

Xlivebg

Live Wallpapers for the X-window System

Γιάννης Τσιομπίκας

nuclear@member.fsf.org

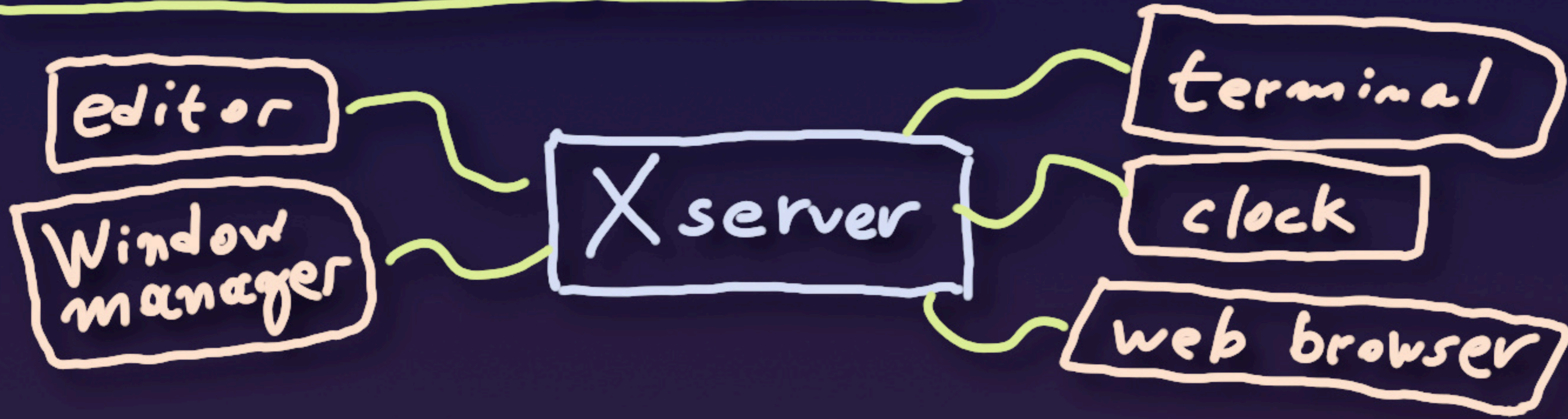
Live Wallpapers ?

- Animated wallpapers στο desktop
αντι για στατικές εικόνες

- Εμπνευσον από το android



X-Window System



- Ο window manager είναι άλλο ένα client χωρίς ειδική μεταχείριση.

- WM conventions :

- ICCCM
- Motif WM hints
- NetWM protocols

Inter-Client
Communication
Conventions
Manual

Animation στο desktop?

