

DESENVOLVIMENTO DE PLUGINS DE GUI PARA A FERRAMENTA MOSAICODE

DCOMP 2018 - UFSJ

Thiago Souto

Novembro de 2017

Graduando em Ciência da Computação
Universidade Federal de São João del-Rei – UFSJ
São João del-Rei/MG – Brasil

INTRODUÇÃO

Metodologia

- Explicar o Mosaiccode;
- Criação de interfaces gráficas;
- Criação de novos Blocos;

Principais funcionalidades:

- Ambiente de programação visual para criação de aplicações de DSL's;
- Gerador de código;
- Protótipos funcionais de aplicações;
- Geração, compilação e execução;
- Visualização do código gerado a qualquer etapa;

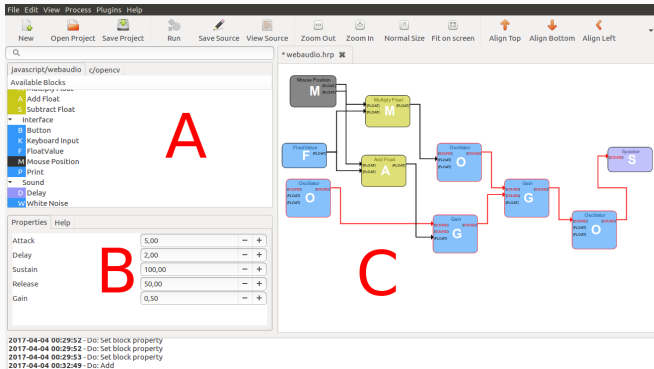


Figura 1: Ambiente de programação Visual Mosaiccode

GTK+ 3

Histórico

- Criação de interfaces gráficas;
- desenvolvido inicialmente para o GIMP;
- escrito em c, mas com orientação a objeto;

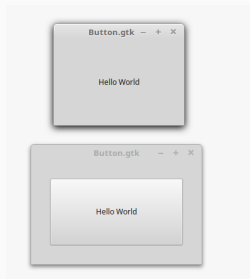


Figura 2: Exemplo de protótipo feito com o GTK: Janela com um botão que ao ser clicado abre a segunda janela

Objetivo

- Criar varios objetos nativos;
- Descobrir padrões;

```

1  + #include <gtk/gtk.h>
2  +
3  + void hello (void)
4  + {
5  +     g_print ("Hello World\n");
6  + }
7  +
8  + void destroy (void)
9  + {
10 +     gtk_main_quit ();
11 + }
12 +
13 + int main (int argc, char *argv[])
14 + {
15 +     GtkWidget *janela;
16 +     GtkWidget *botao;
17 +
18 +     gtk_init (&argc, &argv);
19 +
20 +     janela = gtk_window_new (GTK_WINDOW_TOPLEVEL);
21 +     g_signal_connect (janela, "destroy", G_CALLBACK (destroy), NULL);
22 +     gtk_container_set_border_width (GTK_CONTAINER (janela), 30);
23 +
24 +     botao = gtk_button_new_with_label ("Hello World");
25 +
26 +     g_signal_connect (botao, "clicked", G_CALLBACK (hello), NULL);
27 +     g_signal_connect_object (botao, "clicked", G_CALLBACK (gtk_widget_destroy), NULL, G_CONNECT_AFTER);
28 +     gtk_container_add (GTK_CONTAINER (janela), botao);
29 +
30 +
31 +     gtk_widget_show_all (janela);
32 +
33 +     gtk_main ();
34 +
35 +     return 0;
36 + }

```

Figura 3: Exemplo de Código em C usando o GTK+3.

- Criar vários objetos Para o domínio da Arte;
- Enteder as outras bibliotecas usadas pelos GTK;

