

## ALEGEREA TEMEI

- Sunt absolvent al secției de Orgă Electronică, Școala de Artă, clasa profesor muzicolog M.Ghircoias,
- **Am ales tema muzicii, datorita interesului pentru orga electronică.**
- Compun muzică electronică în Linux cu LMMS.

- ANALIZA - Am efectuat studiul pieței realizând un blog dedicat aplicațiilor muzicale.
- După peste 100 aplicații muzicale postate am constatat lipsa pe piață a unui Launchpad / Launchkey pentru Linux.
- Există proiecte similare Online Launchpad, The DJ Dubstep Music Maker, HIP-HOP Drum-pads (Android)
- CONFIGURATIA dezvoltării include AMD 6 Core, 3GHz, 4 MB RAM, two-channel sound card.
- Cerințe minime rulare includ 1 GHz CPU, 512 MB RAM, placa de sunet.

- **Am realizat MusicBoard, un instrument muzical virtual inovativ**, live, în direct în fața publicului, **primul offline pentru Linux.**
- Un Launchpad este un Controller MIDI performant, multifuncțional și particularizabil. De regulă se folosește cu un DAW pentru [a compune și a interpreta](#).
- AVANTAJE MusicBoard
  - **Aplicația oferă un proces artistic inedit**, live, în direct în fața publicului,
  - **Ajută în compoziția muzicală**
  - **Facilitează interacțiunea computer-muzician**
  - Se poate utiliza în spectacole, concerte
  - După 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 muzicienii.

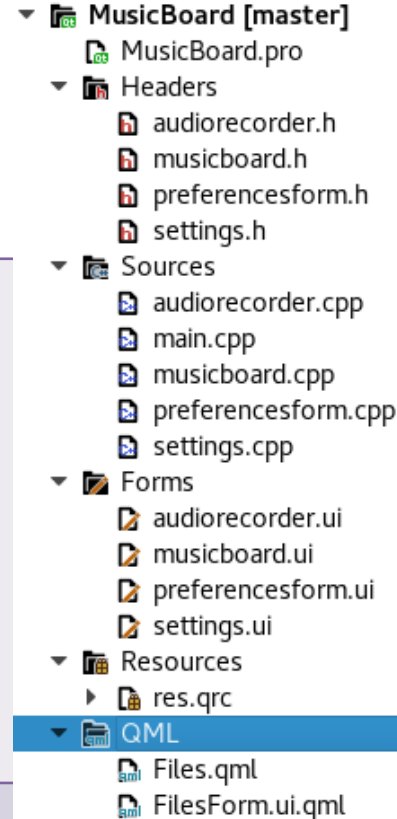
**Tehnologii** - Am utilizat framework-ul Qt folosit și la dezvoltarea Google Earth și Skype.

- Am utilizat LMMS și Audacity pentru editarea sunetului și Blender pentru grafica.
- Imaginile sunt originale, iar sunetele gratuite (www.musicradar.com);
- Licența: Apache License (www.apache.org/licenses/LICENSE-2.0)

**Qt** Framework: sistem inter-platformă de dezvoltare software în **C++**; include o bibliotecă cu elemente de control pentru crearea de programe cu interfață grafică și fără interfață grafică, de ex. servere

**Design** Blender- User Interface (UI)-3 ani experiență

- 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
- Instalare Qt Framework C++
- **Dupa instalările necesare, am realizat proiectul MusicBoard (extensia .pro)**
- **Am inclus clasele default, clasele specifice muzicii si manipularii de fisiere.**
- Am realizat interfața cu utilizatorul folosind butoane și comenzi C++
- **Am realizat funcții custom si am folosit funcții Qt existente**
- Se compilează folosind fișierul Makefile ce include *qmake* specific Qt
- **S-a realizat un fișier shared library (sh) și un installer deb.**
- Rularea aplicației folosind fișierul *sh*, comanda: `./MusicBoard`

## Proiectarea APLICAȚIEI MusicBoard

### ➤ PROGRAMARE ORIENTATĂ OBIECT (POO)

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

**Am folosit POO pentru a oferi facilități cum sunt:**

### ➤ FACILITĂȚI

- **Cross-platform compatibility (Linux, MacOS, etc.)**
- Protejează ochii și este ușor de folosit
- Dezvoltă creativitatea
- **Utilizare fără 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 - Aplicația se distinge prin

- originalitatea UI-ului
- compatibilitatea cu diverse sisteme de operare.

### ➤ PLANUL DE DEZVOLTARE este inclus in fișierul To Do de pe Github.

- Salvarea ca fișier muzicii interpretate, Internal recording,
- includere 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 controler pentru LMMS ([www.lmms.io](http://www.lmms.io)) software gratuit pentru compoziție muzicală in Linux.**
- MusicBoard aplicație online.

