W3WI_AM302 - Fortgeschrittene Systementwicklung

Dozent: Prof. Dr. Michael Eichberg
Kontakt: michael.eichberg@dhbw.de, Raum 149B
Version: 24AMA - 12. Februar 2026

Inhalte 1. Semester

- Entwicklung von verteilten Anwendungen - Grundlagen
  - Linux/Server Handhabung
  - Buildprozesse
  - Versionsverwaltung
  - Container

Bereits bekannt sein sollte:

- Programmiersprache Go aus dem ersten Semester
- Programmiersprache Java aus dem zweiten Semester
- Grundkenntnisse in HTML und CSS sollten parallel in der Web-Programmierung gelehrt werden.
  Was wird noch gelehrt?
- Datenbanken findet auch parallel statt.
- GitHub ist aus dem ersten Semester bekannt.
  Ist GIT als Versionskontrollsystem erklärt worden?

Inhalte 2. Semester

Durchführung eines Entwicklungsprojekts in kleine(re)n Gruppen.

Prüfungsleistung

Prüfungsleistung: Portfolio (insgesamt 120 Punkte)

1. Semester: Kurztest mit 20 Minuten **mit 20 Punkten**

2. Semester: Vorträge, Code Reviews, Ausarbeitungen, Projekt **mit 100 Punkten**

Ablauf

Hintergrund

- Modul: 55 VL
- Modul 5 ECTS

1. Semester: 16 VL

- 01. Sep 2025
- 03. Sep 2025
- 15. Sep 2025
- 06. Okt 2025 (20 Minuten Kurztest)

1. Semester: 39 VL

- 18. Feb 2026 von 09:30 bis 12:45 (4VL)
- 25. Feb 2026 von 09:30 bis 12:45 (4VL)
- 04. März 2026 von 09:30 bis 12:4 (4VL)
- 11. März 2026 von 09:30 bis 12:45 (4VL)
- 18. März 2026 von 09:30 bis 12:45 (4VL)
- 25. März 2026 von 09:30 bis 12:45 (4VL)
- 01. April 2026 von 09:30 bis 12:45 (4VL)
- 08. April 2026 von 09:30 bis 12:45 (4VL)
- 15. April 2026 von 09:30 bis 12:45 (4VL)
- 22. April 2026 von 09:30 bis 12:00 (3VL)

Aufgabenstellung 2. Semester

Gegenstand der Portfolioaufgabe ist die Entwicklung eines verteilten Web-Spiels für mehrere Spieler.

Folgende Anforderungen sind zu erfüllen:

- Ihr Projekt muss am Ende spielbar sein in dem Sinne, dass alle Kursteilnehmer die Möglichkeit haben gleichzeitig zu spielen; ggf. in mehreren Kleingruppen gegeneinander.
- Das Spiel muss rundenbasiert sein.

> **⚠ Achtung!**
> Echtzeitspiele sind aufgrund der damit verbundenen Komplexität ausgeschlossen.

- Sie können sich - müssen aber nicht - von klassischen Brettspielen oder Kartenspielen inspirieren lassen. Es können aber auch neue Spiele entwickelt werden.

  D. h. Die Wahl des Spieles ist weitgehend frei. Poker und Uno sind jedoch ausgeschlossen; triviale Spiele (z. B. Hangman oder Tic-Tac-Toe) sind auch ausgeschlossen. Darüber hinaus darf kein Spiel zweimal entwickelt werden. *Sprechen Sie sich daher frühzeitig mit Ihren Kommilitonen ab, um Überschneidungen zu vermeiden.*
- Die Entwicklung erfolgt in Gruppen von 3 Personen und muss immer einen Webclient und eine Serverkomponente umfassen.
- Die eingesetzten Technologien sind auf JavaScript, CSS und HTML sowie ggf. Python oder Java für die Serverseite beschränkt. Ich empfehle einen JavaScript/TypeScript Einsatz auf beiden Seiten. Frameworks (z. B. Express.js und Socket.IO), um ggf. die Kommunikation zwischen Web-Client und Server zu vereinfachen, dürfen eingesetzt werden.
- Die Webanwendung muss responsive sein und braucht nur auf den neuesten Browsern laufen: Safari, Chrome und Firefox. Die neuesten Web-Technologien sollen verwendet werden. Responsive bedeutet hier, dass die Anwendung mindestens auf Tablets und Desktop-Computern spielbar sein muss.
- Das Spiel sollte Cheat-Prevention Mechanismen enthalten, damit es nicht möglich ist, durch Manipulation der Webanwendung zu betrügen.
- Ein komplexes Rollen-/Sicherheitsmodell ist nicht erforderlich. Es muss jedoch möglich sein, dass Spieler sich mit einem Benutzernamen anmelden und dass die Spieler während des Spiels identifizierbar sind.
- Bewertet wird die Qualität der Software, die Qualität der Dokumentation, die Qualität der Präsentationen und die Einhaltung der Anforderungen.

