Subject: Re: string animations etc.
Posted by MKatz843 on Sun, 19 Sep 2004 19:56:24 GMT
View Forum Message <> Reply to Message

I don't know if this will do what you want, but I thought I'd share.
I wrote this eons ago for displaying a progress-bar in the text
window. This is for Unix-type command-line implementations of IDL
(Mac, Linux, etc.) I have no idea if it would be work in Windows. I'm
guessing it won't.

```
;+
; ---------------------------
; 2/27/96
; IDL procedure: textmeter.pro
;
; This procedure draws a "meter" on the screen to display what
; fraction of a job has been completed.  Calling textmeter with
; x=0 initializes the meter.
; ---------------------------
;-

pro textmeter, x, s0, remain=remain
common block_textmeter, time0, base, n0, n1, backup, name0
  if not defined(s0) then s0=''
  if not defined(name) then name0=''
  if ((x LE 0.01) or (not defined(base))) then begin  ; set-up
    time0 = systime(1)
    name0 = s0
    n0 = strlen(s0+':')
    s1 = '----+'
    base='' & for i=1,10 do base=base+s1
    n1 = strlen(base)   ; should be 50
    nb = n0+n1+8
    backup='' & for i=0,nb do backup = backup + string(8b)
  endif
  done = (x GE 1.)
  x = 0. > x < 0.9999
  nx = floor(x*float(n1))
  st=':' & for i=1,nx do st=st+'*'
  dt = systime(1)-time0
  if keyword_set(remain) $
    then if (x EQ 0.) $
      then time="???????????" $
      else time = string(format="(f8.2,' s')", dt*(1.-x)/x) $
    else time = string(format="(f8.2,' s')",dt)
  out = s0 + st + strmid(base,nx+1,n1-nx-1) + time + backup
  print, format="(a,$)", out
  if done then begin
```

```
    print, ''
    print, ''
  endif
return
end
```

Here's a command-line program to see how it works

```
IDL> textmeter,0.,'hi' & for i=0,100 do begin & textmeter, i/100.,
'hi' & wait, 0.05 & end
```

Basically, you "initialize" it by sending a 0.0 argument.
The second argument is an optional string which you can use to tell the user what function is being processed, for example. When it's running, sent it values between 0.0 and 1.0 which represent the fraction of the job that has been completed. The text-meter will report the elapsed time. When it reaches 1.0 or greater, it stops.

If you want the meter to predict how much time is remaining, set the /remain keyword, and it will do its best to estimate, assuming the progress information you send it is linear in time.

Note that this function relies on string(8b) being able to back-up the cursor when printing. This works in xterm. Perhaps in some other implementations this might need to be tweaked to make it work. If you write any other text to the screen while this is running, it will disrupt the output, but in a non-fatal way. It just won't look nice anymore. The backing-up essentially erases the line it's just printed and allows the terminal to overwrite what it just wrote. You could get the hang of it and make any kind of animation you want.

Best,
M. Katz

ee7klt@sfsu.edu (KL) wrote in message
news:<20fda9c1.0409171417.1319289@posting.google.com>...
> Hi,
> I was trying to do a simple string animation whereby there is a
> running number printed i.e. a number eg. of the form xxxx.xx that gets
> updated continuously next to some plot in the display window (as
> opposed to jumping discretely) while the program is crunching away in
> a 'for loop'. Does any one have ideas on how I may do this?
>
> Also, this program calculates the path of some object thru' space. I'd
> like to be able to have this line drawn in real-time in the display
> window, corresponding to the running number.

>
> Thanks,
> KL

Subject: Re: string animations etc.
Posted by Richard French on Sun, 19 Sep 2004 23:52:43 GMT
View Forum Message <> Reply to Message

Neat! But 'defined()' does not seem to be a built-in IDL function (I cobbled one together to get this to run), and I think the line should read

If not defined(name0) then name0=''

Thanks for posting this!
Dick

```
>  ;+
>  ; ---------------------------
>  ; 2/27/96
>  ; IDL procedure: textmeter.pro
>  ;
>  ; This procedure draws a "meter" on the screen to display what
>  ; fraction of a job has been completed.  Calling textmeter with
>  ; x=0 initializes the meter.
>  ; ---------------------------
>  ;-
>
>  pro textmeter, x, s0, remain=remain
>  common block_textmeter, time0, base, n0, n1, backup, name0
>  if not defined(s0) then s0=''
>  if not defined(name) then name0=''
>  if ((x LE 0.01) or (not defined(base))) then begin  ; set-up
>    time0 = systime(1)
>    name0 = s0
>    n0 = strlen(s0+':')
>    s1 = '----+'
>    base='' & for i=1,10 do base=base+s1
>    n1 = strlen(base)   ; should be 50
>    nb = n0+n1+8
>    backup='' & for i=0,nb do backup = backup + string(8b)
>  endif
>  done = (x GE 1.)
>  x = 0. > x < 0.9999
>  nx = floor(x*float(n1))
>  st=':' & for i=1,nx do st=st+'*'
>  dt = systime(1)-time0
>  if keyword_set(remain) $
```

