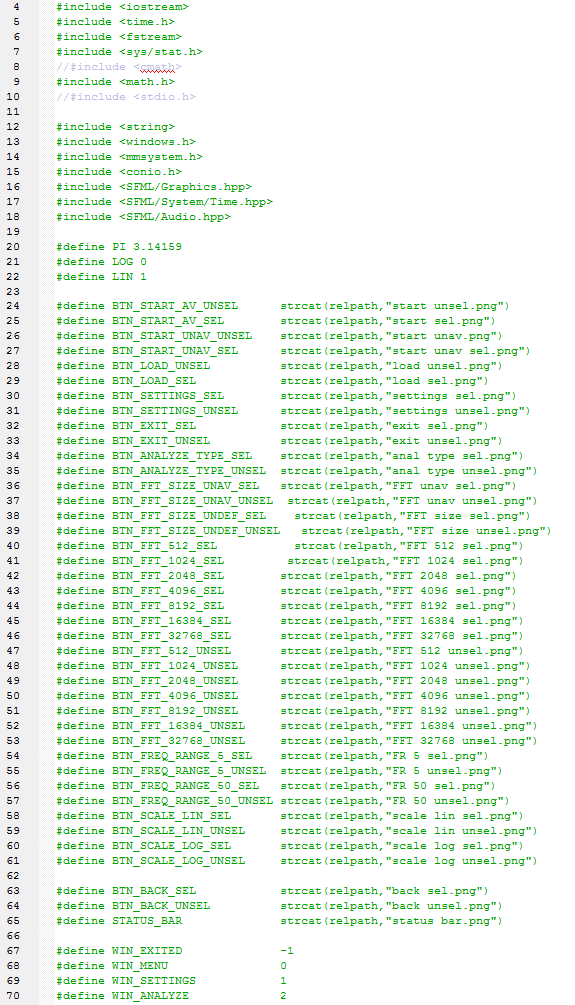
Programul prezentat utilizeaza algoritmi de tip Fourier Transform pentru a prelucra fisiere audio si a analiza gama de frecvente aparute intr-o anumita secventa de sunet. Fisierul main.cpp are peste 1100 de linii, si zeci de functii care fac diferite lucruri. Voi prezenta functiile si macrocomenzile in ordinea in care apar in fisier:

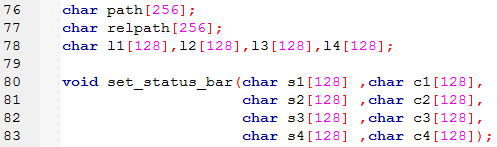
1. Macrocomenzi de preprocesare:



Aceste macrocomenzi sunt folosite pentru includerea headerelor in program si pentru definirea unor comenzi ce usureaza utilizarea lor ulterioare.

Astfel, #define BTN\_(nume) este folosit pentru a specifica calea absoluta a fisierelor imagine ce vor fi folosite pentru afisarea butoanelor din meniul principal.

#define WIN\_ este folosit pentru specificarea ferestrei in care este activ programul, pentru afisarea corecta a meniurilor si ferestrelor.

1. Declarari de variabile globale si antete de functii:

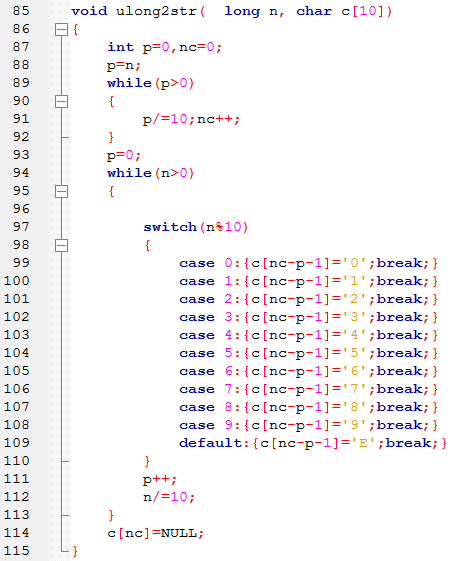
Sirul de caractere path retine calea absoluta a folderului in care este executat preogramul.

Sirul de caractere relpath este folosit pentru concatenarea numelui unui fisier la calea absoluta a folderului cu fisiere.

Sirurile de caractere 11,12,13,14 (l este

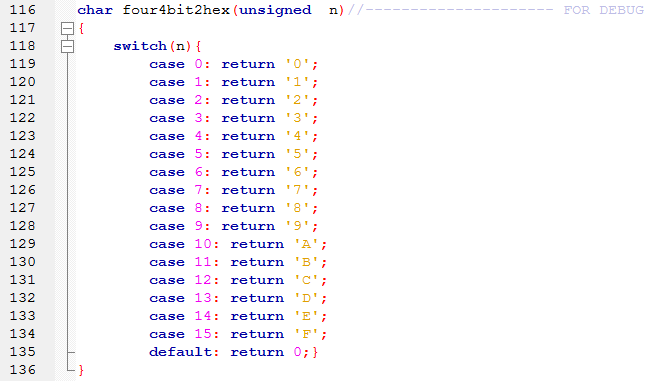
litera) sunt folosite in cele 4 randuri a barii de status.

1. Functia ulong2str:

Aceasta functie primeste un numar de tip long si pune in parametrul c un sir de caractere reprezentand cifrele numarului primit.

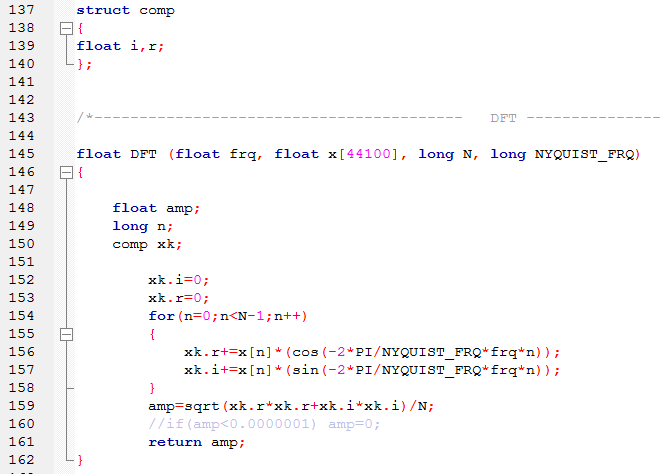
Va fi utilizata pentru afisarea numerelor sub forma de sir de caractere pe ecranul de lucru.

1. Functia four4bit2hex



Aceasta functie converteste un numar din intervalul 0-15 in baza 16.

