**sgl::font**

O pacote sgl::font é o responsável pela parte textual da biblioteca. Ele carrega uma fonte no formato TTF (“True Type Font”) padrão do Windows e a utiliza para escrever um texto na tela. É constituído de duas classes: a classe *FontResource*, especialização da classe *Resource* e a classe *Font*.

A *FontResource* é usada dentro do método load da classe *Font* e funciona da seguinte maneira: recebe um arquivo TTF e um tamanho e, caso ainda não tenha sido colocado, esse arquivo é adicionado em uma instância da classe *ResourceManager*. É possível existir dois arquivos de fonte iguais mas com tamanhos diferentes em *ResourceManager*.

*Font* possui métodos para escrever na tela e mudar as configurações do texto, tais como cor, alinhamento e posição. O pacote utiliza também as funcionalidades das classes *Vector2D*, *Color* e *BoundingBox*. Com os métodos gets temos informações sobre as configurações atuais do objeto. Abaixo um exemplo de uso da classe:

**Color\* c1 = new Color( 200, 0, 0 );**

Instância de um objeto c1 ponteiro de *Color*, variando somente o parâmetro referente à cor vermelha do sistema RGB.

**Vector2D\* v1 = new Vector2D( 320, 120 );**

Instância de um objeto v1 ponteiro de *Vector2D*, passando as coordenadas horizontal e vertical, respectivamente.

**Font\* f1 = new Font( "alger.ttf", 50 );**

Instância de um objeto f1 ponteiro de *Font*, que recebe o nome do arquivo de fonte e o seu tamanho.

**f1->setColorFont( \*c1 );**

**f1->setPosition( \*v1 );**

**f1->setText( "Fonte: Alger" );**

**f1->setAlignment( FontAlignment::CENTRE );**

Faz com que f1 esteja na cor dada por c1 e na posição dada por v1. Altera também o texto e o alinhamento, que pode ser dos tipos LEFT, RIGHT, CENTRE e INTEGER.

**f1->drawText();**

Desenha o texto na tela.

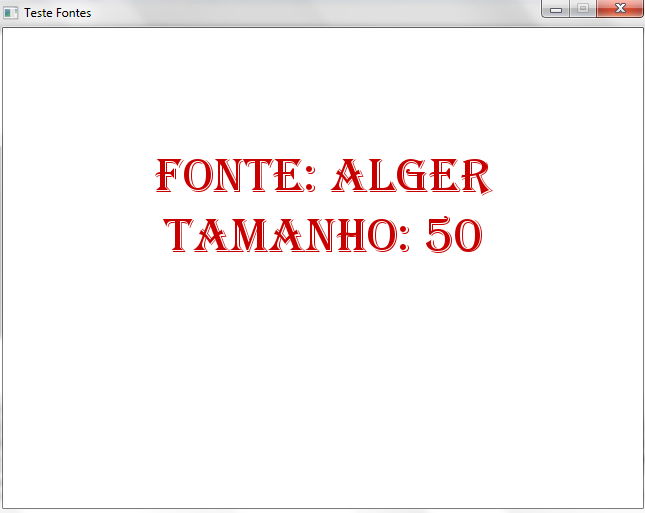
**f1->setText( "Tamanho: 50" );**

**v1->setCoordenate( 320, 180 );**

**f1->setPosition( \*v1 );**

**f1->drawText();**

Altera o texto e a posição na tela e desenha novamente. O trecho de código produz uma saída como na imagem abaixo.



**sgl::audio**

É o pacote que realiza as funcionalidades referentes à parte sonora. É estruturado da seguinte maneira: de um lado a classe *AudioResource,* especialização de *Resource,* e da qual derivam *AudioStreamResource*  e *AudioSampleResource.* Do outro lado, temos a classe abstrata *Audio,* e da qual derivam *AudioStream*  e *AudioSample.* As duas últimas relacionam-se com as respectivas especializações de *AudioResource.* Tal estrutura nos possibilita criar um objeto de *AudioSample* quando desejamos manipular um efeito sonoro, como um soco ou o barulho de uma explosão, ou um objeto de *AudioStream* quando precisamos usar uma trilha sonora.

*AudioSample* e *AudioStream* possuem em comum métodos para carregar, tocar, alterar ganho, balanço, velocidade e modo de repetição. Além desses cada classe possui outros métodos com funções específicas. É importante saber que o método para alterar a velocidade de um arquivo de áudio também altera sua frequência. Seu uso inadequado pode tornar o áudio incompreensível. Não existe suporte ao formato mp3. Exemplo:

**AudioSample\* sample = new AudioSample( "palmas.wav" );**

Cria-se um objeto ponteiro para *AudioSample.*

**sample->setGain( 0.8 );**

**sample->setSpeed( 0.9 );**

Altera o ganho para 80% do valor original e a velocidade para 90% do valor original.

**sample->setLoopingMode( AudioPlayMode::PLAY\_LOOP );**

Faz com que o arquivo continue tocando indefinidamente após chamar o método play().

**sample->play();**

Finalmente, coloca o arquivo para tocar.

**AudioStream\* stream = new AudioStream( "interface.ogg", 4, 1024 );**

Carrega um arquivo para ser reproduzido como trilha sonora e recebe parâmetros para alterar o buffer (em bytes) e a quantidade de amostras.

**stream->setBegin( 14 );**

**stream->play();**

Faz o arquivo tocar a partir dos 14 segundos.