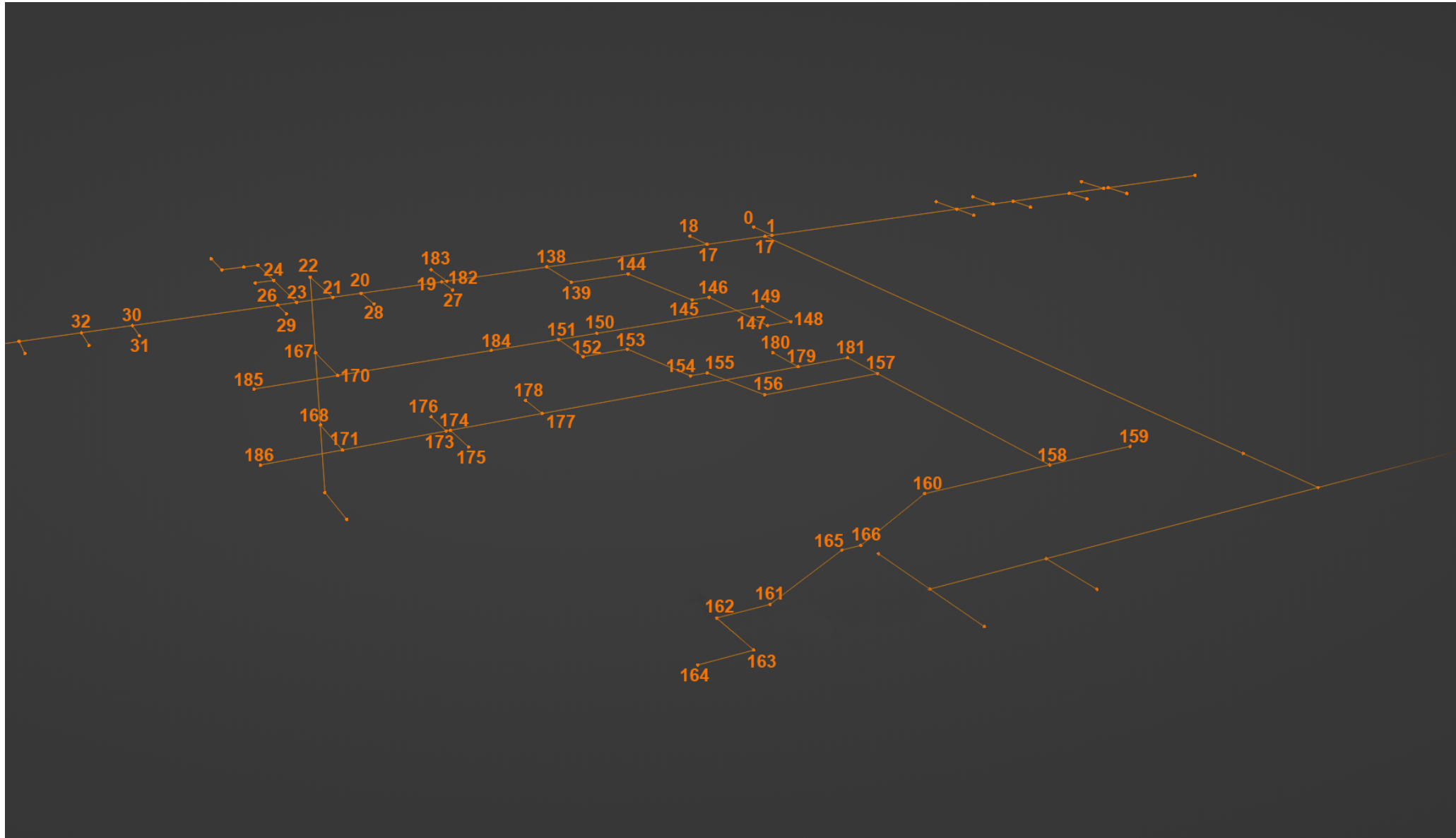
Modellierung der Anwendungslogik



Blender Gebäudemodell und PathMesh



PathMesh mit Vertex-Indizes



```
# Blender 3.3.1
# www.blender.org
mtllib pathMesh_TEST.mtl
o Plane
v 0.000000 0.000000 0.000000
v 1.000000 0.000000 0.000000
v 1.000000 0.000000 -1.000000
v 2.000000 0.000000 0.000000
v 1.000000 0.000000 1.000000
v 1.000000 0.000000 -2.000000
v 2.000000 0.000000 -1.000000
v 2.000000 0.000000 1.000000
v 3.000000 0.000000 1.000000
v 3.000000 0.000000 0.000000
v 3.000000 0.000000 -1.000000
v 2.000000 0.000000 -2.000000
v 4.000000 0.000000 -1.000000
v 3.000000 0.000000 -2.000000
v 4.000000 0.000000 -2.000000
v 5.000000 0.000000 -1.000000
v 5.000000 0.000000 -2.000000
v 6.000000 0.000000 -1.000000
l 2 1
l 3 2
l 2 4
l 2 5
l 6 3
l 4 7
l 5 8
l 8 9
l 4 10
l 7 11
l 12 6
l 11 13
l 14 12
l 15 14
l 13 16
l 17 15
l 16 18
l 3 7
```

