INDIAN INSTITUTE OF TECHNOLOGY PALAKKAD

ID1110-W INTRODUCTION TO PROGRAMMING

HANGMAN

Submitted by :
112301020 - Sanjay Siva A
112301014 - Karthik Krishna M
112301030 - Madhav P Nair

Instructor:
Dr. Anish Hirwe

INDIAN INSTITUTE OF TECHNOLOGY PALAKKAD

INDEX

INTRODUCTION

1.BACKGROUND AND CONTEXT

Hangman is a classic word guessing game which has been enjoyed by generations and owns its legacy in the arena of computer games. It not only entertains but helps the user to ace vocabulary and spelling skills. It's a testament to the game's enduring appeal that the simple act of guessing letters continues to engage and educate players of all ages.

2.PROBLEM STATEMENT AND OBJECTIVES

To develop a digital version of the hangman game using **Python** ,basic graphical user interface(GUI) toolkit **Tkinter** and **MYSQL database**.

Here we present two gaming modes

1.User guesses the word randomly picked by the computer:

- Computer generates a random word from the database.
- Program shows the length of the word as the number of empty characters.
- The user is asked to guess any of the letters present in the word .
- Computer gives the feedback and displays the letter in the corresponding blank spaces if the guess has any correct matches and tracks the number of attempts available.
- Announces the player as the winner or loser based on whether he/she could guess the word within eight attempts.

2.User picks a word and the program guesses the word:

- The user chooses a word in their mind and the program asks to enter the length of the word.
- Program generates random characters and takes feedback from the user about the existence of the letter and the position(s) of the letter.

- The prediction first utilizes random character generation from common frequent characters then based on chances, program shifts to educated guess, where words with characters found in the string from previous guesses are retrieved from the database and is utilized for next character prediction.
- The filtering converges to a single word based on the response from the user.

The GUI toolkit helps to enhance a user-friendly interactive interface for smooth gameplay experience.

3.SIGNIFICANCE AND MOTIVATION

The game has an entertaining as well as educational profile ,the uncomplicated setup and interface also inspires the enthusiasts to go ahead with such projects.

The logic behind the gameplay is elementary but powerful. The sequence led by the pc against the user based on educated guesses opens up new windows of reasoning.

AN OVERVIEW OF THE PROJECT

1.PROJECT GOALS AND SCOPES:

The work simply aims at developing a hangman game with the help of the learnings from the course. The digital replica of the traditional game both preserves and makes it accessible to the modern audience.

2.PROJECT TIMELINE

- 27/02/2024-05/03: Planning and Research - to fix the topic of the project ,to define game mechanics

- 05/03-15/03: Developing the game logic
- 15/03-25/03: Getting clear with database connections and filtering methods using MySQL.
- 25/03-08/04 : Designing GUI elements
- 08/04-15/04 : Integrating the components of game file with database and GUI
- 15/04-20/04 :Conducting test runs to take feedbacks
- 20/04-25/04 : Refining user interface elements and addressing some bugs to get ready for the deployment

- PROJECT REPOSITORY

Link to Project Repository

- Team members and contributions

Karthik Krishna M - Designed the game logic , setup the database connections

Sanjay Siva A- Designing of game logic, setting up the GUI elements

Madhav P Nair- Designing the game logic, setting up the GUI

METHODOLOGY

Approach and methodology employed:

The game mode where the user guessed the word was implemented simply by picking a random word from a database and tracking the correct and wrong guesses of the user. The ‘random’ module was used to select the word from the database.

In the game mode where the computer makes the guesses, the computer initially tries to figure out the position of vowels if present in the word and then proceeds with consonants.

There is also a function that learns the word if the program was unable to guess the word correctly.

The code to process the input in both game modes are written in the files <main.py> and <dbcon.py> .

<gui.py> specifies the appearance of the program window and also contains functions that take inputs and pass them on to the ‘main’ and ‘dbcon’ modules.

In the <dbcon> module mysql commands are implemented as python strings using the <mysql.connector> module. This is the module that shortlists the words from the database using the inputs for the second game mode.

Tools,technologies and frameworks used

The modules used in the code are:

- <os> : To set up the database
- <random>: To generate random word, guess letters
- <mysql.connector> : To execute MySQL commands using python.
- <tkinter> : To develop the GUI.

Other than normal python code and associated modules MySQL was used to represent the wordlist as an easy to lookup dictionary and manipulate the inputs effortlessly.

Experimental setup and data collection methods

The program performed best for 5-8 letter long words which contained vowels.It did not perform well when it came to smaller words and words which didn’t have any vowels.

It was observed that the program guessed the words with a much higher probability after it initially failed and the learn_word() function was executed.

4. RESULTS AND ANALYSIS

1.ACHIEVED RESULTS AND FINDINGS

User plays the hangman:

The program generates a random word (kept hidden) from the database and shows the number of letters to initiate the gameplay from the side of the user.

The code works perfect in providing the accurate feedback to the user about the existence of the letter (he/she entered) in the word and tracks the number of chances (lives) remaining for the player.It analyses the credibility of the user input and announces whether he wins or loses depending on whether guessing the entire word or eight wasted guesses occur first without a fail.

Computer plays the hangman:

The user himself assumes a word and gives the number of letters. The program starts the guess based on a,e,i,o {common characters} and further proceeds to an educated guess based on the list of words filtered out using the MySQL database operations. Many-a-times the program successfully guesses the word but we also encounter situations where the computer spoils the eight attempts to end up in wrong result and prompts the user to give the word.

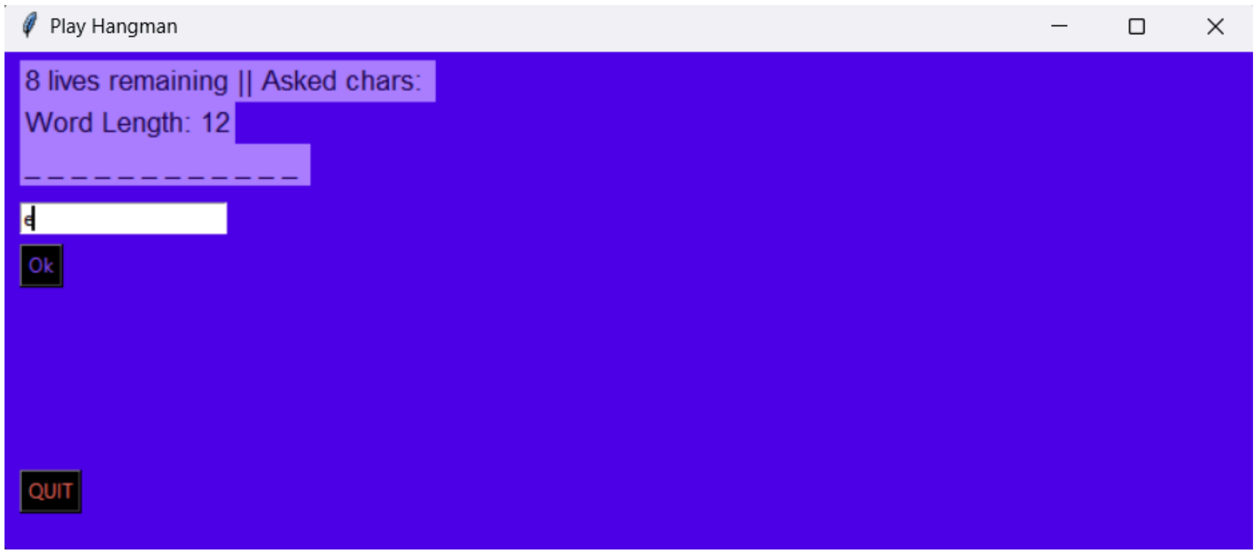
2.DATA ANALYSIS AND INTERPRETATION

The user could guess the word correctly in 72%(40/55) of the test runs. The program tends to produce long words more often than not .

The program victoriously predicts the word in 60%(12/20) of the cases.

3.VISUALISATION OF THE RESULTS:

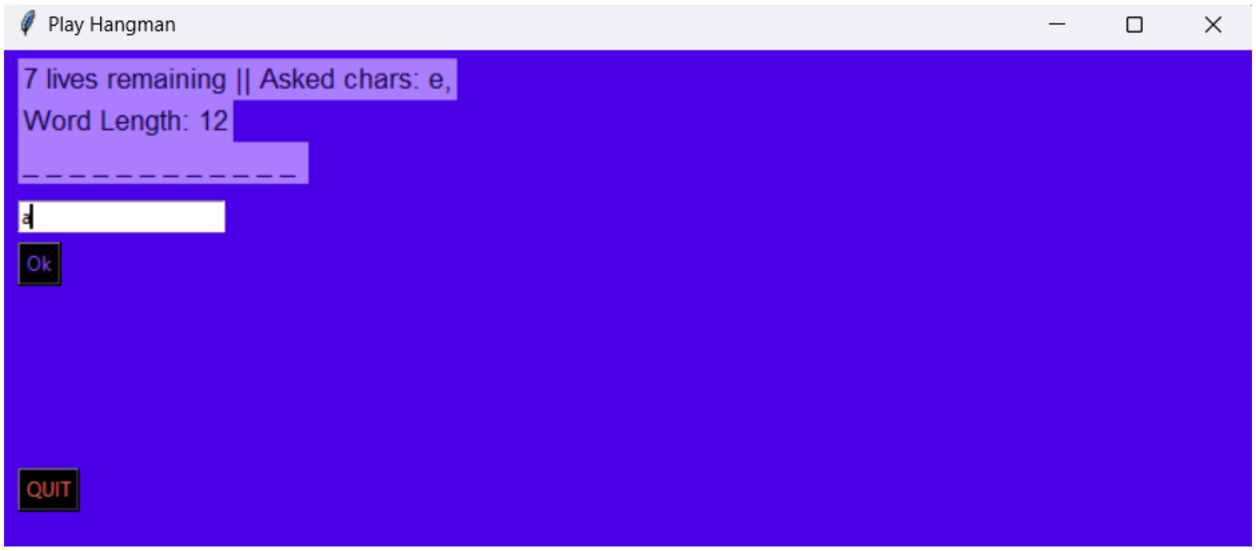
- User successfully guessing the word ‘cladosporium’.



