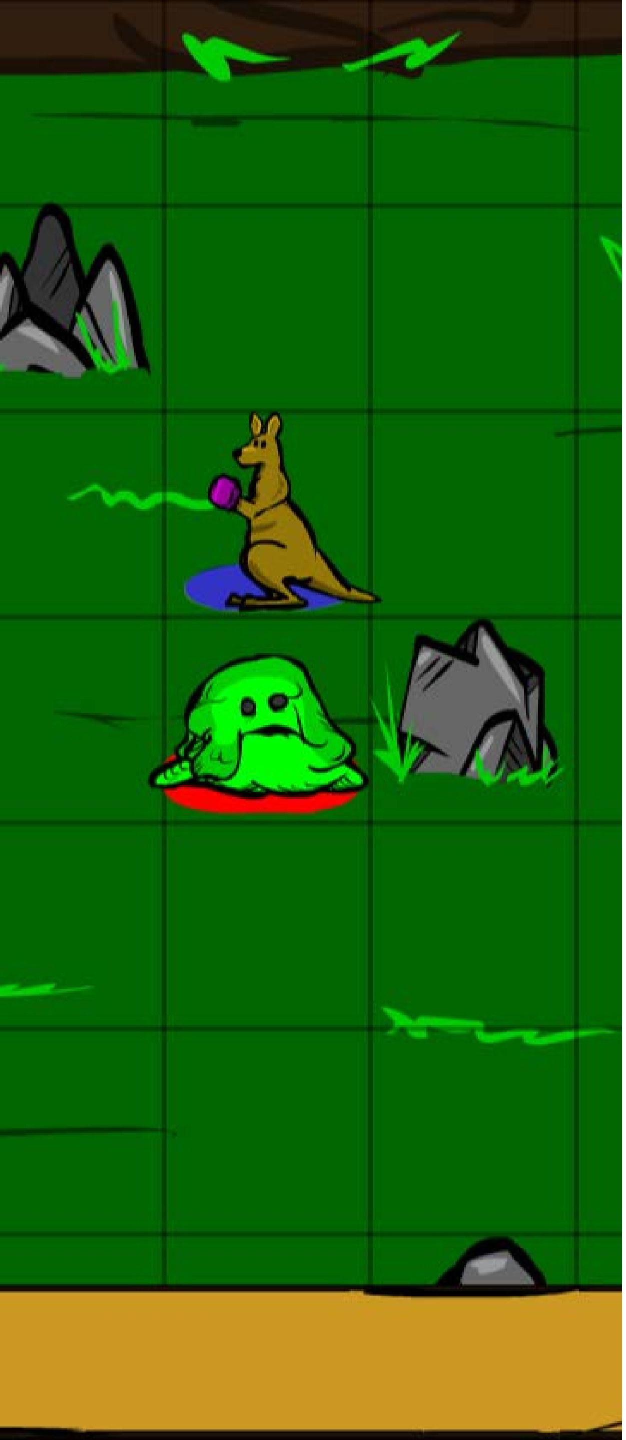


# CS/INFO 4154:

## Analytics-driven Game Design

Lecture 7:

## Artificial Intelligence

# Alpha Prototype

- Thursday, October 1st
- Three playable levels

# Artificial Intelligence!

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

# The Dream

**AI**

Play games automatically

**Procedural Content Generation**

Design games automatically

# What makes an AI good?

# What are examples of good AI?

# Today

- Scripts
- Planning
- Pathfinding

# Today

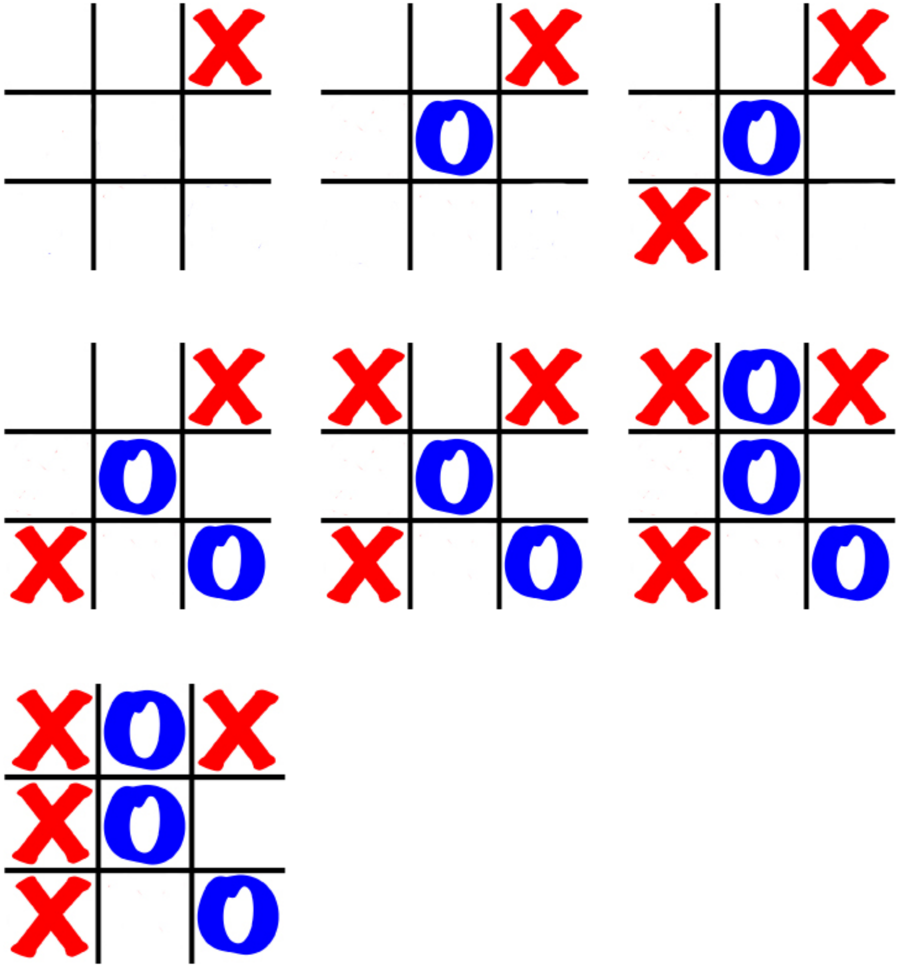
- Scripts
- Planning
- Pathfinding

# Scripts

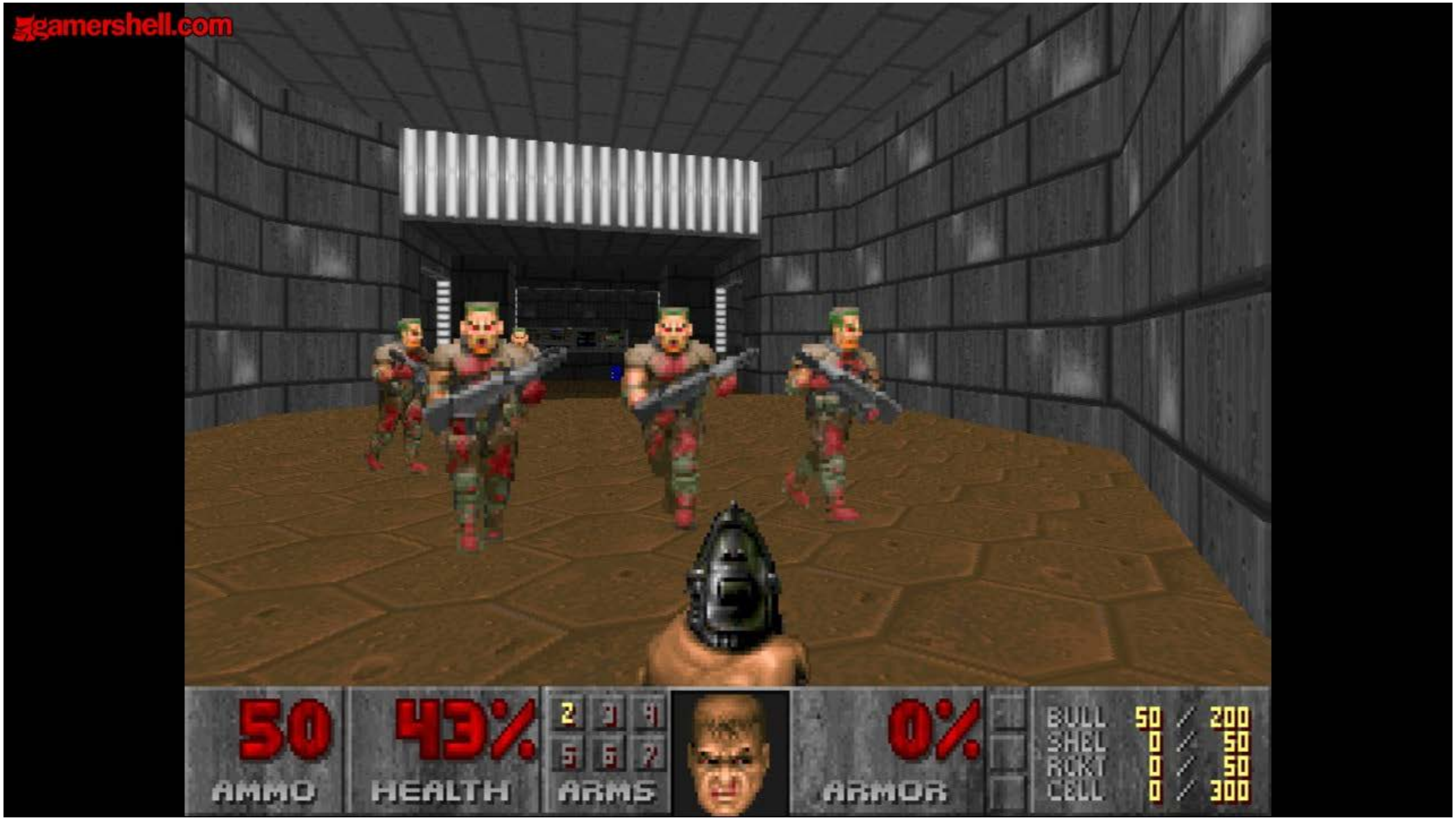
1. **IF** <condition is true> **THEN** <perform action>
2. **IF** <condition is true> **THEN** <perform action>
3. **IF** <condition is true> **THEN** <perform action>

# Example: Tic Tac Toe

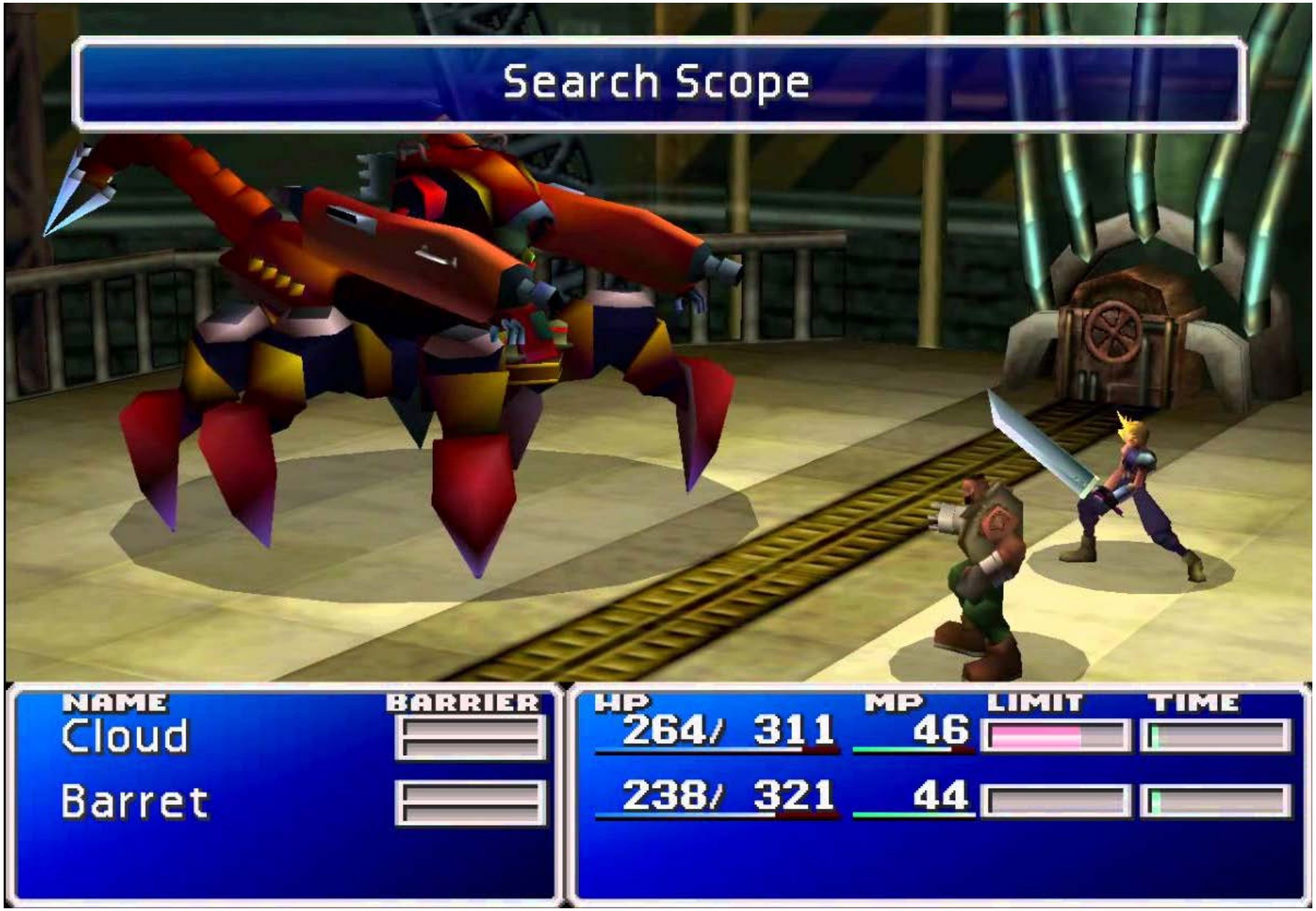
1. If I can win, win
2. If opponent can win, prevent it
3. If center is available, take it
4. If corner is available, take it



# Doom II


gamershell.com
50
AMMO
43%
HEALTH
2 3 4
5 6 7
ARMS
0%
ARMOR
BULL 50 / 200
SHEL 0 / 50
RCKT 0 / 50
CELL 0 / 300

# Final Fantasy VII



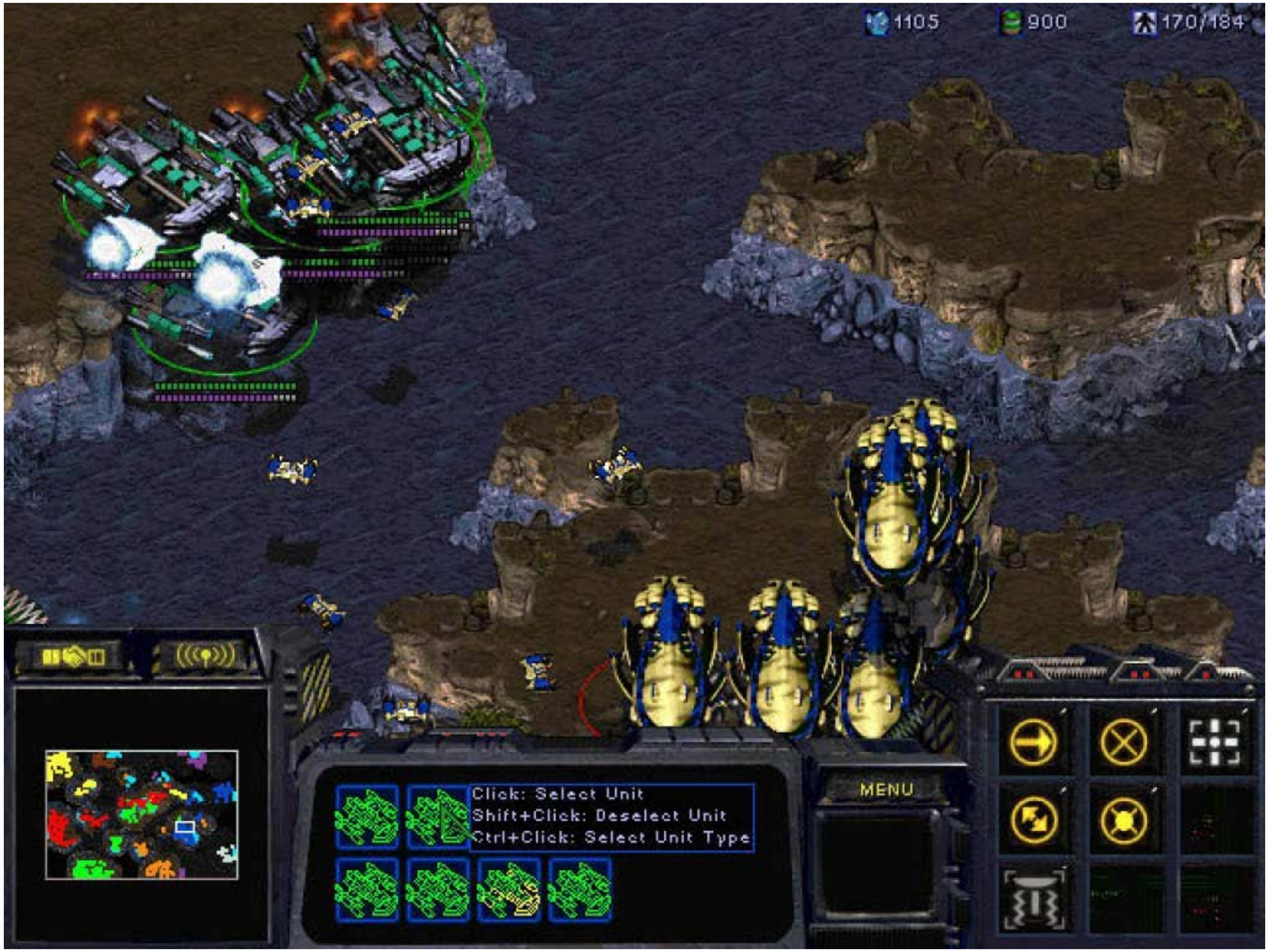
Search Scope

| NAME | BARRIER | HP | MP | LIMIT | TIME |
|---|---|---|---|---|---|
| Cloud | | 264/ 311 | 46 | | |
| Barret | | 238/ 321 | 44 | | |

