GTK 3 ⇨ GTK 4

Matthias Clasen
GUADEC 2020

Principles

Delegation over subclassing

Delegation over subclassing

Everything is a widget

Delegation over subclassing

Everything is a widget

Widgets over containers

Delegation over subclassing

Everything is a widget

Widgets over containers

Popovers over menus

Delegation over subclassing

Everything is a widget

Widgets over containers

Popovers over menus

X11 ⇨ Wayland

Surprises

Widgets are visible by default

Widgets are visible by default

gtk_container_add() is gone

Widgets are visible by default

gtk_container_add() is gone

gtk4-builder-tool can help

```
gtk4-builder-tool simplify --3to4 old.ui > new.ui
```

Drawing

Layout

Input

Actions

Widgets → Render nodes → OpenGL / Vulkan

Drawing

```
gboolean (* draw) (GtkWidget *widget,
                   cairo_t   *cr)
```

⇩

```
void (* snapshot) (GtkWidget   *widget,
                   GtkSnapshot *snapshot)
```

```
gtk_snapshot_append_texture()
gtk_snapshot_append_layout()
gtk_snapshot_append_cairo()
```

GtkDrawingArea

gtk_drawing_area_set_draw_func()

GtkDrawingArea

gtk_drawing_area_set_draw_func()

