Project Proposal

By Scott Bartell

The members of your project team

This project will be completed individually by Scott Bartell

A task list of the tasks that must be completed to execute and deliver your project

Features and Scenarios to be Implemented

Below are the main features and initial scenarios that will be implemented throughout this project:

Feature: Answer free form questions
As a Codecademy user who is taking the Introduction to Ruby course
I want to ask a question on a lesson page
Because I don't understand something about programming

Scenario: User asks if spaces are important[1]
Given I am logged into Codecademy
When I go to an exercise
And ask "Does it matter if there is a space between my = sign and value?"
Then I should see "Nope! In Ruby whitespace is ignored when code is executed."

Feature: Give automatic feedback
As a Codecademy user who is taking the Introduction to Ruby course
I want to receive helpful feedback on my code
Because I don't understand why my code is not working

Scenario: User forgets to wrap a string in quotes[2]
Given I am logged into Codecademy
When I go to Exercise 1.5
And enter code trying to print a string without wrapping the string in quotes
And click "Save & Submit Code"
Then I should see "Hey, it looks like you forgot to put the string you are trying to print in double quotes."

---

[1] https://www.codecademy.com/forum_questions/515d019c16dbeb09f1000d91
[2] https://www.codecademy.com/forum_questions/5427f56b548c35d68500076a

Task List

Milestone 1

- *(partially completed)* Implement a chatbot to answer the first “free form question” scenario (this will be hosted on its own web page) [6]
- Identify, implement, and test 5+ more “free form question” scenarios [4]
- *(partially completed)* Implement an interface for the chatbot to work directly on the Codecademy website [20]
    - Create Chrome Browser Extension that embeds a basic chat window in the user’s browser when visiting Codecademy lessons
- Improve the visual appearance of the browser chat window [3]
- (Stretch Goal) Have chat window support code/syntax highlighting [5]
- Prepare Milestone 1 deliverable [4]

Milestone 2

- Complete user testing
    - Find target users and observe them using the tool [6]
    - Identify and create additional user scenarios [4]
    - Incorporate other feedback from the user testing [5]
- Implement the first “automatic feedback” scenario [20]
- Prepare Milestone 2 deliverable [4]

Final Paper

- Identify, implement, and test 5+ more “automatic feedback” scenarios [6]
- Incorporate any additional feedback [5]
- Prepare project code for submission
    - Package Chrome extension for non-development use and document how to install it [3]
    - Refactor and clean up code [5]
- (Stretch Goal) Submit Browser Extension to Chrome Web Store [3]
- Prepare the project presentation [3]
- Prepare the project paper [5]

Descriptions for your two intermediate milestones

Milestone 1

At the end of milestone 1 a fully functional prototype with about half of the full feature set will be delivered. A Chrome browser extension will be delivered that once installed will allow the user to visit the Introduction to Ruby exercise on Codecademy.com and ask questions about

programming. The questions will be asked through a chat box that is embedded directly on the exercise page. There will be at least 5 different commonly asked questions that the tool will know how to respond to.

Milestone 2

Milestone 2 will start with user testing the features delivered in Milestone 1. The feedback from the user testing will be used to further improve the existing features and to identify and develop additional questions for the tool to support.

Additionally, the second major feature will be implemented and delivered: automatic feedback. When Milestone 2 is delivered, in additional to the features mentioned previously, users who make certain mistakes when working on an exercise from Codecademy.com's Introduction to Ruby course will receive detailed feedback from the tool. There will be at least 5 different common mistakes that the tool will be able to identify and provide feedback for.

A division of responsibilities among the members of the project team

N/A, this project is being completed individually

Development Track Questions

A description of the problem to be solved

Programming is a complex topic and it cannot be effectively taught through explication or traditional lecture alone[3]. Because of this, introduction to programming courseware is often built around project-based work and practical experimentation. A major challenge with this type of project-based programming course, is providing the student with a personalized learning experience and immediate feedback during problem solving[4]. There is a huge opportunity to solve this problem by using technology to help unstick students who get stuck when learning how to program through project based work.

