L-Università ta' Malta

ARI2201 - Assignment

Individual Assigned Practical Task

Nathan Bonavia Zammit

Bachelor of Science in

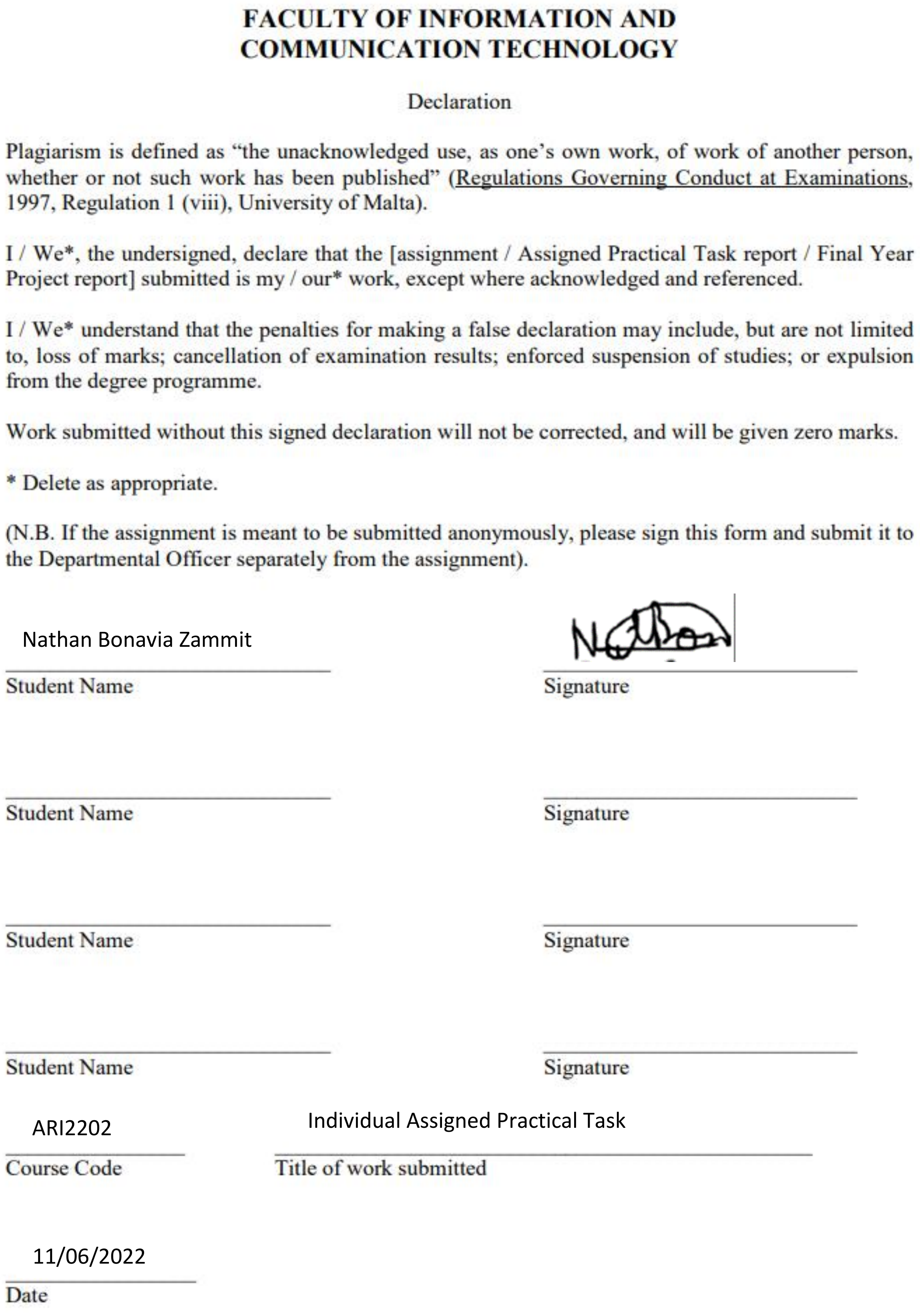
Information Technology (Honours)

(Artificial Intelligence)

June 2022

Contents

Plagiarism Form



FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Declaration

Plagiarism is defined as "the unacknowledged use, as one's own work, of work of another person, whether or not such work has been published" (Regulations Governing Conduct at Examinations, 1997, Regulation 1 (viii), University of Malta).

