

COMPUTER SCIENCE PROJECT REPORT

CODE DUNGEONS

Madhav Krishnan
1747715

Class: XII B



SINGAPORE • MALAYSIA • THAILAND • JAPAN • UAE • INDIA • VIETNAM

Certificate

Department of Information & Technology

**Global Indian International School,
Smart Campus, Singapore**

This is to certify that the project entitled “CODE DUNGEONS” in python is a genuine work of Madhav Krishnan

Roll No. 1747715 undertaken as a part of fulfilment of Computer Science [083] practical syllabus for **A.I.S.S.C.E. 2020-2021** to be conducted by C.B.S.E. and has been completed within stipulated time period under my guidance and supervision.

Ms. Radha Ganesh

Academic Coordinator
Computer Science Teacher
Date:

ACKNOWLEDGEMENT

I take this opportunity to express my acknowledgement and sincere gratitude to Ms Radha Ganesh for her valuable suggestions and able guidance required for this project. It is through her I have learnt efficient debugging skills which were helpful in completing the project on time.

Madhav Krishnan

CONTENTS

1. Introduction	6
2. About the programming tool used.....	7
3. Problem Definition.....	8
4. Design requirements.....	9
5. System requirements.....	13
6. Code listing.....	14
7.Flowchart/Algorithm.....	24
8. User documentation.....	25
9. Bibliography.....	34

PROJECT DOCUMENTATION

1.

INTRODUCTION

In the near future knowing how to code will be as necessary as knowing how to write is today. With 70% of coding jobs being outside the field of technology it proves essential to at least know the basics.

The problem with programming is, unless you're interested in it, it appears to be a dull daunting task, full of dry logic that only hardcore programmers care to understand. Many softwares have been made to make it simple and fun to understand and this project is attempt to make it a bit more exciting and thought provoking along with being a tool for teachers for adding more questions or challenges within the story to enhance learning in a more enjoyable way

This python program allows for the following

- 1) Quick Learning of Basic Python Modules
- 2) Solving Boolean Expressions
- 3) Fun and Interactive
- 4) Statistics shown in tabular form
- 5) Testing knowledge on python code by a text-based adventure game which allows user to type code to be evaluated.
- 6) It is modular so that the questions(along with the game) can be increased.

2. ABOUT THE PROGRAMMING TOOL USED

The tool we use for this program is **PYTHON.**

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic

typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasises readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective.

3.

PROBLEM DEFINITION

In the current world, many people struggle to understand some of the very few basic concepts of python, which makes them make minor mistakes while writing or typing out long codes resulting in many unexpected errors.

Many people don't like learning concepts through books which is very boring for many of us. Books are also an inefficient way of learning. So to counter this problem our team has decided to create a user friendly game.

"CODE DUNGEONS" is a really fun and interesting text based game. It consists of various levels of monsters at each dungeon or level, where the player has to fight these monsters by answering a set of python questions(about 3-5 questions) depending on the amount of questions answered then similarly the damage would be given to the monsters. This project is an attempt to make it a bit more exciting and thought provoking along with being a tool for teachers for adding more questions or challenges within the story to enhance learning in a more enjoyable way.

By this game, our team feels that it'll be a really fun way to learn the basic concepts of python, so that people don't make silly mistakes while typing out long codes. An individual will be drawn to the game due to its simple rag mechanics and will be able to experience an interactive learning session which books are unable to provide to the current generation. This game should further draw students into software development in Python and should open up their minds to what can be made using Python if given time.

4.

DESIGN REQUIREMENTS

A Text Based Game works with **TEXT FILES** which contains the python questions. Using these python questions the player can progress to the further levels/dungeons in the game.

Mysql.connector connects the database which has the player stats like HEALTH,ATTACK etc. It saves these datas as the player progresses through different levels of monsters.

The design for the game requires the following modules and various inputs from the user along with the latest version of Python and MySQL which include:-

- 1)Tkinter
- 2)Pickle module
- 3)Random module
- 4)CSV
- 5)mysql.connector
- 6)Time module

4.1 Data Structures:

We have included data structures which include both non primitive and primitive sections. Further subdivisions include Lists, Text files, and mySQL

4.2 Files:

Text files are used to save data which save players progress health and attack. These files are later read to load the game or overwritten to save the game. Text files are also used to load statistics in a MySQL database.

4.3 Functions:

`def question():`

Used to determine a random question for the fight sequence. Along with answers in accordance with their respective questions to test whether a given attack will work or not.

`def main():`

It is basically the main storyline. It checks user progress and assigns them battle and scenes to go through before proceeding to the boss. It behaves like a game engine and is the core function

`def print_slow():`

Uses commands from the time module to take strings as an input and display them character by character to appear as if they have been typed in real time. It is also used for various coloured text in the shell

`def menu():`

It displays the main menu of the game including the

- 1.New Game
- 2.Load Game
- 3.Quit

It also displays the main logo of the game.

`def stats(name):`

It opens the text file created when the player enters the game and it displays the attack and health of the player.

```
def newgame(name=""):
```

This function opens the text file and updates the datas like

1.Attack

2.Health

As the player/individual starts the game. We use the file.write() operation to do this.

```
def load():
```

It inputs the username from the player and accordingly it creates a text file.

Eg:-If the username is Surge then it creates a Surge.txt text file. Then it reads the created file and assigns the values for

1.Attack 2.Health 3.Progress

We use a if-else loop to check if the file created exists.

In the end we call the menu() function.

```
def qs_prac():
```

All the questions that the user answered incorrectly appear in a file for later practice

```
def gameover():
```

```
    st("""You forever rome the dungeons for eternity...
-----Gameover-----\n""", "ERROR")
it displays the GAMEOVER TEXT in the end.
```

```
def text():
```

We use this to display the storyline for the dungeons which makes the game more interesting.

```
def win():
```

After the player defeats the final boss, this function will display a final text “**Thank you for playing**” where the game ends.

```
def battle(l=[“Monster_name”,[0,0]]):
```

This function displays the image of the monster and also provides the user with a set of questions from a pre made text file. As the player answers the questions they can accordingly attack the monster or heal themselves. After the player defeats the monster at that respective level the player continues the game.

