Terminal Application

T1_A3

Kelsey Hardy

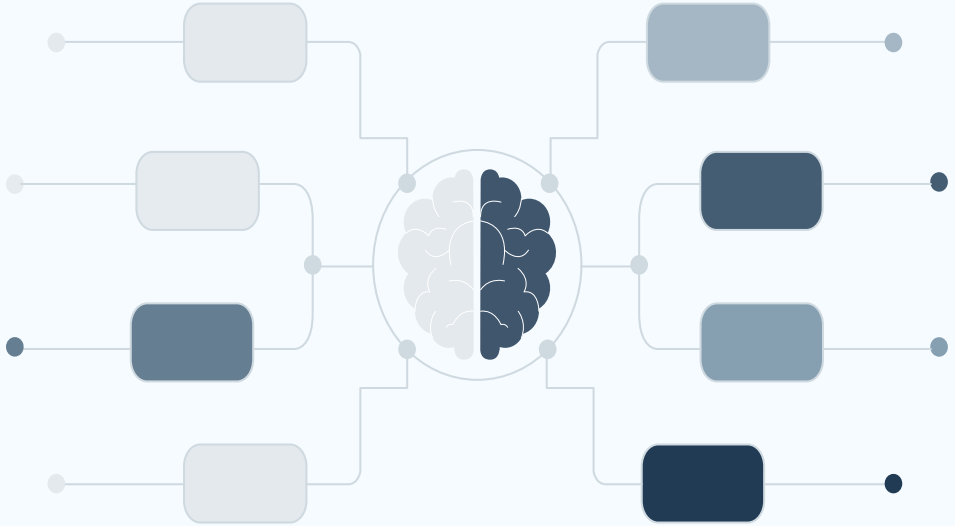
Walk-Through

Concept	The main idea for this Terminal Project
Features	What features and functions are used to create this application
Logic	The use of code for conditionals
Review of Development	The build process: challenges, ethical issues, favourite parts
Overview of App	The overall look and function
Overview of Code	The code and styling within the source folder

01

Concept

Let’s Make Bank

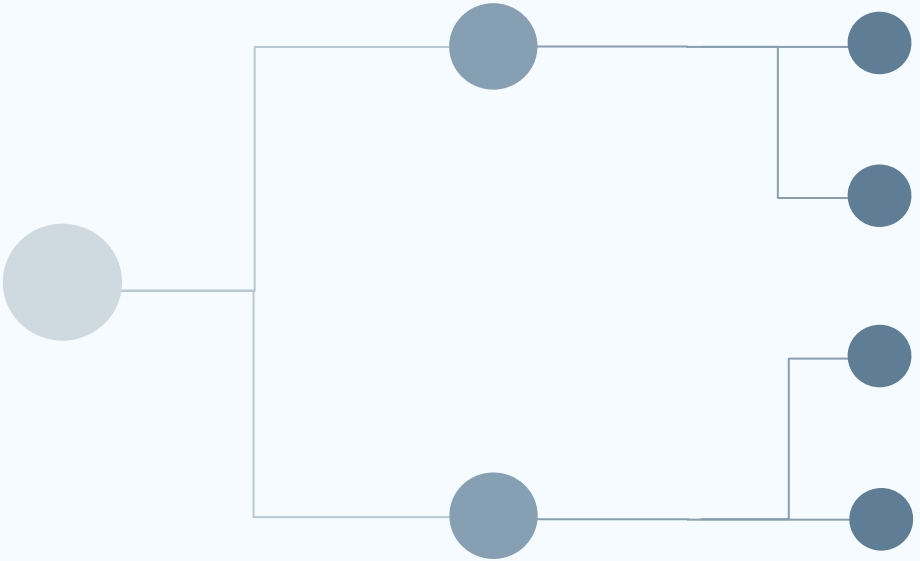


Concept

The idea for this terminal application is based off the tv series, Who Wants to Be a Millionaire in the fact that it functions similarly to a quiz show.
The user is introduced at the start (via user input) and then chooses a category to play a series of 5 questions. Each question has 4 answers to choose from. If answered correctly, the next question is asked. If not, the game ends.

