UNIVERSITY OF CAPE TOWN

DEPARTMENT OF COMPUTER SCIENCE

COMPUTER SCIENCE HONOURS
FINAL PAPER
2016

Title: Pytut: The Implementation of an Online Python Tutor as a Supplement to Introductory Programming Courses.

Author: Ross Kohler

Project Abbreviation: PYTUT

Supervisor(s):Gary Stewart

Category	**Min**	**Max**	**Chosen**
Requirement Analysis and Design	0	20	
Theoretical Analysis	0	25	
Experiment Design and Execution	0	20	
System Development and Implementation	0	15	
Results, Findings and Conclusion	10	20	
Aim Formulation and Background Work	10	15	
Quality of Paper Writing and Presentation	10		
Quality of Deliverables	10		
Overall General Project Evaluation (*this section allowed only with motivation letter from supervisor*)	0	10	
Total marks		**80**	

ABSTRACT

Computer science and programming can often be a difficult subject for new students to grasp. A lecture-style learning environment may not always cater for students struggling with understanding the fundamentals. The prescribed learning material may also be overwhelming and place an unnecessarily high cognitive load of the student. The project aims at implementing and evaluating a web based online platform, Pytut, that can facilitate the learning of Python aimed specifically at the students of the CSC1010H extended program. This platform implemented various tools aimed at helping the student in completing simple programming exercises. The effectiveness of Pytut was evaluated by requesting the students to engage with the platform. The results of this study was drawn from user surveys and usage analytics.

1. INTRODUCTION

Learning how to program is often a daunting task for novice computer scientists. The discipline often requires deep understanding of abstract concepts. It is widely considered one of the more difficult subjects for new students to grasp. In addition to this, because of the advanced nature of the subject, learning resources are often scarce and personal instruction may not be readily available. In a classroom setting groups can be large and instruction that benefits all students are difficult design from a teachers point of view. The result is that there is often a high drop out rate in beginner programming courses [7].

Learning to program is often compared to learning Mathematics. Students generally cannot solely be taught the subject in a lecturing environment, they need to practice and grapple with the problems in their free time. For every programming language there is often a large amount of documentation that one can review to become more acquainted with the syntax and structure of the language. For a novice programmer this documentation can often become overwhelming, difficult to understand and place a high cognitive load on the student. This results in the students becoming disheartened, despondent and having a weak grasp on the material. This calls for a medium through which students can interact with and master the fundamentals of a language. A medium that is responsive and adapts to a students understanding and progresses with the subject at a level that each individual student can be comfortable with.

Over recent years web technologies have grown to become increasingly more popular in managing and delivering content for the purpose of education. Web Browsers have developed to such a point that they are capable of running complex code to facilitate the function of interfaces that used to only be possible in standalone desktop applications. Modern web applications can be run by most browsers without any setup time or requirements on the client-side. These applications can perform complex tasks, are responsive to user interactions and are ideal for delivering static and dynamic content. The capabilities of web applications make them ideal for facilitating an interactive learning environment through which beginners programmers can learn the fundamentals of programming.

This project aims to determine how effective instruction from an e-learning platform can be as a supplement to lecture-style teaching. There is research that suggests that e-learning platforms can be effective if the students have already had some previous instruction in the subject. The platform then just acts as a means to build up that knowledge and improve understanding [11]. This project proposes the implementation of a web application, named Pytut, that provides beginner programmers a means through which they can test their skills and improve their understanding of fundamental programming concepts. Various tools will be implemented for the purpose of gauging their value in aiding the student in improving their grasp of the material.

The scope of Pytut will shape around the curriculum of UCT's extended introduction to programming course, namely CSC1010H. Students who are struggling with the mainstream introduction course are placed in the extended degree program. This program covers the same content at a slower pace and places an greater emphasis on academic support and tutoring for the the students. These students are learning Python as their introductory language and the syllabus covers the majority of introductory concepts such as input/output, conditional statements, loops ,simple data structures etc. The effectiveness of Pytut will evaluated by providing it to the class of CSCS1010H as an additional means through which they can revise their course content in preparation for the end of year exams. In two of their prescribed lab sessions we would ask them to engage with Pytut and then evaluate its effectiveness through the means of user surveys and usage analytics.

2. RESEARCH QUESTION

The research question that this paper hopes to answer is the as follows; Does the introduction of an e-learning system as a supplement to classroom instruction significantly aid beginner programmers and does a responsive and interactive platform provide value to the student's learning ?As opposed to similar e-learning platforms Pytut hopes to evaluate whether an online platform can introduce additional tools to effectively aid the learning of a new programming language, in this case Python. Research was conducted in order to better understand which segments and tools of e-learning platforms have proven to be the most effective.

E-learning and web technology provide a more interactive medium through which learning material can be displayed and manipulated. Traditional teaching methodology makes use of a very one-sided approach where the flow of information is typically uni-directional i.e. originating from the lecturer or textbook and being delivered to the student. A web platform provides the ability to gauge how well the students understand the concepts, provide a means for a student to engage and interact with the material and also provide feedback on their progression through the syllabus to both the student as well as the instructor.

3. BACKGROUND RESEARCH AND RELATED WORK

3.1 Favourable Traits for a Teaching Language

Factors that make up the ideal programming language for teaching are ones that prioritize the human element of coding. Such factors could include how easy it is to grasp the fundamentals of the language (at least for teenagers), how practical it is to use the language to solve real world problems, its ability to encourage problem solving and not obstruct the process as well as the language's versatility in

the context of problems it can solve [3]. There are conditions that are widely considered necessary for learning a new computer language, they are as follows:

- A simple notational machine where the meaning of constructs can be easily understood.
- An interactive system that allows the users to query and inspect the process.
- All commentary found in the language (error messages, API documentation etc.) should be written at a level that can be understood by novices.
- Some of the processes of the environment should be visible to the user. [3]

The language should also ideally be scripted as this allows for the user to construct a rapid solution to a problem when the need arises. Execution time of a scripting language is roughly double that of a compiled language however this is counter balanced by the fact that a solution is formulated in half the time [3].