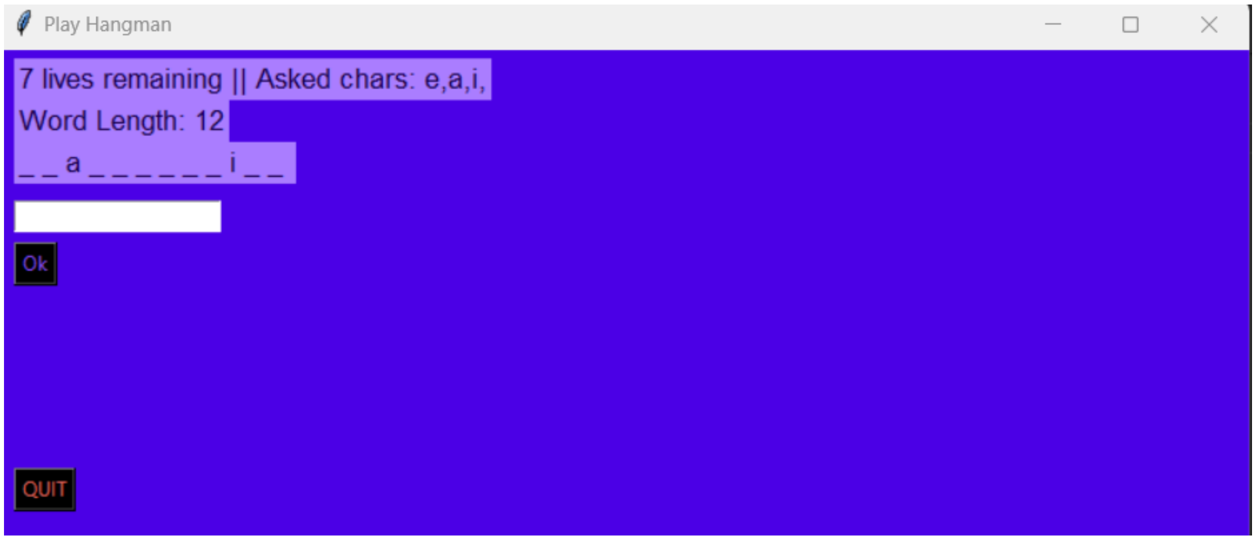
8 lives remaining || Asked chars:

Word Length: 12

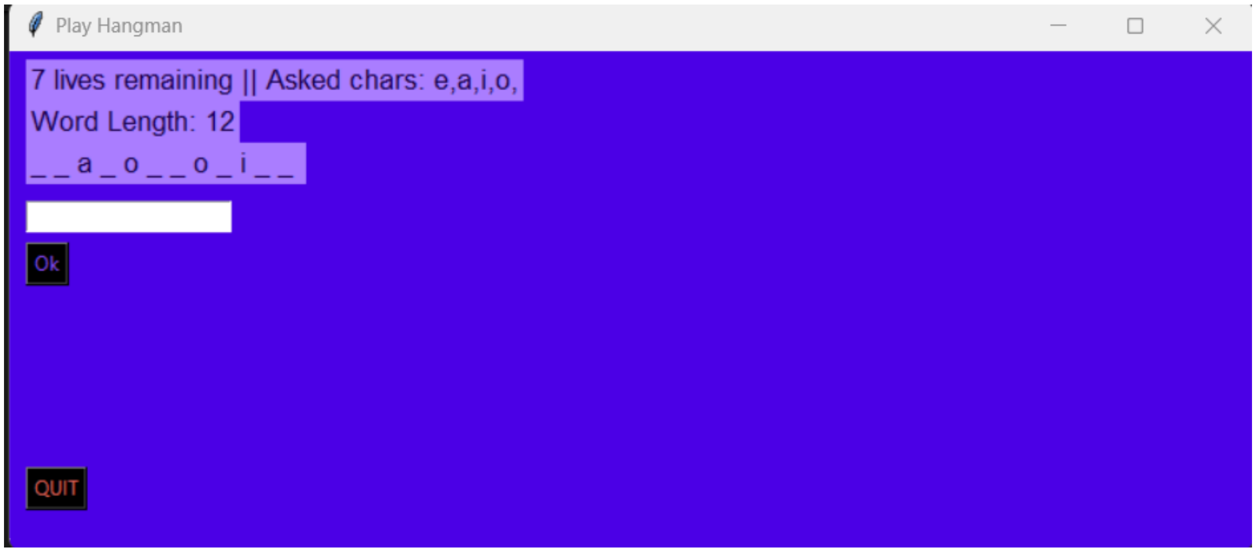
_ _ _ _ _ _ _ _ _ _ _ _



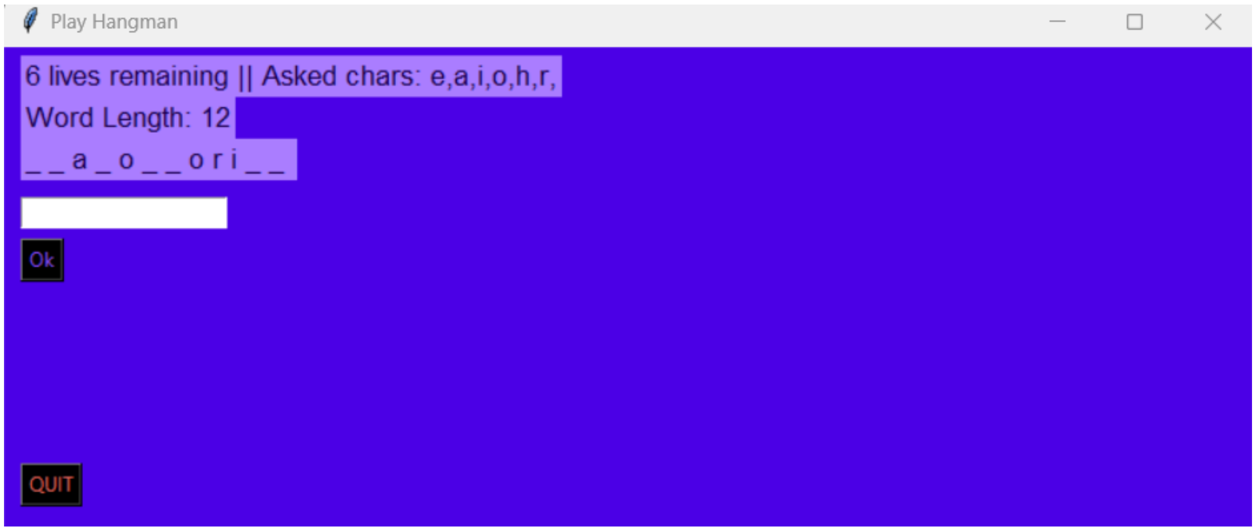
```
7 lives remaining || Asked chars: e,
Word Length: 12
_ _ _ _ _ _ _ _ _ _ _ _
a
```



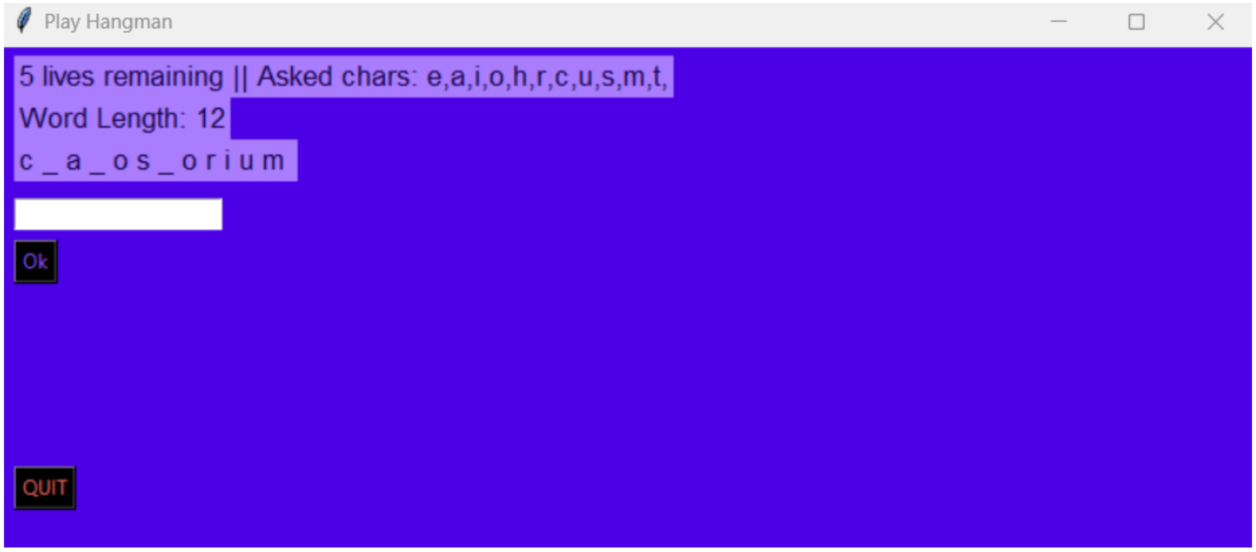
```
7 lives remaining || Asked chars: e,a,i,
Word Length: 12
_ _ a _ _ _ _ _ _ i _ _
```