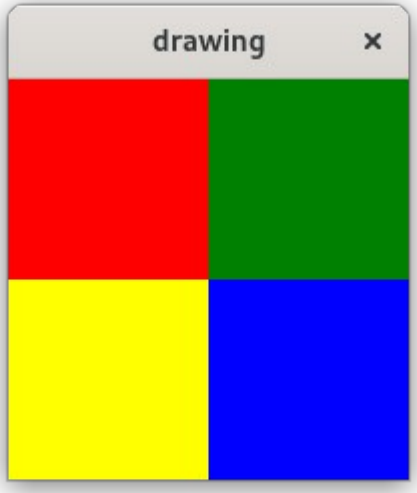
GtkGLArea

Demo

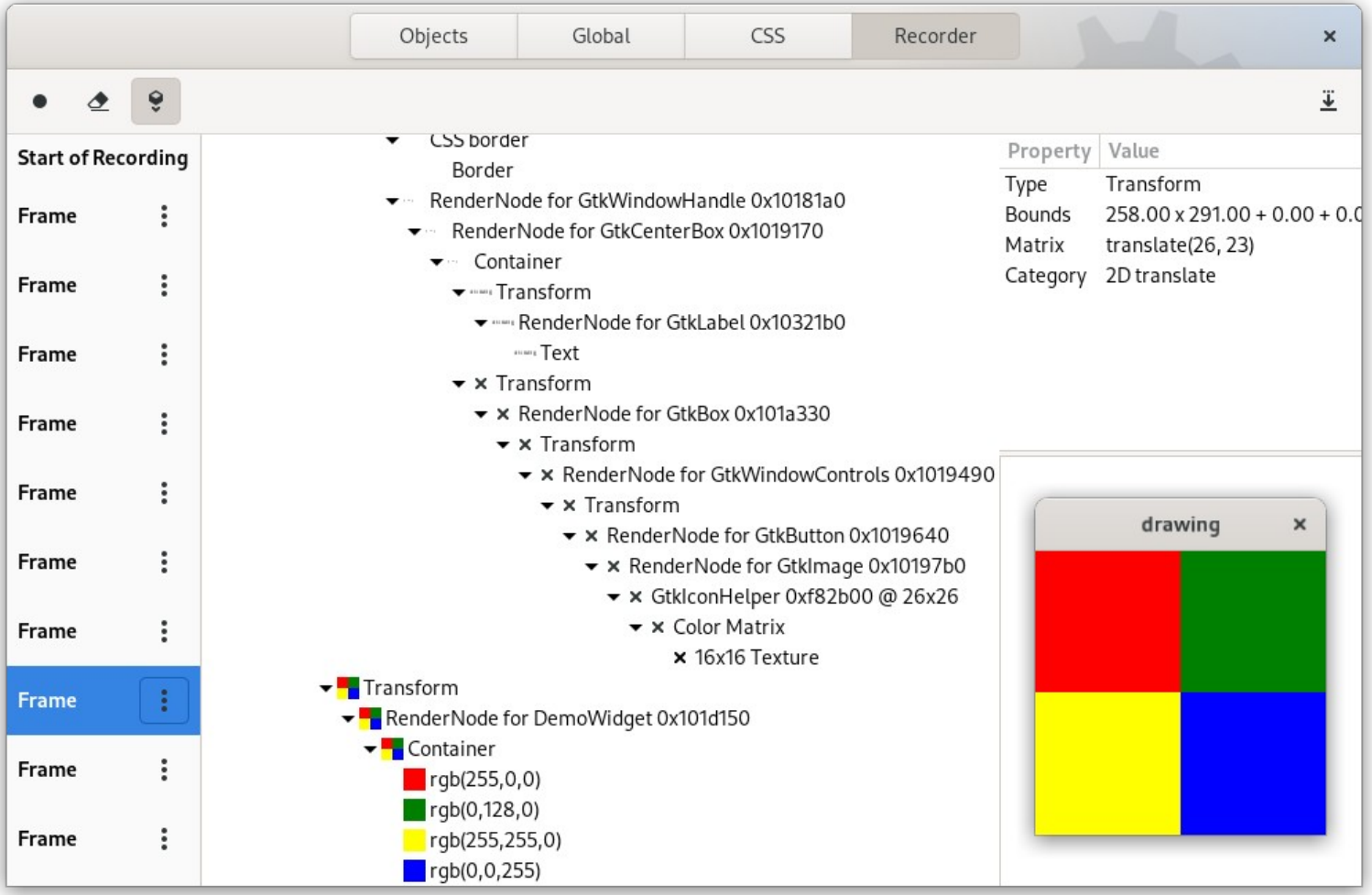
Drawing

```
static void
demo_widget_snapshot (GtkWidget   *widget,
                      GtkSnapshot *snapshot)
{
  GdkRGBA red, green, yellow, blue;
  float w, h;

  gdk_rgba_parse (&red, "red");
  gdk_rgba_parse (&green, "green");
  gdk_rgba_parse (&yellow, "yellow");
  gdk_rgba_parse (&blue, "blue");

  w = gtk_widget_get_width (widget) / 2.0;
  h = gtk_widget_get_height (widget) / 2.0;

  gtk_snapshot_append_color (snapshot, &red,&GRAPHENE_RECT_INIT(0, 0, w, h));
  gtk_snapshot_append_color (snapshot, &green, &GRAPHENE_RECT_INIT(w, 0, w, h));
  gtk_snapshot_append_color (snapshot, &yellow, &GRAPHENE_RECT_INIT(0, h, w, h));
  gtk_snapshot_append_color (snapshot, &blue, &GRAPHENE_RECT_INIT(w, h, w, h));
}
```

Drawing



Drawing


Objects
Global
CSS
Recorder
Start of Recording
Frame
Frame
Frame
Frame
Frame
Frame
Frame
Frame
Frame
Frame
CSS border
Border
RenderNode for GtkWindowHandle 0x10181a0
RenderNode for GtkCenterBox 0x1019170
Container
Transform
RenderNode for GtkLabel 0x10321b0
Text
Transform
RenderNode for GtkBox 0x101a330
Transform
RenderNode for GtkWindowControls 0x1019490
Transform
RenderNode for GtkButton 0x1019640
RenderNode for GtkImage 0x10197b0
GtkIconHelper 0xf82b00 @ 26x26
Color Matrix
16x16 Texture
Transform
RenderNode for DemoWidget 0x101d150
Container
rgb(255,0,0)
rgb(0,128,0)
rgb(255,255,0)
rgb(0,0,255)
Property
Value
Type
Transform
Bounds
258.00 x 291.00 + 0.00 + 0.0
Matrix
translate(26, 23)
Category
2D translate
drawing

