

CS 520

Theory and Practice of Software Engineering Fall 2018

Collaboration and pair programming

September 27, 2018

Coming up

- Next week:
 - Guest lecture on Wednesday, October 3, 4-5PM in CS 151
 - **Please attend.** (if you have a time conflict – we'll post a video)
 - No class on Thursday, October 4
- Homework 1 posted (due Oct 16): <https://people.cs.umass.edu/~brun/class/2018Fall/CS520/hw1.pdf>
- Final project assignment posted



Helena: A Web Automation Language for End Users

Wednesday, October 3
4:00 pm
Computer Science Building, Rm. 151

All lectures are free and open to the public.

Web data is revolutionizing the social sciences. Researchers envision a diverse range of datasets facilitated by the unique properties of web data... (text continues)

[cics.umass.edu/rising-stars](https://people.cs.umass.edu/~brun/class/2018Fall/CS520/hw1.pdf)

CS 520 Homework 1 Code review and Model-View-Controller in action

Due: Tuesday, October 16, 2018, 9:00 AM EDT via Blackboard. You may work with others on this assignment but each student must submit their own, individual write up and code, clearly specifying the collaborators. The write up and code should be individual, not created jointly, and written by the student on their own. Late assignments will not be accepted without prior permission.

Overview and goal

The goal of this assignment is to code review, redesign, and reimplement a Tic Tac Toe game, according to the model-view-controller (MVC) architectural pattern.

The following repository provides a basic implementation of the Tic Tac Toe game: <https://github.com/brun/cs520-hw1>

The quick-and-dirty implementation violates many best practices and needs a major design overhaul. In contrast to the current version, your implementation should support a possible extension without violating the open-closed principle. Additionally, your implementation should enable better testability so that the individual components can be tested in isolation.

Code review

You are expected to code-review the current version of the application. In particular, you are expected to identify **three (3) violations** of best practices. For each violation, briefly explain what the issue is (you may refer to general principles or your design choices with respect to the desired testability) and how to fix it. Your code review should follow the following pattern for each identified violation:

- Brief summary (a few keywords)
- What's the issue (one or two sentences)
- How to fix it (one or two sentences)

Your implementation should reflect and implement your code review suggestions.

Implementation

Dataset consistency

Your implementation must implement the rules of tic tac toe. It must, in that sense, be a correct implementation of the rules.

Your design needs to support the following extensions:

- Using a different visualization (View) for the game board without changing the model or the controller.
- Using a different data representation (Model) without changing the view or the controller.
- Adding game moves from other sources (e.g., loading from the command line).

CS 520 Fall 2018, Homework 1, Code review and Model-View-Controller in action Page 1 of 3

Homework 1

- **Due** : October 16
- **Goal**: Code review, (re)design, and (re)implementation.



Player 2 to play 'O'

Can discuss and work together, but submissions (code, write up, etc.) are individual.

CS 520 Final project description

Final projects will be completed in teams of 4 or 5 students. Each team is responsible for a single project. You should select a team and a project by **Tuesday, October 9, 2018, 9:00AM EDT**. Your mid-point check-in will be on **Tuesday, November 13, 2018, 9:00AM EST**. The final project will be due **Tuesday, December 11, 2018, 11:55 PM EST**. There are five options for a final project (each team will do one):

1. MSR 2019 Mining Challenge
2. Replication study
3. Model Inference for Inferring Processes
4. Elna: Elevation-based Navigation
5. Self-defined software engineering research project

MSR 2019 Mining Challenge

The Mining Software Repositories conference runs an annual challenge in which they provide a dataset and ask you to answer research questions about the dataset. Read the description of this year's dataset, research questions, and challenge here: <https://2019.msrconf.org/track/track/2019-Mining-Challenge#11-for-Mining-Challenge-Paperx>

Replication study

A replication study takes an existing research paper, replicates its experiments on the same data, and then extends the experiments to expanding that data set or which the experiments are run. For this project, we highly recommend selecting a paper with publicly available dataset and code to execute the experiments. The project involves a write up describing the process of replicating the experiment, deviations in the achieved results from the original ones reported in the paper, and lessons learned from applying the experiments to new data.

