

AR Flood Hazard Maps

Final Presentation

Frederik Alpers, Lea Plümacher, Marvin Hagemeister

Freie Universität Berlin

January 30, 2026

Project Motivation

Bestehende Informationsmöglichkeiten beschränkt durch:

- Abstrakte Pegelwerte ohne Bezug zur eigenen Umgebung
- Lokale Geländeunterschiede werden nicht berücksichtigt
- Fehlende visuelle Einschätzung des persönlichen Risikos

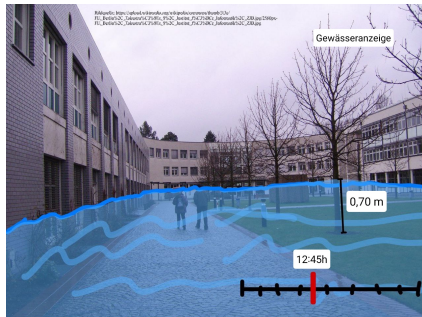


Screenshot: Aktuelle Hochwasserlage Deutschland
(Quelle: hochwasserzentralen.de)

Project Idea

Die Anwendung soll unter anderem folgende Punkte beinhalten:

- AR Handy App
- Interaktive Anzeige des Wasserstandes vor Ort
- Anzeigen von aktuellen Daten und Vorhersagen



Mock-Up

Project Requirements

minimal Requirements

-
-
-
-

additional Requirements

-
-
-
-

Project Requirements

minimal Requirements

- Lauffähige AR App (Android)
-
-
-

additional Requirements

-
-
-
-

Project Requirements

minimal Requirements

- Lauffähige AR App (Android)
- Anzeige eines Flut-Levels (visuell als Plane)
-
-

additional Requirements

-
-
-
-

Project Requirements

minimal Requirements

- Lauffähige AR App (Android)
- Anzeige eines Flut-Levels (visuell als Plane)
- Interface zur Anzeige der Wasserstandshöhe (als Textobjekt)
-

additional Requirements

-
-
-
-

Project Requirements

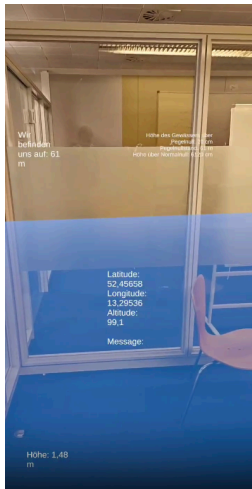
minimal Requirements

- Lauffähige AR App (Android)
- Anzeige eines Flut-Levels (visuell als Plane)
- Interface zur Anzeige der Wasserstandshöhe (als Textobjekt)
- Mock-API (selbes Format wie Real API) für Testing und Demonstration

additional Requirements

-
-
-
-

Implementation Status | Screenshot



minimal viable product

Project Requirements

minimal Requirements

- ✓ Lauffähige AR App (Android)
- ✓ Anzeige eines Flut-Levels (visuell als Plane)
- ✓ Interface zur Anzeige der Wasserstandshöhe (als Textobjekt)
- ✓ Mock-API (selbes Format wie Real API) für Testing und Demonstration

additional Requirements

-
-
-
-

Project Requirements

minimal Requirements

- ✓ Lauffähige AR App (Android)
- ✓ Anzeige eines Flut-Levels (visuell als Plane)
- ✓ Interface zur Anzeige der Wasserstandshöhe (als Textobjekt)
- ✓ Mock-API (selbes Format wie Real API) für Testing und Demonstration

additional Requirements

- Übergang von Mock-API zu echter API
-
-
-

Project Requirements

minimal Requirements

- ✓ Lauffähige AR App (Android)
- ✓ Anzeige eines Flut-Levels (visuell als Plane)
- ✓ Interface zur Anzeige der Wasserstandshöhe (als Textobjekt)
- ✓ Mock-API (selbes Format wie Real API) für Testing und Demonstration

additional Requirements

- Übergang von Mock-API zu echter API
- Verbesserung der Grafik und UI Elemente
-
-

Project Requirements

minimal Requirements

- ✓ Lauffähige AR App (Android)
- ✓ Anzeige eines Flut-Levels (visuell als Plane)
- ✓ Interface zur Anzeige der Wasserstandshöhe (als Textobjekt)
- ✓ Mock-API (selbes Format wie Real API) für Testing und Demonstration

additional Requirements

- Übergang von Mock-API zu echter API
- Verbesserung der Grafik und UI Elemente
- Dokumentation
-

