Professor Leen-Kiat Soh
Department of Computer Science and Engineering
University of Nebraska, Lincoln, NE
E-mail: lksoh@cse.unl.edu

Computational Thinking
Informatics
Chutes and Ladders

http://img2.wikia.nocookie.net/__cb20130510160248/despicableme/images/e/e9/Despicable-Me-Minions_thumb10.jpg

What is Computational Thinking?

- A way of thinking for *logically* and *methodically* solving problems
  - E.g., *purposeful, describable, replicable*
- Includes *skills* such as
  - Decomposition
  - Pattern Recognition
  - Abstraction
  - Generalization
  - Algorithm Design
  - Evaluation

What is Informatics?

- "the collection, classification, storage, retrieval, and dissemination of recorded knowledge" – ***Merriam-Webster***
- "**Informatics** is the study of the structure, behaviour, and interactions of natural and engineered computational systems. **Informatics** studies the representation, processing, and communication of information in natural and engineered systems. It has computational, cognitive and social aspects." – ***University of Edinburgh***
- …
- …

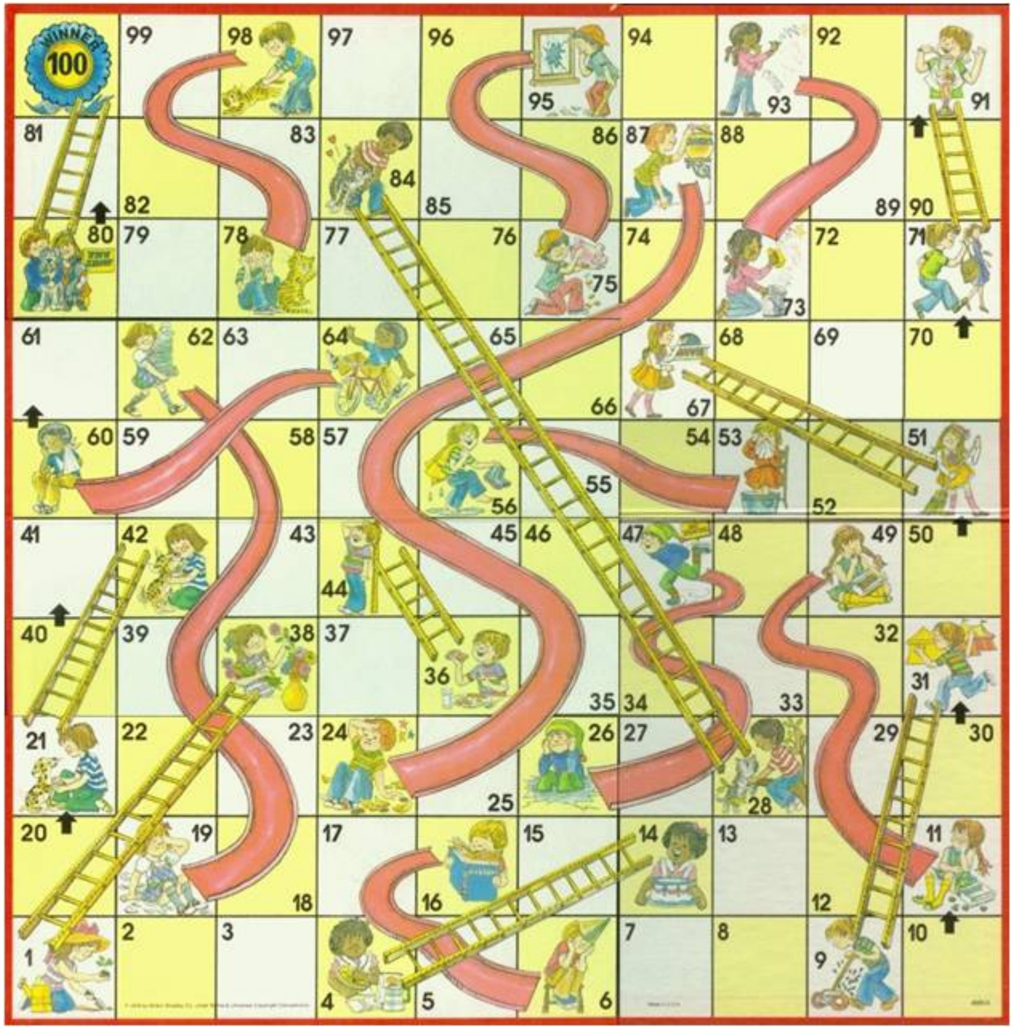
Informatics Minor @ UNL

- The Informatics minor is an interdisciplinary program that prepares students with core computational skill sets and competencies that allow them to solve problems within their chosen discipline or field.
- The program also builds interdisciplinary problem solving skills that are applicable and advantageous across academia and within industry.
- The minor's objectives are anchored around a set of core outcomes, such that students completing the minor will be able to:
  - Apply computational thinking to solve problems effectively and implement it using a programming language;
  - Apply statistical techniques to assess outcomes of empirical studies or experiments, and set up research designs to evaluate tools, techniques or hypotheses effectively;
