COMP 1039

Problem Solving and Programming

Programming Assignment 2

UniSA STEM
The University of South Australia
May, 2023

Contents

INTRODUCTION

This document describes the first assignment for Problem Solving and Programming.

The assignment is intended to provide you with the opportunity to put into practice what you have learnt in the course by applying your knowledge and skills to the implementation of a game called **Wordle.**

This assignment is an **individual task** that will require an **individual submission**. If you are an **internal student,** you will be required to submit your work via learnonline before **Tuesday 13 June (swot-vac), 10 am.** Internal students are **not** required to demonstrate their work in person.

This document is a kind of specification of the required end product that will be generated by implementing the assignment. Like many specifications, it is written in English and hence will contain some imperfectly specified parts. Please make sure you seek clarification if you are not clear on any aspect of this assignment.

ASSIGNMENT OVERVIEW

My Wordle!

You are required to write a Python program that allows a user to play a game called **My Wordle!** Yes, you guessed it, a text-based version of the popular game developed by Josh Wardle called… **Wordle** :) Wordle is a web-based word game created and developed by Josh Wardle (a software engineer). Josh originally created the game for just himself and his partner to play and called it Wordle (a play on his last name). Josh released the game to the public in October 2021. The game soared in popularity and was purchased in 2022 by The New York Times Company for a seven-figure sum! In the original web-based version of the game, players have six attempts to guess a five-letter word, with feedback given for each guess in the form of coloured tiles indicating how close the guess was to the word. Feedback comprises of a green square to indicate the letter is in the word and in the correct spot, a yellow square to indicate the letter is in the word but in the wrong spot and a grey square to indicate the letter is not in the word in any spot.

For this assignment, you are required to implement a text-based version of the game called My Wordle. The program allows the user to repeatedly play the game of My Wordle until the user chooses to stop guessing/playing. Once the user chooses to stop playing, the program will report the game play statistics to the screen.

You may like to read about Wordle here: https://en.wikipedia.org/wiki/Wordle.

You may like to play the NY Times web-based game here: https://www.nytimes.com/games/wordle/index.html.

Please ensure that you carefully read the section titled 'Assignment Specification' below for further details.

GRADUATE QUALITIES

By undertaking this assessment, you will progress in developing the qualities of a University of South Australia graduate.

The Graduate qualities being assessed by this assignment are:

- The ability to demonstrate and apply a body of knowledge (GQ1) gained from the lectures, workshops, practicals and readings. This is demonstrated in your ability to apply problem solving and programming theory to a practical situation.

- The development of skills required for lifelong learning (GQ2), by searching for information and learning to use and understand the resources provided (Python standard library, lectures, workshops, practical exercises, etc); in order to complete a programming exercise.

- The ability to effectively problem solve (GQ3) using Python to complete the programming problem. Effective problem solving is demonstrated by the ability to understand what is required, utilise the relevant information from lectures, workshops and practical work, write Python code, and evaluate the effectiveness of the code by testing it.

- The ability to work autonomously (GQ4) in order to complete the task.

- The use of communication skills (GQ6) by producing code that has been properly formatted; and writing adequate, concise and clear comments.

- The application of international standards (GQ7) by making sure your solution conforms to the standards presented in the Python Style Guide slides (available on the course website).

ASSIGNMENT SPECIFICATION – MY WORDLE!

You are required to write a Python program called *yourEmailId*_my_wordle.py that allows a player to play a game called **My Wordle** (a variation of the popular NY Times game called Wordle).

My Wordle!

You are be required to write a Python program that allows a player to play a game called My Wordle! My Wordle is a text-based version of the game Wordle (NY Times). The program allows the user to repeatedly play the game of My Wordle until the user chooses to stop guessing/playing. Once the user chooses to stop playing, the program will report the game statistics to the screen. You may like to read about Wordle here: https://en.wikipedia.org/wiki/Wordle. You may like to play the NY Times web-based game here: https://www.nytimes.com/games/wordle/index.html.

We will be adhering to the following 'My Wordle' rules and game play for the assignment.

My Wordle Game Play and Rules:

The user must guess the Wordle in 6 tries. Each guess must be a valid five-letter word. After each guess the program will show you how close your guess was to the word.

- To begin, the following text is displayed to the screen and the user is asked whether they would like to play My Wordle, i.e.:

```
File     : wayby001_my_wordle.py
Author   : Batman
Stud ID  : 0123456X
Email ID : wayby001
This is my own work as defined by the University's Academic Misconduct Policy.


---------------------------------
--         My Wordle!          --
-- Guess the Wordle in 6 tries --
---------------------------------


Would you like to play My Wordle [y|n]?
```

If the user enters 'n', the following message is displayed to the screen only:

```
No worries... another time perhaps... :)
```

If the user enters 'y', game play continues as normal.