5.SYSTEM REQUIREMENTS

The Python game will require the following:

1. Computer

Minimum Requirements:

Macbook Air

OS X 10.9 or later.

4GB of memory.

12.5GB of available storage

2. Printer.

3. Basic operations in OS.

6. Code Listing

```
import sys,time,random,pickle,csv,os,shutil
shell=sys.stdout.shell
from tkinter import *

#Mysql connector
import mysql.connector
mydb=mysql.connector.connect(
    host="localhost",
    user="root",
    password="12345678",
    database="menagerie")
mycursor=mydb.cursor()
mycursor.execute("Use menagerie")

#Initialize
name=""
progress=0
health=100
attack=5
heal=5
img=None
label=None
root=None
D=8
C=5+D

#monsters Data
DL1=["D1","D2","D3","D4",["Guardian",[20,4]],["Ghoul",[5,10]],
      ["Wilter",[15,8]]]
DL2=["C1","C2","C3",["Alien",[20,7]],["Banshee",[23,8]],
      ["Warden",[3,13]]]

#Initialize Questions.txt file
f=open("Assets/Questions/Questions.txt","r+")
```

```

Q=f.readlines()
r=[a for a in range(0,len(Q),2)]
f.close()

#Core functions
def menu():
    global root,img,label
    if root==None:
        root=Tk()
    if label!=None:
        label.destroy()
    img = PhotoImage(file="Assets/Titlescreen.png")
    label = Label(root,image=img)
    label.pack()
    root.update()
    print("""-----Code Dungeons-----"
->1. New Game
->2. Load Game
->3. Quit""")
    option=input("Enter your choice : ")
    if option=="1":
        main(0)
    elif option=="2":
        load()
    elif option=="3":
        sys.exit()
    else:
        menu()

def main(p=0):
    global name
    global progress,attack,health,heal
    global root,img,label
    while progress<13:
        if progress==0:
            if root==None:

```

```

root=Tk()
if label!=None:
    label.destroy()
img = PhotoImage(file="Assets/Backgrounds/D0.png")
label = Label(root,image=img)
label.pack()
root.update()
name=input("Enter your name adventurer: ")
newgame(name)
st("When the news about random people disappearing in
the ancient dungeons of the Kingdom reaches him, the king sends
you on a mission to those dungeons to figure out the secrets
entrapped within the walls","ERROR")
st("When you approach the dungeon... you hear a faraway
hiss","ERROR")
st("Your stats")
stats(name)
elif progress<7:
    global DL1
    if root==None:
        root=Tk()
    if label!=None:
        label.destroy()
    r=random.choice(DL1)
    DL1.remove(r)
    if r in ["D1","D2","D3","D4"]:
        img = PhotoImage(file="Assets/Backgrounds/
{}{}.png".format(r))
        label = Label(root,image=img)
        label.pack()
        root.update()
        text()
    else:
        battle(r)
elif progress<13:
    global DL2

```

```

if root==None:
    root=Tk()
if label!=None:
    label.destroy()
r=random.choice(DL2)
DL2.remove(r)
if r in ["C1","C2","C3"]:
    img = PhotoImage(file="Assets/Backgrounds/
{} .png".format(r))
    label = Label(root,image=img)
    label.pack()
    root.update()
    text()
else:
    battle(r)
progress+=1
health+=5
heal+=4
attack+=4
f=open("Player data/{}.txt".format(name),"w")
f.write(str(attack)+"\n"+str(health)+"\n"+str(progress))
f.flush()
f.close()
else:
    battle(["Python",[50,8]])

def newgame(name=""):
f=open("Player data/{}.txt".format(name),"w")
l=[5,100,progress] # Starting Attack Health and progress
for a in l:
    f.write(str(a))
    f.write("\n")
f.flush()
f.close()

def load():

```

```

global progress,name,health,attack
name=input("Enter username of save file ")
if os.path.isfile("Player data/{}.txt".format(name)):
    f=open("Player data/{}.txt".format(name))
    data=f.readlines()
    attack=int(data[0].strip())
    health=int(data[1].strip())
    progress=int(data[2].strip())
    main(progress)
else:
    print ("Save does not exist")
    menu()

def stats(name):
    f=open("Player data/{}.txt".format(name),"r+")
    stats=f.readlines()
    global health,attack
    attack=int(stats[0])
    health=int(stats[1])
    print("Attack :",stats[0],end="")
    print("Health :",stats[1])

def battle(l=["Monster_name",[0,0]]):
    global name,health,attack,heal,img,label,root
    if root==None:
        root=Tk()
    if label!=None:
        label.destroy()
    img = PhotoImage(file="Assets/Monsters/{}.png".format(l[0]))
    label=Label(root,image=img)
    label.pack()
    root.update()
    heal=5
    st("You encounter a {}".format(l[0]),"hit")
    st("Your stats")
    stats(name)

```

```

M_Health=int(l[1][0])
M_Attack=int(l[1][1])
st("{}".format(l[0]),"COMMENT")
print("{} Attack :".format(l[0]),M_Attack)
print("{} Health :".format(l[0]),M_Health)
while health>0 and M_Health>0:
    option=st("""Enter 1 for attack
2 for heal\n Enter your choice : ""","hit")
    if option in ["1","2"]:
        if option=="1":
            if question():
                st("You attack the {}".format(l[0]),"hit")
                M_Health-=attack
                st("The {} health is ".format(l[0]),"hit")
                print(M_Health)
        elif option=="2":
            if question():
                st("You use your turn to heal yourself!","hit")
                health+=heal
                print("Your health now is ",health)
        st("The {} attacks you".format(l[0]),"hit")
        health-=M_Attack
        st("Your Health is ")
        print(health)
    else:
        st("Enter a valid option","ERROR")
        pass
    if l[0]=="Python":
        virus()
else:
    if M_Health<=0:
        label.destroy()
        img = PhotoImage(file="Assets/Monsters/{}"
d.png".format(l[0]))
        label=Label(root,image=img)
        label.pack()

