Reviewer Approved 

This is an example of a basic game of Tic-Tac-Toe in PyS60.

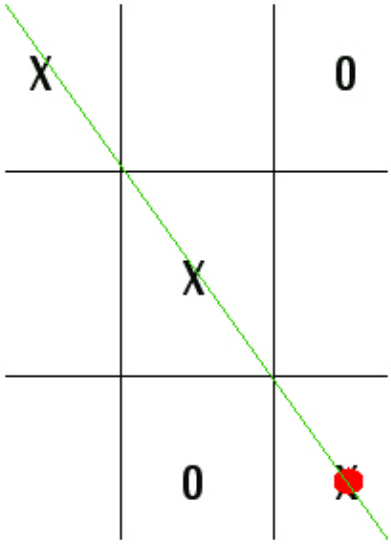
Code

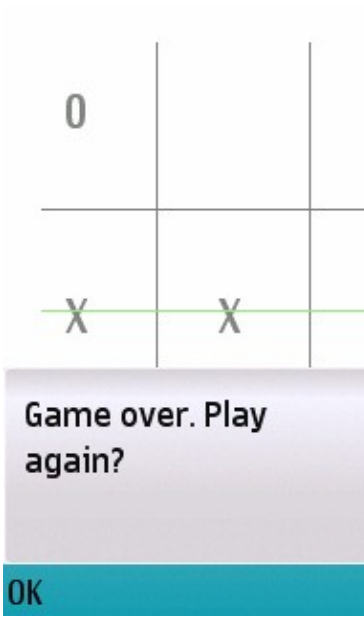
```
import appuifw, e32, random
from graphics import *
from key_codes import *

app_lock=e32.Ao_lock()
def quit():
global running
        running=0
        app_lock.signal()
appuifw.app.exit_key_handler=quit

def newgame():
global running, bg, xcoord, ycoord, gridmatrix, dotx, doty

#The application takes up the entire screen
        appuifw.screen="full"

class Keyboard(object):
def __init__(self,onevent=lambda:None):
self._keyboard_state={}
self._downs={}
self._onevent=onevent
def handle_event(self,event):
             if event['type']==appuifw.EEventKeyDown:
             code=event['scancode']
             if not self.is_down(code):
             self._downs[code]=self._downs.get(code,0)+1
             self._keyboard_state[code]=1
             elif event['type']==appuifw.EEventKeyUp:
             self._keyboard_state[event['scancode']]=0
             self._onevent()
def is_down(self,scancode):
             return self._keyboard_state.get(scancode,0)
def pressed(self,scancode):
             if self._downs.get(scancode,0):
             self._downs[scancode]-=1
             return True
return False
        keyboard=Keyboard()

        bg=Image.new((240,320))
def handle_redraw(rect):canvas.blit(bg)
        canvas=appuifw.Canvas(event_callback=keyboard.handle_event, redraw_callback=handle_redraw)
        appuifw.body=canvas

#In order to keep track of the state of the game, we use a matrix filled with:
#0 if the square is empty, 1 if the square is occupied by the phone and 2 if the square is occupi
        gridmatrix=[[0,0,0],[0,0,0],[0,0,0]]

#Writing X or O requires appropriate coordinates
       xcoord=[32,112,192]
       ycoord=[65,172,279]
```

```
#To show the player where the cursor is, we display a red dot (by default in the middle of the gr
        dotx=doty=1

#To keep track of whose turn it is, we use the variable turn (1 for phone, 2 for player)
#The player starts the game
        turn=2

        gameover=False

#We draw the grid
def drawgrid():
global xcoord, ycoord, gridmatrix, bg, dotx, doty
   clear()      bg.
   line((80,20,80,300), 0)
   line((160,20,160,300), 0)
   line((20,107,220,107), 0)
   line((20,214,220,214), 0)
for i in range(3):
for j in range(3):
if(gridmatrix[i-1][j-1]==1):bg.text((xcoord[i-1],ycoord[j-1]), u"O", font="title")
elif(gridmatrix[i-1][j-1]==2):bg.text((xcoord[i-1],ycoord[j-1]), u"X", font="title")
   point((xcoord[dotx]+7,ycoord[doty]-10), 0xff0000, width=15)
        drawgrid()

#Now we handle the keypress events by moving the dot accordingly
        running=1
while(running==1):
#Set the menu
        app.menu=[(u"New game", newgame), (u"Exit", quit)]

if(keyboard.pressed(EScancodeRightSoftkey)):quit()

#When it's the player's turn...
if(turn==2):
if(keyboard.pressed(EScancodeSelect)):  #If the "select" key is pressed
if(gridmatrix[dotx][doty]==0):  #If the square is available
   text((xcoord[dotx],ycoord[doty]), u"X", bg.font="title")
           [dotx][doty]=2                gridmatrix
              (())                       handle_redraw
