**HANGMAN**

A MINI-PROJECT REPORT

Submitted by

SIDDHARTH.H 180701238

sivaprakash s 180701239

***in partial fulfillment of the award of the degree***

***of***

**BACHELOR OF ENGINEERING**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

**RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI**



**MAY -2019**

**ANNA UNIVERSITY**

**CHENNAI**

**BONAFIDE CERTIFICATE**

Certified that this project **“HANGMAN ”**is the bonafide work of **“SIDDHARTH.H and SIVAPRAKASH.S ”** who carried out the project work under my supervision.

|  |  |
| --- | --- |
| **SIGNATURE** | **SIGNATURE** |
| **Dr.P.KUMAR M.E Ph.D.,** | **Mr.M.CHITHAMBARATHANU** |
| **HEAD OF THE DEPARTMENT** | **ASSISTANT PROFESSOR(SS)** |
| Dept. of Computer Science and Engg, | Dept. of Computer Science and Engg, |
| Rajalakshmi Engineering College,Chennai | Rajalakshmi Engineering College,Chennai |

Submitted for the **ANNA UNIVERSITY** practical examination Mini-Project work viva voce held on \_\_\_\_\_\_\_\_\_\_

**INTERNAL EXAMINER** **EXTERNAL EXAMINER**

**ABSTRACT**

Hangman is a paper pen game usually played by two or more people where one of them gives the question and the rest try to find the answer to it.

The word to be guessed is represented in the form of dashes representing each letter of the word to be guessed, the word to be guessed can be anything.

But in this computerized version of ours only 1 person can play at a time and the user has a choice to either play it the normal mode or can play it in the hard mode.

In order to challenge the creativity of the user we have made sure not to give any hints in the hard mode while, in the normal mode the class of the word is to be chosen by the user himself.

In order to rank the user we made use of totally different ranking system where the user gets grade instead of usual point tally.

This game helps the user to improve their vocabulary skills.

ACKNOWLEDGEMENT

We express our sincere thanks to our beloved and honorable chairman **MR. S. MEGANATHAN** and the chairperson**. M.THANGAM MEGANATHAN** for their timely support and encouragement

We are greatly indebted to our respected and honorable principal **Dr. S.N. MURUGESAN** for his able support and guidance

No Words Of Gratitude Will Suffice For The Unquestioning Support Extended To Us By Our Head Of The Department **Dr.P.KUMARM.E Ph.D.,** For Being Ever Supporting Force During Our Project Work

We also extend our sincere and hearty thanks to our internal guide **Mr.M.CHITHAMBARATHANU B.E.,M.Tech.,**for his valuable guidance and motivation during the completion of this project.

Our sincere thanks to our family members, friends and other staff members of information technology.

1. Siddharth.H
2. STUDENT NAME

**TABLE OF CONTENTS**

CHAPTER NO. TITLE PAGE

**ABSTRACT iv**

**1 INTRODUCTION 1**

1.1 introduction 1

1.2 SCOPE OF THE WORK 6

1.3 PROBLEM STATEMENT 6

1.4 AIM AND OBJECTIVES OF THE PROJECT 7

**2 SYSTEM SPECIFICATIONS 46**

2.1 HARDWARE SPECIFICATIONS 46

2.2 SOFTWARE SPECIFICATIONS 47

**3 MODULE DESCRIPTION 50**

**4 SYSTEM DESIGN 55**

4.1 ARCHITECTURE DIAGRAM 55

4.2 FLOW CHART 57

**5CODING 60**

**6 SCREEN SHOTS 65**

**7CONCLUSION AND FUTURE ENHANCEMENT 70**

**REFERENCES**

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIGURE NO** | **FIGURE NAME** | **PAGE NO.** |

**CHAPTER 1**

**INTRODUCTION**

* 1. INTRODUCTION

Hangman was game which was introduced long before the computers and other electronic devices were invented. Its main intent was to make the player improve in his/her vocabulary skills and enjoy the game at the same time.

The main constraint on it was that the player was only given limited chances which usually tensed them up really quickly. The main reason it lost its popularity is because of its several variations which were developed over the years by the players to make it more interesting.

