

WBE-Praktikum 1

Grafiken im Browser

Einführung

In diesem ersten Praktikum geht es vor allem darum, die HTML- und CSS-Vorkenntnisse¹ für ein kleines Projekt einzusetzen, welches später im Semester fortgesetzt werden soll. Konkret geht es um das Spiel „*Vier gewinnt*“, für das in diesem Praktikum eine statische Grafikausgabe implementiert wird, welche in späteren Lektionen um die Spiellogik ergänzt wird. Nebenbei sollen in dieser Lektion verschiedene Möglichkeiten, Grafiken im Browser zu erzeugen, angesehen werden.

Dokumentation Praktikumsergebnisse

Wir empfehlen Ihnen, ein Projekt unter <https://github.zhaw.ch> zum WBE-Praktikum anzulegen, in welchem sie die Ergebnisse der Praktikumslektionen dieses Semesters dokumentieren.

Abgabe der Praktikumsergebnisse

Die Ergebnisse einzelner Praktikumsaufgaben sollen abgegeben werden. Diese Aufgaben haben im Titel jeweils den Hinweis „(Abgabe)“. In den einzelnen Klassen kann es unterschiedliche Abgabemodalitäten geben.

Auch wenn nicht alle Ergebnisse abzugeben sind, empfiehlt es sich, auch die anderen Aufgaben zu lösen, denn die gehören selbstverständlich ebenfalls zum Prüfungsstoff.

Die Semesternote setzt sich aus diesen vier (gleich gewichteten) Teilen zusammen:

- Die erwähnten Abgaben zu Praktikumsaufgaben
- Das Miniprojekt „*Vier gewinnt*“, das erst gegen Semesterende abgegeben werden soll
- Zwei Kurztests (Moodle)

¹ Bei dieser Gelegenheit sei noch einmal an den Vorbereitungskurs im Moodle erinnert:
<https://moodle.zhaw.ch/course/view.php?id=1007>

Entwicklungsumgebung

Neben den unten beschriebenen Aufgaben zum Start des Miniprojekts ist es wichtig, eine für Webprojekte sinnvolle Infrastruktur einzurichten und die verschiedenen Werkzeuge in diesem Bereich zu kennen. Für den Fall, dass hier noch Nachholbedarf besteht, sind die wichtigsten Tools und ihr Einsatz in einem eigenen Dokument *entwicklungsumgebung.pdf* zusammengefasst. Wir empfehlen Ihnen, dieses Dokument mindestens durchzuschauen und Ihre Werkzeugsammlung und Ihr Wissen bei Bedarf zu ergänzen. Es geht darin um folgende Punkte:

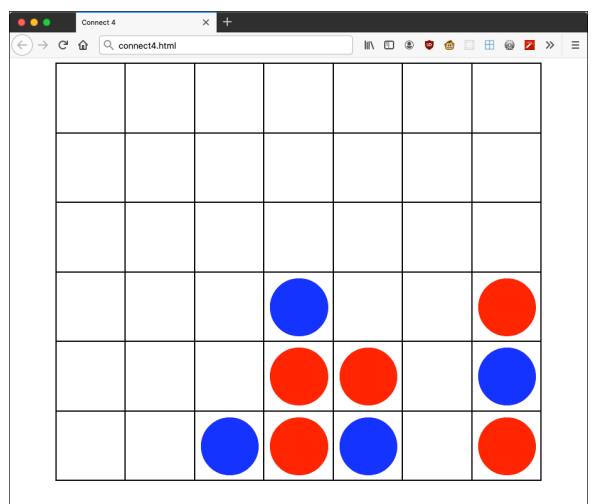
- Browser und ihre experimentellen Vorabversionen
- Developer Tools der Browser
- Code-Editor und geeignete Erweiterungen
- SSH- und SFTP-Tools (GUI und Kommandozeile)
- HTTP-Request testweise auf der Kommandozeile ausführen, sowie Request- und Response-Header in der Browser-Tools untersuchen
- Installation von Node.js

Miniprojekt

Das Miniprojekt, das in mehreren Praktikumsaufgaben in der ersten Semesterwoche sowie in der zweiten Semesterhälfte durchgeführt wird, kann in Zweierteams durchgeführt werden und soll schliesslich auf GitHub Pages zusammen mit einer kurzen Dokumentation verfügbar gemacht werden. Es wird am Ende des Semesters bewertet und fließt damit in die Semesterbewertung ein.

