the gamedesigninitiative at cornell university

Lecture 15:

Artificial Intelligence

Assignment 9 due tonight

- Submit through CMS

Second Prototype Due

- Thursday, October 23rd
- **In-class playtesting**

Logging release later today

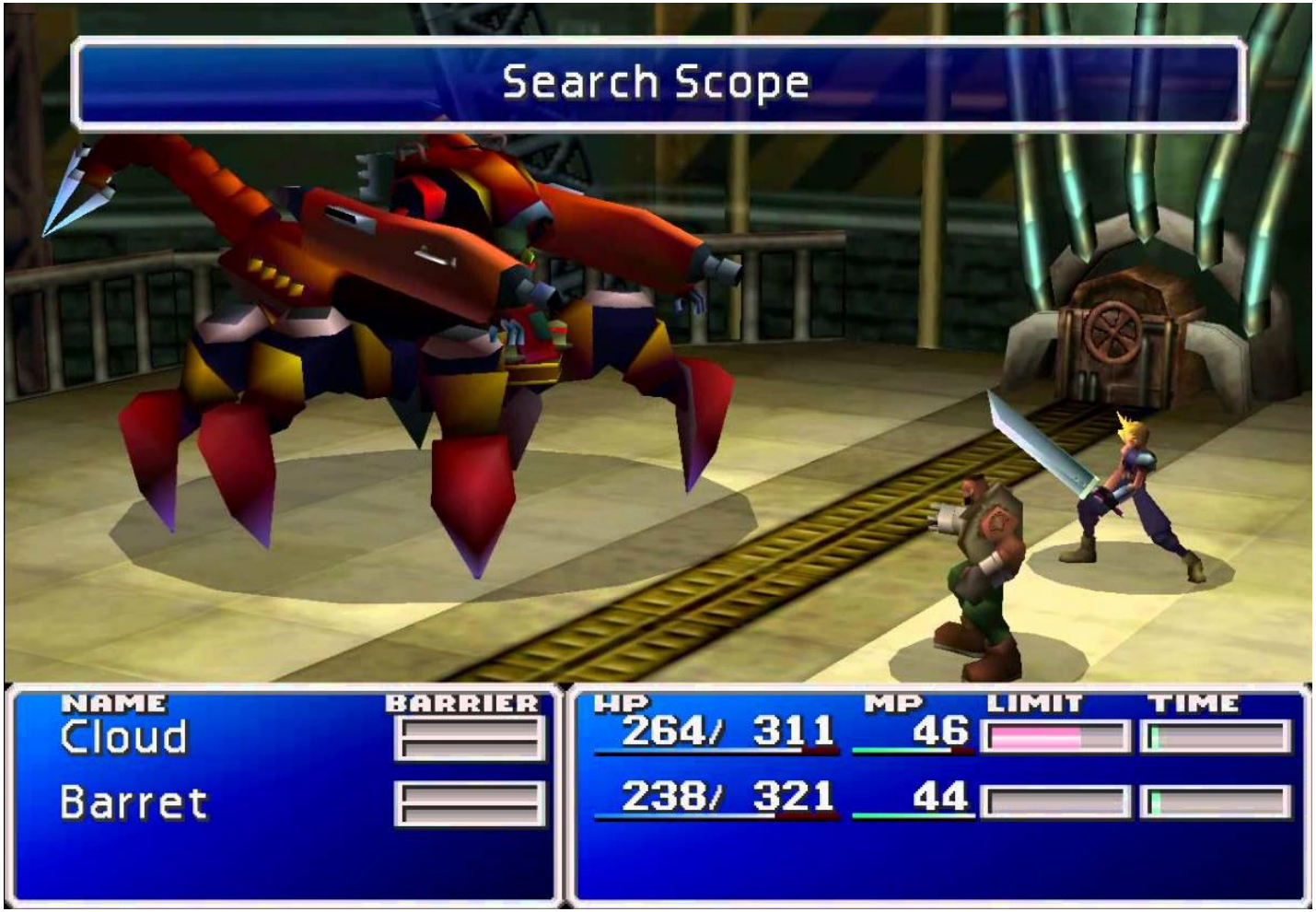
- Will send out an email
- Needed for first release (Nov. 4th)

- Scripts
- Planning
- Pathfinding

Scripts

1. **IF** <condition is true> **THEN** <perform action>
2. **IF** <condition is true> **THEN** <perform action>
3. **IF** <condition is true> **THEN** <perform action>

First boss in Final Fantasy VII

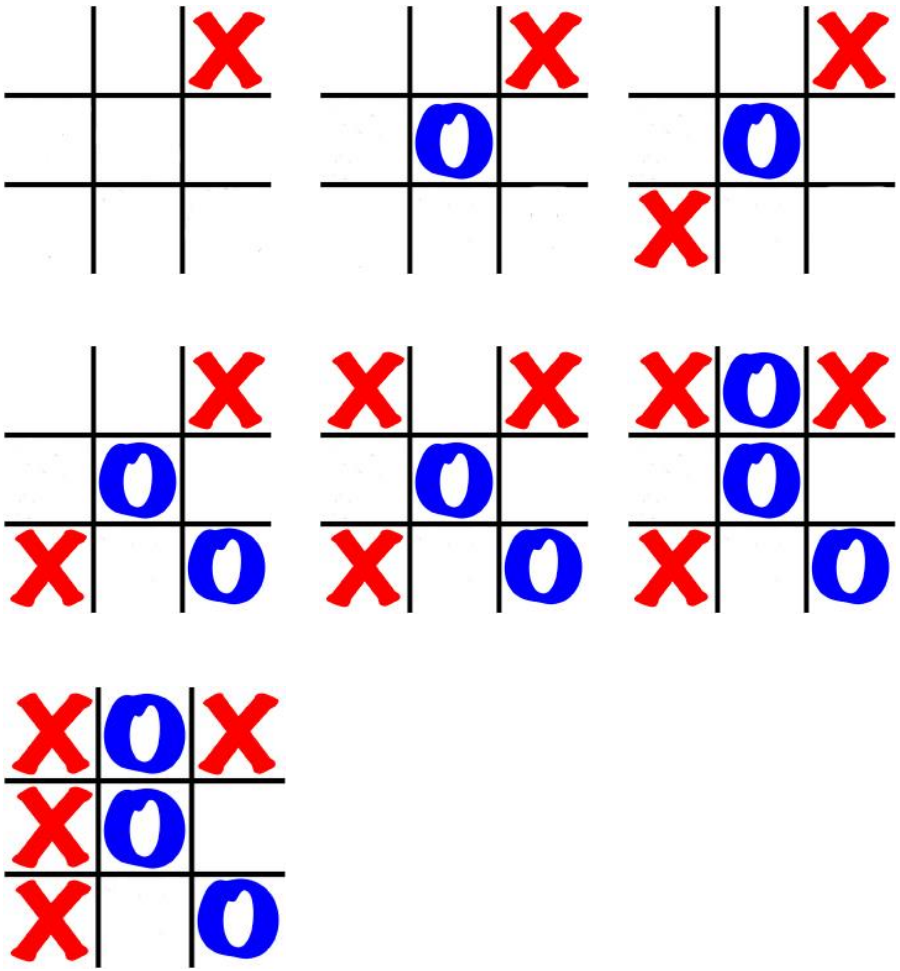


First boss in Final Fantasy VII

- **If (Count == 0 OR Count == 2) Then**
  - SelectedTarget = random opponent
  - Use Search Scope on SelectedTarget
  - Count = Count + 1
- **If (Count == 1 OR Count == 3) Then**
  - With probability 2/3:
    - If Self HP < (Self Max HP / 2) Then
      - Use Scorpion Tail on SelectedTarget
    - Else
      - Use Rifle on SelectedTarget
  - With 1/3 Chance:
    - Use Scorpion Tail on SelectedTarget
  - Count = Count + 1

Example: Tic Tac Toe

1. If I can win, win
2. If opponent can win, prevent it
3. If center is available, take it
4. If corner is available, take it



Advantages/Disadvantages

- ➕ Easy to implement
- ➕ Can express complex behaviors
- ➖ "Smart" behavior can be ***very*** complex
- ➖ Not so scalable
- ➖ No natural way to vary difficulty

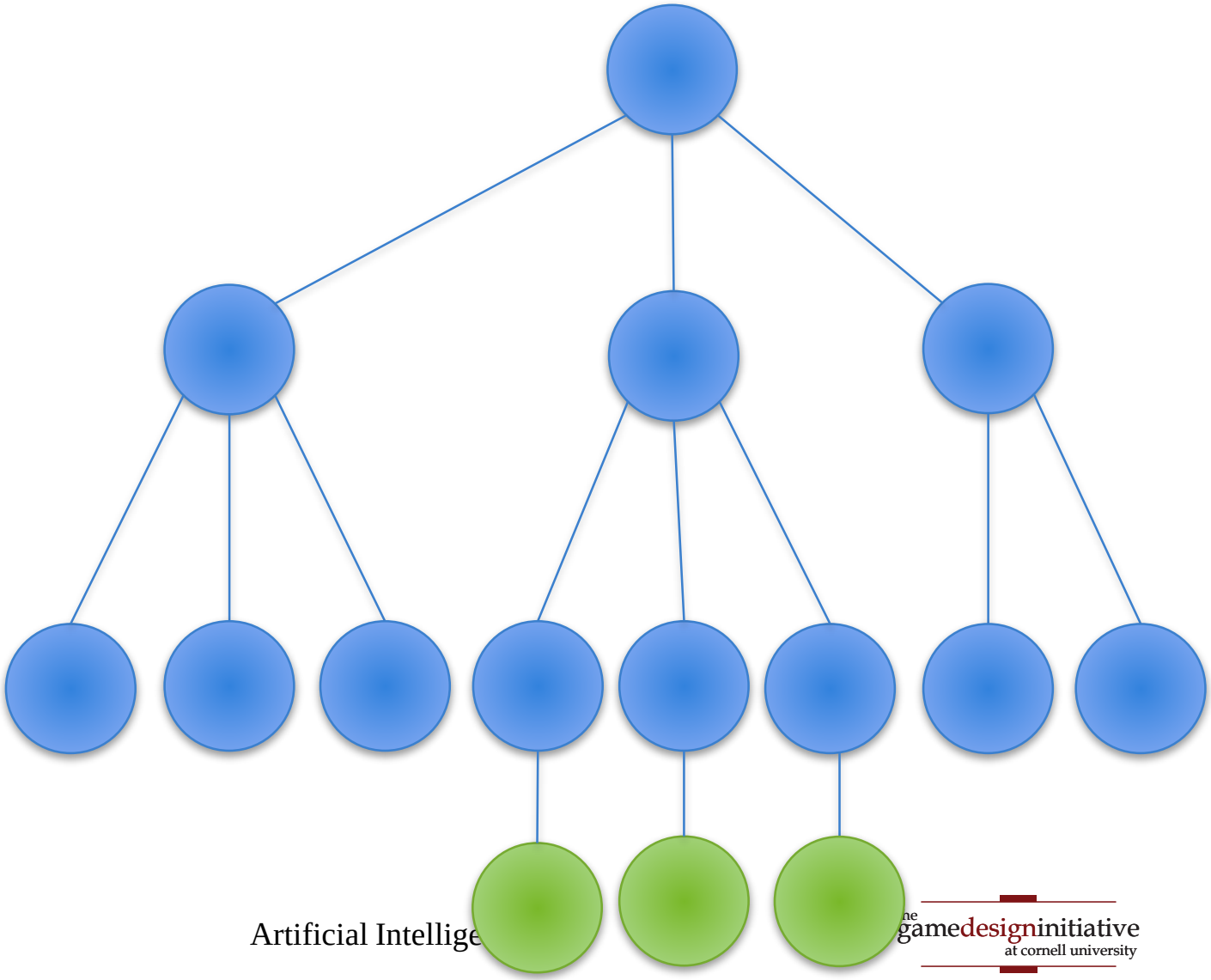
Adversarial Search

Now

I move

Opponent moves

I move



Importance of search depth in Chess

- Novice: ~4
- Master: ~8
- Grandmaster: ~12
- Deep Blue: 6 – 40

Advantages/Disadvantages

- ➕ Potentially ***much*** smarter
- ➕ Natural way to vary difficulty
- ➖ State-space explosion
- ➖ Game must have certain properties
- ➖ Unclear what to do if you can't "see" the end

