

Distraction Free Coding

Team Focus Finders

Holly Brosnahan (hlb)

Amalya Henderson (amalyah)

Michael Richardson (mrichar2)

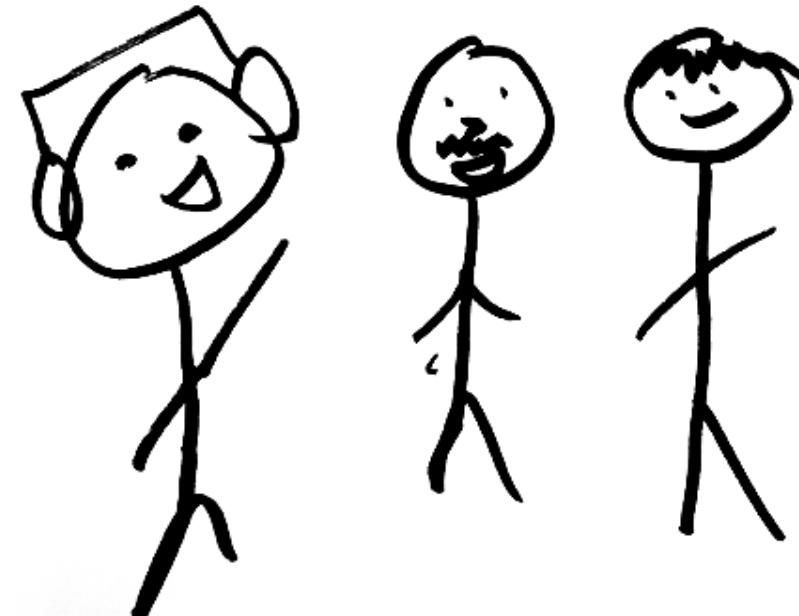
Kaitlyn Sparks (ksparks)

Yueying Tang (yueying)



Outline

- ❖ The Problem
- ❖ The Solution
- ❖ The Vision
- ❖ The Benefits
- ❖ The Risks

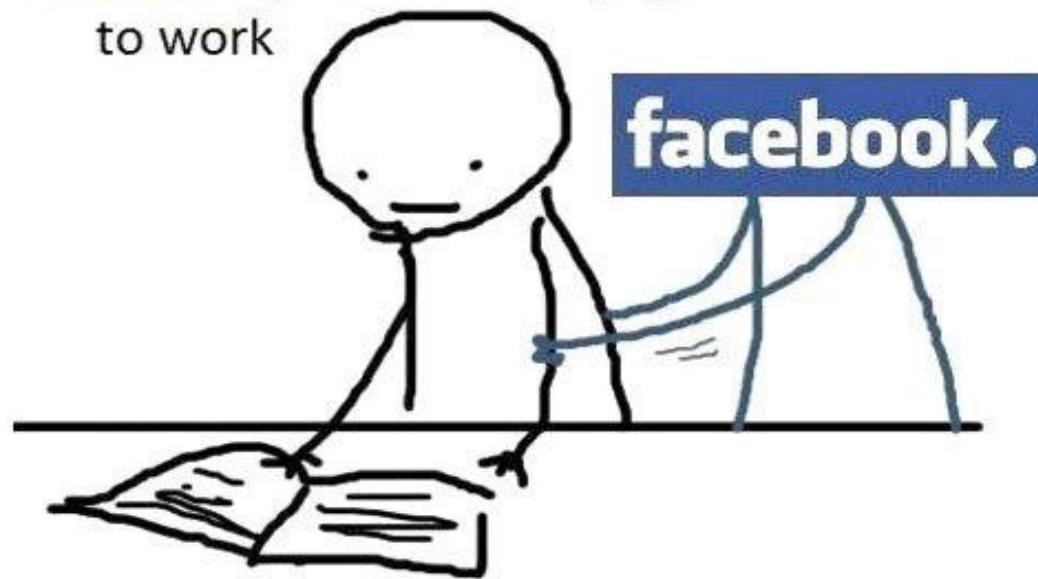


The Problem

People get easily distracted when they are trying to code.

