



ALEGEREA TEMEI

- Sunt absolvent al sectiei de Orgă Electronică, Școala de Artă, clasa profesor muzicolog M.Ghircoias,
- Pasiunea pentru muzică și informatica a dus la realizarea de proiecte ce include ambele domenii.
- Blog despre aplicații noi muzicale.
- Compun muzică electronică în Linux folosind gratuit LMMS

- **ANALIZA PIEȚEI** - Există proiecte similare Online Launchpad, The DJ Dubstep Music Maker, HIP-HOP Drum-pads (Android)
- Configuratia dezvoltarii include AMD 6 Core, 3GHz, 4 MB RAM, two-channel sound card.
- Cerinte minime de rulare includ 1 GHz CPU, 512 MB RAM, placa de sunet.

- Un **Launchpad** este un Controller MID performant, multifuncțional și particularizabil. De regulă se folosește cu un DAW pentru [a compune și a Interpreta](#)
- **AVANTAJE** Launchpad inclusiv **MusicBoard**
 - Este un proces **artistic inovativ și live, în direct în fața publicului**
 - Se poate utiliza in **spectacole, concerte**
 - Ajută muzicianul să interacționeze cu computerul
 - Protejează ochii
 - Dupa o sesiune muzicală MusicBoard Launchpad / Launchkey, muzicianul are mult mai puțin de editat pentru compoziții muzicale.

PUBLICUL ȚINTĂ al aplicației sunt cei învață **MUZICA** și muzicieni.

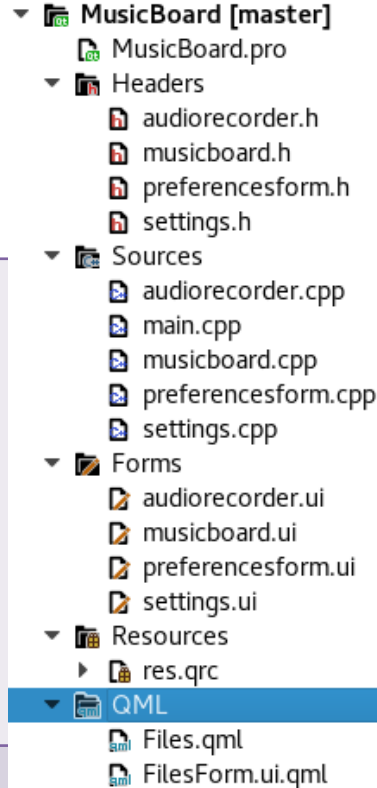
Tehnologii utilizate Qt5.1, C++, QtCreator (IDE), OpenGL, Audacity (muzica), Blender (grafica) in Linux 64 bit (Debian). > Licenta: Apache License (www.apache.org/licenses/LICENSE-2.0) > Imaginile sunt creatii originale ale autorului proiectului si sunt realizate folosind Blender gratuit (www.blender.org). > Toate sunetele folosite in proiect sunt gratuite (www.musicradar.com); unele editate cu Audacity gratuit (www.audacityteam.org).

Qt Framework C++ sistem inter-platformă de dezvoltare software; include o bibliotecă cu elemente de control pentru crearea de programe cu interfață grafică și fără interfață grafică, de ex. servere. Dezvoltare Cross-Platform în C++ inclusiv iOS si Android .

[Skype](#) - [Google Earth](#) - [Opera](#)

Design Blender- User Interface (UI)

- **3 ani experiență în aplicații grafice**
- aplicații de animație grafică în Blender inclusiv pentru Concursul NASA-Space Settlement din 2017
<http://wp-space-settlement.weebly.com/>



ETAPELE DE DEZVOLTARE MusicBoard

- Instalarea sistemului de operare Linux, Ubuntu
- Instalare Qt Framework C++
- Crearea proiectului MusicBoard, fișierul proiect are extensia **.pro**
- Includerea claselor Qt
- Realizarea interfeței cu utilizatorul folosind butoane și comenzi C++
- Includerea funcțiilor Qt specifice și realizate de autor
- Compilarea folosind fișierul Makefile ce include qmake specific Qt și rularea aplicației folosind fișierul **sh, comanda:** **./MusicBoard**

Proiectarea APLICAȚIEI MusicBoard

➤ PROGRAMARE ORIENTATĂ OBIECT

- Tehnologia QtFramework C++
- Extinderea proiectului se face ușor folosind obiectele existente în Qt și cele create de utilizator.

