Lecture 30:

INTERACTION

CSC111: Introduction to CS through Programming

R. Jordan Crouser

Assistant Professor of Computer Science

Smith College

Announcements

- Slight schedule change: **no class on Weds. 25-Nov-2020**
- This week will be our **last week of study groups**
  - Starting next week, study group time slots will be used to workshop Final Projects and get help with debugging
  - Each team will be asked to choose a time slot that works best for (most of) the group members – survey coming to Discord shortly!

Debrief about FP1

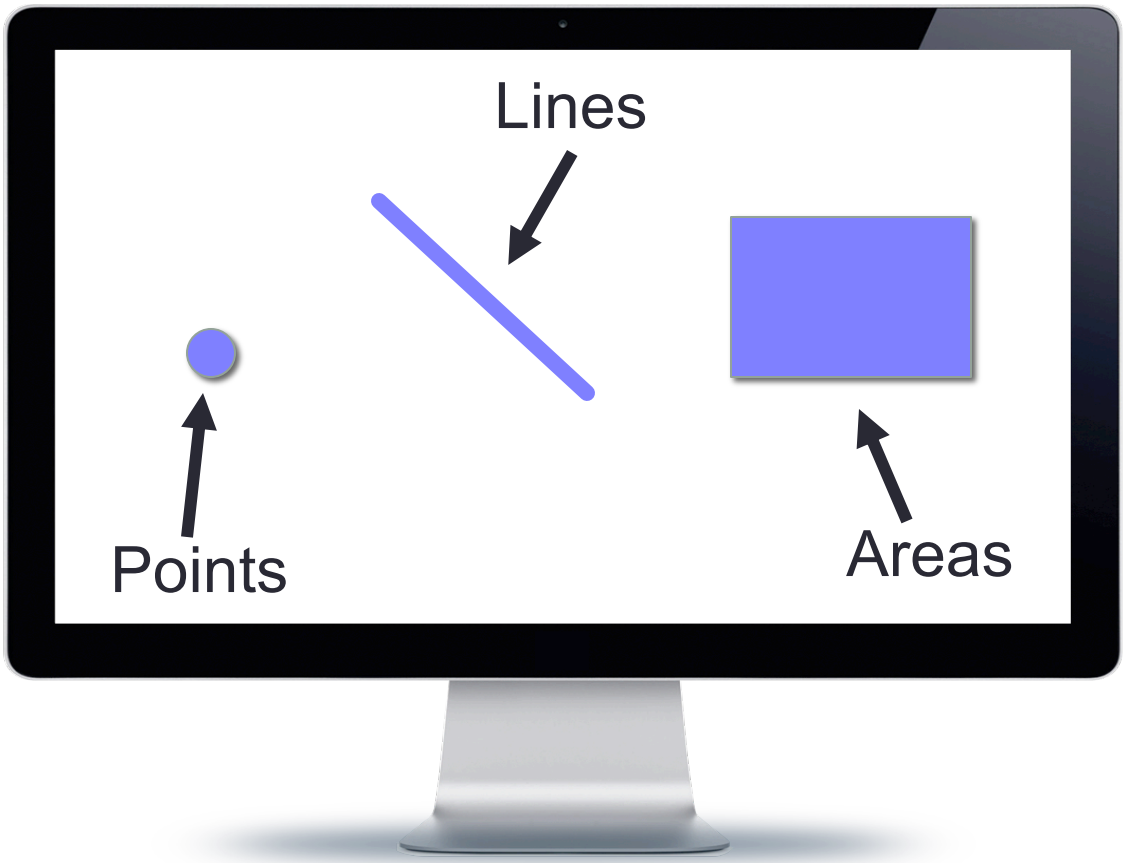
- 41 **amazing** submissions (16 solo projects)
- Ideas ranging from:
  - Videogames
  - Virtual gardens
  - Art galleries
  - Tools for college life (virtual tours, major requirement trackers, etc.)
  - …and many more!
- Written feedback will be out this afternoon
- Your team will then be registered as a group in Moodle
  - one submission for all group members
  - feedback automatically goes to all members

Outline for today

- ✓FP1 debrief
- Interaction basics
  - mouse
  - keyboard
- Starting Lab 10: Interactive Fish Tank

✓ Draw stuff

"graphical primitives"



✓ Draw stuff

using the **`graphics`** module

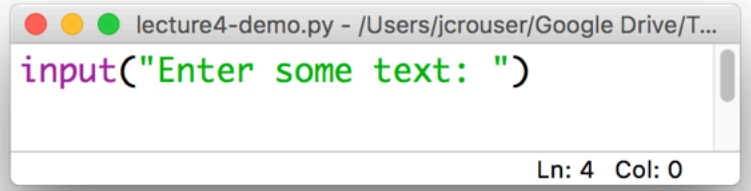
✓ Make it move

3. Get input from the user and react

Lecture 4: first experience with user input

The `.input()` function

- Python has a built-in `.input()` function that allows us to ask the user to type in information
- The `.input()` function takes in a value, which will be printed to the console as a prompt:



Interaction (def.)

