

Developing an App

CSE 120 Spring 2017

Instructor:

Justin Hsia

Teaching Assistants:

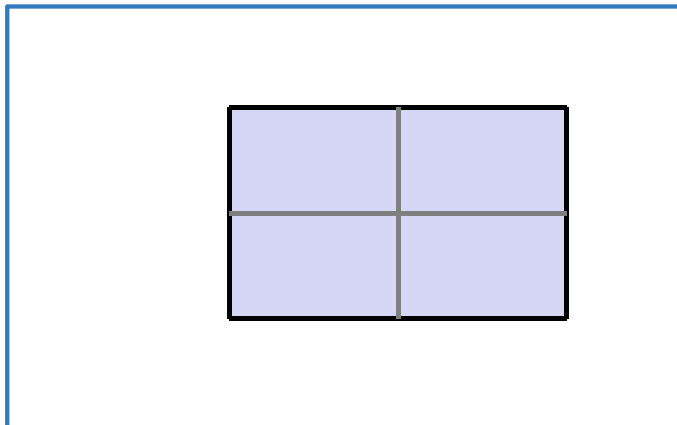
Anupam Gupta, Braydon Hall, Eugene Oh, Savanna Yee

Administrivia

- ❖ Assignments:
 - Word Guessing due Thursday (5/11)
 - Living Computers Museum Report due Sunday (5/14)
 - Birthday Visualization due Monday (5/15)
 - Innovation Exploration post (5/21)
- ❖ Guest lecture on Friday: Security
 - Reading Check (5/11) before lab section

Birthday Visualization

- ❖ Data visualization for birthday frequencies
 - Learn how to read file data into Processing
 - Use color to visualize numbers
 - Detection of mouse location within a *grid*



```
int x_index, y_index;
if( (mouseX >= x)      &&
    (mouseX <= x + w) &&
    (mouseY >= y)      &&
    (mouseY <= y + h) ) {
    x_index = (mouseX-x) / (w/2);
    y_index = (mouseY-y) / (h/2);
}
```

Goals for Today

- ❖ Collaborate!
 - TAs will also wander around to assist and answer questions
- ❖ Practice testing, debugging, and refactoring
- ❖ Don't just wait for solution!
 - The real learning comes from exploring, creating, and making mistakes
 - Any real-world problem requires *you* to come up with a solution, not just follow directions

Outline

- ❖ **The Game**
- ❖ Design Phase
- ❖ Coding Phase

15 Puzzle

- ❖ Sliding puzzle that consists of numbered square tiles in random order with one tile missing
 - Also known as “Mystic Square”



- ❖ We will program just the game mechanics
 - Won't do winning condition, since not all game states are solvable

