Contextualization/Design Check-In

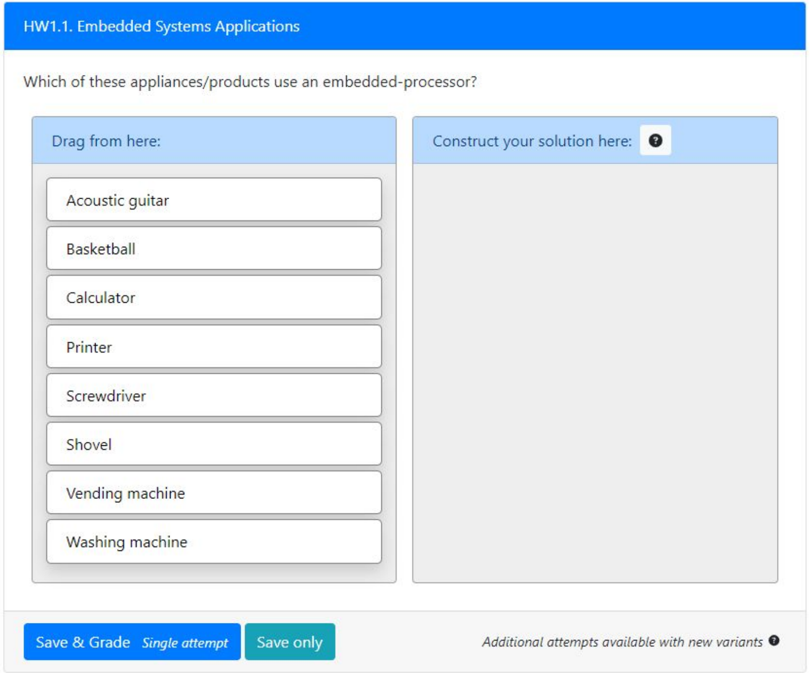
Team: sdmay25-33

Advisor: Phillip Jones

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 emulation 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



HW1.1. Embedded Systems Applications

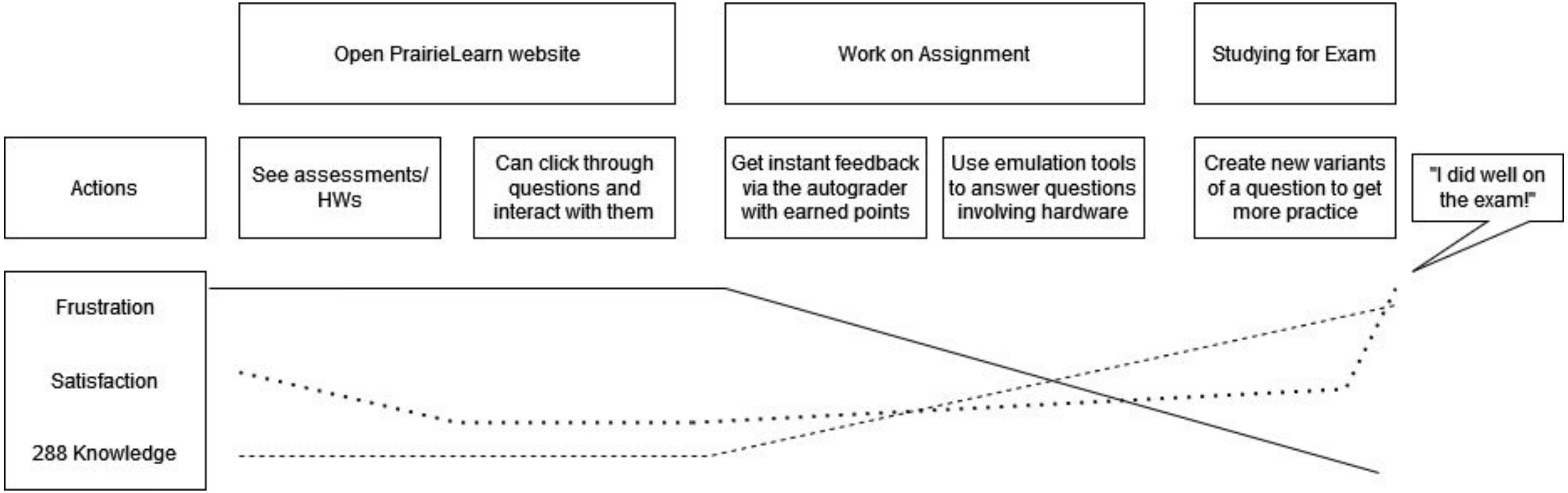
Which of these appliances/products use an embedded-processor?