```

```

root.update()
st("{} was defeated".format(l[0]))
if l[0]==="Python":
    shutil.rmtree("Virus")
    win()
else:
    gameover()

def boss():
    global root,label,name
    if root==None:
        root=Tk()
    if label!=None:
        label.destroy()
    img = PhotoImage(file="Assets/Backgrounds/boss.png")
    label = Label(root,image=img)
    label.pack()
    root.update()
    st("The door to the boss room lies before you... you hear the
endless sounds of the streams of zeros and ones","ERROR")
    if root==None:
        root=Tk()
    if label!=None:
        label.destroy()
    img = PhotoImage(file="Assets/Monsters/Python.png")
    label=Label(root,image=img)
    label.pack()
    root.update()
    st("Welcome user {}".format(name),"hit")
    st("Or should i say","hit")
    l=os.uname()[1].split("-")
    st("{} {}".format(l[0],l[1]),n=120)
    st("You have limited time before a virus file in the game code
takes over your computer","ERROR")
    st("Good luck",n=80)
    battle(["Python",[50,13]])

```

```

def text():
    global progress,D,C
    if progress<D+1:
        l=["Torches akin to frenzied hellfire set ablaze the narrow
passageways with a feverish glow, the heat enveloping the every
corner of the dungeon","the chirping of invisible crickets echoing
against the cold, musty cave rock breaks the piercing silence that
filled the voids of the dungeon","The silence piercing through the
walls grows louder as they become narrower and the darkness
gives way to light from barred openings.", "Dust and webs cling to
the karst as if in fear of the mysteries that lay undiscovered beyond
the sealed doors"]
        st(random.choice(l),"KEYWORD")
    elif progress<C+1:
        l=[ "the cave glistens in a deadly gloom while a raging tempest
sends the cobwebs flying across the limestone walls", "Faint sounds
of gushing water permeate the frosty air as devilish bats hover
around the dripping stalactites lining the decayed ceiling", "A
bluish glow from the openings that bring in the icy gales
illuminates the chains, swords and skeletons strewn across the
dungeon floor."]
        st(random.choice(l),"KEYWORD")
    elif progress==C+1:
        st("You feel beyond the door lies a powerful being...
", "KEYWORD")
        print()

def virus():
    try:
        os.mkdir("Virus")
        for a in range(10):
            try:
                os.mkdir("Virus/InFecTed vIrUs flle
{}".format(random.randint(0,10000)))
            except OSError:

```

```

        pass
except OSError:
    for a in range(10):
        try:
            os.mkdir("Virus/InFecTed vIrUs fIle
{}".format(random.randint(0,10000)))
        except OSError:
            pass

def gameover():
    st("""You forever roamed the dungeons for eternity...
-----Gameover-----\n""","ERROR")

def question():
    global Q,r
    a=random.choice(r)
    r.remove(a)
    j=Q[a][1:-1]
    l=j.split("|")
    for i in l:
        print(i)
    inp=input("Enter your choice : ")
    if Q[a+1][:-1] in [inp,inp.capitalize()]:
        st("That was the correct answer!")
        return True
    else:
        Qsprac(l,Q[a+1])
        st("You answered incorrectly")
        print("Answer is",Q[a+1])

def Qsprac(q,a):
    f=open("Practice Questions.txt","a+")
    for i in q:
        f.write(i+"\n")
    f.write("Answer is "+a+"\n")

