Developing an App II

CSE 120 Spring 2017

Instructor: Justin Hsia

Teaching Assistants: Anupam Gupta, Braydon Hall, Eugene Oh, Savanna Yee

The Art of Data: How Artists With IBM Watson Turned Data Into Unusually Insightful Portraits

When it comes to the installation at Cadillac House, Watson's role was that of collaborator, processing and unpacking data galore to inform the artists' works. Watson was able to recognize common semantic features and themes — even when they weren't explicitly called out in the text. From there, artists took the reins and interpreted Watson's findings in their preferred mediums.

- http://nymag.com/daily/intelligencer/2017/04/art-with-watson.html

LCM Report Wrap-Up

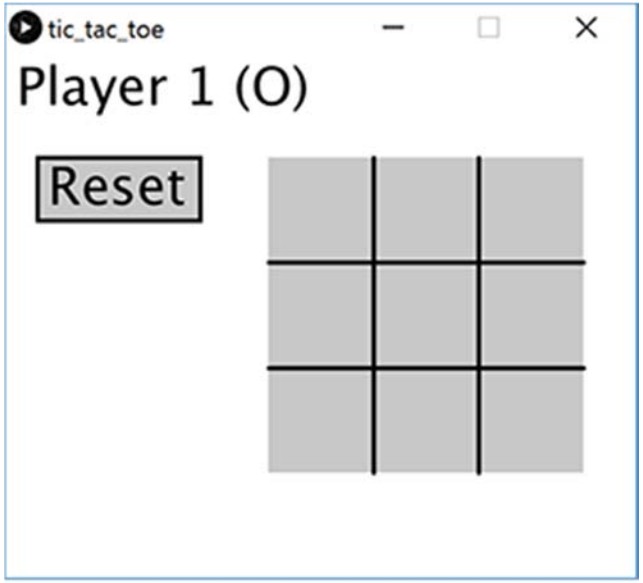
living computers
museum + labs

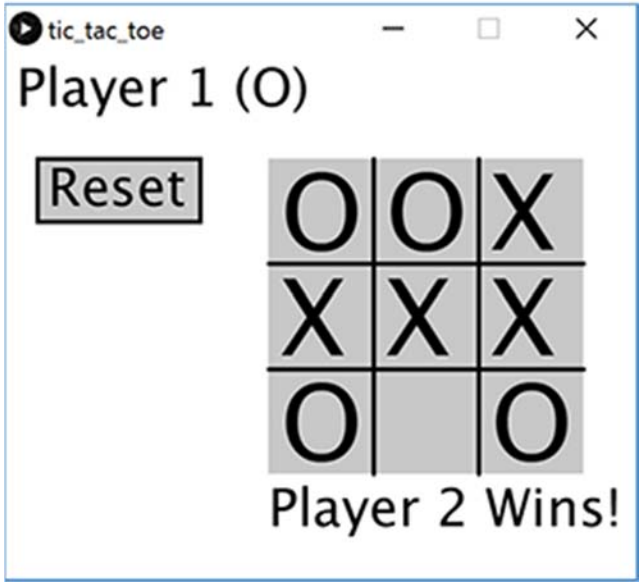
Administrivia

- Assignments:
  - Tic-Tac-Toe due Friday (5/19)
  - Project Proposal due Saturday (5/20)
  - Innovation Exploration post (5/21)
- Big Ideas lecture on Friday: Artificial Intelligence
  - Reading Check (5/18) before lab section

Tic-Tac-Toe

- Put together an app from scratch!
  - Work with a partner
  - Game states, grid clicking, reset button, *winning condition*





Final Project

- Three parts:
  - Proposal due Saturday (5/20)
    - Includes project name and "storyboard"
  - Update due Thursday (5/25) in lab
  - Project due Friday (6/2)
    - Includes video and README

- Single program, done with a partner
  - Must be significantly more substantial than Creativity Assignments
  - Must include 3+ "hand-created" assets

