

Report: Python code for random music player

AI1110: Probability and Random Variables
Indian Institute of Technology Hyderabad

Kudupudi D.V.Sai Aditya
AI22BTECH11013

- 1) **Introduction:** The provided code implements a random song player using the VLC media player library. It allows users to create a playlist folder, select random songs to play from that folder, and provides control options such as skipping, pausing/resuming, and quitting the program. The code utilizes modules like `os` for file operations, `numpy` for random selection, and `vlc` for media player functionality.
- 2) **Code Explanation:**
 - a) Importing Required Modules:
 - The code begins by importing the necessary modules: `os` for file operations, `numpy` (imported as `np`) for random selection, and `vlc` for media player functionality.
 - b) Playlist Folder and Song Retrieval:
 - The code defines the path to the playlist folder where the video files are stored.
 - It retrieves the list of video files from the playlist folder using `os.listdir()`.
 - c) VLC Media Player Instance:
 - An instance of the VLC media player is created using `vlc.Instance("--no-video")`. The `--no-video` option ensures that only audio is played.
 - d) Played Songs Tracking:
 - A list, `played_songs`, is initialized to keep track of the songs that have been played.
 - e) Main Loop for Song Playback:
 - The code enters an indefinite loop to continuously play random songs.
 - It checks if all songs in the playlist have been played by comparing the length of `played_songs` with the length of the playlist.
 - If all songs have been played, `played_songs` is reset, and a message is displayed indicating the completion of a loop.
 - f) Random Song Selection:
 - A random song is chosen from the playlist using `np.random.choice()` to select a song that has not been played.
 - The code ensures that the selected song has not already been played by checking if it exists in `played_songs`.
 - The full path to the selected song file is created using `os.path.join()`.
 - g) Media Player Setup and Playback:
 - A new media player is created using `instance.media_player_new()`.
 - The selected song is loaded using `instance.media_new()` and assigned to the media player.
 - The media is set to play only the audio by adding the `:no-video` option using `media.add_options()`.
 - The media is set to loop using `media.get_mrl()` and `player.set_media()` methods.
 - The audio is played using `player.play()`.
 - h) Displaying the Current Song:
 - The name of the currently playing song is displayed using `print("Now playing:", song_name)`

```

aditya@aditya-950QED:~$ python3 musicplayer.py
Now playing: IMG_0563
Enter 's' to skip or 'q' to quit: s
Now playing: IMG_0571
Enter 's' to skip or 'q' to quit: s
Now playing: IMG_0553
Enter 's' to skip or 'q' to quit: q
aditya@aditya-950QED:~$

```

Fig. 2: Sample image of how code runs in the terminal.

i) User Input and Control:

- A nested loop is started to wait for user input or until the audio finishes playing.
- The user is prompted to enter 's' to skip the current song, or 'q' to quit the program.
- The user input is checked and appropriate actions are taken based on the input:
 - 's': Stops the current song by calling `player.stop()` and breaks out of the nested loop.
 - 'q': Stops the current song by calling `player.stop()`, and the program exits using `exit()`.

```

aditya@aditya-950QED:~$ python3 musicplayer.py
Now playing: IMG_0566
Enter 's' to skip or 'q' to quit: s
Now playing: IMG_0560
Enter 's' to skip or 'q' to quit: s
Now playing: IMG_0569
Enter 's' to skip or 'q' to quit: s
Now playing: IMG_0575
Enter 's' to skip or 'q' to quit: s
Now playing: IMG_0553
Enter 's' to skip or 'q' to quit: s
Now playing: IMG_0571
Enter 's' to skip or 'q' to quit: s
Now playing: IMG_0568
Enter 's' to skip or 'q' to quit: s
Now playing: IMG_0556
Enter 's' to skip or 'q' to quit: s
Now playing: IMG_0563
Enter 's' to skip or 'q' to quit: s
Now playing: IMG_0561
Enter 's' to skip or 'q' to quit: s
Now playing: IMG_0558
Enter 's' to skip or 'q' to quit: s
Now playing: IMG_0565
Enter 's' to skip or 'q' to quit: s
Now playing: IMG_0555
Enter 's' to skip or 'q' to quit: s
Now playing: IMG_0572
Enter 's' to skip or 'q' to quit: s
Now playing: IMG_0567
Enter 's' to skip or 'q' to quit: s
Now playing: IMG_0570
Enter 's' to skip or 'q' to quit: s
Now playing: IMG_0574
Enter 's' to skip or 'q' to quit: s
Now playing: IMG_0562
Enter 's' to skip or 'q' to quit: s
Now playing: IMG_0559
Enter 's' to skip or 'q' to quit: s
Now playing: IMG_0557
Enter 's' to skip or 'q' to quit: s
Loop completed. Starting from the beginning.
Now playing: IMG_0566
Enter 's' to skip or 'q' to quit: q
aditya@aditya-950QED:~$

```

Fig. 2: Indication of completion when all songs are played randomly once.

- After handling the user input, the code checks if the audio has finished playing by calling `player.get_state()` and comparing it to `vlc.State.Ended`.
 - If the audio has ended, the nested loop is exited, and the next iteration of the main loop begins.
- j) Finally, the program stops the player using `player.stop()`.
- 3) **Conclusion:** The provided Python program allows users to watch a random selection of videos from a specified video playlist folder. It utilizes the VLC library for video playback and provides options to skip, or quit the program during video playback.