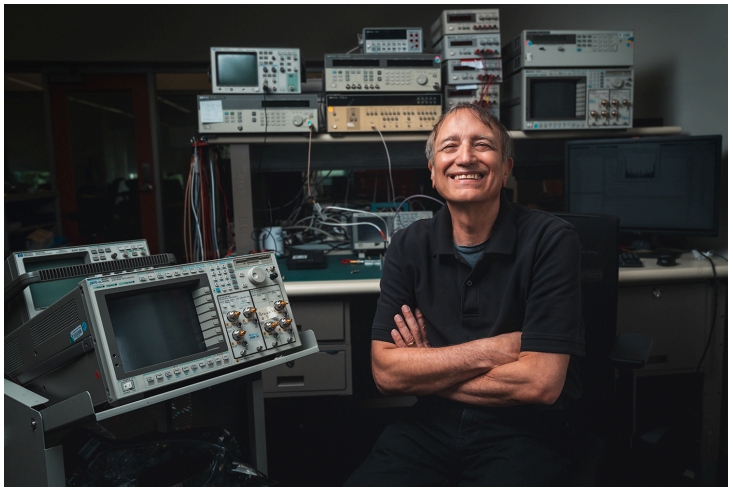
Animation

Chris Piech

CS Bridge 2020

Turing Award Winner Talks to CS106A

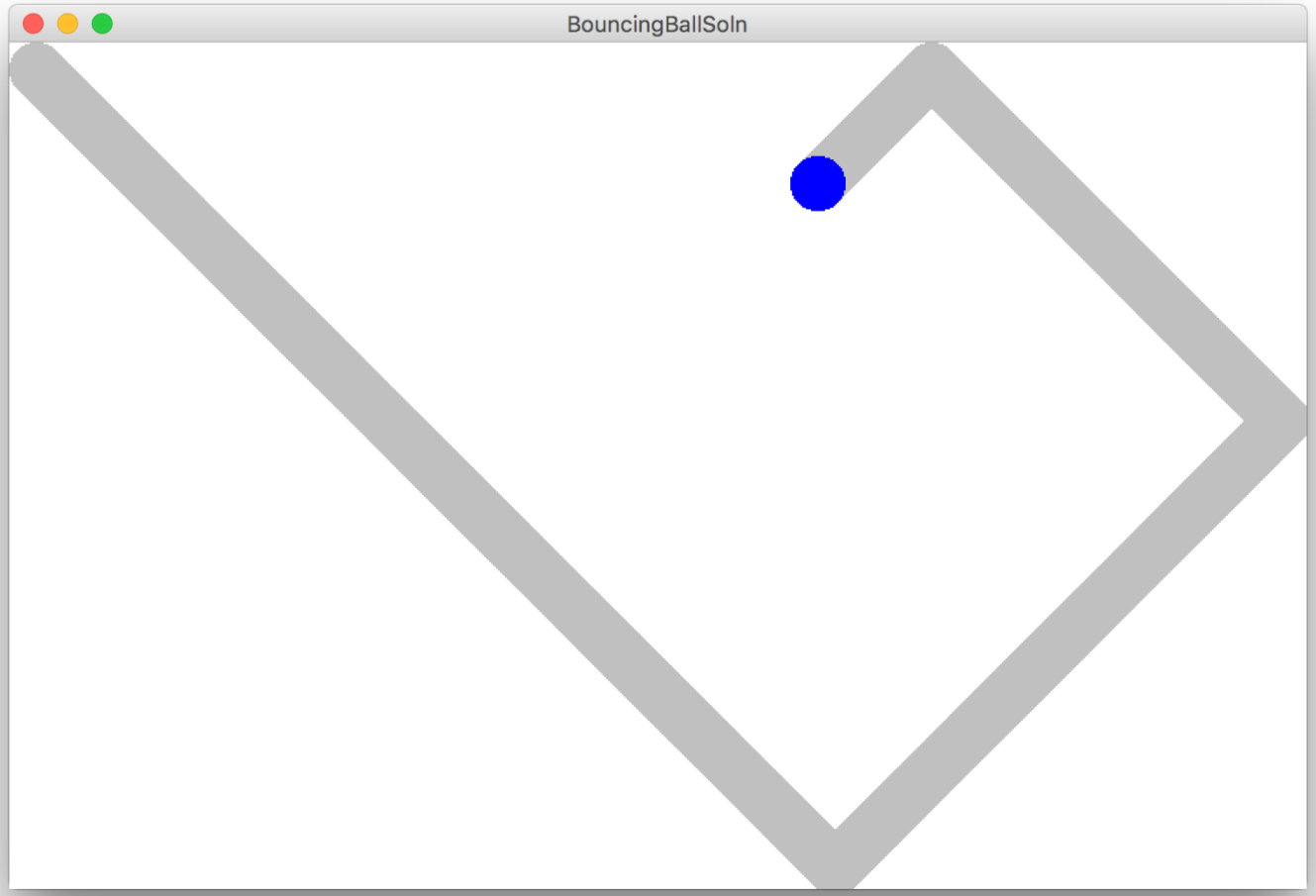


- **Turing Award** is like the nobel prize in CS
- Professor at Stanford
- Founding employee at Pixar
- Wrote RenderMan, won 3 Academy Awards
- And just a really wonderful human.

Learning Goals

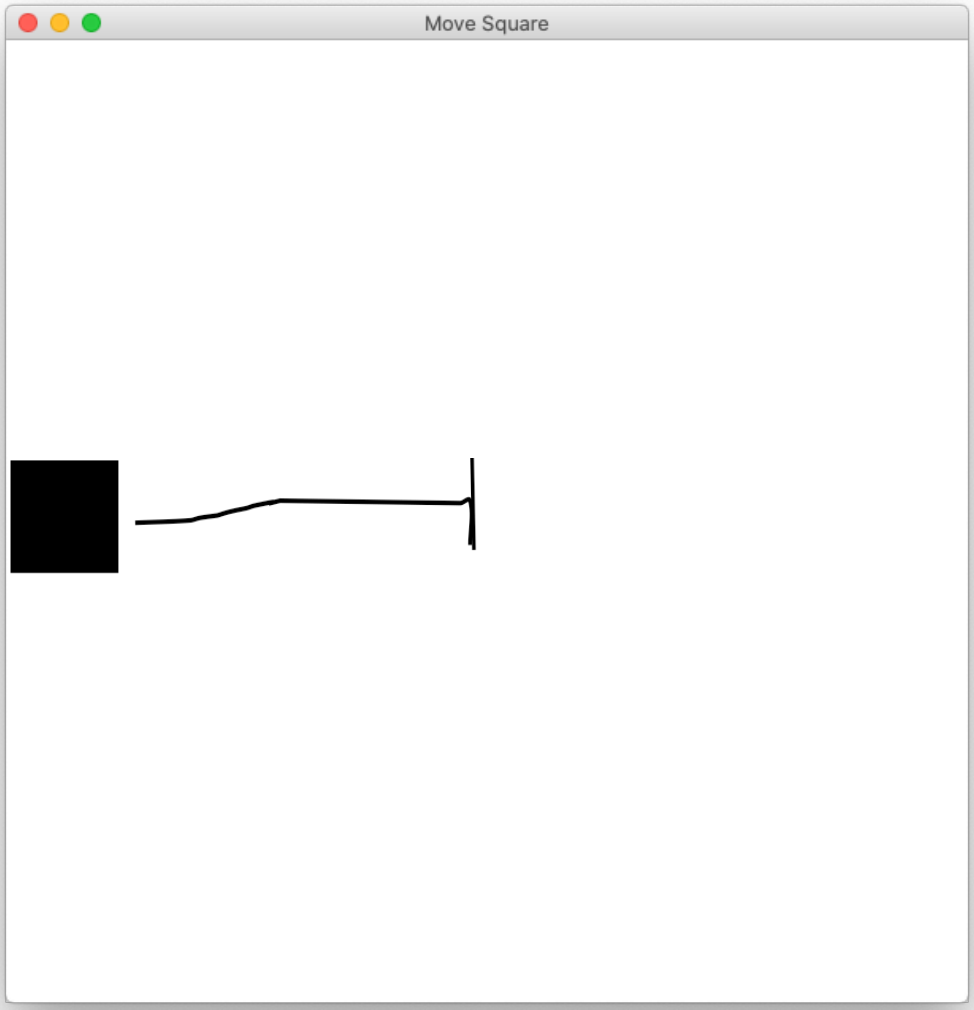
1. Feel more confident debugging
2. Write animated programs

You will be able to write Bouncing Ball



Great foundation

Move to Center

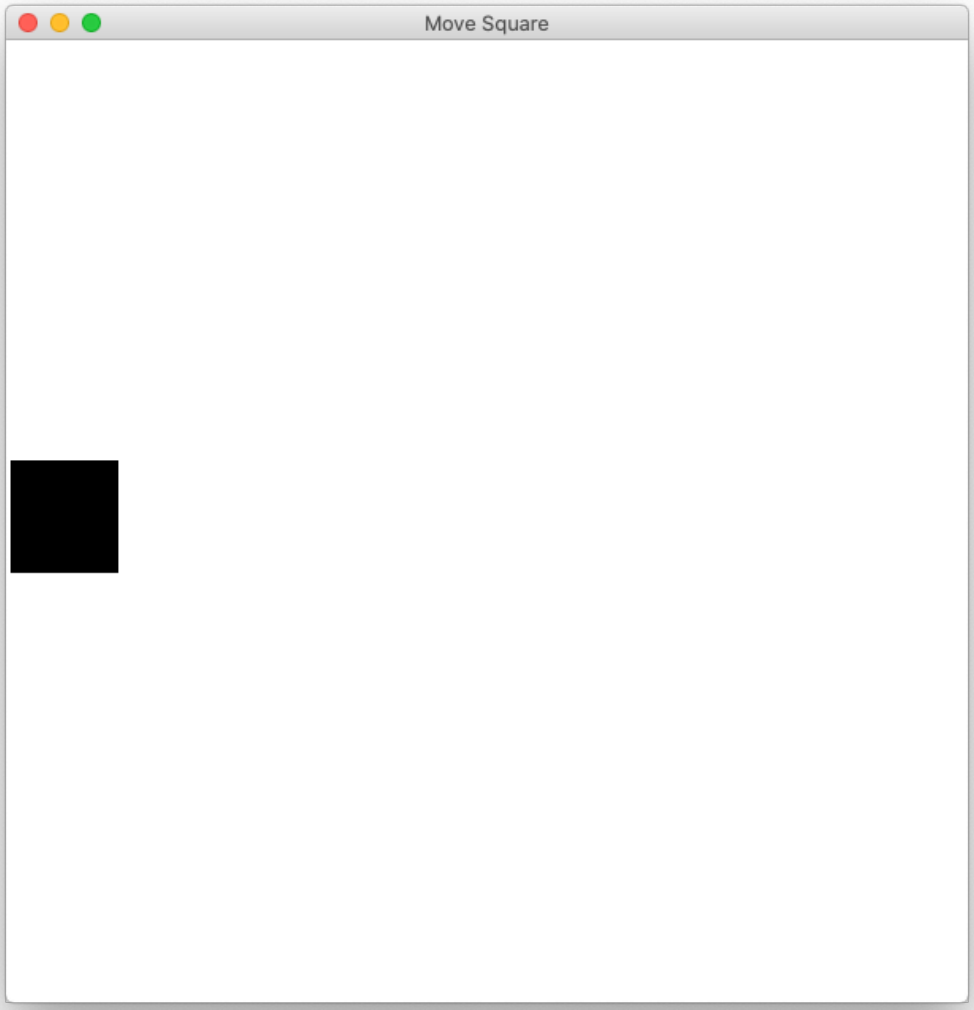


In our last episode...

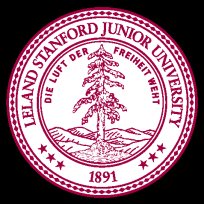
Graphics from tkinter (aka tk)

```
from graphics import Canvas
```


Add square

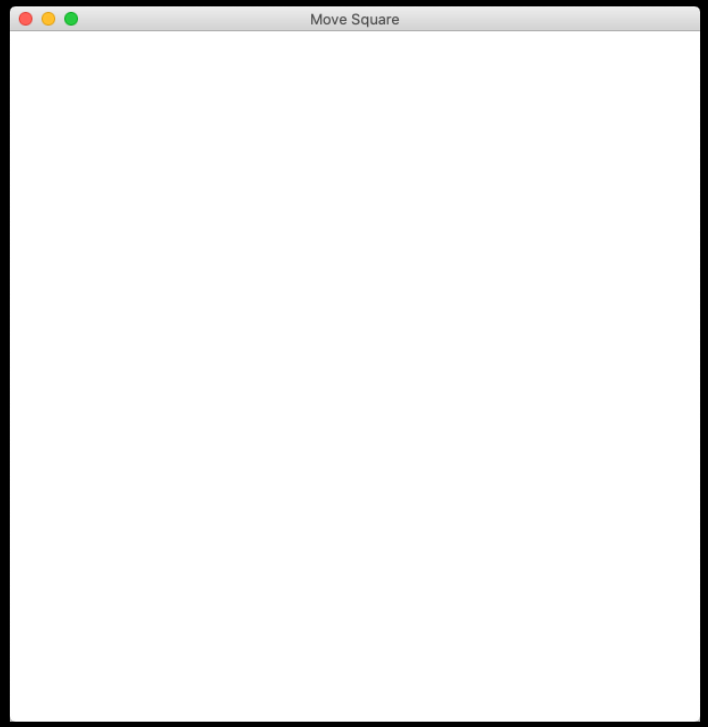


```
def main():
    canvas = Canvas(CANVAS_WIDTH, CANVAS_HEIGHT)
    start_y = CANVAS_HEIGHT / 2 - SQUARE_SIZE / 2
    end_y = start_y + SQUARE_SIZE
    rect = canvas.create_rectangle(0, start_y, SQUARE_SIZE, end_y)
    canvas.set_color('black')
    canvas.mainloop()
```

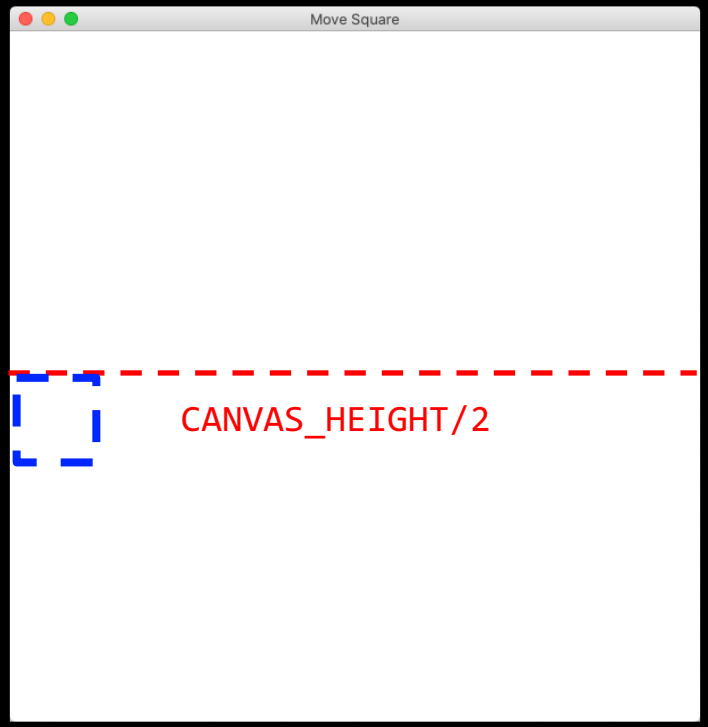



```
def main():
    canvas = Canvas(CANVAS_WIDTH, CANVAS_HEIGHT)
    start_y = CANVAS_HEIGHT / 2 - SQUARE_SIZE / 2
    end_y = start_y + SQUARE_SIZE
    rect = canvas.create_rectangle(0, start_y, SQUARE_SIZE, end_y)
    canvas.set_color('black')
    canvas.mainloop()
```

