

Evidencia de proyecto final: Videojuegos

Valeria Martínez Martínez
A01782413

Paint

```
#Circle function, named differently since Turtle already includes a circle() function
def mod_circle(start, end):
    """Draw circle using two points as the diameter."""
    center = (start + end) / 2 #The is calculated to ensure the middle of the circle is drawn between the 2 points
    radius = abs(end - start) / 2 #This is calculated to move the pen to the bottom edge of a vertical diameter
    begin_fill()

    up()
    goto(center.x, center.y - radius) #The pen is placed on the bottom edge of the diameter
    down()
    circle(radius) #Draws the circle

    end_fill()
```

```
def rectangle(start, end):
    """Draw rectangle from start to end."""
    up()
    goto(start.x, start.y) #The pen is placed on the first chosen point
    down()
    begin_fill()

    #Loop that runs twice to draw the rectangle, by moving first on the x-axis and then on the y-axis
    for _ in range(2):
        forward(end.x - start.x)
        left(90)
        forward(end.y - start.y)
        left(90)

    end_fill()
```

Se agregó la función circle, calculé el centro para poder reajustar el dibujo a que ocupara bien el diámetro porque la función de la librería toma como entrada el radio y al intentarlo multiplicar x2 quedaba un círculo demasiado grande.

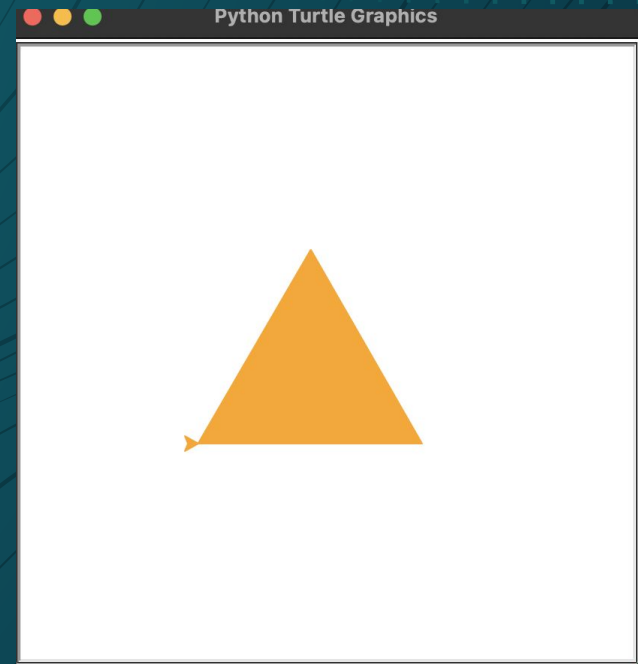
En la de rectángulo fue más sencillo, solo que se moviera del inicio sobre x al final sobre x, y del inicio de y al final sobre y, girar al terminar cada movimiento y repetirlo 2 veces.

Paint

```
def triangle(start, end):  
    """Draw triangle from start to end."""  
    side_length = abs(end - start) #The length of the triangle's sides is calculated (it's the same since it's equilateral)  
    up()  
    goto(start.x, start.y) #The pen is placed on the first chosen point  
    down()  
    begin_fill()  
  
    #Loop that runs three times, draws the triangle side by side from the bottom left corner  
    for _ in range(3):  
        forward(side_length)  
        left(120)  
  
    end_fill()
```

Para la función de triángulo fue muy similar a la de rectángulo, solo fue mover sobre la línea, girar 120 para que quedaran todos los ángulos y lados iguales y repetirlo 3 veces.

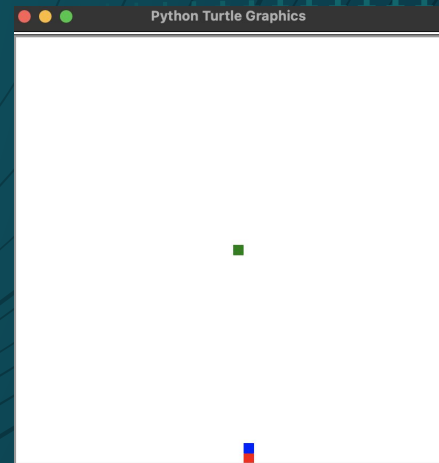
Aquí una captura del color que agregué extra (naranja) siendo usado para dibujar el triángulo equilátero.



