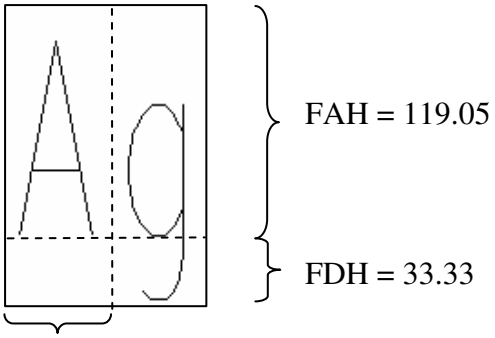
Stroke text using GLUT

Draw a character (ASCII 32 - 127) as a set of lines:

proportionally spaced: `glutStrokeCharacter( GLUT_STROKE_ROMAN, 'A' );`
fixed width: `glutStrokeCharacter( GLUT_STROKE_MONO_ROMAN, '$' );`

GLUT stroke characters are 119.05 pixels from baseline to top ("font ascent height"), 33.33 pixels from baseline to bottom of a descender ("font descent height). A character's width in pixels ("font advance width") is given by:

```
int glutStrokeWidth(GLUTstrokeFont font, int character);
```



For the mono-spaced font, this is a constant 104.76 pixels wide

To position / size a character, use

`glTranslatef / glScalef`

Draw a single character: walkthrough **GLUTStrokeText1**

Why is push/pop of the modelview matrix stack needed?

Modify so viewport is 2 characters wide - draw 2 characters

How to draw BOLD text?

Multiple lines of text: walkthrough **GLUTStrokeText2**

set up a view for drawing text,
draw the text, then
restore the previous view (reshape is quick & dirty)

Multiple lines of text: a better way:

1. **save** the current viewport
2. **set up a viewport for drawing text**
3. **save** the current projection
4. set up a 2D orthographic projection
5. **save** the current modelview matrix
6. position the first character
7. draw all of the characters in the string
8. **restore** the previous modelview matrix
9. **restore** the previous projection matrix
10. **restore** the original viewport

(walkthrough **GLUTStrokeText3**)

```
// save the current viewport
glPushAttrib( GL_VIEWPORT_BIT );

 // set up a viewport for drawing text (a "textport")
  glViewport(tpl, tpb, tpw, tph);

  // save the current projection
  glMatrixMode( GL_PROJECTION );
  glPushMatrix();

    // set up a 2D orthographic projection,
    // nchars wide by nline high
    glLoadIdentity();
    gluOrtho2D( 0, FAW*nchars, -FDH, (FDH_FAH)*nline );

    // save the current modelview matrix
    glMatrixMode( GL_MODELVIEW );
    glPushMatrix();
      glLoadIdentity();

      glPushMatrix();
        // translate the first character into position
        glTranslatef(tx, ty, 0);
        glColor3f(r, g, b);
        // draw all of the characters in the string
        char str[] = "Hello";
        for(int j=0; j<strlen(str); j++)
          glutStrokeCharacter(font, str[j]);
      glPopMatrix();

    // restore the previous projection matrix
    glMatrixMode( GL_PROJECTION );
    glPopMatrix();
    // restore the previous modelview matrix
    glMatrixMode( GL_MODELVIEW );
    glPopMatrix();

// restore the original viewport
glPopAttrib();
```