Aufgabe 1: Vier gewinnt (Miniprojekt)

„**Vier gewinnt** (englisch: **Connect Four** oder **Captain's mistress**) ist ein Zweipersonen-Strategiespiel mit dem Ziel, als Erster vier der eigenen Spielsteine in eine Linie zu bringen. [...] **Regeln:** Das Spiel wird auf einem senkrecht stehenden hohlen Spielbrett gespielt, in das die Spieler abwechselnd ihre Spielsteine fallen lassen. Das Spielbrett besteht aus sieben Spalten (senkrecht) und sechs Reihen (waagrecht). Jeder Spieler besitzt 21 gleichfarbige Spielsteine. Wenn ein Spieler einen Spielstein in eine Spalte fallen lässt, besetzt dieser den untersten freien Platz der Spalte. Gewinner ist der Spieler, der es als erster schafft, vier oder mehr seiner Spielsteine waagerecht, senkrecht oder diagonal in eine Linie zu bringen. [...]” (Wikipedia)



Das Erzeugen der Grafik im Browserfenster (also dem Viewport) kann auf unterschiedliche Weise umgesetzt werden. Statische Bilder im JPEG- oder PNG-Format helfen hier nicht, denn diese sind nicht ohne weiteres clientseitig anpassbar. Die könnten allenfalls als Hintergrund eines Canvas-Elements eingesetzt werden, aber dann kann auch die ganze Grafik mit Canvas erstellt werden.

Somit bleiben im Wesentlichen diese Lösungsansätze:

- Scalable Vector Graphics (SVG)
- Canvas-Element zum Erzeugen einer Bitmap-Grafik
- Möglichkeiten von HTML und CSS

Alle drei Varianten sollen nun kurz getestet werden. Hinweis: Für SVG und Canvas ist das eine Vorschau, denn diese Ansätze werden im Verlauf des Semesters im Unterricht noch aufgegriffen.

Scalable Vector Graphics (SVG)

Wie HTML basiert SVG auf Tags als Markup. Es ist eine XML-Sprache, deren Tags grafische Formen beschreiben. Im SVG-Format beschriebene Grafiken können einfach in ein HTML-Dokument eingefügt werden:

```
<html>
  ...
  <svg>...</svg>
  ...
</html>
```

Damit sind die in SVG beschriebenen Grafikelemente ins DOM (Document Object Model, mehr dazu später) integriert und via JavaScript zugreifbar.

Einfache SVG-Grafiken können von Hand im Code-Editor erstellt werden. Für das Spielfeld von „*Vier gewinnt*“ wäre das sicher möglich. Eine andere Variante ist das Erstellen mit Hilfe einer dafür geeigneten Applikation wie Inkscape², ein mächtiges Tool, dessen Bedienung aber etwas gewöhnungsbedürftig ist. Das Dokument *connect4.svg* aus den Beispielen zu diesem Praktikum wurde mit Inkscape erstellt.

Aufgaben:

- Falls Sie Inkscape installiert haben: Versuchen Sie einmal, ein „*Vier gewinnt*“-Spielfeld damit zu zeichnen. Alternativ öffnen Sie *connect4-inkscape.svg* und machen ein paar Versuche damit (Stärke der Linien, Farben, ...).

² <https://inkscape.org>

- Sehen Sie sich den Source-Code von *connect4.svg* an und identifizieren Sie wichtigsten Elemente des Spielfelds.
- Kopieren Sie das *svg*-Element aus *connect4.svg* in ein ansonsten leeres HTML-Dokument und sehen Sie sich das Ergebnis im Browser an. Testen Sie, ob die Anzeige des Spielfelds im Browser ohne Qualitätsverlust skalierbar ist.
- In den Developer Tools im Browser können Sie sehen, wie das SVG-Element in die Elementhierarchie des HTML-Dokuments integriert ist. Es lassen sich auch die einzelnen Teile der SVG-Grafik auswählen.