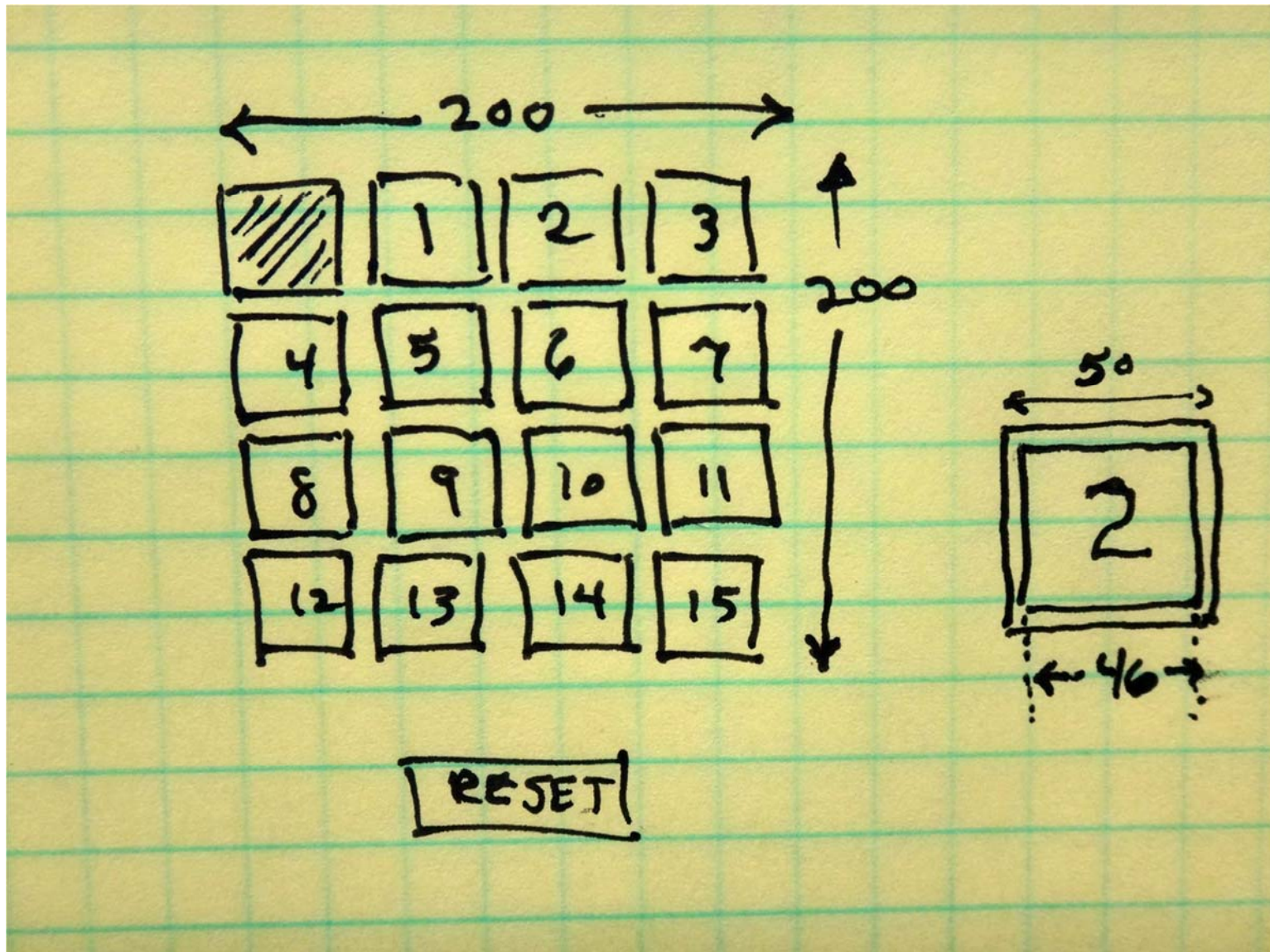
App Mechanics

- ❖ Tiles numbered 1-15 are shown on game board
 - One “open” or “empty” square
- ❖ Clicking a tile next to the empty square will “slide” that tile into the empty space
 - Clicking other tiles has no effect
 - Clicking outside of the game board has no effect
- ❖ Include a Reset button to return the game board to its initial state

