COMP 3200
Artificial Intelligence

Lecture 2

Agents, Actions, and Environments

Note

- This is the most dense lecture of the course with definitions / terminology
- Don’t worry if you don’t understand everything completely as we go along
- Things will be re-iterated in later slides
- Use these slides as a reference for the rest of the course going forward

What is an Agent?

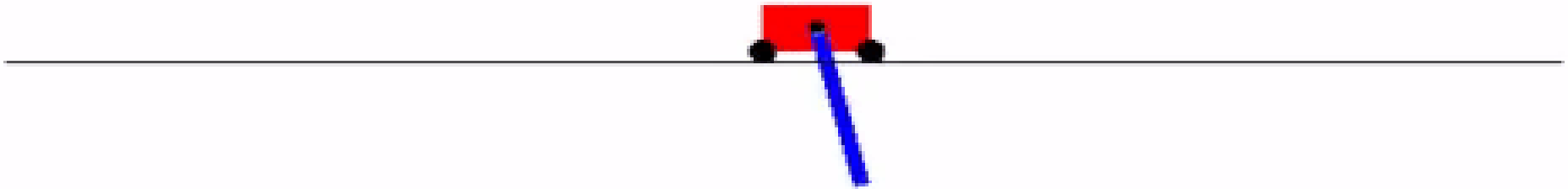
- An agent exists in a specific environment
- The agent is the entity for which we want to make decisions and take actions
- In most cases, the environment defines a problem, while the agent gives a solution

- There are MANY types of agents


Agent: Roomba


Agent: Spot
Boston Dynamics

Agent: Balance Cart




Chess.com
Carlsen Magnus
DUBAI 2014
1
Hikaru Nakamura
DUBAI 2014
Agent: Player

Agent Depends on Problem

Agents & Environments

- An agent within an environment will take actions to achieve a particular goal
- The agent perceives its environment somehow, and makes a decision about which action to take
- The agent takes it action(s), which changes the environment, and the process repeats
- Artificial Intelligence is the algorithmic discipline of making these decisions more intelligent

Agents & Environments

Agents

- Perceives its environment through Sensors
  - Robot: Camera, Laser, Sonar
  - Human: 5 Senses
  - Games: Keyboard Input / RAM / Game State
- Acts on its environment through Actuators
  - Robots: Wheels, Tracks, Arms
  - Humans: Limbs, Tools
  - Games: Predefined Rules / Actions
- May have some knowledge about environment

Agents

- Percept
  - Agent's perceptual inputs at any instant
  - "Agent's current belief of the world"
  - Environment's current State
- Percept Sequence
  - Complete history of everything the agent has ever perceived so far

Agents

- *In general*, an agent's choice of action at any given instant can depend on the entire percept sequence to date (what you did in past matters)
- In this course we will only look at problems where only the current state is sufficient to decide
  - Fully observable states, no state history required
- Agent's behaviour is given by the agent function that maps given percept sequence to an action
  - "Given what I have seen, take this action"


TSM
TL
3
3
37.7k
26:04
0:11
TL PigJet
13
3690
TSM Santorin
11
TSM Dyrus
13
TSM WildTurtle
13
TSM Bjergsen
14
TECH
INSIDER