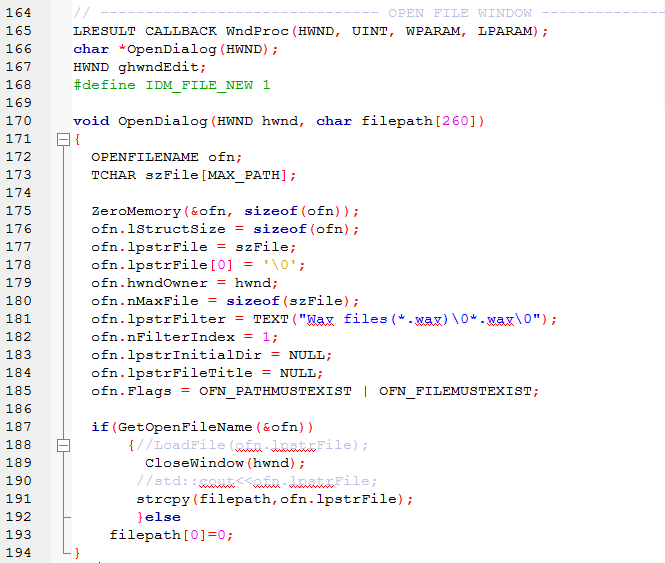
A fost folosita pentru debug la citirea header-ului fisierelor wav si a valorilor din „samples”.­­­­­­­

1. Functia DFT (Direct Fourier Transform)

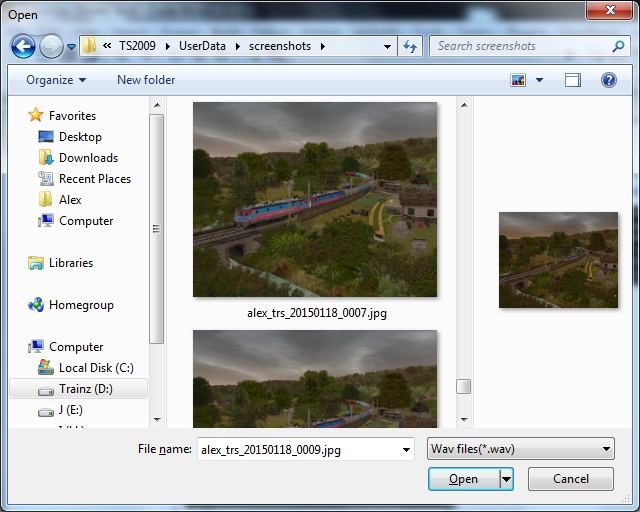
Secventa incepe cu declararea globale a structurii comp, reprezentand un numar complex.

Functia DFT primeste un sir de valori reprezentand amplitudinea semnalului audio, un numar N reprezentand numarul de valori din sir si un numar frq reprezentand frecventa careia i se va calcula amplitudinea. Functia returneaza amplitudinea frecventei frq in sirul dat.

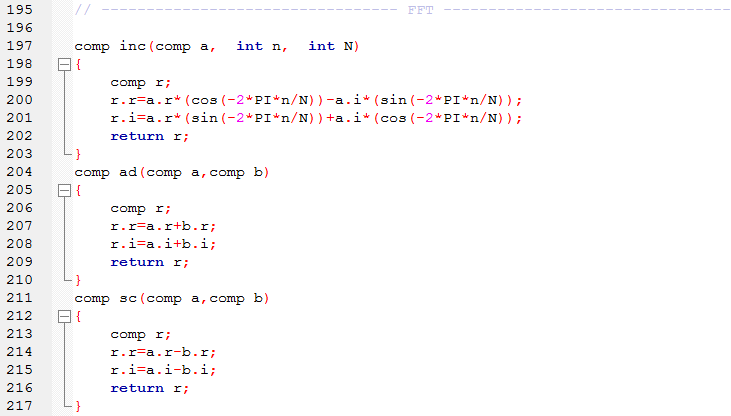
Algoritmul functioneaza adunand fiecare produs dintre o valoare din sir si un numar complex **e­­‑2\*PI/(SAMPLE\_RATE/2)\*frq\*n**­, unde n este pozitia curenta in sir, iar SAMPLE\_RATE reprezinta numarul de „samples” intr-o secunda de audio.

1. Functia OpenDialog

Aceasta este o functie de sistem a Windows care deschide fereastra Open File. Returneaza adresa fisierului WAV deschis.



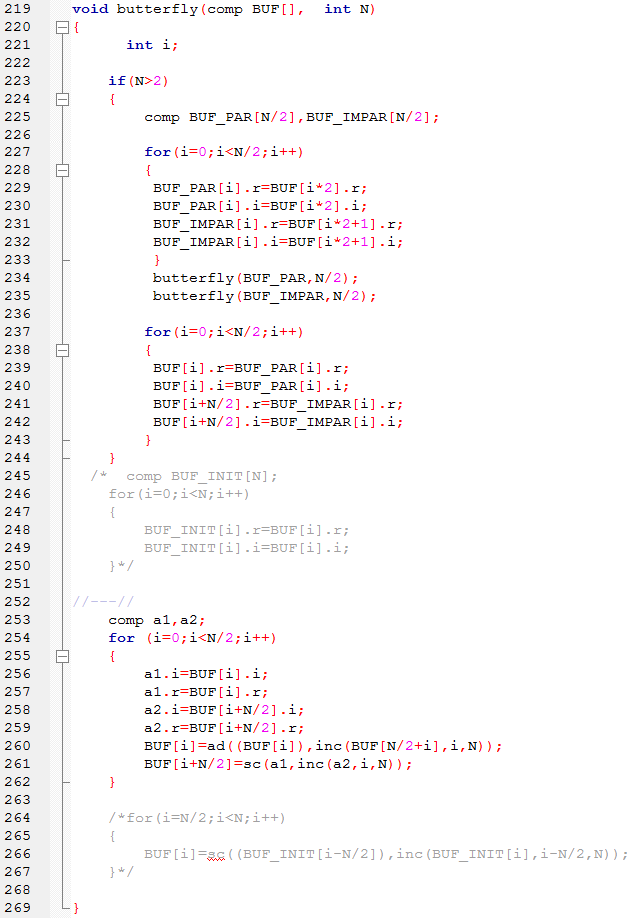
1. Algoritmul FFT (Fast Fourier Transform)



Algoritmul FFT necesita mai multe functii, printre care functii care efectueaza operatii aritmetice intre numere complexe.

Functia inc inmulteste un numar complex a cu un Twiddle Factor. Pentru a intelege ce face aceasta functie, va rugam consultati bibliografia cu privire la algoritmul FFT.

Functia ad aduna doua numere complexe, iar functia sc scade doua numere compelxe.

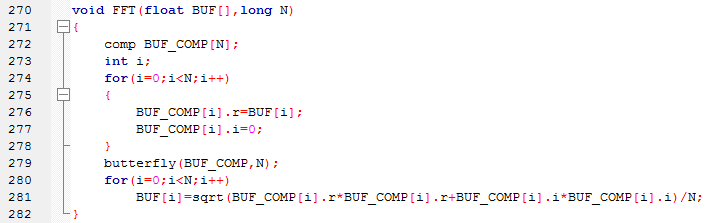


Functia butterfly este partea de baza a algoritmului FFT.

Aceasta este o structura recursiva de tip Divide et Impera si prelucreaza sirul impartind-ul in doua parti egale, unul format din elementele din sirul initial ce se afla pe pozitii pare, iar altul – pozitii impare (224-233) daca lungimea sirului initial este mai mare de 2 (223). Dupa eventuala impartire a sirului, acesta este reconstruit din cele doua siruri (237-242), doar dupa prelucrarea lor printr-un apel al aceleiasi functii pentru fiecare dintre ele (234-235).