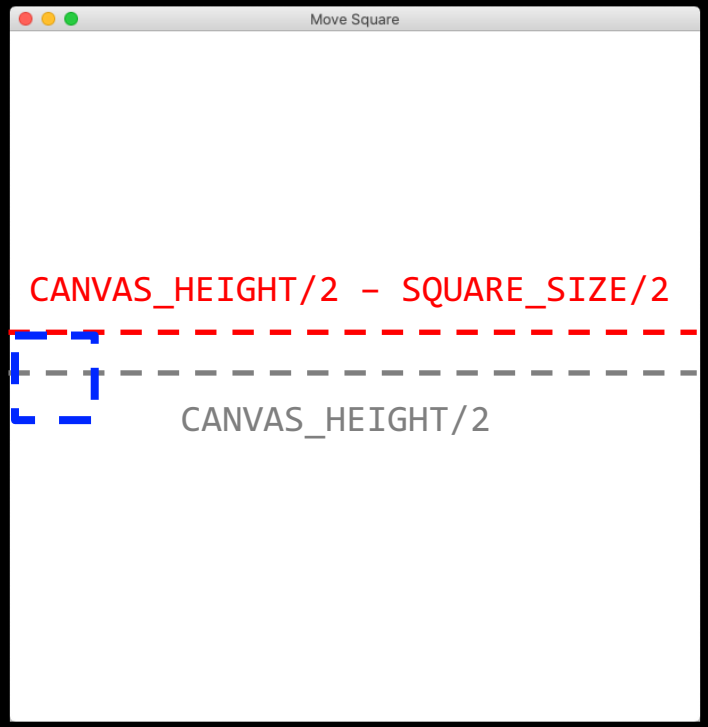
```
def main():
    canvas = Canvas(CANVAS_WIDTH, CANVAS_HEIGHT)
    start_y = CANVAS_HEIGHT / 2 - SQUARE_SIZE / 2
    end_y = start_y + SQUARE_SIZE
    rect = canvas.create_rectangle(0, start_y, SQUARE_SIZE, end_y)
    canvas.set_color('black')
    canvas.mainloop()
```



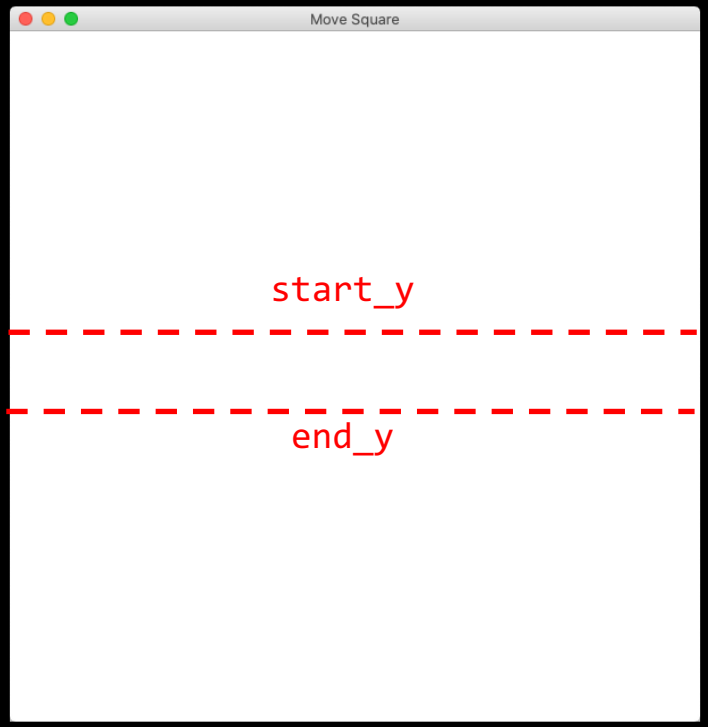
```
def main():
    canvas = Canvas(CANVAS_WIDTH, CANVAS_HEIGHT)
    start_y = CANVAS_HEIGHT / 2 - SQUARE_SIZE / 2
    end_y = start_y + SQUARE_SIZE
    rect = canvas.create_rectangle(0, start_y, SQUARE_SIZE, end_y)
    canvas.set_color('black')
    canvas.mainloop()
```



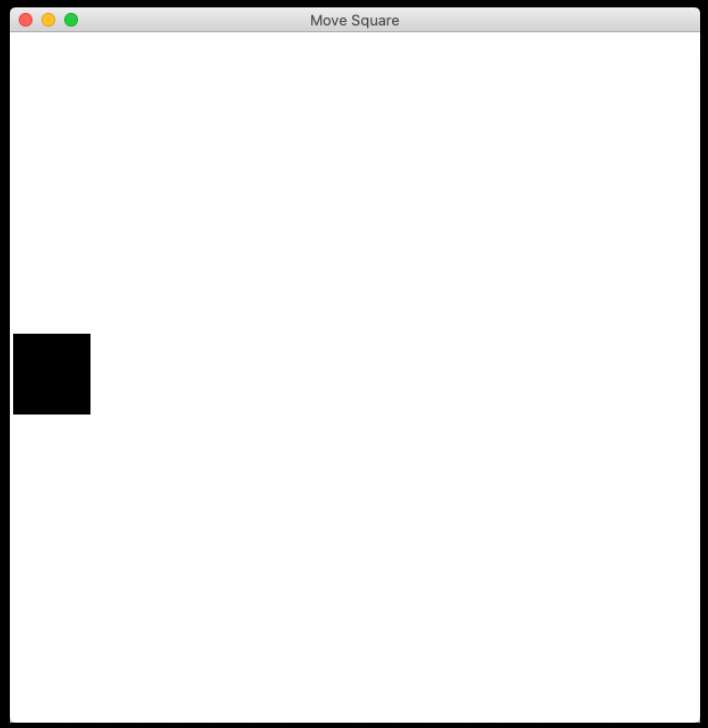
```
def main():
    canvas = Canvas(CANVAS_WIDTH, CANVAS_HEIGHT)
    start_y = CANVAS_HEIGHT / 2 - SQUARE_SIZE / 2
    end_y = start_y + SQUARE_SIZE
    rect = canvas.create_rectangle(0, start_y, SQUARE_SIZE, end_y)
    canvas.set_color('black')
    canvas.mainloop()
```



```
def main():
    canvas = Canvas(CANVAS_WIDTH, CANVAS_HEIGHT)
    start_y = CANVAS_HEIGHT / 2 - SQUARE_SIZE / 2
    end_y = start_y + SQUARE_SIZE
    rect = canvas.create_rectangle(0, start_y, SQUARE_SIZE, end_y)
    canvas.set_color('black')
    canvas.mainloop()
```

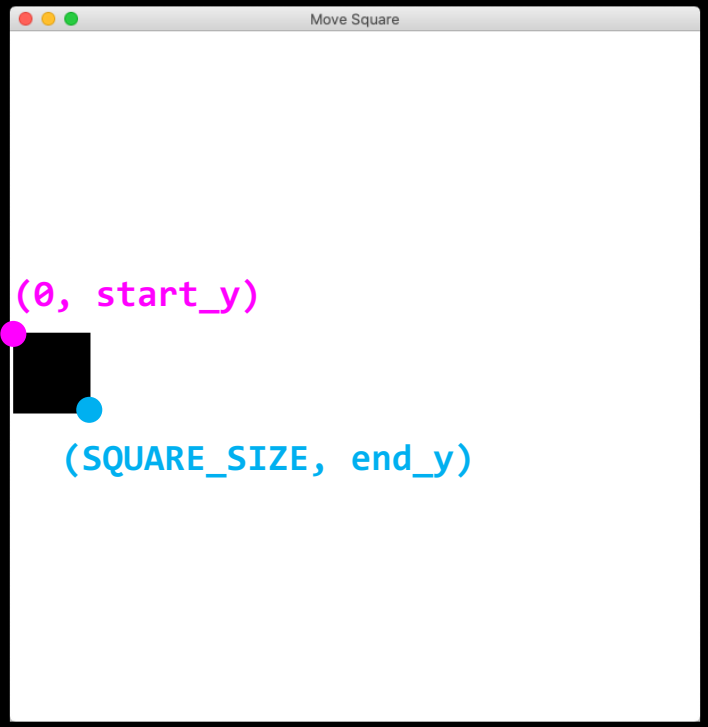


```
def main():
    canvas = Canvas(CANVAS_WIDTH, CANVAS_HEIGHT)
    start_y = CANVAS_HEIGHT / 2 - SQUARE_SIZE / 2
    end_y = start_y + SQUARE_SIZE
    rect = canvas.create_rectangle(0, start_y, SQUARE_SIZE, end_y)
    canvas.set_color('black')
    canvas.mainloop()
```

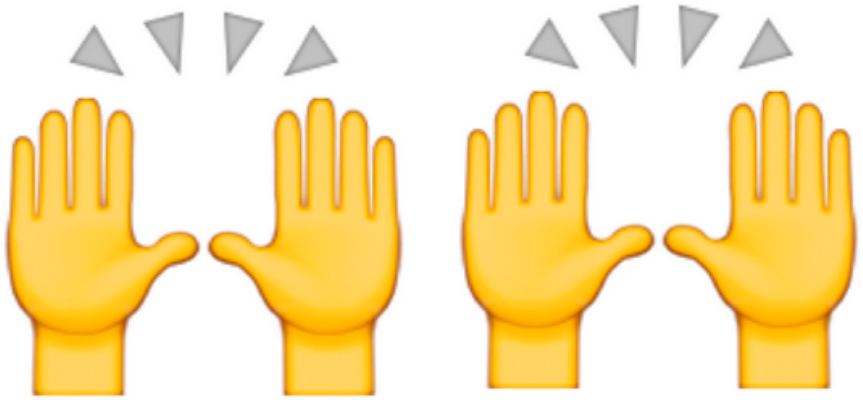


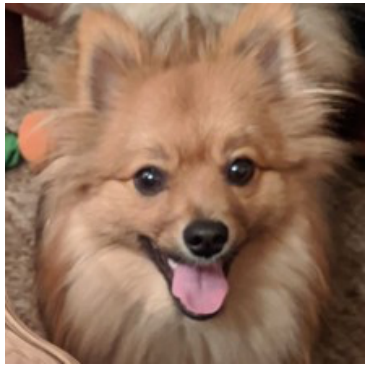
Some “heavy duty” variables allow you to call functions on them

```
def main():
    canvas = Canvas(CANVAS_WIDTH, CANVAS_HEIGHT)
    start_y = CANVAS_HEIGHT / 2 - SQUARE_SIZE / 2
    end_y = start_y + SQUARE_SIZE
    rect = canvas.create_rectangle(0, start_y, SQUARE_SIZE, end_y)
    canvas.set_color('black')
    canvas.mainloop()
```



You're now all graphics programmers!

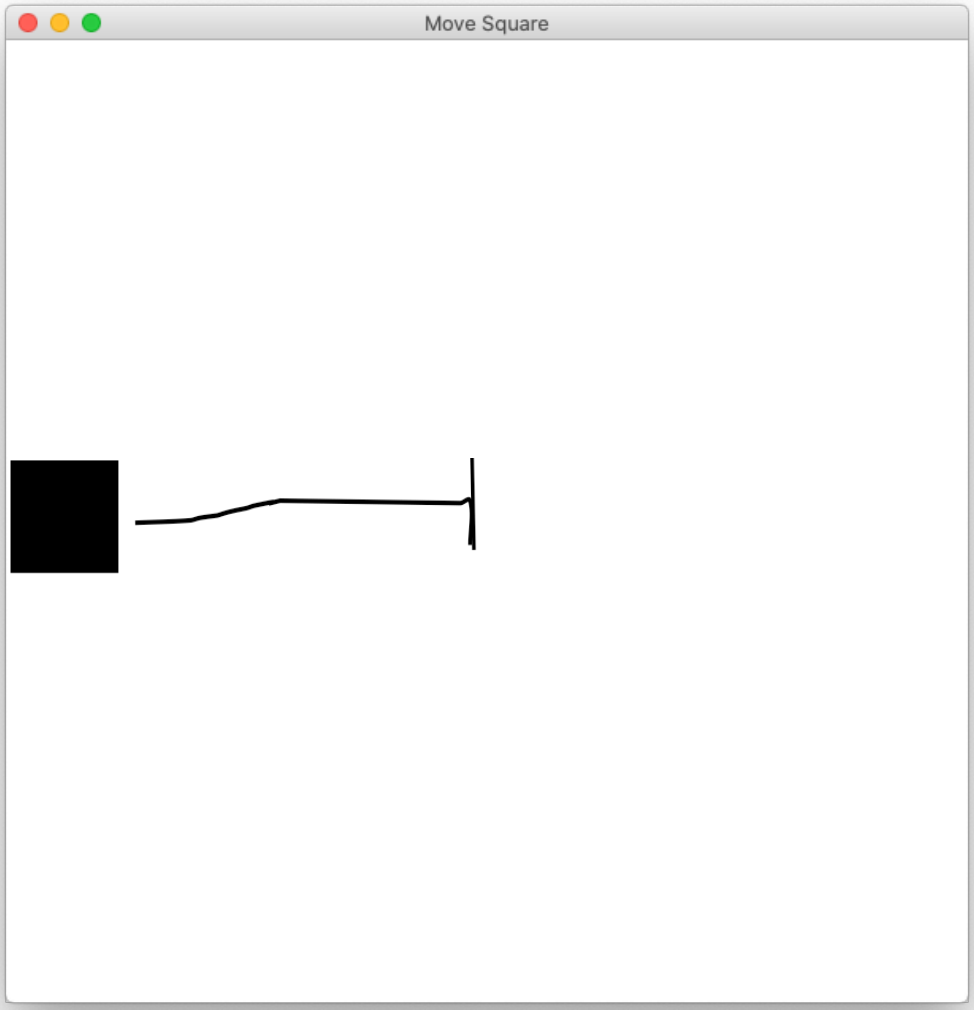




End review...

How do movies or games animate?

Move to Center



* That's not quite toy story, but it is a start…

Animation Loop

```
def main():
    # setup

    while True:
        # update world

        # pause
        time.sleep(DELAY)
```


Animation Loop

```
def main():
    # setup

    while True:
        # update world

        # pause
        time.sleep(DELAY)
```

Make all the variables you need.

Animation Loop

```
def main():
    # setup

    while True:
        # update world

        # pause
        time.sleep(DELAY)
```

The animation loop is a repetition of heartbeats

Animation Loop

```
def main():
    # setup

    while True:
        # update world

        # pause
        time.sleep(DELAY)
```

Each heart-beat, update the world forward one frame

Animation Loop

```
def main():
    # setup

    while True:
        # update world

        # pause
        time.sleep(DELAY)
```

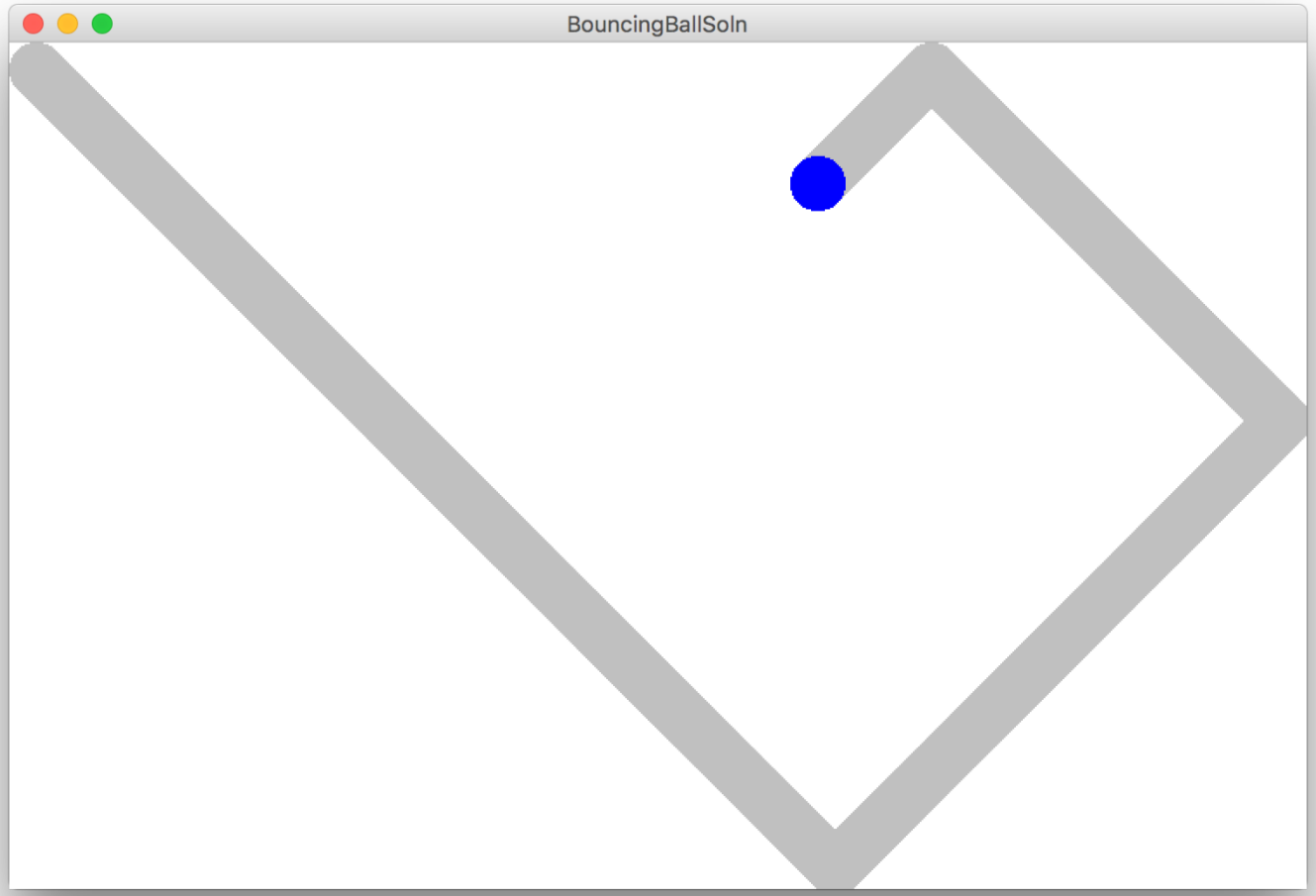
If you don't pause, humans won't be able to see it

