

```
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;
6     this->text = text;
7     this->fontpath = fontpath;
8     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
16     //Render text surface
17     SDL_Surface* text_surface = TTF_RenderText_Solid(font, text.c_str(),      ↗
        color);
18     if (text_surface != NULL)
19     {
20         //Create texture from surface pixels
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
30         //Get rid of old surface
31         SDL_FreeSurface(text_surface);
32         text_surface = NULL;
33     }
34     else
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
```

```
49     //Render text
50     texture = loadTextTexture();
51     if (texture == NULL)
52     {
53         printf("Failed to render texture for text.\n");
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
67     SDL_Renderer* renderer = Window::getInstance()->getRenderer();
68
69     //Render
70     SDL_RenderCopy(renderer, texture, NULL, &appearance);
71 };
72
73 void MenuTexture::tick() {};
74
75 void MenuTexture::setColor(uint8_t r, uint8_t g, uint8_t b, uint8_t a) {
76     //Set Color
77     color.r = r;
78     color.g = g;
79     color.b = b;
80     color.a = a;
81 };
82
83 bool MenuTexture::setFont(std::string path) {
84     bool success = true;
85
86     //Free Font
87     if (font != NULL) {
88         TTF_CloseFont(font);
89         font = NULL;
90     }
91
92     //Open the font
93     font = TTF_OpenFont(path.c_str(), fontsize);
94     if (font == NULL)
95     {
96         printf("Failed to load font. SDL_ttf Error: %s\n", TTF_GetError
97             ());
98         success = false;
99     }
100     return success;
```

```
101 };
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() {
112     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);
134     texture = NULL;
135
136     //Free font
137     TTF_CloseFont(font);
138     font = NULL;
139 };
140
141 MenuTexture::~MenuTexture() {
142     //free
143     free();
144 };
```