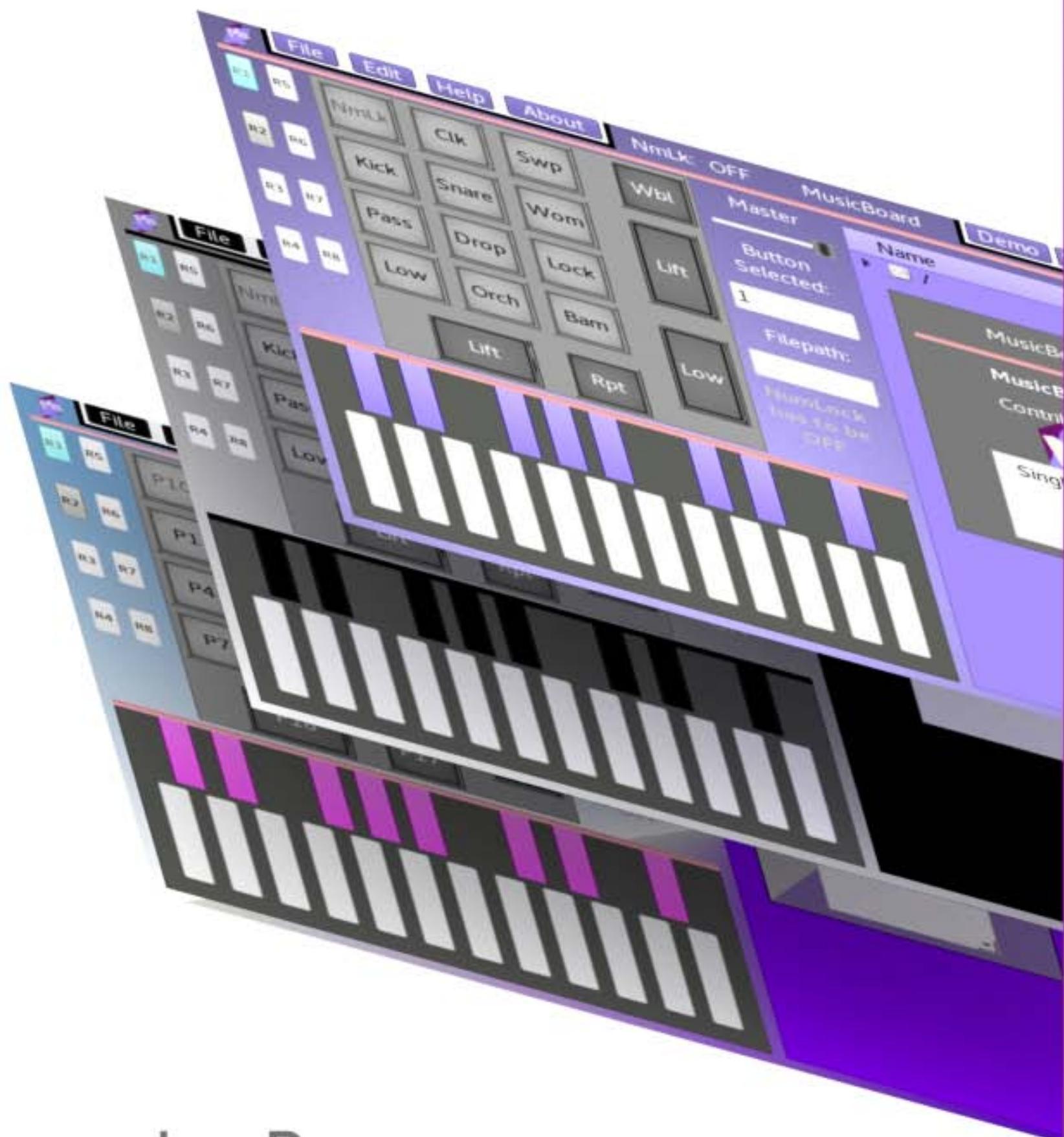


MusicBoard

- Documentație -



Alexandru P.

Cuprins

REZUMAT	3
1 Introducere	5
1.1 Qt Framework	5
1.2 Software existent similar	6
1.3 MusicBoard Launchpad / Launchkey	7
2 Facilități	7
2.1 Avantaje	7
3 Cerințe tehnice	8
4 Etape de dezvoltare și mod de utilizare	9
5 Realizarea MusicBoard 1.4	10
5.1 Librării Qt	10
5.2 Clase MusicBoard	10
5.3 Funcții Qt Utilizate	11
5.3.1. Funcții muzicale	11
5.3.2. Funcții de Browsing a fișierelor	13
5.4 NumLock	14
5.5 Ritm R1	15
5.6 Ritm R2	15
5.7 Ritmurile R3-R8	16
5.8 Demo	16
5.9 Cover	16
5.10 Funcția de selectare a culorii	16
5.11 Elemente	17
Terminalul	17
5.12 Presetări	19
5.13 Teme	20
5.14 Preferences	20
Interfața grafică UI1	20
Interfața grafică UI2	27
RGB	29

IMG.....	30
Panelul TH	31
Preferences	33
Panelul Settings, de setări	34
5.15 Taste și Clape.....	34
Particularizarea denumirilor tastelor Launchpad.....	34
Aranjarea tastelor Keyboard.....	35
5.16 Mixer.....	35
5.17 Website	37
5.18 Recorder, înregistrari	38
5.19 Imagini	40
5.20 Implementare	40
6 Planificarea dezvoltărilor și facilităților viitoare	41
Appendix.....	42
Bibliografie	54

REZUMAT

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
- Un Launchpad este un Controller MID performant, multifuncțional și particularizabil. De regulă se folosește cu un DAW pentru a compune și a interpreta,
-

AVANTAJE

- Este un proces artistic inovativ și live, în direct în fața publicului
 - Se poate utiliza în 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 compozitii muzicale.
-

ANALIZA PIETEI

- Există proiecte similare Online Launchpad, The DJ Dubstep Music Maker, HIP-HOP Drum-pads (Android)

CONFIGURAȚIE SISTEM

- Configurarea dezvoltării include AMD 6 Core, 3GHz, 4 MB RAM, two-channel sound card.
- Cerinte minime de rulare includ 1 GHz CPU, 512 MB RAM, placă de sunet.

PUBLICUL ȚINTĂ al aplicației sunt cei învață MUZICA și muzicieni.

TEHNOLOGII

- Qt5.1, C++, QtCreator (IDE), OpenGL, Audacity (muzica), Blender (grafica) în Linux 64 bit (Debian).
 - Licenta: Apache License (www.apache.org/licenses/LICENSE-2.0)
Imaginiile sunt creații originale ale autorului proiectului și sunt realizate folosind Blender gratuit (www.blender.org). Autorul are peste 3 ani experiență în aplicații grafice în Blender inclusiv pentru Concursul NASA-Space Settlement din 2017 <http://wp-space-settlement.weebly.com/>
 - Toate sunetele folosite în proiect sunt gratuite (www.musicradar.com); unele editate cu Audacity gratuit (www.audacityteam.org).
-

WEBSITE

- Site MusicBoard: <http://musicboard-launchpad.dx.am/>
 - Blog MusicBoard: <https://musicboard-blog.weebly.com/>
 - GitHub Repository: <https://github.com/MusicBoard/musicboard>
 - Website GitHub Repository: <https://github.com/MusicBoard/musicboard>
 - Blog muzical al autorului: <https://single-coil.weebly.com>
-

ETAPELE DE DEZVOLTARE

- Instalarea sistemului de operare Linux
- 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 fisierul sh, comanda: ./MusicBoard

PROIECTAREA APlicațIEI

- 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.
- Aplicatia are posibilitatea să ajungă un external virtual MIDI controler pentru LMMS (www.lmms.io) software gratuit pentru compozitie muzicală in Linux.
- MusicBoard aplicație online.

1 Introducere

1.1 Qt Framework

Qt Framework [5] este o multiplatformă ce poate fi utilizată pentru diverse proiecte cum sunt Qt Widgets application, Qt Console Application, Qt Quick Application, Qt Quick Controls Application, Qt Quick Controls 2 Application și Qt Canvas 3D Application. Qt este utilizat de multe companii mari ca Nokia, Panasonic, Samsung și HP [1]; astfel Qt este un framework C++ stabil și foarte bine documentat.

Interfața grafică, Graphical User Interface (GUI) se poate utiliza ușor folosind drag & drop cu elemente în ferestre. Proprietățile pot fi schimbate folosind panelul de proprietăți. Qt Framework a fost utilizat prima dată de autor la realizarea unui calculator [2] în C++. Qt Framework este utilizat împreună cu Qt Creator, ca IDE, interfață de dezvoltare.

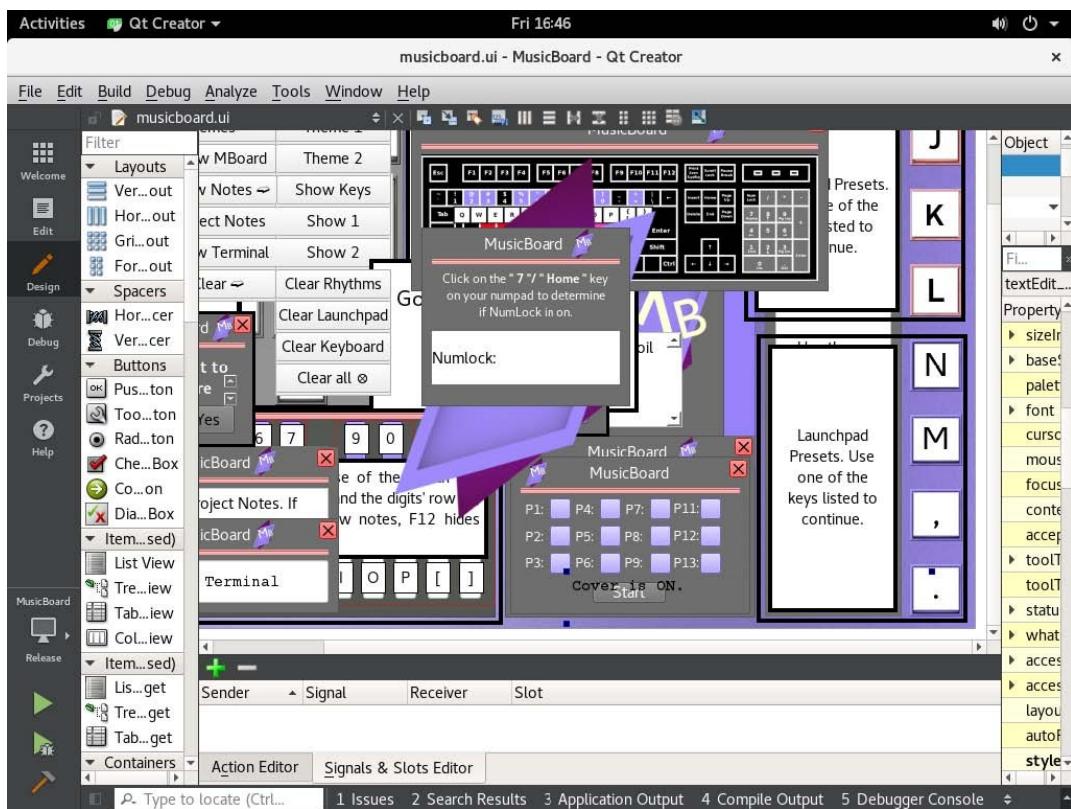


Figure 1: Qt Framework cu interfața de dezvoltare Qt Creator în timpul realizării proiectului MusicBoard.

1.2 Software existent similar

Urmează o listă a celor mai interesante soft-uri existente din domeniul muzical similare cu aplicația MusicBoard.

- Online Launchpad [52].
- Versiune online de Launchpad oficial [53].
- Tableță compatibilă Virtual Launchpad singura cu sunete particularizabile [54].
- Telefon compatibil Virtual Launchpad [31].

Un Launchpad [14] este un Controller MID performant, multifuncțional și particularizabil.

De regulă se folosește cu un DAW pentru a compune și a interpreta.

După cum se observă Launchpad-ul are câteva aplicații din categoria sa. Multe dintre ele însă nu pot fi particularizate și nu au tastatura utilă pentru Launchkey. MusicBoard este primul Launchpad / Launchkey Virtual pentru Linux [11] (Figura 2). Este scris în C++ pe multi-platforma Qt Framework [17] pentru a fi mai departe utilizat pe orice sistem de operare.



Figure 2: MusicBoard, primul Launchpad / Launchkey Virtual Linux.

1.3 MusicBoard Launchpad / Launchkey

MusicBoard este o platformă gratuită open-source și aplicație cross-platform ce poate fi utilizată pentru concerte live, în direct, de muzică folosind calculatorul. Software-ul a fost testat pe Linux [11] dar poate fi testat și pe alte sisteme de operare. Cu număr similar de butoane cu ale unui Launchpad, MusicBoard include facilități de Launchpad și Launchkey pentru a crea un mediu optim compoziției și concertelor muzicale.

2 Facilități

Legat de complexitatea sunetului Launchpad are trei module cărora le sunt atribuite sunete de către utilizator. Cu un total de 23 butoane cu sunete particularizate și 20 taste pentru 3 keyboards preseitate sau sunete custom [34], MusicBoard oferă un Launchpad complet particularizabil și un keyboard decent pentru compoziție [12] și concerte muzicale.

Un Launchpad [14] protejează ochii, vederea muzicianului, îl ajută să interacționeze cu computerul, scurtează timpul necesar pentru a realiza compoziția și oferă un concert muzical inovativ și performant. MusicBoard are preinstalate 29 de sunete Launchpad gratuite [9] și 3 keyboard preseitate [8] (ambele editabile cu Audacity [7]) și cu polifonii [25] nelimitate.

Nu necesită conexiune MIDI cu sistemul intern de sunet al computerului utilizatorului pentru a lucra cu diverse tipuri de fișiere de sunet compatibile: wav, flac, ogg, aac, mp3, aiff, wma, m4p, 3gp și vox.

2.1 Avantaje

MusicBoard oferă următoarele avantaje:

- O configurație deosebită pentru compoziție.
- Un instrument ce poate fi particularizat de utilizator.
- O interfață User Interface (UI) particularizabilă.
- Interacțiune Om-Computer deosebită.
- Vederea compozitorului nu este afectată.
- Înregistrează concertele în direct, live.
- UI este folosit cu tastatura.

- Posibilitatea de a compune muzică indiferent de locație folosind doar tastatura.
- Posibilitatea de a actiona sunete multiple folosind doar un buton.
- Are 23–43 butoane Launchpad și 20 taste Keyboard.
- Are mixer.
- Notele au ambele denumiri utilizate C, D, E, F, etc. și Do, Re, Mi, Fa, etc.
- Are terminal pentru dezvoltatori.
- Codul susră este gratuit pentru dezvoltatori.
- Are opțiunea NumLock [30] pentru validare.
- Are un Cover ce protejează configurația.
- Are opțiunea să salveze note, Notes.
- Imaginele sunt particularizabile.
- Denumiri particularizabile pentru butoanele Launchpad.
- Un Blog pentru sugestii.
- Setări pentru calculatoare cu performanță slabă.
- Are selector de culori, RGB.
- Culorile, temele sunt particularizabile pentru: Frames și Butoane.

3 Cerințe tehnice

MusicBoard are următoarele cerințe minimale pentru computerul [22] utilizatorului:

- GNU/Linux
- Intel Celeron (sau similar) / 1 GHz CPU
- 512 MB RAM
- Two-Channel Sound Card
- NVidia GeForce 820M
- Keyboard
- Monitor
- Boxe

Faza de dezvoltare necesită calculatoare de înaltă performanță; de exemplu aplicația a fost inițial dezvoltată pe un calculator AMD hexacore 3GHz CPU.

4 Etape de dezvoltare și mod de utilizare

ETAPELE DE DEZVOLTARE

- Instalarea sistemului de operare Linux
- Instalare Qt Framework C++ [44,45]
- Crearea proiectului MusicBoard, fișierul project 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 fisierul sh, comanda: ./MusicBoard

MOD DE UTILIZARE

În Musicboard tastatura se utilizează în patru sectoare:

- Launchpad
- Keyboard
- Ritmuri
- Presetari.

Pentru a accesa sunete preseitate ce se realizează după ce aplicația pornește, se folosesc tastele "k", "j" și "h" pentru a realiza presetările Launchkey [15] și "n", "m", "," și ":" pentru Launchpad [14].

Pentru o cântă o melodie folosind tastura se folosește "q", "w", "e", "r", "t", "y", "u", "i", "o", "p", "[" și "]" pentru "clapele" albe și "2", "3", "5", "6", "7", "9", "0" și "=" pentru "clapele" negre.

Pentru a încărca sunete particularizate de pe butoane, click pe butonul căruia îi schimbăm sunetul, cu browse-rul căutăm sunetul, și cu click îl setăm. Pentru a cântă sunetul se folosește shortcut-ul [51] tastaturii sau click cu mouse-ul.