3.2 Benefits of Teaching Python

Python is a clean and intuitive scripting language. The aspects that make the language favourable for novices:

- Small and intuitive syntax compared to other languages such as Java, C++ etc.
- Python is dynamically typed and thus reduces the syntax required for working code even further.
- Python's basic data types are extremely powerful and can be easily manipulated for the context of the application.
- The language enforces proper programming practice through indentation and structure.
- Python is an opensource language widely used in many applications this has resulted in a large resource pool in the form of libraries, exercises, documentation etc. that users can draw from [5] .

The most commonly used programming model in the real world is the imperative model. Thus this model is typically found being taught in classrooms. Other programming paradigms such as Object Orientated and Functional programming are gaining more traction in the world of computer science thus shouldn't be neglected [10]. Python is a good substitute for learning all three paradigms as the language has the mind-set of object-orientation with inherited features from imperative and extensions from functional [3]. In modern computer science more needs are arising that require knowledge from all three paradigms.

3.3 Web-based Teaching

In the past decade there has been a trend in using the Internet as an aid for learning and creating a learning environment through which students can learn complex subjects such a computer programming. The student typically uses these learning resources in solitude where the platform should be sufficiently well designed so as to guide the user through the learning process. The web-based teaching platforms will never replace the traditional instructor-led learning [8]. There are many cases where classroom learning and face-to-face interactions with an instructor are preferable and are required for an effective learning experience [8]. This web-based form of learning does require us to ask some fundamental questions on how to best deliver the content to the student. For example how can the platform be used to best customize the learning to the individual and how can the learners performance be measured and the material tailored to suit the users specific needs. Modern web technologies allow for innovations to be made in this medium of teaching thus improving the overall goal of education.

3.4 Related Work

3.4.1 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].

3.4.2 CS Circles

CS Circles is a free programming website that aims to teach introductory programming skills an an accessible way [9]. 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 [9].

3.4.3 Online Python Tutor

Online Python Tutor is a web-based program visualization tool for Python that has the purpose of allowing students and teacher to write Python code directly in the web browser. The platform visualizes how the program operates by allowing the user to step through their code. The Online Python Tutor is widely popular boasting over 1.5 million users in as many as 180 countries around the world and used as an instruction tool in the top ranking universities.

Online Python Tutor claims to be one of the only Python program visualization tools that runs on any browser without plugins or third-party software installed. They attribute part of their success to the fact that their platform does not need any setup time on a client's computer and that the webpages load as fast as any other website [6]. Another feature is that Online Python Tutor can be embedded into other webpages through simply adding one line of JavaScript code. The platform is usually embedded alongside electronic textbooks so that students can practice and experiment with the content on the page. Students who interacted with Online Python Tutor were found to score higher on their exams [6].

Figure 1 shows how a simple script can be placed as input

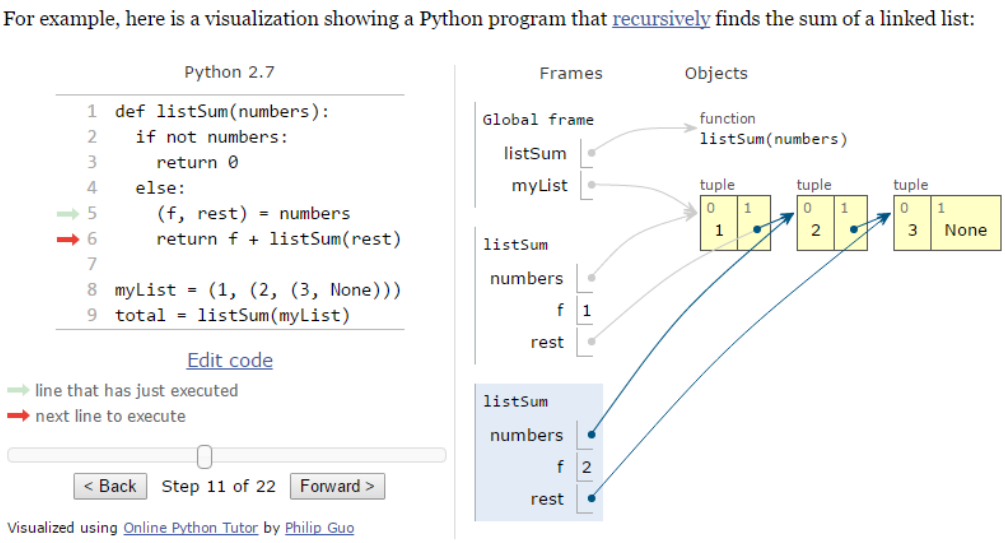

Figure 1: Online Python Tutor (www.pythontutor.com)

into the platform and the output is an execution that allows the user to step through the code and visualise the process that the program is following. The inputted source code is displayed in the top left with a hyperlink below it that allows the user to edit his code. The slider and button input controls allow for the user to navigate through the timeline of the execution fluidly and pause at every instant. The object frame on the right shows different attributes of the variables and objects as well as how they are related to one another. The program output box (not seen in the figure) displays the final output of the program.

The back-end of the program works by taking the input text submitted by the user, executes the text as python source code, generates a stacktrace of the execution and then returns this stacktrace in a defined format [6]. The development team involved started their design by taking common features from existing program visualisation tools and reworking them to be integrated into a web platform. Their initial design was refined and features were added based on feedback from users [6]. In addition to users they also received invaluable advice from instructors in the field.

3.4.4 Developing Effective Web-based Teaching

There are guidelines to follow when it comes to developing an effective platform for web-based learning. Following these guidelines helps to create a more succinct learning environment and ultimately improve the user experience. The following subsections will be divided into two parts; steps that should be taken before the design of the website and factors that need to be considered during the design of the website.

Preparation and Planning.

