

Star Wars Terminal App presentation



```
PROBLEMS  DEBUG CONSOLE  TERMINAL  OUTPUT
(venv) roger@mitchells-MBP Star_wars_app % python3 main.py
```

STAR

WARS

ADVENTURE By M ROGER

Hello there! what is your name?: █

By Mitchell Roger

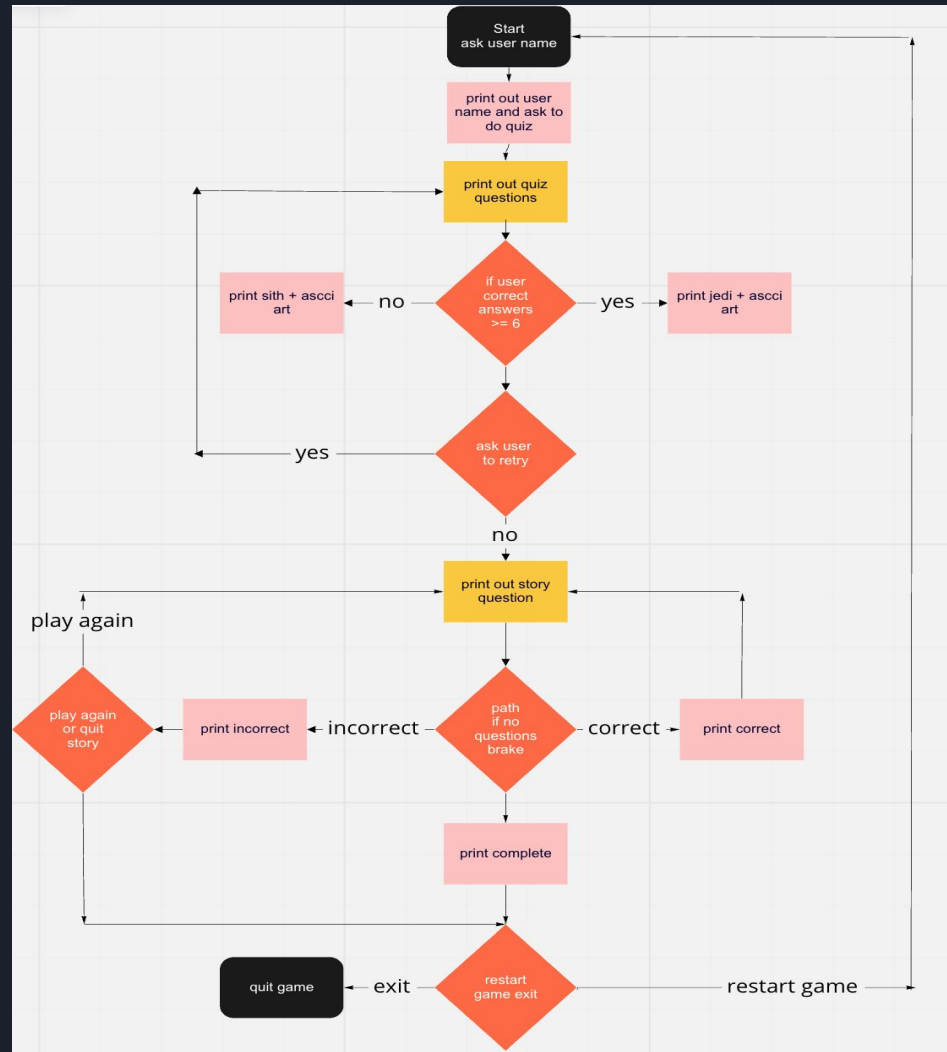


Star Wars adventure game

Features

- After completing the quiz the app will tell you if you are a jedi or sith by counting how many correct answers you have guessed
- Depending on if you are a jedi or a sith will display the corresponding ascii art
- There is a jedi quest and a sith quest that is called depending on what you get in the quiz
- A restart feature is set at the end of the quiz and at the end of the game

Flow chart



Star Wars adventure game

Code walk through

- In this screen shot you you have the for loop that iterates over the questions in a dictionary and append them into a empty list

```
def __init__(self, question_data):  
    """runs a for loop over the questions and apend a new question in a empty list  
  
    Args:  
        | question_data (str): dictionary with question  
        |  
    """  
  
    self.question_number = 0  
    self.counter = 0  
    self.question_list = []  
  
    for question in question_data:  
        question_text = question["question"]  
        question_answer = question["correct_answer"]  
        wrong_answers = question["incorrect_answers"]  
        new_question = Question(  
            | question_text, question_answer, wrong_answers)  
  
        self.question_list.append(new_question)  
        # empty list to store questions
```