[illegible]

```
solid Exported from Blender-3.3.1
facet normal 0.000000 -0.518348 0.855170
outer loop
vertex 6.000000 1.000000 0.000000
vertex 5.000000 1.000000 0.000000
vertex 5.199288 0.031282 -0.587173
endloop
endfacet
facet normal 0.000000 -0.518348 0.855170
outer loop
vertex 6.000000 1.000000 0.000000
vertex 5.199288 0.031282 -0.587173
vertex 6.199288 0.031282 -0.587173
endloop
endfacet
endsolid Exported from Blender-3.3.1
```

```
ply
format ascii 1.0
comment Created by Blender 3.3.1 - www.blender.org
element vertex 6
property float x
property float y
property float z
property float nx
property float ny
property float nz
element face 2
property list uchar uint vertex_indices
end_header
6.000000 1.000000 0.000000 0.986394 0.164399 0.000000
5.000000 1.000000 0.000000 0.980581 0.196116 0.000000
5.000000 1.000000 0.000000 0.000000 -0.518271 0.855217
6.000000 1.000000 0.000000 0.000000 -0.518271 0.855217
5.205523 -0.053939 -0.638698 0.000000 -0.518271 0.855217
6.205523 -0.053939 -0.638698 0.000000 -0.518271 0.855217
4 0 1 2 3
4 3 2 4 5
```

PLY

Import der Daten als PathMesh Point

```
# Blender 3.3.1
# www.blender.org
mtllib pathMesh_TEST.mtl
o Plane
v 0.000000 0.000000 0.000000
v 1.000000 0.000000 0.000000
v 1.000000 0.000000 -1.000000
v 2.000000 0.000000 0.000000
v 1.000000 0.000000 1.000000
v 1.000000 0.000000 -2.000000
v 2.000000 0.000000 -1.000000
v 2.000000 0.000000 1.000000
v 3.000000 0.000000 1.000000
v 3.000000 0.000000 0.000000
v 3.000000 0.000000 -1.000000
v 2.000000 0.000000 -2.000000
v 4.000000 0.000000 -1.000000
v 3.000000 0.000000 -2.000000
v 4.000000 0.000000 -2.000000
v 5.000000 0.000000 -1.000000
v 5.000000 0.000000 -2.000000
v 6.000000 0.000000 -1.000000
l 2 1
l 3 2
l 2 4
l 2 5
l 6 3
l 4 7
l 5 8
l 8 9
l 4 10
l 7 11
l 12 6
l 11 13
l 14 12
l 15 14
l 13 16
l 17 15
l 16 18
l 3 7
```



PathMeshPoint

```
id = Int
pos = Vector3
edges = Edge[]
endFlag = Boolean
prio = Float
depth = Int
stairs = Boolean
```

importPathMesh() Pseudocode

```
function importPathMesh(pfadZumOBJ){
    let objString = ObjInStringKonvertieren(pfadZumOBJ)
    let splitString = objString.inStringArraySplitten(„\n“)
    let punkteUndEdgeData = punktDatenInPathMeshPoints(splitString)
    let pathMeshPoints = punkteMitNachbarnVerbinden(punkteUndEdgeData)
    treppeOderAufzugMarkieren(pathMeshPoints)
    return pathMeshPoints
}

punktDatenInPathMeshPoints(string){
    let points = []
    let edgeData = []
    let index = 0
    for (each in string){
        if (each[0] == „v“){
            points.push(neuer PathMeshPont(index-1, koordinaten))
            index++
        }
        if (each[0] == „l“){
            edgeData.push([ersteVertexID, zweiteVertexID])
        }
    }
    return [points, edgeData]}
```

```
punkteMitNachbarnVerbinden([points, edgeData]){
    for (punkt in points){
        for(daten in edgeData){
            if (punkt.ID in daten gefunden){
                erstelle Edge(NachbarPunkt, entfernungBerechnen)
            }
        }
    }
    return pathMeshPoints
}

treppeOderAufzugMarkieren(punkte){
    wenn (richtungsVektor.Y zu einem Nachbarpunkt != 0){
        wenn (richtungsVektor.Y zu einem Nachbarpunkt == 1 oder -1){
            aufzug = true
        } else{
            treppe = true
        }
    }
}
```

