

Git-Based Platform for Distributed Learning Communities

by

Vahid Fazel-Rezai

Submitted to the Department of Electrical Engineering and Computer
Science

in partial fulfillment of the requirements for the degree of

Master of Engineering in Computer Science and Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2018

© Vahid Fazel-Rezai, MMXVIII. All rights reserved.

The author hereby grants to MIT permission to reproduce and to
distribute publicly paper and electronic copies of this thesis document
in whole or in part in any medium now known or hereafter created.

Author

Department of Electrical Engineering and Computer Science

May 25, 2018

Certified by

Philipp Schmidt

Research Scientist, Director Media Lab Learning Initiative

Thesis Supervisor

Accepted by

Katrina LaCurts

Chairman, Department Committee on Graduate Theses

Git-Based Platform for Distributed Learning Communities

by

Vahid Fazel-Rezai

Submitted to the Department of Electrical Engineering and Computer Science
on May 25, 2018, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Computer Science and Engineering

Abstract

Instructors use software called learning management systems to administer online courses. Traditionally these digital platforms mimic physical classrooms, but new technology opens up potential for interactions not previously possible. In this thesis, we build a learning management system on top of git to create a learning environment that encourages collaboration and exploration. We discuss how it was designed following established learning principles, the technical implementation, and the results of a pilot online class, with 80 learners lasting 6 weeks, as part of the Refugee Learning Accelerator. The platform architecture used in this study can be replicated and reused for other online courses with similar goals.

Thesis Supervisor: Philipp Schmidt

Title: Research Scientist, Director Media Lab Learning Initiative

Acknowledgments

To do...

Contents

1	Introduction	13
1.1	Refugee Learning Accelerator	14
1.2	Methodology	15
2	Background	17
2.1	Traditional Learning Platforms	17
2.2	Foundational Theory	18
2.3	Pedagogical tensions	18
3	Related Work	19
3.1	Overview of git	19
3.2	Communities of practice	19
3.3	Other uses of git	20
3.4	Existing git-based learning systems	20
4	Platform Development	21
4.1	Design	21
4.1.1	Balancing tensions	21
4.1.2	Suitability of git	22
4.1.3	Platform architecture	23
4.2	Third-party Software	24
4.2.1	Git hosting service	24
4.2.2	Chat service	25

4.2.3	Server infrastructure	26
4.3	Technical Implementation	26
5	Results of RLA Pilot	27
5.1	User Experience	27
6	Conclusion	29
A	RLA Challenge Descriptions	31
B	RLA Online Course Survey	35

List of Figures

5-1	The course-central repository.	28
-----	--	----

List of Tables

Chapter 1

Introduction

New education technology, enabled by the Internet, has allowed learners to connect with their instructors and with each other digitally. Pieces of software known as learning management systems take the experience of a classroom online, giving remote learners more access to education and enriching their learning experiences with a variety of technology.

In this thesis I employ git, a version control technology initially created for collaboration among software developers, as the foundation for an experimental learning platform. The decentralized nature of git can be utilized to expand interactions beyond replicating that which is found in traditional course. New interactions can be designed to balance the benefits of a collective learning experience with those of independent open-ended projects. In this study, I develop a prototype of such a platform and examine the results of a pilot run in an online course. As a result of this work, I hope to contribute to lowering the barrier of entry to administer an online course using a git-based learning platform. The design decisions map out an architecture that can be built upon in future courses.

The following sections outline the approach taken towards this study. Chapters 2 and 3 cover the established theory and other similar projects that were taken as guidance in designing the new platform. Chapter 4 steps through the design and implementation of the platform; section 4.3 is the only section in this thesis that assumes a technical background. Finally, results of a pilot run of using the platform

in an online course are presented in chapter 5.

1.1 Refugee Learning Accelerator

The development of a git-based learning platform is part of a larger project called the Refugee Learning Accelerator (RLA). RLA is a program run by the Learning Initiative, a group of the MIT Media Lab. The goal of the program is to gather and support technologists from the Middle East to work on education tools for refugee learners.

The program ran in three phases from fall of 2017 through spring of 2018. First, interested candidates applied as teams through an open application. Around 100 participants were selected for the first phase of the program, which consisted of a 6-week online course during November and December of 2018. Every two weeks of the course, a new technology was introduced and teams were challenged to design, and ideally build a prototype for, an education solution using that technology. Webinars held throughout the 6 weeks helped participants learn about the technologies. Additionally, teams received constructive feedback every week, once at the midpoint and once at the end of each challenge. The topics chosen for the online course were chatbots, human centered design, and AR/VR. A description of each challenge is included in Appendix A.