Star Wars adventure game

Code walk through

- This prints out the question number and the question then it asks the user to enter a or b

```
def next_question(self):
    """adds one to the question number, gets input.
    checks for valid input and displays question and question number
    """

    current_question = self.question_list[self.question_number]
    self.question_number += 1

    while True:
        print(f"\n[blue]Q:{self.question_number}/10[/]\n")
        user_answer = console.input(
            f"{current_question.text}\n[blue]Enter A or B:\n")
        console.rule("")
        user_answer = user_answer.upper()

        if user_answer != "A" and user_answer != "B":
            console.print(
                f"[red][{user_answer}]*INVALID INPUT*[/]Enter A or B:")
            continue
        else:
            break
```

Star Wars adventure game

Code walk through

- The counter is how the quiz decides if the user is a jedi or sith if the counter is over half of the total number of question it will print out you are a jedi

```
def check_answer(self, user_answer, correct_answer, wrong_answers):  
    """checks user input matches with correct answer  
  
    Args:  
        user_answer (str): user input  
        correct_answer (str): correct answer  
        wrong_answers (str): wrong answer  
    """  
  
    if user_answer == correct_answer:  
        self.counter += 1  
        print(f"{self.counter}")
```

```
if self.question_number == 10:  
    if self.counter > 6:  
  
        print(  
            Padding(  
                f"You have completed the quiz  you are a [i]JEDI[i]",  
                (2,  
                 50),  
                style="bold on green",  
                expand=True,  
            ))  
        yoda()
```

Star Wars adventure game

Code walk through

- I created a Story class that inherits the attributes from the questionnaire to cut down duplicate code
- This is where we notice how classes are very useful instead of creating a whole new section for the sith quest i created a sith class that inherited everything from the story class. i just had to modify the final function and that was it

```
Story(Questionnaire):  
    """Constructs all the necessary attributes for story  
  
    Args:  
        Questionnaire (class): Story inherits attributes from Questionnaire  
    """  
  
    def check_answer(self, user_answer, correct_answer, wrong_answers):  
  
        if user_answer == correct_answer:
```

```
class Sith(Story):  
    def final(self):  
        if self.counter == 5:  
  
            sithfinish()  
            self.restart()
```

Star Wars adventure game

Code walk through

- This is how i get the story quest to reset the question and start again by if the question is wrong self.counter is = 0

```
def check_answer(self, user_answer, correct_answer, wrong_path):  
  
    if user_answer == correct_answer:  
        self.counter += 1  
  
        print(Padding("correct", (1, 1), style="on green", expand=False))  
  
    else:  
        self.counter = 0  
        die()  
        print(  
            Padding(  
                f"[black] ❌ ❌ ❌ [{wrong_path}] [black] ❌ ❌ ❌",  
                (2, 40),  
                style="bold black on red",  
                expand=True,  
            )  
        )  
    )
```




Star Wars adventure game

Code walk through

- This is how the quest finishes when the counter is = 5 which is the last question it will end

```
def final(self):  
    if self.counter == 5:  
        jedi_end()
```

```
class Sith(Story):  
    def final(self):  
        if self.counter == 5:  
            sith_end()
```

Star Wars adventure game

Code walk through

- For error handling i created a while loop that if the user inputs anything other than a or b you can't continue
- The second code snippet is the input is for when you enter your name if you don't enter anything it will say no input and it will keep asking until you enter your name

```
while True:

    replay_quiz = console.input(
        "[yellow]continue or rerty[/]\nA. continue\nB. Retry ")
    replay_quiz = replay_quiz.upper()
    if replay_quiz != "A" and replay_quiz != "B":
        console.print(
            f"[red]*INVALID INPUT*[/] Enter A or B:[/]"
        )
        continue
    else:
        break
```

```
while True:
    name = console.input("[yellow>Hello there! what is your name?: ")
    name = name.upper()
    if name == "":
        console.print(f"[red][{name}] **NO INPUT**[/] enter you name ")
    else:
        break
```



Star Wars adventure game

Review of development process

- It was my First time building a terminal app using classes, so I found it hard to understand, but it felt like it was finally starting to click near the end.
- I enjoyed working on the project and was happy with the final result.
- My favourite part was making the quest stories and the overall aesthetics of the app.