Algorithmus der Pfadgenerierung A-Star

- Algorithmus zur Berechnung des Kürzesten Pfades zwischen zwei Punkten bei positiven Gewichtungen
- Elemente: Startpunkt, Endpunkt, Liste von PathMeshpoints, Position, Letzte Position, EndFlag (markieren der Sackgassen)
Funktionen: calcPrio, sortPrio (Priorisieren der Nachbarn)
- Probleme bei der Umsetzung:
 - Sackgassen werden nicht richtig erkannt
 - Der Weg wird gefunden, da aber so viele Knoten betrachtet werden fehlt die Nachverfolgung des kürzesten

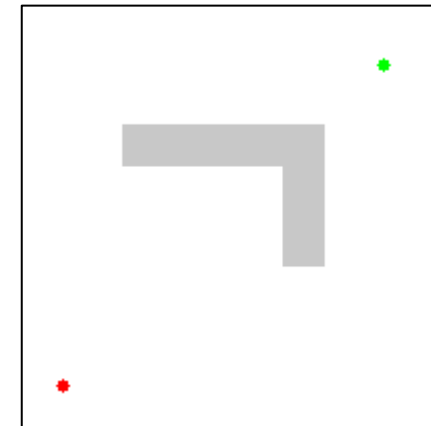


Abb.: A* progress animation

Pathfinding in Three.js

%Evaluation ob Treppe oder Aufzug

if (keine Treppe nehmen) {element.stairs & Endflag auf true setzen}

if (keinen Aufzug nehmen) {element.elevator & Endflag auf true setzen}

startPoint.depth = 0

while (currentLoc !== endPoint){

 %Tiefe der benachbarte Edges um eins erhöhen

 setDepth(currentLoc.edges, currentLoc.depth)

 if (nur eine Edge) {

 if (currentLoc !== endPoint) {

 aktuellen Punkt als Sackgasse makieren, EndFlag= true

 zum vorherigen Punkt zurückkehren

 }

 }else {

 % Kalkulation der Priorität anhand der Entfernung zum
 Endpunkt

 % Sortierung der Edges nach Entfernung zum Endpunkt
 ausführen calcPrio und sortPrio