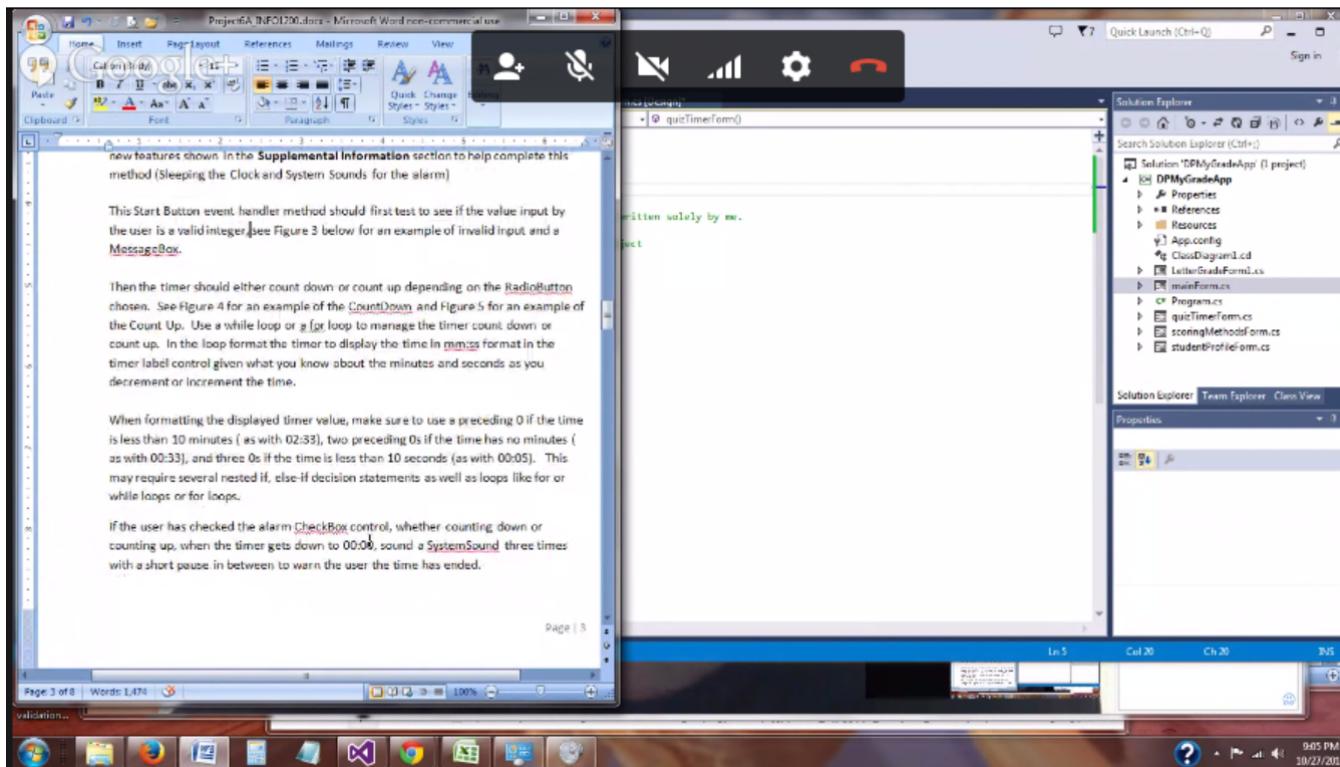
check me, check meehee

Facebook please i am trying
to work



The Problem

Need to switch between many different tools and resources to complete tasks.



The Solution

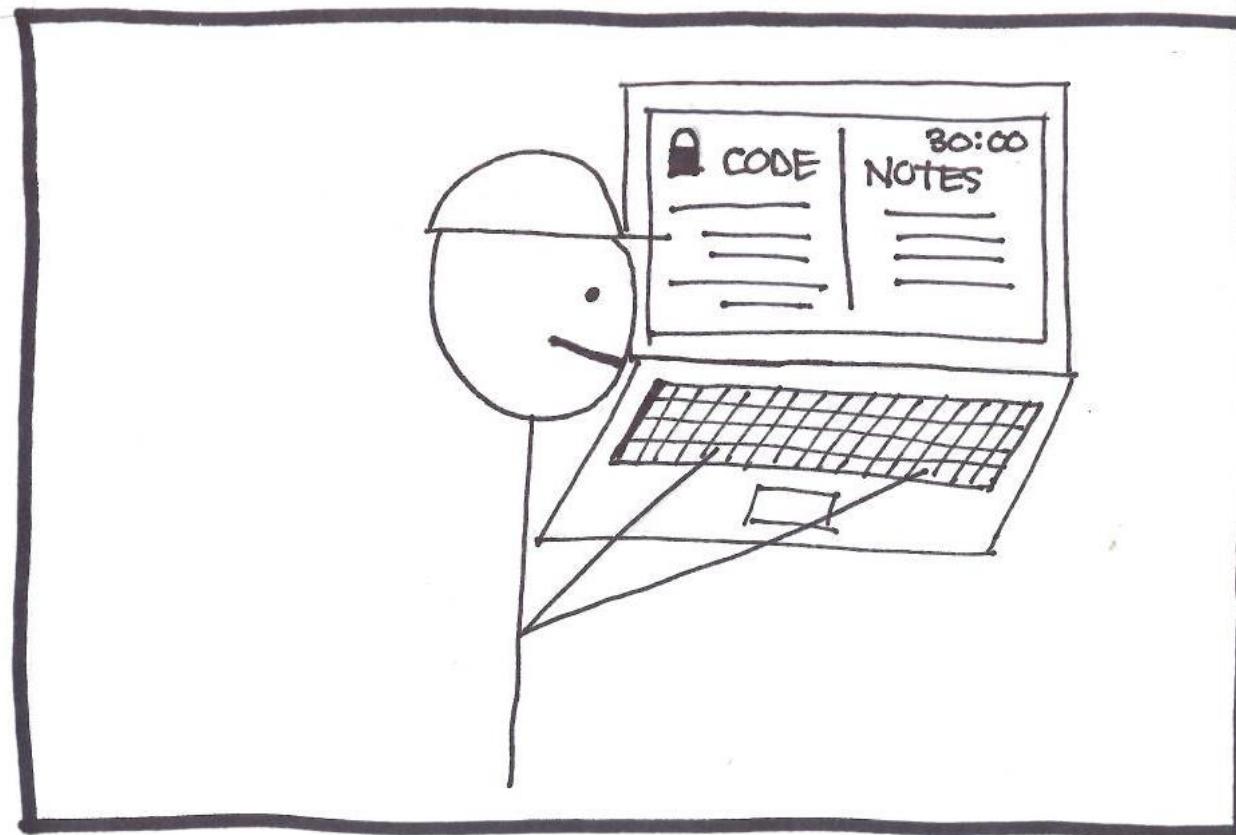
Distraction free coding environment that includes:

- locked environment that blocks extraneous messages
- scheduled breaks
- recent location highlighting
- in program annotations (visual and verbal)