Since the development of the computer the rules of the game were changed to make it seem different every. The original version of the game had no rule s on the number of tries the user could make. This game gives a throwback to the original version of the game where the player was given infinite chances to think on the given word but as he or she made more wrong guesses his or her points were reduced considerably.

* 1. **SCOPE OF THE WORK**

**The scope of the work is to improve the vocabulary skills of the player in a more enjoyable manner. This also can be used as a method of treatment for people with ADD (Attention-Deficit-Disorder).ADD is present in about 2-3% of the children in the world and this game is directed towards them to improve their attentiveness in a fun way.**

* 1. **PROBLEM STATEMENT**

**This game is so plain old and simple that it needs no rules to be written and laid down its very easy to play and due to the attractive sounds it makes it further indulges the player thus helping them to improve their attention and their hand eye coordination and their vocabulary skills at the same time.**

**As a whole it is a fun game that allows the user to overcome their ADD or ADHD (if they have any since it is getting very common all over the world)**

**1.4 AIM AND OBJECTIVES OF THE PROJECT**

**The main aim of the project is to entertain the player and keep him/her busy building up their vocabulary skill set. The player could also learn a new word by playing this game which will further improve his/her speech patterns and it would also help them to reduce the number of spelling mistakes that they make while spelling out the word.**

**Thus the main objective behind this game is to improvise the attention, vocabulary skills and could also act as a time pass agent which would be profitable for the player.**

**This game could also help the player to mainly entertain himself in many ways and hence could act as a serious time killing agent for players of any day and age.**

**CHAPTER 2**

**SYSTEM SPECIFICATIONS**

**2.1 HARDWARE SPECIFICATIONS**

|  |  |  |
| --- | --- | --- |
| Processor Type | **:** | INTEL(R) Pentium(R) 64 bit |
| Memory Size | **:** | 256 GB (Minimum) |
| HDD | **:** | 40 GB (Minimum) |

RAM : 4GB

**2.2 SOFTWARE SPECIFICATIONS**

|  |  |  |
| --- | --- | --- |
| Operating System | **:** | WINDOWS 07 Or 10 |
| Front – End | **:** | Python 3.4 32 bit |
| Back – End | **:** | **Notepad** |
| Language | **:** | Python |
| Open Source Software | **:** | Windows Media Player |
| Video File Extension | **:** | NIL |

Audio File Extension : .wav

**CHAPTER 3**

**MODULE DESCRIPTION**

**Several inbuilt modules of python like random, Winsound, OS, Getpass are used along with a downloaded package module of colorama of 32 bit.**

**1.RANDOM : This is an inbult module present in python which is used for the random string selection purpose. It serves its main role in giving out the question to the user each time he or she plays the game.**

**2.WINSOUND: This is also an in built module in python package which is used to play the .wav tunes played during the gameplay without disturbing the user via pop ups of the vlc media player to play the sounds.**

**3OS: OS or the operating system module is used for the clear screen function which is called during the execution.**

**4.GETPASS: This is yet again an inbuilt module that is used to get the password from the user without echoing at all . It is used in places where the user has to type in the password to login.**

**5.COLORAMA: This is an online package that is downloaded from the command prompt to support the coloration of the text and the background color in some cases in the python terminal when the game is being played by the user .**

**The above listed are the modules that are used to run the game smoothly. Also if the system does not have colorama module pre-installed then the game won’t run and will keep showing up errors up until colorama is installed from the command prompt.**

**CHAPTER 4**

**SYSTEM DESIGN**

**4.1 ARCHITECTURE DIAGRAM**

Unable to LOGIN

LOGIN OR REGISTER

If Selected Hard mode

If Selected Easy mode

Allow the user to select the category

Give the question to the user in form of dashes until the word is found

Allow the user to enter a letter if it is present in the word continue

Replay the game

Else print that the letter user wrote does not exist in the word

Print that the user found the word and rank him ask the user to play again or quit