02

Features



Features

Menu Function

In which the user chooses the category to play.

Question Function

To play through each question and determine correct answers.

User_input

For player input to personalize the gameplay.

User Progression

A tracking system to ensure the user wins after 3 category plays.

Randomization

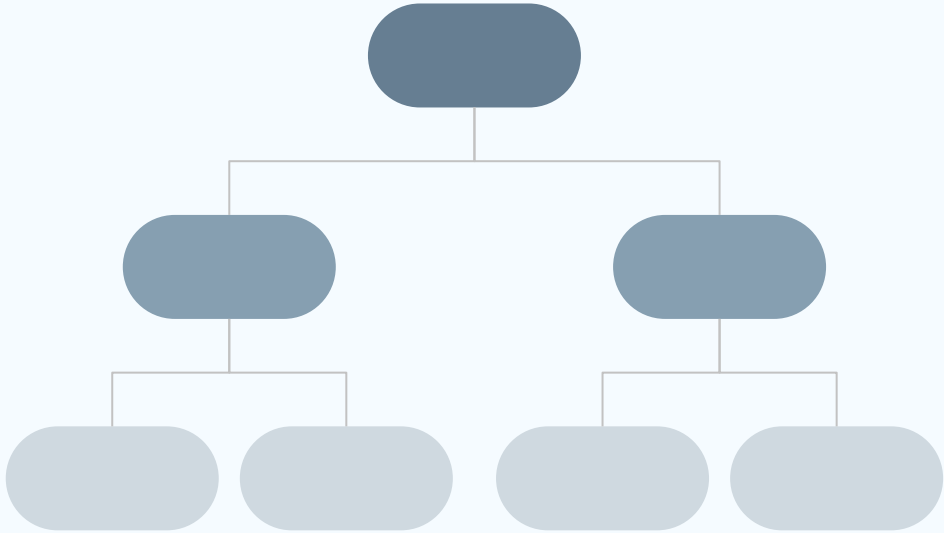
The ability to shuffle the questions to keep them fresh each playthrough.

Error Handles

To make the user aware if any incorrect inputs are entered.

03

Logic



Logic

Things that have been **utilized** to make the code **functional** and **formally correct**

01 Control Flow

Boolean Value for menu function and then directional flow through conditionals.

02 While and For Loops

Menu and Questions make use of this to keep the flow within the loop and only diverse when positioned to.

03 If Statements

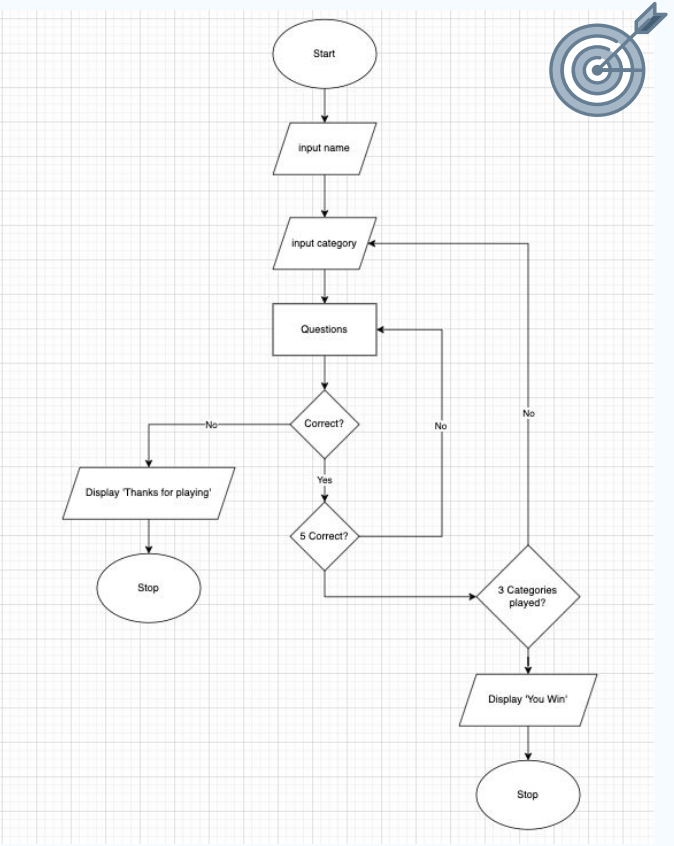
These are used to assist in the control flow and create conditionals for the code.

