**Praktikum Endaufgabe SoSe21**

**Entwickeln Sie ein interaktives Tic-Tac-Toe-Spiel**

**unter Berücksichtigung folgender Anforderungen**

**Spielregeln**

* Auf einem Spielfeld setzen zwei Spieler abwechseln ihre Spielsteine (bspw. rote / schwarz Spielsteine oder Kreuz / Kreis) in ein freies Feld.
* Der Spieler, der als erstes eine Zeile, Spalte oder Diagonale vollständig mit seinen Spielsteinen setzten kann, gewinnt die Spielrunde.
* Das Spiel ist ausgelegt für zwei Spieler, für einen Nutzer/Mensch gegen den Computer.
* Bei Spielstart kann der Nutzer zunächst den Schwierigkeitsgrad (leicht, mittel, schwer) des Spiels wählen (mehr dazu siehe weiter unten).
* Das Spiel besteht aus mehreren Spielrunden in Abhängigkeit zum Schwierigkeitsgrad (mehr dazu siehe weiter unten).
* Eine Spielrunde läuft ab wie folgt: Der Computer fängt an und setzt einen Spielstein. Jetzt ist der Nutzer an der Reihe und setzt einen Spielstein in ein freies Feld. Abwechselnd folgen weitere Spielzüge, wie oben beschrieben.
* Eine Spielrunde ist zu Ende, wenn...
  + ...ein Spieler eine Zeile, Spalte oder Diagonale vollständig mit seinen Spielsteinen setzen konnte. Dieser Spieler gewinnt die aktuelle Spielrunde und erhält einen Punkt.
  + ...es keine freien Spielfelder mehr gibt. Die Spielrunde endet in diesem Fall mit einem Unentschieden.
* Das Spiel ist nach der letzten Spielrunde zu Ende.
* Gewonnen hat der Spieler, der die meisten Punkte hat.

**Weitere Anforderung an die Anwendung:**

* Das Spiel kann in drei Schwierigkeitsstufen gespielt werden:
  1. Leicht: das Spielfeld besteht aus 3x3 quadratischen Feldern. Das Spiel besteht aus 3 Spielrunden.
  2. Mittel: das Spielfeld besteht aus 4x4 quadratischen Feldern. Das Spiel besteht aus 4 Spielrunden.
  3. Schwer: das Spielfeld besteht aus 5x5 quadratischen Feldern. Das Spiel besteht aus 5 Spielrunden.
* Das Spiel ist **optimiert für eine Desktop** Auflösung von 1280x800 Pixel.

Beurteilung

Note der End-Abgabe basiert auf den erreichten Anforderungs-Punkten (siehe weiter unten). Bis zu ein Punkt darf dabei nicht erfüllt sein um die 1 zu erhalten.

Jeder weitere fehlende Punkt verringert die Note um 1. (4.0 <= Bestanden).

Jedes erfüllte optionale Ziel erhöht die Note um 1. (Maximale Bestnote: 1).

Der Kurs wird im Studi-Portal mit "bestanden" / "nicht bestanden" bewertet.

Herangehensweise

Entwickeln Sie ein Konzept auf Papier, das die grundlegenden Mechanik der Anwendung beschreibt, bevor Sie mit der Programmierung beginnen!

Versuchen Sie den Spielfluss in einem Flowchart darzustellen. An welchen Punkten werden Entscheidungen gefällt? Muss der Computer gewisse Dinge abfragen? Wann muss der Spieler interagieren, wie interagiert er, was sind seine Optionen? Wie sehen die Eingabgemöglichkeiten (bspw. Buttons) für den Spieler aus? Wie validiert der Computer die Nutzereingaben?

Anforderungspunkte

Gesamt: 24. (24 = 1.0, 23 = 1.0, 22 = 2.0, 21 = 3.0, 20 = 4.0, <= 19 = NB)

Nach folgenden Kriterien wird Ihre Abgabe bewertet:

**Konzeptentwurf**

* In einem Flussdiagramm haben Sie den grundsätzlichen Interaktionspfad der Anwendung dargestellt (Interaktion des Nutzers und Verhalten des Systems). Den Papier-Entwurf bitte als Scan / Foto digitalisieren und beilegen.
* Ein Entwurf der grafischen Benutzeroberfläche gibt einen groben Überblick über die UI-Elemente des Spiels. Hier bitte auch: den Papier-Entwurf / Entwürfe als Scan / Foto digitalisieren und beilegen.