REPLAY
TL Piglet
13
TSM WildTurtle
13
TSM Bjergsen
14
4
12
9

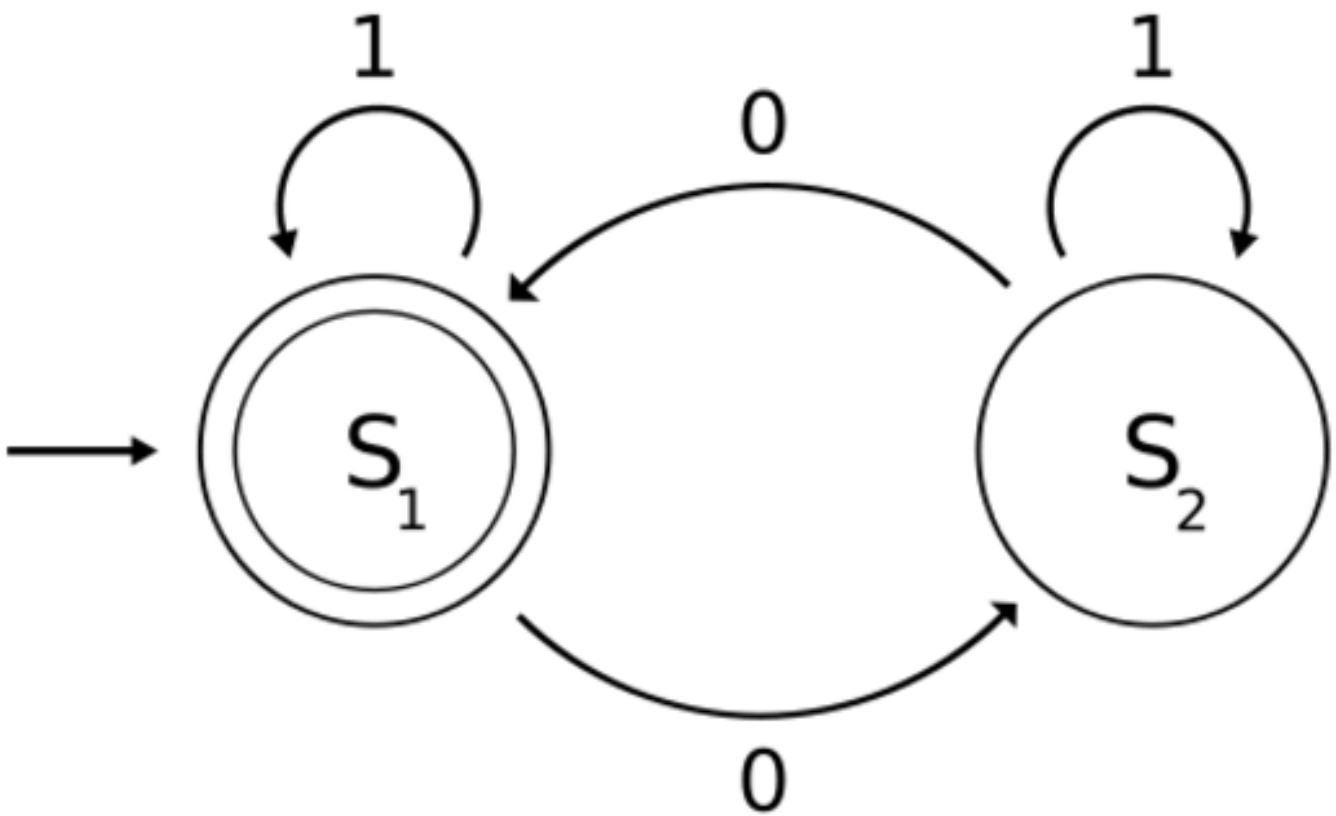
Action

- An action taken by an agent in a given state transitions the state to another

State Transition Table

State \ Input	1	0
S_1	S_1	S_2
S_2	S_2	S_1

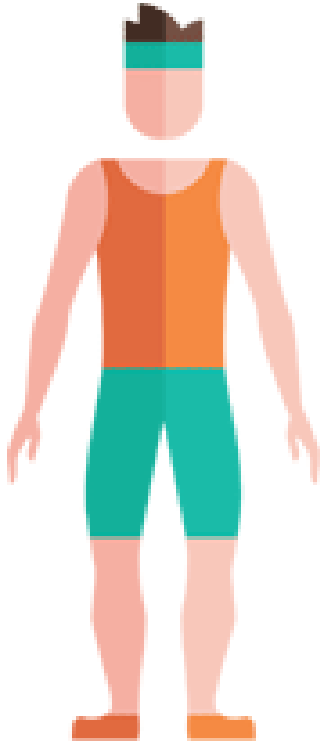
State Diagram



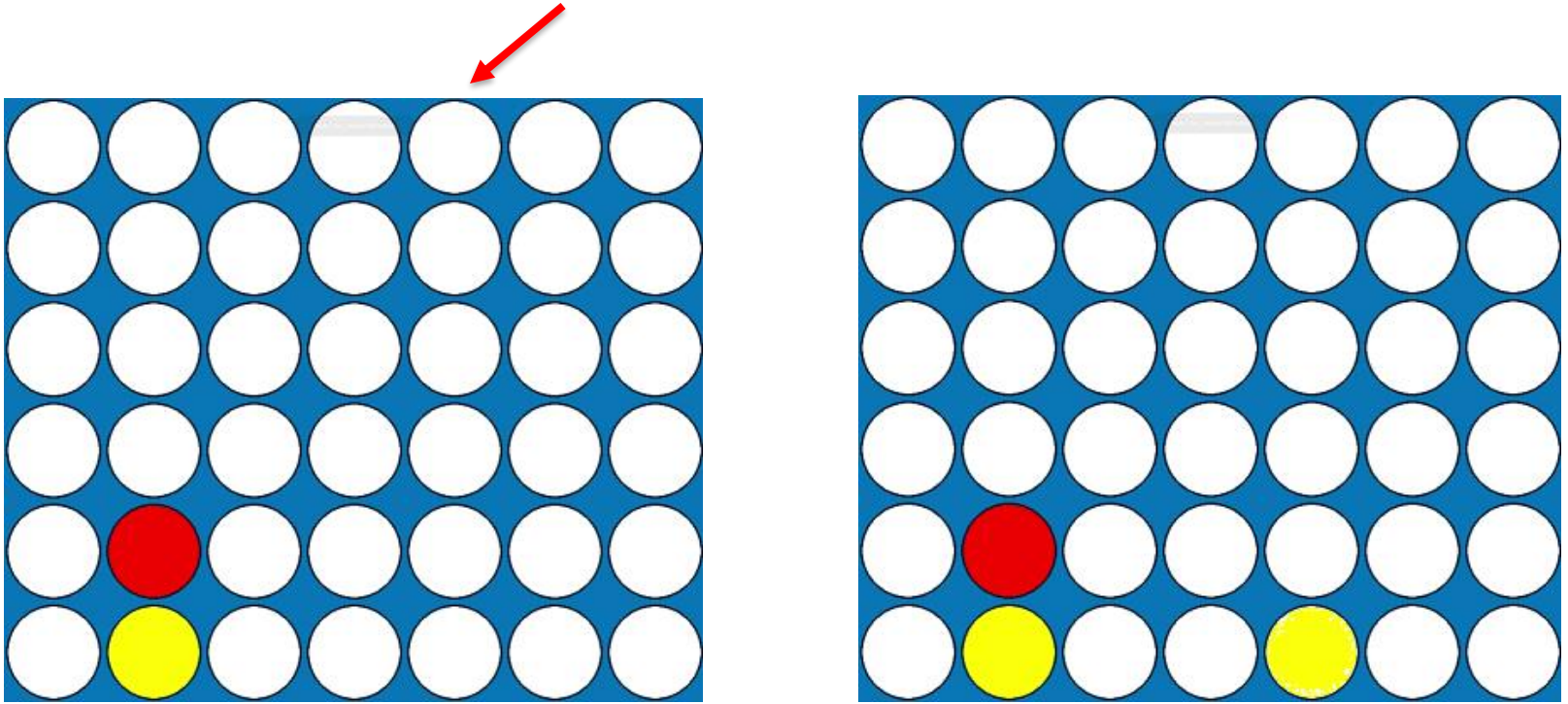
Action

- An action taken by an agent in a given state transitions the state to another
- State Transition Function (STF)
  - Successor Function, Table, Graph
- State S, Action A
- S’ = STF(S, A)

Action



Action Example




SUPER NINTENDO
SELECT
START
X
A
Y
B
Action


Action

Command List

Air

Crouch

For

Spiral Arrow :

FADC : Hold

Crouch

Crouch

For

Spiral Arrow :

