ITP122 ASSESSMENT 3

Student Name: Tia Darvell
Student ID: A00029275
Date: 14 May 2023
Lecturer: Shazi Saremi

Table Of Contents

Task 1

Write a program to calculate how far a tennis ball will touch the ground. For example, if the ball travels 20 meters per minute for seven minutes, the distance travelled is 20*7=140 metres.

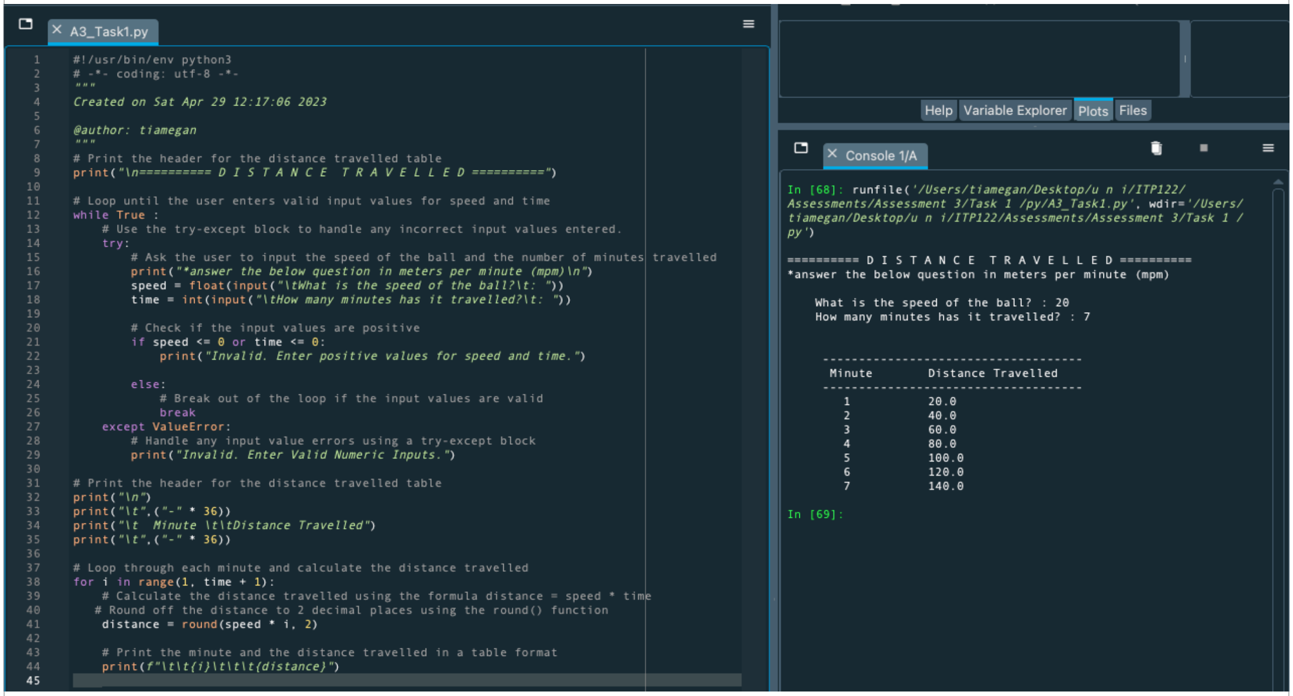
Write a program that asks the user for the speed of a ball (in meters per minute) and the number of minutes it has travelled. You should then use a while loop to display the distance travelled for each minute. (The distance ball travels can be calculated as follows: distance = speed * time).

Sample output:

```
What is the speed of the ball in (in metres per minute)? 20

How many minutes has it travelled? 7

-------------------------------------

Hour        Distance Travelled

-------------------------------------

 1              20

 2              40

 3              60

 4              80

 5              100

 6              120

 7              140
```

The above task should make an appropriate use of try-catch and ensures correct inputs are processed, and incorrect inputs are handled by using exceptions.