Outline

- **15 Puzzle, continued**

Where We Left Off

- Implement game mechanics of sliding puzzle of numbered square tiles

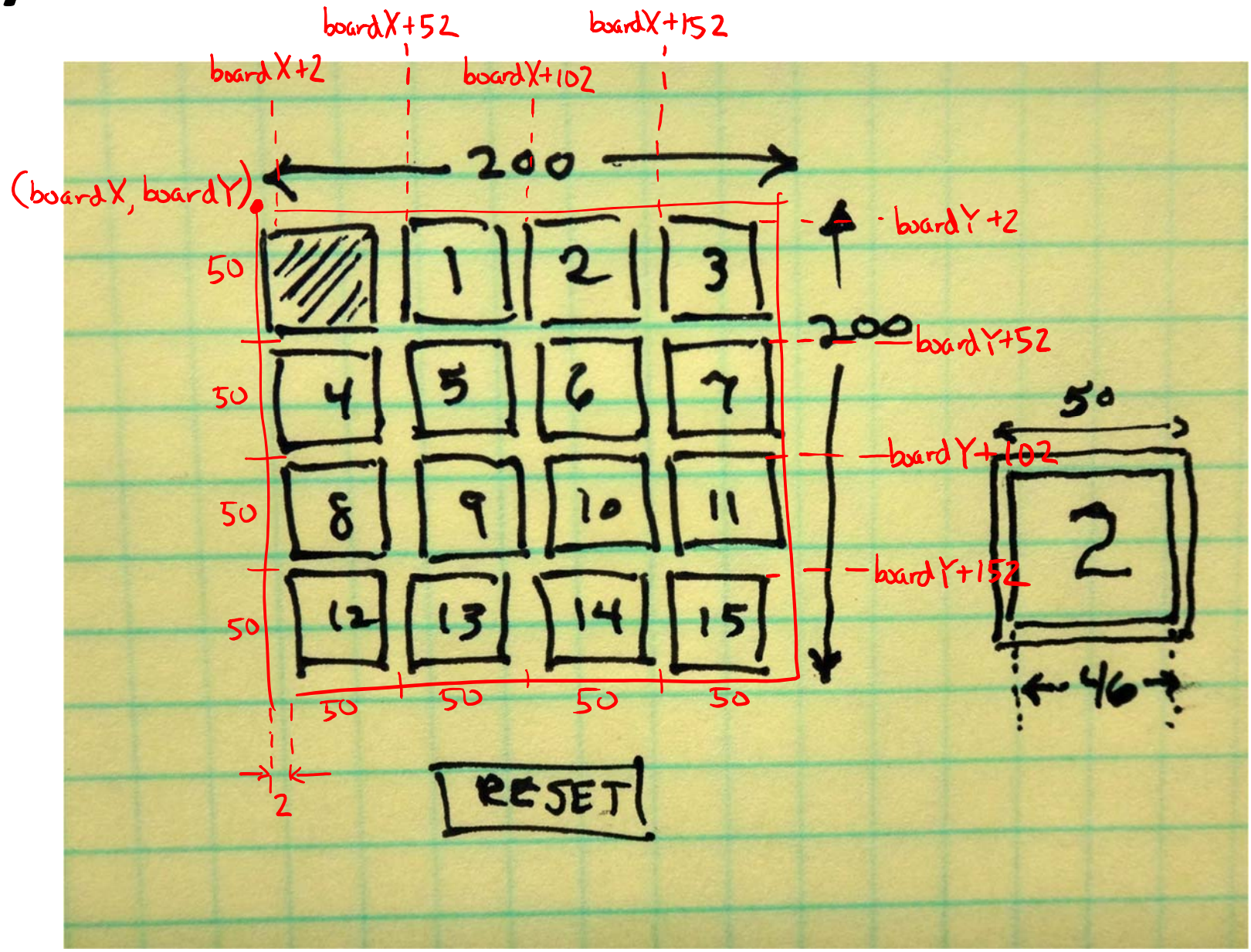
- **Done:**
  - Draw reset button
    - Implement reset function
  - Draw game board (border and tiles)
  - Implement board state and display numbers on tiles
    - Don't display tile with value 0 (empty/open)
  - Detect clicks on reset button and on grid

