PyTut: Examining the Effectiveness of E-Learning

Ross Kohler
University of Cape Town
khlros003@myuct.ac.za

Carla Kirk-Cohen
University of Cape Town
krkcar003@myuct.ac.za

CCS Concepts

•Applied computing → Interactive learning environments; E-learning; Interactive learning environments; E-learning;

Keywords

Python; Web-Based; E-Learning

1. PROJECT DESCRIPTION

Computer Science education is facing a retention crisis [10]. Large numbers of students enroll in programming courses, but very few successfully complete them. Cognitive overload has been identified as one of the difficulties that students face when first attempting to learn to program [2]. This occurs due to difficulty adapting to an algorithmic way of thinking and the challenge of expressing algorithms in programming languages contribute to high failure rates [2]. It has also been found that complex IDE installation and features can be a frustrating barrier to learning [5].

The University of Cape Town attempts to address this issue by streaming first year Computer Science students based on a decant test written after the first term of instruction. Students who perform adequately continue on to complete the CS1 course in one year. Under performing students are moved into an extended degree program, CSC1010H, which covers work at a slower pace and provides additional academic support [15]. To aid these students in learning to program, this project proposes the development of a web-based e-learning prototype to supplement traditional classroom learning in the extended degree program. The prototype will focus on Python programming, as Python is the language of choice in CSC1010H. The prototype will provide interactive code examples, automated assessment and interactive feedback, offering students an interactive environment that facilitates learning at their own pace. The effectiveness of an e-learning prototype as an addition to classroom instruction and the effectiveness of each component will be assessed.

© 2016 ACM. ISBN 978-1-4503-2138-9.
DOI: 10.1145/1235

1.1 Project Significance

Programming is a vital skill that is constantly in demand in the job market and across academic disciplines. If high failure rates cannot be addressed, shortages of skilled professionals in the Computer Science field will only get worse. Learning to program requires time intensive practice that cannot feasibly be completed solely in a lecture or tutorial setting [9]. All students must practice programming in their own time if they wish to succeed. This project will provide learners with an interactive environment aimed at aiding them in self-study programming exercises. Online tools to aid students in learning to program already exist. These could be considered for inclusion as a supplement to the CS1010H class but the effectiveness of e-learning has been found to be conditional on the e-learning initiatives being well-focused and relatively small scale [13]. The systems already available operate on a large scale and are not specifically tailored to the CS1010H class. The prototype proposed for this project is more appropriate because it will closely follow the CS1010H syllabus and will provide students with feedback from their lecturers and tutors. To examine the effectiveness of an e-learning system, it is important that the system being used is the most suitable for the task at hand. For this reason, a prototype will be developed instead of using an already existing system.

1.2 Project Issues

A literature review of 96 research reports, with data cumulatively gathered from 19331 subjects, found that e-learning initiatives were 6% more effective than traditional classroom instruction [18]. Other studies have found that there is no significant difference between classroom instruction and e-learning's effectiveness [20]. The effectiveness of e-learning has not been definitively proven. The effectiveness of the prototype developed for this project is an important issue; the introduction of an unhelpful e-learning prototype will not help to address the problem, it will exacerbate it.

It has also been observed that students that are unfamiliar with or anxious about the use of computers perform worse off in e-learning initiatives than students that are comfortable with computers [12]. It is important to be mindful of this issue so that less computer confident students are not disadvantaged.

2. PROBLEM STATEMENT

2.1 Aims of work

This project aims to determine how effective the introduction of an e-learning prototype as a supplement to classroom instruction is. Evidence collected suggests that the combination of e-learning and traditional classroom instruction can be highly effective, but this point has not been explicitly proven [18]. The prototype developed will provide interactive coding examples, an automated assessment mechanism and interactive feedback from tutors or lecturers. This project will seek to isolate the effectiveness of these individual components to determine how much each contributes to the learning process. This is important for the development of future systems, as components that do not aid in learning should not be included.