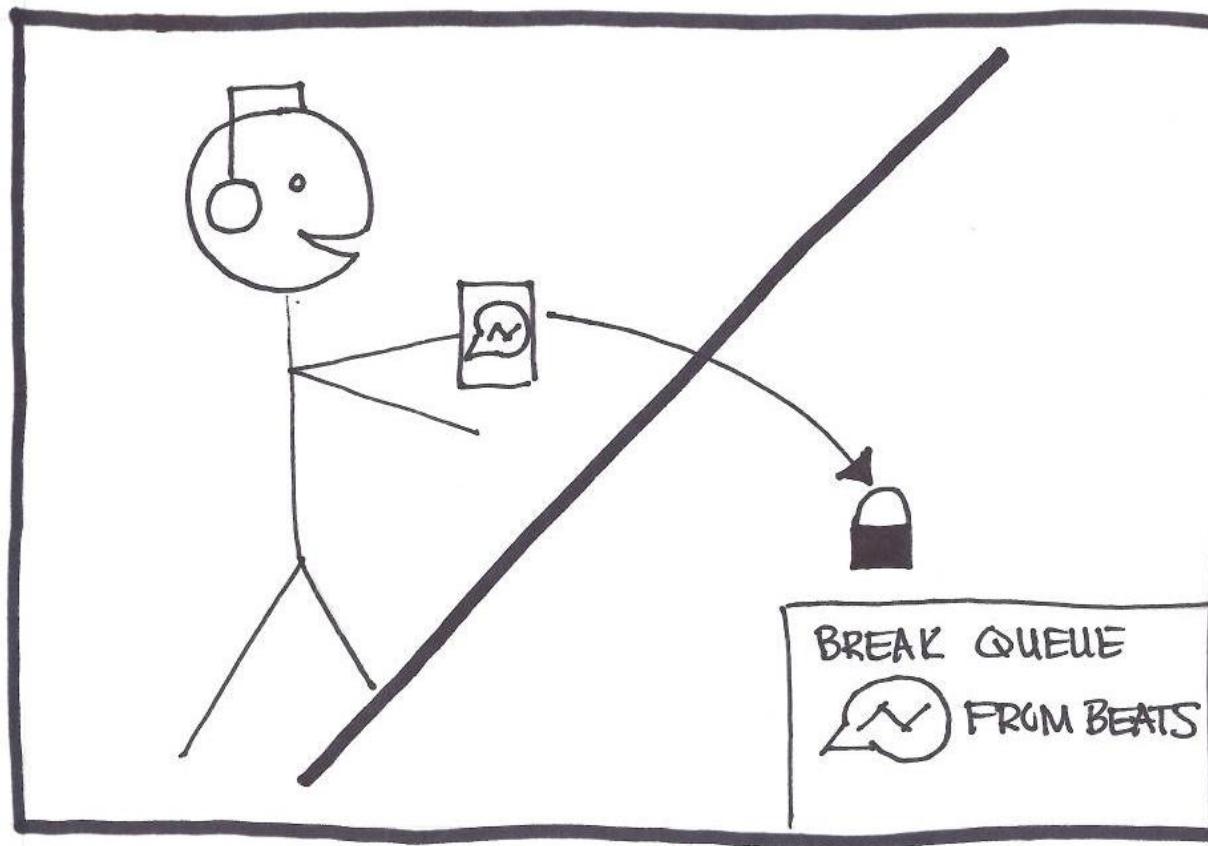
The Vision

Open code environment with distraction blocker, notes, and annotation space



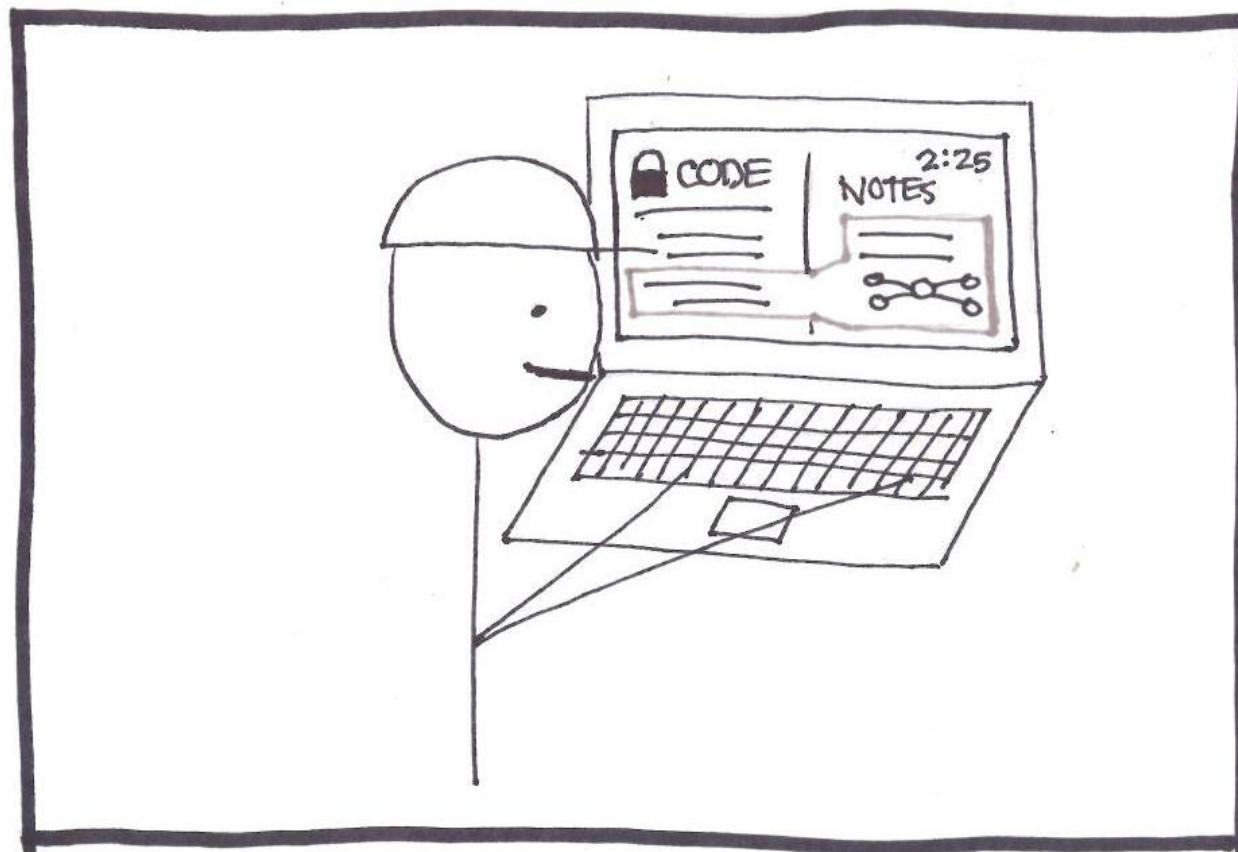
The Vision

Block extraneous messages and save them into msg queue



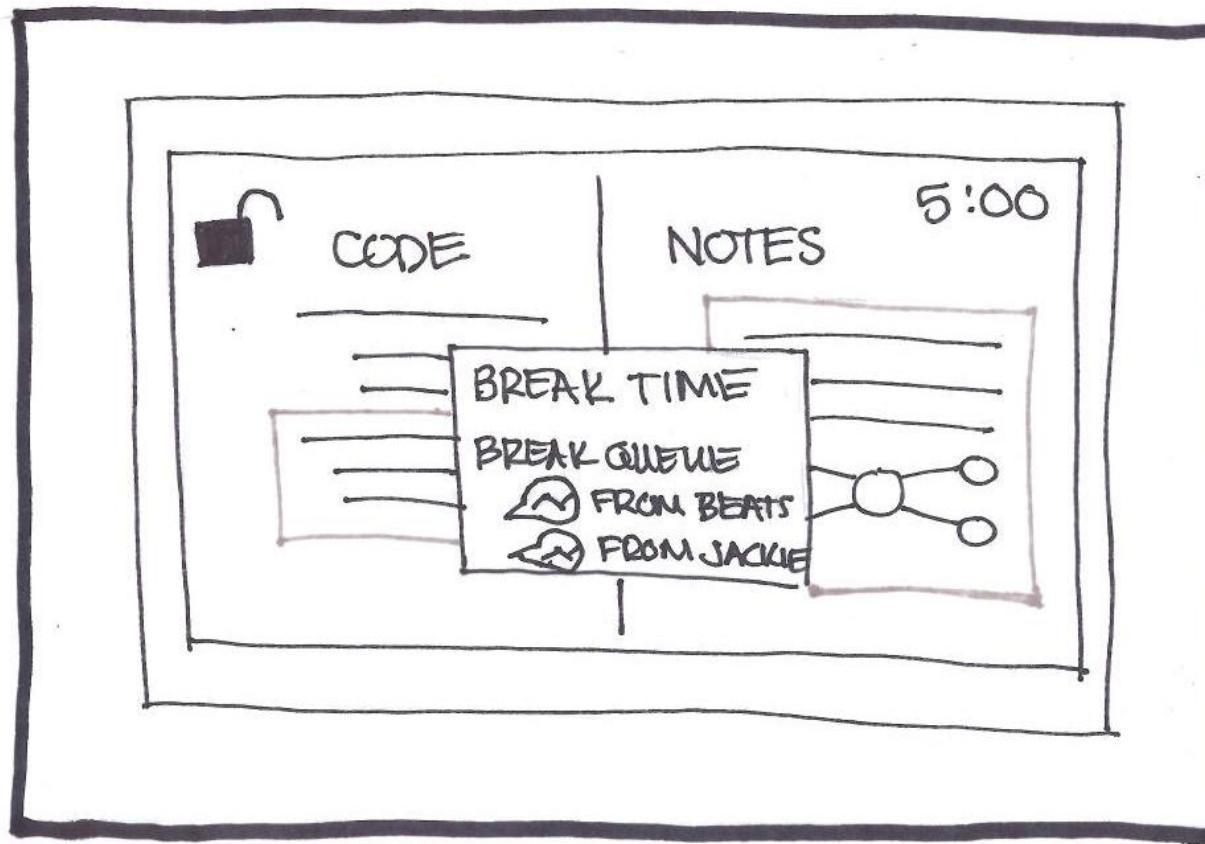
The Vision

Highlight Code and Add Notes Associated With It



