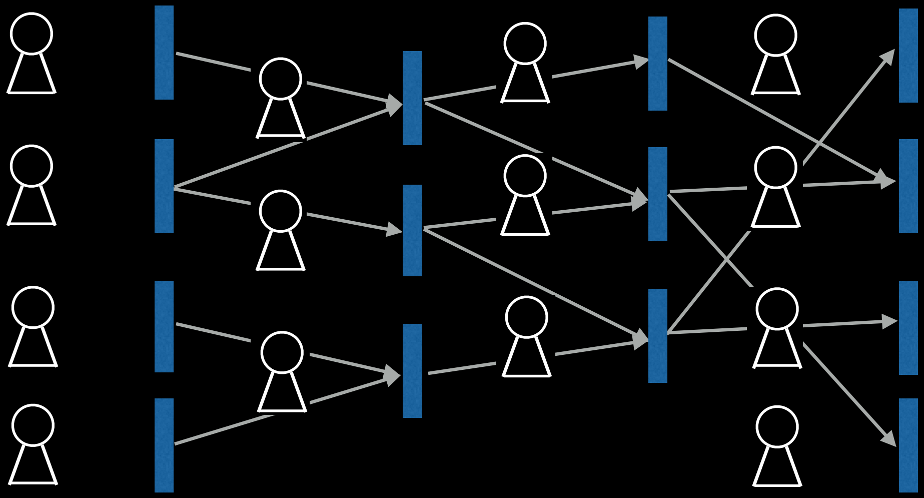
Microtask Programming: Building Software with a Crowd

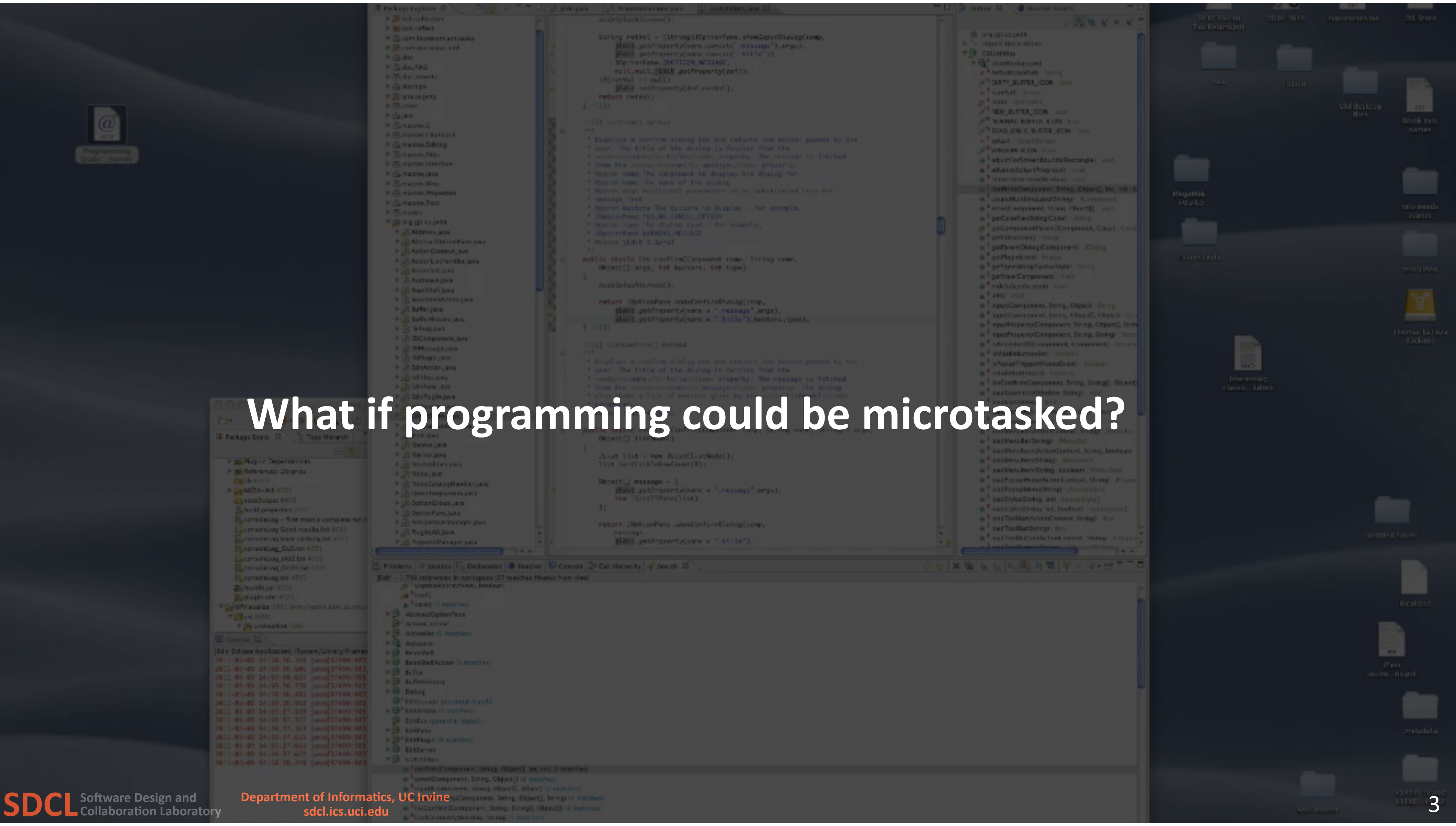
Thomas LaToza[1], Ben Towne[2], Christian Adriano[1], André van der Hoek[1]

