

Urządzenia Peryferyjne

Karta muzyczna

Nikita Stepanenko, 245816

Termin zajęć Środa 11:00 TP

1. Cel ćwiczenia

Celem wykonywanego ćwiczenia było stworzenie programu, który będzie umożliwiać:

- a) wybór pliku od odtworzenia
- b) wybór odtwarzania wybranego pliku metodą:
 - 1. PlaySound
 - 2. Windows Media Player (WMP)
 - 3. WaveOutWrite
 - 4. MCI
 - 5. Direct Sound

Dla każdej metody należało umieścić dwa przyciski START i STOP oraz (jeżeli metoda umożliwia) przycisk PAUSE.

Następnym celem było rozszerzyć ten program w części Direct Sound o możliwość dodawania efektów dźwiękowych.

Ostatnim celem było rozszerzyć program o możliwość nagrywania dźwięku jedną z metod: WaveInWrite, MCI lub DirectSound

2. Wstęp teoretyczny

Karta muzyczna – komputerowa karta rozszerzeń umożliwiająca rejestrację, przetwarzanie i odtwarzanie dźwięku. Obecnie układy dźwiękowe wystarczające do zastosowań amatorskich są zazwyczaj wbudowywane w płytę główną komputera, a nie stanowią karty rozszerzenia. Z powodów historycznych są jednak określane mianem „zintegrowana karta dźwiękowa”. Pojawiły się również zewnętrzne karty dźwiękowe podłączane do komputera przez port USB.

3.Realizacja ćwiczenia

W trakcie zajęć udało się wykonać następujące zadania:

1. Napisać program, który umożliwia odtwarzanie wybranego pliku typu WAV następującymi metodami:

- a) PlaySound
- b) Windows Media Player
- c) MCI
- d) DirectSound.

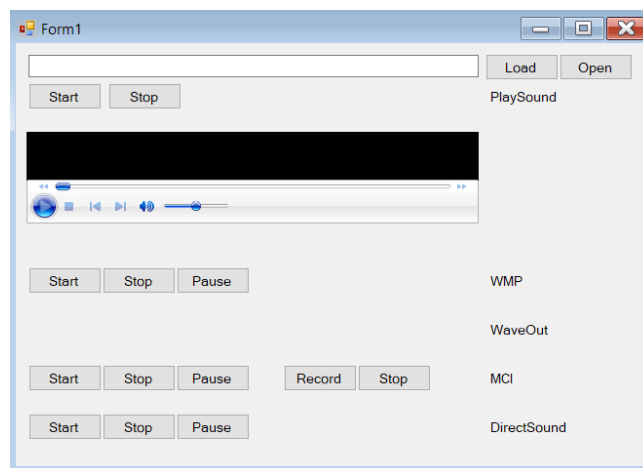
2. Rozszerzyć ten program o możliwość nagrywania dźwięku metodą MCI.

Program został napisany z wykorzystaniem języka obiektowego C#. Po uruchomieniu programu mamy możliwość wybrać plik typu WAV na komputerze, potem mamy możliwość do odtwarzania tego pliku potrzebną nam metodą. Są przyciski START oraz STOP, które posługują dla początku odtwarzania pliku oraz kończenia odpowiednio.

W programie realizowano 4 klasy dla obsługi plików wav: MyDirectSound, MyPlaySound, MyMCI.

1 klasa dla obsługi okna głównego oraz 1 klasa dla uproszczenia pracy z plikiem.

Dla interfejsu graficznego wykorzystywany Windows Forms.(System.Windows.Forms).



