**Minesweeper**

You will be implementing the game Minesweeper. For official rules, please see: [http://www.freeminesweeper.org/help/minehelpinstructions.html (Links to an external site.)Links to an external site.](http://www.freeminesweeper.org/help/minehelpinstructions.html)

**Pair programming (optional)**

You have your option of working on this project either using *pair programming* or individually. Note that pair programming provide some significant benefits, and we generally recommend that you work with a partner unless you have a specific reason you can't easily do it, but there are a few things you should be aware of.

* You can only partner with someone who is *officially enrolled in the same lab section as you*. We cannot and will not make exceptions to this rule.
* Partnerships are expected to be participating in pair programming, which is a term that has a specific meaning — and, most notably, requires both partners to be in the same room, or (less ideally) perhaps on a videoconference with screen-sharing software. Be sure you've read the section entitled *Pair programming* on the [front page of the Project Guide](https://www.ics.uci.edu/~thornton/ics32a/ProjectGuide/index.html), so you're aware of what it means to participate in pair programming.
* Partnerships are expected to work on the project freshly, so if one or both partners have taken this course previously, the previous quarter's code cannot be used, in whole or in part, during your work on the project this quarter. This is so that both partners are taking a fresh journey through the solution; starting with a prior solution deprives at least one partner of that journey, which is how the learning is to be done.
  + This rule does not apply to students who are not partnered, who can reuse their own work. However, if you're one of those students, you should strongly consider whether relying on the work from a prior quarter in which you failed the course, rather than doing the work again and learning the things you misunderstood the first time, is a good way to pass the course this time.

**What to do if you want a partner**

During your lab section on Wednesday, October 24, choose a partner from among the students who are *officially enrolled in the same lab section as you*. It's fine, even preferable, to read this project write-up on your own ahead of time, though, so you and your partner can hit the ground running when you start working together, but do not start working on your solution until you are partnered up. (We understand that you might be eager, but the goal here is to take a shared journey with someone else, not to arrive on the first day and say "Okay, we're already done!")

Once you've selected a partner, notify your TA of your partnership during the lab section. Assuming you're both enrolled in that section, your TA will approve the partnership and make a note of it, at which time you're officially partners! (Until you've received this approval, you are not yet partners.)

For those of you who are unable to attend lab on Wednesday, October 25, there's a backup plan if you want a partner. Notify your TA via email — contact information for the TAs of each lab section is in the [Course Reference](https://www.ics.uci.edu/~thornton/ics32a/CourseReference.html) — that you will not be attending lab, but that you would still like a partner.

After labs meet on Wednesday, October 24, your TA will randomly select partnerships from among the students who did not attend, but who did want a partner, and will notify you and your new partner via email. Once your TA has selected a partner for you in this fashion, we will not allow you to switch to another one, so the best way to control your destiny is to choose a partner yourself during your lab section on Wednesday, October 25.

If you're having trouble finding a partner, notify your TA during your lab section, so that you can be assisted in finding one.

**What to do if you do not want a partner**

Nothing. Entering a partnership requires a small amount of action on your part — notifying your TA during lab, or emailing your TA beforehand and expressing an interest in a partner. Inaction means that you'll be working alone on this project.

**Is there any kind of penalty for working alone?**

No. The way we grade your work is the same, and the due date is the same, regardless of whether you work with a partner or alone. Partnerships are their own reward.

You will be implementing a custom version of the game, where the player will indicate the height and width of the board, as well as the number of bombs. The bombs will be randomly placed on the board. The user will input a coordinate of the board, and after every “turn” an updated board will be printed out. The game will continue until:

* A bomb is selected (Lose)
* All non-bomb locations have been selected (Win)

You will be writing a Minesweeper class that will be the logic behind your game. Define the following methods:

* Constructor – takes in the dimensions of the board and how many bombs, and randomly places the bombs on the board. The number of bombs should not exceed (number of squares - 1).
* print – print the state of the board
* checkValid – check if the coordinates are a valid coordinate on the board and that the state of that coordinate has not been revealed yet
* select – takes the coordinate and returns a message if it is not a valid coordinate or if it is not blank. If it is, it will reveal the appropriate information for the square and its adjacent squares (according to the rules)
* selectFirst – takes the coordinate on the first turn, and if it is a bomb, it will randomly select another position for the bomb and do everything in select.
* flag – takes a coordinate to flag for a bomb. Should give an error message if the coordinate is not blank or if it is invalid.
* unflag - takes a coordinate to unflag a bomb. Should give an error message if the coordinate was not flagged or if it is an invalid coordinate.
* checkStatus – returns the status of the game: win, lose, in progress
* \_getSolution – returns the solution of the board for testing purposes only. This method will allow us to test your program, but you should not be calling/using it, other than for testing purposes.

You will be writing Driver script that will create a Minesweeper object every time a new game is started. This will be what “drives” the game. It should make the appropriate calls to the methods in Minesweeper depending on decisions made by the player. It should also keep track of how many games have been won or lost.

You may write any helper methods that you find necessary. Sample output may look like the following:

Enter a height: 3  
Enter a length: 8  
Enter the number of bombs: 5

  1 2 3 4 5 6 7 8  
1 - - - - - - - -

2 - - - - - - - -  
3 - - - - - - - -  
Select an x coordinate: 1  
Select a y coordinate: 1

  1 2 3 4 5 6 7 8  
1 0 1 - - - - - -

2 1 2 - - - - - -  
3 - - - - - - - -

“select” or “flag” or “unflag”? flag  
Select an x coordinate: 1  
Select a y coordinate: 3

  1 2 3 4 5 6 7 8  
1 0 1 F - - - - -

2 1 2 - - - - - -  
3 - - - - - - - -

“select” or “flag” or “unflag”? unflag  
Select an x coordinate: 3  
Select a y coordinate: 1  
Invalid Input  
  
“select” or “flag” or “unflag”? unflag  
Select an x coordinate: 1  
Select a y coordinate: 3

  1 2 3 4 5 6 7 8  
1 0 1 - - - - - -

2 1 2 - - - - - -  
3 - - - - - - - -

“select” or “flag” or “unflag”? select  
Select an x coordinate: 2  
Select a y coordinate: 6  
  1 2 3 4 5 6 7 8  
1 0 1 - - - - - -

2 1 2 - - - 3 - -  
3 - - - - - - - -

“select” or “flag” or “unflag”? select  
Select an x coordinate: 2  
Select a y coordinate: 6  
  1 2 3 4 5 6 7 8  
1 0 1 - - - B - -

2 1 2 - - - 3 - -  
3 - - - - - - - -

Bomb! You lose

  1 2 3 4 5 6 7 8  
1 0 1 - - - B - -

2 1 2 B - B 3 - -  
3 - B - - B - - -

Do you want to play again ("yes" or "no")? No  
Win: 0 Lose: 1