Roles:

Jace: Dev

Bryce: Team Lead

Kit: Tester

Ian: Documentation, Dev

Jenna: Dev

K: Dev

Task Breakdown:

- Multiplayer AI vs Player - Jace
- AI easy - Bryce
- AI medium - Ian
- AI hard - Ian - k
- UI - Jenna
- Sound- Kit
- UML Diagram - Ian

Person Hour Estimates

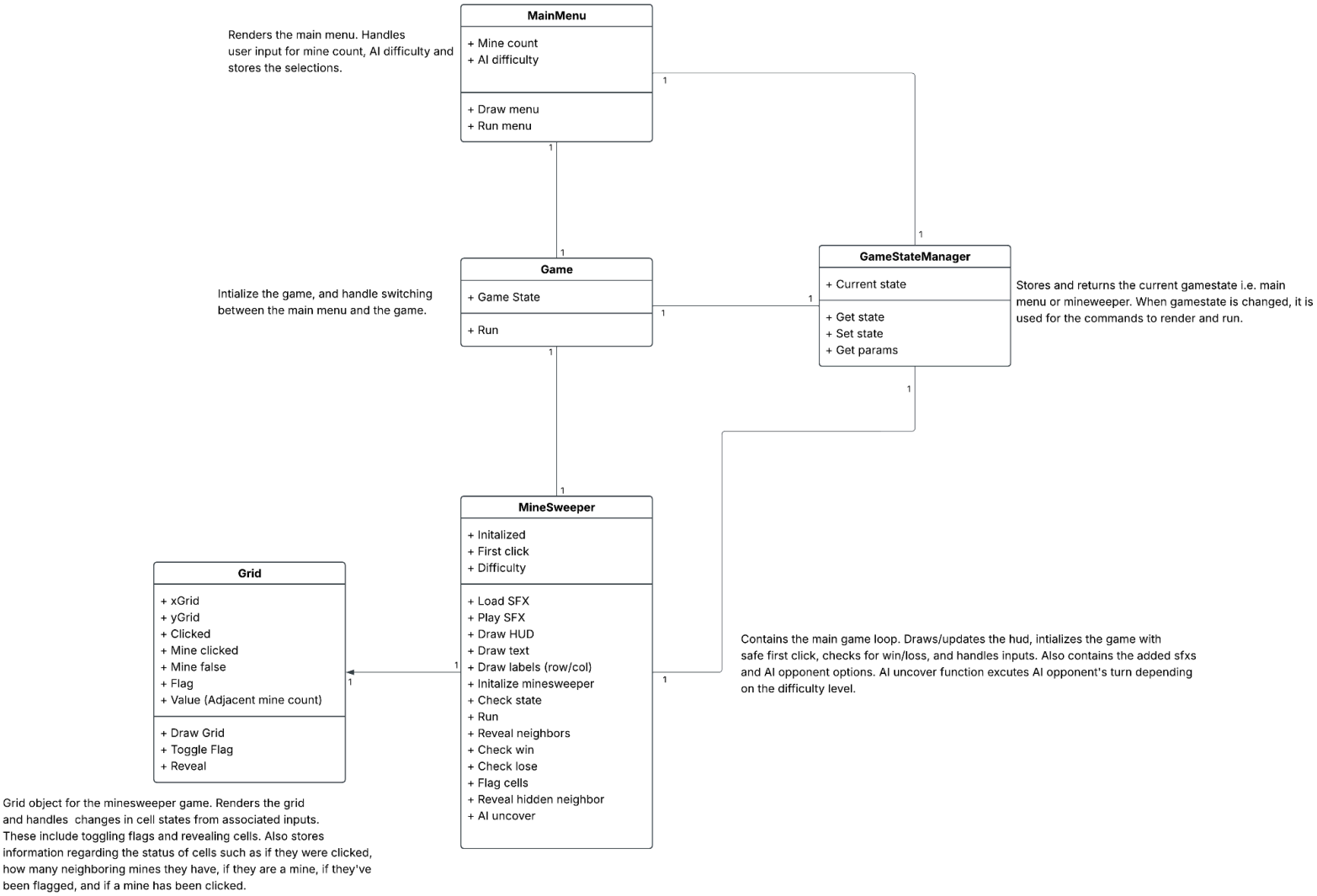
List of Tasks

- Multiplayer AI vs Player
- AI easy
- AI medium
- AI hard
- UI
- Sound
- UML Diagram
- Meeting
- Testing

	Jace	Bryce	Kit	Ian	Jenna	K	Final Estimate
Multiplayer	3	2	1	3	3	2	2
AI Easy	1	1	1	1	1	1	1
AI Medium	1	2	1	3	2	2	2
AI Hard	3	2	1	4	3	4	3
UI	1	1	1	1	1	1	1
Sound	1	1	3	1	1	1	1
Meeting	3	2	1	2	2	2	2
Testing	1	2	3	2	1	3	2
Documentation	1	1	1	1	1	1	1
Peer Evaluation	1	1	1	1	1	1	1
Total	16	15	14	19	16	18	16

For our group's person hour estimations we utilized story points and a version of planning poker. The list of tasks was compiled by referencing the assignment details and brainstorming for other tasks not explicitly detailed. Since we had now completed a project, we had options to choose for a reference story. In project 1 we had unanimously determined adding UI elements to be worth 1 story point. We used that task to estimate the story points for tasks in project 2. We privately voted on our person-hour estimates during one of our meetings and online. We then discussed the results and the average score was chosen for each task.. The list of tasks, and the breakdown of everyone's votes are detailed in the table above.