  - Interact, use and manage data or databases and solve data-centric problems; or organize, visualize, and communicate digital data effectively and efficiently; or use creative competencies to generate creative solutions; and
  - Contribute one's expertise to the solution of interdisciplinary problems by effectively collaborating and communicating with those from other disciplines.

Chutes & Ladders

- If you have never played Chutes and Ladders, it's a simple game involving 1-4 players, a spinner with numbers 1-6 and a game board that is divided into 100 squares.
- On the board are squares that have ladders to advance you forward and chutes, or slides, to take you backwards.
- A player spins the spinner and moves the number of spaces indicated.
- If that square has a chute or ladder, the player must go to the square it leads to.
- A player must get to square 100 exactly, meaning if they go over 100, they must spin again.

Chutes & Ladders



http://www.rpi.edu/dept/eng/otherweb/GK12/indexb30c.html

Chutes & Ladders: Decomposition

- Components:
  - Spinner, Players, Chutes, Ladders
- How are all these components related?
  - Square (i.e., Location)
- How to move?
  - 1. spin, 2. if past square 100, stay put, 3. move player, 4. if chute move accordingly, and 5. if ladder move accordingly.
- How to determine a winner?
  - Check each player’s current square every time after a player has moved

Chutes & Ladders: Algorithm Design

- A loop:
  - Loop until there is a winner
    - 1. Player *i* spins
    - 2. Move player *i*
    - 3. Check winner
- Conditional for moving a player:
  - If current square + spin num > 100, no change to current square; otherwise, current square ← current square + spin num
  - If current square has a chute, then current square ← chute's bottom_square; otherwise, if current square has a ladder, then current square ← ladder's upper_square

