

Một cách để render văn bản với SDL là dùng thư viện mở rộng SDL\_ttf. SDL\_ttf cho phép bạn tạo ảnh từ Truetype fonts, cái mà chúng ta sẽ dùng ở đây là tạo texture từ font text.

|  |
| --- |
| //Using SDL, SDL\_image, SDL\_ttf, standard IO, math, and strings  #include <SDL.h>  #include <SDL\_image.h>  #include <SDL\_ttf.h>  #include <stdio.h>  #include <string>  #include <cmath> |

* Trước khi dùng SDL\_ttf nhớ phải set up như khi set up SDL\_image. Như trước, chỉ cần có header files, library files và binary files ở đúng chỗ với compiler.

|  |
| --- |
| //Texture wrapper class  class LTexture  {      public:          //Initializes variables          LTexture();          //Deallocates memory          ~LTexture();          //Loads image at specified path          bool loadFromFile( std::string path );    **//Creates image from font string**  **bool loadFromRenderedText( std::string textureText, SDL\_Color textColor );**          //Deallocates texture          void free();          //Set color modulation          void setColor( Uint8 red, Uint8 green, Uint8 blue );          //Set blending          void setBlendMode( SDL\_BlendMode blending );          //Set alpha modulation          void setAlpha( Uint8 alpha );            //Renders texture at given point          void render( int x, int y, SDL\_Rect\* clip = NULL, double angle = 0.0, SDL\_Point\* center = NULL, SDL\_RendererFlip flip = SDL\_FLIP\_NONE );          //Gets image dimensions          int getWidth();          int getHeight();      private:          //The actual hardware texture          SDL\_Texture\* mTexture;          //Image dimensions          int mWidth;          int mHeight;  }; |

* Ở đây chúng ta thêm một hàm khác vào texture class, gọi là loadFromRenderedText. Cách mà SDL\_ttf hoạt động là bạn tạo ảnh từ một font và màu. Với texture class của chúng ta, tất cả điều đó nghĩa là chúng ta sẽ load ảnh từ text được render bơi SDL\_ttf thay vì một file.

|  |
| --- |
| //The window we'll be rendering to  SDL\_Window\* gWindow = NULL;  //The window renderer  SDL\_Renderer\* gRenderer = NULL;  //Globally used font  TTF\_Font \*gFont = NULL;  //Rendered texture  LTexture gTextTexture; |

* Từ nay, chúng ta sẽ sử dụng một font cho toàn bộ việc render văn bản của chúng ta. Trong SDL\_ttf, kiểu dữ liệu cho fonts là TTF\_Font .
* Chúng ta cũng có một texture, cái sẽ được sinh ra từ font.

|  |
| --- |
| bool LTexture::loadFromRenderedText( std::string textureText, SDL\_Color textColor )  {      //Get rid of preexisting texture      free();      //Render text surface      SDL\_Surface\* textSurface = TTF\_RenderText\_Solid( gFont, textureText.c\_str(), textColor );      if( textSurface == NULL )      {          printf( "Unable to render text surface! SDL\_ttf Error: %s\n", TTF\_GetError() );      }      else      {          //Create texture from surface pixels          mTexture = SDL\_CreateTextureFromSurface( gRenderer, textSurface );          if( mTexture == NULL )          {              printf( "Unable to create texture from rendered text! SDL Error: %s\n", SDL\_GetError() );          }          else          {              //Get image dimensions              mWidth = textSurface->w;              mHeight = textSurface->h;          }          //Get rid of old surface          SDL\_FreeSurface( textSurface );      }        //Return success      return mTexture != NULL;  } |

