GNOME Streamers

Daniel García Moreno

<danigm@gnome.org>

@danigm

Who is doing live streaming related to GNOME

ebassi

Gtk development, writing doc, creating new GNOME apps...

Georges Stavracas

Shell, Calendar, Builder

jimmacfx

Design, Blender, Icons, Music

Abentogil (me)

Rust gtk app development, Gtranslator maintanership...

Others?

Let me know if you know about someone else to add to the list!

https://wiki.gnome.org/Community/Streamers

The importance of the language for the community

There is a lot of content in English, but

I’m doing my streaming in **Spanish**!

Georges do some live coding in **Portuguese** (BR) too.

The importance of the language for the community

But in English you can reach more people!

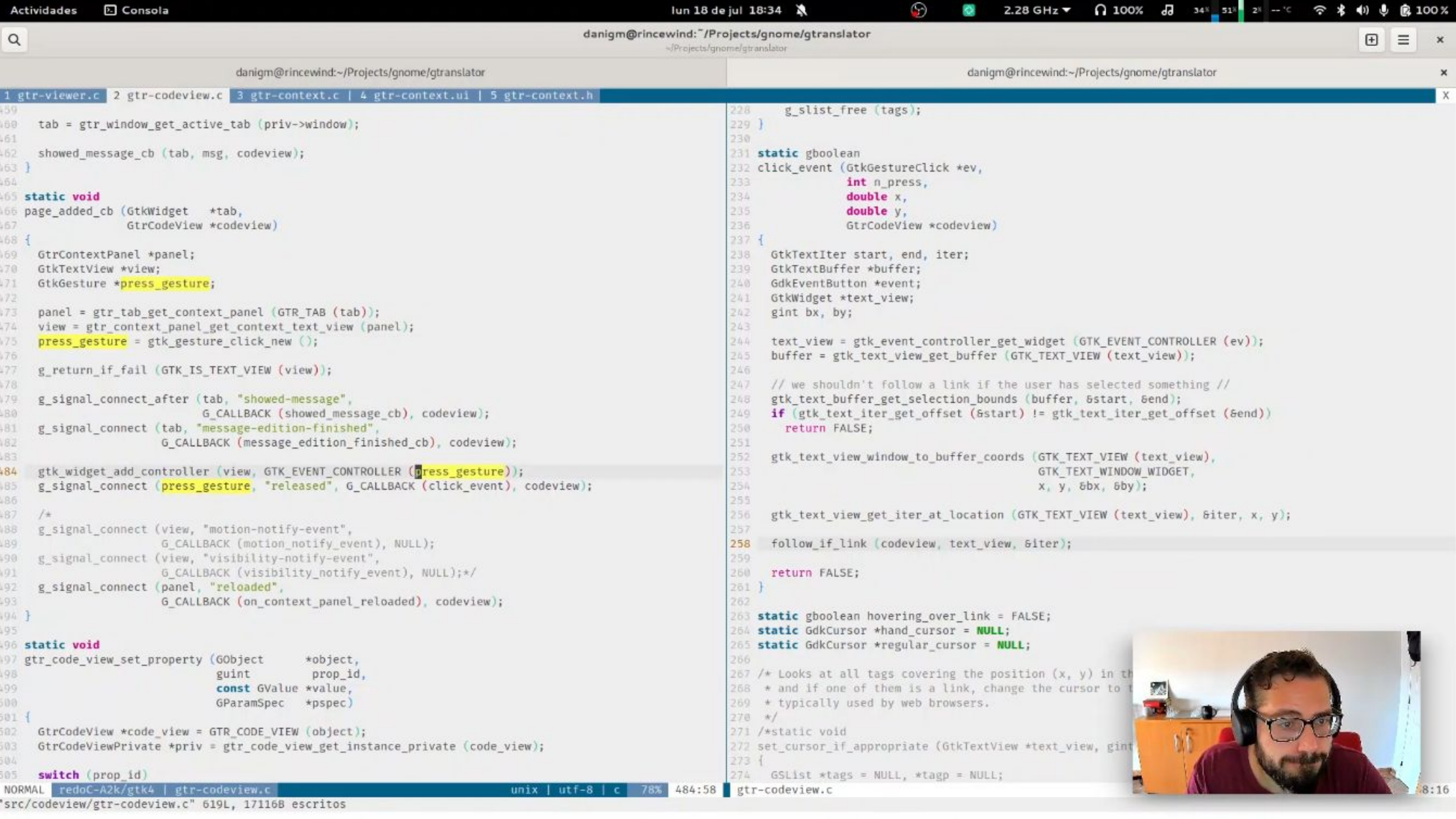
- I’m more comfortable thinking out loud in Spanish
- Not everyone knows or should know English, it’s good to reach other communities
- It’s easier to create a community, people feel more comfortable to interact in their own language

Why I want to see people writing code?