04

Review of Development

Flowchart for Code

This is a flowchart to show the conditions for the code and how it is controlled upon execution.

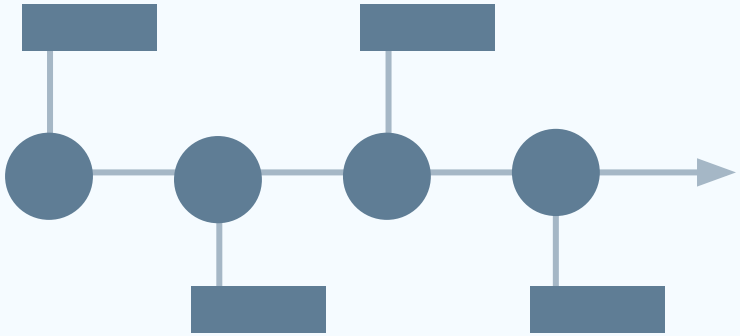


Development/Build Process

Challenges	Trying to link the python files for questions
	Making the Questions work
Ethical Issues	Physical Accessibility
	Accuracy
Favourite Parts	The Game Host!
	Importing ‘colored’ python package

05

Overview of Application



Overview of Application

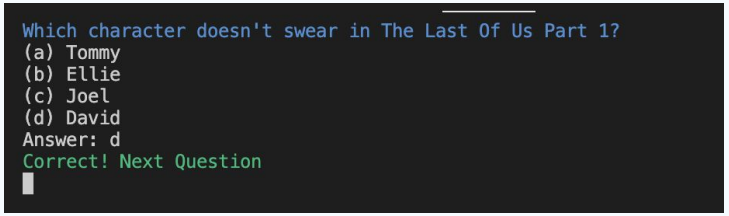
Opening Sequence

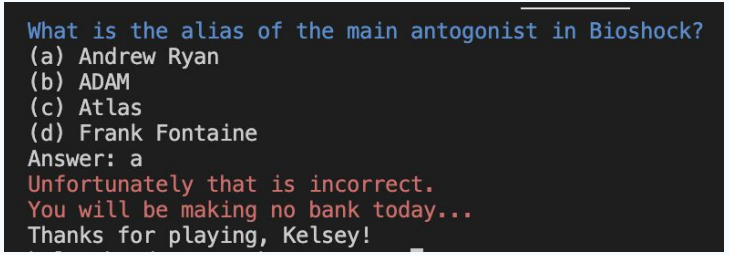
Upon starting, the title is displayed in ascii style featuring colour.
Pressing Enter prompts a user name and welcomes them to the game.
The game host (also in colour) will at first introduce the categories, this becomes a simple prompt later in the application.
To choose a category, the code is written to expect one of these words as an input or it will request inputting again.

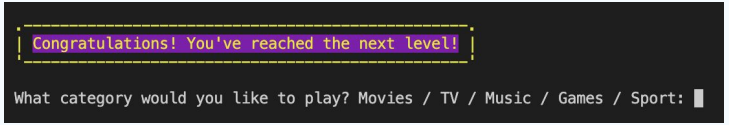
Overview of Application

Trivia Questions

The questions require a letter input for answers.
Only a, b, c and d are valid inputs.
A statement will print out informing the user as to whether they are correct or not.
Colour has been implemented to emphasise the response.
A small ascii display prints on successful category completion.







