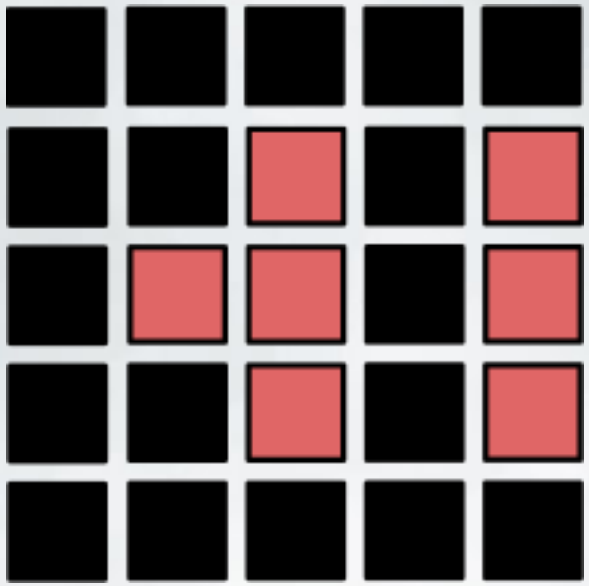
CIS 122

Lights Out

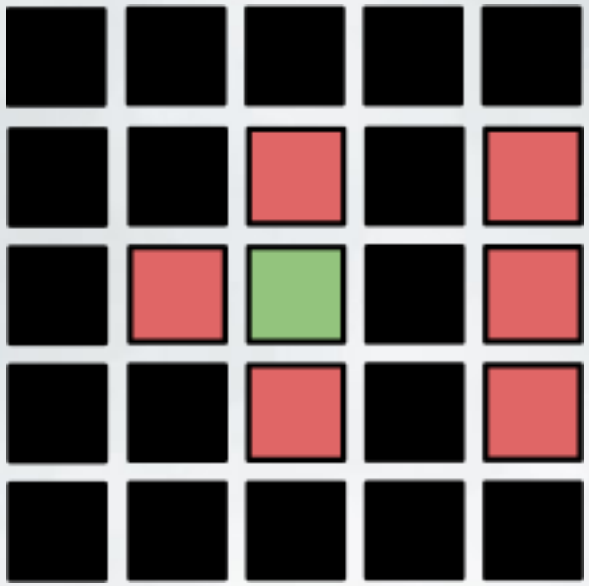
The Game

- In the game of Lights Out, you have a grid of lights
  - Your job is to turn them all off
- You can press lights to toggle them
  - But you also toggle every adjacent light
- The goal is to turn all the lights off as fast as possible
  - Few button presses