Heuristics: Chess

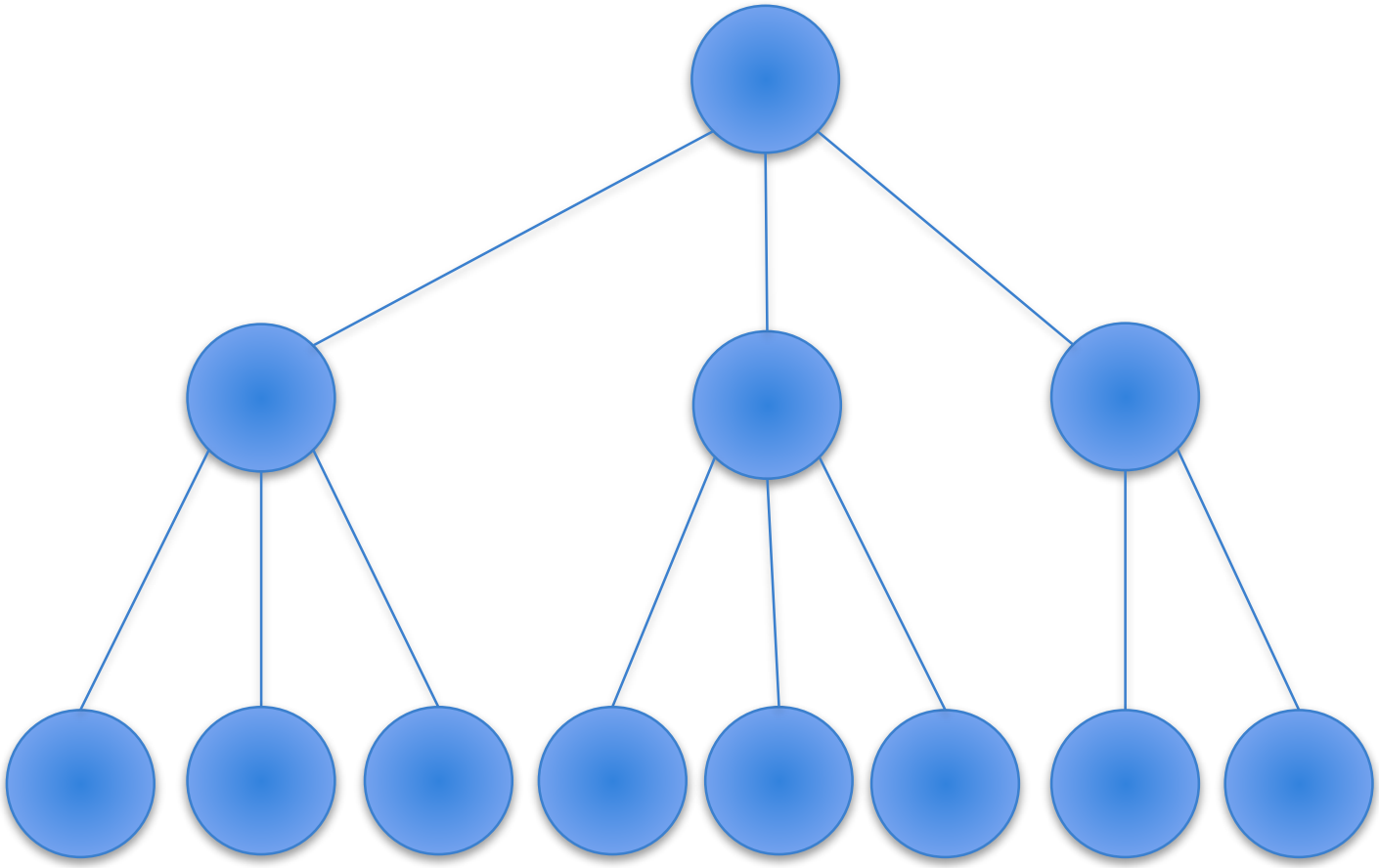
- Pawn: 1 point
- Knight & Bishop: 3 points
- Rook: 5 points
- Queen: 11 points