# 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
```

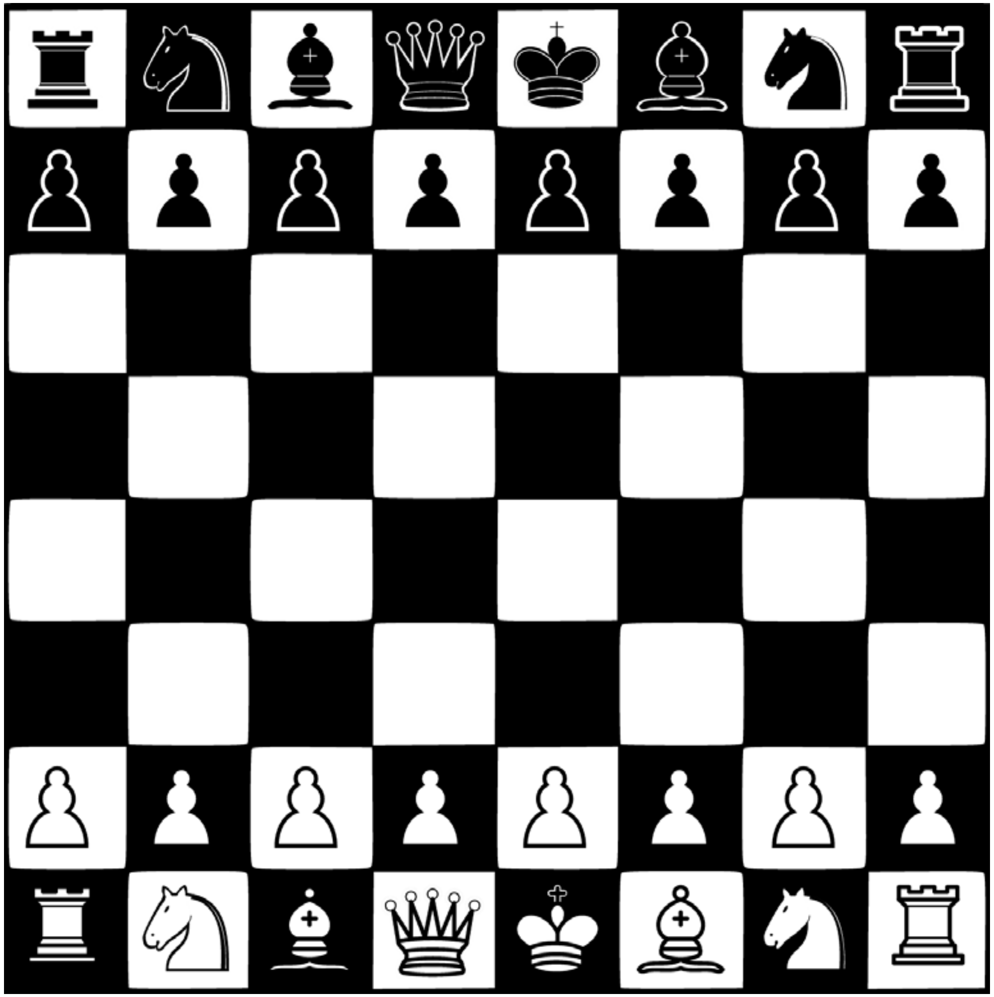
# “Cheating”



Click: Select Unit
Shift+Click: Deselect Unit
Ctrl+Click: Select Unit Type

# Advantages/Disadvantages

- + Easy to implement
- + Can express complex behaviors
- − “Smart” behavior can be ***very*** complex
- − Not so scalable
- − No natural way to vary difficulty



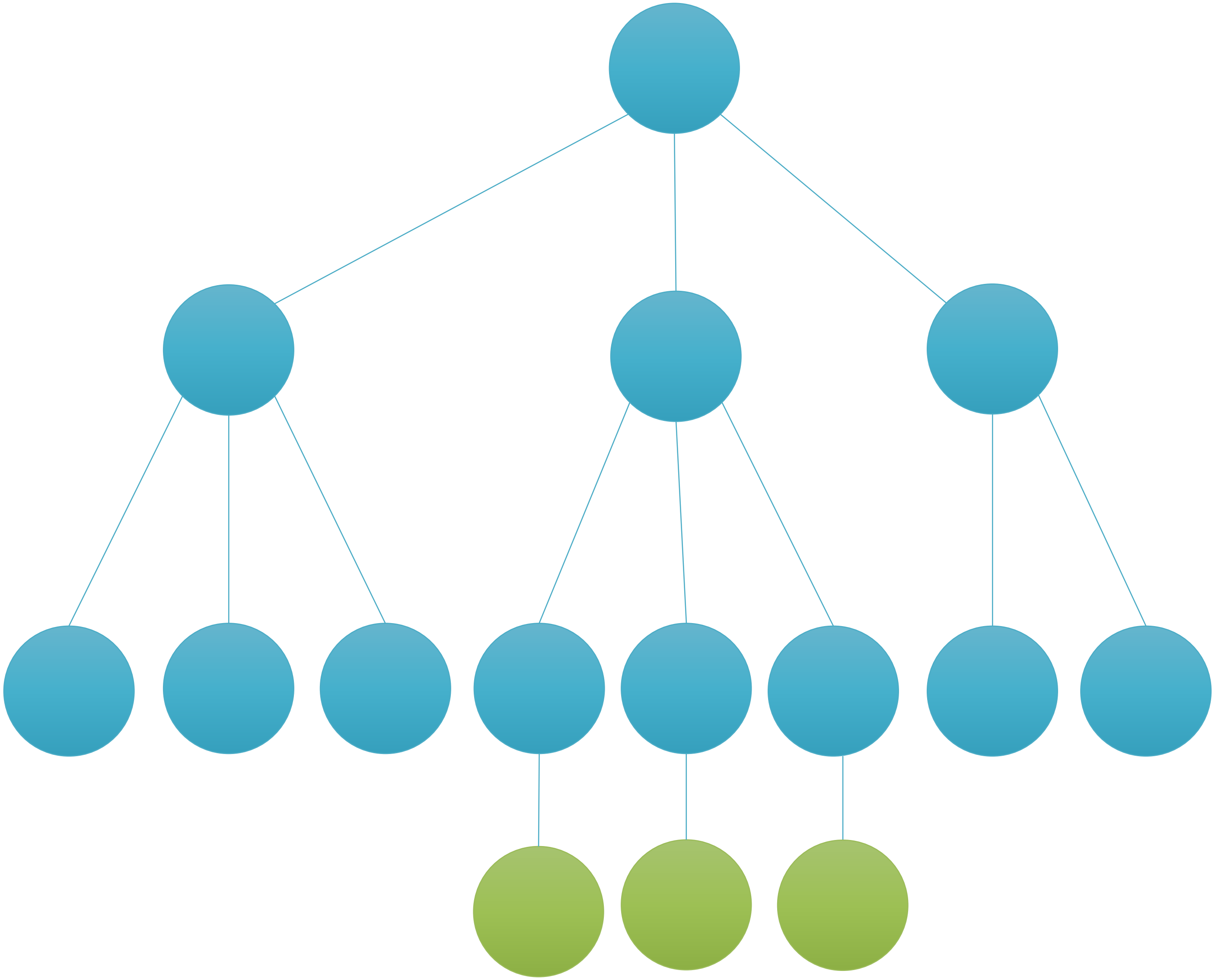
# Today

- Scripts
- Planning
- Pathfinding

# Today

- Scripts
- Planning
- Pathfinding

# 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

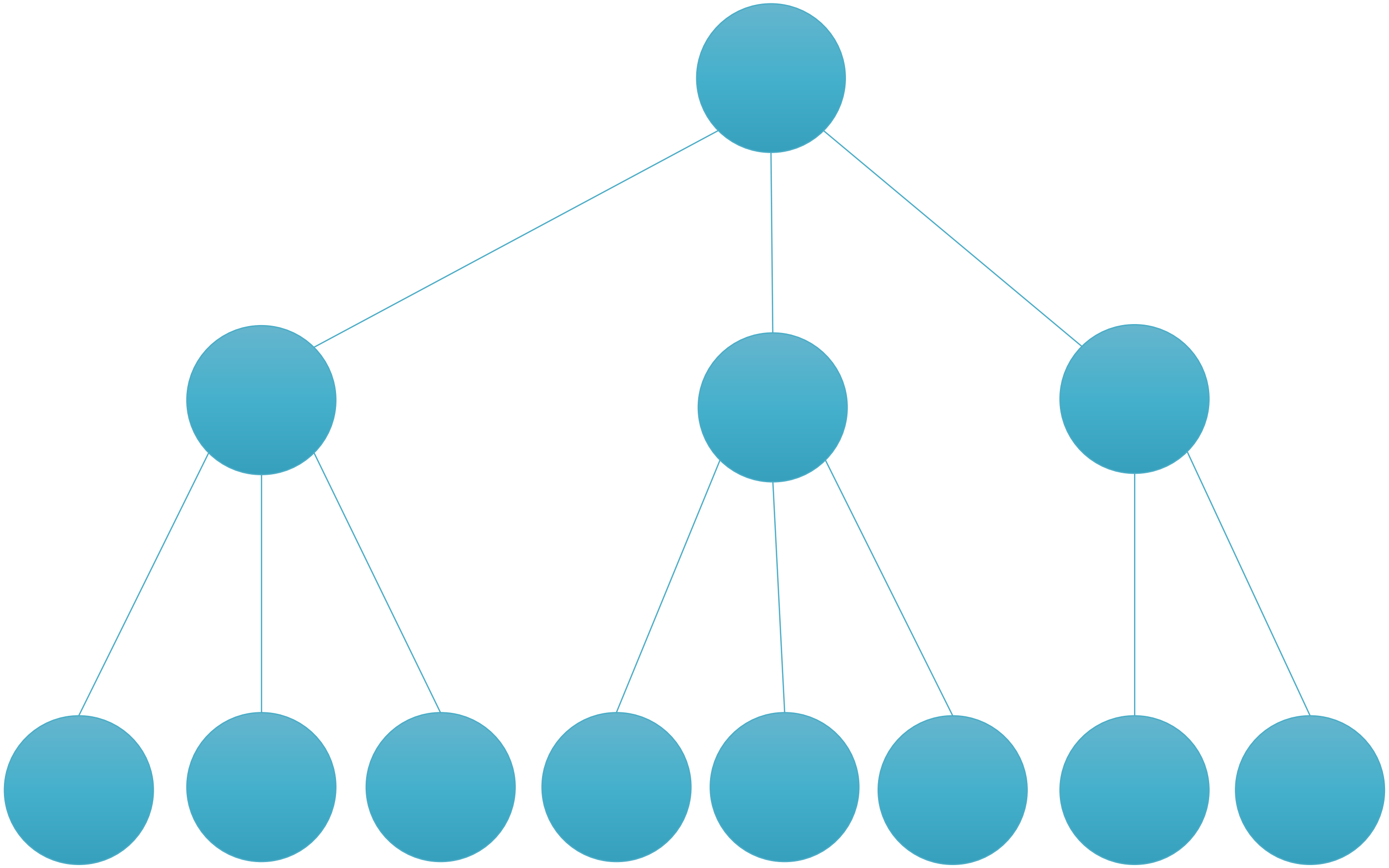
➖ Game must have certain properties

➖ 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

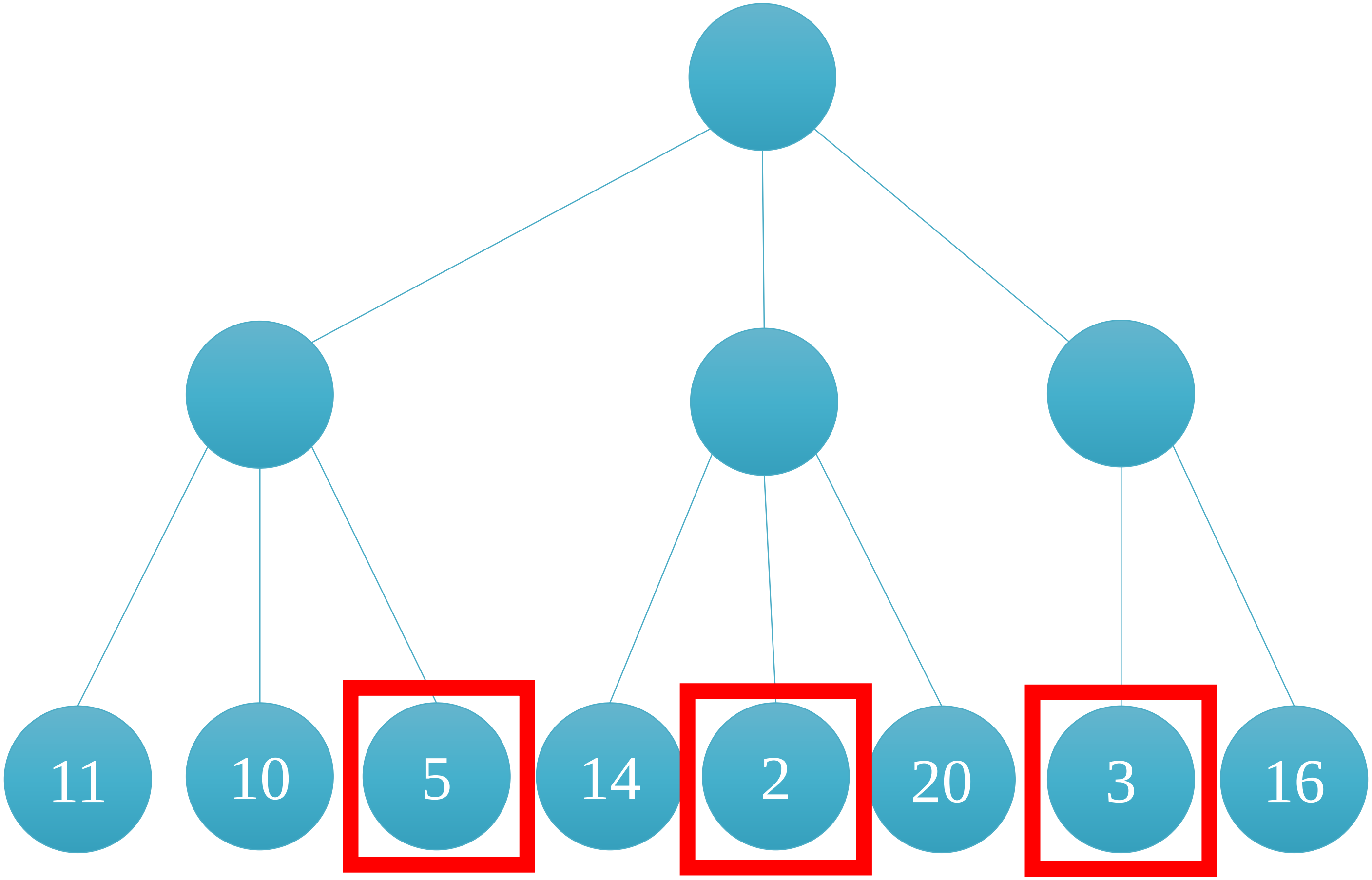
Opponent moves

# Heuristic Planning

Now

I move

Opponent moves

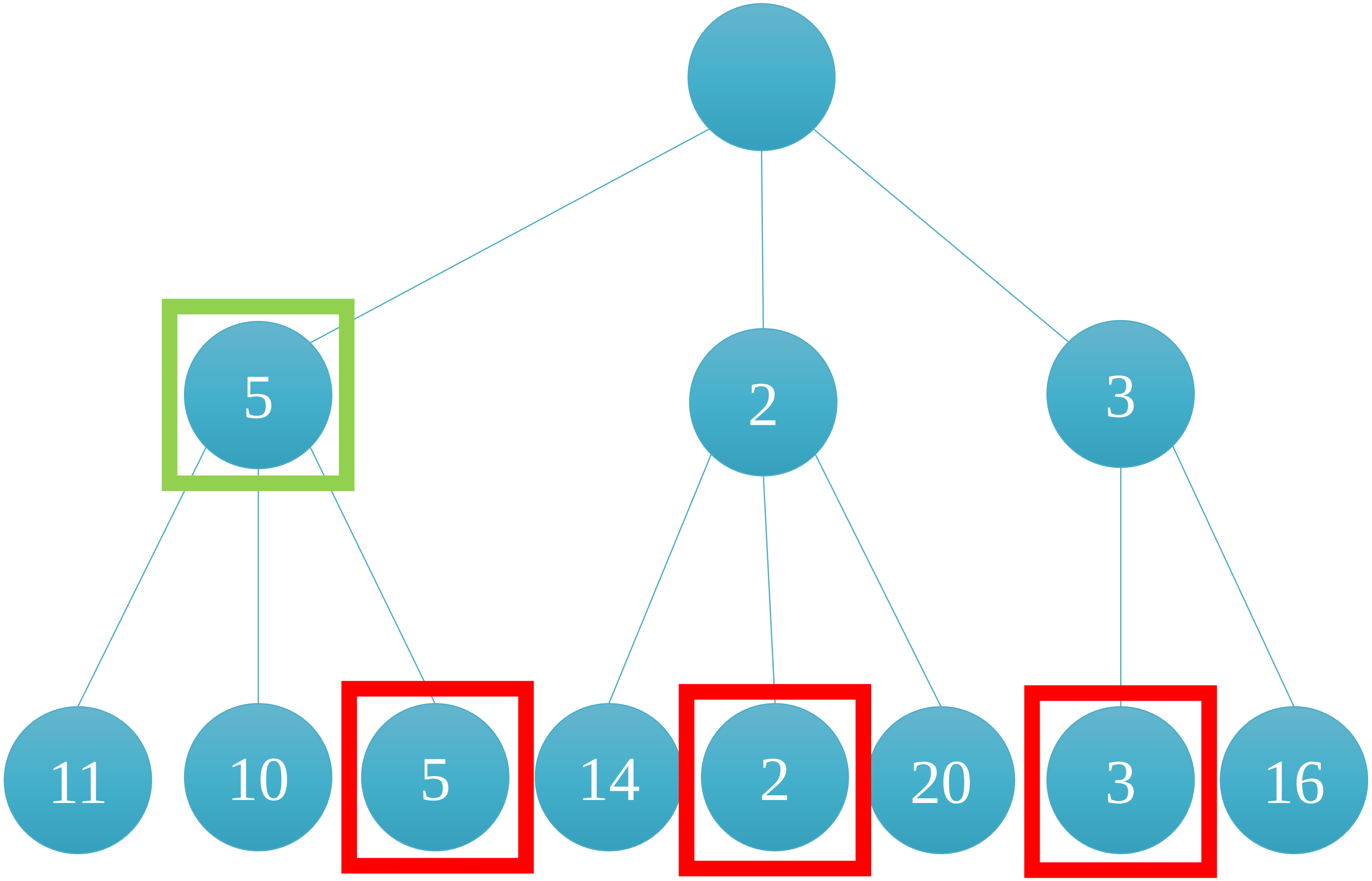


# Heuristic Planning

Now

I move

Opponent moves

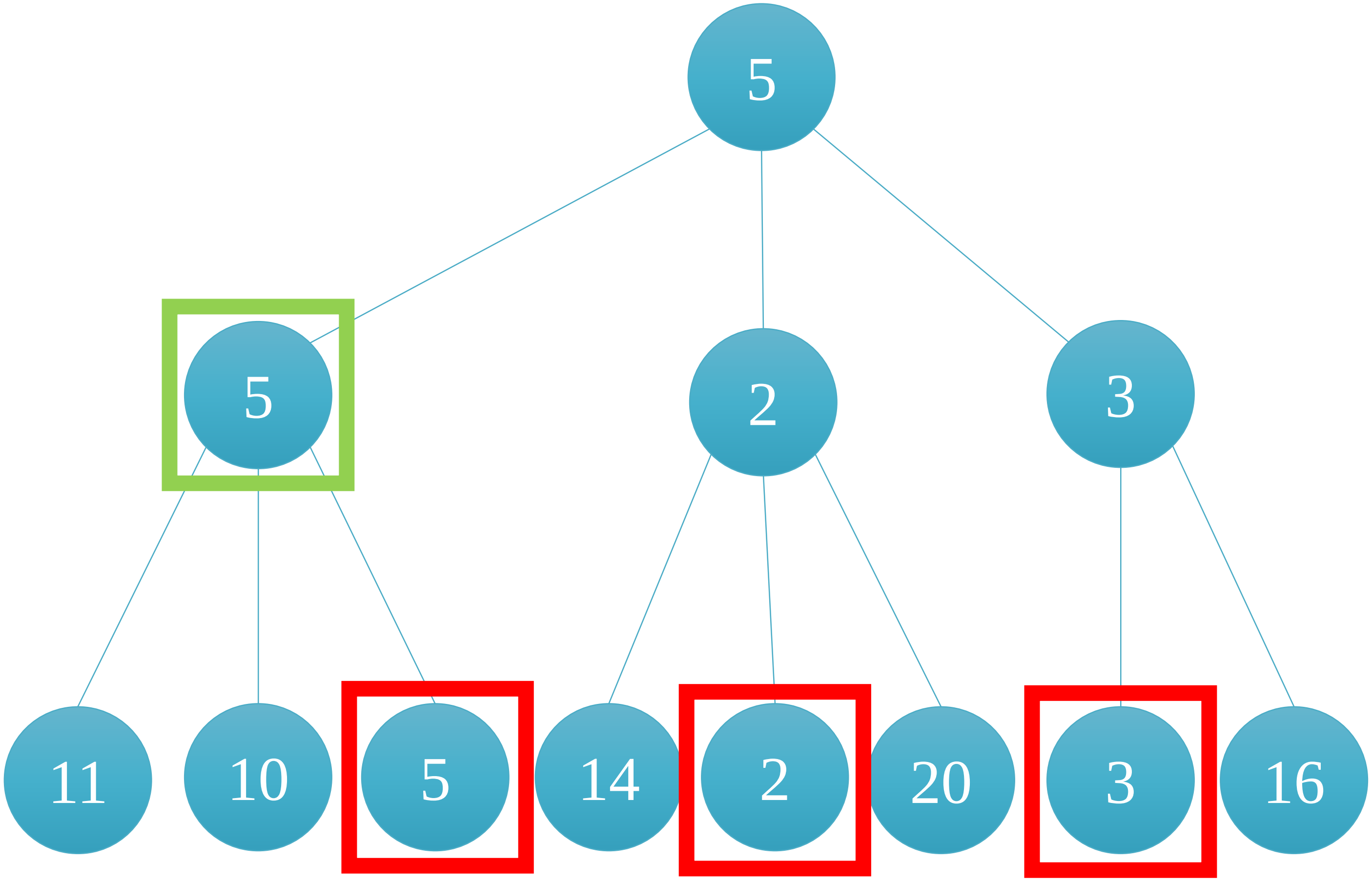


# Heuristic Planning

Now

I move

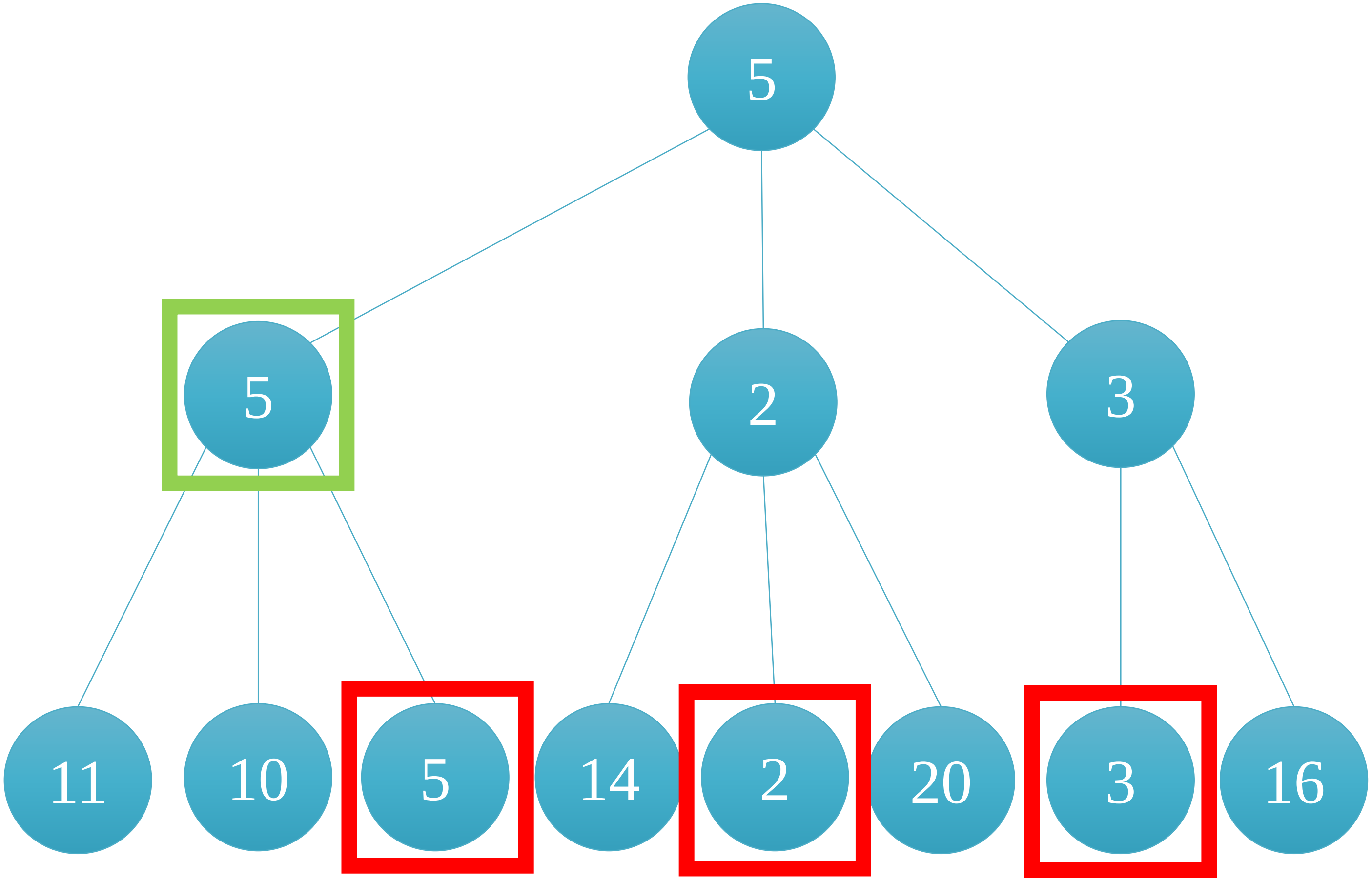
Opponent moves



# Minimax

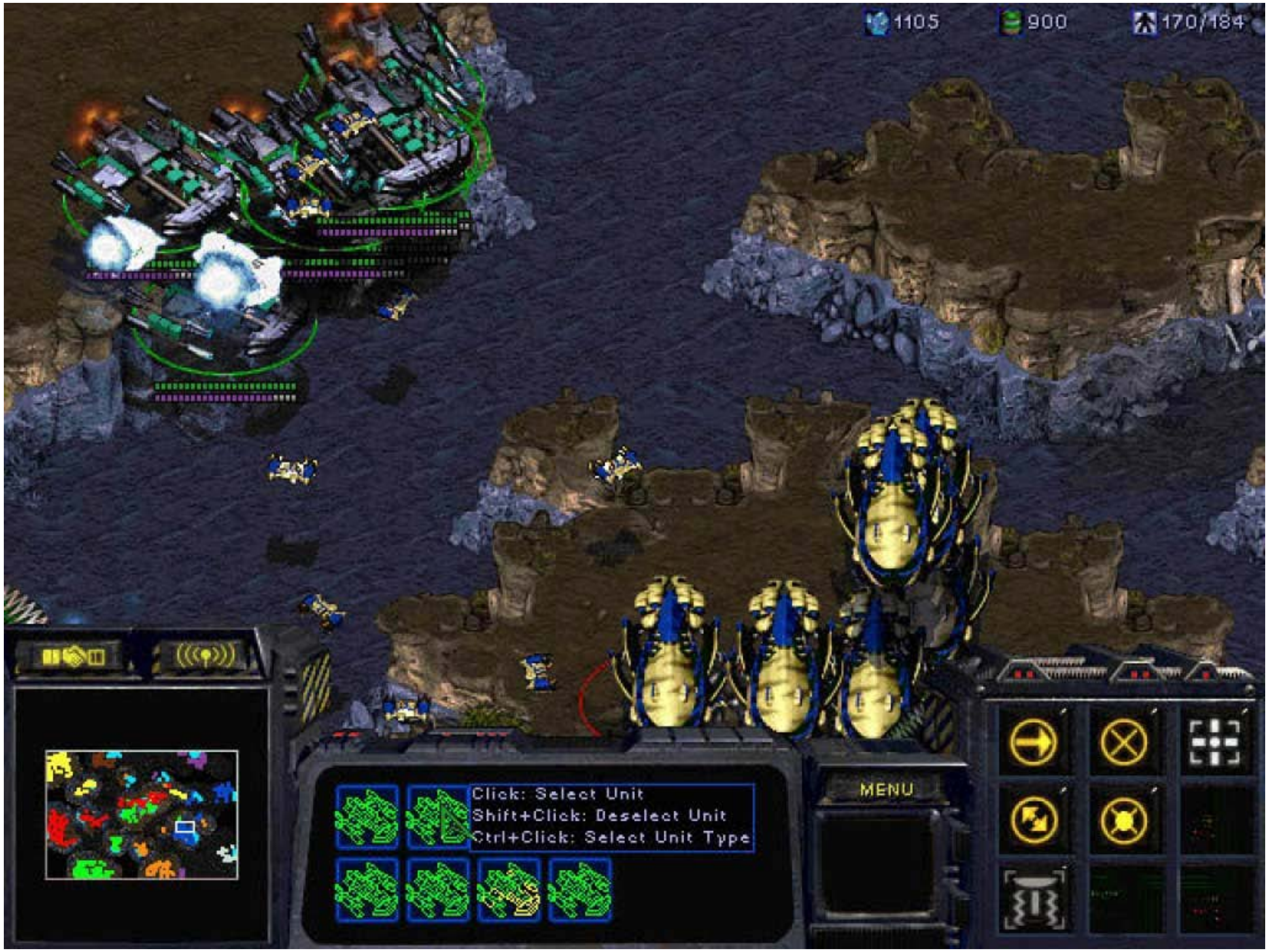
MAX

MIN



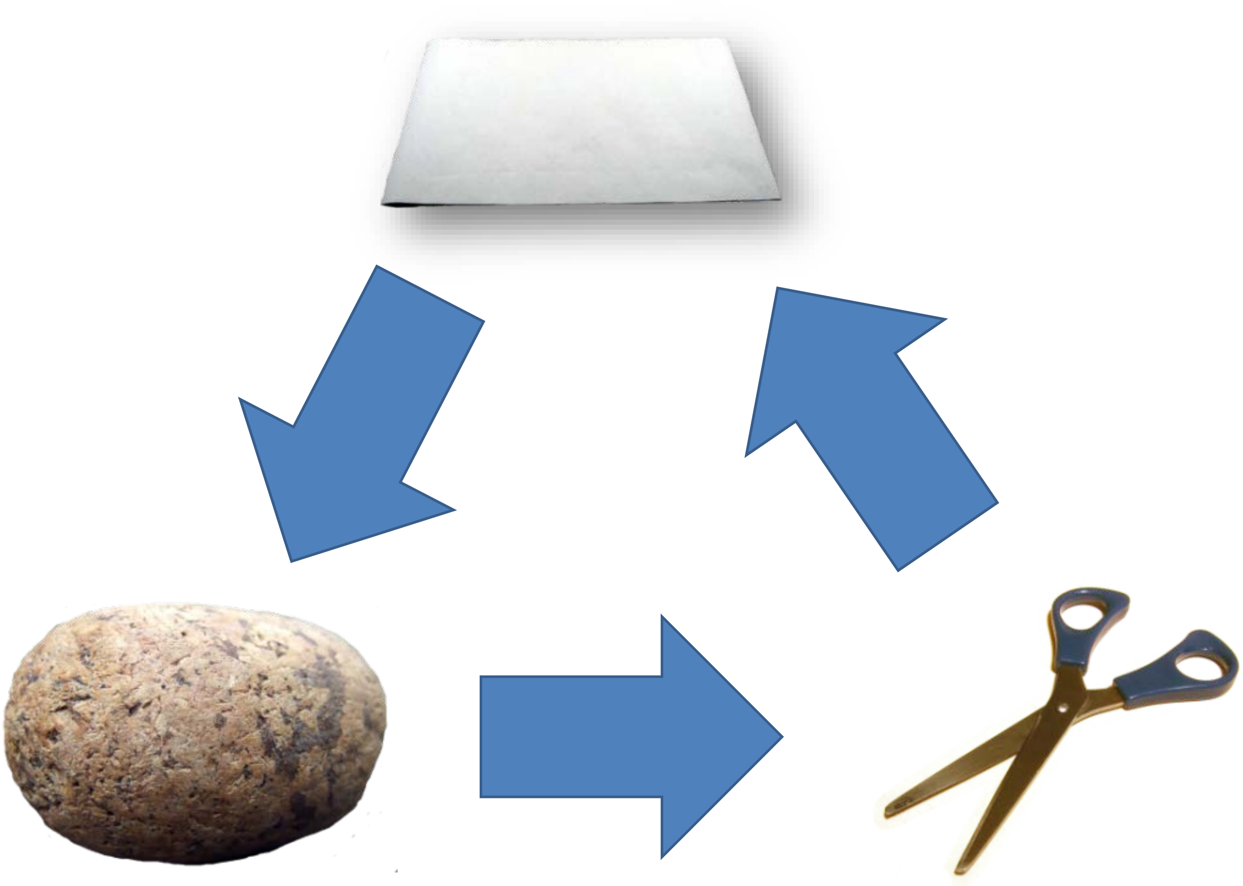
# Advantages/Disadvantages

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

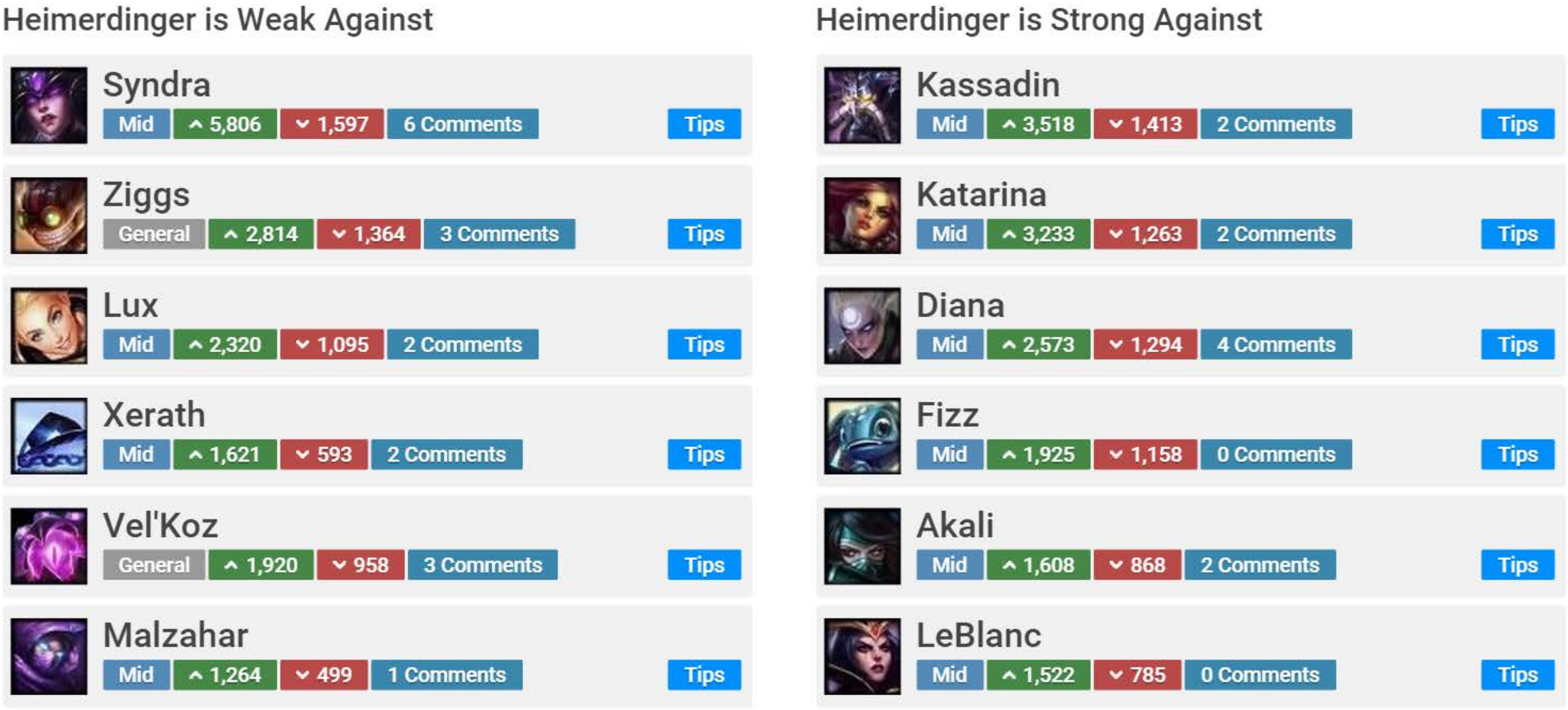


