



MUSIC PLAYER USING PYTHON

M.BHARAT REDDY, R.CHANDRA BABU, ANIRUDH RAMARAJU, SARADHI

Department of Electronics and Communication Engineering, SRM University-AP



ABSTRACT

Audio is an important source of communication and is as important as text in today's time. We know that the audio files are digital files. Therefore, there is a need of a tool to run the digital files or in other words, play the files. Without this tool or player, we'll never be able to listen to music, movies or the contents of any audio file.

Thus, we need MP3 players. It is a device using to play MP3s and other digital audio files. We can build this by ourselves without have to download and install premium music players. The Mp3 player GUI project idea attempts to emulate the physical MP3 Player.

This program will allow you to play songs, music, and all MP3 files on your desktop or laptops. MP3 player using Python is a basic programming application built using the programming language Python. It is a GUI program built by the means of Python libraries Tkinter, Pygame and Mutagen. The MP3 player application should have the capabilities of playing a song, create and display a playlist, pause and resume a long and change the song, that is, play the previous or next song.

INTRODUCTION

We need an application that will allow us to play or listen to digital audio files. MP3 player is the device to play MP3s and other digital audio files. The MP3 GUI program application attempts to emulate the physical MP3 Player. This program will allow you to play songs, music, and all MP3 files on your desktop or laptops.

The main objective of this project is to allow users to play MP3 and digital audio files. To be engaging for users, the application has to have a simple but beautiful user interface. This GUI project is developed using Python programming language. The GUI aspect of the application is built using the Tkinter library of Python. The interactive part of the application that handles the MP3 files uses the Pygame and Mutagen libraries. You can have an interface for listing the available MP3 files. You can also give users the option to list other digital audio files that are not MP3. The users will also expect the MP3 Player to have an interface that shows information on the file that is playing. Some of the information you can include are the name of the file, its length, the amount played, and the amount not played, in minutes and seconds.

Python has libraries that can play audio files, such as Pygame, which allows you to work with multimedia files in few lines of code. Similar libraries are Pymedia and Simple audio. These libraries can handle a lot of digital audio files. They can handle other file types, not just the MP3 files.

CONCLUSION

Now, we have our own music player and we can listen to songs when you work with data. Sometimes working with data could be very challenging and having your own mp3 music player is very helpful. In addition, I think that building an mp3 music player or other applications can really help you to know new Python modules and at the same time learn more about Python programming.

I always say that the better way to improve your programming skills is building something or automating a process. That permits you to have more confidence with programming and it is very useful for your personal portfolio.

WORKING PROCESS

First, we Import the libraries. And then we Create an object of the tkinter and Pygame libraries.

After that we Create a window using Tkinter object. Further we add buttons that provide different functionalities like Playing the song, Pausing the song, Resuming the song and to Stop the song. The Player automatically add songs as we already given the path location.

The song's information will be displayed in the window.

Display screen will display the details of the entire playlist. Close button will automatically clear the song list and will stop playing the song.

INSTALLATIONS REQUIRED

- Installing Python package.
- Pygame - It is a Python library used to create video games. To create a music player with Python, we will be using the Pygame sound component. Pygame is also a very library that gives us the power of playing with different multimedia formats like audio, video, etc. We will be using Pygame's 'mixer.music' module for providing different functionality to are music player application, related to manipulation with the song tracks.
- Tkinter- This is the most popular and very easy to use library that comes with many widgets which helps in creating of seamless and nice-looking GUI Applications.
- Os- To access the song folder.

PROBLEM DESCRIPTION

- To build an MP3 player using Python programming language to be able to play and listen to songs, MP3 files and other digital audio files.
- Determine the functionalities of the MP3 player.
- The player should be have a simple and easy to use GUI with options for various functions, display screen to display the entire playlist and buttons to shut down the player.
- The player should be able to play any song. It should be capable of playing MP3 files or any other digital audio files.
- The player should allow the user to browse through the contents of the computer drive to choose songs to be played or queued.
- It should provide the user with option to pause or resume the song.
- Lastly, the user should get basic details about the current playing song

CODE

```
import pygame
from pygame import mixer
from tkinter import *
import os

def playsong():
    currentsong=playlist.get(ACTIVE)
    print(currentsong)
    mixer.music.load(currentsong)
    songstatus.set("Playing")
    mixer.music.play()
```

```
def pausesong():
    songstatus.set("Paused")
    mixer.music.pause()
```

```
def stopsong():
    songstatus.set("Stopped")
    mixer.music.stop()
```

```
def resumesong():
    songstatus.set("Resuming")
    mixer.music.unpause()
```

```
root=Tk()
root.title('MP3 Music player')
mixer.init()
songstatus=StringVar()
songstatus.set("choosing")
```

```
#playlist-----
playlist=Listbox(root,selectmode=SINGLE,bg="black",fg="white",font=('optima',15),width=32,height=7)
playlist.grid(columnspan=20)

os.chdir(r'D:\songs')
songs=os.listdir()
for s in songs:
    playlist.insert(END,s)

playbtn=Button(root,text="play",command=playsong)
playbtn.config(font=('candara',20),bg="white",fg="green",padx=2,pady=0)
playbtn.grid(row=1,column=0)
```