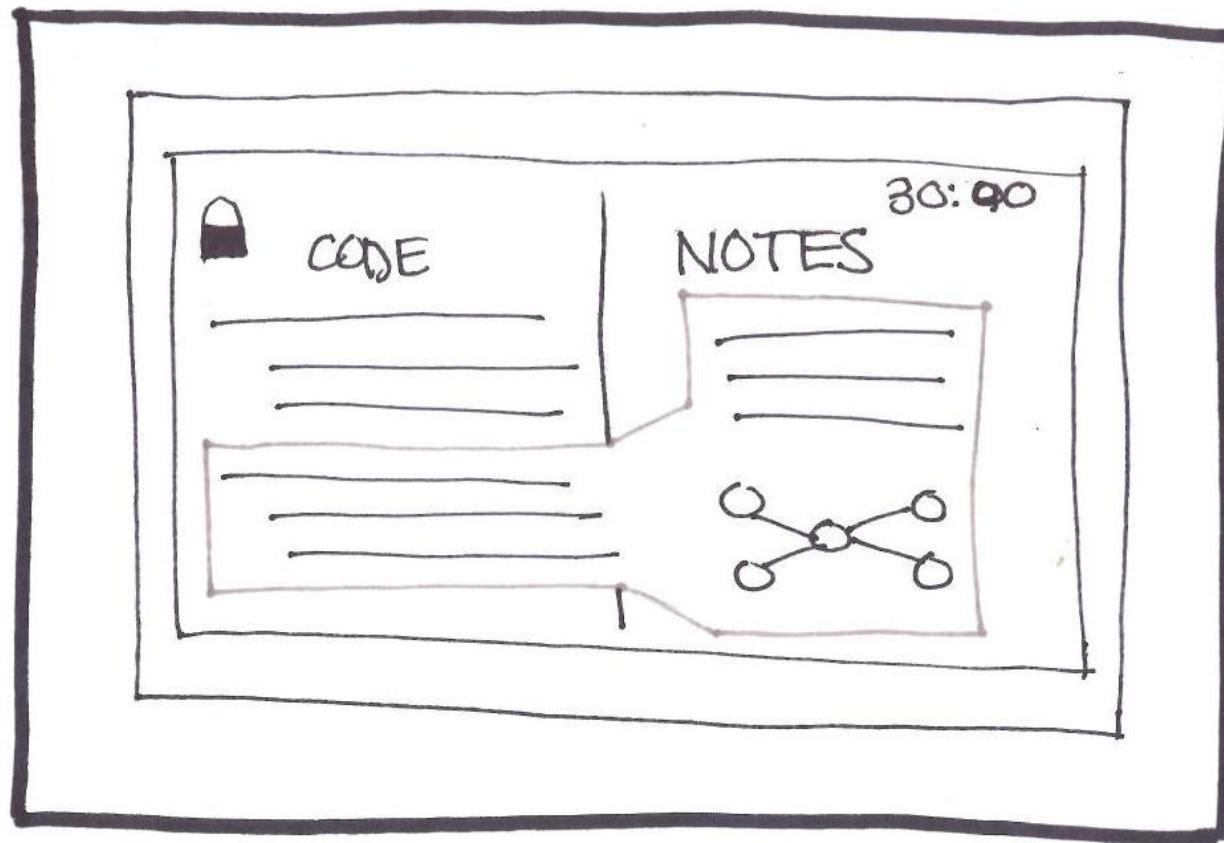
The Vision

During Break Time Queued Messages are Displayed



The Vision

After Break, Highlight the Location of Last Work



The Benefits

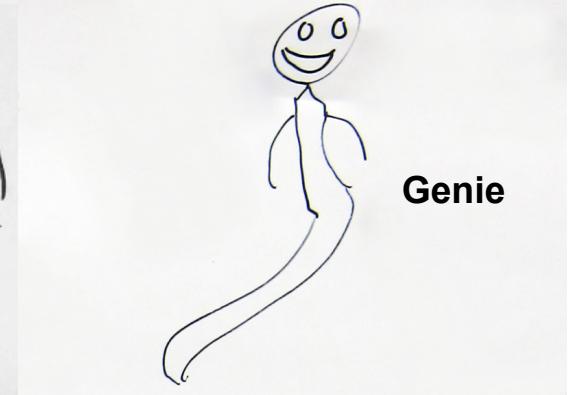
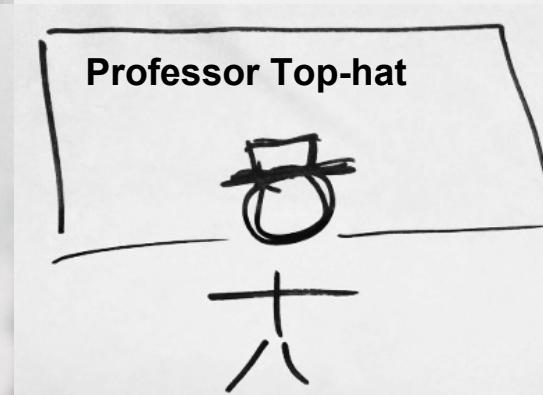
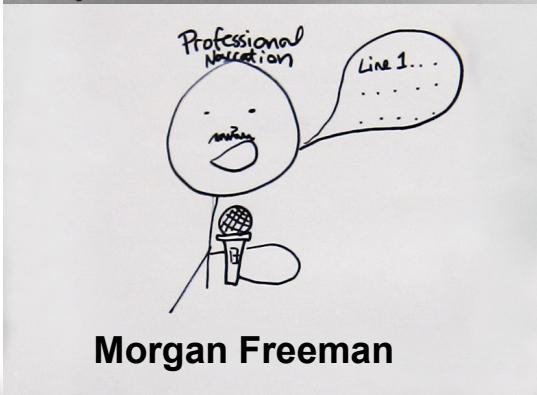
