

Campus Karte

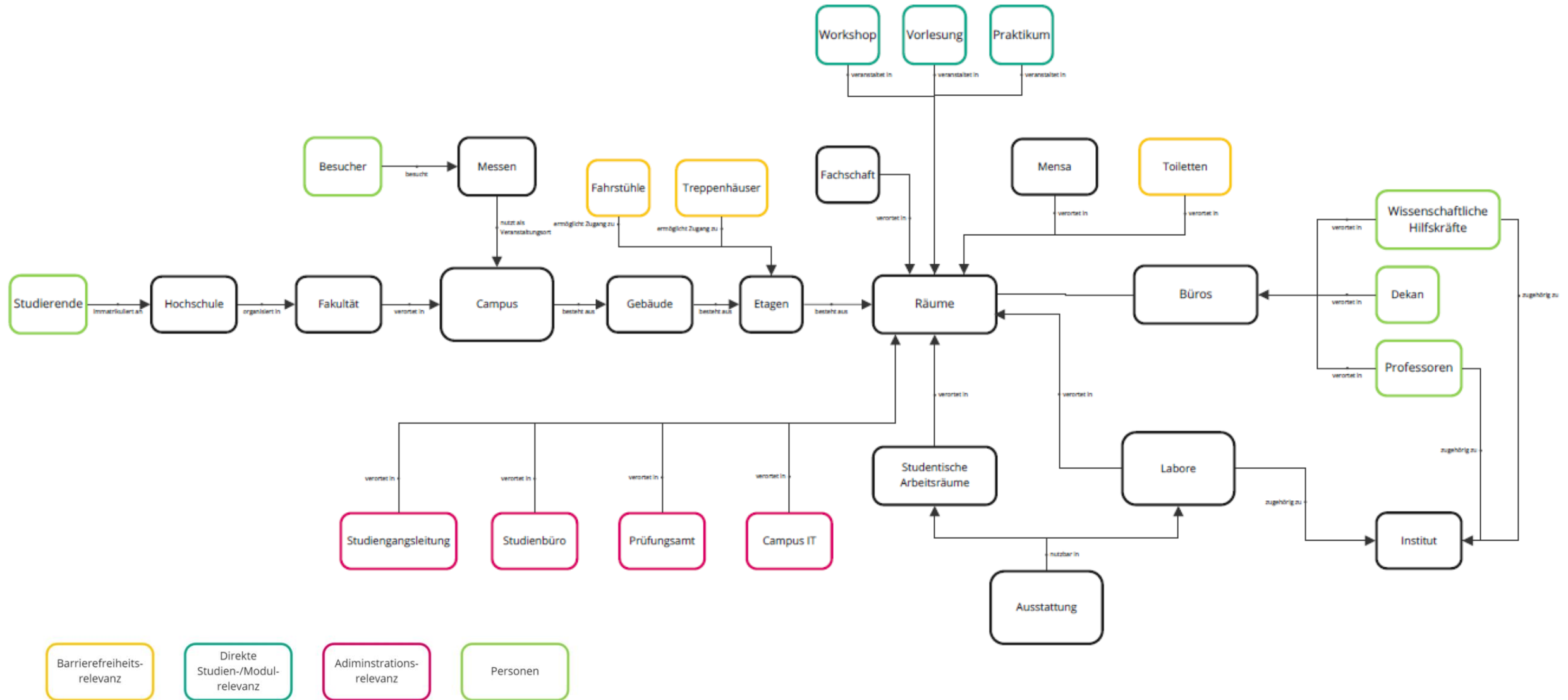
Entwicklungsprojekt WS22

Domenic Wolf

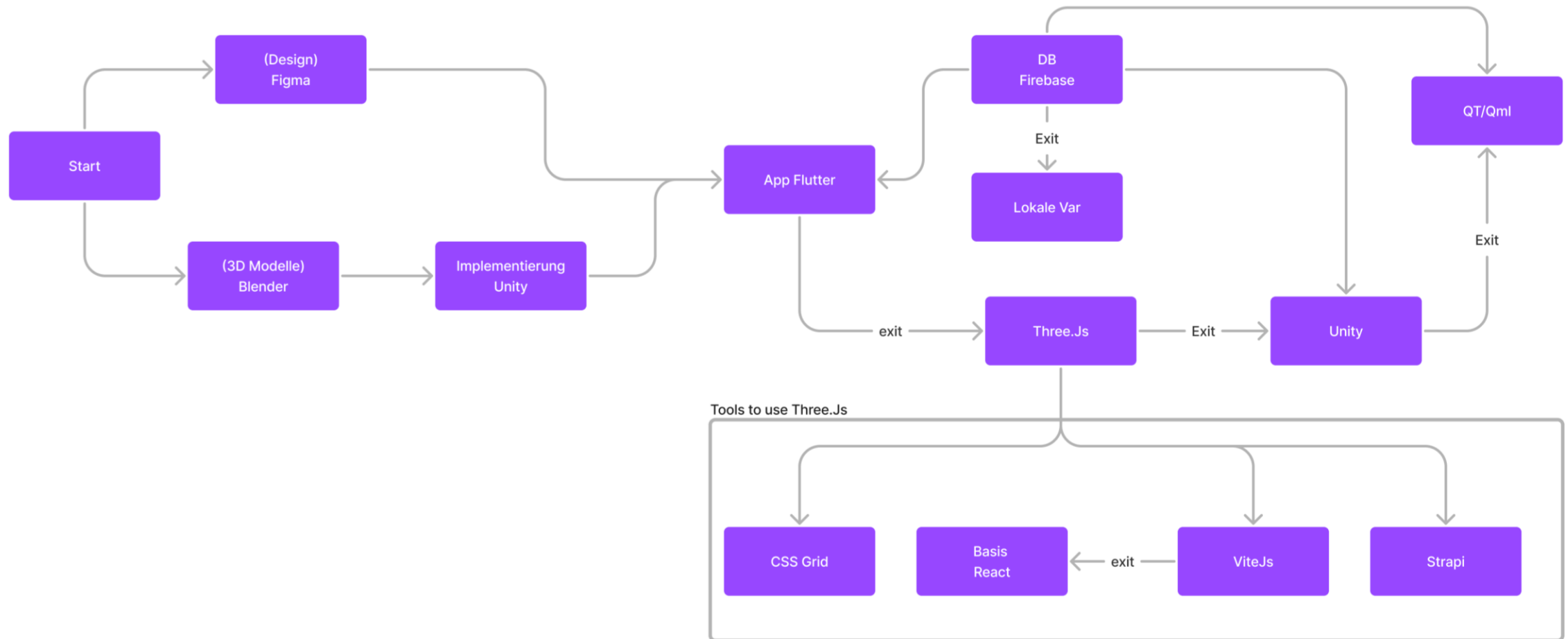
Jens Burgdorf

Anastasia Chouliaras

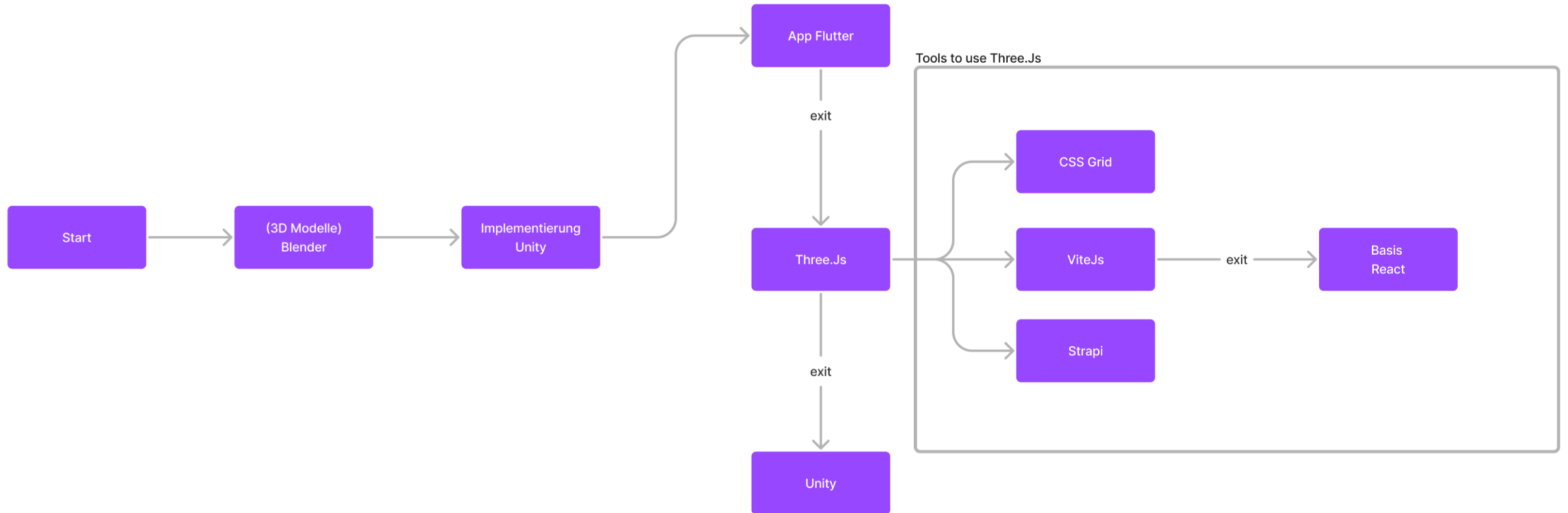
Domänenmodell



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 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.	
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 nicht möglich.	Verwendung von 2D-Karten anstelle der Orthografischen Vogelperspektive	

Ende

Ausführliche Informationen sind in unserem Repo:

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