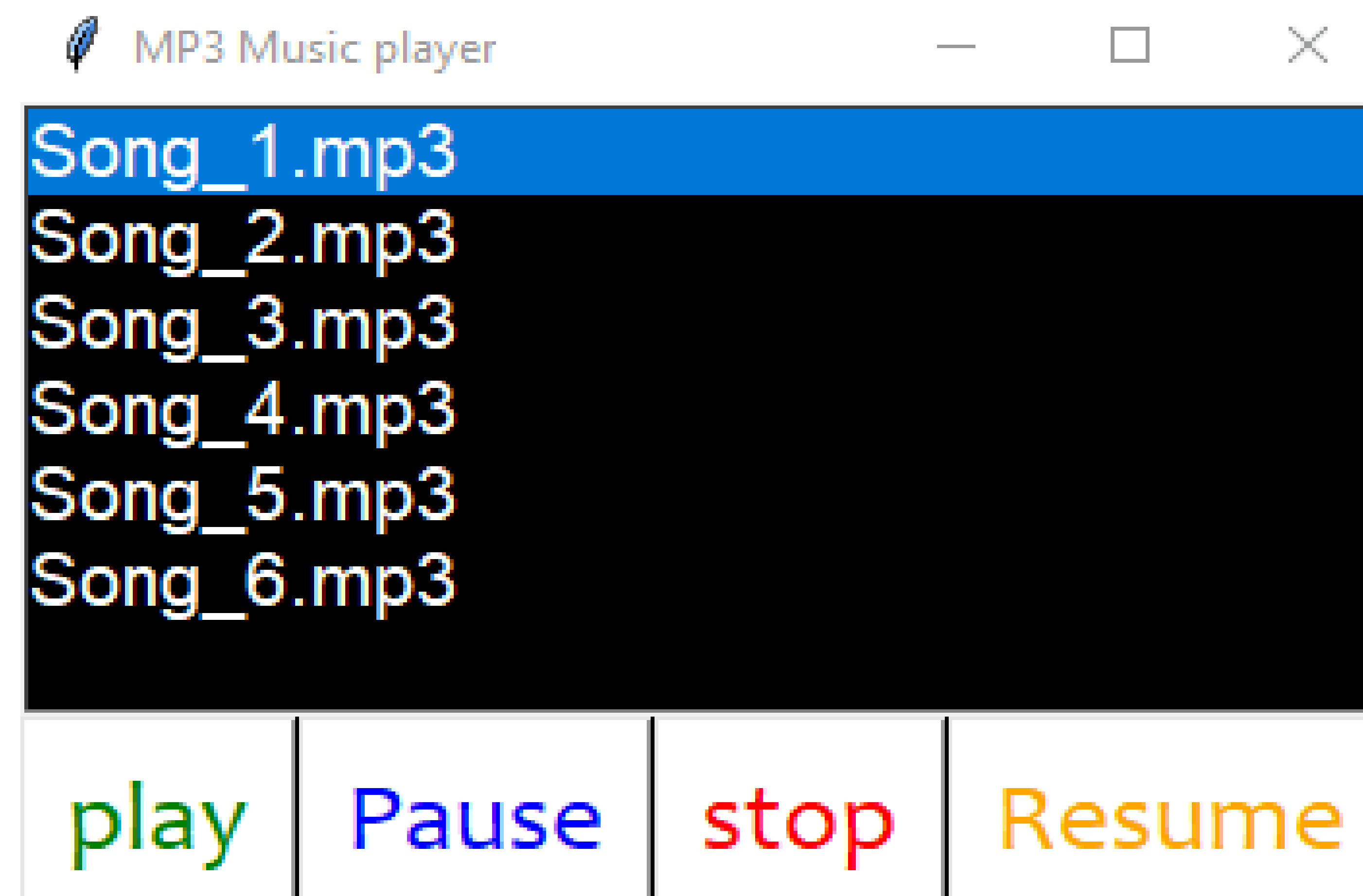
```
pausebtn=Button(root,text="Pause",command=pausesong)
pausebtn.config(font=('candara',20),bg="white",fg="blue",padx=2,pady=0)
pausebtn.grid(row=1,column=1)
```

```
stopbtn=Button(root,text="stop",command=stopsong)
stopbtn.config(font=('candara',20),bg="white",fg="red",padx=2,pady=0)
stopbtn.grid(row=1,column=2)
```

```
Resumebtn=Button(root,text="Resume",command=resumesong)
Resumebtn.config(font=('candara',20),bg="white",fg="orange",padx=2,pady=0)
Resumebtn.grid(row=1,column=3)
```

```
mainloop()
```

SNAPSHOT OF MUSIC PLAYER



ACKNOWLEDGEMENTS

We thank our advisor, Dr. SIBENDU SAMANTA, for his invaluable guidance and support during this project.