The second phase of the program brought 14 selected teams together for a week-long, in-person workshop in Amman, Jordan. There, teams forged their project ideas through rapid iterations of feedback and field observation. One important aspect of the workshop was also the building of a supportive community, where members encourage one another to pursue their projects and help in connecting each other to resources.

Following the workshop, teams worked independently to bring their projects to fruition in partnership with local NGOs and with the continued support of the RLA community. [11]

RLA is the product of the collective effort by members of the Learning Initiative

and others passionate about finding creative ways to nurture learning, including the motivated participants, many of whom were working on their projects meanwhile also taking on a full-time load at work or as a student. The team of instructors included researchers and students from MIT and other affiliated experts. Mentors and speakers from all backgrounds donated their time to assisting teams through the webinars and at the workshop. The development of a learning platform was just one component in the operation of creating and executing the RLA program.

While one of my goals in building the platform was certainly to support the online course of RLA, I also hoped to explore the possibilities in its design. In this way, the online course of RLA served also as a pilot run for the platform in order to evaluate its utility and shortcomings.

1.2 Methodology

Ultimately the research question I hope to answer in this thesis is: how can git be adapted to support effective online learning?

This can be covered in three parts:

- First, I turn to established theory on digital learning pedagogy and principles to guide the characteristics and ultimately define interactions of the platform. In particular, I derive these interactions from first principles of learning and target learning outcomes rather than by attempting to move traditional classroom interactions online.
- Second, I implement a technical platform that supports these interactions. The bulk of this work is done before the pilot course, but also includes the technical operation of the course. As much as possible, I rely on third-party software to increase reliability, reproducibility, and familiarity. Creating a learning platform from scratch is beyond the scope of this work.
- Third, I use the pilot run to evaluate not only how these interactions work in the manifested platform, but also how well they work. In order to gather feedback,

I conducted an anonymous survey among participants in the online course. This consisted of a form send by email, the questions of which are included in Appendix B. Responses to the survey as well as anecdotal evaluation are reported in Chapter 5.

The code and course content are all freely available online for use in future iterations of such research. [11]

Chapter 2

Background

Educational theory and pedagogy is a vast field and far outdates the advent of online education. Here we briefly touch on frameworks that went into consideration for this thesis.

2.1 Traditional Learning Platforms

An online platform to support learning is not a new concept. A learning management system (LMS) is a tool that helps instructors supervise an online course, and popular commercial LMSs exist such as Blackboard, Moodle, and Canvas.

A traditional LMS platform typically provides features including tracking learner grades, file management for assignments and corresponding submissions, and communication mechanisms for instructor announcements or learner questions. These features are often more convenient duplicates of the same in-person interactions of traditional classrooms. Furthermore, LMS usually don't fully encompass the activities of a class, leaving the core of the learning for classroom instruction or learner self-study.

Of note, the issue of learner collaboration, which is often a component of more creative, project-based assignments, is often not properly addressed in traditional LMS platforms. This intersection of collaboration and technology is instead known as computer-supported collaborative work (CSCW), and families of tools exist to

support online collaboration. Learners thus would have to take to these external tools to collaborate, depending on the particular assignment.

2.2 Foundational Theory

We begin with the assumption that humans have innate desire and ability to pursue their interests, and the role of the learning environment is to foster the right conditions. To break down this idea, we rely on

To do...

2.3 Pedagogical tensions

To do...

Chapter 3

Related Work

We set out to develop a new platform to address

3.1 Overview of git

A specific technology used for online collaboration, particularly in software engineering, is git. Created in 2005 by Linus Torvalds as a tool to support the development of the Linux kernel, git allows users to each have a local copy of a set of files and sync between their copy and a central server in a coordinated fashion. Git is also a version control system, hence allowing users to backtrack changes or split into multiple branches of work from the same starting state. [git]

Git is used as a command line tool, in which users type text commands to perform the desired git actions. In addition, there are graphical interfaces available which serve as a wrapper and aid for users who prefer not to use the command line. Overall, git is considered a powerful collaboration tool, but involves an upfront investment to learn its operations.

