Indicações para o carregamento de texturas em Java/JOGL.

Muitos dos formatos de imagem usados frequentemente guardam a informação recorrendo a técnicas de compressão mais ou menos sofisticadas. Ainda por cima, esses mesmos ficheiros possuem zonas com meta-informação, relevante para o processo de descompressão ou carregamento dessas mesmas imagens.

Para podermos usar uma imagem como uma textura em OpenGL, é necessário termos uma zona de memória contígua com a informação não comprimida dos pixels dessa mesma imagem. Por exemplo, se a imagem for a cores, cada pixel será representado por 3 bytes consecutivos com a informação dos valores de R,G e de B. Todos os pixels da linha inferior da imagem deverão surgir em primeiro lugar, da esquerda para a direita, seguindo-se os pixels da segunda linha e assim sucessivamente.

Felizmente, não será necessário desenvolver todo o processo a partir da estaca zero, bastando para tal usar adequadamente as classes oferecidas pelo SDK da linguagem Java. Uma possível abordagem passa pelo uso da classe ImageIO, tal como se ilustra no exemplo:

```
File _textureFile = …; // returned by JFileChooser getSelectedFile() method

BufferedImage img = ImageIO.read(_textureFile);
WritableRaster raster = img.getRaster();
int width = raster.getWidth();
int height = raster.getHeight();
DataBuffer buf = raster.getDataBuffer();
switch( buf.getDataType() ) {
      case DataBuffer.TYPE_BYTE:
            DataBufferByte bb = (DataBufferByte) buf;
            byte im[] = bb.getData();

            // other glTexParameter() calls needed here!

            _gl.glTexImage2D(
                  GL.GL_TEXTURE_2D, 0, GL.GL_RGB, width, height,
                  0, GL.GL_BGR, GL.GL_UNSIGNED_BYTE, ByteBuffer.wrap(im));
            break;
      case DataBuffer.TYPE_UNDEFINED:
            // Report error here!
            break;
}
```

O código neste exemplo requer que o contexto OpenGL, dado pelo objeto **gl** seja váildo, o que não acontece em todos os locais do código. Na realidade, o objeto **gl** é obtido dentro de qualquer das funções: *display*, *displayChanged*, *init* ou *reshape*. Fora dessas funções não temos acesso às funções do OpenGL. Isto significa que o código aqui apresentado não poderá ser executado em resposta direta ao evento gerado pelo utilizador na altura em que manda carregar o ficheiro com a textura. A solução passa por deferir o carregamento dessa mesma textura para a próxima vez que o método *display* for invocado.