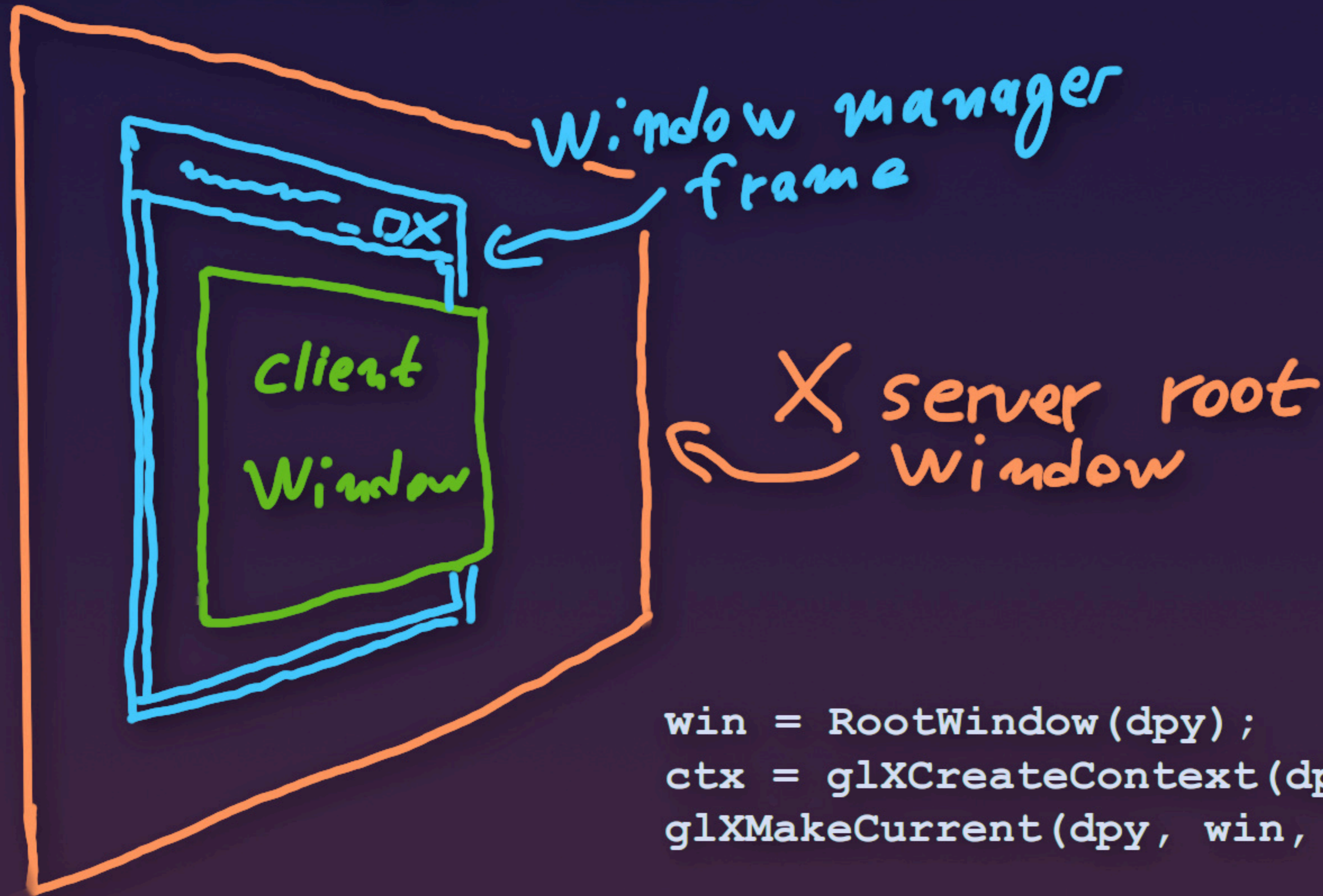
- X client που ζωγραφίζει συνεχώς στο desktop
- OpenGL context συνδεδεμένο με ?
το desktop ??

...

Τι είναι όμως αυτό το desktop?

Αναζητώντας το desktop...

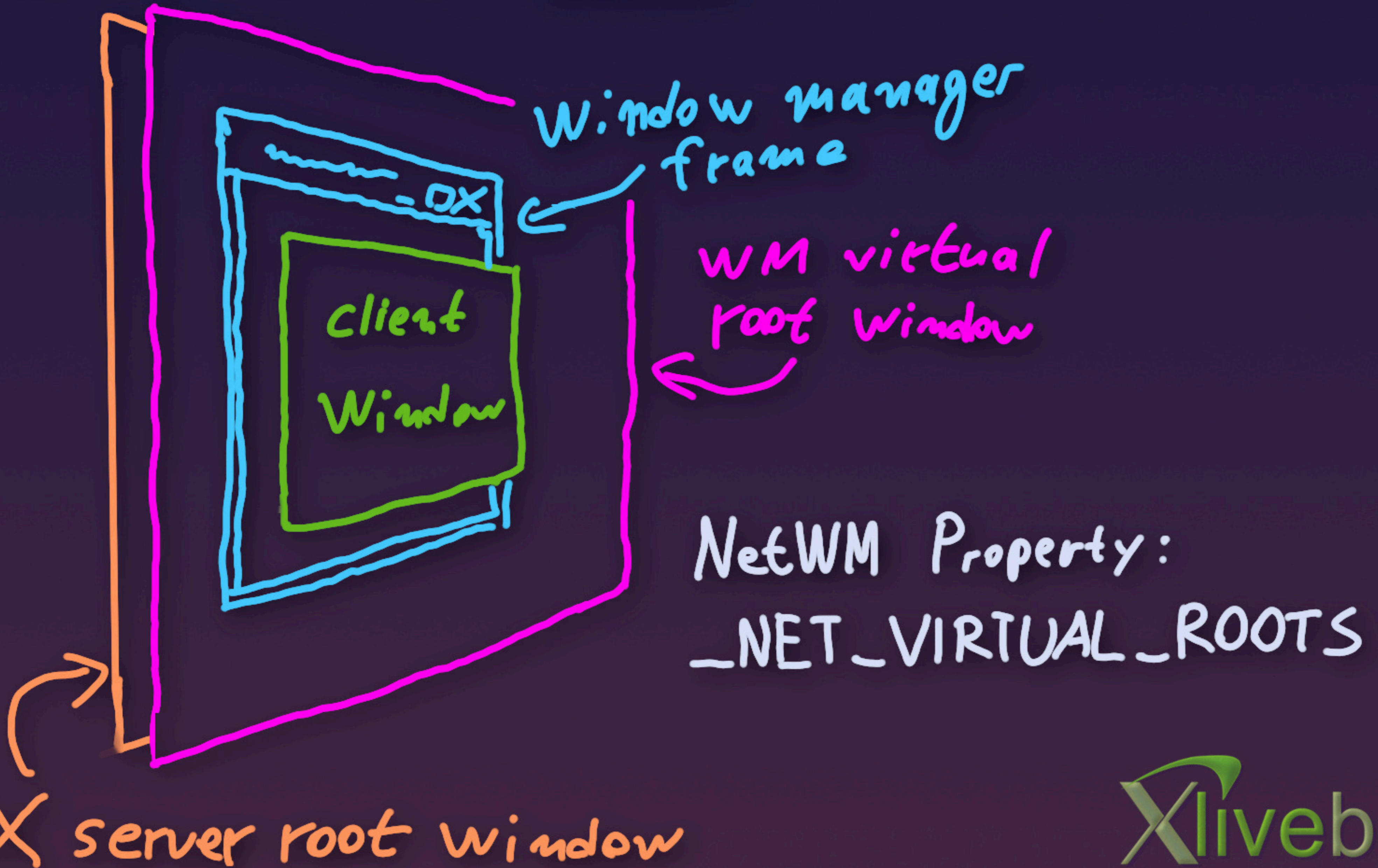
Πρώτη απόπειρα: Root window



```
win = RootWindow(dpy);  
ctx = glXCreateContext(dpy, ...);  
glXMakeCurrent(dpy, win, ctx);
```