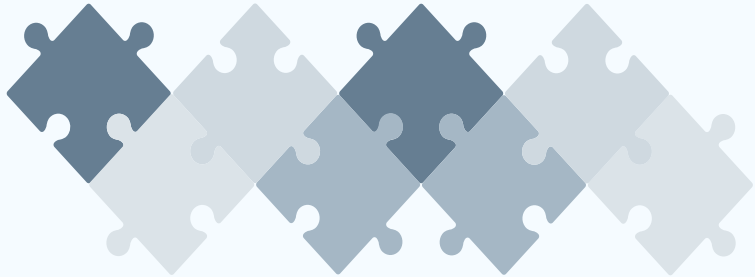
Overview of Application

Winning Sequence

If the 3 categories are answered correctly, the application will display the ‘Winner’ ascii in green colouring.
The game host is seen for the last time in a close off and with a thank you message.

06

Overview of Code



Overview of Code

```
import os
import random
import sys
import time
from colored import fg, bg, attr
```

Imports and Start

```
def print_slow(str):
    for letter in str:
        sys.stdout.write(letter)
        sys.stdout.flush()
        time.sleep(0.08)


def press_start():
    print(f"""{fg('gold_3b')}
 888              888    d8b              888b     d888          888                   888888b.                    888
 888              888    88P              8888b   d8888          888                   888  "88b                   888
 888              888    8P               88888b.d88888          888                   888  .88P                   888
 {fg('gold_1')}888      .d88b.  888888 "   .d8888b       888Y88888P888  8888b.  888  888  .d88b.       8888888K.   8888b.  88888b.
 888     d8P  Y8b 888       88K           888 Y888P 888     "88b 888 .88P d8P  Y8b      888  "Y88b     "88b 888 "88b 888 .88P
 888     88888888 888       "Y8888b.      888  Y8P  888 .d888888 888888K  88888888      888    888 .d888888 888  888 888888K
 {fg('yellow')}888     Y8b.     Y88b.          X88      888   "   888 888  888 888 "88b Y8b.          888   d88P 888  888 888  88
 88888888 "Y8888   "Y888     88888P'      888       888 "Y888888 888  888  "Y8888       8888888P"  "Y888888 888  888 888  888


 {attr('reset')}""")
    print_slow("Press Enter to Start...")
    input()
```

Imported from Python Library and packages to assist in functionality i.e. incorporating colour with ‘colored’.
Created a slow printing function to create a ‘game-like’ opening effect using ‘time’ and ‘sys.
Imported ‘art’ package to create ascii for visual effect.
The import of ‘os’ allowed for the ability to clear the terminal after certain aspects of the gameplay.

Overview of Code

```
def category_menu():
    correct_categories = 0

    while True:
        if correct_categories >= 3:
            print(f"""{fg('71')}
888        888 d8b                                    888
888    o   888 Y8P                                    888
888   d8b  888                                        888
{fg('2')}888 d888b 888 888 88888b.  88888b.   .d88b.  888d888 888
888d88888b888 888 888 "88b 888 "88b d8P  Y8b 888P"   888
88888P Y88888 888 888  888 888  888 888888888 888     Y8P
{fg('29')}8888P   Y8888 888 888  888 888  888 Y8b.     888      "
888P     Y888 888 888  888 888  888  "Y8888  888     888
{attr('reset')}""")
            input("You've completed 3 rounds with every answer correct!\nPress Enter:")
```

Menu Function

Above shows the user progress feature and ultimate win page.
Boolean value is added to determine user progress.
‘If’ statements are created to direct into question loops for each category.
Code designed to handle user input i.e. capital lettering with use of ‘.lower()’ and ‘else:’ for invalid inputs.

```
user_decision = input(
    "What category would you like to play? Mov

if (user_decision.lower() == "games"):
    os.system('clear')
    ask_questions(game_questions)
    Helper.congrats()
    correct_categories += 1
    continue

if (user_decision.lower() == "movies"):
    os.system('clear')
    ask_questions(movie_questions)
    Helper.congrats()
    correct_categories += 1
    continue

if (user_decision.lower() == "tv"):
    os.system('clear')
    ask_questions(tv_questions)
    Helper.congrats()
    correct_categories += 1
    continue
```

