Friday, July 28, 2023

# Developers Documentation

## Introduction/Overview

- The Slick Music Player is a Graphical user interface application developed using the 'Tkitner' library in python. This application allows the user to play MP3 files from a chosen music folder, display the list of available music files, and control the music playback with play, pause, next, previous, and volume control buttons. The application provides an animated GIF that plays when you press the play button to play music. There is also an animation that plays when the music is paused.

## Install/deployment/admin issues:

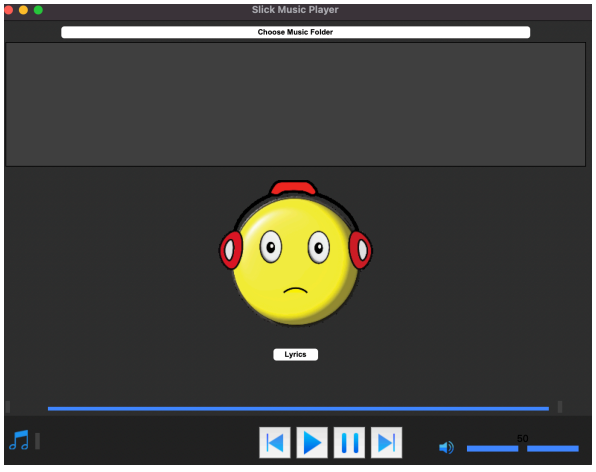
- If you run into a problem with a lyrics. Generate an API from genius.com (Steps Below)

1. Click the link to access the website: https://docs.genius.com/
2. If you do not have a genius account, it will prompt you to sign up.
3. Once signed up click New API client and fill out the information.
4. On the next screen click generate access token
5. In the code find where it directs you to replace the Genius API key.
6. In quotes '' type in your access key and you will the lyrics to work.

- If you run into problems installing stuff make sure you have administration rights.

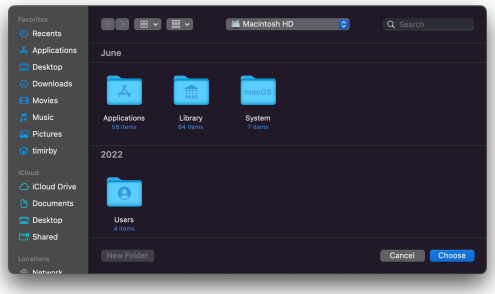
## User interaction

1. Launching the application the main window of the music player is displayed

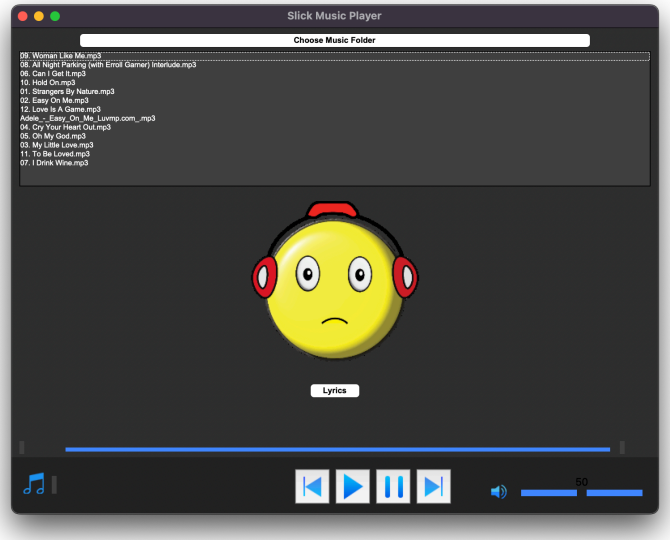



2. The player consist of
   - "Choose Music Folder" Button: This button allows the user to browse and select a folder counting MP3 files.

- Playlist Listbox: This Listbox displays the name of the available music files in the chosen music folder. The user can select a son from the list by clicking on it.
- GIF Display: The application displays an animated GIF below the playlist that shows two types of animations. An animation for when listening to music and animation for not listening to music.
- Play, Pause, Next, and Previous Buttons: These buttons control the music playback. The Play button starts playing the selected song, the Pause button pauses/unpauses the music, the Next button plays the next song in the playlist, and the Previous button  plays the previous song in the playlist
- Volume Slider: The user controls the volume level using the slider.
- Song Progress: This indicates the process of the song currently playing.
- Current Song Label: This label displays the name of the currently playing song.
- Time Labels: These labels show the current position and total duration of the playing song.
- Lyrics Button: This button is used to open a window and you can search an artist and their song and look for lyrics to the song.

