Archived:PySymbian for 5th Edition with sample applications

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All PySymbian articles have been archived. PySymbian is no longer maintained by Nokia and is not guaranteed to work on more recent Symbian devices.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

With the launch of S60 5th Edition SDK, a new gate opens for S60 developers, to develop Touch UI based applications.

PySymbian is Nokia's port of the Python language to the S60 smartphone platform. In addition to the standard features of the Python language, PySymbian provides access to many of the phone's uniquely smartphone-y functions, such as camera, contacts, calendar, audio recording and playing, Bluetooth communications, simple telephony, etc.

All releases of Python for S60 are available at Sourceforge.net The latest release of PySymbian till date is Python 1.4.4 which is available for 2nd edition and 3rd edition devices and Emulators. Nokia hasn't yet released the PySymbian version for 5th edition SDKs. But in this article, we will see how to use PySymbian on 5th edition emulator.

Tools and SDKs Required

- S60 5th edition SDK which is available for download at Nokia Developer website.
- As PySymbian for 5th edition is not yet released and available, we will try to use PySymbian SDK for 3rd edition on 5th edition SDK. Python SDKs are available at Sourceforge.net
- A simple text editor for trying sample applications and scripts - for eg. word or notepad.

Installation and Setup

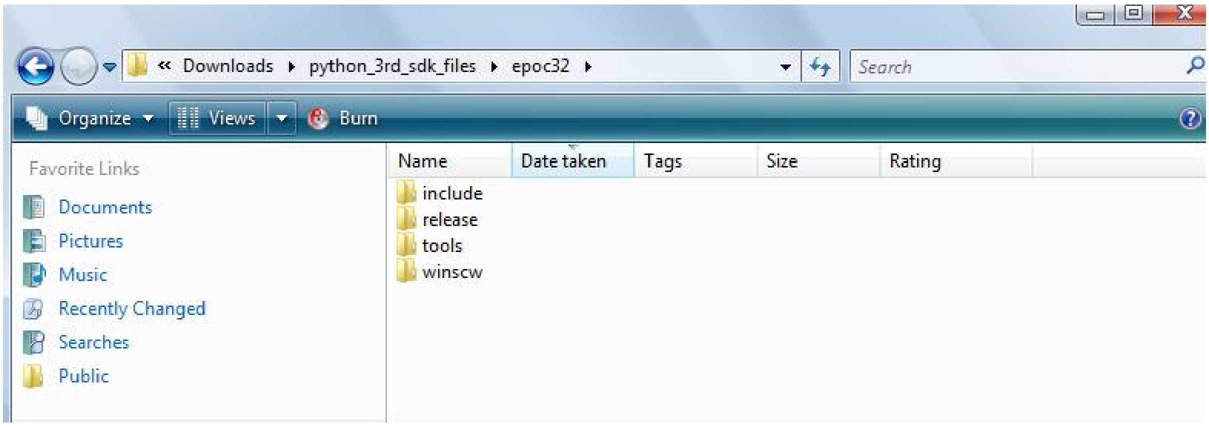
- Download and install S60 5th edition SDK to the default path (C:)

After installing the SDK, launch C:\S60\devices\S60_3rd_FP2_SDK\epoc32\release\winscw\udeb\epoc.exe The emulator should launch, as other emulators, it takes some time to launch. Select the country, date and time as required. After you are done with exploring for a while, close the emulator.

- Download Python SDK from Sourceforge.net - PythonForS60_1_4_4_SDK_3rdEd.zip
- Open the PythonForS60_1_4_4_SDK_3rdEd with winzip or winrar. Unzip the epoc32 folder to a folder named **python_3rd_sdk_files***Italic text*.

The unzipped epoc32 folder should have following 4 folders,

- - include
  - release
  - tools
  - winscw



- Now Open the path C:\S60\devices\S60_5th_Edition_SDK_v0.9\epoc32 in the windows explorer.
- Copy files and folders from the below mentioned to the epoc32 folder of 5th Edition SDK.