Αναζητώντας το desktop.....

Δεύτερη απόπειρα: Virtual Root Window



Αναζητώντας το desktop.....



~~Dennis Ritchie~~

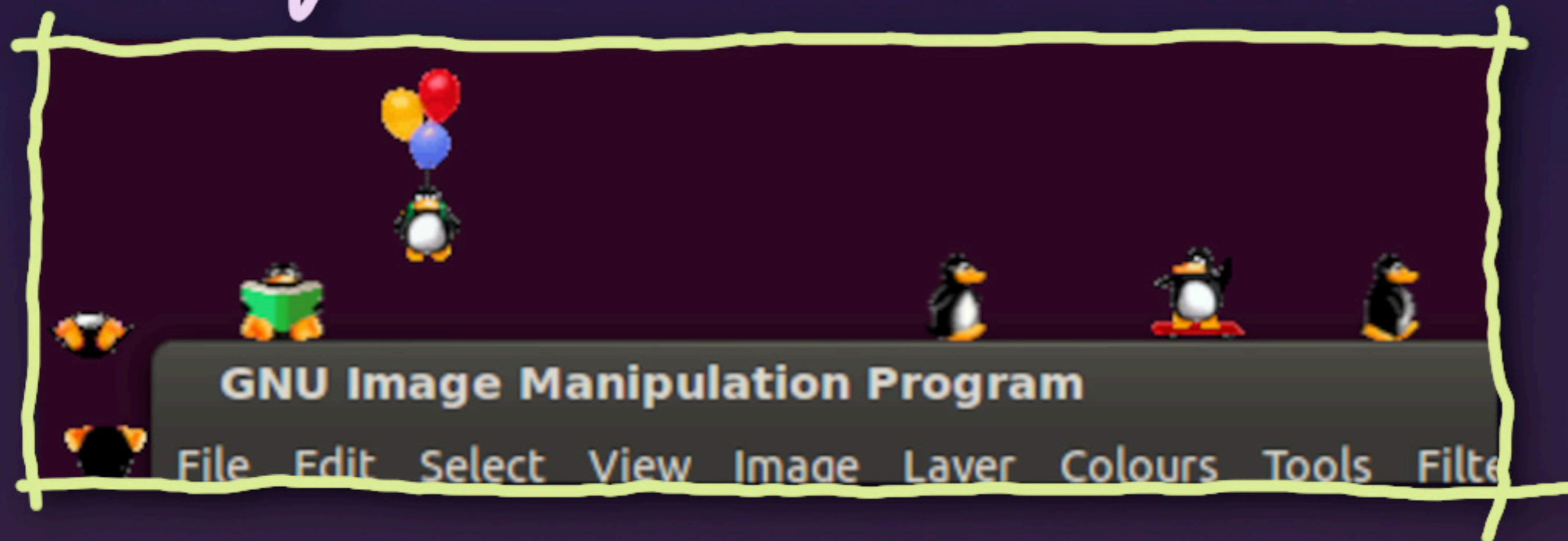
Window manager

NetWM Property:

~~_NET_VIRTUAL_ROOTS~~

Αναζητώντας το desktop

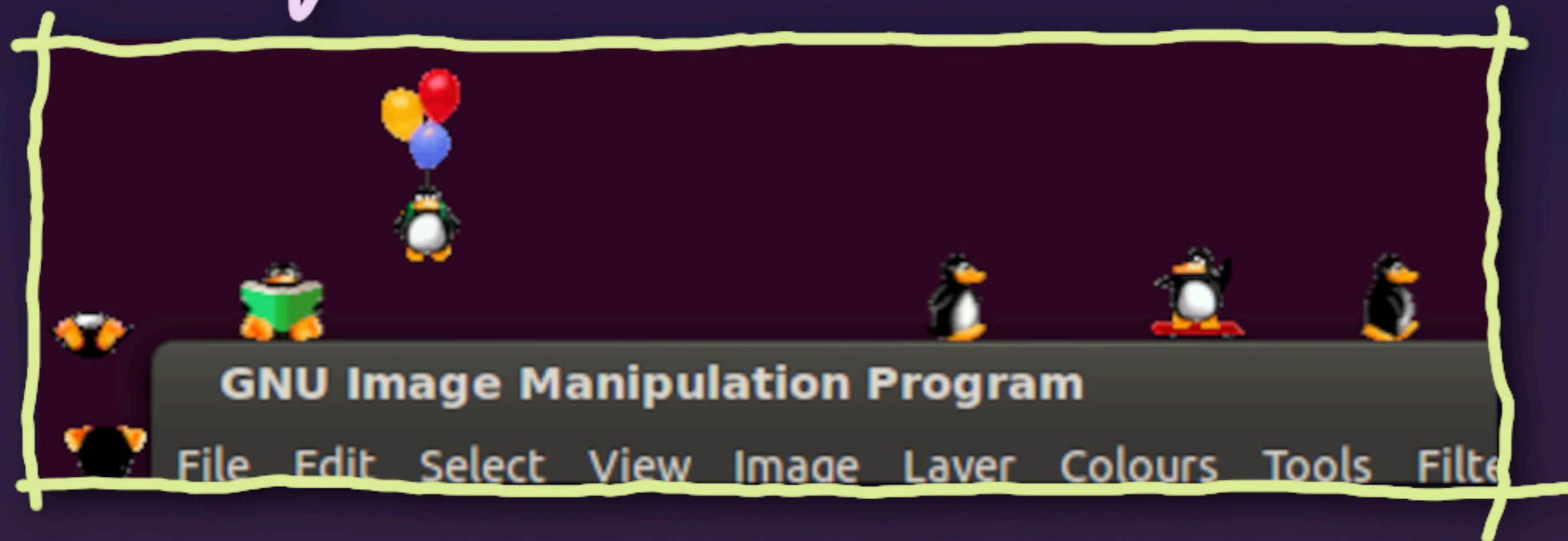
Τρίτη απόπειρα: give up...
χ penquius already did this



```
win = ToonGetRootWindow(dpy, ...);
```