➤ FACILITĂȚI

- Cross-platform compatibility
- Protejează ochii și este ușor de folosit
- Este inclusiv o aplicație educativă (Computer-made Music)
- Dezvoltă Creativitatea
- **Nu** necesită conectare MIDI (ca și alte soft-uri muzicale)
- Compatibilitate cu multe formate de fișiere muzicale (**wav, flac, ogg, aac, mp3, aiff, wma, m4p, 3gp, vox**)

➤ ELEMENTE DISTINCTE

- Utilizarea Qt în Linux pentru implementare
- Originalitatea interfeței

➤ PLAN DE DEZVOLTARE:

- Salvarea ca fișiere muzicale a muzicii interpretate, Internal recording
- Includerea și realizarea de noi biblioteci media-muzicale Qt.
- Conectivitate MIDI.
- Customizare sunet Keyboard.
- Transpoziție; Long and Short Touches.
- **Testare** pe diverse **sisteme de operare**.
- Aplicația are posibilitatea să ajungă un external virtual MIDI controller pentru LMMS (www.lmms.io) software gratuit pentru compoziție muzicală în Linux.
- **MusicBoard** aplicație **online**.

Pentru UI:

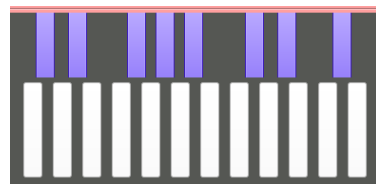
- `ui→File1→hideColumn(1);`
`//ascunde coloana 1 a file browser-ului`
- `QString sPath = "file:///home/n";`
`//adresa file browser`
- `on_C_clicked();` // apasare buton

Verificare tip de fișier:

```
int CheckWAV (QString a){
    QString v; int i, n; n=a.length();
    for (i=0;i<=n;i=i+1){ v[n-i]=a[i]; }
    if (v[1]=='v' && v[2]=='a' && v[3]=='w'
        && v[4]=='.') return 1;
    else return 0;
}
```

FACILITĂȚI ȘI LAYOUT TASTATURA

- Sunete default;
- Presetări Keyboard și Launchpad
- Sunete custom (alese de utilizator)
- Polifonie (nelimitată, dată de performanța computerului)
- 20 clape controlate de tastatură; 8 butoane de ritm (7 controlate de tastatură); 17 butoane de Launchpad controlate de tastatură



Pentru muzică:

- `QMediaPlayer * C = new QMediaPlayer();` // variabila pt. nota Do
- `QString C_Song;` // var. pr. filepath-ul fișierului muzical alocat
- `C→setMedia(QUrl(C_Song));`
`//alocare fișier muzical music player-ului`
- `C→play();` // redare muzică (acționare music player)
- `C_Song="qrc:/sounds/Keyboards/Keyboard2/2C.wav";`
`// alocare filepath către string-ul "C_Song"`
- `C→setMedia(QUrl(C_Song));` //alocare fișier muzical

Pentru alocare Filepath:

```
void MusicBoard::on_File2_clicked(const QModelIndex &index)
{
    QString FilePath = filemodel-> fileInfo(index).absoluteFilePath();
    FilePath = "file://" + FilePath;
    if (CheckWAV(FilePath)==1 ){
        if (setButton==1){P1_Song=FilePath;on_P1_clicked();}
        setButton=0; ui->Error->setText("Set");
    } else { ui->Error->setText("Error"); }
}
```

Filepath:
/006.wav

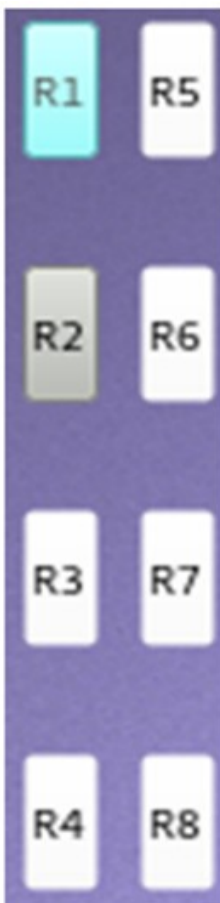
```
void MusicBoard::on_C_clicked()
{
    QMediaPlayer * C = new QMediaPlayer();
    C->setMedia(QUrl(C_Song));
    extern int KeyboardVol;
    C->setVolume(KeyboardVol);
    C->play();
}
```



MusicBoard

Linux Launchpad/Launchkey

Multiple Sound generator



```

void MusicBoard::on_R2_clicked()
{
    setButton=20;
    QMediaPlayer * R2 = new QMediaPlayer(
    ui->Rhythm2->show();
    on_StartRhythm_clicked();
    ui->Error->setText("R2");
    R2->setMedia(QUrl(R2_Song));
    extern int R2_A;
    R2->setVolume(R2_A);
    R2->play();
}

