MINI PROJECT
(2020-21)

Music Player Using Python

MID-TERM REPORT

Institute of Engineering & Technology

Submitted by-
Shubham Singh
(181500698)

Supervised By: -
Mrs. Priya Agarwal
Technical Trainer
Department of Computer Engineering & Applications

Contents

Abstract

Music is important part of our life. We know that the music 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 Music 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 Music player GUI project idea attempts to emulate the physical Music Player.

This program will allow you to play songs, music, and all MP3 files on your desktop or laptops. Music 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 OS.

The Music 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

1.1 General Introduction to the topic

To build a Music Player using Python we need it's Tkinter Toolkit which is it's GUI (Graphical User Interface), package. Also we require different of it's libraries like pygame, os, etc.

Since Python language provides usability to write cross-platform codes that can run all types of the system without even doing any big changes in codes.

About Python: -

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.
It was mainly developed for emphasis on code readability, and its syntax allows programmers to express concepts in fewer lines of code.

PYTHON TKINTER GUI :

Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit.

How Tkinter Works:

Tkinter is a set of wrappers that implement the Tk widgets as Python classes. In addition, the internal module _tkinter provides a threadsafe mechanism which allows Python and Tcl to interact. Tkinter's chief virtues are that it is fast, and that it usually comes bundled with Python..

What is OS Module:

Python OS module provides the facility to establish the interaction between the user and the operating system. It offers many useful OS functions that are used to perform OS-based tasks and get related information about operating system.

The OS comes under Python's standard utility modules. This module offers a portable way of using operating system dependent functionality. The Python OS module lets us work with the files and directories.

What is Pygame:

Pygame is a cross-platform set of Python modules which is used to create video games. It consists of computer graphics and sound libraries designed to be used with the Python programming language. Pygame is suitable to create client-side applications that can be potentially wrapped in a standalone executable.

1.2 Hardware and Software Requirements:

a) **Hardware:**

- 120 GB HDD
- Minimum 4 GB RAM
- i5 Processor

b) **Software:**

- Pycharm
- Operating System(Windows, Linux)
- **Programming Language**
  - ❖ Python 3.8
    - ○ Tkinter (Python GUI library)
  - ○ Pygame (Python Library)
    - ○ OS module

Problem Statement:

1. To build a Music player using Python programming language to be able to play and listen to songs, MP3 files and other digital audio files.

2. The player should be having 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.

3. The player should be able to play any song. It should be capable of playing MP3 files or any other digital audio files.

4. The player should allow the user to browse through the contents of the computer drive to choose song/s to be played or queued.

5. It should provide the user with option to pause or resume the song.

6. The user should be able to play the previous or the next song in the playlist.

7. Lastly, the user should get basic details about the current playing song. The details can include the song name, singer's name, the duration of the song, size of the file, etc.

Objective

The main objective of this project is to design cross-platform music player using python and tkinter, that will help users play, pause and stop a track as per their needs. The application will also help users to increase the volume of their device. Python is really very interesting language because this language provides usability to write cross-platform codes that can run all types of the system without even doing any big changes in codes.

This Python module provides a high-level core Music player interface where you are supposed to provide all the remaining high-level logic like the user interface, the playlist logic and the audio data.

Implementation Details:

1. Import the python Libraries
2. Create an object of the tkinter and Pygame Libraries.
3. Create a window using Tkinter object.
4. Create frames for different widgets.
5. Add buttons that provide different functionalities.

- Play a song
- Pause the song
- Play next song
- Play Previous song
- Increase Decrease Volume
- Load Songs

6. Add a label to display the song's information.
7. Display screen will display the details of the entire playlist.

Progress

This project is divided in two parts:

Part 1 is completed:

1. Import the python Libraries
2. Create an object of the tkinter and Pygame Libraries.
3. Create a window using Tkinter object.
4. Create frames for different widgets.

