Code in Place 2023

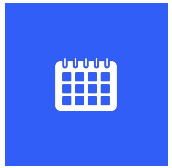
Stanford CS106A

Section - Week 5

Animations

David Tsai, Code in Place

Today's Agenda



1. Check-In

How are we all doing?

2. Course Announcements

Diagnostic Midterm, Final Project

3. Concepts Review

Animations

4. Practice Problem

"Scribble"

Before We Start

How are you all doing?
Hopefully your fourth week of CIP went well!

- What have you liked about the course?
- Is there anything blocking you from completing Week 1-4 assignments?
- Any major questions?

Course Announcements

Diagnostic Midterm

- **Deadline:** Saturday, May 27, 11:59pm, "Anywhere on Earth" timezone (UTC -12)
- **Duration:** 50 minutes
- **Details:** Course Forum pinned post by Prof. Chris Piech
  - Post title: "Week 5, Animation and Diagnostic"
- https://codeinplace.stanford.edu/cip3/diagnostic

Final Project

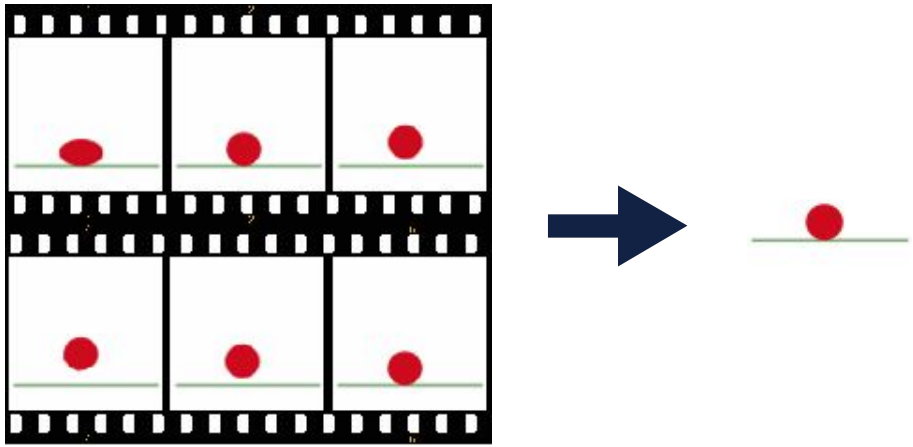
- Start thinking about ideas for a program you might want to build.
  - Open ended, anything you can imagine.
  - Can be something utilitarian, or something creative and fun.
  - An animation or game on the console or canvas.
  - Little app you can use in your personal life for family/work/study/etc.
- CIP 2020 Showcase:
  - https://compedu.stanford.edu/codeinplace/public/
- CIP 2021 Showcase:
  - https://codeinplace.stanford.edu/2021/showcase/

Concepts Review

Animation

- **Animation**: Displaying still image frames successively with a time delay between each frame to create an illusion of movement.

- The “bouncing ball” animation below consists of 6 frames, repeated indefinitely.



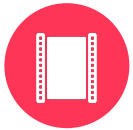
(Image source: https://en.wikipedia.org/wiki/Animation)

Graphics: Time Delay Between Frames

- `time.sleep(DELAY)`
  - This function tells our program to wait for some period of time specified by the `DELAY` parameter.
  - `DELAY` is a period of time, measured in seconds.

- Time delay is what lets us add the small gaps between our frames!
  - Experiment with the `DELAY` value to strike a balance between animation that is choppy vs “too smooth”.

Graphics: Animation Loop



- How to use graphics to make animations using a canonical **animation loop**:

```
CANVAS_WIDTH = 300
CANVAS_HEIGHT = 300
DELAY = 0.01            # delay time (measured in seconds)

def main():
      canvas = Canvas(CANVAS_WIDTH, CANVAS_HEIGHT)

      # Animation loop
      while True:
            # Draw graphics here.


            # Sleep for a delay period so that we see the animation.
            time.sleep(DELAY)
            canvas.mainloop()       # this line is not needed for CIP's web IDE
```

More Graphics Functions

- **Get the x and y coordinates of your mouse's position**
  - `mouse_x = canvas.get_mouse_x()`
  - `mouse_y = canvas.get_mouse_y()`
- **Move a shape to some new coordinates**
  - `canvas.moveto(shape, new_x, new_y)`
- **Move a shape relative to its current position by "offset_x" and "offset_y"**
  - `canvas.move(shape, offset_x, offset_y)`
- **Get the top-left coordinates of a shape**
  - `top_y = canvas.get_top_y(shape)`
  - `left_x = canvas.get_left_x(shape)`
- **Return a list of elements in a rectangle area**
  - `results = canvas.find_overlapping(left_x, top_y, right_x, bottom_y)`

Section Exercise: “Scribble”

Draw a circle wherever the mouse is located on the screen.