**Monday Lab Python Team**

Owen Mellema, Zach Davis, Zach Freund

Gwen Liu, Aris Vinsant, Guanyu Li

Team Meetings:

Feb 8, 2019

* Location-Leep2
* Members: Zach D., Owen M., Guanyu L., Gwen L
* Assigned member tasks based on amount of experience with Python language
* Board design: Owen
* GUI: Zach D.: start design of physical board
* Game logic: Zach F.
* Testing: Guanyu, Gwen: learn python, debug project files

Feb 11, 2019

* Location-Leep2, rm 2328
* Members: Zach D., Zach F., Owen M., Guanyu L., Gwen L., Aris V.
* New assignment: GUI: Aris V.
* Tasks completed
* GUI: decide on size of the board, set side length of tiles, color of bombs/squares, flags, click time delay, text, menu/title display
* Game logic: use integer division, decide on which functions need to be added for tile class
* Testing: check errors and documentation of files
* Unfinished tasks: bomb randomization(initialize), image files, linking implementation with GUI, left click(check to reveal recursively, if hit bomb), right click(set dictionary), set bool win/lose, check number of tiles

Feb 14, 2019

* Location: Leep2, Groupme messaging
* Members: Aris V., Guanyu L., Zach D., Gwen L., Owen M.
* Finished tasks:
* Created numbers displaying surrounding bombs
* Added explosion animations to game over
* Added gameWon() situation

Feb 15, 2019

* Location: Leep2
* Members: Guanyu L., Zach D., Zach F., Gwen L., Owen M.
* Finished tasks:
* Fixed board dimensions and text scaling
* Fixed bug causing game to crash
* Unfinished: setting user input for board size and number of bombs

Feb 17, 2019

* Groupme messaging
* Members: Zach D., Zach F., Owen M., Guanyu L., Gwen L., Aris V.
* Final review of code and debug

Tasks:

Work was divided based on people's knowledge of python. Since our group was a little bigger due to some miscommunication a lot of us worked on the same parts of the program. A couple people already knew python so they went ahead and started writing parts of the program while other members tested the changes being made while learning python themselves and analyzing the code that was being written. Pygame was new to everyone in the group, so we all had to learn and implement as we went. Because of this everyone participated in the implementation of Pygame and making final changes/tweaks to existing code.

Challenges:

Some members of the group had to learn python from scratch while writing code. This was a bit challenging because learning a new language always takes a little bit of time. Luckily python is a fairly user-friendly language so everyone caught on pretty quick. Another issue we ran into was implementing pygame for our GUI. We all had to take the time to look over pygame’s documentation and figure out how we were going to implement it into our project. A non-technical challenge we faced was trying to get everyone together for meetings; everyone has different schedules and we had 6 people to try and get together so it was fairly challenging.

Another challenge we faced was learning how to effectively use github and collaborate appropriately so that we were making sure changes were pushed correctly. Although we have now discussed our version control systems using git and github during the lecture portion, this was the first time a few of the group members had actually used github to work on a project. Making sure everyone understood the commands and their appropriate use was a very valuable challenge that we were able to overcome.

Features:

These are some things we wanted to add but are not in the demo:

* Different colored numbers like the classic minesweeper
* Menu bar displaying the time and also how many bombs are left
* A restart button for after you win/lose

Retrospective:

There was not a ton we wished we could have done differently. We were very proud of what we were able to accomplish in the amount of time that we had. The distribution of who wrote the most amount of code was uneven, but that was at least in some part indicative of how we decided to split up given everybody’s individual skill level and comfortability in python and pygame.

We thought we made a great decision selecting python and pygame to build our minesweeper game because by the end of the project, we could all say that we learned tons of new information and we all have a new tool at our programming disposal. Pygame was a great platform because it was difficult enough that we all had to learn, but not so difficult that a successful project was out of our grasp.

Our team did a good job keeping up with how the project was coming along and contributing where we could to advance the project. We wish we would have had more time to implement what is outlined in the “Features” section, but overall we are very content with what we have created.