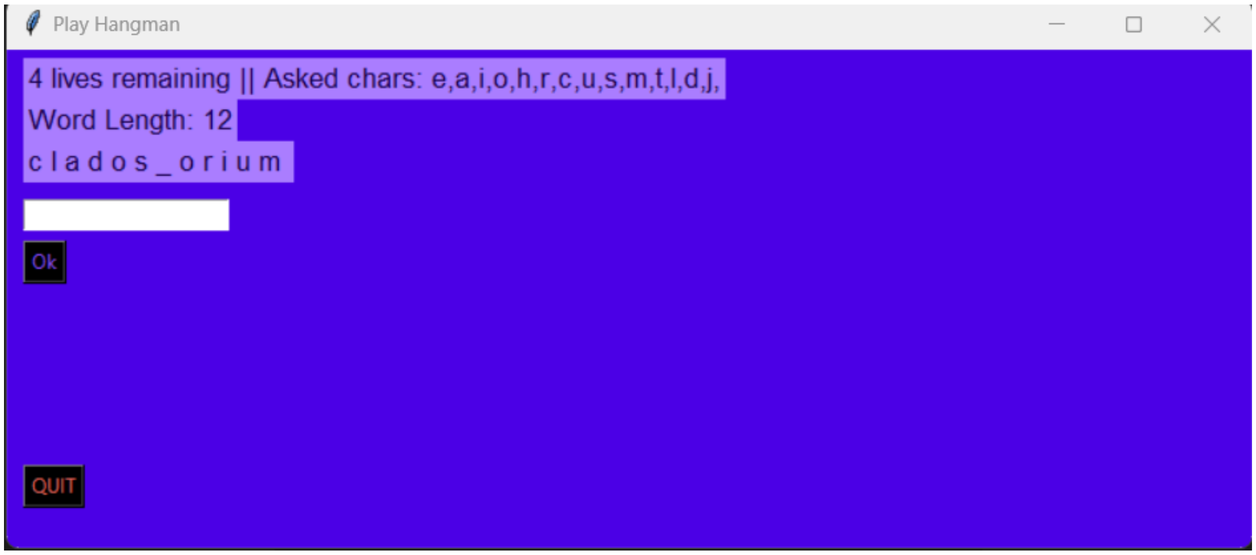
```
7 lives remaining || Asked chars: e,a,i,o,
Word Length: 12
_ _ a _ o _ _ o _ i _ _
```

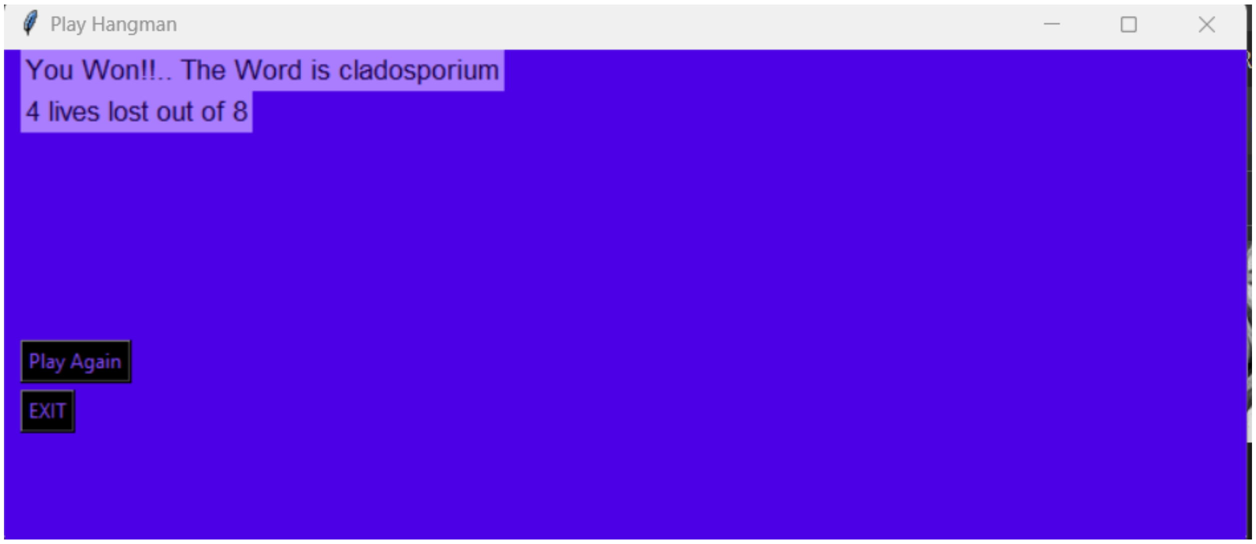


```
6 lives remaining || Asked chars: e,a,i,o,h,r,
Word Length: 12
_ _ a _ o _ _ o r i _ _
```

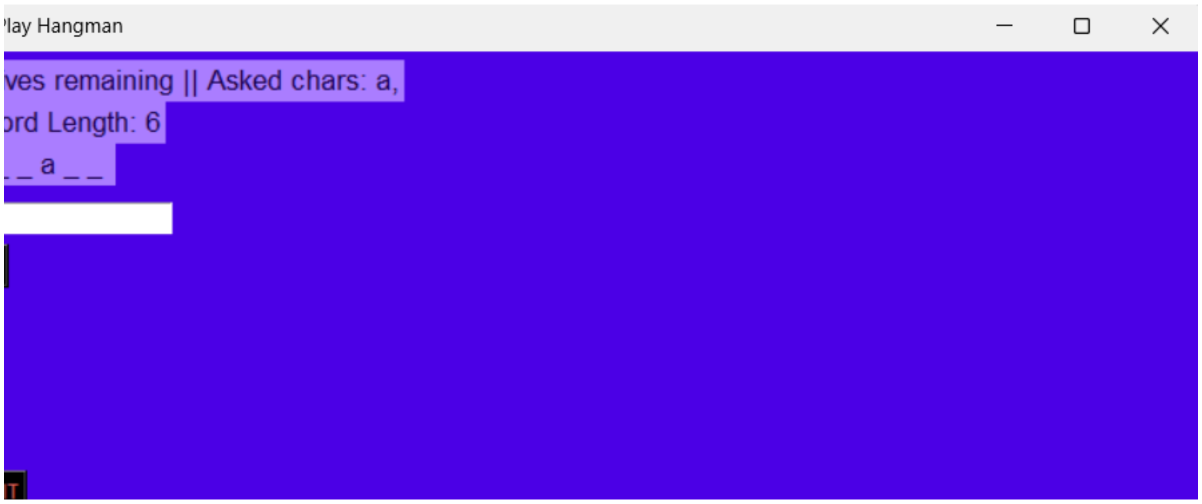


```
5 lives remaining || Asked chars: e,a,i,o,h,r,c,u,s,m,t,
Word Length: 12
c _ a _ o s _ o r i u m
```

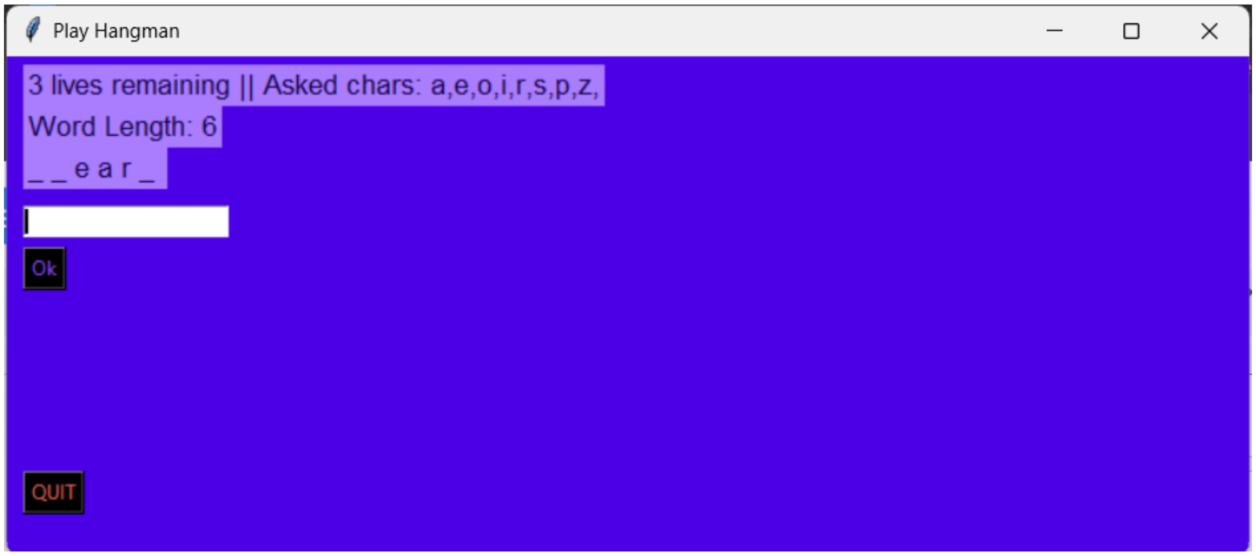

```
4 lives remaining || Asked chars: e,a,i,o,h,r,c,u,s,m,t,l,d,j,
Word Length: 12
c l a d o s _ o r i u m
```


```
You Won!!.. The Word is cladosporium
4 lives lost out of 8
```

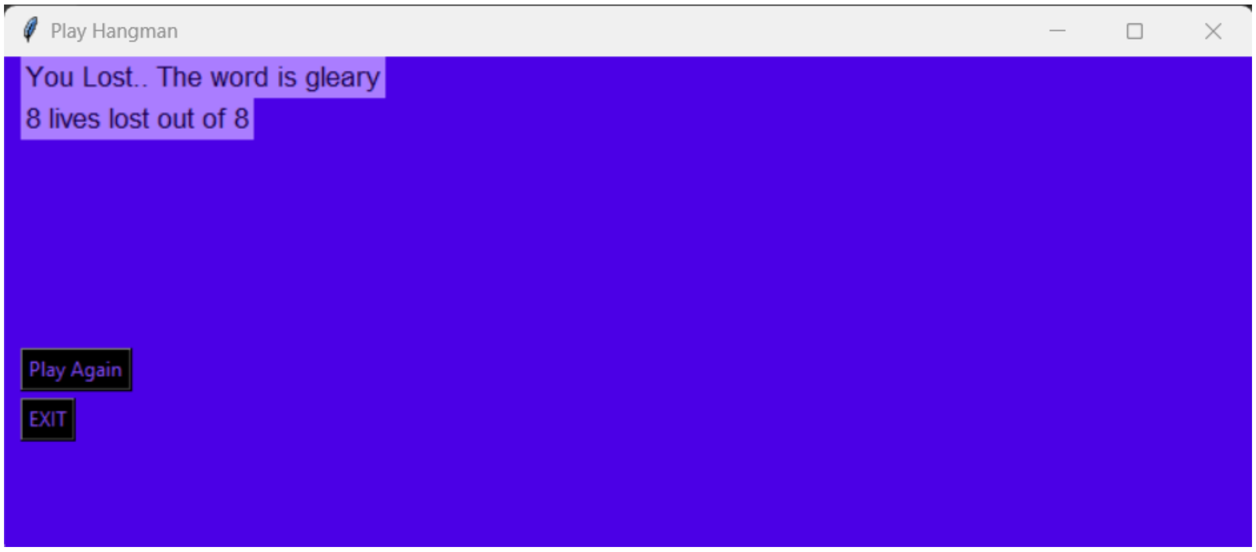
- User failing to predict ‘gleary’



```
'lay Hangman
ves remaining || Asked chars: a,
ord Length: 6
_ _ a _ _
```