[1] University of California, Irvine [2] Carnegie Mellon University

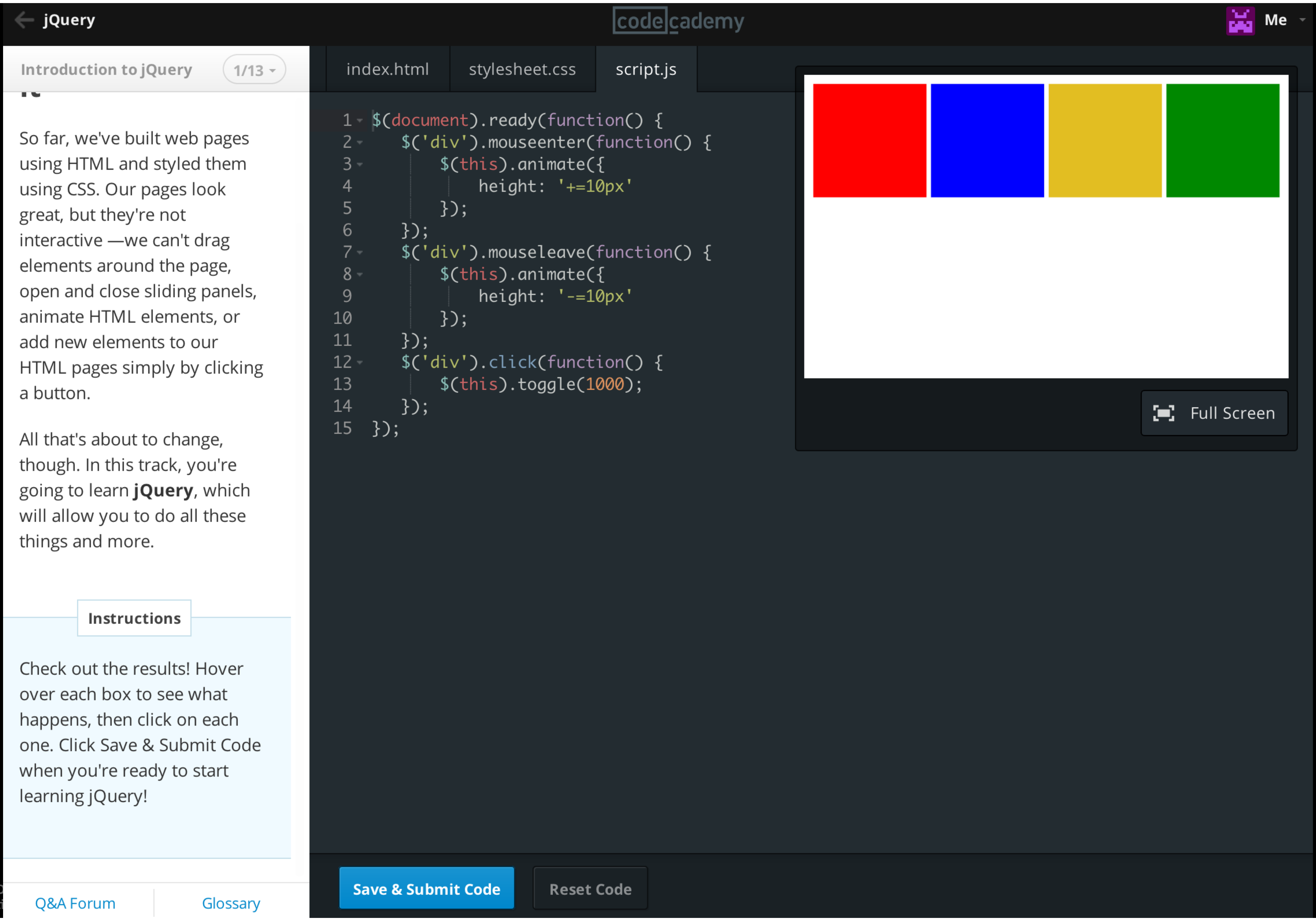
task

microtasks





What if programming could be microtasked?



So far, we've built web pages using HTML and styled them using CSS. Our pages look great, but they're not interactive —we can't drag elements around the page, open and close sliding panels, animate HTML elements, or add new elements to our HTML pages simply by clicking a button.

All that's about to change, though. In this track, you're going to learn **jQuery**, which will allow you to do all these things and more.

Instructions

Check out the results! Hover over each box to see what happens, then click on each one. Click Save & Submit Code when you're ready to start learning jQuery!

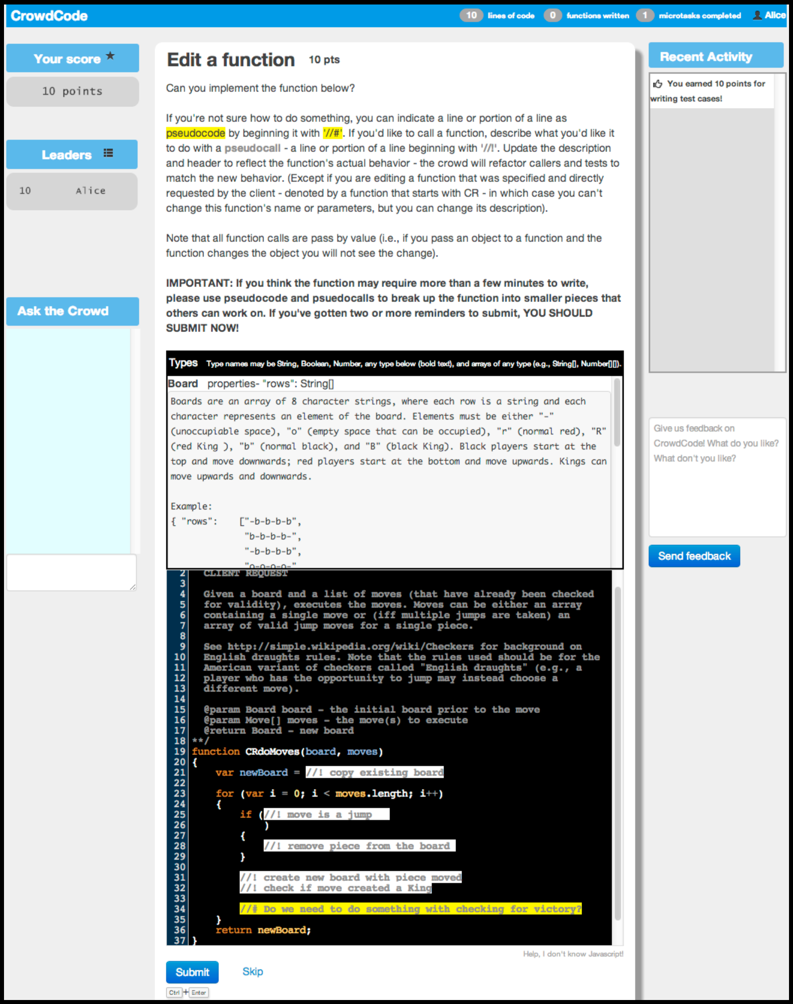
```
$(document).ready(function() {
    $('div').mouseenter(function() {
        $(this).animate({
            height: '+=10px'
        });
    });
    $('div').mouseleave(function() {
        $(this).animate({
            height: '-=10px'
        });
    });
    $('div').click(function() {
        $(this).toggle(1000);
    });
});
```

24 million users x 1 day = ???

codecademy

How can programming work be decomposed into microtasks?

CrowdCode: A System for Microtask Programming



Edit a function 10 pts

Can you implement the function below?

If you're not sure how to do something, you can indicate a line or portion of a line as pseudocode by beginning it with '//#'. If you'd like to call a function, describe what you'd like it to do with a pseudocall - a line or portion of a line beginning with '//!'. Update the description and header to reflect the function's actual behavior - the crowd will refactor callers and tests to match the new behavior. (Except if you are editing a function that was specified and directly requested by the client - denoted by a function that starts with CR - in which case you can't change this function's name or parameters, but you can change its description).

Note that all function calls are pass by value (i.e., if you pass an object to a function and the function changes the object you will not see the change).

IMPORTANT: If you think the function may require more than a few minutes to write, please use pseudocode and psuedocalls to break up the function into smaller pieces that others can work on. If you've gotten two or more reminders to submit, YOU SHOULD SUBMIT NOW!

Types Type names may be String, Boolean, Number, any type below (bold text), and arrays of any type (e.g., String[], Number[][]).

Board properties- "rows": String[]

```
Boards are an array of 8 character strings, where each row is a string and each
character represents an element of the board. Elements must be either "-"
(unoccupiable space), "o" (empty space that can be occupied), "r" (normal red), "R"
(red King ), "b" (normal black), and "B" (black King). Black players start at the
top and move downwards; red players start at the bottom and move upwards. Kings can
move upwards and downwards.

Example:
{ "rows":    ["-b-b-b-b",
              "b-b-b-b-",
              "-b-b-b-b",
```

```
  CLIENT REQUEST

  Given a board and a list of moves (that have already been checked
  for validity), executes the moves. Moves can be either an array
  containing a single move or (iff multiple jumps are taken) an
  array of valid jump moves for a single piece.

  See http://simple.wikipedia.org/wiki/Checkers for background on
  English draughts rules. Note that the rules used should be for the
  American variant of checkers called "English draughts" (e.g., a
  player who has the opportunity to jump may instead choose a
  different move).

  @param Board board - the initial board prior to the move
  @param Move[] moves - the move(s) to execute
  @return Board - new board
**/
function CRdoMoves(board, moves)
{
    var newBoard = //! copy existing board

    for (var i = 0; i < moves.length; i++)
    {
        if (//! move is a jump
            )
        {
            //! remove piece from the board
        }

        //! create new board with piece moved
        //! check if move created a King

        //# Do we need to do something with checking for victory?
    }
    return newBoard;
}
```

You earned 10 points for writing test cases!