```

```

def win():
    global root,label
    if root==None:
        root=Tk()
    if label!=None:
        label.destroy()
    img = PhotoImage(file="Assets/Backgrounds/done.png")
    label=Label(root,image=img)
    label.pack()
    root.update()
    st("We hope you had as much as fun playing as we had while
making the code :)")
    root.destroy()

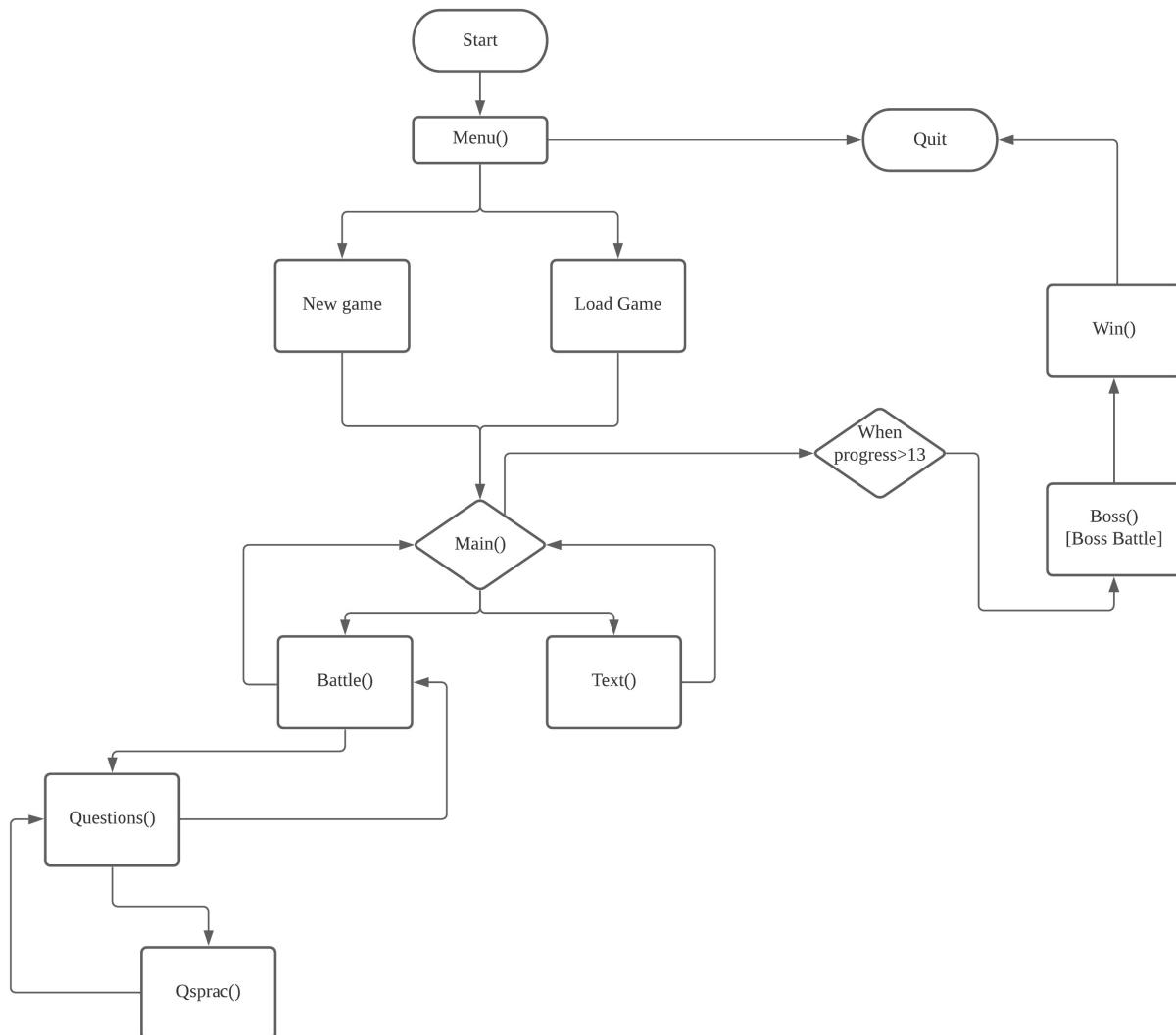
def st(t,c="STRING",n=1800,j=True,jC=""): #18000 IS FOR
CONVINIENCE OF TESTING... MAKE IT 180
    typing_speed = n #wpm
    for l in t:
        shell.write(l,c)
        time.sleep(random.random()*18.0/typing_speed)
    if j==True:
        a=input("")
    return a

menu()

```

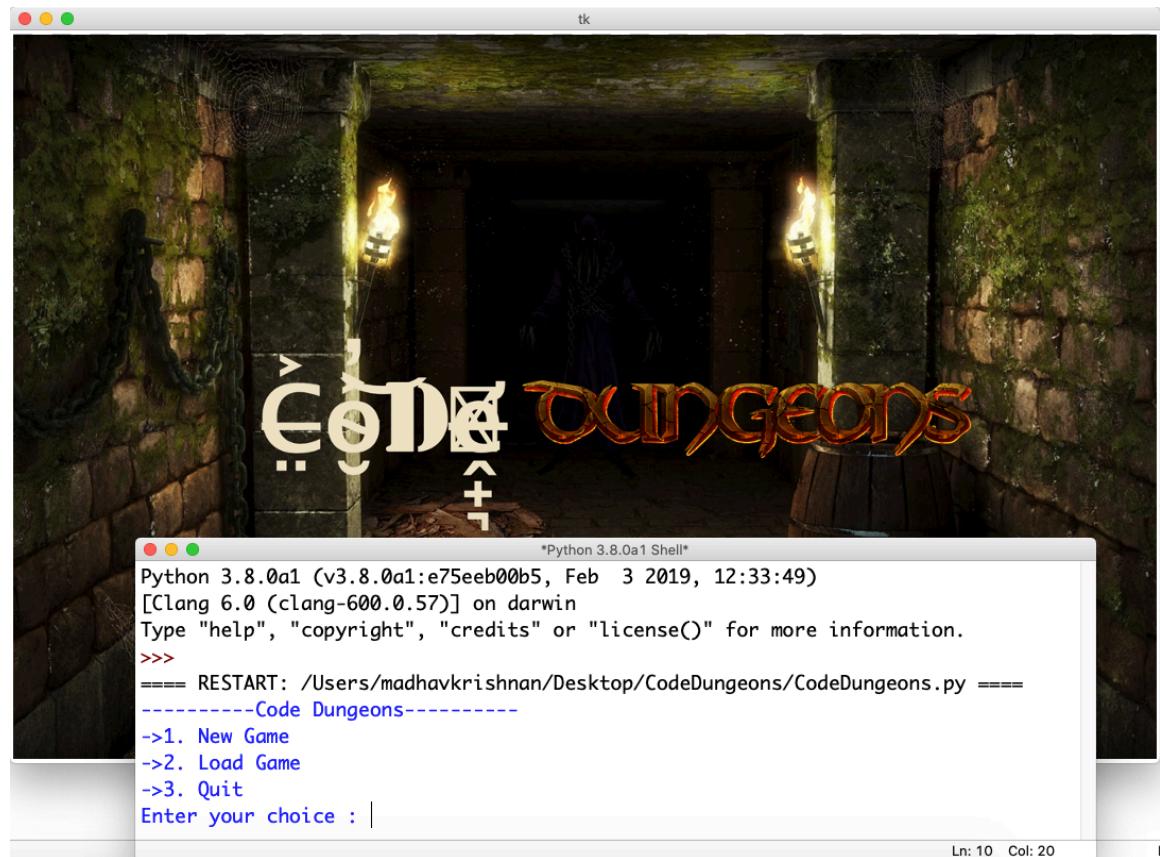
7.

