Minesweeper

A Command Line based ASCII Game

I grew up playing the traditional Microsoft version of the Minesweeper game; alas, the MacBook does not come with this game pre-installed, so I decided to program my own. The gameplay is really simple — First, you select a difficulty level; then you select a cell by inputting its position and lastly you decide whether to put a flag at the cell in interest. This manual assumes that the reader is familiar with the minesweeper rules, if not click this.

In order to enjoy this game, direct to the folder with the `minesweeper.py` script using command line. And then you fire up the game by typing the following command —

```
python minesweeper.py
```

This command should start the game and prompt user to insert his/her name like —

After typing your name, press enter —

Now, select the difficulty level with 1 being easiest and 3 being hardest —

Wow! Look you have a nice grid now! From here, you can start playing the game by selecting a cell from the command like by inserting the row and the column. Also you have to decide whether to put a flag at this cell or not.

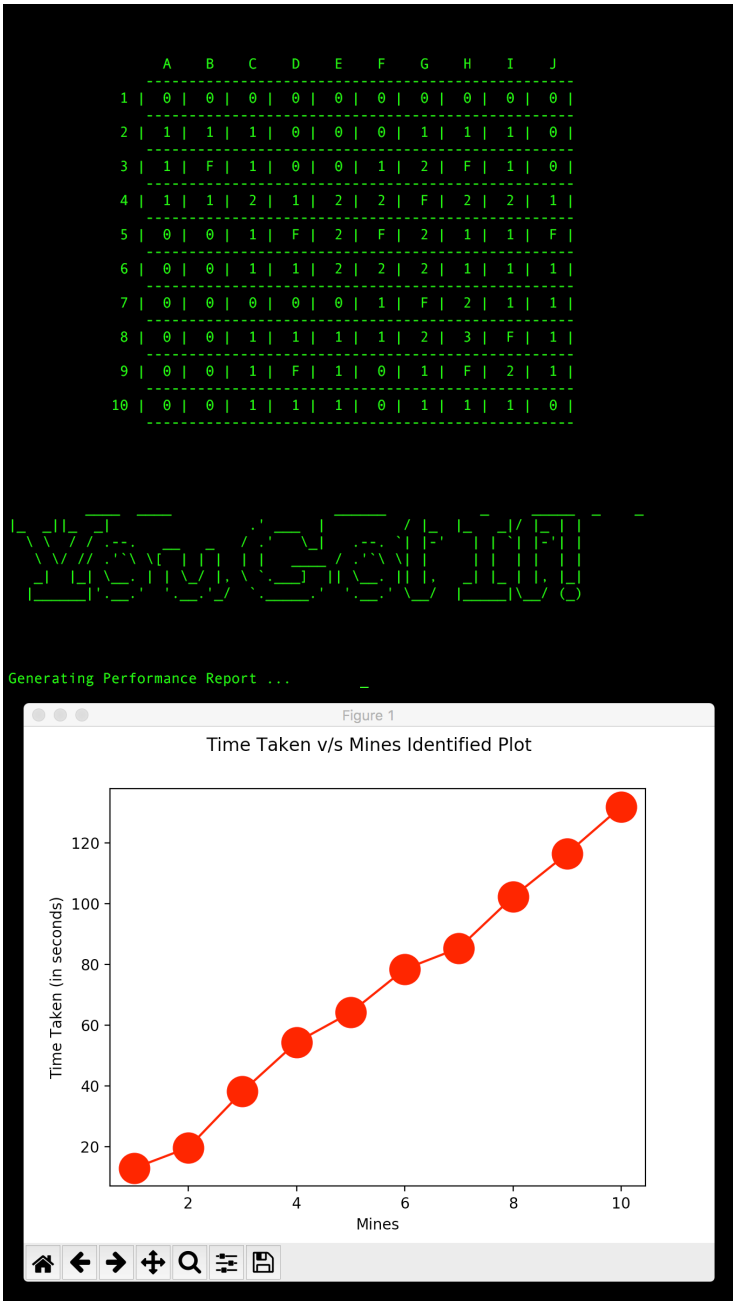
Let's go for row 5 and column E in our case —

```
Type ROW index : 5
{'rows': '[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]'}
Type COLUMN index : E
Do you want to place/remove FLAG at (5, E)? (y/n) : N
```

	A	B	C	D	E	F	G	H	I	J
1										
2			1	1	1	1	1	1	1	
3			1	0	0	0	0	0	1	1
4			1	0	0	0	1	1	1	0
5			1	0	0	0	2		2	0
6			1	1	1	0	2		2	0
7					1	0	1	1	1	0
8					1	0	0	0	0	0
9					1	0	0	0	0	0
10					1	0	0	0	0	0

Looks like 5E was really a good choice! Now by following the traditional minesweeper rules, we can go ahead and finish this game. Be right back!

Phew! That took a while. Finally, we have —



Wow! What's that graph? That's something new! That's right! For my advanced feature, I added a performance report to summarize your moves. How cool is that?!

Complexity

Given that CS10 introduced me to the exciting world of programming, I wanted to end my course journey with an exciting, yet challenging, project. Even though we covered Python briefly, it was brought to my knowledge that it is much widely used language over Snap! So I decided to implement my final project in Python, even though I was relatively well versed in Snap!

Before I started writing any code, I made a rough sketch and logic flow on a piece of paper. As I was doing this, I realized programming is nothing more that taking a complex problem and dividing it into smaller chunks and solving those smaller chunks one at a time. For this purpose, functions came handy. Bringing everything together made me realize importance of a good naming scheme and comments.

Since, I was making a minesweeper game, I had to create two game boards — one for internal bookkeeping and one for displaying a grid on to the screen. For this I used 2 two-dimensional arrays. Making changes to one based on user input was a bit challenging, so I had to do some sanity checking (turns out it is not that easy!).

Even though I feel that this project was a bit too challenging and time consuming during the finals week, the journey in itself was quiet eventful and I realized how time flies when you are coding! :)

FAQs

1) How did you implement this game?

Well, on the abstract level I created two 2-dimensional arrays to fulfill the purpose of having a grid/board. One of them is kept hidden from the user as it contains the solution to a given gameplay and the another one is used for display purposes, the one user sees.

2) Which version of Python you wrote this for?

Good question! Python 2.7.13 can support this game.

3) Any Python packages you have used in your project.

Yes, 3 of them. All 3 packages are very common and can be installed by simply doing pip install.

- matplotlib
- random
- time

4) Have you tested this game?

Of course! I made a manual test plan to extensively test this game.

5) Have you implemented any advanced functionality?

Yes! I added a performance report generation functionality which was out of the scope of CS10. For this, after doing a lot of research, matplotlib served as the library to get this feature working. It is pretty cool, have a look!

6) What did you use for code versioning?

GitHub

Github is my best friend! My repo is on — https://github.com/divyamguptaedu

7) Is there a video demo available for this game?

I am glad you asked! Yes, there is. It's on YouTube and you can check it by clicking this.