Web Applications for Technology Enhanced Learning

Winter term 2017/18
Sebastian Mader
Topics of the practical course

  ○ “Technology Enhanced Learning”
    ■ How can technology be used to enhance learning and / or teaching?
  ○ “Collaboration”
    ■ How can technology be used to bring people together to enhance their learning and / or teaching efficiency?
Goals

● Develop a web application in a team of three to four members
● Learn about the current state of JavaScript development
Criteria for Passing

- Prepare and give a talk about a JavaScript framework / library / programming technique (see topics later)
  - about 30 minutes, questions / discussion afterwards
- Attendance in all plenum sessions and tutor meetings
- Develop a JavaScript application in a team of four members
- Present the developed application at the end of the term for the whole course
- Oral examination including a demonstration of the developed application
Next Steps: Group Formation

- Form groups consisting of **three to four members**
- Find a group name that is not a combination of your initials
- Find a date where each team member has time for an **one hour long bi-weekly tutor meeting**
  - Mondays, 08 - 16, not from 12-13
  - Tuesdays, 08 - 16, not from 12-13
  - Tutor meetings start beginning from next week (23.10 / 24.10)
Course of the Practical

- Starting from today, each team should begin to **search for an idea** for the application
  - Going to introduce four ideas from the research unit
    - if your team takes one of the given ideas, it has to be **extended with own ideas**!
  - Own ideas are welcome!
    - can be discussed in the **tutor meeting next week**
- In three weeks time, the concept for the application should be finished and will be talked about in the tutor meeting
- From then on: You can start coding
  - Requirements: JavaScript only.
Synchronized Automaton Simulator

Automatons (finite automatons, turing machines, petri nets, ...) are something that cannot be really be worked on collaboratively with a greater number of people.

- A web application where a small group of people can work together on an automaton, test it, fix it and talk about its functionality.

Challenges

- **Synchronization**: What to do, if two collaborators made a change to the same element?
- **Simulation**: Depending on the type of automaton, writing the actual simulation can be a challenging task.
- **Interface**: How are changes of other users communicated, so that every user still understands what happens?
Quizzes (or small units to test knowledge) are a way to check one’s own learning progress.

- A quiz application that resembles a real game - not just a collection of quizzes with scores and leaderboards.

Challenges

- **Concept**: A concept and game idea that is suitable for a quiz game has to be developed.
A real REPL

A real REPL: Many platforms offer functionality to compile code, but REPL functionality, where one can get things instantly evaluated is non-existent for most programming languages (Haskell, Scala, Java, Prolog, ...)

- A REPL for a programming language that can be used within the browser.

Challenges

- **Implementation**: The REPL has to run somewhere on a server, and run as long as the user interacts with it.
- **Security concerns**: Users should not be able to do malicious things using the REPL.
- **Collaboration**: What kind of collaboration is offered?
“Places to Learn”

Learning is a social activity and finding free rooms at LMU and people to learn with can be a rather arduous task.

- An application, where students can record, where and what they are currently learning and lets other people see, where people are learning, and if they could potentially help someone

Challenges

- **Maps**: How can maps be displayed?
- **Localisation**: How does the user determine, where he currently is?
- **Concept**: How are users motivated to help each other and use the application in general?
Topics for Talks

- **Frontend Frameworks**
  - React
  - Vue.js
  - Angular
  - Redux
  - MobX

- **Backend Frameworks**
  - Socket.IO
  - Express
  - Koa

- **Testing and Debugging**
  - Debugging JavaScript
  - Unit Testing
  - End-to-End Testing

- **Miscellaneous**
  - Flow
  - TypeScript
  - WebComponents / Polymer
  - JavaScript Design Patterns
  - GraphQL
  - RxJS
  - D3.js
  - Meteor
  - JavaScript’s Event Loop
Schedule  Group Dates

- Tutor meeting next week (23.10.2017, 24.10.2017)
  - Have a rough idea, what kind of application your group wants to implement
- Tutor meeting two weeks later (06.11.2017, 07.11.2017)
  - elaborated concept with mock-ups
  - if your team took one of the given ideas: own ideas and extensions have to be included!
- From then one: bi-weekly tutor meetings to discuss current progress, problems, and challenges
Schedule

Plenum Dates

- 25.10: JavaScript: Client and Server
- 01.11: public holiday
- 08.11: JavaScript: Bundling JavaScript, Asynchronous JavaScript
- starting from 14.11: student talks
- 24.01. / 31.01: final presentations