

sauvage

sauvage is a [Python](#) library that displays [SVG](#) graphics using [OpenGL](#), and is used to implement demanding interaction techniques, such as [ZUI](#), or [see-through tools](#). It is available on unix platforms (Linux, MacOSX) and should work on Windows, under the LGPL license.

sauvage is build upon [svg!](#) concepts (a previous library I wrote in c++), but does not use code from it. Instead, sauvage is written in python, as it allows easier modification when I have new ideas on how to implement things (no compilation, no static typing when I don't need it, etc.). The goal is to design a library *for* OpenGL and have fast and high quality zooming features.

features: all SVG shapes, styling, <use>, high quality autoscaled text, antialiasing, picking, SVG loading, images etc.

API example

see at the end of this page the code for the first sceenshot.

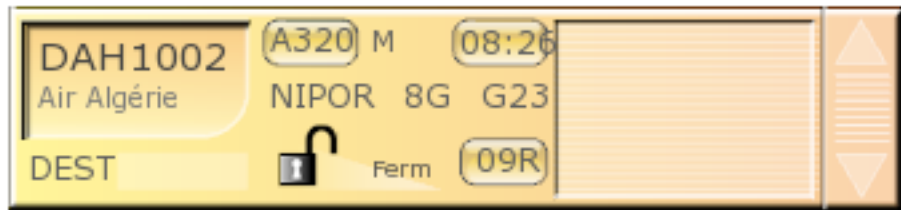
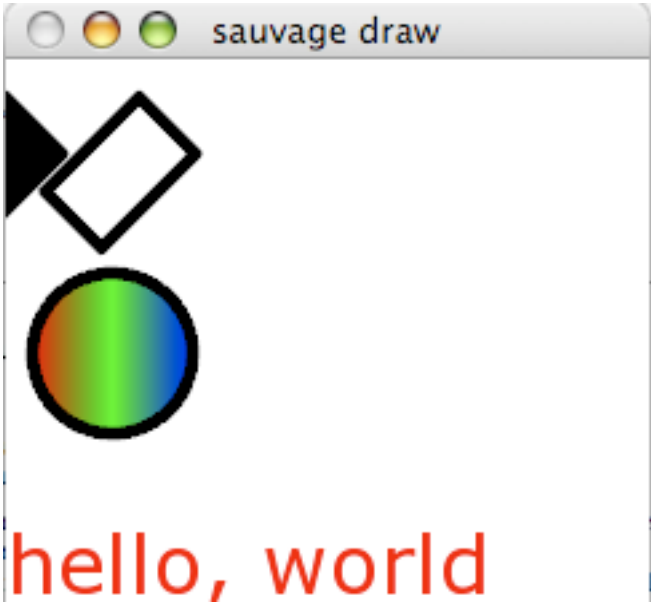
download

