T1A3 –Terminal Application
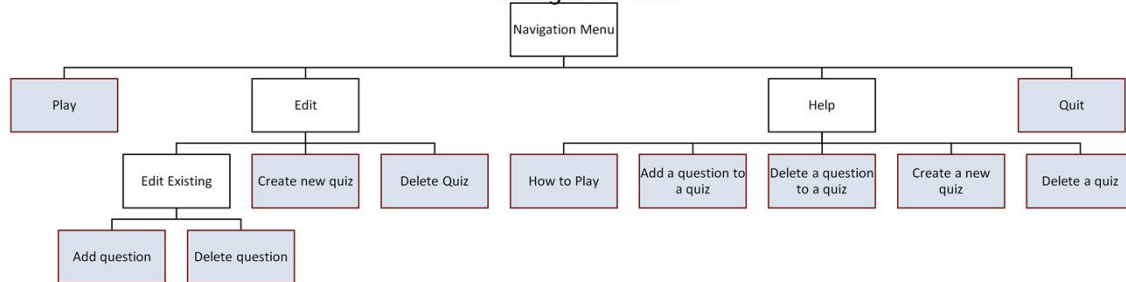
Dirk Booy

Walkthrough of Terminal Application
Menu & navigation

The primary navigation through the application functions will be through a series of menus

Navigation chains for the program will be as follows:
* Play quiz
    * Topic selection --> start quiz
* Edit quiz
    * edit existing
        * select topic
            * add question --> add question function
            * remove question --> remove question
    * Add new quiz topic --> start new quiz & question add function
    * Delete a quiz --> delete quiz file
* Help
    * How to play --> displays help topic
    * How to Add a question to a quiz,--> displays help topic
    * How to Delete a question to a quiz,--> displays help topic
    * How to Create a new quiz,--> displays help topic
    * How to Delete a quiz,--> displays help topic
    * How to edit --> displays help topic
* Quit --> terminates program

Walkthrough of Terminal Application
Play a Quiz!

Example of Results

Quiz selection

```
############################# Select a quiz #############################
-----------------------------------------------------------------------
Please select a Quiz topic:
1. 50 Test Questions
2. Broken Quiz
3. Caitlans Music Quiz
4. Test Questions

Select topic number:
```

Example of question

```
######################### Caitlans Music Quiz #########################
-----------------------------------------------------------------------
Question 2: In what Country was Alanis Morrisette born?
A: France
B: Canada
C: England
D: America
Enter your selection: 2
Correct! You answered in 3 seconds and get 8 points
2
Press Enter to continue...
```

```
############################## Results ##############################
-----------------------------------------------------------------------
You got 5 / 10 correct, scoring 30 points!

You got the following question wrong:
Question 1:
As voted by Triple J listeners what song came number 1 in the hottest 100
 of all time in 2009?
The correct answer was:
Smells like teen spirit - Nirvana

Question 3:
David McCormack the voice of popular Cartoon dog Bandit Heeler from the T
V show Bluey is the frontman of which Australian Band?
The correct answer was:
Custard

Question 6:
Crowded House formed in which Country?
The correct answer was:
Australia

Question 7:
Which 'Roger' was in British band; The Who?
The correct answer was:
Roger Daltrey

Question 10:
In what year was the first Woodstock held?
The correct answer was:
1969

Press Enter to continue...
```

3

The Main Feature of the app is playing the quiz.
- When you select play quiz you will be presented with the list of available quiz topics (which correspond to csv files stored in the quiz_data folder).
- When you select a topic you will be asked a series of 10 questions, randomly selected from the quiz file
- You get points for answering the question correctly & bonus points for faster answers
- At the end of the 10 questions your results will be displayed including a list of the questions you got wrong and their correct answers so you can review them.