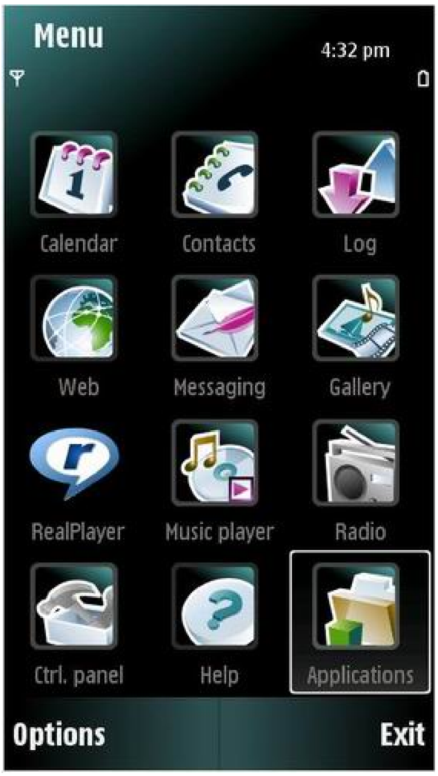
This may take some manual labour, But please take care not to replace the epoc32 folder with the unzipped folder.

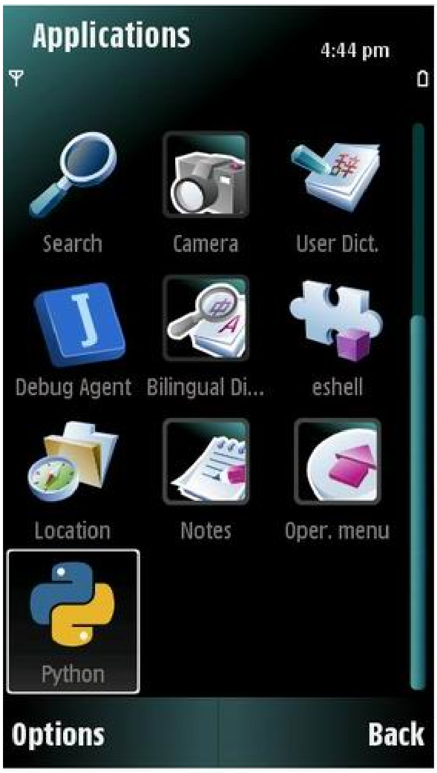
```
\python_3rd_sdk_files\epoc32\include
\python_3rd_sdk_files\epoc32\include\python
\python_3rd_sdk_files\epoc32\release\armv5\lib
\python_3rd_sdk_files\epoc32\release\winscw\udeb
\python_3rd_sdk_files\epoc32\release\winscw\udeb\z\private\10003A3F\apps
\python_3rd_sdk_files\epoc32\release\winscw\udeb\z\resource\apps
```

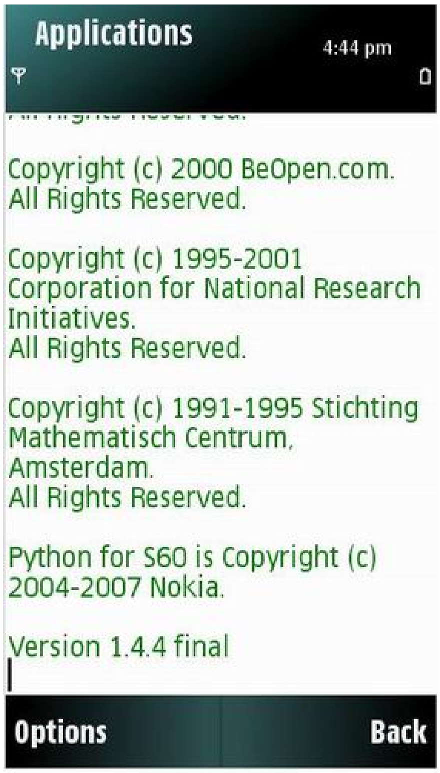
```
\python_3rd_sdk_files\epoc32\tools
\python_3rd_sdk_files\epoc32\winscw\c\private
\python_3rd_sdk_files\epoc32\winscw\c\python
\python_3rd_sdk_files\epoc32\winscw\c\resource
\python_3rd_sdk_files\epoc32\winscw\c\resource\encodings
\python_3rd_sdk_files\epoc32\winscw\c\sys\bin
```

- Launch the emulator again -> C:\S60\devices\S60_3rd_FP2_SDK\epoc32\release\winscw\udeb\epoc.exe
- Go to Menu -> Applications-> Python !

Follow the screen shots below.







All Rights Reserved.

Copyright (c) 2000 BeOpen.com.
All Rights Reserved.

Copyright (c) 1995-2001
Corporation for National Research
Initiatives.
All Rights Reserved.

Copyright (c) 1991-1995 Stichting
Mathematisch Centrum,
Amsterdam.
All Rights Reserved.

Python for S60 is Copyright (c)
2004-2007 Nokia.

