Subject: Keeping objects fixed in function graphics
Posted by Helder Marchetto on Thu, 19 Dec 2013 12:40:06 GMT
View Forum Message <> Reply to Message

Hi,
since I spent the last half an hour trying to figure this out, I thought I might as well share this.
The reason and idea behind this, was to draw in a window where I have an image some sort of markers that stay where they are. For example a grid or an aiming target or crosshair.
One should be able to pan and zoom the image below it, but not these objects on top.
Well, this is how I did it. Let me know if you know of a better/cleaner way, otherwise I'll stick to this.
What I did was basically turn off the event handlers for mouse movements and any other sort.
Here is the code:

```
;#####################################
FUNCTION AvoidMovingObj::MouseDown, oWin, x, y, iButton, KeyMods, nClicks
RETURN, 1
END

FUNCTION AvoidMovingObj::MouseMotion, oWin, x, y, KeyMods
RETURN, ~ISA(oWin.GetSelect(), 'ELLIPSE')
END

FUNCTION AvoidMovingObj::MouseUp, oWin, x, y, iButton
RETURN, ~ISA(oWin.GetSelect(), 'ELLIPSE')
END

FUNCTION AvoidMovingObj::MouseWheel, oWin, x, y, Delta, KeyMods
RETURN, ~ISA(oWin.GetSelect(), 'ELLIPSE')
END

PRO AvoidMovingObj__define
void = {AvoidMovingObj, inherits GraphicsEventAdapter}
END

PRO AvoidMovingObjTest
p = PLOT(/test)
e = ellipse(0.5,0.5, '-r2', FILL_BACKGROUND=0, /norm)
e.window.EVENT_HANDLER=Obj_New('AvoidMovingObj')
END
;#####################################
```

There are two clear drawbacks in this way of working:
1) if there are ellipses that one would like to move, than I should make sure that the correct ellipse (or object) is not moved and the rest is moved. I think this is solvable, but I didn't spend time on it yet
2) this seems to be an intrinsic drawback of this method: when clicking on the "unmovable" object, the mouse cursor will stay as it is until another object has been clicked. Not terrible, but not

elegant.

I hope I'm not the only one in need for this and if you have suggestion on how to improve this... very welcome!
Cheers,
Helder

Subject: Re: Keeping objects fixed in function graphics
Posted by Helder Marchetto on Thu, 19 Dec 2013 13:10:03 GMT
View Forum Message <> Reply to Message

On Thursday, December 19, 2013 1:40:06 PM UTC+1, Helder wrote:
> Hi,
>
> since I spent the last half an hour trying to figure this out, I thought I might as well share this.
>
> The reason and idea behind this, was to draw in a window where I have an image some sort of markers that stay where they are. For example a grid or an aiming target or crosshair.
>
> One should be able to pan and zoom the image below it, but not these objects on top.
>
> Well, this is how I did it. Let me know if you know of a better/cleaner way, otherwise I'll stick to this.
>
> What I did was basically turn off the event handlers for mouse movements and any other sort. Here is the code:
>
>
>
> ;#####################################
>
> FUNCTION AvoidMovingObj::MouseDown, oWin, x, y, iButton, KeyMods, nClicks
>
> RETURN, 1
>
> END
>
>
>
> FUNCTION AvoidMovingObj::MouseMotion, oWin, x, y, KeyMods
>
> RETURN, ~ISA(oWin.GetSelect(), 'ELLIPSE')
>
> END
>
>
>

```
>  FUNCTION AvoidMovingObj::MouseUp, oWin, x, y, iButton
>
>  RETURN, ~ISA(oWin.GetSelect(), 'ELLIPSE')
>
>  END
>
>
>
>  FUNCTION AvoidMovingObj::MouseWheel, oWin, x, y, Delta, KeyMods
>
>  RETURN, ~ISA(oWin.GetSelect(), 'ELLIPSE')
>
>  END
>
>
>
>  PRO AvoidMovingObj__define
>
>  void = {AvoidMovingObj, inherits GraphicsEventAdapter}
>
>  END
>
>
>
>  PRO AvoidMovingObjTest
>
>  p = PLOT(/test)
>
>  e = ellipse(0.5,0.5, '-r2', FILL_BACKGROUND=0, /norm)
>
>  e.window.EVENT_HANDLER=Obj_New('AvoidMovingObj')
>
>  END
>
>  ;#####################################
>
>
>
>  There are two clear drawbacks in this way of working:
>
>  1) if there are ellipses that one would like to move, than I should make sure that the correct
ellipse (or object) is not moved and the rest is moved. I think this is solvable, but I didn't spend
time on it yet
>
>  2) this seems to be an intrinsic drawback of this method: when clicking on the "unmovable"
object, the mouse cursor will stay as it is until another object has been clicked. Not terrible, but not
elegant.
>
```