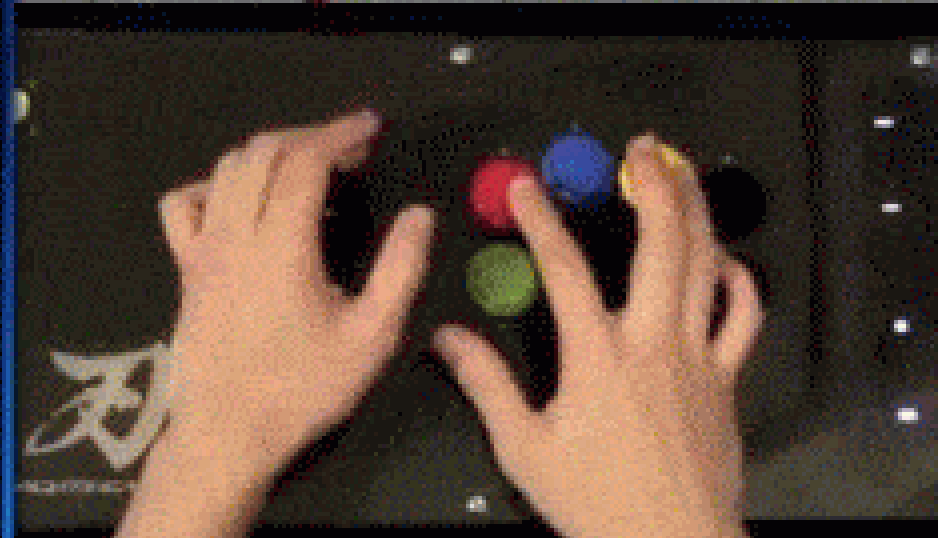
FADC : Hold

Crouch

Crouch

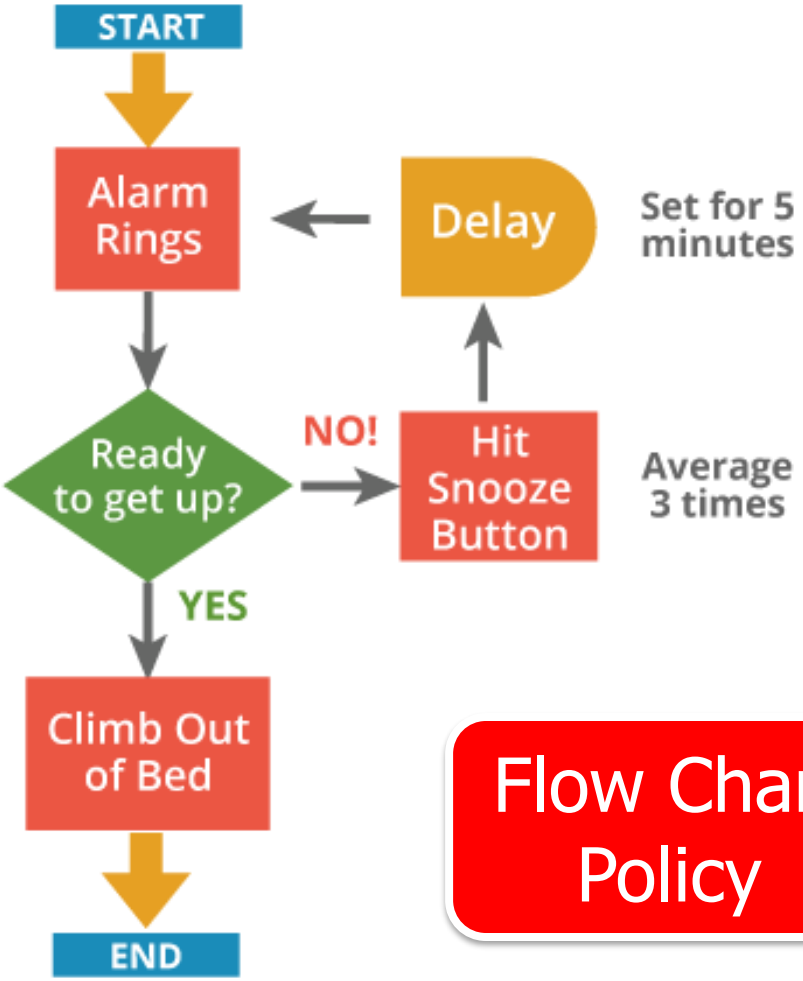
Action

FIGHTING EDGE X360

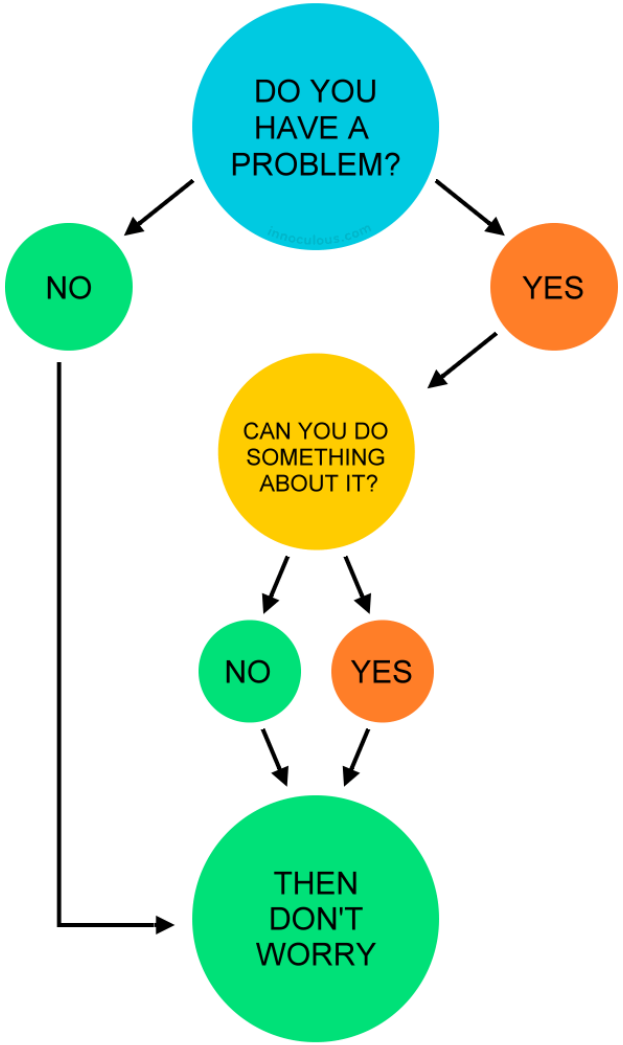


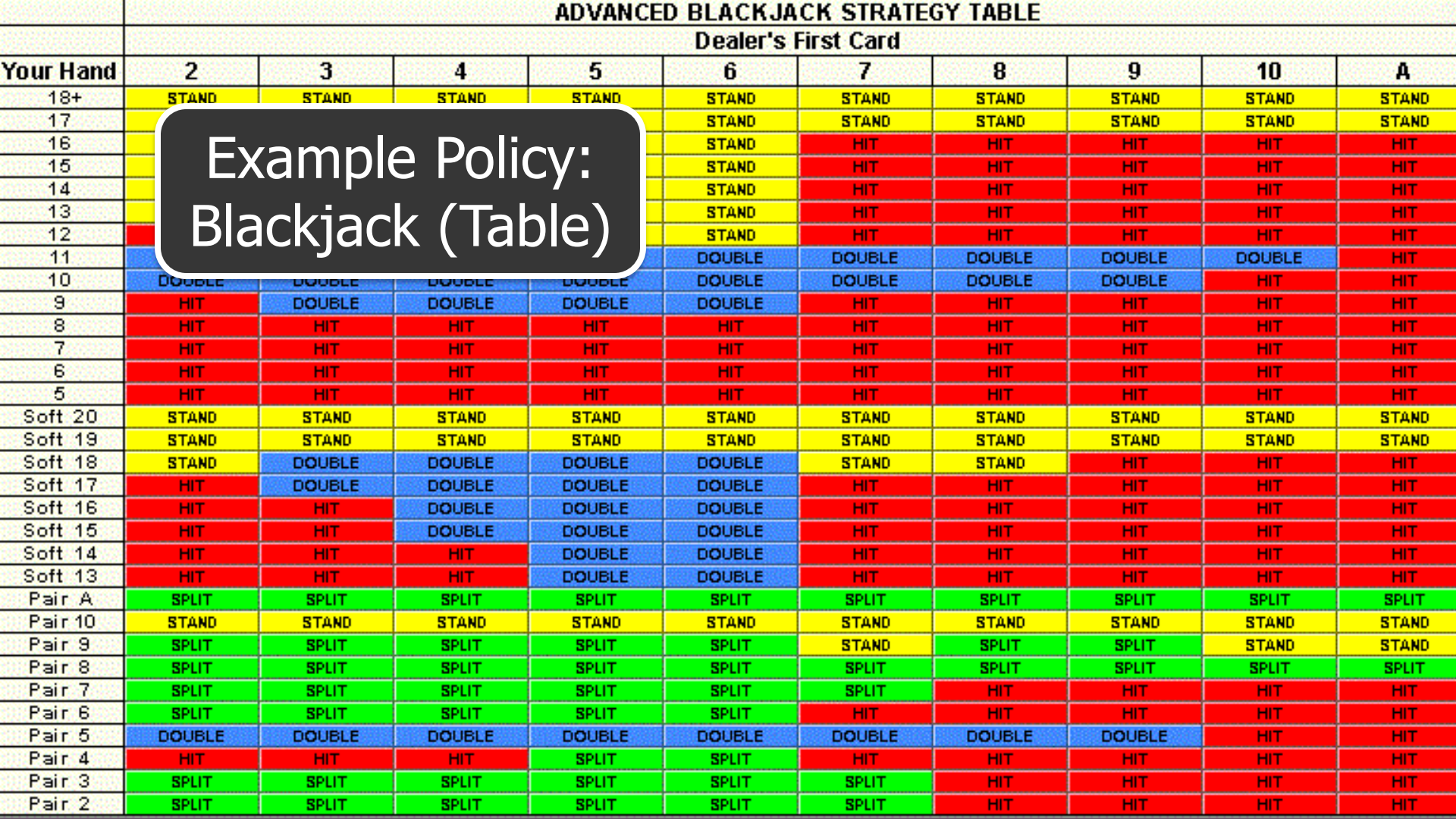
Policies

- Policy = Map from States to Actions
- Policy π, Optimal Policy π *
- π(s) = a
  - “Given we are in state S, do action A”
- Policy could be implemented as
  - Lookup Table
  - If-Statements
  - Arbitrary Function Calculation
  - Neural Network



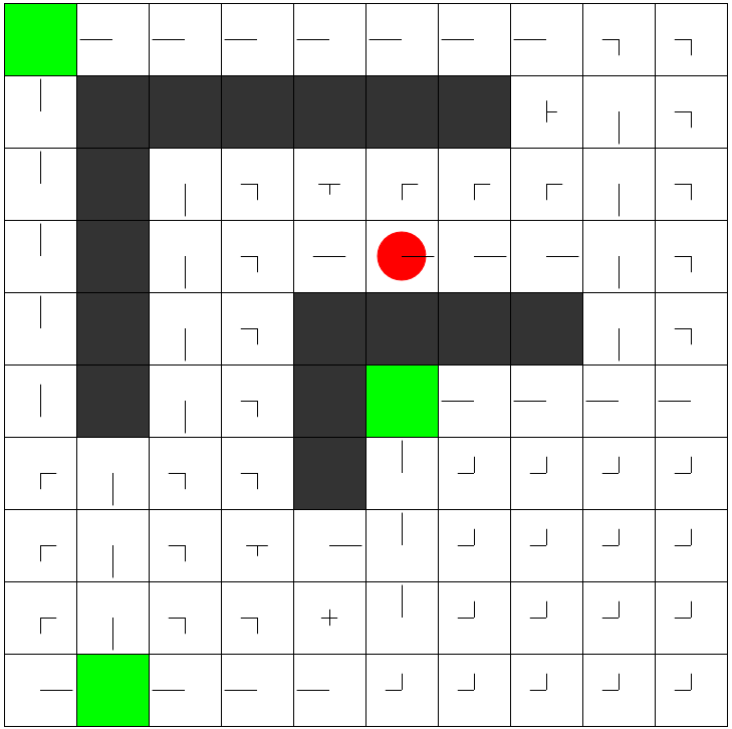
Flow Chart Policy





	ADVANCED BLACKJACK STRATEGY TABLE									
	Dealer's First Card									
Your Hand	2	3	4	5	6	7	8	9	10	A
18+	STAND	STAND	STAND	STAND	STAND	STAND	STAND	STAND	STAND	STAND
17					STAND	STAND	STAND	STAND	STAND	STAND
16					STAND	HIT	HIT	HIT	HIT	HIT
15					STAND	HIT	HIT	HIT	HIT	HIT
14					STAND	HIT	HIT	HIT	HIT	HIT
13					STAND	HIT	HIT	HIT	HIT	HIT
12					STAND	HIT	HIT	HIT	HIT	HIT
11					DOUBLE	DOUBLE	DOUBLE	DOUBLE	DOUBLE	HIT
10	DOUBLE	DOUBLE	DOUBLE	DOUBLE	DOUBLE	DOUBLE	DOUBLE	DOUBLE	HIT	HIT
9	HIT	DOUBLE	DOUBLE	DOUBLE	DOUBLE	HIT	HIT	HIT	HIT	HIT
8	HIT	HIT	HIT	HIT	HIT	HIT	HIT	HIT	HIT	HIT
7	HIT	HIT	HIT	HIT	HIT	HIT	HIT	HIT	HIT	HIT
6	HIT	HIT	HIT	HIT	HIT	HIT	HIT	HIT	HIT	HIT
5	HIT	HIT	HIT	HIT	HIT	HIT	HIT	HIT	HIT	HIT
Soft 20	STAND	STAND	STAND	STAND	STAND	STAND	STAND	STAND	STAND	STAND
Soft 19	STAND	STAND	STAND	STAND	STAND	STAND	STAND	STAND	STAND	STAND
Soft 18	STAND	DOUBLE	DOUBLE	DOUBLE	DOUBLE	STAND	STAND	HIT	HIT	HIT
Soft 17	HIT	DOUBLE	DOUBLE	DOUBLE	DOUBLE	HIT	HIT	HIT	HIT	HIT
Soft 16	HIT	HIT	DOUBLE	DOUBLE	DOUBLE	HIT	HIT	HIT	HIT	HIT
Soft 15	HIT	HIT	DOUBLE	DOUBLE	DOUBLE	HIT	HIT	HIT	HIT	HIT
Soft 14	HIT	HIT	HIT	DOUBLE	DOUBLE	HIT	HIT	HIT	HIT	HIT
Soft 13	HIT	HIT	HIT	DOUBLE	DOUBLE	HIT	HIT	HIT	HIT	HIT
Pair A	SPLIT	SPLIT	SPLIT	SPLIT	SPLIT	SPLIT	SPLIT	SPLIT	SPLIT	SPLIT
Pair 10	STAND	STAND	STAND	STAND	STAND	STAND	STAND	STAND	STAND	STAND
Pair 9	SPLIT	SPLIT	SPLIT	SPLIT	SPLIT	STAND	SPLIT	SPLIT	STAND	STAND
Pair 8	SPLIT	SPLIT	SPLIT	SPLIT	SPLIT	SPLIT	SPLIT	SPLIT	SPLIT	SPLIT
Pair 7	SPLIT	SPLIT	SPLIT	SPLIT	SPLIT	SPLIT	HIT	HIT	HIT	HIT
Pair 6	SPLIT	SPLIT	SPLIT	SPLIT	SPLIT	HIT	HIT	HIT	HIT	HIT
Pair 5	DOUBLE	DOUBLE	DOUBLE	DOUBLE	DOUBLE	DOUBLE	DOUBLE	DOUBLE	HIT	HIT
Pair 4	HIT	HIT	HIT	SPLIT	SPLIT	HIT	HIT	HIT	HIT	HIT
Pair 3	SPLIT	SPLIT	SPLIT	SPLIT	SPLIT	SPLIT	HIT	HIT	HIT	HIT
Pair 2	SPLIT	SPLIT	SPLIT	SPLIT	SPLIT	SPLIT	HIT	HIT	HIT	HIT