Chutes & Ladders: Evaluation

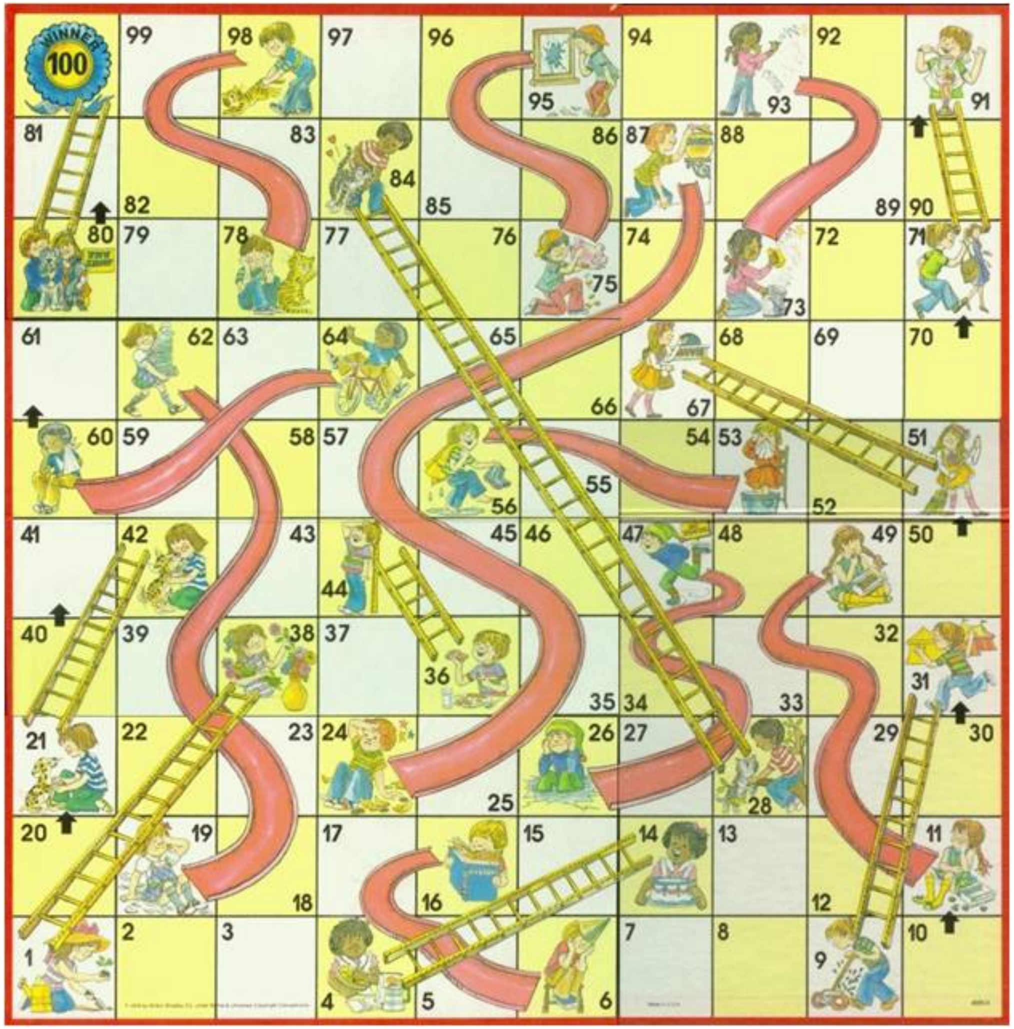
- Conditional for moving a player:
  - If current square + spin num > 100, no change to current square; otherwise, current square ← current square + spin num
  - If current square has a chute, then current square ← chute's bottom_square; otherwise, if current square has a ladder, then current square ← ladder's upper_square

Is there anything wrong with the above algorithm?

- *If current square + spin num > 100, will the second conditional still get executed?*