Snake

```
colors = ['green', 'blue', 'yellow', 'purple', 'orange'] #List of colors to be chosen at random, excluding red
snake_color = choice(colors) #Choose a random color for the snake
colors.remove(snake_color) # Remove the chosen color from the list to ensure it's not picked for the food color
food_color = choice(colors) # Choose a random color for the food from the remaining colors
```

```
#Function to change the food one step at a time, at random
def move_food():
    """Move food one step randomly."""
    if inside(food):
        food.x += randrange(-10, 11, 10)
        food.y += randrange(-10, 11, 10)
    else:
        food.x = randrange(-15, 16, 10)
        food.y = randrange(-15, 16, 10)
```



Para que los colores no se repitieran inicié unas opciones como lista, dejé que se escogiera primero la de la serpiente, luego quité esa opción de la lista y ya sobre esa nueva lista modificada que se escogiera la de la comida. Igual dejo una captura donde se ve la diferencia de los colores.

Además la función para que se mueva la comida de paso en paso de manera aleatoria se encuentra aquí, y se asegura de que no se salga de los límites.

Pacman

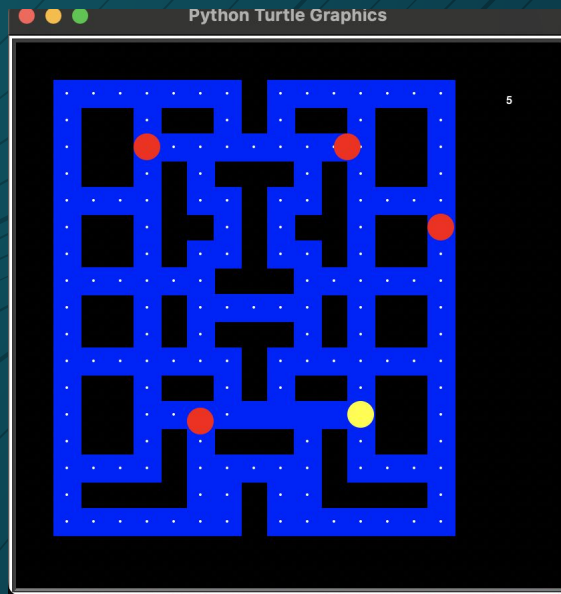
```
#Added a loop to check if the ghosts are colliding with each other, and if they are, they change direction
for j in range(len(ghosts)):
    if i != j and abs(ghosts[i][0] - ghosts[j][0]) < 20:
        ghosts[i][1] *= -1
        ghosts[j][1] *= -1
```

```
ontimer(move, 30) #Changed the time parameter so the ghosts can move quicker
```

El principal cambio fue añadir ese loop para que si los fantasmas chocaban entre sí se fueran para el lado opuesto.

También cambié el parámetro de velocidad.

Igual dejo captura del tablero ya modificado.



Cannon

```
#By removing the loop that checks if the target is inside the screen, the game will never end
"""for target in targets:
    if not inside(target):
        return"""

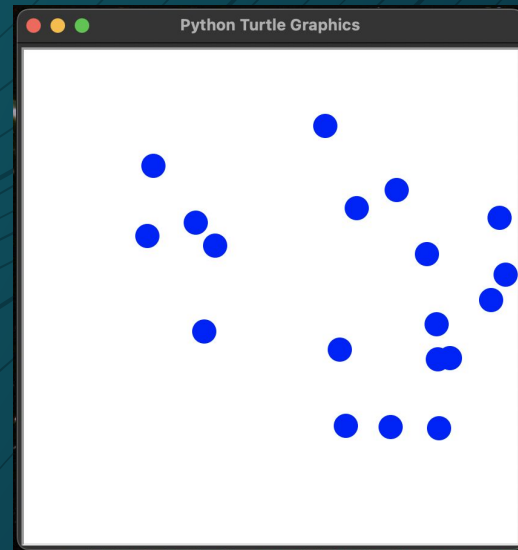
ontimer(move, 20) #Changed the time parameter from 50 so the ball moves quicker
```

```
#This condition checks if the target is outside the screen, if it is then it is reset (moved to the other side of the screen)
if target.x < -200 or target.x > 200:
    target.x *= -1
```

Los principales cambios fueron la eliminación del loop, que dejé comentado para que se viera la diferencia y cambiar el parámetro de ontimer para que fuera más corto.