Click: Select Unit
Shift+Click: Deselect Unit
Ctrl+Click: Select Unit Type

# Simultaneous Actions



# A beats B beats C beats A



Heimerdinger is Weak Against

| | | | | | |
|---|---|---|---|---|---|
| Syndra | Mid | ⌃ 5,806 | ⌄ 1,597 | 6 Comments | Tips |
| Ziggs | General | ⌃ 2,814 | ⌄ 1,364 | 3 Comments | Tips |
| Lux | Mid | ⌃ 2,320 | ⌄ 1,095 | 2 Comments | Tips |
| Xerath | Mid | ⌃ 1,621 | ⌄ 593 | 2 Comments | Tips |
| Vel'Koz | General | ⌃ 1,920 | ⌄ 958 | 3 Comments | Tips |
| Malzahar | Mid | ⌃ 1,264 | ⌄ 499 | 1 Comments | Tips |

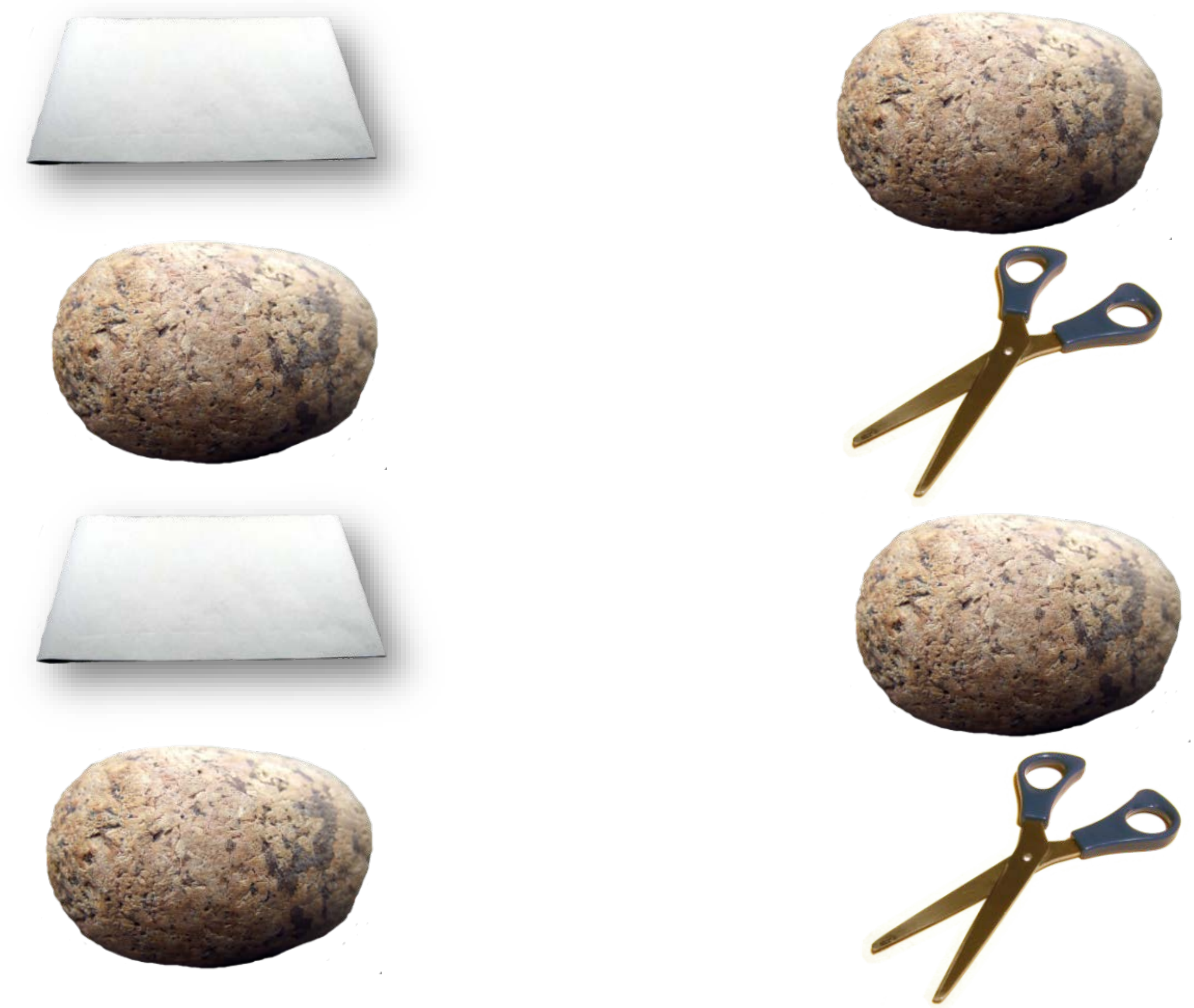
Heimerdinger is Strong Against

| | | | | | |
|---|---|---|---|---|---|
| Kassadin | Mid | ⌃ 3,518 | ⌄ 1,413 | 2 Comments | Tips |
| Katarina | Mid | ⌃ 3,233 | ⌄ 1,263 | 2 Comments | Tips |
| Diana | Mid | ⌃ 2,573 | ⌄ 1,294 | 4 Comments | Tips |
| Fizz | Mid | ⌃ 1,925 | ⌄ 1,158 | 0 Comments | Tips |
| Akali | Mid | ⌃ 1,608 | ⌄ 868 | 2 Comments | Tips |
| LeBlanc | Mid | ⌃ 1,522 | ⌄ 785 | 0 Comments | Tips |

# A beats B beats C beats A

# Opponent always plays rock…

# Opponent plays…



# Idea: Mixed Strategy

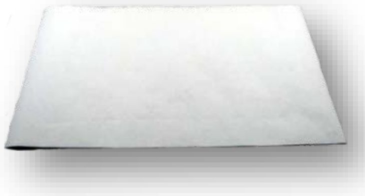
$$\frac{2}{3}$$

$$\frac{2}{3} \qquad \frac{1}{3}$$

# Idea: Mixed Strategy

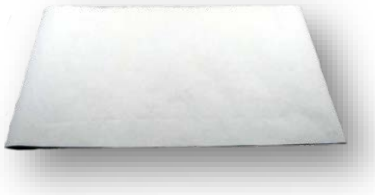
?

$\frac{2}{3}$



?

$\frac{1}{3}$