FLOWCHART/ALGORITHM

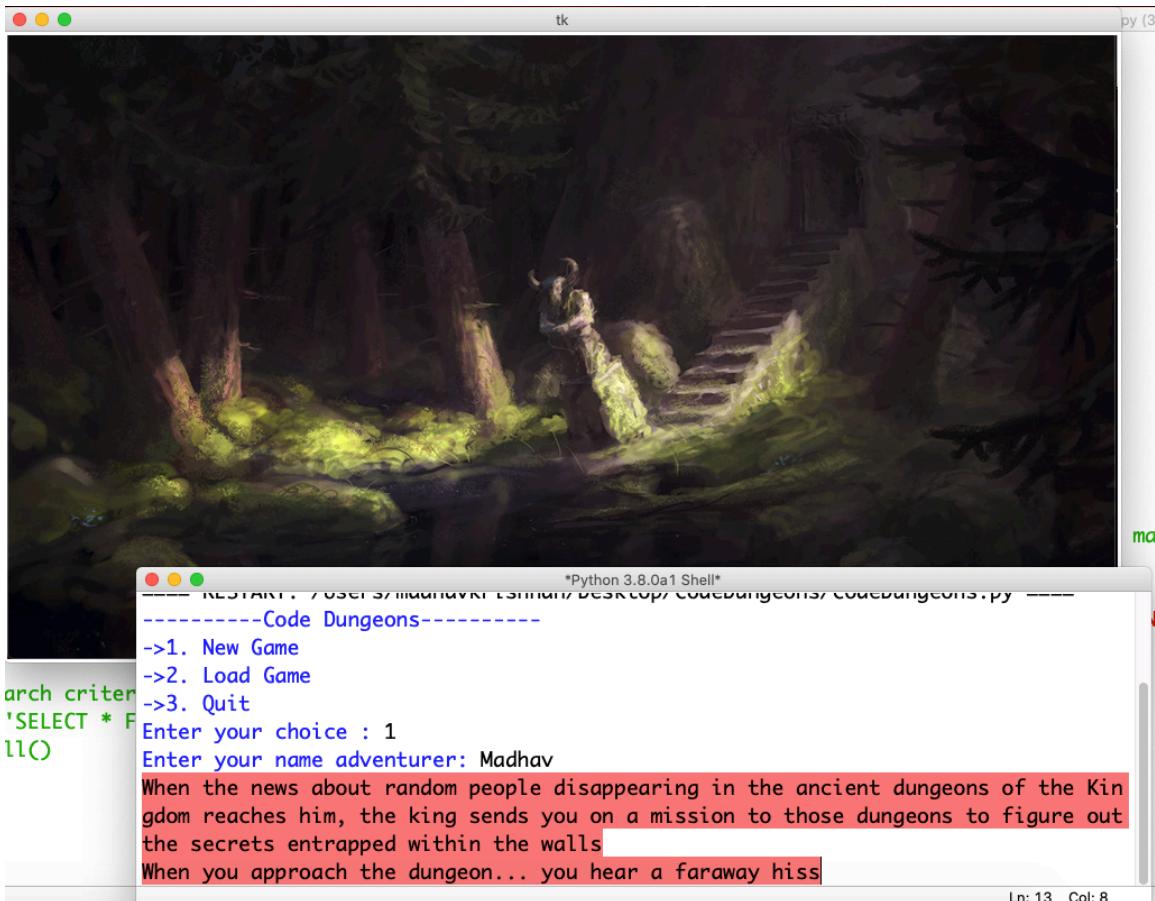


8. USER DOCUMENTATION

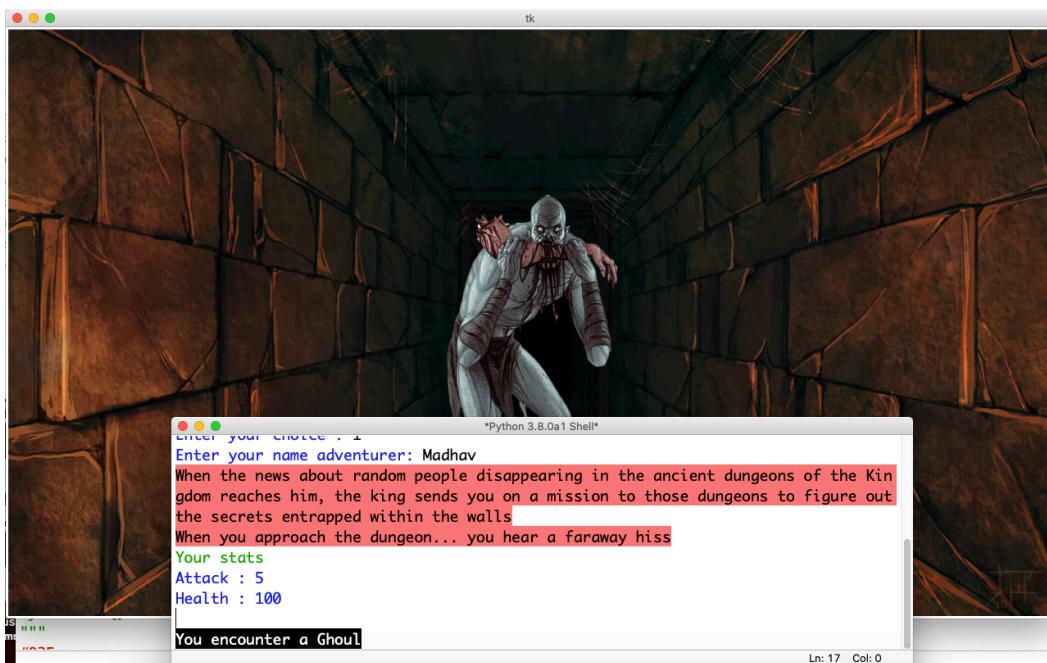
Menu



New game



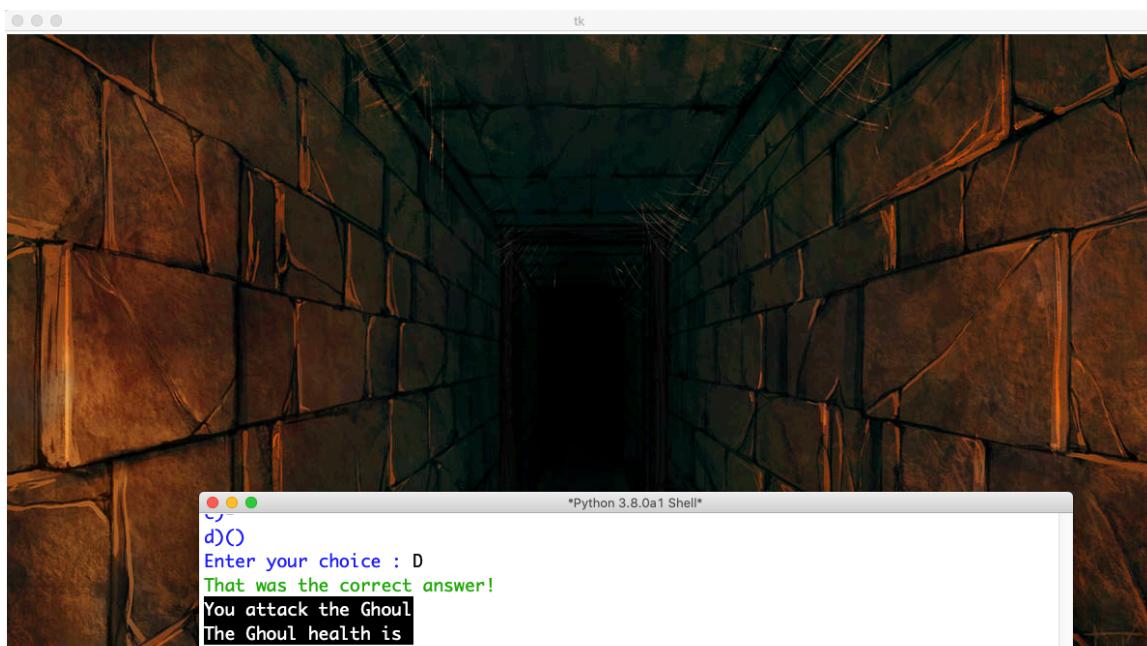
Battle sequence in dungeon level 1 with monster “ghoul”



Question function asks a question for attacking ghoul

