

PYGAME ZERO

GAME CODING MADE EASY

It's designed to make it easier for teachers and students to start making games with Python. Sean McManus speaks to its creator, Daniel Pope...

If you've written a Python game, you've probably wrestled with Pygame. It's one of the go-to libraries for games programming, providing code for displaying images, drawing shapes, playing sounds, and writing text. The Python games in Raspbian also use it, but if you look at any Pygame code, you'll see that it can be a bit convoluted.

At the 2012 UK Python Conference, PyCon UK, professional programmer Daniel Pope participated in the education

track. "I dashed off the most simple Pygame program I could, a rabbit sliding across the screen – about 15 lines of Python," he tells us. "The teachers told me it was too complicated to teach – that in the context of a 30-minute lesson, just tasked with reproducing my code, some of the students would still be struggling with non-runnable code at the same time that others were done and waiting for the next challenge."

He's now released Pygame Zero as a solution to that problem. "The simplest programs are just a couple of lines. And Pygame Zero lets teachers easily break lessons into bite-size steps: it feels like you can keep changing a couple of lines at a time and keep making meaningful progress."

Outside the classroom, it'll also help hobbyist programmers to get results more quickly.

Daniel is a two-times winner and nine-times entrant in PyWeek (pyweek.org), a week-long games programming competition. *Art Attack* from PyWeek 12 and *Legend Of Goblit* from PyWeek 19 were made using Pygame. His other games were made with Pyglet (pyglet.org), another library which had some influence on parts of Pygame Zero's design.

As an experienced Pygame coder, he could see the opportunity to improve upon it. "Pygame is a library that allows you to access input, load and draw graphics, play sounds – but you

have to decide how to make the library do those things and write a little bit of code to do it," he says. "Mostly, people make the same decisions, so that code ends up looking the same. Pygame Zero makes some of the decisions for you, so your program doesn't need to include that code."

Pygame Zero also adds a few features on top of Pygame. "Pygame is just a library for input, graphics, sound, and so on," Daniel says. "We have provided some extra ability to animate objects, draw text in a number of styles, and schedule in-game events – for example, to create a power-up every 60 seconds."

The main event

This is one way that Pygame Zero enables more events-driven programming. Instead of having to loop endlessly to check for a mouse click, for example, you can write a function that is triggered when the mouse button is pressed. Pygame Zero does the checking for you in the background.

"Event-driven code is a very natural way to write programs that respond to input from the user, a hardware device, or a network," says Daniel. "Most Pygame programs use an event loop but require you to write it yourself, and call your own code when certain types of events occur. In Pygame Zero, we keep the event loop inside the framework so you don't have to write it, and it will call any handlers you define. It's really just a simpler way of working

Below *Art Attack*, one of Daniel's PyWeek entries made with Pygame



INSTALLING PYGAME ZERO

Pygame Zero will be included in a future Raspbian release, but for now you can install using the Python package manager, pip. From the Raspbian command line, enter these commands to install pip (if necessary) and Pygame Zero (which utilises Python 3):

```
sudo apt-get update
sudo apt-get install python3-
setuptools python3-pip
sudo pip-3.2 install pgzero
Read the docs at: bit.ly/PgZero
```

with the same programming model, but it does feel different to work with, more organised.”

The new library also simplifies image handling by introducing the concept of actors, which provide an easy way to load and manage picture files. These are ideal for game characters, because you can easily move them around by just changing their x and/or y position. There’s support for positioning by one of the actor’s corners too, so you can avoid the messy calculations that might otherwise be associated with that.

Pygame Zero can best help with 2D graphical games. “Retro games would be a very good fit,” Daniel reckons, “but some more modern 2D smartphone games can be easily written too, like *Flappy Bird*.”

For those who want to write more advanced games, Pygame Zero can be a useful stepping stone. “Pygame Zero is likely to be slow if you have a lot of things to draw,” advises Daniel. “Lots of the ways of



Photo by Richard Cooper

Above Daniel Pope: “We write games so that others can dip into our imaginations”

Left *The Legend of Goblit*, Daniel’s winning PyWeek entry last year, based on Pygame

Pygame Zero improves compatibility by enforcing some simple rules. It requires images to be stored in a folder called ‘images’ and sounds to be stored in a folder called ‘sounds’. There are rules on valid filenames too, to avoid cross-platform compatibility issues.

As for next steps, Daniel says the short-term focus is on ensuring that

“ Pygame Zero enables more events-driven programming – a more natural way to code ”

improving performance will require some control of Surfaces and when to draw or invalidate them. Support for that kind of thing is mostly in Pygame itself. Also, there are APIs in Pygame to rotate sprites, access webcams, joysticks, and more. So there’s lots in the Pygame toolbox that you can experiment with as you get more experienced.”

Sharing is caring

“The reason we write games is to share them,” says Daniel. “We write games so that others can dip into our imaginations and have fun with our ideas. It’s always very disappointing when you want to play someone else’s game but it doesn’t work, and sadly this is often the situation when sharing games written on a computer with different hardware and software.”

Pygame Zero meets its educational goals. “If something seems confusing to use, or crashes with an error message that doesn’t give good information about what went wrong, that’s a bug – let us know,” he says. “Longer term, I’m keen to explore better ways of sharing and distributing games; I’d love to see a website dedicated to sharing Pygame Zero games. But we could also try to make Pygame Zero games work even without Python pre-installed. And maybe we will make progress on the lack of hardware acceleration in Pygame. I’ve spoken to people in the Python community who have part of the solution to these problems. Pygame Zero, Pygame, and Python are all open-source projects and benefit from the collaboration of lots of people solving little parts of the problem.”

HELLO_WORLD.PY

```
Pygame Zero Game
Hello World!

Try our example Pygame Zero script. Create a subfolder in the same directory called images and copy the file princess.png from Raspbian's python_games folder into it. To run the code, use pgzrun helloworld.py from the command line. The update() function is called once per frame, moving the princess across the window. Click the mouse button to make her jump.
Read the docs at: bit.ly/PgZero

# Pygame Zero demo from The MagPi
player = Actor('princess')
player.pos = 50, 50

WIDTH = 500 # window size
HEIGHT = 80

def draw():
    screen.fill((128,128,0)) # green stage
    player.draw() # show princess.png
    screen.draw.text("Hello World!", topright=(480,30))

def update():
    player.x += 2 # move princess.png
    if player.topleft[0] > WIDTH:
        player.x = -50

def on_mouse_down():
    player.y -= 10 # jump up
    clock.schedule_unique(player_reset, 0.5)

def player_reset():
    player.y += 10 # fall down again
```