Say goodbye to GtkContainer

size_allocate ⇨ GtkLayoutManager

size_allocate ⇨ GtkLayoutManager

gtk_widget_class_set_layout_manager_type()

size_allocate ⇨ GtkLayoutManager

gtk_widget_class_set_layout_manager_type()

gtk_widget_set_parent()

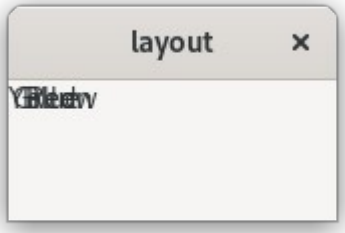
Demo

Layout

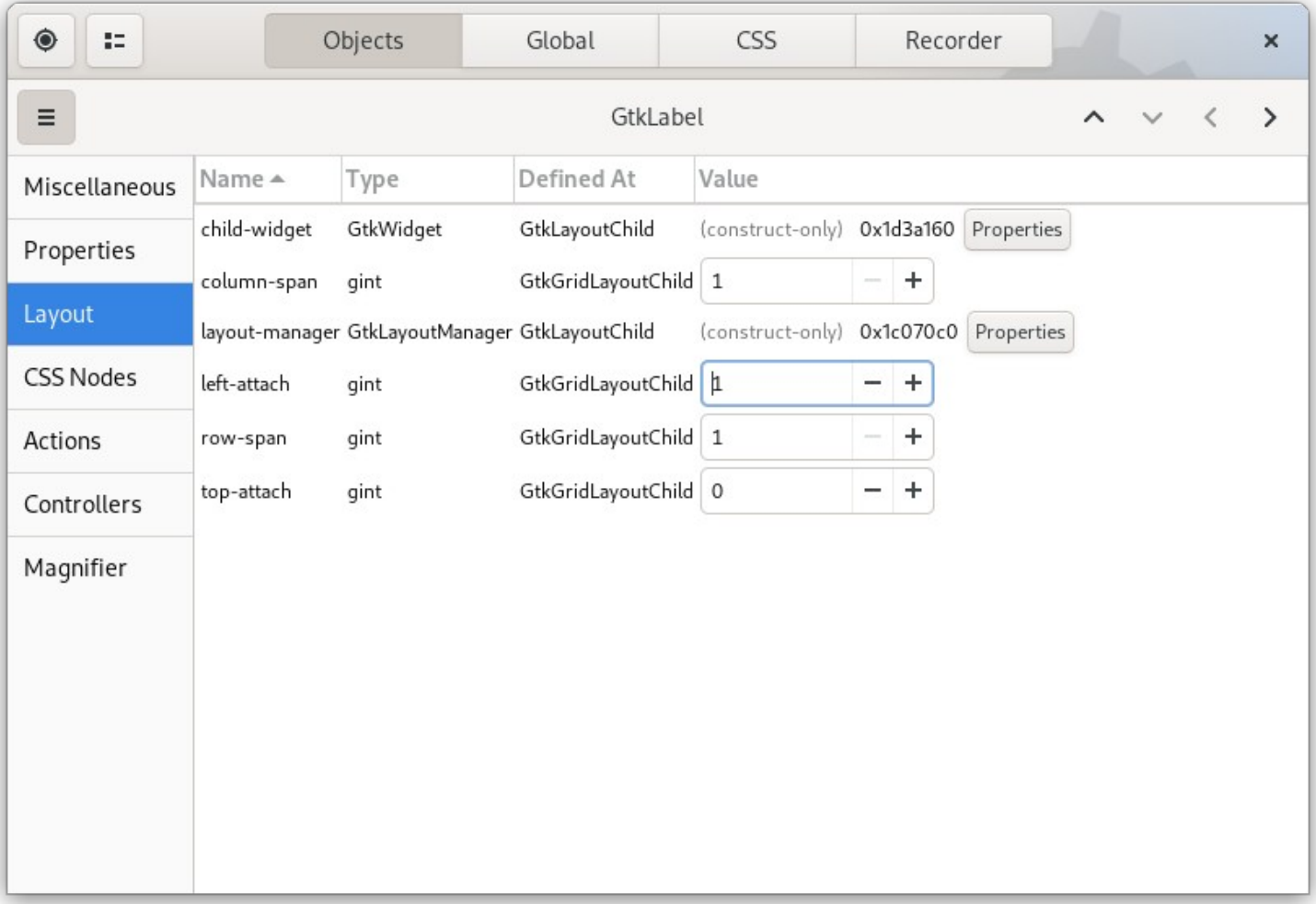
```
static void
demo_widget_init (DemoWidget *demo)
{
  demo->label1 = gtk_label_new ("Red");
  gtk_widget_set_parent (demo->label1, GTK_WIDGET (demo));

  demo->label2 = gtk_label_new ("Green");
  gtk_widget_set_parent (demo->label2, GTK_WIDGET (demo));

  demo->label3 = gtk_label_new ("Blue");
  gtk_widget_set_parent (demo->label3, GTK_WIDGET (demo));

  demo->label4 = gtk_label_new ("Yellow");
  gtk_widget_set_parent (demo->label4, GTK_WIDGET (demo));
}
```

```
static void
demo_widget_dispose (GObject *object)
{
  DemoWidget *demo = DEMO_WIDGET (object);

  g_clear_pointer (&demo->label1, gtk_widget_unparent);
  g_clear_pointer (&demo->label2, gtk_widget_unparent);
  g_clear_pointer (&demo->label3, gtk_widget_unparent);
  g_clear_pointer (&demo->label4, gtk_widget_unparent);

  G_OBJECT_CLASS (demo_widget_parent_class)->dispose (object);
}
```

