Programming Fundamentals Using Python

2018

Problem Set 4

Most recent updated: July 12, 2018

Objectives

1. Understand what is object-oriented programming (OOP).
2. Learn classes and methods.

Note: Solve the programming problems listed using your favorite text editor. Make sure you save your programs in files with suitably chosen names, **and try as much as possible to write your code with good style (see the style guide for python code)**. In each problem find out a way to test the correctness of your program. After writing each program, test it, debug it if the program is incorrect, correct it, and repeat this process until you have a fully working program. Show your working program to one of the cohort instructors.

Problems: Cohort sessions

1. *Classes and Methods: Make a Coordinate class.* Implement a class named `Coordinate` that represents a coordinate of a point in two dimensional space. The class has two attributes: x and y. It also has several methods:

   - `__init__(self, x=0, y=0)` : This method takes in x and y to initialize the attributes. If x and y are not provided, the attributes are initialized to 0.
   - `magnitude(self)`: This method returns the magnitude, which is defined as $\sqrt{x^2 + y^2}$.
   - `translate(self, dx, dy)`: This method translates the coordinate by `dx` and `dy` so that it now represents the coordinate `(x + dx, y + dy)`.
   - `__eq__(self, other)`: This method takes another coordinate object and returns `True` or `False` depending on whether this coordinate specifies the same point in space as the other.

   A sample interactive session using class `Coordinate` is shown below.

```
>>> p = Coordinate()
>>> print(p.x, p.y)
0 0
>>> print(p.magnitude())
0.0
>>> p.x = 3
>>> p.y = 4
>>> print(p.magnitude())
5.0
>>> q = Coordinate(3,4)
>>> print(p == q)
>>> True
>>> q.translate(1, 2)
>>> print(q.x)
4
>>> print(p == q)
>>> False
```

2. *Classes: Fraction* Implement a class named `Fraction`. A fraction has two parts: numerator and denominator. Note that the denominator must be greater than zero. Note that the numerator and denominator are stored in the smallest terms. Implement a method called `reduce` without any argument to reduce the current fraction to its simplest terms. The class support the following math operations: add, subtraction, and multiplication. It should also support the relational operators: equal, less than, and less than equal. Moreover, the class should support `__str__` method to display the fraction as `numerator/denominator`.

```
>>> f1 = Fraction(1,4)
>>> f2 = Fraction(1,2)
>>> f3 = f1 + f2
```

```
>>> print(f3)
3/4
>>> f4 = Fraction(6,8)
>>> print(f4)
3/4
```

3. *Classes: Minesweeper* Create a console based Minsweeper game. You should use object oriented design to implement this game. The program should prompt users for the commands to be executed. There are three kinds of commands: 1) open, 2) flag, 3) quit. The quit command is executed by typing `quit` in the prompt. The open and flag command has the following syntax `[o`f][row][col]—. For example, to open row A col 3, you type:

   ```
   oa3 # or OA3, it should be case insensitive
   ```

   and to flag row B col 31, you type:

   ```
   fB31 # or Fb31, it should be case insensitive
   ```

End of Problem Set 4.