Heuristic Planning

Now

I move

Opponent moves

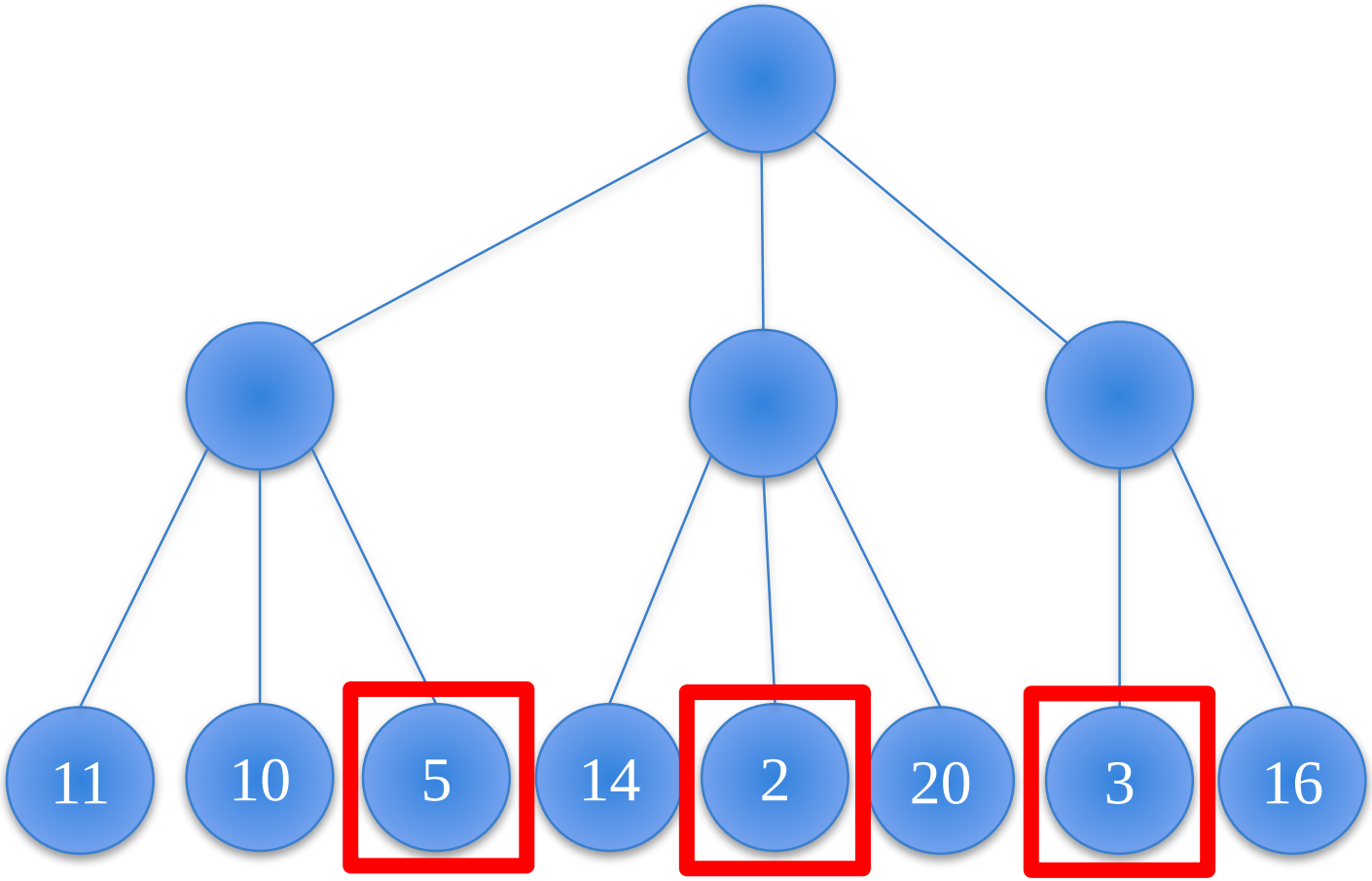


Heuristic Planning

Now

I move

Opponent moves

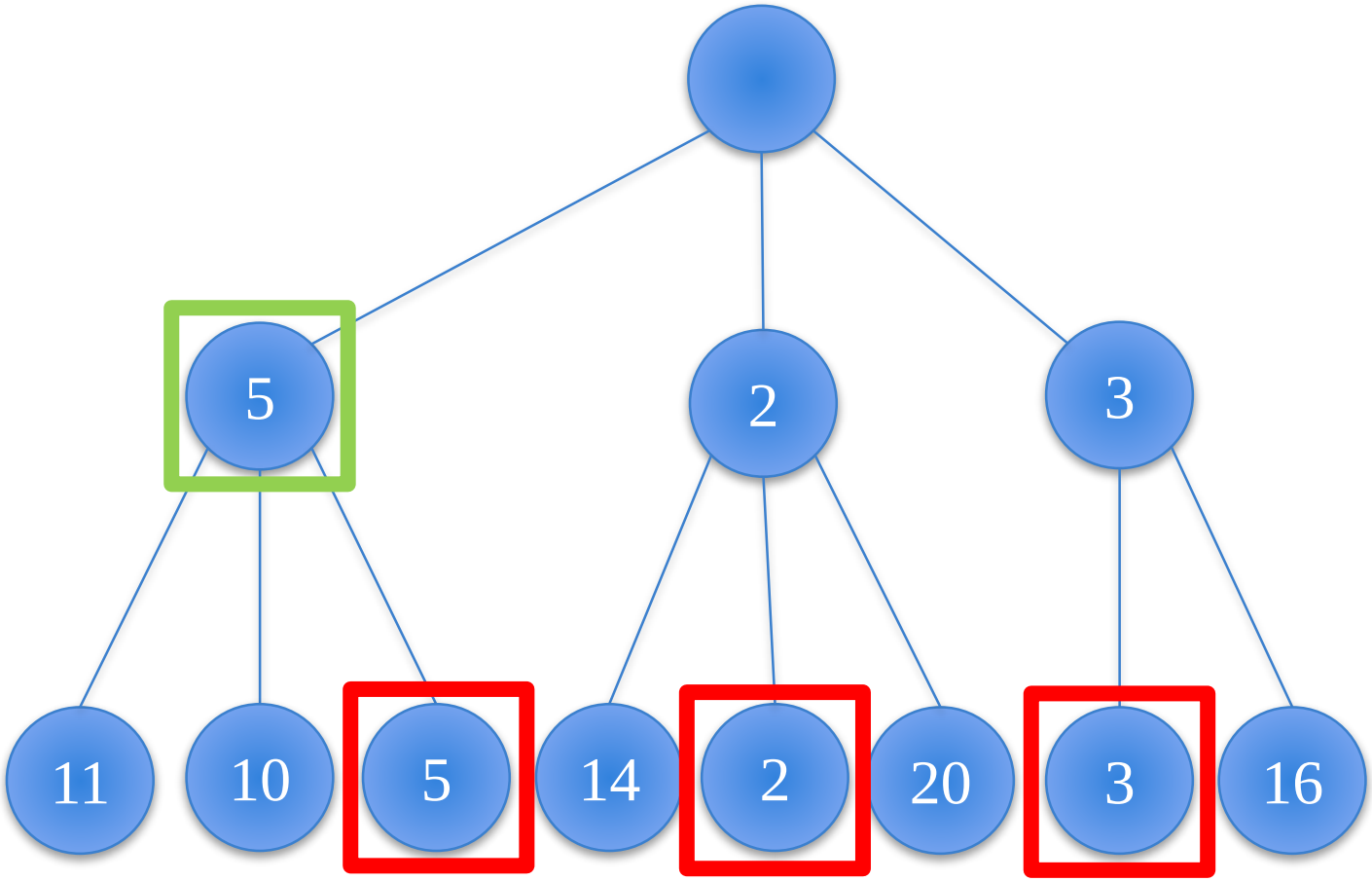


Heuristic Planning

Now

I move

Opponent moves

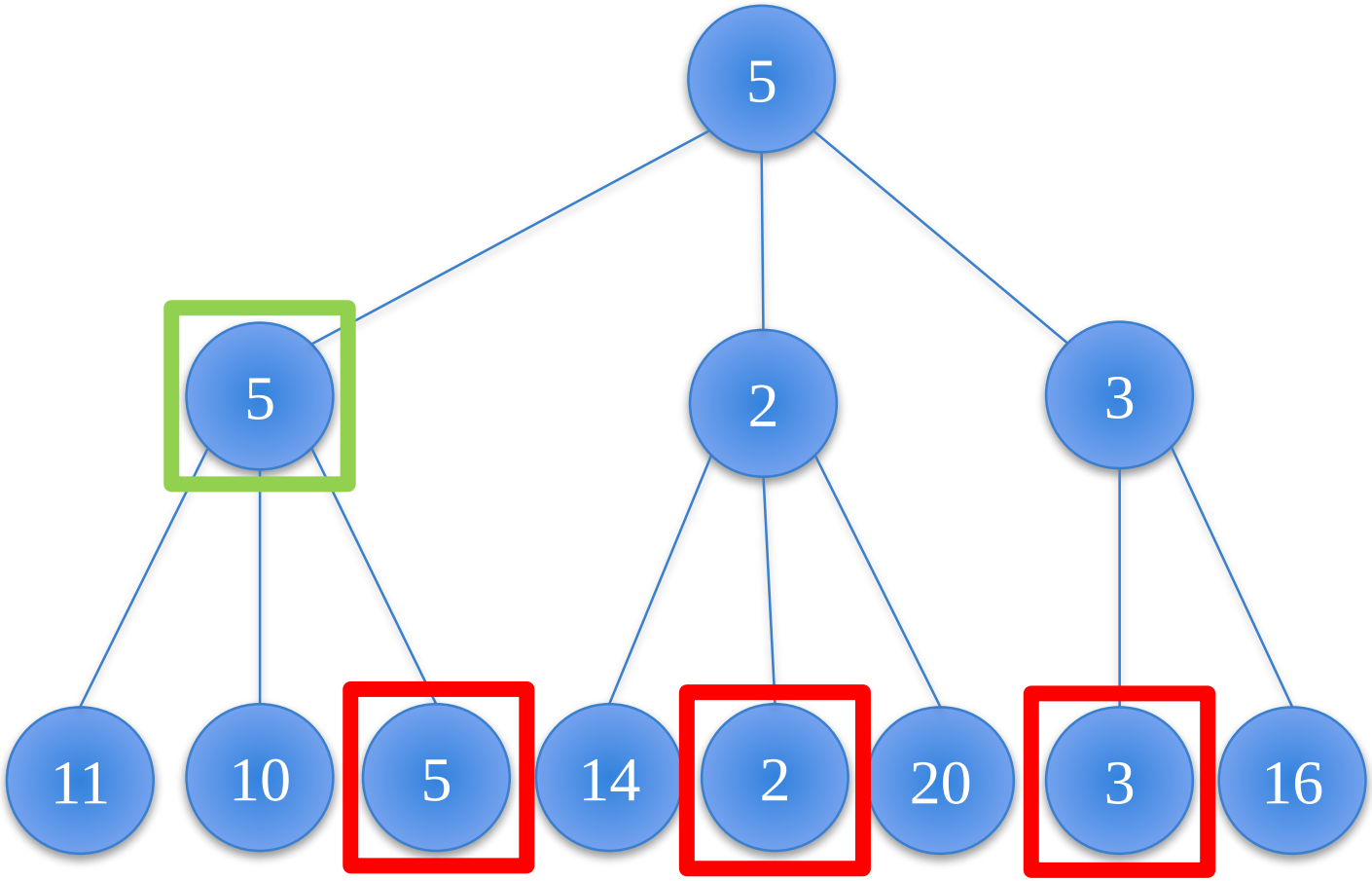


Heuristic Planning

Now

I move

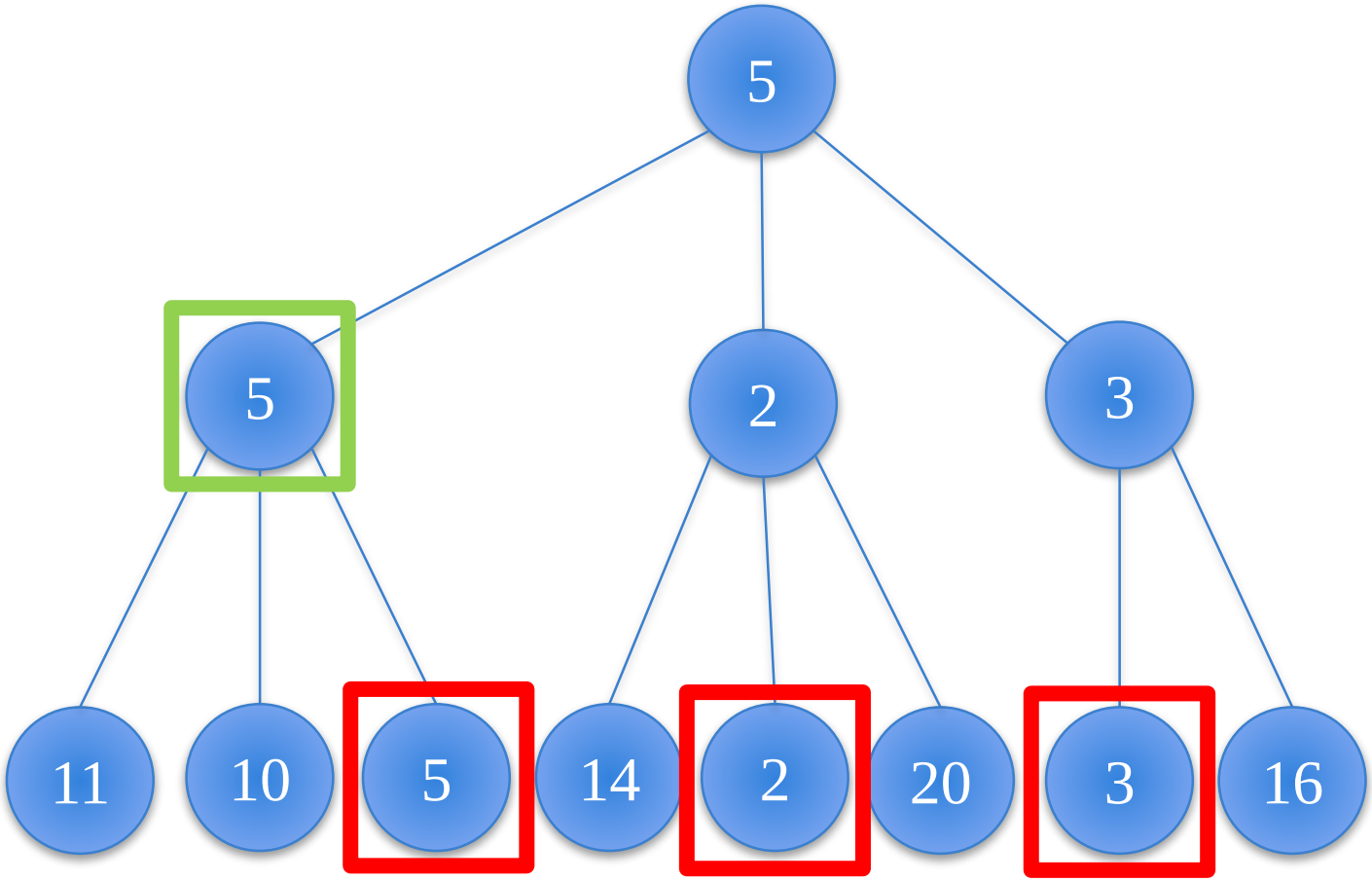
Opponent moves



Minimax

MAX

MIN



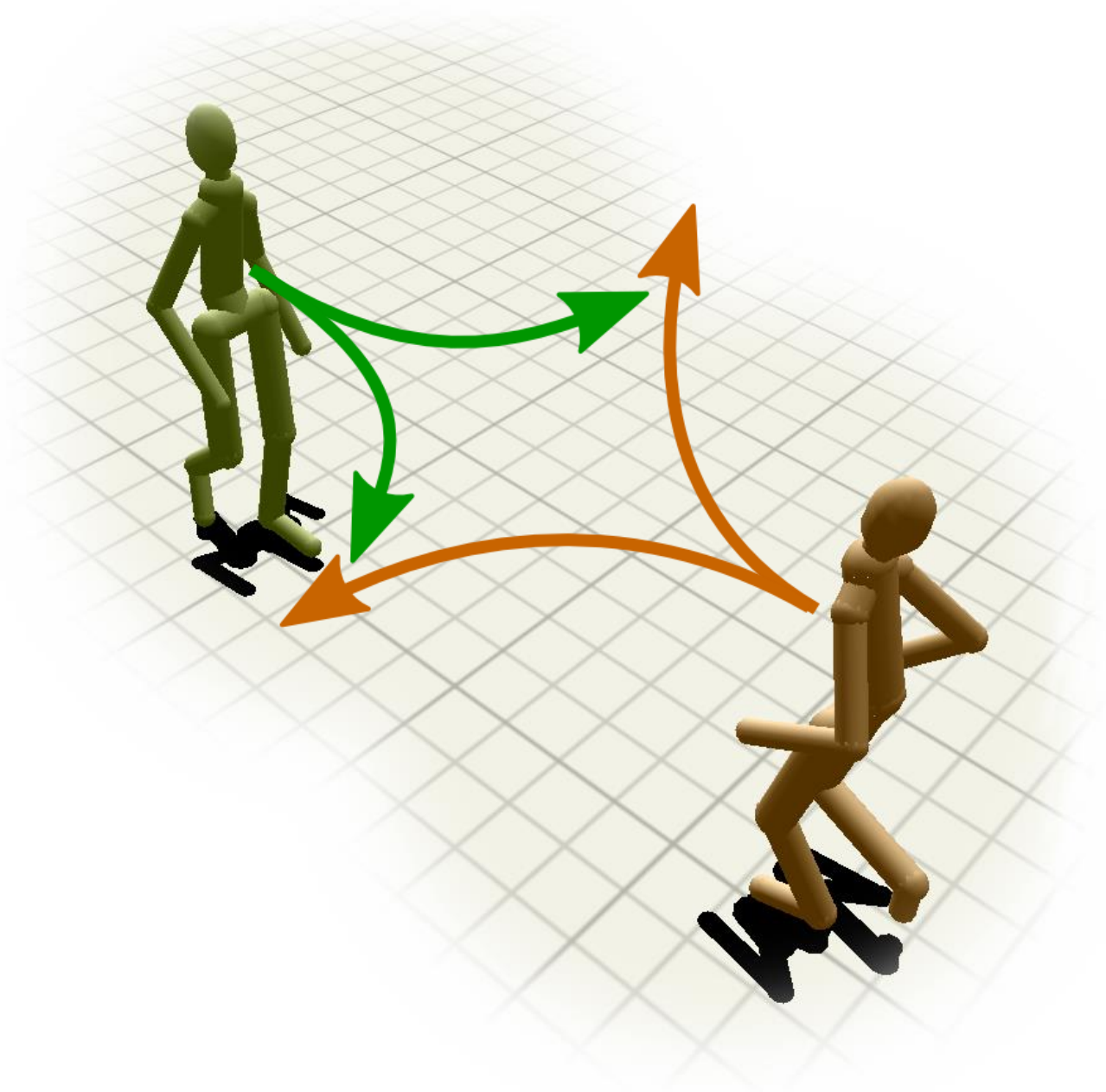
Advantages/Disadvantages

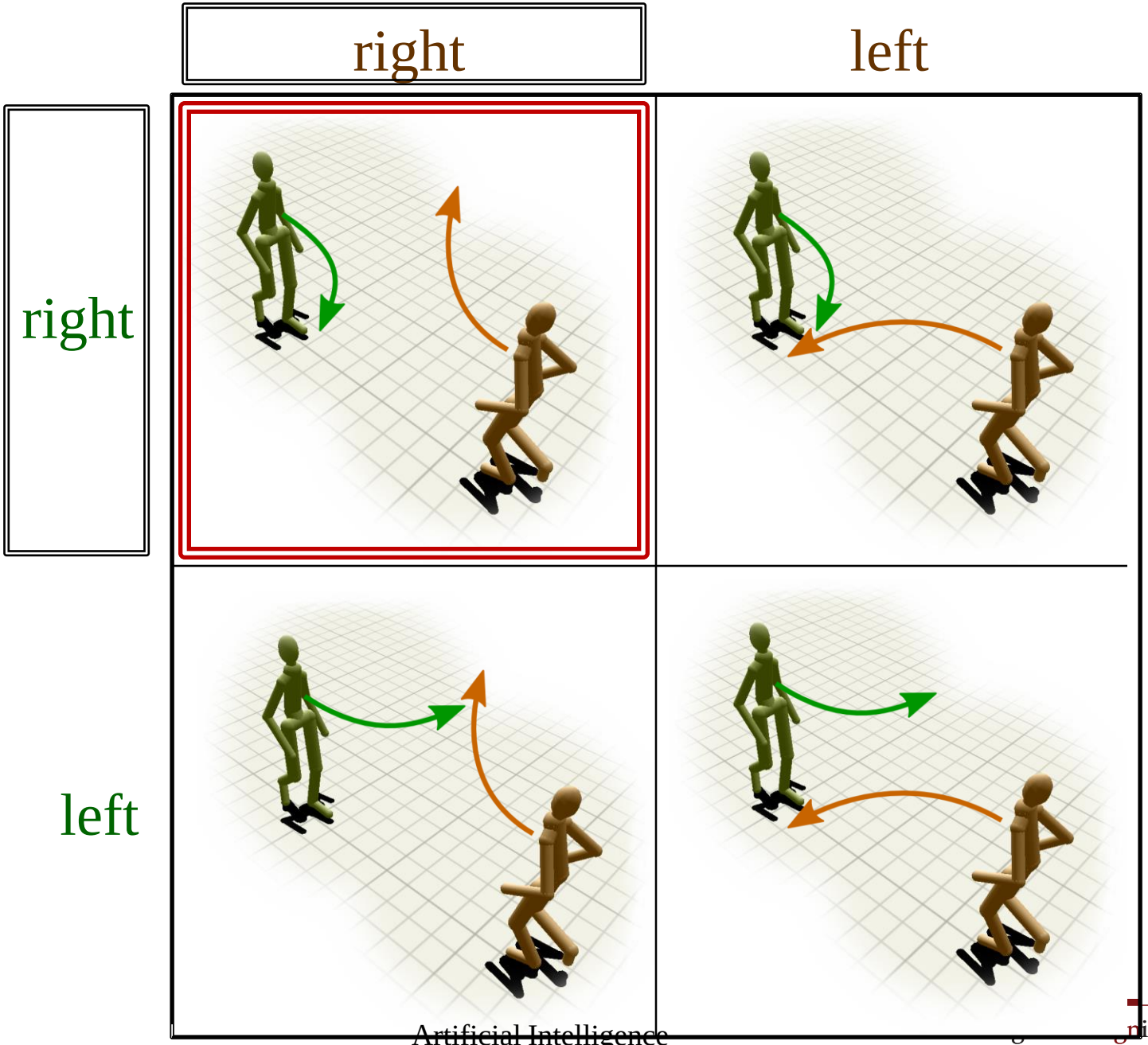
- ➕ Scalable
- ➕ Can be rational without “seeing” the endgame
- ➖ Strength depends a lot on the heuristic
- ➖ Still only works for some games

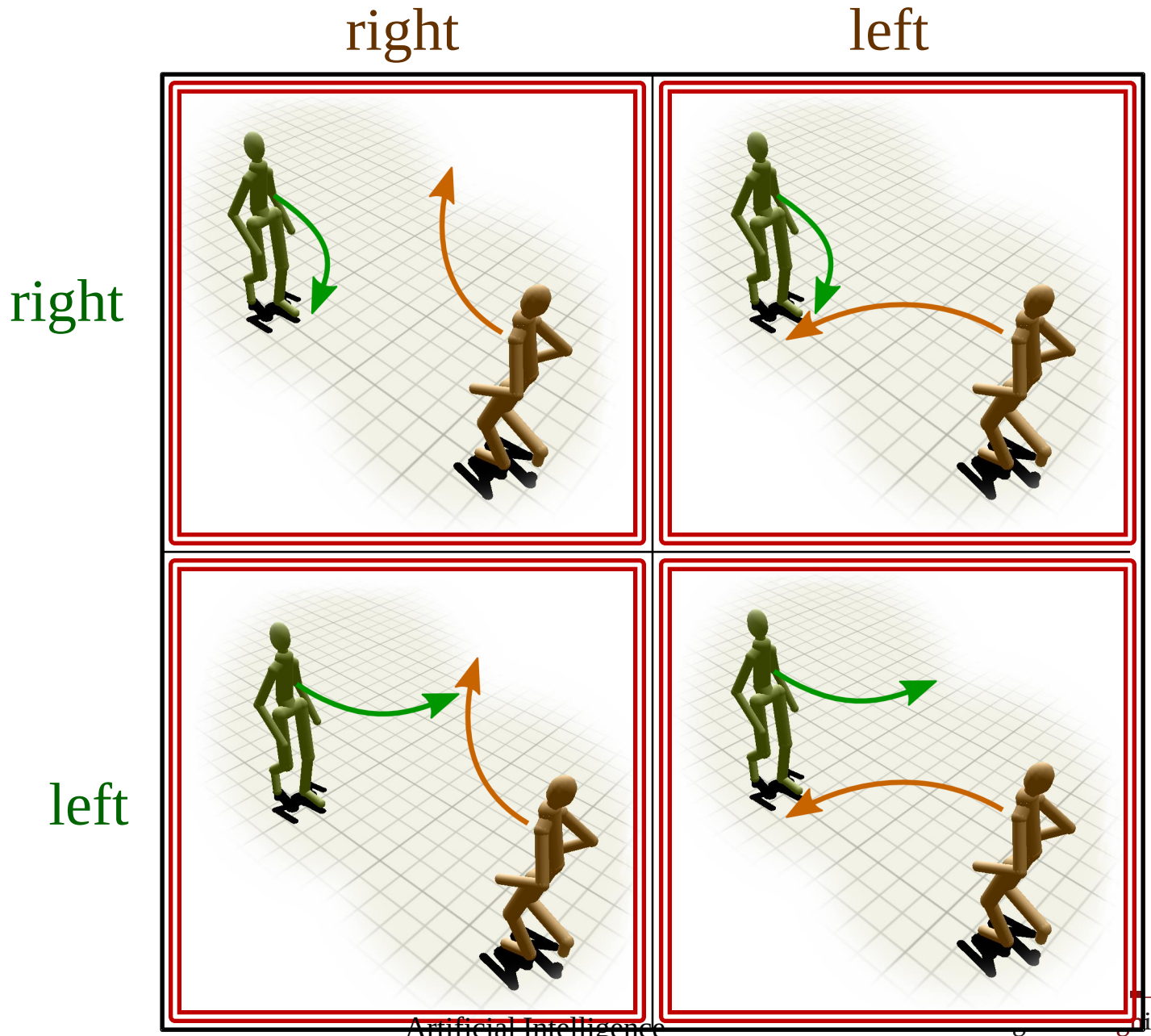

1105
900
170/184
Click: Select Unit
Shift+Click: Deselect Unit