Where We Left Off

- Implement game mechanics of sliding puzzle of numbered square tiles

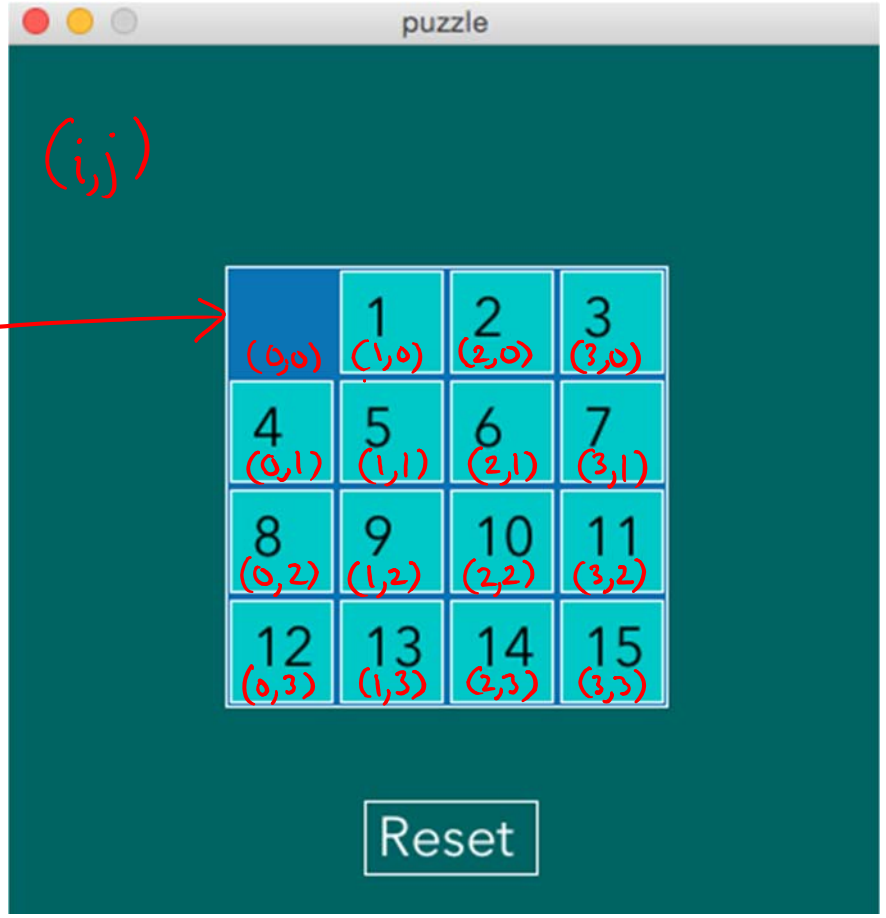
- **To Do:**
  - Determine if click is adjacent to open square
    - Can only be in up/down/left/right directions
  - Implement “sliding” (swap function)
  - [if time] Extra functionality!

Layout Reminder



Tile Grid Reminder

don't draw tile for open space (if tile value == 0)



i	j	array index
0	0	0
1	0	1
2	0	2
3	0	3
0	1	4
1	1	5
2	1	6
3	1	7
0	2	8
⋮		⋮

index = 4*j + i

row, col

following (x,y) convention

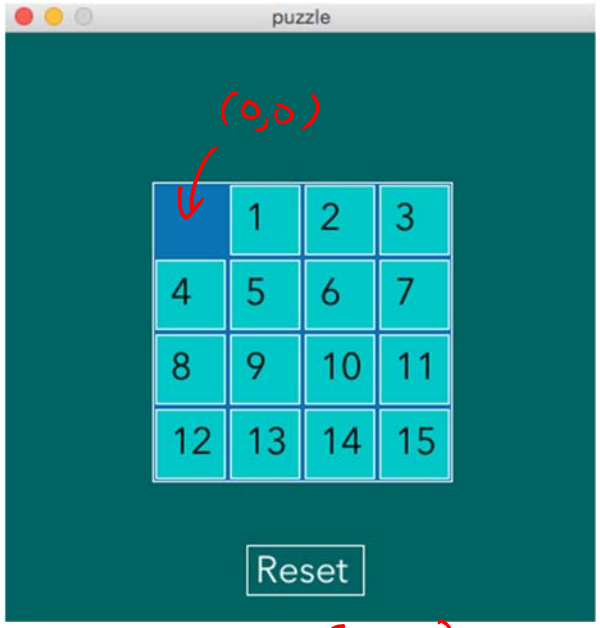
int index(int col, int row)

probably going to do this calculation a lot so write a separate function to do it!