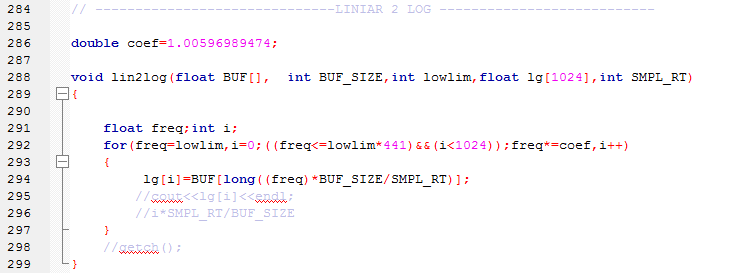
Se aplica apoi algoritmul propriu-zis al FFT-ului, si anume o serie de inmultiri si adunari cu factori complecsi.

Portiunile in comentarii sunt portiuni de cod ce au fost ulterior optimizate sau integrate in alte parti de cod.

Functia principala a FFT.

Foloseste un sir de numere complexe in care se transfera numerele din sirul buffer. Se apeleaza functia butterfly pe sirul de numere complexe, sir care apoi se converteste in modul de numar complex in sirul primit ca parametru.

1. Functia lin2log

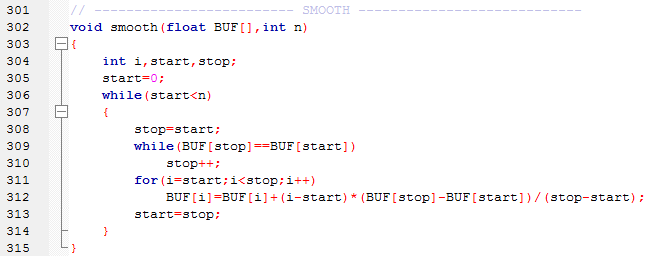


Aceasta functie va fi folosita pentru aducerea de la scara liniara la scara logaritmica in cadrul afisarii spectrogramei.

Primeste sirul returnat de functia FFT si parametri precum frecventa de baza (cea mai joasa, prima din

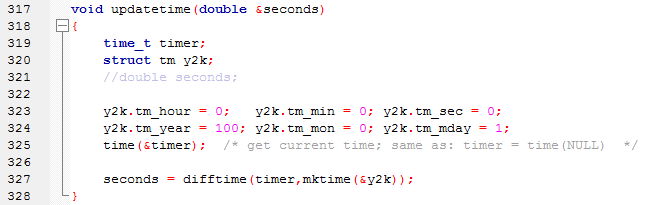
sirul nou creat) si sample rate-ul.

1. Functia smooth



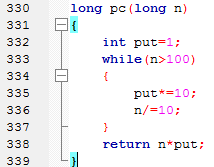
Aceasta functie atenueaza diferentele de amplitudini din sirul de amplitudini creat de FFT, adica va „nivela” amplitudinile ce se repteta datorita sensibilitatii scazute a FFT-ului pentru siruri de lungimi mici sau la frecvente joase.

1. Functia updatetime



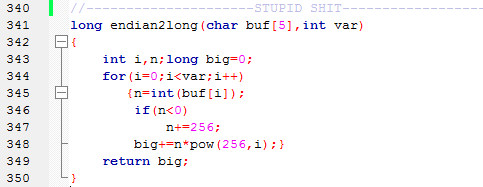
Nu mai este folosita. Inlocuita de functiile din clasa sf::Time.

1. Functia pc

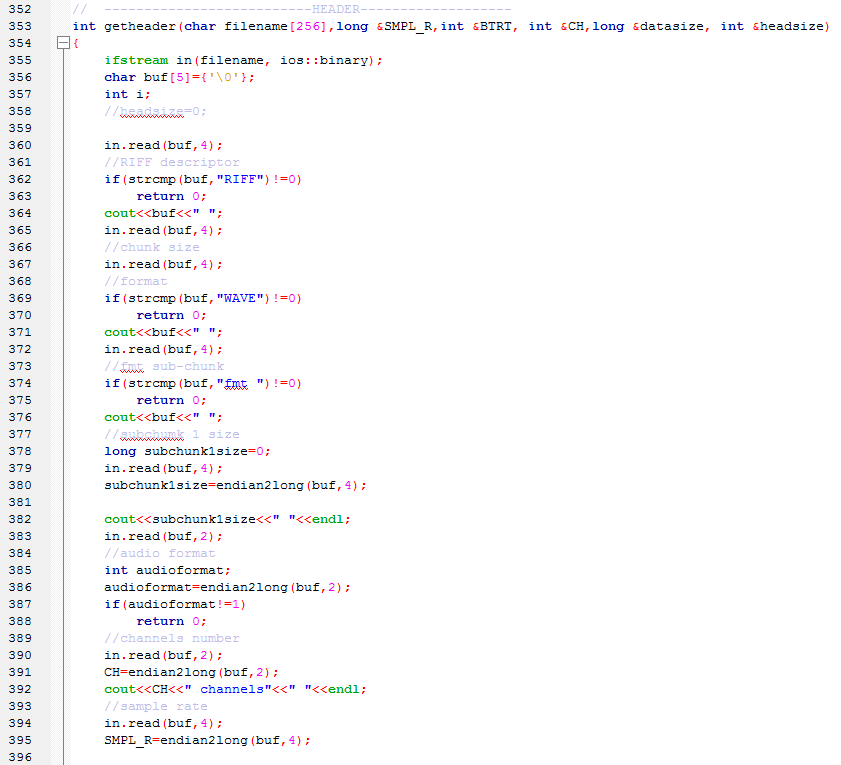
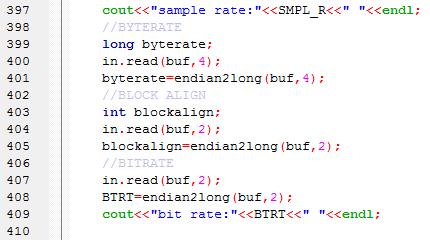
Functia este folosita pentru rotunjirea numerelor la 2 cifre semnificative.

Folosita in fereastra de ecranare a FFT-ului.

