

Game of Throne Quiz Game

Created by Panisara Sonkom



Application Description



- This app is a Game of Throne quiz game which will display quotes said by characters from Game of Throne and challenge user to pick the correct character



Main Features

1. Allow user to input the amount of question they want
2. Randomise order of questions and answer choices to increase difficulty
3. Allow user to replay the game



Welcome page

Welcome! Are you ready to fight for the Throne?
The game is simple.You'll be given a quote and
ask which character said it. You have four
characteroptions but one chance to guess so
answer carefully!! Goodluck!

```

      /\  /\  /\  /\
    /\  /\  /\  /\  /\
  /\  /\  /\  /\  /\  /\
 /\  /\  /\  /\  /\  /\  /\
/\  /\  /\  /\  /\  /\  /\  /\
\@_^\_@'/  /\_  /\  |  /\  /\
  /\_^\_  /\_  /\  |  /\  /\
( // ) |  /\_/\_/\_/\_/\_/\_/\_/\
( / / ) _|_ / ) //  |  /\  /\  /\
( // / ) '/_ _ _ / ( ; -.  |  _ _ \ \ .~  .~ ~ ~ ^ .
(( / / ) ) ,-{  _  ' . | .~ .  .~  .~  .~  .~  .~  .~  .~
(( // / ) ) ' /\  /  ~ _ . _ .~  .~  .~  .~  .~  .~  .~  .~
(( / / / ) )  ' . {  }  /  .~ ^ .  .~  .~  .~  .~  .~  .~  .~
(( / ) )  .----~ . \  \ _ '  .~  .~  .~  .~  .~  .~  .~  .~
  /// .----. .>  \  \  _ _  _ _  _ _  _ _  _ _  _ _  _ _  _ _
    /// - . _ _ _ _ _ } ^ _ _ _ _ _ ~  _ _ _ _ _  _ _ _ _ _  _ _ _ _ _

```

How many questions would you like? There is a maximum of 15 question:

- Welcome page
- Ask user to input the amount of question they would like



Error Handling: user input

```
How many questions would you like? There is a maximum of 15 question: jfkdl  
Please provide a positive integer value  
How many questions would you like? There is a maximum of 15 question: 16  
There is a maximum of 15 questions  
How many questions would you like? There is a maximum of 15 question: █
```

- Prompt user to re-input number of question wanted if they input an invalid value



Question and Answer Choices Layout

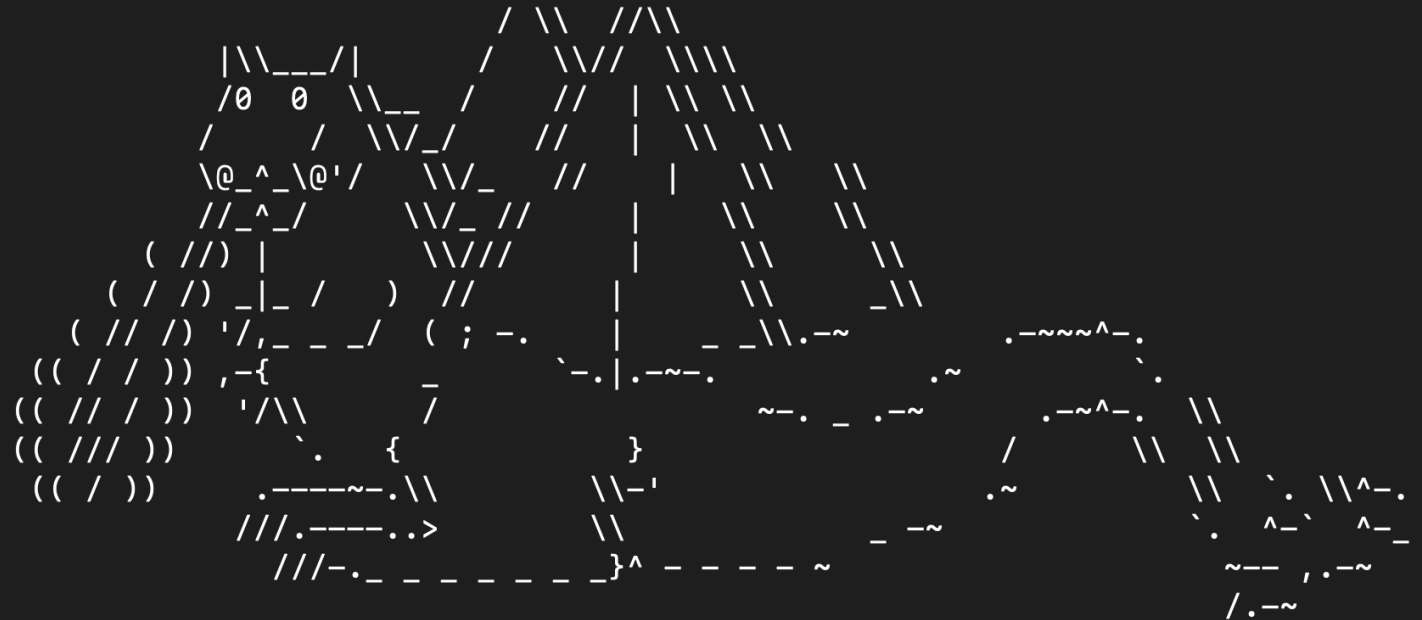
```
How many questions would you like? There is a maximum of 15 question: 15
-----
This quiz will have 15 questions
-----
Question 1:
Who said, 'Let's play a game. Run...'?
-----
A. Joffrey Baratheon
B. Cersei Lannister
C. Little Finger
D. Ramsay Bolton
Enter your answer (A, B, C, or D): █
```