Canvas-Element

„Ein Canvas-Element (vom englischen canvas für „Leinwand“ oder „Gemälde“) ist ein – in der Sprache HTML – mit Höhen- und Breiten-Angaben beschriebener Bereich, in den per JavaScript gezeichnet werden kann. Ursprünglich von der Firma Apple als Bestandteil des WebKit entwickelt, ist es später von der Arbeitsgruppe WHATWG als Bestandteil der Auszeichnungssprache HTML5 standardisiert worden.“ (Wikipedia)

Aufgaben:

- Öffnen Sie die Datei *canvas.html* im Code-Editor. Auch wenn viele Details zu JavaScript erst noch behandelt werden, ist zu sehen, dass hier mit JavaScript-Aufrufen in einen Canvas-Bereich gezeichnet wird. Die Linien des Spielfelds werden nun mit *for*-Schleifen angelegt.
- Öffnen Sie die Datei im Browser. Da in diesem Fall Bitmap-Grafiken erzeugt werden, ist ein Vergrößern ohne Qualitätsverlust nicht zu erwarten. Probieren Sie es aus.
- Passen Sie das Script so an, dass wie im SVG-Beispiel im zweiten Feld der obersten Reihe noch eine rote Spielfigur angezeigt wird.

HTML und CSS

Nach SVG und Canvas sehen wir noch eine weitere Variante an: für einfache Grafiken wie unser Spielfeld reichen sogar die Möglichkeiten von HTML und CSS aus.³ Dies ist auch die Variante, die wir der Einfachheit halber wir für das Miniprojekt empfehlen. Zumindest bis die ganze Logik funktioniert. Was natürlich nicht ausschliesst, dass Sie am Ende noch ein schöneres GUI mit Hilfe von SVG oder Canvas anbauen.

In der Datei *connect4.html* finden Sie ein Beispiel, wie das gehen könnte. Das CSS ist zunächst mit im HTML-Code, sollte später aber in eine separate CSS-Datei ausgelagert werden.

³ Es soll nicht verschwiegen werden, dass es eigentlich nicht im Sinne von HTML ist, damit Grafiken aufzubauen. Wir wollen das hier für einmal nicht so genau nehmen...

Aufgaben:

- Betrachten Sie den Beispielcode in *connect4.html* im Editor. Hier sind zunächst nur zwei Reihen à sieben Feldern vorgesehen. Bis auf das erste Feld sind aber alle ausgeblendet.
- Sehen Sie sich die Ausgabe im Browser-Fenster an. Welchem HTML-Element entspricht der blaue Kreis und welche CSS-Eigenschaften werden für seine Darstellung verwendet? Wie wird erreicht, dass sich die Grösse des Spielfelds an der Grösse des Viewports orientiert?
- Entfernen Sie die CSS-Angaben, welche dafür sorgen, dass nur das erste Feld angezeigt wird. Da *div* ein Blockelement ist, werden die Felder nun untereinander dargestellt.
- Passen Sie das CSS so an, dass zwei Zeilen à sieben Feldern angezeigt werden. Am besten lösen Sie das mit fließenden Boxen (*float*). Wenn Sie Flexbox oder CSS-Grid kennen, können Sie auch eine von diesen Varianten verwenden – in diesem einfachen Beispiel sind alle drei Varianten nicht sehr aufwändig.
- Um das gesamte Spielfeld mit sechs Zeilen à sieben Feldern anzuzeigen, muss nun nur noch der HTML-Code erweitert werden.

Damit haben wir nun eine Anzeige des Spielfelds für „*Vier gewinnt*“. Die Aufgabe, dieses dynamisch zu generieren, verschieben wir auf eine spätere Lektion ☺.