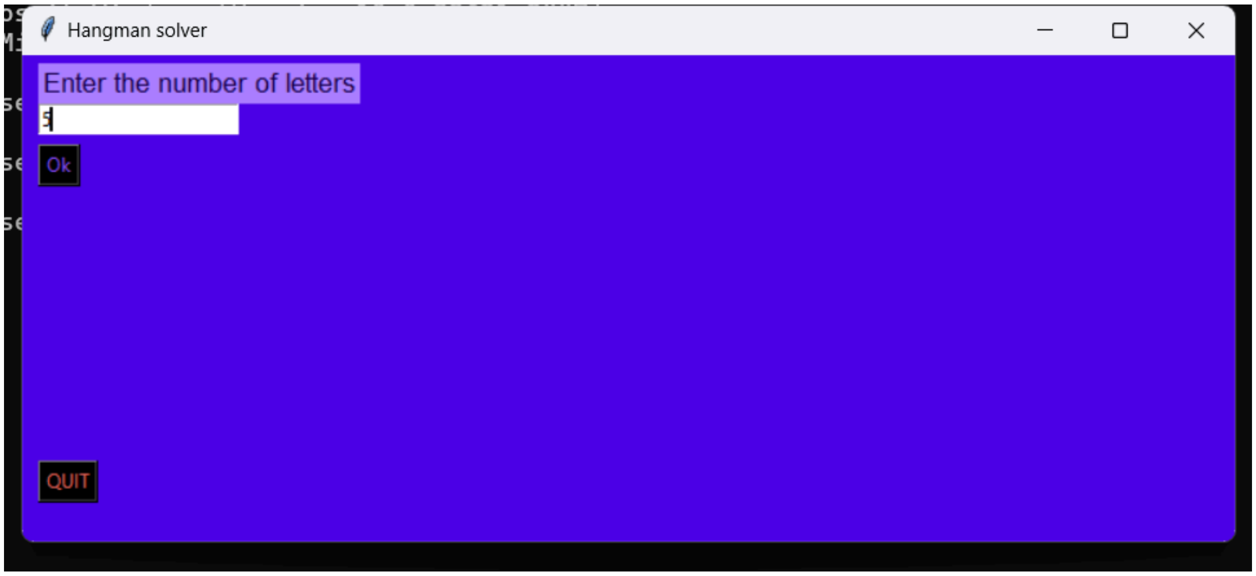


```
Play Hangman
3 lives remaining || Asked chars: a,e,o,i,r,s,p,z,
Word Length: 6
_ _ e a r _
```

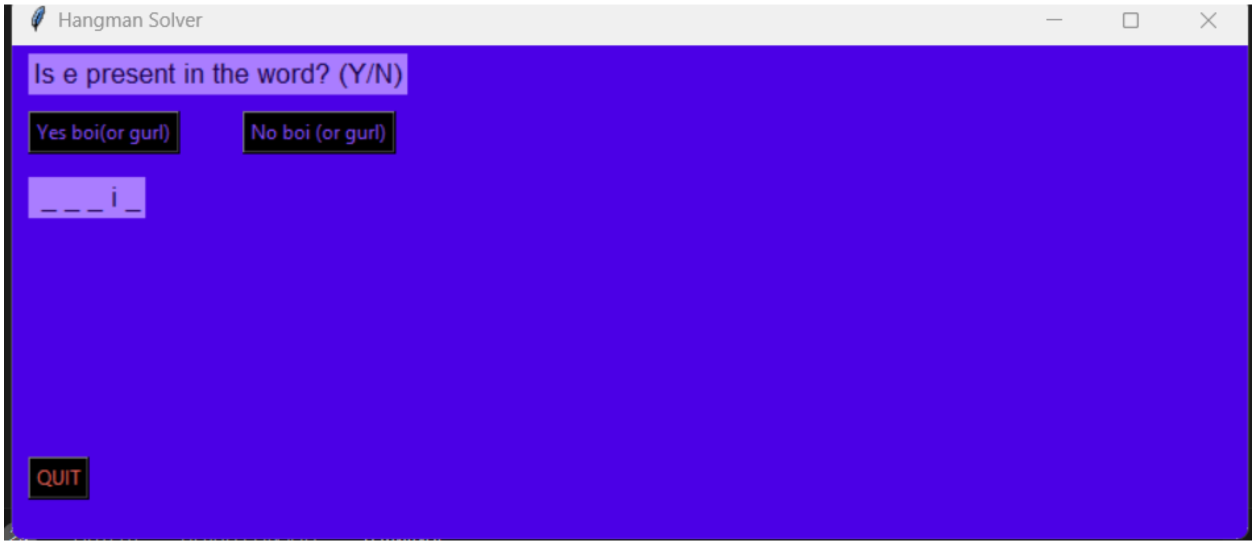


You Lost.. The word is gleary
8 lives lost out of 8

- Computer guessing the word ‘chair’ :



Enter the number of letters
5

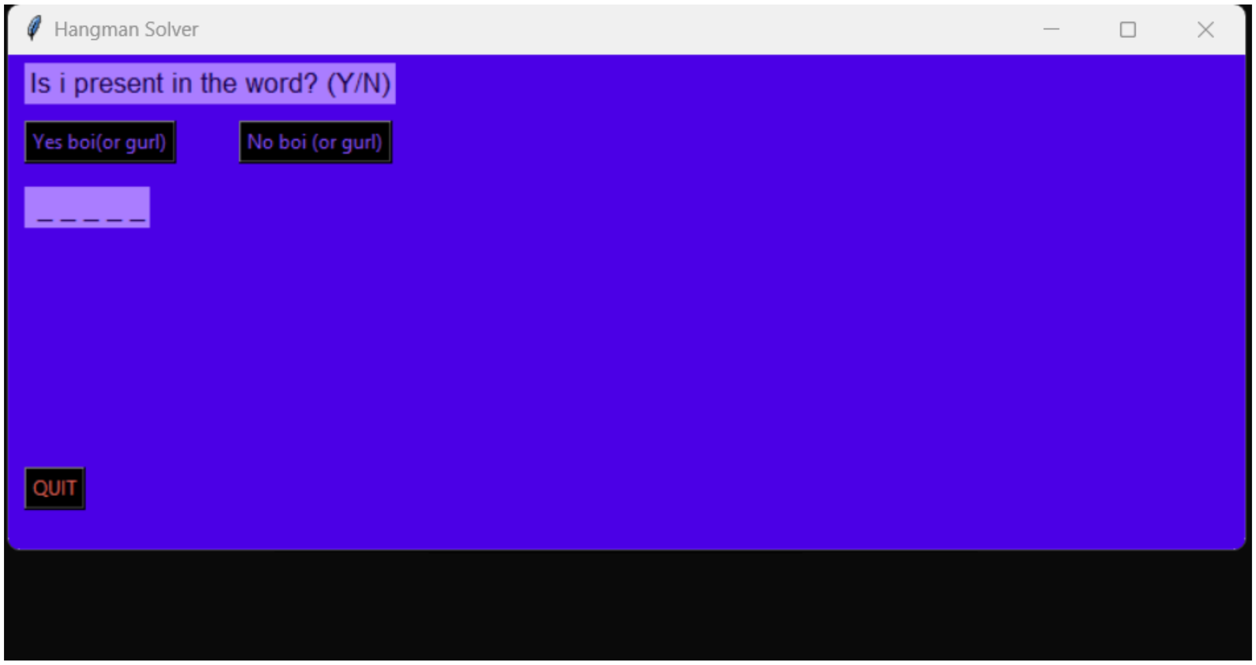

Hangman Solver

Is e present in the word? (Y/N)

Yes boi(or gurl) No boi (or gurl)

_ _ _ i _

QUIT

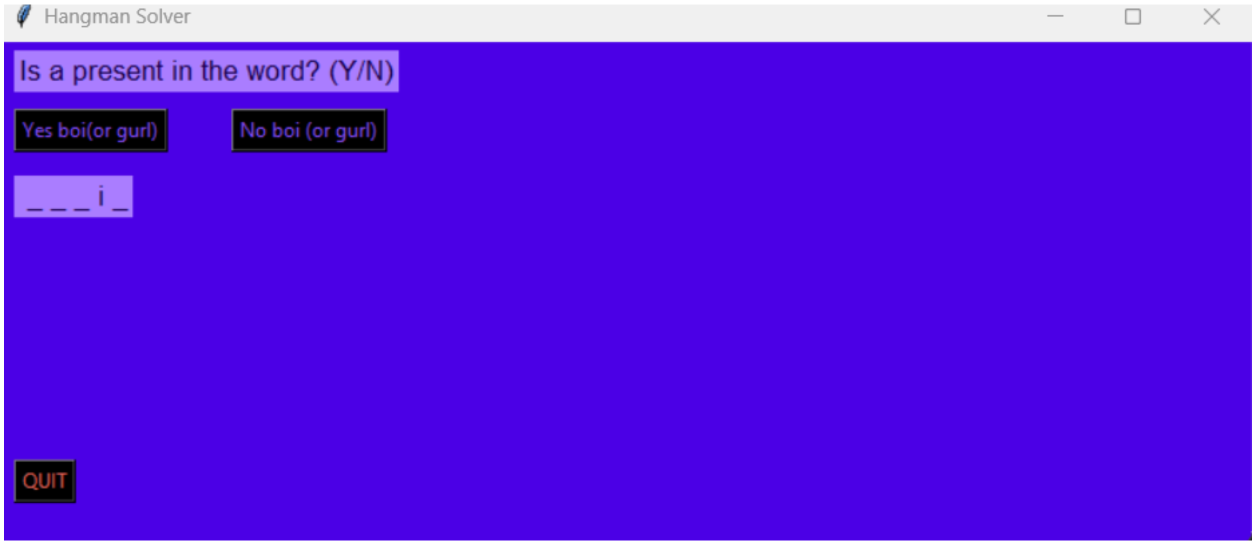

Hangman Solver

Is i present in the word? (Y/N)

