WORD WORLD

A

Mini Project Report

Submitted in partial fulfilment of the

Requirements for the award of the Degree of

BACHELOR OF ENGINEERING

IN

INFORMATION TECHNOLOGY

By

KABIR ANIRUDH 1602-20-737-124

SWAMI MANMATEJA 1602-20-737-145

AMGOTH NITHIN 1602-20-737-148

Department of Information Technology

Vasavi College of Engineering (Autonomous)

ACCREDITED BY NAAC WITH 'A++' GRADE

(Affiliated to Osmania University and Approved by AICTE)

Ibrahimbagh, Hyderabad-31

2022

Vasavi College of Engineering (Autonomous)

ACCREDITED BY NAAC WITH 'A++' GRADE

(Affiliated to Osmania University)

Hyderabad-500 031

DECLARATION BY THE CANDIDATE

We, **KABIR ANIRUDH, SWAMI MANMATEJA**, and **AMGOTH NITHIN**, bearing hall ticket numbers, **1602-20-737-124, 1602-20-737-145** and **1602-20-737-148**, hereby declare that the project report entitled **WORD WORLD** is submitted in partial fulfilment of the requirement for the award of the degree of **Bachelor of Engineering** in **Information Technology.**

This is a record of bonafide work carried out by us and the results embodied in this project report have not been submitted to any other university or institute for the award of any other degree or diploma.

KABIR ANIRUDH

1602-20-737-124

SWAMI MANMATEJA

1602-20-737-145

AMGOTH NITHIN

1602-20-737-148

(Faculty In-Charge) (Head,Dept of IT)

ACKNOWLEDGEMENT

We extend our sincere thanks to Dr. S. V. Ramana, Principal, Vasavi College of Engineering for his encouragement.

We express our sincere gratitude to Dr. K. Ram Mohan Rao, Professor & Head, Department of Information Technology, Vasavi College of Engineering, for introducing the Mini-Project module in our curriculum, and also for his suggestions, motivation, and co-operation for the successful completion of our Mini Project.

We also want to thank and convey our gratitude towards our mini project coordinators C. SIREESHA and N. DAVID RAJU, for guiding us in understanding the process of project development & giving us timely suggestions at every phase. We would also like to sincerely thank the project reviewers for their valuable inputs and suggestions.

ABSTRACT

The main aim of the project is to write a python script which produces platform where the user can test his/her technical skills by playing this game. This can be played by anyone who has basic technical knowledge and this can also be their stress buster, this game actually consists of three levels (easy, medium, hard), user should select a level to play the game, depending on the user's choice the level of questions would be varied, as a part of this game a picture will be displayed and the user have to guess the required word which is asked in that image or the image would be related to that word and the user have to enter that word, user can also enter his answer in the form of voice through the voice option available in the window. If the entered word is correct then 10 points will be added to the player's score else no points would be awarded.

TABLE OF CONTENTS

S.No	Topic	Page No.
1.	Introduction	6
2.	Technology	7
3.	Design	8
4.	Implementation	18
5.	Testing	29
6	Results	33
6.	Additional Knowledge Acquired	38
7.	Conclusion and Future work	39
8.	References	40

Introduction

Human brain stores the information mainly in the form of images. It is a fact that if a human memorizes any particular information in the form of images/scenario it will be remembered for a longer time.

Word world is a kind of technical quiz game which is designed to implement the above fact. It is a simple game and can be played by anyone having basic technical knowledge. This game is for the one who wants to test their technical knowledge in an unique way and can learn by playing.

Firstly, the user needs to enter his name and select the level from the options provided, this game has three levels- Easy, medium and hard. Based on the user's choice of level that level questions will be displayed.

The player needs to enter the word that will be asked in that image or the image will be about that word, player can also give his response in the form of voice by clicking the voice option which is provided in the game window, then the user needs to click save and next button to go to next question and to evaluate his previous response.

If the user enters the correct answer then 10 points will be added to his score, else no points will be awarded and next question will be displayed. If the user wants to end the game in between he can end by clicking the exit button provided in the game window.

TECHNOLOGY

To implement any project successfully, there will be technological requirements which can either be software or hardware requirements.

Software Requirements

Since our project is based on the Python programming language, it is a bare necessity to have the knowledge over syntax of the language and a proper interpreter and text editor to run and write the programs.

We have used Libraries(modules) such as

- Tkinter – for GUI
- PIL – for displaying image
- speech_recognition – for recognizing speech
- pyttsx3 - for recognizing speech
- time – to hold part of execution (sleep)

The above specified libraries should be downloaded and linked to your interpreter.