I / We*, the undersigned, declare that the [assignment / Assigned Practical Task report / Final Year Project report] submitted is my / our* work, except where acknowledged and referenced.

I / We* understand that the penalties for making a false declaration may include, but are not limited to, loss of marks; cancellation of examination results; enforced suspension of studies; or expulsion from the degree programme.

Work submitted without this signed declaration will not be corrected, and will be given zero marks.

* Delete as appropriate.

(N.B. If the assignment is meant to be submitted anonymously, please sign this form and submit it to the Departmental Officer separately from the assignment).

Nathan Bonavia Zammit

Student Name Signature

Student Name Signature

Student Name Signature

Student Name Signature

ARI2202 Individual Assigned Practical Task

Course Code Title of work submitted

11/06/2022

Date

Introduction

For our IAPT this year we were given multiple different options to choose from. I had applied for and was chosen to build an Automated Maltese Wordle Solver. The requirements of the project consisted of a program which would automate the process of playing the game known as 'Wordle' in an automated way. The only difference was that the solver had to cater for the Maltese version of the game. To handle this, different resources were investigated such as Ġabra and so forth.

Goals

When it came to the approach on how to build the solver the first objective was to build a solver in the English language. This was attempted first since finding the wordlists of all possible guesses and possible answers was far easier as due to the popularity of the game, people had already extracted them from the official Wordle game and as such they were easier to find. After looking into this, I had found them on reddit in someone's comments [1].

Building the solver in English was a success, and as such all that needed to be altered was for the solver to be able to play the game in Maltese. To do so, Wordle Clones were looked into.

Wordle Clones

Due to the massive success of the Wordle game, various Wordle clones were created in the process. Some of these include:

- Worldle – Invites the user to guess countries based on their outline [2]
- Squabble – A multiplayer Battle Royale version of the Wordle game [3]
- Dordle – Forces the player to solve two Wordles at once [4]
- Wordle in different languages such as in Maltese
    - Kelma [5]
    - Werdil [6]

For the solver, which was built, the Wordle clone known as 'Werdil' was used as it caters for the Maltese language including special characters such as 'ġ', 'ħ', 'ż', 'ċ', 'għ' and 'ie'. Kelma on the other hand does not cater for 'għ' or 'ie' therefore it was not used. Werdil also allowed to have a daily word and a continous number of words which made testing of the solver much easier.

Literature Review

What is Wordle exactly?

Wordle is a daily online word game in which the player must guess the chosen word by no less than 6 tries to guess a randomly chosen 5 letter word. The player does so by entering a valid 5 letter word of their own choosing and the game returns feedback based off of the chosen word. If a letter of the chosen word is incorrect, then the letter would be grayed out, if the letter is in the word but not in its correct place, then the letter would be yellowed out, if on the other hand, the letter is in the word and in its correct place, then the letter would be greened out. Through this feedback the player can figure out the word more efficiently as it helps them eliminate possible words and narrow down and figure out possible words that they can use. It is encouraged to avoid using any grayed-out letters to get better feedback however in certain scenarios it would be smart to sacrifice a turn to get closer to the correct word.