Yes boi(or gurl) No boi (or gurl)

_ _ _ _ _

QUIT


Hangman Solver

Is a present in the word? (Y/N)

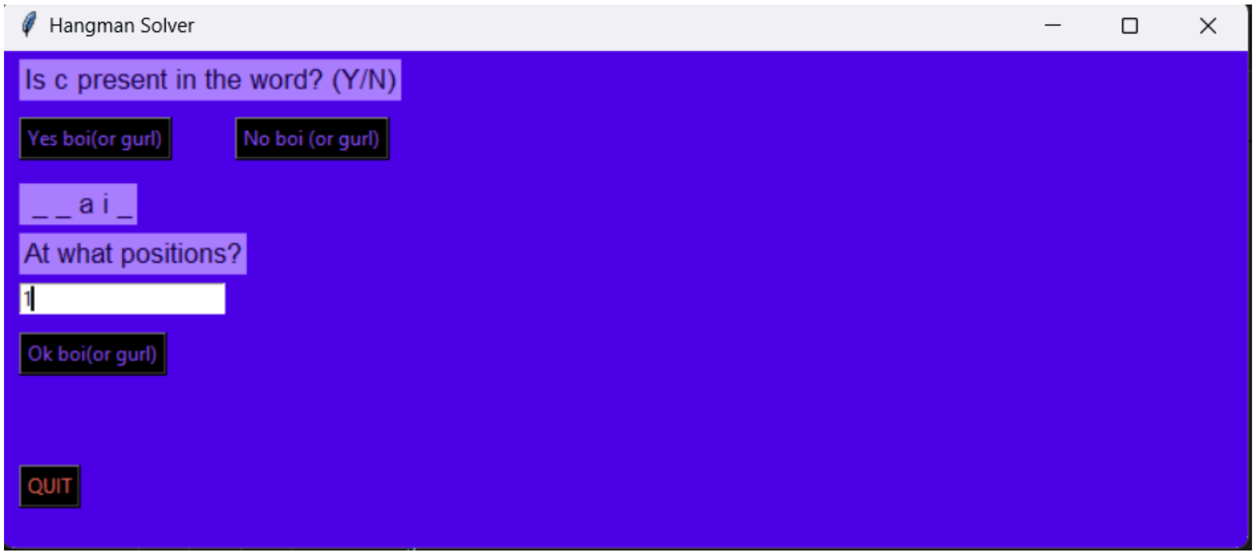
Yes boi(or gurl) No boi (or gurl)

_ _ _ i _

QUIT

Hangman Solver

Is o present in the word? (Y/N)

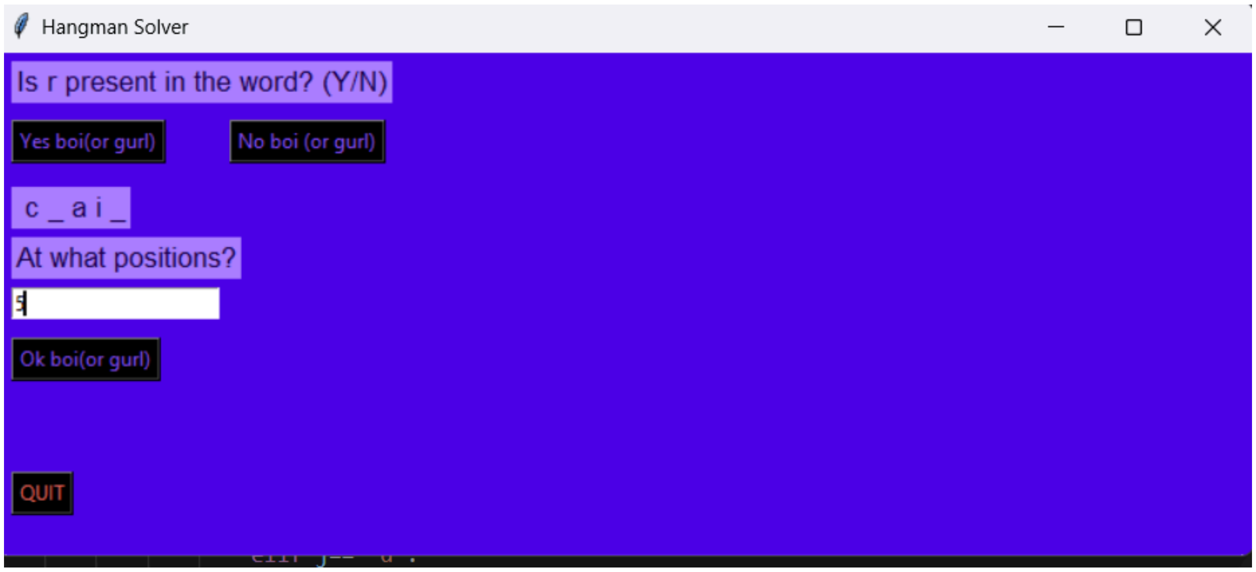


Is c present in the word? (Y/N)

_ _ a i _

At what positions?

1

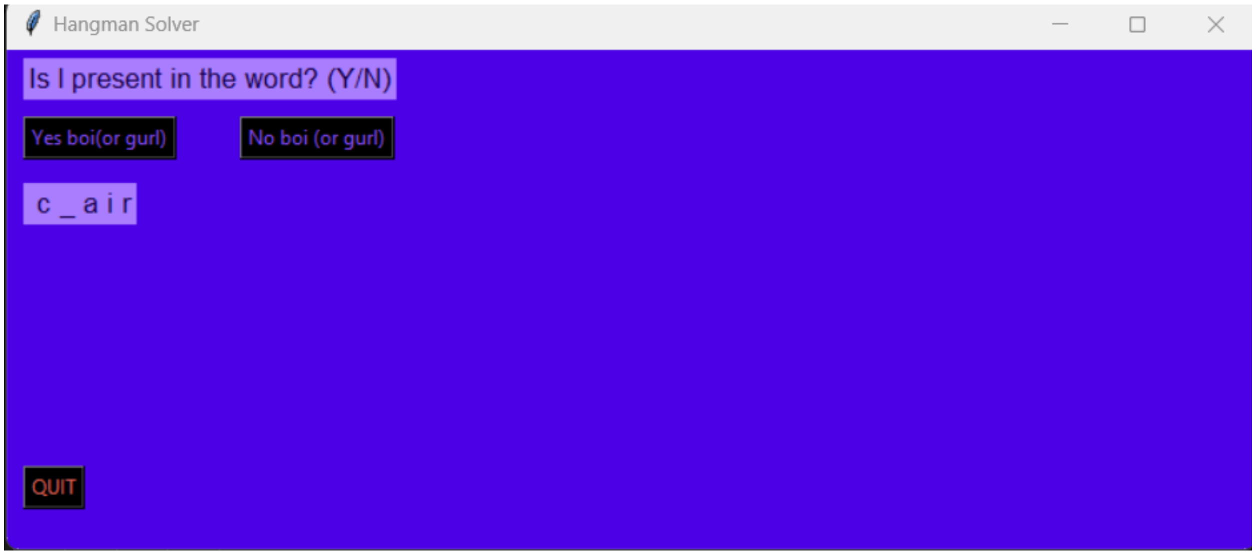


Is r present in the word? (Y/N)

c _ a i _

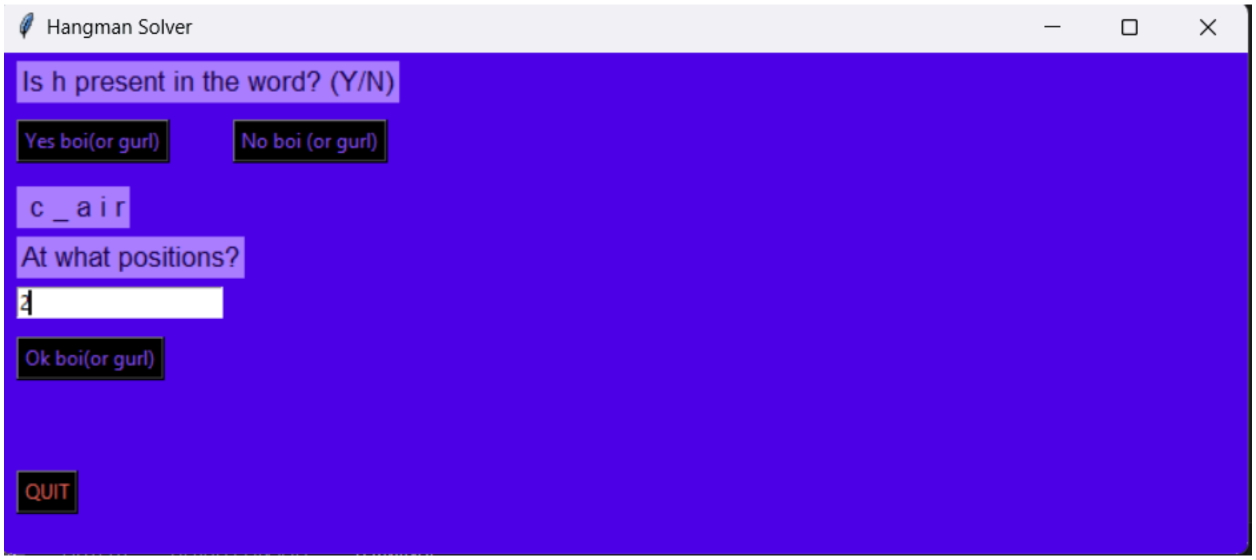
At what positions?

