Lfo

General Purpose Programming Language

Alexandru Pintea-clasa XI-Profesor coordonator Ilea Lucian
Colegiul National Emil **Racoviţă Cluj-Napoca**

ALEGEREA TEMEI

- **Am ales aceasta tema datorita interesului pentru muzica** al participantului, si datorita lipsei de optimizare a limbajelor de programare curente.
- Un astfel de limbaj de programare nu permite mai mult de un tip de alocare de memorie.

- ANALIZA - **Am efectuat studiul pieţei:**

- **JavaScript, Python**: Allocation is made by Java, but it is not customizable.
- **C++**: Allocation is made by the programmer, which, even if optimal (faster at runtime), is challanging.
- **C#**: Too many simmilar classes
- **Swift**: Slow Runtime
- **Definirea Problemei: - imposibilitatea de a alege modul de alocare a memoriei (de catre softist sau de catre soft) folosind un limbaj de tip interpretor.**
-

Lfo este un limbaj de programare general purpose bazat pe C++, care este focusat pe alocare avansata de memorie.

Am realizat 2 fisiere `sh`, pentru platforma Linux.

- `lfointerpretor` = interpretator de limaje high-level
- `lfocreator` = IDE pentru lfointerpretator **+ 4 demos**

AVANTAJE

- **Sintaxa usor de utilizat**
- **Sintaxa faciliteaza inbunatatirea runtime**
- **Functioneaza fara instalare**
- **IDE Customizabil**
- **Extensie unica .lfop**
- **Functii predefinite implementate avansat**

PUBLICUL ŢINTĂ al aplicaţiei sunt **programatorii**.

Tehnologii - Am utilizat framework-ul Qt folosit şi la dezvoltarea Opera şi Skype.

- Am utilizat LMMS si Audacity pentru editarea sunetului si Blender pentru grafica.
- Imaginile sunt originale, iar sunetele gratuite (www.musicradar.com);
- Licenţa: 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 interfaţă grafică şi fără interfaţă grafica, de ex. servere

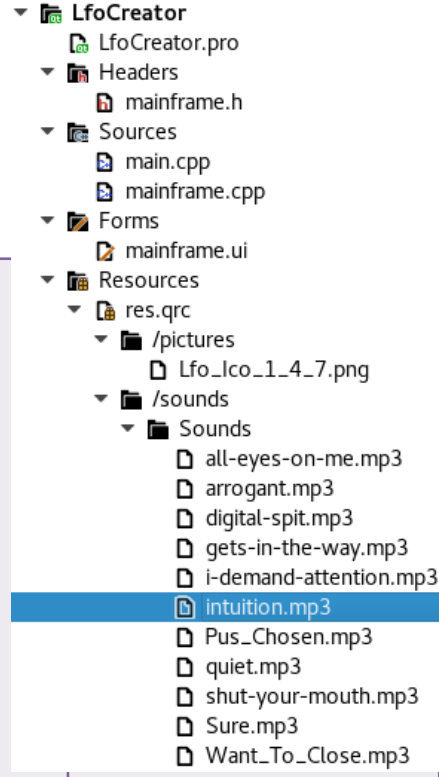
Design Blender- User Interface (UI)-3 ani experienţă

- aplicaţii de animaţie grafică în Blender inclusiv pentru Concursul NASA-Space Settlement din 2017 http://wp-space-settlement.weebly.com/

Lfo

General Purpose Programming Language

APLICAŢIA Lfo Creator



ETAPELE DE DEZVOLTARE Lfo

- Instalarea sistemului de operare Linux
- Instalare Qt Framework C++
- **Dupa instalările necesare, am realizat proiectul Lfo Creator (**extensia .pro**)**
- **Am inclus clasele default, si clasele specifice muzicii (pentru UI).**
- Am realizat interfaţa cu utilizatorul folosind butoane şi comenzi C++
- **Am realizat funcţii custom si am folosit funcţii Qt existente**
- Se compilează folosind fişierul Makefile ce include *qmake* specific Qt
- **S-au realizat doua fişiere shared library (sh) (lfointerpretator - Cpp, lfocreator - Qt)**
- Rularea aplicaţiei folosind fisierul *sh,* comanda*:* ./Lfo Creator

Proiectarea APLICAŢIEI Lfo Creator

PROGRAMARE ORIENTATĂ OBIECT (POO)

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

Am folosit POO pentru a oferi facilităţi cum sunt:

FACILITĂŢI

- **Cross-platform compatibility (Linux, Windows, etc.)**
- **UI customizabil in IDE**
- **RGB Selector → Send Colors to Elements**
- **Bine Documentat**
-

ELEMENTE DISTINCTE - Aplicaţia se distinge prin

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

PLANUL DE DEZVOLTARE este inclus in fisierul To Do de pe Github.