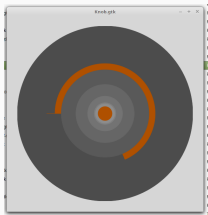


Figura 4: Elemento de GUI criado nesta pesquisa: Botão do tipo Knob rotativo

```

81 gboolean draw(GtkWidget *widget, GdkEventConfigure *event, gpointer user_data)
82 {
83
84     cairo_t *cr;
85
86     t_knob *knob = (t_knob *) user_data;
87
88     cr = gdk_cairo_create(GTK_WIDGET_GET_WINDOW(knob->drawing_area));
89     int width;
90     int height;
91
92     width = gtk_widget_get_allocated_width(widget);
93     height = gtk_widget_get_allocated_height(widget);
94
95     // Background circle
96     cairo_set_source_rgb(cr, 0, 1, 0);
97     cairo_set_line_width(cr, 1);
98     cairo_arc(cr, width / 2, height / 2.0, MIN(width, height)/2.0, 0., 2.0 * 0.01);
99     cairo_fill_preserve(cr);
100    cairo_set_source_rgb(cr, 0, 0, 0);
101    cairo_stroke(cr);
102
103    float angle1 = 0.01; // Start point
104    float angle2 = angle1 + knob->value * (2.0 * 0.01 / 120.0); // Start point
105
106    cairo_new_sub_path(cr);
107    cairo_set_source_rgb(cr, 1, 0, 0);
108    cairo_arc(cr, width / 2, height / 2.0, MIN(width, height)/3.0, angle1, angle2);
109    cairo_line_to (cr, width / 2, height / 2);
110    cairo_line_to (cr, MIN(width, height)/2.0 - MIN(width, height)/3.0 , height / 2);
111
112    cairo_fill_preserve(cr);
113    cairo_set_source_rgb(cr, 0, 0, 0);
114    cairo_stroke(cr);
115    cairo_destroy(cr);
116    return FALSE;
117
118 }

```

Figura 5: Código Botão to tipo Knob rotativo

Encapsular

- Encapsular o Código em Blocos;

```

# -*- coding: utf-8 -*-
# [MOZAICODÉ PROJECT]
#
---
This module contains the JavaScriptTemplate class.
---
from mosaicode.model.codetemplate import CodeTemplate

class CFile(CodeTemplate):
    ---
    This class contains methods related the JavaScriptTemplate class.
    ---
    # -----
    def __init__(self):
        CodeTemplate.__init__(self)
        self.name = "gtk"
        self.language = "c"
        self.description = "A full template to generate gtk code"
        self.extension = ".c"
        self.command = "g++ -o $dir_name$filename$ $dir_name$filename$extension` pkg-config --cflags --libs gtk+-3.0` \\"
        self.command += "$dir_name`/$filename"
        self.code_parts = ["function", "declaration", "configuration"]
        self.code = ""

#include <gtk/gtk.h>

void destroy(void){
    gtk_main_quit ();
}

$code{function}$

int main(int argc, char *argv[]){
    GtkWidget *window;
    GtkWidget *vbox;
    $code{declaration}$

    gtk_init(&argc, &argv);

    window = gtk_window_new(GTK_WINDOW_TOPLEVEL);
    g_signal_connect(window, "destroy", G_CALLBACK(destroy), NULL);
    gtk_window_resize(GTK_WINDOW(window), 800, 600);

    vbox = gtk_box_new(GTK_ORIENTATION_VERTICAL, 3);
    gtk_container_add(GTK_CONTAINER(window), vbox);

    $code{configuration}$
    $connections$

    gtk_widget_show_all(window);
    gtk_main();
    return 0;
}
---
# -----

```

Figura 6: Código para GTK

```

#!/usr/bin/env python
# -*- coding: utf-8 -*-
"""
This module contains the FloatValue class.
"""
from mosaiccode.GUI.fieldtypes import *
from mosaiccode.model.blockmodel import BlockModel

class Button(BlockModel):
    # -----
    def __init__(self):
        BlockModel.__init__(self)

        self.language = "c"
        self.framework = "gtk"
        self.help = "not to declare"
        self.label = "Button"
        self.color = "290:150:150:150"
        self.group = "Form"
        self.ports = []

        self.properties = [{"name": "label",
                             "label": "Label",
                             "type": MOSAICODE_STRING,
                             "value": "Label"}
                            ]

        self.codes["function"] = """
void button_sids_callback(void){
    g_print("Hello World!\n");
}
"""

        self.codes["declaration"] = """
GtkWidget *$labels_$ids;
"""

        self.codes["configuration"] = """
$labels $ids = gtk_button_new_with_label("$prop[label]$");
g_signal_connect($labels $ids, "clicked", G_CALLBACK(button_sids_callback), NULL);
gtk_container_add(GTK_CONTAINER(vbox), $labels $ids);
"""

```

Figura 7: Código Bloco de GTK

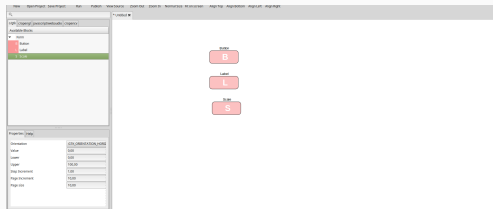


Figura 8: Mosaicode com Bloco de GTK

Trabalhos Futuros

- Ligar os Blocos OpenGL E OpenCV;
- Fazer as conexões;
- Criar outros objetos para o domínio artístico;

OBRIGADO!

Thiago Souto



Projeto Mosaicode

mosaicodedev@googlegroups.com

<https://mosaicode.github.io/>