Move To Center

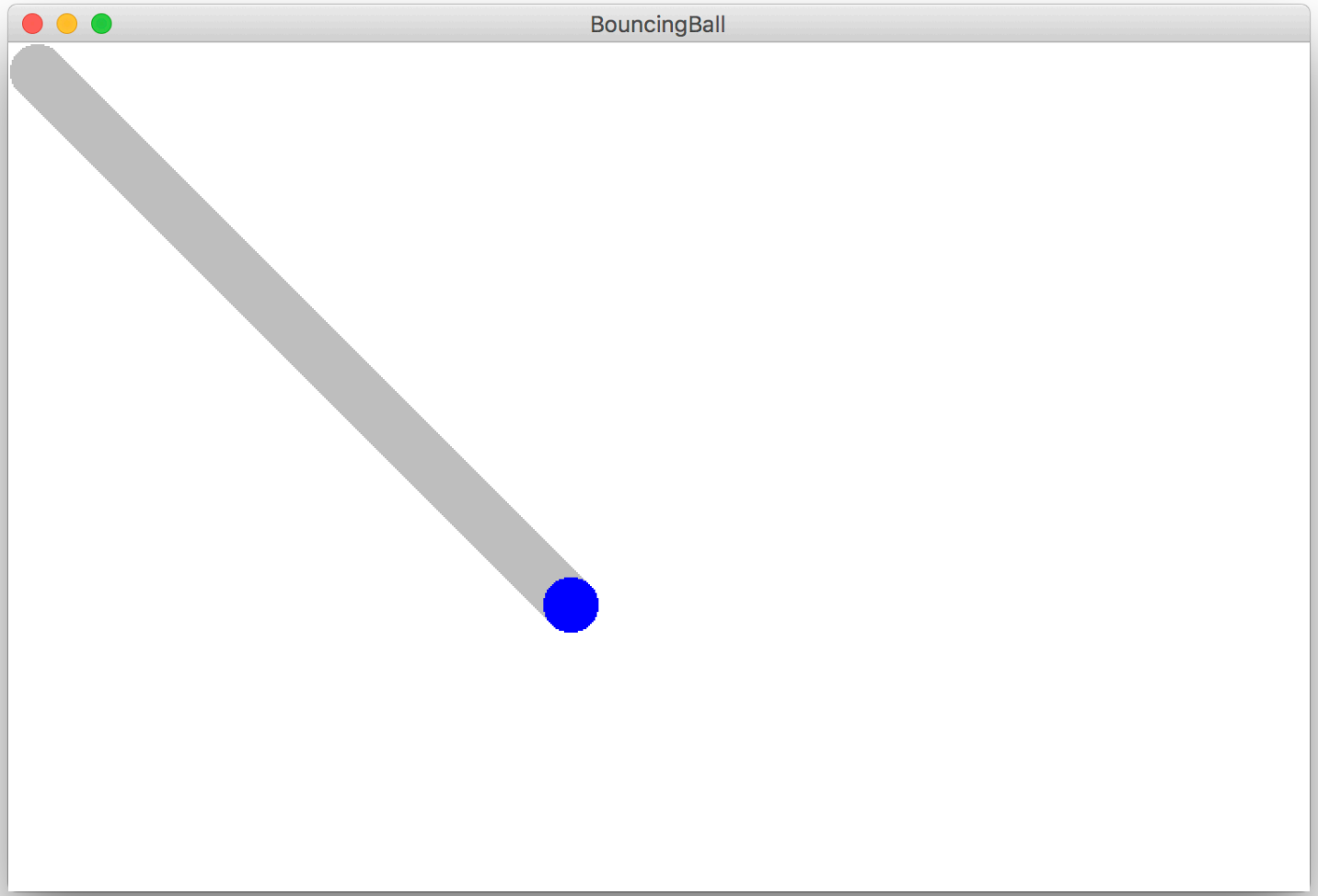
```
def main():
    # setup
    canvas = Canvas(CANVAS_WIDTH, CANVAS_HEIGHT)
    r = canvas.create_rectangle(0, 0, 100, 100)
    while not is_past_center(canvas, r):
        # update world
        canvas.move(r, 1, 0)
        canvas.update()
        # pause
        time.sleep(DELAY)
```

We are ready…

Bouncing Ball

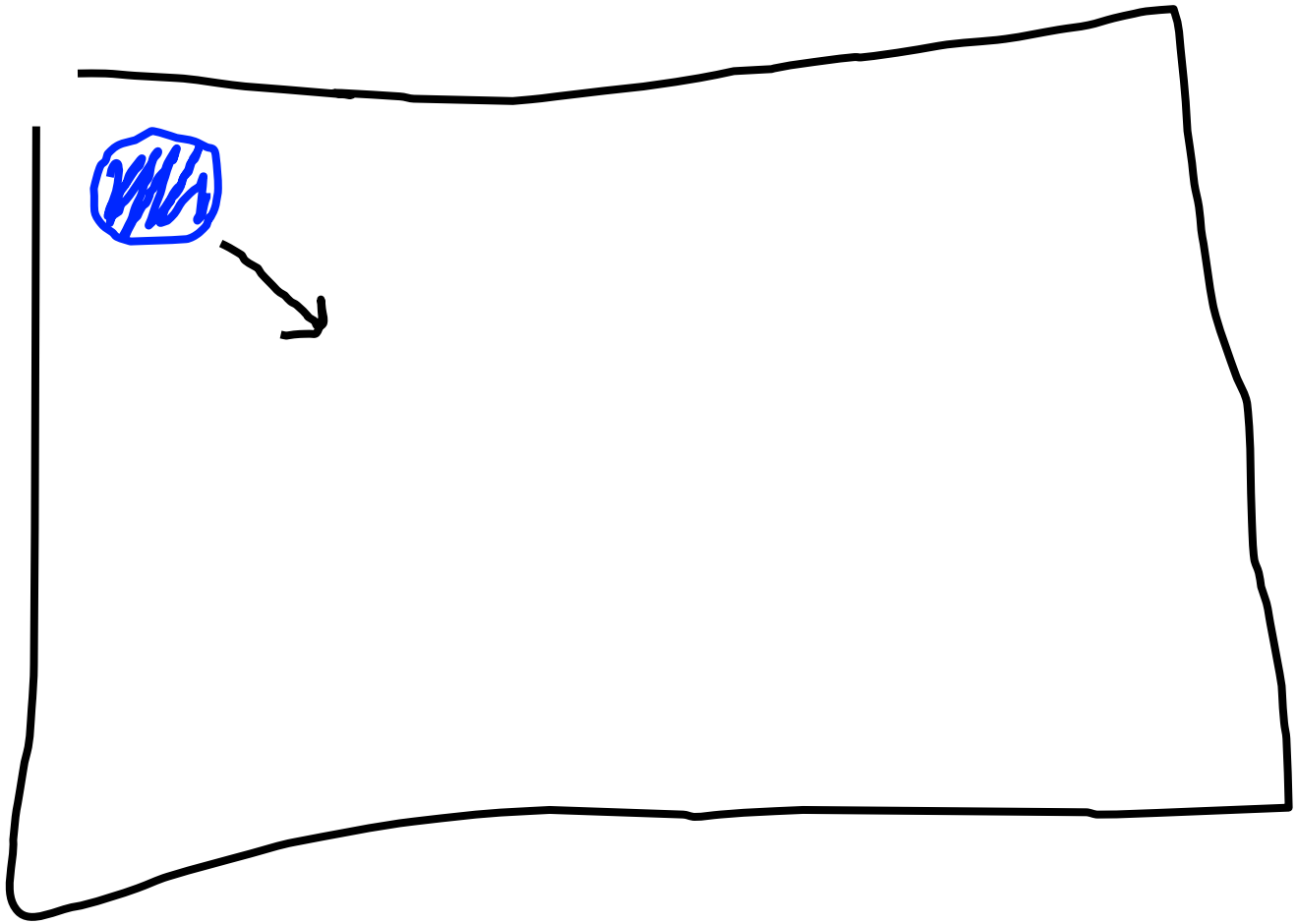


Milestone #1



Bouncing Ball

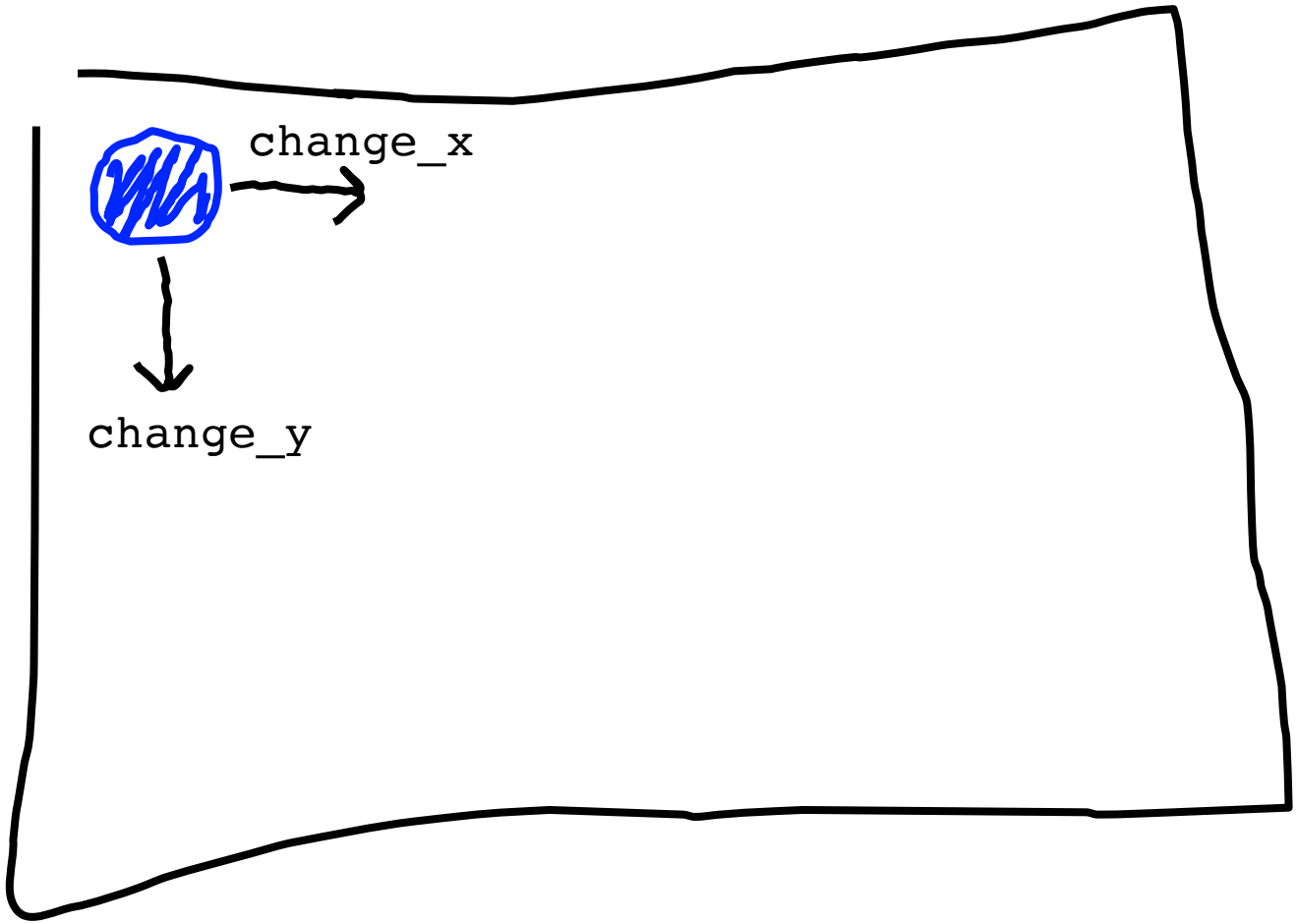
First heartbeat



Key variable: how much the ball position change each heartbeat?

