Simple Python "Games" pt. 2

Skills/Concepts

This module is a continuation of the Python Games module.

This module will continue to build on the previous concepts of:

- SDL
- blitting
- Drawing APIs

In addition, this module will introduce classes, inheritence, and finite state machines.

Preparation

Continue with the same environment from the previous (Python Games) module.

We will be using an existing git repository that contains a fair amount of code instead of producing a new program from scratch. This is to avoid copy and paste errors and allow us to cover material faster.

To get the code, execute the following command:

```
git clone https://www.github.com/cyberpost500/modules/
```

Then extract the project repository to your desktop:

```
cd modules/python-games/
tar -C ~/Desktop/ -xf project.tar
```

The code will be in the Desktop/project directory.

Shapes

Background

In addition to blitting text and filling a region of pixels with a color, simple shapes are also possible. Simple shapes will help to form a basis for more complicated programming structures.

These APIs are demonstrated on the `cirlces` branch of the project repository. Switch to that branch with the following command:

```
git checkout circles
```

Review/Run/Modify the Code

0. View the Documentation for the program: ~~ pydoc game ~~

Press Q to exit when you're done.

1. Run the Program:

```
python game.py
```

2. Move the position where the circle is drawn:

```
pygame.draw.circle(screen, (0xff, 0xff, 0), (shape_x, shape_y), 10, 1
```

This is the line responsible for drawin the circle. The position where the circle is drawn is controlled by variables `shape_x` and `shape_y`.

Modify either the values passed on line 58 or change the initial values for `shape_x` and `shape_y` on line 43.

3. Save and run your program.

4. Add another shape to screen as well. Some other functions:

Function	Description
pygame.draw.rect	rectdraw a rectangle shape
pygame.draw.polygon	polygondraw a shape with any number of sides
pygame.draw.circle	circledraw a circle around a point
pygame.draw.ellipse	ellipsedraw a round shape inside a rectangle
pygame.draw.arc	arcdraw a partial section of an ellipse
pygame.draw.line	linedraw a straight line segment
pygame.draw.lines	linesdraw multiple contiguous line segments
pygame.draw.aaline	aalinedraw fine antialiased lines
pygame.draw.aalines	aalinesdraw a connected sequence of antialiased lines

Look up help for the function you want with pydoc:

```
pydoc pygame.draw.circle
```

Add the instruction for drawing the new shape directly beneath the instruction for drawing the existing circle.

5. Save and run your program.
6. Revert the file changes back

Undo your edits to the file using the following git command:

```
git checkout game.py
```

Conclusion

Now the program displays a circle and optionally any other shapes you may have put into it. The arguments used to make the circle are similatr to those used for the other primitive shapes, there shouldn't be anything too surprising. But, if the Python documentation is a little daunting, feel free to search online for other examples.

Animated Shapes

Background

Animated shapes via primative graphics is much like the old fashioned cartoon animations. A new image is drawn that is slightly different than the last, providing the illusion that the object has moved from one location to another. Accomplishing this with Python is as simple as changing the coordinates used to draw successive versions of the shape being animated.