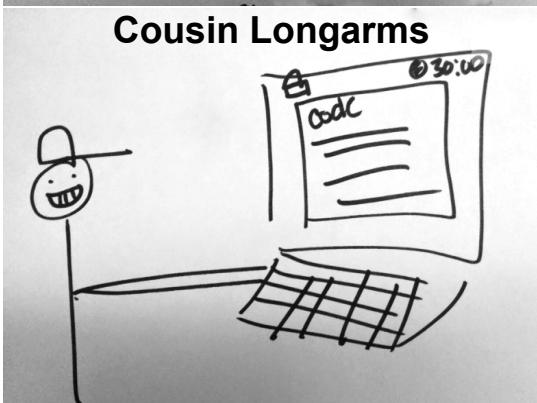
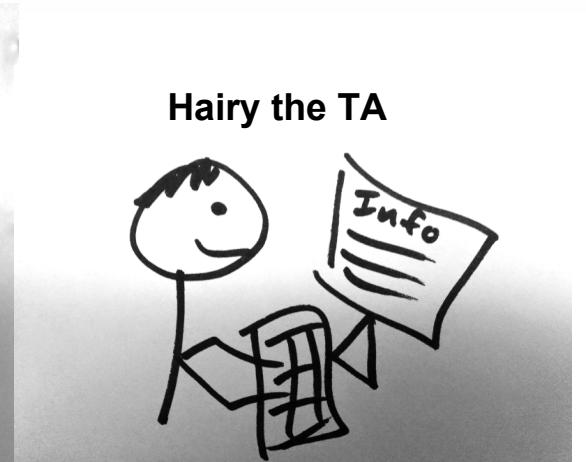
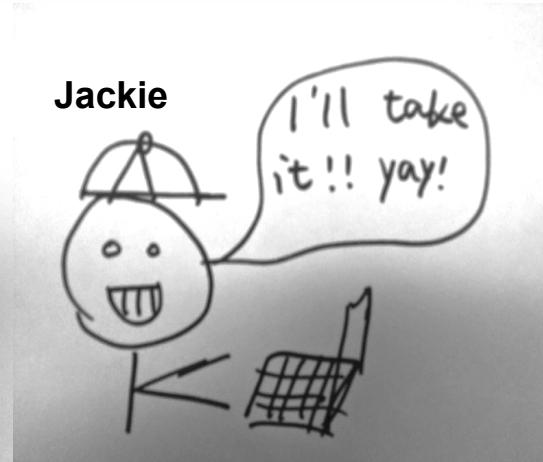
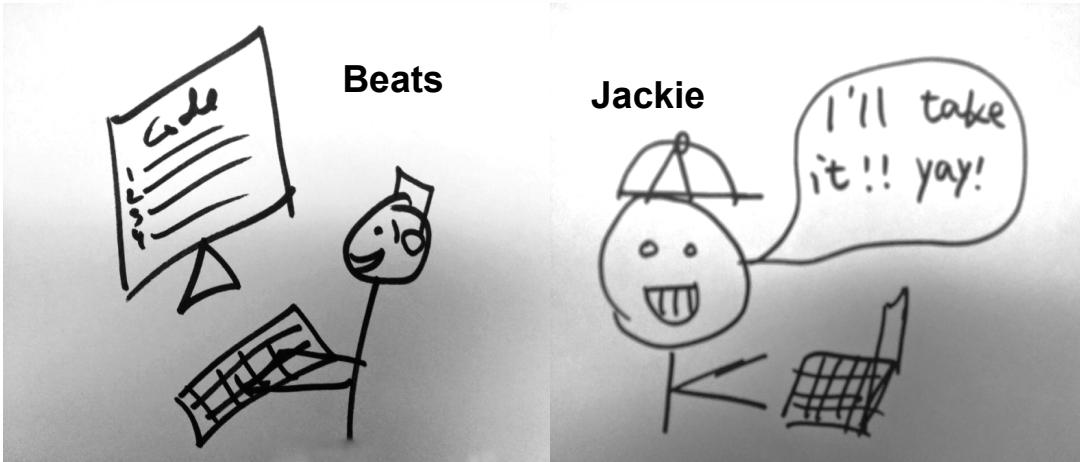
- Delays distractions until it's break time
- Provides strong affordances to maintain and resume focus
- Integrates thinking methods with programming task