- Game play is as follows:
- The following text showing the number of blank letters in the word is displayed to the screen and the user is asked to guess the wordle, for example:

```
-------------
| - - - - - |

Please enter your guess - attempt 1:
```

- After each guess, the program provides feedback as to how close the user's guess was to the word:
    - For every letter in the user's guess that is in the correct position, the program displays a caret (^) symbol under the letter, for example:

```
Please enter your guess - attempt 1: woods

-------------
| w o o d s |
| ^ - - - - |
|
| Correct spot(^): 1
| Wrong spot(*):   0
```

    - For every letter that appears in the word but is not in the correct position, the program displays an asterisk (*) under the letter, for example:

```
Please enter your guess - attempt 1: times

-------------
| t i m e s |
| - - - * - |
|
| Correct spot(^): 0
| Wrong spot(*):   1
```

    - For every letter in the user's guess that doesn't appear in the word, the program displays a dash (-) under the letter, for example:

```
Please enter your guess - attempt 1: pumps

-------------
| p u m p s |
| - - - - - |
|
| Correct spot(^): 0
| Wrong spot(*):   0
```

- If the user does not guess the correct word, the user is prompted to guess again. Game play continues until the user either guesses the correct word (within six attempts) or fails to guess the correct word (i.e. does not guess the word within six attempts).

- If the user guesses the word, the following message is displayed to the screen:

```
Please enter your guess - attempt 1: relic

-------------
| r e l i c |
| ^ ^ ^ ^ ^ |
|
| Correct spot(^): 5
| Wrong spot(*):   0
|
| Correct letters: r e l i c
| Used letters:


Solved in 1 tries!  Well done!
```

- If the user does not guess the correct word after six attempts, the following message is displayed to the screen:

```
Please enter your guess - attempt 6: smoke

-------------
| s m o k e |
| - - - - * |
|
| Correct spot(^): 0
| Wrong spot(*):   1
|
| Correct letters: e r
| Used letters: t a m s w k f o g


Oh no! Better luck next time!

The wordle was: relic
```

- The user is then asked whether they would like to play again with the following prompt (seen below). Game play continues while the user enters 'y' at the prompt:

```
Would you like to play again [y|n]?
```

- Once the user chooses to quit (after having played at least one game), the game summary is displayed to the screen, for example:

```
My Wordle Summary
=================

You played 2 games:
  |--> Number of wordles solved:   1
  |--> Number of wordles unsolved: 1


Thanks for playing!
```

You do not have to worry about defining a list that contains the five-letter words available for this assignment. A module containing a list with the five-letter words for use in this assignment has been provided for you. You are required to use this as part of this assignment, however, **please do not modify the `words.py` file.**

PRACTICAL REQUIREMENTS

It is recommended that you develop this part of the assignment in the suggested stages.

It is expected that your solution MUST include the use of:

- Your solution in one file called *`yourEmailId`*`_my_wordle.py`.
- Appropriate and well constructed `while` and/or `for` loops (as necessary).
- Appropriate `if, if-else, if-elif-else` statements (as necessary).
- The supplied `word_file.txt`. This contains the words that will be used for this assignment and is provided for you – **please DO NOT modify this file**.
- The use of the `random.choice(word_list)` function in order to randomly choose a word from the list `word_list` (word_list list is provided in stage 1).
- The use of a **list** in order to store the feedback provided to the user (i.e. `^`, `*` or `-`).
- The use of **lists** or **strings** in order to keep track of the correct and incorrect letters used.
- The use of a loop(s) to determine how many letters the user has in the correct position.
- The use of a loop(s) to determine how many letters the user has in the incorrect position.
- The following three functions (refer to stage 13 for the description of functions):
  - `display_details()`
  - `get_wordle_guess()`
  - `create_word_list()`
- Output that **strictly** adheres to the assignment specifications. If you are not sure about these details, you should check with the 'Sample Output' provided at the end of this document or post a message to the discussion forum for clarification.
- Good programming practice:
  - Consistent commenting, layout and indentation. You are to provide comments to describe: your details, program description, all variable definitions, and every significant section of code.
  - Meaningful variable names (no single letter identifier names).

Your solution **MAY** make use of the following:

- You may make use of the `print(), input(), len()` and `range()` built-in functions.
- You may make use of the `list.append()` and `list.remove()` methods.
- In the `create_word_list()` function only, any List or String methods as appropriate.
- The keywords `in` and `not in` as appropriate.
- Access the individual elements in a list or string with an index (one element only). i.e. `list_name[index]`.

Your solutions **MUST NOT** use:

- Built-in functions (other than those specified above).
- List or String methods (other than those specified above).
- `break`, or `continue` statements in your solution. **Do not** use the `quit()` or `exit()` functions or the `break` or `return` statements (or any other techniques) as a way to break out of loops. Doing so will result in a significant mark deduction.
- The enumerate() function.
- List comprehensions.
- Dictionary to store data items.

PLEASE NOTE: You are reminded that you should ensure that all input and output conform to the specifications listed here; if you are not sure about these details you should check with the sample output provided at the end of this document or post a message to the discussion forum in order to seek clarification.

Please ensure that you use Python 3.11.3 or a later version (i.e. the latest version) in order to complete your assignments. Your programs **MUST** run using Python 3.11.3 (or latest version).

STAGES

It is recommended that you develop this part of the assignment in the suggested stages. Many problems in later stages are due to errors in early stages. **Make sure you have finished and thoroughly tested each stage before continuing.**

The following stages of development are recommended:

Stage 1

You will need the `word_file.txt` file for this assignment. This has been provided for you. Please download this file from the course website (Assessments tab) and ensure that it is in the same directory as the *`yourEmailId`*`_my_wordle.py` file. The `word_file.txt` file contains the five letter words that will used to create a list of words used for this assignment. In the first instance, let's define a list of words as seen below. In stage 13, we will read the words contained in the file (word_file.txt) in order to create a list of words (but more about that later).