>
>
> I hope I'm not the only one in need for this and if you have suggestion on how to improve this... very welcome!
>
> Cheers,
>
> Helder

Ok,
So the solution for problem 1) (see above) is to substitute the lines with:
RETURN, ~ISA(oWin.GetSelect(), 'ELLIPSE')
with this line:
o = oWin.GetSelect()
IF ISA(oWin.GetSelect(), 'ELLIPSE') && (o.NAME EQ self.Name) THEN RETURN, 0 $
ELSE RETURN, 1

and to add an Init method:

```
FUNCTION AvoidMovingObj::Init, Name
self.Name = Name
RETURN, 1
END

PRO AvoidMovingObj__define
void = {AvoidMovingObj, inherits GraphicsEventAdapter, Name:''}
END
```

and then to set the event_handler property like this:
e.window.EVENT_HANDLER=Obj_New('AvoidMovingObj', 'Obj1Name')

That solves that...

Cheers,
h

Subject: Re: Keeping objects fixed in function graphics
Posted by lecacheux.alain on Thu, 19 Dec 2013 13:56:51 GMT
View Forum Message <> Reply to Message

Le jeudi 19 décembre 2013 14:10:03 UTC+1, Helder a écrit :
> On Thursday, December 19, 2013 1:40:06 PM UTC+1, Helder wrote:
>
>> Hi,
>
>>
>

>> since I spent the last half an hour trying to figure this out, I thought I might as well share this.
>
>>
>
>> The reason and idea behind this, was to draw in a window where I have an image some sort of markers that stay where they are. For example a grid or an aiming target or crosshair.
>
>>
>
>> One should be able to pan and zoom the image below it, but not these objects on top.
>
>>
>
>> Well, this is how I did it. Let me know if you know of a better/cleaner way, otherwise I'll stick to this.
>
>>
>
>> What I did was basically turn off the event handlers for mouse movements and any other sort. Here is the code:
>
>>
>
>>
>
>>
>
>> ;#####################################
>
>>
>
>> FUNCTION AvoidMovingObj::MouseDown, oWin, x, y, iButton, KeyMods, nClicks
>
>>
>
>> RETURN, 1
>
>>
>
>> END
>
>>
>
>>
>
>>
>
>> FUNCTION AvoidMovingObj::MouseMotion, oWin, x, y, KeyMods

```
>
>>
>
>>  RETURN, ~ISA(oWin.GetSelect(), 'ELLIPSE')
>
>>
>
>>  END
>
>>
>
>>
>
>>
>
>>  FUNCTION AvoidMovingObj::MouseUp, oWin, x, y, iButton
>
>>
>
>>  RETURN, ~ISA(oWin.GetSelect(), 'ELLIPSE')
>
>>
>
>>  END
>
>>
>
>>
>
>>
>
>>  FUNCTION AvoidMovingObj::MouseWheel, oWin, x, y, Delta, KeyMods
>
>>
>
>>  RETURN, ~ISA(oWin.GetSelect(), 'ELLIPSE')
>
>>
>
>>  END
>
>>
>
>>
>
>>
>
>>  PRO AvoidMovingObj__define
```

>
>>
>
>> void = {AvoidMovingObj, inherits GraphicsEventAdapter}
>
>>
>
>> END
>
>>
>
>>
>
>>
>
>> PRO AvoidMovingObjTest
>
>>
>
>> p = PLOT(/test)
>
>>
>
>> e = ellipse(0.5,0.5, '-r2', FILL_BACKGROUND=0, /norm)
>
>>
>
>> e.window.EVENT_HANDLER=Obj_New('AvoidMovingObj')
>
>>
>
>> END
>
>>
>
>> ;#####################################
>
>>
>
>>
>
>>
>
>> There are two clear drawbacks in this way of working:
>
>>
>
>> 1) if there are ellipses that one would like to move, than I should make sure that the correct

ellipse (or object) is not moved and the rest is moved. I think this is solvable, but I didn't spend time on it yet
>
>>
>
>> 2) this seems to be an intrinsic drawback of this method: when clicking on the "unmovable" object, the mouse cursor will stay as it is until another object has been clicked. Not terrible, but not elegant.
>
>>
>
>>
>
>>
>
>> I hope I'm not the only one in need for this and if you have suggestion on how to improve this... very welcome!
>
>>
>
>> Cheers,
>
>>
>
>> Helder
>
>
>
> Ok,
>
> So the solution for problem 1) (see above) is to substitute the lines with:
>
> RETURN, ~ISA(oWin.GetSelect(), 'ELLIPSE')
>
> with this line:
>
> o = oWin.GetSelect()
>
> IF ISA(oWin.GetSelect(), 'ELLIPSE') && (o.NAME EQ self.Name) THEN RETURN, 0 $
>
>   ELSE RETURN, 1
>
>
>
> and to add an Init method:
>
>
>

> FUNCTION AvoidMovingObj::Init, Name
>
> self.Name = Name
>
> RETURN, 1
>
> END
>
>
>
> PRO AvoidMovingObj__define
>
> void = {AvoidMovingObj, inherits GraphicsEventAdapter, Name:''}
>
> END
>
>
>
> and then to set the event_handler property like this:
>
> e.window.EVENT_HANDLER=Obj_New('AvoidMovingObj', 'Obj1Name')
>
>
>
> That solves that...
>
>
>
> Cheers,
>
> h

If you put your "steady" objects as "annotation" objects (TEXT, ELLIPSE, POLYLINE, etc..) bu using /RELATIVE keyword, I guess that you will get what you want.
alx.

Subject: Re: Keeping objects fixed in function graphics
Posted by Helder Marchetto on Thu, 19 Dec 2013 14:14:53 GMT
View Forum Message <> Reply to Message

On Thursday, December 19, 2013 2:56:51 PM UTC+1, alx wrote:
> Le jeudi 19 décembre 2013 14:10:03 UTC+1, Helder a écrit :
>
>> On Thursday, December 19, 2013 1:40:06 PM UTC+1, Helder wrote:
>
>>

>
>>> Hi,
>
>>
>
>>>
>
>>
>
>>> since I spent the last half an hour trying to figure this out, I thought I might as well share this.
>
>>
>
>>>
>
>>
>
>>> The reason and idea behind this, was to draw in a window where I have an image some sort of markers that stay where they are. For example a grid or an aiming target or crosshair.
>
>>
>
>>>
>
>>
>
>>> One should be able to pan and zoom the image below it, but not these objects on top.
>
>>
>
>>>
>
>>
>
>>> Well, this is how I did it. Let me know if you know of a better/cleaner way, otherwise I'll stick to this.
>
>>
>
>>>
>
>>
>
>>> What I did was basically turn off the event handlers for mouse movements and any other sort. Here is the code:
>
>>
>

```
>>>
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>> ;####################################
>
>>
>
>>>
>
>>
>
>>> FUNCTION AvoidMovingObj::MouseDown, oWin, x, y, iButton, KeyMods, nClicks
>
>>
>
>>>
>
>>
>
>>> RETURN, 1
>
>>
>
>>>
>
>>
>
>>> END
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
```

```
>>>
>
>>
>
>>> FUNCTION AvoidMovingObj::MouseMotion, oWin, x, y, KeyMods
>
>>
>
>>>
>
>>
>
>>> RETURN, ~ISA(oWin.GetSelect(), 'ELLIPSE')
>
>>
>
>>>
>
>>
>
>>> END
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>> FUNCTION AvoidMovingObj::MouseUp, oWin, x, y, iButton
>
>>
>
>>>
>
>>
>
>>> RETURN, ~ISA(oWin.GetSelect(), 'ELLIPSE')
>
>>
>
```

```
>>>
>
>>
>
>>> END
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>> FUNCTION AvoidMovingObj::MouseWheel, oWin, x, y, Delta, KeyMods
>
>>
>
>>>
>
>>
>
>>> RETURN, ~ISA(oWin.GetSelect(), 'ELLIPSE')
>
>>
>
>>>
>
>>
>
>>> END
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
```

```
>>>
>
>>
>
>>> PRO AvoidMovingObj__define
>
>>
>
>>>
>
>>
>
>>> void = {AvoidMovingObj, inherits GraphicsEventAdapter}
>
>>
>
>>>
>
>>
>
>>> END
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>> PRO AvoidMovingObjTest
>
>>
>
>>>
>
>>
>
>>> p = PLOT(/test)
>
>>
>
```

```
>>>
>
>>
>
>>>  e = ellipse(0.5,0.5, '-r2', FILL_BACKGROUND=0, /norm)
>
>>
>
>>>
>
>>
>
>>>  e.window.EVENT_HANDLER=Obj_New('AvoidMovingObj')
>
>>
>
>>>
>
>>
>
>>>  END
>
>>
>
>>>
>
>>
>
>>>  ;####################################
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>>  There are two clear drawbacks in this way of working:
>
>>
>
```

