GPU Cheatsheet V1.2

▶ Colors

color(id)
id = **color**()
pushColor()
popColor()

pal(false,false,[p])
pal(c,false,[p])
pal(c0,c1,0) - *Draw*
pal(c0,c1,1) - *Image*
pal(c0,c1) - *Both*
palt(c,transparent)

colorPalette(id,[r],[g],[b])
r,g,b = **colorPalette**(id)

pushPalette()
popPalette()

▶ Image

img = **image**(lk12Image)
img = **image**(pngImage)

img:draw([x],[y],[r],[sx],[sy],[imgquad])
img:refresh()
w,h = **img:size**()
w = **img:width**()
h = **img:height**()
imgdata = **img:data**()
imgquad = **img:quad**(x,y,[w],[h])
objtype = **img:type**() - *GPU.image*
isit = **img:typeOf**() - *GPU, image, GPU.image, LK12*

▶ ImageData

imgdata = **imagedata**(lk12Image)
imgdata = **imagedata**(pngImage)
imgdata = **imagedata**(w,h)
imgdata = **screenshot**([x],[y],[w],[h])
imgdata = **getLabelImage**()

w,h = **imgdata:size**()
w = **imgdata:width**()
h = **imgdata:height**()
imgdata:map(function(x,y,c) return c end)
imgdata:paste(imgData,[dx],[dy],[sx],[sy],[sw],[sh])
imgquad = **imgdata:quad**(x,y,[w],[h])
img = **imgdata:image**()
lk12Image = **imgdata:encode**()
pngImage = **imgdata:export**()
pngImage = **imgdata:exportOpaque**()
objtype = **imgdata:type**() - *GPU.imagedata*
isit = **imgdata:typeOf**() - *GPU, imagedata, GPU.imagedata, LK12*

▶ Terminal Printing

print(text).
print(text,false) - *No newline.*
print(text,false,true) - *Normal print but on grid.*

printBackspace([bgColor])
printBackspace([bgColor],true) - *No pervious line.*

printCursor([cx],[cy],[bgColor])
cx, cy, bgColor = **printCursor**()

---

lines, maxLength = **wrapText**(text,length)

▶ Dimensions

sw, sh = **screenSize**()
sw = **screenWidth**()
sh = **screenHeight**()

tw, th = **termSize**()
tw = **termWidth**()
th = **termHeight**()

fw,fh = **fontSize**()
fw = **fontWidth**()
fh = **fontHeight**()

▶ Mouse & Cursor

isMDown(b) - *1(left), 2(middle), 3(right)*

name = **cursor**()

cursor(name)
cursor("none")
cursor(name,true) - *Grapped to the pixelated screen.*

cursor(imgdata,name,[hx],[hy])

▶ Matrix Camera

cam("translate",[x],[y])
cam("scale",[sx],[sy])
cam("rotate",[r])
cam("shear",[kx],[ky])
cam() - *Reset*

pushMatrix()
popMatrix()

clearMatrixStack()

▶ Screen State

flip()
didIt = **_hasFlipped**()
isit = **isGifRecording**()

clip(x,y,w,h)
x,y,w,h = **clip**() - *Reset*

patternFill(imgdata)
patternFill() - *Reset*

▶ Drawing

point / **points**(x,y,[x2,y2],...,[c])
line / **lines**(x1,y1,x2,y2,[x3,y3],...,[c])
rect(x,y,w,h,[l],[c])
triangle(x1,y1,x2,y2,x3,y3,[c])
polygon(x1,y1,x2,y2,x3,y3,[x4,y4],...,[c])
circle(x,y,r,[c],[s])
ellipse(x,y,rx,ry,[l],[c],[s])
clear([c])

▶ ImageQuad

imgquad = **quad**(x,y, w,h, srcW,srcH)

srcW,srcH = **imgquad:getTextureDimensions**()

x,y,w,h = **imgquad:getViewport**()

imgquad:setViewport(x,y,w,h)

▶ Printing

print(text, x,y, false,false, [r], [sx],[sy], [ox],[oy], [kx][ky])
print(text, x,y, limit,[align], [r], [sx],[sy], [ox],[oy], [kx][ky]) - *Align can be: "left", "center", "right".*

_systemMessage(msg,[time],[textC],[bgC],[hideInGif])