**CSE 423  
Moinul Hossain Bhuiyan  
ID: 20301002   
Lab Section: 01**

**Assignment 03**

**BRAC UNIVERSITY**

!pip install tensorflow==1.15.0

!pip install -q lucid>=0.2.3

!pip install -q moviepy

!pip install imageio==2.4.1

import numpy as np

import json

import moviepy.editor as mvp

from google.colab import files

import tensorflow.compat.v1.gfile as gfile

import lucid.misc.io.showing as show

from lucid.misc.gl.glcontext import create\_opengl\_context

# Now it's safe to import OpenGL and EGL functions

import OpenGL.GL as gl

from OpenGL.GLU import \*

# create\_opengl\_context() creates GL context that is attached to an

# offscreen surface of specified size. Note that rendering to buffers

# of different size and format is still possible with OpenGL Framebuffers.

#

# Users are expected to directly use EGL calls in case more advanced

# context management is required.

WIDTH, HEIGHT = 1600,1600

create\_opengl\_context((WIDTH, HEIGHT))

# OpenGL context is available here.

print(gl.glGetString(gl.GL\_VERSION))

print(gl.glGetString(gl.GL\_VENDOR))

#print(gl.glGetString(gl.GL\_EXTENSIONS))

def mcl(x\_axis,y\_axis):

  X = x\_axis/500

  Y = y\_axis/500

  gl.glColor3f(0,255,255)

  gl.glBegin(gl.GL\_POINTS)

  gl.glVertex2f(X,Y)

  gl.glEnd()

gl.glClear(gl.GL\_COLOR\_BUFFER\_BIT)

gl.glColor3f(0.184314,0.184314,0.309804)

gl.glPointSize(3)

def mcl\_draw(radius,c\_1,c\_2):

    x\_axis = 0

    y\_axis = radius

    dinit  = 1-radius

    while x\_axis <= y\_axis:

        if dinit<0:

            dinit = dinit+(2\*x\_axis)+3                  #D\_EAST

            x\_axis = x\_axis+1

        else:

            dinit = dinit+2\*(x\_axis-y\_axis)+5           #D\_NorthEast

            x\_axis = x\_axis+1

            y\_axis = y\_axis-1

        mcl(c\_1+x\_axis,c\_2+y\_axis)

        mcl(c\_1+y\_axis,c\_2+x\_axis)

        mcl(c\_1+y\_axis,c\_2-x\_axis)

        mcl(c\_1+x\_axis,c\_2-y\_axis)

        mcl(c\_1-x\_axis,c\_2-y\_axis)

        mcl(c\_1-y\_axis,c\_2-x\_axis)

        mcl(c\_1-y\_axis,c\_2+x\_axis)

        mcl(c\_1-x\_axis,c\_2+y\_axis)

mcl\_draw(60,60,0)

mcl\_draw(60,-60,0)

mcl\_draw(60,0,60)

mcl\_draw(60,0,-60)

mcl\_draw(60,42.42,42.42)

mcl\_draw(60,-42.42,42.42)

mcl\_draw(60,-42.42,-42.42)

mcl\_draw(60,42.42,-42.42)

mcl\_draw(120,0,0)

img\_buf = gl.glReadPixelsub(0, 0, WIDTH, HEIGHT, gl.GL\_RGB, gl.GL\_UNSIGNED\_BYTE)

img = np.frombuffer(img\_buf, np.uint8).reshape(HEIGHT, WIDTH, 3)[::-1]

show.image(img/255.0)