Part 2:

1. Add buttons that provide different functionalities.

- Play a song
- Pause the song
- Play next song
- Play Previous song
- Increase Decrease Volume
- Load Songs

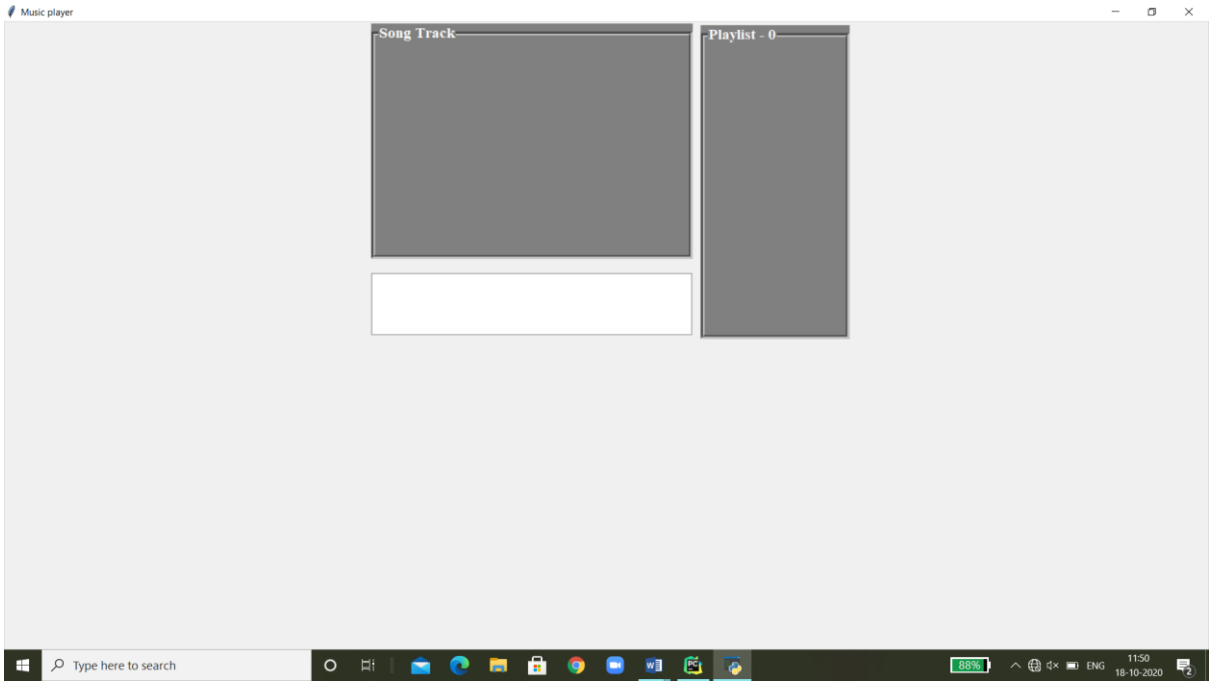
2. Add a label to display the song's information.
3. Display screen will display the details of the entire playlist.

Source code:

```
import os
import pickle
import tkinter as tk
from pygame import mixer

class Player(tk.Frame):       #player class
    def __init__(self,master):
        super().__init__(master)
        self.master=master
        self.pack()

        self.playlist=[]

        self.create_frames()
        self.track_widgets()
        self.control_widgets()
        self.tracklist_widget()

    def create_frames(self):
        self.track=tk.LabelFrame(self,text='Song Track',
```

```
                        font=("Times New Roman",14,"bold"),

bg="grey",fg="white",bd=5,relief=tk.GROOVE)
        self.track.configure(width=410,height=300)
        self.track.grid(row=0,column=0,padx=10)

        self.tracklist = tk.LabelFrame(self, text=f'Playlist -
{len(self.playlist)}',
                                       font=("Times New Roman", 14, "bold"),
                                       bg="grey", fg="white", bd=5,
relief=tk.GROOVE)
        self.tracklist.configure(width=190, height=400)
        self.tracklist.grid(row=0, column=1, rowspan=3,pady=5)

        self.controls = tk.LabelFrame(self,
                                       font=("Times New Roman", 14, "bold"),
                                       bg="white", fg="white", bd=2,
relief=tk.GROOVE)
        self.controls.configure(width=410, height=80)
        self.controls.grid(row=2, column=0,pady=5,padx=10)

    def track_widgets(self):
        pass

    def control_widgets(self):
        pass

    def tracklist_widget(self):
        pass


root=tk.Tk()
root.geometry('600x400')
root.wm_title('Music player')

app=Player(master=root)
app.mainloop()
```

Screenshots



References

- WWW.tkdocs.com
- WWW.python.org
- www.geeksforgeeks.com