The initial step in planning involves conducting a needs analysis that includes the problem your website is trying to solve, the needs of the learner that your website will be required to satisfy and an assessment of the current teaching environment [4]. A study needs to be conducted on the current performance of the students in the form of their knowledge and skills and in what aspects they are lacking. A needs analysis will clarify what goals should be set for the platform and in what parts of the current learning experience there is room for improvement [2]. The designer needs to decide whether his platform will supplement existing course material or whether it will become the primary method of instruction. Typically class size is unimportant in an online learning platform as an increase in students generally only result in higher traffic and server load however, if the course requires an instructor to facilitate the learning process, then plans need to be in place in the event of high user inflow. A thorough investigation will help to focus the design and function of the website in order to achieve the educational goals.

Effective design requires that there is an understanding of the technical resources and needs of the website and its users. Online education platforms have unique needs when it comes to the interaction with their students. A specification should be created that clarifies the hardware requirements of the system. This includes an analysis of the system architecture and the required network and computer capacity [11]. Periodic maintenance of the hardware should also be factored in to the specification. The technical requirements of the user should also be considered. Factors such as whether they're comfortable using the Internet, if they've used an online learning website before, what device do they typically use to browse the web i.e. PC, smart-phone, laptop etc. and whether or not they have access to high-speed Internet [2].

Preparation for development requires a research on what features are essential for your website and what software already exists that will aid in implementing those features. It is highly recommended that developers make use of what's available instead of attempting to code new software from scratch. It is also very easy for developers to be caught devoting time to features that are non-essential and do not aid in reaching the goals of the platform.

Commitment from all parties involved in development is a must. In the case of an education platform, development

will require regular advice and feedback from instructors and specialists in the relevant fields. The design of the web platform should be flexible enough so that feedback can be acted upon so as to better reach the needs of the target user and the goals of the system.

Development.

During the development phase of your website there are a few guidelines that should be followed in ensuring an effective learning environment. One of the first important aspects to remember is that the design of the website must go hand in hand with the development of the learning material [2] A common mistake is to create a website and simply place existing content onto the page without any modification or thought as to how applicable it is in an online context [2]. This changes the purpose of the website to an interactive learning platform to more of an information repository. Effective e-learning environments usually make full use of the power of modern web technologies through the modification of how the content is presented to the user.

This highlights the next question the website designer should ask himself during development, how best can I use various web components and features to achieve the goal of the website? Multimedia can be used to supplement and provide value to the learning material [2]. This could be through the use of explaining concepts through visuals such as images and animations or a means through which information can be highlighted by changing the font style and colouring. It's important to keep the goal of the website in mind, as too much multimedia can detract from the learning experience. Something else to also consider is copyright of certain materials. Content that allows individual copies for personal use may not always allow for open distribution on the Internet. The licenses of any material placed on the platform needs to be checked in order to ensure that you are permitted to use the content in your context.

The Internet allows for various forms of communication. These can include emails, forums, chat-rooms, voice calling and instant messaging. Since the ability to interact with people is a big feature of the Internet it is definitely essential for any e-learning platform. There are two types of communication that the designer can choose from. Asynchronous communication where there is a delay in receiving a reply e.g. posting on a forum, chat-room or even sending an email. This has the advantage of allowing the student to write down their wthoughts, reflecting and perhaps arriving at a solution before even receiving a reply. Synchronous communication where the conversation is real-time e.g. voice chat or instant messaging [12]. This closely follows communication between and instructor and student in traditional communication.

The instruction and feedback has an important role to play in an e-learning environment. The same processes that are typically implemented in traditional teaching should be implemented in a web based program [2]. Instruction is conducted through the use of short focused lectures (using the multimedia such as video or images) and feedback through the communication as discussed in the previous paragraph. Regardless of how the feedback presented and delivered it is important that the student always has the opportunity to interact with an instructor.

4. EXPERIMENT DESIGN AND EXECUTION

The Pytut platform and its effectiveness at facilitating the learning of Python concepts was evaluated using a combination of quantitative and qualitative metrics.

4.1 Quantitative Metrics

Google Analytics has been attached to the web platform for the purpose of gathering information on how the user interacts with Pytut. Google Analytics tracks how the user navigates to the various views, the time the user spends on each page as well as which HTML elements the user clicks on and interacts with. The exercises have been separated so that each one is within its own view. Each tool is opened using a button on the interface, clicking each one of these buttons triggers an event in the analytics dashboard. From this we can better understand what sort of process the student goes through in attempting to produce a solution to the exercise problem. We can also correlate the number of button clicks for each tool button with how valuable the user felt the tool was in helping them complete the task. A tool that the user felt was invaluable would have a high number of button clicks, while one that wasn't deemed useful would have hardly been clicked at all.

4.2 Qualitative Metrics

The platform has be assessed from a qualitative perspective through surveys that were completed by the student. These surveys were completed after the test users had spent a significant amount of time navigating around the platform and interacting with the various features. These questions aimed at obtaining a better understanding on how valuable students found the Pytut platform in aiding them to complete basic programming tasks, if they believed that the platform was an effective medium on which programming tasks could be posed to students, whether they thought that the platform alone was sufficient to improve their skills and whether they they needed some sort of introduction to the material beforehand.

4.3 Experiment Groups

It was planned that the testing group for the Pytut platform was going to be the students from the CSC1010H class. These students were new to the Python language,Computer Science and programming in general. They would have been ideal candidates as they would have experienced the hardships of learning this content first hand and would have been motivated to find additional resources to prepare for their end-of-year exams. The original plan was to use two of the CSC1010H's lab sessions and get the students to interact with the Pytut prototype. The first session would have been used of more of a feedback session, where students could voice what they found difficult to use in the platform or what they thought it could be improved, and the second session used as an iteration including those improvements. However, the student protest action brought about unforeseen consequences and were unable to test with our ideal user. To mitigate the effect on the research the advice of 'expert users' was sought to critique the Pytut platform. This expert group was a combination of Computer Science Honours students, Masters students and a few volunteer CSC1010H students. These experiments followed the same structure as what was originally planned i.e a first iteration for critique and a second iteration as a chance to implement some of the feedback.