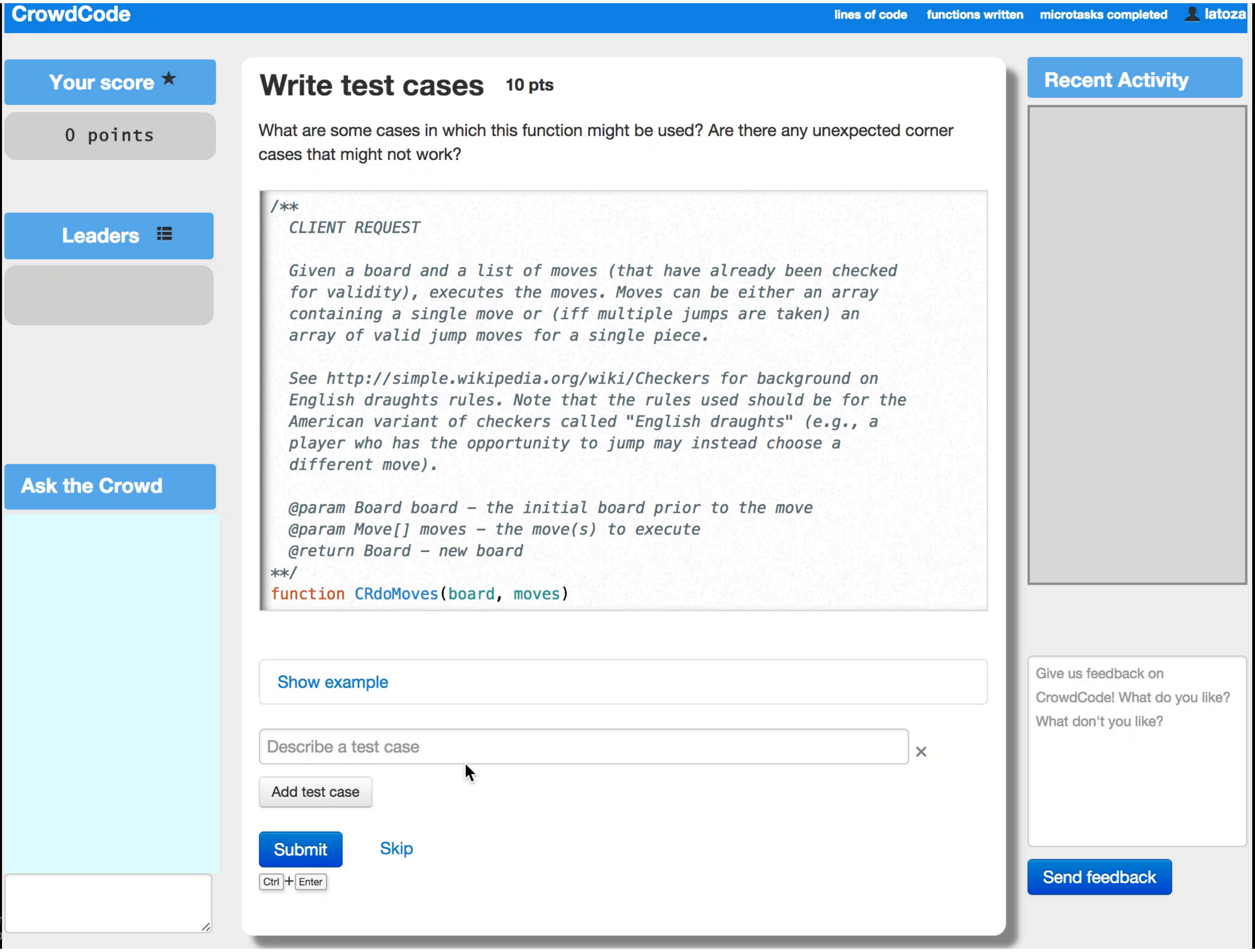
Give us feedback on CrowdCode! What do you like? What don't you like?

All work done in self-contained microtasks, enabling workers to edit only a single a function or test and providing relevant background

Microtasks automatically generated by the system and assigned to workers

Provides to write code, test, debug, and respond to changes

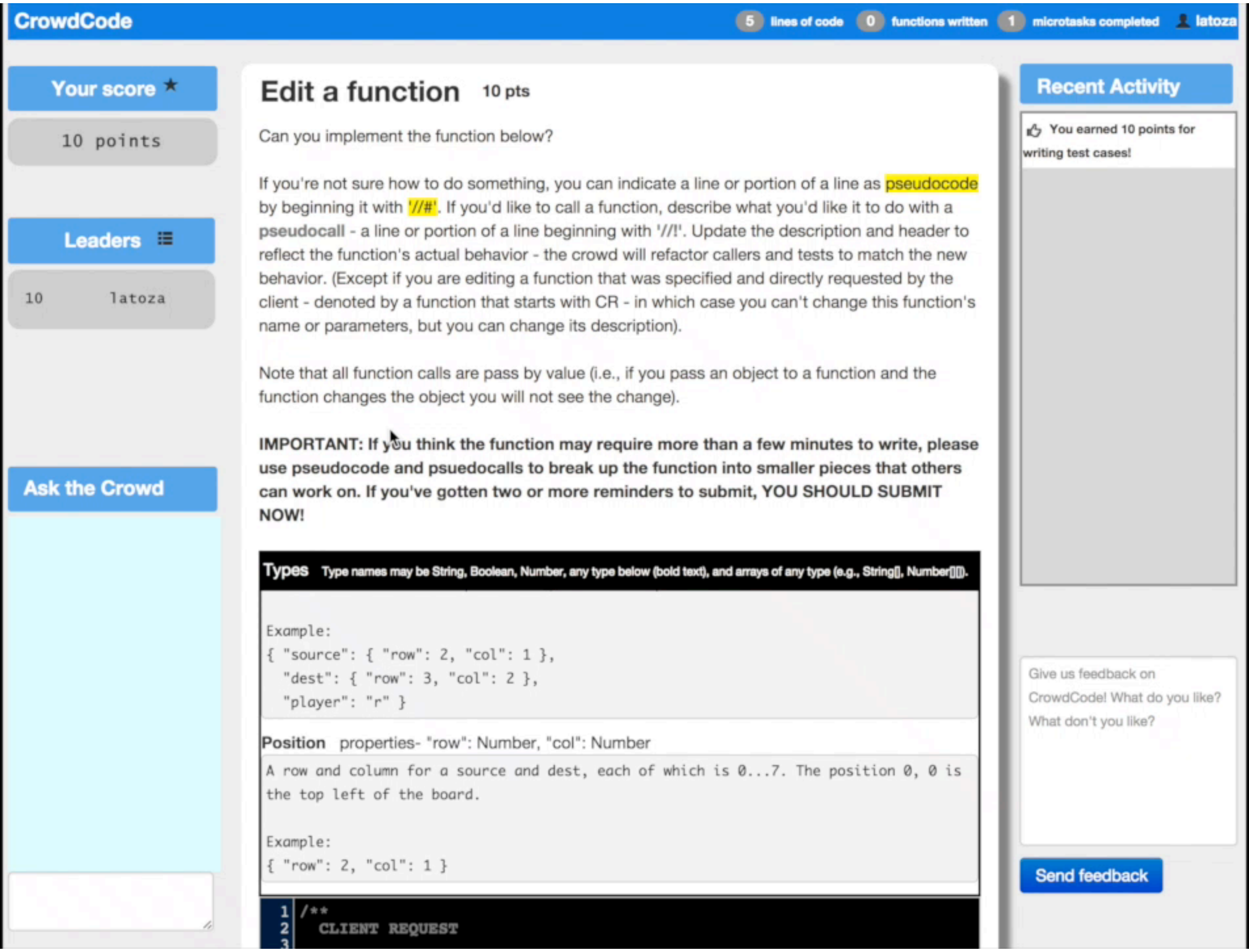
Online IDE for Javascript programming, enabling developers to login and work for 5 mins or 5 hours



Write test cases 10 pts

What are some cases in which this function might be used? Are there any unexpected corner cases that might not work?

```
/**
  CLIENT REQUEST

  Given a board and a list of moves (that have already been checked
  for validity), executes the moves. Moves can be either an array
  containing a single move or (iff multiple jumps are taken) an
  array of valid jump moves for a single piece.

  See http://simple.wikipedia.org/wiki/Checkers for background on
  English draughts rules. Note that the rules used should be for the
  American variant of checkers called "English draughts" (e.g., a
  player who has the opportunity to jump may instead choose a
  different move).

  @param Board board - the initial board prior to the move
  @param Move[] moves - the move(s) to execute
  @return Board - new board
**/
function CRdoMoves(board, moves)
```


CrowdCode

5 lines of code 0 functions written 1 microtasks completed latoza

Your score ★

10 points

Leaders

10	latoza

Ask the Crowd

Edit a function 10 pts

Can you implement the function below?

