

A_simple_random_rectangle_demo

This example has been modified from [The Mobile technology group example : rectangle.py](#)

```
# random rectangle

from appuifw import *
import e32
from random import randrange

running = 1
def quit():
    global running
    running = 0
app.exit_key_handler = quit

app.screen = 'large'
app.body = canvas = Canvas()
res_x, res_y = canvas.size

while running:
    x1 = randrange(0, res_x)
    x2 = randrange(x1, res_x)
    y1 = randrange(0, res_y)
    y2 = randrange(y1, res_y)
    color = randrange(0, 255)
    canvas.rectangle((x1, y1, x2, y2), fill=color)
    canvas.refresh()
```

Here's a variation : It bounces a rectangle round the screen leaving a trail in different colors.

```
from appuifw import *
import e32
from random import randrange

running = 1
def quit():
    global running
    running = 0
app.exit_key_handler = quit

app.screen = 'large'
app.body = canvas = Canvas()
res_x, res_y = canvas.size

dy = 1
dx = 1
x1 = 10
y1 = 10

while running:
    x1 = x1 + dx
    y1 = y1 + dy
    x2 = x1 + 10
    y2 = y1 + 10
    canvas.rectangle((x1, y1, x2, y2), fill=randrange(0, 255))
    canvas.refresh()
    if (x1 < 0):
        dx = -1
    if (y1 < 0):
        dy = -1
    if (x1 > res_x - 10):
        dx = -1
    if (y1 > res_y - 10):
        dy = -1
```

A_simple_random_rectangle_demo

```
color = random.randrange(256)
rectangle((x1, y1, x2, y2), fill=color)
ao_sleep(2)
```

The following does not leave a trailing image.

```
import e32, random
import time
import appuifw
import graphics
import math
running=1
def quit():
    global running
    running=0
appuifw.app.exit_key_handler=quit
appuifw.app.screen='large'
can=appuifw.Canvas()
appuifw.app.body=can
j,k=can.size
dx=1
dy=1
x1=10
y1=10
while running:
    x1=x1+dx
    y1=y1+dy
    x2=x1+10
    y2=y1+10
    if x1<1:
        dx=dx*-1
    if y1<1:
        dy=dy*-1
    if x1>j-15:
        dx=dx*-1
    if y1>k-15:
        dy=dy*-1
    c=random.randrange(0xfffff)
    can.rectangle((x1,y1,x2,y2),outline=None,fill=c)
    e32.ao_sleep(.01)
    can.clear(0xfffff)
    e32.ao_yield()
```