

```
1: // Copyright 2024 Chris Lambert
2: #include <SFML/Graphics.hpp>
3: #include <SFML/Window/Keyboard.hpp>
4: #include "AnimatedSprite.hpp"
5:
6: int main() {
7:     sf::Vector2i screenDimensions(800, 600);
8:     sf::RenderWindow window(sf::VideoMode(screenDimensions.x, screenDimen
sions.y), "SFML works!");
9:     window.setFramerateLimit(60);
10:
11:     sf::CircleShape shape(100.f);
12:     shape.setFillColor(sf::Color::Green);
13:
14:     sf::Texture texture;
15:     if (!texture.loadFromFile("./sprite.png"))
16:         return EXIT_FAILURE;
17:
18:     sf::Sprite sprite(texture);
19:     Animation walkingAnimationDown;
20:     walkingAnimationDown.setSpriteSheet(texture);
21:     walkingAnimationDown.addFrame(sf::IntRect(32, 0, 32, 32));
22:     walkingAnimationDown.addFrame(sf::IntRect(64, 0, 32, 32));
23:     walkingAnimationDown.addFrame(sf::IntRect(32, 0, 32, 32));
24:     walkingAnimationDown.addFrame(sf::IntRect(0, 0, 32, 32));
25:
26:     Animation walkingAnimationLeft;
27:     walkingAnimationLeft.setSpriteSheet(texture);
28:     walkingAnimationLeft.addFrame(sf::IntRect(32, 32, 32, 32));
29:     walkingAnimationLeft.addFrame(sf::IntRect(64, 32, 32, 32));
30:     walkingAnimationLeft.addFrame(sf::IntRect(32, 32, 32, 32));
31:     walkingAnimationLeft.addFrame(sf::IntRect(0, 32, 32, 32));
32:
33:     Animation walkingAnimationRight;
34:     walkingAnimationRight.setSpriteSheet(texture);
35:     walkingAnimationRight.addFrame(sf::IntRect(32, 64, 32, 32));
36:     walkingAnimationRight.addFrame(sf::IntRect(64, 64, 32, 32));
37:     walkingAnimationRight.addFrame(sf::IntRect(32, 64, 32, 32));
38:     walkingAnimationRight.addFrame(sf::IntRect(0, 64, 32, 32));
39:
40:     Animation walkingAnimationUp;
41:     walkingAnimationUp.setSpriteSheet(texture);
42:     walkingAnimationUp.addFrame(sf::IntRect(32, 96, 32, 32));
43:     walkingAnimationUp.addFrame(sf::IntRect(64, 96, 32, 32));
44:     walkingAnimationUp.addFrame(sf::IntRect(32, 96, 32, 32));
45:     walkingAnimationUp.addFrame(sf::IntRect(0, 96, 32, 32));
46:
47:     Animation* currentAnimation = &walkingAnimationDown;
48:
49:     // set up AnimatedSprite
50:     AnimatedSprite animatedSprite(sf::seconds(0.2), true, false);
51:     animatedSprite.setPosition(sf::Vector2f(screenDimensions / 2));
52:
53:     sf::Clock frameClock;
54:
55:     float speed = 80.f;
56:     bool noKeyWasPressed = true;
57:
58:     while (window.isOpen()) {
59:         sf::Event event;
60:         while (window.pollEvent(event)) {
61:             if (event.type == sf::Event::Closed)
62:                 window.close();
63:         }
64:         sf::Time frameTime = frameClock.restart();
```

```
65:
66: // if a key was pressed set the correct animation and move correc
tly
67: sf::Vector2f movement(0.f, 0.f);
68: if (sf::Keyboard::isKeyPressed(sf::Keyboard::Up)) {
69:     currentAnimation = &walkingAnimationUp;
70:     movement.y -= speed;
71:     noKeyWasPressed = false;
72: }
73: if (sf::Keyboard::isKeyPressed(sf::Keyboard::Down)) {
74:     currentAnimation = &walkingAnimationDown;
75:     movement.y += speed;
76:     noKeyWasPressed = false;
77: }
78: if (sf::Keyboard::isKeyPressed(sf::Keyboard::Left)) {
79:     currentAnimation = &walkingAnimationLeft;
80:     movement.x -= speed;
81:     noKeyWasPressed = false;
82: }
83: if (sf::Keyboard::isKeyPressed(sf::Keyboard::Right)) {
84:     currentAnimation = &walkingAnimationRight;
85:     movement.x += speed;
86:     noKeyWasPressed = false;
87: }
88: animatedSprite.play(*currentAnimation);
89: animatedSprite.move(movement * frameTime.asSeconds());
90:
91: // if no key was pressed stop the animation
92: if (noKeyWasPressed) {
93:     animatedSprite.stop();
94: }
95: noKeyWasPressed = true;
96:
97: // update AnimatedSprite
98: animatedSprite.update(frameTime);
99:
100: window.clear();
101: window.draw(shape);
102: window.draw(animatedSprite);
103: window.display();
104: }
105: return 0;
106: }
```