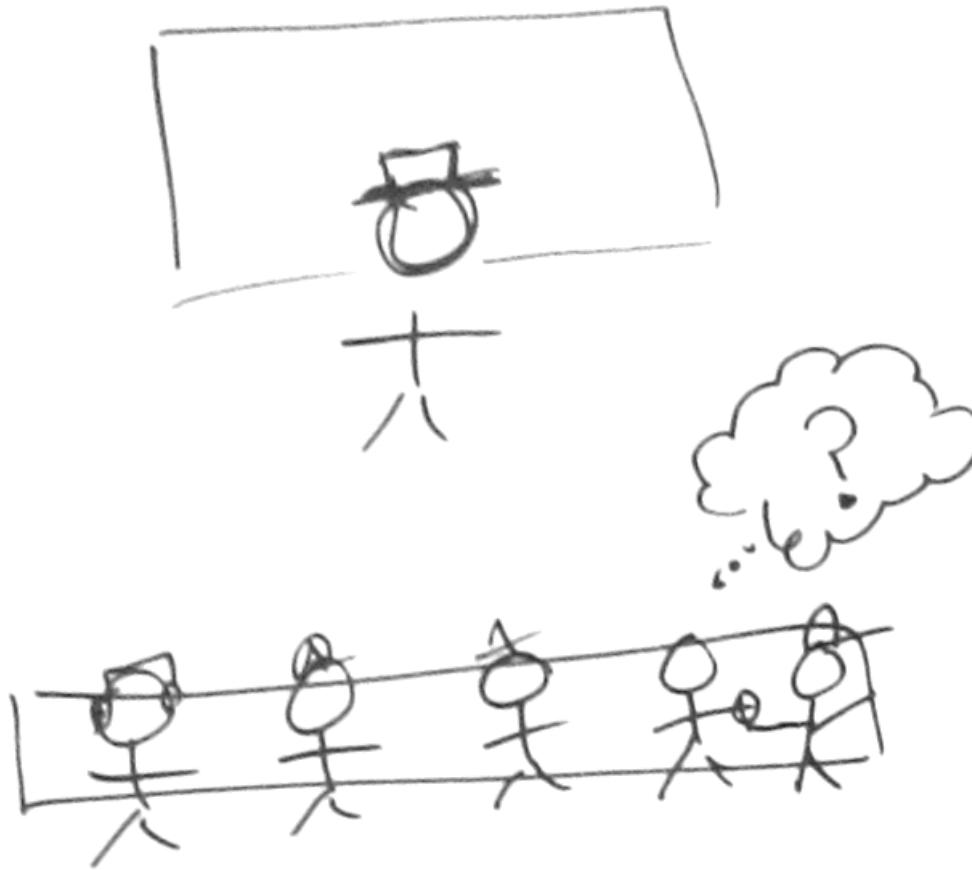
The Risks

- Students may be blocked from social support and relevant discussion
- Timed breaks may not occur at natural breaking points
- The distraction lock may need to open early for special cases
- The resources students need may be blocked





Questions?