Project Requirements

minimal Requirements

- ✓ Lauffähige AR App (Android)
- ✓ Anzeige eines Flut-Levels (visuell als Plane)
- ✓ Interface zur Anzeige der Wasserstandshöhe (als Textobjekt)
- ✓ Mock-API (selbes Format wie Real API) für Testing und Demonstration

additional Requirements

- Übergang von Mock-API zu echter API
- Verbesserung der Grafik und UI Elemente
- Dokumentation
- Zeitstrahl

Current Implementation Status | Screenshot 2



Finale Implementierung mit simuliertem Wasserstand

Current Implementation Status | Screenshot 2



Finale Implementierung mit simuliertem Wasserstand

Project Requirements

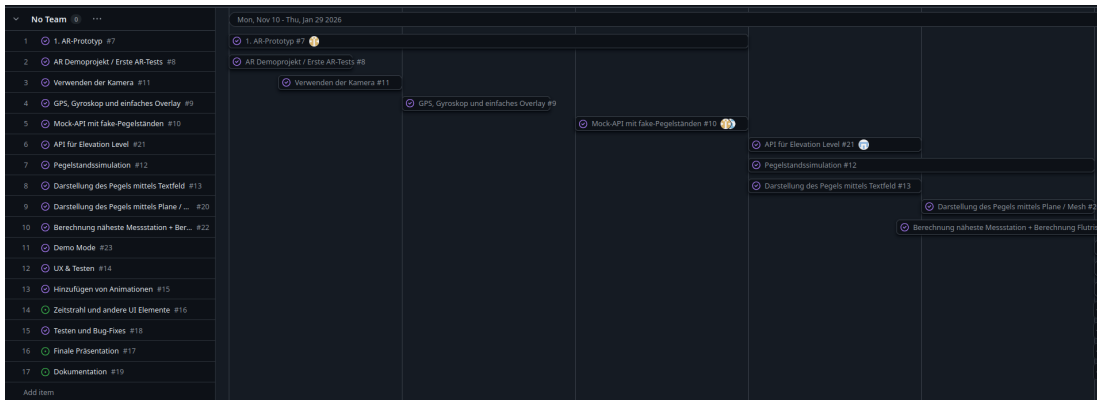
minimal Requirements

- ✓ Lauffähige AR App (Android)
- ✓ Anzeige eines Flut-Levels (visuell als Plane)
- ✓ Interface zur Anzeige der Wasserstandshöhe (als Textobjekt)
- ✓ Mock-API (selbes Format wie Real API) für Testing und Demonstration

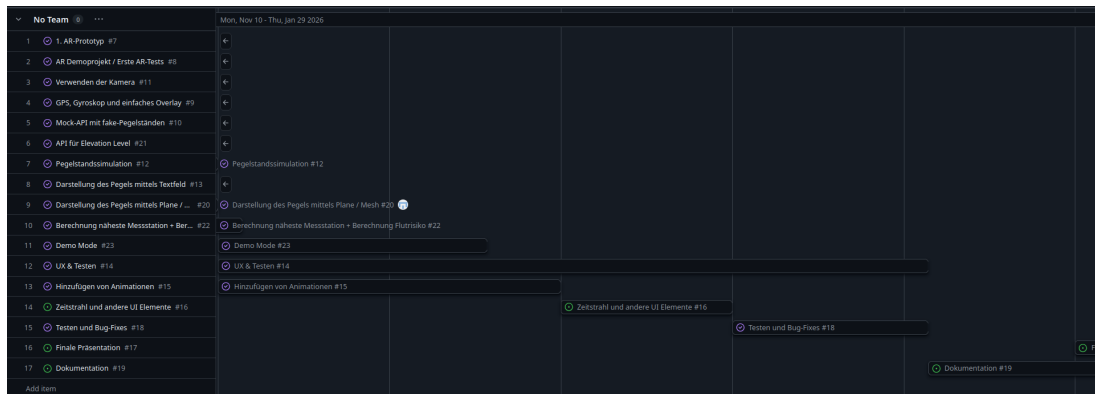
additional Requirements

- ✓ Übergang von Mock-API zu echter API
- Verbesserung der Grafik und UI Elemente
- Dokumentation
- Zeitstrahl

