

Archived:Creating screencasts with mmaker



Aquivado: Este artigo foi [arquivado](#), pois o conteúdo não é mais considerado relevante para se criar soluções comerciais atuais. Se você achar que este artigo ainda é importante, inclua o template `{{ForArchiveReview|escreva a sua justificativa}}` .

All PySymbian articles have been archived. PySymbian is no longer maintained by Nokia and is not guaranteed to work on more recent Symbian devices. It is not possible to submit apps to Nokia Store.

Introduction

In the last months I have been publishing my applications at Wiki with some screencasts. These screencasts are really useful and may help people to understand the application purpose. In this article I want to show how these screencasts are created.

MMaker

The strategy behind screencasts are simple: just a sequence of screenshots, presented as a slide show. A Python application called mmaker is used to save all screenshots and other application like Microsoft Movie Maker may be employed to create the movie. There are commercial applications that could be used to the same task but mmaker is a free alternative even with some additional work involved.

The process can be resume as:

- Define sample rate and directory for saving images and start MMaker. Phone memory may be faster for * saving images.
 - Start you application or whatever you want to show.
 - Finished ? Stop mmaker and copy all images to your PC. Remove images that you do not want to see in your movie.
 - Using Movie Maker or similar application, copy all images to the movie timeline, in the same order they were created. Do not forget to set the transition time and slide time before. I have been using 0.25s for transition and 0.5s for slide time.
 - Create and publish your movie.



Source code

The code (Python 1.4.5 but it should work with Python 1.9.6) and screenshots are below. You can download the [sis](#) or [source code](#) from the program repository.

```

# -*- coding: cp1252 -*-
# (c) Marcelo Barros de Almeida
# marcelobarrosalmeida@gmail.com
# License: GPL3

import graphics
import time
from appuifw import *
import os
import e32

class MMaker(object):
    NL = u"\u02029"
    def __init__(self, dir=u"e:\mmaker", etime=0.25):
        self.dir = dir
        self.etime = etime
        self.cnt = 0
        self.running = False
        self.timer = e32.Ao_timer()
        menu = [(u"Start", self.start),
                 (u"Stop", self.stop),
                 (u"Set dir", self.set_dir),
                 (u"Set time", self.set_time),
                 (u"Clean dir", self.clean_dir),
                 (u"About", self.about),
                 (u"Quit", self.close_app)]
        self.body = Text()
        app.screen = "normal"
        app.body = self.body
        app.menu = menu
        app.tile = u"MMaker"

```

```
self.lock = e32.Ao_lock()
app.exit_handler = self.close_app

def close_app(self):
    self.running = False
    self.timer.cancel()
    self.lock.signal()

def run(self):
    self.lock.wait()
    app.set_tabs( [], None )
    app.menu = []
    app.body = None
    app.set_exit()

def set_dir(self):
    dir = query(u"Images dir:","text",self.dir)
    if dir is not None:
        if not os.path.isdir(dir):
            try:
                os.makedirs(dir)
            except:
                self.body.add((u"Can't create %s" % dir) + self.NL)
            return
    self.dir = dir
    self.body.add((u"New dir is %s" % self.dir) + self.NL)

def clean_dir(self):
    yn = popup_menu([u"No",u"Yes"],u"Clean %s?" % self.dir)
    if yn is not None:
        if yn == 1:
            self.cnt = 0
            files = os.listdir(self.dir)
            for f in files:
                fp = os.path.join(self.dir,f)
                try:
                    os.remove(fp)
                    self.body.add((u"%s deleted" % fp) + self.NL)
                except:
                    self.body.add((u"Cant't delete %s" % fp) + self.NL)

def set_time(self):
    tm = query(u"Time","float",self.etime)
    if tm is not None:
        self.body.add((u"New time is %f" % tm) + self.NL)
        self.etime = tm

def start(self):
    self.timer.after(self.etime,self.take_screenshot)
    self.running = True
    self.body.add(u"Started" + self.NL)

def stop(self):
    self.running = False
    self.timer.Cancel()
    self.body.add(u"Stopped" + self.NL)

def take_screenshot(self):
    if self.running:
        ss = graphics.screenshot()
        name = os.path.join(self.dir,"mm%04d.png" % self.cnt)
        self.cnt += 1
        ss.save(name)
        self.body.add((u"Screenshot %s" % name) + self.NL)
        self.timer.after(self.etime,self.take_screenshot)

def about(self):
    note(u"MMaker by Marcelo Barros (marcelobarrosalmeida@gmail.com)","info")

mm = MMaker()
mm.run()
```