Here is a list of several papers that are good candidates for replication:

1. Automatic generation of oracles for exceptional behaviors from Javac's comments.
Paper: <https://dl.acm.org/citation.cfm?id=3232414>
Source code: <https://github.com/albertogoffi/ctad4code>
2. SimFix: Automated program repair
Paper: <https://se.inf.ku.edu.cn/~xinyu/f04/papers/15DTA1a.pdf>
Source code: <https://github.com/xpam11aboy/SimFix>
Dataset: <https://github.com/xpam11aboy/SimFix>

Coming up

- Next week:
 - Guest lecture on Wednesday, October 3, 4-5PM in CS 151
 - **Please attend.** (if you have a time conflict – we'll post a video)
 - No class on Thursday, October 4
- Homework 1 posted (due Oct 16): <https://people.cs.umass.edu/~brun/class/2018Fall/CS520/hw1.pdf>
- Final project assignment posted

Agile development

- Fast paced
- Frequent releases
- Developer centered
 - do we need managers?

Scrum

- A very popular flavor of agile
- Three pillars:
 - transparency
 - inspection
 - adaptation

Three roles

- Product owner
 - represents the customer
- Development team
 - performs sprints
 - delivers software product
- Scrum master
 - Buffer between team and outside world
 - Prevents distractions, barriers

Many aspects of Scrum

- Sprints
- Scrums
- Stand-up meetings
 - what did I do yesterday?
 - what will I do today?
 - do I see any impediment from our goal?
- Reviews

Pair programming

- Coding, testing, designing, etc.
- Pair-work facilitates
 - transparency
 - no single point of failure
 - decision making
 - focus

Everything's better in pairs

Collaboration Exercise

- An exercise game for learning about collaboration
 - Developed by Laurie Williams and Lucas Layman at NCSU
- ©Williams and Layman 2007

NC STATE UNIVERSITY

http://www.youtube.com/watch?v=rG_U12uqRhE



UMassAmherst College of Information & Computer Sciences

RISING STARS IN COMPUTER SCIENCE 2018

Cutting edge research from the brightest rising stars in our field

Sarah Chasins
PhD Candidate, University of California Berkeley

Helena: A Web Automation Language for End Users

**Wednesday, October 3
4:00 pm
Computer Science Building,
Rm. 151**

All lectures are free and open to the public.

Web data is revolutionizing the social sciences. Researchers envision a diverse range of models facilitated by the unique properties of web data — its scale, ecological validity, connectedness. With the wide variety of web scraping libraries, no other programmers have access to accessing and gain insight for analyzing web data. However, these libraries are hard-coded to their programming, and engineering new programs to collect these datasets is a time-consuming task. To demonstrate access to web data, we designed the Helena web automation language. Helena brings together the following key innovations, which together empower end users to write robust web scraping programs. (1) The Helena programming environment uses Programming by Demonstration (PbD), which makes this task to derive the code with a single-point learning approach, creating scripts based on recording a single interaction of the user with a set of webpages. Intuitively, users can learn the tool and use it to write a robust large-scale scraping script in under 10 minutes, while programmers lacking the same skills with the traditional domain languages take an hour. (2) Helena's adaptive compiler makes scripts robust to webpage redesigns and obfuscation, which enables longitudinal experiments. (3) Helena's novel runtime can parallelize and distribute scraping programs to thousands of VMs, facilitating large-scale scraping. Our approach relied on novel insights into the web scraping domain but also on long-standing techniques to learn by combining techniques from the Programming Language community and the Human-Computer Interaction community, we arrived at a language design that meets our user needs.

ctic.umass.edu/rising-stars