%Suchen der neuen/nächsten Edge

 newPoint = edges.find{

 erste Edge die kein EndFlag und eine höhere Tiefe hat

 }

 if (newPoint = undefined) {

 endFlag = true &

 edges.find{

 gehe zur vorherigen Edge, erkennbar an der Tiefe-1

 }

 }else{

 currentLoc = newPoint

 }

 }

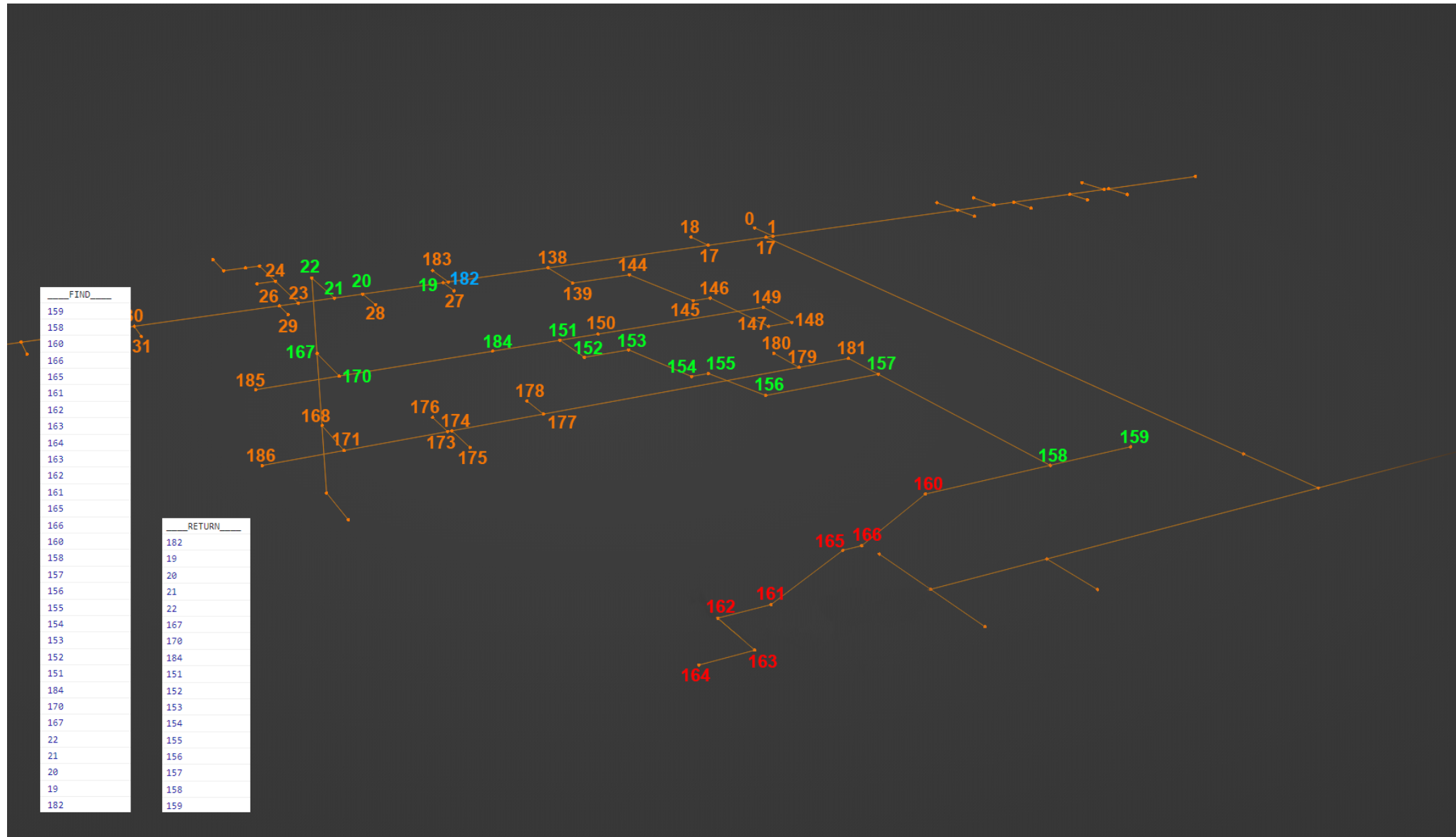
 }Ende while

%kürzesten Pfad ausgeben

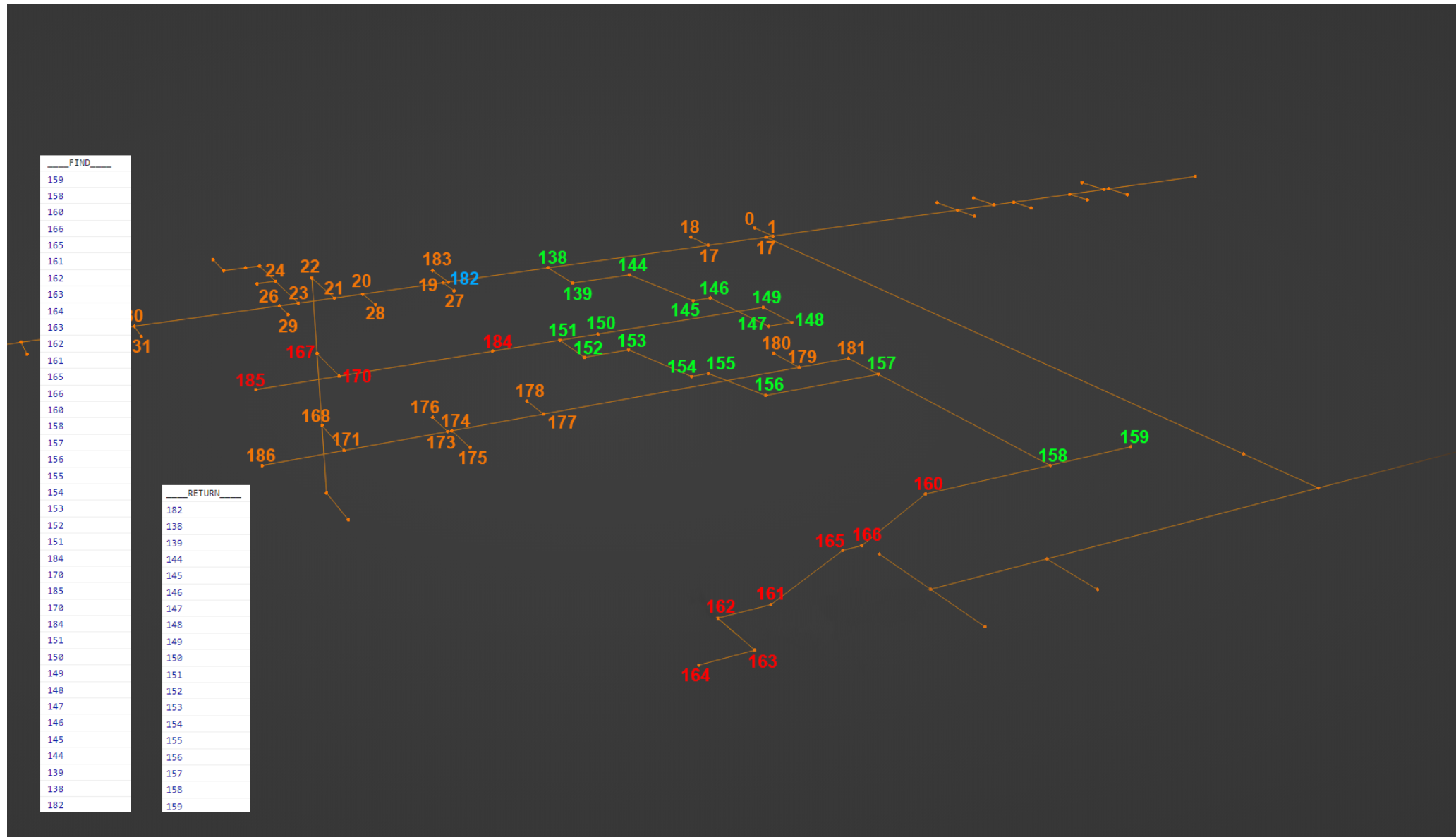
getPath()