2.2 Research questions

In keeping with the aims of this project, the following research questions will be addressed:

1. Does the an e-learning prototype as a supplement to classroom instruction significantly aid beginner programmers?

2. How effective are interactive code, automated assessment and interactive feedback in aiding beginner programmers?

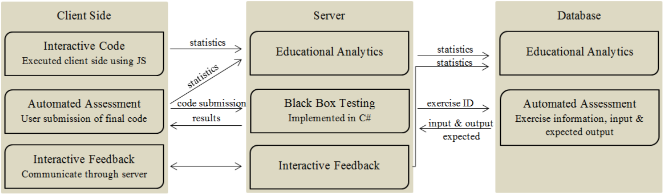
3. PROCEDURES AND METHODS

The effectiveness of e-learning systems and e-learning components will be tested through the development of an e-learning prototype and subsequent user surveys. This prototype will have the goal of reducing the cognitive load and barriers to learning that beginner programmers face. This will be achieved through the use of editable interactive code snippets, an automated assessment system and interactive instructor feedback. In addition, information about how the prototype is used will be stored in a back end database to provide educational analytics.

3.1 Design Features

The e-learning prototype developed will have a typical client-server architecture, as illustrated in figure 1. The client side will handle interactive code execution to prevent server-side congestion. This is important for beginner programmers because programs often need to be executed multiple times to detect simple syntactic and semantic errors. The server will handle automated assessment, database interaction to store usage information and facilitating interactive communication.

Figure 1: Prototype Architecture



3.1.1 High-level Overview

The user begins by logging into our e-learning platform on any web browser. He is presented with a list of assignments or exercises he is required to complete within a certain module of study. Opening an exercise he is provided with the information required to complete the problem as well as a text box within which he can code the solution. While he is writing his solution the platform provides him real time feedback on what is incorrect or what can be improved upon. The user can then submit the code for assessment. The platform will respond back with the detailed results of the assessment and his score for the exercise. The user will then have the option to contact one of the course tutors for questions or help in solving the problem. Upon successful completion of the exercise, subsequent exercises will be unlocked.

3.1.2 Interactive Code

Simple, web-based execution of code snippets has been found to be a much more beginner-friendly way of introducing a student to programming. In the context of web-based systems, interactive code comes in the form of a notepad style frame, which may contain code pre-loaded by instructors, that students can execute and edit. JavaScript will be used to execute the students' Python code in browser. In browser JavaScript execution is more suitable than traditional IDEs because web-browsers automatically download the necessary resources to execute code [19]. In browser execution will be achieved through the use of Skulpt, an open source in browser implementation of Python programmed in JavaScript [21]. One of the greatest challenges faced in interactive code execution will be the large variety of libraries available in Python; Skulpt must include these libraries for students to use them. This challenge can be addressed by determining the set of libraries that students are most likely to use and ensuring that they are implemented in Skulpt, an acceptably minor task as this is a beginners course. As a stretch goal, this project could extend the existing Skulpt implementation to include a library that students are likely to use.

3.1.3 Automated Assessment

Automated assessment allows students to receive instant feedback on their work which can be helpful when tackling problems without a teacher present. This is particularly useful for beginner programmers who often make trivial mistakes which are easy to fix when provided with basic guidance [6]. In introductory programming courses, the use of an automated assessment system has the potential to free lecturers and tutors to answer more complex issues instead of wasting time on fixing trivial errors. It is, however, important to place limits on use of automated assessment systems to prevent students from adopting a trail-and-error, brute force approach to programming [16]. Automated Assessment will be dealt with on the server side. Once students are satisfied with their work, they can choose to submit it for marking. Note that this is different to merely running the program, as marking submissions will be limited. A limited number of submissions will help to prevent server side congestion. Black box testing will be used to mark students' work. This is a method of testing that compares the output of the submitted code to the output of a model solution for a given set of input [7]. The database will store input and expected output for each exercise. Input tested will be randomized to prevent students from trying to cheat the system by hard coding output.