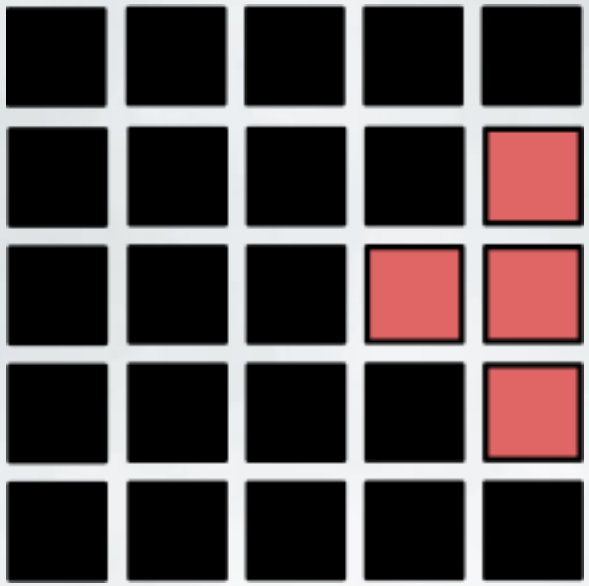
The Game



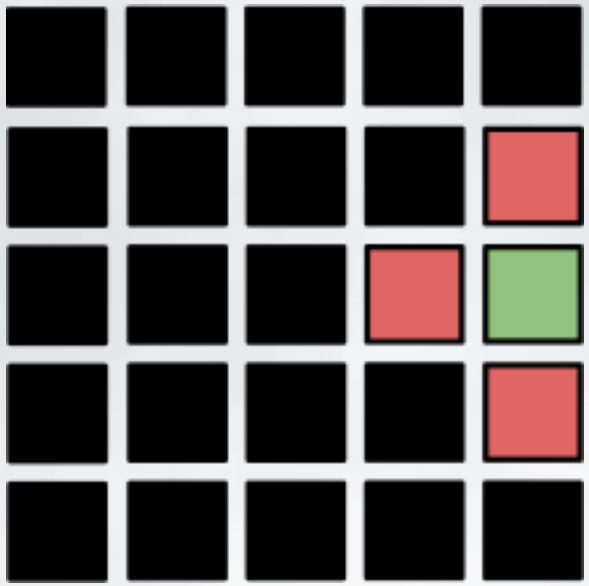
The Game



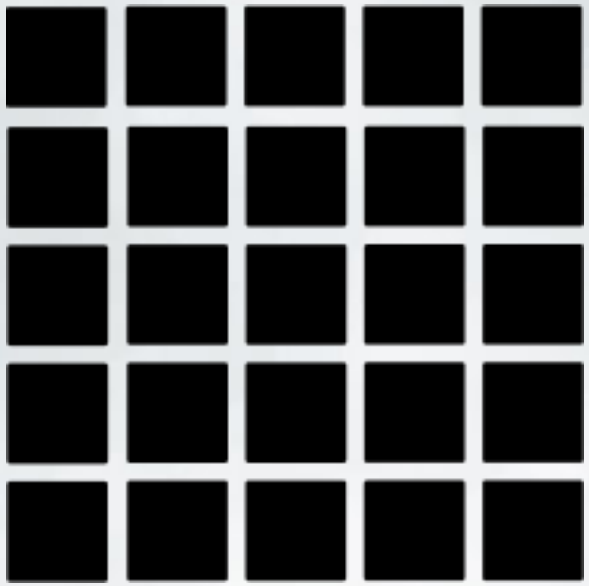
The Game



The Game



The Game



The Problem

- I have provided a skeleton LightsOut class
  - Constructor
  - Representation
- You're responsible for filling in the rest
  - Toggling lights
  - Pressing lights
  - Checking if all lights are off
  - Initially scrambling lights
- Ultimately, you'll use this class to code an interactive game

The Class

- What information does our LightsOutClass store?
  - self.grid
  - self.numRows
  - self.numCols
- But what are those properties?

The Class

- What information does our LightsOutClass store?
  - self.grid
  - self.numRows
  - self.numCols
- But what are those properties?
  - self.grid is a **nested list** of lights (characters)
  - self.numRows is the **integer** number of rows
  - self.numCols is the **integer** number of columns

The Class

- Right now, the constructor takes no arguments
  - It always constructs a 5 x 8 Lights Out grid
- But we could change that...

