

## Criando animações em Python para S60

O Python para S60 é uma linguagem robusta, simples e bastante poderosa. Apesar de não prover APIs nativas para desenvolvimento de games, como as existentes no PyGame, podemos facilmente criar algumas destas classes.

O exemplo a seguir serve apenas como base para o desenvolvimento da classe Sprite existente no MIDP 2.0 do Java ME. A criação de um framework para games ajudaria a simplificar o desenvolvimento de aplicações com o Python para S60.

## Primeiros passos

Se você está iniciando em Python para S60, aconselho a [leitura prévia de materiais de introdução](#).

## Exemplo

Caso este seja seu primeiro contato com o Python para S60, faça o [download](#) do runtime e do python script shell.

Após a instalação, crie um diretório com o nome Python no seu cartão de memória e copie o código abaixo em um novo arquivo com a extensão .py

A imagem também deve ser copiada para a pasta Python recém-criada no cartão de memória.



```
import appuifw
import e32
from graphics import Image

## Sprite
# Similar a classe Sprite do pacote game disponível na MIDP 2.0
# Este trecho de código foi criado em apenas 30 minutos, e nao foi exaustivamente testado
# podendo existir diversas falhas em sua implementacao, qualquer melhoria é bem vinda.
class Sprite:
    ## Construtor da classe
    # @param self The object pointer.
    def __init__( self, aImg, aFrWidth, aFrHeight ):
        self._iImg = aImg
        self._iFrWidth = aFrWidth
        self._iFrHeight = aFrHeight
        self._iCurrentFrame = 1
        self._iIndFrame = 0
        self._iWidth, self._iHeight = self._iImg.size
        self._iMaxWidthFrames = self._iWidth / self._iFrWidth
        self._iMaxHeightFrames = self._iHeight / self._iFrHeight
        self._iFrameSequence = [0]
        self._iPosX = 0
        self._iPosY = 0
    # end __init__

    ## Seta a sequencia de frames
```

## Criando animações em Python para S60

```
# @param self The object pointer.
# @param aSeq Sequencia dos frames.
def set_frame_sequence( self, aSeq ):
    self._iIndFrame = 0
    self._iFrameSequence = aSeq
    self._iCurrentFrame = self._iFrameSequence[self._iIndFrame]
# end set_frame_sequence

## Seta o posicionamento do objeto na tela
# @param self The object pointer.
# @param aPos Posicionamento do objeto.
def set_position( self, aPos ):
    self._iPosX, self._iPosY = aPos
# end set_position

## Seta o próximo frame
# @param self The object pointer.
def next_frame( self ):
    self._iIndFrame += 1
    if self._iIndFrame == len(self._iFrameSequence):
        self._iIndFrame = 0
    self._iCurrentFrame = self._iFrameSequence[self._iIndFrame]
# end next_frame

## Seta o frame atual
# @param self The object pointer.
# @param aFrame frame.
def set_frame( self, aFrame ):
    self._iCurrentFrame = self._iFrameSequence[aFrame]
# end set_frame

## Desenha o objeto na tela de acordo com a posicao e o frame atual
# @param self The object pointer.
# @param aGraphics Canvas.
def paint ( self, aGraphics ):
    aGraphics.blit(self._iImg, target=(self._iPosX, self._iPosY), \
                    source=((self._iCurrentFrame-1) * self._iFrWidth, 0, self._iCurrentFrame

# end paint

## SpriteAnimation
# Responsavel por criar os objetos na tela
class SpriteAnimation:
    ## Construtor da classe
    # @param self The object pointer.
    def __init__( self ):
        # Define o corpo da aplicação
        self._iCanvas = appuifw.Canvas()
        appuifw.body = self._iCanvas

        # altura e largura do canvas
        self._cWidth, self._cHeight = self._iCanvas.size

        # Sprite
        self._iImgPlayer = Image.open("e:\\Python\\tiled_owner_flatten.png")
        self._iPlayer = Sprite(self._iImgPlayer, 60, 70)
        self._iPlayer.set_frame_sequence([5, 6, 7])
        self._iPlayer.set_position((self._cWidth/2 - 30, self._cHeight/2 - 35))

        # define o evento de saída da aplicação
        appuifw.app.exit_key_handler = self.quit

        # seta o status da aplicacao
```

## Criando animaÃ§Ãµes em Python para S60

```
self._iRunning = True

# define um background branco
self._iCanvas.rectangle((0, 0, 240, 320), outline = 0xFFFFFF, width=240)

# define a resolucao da tela
appuifw.app.screen = 'full' # (full, large, normal)

# cria um loop principal da aplicaÃ§Ã£o
self.paint()
# end __init__

## Redesenha os objetos na tela
# @param self The object pointer.
def paint( self ):
    while self._iRunning:
        self._iCanvas.clear()
        self._iPlayer.paint( self._iCanvas )
        self._iPlayer.next_frame()

        # Processa eventos pendentes
        e32.ao_yield()

        # taxa de atualizaÃ§Ã£o
        e32.ao_sleep(0.1)
# end paint

## Manipula a saÃ­da do aplicativo
# @param self The object pointer.
def quit( self ):
    self._iRunning = 0
# end quit

animation = SpriteAnimation()
```

## Autor

FelipeAndrade -- 04:35, 24 June 2008 (EEST)