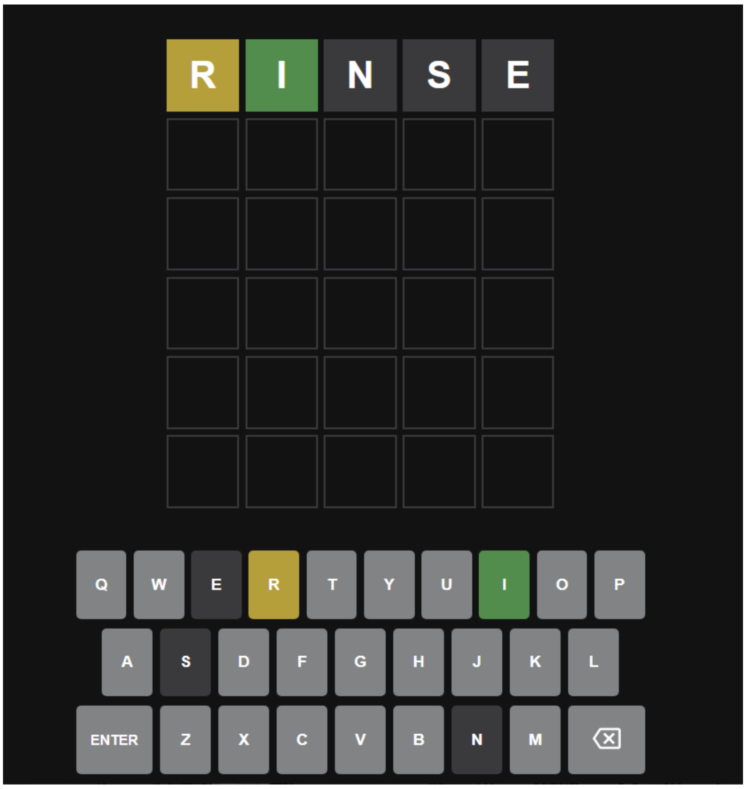
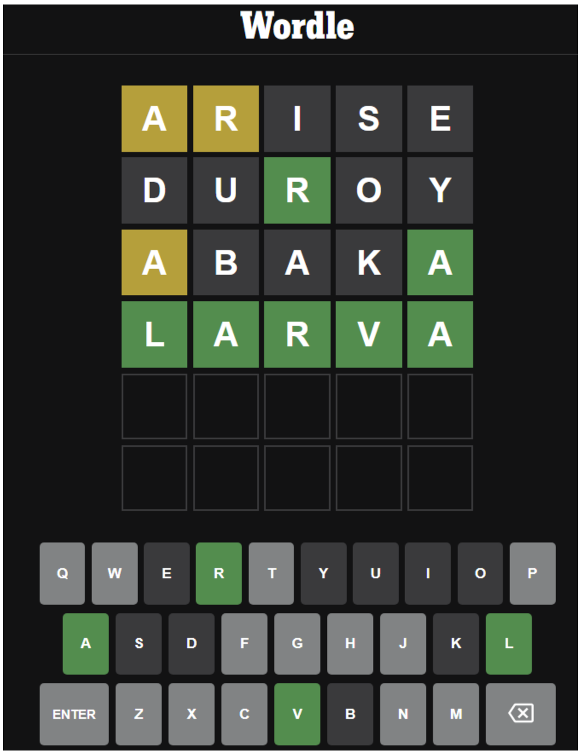


Fig.1. Example of user inputting a word into the Wordle Game

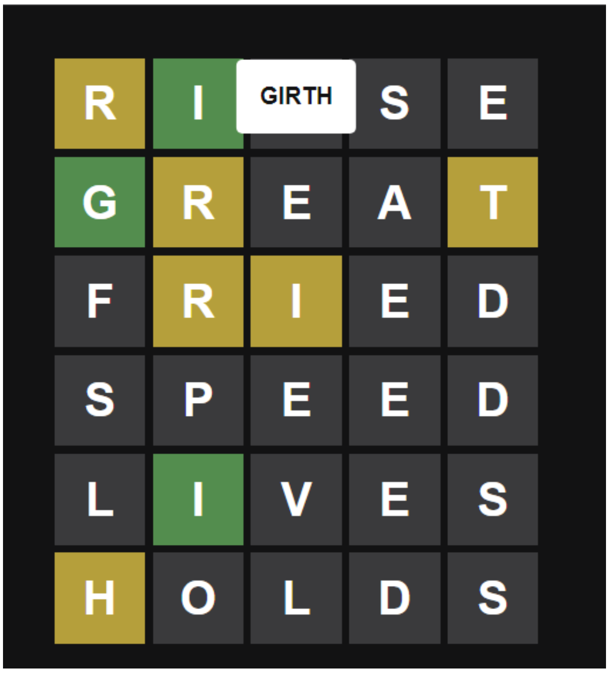
The game finishes once the user either wins the game by guessing the word, or by using up all 6 tries and not guessing the word correctly. The user will be then given the option to copy their score if they wish to share it throughout social media, and if they lost, they will be informed what the correct word was.



