

## Projektplan

### Projektidee:

Eine E-Learning Plattform für Lehrer und Schüler mit Gamification-Aspekten entwerfen, die nicht nur „trockene“ Quests in Form von Lückentexten etc. enthält, sondern auch interaktive Quests, wie z.B. Labyrinth und Puzzle. Die Lerninhalte sollen dabei nicht vorgegeben werden, sondern als Maske in Form eines Questtoolbuilders vorliegen, so dass die Lehrer ihre eigenen Lehrinhalte in die Quests einbringen können. Dies ermöglicht der Website sich an die Lehrpläne der Lehrer anzupassen.

Der Fokus des Projekts liegt auf dem Layout der Website und dem Erstellen geeigneter der Toolkits für die einzelnen Questformen, welche die Lerninhalte gut darstellen.

### Aufbau:

- Layout der Seite mit Navigationsfunktionen (HTML, CSS)
- Einbinden der Spielelemente via Javascript / Java-Web-Frameworks (Spring MVC?)
- Entwerfen einiger Questtoolkits (Labyrinth, Puzzle), die gebaut und gespeichert werden können
- Lückentexte, die über .txt-Dateien eingelesen werden und von den Lehrern nach Belieben bearbeitet werden können
- Interaktion nur über Website
- Profil etc. werden über Datenbank erstellt (→ speichern von Nutzernamen, Passwort, Avatar, Fortschritt usw.) → kein Must-Have, da DBS noch nicht belegt wurde

### Wichtige Funktionen:

#### Labyrinth:

- Tabellenfunktionen, um das Labyrinth zu erstellen
- getElementById(): bestimmen den aktuellen Ort des Spielers im Labyrinth und den Ort der einzusammelnden Objekte
- indexOf(): prüfen, ob eine Kollision mit der Wand stattfindet oder ein Objekt eingesammelt wurde

#### Puzzle:

- Rectangle(): Rechtecke als Puzzle-Objekte zeichnen
- MouseEvent(): Wichtig, um Drag&Drop und Doppelklick zu implementieren
- fillText(): Text in die Puzzleteile einfügen
- getImageData(): Bilder in die Teile einfügen (oder in Java auslagern?)

#### Lückentext:

- Array: zum Speichern der richtigen Antworten

Kevin Ehrhardt

- `indexOf()`: vergleichen der Antworten mit Lösung

Multiple-Choice:

- `MouseEvent()`: Zum Klicken der Buttons

### Toolkit-Ideen:

#### **Puzzle:**

- Per Drag & Drop werden die Elemente verschoben
- Die Elemente werden durchnummeriert, um anhand dessen die Richtigkeit der Aufgabe zu überprüfen
- Mit einem Doppelklick kann Text oder Bild in die Elemente gespeichert werden
- Mit einem Speichern-Button werden die Positionen der Elemente gespeichert
- Die Positionen dienen beim Ausführen als Zeiger zum Einrasten der Elemente
- Lerneffekt: Verstehen von Prozessen bzw. Sachen richtig benennen können

#### **Labyrinth:**

- Labyrinthvorlage ist gegeben, Elemente die beim Spielen eingesammelt werden müssen, müssen eingefügt werden
- Elemente können per Drag & Drop in das Labyrinth gezogen werden
- Es muss markiert werden, welche Elemente eingesammelt werden sollen und welche nicht
- Legende zur Objektbeschreibung muss eingefügt werden
- Infonachricht, wenn Element eingesammelt wurde (z.B. du hast Objekt X gefunden)
- Inventar, welches gesammelte Elemente anzeigt
- „Müllstation“ vorm Ende, bei der fehlgesammelte Elemente abgelegt werden können
- Lerneffekt: interaktives Anwenden / selektieren von Informationen in richtig und falsch

#### **Lückentext:**

- Text wird als .txt-Datei hochgeladen
- Worte werden in Listenelemente gespeichert
- Lücken werden gekennzeichnet (Worte entfernt)
- Lückentext wird wieder als Liste gespeichert und mit vorheriger Liste verglichen
- Listenelemente, die fehlen, werden mit \$\$\$ gefüllt, um Lücke zu kennzeichnen
- Lückenwörter werden als Liste gespeichert, Lücken werden durchnummeriert zum Korrigieren
- Lerneffekt: Anwendung von Fachbegriffen im Kontext

#### **Multiple-Choice:**

- Eingabefeld, um Frage einzutippen
- Eingabefeld, um Antwortmöglichkeiten einzutippen
- Buttons, um richtige Fragen zu markieren
- Lerneffekt: Abfragen von Faktenwissen

### Komponenten für die Websiteoberfläche:

#### **Schüler:**

- Profil
  - o Leistungsstand
  - o Avatar
  - o Nutzernamen
- Abgeschlossene Quests

#### **Lehrer:**

- Profil
  - o Nutzernamen
  - o Avatar
  - o Unterrichtete Klassen
- Questtoolkits
- Leistungsstand der Schüler

### Schwierigkeiten:

- Öffnen und Schreiben von Dateien mit Javascript nicht möglich → Methode in Java auslagern
- Arbeiten mit Javascript neu
- Kein Vorwissen zum Erstellen von Datenbanken

### Must-Have:

- Entwerfen von Beispieldateien zum Testen der Schülerseite
- Komplettes Layout der Seite mit allen notwendigen Komponenten
- Toolkits zum Erstellen von Aufgaben
- Einbindung der Aufgaben in spielerische Rollenspiel-Elemente

### Mögliche Extras:

- Weitere Spielobjekte (z.B. Pacman artiges Spiel?, Kreuzworträtsel)
- Anmeldeverfahren mit Nutzernamen und Passwort entwickeln → Datenbank
- Avatarerstellung