AcroTEX.Net

Custom Designing Anime Control Button Layout

D. P. Story

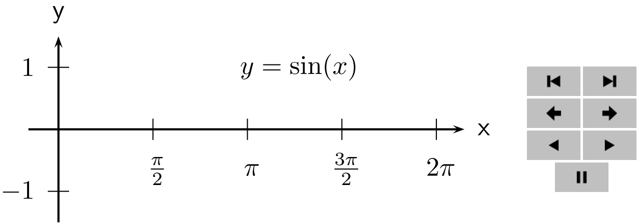
AeB Blog July 19, 2010

http://www.acrotex.net
Published: July 17, 2010

1. Creating a Custom Layout

We demonstrate how to move the control buttons for anime to a location other than the default one (which is beneath the anime). See the AeB Pro documentation for details of ocgAnime and \btnAnime.

To move the buttons, use the commands \animeSetup and \insertCtrlButtons. The command \animeSetup takes the key-value pairs of \btAnime and ocgAnime, it processes the key-value pairs for the animation. Following \animeSetup, use \insertCtrlButtons to insert the control buttons for the animation. If the control buttons are for OCG animation, do not use the iconName key; aeb_pro uses this key to differentiate between button and OCG anime. If button animation is being used, then specify a value for iconName.



The verbatim listing of this anime follows.

```
\begin{minipage}[c]{190pt}
\begin{ocgAnime}{ocgAnimeName=sineAnime1,nFrames=41,controls=none}
```

```
3       <sine animation code, see attached source>
4   \end{ocgAnime}
5   \end{minipage}\quad{\animeSetup{ocgAnimeName=sineAnime1,nFrames=41,
6       speed=60,type=palindrome}\parbox{\btnAnimeCtrlW*2+4bp}{%
7           \offinterlineskip\centering
8           \btnAnimeGoToFirst\kern3bp\btnAnimeGoToLast\\[3bp]
9           \btnAnimeStepBack\kern3bp\btnAnimeStepForward\\[3bp]
10          \btnAnimePlayBack\kern3bp\btnAnimePlayForward\\[3bp]
11          \btnAnimePause
12      }}%
```

Comments: Following the OCG animation, in line (5), we begin a group with a left brace ({) to make changes to the parameters of \animeSetup local, and we execute \animeSetup with our selected key-value pairs.

The layout of the buttons is in two columns of two buttons in each column, buttons in each row are separated by 1bp. The internal command for the width of the buttons is \btnAnimeCtrlW, and we calculate, in line (6) using the calc package, the width of the rows.

In line (7) we execute \offinterlineskip to suppress the natural glue between lines, and \centering so all widgets are centered within the \parbox.

Internally, the control buttons are named,

```
\btnAnimeGoToFirst  \btnAnimeStepBack     \btnAnimePlayBack
\btnAnimePause      \btnAnimePlayForward  \btnAnimeStepForward,
\btnAnimeGoToLast   \btnAnimePlus         \btnAnimeMinus
```

Lines (8)–(11): Within the \parbox, we place the buttons we want to be included in the control set.

Finally, we close up the \parbox and close up the group in line (12).

2. Designing a Skin

The controls takes one of seven values none, skin1–skin6. skin5 and skin6 are undefined and default to skin1, they are meant to be user-defined skins. In this section we demonstrate how to define your own custom skin.

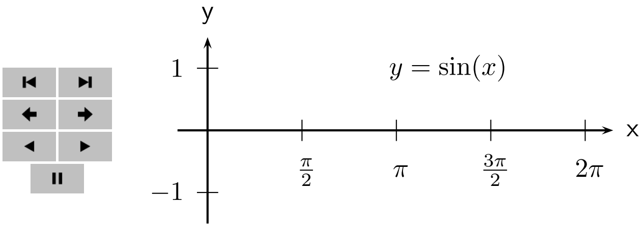
In the preamble, you will find the following definitions

```
1  \renewcommand{\btnAnimeSkinv}{%
2      \btnAnimeGoToFirst\kern3bp\btnAnimeGoToLast\\[3bp]
3      \btnAnimeStepBack\kern3bp\btnAnimeStepForward\\[3bp]
4      \btnAnimePlayBack\kern3bp\btnAnimePlayForward\\[3bp]
5      \btnAnimePause}
6  \renewcommand{\numWidgetsFirstRowV}{\aepnumWidgetsFirstRow{2}{2}}
```

In line (1) we \renewcommand the definition of \btnAnimeSkinv, which is the internal name of the fifth button set (\btnAnimeSkinvi is the name of the sixth one, in case you need to know). We define the same design seen earlier in lines (2)–(5).