Dacă se dorește încărcarea unui sunet lung, de exemplu un ritm, se folosește butonul "R1" folosit special pentru acest lucru. Acesta ajută utilizatorul să izoleze un sunet lung, astfel încât să nu se suprapună sau să determine un sunet final distorsionat. Acest buton se folosește imediat după încărcarea presetărilor, și poate fi utilizat doar cu sunete particulare din fișiere.

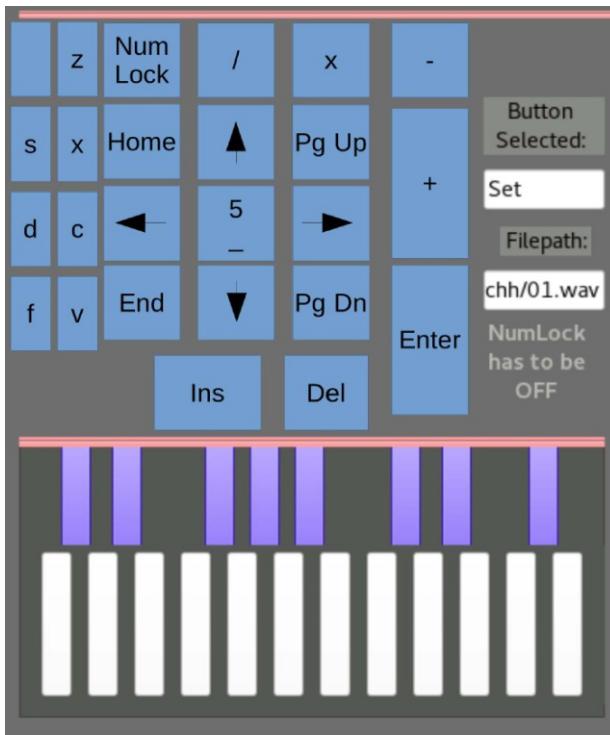


Figure 3: MusicBoard Layout.

5 Realizarea MusicBoard 1.4

5.1 Librării Qt.

MusicBoard a început [18] ca un "Qt Widgets Application", care este proiect implicit în Qt Creator. Librăriile utilizate sunt QtCore, QtGui, QDialog [38], QMainWindow, QString, incluse implicit și QMediaPlayer, QDir QFileSystemModel [35,36] ce sunt incluse specific pentru MusicBoard. QMediaPlayer este librăria ce permite toate acțiunile cu sunete din MusicBoard, QDir ajută QFileSystemModel pentru realizarea browser-ului [21,40] de fișiere folosit de MusicBoard. Toate sunt implementate în Qt 5.

5.2 Clase MusicBoard

MusicBoard are 4 clase:

- musicboard.h (principală)
- preferencesform.h
- settings.h
- audiorecorder.h

Fiecare clasă are fișierul .cpp și .ui corespondent. Există un fișier de resurse [49,50], res.qrc, cu toate adresele fișierelor din proiect. Fișierul **MusicBoard.pro** include fișierele menționate și fișierul main.cpp ce lansează aplicația în IDE.

5.3 Funcții Qt Utilizate

Cele două categorii de funcții [17] principale din MusicBoard sunt funcții muzicale sau de sunet și funcții de browsing.

5.3.1. Funcții muzicale

În prima categorie sunt funcții de exemplu ca `play()`; și `setMedia()`; . Urmează un exemplu de implementare:

Cod:

```
1. void MusicBoard::on_P1_clicked()
2. {
3.     QMediaPlayer * P1 = new QMediaPlayer();
4.     P1->setMedia(QUrl(C_Song));
5.     P1->play();
6. }
```

Această secvență este `on_P1_clicked()`; funcție ce este acționată cu click pe butonul P1.

În linia 4, QMediaPlayer P1 primește QUrl, calea, către sunetul specific. Variabila de sunet P1_Song este de tip QString și a fost declarată global [41]. Variabila P1_Song este atribuită sau la o cale, Filepath [37] particulară sau la una implicit.

Cod:

1. Filepath Custom pentru Butoane Launchpad.

```
1. void MusicBoard::on_File2_clicked(const QModelIndex &index)
2. {
3.     QString FilePath = filemodel->fileInfo(index).absoluteFilePath();
4.     FilePath = "file://" + FilePath;
5.     if (CheckWAV(FilePath)==1 || CheckMP3(FilePath)==1
|| CheckOgg(FilePath)==1 || CheckM4p(FilePath)==1
|| CheckWma(FilePath)==1 || CheckAiff(FilePath)==1
|| CheckVox(FilePath)==1 || CheckFlac(FilePath)==1
|| Check3gp(FilePath)==1 || CheckAac(FilePath)==1) {
6.         if (setButton==1){P1_Song = FilePath; on_P1_clicked();}
7.         if (setButton==2){P2_Song = FilePath; on_P2_clicked();}
8.
9. //and so on until the last button
10.
11.        if (setButton==26){R8_Song = FilePath; on_R8_clicked();}
12.        setButton=0;
13.        ui->Error->setText("Set");
14.    } else {
15.        ui->Error->setText("Error");
16.    }
17. }
```

2. Filepath implicit pentru clape

```

1. void MusicBoard::on_K4_clicked()
2. {
3.     C_Song="qrc:/sounds/Keyboards/Keyboard4/4_1_C.wav";
4.     C_B_Song="qrc:/sounds/Keyboards/Keyboard4/4_2_CB.wav";
5.
6.     // \c si so on for all keys
7.
8.     G2_Song="qrc:/sounds/Keyboards/Keyboard4/4_20_G.wav";
9.
10. }
```

2.1 Filepath implicit pentru butoane Launchpad și clape

```

1. void MusicBoard::on_Set4_clicked()
2. {
3.     P1_Song="qrc:/sounds/Dub_Selection/TMS3_NuKick_165.wav";
4.     [and so on]
5.     R7_Song="qrc:/sounds/Dub_Selection/Low_End_FX/LowEnd_FX_14.wav";
6.
7.
8.     extern int isKeyboardAsLaunchpadShow;
9.     if (isKeyboardAsLaunchpadShow==1) {
10.         C_Song="qrc:/sounds/Dub_Selection/Low_End_FX/LowEnd_FX_16.wav";
11.         [and so on]
12.         F2_B_Song="qrc:/sounds/Dub_Selection/TMS3_NuKick_165.wav";
13.     }
14.     [unrelated code]
15. }
```

Funcția `on_K4_clicked()`; trimite către al 4-lea preset pentru clape. Această presetare este apelată și pentru al 3-lea preset (`on_K3_clicked()`); datorită limitărilor de resurse date de C++ compiler. Deci, avem doar 3 presetări pentru clape și doar 3 butoane le corespund. În fișierul de resurse [49,50] al proiectului Qt, sunetele menite pentru al 3-lea preset sunt comentate [4], și la fel și în fișierul `musicboard.cpp`. Îmbunătățiri vor fi aduse în versiunile viitoare. Deci, pentru al 3-lea buton avem:

Cod:

```

1. void MusicBoard::on_K3_clicked()
2. {
3.     on_K4_clicked();
4. }
```

Un alt tip de funcție folosită de MusicBoard pentru fișierele de sunet este funcția de verificare a formatului [3]. Toate aceste funcții sunt apelate în linia 5 din codul 1.Filepath particularizat pentru butoane Launchpad. O astfel de funcție este:

Cod:

```

1. int CheckWAV (QString a) {
2.     QString v;
3.     int i;
4.     int n;
5.     n=a.length();
6.     for (i=0;i<=n;i=i+1) {
```

```
7. v[n-i]=a[i];
8. }
9. if (v[1]=='v' && v[2]=='a' && v[3]=='w' && v[4]=='.') {
10. return 1;
11. }
12. else {
13. return 0;
14. }
15. }
```

5.3.2. Funcții de Browsing a fișierelor

Pentru browsing [21, 36] prin fișiere MusicBoard folosește două liste, una pentru foldere și celaltă pentru fișiere propriu-zise. Ele sunt declarate după cum urmează:

Cod:

```
1. dirmodel = new QFileSystemModel(this);
2. ui->File1->setModel(dirmodel);
3. filemodel = new QFileSystemModel(this);
4. ui->File2->setModel(filemodel);
5. ui->File1->hideColumn(1);
7. ui->File1->hideColumn(2);
8. ui->File1->hideColumn(3);
9. QString sPath = "file:///home/n";
```

MusicBoard are două QFileSystemModels [36], dirmodel filemodel, ce sunt correlate cu File1 și File2 în UI.

Ambele sunt setate cu calea implicită către variabila QString, sPath din software, unde sunt restrânse să arate doar fișiere sau foldere.

Coloanele sunt ascunse cu funcția hideColumn () ; pentru a nu arăta date nerelevante.

Pentru browsing prin fișiere este folosită o variabilă QString ca în linia 3 și 4 din 1. Filepath [37] particularizat pentru butoane Launchpad:

Cod:

```
3. QString FilePath = filemodel->fileInfo(index).absoluteFilePath();
4. FilePath = "file://" + FilePath;
```

În linia 3 este filepath luat din filemodel și în linia 4 este completat cu numele fișierului necesar căii filepath corecte. Transmite compilatorului faptul că calea aparține unui fișier și nu unui folder.

5.4 NumLock

Pentru a obține starea tastei NumLock, [30] MusicBoard comunică cu sistemul de operare printr-un modul particular NumLock. Acesta este vizibil [43] la lansarea aplicației, când cere utilizatorului să testeze tasta 7/Home pentru a asigura că NumLock este închis, off. Aceasta este relevant pentru MusicBoard deoarece fără NumLock off, se amestecă shortcuts [51] Keyboard și Launchpad. Ca să fim siguri că utilizatorul nu folosește mouse-ul pentru acționarea butoanelor prin click, când starea NumLock este determinată este plasat peste MusicBoard un Cover de siguranță.

Pe tasta Numlock (P10 din Launchpad) cover-ul asigură că variațiile [47] NumLock din cod sunt date doar de tastatura curentă. Cover-ul este transparent și are eticheta de vizibilitate pentru utilizatori și dezvoltatori.

Cod:

1. Initializare:

```
1. extern int NumLock; // declared in main.cpp, initialized with 10
2. extern int NumLockShow; // declared in main.cpp, initialized with 1, used
   to know the dialog's visibility
3. ui->NumlockLabel->setReadOnly(true);
4. ui->NumlockLabel 2->hide();
```

2.0 Dacă NumLock este On, o clapă de pian va fi acționată în locul tastei Launchpad în NumPad:

```
1. void MusicBoard::on_A_B_clicked()
2. {
3. [unrelated code]
4. ui->NumLockLabel->setText("Click on \"NumLock\"");
5. extern int NumLock;
6. if (NumLock==10){
7. NumLock=7;
8. }
9. [unrelated code]
10. }
11.
```

2.1 Dacă NumLock este off, tasta Launchpad va acționa și NumLock Dialog [38] va fi închis:

```
1. void MusicBoard::on_P1_clicked()
2. {
3. [unrelated code]
4. extern int NumLock;
5. if (NumLock==10){
6. ui->NumLockInfo->hide();
7. ui->Cover->hide();
8. extern int isCover;
9. isCover=0;
10. ui->LButton->setEnabled(true);
11. ui->LButton->setStyleSheet("background-color: rgb(186, 189, 182);");
12. ui->NmLkOnOff->setStyleSheet("color: rgb(255, 255, 255);
   background-color: rgba(136, 138, 133, 0');");
13. ui->NmLkOnOff->setText("OFF");
14. NumLock=0;
```

```
15. }
16. [unrelated code]
17. }
```

3.0 După ce se cunoaște starea NumLock se asigură alertarea utilizatorului dacă acționeaza Numlock:

```
1. void MusicBoard::on_P10_clicked()
2. {
3. int k=0;
4. extern int NumLock;
5. if (NumLock==0) {
6. NumLock=1;
7. ui->NmLkOnOff->setStyleSheet("background-color: rgba(255, 255, 255, 0);
color: rgb(239, 41, 41);");
8. ui->NmLkOnOff->setText("ON");
9. ui->UnderOn->show();
10. k=1;
11. }
12. if (NumLock==1 && k==0) {
13. ui->NmLkOnOff->setStyleSheet("color: rgb(255, 255, 255);
background-color: rgba(136, 138, 133, 0);");
14. ui->NmLkOnOff->setText("OFF");
15. ui->UnderOn->hide();
16. NumLock=0;
17. }
18. if (NumLock==7) {
19. ui->NumLockInfo->hide();
20. ui->Cover->hide();
21. extern int isCover;
22. isCover=0;
23. ui->LButton->setStyleSheet("background-color: rgb(186, 189, 182);");
24. ui->LButton->setEnabled(true);
25. ui->NmLkOnOff->setText("OFF");
26. ui->NmLkOnOff->setStyleSheet("color: rgb(255, 255, 255);
background-color: rgba(136, 138, 133, 0);");
27. NumLock=0;
28. k=1;
29. }
30. k=1;
31. }
```

Deci, variabila NumLock este inițial 10, după aceea poate fi 7 dacă NumLock este on, sau 0 dacă este off. După apăsarea tastei 7, valorile pot fi doar 1 sau 0. După verificare, Cover-ul de siguranță este retras. UI arată utilizatorului starea NumLock.

5.5 Ritm R1

Butonul R1 este utilizat pentru sunete foarte lungi, deci nu are un shortcut, altfel sunetele se suprapun.

5.6 Ritm R2

Butonul Ritm R2 se folosește cu tasta "s" key, pentru mai multe acțiuni ale butoanelor Launchpad, butoane de la P1 la P13. Sunetele corespondente butonului selectat din grid va fi atribuit lui R2, astfel că toate sunetele sunt executate în același timp.

5.7 Ritmurile R3-R8

Ritmurile sunt în general folosite pentru sunete lungi din fișiere pe care nu trebuie să le activezi prin click tot timpul. Au tastele plasate în zona ce nu include alte sunete astfel este acces doar când se vrea explicit acest lucru.

5.8 Demo

MusicBoard include o demonstrație a aplicației, un Demo ce ilustrează muzicienilor (sau celor ce învață muzică) cum să seteze și să utilizeze MusicBoard.

Cod:

```
1.0 Inițializare Demo
1. ui->Demo1->hide();
2. ui->Demo2->hide();
3. ui->Demo3->hide();
4. ui->Demo4->hide();
5. ui->Demo5->hide();
6. ui->Demo6->hide();
7. ui->Demo7->hide();
2.0 Un Keyboard presetat acționat prin Click
1. if (isDemo==1) {
2. on_HiddenHideAllButton_clicked();
3. ui->Demo1->hide();
4. ui->Demo2->show();
5. isDemo=2;
6. }
```

5.9 Cover

Secvența Cover este utilizată de procesul NumLock și este activată de butonul LButton.

Cod:

```
1. void MusicBoard::on_LButton_clicked()
2. { extern int isCoverShow;
3. if(isCoverShow==1){
4. if (isCover==1){
5. ui->Cover->hide();
6. isCover=0;
7. ui->LButton->setStyleSheet("background-color: rgb(186, 189, 182);");
8. }
9. else {
10. ui->Cover->show();
11. isCover=1;
12. ui->LButton->setStyleSheet("color: rgb(239, 41, 41); background-color:
    rgb(255, 255, 255);");
13. }
14. }
15. }
```

5.10 Funcția de selectare a culorii

O culoare este un sir de caractere, un RGB String (QString); se poate seta și un contrast de culoare (B sau W). Acesta este utilizat pentru configurațiile de culori pentru toate butoanele.

Cod:

```
1. if (CFile.length()<17) {  
2.     ui->File1->setStyleSheet("background-color: "+CFile+"; color:rgb(255, 255,  
255);");  
3.     ui->File2->setStyleSheet("background-color: "+CFile+"; color:    rgb(255,  
255, 255);");  
4. }  
5. else {  
6.     ui->File1->setStyleSheet("background-color: "+CFile+"; color:rgb(0, 0,  
0));;  
7.     ui->File2->setStyleSheet("background-color: "+CFile+"; color:rgb(0, 0,  
0);;");  
8. }
```

Funcția Bright determină Strălucirea culorii și utilizează B (black, negru) sau W (white, alb). Se compară lungimea sirului de caractere RGB QString cu Num. 17. Funcția a fost testată pentru 7 seturi de 64 butoane cu prima culoare setată aleator cu funcția random, rand [28].

5.11 Elemente

Următoarele elemente sunt incluse în MusicBoard:

- Help;
- About;
- Terminal;
- Project Notes;
- MBoard (Launchpad Layout computer keyboard);
- R2 Button Grid;
- Licență [29];
- Certitudine de ștergere sau ieșire din ferestre: "Are You Sure" ("to Clear" și "to Exit").

Acestea sunt controlate de variabile din fișierul musicboard.cpp. Procedura este la fel, 1 sau 0 pentru vizibilitate [43] sau ascundere. Butoanele variabilelor nu sunt todeauna accesibile prin shortcut [51]. Hidden Buttons, butoanele ascunse implementate se fac accesibile prin shortcut.

Terminalul

Terminalul [23] este utilizat în dezvoltarea aplicației pentru a vizualiza diversi parametri sau comenzi ale utilizatorului. Terminalul are explicații în fișierul de Help.txt din panelul Edit.