Supporting Evidence



I try to keep myself focused.

Clean code helps me keep myself organized.

When I get stuck on a hard problem, I switch to an easier one.

I take breaks to be more productive

I find it difficult to stay focused when doing my assignment



Clean code
helps me keep
myself organized.

- U2-37 DI: Intelli-sense for style

U2-22 Considers style points
for the assignment during his
code

- U2-38 DI: ability to auto clean
up line breaks

U4-41 Cleared out print statements at
the end to make the code less cluttered
and more legible.

U2-26 Cleans up style in
provided code before starting

- U2-27 Breakdown: needs to
reformat style before starting
code on provided file

U5-12 When opened code he was
previously working on, first went
through code he had already
written to try to keep the same
naming convention and to be
consistent programming and use
routines without looking them up.

U3-40 Removes some of the print
statements that are now unnecessary.
He was thoroughly convinced that the
related code works.

When I get
stuck on a
hard problem,
I switch to
an easier one.

U5-19 When he didn't understand
how one function worked, went to
the parts of assignment that he
thought were easy.

U5-29 Low tolerance for frustration
with hard problems, so looked for
easy ones to solve.

U5-23 Works on other tasks to avoid
large task that he doesn't
understand (thread.sleep)

U5-24 BD: He couldn't quickly find
what he was looking for in the book
(thread.sleep), so he moved to
working on a different problem.

U5-15 Found a term that he didn't
understand (thread.sleep) in the
assignment, but didn't look it up
because he thought it might be
described later in the assignment.

U5-31 Worked on small easy
problems for a sense of progress.

U5-38 DI: Break complex tasks in
assignment down into smaller,
simpler steps to feel a small sense
of accomplishment.

U5-17 Made small tweaks to location
of items in UI builder because
professor said that students should
be creative for this assignment.

I take
breaks to
be more
productive

U4-43 Insight: taking a break allowed
him to restart and solve the problem
when he was getting stuck before the
break

U4-42 Takes break when he can't
figure things out

U4-46 BD: wants to take a break and
check email, but phone didn't work

U4-34 Insight: uses breaks as rewards
for completing small sections of his
code

U4-53 Takes breaks between coding
tasks to check his phone for messages

I find it
difficult to
stay focused
when doing
my assignment

U5-9 BD: Interruption when his son
was screaming about not wanting to
go to bed.

- U5-40 DI: Environment that
eliminates distractions (noise
canceling headphones, etc.)

U4-51 BD: turned on music because he
didn't think he needed much brain
power. Had to close it because he
needed to think more

U4-52 listens to music while less
because he thinks he doesn't need
much brain power to do this part of the
assignment

- U5-28 DI: Affordances to keep
person focused important. It takes
too much mental effort to focus
again.

- U4-57 BD: Gets interrupted by
Facebook message from friend

U1-45 Interruption: Listens to what
others are asking the TA



I use a variety
of techniques to
break down and
understand
difficult concepts

I don't
always get
a complete
grasp of
concepts

I ~~find~~ use scratch
paper to think
through problems.

I find drawing
and diagramming
useful to help me
understand complex
concepts

I read assignment
out loud to keep
~~focused~~ and
have a better
understanding



I don't always get a complete grasp of concepts

U5-18 • BD: He didn't understand how to use a certain part of code (thread.sleep).

U4-41 • BU: code was too hard for him to figure out

U1-47 Lots of classes makes him distracted so he only gets a shallow understanding of things instead of reading the text to get a deep understanding

U1-36 User doesn't understand all of the terminology that the TA uses

U1-41 Concepts are easy to talk through with TA but are harder to do in code

U2-17 Uses Google as a resource to find more information about topics

I ~~find~~ use scratch paper to think through problems.

U4-40 Insight: writes on paper to work through details after he tries and fails to code it

U1-50 Writes physical notes while reading textbook to make sure he understands the concepts

U1-33 Writes down things the TA says in a physical notebook because it is faster

U4-8 Uses scratch paper to help think through approach to assignment

U4-7 writes numbers and calculations to think through problem.

U1-43b

writes out notes on paper to try to understand concepts

I find drawing and diagramming useful to help me understand complex concepts

U3-23 Diagrams are for better understanding what the code is doing with child and parent processes.

3-22 • DI: Add an integrated drawing tool

U3-20 Uses Gimp for drawing diagrams. He doesn't have paper readily accessible.

26 Uses diagram as an overall plan. Checks back to see what he needs to do next.

U1-34 TA uses white board to explain things to student

U1-35 TA draws pictures to explain concepts

I read assignment out loud to keep focused and have a better understanding

U5-4 Before starting, the student read the assignment out loud because he said that learns better via auditory methods.

• U5-35 DI: Audio of assignment created by computer.

U5-8 For a concept that he didn't know, reread it out loud.