>>>
>
>>
>
>>> 1) if there are ellipses that one would like to move, than I should make sure that the correct ellipse (or object) is not moved and the rest is moved. I think this is solvable, but I didn't spend time on it yet
>
>>
>
>>>
>
>>
>
>>> 2) this seems to be an intrinsic drawback of this method: when clicking on the "unmovable" object, the mouse cursor will stay as it is until another object has been clicked. Not terrible, but not elegant.
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>> I hope I'm not the only one in need for this and if you have suggestion on how to improve this... very welcome!
>
>>
>
>>>
>
>>
>
>>> Cheers,
>
>>
>
>>>
>
>>

>
>>> Helder
>
>>
>
>>
>
>>
>
>> Ok,
>
>>
>
>> So the solution for problem 1) (see above) is to substitute the lines with:
>
>>
>
>> RETURN, ~ISA(oWin.GetSelect(), 'ELLIPSE')
>
>>
>
>> with this line:
>
>>
>
>> o = oWin.GetSelect()
>
>>
>
>> IF ISA(oWin.GetSelect(), 'ELLIPSE') && (o.NAME EQ self.Name) THEN RETURN, 0 $
>
>>
>
>>   ELSE RETURN, 1
>
>>
>
>>
>
>>
>
>> and to add an Init method:
>
>>
>
>>
>
>>
>
>>

```
>
>>  FUNCTION AvoidMovingObj::Init, Name
>
>>
>
>>  self.Name = Name
>
>>
>
>>  RETURN, 1
>
>>
>
>>  END
>
>>
>
>>
>
>>
>
>>  PRO AvoidMovingObj__define
>
>>
>
>>  void = {AvoidMovingObj, inherits GraphicsEventAdapter, Name:''}
>
>>
>
>>  END
>
>>
>
>>
>
>>
>
>>  and then to set the event_handler property like this:
>
>>
>
>>  e.window.EVENT_HANDLER=Obj_New('AvoidMovingObj', 'Obj1Name')
>
>>
>
>>
>
>>
>
>>
```

>
>> That solves that...
>
>>
>
>>
>
>>
>
>> Cheers,
>
>>
>
>> h
>
>
>
>
>
> If you put your "steady" objects as "annotation" objects (TEXT, ELLIPSE, POLYLINE, etc..) bu using /RELATIVE keyword, I guess that you will get what you want.
>
> alx.

Nice, thanks.
However, you can still select, pan, move and rotate the object by clicking on it. This is not very useful when overlaying a grid and the mouse is constantly going over the grid and if you click on it you might move/change it.
But yes, coordinates are now normalize for this object and don't change when the underlying object is changing in size or position (pan).

Cheers,
Helder

Subject: Re: Keeping objects fixed in function graphics
Posted by lecacheux.alain on Thu, 19 Dec 2013 14:25:44 GMT
View Forum Message <> Reply to Message

Le jeudi 19 décembre 2013 15:14:53 UTC+1, Helder a écrit :
> On Thursday, December 19, 2013 2:56:51 PM UTC+1, alx wrote:
>
>> Le jeudi 19 décembre 2013 14:10:03 UTC+1, Helder a écrit :
>
>>
>
>>> On Thursday, December 19, 2013 1:40:06 PM UTC+1, Helder wrote:
>

>>
>
>>>
>
>>
>
>>>> Hi,
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>> since I spent the last half an hour trying to figure this out, I thought I might as well share this.
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>> The reason and idea behind this, was to draw in a window where I have an image some sort of markers that stay where they are. For example a grid or an aiming target or crosshair.
>
>>
>
>>>
>
>>
>

>>>>
>
>>
>
>>>
>
>>
>
>>>> One should be able to pan and zoom the image below it, but not these objects on top.
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>> Well, this is how I did it. Let me know if you know of a better/cleaner way, otherwise I'll stick to this.
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>> What I did was basically turn off the event handlers for mouse movements and any other sort. Here is the code:
>
>>
>
>>>
>