One of the challenges faced is that black box testing is limited to checking program correctness, as tests cannot assess students programming style [4]. This can be specifically problematic in beginners' programs where students may produce the right output but do so in a clumsy way that does not incorporate the components the assignment was aimed to test (using multiple nested if statements instead of a switch statement, for example). To address this challenge, basic analytic information will be included with the submission (number of lines, types of loops used and other stylistic features) for lecturers to review if desirable. Another drawback of black box testing is that creating sufficient tests to check for correctness is time consuming for lecturers. This challenge must be balanced against the time taken to individually mark students work. If specifying tests is found to be overly time consuming, further work must be done to address this issue.

3.1.4 Interactive Feedback

Personal interaction in lectures and tutorials has been proven to be more effective than textbooks and example programs [3]. This is problematic because introductory Computer Science classes are often very large and must cover a substantial syllabus in a limited amount of time; lecturers often have to discourage questions in class, or limit the time they spend answering them for the sake of covering the material allocated to the time. Lecturers play a vital role in the use of e-learning in courses [11]. It has been found that web based learning is 19% more effective when students receive feedback throughout the course [18].

The option to ask a lecturer or tutor for help will be provided through the introduction of an online chat feature run through the server. For convenience, the code that students are struggling with will automatically be sent to the instructor. In practise, it is not feasible to have lecturers and tutors available at all times. This challenge must be addressed by determining designated times for interactive feedback.

3.1.5 Data Storage

All data will be stored server side in a Microsoft SQL database. Tables will be created to store user information, user statistics, unit tests for automatic marking as well as feedback information. All transactions and table management will be handled by the ASP.NET framework. This storage is persistent and will be vital for the final analysis of how students performed on the platform.

3.2 Development Platform

The e-learning prototype developed for this project will be implemented on the web. The ASP.NET framework will be used to develop the website that the prototype will run on. A Microsoft SQL database will be used to store all of the relevant data required for out research. The web app will be hosted on a server provided by the University of Cape Town. This integration will help alleviate some of the pains of normal web development.

3.3 Prototype Evaluation

The e-learning prototype will be evaluated through the use of user surveys. Half of the class will use the e-learning prototype during practical sessions and at home in their own time and the remaining half will continue as before. A general survey with questions about the learning environment will be given to the whole class. No reference to the e-learning prototype will be made so that results are not skewed towards our prototype. The survey will use a 1-5 likert scale to determine how useful the environment is. A second survey will be given to students using the e-learning prototype. This survey will use a 1-5 likert scale to evaluate the usefulness of individual components of the prototype. It will also have open ended questions about what students like and disliked about the prototype. Students will also be asked to rank the features in order of how useful they found them. The educational analytics gathered during testing will be used wherever possible to supplement the results obtained in the surveys.

4. ETHICAL,PROFESSIONAL AND LEGAL ISSUES

Since our project is primarily research based we'll be consulting the ethics of research specific to the computer science context. The Ethics in Research guide [14] discusses a number of points that need to be remembered while the research is being conducted. In our research we'll be focusing on the evaluation of how various e-learning techniques can be used to improve the understanding of basic programming concepts that are being taught in the 2016 CSC1010H course.

Our experiment requires that only half of the class uses the e-learning prototype so that we have a baseline set of statistics for learning without it. Depending on the results of our experiment, different ethical issues arise. Should the prototype be effective, half of the class is being deprived of a potentially advantageous tool. Alternately, asking half of the class to use a tool which does not aid their learning also raises an ethical problem. These concerns can be mitigated due to the short amount of time that our experiment will run for. Students will only experience the advantages or disadvantages of our prototype for a very short time. For the sake of fairness, the class will be randomly split. Usage statistics will be gathered from students using the e-learning prototype. Some ethical questions have risen on student privacy and what we are allowed to do with the information that we gather. We also need to ensure that experiments are conducted ethically and that results are not biased towards certain methods.

Some Unethical conduct listed in the Ethics in Research Guide [14] are as follows:

- Interference with other research.
- Taking data without permission.
- Infringing on the confidentiality or privacy of others.
- Recording observation in secret without the knowledge of the subject.
- The use of unsubstantiated or unvalidated data.
- Destroying or not publishing data that does not fit the desired result.