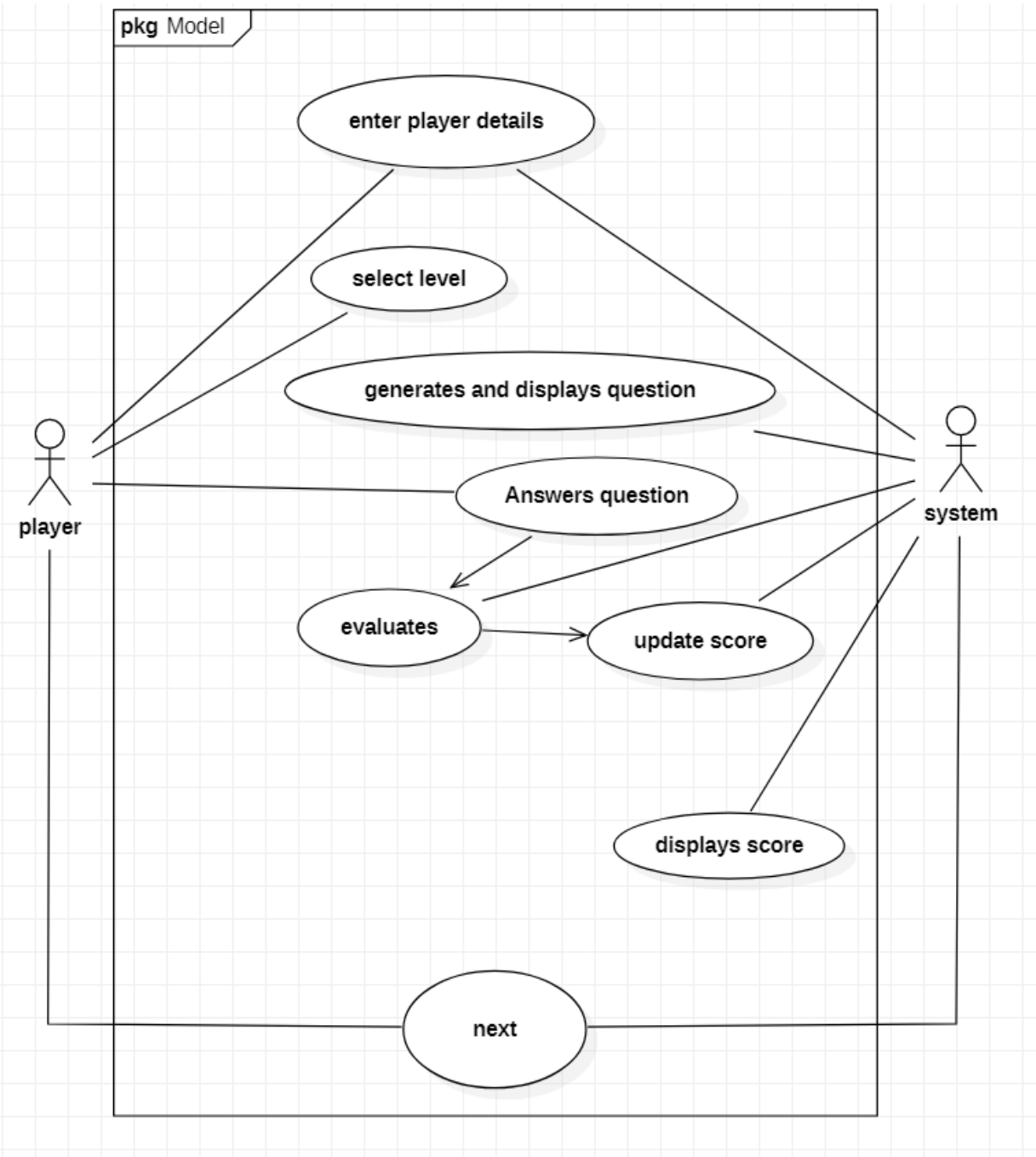
Hardware Requirements

The hardware requirements are quite low and there is no specific hardware required to run this program. We just need a processor with decent multi – tasking capabilities and minimum of 4GB RAM and it should have core i3 or i5.

PROPOSED WORK

DESIGN

i) Use case diagram



Use case descriptions

Use case ID : UC01

Name : select level

Actors : Player

Description : Allows the player to select a level

Pre-conditions : Player details

Post-conditions : A level must be selected

Main flow:

Player	System
	1. Displays a menu
2. Player selects a level	
	3. Validates the input and proceeds to next step

Use case ID : UC02

Name: generates question

Actors: System

Description: displays a question

Pre-conditions: selection of level

Main Flow:

Player	System
1. Clicks ‘start’ button	
	2. Displays the question

Use case ID : UC03

Name: Answer a question

Actors: Player

Description : The player answers the question

Pre-condition: Display of the question

Post-condition: response is given to the system

Main-Flow:

Player	System
	1. Prompts the user to enter the answer(word) for the given images
2. Player enters his answer	

Use case ID : UC04

Name: Evaluates

Actors: System

Description: System evaluates the answer given by player, if it is correct adds points to his score

Pre-conditions: User should give his response to the question

Post-conditions: updating score

Main-Flow:

Player	System
	1. Evaluates the response given by the user 1.1: adds points for correct answer

Use case ID : UC05

Name: displays score

Actors: system

Description : Displays the current score of the player

Pre-condition : clicking save and next button

Post-condition : displaying the updated score

Main-Flow:

Player	System
	1. Displays the current score

Use case ID : UC06

Name: Save and Next

Actors: Player, System

Description: prompts the player to go to next question

Pre-condition: Answering the previous question

Post-condition: system choosing the next question

Main Flow:

Player	System
1. Clicks the save and next button if player wants to go to the next question	
	2. displays the next question

Use case ID : UC07

Name: Exit

Actors: Player, System

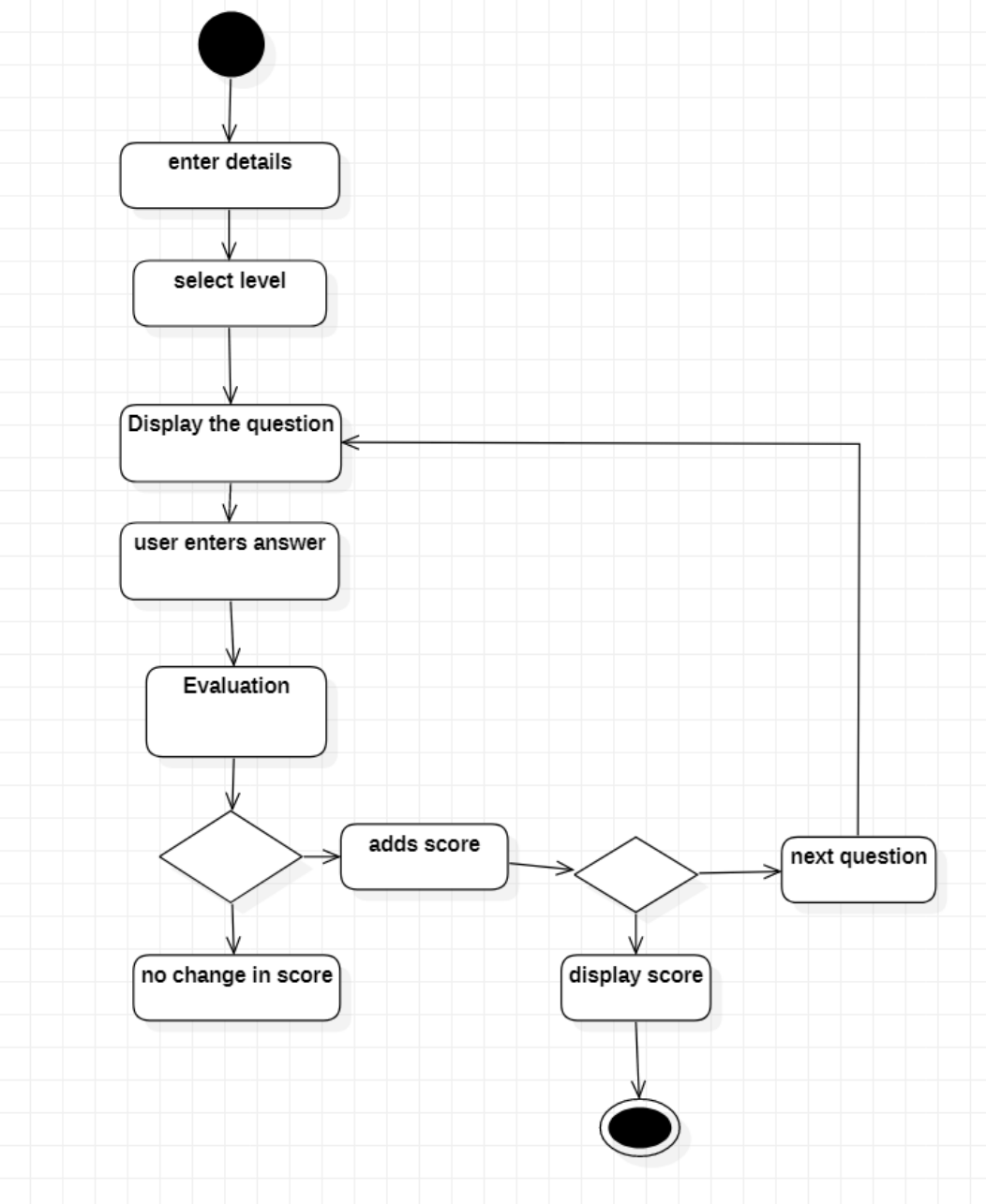
Description: prompts the player to exit the game