Answer:



```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Sat Apr 29 12:17:06 2023

@author: tiamegan
"""
# Print the header for the distance travelled table
print("\n========== D I S T A N C E  T R A V E L L E D ==========")

# Loop until the user enters valid input values for speed and time
while True :
    # Use the try-except block to handle any incorrect input values entered.
    try:
        # Ask the user to input the speed of the ball and the number of minutes travelled
        print("*answer the below question in meters per minute (mpm)\n")
        speed = float(input("\tWhat is the speed of the ball?\t: "))
        time = int(input("\tHow many minutes has it travelled?\t: "))

        # Check if the input values are positive
        if speed <= 0 or time <= 0:
            print("Invalid. Enter positive values for speed and time.")

        else:
            # Break out of the loop if the input values are valid
            break
    except ValueError:
        # Handle any input value errors using a try-except block
        print("Invalid. Enter Valid Numeric Inputs.")

# Print the header for the distance travelled table
print("\n")
print("\t",("-" * 36))
print("\t  Minute \t\tDistance Travelled")
print("\t",("-" * 36))

# Loop through each minute and calculate the distance travelled
for i in range(1, time + 1):
    # Calculate the distance travelled using the formula distance = speed * time
   # Round off the distance to 2 decimal places using the round() function
    distance = round(speed * i, 2)

    # Print the minute and the distance travelled in a table format
    print(f"\t\t{i}\t\t\t{distance}")

```

```
In [68]: runfile('/Users/tiamegan/Desktop/u n i/ITP122/
Assessments/Assessment 3/Task 1 /py/A3_Task1.py', wdir='/Users/
tiamegan/Desktop/u n i/ITP122/Assessments/Assessment 3/Task 1 /
py')

========== D I S T A N C E  T R A V E L L E D ==========
*answer the below question in meters per minute (mpm)

    What is the speed of the ball? : 20
    How many minutes has it travelled? : 7


     ------------------------------------
      Minute         Distance Travelled
     ------------------------------------
        1            20.0
        2            40.0
        3            60.0
        4            80.0
        5            100.0
        6            120.0
        7            140.0

In [69]:
```

This Python code prompts the user to enter the speed of a ball and the number of minutes it has travelled. It validates the input values by using a try-except block and handles any input value errors. If the input values are valid, the code calculates the distance travelled by the ball for each minute using the formula distance = speed * time and prints the minute and stance travel in a formatted way. This code provides an easy-to-use interface for calculating the distance travel by a ball.

Filepath: Assessment 3 / Task 1 / py / A3_Task1.py

Task 2

The Champions Soccer Club has a tournament every weekend. The club president has asked you to write a program to store the player and score details.

A program needs to manipulate each player's name and soccer score as entered from the input console. Make an appropriate use of dictionary data structure in python to handle this task. You can use key : value pair as PlayerName : Score.

The main program should first display a menu as follows. A user needs to select an operation from the main menu.

```
============================================
**Welcome to Champions Soccer Club**
Please choose an option from the followings.
1)  Add player name and score
2)  Display all the player information and scores
3)  Quit.
============================================
```

When option-1 is selected, the program allows the user to enter the player's name and score then store this information into the dictionary and provides the option to repeat this operation to add another player again. i.e., If user choose 'Y' or 'Yes' then repeat the Add operation. If the user chooses 'N' or 'No' then come back to main menu.

The following input validations must be performed on the data before saving the information into dictionary. Player name should be alphabetic letter (should not contain any digits). Player score should be digits and it should contain the values only from 0-100. If the input validations are incorrect, then display the appropriate message to user and allow to repeat the entry.

When option-2 is selected, the program reads the displays the contents of dictionary with player name and score on the screen.