Klasa obsługi okna:

Ta klasa jest bardzo prosta. Po naciśnięciu odpowiedniego przycisku w oknie programu jest wywołana odpowiadająca metoda, która wywoła obiekt potrzebującej metody i wykonuje pewną funkcję. Na przykład: naciskamy „Start” dla metody MCI. Funkcja wywoła funkcję obiektu myMCI która już odtwarza plik.

```
private void buttonStart_Click(object sender, EventArgs e)//MCI
{
    myMCI.startMCI();
}
```

Realizacja metody PlaySound:

Biblioteki:

```
System;
System.Threading;
System.Security.Permissions;
```

```
System.Runtime.InteropServices;
```

```
[System.Runtime.InteropServices.DllImport("winmm.DLL", EntryPoint = "PlaySound",  
SetLastError = true, CharSet = CharSet.Unicode, ThrowOnUnmappableChar = true)]
```

Odtwarzanie pliku. Wykorzystam dodatkowy wątek żeby główny program działał niezależnie.

```
2 references  
private void run()  
{  
    PlaySound(fileName, new System.IntPtr(), PlaySoundFlags.SND_SYNC);  
}  
  
1 reference  
public void startPlaySound()  
{  
    threadPlaySound = new Thread(run);  
    threadPlaySound.Start();  
    run();  
}
```

Stop. Tu mam jakiś problem z zamknięciem wątku, dlatego nie działa, chociaż musi.

```
[SecurityPermissionAttribute(SecurityAction.Demand, ControlThread = true)]  
1 reference  
public void stopPlaySound()  
{  
    threadPlaySound.Abort();  
    PlaySound("", new System.IntPtr(), PlaySoundFlags.SND_ASYNC);  
    threadPlaySound.Abort();  
}
```

Realizacja metody WindowsMediaPlayer:

Biblioteki:

WMPLib

axWMPLib

Odtwarzanie pliku.

```
axWindowsMediaPlayer1.URL = mediaFile.Path;  
axWindowsMediaPlayer1.Ctlcontrols.stop();
```

Start/Play.

```
axWindowsMediaPlayer1.Ctlcontrols.play();
```

Stop.

```
axWindowsMediaPlayer1.Ctlcontrols.stop();
```

Pause.

```
axWindowsMediaPlayer1.Ctlcontrols.pause();
```

Realizacja metody MCI:

Biblioteki:

System;
System.Text;

```
[System.Runtime.InteropServices.DllImport("winmm.dll")]  
private static extern int mciSendString(string strCommand,  
    StringBuilder strReturn, int iReturnLength, IntPtr hwndCallback);  
[System.Runtime.InteropServices.DllImport("winmm.dll")]  
public static extern int mciGetErrorString(int errCode,  
    StringBuilder errMsg, int buflen);
```

Odtwarzanie pliku.

```
mciSendString("open waveaudio!" + fileName + " alias myaudio wait", null, 0, IntPtr.Zero);
```

Start/Play.

```
mciSendString("play myaudio", null, 0, IntPtr.Zero);
```

Stop.

```
loadMCI(fileName);  
mciSendString("stop myaudio", null, 0, IntPtr.Zero);
```

Pause.

```
mciSendString("pause myaudio", null, 0, IntPtr.Zero);
```

Nagrywanie.

```
mciSendString("open new Type waveaudio alias recsound", null, 0, IntPtr.Zero);  
mciSendString("record recsound", null, 0, IntPtr.Zero);
```

Stop nagrywanie.

```
mciSendString("save recsound 123.wav", null, 0, IntPtr.Zero);  
mciSendString("close recsound ", null, 0, IntPtr.Zero);
```

Realizacja metody DirectSound:

Biblioteki:

System;
Microsoft.DirectX.DirectSound;
System.IO;

Odtwarzanie pliku.

```
Device dSound = new Device();  
dSound = new Device();  
dSound.SetCooperativeLevel(handle, CooperativeLevel.Normal);  
BufferDescription bufferDescription = new BufferDescription();  
bufferDescription.ControlPan = false;  
bufferDescription.ControlVolume = false;  
bufferDescription.ControlFrequency = false;  
bufferDescription.ControlEffects = false;  
sound = new SecondaryBuffer(fileName, bufferDescription, dSound);
```

Start/Play.

```
sound.Play(0, BufferPlayFlags.Default);
```

Stop.

```
sound.Stop();  
sound.SetCurrentPosition(0);
```

Pause.

```
sound.Stop();
```

3.Wnioski

W trakcie wykonywania napotkaliśmy na problem awaryjnego zamykania wątku, tzn nie da się zamknąć go.

Bardzo przydatne były uwagi dotyczące DirectSound.