>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>> ;#####################################
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>> FUNCTION AvoidMovingObj::MouseDown, oWin, x, y, iButton, KeyMods, nClicks
>
>>
>
>>>
>

```
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>> RETURN, 1
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>> END
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
```

```
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>>  FUNCTION AvoidMovingObj::MouseMotion, oWin, x, y, KeyMods
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>>  RETURN, ~ISA(oWin.GetSelect(), 'ELLIPSE')
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>>  END
>
>>
>
>>>
>
```

```
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>>  FUNCTION AvoidMovingObj::MouseUp, oWin, x, y, iButton
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>>  RETURN, ~ISA(oWin.GetSelect(), 'ELLIPSE')
>
>>
>
>>>
>
```

```
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>> END
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>> FUNCTION AvoidMovingObj::MouseWheel, oWin, x, y, Delta, KeyMods
>
>>
>
>>>
>
```

```
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>>  RETURN, ~ISA(oWin.GetSelect(), 'ELLIPSE')
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>>  END
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
```

```
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>>  PRO AvoidMovingObj__define
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>>  void = {AvoidMovingObj, inherits GraphicsEventAdapter}
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>>  END
>
>>
>
>>>
>
```

```
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>>  PRO AvoidMovingObjTest
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>>  p = PLOT(/test)
>
>>
>
>>>
>
```

```
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>>  e = ellipse(0.5,0.5, '-r2', FILL_BACKGROUND=0, /norm)
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>>  e.window.EVENT_HANDLER=Obj_New('AvoidMovingObj')
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>>  END
>
>>
>
>>>
>
```

>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>> ;####################################
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>> There are two clear drawbacks in this way of working:
>
>>
>
>>>
>

>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>> 1) if there are ellipses that one would like to move, than I should make sure that the correct ellipse (or object) is not moved and the rest is moved. I think this is solvable, but I didn't spend time on it yet
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>> 2) this seems to be an intrinsic drawback of this method: when clicking on the "unmovable" object, the mouse cursor will stay as it is until another object has been clicked. Not terrible, but not elegant.
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>>
>

>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>> I hope I'm not the only one in need for this and if you have suggestion on how to improve this... very welcome!
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>> Cheers,
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>> Helder

>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>> Ok,
>
>>
>
>>>
>
>>
>
>>> So the solution for problem 1) (see above) is to substitute the lines with:
>
>>
>
>>>
>
>>
>
>>> RETURN, ~ISA(oWin.GetSelect(), 'ELLIPSE')
>
>>
>
>>>
>
>>
>
>>> with this line:
>
>>
>
>>>
>
>>
>
>>> o = oWin.GetSelect()

>
>>
>
>>>
>
>>
>
>>> IF ISA(oWin.GetSelect(), 'ELLIPSE') && (o.NAME EQ self.Name) THEN RETURN, 0 $
>
>>
>
>>>
>
>>
>
>>>   ELSE RETURN, 1
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>> and to add an Init method:
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>> FUNCTION AvoidMovingObj::Init, Name

```
>
>>
>
>>>
>
>>
>
>>>  self.Name = Name
>
>>
>
>>>
>
>>
>
>>>  RETURN, 1
>
>>
>
>>>
>
>>
>
>>>  END
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>>  PRO AvoidMovingObj__define
>
>>
>
>>>
>
>>
>
>>>  void = {AvoidMovingObj, inherits GraphicsEventAdapter, Name:''}
```

>
>>
>
>>>
>
>>
>
>>> END
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>> and then to set the event_handler property like this:
>
>>
>
>>>
>
>>
>
>>> e.window.EVENT_HANDLER=Obj_New('AvoidMovingObj', 'Obj1Name')
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>> That solves that...

>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>> Cheers,
>
>>
>
>>>
>
>>
>
>>> h
>
>>
>
>>
>
>>
>
>>
>
>>
>
>> If you put your "steady" objects as "annotation" objects (TEXT, ELLIPSE, POLYLINE, etc..) bu using /RELATIVE keyword, I guess that you will get what you want.
>
>>
>
>> alx.
>
>
>
> Nice, thanks.
>
> However, you can still select, pan, move and rotate the object by clicking on it. This is not very useful when overlaying a grid and the mouse is constantly going over the grid and if you click on it

you might move/change it.
>
> But yes, coordinates are now normalize for this object and don't change when the underlying object is changing in size or position (pan).
>
>
>
> Cheers,
>
> Helder

Please note that by using WINDOW handler functions combined with HitTest function, you might get a finer control over what should be moved and what should be not.
alx.