Cod:

```
1. void MusicBoard::on_MBTerminalTextActual_editingFinished()
2. {
3.     extern QString P1_R;
4.     [and so on]
5.     extern QString Cfile_Custom;
6.
7.     extern int P1_A;
8.     [and so on]
9.     extern int Notes;
10.
11.    QString MBTerminalText;
12.    MBTerminalText=ui->MBTerminalTextActual->text();
13.    ui->MBTerminalTextActual->setText("");
14.    if (MBTerminalText=="Mb Dev") {
15.        ui->MBTerminalTextActual->setText("Tr, Val");
16.    }
17.    if (MBTerminalText=="Mb Dev Tr") {
18.        extern int Bg;
19.        Bg=0;
20.        ui->MBoard->setStyleSheet("background-color: rgba(0, 0, 0,
0');");
21.        [and so on]
22.        ui->L11->setStyleSheet("background-color: rgba(0, 0, 0,
0');");
23.
24.    }
25.    if (MBTerminalText=="Mb Dev Val") {
26.
27.        ui->MBTerminalTextActual->setText("Enter Var");
28.
29.        if (MBTerminalText=="P1_R") {
30.            ui->MBTerminalTextActual->setText(P1_R);
31.        }
32.        [and so on]
33.        if (MBTerminalText=="P7_A") {
34.            ui->MBTerminalTextActual->setText(QString::number(P7_A));
35.        }
36.    }
37.    (MBTerminalText=="Mb Dev Val P1_A") {
38.        ui->MBTerminalTextActual->setText(QString::number(P1_A));
39.    }
40.    [and so on]
41.    if (MBTerminalText=="Mb Dev Val CFile_Custom") {
42.        ui->MBTerminalTextActual->setText(Cfile);
43.    }
44.    [unrelated code]
45. }
```

5.12 Presetări

Sunt disponibile o varietate de presetări în MusicBoard:

- 29 WAV, fișiere de sunete pentru Launchpad;
- 3 x 20 WAV, fișiere de sunete pentru Keyboard.

ACEstea se pot aplica celor 7 butoane din dreapta, 3 pentru presetările de keyboard și 4 pentru keyboard și Launchpad. Presetările sunt incluse în fișierul res.qrc.

Cod:

```
1. <RCC>
2. <qresource prefix="/sounds">
3. <file>Dub_Selection/006.wav</file>
4. [and so on]
5. <file>Keyboards/Keyboard4/4_19_FB.wav</file>
6. </qresource>
7. <qresource prefix="/pictures">
8. <file>MBoard.jpg</file>
9. [and so on]
10. <file>Theme24.png</file>
11. </qresource>
12. </RCC>
```

Presetările sunt atribuite tastelor după cum urmează:

Cod:

1.0 Launchpad

```
1. void MusicBoard::on_Set1_clicked()
2. {
3. [unrelated code]
4. P1_Song="qrc:/sounds/Dub_Selection/TMS3_NuKick_165.wav";
5. [and so on]
6. R7_Song="qrc:/sounds/Dub_Selection/Low_End_FX/LowEnd_FX_017.wav";
7. [unrelated code]
8. }
```

1.1 Animatie Launchpad

```
1. void MusicBoard::on_Set1_pressed()
2. {
3. extern int isCPUEnabled;
4. if (isCPUEnabled==1){
5. ui->Set1->setStyleSheet("color: rgb(255, 255, 255);background-image:
url(:/pictures/Sets1.png);image: url(:/pictures/Sets1.png);background-color:
rgba(159, 138, 252, 153);border-color: rgba(159, 138, 252, 153);");
6. }
7. }
8. void MusicBoard::on_Set1_released()
9. {
10. ui->Set1->setStyleSheet("color: rgb(255, 255, 255);background-image:
url(:/pictures/Sets.png);image: url(:/pictures/Sets.png);background-color:
rgba(159, 138, 252, 153);border-color: rgba(159, 138, 252, 153);");
11. }
```

2.0 Keyboard

```
1. void MusicBoard::on_K1_clicked()
2. {
3. [unrelated code]
4. C_Song="qrc:/sounds/PianoSounds/Edit/C.wav";
5. [and so on]
6. F2_Song="qrc:/sounds/PianoSounds/Edit/F2.wav";
7. [unrelated code]
8. }
```

2.1 Animatie Keyboard

```
1. void MusicBoard::on_K1_released()
2. {
3. ui->K1->setStyleSheet(K_Released);
4. }
5. void MusicBoard::on_K1_pressed()
6. {
7. extern int isCPUEnabled;
8. if (isCPUEnabled==1){
9. ui->K1->setStyleSheet(K_Pre);
10. }
11. }
```

5.13 Teme

Temele sunt un set de culori atașate tuturor elementelor. Imaginele sunt setate folosind cod CSS [4,19] și proprietatea ” pixmap” etichetei unui element, când imaginea [42] nu este o resursă a proiectului. Acesta este folosit pentru background-ul aplicației principale și pentru setări.

Launchpad și presetările sunt realizate cu CSS [4,19], la fel și Slider Headers. Temele pot fi accesate doar din panelul Edit. Mai multe particularizări se găsesc în Preferences.

5.14 Preferences

Preferences are panel pentru:

- UI [26] (două interfețe grafice UI1 și UI2)
- RGB (selector de culori)
- IMG (selector de imagini)

Interfața grafică UI1

- Font Selector
- Color 1 Selector (pentru File Browser și elemente MusicBoard)
- Color 2 Selector (pentru separatori)
- Text Color Selector
- Keyboard Color Selector
- Buttons Color Selector

Toate folosesc variabila globală `QString` ce folosesc culori RGB [20] pentru butonul `ReInit` din frame-ul principal MusicBoard. Când este acționat butonul `ReInit` atribuie șirul de RGB elementelor.



Figure 4: Panelul de interfață utilizator predefinit UI1 din Preferences.

Cod:
1.0 Obținerea de culoare particularizată.

```
1. C12=ui5->CustomColor_5->text(); // if color is custom
2. [or]
3. C12="rgb(255, 159, 159)"; // if color is preset
```

1.1 Setarea stilului etichetei culorii, Label style.

```
1. ui5->ColorsLabel_5->setStyleSheet("background-color: "+C12+");"
2.0 ReInit în musicboard.cpp .
1. void MusicBoard::on_ReInit_clicked()
2. {
3.     extern int NumLock;
4.     [and so on]
5.     extern int Notes;
6.
7.     if (Notes==0) {
8.         C_Text="C";
9.         [and so on]
10.        F2_B_Text="F2#";
11.
12.    }
13.    else {
14.        C_Text="Do";
15.        [and so on]
16.        F2_B_Text="Fa2#";
17.    }
18.
19.
20.
21. ui->Keys->setStyleSheet("background-color: rgb(85, 87, 83);");
```

```
22. ui->Launchpad->setStyleSheet("background-color: rgb(148, 148, 148);");
23.
24.
25. ui->MBoard->setFont(Font);
26. ui->Help->setFont(Font);
27. ui->About->setFont(Font);
28. ui->Licence->setFont(Font);
29. ui->MBTerminal->setFont(Font);
30. ui->NumLockInfo->setFont(Font);
31. ui->ProjectNotes->setFont(Font);
32. ui->Rhythm2->setFont(Font);
33. ui->MBoard->setStyleSheet("background-color: "+C1+");"
34. ui->Clear->setStyleSheet("background-color: "+C1+");"
35. ui->Help->setStyleSheet("background-color: "+C1+");"
36. ui->About->setStyleSheet("background-color: "+C1+");"
37. ui->Licence->setStyleSheet("background-color: "+C1+");"
38. ui->MBTerminal->setStyleSheet("background-color: "+C1+");"
39. ui->NumLockInfo->setStyleSheet("background-color: "+C1+");"
40. ui->ProjectNotes->setStyleSheet("background-color: "+C1+");"
41. ui->Rhythm2->setStyleSheet("background-color: "+C1+");"
42.
43. if (isNumLockShow==1) {
44. ui->UnderOn->show();
45. ui->label_42->show();
46. ui->NmLkOnOff->show();
47. } else {
48. ui->UnderOn->hide();
49. ui->label_42->hide();
50. ui->NmLkOnOff->hide();
51. }
52.
53. if (isImages==0) {
54. ui->MB->setStyleSheet(" background-color: rgba(255, 255, 255, 0);");
55. [and so on]
56. ui->MB_11->setStyleSheet(" background-color: rgba(255, 255, 255, 0);");
57. }
58. else {
59. ui->MB->setStyleSheet("background-image: url(:/pictures/MB.png);
background-color: rgba(255, 255, 255, 0); image: url(:/pictures/MB.png);");
60. ui->MB_2->setStyleSheet("background-image: url(:/pictures/MB.png);
background-color: rgba(255, 255, 255, 0);");
61. [and so on]
62. ui->MB_11->setStyleSheet("background-image: url(:/pictures/MB.png);
background-color: rgba(255, 255, 255, 0); image: url(:/pictures/MB.png);");
63. }
64.
65. ui->label_2->setStyleSheet("color: "+CFont+"; background-color:
rgba(0, 0, 0, 0)");
66. [and so on]
67. ui->label_43->setStyleSheet("color: "+CFont+"; background-color:
rgba(0, 0, 0, 0)");
68.
69.
70. if (CFile.length()<17) {
71. ui->File1->setStyleSheet("background-color: "+CFile+";
color: rgb(255, 255, 255);");
72. ui->File2->setStyleSheet("background-color: "+CFile+";
color: rgb(255, 255, 255);");
73.
```

```
74. }
75. else {
76. ui->File1->setStyleSheet("background-color: "+CFile+";");
77. ui->File2->setStyleSheet("background-color: "+CFile+";");
78. }
79.
80. ui->L1->setStyleSheet("background-color: "+C12+";");
81. [and so on]
82. ui->L11->setStyleSheet("background-color: "+C12+";");
83.
84.
85. if (CKeyboard.length()<17) {
86. ui->C_B->setStyleSheet("background-color: "+CKeyboard+"; font: 7.4pt
\ "Cantarell\"; color: rgb(255, 255, 255);");
87. [and so on]
88. ui->F2_B->setStyleSheet("background-color: "+CKeyboard+"; font: 7.4pt
\ "Cantarell\"; color: rgb(255, 255, 255);");
89. }
90. else {
91. ui->C_B->setStyleSheet("background-color: "+CKeyboard+"; font: 7.4pt
\ "Cantarell\";");
92. [and so on]
93. ui->F2_B->setStyleSheet("background-color: "+CKeyboard+"; font: 7.4pt
\ "Cantarell\";");
94. }
95.
96. if (CButtons.length()<17) {
97. ui->FileButton->setStyleSheet("background-color: "+CButtons+"; color:
rgb(255, 255, 255);");
98. ui->EditButton->setStyleSheet("background-color: "+CButtons+"; color:
rgb(255, 255, 255);");
99. ui->AboutButton->setStyleSheet("background-color: "+CButtons+"; color:
rgb(255, 255, 255);");
100. ui->HelpButton->setStyleSheet("background-color: "+CButtons+"; color:
rgb(255, 255, 255);");
101. ui->PreferencesButton->setStyleSheet("background-color: "+CButtons+";
color: rgb(255, 255, 255);");
102. ui->DemoButton->setStyleSheet("background-color: "+CButtons+"; color:
rgb(255, 255, 255);");
103. ui->RecButton->setStyleSheet("background-color: "+CButtons+"; color:
rgb(255, 255, 255);");
104. } else {
105. ui->FileButton->setStyleSheet("background-color: "+CButtons+"; color:
rgb(0, 0, 0);");
106. ui->EditButton->setStyleSheet("background-color: "+CButtons+"; color:
rgb(0, 0, 0);");
107. ui->AboutButton->setStyleSheet("background-color: "+CButtons+"; color:
rgb(0, 0, 0);");
108. ui->HelpButton->setStyleSheet("background-color: "+CButtons+"; color:
rgb(0, 0, 0);");
109. ui->PreferencesButton->setStyleSheet("background-color: "+CButtons+";
color: rgb(0, 0, 0);");
110. ui->DemoButton->setStyleSheet("background-color: "+CButtons+"; color:
rgb(0, 0, 0);");
111. ui->RecButton->setStyleSheet("background-color: "+CButtons+"; color:
rgb(0, 0, 0);");
112.
113. if (Bg==0){ui->Picture->setStyleSheet("background-image:
url(:/pictures/Picture1_4.png); image: url(:/pictures/Picture1_4.png);
```

```

background-color: rgba(0, 0, 0, 0)");
114. [and so on]
115. if (Bg==10){ui->Picture->setStyleSheet("background-color:
qconicalgradient(cx:0.5, cy:0.5, angle:224.8, stop:0
rgba(211, 215, 207, 255), stop:0.189055 rgba(105, 0, 219, 255),
stop:0.40796 rgba(0, 0, 131, 255), stop:0.606965
rgba(175, 0, 194, 255), stop:0.741294 rgba(97, 146, 189, 255), stop:1
rgba(255, 255, 255, 255))");}
116. if (NumLock==1){
117. ui->NmLkOnOff->setStyleSheet("background-color: rgba(255, 255, 255, 0);
color: rgb(239, 41, 41);");
118. ui->NmLkOnOff->setText("ON");
119. ui->UnderOn->show();
120. }
121. else {
122. ui->NumlockLabel->setStyleSheet("background-color: rgba(255, 255, 255,
0);");
123. ui->NmLkOnOff->setText("OFF");
124. ui->UnderOn->hide();
125. }
126.
127. extern QString P1_R;
128. ui->P1->setText(P1_R);
129. ui->P1->setShortcut(Qt::Key_Home);
130. [and so on]
131. extern QString P17_R;
132. ui->P17->setText(P17_R);
133. ui->P17->setShortcut(Qt::Key_Delete);
134.
135. ui->C->setShortcut(Qt::Key_Q);
136. [and so on]
137. ui->F2_B->setShortcut(Qt::Key_Equal);
139. ui->R2->setShortcut(Qt::Key_S);
140. [and so on]
141. ui->R7->setShortcut(Qt::Key_C);
142.
143.
144. extern int P1_A;
145. [and so on]
146. extern int R7_A;
147.
148. P1->setVolume(P1_A);
149. [and so on]
150. P17->setVolume(P17_A);
151. R1->setVolume(R1_A);
152. [and so on]
153. R7->setVolume(R7_A);
154. }

```

3.1 Obținere Font

```

1. void PreferencesForm::on_FontsCombo_2_currentFontChanged(const QFont &f)
2. {
3. [unrelated code]
4. extern QFont Font;
5. [unrelated code]
6. ui4->FontLabel_2->setFont(f);
7. ui4->ColorsLabel_4->setFont(f);
8. [and so on]
9. ui4->ColorsLabel_9->setFont(f);

```

```

10. ui4->SelectImage_2->setFont(f);
11. Font=f;
12. [unrelated code]
13. if (isFont==10){
14. isFont=1;
15. }
16.
17. }
```

3.2 Aplicare font

```

1. ui->MBoard->setFont(Font);
2. ui->Help->setFont(Font);
3. ui->About->setFont(Font);
4. ui->Licence->setFont(Font);
5. ui->MBTerminal->setFont(Font);
6. ui->NumLockInfo->setFont(Font);
7. ui->ProjectNotes->setFont(Font);
8. ui->Rhythm2->setFont(Font);
```

Exemplu de schema de culori:

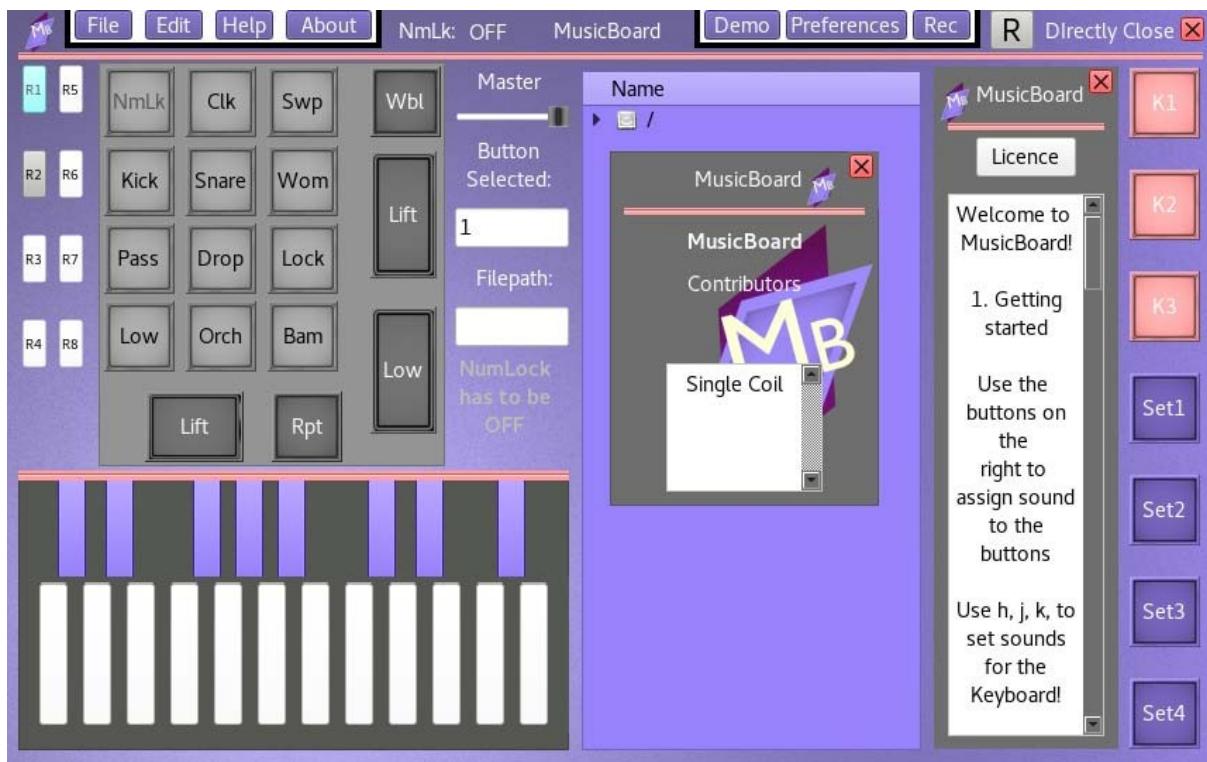


Figure 5: Color Scheme 1.



