Projektdokumentation Leiterspiel

# Projektidee

Wir haben das Ziel verfolgt, ein Leiterspiel zu programmieren. Das Spielfeld besteht aus hundert Feldern. Dabei ist das erste Feld gleichzeitig auch das Startfeld und das letzte Feld das Zielfeld. Im Spiel sind die nummerierten Spielfelder mit Leitern verbunden, die man je nachdem hochklettern kann, um den Weg abzukürzen, oder runterklettern muss, um somit einen Teil des Weges erneut zurückzulegen. Mit einem Würfel wird bestimmt, wie weit die Spielfigur fahren darf. Würfelt man die Zahl Sechs, darf man einen zusätzlichen Zug ausführen. Der Spieler, der als erstes das Zielfeld erreicht gewinnt.

## Ursprünglich geplant

Für den Prototyp hatten wir uns vorgenommen, eine ganz klassische Version (wie oben beschrieben) zu programmieren. Diese sollte mit zwei Spielern, aber nur auf einem Gerät spielbar sein. Durch Klick auf das Würfelfeld wird eine Zahl zwischen 1 und 6 generiert. Die Figur des Spielers/der Spielerin, der/die an der Reihe ist, wird anschliessend automatisch um die gewürfelte Anzahl Felder vorgerückt. Landet man auf einem Feld mit einer Leiter (bzw. anderem Element), wird die Figur auf das mit dem Element verbundene Feld verschoben (Verschiebung nur in eine Richtung, abhängig vom Element). Wird die Zahl Sechs gewürfelt, kann der Spieler erneut einen Zug machen (unbegrenzt oft nacheinander möglich). Nach Abschluss des Zuges ist die andere Person an der Reihe. Die Endversion des Spiels soll gleich funktionieren wie der Prototyp, neu soll jedoch auch von zwei verschiedenen Geräten aus miteinander gespielt werden können.

## Umsetzung im Projekt

Bereits durch die administrative Umstrukturierung des Moduls und die daraus resultierenden Änderungen an den Anforderungen fürs Projekt, haben wir einige Änderungen an der finalen Version vorgenommen. Das Spiel ist nun nicht wie geplant an zwei Endgeräten spielbar. Stattdessen haben wir die freigewordenen Ressourcen darauf verwendet, das Spiel interaktiver zu gestalten. Anfängliche Unschönheiten wie beispielsweise, dass man das Zielfeld nicht genau treffen muss, sondern auch darüber hinausfahren kann, um zu gewinnen, haben wir überarbeitet. In der Endversion muss das Zielfeld genau getroffen werden, ansonsten wird für die restliche Anzahl Würfelaugen rückwärtsgefahren. Dadurch ist es nicht mehr gleich einfach zu gewinnen, da man beim Rückwärtsfahren auch noch nach unten fallen kann. Des Weiteren gibt es nun zwei verschiedene Würfel-Buttons: Ein gewöhnlicher Spielwürfel mit den Augenzahlen 1-6 und ein Spezialwürfel der Augenzahlen von -5 bis 10 (0 inklusive) ermöglicht. Man kann jederzeit auswählen, mit welchem Würfel man würfeln will. In der Mitte vom Spielfeld, auf Feld 55 befindet sich ein Tauschfeld. Landet ein Spieler darauf, erscheint ein Popup welches fragt, ob man den Platz mit der gegnerischen Spielfigur tauschen möchte. Bei Klick auf OK werden die Figuren getauscht, bei Klick auf Abbrechen bleiben die Figuren an ihrem bisherigen Platz. Neben diesen Erweiterungen haben wir auch das Design unseres Spiels überarbeitet. Neu gibt es eine Startseite, auf welcher die Spieler ihre Namen eingeben und eine Farbe für ihre Spielfigur wählen können. Je nach Einstellungen des Geräts, an welchem man das Spiel spielt, erscheint das Spiel entweder in einem hellen oder in einem dunklen Modus. Im Spiel selbst gelangen die Spielfiguren ausserdem anstatt über Leitern neu durch Röhren zu anderen Feldern.

# Erläuterung des Source Codes

## GitLab und Branches

URL zum Repository: https://gitlab.com/yxaw/front-projekt

Wir haben während des gesamten Projektes GitLab verwendet, um die Versionskontrolle für unsere Dateien zu gewährleisten. Dazu haben wir folgende Branches erstellt:

**main**

Hauptbranch, enthält nur funktionierende Zustände

**develop**

Entwicklungsbranch, darauf haben wir primär gearbeitet

**develop\_with\_api**

Stand des Entwicklungsbranches vor der Umstrukturierung des Moduls, enthält erste Versuche zur Einbindung der Serverkommunikation, Stand festgehalten, falls wir zu späterem Zeitpunkt die Serverkommunikation doch einbinden möchten

**develop\_with\_dragndrop**

Stand des Entwicklungsbranches, enthält den Versuch die Spielfigur per Drag & Drop bewegen zu können, falls wir dieses Feature zu einem späteren Zeitpunkt noch fertigstellen möchten

**abgabe-prototyp**

Branch zur Abgabe des Prototyps am Ende des HS22. Hält den damaligen Entwicklungsstand fest.