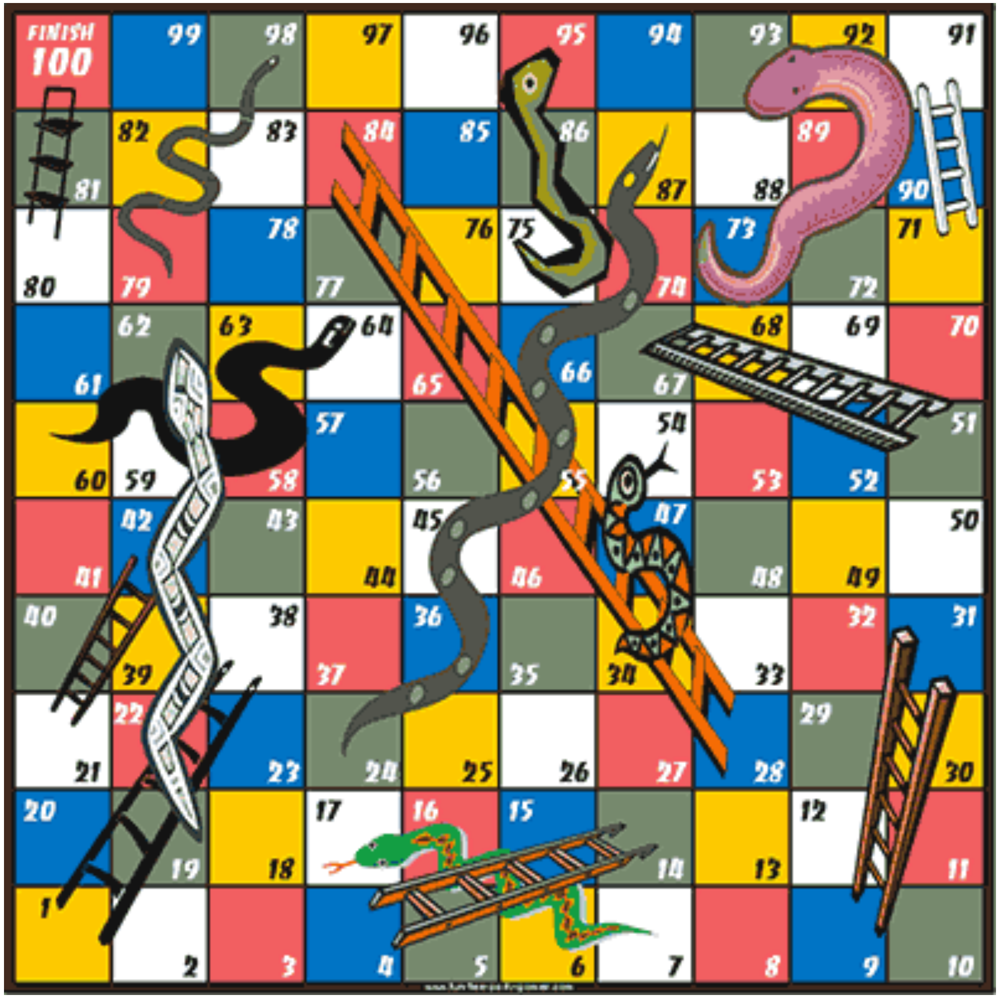
Chutes & Ladders: Abstraction

- Do we need to consider the colors of the squares?
- Do we need to consider the cartoons on the board?
- Do we need to consider the creases on the board?