If you're not sure how to do something, you can indicate a line or portion of a line as pseudocode by beginning it with '//#'. If you'd like to call a function, describe what you'd like it to do with a pseudocall - a line or portion of a line beginning with '//!'. Update the description and header to reflect the function's actual behavior - the crowd will refactor callers and tests to match the new behavior. (Except if you are editing a function that was specified and directly requested by the client - denoted by a function that starts with CR - in which case you can't change this function's name or parameters, but you can change its description).

Note that all function calls are pass by value (i.e., if you pass an object to a function and the function changes the object you will not see the change).

IMPORTANT: If you think the function may require more than a few minutes to write, please use pseudocode and psuedocalls to break up the function into smaller pieces that others can work on. If you've gotten two or more reminders to submit, YOU SHOULD SUBMIT NOW!

Types Type names may be String, Boolean, Number, any type below (bold text), and arrays of any type (e.g., String[], Number[][]).

```
Example:
{ "source": { "row": 2, "col": 1 },
  "dest": { "row": 3, "col": 2 },
  "player": "r" }
```

Position properties- "row": Number, "col": Number

```
A row and column for a source and dest, each of which is 0...7. The position 0, 0 is
the top left of the board.

Example:
{ "row": 2, "col": 1 }
```

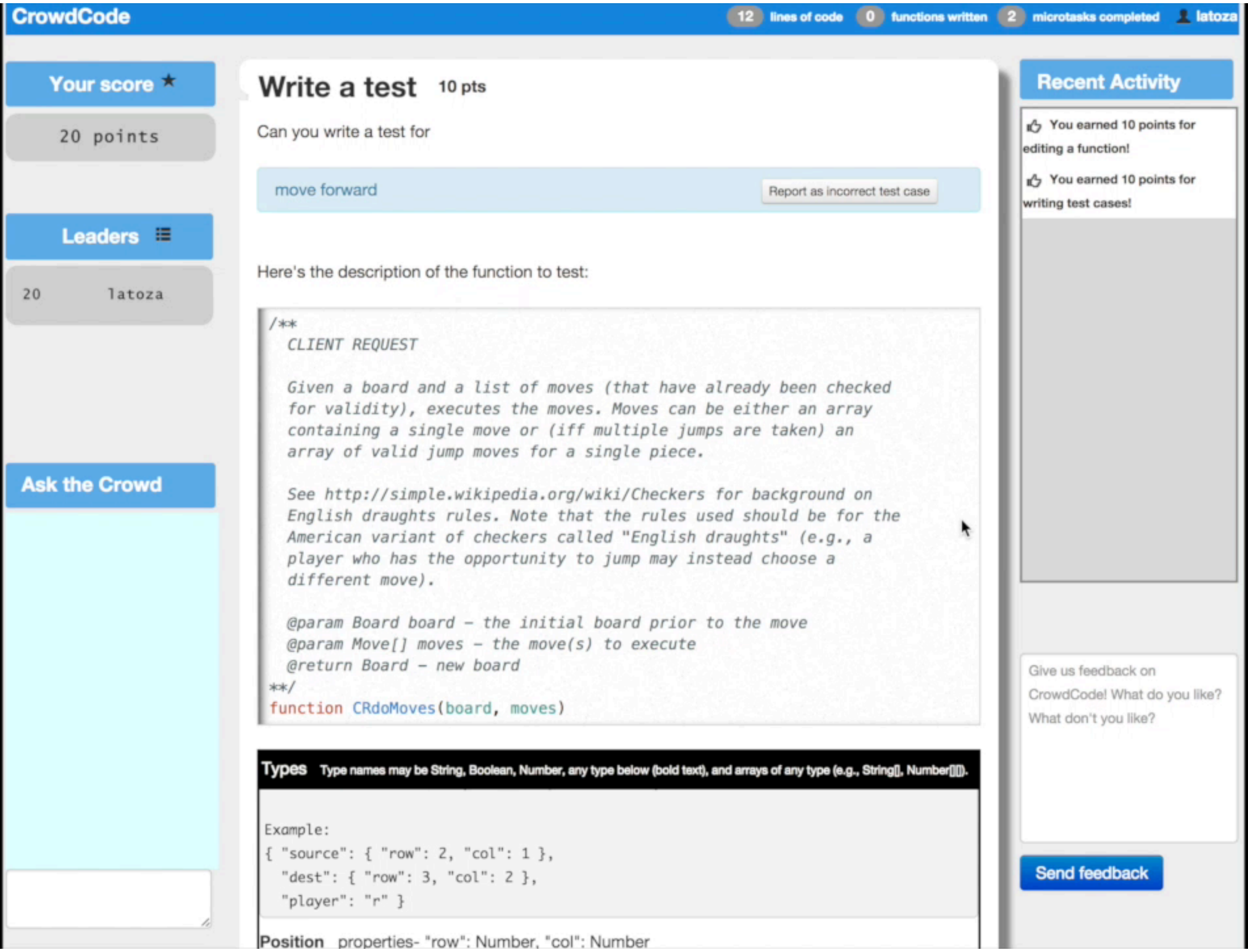
```
/**
  CLIENT REQUEST
```

Recent Activity

You earned 10 points for writing test cases!

Give us feedback on CrowdCode! What do you like? What don't you like?

Send feedback


CrowdCode
12 lines of code 0 functions written 2 microtasks completed latoza

Your score
20 points

Leaders
20 latoza

Ask the Crowd

Write a test 10 pts

Can you write a test for

move forward

Report as incorrect test case

Here's the description of the function to test:

```
/**
  CLIENT REQUEST

  Given a board and a list of moves (that have already been checked
  for validity), executes the moves. Moves can be either an array
  containing a single move or (iff multiple jumps are taken) an
  array of valid jump moves for a single piece.

  See http://simple.wikipedia.org/wiki/Checkers for background on
  English draughts rules. Note that the rules used should be for the
  American variant of checkers called "English draughts" (e.g., a
  player who has the opportunity to jump may instead choose a
  different move).

  @param Board board - the initial board prior to the move
  @param Move[] moves - the move(s) to execute
  @return Board - new board
**/
function CRdoMoves(board, moves)
```

Types Type names may be String, Boolean, Number, any type below (bold text), and arrays of any type (e.g., String[], Number[][]).

```
Example:
{ "source": { "row": 2, "col": 1 },
  "dest": { "row": 3, "col": 2 },
  "player": "r" }
```