$\frac{2}{3}$

?

0

$\frac{2}{3}$

# Idea: Mixed Strategy

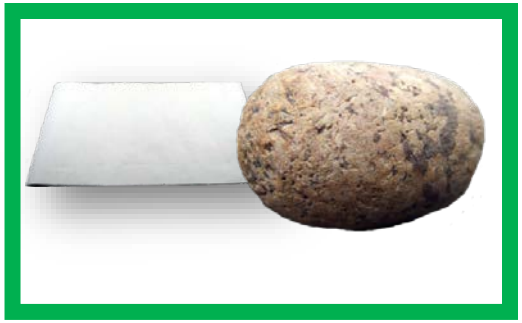
0 $\frac{2}{3}$

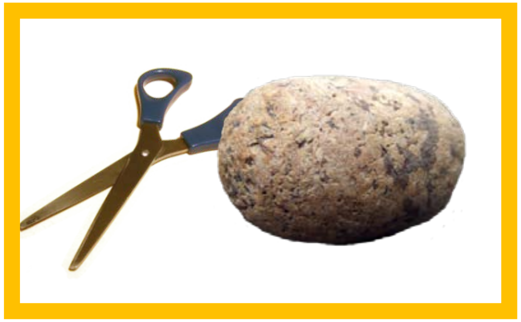
1 $\frac{1}{3}$

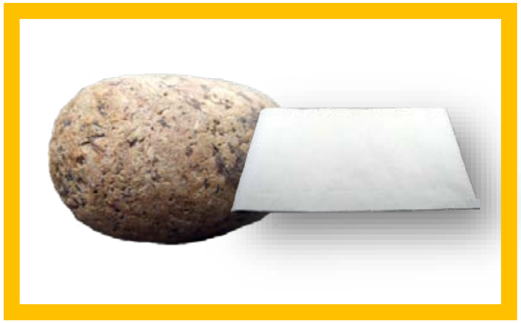
$\frac{2}{3}$ 0 0 $\frac{2}{3}$

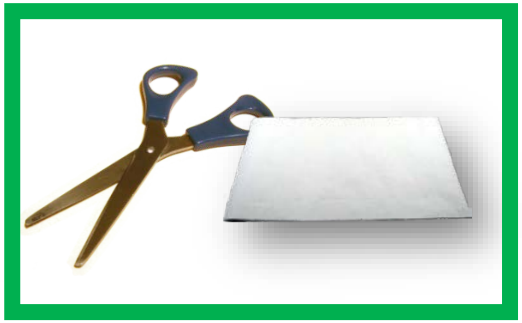
# Evaluating a strategy

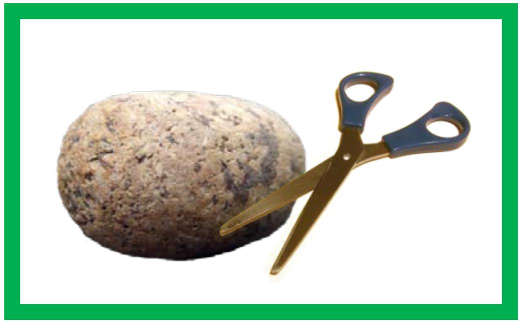
- Idea: compute the ***expected reward*** for a strategy

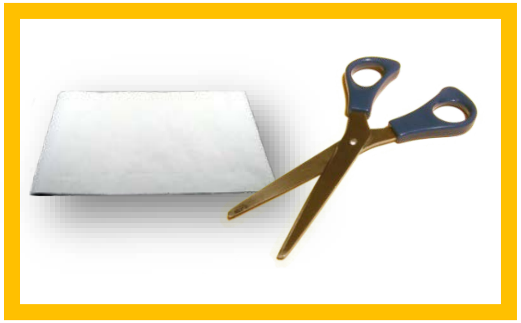


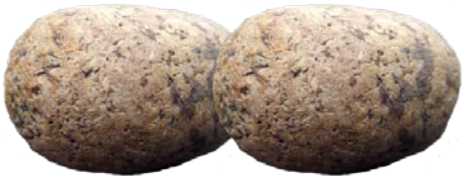




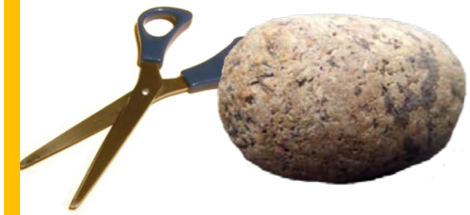


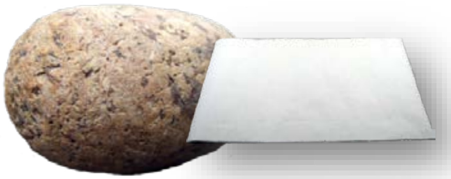


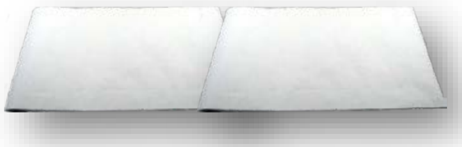




1

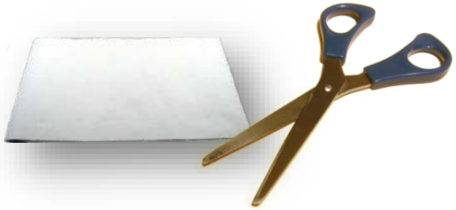


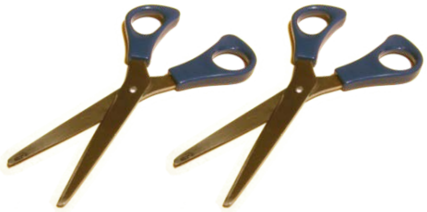


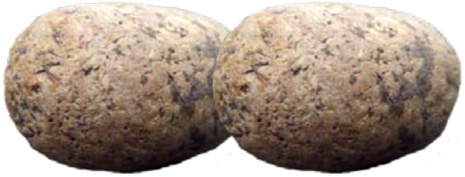
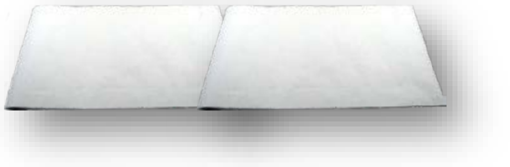
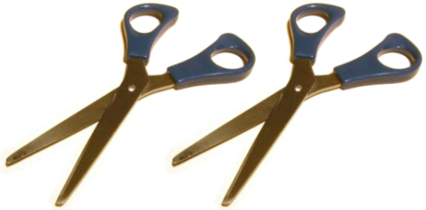


1

1





| | | |
|---|---|---|
|  | 1 | -1 |
| -1 |  | 1 |
| 1 | -1 |  |

|  |  |  |
|---|---|---|
| 0 | 1 | -1 |
| -1 | 0 | 1 |
| 1 | -1 | 0 |

# Reward Matrix

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

$0$

$\frac{2}{3}$

$1$

$\frac{1}{3}$

$0$

$0$

# Expected Reward

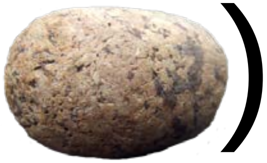
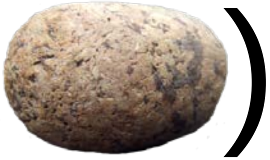
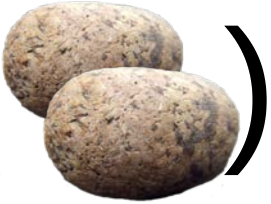
