Pyganim Documentation

Release 0.9.0

Al Sweigart

November 24, 2016

Contents

Contents:

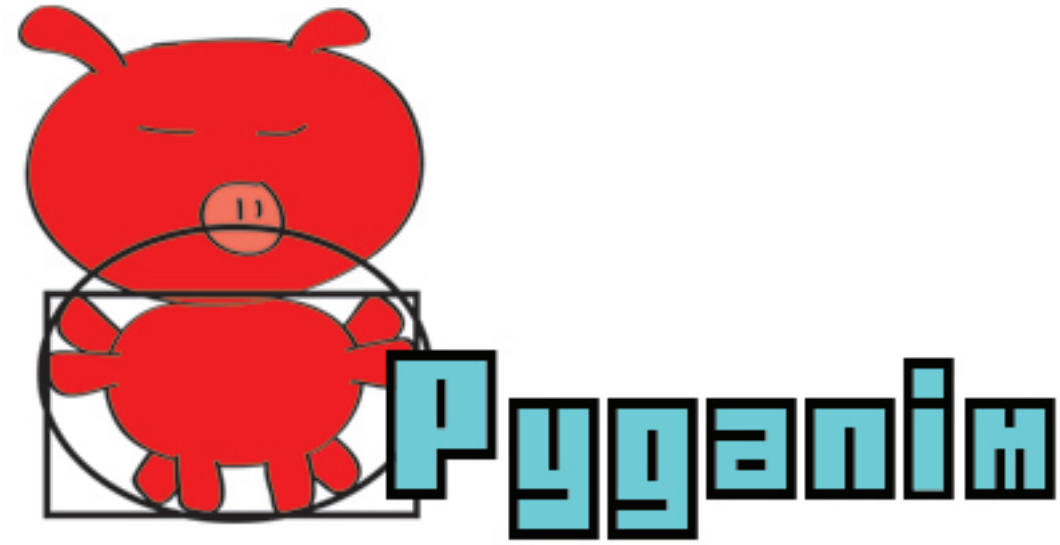
CHAPTER 1

Installation

1.1 Background Information

Pyganim (pronounced like "pig" and "animation") is a Python module for Pygame that makes it easy to add sprite animations to your Pygame programs. Pyganim works with Python 2 and Python 3.

The mascot of Pyganim is a red vitruvian pig.



Pyganim was written by Al Sweigart and released under a "Simplified BSD" license. Contact Al with any questions/bug reports: al@inventwithpython.com

This documentation can be found at https://pyganim.readthedocs.org

Pyganim requires Pygame to run, and also requires PIL or Pillow to use the animated GIF loading feature.

Pyganim runs on Python 2.5, 2.6, 2.7, 3.1, 3.2, 3.3, 3.4.

Currently there is no Pillow module for Python 3.1, so the animated GIF loading does not work on that version. There is no Pygame version currently (Aug 2015) available for Python 3.5.

1.2 Installation

Pyganim can be installed using pip by running:

```
pip install pyganim
```

The PyPI entry is at https://pypi.python.org/pypi/Pyganim

To test if the installation worked, run `import pyganim` from the interactive shell. Pygame (and, optionally, PIL or Pillow) will need to be installed separately to load animated gifs.

CHAPTER 2

Pyganim Basics

2.1 Quick Start

First, create an animation object by calling the PygAnimation constructor and passing it a list of tuples. These tuples represent a single "frame" of the animation. The tuples have an image's filename and the number of milliseconds it is displayed before displaying the next frame:

```
>>> import pyganim
>>> animObj = pyganim.PygAnimation([('frame1.png', 200), ('frame2.png', 200), ('frame3.png', 600)])
>>> animObj.play()
```

Then, during the program's loop when it must draw to the Surface object, call the `blit()` method and pass it the Surface object to draw on along with the XY coordinates:

```
animObj.blit(windowSurface, (x, y))
```

The correct frame will be drawn to the Surface depending on the system time when `blit()` was called.

