

Semester Project Report
CS-212 Object Oriented Programming

In-Car Media Player



Project Report of

Azlaan Ranjha

Laiba Jabbar

Rana Abdullah Azhar

Degree

DE-43

Department

EE

Syndicate

B

DEPARTMENT OF ELECTRICAL ENGINEERING
NUST College of Electrical & Mechanical Engineering (CEME)

Problem Statement

"Audio Players" are defined as a media player explicitly designed to play audio files, with limited or no support for video playback. An In-car audio player is an audio player among many other types of audio players. Every audio player has a manufacturer name and a model number. The In-car audio players have a storage media which can be mounted or unmounted. When a media is mounted, player starts playing the first song and then keeps track of current song being played and on unmounting it resets the song tracking. When a song is played, title of the current song is also displayed. Storage media stores a list of songs and it can be of two types namely CD and USB. CD and USB are storage medias, where CD and USB has a fixed storage capacity. A CD can have a limited number of Songs where each song has a unique title and a USB can have a number of Songs with unique titles. The In-car media player only supports CDs of MP3 and USBs up to 16GBs only. If any other CD type or larger USB is mounted the player displays an error message explaining the problem. The In-car media player (discussed in the above scenario) can also be powered on or off. When the player is turned off, its current state (including everything) is persisted in it. And when it is powered back on, that already persisted state is restored.

Now Extend the class 'Audio Player' you designed before with an updated datatype 'Audio PlayerV2' capable to perform following functionalities in addition to the functionalities provided in 'Audio Player' class.

- Shuffle playlist – capable to shuffle your current playlist randomly
- Save playlist – capable to save a play list on disk
- Load playlist – capable to load any of the previously created playlists
- Add new audio files to playlist
- Delete files from playlist
- Search an audio file

2. Design an efficient GUI for your application

Introduction

An In-car media player is a device or system that allows drivers and passengers to play and control various forms of media, such as music and videos while in a vehicle. These systems can be built into the car by the manufacturer or can be added as aftermarket devices. They typically include features such as USB and can be controlled through buttons on the steering wheel, or the car's infotainment system. They also have support for various audio file formats, such as MP3 and WAV. In-car media players have different levels of integration with the car's systems.

Hardware/Software Required

Hardware: Laptop

Software: Visual Studio

Pseudo Code

The given problem describes a scenario where an "Audio Player" is defined as a media player explicitly designed to play audio files, with limited or no support for video playback. An "In-car audio player" is a specific type of audio player that can have a storage media (CD or USB) mounted or unmounted. When a media is mounted, the player starts playing the first song and keeps track of the current song being played. When the media is unmounted, the song tracking is reset. The In-car audio player only supports CDs of MP3 and USBs up to 16GBs only.

The problem also asks to extend the "Audio Player" class with a new class "AudioPlayerV2" which has additional functionalities like shuffle playlist, save playlist, load playlist, add new audio files to playlist, delete files from playlist and search an audio file.

The pseudocode provided above defines the basic structure of the AudioPlayer class which includes properties like manufacturer name, model number, storage media, current song, and

isPlaying. It also includes methods like mountStorageMedia, unmountStorageMedia, displayCurrentSong, powerOn and powerOff. The AudioPlayerV2 class extends the AudioPlayer class and adds new functionalities like shufflePlaylist, savePlaylist, loadPlaylist, addAudioFileToPlaylist, deleteFileFromPlaylist, searchAudioFile.

The GUI for the application can include features like displaying the current song, the list of songs in the storage media, options to mount/unmount storage media, options to power on/off the player, options to shuffle, save, load, add and delete files from the playlist, and a search bar to search for a specific audio file.

Header Files

```
#include<iostream>
#include<fstream>
#include<Windows.h>
#include<cstdlib>
#include<ctime>
#include<fileapi.h>
#include<exception>
#include<vector>
#include<dirent.h>
#include<string>
#include <algorithm>
#include "TotalTime.h" //User Defined Header File
```

Classes and Functions

Classes:

AudioPlayer

Song

AudioPlayerV2

Functions:

AudioPlayer:

```
class AudioPlayer
{
protected:
```

```

string ManufacturerName, ModelNum;
BOOLEAN PowerState;
LPCWSTR Drive = { L"E:\\ " };
UINT Check;
DWORD SectorsPerCluster, BytesPerSector, FreeClusters, TotalClusters;
BOOLEAN SuccessfulFileOpen = GetDiskFreeSpaceA("E:", &SectorsPerCluster,
&BytesPerSector, &FreeClusters, &TotalClusters);
BOOLEAN FileOpenSuccess = GetDriveType(Drive);
static int No_of_Songs;

```

public:

```
//CONSTRUCTORS
```

```
AudioPlayer()
```

```
AudioPlayer(string MAN, string MON, bool PS)
```

```
//GETTERS
```

```
string getManufacturerName()
```

```
string getModelNum()
```

```
UINT getCheck()
```

```
//SETTERS
```

```
void setManufacturerName(string MAN)
```

```
void setModelNum(string MON)
```

```
//MEMBER FUNCTIONS
```

```
void CheckDriveType()
```

```
void CheckDriveFreeSpace()
}
```

