

The background is a solid pink color. It is decorated with various hand-drawn geometric shapes in white and black. These include a dashed line in the top left, a white triangle in the top center, a black zigzag line in the top right, a white circle in the top right, two parallel black lines in the top right, a white triangle in the top right, a black circle in the bottom left, a white triangle in the bottom center, a dashed line in the bottom center, a black circle in the bottom right, and a white circle in the bottom right.

Welcome!

We'll get started shortly ...



CS 49 Section

Week 9

Surajit A Bose





Agenda



- Logistics and check-ins
- Review of lecture concepts
 - Animation loop
 - Graphics functions
- Section Problem : [Scribble](#)




Logistics






How to get hold of me / get help+

- The [class forum](#). Feel free not only to ask, but also to answer questions!
 - Surajit's office hours:
 - Fridays 12 noon–1p, directly after section
 - By appointment on [Zoom](#)
 - [Lane's office hours](#)
 - Canvas inbox or Pronto inbox for Lane
 - Canvas inbox (preferred) or Pronto for Surajit
 - [Sina's support section](#), Fridays 1p–2p on [Zoom](#)
 - Email bozesurajit@fhda.edu, 24 hr turnaround
 - [Online](#) or [in-person](#) tutoring via the STEM center (Room 4213)
 - The section [GitHub repo](#) has lecture and section slides and solutions
- 



Check In

- Any questions about:
 - Medical test simulation problem
 - Program information flow (parameters, arguments, returns)
 - Concepts from previous weeks
 - Any homework exercises or problems
 - Please take the Zoom survey!
- 



Lecture Review



Animation Loop

- Animations are effected by rendering the same graphic over and over in incrementally different positions
- First, get the canvas, the shape dimensions, and the starting coordinates
- Then, in a loop (while some condition is true):
 - Draw the shape at the current coordinates
 - Get the new coordinates (incrementally different from original)
 - Pause for a small delay so the viewer's eye can track the shape to its new position



Animation Loop

```
from graphics import Canvas
```

```
CANVAS_WIDTH = 400
```

```
CANVAS_HEIGHT = 400
```

```
VELOCITY = 2           # the rate at which the animation changes
```

```
DELAY = 0.05           # the pause between successive renderings
```

```
START_SIZE = 0
```

```
END_SIZE = 200
```



Animation Loop

+



```
def embiggen_circle():  
    canvas = Canvas(CANVAS_WIDTH, CANVAS_HEIGHT)  
    size = START_SIZE  
    while size <= END_SIZE:  
        start_x = CANVAS_WIDTH / 2 - size / 2  
        start_y = CANVAS_HEIGHT / 2 - size / 2  
        circle = canvas.create_oval(start_x, start_y,  
                                    start_x + size, start_y + size, 'purple')  
        size += VELOCITY  
        time.sleep(DELAY)
```



Any Questions?

Graphics Functions

+



```
# get the x and y location of the mouse
```

```
mouse_x = canvas.get_mouse_x()
```

```
mouse_y = canvas.get_mouse_y()
```

```
# move shape to some new coordinates
```

```
canvas.moveto(shape, new_x, new_y)
```





Graphics Functions

Write a function to highlight the location of the mouse, keeping track of it wherever it moves.


```
from graphics import Canvas
```

```
# Constants
```

```
CANVAS_SIZE = 400
```

```
SQUARE_SIZE = 40
```

```
DELAY = 0.01
```



Graphics Functions

+



```
def highlight_mouse():  
    canvas = Canvas(CANVAS_SIZE, CANVAS_SIZE)  
    square = canvas.create_rectangle(0, 0, SQUARE_SIZE,  
                                     SQUARE_SIZE, 'pink')  
    while True:  
        mouse_x = canvas.get_mouse_x()  
        mouse_y = canvas.get_mouse_y()  
        canvas.moveto(square, mouse_x - SQUARE_SIZE / 2,  
                      mouse_y - SQUARE_SIZE / 2)  
        time.sleep(DELAY)
```






Section problem: Scribble

<https://codeinplace.stanford.edu/cs49-w24/ide/a/sectionscribble>





Scribble

- Draw a circle wherever the mouse pointer is on the screen
 - As the user moves the mouse within the canvas, a circle of size **CIRCLE_SIZE** is drawn with the mouse position as the top left of the bounding box
 - Other given constants: **CANVAS_WIDTH**, **CANVAS_HEIGHT**, **DELAY**
 - Bonus:
 - Check that the mouse is within the canvas before drawing the circle
 - Draw the circles in random colors
- 

The background is a solid orange color. It is decorated with various hand-drawn geometric shapes in white and black. These include a dashed line in the top left, a white triangle in the top center, a black zigzag line in the top right, a white circle in the top right, two parallel black lines in the top right, a white triangle in the top right, a black plus sign in the bottom left, a white circle in the bottom center, a white triangle in the bottom center, a black plus sign in the bottom center, a black circle in the bottom center, and a white circle in the bottom right.

Questions Before We Begin?

The background is a solid pink color. It is decorated with various hand-drawn geometric shapes in white and black. These include a dashed line in the top left, a white triangle in the top center, a black zigzag line in the top right, a white circle in the top right, two parallel black lines in the top right, a white triangle in the top right, a large black circle on the right edge, a white triangle in the bottom left, a black plus sign in the bottom left, a white circle in the bottom center, a white triangle in the bottom center, a dashed line in the bottom center, a black circle in the bottom center, and a white circle in the bottom right.

That's all, folks!

Next up: Lists!