Version 1.4.4 final

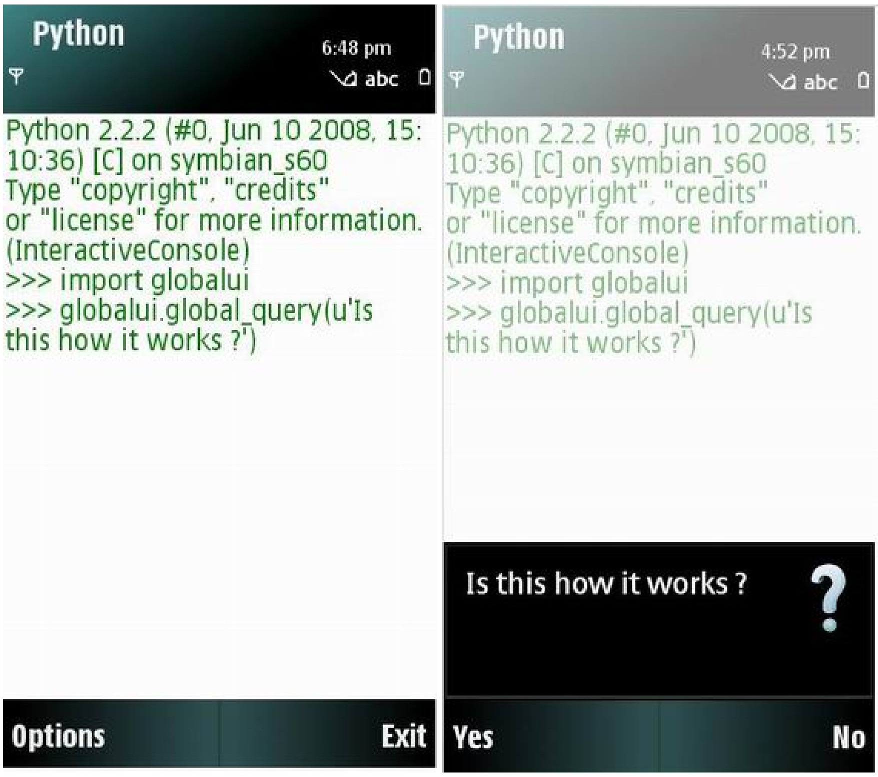
Thus PySymbian is now installed on your S60 5th edition emulator. Let us try few things.

Trying the Touch UI with PySymbian

Launch Python on the emulator -> Options -> Interactive Console

Write the following commands...

```
>>>import globalui
>>>globalui.global_query(u"Is this how it works?")
```

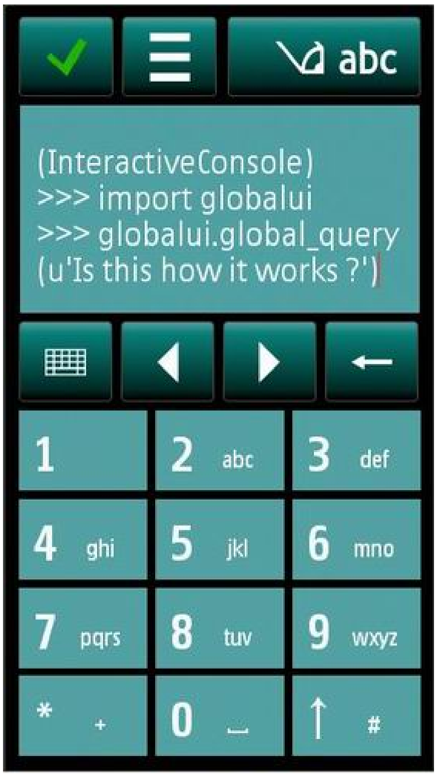


```
Python 2.2.2 (#0, Jun 10 2008, 15:
10:36) [C] on symbian_s60
Type "copyright", "credits"
or "license" for more information.
(InteractiveConsole)
>>> import globalui
>>> globalui.global_query(u'Is
this how it works ?')
```

```
Python 2.2.2 (#0, Jun 10 2008, 15:
10:36) [C] on symbian_s60
Type "copyright", "credits"
or "license" for more information.
(InteractiveConsole)
>>> import globalui
>>> globalui.global_query(u'Is
this how it works ?')
```

Is this how it works ?