5



Is l present in the word? (Y/N)

c _ a i r

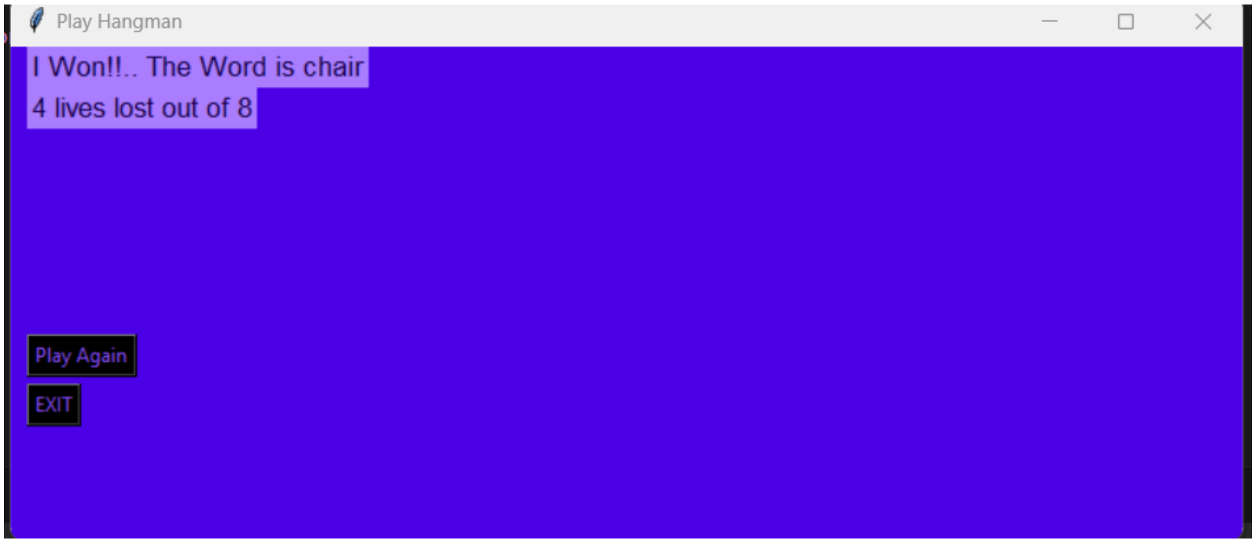


Is h present in the word? (Y/N)

c _ a i r

At what positions?

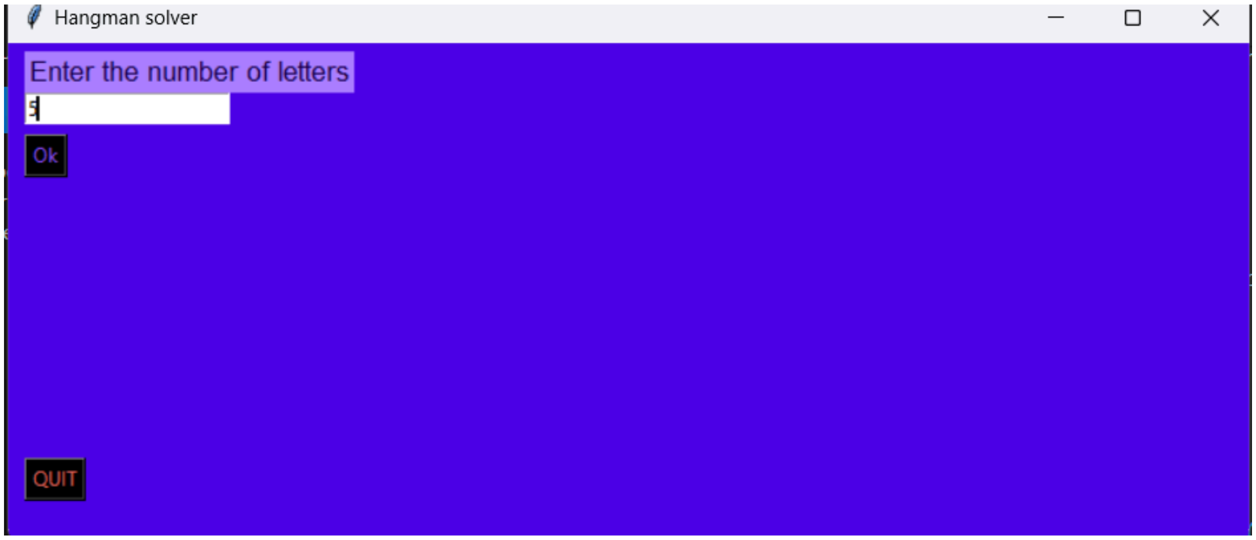
2


Play Hangman

I Won!!.. The Word is chair

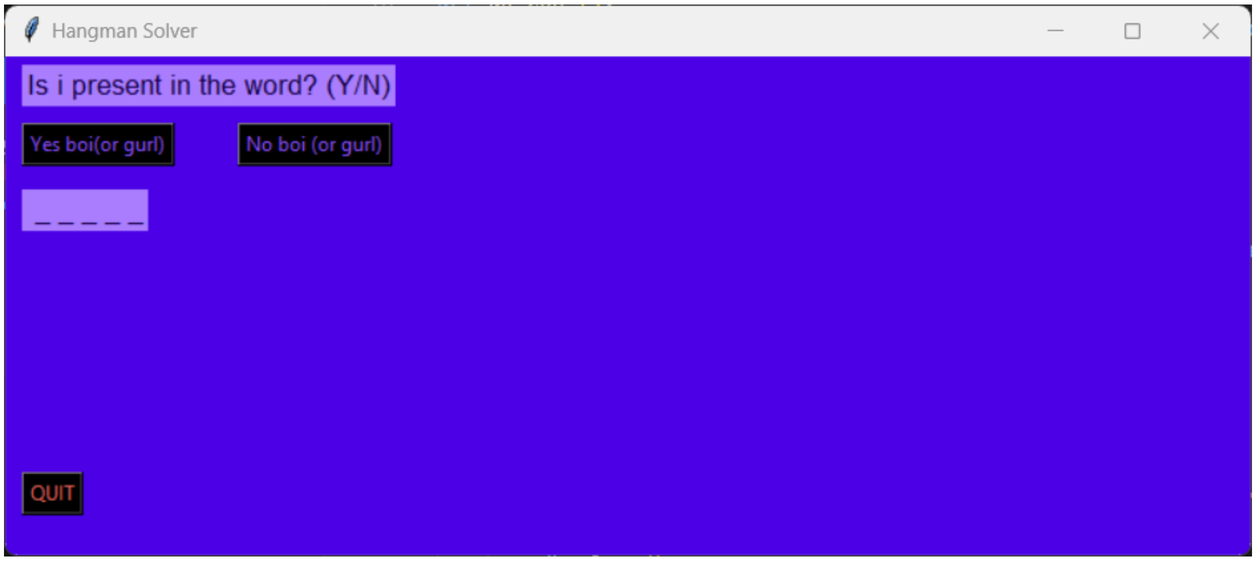
4 lives lost out of 8

- Computer failing to guess ‘crypt’