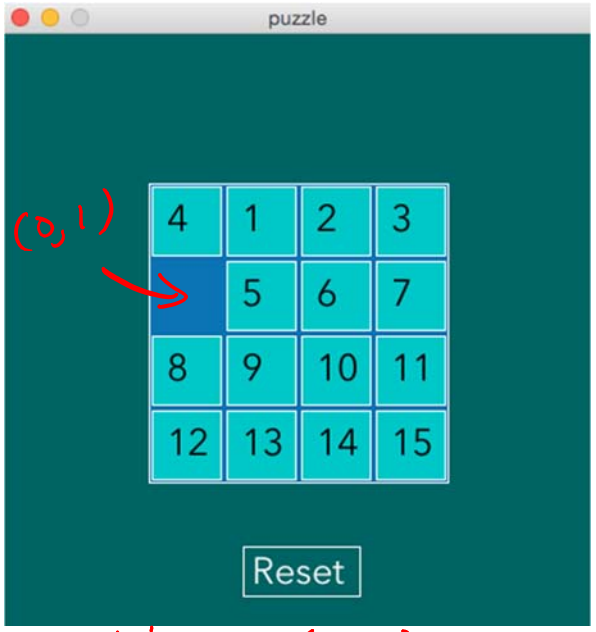
Tile Movements

swap

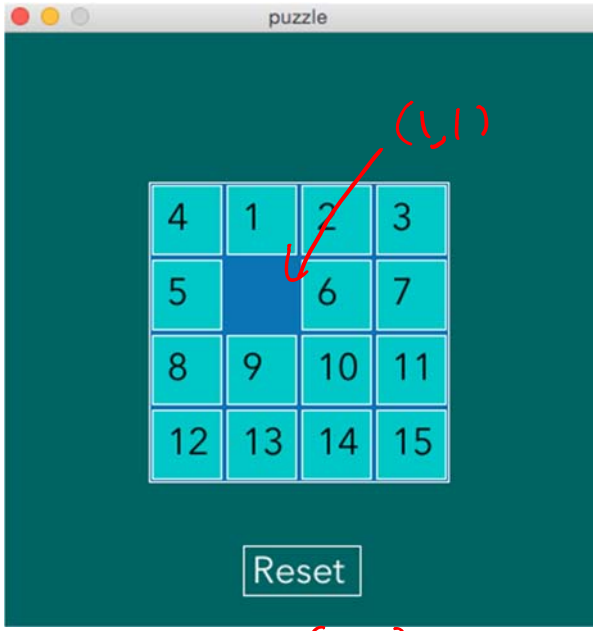
```
int temp = a;
a = b;
b = temp;
```



neighbors: (0,1)
(1,0)



neighbors: (0,0)
(1,1)
(0,2)



neighbors: (1,0)
(0,1)
(1,2)
(2,1)

---

① create variables to store current open tile grid coordinates (openX, openY)

② check if neighbor using: abs(openX - gridX)
abs(openY - gridY)

If Time: Extensions

- Change Reset button hover color
  - Create `overReset()` function that returns a `boolean`
- Randomize initial tile placements
  - Tricky! How to avoid repeats?
- Check for win condition: tiles ordered 0-15
  - **Note:** This is not achievable for many randomized starting orderings

Summary

- Sketched the idea on paper
- Planned out coding representations
- Started with the things we knew how to do first
- Built on previous work by adding one function or idea at a time
- Ran the program after *every* improvement to make sure that it worked correctly
  - Unit and integration testing!!!