

# Tarea no. 2: Ilusiones Ópticas

Visualización de la Información

Daniel Fragoso Alvarado

IIMAS, UNAM

15 de febrero, 2022

## 1. El paquete Turtle

Imagina una tortuga robot que empieza en las coordenadas (0, 0) en un plano x-y. Después de un `import turtle`, la combinación de una serie de comandos hará que la tortuga se mueva en la pantalla dibujando a su paso.

La clase `TurtleScreen` define una ventana gráfica como base para las tortugas dibujantes. Su constructor necesita una clase `tkinter.Canvas` o una `ScrolledCanvas` como argumento. Se debe usar cuando `turtle` es usado como parte de una aplicación.

## 2. Ejercicio 1

Para este ejercicio hicimos uso de la función `draw_line` que vimos en clase. Para esta primera imagen, consideramos la simetría de las líneas para poder construir estas, es decir que extendimos todas las líneas desde su cuadrante hasta el opuesto haciendo que todas las líneas pasan por el origen.

**Código:** Las funciones utilizadas para dibujar una línea no están integradas, para simplificar el documento, sin embargo se pueden consultar en el archivo `T2_a.py` anexo a este documento. Básicamente consiste en un ciclo `for` que recorre todo el canvas, desde un extremo al contrario, el cual se recorre cada 50 píxeles.

```
1 # Vamos a definir a 'Mylapiz' como un objeto de la clase Turtle
2 mylapiz = ts.Turtle()
3
4 # Vamos a crear un ciclo que nos permita crear varias veces una linea
5
6 for x in range(-400, 400, 50):
7     draw_line(x, 200, -x, -200, 'black')
8
9 for y in [-100, 100]:
10    draw_line(400, y, -400, y, 'purple', size = 3)
```

```
11  
12 ts.getscreen()  
13  
14 # Guardar  
15 ts.getcanvas().postscript(file="Tareas/T2/t2_a.eps")
```

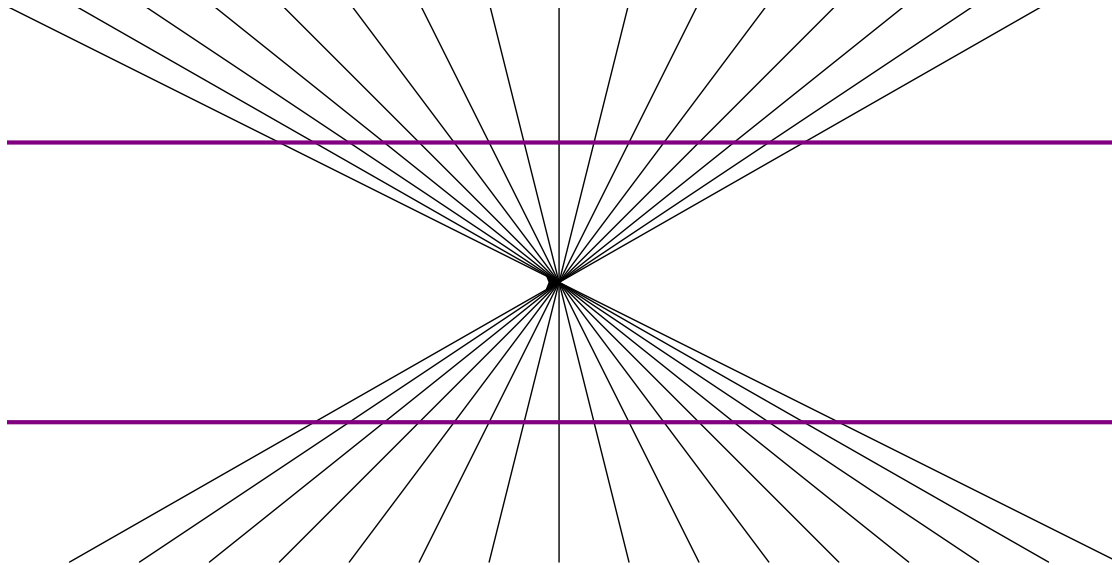


Figura 1: Primer ilusión óptica resultado del código T2\_a.py

**¿Son ambas líneas purpura perfectamente rectas?** Dado la construcción de nuestra figura podemos asegurar sin miedo a equivocarnos que las líneas en efecto son completamente rectas. Sin embargo las perspectivas del resto de las líneas, distorsionan nuestra percepción y las percibimos con una ligera curvatura.