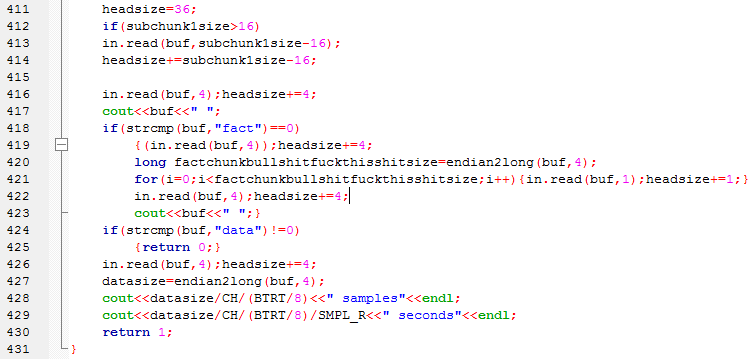
1. Functia endian2long

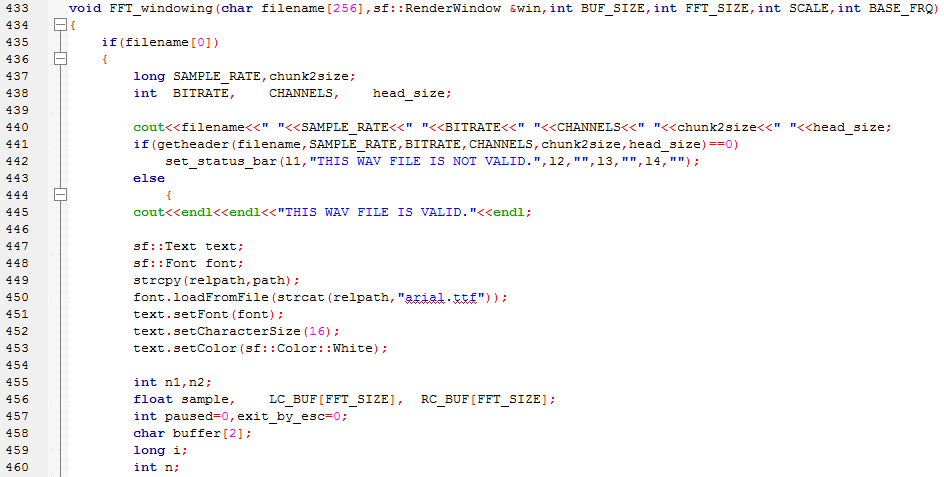


Folosita la conversiunea dintr-un sir de caractere codificat in little sau big endian intr-un numar de tip long.

1. Functia getheader

Aceasta functie primeste calea unui fisier WAV si verifica validitatea acestuia si extrage date despre fisier precum SAMPLE RATE, BIT RATE, numarul de canale, lungimea headerului si a fisierului s.a.



1. Functia FFT\_windowing

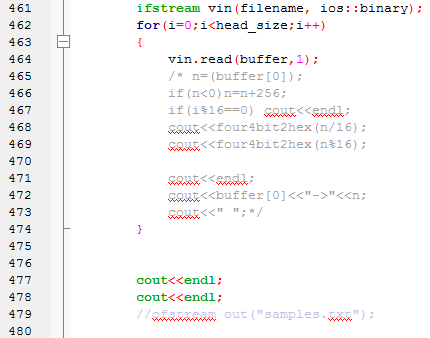
Functia FFT\_windowing este cea mai importanta functie din program.

Aceasta se ocupa de la citirea fisierului WAV pana la analiza acestuia si afisarea pe ecran a spectrogramei.

Aceste prime 25 de randuri reprezinta declarari de variabile de toate felurile, precum si verificarea daca fisierul citit este valid (441).

Sunt declarate siruri de valori numerice pentru (maxim) cele doua canale audio (stanga - dreapta), bufferul de citire si diferiti paramteri si contori ce vor fi utilizate mai departe.

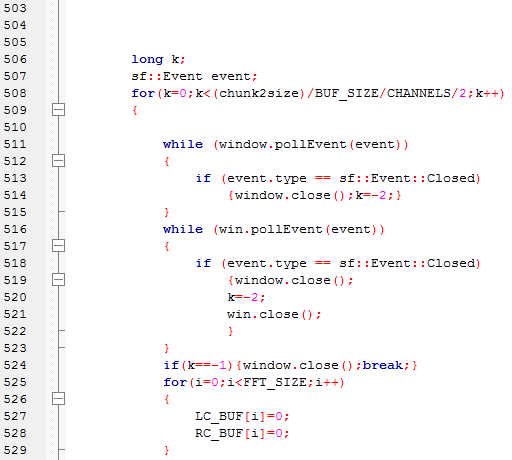
Pe liniile 462-480 se sare peste header-ul fisierului, citit deja la linia 441. Comentariile sunt ramase de la partea de debuging, afisandu-se header-ul in forma hexadecimala.





Linia 482 construieste fereastra pe tot ecranul, inlocuind functia initwindow a WinBGI.

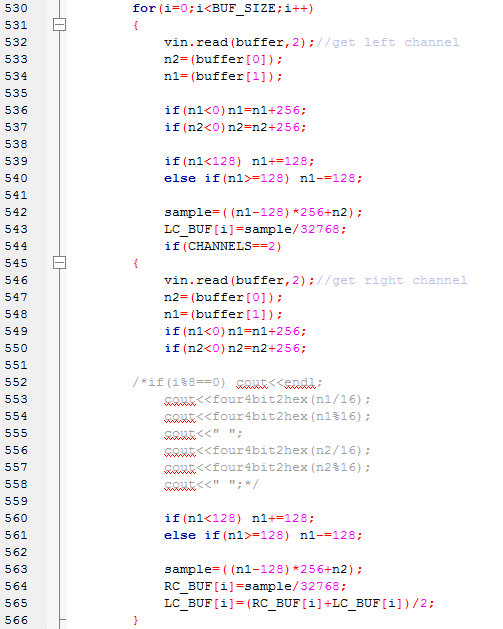
Inca 15 randuri de declarari, printre care o „muzica”, niste „ceasuri” si alte variabile.



Linia 501 constituie conditia de iesire din fereastra nou creata. Desi nu se va mai verifica niciodata, este necesara pentru verificarea daca fereastra s-a deschis corect.

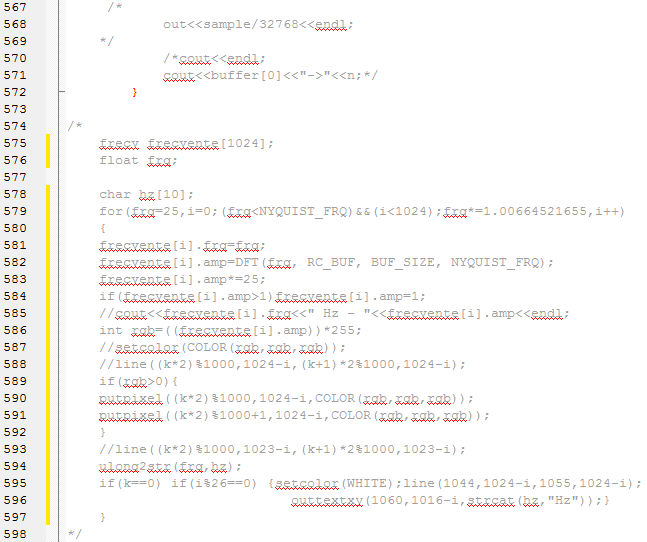
Linia 507 contine o declarare foarte importanta: un eveniment, adica va face programul sa raspunda la apasarea butonului , asta in conditii verificate periodic (511-515 pentru fereastra analizei, 516-522 pentru fereastra meniului).