Αναζητώντας το desktop

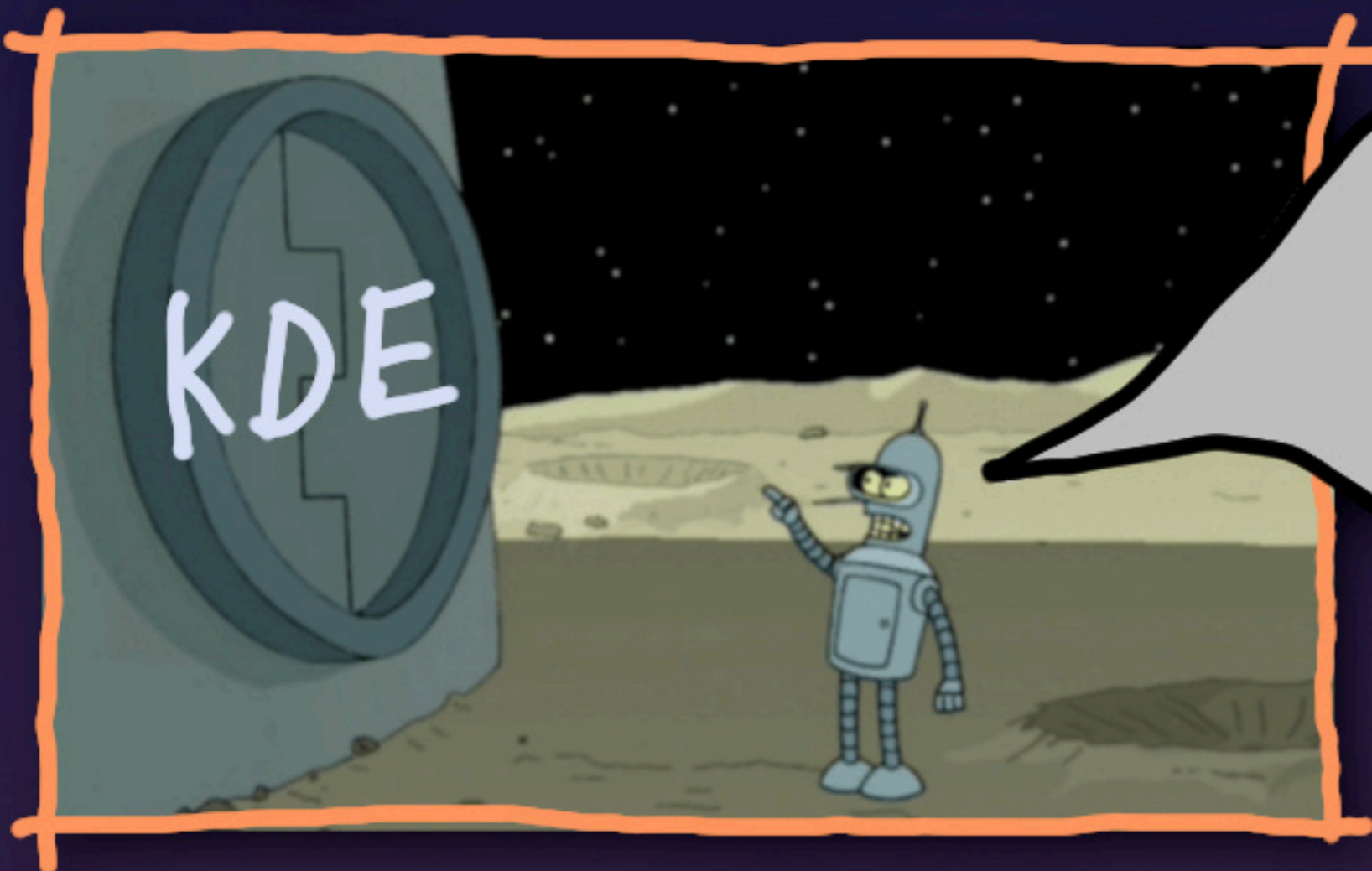
Τρίτη απόπειρα: give up...
χ penquius already did this



`win = Toon GetRoot Window (dpy, ...);`

δυστυχώς still not enough...

~~Arbeitsaktivität~~ to desktop



I'll make my own desktop with blackjack and hookers

\$ xlivebg -w

- Create new window
- Set the property: `_NET_WM_WINDOW_TYPE`
`_NET_WM_WINDOW_TYPE_DESKTOP`