Ctrl+Click: Select Unit Type
MENU

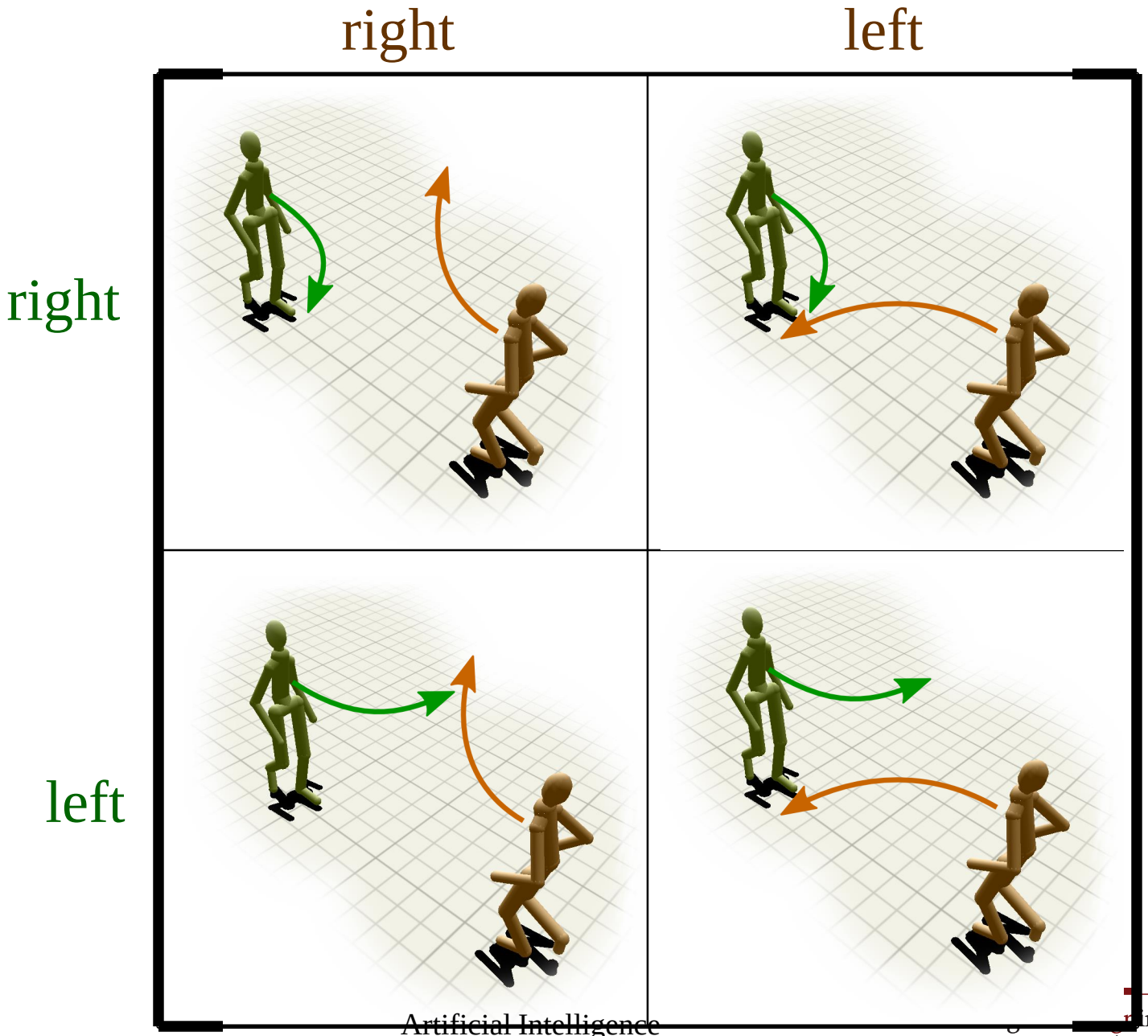
Simultaneous Actions

$$\frac{1}{3} \qquad \frac{1}{3} \qquad \frac{1}{3}$$




right
left
right
left


right
left
right
left


right
left
right
left

	right	left
right	1	-1
left	-1	1

	right	left
right	1	-1
left	-1	1

What Is an Optimal Policy?

Agent Policy

$$\begin{bmatrix} \frac{2}{3} \\ \frac{1}{3} \end{bmatrix} \begin{matrix} \text{right} \\ \text{left} \end{matrix}$$

What Is an Optimal Policy?

$$\begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad \begin{bmatrix} 1 & \begin{bmatrix} \frac{1}{3} & -1 \\ -\frac{1}{3} & 1 \end{bmatrix} \\ -1 & \end{bmatrix} \cdot \begin{bmatrix} \frac{2}{3} \\ \frac{1}{3} \end{bmatrix}$$

What Is an Optimal Policy?