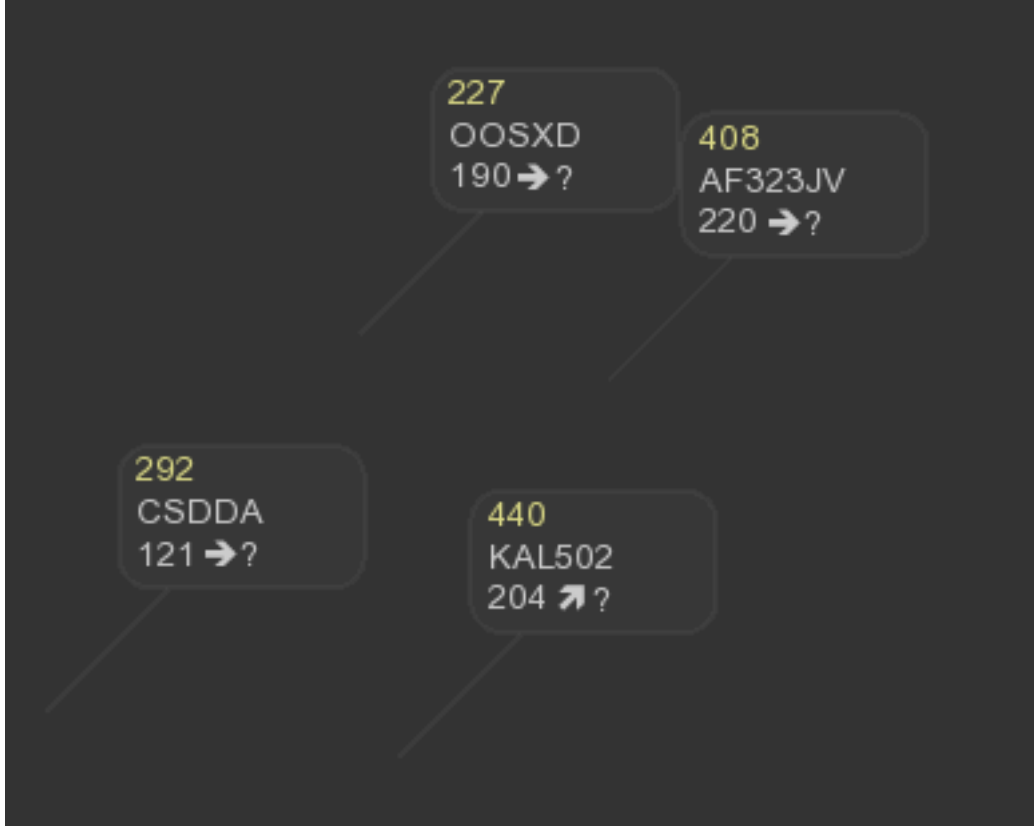
You need [sauvage.tbz](#). Follow instructions in sauvage/[README](#). Alternatively, you can use this macosx "easy install" [script](#).

status

consider it as alpha, it works well, but things may be implemented for a subset of shapes etc. sauvage is a research tool : currently, I don't have time to make it a robust library that can be used in any application. Feel free to try it and tweak it, I would be glad to add valuable additions.

screenshots





links to similar free projects

[batik](#)
[smoke](#)
[libart](#)
[agg2](#)
[zinc](#)
[xsvg](#), [cairo](#) and [glitz](#)
[rsvg](#)
[ksvg](#)

glut example

```
import sys
from OpenGL.GLUT import *
from OpenGL.GL import *
import svg
import savage

def sampleScene():
    gradient = canvas.create_lineargradient(stops=[ ((1,0,0,1), 0), ((0,1,0,1), 0.5) , ((0,0,1,1), 1) ],
    gradientUnits=svg.GradientElement.objectBoundingBox)
    canvas.create_circle(cx=10, cy=10, r=30, style={'fill':gradient, 'stroke':(0,0,0), 'stroke-width':5, 'stroke-
linejoin':svg.round_join}, transforms=[svg.Translate(30,100)])
    rect = canvas.create_rect(x=10, y=10, width=30, height=50, style={'stroke':(0,0,0), 'stroke-width':5, 'stroke-
linejoin':svg.round_join}, transforms=[svg.Rotate(45)])
    canvas.create_use(href=rect, style={'fill':None}, transforms=[svg.Translate(50, 0)])
    canvas.create_text(text="hello, world", y=200, style={'fill':(1,0,0), 'font-style': 'normal', 'font-size':30})

def display():
    try:
        glClear(GL_COLOR_BUFFER_BIT)
        canvas.gl()
        glutSwapBuffers()
    except:
        raise

def reshape(wi, hi):
    global w,h
    w,h=wi,hi
    glViewport(0,0,w,h)
    glMatrixMode(GL_PROJECTION)
```

```
glLoadIdentity()  
glOrtho(0, w, h, 0, 0, -255)  
glMatrixMode(GL_MODELVIEW)
```

```
glutInit(sys.argv)  
glutInitDisplayString("rgb double samples=4")  
glutCreateWindow("sauvage draw")  
glutDisplayFunc(display)  
glutReshapeFunc(reshape)
```

```
glEnable(GL_BLEND)  
glBlendFunc(GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA)
```

```
w,h=512,512  
canvas = sauvage.Canvas(w,h)
```

```
sampleScene()
```

```
glutMainLoop()
```

```
--
```

[conversy](http://conversy.at.cena.dot.fr) .at. cena .dot. fr