```
1 #include "MenuTexture.h"
 2 #include "Window.h"
 3
4
  MenuTexture::MenuTexture(std::string text, std::string fontpath, int
     fontsize, SDL Color color) {
 5
       texture = NULL;
       this->text = text;
 6
 7
       this->fontpath = fontpath;
       this->color = color;
9
       this->appearance = {0,0,0,0};
10
       this->fontsize = fontsize;
11 };
12
13 SDL Texture* MenuTexture::loadTextTexture() {
14
       SDL_Texture* text_tx = NULL;
15
       //Render text surface
16
17
       SDL_Surface* text_surface = TTF_RenderText_Solid(font, text.c_str(),
          color);
18
       if (text surface != NULL)
19
            //Create texture from surface pixels
20
21
            text_tx = SDL_CreateTextureFromSurface(Window::getInstance()-
              >getRenderer(), text_surface);
22
            if (text tx == NULL)
23
            {
24
                printf("Failed to create texture from rendered text. SDL
                  Error: %s\n", SDL_GetError());
25
           }
26
           //Get Correct Size
27
            appearance.w = text surface->w;
28
           appearance.h = text_surface->h;
29
           //Get rid of old surface
30
31
           SDL_FreeSurface(text_surface);
32
           text_surface = NULL;
33
       }
       else
34
35
       {
36
            printf("Failed to render text surface. SDL_ttf Error: %s\n",
              TTF_GetError());
37
       }
38
39
       return text_tx;
40 };
41
42
   bool MenuTexture::loadMedia() {
43
       bool success = true;
44
45
       //Open the font
46
       if (setFont(fontpath.c str()) == true)
47
       {
48
           //Color is already set
```

```
C:\Dev\FinalProject\ClassFiles\MenuTexture.cpp
```

```
2
```

```
49
            //Render text
50
            texture = loadTextTexture();
51
            if (texture == NULL)
52
                 printf("Failed to render texture for text.\n");
53
54
                 success = false;
55
            }
        }
56
57
        else
58
        {
59
            success = false;
60
        }
61
62
        return success;
63 };
64
65 void MenuTexture::render() {
66
        //Set Renderer
        SDL_Renderer* renderer = Window::getInstance()->getRenderer();
67
68
        //Render
69
        SDL_RenderCopy(renderer, texture, NULL, &appearance);
70
71 };
72
73 void MenuTexture::tick() {};
74
75 void MenuTexture::setColor(Uint8 r, Uint8 g, Uint8 b, Uint8 a) {
76
        //Set Color
77
        color.r = r;
        color.g = g;
78
79
        color.b = b;
80
        color.a = a;
81 };
82
83 bool MenuTexture::setFont(std::string path) {
        bool success = true;
84
85
        //Free Font
86
87
        if (font != NULL) {
88
            TTF CloseFont(font);
            font = NULL;
89
90
        }
91
92
        //Open the font
93
        font = TTF_OpenFont(path.c_str(), fontsize);
        if (font == NULL)
94
95
        {
96
            printf("Failed to load font. SDL_ttf Error: %s\n", TTF_GetError
               ());
97
            success = false;
98
        }
99
100
        return success;
```

```
C:\Dev\FinalProject\ClassFiles\MenuTexture.cpp
```

```
3
```

```
<del>101</del> };
102
103 void MenuTexture::setX(int x) {
104
        appearance.x = x;
105 };
106
107 void MenuTexture::setY(int y) {
108
        appearance.y = y;
109 };
110
111 int MenuTexture::getX() {
        return appearance.x;
113 };
114
115 int MenuTexture::getY() {
116
        return appearance.y;
117 };
118
119 int MenuTexture::getW() {
120
        return appearance.w;
121 }
122
123 int MenuTexture::getH() {
124
        return appearance.h;
125 };
126
127 std::string MenuTexture::getType() {
128
       return "TEXTURE";
129 };
130
131 void MenuTexture::free() {
132
        //Free texture
133
        SDL_DestroyTexture(texture);
        texture = NULL;
134
135
136
       //Free font
        TTF_CloseFont(font);
137
138
        font = NULL;
139 };
140
141 MenuTexture::~MenuTexture() {
142
        //free
143
        free();
144 };
```