Table 1: Expert Survey Results

Question	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
Pytut Learning Platform	1	2	3	4	5
This platform will be useful to first year CS students	0	7.1%	7.1%	50%	35.7%
This platform is easy to start working with upon first use	0	0	7.1%	78.6%	14.3%
This platform is an appropriate tool to help students complete lab exercises	0	7.1%	7.1%	35.7%	50%
This platform should be used in conjunction with the current curriculum	0	0	14.3%	57.1%	28.6%
Overall this platform was interact with and easy to use.	0	7.1%	14.3%	42.9%	35.7%
Interactive Code					
This feature will engage students in the programming problem presented to them	0	0	0	71.4%	28.6%
This feature will help beginner programmers detect syntactic errors in their work	7.1%	7.1%	7.1%	42.9%	35.7%
This feature will help beginner programmers detect logical errors in their work	7.1%	21.4%	14.3%	42.9%	14.3%
This feature will allow students to work independently at their own pace	0	0	35.7%	42.9%	21.4%
This feature should be included in future e-learning platforms	0	0	7.1%	57.1%	35.7%
Visual Debugger					
This feature will engage students in the programming problem presented to them	0	0	21.4%	50%	28.6%
This feature will help beginner programmers detect syntactic errors in their work	7.1%	28.6%	28.6%	14.3%	21.4%
This feature will help beginner programmers detect logical errors in their work	7.1%	0%	14.3%	50%	28.6%
This feature will allow students to work independently at their own pace	0	0	35.7%	42.9%	21.4%
This feature should be included in future e-learning platforms	0	0	14.3%	64.3%	21.4%
Interactive Chat					
This feature will engage students in the programming problem presented to them	0	7.1%	7.1%	35.7%	50%
This feature will help beginner programmers detect syntactic errors in their work	21.4%	28.6%	28.6%	14.3%	7.1%
This feature will help beginner programmers detect logical errors in their work	21.4%	14.3%	35.7%	14.3%	14.3%
This feature will allow students to work independently at their own pace	0	0	35.7%	35.7%	28.6%
This feature should be included in future e-learning platforms	0	0	0	35.7%	64.3%

5. SYSTEM DEVELOPMENT AND IMPLEMENTATION

5.1 High Level Overview

The Pytut web platform consists of two core components. The first component is a web server that delivers JavaScript and static HTML to the client and the second component is a NoSQL database. The JavaScript delivered to client runs the various interpreter/tools that facilitate the learning of the Python programming language. The JavaScript also enables the platform to record analytics and generate reports on how the user interacts with the various learning tools. The database is used to authenticate users, store data that is required by the assessment tools as well as storage for the result data that these tools generate.

User flow through the platform typically starts with the user entering the Pytut web server address into their browser. The user is directed to the Pytut login screen where they are prompted for their login credentials. When they have successfully logged in, they are greeted with a dashboard that allows them to navigate to various programming exercises. These exercises appear in the form of an explanation of the problem, a text box in which they can enter their solution and then various tools they can use to aid them in understanding the concepts that the exercise is testing as well as help them generate a high quality solution. The tools used by Pytut will be discussed in the sections to follow.

5.2 Pytut Tools

The Pytut application has various features that are at helping the user complete the assigned programming tasks successfully. These features include Interactive Code, Interactive Feedback and Automatic Assessment.

The Interactive Code feature allows for the user to execute Python code within the browser environment. In the Pytut this is separated into three coding windows. The input code window is where the user writes his solution as well as gives the input to the program. The output window displays the Python interpreted result of running their code. The debug window allows the user to step through their line by line. This helps students evaluate the solution step by step so that they catch mistakes that would of otherwise been hard to see.

Automatic assessment is designed to allow the user to receive instant feedback on their solution attempt. For beginner programmers this can be particularly useful as often incorrect solutions can be easily corrected by pointing out the mistake and providing guidance into what may be wrong. The purpose of this tool is primarily to help students with trivial mistakes in their code, allowing tutors to be free to help students with more complex problems. Automatic Assessment works by comparing the result of running the student's code with the stored model answers for the task. If the student's code produces incorrect results, they are notified that their code doesn't solve the problem correctly and are given the results of the comparison.

Research has shown that personal interaction is invaluable in the learning process [2]. This can be a problem as computer science courses are often too large for consistent one-on-one interaction with the students. The Interactive Feedback tool allows for students to at any time get in direct contact with one of the platform's allocated tutors. Students can ask questions relating to the tasks, send screenshots of the problems they're facing and ask general advice how they can improve on their solution.

5.3 Data Storage

All data has been stored server side in a NoSQL Firebase database. Tables were created to store user information, user statistics, unit tests for automatic marking as well as feedback information. All transactions and table management is handled by Firebase. This storage is persistent and was vital for the final analysis of how students performed on the platform.

5.4 Development Platform

Pytut was built using the AngularJS JavaScript framework. This framework follows the MVC design paradigm where each view in the platform has its own controller. All JavaScript is delivered and run client-side. The Firebase real-time database is used to store the questions and model answers for the automatic assessment, the users solutions for each exercise as well as the authentication profile for each user. Google Analytics has been attached to the website and is used to monitor the performance of the user for each exercise as well as how the user interacts with the various tools we've provided to help them. The website is hosted on Firebase's hosting platform.

6. RESULTS AND FINDINGS

6.1 Implementation of Design

The final implementation consists of a simple dashboard written using the AngularJS framework. Each of the test users received a simple email that welcomed them and introduced them to the research project and what we required from them. In addition to this they were given their login credentials to the platform. They were instructed to visit the webpage located at https://pytut-eedf2.firebaseapp.com and follow the instructions from there. Once the student had successfully logged in and accepted the terms of use for the website they were presented with a view that laid out their first programming task. This layout includes a description of the problem, a text area that the user could enter their solution, a textbox for 'raw_input' entries,an output window that shows the result of their attempt as well as three buttons that corresponded to the additional tools that can be used to aid the student in finding a solution. Figure 2 shows a screenshot of the final design.

