

```

-- Attention: pas d'accentuation dans les commentaires
-----

-- Surveiller les evenements
-----

-- les variables de l'application
var.monitor("a")
var.monitor("b")
var.monitor("c")
-- Reactualiser la fenetre a l'arrivee de nouvelles donnees
function on.varChange(varlist)
    platform.window:invalidate()
end

-----

-- Gestion de l'affichage sur demande
-----

function on.paint(gc)
    local ww,x,a,b,c
    local result={"",""}
-- nombre de pixels de la fenetre
-- ce nombre permettra de gerer la position des affichages
    ww=platform.window:width()

-- Tracer du tableau de signe
    gc:setColorRGB(100, 0, 255) -- un melange de rouge et bleu
    gc:setFont("serif" , "r", 10)
    gc:drawRect( 5, 5, ww-10, 20)
    gc:drawRect( 5, 25, ww-10, 20)
    gc:drawLine(35, 5, 35, 45)
    x=gc:drawString("x", 6, 6,"top")
    x=gc:drawString("-o", x+26, 6,"top")
    x=gc:drawString("o", x-2, 6,"top")
    x=gc:drawString("+o", ww-25, 6,"top")
    gc:drawString("o", x-2, 6,"top")
    gc:drawString("expr.", 6, 26,"top")

-- Lecture des donnees dans la table des variables
    a=var.recall("a")
    b=var.recall("b")
    c=var.recall("c")
-- Determiner le signe et affichage
    result={etude(a,b,c)}
    gc:drawString(result[1], (ww-gc:getStringWidth(result[1]))/2, 6,"top")
    gc:drawString(result[2], (ww-gc:getStringWidth(result[2]))/2, 26,"top")
end

-----

-- FONCTIONS PROPRES A CE SCRIPT
-----

-- calcul des racines et mise en page de l'affichage
-- les resultats sont donnees en deux lignes de texte
function etude(a,b,c)
    local d=b^2-4*a*c
    local x1,x2=0,0
    local ligne={"",""}
    if d<0 then
        ligne[2]=signe(a)
    elseif d>0 then

```

```

x1=tronc((-b-math.sqrt(d))/(2*a))
x2=tronc((-b+math.sqrt(d))/(2*a))
if x1>x2 then
    x1,x2=x2,x1
end
ligne[1]=x1.." " ..x2
ligne[2]=signe(a).." " 0 " ..signe(-a).." " 0 " ..signe(a)
else
x1=tronc(-b/(2*a))
ligne[1]=" " ..x1.." " "
ligne[2]=signe(a).." " 0 " ..signe(a)
end
return ligne[1],ligne[2]
end

function signe(a)
if a>0 then
return "+"
else
return "-"
end
end

-- arrondi des racines au dixieme par default
function tronc(x)
if x<0 then
return -math.floor(10*-x)/10
else
return math.floor(10*x)/10
end
end
end

```