2.2 Example Usage

Here's a small example program, given the following lightning bolt images:

The source code is:

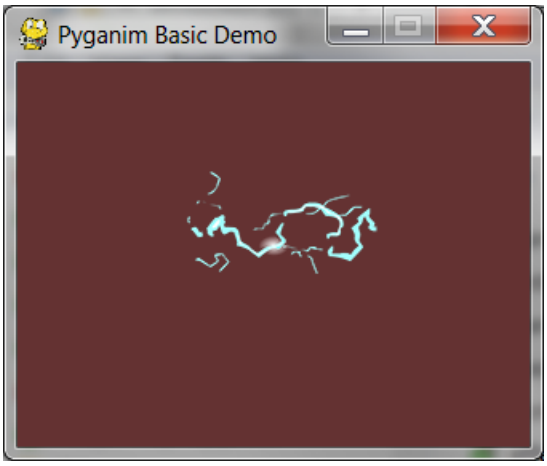
```
import pygame
from pygame.locals import *
import pyganim

pygame.init()
windowSurface = pygame.display.set_mode((320, 240), 0, 32)
pygame.display.set_caption('Pyganim Basic Demo')

boltAnim = pyganim.PygAnimation([('bolt_strike_0001.png', 100),
                                 ('bolt_strike_0002.png', 100),
                                 ('bolt_strike_0003.png', 100),
                                 ('bolt_strike_0004.png', 100),
                                 ('bolt_strike_0005.png', 100),
                                 ('bolt_strike_0006.png', 100),
                                 ('bolt_strike_0007.png', 100),
                                 ('bolt_strike_0008.png', 100),
                                 ('bolt_strike_0009.png', 100),
                                 ('bolt_strike_0010.png', 100)])
boltAnim.play()

while True:
    for event in pygame.event.get():
        if event.type == QUIT:
            pygame.quit()
            sys.exit()

    windowSurface.fill((100, 50, 50))
    boltAnim.blit(windowSurface, (100, 50))
    pygame.display.update()
```



Other examples exist in the `/examples` folder of the repo at https://github.com/asweigart/pyganim

CHAPTER 3

Play, Pause, Stop

PygAnimation objects have `play()`, `pause()`, and `stop()` methods. The

CHAPTER 4

Loading from Sprite Sheets

Sprite animations commonly come in "sprite sheet" images, such as this one:

Sprites sheets can be loaded into a PygAnimation object without having to slice the sheet up into individual image files. The sprite sheet's filename is passed to the `getImagesFromSpriteSheet()`, which returns a list of `pygame.Surface` objects.

All the individual images must be the same size. There are three ways to specify how to get the individual images from the sprite sheet:

- Two integers can be passed for the `width` and `height` parameters for the size of the individual cells. The order of the images in the returned list start at the top left, go right across the row, and then to the left side of the next row.
- Two integers can be passed for the `rows` and `cols` parameters for the number of rows and columns of images. The width and height are automatically calculated from the sprite sheet size. The order of the images in the returned list start at the top left, go right across the row, and then to the left side of the next row.
- A list of `(left, top, width, height)` tuples passed for the `rects` parameter for each image from the sprite sheet. The order of the images in the returned list are the same as the `rects` tuples.

Note that the return value from `getImagesFromSpriteSheet()` is just a list of `pygame.Surface` objects, but the `PygAnimation()` constructor requires a list of tuples: a `pygame.Surface` object and the duration of that frame in milliseconds. The built-in `zip()` function is useful for this:

```
>>> import pyganim
>>> images = pyganim.getImagesFromSpriteSheet(rows=1, cols=3)
>>> frames = list(zip(images, [200, 200, 600]))
>>> animObj = pyganim.PygAnimation(frames)
>>> animObj.play()
```

Note that in Python 3, `zip()` returns a "zip object" which must be converted into a list for `PygAnimation()`.

See the `examples/sprite_sheet_demo.py` program for an example of loading from a sprite sheet.

CHAPTER 5

Transformations

TODO

See, flip(), scale(), rotate(), rotozoom(), scale2x(), smoothscale(), convert(), convert_alpha(). When called on the PygAnimation object, they are applied to all the pygame.Surface objects in the animation. They do the same thing as the pygame.Surface methods of the same names.

CHAPTER 6

Unit Tests

The unit tests under /tests can be run from Python 2 or 3. In that folder are several test image files needed to run the tests (bolt1.png to bolt10.png, etc.)

```
> python basicTests.py
```

CHAPTER 7

Indices and tables

- genindex
- modindex
- search