Agent Policy

$$\begin{bmatrix} \frac{2}{3} \\ \frac{1}{3} \end{bmatrix}$$

Opponent Response

$$\begin{bmatrix} 0 & 1 \end{bmatrix} \longrightarrow -\frac{1}{3}$$

Expected Reward

What Is an Optimal Policy?

$$\begin{bmatrix} \frac{2}{3} \\ \frac{1}{3} \end{bmatrix} \xrightarrow{\begin{bmatrix} 0 & 1 \end{bmatrix}} \boxed{-\frac{1}{3}}$$

$$\begin{bmatrix} 0 \\ 1 \end{bmatrix} \xrightarrow{\begin{bmatrix} 1 & 0 \end{bmatrix}} \boxed{-1}$$

$$\begin{bmatrix} \frac{4}{5} \\ \frac{1}{5} \end{bmatrix} \xrightarrow{\begin{bmatrix} 0 & 1 \end{bmatrix}} \boxed{-\frac{3}{5}}$$

$$\begin{bmatrix} \frac{1}{4} \\ \frac{3}{4} \end{bmatrix} \xrightarrow{\begin{bmatrix} 1 & 0 \end{bmatrix}} \boxed{-\frac{1}{2}}$$

$$\begin{bmatrix} 1 \\ 0 \end{bmatrix} \xrightarrow{\begin{bmatrix} 0 & 1 \end{bmatrix}} \boxed{-1}$$

$$\begin{bmatrix} \frac{7}{8} \\ \frac{1}{8} \end{bmatrix} \xrightarrow{\begin{bmatrix} 0 & 1 \end{bmatrix}} \boxed{-\frac{3}{4}}$$

What Is an Optimal Policy?

$$\begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \end{bmatrix} \longrightarrow 0$$

$$\begin{bmatrix} \frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \end{bmatrix}$$