= probability of $\text{event}_1$ * reward of $\text{event}_1$

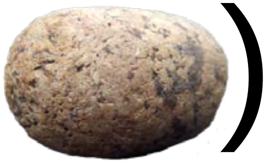
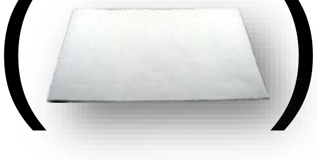
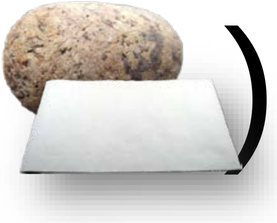
+ probability of $\text{event}_2$ * reward of $\text{event}_2$

+ probability of $\text{event}_3$ * reward of $\text{event}_3$

…

# Expected Reward

$=p_{me}($  $)\times p_{you}($  $)\times R($  $)$

$+p_{me}($  $)\times p_{you}($  $)\times R($  $)$

…

0

1

0

$\frac{2}{3}$

$\frac{1}{3}$

0

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

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

$$\overset{\pi_{you}{}^{T}}{\begin{bmatrix} \frac{2}{3} & \frac{1}{3} & 0 \end{bmatrix}} \begin{bmatrix} 0 & 1 & -1 \\ -1 & 0 & 1 \\ 1 & -1 & 0 \end{bmatrix} \underset{\pi_{me}}{\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}} = \frac{1}{3}$$

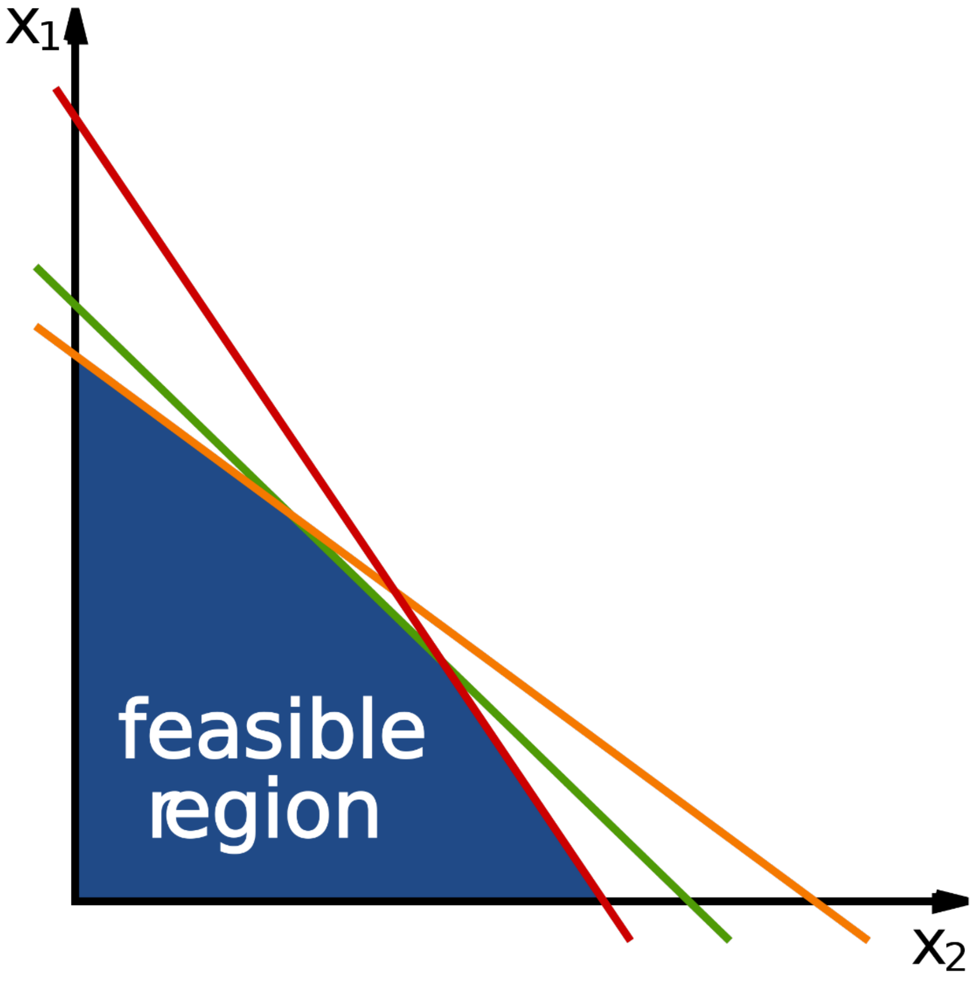
# Expected Reward

$$reward = {\pi_{you}}^{T} \boldsymbol{R} \pi_{me}$$

# Minimax!

$$\max_{\pi_{me}} \left( \min_{\pi_{you}} {\pi_{you}}^{T} \boldsymbol{R} \pi_{me} \right)$$

# Linear Programming



# Calculating Reward

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

$$\sum_i \pi_i = 1$$

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

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