Projektergebnisse (Grobübersicht)

1. Die gehaltenen Präsentationen (sowohl bzgl. des Projekts als auch die fachlichen Präsentationen) inkl. unterschriebener und abgearbeiteter Checkliste
2. Die Code Reviews
3. Code und Anwendung
   - Lauffähige Webanwendung mit allen Kernfunktionen
   - README mit Bau- und Installationsanleitung
   - API-Spezifikation bzw. Dokumentation der Schnittstellen zwischen Client und Server
4. Dokument, das dokumentiert welche Team-Mitglieder welche Teile bearbeitet haben und mit welchem Anteil. **Wenn Sie als Team bewertet werden möchten, dann teilen Sie mir dies bitte mit und vermerken es explizit in dem Dokument.**
5. Dokument bzgl. KI-Einsatz: Wo wurde welche KI wie eingesetzt. Wie war Ihre Erfahrung.

Zur Verfügung gestellt wird

Zugriff auf einem Server, der aus dem UNI Netz (ggf. mittels VPN) erreichbar ist. Gruppenzugänge werden noch eingerichtet.

Vortragsthemen

- Alle Vorträge behandeln Themen, die für das Projekt direkt relevant sind.
- Die Vorschläge zu den konkreten Inhalten der Vorträge sind als Anregung zu verstehen und kleinere Abweichungen sind ggf. möglich.
- Führen Sie immer in das Thema ein und geben Sie auch praktische Beispiele, damit die Relevanz für die Projektarbeit klar wird. D. h. die Vorträge sollen einen hohen „Hands-on“-Anteil haben.
- **Sprechen Sie sich ggf. untereinander ab, damit die Vorträge nicht inhaltlich überlappen.**
- Beachten Sie die Hinweise zur Vortragsgestaltung.
- Bitte beachten Sie die Zeitvorgaben.

> ⚠ Achtung!
>
> Vorträge sind bis 6 Uhr Morgens am jeweiligen Vortragstag als PDF-Datei in Moodle hochzuladen.

Vortragsthemen - Frontend-Entwicklung

CSS Grid Layout für Spieloberflächen

Personen: 1 Person

Dauer: 10 Minuten

Inhalte:

- CSS Grid Grundlagen und Terminologie
- Responsive Spielfelder mit Grid
- Praktische Beispiele: Kartenspiele, Brettspiele
- Grid vs. Flexbox: Wann was verwenden?

CSS Animationen und Transitionen für Spiele

Personen: 2 Personen

Dauer: 20 Minuten

CSS Transitions:

- Grundlagen von Transitions
- Animierte Spielelemente (Kartenbewegungen, Figurenbewegungen)
- Performance-Aspekte
- Best Practices für flüssige Animationen

CSS Keyframe Animations:

- @keyframes Syntax
- Animation Timing Functions
- Komplexe Animationsabläufe
- Beispiele: Würfelwurf, Kartenausteilen

CSS Custom Properties und Design Systems

Personen: 1 Person

Dauer: 10 Minuten

Inhalte:

- CSS Variablen und CSS Properties (`@property`) Grundlagen
- Scope und Vererbung
- Performance-Überlegungen
- Design Tokens und konsistente Gestaltung

Responsive Design für Spiele

Personen: 1 Person

Dauer: 10 Minuten

Inhalte:

- Mobile-First vs. Desktop-First
- Media Queries für verschiedene Geräte
- Viewport-Units und Container Queries
- Touch vs. Maus-Interaktion
- Responsive Spielfelder und UI-Elemente

CSS Theming und Farbsysteme

Personen: 1-2 Personen
Dauer: 10-20 Minuten

Moderne Color Spaces:

- sRGB, Display P3, Lab, LCH
- color-mix() Funktion
- color-contrast() / contrast-color()
- Barrierefreiheit bei Farbwahl

Light/Dark Themes:

- light-dark() Funktion
- color-scheme Property
- Theme-Switcher implementieren
- CSS Custom Properties für Themes