The first button "Run Code" allows the user to see the output of the solution they've typed in the code window. This uses a Skulpt as a JavaScript based Python interpreter and allows the user to see if there are any errors in the code and whether their attempt at a solution to the task is producing desirable results. The second button is "debug code" which opens a third window and allows the user to step through their code line by line as seen in Figure 3. This debugger uses the Online Python Tutor tool developed by Philip Guo. The debugger helps students have a deeper understanding on how Python is interpreting their code and at which point their code may be failing or performing incorrectly. In each step the student can see which line has just executed, which line will execute next, all of the currently defined variables and the effect that each step has on those variables. The third button "Submit for grading" allows the user to run their code and compare it to predefined model

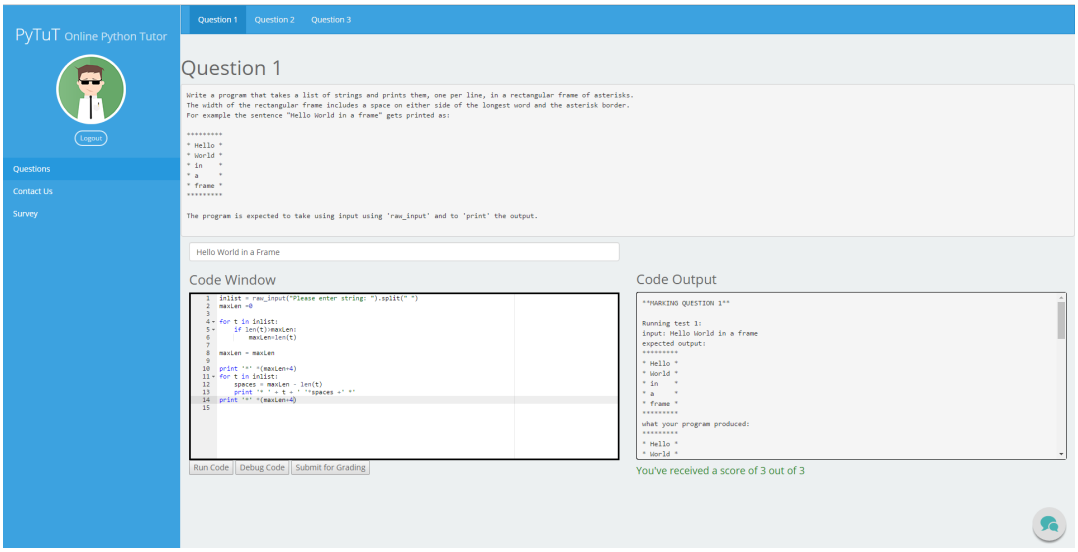

Figure 2: screenshot of Pytut with complete solution

answers. The user receives a score based on the correctness of their solution, if the code produces incorrect results, the marker provides feedback on differences between what the user's code produced and what was expected by the model answer. In each iteration there were three different questions that were asked, these can be found in Appendix A.1 and A.2. These questions aimed at testing the student's understanding of some of the fundamental concepts in Python programming. All of the questions involved writing simple input/output programs that solved a simple problem. The areas covered were string manipulation, simple control flow, conditional statements and basic algorithms.

The Pytut prototype implements a chat feature using a plugin from Tawk.to. On every page of the website there is a chat icon in the bottom right. At any point the user has trouble with something i.e logging in, navigating the platform or more importantly, difficulty in solving one of the problems, the user can open a one-one-one realtime chat with one of the tutors on standby. This chat supports uploading files and media so that the user can further clarify their issue. A "Contact Us" tab provides students with details through which they can contact the project administrators and the "Surveys" tab directs students to the survey that is hosted as a Google Form. The users are instructed to complete this once they have attempted each question.

The model answers, the user profiles, partial and completed solutions are all saved in the Firebase Database. Analytics for the usage on the chat tool is stored in the Tawk.to admin dashboard.

6.2 Survey Results

6.2.1 Student Survey

Students were asked to complete the survey after they had put in a significant amount of effort in completing the coding tasks. The survey questions were mostly posed as multiple choice questions where the choices ranged from 1 (Strongly disagree) to 5 (Strongly Agree). Unfortunately due to unforeseen protest actions we were unable to get significant engagement from the CSC1010H students. There was a relatively low response for the student survey and thus the results didn't bring conclusive results.

6.2.2 Expert Survey

Experts were asked to complete the survey after they had spent some time navigating around and interacting with the Pytut platform. The survey was mostly posed multiple choice questions where the choices ranged from 1 (Strongly disagree) to 5(Strongly agree). The goal of the survey was to obtain a deeper understanding on whether the users (from an expert view point) could find value in what the Pytut prototype was providing in terms of tools to help the novice programmer. The questions as well as the percentage spread for the answers has been tabulated in table 1. In brief the majority of expert users felt there was definite value in the tools that the Pytut platform could provide novice programmers. A large portion felt that Pytut would be valuable in the current curriculum as can be in see in Figure 5 and would be a valuable tool in lab exercises as seen in Figure 5. The majority of the expert users felt that the online tutor allowed for students to work at their own pace and the majority either 'agreed' or 'strongly agreed' that this kind of tool would be best used in conjunction with regular teaching. Users felt that all three tools could be further improved in aiding students find errors in their code. Suggestions included highlighting the lines in code where the errors were reached, providing error messages that are more 'novice friendly' and providing a debugging feature that gave the user more control with easier to understand code tracing. Another suggestion was to introduce the student to the exercise with code that was at least partially scaffolded thus providing direction to students who may be struggling with the content.