Agent's Expected Reward $= 0$

Expected Reward

$$v = \pi_o{}^T \boldsymbol{R} \pi_a$$

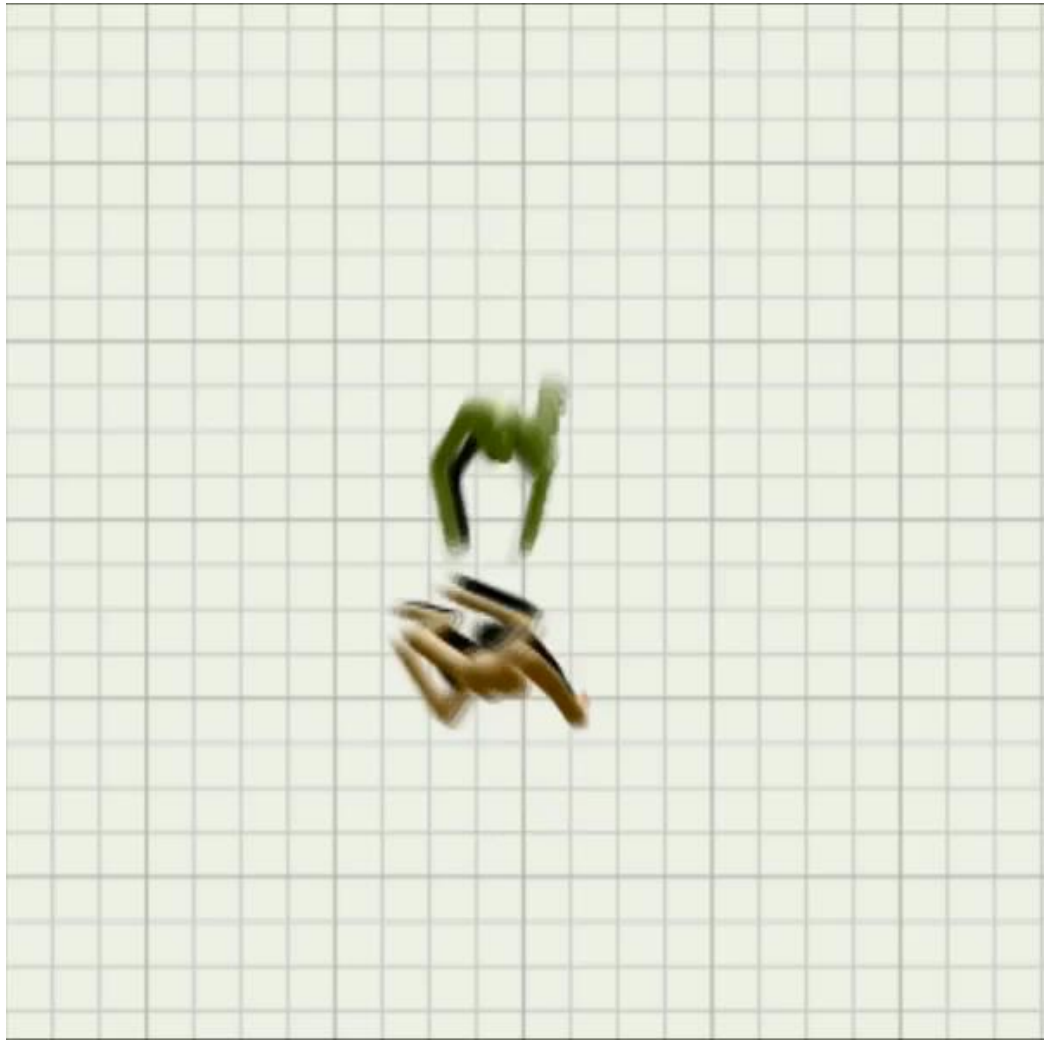
Calculating Reward

$$\max_{v,\pi} v \text{ such that}$$

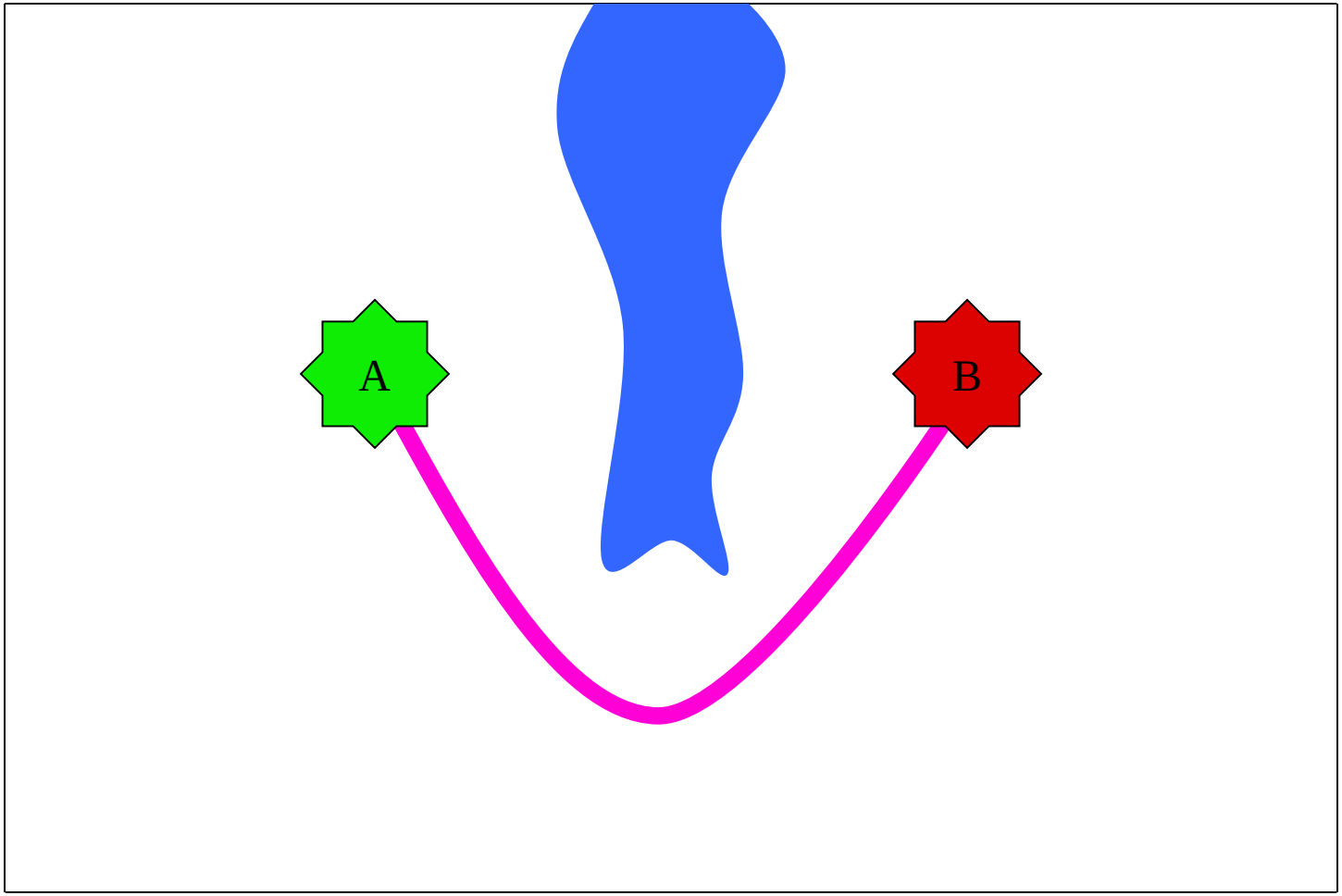
$$\sum_i \pi_i = 1$$

$$\pi \geq \mathbf{0}$$

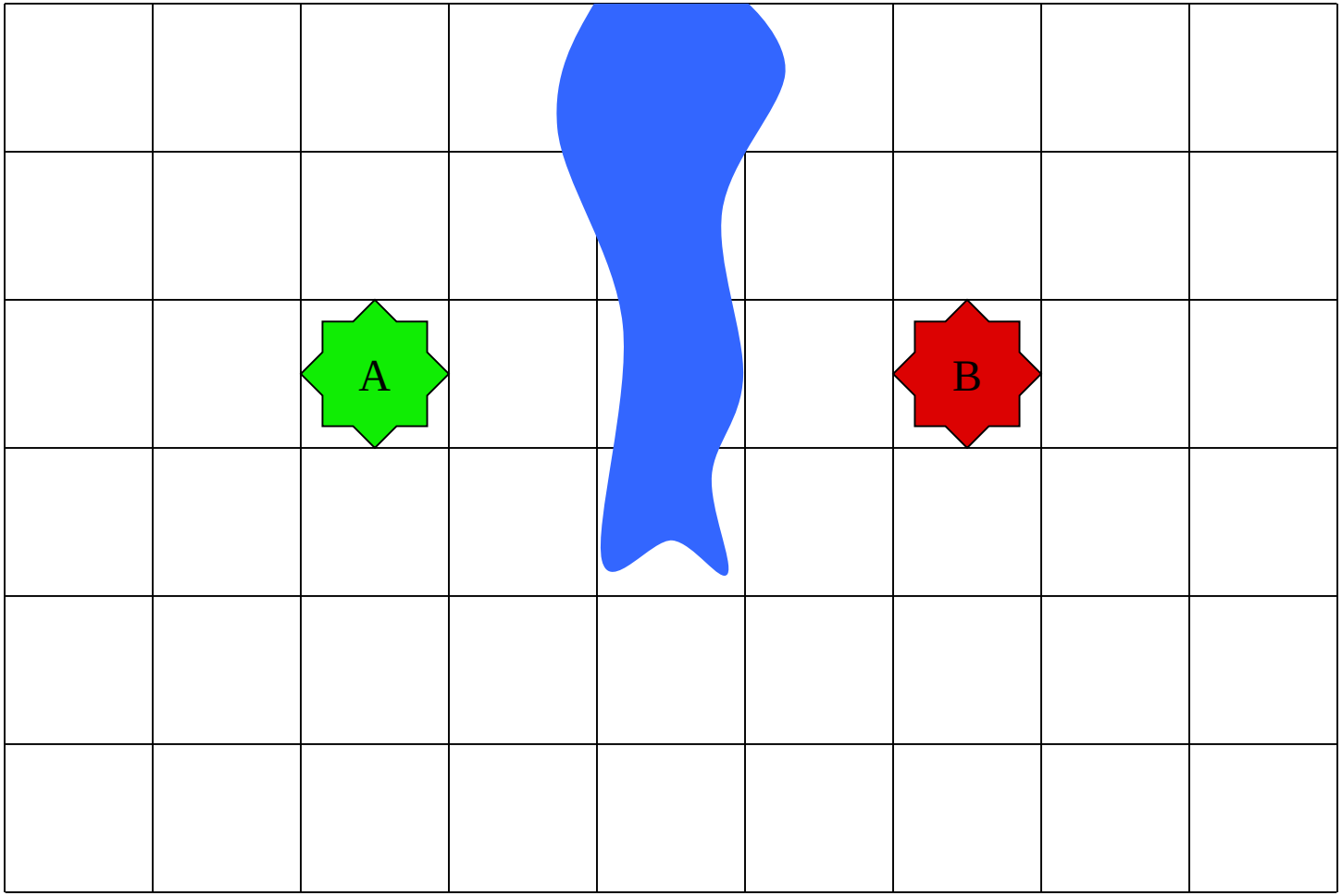
$$v \leq \boldsymbol{R}\,\pi$$



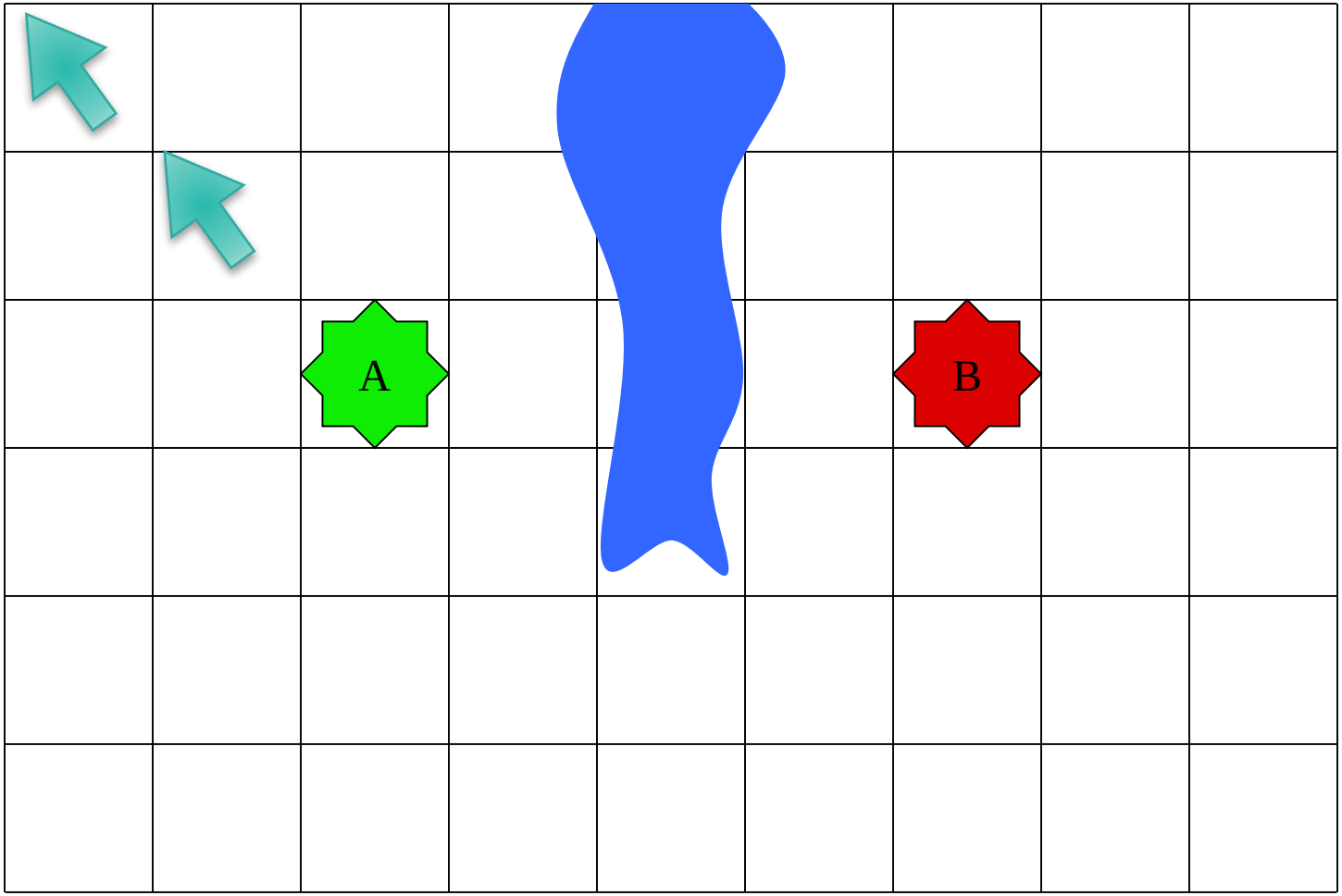
Pathfinding



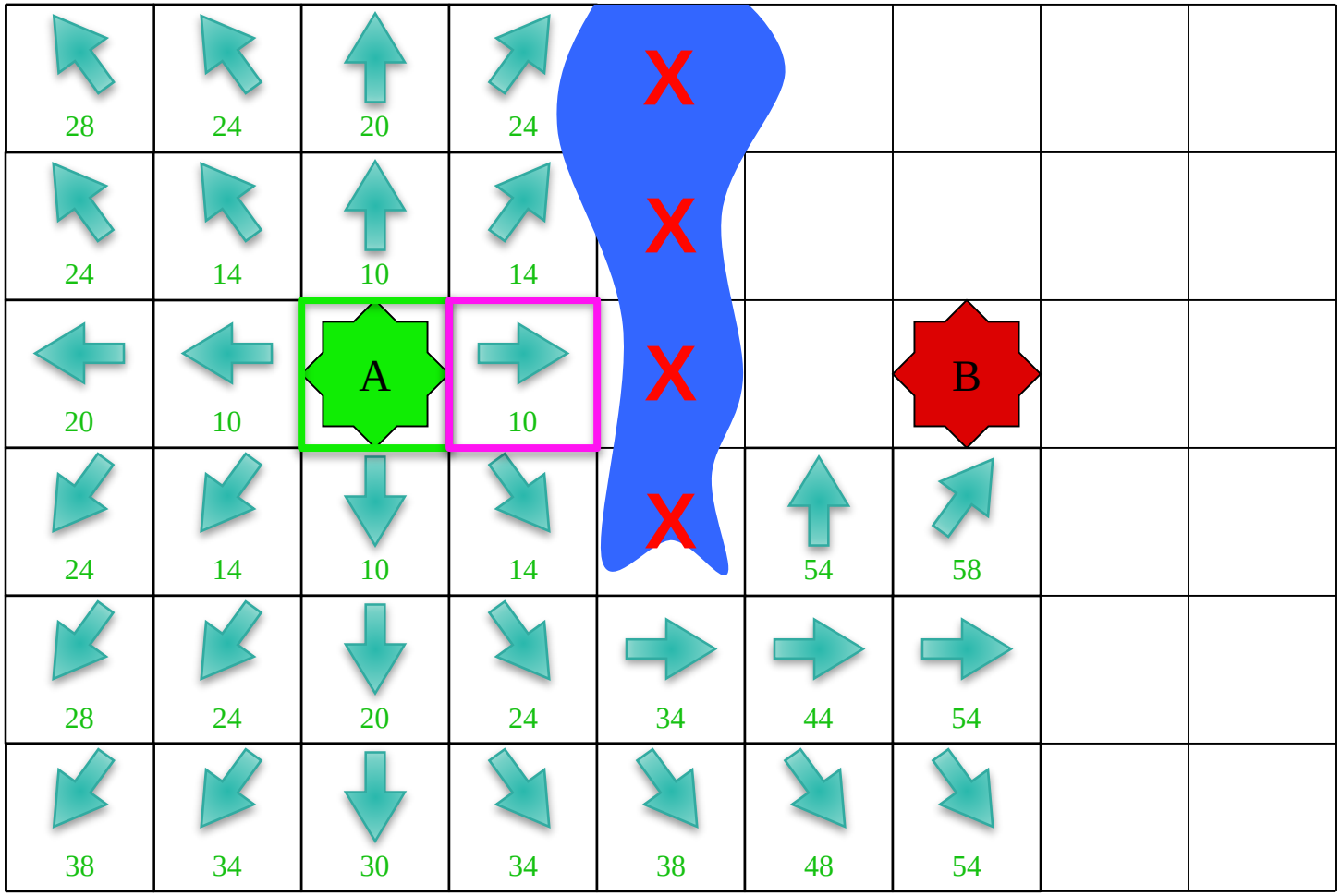
Make a grid!



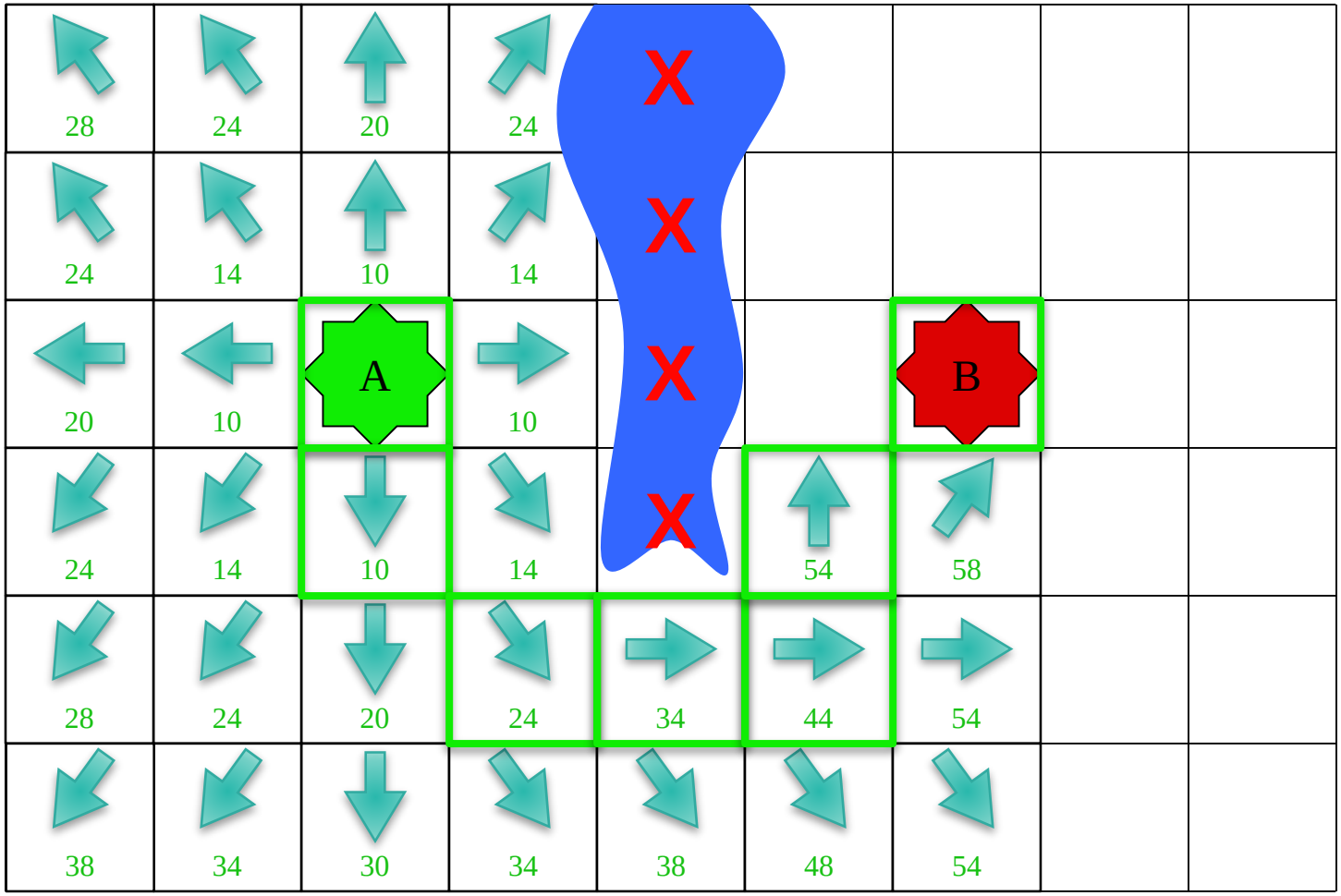
Pathfinding: Depth-First



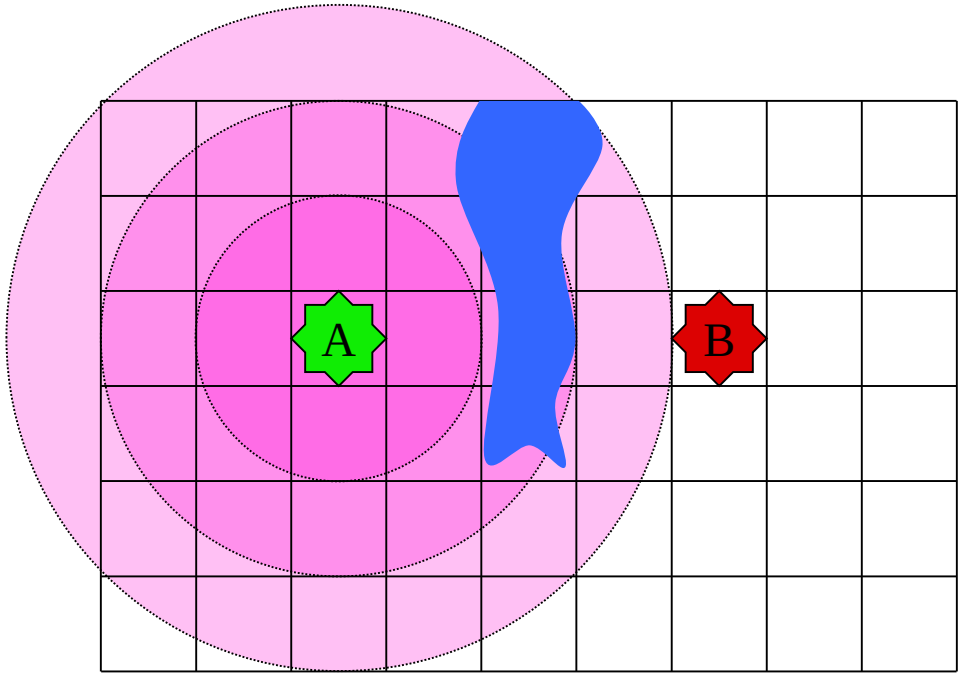
Pathfinding: Breadth-First



Pathfinding: Breadth-First



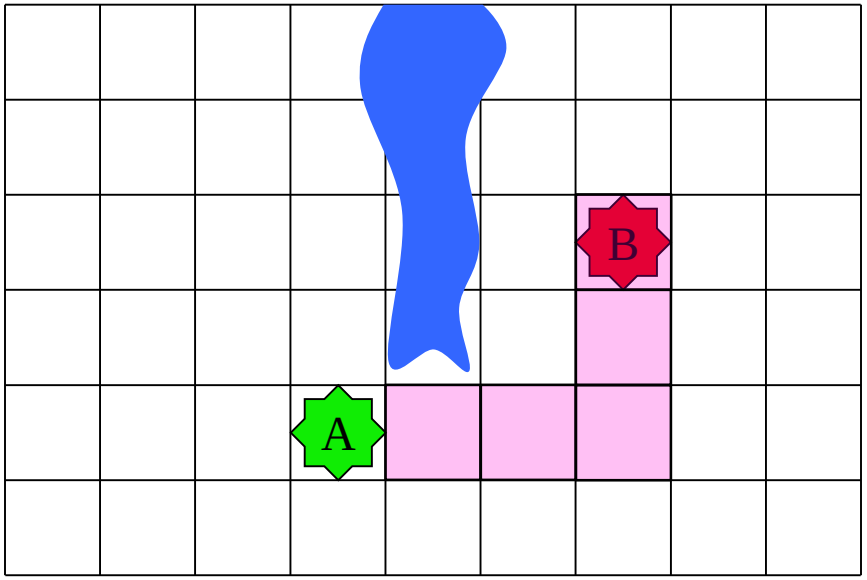
Breadth-First is Slow!