Overview of Code

```
def ask_questions(questions):
    random.shuffle(questions)
    for question in questions[0:5]:

        answer = ""
        while answer.lower() not in ["a", "b", "c", "d"]:
            answer = input(question.prompt)
            if answer == question.answer:
                print(f"{fg('green')}Correct! Next Question{attr('reset')}")
                input()
                os.system('clear')
            elif answer != question.answer and answer.lower() not in ["a", "b", "c", "d"]:
                os.system('clear')
                print(
                    f"{fg('214')}Oi mate, that's not a real answer!{attr('reset')}")
            else:
                print_slow(
                    f"{fg('red')}Unfortunately that is incorrect.\nYou will be making no bank today...{attr('reset')}")
                input()
                exit(f"Thanks for playing, {your_name}!")
```

Trivia Questions Function

Wrapped in a For and While Loop.
‘For’ makes sure to only ask 5 questions in each category.
‘While’ helps to eliminate invalid input.
‘If’ statements create conditionals for correct and incorrect answers as well as user input.

Overview of Code

```
question_prompts = [
    f"{fg('114')}How long is a quarter of an NBA game?{attr('reset')}\n(a) 20 minutes\n(b) 12 minutes\n(c) 15 minutes\n(d) 10 minutes\nAnswer: ",
    f"{fg('114')}In what sport do they play off for the Stanley Cup?{attr('reset')}\n(a) NFL\n(b) Golf\n(c) EPL\n(d) NHL\nAnswer: ",
    f"{fg('114')}Who has the most superbowl titles in NFL history?{attr('reset')}\n(a) Tom Brady\n(b) Dallas Cowboys\n(c) Pittsburgh Steelers\n(d) New England Patr
    f"{fg('114')}How many players are on the pitch at one time in a game of football/soccer?{attr('reset')}\n(a) 10\n(b) 11\n(c) 12\n(d) 13\nAnswer: ",
    f"{fg('114')}What is the record for the most points scored by a single player in a game of NBA?{attr('reset')}\n(a) 71\n(b) 91\n(c) 100\n(d) 81\nAnswer: ",
    f"{fg('114')}In what year did the VFL become the AFL?{attr('reset')}\n(a) 1988\n(b) 1989\n(c) 1990\n(d) 1991\nAnswer: ",
    f"{fg('114')}The first ever FIFA world cup was won by which country?{attr('reset')}\n(a) England\n(b) France\n(c) Brazil\n(d) Uruguay\nAnswer: ",
    f"{fg('114')}How old was Tiger Woods when he won The Masters for the first time?{attr('reset')}\n(a) 21\n(b) 19\n(c) 17\n(d) 23\nAnswer: ",
    f"{fg('114')}In NFL, a touchdown is worth how many points?{attr('reset')}\n(a) 5\n(b) 6\n(c) 7\n(d) 8\nAnswer: ",
    f"{fg('114')}What is Canada's national sport?{attr('reset')}\n(a) Ice Hockey\n(b) Shooting\n(c) Handball\n(d) Lacrosse\nAnswer: "
]

sport_questions = [
    Question(question_prompts[0], "b"),
    Question(question_prompts[1], "d"),
    Question(question_prompts[2], "a"),
    Question(question_prompts[3], "b"),
    Question(question_prompts[4], "c"),
    Question(question_prompts[5], "c"),
    Question(question_prompts[6], "d"),
    Question(question_prompts[7], "a"),
    Question(question_prompts[8], "b"),
    Question(question_prompts[9], "d")
]
```

Trivia Questions Lists

Tried multiple ways to create these variables.
Lists had the most success when testing my code, and output matched each time.
Created a Question class to put the list of questions to answers as well as randomizing the indexes.

Thanks!

CREDITS: This presentation template was created by **Slidesgo**, and includes icons by **Flaticon**, and infographics & images by **Freepik**