Criteria for the Appearance of Scrollbars in Windows

Robert C. Tausworthe

who?

Abstract

This study examines when scroll bars should be displayed in scrollable windows. In particular, the available area for application display is present when no scroll bar is present to limit it; however, the application display must be reduced when the scroll bar appears in its path This study provides the logic and rationale for showing and hiding scroll bars in a scrollable windowed application.

Preliminaries

This analysis is written as a *Mathematica* 8.0 notebook. The notations are therefore expressed in *Mathematica* terms, which may appear somewhat unfamiliar to the casual reader. However, it is hoped that the expressions resemble ordinary mathematics and English usage enough that the reader may follow the approach, reasoning, and method of solution, and thereby recognize the significance of the results.

In reading the analysis below, remember that *Mathematica* denotes functions using square brackets, rather than parentheses; thus, the function *f(x)* in *Mathematica* will be written as `f[x]`. Also, lists of terms appear in braces, as `{...}`. Vectors take the form of lists; the dot product of two vectors takes the form `vector1.vector2`; right matrix multiplication by a column vector takes the form `matrix.vector`, row-vector left multiplication of a matrix is written `vector.matrix`, and the vector-matrix-vector product normally written x^T`My` in mathematics takes the form `x.M.y`. The (undotted) product `(vector1 vector2)` evaluates to the vector of termwise element products. The i-th element of `vector1` is denoted `vector1[[i]]`. Lists may be displayed in columnar form using the `MatrixForm` function. `Simplify[]` and `FullSimplify[]` combine and rearrange terms usually to produce a more compact and readable expression. `Collect[e, λ]` combines and arranges the terms of the expression `e` into a linear sum of coefficients of the elements in the list λ. `N[e]` produces a numerical evaluation of the expression `e`. Writing `e/.a->b` causes `b` to be substituted for `a` in `e`. A % appearing in an expression represents the result of the previous computation. Finally, the form `e//Function` is the same as `Function[e]`.

But first, we apply some initialization. The first of these reads in a set of special functions and executes one of them that clears out the *Mathematica* namespace, restarting line numbering at 1; this is done so that whenever the *Mathematica* kernel is called upon to reevaluate the notebook, it starts with a clear slate. The second action reads in a set of basic constants (e.g., Earth's radius, Jupiter's mass), should these be needed, and applies a time stamp of the last execution of the notebook.

```
$$V = StringTake[$Version, 3];
If[ToExpression[$$V] > 5, $$V = "6.0"];
<< Tausworthe`TauReferences`;
Get[TauDirectory <> "~TauFunctions-" <> $$V <> ".mx"];
Get[TauDirectory <> "~TauConstants-" <> $$V <> ".mx"];
FreshStart[];
Today[]
```

```
Jan 22 2017, 10:32:54
```

```
Off[General::"spell1", General::"spell",
 Solve::"ifun" , ParametricPlot3D::"ppcom"]
```

Introduction

Consideration of when the horizontal and scroll bars should be visible or not involve the quantities

app_x, app_y	the application horizontal and vertical sizes
client_x, client_y	the client window horizontal and vertical sizes
view_x, view_y	the view horizontal and vertical size with scroll bars present

The criteria to be studied are

$$\begin{aligned} A &= \text{app\_x} > \text{view\_x} \\ B &= \text{app\_x} > \text{client\_x} \\ C &= \text{app\_y} > \text{view\_y} \\ D &= \text{app\_y} > \text{client\_y} \end{aligned} \tag{1}$$

A and C are true if the app exceeds the window with scroll bars present, and B and D are true when the app exceeds the window area without scroll bars. Clearly, if B is true, then A is true, and if D is true, then so is C.

$$\begin{aligned} &B \Longrightarrow A \\ &\overline{A} \Longrightarrow \overline{B} \\ &D \Longrightarrow C \\ &\overline{C} \Longrightarrow \overline{D} \end{aligned} \tag{2}$$

The Boolean value hb will be set true when the horizontal bar is to appear, and the Boolean value vb will be set true when the vertical bar is to appear. The criteria for setting the scroll bars are then

$$\begin{aligned} hb &== vb\,A \mid\mid \overline{vb}\,B \\ vb &== hb\,C \mid\mid \overline{hb}\;C \end{aligned} \tag{3}$$

In words, for example, the horizontal bar is present either when there is a vertical bar, but the app width exceeds the reduced client view width, or when there is not a vertical bar, but the app width exceeds the entire client width. A similar description holds for the vertical bar criterion.

