Today

- Handouts
- Introduction to course project
  - Play against my AI
  - Play Hop Step on iPads against each other
  - Discuss the game and its properties
- State Spaces
  - How do we represent states?
  - How should we implement Hop Step?

State Spaces

- What is a state?
  - What does it contain?
  - How many of them are in a game?

Analyze tic-tac-toe

- How many states are possible?
- Several ways to calculate:
  - 9 locations, each of which can have 3 values
    - 19,683 possibilities
    - But, the number of x’s and o’s should be similar!
  - 9! (362,880) ways to play a game
  - Only 138 ways a game can end
    - Symmetry! (eg Only 3 first moves)
Analyze Nim

- Start with a stack of N items
  - Each person removes either 1 or 2 items
  - The first person who can't remove an item loses
- (Play a game between two students)
- How many states?
  - Only 2*N possible states (why not 2*N+2?)
- How many outcomes?
  - Just 2

Scaling it up?

- What happens if we increased the board size?
  - Tic-tac-toe
    - Number of legal moves would greatly increase
    - Number of states would increase
  - Nim
    - Number of legal moves stays the same
    - Number of states increases
    - Difference between placing pieces and moving pieces

First-level analysis

- How would we build a general model?
  - Assume that there are b actions (always the same)
  - Assume we are searching to depth d
  - \( b^d \) states! (Growing exponentially)
  - Similar implications and algorithms as used in single-agent search

How large are some common games?

- Connect Four has \( 10^{13} \)
- Checkers has \( 10^{20} \) states; \( 10^{18} \) reachable
- Chess has about \( 10^{47} \) states
- Go has about \( 10^{171} \)
What does this imply?

- If Deep Blue is looking at 300,000,000 states / sec
  - Can’t be analyzing anything close to the whole game
- How much memory does our machine have?
  - Can’t be storing all these states in memory
  - May want/need an efficient state representation
  - (Allocating memory is very slow)

State Model

- A state is an abstract representation of a game
- A state should provide:
  - A successor function:
    - Get legal moves or legal successor states?
    - Generally use depth-first algorithms and only keep 1 copy of the full game state in memory
    - Moves can often be represented efficiently

Why do we care about this?

- Ken Thompson showed a strong correlation between depth of search and playing strength
  - That’s why Deep Blue was engineered to explore 300 million positions per second
- How do you improve depth of search?
  - Make your program faster

State Model

- A state should provide:
  - Functions to apply and undo actions
  - A hash function
    - Will discuss in more depth later
  - Information about whose move it is
  - A test to see if the game is over
  - Information about who won
  - [Ability to copy a state]