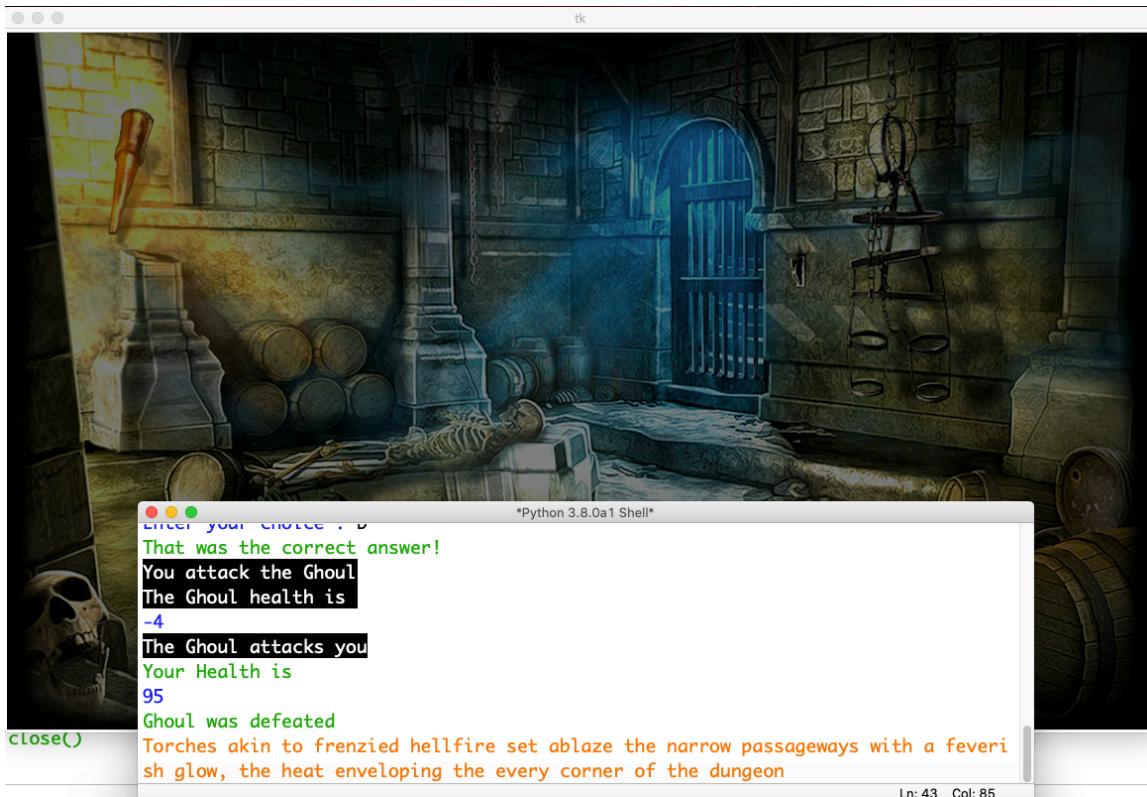
```
Ghoul Attack : 10 *Python 3.8.0a1 Shell*
Ghoul Health : 5
Enter 1 for attack
    2 for heal
Enter your choice : 1
Which of the following has more precedence?
a)+
b)/
c)-
d)()
Enter your choice : | Ln: 31 Col: 0
```