- Do we need to consider the dimensions of the board?
- Etc.



Chutes & Ladders: Generalization

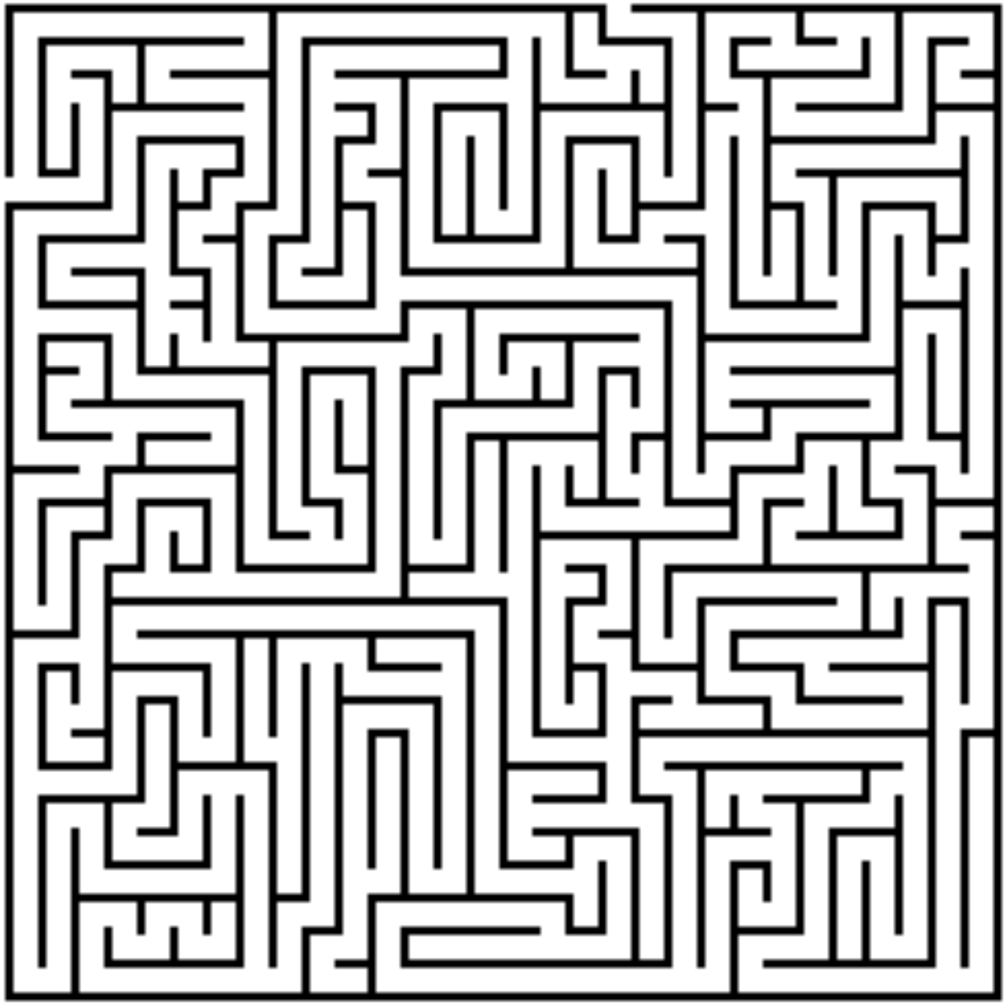
- Well, Snakes & Ladders game …
- Anything else?
- How about …