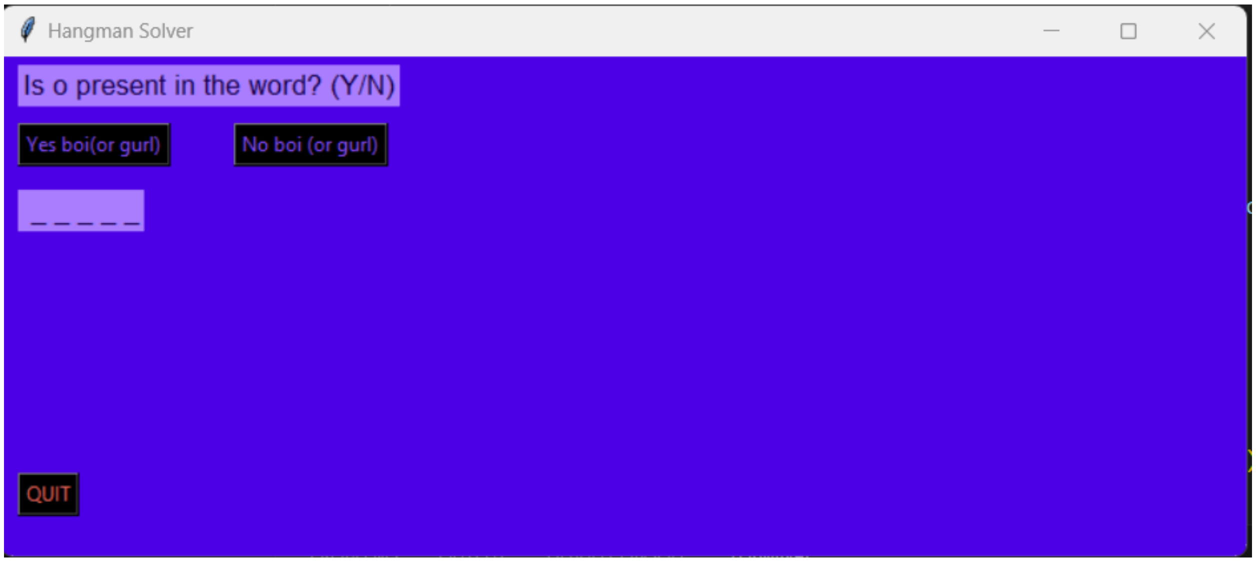

Hangman solver

Enter the number of letters

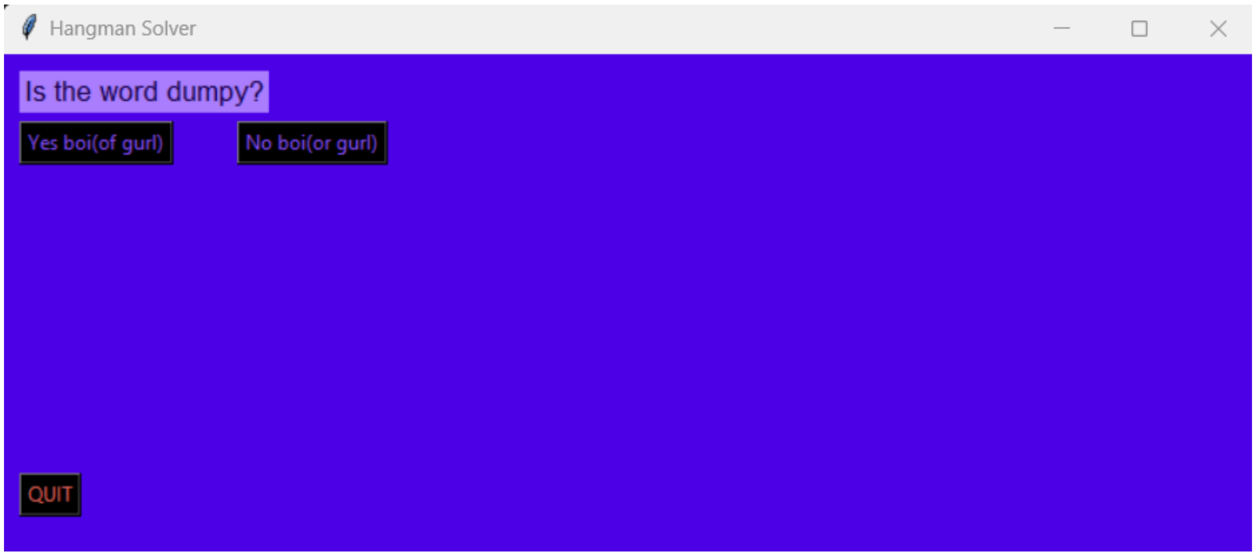
5



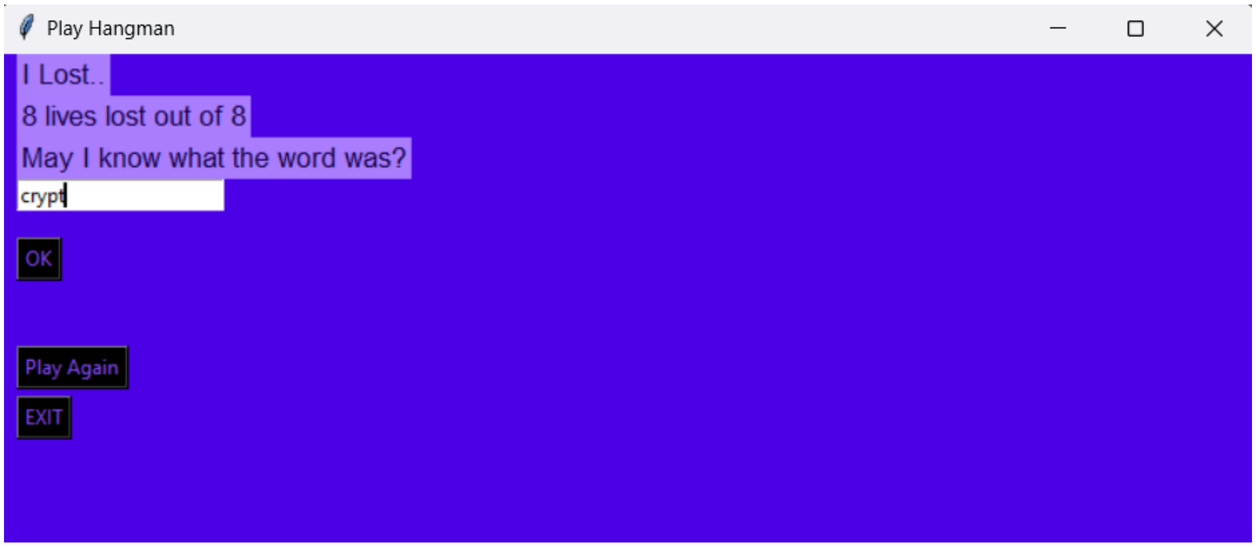
```
Is i present in the word? (Y/N)
_ _ _ _ _
```



```
Is o present in the word? (Y/N)
_ _ _ _ _
```



```
Is the word dumpy?
```



Play Hangman

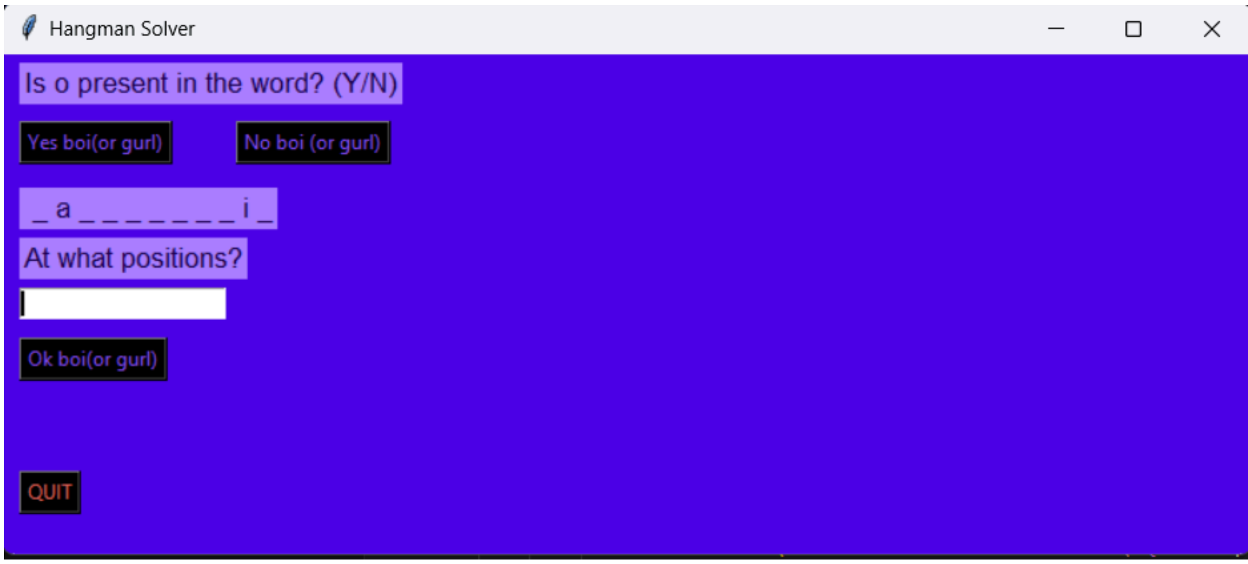
I Lost..

8 lives lost out of 8

May I know what the word was?

crypt

- Computer successfully guessing the word ‘haemoglobin’

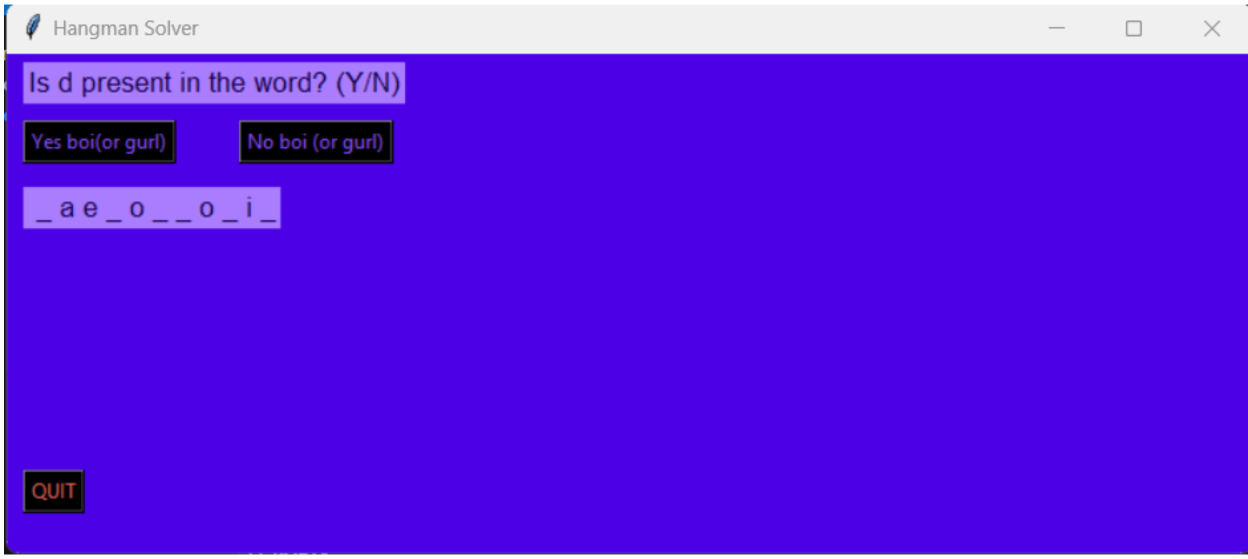


Hangman Solver

Is o present in the word? (Y/N)

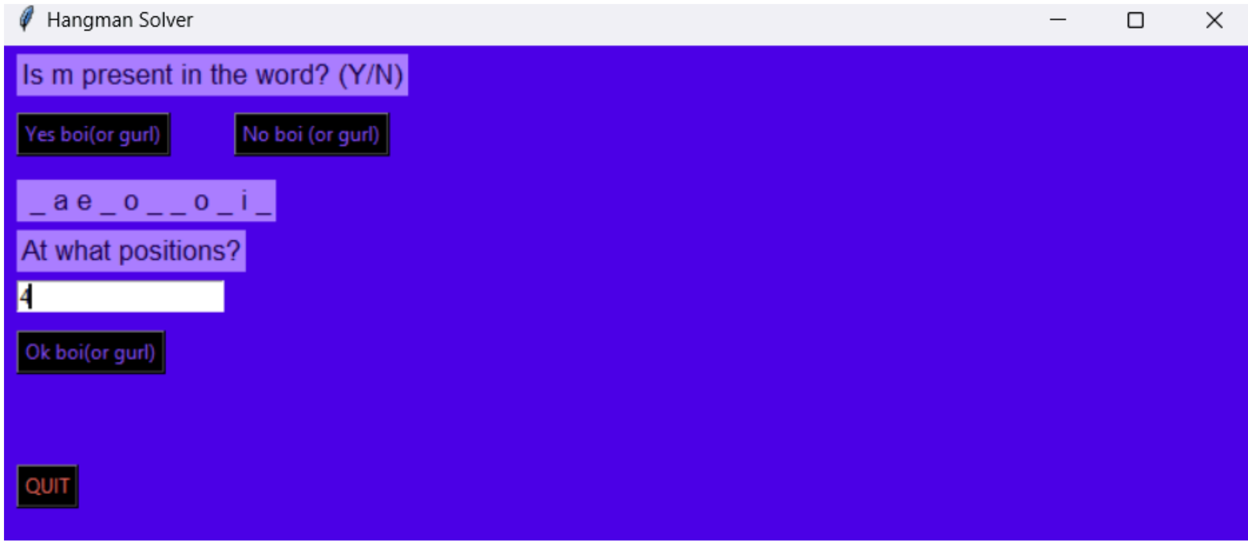
_ a _ _ _ _ _ _ _ i _

At what positions?