- Display the number of question user inputted
- When a new question pops up the first line will tell user what number of question they are on
- The second line will be the question
- Below will be the list of choices user have
- The bottom line will ask user to input their choice



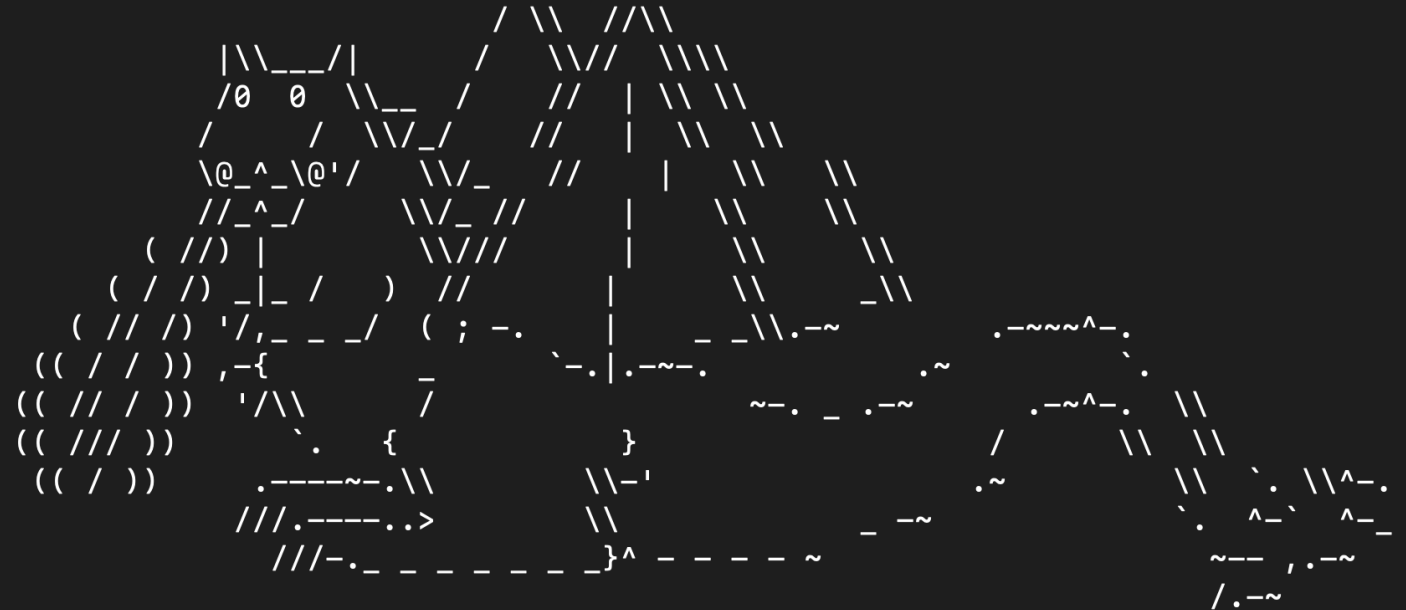
Display Message For When Answer is Correct

| Correct!! You have a good chance on the Throne |



Display Message For When Answer is Incorrect

| Tough luck! The answer was 'Cersei Lannister' |



User Input Invalid Option

```
Enter your answer (A, B, C, or D): 1  
Not an option. Be sure to capitalise your answer. Please choose between A, B, C, or D  
Enter your answer (A, B, C, or D): █
```