Figure 6: Color Scheme 2.

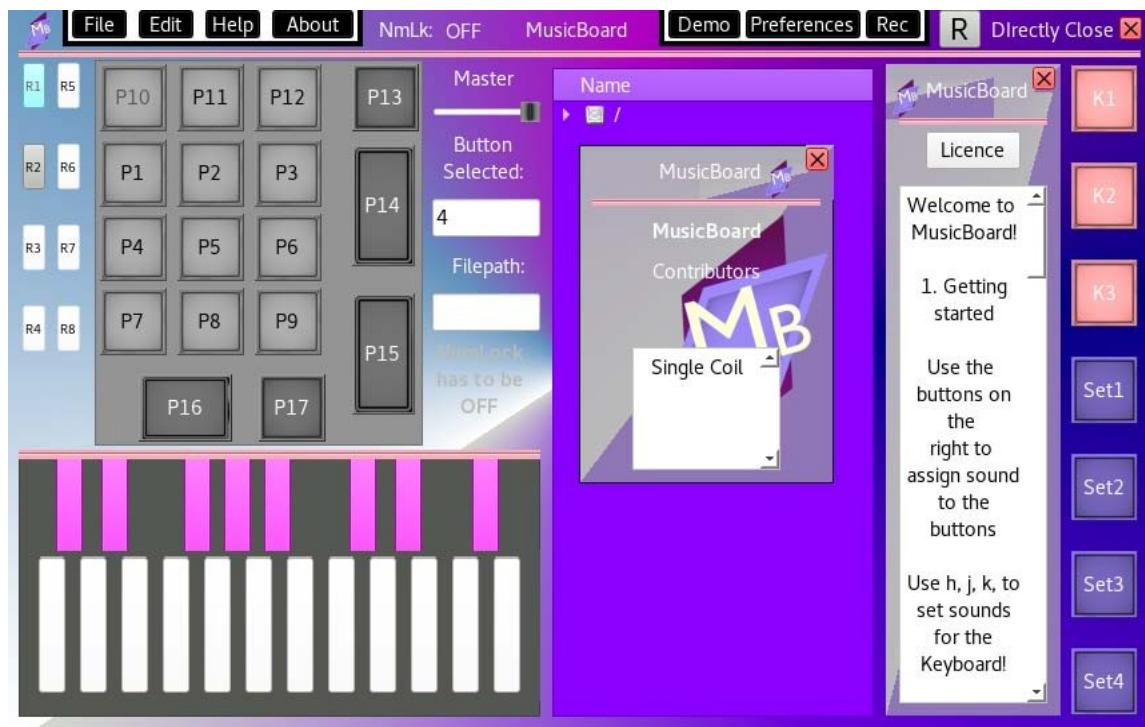


Figure 7: Color Scheme 3.

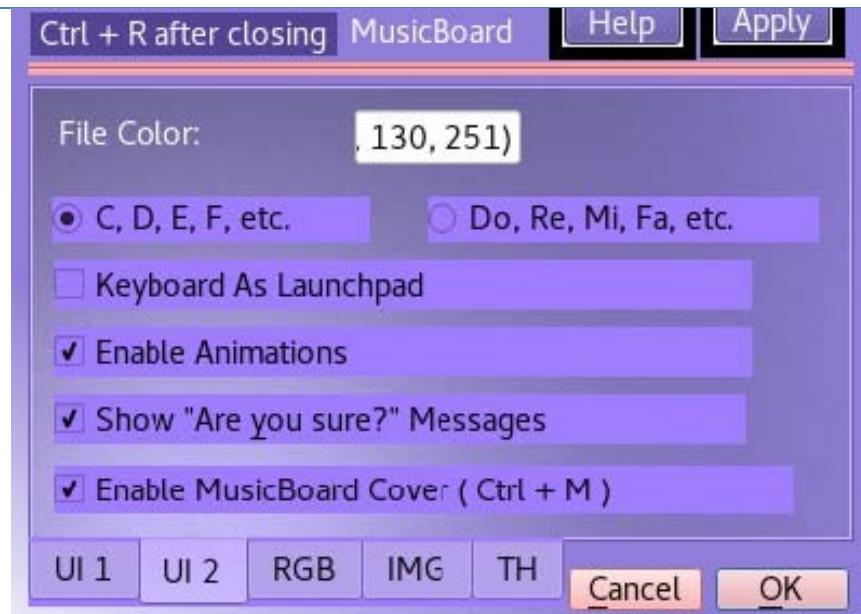


Figure 8: Panel UI2 din Preferences.

Interfață grafică UI2

Un panel de interfață grafică, UI panel, UI2 are un Font Color Selector și selectori on / off pentru:

- Note (C, D, E, F, etc. sau Do, Re, Mi, Fa, etc.)
- Keyboard ca Launchpad (atribuie presetărilor Launchpad sau sunete custom clapelor keyboard)
- Activare animație
- Vizualizare mesaje ”Are you sure?” (Ești sigur ?)
- Activare Cover de siguranță MusicBoard

Cod:

1.0 Setare note muzicale C, D, E, etc. sau Do, Re, Mi, etc.:

```

1. void PreferencesForm::on_Notes1_clicked() // C, D, E, F, etc.
2. {
3.     extern int Notes;
4.     Notes=0;
5. }
6. void PreferencesForm::on_Notes2_clicked() // Do, Re, Mi, Fa, etc.
7. {
8.     extern int Notes;
9.     Notes=1;
10. }
```

1.1 Atribuire notății muzicale C, D, E, etc, sau Do, Re, Mi, etc.

```

1. void MusicBoard::on_ReInit_clicked()
2. {
3. [unrelated code]
4.     extern int Notes;
5.     if (Notes==0){
6.         C_Text="C";
7.         [and so on]
8.         G2_Text="G2";
9.     }
10. else {
11.     C_Text="Do";
12.     [and so on]
13.     G2_Text="Sol2";
14. }
15. [unrelated code]
16. }
```

2.0 Keyboard ca Launchpad:

```

1. void PreferencesForm::on_KeyboardAsLaunchpadShow_clicked()
2. {
3.     extern int isKeyboardAsLaunchpadShow;
4.     if (isKeyboardAsLaunchpadShow==0){
5.         isKeyboardAsLaunchpadShow=1;
6.         ui4>KeyboardAsLaunchpadShow>setChecked(true);
7.     }else{
8.         isKeyboardAsLaunchpadShow=0;
9.         ui4>KeyboardAsLaunchpadShow>setChecked(false);
10. }
11. }
```

3.0 Activare Animatie:

```

1. void MusicBoard::on_Set1_pressed() /// for everything preset \c si
Launchpad Key
2. {
3.     extern int isCPUEnabled;
4.     if (isCPUEnabled==1){
5.         ui>Set1>setStyleSheet("color: rgb(255, 255, 255);backgroundimage:
url(:/pictures/Sets1.png);image: url(:/pictures/Sets1.png);backgroundcolor:
rgba(159, 138, 252, 153);bordercolor: rgba(159, 138, 252, 153);");
6.     }
7. }
```

4.0 Vizualizare mesaje "Are you sure?" (Ești sigur ?)

```

1. void MusicBoard::on_ClearLaunchpad_clicked() /// for everything alert
2. {
3. ClearWhat=2; /// determine what to clear
4. ui>ClearText>setText("Clear Launchpad?");
5. ui>Clear>show();
6. extern int isSureShow;
7. if (isSureShow==0){
8. on_ClearYes_clicked();
9. on_ClearYes_clicked(); // double clicking
10. ui>Clear>hide();
11. }
12. }
```

5.0 Activare Cover de siguranță MusicBoard

```

1. void MusicBoard::on_LButton_clicked()
2. { extern int isCoverShow;
3. if(isCoverShow==1){
4. if (isCover==1){
5. ui>Cover>hide();
6. isCover=0;
7. ui>LButton>setStyleSheet("backgroundcolor: rgb(186, 189, 182);");
8. }
9. else {
10. ui>Cover>show();
11. isCover=1;
12. ui>LButton>setStyleSheet("color: rgb(239, 41, 41); backgroundcolor:
rgb(255, 255, 255);");
13. }
14. }
15. }

```

RGB



Figure 9: Panel RGB din Preferences.

Panelul RGB panel are selectori on / off pentru eticheta NumLock Label și pentru vizualizare Imagini, Show Images. Are și un selector RGB de [16,20] culoare, RGB Color Selector. Este realizat din cursoare (0 la 255), unde fiecare setează o culoare în sirul RGB pe care utilizatorul o atribuie selectorului. Transformarea din int în QString se realizează cu `QString::number(<int variable>)`. Culorile pot fi trimise elementelor din panoul RGB.

Cod:
1.0. Obținerea numărului de culoare și setarea lui cadrului selectabil RGB:

```

1. void PreferencesForm::on_RSlider_2_valueChanged(int value)
2. {
3.     ReVal=ui4>RSlider_2>value();
4.     ui4>RgbString_2>setText("rgb("+QString::number(ReVal)+", "+QString::number(GrVal)+", "+QString::number(BlVal)+")");
5.     ui4>ColorPLabel_2>setStyleSheet("backgroundcolor: "+ ui4>RgbString_2>text());
6. }

```

2.0 Trimitearea unui sir de Culori, Color String, elementului selectat, Selected Element

```

1. void PreferencesForm::on_SendTextButton_clicked()
2. {
3.     on_ColorsCombo_6_currentIndexChanged(9);
4.     ui4>CustomColor_6>setText(ui4>RgbString_2>text());
5.     extern QString CFont;
6.     CFont=ui4>CustomColor_6>text();
7.     on_CustomColor_6_editingFinished();
8. }

```

IMG


Figure 10: IMG panel in Preferences.

Panelul IMG panel are gradienți ce pot fi aplicați etichetelor ce acoperă cadrele, frame-urile.

Cod:

1.0 După ie variabila Bg prime, ia se aplică selectește valoarea de la 0 la 10. Selecta astfel:

```

1. void MusicBoard::on_ReInit_clicked()
2. {
3. [unrelated code]
4. extern int Bg;
5. [unrelated code]
6. if (Bg==0){ui>Picture>setStyleSheet("backgroundimage:
1. url(:/pictures/Picture1_4.png); image: url(:/pictures/Picture1_4.png); backgroundcolor: rgba(0, 0, 0, 0)");}
7. [unrelated code]
8. if (Bg==10){ui>Picture>setStyleSheet("backgroundcolor: qconicalgradient (cx:0.5, cy:0.5, angle:224.8, stop:0
rgba(211, 215, 207, 255), stop: 0.189055 rgba(105, 0, 219, 255), stop:0.40796 rgba(0, 0, 131, 255), stop:
0.606965 rgba(175, 0, 194, 255), stop: 0.741294 rgba(97, 146, 189, 255), stop:1 rgba(255, 255, 255, 255))");}
9. [unrelated code]
10. }

```

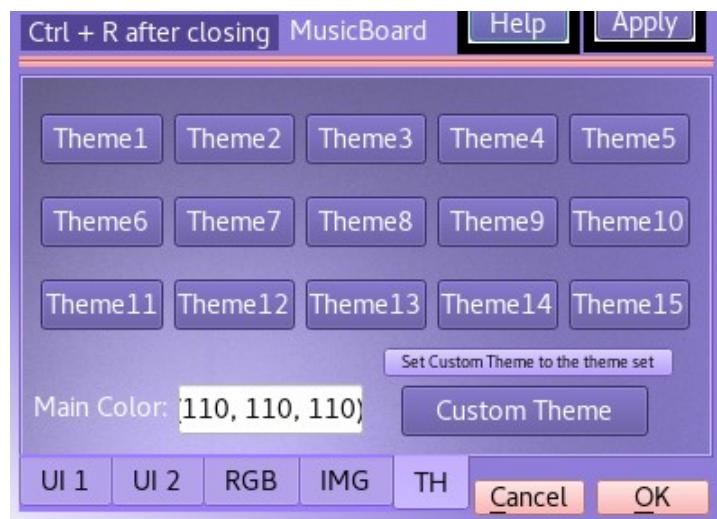
Panelul TH


Figure 11: TH panel in Preferences

In Preferences, MusicBoard are panelul TH pentru aplicarea de teme preseitate interfata ,a, si un selector ,ei utilizatorului. Există și o temă custom care poate fi creată de culori principale.

Cod:
1.0 O temă Preseitată inițializată cu Hidden Button.

```

1. void PreferencesForm::on_Theme_4_clicked()
2. {
3.     extern QString C1;
4.     extern QString C12;
5.     extern QFont CFont;
6.     extern QKeyboard CKeyboard;
7.     extern Buttons CButtons;
8.     extern QString isKeyboardsAsLaunchpadShow;
9.     extern QFont Font;
10.    extern QFile CFile;
11.    extern int Bg;
12.    extern int On;
13.    C1="qconicalgradient(cx:1, cy:0, angle:225.6, stop:0.502488 rgba(143,
118, 183, 255), stop:1 rgba(186, 186, 186, 255))";

```

```
14. C12="rgb(255, 159, 159)";  
15. CFont="rgb(255, 255, 255)";  
16. CKeyboard="rgb(255, 0, 255)";  
17. CButtons="rgb(0, 0, 0)";  
18. CFile="rgb(137, 0, 255)";  
19. Bg=10;  
20. on_HiddenReInitButton_clicked();  
21. }
```

2.0 Temă Custom:

```
1. void PreferencesForm::on_ApplyCustomTheme_clicked()  
2. {  
3.     extern QString C1_Custom;  
4.     extern QString C12_Custom;  
5.     extern QString CFont_Custom;  
6.     extern QString CKeyboard_Custom;  
7.     extern QString CFile_Custom;  
8.     extern QString CButtons_Custom;  
9.     extern int Bg_Custom;  
10.    extern QString C1;  
11.    extern QString C12;  
12.    extern QString CFont;  
13.    extern QString CKeyboard;  
14.    extern QString CFile;  
15.    extern QString CButtons;  
16.    extern int Bg;  
17.    C1_Custom=C1;  
18.    C12_Custom=C12;  
19.    Bg_Custom=Bg;  
20.    CFont_Custom=CFont;  
21.    CKeyboard_Custom=CKeyboard;  
22.    CFile_Custom=CFile;  
23.    CButtons_Custom=CButtons;  
24. }  
25. void PreferencesForm::on_Theme_clicked()  
26. {  
27.     extern QString C1_Custom;  
28.     extern QString C12_Custom;  
29.     extern QString CFont_Custom;  
30.     extern QString CKeyboard_Custom;  
31.     extern QString CFile_Custom;  
32.     extern QString CButtons_Custom;  
33.     extern int Bg_Custom;  
34.     extern QString C1;  
35.     extern QString C12;  
36.     extern QString CFont;  
37.     extern QString CKeyboard;  
38.     extern QString CFile;  
39.     extern QString CButtons;  
40.     extern int Bg;  
41.     C1=C1_Custom;  
42.     C12=C12_Custom;  
43.     Bg=Bg_Custom;  
44.     CFont=CFont_Custom;  
45.     CKeyboard=CKeyboard_Custom;  
46.     CFile=CFile_Custom;  
47.     CButtons=CButtons_Custom;  
48.     on_HiddenReInitButton_clicked();  
49. }
```

Preferences

Panelul de preferințe are și un frame de help, de ajutor.



Figure 12: Help din Preferences.



Figure 13: Text editabil pentru sunete.

Panelul Settings, de setări.

Panelul Settings de setări este utilizat pentru:

- Particularizarea denumirilor tastelor Launchpad
- Vizualizarea tastelor Keyboard
- Mixer
- Website-uri

5.15 Taste și Clape

Particularizarea denumirilor tastelor Launchpad.

Tastele Launchpad sunt editabile, li se pot schimba sau șterge denumirile.

