

[Return to "Intro to Programming Nanodegree" in the classroom](#)

Adventure Game

REVIEW

CODE REVIEW 6

HISTORY

▼ adventure_game_vamshi_krishna_p.py 6

```
1 import time
2 import random
```

AWESOME

Awesome!! Great work importing (random) class.

```
3
4 # List of items to be used in the game, from line 5 - 130
5 shop_items = ["Health Portion      - Worth 50$",
```

AWESOME

Awesome!! Great work using lists to store your items

- Your list is well organized and sound good.
- Here to learn more about this.
<https://swcarpentry.github.io/python-novice-inflammation/03-lists/>

```
6         "Robes of Freya          - Worth 300$",
7         "Dress of a princess    - Worth 500$",
8         "Glittering steel armour - Worth 400$",
9         "A grand Tribute        - Not for sale",
10        "A Talisman              - Worth 300$",
11        "Jewellery               - Worth 400$",
12        "Durandal,a mighty sword - Worth 600$",
13        "Steel Helmet            - Worth 200$",
14        "Go back to village      - None      "]
15
16 player_items = ["trusty dagger",
17                 "Steel armour"]
18
19 gift_items = ["Leave"]
20
21 princess_items = []
22
23 bar_info_day = ["INFO: Don't go into the forest un-prepared",
24                "INFO: King of this country like presents",
25                "INFO: Check out the contests at night too",
26                "INFO: The cave is nest to stronger enemies",
27                "INFO: Forest is vast, explore out of options to "
28                "find many things",
29                "INFO: Did you know there is a legendary tomb in the forest?",
30                "INFO: Animals in forest are faster than you, you got "
31                "my point?"]
32
33 bar_info_night = ["INFO: Forest at night is dark. You may need fire source",
34                  "INFO: Did you know that the forest has a range "
35                  "between 1 to 50",
36                  "INFO: The princess of this country, likes flowers",
37                  "INFO: check out the secreat contests at night",
38                  "INFO: Do not pick a fight with enemies stronger than you",
39                  "INFO: The village next to us has been attacked by "
40                  "Werewolves",
41                  "INFO: Access to Fort is restricted at night",
42                  "INFO: Sneaking into fort at night time might "
43                  "result in imprisonment",
44                  "INFO: Having full health is always an advantage"]
45
46 village_locations = ["Traven",
47                      "Inn",
48                      "Contest grounds",
49                      "Fort",
50                      "Cave",
51                      "Forest"]
52
53 fort_locations_day = ["Library",
54                       "Quaters",
```

```

55         "Garden",
56         "Central Hall",
57         "Go back to Village"]
58
59 fort_locations_night = ["Library",
60                         "Quaters",
61                         "Garden",
62                         "Central Hall"]
63
64 contest_items = ["Unknown Key",
65                 "Mystery Box"]
66
67 rewards_day = ["Special Prize: Unknown Key",
68               "Deposit Gold and win back Double",
69               "Deposit 10 Gold, Win back 50 Gold"]
70
71 rewards_night = ["Special Prize: Mystery Box",
72                 "Deposit Gold and win back Double",
73                 "Deposit no Gold, Win 50 back Gold"]
74
75 player_offense_moves = ["You step forward to slash the enemy",
76                         "You rise high to deliver a mighty blow",
77                         "You make a lunge and slash attack",
78                         "You deliver an counter attack",
79                         "A power blow was directed towards the enemy",
80                         "You rain down a series of short swings",
81                         "A sharp delivery to penetrate the defense of enemy",
82                         "You run towards the enemy at great speed",
83                         "You duck down and deliver a low swing",
84                         "You stand tall and attck the enemy from the front",
85                         "You run around and penetrate the back of the enemy"]
86
87 player_defense_moves = ["You hold the sword to counter the enemy attck",
88                         "The enemy attacks are repelled by you",
89                         "The enemy jumps on you and you roll to the side",
90                         "You swing your sword in rythm to the enemy attacks",
91                         "You step back away from the enemy",
92                         "You dodge the enemy sword and ready to make an attack"
93                         ]
94
95 enemy_offense_moves = ["The enemy charges at you like a mad bull",
96                       "The enemy has a arm streangth of a boulder",
97                       "Enemy has wits and great strategy to corner you",
98                       "The enemy attacks from behind",
99                       "The enemy rolls around and attacks from the side",
100                      "The enemy jumps on you",
101                      "The enemy fights you head on"]
102
103 enemy_defense_moves = ["Enemy has a strong defense",
104                       "Your enemy has brains to dodge your attacks",
105                       "Your enemy has speed advantage to evade your attacks",
106                       "Your enemy has a strength to with stand your attack",
107                       "Enemy is determined to take you head on",
108                       "Your enemy is a formidable opponent"]
109
110 animal_offense_moves = ["The animal charges straight at you ",
111                        "The animal attacks you with ferocity ",
112                        "The animal attacks you from behind",
113                        "The animal tries to bite you",
114                        "The animal jumps on you",
115                        "The animal uses its claws"]
116
117 animal_defense_moves = ["The animal has a thick skin",
118                        "The animal has brains to dodge your attacks",
119                        "The animal has speed advantage to evade your attacks",
120                        "The animal deflects your attacks with claws",
121                        "The animal steps back to counter"]
122
123 animal_list = ["Boar", "Bear", "Wolf"]
124
125 strong_animal_list = ["Leopard", "Tiger", "Werewolf"]
126
127 monster_list = ["BARTENDER",
128                "MINISTER",
129                "SHOPKEEPER",
130                "HOST of the contest grounds"]
131
132 # triggers and declarations to be used in the code, from line 133 - 181
133 player = ""
134 shop_count = 0
135 freya_dress = "available"
136 princess_dress = "available"
137 armour = "available"
138 durandal = "available"
139 tribute = "available"
140 helmet = "available"
141 bar_count = 0
142 grapes_available = ""
143 player_health = 100
144 gold = 500
145 drink_count = 0
146 info_count = 0
147 time_zone = "day"
148 player_offense_min = 0
149 player_offense_max = 30
150 player_defense = 10
151 contest_day_count = 0
152 contest_night_count = 0
153 game_night_count = 0
154 glowing_blossom_day_count = 0
155 flowers_day_count = 0
156 grapes_day_count = 0
157 keyword_count = 0
158 forest_explore_count = 0

```

```

159 forest_trigger = "deactivate"
160 game_over = "deactivate"
161 grapes_available = ""
162 trade_contract_sequense = ""

```

AWESOME

Your variables are very descriptive. 🙌

```

163 troll = "alive"
164 treasure_box = "not_opened"
165 chest = "not_opened"
166 monster = "alive"
167 huge_monster = "alive"
168 fort_count = 0
169 toll_status = ""
170 inner_fort_count = 0
171 fort_access = "restricted"
172 bell = "deactivated"
173 library_status = ""
174 princess_count = 0
175 princess_introduction = ""
176 name = ""
177 princess_quest_status = ""
178 player_status = ""
179 tomb_password = ""
180 witch_room_count = 0
181 kings_status = ""
182
183
184 # function to print messages with a 2 second delay
185 def print_pause(message):
186     print(message)
187     time.sleep(2)
188
189
190 # function to print messages with a 1 second delay
191 def print_pause1(message):

```

SUGGESTION

Instead of making 2 functions:

```

# function to print messages with a 2 second delay
def print_pause(message):
    print(message)
    time.sleep(2)

# function to print messages with a 1 second delay
def print_pause1(message):
    print(message)
    time.sleep(1)

```

You could make one function that takes another argument to take the delay time that you want:

```

def print_pause1(message, delay):
    print(message)
    time.sleep(delay)

```

```

192     print(message)
193     time.sleep(1)
194
195
196 # function to display the 'items in the list' with a serial number
197 def list_serial(list):
198     for index in range(len(list)):
199         print_pause1(str(index+1) + ". " + list[index])
200     print_pause1("\n")
201     return ""
202
203
204 # function to remove a specific item from a list
205 def remove_item(list, item):
206     new_list = []
207     for index in range(len(list)):
208         if list[index] != item:
209             new_list.append(list[index])
210     list = new_list
211     return list
212
213
214 # function to increase player's health within the limit of 100
215 def health_increase(num):
216     global player_health
217     if player_health < 100 and player_health > 0:
218         player_health += num
219         if player_health > 100:
220             player_health = 100
221     return player_health
222
223
224 # function to decrease player's health within the threshold of 0
225 def health_decrease(health, num):
226     if health > 0:
227         health -= num
228         if health < 0:

```

```

229         health = 0
230         return health
231     else:
232         health = 0
233         return health
234
235 # function to display a message whenever the gold is insufficient
236 def low_gold_message():
237     global gold
238     global player
239     print_pause("NOTE: You do not have enough cash " + player + "!")
240     print_pause("Your Gold : " + gold)
241     print_pause("NOTE: Try earning some in the town contest")
242
243
244 # -----main function to trigger the game-----
245 def adventure_game():
246     global player
247     # input to ask for the player's name
248     player = input("\nWhat is your name?\n")
249     print_pause("\nOn gaurd " + player + ".")
250     print_pause("This is a royal quest to slay the mighty monster")
251     village()
252
253
254 # A branch sequence to the function 'adventure_game'
255 def village():
256     # trigger to end the game from the function 'fight_sequence'
257     if game_over == "activate":
258         print_pause("\nGame over. Try again .....")
259         exit()
260
261     # village intro
262     # player's choice
263     print_pause("\nwelcome to the fabled village of Yorkshire")
264     print_pause("Where would you like to venture into:")
265     # displays list of village locations in a serial list
266     village_choice = input(list_serial(village_locations)).lower()
267     # if player decides to enter into the traven
268     if "traven" in village_choice:
269         print_pause("\nYou have entered Traven to make business")
270         shop()
271     # if player decides to enter into the Inn
272     elif "inn" in village_choice:
273         print_pause("\nYou have checked into an Inn")
274         inn()
275     # if player decides to enter into the contest grounds
276     elif "contest" in village_choice:
277         print_pause("\nYou head towards the contest grounds with curiosity")
278         contest_grounds()
279     # if player decides to enter into the forest
280     elif "forest" in village_choice:
281         forest()
282     # if player decides to enter into the cave
283     elif "cave" in village_choice:
284         print_pause("You step into the darkness of cave courageously !!")
285         print_pause("The cave is a labyrinth of multiple partitions")
286         cave()
287     # if player decides to enter into the fort

```

SUGGESTION

You don't need that, let your code explain itself, and it explains itself very well.

```

288     elif "fort" in village_choice:
289         fort()
290     # if player decides to enter into the secret passage
291     elif "secret" in village_choice and "Secret Passage" in village_locations:
292         secret_passage()
293     # condition to deal with unrecognized input
294     else:
295         print_pause("The place you are looking for does not exist "
296                     "in this village")
297         village()
298
299
300 # funtion to trigger shop sequence in the game
301 def shop():
302     global gold
303     global shop_items
304     global player_items
305     global player
306     global player_status
307     global player_offense_min
308     global player_offense_max
309     global player_defense
310     global player_health
311     global trade_contract_sequense
312     global freya_dress
313     global princess_dress
314     global armour
315     global durandal
316     global tribute
317     global helmet
318     global gift_items
319     global shop_count
320     shop_count += 1
321     # An introduction to be used only one time in the game
322     if shop_count == 1:
323         print_pause("SHOPKEEPER: Haah.., a fine warrior.")
324         print_pause("SHOPKEEPER: Welcome to the Traven of trade and purchase.")