Edit existing quiz features
- On selecting edit existing quiz the user is presented with the list of quizzes available and asked to select the one they want to edit

Add a question:
- Allows user to add a question to an existing quiz
- Guides user through process step by step & gets positive confirmation before saving

Delete a question
- Allows user to delete a question from file
- Displays numbered list of questions in topic
- On selection will display question and answers for selected question and ask for confirmation

All changes will be saved to file and persist after application has closed

Walkthrough of Terminal Application
Add a new quiz or delete a quiz

```
######################## New Quiz ######################
----------------------------------------------------------
Note: Quiz title must be unique, and cannot contain s
pecial characters
Please enter quiz title: New Quiz Example
----------------------------------------------------------
################# Question Preview ################
----------------------------------------------------------
Question 1: This is the first question
A: This answer is also wrong
B: This is the answer to the first question
C: This is not the answer either
D: This is not the answer

The correct answer is:
B: This is the answer to the first question

Do you want to add this question? [Y/N]: y
Question added!

Would you like to enter another question?y
----------------------------------------------------------
########### Success! New quiz was created ###########
----------------------------------------------------------
Press Enter to continue...
```

```
----------------------------------------------------------
################ Delete quiz ################
----------------------------------------------------------
Select quiz to delete
1. 100 Test Questions
2. 50 Test Questions
3. Caitlans Music Quiz

Select topic number:1

WARNING THIS ACTION CANNOT BE UNDONE!

Are you sure you want to delete quiz: 100 Test
Questions? [Y/N] :
```

Create a new quiz
- Guides the user through the process of creating a new quiz
- Will ask if they want to enter another question every time they finish a question
- Can enter as many questions as they like
- Saves quiz to file

Delete quiz
• Will present the list of quizzes saved in the directory
• User can select one to delete.
• Will prompt user for confirmation and give warning that this cannot be undone.

Walkthrough of Terminal Application
Help


```
---------------------------------------------------------------
############################# Help #############################
---------------------------------------------------------------
What do you need help with?
1. How to Play
2. Add a question to a quiz
3. Delete a question to a quiz
4. Create a new quiz
5. Delete a quiz
6. Return to menu

Enter selection:
```

```
---------------------------------------------------------------
######################### How to quiz! #########################
---------------------------------------------------------------

To play a quiz select option 1. Play from the main menu. You will be
presented with a list of the available quizzes. select the option
number the quiz you would like to play to begin a round.

A round will consist of 10 questions (or less if there are not 10
questions saved in the quiz file selected). Each question will be
presented with 4 multiple choice answers. To select an answer type its
corresponding letter and press enter (answers can be in upper or lower
case). Alternatively you can use numbers instead of letters ie: 1 for
A, 2 for B, 3 for C and 4 for D.

Once you have submitted your answer, the correct answer will be
revealed. You will receive points for correct answers and your total
will be displayed at the end of the round.

Note: Questions are selected randomly from the pool of questions for
the topic, and the order of the multiple choice answers is randomised
each round. So every round should be different.

Press Enter to continue...
```

Help menu displays a list of topics.
When selected the help text for that item will be printed to the console.
After a help file is displayed the user is returned to the help menu, where they can select a another topic or return to the main menu.

Functions
- menu_select(menu_items: list)
- main_menu()
- select_topic(topic_list: list)
- play_menu(topic_list)
- edit_menu(topic_list)
- help_menu()

Modules used
- pyinputplus
  - used for simple user input validation
- getpass
  - used to hides user input in terminal
- os
  - used to get directory path
- json
  - used to read help text from file

menu_select(menu_items: list)
Gets valid user selection from menu. Return index of selection.
Input: list of menu items (list)
Output: index of select menu item (int)
Process
Prints list to terminal in a numbered format
Requests selection as input from user, loops until a valid input is given. Validity check is CAPS insensitive. Valid selections are:
- Full menu name: str
- First word: str
- First letter: str
- option number: int(str)