Now line (6) requires explanation. The aeb_pro package needs to know the width of the first row (actually, it needs to know the width of the widest row, which is usually the first). For the custom skins, \numWidgetsFirstRowV and \numWidgetsFirstRowVI are defined to make it easy to pass this information to the package. You can see in line (6), \numWidgetsFirstRowV executes another internal command \aepnumWidgetsFirstRow; this command takes two arguments, the first is the number of buttons (on the first row, or the widest row) when the there is no speed control, and the second is the number of buttons when there is speed control (usually two more than the first). In our example, we have no speed control and so we put {2}{2}, as the number of buttons in the widest row does not change when using the nospeedcontrol option (or the usetworows option).(For another example, in skin1 these two parameters are {7}{9}.)

While we have the space on the page, let's do the animation!



The verbatim listing is. . .

```
{\animeSetup{ocgAnimeName=sineAnime2,nFrames=41,
    speed=60,type=palindrome,controls=skin5}\insertCtrlButtons
}\quad\begin{minipage}[c]{190pt}
```

```
4  \begin{ocgAnime}{ocgAnimeName=sineAnime2,nFrames=41,controls=none}
5      <sine animation code, see attached source>
6  \end{ocgAnime}
7  \end{minipage}
```

Comments: In this example, we place the controls to the left of the anime and use the techniques described in the AeB Blog article *Moving the Control Buttons for Button and OCG Animation*, now that our button design has been integrated into the anime system.

In line (2), note that we specify `controls=skin5`, and the use of `\insertCtrlButtons`. In the previous section we used `\offinterlineskip` and `\centering`, we do not do that here because `\insertCtrlButtons` does that for us.

Once the new skin is defined, it would behave like the other skins, its default location is beneath the anime.

The definition of the skin can be generalized a bit, right now, the definitions do not obey the two of the other documented parameters, `\btnanimebtnsep` (horizontal space between buttons) and `\btnanimerowsep` (vertical space between rows). For greater generality, we can revise our original definitions given earlier to now read,

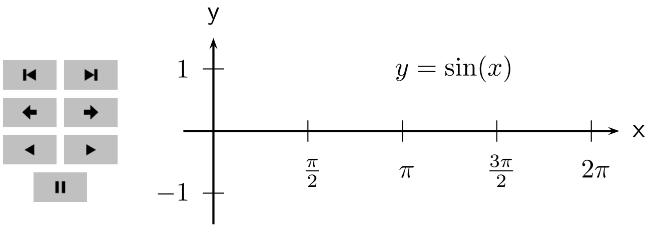
```
\renewcommand{\btnAnimeSkinv}{%
    \btnAnimeGoToFirst\@ctrlsbtnsep
    \btnAnimeGoToLast\\[\@btnanimerowsep]
    \btnAnimeStepBack\@ctrlsbtnsep
    \btnAnimeStepForward\\[\@btnanimerowsep]
```

```
        \btnAnimePlayBack\@ctrlsbtnsep
        \btnAnimePlayForward\\[\@btnanimerowsep]
        \btnAnimePause}
\renewcommand{\numWidgetsFirstRowV}{\aepnumWidgetsFirstRow{2}{2}}
```

Comments: The \@ctrlsbtnsep and \@btnanimerowsep are macros that expand to the correct spacing (horizontal/vertical respectively) based on the current parameters.

Let's clone the last example, and try our new definition. We also make the following definitions,

```
\renewcommand{\btnanimebtnsep}{3bp}  % default is 1bp
\renewcommand{\btnanimerowsep}{3bp}  % default is 1bp
```



. . . and there you go now, more space between the buttons and the rows.

Well, that's all for now, I simply must get back to my retirement!