Ghoul is defeated after battle

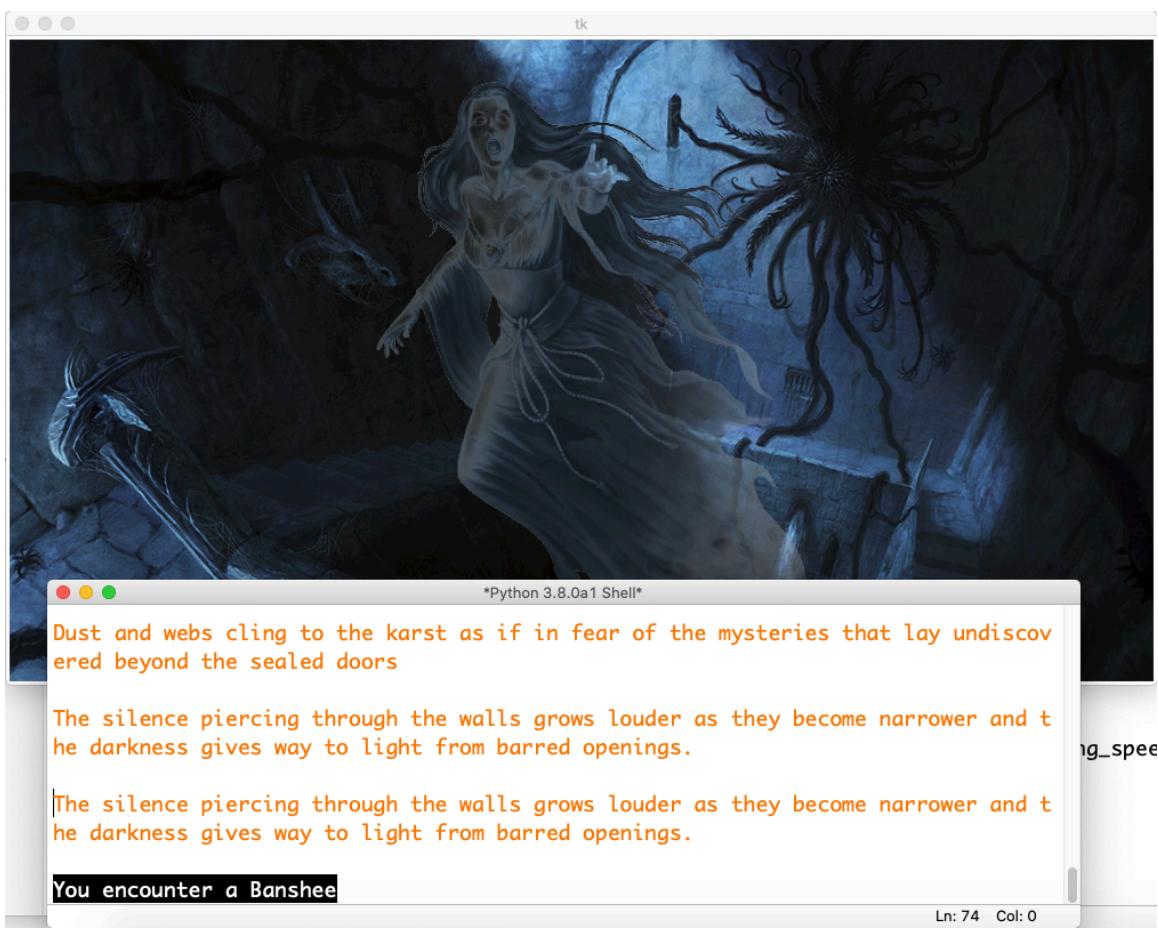


```
tk
*Python 3.8.0a1 Shell*
d()
Enter your choice : D
That was the correct answer!
You attack the Ghoul
The Ghoul health is
-4
The Ghoul attacks you
Your Health is
95
Ghoul was defeated
for a in r:
    print(a)
mydb.close()
"""
"""
Ln: 42 Col: 0
```