For now, we will use the following word list:

```
word_list = ["woods", "times", "pumps", "relic", "smoke", "cards", "slate", "tonic",
             "solid", "speak", "wants", "angle", "adapt", "apple", "smart", "gates",
             "water", "hagle", "honey", "other", "table", "fakes", "sully", "flesh",
             "sheer", "tense", "tease", "drake", "rains", "stone", "phone", "flown",
             "glass"]
```

Test to ensure that this is working correctly by entering the following in your *`yourEmailId`*`_my_wordle.py` file:

```
import random

word_list = ["woods", "times", "pumps", "relic", "smoke", "cards", "slate", "tonic",
             "solid", "speak", "wants", "angle", "adapt", "apple", "smart", "gates",
             "water", "hagle", "honey", "other", "table", "fakes", "sully", "flesh",
             "sheer", "tense", "tease", "drake", "rains", "stone", "phone", "flown",
             "glass"]

wordle = random.choice(word_list)

print("Wordle is:", wordle)
```

Run the *`yourEmailId`*`_my_wordle.py` file. If this is working correctly, you should now see the following output in the Python shell when you run your program:

```
Wordle is: cards
```

Note, this is for developmental purposes only, and you will need to modify and correctly position the above code... and eventually remove the display of wordle to the screen. Please also note that the word displayed to the screen when you run your program may be different as it randomly chooses a word.

Make sure the program runs correctly. Once you have that working, back up your program. *Note: When developing software, you should always have fixed points in your development where you know your software is bug free and runs correctly.*

Stage 2

Add code to display the following text showing the number of blank letters in the word to the screen and prompt for and read the user's guess. Display the user's guess to the screen, for example:

```
Wordle is: slate

-------------
| - - - - - |

Please enter your guess - attempt 1: tests

You guessed: tests
```

Stage 3

Now let's add a play again loop that loops until the user enters 'n' (to stop playing the game). Think about where this code should go – what needs to be repeated, etc. For example:

Sample output 1:

```
Would you like to play My Wordle [y|n]? y

Wordle is: tonic

-------------
| - - - - - |

Please enter your guess - attempt 1: water

You guessed: water


Would you like to play again [y|n]? y

Wordle is: cards

-------------
| - - - - - |

Please enter your guess - attempt 1: solid

You guessed: solid


Would you like to play again [y|n]? y

Wordle is: speak

-------------
| - - - - - |

Please enter your guess - attempt 1: wants

You guessed: wants


Would you like to play again [y|n]? n


Thanks for playing!
```

Stage 4

Add code to let the user know how many letters they have that are in the word and in the correct position. Think about where this code should be placed (inside the while loop we added in stage 3). For example:

Sample output 1:

```
Wordle is: angle

-------------
| - - - - - |

Please enter your guess - attempt 1: adapt

You guessed: adapt

-------------
| a d a p t |
| ^         |
```

Sample output 2:

```
Wordle is: angle

-------------
| - - - - - |

Please enter your guess - attempt 1: apple

You guessed: apple

-------------
| a p p l e |
| ^     ^ ^ |
```

Stage 5

Add code to let the user know how many letters they have that are in the word but in the wrong position. For example

Sample output 1:

```
Wordle is: angle

-------------
| - - - - - |

Please enter your guess - attempt 1: smart

You guessed: smart

-------------
| s m a r t |
|     *     |
```

Sample output 2:

```
Wordle is: angle

-------------
| - - - - - |
```

```
Please enter your guess - attempt 1: gates

You guessed: gates

-------------
| g a t e s |
| * *   *   |
```

Stage 6

Add code to let the user know how many letters do not appear in the word at all. For example

Sample output 1:

```
Wordle is: angle

-------------
| - - - - - |

Please enter your guess - attempt 1: droop

You guessed: droop

-------------
| d r o o p |
| - - - - - |
```

Sample output 2:

```
Wordle is: angle

-------------
| - - - - - |

Please enter your guess - attempt 1: haggle

You guessed: hagle

-------------
| h a g l e |
| - * ^ ^ ^ |
```

Stage 7

Add code to count and display how many letters were in the correct spot and how many were in the wrong spot. For example

Sample output 1:

```
Wordle is: speak

-------------
| - - - - - |

Please enter your guess - attempt 1: peaks

-------------
| p e a k s |
| * * * * * |
|
| Correct spot(^): 0
| Wrong spot(*):   5
```

Sample output 2:

```
Wordle is: honey

-------------
| - - - - - |

Please enter your guess - attempt 1: other

-------------
| o t h e r |
| * - * ^ - |
|
| Correct spot(^): 1
| Wrong spot(*):   2
```

Well done on getting this far… there is still a little way to go… :) Your output should now look something like this:

Sample output 1:

```
Would you like to play My Wordle [y|n]? y

Wordle is: tepid

-------------
| - - - - - |

Please enter your guess - attempt 1: table

You guessed: table

-------------
| t a b l e |
| ^ - - - * |
|
| Correct spot(^): 1
| Wrong spot(*):   1


Would you like to play again [y|n]? y

Wordle is: sully

-------------
| - - - - - |

Please enter your guess - attempt 1: honey

You guessed: honey

-------------
| h o n e y |
| - - - - ^ |
|
| Correct spot(^): 1
| Wrong spot(*):   0


Would you like to play again [y|n]? y

Wordle is: plums

-------------
```

```
| - - - - - |

Please enter your guess - attempt 1: punts

You guessed: punts

-------------
| p u n t s |
| ^ * - - ^ |
|
| Correct spot(^): 2
| Wrong spot(*):   1


Would you like to play again [y|n]? n


Thanks for playing!
```

Stage 8

Add code to keep a track of and display the correct letters (letters in the word and in the correct position) and all of the letters used. For example

Sample output:

```
Wordle is: flesh

-------------
| - - - - - |

Please enter your guess - attempt 1: sheer

-------------
| s h e e r |
| * * ^ - - |
|
| Correct spot(^): 1
| Wrong spot(*):   2
|
| Correct letters: e
| Used letters: s h r
```

Stage 9

Now… it's time to allow the player to have more than one guess for each word (six to be exact). Let's add another loop that loops until the user either guesses the correct word or runs out of attempts (the user must guess the word within six attempts). Think about where this code should go – what needs to be repeated, etc.

Sample output:

```
Would you like to play My Wordle [y|n]? y
Wordle is: tense

-------------
| - - - - - |

Please enter your guess - attempt 1: smart

-------------
| s m a r t |
| * - - - * |
|
| Correct spot(^): 0
| Wrong spot(*):   2
```

```
|
| Correct letters:
| Used letters: s m a r t

Please enter your guess - attempt 2: tears

-------------
| t e a r s |
| ^ ^ - - * |
|
| Correct spot(^): 2
| Wrong spot(*):   1
|
| Correct letters: t e
| Used letters: s m a r

Please enter your guess - attempt 3: tease

-------------
| t e a s e |
| ^ ^ - ^ ^ |
|
| Correct spot(^): 4
| Wrong spot(*):   0
|
| Correct letters: t e s
| Used letters: m a r

Please enter your guess - attempt 4: tense

-------------
| t e n s e |
| ^ ^ ^ ^ ^ |
|
| Correct spot(^): 5
| Wrong spot(*):   0
|
| Correct letters: t e s n
| Used letters: m a r


Solved in 4 tries!  Well done!


Would you like to play again [y|n]? n
```

Sample output:

```
Would you like to play My Wordle [y|n]? y
Wordle is: fraud

-------------
| - - - - - |

Please enter your guess - attempt 1: drake

-------------
| d r a k e |
| * ^ ^ - - |
|
| Correct spot(^): 2
| Wrong spot(*):   1
|
| Correct letters: r a
| Used letters: d k e
```

```
Please enter your guess - attempt 2: smoke

-------------
| s m o k e |
| - - - - - |
|
| Correct spot(^): 0
| Wrong spot(*):   0
|
| Correct letters: r a
| Used letters: d k e s m o

Please enter your guess - attempt 3: rains

-------------
| r a i n s |
| * * - - - |
|
| Correct spot(^): 0
| Wrong spot(*):   2
|
| Correct letters: r a
| Used letters: d k e s m o i n

Please enter your guess - attempt 4: stone

-------------
| s t o n e |
| - - - - - |
|
| Correct spot(^): 0
| Wrong spot(*):   0
|
| Correct letters: r a
| Used letters: d k e s m o i n t

Please enter your guess - attempt 5: phone

-------------
| p h o n e |
| - - - - - |
|
| Correct spot(^): 0
| Wrong spot(*):   0
|
| Correct letters: r a
| Used letters: d k e s m o i n t p h

Please enter your guess - attempt 6: flown

-------------
| f l o w n |
| ^ - - - - |
|
| Correct spot(^): 1
| Wrong spot(*):   0
|
| Correct letters: r a f
| Used letters: d k e s m o i n t p h l w


Oh no! Better luck next time!

The wordle was: fraud
```

```
Would you like to play again [y|n]? n
```

Stage 10

Add code to keep track of how many games were played, the number of wordles solved and the number of wordles unsolved. Display this to the screen as seen in the sample output.

Stage 11

Add code to validate the following user input:

- Would you like to play My Wordle [y|n]?

  Sample output:
  ```
  Would you like to play My Wordle [y|n]? z
  Would you like to play My Wordle [y|n]? p
  Would you like to play My Wordle [y|n]? y
  ```

- Please enter your guess - attempt #:

  Should check to ensure guess entered is a five letter word and that the word is a valid word (i.e. exists in the `word_list` list).

  Sample output:
  ```
  Please enter your guess - attempt 1: sat

  Five letter words only please.

  Please enter your guess - attempt 1: zzzzz

  Not in word list!

  Please enter your guess - attempt 1: water
  ```

- Would you like to play again [y|n]?

  Sample output:
  ```
  Would you like to play again [y|n]? z
  Would you like to play again [y|n]? p
  Would you like to play again [y|n]? y
  ```

Stage 12

Add code to display the My Wordle game summary to the screen (i.e. displayed once the user enters 'n'). Display this to the screen as seen in the sample output.