- Site
- Deb installer
- Implementare sunete custom
- Data Types
- Io for Data types
- **Eliminate user terminal interaction**
- **Optimize code (both raw Cpp and Qt code**
- **Deploy for MacOs**
-

Lfo

General Purpose Programming Language

FUNCŢII Qt standard sunt utilizate si sunt implementate noi funcţii C++ specific pentru Lfo Creator .
(Specifice QMediaPlayer, QTextStream.)

Am folosit funcţii şi tipuri de date Qt de ex. pentru:

Pentru Fisiere:

```
QFile data(ui->MainLineNameIn->text()+".lfop");
    if (data.open(QFile::WriteOnly | QFile::Truncate)) {
        QTextStream out(&data);
         QTextStream stream(&data);
        stream << ui->InText->toPlainText();
                data.flush();
                data.close();
        out << "Result: " << qSetFieldWidth(10) << left << 3.14 << 2.7;
    }
```

Cod preluat si adaptat./ Metoda de realizare de fisiere noi Qt

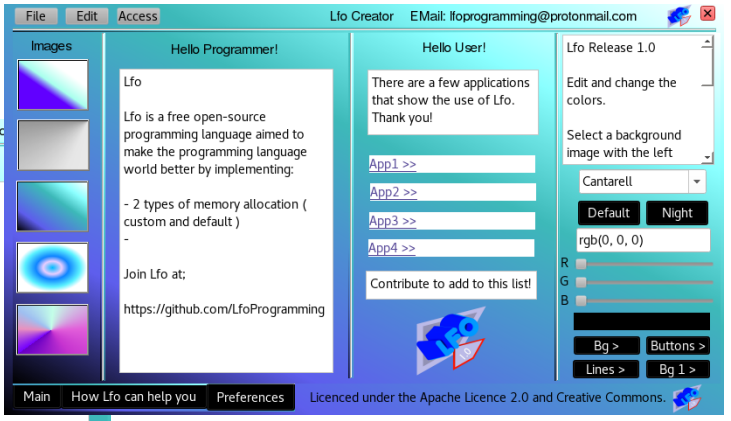
Am realizat de ex.:

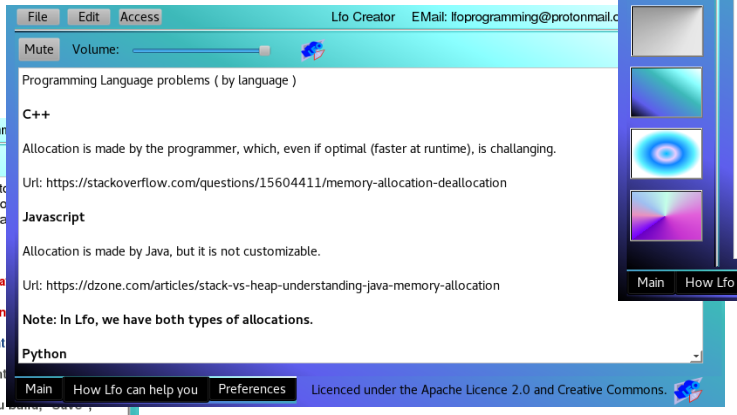
- functii de meniu (cu butoane hidden)

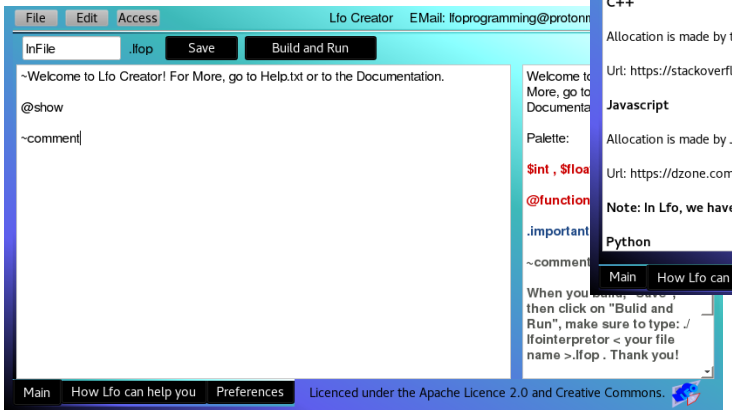
```
void MainFrame::on_DemoButton_clicked()
{
 on_HiddenClearButton_clicked();
 if (isDemo==0){
    ui->Demo->show();
    isDemo=1;
 } else {
    ui->Demo->hide();
    isDemo=0;
 }
}
```

```
ui->Site->setText("www.lfoprogramming.dx.am");
ui->SitesLabel->setText("<a style=\"color:#534496;\" href=\"http://www.lfoprogramming.dx.am/\">Go! \>\><</a>");
ui->SitesLabel->setTextFormat(Qt::RichText);
ui->SitesLabel->setTextInteractionFlags(Qt::TextBrowserInteraction);
ui→SitesLabel→setOpenExternalLinks(true);
```