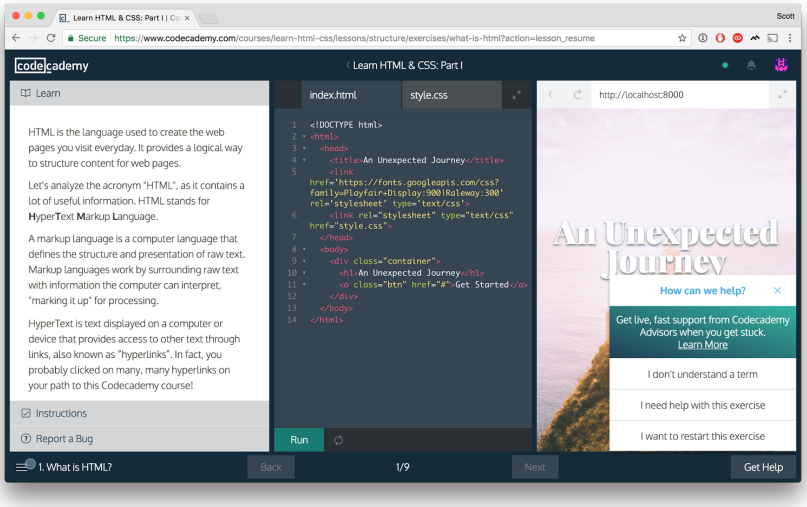
[3] Neve, Paul, et al. "NoobLab: an intelligent learning environment for teaching programming." Proceedings of the The 2012 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology-Volume 03. IEEE Computer Society, 2012. APA

[4] Gomes, Anabela, and António José Mendes. "Learning to program-difficulties and solutions." International Conference on Engineering Education–ICEE. Vol. 2007. 2007.

A description of existing solutions for that problem, specifically to contextualize why your solution is needed.

Codecademy.com is an online interactive platform that offers free self-paced student-driven coding classes to the public[5]. Codecademy provides students with programming projects and exercises along a browser based code interpreter that allows students to enter and execute their solutions directly in the browser. When you start a new exercise on Codecademy there is a “Get Help” button on the bottom right to help students who have free form questions get help (see the bottom right box in the image below)[6]. These questions can be answered in one of three ways:

1. By having a reference glossary of common terms
2. By having community driven forums where students can post and answer questions
3. By having real Advisors help you (this is not free and costs ~$20/mo)

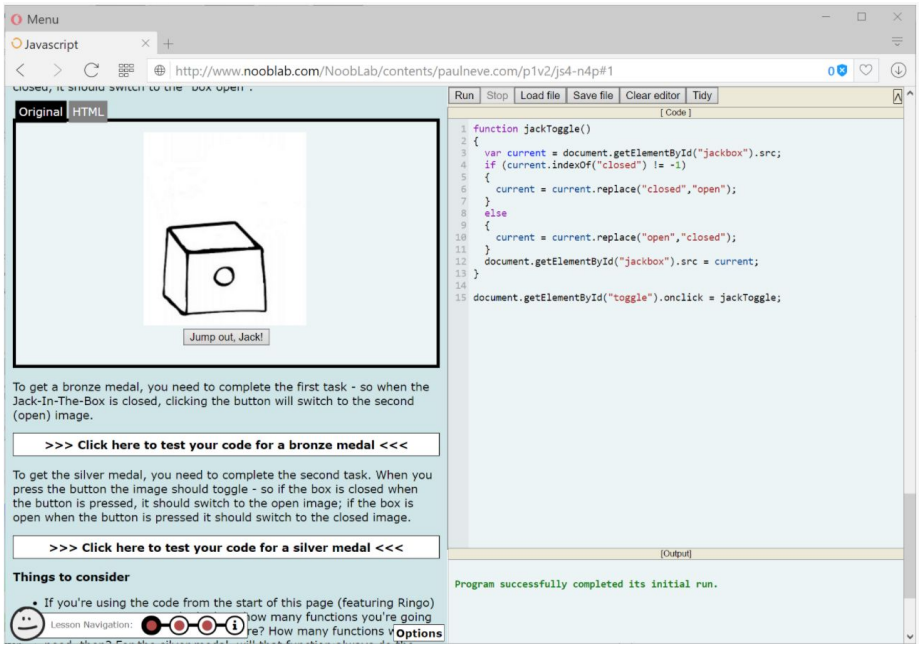


Traditional classrooms also are attempting to solve this problem. Some classrooms have modified their introduction to programming curriculum to center around practical experimentation. In order to ensure that students are learning efficiently these classes have retained some lecture-style learning but reduced it to a minimum. Additionally, the teacher’s role in these classrooms has shifted to be a “guide on the side”.[7] In theory, this means that when a student is learning to program a teacher is available to provide the student with a personalized learning experience and immediate feedback during problem solving. However, in practice, the teacher isn’t always available when the student needs them.