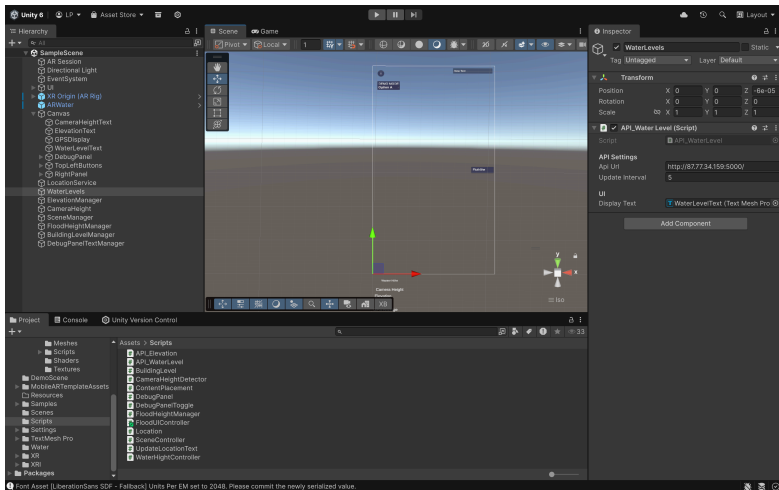
Progress | Timeline part 1/2



Progress | Timeline part 2/2

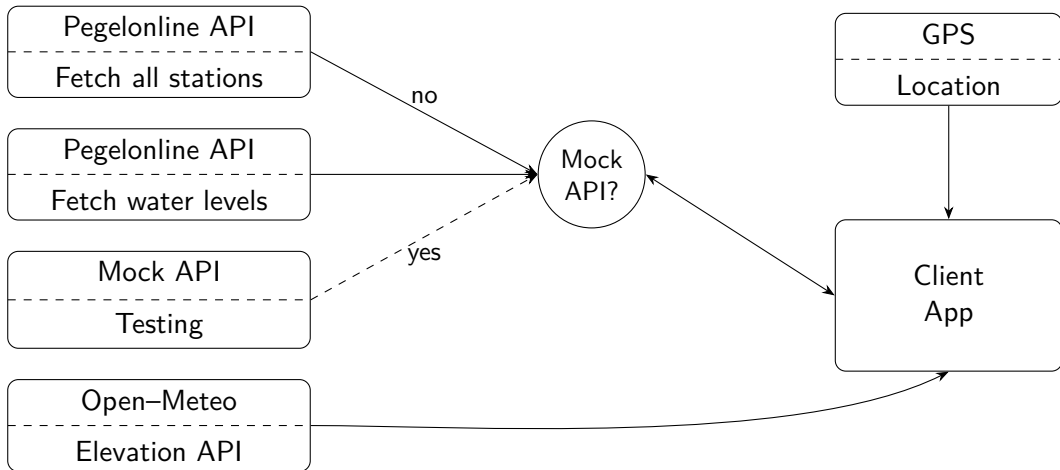


Technical implementation | Software Used



Screenshot of Unity

Technical Implementation | Diagram



Conclusion

Projektergebnisse

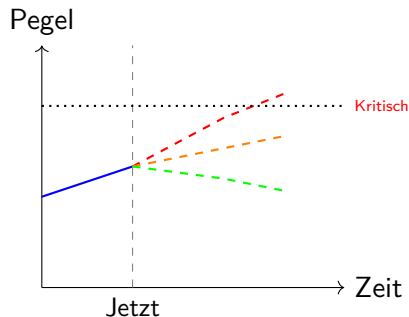
- ✓ Funktionale AR-App für Android entwickelt
- ✓ Echtzeit-Visualisierung von Hochwasserdaten
- ✓ Integration echter APIs (Pegelonline, Open-Meteo)
- ✓ Intuitive Benutzeroberfläche mit AR-Overlay

Predictive Flood Modeling

- Integration von Wettervorhersagen
- Zeitbasierte Animation des Pegelanstiegs/-rückgangs
- Frühwarnsystem bei kritischen Schwellenwerten

Erwarteter Nutzen:

- Bessere Vorbereitung für Anwohner
- Rechtzeitige Evakuierung möglich
- Visualisierung von möglichen Szenarien



— aktuell - - Worst Case - - Wahrscheinlich - - Best Case

Konzept: Vorhersagemodell mit verschiedenen Szenarien

Offline-Modus

- Daten-Caching für Gebiete ohne Netz
- Lokale Datenspeicherung
- Synchronisation bei Verbindung

Evakuierungsrouten

- Integration mit Navigation
- Sichere Wege zu Sammelstellen
- Echtzeit-Aktualisierung bei Überflutung

Optimierung

- Performance-Optimierung (Akku, Rendering)
- Wasser-Animation (Richtung, Geschwindigkeit)
- auf mehr Geräten testen

Reflection | Lessons Learnt

- Unity
- AR
- Als Team zusammenarbeiten

Reflection | What worked well, what did not work?

- ✓ Teamorganisation
- ✓ Entwicklung mit einer Mock API
- ✓ AR (Die vorgegebene Implementierung hat gut funktioniert)
- ✓ APIs
- ✗ Unity Version Control
- ✗ Unity Editor

Reflection | What would we do different next time?

- WebAR benutzen, wenn möglich
- Version Controll über Git

Fragen?