También la función en la captura de abajo, hace que una vez que se sale de límites uno de los targets continúe con su trayectoria saliendo otra vez desde la orilla izquierda.

En la captura se puede ver que ya hay más pelotas porque al cambiar la velocidad de movimiento implica que salen más seguido a comparación de antes



Memory

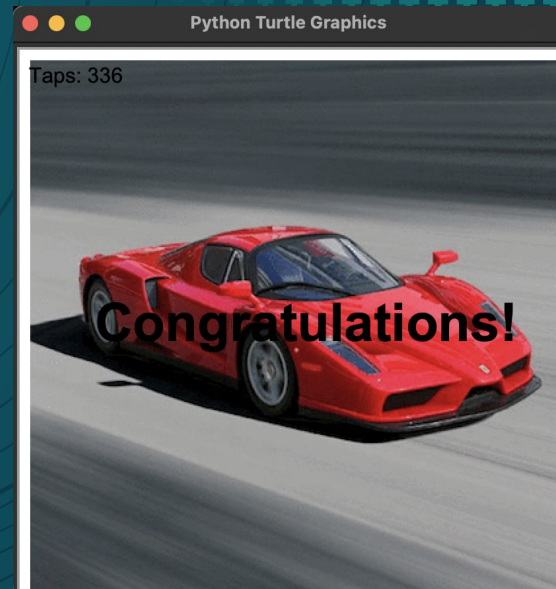
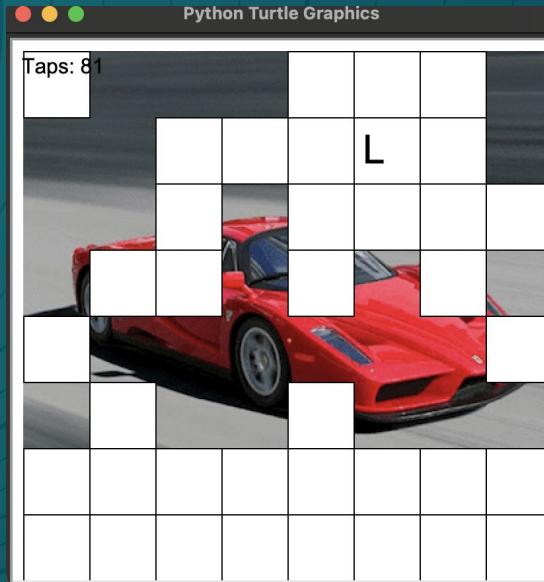
```
taps = 0 # Initialize counter for taps

# Function to count the taps
def count_taps():
    global taps
    taps += 1

def tap(x, y):
    """Update mark and hidden tiles based on tap."""
    spot = index(x, y)
    mark = state['mark']
    count_taps() # Call the tap counting function at each tap

    if mark is None or mark == spot or tiles[mark] != tiles[spot]:
        state['mark'] = spot
    else:
        hide[spot] = False
        hide[mark] = False
        state['mark'] = None

# Function to display the tap count
def show_taps():
    up()
    goto(-200, 180)
    color('black')
    write(f'Taps: {taps}', font=('Arial', 16, 'normal'))
```



Los snippets de la izquierda son todos los cambios que hice para que se pudieran contar y mostrar los taps en el juego. Fue agregar la variable para el contador, la función que contara, incluirla para que se llamara cada vez que se hacía un tap y aparte crear la función que despliega ese número.

Las capturas demuestran estos cambios y además el de utilizar algo diferente de números, que en lo personal sí me ayudó a hacer menos taps, centrar eso en los tiles y el mensaje de que se detectó que ya se voltearon todos los tiles felicitando al jugador.