Stage 13

Modify your code to include and make use of the following **three** functions:

- `display_details()`
- `get_wordle_guess()`
- `create_word_list()`

1. Write a function called `display_details()` that will display your details to the screen. The function takes no parameters and does not return a value. The function simply displays your details to the screen. Your function should produce the following output (with your details).

   Output:
   ```
   File     : wayby001_poker.py
   Author   : Batman
   Stud ID  : 0123456X
   Email ID : wayby001
   This is my own work as defined by the
   University's Academic Misconduct Policy.
   ```

2. Write a function called `get_wordle_guess()` that prompts for and reads the user's guess. The function takes the list of words (`word_list`) as a parameter and validates the user's guess to ensure that it is a five-letter word and the word exists in the list of words (`word_list`) passed in as a parameter. The function returns the valid guess entered by the user.

3. Write a function called `create_word_list()` that takes a filename (as a string) as a parameter. The function opens the file for reading, read the words stored in the words file (word_file.txt) and creates a list of words with the words contained in the word file. The function returns the list of words read in from the file. Modify your code so that it now uses the list of words returned from this function in place of the original list created in stage 1 (`word_list`).

Stage 14

Think about and check that you are correctly handling the special cases. One interesting special case is whether you are correctly handling multiple copies of the same letter. For example, if the wordle is glass and you guess sassy, the s in the fourth position is positioned correctly and shown with a caret (^), the s in the first position appears in the word but is in the wrong spot and shown with an asterisk (*). However, the s in the third position is displayed with a dash as the wordle does not contain three instances of the letter s.

```
Would you like to play My Wordle [y|n]? y

Wordle is: glass

-------------
| - - - - - |

Please enter your guess - attempt 1: sassy

-------------
| s a s s y |
| * * - ^ - |
|
| Correct spot(^): 1
| Wrong spot(*):   2
|
| Correct letters: s
| Used letters: a y
```

Stage 15 – THIS IS IMPORTANT!

Finally, check the **sample output (see section titled 'Sample Output' towards the end of this document) and if necessary, modify your code so that:**

- The output produced by your program **EXACTLY** matches the sample output provided.
- Your program **EXACTLY** behaves as described in these specs **and** the sample output provided.

SUBMISSION DETAILS

You are required to do the following in order to submit your work and have it marked:

- Internal students:
  - You are required to submit an electronic copy of your program via learnonline **before Tuesday 13 June (swot-vac), 10 am.**

All students must follow the submission instructions below:

Ensure that your files are named correctly (as per instructions outlined in this document).

Ensure that the following files are included in your submission:

- `yourEmailId_my_wordle.py`

For example (if your name is James Bond, your submission files would be as follows):

- `bonjy007_my_wordle.py`

All files that you submit must include the following comments.

```
#
# File: file_name.py
# Author: your name
# Email Id: your email id
# Description: Assignment 2 - place assignment description here…
# This is my own work as defined by the University's
# Academic Misconduct policy.
#
```

Assignments that do not contain these details may not be marked.

You must submit your program **before the online due date**. Work that has not been correctly submitted to learnonline will not be marked.

It is expected that students will make copies of all assignments and be able to provide these if required.

EXTENSIONS AND LATE SUBMISSIONS

There will be **no** extensions/late submissions for this course without one of the following exceptions:

1. A medical certificate is provided that has the timing and duration of the illness and an opinion on how much the student's ability to perform has been compromised by the illness. **Please note** if this information is not provided the medical certificate WILL NOT BE ACCEPTED. Late assessment items will not be accepted unless a medical certificate is presented to the Course Coordinator. The certificate must be produced as soon as possible and must cover the dates during which the assessment was to be attempted. In the case where you have a valid medical certificate, the due date will be extended by the number of days stated on the certificate up to five working days.

2. A Learning and Teaching Unit councillor contacts the Course Coordinator on your behalf requesting an extension. Normally you would use this if you have events outside your control adversely affecting your course work.

3. Unexpected work commitments. In this case, you will need to attach a letter from your work supervisor with your application stating the impact on your ability to complete your assessment.

4. Military obligations with proof.

Applications for extensions must be lodged via learnonline before the due date of the assignment.

Note: Equipment failure, loss of data, 'Heavy work commitments' or late starting of the course are not sufficient grounds for an extension.

ACADEMIC MISCONDUCT

ACADEMIC MISCONDUCT

Students are reminded that they should be aware of the academic misconduct guidelines available from the University of South Australia website.

Deliberate academic misconduct such as plagiarism is subject to penalties. Information about Academic integrity can be found in Section 9 of the *Assessment policies and procedures manual* at:
http://www.unisa.edu.au/policies/manual/