- Ways for the user to **affect change** in what's happening in the program

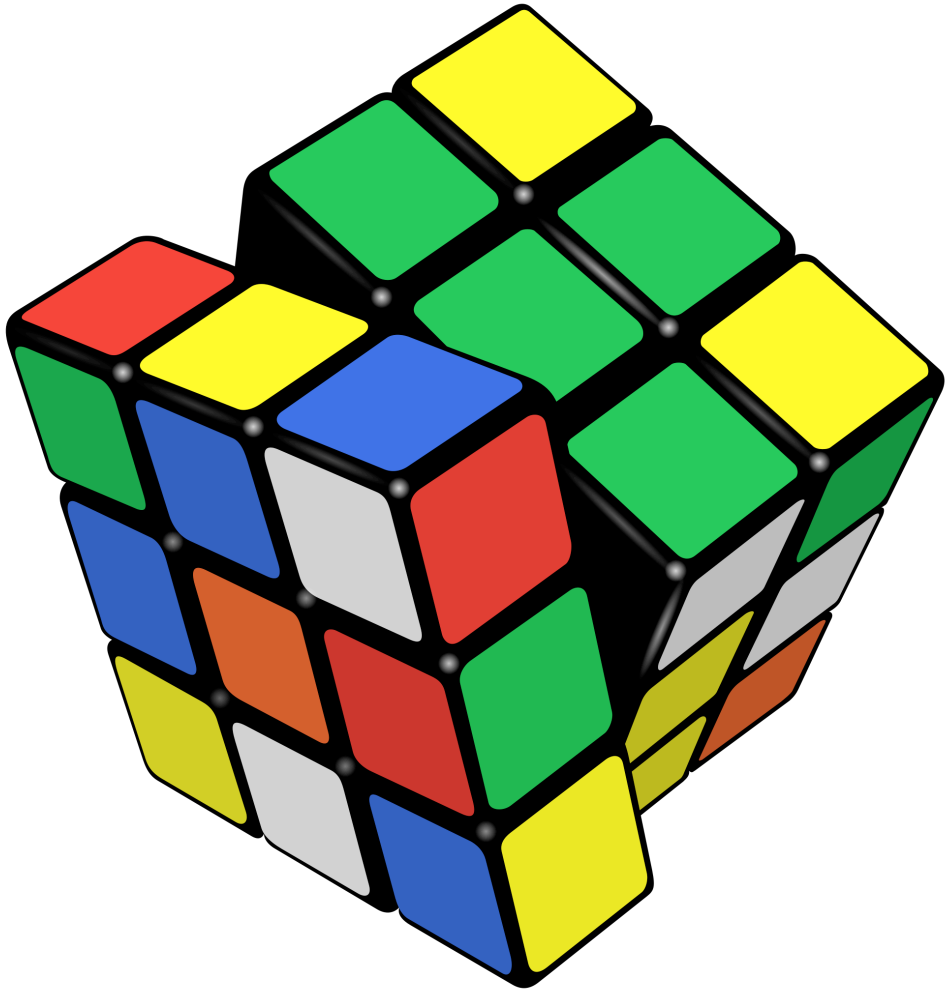
- Low level: **between human and interface**
  - the set of operations available
  - happens between the human and the physical computer

- High level: between **human and problem space**
  - a cognitive act *enabled* by the interface
  - happens between the human and the digital objects

Example: Rubik's Cube

What **low-level** interactions can you have?

What **high-level** interactions can you have?