```

```

325     print_pause("SHOPKEEPER: We have a large collection from far lands "
326                 "of Euthopia.")
327     print_pause("SHOPKEEPER: Best and most rarest of all, if you can "
328                 "afford that is!")
329     print_pause("SHOPKEEPER: Feel free to browse through them.")
330     # A message to be displayed only if the player is wanted
331     elif player_status == "wanted":
332         print_pause("SHOPKEEPER: I am not allowed to trade with "
333                     "criminals " + player + ".")
334         print_pause("SHOPKEEPER: But I am willing to keep this as a "
335                     "secret between us")
336     # A message to be used throughout the game, if player revisit the shop
337     else:
338         print_pause("\nSHOPKEEPER: Welcome back " + player + " !!")
339         print_pause("SHOPKEEPER: Back for more?")
340         print_pause("SHOPKEEPER: Let me see if I can find, what you need.")
341
342     print_pause("SHOPKEEPER: What do you want to purchase? Name it.")
343     # Displays shops items in a serial list and take desired input from player
344     shop_choice = input(list_serial(shop_items)).lower()
345     # if player wants to buy health potion in the shop
346     if "health" in shop_choice:
347         if gold >= 50 and player_health < 100:
348             print_pause("SHOPKEEPER: One must stock up on health or face "
349                         "certain death")
350             print_pause("Gold: " + str(gold) + ", - 50")
351             gold -= 50
352             print_pause("Your Gold: " + str(gold))
353             health_increased(25)
354             print_pause("Your health increased +25, feel stronger now ?")
355             print_pause("Your Health: " + str(player_health))
356         elif player_health == 100:
357             print_pause("SHOPKEEPER: You already have full health")
358         elif gold < 50:
359             low_gold_message()
360     # if player wants to buy 'freya dress' in the shop
361     # by adding /freya_dress == "available"/ in the below statement-
362     # player cannot buy this item, if the item is already sold out
363     elif "freya" in shop_choice and freya_dress == "available":
364         if gold >= 300:
365             print_pause("SHOPKEEPER: A rare robe, legend has it's "
366                         "been worn by Goddess freya herself !!\n")
367             print_pause("Note: Now you have 'Robes of Freya' in your "
368                         "possession")
369             shop_items = remove_item(shop_items,
370                                     "Robes of Freya" - Worth 300$")
371             player_items.append("Robes of Freya")
372             gift_items.append("Robes of Freya")
373             freya_dress = "not available"
374             print_pause("Gold: " + str(gold) + ", - 300")
375             gold -= 300
376             print_pause("Your Gold: " + str(gold))
377         else:
378             low_gold_message()
379     # if player wants to buy 'princess dress' in the shop
380     # by adding /princess_dress == "available"/ in below statement-
381     # player cannot buy this item, if the item is already sold out
382     elif "princess" in shop_choice and princess_dress == "available":
383         if gold >= 500:
384             print_pause("SHOPKEEPER: A fine dress! " + player)
385             print_pause("SHOPKEEPER: I guess it's not for you to wear. "
386                         "Hmm., a gift perhaps!!")
387             print_pause("Note: Now you have 'Dress of a princess' in "
388                         "your possession\n")
389             shop_items = remove_item(shop_items,
390                                     "Dress of a princess" - Worth 500$")
391             player_items.append("Dress of a princess")
392             gift_items.append("Dress of a princess")
393             princess_dress = "not available"
394             print_pause("Gold: " + str(gold) + ", - 500")
395             gold -= 500
396             print_pause("Your Gold: " + str(gold))
397         else:
398             low_gold_message()
399     # if player wants to buy 'steel armour' in the shop
400     # by adding / armour == "available" / in the below statement-
401     # player cannot buy this item, if the item is already sold out
402     elif "armour" in shop_choice and armour == "available":
403         if gold >= 400:
404             print_pause("SHOPKEEPER: Shiny shiny, makes enemies whiny tiny")
405             print_pause("Note: Now you have 'Steel Armour' in your possession")
406             shop_items = remove_item(shop_items,
407                                     "Glittering steel armour - Worth 400$")
408             player_items.append("Glittering steel armour")
409             gift_items.append("Glittering steel armour")
410             armour = "not available"
411             print_pause("Gold: " + str(gold) + ", - 400")
412             gold -= 400
413             print_pause("Your Gold: " + str(gold))
414         else:
415             low_gold_message()
416     # if player wants to buy 'talisman' in the shop
417     elif "talisman" in shop_choice:
418         if gold >= 300:
419             print_pause("SHOPKEEPER: I see you are in serious "
420                         "business " + player)
421             print_pause("SHOPKEEPER: Good luck. You are gonna need it.")
422             print_pause("Note: You have the talisman, "
423                         "it adds fire element to weapons")
424             shop_items = remove_item(shop_items,
425                                     "A Talisman" - Worth 300$")
426             player_items.append("Talisman")
427             print_pause("Gold: " + str(gold) + ", - 300")
428             gold -= 300

```

```

428     print_pause("Your Gold: " + str(gold))
429 else:
430     low_gold_message()
431 # if player wants to buy 'jewellery' in the shop
432 elif "jewellery" in shop_choice:
433     if gold >= 400:
434         print_pause("SHOPKEEPER: You have a keen eye in arts" + player)
435         print_pause("Note: You have the Jewellery in your possession")
436         shop_items = remove_item(shop_items,
437                                 "Jewellery" - Worth 400$")
438         player_items.append("Jewellery")
439         gift_items.append("Jewellery")
440         print_pause("Gold: " + str(gold) + ", - 400")
441         gold -= 400
442         print_pause("Your Gold: " + str(gold))
443     else:
444         low_gold_message()
445 # if player wants to buy 'durandal' in the shop
446 # by adding / durandal == "available" / in the below statement-
447 # player cannot buy this item, if the item is already sold out
448 elif "durandal" in shop_choice and durandal == "available":
449     if gold >= 600 and "Excaliber" not in player_items:
450         print_pause("SHOPKEEPER: A great sword for the true warrior")
451         print_pause("SHOPKEEPER: It is said to bring out the inner "
452                     "streth of the warrior from the depths of volcano")
453         print_pause("NOTE: You now possess the 'Durandal', a mighty sword")
454         player_offense_min = 20
455         player_offense_max = 50
456         print_pause("NOTE: Your offense has increased "
457                     "to: " + str(player_offense_max))
458         print_pause("NOTE: You are more stronger now and")
459         print_pause("    ready to fight stronger enemies.")
460         shop_items = remove_item(shop_items,
461                                 "Durandal,a mighty sword - Worth 600$")
462         player_items.append("Durandal")
463         durandal = "not available"
464         print_pause("Gold: " + str(gold) + ", - 600")
465         gold -= 600
466         print_pause("Your Gold: " + str(gold))
467     elif "Excaliber" in player_items:
468         print_pause("SHOPKEEPER: You already possess a more "
469                     "powerful sword")
470         print_pause("SHOPKEEPER: These is no need to buy this sword")
471     elif gold < 600:
472         low_gold_message()
473 # if player wants to buy 'tribute' in the shop
474 # by adding / tribute == "available" / in the below statement-
475 # player cannot buy this item, if the item is already sold out
476 elif "tribute" in shop_choice and tribute == "available":
477     # if the player has 'trade contract' in his possession
478     if "Trade Contract" in player_items:
479         print_pause("SHOPKEEPER: You found my lost 'Trade contract' !!")
480         print_pause("SHOPKEEPER: You have my thanks " + player + ".")
481         print_pause("SHOPKEEPER: As agreed, I will trade you the "
482                     "'Grand Tribute' for the 'Trade contract' of mine.")
483         print_pause("Note: You have the 'Grand Tribute' - "
484                     "a present worthy of kings, in your possession.")
485         shop_items = remove_item(shop_items,
486                                 "A grand Tribute" - Not for sale")
487         player_items = remove_item(player_items, "Trade Contract")
488         player_items.append("A Grand Tribute")
489         tribute = "not available"
490 # if the player does not have trade contract in his possession
491 else:
492     print_pause("SHOPKEEPER: Not for sale " + player + "!")
493     print_pause("SHOPKEEPER: But I'm willing to trade it ")
494     print_pause("    for a personal artifact of mine")
495     print_pause("SHOPKEEPER: I lost my trade contract, somewhere "
496                 "in the town.")
497     print_pause("SHOPKEEPER: It is the pact made by the king himself")
498     print_pause("SHOPKEEPER: Find it and the Tribute is yours")
499     trade_contract_sequence = "initiated"
500 # if player wants to buy 'steel helmet' in the shop
501 # by adding / helmet == "available" / in the below statement-
502 # player cannot buy this item, if the item is already sold out
503 elif "helmet" in shop_choice and helmet == "available":
504     if gold >= 200:
505         print_pause("SHOPKEEPER: Good choice, better safe than sorry.")
506         print_pause("NOTE: You have acquired a 'Steel Helmet'")
507         shop_items = remove_item(shop_items,
508                                 "Steel Helmet" - Worth 200$")
509         player_items.append("Steel Helmet")
510         helmet = "not available"
511         print_pause("Gold: " + str(gold) + ", - 200")
512         gold -= 200
513         print_pause("Your Gold: " + str(gold) + "\n")
514         print_pause("NOTE: Your defense has been increased")
515         print_pause("Defense : " + str(player_defense) + ", + 10")
516         player_defense += 10
517         print_pause("Player Defense increased to: " + str(player_defense))
518     elif gold < 200:
519         low_gold_message()
520 # if player wants to get back to the village
521 elif "village" in shop_choice:
522     print_pause("You head back to the village")
523     village()
524 # if player gives an unrecognized input
525 else:
526     print_pause("SHOPKEEPER: I don't understand, "
527                 "Specify the name of the item clearly")
528     shop()
529 # After trading, player has to get back to the village
530 print_pause("\nYou took care of the business in Traven, as of now "
531            "and head back to the village")

```

```

532     village()
533
534
535
536 # a function to run 'inn' sequence
537 def inn():
538     print_pause("This is a place to rest & refresh your body and soul")
539     inner_inn()
540
541
542 # A branch sequence to the funtion 'inn'
543 def inner_inn():
544     # player's choice
545     print_pause("\nWhat would you like to do:")
546     inn_choice = input("1. Go to the Bar"
547                        "\n2. Rent a Room"
548                        "\n3. Village\n").lower()
549     # if player chooses to enter the bar
550     if "bar" in inn_choice:
551         bar()
552     # if player chooses to enter the room
553     elif "room" in inn_choice:
554         room()
555     # if player chooses to enter the village
556     elif "village" in inn_choice:
557         village()
558     # condition to deal with unrecognized input
559     else:
560         print_pause("Enter a valid choice: Bar? or Room? or village?")
561         inner_inn()
562
563
564 # A branch sequence to the funtion 'inner_inn'
565 def bar():
566     global bar_count
567     global info_count
568     global player_items
569     global grapes_available
570     global gold
571     global time_zone
572     print_pause("")
573     print_pause("BARTENDER: What do you like to have?")
574     bar_choice = input("1. Have a drink "
575                       "\n2. Information\n").lower()
576     if "drink" in bar_choice:
577         bar_count += 1
578         # if player has sufficient money and timezone is day, he gets drunk
579         if gold >= 10 and time_zone == "day":
580             if bar_count == 1:
581                 print_pause("BARTENDER: Here's the drink you ordered")
582                 # wine/grapes in bartender's possession will be diminished
583                 grapes_available = "no"
584                 drunk()
585             elif bar_count > 1 and bar_count < 5:
586                 # if you have grapes and bartender doesn't, you get free drink
587                 if "Grapes" in player_items and grapes_available == "no":
588                     print_pause("\nNOTE: Grapes are given to the Bartender")
589                     print_pause("BARTENDER: As promised next round is on "
590                               "the house")
591                     player_items = remove_item(player_items, "Grapes")
592                     # grapes in your possession will be transferred to bartender
593                     grapes_available = "yes"
594                     drunk()
595                 elif grapes_available == "yes":
596                     # if bartender has grapes, you get drunk
597                     drunk()
598                 else:
599                     # if both player and bartender don't have grapes
600                     # he asks you to go fetch some from the forest
601                     print_pause("BARTENDER: My apologies, we are out of wine.")
602                     print_pause("BARTENDER: If you are desperate, "
603                               "you can run an errand for me.")
604                     print_pause("BARTENDER: There are some wild grapes "
605                               "growing in the forest")
606                     print_pause("BARTENDER: It is dangerous for me to venture "
607                               "into forest by myself")
608                     print_pause("BARTENDER: Fetch some for me and next round "
609                               "will be on the house")
610                     # After every 5 counts, grapes in bartender's possession will be-
611                     # diminished and count is reset to 0
612                     if bar_count >= 5:
613                         print_pause("BARTENDER: We are out of wine again!!")
614                         print_pause("BARTENDER: I'm afraid you gonna have to "
615                                   "bring some more from the forest")
616                         grapes_available = "no"
617                         bar_count = 0
618                     # at night time, drinks are not served
619                     elif time_zone == "night":
620                         print_pause("Bartender is not serving drinks at night")
621                         print_pause("But you can hear extra gossips from him at night.")
622                     # if player does not have enough cash
623                     elif gold < 10:
624                         print_pause("Bartender: I'm afraid you don't have enough gold")
625                         print_pause("BARTENDER: Try to earn some gold in contest grounds")
626                 # if player wants info from the bartender
627                 elif "info" in bar_choice:
628                     info_count += 1
629                     # after getting info for more than 2 times,
630                     # player is sent back to the room to take rest
631                     if info_count > 2:
632                         info_count = 0
633                         print_pause("BARTENDER: Don't exert yourself, come back later")
634                         print_pause("BARTENDER: I suggest you take a room and "
635                                   "rest for the day")
636                     room()

```

```

638     # random info is displayed during daytime
639     elif time_zone == "day":
640         print_pause(random.choice(bar_info_day))
641         bar()
642     # random info different from daytime is displayed during night time
643     elif time_zone == "night":
644         print_pause(random.choice(bar_info_night))
645         bar()
646     # code to deal with unrecognized input
647     else:
648         print_pause("BARTENDER: I'm sorry, I don't understand your language")
649         bar()
650     inner_inn()
651
652 # A branch sequence to the function 'bar'
653 def drunk():
654     global gold
655     print_pause("You get drunk again")
656     print_pause("Gold: " + str(gold) + ", -10")
657     gold -= 10
658     print_pause("Your Gold: " + str(gold))
659     print_pause("You have no choice but to take a room\n")
660     room()
661
662 # A branch sequence to the function 'inner_inn'
663 def room():
664     global gold
665     print_pause("\nYou entered the room all tired and deprived of sleep")
666     print_pause("You were charged 20 gold for the room")
667     print_pause("Gold: " + str(gold) + ", -20")
668     gold -= 20
669     print_pause("Your Gold: " + str(gold))
670     room_sequence()
671
672 # A branch sequence to the function 'room'
673 def room_sequence():
674     global player_items
675     global trade_contract_sequence
676     print_pause("\nWhat would you do now?")
677     room_action = input("1. Crash on the bed\n2. Examine the room\n").lower()
678     # if player wants to sleep on bed
679     if "bed" in room_action:
680         print_pause("\nIt is time to rest your body")
681         room_choice()
682     # if player wants to examine the room
683     elif "examine" in room_action:
684         print_pause("\nYou examine the room thoroughly for any "
685                     "possible anomaly")
686         # if trade contract sequence initiated by the shopkeeper
687         if trade_contract_sequence == "initiated":
688             print_pause("You find an old article eating dust behind the "
689                         "closet")
690             print_pause("You read it, out of curiosity")
691             print_pause("You have found some sort of 'Trade Contract'")
692             player_items.append("Trade Contract")
693             print_pause("\nIt is time to rest your body")
694             trade_contract_sequence = "end"
695             room_choice()
696         # if trade contract sequence initiated by the shopkeeper
697         else:
698             print_pause("You find nothing abnormal")
699             print_pause("It is time to rest your body")
700             room_choice()
701     # condition to deal with unrecognized input
702     else:
703         print_pause("Enter a valid action in words")
704         room_sequence()
705
706 # A branch sequence to the function 'room_sequence'
707 def room_choice():
708     global time_zone
709     global flowers_day_count
710     global grapes_day_count
711     global glowing_blossom_day_count
712     global keyword_count
713     global forest_trigger
714     global player_items
715     global gift_items
716     global bell
717
718     print_pause("\nHow long do you need to sleep?")
719     sleep_till = input("1. Wake up at night\n2. Wake up at morning\n").lower()
720     # if player wants to sleep till night and timezone is day
721     if "night" in sleep_till and time_zone == "day":
722         print_pause("You wake up at night")
723         time_zone = "night"
724         # increase player's health by 25 after some rest
725         print_pause("Your health increased by 25")
726         print_pause("Health : " + str(player_health) + ", + 25")
727         health_increase(25)
728         print_pause("Your Health: " + str(player_health))
729         print_pause("\nsome events are different at night, go check them out")
730         print_pause("You come down to the inn entrance")
731         inn()
732     # if player wants to sleep till night and timezone is already night
733     if "night" in sleep_till and time_zone == "night":
734         print_pause("It's already night time")
735         room_choice()

