

Create scripts for RealCADD

This new feature of RealCADD allow you to create your owns tools.

Some examples are giving with RealCADD. So look the scripts of this examples.

RealCADD use RBScript of RealBasic to do this.

So in a RealCADD's script you can use functions of RBScript (download RealBasic to learn more about RBScript) but also some own functions of RealCADD (see the list below)

In RealCADD, objects are store in a list (1 based array). Points of objects (0 based array) are store in pixel at the scale 1/1, but the object have the current scale.

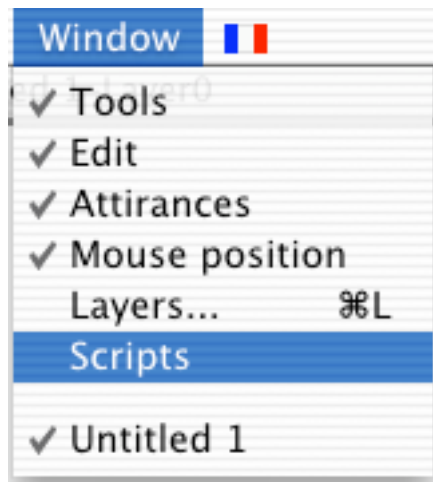
For example if you call `Add_Line(10,10,150,250,1,7,false,false,false,true)` that create a new line with 140 pixels on x and 240 pixels on y.

If you draw in pixels at the scale 1/1, in the edit palette you see $dx = 140$ and $dy = 240$ when you select the line.

But if you draw in meter at the scale 1/50, $dx = 2,46944$ and $dy = 4,23333$ and the line is the same.

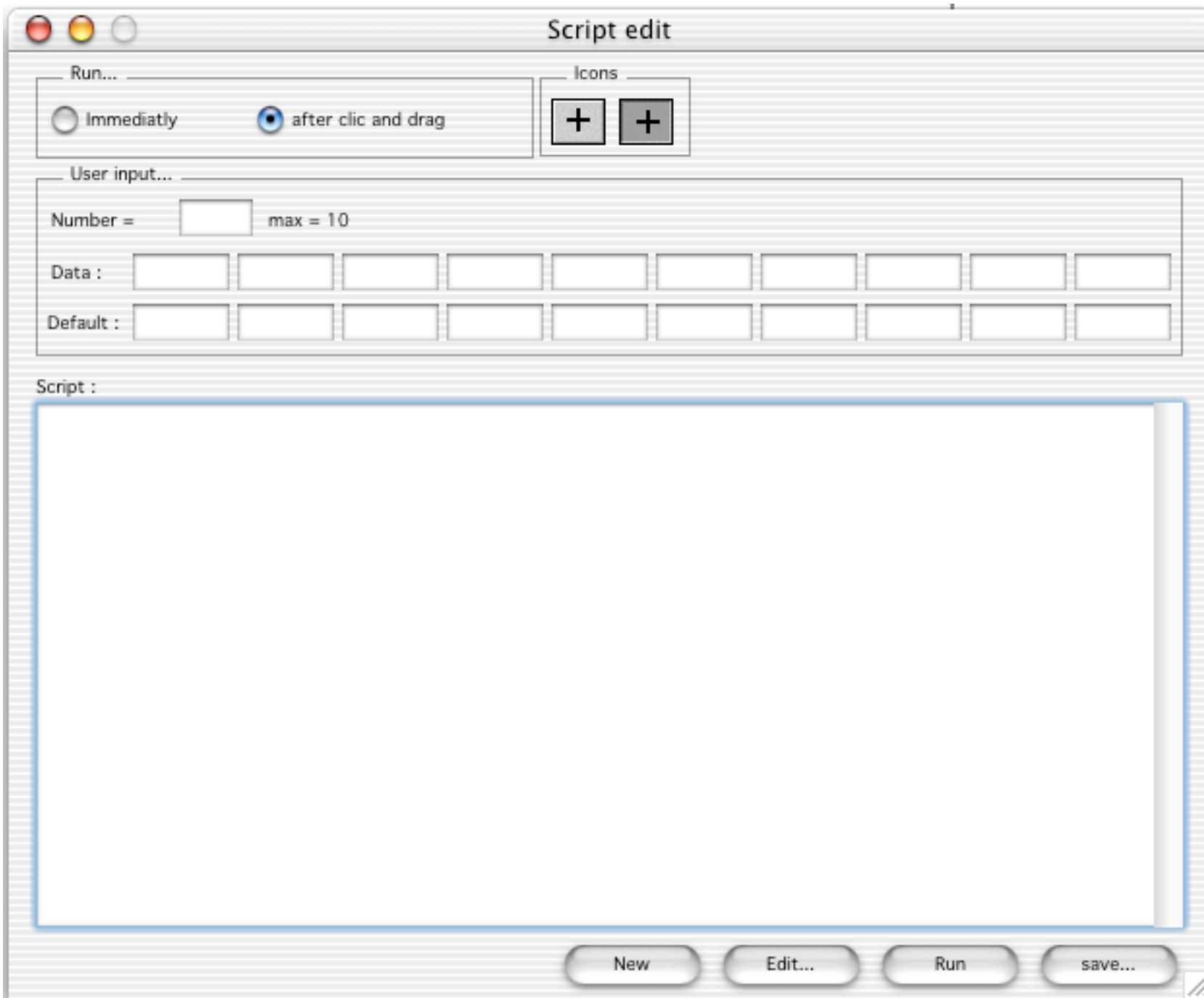
All drawing procedures of RealCADD works in pixels at the scale 1/1.

