```cpp
#include "MenuTexture.h"
#include "Window.h"

MenuTexture::MenuTexture(std::string text, std::string fontpath, int fontsize, SDL_Color color) {
    texture = NULL;
    this->text = text;
    this->fontpath = fontpath;
    this->color = color;
    this->appearance = {0,0,0,0};
    this->fontsize = fontsize;
};

SDL_Texture* MenuTexture::loadTextTexture() {
    SDL_Texture* text_tx = NULL;

    //Render text surface
    SDL_Surface* text_surface = TTF_RenderText_Solid(font, text.c_str(), color);
    if (text_surface != NULL)
    {
        //Create texture from surface pixels
        text_tx = SDL_CreateTextureFromSurface( Window::getInstance()->getRenderer(), text_surface);
        if (text_tx == NULL)
        {
            printf("Failed to create texture from rendered text. SDL Error: %s\n", SDL_GetError());
        }
        //Get Correct Size
        appearance.w = text_surface->w;
        appearance.h = text_surface->h;

        //Get rid of old surface
        SDL_FreeSurface(text_surface);
        text_surface = NULL;
    }
    else
    {
        printf("Failed to render text surface. SDL_ttf Error: %s\n" , TTF_GetError());
    }

    return text_tx;
};

bool MenuTexture::loadMedia() {
    bool success = true;

    //Open the font
    if (setFont(fontpath.c_str()) == true)
    {
        //Color is already set
        //Render text
        texture = loadTextTexture();
        if (texture == NULL)
        {
            printf("Failed to render texture for text.\n" );
            success = false;
        }
    }
    else
    {
        success = false;
    }

    return success;
};

void MenuTexture::render() {
    //Set Renderer
    SDL_Renderer* renderer = Window::getInstance()->getRenderer();

    //Render
    SDL_RenderCopy(renderer, texture, NULL, &appearance);
};

void MenuTexture::tick() {};

void MenuTexture::setColor(Uint8 r, Uint8 g, Uint8 b, Uint8 a) {
    //Set Color
    color.r = r;
    color.g = g;
    color.b = b;
    color.a = a;
};

bool MenuTexture::setFont(std::string path) {
    bool success = true;

    //Free Font
    if (font != NULL) {
        TTF_CloseFont(font);
        font = NULL;
    }

    //Open the font
    font = TTF_OpenFont(path.c_str(), fontsize);
    if (font == NULL)
    {
        printf("Failed to load font. SDL_ttf Error: %s\n" , TTF_GetError());
        success = false;
    }

    return success;
```

```cpp
};

void MenuTexture::setX(int x) {
    appearance.x = x;
};

void MenuTexture::setY(int y) {
    appearance.y = y;
};

int MenuTexture::getX() {
    return appearance.x;
};

int MenuTexture::getY() {
    return appearance.y;
};

int MenuTexture::getW() {
    return appearance.w;
}

int MenuTexture::getH() {
    return appearance.h;
};

std::string MenuTexture::getType() {
    return "TEXTURE";
};

void MenuTexture::free() {
    //Free texture
    SDL_DestroyTexture(texture);
    texture = NULL;

    //Free font
    TTF_CloseFont(font);
    font = NULL;
};

MenuTexture::~MenuTexture() {
    //free
    free();
};
```