Wordle

A	R	I	S	E
D	U	R	O	Y
A	B	A	K	A
L	A	R	V	A

Fig.2. Example of a correct guessed word



GIRTH

R	I		S	E
G	R	E	A	T
F	R	I	E	D
S	P	E	E	D
L	I	V	E	S
H	O	L	D	S

Fig.3. Example of a failed Wordle attempt

Who made Wordle?

Wordle is the work of software engineer Josh Wardle, who originally had created the game for his partner who is a fan of word games. Wardle is also known as the creator of ‘Place’ which was a collaborative art project/social experiment that had taken place back in 2017. Wordle as a game was even mentioned by the New York Times, however what really made it popular was the share function which was added later. The share function allowed users to have the option to share their scores in a way in which it does not spoil anything for anyone else. This led to a large number of people sharing their daily Wordle scores all over social media, especially twitter. Due to the rising popularity of the game, by the end of January 2022, Josh Wardle sold Wordle to the New York Times for a good price and promised to keep Wordle free to play for everybody. If one wishes to play the official Wordle game it can be found here: https://www.nytimes.com/games/wordle/ [7].

Design of the Solution.

Selenium

The solver that was built required the use of the python library Selenium to access the Werdil game's contents. Selenium is considered to be one of the most preferred tool suites for automation testing of web applications since it supports multiple different popular web browsers. These browsers range from Google Chrome to Mozilla Firefox. In our solver we used Selenium to access Google Chrome. Selenium also provides compatibility with various different programaming languages, in our case it is compatible with Python.

To access the 'Werdil' game we mainly looked into the 'Selenium WebDriver' which is a more enhanced version of 'Selenium RC'. Selenium RC (Remote Control) is a tool which allows one to develop responsive design tests in any scripting language which the user chooses. The main two components of Selenium RC are the Server and client libraries, and its architecture is complex and therefore has limitations. Selenoum WebDriver was then introduced in the market to overcome Selenium RC's limitations. Although it is an enhanced version of Selenium RC, it's architecture is completely different.

Selenium WebDriver is a web framework which permits the user to excecute cross-browser tests. This tool is used for automating web-based application testing to verify that it performs expectedly [8].

When it comes to setting up a proper Selenium WebDriver Script, there are a number of steps. Those being:

- Create a WebDriver Instance
- Navigate to a webpage
- Locate an element through the use of locators in selenium
- Perform the required user actions on said element
- Preload the expected output/browser response to the action
- Run test
- Record results and compare results from them to the expected output

These steps were followed when it came it to comparing a human trial and the solution's trial at playing 'Werdil'.

Obtaining the Wordlists

When it came to the implemenation two wordlists were required. One containing the best possible answers, and one containing all possible guesses. The 'Werdil' game replaced the keyboard inputs '/' and '?' into 'għ' and 'ie' respectively. So, the wordlists that needed to be used had to cater for this by changing all instances of 'għ' and 'ie' into '/' and '?' accordingly. The data for the 'Werdil' game could be found on github as the creator of this Wordle clone, Michael Pulis, had posted everything online. As such the wordlists for the guesses and answers were obtained from there. The only issue which was found was that when taking into consideration all possible guesses there were multiple incorrect words in the list and the program kept crashing when trying to access it. Therefore, to make the algorithm possible, and more efficient, the wordlist for the answers was used twice.