```



```

740 # if player wants to sleep till morning of next day
741 elif "morning" in sleep_till:
742     keyword_count = 0
743     forest_trigger = "deactivate"
744     flowers_day_count += 1
745     grapes_day_count += 1
746     bell = "deactivated"
747     # if player has flowers in his possession for more than 2 days
748     if flowers_day_count > 2:
749         print_pause("\nNOTE: Flowers in your possession are dried up.")
750         print_pause("NOTE: They are no longer useful")
751         player_items = remove_item(player_items, "Flowers")
752         gift_items = remove_item(player_items, "Flowers")
753     # if player has glowing blossom in his possession for more than 2 days
754     if glowing_blossom_day_count > 2:
755         print_pause("\nNOTE: Glowing Blossom in your possession "
756                     "are dried up.")
757         print_pause("NOTE: They are no longer useful")
758         player_items = remove_item(player_items, "Glowing Blossom")
759         gift_items = remove_item(player_items, "Glowing Blossom")
760     # if player has grapes in his possession for more than 2 days
761     if grapes_day_count > 2:
762         print_pause("\nNOTE: Grapes in your possession have gone bad")
763         print_pause("NOTE: They are no longer useful.")
764         player_items = remove_item(player_items, "Grapes")
765         gift_items = remove_item(player_items, "Grapes")
766     print_pause("\nYou wake Up in the morning fresh to resume "
767               "your journey")
768     time_zone = "day"
769     # increse player's health by 50 after plenty of rest
770     print_pause("Your health increased by 50")
771     print_pause("Health : " + str(player_health) + ", + 50")
772     health_increase(50)
773     print_pause("Your Health: " + str(player_health))
774     print_pause("\nYou head back to the Inn entrance")
775     inn()
776 # condition to deal with unrecognized input
777 else:
778     print_pause("Enter a valid time zone. Night? or Morning?")
779     room_choice()
780
781
782 # funtion to trigger the location 'contest grounds' in the village
783 def contest_grounds():
784     global time_zone
785     global contest_day_count
786     global contest_night_count
787     # if player visits 'contest grounds' during daytime
788     if time_zone == "day":
789         contest_day_count += 1
790         # intro to be used only once in the game
791         if contest_day_count == 1:
792             print_pause("surprisingly, this country holds contests "
793                       "in gambling")
794             print_pause("People of this country prefers peace over voilance")
795         # if player visits more than once
796         if contest_day_count >= 1:
797             print_pause("Would you like to take part in the gambling?")
798             contest_choice_day = input("1. Yes\n2. No\n").lower()
799             if "yes" in contest_choice_day:
800                 game_day()
801             elif "no" in contest_choice_day:
802                 print_pause("HOST: Farewell, stranger.")
803                 print_pause("HOST: Do come back another time")
804                 village()
805             # condition to handle unrecognized inputs from the player
806             else:
807                 print_pause("HOST: Make up your mind stranger. 'yes' "
808                           "or 'no' ?")
809                 contest_grounds()
810     # if player visits 'contest grounds' during night time
811     elif time_zone == "night":
812         contest_night_count += 1
813         # intro to be used only once in the game
814         if contest_night_count == 1:
815             print_pause("Enen thought voilance is prohibited in face "
816                       "the of public")
817             print_pause("Under the clock of shadows, the Arena conducts "
818                       "battles")
819             print_pause("Would you dare to confront the person "
820                       "reponsible for the fights at night time?")
821         # if player visits more than once
822         if contest_night_count > 1:
823             print_pause("Would you dare to take part in the fights?")
824         # take input from the player
825         contest_choice_night = input("1. Yes\n2. No\n").lower()
826         if "yes" in contest_choice_night:
827             game_night()
828         elif "no" in contest_choice_night:
829             print_pause("Perhaps you are not ready yet")
830             print_pause("Do come back another time...")
831             village()
832         # condition to handle unrecognized inputs from the player
833         else:
834             print_pause("HOST: Make up your mind stranger. 'yes' or 'no' ?")
835             contest_grounds()
836
837 # A branch sequence to the funtion 'contest_grounds'
838 def game_day():
839     global gold
840     global contest_items

```

```

845 global rewards_day
846 global contest_items
847 global player_items
848 global player_status
849 # a trigger if the player is wanted
850 if player_status == "wanted":
851     print_pause("Leave stranger, you are gonna get me in trouble")
852     print_pause("You are forced to leave the contest grounds")
853     print_pause("You head back to the village")
854     village()
855 # choice to play different games
856 print_pause("\nHOST: welcome stranger, what would you like to play?")
857 contest_day_choice = input("1. Evens & Odds"
858                             "\n2. Double or nothing"
859                             "\n3. View rewards\n").lower()
860 # if player choose 'evens & odds' game
861 if "even" in contest_day_choice:
862     if gold >= 10:
863         print_pause("HOST: Deposit 10 gold to play the game")
864         print_pause("Your Gold: " + str(gold) + " , -10")
865         gold -= 10
866         print_pause("Your Gold: " + str(gold))
867         value1, value2 = roll_dice()
868         if (value1+value2) % 2 == 0:
869             value = "even"
870         else:
871             value = "odd"
872
873         print_pause(str(value1) + "+" + str(value2) + " = "
874                     + str(value1 + value2) + " , " + value)
875
876         if value == "even":
877             print_pause("HOST: Evens it is..., Congratulations stranger")
878             print_pause("HOST: You have won 50 gold")
879             print_pause("Your Gold: " + str(gold) + " , +50")
880             gold += 50
881             print_pause("Your Gold: " + str(gold))
882         else:
883             print_pause("HOST: Odds are against you stranger.., "
884                         "you lost your deposit")
885     # if player does not have enough gold
886     else:
887         low_gold_message()
888         village()
889 # if player choose 'double or nothing' game
890 elif "double" in contest_day_choice:
891     print_pause("\nHOST: Deposit gold as you desire")
892     print_pause("HOST: If both dices are evens or odd , you win")
893     print_pause("HOST: If you win , you will have double the deposit")
894     print_pause("HOST: Lose, I keep the gold.")
895     # if player has not win the special prize already
896     if "Unknown Key" in contest_items:
897         print_pause("Also I have a special prize waiting to be won.")
898         # take a desired amount of bet from the player's input
899         gold_deposited = int(input("How much would you like to bet:"))
900         print_pause("Your Gold: " + str(gold) + " , - " + str(gold_deposited))
901         gold -= gold_deposited
902         if gold >= 0:
903             print_pause("Your Gold= " + str(gold))
904             value1, value2 = roll_dice()
905             print_pause("Dices : " + str(value1) + " , " + str(value2))
906             # if player wins
907             if (value1 % 2) == (value2 % 2):
908                 # if player has not win the special prize already
909                 if "Unknown Key" in contest_items:
910                     print_pause("HOST: Marvelous stranger!, it's your"
911                                 " lucky day")
912                     print_pause("HOST: I have just the rare item as the prize")
913                     print_pause("Note: You have received an 'Unknown Key'.")
914                     contest_items = remove_item(contest_items, "Unknown Key")
915                     player_items.append("Unknown Key")
916                     rewards_day = remove_item(rewards_day,
917                                             "Special Prize: Unknown Key")
918                     print_pause("Host: I'll keep the deposit just this once")
919                     print_pause("HOST: consider it an even trade for the "
920                                 "rare item")
921                 # if player has already won the special prize
922                 else:
923                     print_pause("HOST: Congratulation, stranger! you won "
924                                 "this game")
925                     gold_deposited = 2*gold_deposited
926                     print_pause("Your Gold: " + str(gold) + " , "
927                                 "+ " + str(gold_deposited))
928                     gold += gold_deposited
929                     print_pause("Your Gold: " + str(gold))
930                 # if player loses
931                 else:
932                     print_pause("Host: It is not as easy as it sounds stranger!!")
933                     print_pause("HOST: I get to keep the gold")
934             # if player does not have enough gold
935             else:
936                 low_gold_message()
937                 gold += gold_deposited
938                 print_pause("Your Gold= " + str(gold))
939                 village()
940         # if player wants to check the rewards before playing the game
941         elif "reward" in contest_day_choice:
942             print_pause(list_serial(rewards_day))
943             game_day()
944         # condition to deal unrecognized input
945         else:
946             print_pause("HOST: Say the correct words stranger")
947             game_day()
948     # choice to play the game again

```

```

948 print_pause("\nHOST: would you like to play again stranger ?")
949 play_again = input("1.yes\n2.no\n").lower()
950 if "yes" in play_again:
951     game_day()
952 # if player wants to leave the contest grounds
953 else:
954     print_pause("HOST: Good bye, stranger")
955     village()
956
957
958
959 # A branch sequence to the funtion 'game_day'
960 def roll_dice():
961     print_pause("\nRolling Dice:")
962     value1 = random.randint(1, 6)
963     print_pause("Dice 1: " + str(value1))
964     value2 = random.randint(1, 6)
965     print_pause("Dice 2: " + str(value2))
966     # returns a random number from each dice
967     return value1, value2
968
969
970 # A branch sequence to the funtion 'contest_grounds'
971 def game_night():
972     global game_night_count
973     game_night_count += 1
974     print_pause("\nHOST: Welcome Stranger")
975     # an intro to be used only once in the game
976     if game_night_count == 1:
977         print_pause("HOST: What brings you here at this hour of time?")
978         input("Respond in your own words:\n")
979         # irrespective of player's input game moves forward
980         print_pause("HOST: I see you knew about our little secret contests")
981         print_pause("HOST: Can you handle the voilance, stranger?")
982         game_forward = input("1. Yes"
983                               "\n2. No\n").lower()
984         if "yes" in game_forward:
985             game_night_contest()
986         elif "no" in game_forward:
987             village()
988         else:
989             print_pause("HOST: Hmm., I don't understand. Better meet later")
990             village()
991     # if player visits more than once
992     elif game_night_count > 1:
993         game_night_contest()
994
995
996 # A branch sequence to the funtion 'game_night'
997 def game_night_contest():
998     global gold
999     global contest_items
1000     global player_items
1001     global player_status
1002     global rewards_day
1003     global rewards_night
1004     # a trigger is player is wanted for a crime
1005     if player_status == "wanted":
1006         print_pause("HOST: I see you got yourself in some trouble stranger !!")
1007         print_pause("HOST: Fear not, I am already operating an illegal "
1008                     "activity myself")
1009         print_pause("HOST: I'm not gonna turn you in, feel free to come at "
1010                     "night time")
1011     # game choice to fight the desired enemy
1012     print_pause("what would you like to to stranger?")
1013     contest_night_choice = input("1. Fight Gladiator\n2. Fight a warrior\n"
1014                                   "3. View rewards\n").lower()
1015     # if player wants to fight gladiator
1016     if "gladiator" in contest_night_choice:
1017         print_pause("\nHOST: Are you sure stranger?")
1018         print_pause("HOST: He is the best fighter around "
1019                     "all the nearby villages")
1020         print_pause("HOST: His strength rivals a chariot")
1021         fight_choice = input("1.yes"
1022                               "\n2.no\n").lower()
1023     # if player chooses to fight the gladiator
1024     if "yes" in fight_choice:
1025         print_pause("\nHOST: I hope you don't regret your choice "
1026                     "stranger!!")
1027         print_pause("HOST: Deposit gold as you desire")
1028         print_pause("HOST: If you win , you will have double the deposit")
1029         print_pause("HOST: Lose, I keep the gold.")
1030         # input to bet desired amount of gold
1031         gold_deposited = int(input("How much would you like to bet:"))
1032         print_pause("Your Gold: " + str(gold) + " , -"
1033                     " " + str(gold_deposited))
1034         gold -= gold_deposited
1035         print_pause("Your Gold= " + str(gold))
1036         # if player does not have enough gold
1037         if gold < 0:
1038             low_gold_message()
1039             print_pause("You are sent back to the village")
1040             village()
1041         # if player has enough gold, game resumes
1042         print_pause("\n The battle begins ..... \n")
1043         fight_result = fight_sequence("gladiator", 20, 50, 20)
1044         # if player wins
1045         if fight_result == "player wins":
1046             print_pause("You have emerged victorious in this battle")
1047             print_pause("The Gladiator kneels down before you\n")
1048             print_pause("HOST: A combat to remember by stranger!!")
1049             # if player has not won the special night prize before
1050             if "Mystery Box" in contest_items:
1051                 print_pause("HOST: You are the first to bring down my "
1052                             "gladiator. stranger!!")