Xlivebg project

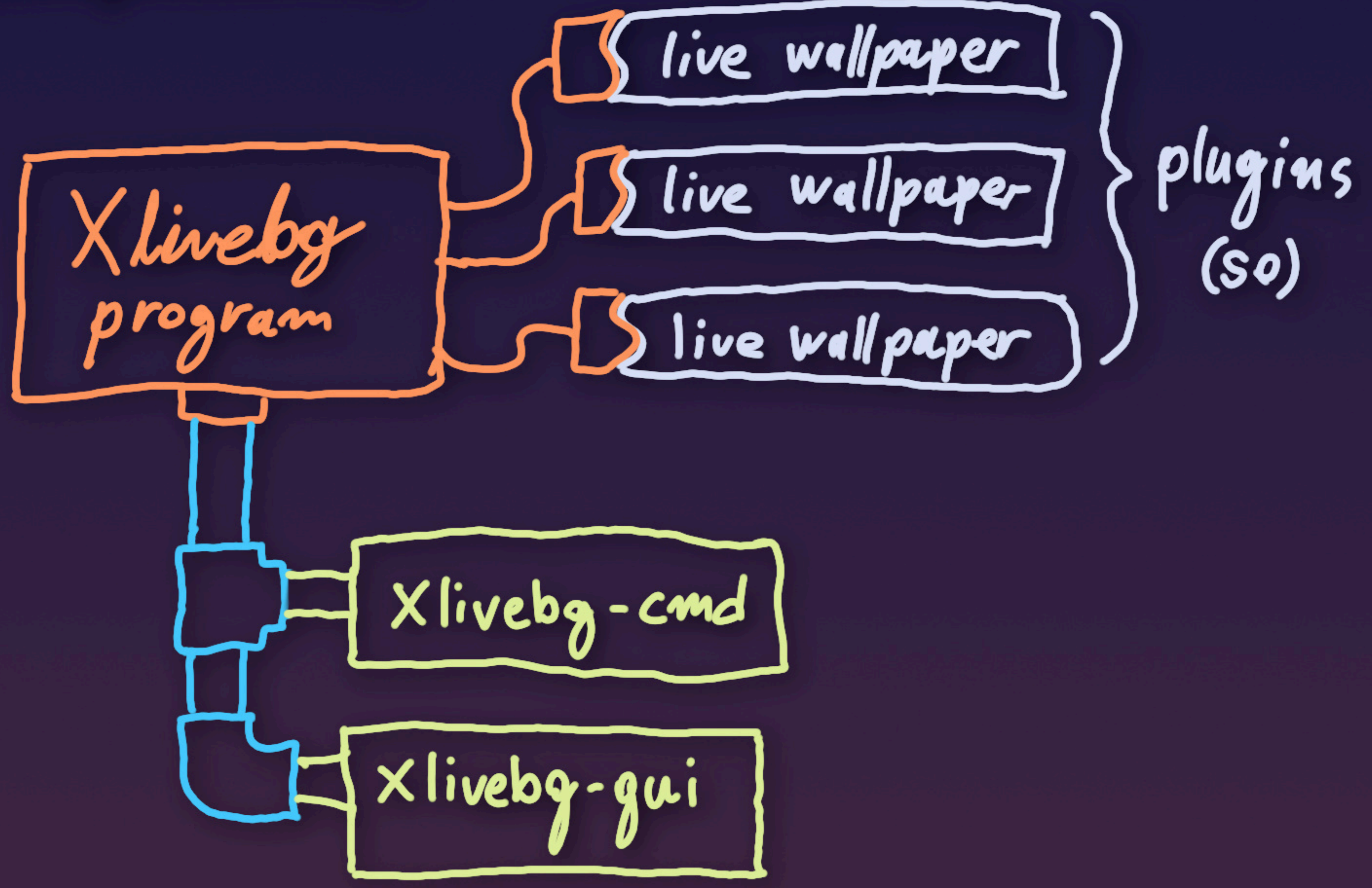
σαν Xscreensaver
για live
wallpapers

Στόχος: framework για 3rd party
live wallpapers

↳ πρέπει να απλοποιεί την υλοποίηση live
wallpapers αναλαμβάνοντας:

- όλες τις αλληλεπιδράσεις με το X window system
- το initialization του OpenGL context
- παραμετροποίηση και επιλογές
- παροχή helpers για συνήθεις ενέργειες
- φόρτωση και διαχείριση εικόνων

Xlivebg parts



Configuration files

~/xlivebg/config

~/config/xlivebg.conf

/etc/xlivebg.conf

} Search paths

```
xlivebg {  
  active = "ripple"  
  
  # background image  
  image = "bgimage.jpg"  
  fit = "stretch"
```

} global settings

```
# --- plugin-specific configuration ---
```

```
distort {  
  amplitude = 0.025  
  frequency = 8.0  
}
```

