User Stories

User Story 1

Description: As a developer I would like a simple interface to connect a logic solver to allow users to play turns with an AI.
Acceptance Criteria: A developer should be able to pass their solver's logic to a method, which then creates gameplay with one turn for the user, then one turn for the solver and so on. The nextMove() method of a solver should be called to complete the solver's turn each time.
Derived Task: Implement turns interface for AI solvers.
Story Points: 3

User Story 2

Description: As a user I would like to play against an AI in easy difficulty mode.
Acceptance Criteria: The solver should be able to interface with the board, and decide which move to play. A nextMove() method should be exposed to play each next move.The AI will uncover cells randomly, avoiding flagged or already uncovered cells.
Derived Task: Implement easy mode AI solver.
Story Points: 3

User Story 3

Description: As a user I would like to play against an AI in medium difficulty mode.
Acceptance Criteria: The solver should be able to interface with the board, and decide which move to play. A nextMove() method should be exposed to play each next move.The AI will uncover randomly until a safe cell is revealed (zero adjacent mines), then uncovers adjacent cells strategically using revealed numbers.
Derived Task: Implement medium mode AI solver.
Story Points: 3

User Story 4.

Description: As a user I would like to play against an AI in hard difficulty mode.
Acceptance Criteria: The solver should be able to interface with the board, and decide which move to play. A nextMove() method should be exposed to play each next move.The AI will "cheat" by always uncovering a safe cell (non-mine), simulating perfect knowledge without detonating mines.
Derived Task: Implement difficult mode AI solver.
Story Points: 3

User Story 5.

Description: As a user, I would like to play against a friend cooperatively with alternating turns.
Acceptance Criteria: The user will be able to select a multiplayer mode and play with a friend. The game will alternate between first and second players and will track which player hits a bomb and display a message stating who has won. In the event of all mines being cleared, they will tie.
Derived Task: Implement a multiplayer mode in the game.
Story Points: 1

Time Tracking

Estimated Time per Story Point: 31 / 15.6 = 0.5
This was calculated based off project 1, in which we had 31 story points and 15.6 hours.

Time Estimate: (3 + 3 + 3 + 3 + 1) * 0.5 = 6.5 hours.
This is based off 3 story points for user story 1-4 and 1 story point for user story 5, and the estimated time per story point above.

User Story 1. (Sam)
Time: 40 minutes (10/05/2025)

User Story 2. (Meg)
Time: 55 minutes (10/05/2025)

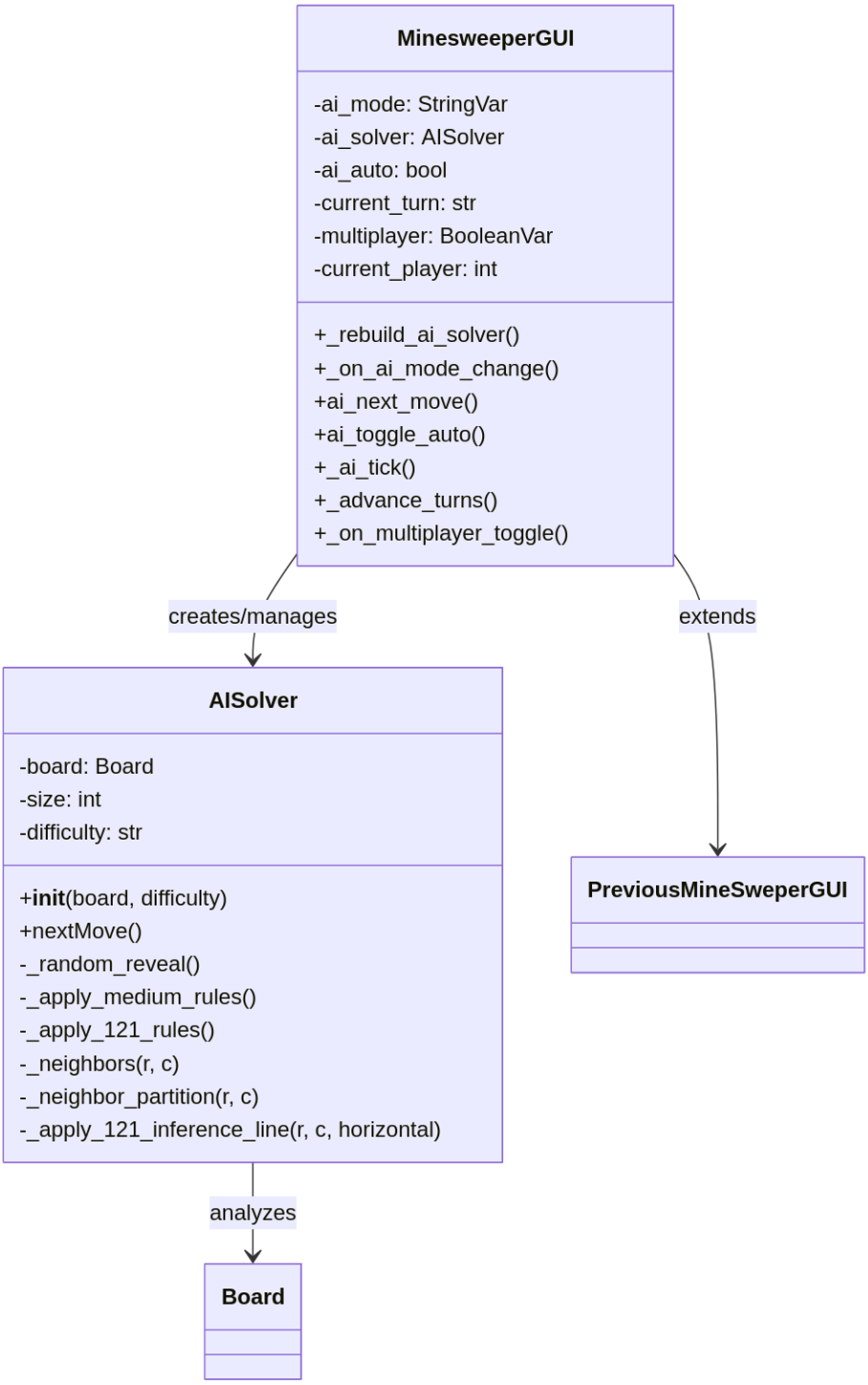
User Story 3. (Jenny)
Time: 60 minutes (10/04/2025)

User Story 4. (Genea)
Time: 75 minutes (10/05/2025)

User Story 5. (Sam)
Time: 20 minutes (10/05/2025)

Actual Time Time: 40 min + 55 min + 60 min + 75 min + 20 min = 4.16 hours

Architecture



Note: Here extends is purely for clarification. MineSweeperGUI “exends” PreviousMineSweeperGUI clarifies that the members and methods in MineSweeperGUI are the additional members and methods on top of what already existed from the previous version.

MineSweeperGUI creates an AISolver, as needed. Then, when it is the AI's turn, MineSweeperGUI calls the nextMove() method of AISolver to choose the AI's move. MineSweeperGUI manages the turns between the AI and the user, and allows the user to select between the hard, medium, and easy modes. This difficulty is stored in AISolver, and controls which logic is used to decide the next move.