MARKING CRITERIA

University of South Australia

Assessment feedback

COMP 1039 Problem Solving and Programming – SP2, 2023

NAME:	AVAILABLE MARKS	MARK	COMMENT
PRODUCES CORRECT RESULTS (OUTPUT)	**50 MARKS**		
	☐ Line spacing correct (2 marks)		
File : wayby001_my_wordle.py Author : Batman Stud ID : 0123456X Email ID : wayby001 This is my own work as defined by the University's Academic Misconduct Policy. ---------- -- My Wordle! -- -- Guess the Wordle in 6 tries -- ----------	☐ Details display (1 mark) ☐ Title display and Instructions display (1)		
Would you like to play My Wordle [y|n]?	☐ Prompt ('y' or 'n') (1)		
Wordle is: plays ---------- | - - - - - |	☐ Wordle display (1) ☐ Word display & dec (4) -1 For each formatting not to specs (up to 4 marks)		
Please enter your guess - attempt 1:	☐ Guess prompt (1) ☐ Attempt no correct and incrementing correctly (1)		
Please enter your guess - attempt 1: pours ---------- **| p o u r s |** | ^ - - - ^ | |	☐ Letter display correct (1)		
Please enter your guess - attempt 1: pours ---------- **| p o u r s |** | ^ - - - ^ | | **| Correct spot(^): 2** | Wrong spot(*): 0 | | Correct letters: p s | Used letters: o u r	**Letters in correct spot:** ☐ ^ in correct pos (4) ☐ Correct count (1) ☐ Correct count text (1)		
Please enter your guess - attempt 1: spilt ---------- **| s p i l t |** **| * * - * - |** | | Correct spot(^): 0 **| Wrong spot(*): 3** | | Correct letters: | Used letters: s p i l t	**Letters in wrong spot:** ☐ * in correct pos (4) ☐ Wrong count (1) ☐ Wrong count text (1)		

Please enter your guess - attempt 1: truck ---------- **| t r u c k |** | - - - - - | | **| Correct spot(^): 0** **| Wrong spot(*): 0** | | Correct letters: | Used letters: t r u c k	**Letters not in word at all:** ☐ - in correct pos (2) -1 For each count/text not to specs (up to 2 marks)		
Please enter your guess - attempt 2: slips ---------- | s l i p s | **| - ^ - * ^ |** **|** **| Correct spot(^): 2** **| Wrong spot(*): 1** | | Correct letters: l s | Used letters: p i t	**Special cases** - multiple copies of the same letter: ☐ Correct handling of multiple copies of same letters (4)		
Wordle is: fakes ---------- | - - - - - | Please enter your guess - attempt 1: steal ---------- | s t e a l | | * - * * - | | | Correct spot(^): 0 | Wrong spot(*): 3 | **| Correct letters:** **| Used letters: s t e a l** Please enter your guess - attempt 2: takes ---------- | t a k e s | | - ^ ^ ^ ^ | | | Correct spot(^): 4 | Wrong spot(*): 0 | **| Correct letters: a k e s** **| Used letters: t l** Please enter your guess - attempt 3:	**Letters correct and used:** ☐ Correct letters (2) ☐ Used letters (2) ☐ Correctly updating from used to correct as appropriate (2)		
Solved in 2 tries! Well done!	☐ Correct solved text (1) ☐ Correct solved count (1)		
Phew! Solved in 6 tries! Well done!	☐ Correct Phew text (1)		
Oh no! Better luck next time! The wordle was: plays	☐ Correct Oh no! text (1) ☐ Correct wordle text (1)		
Would you like to play again [y|n]?	☐ Prompt ('y' or 'n') (1)		
My Wordle Summary ========== You played 1 games: |--> Number of wordles solved: 0 |--> Number of wordles unsolved: 1	☐ Correct layout (1) ☐ Correct text (1) ☐ Games played value (1) ☐ Correct guess count (1) ☐ Incorrect count (1)		

☐

`Thanks for playing!`
 	
'N' on first input test:
`No worries... another time perhaps... :)`
	☐ Thanks message (1)
	
	☐ Another time msg (1)		

ADHERES TO SPECIFICATIONS (CODE)		**COMMENT**
Use of `random.choice(word_list)` for random wordle selection		☐ -2 No or incorrect use or r.choice()
Use of list of words (`word_list`) to store wordle words		☐ -2 No or incorrect use of word list
While loop for play again (`play_again == 'y' or equivalent boolean expression`)		☐ -2 No or incorrect loop
While loop for six guesses or until guess correct word		☐ -2 No or incorrect loop
List to store feedback provided to the user (i.e. ^, *, or -)		☐ -4 No or incorrect list (-2 only if printing feedback instead of storing feedback in list)
List or string to store correct and incorrect letters used		☐ -4 No or incorrect lists or strings
Loop to determine how many letters in correct position		☐ -2 No or incorrect loop
Loop to determine how many letters in wrong position		☐ -2 No or incorrect loop
Function `display_details()`		☐ -3 No or -2 incorrect display function
Function `get_wordle_guess(word_list)` *Returns validated guess.*		☐ -5 No or -2 incorrect get function
Function `create_word_list(file)` *Returns list of words read in from file.*		☐ -5 No or -2 incorrect create function

Validation of user input messages – both (y/n) prompts:
 	
`Would you like to play My Wordle [y|n]?`
 	
`Would you like to play again [y|n]?`
		☐ -1 No validation of user input
		
		☐ -1 No validation of user input

Validation of user input:
 	
`Please enter your guess - attempt 1: zzzyy`
 	
`Not in word list!`
		☐ -2 No validation of user input

Validation of user input:
 	
`Please enter your guess - attempt 1: rubbish`
 	
`Five letter words only please.`
		☐ -2 No validation of user input

Good loops (i.e. no break, continue, return, goto, etc statements to exit loops). Must exit loop via Boolean expression.		☐ -2 For using break/return/etc statements to exit loops

