```python
"""
CS241 Team Activity 05 - Stretch
Updated by Chad Macbeth
"""

"""
Starting Template

Once you have learned how to use classes, you can begin your program with this
template.

A walk-through of this code is available at:
https://vimeo.com/168051968

If Python and Arcade are installed, this example can be run from the command line with:
python -m arcade.examples.sprite_starting_template
"""

### If using pycharm, then need to install arcade package
### in the virtual envrionment.  Goto File | Settings | Project | Project Interpreter | Add
import arcade

SCREEN_WIDTH = 500
SCREEN_HEIGHT = 600
BALL_RADIUS = 40 # Updated from 20


class MyGame(arcade.Window):
    """
    Main application class.

    NOTE: Go ahead and delete the methods you don't need.
    If you do need a method, delete the 'pass' and replace it
    with your own code. Don't leave 'pass' in this program.
    """

    def __init__(self, width, height):
        super().__init__(width, height)

        self.ball_x_position = BALL_RADIUS
        self.ball_x_pixels_per_second = 70

        arcade.set_background_color(arcade.color.WHITE)
        self.draw_rectangle = False

        # Note:
        # You can change how often the animate() method is called by using the
        # set_update_rate() method in the parent class.
        # The default is once every 1/80 of a second.
        # self.set_update_rate(1/80)

    def on_draw(self):
        """
        Render the screen.
        """

        # This command should happen before we start drawing. It will clear
        # the screen to the background color, and erase what we drew last frame.
        arcade.start_render()

        # Draw the circle
        arcade.draw_circle_filled(self.ball_x_position, SCREEN_HEIGHT // 2,
                                  BALL_RADIUS, arcade.color.RED)  # Updated from Green

        # Draw the text
```

```
66          arcade.draw_text("This is a simple template to start your game.",
67                           10, SCREEN_HEIGHT // 2, arcade.color.BLACK, 20)
68
69          # Draw Rectangle in the center of the screen.
70          # The draw_rectangle variable will be set True when the space
71          # bar is being pressed.
72          if self.draw_rectangle:
73              arcade.draw_rectangle_filled(SCREEN_WIDTH // 2, SCREEN_HEIGHT // 2,
74                                           100, 100, arcade.color.BLUE)
75
76      def update(self, delta_time):
77          """
78          All the logic to move, and the game logic goes here.
79          """
80          # Move the ball
81          self.ball_x_position += self.ball_x_pixels_per_second * delta_time
82
83          # Did the ball hit the right side of the screen while moving right?
84          if self.ball_x_position > SCREEN_WIDTH - BALL_RADIUS \
85                  and self.ball_x_pixels_per_second > 0:
86              self.ball_x_pixels_per_second *= -1
87
88          # Did the ball hit the left side of the screen while moving left?
89          if self.ball_x_position < BALL_RADIUS \
90                  and self.ball_x_pixels_per_second < 0:
91              self.ball_x_pixels_per_second *= -1
92
93      def on_key_press(self, key, key_modifiers):
94          """
95          Called whenever a key on the keyboard is pressed.
96
97          For a full list of keys, see:
98          http://pythonhosted.org/arcade/arcade.key.html
99          """
100
101         # See if the user hit Shift-Space
102         # (Key modifiers are in powers of two, so you can detect multiple
103         # modifiers by using a bit-wise 'and'.)
104         if key == arcade.key.SPACE and key_modifiers == arcade.key.MOD_SHIFT:
105             print("You pressed shift-space")
106
107         # See if the user just hit space.
108         elif key == arcade.key.SPACE:
109             print("You pressed the space bar.")
110             self.draw_rectangle = True
111
112     def on_key_release(self, key, key_modifiers):
113         """
114         Called whenever the user lets off a previously pressed key.
115         """
116         if key == arcade.key.SPACE:
117             print("You stopped pressing the space bar.")
118             self.draw_rectangle = False
119
120     def on_mouse_motion(self, x, y, delta_x, delta_y):
121         """
122         Called whenever the mouse moves.
123         """
124         pass
125
126     def on_mouse_press(self, x, y, button, key_modifiers):
127         """
128         Called when the user presses a mouse button.
129         """
130         if self.ball_x_pixels_per_second >= 0:
131             self.ball_x_pixels_per_second += 10
```

```
132             else:
133                 self.ball_x_pixels_per_second -= 10
134 
135         def on_mouse_release(self, x, y, button, key_modifiers):
136             """
137             Called when a user releases a mouse button.
138             """
139             pass
140 
141 
142     def main():
143         """ Main method """
144         MyGame(SCREEN_WIDTH, SCREEN_HEIGHT)
145         arcade.run()
146 
147 
148     if __name__ == "__main__":
149         main()
```