3.2 Communities of practice

Use of group chats such as gitter and mattermost to complement projects

3.3 Other uses of git

While git was initially created for and has been largely relied on by the software development community, its use has spread to other knowledge workers.

An example of applying git to a traditional use case is cookbooks: a service called Fork the Cookbook exists in which users can fork, or copy, an existing recipe and tweak it to make it their own. This creates a collaborative and decentralized platform for sharing cooking recipes. [1]

Example applications include recipes, legislation, datasets, general CMS, etc.

3.4 Existing git-based learning systems

Chapter 4

Platform Development

4.1 Design

In this section we lay out the high-level design for the RLA course. The git-based platform is developed as just one of the components of this course, but is always intended to be applicable outside the context of RLA to other future courses.

4.1.1 Balancing tensions

We also aimed to foster a community that would extend into the later phases of the Accelerator.

Finally, we rely on good software development practices and use popular tools where possible.

Communication. Communication lays the groundwork for the type of community formed. We aim to have instructors clearly communicate how to use the platform and be able to guide students throughout the course. We would like to allow natural and structured opportunities to ask questions and give feedback. General course information should be readily accessible to students.

Permissions. We will have to decide how to control authority to perform actions and visibility of content. On one hand, we believe that the platform should be as open as possible. This would allow students the opportunity to pursue their curiosities, to

learn from each other's work, and to feel that they have a complete understanding of the platform. On the other hand, we would like to prevent either intentional or accidental actions that may destroy or hinder other participants' work. We also anticipate that sensitive topics would not be expressed without a private way of doing so and thus would like to allow for that.

4.1.2 Suitability of git

While git was not designed as a learning platform, we found in it suitable qualities. Here we discuss the benefits of git as a tool for fostering positive educational interactions.

As a version control system, the currency is units of work rather than final deliverables. It is oriented around building larger projects through a series of work. Git gives learners room to play, since branches can be made to explore new ideas and unsuccessful changes can be easily reverted. It is also very much designed for collaborating with peers. After all, its *raison d'être* is to be able to build upon each others' work without conflict. Documenting everyone's work openly and allowing collaboration creates a shared space from which a community could form, just as many have in the open source world. It gives flexibility in what tools people can use for their work, since all file types are supported. Finally, because git is already widely-used, its kinks have been worked out and help material is readily available online, making it easier for new adopters.

As discussed in the previous section, git can also be adapted as the underlying mechanism of a content management system. As a versioning system, git enables publishing new content, updates, and revisions in a fashion that can be distributed. Popular git hosting websites such as Github and GitLab have built-in support for Markdown-formatted files and support and static web pages.

4.1.3 Platform architecture

The details of the design were chosen to support the needs of RLA’s weekly cycle. In each week of the cycle, content is released at the beginning and learners worked towards a particular milestone by the end of the week. Throughout the week learners can collaborate and participate in seminars.

To achieve these requirements, there are four interactions that the core git-based platform needs to support:

- **Onboarding.** We want learners to be able to register for the course as easily as possible, yet still provide basic information to introduce themselves to the rest of the learner community.
- **Content distribution.** New material posted at the beginning of the week includes a description of the week’s challenge and targeted milestones. In addition, relevant resources relevant for the week are gathered and provided for learners’ reference. This content may also sometimes include skeleton code or template documents in which learners can report their progress. These pieces of content require the additional ability of being pulled into learners’ working directories.
- **Submission.** At the end of the week, learners need a mechanism to put together a package representing the state of their work on which to receive feedback. We also encourage learners to use these submissions as a resource to learn from other learners’ approach to the challenge.
- **Feedback.** We believe frequent personalized feedback can be helpful to learners and also cultivates a culture of collaboration. Hence we’d like our platform to provide an interface where a reviewer can easily peruse a learner’s work and leave comments. Note that this process is similar in structure to code review used in software engineering, so we can utilize similar features in the git hosting service.

In addition to managing the course content, we also want to support a few forms of communication that empower learners:

- For seminars, we need a way to broadcast video to learners. Additionally, we would like to engage learners to drive the conversation according to their passions and have face-time with their peers.
- For technical support and other miscellaneous discussion, we would like to have an open, real-time, group chat. In particular, the group aspect will expose the entire community to conversations as appropriate so that everyone feels included and in-the-know.
- For showcasing teams' work, especially to parties outside the course, it would be ideal to have a static website summarizing the challenges and the progress coming out of the course. Anyone could then share the link and in this way feel ownership over their work.