To display the scripts palette, use the menu Window



Here, the script palette have 7 differents scripts.

You can see a “New” button at the top of the palette. That is to create a new script, a clic on it display this window :



A script can run “Immediately” when you clic on it in the script palette or “after a clic and drag” do by the user. Of course, in this case RealCADD can give you the point the user has clicked.

Each script can have own icons.

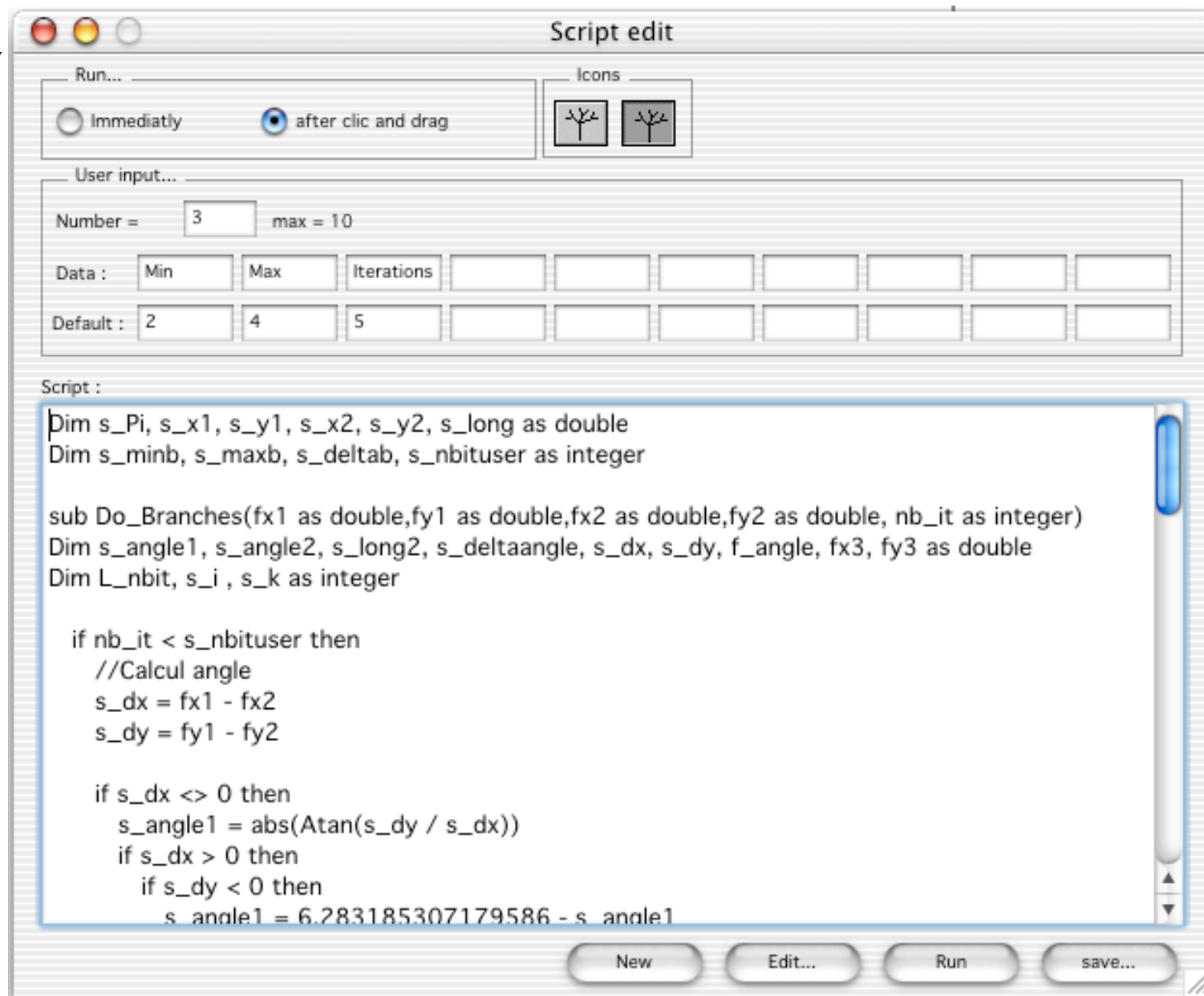
You must draw then and select then by clicking on the icons.