} ρυθμισεις για wallpaper
to distort

```
ripple {  
  raindrops = 0  
}
```

} ρυθμισεις για wallpaper
to ripple

```
}
```

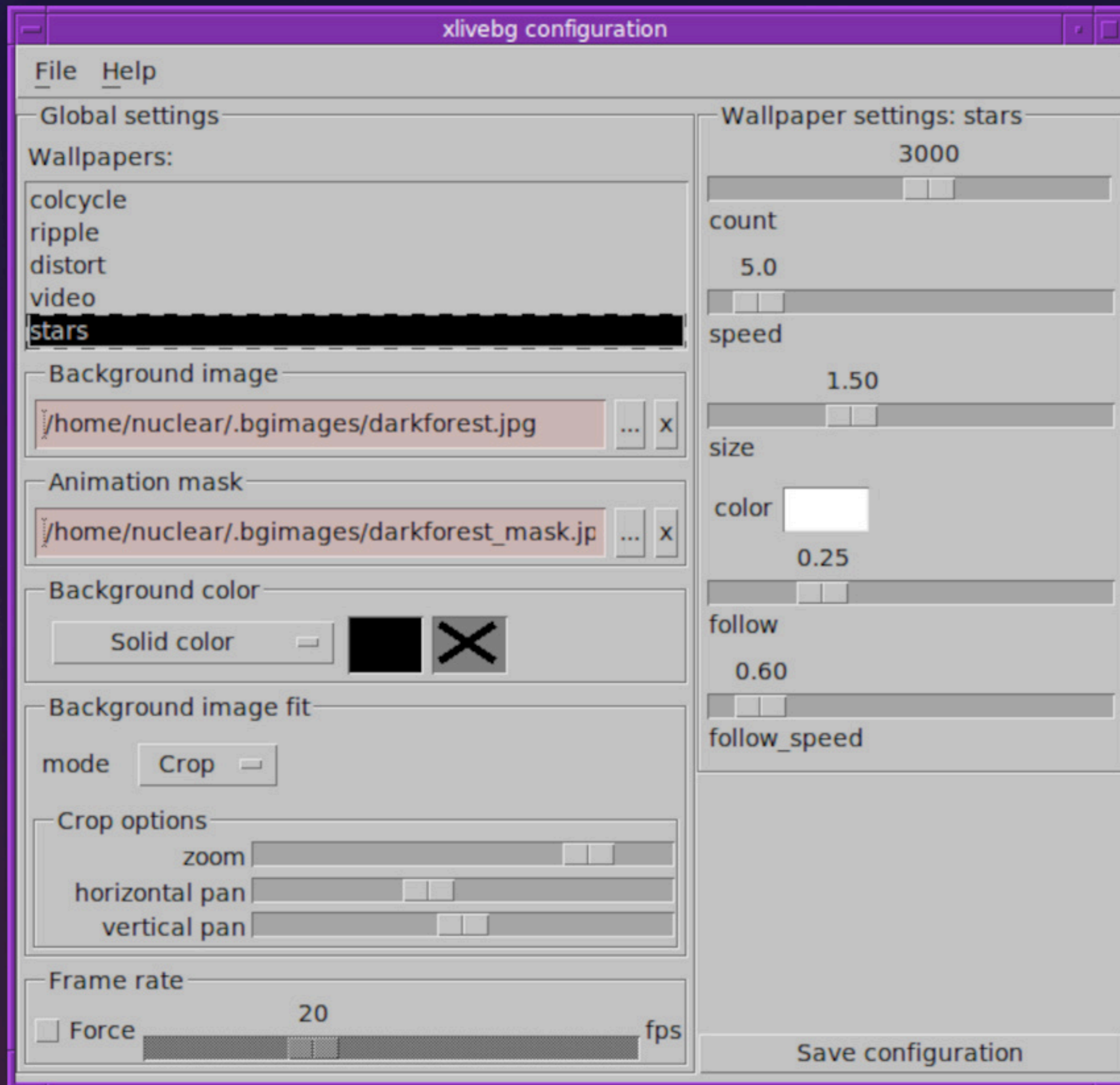
Interactive Configuration

- AF_UNIX socket: `/tmp/xlivebg.sock`
- command-line client: `xlivebg-cmd`
- plugin property description

```
proplist {  
  prop {  
    id = "count"  
    desc = "number of foo"  
    type = "integer"  
    range = [500, 5000]  
  }  
  prop {  
    id = "size"  
    desc = "foo size"  
    type = "number"  
    range = [0.25, 4.0]  
  }  
  prop {  
    id = "video"  
    desc = "background video file"  
    type = "filename"  
  }  
}
```

Κάθε wallpaper plugin
παρέχει λίστα με
τις παραμέτρους που
καταλαμβάνει

Interactive Configuration GUI



- Επικοινωνία μέσω του socket
- Γενικές ρυθμίσεις αριστ.
- Αυτόματο UI για properties του ενεργού plugin δεξιά

