Tarea 3: Realizar el algoritmo de Bresenham para circunferencia y líneas.

Edgar Daniel Barcenas Martínez. Computación Gráfica e Interacción Humano computadora

El algoritmo para líneas y circunferencia se realizo con Python utilizando un modulo llamado pygame para poder representar los pixeles en una pantalla de 640x400.

Para ejecutarlo instalar el modulo pygame. Recomiendo usar Anaconda, Júpiter.

Algoritmo de Bresenham para circunferencia:

```
BresenhamCircle(0,0,100) #Coordenada X,Coordenada Y, Radio
```

```python
import pygame
import random

ANCHO = 640
ALTO = 400
centrox =320
centroy =200

def BresenhamCircle(xc,yc,r):
    # Window dimensions
    width = 640
    height = 400
    screen = pygame.display.set_mode((width, height))
    clock = pygame.time.Clock()
    running = True
    while running:
        x = 0 + xc
        y = r + yc
        p = 2*(x + 1)*(x + 1) + y*y + (y-1)*(y-1) - 2*r*r
        red = random.randint(0, 255)
        green = random.randint(0, 255)
        blue = random.randint(0, 255)
        screen.set_at((centrox+  x, centroy - y), (red, green, blue))
        screen.set_at((centrox + xc,centroy - yc), (red, green, blue))
        while x<=y:
            x = x + 1 #siempre incrementamos en unidades enteras a x
            if (p < 0) : #pintamos xi+1,yi
                p = p + 4*x + 10 #segun mi ecuacion
                #p = p + 4*x*x + 12*x + 4*y + 10
            else : #pintamos xi+1,yi-1
                y = y - 1
                p = p + 4*(x - y) + 10
                #p = p + y*y + 4*x - 5*y + 8
            screen.set_at((centrox + (x + xc),centroy- y - yc), (red, green, blue))
            screen.set_at((centrox+(-x + xc),centroy-(y + yc)), (red, green, blue))
            screen.set_at((centrox+(x + xc),centroy-(-y + yc)), (red, green, blue))
            screen.set_at((centrox+(-x + xc),centroy-(-y + yc)), (red, green, blue))
            screen.set_at((centrox+(y + xc),centroy-(x + yc)), (red, green, blue))
            screen.set_at((centrox+(-y + xc),centroy-(x + yc)), (red, green, blue))
            screen.set_at((centrox+(y + xc),centroy-(-x + yc)), (red, green, blue))
            screen.set_at((centrox+(-y + xc),centroy-(-x + yc)), (red, green, blue))

        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                running = False

        pygame.display.flip()
        clock.tick(240)


BresenhamCircle(0,0,100)

```

Algoritmo de Bresenham para línea:

```
BresenhamCircleLine(10,10,200,200) #Coordenada X0,Y0, Coordenada X1,Y1
```

```python
import pygame
import random

def BresenhamCircleLine(x0,y0,x1,y1):
    width = 640 # Window dimensions
    height = 400
    screen = pygame.display.set_mode((width, height))
    clock = pygame.time.Clock()
    running = True
    while running:
        dx=x1-x0
        dy=y1-y0
        x=x0
        y=y0
        p=2*dy-dx
        red = random.randint(0, 255)
        green = random.randint(0, 255)
        blue = random.randint(0, 255)
        while x<x1:
            if (p >= 0) : #pintamos xi+1,yi
                screen.set_at((x,y), (red, green, blue))
                y=y+1
                p = p+2*dy-2*dx
            else : #pintamos xi+1,yi-1
                screen.set_at((x,y), (red, green, blue))
                p = p + 2*dy
            x = x + 1
            #p = p + y*y + 4*x - 5*y + 8
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                running = False
        pygame.display.flip()
        clock.tick(240)

BresenhamCircleLine(10,10,200,200)

```

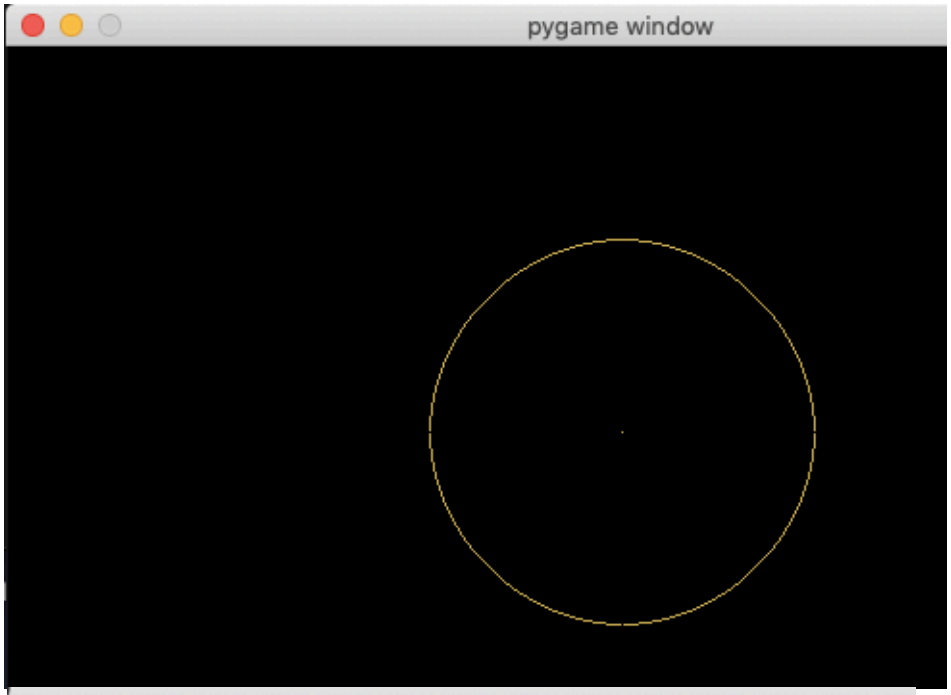
Modo de ejecutar:

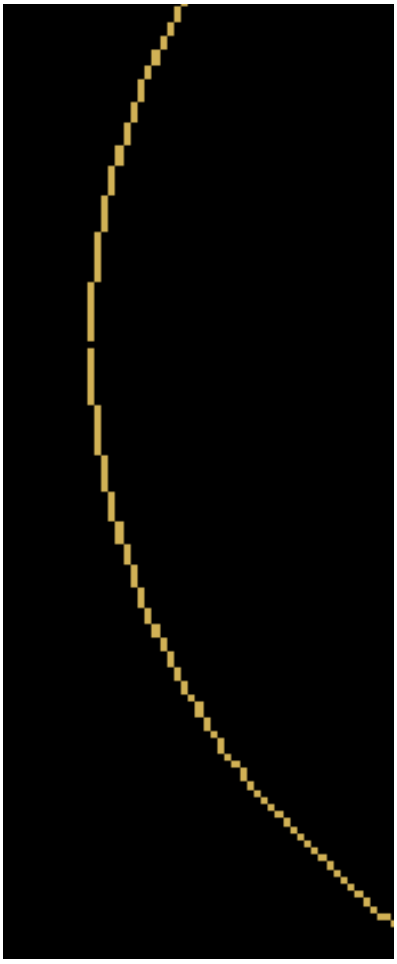
python3 AlgorithmsOfBresenhamCircunference.py
python3 AlgorithmsOfBresenhamLines.py

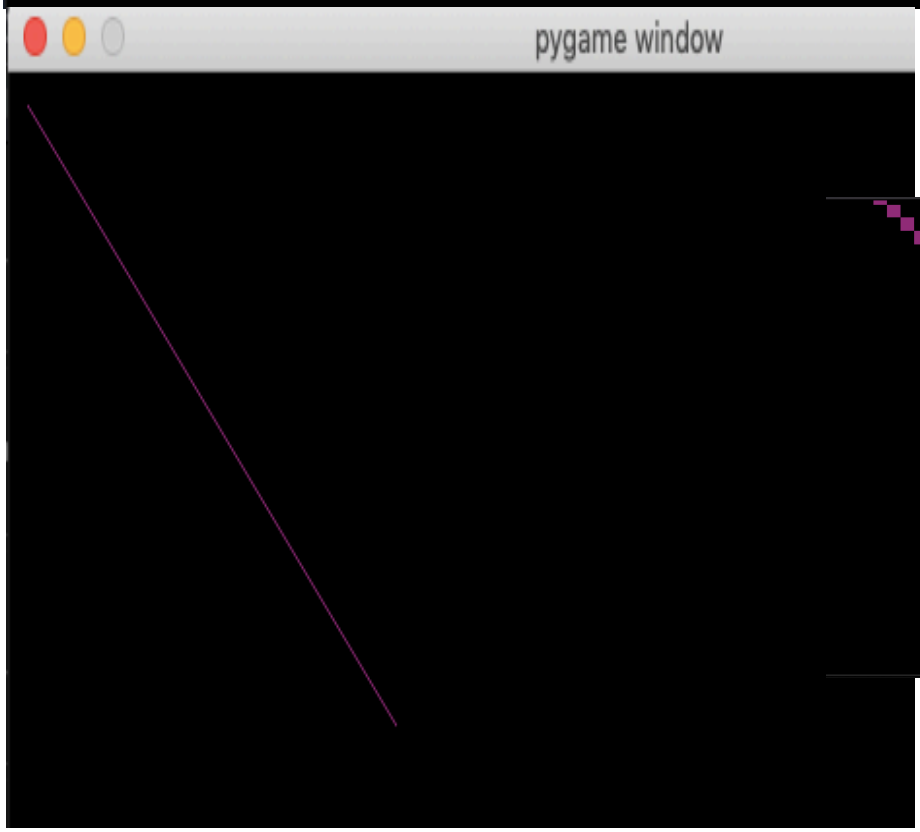
Tarea 3: Realizar el algoritmo de Bresenham para circunferencia y líneas.

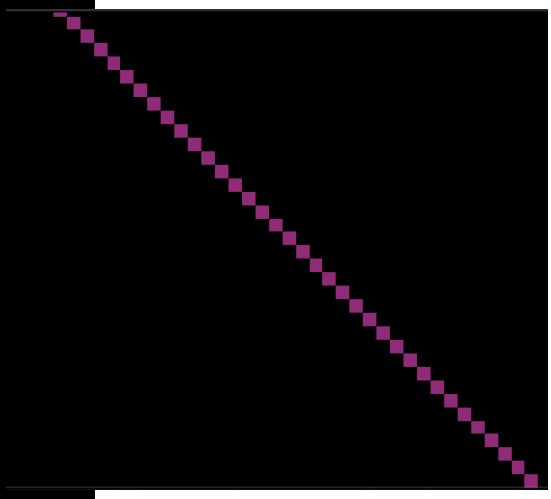
Edgar Daniel Barcenas Martínez. Computación Gráfica e Interacción Humano computadora

Resultados:









Link: https://github.com/DanielBarcenas97/AlgorithmsAndDataStructures