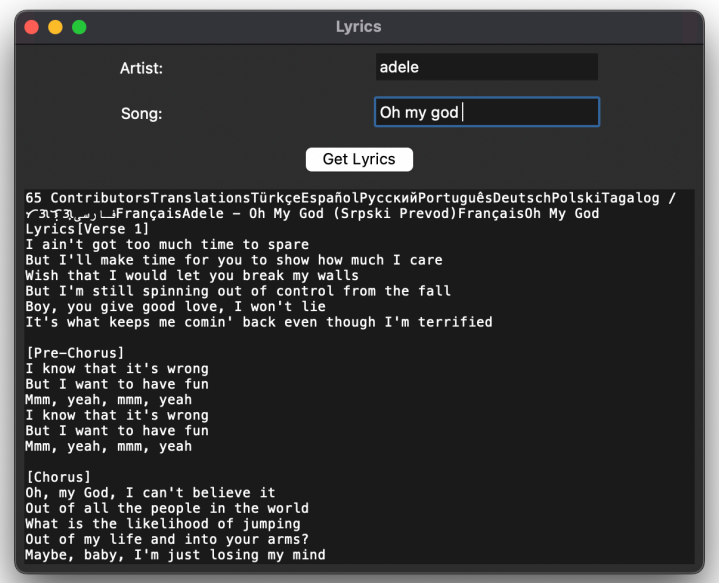


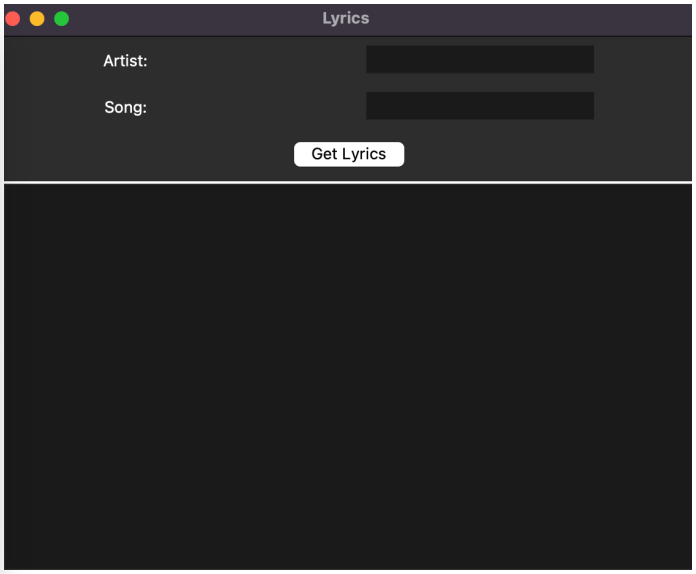

3. When the “Choose Music Folder” button is clicked, a file dialog opens, allowing the user to select a folder containing music files. The application will then populate the playlist Listbox with the names of the available music files in that folder.
4. The user can select the song from the playlist Listbox by clicking. When a song is double-clicked on or the Play button is clicked, the music will play.
5. The song progress slider will indicate the song progress slider. When the music is playing the Gif will change to the listening animation.
6. When the pause button, the application will pause the music and switch to the not-listening animation. Clicking the pause button will resume the music which will switch it back to the original animation.




7. The user can use the Next and Previous buttons to play the next and previous songs in the playlist, respectively. The animated GIF will change to the appropriate animation for the current

state of music. (If playing will perform listening animation, if not playing will perform not listening)
8. The user can control the volume level using the volume slider. The sliders position will indicate the current volume level, and the application will adjust the music volume accordingly. When the slider is moved to 0 the icon will change to the mute button.
9. The time labels will display the current position and total duration of the currently playing song, updating every second.
10. The Lyrics button when clicked, will display another TkInter window that will give the user




the ability to search the artist and song and display the lyrics. If nothing is typed or a song can not be found “Lyrics can not be found” will be display in the box.
11. The user can interact with the UI, selecting songs, adjusting volume, and viewing song progress and lyrics as needed.