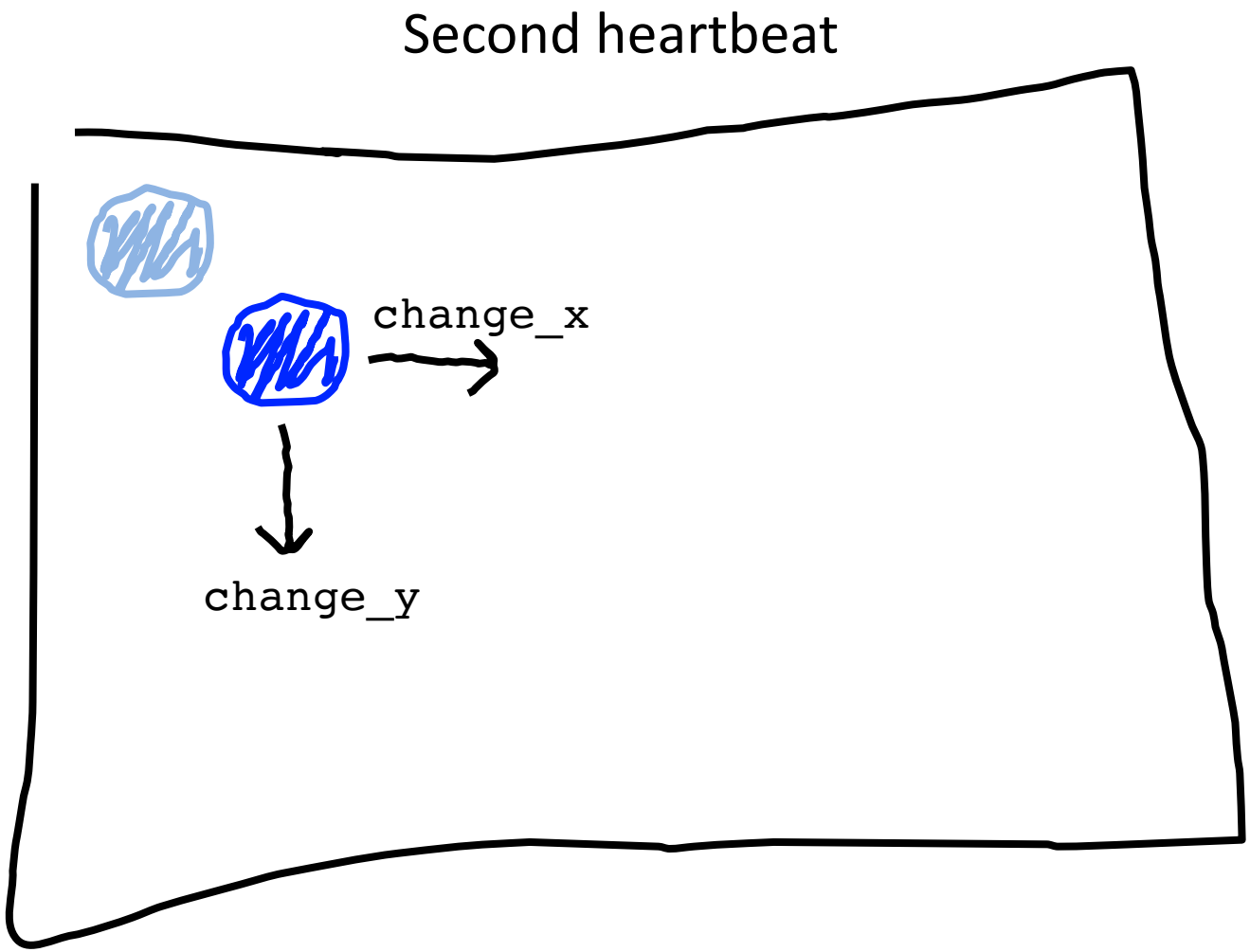
Bouncing Ball

First heartbeat

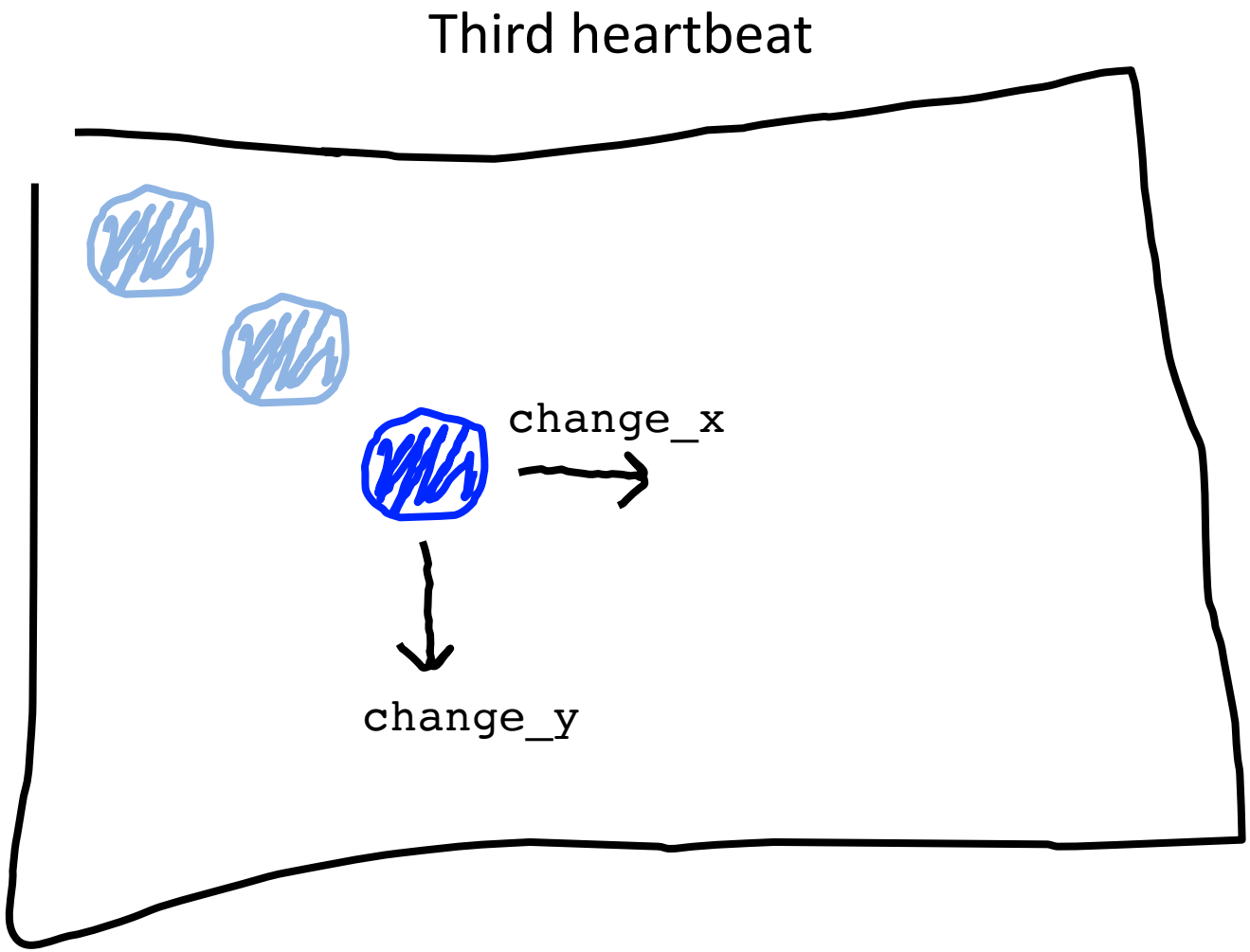


The **`move`** function takes in
a change in x and a change in y

Bouncing Ball

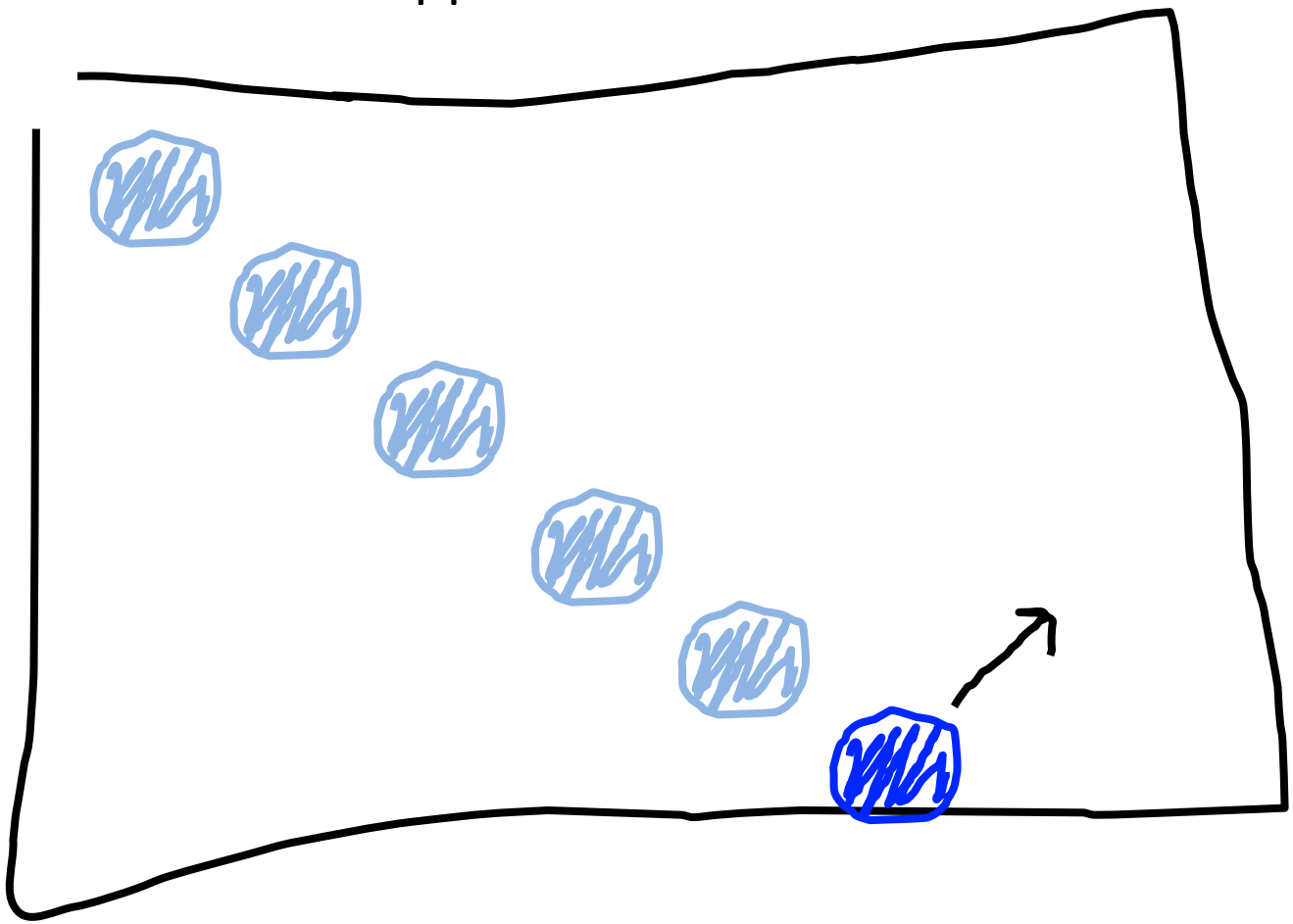


Bouncing Ball



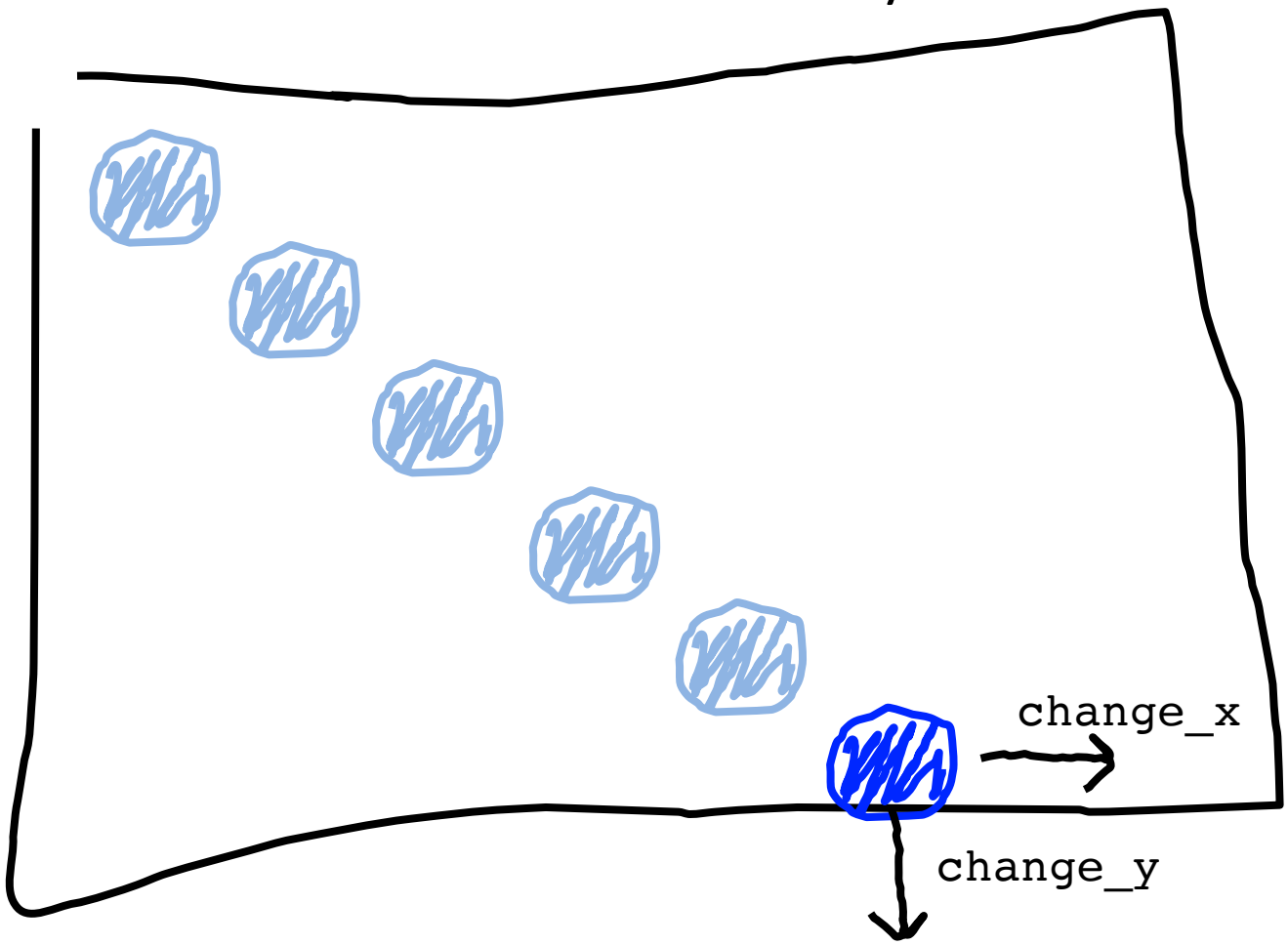
Bouncing Ball

What happens when we hit a wall?