Certain aspects of the platform were altered based on feedback given from the first iteration. These changes included the saving of partial and completed solutions which allows students to continue where they left off should they choose to leave the website, removing the alert prompt for user input and instead placing a dedicated text box and finally fixing the CSS and HTML to make the exercise views more

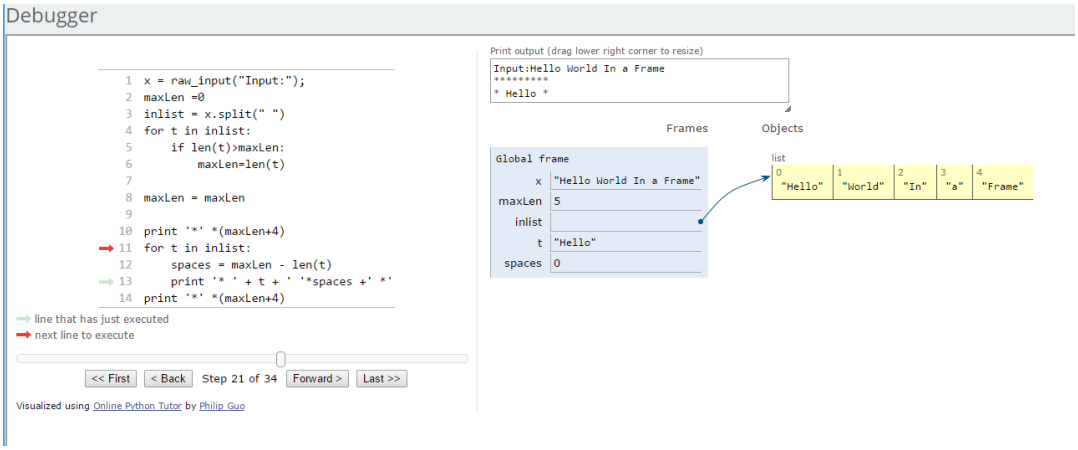

Figure 3: screenshot of viusal debugger (using Online Python Tutor by Philip Guo)

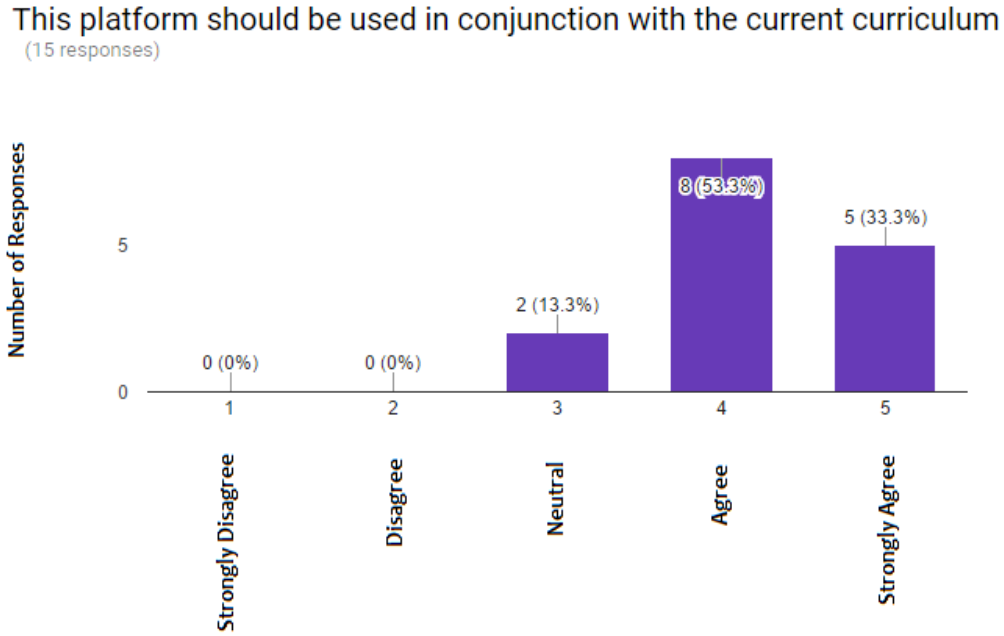

Figure 4: Graph showing user response on whether Pytut was a good supplement to the current curriculum

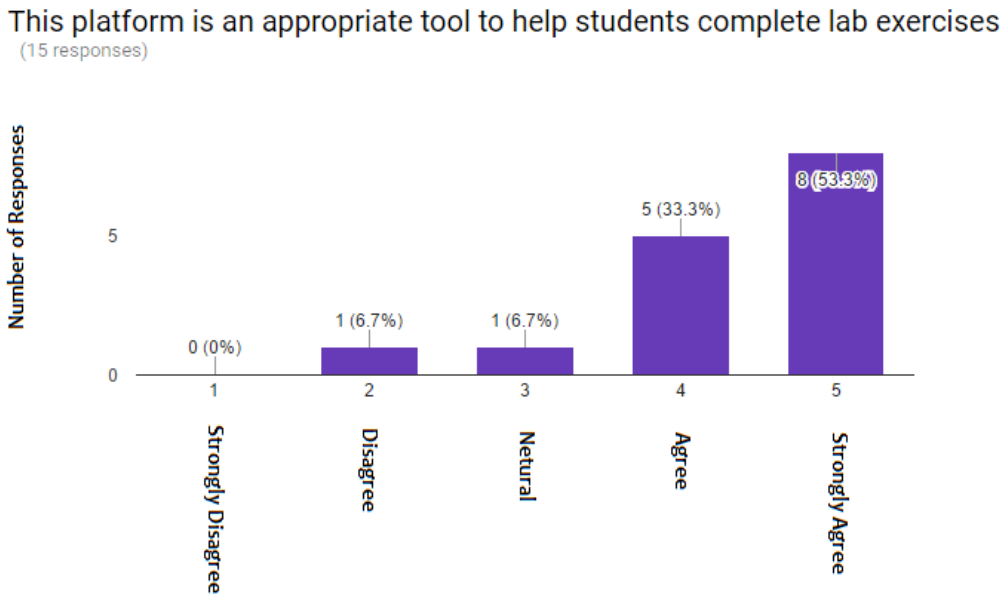

Figure 5: Graph showing user response on whether Pytut was a good supplement to the current curriculum

visually appealing.