If the user quit give him his average rank for his game play

**4.2 FLOW CHART**

Get choice

Register

Login

Get choice for mode

Hard mode

Easy mode

B

A1

B

Get category

A

Generate the question using random module in terms of dashes

Return list from category

D

Get user choice alphabet

While the word is not found

If input in word

Print user choice is wrong

Increment point

Print user choice is correct

D

Get choice to play or quit

Rank the player

E

Provide the average rank of the user

E

**CHAPTER 5**

**SAMPLE CODING**

**#to make use of the "random.choice" function**

**import random**

**import winsound**

**import getpass**

**#importing colorama**

**from colorama import\***

**init()**

**#list of words in easy mode**

**class level\_easy:**

**def animals(self):**

**return ['mouse','rhino','lion','tiger','leopard','jaguar','panther']**

**def birds(self):**

**return ['owl','dove','parrot','flamingo','sparrow','crow','parakeet']**

**def tree(self):**

**return ['banyan','eucalyptus','coconut','palm',]**

**def fruits(self):**

**return['apple','orange','peach','banana','mango','strawberry','fig']**

**def transport(self):**

**return ['plane','train','jet','tank','submarine','jeep','bike','tonga']**

**def beverages(self):**

**return ['tea','coffee','pepsi','coke','boost']**

**def occupation(self):**

**return ['wrestler','engineer','architect','doctor','surgeon','manager','security','police','army','navy']**

**#list of words for the hard mode**

**class level\_hard:**

**def words(self):**

**return['mouse','rhino','lion','tiger','leopard','jaguar','panther','wrestler','engineer','architect','doctor','surgeon','manager','security','police','army','navy','apple','orange','peach','banana','mango','strawberry','fig']**