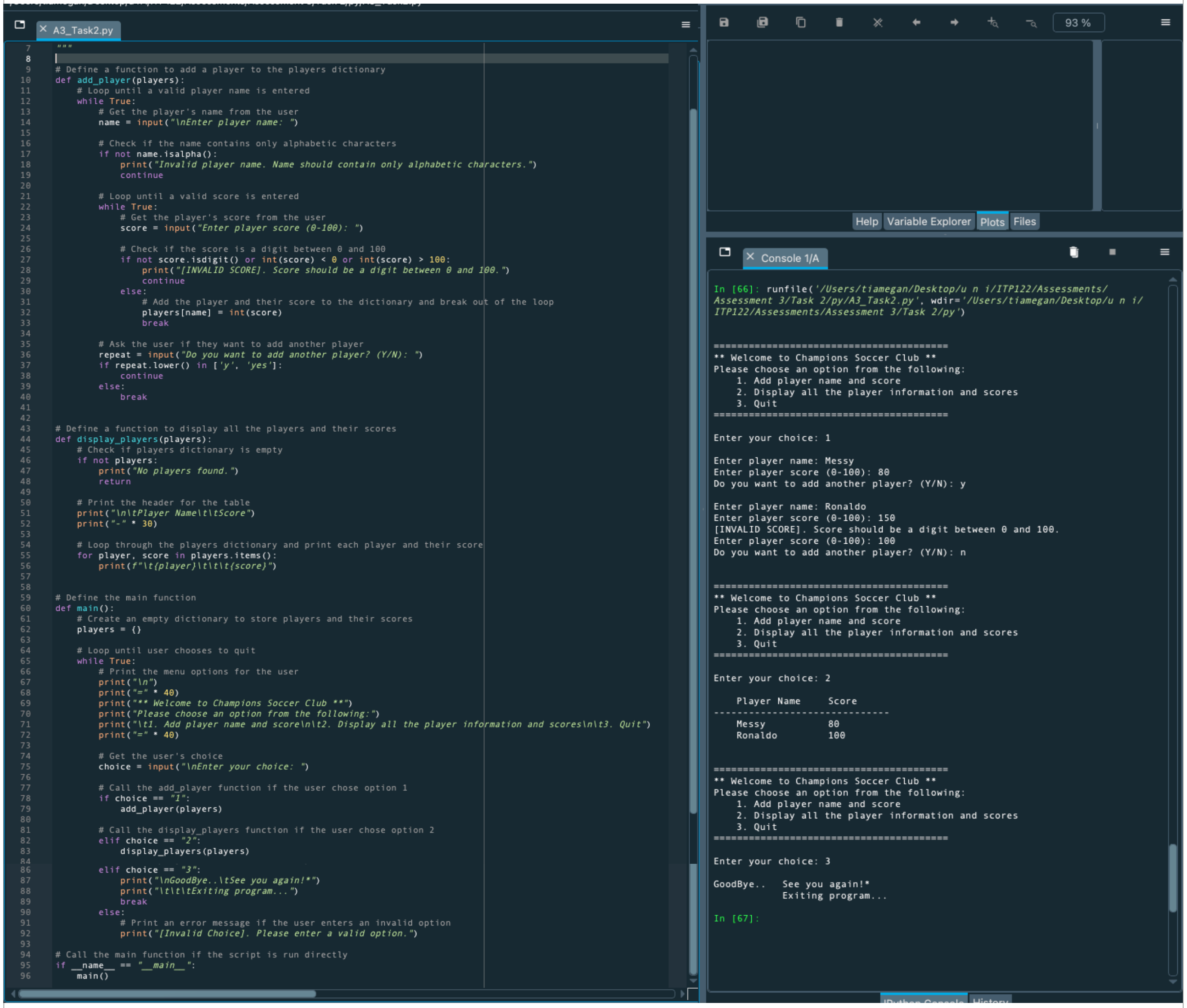
When option-3 is selected, the program quits/terminates.

Sample output screen:

```
============================================
**Welcome to Champions Soccer Club **
Please choose an option from the followings.
1)  Add player name and score
2)  Display all the player information and scores
3)  Quit.
============================================
Enter your choice            :1


Enter the player name        :Messy
Enter the score              :80
Do you want to add another player? Y


Enter the player name        :Ronaldo
Enter the score              :150
**Incorrect score** The scores should be from 0-100. Please try again!
Enter the score              :100
Do you want to add another player? N
```

```
==============================================
**Welcome to Champions Soccer Club **
Please choose an option from the followings.
1)  Add player name and score
2)  Display all the player information and scores
3)  Quit.
==============================================
Enter your choice                  -2

Player & Score Details:
-------------------------------
Player          Score
Messy           80
Ronaldo         100

==============================================
**Welcome to Champions Soccer Club**
Please choose an option from the followings.
1)  Add player name and score
2)  Display all the player information and scores
3)  Quit.
==============================================
Enter your choice                  - 3
**GoodBye.. See you again!**
```