ADHERES TO SPECIFICATIONS (CODE)		COMMENT
Should **NOT use** the following: Built-in functions (other than those allowed) List or String methods (other than those allowed) Enumerate() function List comprehensions Dictionary to store data items		☐ -4 if using built-in functions ☐ -4 if using list or string methods ☐ -4 if using enumerate ☐ -4 if using list comprehensions ☐ -4 if using dictionary

STYLE (BOTH PARTS)	5 MARKS	MARK	COMMENT
Comments: ○ your details at top of file ○ program description ○ all variable definitions ○ all function definitions ○ significant sections of code			☐ -2 Insufficient comments ○ your details at top of file, ○ program description, ○ all variable definitions, ○ functions, and ○ significant sections of code
Consistent code layout and indentation.			☐ -2 Inconsistent indentation and layout
Meaningful variable names (no single letter variable names).			☐ -2 Non-descriptive variable names
TOTAL	**55 MARKS**		

The Graduate qualities being assessed by this assignment are indicated by an X:

X	GQ1: operate effectively with and upon a body of knowledge		GQ5: are committed to ethical action and social responsibility
	GQ2: are prepared for lifelong learning	X	GQ6: communicate effectively
	GQ3: are effective problem solvers		GQ7: demonstrate an international perspective
	GQ4:can work both autonomously and collaboratively		

This form meets the 2006 requirements of UniSA's Code of Good Practice: Student Assessment

Possible deductions:

- *Programming style:* Things to watch for are poor or no commenting, poor variable names, etc.
- *Submitted incorrectly:* -10 marks if assignment is submitted incorrectly (i.e. not adhering to the specs).

SAMPLE OUTPUT – MY WORDLE!

Sample output 1:

```
File     : wayby001_my_wordle.py
Author   : Batman
Stud ID  : 0123456X
Email ID : wayby001
This is my own work as defined by the University's Academic Misconduct Policy.


--------------------------------
--         My Wordle!         --
-- Guess the Wordle in 6 tries --
--------------------------------


Would you like to play My Wordle [y|n]? n


No worries... another time perhaps... :)
```

Sample output 2:

```
File     : wayby001_my_wordle.py
Author   : Batman
Stud ID  : 0123456X
Email ID : wayby001
This is my own work as defined by the University's Academic Misconduct Policy.


--------------------------------
--         My Wordle!         --
-- Guess the Wordle in 6 tries --
--------------------------------


Would you like to play My Wordle [y|n]? p
Would you like to play My Wordle [y|n]? y

Wordle is: water

-------------
| - - - - - |

Please enter your guess - attempt 1: rates

-------------
| r a t e s |
| * ^ ^ ^ - |
|
| Correct spot(^): 3
| Wrong spot(*):   1
|
| Correct letters: a t e
| Used letters: r s

Please enter your guess - attempt 2: rat

Five letter words only please.

Please enter your guess - attempt 2: zippo

Not in word list!

Please enter your guess - attempt 2: rather

Five letter words only please.

Please enter your guess - attempt 2: states

Five letter words only please.

Please enter your guess - attempt 2: state

-------------
| s t a t e |
| - * * - * |
|
```

```
| Correct spot(^): 0
| Wrong spot(*):   3
|
| Correct letters: a t e
| Used letters: r s

Please enter your guess - attempt 3: water

-------------
| w a t e r |
| ^ ^ ^ ^ ^ |
|
| Correct spot(^): 5
| Wrong spot(*):   0
|
| Correct letters: a t e w r
| Used letters: s


Solved in 3 tries!  Well done!


Would you like to play again [y|n]? n


My Wordle Summary
=================

You played 1 games:
  |--> Number of wordles solved:   1
  |--> Number of wordles unsolved: 0


Thanks for playing!
```