> then if (x EQ 0.) $
> then time="??????????" $
> else time = string(format="(f8.2,' s')", dt*(1.-x)/x) $
> else time = string(format="(f8.2,' s')",dt)
> out = s0 + st + strmid(base,nx+1,n1-nx-1) + time + backup
> print, format="(a,$)", out
> if done then begin
> print, ''
> print, ''
> endif
> return
> end
>
>
> Here's a command-line program to see how it works
>
> IDL> textmeter,0.,'hi' & for i=0,100 do begin & textmeter, i/100.,
> 'hi' & wait, 0.05 & end
>
> Basically, you "initialize" it by sending a 0.0 argument.
> The second argument is an optional string which you can use to tell
> the user what function is being processed, for example. When it's
> running, sent it values between 0.0 and 1.0 which represent the
> fraction of the job that has been completed. The text-meter will
> report the elapsed time. When it reaches 1.0 or greater, it stops.
>
> If you want the meter to predict how much time is remaining, set the
> /remain keyword, and it will do its best to estimate, assuming the
> progress information you send it is linear in time.
>
> Note that this function relies on string(8b) being able to back-up the
> cursor when printing. This works in xterm. Perhaps in some other
> implementations this might need to be tweaked to make it work. If you
> write any other text to the screen while this is running, it will
> disrupt the output, but in a non-fatal way. It just won't look nice
> anymore. The backing-up essentially erases the line it's just printed
> and allows the terminal to overwrite what it just wrote. You could get
> the hang of it and make any kind of animation you want.
>
> Best,
> M. Katz
>
>
> ee7klt@sfsu.edu (KL) wrote in message
> news:<20fda9c1.0409171417.1319289@posting.google.com>...
>> Hi,
>> I was trying to do a simple string animation whereby there is a
>> running number printed i.e. a number eg. of the form xxxx.xx that gets

>> updated continuously next to some plot in the display window (as
>> opposed to jumping discretely) while the program is crunching away in
>> a 'for loop'. Does any one have ideas on how I may do this?
>>
>> Also, this program calculates the path of some object thru' space. I'd
>> like to be able to have this line drawn in real-time in the display
>> window, corresponding to the running number.
>>
>> Thanks,
>> KL

Subject: Re: string animations etc.
Posted by David Fanning on Mon, 20 Sep 2004 06:51:54 GMT
View Forum Message <> Reply to Message

KL writes:

> I was trying to do a simple string animation whereby there is a
> running number printed i.e. a number eg. of the form xxxx.xx that gets
> updated continuously next to some plot in the display window (as
> opposed to jumping discretely) while the program is crunching away in
> a 'for loop'. Does any one have ideas on how I may do this?
>
> Also, this program calculates the path of some object thru' space. I'd
> like to be able to have this line drawn in real-time in the display
> window, corresponding to the running number.

A technique I have used in the past is to create a very small
pixmap window that I can erase and write a number in. Then I
just copy the pixmap window to the display window with a
Device, Copy command. In the FOR loop, the code looks like
this:

```
FOR j=0,n DO BEGIN
   ... ; Whatever you are doing.
   WSet, pixmapWindow
   Erase
   XYOUTS, StrTrim(j,2), 0.5, 0.5, Alignment=0.5, /Normal
   WSet, displayWindow
   DEVICE, COPY=[0, 0, 25, 25, 0, 0, pixmapWindow]
ENDFOR
```

Cheers,

David
--
David W. Fanning, Ph.D.

Subject: Re: string animations etc.
Posted by MKatz843 on Mon, 20 Sep 2004 22:03:23 GMT
View Forum Message <> Reply to Message

"Richard G. French" <rfrench@wellesley.edu> wrote in message
news:<BD73940A.3502%rfrench@wellesley.edu>...
> Neat! But 'defined()' does not seem to be a built-in IDL function (I cobbled
> one together to get this to run), and I think the line should read
>
> If not defined(name0) then name0=''
>
Sorry about that! Here are two handy functions I use. The first is
defined() and the second is defined_and_equals(). The second one uses
the first one, and it lets you catch two birds with one net. Trust me,
it comes in handy sometimes.

```
;+
; ------------------------
; 12/04/94
; IDL function: defined.pro
;   For a given input variable, this function returns
;     1b if the variable is defined
;     0b if the variable is undefined
; ------------------------
;-

function defined, a
return, (size(a, /type) NE 0)
end


;+
;----------------------------
; 12/11/03
; IDL function: defined_and_equals.pro
;
; This boolean function does two things.
; The doesn't know if the argument is defined, but want to know:
;   if it IS defined, then does it equal arg2?
;----------------------------
;-

function defined_and_equals, a, b
```

```
return, defined(a) ? (a EQ b) : 0b
end
```

M. Katz