Example Policy: Grid Pathfinding

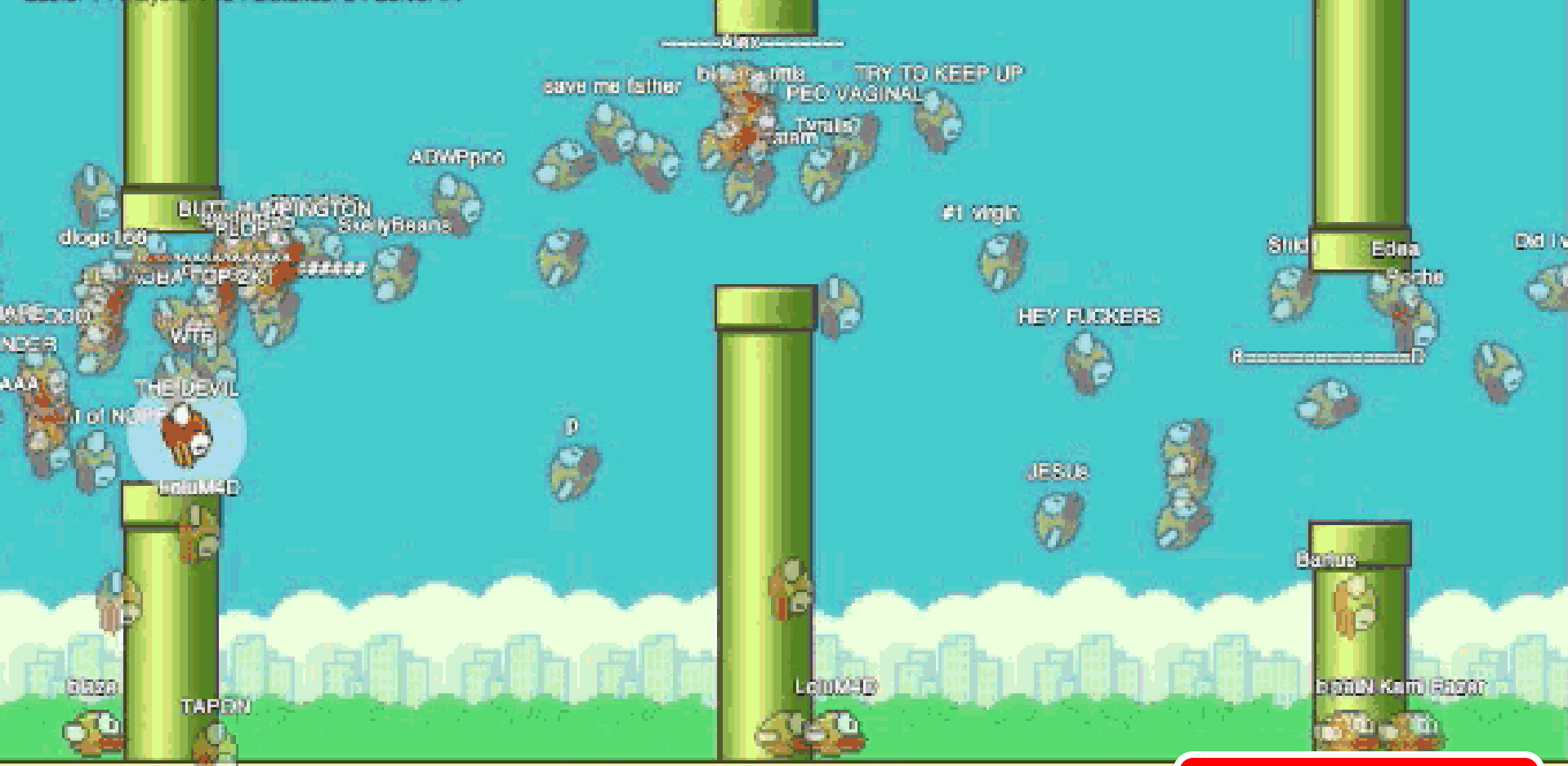


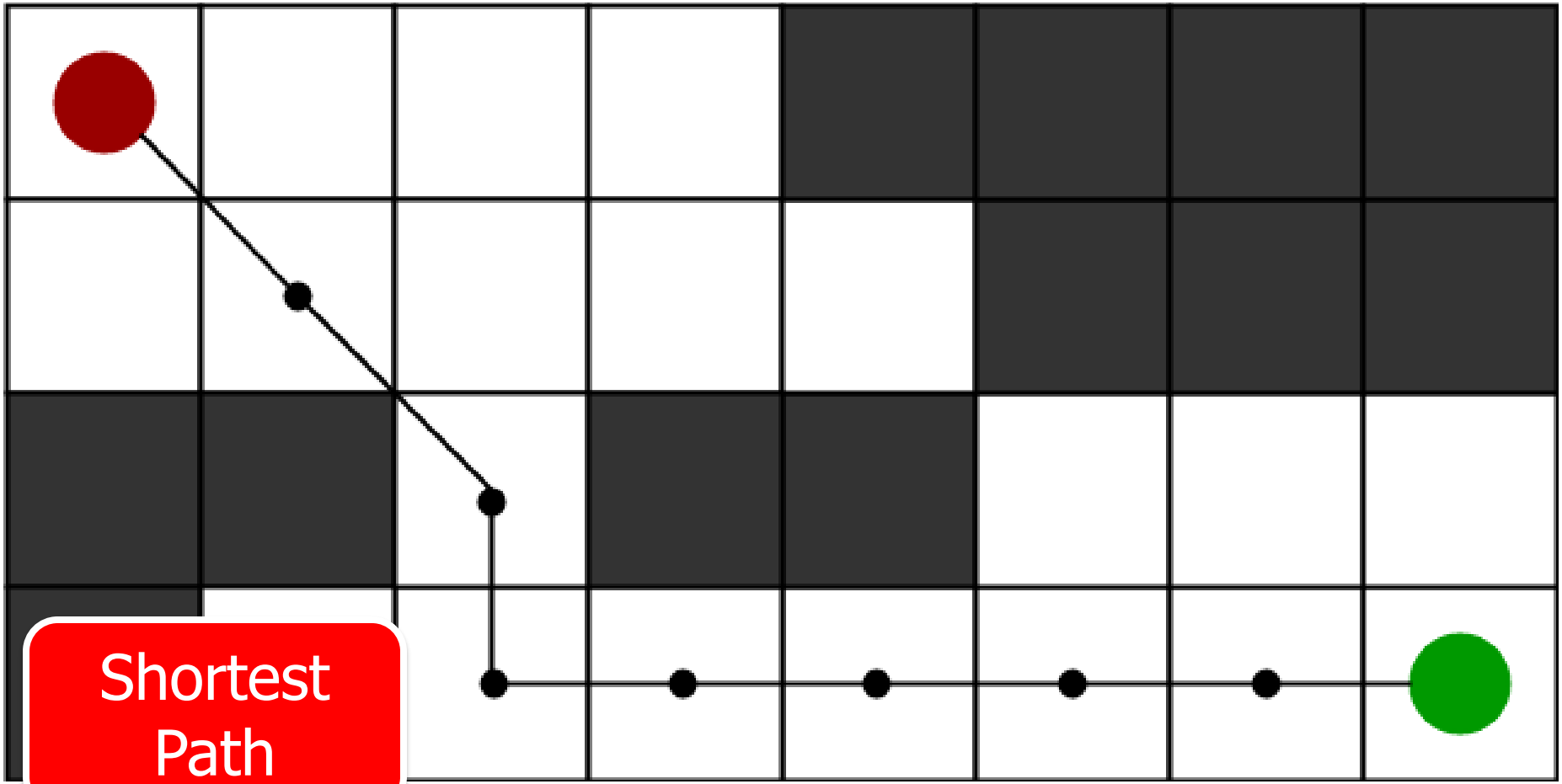
Rationality

- A rational agent is one that does the optimal action for every possible state
- What is the optimal action
  - Makes the agent ‘most successful’
- Must define a Performance Measure
  - Measures how successful the agent has been
- Rational agent has an Optimal Policy
  - Optimal within its level of knowledge