Outline

- ❖ The Game
- ❖ **Design Phase**
- ❖ Coding Phase

Design the Layout



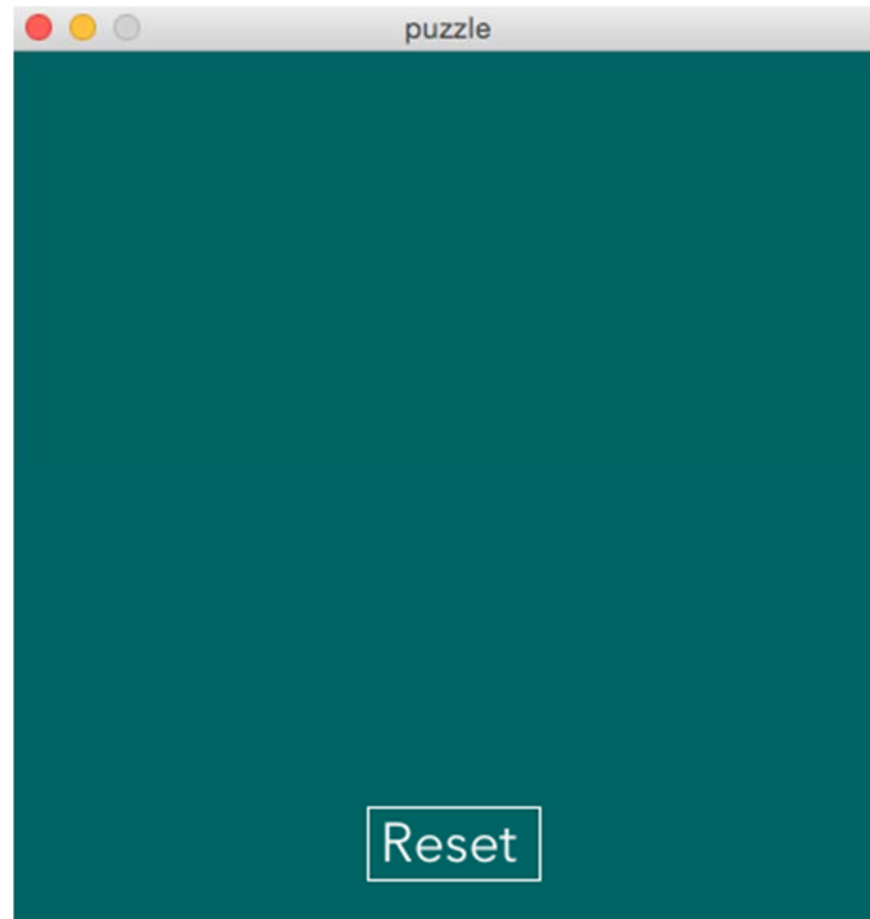
Coding Decisions

- ❖ How to represent the state of the game board?
- ❖ How to implement the “slide” functionality?
- ❖ How to respond to clicks?