* Đây là nơi chúng ra thực sự tạo text texture mà chúng ta sẽ render từ font. Hàm này lấy string của văn bản chúng ta muốn render và màu chúng ta dùng để render nó. Sau đó, hàm này hoạt động khá giống như load từ một file, chỉ khác là chúng ta sử dụng SDL\_Surface tạo bơi SDL\_ttf thay vì là file.
* Sau khi giải phóng toàn bộ texture tồn tại trước đó, chúng ta load một surface sử dụng TTF\_RenderText\_Solid(). Hàm này tạo ra một surface màu đồng nhất từ font, chữ và màu đã cho. Nếu surface được tạo thành công, chúng ta sẽ tạo texture từ nó như trước, lúc mà ta load surface từ file. Sau khi text texture được tạo, chúng ta có thể render với nó như những texture khác.
* Có nhiều cách khác để render text mà mượt hơn hay được pha trộn. Trải nghiệm với nhiều kiểu render được tóm tắt trên trang của SDL. Render được đổ bóng hay pha trộn có thể hoạt động tốt hơn trên nhiều size chữ.

|  |
| --- |
| //Initialize PNG loading  int imgFlags = IMG\_INIT\_PNG;  if( !( IMG\_Init( imgFlags ) & imgFlags ) )  {      printf( "SDL\_image could not initialize! SDL\_image Error: %s\n", IMG\_GetError() );      success = false;  }  **//Initialize SDL\_ttf**  **if( TTF\_Init() == -1 )**  **{**  **printf( "SDL\_ttf could not initialize! SDL\_ttf Error: %s\n", TTF\_GetError() );**  **success = false;**  **}** |

* Cũng như SDL\_image, chúng ta phải khởi tạo nó, nếu không chức năng load font chữ hay render sẽ khong hoạt động bình thường. Chúng ta khởi động SDL\_ttf sử dụng TTF\_Init(). Chúng ta có thể kiểm tra lỗi dùng TTF\_GetError().

|  |
| --- |
| bool loadMedia()  {      //Loading success flag      bool success = true;      //Open the font      gFont = TTF\_OpenFont( "16\_true\_type\_fonts/lazy.ttf", 28 );      if( gFont == NULL )      {          printf( "Failed to load lazy font! SDL\_ttf Error: %s\n", TTF\_GetError() );          success = false;      }      else      {          //Render text          SDL\_Color textColor = { 0, 0, 0 };          if( !gTextTexture.loadFromRenderedText( "The quick brown fox jumps over the lazy dog", textColor ) )          {              printf( "Failed to render text texture!\n" );              success = false;          }      }      return success;  } |

* Trong hàm load, chúng ta load font của chúng ta sử dụng TTF\_OpenFont(). Hàm này lấy đường dẫn từ font file và kích thước điểm mà ta muốn render.
* Nếu font load được thành công, chúng ta sẽ load text texture sử dụng cách load của chúng ta. Theo quy tắc thông thường, bạn sẽ minimize số lần bạn render text. Chỉ render nó khi bạn cần và vì chúng ta dùng cùng một text surface cho toàn bộ chương trình nên chúng ta chỉ muốn render một lần.

|  |
| --- |
| void close(){      //Free loaded images      gTextTexture.free();      //Free global font      TTF\_CloseFont( gFont );      gFont = NULL;      //Destroy window      SDL\_DestroyRenderer( gRenderer );      SDL\_DestroyWindow( gWindow );      gWindow = NULL;      gRenderer = NULL;      //Quit SDL subsystems      TTF\_Quit();      IMG\_Quit();      SDL\_Quit();  } |

* Trong hàm clean up, chúng ta sẽ giải phóng font sử dụng TTF\_CloseFont(). Chúng ta cũng sẽ đóng thư viện SDL\_ttf với TTF\_Quit() để hoàn tất clean up.

|  |
| --- |
| //While application is running  while( !quit ){      //Handle events on queue      while( SDL\_PollEvent( &e ) != 0 ){          //User requests quit          if( e.type == SDL\_QUIT ){              quit = true;          }      }      //Clear screen      SDL\_SetRenderDrawColor( gRenderer, 0xFF, 0xFF, 0xFF, 0xFF );      SDL\_RenderClear( gRenderer );      //Render current frame      gTextTexture.render( ( SCREEN\_WIDTH - gTextTexture.getWidth() ) / 2, ( SCREEN\_HEIGHT - gTextTexture.getHeight() ) / 2 );      //Update screen      SDL\_RenderPresent( gRenderer );  } |

* Như bạn có thể thấy, sau khi chúng ta render text texture, chúng ta có thể render nó như texture khác.