Moderne UI-Komponenten mit HTML/CSS

Personen: 1 Person
Dauer: 10 Minuten

Inhalte:

- Dialog/Modal für Spielregeln und Einstellungen
- Custom Buttons und Controls (`<button>`, etc.)
- Tooltips und Notifications (`popover`, etc.)
- Loading States und Progress Bars

Vortragsthemen - JavaScript für interaktive Spiele

Grundlagen von modernem JavaScript (ES6+)

Personen: 2 Personen
Dauer: 20 Minuten

ES6+ Grundlegende Features:

- Arrow Functions
- Destructuring
- Spread/Rest Operator
- Default Parameters

ES6+Erweiterte Features:

- Template Literals
- Promises und async/await
- Modules (import/export)
- Array-Methoden (map, filter, reduce)

JavaScript Event Handling für Spiele

Personen: 1 Person
Dauer: 10 Minuten

Inhalte:

- Event Delegation
- Keyboard Events für Spiele
- Touch Events für Mobile
- Event-Loop und Performance

DOM-Manipulation und Virtual DOM Konzepte

Personen: 1 Person
Dauer: 10 Minuten

Inhalte:

- Effiziente DOM-Updates
- DocumentFragment
- Template-Element
- Shadow DOM Grundlagen
- Performance-Best-Practices

JavaScript State Management

Personen: 1 Person
Dauer: 10 Minuten

Inhalte:

- Lokales State Management
- State-Patterns für Spiele
- Immutability
- State Synchronisation Client/Server
- Observer Pattern

JavaScript Error Handling und Debugging

Personen: 1 Person
Dauer: 10 Minuten

Inhalte:

- try/catch Best Practices
- Browser DevTools für Debugging
- Error Boundaries
- Logging-Strategien

Browser Storage APIs

Personen: 1 Person
Dauer: 10 Minuten

Inhalte:

- LocalStorage vs. SessionStorage
- IndexedDB Grundlagen
- Speichern von Spielständen
- Cache API
- Privacy-Aspekte

Vortragsthemen - Client-Server-Kommunikation

API Design

Personen: 1 Person
Dauer: 10 Minuten

Inhalte:

- HTTP Methoden (GET, POST)
- Status Codes
- URL-Design für Spiel-APIs
- Request/Response Format (JSON)
- API-Dokumentation

Fetch API und AJAX

Personen: 1 Person
Dauer: 10 Minuten

Inhalte:

- Fetch API Grundlagen
- Promises und async/await mit Fetch
- Request/Response Handling
- Error Handling
- CORS-Problematik

WebSockets Grundlagen

Personen: 1 Person
Dauer: 10 Minuten

Inhalte:

- WebSocket Protokoll
- Native WebSocket API
- Connection Lifecycle
- Heartbeats und Reconnection
- Wann WebSockets vs. HTTP

Socket.IO für Bidirektionale Echtzeitkommunikation

Personen: 2 Personen
Dauer: 20 Minuten

Socket.IO Basics

- Installation und Setup
- Events und Rooms
- Namespaces
- Broadcasting
- Verbindungsmanagement

Socket.IO für Multiplayer-Spiele

- Spieler-Räume verwalten
- Turn-based Game Logic
- Synchronisation von Spielzuständen
- Latenz-Handling
- Reconnection-Strategien

> Bemerkung
>
> Diese Talks sollten sich insbesondere auf die Verwendung von Socket.IO für die Entwicklung von Multiplayer-Spielen konzentrieren.

Express.js Grundlagen

Personen: 2 Person

Dauer: 20 Minuten

Express.js Basics

- Routing
- Middleware-Konzept
- Request/Response Handling
- Static File Serving
- Errorhandling

Express.js für Spiele-Backend

- Body Parsing
- CORS-Konfiguration
- API-Struktur
- Integration mit Socket.IO

> Bemerkung
>
> Diese Talks sollten sich insbesondere auf die Verwendung von Express.js für die Entwicklung von Multiplayer-Spielen konzentrieren. Es sollten konkrete Beispiele gegeben werden

API-Testing und Dokumentation

Personen: 1 Person

Dauer: 10 Minuten

Inhalte:

- Postman/Thunder Client
- Unit Tests für APIs
- Automatisierte Tests
- Mock-Server/-Daten

Vortragsthemen - Sicherheit

Cheat Prevention

Personen: 1 Person

Dauer: 10 Minuten

Inhalte:

- Häufige Cheat-Vektoren in Web-Spielen
- Server-Authority Prinzip
- Client-Side Validation vs. Server-Side Validation
- Input Sanitization
- Rate Limiting

Vortragsthemen - Software-Architektur

Software-Architektur für verteilte Anwendungen

Personen: 1 Person

Dauer: 10 Minuten

Inhalte:

- Client-Server Architektur
- MVC/MVVM Patterns
- Separation of Concerns
- Modulare Struktur
- Code-Organisation

Vortragsthemen - Softwareengineering Praktiken

Testing für Web-Anwendungen

Personen: 1 Person

Dauer: 10 Minuten

Inhalte:

- Unit Testing (Jest, Vitest)
- Integration Testing
- E2E Testing Grundlagen
- Mocking und Stubs

Build-Tools und Bundling

Personen: 1 Person

Dauer: 10 Minuten

Inhalte:

- Vite/Webpack Grundlagen
- Module Bundling
- Code Splitting
- Minification und Optimization
- Development vs. Production Builds

Vortragsthemen - Barrierefreiheit

Accessibility (a11y) für Spiele

Personen: 1 Person

Dauer: 10 Minuten

Inhalte:

- WCAG Grundlagen
- ARIA Roles und Attributes
- Keyboard Navigation
- Screen Reader Support
- Color Contrast und Lesbarkeit

1. Ablauf 2. Semester (Wann passiert was?)

18. Feb 2026 von 09:30 bis 12:45 (4VL)

- Besprechung der Projektaufgabe
- Besprechung des Ablaufs des Semesters
- Vortragsthemenvergabe
- Aufteilung der Teams

25. Feb 2026 von 09:30 bis 12:45 (4VL)

- Elevator Pitch des Spielkonzepts
- Vorträge
- Beantwortung von Fragen/Teamindividuelle Beratung

4. März 2026 von 09:30 bis 12:4 (4VL)

- Vorträge
- Beantwortung von Fragen/Teamindividuelle Beratung

11. März 2026 von 09:30 bis 12:45 (4VL)

- Vorträge
- Beantwortung von Fragen/Teamindividuelle Beratung

18. März 2026 von 09:30 bis 12:45 (4VL)

- Vorträge
- Präsentation des Spielkonzepts (pro Team ca. 10 Minuten)

  D.h. Fake Screenshots, Mockups, Storyboard, etc. - alles ist erlaubt, um die Idee zu vermitteln. Es muss jedoch klar werden, um was für ein Spiel es sich handelt, wie es gespielt wird, in welcher Weise die Spieler miteinander interagieren, etc.
- Beantwortung von Fragen/Teamindividuelle Beratung

25. März 2026 von 09:30 bis 12:45 (4VL)

- Vorführung des aktuellen Stands der Anwendung durch jedes Team (ca. 10 Minuten pro Team)

  (Keine explizite Abgabe erforderlich.)
- Feedbackrunde und Fragen

1. April 2026 von 09:30 bis 12:45 (4VL)

- Einführung in strukturierte Code Reviews
- Jede Gruppe erhält eine Einführung in den Code zweier anderer Teams (je Partnerteam ca. 45 Minuten)

  (Den Gruppen, die die Code Reviews durchführen, wird Zugriff auf den Code eingerichtet.)
- Jede Gruppe präsentiert informell ihre allerersten Erkenntnisse/Eindrücke aus dem Code Review (ca. 2-3 Minuten pro Review)

  (Die Sessions von jeweils 45 Minuten dienen dazu, dass Sie den Code der anderen Teams kennenlernen und sich auf das Code Review vorbereiten können; die eigentlichen Code Reviews finden außerhalb der Vorlesungszeit statt.)
- Ich stehe für allg. und gruppenspezifische Fragen zur Verfügung.

8. April: 7:00 (Ereignis)

- Abgabe der dokumentierten Code-Review Ergebnisse an die Teams und „an mich“
- Abgabe der Präsentationen zu den Code Reviews
- Abgabe über Moodle als 4 PDF-Dateien mit folgenden Namensschema:
  - CodeReviewDokumentation_TeamX_TeamY.pdf (Code Review von Team X für Team Y)
  - CodeReviewPräsentation_TeamX_TeamZ.pdf (Code Review von Team X für Team Z)

  (X,Y und Z sind durch die jeweiligen Teambezeichner zu ersetzen.)