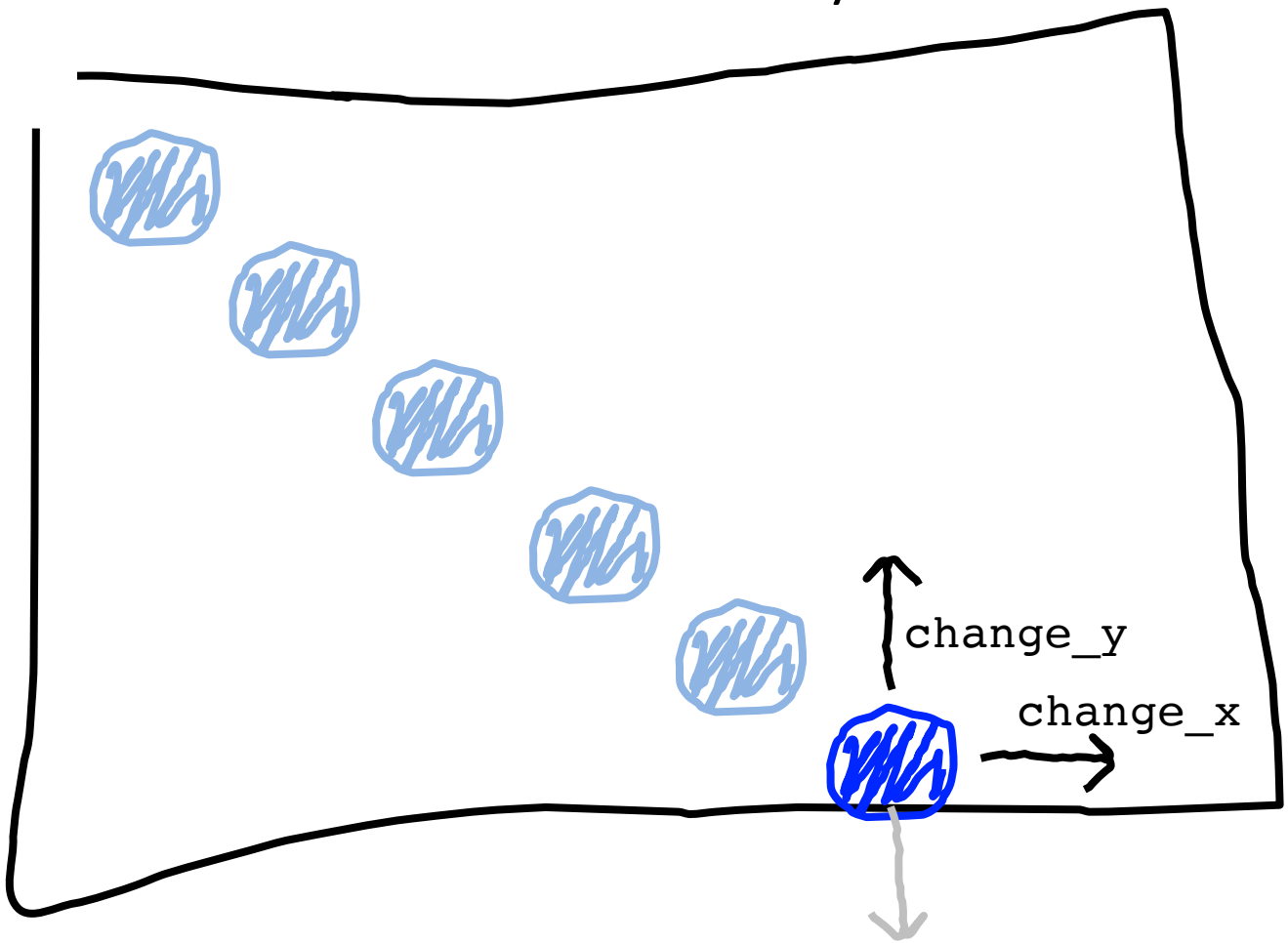
Bouncing Ball

We have this velocity



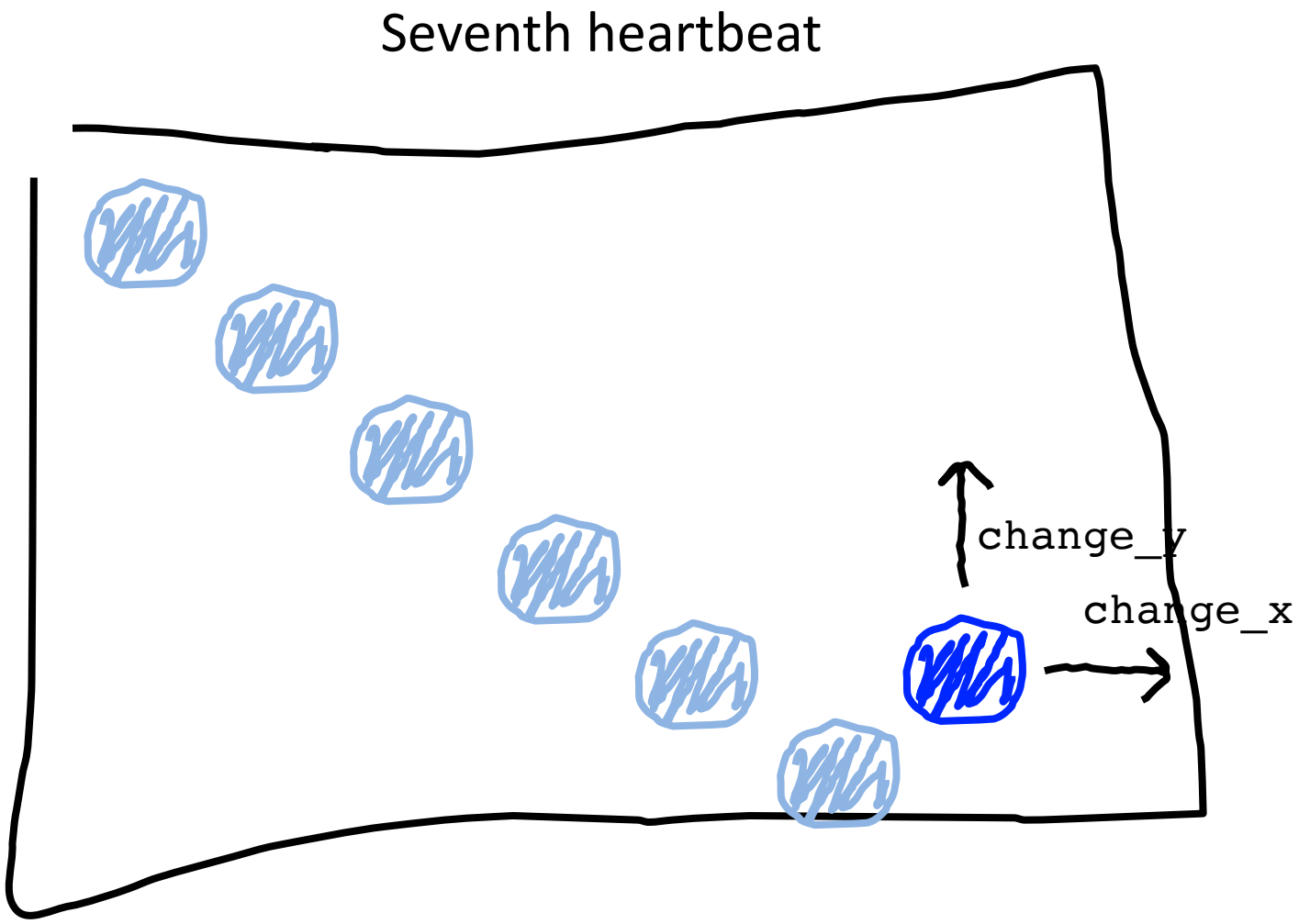
Bouncing Ball

Our new velocity

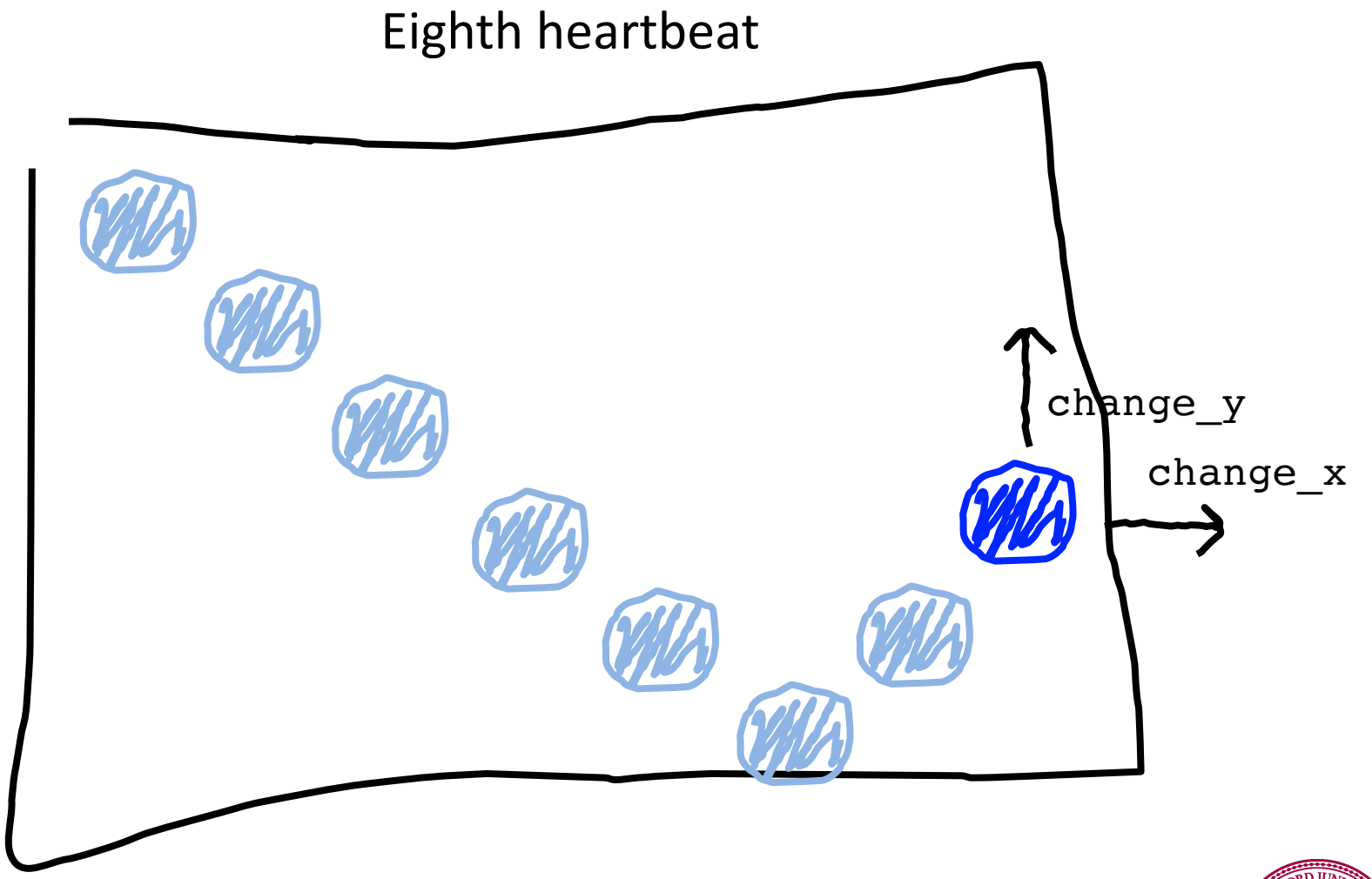


Piec

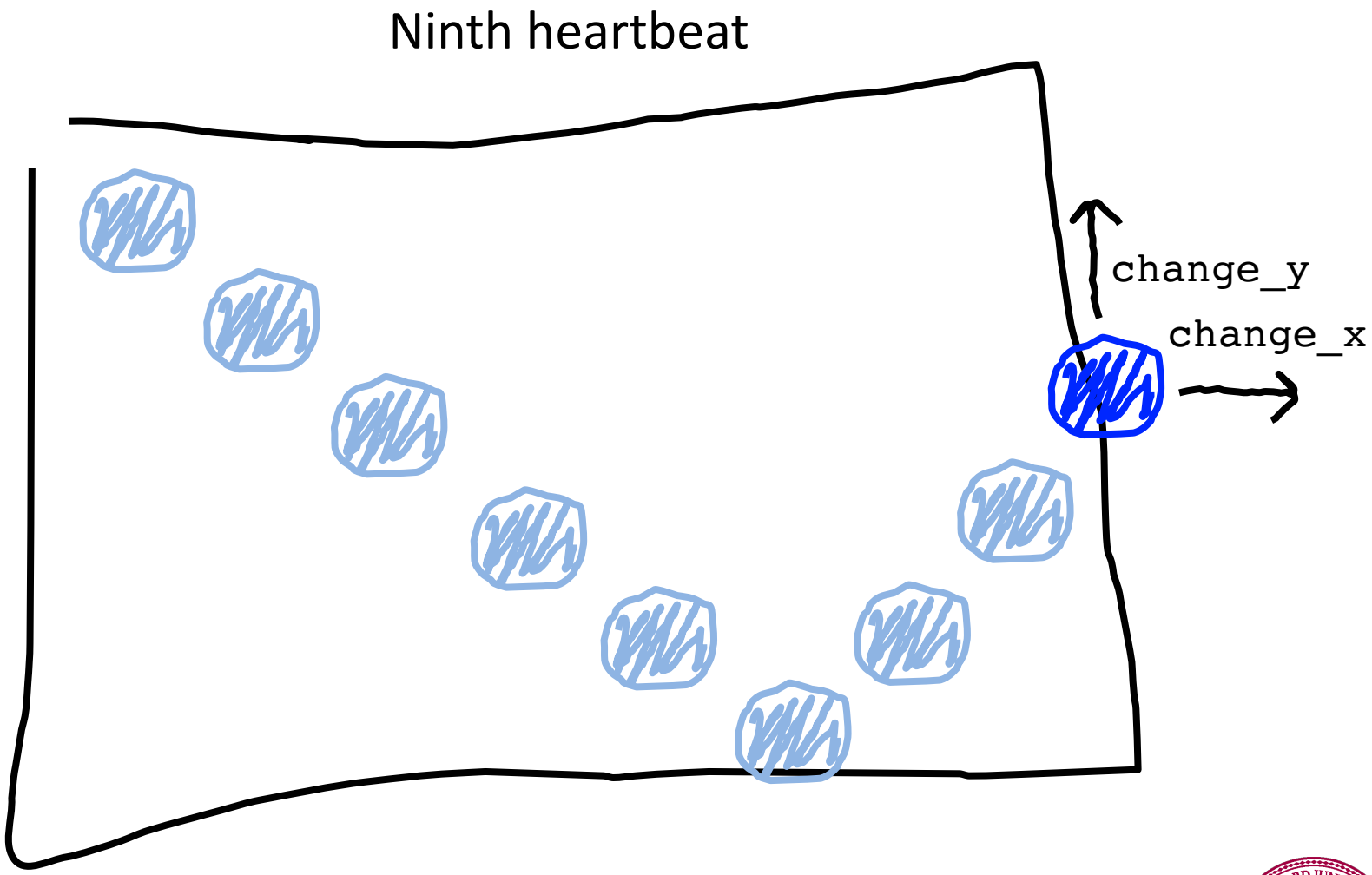
Bouncing Ball



Bouncing Ball

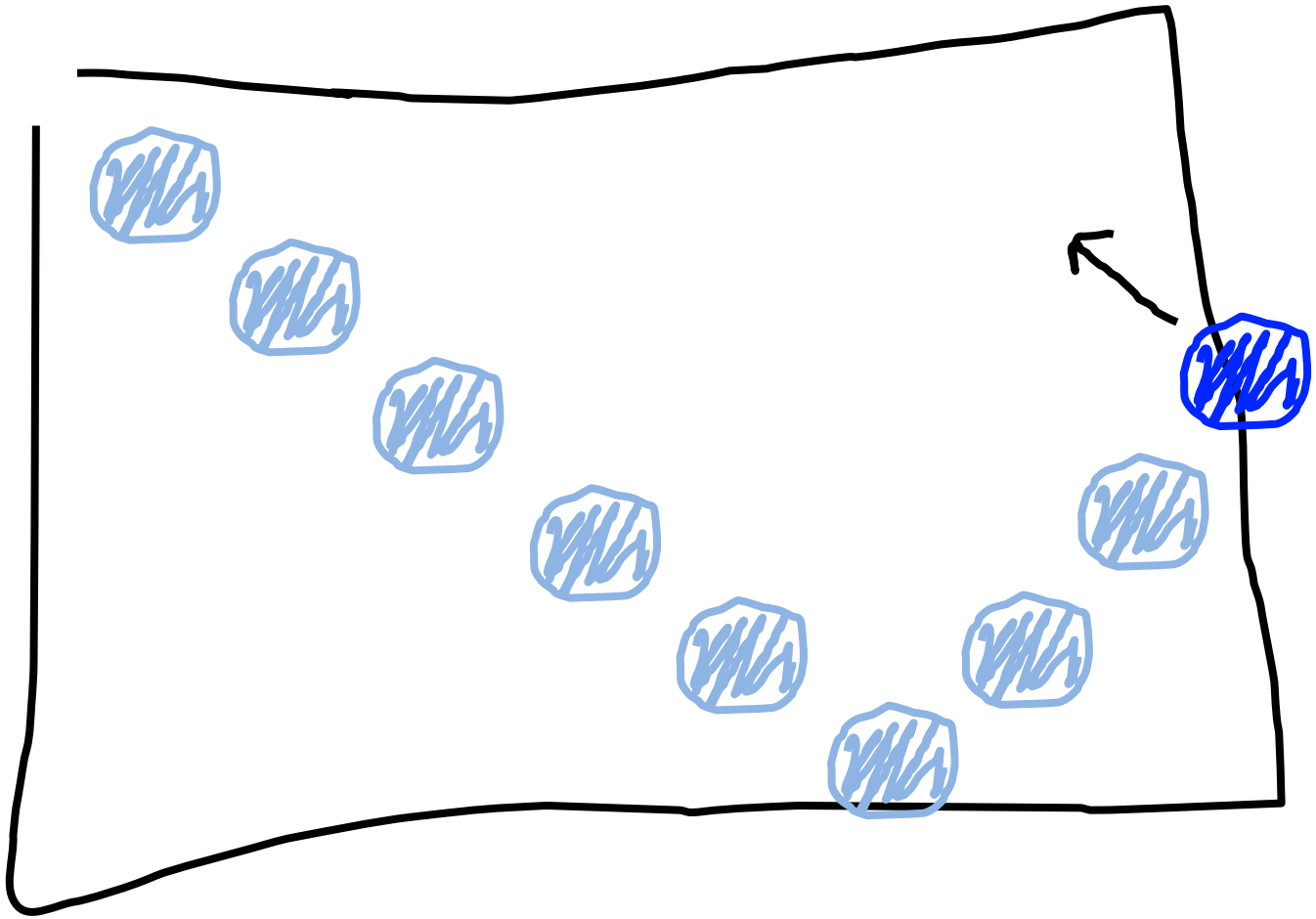


Bouncing Ball



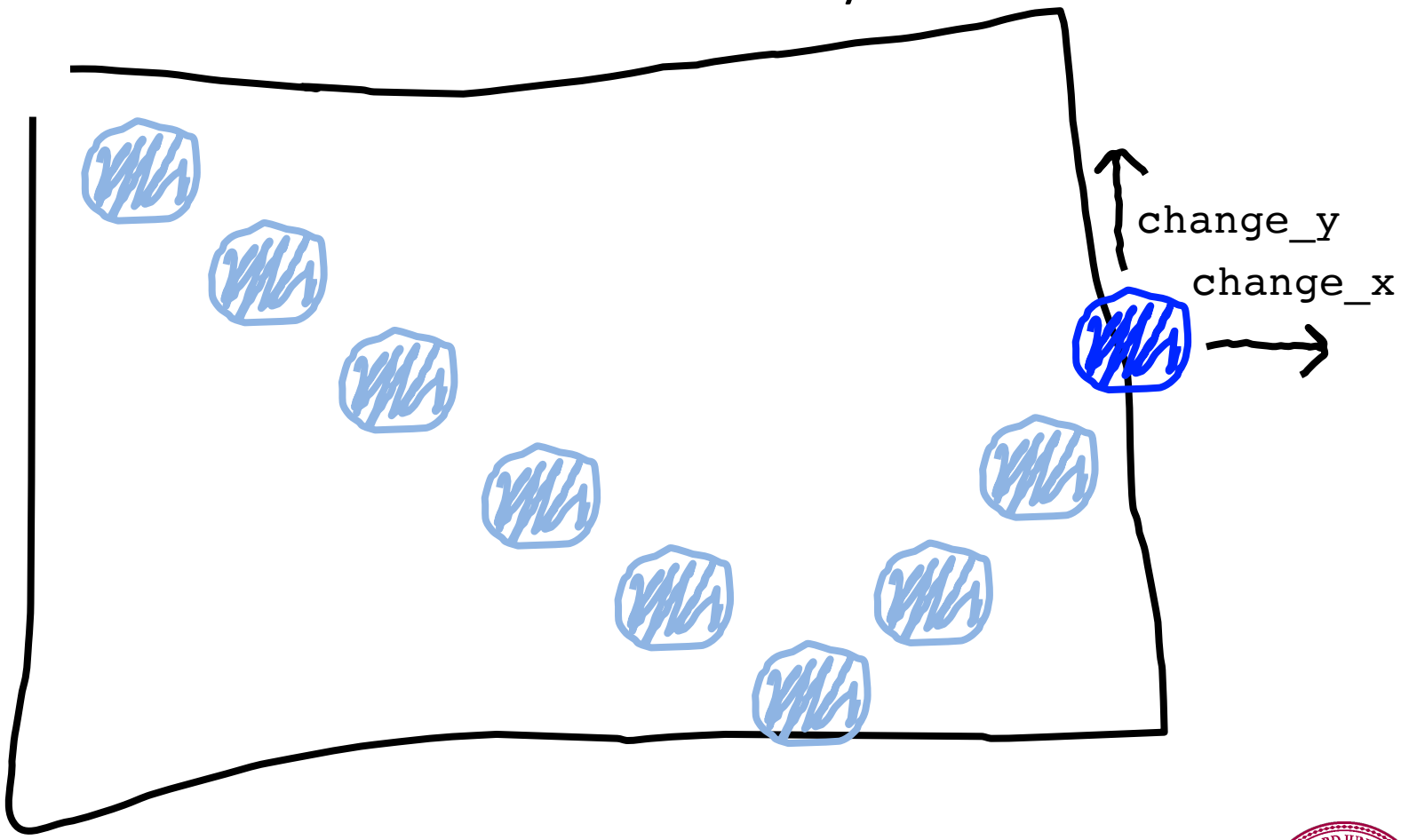
Bouncing Ball

We want this!