6.2.3 *Analytics*

Google Analytics recorded a total of 13 users who successfully logged on to the Pytut prototype. The average time spent on the platform was recorded at 33 minutes. Of the 13 users who logged in only 7 attempted every question in the first iteration with the most interaction being on question 1 (of which 9 out of 13 participants attempted). The average time spent on question 1 of the first iteration was 23 minutes. Since this question received the most attention in the first iteration we'll explore some of its recorded metrics a bit more in depth. For question 1 on average a user ran their code 18 times, they interacted with the debugging feature 3 times and they submitted their code for grading twice. Three participants managed to submit their code for grading and receive a perfect score i.e. the output of the user code was compared to the model answers and deemed to be equal.

Records of the second iteration show that there were 4 return users from the previous iteration and 3 new users who have joined the study. This bring a total of 7 users who have participated in the second stage of testing. Given the three questions in this stage, question 1 had the most user time with an average of 17 minutes. The runner-up was question 2 which had an average time of 16 minutes. Question 3 didn't receive any engagement. When looking at question 1 the user ran their code on an average of 26 times, debugged their code once and submitted their code for grading twice.Three participants managed to submit their code for grading and receive a perfect score for comparison against the model answers.

Analytics for the chat feature shows engagement with 8 users with a total of 30 messages being passed between the tutors on standby and the users of the Pytut prototype.

All 13 users recorded in Google Analytics were either Honours or Masters Computer Science students so the analysis of data should take into account that the prototype was being used by experts. This isn't ideal usage as the exercises as well as the tools are designed for novice programmers still learning the basics of Python. This would explain the lack of engagement for all of the questions in each iteration. The critiques given by results from the Expert Survey does hold more weight when compared to the CSC1010H students as the Masters and Honours participants have had more experience in learning programming concepts from various e-learning resources. When comparing the analytics to the survey results one can see why there has been a lack of interaction with the debugging and automatic assessment tools. Feedback from the survey brought to our attention that the debugger would be far more user friendly and effective if it was attached to users code window instead of rendering a separate window that had different styling and seemed to throw the entire trace of their code at them all at once. An example of this is illustrated by Figure **??**. The user can clearly see the cursor move through his code as he steps through each line. The list on the left shows all of the currently defined variables and the result of each line that has been executed.

7. CONCLUSION

The aim of this research project was to discover whether an online learning platform that had the purpose of teaching novice programmers how to code in Python could provide value as a supplement to classroom teaching. A secondary aim was to determine whether a teaching platform that was interactive and responsive to the student's actions could be valuable to their learning of a new programming language.

Due to the student protest action our experiment group mainly consisted of a collection of experts who have experience in programming languages as well as how they're taught. A small number of volunteer students from CSC1010H were also included in the study. The experiment was conducted by getting everyone to spend 45 minutes interacting

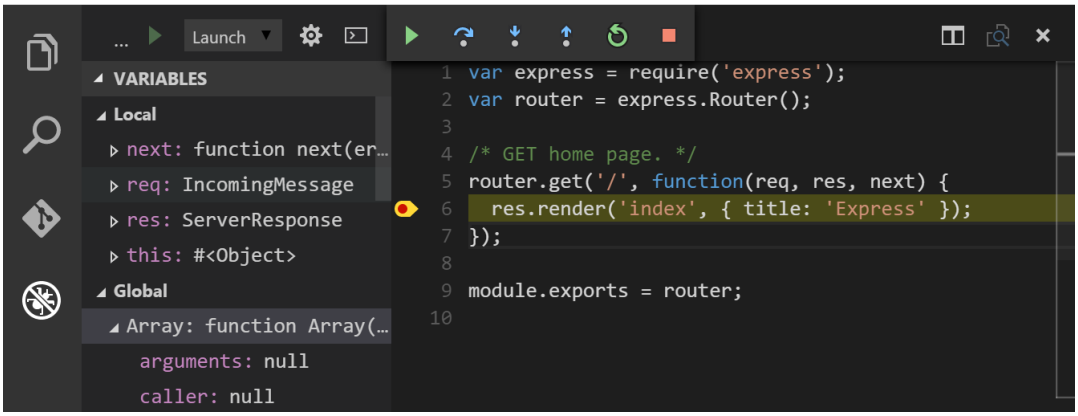

Figure 6: NodeJS code being stepped through in the Visual Studio Code IDE

with Pytut and gauging platforms success through usage analytics as well as user surveys. The expert and students groups were separated and their responses were weighted accordingly. The survey results from users who had longer session times, were found to have made significant attempts in solving the problems and explored the various views were higher valued This experiment was completed twice, once to obtain feedback and ways in which the platform could be improved and a second time to see whether these suggested improvements increased the effectiveness of the platform as a whole.

The findings from both the analytics and user surveys showed that the platform definitely provided features that aided the student in completing programming tasks and helping them improve their understanding of the Python programming language. The general consensus from the usage analytics and survey results from expert users suggests that Pytut is an ideal supplement for the first year CS students to become more acquainted with Python. That the platform is easy to use and should be used in conjunction with lab exercises and the current curriculum. The results from the survey suggest that the Interactive Code, Visual Debugger and the Automatic Assessment would all be effective in aiding students with the given coding tasks. Experts strongly agreed that Interactive Chat was the most engaging feature, however, they also found it the least effective in aiding the student with the coding tasks. The analytics gathered showed significant engagement with the platform. The Interactive code feature was by far the most used feature with Automatic Assessment being second, Interactive Feedback with a tutor third and finally the Visual Debugger. Most users didn't interact with all of the programming tasks, picking one and trying to complete it fully during their session. This is to be expected as the users weren't novice programmers but rather part of our expert testing group. The two first year volunteers showed more engagement by attempting each task at least once.