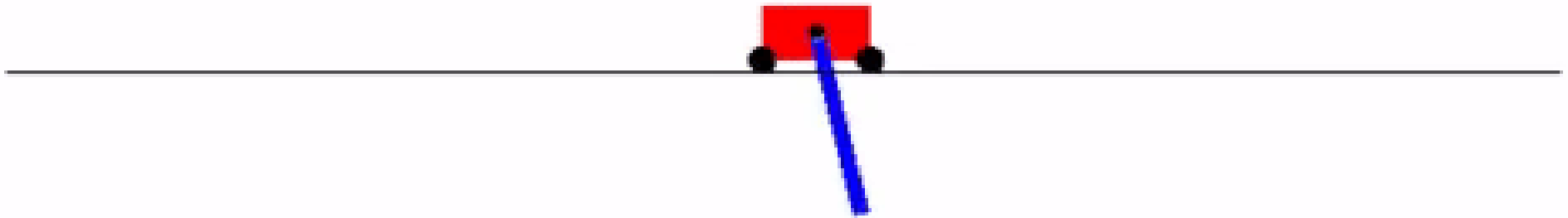
Performance Measure

- Criterion for success of an agent
- Can also be called an Evaluation Function
- Typical scenario
  1. Agent placed in an environment
  2. Agent performs sequence of actions
  3. Actions cause environment to go through a sequence of states
  4. If the sequence of states is desirable, the agent has performed well, the evaluation is higher


save me father
TRY TO KEEP UP
PEO VAGINAL
ADWPpro
#1 virgin
HEY FUCKERS
JESUS
THE DEVIL
TAPPY
Barius
Move Right


Shortest
Path

Pole Balance Time



Performance Measure

Win the Game

Rationality

- Rationality depends on 4 things:
  - The performance measure that defines success
  - The agent’s prior knowledge of the environment
  - The actions the agent can perform
  - The agent’s percept sequence

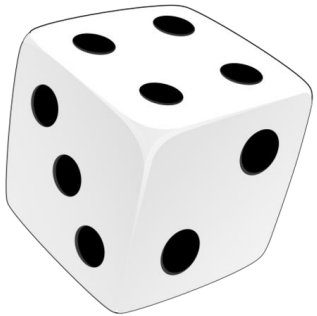
Expected Value

- Consider random variable X
  - Possible outcomes $x_1, x_2, \ldots, x_n$
  - With probabilities $p_1, p_2, \ldots, p_n$

- E(X) = Expected value of x
  $= x_1p_1 + x_2p_2 + \ldots + x_np_n$

Expected Value Example

- Rolling Dice
- 6 sides, numbers 1-6
- 1/6 probability each



$$\mathrm{E}[X] = 1 \cdot \frac{1}{6} + 2 \cdot \frac{1}{6} + 3 \cdot \frac{1}{6} + 4 \cdot \frac{1}{6} + 5 \cdot \frac{1}{6} + 6 \cdot \frac{1}{6} = 3.5$$

- Rational agents usually base decisions on expected (on average) values

Omniscience

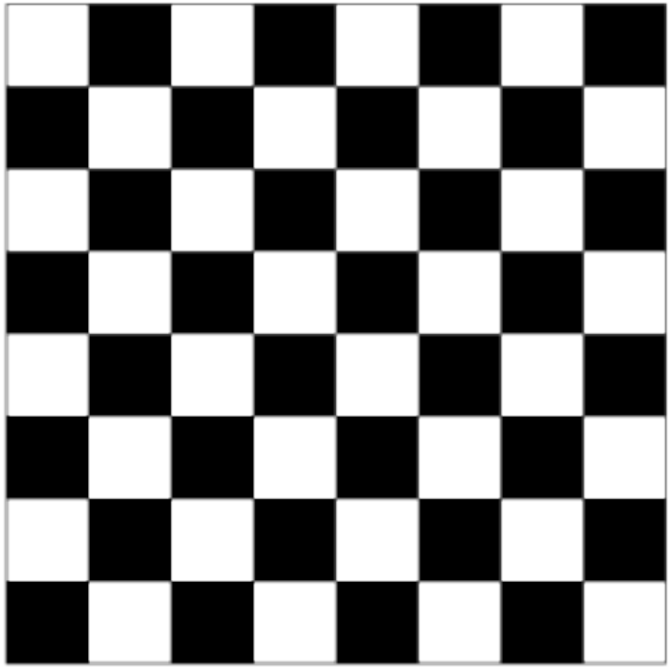
- An omniscient agent:
  - Knows the exact outcome of actions
  - Knows the outcome of ‘randomness’
  - Knows the entire state of the environment
- Rationality is not Omniscience
  - Rational agent says outcome of a dice roll will be 3.5 on average, and bases decisions on that
  - An omniscient agent knows the next dice roll, so would never have to estimate the outcome of events

Environments

Environments

- Task Environment = Problem
- Rational Agent = Solution
- The specification and properties of an environment greatly impact the design of a rational agent to act within it
- A State is a particular configuration of a given environment