The Algorithm

The program will take in two wordlists as mentioned before, the guess list and the answer list. Let's say that we have words g1 through to gn in our guess list and we estimate that our guess list has an approximate 12,000 words in it. Then we have our answer list from a1 through to an and it is approximately 2,000 words long. The goal is to, through the use of the guess list, decrease our answer list more and more after each guess, until we only have one possible answer left.

In the code we will have a matrix. In the implementation, the guess list and the answer list are the same, therefore let us consider them as below where each word is denoted as w1, w2, and so forth.

			Guesses	
		w1	**w2**	**w3**
	w1	1	5	7
Answers	**w2**	3	1	3
	w3	6	8	1
	w4	2	3	3

Fig.4. Example Matrix used to help describe the Algorithm

The numbers in this matrix are the number of possible answers that we narrowed down our answer list into after making a specific guess. So, if the guess which we implemented is w1 and the real answer is w1, then the correct answer is met, and the word list has been narrowed down to just one word. This will rarely occur after the first guess, and it is very lucky. In the second guess, if we guess w2 and the answer is w2 then we found the correct answer once again, and same with guessing w3 and so forth.

However, in most cases the first guess will not be the correct one. Let us take into example if we guess w1, and the correct word is w2. In the example above, the information which we obtained from that guess narrows down our wordlist to just 3 words. If we guessed w1 in the beginning, and the correct word is w3, then with that information our wordlist is narrowed down to 6 words and so on and so forth for the rest of the numbers in our example matrix.

These numbers will be taken into consideration when making a guess and this can be done in multiple ways. One way is to take the average of each column and choose the column with the lowest average and therefore keep narrowing the answer list more and more. However, for this implementation, the technique chosen is to always minimize the worst-case scenario. So, from the example above, the guess w1 will narrow down the answer list to 6 or fewer. It can possible be narrowed down to 3 or 1 but we have the guarantee that the answer list will be no more than 6 possible answers. If we choose guess w2, then the answer list is decreased to 8 and for w3 the answer list will decrease to 7. In this example, the best worst-case scenario is for w1 as w2 and w3 have worse guarantees. This algorithm will take place after each guess and a matrix will be built after each guess. The best worst case scenario column will be chosen and this will take place until the answer list is equal to 1, or the correct word is reached.

Implementation of The Solution

Obtaining the wordlists

The functions ‘get_maltese_guesses’ and ‘get_maltese_answers’ were used to obtained the wordlists that contained the guesses and the answers which the algorithm would use to play the game automatically. It was important that the encoding type was ‘utf-8’ as otherwise the program crashed multiple times because it wouldn’t be able to identify special Maltese characters such as ‘ġ’, ‘ħ’, and so forth.

The Play Function

In the ‘play’ function the main algorithm will take place. First, the narrowed down list will start as all possible answers, however this will be narrowed down throughout the code. Then the program will run the algorithm for the next 5 guesses. While the Wordle game accepts 6 guesses there is a check at the end which will check that if that from our narrowed down list only one word is left after the 5 guesses, then that word will be our 6th guess. Therefore, there are actually 6 guesses, however it will loop 5 times.

The variable ‘min_wordcount’ is declared to a very large amount. It is the variable that we are trying to minimize. It is essentially the length of the narrowed down word list. Then the program checks if the guess number isn’t zero. If not, then it will take into consider the list of possible guesses. However, if it is, then the word ‘stari’ is used. After some testing it was seen that it was the best first guessing word for the algorithm.

Later on we look through every word that is being considered and every possible answer in our narrowed down list and they are evaluated with each other. The ‘get_evaluation’ function is very simple. If a letter is wrong, then it returns the value 0, if the letter is correct but in the right place then it returns the value 1, and if it’s correct and in the right place then it returns the value 2. After that evaluation is obtained, we save the evaluation as a tuple and then the value would be a list of all the words which contain that evaluation of all possible answers that have that evaluation.

After we obtain said list, we look at the worst case length of the narrowed down list after making a specific guess, and if we see that that wort case is better than the one which is currently save (which is set to a million at the beginning) then we update that min_wordcount variable and update the chosen word.

