NAME- GRACY JAIN

PROJECT NAME- MUSIC PLAYER IN PYTHON

The necessary steps in your music player project as a theory form:

1 Importing the Pygame Library: The code begins by importing the pygame library, which is required for handling music playback.

2 Initializing Pygame: The pygame.init() function is called to initialize the pygame library and set up the necessary resources for music playback.

3 Creating the MusicPlayer Class: A class named MusicPlayer is defined. This class represents the music player and encapsulates its functionality.

4 Initializing the MusicPlayer: In the init method of the MusicPlayer class, the paused attribute is set to False initially, indicating that the music is not paused.

5 Playing Music: The play_music method of the MusicPlayer class is responsible for playing music. It takes a file_path parameter, which represents the path to the music file. Inside this method, the music file is loaded using pygame.mixer.music.load(file_path), and then the playback is started using pygame.mixer.music.play().

6 Pausing Music: The pause_music method pauses the currently playing music. It uses pygame.mixer.music.pause() to pause the playback and sets the paused attribute to True.

7 Resuming Music: The resume_music method resumes the paused music. It utilizes pygame.mixer.music.unpause() to resume the playback and sets the paused attribute to False.

8 Stopping Music: The stop_music method stops the currently playing music. It uses pygame.mixer.music.stop() to stop the playback and sets the paused attribute to False.

9 Creating an Instance of the MusicPlayer Class: An instance of the MusicPlayer class named player is created.

10 Main Program Loop: The code enters a while loop that repeatedly displays the menu options and prompts the user for their choice.

11 User Input and Functionality: Based on the user's choice, the code executes the corresponding functionality. For example, if the user chooses to play music (option 1), they are prompted to enter the file path, and then the play_music method is called with the provided file path. Similarly, for other options, the relevant methods of the MusicPlayer class are invoked.

12 Exiting the Program: If the user selects option 5, the code breaks out of the while loop and prints "Exiting..." to indicate that the program is terminating.

13 Quitting Pygame: After the loop, the pygame.quit() function is called to cleanly shut down the pygame library.

In [*]:

```python
import pygame

# Initialize Pygame
pygame.init()

# Create the MusicPlayer class
class MusicPlayer:
    def __init__(self):
        self.paused = False

    def play_music(self, file_path):
        # Load the music file
        pygame.mixer.music.load(file_path)
        # Start playing the music
        pygame.mixer.music.play()

    def pause_music(self):
        # Pause the currently playing music
        pygame.mixer.music.pause()
        self.paused = True

    def resume_music(self):
        # Resume the paused music
        pygame.mixer.music.unpause()
        self.paused = False

    def stop_music(self):
        # Stop the currently playing music
        pygame.mixer.music.stop()
        self.paused = False

# Create an instance of the MusicPlayer class
player = MusicPlayer()

# Main program Loop
while True:
    # Display the menu options
    print("1. Play music")
    print("2. Pause music")
    print("3. Resume music")
    print("4. Stop music")
    print("5. Exit")

    # Get the user's choice
    choice = input("Enter your choice: ")
    if choice == '1':
        # Ask for the file path of the music
        file_path = input("Enter the file path of the music: ")
        # Call the play_music method to play the music
        player.play_music(file_path)

    elif choice == '2':
        if player.paused:
            print("Music is already paused.")
        else:
            # Call the pause_music method to pause the music
            player.pause_music()
            print("Music paused.")

    elif choice == '3':
        if not player.paused:
            print("Music is not paused.")
        else:
            # Call the resume_music method to resume the music
            player.resume_music()
            print("Music resumed.")

    elif choice == '4':
        # Call the stop_music method to stop the music
        player.stop_music()
        print("Music stopped.")

    elif choice == '5':
        print("Exiting...")
        break

    else:
        print("Invalid choice. Please try again.")

# Quit Pygame
pygame.quit()
```