

Author: Ashley Batesole

Student ID: 150161899

User ID: rfasba

Email: ashley.batesole@tuni.fi

General gameplay instructions:

- The goal of the game is to locate where all the mines are and avoid setting any of them off.
- Begin the game by clicking on any of the squares. If the square contains a mine, an image of a mine will be displayed and the game will be over. If the square does not contain a mine, a number will be displayed that represents how many mines are adjacent to that square. Diagonal squares count as being adjacent, so there can be anywhere between 0-8 adjacent mines.
- A flag can be placed on a square to mark where you think a mine is. A flag must be removed from a square before it can be opened.
- The game is won either when all unopened squares only have mines left, or when all flags are placed correctly.

Other options:

- Seed Value: provides a seed value to generate the gameboard. The seed value is used to determine the location of the mines. The game initializes with a seed of 0. The user can give any integer value between 0-9999. Click update to create a new gameboard based on the seed shown in the spin box.
- Game Board Size: provides a size to generate the gameboard. The value is the length of one side of the gameboard and is initialized at 10 (so the initialized gameboard is a 10x10 grid). The user can give any integer value between 0-16. Click update to create a new gameboard based on the size shown in the spin box.
- Flag/Mouse button: click to switch between opening a square or placing a flag. The icon that is colored indicates which option is currently selected.
- Timer: displays how much time has passed (in seconds) since the game started.
- Reset button: can be clicked to reset the current game. Does not change the gameboard size or seed value.

Program structure and design decisions:

- The functional mechanics of minesweeper itself are done using the given code in the gameboard and square source and header files. Mainwindow source and header files create the gui and convert user clicks to their respective inputs in the gameboard and square files.
- A button grid is used to create the gameboard. When a button is clicked its location is converted to x,y coordinates to update the gameboard and square objects, and the

outputs from those files are then converted into an image to be displayed on the button(s). Buttons are disabled after they have been opened.

- The gameboard can be between 1-16 squares long per side. The maximum value of 16 can be changed in the mainwindow.hh file and everything still works, but having over 16 buttons made the board too big for the gui window so I capped it there.
- The seed value can be between 0-9999. These values can be changed in the mainwindow.hh file. The maximum of 9999 was chosen simply because it seemed large enough.
- The timer begins when a button is clicked on the gameboard to show an accurate score of a game. The timer stops when the game is over, and is reset when the game is reset, the seed value is updated, or the gameboard size is updated.

Additional features implemented:

- Instead of printing a character in an opening square, a suitable figure or icon will be shown. (20 p.)
- Instead of (or besides to) using a radio button to select between opening a square and setting a flag (and possibly setting a question mark), there is a single button for the same purpose. The button can be clicked in turns to make the selection. The button also somehow shows which selection has been made. (15 p.)
- The user interface provides a possibility to adjust the size of the game grid. (10 p.)
- If all unopened squares have mines, they will be opened automatically. (5 p.)

All images were created by me using microsoft paint.