- To learn!
- A nice way to discover new things without much effort, just watching someone.
- To interact with the developer in real time

Why I’m doing live streaming?

- A way to committed myself to spent some time working on GNOME weekly
- A simple way to practice my communication skills with small audience
- To show the world how (easy) it’s the free software development
- Create some content in Spanish!


Actividades
Consola
lun 18 de jul 18:34
2.28 GHz
100%
100%
danigm@rincewind:~/Projects/gnome/gtranslator
~/Projects/gnome/gtranslator
danigm@rincewind:~/Projects/gnome/gtranslator
danigm@rincewind:~/Projects/gnome/gtranslator

```
1 gtr-viewer.c  2 gtr-codeview.c  3 gtr-context.c | 4 gtr-context.ui | 5 gtr-context.h
```

```c

  tab = gtr_window_get_active_tab (priv->window);

  showed_message_cb (tab, msg, codeview);
}

static void
page_added_cb (GtkWidget   *tab,
               GtrCodeView *codeview)
{
  GtrContextPanel *panel;
  GtkTextView *view;
  GtkGesture *press_gesture;

  panel = gtr_tab_get_context_panel (GTR_TAB (tab));
  view = gtr_context_panel_get_context_text_view (panel);
  press_gesture = gtk_gesture_click_new ();

  g_return_if_fail (GTK_IS_TEXT_VIEW (view));

  g_signal_connect_after (tab, "showed-message",
                          G_CALLBACK (showed_message_cb), codeview);
  g_signal_connect (tab, "message-edition-finished",
                    G_CALLBACK (message_edition_finished_cb), codeview);

  gtk_widget_add_controller (view, GTK_EVENT_CONTROLLER (press_gesture));
  g_signal_connect (press_gesture, "released", G_CALLBACK (click_event), codeview);

  /*
  g_signal_connect (view, "motion-notify-event",
                    G_CALLBACK (motion_notify_event), NULL);
  g_signal_connect (view, "visibility-notify-event",
                    G_CALLBACK (visibility_notify_event), NULL);*/
  g_signal_connect (panel, "reloaded",
                    G_CALLBACK (on_context_panel_reloaded), codeview);
}

static void
gtr_code_view_set_property (GObject      *object,
                            guint         prop_id,
                            const GValue *value,
                            GParamSpec   *pspec)
{
  GtrCodeView *code_view = GTR_CODE_VIEW (object);
  GtrCodeViewPrivate *priv = gtr_code_view_get_instance_private (code_view);

  switch (prop_id)
```

```
NORMAL  redoC-A2k/gtk4 | gtr-codeview.c                unix | utf-8 | c   78%   484:58
"src/codeview/gtr-codeview.c" 619L, 17116B escritos
```

```c
    g_slist_free (tags);
}

static gboolean
click_event (GtkGestureClick *ev,
             int n_press,
             double x,
             double y,
             GtrCodeView *codeview)
{
  GtkTextIter start, end, iter;
  GtkTextBuffer *buffer;
  GdkEventButton *event;
  GtkWidget *text_view;
  gint bx, by;

  text_view = gtk_event_controller_get_widget (GTK_EVENT_CONTROLLER (ev));
  buffer = gtk_text_view_get_buffer (GTK_TEXT_VIEW (text_view));

  // we shouldn't follow a link if the user has selected something //
  gtk_text_buffer_get_selection_bounds (buffer, &start, &end);
  if (gtk_text_iter_get_offset (&start) != gtk_text_iter_get_offset (&end))
    return FALSE;

  gtk_text_view_window_to_buffer_coords (GTK_TEXT_VIEW (text_view),
                                         GTK_TEXT_WINDOW_WIDGET,
                                         x, y, &bx, &by);

  gtk_text_view_get_iter_at_location (GTK_TEXT_VIEW (text_view), &iter, x, y);

  follow_if_link (codeview, text_view, &iter);

  return FALSE;
}

static gboolean hovering_over_link = FALSE;
static GdkCursor *hand_cursor = NULL;
static GdkCursor *regular_cursor = NULL;

/* Looks at all tags covering the position (x, y) in th
 * and if one of them is a link, change the cursor to t
 * typically used by web browsers.
 */
/*static void
set_cursor_if_appropriate (GtkTextView *text_view, gint
{
  GSList *tags = NULL, *tagp = NULL;
```

```
gtr-codeview.c                                                                  8:16
```

What I’m doing

- Some GNOME Translation Editor maintenance
- A chess application with Rust and Gtk4: gambito
- A music application with python and Gtk4: loop
- Some "tutorial", gnome-extensions, etc

What I’ve learned

- It’s really easy to start streaming with cheap hardware
- The mic is important, good sound is the real difference (my mic is not really good)
- Sometimes is hard to write code in live streaming
- It’s fun and helps me to do something every week

Thank you!

@danigm
<danigm@gnome.org>
https://danigm.net
https://twitch.tv/abentogil