In the script folder, you have two base icons (baseup and basedown) to make icons easier.

A script can also ask the user for some data (10 maxi).

You can also edit a script by clicking on its icons with the option or space key down.

Here the tree script is editing :



RealCADD Procedures or Functions that can be used in scripts:

Sub Add_Line(x1 as double, y1 as double, x2 as double, y2 as double, LineSize as double, CLine as integer, LSelected as boolean, ArrowLeft as boolean, ArrowRight as boolean)

- Add a line in the drawing from point (x1,y1) to point (x2,y2), LineSize is the thickness, CLine the color of the line (like in the color palette : 0 is white, 1 red, 2 yellow, ..., 7 black)

LSelected have no effect in script so can be True or False

ArrowLeft : True = current arrow from Tools palette at the beginning of the line, False = no arrow

ArrowRight : True = current arrow from Tools palette at the end of the line, False = no arrow

Sub Add_Oval(x1 as double, y1 as double, x2 as double, y2 as double, LineSize as double, CLine as integer, CFond as integer, Filled as boolean)

- Add an oval in the drawing from point (x1,y1) to point (x2,y2), LineSize is the thickness, CLine the color of the line, CFond is the back color if Filled is true

Sub Add_Rect(x1 as double, y1 as double, x2 as double, y2 as double, LineSize as double, CLine as integer, CFond as integer, Filled as boolean)

- idem with rectangle

Sub Add_RRect(x1 as double, y1 as double, x2 as double, y2 as double, O1 as double, O2 as double, LineSize as double, CLine as integer, CFond as integer, Filled as boolean)

- idem with rounded rectangle

Sub Add_Arc(x1 as double, y1 as double, rayon as double, startangle as double, arcangle as double, LineSize as double, CLine as integer, CFond as integer, Filled as boolean, LSelected as boolean)

- idem with arc, angle in degrees

Sub Add_String(Str as String, x as double, y as double, Size as integer, pos as integer, isBold as boolean, isItalic as Boolean, isUnderline as boolean, CLine as integer, CFond as integer, Font as String, LSelected as boolean, Filled as boolean, LRotAngle as double)

- Add the string Str in the drawing at the point (x,y) Size is the font size, pos is 0 for left align or 1 for center align or 2 for right, CLine the color of the line, CFond is the back color if Filled is true, Font is the font

Sub Add_Polygon(x1 as double, y1 as double, LineSize as double, CLine as integer, CFond as integer, Filled as boolean, LSelected as boolean)

- Add a polygon in the drawing with only one point (x,y), after call Add_PointToPolygon to add point to the polygon

Sub Add_PointToPolygon(x as double, y as double)

- Add a point (x,y) to the last object created (must be a polygon)

Sub Change_Font(i as integer, Font as String, Size as integer, isBold as boolean, isItalic as boolean, isUnderline as boolean)

- change the attribute of the object i (must be a string object)

Sub Pointille_Object(nb_obj as integer, dl1 as double, de1 as double, dl2 as double, de2 as double)

- draw the object nb_obj with the dotted line dl1,de1,dl2,de2 where dl1 is the first length, de1 is the first space, dl2 the second length, de2 the second space

Sub Delete_Objct(nb_obj as integer)

- Delete the object nb_obj

Sub Clone_Objct(nb_obj as integer, Fl_Dessine as boolean)

- Clone the object nb_obj

Sub DoConvertObjectToLines(nb_obj as integer)

- Convert the object nb_obj to lines

Sub DoConvertObjectToPolygon(nb_obj as integer)

- Convert the object nb_obj to a polygon

Sub BoundRectLastObject()

- when you call BoundRectLastObject, 4 globales variables are used to store the bound rectangle of the last object created :

BoundRectMinx, BoundRectMiny, BoundRectMaxx, BoundRectMaxy as double

Function Global_x(x as double) As Double

- convert x in pixels at the scale 1/1 in the current unite at the current scale

Function Global_y(y as double) As Double

- convert y in pixels at the scale 1/1 in the current unite at the current scale

Function XValueObjet_Point(nb_obj as integer, nb_point as integer) As Double

- return the x value of the point nb_point of the object nb_obj

Function YValueObjet_Point(nb_obj as integer, nb_point as integer) As Double

- return the y value of the point nb_point of the object nb_obj

Global properties :

Dim_Liste as integer : is the index of the last object in the list of objects. You must only read this value. Don't change it !

firstx, firsty, lastx, lasty as double : when you choose to write a script that will run after a clic and drag, RealCADD give you the first point you clic with firstx and firsty and the end point with lastx and lasty

UserInput1 ... UserInput10 as string : if you write a script with user input, you have here the response of the user.

GlobalScript1 ... GlobalScript10 as double : are reserved properties for scripts. You can use it to store values you want to remember between two uses of a script (see script x= y= for example). When the user selects a script tool in the script palette all GlobalScript are initialized to -1000