Liniile 525-529 initializeaza bufferele cu 0 pentru a nu se utiliza valori reziduale in cadrul analizei FFT.

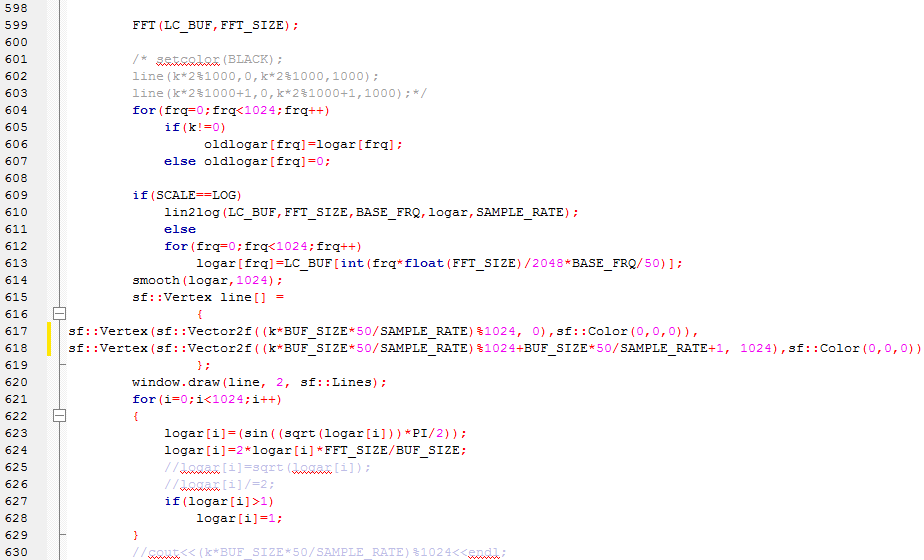
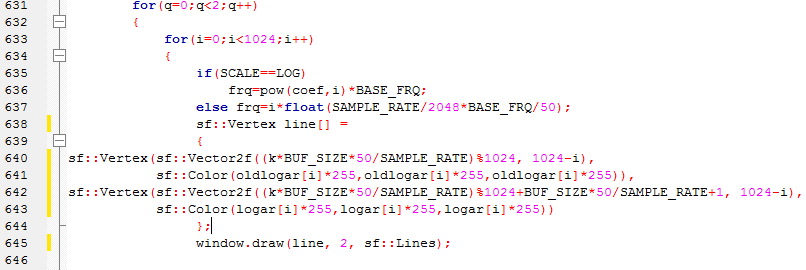


Portiunea de cod cuprinsa intre liniile 530-572 face citirea bufferului din fisierul audio, buffer a carui lungime este stabilit de marimea FFT-ului (lungimea sirului introdus in FFT; trebuie se fie putere a lui 2).

Citirea se face in mod binar, urmand ca cele doua valori (2 octeti) sa fie convertiti intr-un numar din intervalul (-1,1].



Liniile 574-598 cuprind vechea implementare, in modul grafic WinBGI, care a fost inlocuita cu o librarie grafica mai performanta



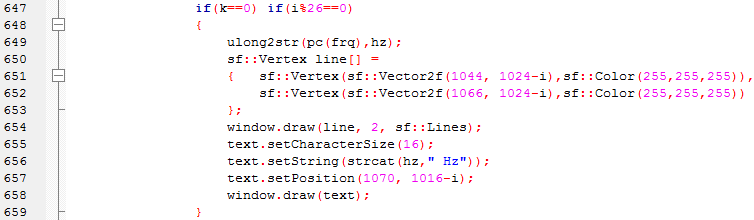
Se foloseste functia smooth (614) pentru a „nivela” sirul, evitand aspectul patratos la frecvente mici.

Portiunea 615-620 traseaza o linie neagra peste locul in care se va afisa urmatorul sir de pixeli.

Portiunea 621-629 modifica amplitudinile astfel incat frecventele cu amplitudini mici dar nenule sa fie mai usor observabile.

Se apeleaza functia FFT (599), iar apoi se construieste sirul logar, sir ce contine cate o valoare reprezentand o amplitudine a unei frecvente pentru fiecare din cei 1024 de pixeli ai inaltimii ferestrei de redare.Sirul se construieste din sirul returnat de functia FFT(604-607), utilizand eventual functia lin2log pentr a aduce sirul la scara logaritmica (609-613).

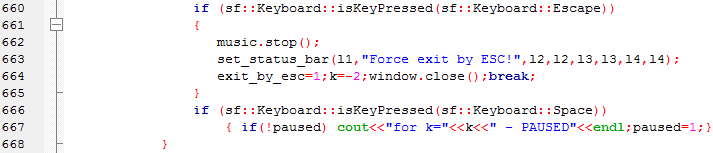
Se foloseste contorul i (633) pentru a parcurge fiecare pozitie in sirul logar, deci pentru a marca fiecare pozitie in inaltimea ferestrei.



Pe liniile 635-637 se calculeaza frecventa ce corespunde fiecarei pozitii din sirul logar.

Pe liniile 638-644 se declara o linie ce va face legatura dintre pozitia in lungime a afisarii precedente si a celei curente, aceasta linie fiind desenata in linia 645.

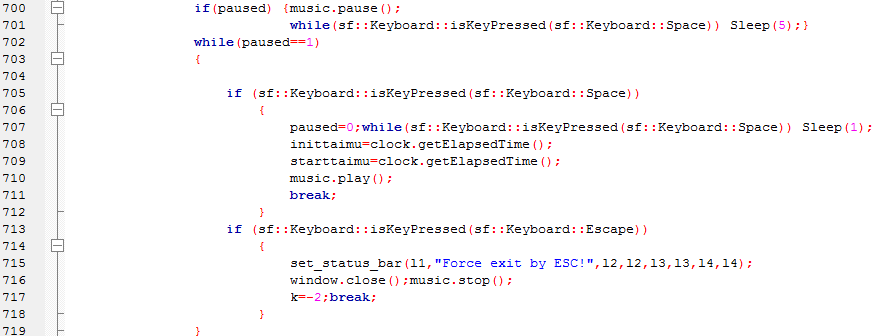
Liniile 647-659 construiesc grila de frecvente din dreapta ecranului, la un interval de 26 de pixeli.