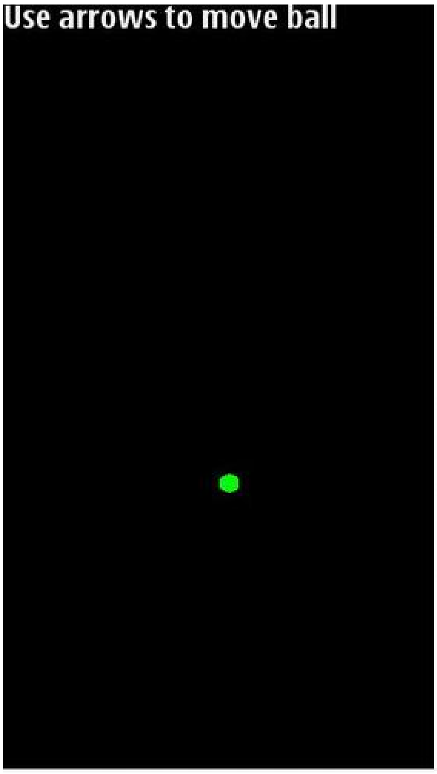
The 5th edition emulator has an Alphanumeric Keyboard - so the brackets would be accomplished in that keyboard by pressing the '*' key. The screen shots are shown below.



```
(InteractiveConsole)
>>> import globalui
>>> globalui.global_query
(u'Is this how it works ?')
```

Next lets try the ball.py which takes keypress as input in 5th edition pre-releases (3rd, 2nd and 1st edition) devices.

Go to Python -> Options -> Run Script -> Ball.py



Use arrows to move ball

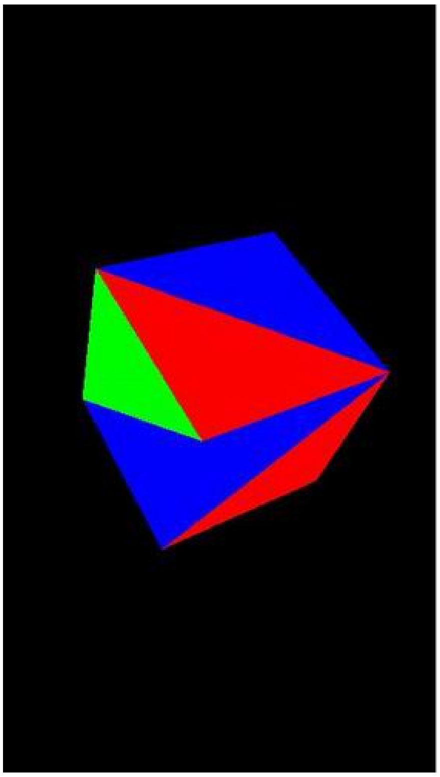
Move the mouse on the emulator screen and the ball will follow it.

Other Sample Applications

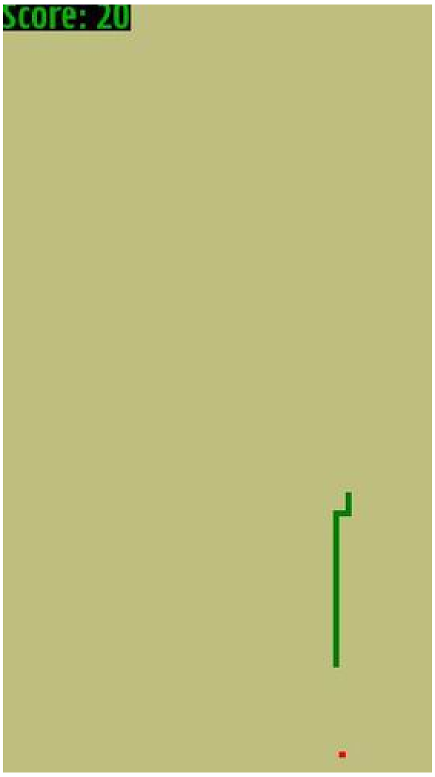
If all the folders mentioned above have been copied properly, then some scripts can be found in epoc32\winscw\c\python folder.

Launch Python on the emulator -> Options -> Run Script ->

1) Samplecube.py



2) Snake.py



Strange enough the snake doesn't respond to Touch interface as the Ball.py does.

