Meeting 2/6/19 9:00 – 9:50 am in LEEP2 study alcove

Ben, Jeff, Jon

* Discussed plans for what language to use and packages – settled on python with pygame
* Discussed what classes and features we would need to implement minesweeper in python
  + Decided on Board, Cell, and GUI classes to get going
  + Would need to get the game logic behind the game
  + Images for mines and flags
  + Decide if we would break the classes into files or put them on one file (there is no standard for python but most put classes on one file)
* Figured out some pygame modules by researching built in functions on the pygame website
* Discussed that each of us would build a board as practice for working in a python environment
* Decided to meet Thursday after 448 lab and Friday after 448 class again to try and finish as much as possible (weather didn’t allow for Thursday’s meeting)
* Breakdown for group members was as followed: Ben works on game logic needed to implement minesweeper (general logic, not a python implementation yet), Jon would focus on python and pygame implementations including a starting a skeleton for the project, Jeff would work on documentation including logging meetings and begin work on the GUI including design elements, Thomas would begin work on other GUI elements such as the how to play instructions and showing the user they won or lost along with aiding in.
* The duties are left loose purposely as things are going to change and problems will arise.

Meeting 2/6/19 10:00 – 11:00 pm Discord video chat

Jeff, Jon, Thomas

* Ensured each member of the team made a commit to the github for the project by spelling the names wrong in a txt file and having each member fix their name.
* Made sure each member was using the same version of python and pygame

Meeting 2/8/19 9 – 9:50, 10- 10:50, 11 – 3:30 in the Fishbowl at Eaton

Ben, Jeff, Jon

* Began work on Board class
* Defined Cell class, which only has bool values for mine, revealed, and flagged cells. We initially set revealed to True so we could see if everything was working properly.
* Completed most of the ground work for Boarder class including generateGrid function, decided on a final design. Got the board centered within the window.
* Added space for a toolbar GUI above the game board.
* Decided on user interface for toolbar.
* Added a function for mouse click events, this is used for testing purposes to make sure our first board for now.
* Added in a boarder window in the tool bar.
* Defined drawboard function, uiElement, mouseClick within the GUI class
  + drawBoard prints the board to the screen and handles the colors of each cell on the board
* Defined revealCell, countNearbyMines, flagCell within the Board class
* Started a new Class within the GUI called InputBox that will be a text box in the toolbar for user input
* Began Documentation
* Started investigating how to add images to the game

Meeting 2/9/19 ~10:00 – 10:30 am GroupMe chat

Ben, Jeff, Jon, Thomas

* Discussed a minor bug with the flags coming back when the squares are revealed.
* Tried to determine if this was actually a bug or not
* Discussed the added images to the game
* Discussed what might be worked on soon

Meeting 2/11/19, Outside of 448 classroom 9:00 – 9:50 am

Ben, Jeff, Jon, Thomas

* Assigned remaining parts for each team member to work on.

Ben

* Debugging code

Jeff

* Documentation
* Pydoc (Comments that can be output to HTML file, work in command line)
* Help with prompts
* Variable updates

Jon

* Refactoring code

Thomas

* Gameover/win and play again prompts
* Help with pydoc
* How to play instructions

2/12/19, LEEP2 ground floor alcove 9:20 – 10:00 am

Jeff and Thomas

* Went over commenting code
* Shot around ideas for Game Over/win and play again prompt
* Floated around ideas for How to play instruction
* Jeff designed a small splash screen that displays when a Game Over, win or loss, is reached
* Thomas aided and started implementation on click box for the user to play again

2/13/19, Discord Video Chat 4:00 – 5:30 pm

Jon and Thomas

* Implemented Play Again/Quit buttons
* Added inputbutton subclass
* Added Minesweeper caption to the game window

2/14/19, Computer Lab during normal EECS 448 Lab hours 12:30 – 2:15 pm

Ben, Jeff, Jon, Thomas

* Refactored code, made things easier to read, more friendly for others who may not understand python
* Added some pydoc formatting
* Added Help button and to show the user how to play
* Added icons (unused at this time)
* Bug fixes, general quality of life additions

2/14/19, Discord Chat 10 pm – 11:30

Jeff and Jon

* Refactored a whole lot of the code in pysweeper.py
* Refactored even more code in GUI.py
* Jeff – played with adding sounds and did in fact add sounds on a separate branch.

2/15/19, Eaton Fishbowl 8:15 – 9:50 am

Ben and Jeff

* Added a lot of the pydoc documentation
* Generated HTML files for the pydoc documentation

Final workload split:

Benjamin

* Minesweeper game logic as seen in the Board class
* Help with python implementation in the main pysweeper.py, the Board class, the GUI class, the InputBox class, InputButton class, and cell class.
* Fix bugs with game logic and user input
* Aid with design elements
* Aid with GUI elements
* Aid with text and general quality of life additions
* Refactor portions of code
* Aid with pydoc, generating HTML from pydoc, and comments

Jeff

* Keep a log of all meetings and outcomes
* Write the retrospective write up
* Contact Klein or Oqi as needed with questions or concerns
* Find Bugs, try to break the game
* Add and implement images
* Implement portions of pysweeper.py and the GUI class
* Aid with deisgn and GUI elements such as drawing boxes, mouse events, and text
* Refactor portions of code
* General documentation, pydoc, comments

Jon

* Outline project and what will be needed to finish
* Main game loop, mouse events, drawing boxes, and other pysweeper.py implementations
* Python implementations of the GUI class, InputBox class, UIElement class, InputButton class
* Fix Bugs
* Refactor code, a whole lot of refactoring
* Implement further design elements for quality of life
* Help button, how to play instructions
* Aid with documentation
* Testing the code

Thomas

* Game Over splash screen
* Winner splash screen
* Quit and Restart Buttons
* Aid with InputButton class
* Aid with finding bugs
* Aid with design elements
* Refactoring code
* Note: everyone contributed all over each part of the project. From influencing design of the code, UI, features to add, or direction to go in everyone had a hand in every portion. Many parts were done as a group, some parts were done by specific members but for the most part everyone used their strengths to make the project the best it could be.

Challenges:

Assigning work evenly, compromises, snow days and general meeting as a group

Repetitive code, bugs, refactoring code introduced bugs, too much recursion (falls under bugs), pygame can be confusing (this could go with repetitive code and refactoring?)

Features that didn’t make the final product:

Box around the toolbar, middle mouse, flags clickable, sounds? maybe, difficulty buttons

What we would have done differently

Not written the code in mostly one Friday afternoon, had everyone present at the initial meetings? Probably wont go here as it feels like placing blame on others, learned our strengths and desires for the program faster, much more planning (avoid spaghetti code) so outline everything rigorously to prevent consistent need to refactor code (spaghetti code)

Honestly, that’s all tied together.