Welcome!

We'll get started shortly. Please take the Zoom poll in the meanwhile!

CS 49 Week 8

Surajit A. Bose

Agenda

- Old business
  - Solution to Medical Test Simulator
- New business
  - Animation loops
  - Worked example: Embiggen Circle
  - Graphic functions
  - Worked example: Highlight Mouse
  - Section problem: Scribble

Animations

Animation Loops

- 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

Example: Embiggen Circle (Slide 1 of 2)

- To animate a circle growing on the canvas, first, set up some constants:

```
from graphics import Canvas

CANVAS_WIDTH = 400
CANVAS_HEIGHT = 400
START_SIZE = 0     # starting size of circle
END_SIZE = 200     # ending size of circle
CHANGE_RATE = 2    # rate at which animation changes, in pixels
DELAY = 0.02       # pause after each rendering, in seconds
```

Example: Embiggen Circle (Slide 2 of 2)

- Next, create the canvas, set the start size, and set up the 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 += CHANGE_RATE
        time.sleep(DELAY)
```

Graphics Functions

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)
```

Example: Highlight_Mouse (Slide 1 of 2)

- To highlight the mouse position wherever it moves, first set up the constants:

```
from graphics import Canvas

CANVAS_SIZE = 400
SQUARE_SIZE = 40        # size of square that highlights mouse
DELAY = 0.01            # check mouse position each 0.01 second
```

- The animation loop here needs to run for as long as the program runs, so we use `while True`

Example: Highlight_Mouse (Slide 2 of 2)

```
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

Section Problem: 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`**
- Hint: the code for **`highlight_mouse()`** should come in handy
- Bonus:
    - Check that the mouse is within the canvas before drawing the circle
    - Draw the circles in random colors
    - Hint: we implemented both these in our solution to **`random_circles()`**

That's all, folks!

Next up: Lists and Dictionaries!