The ‘get_werdil_evaluation’ function uses the game’s own CSS elements to obtain the evaluation of each answer which is used. Afterwards, said evaluation is compared with the next chosen word’s evaluation which will further narrow down our answer list.

The Main Function

In our main function we set up how the solution will play the game. We first set up the button on the keyboard which will play our game as required. Then we set up the Selenoum browser and the declare the number of times the game will be played. This was not fully used correctly and will be mentioned why further on. After that, through the Selenium browser we open the Werdil game and set it up accordingly where we will simply press the aforementioned start button and the game will be played automatically using the algorithm which was declared in the ‘play’ function. It detects on which row the user is on by using CSS to find the element which refers to the game rows.

Potential Additions

When it came to picking the word to use for the first guess, the function ‘make_freq_guess’ was taken into consideration where it takes the frequency of each letter in each word in the answer list and then picks the best possible word through said frequencies. However, this function kept crashing the algorithm and due to time constraints, it was simply commented out.

The variable num_games in the main function was supposed to be used to keep playing the game based on the number which was assigned. However, when it came to obtaining the CSS elements which would allow the program to keep playing the game, said elements could not be found and due to time constraints were also commented out.

Results

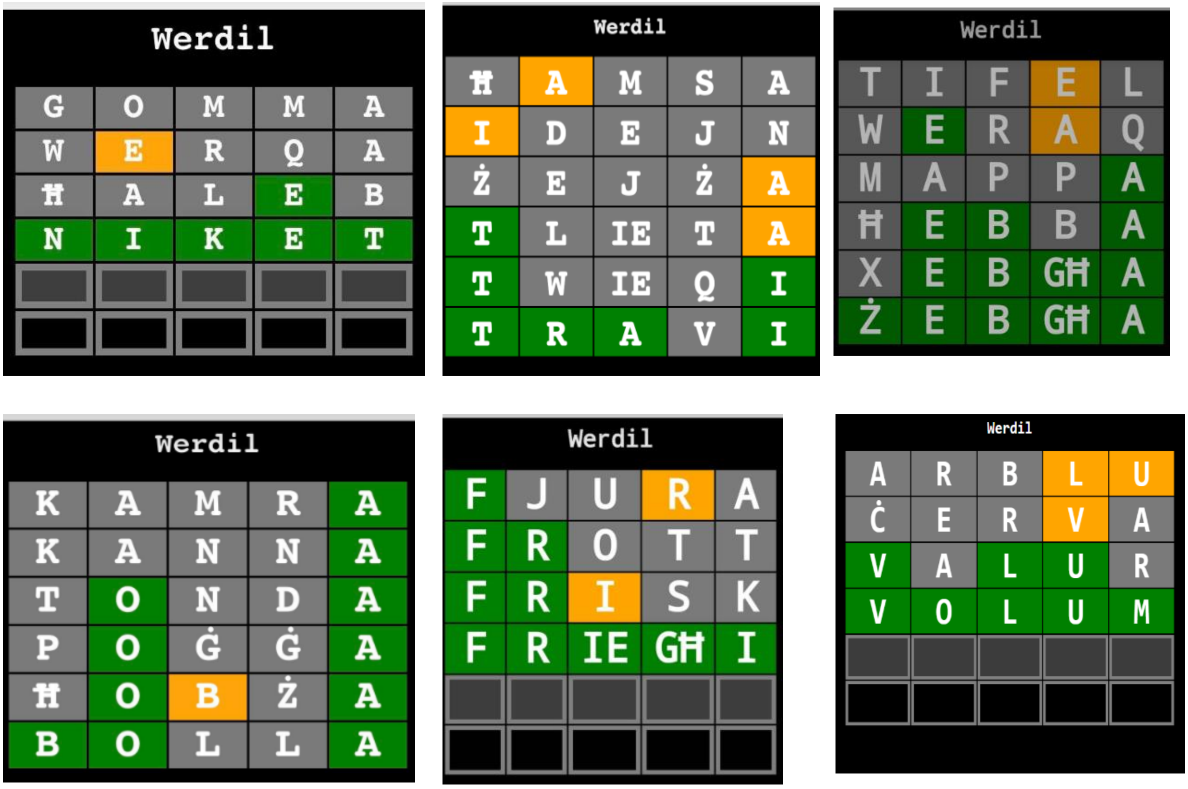
Human trials were attempted as I shared the game with some people and recorded their results. It was observed that when comparing to human trials the automated solver had a similar if not superior success rate as can be seen below:

Human Trials:

- Attempt 1 - 4/6
- Attempt 2 - unsuccessful
- Attempt 3 - 6/6
- Attempt 4 - 3/6
- Attempt 5 - 3/6
- Attempt 6 - unsuccessful
- Attempt 7 - unsuccessful
- Attempt 8 - unsuccessful
- Attempt 9 - unsuccessful
- Attempt 10 – unsuccessful

Success Rate: 40%

6 Human Trials (Attempted on different devices and as such all look different)



Werdil

G O M M A
W E R Q A
Ħ A L E B
N I K E T

Werdil

Ħ A M S A
I D E J N
Ż E J Ż A
T L IE T A
T W IE Q I
T R A V I

Werdil

T I F E L
W E R A Q
M A P P A
Ħ E B B A
X E B GĦ A
Ż E B GĦ A

Werdil

K A M R A
K A N N A
T O N D A
P O Ġ Ġ A
Ħ O B Ż A
B O L L A

Werdil

F J U R A
F R O T T
F R I S K
F R IE GĦ I

Werdil

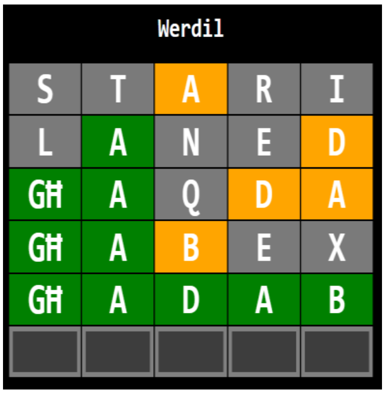
A R B L U
Ċ E R V A
V A L U R
V O L U M

Solver Trials:

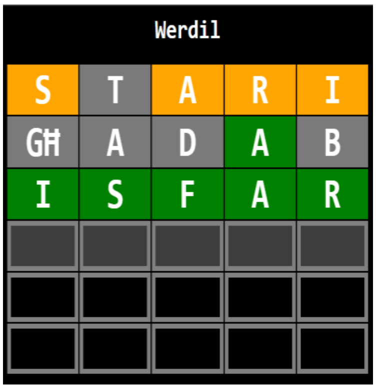
- Attempt 1 - 5/6
- Attempt 2 - 4/6
- Attempt 3 - 4/6
- Attempt 4 - 4/6
- Attempt 5 - 4/6
- Attempt 6 - 3/6
- Attempt 7 - 4/6
- Attempt 8 - 4/6
- Attempt 9 - 3/6
- Attempt 10 - 3/6

