Probability Software Assignment

Name -: MEDOJU SRIMANNARAYANA
Roll no -: CS22BTECH11038

Reneration of Random number using python

1 Introduction

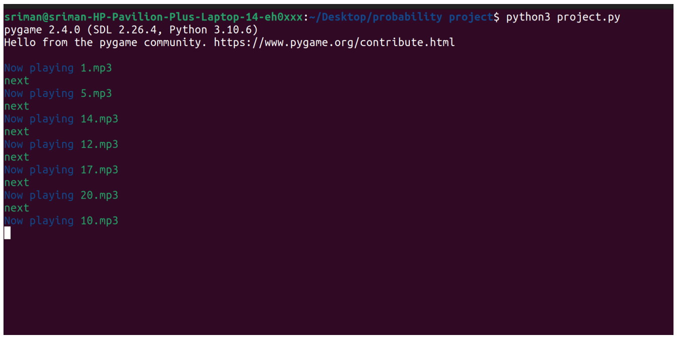
The provided code implements a music player using the Pygame library in Python. The program creates a playlist of 20 music files and allows the user to control the playback. The playlist is generated randomly, ensuring that each song is played once before the program terminates.

2 Implementation Details

1) **Library and Module Import:**
   - The required libraries and modules are imported at the beginning of the code.
   - Pygame is initialized to handle audio playback.
   - The colorama library is imported to provide colored output in the console.
2) **Music Playlist Initialization:**
   - An empty list `musicFiles` is created to store the paths of the music files.
   - A set `usedValues` is initialized to keep track of the songs that have been played.
   - A loop is used to generate the paths for the music files and append them to `musicFiles` list.
   - Each music file path is based on a numerical index, ranging from 1 to 20.
3) **Music Playback:**
   - A random index `d` between 0 and 19 is generated to select the first song in the playlist.
   - The selected song index is added to the `usedValues` set to mark it as played.
   - The corresponding music file is loaded using `pygame.mixer.music.load()` and played using `pygame.mixer.music.play()`.
   - The details of the currently playing song are printed to the console using colorama for formatting.
4) **User Interaction:**
   - A while loop is initiated to continuously listen for user input.
   - If the user enters ”next”, the program selects a new random index `d` that has not been used before.
   - The program stops the currently playing music, loads the new music file based on the new index, and plays it.
   - The details of the newly selected song are printed to the console.
   - The program keeps track of the number of songs played through the `count` variable.
   - If all 20 songs have been played (count reaches 20), a completion message is displayed, and the loop breaks.
   - If the user enters ”pause”, the program stops the currently playing music.
   - If the user enters ”play”, the program resumes playing the paused music.
   - If the user enters ”quit”, the program terminates by calling `pygame.quit()`.

3 Conclusion

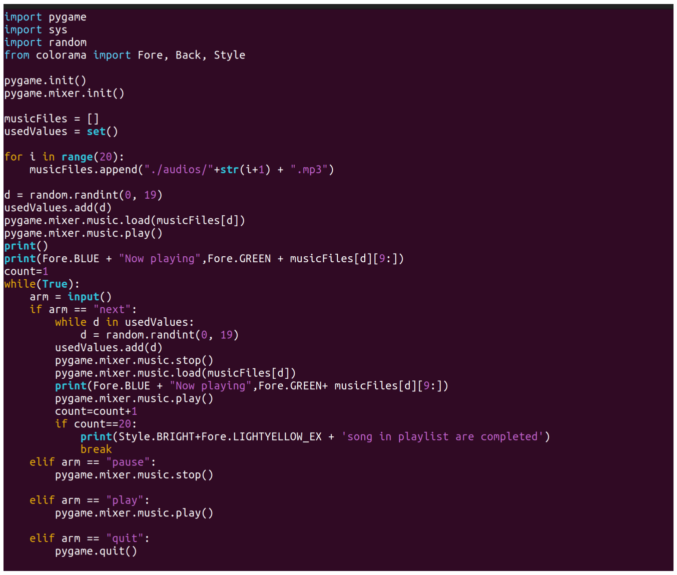
The implemented music player provides a simple interface for playing a random playlist of 20 songs. It allows the user to navigate through the playlist, pause and resume playback, and quit the program. The use of the Pygame library ensures smooth audio playback, while the colorama library enhances the console output with colored text. The code can be further expanded and customized to include additional features such as volume



```
sriman@sriman-HP-Pavilion-Plus-Laptop-14-eh0xxx:~/Desktop/probability project$ python3 project.py
pygame 2.4.0 (SDL 2.26.4, Python 3.10.6)
Hello from the pygame community. https://www.pygame.org/contribute.html

Now playing 1.mp3
next
Now playing 5.mp3
next
Now playing 14.mp3
next
Now playing 12.mp3
next
Now playing 17.mp3
next
Now playing 20.mp3
next
Now playing 10.mp3
```

Fig. 4. terminal



```
import pygame
import sys
import random
from colorama import Fore, Back, Style

pygame.init()
pygame.mixer.init()

musicFiles = []
usedValues = set()

for i in range(20):
    musicFiles.append("./audios/"+str(i+1) + ".mp3")

d = random.randint(0, 19)
usedValues.add(d)
pygame.mixer.music.load(musicFiles[d])
pygame.mixer.music.play()
print()
print(Fore.BLUE + "Now playing",Fore.GREEN + musicFiles[d][9:])
count=1
while(True):
    arm = input()
    if arm == "next":
        while d in usedValues:
            d = random.randint(0, 19)
        usedValues.add(d)
        pygame.mixer.music.stop()
        pygame.mixer.music.load(musicFiles[d])
        print(Fore.BLUE + "Now playing",Fore.GREEN+ musicFiles[d][9:])
        pygame.mixer.music.play()
        count=count+1
        if count==20:
            print(Style.BRIGHT+Fore.LIGHTYELLOW_EX + 'song in playlist are completed')
            break
    elif arm == "pause":
        pygame.mixer.music.stop()

    elif arm == "play":
        pygame.mixer.music.play()

    elif arm == "quit":
        pygame.quit()
```

Fig. 4. code