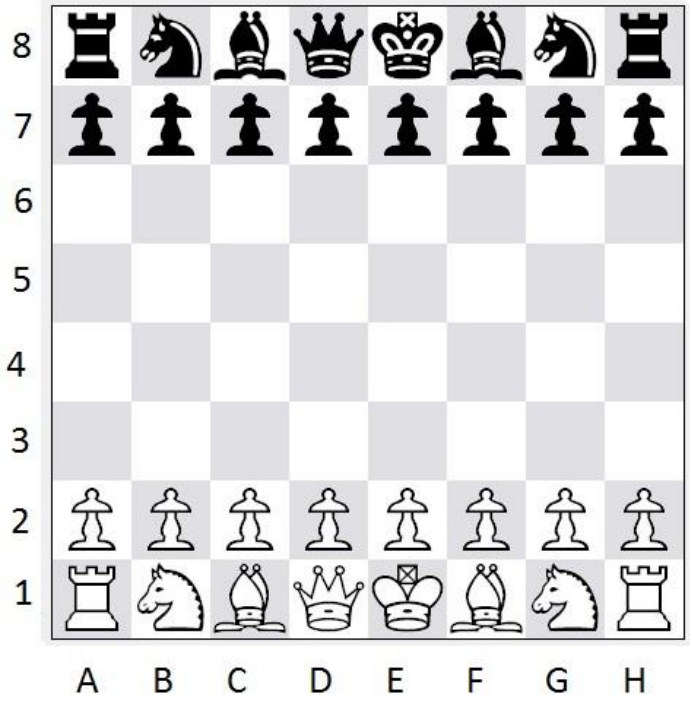
Example Environment: Chess



Environment

- Rules / Boundaries
- Legal Actions
- Starting State
- Goal Condition / State

Example State: Chess

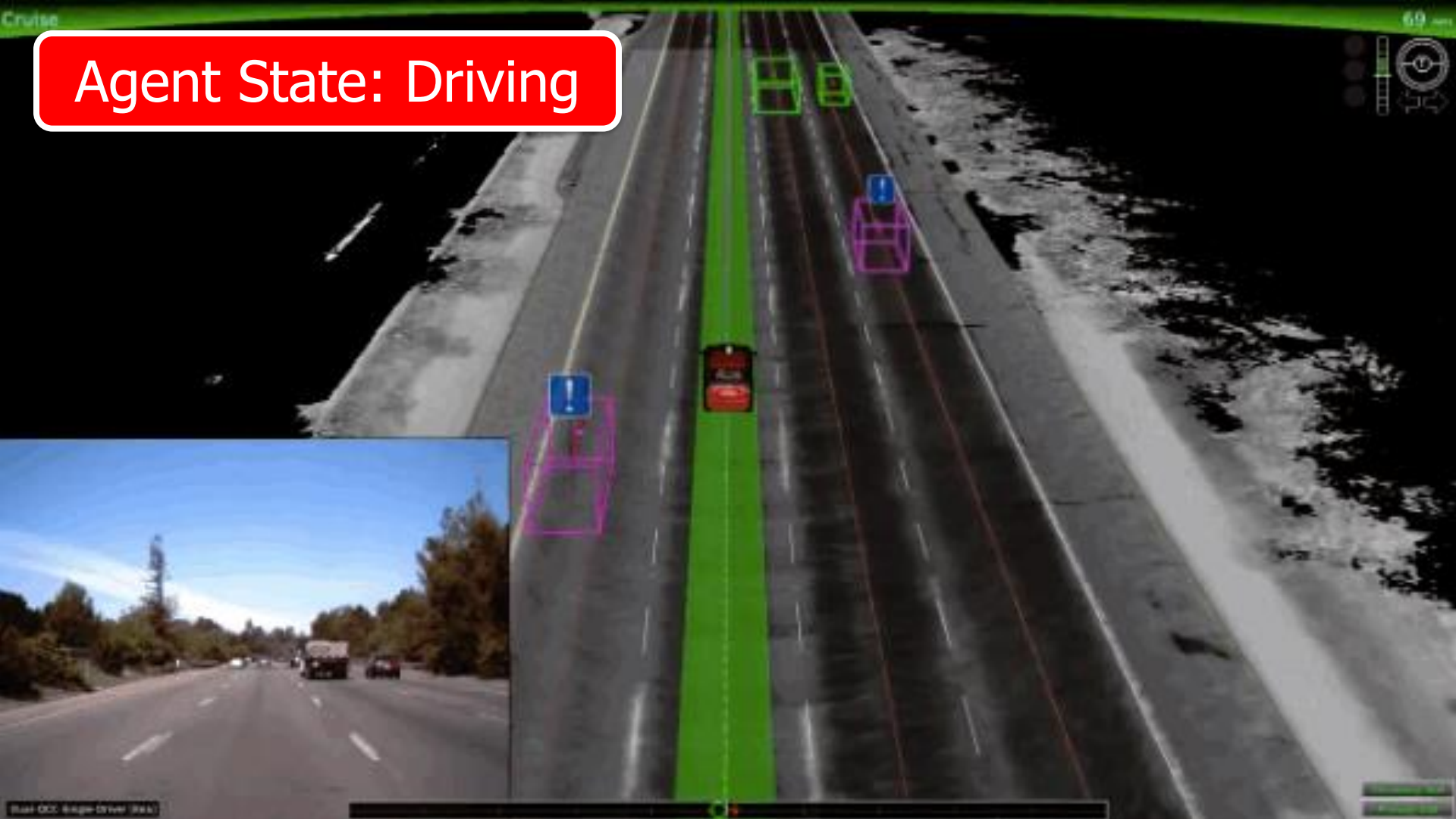


Agent Observation / State

- For the agent, the state of the environment is based on an observation it can make
- That observation may not be a complete representation of the environment state
- The agent’s observation (state) is all that it can use to make a decision

Example State: Chess


Environment: Driving


Agent State: Driving

State / Action Space

- State Space
  - Number of possible configurations of the environment for a given problem
- Action Space
  - Number of actions possible from a state
  - Given as average or worst-case
- Used as an estimation of complexity

Source: https://www.pipmodern.com/post/complexity-state-space-game-tree

Tic-Tac-Toe

State-Space Complexity Calculation

Chess

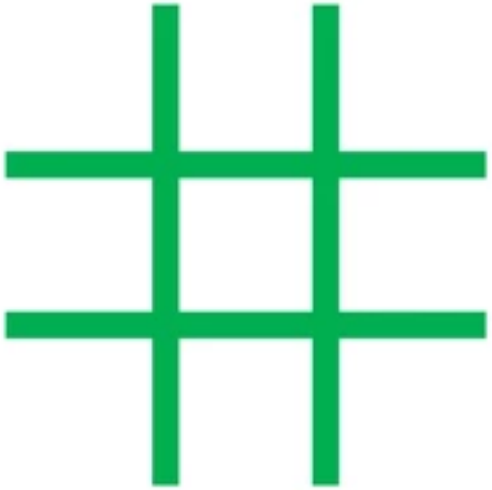
State-Space Complexity Calculation

Go

State-Space Complexity Calculation

STATE SPACE COMPLEXITY

TIC-TAC-TOE



4

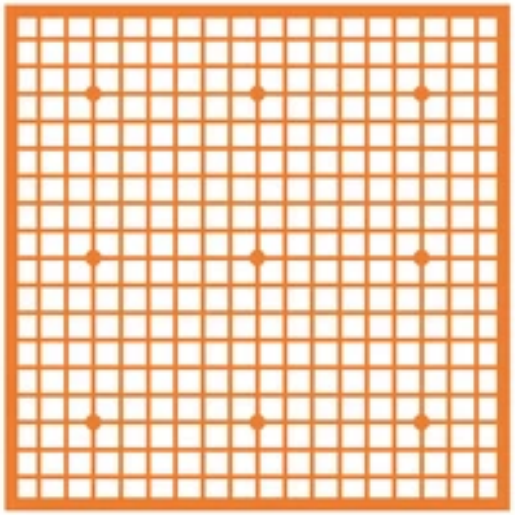
(TEN THOUSAND)

CHESS

43

(TEN TREDECILLION)

GO

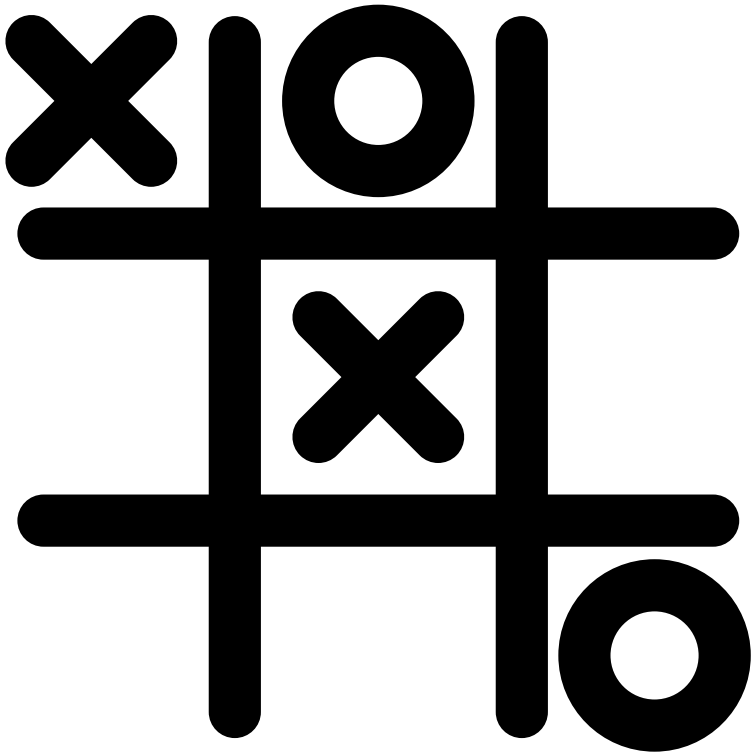


172

(TEN SEXQUINQUAGINTILLION)

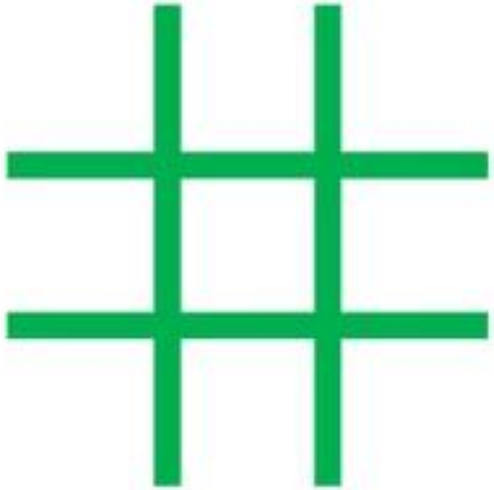
Game Tree Complexity

- State Space Complexity
  - The number of configurations of env
- Game Tree Complexity
  - The number of ways to take a sequence of actions through the game
  - Some states may be able to be reached via many different sequences



GAME-TREE COMPLEXITY

TIC-TAC-TOE



5

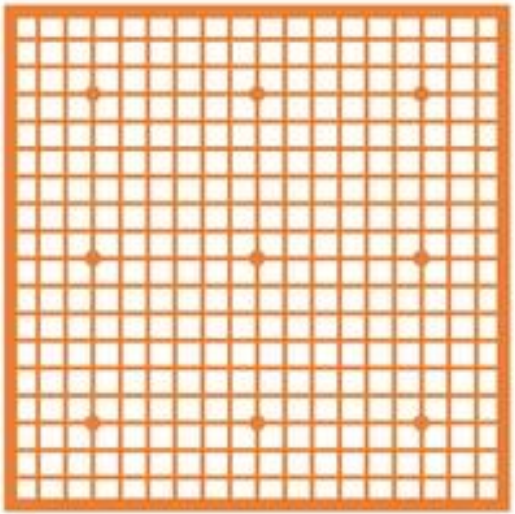
(ONE HUNDRED THOUSAND)