When conducting our experiments it is important for us to find out whether we'll be interfering with other research, this includes the two other research groups who will be running their experiments at the same time as ours. This can be determined by speaking to the course convener, Gary Stewart. Getting permission from him and students to conduct the experiments are also ethical requirements. Any data that we retrieve from students must not be distributed so as to not infringe on their privacy, research results must be published with anonymity of the subjects (unless permission has been provided). When performing the experiments it would be unethical for the researchers to bias the experiment by favouring one of the learning techniques. We as researchers need to keep an open mind during the process and remember the purpose of the project is to help students.

Our final report can't violate the following ethical guidelines:

- It can't steal ideas on plagiarise on existing research.
- It can't conceal problems or distort results.
- It can't be presented in an overly complex manner.

The UCT Science department requires all research to to be approved through the completion and submission of the "Faculty of Science Researcher Ethics Statement Form 2016". Upon approval we'll be able to follow through with our research.

5. RELATED WORK

5.1 Online Tutors

5.1.1 PiLet

PiLet is a unique web-based tool for teaching Python programming because it was designed to cater for students of all learning styles. The tool provides executable examples of code, videos explaining concepts, programming puzzles in the form of code blocks that must be rearranged and assessment in the form of randomized multiple choice questions [1]. In future work, developers hope to adapt the system to use analytics of student use to personalize their learning experience to fit their learning style [1].

5.1.2 CS Circles

CS Circles is a free programming website that aims to teach introductory programming skills an an accessible way [17]. Lessons contain embedded executable code blocks that students are encouraged to interact with. When necessary, students can send messages with their code automatically included to their lecturer. These help messages, along with all code students have produced, are stored as part of the student's progress report, which can be accessed by lecturers at any time. CS Circles does not have an anti-plagiarism check so lecturer access to student records must be relied on to determine authenticity. An automated assessment system that runs on a central server is used to check the correctness of student submissions. Code can be debugged visually and pop up hints are available to help struggling students [17].

6. ANTICIPATED OUTCOMES

6.1 Prototype

Students are expected to engage in the e-learning prototype. The platform designed should provide students with an effective and helpful learning environment that lowers barriers to programming. The prototype will be developed using a build, measure and learn cycle. Initially, a prototype will be developed for demonstration and testing purposes. This prototype will specifically focus on the content that the CSC1010H class will be covering at the time of testing. The results of this project, regarding the effectiveness of the prototype as a whole and individual elements, will be used to determine the form that future iterations take.

6.2 Expected Impact

A review of available literature indicated that the use of a web based e-learning system as a supplement to classroom instruction can improve the learning experience for users. It has, however, been found that student engagement depends of the degree to which the system interactively engages them [8]. The prototype developed is expected to provide users with an interactive learning environment that enhances their learning experience and therefore contributes to success in the CS1010H course. This expectation will be verified using student feedback on the prototype. As interactive elements tend to be more effective, it is expected that the interactive code snippets will be most useful to students. The effectiveness of the prototype as a whole and the individual components will aid in determining how to create a meaningfully helpful e-learning system.

6.3 Success Factors

The following factors will be used to evaluate the success of this project:

- The e-learning prototype reduces barriers to programming
- Overall, students find working within the e-learning prototype to be a valuable addition to learning
- Students use the tools provided (determined from analytics gathered)
- Interactive code, automated assessment and interactive feedback positively contribute to learning

Evaluating the use of the e-learning prototype and subsequent student results would be an appealing way to test success of the project. Tests of this nature are, unfortunately, infeasible due to the short amount of time available for testing. This project will be a success if it provides meaningful insight into the use of e-learning systems as a supplement to classroom instruction. The prototype developed for this project will be considered to be a successful model for future e-learning systems if students experience it as a more effective way of learning than traditional methods.

7. PROJECT PLAN

7.1 Risks

The following are some risks that have the potential of slowing research progress:-

- Software that we have planned to use does not support the function or feature that the prototype requires.
- Integration of the front-end and the back-end of the prototype fails and the platform does not achieve its function.
- Poor planning leads to us missing deadlines and important features have to be removed from the platform.