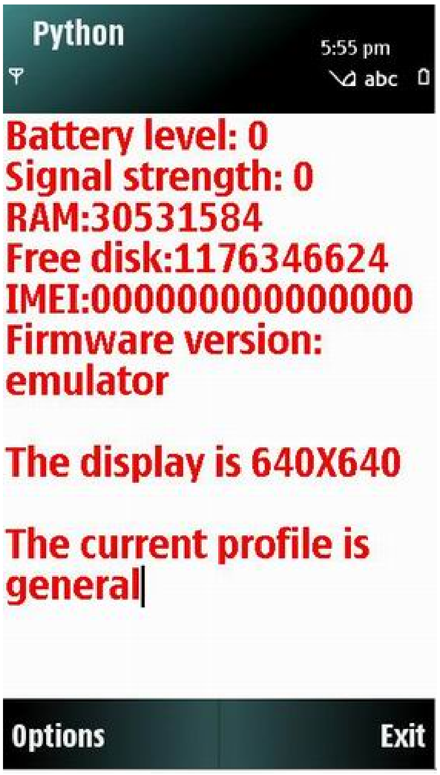
3) Further, lets try out the following application which gives system information.

The module we use is sysinfo.

```
#import modules
import sysinfo,appuifw, e32
#switch screen to large
#appuifw.app.screen='large'
```

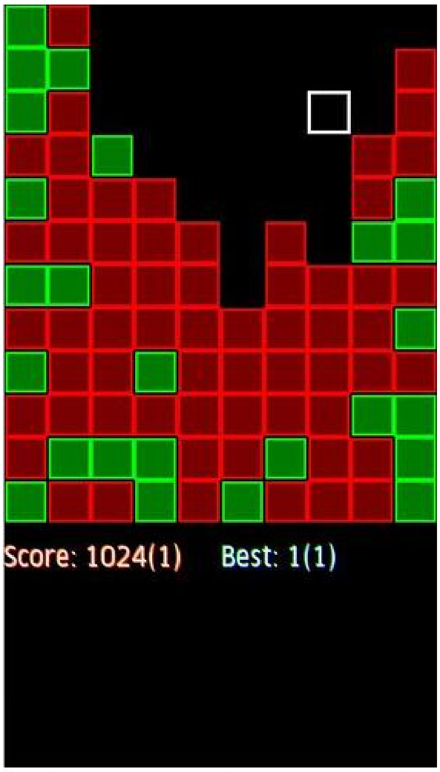
```
app_lock = e32.Ao_lock()

#define exit function
def exit_key_handler():

    app_lock.signal()
    sys.exit()

#Prepare for red text,
tx=appuifw.Text()
tx.color=(255,0,0)
tx.font=u"Nokia Hindi S6017" # specify font
tx.style = appuifw.STYLE_BOLD
tx.clear()


appuifw.app.body=tx

tx.add("Battery level: "+unicode(sysinfo.battery())) #prints battery strength
tx.add("\n")
tx.add(u"\nSignal strength: "+unicode(sysinfo.signal_bars())) #prints signal strength

#Both values are between 0 and 7

tx.add(u"\nRAM:"+unicode(sysinfo.free_ram())) #print free ram
tx.add("\n")
tx.add(u"\nFree disk:"+unicode(sysinfo.free_drivespace()['C:'])) #Or E: for memory card
#prints disk space
tx.add("\n")
tx.add(u"\nIMEI:")
tx.add(unicode(sysinfo.imei())) #prints IMEI of the device
tx.add("\n")
tx.add(u"\nFirmware version:")
tx.add(unicode(sysinfo.sw_version()))  #prints firmware version
tx.add("\n")
tx.add(u"\n\nThe display is
"+unicode(sysinfo.display_pixels()[0])+"X"+unicode(sysinfo.display_pixels()[1])) #prints
phone's display in pixels
tx.add(u"\n\nThe current profile is ")
tx.add(unicode(sysinfo.active_profile())) #prints current profile

appuifw.app.exit_key_handler = exit_key_handler
app_lock.wait()
```

When we run the script, we get the following output.



```
Battery level: 0
Signal strength: 0
RAM:30531584
Free disk:1176346624
IMEI:000000000000000
Firmware version:
emulator

The display is 640X640

The current profile is
general
```

4) Next, lets try Unity game by JOM



The Unity game also, doesn't respond to the touch interface of the emulator as the Ball.py does.

Conclusion

The new Touch UI offers great opportunities in mobile application development and could prove versatile with rapid prototyping languages like Python. PySymbian for 3rd edition runs on S60 5th edition SDK,but there are some hinges, which hopefully would be sorted out in the official PySymbian release for 5th edition devices.

This article is written by croozeus 17:41, 7 October 2008 (EEST)

See Also

- Archived:Progress bar on S60 5th Edition using PySymbian
- Archived:Simple PySymbian alarm clock for S60 5th Edition