*DancePracticeMusicPlayer* is a Kivy application written in Python that creates a music player with features useful at dance practices using a predetermined sequence of dances types. It automatically generates a playlist with an announcement of each dance type before those dance selections are played.

Overview

The application is designed to play music files (MP3, WAV, OGG) from a selected directory. It has a user interface with buttons to play, pause, stop, and restart the music. The application also allows users to select a music directory, adjust the volume, and choose a practice length (e.g., 60 minutes, 90 minutes, etc.).

Here's a breakdown of the code:

Classes

The code defines two main classes: MyFileChooser and MusicPlayer.

1. MyFileChooser: This class creates a file chooser dialog that allows users to select a music directory. It inherits from Kivy's GridLayout class.
2. MusicPlayer: This class is the main application class, which inherits from Kivy's BoxLayout class. It contains the music player's user interface and functionality.

MusicPlayer Class

The MusicPlayer class has several key features:

* Music Directory: The application loads music files from a selected directory. The directory can be changed using the "Select Music" button.
* Playlist: The application creates a playlist of music files from the selected directory. The playlist is displayed as a list of buttons, each representing a song.
* Music Player Controls: The application has buttons to play, pause, stop, and restart the music.
* Volume Control: The application has a volume slider to adjust the music volume.
* Practice Length: The application allows users to choose a practice length (e.g., 60 minutes, 90 minutes, etc.). This feature is used to adjust the number of songs played.
* Song Information: The application displays song information, such as the song title, artist, album, and genre.

Key Methods

Some key methods in the MusicPlayer class include:

* play\_sound: Plays the selected song.
* pause\_sound: Pauses the music (not working on Windows).
* stop\_sound: Stops the music.
* restart\_sound: Restarts the music from the beginning.
* update\_progress: Updates the music progress bar and song information.
* update\_playlist: Updates the playlist when the music directory changes.
* display\_playlist: Displays the playlist as a list of buttons.

Config File

The application loads configuration data from a JSON file named config.json. The configuration data includes the music directory, volume, and practice length.

Practice Dances

The application has a feature called "Practice Dances," which allows users to select a practice length and dance style (e.g., Waltz, Tango, etc.). The application adjusts the number of songs played based on the selected practice length and dance style.

Overall, this code creates a functional music player with various features, including playlist management, music controls, and practice length selection.