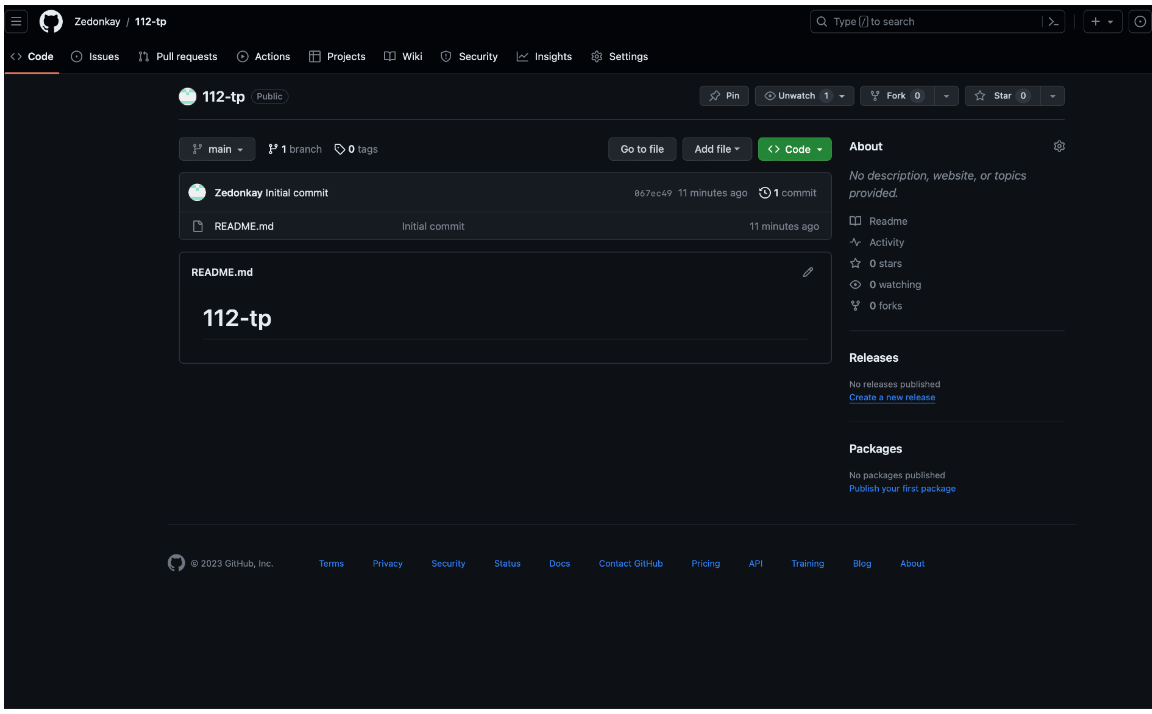
1. Project Proposal
    a. Project Description
        i. The name of my project is Mariosweeper. This project is based on minesweeper. It will have the same basic logic. There will be a grid, with flowers that are undetected to the player. The player will click a random square, and then the board will display empty squares and numbers. The numbers indicate that there is a flower touching it. If a player clicks on a safe square, it will uncover that square and possibly more squares. These will also inform the player if there is a flower next to it. Aditionally, there will be Mario who runs around the board following the player's mouse (but stays bounded in the clear area), and places a turtle shell where the player right clicks (the "flag"). Also, if the player clicks on a flower, the flower will eat Mario.
    b. Similar Projects
        i. I've seen many projects that simply have the standard minesweeper game. The most prominent being the google minesweeper game that this project takes influence from.
            1. Similarities
                a. It would have the same basic structure with there being randomly generated "mines". Also, the first move would never result in a loss. Also, it would create a large opening in the board upon the first click. It has the same basic structure in terms of placing flags and clicking a mine kills you. Other similaritiies include having the different difficulties and it being a race against the clock
            2. Differences
                a. I was hoping to add a theme to my project. I wanted to have a mario who would run around the board placing the flag, and then in case of loss, have the flower eat the player and show some basic graphic of this. Also, have the board and color scheme take inspiration from Mario. Also, instead of flags I was going to have turtle shells. Another difference would be that if you misplace a turtle shell, you no longer have enough turtle shells to finish the board and this will result in a loss.
    c. Structural Plan
        i. Files
            1. Classes
                a. Classes for each tile and possibly mario and the flower
            2. Flower generation
                a. Function to place flowers
                b. Helper function to find probability for flowers at each location