Position properties- "row": Number, "col": Number

Recent Activity

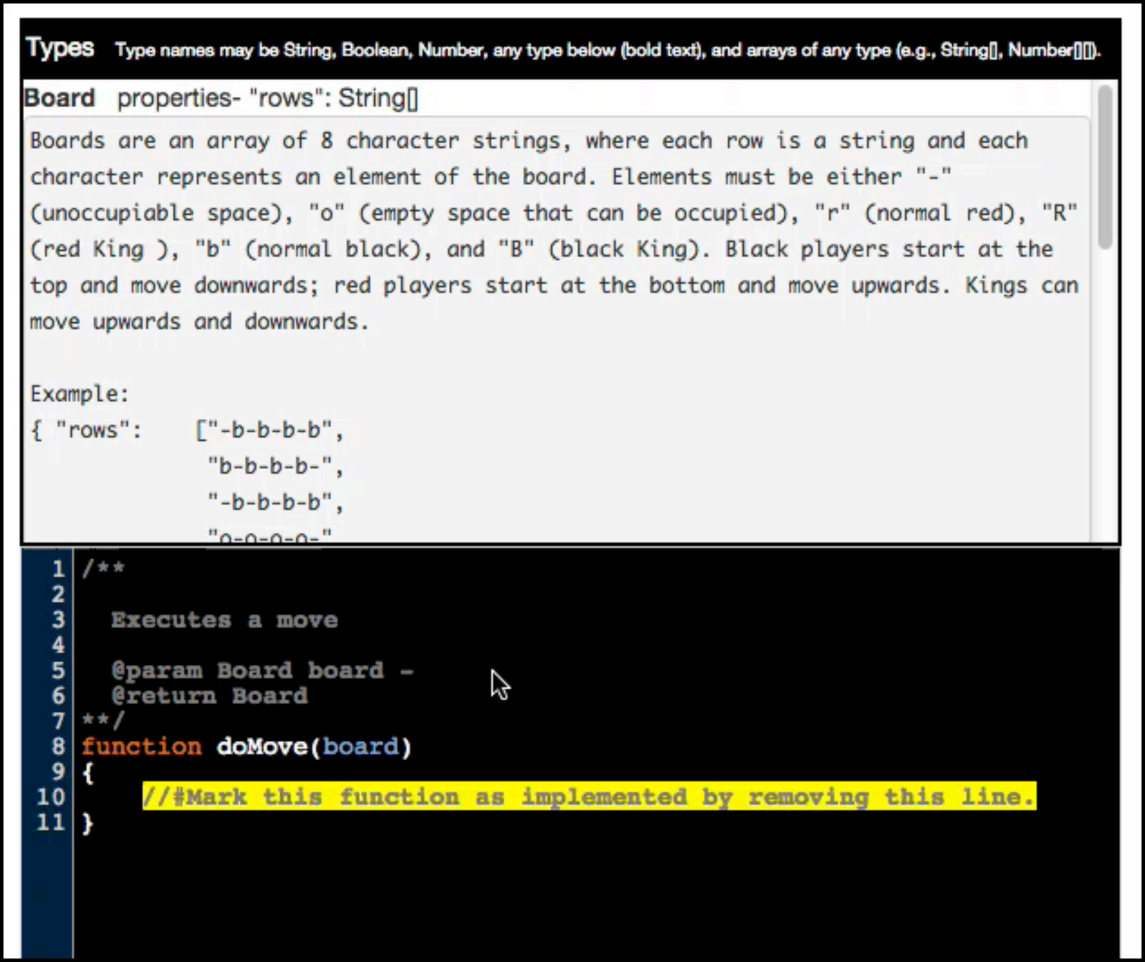
You earned 10 points for editing a function!

You earned 10 points for writing test cases!

Give us feedback on CrowdCode! What do you like? What don't you like?

Send feedback

What if you needed to add a parameter…



Types Type names may be String, Boolean, Number, any type below (bold text), and arrays of any type (e.g., String[], Number[][]).

Board properties- "rows": String[]

Boards are an array of 8 character strings, where each row is a string and each character represents an element of the board. Elements must be either "-" (unoccupiable space), "o" (empty space that can be occupied), "r" (normal red), "R" (red King), "b" (normal black), and "B" (black King). Black players start at the top and move downwards; red players start at the bottom and move upwards. Kings can move upwards and downwards.

```
Example:
{ "rows":    ["-b-b-b-b",
              "b-b-b-b-",
              "-b-b-b-b",
```

```
/**

  Executes a move

  @param Board board -
  @return Board
**/
function doMove(board)
{
    //#Mark this function as implemented by removing this line.
}
```

Programming work is dynamic

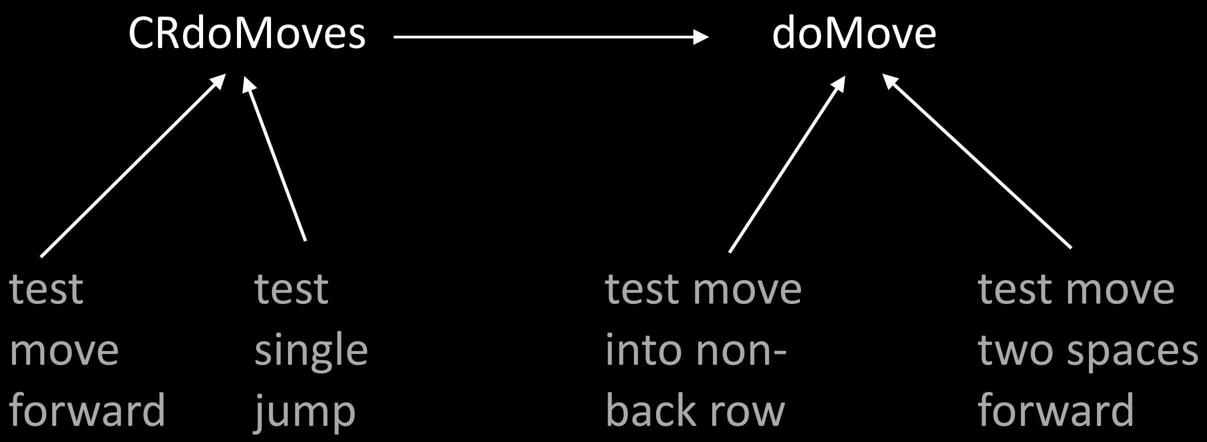
Existing approaches to microtasking complex work rely on a **static** workflow specified by a single requestor or worker
e.g., MapReduce approach in CrowdForge [Kittur+ 2011]

Programming is **dynamic**, cannot enumerate tasks a priori

- Discover need for additional functions
- Need to debug the bugs that emerge when they occur
- Functions may change their signature, necessitating changes to their callers

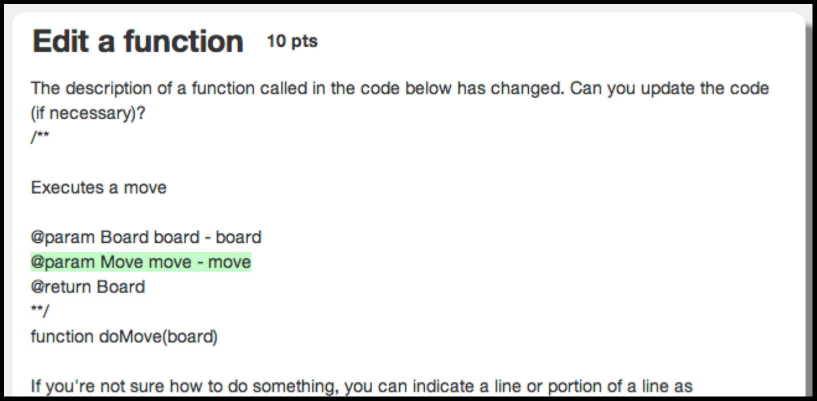
How can microtasks be appropriately generated and coordinated for **dynamic**, complex work?

