Report of Software Assignment

Randomised Playlist Using Python

AI1110: Probability and Random Variables

Rishitha Surineni
CS22BTECH11050

1 INTRODUCTION

This code is a music player program that utilizes the Pygame library in Python. It allows users to play a playlist of MP3 files randomly sorted from a given directory. Users can control the playback by entering specific commands.This code implements the features like Pause, Play, Skip, Quit .The main objective of this program is to play the songs in a random order everytime by assigning a random variable to each song.

2 DESCRIPTION

1) The program starts by importing necessary modules such as ‘os’, ‘random’, ‘pygame’, ‘time’, ‘mutagen.mp3’, ‘sys’, and ‘select’.
2) The code contains an infinite loop ‘(while True)’ that continues indefinitely until the program is manually interrupted which means the songs keep playing in random order till the program is exited.
3) Then the program sets the directory variable to the path of the ‘directory’ containing the MP3 files.Uses the ‘os.listdir’ function to retrieve a list of files in the directory and filters it to only include files with the ‘.mp3’ extension.
4) Assigns random numbers to each MP3 file and sorts them based on those numbers.This is done with the help of ‘random.randint’ function from the random module of Numpy library
5) Creates a playlist variable containing the names of the ‘MP3’ files in the sorted order.
6) Initializes the Pygame mixer by calling ‘pygame.mixer.init()’.
7) Iterates over each file in the playlist.Then
   a) Constructs the full file path.Loads the ‘MP3’ file using ‘pygame.mixer.music.load()’.
   b) Retrieves the length of the audio file using the ‘mutagen.mp3’ library.And then sets the ‘timeout’ variable to the length of the audio.
   c) Starts playing the music using ‘pygame.mixer.music.play()’.Records the start time.
   d) Waits for user input while the music is playing till a specified time.This is done using select.select function.The program waits for the user input till the remaining duration of the song is played.If there is user input within the specified timeout, it checks the input:
      - If the input is ‘next’, it prints a message and breaks the inner loop to move to the next song.
      - If the input is an empty string (Enter key), it pauses the music, records the current position, prompts the user to press Enter again to resume, and resumes the music if the user inputs an empty string.
      - If the input is ‘quit’, it prints a message and exits the program.
   e) If there is no user input within the timeout, it prints a message and breaks the inner loop to move to the next song.
   f) Sleeps for a short duration before checking for user input again.
   g) After all the songs in the directory are played, new random variables are assigned to each song.And the songs play according to new order with same functionalities as above.

3 CONCLUSION

To Conclude, the main purpose of this program is to assign a random variable to each song and then play them according to the random variable assigned

to them.The infinite while loop in the code ensures that when once the list of songs are done playing then the songs are played again in another random order.

4 Output Image

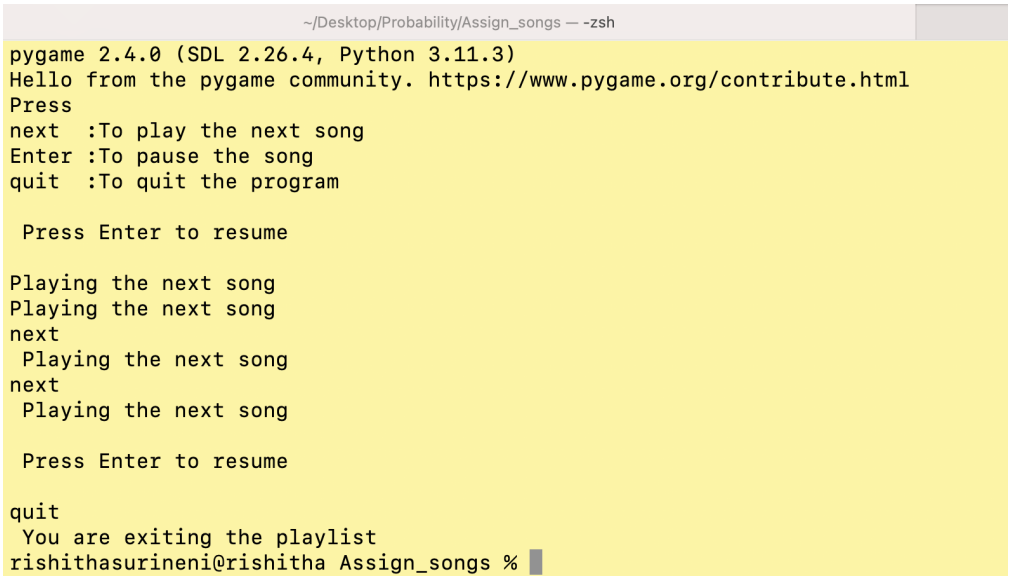

Image of Output displayed in the Terminal