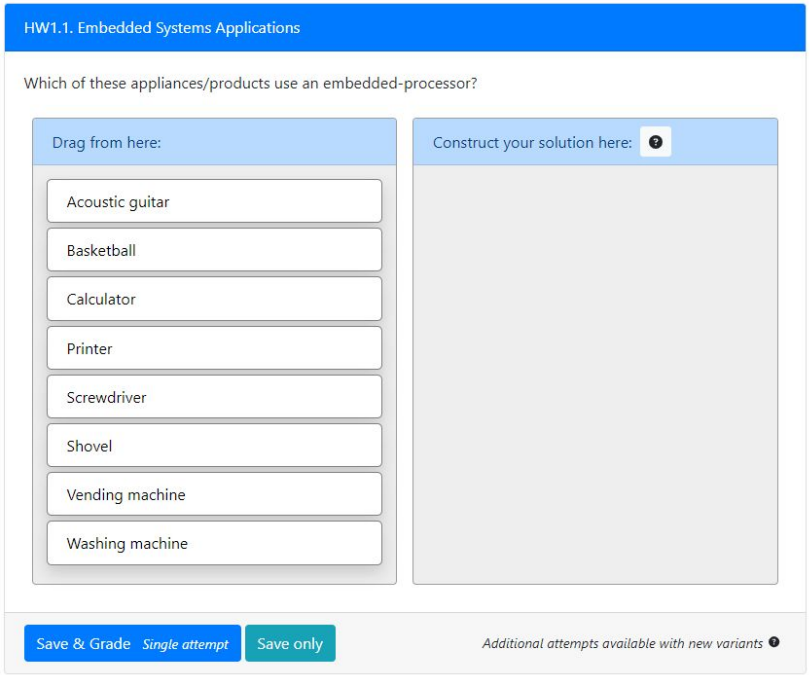
User Needs and Requirements

sdmay25-33

Caden Otis, Devin Alamsya, Justin Cano, Joseph Krejchi, Rachel Druce-Hoffman

Project Overview

- Create an interactive application for CPRE 2880 students to better understand the concepts
  - HWs and quizzes
  - Randomized questions and autograding
  - Use emulator tools to simulate microcontrollers
  - Potentially have an emulated Cybot robot interface
- PrairieLearn framework to host the application
- Utilize Python, JavaScript, C and other programming languages
- Hope to inspire other professors to build similar interactive tools for their students



Problem Statement

- Students don’t get enough practice of concepts
    - Little feedback on Canvas HW submissions
- Not always availability to practice programming on the microcontroller in the lab
- Limited time to meet with Professor and TAs
    - Lab, class, office hours
- Limited capabilities with Canvas platform

CPRE 2880 Student Needs

- Get specific and quick feedback on assignments
    - Want to learn from their mistakes and improve their knowledge
- Have questions that are engaging and interactive
    - Students are involved enough with the current questions asked in the course
- Randomize problems for unlimited practice
    - Questions from HWs and quizzes don’t provide students with enough practice

CPRE 2880 Professor Needs

- Have autogradable questions
  - Professors juggle multiple classes, research and personal life
  - Saves time from having to grade
- Makes concepts easier to understand
  - Will lead to excellent student performance in the course
- Randomization for less time designing questions
  - Will allow students to practice different variations of questions
  - Will help increase student performance

CPRE 2880 TA Needs

- Have autogradable questions
  - TAs are students themselves
  - Saves time from having to grade
- Help students explore the subject instead of answering introductory questions
  - Encourages more critical thinking and active learning
  - Makes students engage with concepts on a more meaningful level

Functional Requirements

- All homeworks should be implemented
  - Code of each homework should also be documented for future development
- Most questions should be autograded
  - Includes student-written coding segments
- All questions should be randomized for unlimited practice
  - As many parameters within the problem should be randomized as possible

UX/Resource/Aesthetic Requirements

- USER EXPERIENCE: New question types thought of and implemented with a focus on interactiveness
- USER EXPERIENCE: Questions formatted in a way that is easy for the user to understand and interact with
- RESOURCE: Implement the Virtual Cybot/Emulated Cybot interface into our project so students can test their code.
- RESOURCE: Documentation written about each aspect of our implementation
    - Allow for continued development
    - Tutorials for other classes setting up PL
- AESTHETIC: No bugs/typos

Engineering Standards

- **IEEE/ISO/IEC 14764-2021**- *Software engineering, Software life cycle processes, Maintenance*
  - Specifies requirements for long term use of the software.
  - Important to consider how software change over time (supporting API changes, changes to the course, etc.)
- **IEEE/ISO/IEC 29119-1-2021**- *Software and systems engineering, Software testing Part 1*
  - Describes the testing process for software.

Conclusion

- Three different user groups with varying needs
  - CPRE 2880 Students - need practice questions that encourage learning
  - CPRE 2880 Professors - have autogradable questions that can be randomized
  - CPRE 2880 TAs - have questions that are autogradable and reinforces the basics of concepts
- Our project should be easy to use for students and make grading seamless for professors and TAs
- Our project can incorporate several different engineering standards to make a software product that will last for a long time with minor bugs or issues

Any Questions or Suggestions?