Cod:

1.0 Clear, ștergere deumire:

```
1. void Settings::on_SettingsClearLabels_clicked()
2. {
3.     ui1>P1_Text>setText("");
4.     [unrelated code]
5.     ui1>P17_Text>setText("");
6. }
```

2.0 Utilizarea unui sir de caractere pentru denumire

```
1. void Settings::on_SettingsApplyTextButton_clicked()
2. {
3.     extern QString P1_R;
4.     P1_R=ui1>P1_Text>text();
5.     [and so on]
6.     extern QString P17_R;
7.     P17_R=ui1>P17_Text>text();
8. }
```

3.0 Aplicarea sirului de caractere ales pentru denumire

```
1. void MusicBoard::on_ReInit_clicked()
2. {
3.     [unrelated code]
4.     extern QString P1_R;
5.     ui>P1>setText(P1_R);
6.     ui>P1>setShortcut(Qt::Key_Home);
7.     [and so on]
8.     extern QString P17_R;
9.     ui>P17>setText(P17_R);
10.    ui>P17>setShortcut(Qt::Key_Delete);
11.    [unrelated code]
12. }
```

Aranjarea tastelor Keyboard.

Layoutul Keyboard este static în această versiune.



Figure 14: Keyboard Layout.

5.16 Mixer

Mixer-ul [13] are incluse slidere [39], cursoare de volum pentru toate tastele de sunet din MusicBoard. Master Volume face ca toate cursoarele să aibă aceeași valoare în acest release. Sunetul se poate seta la orice volum în panelul de setări Settings, prin click pe slider [46].

Sunt trei categorii de control de volume:

- Volum Master
- Volum Keyboard
- Volumele de Launchpad și Ritm

Fiecare au atașate varibile globale [41] ce activează toate clasele cu care comunică. După declarare în main.cpp, variabila int MasterVolume; de exemplu este inclusă în settings.cpp și musicboard.cpp prin declararea extern int MasterVolume; .

Există variabilă globală pentru fiecare tastă/clapă Launchpad și ritm. Acestea sunt atribuite de elemente de slider (cu butonul Apply, reinițializate cu setSliderPosition(<global variable>) și setSliderValue(<global variable>)) din settings.ui.

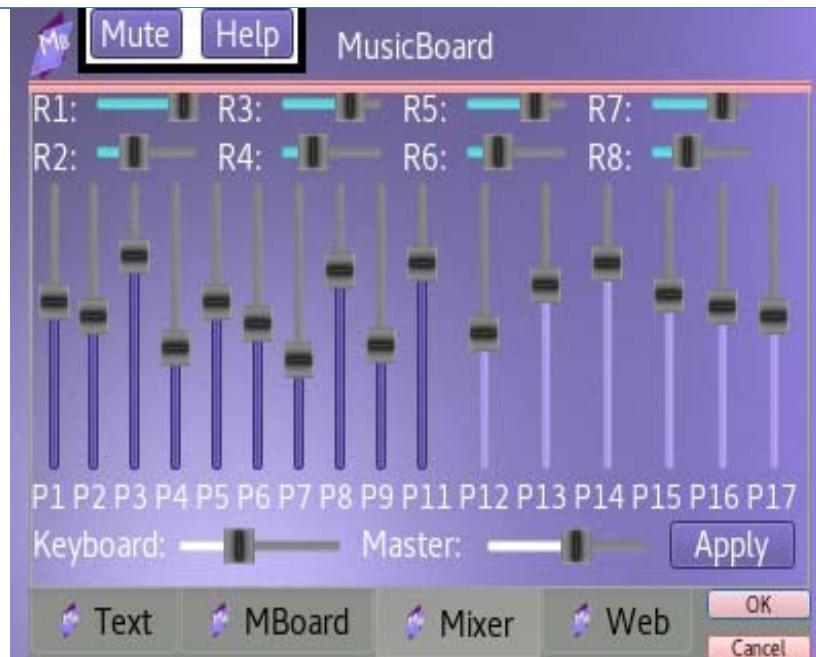


Figure 15: MusicBoard Mixer.

Cod:

1.0 Atribuirea valorii unui Slider unei variabile globale:

```

1. void Settings::on_SettingsApplyMixerButton_clicked()
2. {
3.     extern int R1_A;
4.     [and so on]
5.     extern int MasterVol;
6.
7.     P1_A=uil>P_1>value();
8.     [and so on]
9.     MasterVol=uil>MasterSlider>value();
10. }
```

2.0 Mute:

```

1. void Settings::on_SettingsMuteButton_clicked()
2. {
3.     uil>P_1>setSliderPosition(0);
4.     [and so on]
5.     uil>MasterSlider>setSliderPosition(0);
6.
7.     uil>P_1>setSliderPosition(0);
8.     [and so on]
9.     uil>MasterSlider>setValue(0);
10. }
```

3.0 Atribuirea volumului unui sunet:

```

1. void MusicBoard::on_ReInit_clicked()
2. {
3.     extern int P1_A;
4.     [unrelated code]
5.     extern int R7_A;
6.
7.     P1>setVolume(P1_A);
8.     [unrelated code]
9.     R7>setVolume(R7_A);
10. }
```

4.0 Nivelul de volum:

```

1. void Settings::on_P_1_sliderPressed()
2. {
3.     on_SettingsApplyMixerButton_clicked();
4.     MusicBoard a; /// fct. in MusicBoard
5.     a.on_P1_clicked();
6. }
```

5.17 Website

Website-urile [40] sunt:

- Site MusicBoard: <http://musicboard-launchpad.dx.am/>
- Blog MusicBoard: <https://musicboard-blog.weebly.com/>
- GitHub Repository [48]: <https://github.com/MusicBoard/musicboard>
- Website GitHub Repository [48]: <https://github.com/MusicBoard/musicboard> website
- Blog muzical al autorului: <https://single-coil.weebly.com/>

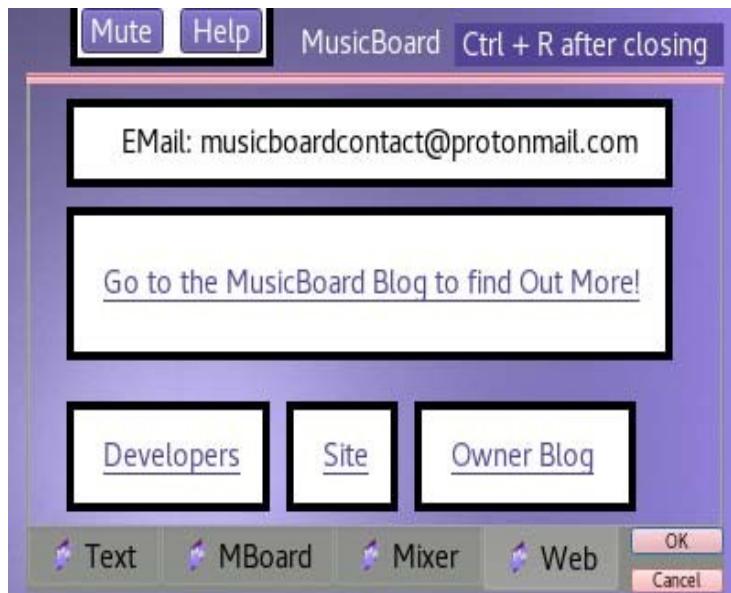


Figure 16: Website-uri.

Cod:

```

1. ui1>LabelWeb1>setText("<a style=\"color:#534496;\""
href=\"https://github.com/MusicBoard/musicboard\">Developers</a>");
2. ui1>LabelWeb1>setTextFormat(Qt::RichText);
3. ui1>LabelWeb1>setTextInteractionFlags(Qt::TextBrowserInteraction);
4. ui1>LabelWeb1>setOpenExternalLinks(true);
```

Panelul de setări, Settings panel cu frame-ul de help.

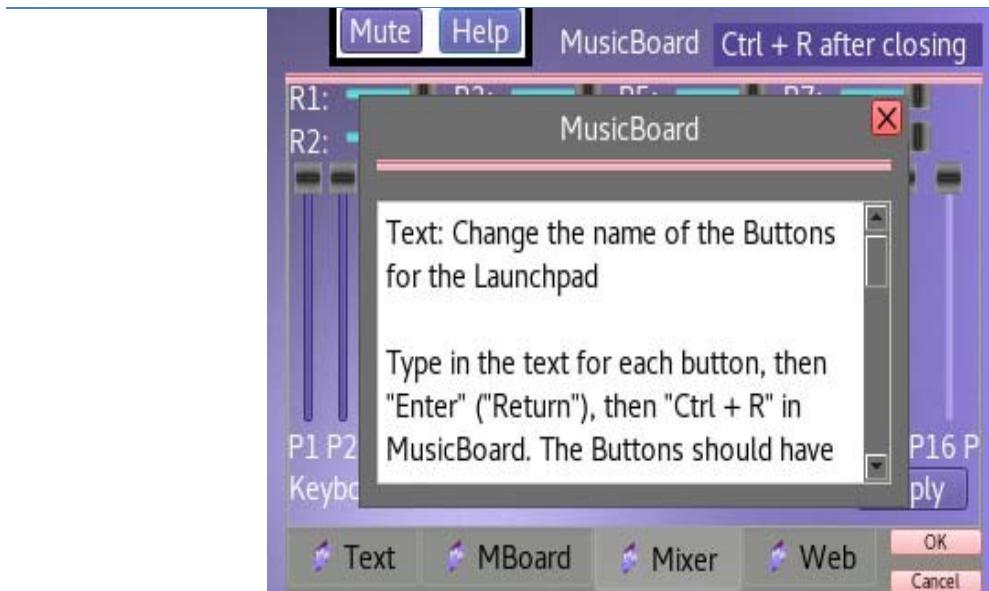


Figure 17: Help.



Figure 18: MusicBoard Recorder.

5.18 Recorder, înregistrari.

Posibilitatea de înregistrare a sunetelor în MusicBoard se realizează pe baza exemplului de audio recorder din exemplele implicate Qt [5,33].

Cod:

```
1. ui5>setupUi(this);
2.
3.     extern QString C1;
4. [and so on]
5.     extern int BgColor2;
6.
7.     ui5>Picture>setStyleSheet("backgroundcolor: "+C1+"); // bg.
selection
8.     if (Bg==0){ui5>Picture>setStyleSheet("backgroundcolor:
qlineargradient(spread:pad, x1:0, y1:1, x2:0.313, y2:0.630682, stop:0
rgba(255, 255, 255, 255), stop:1 rgba(142, 125, 217, 255));}
9. [and so on]
10.    if (Bg==10){ui5>Picture>setStyleSheet("backgroundcolor:
qconicalgradient(cx:0.5, cy:0.5, angle:224.8, stop:0 rgba(211, 215, 207,
255), stop:0.189055 rgba(105, 0, 219, 255), stop:0.40796 rgba(0, 0, 131,
255), stop:0.606965 rgba(175, 0, 194, 255), stop:0.741294 rgba(97, 146, 189,
255), stop:1 rgba(255, 255, 255, 255));}
11.    if (Bg==14){ui5>Picture>setStyleSheet("backgroundcolor:
qlineargradient(spread:pad, x1:0, y1:1, x2:0.313, y2:0.630682, stop:0
rgba(255, 255, 255, 255), stop:1 rgba(142, 125, 217, 255));}
12.
13.    ui5>L1>setStyleSheet("backgroundcolor: "+C12+"); // line color
14.    if (CButtons.length()<17){ // bts. corresponding color determination
15.        ui5>outputButton>setStyleSheet("backgroundcolor: "+CButtons+"; color:
rgb(255, 255, 255);");
16.        ui5>pauseButton>setStyleSheet("backgroundcolor: "+CButtons+"; color:
rgb(255, 255, 255);");
17.        ui5>recordButton>setStyleSheet("backgroundcolor: "+CButtons+"; color:
rgb(255, 255, 255);");
18.    }
19.    else {
20.        ui5>outputButton>setStyleSheet("backgroundcolor: "+CButtons+"; color:
rgb(0, 0, 0);");
21.        ui5>pauseButton>setStyleSheet("backgroundcolor: "+CButtons+"; color:
rgb(0, 0, 0);");
22.        ui5>recordButton>setStyleSheet("backgroundcolor: "+CButtons+"; color:
rgb(0, 0, 0);");
23.    }
24.
25.    ui5>label_12>setStyleSheet("color: "+CFont+"); "
26. [and so on]
27.    ui5>label_34>setStyleSheet("color: "+CFont+"); "
```

5.19 Imagini

MusicBoard include doar imagini realizate de autor, chiar dacă sunt localizate în aplicația principală, butoane sau logo, toate sunt realizate în Blender [6], un software gratuit de modelare grafică.

5.20 Implementare

Implementarea acestei versiuni a aplicației este Linux 64 și 32 biți. Software-ul, MusicBoard, se descarcă ca fișier arhivat **.zip** cu:

- fișer de instalare **.deb** [10, 24]
- fișer executabil **.sh**
- fișer **Readme** (pentru instalare și lansare în execuție)
- fișer **Help.txt**
- fișer **To_Do.txt**
- fișer **Licence.txt** [29]
- fișer **Q&A**



Figure 19: Fișier descărcat .deb [10] și dezarchivat pentru instalare ulterioară executabilului MusicBoard.



Figure 20: Folderul Sh din fișierul descărcat.

6 Planificarea dezvoltărilor și facilităților viitoare

MusicBoard are posibilitate dezvoltare [27]. O listă cu posibile îmbunătățiri este detaliată aici.

- Transpoziții (să se schimbe C în E de exemplu, și restul notelor să urmeze în continuare)
- Setare Ritm, Keyboards, Settings pentru Pads (este realizat parțial, dar ritmurile nu se pot repeta, sunt doar fișiere de sunete foarte lungi)
- Setare Volum
- Setare Icon-uri
- Creare și includere imagini
- Save, Save as, etc.
- Înregistrare internă, Record
- MIDI
- Setare implicită la filepath
- Setare sensitivitate clape, la apăsare lungă sau scurtă pe clape /taste
- Realizare buton "Clear all"
- Tasta P17 -> (Shortcut "Del"; acum sunt șterse)
- Realizare de Widgets separat
- Ultima clapă last neagră (violet) să rămână fixă duă mai multe compilări (cu minimum și maximum lungime și lățime)
- Conturi de Social Media pentru promovarea aplicației MusicBoard [32]
- Copyright (Copyrighted.com)
- Testare pe diverse sisteme de operare.
- Aplicatia are posibilitatea să ajungă un external virtual MIDI controller pentru LMMS (www.lmms.io) software gratuit pentru compoziție muzicală in Linux [11].
- MusicBoard aplicație online.

Appendix

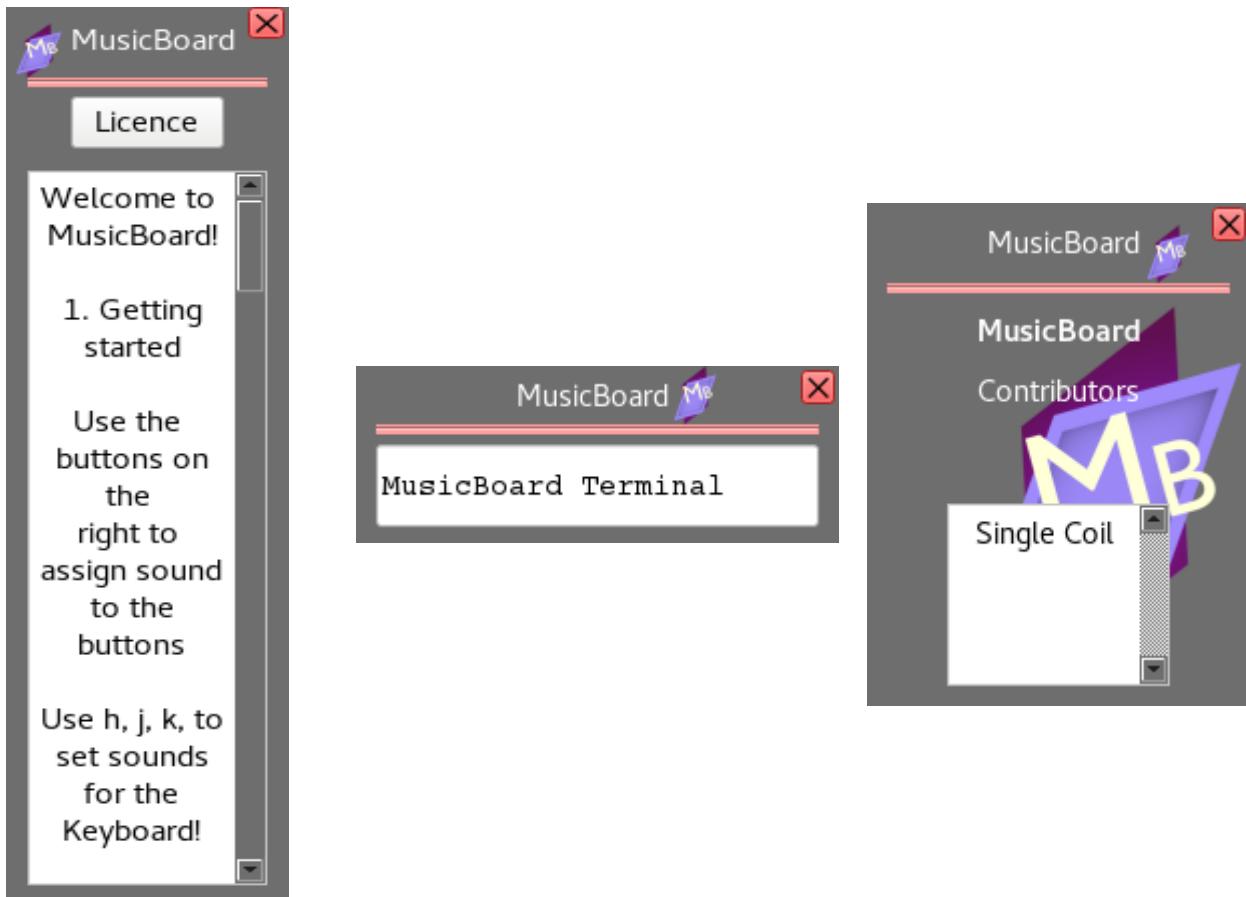


Figure 21: MusicBoard Help, TERMINAL și About.

