```cpp
#include "Healthbar.h"
#include "Window.h"
#include "Mixer.h"

enum stats_attribute { HEALTH, MAX_HEALTH, PERCENTAGE };

Healthbar::Healthbar(bool left) {
    int window_size_h = Window::getInstance()->getWindowSizeH();
    int window_size_w = Window::getInstance()->getWindowSizeW();

    //Set up attributes
    appearance.w = window_size_w / 5;
    appearance.h = window_size_h / 35;
    if (left) {
        appearance.x = window_size_w / 20;
        appearance.y = (window_size_h / 20);
    }
    else {
        appearance.x = window_size_w - (appearance.w + (window_size_w / 20));
        appearance.y = (window_size_h / 20);
    }
    this->left = left;
    stats[HEALTH] = 1000.0;
    stats[MAX_HEALTH] = 1000.0;
    stats[PERCENTAGE] = 100.0;
    color.r = 0x00;
    color.g = 0xFF;
    color.b = 0x00;
    color.a = 0xA0;
};

Healthbar::~Healthbar() {};

void Healthbar::render() {
    SDL_Renderer* renderer = Window::getInstance()->getRenderer();

    //Saving the old rendercolor
    SDL_Color old_color;
    SDL_GetRenderDrawColor(renderer, &old_color.r, &old_color.g, &old_color.b, &old_color.a);

    SDL_Rect colored_rect, black_rect;
    colored_rect.x = appearance.x;
    colored_rect.y = appearance.y;
    colored_rect.w = (double)appearance.w * (stats[PERCENTAGE] / 100.0);
    colored_rect.h = appearance.h;

    black_rect.x = appearance.x + colored_rect.w;
    black_rect.y = appearance.y;
    black_rect.w = appearance.w - colored_rect.w;
    black_rect.h = appearance.h;

```

```
52      SDL_SetRenderDrawBlendMode(renderer, SDL_BLENDMODE_BLEND);
53
54      if (left) {
55          //Draw Colored Rect
56          SDL_SetRenderDrawColor(renderer, color.r, color.g, color.b,
              color.a);
57          SDL_RenderFillRect(renderer, &colored_rect);
58
59          //Draw Black Rect
60          SDL_SetRenderDrawColor(renderer, 0x00, 0x00, 0x00, color.a);
61          SDL_RenderFillRect(renderer, &black_rect);
62      }
63      else {
64          black_rect.x = appearance.x;
65          colored_rect.x = appearance.x + black_rect.w;
66
67          //Draw Black Rect
68          SDL_SetRenderDrawColor(renderer, 0x00, 0x00, 0x00, color.a);
69          SDL_RenderFillRect(renderer, &black_rect);
70
71          //Draw Colored Rect
72          SDL_SetRenderDrawColor(renderer, color.r, color.g, color.b,
              color.a);
73          SDL_RenderFillRect(renderer, &colored_rect);
74      }
75
76      //Reset the old rendercolor
77      SDL_SetRenderDrawColor(renderer, old_color.r, old_color.g,
          old_color.b, old_color.a);
78  };
79
80  bool Healthbar::takeDamage(double damage) {
81      bool dead = false;
82
83      if (damage >= stats[HEALTH]) {
84          stats[HEALTH] = 0.0;
85          stats[PERCENTAGE] = 0.0;
86          setColor();
87          dead = true;
88          Mixer::getInstance()->play(Mixer::DYING);
89          printf("DYING");
90      }
91      else {
92          stats[HEALTH] -= damage;
93          stats[PERCENTAGE] -= ((damage / stats[MAX_HEALTH]) * 100);
94
95          //Set Color
96          double tmp = (510.0 * (stats[PERCENTAGE] / 100.0)) - 255.0;
97          if (tmp >= 0) {
98              setColor();
99          }
100         else {
101             setColor();
```

```
102         }
103     }
104
105     return dead;
106 };
107
108 bool Healthbar::isEmpty() {
109     bool empty = false;
110
111     if (stats[HEALTH] <= 0) {
112         empty = true;
113     }
114
115     return empty;
116 };
117
118 void Healthbar::refill() {
119     stats[HEALTH] = stats[MAX_HEALTH];
120     stats[PERCENTAGE] = 100.0;
121     setColor();
122 };
123
124 void Healthbar::setColor() {
125     //Set Color
126     double tmp = (510.0 * (stats[PERCENTAGE] / 100.0)) - 255.0;
127     if (tmp >= 0) {
128         color.r = 0xFF - tmp;
129         color.g = 0xFF;
130     }
131     else {
132         color.r = 0xFF;
133         color.g = 0xFF+tmp;
134     }
135 };
```