**import os**

**def cls():**

**os.system('CLS')**

**def questions(a):**

**global point**

**point=0**

**wletter=[]**

**answer=list(a)**

**word=[]**

**t=[]**

**wcount=len(answer)**

**t.extend(answer)**

**kn=0**

**for i in range (len(t)) :**

**t[i]=" \_ "**

**print(' '.join(t))**

**while kn<len(answer):**

**print()**

**gans=input(Fore.GREEN+"pls enter a alphabet ")**

**if gans in wletter:**

**print(Fore.WHITE,Back.RED+"you already wrote this alphabet")**

**print(Fore.RESET,Back.RESET)**

**winsound.PlaySound("smb\_warning.wav",winsound.SND\_ASYNC)**

**continue**

**else:**

**gans=gans.lower()**

**wletter.append(gans)**

**if gans not in answer:**

**point=point+1**

**print(Fore.WHITE,Back.RED+"oops you guessed wrong")**

**print(Fore.RESET,Back.RESET)**

**winsound.PlaySound("smb\_mariodie.wav",winsound.SND\_ASYNC)**

**else:**

**print(Fore.RED,Back.CYAN+"you guessed right")**

**print(Fore.RESET,Back.RESET)**

**cls()**

**for i in range(wcount):**

**if answer[i]==gans:**

**t[i]=gans**

**kn=kn+1**

**word=' '.join(t)**

**print(word)**

**print()**

**print(Fore.RED,Back.YELLOW+"you win")**

**print(Fore.GREEN+"you made ",point,"wrong choice(s)")**

**print(Fore.RESET,Back.RESET)**

**print(Fore.GREEN+"| | / \ |\ | \_\_\_\_\_ /\ /\ /\ |\ |")**

**print(Fore.GREEN+"|\_\_\_\_| / \ | \ | / / \ / \ / \ | \ |")**

**print(Fore.GREEN+"| | /\_\_\_\_\_\_\_\_\ | \ | \_ / \/ \ /\_\_\_\_\_\_ \ | \ |")**

**print(Fore.GREEN+"| |/ \| \| \\_\_\_| | / \ / \ | \|" )**

**winsound.PlaySound("smb\_world\_clear.wav",winsound.SND\_ASYNC)**

**print()**

**print()**

**print("press enter")**

**derp=input()**

**print("WELCOME TO HANGMAN")**

**def login():**

**un=input("enter your name \t")**

**pw=getpass.getpass("enter the pw ")**

**file=open("games.txt","r")**

**a=file.readlines()**

**file.close()**

**global pwd**

**pwd=1**

**for line in a:**

**login\_info=line.split()**

**if(un in login\_info[0] and pw in login\_info[1]):**

**pwd=1**

**break**

**else:**

**pwd=0**

**if pwd==1:**

**print(Back.GREEN+"logged in")**

**print(Back.RESET)**

**winsound.PlaySound("smb\_stage\_clear.wav",winsound.SND\_ASYNC)**

**global kk**

**kk=1**

**fummy=input()**

**cls()**

**else:**

**print(Back.RED+"unable to login")**

**print(Back.RESET)**

**winsound.PlaySound("smb\_mariodie.wav",winsound.SND\_ASYNC)**

**fummy=input()**

**cls()**

**#login register for new user**

**def regester():**

**print("REGISTER")**

**un=input("enter your name pls\t")**

**pw=input("enter the designated password\t")**

**file=open("games.txt","a")**

**file.write(un)**

**file.write("\t ")**

**file.write(pw)**

**file.write("\n")**

**file.close()**

**login()**

**print("enter 1 to login 2 to regester")**

**thing=int(input("make your choice"))**

**if thing==1:**

**login()**

**else:**

**regester()**

**if kk==1:**

**x=level\_easy()**

**y=level\_hard()**

**words=[]**

**print("1. for easy mode")**