zurücksetzen von Prio, Tiefe und EndFlags

findPath(mit Treppen und Fahstühle)



findPath(nur Treppen)



Selektion von Teilen des 3D-Modells

```
function RenderChild(e){
  const [active, setActive] = useState(false)
  const [hovered, setHover] = useState(false)

  return(
    <mesh
      scale = {e.scale}
      material = {e.material}
      geometry = {e.geometry}
      onClick = {(e) => setActive(!active)}
      key = {e.id}
      onPointerOver={(event) => setHover(true)}
      onPointerOut={(event) => setHover(false)}
    >
    <meshStandardMaterial color={active ? 'green' : 'red' && hovered ? 'yellow' : 'red'} />
  </mesh>
  )
}

function AddSingleMesh(){
  const obj = useLoader(OBJLoader, "obj/32xx_full.obj", (loader) => {})

  return obj.children.map(e => {
    return RenderChild(e)
  })
}
```

Warum React Js?



Projektplan

Aufgabe	KW 50 (12.12-18.12.22)	KW 51 (19.12-25.12.22)	KW 52 (26.12-01.01.23)	KW 01 (02.01-08.01.23)	KW 02 (09.01-15.01.23)
Expose Abgabe (31.11.22)					
Domänenmodell					
Zielhierarchie/Plattform festlegen					
Festlegung der Arbeitsweise/Tools					
Spezifikation PoC					
Projektplan					
Auditpräsentation					
Audit 1 (14.11.22)					
Rapid Prototype- Durchführung					
Rapid Prototype- Auswertung					
Anforderungen an finalen Prototyp					
Risikoanalyse					
Auditpräsentation					
Audit 2 (12.12.22)					
Codeinspektion des PoC					
Selektion von Teilen des 3D-Modells					
Pathfinding in Three.js					
Modellierung der Anwendungslogik					
Algorithmus der Pfadgenerierung raussuchen					
Auditpräsentation					
Audit 3 (16.01.23)					
Funktionaler Prototyp (Zusammenführung der Codes)					
Optimierung der Darstellung der 3D-Karte					
Implementierung der Interface Elemente					
Auswertung der gestellten Anforderung					
Wissenschaftliches Poster					
Arbeitsmatrix					
Auditpräsentation					
Audit 4 (27.02.23)					

Legende		Unbearbeitet
		In Bearbeitung
		Fertig gestellt
		Iteration

Nächste Schritte

- Unabhängige Transformation der Kameras
- Live-Generierung des Pfads im zusammengeführten Code
- Implementieren der Userinput/Interface
- Optimierung der Kartengestaltung
- Auf mobilen Geräten testen
- Erstellen des wissenschaftlichen Posters

Ende

Ausführliche Informationen sind in unserem Repo:

[EPW2122ChouliarasBurgdorfWolf/Audit 1.pptx at main · Splashpixmap/EPW2122ChouliarasBurgdorfWolf \(github.com\)](#)