Layout



Layout


Objects
Global
CSS
Recorder
GtkLabel
Miscellaneous
Properties
Layout
CSS Nodes
Actions
Controllers
Magnifier
Name
Type
Defined At
Value
child-widget GtkWidget GtkLayoutChild (construct-only) 0x1d3a160 Properties
column-span gint GtkGridLayoutChild 1
layout-manager GtkLayoutManager GtkLayoutChild (construct-only) 0x1c070c0 Properties
left-attach gint GtkGridLayoutChild 1
row-span gint GtkGridLayoutChild 1
top-attach gint GtkGridLayoutChild 0

Layout

::button-press-event ⇨ GtkGestureClick

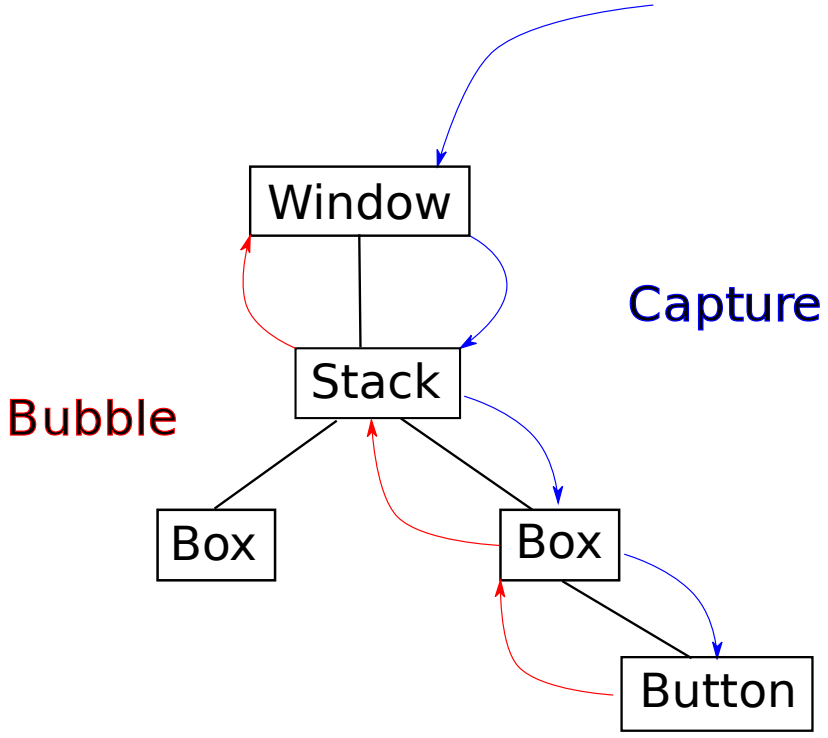
::focus-in-event ⇨ GtkEventControllerFocus