# Optimal Strategy

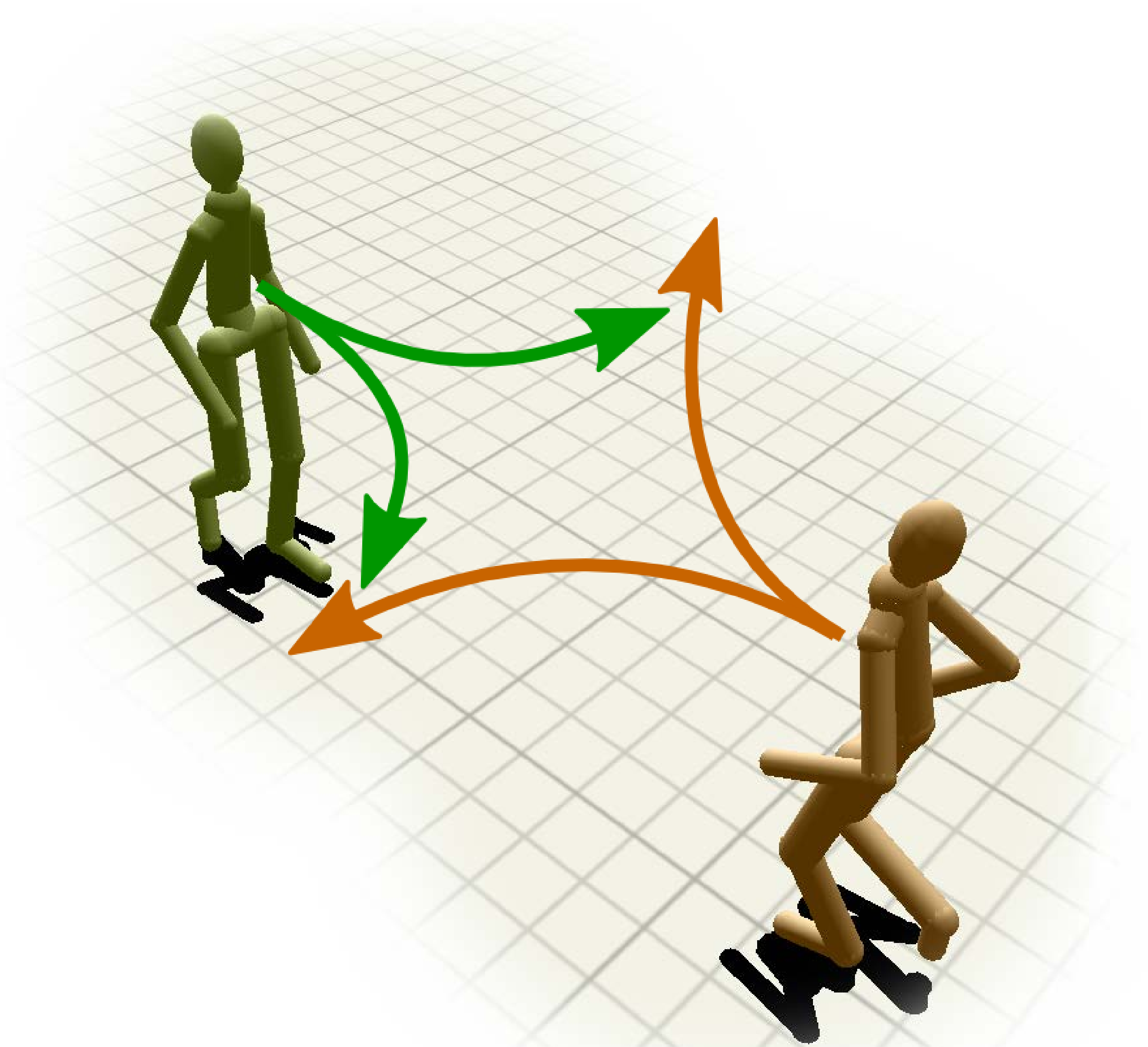
$$\frac{1}{3}$$

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

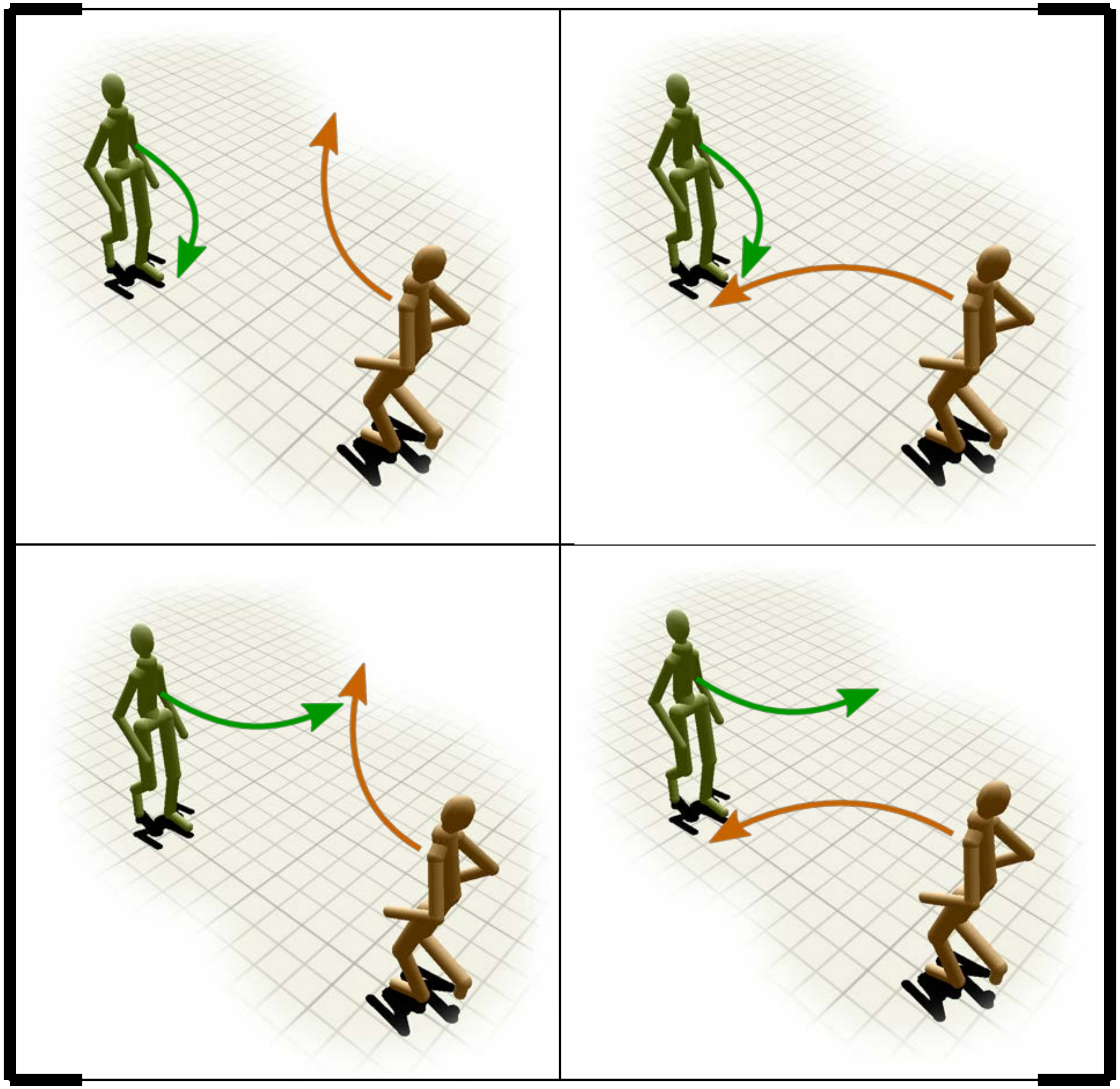
$$\frac{1}{3}$$

# Rock Paper Scissors

- http://www.bbc.com/news/technology-24803751

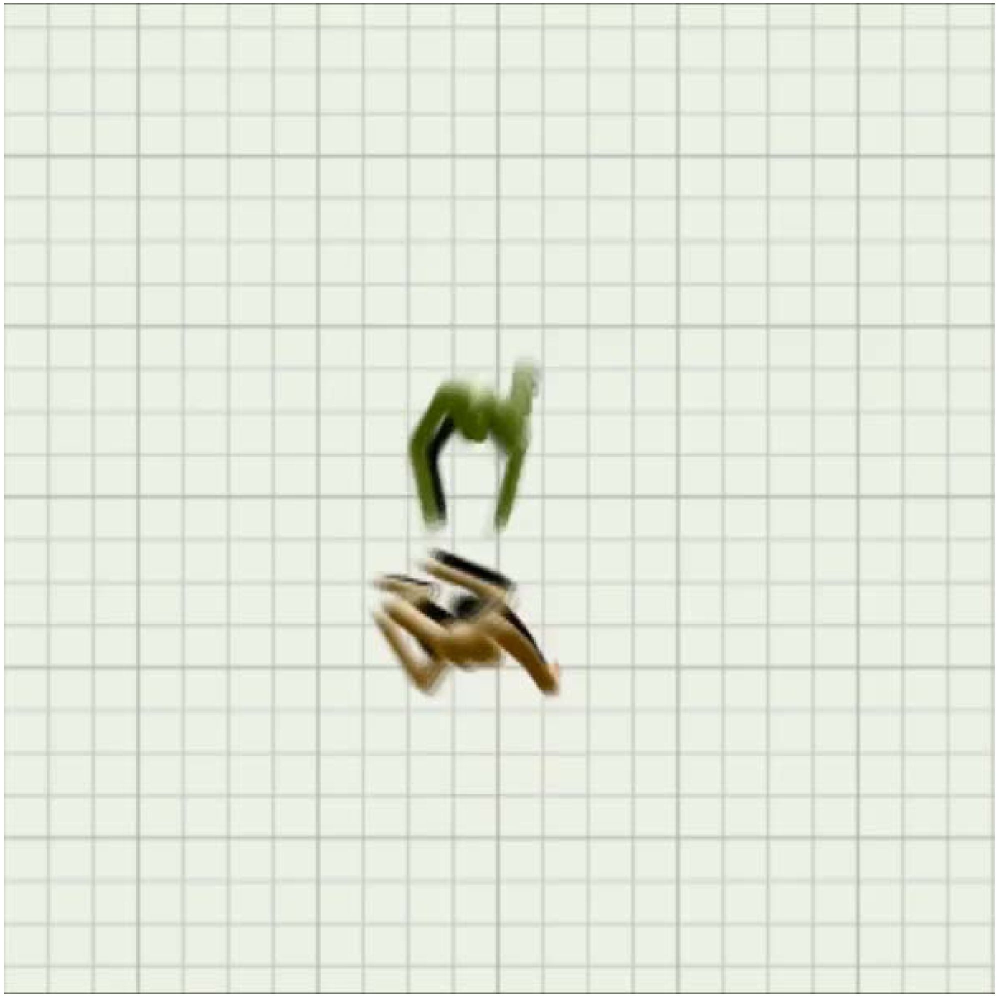


right left

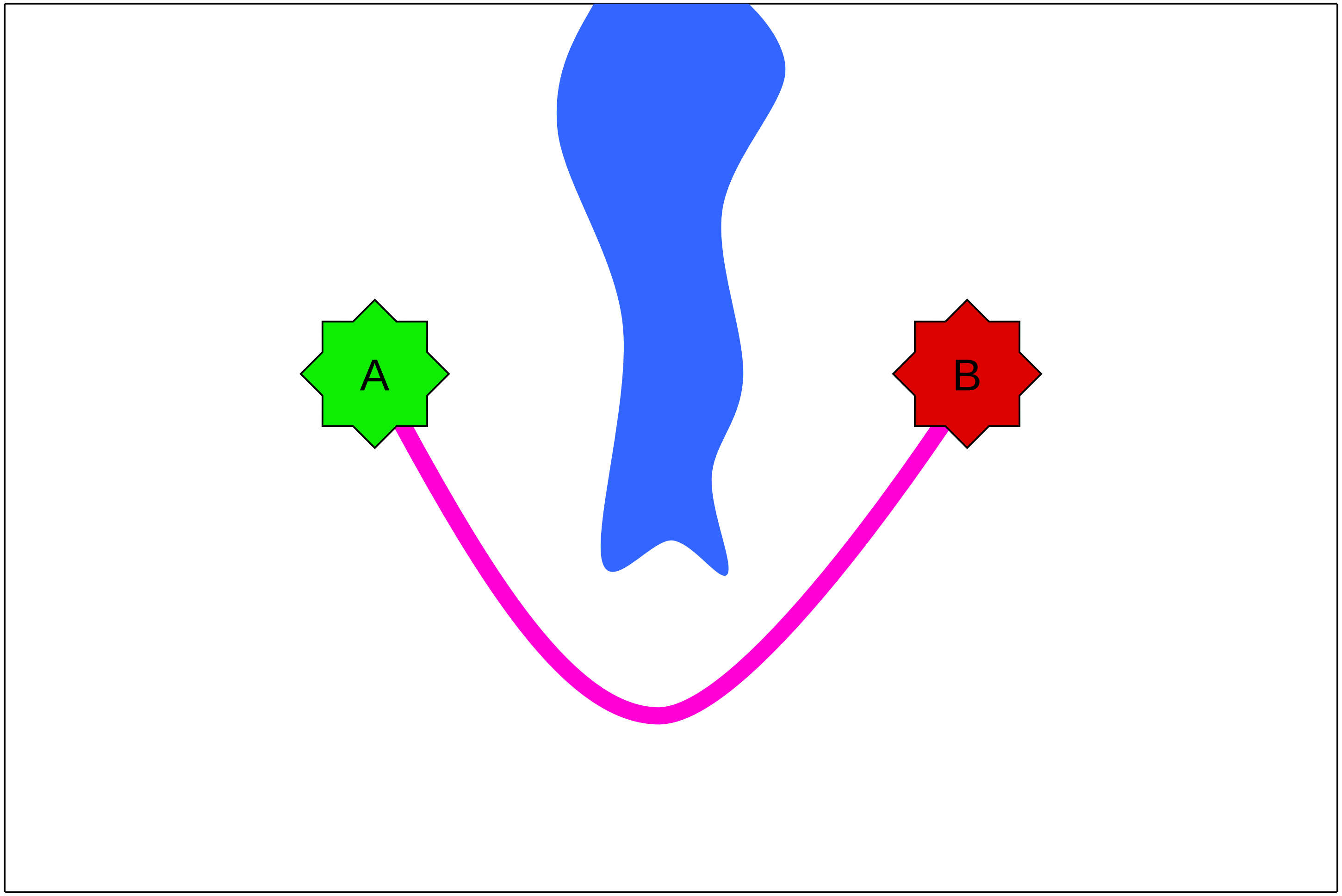


right

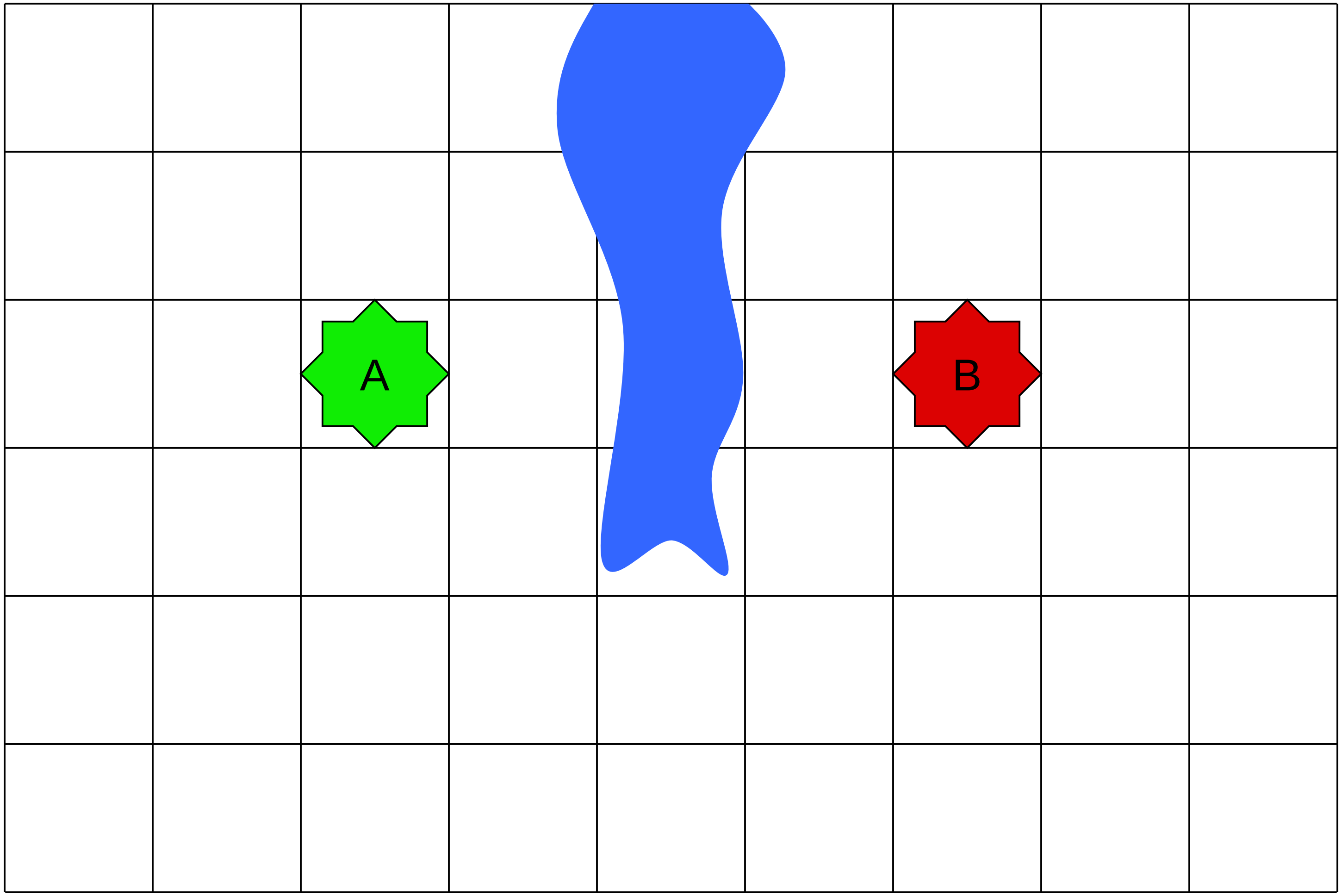
left



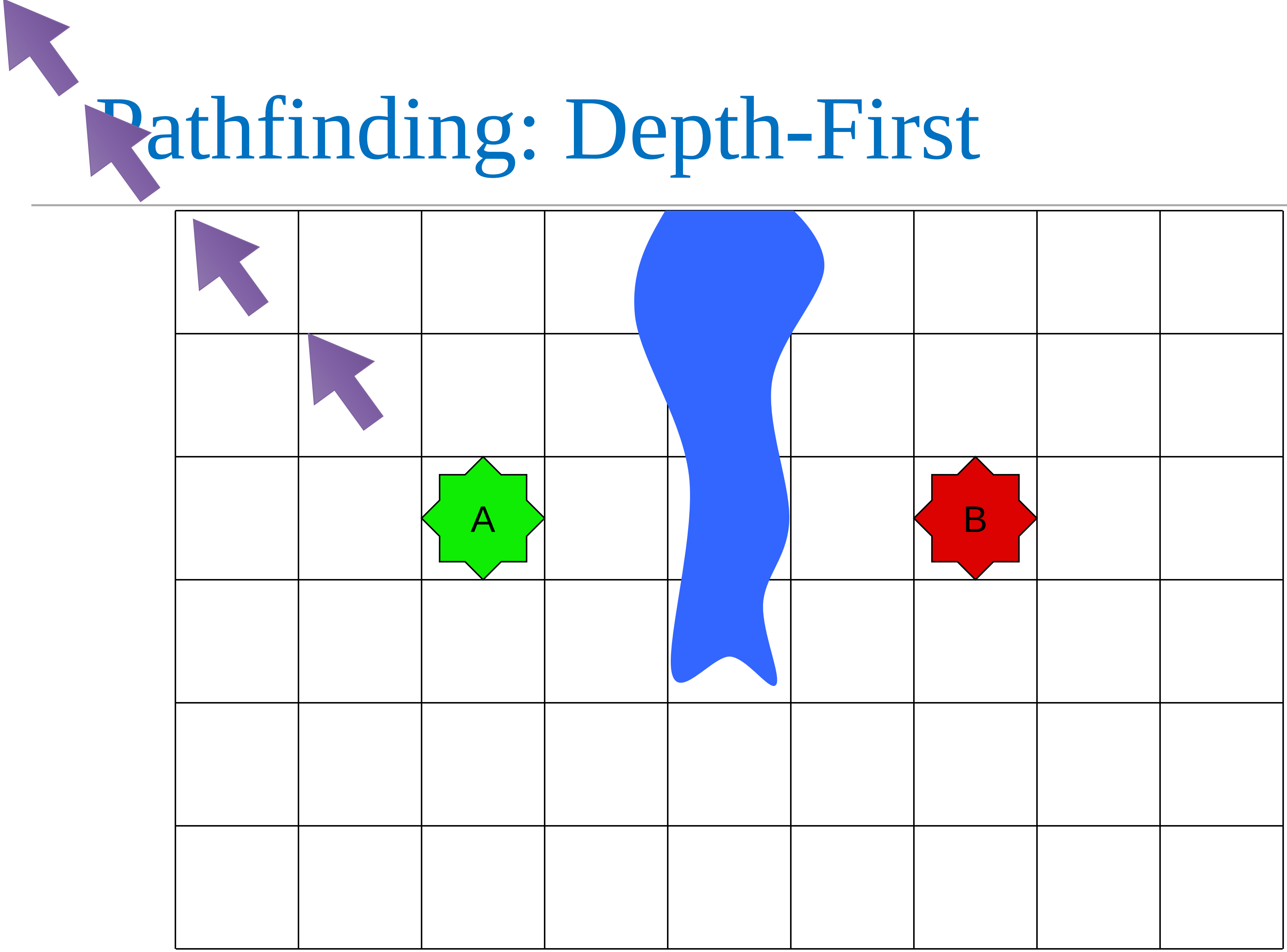
# Pathfinding



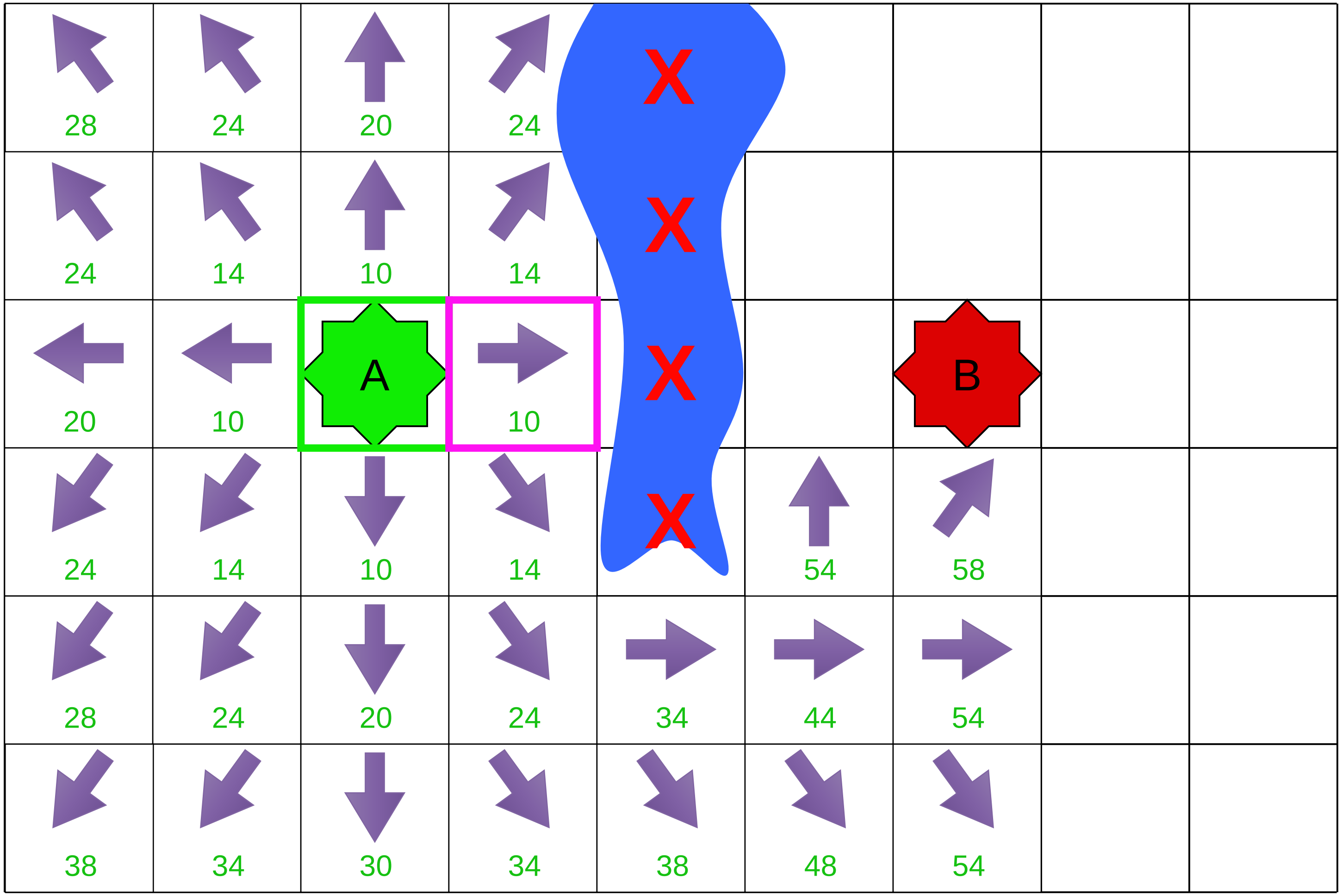
# Make a grid!



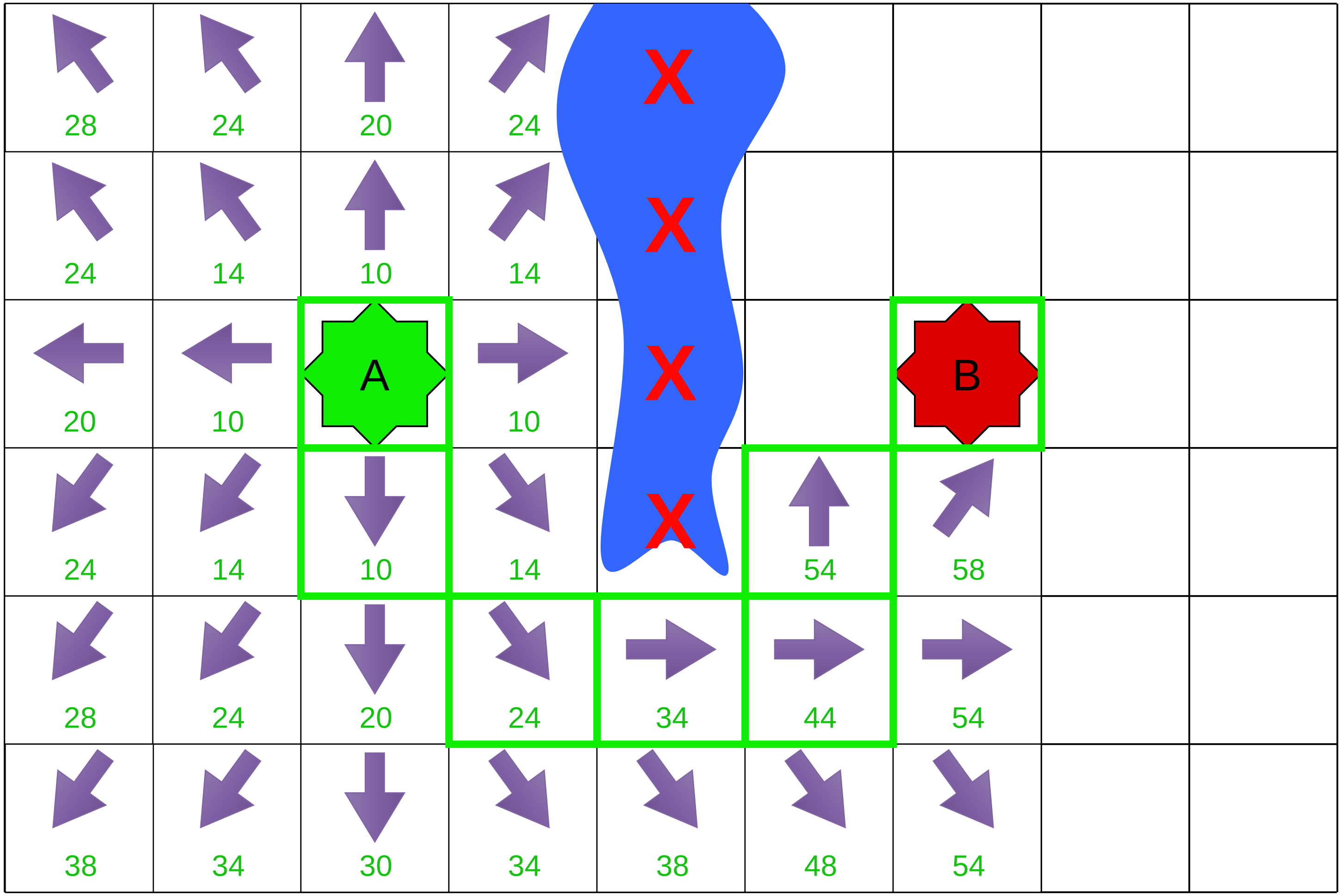
# Pathfinding: Depth-First



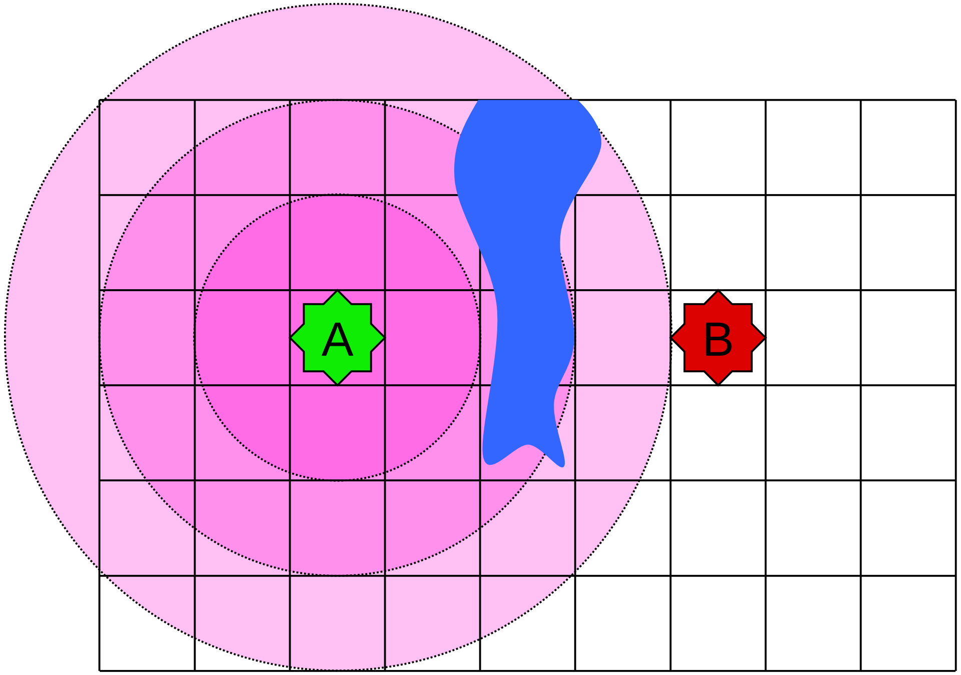
# Pathfinding: Breadth-First



# Pathfinding: Breadth-First



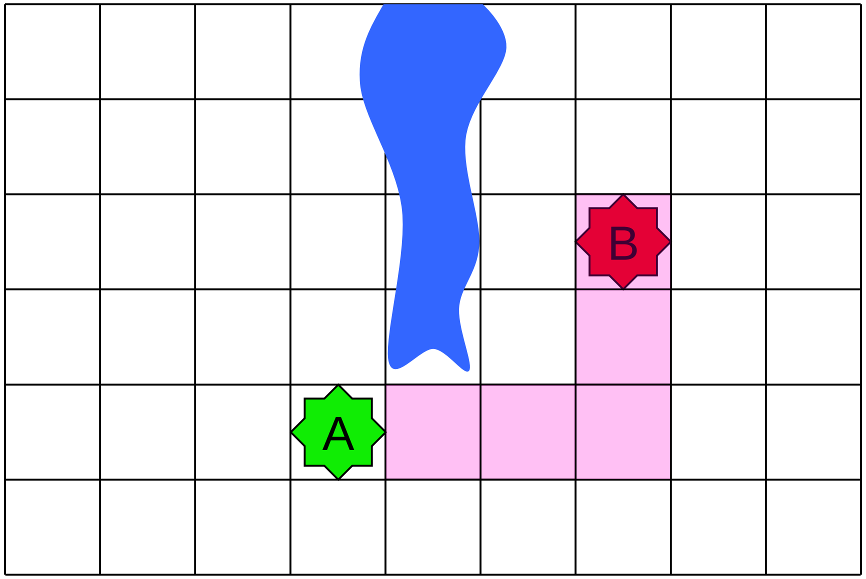
# Breadth-First is Slow!



**Idea:** use heuristics

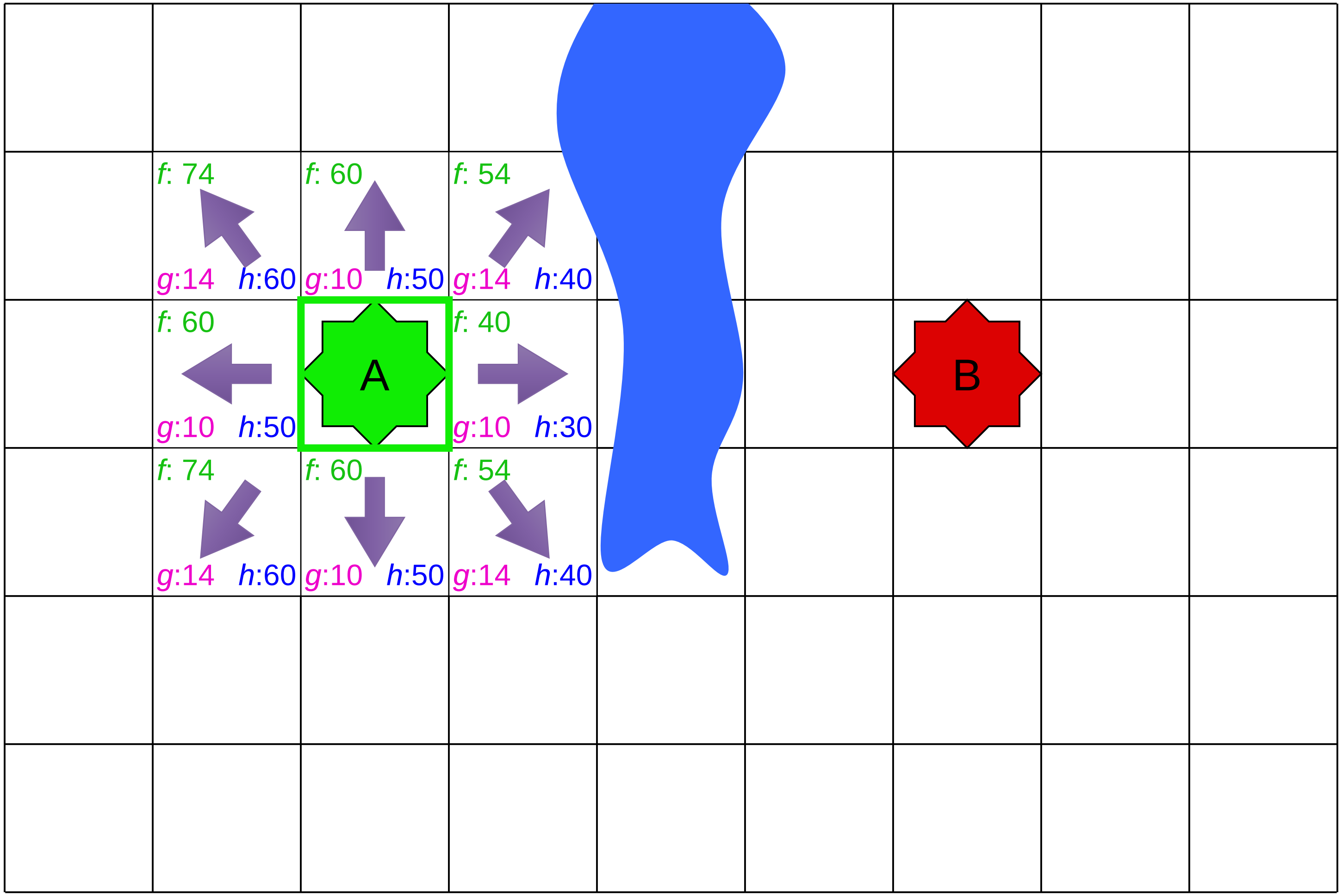
# A* Algorithm

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

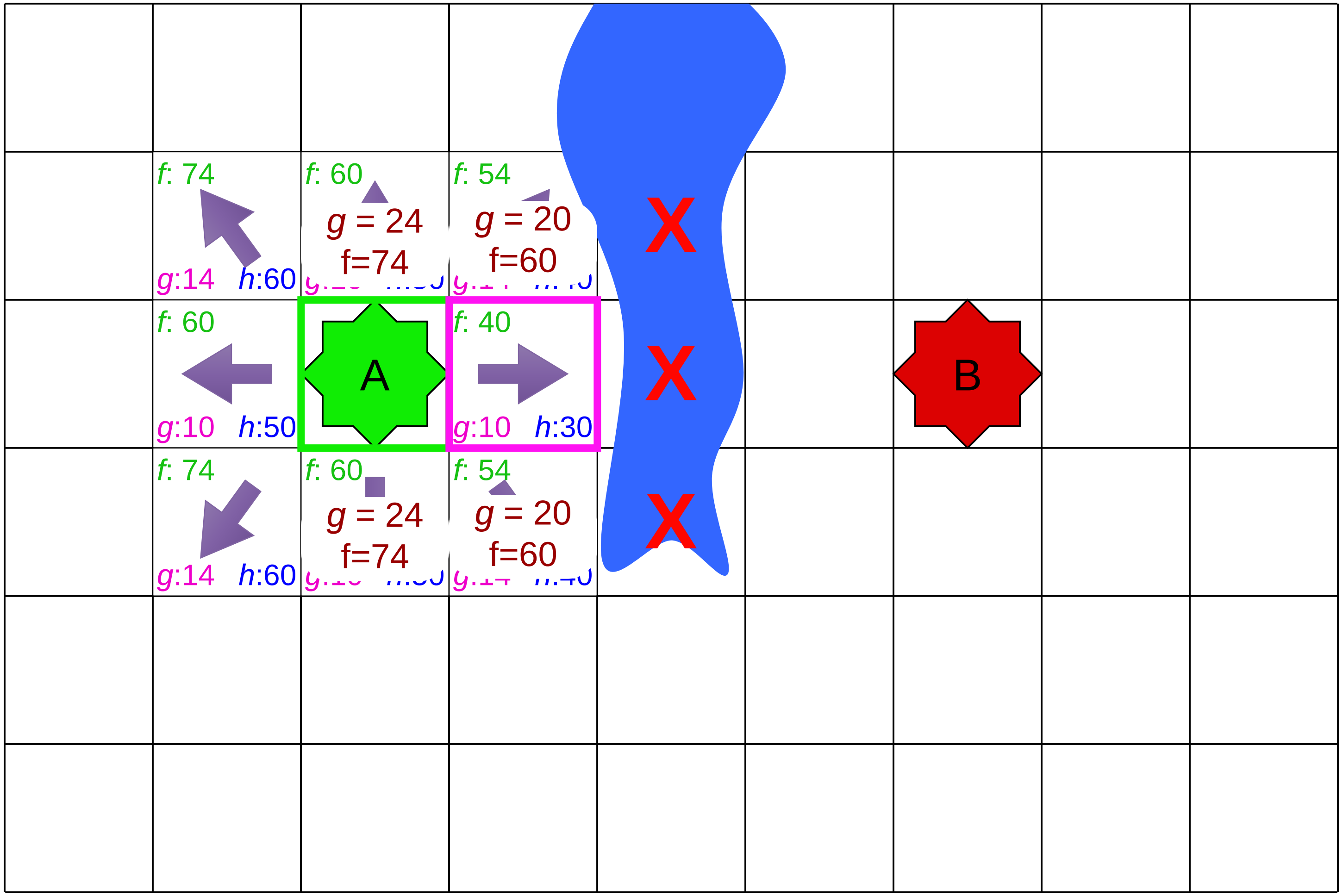


Manhattan distance = 30+20 = 50

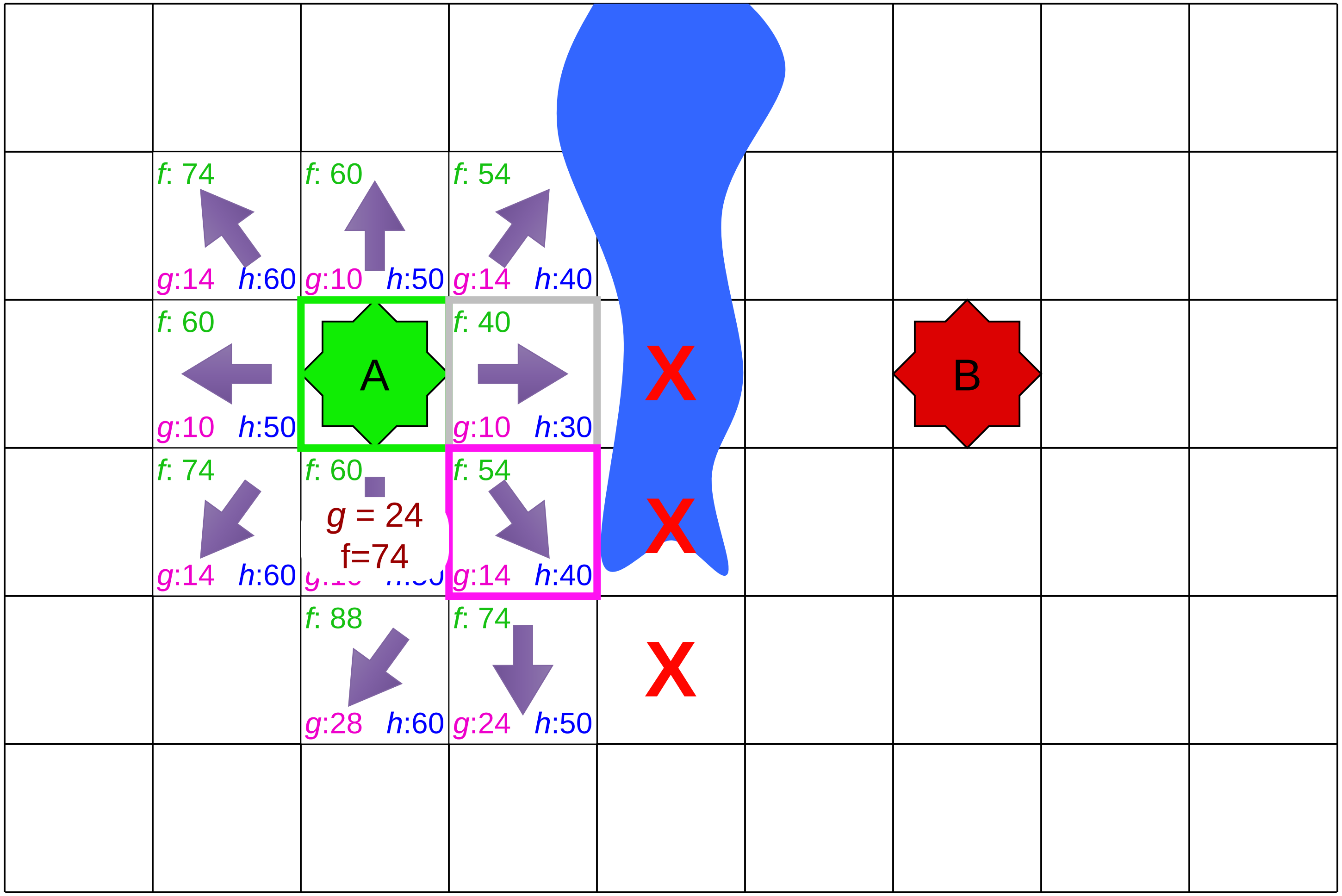
# Pathfinding: A* Algorithm



# Pathfinding: A* Algorithm



# Pathfinding: A* Algorithm



*f*: 74
*g*:14 *h*:60

*f*: 60
*g*:10 *h*:50

*f*: 54
*g*:14 *h*:40

*f*: 60
*g*:10 *h*:50

A

*f*: 40
*g*:10 *h*:30

X

B

*f*: 74
*g*:14 *h*:60

*f*: 60
*g* = 24
f=74

*f*: 54
