Welcome back to the fifth tutorial already! You should have a decent understanding of how to position your objects.

Anyways, today we’ll talk about frames and a little about containment. Let’s get the frames out the way first.

**Frames:** Frames are like an empty canvas something to put objects in. They can provide a layer for you to use in your programming. Frames are important if you want to add structure to your ui, without actually placing objects

**How do I make frames?** Easy, frames are just objects. Just use:

[CODE]

Frame=uiz\_c(obj\_uiZ\_frame)

[/CODE]

An example:

EXAMPLE 16:

[CODE]

//initialize uiz

uiz\_init()

//create our frame

frame=uiz\_c(obj\_uiZ\_frame)

//our parent is the uiz controller object.

//setup some variables

frame.posinframex=px;

frame.posinframey=px;

frame.posvalx=50;

frame.posvaly=50;

frame.posvalwtype=px;

frame.posvalhtype=px;

frame.posvalw=200;

frame.posvalh=200;

//fix our square object.

uiz\_fixgeneralpos(frame)

//create our square object

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

//set the parent

uiz\_setparent(square,frame)

//setup some variables

square.posinframex=fc;

square.posinframey=fc;

square.posvalx=0.3;

square.posvaly=0.6;

square.posvalwtype=px;

square.posvalhtype=px;

square.posvalw=40;

square.posvalh=50;

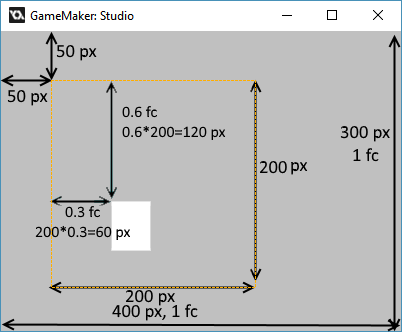
//fix our square object.

uiz\_fixgeneralpos(square)

[/CODE]

Which looks like (image edited to clearify things, the yellow outline is the outline of the frame, which is edited in):

IMAGE 9:



**Containment.**

What is this? This is some stuff can be your greatest friend and your worst enemy at the same time. It is a feature built into every uiz object and makes sure that that object is never drawn at places where it isn’t supposed to be drawn.

Example:

*An image where the window is big enough for the button:*



*And, an image where the window is not big enough for the window:*



In the bottom picture you can see that button isn’t being drawn outside the window.

No example code will be given for this here, since there hasn’t been any explination on windows, but if you want to see it, look for example 17 in the obj\_uiz\_basictutorial object.

Technical details:

There are 2 ways in which uiz “contains” stuff, the default is using a shader, and the secondary option for when shaders aren’t compatible is using surfaces. Because of uiz’s implementation, you’ll need to make some workarounds when you are making your own uiZ objects. But you don’t need to worry about that now, you can read all about it in the tutorials showing you how to make your own objects.

Anyways this is it for this tutorial, until next time.