4.2 Third-party Software

4.2.1 Git hosting service

In order to have a group of distributed learners participate in the course using git, we needed to have a server accessible by the Internet running git where the content would be hosted. We looked at two of the most popular git hosting services, Github and GitLab. Both offer a SaaS model where they host git on their servers and simply provide an interface, as well as a self-hosted version where they provide the software but leave running a server to the customer.

We compared options the two in terms of features, styles, and technical performance.

The two options had very similar sets of features. Github has organizations, teams, repositories, and pull requests, while GitLab has groups, subgroups, projects, and merge requests. Both expose a webhook for git commits, allow in-browser editing of

files, and can serve hosted files as static websites. Both offer free or discounted private repositories/projects for educational uses. On a few points GitLab had additional features, such as being able to request to join a group and being able to trigger scripts that run before serving static websites.

Many software companies use Github to host either all their code or even just their open source projects, and as a result a Github profile feels very much like a professional portfolio. We wanted to make sure learners are comfortable experimenting and making mistakes, and took into consideration that some may feel less inclined to do so if the work would permanently be associated with their professional portfolio.

Finally, we took into consideration the technical reliability of each option. We found that GitLab's SaaS offering, GitLab.com, was often slow to load. For both, a self-hosted instance would include considerable setup and costs to run a server, but would allow customizations such as the domain name. Finally, though both are welcoming homes to open source projects, Gitlab has made the web platform itself open source, whereas Github's is not.

For our course, we ended up opting for GitLab, and chose to mitigate performance issues by using a self-hosted instance of the free tier. However, it would certainly be possible to adapt our approach for either GitLab or Github, in the both SaaS and self-hosted models.

4.2.2 Chat service

To allow the development of a community, we wanted to make communication among learners as well as between learners and instructors as open as possible. GitLab alone, with commit messages and comments on merge requests, did not offer as fluid of a setting as we wanted. We decided to augment GitLab with a team-based chat service where learners could communicate in real time.

We looked at existing cloud messaging services including Slack, Gitter, and Mattermost, all of which have a very similar design. They provide open channels where any user can join and chat, as well as messaging between users and in private groups for more direct communication. We hoped to use these spaces as a forum for dis-

cussing ideas, asking each other for help, and even just to meet other learners.

From our options we chose to run a self-hosted instance of Mattermost, largely due to our choice of self-hosted GitLab. Mattermost and GitLab have a built-in integration for authentication and backend configuration. Mattermost is also open source, continuing in our spirit of relying on open source tools.

4.2.3 Server infrastructure

To do...

4.3 Technical Implementation

To do...

Chapter 5

Results of RLA Pilot

For six weeks in October and November of 2017, the approximately 80 learners participated in an online course using a git-based learning platform as the first phase of the RLA program. Following the course, an optional survey was conducted, the questions of which are included in the appendix. In this section we report the experience, results, and feedback we observed in the course and survey responses.

5.1 User Experience

To do...

