

Sommersemester Projekt



Entwickle von Grund auf **ein eigenständiges, professionell gestaltetes Browser-Spiel**, das vollständig clientseitig mittels HTML, CSS und JavaScript umgesetzt wird. Es sind keine anderen Technologien oder Programmier-/Scriptingsprachen bei dieser Projektarbeit zugelassen!

Von der ersten Idee über Planung und Prototyp bis zur finalen Umsetzung durchläufst du einen vollständigen Entwicklungsprozess von **deinem eigenen, interaktiven Spiel!**

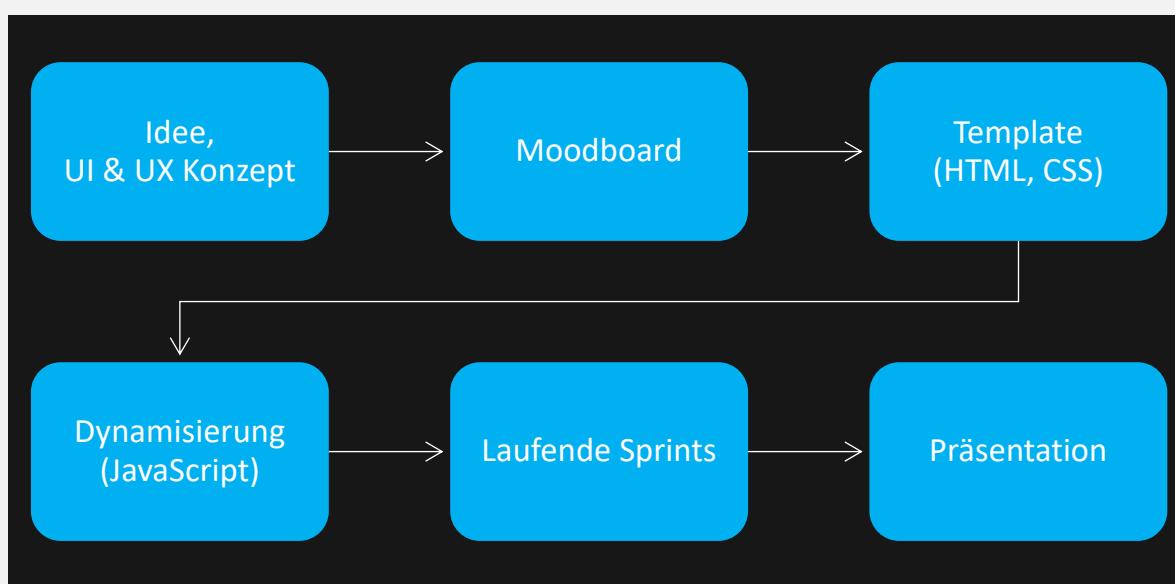
Im **Rampenlicht der Entwicklung** liegen die Bereiche

- **UI – User Interface** Intuitive, ansprechende Oberfläche
- **UX – User Experience** was dem User geboten wird, der User soll etwas erleben
- **Automatisierung** Repetitiven HTML-Content mit JavaScript erzeugen
- **Interaktion** dem User zahlreiche, spannende Möglichkeiten bieten, mit dem Spiel zu interagieren!

Dabei gelten für dich folgende Rahmenbedingungen:

- Umsetzung ausschließlich clientseitig, KEINE Serverlogik implementieren!
- Ausschließliche Verwendung von HTML/CSS/JavaScript
- Einsatz von mindestens einer externen JavaScript Library

Workflow



Kriterien & Mindestanforderungen

Um dein Spiel rein clientseitig umzusetzen stehen dir die folgenden Themen zur Verfügung:

➤ **Interaktive Rätsel, Puzzle, Logik („Escape Room“)**

(Beispiel: Ein digitaler Raum, aus welchen man nur entkommt, wenn man Hinweise kombiniert, Codes entschlüsselt, oder versteckte Objekte anklickt. Hier können unterschiedliche Möglichkeiten zur Eingabe kombiniert werden: Text-Eingabe, Drag-and-Drop, ... > dabei muss eine zusammenhängende Geschichte erzählt werden!)