- Our ethics form gets rejected by the faculty and we are unable to use the students of CSC1010H as a sample group for our research.

In order for us to ensure this project runs smoothly we need to make plans to mitigate these risks. To ensure our software stack supports all functions and features for our prototype we need to do an in depth design of the platform before attempting to build it. In this design phase the inner workings of the prototype should be explored in detail. Integration of the two pieces of the web platform is a crucial part of our project. This is why it should not be left to the last minute. To mitigate the risk of the integration it's important that during the period of the project that there are multiple integrations that produce a functional product. Any problems will arise at these points and they can be solved way before the due date of the final project. Poor planning is the cause of most failed projects. Regular meetings with our supervisor as well as the team members will ensure that everyone understands where the project is currently and what needs to be completed next. The team should follow the project timeline as closely as possible so they can meet all of the milestones. Our ethics form should be accepted as long as we follow all of the ethical guidelines that are stated in the Ethics in Research Guide [14]. This document should be read thoroughly before submitting the form to the faculty for approval.

7.2 Timeline

Please see Figure 2 for the projects timeline illustrated in the form of a Gantt chart.

7.3 Required Resources

The main resources that this project requires is in the form of software. This should all be available for free online or will be created by us for the purpose of research. The web platform will either be hosted on a computer within the UCT network or a free hosting solution will be found online. The last resource required will be the students of the CS1010H course who will be split into groups to evaluate our various e-learning techniques.

7.4 Deliverables

Date	Deliverable
Tue 17-May	Project Proposal due, including project plan
Wed 8-Jun	Revised Proposal Finalized and uploaded to Vula
Fri 22-Jul	Background/Theory section (based on Lit Survey) done
Mon 29-Aug	Paper plan/ scaffold completed
Tue 20-Sep	First Implementation/Experiment/Performance Test + Writeup
Thu 29-Sep	Final Prototype/Experiment/Performance Test + Writeup
Tue 4-Oct	sections on Implementation and Testing
Tue 11-Oct	Outline of complete paper: section and major section headings
Tue 18-Oct	Final Complete Draft of paper
Fri 28-Oct	Project Paper Final Submission
Mon 31-Oct	Project Code Final Submission
Mon 07-Nov	Poster Due
Fri 11-Nov	Web Page
Mon 14-Nov	Reflection Paper

7.5 Milestones

7.5.1 *Prototype Iterations*

The platform will be developed with the Agile methodology in mind. The e-learning tools of the prototype will be designed and built independently, connected to make a complete solution and then tested to find the bugs and features that should be implemented in the next iteration. Throughout the lifespan of the project the platform is scheduled to go through three iterations. The first iteration will implement the minimum function required for the platform to feasibly achieve its purpose. The second iteration will add some more functions while making changes and fixes to what was done in the first iteration. The final iteration will have all of the required features working as well as most of the bugs from the previous two iterations fixed. Each iteration will be given 4 weeks to be built and tested. The first will be due on 11 July, the second on 15 August and the third on 19 September.