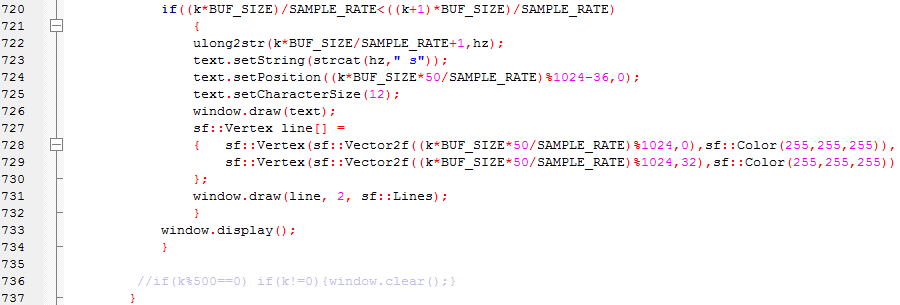
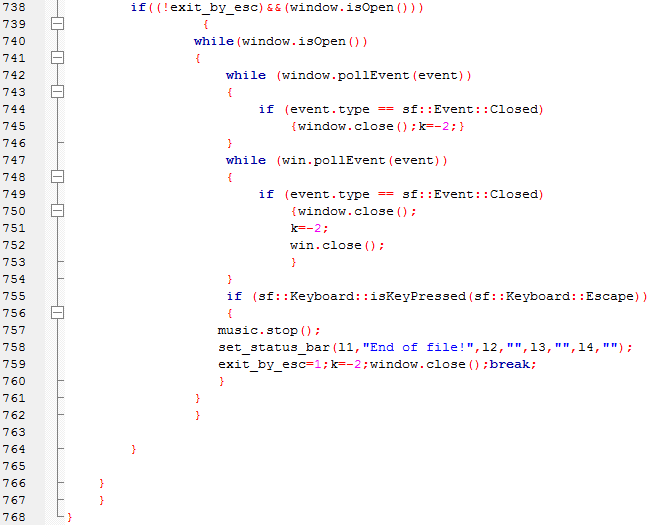


Secventa 660-668 are rolul de a reactiona la tastele Space si Esc, provocand pauzarea respectiv iesirea din fereastra de analiza.

Secventa 670-699 are rolul de a opri analiza in cazul in care intarzierea dintre muzica si afisare este prea mare.



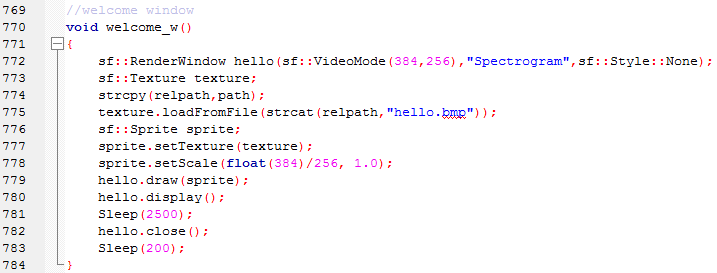
Portiunea 700-719 se ocupa de situatia in care analiza a fost pauzata, asteptand din nou tasta Space pentru a relua sau tasta Esc pentru a iesi.



Liniile 738-762 sunt executate cand se ajunge la finalul fisierului audio, facand ca programul sa astepte iesirea cu Esc pentru a inchide fereastra.

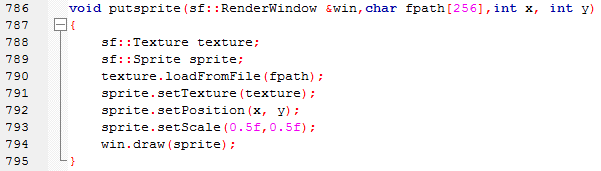
Secventa cuprinsa intre liniile 720 si 737 are rolul de a afisa grila cu timpul parcurs in partea de sus a ecranului.

1. Functia welcome\_w



Functia welcome\_w afiseaza o imagine de tip splash la pornirea programului.

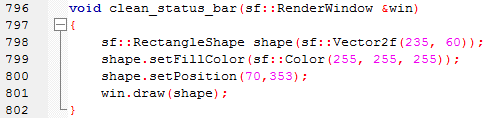
1. Functia putsprite



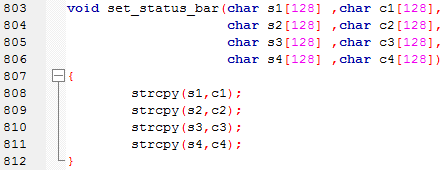
Aceasta functie afiseaza un „sprite” intr-o fereastra transmisa ca parametru, la pozitia x,y.

Va fi folosit in meniul principal.

1. Functia clean\_status\_bar

Aceasta functie curata bara de status pentru o afisare ulterioara.

1. Functia set\_status\_bar



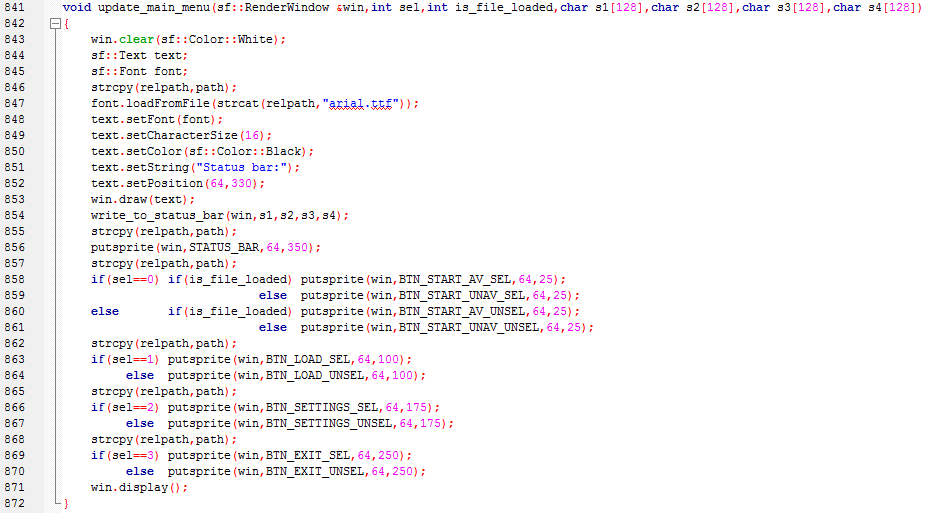
Aceasta functie atribuie celor patru linii de text din bara de status cate un sir de caractere dat ca parametru. Afisarea celor patru linii se va face ulterior.

1. Functia write\_to\_status\_bar



Aceasta functie afiseaza in bara de status cele patru linii de text, date prin paramteri.

