Team 2 Project 1 Requirements Documentation

Task Overview

Task Name	Priority Level	Description
Board Implementation	1 - Necessary	The board should be comprised of an array comprised of lists and support each character being changed to represent what is on the tile.
Mine Placement Algorithm	1 - Necessary	An algorithm to randomly place mines across the board.
Bomb Adjacency Algorithm	1 - Necessary	An algorithm (may be implemented within the placement algorithm) to determine the number of mines within each tile
Victory/Defeat State	1 - Necessary	States to prevent the user from acting out of turn after the game is completed as well as revealing all tiles
Recursive Grid Reveal	1 - Necessary	Algorithm to reveal any squares adjacent to ones with 0 bombs that have been selected
User Reveal	1 - Necessary	Function handling the user revealing tiles
User Flagging/Unflagging	1 - Necessary	Function handling the user's ability to flag/unflag tiles
Replay Function	2 -	Placing code within a loop to allow the user to play multiple times
GUI	2	Tying the Code to a readable and clean GUI.

Team 2 Project 1 Task History

Task Name	Hours (Est.)	Hours (Actual)	Difficulty Level (Est.) (1-10)	Implementation Notes
Board Implementation	0.5	0.5	1	Implemented by Gunther Luechtefeld. The only issue I ran into was labeling the axes.
Mine Placement Algorithm	1	1	3	Implemented by Gunther Luechtefeld. No issues encountered.
Bomb Adjacency Algorithm	1.5	2	5	Encountered off-by-one error in loop ranges, testing also took longer than planned due to needing to develop function to reveal all cells – Implemented by Jacob Kice
Victory/Defeat State	0.5	1	2	Implemented by Srihari Meyoor Prevented flagging every cell to win, forced to clear every safe cell
Recursive Grid Reveal	1	1	7	Implemented by Jacob Kice Uses user reveal to reveal specific cells, also uses cell adjacency technique developed for bomb adjacency algorithm
User Reveal	0.5	0.5	5	Implemented by Jacob Kice Uses recursive grid reveal when appropriate
User Flagging/Unflagging	1.5	1.5	4	Implemented by Srihari Meyoor The only issue I ran into was the adjusting flag count for unflagging.
Replay Function	0.5	0.01	1	Simply needed to place all relevant code within a while loop.

GUI	4	5.5	8	Implemented by Jamie King, no major issues encountered.
Code Review/Testing/Debugging		3		Conducted by Jacob Kice Added mines remaining indicator to display, fixed bug in victory check function
Architecture Documentation and Read Me	0.5	1		Created by Jacob Kice
Prologue Comments	.5	0.5	1	Created by Jacob Kice
In function Comments	0.5	0.5	1	Created by Srihari Meyoor Implemented by Jacob Kice, Srihari Meyoor Multiline comment with args, output, purpose. Inline comments as well.
Requirements Doc – Task Overview	0.25	0.25	1	Created by Joe Hotze
Requirements Doc - Meeting History	2	2	1	Created by Joe Hotze
Requirements Doc - User Stories	1	1.5	1	Created by Joe Hotze
Requirements Doc – Precedence Diagram	1	1	2	Created by Joe Hotze

Team 2 Project 1 Meeting History

Meeting Date	Topic	Description
September 2	Requirements	Discussed Requirements of project and created documentation of necessary functionality to implement
September 4	TA Meeting	Discussed progress so far, needing to implement more into the code and requirements doc. Future plan – functional code finished within a week, code documentation and GUI implementation planned for the following week. Meetings now at 12:30.
September 11	TA Meeting	Discussed new progress, original plan going well - functional code fully completed.

		Moving forward, only need GUI as well as documentation within codebase.
September 18	TA Meeting	Presented project with working GUI, complete features, and completion.

Team 2 Project 1 User Stories

Title: Board Implementation	**Priority:** 1 - Necessary	**Estimate: 1**
User Story: As a player, I want to open the game and see a representation of the minesweeper board.		
Acceptance Criteria: Player is able to open the game and see a 10x10 grid that represents the minesweeper board. Board is properly able to represent a mine, a number of adjacent mines, a flag, or nothing present.		

Title: Mine Placement Algorithm	**Priority:** 1 - Necessary	**Estimate: 3**
User Story: As a player, I want to click on the board without immediately losing and know that mines have been placed randomly on the board.		
Acceptance Criteria: Program successfully generates mines randomly AFTER the user has selected their first space, as to avoid instant losses. Mines are not placed on top of one another.		