```

```

1052         print_pause("HOST: Let me reward you with the best "
1053                     "prize in my possession")
1054         contest_items = remove_item(contest_items, "Mystery Box")
1055         player_items.append("Mystery Box")
1056         rewards_day = remove_item(rewards_day, "Special Prize"
1057                                 ": Mystery Box")
1058         print_pause("Note: You have received a 'Mystery Box'.")
1059         # if player has won the special night prize already
1060         else:
1061             print_pause("HOST: Congratulation, stranger! you won "
1062                         "this battle")
1063             print_pause("HOST: You earn the right to call yourself"
1064                         "- 'The strongest among the village'")
1065             gold_deposited = 2*gold_deposited
1066             print_pause("Your Gold: " + str(gold) + " , +"
1067                         " " + str(gold_deposited))
1068             gold += gold_deposited
1069             print_pause("Your Gold: " + str(gold))
1070         # if player loses
1071         else:
1072             print_pause("Host: It is no simple task to defeat "
1073                         "the gladiator stranger!!")
1074         # if player chooses to fight the gladiator
1075         elif "no" in fight_choice:
1076             print_pause("HOST: There is no shame in retreat, stranger")
1077             game_night_contest()
1078         # condition to deal unrecognized input
1079         else:
1080             print_pause("HOST: I don't understand your words stranger")
1081             game_night_contest()
1082         # if player wants to fight the warrior
1083         elif "warrior" in contest_night_choice:
1084             # a special situation for the player to raise gold if broken completely
1085             print_pause("\nHOST: You don't have to Deposit any gold to "
1086                         "play the game")
1087             print_pause("HOST: You get 50 gold on win and nothing to lose")
1088             print_pause("Your Gold: " + str(gold))
1089             print_pause("\n The battle begins ..... \n")
1090             fight_result = fight_sequence("warrior", 10, 30, 10)
1091             # if player wins
1092             if fight_result == "player wins":
1093                 print_pause("HOST: Well done stranger!")
1094                 print_pause("HOST: You are tougher than you look")
1095                 print_pause("HOST: Perhaps you might have a chance to "
1096                             "defeat our gladiator")
1097                 print_pause("HOST: You have won 50 gold")
1098                 print_pause("Your Gold: " + str(gold) + " , +50")
1099                 gold += 50
1100                 print_pause("Your Gold: " + str(gold))
1101             # if player loses
1102             else:
1103                 print_pause("HOST: Arena is merciless stranger..., "
1104                             "you lost your deposit")
1105         # if player wants to check the reward before fighting
1106         elif "reward" in contest_night_choice:
1107             print_pause(list_serial(rewards_night))
1108             game_night_contest()
1109         # condition to deal unrecognized input
1110         else:
1111             print_pause("HOST: Say the correct words stranger")
1112             game_night_contest()
1113         # trigger to let the player compete again
1114         print_pause("HOST: would you like to fight again stranger ?")
1115         play_again = input("1.yes\n2.no\n").lower()
1116         if "yes" in play_again:
1117             game_night_contest()
1118         else:
1119             print_pause("HOST: Good bye, stranger")
1120             village()
1121
1122
1123     # A branch sequence to the funtion 'game_night_contest'
1124     def fight_sequence(enemy, enemy_offense_min, enemy_offense_max, enemy_defense):
1125         global player_health
1126         global player_offense_min
1127         global player_offense_max
1128         global player_defense
1129         global player_offense_moves
1130         global player_defense_moves
1131         global enemy_offense_moves
1132         global enemy_defense_moves
1133         global animal_offense_moves
1134         global animal_defense_moves
1135         global rewards_night
1136         global animal_list
1137         global strong_animal_list
1138         global game_over
1139
1140         enemy_health = 100
1141         # a loop to let player and enemy deal random damage in each turn
1142         for turn in range(40):
1143             # if trun is even number and player is alive
1144             if turn % 2 == 0 and player_health > 0:
1145                 print_pause("Player Attacks.....//")
1146                 # print random message from the list 'player_offense_moves'
1147                 print_pause(random.choice(player_offense_moves))
1148             # if enemy is an animal
1149             if enemy in animal_list or enemy in strong_animal_list:
1150                 # print random message from the list 'animal_defense_moves'
1151                 print_pause(random.choice(animal_defense_moves))
1152             # if enemy is not an animal
1153             else:
1154                 # print random message from the list 'enemy_defense_moves'
1155                 print_pause(random.choice(enemy_defense_moves))

```

```

1158     # assign random number with in the range to player's offence
1159     player_offense = random.randint(player_offense_min,
1160                                     player_offense_max)
1161     enemy_damage = enemy_defense - player_offense
1162     # as damage decreases the health, it cannot be a positive number
1163     if enemy_damage > 0:
1164         enemy_damage = 0
1165     print_pause("Enemy damage : " + str(enemy_damage))
1166     # decrement enemy health based on damage
1167     enemy_health = health_decrease(enemy_health, enemy_damage)
1168     # display both player's and enemy's health
1169     print_pause("Player health: " + str(player_health))
1170     print_pause("Enemy health : " + str(enemy_health) + "\n")
1171     # if turn is odd number and enemy is alive
1172     elif turn % 2 == 1 and enemy_health > 0:
1173         print_pause(enemy + " Attacks....//")
1174         # if enemy is an animal
1175         if enemy in animal_list or enemy in strong_animal_list:
1176             # print random message from the list 'animal_offense_moves'
1177             print_pause(random.choice(animal_offense_moves))
1178         # if enemy is not an animal
1179         else:
1180             # print random message from the list 'enemy_offense_moves'
1181             print_pause(random.choice(enemy_offense_moves))
1182             # print random message from the list 'player_defense_moves'
1183             print_pause(random.choice(player_defense_moves))
1184             # assign random number with in the range to enemy's offence
1185             enemy_offense = random.randint(enemy_offense_min,
1186                                             enemy_offense_max)
1187             player_damage = player_defense - enemy_offense
1188             # as damage decreases the health, it cannot be a positive number
1189             if player_damage > 0:
1190                 player_damage = 0
1191                 print_pause("Player damage: " + str(player_damage))
1192                 # decrement player's health based on damage
1193                 player_health = health_decrease(player_health, player_damage)
1194                 # display both player's and enemy's health
1195                 print_pause("Player health: " + str(player_health))
1196                 print_pause("Enemy health : " + str(enemy_health) + "\n")
1197             # if enemy loses the battle
1198             if player_health > 0 and enemy_health <= 0:
1199                 print_pause("You have successfully slain the " + enemy)
1200                 print_pause("You emerge victorious !!")
1201                 return "player wins"
1202             # if player loses the battle
1203             elif player_health <= 0 and enemy_health > 0:
1204                 print_pause("You lost this battle....")
1205                 # if the fight is in the contest grounds
1206                 # your health will be restored to bare minimum
1207                 if enemy == "warrior" or enemy == "gladiator":
1208                     print_pause("NOTE: Your health restored to 10")
1209                     print_pause("NOTE: Find a way to increase your health")
1210                     player_health = 10
1211                     print_pause("Your Health: " + str(player_health) + "\n")
1212                     return "enemy wins"
1213                 # if the fight is not in the contest grounds
1214                 # game ends
1215                 print_pause("You are dead.")
1216                 game_over = "activate"
1217                 village()
1218
1219 # funtion to trigger the location 'forest' in the village
1220 def forest():
1221     global forest_trigger
1222     global time_zone
1223     global player_items
1224     # if player got tired from exploring forest
1225     if forest_trigger == "activate":
1226         print_pause("You are tired, comeback tomorrow")
1227         village()
1228     # if player tries to enter forest at night time
1229     if time_zone == "night":
1230         # player must have a fire element to enter forest at night time
1231         if "Talisman" in player_items:
1232             print_pause("You venture into the forest tearing the shadows"
1233                         " with your fire element.")
1234             forest_choice()
1235         # if player does not have a fire element, he must be rejected
1236         else:
1237             print_pause("The forest is dark and dangerous at night")
1238             print_pause("You need a light source to move forward")
1239             print_pause("You head back to the village")
1240             village()
1241     # if player enters forest at daytime
1242     # as we bypassed night time conditional statement,
1243     # no conditional statement is required to specify day time trigger
1244     print_pause("You venture deep into the forest")
1245     print_pause("The forest is vast and home to all kinds of things."
1246               " This includes wild animals")
1247     forest_choice()
1248
1249 # A branch sequence to the funtion 'forest'
1250 def forest_choice():
1251     global time_zone
1252     global player_items
1253     global gift_items
1254     global grapes_day_count
1255     global flowers_day_count
1256     global glowing_blossom_day_count
1257
1258     print_pause("\nEnter the number to perform the desired action:")
1259     forest_action = input("1. Flowers")

```

```

1260         "\n2. Grapes"
1261         "\n3. Explore"
1262         "\n4. Go back to village\n")
1263 # if player chooses to collect flowers during daytime
1264 if forest_action == "1" and time_zone == "day":
1265     # if player already possess flowers in his items
1266     if "Flowers" in player_items:
1267         print_pause("You already have flowers in your inventory")
1268         forest_choice()
1269     # if player does not have flowers in his items
1270     else:
1271         print_pause("You collect different kinds of flowers")
1272         flowers_day_count = 0
1273         player_items.append("Flowers")
1274         gift_items.append("Flowers")
1275         forest_choice()
1276 # if player chooses to collect flowers during daytime
1277 # flowers are replaced by glowing blossom during night time
1278 elif forest_action == "1" and time_zone == "night":
1279     # if player already possess glowing blossom in his items
1280     if "Glowing Blossom" in player_items:
1281         print_pause("You already have glowing blossom in your inventory")
1282         forest_choice()
1283     # if player does not have glowing blossom in his items
1284     else:
1285         print_pause("You witness flowers glowing in the dark")
1286         print_pause("You collected some glowing blossom")
1287         glowing_blossom_day_count = 0
1288         player_items.append("Glowing Blossom")
1289         gift_items.append("Glowing Blossom")
1290         forest_choice()
1291 # if player chooses to collect grapes irrespective of time
1292 elif forest_action == "2":
1293     # if player already possess grapes in his items
1294     if "Grapes" in player_items:
1295         print_pause("You already have grapes in your inventory")
1296         forest_choice()
1297     # if player does not possess grapes in his items
1298     else:
1299         print_pause("You gather grapes and pack them carefully.")
1300         grapes_day_count = 0
1301         player_items.append("Grapes")
1302         gift_items.append("Grapes")
1303         forest_choice()
1304 # if player chooses to explore forest irrespective of time
1305 elif forest_action == "3":
1306     print_pause("NOTE: The forest is vast")
1307     print_pause("Enter the number of the location, you want to search:")
1308     forest_explore()
1309 # if player chooses to return to the village
1310 elif forest_action == "4":
1311     print_pause("You head back to the village")
1312     village()
1313 # condition to deal unrecognized input
1314 else:
1315     print_pause("Enter a valid number")
1316     forest_choice()
1317
1318
1319
1320 # A branch sequence to the function 'forest_choice'
1321 def forest_explore():
1322     global player_items
1323     global forest_explore_count
1324     global forest_trigger
1325     global animal_list
1326     global strong_animal_list
1327     # animal attack triggers at random with a probability of '1/4'
1328     animal_attack = random.choice(["no", "yes", "no", "no"])
1329     # if animal attack triggered
1330     if animal_attack == "yes":
1331         # if animal attack is during daytime,
1332         # random animal is selected from 'animal_list'
1333         if time_zone == "day":
1334             animal = random.choice(animal_list)
1335             print_pause("You have been ambushed by " + animal)
1336             fight_result = fight_sequence(animal, 0, 30, 0)
1337         # if animal attack is during night time,
1338         # random animal is selected from 'strong_animal_list'
1339         else:
1340             animal = random.choice(strong_animal_list)
1341             print_pause("You have been ambushed by " + animal)
1342             fight_result = fight_sequence(animal, 20, 40, 10)
1343         # if player wins
1344         if fight_result == "player wins":
1345             print_pause("You have courageously fought the animal and"
1346                         " killed it")
1347             print_pause("You resume your search")
1348         # if player loses, the sequence is executed in 'fight_sequence'
1349     # allows the player to enter an input to explore the forest
1350     forest_explore_action = int(input("Location : "))
1351     forest_explore_count += 1
1352     # if explore input/count is given more than 3 times,
1353     # player is restricted to enter until the next day
1354     if forest_explore_count > 3:
1355         print_pause("You are tired, come back tomorrow")
1356         print_pause("You have returned back to the village")
1357         forest_trigger = "activate"
1358         village()
1359     if forest_explore_action >= 1 and forest_explore_action < 20:
1360         print_pause("NOTE: You find nothing in this location, "
1361                     "better look somewhere else")
1362     elif forest_explore_action >= 20 and forest_explore_action < 25:
1363         print_pause("NOTE: You sense something not far from here")
1364     elif forest_explore_action >= 25 and forest_explore_action < 28:

```

```

1365     print_pause("NOTE: You are close to finding something")
1366 elif forest_explore_action == 28:
1367     print_pause("You found the 'Lost Tomb of King Authur'")
1368     print_pause("The tomb is sealed off and require a keyword password")
1369     tomb()
1370 elif forest_explore_action > 28 and forest_explore_action <= 32:
1371     print_pause("NOTE: You are close to finding something")
1372 elif forest_explore_action > 32 and forest_explore_action <= 36:
1373     print_pause("NOTE: You sense something not far from here")
1374 elif forest_explore_action > 36:
1375     print_pause("NOTE: You find nothing in this location, "
1376                "better look somewhere else")
1377 # condition to loop back to explore option
1378 else:
1379     print_pause("Enter a number to search the respective location")
1380     forest_explore()
1381 forest_explore()
1382
1383
1384 # A branch sequence to the funtion 'forest_explore'
1385 def tomb():
1386     global player_items
1387     global player_offense_min
1388     global player_offense_max
1389     global Keyword_count
1390
1391     tomb_keyword = input("Enter the keyword: ")
1392     # if the keyword input is correct
1393     if tomb_keyword == "Vamshi Krishna P":
1394         print_pause("\nThe Authur's Tomb gates open with crackling sound")
1395         print_pause("You enter thr tomb and inspect with utmost care")
1396         print_pause("The King Authur's body still holds the legendary "
1397                    "sword: The Excaliber")
1398         print_pause("You have acquired the Legendary sword: Excaliber")
1399         player_items.append("Excaliber")
1400         # player stats will be increased
1401         player_offense_min = 50
1402         player_offense_max = 80
1403         print_pause("\nNOTE: Your offense has increased "
1404                    "to: " + str(player_offense_max))
1405         print_pause("NOTE: You are stronger than ever")
1406         print_pause("You now head back to the village")
1407         village()
1408     # if the keyword input is not matched
1409     else:
1410         Keyword_count += 1
1411         # if number of tries is not exceeded
1412         if Keyword_count <= 3:
1413             print_pause("NOTE: Wrong password, try again")
1414             tomb()
1415         # if the keyword is mismatched 3 times in a row
1416         else:
1417             Keyword_count = 0
1418             print_pause("\nYou have exceeded the number of tries")
1419             print_pause("You have been locked out for the rest of the day")
1420             print_pause("come back tommorow to try again")
1421             print_pause("You head back to the village")
1422             village()
1423
1424
1425 # funtion to trigger the location 'cave' in the village
1426 def cave():
1427     global troll
1428     global player_items
1429     global gold
1430     global treasure_box
1431     global chest
1432     global monster
1433     # if the player have fire element to enter the cave
1434     if "Talisman" not in player_items:
1435         print_pause("You need fire element to explore the dark cave")
1436         print_pause("You head back to the village")
1437         village()
1438     print_pause("Enter the number of the route you want to take.")
1439     cave_choice = input("\n1. Dungeon1
1440                        "\n2. Dungeon2
1441                        "\n3. Dungeon3
1442                        "\n4. Back to village\n")
1443     # if player wants to enter route 1
1444     if cave_choice == "1":
1445         # if troll gaurding the route 1 is alive
1446         if troll == "alive":
1447             print_pause("This route is gaurded by a troll")
1448             print_pause("Troll is a strong opponent with thick skin")
1449             print_pause("what would you do?")
1450             route1_choice = input("\n1. Fight
1451                                "\n2. Run away\n").lower()
1452             # if player chooses to fight the troll
1453             if "fight" in route1_choice:
1454                 print_pause("You decided to fight the troll\n")
1455                 fight_result = fight_sequence("Troll", 20, 40, 10)
1456                 if fight_result == "player wins":
1457                     print_pause("You have triumphed over the troll")
1458                     # trigger to skip this part if visited again
1459                     troll = "defeated"
1460             # if player chooses to run
1461             elif "run" in route1_choice:
1462                 print_pause("You have strategically with drawn from a "
1463                            "death battle")
1464                 print_pause("You should head back to the village to "
1465                            "prepare for the battle, before trying again")
1466             cave()
1467         # condition to deal with unrecognized input
1468     else:

```



```

1468         print_pause("NOTE: Don't sweat, camly select "
1470                       "the number of your choice")
1471     cave()
1472 # if the troll has been already slain
1473 else:
1474     print_pause("The troll gaurding the route has been slain")
1475     print_pause("You can travel furthur deep into the route now")
1476 # After the troll part
1477 print_pause("\nYou travel deep into the Route 1\n")
1478 print_pause("There is a Treasure box at the certer of the room")
1479 print_pause("Would you like to open the Treasure box?")
1480 treasure_box = input("\n1. Yes"
1481                      "\n2. No\n").lower()
1482 # if player chooses to open the box
1483 if "yes" in treasure_box:
1484     # if the treasure has been already looted
1485     if treasure_box == "empty":
1486         print_pause("The Treasure Box is already looted by you")
1487         print_pause("There is nothing intersting here")
1488         print_pause("You head back to the cave entrance\n")
1489         cave()
1490     # if the treasure is not looted
1491     print_pause("Treasure Box is sealed with a peculiar lock")
1492     # if the player has the key acquired from princess
1493     if "Empress Sigil" in player_items:
1494         print_pause("You use the 'Empress Sigil' given by the "
1495                     "princess herself")
1496         print_pause("You opened the Treasure box and finds "
1497                     "a 'Locket'")
1498         print_pause("This looks like some kind of Family Heirloom")
1499         print_pause("NOTE: You have acquired a 'Royal Locket'")
1500         player_items.append("Royal Locket")
1501         # trigger to indicate, the treasure has been looted
1502         treasure_box = "empty"
1503         print_pause("You head back to the cave entrance\n")
1504         cave()
1505     # if the player does not have the key
1506     else:
1507         print_pause("You head back to the cave entrance\n")
1508         cave()
1509 # if player chooses not to open the box/ gives unrecognized input
1510 else:
1511     print_pause("You head back to the cave entrance\n")
1512     cave()
1513
1514 # if player wants to enter route 2
1515 elif cave_choice == "2":
1516     print_pause("NOTE: The entrance to the cave is locked and "
1517                 "requires a key to open")
1518     # if player has the key, won in the contest grounds
1519     if "Unknown Key" in player_items:
1520         print_pause("Would you like to try the 'Unknown Key', "
1521                     "you have won in the contest grounds?")
1522         route2_choice = input("\n1 Yes"
1523                               "\n2. No\n").lower()
1524     # if player tries to open the door using unknown key
1525     if "yes" in route2_choice:
1526         print_pause("The key matches the lock")
1527         print_pause("You opened the door and head furthur deep "
1528                     "into Route2")
1529         print_pause("In the room you found a chest eating dust")
1530         print_pause("The chest looks old and dangerous")
1531         print_pause("What would you do ?")
1532         chest_choice = input("\n1. Open"
1533                              "\n2. Head back\n")
1534     # if player wants to open the chest
1535     if "open" in chest_choice:
1536         # if the chest is already looted
1537         if chest == "empty":
1538             print_pause("You greedily open the chest again")
1539             print_pause("Unfortunately, the chest does not get "
1540                         "filled with gold automatically")
1541             print_pause("The chest is already looted by you")
1542             print_pause("There is nothing to do here")
1543             print_pause("You head back to the cave entrance\n")
1544             cave()
1545         # if the chest is not looted
1546         print_pause("You reluctantly open the chest")
1547         print_pause("Voila...!! Its a treasure chest "
1548                     "filled with gold....")
1549         print_pause("NOTE: You received 2000 gold")
1550         print_pause("Gold: " + str(gold) + ", + 2000")
1551         gold += 2000
1552         print_pause("Your Gold: " + str(gold))
1553         # trigger to indicate, the chest has been already looted
1554         chest = "empty"
1555         print_pause("You head back to the cave entrance\n")
1556         cave()
1557     # if player do not want to open chest/gives unrecognized input
1558     else:
1559         print_pause("You head back to the cave entrance\n")
1560         cave()
1561 # if player do not want to open door/gives unrecognized input
1562 else:
1563     print_pause("You head back to the cave entrance\n")
1564     cave()
1565 # if player does not have the key, won in the contest grounds
1566 else:
1567     print_pause("NOTE: Come back after you accquired "
1568                 "the key to 'Route2'")
1569     print_pause("You head back to the cave entrance\n")
1570     cave()
1571
1572 # if player wants to enter route 3

```



```

1572 elif cave_choice == "3":
1573     print_pause("This looks like a man made cellar")
1574     print_pause("There is a wooden post depicting as restricted region")
1575     print_pause("The cellar entrance is locked")
1576     # if player has the cellar key stolen from kings quarters
1577     if "Cellar Key" in player_items:
1578         print_pause("\nWould you like to open the cellar?")
1579         route3_choice = input("1. Yes\n2. No\n").lower()
1580         # if player wants to open the cellar door
1581         if "yes" in route3_choice:
1582             print_pause("\nYou entered the cellar while staying on guard")
1583             print_pause("You venture deep in inner section of the cave")
1584             # if the monster is alive
1585             if monster == "alive":
1586                 print_pause("\nThere is a huge monster sleeping across "
1587                             "the cellar, blocking the path")
1588                 print_pause("The monster gives off a deathly vibe and "
1589                             "extremely vicious")
1590                 print_pause("\nNOTE: This is a high level monster")
1591                 print_pause("NOTE: You probably need a special "
1592                             "sword-Excaliber,")
1593                 print_pause("equipped with fire element to defeat "
1594                             "the monster")
1595                 print_pause("\nDecide what to do?")
1596                 monster_choice = input("1. Wake the monster "
1597                                         "\n2. Come back later\n").lower()
1598                 # if player chooses to wake the monster
1599                 if "wake" in monster_choice:
1600                     print_pause("\nYou courageously draws your sword, "
1601                                 "ready to fight the monster")
1602                     print_pause("The monster lunge at you with great "
1603                                 "ferocity")
1604                     fight_result = fight_sequence("monster", 30, 50, 20)
1605                     # if player wins the battle
1606                     if fight_result == "player wins":
1607                         # trigger to indicate monster has been slain
1608                         monster = "defeated"
1609                         print_pause("\nYou gather remaining strength and "
1610                                     "venture further deep into the "
1611                                     "cellar\n")
1612                         cellar_rooms()
1613                         # if player loses the battle,
1614                         # result will be executed in the fight_sequence
1615                         # if player does not want to wake the monster
1616                         else:
1617                             print_pause("\nPerhaps you are not ready to face the "
1618                                         "monster, come back later")
1619                             print_pause("You head back to cave entrance\n")
1620                             cave()
1621                         # if monster has been slain
1622                         elif monster == "defeated":
1623                             print_pause("\nThere a monster lying dead in the middle of "
1624                                         "the cellar")
1625                             print_pause("You move forward\n")
1626                             cellar_rooms()
1627                         # if player does not want to open the cellar door
1628                         # or give unrecognized input
1629                         else:
1630                             print_pause("\nYou are not sure about the choice")
1631                             print_pause("Perhaps you should come back later !")
1632                             print_pause("You head back to the cave entrance")
1633                             cave()
1634                     # if player does not have the cellar key stolen from kings quarters
1635                     else:
1636                         print_pause("\nNOTE: Come back after you have acquired "
1637                                     "cellar key")
1638                         print_pause("You head back to cave entrance\n")
1639                         cave()
1640
1641 # if player choose to head back to the village
1642 elif cave_choice == "4":
1643     print_pause("You head back to the village\n")
1644     village()
1645 # condition to deal with unrecognized input
1646 else:
1647     print_pause("\nEnter a valid number")
1648     cave()
1649
1650
1651 # A branch sequence to the function 'cave'
1652 def cellar_rooms():
1653     global huge_monster
1654     global player_defense
1655     # allows the player to choose between two doors
1656     print_pause("\nYou find 2 doors- an Iron door and a "
1657                 "Wooden door, at the end of the cellar")
1658     print_pause("\nWhat would you do?")
1659     door_choice = input("1. Open Iron Door "
1660                        "\n2. Open Wooden Door "
1661                        "\n3. Head back\n").lower()
1662     # if player chooses to open the iron door
1663     if "iron" in door_choice:
1664         print_pause("\nYou enter the room with the Iron Door")
1665         # if huge monster is already slain
1666         if huge_monster == "defeated":
1667             print_pause("There a monster corpse rotting in the corner of room")
1668             print_pause("There is nothing else in this room")
1669             print_pause("You head back")
1670             cellar_rooms()
1671         # if huge monster is not slain/ alive
1672         print_pause("Suddenly, a Huge Monster jumps between "
1673                     "you and the entrance")
1674         print_pause("You cannot escape, leaving you no "
1675                     "choice but fight")

```

```

1678 fight_result = fight_sequence("Huge monster", 30, 60, 30)
1679 # if player wins the battle
1680 if fight_result == "player wins":
1681     print_pause("\nThe sudden battle made your blood rush to head")
1682     print_pause("It appears the room is used to trap the Huge Monster")
1683     print_pause("With the monster defeated you were able to "
1684                 "leave the room")
1685     # trigger to indicate, huge monster has been slain
1686     huge_monster = "defeated"
1687     cellar_rooms()
1688 # if player loses,
1689 # result will be executed in the fight_sequence
1690 # if player chooses to open wooden door
1691 elif "wooden" in door_choice:
1692     print_pause("\nThe room is very old and filled with corpses "
1693                 "of fighters")
1694     print_pause("It appears there were numerous attempts to defeat the "
1695                 "monster")
1696     print_pause("Apparently none of them succeeded")
1697     # if steel shield has not yet been collected
1698     if "Steel Shield" not in player_items:
1699         print_pause("\nYou search the room and finds a steel shield,\n "
1700                     "in the hands of a dead commander")
1701         print_pause("NOTE: You have acquired 'Steel Shield'")
1702         print_pause("NOTE: Your defense has been increased")
1703         print_pause("Defense : " + str(player_defense) + " , + 40")
1704         # player stats increased
1705         player_defense += 40
1706         player_items.append("Steel Shield")
1707         print_pause("Player Defense increased to: " + str(player_defense))
1708     # if steel shield has already been collected
1709     print_pause("\nThere is nothing else except for the rotting corpses "
1710                 "in this room ")
1711     print_pause("You head back to the cave entrance")
1712     cave()
1713 # if player chooses to head back to the cave entrance
1714 elif "back" in door_choice:
1715     print_pause("You head back to the cave entrance")
1716     cave()
1717 # condition to deal with unrecognized input
1718 else:
1719     print_pause("Enter your specific choice in words")
1720     cellar_rooms()
1721
1722 # function to trigger the location 'fort' in the village
1723 def fort():
1724     global player_items
1725     global fort_count
1726     global gold
1727     global time_zone
1728     global player_status
1729     global toll_status
1730     global kings_status
1731
1732     fort_count += 1
1733     # if player tries to enter fort at night time, he should be restricted
1734     if time_zone == "night":
1735         print_pause("The access to fort is restricted at night time")
1736         print_pause("Come back later")
1737         print_pause("You head back to the village")
1738         village()
1739     # if kings identity is revealed to the player, player should be
1740     # allowed into the fort even if the player is wanted for the crime
1741     if kings_status == "revealed":
1742         print_pause("You enter the fort demanding a meeting with the king")
1743         print_pause("You were allowed inside the inner halls of the fort")
1744         inner_fort()
1745     # player will be imprisoned, if he tries to
1746     # enter the fort and is wanted for a crime
1747     if player_status == "wanted":
1748         print_pause("It was not a wise choice to place yourself in front of "
1749                     "guards\n while wanted for trespassing.")
1750         print_pause("You got caught and sent to prison")
1751         prison()
1752     # if fort access is already granted to the player by the king
1753     if fort_access == "granted":
1754         print_pause("You have entered inside the fort")
1755         inner_fort()
1756     # if player enters the fort gates for the first time
1757     if fort_count == 1:
1758         print_pause("The village has a Magnificent Giant Fortress "
1759                     "made of stone")
1760         print_pause("You entered the fort entrance in astonishment "
1761                     "to stunning art engraved into the walls")
1762         print_pause("You seek an audience with the king but was denied "
1763                     "by the minister of the kingdom")
1764         print_pause("MINISTER: I'm afraid, I cannot let just anyone pass, kid")
1765         print_pause("MINISTER: But I will make an exemption for 500 gold, "
1766                     "what do you say :")
1767     # if player enters the fort gate more than once
1768     if fort_count > 1:
1769         print_pause("\nYou are at the fort gate")
1770         # if toll was already paid and possess tribute
1771         if toll_status == "paid" and "A Grand Tribute" in player_items:
1772             print_pause("MINISTER: You may pass now kid")
1773             print_pause("You head inside the fort")
1774             inner_fort()
1775         # if toll was already paid and does not possess tribute
1776         elif toll_status == "paid" and "A Grand Tribute" not in player_items:
1777             print_pause("MINISTER: Obtain a suitable gift to present "
1778                         "to the king first.")
1779             print_pause("MINISTER: Then come seek his audience, kid !")
1780             print_pause("\nYou head back to the village in search of the ")