Converts the valid selection into the index position of the menu item selected and returns it.
Note: this could have been done using the pyinputplus module, however I chose to build this function to give expand the valid answer options for the user (to provide for users who do not read the manual)

main_menu()
Loop which controls the main menu navigation:
Input: none
Output: none
Process
This is the main loop of the program, allows navigation to the rest of the application functions

select_topic(topic_list: list)
Input: list of quiz titles
Output: index of selected list
Process
- Prints out list of available quizzes to terminal
- Requests valid user input to select quiz
- Converts selection to index of selection
- Returns index

play_menu(topic_list)
Input: list of quiz topics
Output: none
Process
- Passes topic list to select_topic() function to get the user to select a quiz to play
- Passes the selected topic to the playquiz.quiz_round() function to start the quiz

edit_menu(topic_list)
Runs through the edit mode menu tree, and calls the function requested by the user
Input: list of available quiz titles / topics
Output: None
Process
Presents the edit mode menus and calls the edit functions in the file_handling.py as requested by the user input. Input list is used to select which quiz to edit.

help_menu()
Displays Help menu & prints selected help topic to the terminal
Input: None
Output: None
Process
- Reads help topics from file into dictionary
- Calls menu selector with help menu items list
- Prints selected help topic to terminal or returns to main menu (if return is selected)
- Loops back to help menu

Functions

- clear_screen() -> None:
- print_title(title: str) -> None:
- question_selector(topic_questions: list) -> list:
- ask_question(question: list, q_num: int) -> int:
- get_valid_answer() -> int:
- quiz_round(topic: str):

Modules used:

- random
  - used to randomise question selection & shuffle answers
- os
  - used to detect Operating System & clear terminal screen
- time
  - used for time keeping for scoring
- getpass
  - used to hide user input for "press enter to continue" breaks
- shutil
  - get_terminal_size() function to determine screen width

8

Description of functions

clear_screen() -> None:
Purpose: Clears the terminal screen
Input: none
Output: none
Process:
- Checks the operating system (for future compatibility with windows os)
- Uses the appropriate os method to clear the terminal screen

print_title(title: str) -> None:
Purpose: Prints a decorative header to the terminal
Input: header title **(string)**
Output: None
Process
- calls clear screen function to clear the terminal
- Prints a standard decorative header to the terminal using the header title string
- Adds decorative lines of "-" above and below and "#" padding the title to center it
- Function calculates the current width of the terminal screen and makes the title to fit (upto a maximum of 100 characters wide)

question_selector(topic_questions: list) -> list:
Purpose: Selects 10 random questions from the questions list of questions it is given
Input: long list of questions
Output: list of 10 randomly selected questions form input list
Process
- Creates an empty output list
- For 10 loops:
  - Generates a random number to select a question in the input list
  - Removes question from the input list and adds to the output list (to prevent repeating questions)
- If the input list is less than 10 it will return all questions in a random order

ask_question(question: list, q_num: int) -> int:
Prints a question and its multiple choice answers to the terminal (answer options are in a random order)
Input: A list containing 5 strings: 1 question, 1 correct answer then 3 incorrect answers
Output: returns the index / position of the correct answer as displayed on the screen
Process
- Copies answers into new list
- Shuffles list & confirms position of correct answer
- Prints to terminal question and shuffled answers in the form correct form ie: A. ans1, B. ans2, etc
- Returns position of correct question

get_valid_answer() -> int:
Gets a valid multiple choice answer from the user and returns it in the correct form
Input: None
Output: position of answer selected by user
Process:
Prompts user for input, will only accept [1,2,3,4,A,B,C,D,a,b,c,d]
Uses a loop, so if an incorrect input is given user will be prompted to input a valid selection until a valid answer is given.
Answer is converted into an integer coresponding to the position of the selected answer and returned

quiz_round(topic: str):
Function handles the quiz round – calls other functions when required, keeps a tally of the users score, and presents results on conclusion of quiz
Input: quiz name (string)
Output: None
Process
- Calls file_handling.get_question_list_from_file() to get the list of questions from the file related to the input topic
- Pass list of questions to question_selector() function to get quiz round questions list
- Loops through the quiz round questions list and
    - Calls ask_question to display question
    - Calls get_valid_answer() to get the user answer (measures time between calling get_valid_answer() and the return from the function)
    - Compares the returns from ask_question & get_valid_answer – if they are the same the question was answered correctly
    - Calculates score if correct, records question in list if incorrect
- At end of loop displays results – prints out final score & list of incorrect questions (if any) and their correct answer to terminal