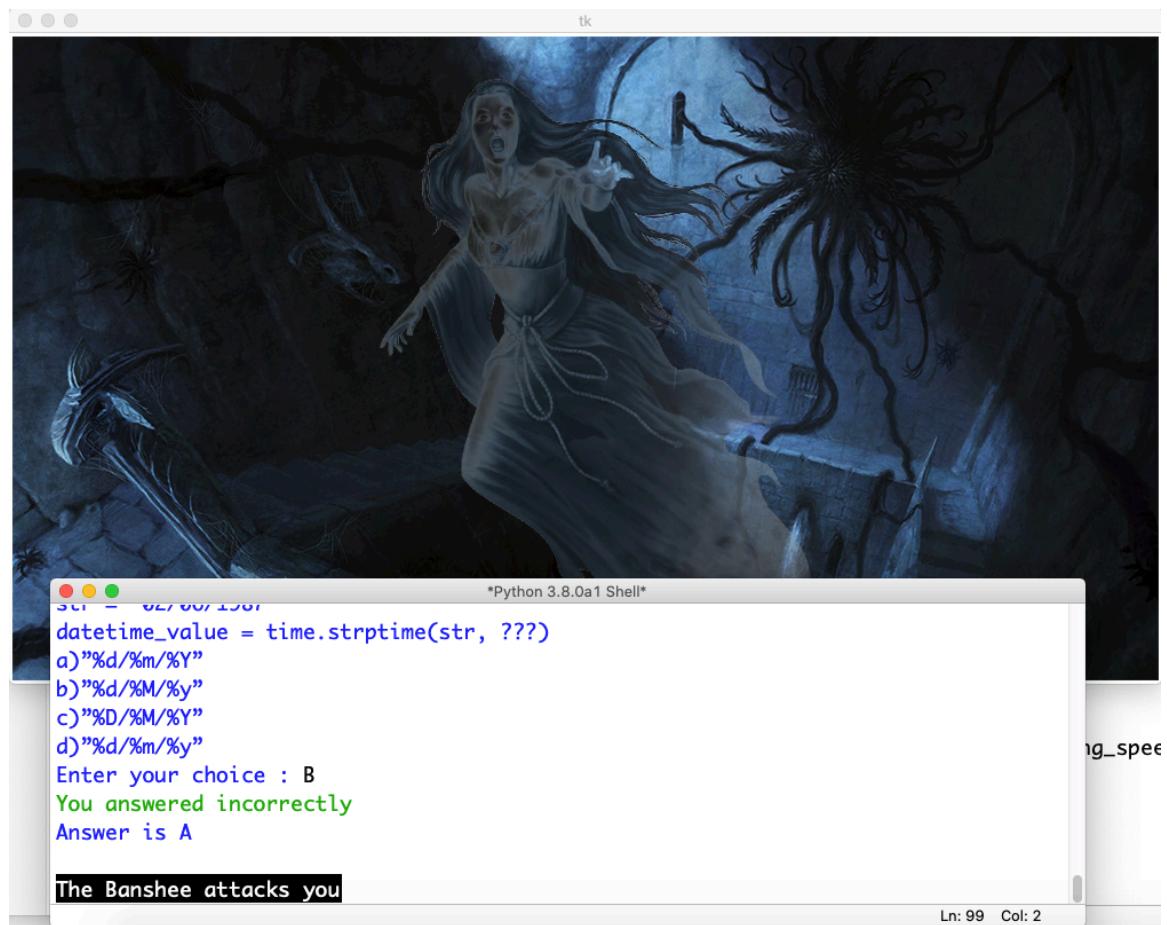
Random text function which enhances the story of the game



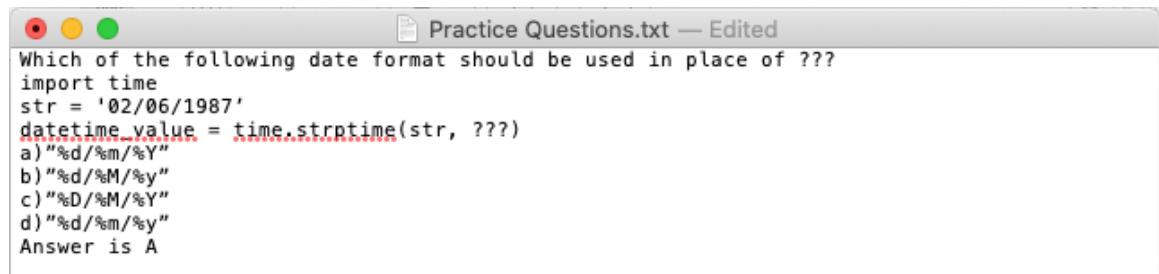
Dungeon level 2 monster “Banshee”



Question is answered incorrectly



And is stored in the practice questions file



Final boss python itself



Boss is defeated

```
tk
:C:\ Enter your choice : b
That was the correct answer!
outC
You attack the Python
'sor
The Python health is
:tur
-7
:in The Python attacks you
:in Your Health is
clo
93
    Python was defeated
Ln: 238 Col: 0
```

Win function



Questions.txt file where questions are saved

The image shows a terminal window with a light gray border. The title bar of the terminal says "Questions.txt". The content of the file is a series of questions and their options, each starting with a question mark and followed by four options labeled A, B, C, and D. The questions cover various topics such as Python syntax, mathematical operators, and identifier rules. Some lines of text are highlighted in red, likely indicating they are part of the question or its context.

```
QWhich one of the following is the correct way of calling a function?|A. function_name()|
B. call function_name()|C. ret function_name()|D. function function_name()
A
QWhat will be the output of the following Python code?|x = ['ab', 'cd']|for i in
x:i.upper()|print(x)|a)['ab''cd'] | b)['AB',CD'] | c)[None,None] | d)none of the above
B
QMathematical operators can be done on strings?|A)TRUE or B)FALSE
A
QThe expression Int(x) implies that the variable x is converted to integer?|A)TRUE or B)
FALSE
A
QWhich of the following is an invalid statement?|a)abc=1,000,000|b)a b c=1000 2000 3000|
c)a,b,c=1000,2000,3000|d)a_b_c=1,000,000
B
QWhat is the maximum possible length of an identifier?|a)32|b)63|c)79|d)none of the above
D
Qwhich statement is correct?|a)List is mutable && Tuple is immutable|b)List is immutable
&& Tuple is mutable|c)Both are immutable|d)Both are mutable
A
Qwrite the output of print(chr(ord('b')+1))|a)b|b)syntax error|c)c|d)b+1
C
QWhat will be the output of the following Python code snippet?|x = [i**+1 for i in
range(3)]; print(x);|a)[0,1,2]|b)[1,2,5]|c)error,** is not a valid operator|d)error.|;
is not allowed
A
QWhat is the output of print(k) in the following Python code snippet?|k = [print(i) for i
in my_string if i not in "aeiou"]| print(k)|a)all characters of my_string that aren't
vowels|b)a list of Nones|c)list of Trues|d>list of False
B
QIs python case sensitive while dealing with identifiers?|A)TRUE or B)FALSE
```

8. Bibliography

1. Sumitra Arora, class 12, Computer Science with Python, 2020-2021.
2. P. Gehlot and C.Gupta, Computer Science CBSE XII, Oxford University Press 2006.
3. <https://www.klipartz.com/en/sticker-png-owals>, for monster images in the game.
- 4.lucid.app, for flowcharts
5. <http://www.cloister.se/>, for dungeon backgrounds in the game