```

```

1780         print_pause("worthy present to offer the king")
1781         village()
1782     # if toll is not paid
1783     print_pause("MINISTER: You have to pay the Toll kid")
1784
1785     # prompted to pay the toll on first visit to the fort gate
1786     print_pause("\nNOTE: Will you pay the TOLL ? what do you choose to do ?")
1787     toll = input("1. Pay 500 Gold"
1788                 "\n2. Deny"
1789                 "\n3. Persuade\n").lower()
1790
1791     # if player chooses to pay the toll
1792     if "gold" in toll:
1793         # if player does not have enough gold to pay the toll
1794         if gold < 500:
1795             print_pause("\nNOTE: Low in gold")
1796             print_pause("MINISTER: Perhaps you should come back after "
1797                         "earning the gold, kid")
1798             print_pause("You are sent back to the village")
1799             village()
1800         # if player has enough gold to pay the toll
1801         print_pause("\nYou pay the toll")
1802         print_pause("Gold: " + str(gold) + ", -500")
1803         gold -= 500
1804         print_pause("Your Gold: " + str(gold))
1805         # trigger to indicate toll has been paid
1806         toll_status = "paid"
1807         print_pause("\nMINISTER: Good choice kid., but I cannot just "
1808                   "let you pass yet")
1809         print_pause("MINISTER: Do you really think I'll let you pass for "
1810                   "500 gold alone ?")
1811         print_pause("MINISTER: Bring a gift to present to the king")
1812         print_pause("MINISTER: Can't really put you in his presense, "
1813                   "empty handed")
1814         # if player already has tribute in his possession on first visit
1815         if "A Grand Tribute" in player_items:
1816             print_pause("Luckily you have the right gift in your possession "
1817                       "to present to the king")
1818             print_pause("You were allowed inside the fort")
1819             inner_fort()
1820         # if player does not have tribute in his possession on first visit
1821         else:
1822             print_pause("MINISTER: Come back after you have the anything "
1823                       "worthy to present to a king")
1824             print_pause("You were sent back to the village")
1825             village()
1826     # if player chooses not to pay the toll
1827     elif "deny" in toll:
1828         print_pause("\nMINISTER: Then I won't let you pass kid")
1829         print_pause("You were sent back to the village")
1830         village()
1831     # if player chooses to convince without paying the toll
1832     elif "persuade" in toll:
1833         # player must have tribute in his possession to convince
1834         if "A Grand Tribute" in player_items:
1835             print_pause(player + ": I came from far lands bearing gifts "
1836                       "to the king")
1837             print_pause(player + ": Do you think it is a good idea to stop "
1838                       "me at the fort gate?")
1839             print_pause("The minister stay silent for a minute and "
1840                       "lets you pass")
1841             inner_fort()
1842         # if player does not have tribute in his possession to convince
1843         else:
1844             print_pause("\nYou have nothing to bargain with the access "
1845                       "to the fort")
1846             print_pause("NOTE: You can pay the Toll or acquire certain "
1847                       "item to persuade")
1848             print_pause("You head back to the village")
1849             village()
1850     # condition to deal with unrecognized input
1851     else:
1852         print_pause("MINISTER: Say the correct words kid")
1853         fort()
1854
1855     # A branch sequence to the funtion 'fort'
1856     def inner_fort():
1857         global player_items
1858         global time_zone
1859         global inner_fort_count
1860         global bell
1861         global fort_access
1862         global library_status
1863         global kings_status
1864
1865         print_pause("\nYou are inside the fort")
1866         print_pause("What do you choose :")
1867         # displays list of locations in a serial order, during daytime
1868         if time_zone == "day":
1869             inner_fort_choice = input(list_serial(fort_locations_day)).lower()
1870         # displays different list of locations in a serial, during night time
1871         elif time_zone == "night":
1872             inner_fort_choice = input(list_serial(fort_locations_night)).lower()
1873         # if player enters central hall during daytime
1874         if "hall" in inner_fort_choice and time_zone == "day":
1875             inner_fort_count += 1
1876             # intro to be triggered only once
1877             if inner_fort_count == 1:
1878                 print_pause("The central hall is filled with many "
1879                           "dignitaries and villagers")
1880                 print_pause("In the middle is the king occupying the grand throne")
1881                 print_pause("You present yourself to the king bearing "
1882                           "the 'Grand Tribute' as the gift to his highness")
1883                 print_pause("The king is pleased to the gift and welcomes ")

```

```

1888         "you to his fort")
1889     print_pause("\nNOTE: You gain access to the fort")
1890     print_pause("\nNOTE: You gain access to the 'Library' and 'Garden'")
1891     # trigger to give access to the fort
1892     fort_access = "granted"
1893     inner_fort()
1894 # if player enters central hall more than once, during daytime
1895 if inner_fort_count > 1:
1896     print_pause("The central hall is occupied as usual with "
1897                 "nobels and villagers")
1898     print_pause("They seemed to be busy")
1899     # if kings identity is revealed to the player by witch
1900     if kings_status == "revealed":
1901         challenge_king()
1902     # if kings identity is not revealed to the player by witch
1903     print_pause("NOTE: Perhaps you should come back later")
1904     print_pause("You head back to the inner halls")
1905     inner_fort()
1906 # if player enters central hall during night time
1907 elif "hall" in inner_fort_choice and time_zone == "night":
1908     # if security is alerted
1909     if bell == "activated":
1910         prison()
1911     # if security is not alerted
1912     print_pause("The central hall is empty and lack any security patrols")
1913     print_pause("This place is perfect to create any distractions")
1914     print_pause("\nWhat would you do ?")
1915     hall_choice = input("1. Ring the central hall bell"
1916                        "\n2. Go back\n").lower()
1917     # if player chooses to create a distraction
1918     if "bell" in hall_choice:
1919         print_pause("You rang the central hall bell")
1920         print_pause("The sound of the bell echoes throught out the fort")
1921         print_pause("The king and the gaurds are alerted !!")
1922         print_pause("\nNOTE: It is advised to avoid the "
1923                     "central hall untill next day")
1924         # trigger to indicate gaurds are alerted
1925         bell = "activated"
1926         print_pause("You leave the central hall")
1927         inner_fort()
1928     # if player chooses to not to create a distraction
1929     elif "back" in hall_choice:
1930         print_pause("May be it is best to consider all options")
1931         print_pause("You head back to the inner fort")
1932         inner_fort()
1933     # condition to deal with unrecognized input
1934     else:
1935         print_pause("Choose the correct words, ")
1936         print_pause("come back after you make up your mind")
1937         inner_fort()
1938 # if player chooses to enter the library
1939 elif "library" in inner_fort_choice:
1940     # if library is already brunt by player as a distraction
1941     if library_status == "burnt":
1942         # if gaurds are alerted
1943         if bell == "activated":
1944             prison()
1945         # if gaurds are not alerted
1946         print_pause("The library has been burnt to crisp")
1947         print_pause("There is nothing to salvage")
1948         print_pause("You head back to inner halls of fort")
1949         inner_fort()
1950     if time_zone == "night":
1951         # if gaurds are alerted
1952         if bell == "activated":
1953             prison()
1954         # if gaurds are not alerted
1955         print_pause("\nThis is another choice to create a distraction")
1956         print_pause("What would you do ?")
1957         library_choice = input("1. Burn the Library"
1958                               "\n2. Head back to inner fort\n").lower()
1959         # if player chooses to burn library as a distraction
1960         if "burn" in library_choice:
1961             print_pause("\nYou take a lantern in the corner of the "
1962                         " library,")
1963             print_pause("and started fire in multiple places to "
1964                         "confuse the soldiers")
1965             print_pause("NOTE: It is advisable to avoid library for "
1966                         "the rest of the day")
1967             # trigger to indicate library is burnt
1968             library_status = "burnt"
1969             # trigger to indicate gaurds are alerted
1970             bell = "activated"
1971             print_pause("\nyou head back to the inner halls")
1972             inner_fort()
1973         # if player chooses not to burn library as a distraction
1974         elif "fort" in library_choice:
1975             print_pause("\nIt is wise to consider other options")
1976             print_pause("You head back to inner halls of the fort")
1977             inner_fort()
1978         # condition to deal with unrecognized input
1979         else:
1980             print_pause("\nEnter the correct words next time")
1981             print_pause("You were sent back to inner halls of the fort")
1982             inner_fort()
1983 # if library access is given to the player by the king
1984 if fort_access == "granted":
1985     print_pause("\nYou have entered the Great Library of the village")
1986     print_pause("The library has more than 1000 books "
1987                 "collected for over 300 years")
1988     print_pause("You may find important information here:")
1989     print_pause("what would you like to study ?")
1990     library_study = input("\n1. Ancient Monster")

```

```

1989         "\n2. Village"
1990         "\n3. Fort"
1991         "\n4. Kingdom History\n").lower()
1992 # if player wants to study about the monster
1993 if "monster" in library_study:
1994     print_pause("\nThe monster is old and witty")
1995     print_pause("It is said to possess the strength greater than "
1996                 "any creature known")
1997     print_pause("The monster has the ability to disguise itself "
1998                 "as human being")
1999     print_pause("It can also manipulate other humans according "
2000                 "to its will")
2001     print_pause("The monster's skin is thick enough to deflect any"
2002                 " weapon's attacks")
2003 # if player wants to study about the village
2004 elif "village" in library_study:
2005     print_pause("\nThe village is formed 40 years after the "
2006                 "construction of fort")
2007     print_pause("The early settlers of the villagers were "
2008                 "Immigrants of other kingdom")
2009     print_pause("Due to famine, most of the people in the "
2010                 "neighbouring kingdoms,")
2011     print_pause("immigrated to this kingdom in hoope of new life")
2012     print_pause("The king was kind enough to provide food and "
2013                 "shelter to every one")
2014     print_pause("Eventually the village prospered and became an "
2015                 "important trade center")
2016 # if player wants to study about the fort
2017 elif "fort" in library_study:
2018     print_pause("\nThe fort is old and aged over 200 years old")
2019     print_pause("It has been built as a military post to overwatch"
2020                 " any enemy activities")
2021     print_pause("The fort has withstood many seizes and "
2022                 "weather conditions")
2023     print_pause("Over the time, only few minor renovations are "
2024                 "made to the fort")
2025 # if player wants to study about the kingdom
2026 elif "kingdom" in library_study:
2027     print_pause("\nThe kingdom has a history of over 600 years")
2028     print_pause("Many kings have ruled the kingdom over the time")
2029     print_pause("The kingdom has been ruled by the 'Rafel "
2030                 "Dynasty' and has never been fallen")
2031     print_pause("The kingdom eventually expanded and divided "
2032                 "among many kings for effective ruling")
2033 # condition to deal with unrecognized input
2034 else:
2035     print_pause("\nThere are no records of the word you mentioned")
2036     print_pause("You were sent back to inner halls of the fort")
2037     inner_fort()
2038 # after each study player is sent back to the 'inn'
2039     print_pause("\nYou are tired from all the reading")
2040     print_pause("You have no choice but to rest till morning")
2041     print_pause("You head back to the INN to rent a room")
2042     room()
2043 # if fort access is not given to the player
2044 elif fort_access != "granted":
2045     print_pause("\nLibrary is not accessable to everyone !")
2046     print_pause("You need permit from the king himself")
2047     print_pause("You were sent back to the inner halls of the fort")
2048     inner_fort()
2049
2050 # if player chooses to enter the garden
2051 elif "garden" in inner_fort_choice:
2052     # if player enters the garden during night time
2053     if time_zone == "night":
2054         print_pause("\nThe garden is dark without any light")
2055         print_pause("One can hide / sneak in the garden at night "
2056                     "without getting detected")
2057         print_pause("There is nothing interesting here")
2058         print_pause("\nWhat would you do ?")
2059         garden_choice = input("1. Go Back to Fort"
2060                               "\n2. Enter Secret Passage\n").lower()
2061         # if player wants to enter fort
2062         if "fort" in garden_choice:
2063             print_pause("\nYou sneaks into the inner halls of fort")
2064             inner_fort()
2065         # if player wants to enter secret passage
2066         elif "secret" in garden_choice:
2067             print_pause("\nYou enter the Secret Passage")
2068             secret_passage()
2069         # condition to deal with unrecognized input
2070         else:
2071             print_pause("\nEnter the correct choice")
2072             print_pause("You were sent to the inner halls anyway")
2073             inner_fort()
2074     # if fort/garden access was not granted to the player
2075     if fort_access != "granted":
2076         print_pause("\nThe garden area is restricted to strangers")
2077         print_pause("You need permit from the king himself")
2078         print_pause("You were sent back to the inner halls of the fort")
2079         inner_fort()
2080     # if fort/garden access is granted to the player
2081     elif fort_access == "granted" and time_zone == "day":
2082         print_pause("\nThe garden looks beautiful under the radiant sun")
2083         print_pause("You find princess roaming in the garden")
2084         princess()
2085
2086 # if player chooses to enter the quaters
2087 elif "quaters" in inner_fort_choice:
2088     # quaters are restricted during daytime
2089     if time_zone == "day":
2090         print_pause("The quaters are not accessable to anyone "
2091                     "but royal family")
2092         print_pause("You head back to the inner halls")