Improvements to Pytut could include the addition of learning material before the student solves each coding task. This material would introduce the user to the concepts they will require in solving the exercise. Because this learning material is relevant to the student and has been reviewed shortly before its application in a programming task, it is believed that the student will have a better understanding of the concept and the knowledge will be retained easier. There are numerous improvements that could be made Pytut's current features. The live chat feature could be altered to introduce the student to a tutor when they login. This tutor would then direct them to the tasks they have to complete as well actively engage them to see provide advice on how their code could be improved and how they can overcome the challenges they may face in each of the programming tasks. Results from the surveys suggest that the function of the interactive code and visual debugger can at times be confusing. The interactive code should report errors that are more understandable from the perspective of a novice programmer. The debugger,where users can step through their code, should be integrated into the user's coding window and not separated. Finally the exercises should be placed in different sections, each section covering a different aspect of Python i.e string manipulation, data structures etc.

8. FUTURE RESEARCH

Future researching in the field could include implementing the above improvements in a larger experiment group or possibly even as a prescribed supplement to UCT's CSC1010H curriculum. The larger testing group could discover other tools and features that are effective in teaching Python to first time programmers. Various other forms of interactive media could be investigated such as the use of a digital whiteboard where tutors could better explain concepts through the use of illustrating them on the Pytut platform.

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

[3] F. Georgatos. How applicable is python as first computer language for teaching programming in a preuniversity educational environment, from a teacher's point of view? *CoRR*, abs/0809.1437, 2008.

[4] T. Govindasamy. Successful implementation of elearning: Pedagogical considerations. *The Internet and Higher Education*, 4(3âĂŞ4):287 - 299, 2001. {SOLE} 2001: Exploring the Evolution of ELearning.

[5] L. Grandell, M. Peltomäki, R.-J. Back, and T. Salakoski. Why complicate things?: Introducing programming in high school using python. In *Proceedings of the 8th Australasian Conference on Computing Education - Volume 52*, ACE '06, pages 71--80, Darlinghurst, Australia, Australia, 2006. Australian Computer Society, Inc.

[6] P. Guo. Online python tutor: Embeddable webbased program visualization for cs education. In *Proceedings of the ACM Technical Symposium on Computer Science Education (SIGCSE)*, 2013.

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

[8] B. Lin and C. tseh Hsieh. Webbased teaching and learner control: a research review. *Computers & Education*, 37(3âĂŞ4):377 - 386, 2001.

[9] D. Pritchard and T. Vasiga. Cs circles: An inbrowser 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.

[10] E. Shein. Python for beginners. *Commun. ACM*, 58(3):19--21, Feb. 2015.

[11] T. Sitzmann, K. Kraiger, D. Stewart, and R. Wisher. The comparative effectiveness of webbased and classroom instruction: A metaanalysis. *Personnel psychology*, 59(3):623--664, 2006.

[12] D. Zhang and J. F. Nunamaker. Powering e-learning in the new millennium: An overview of e-learning and enabling technology. *Information Systems Frontiers*, 5(2):207--218, 2003.

APPENDIX

A. EXERCISE QUESTIONS

A.1 Iteration 1

A.1.1 Question 1

Write a program that takes a list of strings and prints them, one per line, in a rectangular frame of asterisks. The width of the rectangular frame includes a space on either side of the longest word and the asterisk border.
For example the sentence "Hello World in a frame" gets printed as:

```
*********
* Hello *
* World *
* in *
* a *
* frame *
*********
```

The program is expected to take using input using 'raw_input' and to 'print' the output.

A.1.2 Question 2

Only South African citizens who have registered in their municipality and are over the age of 18 may vote in the local elections. A special round of voting runs before the main election for elderly citizens (75 years or older) and people with disabilities. Write a program that takes a list indicating an individuals age, whether they are registered, with 1 indicating being registered and 0 indicating not registered, and whether they are disabled, with 1 indicating being disabled and 0 indicating not disabled. The program then outputs whether they can vote and which round of votes they will vote in (regular or general).

Note that output should take the form "can vote" or "cannot vote" and "special election" or "regular election"

Input format: [age, registered, disability] For Example 18 0 0 = 18 years old, not registered, not disabled -> cannot vote

75 1 0 = 75 years old, registered, not disabled -> can vote special election

60 1 1 = 65 years old, registered, has disability -> can vote special election

35 1 0 = 35 years old, registered, not disabled -> can vote regular election

The program is expected to accept input using 'raw_input' and to 'print' the output. The input should be space seperated.

A.1.3 Question 3

Write a program that takes an input string that describes a calculation (eg: "1 + 3 - 1 =") and calculate the answer to the calculation. If the string does not end with an "=", it is considered to be invalid. Strings will only contain + and - so you do not need to account for arithmetic precedence.

For example: "1 + 5 -2 =" -> 4 "1 + 3" -> invalid input

The program is expected to take using input using 'raw_input' and to 'print' the output.

A.2 Iteration2

A.2.1 Question 1

Write a program that asks the user how many Fibonacci numbers to generate and then generates them. Make sure to ask the user to enter the number of numbers in the sequence (using raw_input) to generate. (Hint: The Fibonacci sequence is a sequence of numbers where the next number in the sequence is the sum of the previous two numbers in the sequence. The sequence looks like this: 1, 1, 2, 3, 5, 8, 13,...). The result should be output using 'print' and the numbers should be separated with a ',' i.e. 1,1,2,3,5

A.2.2 Question 2

Write a program that asks the user (using raw_input) for a long string containing multiple words. Print back to the user the same string, except with the words in backwards order. For example, say I type the string:

My name is Michele

Then I would see the string:

Michele is name My

shown back to me. Please 'print' the result.

A.2.3 Question 3

Create a program that asks the user for a number (using raw_input) and then prints out a list of all the divisors of that number. (If you don't know what a divisor is, it is a number that divides evenly into another number. For example, 13 is a divisor of 26 because 26 / 13 has no remainder.) Each

number of the result should be printed on a newline i.e The result when the user inputs 4 would be : 1 2 4

B. QUESTIONNAIRES

B.1 Student Questionnaire

Please refer to https://goo.gl/forms/8qqWGZOj0VOwkxUH3 to view student questionnaire.

B.2 Expert Questionnaire

Please refer to https://goo.gl/forms/cN2OgU3BoPJl3A4x1 to view form.