Song:

```

class Song :public AudioPlayer
{

```

```

public:
    string Name;
    time_t PlayTimeSec, PlayTimeMin, TotalTimeSec, TotalTimeMin, RemSec,
    RemMin;
    vector<string> Songlist;
    string CurrentSong;
    static int No_of_Songs;
    friend AudioPlayer;
    bool Shuffle = false;
    bool Repeat = false;

    //CONSTRUCTORS
    Song();
    Song(string N, time_t PTS, time_t PTM, time_t TTS, time_t TTM);
    void TurnOn();
    void TurnOff();
    void Play_Pause_Songlist(string GetFileName);
    int Next_Song_File();
    int Previous_Song_File();
    bool TurnOnShuffle();
    bool TurnOnRepeat();
    void shuffle();
    int AddSongsFromFile();
    int DisplayFile();
    int Search(string N);
    bool checkMP3(char name[10000]);
}

```

```

int Song::No_of_Songs = 0;

```

AudioPlayerV2:

```

class AudioPlayerV2 : public AudioPlayer, public Song
{
public:
    vector<string> playlist;
    string currentPlaylistName;
    string CurrentPlaylistSong;
    time_t SongPlayTimeSec, SongPlayTimeMin, SongTotalTimeSec, SongTotalTimeMin, SongRemSec,
SongRemMin;
    static int Playlist_Songs;
    bool FileMode = true;
    bool PlaylistMode = false;

    //CONSTRUCTORS
    AudioPlayerV2() : AudioPlayer();
    AudioPlayerV2(string MAN, string MON, bool PS, string CPN) : AudioPlayer(MAN, MON, PS);

    //GETTERS
    string getCurrentPlaylistName();
    //SETTERS
    void setCurrentPlaylistName(string CPN);
    //MEMBER FUNCTIONS

    void createNewPlaylist(string playlistName);
    void deletePlaylist();
    int addSong();
    void ResumeSong_Playlist();
    void PlaySong_Playlist(string song);
    void PauseSong_Playlist();
    void NextSong_Playlist();
    void PreviousSong_Playlist();
    int Next_Song_Playlist();
    int Previous_Song_Playlist();
    void removeSong();
    void displayPlaylist()
    void shufflePlaylist();
    void savePlaylist(string fileName);
    void loadPlaylist(string fileName);
    bool searchPlaylist(string fileName);
}

int AudioPlayerV2::Playlist_Songs = 0;

```

Graphic User Interface



x64	1/18/2023 5:01 PM	File folder	
Code.h	1/19/2023 9:35 PM	C/C++ Header	24 KB
MusicPlayerState.txt	1/19/2023 10:17 AM	Text Document	1 KB
MyForm.cpp	1/18/2023 5:43 PM	C++ Source file	1 KB
MyForm.h	1/19/2023 9:38 PM	C/C++ Header	37 KB
MyForm.resx	1/19/2023 9:38 PM	Microsoft .NET Ma...	213 KB
Playlist1.txt	1/19/2023 1:24 PM	Text Document	1 KB
Project GUI.sln	1/18/2023 5:01 PM	Visual Studio Solut...	2 KB
Project2.vcxproj	1/19/2023 12:14 AM	VC++ Project	7 KB
Project2.vcxproj.filters	1/18/2023 5:50 PM	VC++ Project Filte...	2 KB
Project2.vcxproj.user	1/18/2023 5:01 PM	Per-User Project O...	1 KB
TotalTime.h	1/9/2023 11:30 PM	C/C++ Header	2 KB

```

C:\Users\Azlaan\source\repos\Project2\x64\Debug\Project GUI.exe
Drive 00007FF6A9AD40B0 is type 2 - USB.
Bytes per sector:512
Sectors per cluster: 8
Free clusters: 1921709
Total clusters: 1996287
You have 7.32884 GB of free storage
You have 8 GB of total storage
Bytes per sector:512
Sectors per cluster: 8
Free clusters: 1921709
Total clusters: 1996287
You have 7.32884 GB of free storage
You have 8 GB of total storage
open "E:\Music\Besharam Rang Song Pathaan Shah Rukh Khan Deepika Padukone Vishal Sheykhar Shilpa Kumar.mp3" type m
pegvideo alias mp3

```

Conclusion

Based on the knowledge that has been gained by us in the process of the making of this project. We have been able to immaculately showcase various concepts of file handling, polymorphism, directory and file reading and media file handling which was dependent on a number of concepts. These were on a spectrum of things that we learned in this course and also external knowledge. A unique experience was the GUI design which has been an added skill set to compliment coding capabilities of the team. Lastly, we were better able to learn about the workings of media players and streaming applications in general and In-car media player in particular.