### 3. Ejercicio 2

Para este ejercicio hicimos uso de la función `draw_square`, `draw_circle` que vimos en clase. Para esta imagen primero creamos un rectángulo, con una pequeña variación de la función para los cuadrados, la cual se puede consultar en los archivos anexos, el color de este rectángulo que funge como base es gris. Posteriormente dibujamos una serie de cuadrados, de 10 px menos que el periodo que recorría el lápiz alzado, con el fin de dejar un margen de 10 px.

Para finalizar, dibujamos una serie de círculos de 10 px de diámetro con el fin de que estuvieran centrados, entre los espacios de los cuadrados, para esto el ciclo `for` en el paso de cuadrado a cuadrado, sufrió de un ajuste de 5px para que pudieran estar correctamente centrados.

```
1 mylapiz = ts.Turtle()  
2  
3 # rectangulo inicial
```

```
4
5 draw_rectangle(-420, -220, 830, 430, 'grey')
6
7 # Vamos a reducir 10px de cada lado para generar una margen
8
9 for x in range(-400, 400, 80):
10     for y in range(-200, 200, 80):
11         # El cuadrado que queremos dibujar será más chico para obtener una
           margen de 10 px
12         draw_square(x, y, 70, 'black')
13
14 # Generaremos los círculos
15
16 for x in range(-325, 320, 80):
17     for y in range(-125, 120, 80):
18         draw_circle(x, y, 10, 'white')
19
20 ts.end_fill()
21
22 # Guardar
23 ts.getcanvas().postscript(file="Tareas/T2/t2_b.eps")
```

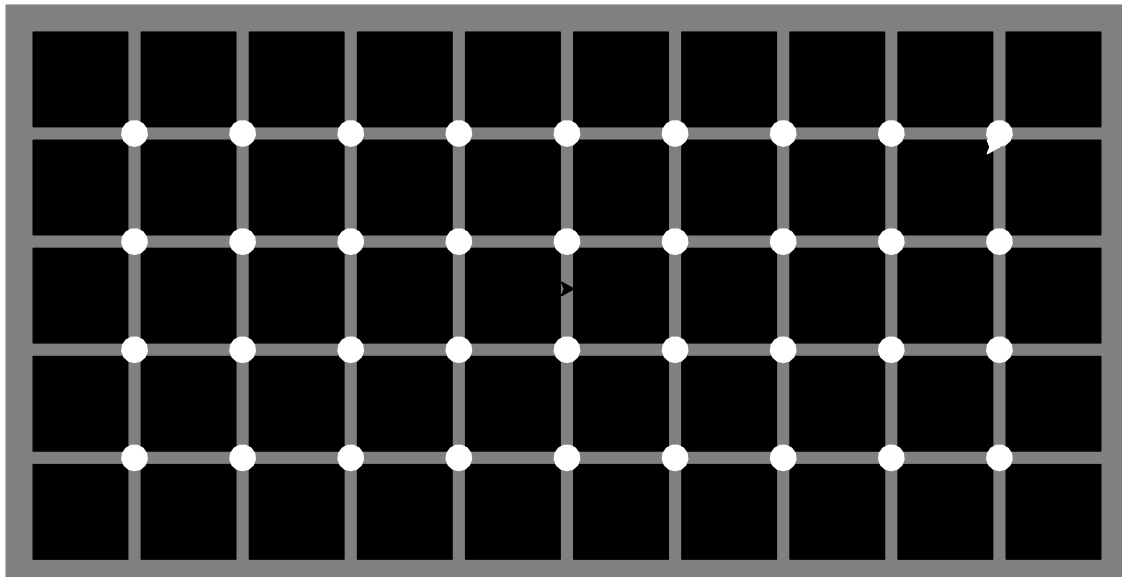


Figura 2: Primer ilusión óptica resultado del código T2\_b.py

**¿Todos los puntos blancos son realmente blancos?** Una vez más dada la construcción de nuestra figura sí podemos asegurar que se tratan de únicamente círculos blancos, sin embargo los contrastes de grises de los cuadrados y el fondo, generan la percepción de un cambio de color aunque estos no sea así.

