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 haben wir uns vorgenommen, eine ganz klassische Version (wie oben beschrieben) zu programmieren. Diese soll 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 funktioniert gleich 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 durch eine Röhre 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 minus 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.

# 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, auf diesem Branch 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, welcher den Versuch die Spielfigur per Drag & Drop bewegen zu können enthält, 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.

Zur Entwicklung wurde jeweils gemeinsam auf dem aktuellen develop branch gearbeitet, ohne für jede Entwicklerin einen eigenen Branch zu erstellen und erst danach in den develop zu mergen.

## 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 es ist übersichtlicher, diese beide im root zu lassen.

Im Ordner documents befinden sich ein Ordner mit allen Dateien die wir erstellt haben für das Thema Javascript Libraries, ein Ordner mit allen Projektmanagement Dateien und ein Ordner für alle mockups, welche wir fürs 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 erstellen mussten.

Der Ordner assets enthält die Javascript-Dateien, die CSS-Datei und einen Ordner «images» mit allen verwendeten Bildern für die Hintergründe der einzelnen Felder und der gesamten Seite.

### Javascript

Im Javascript haben wir einzelne Dateien erstellt, um die Funktionalitäten zu gruppieren und den Code übersichtlicher zu behalten. Dazu eine Bemerkung: da wir uns erst am Schluss für das Design mit den Röhren als Transportelemente entschieden haben, und die Grundidee das Leiterspiel ist, wird im Javascript-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 und enthält 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.

**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 instanziiert und attribuiert.

**Spielfigur.js**

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

**Startseite.js**

Erstellt die nötigen Variablen, um die Werte Spielernamen und Spielfigurfarbe zu speichern und dem SessionStorage zu übergeben. 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 zu speichern.

**Wuerfel.js**

Enthält die Klasse Wuerfel und alle zugehörigen Funktionen und Methoden. Ermöglicht 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.

**Spiel.js**

Hierbei handelt es sich um die komplexeste Datei. Einserseits 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 werden auf alle möglichen Eventualitäten getestet (Muss ein Leiterfeld benutzt werden, wurde eine 6 gewürfelt, hat der Spieler gewonnen, …) und dementsprechend wird der Spielzug korrekt durchgeführt. 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
* Was hat funktioniert, was nicht
* Einschätzung zur ursprünglichen Projektidee
* Wo gab es Probleme und wie sind diese gelöst worden (technischer, wie auch gruppendynamischer Natur);
* Was sind die persönlichen Learnings.

# 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