CHESS

123

(ONE QUADRAGINTILLION)

GO



360

(ONE CENNOVEMDECILLION)

9: Seconds in a Century

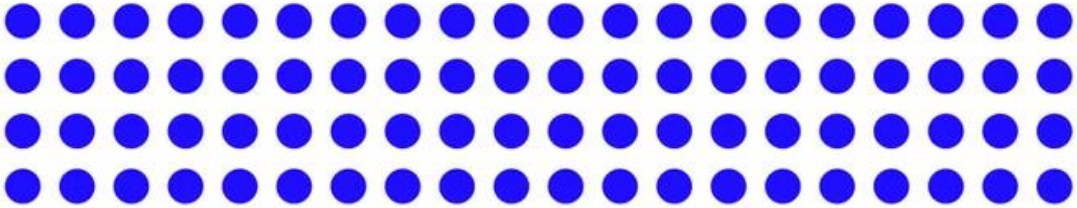
11: Humans Born (IN HISTORY)

17: Age of Universe (SECONDS)

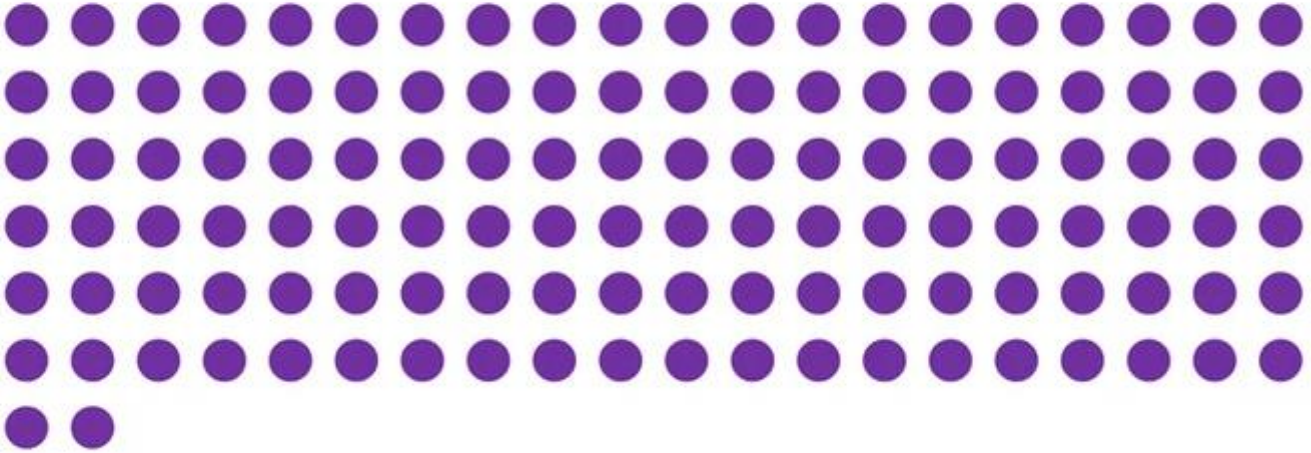
26: Diameter of Universe (METERS)