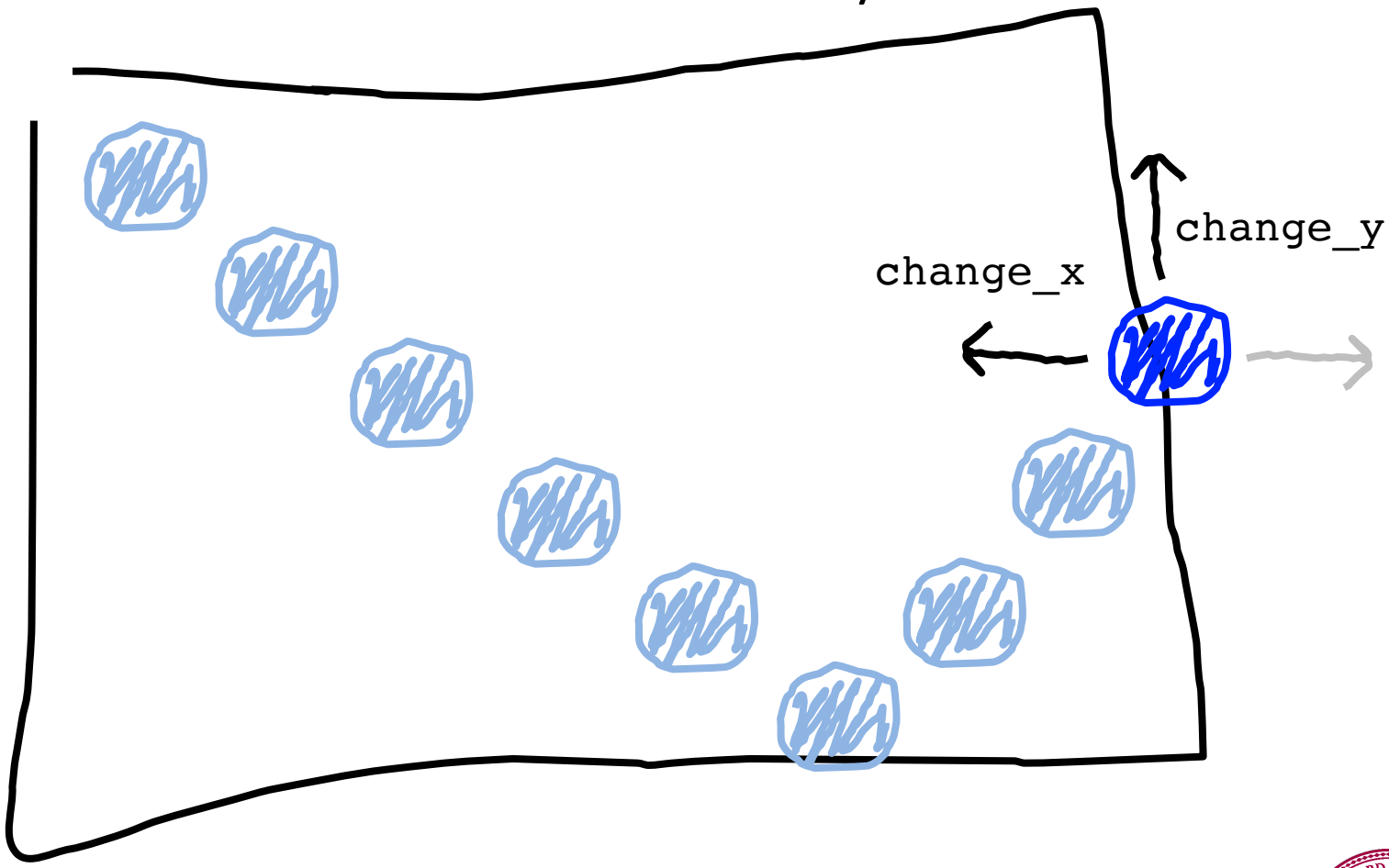
Bouncing Ball

This was our old velocity



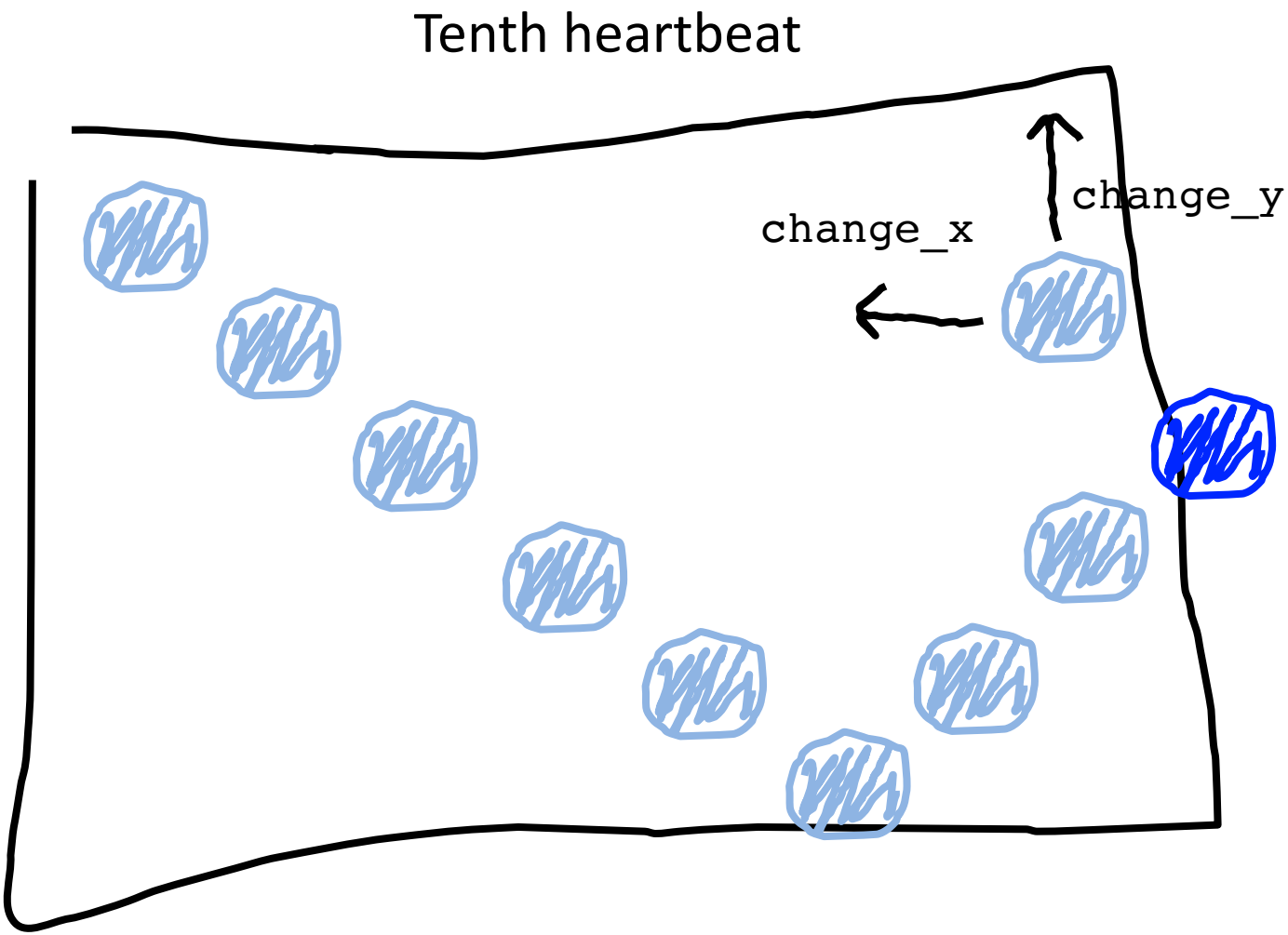
Bouncing Ball

This is our new velocity



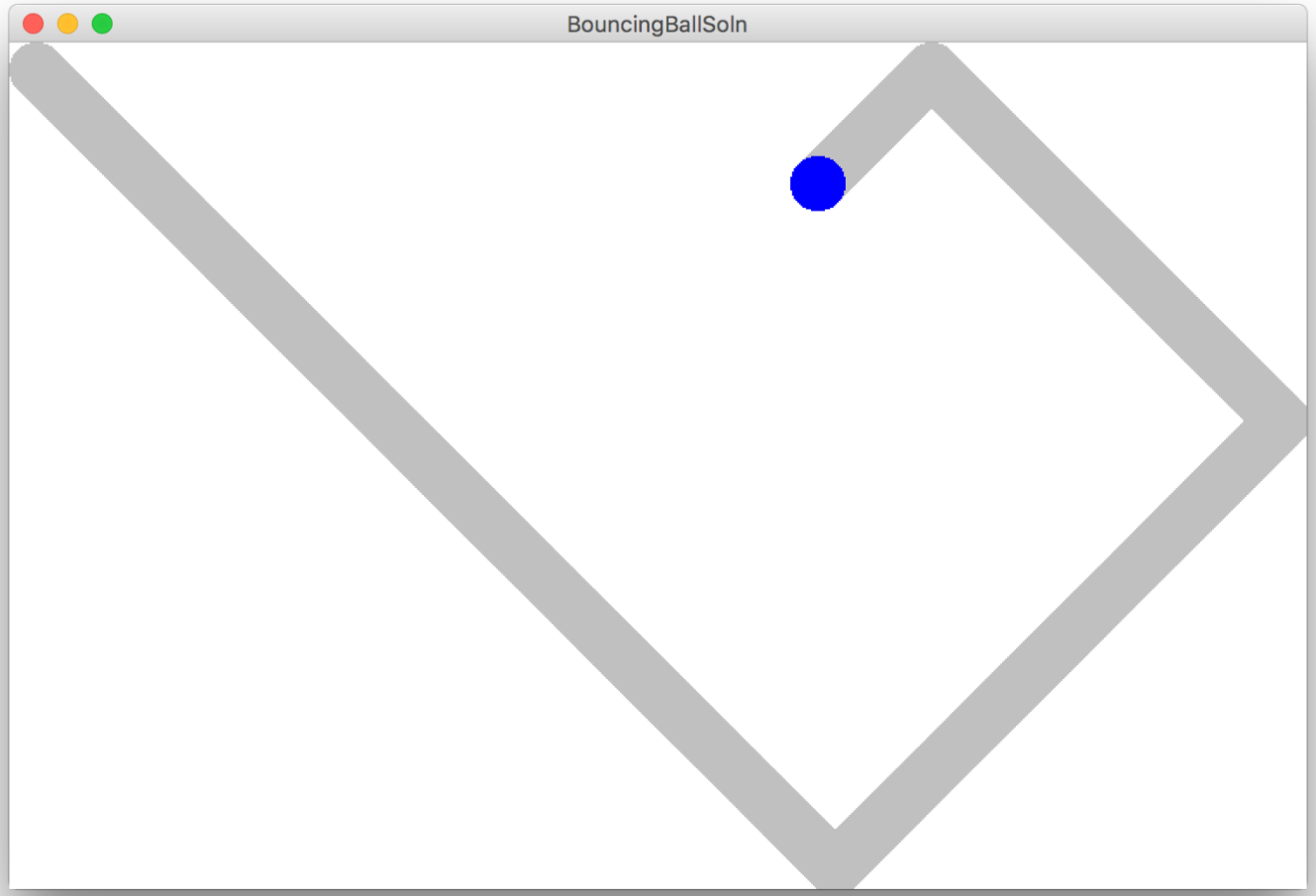
When reflecting horizontally: `change_x = -change_x`

Bouncing Ball



When reflecting horizontally: `change_x = -change_x`

Bouncing Ball



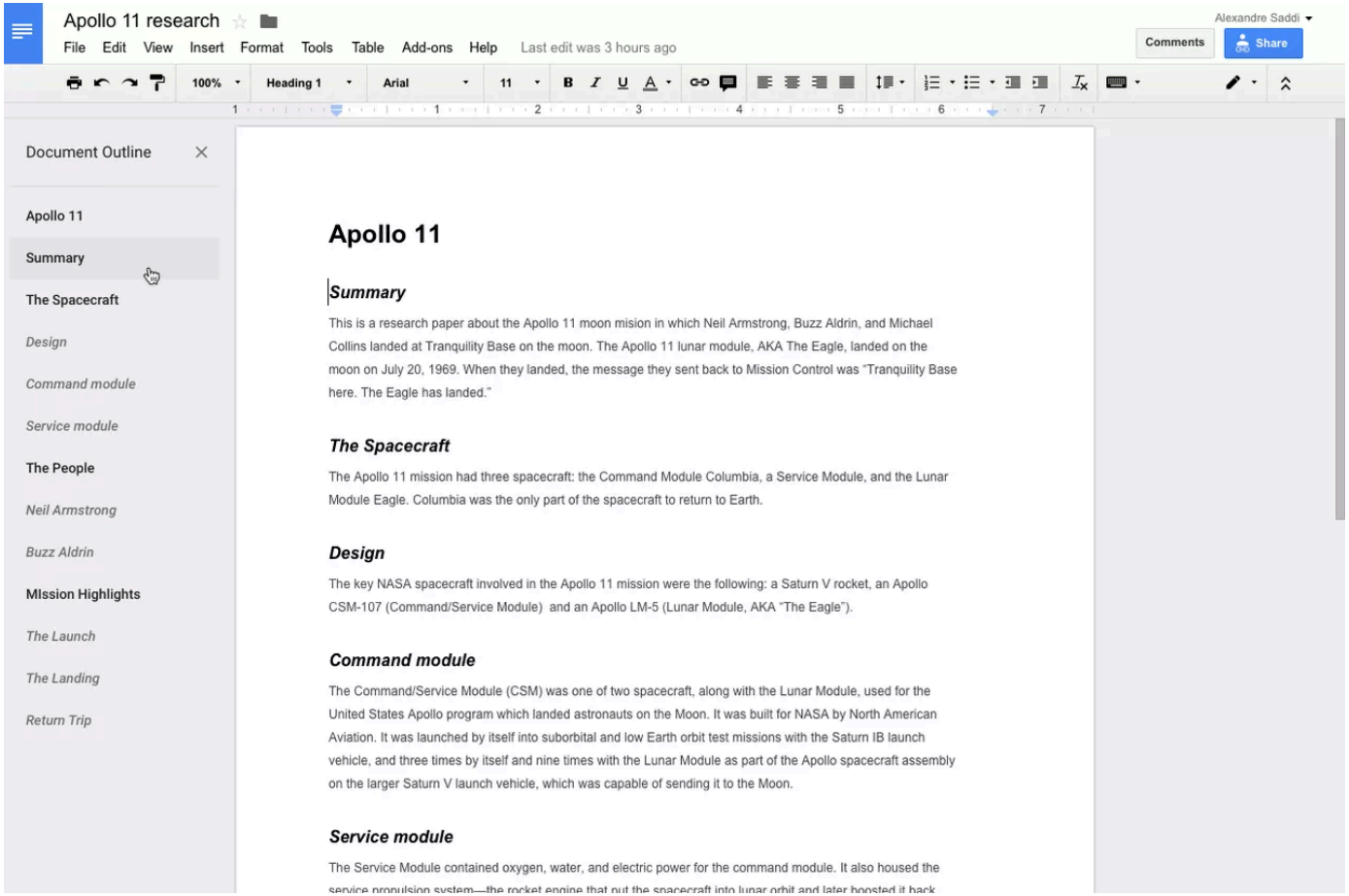
Hold up!

```
def make_ball(canvas):
```

Does this copy the canvas??!!