Modules used:
- random module used to randomise question selection & shuffle answers
- os module used to detect Operating System & clear terminal screen
- time module used for delay timing
- getpass module used to hide user input for "press enter to continue" breaks
- shutil module get_terminal_size() function to determine screen width

get_topics_from_directory()
get_question_list_from_file(quiz_title: str)
get_new_question_from_user_input()
write_new_question_to_file(topic: str)
delete_question(topic: str)
new_quiz_name(topic_list: list)
new_quiz_topic(topic_list: str)
write_full_quiz_to_file(topic: str, question_list: list)
delete_quiz_file(topic: str)

- import csv
  - used to read / write quiz question files.
- os
  - used to get directory list.
- pyinputplus
  - used to get valid user inputs.
- getpass
  - used to hide user input when asked to press enter to continue.

Description of function
get_topics_from_directory()
Input: none,
Output: returns list of quiz topics
Process
- Reads all the files in the ./quiz_data folder into a list
- filters the list to only include quiz files, which are saved in the format **quiz_[title_of_quiz].csv**
- Converts the file names into quiz names eg: the above would become "*Title Of Quiz*"
- Sorts the list into alphabetical order
- Returns list of quiz topics

get_question_list_from_file(quiz_title: str)
Input: string containing the quiz name
Output: list of questions (list)
Process
- Converts quiz name into a file name
- Retrieves all questions from that file
- Validates that the questions in the file are in the correct format, will remove any in incorrect format from list and display a warning message to the user that the file may be corrupted and not work properly.
- Returns list of questions

get_new_question_from_user_input()
Prompts user through writing a question.
Input: None
Output: New question format [Q,CA,WA,WA,WA]
Process
- Prompts user for:
  - Question
  - Correct answer
  - Wrong answer 1
  - Wrong answer 2
  - Wrong answer 3
- Prompts are validated using pyinputplus inputStr module, which reuires a none blank string input
- Displays question to user by calling playquiz.ask_question() function
- Prints the correct answer.
- Asks user for confirmation the question is correct (validated using the pyinputplus YesNo module)
  - If question is incorrect, prompt if user wants to restart question writing process
  - If they don't raises a Value Error – Add question process cancelled, wihich will need to be handled by the calling function
- If user confirms question is correct – returns the question & answers in list format [Q,CA,WA,WA,WA]

write_new_question_to_file(topic: str)
Get question & quiz topic, adds question to the quiz topic file
Input: quiz title / topic (string)
Output: none
Process
- Gets question by calling get_new_question_from_user_input() function
    - Handles Value error by printing message to screen and returning to main menu
- Converts quiz title / topic to file name
- Opens file and appends question in list format to file
- Prints message to terminal & returns user to main menu

delete_question(topic: str)
Presents list of questions in quiz to user, deletes selected question
Input: quiz title / topic (string)
Output: None
Process
- Passes quiz title / topic to get_question_list_from_file() function to gegt alist of all questions in quiz
- Prints a numbered list of the questions to terminal
- Get user input - selection of question to delete (validated by pyinputplus inputInt module (input must be integer in range of length of questions list)
- Print selected question to terminal (using playquiz.ask_question() function)
- Get confirmation of deletion (validated using the pyinputplus YesNo module)
- On confirmation remove question from question list
- Use write_full_quiz_to_file() to overwrite the old quiz file with the current question list (with the question removed)
    - If there are no questions left in the questions list the quiz file will be deleted
- Prints confirmation message and returns user to main menu

new_quiz_name(topic_list: list)
Gets a valid new quiz name from user input
Input: list of quizzes / topics (list)
Output: quiz name (string)
Process
- Prompts user for the title of the new quiz & valdidates it is:
    - Not an existing quiz name
    - Not blank
    - Does not contain special characters
- Will loop until valid name is entered (will prompt user why the name was rejected)
- Returns valid name