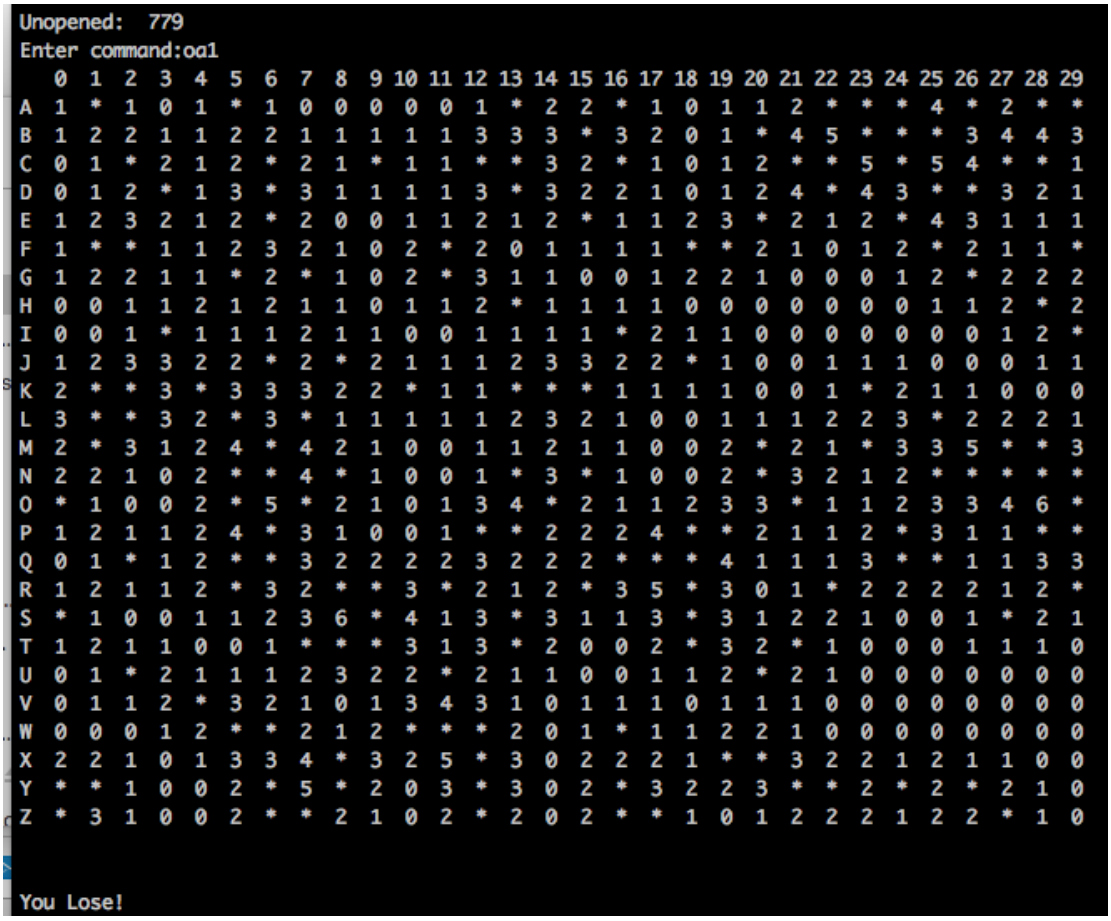

Figure 1: State of game when lost with 26 rows and 30 cols.

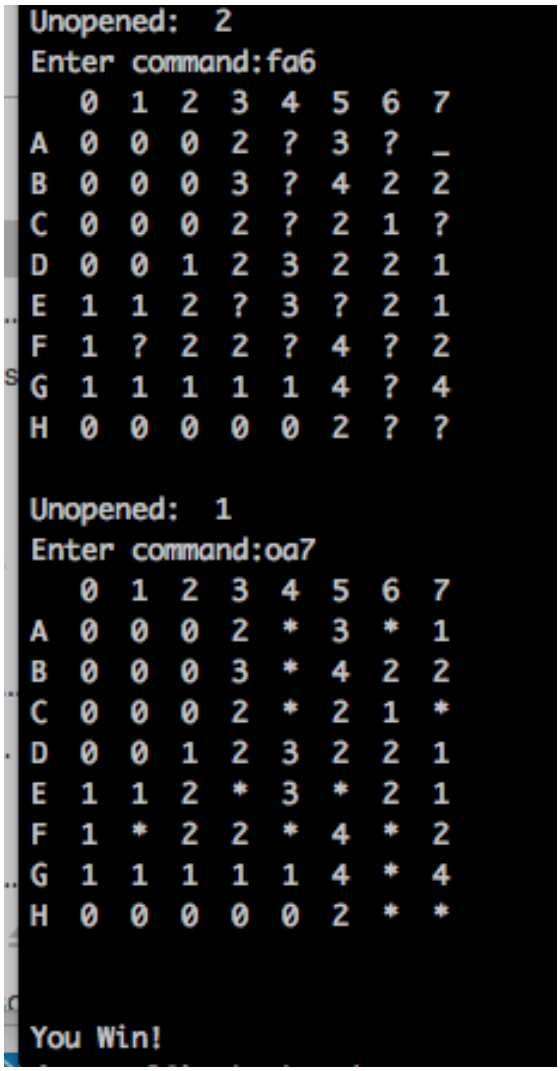

Figure 2: State of game when lost with 8 rows and 8 cols.