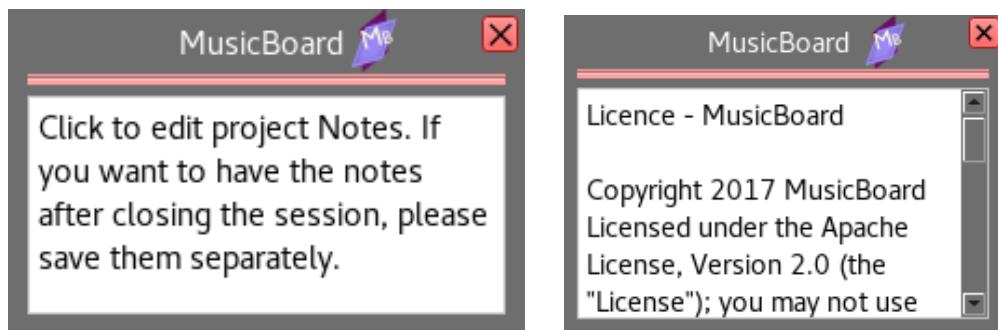


Figure 22: MusicBoard notițe, Notes și Licență.

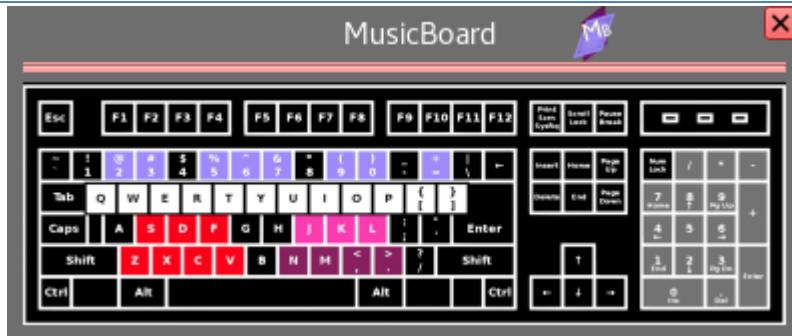


Figure 23: Keyboard Layout.

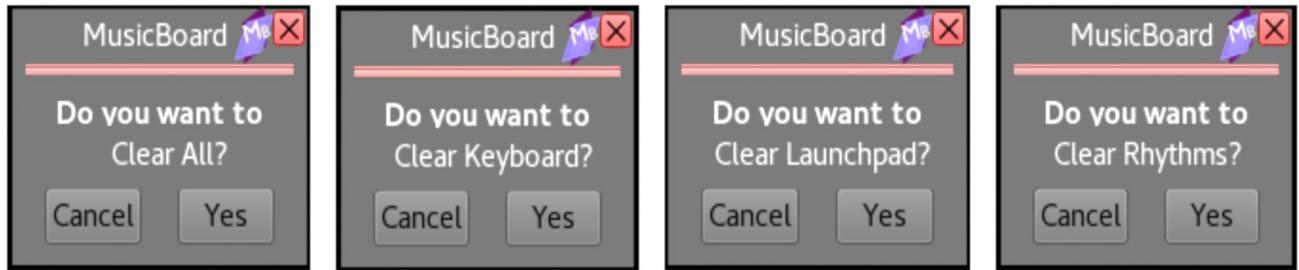


Figure 24: Paneluri cu mesaje pentru siguranță ștergerii.

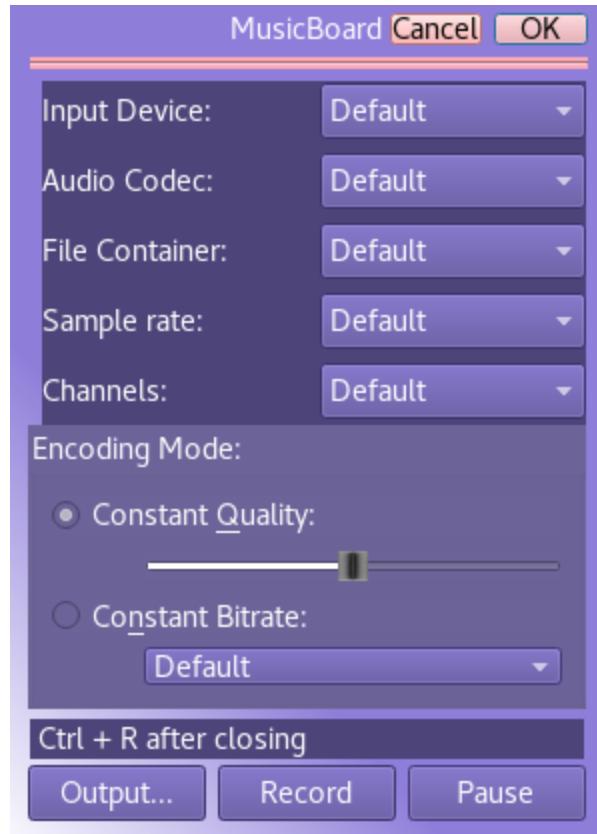


Figure 25: Audio Recorder.



Figure 26: R2 Multiple Sound Grid.

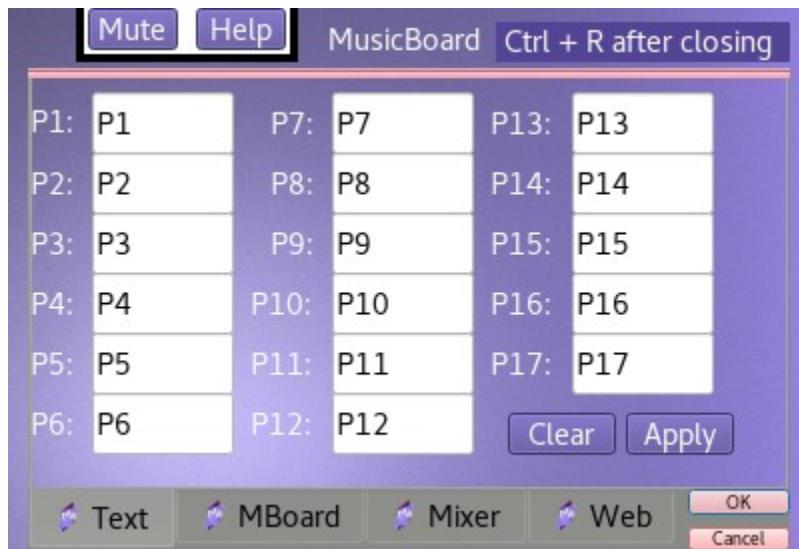


Figure 27: Settings Text.



Figure 28: Preferences UI1.

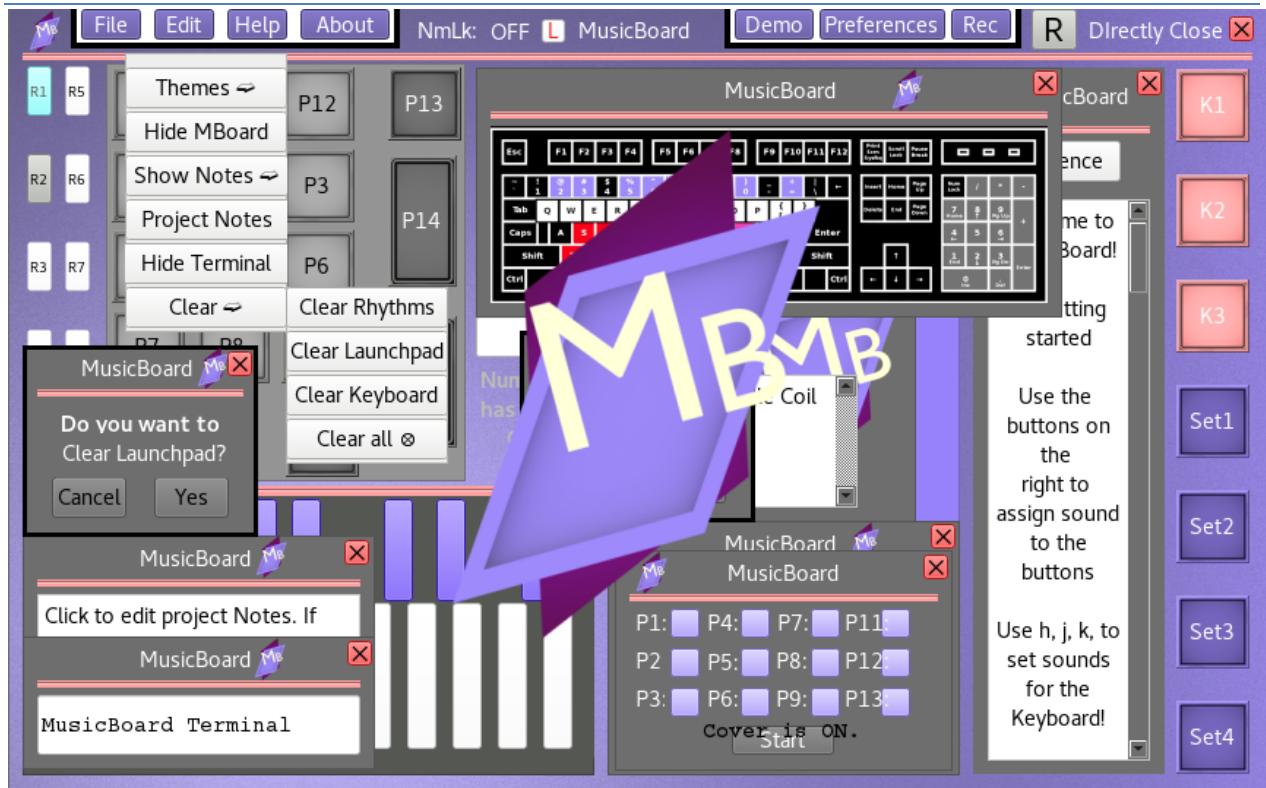


Figure 29: MusicBoard 1.

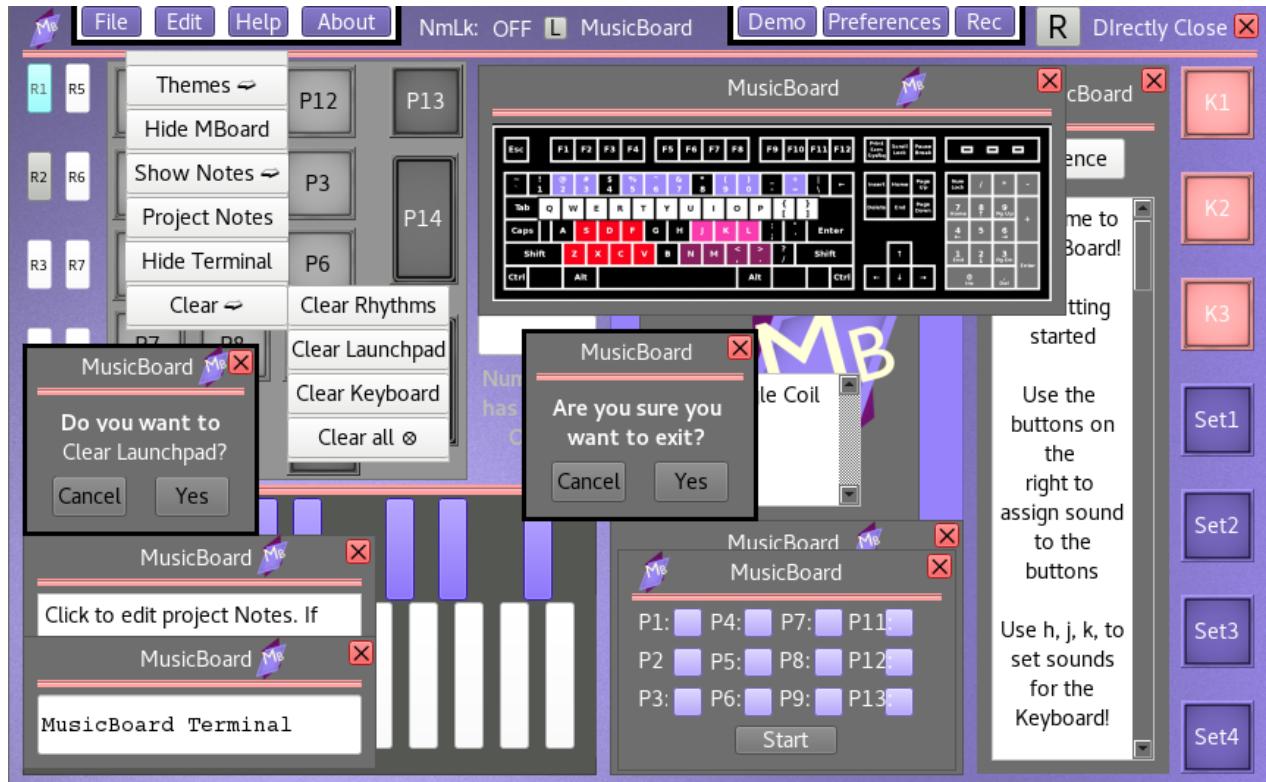


Figure 30: MusicBoard 2.



Figure 31: MusicBoard 3.



Figure 32: MusicBoard 4.

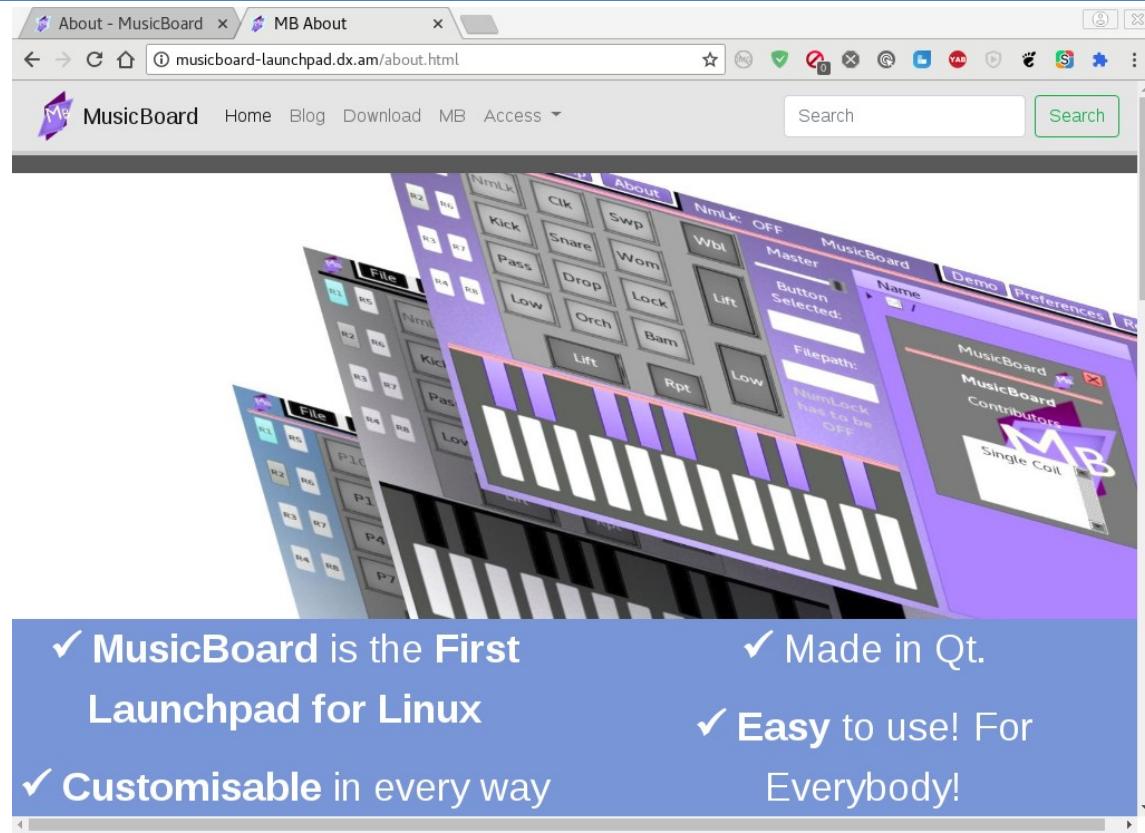
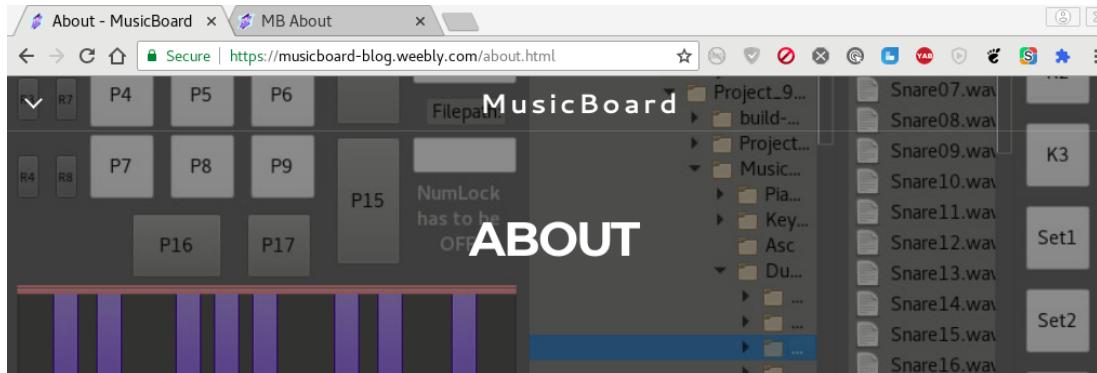


Figure 33: MusicBoard Website Ofcial.