---

## Class: ‘MusicPlayer’ (“Code Walkthrough” )

### Description

- The ‘MusicPlayer’ class is the main class that represent the music player application. This class contains the GUI elements, music playback, volume control, animation display, and the functions and GUI for the lyrics are also in this class as well.

### Attributes

- ‘window’: The main application window created ‘tkinter.Tk’. It holds all the GUI elements of the application

- ‘vol’: An ‘IntVar’ that keeps track of the current volume level.
- ‘extention’: A string representing the file extension of music files (default value: ‘.mp3’)
- ‘gif1’: A string containing the path to the first animated GIF file for listening animation.
- ‘gif2’: An string contains the path to second animated GIF for not-listening animation.
- ‘Count’: An integer representing the current frame index for animation display.
- ‘change_anim’: An integer representing the current frame index for animation display(used in the pause animation).
- ‘motion1’: An instance of ‘PIL.Image’ representing the first animate GIF for the listening+animation
- ‘motion2’: An instant of ‘PIL.Image’ representing the second animated GIF for not_listening animation.
- ‘listening_frames’/‘not_listening_frames’: this sets up the frames for the GIF.
- ‘frame_list1’/‘frame_list2’: These two represent a list of ‘PhotoImage” objects representing the frames of the listening_animation and not_listening animation.
- ‘current_index’: An integer representing the index of the currently playing song in the music list(initialized to -1)
- ‘play’: A ‘PhotoImage’ object representing the play button icon.
- ‘pause’: A ‘PhotoImage’ object representing the pause button icon.
- ‘forward’ A ‘PhotoImage’ object representing the next button icon.
- ‘back’:  A ‘PhotoImage’ object representing the previous button icon.
- ‘audio’: A ‘PhotoImage’ object repressing the speaker icon
- ‘mute’: A ‘PhotoImage’ object representing the mute icon
- ‘icon’: A ‘PhotoImage’ object representing the music icon.

Methods

- ‘__init__(self): The constructor model that initializes the application window and sets up the GUI elements, animations, and music player

- ‘open(self)’: Opens a file dialog to choose the music folder and displays the list of available music files in the Listbox.
- ‘play_selected_song(self, event)’: Plays the selected song from the Listbox when the user double clicks on it.
- ‘Play_music(self)’: Play the selected music from the Listbox where the ‘Play’ button is clicked/.
- ‘Pause_Unpause(self)’: Pauses or unpauses the currently playing music based on he state of the “Pause” button
- ‘Next_song(self)’: Plays the next song in the list without requiring explicit selection.
- ‘Prev_song(self)’: Plays the previous song in the list without requiring explicit selection.
- ‘ volume (self, vol)’: Controls the volume of the music a slider.
- ‘playing_time(self)’: Updates the process of the currently playing song and displays the current and total time.
- ‘listening_animation(self, count): Displays the listing animation(animated GIF) while paused.
- ‘not_listening_animation(self, count)’: Displays the not listening animation(animated Gif) while paused.
- ‘create_gui(self)’: Private method to create and pack all GUI elements in the application window.

---

## Known Issues

Minor:
- Sometimes may need to press the pause button more then once for files to play
- GIF may not always play on a double click, press the play button to resolve
- Play button will restart the music so press the pause button to resume.
- Does not support audio files outside of MP3. Trying to add folders that don’t contain mp3 file will not be added to the playlist

Major:
- When using the lyrics function sometimes it may freeze while looking for lyrics. To fix restart the application.
- Sometimes pressing the back button in the beginning of the playlist might cause problems (looking at the index and seeing how to repeat may fix)

## Future Work

- One thing I wanted to do was create a audio mesh and have a scene that responds to the music. I started working on one but trying to implement it in this app would of taken to much time.
- Another thing I would of liked to add was pulling the music directly from the mp3 file or having the lyrics pulled up based on the song that was playing. I wanted a lyrics function in the player so I went with the user looking it up itself.
- Giving the user the ability to implement their own gifs for playing and pausing animation. One thing I wanted to do was make this personalizable toward the user so the user could upload their gif.
- Create a GUI using KIVY instead to possibly have access to clean GUI that will display album art and have more options to switch between album art, gifs, and even implement clips of the music video in the song.
- Additional file formats could be added to make it a more usable application.
- A shuffle and repeat function to mix the playlist up if you are adding multiple files.
- Make a mini version of the player so it does not take up as much space on the computer.
- Add an AI program to the music player to give suggestion and related songs based on the files within the playlist.