    c. Helper function that determines if each locations should have a flower
3. On click
    a. Function that determines if click is safe or not
    b. Function that works if the click is safe
        i. Clears out every single block that needs to be cleared out
    c. Function that works if click is not safe
        i. Leads to losing screen
    d. Function for on first click first click
        i. Calls function to place the flowers
        ii. Place mario on the board
4. On right click
    a. Function that determines if there was a flower there or not
        i. If flower
            1. Place shell
        ii. If no flower
            1. Losing screen
5. On Start
    a. give buttons for level choices
        i. Model
            1. Store selected difficulty
        ii. View
            1. Draw buttons
        iii. Controller
            1. Let user decide what button to click
6. On level select
    a. Funcntion to initialize the board (no mines yet)
    b. Function to attach mario character to mouse
7. Losing screen
    a. Start a new app
    b. Create an animation of mario being eaten by a flower (no user input)
        i. Model
            1. Store which mario/flower image to use
        ii. View
            1. Place this image ont he board
        iii. Controller
            1. On step update this image
8. Winning screen
    a. Start a new app
    b. Animate mario jumping (no user input)
        i. Model
            1. Store which amrio image to use

                ii. View
                    1. Display the current mario image and move it up and down
                iii. Controller
                    1. Update which mario image to use on step

d. Algorithmic Plan
    i. I believe that the part of my project that will be the most algorithmically most difficult is the generation of flower locations. Other than this the only part of my project I think would be hard is determining how large of an area to clear based on the current click.
        1. The first part I will be explaining my approach on is the generation of flower locations. This will occur after the users first click. This will be done first so the user does not accidentally click a flower on their first click, but also to ensure that the first click where they "break" the board results in enough useful information so that the user could effectively complete the puzzle with consistent difficulty.
            a. The algorithm would create a empty block one square out in every direction
            b. It would then set this block as a clear space where no flowers can be generated
            c. It would then randomly generate flowers across the rest of the board
                i. This would be done by there first being a set number of flowers to generate
                    1. Determined by difficulty level
                ii. Then, I would loop through the board until all of the flowers were placed
                    1. I'd go through one block at a time
                    2. I would look at all of the blocks next to the current block
                        a. For every block next to the current block that has a flower on it, it would lower the probability that the current block gets a flower
                            i. Id do this by starting with a probability = the number of blocks/number of flowers
                            ii. Subtract some constant from this starting probability for every adjacent flower
                            iii. Then have a helper function that returned if a randomly generated number between 0

                              and 1 was lower than the probability for that block
                            iv. If the helper function returned true then place a flower
        2. The function that decides what block to show would first start at the block clicked
            a. There would be a 2d list that had a list of where there was a flower, and which spots were safe
            b. There would be a function that would take the first 2d list and determine for which of the blocks without flowers, there should be numbers (referring to there being flowers adjacent to the block) and for which of the safe blocks there should be nothing.
            c. Finally, there would be another 2d list that stored where there were flags placed by the player
            d. Whenever the player clicked a safe square, if that square had a number on it, the function would only return to show the square clicked
            e. If the square had no number on it, the function would use recursion
                i. For every numberless square, it would check the squares next to it
                    1. If the square next to it had either a number or was numberless
                        a. And did not have a flag
                        b. It would uncover it
                    2. It would then recursively go through all numberless squares
                        a. NOT NUMBERED SQUARES WITHOUT FLOWERS
e. Timeline Plan
    i. Generation of board
        1. By Nov 27, 2023
    ii. Creation of all object classes
        1. By Nov 27, 2023
    iii. Initial Graphics interface (minimum to allow some level of visualization)
        1. Dec 1, 2023
    iv. Code for letting the player interact with the game
        1. Dec 1, 2023
    v. Final Graphics Design
        1. Dec 5, 2023
    vi. Final Project (just check everything to make sure it works)
        1. Dec 6, 2023

f. Version Control Plan
    i. I’ve created a github repository and I was just going to store my files on here. I will clone this repository to my computer. Then I will work on my files locally on my computer. After this, I will push to github so that my code is stored on github and I don’t risk issues with losing code on my computer.
    ii. 

g. Module List
    i. No planned external modules

2. TP1 Update
    a. A couple changes have been made during TP1. The main structural change is that I have added a bot that solves the board and another mechanism to provide hints. The bot performs the next move that it determines to be optimal, and the hint tells you if there are any guaranteed flags on the board currently. This has resulted in some new files (one for each of the new features). Also, my timeline has changed. I have finished class creation, generation of board, initial graphics interface, and code for letting player interact with game, as well as my hints code, and I’d estimate 75% of my bot (that performs next move) code. My goal from now until tp2 is to finish my bot, make my graphics design look appealing, add in animations, and theme the game to Mario. Also, I would like to revamp my mine generation algorithm as I feel it is overly simple and leads to too many cases that require guessing.

3. TP2 Update
    a. The main change that has been made is removing the hint system and keeping only the bot. I thought it was redundant to have one system that tells the user what move to make and one that makes the move so I just simplified it to a single system that makes the next move. Also, I abandoned my plan of making the

board always solvable without guessing because I feel like this gets rid of the feel of minesweeper and waht makes the game fun.

4. TP3 update
    a. I eliminated the hint function. I tidied up my graphics and added sounds. I also added an info screen.