::motion-notify-event ⇨ GtkEventControllerMotion

gtk_drag_source_set() ⇨ GtkDragSource

gtk_drag_dest_set() ⇨ GtkDropTarget

Input



Demo

Input

```
static void
demo_widget_init (DemoWidget *demo)
{
  GtkEventController *controller;

  demo->label1 = gtk_label_new ("Red");
  demo->label2 = gtk_label_new ("Blue");

  controller = gtk_gesture_click_new ();
  g_signal_connect (controller, "pressed",
                    G_CALLBACK (red_clicked_cb), demo);
  gtk_widget_add_controller (demo->label1, controller);

  controller = gtk_gesture_click_new ();
  g_signal_connect (controller, "pressed",
                    G_CALLBACK (blue_clicked_cb), demo);
  gtk_widget_add_controller (demo->label2, controller);
}
```

Input

```
static void
red_clicked_cb (GtkGestureClick *gesture,
                int              n_pressed,
                double           x,
                double           y,
                DemoWidget      *demo)
{
  g_print ("Red clicked\n");
}
```

GAction, app menus

GAction, app menus

gtk_widget_insert_action_group()

GAction, app menus

gtk_widget_insert_action_group()

gtk_widget_class_install_action()

GAction, app menus

gtk_widget_insert_action_group()

gtk_widget_class_install_action()

gtk_widget_class_add_binding_action()

Demo

Actions

```
static void
demo_widget_class_init (DemoWidgetClass *class)
{
  GtkWidgetClass *widget_class = GTK_WIDGET_CLASS
(class);

  /* ... */

  gtk_widget_class_install_action (widget_class,
                                   "turn-red", NULL,
                                   turn_red);
  gtk_widget_class_install_action (widget_class,
                                   "turn-blue", NULL,
                                   turn_blue);
}
```

Actions

```
static void
turn_red (GtkWidget  *widget,
          const char *actionname,
          GVariant   *parameter)
{
  gtk_widget_add_css_class (widget, "red");
}

static void
red_clicked_cb (GtkGestureClick *gesture,
                int              n_pressed,
                double           x,
                double           y,
                DemoWidget      *demo)
{
  gtk_widget_activate_action (demo, "turn-red", NULL);
}
```

Migration guide	`https://developer.gnome.org/gtk4/3.98/migrating.html`
GTK blog	`https://blog.gtk.org/`
Discourse	`https://discourse.gnome.org`
This talk	`https://gitlab.gnome.org/matthiasc/guadec-2020`

Media support

GdkPixbuf ⇨ GdkPaintable

GtkMediaFile, GtkPicture, GtkVideo

New things

Scalable lists

Widget recycling

GtkTreeModel ⇨ GListModel

GtkIconView ⇨ GtkGridView

GtkTreeView ⇨ GtkColumnView

GTK 3 ⇨ GTK 4