*Variables are stored using a reference. Which is like a **URL**. The URL gets copied when you pass the variable*

How do you share google docs?



https://docs.google.com/document/d/1eBtnEiII3KHefFS-kSAOpXqeSXpbfTTMlmOgj6I9dvk/

```
def main():
    canvas = Canvas(…)
    make_ball(canvas)

def make_ball(canvas):
    canvas.create_rectangle( … )
```

---

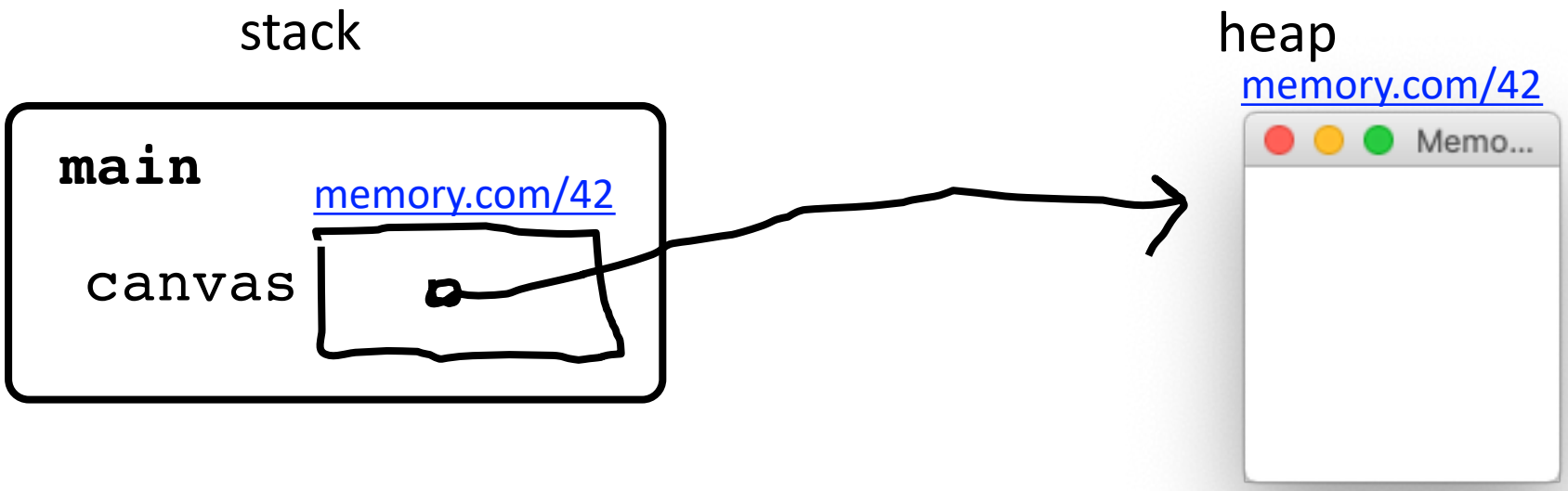
stack

heap

main

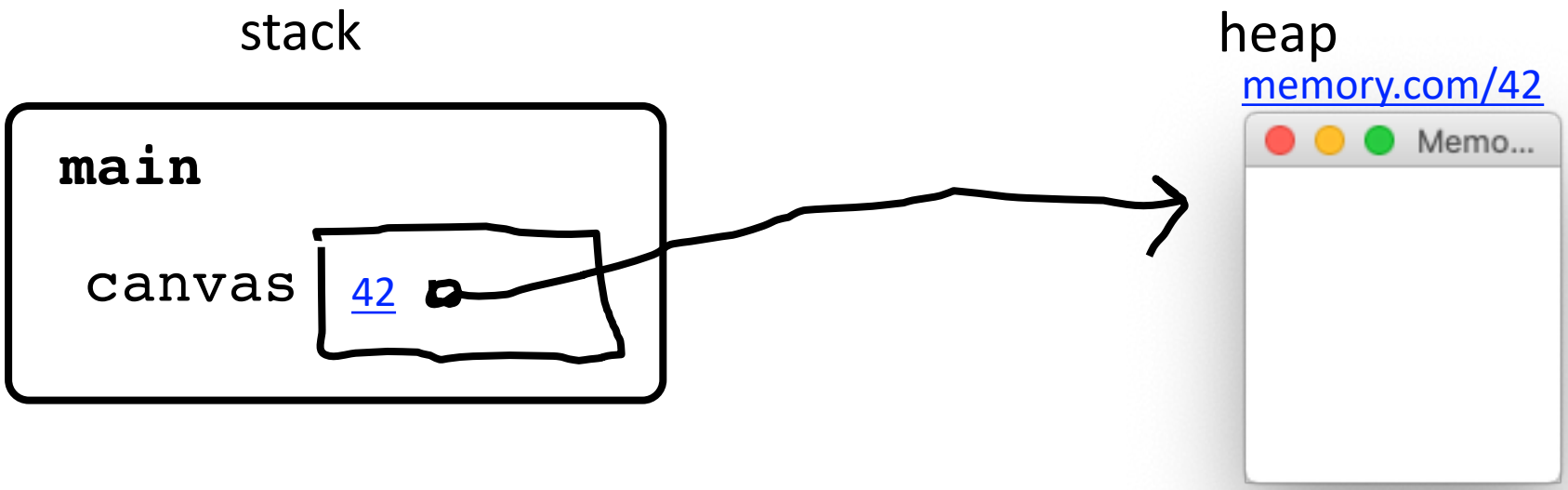
```
def main():
    canvas = Canvas(…)
    make_ball(canvas)

def make_ball(canvas):
    canvas.create_oval( … )
```



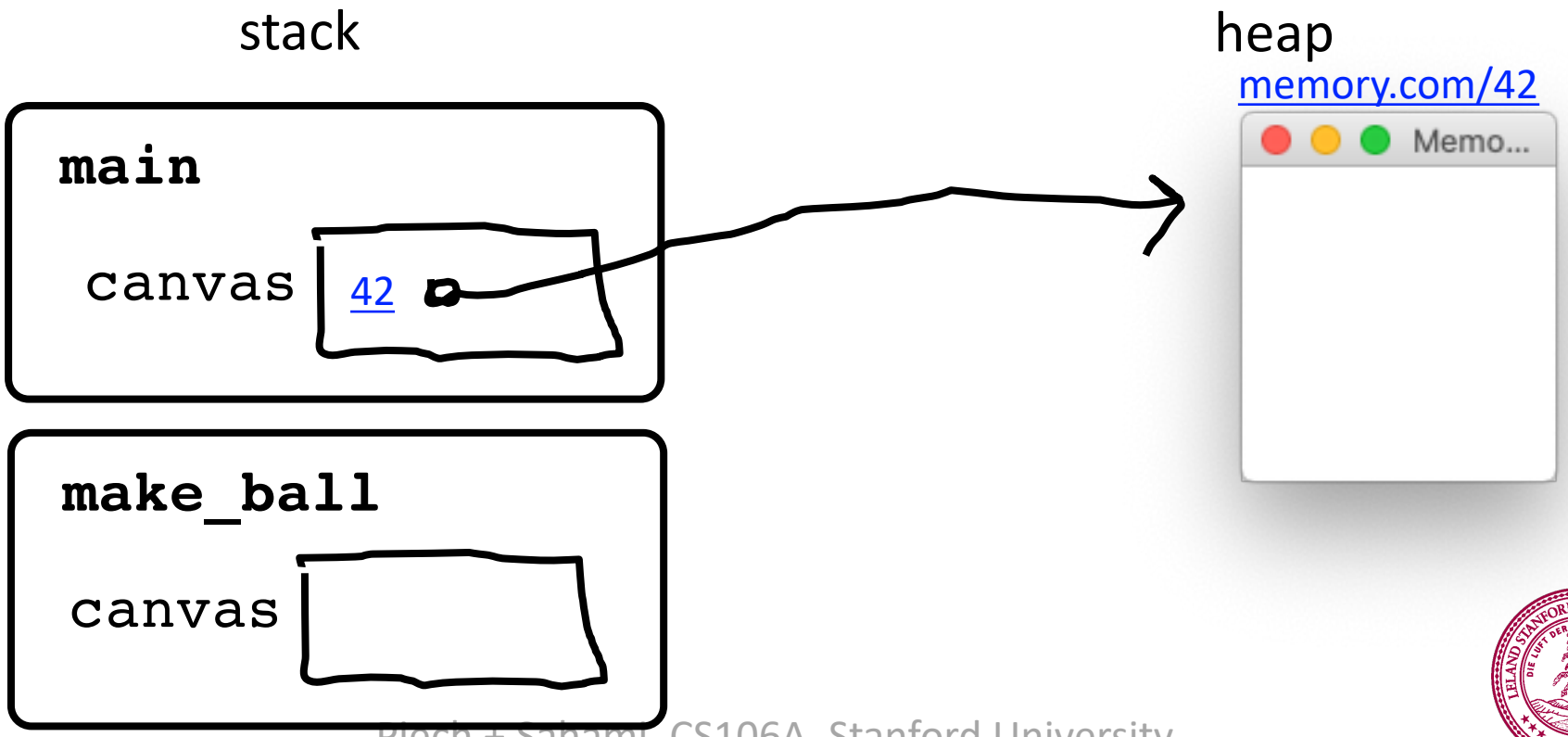
```
def main():
    canvas = Canvas(…)
    make_ball(canvas)

def make_ball(canvas):
    canvas.create_oval( … )
```



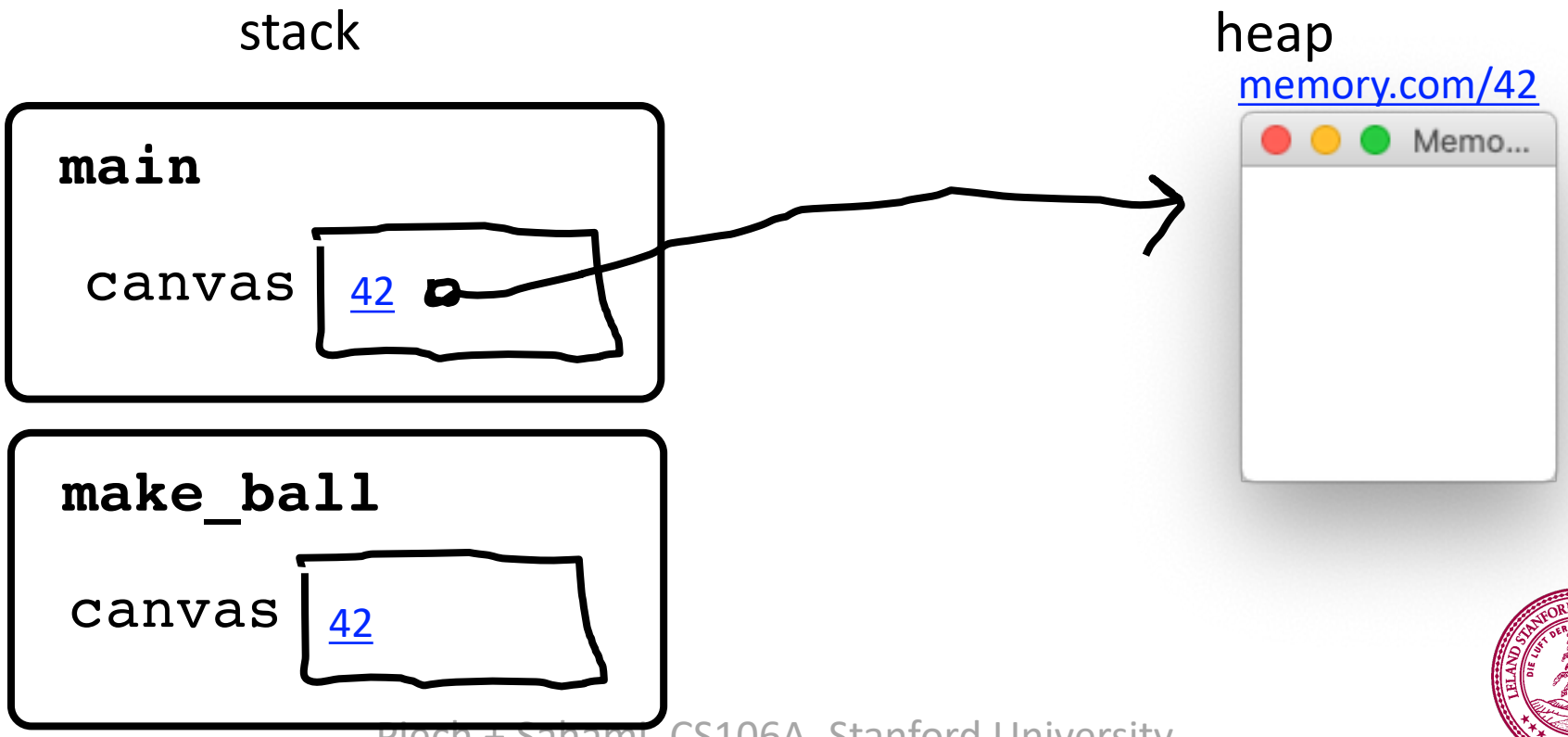
```
def main():
    canvas = Canvas(…)
    make_ball(canvas)

def make_ball(canvas):
    canvas.create_oval( … )
```



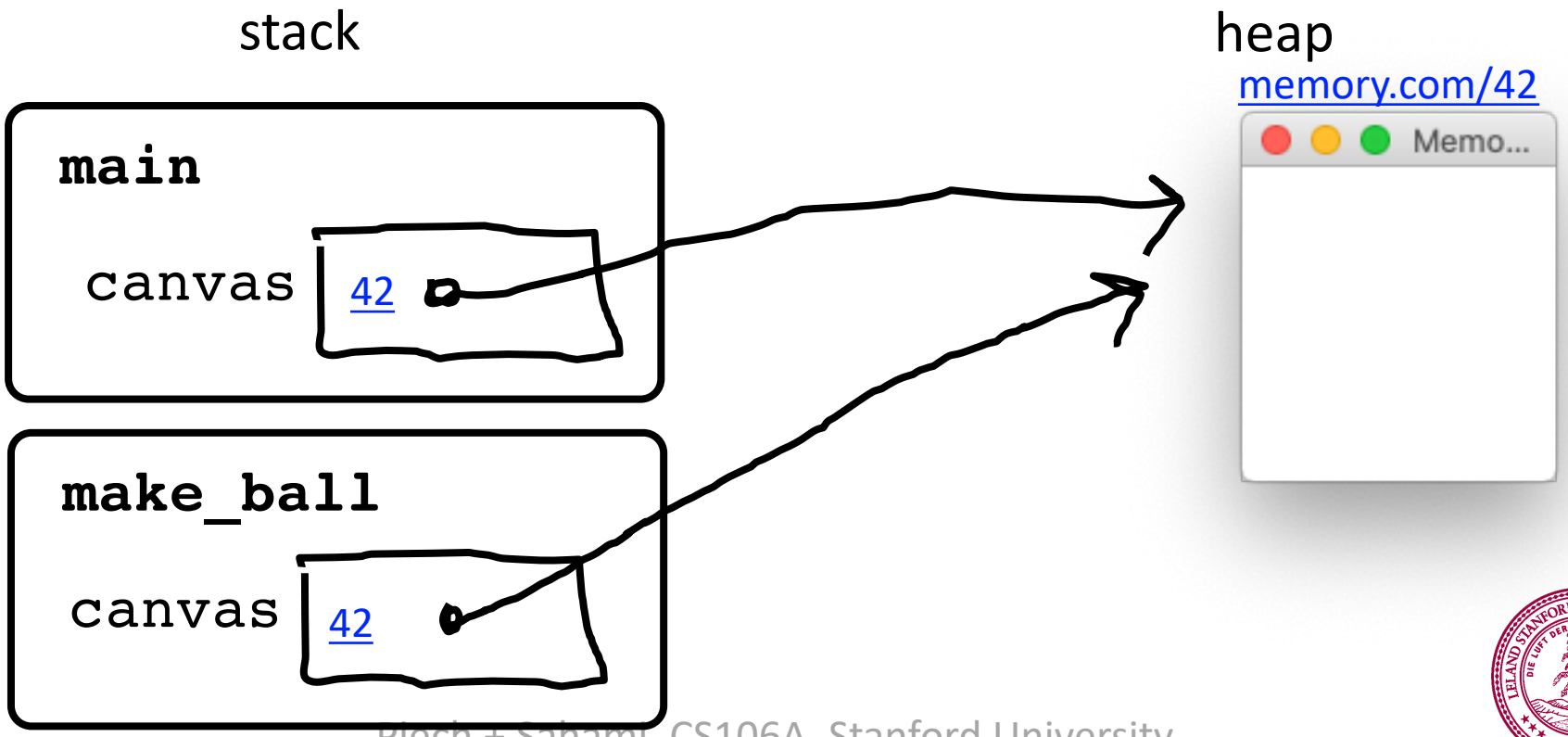
```
def main():
    canvas = Canvas(…)
    make_ball(canvas)

def make_ball(canvas):
    canvas.create_oval( … )
```



