
Python Arcade: The Manual

Python Arcade Academy

Apr 06, 2022

CONTENTS:

1	Application Window	1
2	Sprites	5
3	Sprite Lists	7

APPLICATION WINDOW

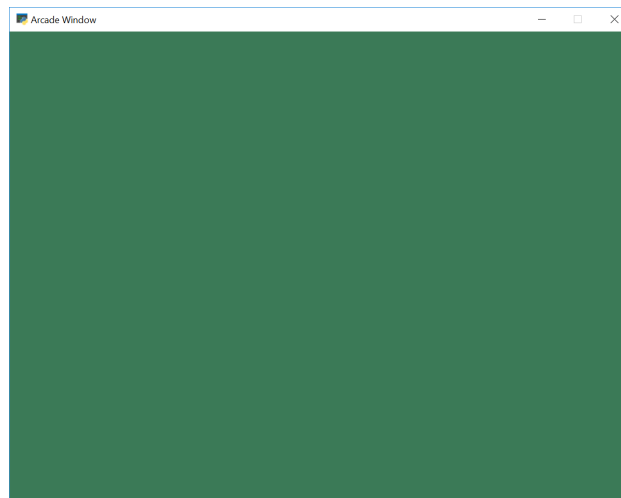


Fig. 1.1: Screenshot of the starting template.

See Fig. 1.1 for a screenshot of what this program looks like.

```
1  """
2  Starting Template
3
4  Once you have learned how to use classes, you can begin your program with this
5  template.
6
7  If Python and Arcade are installed, this example can be run from the command line with:
8  python -m arcade.examples.starting_template
9  """
10 import arcade
11
12 SCREEN_WIDTH = 800
13 SCREEN_HEIGHT = 600
14 SCREEN_TITLE = "Starting Template"
15
16
17 class MyGame(arcade.Window):
18     """
19     Main application class.
20
```

(continues on next page)

(continued from previous page)

```
21  NOTE: Go ahead and delete the methods you don't need.
22  If you do need a method, delete the 'pass' and replace it
23  with your own code. Don't leave 'pass' in this program.
24  """
25
26  def __init__(self, width, height, title):
27      super().__init__(width, height, title)
28
29      arcade.set_background_color(arcade.color.AMAZON)
30
31      # If you have sprite lists, you should create them here,
32      # and set them to None
33
34  def setup(self):
35      """ Set up the game variables. Call to re-start the game. """
36      # Create your sprites and sprite lists here
37      pass
38
39  def on_draw(self):
40      """
41      Render the screen.
42      """
43
44      # This command should happen before we start drawing. It will clear
45      # the screen to the background color, and erase what we drew last frame.
46      self.clear()
47
48      # Call draw() on all your sprite lists below
49
50  def on_update(self, delta_time):
51      """
52      All the logic to move, and the game logic goes here.
53      Normally, you'll call update() on the sprite lists that
54      need it.
55      """
56      pass
57
58  def on_key_press(self, key, key_modifiers):
59      """
60      Called whenever a key on the keyboard is pressed.
61
62      For a full list of keys, see:
63      https://api.arcade.academy/en/latest/arcade.key.html
64      """
65      pass
66
67  def on_key_release(self, key, key_modifiers):
68      """
69      Called whenever the user lets off a previously pressed key.
70      """
71      pass
72
```

(continues on next page)

(continued from previous page)

```
73 def on_mouse_motion(self, x, y, delta_x, delta_y):
74     """
75     Called whenever the mouse moves.
76     """
77     pass
78
79 def on_mouse_press(self, x, y, button, key_modifiers):
80     """
81     Called when the user presses a mouse button.
82     """
83     pass
84
85 def on_mouse_release(self, x, y, button, key_modifiers):
86     """
87     Called when a user releases a mouse button.
88     """
89     pass
90
91
92 def main():
93     """ Main function """
94     game = MyGame(SCREEN_WIDTH, SCREEN_HEIGHT, SCREEN_TITLE)
95     game.setup()
96     arcade.run()
97
98
99 if __name__ == "__main__":
100     main()
```

CHAPTER TWO

SPRITES

Don't forget to see [Chapter 3 *Sprite Lists*](#) for more information.

SPRITE LISTS

See [Chapter 2 Sprites](#) for more info about individual sprites.