Post-condition: game will be ended and final window will be displayed

Main Flow:

Player	**System**
1. Clicks the exit button	
	2. Game is ended and final window is displayed

ii) Activity diagram



IMPLEMENTATION

Source code

```
#importing all required libraries

from tkinter import *

from tkinter import ttk

import tkinter as tk

import time

from PIL import ImageTk,Image

import speech_recognition as sr

import pyttsx3


#storing answers of questions of each level in lists

easy_ans = ['precedence','associativity','flowchart','binary tree','binary search
tree','compiler','conditional operator','variable']

med_ans =
['pointer','dictionary','structure','lambda','abstraction','indentation','new','byte']

hard_ans = ['method overloading','module','jdb','lambda','strlen','abstract method']

easy = [1]

medium = [1]

hard = [1]

itr = [2]

name = ['']

level_selected = []

#creating a tkinter welcome window

window = tk.Tk()
```

```
width= window.winfo_screenwidth()
height= window.winfo_screenheight()
window.geometry("%dx%d" % (width, height))
window.title("Welcome")
title = Label(window, text = "Enter your Details and select the level",font =("Algerian", 15))
title.place(x= 400,y=160)
fname = Label(window, text = "First Name",font =("Times New Roman", 13))
fname.place(x = 400, y= 220)
lname = tk.Label(text = "Last Name",font = ("Times New Roman",13))
lname.place(x=400, y= 260)
fentry = tk.Entry()
lentry = tk.Entry()
fentry.place(x= 650, y=220)
lentry.place(x= 650, y=260)

#function for retrieving name entered in textbox
def getname():
    name[0] = fentry.get() +' '+ lentry.get()
    print(name[0])
getname()

#creating combobox to display a menu of levels
level=ttk.Combobox(window, width = 20,font=("Times New Roman",15))
level.insert(END,'Select level')
level['values']=('Easy','Medium','Hard')
```

```
level.place(x=500,y=300)


#function to get the level selected by the user
def getlevel(event):
    l = level.get()
    level_selected.append(l)
    print(l)
level.bind("<<ComboboxSelected>>",getlevel)



#list to store the score of the user
score = [0]


#function for new window(game window)
def level_implementation():
    window.destroy()
    root = tk.Tk()
    global ren
    root.title("Game")
    width = root.winfo_screenwidth()
    height = root.winfo_screenheight()
    root.geometry("%dx%d"%(width,height))
    root.configure(bg = 'LightSkyBlue2')
    msg = Label(root, text="Word world", font=("Algerian", 60), bg='LightSkyBlue2')
    msg.place(x=390, y=10)
    w = Canvas(root, width=1325, height=30, bg='LightSkyBlue2', highlightthickness=0)
```

```
    w.create_line(15, 25, 10000, 25, width=2)
    w.place(x=0, y=120)
    path = R"C:\Users\HP\Downloads\logo.png"
    load = Image.open(path)
    rload = load.resize((200, 125))
    ren = ImageTk.PhotoImage(rload)
    img = Label(root, image=ren)
    img.place(x=0, y=0)
    ans =  Label(root,text = "Enter answer",font = ("Times New Roman",13),bg = 'LightSkyBlue2')
    ans.place(x=400, y= 550)
    answer = tk.Entry()
    answer.place(x=500,y=550)



    #Function to get the current score of the player
    def get_score():
        sc =  Label(root,text = "Current score:"+str(score[0]),font = ("Times New Roman",13),bg = 'LightSkyBlue2')
        sc.place(x = 500, y= 600)
        root.after(3000,sc.destroy)



    #Function to display the final window
    def final_root():
        final = Toplevel(root)
```

```
        final.title("END")
        width = final.winfo_screenwidth()
        height = final.winfo_screenheight()
        final.geometry("%dx%d"%(width,height))
        my_canvas = Canvas(final, bg="MistyRose2")
        my_canvas.pack(fill="both", expand=True)
        my_canvas.create_line(200,200,250,250, fill="navy", width=2)
        my_canvas.create_line(1050, 200, 1000,250, fill="navy", width=2)
        my_canvas.create_line(200, 550, 250, 500, fill="navy", width=2)
        my_canvas.create_line(1050, 550, 1000, 500, fill="navy", width=2)
        my_canvas.create_rectangle(200,200,1050,550, outline="navy", width=2)
        my_canvas.create_rectangle(250,250,1000,500, outline="navy", width=2)
        my_canvas.create_text(width/2,height/2, text="Thanks for playing
"+name[0]+"\nYour final score is "+str(score[0]), font=("Helvetica", 30), fill="black")


    #Function for saving the player response and going to the next question
    def Sandn():
        if(easy[0] == itr[0]):
            itr[0]+=1
        while(easy[0]<itr[0]):
            if(itr[0] == 2 ):
                itr[0]+=1
            easy[0]+=1
            print("easy:"+str(easy[0]))
            #time.sleep(2.4)
```

```
            #get_val()
            print("itr :"+str(itr[0]))
            if(easy[0]<10):
                display_easy() #calling display easy function for displaying the next image
            else:
                final_root()



    #Function for displaying an image of level easy
    def display_easy():
        global render
        print(easy[0])



        #Function for evaluating user response
        def get_val():
            ans = answer.get()
            print(ans.lower())
            if(ans.lower() == easy_ans[easy[0]-3]):
                correct = Label(root,text = "Correct Answer",font = ("Times new
Roman",13),bg = 'LightSkyBlue2')
                correct.place(x = 400, y= 600)
                root.after(2000,correct.destroy) #after 2 seconds it will remove the label
                score[0]+=10
```

```
            answer.delete(0,END)


        else:
            wrong = Label(root,text = "Wrong Answer",font = ("Times new
Roman",13),bg = 'LightSkyBlue2')
            wrong.place(x = 400, y= 600)
            root.after(2000,wrong.destroy)
            answer.delete(0,END)


        score1 = Label(root,text = "Score:"+str(score[0]),font = ("Times new
Roman",13),bg = 'LightSkyBlue2')
        score1.place(x=530, y = 600)
        root.after(2000,score1.destroy)
        print(score[0])


    path = R"C:\Users\HP\OneDrive\Desktop\easy_im\ "
    path = path[:-1]
    path = path +str(easy[0])+".jpeg"
    load= Image.open(path)
    rload=load.resize((300,250))
    render = ImageTk.PhotoImage(rload)
    img = Label(root, image = render)
    img.place(x=400, y=250)
    #time.sleep(2.4)
    n = Button(root,text  = "Save and Next",command =lambda:[Sandn(),get_val()],bg
= "orange" ) #calling Sandn function using the button
    n.place(x=800, y = 600)
```

```