8. April 2026 von 09:30 bis 12:45 (4VL)

- Präsentation der Ergebnisse des Code Reviews

  10 Minuten pro Code Review; d.h. 20 Minuten pro Team.

15. April 2026 von 09:30 bis 12:45 (4VL) **Online**

- Finale Besprechung der Anforderungen an die Abgaben
- Teamindividuelle Beratung

21. April 2026 7:00 (Ereignis)

- Abgabe aller Projektergebnisse über Moodle

22. April 2026 von 09:30 bis 12:00 (3VL)

- Spielzeit

  Dauer ca. 15 Minuten pro Team.

Aufteilung der Vortragsthemen

ID	Titel	Datum
1	CSS Grid Layout für Spieloberflächen	25. Februar
2	CSS Animationen und Transitionen für Spiele	25. Februar
3	CSS Custom Properties und Design Systems	25. Februar
4	Responsive Design für Spiele	25. Februar
5	Moderne Color Spaces	25. Februar
6	Light/Dark Themes	25. Februar
7	Moderne UI-Komponenten mit HTML/CSS	25. Februar
8	ES6+ Grundlegende Features	4. März
9	ES6+ Erweiterte Features	4. März
10	JavaScript Event Handling für Spiele	4. März
11	DOM-Manipulation und Virtual DOM Konzepte	4. März
12	JavaScript State Management	4. März
13	Browser Storage APIs	4. März
14	API Design	4. März
15	Fetch API und AJAX	4. März
16	JavaScript Error Handling und Debugging	11. März
17	WebSockets Grundlagen	11. März
18	Socket.IO für Bidirektionale Echtzeitkommunikation	11. März
19	Express.js Grundlagen	11. März
20	API-Testing und Dokumentation	11. März
21	Cheat Prevention	11. März
22	[Software-Architektur für verteilte Anwendungen]	11. März
23	Testing für Web-Anwendungen	18. März
24	Build-Tools und Bundling	18. März
25	[Accessibility (a11y) für Spiele]	18. März

Teams

- die Teamgröße beträgt 3 Studierende
- die Aufteilung der Aufgaben im Team erfolgt selbstorganisiert, muss aber dokumentiert werden
- Eine grobe Aufteilung entlang der Dimensionen: Frontend, Backend und querschneidende Belange ist naheliegend, aber nicht zwingend erforderlich; auf jeden Fall muss *jedes* Teammitglied in der Lage sein jeden Teil der Anwendung zu verstehen und bei Bedarf zu bearbeiten.

Bewertung

- **[20P] Vorträge**
  - [19P] Inhaltliche Qualität, Verständlichkeit, optische Präsentation, Persönliches Auftreten
  - [1P] Abgabe der Checkliste (als PDF) unterschrieben und abgearbeitet
- **[20P] Code Reviews**

  Pro Code Review (insgesamt 2 pro Team) werden folgende Kriterien bewertet:
  - [1P] Kurzpräsentation des ersten Eindrucks
  - [6P] Qualität der Code Reviews
  - [3P] Qualität der Präsentation der Code Reviews
- **[56P] Spiel**
  - [5P] Präsentation des Spielkonzepts (Fake Screenshots, Mockups, Storyboard, etc.)
  - [3P] Vorführung des aktuellen Stands der Anwendung (d. h. es gibt eine erste (in Teilen spielbare) Version des Spiels)
  - [12P] Es ist direkt spielbar, d.h. es können mehrere Spieler gleichzeitig spielen und es kommt zu keinen Fehlern, die das Spielen einiger oder aller verhindern
  - [4P] Cheat-Prevention Mechanismus ist implementiert und dokumentiert (in README)
  - [3P] Die Anwendung ist responsive und unterstützt grundlegendes Theming
  - [3P] Die Anwendung lässt sich auch über Handys/Tablets spielen (Touch-fähig)
  - [9P] Code-Qualität Backend (z. B. Lesbarkeit, Modularität, etc.)
  - [9P] Code-Qualität Frontend (s. O.)
  - [3P] Qualität des Build- und Deploymentprozesses (z. B. Automatisierung, etc.)
  - [3P] README mit Bau- und Installationsanleitung
  - [2P] HTTP API-Spezifikation bzw. Schnittstellendokumentation (in README)
- **[4P] Organisation**
  - [2P] Dokumentation des KI Einsatzes und Reflexion darüber
  - [2P] Dokumentation der Teamarbeit, d. h. wer hat welchen Anteil an der Entwicklung welcher Teile der Anwendung