7.5.2 *Prototype Demonstrations*

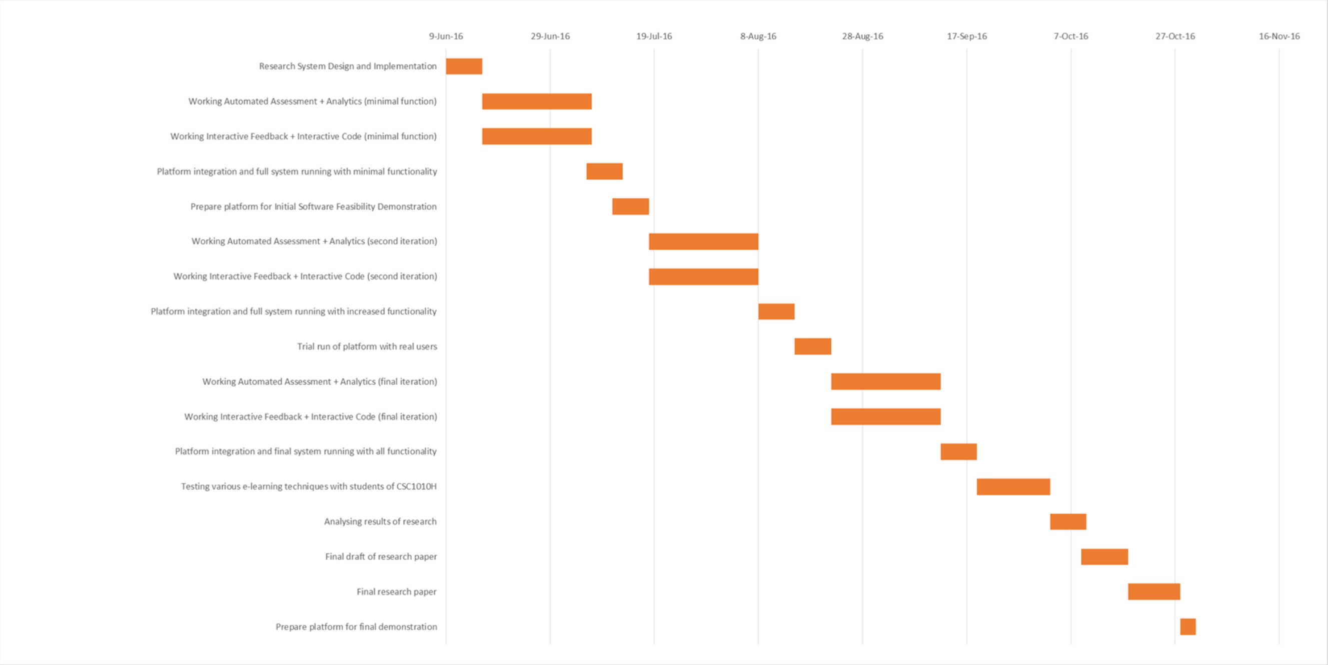
The first demonstration is for initial software feasibility and will essentially be a demonstration of our first iteration with the minimum viable function. The final demonstration will show the final platform that we conduct the tutoring with. Both demonstration will be in front of our class group, supervisor as well as second reader. The first demonstration is scheduled for the period between 18-22 July and the final demonstration is schedule between 31-October to 4-November.

7.5.3 *Report Submissions*

A complete draft of the research report is expected by 18-October. The final project papers is due 28-October. There is also a due date for all of the code to be submitted which is set at 31-October.

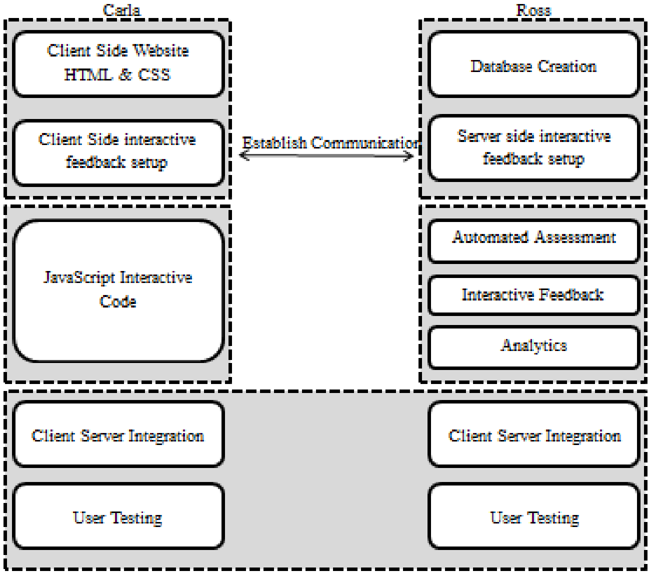
7.6 Work Allocations

Figure 2: Gantt Chart



In the development of the web platform Carla will be tasked with writing the interactive code and Interactive feedback components. Ross will be tasked with writing the Automated Assessment components as well as the setup of the database and the code that will monitor the student's performance and generate actionable analytics that we can use for our study. Using the students of CSC1010H we'll both be conducting experiments to see how various e-learning techniques affect their performance in the computer science context. The allocation of work is visualized in figure 3.