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

Journey Map

Persona: CPRE 2880 Student



Open PrairieLearn website

Work on Assignment

Studying for Exam

Actions

See assessments/ HWs

Can click through questions and interact with them

Get instant feedback via the autograder with earned points

Use emulation tools to answer questions involving hardware

Create new variants of a question to get more practice

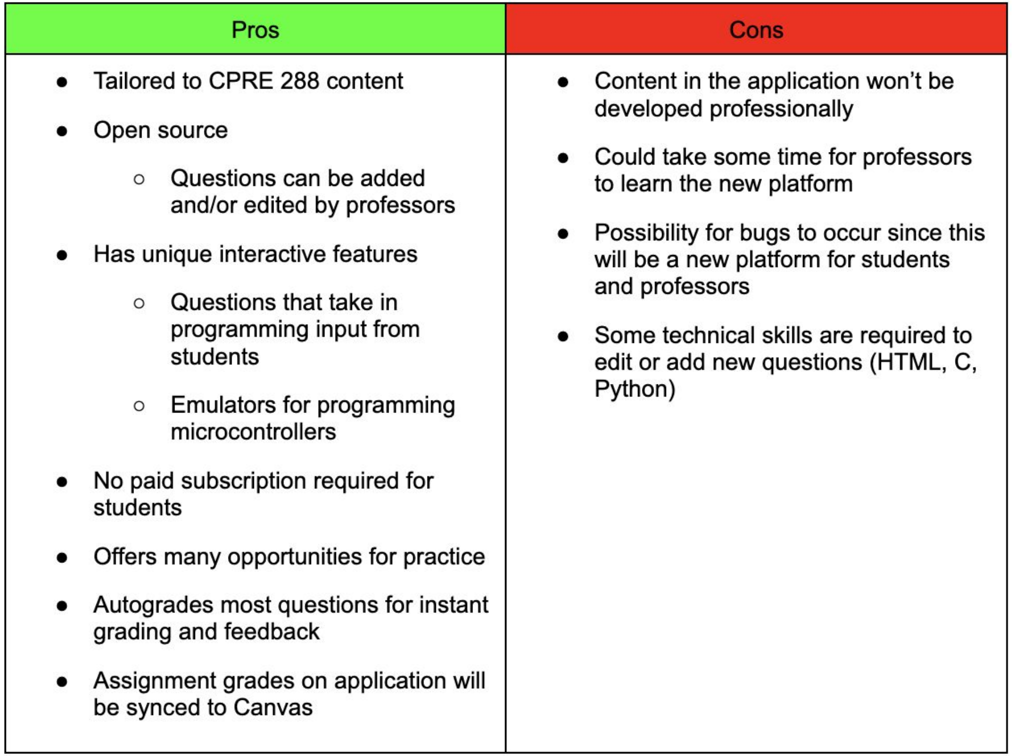
"I did well on the exam!"

Frustration

Satisfaction

288 Knowledge

Pros and Cons Table



Pros	Cons
● Tailored to CPRE 288 content ● Open source ○ Questions can be added and/or edited by professors ● Has unique interactive features ○ Questions that take in programming input from students ○ Emulators for programming microcontrollers ● No paid subscription required for students ● Offers many opportunities for practice ● Autogrades most questions for instant grading and feedback ● Assignment grades on application will be synced to Canvas	● Content in the application won't be developed professionally ● Could take some time for professors to learn the new platform ● Possibility for bugs to occur since this will be a new platform for students and professors ● Some technical skills are required to edit or add new questions (HTML, C, Python)

Internal and External Tech. Complexities

- Internal Complexities
  - Linux
  - Docker
  - ISU VM
  - Python, C, HTML
  - Course/Student Management
  - Custom Emulators for ARM and Cybot
  - Custom autograders container for ARM assembly questions
- External Complexities
  - Engaging question elements (use what’s available or develop our own from ground up)
  - Other solutions are equivalent or lesser (288 relevance, ease to implement, flexibility)

Addressing User Needs

- All existing CPRE 2880 HWs have been uploaded to our application with randomization features
  - Useful for practice and studying
- Questions can be autograded and provide instant feedback to students
  - Relieves burden of professors and TAs manually grading assignments
- PrairieLearn is free to use for students
- Change: integrate Canvas into our application to sync grades
  - Further reduces workload for professors and TAs

Economic Solution

- Current solutions require payment from either the University or the students
  - The university would rather spend this money elsewhere
  - Adds added frustration and financial burden on students
  - Examples: Zybooks, Quizlet, TopHat
- Our PrairieLearn Solution would be free and open source
  - Courses can be created and personalized
  - Only requires that the department has people to work on it
    - Fair wages < buying commercial product

Technical Justification

- Design is very software-focused
  - A lot of tools we’ve never used before
- PrairieLearn is very complex
- Use of a VM
  - Manage and operate software in a virtual environment
- Custom emulation tools and autograder containers
  - Hardware simulation and automated testing
- Variety of programming languages
  - Python, C, HTML
  - Demonstrates our versatility to develop software in many languages.

Conclusion

- Our PrairieLearn solution addresses our users needs
  - Gives students a free and useful study tool
  - Relieves the grading burden from TA’s and Professor’s
- Our solution economically feasible
  - Free and open source for the university
- Our project is technically complex
  - Many different softwares, tools, and languages being used

Any Questions or Suggestions?