http://www.fun-free-party-games.com/traditional-games/traditional-board-games-snakes-and-ladders.html

Chutes & Ladders: Generalization 2

- How about …
  - Maze traversal?
    - Think about conditionals when selecting the next location to move to …



http://www.uefap.com/speaking/exercise/mazes/mazes.htm

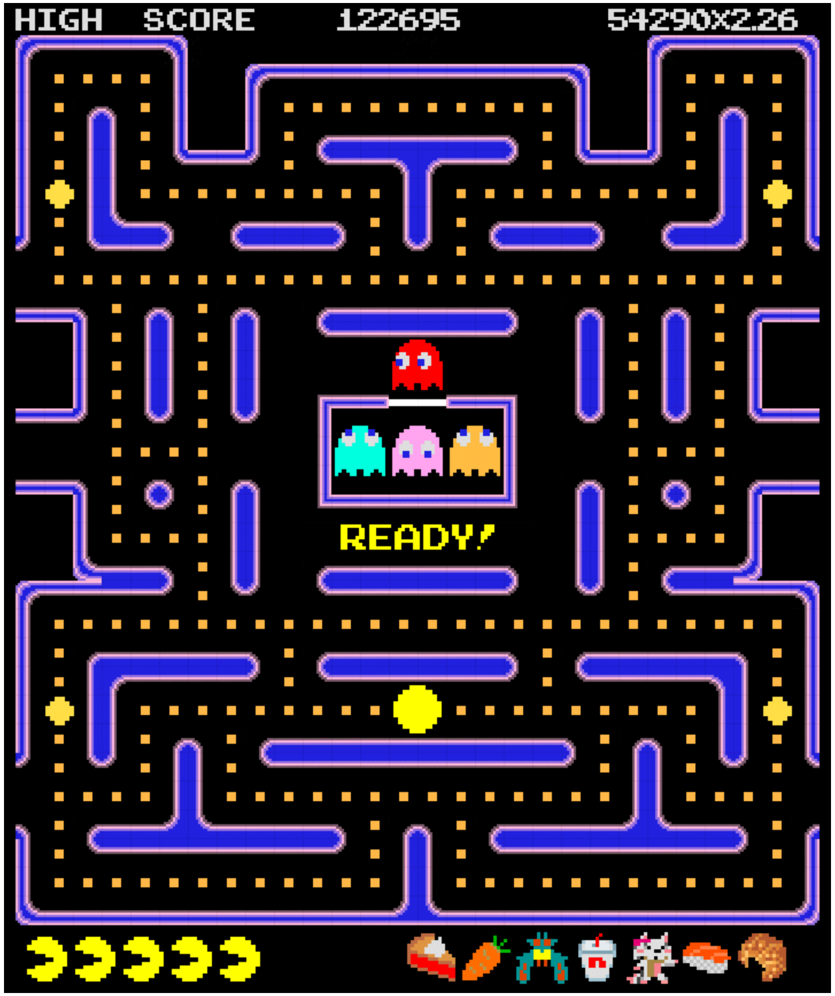
Chutes & Ladders: Generalization 3

- How about …
  - Maze traversal?
  - Monopoly?
    - Think about the chance factor
    - Think about the movement
    - Think about the get-out-of-jail card

http://legionofleia.com/2015/07/monopoly-movie-lionsgate-and-hasbro-team-up-to-pass-go/

Chutes & Ladders: Generalization 4

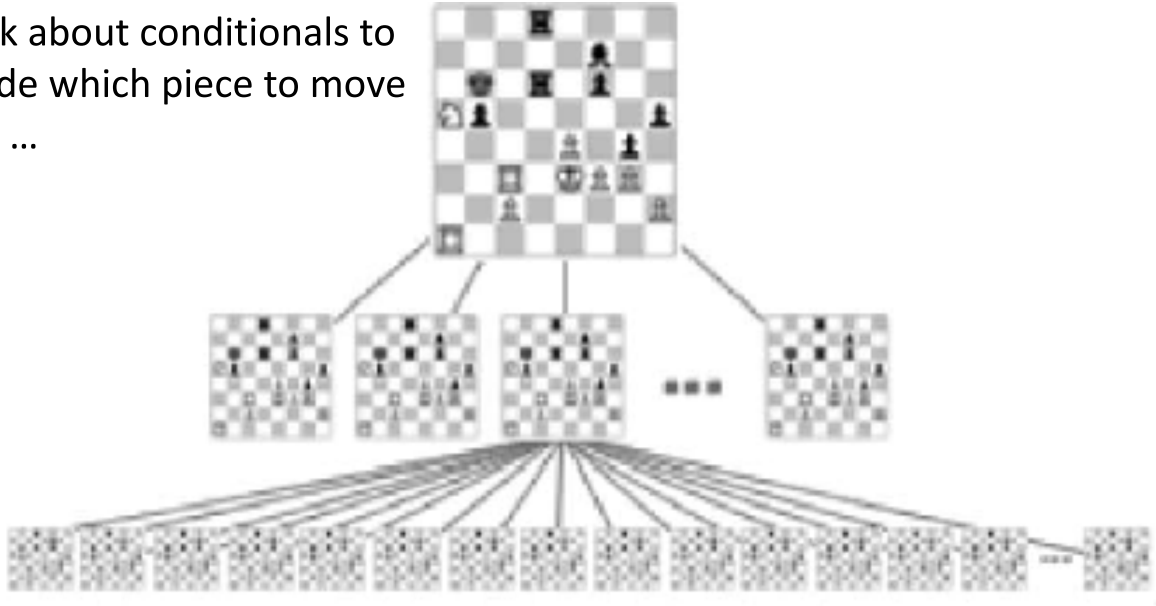
- How about …
  - Maze traversal?
  - Monopoly?
  - Pacman?
    - Think about getting eaten by the monsters …
    - Think about the monsters chasing the pacman ….
    - Thinking about selecting the next location to move to …