Set Link to go to Site








```
#include "mainframe.h" #include "ui_mainframe.h"
#include <fstream> #include <string.h>
```

Lfo

General Purpose Programming Language

Lfo Libraries:

lfoio.h
lfomath.h

+

Functii neincluse in clase:

$void @dealoc $int . . .

Applicatii:

- Ode to Joy
- Distanta intre doua puncte

Interpretare:

Citeste fisierul si se ocupa de apelarea functiilor si alocarea de memorie

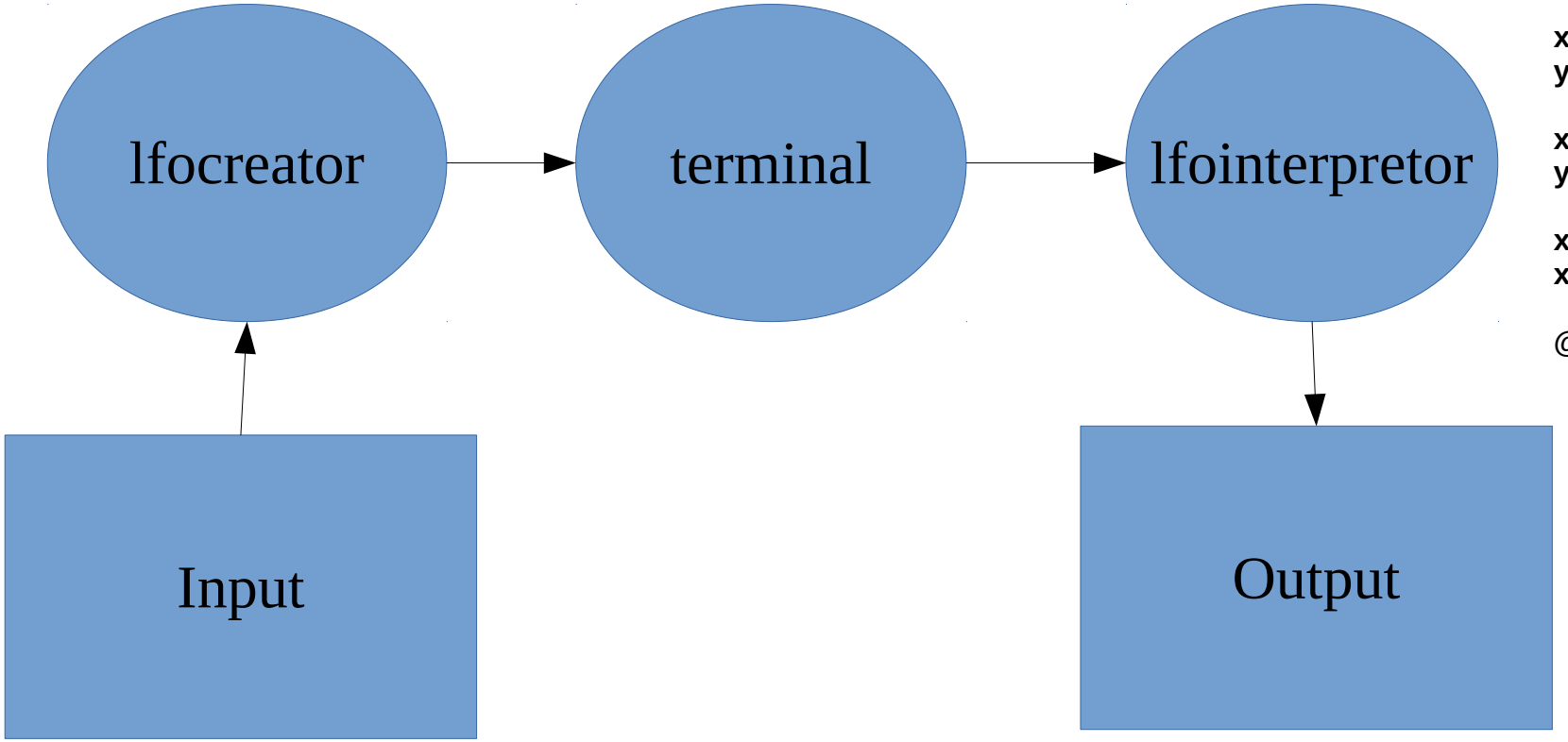
Caracteristici ale Limbajului:

Variabile si functii, biblioteci.

Elemente Metaprogramare:

Lfo foloseste sub anumite aspecte metaprogramare.

```
~distanta intre 2
~puncte

#include lfoio
#include lfomath

$int xa
$int xb
$int ya
$int yb

xa = @read
xb = @read
ya = @read
yb = @read

xa = @minus $xa $xb
ya = @minus $ya $yb

xa = @multiply $xa $xa
ya = @multiply $ya $ya

xa = @sum $xa $ya
xa = @sqrt $xa

@print $xa
```



Lfo Creator

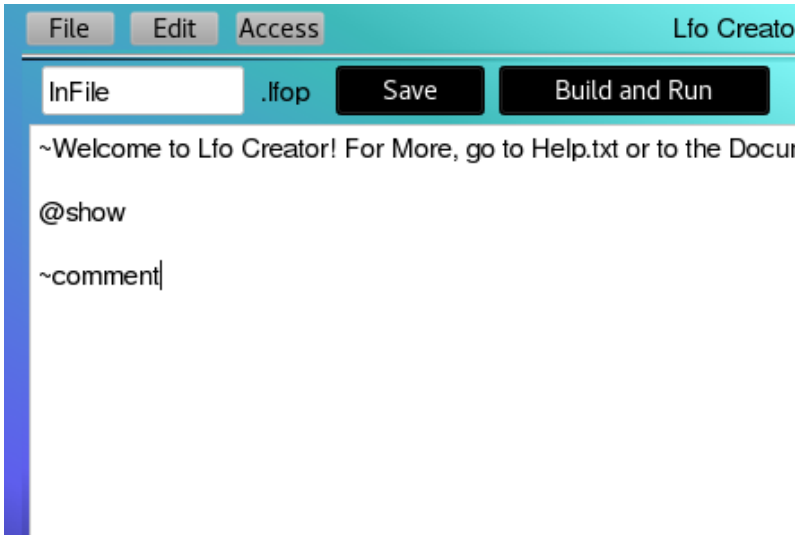
General Purpose Programming Language

Ode to Joy

```
#include lfoio
#include lfomath

$int do = 261
$int dod = 277
$int re = 293
$int red = 311
$int mi = 329
$int fa = 349
$int fad = 370
$int sol = 392
$int sold = 415
$int la = 440
$int lad = 446
$int si = 493
$int dos = 523
~notes

$int time = 400
$int timeb = 200
$int timec = 300
$int timed = 100

@beep $mi $time
@sleep $timeb
@beep $mi $time
@sleep $timeb
@beep $fa $time
@sleep $timeb
@beep $sol $time
@sleep $timeb
@beep $sol $time
@sleep $timeb
@beep $fa $time
```

```
@sleep $timeb
@beep $fa $time
@sleep $timeb
@beep $mi $time
@sleep $timeb
@beep $re $time
@sleep $timeb
@beep $do $time
@sleep $timeb
@beep $do $time
@sleep $timeb
@beep $re $time
@sleep $timeb
@beep $mi $time
@sleep $timeb
@beep $re $timec
@sleep $timec
@sleep $timec
@beep $mi $time
@sleep $timeb
@beep $mi $time
@sleep $timeb
@beep $fa $time
@sleep $timeb
@beep $sol $time
@sleep $timeb
@beep $sol $time
@sleep $timeb
@beep $fa $time
```

```
@sleep $timeb
@beep $mi $time
@sleep $timeb
@beep $re $time
@sleep $timeb
@beep $do $time
@sleep $timeb
@beep $do $time
@sleep $timeb
@beep $re $time
@sleep $timeb
@beep $mi $time
@sleep $timeb
@beep $re $time
@sleep $time
@beep $do $time
@sleep $timed
@beep $do $time
@sleep $time
```



Demo