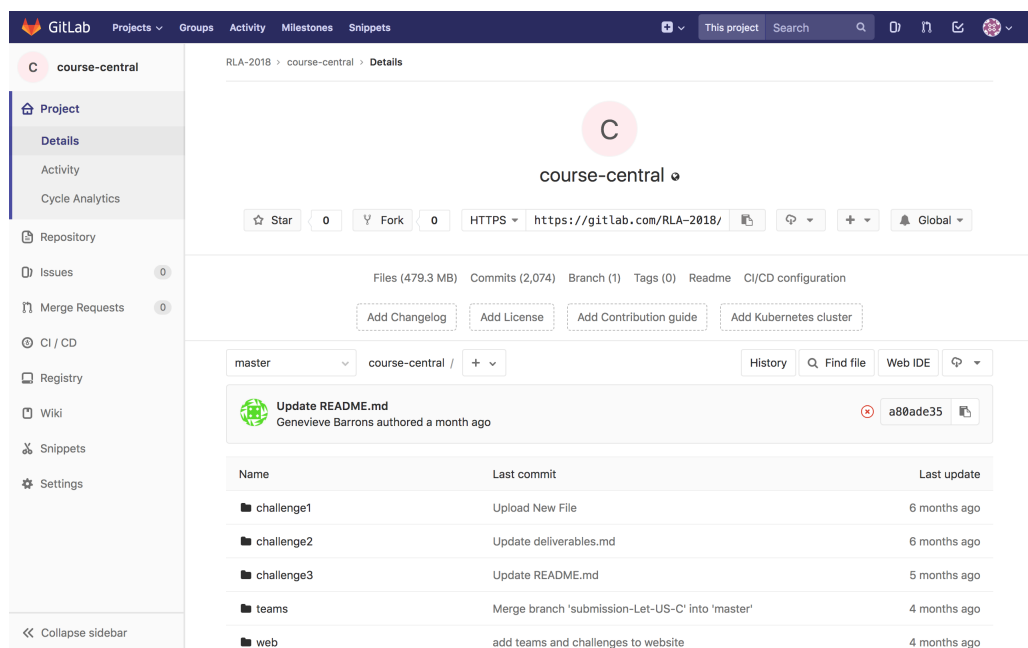


Figure 5-1: The `course-central` repository.

Chapter 6

Conclusion

To do...

Appendix A

RLA Challenge Descriptions

The online course was divided into three challenges, each of which had two sets of deliverables (one for each week). Here we provide brief descriptions of these. More information as well as various resources related to the topics of each challenge is available online. [11] All content was developed by members of the MIT Media Lab Learning Initiative unless otherwise noted.

1. **Challenge 1 - chatbots:** How might we use chatbots to support refugee learners inside or outside of the classroom?
 - Describe your concept for a chatbot in one or two paragraphs. Use visuals (mockups, diagrams) if possible. It is important to be as specific as possible while mapping out the scope of the project as this will greatly influence implementation and the tools to use. Answer the following questions:
 - What problem/ challenge will the chatbot solve?
 - How will the chatbot solve it?
 - Who is the primary user and how will the chatbot engage the user?
 - What activity does the chatbot facilitate that would not otherwise be possible?
 - What challenges do you expect to encounter?
 - Demonstrate how your chatbot works. Videos, screenshots or images are encouraged. Add a short description and answer the following questions:

- How did you build it? (Platform and technology)
- What challenges did you face?
- What aspect of the chatbot do you like best?
- What would you different in the future?
- Please also add a link to your code!

2. **Challenge 2 - human centered design:** How might we use everyday technologies as learning tools?

- Make an initial prototype. Questions to answer:
 - What solution are you testing? (and why did you choose it?)
 - Submit your prototype (use photos, video, diagrams etc.)
 - Describe the prototype and why you chose this prototyping method.
 - What did you learn during the prototyping process?
 - Who are your intended users for testing?
- Test it with users. Questions to answer:
 - How did you select your test users?
 - What was the setting of the test?
 - What were the main points of feedback you received (share a summary)?
 - What changes would make to your idea/project based on the feedback?
 - What parts of the Human Centered Design process were new to you?
 - What parts of the Human Centered Design process seemed most useful to you?

3. **Challenge 3 - augmented reality (AR):** How might we use Augmented Reality (AR) to support refugee language learning outside the classroom?

- Describe your concept for how low fidelity Augmented Reality could be used to support language learning in one or two paragraphs. Use visuals

(mockups, diagrams) if possible. It is important to be as specific as possible while mapping out the scope of the project as this will greatly influence implementation and the tools to use. Answer the following questions:

- What problem/challenge will the AR experience solve?
 - How will the AR experience solve it?
 - Who is the primary user and how will the AR experience engage the user?
 - What hardware does the user need? Is this realistic in the refugee context?
 - What activity does the AR experience facilitate that would not otherwise be possible?
 - What challenges do you expect to encounter?
- Demonstrate how your AR experience works. Videos, screenshots or images are encouraged. Add a short description and answer the following questions:
 - How did you build it? (Platform and technology)
 - What challenges did you face?
 - What aspect of the AR experience do you like best?
 - What would you do differently in the future?
 - Please also add a link to your code!

Appendix B

RLA Online Course Survey

The following questions were sent in an email to participants to gather feedback and evaluate how well the platform worked. The survey was anonymous and asked for consent for responses to be used in this research study. A total of 11 participants chose to respond.

1. Git

- How much had you used git before the course?
- To what extent did you learn git during the course?
- How useful would you say knowledge of git was for collaborating with your team?

2. Course materials

- Did you have any difficulty accessing course materials? If so, what were they?
- Did you find the course materials helpful in working on your projects? If so, which materials were most helpful?

3. Signing up

- Did everyone on your team submit their own initial sign up on GitLab?