    b4 = Button(root,text  = "Exit", command = final_root,bg = "yellow" )
    b4.place(x = 100, y = 600)



#save and next function for medium level
def Sandn2():
        if(medium[0] == itr[0]):
            itr[0]+=1
        while(medium[0]<itr[0]):
            if(itr[0] == 2 ):
                itr[0]+=1
            medium[0]+=1
            print("med:"+str(medium[0]))
            #time.sleep(2.4)
            #get_val()
            print("itr :"+str(itr[0]))
            if(medium[0]<10):
                display_med() #calling display easy function for displaying the next image
            else:
                final_root()


#Function for displaying image of level medium
def display_med():
    global render
    print(medium[0])
```

```
    ans =  Label(root,text = "Enter answer",font = ("Times New Roman",13),bg =
'LightSkyBlue2')



    #Function for evaluating user response
    def get_val2():
        ans = answer.get()
        print(ans.lower())
        if(ans.lower() == med_ans[medium[0]-3]):
            correct = Label(root,text = "Correct Answer",font = ("Times new
Roman",13),bg = 'LightSkyBlue2')
            correct.place(x = 400, y= 600)
            root.after(2000,correct.destroy) #after 2 seconds it will remove the label
            score[0]+=10
            answer.delete(0,END)
        else:
            print("Your answer is "+ans.lower()+" Actual answer is
"+med_ans[medium[0]-3])
            wrong = Label(root,text = "Wrong Answer",font = ("Times new
Roman",13),bg = 'LightSkyBlue2')
            wrong.place(x = 400, y= 600)
            root.after(2000,wrong.destroy)
            answer.delete(0,END)
        score1 = Label(root,text = "Score:"+str(score[0]),font = ("Times new
Roman",13),bg = 'LightSkyBlue2')
        score1.place(x=530, y = 600)
        root.after(2000,score1.destroy)
```

```
        print(score[0])


        path = R"C:\Users\HP\OneDrive\Desktop\med_im\ "
        path = path[:-1]
        path = path +str(medium[0])+".jpeg"
        load= Image.open(path)
        rload=load.resize((350,250))
        render = ImageTk.PhotoImage(rload)
        img = Label(root, image = render)
        img.place(x=400, y=250)
        #time.sleep(2.4)
        n = Button(root,text  = "Save and Next",command
=lambda:[Sandn2(),get_val2()],bg = "orange" ) #calling Sandn function using the button
        n.place(x=800, y = 600)
        b4 = Button(root,text  = "Exit", command = final_root,bg = "yellow" )
        b4.place(x = 100, y = 600)



    #save and next function for level hard
    def Sandn3():
        if(hard[0] == itr[0]):
            itr[0]+=1
        while(hard[0]<itr[0]):
            if(itr[0] == 2 ):
                itr[0]+=1
            hard[0]+=1
```

```
            print("hard:"+str(hard[0]))
            #time.sleep(2.4)
            #get_val()
            print("itr :"+str(itr[0]))
            if(hard[0]<8):
                display_hard() #calling display easy function for displaying the next image
            else:
                final_root()



    #function for displaying image of level hard
    def display_hard():
        global render
        print(hard[0])



        #Function for evaluating user response
        def get_val3():
            ans = answer.get()
            print(ans.lower())
            if(ans.lower() == hard_ans[hard[0]-3]):
                correct = Label(root,text = "Correct Answer",font = ("Times new
Roman",13),bg = 'LightSkyBlue2')
                correct.place(x = 400, y= 600)
                root.after(2000,correct.destroy) #after 2 seconds it will remove the label
```

```
            score[0]+=10
            answer.delete(0,END)

        else:
            wrong = Label(root,text = "Wrong Answer",font = ("Times new Roman",13),bg = 'LightSkyBlue2')
            wrong.place(x = 400, y= 600)
            root.after(2000,wrong.destroy)
            answer.delete(0,END)
        score1 = Label(root,text = "Score:"+str(score[0]),font = ("Times new Roman",13),bg = 'LightSkyBlue2')
        score1.place(x=530, y = 600)
        root.after(2000,score1.destroy)
        print(score[0])

    path = R"C:\Users\HP\OneDrive\Desktop\hard_im\ "
    path = path[:-1]
    path = path +str(hard[0])+".jpeg"
    load= Image.open(path)
    rload=load.resize((350,250))
    render = ImageTk.PhotoImage(rload)
    img = Label(root, image = render)
    img.place(x=400, y=250)
    #time.sleep(2.4)
    n = Button(root,text  = "Save and Next",command =lambda:[Sandn3(),get_val3()],bg = "orange" ) #calling Sandn function using the button
    n.place(x=800, y = 600)
```

```
    b4 = Button(root,text  = "Exit", command = final_root,bg = "yellow" )
    b4.place(x = 100, y = 600)
if(level_selected[-1] == 'Easy'):
    display_easy()
elif(level_selected[-1]=='Medium'):
    display_med()
elif(level_selected[-1]=='Hard'):
    display_hard()
voiceans = ['']

#function for inserting the user's answer recieved through voice into textbox
def a():
        answer.insert(0,voiceans[0])



#Function for recognizing the user's voice i.e. for converting voice to text
def voice_recognizer():
# Initialize the recognizer
        r = sr.Recognizer()

    # Loop infinitely for user to speak
        while(1):
            try:
                # use the microphone as source for input.
                with sr.Microphone() as source2:
```

```
                # wait for a second to let the recognizer adjust the energy threshold based
on the surrounding noise level
                r.adjust_for_ambient_noise(source2, duration=0.2)

                #listens for the user's input
                audio2 = r.listen(source2)

                # Using google to recognize audio
                MyText = r.recognize_google(audio2)
                voiceans[0] = str(MyText)
                print(voiceans[0])
                a()
                return MyText



            #exceptional handling
            except sr.RequestError as e:
                print("Could not request results; {0}".format(e))

            except sr.UnknownValueError:
                print("unknown error occured")

    #button for calling voice_recognizer function
    b5 = Button(root,text = "Voice",command = voice_recognizer)
    b5.place(x = 200,y = 600)
```

```
#button for calling level_implementation function
b3 = tk.Button(window,text  = "Start", command = level_implementation,bg ="yellow")
b3.place(x = 500, y = 340)


window.mainloop()
root.mainloop()
```