**abgabe-projekt**

Branch zur Abgabe des gesamten Projektes am Ende des Moduls im FS23

Diese Infos und weitere Guidelines sind auch in der Readme.md Datei einsehbar.

Wir haben nicht für jede Entwicklerin einen eigenen Branch erstellt, sondern jeweils gemeinsam auf dem aktuellen develop branch gearbeitet.

## Dateistruktur

Das root Verzeichnis enthält die beiden html-Dateien index.html und spiel.html, die readme-Datei und die Ordner assets, documents und mockups. Da wir nur zwei html-Dateien haben, entschieden wir, dass es ist übersichtlicher ist diese beide im root-Ordner zu lassen.

Der Ordner documents enthält einen Ordner mit allen Dateien zum Thema Javascript Libraries, einen Ordner mit allen Projektmanagement-Dateien und einen Ordner für alle Mockups, welche wir für das Design und die Funktionalitäten erstellt haben. Des Weiteren befinden sich darin die Dateien für die Dokumentation des Projektes, die ursprünglich formulierte Projektidee und die Schnittstellendefinition, welche wir im HS22 erstellt haben.

Der Ordner assets enthält die Javascript-Dateien, die CSS-Datei und einen Ordner «images», welcher alle Bilder enthält, welche wir für die Hintergründe der einzelnen Felder und der gesamten Seite verwendet haben.

### Javascript

Um die Funktionalitäten zu gruppieren und den Code übersichtlicher zu behalten haben wir mehrere JavaScript-Dateien erstellt. Dazu eine Bemerkung: Da wir uns erst am Schluss für das Design mit den Röhren als Transportelemente entschieden haben, und die Grundidee das Leiterspiel war, wird im Code immer von Leiterfeldern gesprochen, diese Bezeichnung belassen wir so, weil wir auch das Spiel selbst weiterhin als Leiterspiel bezeichnen.

**Feld.js**

In dieser Datei werden die Klassen Feld und Leiterfeld konstruiert. Darin enthalten sind Methoden, um die DOM-Elemente der Felder mit den richtigen Klassen und IDs zu attribuieren. Ebenso wird die Konfiguration der Leiterfelder in einer Konstanten festgehalten (hier hat uns eine Arbeitskollegin von Yara bei der Umsetzung geholfen).

**Spielfeld.js**

Mithilfe der in feld.js erstellten Klassen wird in der Datei Spielfeld.js. das Array der einzelnen Feld-Instanzen erstellt, welches zur Anzeige des Spielfelds im HTML benötigt wird. Je nach Fall werden so die Felder entsprechend ihrer Aufgabe korrekt instanziert und attribuiert.

**Spielfigur.js**

In dieser Datei wird die Klasse Spielfigur definiert, das DOM-Element Spielfigur erstellt und attribuiert. Durch entsprechende Methoden kann die Position der Figur ermittelt und aktualisiert werden.

**Startseite.js**

Hier werden die nötigen Variablen erstellt, um die Werte Spielernamen und Spielfigurfarbe zu speichern und dem SessionStorage zu übergeben. Dies wird benötigt, um die Farb- und Namensselektion an die DOM-Elemente zu übergeben und bei einem Seitenrefresh zu speichern.

**Spielfiguranzeige.js**

Hier wird das DOM-Element für die Anzeige des aktuellen Spielers im HTML erstellt. Angezeigt werden der Name und die Farbe des aktuellen Spielers, welche auf der Startseite eingegeben wurden.

**Storage.js**

Enthält die Klasse StorageService und alle notwendigen Funktionen und Methoden, um den SessionStorgae des Browsers zu befüllen und auszulesen, sowie den aktuellen Spielstand darin zu speichern.

**Wuerfel.js**

Die Datei enthält die Klasse Wuerfel und alle zugehörigen Funktionen und Methoden. Darin wird ebenso die Anzeige der zuletzt gewürfelten Zahl und das Sperren der Würfelbuttons, während dem der Spielzug durchgeführt wird und das Entsperren der Buttons, wenn der Spielzug beendet wurde, ermöglicht.