**print("2.for hard mode")**

**kt=int(input("please make your choice"))**

**#for grading of the player**

**class List:**

**def \_\_init\_\_(self):**

**#determining the no. of turns**

**self.lista=[5,7,10,13]**

**#grading system**

**self.listb=['S','A','B','C','D']**

**def ranklist (self,t):**

**if self.lista[-1]<=t:**

**print(Fore.WHITE,Back.RED+"you came at the last position")**

**print("your garde is",self.listb[-1])**

**for i in self.lista:**

**if i>t:**

**pl=self.lista.index(i)**

**print("you are given",self.listb[pl] ,"rank :3")**

**break**

**lis=List()**

**class STACK:**

**def \_\_init\_\_(self):**

**self.stack=[]**

**def push(self,no):**

**self.stack.append(no)**

**def pop(self):**

**return self.stack.pop()**

**def length(self):**

**return len(self.stack)**

**stack=STACK()**

**stack\_value=1**

**#if user enters y he/ she cann play else they quit the game and get their pionts**

**while True:**

**print(Fore.GREEN+"to get the question press- Y-,to quit press any other button")**

**choice=input()**

**#user can make choice in the easy mode**

**if choice=="Y"or choice=="y":**

**if kt==1:**

**print(Fore.GREEN+"please enter :")**

**print(Fore.GREEN+"1.for animals")**

**print(Fore.GREEN+"2.for birds")**

**print(Fore.GREEN+"3.for trees")**

**print(Fore.GREEN+"4.for fruits")**

**print(Fore.GREEN+"5.for transport")**

**print(Fore.GREEN+"6.for beverages")**

**print(Fore.GREEN+"7.for occupation")**

**p=int(input())**

**if p==1:**

**words=x.animals()**

**elif p==2:**

**words=x.birds()**

**elif p==3:**

**words=x.tree()**

**elif p==4:**

**words=x.fruits()**

**elif p==5:**

**words=x.transport()**

**elif p==6:**

**words=x.beverages()**

**elif p==7:**

**words=x.occupation()**

**k=random.choice(words)**

**else:**

**print(Fore.GREEN+"are you redy for it...")**

**print(Fore.GREEN+"enter 1 to get your question:")**

**l=input()**

**if l=="1":**

**wordk=y.words()**

**k=random.choice(wordk)**

**else:**

**print(Fore.WHITE,Back.RED+"you pressed the wrong button ")**

**print(Fore.RESET,Back.RESET)**

**cls()**

**print(Fore.GREEN+"this is your question")**

**questions(k)**

**lis.ranklist(point)**

**stack\_value=point**

**stack.push(stack\_value)**

**#average point calculation**

**else:**

**count=0**

**if stack.length()>1:**

**while stack.length()!=1:**

**a=stack.pop()**

**b=stack.pop()**

**c=a+b**

**stack.push(c)**

**count=count+1**

**addition=stack.pop()**

**lis.ranklist(addition//count)**

**print(Fore.GREEN,Back.YELLOW+"the above is the average score for your total tries")**

**winsound.PlaySound("smb\_world\_clear.wav",winsound.SND\_ASYNC)**

**print(Fore.RESET,Back.RESET)**

**break**

**else:**

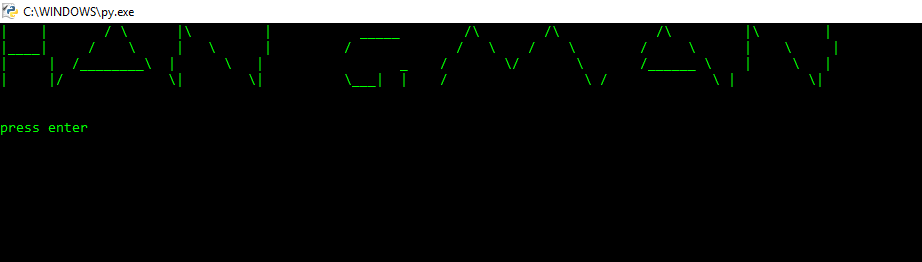
**print(Fore.GREEN+"please follow the instructions to follow the game properly")**

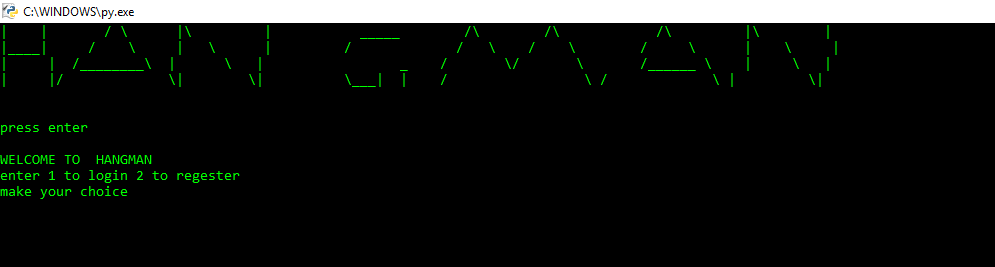
**winsound.PlaySound("smb\_warning.wav",winsound.SND\_ASYNC)**