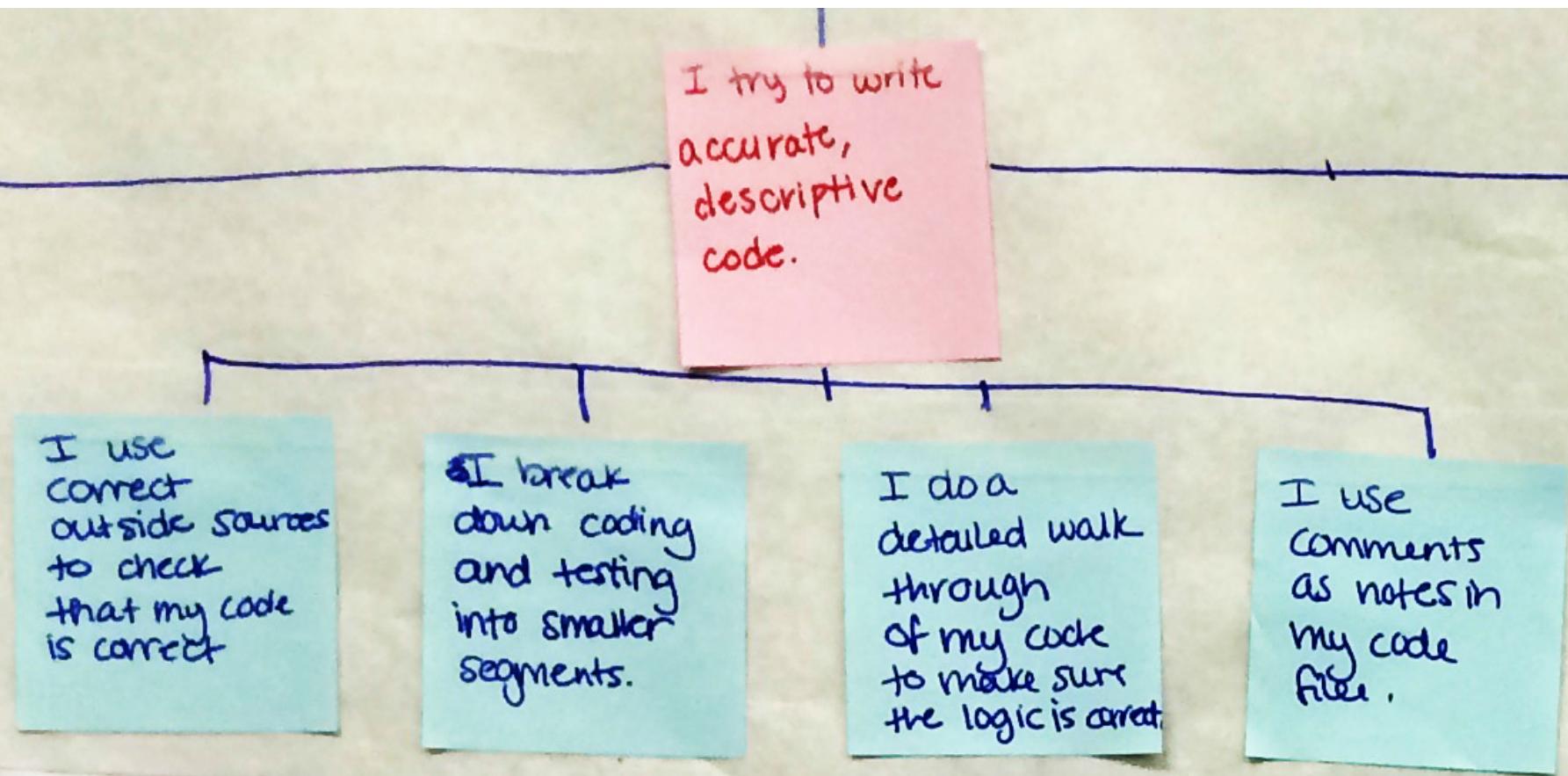
U5-6 Selected part of assignment working on, and moved his cursor across the text while he read it out loud.

• U5-34 DI: Audio of assignment recorded by professor.

U5-36 DI: Way for student to record • himself reading the assignment where he could play parts of it.

U5-7 Rereads part of text out load over again that is related to the rubric of what is graded.





I use correct outside sources to check that my code is correct

U4-11 Required calculator to test assignment

U4-27 Compares results from code to results from calculator to determine correctness

U2-24 After compiling, uses provided examples to test his code

U4-59 Used wolfram alpha calculator that can do multiple matrix calculations

U4-10 Q: Did teacher teach the process through writing down number and calculations in class?
U4-11 Used online calculator from Google

U2-28 In HW file, a trace table is provided to give an example of what each file should do

U1-20 Reference handout for how to test

U1-5 Everything is auto-graded through Autolab

I break down coding and testing into smaller segments.

U5-30 Worked on hard problems in segments to minimize frustration.

US-32 Student broke apart assignment into meaningful chunks to work on.

U4-20 Breaks tests into smaller sub tests at the beginning

J3-24 From diagrams, he decides to divide the task into 2 halves that he can do one at a time.

U4-30 When he can't get code to work with full tests, splits into smaller tests and tries to make those work

U4-21 tests broken down with multiple cases per code function

U2-18 Student breaks down tasks by the traces provided by the instructor

34 Uncomments out the functions that he saved for later when he's ready for the second half of the task.

I do a detailed walk through of my code to make sure the logic is correct.

J3-33 Reads through all comments towards the end as a sanity check to make sure that the logic is correct.

3-35 Program seems to be working, but needs more detailed information.
Programs more specific print statements.

U4-31 DI: Allow user to walk through code line by line to find error

I use comments as notes in my code file.

U1-42 Code contains comments from the professor to give students and indication of what they should be doing in their code

3-10 Uses comments to remind him what the relevant/related functions are from the man pages.

-28 DI: Create an annotation system that goes side by side with the code

U4-13 Uses comments in his test case code to specify an example correct functionality.

U1-43 Gets a concept out of the comments in the code that he had written.

U5-13-B

Dated
with date assignment was due to remember how much time he has left.



Intent:
Understand
Coding
Concept

Think through
code by writing.

U4.1.17 U4.1.7
U4.2.15 U4.1.71
U4.1.24
U4.1.49
U4.1.73

Think through
code by
diagramming.

U2.1.36
U3.1.94-36
U3.1.71
U3.1.55

Think through
code.

U4.2.7 U5.1.33
U4.2.9
U4.2.11
U4.2.13