The dependency structure of software work



Adding a parameter

Signature change microtask

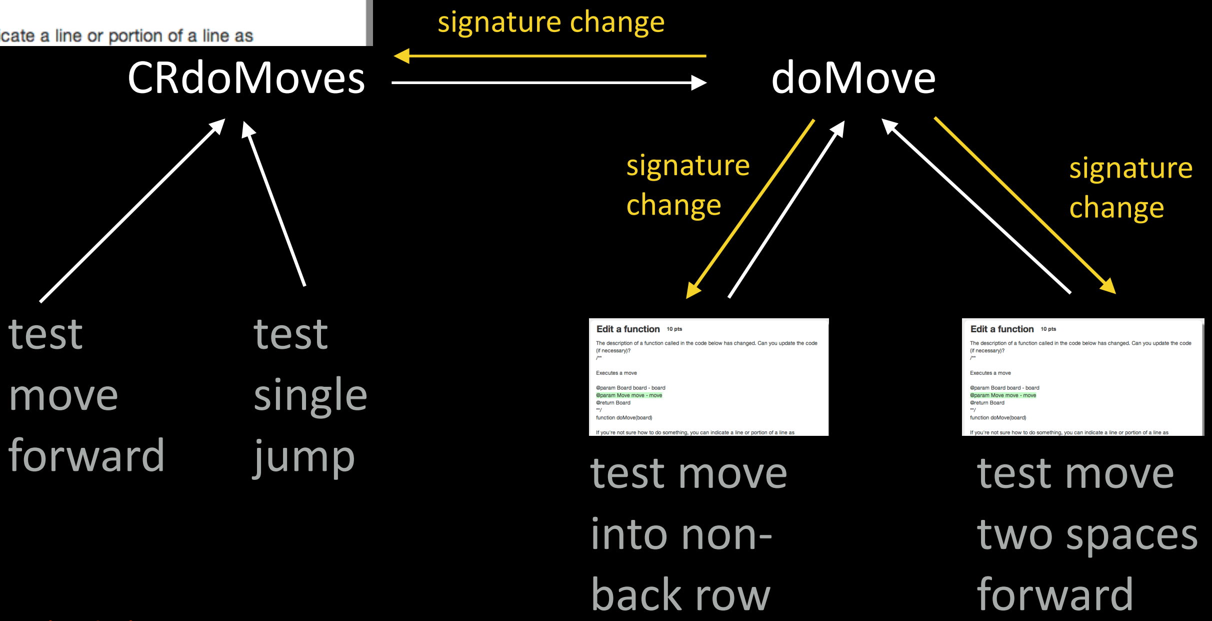


Edit a function 10 pts

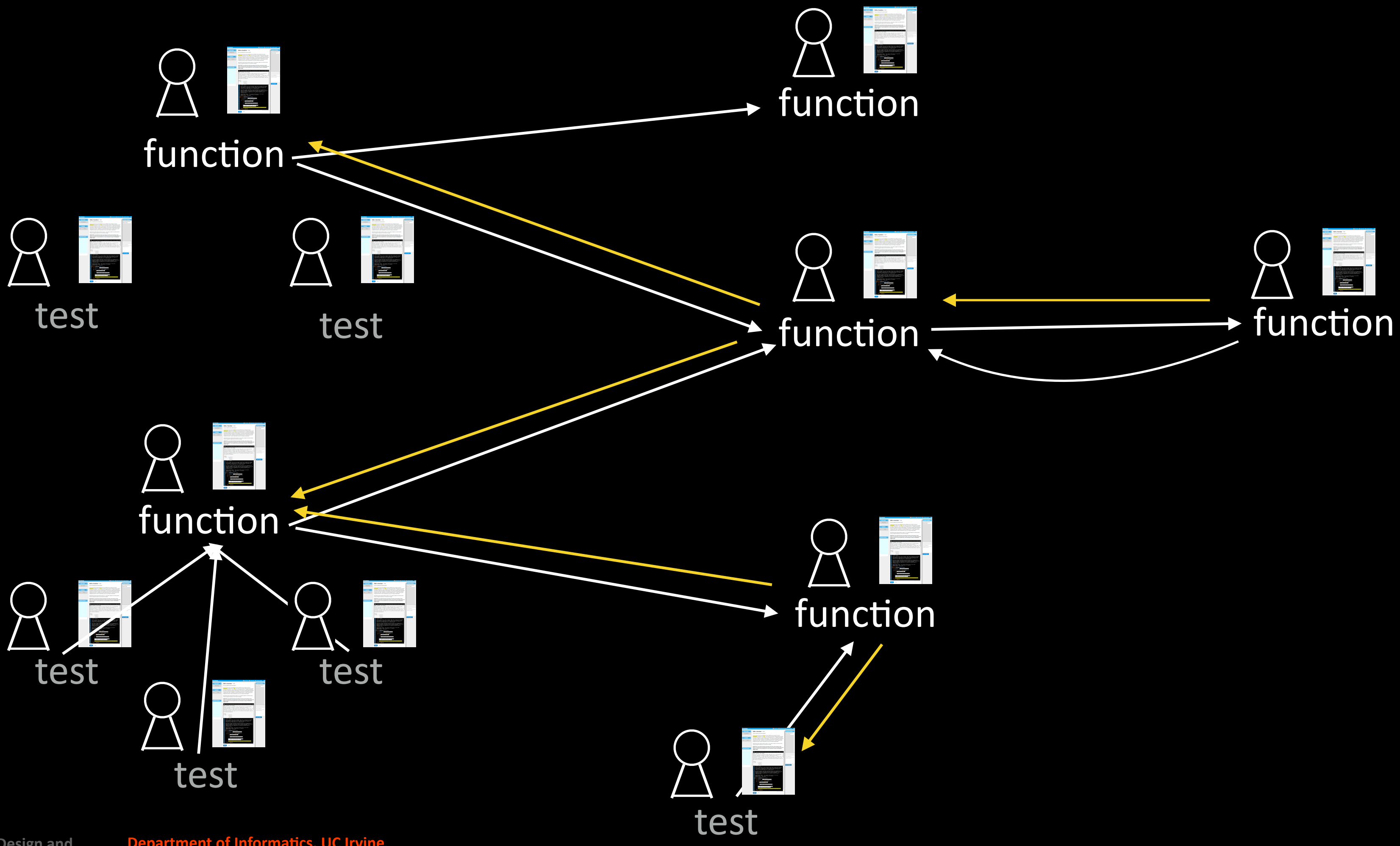
The description of a function called in the code below has changed. Can you update the code (if necessary)?

```
/**

Executes a move

@param Board board - board
@param Move move - move
@return Board
**/
function doMove(board)
```

If you're not sure how to do something, you can indicate a line or portion of a line as



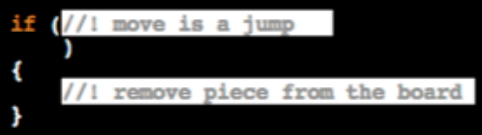
The dependency structure of software work



Coordinating programming work

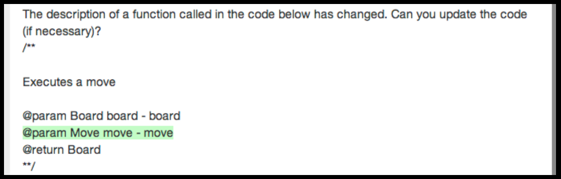
Artifacts send messages to other artifacts

- Request an artifact to be found or created



```
if (//! move is a jump
   )
{
   //! remove piece from the board
}
```

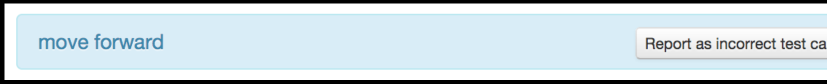
- Change a function signature



The description of a function called in the code below has changed. Can you update the code (if necessary)?

```
/**

Executes a move

@param Board board - board
@param Move move - move
@return Board
**/
```