Idea: use heuristics

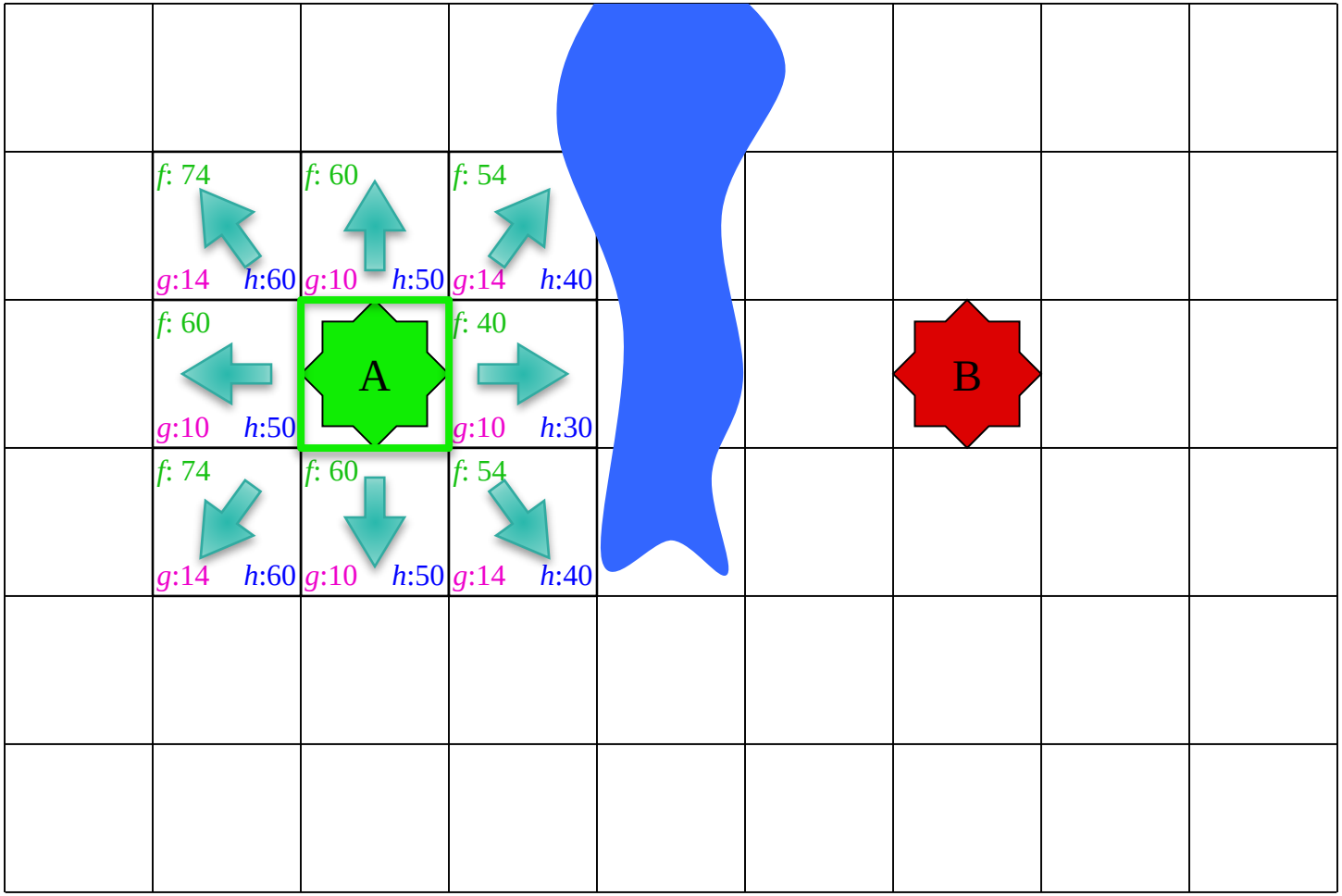
A* Algorithm

- Score $f = g + h$
  - g: distance on **best path**
  - h: naïve distance to **goal**