**Spiel.js**

Hierbei handelt es sich um die komplexeste Datei. Einerseits wird hier das gesamte Spiel konstruiert und mit allen zugehörigen Elementen wie Spielfigur und Würfel instanziert. Andererseits enthält es die Methoden der Spielzüge. In den Spielzügen wird auf alle möglichen Eventualitäten getestet (muss ein Leiterfeld benutzt werden, wurde eine 6 gewürfelt, hat der Spieler gewonnen, …) und der entsprechende Code ausgeführt. Weiterhin gehört zu einem Spielzug auch das Errechnen des Feldes, welches der Spieler mit seinem Wurf erreicht und das Platzieren des DOM-Elements auf diesem, so dass die Spielzüge auch im HTML entsprechend angezeigt werden. Weiterhin sind die Methoden erhalten, welche das Landen auf dem Tauschfeld prüfen und das Durchführen des Tausches ermöglichen, sowie das korrekte Abspeichern und Auslesen des Spielstands, sodass dieser bei einem Seitenrefresh nicht verloren geht. Es bietet auch die Möglichkeit zu jedem Zeitpunkt ein neues Spiel, entweder mit den gleichen Spielern (Button Nochmal Spielen) oder mit neu gewählten Spielernamen und Farben (Button Neues Spiel, Umleitung zurück auf die Startseite), zu starten.

# Reflexion

* Arbeit mit versionsverwaltung

Die Arbeit mit git war für uns alle neu und hat zu Beginn eine gewisse Gewöhnungsphase gebraucht. Vor allem auch die Verwaltung von MS Word und Excel Dateien in git erschien uns zu Beginn fremd, da wir nicht über die IDE (in unserem Fall Visual Studio Code) darauf zugreifen konnten und wir uns eher die Verwendung von Tools wie Google Docs gewöhnt waren. In diesem Fall wäre es evtl. auch sinnvoller gewesen, weiterhin mit gewohnten Tools zu arbeiten, da die gemeinsame Bearbeitung einfacher gewesen wäre. So mussten wir immer sicherstellen, dass die Bearbeitung der Word- und Exceldateien abgeschlossen ist, um merge-Konflikte, die unseres Wissens nicht klug zu bearbeiten gewesen wären, zu vermeiden. Bei den Code-files hingegen konnten wir sehr von git profitieren. Jedoch hätten wir auf hier jeden Fall noch mehr kleinere commits mit genaueren commit messages machen können. Stattdessen haben wir dazu tendiert, sehr viele, verschiedene Änderungen aufs Mal mit einem einzigen commit zu pushen. Einmal ….. Durch die fehlende Erfahrung mit git haben wir sicherlich nicht das volle Potenzial ausgeschöpft. Wir haben generell möglichst vermieden, zur gleichen Zeit an unterschiedlichen Stellen im Projekt zu arbeiten, um Konflikte zu vermeiden. Hier hätten wir sicher mehr gewagt, wenn wir mit der Benutzung von git vertrauter gewesen wären. Des Weiteren haben wir den main-Branch zu wenig aktiv weitergeführt. Wir hätten regelmässiger die funktionierenden Zustände mergen sollen. Stattdessen haben wir den main bis zu den Abgaben vernachlässigt. In einem nächsten Projekt würden wir versuchen dies besser zu machen. Für uns ist es auf jeden Fall denkbar, in zukünftigen Studiums-Projekten weiter mit git zu arbeiten und dadurch die Funktionalitäten und Möglichkeiten noch besser kennenzulernen und ausschöpfen zu können.

* Was hat funktioniert, was nicht
* Einschätzung zur ursprünglichen Projektidee

Wir sind der Meinung, ein sehr solides Spiel auf die Beine gestellt zu haben. Die Ziele, die wir uns zu Beginn gesetzt haben, konnten wir eigentlich alle erfüllen. Beim Design, welches uns von anderen Online-Leiterspielen abheben sollte, hatten wir lange Mühe, eine geeignete Lösung zu finden. Uns war wichtig, dass das Spielfeld nicht zu vollgepackt wirkte, sondern die einzelnen Felder und

* Grösse der Felder, wenn beide Figuren auf dem selben Feld landen
* Überlappen der Figur und der Röhre

Vergleich Soll - Ist: keine Serverkommunikation, kein manueller Spielzug, eigenes Design umgesetzt, Spiel spannender gestalten durch Extras,

* Wo gab es Probleme und wie sind diese gelöst worden (technischer, wie auch gruppendynamischer Natur);