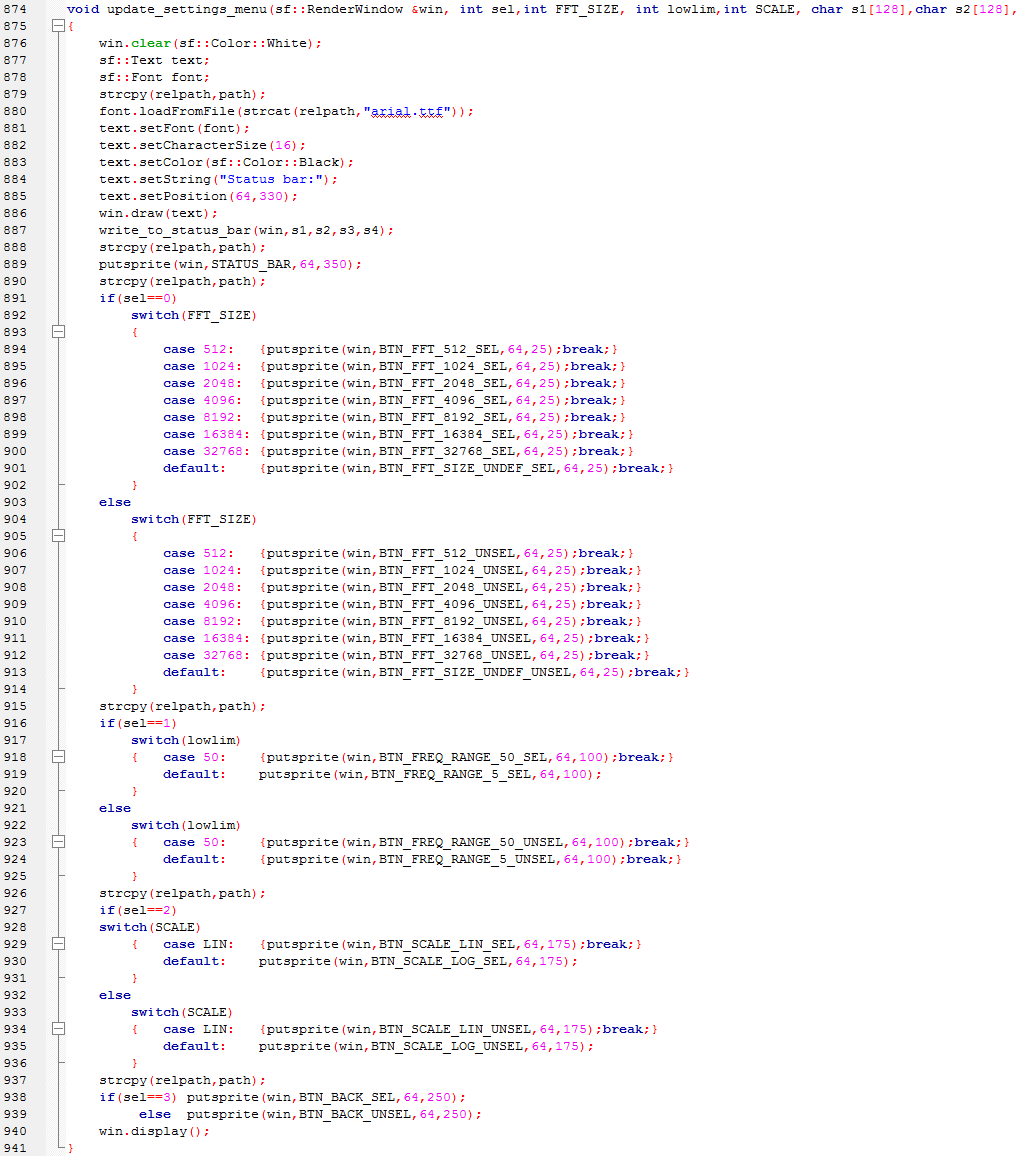
1. Functia update\_main\_menu reafiseaza meniul principal de fiecare data cand acesta se modifica.

Aceasta functie afiseaza butoanele meniului principal in functie de butonul selectat si de disponibilitatea anumitor butoane.

Fereastra aparuta la deschiderea programului

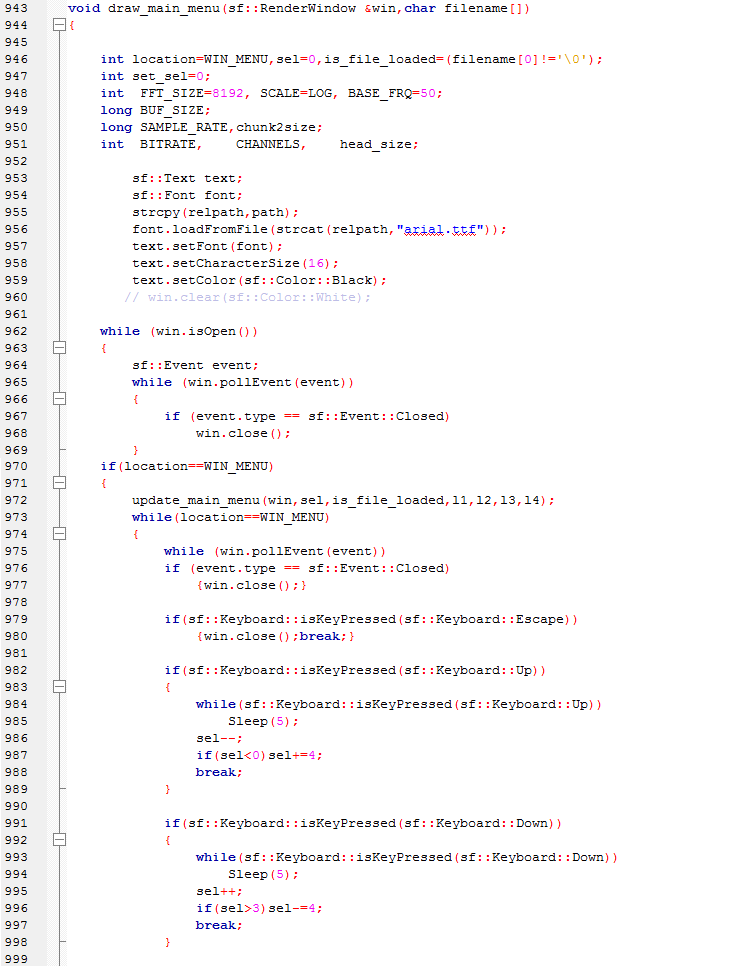


1. Functia update\_settings\_menu face acelasi lucru ca functia update\_main\_menu, insa functioneaza asupra meniului de setari.



1. Functia draw\_main\_menu

Aceasta functie se ocupa de comportamentul ferestrei meniului. Aceasta da acces la meniul principal, din care mai apoi se pot accesa meniul setari, fereastra de deschidere fisier si fereastra de analiza.



Liniile 946-960 reprezinta declarari, initializari si incarcari de fisiere.

Prin variabila location se stabileste care dintre ferestre este accesata la un moment dat, in functie de aceasta apelandu-se functiile update\_main\_menu sau update\_settings\_menu.

Tot in aceasta functie se reactioneaza la taste, si anume sageti si tasta Enter.

