Emmanuel Macron and how political campaigns will never be the same

The internet has become so big that trolls started to have a significant influence on the results of the elections. If you think Macron’s election proves that Facebook, Twitter, YouTube, 4chan, Reddit and other social media platforms don’t have a fake news problem anymore, you’re wrong. It’s been a nightmare, and it’s still going to be a nightmare for future elections.

It’s still hard to fix all your security weaknesses. It’s clear that all elections are going to be like this now. Political team members will all need to take a course on “Encryption 101” before joining a campaign.

- [https://techcrunch.com/2017/05/07/emmanuel-macron-and-how-political-campaigns-will-never-be-the-same/](https://techcrunch.com/2017/05/07/emmanuel-macron-and-how-political-campaigns-will-never-be-the-same/)
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

```java
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);
    // get rounded down to an integer
}
```
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?
  ```
  int array of size 16
  {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15}
  ```

- How to implement the “slide” functionality?
  ```
  swap values in array
  ```

- How to respond to clicks?
  1. detect click on Reset button (reset)
     - rectangular region
  2. detect click on tile (possibly slide)
     - rectangular region of game board
     - calculate tile coordinates of click
Outline

- The Game
- Design Phase
- Coding Phase
Create the Reset Button
Create the Tile Layout

nested for-loops!
loop variable i will indicate column
loop variable j will indicate row
Add Numbers to Tiles

- Don't draw the tile for open space (if tile value == 0)

- Array index formula:
  \[ \text{index} = 4 \times j + i \]

- Example:
  \[
  \begin{array}{c|c|c}
  i & j & \text{array index} \\
  \hline
  0 & 0 & 0 \\
  1 & 0 & 1 \\
  2 & 0 & 2 \\
  3 & 0 & 3 \\
  0 & 1 & 4 \\
  1 & 1 & 5 \\
  2 & 1 & 6 \\
  3 & 1 & 7 \\
  0 & 2 & 8 \\
  \end{array}
  \]
Reset Functionality

In this state, all tile values are equal to their array indices:

```
tiles[0] = 0;
tiles[1] = 1;
tiles[2] = 2;
```

So can condense into a for-loop:

```
for(int i = 0; i < 16; i++) {
    tiles[i] = i;
}
```
Tile Movements

\[
\begin{align*}
\text{swap} & \quad \begin{cases}
\text{int} \ temp = a; \\
\text{a} = b; \\
\text{b} = \text{temp};
\end{cases}
\end{align*}
\]
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!!!