## 4. Extra

Como una figura extra, solo para practicar, generamos una estrella a partir de únicamente líneas rectas, siendo un ejercicio común en las escuelas. El código se encuentra en el archivo `T2_c.py`, aunque básicamente consiste en 8 ciclos que trazan líneas rectas:

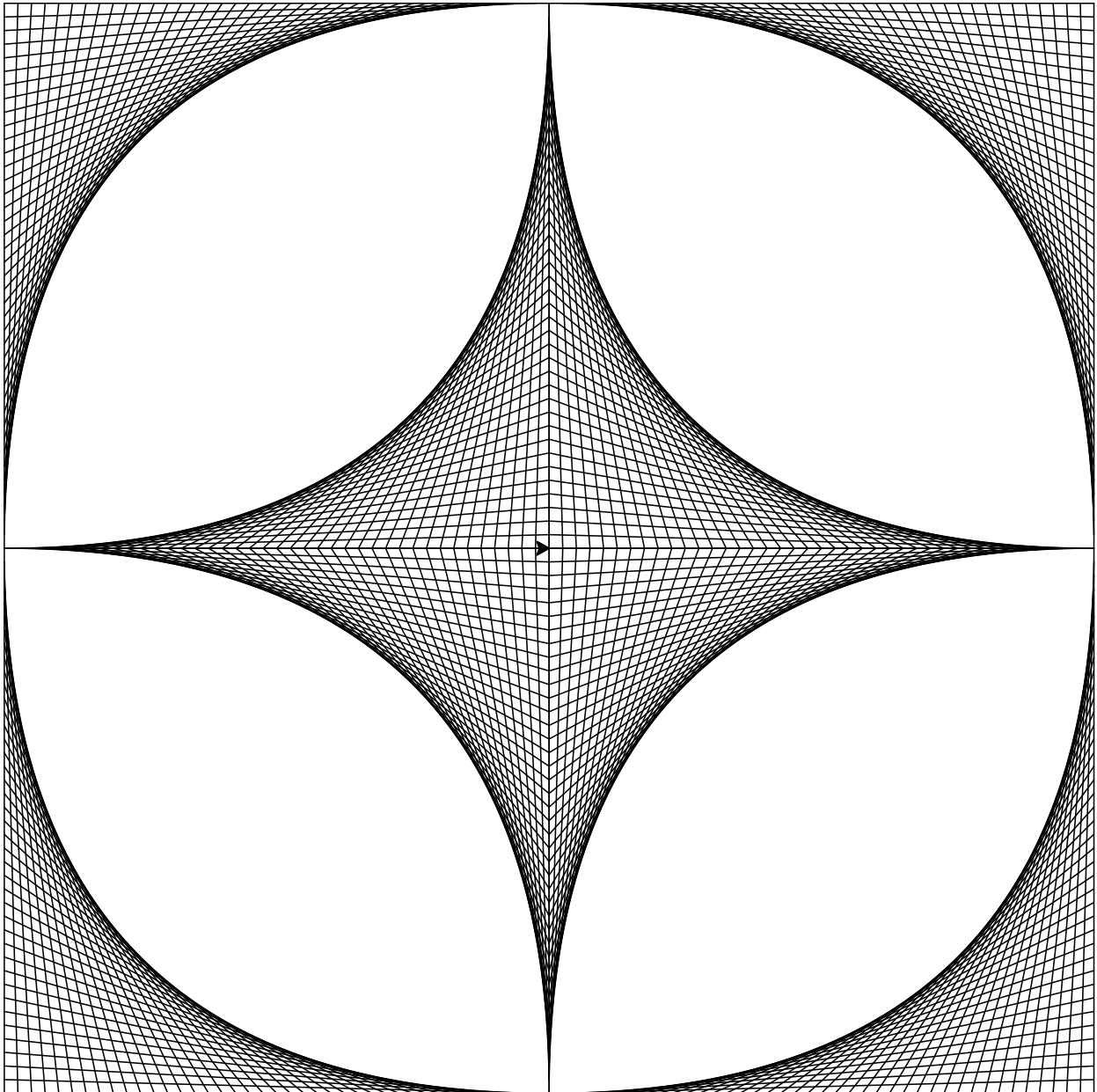


Figura 3: Resultado del programa `T2_c.py`