What is MusicBoard?

MusicBoard is a software that aims to take the musician's eyes off of the computer screen. This is to be done with great attention to how we use the keyboard to control as many musical parameters. In order to do so, the software will be aimed at the mode with the computer screen off. This means that it will have to be able to make complex rhythms and modulate samples without effort. Download MusicBoard for your computer!

[DOWNLOAD MUSICBOARD! NOT A DIME REQUESTED!](#)

Share!



This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.

Figure 34: Blog Oficial MusicBoard.

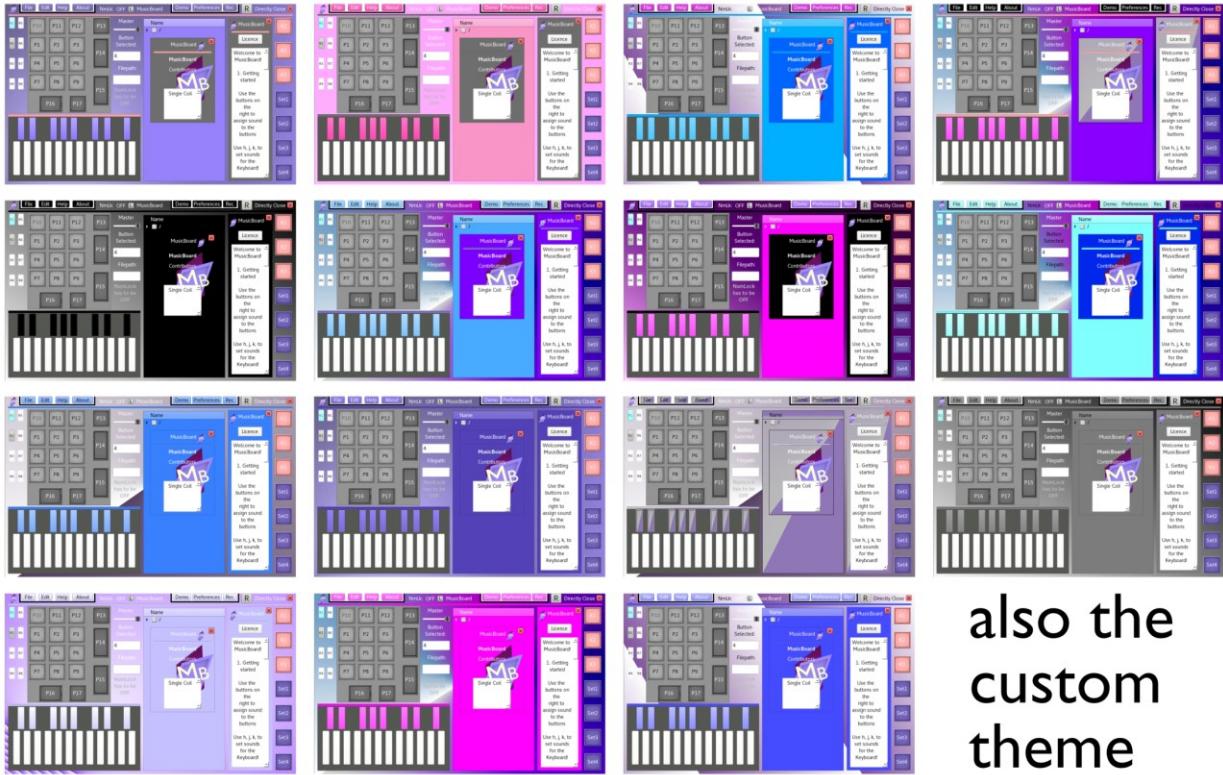


Figure 35: MusicBoard Preset Themes.

also the
custom
theme



Figure 36: Firs, 52//64=\\$2.54/100, second 52/64=82.54/100 and third 41/64=80.95/100, Color Selection Visible.

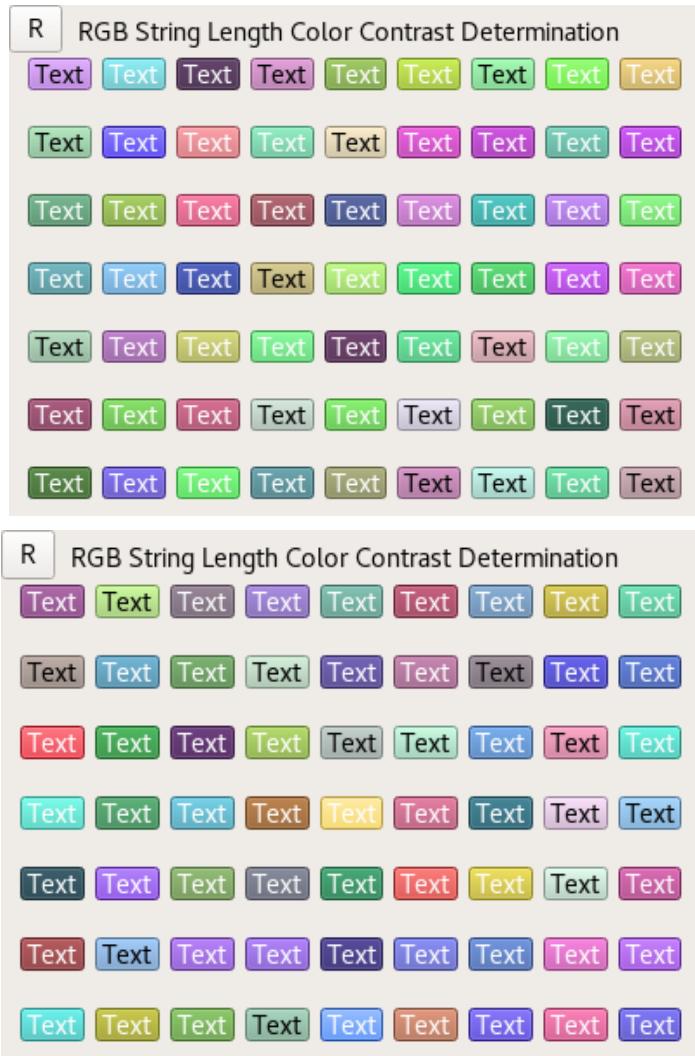


Figure 37: Fourth $49/64=76.2/100$ and fifth $54/64=85.71/100$ Color Selection Visible.

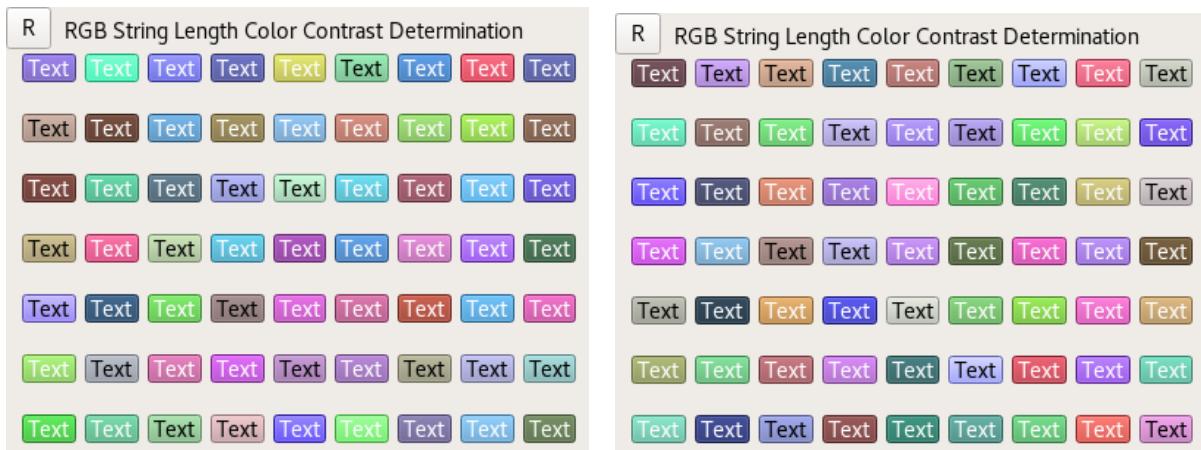


Figure 38: Sixth $54/64=85.71/100$ and seventh $58/64=92.06/100$ Color Selection Visible.

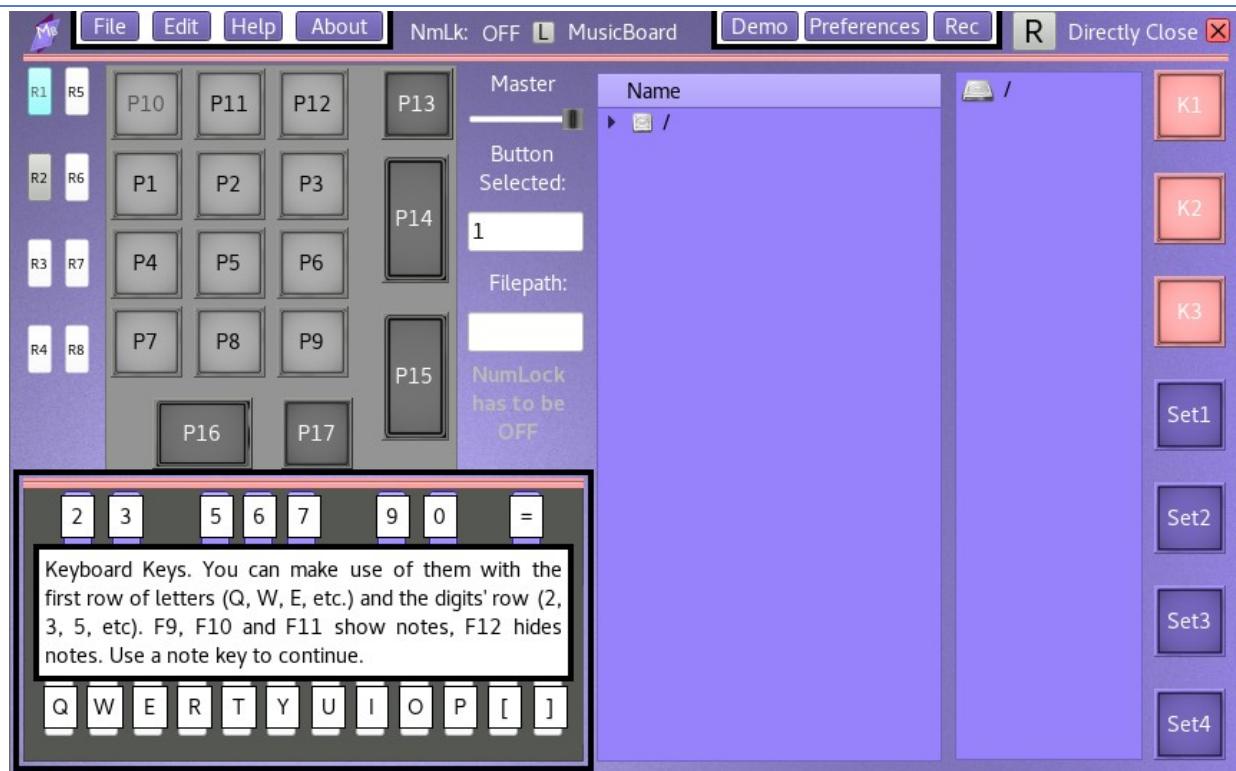


Figure 39: Demo 2.

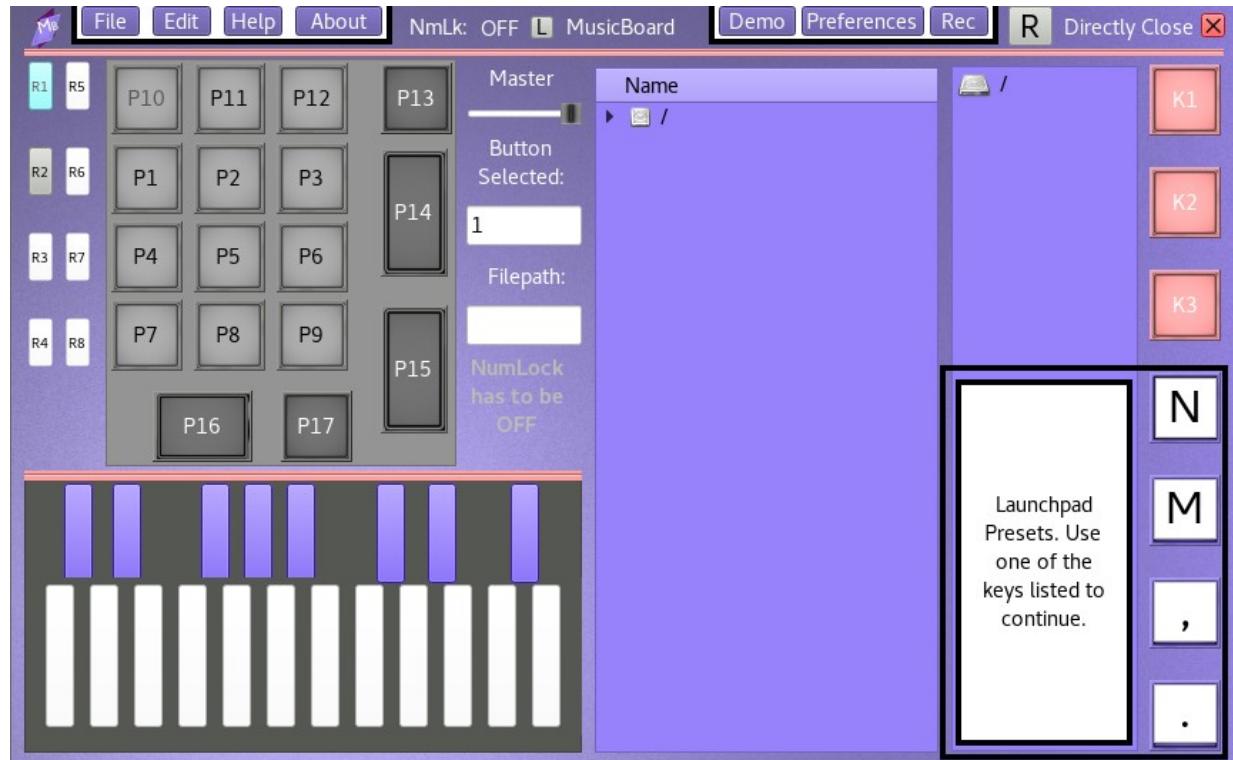


Figure 40: Demo 3.