# MusicBoard

Linux Launchpad/Launchkey

Am folosit funcții și tipuri de date Qt de ex. pentru:

## Pentru UI:

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

## Pentru muzică:

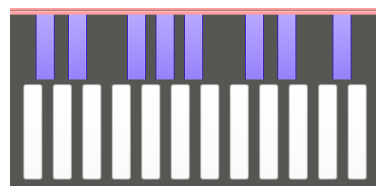
- `QMediaPlayer * C = new QMediaPlayer();` // nota Do
- `QString C_Song;` // filepath-ul fișierului muzical alocat
- `C->setMedia(QUrl(C_Song));`  
//alocare fișier muzical music player-ului
- `C->play();` // redare sunet muzical (acționare music player)
- `C_Song="qrc:/sounds/Keyboards/Keyboard2/2C.wav";`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

## FACILITĂȚI, LAYOUT TASTATURA

Astfel, s-au obținut:

- Numeroase presetări Keyboard și Launchpad
- Integrare de sunete Custom
- claviatura, ritmuri și butoane Launchpad
- organizare keyboard
- Polifonie (nelimitată- performanța computerului)

20 clape controlate de tastatură; 8 butoane de ritm (7 controlate de tastatură); 17 butoane. Launchpad controlate de tastatură



FUNCȚII Qt standard sunt utilizate și sunt implementate noi funcții C++ specific pentru MusicBoard.  
(Specifice QMediaPlayer, QAudioRecorder.)

Am realizat de ex.:

## Verificare tip de fișier, extensie pentru fișiere muzicale

```
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;
}
```

## 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

## REDARE SUNET

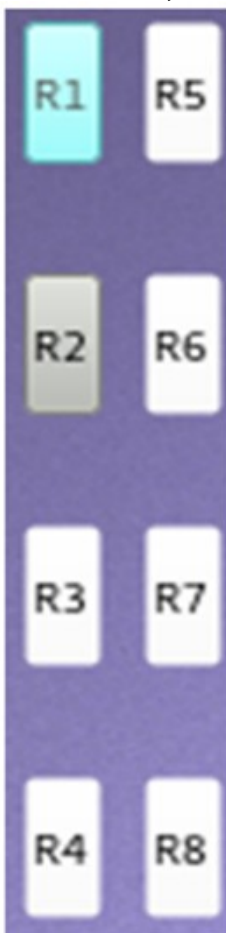
```
void MusicBoard::on_C_clicked()
{
    QMediaPlayer * C = new QMediaPlayer();
    C->setMedia(QUrl(C_Song));
    extern int KeyboardVol;
    C->setVolume(KeyboardVol);
    C->play();
}
```

# MusicBoard

Linux Launchpad/Launchkey

Alte exemple realizate:

- Funcții buton Launchpad
- Funcții pentru sunete redat simultan



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

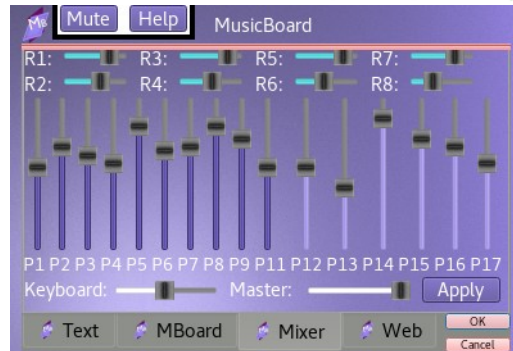




## MIXER

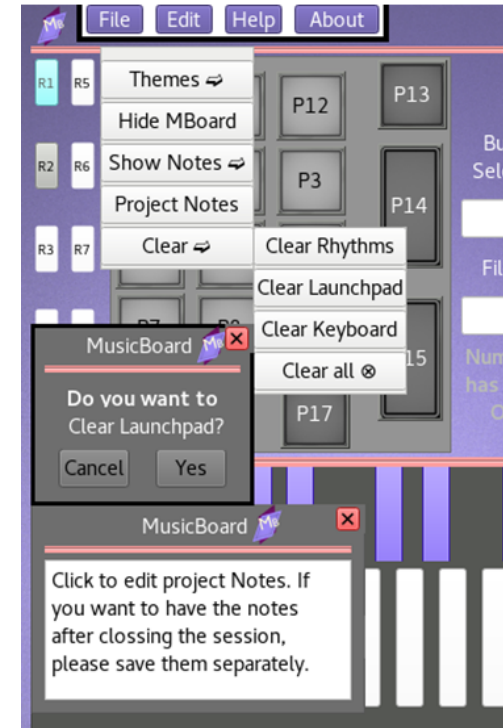
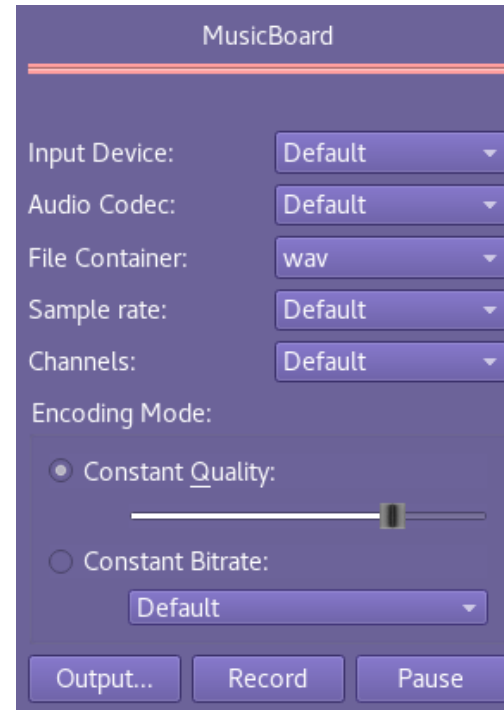
```
void Settings::on_P_1_sliderPressed()
{
    on_SettingsApplyMixerButton_clicked();
    MusicBoard a;
    a.on_P1_clicked();
}

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, 0, 0);");
        ui->NmLkOnOff->setText("OFF");
        NumLock=0;
    }
}
```



## Recorder & Menu

```
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;
```



## AUXILIARE