        ()                               drawgrid
                                         turn=1
if((keyboard.pressed(EScancodeLeftArrow)) and (dotx>0)):
                                dotx-=1
              (())              handle_redraw
        ()                      drawgrid
if((keyboard.pressed(EScancodeRightArrow)) and (dotx<2)):
                                dotx+=1
              (())              handle_redraw
        ()                      drawgrid
if((keyboard.pressed(EScancodeUpArrow)) and (doty>0)):
                                doty-=1
              (())              handle_redraw
        ()                      drawgrid
if((keyboard.pressed(EScancodeDownArrow)) and (doty<2)):
      1                         doty+=
              (())              handle_redraw
        ()                      drawgrid

#We now check if the game is over. This can happen if one of the two has a sequence of 3 symbols,
         True   gameover=
for i in range(3):
for j in range(3):
```

```
if(gridmatrix[i-1][j-1]==0):gameover=False

#To show the player the winning row, we strike it through with a green line
if(gridmatrix[0][0]==gridmatrix[0][1]==gridmatrix[0][2]<>0):
   line(((40,20),(40,300)),bg, 0x33CC00)
         True           gameover=
    ao_sleep(1)         e32.
if(gridmatrix[1][0]==gridmatrix[1][1]==gridmatrix[1][2]<>0):
   line(((120,20),(120,300)),bg, 0x33CC00)
         True           gameover=
    ao_sleep(1)         e32.
if(gridmatrix[2][0]==gridmatrix[2][1]==gridmatrix[2][2]<>0):
   line(((200,20),(200,300)),bg, 0x33CC00)
         True           gameover=
    ao_sleep(1)         e32.
if(gridmatrix[0][0]==gridmatrix[1][0]==gridmatrix[2][0]<>0):
   line(((20,53),(220,53)),bg, 0x33CC00)
         True           gameover=
    ao_sleep(1)         e32.
if(gridmatrix[0][1]==gridmatrix[1][1]==gridmatrix[2][1]<>0):
   line(((20,160),(220,160)),bg, 0x33CC00)
         True           gameover=
    ao_sleep(1)         e32.
if(gridmatrix[0][2]==gridmatrix[1][2]==gridmatrix[2][2]<>0):
   line(((20,267),(220,267)),bg, 0x33CC00)
         True           gameover=
    ao_sleep(1)         e32.
if(gridmatrix[0][0]==gridmatrix[1][1]==gridmatrix[2][2]<>0):
   line(((20,20),(220,300)),bg, 0x33CC00)
         True           gameover=
    ao_sleep(1)         e32.
if(gridmatrix[2][0]==gridmatrix[1][1]==gridmatrix[0][2]<>0):
   line(((20,300),(220,20)),bg, 0x33CC00)
         True           gameover=
    ao_sleep(1)         e32.

if(gameover==True):
if(appuifw.query(u"Game over. Play again?", "query")):newgame()
else:quit()

#When it's the phone's turn and the game is not over yet...
while((turn==1) and (gameover==False)):
   random.choice([0,1,2])px=
   random.choice([0,1,2])py=
if(gridmatrix[px][py]==0):
   text((xcoord[px],ycoord[py]),u"O", font="title")
          [px][py]=1        gridmatrix
             (())           handle_redraw
        ()                  drawgrid
     2                      turn=

    ao_yield()  e32.

#Tell the application to start a new game immediately after launch
newgame()
```

Screenshots





Download

The install package (signed with Ensymble's certificate) is available to download here.