Message for Completion of the test



- Congratulation message once user completed the game
- Display how many question they got correct and show score in percentage
- At the bottom a question will display to ask whether or not user wants to play again
- If yes, game will restart
- If no, goodbye message will be print



Code Overview



Question Dictionary

```
1 char_question ={
2     "Who said, 'When you play the game of thrones, you win or you die'?:": [
3         "Cersei Lannister",
4         "Olenna Tyrell",
5         "Tyrion Lannister",
6         "Catelyn Stark",
7     ],
8     "Who said, 'You'll be ruling over a graveyard if we don't defeat the Night King'?:": [
9         "Jon Snow",
10        "Tyrion Lannister",
11        "Theon Greyjoy",
12        "Bran Stark",
13    ],
14    "Who said, 'I will take back what is mine with fire and blood'?:": [
15        "Daenerys Targaryen",
16        "Rhaegar Targaryen",
17        "Rhaenyra Targaryen",
18        "Daemon Targaryen",
19    ],
20    "Who said, 'Your words will disappear. Your house will disappear. Your name will disappear. All memory of you will disappear'?:": [
21        "Sansa Stark",
22        "Jon Snow",
23        "Margaery Tyrell",
24        "Catelyn Stark",
25    ],
26    "Who said, 'When people ask you what happened here, tell them the North remembers. Tell them winter came for House Frey'?:": [
27        "Arya Stark",
28        "Rob Stark",
29        "Lyanna Mormont",
30        "Sansa Stark",
31    ],
32    "Who said, 'An endless night. He wants to erase this world, and I am its memory'?:": [
33        "Bran Stark",
34        "Lord varys",
35        "Melisandre",
36        "High Sparrow",
37    ],
38    "Who said, 'Everyone is mine to torment'?:": [
39        "Joffrey Baratheon",
40        "Ramsay Bolton",
```

- Dictionary for 15 characters in Game of Throne
- **Key:** Question for each character
- **Value:** Answer choices for the character
 - Value is nested in a list as there are multiple alternative choices




Functions

1. `def get_int()` = retrieve user input for number of question they want
2. `def new_game()` = Main structure of the game
3. `def char_prep_question()` = retrieve amount of question wanted and randomize question selected and order
4. `def ask_char_question()` = retrieve answer choices and check answer
5. `def get_char_answer()` = get user's answer and label available choices
6. `def play_again()` = Allow user to play again or quit game



Package used for the game to run

```
source_code >  main.py > ...  
1  import random  
2  from string import ascii_uppercase  
3  import cowsay  
4  from finalcharacter import char_question
```

- import random: To enable randomisation of question order, question selected, and display order of answer choices
- Import ascii_uppercase: to assign label to answer choices and enable randomization of order of answer choice each time user play
- Import cowsay: Decoration and making game fun
- Import char-question: Dictionary location



Def get_int()

```
11 def get_int():
12     '''let user input how many question they want'''
13     return int(input("How many questions would you like? There is a maximum of 15 question: "))
14 if __name__ == '__main__':
15     while True:
16         try:
17             user_number_q_per_game = get_int()
18             if user_number_q_per_game <= 15:
19                 print("-----"
20                       "-----")
21                 print(f"This quiz will have {user_number_q_per_game} questions")
22                 break
23             if user_number_q_per_game < 0:
24                 print("Out of range. Please select between 1-15")
25             else:
26                 print("There is a maximum of 15 questions")
27         except (ValueError, ZeroDivisionError, UnboundLocalError, TypeError) as err:
28             print("Please provide a positive integer value")
29
```

- **Main functionality:** retrieve user input for how many question they want
- **Error:** Message to warn user when they input invalid value such as integer that is out of range or when they enter a string value instead
- The while loop was used so that the question is ask until the user input a valid value



Def new_game()

```
31  ✓ def new_game():
32      '''main game function'''
33  ✓  questions = prep_char_questions(
34      |      char_question, user_number_q_per_game)
35      num_correct = 0
36
37  ✓  for num, (question, alternatives) in enumerate(questions, start=1):
38  ✓      print("-----" \
39      |      "-----")
40      print(f"Question {num}:")
41      num_correct += check_char_answer(question, alternatives)
42      total_score_perc = int((num_correct/num)*100)
43  ✓  print("-----" \
44      |      "-----")
45  ✓  cowsay.dragon(f"You completed the quiz!! You got {num_correct} out " \
46  ✓  |      f"of {num}\n questions. A " +str(total_score_perc) + " %!!" \
47      |      "You can only sit on the\n throne if you got all the question." \
48      |      "So if\n you did, Congrats!!! The Throne is yours!!!\n If not try againg!!")
49  ✓  print("-----" \
50      |      "-----")
```