```

```

2092         inner_fort()
2093     # player can sneak into the quaters during night time
2094     if time_zone == "night":
2095         print_pause("You sneak into the royal quaters")
2096         royal_quaters()
2097     # if player chooses to return to the village
2098     elif "village" in inner_fort_choice:
2099         print_pause("You head back to the village")
2100         village()
2101     # condition to deal with unrecognized input
2102     else:
2103         print_pause("NOTE: Enter the correct words....\n")
2104         inner_fort()
2105
2106
2107 # A branch sequence to the funtion 'inner_fort'
2108 def royal_quaters():
2109     global bell
2110     global player_items
2111     # player's choice
2112     print_pause("Where would you like to go ?")
2113     royal_quaters_choice = input("\n1. Kings Quaters
2114                                "\n2. Princess Quaters
2115                                "\n3. Go back to fort \n").lower()
2116     # if player chooses to sneak into king's quaters
2117     if "king" in royal_quaters_choice:
2118         # if gaurds are alerted and deistraction created
2119         if bell == "activated":
2120             print_pause("\nThe room is empty, perhaps the distraction worked"
2121                        " fine!!")
2122             print_pause("You tip-toed into the king's quaters")
2123             print_pause("You searched the room for the cellar keys")
2124             print_pause("The cellar keys are were found inside a cabinet")
2125             print_pause("NOTE: You have acquired the Cellar keys")
2126             player_items.append("Cellar Key")
2127             print_pause("NOTE: It is adviced to leave the fort for the day")
2128             print_pause("You head back to the inner halls of fort")
2129             inner_fort()
2130             print_pause("\nThe king is sleeping on the Royal bed")
2131             print_pause("NOTE: It is adviced to create a distraction before"
2132                        " going inside")
2133             kings_quaters()
2134         # if player chooses to sneak into princess quaters
2135         elif "princess" in royal_quaters_choice:
2136             print_pause("\nThe princess is fast asleep")
2137             print_pause("Perhaps it is better to leave")
2138             print_pause("You head back to the inner halls")
2139             inner_fort()
2140         # if player chooses to return to the fort
2141         elif "fort" in royal_quaters_choice:
2142             print_pause("\nYou head back to the inner halls of the fort")
2143             inner_fort()
2144         # condition to deal with unrecognized input
2145         else:
2146             print_pause("\nEnter the correct choice of words")
2147             royal_quaters()
2148
2149
2150 # A branch sequence to the funtion 'inner_fort'
2151 def kings_quaters():
2152     # if player already stole the cellar key from kings quaters
2153     if "Cellar Key" in player_items:
2154         print_pause("\nNOTE: You alredy got the Cellar Key")
2155         print_pause("There is nothing else to accomplish here")
2156         print_pause("You head back to the inner halls")
2157         inner_fort()
2158     # if player has not yet stolen the cellar key from kings quaters
2159     print_pause("\nWill you take the risk ?")
2160     print_pause("Enter the number of your choice:")
2161     kings_quaters_choice = input("1. Yes, go inside and take the risk"
2162                                "\n2. No, go back and consider a "
2163                                "distraction\n").lower()
2164     # if player decided to sneak into the kings quaters
2165     if kings_quaters_choice == "1":
2166         print_pause("\nYou sneak into the kings quaters to search for "
2167                    "cellar key")
2168         print_pause("The king started to wake up")
2169         print_pause("\nWould you still look for the cellar key ?")
2170         search_choice = input("1. Yes"
2171                              "\n2. No\n").lower()
2172     # if player chooses to take take the risk
2173     if "yes" in search_choice:
2174         print_pause("\nThe king wakes up and find you sneaking in "
2175                    "his room")
2176         print_pause("The king does not like Trespassers")
2177         print_pause("You were caught and imprisoned")
2178         prison()
2179     # if player chooses not to take take the risk
2180     elif "no" in search_choice:
2181         print_pause("\nYou avoided getting caught by withdrawing from "
2182                    "the room")
2183         print_pause("You head back to the inner halls of the fort")
2184         print_pause("Perhaps it is best to consider a distraction, \n"
2185                    "before trying again !!")
2186         inner_fort()
2187     # condition to deal with unrecognized input
2188     else:
2189         print_pause("NOTE: You were not sure of your choice")
2190         print_pause("Hence you are sent back to the inner halls to "
2191                    "re-evalute your choices")
2192         inner_fort()
2193     # if player decided not to sneak into the kings quaters
2194     elif kings_quaters_choice == "2":
2195         print_pause("You decided to wise up and create a distraction ")

```



```

2300     # a secret passage will be added to the village locations
2301     village_locations.append("Secret Passage")
2302     print_pause("You head back to the inner halls of the fort")
2303     inner_fort()
2304     # if quest is not yet completed but briefed already
2305     elif "Empress Sigil" in player_items:
2306         print_pause("\nPRINCESS: Did you get the family "
2307                     "heirloom " + player + " ?")
2308         print_pause("PRINCESS: I hope you find it !!")
2309         print_pause("You head out to retrieve the stolen item")
2310         village()
2311     # if quest is not yet completed nor briefed
2312     print_pause("\nPRINCESS: I have a request to make of you " + player)
2313     print_pause("PRINCESS: There ia a family heirloom passed down for"
2314                 " generations")
2315     print_pause("PRINCESS: My mother gave it to me as a present,"
2316                 " on my 16th birthday, before she passed away")
2317     print_pause("PRINCESS: Apparently its been lost, more like stolen")
2318     print_pause("PRINCESS: I heard it has been stashed in one of the cave"
2319                 " partitions")
2320     print_pause("PRINCESS: Will you bring it for me " + player + " ??")
2321     print_pause("PRINCESS: The heirloom is sealed in a royal box")
2322     print_pause("PRINCESS: You may need this royal sigil to open it")
2323     player_items.append("Empress Sigil")
2324     print_pause("NOTE: You have acquired 'Empress Sigil'.")
2325     print_pause("You head back to the inner halls of the fort")
2326     inner_fort()
2327
2328     # if princesss quest is does not meet required conditions to get triggered
2329     # and player wants to give gift to the princess
2330     print_pause("\nWhat do you choose to offer ?")
2331     # if player does not have anything to offer to the princesss
2332     # / len(gift_items) == 1 / is used because / leave / option
2333     # is used inside gift list
2334     if len(gift_items) == 1:
2335         print_pause("\nNOTE:You have nothing to offer to the princess")
2336         print_pause("NOTE: Come back later")
2337         print_pause("You head back to the inner halls")
2338         inner_fort()
2339     # displays list of gift items in a serial list
2340     princess_choice = input(list_serial(gift_items)).lower()
2341     # if player wants to gift flowers and have them in his possession
2342     if "flowers" in princess_choice and "Flowers" in gift_items:
2343         print_pause("PRINCESS: Beautiful flowers but nothing I haven't "
2344                     "seen in this garden")
2345         player_items = remove_item(player_items, "Flowers")
2346         gift_items = remove_item(gift_items, "Flowers")
2347         print_pause("NOTE: 'Flowers' has been removed from your possession\n")
2348         princess_quest()
2349     # if player wants to gift glowing blossom and have them in his possession
2350     elif "glowing" in princess_choice and "Glowing Blossom" in gift_items:
2351         print_pause("PRINCESS: I have never seen a flower of this kind")
2352         print_pause("PRINCESS: Thank you " + player)
2353         player_items = remove_item(player_items, "Glowing Blossom")
2354         gift_items = remove_item(gift_items, "Glowing Blossom")
2355         print_pause("NOTE: 'Glowing Blossom' has been removed from your "
2356                     "possession\n")
2357     # if princess do not have them in her possession
2358     # she consider them as a gift item
2359     if "Glowing Blossom" not in princess_items:
2360         princess_items.append("Glowing Blossom")
2361         print_pause("NOTE: You relationship strengthened")
2362         princess_quest()
2363     # if player wants to gift freya dress and have them in his possession
2364     elif "freya" in princess_choice and "Robes of Freya" in gift_items:
2365         print_pause("PRINCESS: Odd dress !!, though unique it is not to "
2366                     "my taste")
2367         player_items = remove_item(player_items, "Robes of Freya")
2368         gift_items = remove_item(gift_items, "Robes of Freya")
2369         print_pause("NOTE: 'Robes of Freya' has been removed from your "
2370                     "possession\n")
2371         princess_quest()
2372     # if player wants to gift princess dress and have them in his possession
2373     elif "princess" in princess_choice and "Dress of a princess" in gift_items:
2374         print_pause("PRINCESS: Elegant dress, I like it " + player)
2375         player_items = remove_item(player_items, "Dress of a princess")
2376         gift_items = remove_item(gift_items, "Dress of a princess")
2377         print_pause("NOTE: 'Dress of a princess' has been removed from "
2378                     "your possession\n")
2379         princess_quest()
2380     # if player wants to gift steel armour and have them in his possession
2381     elif "armour" in princess_choice:
2382         if "Glittering steel armour" in gift_items:
2383             print_pause("PRINCESS: Thank you very much " + player + " !!")
2384             print_pause("PRINCESS: I always wanted to try one of these")
2385             print_pause("PRINCESS: But my father never let me")
2386             print_pause("PRINCESS: You have my gratitude " + player)
2387             player_items = remove_item(player_items, "Glittering steel armour")
2388             gift_items = remove_item(gift_items, "Glittering steel armour")
2389             print_pause("NOTE: 'Glittering steel armour' has been removed "
2390                         "from your possession\n")
2391             # if princess do not have them in her possession
2392             # she consider them as a gift item
2393             if "Glittering steel armour" not in princess_items:
2394                 princess_items.append("Glittering steel armour")
2395                 print_pause("NOTE: You relationship strengthened")
2396         else:
2397             print_pause("you do not have Glittering steel armour")
2398             princess_quest()
2399     # if player wants to gift grapes and have them in his possession
2400     elif "grapes" in princess_choice and "Grapes" in gift_items:
2401         print_pause("PRINCESS: I appreciate the intention but I am not "
2402                     "hungry now")
2403         player_items = remove_item(player_items, "Grapes")

```



```

2408     gift_items = remove_item(gift_items, "Grapes")
2409     print_pause("NOTE: 'Grapes' has been removed from your possession\n")
2410     princess_quest()
2411 # if player wants to gift jewellery and have them in his possession
2412 elif "jewellery" in princess_choice and "Jewellery" in gift_items:
2413     print_pause("PRINCESS: Splendid jewellery !!")
2414     print_pause("PRINCESS: Perhaps I will wear it for our next meeting")
2415     player_items = remove_item(player_items, "Jewellery")
2416     gift_items = remove_item(gift_items, "Jewellery")
2417     print_pause("NOTE: 'Jewellery' has been removed from your"
2418                 " possession\n")
2419     # if princess do not have them in her possession
2420     # she consider them as a gift item
2421     if "Jewellery" not in princess_items:
2422         princess_items.append("Jewellery")
2423         print_pause("NOTE: You relationship strengthened")
2424     princess_quest()
2425 # if player chooses to leave the garden
2426 elif "leave" in princess_choice:
2427     print_pause("You head back to the inner halls")
2428     inner_fort()
2429 # condition to deal with unrecognized input
2430 else:
2431     print_pause("Enter the correct gift...")
2432     princess_quest()
2433
2434 # A branch sequence to the multiple funstions
2435 def prison():
2436     global bell
2437     global player_items
2438     global gold
2439     global time_zone
2440     global village_locations
2441     global fort_locations
2442     global player_status
2443     print_pause("\nYou were caught by the gaurds")
2444     print_pause("The king is not tolrent of trespassers")
2445     print_pause("You were sent to prison")
2446     print_pause("After 4 days of imprisonment, you strtred to lose hope")
2447     print_pause("Many time sensitive items in your inventory, "
2448                 "like grapes and flowers are diminished")
2449     # time is reset to night
2450     # gaurds are reset to not alerted as player is captured
2451     # player status is changed to wanted
2452     time_zone = "night"
2453     bell = "deactivated"
2454     player_status = "wanted"
2455     # any time sensitive items in player's possession will be removed
2456     player_items = remove_item(player_items, "Grapes")
2457     player_items = remove_item(player_items, "Flowers")
2458     player_items = remove_item(player_items, "Glowing Flowers")
2459     print_pause("\nOn 4th day night, when the new moon is in the sky "
2460                 "with no light,")
2461     print_pause("Princess herslf comes to save you from the shackles "
2462                 "of imprisonment")
2463     print_pause("You were rescued and sneaked into the inner fort")
2464     print_pause("You are on your own from here")
2465     inner_fort()
2466
2467 # A branch sequence to the multiple funstions
2468 def secret_passage():
2469     global time_zone
2470     # player's choice
2471     print_pause("You are in the secret passage")
2472     print_pause("\nchoose a location to travel undetected:")
2473     secret_passage_choice = input("1. Royal Garden"
2474                                   "\n2. Village"
2475                                   "\n3. Explore the passage\n").lower()
2476 # if player chooses to enter garden during daytime
2477 if "garden" in secret_passage_choice and time_zone == "day":
2478     print_pause("\nYou head towards the garden in the fort")
2479     print_pause("NOTE: You might expose the secret passage in"
2480                 " broad-day-light")
2481     print_pause("\nDo you want to risk it ?")
2482     passage_to_garden_choice = input("1. Yes"
2483                                     "\n2. No\n").lower()
2484     # if player wants to risk it
2485     if "yes" in passage_to_garden_choice:
2486         print_pause("\nYou have entered the garden during the day time")
2487         print_pause("The gaurds caught you and sent you to prison")
2488         prison()
2489     # if player does not want to risk it
2490     else:
2491         print_pause("You have decided to use another route \nto avoid risk"
2492                     " exposing the secret path")
2493         secret_passage()
2494 # if player chooses to enter garden during night time
2495 elif "garden" in secret_passage_choice and time_zone == "night":
2496     print_pause("\nYou entered the garden area through secret passage")
2497     print_pause("As it is dark, you are able to sneak past gaurds")
2498     print_pause("You then enter the inner halls of fort")
2499     inner_fort()
2500 # if player chooses to enter village during night time
2501 elif "village" in secret_passage_choice:
2502     print_pause("You head towards the village")
2503     village()
2504 # if player chooses to explore the passage
2505 elif "explore" in secret_passage_choice:
2506     print_pause("\nYou hear peculiar sounds coming from inside "
2507                 "the passage")
2508     print_pause("You decided to investigate the passage")
2509     print_pause("You went furthur deep into the passage")