http://xmedia.ex.ac.uk/wp/wordpress/loading-new-game-i-hate-pac-man/

Chutes & Ladders: Generalization 5

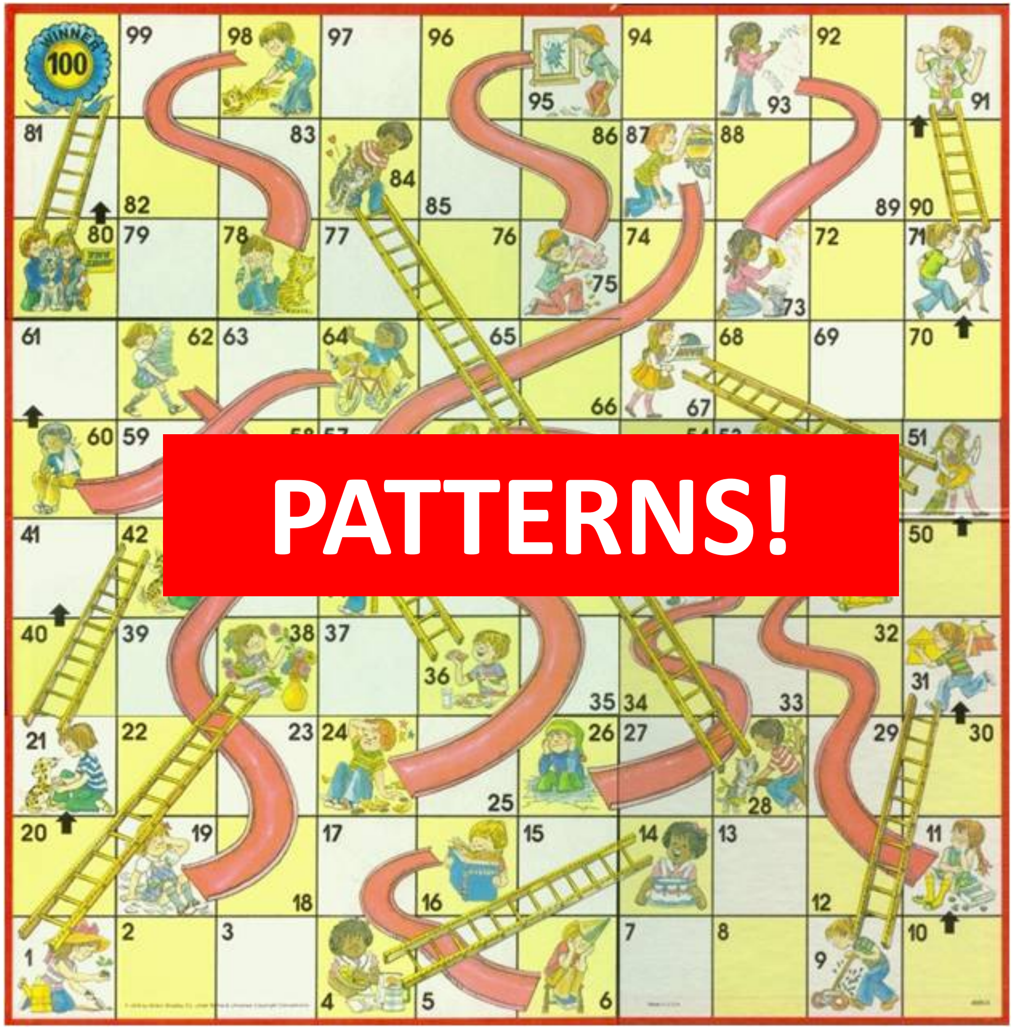
- How about …
  - Maze traversal?
  - Monopoly?
  - Pacman?
  - Chess playing???
    - Think about conditionals to decide which piece to move next …



http://stanford.edu/~cpiech/cs221/apps/deepBlue.html

Chutes & Ladders: Pattern Recognition

- What if now we wanted to allow players to generate new “configuration” of the game?
  - How would you make the game really boring?
  - How would you make the game really challenging?
  - How would you make the game really exciting?
  - …
  - Is it possible to make the game impossible for any player to win?



PATTERNS!

References

- http://www.google.com/edu/computational-thinking
- Paul Curzon’s “So What is Computational Thinking”
- Jeannette M. Wing’s “Computational Thinking”, *Communications of the ACM*, March 2006, pp. 33-35
- https://en.wikipedia.org/wiki/Computer_science