These logical equations are not easily solved directly in *Mathematica*, so the following mapping from Boolean algebra to integer algebra makes the solution tractable: We take the integer 1 to represent true and 0 as false. Then the Boolean and, or, and not operations are, in numeric form,

$$\begin{array}{lcl} a\ b & \rightarrow & a * b \\ a \,||\, b & \rightarrow & a + b - a * b \\ \overline{a} & \rightarrow & (1 - a) \\ a \,||\, \overline{a}\ b & \rightarrow & a + b \end{array} \qquad (4)$$

Mathematica can solve simultaneous numeric equations, as shown below.

Method

Inasmuch as the symbols C and D are preempted by *Mathematica* to represent other usages, the Boolean A, B, C, and D values will, respectively, be represented by AA, BB, CC, and DD in the analysis to follow. The scroll bar presence criteria given in (3) above, translated into integer algebraic form, are

```
eqs = {hb == (vb * AA) + (1 - vb) BB, vb == (hb * CC) + (1 - hb) DD};
```

```
{hb == BB (1 - vb) + AA vb, vb == DD (1 - hb) + CC hb}
```

Simplification and applying the lemma that for any Boolean integer value x it is true that $x^2 == x$ yield

```
ceqs = Collect[eqs, {hb, vb}];

FullSimplify[Solve[ceqs, {hb, vb}]]
```

Further manual reduction produces

$$\left\{hb \rightarrow \frac{BB + (AA - BB)\ DD}{1 - (AA - BB)\ (CC - DD)},\ vb \rightarrow \frac{BB\ (CC - DD) + DD}{1 - (AA - BB)\ (CC - DD)}\right\};$$

Because of the relationships between A and B, and between C and D, one may verify using a truth table that

$$\begin{array}{l} A - B == A\ \overline{B} \\ C - D == C\ \overline{D} \end{array} \qquad (5)$$

The first of these equations in (5) is true only when the app and window sizes place the horizontal app extent within the scroll bar gap, and the second is only true when the situation applies to the vertical extent. The denominators in the numeric solutions above thus vanish when the both the horizontal and vertical app and window sizes are simultaneously within their scroll bar gaps. Translated back into logical terms, the criteria for showing the scroll bars are undetermined in this situation. However, when the size conditions are not in this condition, the numeric solution may be used to determine the logical rules for displaying the scroll bars. These are

$$\begin{array}{l} hb == B \,||\, \overline{B}\ A\ D == B \,||\, A\ D \\ vb == D \,||\, \overline{D}\ C\ B == D \,||\, C\ B \end{array} \qquad (6)$$

In the excluded situation, $(A\ \overline{B}\ C\ \overline{D})$, both bars should appear. The full criteria may thus be expressed as a combination of three conditions, which may then be reduced using Boolean algebra, as follows:

$$
\begin{aligned}
hb &== B \;||\; \overline{B}\, A\, D \;||\; \left(A\, \overline{B}\, C\, \overline{D}\right) \\
&== B \;||\; A \left(D \;||\; C\, \overline{D}\right) \\
&== B \;||\; A\, (D \;||\; C) \\
&== B \;||\; A\, C \\
vb &== D \;||\; \overline{D}\, C\, B \;||\; A\, \overline{B}\, C\, \overline{D} \\
&== D \;||\; C \left(B \;||\; A\, \overline{B}\right) \\
&== D \;||\; C\, (B \;||\; A) \\
&== D \;||\; C\, A
\end{aligned}
\tag{7}
$$

Conclusions

This study has derived the conditions under which scroll bars should be displayed in scrollable windows. It has presented the logic and rationale for showing and hiding scroll bars in a scrollable windowed application. The criteria are

$$
\begin{aligned}
hb &== B \;||\; A\, C \\
vb &== D \;||\; A\, C
\end{aligned}
\tag{8}
$$