- Report an issue in an artifact



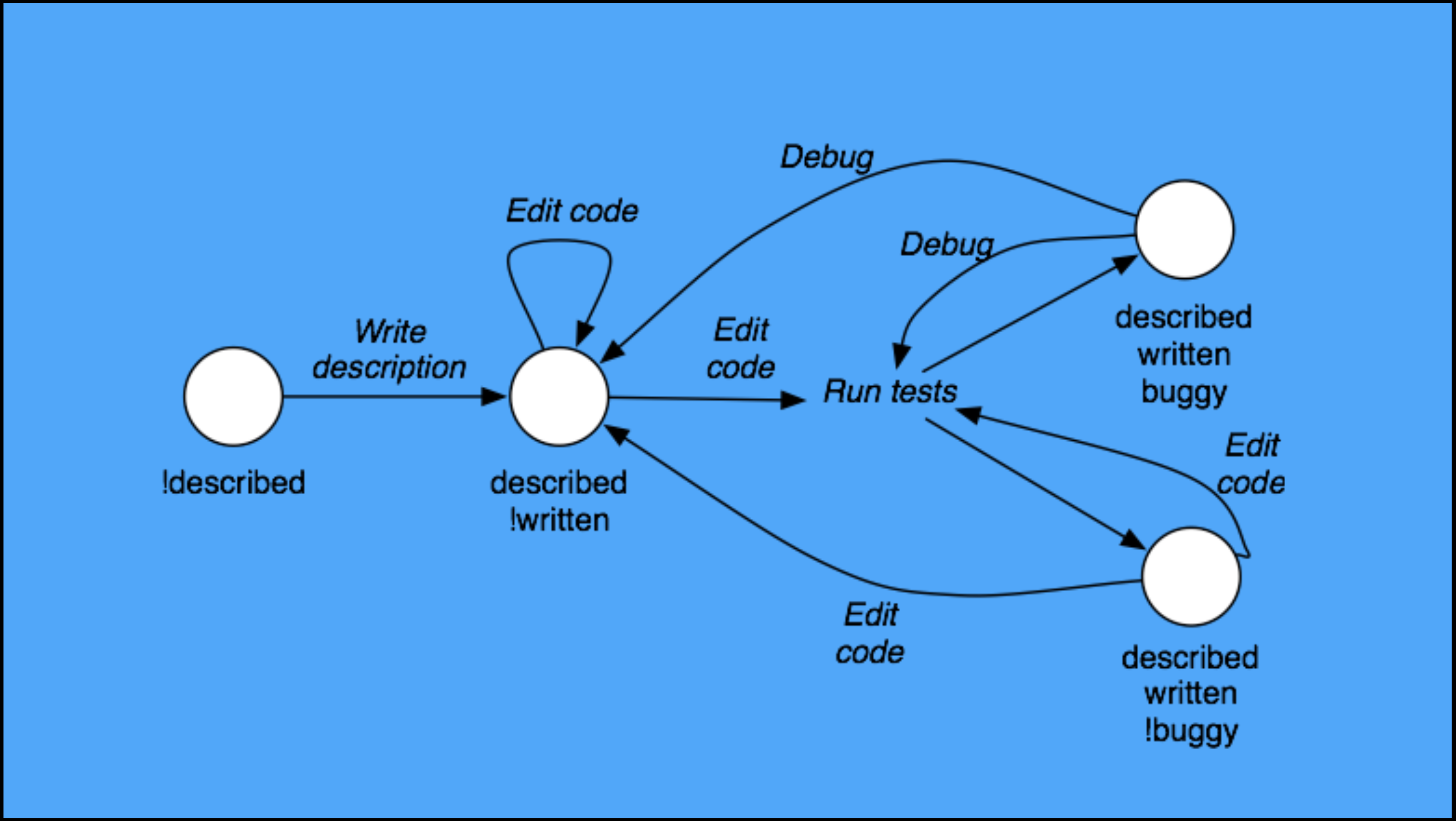
move forward

Each artifact may have an active microtask, enabling parallel work

- Messages may generate multiple microtasks to do on a single artifact
- To prevent merge conflicts, microtasks queued on artifacts

State of artifact tracked, used to generate microtasks

Function state machine



Testing

Given description, separate microtasks to write code, write tests

- Adds redundancy - code must pass tests

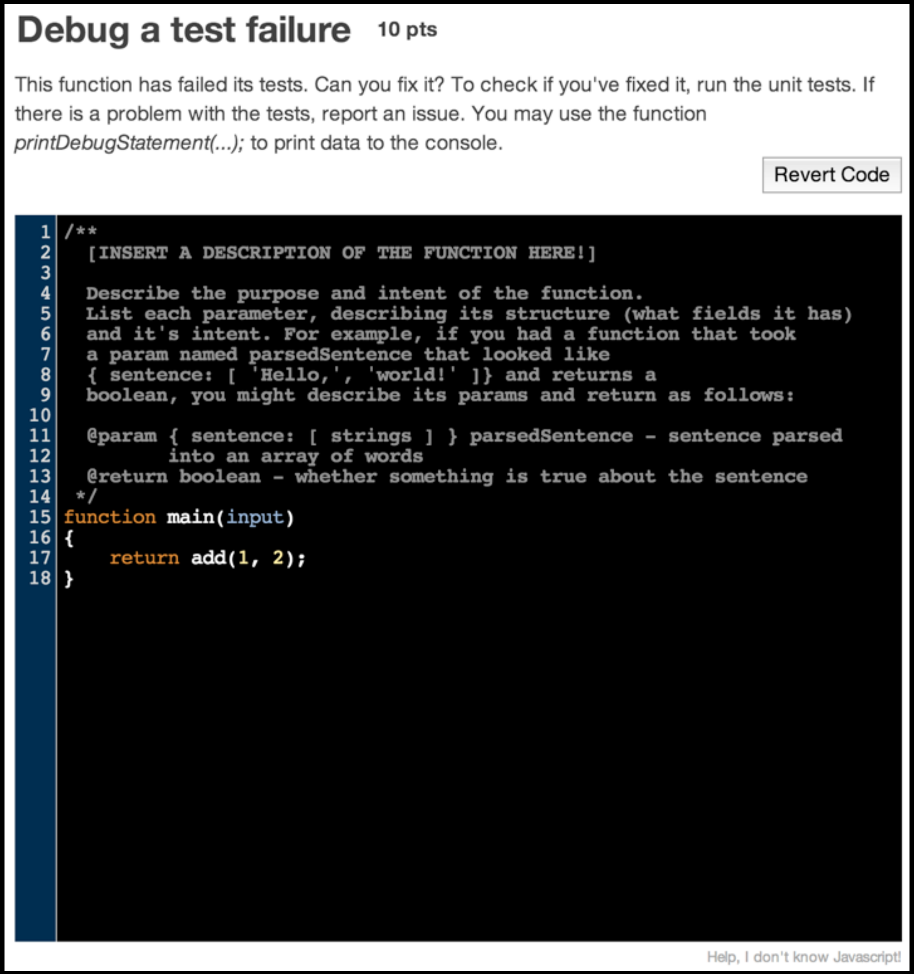
If function passes its tests, it is correct

- Assumes purely functional code (e.g., no shared mutable state or environment)
- Suitable for writing libraries

Run tests

- When function changes and is fully written (no pseudocode)
- Or when test changes
- If function's callees are not implemented, discard test results