30: Stars in Universe

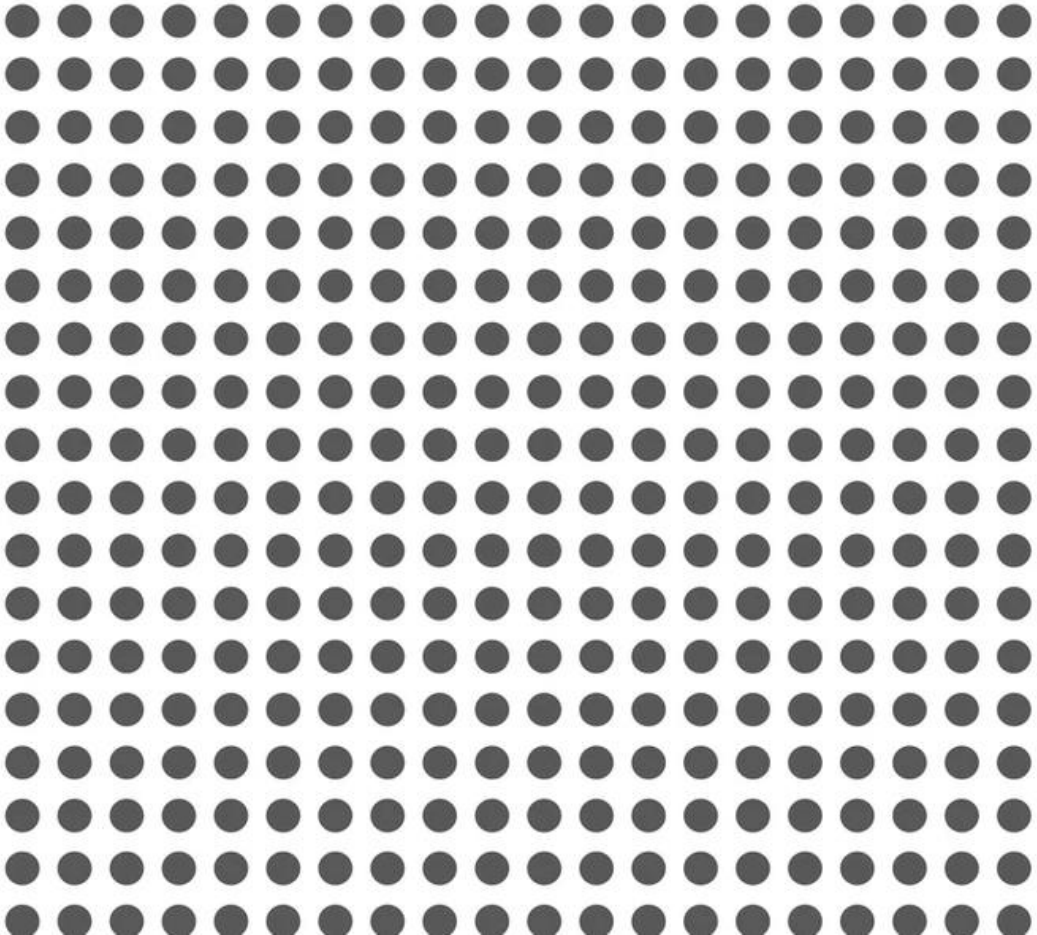
80: Atoms in Universe



122: Protons Needed to Fill Universe



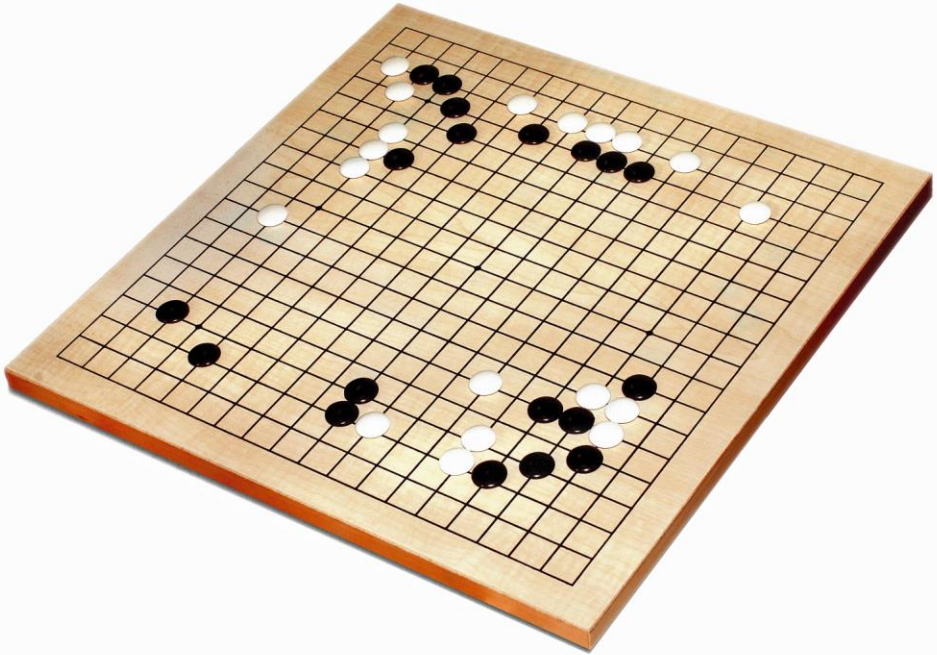
360: Go Game-Tree Complexity

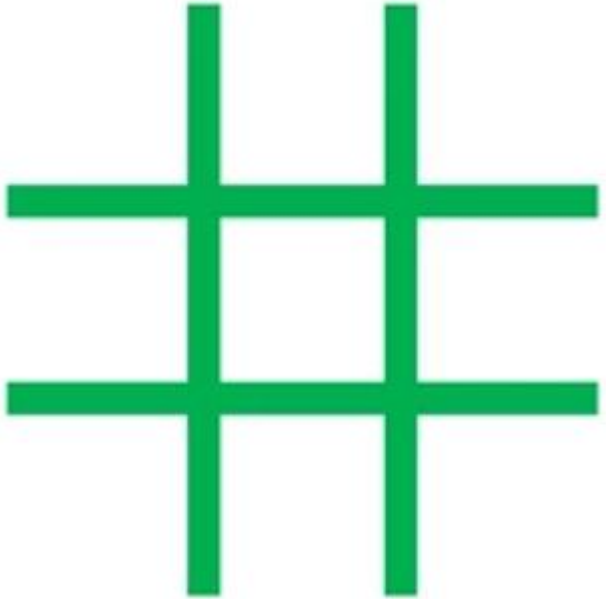



Lower Bound
10^{1000}

State Space and Complexity

- Important note: a large state space does not PROVE that a game is complex
- Consider the game of Go with no captures
  - Whoever moves first wins, game is trivial
- However, it does show us the computational complexity required to solve the game via exhaustive search (tic tac toe is easy)





Environment Definition

- The rules / dynamics that govern it
  - Physics, Game Rules, etc
- The legal actions allowed at any State
- Initial / Goal States
  - Usually given by the problem instance

- Environments have several Properties

Fully vs. Partially Observable

- Fully Observable
  - Agent's sensors give it access to the complete state of the environment at any given time
  - Access to 'relevant' part of state for actions
  - Games: 'Perfect Information Game'
- Partially Observable:
  - Data is hidden or occluded
  - Sensors may be noisy
  - Games: 'Imperfect Information Game'

Fully vs. Partially Observable

Deterministic vs. Stochastic

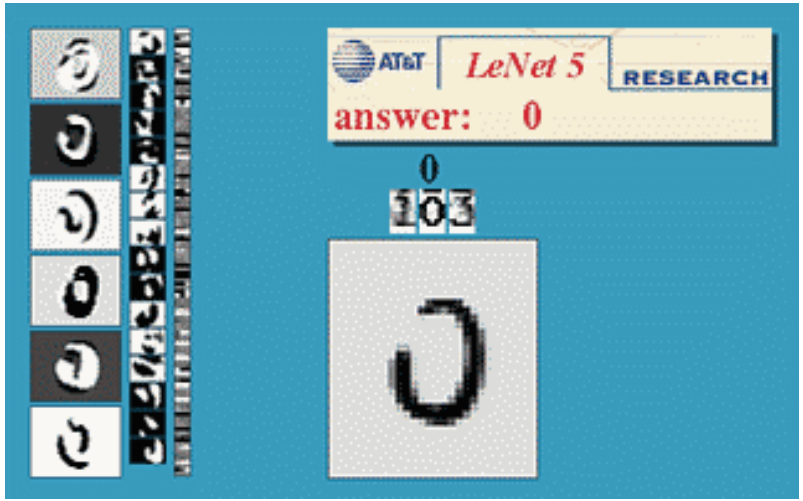
- Deterministic
  - The next state of the environment is entirely determined by the current state and the action of the agent
  - There is no randomness in the environment
- Stochastic
  - Uncertainty about the outcome of actions
    - Random Chance: Dice Roll, Card Draw, etc
  - Random nature to the environment
- Think from the point of view of the agent

Deterministic vs. Stochastic

Episodic vs. Sequential

- Episodic
  - Agent's experience and actions are divided into atomic episodes
  - Agent perceives current state, takes an action
  - Next episode does not depend on previous
- Sequential
  - Current action can influence future decisions
  - Actions may have long-term consequences

Episodic vs. Sequential



Dynamic vs. Static

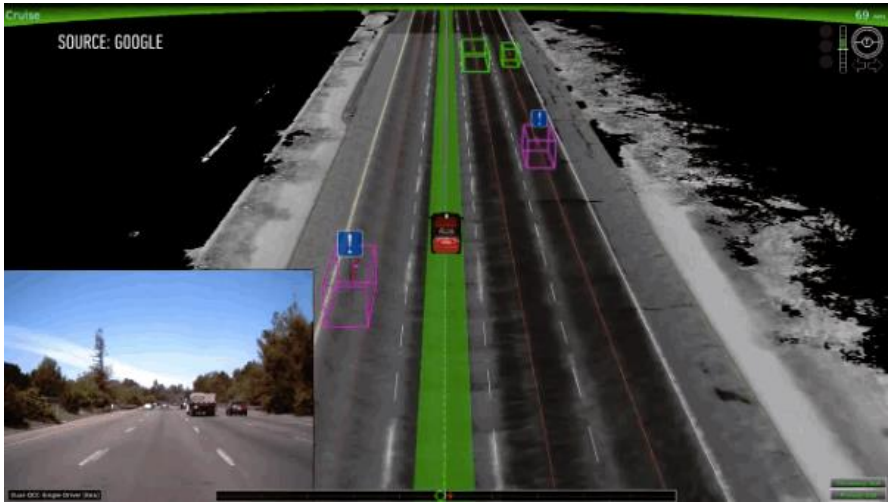
- Dynamic
  - Environment may change while agent decides
  - Continuously asking agent what to do
- Static
  - Environment only changes on agent action
  - Decision making time doesn't matter
- Games: Real-Time vs. Turn-Based

Dynamic vs. Static

Discrete vs. Continuous

- Distinction applied to the state of env.
- Determines the way time is handled
- Discrete
  - Finite number of distinct states
  - Discrete set of percepts and actions
- Continuous
  - Continuous time / actions

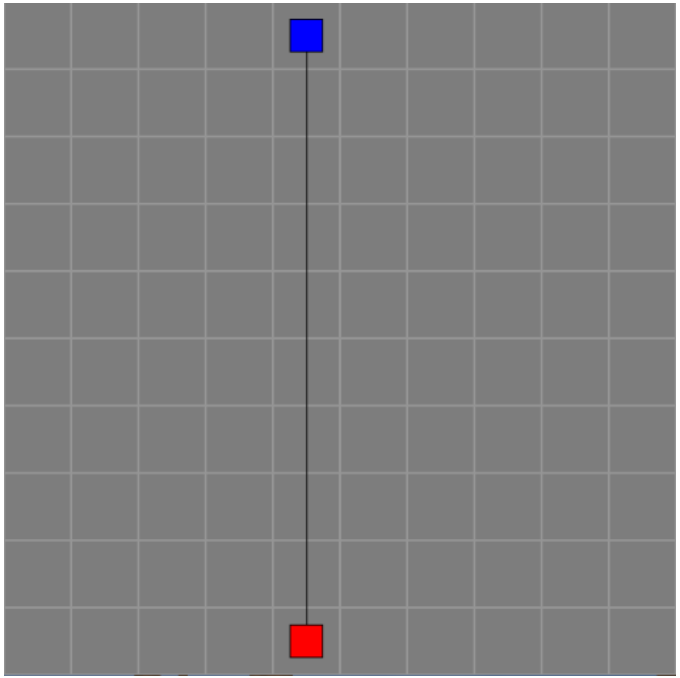
Discrete vs. Continuous



Single Agent vs. Multi-agent

- Single Agent
  - One agent or acting at a time
  - Ex: Puzzle solving, Path finding
- Multi-agent
  - Multiple agents acting together
  - May be cooperative or competitive

Single Agent vs. Multi-agent



Complete vs. Incomplete Information

- Complete Information
  - All game rules / physics are known
  - All possible actions are known

- Incomplete Information
  - Game rules / physics may not be known
  - Actions may have to be discovered

Example Game Environments

Environ.	Episodic	Agents	Determ.	Observe	Static	Discrete
CROSSWORD	SEQUENTIAL	SINGLE	DETERM	FULLY	STATIC	DISCRETE
CHESS	SEQUENTIAL	MULTI	DETERM	FULLY	STATIC	DISCRETE
BACK GAMMON	SEQUENTIAL	MULTI	STOCHASTIC	FULLY	STATIC	DISCRETE
POKER	SEQUENTIAL	MULTI	STOCHASTIC	PARTIALLY	STATIC	DISCRETE
STARCRAFT	SEQUENTIAL	MULTI	STOCHASTIC	PARTIALLY	DYNAMIC	DISCRETE
REAL WORLD	SEQUENTIAL	MULTI	STOCHASTIC	PARTIALLY	DYNAMIC	CONTINUOUS