**HTML und CSS**

* Ein schlichter Footer mit Ihrem Namen und Matrikelnummer.
* Semantische HTML-Tags wurden genutzt, Tags werden sinnvoll eingesetzt.
* Die Spielfläche besitzt eine passende Hintergrundfarbe (oder Hintergrundtextur).
* Vor dem Spiel wird dem Nutzer eine Möglichkeit zur Auswahl des Schwierigkeitsgrads angezeigt.
* Nach Auswahl des Schwierigkeitsgrads werden dem Nutzer auf der Spielfläche entsprechend viele Felder, die Gesamtzahl der Spielrunden angezeigt.
* Die quadratischen Felder der Spielfläche sind entweder leer oder zeigen die gesetzten Spielsteine der jeweiligen Spieler an.
* Für den Nutzer und für den Computer gibt es eine Anzeige des aktuellen Punktestands und es gibt eine Anzeige der Spielrunde.

Tipp 1: Sie können zur Gestaltung der Spielsteine auch Grafiken nutzen.

Tipp 2: Sie können auch auf die Icon-Font [Fontawesome](https://fontawesome.com/icons?d=gallery) zurückgreifen, um passende Piktogramme für die entsprechenden Spielsteine darstellen zu können.

**Programmierung**

* Das Skript erzeugt keine Fehler in der Konsole.
* Die Code-Style Regeln des Linters wurden berücksichtigt (bspw. Typisierung von Variablen).
* Die Anwendung basiert auf nur einer HTML-Datei. Für andere Ressourcen bzw. Dateien-Typen gilt diese Einschränkung nicht (d.h. neben der einen HTML-Datei können also beliebig viele CSS-, TypeScript-Dateien, Config-Dateien, wie Linter und TSConfig, etc. im Projektordner genutzt werden).
* Der Computer generiert zu Beginn des Spiels die Spielfläche mit entsprechend vielen Feldern. Die Anzahl der Felder entspricht dem Schwierigkeitsgrad.
* Die Anzahl der zu spielenden Runden entspricht dem gewählten Schwierigkeitsgrad (Definition siehe oben).
* Das Spielfeld und jedes Feld des Spielfelds wird auf Code-Ebene durch eine sinnvolle Datenstruktur abgebildet.
* Der Computer kann in ein freies Feld einen Spielstein setzen.
* Der Nutzer kann durch Klick auf ein freies Feld einen Spielstein setzen.
* Pro Spielzug kann der Computer, als auch der Nutzer, nur einen Spielstein setzen.
* Pro Spielzug kann immer nur derjenige, der am Zug ist, einen Spielstein setzen.
* Sobald nach oben genannten Regeln ein Spieler die Spielrunde gewonnen hat oder es kein freies Feld mehr gibt, dann ist die aktuelle Spielrunde zu Ende.
* Wenn ein Spieler eine Spielrunde gewonnen hat, dann erhält dieser Spieler einen Punkt.
* Das Spiel ist zu Ende, wenn die letzte Spielrunde beendet ist. Der Spieler mit den meisten Punkten gewinnt das Spiel. Der Nutzer erhält eine Nachricht, wenn das Spiel beendet wurde.
* Der Nutzer kann das Spiel erneut starten, wenn es zu Ende ist. Das Angebot des Schwierigkeitsgrades ist in diesem Interaktionsschritt optional.

**Annotationen**

* Erklären Sie Ihren Code. Die Skript-Datei sollte mit Kommentarblöcken ergänzt sein, um relevante Anwendungsblöcke und Anweisungen beschreiben zu können (vgl. letzte Aufgaben-Vorlage).

**Optional** Diese Punkte MÜSSEN nicht erfüllt werden! Sie fließen jedoch positiv in die Berwertung mit ein (siehe oben).

* Das Setzen eines Spielsteins wird animiert dargestellt.
* Ein **visueller Fortschrittsbalken** plus Textinformation (bspw. Runde 1 von 3) wird am Bildschirm angezeigt.
* Wenn ein Spieler gewinnt, dann wird eine Nachricht mit einer Zusammenfassung des Spiels angezeigt und eine Fanfare ertönt. In Textform wird angezeigt: gewählter Schwierigkeitsgrad, feierliche Nachricht und Dauer des Spiels (bspw. 53 Sekunden).
* Das Spiel funktioniert auch auf mobilen Endgeräten mit Touch-Input.
* 2 Extra-Punkte: das Spiel kann auch mit zwei Nutzern an einem Gerät gespielt werden. D.h. zu Spielbeginn kann der Schwierigkeitsgrad, als auch der Modus "Mensch gegen Computer" und "Mensch gegen Mensch" ausgewählt werden. Bei "Mensch gegen Mensch" können sich dann zwei Nutzer an einem Gerät abwechseln.

Abgabe

Die Abgabe der Arbeit erfolgt diesmal nicht über Github, sondern im Felix-Kurs, auf der Unterseite "Abgabe Endaufgabe". Verpacken Sie Ihr Ergebnis als ZIP-Datei und laden Sie Ihre Abgabe in Ihren Teilnehmerordner in Felix hoch. Der Abgabebereich ist bis zur Abgabefrist zugänglich. Ihre ZIP-Datei sollte die Größer von 300MB nicht überschreiten.

**Abgabe bis Sonntag, 04.07.2021 23:59 Uhr**