Subject: Re: grROI Style and anROI Type
Posted by Dan Carreira on Mon, 28 Apr 2003 15:50:09 GMT
View Forum Message <> Reply to Message

"Ben Tupper" <btupper@bigelow.org> wrote in message
news:3EAD3488.7060702@bigelow.org...
> Dan Carreira wrote:
>> Hi all,
>>
>> I'm new (2 months) to IDL and have been reading through a lot of the
older
>> posts in an attempt to absorb every little tidbit of information that I
can
>> and I just wanted to say thanks before I jump into my question.
>>
>> So Thanks.
>>
>> Now my question.
>>
>> I'm writing an application that creates and modifies ROI's through mouse
>> interaction.
>>
>> My application currently allows the user to define 2 ROI's (ROI &
CutAddROI)
>> and a seed point. The user defines the ROI and can then select an "add
tool"
>> which allows them to create another ROI which I call CutAddROI. The seed
>> point will later be used to identify which part of ROI I want to keep
when I
>> cut a section off of ROI.
>>
>> When generating masks of the ROI's I'm running into a problem.
>>
>> ROI is and always will be a closed polyline in my application. So the
mask
>> is generated exactly the way I need it. If I draw it in the shape of a U
it
>> closes the top when it generates the boundary of the ROI.
>>
>> CutAddROI can be an open polyline or if the user wants can close it by
>> drawing it that way using the mouse. If I draw a U then I want the top
of
>> the U open when I generate the boundary of the region. My problem is
that
>> when I generate the mask for CutAddROI it always generates it as if it
was a
>> closed polyline, essentially closing the top of the U in the boundary
mask.

>>
>> I've tried changing the Style of the CutAddROI but nothing changes as
far as
>> the Mask is concerned.
>>
>> I also noticed that the grROI and anROI Init methods have different
keywords
>> for almost identical variables grROI has STYLE and anROI has TYPE. Why
are
>> they different and will changing my IDLgrROI CutAddROI to an IDLanROI
fix my
>> problem?
>>
>> Here's a code snippet in case anyone's interested. This code gets called
>> when the user releases the left mouse button when they're done drawing
the
>> region to add. I'm using David Fannings Find_Boundary (thank you David)
>> function to determine the boundary of the mask which I then use to set
the
>> ROI data to. I made a small modification to Find_Boundary so that it can
>> accept a starting point for the boundary calculations. In a mask with
>> multiple "islands" I can choose the island that I want the boundary for
by
>> setting the FirstPt to a location on that island's boundary.
>>
>>   oCutAddROI->AppendData, ImageX, ImageY, 0.01
>>   oCutAddROI->SetProperty, Style = 2; I've changed the style to 1 and
there
>> was no difference in generated mask
>>
>>   ;ADD the region
>>   Mask1 = oCutAddROI->ComputeMask(Location =[-128,-128,0.01],
Dimensions =
>> [256,256], Initialize = 0, mask_rule = 0)
>>   Mask = oROI->ComputeMask(Location =[-128,-128,0.01], Dimensions =
>> [256,256], Initialize = 0, mask_rule = 0)
>>   Mask = Mask + Mask1
>>
>>   indices = Where(Mask GT 0, num_indices)
>>   FirstPt = [Seed.X + 128, Seed.Y + 128]
>>
>>   if num_indices GT 0 then begin
>>   Boundary = Find_Boundary(indices, XSize=256, YSize=256, FirstPt =
>> FirstPt)
>>   if N_ELEMENTS(Boundary) GT 1 then begin
>>   SizeBoundary = SIZE(Boundary)
>>   Data = MAKE_ARRAY(3,SizeBoundary[2],VALUE = 0.01); 0.01 is the
Z

>> position for ROI's
>> Data[0,*] = Boundary[0,*] - 128
>> Data[1,*] = Boundary[1,*] - 128
>> oROI->SetProperty, Data = Data
>> endif
>> endif
>>
>> oImage->SetProperty, Data = Mask
>>
>> ;Delete the CutAddROI Data
>> oCutAddROI->SetProperty, Data = [0,0], /Hide
>> oCutAddROI->RemoveData
>>
>> oWindow->Draw, oView
>>
>> Thanks in advance
>>
>> Dan
>>
>>
>
> Hello,
>
> You have a number of interesting things going here. I have only dabbled
> with the ROIs before so I have limited help to offer and others will
> hopefully have more insight.
>
> The STYLE vs TYPE keyword issue is a bit surprising to me as it clearly
> defines the same attribute in the docs. Since IDLgrROI inherits from
> IDLanROI you would think that IDLgrROI would not need to define this
> attribute at all - but if it does, it should at least have the same
> keyword name (in this case it seems like it should be TYPE). I think
> there is something a bit lacking in the documentation for IDLgrROI in
> this case.
>
> I recall that the TYPE definition for IDLanROI (which will be true for
> IDLgrROI, too) is static for the throughout the lifecycle of the
> particular object's instance. You can get it but not set it. So , how
> you define the object upon initialization - is its definition until it
> is destroyed. To make a particular ROI change from closed-to-open you
> will have to create a new open polygon object and transfer all the
> attributes of the closed polygon ROI to the new open polygon ROI. The
> new ROI then replaces the old ROI.
>
> Cheers,
> Ben
>

I had tried changing the Type with the SetProperty method but it gave me an error. I thought that the error was a result of me trying to specify a variable from the anROI class. I thought that maybe because the grROI class had a Init method that I couldn't access the anROI Init method.

I overlooked the static variable part when looking at the documentation I hadn't run across a variable that was static yet so I didn't pay the {Get} part much attention, I'll have to pay more attention in the future.

Anyway after reading your e-mail I went bakc and set my CutAddROI type to an open polyline when I created it and now now it's generating the mask properly.

Thanks for your help Ben.

Dan