*g*:14 *h*:40

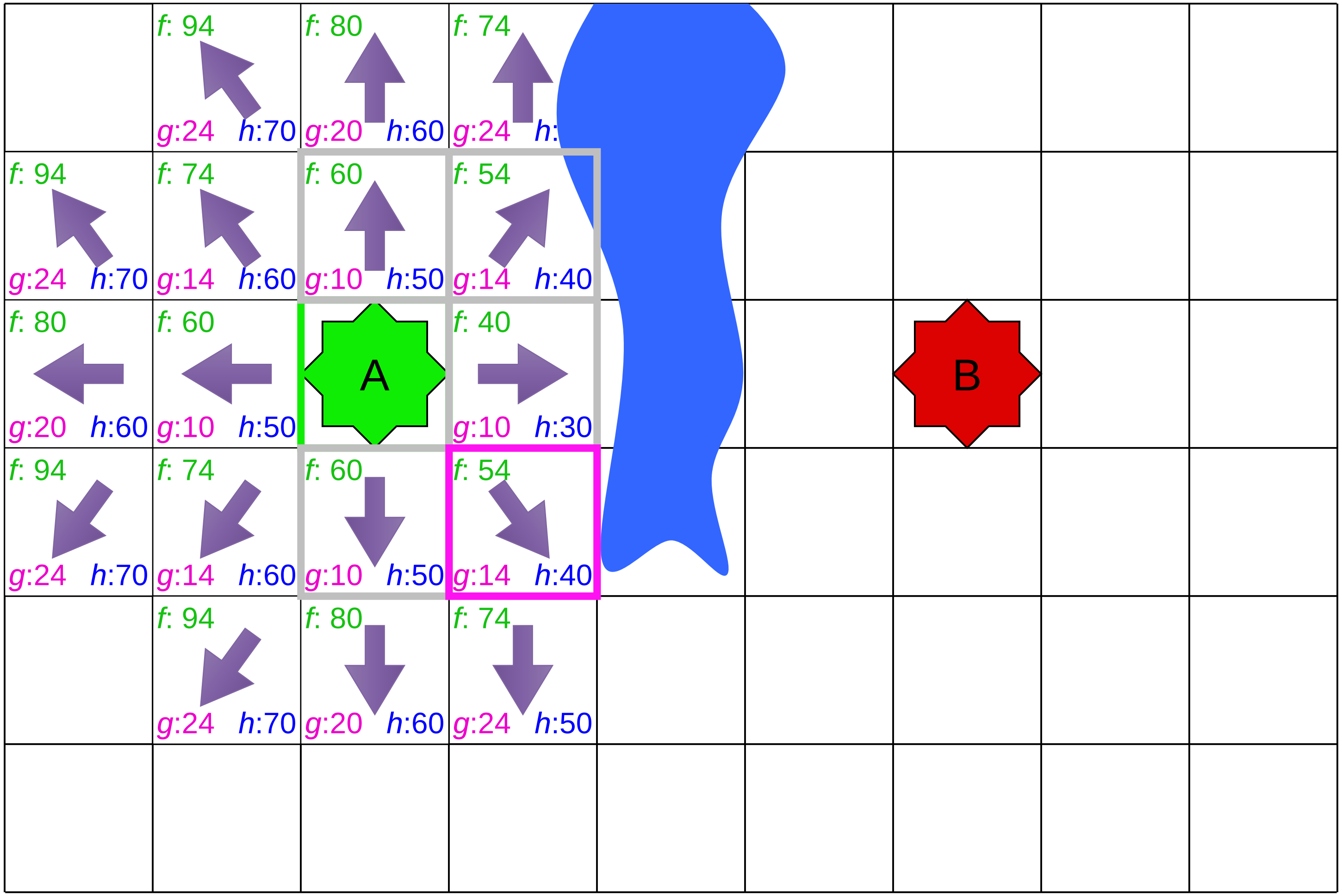
X

*f*: 88
*g*:28 *h*:60

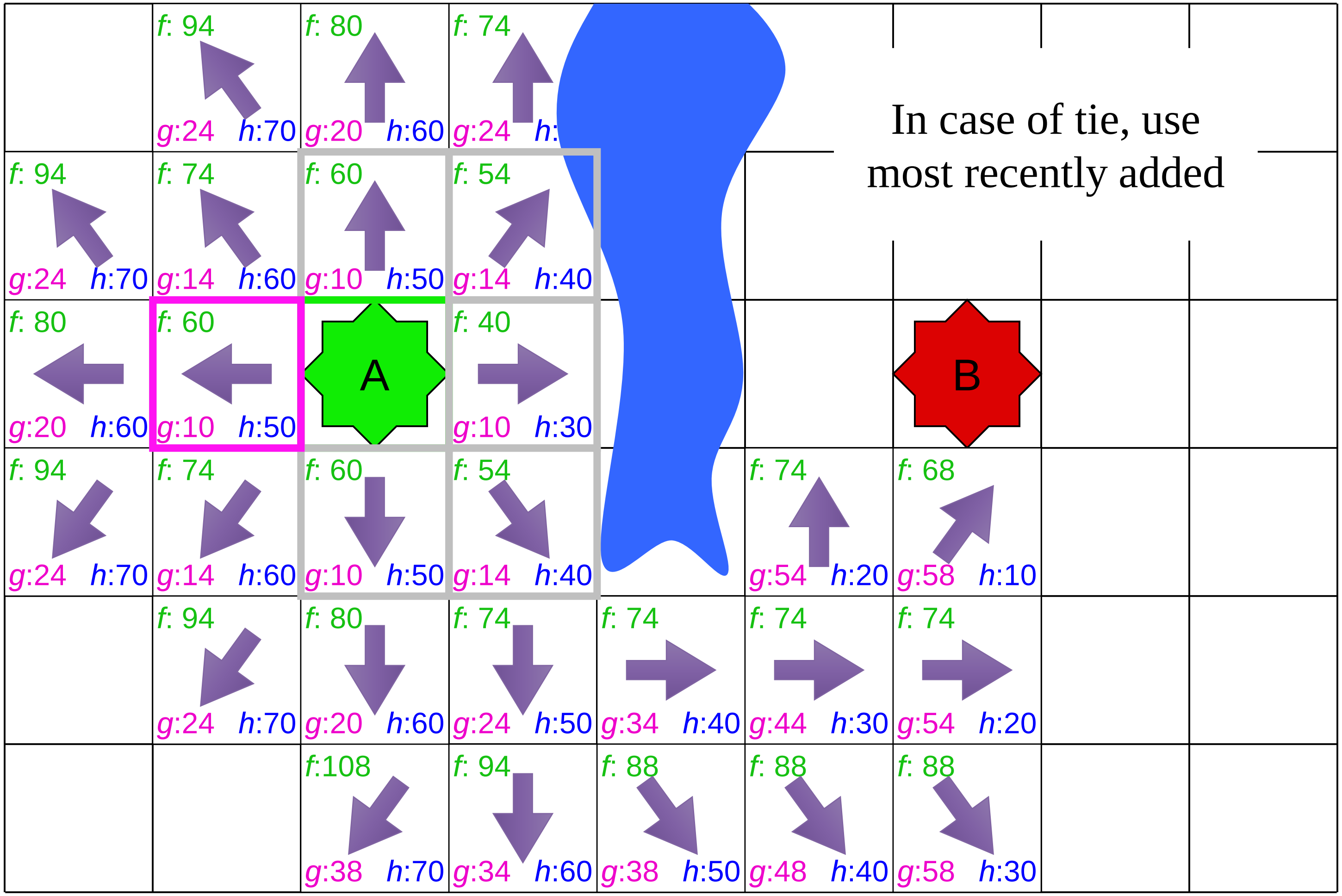
*f*: 74
*g*:24 *h*:50

X

# Pathfinding: A* Algorithm



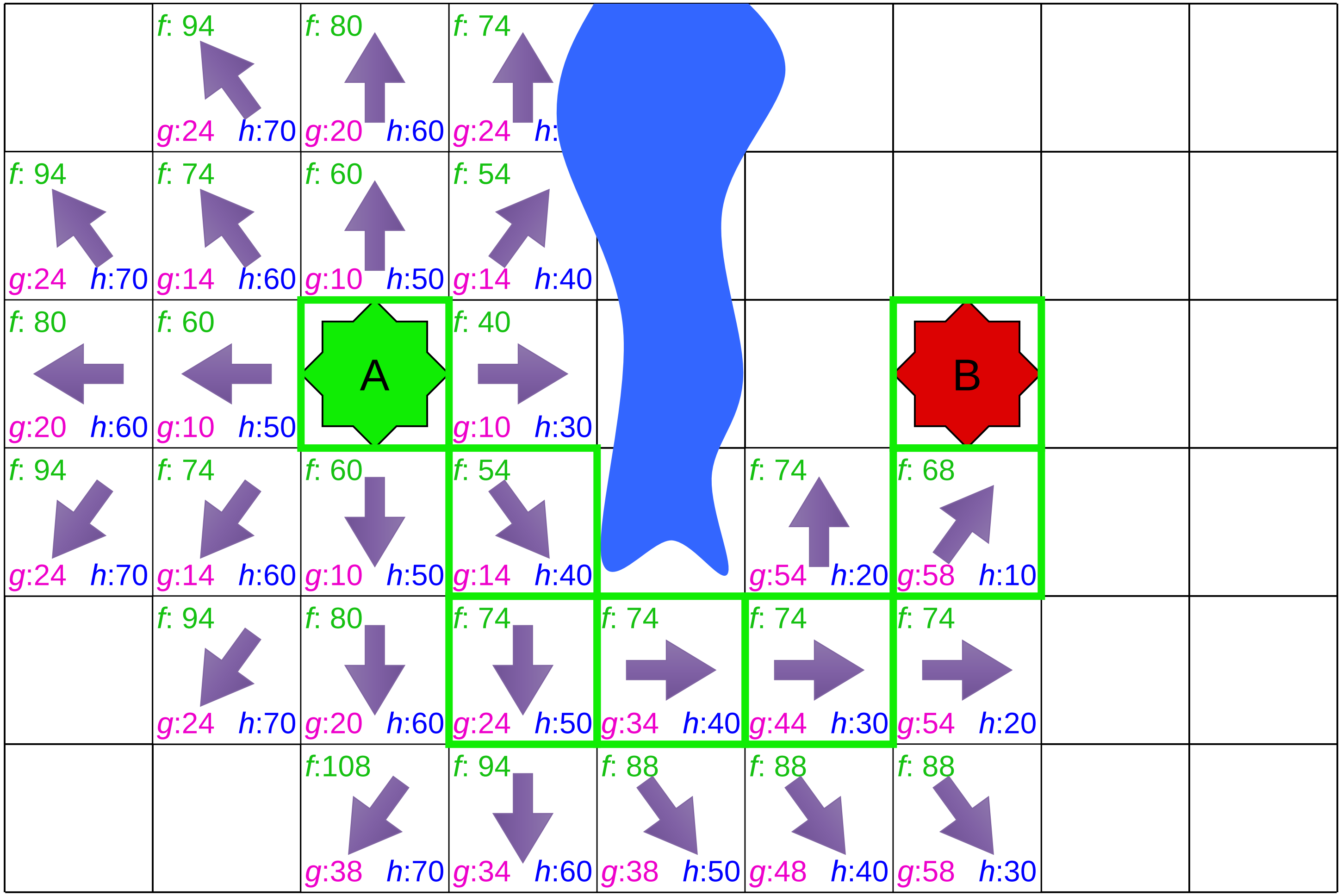
# Pathfinding: A* Algorithm



| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | f: 94<br>↖<br>g:24 h:70 | f: 80<br>↑<br>g:20 h:60 | f: 74<br>↑<br>g:24 h: | | | | | |
| f: 94<br>↖<br>g:24 h:70 | f: 74<br>↖<br>g:14 h:60 | f: 60<br>↑<br>g:10 h:50 | f: 54<br>↗<br>g:14 h:40 | | | | | |
| f: 80<br>←<br>g:20 h:60 | f: 60<br>←<br>g:10 h:50 | A | f: 40<br>→<br>g:10 h:30 | | | B | | |
| f: 94<br>↙<br>g:24 h:70 | f: 74<br>↙<br>g:14 h:60 | f: 60<br>↓<br>g:10 h:50 | f: 54<br>↘<br>g:14 h:40 | | f: 74<br>↑<br>g:54 h:20 | f: 68<br>↗<br>g:58 h:10 | | |
| | f: 94<br>↙<br>g:24 h:70 | f: 80<br>↓<br>g:20 h:60 | f: 74<br>↓<br>g:24 h:50 | f: 74<br>→<br>g:34 h:40 | f: 74<br>→<br>g:44 h:30 | f: 74<br>→<br>g:54 h:20 | | |
| | | f:108<br>↙<br>g:38 h:70 | f: 94<br>↓<br>g:34 h:60 | f: 88<br>↘<br>g:38 h:50 | f: 88<br>↘<br>g:48 h:40 | f: 88<br>↘<br>g:58 h:30 | | |

In case of tie, use
most recently added

# Pathfinding: A* Algorithm

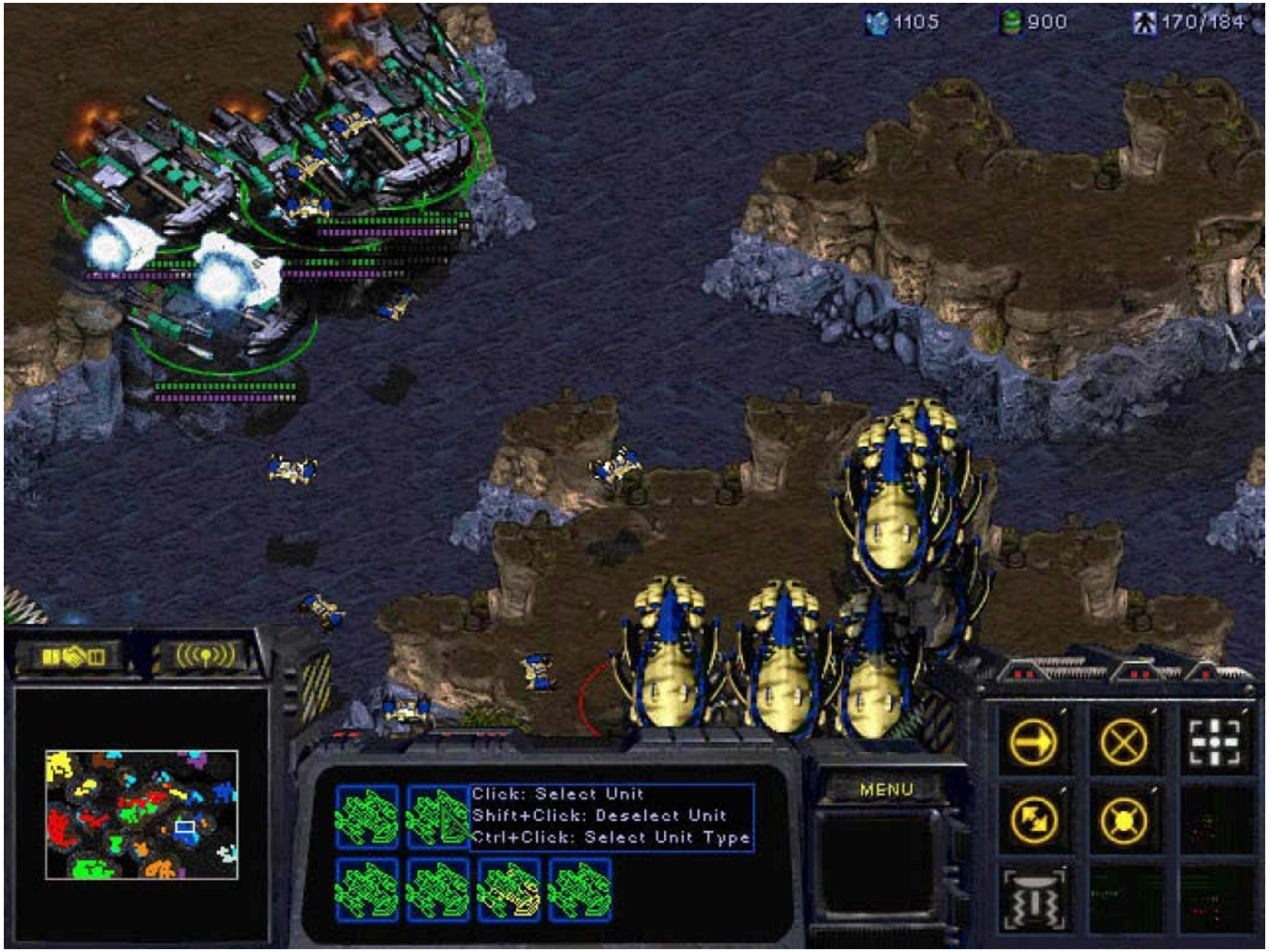


| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | f: 94<br>↖<br>g:24 h:70 | f: 80<br>↑<br>g:20 h:60 | f: 74<br>↑<br>g:24 h: | | | | | |
