**Motivation:**
Demonstration is an important part of teaching people how to program for the first time. In a lecture environment, the two ways to demonstrate coding is either by typing code on a computer or by handwriting code on a board. Typing code ties the lecturer to their laptop. This is problematic because it reduces lecturer mobility, reduces eye contact and engagement with the class, and makes it difficult and unnatural for the lecturer to point out things in the code (since the projection is high above the lecturer). Handwriting code on the board solves all of these problems. However, it also introduces another one: the code can not actually run. Often times, it is useful to show the end result or to step through a program (debugging).

**Features:**
I aim to develop a system that will allow a lecturer to handwrite C code and run code all without touching the keyboard.

If time allows, I also aim to implement setting breakpoints and stepping through code.

**Implementation:**
The Codeable program will be running on a laptop equipped with a camera. Codeable will display the results of running the program or compilation errors on the laptop screen or a projector.

Codeable requires reserved areas on the board for the “run” button and “debug” “step” buttons. To run code, the lecturer writes an “X” on the run button. At this point, Codeable will use OCR to generate source code, which it then compiles and runs. If the compilation is successful, Codeable will execute the program, otherwise it will show an error.

First, Codeable will look for a large white rectangle or trapezoid. It will ignore everything outside this perimeter.

Second, Codeable needs to distinguish between the lecturer and the board. This will likely involve thresholding and region analysis to identify and remove the lecturer.

Third: Codeable needs to recognize the “run” button and whether it has an “X” in it.

Fourth: Codeable needs to binarize the board and perform OCR to read the text. I hope to use Google's “tesseract” OCR engine to help implement this. Since C has semantic rules and keywords, it is possible to use this to improve OCR accuracy. In addition, since variable names often appear more than once, Codeable can also use this to improve accuracy.

Fifth: Codeable needs to hand off the source code the the compiler. If there are errors, display them. Otherwise, run the program and display the output.
References:

