Logo, company name

Description automatically generated

ADVENTURE OF A THOUSAND WORDS

**PROGRAMMING AND PROBLEM SOLVING**

**NIHAR PHANSALKAR NARASIMHA LOKHESH NIDADAVOLE MUDIT GARG 21070122102 21070122101 21070122098**

**COMPUTER SCIENCE AND ENGINEERING (2021 – 25)**

# **INTRODUCTION TO PROJECT IDEA**

The project is inspired by RPG (Role Playing Games) like Skyrim which usually gives the player, the freedom to choose their own path in a roughly pre-defined story thorough a set of options.

The project essentially gives the user a taste of the writer’s hat without burdening them with its responsibilities allowing an inculcation of creative thinking smoothly.

This game aims to provide the user a set of options at each stage of the game where the player initially begins with a pre-defined set of stats, some of which will alter throughout the game as he proceeds through the path he has chosen.

The users can use this basic game to understand and analyze the working of a game, convert their ideas into a logic/ algorithm and thus implement those as a full-fledged working advanced gaming environment.

# **PURPOSE OF PROJECT**

The purpose of this project is to apply the programming

knowledge into a real- world situation/problem and understand how programming skills helps in developing a good software and increase logical reasoning.

Also, to :

* Master the fundamentals of writing Python scripts
* Write Python functions to facilitate code reuse
* Make the code perfect by handling errors
* Work with the Python standard library
* Explore Python's object-oriented features
* Apply object-oriented programming principles effectively when developing small to medium sized projects.
* Write effective procedural code to solve small to medium sized problems.
* To demonstrate a breadth of knowledge in computer science, as exemplified in the areas of systems, theory and software development.
* Demonstrate ability to conduct research or applied Computer Science project, requiring writing and presentation skills which exemplify scholarly style in computer science.

# **INTRODUCTION TO PROGRAMMING LANGUAGE: WHAT 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 emphasizes 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.

Python was created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented, and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library.

Python was conceived in the late 1980s as a successor to the ABC language.

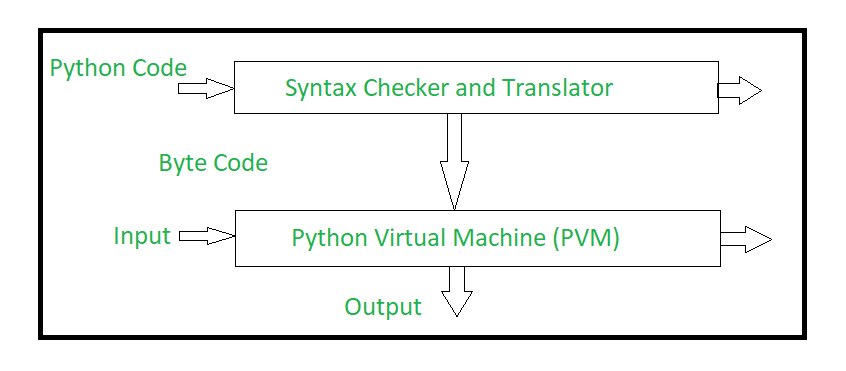
# **HOW PYTHON WORKS?: INTERNAL WORKING OF PYTHON**

Python is called an interpreted language. Python uses code modules that are interchangeable instead of a single long list of instructions that was standard for functional programming languages. The standard implementation of python is called “cpython”. It is the default and widely used implementation of the Python.  
Python doesn’t convert its code into machine code, something that hardware can understand. It actually converts it into something called byte code. So, within python, compilation happens, but it’s just not into a machine language. It is into byte code and this byte code can’t be understood by CPU. So, we need actually an interpreter called the python virtual machine. The python virtual machine executes the byte codes.

*The Python interpreter performs following tasks to execute a Python program :*

**Step 1 :** The interpreter reads a python code or instruction. Then it verifies that the instruction is well formatted, i.e., it checks the syntax of each line. If it encounters any error, it immediately halts the translation and shows an error message.

**Step 2 :**If there is no error, i.e., if the python instruction or code is well formatted then the interpreter translates it into its equivalent form in intermediate language called “Byte code”. Thus, after successful execution of Python script or code, it is completely translated into Byte code.

**Step 3 :**Byte code is sent to the Python Virtual Machine(PVM).Here again the byte code is executed on PVM. If an error occurs during this execution, then the execution is halted with an error message.

# **WHY PYTHON? – ADVANTAGES OF USING PYTHON**

Python is a dynamic, interpreted (bytecode-compiled) language. There are no type declarations of variables, parameters, functions, or methods in source code. This makes the code short and flexible, and you lose the compile-time type checking of the source code. Python tracks the types of all values at runtime and flags code that does not make sense as it runs.

The Advantages of using Python are:

* Versatile and Easy to Use Language
* Easy to learn and fast to develop
* Efficient Programming
* Well Structured, Readable and easy to Understand
* Is Open Source
* Has a vibrant community of users, learners and experts for guidance at each step
* Has a Built-in library for basically everything: from web development, through game development, to machine learning
* User Friendly Data Structures
* High Level Language
* Dynamically typed language (No need to mention data type based on value assigned, it takes data type)
* Object-oriented language
* Portable across Operating systems
* Portable and Interactive
* Supports Platform Independent Program Development

# **MINIMUM SYSTEM REQUIREMENTS FOR THE PROJECT**

**HARDWARE**

A PC with following or similar configuration:

* Intel Atom® processor or Intel® Core™ i3 processor
* 1 GB RAM or 1GB free disk space
* 1024x768 screen resolution

**SOFTWARE**

* Python 2.4

OPERATING SYSTEM:

* Windows:
  + Microsoft Windows 10/8/7/Vista/2003/XP (incl.64-bit)
* Mac:
  + Mac OS X 10.8 or higher
* Linux:

**PYTHON MODULES**

* time
* sys
* random

# **ALGORITHM FOR THE PROGRAM**

1. Initialize program
2. Import Modules – time, sys, random
3. Create Class for defining character with required attributes and print them
4. Create object for user inputs for the above Character Class
5. Repeat the above steps for defining Weapon of the character
6. Create classes for each chapter of the Game Story
7. Each scene of the chapter is defined as a function
8. Whichever scenes have a choice for the user, a variable is created and the value from the user input is assigned to it. Respective choices have print statements displayed above. The user choice is returned and displayed on the screen
9. The scenes with repeated inputs have been equipped with loops to perform the required function
10. Some scenes include functions from imported modules to manage the input/output mechanism while performing required function.
11. For optimizing output, f-string format has been used
12. Based on the choices from the user assigned to appropriate variables, the next scene is being displayed by using the call function.
13. End the program once the story reaches the end.

# **SOURCE CODE**

import time

import sys

import random

def Invalid\_Syntax(user\_input):

if user\_input == 'A' or user\_input == 'B' or user\_input == 'C':

return 1

else:

return 0

class Character:

character\_gender = ""

character\_name = ""

character\_health = 100

character\_weapon = "Sword"

character\_intelligence = 50

character\_power = 50

character\_charisma = 50

character\_coins = 0

def print\_stats(self):

print("Name:",self.character\_name)

print("Gender:",self.character\_gender)

print("Health:",self.character\_health)

print("Intelligence:",self.character\_intelligence)

print("Power:",self.character\_power)

print("Charisma:",self.character\_charisma)

print("Coins:",self.character\_coins)

print()

obj\_character = Character()

class Weapon:

weapon\_damage = 0

weapon\_special\_ability = ""

class Choose\_Character:

n=0

while True:

if n < 3:

gender = input("Please choose what gender is your character:\nMale\nFemale\nOthers\n")

gender = gender.title()

gender\_list = ['Male', 'Female', 'Others']

if gender not in gender\_list:

n+=1

continue

else:

obj\_character.character\_gender = gender

break

else:

sys.exit("Repeated Invalid Output")

name = input("Please enter a name for your character\n")

obj\_character.character\_name = name.title()

print()

class Chapter\_Cave:

def scene\_1(self):

print("So, what will you do? (Please select the letters)")

print("A. Explore the cave")

print("B. Run straight into the cave")

print("C. Call out for people")

choice\_1 = input().upper()

while Invalid\_Syntax(choice\_1) == 0:

print("Please choose the correct option.")

choice\_1 = input().upper()

Invalid\_Syntax(choice\_1)

return choice\_1

def scene\_2\_a(self):

print()

print("You start looking around the cave. The damp air, the feeling of uncertainity, the cold touch of the walls, all caution you.", end="")

print(' "Maybe I should come here with some help?" Is what you think to yourself, when suddenly, your instinct warns you.')

print("You hear soft footsteps, approaching towards you, not too quickly, but with quite a steady pace.")

print("What do you do?")

print("A. Freeze in place")

print("B. Look in the direction of the footsteps")

print("C. Close your eyes and sprint down the cave")

choice\_4\_a\_a = input().upper()

while Invalid\_Syntax(choice\_4\_a\_a) == 0:

print("Please choose the correct option.")

choice\_4\_a\_a = input().upper()

Invalid\_Syntax(choice\_4\_a\_a)

return choice\_4\_a\_a

def scene\_2\_b(self):

print("As you sprint down the darkness of the cave, grinning, you hear a voice.",end="")

print(' Help!', end = "")

print(" What will you do?")

print("A. Ignore the voice and go ahead.")

print("B. Go to where the voice is coming from.")

choice\_4\_a\_b = input().upper()

while Invalid\_Syntax(choice\_4\_a\_b) == 0:

print("Please choose the correct option.")

choice\_4\_a\_b = input().upper()

Invalid\_Syntax(choice\_4\_a\_b)

return choice\_4\_a\_b

def scene\_2\_c(self):

print("You use all the strength your lungs can conjure to shout asking for anyone's presence")

print("Your voice echoes, but you get no reply, instead", end = "")

print(" you hear soft footsteps, approaching towards you, not too quickly, but with quite a steady pace.")

print("As you turn your head to look in the direction of the footsteps, you see an old woman. She is sitting with her eyes closed", end ="")

print(" seeming to be in deep meditation.")

print(" She suddenly opens her eyes and stares into your's, as if piercing your soul itself.", end="")

print(''' You try to appreciate her beauty, but your heart feels cold.\n"Do not make the same mistake I made. You will only end up as a dead man, stalking the night aimlessly."''', end="")

print(" Turn back! Run! \*Screams gibberish\*")

print("A. Try to fight the woman")

print("B. Sprint down the cave")

choice\_4\_a\_c = input().upper()

while Invalid\_Syntax(choice\_4\_a\_c) == 0:

print("Please choose the correct option.")

choice\_4\_a\_c = input().upper()

Invalid\_Syntax(choice\_4\_a\_c)

return choice\_4\_a\_c

def scene\_2\_conclude\_a(self):

print("As you freeze up, you feel a peculiar coldness on your neck. ", end="")

print("Upon touching your neck, you feel a familiar human touch.", end = "")

print("As you turn your head, you see an apparition. It's face looks like that of a handsome man,", end ="")

print(" however, it is weirdly disturbing. The loss of a hand makes it no more appealing.", end = "")

print(" It's eye, rolling aimlessly in it's socket, suddenly turns and stares into your's, as if piercing your soul itself.", end="")

print(''' As you smell it's horrid breath, you hear, "Do not make the same mistake I made. You will only end up as a dead man, stalking the night aimlessly."''', end="")

print(" Turn back! Run! \*Screams gibberish\*")

print()

print("A. Try to fight the apparition")

print("B. Sprint down the cave")

choice\_4\_a\_conclude\_a = input().upper()

while Invalid\_Syntax(choice\_4\_a\_conclude\_a) == 0:

print("Please choose the correct option.")

choice\_4\_a\_conclude\_a = input().upper()

Invalid\_Syntax(choice\_4\_a\_conclude\_a)

return choice\_4\_a\_conclude\_a

def scene\_2\_conclude\_b(self):

print("As you turn your head, you see an apparition. It's face looks like that of a handsome man,", end ="")

print(" however, it is weirdly disturbing. The loss of a hand makes it no more appealing.", end = "")

print(" It's eye, rolling aimlessly in it's socket, suddenly turns and stares into your's, as if piercing your soul itself.", end="")

print(''' As you smell it's horrid breath, you hear, "Do not make the same mistake I made. You will only end up as a dead man, stalking the night aimlessly."''', end="")

print(" Turn back! Run! \*Screams gibberish\*")

print()

print("A. Try to fight the apparition")

print("B. Sprint down the cave")

choice\_4\_a\_conclude\_b = input().upper()

while Invalid\_Syntax(choice\_4\_a\_conclude\_b) == 0:

print("Please choose the correct option.")

choice\_4\_a\_conclude\_b = input().upper()

Invalid\_Syntax(choice\_4\_a\_conclude\_b)

return choice\_4\_a\_conclude\_b

def scene\_2\_5\_a(self):

print("You ignore the voice and run ahead. You see the cave exit in the distance.",end="")

print("As you are about to rejoice your fiding of the exit, you come across a big boulder.",end="")

print("It takes you weeks to move the boulder, but you finally do it. Your power has increased due to this.")

obj\_character.character\_power += 20

def scene\_2\_5\_b(self):

print("You run towards the voice that is calling out for help.",end="")

print(f"You see a man who is badly wounded. You run towards him. Since your intelligence is {obj\_character.character\_intelligence}",end="")

print("You are able to perform basic aid to help him.")

print("The both of you have a talk, and he let's you know how to get out of the cave from a shortcut.",end="")

print("As you turn away for a second to look for the shortcut, the man disappears. You walk out of the cave. Your charisma has increased")

obj\_character.character\_charisma += 20

obj\_cavechp = Chapter\_Cave()

class Story:

print("Around 250 A.D, the world had fallen into utter chaos. Today's historians would come to call this era as the dark ages. An era, which had lost it's light.", end = " ")

print(f"Around this time, in a small town called Alryne, a child was born, who was named, {obj\_character.character\_name}")

print()

print(f"Born into poverty, the young one did not have too many options. Their father was a stubborn man, whose objective in life had become war.", end = "")

print(f"He dreamed of achieving glory day and night, and only pushed his own expectations onto {obj\_character.character\_name}.")

print()

print(f"Many years passed by, and {obj\_character.character\_name} grew to the age of 15.")

print(f"Since a very young age, {obj\_character.character\_name} was fond of exploration. And what a day it is today, as you come across a cave and enter. That is right,", end = "")

print(f" you are {obj\_character.character\_name}. Your choices will affect how {obj\_character.character\_name}'s life goes.")

print("Your Current stats are as follows, they will change as your progress.\n")

obj\_character.print\_stats()

chp1\_result\_1 = obj\_cavechp.scene\_1().upper()

if chp1\_result\_1 == 'A':

result\_2\_a\_a = obj\_cavechp.scene\_2\_a().upper()

result\_2\_b = "NIL"

result\_2\_a\_c = "NIL"

elif chp1\_result\_1 == 'B':

result\_2\_b = obj\_cavechp.scene\_2\_b().upper()

result\_2\_a\_a = "NIL"

result\_2\_a\_c = "NIL"

elif chp1\_result\_1 == 'C':

result\_2\_a\_c = obj\_cavechp.scene\_2\_c().upper()

result\_2\_b = "NIL"

result\_2\_a\_a = "NIL"

if result\_2\_a\_a != "NIL":

if result\_2\_a\_a == 'A':

result\_2\_a\_c = obj\_cavechp.scene\_2\_conclude\_a().upper()

elif result\_2\_a\_a == 'B':

result\_2\_a\_c = obj\_cavechp.scene\_2\_conclude\_b().upper()

elif result\_2\_a\_a == 'C':

result\_2\_b = obj\_cavechp.scene\_2\_b().upper()

else:

print("Please choose the correct option")

if result\_2\_a\_c != "NIL":

if result\_2\_a\_c == 'A':

print(f"Since your current power is {obj\_character.character\_power}, you are unable to defeat the apparition, and it kills you.")

print(f"You lost.")

exit()

elif result\_2\_a\_c == 'B':

result\_2\_b = obj\_cavechp.scene\_2\_b().upper()

if result\_2\_b != "NIL":

if result\_2\_b == 'A':

obj\_cavechp.scene\_2\_5\_a()

elif result\_2\_b == 'B':

obj\_cavechp.scene\_2\_5\_b()

class Chapter\_Inn:

print(f"Entering a small town near Old Alryne {obj\_character.character\_name} disturbed now seeks out an Inn to rest in.")

def Scene\_3(self) :

print("You have no money ! How shall you convince the Innkeeper to let you in ? (Please select the letters)")

print("A. Sway the Innkeeper to let you in")

print("B. Intimidate the Innkeeper to let you in")

choice\_2 = input().upper()

while Invalid\_Syntax(choice\_2) == 0:

print("Please choose the correct option.")

choice\_2 = input().upper()

Invalid\_Syntax(choice\_2)

return choice\_2

def Scene\_3\_a(self) :

print()

if obj\_character.character\_charisma <60 :

print("Nice try mate but you shan't be bewitching me ")

print("You have no choice left now but to Intimidate him. The poor bastard !")

choice\_2 = 'B\_'

return choice\_2

else :

print("Sure mate I'd be delighed to have thee")

print(f"{obj\_character.character\_name} enters the inn still traumatised by the encounter with the Apparition.")

print("Feeling tired you conceal yourself under the warm blanket while lying on bed.")

print("How long do you plan to sleep ?")

print("A. 2 hours")

print("B. 5 hours")

print("C. 7 hours")

choice\_3 = input().upper()

while Invalid\_Syntax(choice\_3) == 0:

print("Please choose the correct option.")

choice\_3 = input().upper()

Invalid\_Syntax(choice\_3)

return choice\_3

def Scene\_3\_b(self) :

print()

print("Umm yeah sure mate no need to get edgy it's alright the room's over there")

print("Annoyed and still contemplating about the encounter you lie down on the bed knowing that you can't stay at the Inn too long having threatened the Innkeeper ")

print(" How long do you plan to sleep ?")

print(" A. 1 hour")

print(" B. 2 hours")

print(" C. 3 hours")

choice\_4 = input().upper()

while Invalid\_Syntax(choice\_4) == 0:

print("Please choose the correct option.")

choice\_4 = input().upper()

Invalid\_Syntax(choice\_4)

return choice\_4

def Scene\_choice\_3\_a(self) :

time.sleep(2)

obj\_character.character\_health += 10

obj\_character.character\_power += 5

obj\_character.character\_charisma += 10

print(" You slept for only 2 hours")

obj\_character.print\_stats()

def Scene\_choice\_3\_b(self) :

time.sleep(5)

obj\_character.character\_health += 20

obj\_character.character\_power += 10

obj\_character.character\_charisma += 15

print(" You slept for 5 hours")

obj\_character.print\_stats()

def Scene\_choice\_3\_c(self) :

time.sleep(7)

obj\_character.character\_health += 25

obj\_character.character\_power += 15

obj\_character.character\_charisma += 20

print(" You slept for 7 hours")

obj\_character.print\_stats()

def Scene\_choice\_4\_a(self) :

time.sleep(1)

obj\_character.character\_health += 5

obj\_character.character\_charisma += 5

print(" You slept for 1 hour")

obj\_character.print\_stats()

def Scene\_choice\_4\_b(self) :

time.sleep(2)

obj\_character.character\_health += 10

obj\_character.character\_power += 5

obj\_character.character\_charisma += 10

print(" You slept for only two hours")

obj\_character.print\_stats()

def Scene\_choice\_4\_c(self) :

time.sleep(3)

obj\_character.character\_health += 15

obj\_character.character\_power += 8

obj\_character.character\_charisma += 12

print(" You slept for only three hours")

obj\_character.print\_stats()

obj\_innchp = Chapter\_Inn()

chp2\_result\_1 = obj\_innchp.Scene\_3().upper()

if chp2\_result\_1 == 'A':

result\_3\_a = obj\_innchp.Scene\_3\_a().upper()

result\_3\_b = "NIL"

elif chp2\_result\_1 == 'B':

result\_3\_b = obj\_innchp.Scene\_3\_b().upper()

result\_3\_a = "NIL"

if result\_3\_a != "NIL":

if result\_3\_a == 'A':

result\_inn\_choice\_3\_a = obj\_innchp.Scene\_choice\_3\_a()

elif result\_3\_a == 'B':

result\_inn\_choice\_3\_a = obj\_innchp.Scene\_choice\_3\_b()

elif result\_3\_a == "B\_":

result\_inn\_choice\_3\_a = obj\_innchp.Scene\_3\_b()

elif result\_3\_a == 'C':

result\_inn\_choice\_3\_a = obj\_innchp.Scene\_choice\_3\_c()

if result\_3\_b != "NIL":

if result\_3\_b == 'A':

result\_inn\_choice\_3\_b = obj\_innchp.Scene\_choice\_4\_a()

elif result\_3\_b == 'B':

result\_inn\_choice\_3\_b = obj\_innchp.Scene\_choice\_4\_b()

elif result\_3\_b == 'C':

result\_inn\_choice\_3\_b = obj\_innchp.Scene\_choice\_4\_c()

class Chapter\_Town :

def Scene\_5(self) :

print("You walk through the narrow and dim corridor exiting the Inn with a rather cautiously optimistic outlook.")

print("Intending to find the ancient map, you head towards the local Imperial archive.")

print("Little do you know that someone had already gotten to it first. You hear the aghast from the bystanding crowd. You see the look on Imperial Guards faces. The archive was burnt down to the last book. All knowledge of the past now lost.")

print("But as the Gods willed it you hear whispers also. Rumors that can lead you to them. The bandits who stole the Map. Rather crudely had they attempted to steal this old map giving away what they were after. Yet the Imperials had other concerns than to chase after some bandits and a mythical map. They had a war to prepare for.")

print(" You aren't able to eavesdrop a whole lot from the crowd. But you are able to make out possible ways in which you could find the bandits.")

print(" A. You overhear a middle aged man being melancholic and walking away from the scene. There is something different about him. You follow him")

print(" B. You eavesdrop on the conversion of two Imperial Officers. You hear them talking about the Bandits possible location. You can try to get the information at the Imperial Office")

print(" C. You see a couple of suspicious looking folk wearing some rusted gear who exit the scene in a very suspicious manner. You can try to find their whereabouts. They are possibly Bandits themselves.")

choice\_5 = input().upper()

while Invalid\_Syntax(choice\_5) == 0:

print("Please choose the correct option.")

choice\_5 = input().upper()

Invalid\_Syntax(choice\_5)

return choice\_5

def Scene\_6\_a(self) :

print()

print("You try to follow him upto a distance but fail to catch up to him due to the surrounding crowd. Now you try to look for him.")

if obj\_character.character\_charisma > 70 :

print(" You ask around the to know more about the Man")

print(" A villager informs you about the man's whereabouts and you start travelling there. ")

elif obj\_character.character\_intelligence > 60 :

print(" You follow in the last known direction of the man and look around & are able to find and follow his footsteps")

else :

time.sleep(10)

print("It took you ten hours to find the old man's house ! But hey better late than never right ? ")

def Scene\_6\_b(self) :

print()

print(" You reach the middling structure. Now you must enquire")

print(" A. You feel creative and under the guise of applying for the Military you get into the Office ")

print(" B. You try to bribe the small level officers into giving you the Bandits' whereabouts.")

choice\_scene\_6\_b = input().upper()

while Invalid\_Syntax(choice\_scene\_6\_b) == 0:

print("Please choose the correct option.")

choice\_scene\_6\_b = input().upper()

Invalid\_Syntax(choice\_scene\_6\_b)

return choice\_scene\_6\_b

def Scene\_6\_c(self) :

print()

print(" A dangerous approach. You sneakily follow them into a small compound which by the looks of it looks like a Outlaw house. What do you do ? ")

print(f" A. There seems to be {obj\_character.character\_weapon} in the frontyard pick it up and barge through the front door.")

print(f" B. Grab the {obj\_character.character\_weapon} and sneak in from the back.")

choice\_scene\_6\_c = input().upper()

while Invalid\_Syntax(choice\_scene\_6\_c) == 0:

print("Please choose the correct option.")

choice\_scene\_6\_c = input().upper()

Invalid\_Syntax(choice\_scene\_6\_c)

return choice\_scene\_6\_c

def Scene\_6\_a\_conclude(self) :

print()

print(" You enter the ramshackled house. The old man is a little startled by the visit but he lets you in.")

print(" He looks at your youthful appearance. Tears roll over his eyes.")

print(" He apologizes and asks you why you are here ? ")

print(" You respond by saying you saw him at the Archive. That you suspected that he was somehow related to the bandits and he could make this out by his dejected face.")

print(" You tell him that you require the map that was stolen. And that you need to find the location of the bandits camp if there exists one.")

print(" My son is part of their wretched gang. I haven't seen him in two years. If they are here then they maybe at a place not so far from here but what would still be the outskirts")

print(" I do not know if my Son is still alive. Young one would you please spare him and ask him to return if he is ? He isn't a barbaric man. I am assuming you are going to fight them.")

print(" You accept the old Man's plea and take an oath no to kill him were to find him.")

print(" He will wearing a star medallion given to him by his mother. He will never give that up. Atleast I hope not. He is a lanky fella and has small scar on his left ear.")

print(f" You thank the old man and say that you will do your best to help him and leave having got a lead but before you go he offers you his family {obj\_character.character\_weapon} which you take.")

def Scene\_6\_b\_a(self) :

print()

print(" A rather risky approach. They find out and you take an arrow to the Knee. Yes? You apply for the post ")

if obj\_character.character\_power > 70 :

print(" Welcome Young Lad. I see you have an impressive physique and we would be delighted to have you. Go to the armory see what fits you. Let me remind you that it isn't an easy post choose what suits you best and are comfortable with. Of course you won't have access to everything yet. You're still a rookie ")

elif obj\_character.character\_charisma > 60 :

print(" I come from a very martial background Sir. My father was a Marshall in Alryne I have been trained to for combat since childhood. If only you could give me a chance to stand for the virtues of the Empire ")

print(" Fine! We will watch your career with great interest! Go to the Armoury and see what you like & can use")

else :

print(" We got a maggot here on our hands don't we boys ! Yeah we will make you straight. Go to the armoury and pick that dagger up !")

def Scene\_6\_b\_b(self) :

print()

if obj\_character.character\_coins > 10 :

print(" Sure friend! He is at the Outskirts to the left of the Watchtower.")

else :

print(" Hmm..you don't have any money ! Beat it imbecile !")

print(" You now have to improvise. Try getting into the Armoury by posing as a recruit for the Imperial Military. Yes?")

choice\_scene\_6\_b = 'A\_'

return choice\_scene\_6\_b

def Scene\_6\_c\_a(self) :

print()

print("As you barge in through the front door, all eyes turn towards you. None of the gangsters are pleased.")

print('"Hey! Who is that?", one of them asks.')

print('"You know the rule lad, says a man who looks like the leader of the gang. A stranger, whether they be a danger or not, never leaves alive from this place."')

print("Grab your weapons! Let's get em.")

def Scene\_6\_c\_b(self) :

print()

print("As you sneak in from the back, you take down two gangsters, but one of them ring's the alarm, making everyone aware of your presence.")

print('"You know the rule lad. A stranger, whether they be a danger or not, never leaves alive from this place."')

print("Grab your weapons! Let's get em,",end="")

print(" shouts a person who looks like their leader.")

def Scene\_6\_b\_a\_conclude(self) :

print()

print(" You now enter the Armoury. Oh how magnicficent are these art of men ! However before taking the armoury you enter the nearby Records office and as fate would have it the manuscript that describes their hideout is right there infront of you")

print(f"After having seized and planning an escape through the windows. You take {obj\_character.character\_weapon} and walk out. Ready for a fight.")

obj\_townchp = Chapter\_Town()

chp3\_result\_1 = obj\_townchp.Scene\_5().upper()

if chp3\_result\_1 == 'A':

chp3\_result\_b = "NIL"

chp3\_result\_c = "NIL"

obj\_townchp.Scene\_6\_a()

obj\_townchp.Scene\_6\_a\_conclude()

elif chp3\_result\_1 == 'B':

chp3\_result\_b = obj\_townchp.Scene\_6\_b()

chp3\_result\_c = "NIL"

elif chp3\_result\_1 == 'C':

chp3\_result\_c = obj\_townchp.Scene\_6\_c()

chp3\_result\_b = "NIL"

if chp3\_result\_b != "NIL":

if chp3\_result\_b == 'A':

obj\_townchp.Scene\_6\_b\_a()

obj\_townchp.Scene\_6\_b\_a\_conclude()

else:

obj\_townchp.Scene\_6\_b\_b()

obj\_townchp.Scene\_6\_b\_a()

obj\_townchp.Scene\_6\_b\_a\_conclude()

if chp3\_result\_c != "NIL":

if chp3\_result\_c == 'A':

obj\_townchp.Scene\_6\_c\_a()

else:

obj\_townchp.Scene\_6\_c\_b()

class Chapter\_Fight:

def scene\_7(self):

print()

print("You have no other option, but to fight the leader of the gang.")

enemy\_health = 100

print(f"The boss's health is {enemy\_health}.")

print(f"Your health is {obj\_character.character\_health}.")

while enemy\_health >= 0 and obj\_character.character\_health >= 0:

print(f"A. Slash with the {obj\_character.character\_weapon}.")

print("B. Kick")

print("C. Punch")

fight\_style = input().upper()

if fight\_style == 'A':

your\_damage = random.randint(5, 20)

print(f"You did {your\_damage} damage to the boss.")

enemy\_health -= your\_damage

print(f"The boss's health is {enemy\_health}.")

elif fight\_style == 'B':

your\_damage = random.randint(0, 10)

print(f"You did {your\_damage} damage to the boss.")

enemy\_health -= your\_damage

print(f"The boss's health is {enemy\_health}.")

elif fight\_style == 'C':

your\_damage = random.randint(0, 10)

print(f"You did {your\_damage} damage to the boss.")

enemy\_health -= your\_damage

print(f"The boss's health is {enemy\_health}.")

else:

while Invalid\_Syntax(fight\_style) == 0:

print("Please choose the correct option.")

fight\_style = input().upper()

Invalid\_Syntax(fight\_style)

continue

print("The boss strikes back!")

boss\_damage = random.randint(0, 10)

print(f"The boss does {boss\_damage} damage to you!")

obj\_character.character\_health -= boss\_damage

print(f"Your health is {obj\_character.character\_health}.")

if enemy\_health <= 0 and obj\_character.character\_health >= 0:

print("You defeat the enemy leader, but are heavily bruised.")

print("As you try to stand up, you feel some support. A guy helps you up to your feet.")

print('You look at him, he has a scar on his left ear. He whispers, "I was on the wrong path. Seeing your fight I have understood, I will get strong to be good."')

print("I have a old man waiting for me. I shall go. Thank you.")

print("He helps you to your feet, and declares that he will not be a part of the gang. You smile.")

print("As you get out of the hideout, the moonlight falls on your face. You feel tired, but relieved.")

print("As you gaze far into the fields, you see a girl. It's as if she is calling to you.")

print('You think to yourself...\n"The adventure is not yet over huh?"')

print()

print("To be continued.")

print("Thank you for playing!")

else:

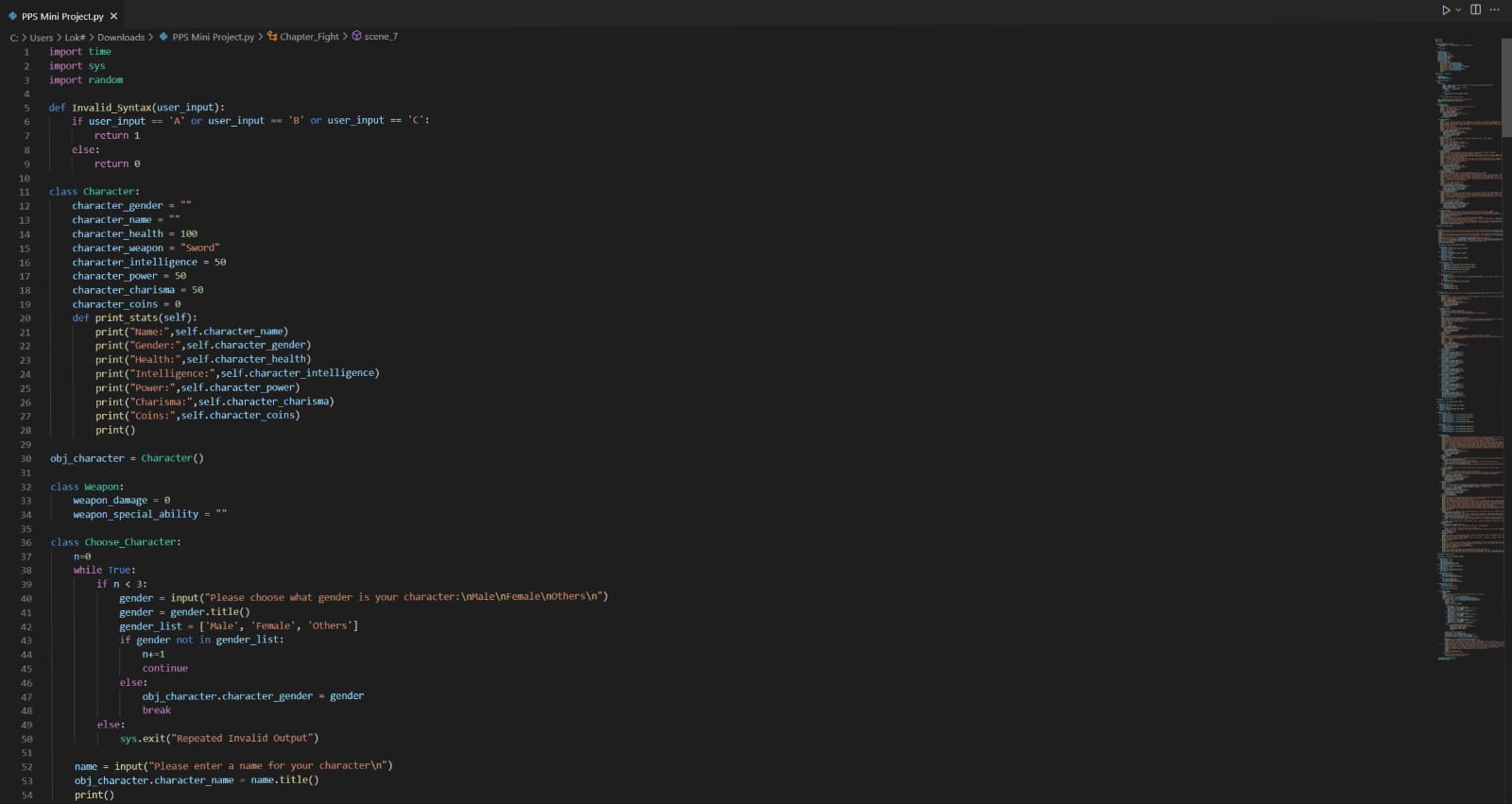
print("The boss defeats you and you die.")

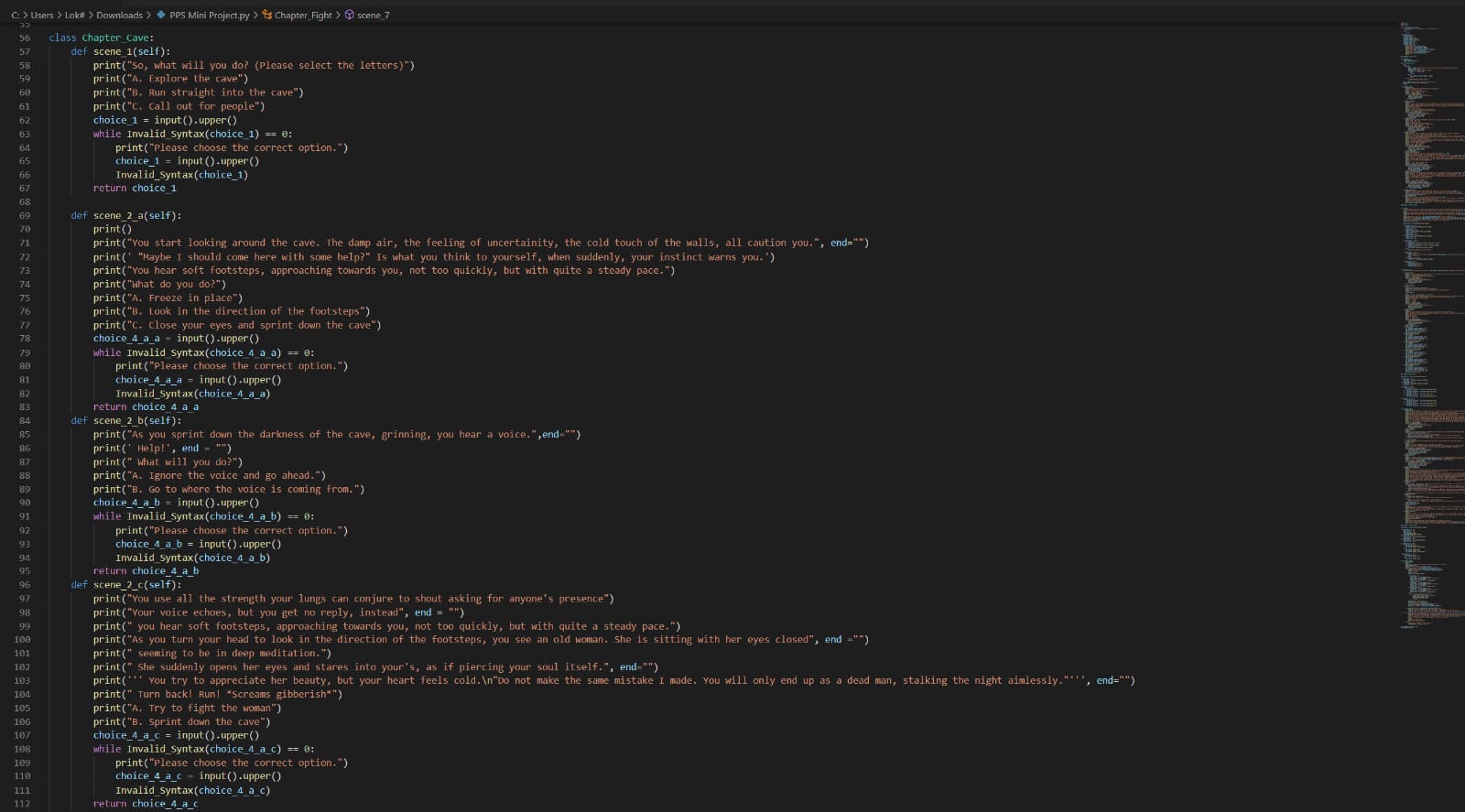
sys.exit("Better Luck Next Time !")

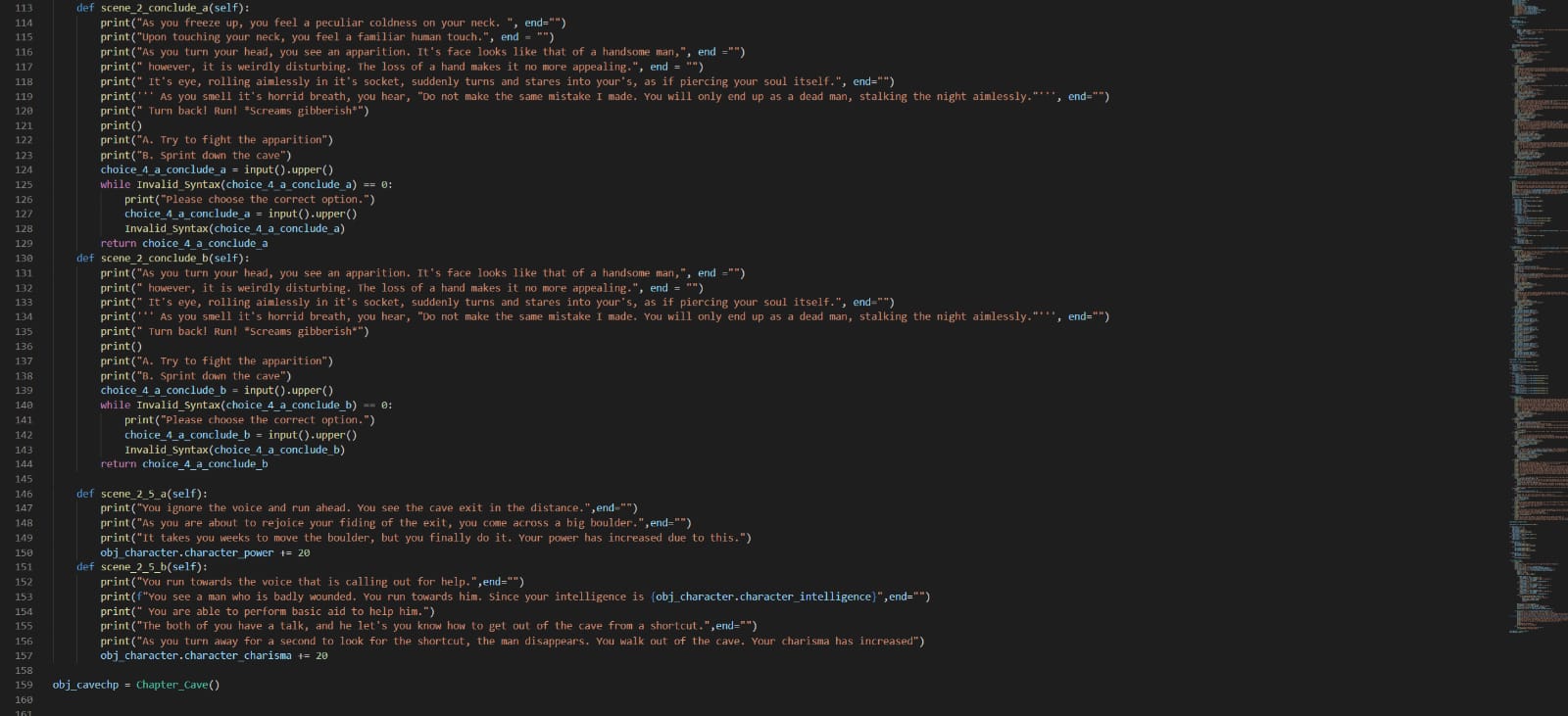
obj\_fightchp = Chapter\_Fight()

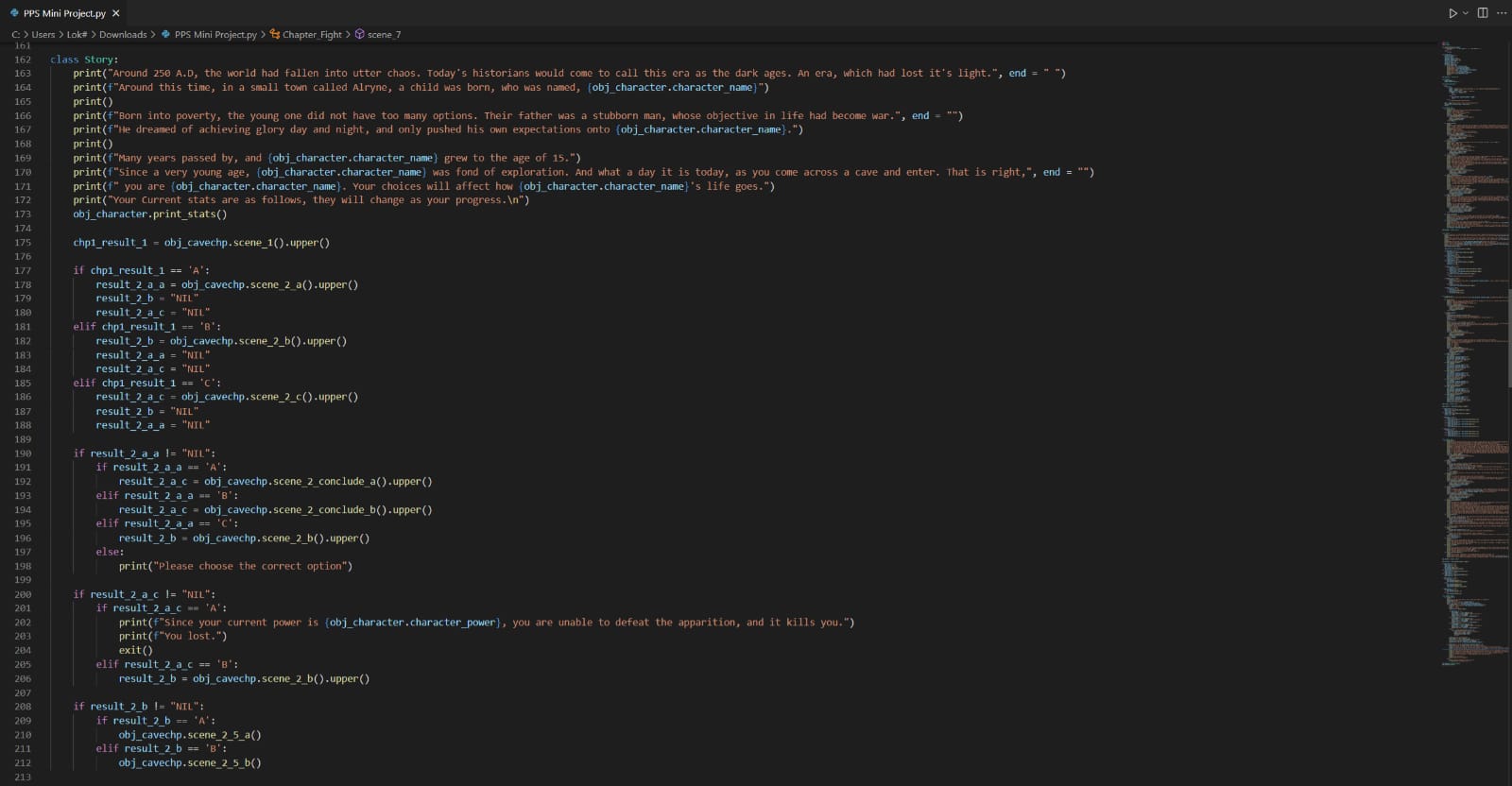
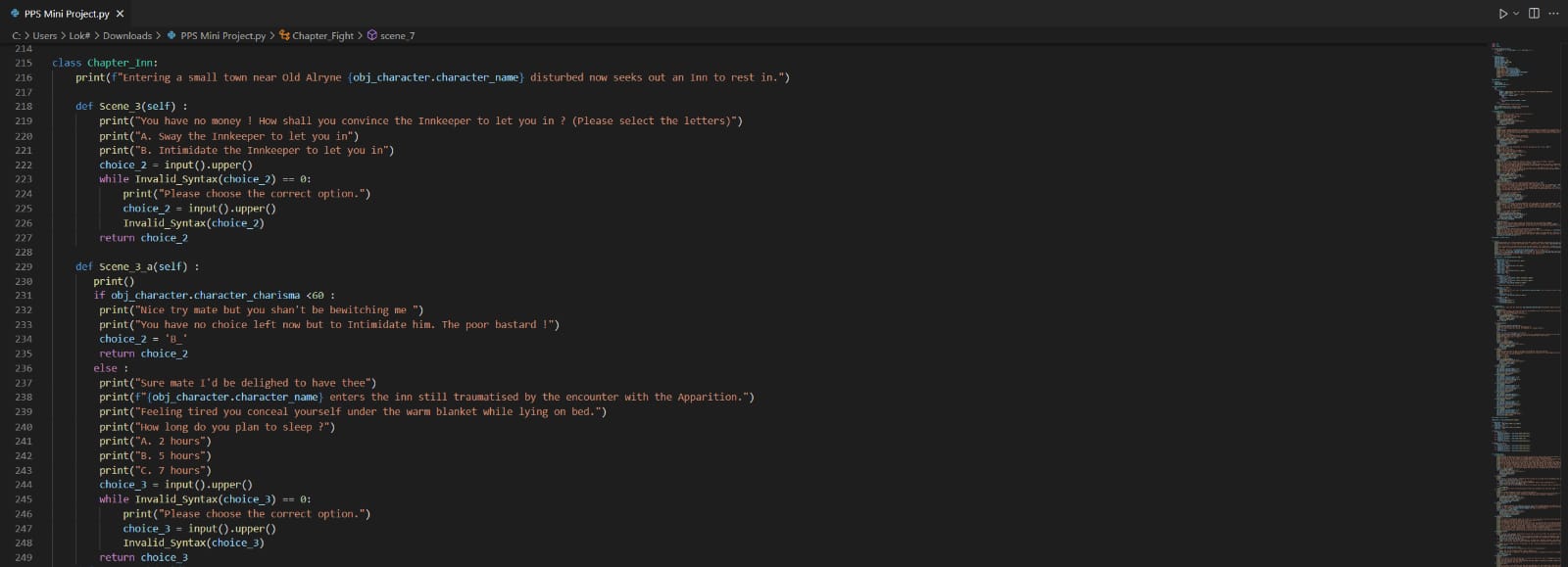
obj\_fightchp.scene\_7()

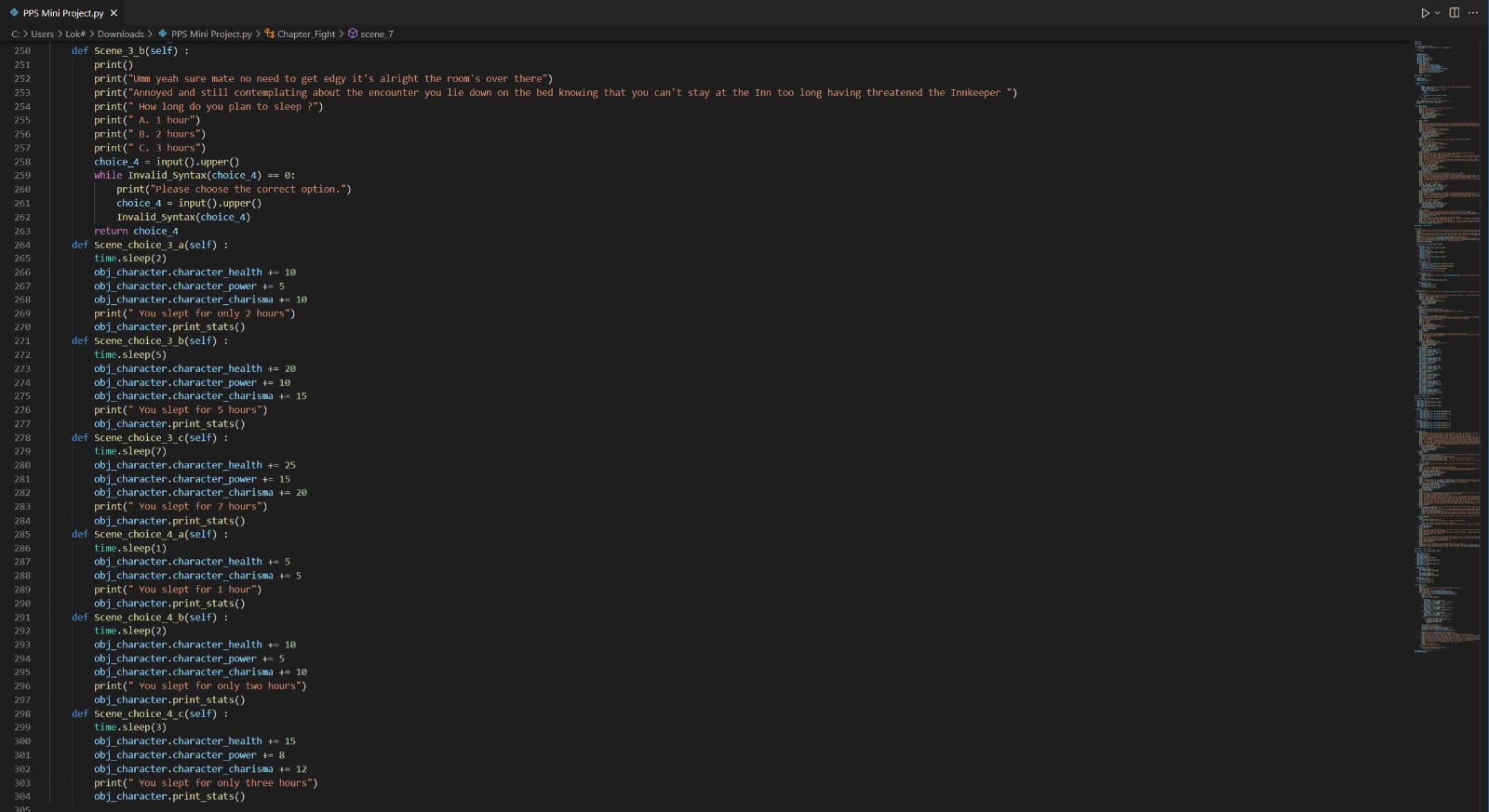
# **SOURCE CODE SCREENSHOTS**

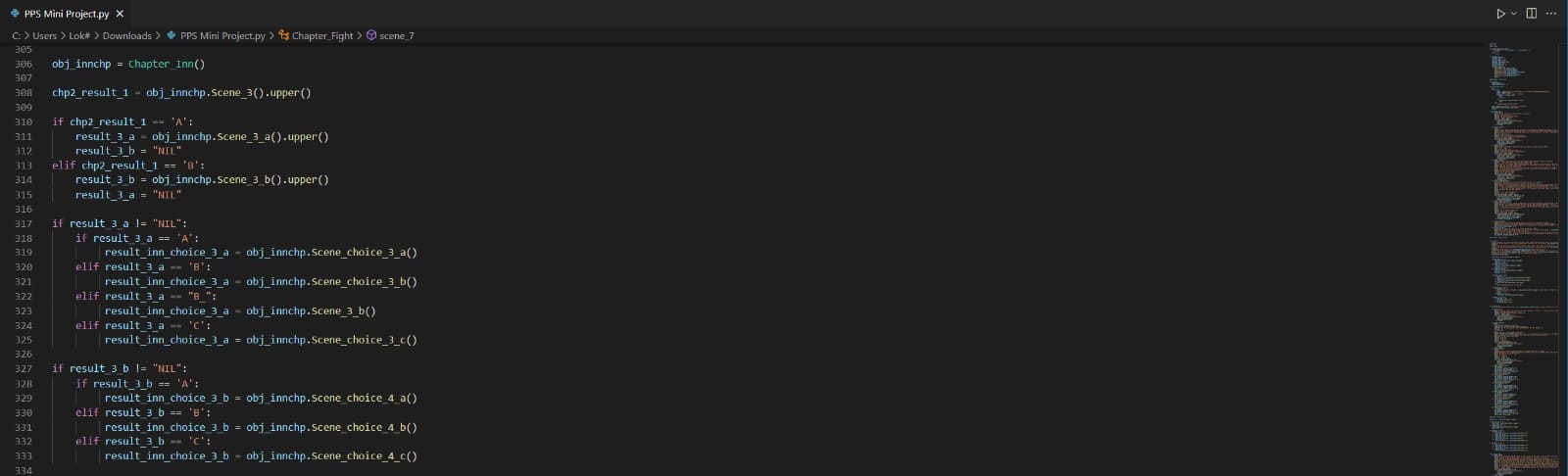


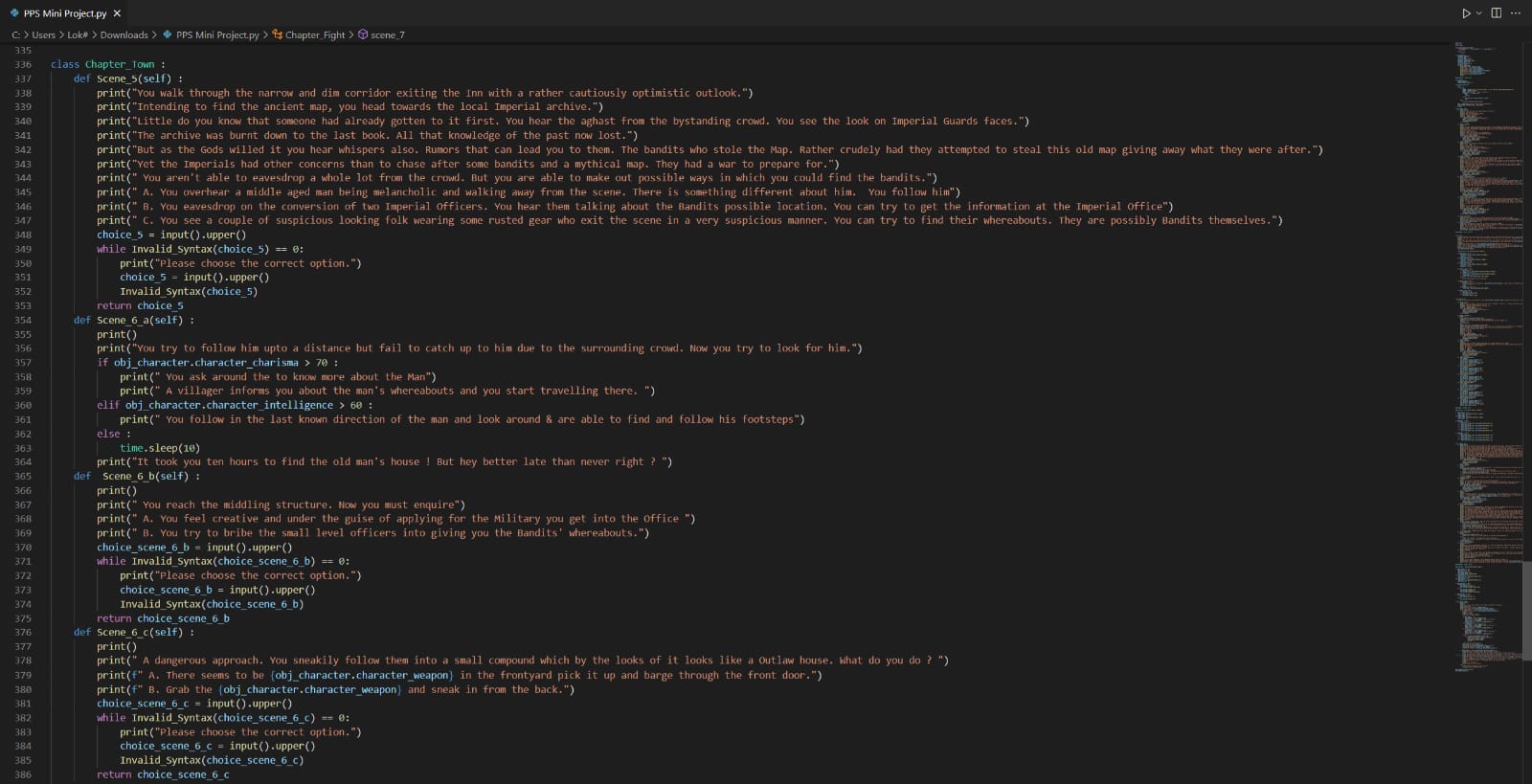


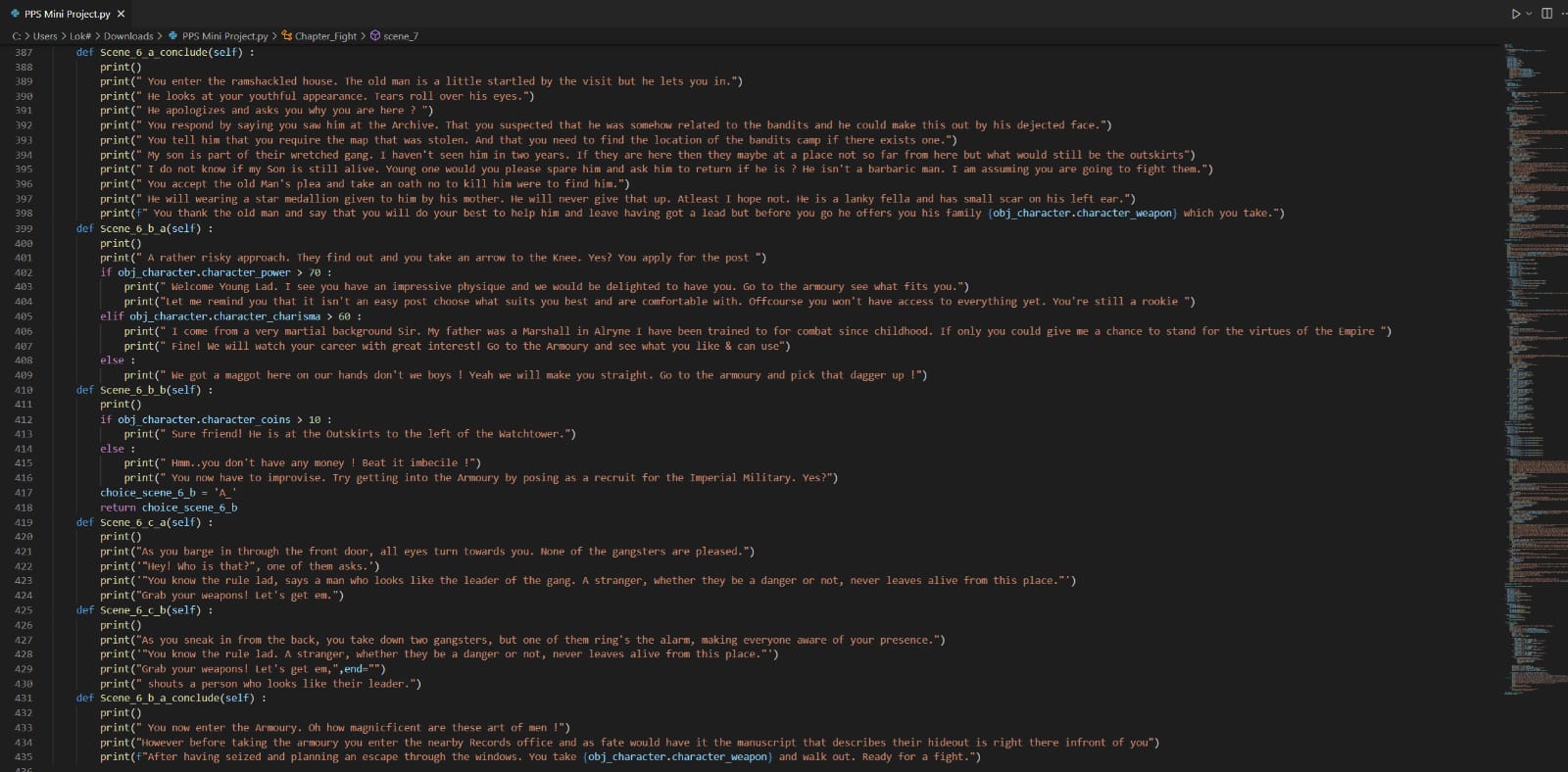


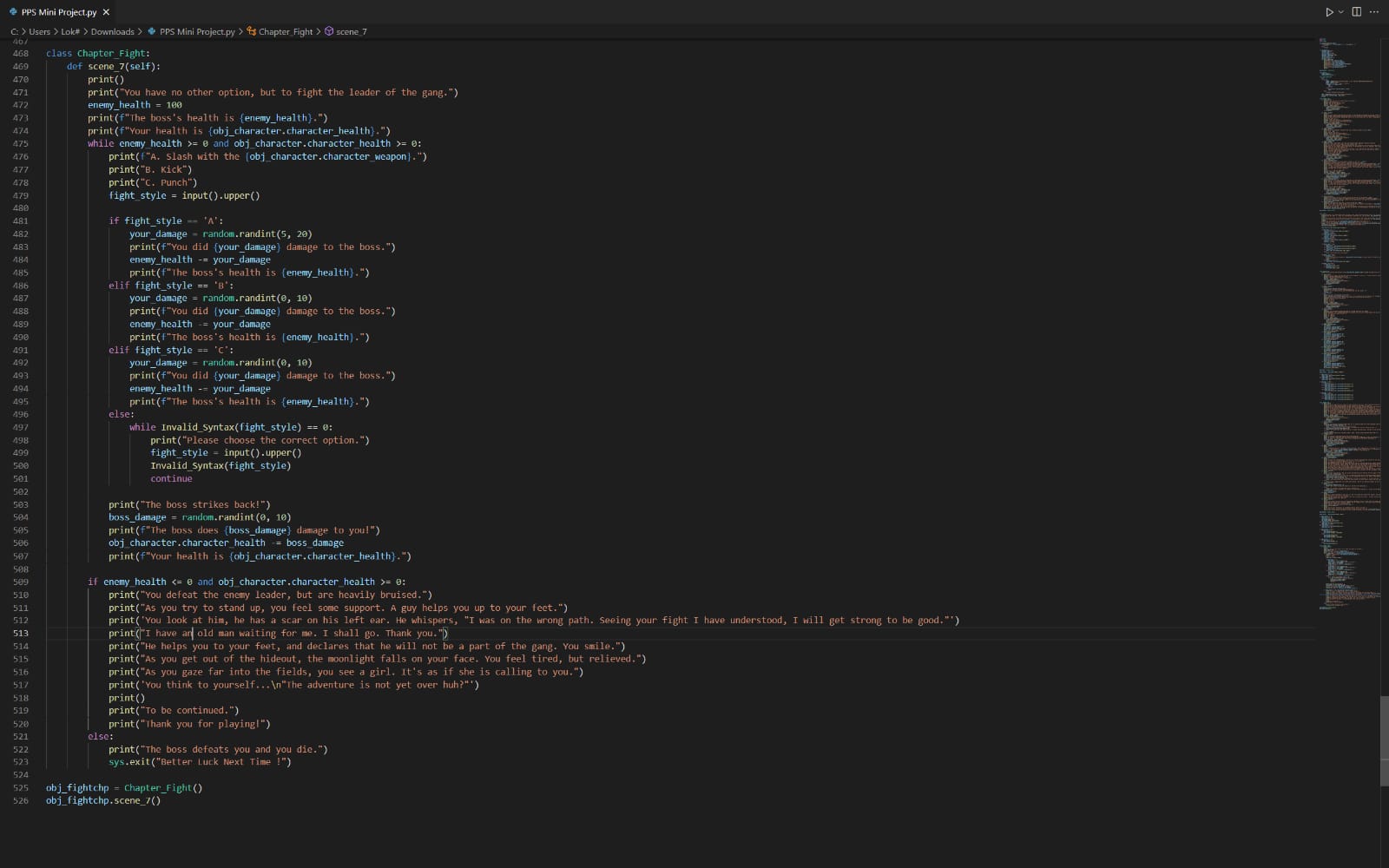












# **OUTPUT SCREENSHOTS**

# 