➤ **Quiz- und Wissensspiele**

(Themenquiz mit Fragen aus JSON-Files, mit JOKER, Schwierigkeitsstufen, ...)

➤ **2D-Spiele mit Physics**

(Jump-and-Run im 2D/2,5D-Stil mit Gravitation, Kollisionserkennung, ... - hier muss eine deutliche Weiterentwicklung des typischen Sprite-Games erfolgen! Physikalische Komponenten müssen deutlich eingearbeitet werden, z. B: ein Katapult-Spiel mit Flugbahn-Berechnung)

➤ **Eigene kreative Idee einbringen**

(Zeichen-Spiel mit Bewertungen in einem CANVAS, Story-Game mit Entscheidungen, Text-Adventure/Visual Novel mit Generierung aus JSON Files, ...)

Zusätzlich dazu muss dein Spiel in folgenden Kategorien überzeugen:

➤ **UI – User Interface**

- ✓ **Ansprechende Oberfläche**, an die Thematik und Zielgruppe angepasst
- ✓ **Bedienbarkeit & Nutzbarkeit** (einwandfreie Funktionalität)

➤ **UX – User Experience**

- ✓ Spannende Features, die den **User** einnehmen und das **Interesse** wecken
- ✓ **Interaktionsmöglichkeiten**, durch die der User auf verschiedenste Art und Weisen die Möglichkeit hat, die Webapp zu beeinflussen
- ✓ **Branding – Wiedererkennungswert** (vor allem auch auf Unterseiten)
- ✓ **Gesamtes Spiel muss innerhalb des Viewports navigiert und gesteuert werden können** (keine Navigation in der URL-Leiste mit Zurück-Buttons etc.)

➤ **Landing Page des Spiels**

- ✓ **Spiel vorstellen** und kreativ und ansprechend **beschreiben, USP aufzeigen!**
- ✓ Anleitung und Tutorial – wie wird gespielt?
- ✓ Settings wenn notwendig, START-Button integrieren

➤ **Technische Anforderungen**

- ✓ **Automatisierung:** repetitiven Inhalt via JSON abspeichern und HTML generieren
- ✓ **Permanente Daten:** Infos längerfristig via LocalStorage speichern und verwerten
- ✓ Umfassendes, spannendes **Audio-Design** bei dem Spiel
- ✓ Einsatz von **min. 1 externen JavaScript Library**

➤ **Abschlusspräsentation mit visueller Unterstützung (Canva, Prezi, Reveal.js, PPT, ..)**

- ✓ Erklärung der **wichtigsten Code-Bausteine**
- ✓ Kurze **Reflexion** der Projektarbeit – was lief gut, was war ausbaufähig?
- ✓ **Live-Demo** des finalen und funktionsfähigen Spieles

WICHTIG – FREMDCODE UND KI:

Fremdcode (ChatGPT, Codepilot, Codepen, ...) ist ausnahmslos mit Kommentaren und Kurz-Erläuterungen zu kennzeichnen! Aber beachte: **du musst zu jedem Zeitpunkt der/die Chef/in deines Codes sein und bleiben.** Insbesondere bei den **Reviews** musst du Rede und Antwort stehen, hier wird **100% Codeverständnis** gefordert.

ABGABEKRITERIEN

- Für das Projekt ist ein GitHub-Education Account erforderlich
 - Get your Student benefits:
https://education.github.com/discount_requests/application
- Für das Projekt ist ein privates GitHub Repository zu verwenden
 - Du erhältst zu Beginn einen Invite-Link für GitHub-Classroom, dadurch bekommst du Zugang zu deinem privaten Repo.
 - Im Bereich documentation sind der Projektantrag, das UI/UX-Design sowie sämtliche Sprint-Dokus abzugeben
 - Der Bereich project ist deinen Projektdateien vorbehalten. Hier wird gecode!
 - Progressive Änderungen sind permanent per **Commits** zu hinterlegen
(Richtlinie > max. eine Stunde Arbeit > REPO aktualisieren! In den MEDTWT Stunden ist am Ende jeder Stunde ein verpflichtender, aussagekräftiger Commit durchzuführen!)
 - Fremdcode (Codepen, ChatGPT, Codepilot, ...) ist ausnahmslos mit Kommentaren und Kurz-Erläuterungen zu kennzeichnen!
- Die Abgabe erfolgt via Commits in euer privates Repo in 2-wöchentlichen Sprints.
- Jeder Abgabe ist ein UPDATE PDF-Dokument hinzuzufügen:
 - Formaler Header (Name, Klasse, Projektname, Link zu GitHub-Repo)
 - Neuerungen bzw. Änderungen seit letztem Sprint als Liste, in Stichworten
 - Screenshot der GitHub-Insights-Statistik
 - Verpflichtende Ziele bis zum nächsten Sprint