GITHUB LINK

https://github.com/K-Anirudh/miniproject1

Testing

Test case ID: TC01	Use case ID: UC01	
Test case Title: level selection		
Test case description: user selects a level		
Test steps	Expected result	Actual result
Selection of a level from the levels list provided	Selected level should be displayed in the combobox	Selected level is displayed in the combobox

Test case ID: TC02	Use case ID: UC02	
Test case Title: question generation		
Test case description: a question based on the selected level will be displayed		
Test steps	Expected result	Actual result
Player should click start button after selection of level	Question should be displayed based on player's selected level	Question is successfully displayed based on player's selected level

Test case ID: TC03	Use case ID: UC03	
Test case Title: answering the question		
Test case description: a text box will be displayed where the player needs to enter his answer		
Test steps	Expected result	Actual result
Player should enter his answer in the textbox provided	Should display the textbox where the user can enter his answer	Textbox is displayed successfully

Test case ID: TC04	Use case ID: UC04	
Test case Title: Evaluation of the user's response		
Test case description: points should be added if user enters the correct answer else no modification in score		
Test steps	Expected result	Actual result
Player should click save and next button	Should display whether it the response is correct or not	It is displayed whether user's response is correct or not

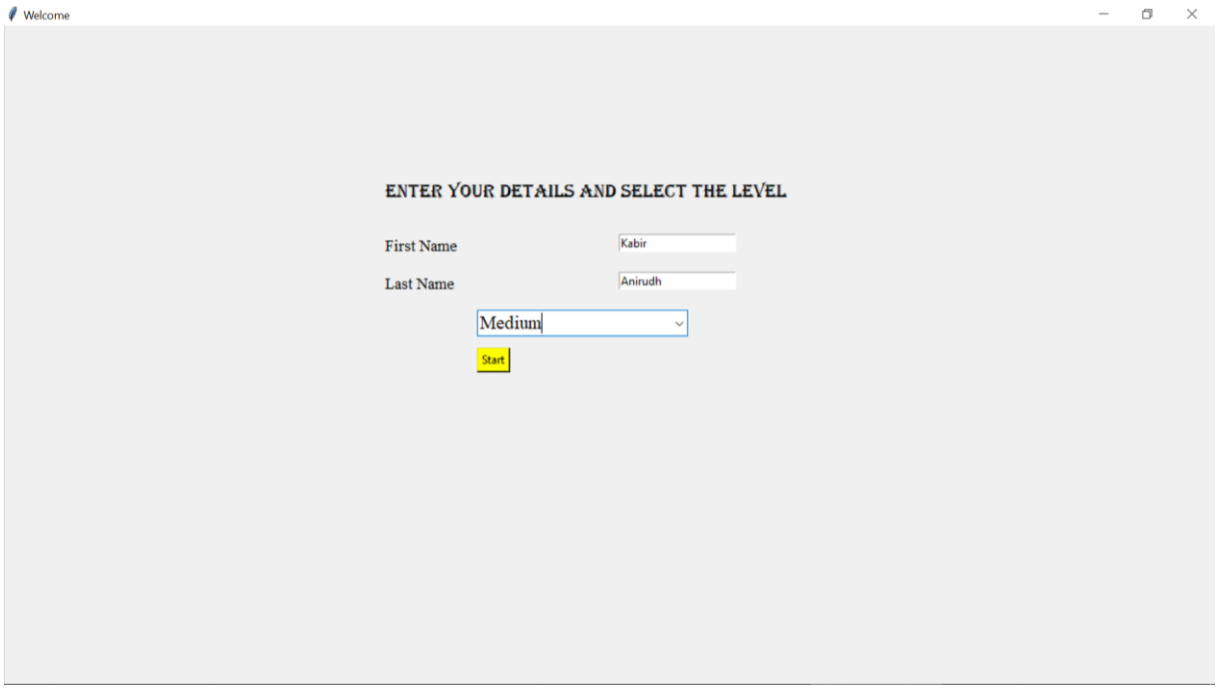
Test case ID: TC05	Use case ID: UC05	
Test case Title: Score display		
Test case description: The modified score should be displayed on the screen		
Test steps	Expected result	Actual result
Player should click save and next button	Should display the current score after modification	Modified score is displayed successfully

Test case ID: TC06	Use case ID: UC06	
Test case Title: next question display		
Test case description: next question displayed on the screen upon player request		
Test steps	Expected result	Actual result
Player should click save and next button	Should display the next question	Next question is displayed successfully

Test case ID: TC07	Use case ID: UC07	
Test case Title: ending the game		
Test case description: Game is ended after choosing this option		
Test steps	Expected result	Actual result
Player should click exit button	Final window should be displayed which contains player's final score	Final window is displayed successfully