| f: 94<br>↖<br>g:24 h:70 | f: 74<br>↖<br>g:14 h:60 | f: 60<br>↑<br>g:10 h:50 | f: 54<br>↗<br>g:14 h:40 | | | | | |
| f: 80<br>←<br>g:20 h:60 | f: 60<br>←<br>g:10 h:50 | A | f: 40<br>→<br>g:10 h:30 | | | B | | |
| f: 94<br>↙<br>g:24 h:70 | f: 74<br>↙<br>g:14 h:60 | f: 60<br>↓<br>g:10 h:50 | f: 54<br>↘<br>g:14 h:40 | | f: 74<br>↑<br>g:54 h:20 | f: 68<br>↗<br>g:58 h:10 | | |
| | f: 94<br>↙<br>g:24 h:70 | f: 80<br>↓<br>g:20 h:60 | f: 74<br>↓<br>g:24 h:50 | f: 74<br>→<br>g:34 h:40 | f: 74<br>→<br>g:44 h:30 | f: 74<br>→<br>g:54 h:20 | | |
| | | f:108<br>↙<br>g:38 h:70 | f: 94<br>↓<br>g:34 h:60 | f: 88<br>↘<br>g:38 h:50 | f: 88<br>↘<br>g:48 h:40 | f: 88<br>↘<br>g:58 h:30 | | |

# A* Mario

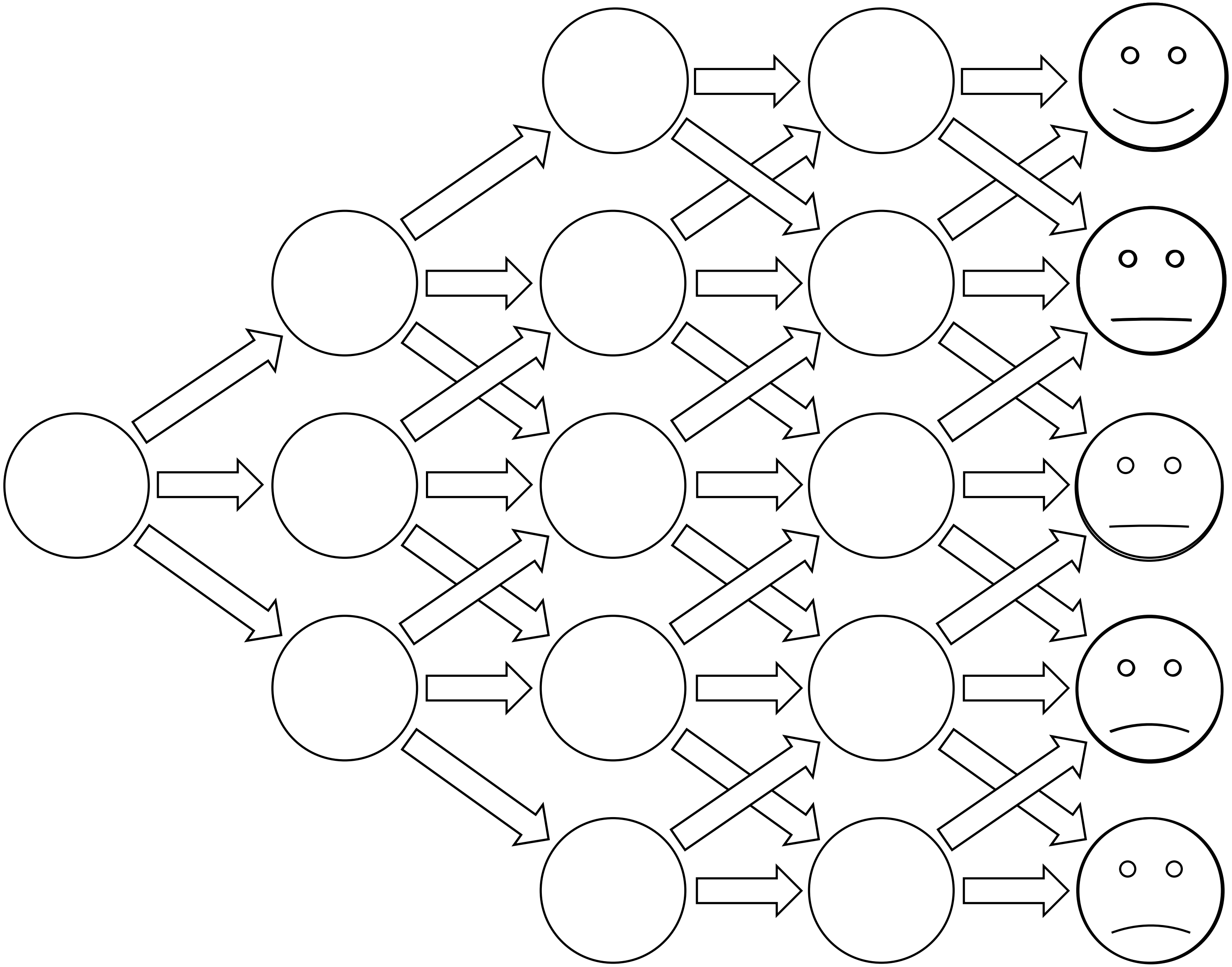
FPS: 24
Attempt: 1 of 1
AStarAgent
Selected Actions:

RIGHT SPEED

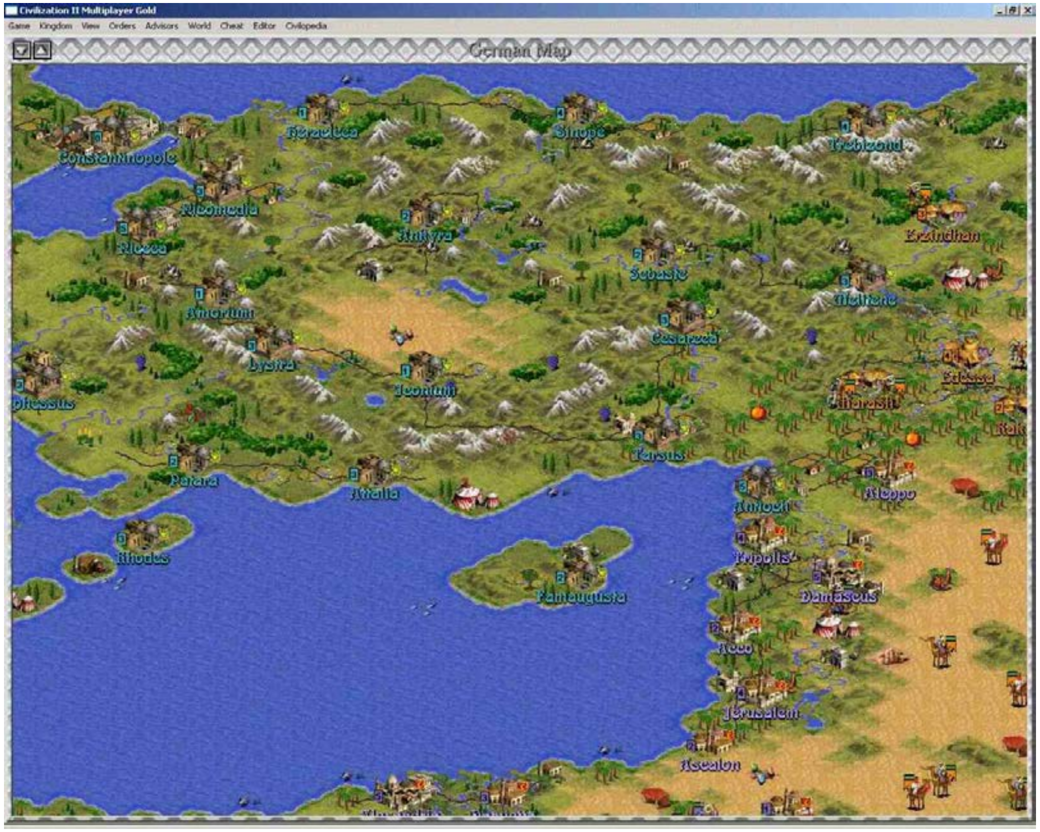


Click: Select Unit
Shift+Click: Deselect Unit
Ctrl+Click: Select Unit Type

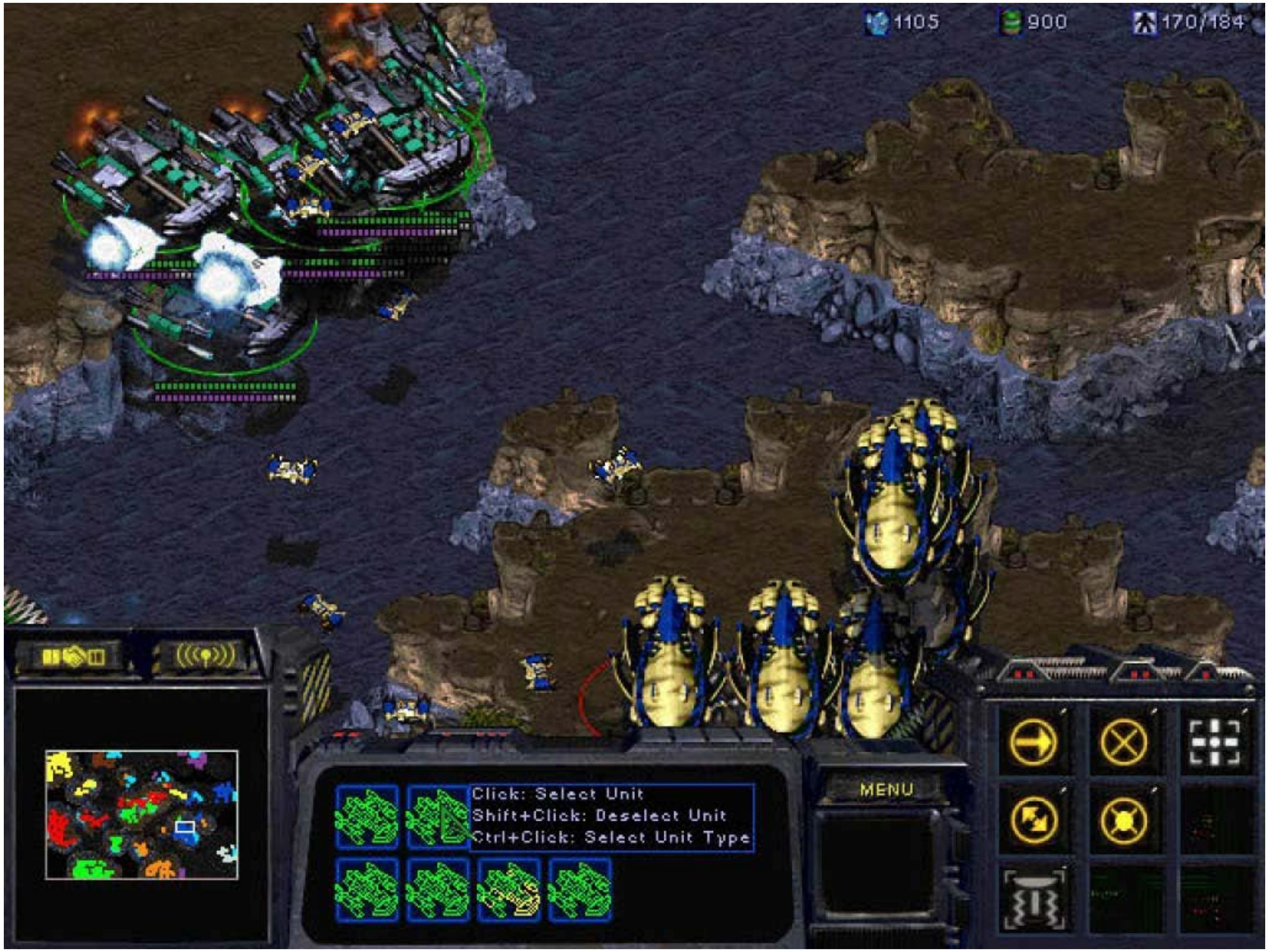
# Reinforcement learning



# Civilization II



Wins 78% of games!



Click: Select Unit
Shift+Click: Deselect Unit
Ctrl+Click: Select Unit Type

# Group Activity Choice

1. Think about AI and write some scripts
2. Discuss how an AI might solve your game
3. Just work on your games