- How many people on your team used GitLab after the initial sign up? Was there a reason some did not?
- Did you find it useful to have a list of participants in the course?

4. Submission

- Would you have preferred an automatic, continuous submission (as opposed to your work being snapshot at a set time)?
- Did you run into any issues in submitting your work? If so, what were they?
- Were you able to see other team's submitted work? If so, did you find that useful?

5. Seminars

- What were your favorite and least favorite features of the Unhangout platform for seminars?
- Did you find the seminars useful in ideating for your projects? If so, were any seminars particularly useful?
- Did you find the seminars useful for the technical side of your projects? If so, were any particularly useful?

6. Team

- How was the technical work distributed among your team?
- Was your team physically located in the same place?

Bibliography

- [1] About Fork The Cookbook. About Fork The Cookbook. <http://forkthecookbook.com/about>. Accessed: 19-May-2018.
- [2] Bates, A. W. (2015). *Teaching in a Digital Age: Guidelines for Designing Teaching and Learning*. Tony Bates Associates, Vancouver.
- [3] Brennan, K. and Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 annual meeting of the American Educational Research Association*.
- [4] Chacon, S. Getting Started - A Short History of Git. <https://git-scm.com/book/en/v2/Getting-Started-A-Short-History-of-Git>. Accessed: 19-May-2018.
- [5] Cherry, K. What Is Self-Determination Theory? <https://www.verywellmind.com/what-is-self-determination-theory-2795387>. Accessed: 19-May-2018.
- [6] David, L. Self-Determination Theory. <https://www.learning-theories.com/self-determination-theory-deci-and-ryan.html>. Accessed: 19-May-2018.
- [7] DevArt: Art Made with Code. DevArt: Art Made with Code. <https://devart.withgoogle.com/>. Accessed: 19-May-2018.
- [8] Feliciano, J. (2015). Towards a Collaborative Learning Platform: The Use of GitHub in Computer Science and Software Engineering Courses. Master's thesis, Department of Computer Science, University of Victoria. An optional note.
- [9] Github Classroom. Github Classroom. <https://classroom.github.com/>. Accessed: 19-May-2018.
- [10] Haaranen, L. and Lehtinen, T. (2015). Teaching git on the side: Version control system as a course platform. In *Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education*, pages 87–92. ACM.
- [11] Initiative, L. Refugee Learning Accelerator. <https://refugeelearning.media.mit.edu/>. Accessed: 19-May-2018.
- [12] Kelleher, J. (2014). Employing git in the classroom. In *Computer Applications and Information Systems (WCCAIS), 2014 World Congress on*, pages 1–4. IEEE.

- [13] Mozilla’s Global Sprint. Mozilla’s Global Sprint. <https://mozilla.github.io/global-sprint/>. Accessed: 19-May-2018.
- [14] Niemiec, C. P. and Ryan, R. M. (2009). Autonomy, competence, and relatedness in the classroom: Applying self-determination theory to educational practice. *School Field*, 7(2):133–144.
- [15] P. Schmidt, M. R. and Rusk, N. Learning Creative Learning: How we tinkered with MOOCs. <http://reports.p2pu.org/learning-creative-learning/>. Accessed: 19-May-2018.
- [16] Resnick, M. (2014). Give P’s a chance: Projects, peers, passion, play. In *Constructionism and creativity: Proceedings of the Third International Constructionism Conference*. Austrian Computer Society.
- [17] Resnick, M. (2017). *Cultivating Creativity through Projects, Passion, Peers, and Play*. The MIT Press, Cambridge.
- [18] Rheingold, H. Mitch Resnick: The Role of Making, Tinkering, Remixing in Next-Generation Learning. <https://dmlcentral.net/mitch-resnick-the-role-of-making-tinkering-remixing-in-next-generation-learning/>. Accessed: 19-May-2018.
- [19] Rocco, D. and Lloyd, W. (2011). Distributed version control in the classroom.
- [20] ScP. Schmidt, M. R. and Ito, J. Creative Learning and the Future of Work. *Disrupting*, page 147.
- [21] Zagalsky, A. Why you should use GitHub: Lessons for the classroom and newsroom. <http://www.storybench.org/use-github-lessons-classroom-newsroom/>. Accessed: 19-May-2018.
- [22] Zagalsky, A., Feliciano, J., Storey, M.-A., Zhao, Y., and Wang, W. (2015). The emergence of github as a collaborative platform for education. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*, pages 1906–1917. ACM.