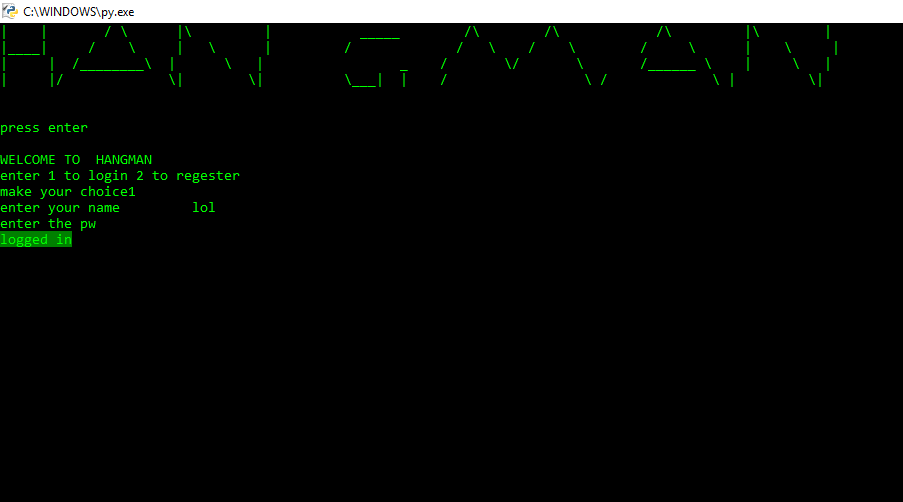
**dummy=input()**

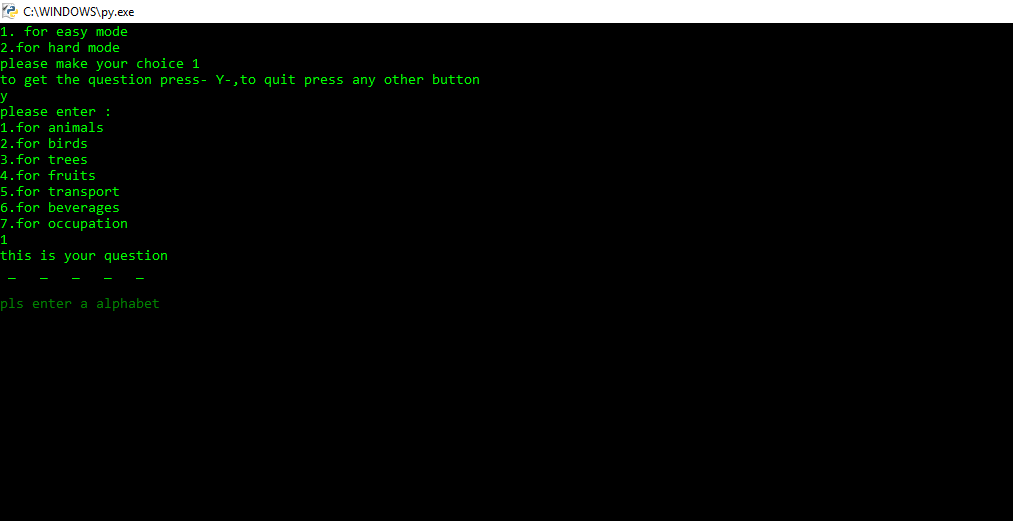
**CHAPTER 6**

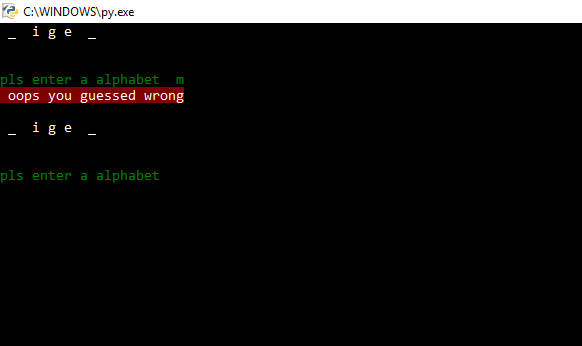
**SCREEN SHOTS**

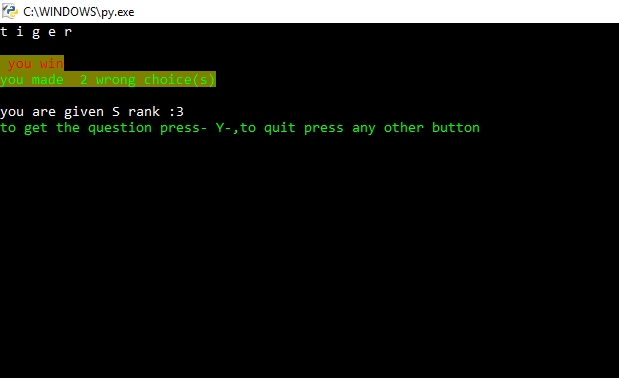


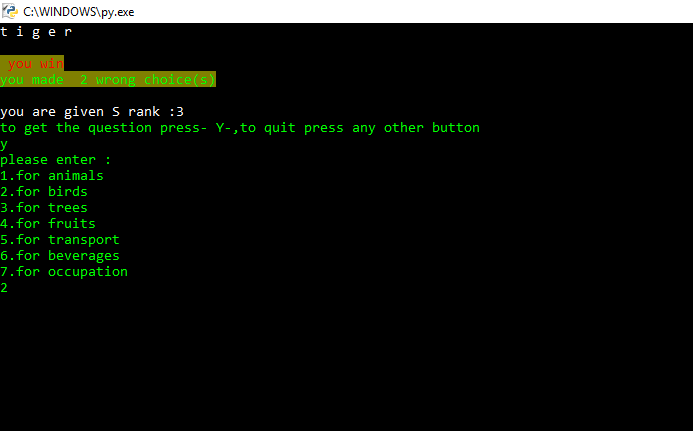


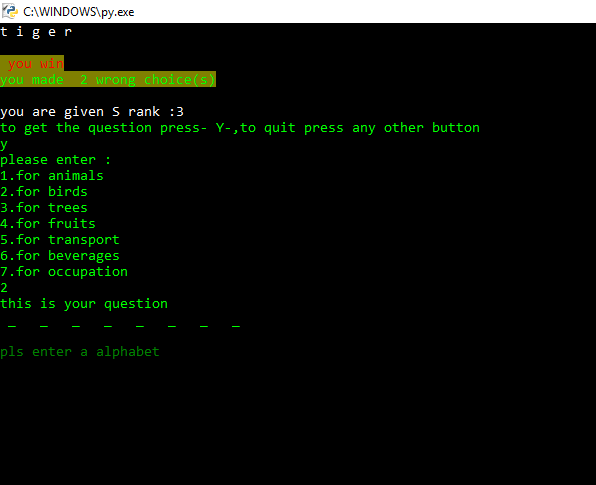


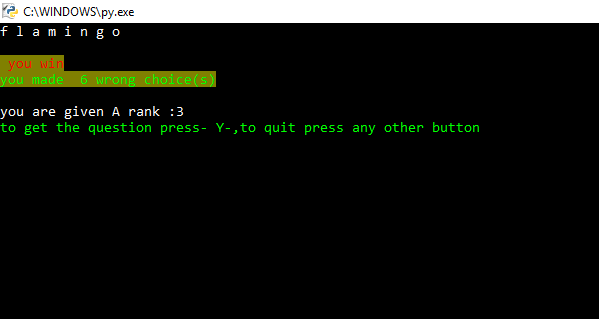


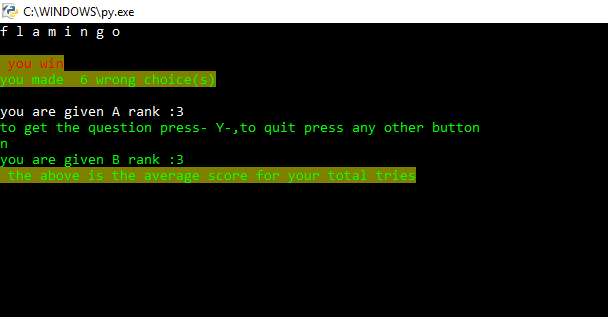












**CHAPTER 7**

**CONCLUSION AND FUTURE ENHANCEMENT**

**FUTURE ENHANCEMENT  
In games like such future enhancement can be made by making an online multiplayer mode with many words from the dictionary. This game could be enhanced in such a way that the user could go for different languages other than English which would bring a lot of people for around the world to play it and thus making it more popular.**

**In the online multiplayer mode a person could give a word to the computer(pass in as input) which would basically act as the question while the others would have to guess it as they would be present on a team chat and could ask for hints from the player who gave the question himself. Such actions would end up connecting the players of completely different society to play a game and spend their time.