

CSSE 220 Minesweeper Programming assignment – enhanced

This is a pair assignment. If there are an odd number of students in your section, there will be one team of three.

My intention is not that you "divide and conquer" as much as that you have someone to talk with as you write and test this program. If you have not already done so, read this short article on Pair Programming and discuss it with your partner: http://en.wikipedia.org/wiki/Pair_programming . In particular, note what it says about who should be the driver if you are a "mismatched pair".



Everything that you submit for this project should be understood by both partners. It is your responsibility to (a) Not submit anything without first discussing it with your partner, and (b) not let something your partner writes go "over your head" without making a strong effort to understand it, including having your partner explain it to you of course.

Grading note: Usually both partners will receive the same score for this program. But if there is ample evidence that one person did not fully participate in the learning and the doing, I reserve the right to give different grades. A peer evaluation survey at the end of the project will help me to determine this. If the survey or my observations indicate the need, I may ask you to explain parts of your project to me.

Game description pointer: You will write a game that is patterned after the popular Minesweeper game that has been available with the MS Windows operating systems for many years. You can read the Help that is provided with this program to get an idea of how it should work. You do not need to have a help menu in your program.

Repository name: <http://svn.cs.rose-hulman.edu/repos/minesweeperX> , where X is your team number (see the last page of this document for partner and team number information). You should check out the project from this repository, and all subsequent work should be placed in this project folder and committed back to your repository.

Milestones:

Tuesday, December 18 (end of class)	UML class diagram, also a plan for when you will meet this week.
Friday, Dec 21 (5 PM)	Iterative enhancement plan.
Monday Jan 7 (end of class)	Progress report 1. What in your plan has been written/tested?
Tuesday, Jan 8 (end of class)	Progress report 2
Thursday, Jan 10 (end of class)	Progress report 3
Monday, Jan 14 (8:05 AM)	Final working code, unit tests, revised UML class diagram
Tuesday Jan 15 (8:05 AM)	Evaluation of yours and your partner's work on this project.

UML Class Diagram (first milestone, due Day 11)

First you should figure out what classes you need and how they will interact. Put these ideas in a UML class diagram that you save in your project folder and commit to your repository by the end of the Day 11 class period. Update your diagram whenever your design changes.

How to commit your diagram: In Violet, use *Save As* to save the file in your Minesweeper project folder inside your Eclipse Workspace folder. In the Eclipse program, right-click the project folder in the Package Explorer view, and choose Refresh. Your Violet file should appear. Now right-click the project folder again, and choose Team→Commit. Select the file(s) to be committed.

Iterative Enhancement Plan (second milestone, due Friday of week 4)

Software engineers typically implement their projects by using **Iterative Enhancement**, which means:

- They implement the project in *stages*.
- Each stage adds more functionality.
- **Each stage must be testable by running the application.**
- The final stage implements all the required functionality.
- They **test each stage as they implement it**, rather than doing all the testing at the end.

An **Iterative Enhancement Plan (IEP)** is a list (in ordinary English) of the stages.

- Each stage's description explains what functionality must be supplied at that stage.

- **The description must refer only to the project's functionality, not the project's code.**
(e.g. "The pull-down menus appear, without functionality" is correct; "Write the menu classes" is not correct)
- **The description should make it clear what test(s) one would run in order to test that stage.**
- Your Iterative Enhancement Plan should list the stages of your plan and for each stage, give your single statement of the work to be accomplished at each stage and the detailed requirements that stage will fulfill.
- I have provided an Excel form in which you can create your IEP. It is in the [assignment folder](#). Copy this form, fill it out, and save it (with the same file name) in the Minesweeper folder in your Project folder and commit it to your minesweeper repository. The instructions for doing this are similar to the ones given above for saving your UML class diagram.

Examples of some possible steps in a MineSweeper plan (not necessarily the first two steps you should do):

Project name: Minesweeper		Team number and members: 22 Peter Parker, Mary Jane Watson		
Stage	What the program will do at this stage	How we plan to test this feature	expected completion date	actual completion date
1	Display a frame with a properly-sized grid of blank buttons	Run the program, count the rows and columns of buttons.	20-Dec	
2	There is a menu that allows user to select from Beginner, Intermediate and advanced, and produces the proper size grids.	Select these menu-items, again count buttons and make sure that the window resizes accordingly.	21-Dec	

The rest:

Begin implementing, commenting, and testing your code. Create JUnit tests for any parts of the program that can be tested without the GUI. Commit your project often.

You should include as much of the Minesweeper functionality as you can in the time allowed. If you can't get it all done, here are some things that I consider lower priority than other things (the least important things to implement are at the top) of the list.

1. Everything in the Help menu.
2. Sound.
3. The ability to place the ? mark on squares.
4. The ability to simultaneously left-and-right click (as described in Minesweeper help).
5. Colors.
6. Best Times list.
7. Custom mode.
8. Displaying the elapsed time.

An "A" project should do many of the things on this "least important" list - perhaps all of them. I'll have to see how things go; I have never assigned this project before. For now, I can say that a team that delivers all of the milestones well-done and on time, and produces a project that correctly implements all of Minesweeper except the above 8 items, will earn at least 80% for their Minesweeper score.

Repsitory name	student 1	student 2	student 3
minesweeper1	allencw	wenzel	
minesweeper2	andersar	cranemd	
minesweeper3	baeaep	brumbams	
minesweeper4	baekj	reillywj	
minesweeper5	bergencb	berrygl	
minesweeper6	blankeaz	burnsjl	
minesweeper7	dunnmp	vanderkl	
minesweeper8	fehribrm	sullivsd	
minesweeper9	junkersa	leroycw	
minesweeper10	williakl	middlemp	
minesweeper11	pietars	pridalmj	
minesweeper12	salisbjm	watersbt	
minesweeper13	bottjd	thomass2	
minesweeper14	gerthwd	pickdp	
minesweeper15	goodrijk	snivelee	
minesweeper16	hansenrl	merseljp	
minesweeper17	hilljd	tamal	
minesweeper18	iversopn	bussinjr	
minesweeper19	jennemj	buetowbp	
minesweeper20	kotsybja	wisejl	
minesweeper21	kriesbsd	yimah	
minesweeper22	nowickpj	watersdc	kaiserja
minesweeper23	skaggskd	thiememd	