

Anforderungen:

Spieler konfigurieren

- Formular mit Input Field in Modal Overlay

(Re-) Start Game

- „Start Game“ Button setzt aktuellen Stand zurück

Gameplay

- Spieler sind abwechseln am Zug → O / X
- Spielfelder sind klickbar
- Nach jedem Zug checken, ob es einen Gewinner/Unentschieden gibt

Game Over Window

- Popup mit „Game over“ Nachricht + Sieger angeben

Exercise 1 – Projekt & Grundstruktur anlegen.

Erstelle ein Projekt und lege die Grundstruktur an:

- index.html
- Ordner: styles (für css-Files)
- Ordner: scripts (für js-Files)

Exercise 2 – HTML Struktur anlegen

Lege die Struktur mit statischen Daten. Anhaltspunkte:

Header

Main mit 3 Sections: Modal Overlay, Game-Configuration und Active-Game

Play Tic, Tac, Toe

Built with HTML, CSS, JavaScript and - of course - lots of love!

Choose your name

Player name

1. Player 1

PLAYER NAME

X

2. Player 2

PLAYER NAME

O

You won, PLAYER NAME!

Click "Start New Game" above, to start a new game!

It's your turn PLAYER NAME!

Exercise 3 – Styling

Base Style

- Binde eine Google Fonts Schrift ein: Open-Sans
- Erstelle das style.css und binde es in der index.html ein
 - o Style den Header!
 - o Style die Buttons!

Cancel

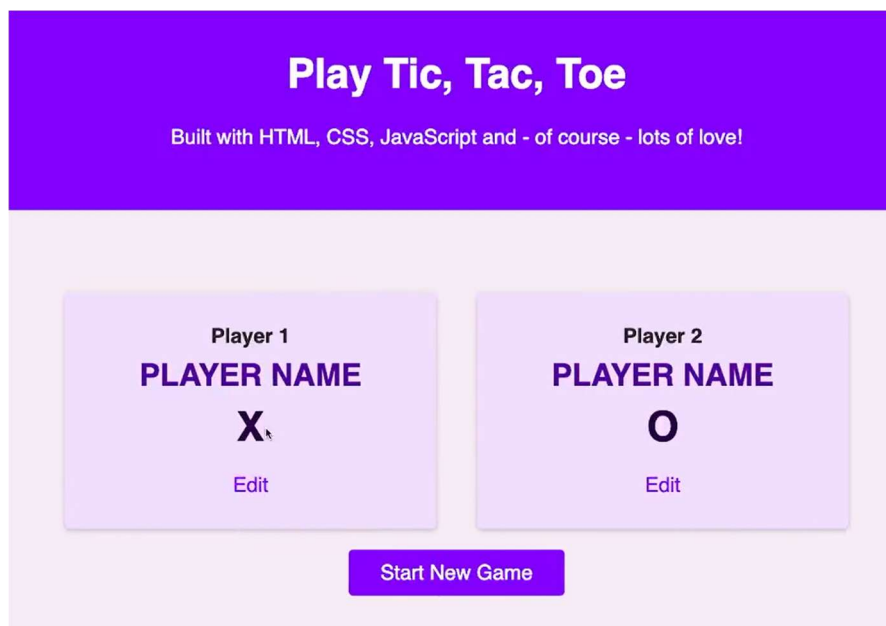
Confirm

Overlays

- Overlay (Modal) stylen
- Erstelle dazu das overlays.css und binde es in der index.html ein
- Erstelle ein generisches Styling für Overlays

Styling Configuration

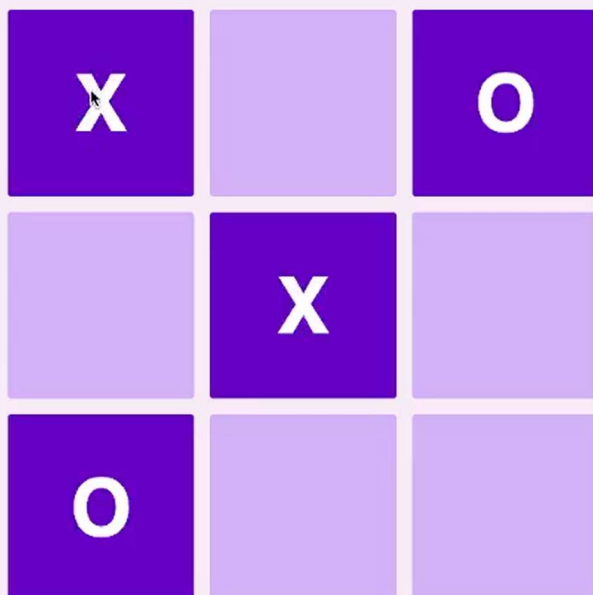
- Styling der Inhalte im Overlay
- Styling section „game-configuration“
- Erstelle dazu das configuration.css und binde es in der index.html ein



Styling Game Field

- Styling Section „active-game“
- Erstelle dazu das games.css und binde es in der index.html ein
- Achtung! Spielfeld wird nur eingeblendet, wenn Spiel gestartet wurde
- Wenn man über Feld fährt, soll sich Farbe ändern
- Wenn Feld bereits einen Wert hat, soll das Feld disabled sein

It's your turn **PLAYER NAME!**



Exercise 4 – Logik hinzufügen

Nun müssen wir die Logik mit JavaScript entwickeln. Erstelle dazu Dateien:

- app.js
 - o Allgemeines JS-File mit grundlegenden Daten
- config.js
 - o Die Logik rund um die Player-Configuration
- game.js
 - o Logik rund um das Spiel selbst (Tic-Tac-Toe Regeln)

Natürlich könnte man es auch anders abbilden. Damit wir jedoch eine ähnliche Struktur beibehalten, bitte so durchführen. Ziel ist es, die JS Files möglichst schlank zu halten und eine logische Trennung zu erreichen.

Player Configuration:

- Öffne das Modal (Player-Config), wenn man auf „Edit“ klickt.
- Bei Klick auf „Close“ soll sich das Modal schließen
- Spielernamen setzen mit JS
 - o Recherchiere im Web: preventDefault, FormData, data- Attribut

Optionale Tasks

- o Validiere den Name Input

Game logic

Implementiere die Game Logik!