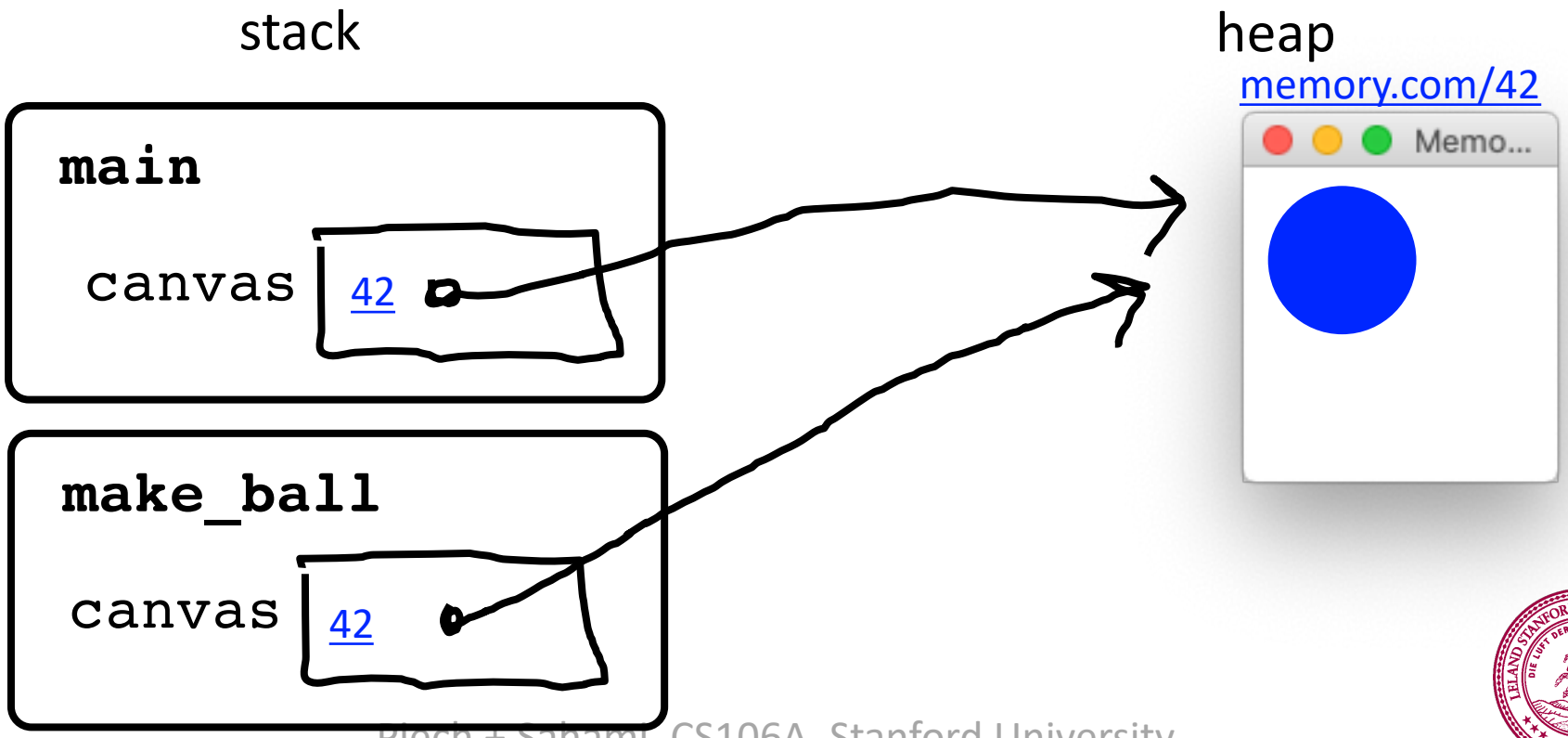
```
def main():
    canvas = Canvas(…)
    make_ball(canvas)

def make_ball(canvas):
    canvas.create_oval( … )
```



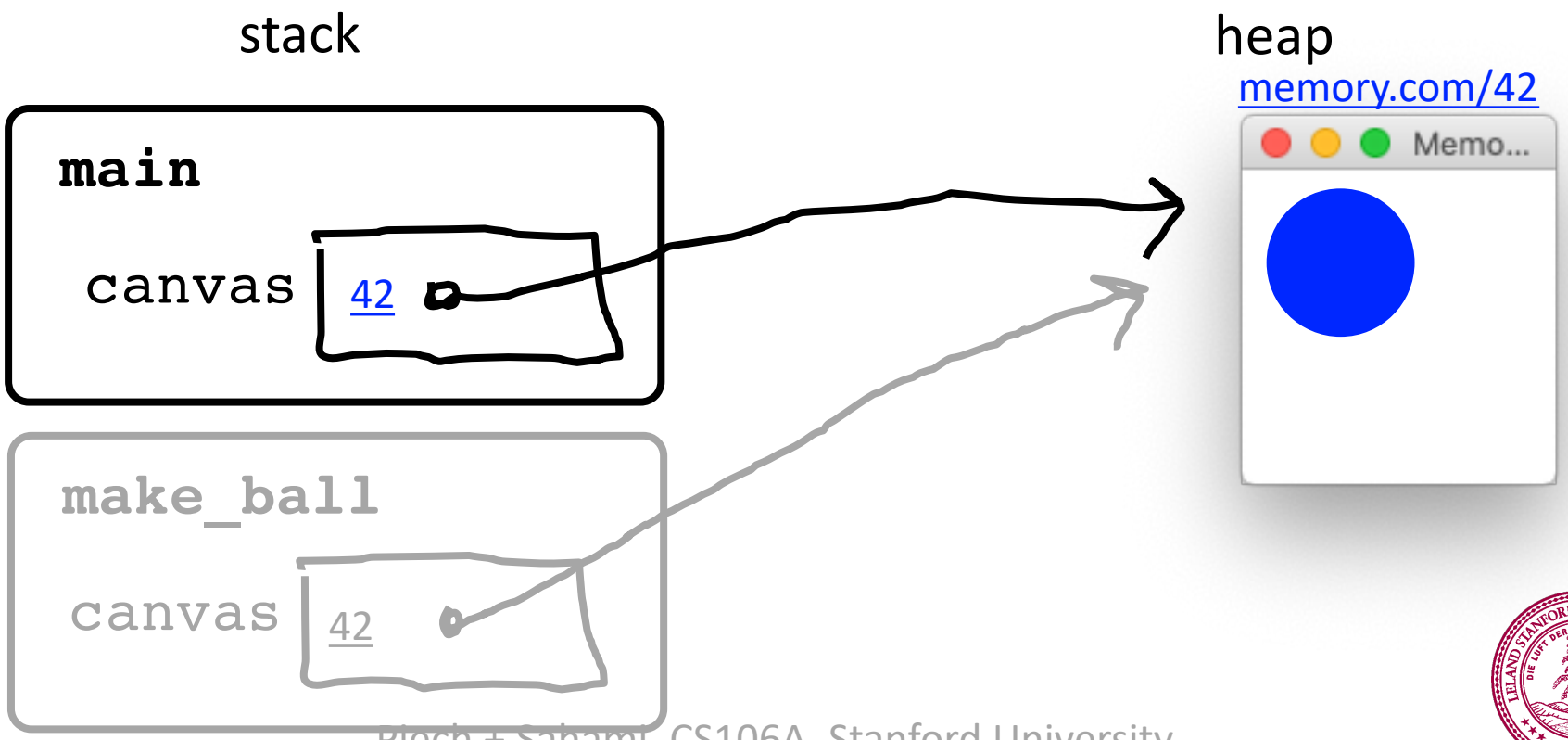
```
def main():
    canvas = Canvas(…)
    make_ball(canvas)

def make_ball(canvas):
    canvas.create_oval( … )
```



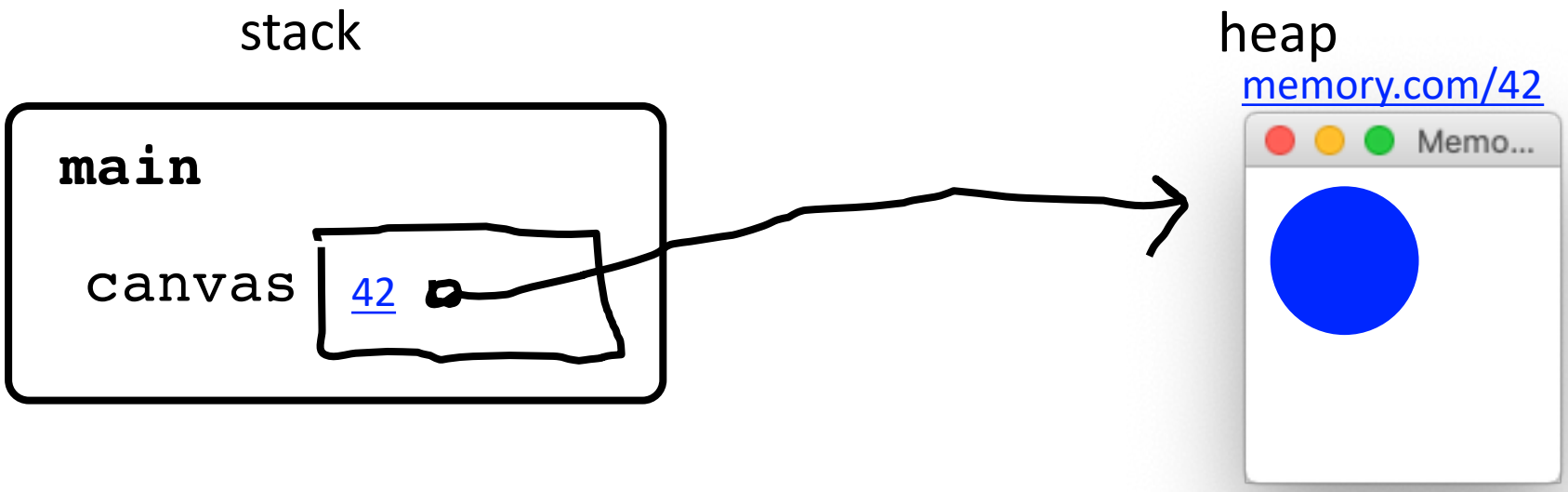
```
def main():
    canvas = Canvas(…)
    make_ball(canvas)

def make_ball(canvas):
    canvas.create_oval( … )
```



```
def main():
    canvas = Canvas(…)
    make_ball(canvas)

def make_ball(canvas):
    canvas.create_oval( … )
```



When passing variables, some act just like you are passing a URL.

That allows functions to modify the variable

Learning Goals

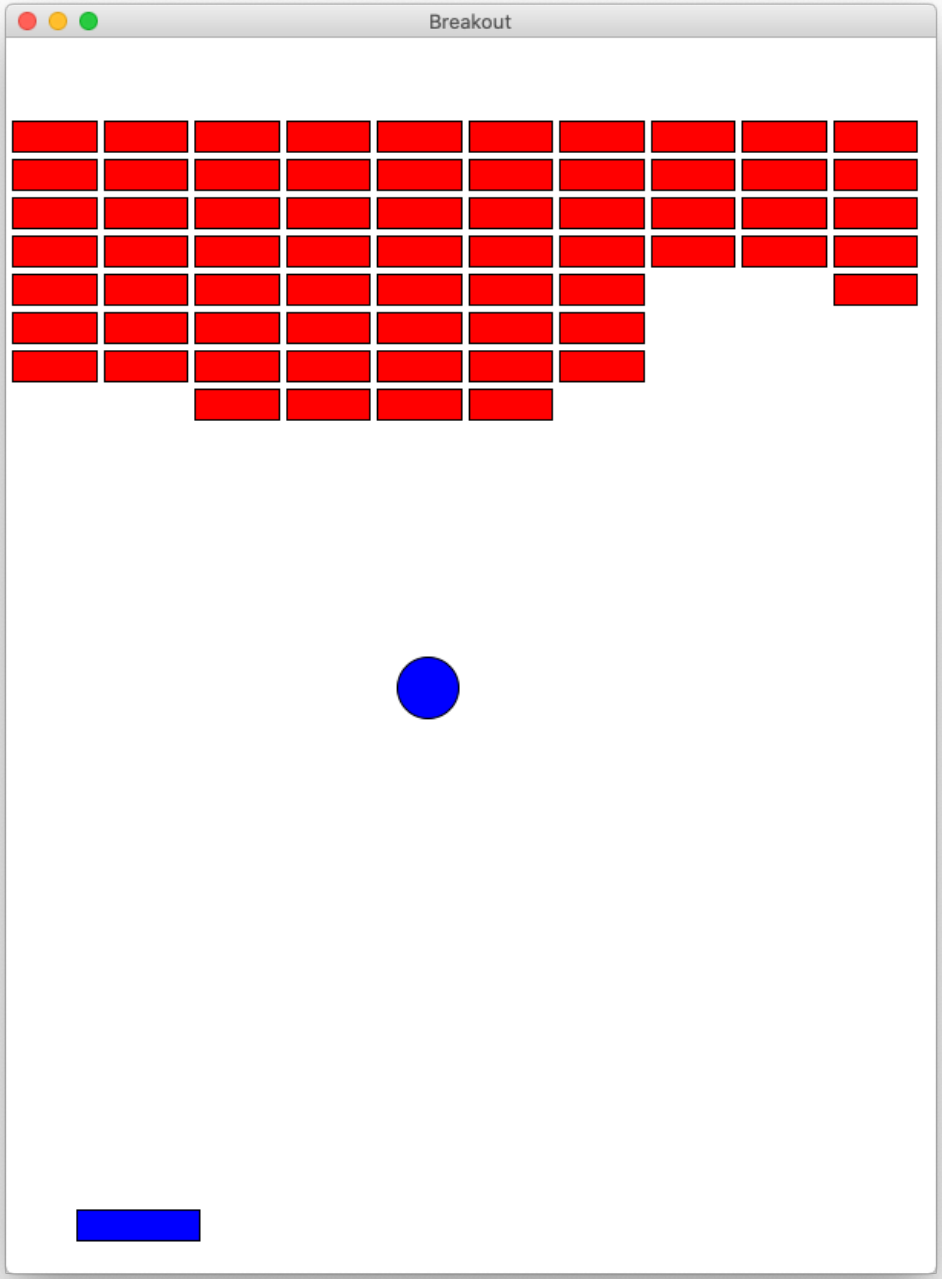
1. Feel more confident writing methods
2. Write animated programs

Special Graphics Functions

```
# move shape to some new coordinates
canvas.moveto(shape, new_x, new_y)

# move shape by a given change_x and change_y
canvas.move(shape, change_x, change_y)
```

Come back tomorrow to learn about lists!


Breakout

A Sticky Situation