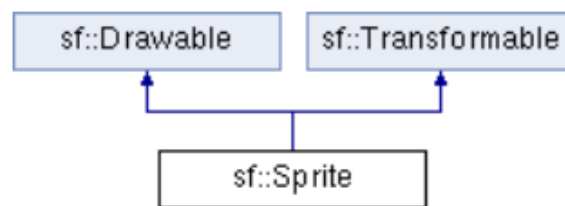




Design Advice: Architecture Advice

Exercise 1

Identify the layer or layers that each of the following SFML classes belong to.



```
sf::Sprite sprite(texture);
// Draw the sprite
window.draw(sprite);
```

Listing 1: Sprite

```
sf::Event event;
while (window.pollEvent(event))
{
    // Window closed or escape key pressed: exit
    if((event.type == sf::Event::Closed) || ((event.type ==
        sf::Event::KeyPressed) && (event.key.code == sf::Keyboard::Escape)))
    {
        window.close();
        break;
    }
    // ...
}
```

Listing 2: Event

```

// sf::Texture stores pixels that can be drawn, with a sprite for example.
// It is an Image living on the graphics card that can be used for drawing.

// Load a sprite to display
sf::Texture texture;
if (!texture.loadFromFile("cute_image.jpg"))
return EXIT_FAILURE;

```

Listing 3: Texture

```

sf::Clock AITimer;
const sf::Time AITime = sf::seconds(0.1f);
if (AITimer.getElapsedTime() > AITime)
{
    // ...
}

```

Listing 4: Clock

```

// Create the main window
sf::RenderWindow window(sf::VideoMode(800, 600), "SFML window");

// Clear screen
window.clear();

```

Listing 5: RenderWindow

```

// Create the ball
sf::CircleShape ball;
ball.setRadius(ballRadius - 3);
ball.setOutlineThickness(3);
ball.setOutlineColor(sf::Color::Black);
ball.setFillColor(sf::Color::White);
ball.setOrigin(ballRadius / 2, ballRadius / 2);

```

Listing 6: CircleShape

```

sf::Vector2f paddleSize(25, 100);

//...

leftPaddle.setPosition(10 + paddleSize.x / 2, gameHeight / 2);

```

Listing 7: Vector2f

In what ways does SFML allow, and perhaps even encourage, the merging of application layers (presentation, domain, and data access layers)?