

Meeting 2 - 9/3/2025

For project requirement: <https://people.eecs.ku.edu/~saiedian/581/Proj/proj1>

Links:

Github: <https://github.com/rileyfaith/EECS581-Project1>

Minesweeper User stories and Task division:

https://docs.google.com/document/d/1hSBCCqygbLink_lQrr8rBYbfAskKCYZmFcxNLoWkwgc/edit?tab=t.0#heading=h.2muxkmv23z4

- Due Date of Project 1: 9/21/2025 (Code Freeze)
- Demo for Project 1: 9/24/2025

Previous Tasks:

- Riley: Play Minesweeper for research
- Cole: Implement core gameplay user stories
- Evans: Reviewed the game rules and requirements, and researched the best implementation option (Python).
- Jackson: synergized cross-functional workflows to optimize value streams and drive scalable alignment across key deliverables (played minesweeper)
- Manu: studied assignment rubric for requirements engineering.

This Week's Tasks:

- Riley: Implement UI User Stories / Experiment with UI
- Cole: Implement core gameplay user stories
- Evans: Beginning and End
- Jackson: Implement core gameplay user stories
- Manu: Implement core gameplay user stories

Notes:**User Stories:****Beginning and End**

- As a player, I want the game to end with the message “Game Over” when I click on a mine.
 - All mines are uncovered
- As a player, I want to choose how many mines are in the grid (10-20).
- As a player, I want to see a “victory” message and have the game be over when I have uncovered all of the cells without mines.

Core Gameplay

- As a player, I want my first click to have no mine.
- As a player, I want all adjacent blank cells to be uncovered upon clicking one.
- As a player, I want to click on a cell to know whether it is a mine or not.
- As a player, I want the boxes with numbers to correctly indicate the number of adjacent mines.
- As a player, I want the mines to be randomly placed such that no two games are the same.

- As a player, I want to be able to flag/unflag cells.
 - As a player I want flagged cells to not be uncovered until unflagged

UI

- As a player, I want to see an indicator with the number of remaining mines (minus flagged mines).
- As a player, I want to see an indicator with the number of remaining flags (total mines - flags placed)
- As a player, I want to play the game on a 10 by 10 grid with columns labeled A - J and rows labeled 1 - 10.
- As a player, I want to be able to right click on a cell in order to flag/unflag cells.
 - As a player, I want to see flags on the cells.
 - As a player I want flagged cells to not be uncovered until unflagged
- As a player, I want the grids to be covered when I start.
- As a player, I want to see a game status of 'Playing', 'Game Over' or 'Victory'

Meeting 3: 9/10/2025

Previous Tasks:

- Riley: Experiment with UI, Familiarize with PyGame
- Cole: Implement core gameplay user stories
- Evans: Created a program that prompts the user for the number of mines they want (between 10-20). Also created the pygame window and fonts, which have status messages, mine count, flags left, and restart hints at the bottom of the screen
- Jackson: Implemented core gameplay: Flags, occupancy grid, first click logic

- Manu: Implement core gameplay: Unflagging of cells functionality, restart game functionality

This Week's Tasks:

- Riley: Implement UI User Stories
- Cole: Implement core gameplay user stories
- Evans: Created an interactive main menu for the game with “Play Game” and “Quit” buttons using Buttons that respond to mouse clicks, allowing the user to start the game or exit. Implemented a slider bar for the player to choose the number of mines before they start the game.
- Jackson: Implement core gameplay user stories
- Manu: Ensure compatibility of the core gameplay implementation with the UI and beginning and end features created by my teammates.

Meeting 4: 9/17/2025

Previous Tasks:

- Riley: Implement UI User Stories
- Cole: Implement core gameplay user stories
- Evans: Beginning and End
- Jackson: Implement core gameplay user stories

- Manu: Ensure compatibility of the core gameplay implementation with the UI and beginning and end features created by my teammates. And improved commenting in main.py

This Week's Tasks:

- Riley: Continue to Implement UI User Stories
- Cole: Work on system documentation
- Evans:
- Jackson: Implement core gameplay user stories
- Manu: Finish up the readme

Notes

- Need to adjust core gameplay so that number of mines is utilized
- Need to adjust gameplay such that Victory is a game state (check for # mines and cells left unrevealed)
- Center Gameboard?
- Need to complete system documentation
- Need to merge main with beginning and end