Figure 3: Work Allocations


Carla
Ross
Client Side Website HTML & CSS
Database Creation
Client Side interactive feedback setup
Establish Communication
Server side interactive feedback setup
JavaScript Interactive Code
Automated Assessment
Interactive Feedback
Analytics
Client Server Integration
Client Server Integration
User Testing
User Testing

8. REFERENCES

[1] B. Alshaigy, S. Kamal, F. Mitchell, C. Martin, and A. Aldea. Pilet: an interactive learning tool to teach python. In *Proceedings of the Workshop in Primary and Secondary Computing Education*, pages 76–79. ACM, London, UK, 2015.

[2] C. Alvarado, B. Morrison, B. Ericson, M. Guzdial, B. Miller, and D. L. Ranum. Performance and use evaluation of an electronic book for introductory python programming. 2012.

[3] D. A. Cook and D. M. Dupras. A practical guide to developing effective web-based learning. *Journal of General Internal Medicine*, 19(6):698–707, 2004.

[4] C. Douce, D. Livingstone, and J. Orwell. Automatic test-based assessment of programming: A review. *J. Educ. Resour. Comput.*, 5(3), Sept. 2005.

[5] S. H. Edwards, D. S. Tilden, and A. Allevato. Pythy: Improving the introductory python programming experience. In *Proceedings of the 45th ACM Technical Symposium on Computer Science Education*, SIGCSE '14, pages 641–646, New York, NY, USA, 2014. ACM, Blacksburg USA.

[6] A. J. Hartz. *CAT-SOOP: A tool for automatic collection and assessment of homework exercises*. PhD thesis, Massachusetts Institute of Technology, 2012.

[7] M. Joy, N. Griffiths, and R. Boyatt. The boss online submission and assessment system. *J. Educ. Resour. Comput.*, 5(3), Sept. 2005.

[8] S. Kalyuga. Enhancing instructional efficiency of interactive e-learning environments: A cognitive load perspective. *Educational Psychology Review*, 19(3):387–399, 2007.

[9] C. Kelleher and R. Pausch. Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers.

ACM Computing Surveys (CSUR), 37(2):83–137, 2005.

[10] E. Lahtinen, K. Ala-Mutka, and H. M. Järvinen. A study of the difficulties of novice programmers. In *ACM SIGCSE Bulletin*, volume 37, pages 14–18. ACM, 2005.

[11] S. Liu. Faculty use of technologies in online courses. 2005.

[12] J. J. Martocchio. Effects of conceptions of ability on anxiety, self-efficacy, and learning in training. *Journal of applied psychology*, 79(6):819, 1994.

[13] R. Michaelson. Does e-learning work? 2002.

[14] U. of Cape Town Computer Science Department. Research methods, 2016.

[15] U. of Cape Town Science Faculty. Extended degree programme, 2016.

[16] A. Pears, S. Seidman, L. Malmi, L. Mannila, E. Adams, J. Bennedsen, M. Devlin, and J. Paterson. A survey of literature on the teaching of introductory programming. In *Working Group Reports on ITiCSE on Innovation and Technology in Computer Science Education*, ITiCSE-WGR '07, pages 204–223, New York, NY, USA, 2007. ACM, New York, NY, USA.

[17] D. Pritchard and T. Vasiga. Cs circles: An in-browser python course for beginners. In *Proceeding of the 44th ACM Technical Symposium on Computer Science Education*, SIGCSE '13, pages 591–596, New York, NY, USA, 2013. ACM, New York, NY, USA.

[18] T. Sitzmann, K. Kraiger, D. Stewart, and R. Wisher. The comparative effectiveness of web-based and classroom instruction: A meta-analysis. *Personnel psychology*, 59(3):623–664, 2006.

[19] L. Tang. A browser-based program execution visualizer for learning interactive programming in python, 2015.

[20] E. T. Welsh, C. R. Wanberg, K. G. Brown, and M. J. Simmering. E-learning: emerging uses, empirical results and future directions. *International Journal of Training and Development*, 7(4):245–258, 2003.

[21] www.skulpt.com.