
CMSC 201 Spring 2016

Project 2 – Recursive Word Search

Assignment: Project 2 – Recursive Word Search
Due Date: Wednesday, May 4th, 2016 by 8:59:59 PM
Value: 80 points

Project 2 is the second and final project assignment in CMSC 201. As in Project 1, you will get the chance to make your own decisions about how you want your program to handle things, what its functions should be called, and how you want to go about designing it.

Project 2 will also be ***substantially longer*** than any of the single homework assignments you’ve completed so far, so make sure to take the time to plan ahead, and don’t do any “cowboy” coding!

Remember to enable Python 3 before you run your programs:
`sc1 enable python33 bash`

Instructions

For this assignment, you'll need to follow the class coding standards, a set of rules designed to make your code clear and readable. The class coding standards are on Blackboard under “Course Documents” in a file titled “CMSC 201 - Python Coding Standards.”

You should be commenting your code, and using constants in your code (not magic numbers or strings). You should also have a function header comment for every function that is not `main()`!

Re-read the coding standards!

You will **lose major points** if you do not following the 201 coding standards.

A very important piece of following the coding standards is writing a complete **file header comment block**. Make sure that your file has a comment block at the top (see the coding standards document for an example).

NOTE: Your filename for this project **must** be `proj2.py`

NOTE: You must use `main()` in your file.

Details

The purpose of this assignment is to give you practice with recursion, using two-dimensional lists, and to do some file handling where it's necessary to detect the end of the file or end of the list. You'll also be getting some experience with using datasets of a variable size. As always, you should continue to practice detailed project design (prior to coding!) and good implementation techniques like incremental programming.

A popular form of puzzle is known as the "word search". Besides being good entertainment when waiting at the airport or at the doctor's office, this type of puzzle is often used to help young children build vocabulary and practice spelling.

A word search is a 2-dimensional grid or matrix of letters which contains "hidden words". The person working the puzzle is given a list of words that are hidden in the matrix and is asked to locate and circle them. The fun part is that the words may appear horizontally, vertically or diagonally in the grid. Horizontal words may be written left-to-right or right-to-left. Vertically oriented words may be written top-down or bottom-up. Similarly for diagonally oriented words.

Puzzles will be any size but they will always be rectangular.

Here's a simple word search puzzle for you. (The grid below is 12 x 12.)

```
G J T P B A V K U V L V
M N Q H S G M N T C E E
Y H I J S G Q E N Y C W
G S K M G H C B M U T H
R A T V M N V D G V U T
E P G U E A B P W Q R T
T J C I D D R Q T E E C
U P C I S E N G B U O B
P S J C I V N F O U N N
M P R O J E C T R R A M
O H Q T P P D S H A P G
C O W U K Q E G I J M S
```

Here is a simple word list for the above word search puzzle:

COMPUTER
SCRAM
COURSE
LECTURE
PROGRAMMING
PROJECT
SCIENCE
STUDENT
UMBC

Some additional requirements:

1. The game board may be of any size – as long as it is rectangular.
2. The game board will consist of letters separated by spaces.
3. There can be any number of words on the list (there will always be at least one word, however).

For your output:

4. If a word is not in the puzzle, the user should be notified.
5. If a word is in the puzzle, the starting coordinate should be shown, as well as the direction that the word is going based on the direction listed below. (See the sample output for some examples.)

Diagonally (up and left)	Up	Diagonally (up and right)
Backwards (left)	WORD STARTS HERE	Right
Diagonally (down and left)	Down	Diagonally (down and right)

6. **You must use recursion for the search function that looks for the word in the puzzle.** Failure to use recursion will result in a significantly reduced grade.

Sample Output – General Gameplay

Here is some sample output of a game being played, with the user input in blue.

```
bash-4.1$ python proj2.py
Welcome to the Word Search
For this, you will import two files: 1. The puzzle board, and 2.
The word list.
What is the puzzle file would like to import?: p.txt
What is the word list file you would like to import?: w.txt
The word COMPUTER starts in 11 , 0 and goes up
The word SCRAM does not appear in the puzzle.
The word COURSE starts in 6 , 11 and goes diagonally down and
left
The word LECTURE starts in 0 , 10 and goes down
The word PROGRAMMING starts in 10 , 10 and goes diagonally up
and left
The word PROJECT starts in 9 , 1 and goes right
The word SCIENCE starts in 8 , 1 and goes diagonally up and
right
The word STUDENT starts in 3 , 1 and goes diagonally down and
right
The word UMBC starts in 3 , 9 and goes backwards left
```

Submitting

Once your Project 2 is complete, it is time to turn it in with the `submit` command.

Don't forget to complete the header block comment for your file! Make sure that you updated the header block's file name and description.

You must be logged into your GL account, and you must be in the same directory as the Project 2 file. To double check this, you can type `ls`.

```
linux1[3]% ls
proj2.py
linux1[4]% █
```

To submit your files, we use the `submit` command, where the class is `cs201`, and the assignment is `PROJ2`. Type in (all on one line) `submit cs201 PROJ2 proj2.py` and press enter.

```
linux1[4]% submit cs201 PROJ2 proj2.py
Submitting proj2.py...OK
linux1[5]% █
```

If you don't get a confirmation like the one above, check that you have not made any typos or errors in the command.

You can **double-check that your file was submitted** by using the `submitls` command. Type in `submitls cs201 PROJ2` and hit enter.

And you're done!