← αποθήκευση στο configuration file

global settings ↑

plugin properties ↑

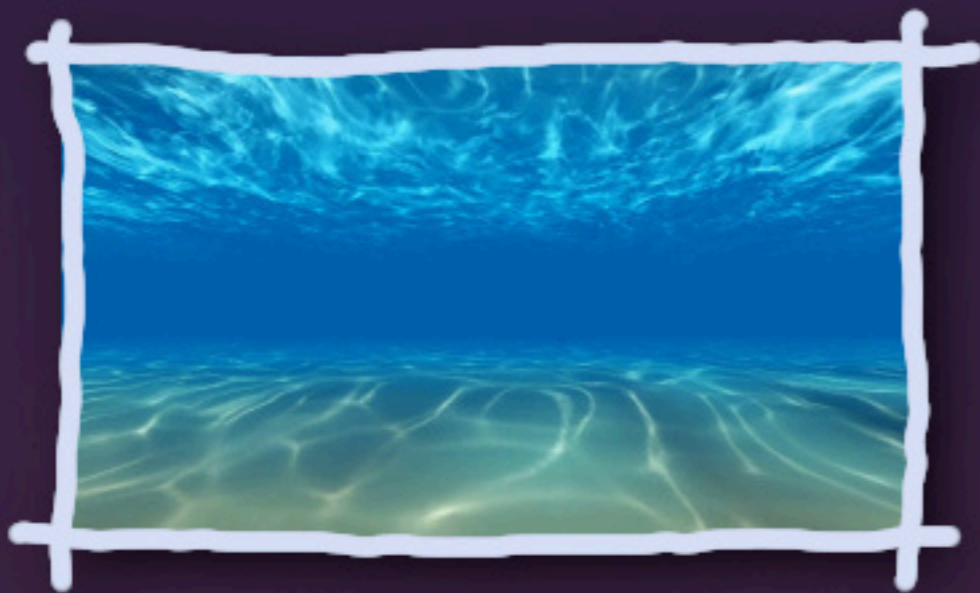
Bundled live wallpapers (v1.0)

Color cycling



ripple effect

starfield



distortion

video playback



Παράδειγμα plugin (1/2)

```
#define PROPLIST \
    "proplist {\n" \
    "    prop {\n" \
    "        id = \"speed\"\n" \
    "        desc = \"animation speed\"\n" \
    "        type = \"number\"\n" \
    "        range = [0, 10]\n" \
    "    }\n" \
    "}\n"
```

prop. list

```
static struct xlivebg_plugin plugin = {
    "minimal",
    "Minimal live wallpaper example",
    PROPLIST,
    XLIVEBG_20FPS,
    init, 0,
    start, 0,
    draw,
    prop,
    0, 0
};

static float speed;

int register_plugin(void)
{
    return xlivebg_register_plugin(&plugin);
}
```

callbacks

requested frame rate

```
struct xlivebg_plugin {
    char *name, *desc;
    char *props;
    long upd_interval;
    xlivebg_init_func init;
    xlivebg_cleanup_func cleanup;
    xlivebg_start_func start;
    xlivebg_stop_func stop;
    xlivebg_draw_func draw;
    xlivebg_prop_func prop;
    void *data, *so;
};
```

στο xlivebg.h

plugin init

```
so = dlopen("foo.so", ..);
reg = dlsym(so, "register_plugin");
reg();
```



Παράδειγμα plugin (2/2)

Callbacks

```
static int init(void *cls)
{
    xlivebg_defcfg_num("xlivebg.minimal.speed", 1.0f);
    return 0;
}

static void start(long tmsec, void *cls)
{
    prop("speed", 0);
}

static void prop(const char *prop, void *cls)
{
    if(strcmp(prop, "speed") == 0) {
        speed = xlivebg_getcfg_num("xlivebg.minimal.speed", 1.0f);
    }
}

static void draw(long tmsec, void *cls)
{
    int i, num_scr = xlivebg_screen_count();

    xlivebg_clear(GL_COLOR_BUFFER_BIT);

    /* for every screen ... */
    for(i=0; i<num_scr; i++) {
        xlivebg_gl_viewport(i);
        /* ... draw using OpenGL ... */
    }
}
```

← default τιμή αν δεν έχει ρυθμιστεί

← καλείται όταν ενεργοποιηθεί το plugin

← καλείται όταν αλλάζει κάποιο property

← καλείται για κάθε frame

Για τα ΠΕΡΑΛΤΕΡΩ...

- web site:

<http://nuclear.mutantstargod.com/sw/xlivebg>

- github repo:

<https://github.com/jtsiomb/xlivebg>

- setup & demo video:

https://www.youtube.com/watch?v=JZ_RX0BWPD8