Sample output 3:

```
File     : wayby001_my_wordle.py
Author   : Batman
Stud ID  : 0123456X
Email ID : wayby001
This is my own work as defined by the University's Academic Misconduct Policy.


---------------------------------
--         My Wordle!          --
-- Guess the Wordle in 6 tries --
---------------------------------


Would you like to play My Wordle [y|n]? y

Wordle is: smoke

-------------
| - - - - - |

Please enter your guess - attempt 1: mores

-------------
| m o r e s |
| * * - * * |
|
| Correct spot(^): 0
| Wrong spot(*):   4
|
| Correct letters:
| Used letters: m o r e s

Please enter your guess - attempt 2: smart

-------------
| s m a r t |
| ^ ^ - - - |
|
| Correct spot(^): 2
| Wrong spot(*):   0
|
| Correct letters: s m
| Used letters: o r e a t

Please enter your guess - attempt 3: flake
```

```
-------------
| f l a k e |
| - - - ^ ^ |
|
| Correct spot(^): 2
| Wrong spot(*):   0
|
| Correct letters: s m k e
| Used letters: o r a t f l

Please enter your guess - attempt 4: makes

-------------
| m a k e s |
| * - * * * |
|
| Correct spot(^): 0
| Wrong spot(*):   4
|
| Correct letters: s m k e
| Used letters: o r a t f l

Please enter your guess - attempt 5: trust

-------------
| t r u s t |
| - - - * - |
|
| Correct spot(^): 0
| Wrong spot(*):   1
|
| Correct letters: s m k e
| Used letters: o r a t f l u

Please enter your guess - attempt 6: skate

-------------
| s k a t e |
| ^ * - - ^ |
|
| Correct spot(^): 2
| Wrong spot(*):   1
|
| Correct letters: s m k e
| Used letters: o r a t f l u


Oh no! Better luck next time!

The wordle was: smoke


Would you like to play again [y|n]? y

Wordle is: trust

-------------
| - - - - - |

Please enter your guess - attempt 1: tests

-------------
| t e s t s |
| ^ - * * - |
|
| Correct spot(^): 1
| Wrong spot(*):   2
|
| Correct letters: t
| Used letters: e s

Please enter your guess - attempt 2: rusts

-------------
| r u s t s |
| * * * * - |
|
| Correct spot(^): 0
| Wrong spot(*):   4
|
| Correct letters: t
```

```
| Used letters: e s r u

Please enter your guess - attempt 3: truck

-------------
| t r u c k |
| ^ ^ ^ - - |
|
| Correct spot(^): 3
| Wrong spot(*):   0
|
| Correct letters: t r u
| Used letters: e s c k

Please enter your guess - attempt 4: trust

-------------
| t r u s t |
| ^ ^ ^ ^ ^ |
|
| Correct spot(^): 5
| Wrong spot(*):   0
|
| Correct letters: t r u s
| Used letters: e c k


Solved in 4 tries!  Well done!


Would you like to play again [y|n]? n


My Wordle Summary
=================

You played 2 games:
  |--> Number of wordles solved:   1
  |--> Number of wordles unsolved: 1


Thanks for playing!
```

Sample output 4:

```
File     : wayby001_my_wordle.py
Author   : Batman
Stud ID  : 0123456X
Email ID : wayby001
This is my own work as defined by the University's Academic Misconduct Policy.


---------------------------------
--         My Wordle!          --
-- Guess the Wordle in 6 tries --
---------------------------------


Would you like to play My Wordle [y|n]? y

Wordle is: glass

-------------
| - - - - - |

Please enter your guess - attempt 1: grace

-------------
| g r a c e |
| ^ - ^ - - |
|
| Correct spot(^): 2
| Wrong spot(*):   0
|
| Correct letters: g a
| Used letters: r c e

Please enter your guess - attempt 2: start

-------------
| s t a r t |
| * - ^ - - |
```

```
|
| Correct spot(^): 1
| Wrong spot(*):   1
|
| Correct letters: g a
| Used letters: r c e s t

Please enter your guess - attempt 3: sassy

-------------
| s a s s y |
| * * - ^ - |
|
| Correct spot(^): 1
| Wrong spot(*):   2
|
| Correct letters: g a s
| Used letters: r c e t y

Please enter your guess - attempt 4: glass

-------------
| g l a s s |
| ^ ^ ^ ^ ^ |
|
| Correct spot(^): 5
| Wrong spot(*):   0
|
| Correct letters: g a s l
| Used letters: r c e t y


Solved in 4 tries!  Well done!


Would you like to play again [y|n]? n


My Wordle Summary
=================

You played 1 games:
  |--> Number of wordles solved:   1
  |--> Number of wordles unsolved: 0


Thanks for playing!
```

Sample output 5:

```
File     : wayby001_my_wordle.py
Author   : Batman
Stud ID  : 0123456X
Email ID : wayby001
This is my own work as defined by the University's Academic Misconduct Policy.


---------------------------------
--         My Wordle!          --
-- Guess the Wordle in 6 tries --
---------------------------------


Would you like to play My Wordle [y|n]? y

Wordle is: fakes

-------------
| - - - - - |

Please enter your guess - attempt 1: spade

-------------
| s p a d e |
| * - * - * |
|
| Correct spot(^): 0
| Wrong spot(*):   3
|
| Correct letters:
| Used letters: s p a d e
```

```
Please enter your guess - attempt 2: trade

-------------
| t r a d e |
| - - * - * |
|
| Correct spot(^): 0
| Wrong spot(*):   2
|
| Correct letters:
| Used letters: s p a d e t r

Please enter your guess - attempt 3: draft

-------------
| d r a f t |
| - - * * - |
|
| Correct spot(^): 0
| Wrong spot(*):   2
|
| Correct letters:
| Used letters: s p a d e t r f

Please enter your guess - attempt 4: fouls

-------------
| f o u l s |
| ^ - - - ^ |
|
| Correct spot(^): 2
| Wrong spot(*):   0
|
| Correct letters: f s
| Used letters: p a d e t r o u l

Please enter your guess - attempt 5: lakes

-------------
| l a k e s |
| - ^ ^ ^ ^ |
|
| Correct spot(^): 4
| Wrong spot(*):   0
|
| Correct letters: f s a k e
| Used letters: p d t r o u l

Please enter your guess - attempt 6: fakes

-------------
| f a k e s |
| ^ ^ ^ ^ ^ |
|
| Correct spot(^): 5
| Wrong spot(*):   0
|
| Correct letters: f s a k e
| Used letters: p d t r o u l


Phew! Solved in 6 tries! Well done!


Would you like to play again [y|n]? n


My Wordle Summary
=================

You played 1 games:
  |--> Number of wordles solved:   1
  |--> Number of wordles unsolved: 0


Thanks for playing!
```