Results

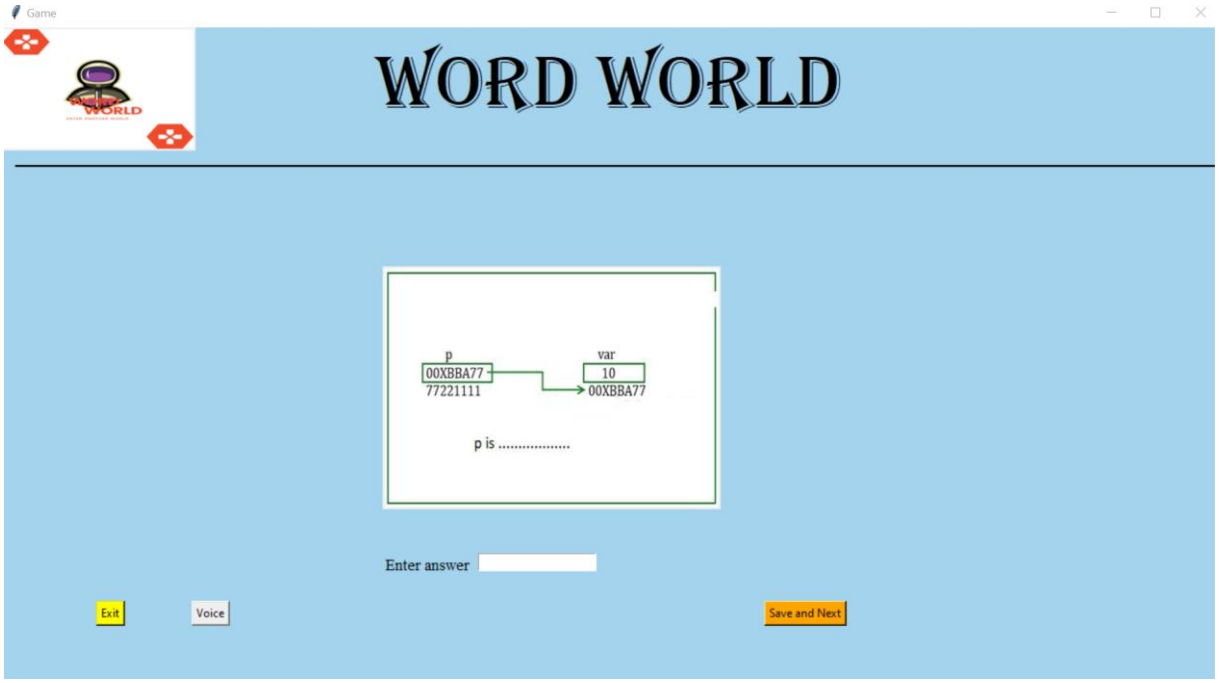
TC01



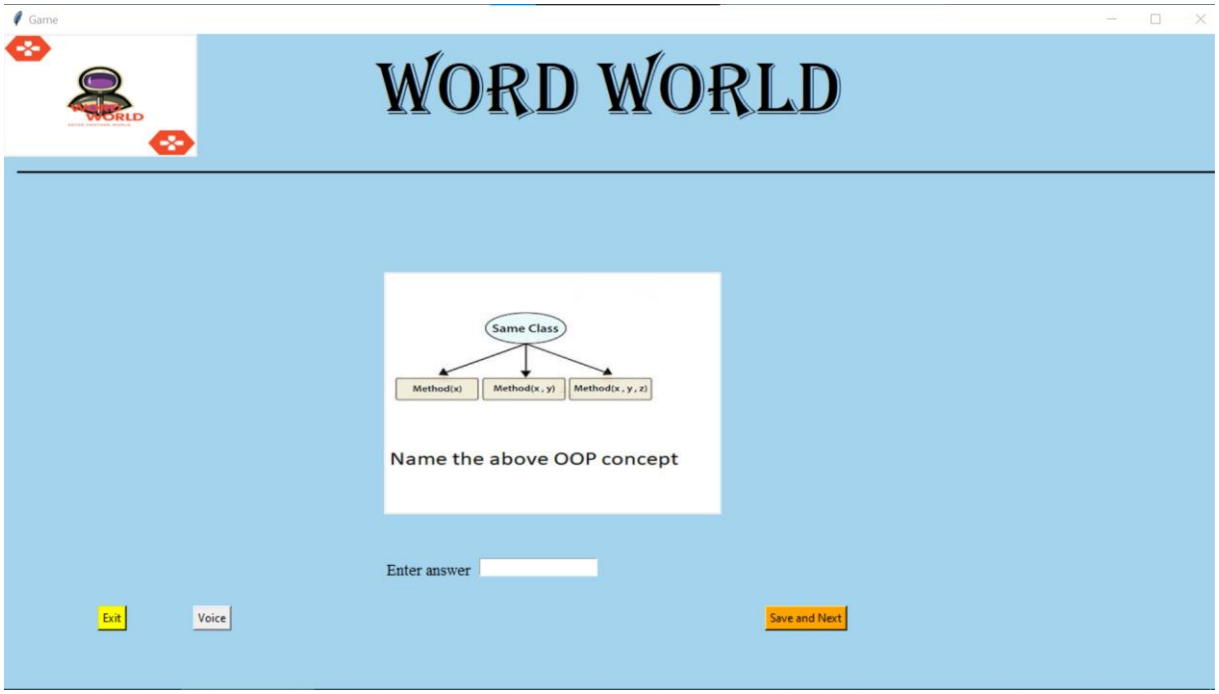
TC02

Level: Easy

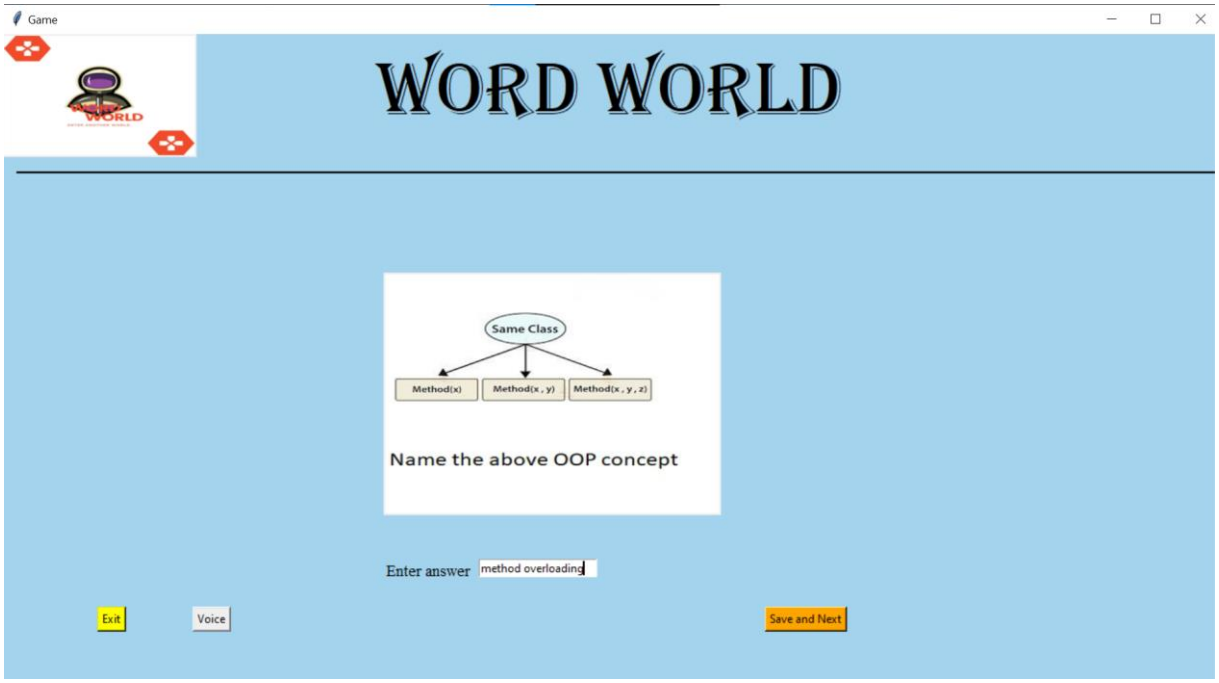
Level: Medium

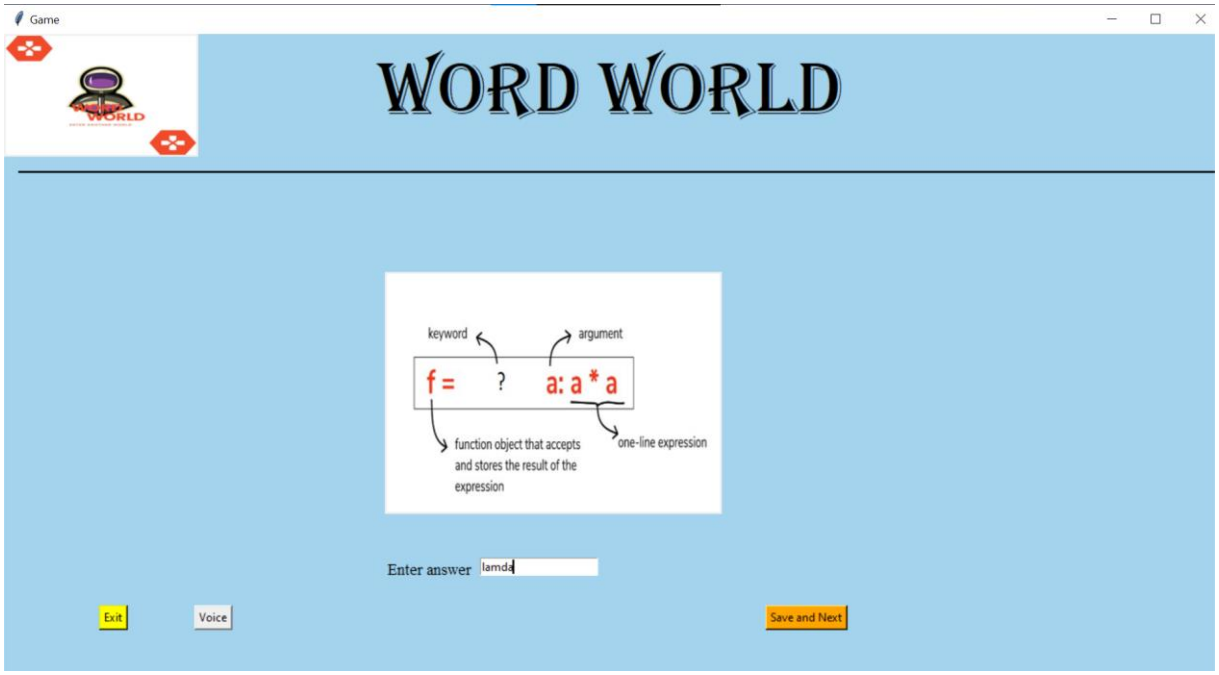


Level: Hard

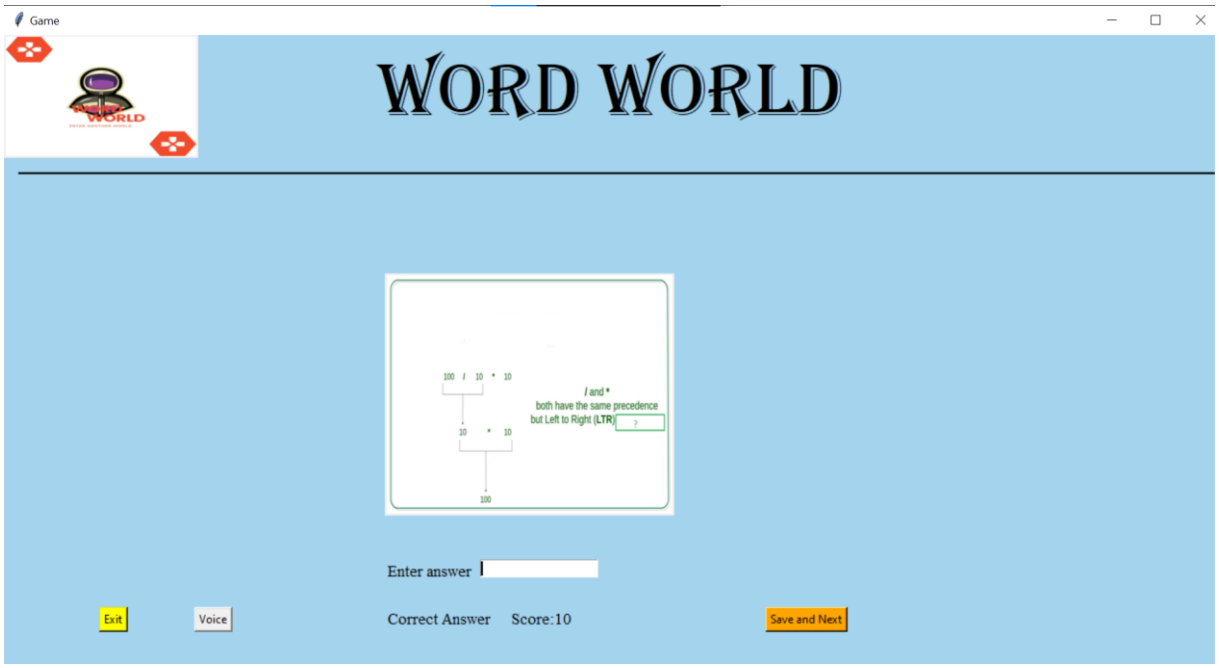


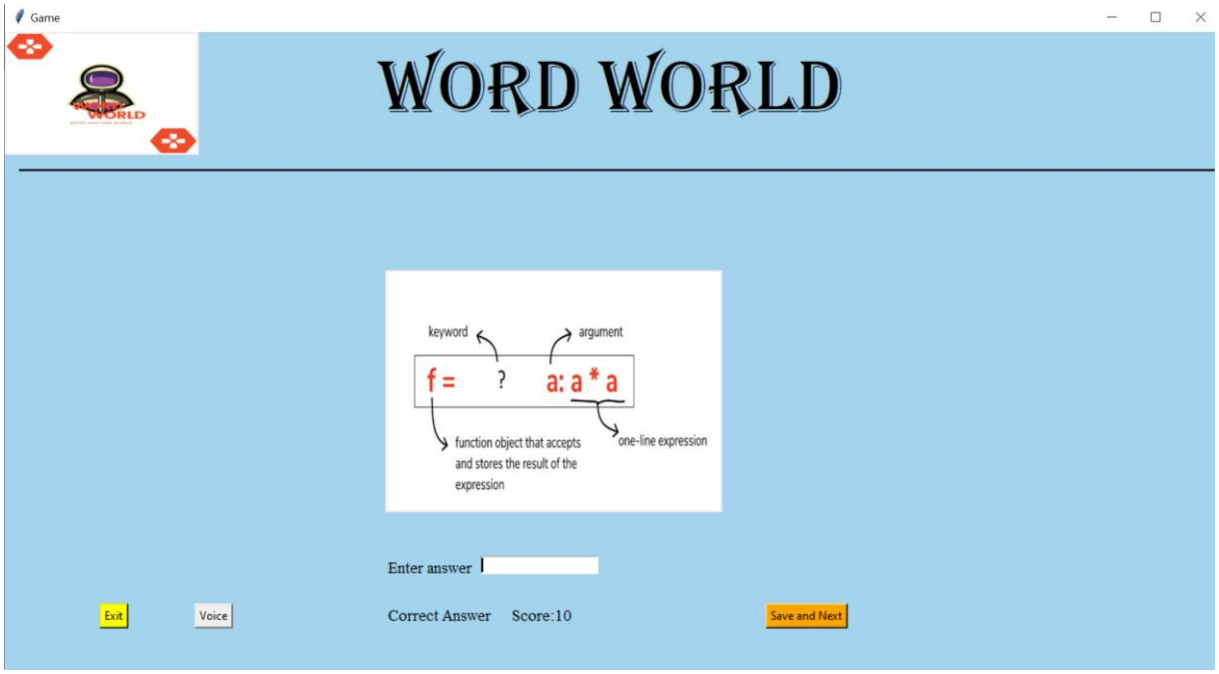
TC03


Game
WORD WORLD
Same Class
Method(x)
Method(x , y)
Method(x , y , z)
Name the above OOP concept
Enter answer
method overloading
Exit
Voice
Save and Next

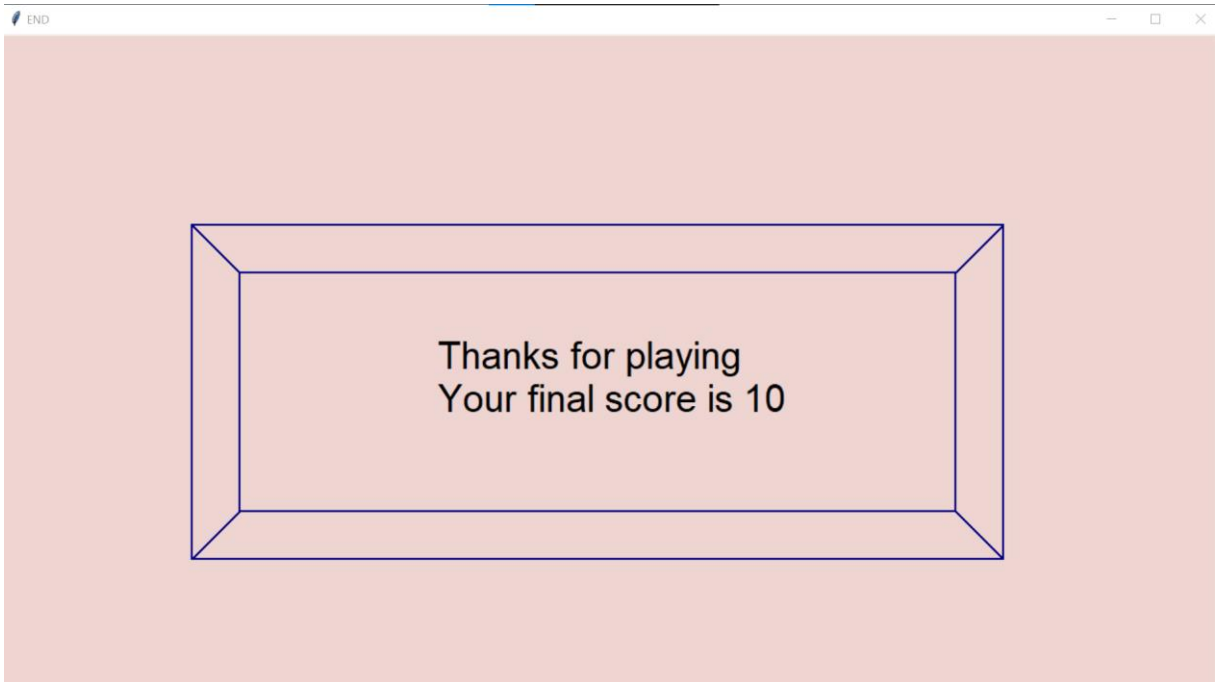

Game
WORD WORLD
keyword
argument
f =
?
a: a * a
function object that accepts and stores the result of the expression
one-line expression
Enter answer
lamda
Exit
Voice
Save and Next

TC04, TC05 and TC06





TC07



ADDITIONAL KNOWLEDGE ACQUIRED