**

**CONCLUSION**

**Thus a hangman game is made in its very classic form by us, that helps the player to improvise on their attention and vocabulary skills and also acts as a time killing agent, which is required for the society currently since the most of its population is getting affected by the attention deficiency disorder in an alarming rate.**

**Thus acts by making games like such to improve the attention of the user would be very useful to the society to understand and create awareness regarding ADD and ADHD while the players enjoy such simple games.**

**REFERENCES**

1. Akyildiz, IA, Melodia, T &Chowdhury, KR 2007, ‘A Survey on Wireless Multimedia Sensor Networks’, Computer Networks (Elsevier) vol. 51, pp. 921-960.
2. Athina Kalampogia1 &PolychronisKoutsakis 2016, ‘H.264 and H.265 Video BandwidthPrediction’,1520-9210 (c)2016 IEEE DOI 10.1109/TMM.2017.2713642, IEEE Transactions on Multimedia.
3. Boards, P, Clare, G, Henry, F, Raulet, M &Vieron, J 2012, 'An Overview of the Emerging HEVC’, in SIVC 2012.
4. Bross, B, Helle, P, Qudin, S &Nguye, T 2012, ‘Quadtree Structures and Improved Techniques for Motion Representation and Entropy Coding in HEVC', IEEE International Conference on Consumer Electronics, pp. 26-30.