Success Rate: 100%

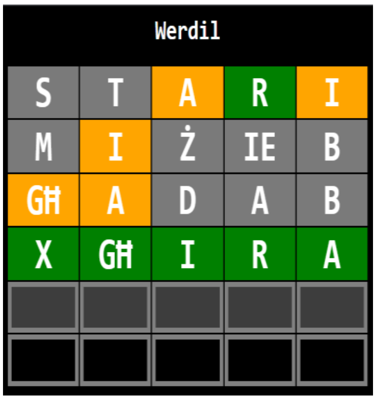
6 Computer Trials


Werdil

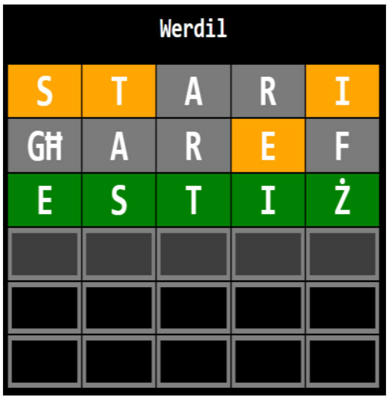
S	T	A	R	I
L	A	N	E	D
GĦ	A	Q	D	A
GĦ	A	B	E	X
GĦ	A	D	A	B


Werdil

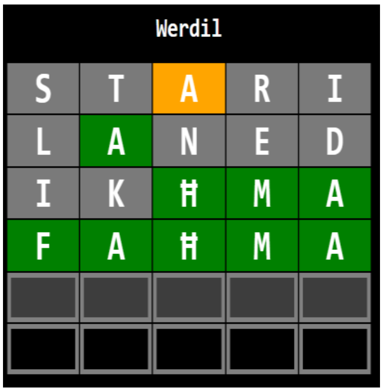
S	T	A	R	I
GĦ	A	D	A	B
I	S	F	A	R


Werdil

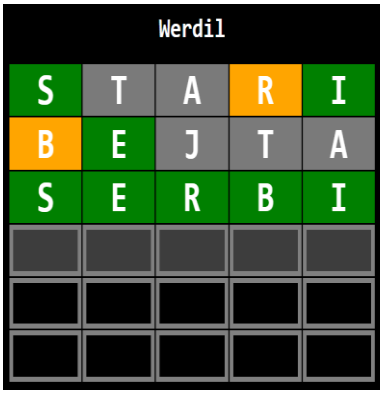
S	T	A	R	I
M	I	Ż	IE	B
GĦ	A	D	A	B
X	GĦ	I	R	A


Werdil

S	T	A	R	I
GĦ	A	R	E	F
E	S	T	I	Ż


Werdil

S	T	A	R	I
L	A	N	E	D
I	K	Ħ	M	A
F	A	Ħ	M	A


Werdil

S	T	A	R	I
B	E	J	T	A
S	E	R	B	I

As can be seen, the Solver trials had a far more superior success rate than the Human trails. This could be affected by the fact that most people who attempted the human trials did not have a great knowledge of Maltese words which are made up of 5 letters, and the fact that the solver has access to a dictionary of 5 letter words in Maltese. In possible future trials, giving the users a dictionary to refer to might have produced similar results.

Conclusion

The creation of Wordle back in the year of 2020 during the lonely lockdown days, led to a massive internet success where people could pass by the time just a little bit more by doing a simple daily game. The success of Wordle has led to various games which all follow the same simple format and are known as Wordle clones.

When it came to building a Solver, there are probably various different methods at processing the feedback gained from each answer attempted, however I believe that the best worst case scenario algorithm was efficient as it takes the maximum number of steps on input data of size 'n' of multiple different inputs and chooses the best one which would lead to the fastest output possible. Just having a best-case scenario would not have been optimal due to the large number of words which are dealt with.

References:

[1] – English Guess List and Answer List:
https://www.reddit.com/r/wordle/comments/s4tcw8/a_note_on_wordles_word_list/ (13/06/2022)

[2] – Worldle Game:

https://worldle.teuteuf.fr/

[3] – Squabble Game:

https://squabble.me/

[4] – Dordle Game:

https://dordle.io/

[5] – Kelma – Maltese Wordle:

https://kelma.mt/

[6] – Werdil – Maltese Wordle:

https://wordle-malti.github.io/

[7] – M. Serrels & D. Van Boom, "Wordle: What to Know About the Viral Word Game

https://www.cnet.com/culture/internet/wordle-what-to-know-about-the-viral-word-game/ (12/06/2022)

[7] – New York Times Official Wordle Game:

https://www.nytimes.com/games/wordle/

[8] – What is Selenium?

https://www.browserstack.com/guide/selenium-webdriver-tutorial#:~:text=Selenium%20WebDriver%20is%20a%20web,language%20to%20create%20test%20scripts. (12/06/2022)