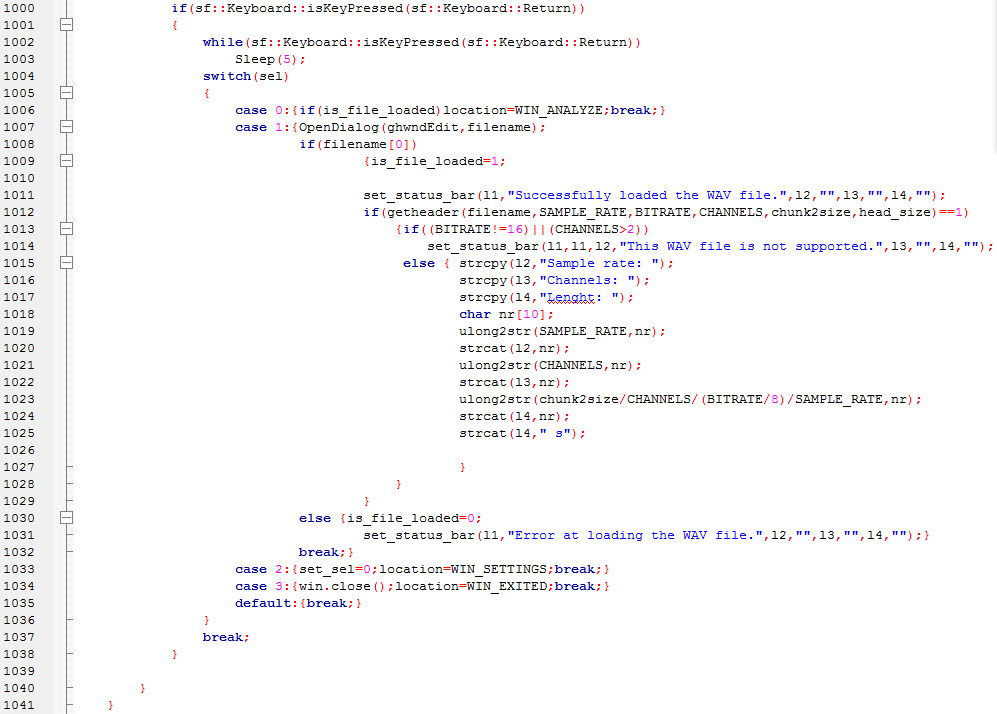
Liniile 1000-1041 sunt reactiile la tasta Enter in meniul principal care, in functie de butonul selectat poate face:

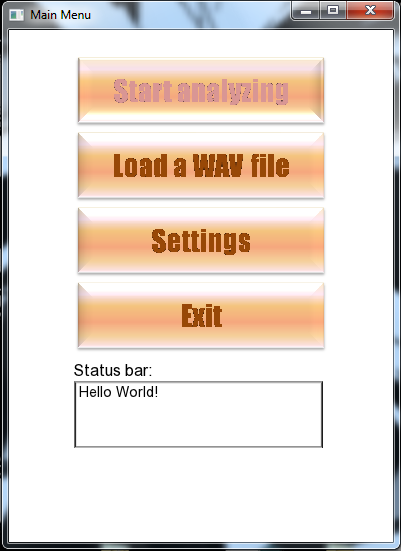
-sa deschida fereastra analizei

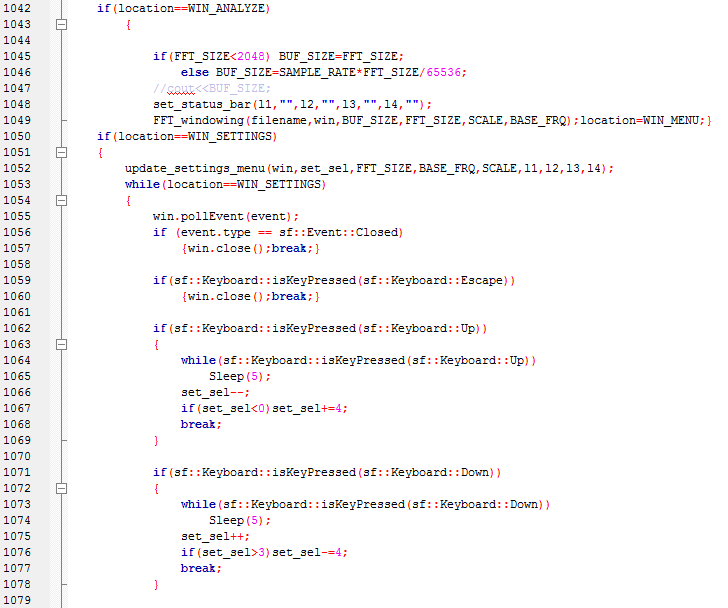
-sa incarce un fisier WAV

-sa deschida meniul setari

-sa inchida programul

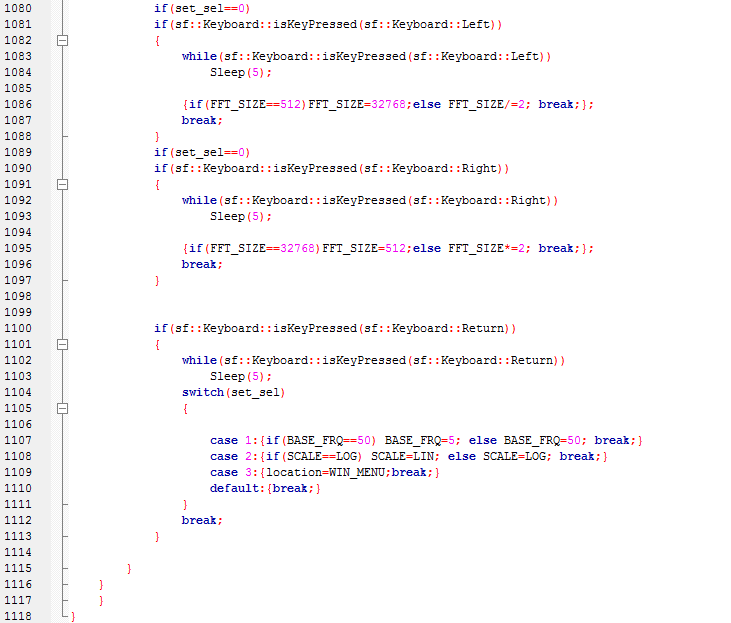
 Comutarea de la un meniu la altul se face prin iesire fortata din bucla de asteptare a unei comenzi si prin modificarea parametrilor ce memoreaza fereastra activa si butonul selectat.



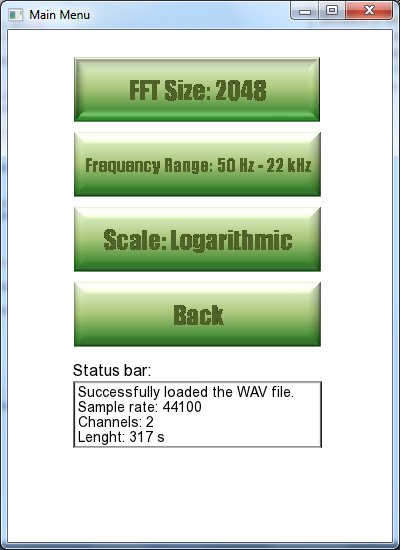


Liniile 1042-1049 prezinta schimbarea locatiei catre fereastra de analiza, intializand marimea bufferului si marimea FFT (setata in meniul setari) si apeland functia FFT\_windowing

Liniile 1050-1113 reprezinta reactiile la tastele sus, jos, stanga, dreapta si Enter in cadrul meniului Settings, impreuna cu modificarea parametrilor privind fereastra si selectia, dar si a parametrilor modificati in acest meniu, si anume dimensiunea FFT, scara si frecventa fundamentala.



Fereastra meniului de setari.



1. Functia main (in sfarsit!)

* relativ scurta si simpla
* afla calea folderului din care este executat programul si in care se afla fisierele dependente
* afiseaza ecranul de primire
* afiseaza meniul, unde mai apoi se intampla toata actiunea
* returneaza 0 daca totul merge bine