- **Main Functionality:** Main game structure
- **Print** game question number
 - Used enumerate to number the question and always start from 1
- **Add** number of correct answer to score
- **Display** congrats message and calculate score



Def prep_char_question()

```
52 def prep_char_questions(questions, number_questions):  
53     '''randomize question order and retrieve number of question input by` user'''  
54     number_questions = min(number_questions, len(questions))  
55     return random.sample(list(questions.items()), k=number_questions)  
56
```

- **Main Functionality:** Randomise question order and retrieve inputted amount of question
- This function interacts with `get_int()` and `new_game()` to generate a new order of question everytime a game start
- When user want less than 15 question, this function also pick random and different question in each round



Example of Randomisation

How many questions would you like? There is a maximum of 15 question: 15

This quiz will have 15 questions

Question 1:

Who said, 'Let's play a game. Run...'?:

-
- A. Joffrey Baratheon
 - B. Cersei Lannister
 - C. Little Finger
 - D. Ramsay Bolton

Enter your answer (A, B, C, or D):

How many questions would you like? There is a maximum of 15 question: 3

This quiz will have 3 questions

Question 1:

Who said, 'Never forget what you are, the rest of the world will not. Wear it like armor and it can never be used to hurt you'?:

-
- A. Tyrion Lannister
 - B. Jaime Lannister
 - C. Brienne of Tarth
 - D. Little Finger

Enter your answer (A, B, C, or D):



Def ask_char_question()

```
57 def check_char_answer(question, alternatives):
58     '''randomize choice order and check answer'''
59     correct_ans = alternatives[0]
60     ordered_alternatives = random.sample(alternatives, k=len(alternatives))
61
62     answer = get_char_answer(question, ordered_alternatives)
63     if answer == correct_ans:
64         cowsay.dragon("Correct!! You have a good chance on the Throne")
65         return 1
66     else:
67         cowsay.dragon(f"Tough luck! The answer was {correct_ans!r}")
68         return 0
```

- **Main functionality:** Check user answer with the value retrieved from the character dictionary and randomize order of list of answer
- To enable this, I placed the correct answer of every question as the first element in each character value list
- And assigned that 'alternatives[0]' to the correct_ans
- The if statement is used so that if the user answer correctly it will print 'correct' message and 'incorrect' message if they answer incorrectly



Def get_char_answer()

```
def get_char_answer(question, alternatives):
    '''retrieve answer and randomise choice order'''
    print(f"{question}")
    print("-----" + "\n"
          "-----")
    labeled_choice = dict(zip(ascii_uppercase, alternatives))
    for label, alternative in labeled_choice.items():
        print(f" {label}. {alternative}")
    while (ans_label := input("Enter your answer (A, B, C, or D): ")) not in labeled_choice:
        print("Not an option. Be sure to capitalise your answer." +
              "Please choose between A, B, C, or D")
    return labeled_choice[ans_label]
```

- **Main functionality:** Print the question; retrieve all answer choice for each character and label them
- The reason I used `ascii_upper` case to assign the labels instead of manually inputting is to enable the game to randomise the random order of answer choice displayed
- If label were pre-assigned the choice would not display in alphabetical order
- The while loop was implemented here so that if user input an invalid choice it will ask use again until a valid choice was inputted



Def play_again()

```
83 def play_again():
84     '''let user play again'''
85     user_command = input("Do you want to try and fight for the Throne again? (yes or no): ")
86     user_command = user_command.upper()
87
88     if user_command == "YES":
89         return True
90     else:
91         return False
92 if __name__ == "__main__":
93     new_game()
94
95     while play_again():
96         new_game()
97
98     cowsay.dragon("Thanks for playing!!!")
```

- **Main Functionality:** Allow user to play the game again
- **Error handling:** user can print yes or no in either uppercase or lowercase
- Use if statement to check which condition is met
- Use while loop to restart the game until user type no
- If user doesn't want to play again a goodbye message will be displayed



Challenges

1. Writing and re-writing code to see if it will output the desire functionality
2. Creating the dictionary so that it can be randomise and called correctly
3. Time management: Although I made a clear plan of what to do and set date for each task to be complete by, I went overtime for all tasks as I had tto re-write code several time before the game came together
4. Pytest: Instead of working on it work to test whether each functionality work, I did it last which caused many revision

