Joy in Creating

As the child delights in his mud pie, so the adult enjoys building things, especially things of his own design. I think this delight must be an image of God's delight in making things, a delight shown in the distinctness and newness of each leaf and each snowflake.

- F. P. Brooks, Jr. *The Mythical Man-Month*, 1975

Canvas

- A widget for displaying shapes
  - rectangles
  - ovals/circles
  - arcs
  - polygons (shapes of your own choosing)
  - lines

```
canvas = Canvas(window, width = 600, height = 600,
                bg = 'white')
```

- References:
  - http://infohost.nmt.edu/tcc/help/pubs/tkinter/web/canvas.html
  - http://www.python-course.eu/tkinter_canvas.php

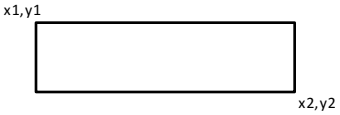
Line

```
canvas.create_line(x1, y1, x2, y2)
```

- Common optional arguments:
  - **arrow** - `start`, `end` or `both`
  - **tags** - to refer to later
  - **fill** - inside color
  - **activefill** - color when mouse is over (if different from fill)
  - **width** - thickness of line

Rectangle

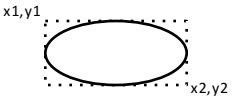
```
canvas.create_rectangle(x1, y1, x2, y2)
```



- Common optional arguments:
  - **tags** - to refer to later
  - **fill** - inside color
  - **activefill** - color when mouse is over (if different from fill)
  - **width** - thickness of line

Oval

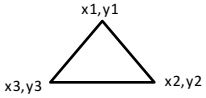
```
canvas.create_oval(x1, y1, x2, y2)
```



- Common optional arguments:
  - **tags** - to refer to later
  - **fill** - inside color
  - **activefill** - color when mouse is over (if different from fill)
  - **width** - thickness of line

Polygon

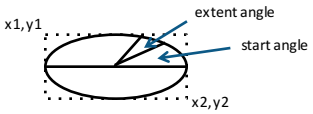
```
canvas.create_polygon(x1, y1, x2, y2, x3, y3)
```



- Can use as many pairs as desired
  - points are joined in order given
- Common optional arguments:
  - **tags** - to refer to later
  - **fill** - inside color
  - **activefill** - color when mouse is over (if different from fill)
  - **width** - thickness of line

Arc

```
canvas.create_arc(x1, y1, x2, y2, extent = a1, start = a2)
```



- Common optional arguments:
  - **`tags`** - to refer to later
  - **`fill`** - inside color
  - **`activefill`** - color when mouse is over (if different from fill)
  - **`width`** - thickness of line

Text

```
canvas.create_text(x1, y1, text = "ABCDE")
```

x1, y1 indicates the center of the invisible box surrounding the text

- Common optional arguments:
  - **`tags`** - to refer to later
  - **`fill`** - inside color
  - **`activefill`** - color when mouse is over (if different from fill)
  - **`width`** - thickness of line

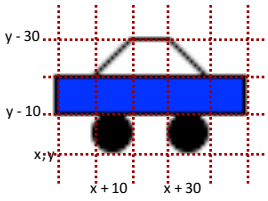
Removing from canvas

- If you tagged an object when created can use **`delete`** to remove it from the canvas
  - e.g. `canvas.delete("tagName")`

Example

- Create a program with two radio buttons:
  - If the "On" button is clicked, a house appears on the screen.
  - If the "Off" button is clicked, the house disappears

Draw Car



Draw Car

```
def drawCar(canvas, x, y):
    canvas.create_oval(x+10, y-10, x+20, y, fill = 'black') #wheel one
    canvas.create_oval(x+30, y-10, x+40, y, fill = 'black') #wheel two
    canvas.create_rectangle(x, y-20, x+50, y-10, fill = "blue") #body
    canvas.create_polygon(x+10, y-20, x+20, y-30,
                          x+30, y-30, x+40, y-20,
                          outline = 'black', fill = 'white') #windows
```

Animation

- An illusion created by presenting carefully designed image *frames* at a sufficient *frame rate*

Tkinter Animation

- Continuously repeat these steps:
  - Make a small change to the canvas
  - Briefly pause the program
  - Update the canvas
- Pause - **`canvas.after(time)`**
- Update - **`canvas.update()`**

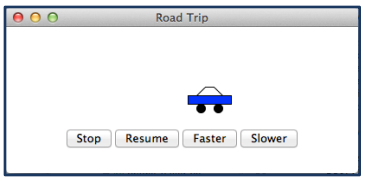
```
#... drawCar defined above ...
class CarAnimation:
    def __init__(self, window):
        canvas = Canvas(window, width = 200, height = 110)
        canvas.pack()

        #Animation variables
        carX = -50
        speedX = 5

        while True:
            canvas.delete(ALL)
            carX += speedX
            drawCar(self._canvas, carX, 100)
            canvas.after(50)
            canvas.update()

if __name__ == '__main__':
    root = Tk()
    root.title('Road Trip')
    app = CarAnimation(root)
    root.mainloop()
```

Animation Controller



Expanding Circle

- Create an animation that draws a circle centered on the canvas and then the circle appears to grow.

Random Color

```
# Convert an integer to a single hex digit in a character
def toHexChar(hexValue):
    if 0 <= hexValue <= 9:
        return chr(hexValue + ord('0'))
    else:  # 10 <= hexValue <= 15
        return chr(hexValue - 10 + ord('A'))


# Return a random color string in the form #RRGGBB
def getRandomColor():
    color = "#"
    for i in range(6):
        # Add a random digit
        color += toHexChar(randint(0, 15))
    return color
```

Raindrops

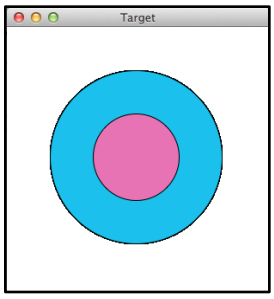
Events

- We have seen that clicking a button causes an event to be fired
- Mouse clicks/movement and key presses also cause events to be fired

Mouse events

- <Button-1> <Button-2> <Button-3>
  - Left, middle and right mouse buttons.
  - When the mouse is pressed over the widget, Tkinter automatically grabs the x and y of the pointer
    - event.x, event.y
- <ButtonReleased-i>
- <Double-Button-i>
  - Double click button 1 2 or 3
- <Leave>
  - An event occurs when the mouse pointer leaves the widget
- <Bi-Motion>
  - An event occurs when a mouse button is moved while being held down on the widget

Target Demo



Key Events

- <Enter>
  - An event occurs when Enter key is pressed
- <Key>
  - An event occurs when any key is pressed
- Event properties:
  - char - character entered
  - keycode - unicode
  - keysym - symbol for character
- Must setfocus on widget so it can receive keyboard input
  - E.g. `canvas.focus_set()`

Lines with arrow keys