---

[5] "Codecademy." https://www.codecademy.com/. Accessed 4 Jun. 2017.
[6] "Learn HTML & CSS: Part I | Codecademy." https://www.codecademy.com/en/courses/learn-html-css/lessons/structure/exercises/what-is-html. Accessed 28 May. 2017.
[7] Neve, Paul, et al. "NoobLab: an intelligent learning environment for teaching programming." Proceedings of the The 2012 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology-Volume 03. IEEE Computer Society, 2012. APA

Nooblab is an online learning environment for teaching programming that takes the concepts from Codecademy and traditional classrooms and attempts combine them[8]. Nooblab presents learning content alongside an area which allows simple Javascript-based programs to be composed and run in a virtual console (as shown in the picture below).



This allows Nooblab to supplement a teacher's "guide on the side" role by providing automated feedback directly to the student as they type code into the tool. This takes some of the burden off of the teacher by automating student feedback for some specific problems (e.g. syntax errors).

Current Solution Pitfalls

The current solutions for this problem are either inadequate or expensive.

The ideal solution to this problem is providing a dedicated teacher that is always available to help every student learn how to program. This solution has one fundamental problem: it is expensive.

The best solutions to this problem offer some variant of the ideal solution: Codecademy allows you to purchase access to their "advisors" and traditional classrooms have teachers whose students can request and receive feedback. In both of these examples the teachers are actually shared resources among many students; this is done entirely to reduce the cost.

Nooblab attempts to further reduce costs by automating some feedback to the student and

[8] "NoobLab: The online environment for teaching programming." http://www.nooblab.com/. Accessed 28 May. 2017.

therefore allowing teachers to either increase the size of the pool of students or focus on other things. While this tool does show great promise it has two major shortcomings:

1. The provided feedback is limited
2. It isn't accessible to the public

Nooblab can only give feedback based on the code that student types in; this is inherently limiting and means that any free form questions that a student has will still need to be directed towards teachers. Additionally, Nooblab is designed and intended to only be used inside schools and Universities and not the public as a whole[9].

A description of the design of the tool you will create.

The tool will be an intelligent tutoring system for students who are learning to program through Codecademy's Introduction to Ruby course. To install the tool students will need to simply install a Chrome Browser Extension. After the tool is installed, if students make some common mistakes while taking the course they will automatically receive feedback and helpful messages from the tool guiding them through their learning process. Additionally, students who have general questions will be able to ask free form questions in plain english to the tool and receive helpful answers.

A technical description of the tools, languages, and other resources that will be used.

There are two main technical components for this tool: the Chrome Browser extension[10] and the conversational backend.

The Chrome Browser extension will hold all of front end logic for the tool including HTML, CSS, and JavaScript and interface with the conversational backend. The extension will include a component for the chat box, an integration of that component to the conversational backend, and logic that dynamically identifies user interactions (e.g. code submission) and passes that information to the conversational backend.

The conversational backend will be driven by api.ai. Most of the technical work in this area will be spent integrating with and configuring this tool.

---

[9] "NoobLab: An Intelligent Learning Environment for ... - Paul Neve." http://paulneve.com/macau.pdf. Accessed 28 May. 2017.

[10] https://developer.chrome.com/extensions

A description of the integrations or external resources that will need to be obtained, as well as fall-back plans in case portions of these details cannot be completed.

Browser extension - since the tool is going to be supplementing Codecademy's existing content a browser extension is going to be implemented. If any unforeseen challenges arise with this approach a very basic version of Codecademy can be implemented manually (we just need a web page that can execute code) to demonstrate the basic concepts.

Api.ai - the tool will be integrating with Api.ai for the conversational backend. If any problems arise with the integration either other tools can be considered (such as IBM Watson Conversation) or a basic, more limiting, tool can be developed from scratch.