Title: Bomb Adjacency Algorithm	**Priority:**	**Estimate: 5**
User Story: As a player, when I select a tile with mines next to it, I want to see the exact number of adjacent mines displayed to me.		
Acceptance Criteria:		

Program successfully generates a number for each non-mine tile representing the number of mines

Title: Victory/Defeat States	**Priority:** 1 - Necessary	**Estimate: 2**
User Story: As a player, I want to be informed by the game exactly when I've won or lost, as well as being able to play again.		
Acceptance Criteria: Program properly terminates the game on a loss or victory and presents the user with an option to play again. No further actions can be taken by the player while		

Title: Recursive Grid Reveal	**Priority:** 1 - Necessary	**Estimate: 7**
User Story: As a player, when I click on a square with no adjacent mines, I want to see all squares nearby revealed until tiles with adjacent mines are uncovered.		
Acceptance Criteria: Program is able to recursively and correctly reveal proper tiles when necessary. Any incorrectly revealed or unrevealed tiles that should be revealed will be considered a failure.		

Title: User Reveal	**Priority:** 1 - Necessary	**Estimate: 5**
User Story: As a player, I want to click my mouse in order to reveal whether a tile contains a bomb, nothing, or an adjacency number displaying the amount of nearby mines.		

Acceptance Criteria: The program is correctly able to reveal and resolve the hidden tile depending on what lies underneath.

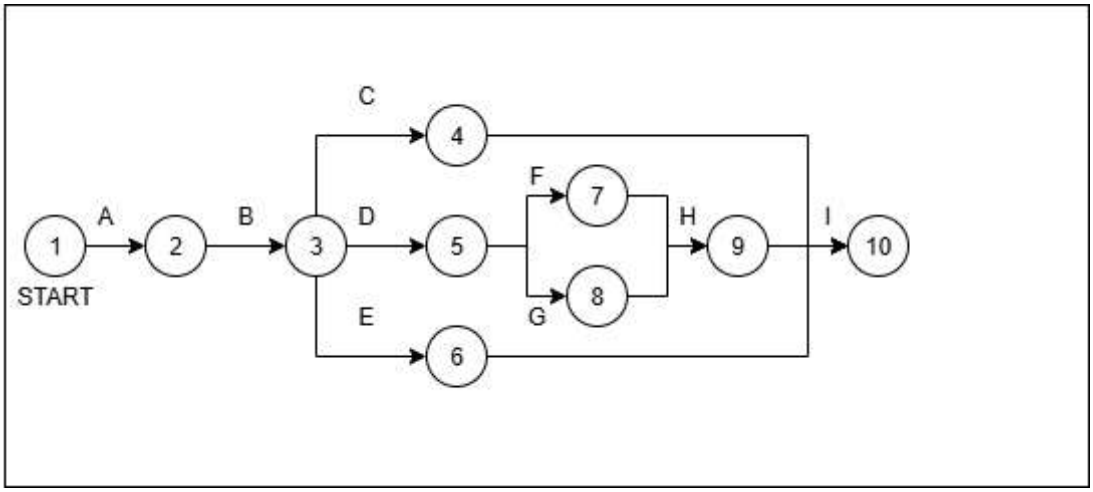
Title: User Flagging/Unflagging	**Priority:** 1 - Necessary	**Estimate: 4**
User Story: As a player, I want to right click my mouse in order to place a flag on it. I also want to be able to do the same to a flagged tile in order to remove the flag.		
Acceptance Criteria: The program allows the user to flag by recognizing right clicks, assigning the flag to the proper tile, and ensuring unflagging is used as requested. These flags should reduce the mine counter appropriately.		

Title: Replay Function	**Priority:** 2 - Preferred	**Estimate: 1**
User Story: As a player, I want to be offered the ability to play the game again after a defeat or victory.		
Acceptance Criteria: The program gives the user the option to restart the game after it is complete, resetting the board and mine placements to new spots.		

Title: GUI	**Priority:** 2 - Preferred	**Estimate: 8**
User Story: As a player, I want to be able to see and interact with the board using my mouse.		

Acceptance Criteria:
The program generates a visualization of the matrices that can be interacted with using the mouse.

Team 2 Project 1 Precedence Network



Activity	Immediately Preceding Activities	Expected Completion Time (Hours)
A – Board Implementation	None	0.5
B – Mine Placement Alg.	A	1.0
C – Bomb Adjacency Alg.	B	1.5
D – User Reveal	B	0.5
E – User Flagging/Unflagging	B	1.5
F – Victory/Defeat States	D	0.5
G – Recursive Grid Reveal	D	1.0
H – Replay Function	F,G	0.5
I - GUI	C,H,E	4.0