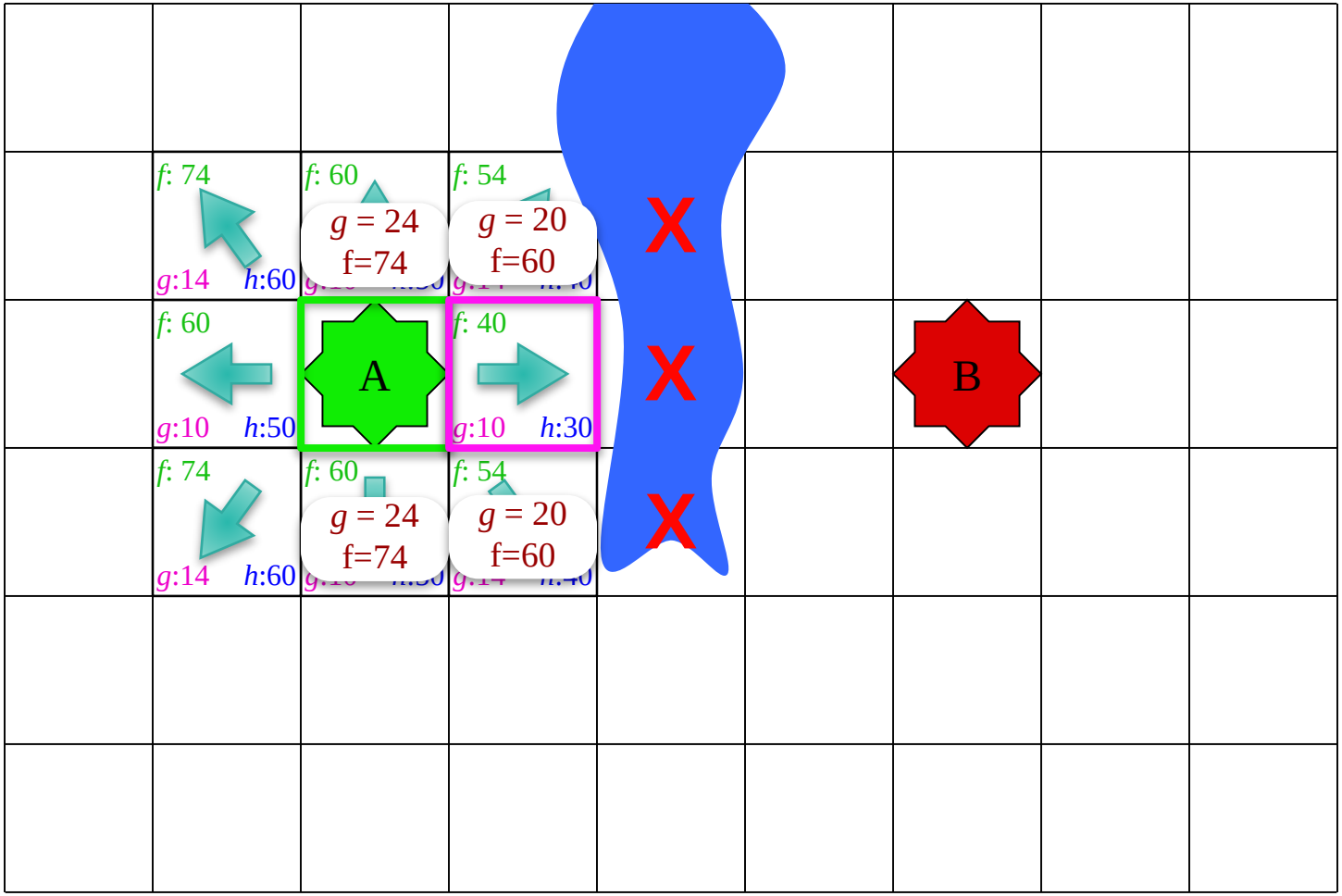


Manhattan distance = 30+20 = 50

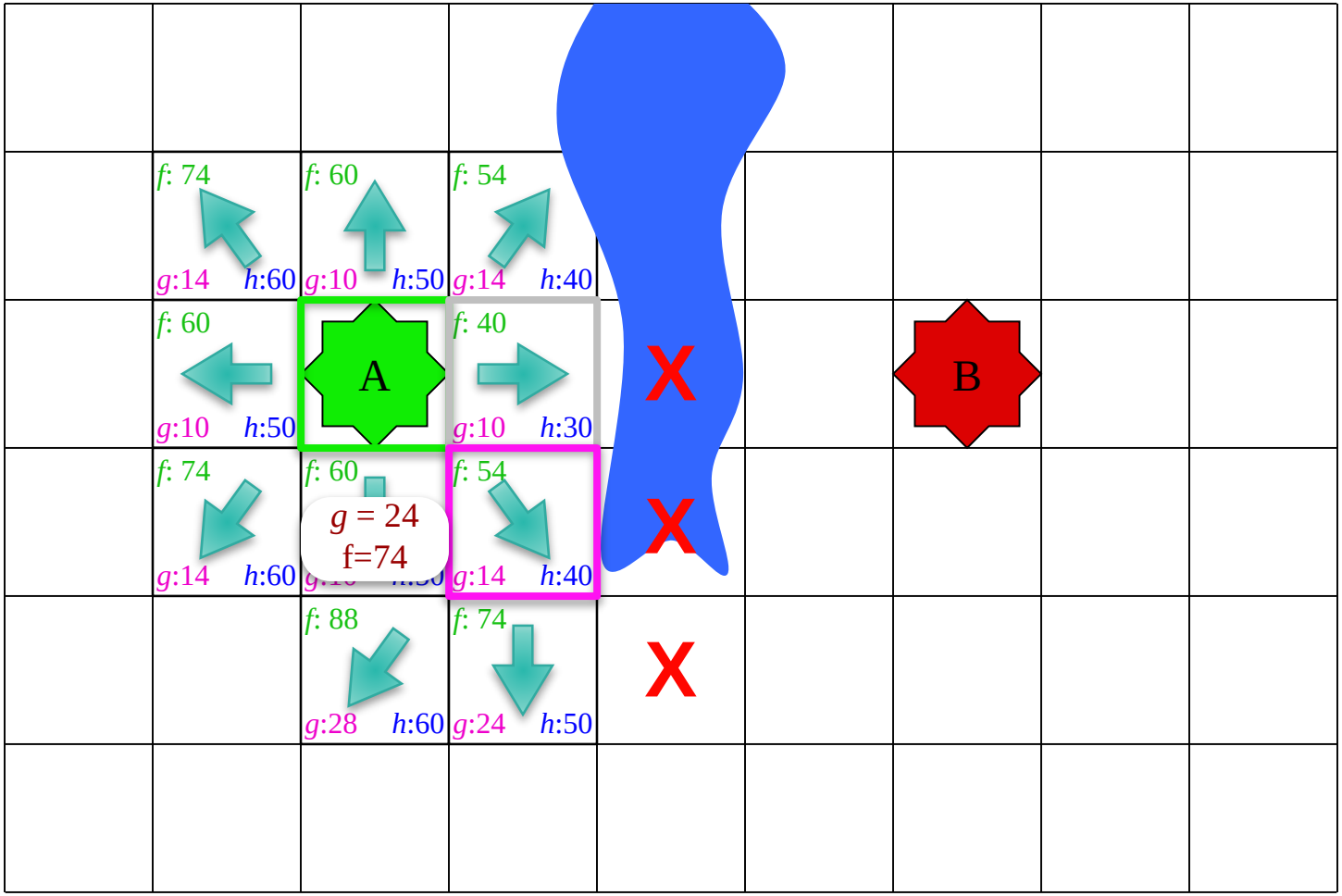
Pathfinding: A* Algorithm



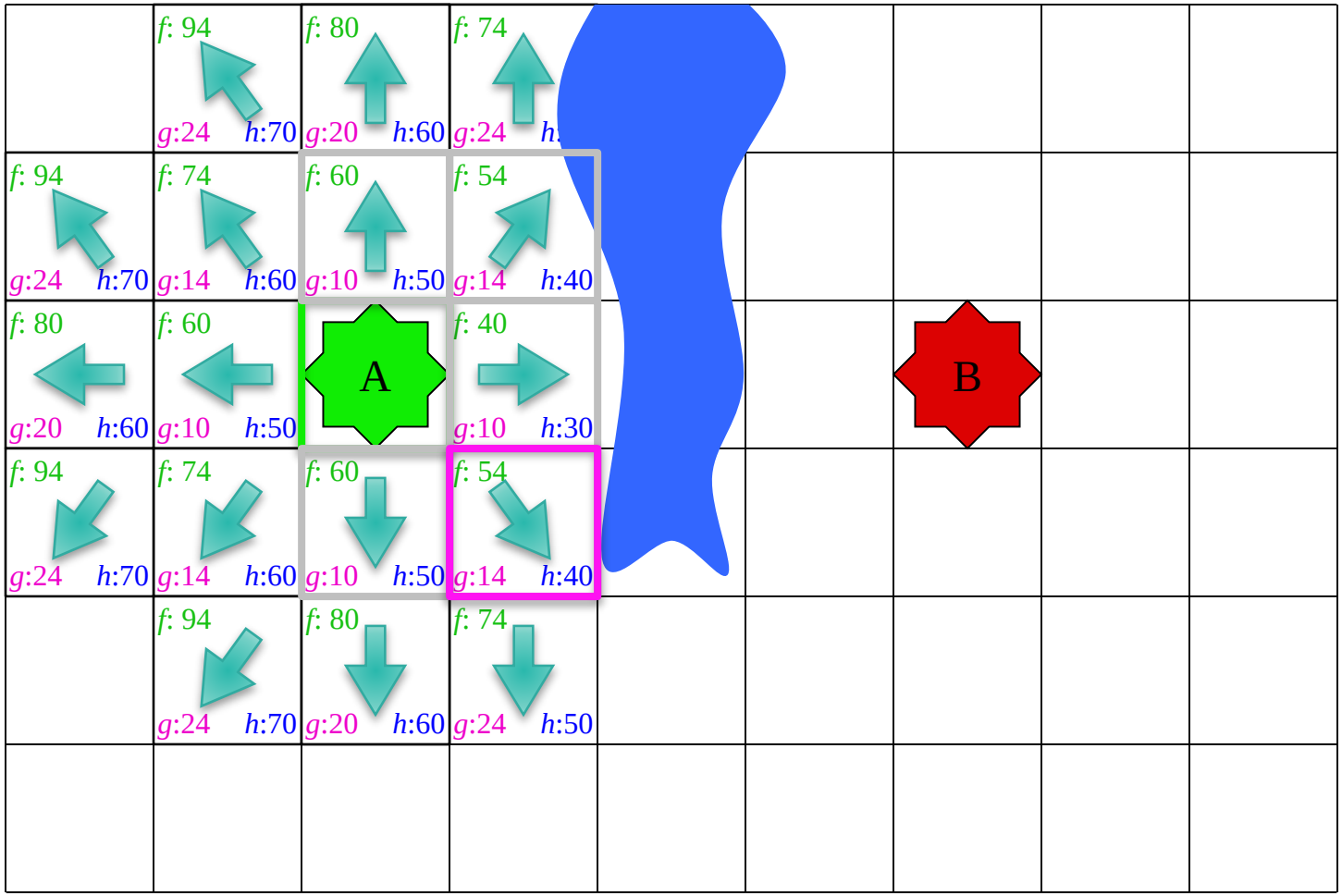
Pathfinding: A* Algorithm



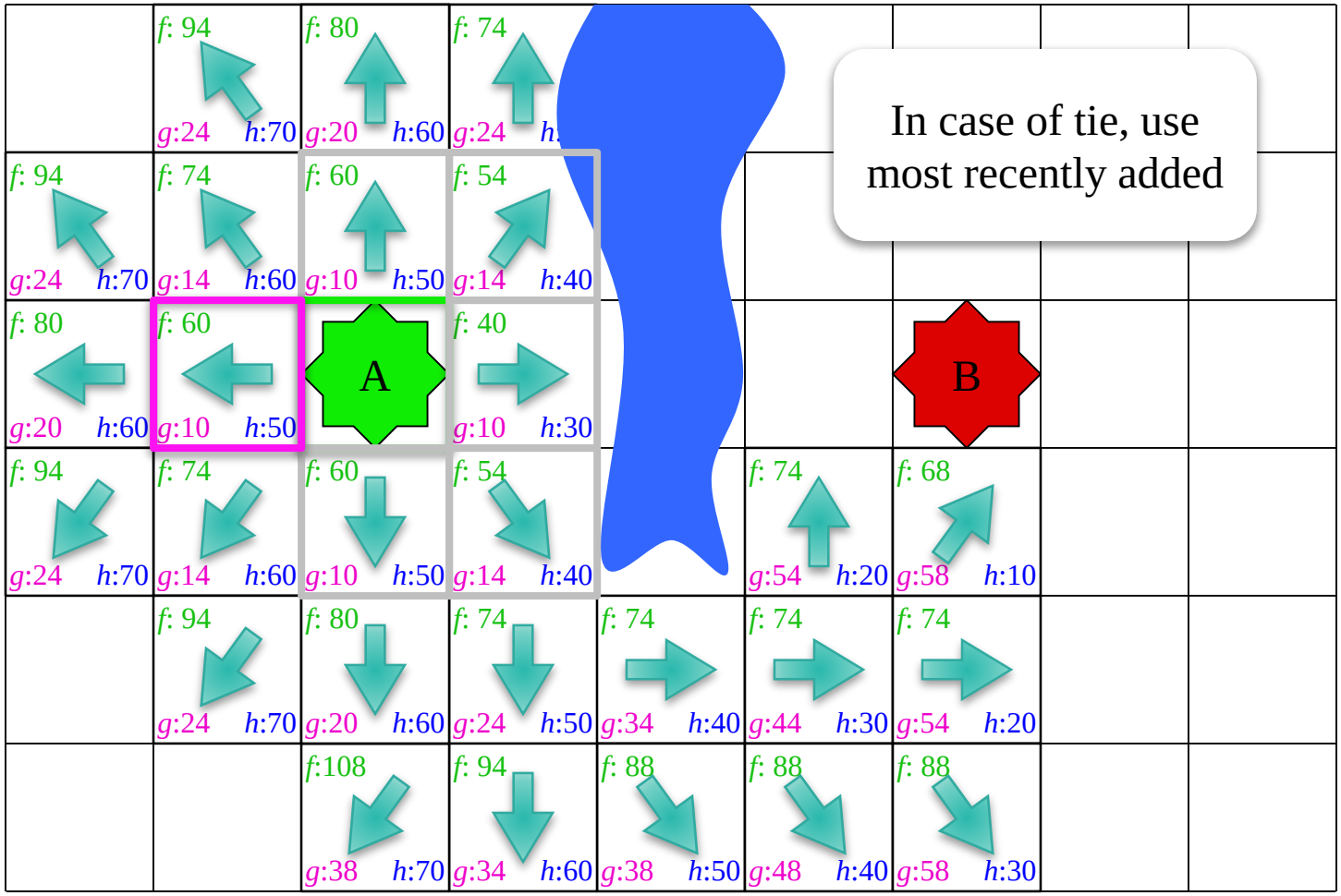
Pathfinding: A* Algorithm



Pathfinding: A* Algorithm

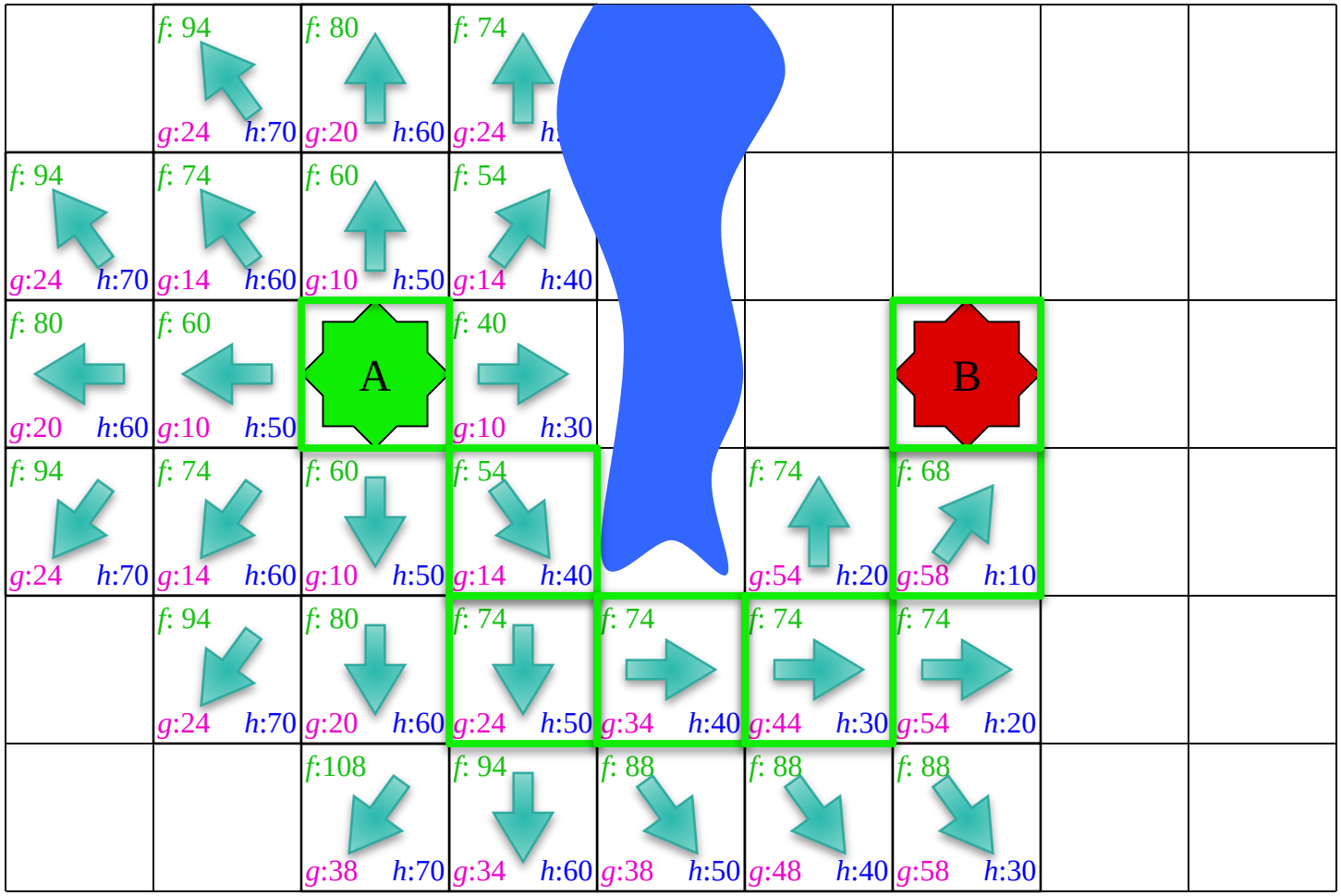


Pathfinding: A* Algorithm



In case of tie, use most recently added

Pathfinding: A* Algorithm



A* Mario


FPS: 24
Attempt: 1 of 1
AStarAgent
Selected Actions:
RIGHT
SPEED

Group Activity: AI++

- IF your game has no AI
  - That's How We Roll
  - Box
  - Thermo
  - THEN write a script that solves a level of your game
- IF your game has minimal AI
  - Sleep Walker
  - Pyrokid
  - Epic's Epic Epic
  - THEN improve the AI script
- IF your game has heavy AI
  - Zombify
  - Nameless
  - THEN discuss how you would add planning, heuristics, or intelligent randomness