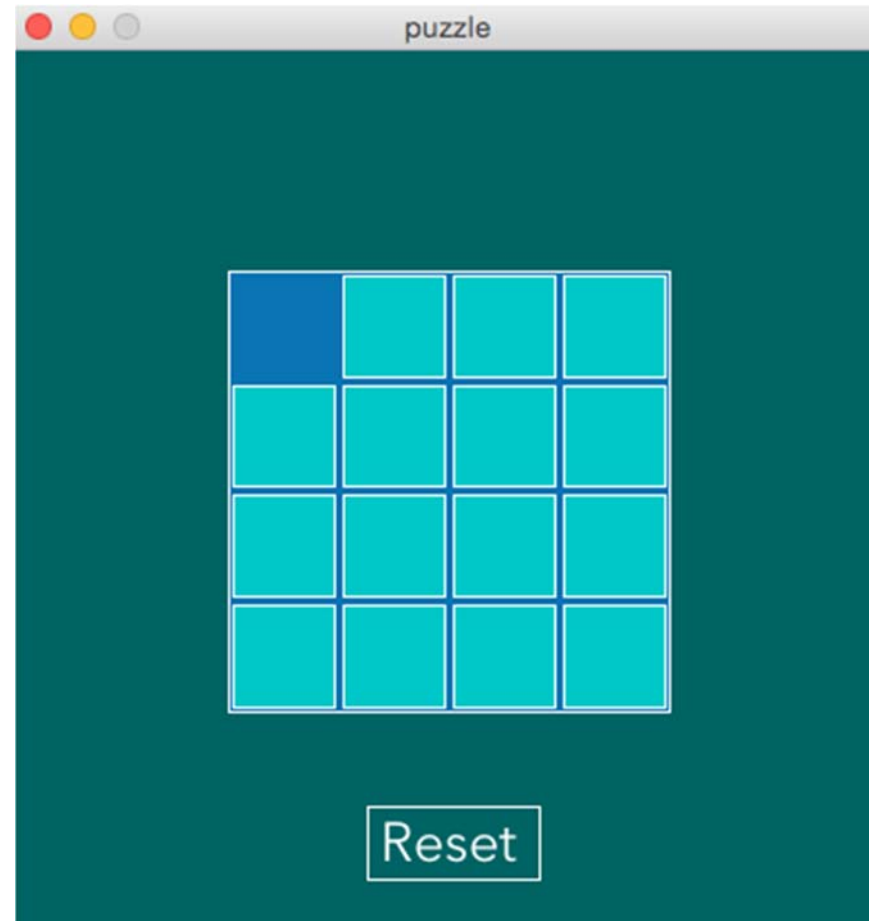
Outline

- ❖ The Game
- ❖ Design Phase
- ❖ **Coding Phase**

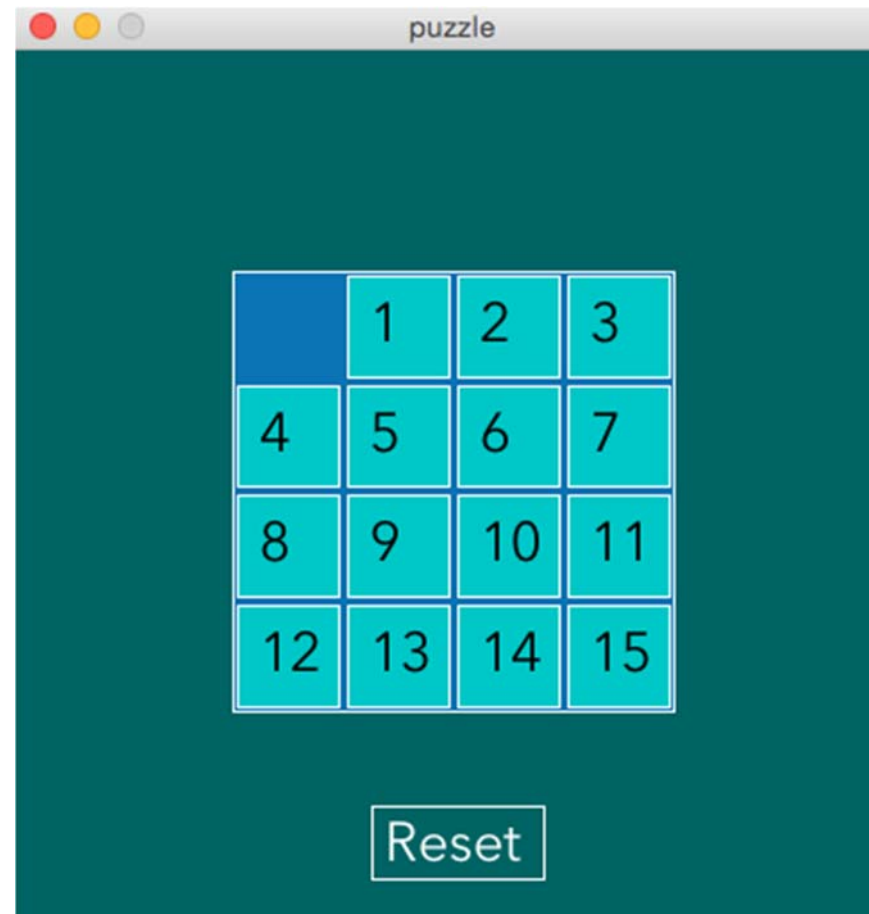
Create the Reset Button



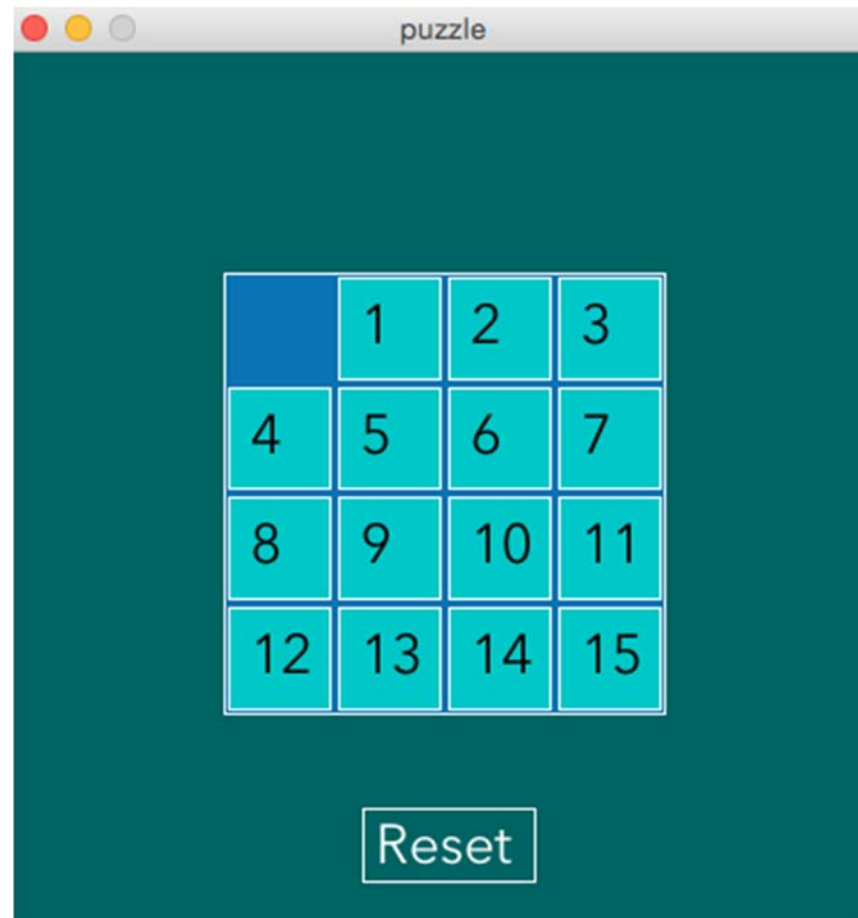
Create the Tile Layout



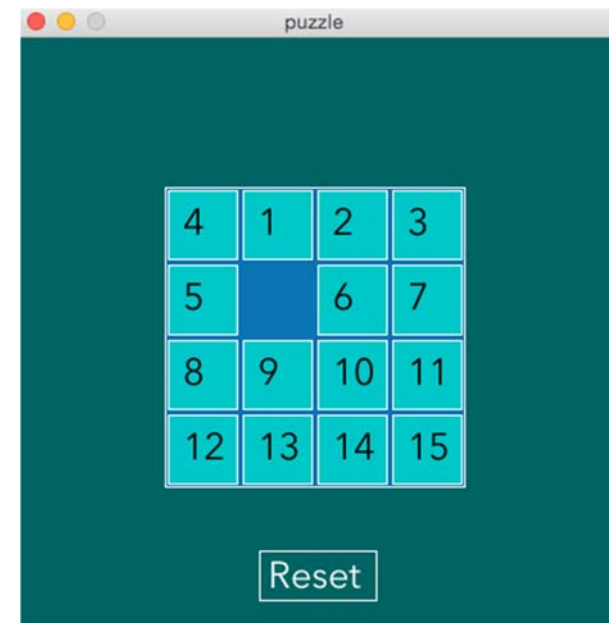