System Architecture Overview



Our additions to the existing minesweeper program were implemented mostly in the MineSweeper class. We added the structure to handle two players, the different AI difficulty levels, and sound effects for different events.

AI Easy:

- Randomly chooses an unclicked cell

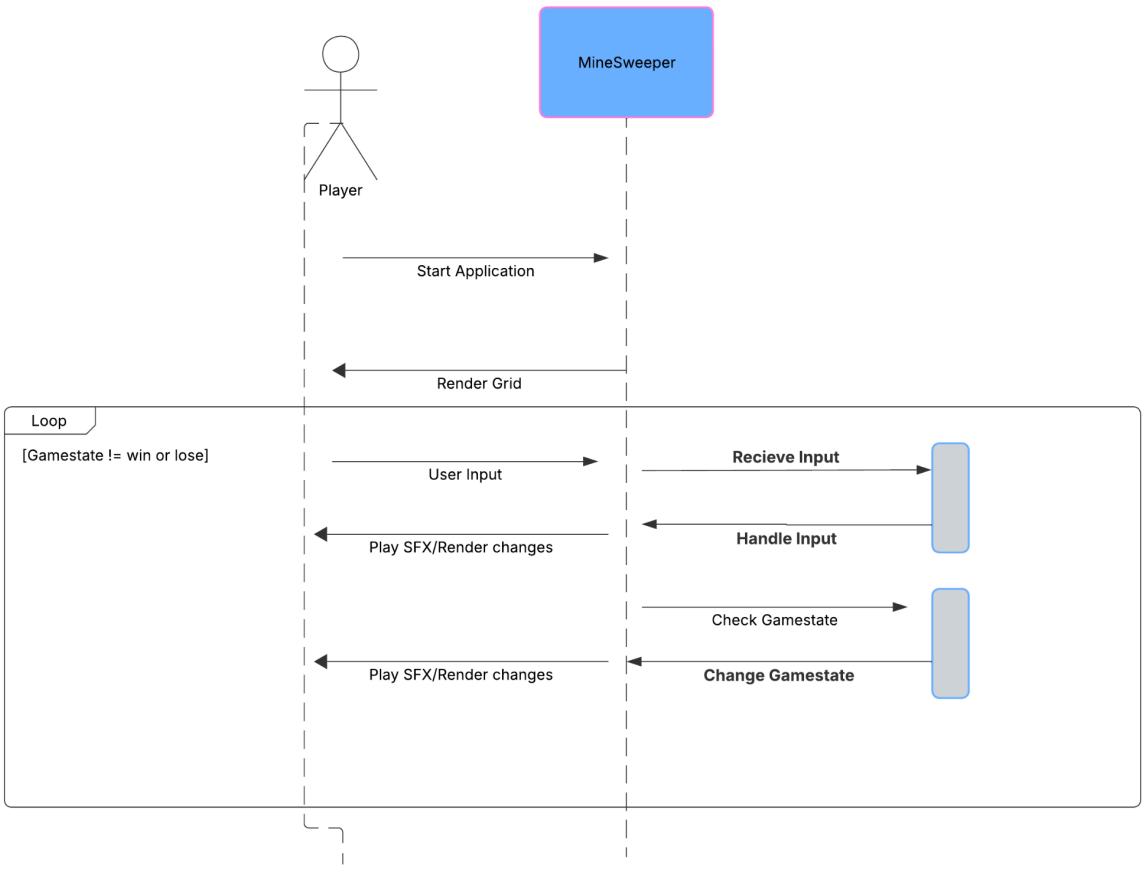
AI Medium:

- If a cell value matches the number of hidden neighbors, flag all hidden neighbors
- If number of flagged neighbors matches the value of a cell, reveal one hidden neighbor
- If no other conditions are matched, randomly chooses an unclicked cell

AI Hard:

- Same as medium, but after checking the second step, check for revealed cells with the 1-2-1 pattern. Flag outside cells, and reveal the middle cell.
- If no other conditions are matched, randomly chooses an unclicked cell

The player and AI take turns playing. A turn ends when an unclicked cell is clicked. No limit on the number of flags that can be placed per turn. The game ends when a player makes a mistake or if a player solves the grid first.



Sound effects were added for clicking a cell, flagging and unflagging a cell, winning, and losing.

Actual Person-Hour Tracking

Jace: 3 hours

Bryce: 6 hours

Kit: 5 hours

Ian: 6 hours

Jenna: 4 hours

K: 4 hours

Total = 23 hours

Timesheet

Jace		
Date (mm-dd)	Hours	
09/30	3	Laid out structure for AI solving allowing plug in of logic.
Bryce		
Date (mm-dd)	Hours	
9/25	1	Meeting
9/30	2	Implemented logic to pass difficulty in from the main menu.
9/30	1	Added Ai easy mode, randomly choose uncovered spots.
10/2	1	Clean up bugs from previous group, fix small AI bug
10/2	1	Demo / Meeting
Kit		
Date (mm-dd)	Hours	
9/30	1	testing program
10/1	3	adding sound effects to the program
10/2	1	demo/meeting

Jenna		
Date (mm-dd)	Hours	
9/25	1	Meeting
9/30	1	Added easy, medium, hard buttons to main menu + fixing positioning
10/2	1	meeting
10/4	1	final testing to make sure modes work correctly
K		
Date (mm-dd)	Hours	
10/2	4	Meeting, update the hard difficulty
Ian		
Date (mm-dd)	Hours	
9/25	1	Meeting
10/1	4	Added medium difficulty and some parts of hard difficulty
10/2	1	Meeting
10/5	1	Documentation