- Python GUI programming using tkinter module.
- Recognising speech (converting speech to text) using pyttsx3 module.
- Image displaying using Pillow module.
- Describing projects with diagrams i.e. using use case and activity diagrams

CONCLUSION

Word world is a game which is designed to test one's skills, this is designed in a way that anyone can use it easily. This game emphasizes on the fact that humans can remember things for longer time when they try to memorize the things in the form of images. This game also emphasizes the saying learning by playing. It took lot of efforts from us to put this project into prominence and learnt many new things during this project.

FUTURE WORK

- This idea can be further improved by implementing it with an application for various OS and it will be helpful to learn something in an innovative and effective way
- This can be extended for online multiplayer by using Sockets/Networking and Pygame
- Can be implemented as spell checker

References

Basic Python

1. Course covered during 2nd semester by Dr. Srinivas Chakravarthy
2. PPTs and Handouts provided by the sir.
3. Python Programming-Using Problem solving approach first edition book by Reema Thareja

Tkinter tutorials

1. https://www.geeksforgeeks.org/python-tkinter-tutorial/
2. https://www.geeksforgeeks.org/python-gui-tkinter/
3. https://youtu.be/VMP1oQOxfM0
4. https://docs.python.org/3/library/tkinter.html

Use cases and Activity diagram

1. Handouts provided by C. Sireesha.
2. https://youtu.be/knM8BGY9yVI