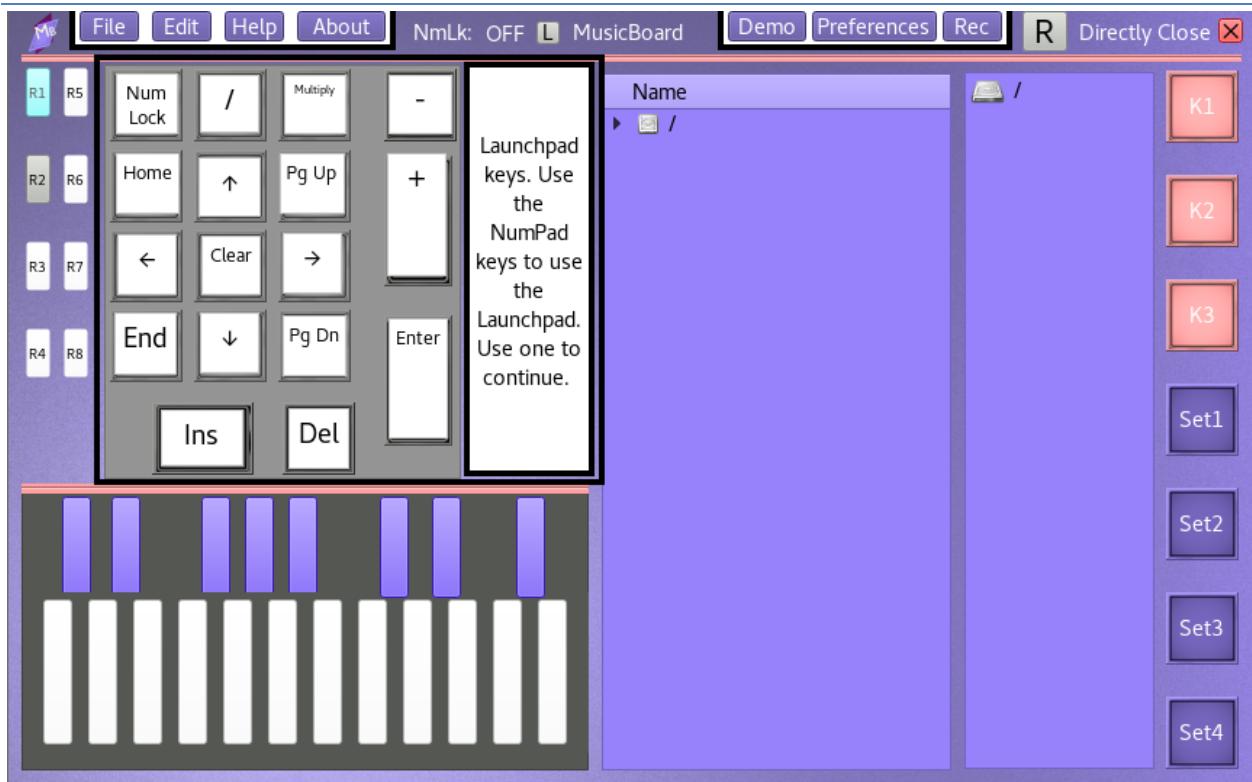


Figure 41: Demo 4

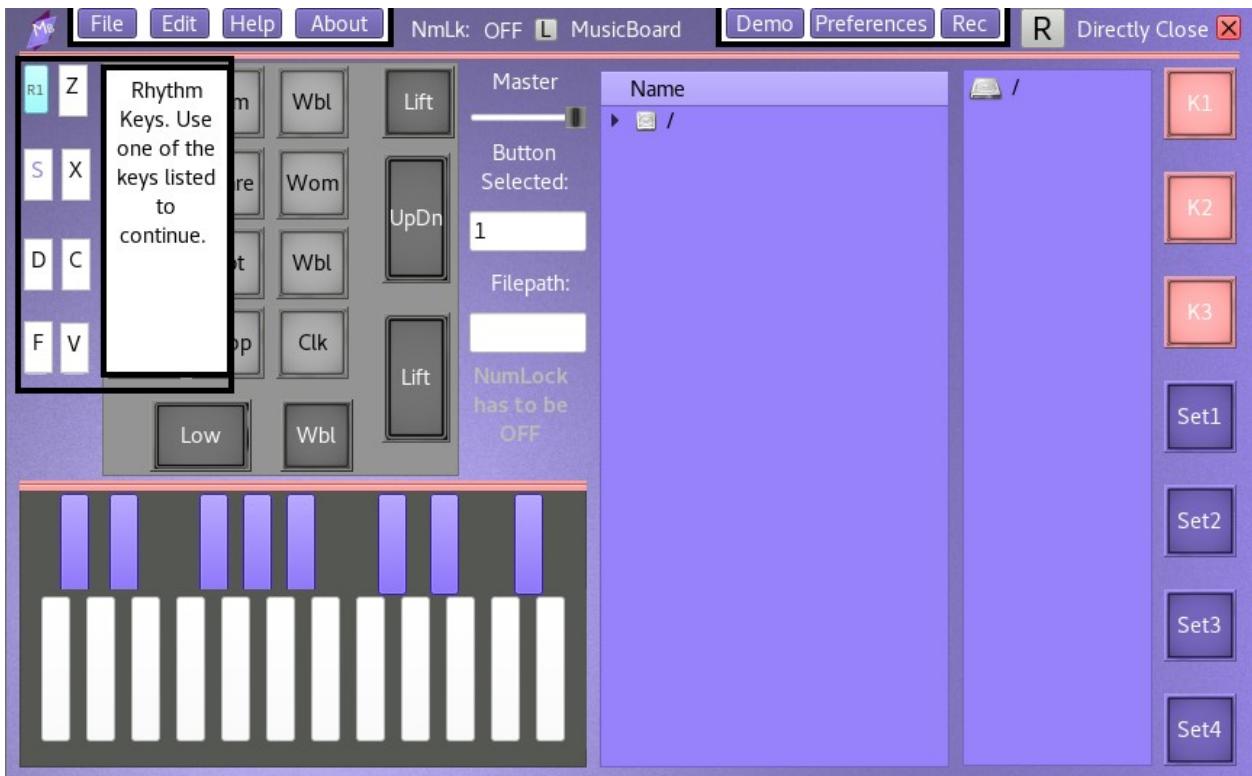


Figure 42: Demo 5.

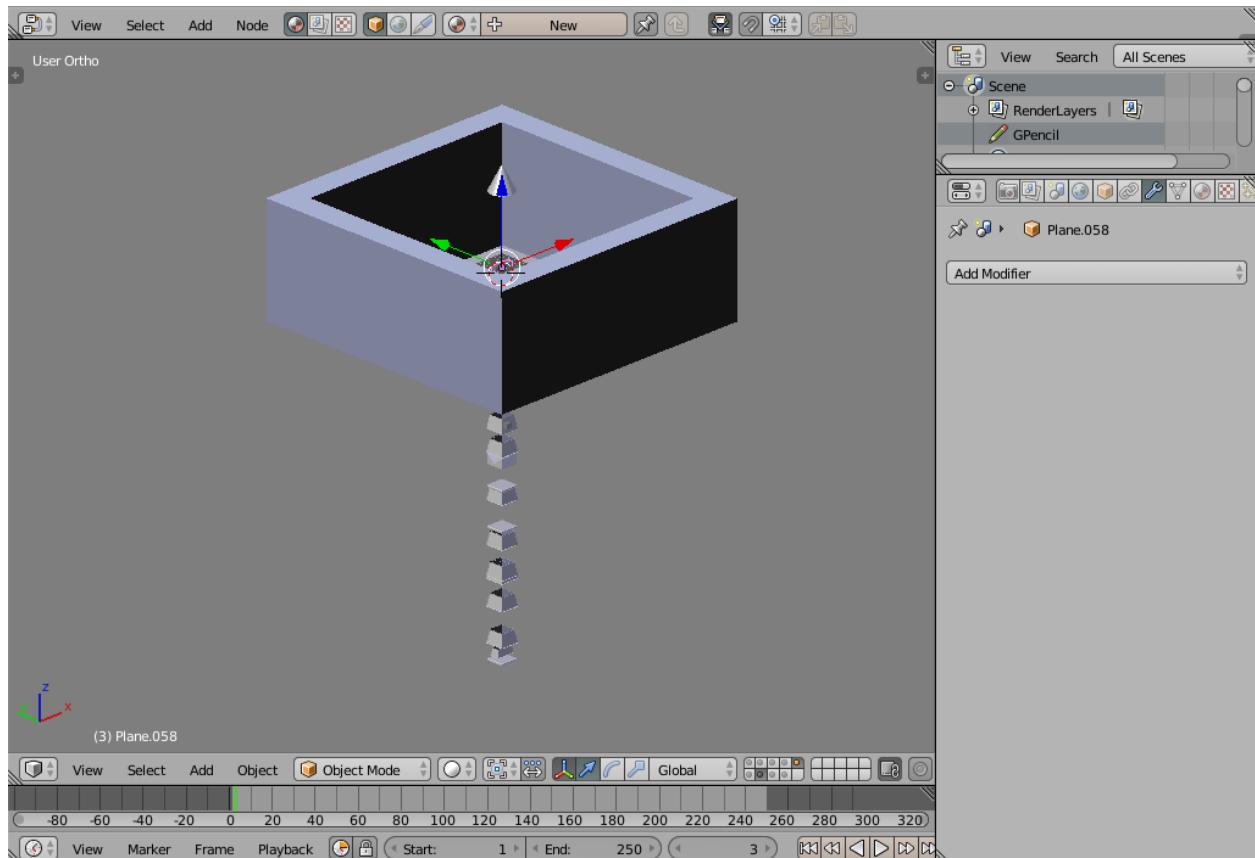


Figure 43: Blender Software. Making buttons for MusicBoard.

Licence.txt

Licence - MusicBoard

-----|
Copyright 2017 MusicBoard Licensed under the **Apache License**, Version 2.0
(the "License"); you may not use this file except in compliance with the
License. You may obtain a copy of the License at

www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
License for the specific language governing permissions and limitations
under the License.

Also Licenced under Creative Commons Attribution-ShareAlike 4.0
International Licence.

Help.txt

Welcome to MusicBoard!

1. Getting started

Use the buttons on the right to assign sound to the buttons

Use h, j, k, to set sounds for the Keyboard!

Use n, m, ",", "." to set sounds for the Keyboard and for the Launchpad!

Use the Keyboard to play the Keyboard!

(q, w, e, r, t, y, u, i, o, p, 2, 3, 5, 6, 7, 9, 0, =)

Use the Keyboard to play the Launchpad!

(Home, UpArrow, Pg Up, LeftArrow, 5, RightArrow, End, DownArrow, Pg Dn, Ins,
Del, /, Multiply, -, Plus, Enter)

Quick Keys

" ` " = Show / Hide Project Notes

-- If Edit is shown --

Ctrl + A = Clear All

Ctrl + L = Clear Launchpad

Ctrl + W = Clear Keyboard

--

Ctrl + H = Slow / Hide Help

Ctrl + I = Show / Hide About

Ctrl + L = Show / Hide MBoard

Esc = Close all the Wdos

Ctrl + Esc = Close the application directly

Ctrl + E = Close the application

Ctrl + A = Show Settings

Enter (In Settings) = Close Settings

Ctrl + S (In Settings) = Save Settings

Ctrl + R (In MusicBoard) = Apply / Set Settings

F6 = Increase Master

F7 = Decrease Master

Ctrl + M = Show/Hide Cover (if enabled in Preferences)

Bibliografie

- [1.] Shneor, Rotem (2012) Handbook of Research on Born Globals, Chapter 10: Born Global Firms, Internet, și New Forms of Internationalization. Retrieved 8 June 2017.
- [2.] Alexandru P. Simple Calculator made in Qt, GitHub AlexPintea.
- [3.] C++ Documentation: <http://devdocs.io/cpp/>
- [4.] W3 Schools: <https://www.w3schools.com/html/>
- [5.] Qt Framework at: <http://doc.qt.io/>
- [6.] Blender at: www.blender.org
- [7.] Audacity at: www.audacityteam.org
- [8.] LMMS at: www.lmms.io
- [9.] MusicRadar at: <https://www.musicradar.com/news/tech/free-music-samples-download-loops-hits-and-multis-627820>
- [10.] GDebi at: <https://gdebi.en.uptodown.com/ubuntu>
- [11.] Linux Documentation: <https://www.tldp.org/>
- [12.] Music Theory: <https://cnx.org/exports/2ad74b7b-a72f-42a9-a31b-7e75542e54bd\%403.74.pdf/understanding-basic-music-theory-3.74.pdf>
- [13.] Mixer Documentation: <https://www.mediacollege.com/audio/mixer/channel.html>
- [14.] Launchpad Documentation: <https://global.novationmusic.com/launch/launchpad#>
- [15.] Launchkey Documentation: <https://us.novationmusic.com/keys/launchkey>
- [16.] RGB Code Documentation: <https://www.rapidtables.com/web/color/RGBColor.html>
- [17.] About Qt: <https://stackoverflow.com/questions/41926059/offlineqt-documentation>
- [18.] Informatics Theory: <https://www.slideshare.net/kexonlagod/nursing-informaticstheoriesmodelsandframeworks>
- [19.] Html si CSS: <https://www.w3schools.com/>
- [20.] RGB Color Determination <https://24ways.org/2010/calculatingcolorcontrast/>
- [21.] File Browser Documentation: <http://doc.qt.io/qt5/qtwidgetsdialogsfindfiles-example.html>
- [22.] Computer Standards: <https://codebeamer.com/cb/wiki/8690>
- [23.] About the Terminal Rojas, R. (1998) How to make Zuse's Z3 a universal computer IEEE Annals of the History of Computing. 20 (3): 51–54.doi:10.1109/85.707574.
- [24.] Making a Deb installer: <https://wiki.debian.org/HowToPackageForDebian>
- [25.] About Polyphony: http://www.waysidemusic.com/MusicProducts/Poliphony-Poliphony_15AAspc064.aspx
- [26.] About the UI: <http://doc.qt.io/qt5/topicsui.html>
- [27.] Software Development at: Hasted, Edward (2005). Software That Sells: A Practical Guide to Developing și Marketing Your Software Project. Wiley Publishing. ISBN 0764597833.
- [28.] Rand. function at: <https://doc.qt.io/qt5.11/qrandomgenerator.html>
- [29.] Licences at: <https://choosealicense.com/>
- [30.] NumLock at: <http://doc.qt.io/archives/qt4.8/qwskeyboardhandler.html>
- [31.] Phone compatible Virtual Launchpad: <https://apkpure.com/hip-hop-drum-pads-24/com.paullipnyagov.hiphopdrumpads24>
- [32.] SEO at: <https://moz.com/beginnersguidetoseo>
- [33.] Audio recorder: <http://doc.qt.io/archives/qt5.5/qtmultimediaaudiorecorder-example.html>
- [34.] How to make sound <https://doc.qt.io/Qt5/audiooverview.html>
- [35.] How to install QMediaPlayer <https://github.com/luminadesktop/lumina/issues/164>
- [36.] How to install QFileListModel <http://doc.qt.io/qt5/qfilesystemmodel.html>

- [37.] How to configure the filepath [https://stackoverflow.com/questions/17334937/howto-icreateanabsolutefilepathfromaqdirandarelativefilepath](https://stackoverflow.com/questions/17334937/how-to-create-an-absolute-filepath-from-a-qdir-and-a-relative-filepath)
- [38.] How to make a QDialog [https://stackoverflow.com/questions/6743970/qtmenuhow-to-create-a-new-window](https://stackoverflow.com/questions/6743970/qt-menu-how-to-create-a-new-window)
- [39.] How to set Slider Image [https://stackoverflow.com/questions/11533442/howtoshowimagesinslidestyleinqt](https://stackoverflow.com/questions/11533442/how-to-show-images-in-slider-style-in-qt)
- [40.] How to set a Web Address [https://stackoverflow.com/questions/10250332/howtoopenalinkinadefaultuserbrowsingqt](https://stackoverflow.com/questions/10250332/how-to-open-a-link-in-a-default-user-browser-in-qt)
- [41.] How to set Global variables url
[https://stackoverflow.com/questions/23922319/globalvariablesinqt53](https://stackoverflow.com/questions/23922319/global-variables-in-qt5)
- [42.] How to set an image to a label
[https://stackoverflow.com/questions/16013027/settinganimagetoalabelinqt](https://stackoverflow.com/questions/16013027/setting-an-image-to-a-label-in-qt)
- [43.] How To set element visibility [https://stackoverflow.com/questions/9260169/is-thereanywaytosetthevisibilityofqlabelinqtdesigner](https://stackoverflow.com/questions/9260169/is-there-any-way-to-set-the-visibility-of-qlabel-in-qt-designer)
- [44.] How To install Qt <http://doc.qt.io/qt5/gettingstarted.html>
- [45.] How to install Qt Creator [https://www.ics.com/blog/gettingstartedqtandqt-creatorlinux](https://www.ics.com/blog/getting-started-qt-and-qt-creator-linux)
- [46.] How to set slider position <http://doc.qt.io/qt5/qslider.html>
- [47.] How to set read only <http://doc.qt.io/archives/qt4.8/qlineedit.html>
- [48.] How to GitHub [https://product.hubspot.com/blog/gitandgithubtutorialforbeginners](https://product.hubspot.com/blog/git-and-github-tutorial-for-beginners)
- [49.] How to index files as resource <http://doc.qt.io/qt5/resources.html>
- [50.] How to tackle qrc res.o [http://www.qtcentre.org/threads/51579thequotqrc_rescpp-quotdoesntexistaftercompiling](http://www.qtcentre.org/threads/51579-the-qrc-resource-does-not-exist-after-compiling)
- [51.] How to declare Qt ShortCuts <http://doc.qt.io/archives/qt4.8/qshortcut.html>
- [52.] Online Launchpad: <http://samplerulator.com/>
- [53.] Online version of the official Launchpad: <https://intro.novationmusic.com/analoguejewels>
- [54.] Tablet compatible Virtual Launchpad with customizable sounds. [https://apkpure.com/dj-dubstepmusicmakerpad/com.elfizmedia.Djdubstepmscmkrpd](https://apkpure.com/dj-dubstep-music-maker-pad-com.elfizmedia.Djdubstepmscmkrpd)