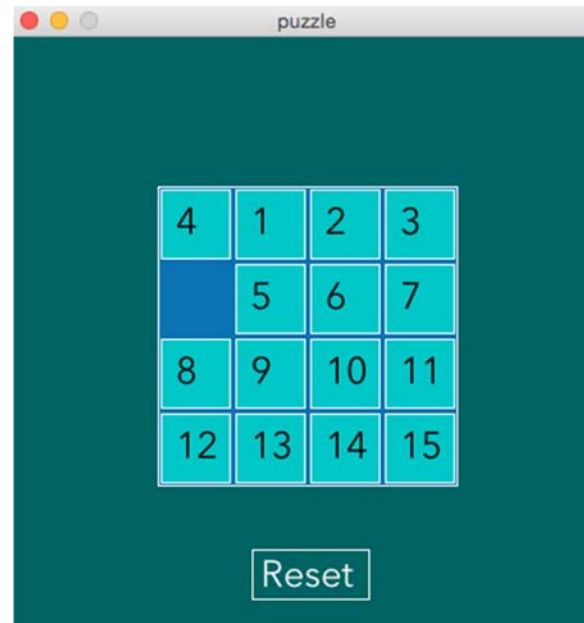
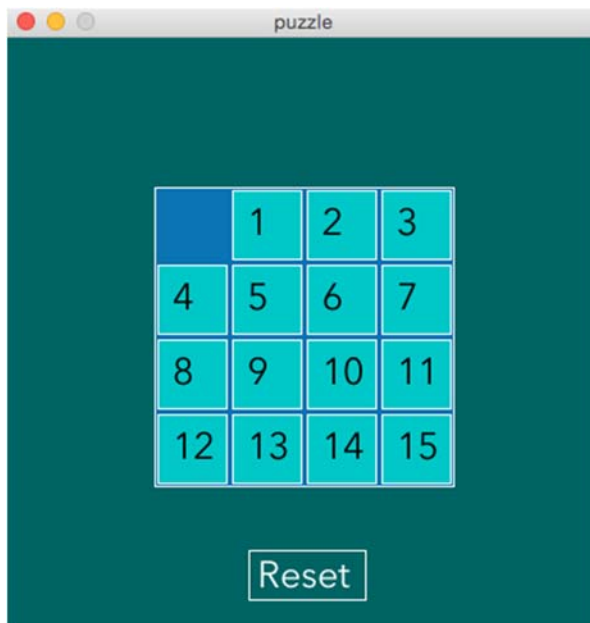
Add Numbers to Tiles



Reset Functionality



Tile Movements



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


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!!!