new_quiz_topic(topic_list: str)
Guides user through process of creating new quiz
Input: list of quizzes / topics (str)
Output: none
Process
- Gets new quiz name by calling new_quiz_name() function
- Will loop until user declines to continue:
    - get_new_question_from_user_input()
    - If question is returned Append new question to question list
    - Ask if user would like to add new question
        - If no: break loop
- If question list is empty cancel process and return to main menu
- Write questions to file by passing quiz title and question list to write_full_quiz_to_file()

write_full_quiz_to_file(topic: str, question_list: list)
Takes quiz name and question list and writes them to a new quiz file
Input: quiz name(string), list of questions (list)
Output: None
Process
- Converts quiz name to filename
- Writes to filename:
    - Header rows
    - Question list

delete_quiz_file(topic: str)
Deletes file from directory
Input: Quiz name
Output: None
Process
- Converts quiz name back to a file name
- Deletes file

test_read_write_file():
This test will:
1. Create a test quiz,
2. write it to a file,
3. **assert** that the newly generated quiz appears in the topics list
4. Read the quiz from file back into the program and **assert** it is the same as the original generated quiz.
5. Delete the generated quiz file
6. **assert** the final topics list equals the original topics list and therefore all changes have been reverted

It will test the following functions in file_handling.py:
1. write_full_quiz_to_file()
2. get_topics_from_directory()
3. get_question_list_from_file()
4. delete_quiz_file()

test_edit_file(monkeypatch)
This test will:
1. Get the list of quiz topics
2. Write a test question to the first quiz file
3. Load questions from quiz file
4. **assert** the test question is in the new questions list
5. Delete test question from file
6. **assert** test question is no longer in the questions list, confirming deletion of test question form file

It will test the following functions in file_handling.py:
1. get_topics_from_dir()
2. write_new_question_to_file(topic)
3. get_question_list_from_file(quiz_title)
4. delete_question(topic)

10

2 Automated tests have been written to test the file handling functionality in the test_file_handling.py file

Test_read_write_file
This test essentially generates a test quiz, writes that quiz to file, reads back the quiz form the file and confirms it is what was originally generated then deletes the quiz file and confirms it is no longer in the quiz_data folder. Each step is confirmed through pytest using assert statements

test_edit_file
This test generates a test question, writes that question to the first quiz in the topics list, then deletes the question from the quiz (to revert any changes). User input is replaced using the pytest monkeypatch feature. Each step is confirmed through pytest using assert statements.

Overview of Code
Testing Automated

Navigation (main.py)
- Primarily tested through exploratory testing
- Key test is to go through the entire menu tree, one by one and ensure
  - The navigation pathways are working as designed
  - The display being printed to the terminal is working as intended
  - The end points are leading to the correct functions

Functional testing (playquiz.py)
- The quiz functions, are not conducive to automated testing using pytest
- Used the manual testing with automated support. (manual_test_playwiz.py)
- The test generates a test quiz file with numbered questions and the correct answer indicated and runs it through the quiz round handler.
- It is used to test that the quiz is displaying correctly on the screen, and the application is correctly identifying the correct and incorrect answers when playing the quiz.

11

main.py
Primarily tested through planned exploratory testing
Test methodology is to go through the entire menu tree, one by one and ensure
- The navigation pathways are working as designed
- The display being printed to the terminal is working as intended
- The end points are leading to the correct functions

playquiz.py
Due to the nature of the quiz functions ie;
- Randomisation
- Printing output to terminal
- Requiring significant user input

it was concluded that automated testing using pytest would be impractical for the quiz portion of the application – used manual testing supported by automated quiz generation (namely the manual_test_playwiz.py file)

The manual_test_playwiz.py code generates a test quiz file with numbered questions and the correct answer indicated and runs it through the quiz round handler.
It is used to test and manually confirm that the quiz is displaying correctly on the screen, and the application is correctly identifying the correct and incorrect answers when playing the quiz.