Answer:



```python
"""

# Define a function to add a player to the players dictionary
def add_player(players):
    # Loop until a valid player name is entered
    while True:
        # Get the player's name from the user
        name = input("\nEnter player name: ")

        # Check if the name contains only alphabetic characters
        if not name.isalpha():
            print("Invalid player name. Name should contain only alphabetic characters.")
            continue

        # Loop until a valid score is entered
        while True:
            # Get the player's score from the user
            score = input("Enter player score (0-100): ")

            # Check if the score is a digit between 0 and 100
            if not score.isdigit() or int(score) < 0 or int(score) > 100:
                print("[INVALID SCORE]. Score should be a digit between 0 and 100.")
                continue
            else:
                # Add the player and their score to the dictionary and break out of the loop
                players[name] = int(score)
                break

        # Ask the user if they want to add another player
        repeat = input("Do you want to add another player? (Y/N): ")
        if repeat.lower() in ['y', 'yes']:
            continue
        else:
            break


# Define a function to display all the players and their scores
def display_players(players):
    # Check if players dictionary is empty
    if not players:
        print("No players found.")
        return

    # Print the header for the table
    print("\n\tPlayer Name\t\tScore")
    print("-" * 30)

    # Loop through the players dictionary and print each player and their score
    for player, score in players.items():
        print(f"\t{player}\t\t\t{score}")


# Define the main function
def main():
    # Create an empty dictionary to store players and their scores
    players = {}

    # Loop until user chooses to quit
    while True:
        # Print the menu options for the user
        print("\n")
        print("=" * 40)
        print("** Welcome to Champions Soccer Club **")
        print("Please choose an option from the following:")
        print("\t1. Add player name and score\n\t2. Display all the player information and scores\n\t3. Quit")
        print("=" * 40)

        # Get the user's choice
        choice = input("\nEnter your choice: ")

        # Call the add_player function if the user chose option 1
        if choice == "1":
            add_player(players)

        # Call the display_players function if the user chose option 2
        elif choice == "2":
            display_players(players)

        elif choice == "3":
            print("\nGoodBye..\tSee you again!*")
            print("\t\t\tExiting program...")
            break
        else:
            # Print an error message if the user enters an invalid option
            print("[Invalid Choice]. Please enter a valid option.")

# Call the main function if the script is run directly
if __name__ == "__main__":
    main()
```

```
In [66]: runfile('/Users/tiamegan/Desktop/u n i/ITP122/Assessments/
Assessment 3/Task 2/py/A3_Task2.py', wdir='/Users/tiamegan/Desktop/u n i/
ITP122/Assessments/Assessment 3/Task 2/py')


========================================
** Welcome to Champions Soccer Club **
Please choose an option from the following:
    1. Add player name and score
    2. Display all the player information and scores
    3. Quit
========================================

Enter your choice: 1

Enter player name: Messy
Enter player score (0-100): 80
Do you want to add another player? (Y/N): y

Enter player name: Ronaldo
Enter player score (0-100): 150
[INVALID SCORE]. Score should be a digit between 0 and 100.
Enter player score (0-100): 100
Do you want to add another player? (Y/N): n


========================================
** Welcome to Champions Soccer Club **
Please choose an option from the following:
    1. Add player name and score
    2. Display all the player information and scores
    3. Quit
========================================

Enter your choice: 2

    Player Name     Score
------------------------------
    Messy           80
    Ronaldo         100


========================================
** Welcome to Champions Soccer Club **
Please choose an option from the following:
    1. Add player name and score
    2. Display all the player information and scores
    3. Quit
========================================

Enter your choice: 3

GoodBye..   See you again!*
            Exiting program...

In [67]:
```

This Python script is for managing a soccer club's player information. It has a menu allowing the user to add players and their scores, display all player information, or quit the program.

The **add_player** function adds a player and their score to a dictionary of players after validating the input.

The **display_players** function prints out a table of all the players' names and scores.

The **main** function initialises an empty dictionary of players and presents the user with the menu options. It calls the appropriate functions based on the user's choice.

Overall, this script provides an easy-to-use interface for managing a soccer club's player information.

Filepath: Assessment 3 / Task 2 / py / A3_Task2.py

End of Assessment 3.