```

```

void MusicBoard::on_P4_clicked()
{
    setButton=4;
    QMediaPlayer * P4 = new QMediaPlayer();
    ui->Error->setText(QString::number(setButton));
    P4->setMedia(QUrl(P4_Song));
    extern int P4_A;
    P4->setVolume(P4_A);
    P4->play();
}

```

LaunchPad

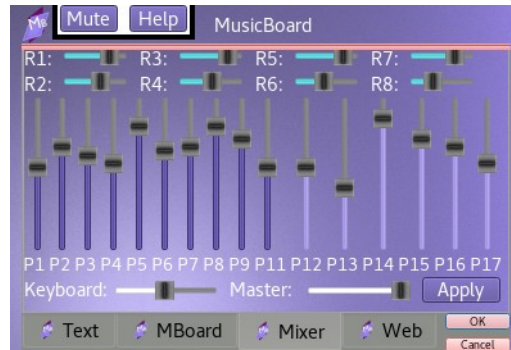
Show Shortcuts



MIXER

```
void Settings::on_P_1_sliderPressed()
{
    on_SettingsApplyMixerButton_clicked();
    MusicBoard a;
    a.on_P1_clicked();
}

void MusicBoard::on_VolSlider_valueChanged(int position)
{
    ui->VolSlider->setStyleSheet("QSlider {background-color: #000000; color: rgb(255, 255, 255);}");
    extern int P1_A;
    extern int P2_A;
    extern int P3_A;
    extern int P4_A;
```

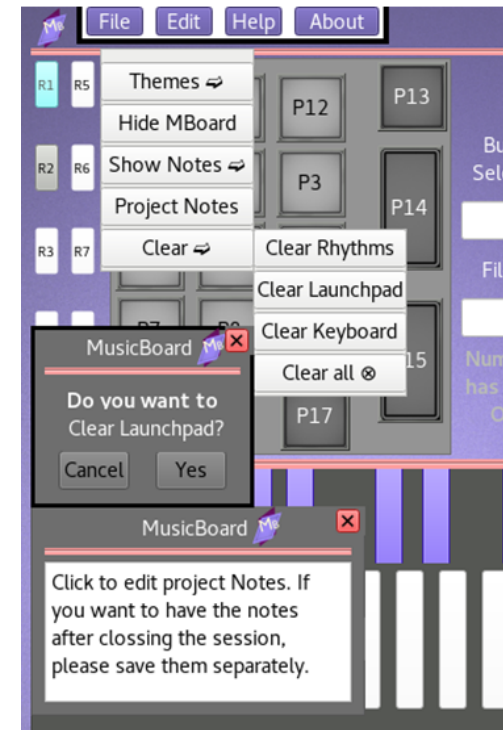
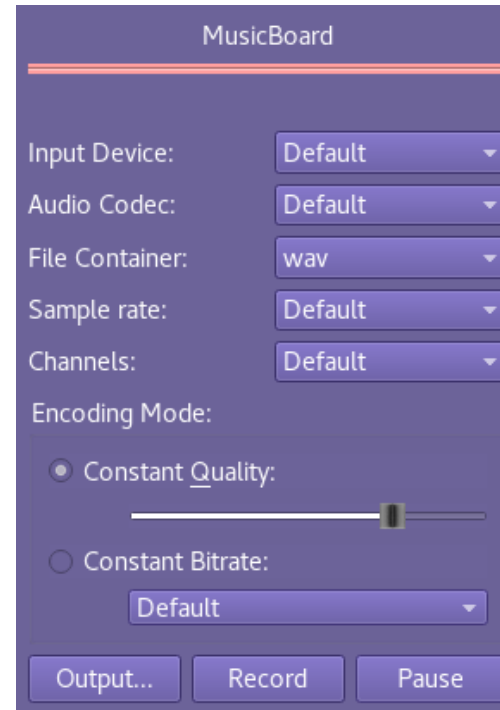


NumLock

```
void MusicBoard::on_P1_clicked()
{
    setButton=1;
    QMediaPlayer * P1 = new QMediaPlayer();
    ui->Error->setText(QString::number(setButton));
    P1->setMedia(QUrl(P1_Song));
    extern int P1_A;
    P1->setVolume(P1_A);
    P1->play();
    if (NumLock==10){
        ui->NumLockInfo->hide();
        ui->NmLkOnOff->setStyleSheet("color: rgb(255, 255, 255);");
        ui->NmLkOnOff->setText("OFF");
        NumLock=0;
    }
}
```



Recorder & Menu



Auxiliare

```
void MusicBoard::on_VolSlider_valueChanged(int position)
{
    ui->VolSlider->setStyleSheet("QSlider {background-color: #000000; color: rgb(255, 255, 255);}");
    extern int P1_A;
    extern int P2_A;
    extern int P3_A;
    extern int P4_A;
```