Modular debugging by testing



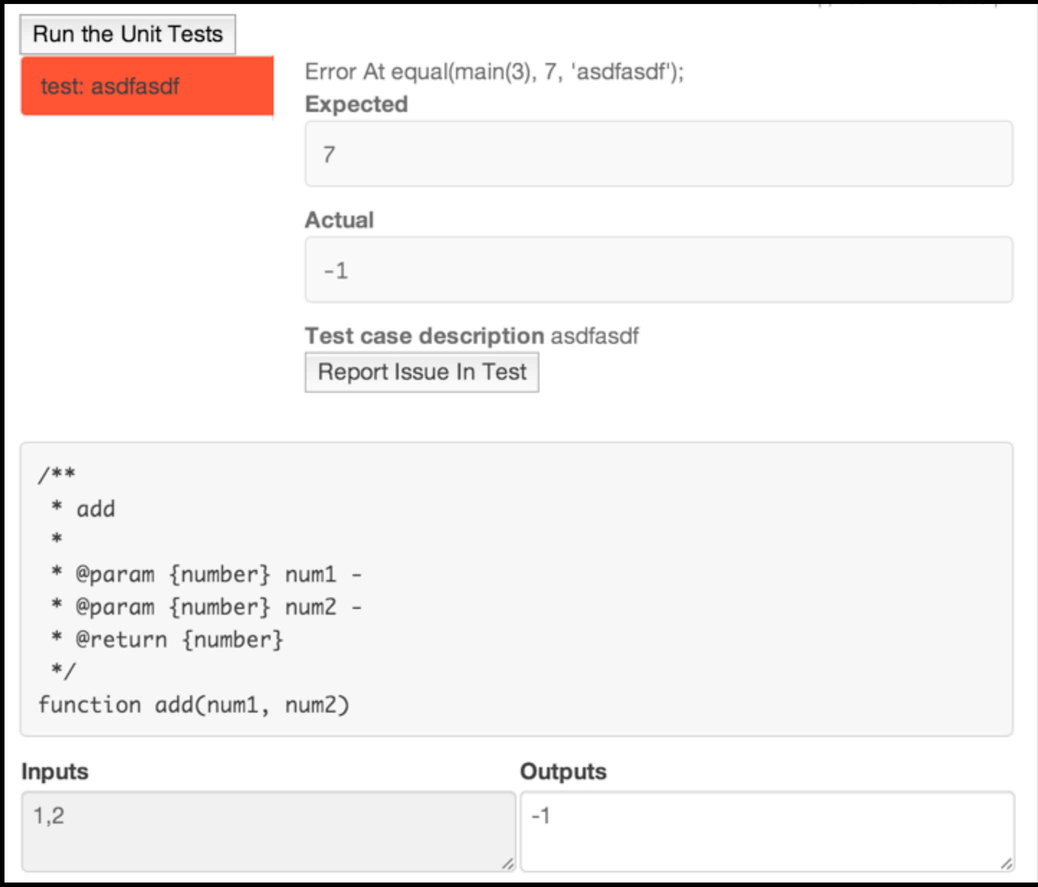
Debug a test failure 10 pts

This function has failed its tests. Can you fix it? To check if you've fixed it, run the unit tests. If there is a problem with the tests, report an issue. You may use the function *printDebugStatement(...);* to print data to the console.

Revert Code

```
/**
  [INSERT A DESCRIPTION OF THE FUNCTION HERE!]

  Describe the purpose and intent of the function.
  List each parameter, describing its structure (what fields it has)
  and it's intent. For example, if you had a function that took
  a param named parsedSentence that looked like
  { sentence: [ 'Hello,', 'world!' ]} and returns a
  boolean, you might describe its params and return as follows:

  @param { sentence: [ strings ] } parsedSentence - sentence parsed
         into an array of words
  @return boolean - whether something is true about the sentence
 */
function main(input)
{
    return add(1, 2);
}
```

Help, I don't know Javascript!



Run the Unit Tests

test: asdfasdf

Error At equal(main(3), 7, 'asdfasdf');

Expected

7

Actual

-1

Test case description asdfasdf

Report Issue In Test

```
/**
 * add
 *
 * @param {number} num1 -
 * @param {number} num2 -
 * @return {number}
 */
function add(num1, num2)
```

Inputs	Outputs
1,2	-1

Feasibility Evaluation: Is it possible to program using microtasks?

Lab study

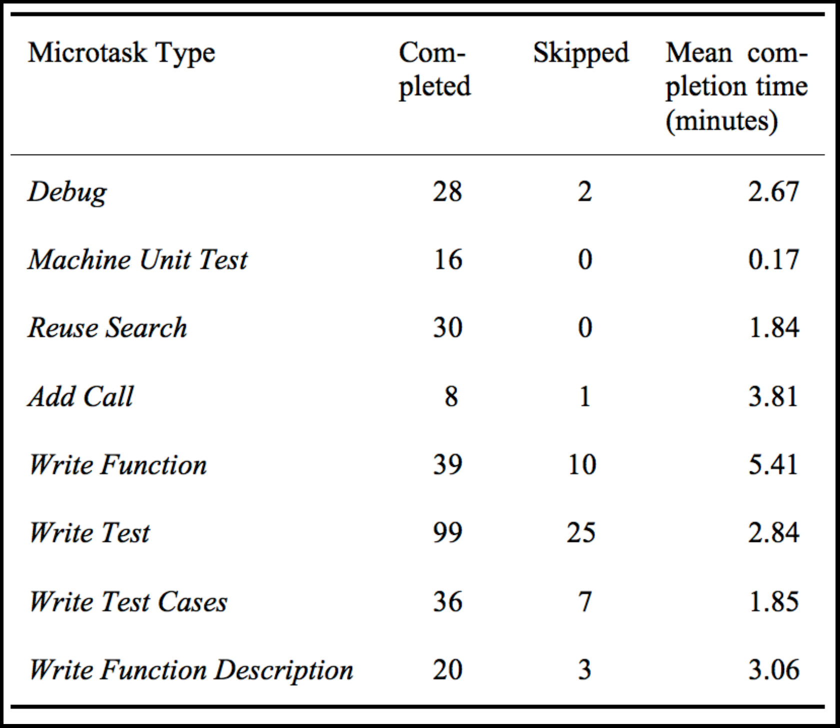
Crowd of 12 grad students & staff

Each provided separate room, only communication through system

Worked together for ~1.25 hours implementing checkers

Results

- Worked for a total of 14.25 person-hours
- Completed **265 microtasks**
- Wrote **480 lines of code** across 16 functions, and an additional 61 unit tests
- Did not finish implementing checkers



Microtask Type	Com-pleted	Skipped	Mean com-pletion time (minutes)
Debug	28	2	2.67
Machine Unit Test	16	0	0.17
Reuse Search	30	0	1.84
Add Call	8	1	3.81
Write Function	39	10	5.41
Write Test	99	25	2.84
Write Test Cases	36	7	1.85
Write Function Description	20	3	3.06

Perceptions of CrowdCode

Participants differed in reaction to the loss of **context** in microtasking:

- Some found it freeing: "*I had to keep less context in my head when writing functions, because I could not make assumptions [about] the rest of the program*" (P6)
- Others found it burdensome and wanted more information not provided

Participants appreciated ability to **specialize**:

- "*I think that CrowdCode would make me more likely to contribute as I could solve the tasks which I could do, and skip the others. I could take on tasks with higher difficulty as and when I feel comfortable. Hence, CrowdCode would be ideal in working in an OSS project…* "(P11)
- "*I was willing to be **imperfect** with my work. It was more important for me to constantly push out new work.*" (P1)

Found social features (esp. points and leaderboard) motivating

- "*help building a **productive vibe** to coding*" (P10)

11 of 12 participants agreed would be **more likely to contribute** to OSS project with CrowdCode

- Lower barrier to entry compared to "taxing" "learning and involvement curve" (P7) of OSS
- Ability to specialize by skipping some tasks
- Might be too constraining for seasoned developers but may be better for newcomers (P1)

Majority agreed that more **communication** would help them work more effectively

- Cited a desire to share technical experience, clarify tasks, ask questions about others' work

Conclusions

Basic programming tasks can be done modularly

- Decontextualization of work may have both benefits and drawbacks
- May enable transient work, specialization, and more fun (?)

Much more to software development work

- Discussion, GUIs, software design
- Can all software tasks be microtasked? Should they?

Generating microtasks through artifact state machines enables dynamic, creative work to be microtasked

- May be applicable to other domains (e.g., authoring a large document)