Toggling Lights

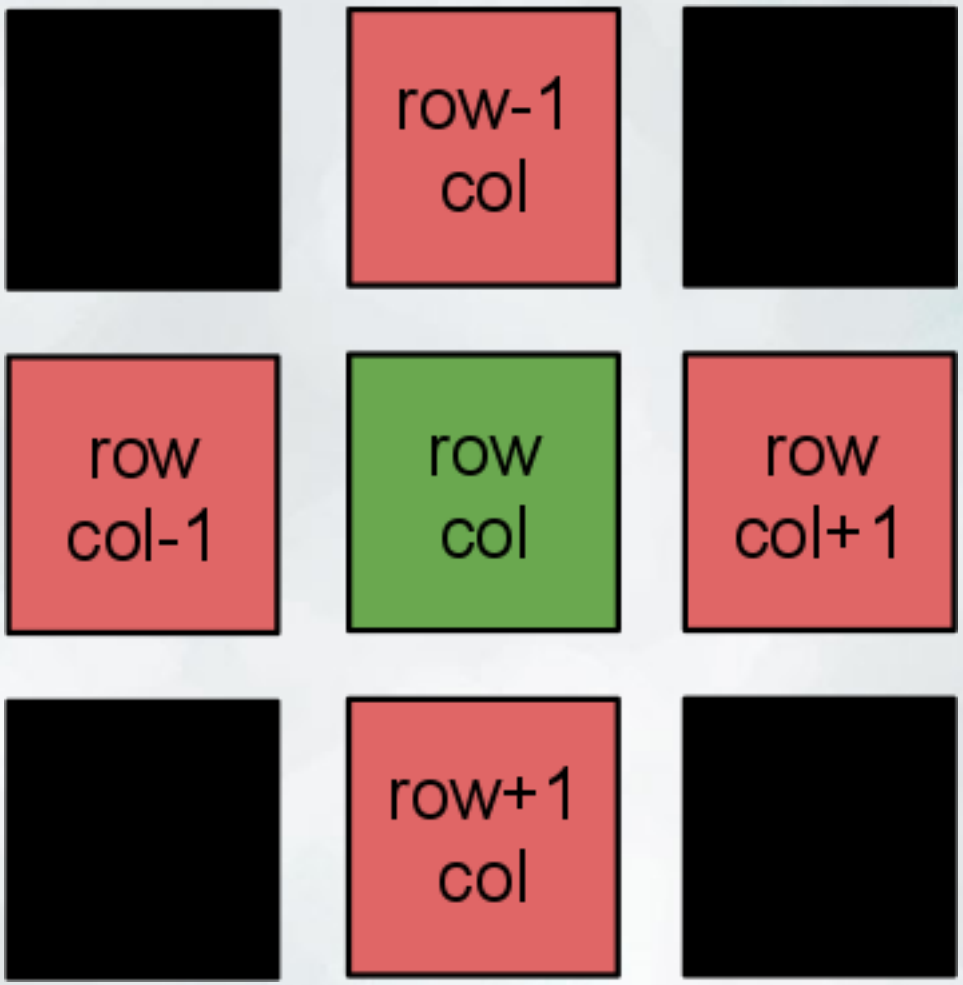
- Define toggle(self, row, col)
  - Toggle the light at the given position
- What does it mean to toggle a light?
  - If it's on, turn it off
  - If it's off, turn it on
- How do we access elements in a nested list?
  - nestedList[x] gets row x of the list
  - nestedList[x][y] gets the yth element of that row

Pressing Buttons

- Define press(self, row, col)
  - Toggle light at the given position
  - Toggle all lights adjacent to that position

- Given a specific row and column
  - What are the coordinates of the adjacent lights?

Pressing Buttons



Pressing Buttons

- So what's the plan?
  - Toggle the given light
  - We toggle all four adjacent lights
- Any issues?

Pressing Buttons

- So what's the plan?
  - Toggle the given light
  - We toggle all four adjacent lights
- Any issues?
- Not all lights have four adjacent lights
  - Sides only have 3
  - Corners only have 2
- Before you toggle a light, make sure it exists!

Checking your Lights

- Define allOff(self)
  - Return true if all lights are off
  - False otherwise
- Search through nested list
- Make sure no lights are on
- If you need inspiration, look over monday's slides

Scrambling the Grid

- Define scramble(self, num)
  - Scramble the lights on the grid
  - Randomly press num lights
- How do we scramble things?
  - Select a random row and column
  - Press that light
  - Repeat

Playing the Game

- Once your class is done, put it all together
  - Define the playGame function
- Make a new LightsOut object
- Scramble the lights
- While there are still lights on...
  - Ask user for a light (ask for a row, ask for a column)
  - Press that light
  - Repeat
- Print out a congratulatory message

Extensions

- Keep track of how many presses the user takes
  - "You took 10 moves"
- Allow the user to select a game size
  - Small - 5 x 5
  - Medium - 7 x 7
  - Large - 10 x 10
  - Custom - ???
- Incorporate turtle graphics
  - I would be very impressed