

Projektaufgabenstellung im WS 20/21

Modul "Webprogrammierung"/

Aufgabe:

Die Projektaufgabe besteht in der **Programmierung eines browserbasierten 2-D-Computerspiels**.

Sie können die Projektaufgabe **allein oder in der Gruppe zu max. 2 Personen** bewältigen.

- **Einzelpersonen** bauen bitte das **Spiel Pong** (Beispiel unter <https://youtu.be/ncB0ov5hT48>) oder das **Spiel Snake** (Beispiel unter <https://www.youtube.com/watch?v=YTFOWqZcEP4>) nach,
- während für ein **Zweierteam** das **Spiel Tetris** (Beispiel unter <https://www.youtube.com/watch?v=keeSEJG4XzU>) oder das **Spiel Space Invaders** (Beispiel unter <https://www.youtube.com/watch?v=MU4psw3ccUI>) programmiert werden soll.

Nutzen Sie zur **eigenständigen** Konzeption und Programmierung **HTML5 Canvas, ECMAScript 6 (Javascript) und CSS Level 3**. Ihr Spiel muss in der aktuellen Version des Firefox-Browsers (Desktop) laufen. Das Canvas ist 640*480 Pixel groß.

- Recherchieren Sie zu den Vorbildern im Web. Bei Unklarheiten kontaktieren Sie die Dozentin.
- Mindestens die ersten beiden Level oder zumindest Teilabschnitte sollten gem. des Originals funktionieren (Einzelspieler-Modus).
- Verschönerungen/Modernisierungen sind erlaubt. Die Nutzung von Libraries ist nicht möglich, z. B. JQuery

Eine sinnvolle Aufgabenteilung ist:

1. I/O-Programmierung (ECMAScript; Verhalten der Spielfigur, Tastatur, Maus, Netzwerk, Fehlerabfragen, Protokollierung)
2. Logik (ECMAScript; Regelwerk des Spiels, NPC-Verhalten, Level im JSON-Format, nachladen per AJAX)
3. Grafik-Programmierung (Canvas, CSS; Anzeigen)
4. Sound-Programmierung (Audio API; Geräusche, Sprache, Musik)
5. Umsetzung Design (validiertes HTML5), Assets (mit Quellennachweis) und Level-Daten (im JSON-Format)
6. Programmierung der Hilfsroutinen, API und Tools; insb. Level-Editor

Querschnitt: Anforderungsanalyse, Integration, Test, Projektmanagement, Dokumentation

Umfang:

Das Spiel umfasst nebst mind. je einer CSS- und ECMAScript-Datei:

1. Einstiegsseite/Menü (HTML)
2. Spielfeld (Canvas) auf separater HTML-Seite, in dem das eigentliche Spiel gespielt werden kann
3. Highscore-Liste (HTML; Bestenliste mit Webstorage: Localstorage zu speichern)
4. Hilfeseite (HTML; Spielzweck, Gewinn, Niederlage, Tastatur-/Maus-Kommandos)
5. Credits+Impressum (HTML; Namen aller Gruppenmitglieder, Ostfalia-Email-Adresse und Matr.-Nr.)

Ein Intro/Extro oder Demo-Modus ist nicht erforderlich, ebenso wenig ein Mehrspielermodus und verschiedene Schwierigkeitsgrade. Es muss einen Pausenmodus geben und der Ton muss aus- und einschaltbar sein.

Bewertungsschema:

Legen Sie Ihr fertiges Spiel inkl. Quellcode und kurzer Anleitung komplett als Zip bis spätestens 27.01.2021 23:59 Uhr im Ablageordner „Prüfung“ ab. In der mündlichen Prüfung am ??? stellen Sie allein oder im Team Ihr Spiel vor.

Die **pünktliche Abgabe** stellt bereits einen **Teil der mündlichen Prüfung** dar.

Eine nicht pünktlich hochgeladene, unvollständige oder nicht lesbare Datei führt zu "nicht bestanden". Bei technischen Ausfällen ist eine Verlängerung der Abgabefrist ausgeschlossen. Nichtbeachtung der o.g. Aufgabenstellung, auch in Teilen, führt zu Punkteabzug und kann z. B. beim Einsatz von 3rd-party Libraries zu "nicht bestanden" führen.

Die Bewertung variiert je nach Qualität, Angemessenheit und Vollständigkeit der systematischen Umsetzung zwischen 0% (n.b.) und 100% (1,0). Für eine 4,0 sind 51% zu erreichen. Die jeweils mit 10% bzw. 20% gewichteten Bewertungskriterien lauten:

1. Sinnvolle Ordner- und Dateistruktur gem. Arbeitspaketen, angemessene Dokumentation des Quelltexts, Anleitung vorhanden (readme.md)
2. Keine alerts oder toter Code im Quelltext, einheitliche Code-Richtlinien umgesetzt, Sicherheitsaspekte beachtet
3. Validiertes HTML5 (nutzen Sie validator.w3.org) und CSS Level 3
4. Objektorientiertes ECMAScript ≥ 6 (nur im Strict-Modus!), Protokollierung mittels der Funktion `console.log()`, Ausnahmebehandlung, Timeouts, AJAX und Fehlerabfragen
5. Mittels aktuellem Firefox-Browser auf einem Web-Server nutzbar, d.h. Ihr Projekt sollte auf dem Apache Web Server laufen (<http://localhost...>)
6. Level-Daten und Konfigurationsdateien sind im JSON-Format
7. (doppelte Gewichtung) Umsetzung des Spielverhaltens gem. Vorlage
8. (doppelte Gewichtung) Umsetzung der audio-visuellen Charakteristika gem. Vorlage oder wahlw. modernisiert

Referenzen:

- w3schools.com
- selfhtml.org
- [Canvas-Referenz](#)
- [Beispiel eines weiteren Effekts in CSS: Parallax-Scrolling \(performantere Lösung\)](#)