Hi, welcome back. How are you? Are you able to understand everything well? Don’t forget we have a manual, everything learned in this tutorial can be found as a reference in there.

So we know that we can set posinframex/y to different values to influence the way posvalx/y work. But what about posvalwtype and posvalhtype. This has about the same basic options as posinframex/y.

Here is a list:

-px: gives a width or a height in pixels.

-dp: gives a width or a height in inches.

-fc: gives a width or height relative to the parent’s size.

There are a lot more options, but these are the one’s you’ll be using 99% of the time. Here is an example using fc values:

EXAMPLE 14:

[CODE]

//create event of a newly created object.

//initialize uiz

uiz\_init()

//create our square object

square=uiz\_c(obj\_uiZ\_square)

//setup some variables

square.posinframex=fc;

square.posinframey=fc;

square.posvalx=0.3;

square.posvaly=0.6;

square.posvalwtype=fc;

square.posvalhtype=fc;

square.posvalw=0.3;

square.posvalh=0.1;

//fix our square object.

uiz\_fixgeneralpos(square)

[/CODE]

IMAGE 15

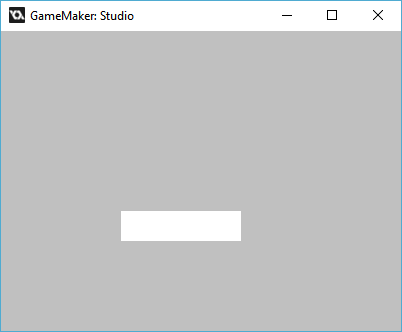
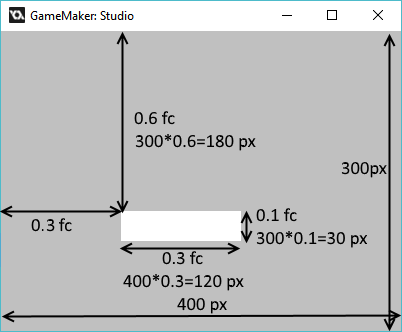


IMAGE 16:



Now that you also know how to size your object, you are able to put your object in any position with any size.

There is one more thing that defines position though. If you look at image 16, you can see that (0.3,0.6) fc gives the position of the top-left corner of the object. You might not always want this tough. You might want the middle of the object to be around the position (0.3,0.6)fc.

Well, this is possible in uiz! There are variables called setpointx and setpointy which do this. Setpointx/y can be the values:

-uiz\_left

-uiz\_middle

-uiz\_right

-uiz\_top

-uiz\_bottom

-uiz\_auto.

So what can you do with this? You can for example set setpointx to uiz\_right and setpointy to uiz\_middle like this:

EXAMPLE 15:

[CODE]

//initialize uiz

uiz\_init()

//create our square object

square=uiz\_c(obj\_uiZ\_square)

//setup some variables

square.posinframex=fc;

square.posinframey=fc;

square.posvalx=0.3;

square.posvaly=0.6;

square.setpointx=uiz\_right

square.setpointy=uiz\_middle

square.posvalwtype=fc;

square.posvalhtype=fc;

square.posvalw=0.3;

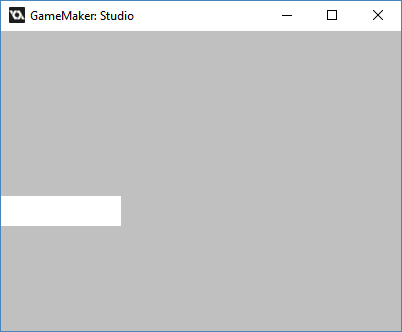
square.posvalh=0.1;

//fix our square object.

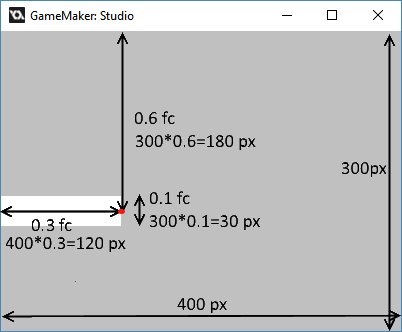
uiz\_fixgeneralpos(square)

[/CODE]

IMAGE 17:



In the following image we can see that the position(0.3,0.6) fc now refers to the right-middle part of the object.



“Do I need to set those variables?” you ask? Well, no you don’t. By default setpointx and setpointy are set to uiz\_auto. If these values are set to uiz\_auto then it will automatically pick a position in the object that fits the posinframex or posinframey the best. For example, for uiz\_snapright, it will automatically pick a value of uiz\_right for posvalx.

I hope you got what posvalx and posvaly does. How you can use this in reality will come to use when you start to code and use uiz in your projects.