Gruppendynamisch gab es bei uns eher wenig Probleme, zumindest keine mit denen wir nicht umgehen konnten. Schon sehr früh wurde klar, dass Anna und Yara sich in der Programmierung sicherer fühlten als Morena. So hat Morena mehr administrative Arbeiten übernommen und mehr Zeit dazu aufgewendet, den von Yara und Anna geschriebenen Code zu verstehen, als selbst Codezeilen zu verfassen. Für Anna und Yara war diese Abmachung so in Ordnung. Im letzten Monat vor der Abgabe hatten wir das Problem, dass die persönlichen Situationen der Gruppenmitglieder dazu führten, dass niemand wirklich die Gelegenheit hatte, im notwendigen Umfang am Projekt weiterzuarbeiten. In diesem Semester war geplant, den Spielern zu ermöglichen ihre Figur selbst zu bewegen (entweder per Drag&Drop oder durch Anklicken des Landefelds), um das Spiel interaktiver zu machen. Aus persönlichen Gründen kamen wir erst spät im Semester dazu, mit der Umsetzung dieses Features zu beginnen. Dabei stellte sich jedoch heraus, dass dieses Feature nicht so einfach umsetzbar war. Die Implementierung hat zu Timing-Problemen geführt. Beim Versuch diese Probleme zu beheben, entstanden jedoch weitere Probleme. Da wir es ohne Hilfe nicht schafften, das Feature vollständig einzubauen und leider keine Zeit mehr für ein Coaching blieb, vereinbarten wir mit der Dozentin zu unserer grossen Erleichterung, das Feature wegzulassen. Den bisher umgesetzten Stand haben wir jedoch in einem separaten branch (develop\_with\_dragndrop) eingecheckt, damit zu einem späteren Zeitpunkt daran weitergearbeitet werden könnte. Ausserdem sind unsere Bemühungen so auch für die Dozentin sichtbar.

Wenn wir technische Probleme hatten oder nicht weiter wussten, haben wir uns externe Hilfe geholt, meistens direkt bei den Dozierenden. Die Coaching-Sessions haben uns sehr geholfen.

* Was sind die persönlichen Learnings.

Morena: Wie bereits oben erwähnt, habe ich selbst eher wenig Sicherheit in der Programmierung von Javascript erlangt. Die Struktur des Unterrichts hat für mich nicht gepasst, in den Theorieblöcken gab es zu wenig praktische Übungen, und die Praxisblöcke haben zu fortgeschritten begonnen und gaben zu wenig Zeit zum begleiteten Üben, respektive man hatte einfach zu wenig Zeit zum selbst Code schreiben und musste zu sehr der Lektion hinterher hetzen und hatte keine Zeit zum Nachvollziehen, wo ein Fehler passiert ist, wenn etwas nicht funktioniert hat. Im Selbststudium habe ich versucht, mir diese Kenntnisse anzueignen, hatte dabei aber wenig Erfolg, was wiederum zu einer verringerten Motivation geführt hat. Ich möchte aber damit nicht sagen, dass das Modul ein «Reinfall» war, im Gegenteil. Ich fand es äusserst interessant. In der Theorie habe ich sehr viel mitgenommen und ich denke die Logik hinter dem Code sehr gut verstanden zu haben und somit ein gutes Verständnis für mögliche Problemlösungen zu haben. Ich sehe es hier ein bisschen wie das Erlernen einer gesprochenen Sprache, ich verstehe was gesagt wird, kann aber selbst nur gebrochen sprechen.

Yara: Der Start in die Programmierung im ersten Semester war ziemlich holprig. Gleich wie bei Morena hatte ich das Gefühl, die Theorie verstanden zu haben, bei der selbständigen Umsetzung hatte ich jedoch Probleme. Durch die Repetition, das gemeinsame Coden in der Gruppe, die Coaching-Sessions mit den Dozierenden und das selbständige Ausprobieren habe ich jedoch sehr viel dazugelernt und immer besser verstanden, wie der Code geschrieben werden musste, um das gewünschte Ergebnis zu erhalten. Ab Mitte des zweiten Semesters fühlte ich mich schon viel sicherer und hatte Freude am Schreiben des Codes, was unter anderem an den Erfolgserlebnissen lag, wenn ich etwas geschafft hatte wie gewünscht umzusetzen. Auch bei der Arbeit mit GitLab habe ich einiges dazugelernt und werde das Tool bestimmt auch in Zukunft für nutzen.

Ausblick

Serverkommunikation, manueller Spielzug, Code verbessern (gleiche Codestellen zusammenführen, …), viele weitere Ideen, um das Spiel spannender zu gestalten

# Eidesstaatliche Erklärung

Wir erklären hiermit, dass wir diese Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und erlaubten Hilfsmittel benutzt haben. Alle Stellen, die wörtlich oder sinngemäss aus Quellen entnommen worden sind, haben wir als solche gekennzeichnet. Uns ist bekannt, dass andernfalls die Hochschulleitung zum Entzug der aufgrund unserer Arbeit verliehenen Qualifikation oder der für unsere Arbeit verliehenen Titel berechtigt ist.



Lenzburg,

--------------------------------------- ---------------------------------------

Ort, Datum Unterschrift

Morena Sager

--------------------------------------- ---------------------------------------

Ort, Datum Unterschrift

Anna Staub



--------------------------------------- ---------------------------------------

Ort, Datum Unterschrift

Yara Wagner