Is d present in the word? (Y/N)

_ a e _ o _ _ o _ i _

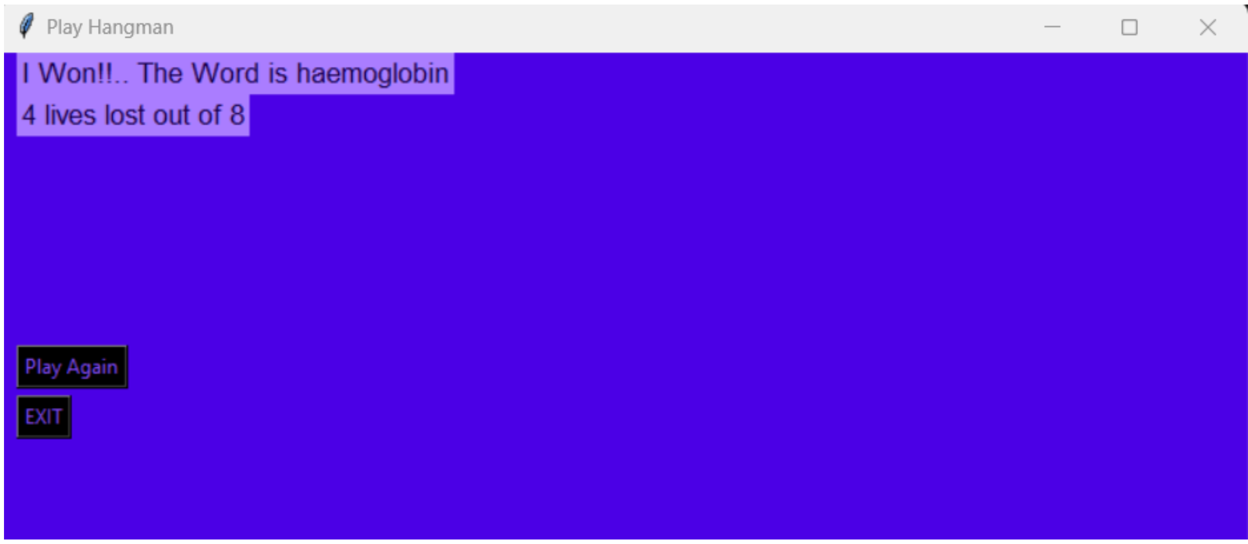


Is m present in the word? (Y/N)

_ a e _ o _ _ o _ i _

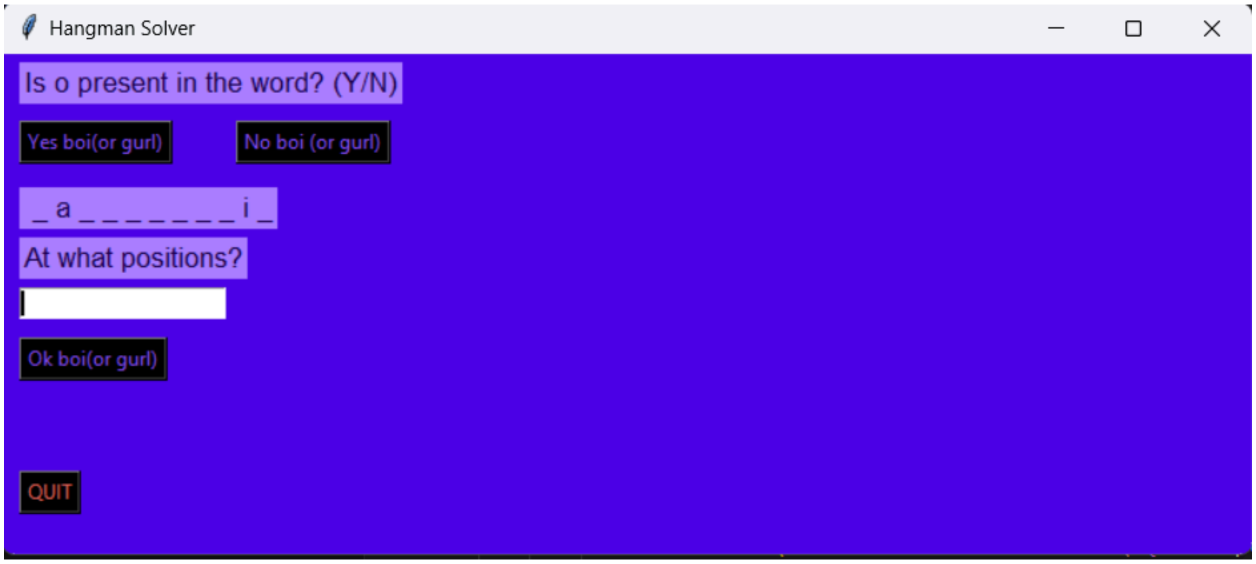
At what positions?

4



I Won!!.. The Word is haemoglobin

4 lives lost out of 8



CHALLENGES AND SOLUTIONS

The program has its own limitations while guessing smaller words (length of 2,3) since they contain only one or two vowel(s) and still poses a large set of unfiltered words even after the predicted skeleton.
It becomes difficult for the computer to predict the remaining consonants within eight lives since the algorithm prefers to go for vowels first.

We are in search of a better algorithm in which smaller words are efficiently predicted and to reduce the number of misdirected attempts.

5.CONCLUSION AND FUTURE WORK

1.SUMMARY OF OUTCOMES AND CONTRIBUTIONS

The program works well in setting up a gameplay for the user to guess each letter of the randomly picked word (by the computer), tracking the number of remaining attempts and to display the final status(win or loss), also the word of discussion . The GUI elements create a simple but interesting atmosphere of the interactive gameplay.

The mode in which the program predicts the word is a bit challenging but never fails without giving a close competition.It is satisfactory to observe that the program ends up in a win with an ample count of lives in hand.

2.ASSESSMENT OF PROJECT SUCCESS

It is contented to present a simple, traditional but compelling game like hangman in the digital platform with two gaming modes. We tried our level best in improving the logistics of the game and to provide a user friendly interface. The main objectives of coming up with a gaming experience to the user when he/she plays the dice and upgrading the credibility of the computer predictions are accomplished.

3.LESSONS LEARNED AND FUTURE IMPROVEMENTS

In the process of making this program we were able to explore database management, GUI development using tkinter and also delve deeper into multi-file programming. The roles of each file was fixed(database management, gui, processing,etc).

However in future versions we plan to make the individual files more independent so that making changes to the program would be more seamless. We also plan on coming up with a better algorithm for computer to guess the word by using methods that are statistically more sound.

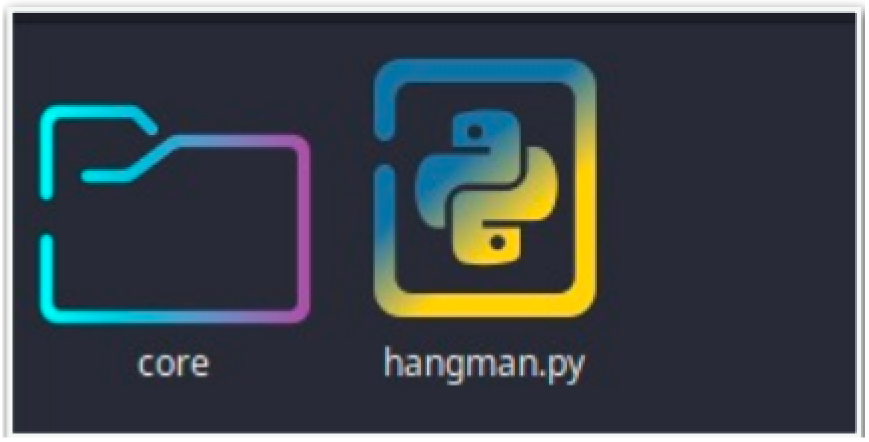
Team member's GitHub Accounts:
Karthik Krishna M CS-https://github.com/KarthikKrishnaMCS
Sanjay Siva A-https://github.com/SanjaySivaA
Madhav P Nair-https://github.com/madhavpnair

6.REFERENCES

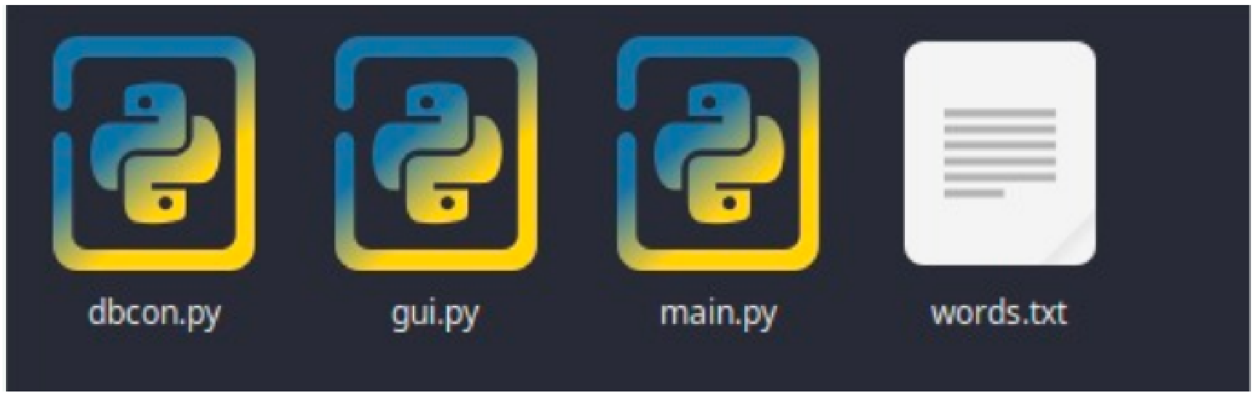
- Python documentation on Tkinter
- MySQL documentation
- Documentation on mysql python connector
- Python documentation on PEP 8 style of coding
- Hangman-Wikipedia

FOLDER STRUCTURE AND FILES

hangman - core ,hangman.py



core - dbcon.py ,main.py, gui.py, words.txt


dbcon.py
gui.py
main.py
words.txt