```

```

2508     print_pause("You have stumbled upon an old door")
2509     print_pause("The lock resembles the 'Empress Sigil' the princess"
2510                 " gave you")
2511
2512     witch()
2513     # condition to deal with unrecognized input
2514     else:
2515         print_pause("\nNOTE: Enter the correct choice of words\n")
2516         secret_passage()
2517
2518
2519 # A branch sequence to the function 'secret_passage'
2520 def witch():
2521     global witch_room_count
2522     # player's choice
2523     print_pause("\nWould you open the lock ?")
2524     witch_choice = input("1. open"
2525                          "\n2. Head back\n").lower()
2526     # if players chooses to open the door
2527     if "open" in witch_choice:
2528         witch_room_count += 1
2529         # intro to be displayed only once
2530         if witch_room_count == 1:
2531             print_pause("\nYou enter the old room while being on guard")
2532             print_pause("The room is full of tree roots from the ground "
2533                         "above, \nwhich happen to be the royal garden")
2534             print_pause("You witness an old figure lurking behind your back")
2535             print_pause("You react by swinging your sword but to no avail")
2536             print_pause("The sword passes right through the thing doing no "
2537                         "harm")
2538             print_pause("The thing asks you to calm down and try to reason"
2539                         " with you")
2540             print_pause("It introduces itself as an ghost and an victim to"
2541                         " king's true nature\n")
2542             print_pause("GHOST: The king is not who the people think he is.. "
2543                         "!!")
2544             print_pause("GHOST: I have been branded as witch for trying to "
2545                         "expose the king's true self !!")
2546             print_pause("GHOST: He then locks me here to die")
2547             print_pause("GHOST: I want you to take revenge on the king in my "
2548                         "stead")
2549             print_pause("GHOST: The king is the monster in disguise")
2550             print_pause("GHOST: You cannot kill him with normal weapons")
2551             print_pause("GHOST: I can help you defeat him")
2552             print_pause("GHOST: Will you do it, o warrior ??")
2553             witch_sequence()
2554         # if player enters the room more than once
2555         if witch_room_count > 1:
2556             witch_sequence()
2557     # if players chooses not to open the door and head back
2558     elif "back" in witch_choice:
2559         print_pause("\nYou decided to come back later")
2560         print_pause("You head back to the secret passage intersection")
2561         secret_passage()
2562     # condition to deal with unrecognized input
2563     else:
2564         print_pause("\nNOTE: Enter the correct words\n")
2565         witch()
2566
2567
2568 # A branch sequence to the function 'witch'
2569 def witch_sequence():
2570     global player
2571     global tomb_password
2572     global kings_status
2573     # if tomb password is already revealed by the witch
2574     if tomb_password == "revealed":
2575         print_pause("\nGHOST : What do you need warrior?")
2576         print_pause("GHOST : I'm afraid I can't help you except with the "
2577                     "tomb spell")
2578         print_pause("GHOST: Do you want me to repeat the spell to you again"
2579                     " warrior ?")
2580         password_again = input("\n1. Yes"
2581                               "\n2. No\n").lower()
2582         # if player wants to hear the password again
2583         if "yes" in password_again:
2584             print_pause("\nGHOST: Tomb Spell - Vamshi Krishna P ")
2585             print_pause("NOTE: Password is case sensitive")
2586             print_pause("\nYou travel back to the secret passage intersection")
2587             secret_passage()
2588         # if player does not need to hear the password again
2589         elif "no" in password_again:
2590             print_pause("You travel back to the secret passage intersection")
2591             secret_passage()
2592         # condition to deal with unrecognized input
2593         else:
2594             print_pause("\nGHOST: I don't understand your words warrior !!")
2595             witch_sequence()
2596     # if tomb password is not revealed by the witch yet
2597     print_pause("\nGHOST: Will you help me expose the king ?")
2598     witch_sequence_choice = input("1. Yes, I will"
2599                                   "\n2. My be later\n").lower()
2600     # if player choose to expose king's identity
2601     if "yes" in witch_sequence_choice:
2602         print_pause("\n" + player + " : Yes, I will help you")
2603         print_pause("GHOST: You have my gratitude warrior")
2604         print_pause("GHOST: Now listen carefully, normal swords cannot do any"
2605                     " harm to the monster")
2606         print_pause("GHOST: He can only be harmed by the legendary sword - "
2607                     "Excaliber")
2608         print_pause("GHOST: Legend has it, The sword is made to kill all evil")
2609         print_pause("GHOST: The sword has been sealed along with the king "
2610                     "Arthur's corpse")
2611         print_pause("GHOST: If you haven't found about the whereabouts of "
2612                     "the tomb yet, ")

```

```

2612     print_pause("GHOST: It is located somewhere inside the forest")
2613     print_pause("\nGHOST: The spell to unbind the seal "
2614                 "is : " + "'Vamshi Krishna P'\n")
2615     print_pause("NOTE: You may enter King Authur's Tomb now")
2616     print_pause("GHOST: Also you might need a sturdy sheild to defend "
2617                 "against monster's attacks")
2618     print_pause("GHOST: I can't help you with the sheild though !!")
2619     print_pause("GHOST: I plead you warrior, bring peace to my "
2620                 "agonized soul")
2621     print_pause(player + " : I will, you may rest in peace now\n")
2622     # tomb password and king's identity is revealed by the witch
2623     tomb_password = "revealed"
2624     kings_status = "revealed"
2625     print_pause("You head back to the secret passage intersection")
2626     secret_passage()
2627
2628     # if player chooses not to expose king's identity
2629     elif "later" in witch_sequence_choice:
2630         print_pause("\nYou were not sure of the choice")
2631         print_pause("You decide to come back later")
2632         print_pause("You head back to the secret passage intersection")
2633         secret_passage()
2634     # condition to deal with unrecognized input
2635     else:
2636         print_pause("\nGHOST: I don't understand your words warrior !!")
2637         witch_sequence()
2638
2639 # A branch sequence to the function 'inner_fort'
2640 def challenge_king():
2641     # player's choice
2642     print_pause("\nWould you challenge the king to a battle ?")
2643     challenge = input("1. yes, challenge the king "
2644                     "\n2. No, not yet\n").lower()
2645     # if player chooses to fight the king
2646     if "yes" in challenge:
2647         print_pause("\nYou challenge the king to single combat")
2648         print_pause("The king did not hesitate to accept the challenge\n")
2649         fight_result = fight_sequence("King", 50, 70, 50)
2650         # if player wins
2651         if fight_result == "player wins":
2652             print_pause("The king is on his final breath")
2653             print_pause("You approach him with your sword drawn")
2654             kings_fate_scenario()
2655     # if player chooses not to fight the king
2656     elif "no" in challenge:
2657         print_pause("\nYou were not ready yet")
2658         print_pause("You decided to wait for the right time")
2659         print_pause("You head back to the village")
2660         village()
2661     # condition to deal with unrecognized input
2662     else:
2663         print_pause("\nNOTE: Enter the correct words\n")
2664         challenge_king()
2665
2666 # A branch sequence to the function 'challenge_king'
2667 def kings_fate_scenario():
2668     global monster_list
2669     # player's choice
2670     print_pause("\nEnter the number of your choice :")
2671     kings_fate = input("1. Kill the king and finish once and for all"
2672                      "\n2. Somethings not right, spare his life\n")
2673     # if player decides to kill the king
2674     if kings_fate == "1":
2675         print_pause("\nYou deliver a powerful blow to the king ")
2676         print_pause("The king took the final blow with out any "
2677                     "resistance")
2678         print_pause("almost as if he wanted you to slay him !!")
2679         print_pause("You have slain the monster, but something feels unsettled"
2680                     " to you.")
2681         print_pause("You were branded as king's slayer and abolished "
2682                     "from the kingdom")
2683         print_pause(".")
2684         print_pause(".")
2685         print_pause(".")
2686         print_pause(".")
2687         print_pause("After 4 weeks of the incident")
2688         print_pause("You were drinking in an Inn in the neighbouring"
2689                     " town")
2690         print_pause("You enquire the BARTENDER for the info as a daily"
2691                     " routine")
2692         print_pause("According to the information,")
2693         print_pause("The princess was kidnapped by the monster and "
2694                     "never returned !!")
2695         print_pause("As suspected, the monster was not the king !!")
2696         print_pause("There is something behind the scenes playing the king")
2697         print_pause("NOTE: Play the game again for different end "
2698                     "scenario")
2699         print_pause("Time to return to the village ....//\n")
2700         print_pause("----- GAME ENDED -----")
2701         credits()
2702     # if player decides not to kill the king
2703     elif kings_fate == "2":
2704         print_pause("\nYou take a leap of faith, to spare the king's life")
2705         print_pause("You gave the king a chance to explain his actions")
2706         print_pause("The king is in no condition to talk to you,")
2707         print_pause("He appears to be be possessed")
2708         print_pause("\nYou sense a dark presence lurking in the shadows !!")
2709         print_pause("It appears to be the monster, manipulating the king all"
2710                     " along")
2711         print_pause("The king is nothing but a puppet,\n a victim to monster's"
2712                     " vicious nature")
2713         print_pause("You leap towards the monster at full throttle")
2714         print_pause("The monster was caught off gaurd and wonded badly")

```

```

2718 # monster will be revealed as a random from a list of characters
2718 monster_trueform = random.choice(monster_list)
2719 print_pause("\n\nThe monster actually is the " + monster_trueform + ", "
2720             "\ndisguised itself as a human !!\n")
2721 print_pause("The monster is both wounded and drain from "
2722             "manipulating king")
2723 print_pause("This is the perfect chance to end this once "
2724             "and for all..\n")
2725 fight_result = fight_sequence("Monster", 40, 60, 40)
2726 if fight_result == "player wins":
2727     print_pause("The monster is finally slain")
2728     print_pause("You rescued the king and the village from the "
2729                 "monster's evil plan")
2730     print_pause("You were rewarded with gold and title in the "
2731                 "kingdom\n")
2732     print_pause("----- GAME ENDED -----")
2733     credits()
2734 # condition to deal with unrecognized input
2735 else:
2736     print_pause("\nEnter the number of your choice to decide king's fate")
2737     kings_fate_scenario()
2738
2739
2740 # function to be displayed if player successfully complete the game
2741 def credits():
2742     global player
2743     print_pause("\n\n          // Congratulations on completing the game //")
2744     print_pause("\n\n\n\n          DISCLAIMER          ")
2745     print_pause("\n>> This game is a pure work of fiction,")
2746     print_pause("resemblance to any king or Protagonist is a pure "
2747                 "coincidence")
2748     print_pause(">> No monsters are actually slain / harrassed in "
2749                 "this game")
2750     print_pause(">> We strictly followed the 'Player-Monster-Coexist-World"
2751                 "' Peace' rule in this game\n\n")
2752     print_pause("CREDITS ")
2753     print_pause("\n\nThis game was developed by and\nsole property of "
2754                 "'// 'Vamshi Krishna P' //'\n")
2755     print_pause("Cast and Crew:")
2756     print_pause("Protagonist - " + player)
2757     print_pause("Princess - princess herself")
2758     print_pause("king - Acted against his will, manipulated"
2759                 "by monster!")
2760     print_pause("Shopkeeper - Shopkeeper himself")
2761     print_pause("Host - remains anonymous")
2762     print_pause("Minister - special appearance")
2763     print_pause("Bartender - bartender himself")
2764     print_pause("Monster - actually slain")
2765     print_pause("\n\n")
2766     print_pause("Thanks for playing the game, wait of the next sequel....//")
2767     exit()
2768
2769
2770 # ----- Game Trigger -----
2771
2772
2773 adventure_game()
2774

```

AWESOME

✓ The `pycodestyle` tool reports zero errors and zero warnings. Nice job! 👍

► README.md

RETURN TO PATH

Rate this review

