Calculating the number of lines of text displayed on a monitor

Call n the number of lines of code (*loc*) or of text displayed on a given monitor M. A monitor M is specified by a tuple $M = (d, r_x, r_y, k)$, where $d > 0$ is the diagonal size, $r_x \in \mathbb{N}$ is the horizontal resolution (or number of pixels), $r_y \in \mathbb{N}$ is the vertical resolution, and $k \geq 1$ is the zoom level. Note that zoom levels less than 1 are not allowed.

1 Density

In typography, the font size is measured in points (pt), where 1 pt = 1/72 inch. Computer displays work by lighting up pixels, so the font size needs to be expressed in pixels rather than in length. However, the physical size of a pixel varies among different monitors because of two factors:

- the resolution, that is, r_x and r_y,
- the screen physical size, expressed by its diagonal d.

To capture this idea, we define the *density* δ of a monitor M as

$$\delta = \frac{\sqrt{r_x^2 + r_y^2}}{d}. \tag{1}$$

The density is measured in pixels per inch (ppi).
In the Windows operating system, the font size expressed in pt and the number of vertical pixels y required to draw a character are related by the formula [2]:

$$y(\text{font size}) = \frac{\text{font size (pt)}}{\text{72 inch}} * 120 \text{ ppi}$$

Note that, in [2], it is written that Windows use 96 ppi, instead of 120 ppi, to perform the above conversion. However, experimentally it was noticed that 120 ppi is closer to the actual conversion that Windows 10 is performing.

2 Readability

To ensure good readability, it is important to carefully choose the right character height for displayed text. Let x be the height of a character. x is expressed in inches. Among other factors, x depends on the distance between the viewer and the display: the further the viewer is, the larger it should be. This relationship is expressed using the angle A in the following figure:

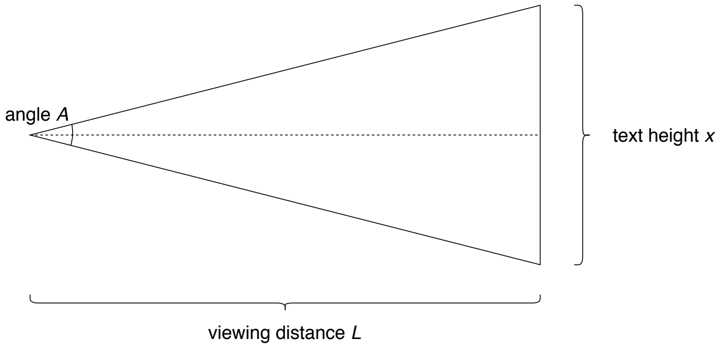

Figure 1: Relationship between the angle A, the viewing distance and the height of a character.

The angle A is usually expressed in arcminutes. 1 arcminute is 1/60 of 1 degree. A can be then computed by the formula

$$A \text{ (arcminute)} = 60 * arctan\left(\frac{x}{L}\right).$$

It is recommended to use at least 15 to 20 arcminutes [3]. For comparison, the angular acuity, which is the resolution of an eye, of a 10/10 eye is 1 arcminute [4].

3 Computing the number of lines of text

Turning to the number of lines of text of a monitor, each line uses a certain number y of vertical pixels. The number of lines of text shown in a monitor M is therefore:

$$n = \frac{r_y}{y}. \tag{2}$$

Given the height x of a line, the number of vertical pixels required to draw that line is:

$$y = x\delta. \tag{3}$$

Expanding Equation (2) using Equations (1) and (3):

$$n = \frac{r_y}{y} = \frac{r_y}{x\delta} = \frac{r_y d}{x\sqrt{r_x^2 + r_y^2}}. \tag{4}$$

The number of lines of text increases when the vertical resolution increases and it decreases when the line height increases. It also decreases when the density increases, having fixed the resolution and x. Indeed, increasing δ while keeping the resolution fixed means decreasing the diagonal of the monitor, which leads to a lower number of lines of text that can be displayed. In other words, n increases when increasing the diagonal of the monitor. The increase has clearly a maximum, which is explained in Subsection 4.1.

4 Zoom level

Windows 10 provides display scaling, which can increase the size of text and objects to make them easier to see on high PPI monitors. Windows uses a scale factor relative to a standard desktop of 96 PPI sitting at a 28 inch distance of the viewer [1]. Here, the scale factor, also called the zoom level, is computed relative to the number $\bar{y}$ of vertical pixels used by a font size of 11 pt according to Equation 1:

$$\bar{y} = y(11) = \frac{11}{72} * 120 = \frac{55}{3} = 18.\bar{3} \text{ px}$$

The above value will not be used directly in the following, so it is kept as a decimal value for convenience.

Given the number of pixels y needed to draw one character, the zoom level is then:

$$k = \frac{y}{\bar{y}}. \tag{5}$$

Note that y can be computed using Equation (3). If k is known, one can write Equation (2) as:

$$n = \frac{r_y}{k\bar{y}}.$$

4.1 Details

When writing software to compute n, some constraints must be considered. First of all, $k \geq 1$, so, from Equation (5):

$$y \geq \bar{y}.$$

Secondly, y cannot exceed r_y. To take into account these constraint, Equation (3) is rewritten as:

$$y = \begin{cases} x\delta, & \text{if } x\delta \geq \bar{y} \land x\delta \leq r_y \\ \bar{y}, & \text{if } x\delta < \bar{y} \\ r_y, & \text{if } x\delta > r_y. \end{cases}$$

Therefore, Equation (4) is rewritten as:

$$n = \begin{cases} \frac{r_y d}{x\sqrt{r_x^2 + r_y^2}}, & \text{if } x\delta \geq \bar{y} \land x\delta \leq r_y \\ \frac{r_y}{\bar{y}}, & \text{if } x\delta < \bar{y} \\ 1, & \text{if } x\delta > r_y. \end{cases}$$

In other words, n cannot be smaller than 1 and cannot be larger than the number of lines of text whose height is equal to $\bar{y}$.

References

[1] *Display Scaling: What It Is and What You Need to Know About it to Have Great Visuals.* URL: https://channel9.msdn.com/Events/Build/2015/2-63.

[2] *DPI and Device-Independent Pixels.* URL: https://docs.microsoft.com/en-us/windows/win32/learnwin32/dpi-and-device-independent-pixels.

[3] *Font Size and Legibility for Videowall Content.* URL: https://www.extron.com/article/videowallfontsize.

[4] *Visual acuity.* URL: https://en.wikipedia.org/wiki/Visual_acuity#Definition.