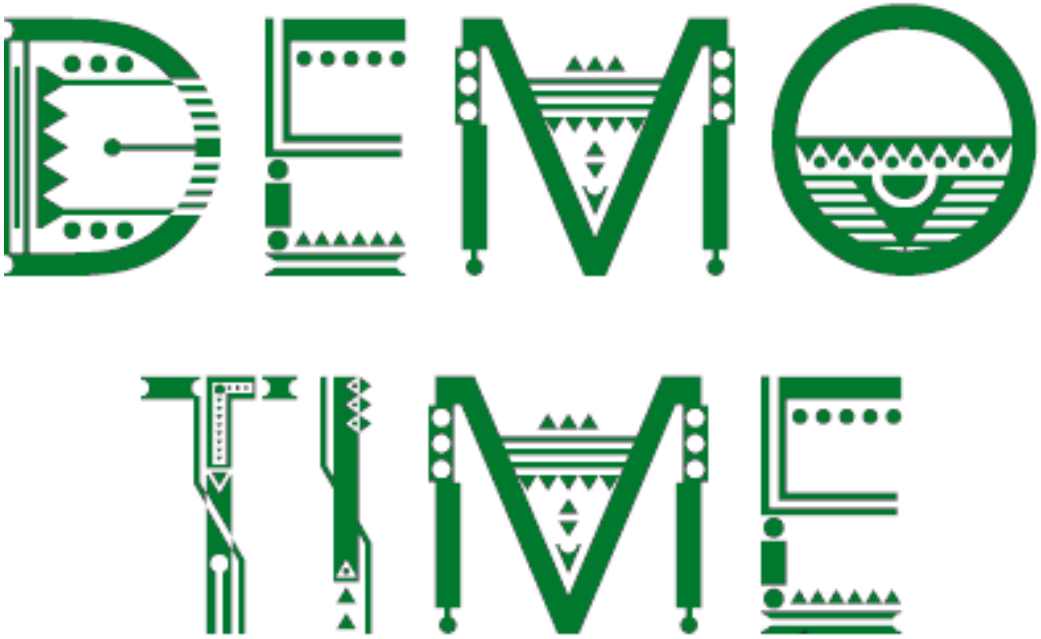


Low-level vs. high-level interactions

Interaction with **`graphics`** objects

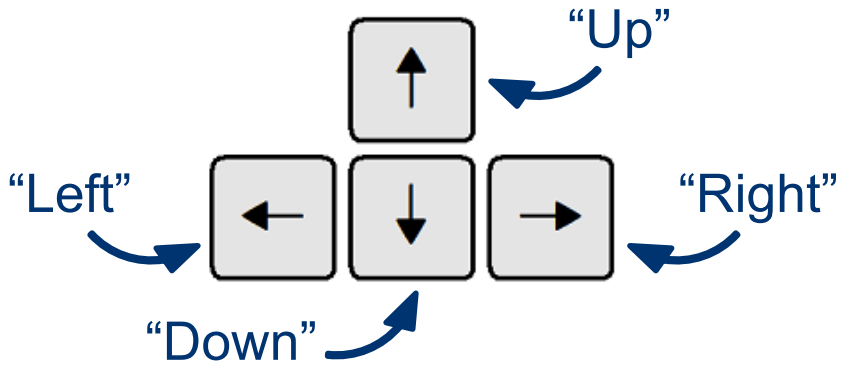
- The **`GraphWin`** object has methods to detect interactions
- Mouse:
  - **`.getMouse()`**: stop the program and wait for user to **click**
  - **`.checkMouse()`**: continuously check if the user has **clicked**
  - both return a **`Point`** object
- Keyboard:
  - **`.getKey()`**: stop the program and wait for user to **type**
  - **`.checkKey()`**: continuously check if the user has **typed**
  - both return a **`string`**

Our first interactive **`graphics`** program



Notes about keyboard interaction

- The strings returned by the **`.getKey()`** / **`.checkKey()`** methods ae called **keycodes**
- Some keys don't have an obvious letter attached to them, but their keycodes are still pretty intuitive, e.g.



- See also: **"space"**, **"Escape"**, **"minus"**, **"underscore"**, **"equal"**, **"plus"**, **"BackSpace"**, **"Return"**, **etc.**

Outline

- ✓Animation debrief
- ✓Interaction basics
  - ✓mouse
  - ✓keyboard
- In-Class Lab: Interactive Fish Tank

Challenge 1: press ’q’ to quit

Challenge 2: fish position

Challenge 3: fish frenzy

Coming up

- ✓Monday: Interaction
- Lab 10: Interactive Fish Tank
- Wednesday: Handling Exceptions
- Friday: Special Topic – Ethical Issues in Tech
- FP2: Persona, Paper Prototype, and Architecture Diagram