Deadlines 25/26

	Deadline (2AHITM)	Deadline (2BHITM)	Mindestanforderung	Sprint-Inhalte (Beispiel)
PROJEKTANTRAG	03.03.2026	04.03.2026	Vollständiger Projektantrag	Spieldie, Zielgruppe, Kermechanik, Feature-Planung
UI/UX KONZEPT	17.03.2026	18.03.2026	Vollständiger UI/UX Prototyp	Wireframes aller Bereiche (Landing Page, Spielansicht, Highscore-Bereich, Level-Auswahl, ...), Styleguide, Interaktionskonzept
DEV-SPRINT #1	07.04.2026	08.04.2026	HTML/CSS Grundgerüst komplett funktional	Navigation, alle grundlegenden Bereiche bedienbar, erste Funktionalitäten des Spiels
DEV-SPRINT #2	21.04.2026	22.04.2026	Kernmechanik implementiert, Spiel in Basisversion spielbar	JSON-Struktur angelegt, Mechaniken implementiert
DEV-SPRINT #3	05.05.2026	06.05.2026	Library eingebunden, mind. 1 Zusatzfeature	Punktesystem, Timer/Lives, Game-Over-Logik, Restart-Funktion, Entscheidungslogik
DEV-SPRINT #4	19.05.2026	27.05.2026	LocalStorage integriert (Highscore/Fortschritt)	Level oder Schwierigkeitsstufen verfügbar, weitere Story- oder Gameplay-Entscheidungen
DEV-SPRINT #5	02.06.2026	10.06.2026	Audio vollständig integriert	UX-Optimierung, Animationen, Soundeffekte, Bugfixing, Feinschliff
ENDABGABE	12. Juni 2026 – beide Klassen		Finales und funktionsfähiges Spiel inkl. Landing Page, JSON-Automatisierung, Präsentation	Vollständige Feature-Integration, finale Tests, sauberes Repository
PRÄSENTATIONEN	ab 16. Juni 2026 im Rahmen der MEDTWT-Stunden		Präsentation des Projekts	Live-Demo des Spiels, Erklärung der Mechaniken, Reflexion des Entwicklungsprozesses

PROJEKTANTRAG

Verfasse einen ansprechenden und ausführlichen Projektantrag! Der Antrag soll zeigen, dass du dein Projekt konzeptionell durchdacht hast – sowohl gestalterisch als auch technisch.

Decke dabei die folgenden Bereiche vollständig ab:

🎮 - GRUNDIDEE - 🎮

- Projektname (prägnanter, thematisch passender Titel)
- Art des Spiels (Um welchen Spieltyp handelt es sich?)
- Spielidee und zentrales Ziel des Spieles
- Soll eine Geschichte erzählt werden – wenn ja, welche?

🎮 - MECHANIKEN - 🎮

- Wie gewinnt oder verliert man?
- Erklärung der grundlegenden Spielemechanik
- Welche Aktionen kann der User im Spiel ausführen? Mit welchen Tasten?
- Welche Herausforderungen warten auf den User?
- Gibt es Level, Schwierigkeitsstufen, Progression, Entscheidungssysteme, ...?
- USP – was macht dein Spiel besonders?
- Spielablauf skizzieren

🎮 - ZIELGRUPPE - 🎮

- Welche Altersgruppe möchtest du ansprechen?
- Gelegenheitsspieler oder erfahrene Gamer?

**Abgabe des schriftlichen Antrages ausschließlich als .PDF oder .MD
(schön formatiert!) in deinem GitHub Repository!**

(Andere Formate, unpassende/keine Formatierung, Rechtschreibfehler, keine vollständigen Sätze = 0 Punkte!)