

```
1 using SplashKitSDK;
2 using DragonBallGame.CharacterClass;
3
4 namespace DragonBallGame
5 {
6     public class BattleUI
7     {
8         private Window _window;
9         private ImageManager _imageManager;
10
11         public Window Window => _window;
12
13         public BattleUI(Window window, ImageManager imageManager)
14         {
15             _window = window;
16             _imageManager = imageManager;
17         }
18
19         public void DrawBattleState(Character player, Character villain)
20         {
21             // Draw Character Images
22             Bitmap playerImage = _imageManager.GetCharacterImage(player);
23             Bitmap villainImage = _imageManager.GetCharacterImage(villain);
24
25             SplashKit.DrawBitmap(playerImage, 50, 130); // Player image on
26                 the left
27             SplashKit.DrawBitmap(villainImage, _window.Width - 250,
28                 150); // Villain image on the right
29
30             // Draw Health Bars and Energy Bars
31             DrawHealthBar(50, 80, player.Health, player.MaxHealth,
32                 Color.Green, $"{player.Name}");
33             DrawEnergyBar(50, 110, player.Energy, 100, Color.Cyan);
34
35             DrawHealthBar(_window.Width - 250, 80, villain.Health,
36                 villain.MaxHealth, Color.Red, $"{villain.Name}");
37             DrawEnergyBar(_window.Width - 250, 110, villain.Energy, 100,
38                 Color.Cyan);
39
40             SplashKit.DrawText("Press SPACE to attack, B to block, E to
41                 escape", Color.Red, 220, 30);
42         }
43
44         private void DrawHealthBar(int x, int y, int currentHealth, int
45             maxHealth, Color barColor, string name)
46         {
47             int barWidth = 200;
48             int barHeight = 20;
49             double healthPercentage = (double)currentHealth / maxHealth;
```

```
43         int currentBarWidth = (int)(barWidth * healthPercentage);
44
45         // Draw health bar background (grey)
46         SplashKit.FillRectangle(Color.Gray, x, y, barWidth, barHeight);
47
48         // Draw current health (colored bar)
49         SplashKit.FillRectangle(barColor, x, y, currentBarWidth, barHeight);
50
51         // Draw character name above health bar
52         SplashKit.DrawText(name, Color.Black, x, y - 20);
53     }
54
55     private void DrawEnergyBar(int x, int y, int currentEnergy, int maxEnergy, Color barColor)
56     {
57         int barWidth = 200;
58         int barHeight = 10;
59         double energyPercentage = (double)currentEnergy / maxEnergy;
60         int currentBarWidth = (int)(barWidth * energyPercentage);
61
62         // Draw energy bar background (grey)
63         SplashKit.FillRectangle(Color.Gray, x, y, barWidth, barHeight);
64
65         // Draw current energy (colored bar)
66         SplashKit.FillRectangle(barColor, x, y, currentBarWidth, barHeight);
67     }
68 }
69 }
```