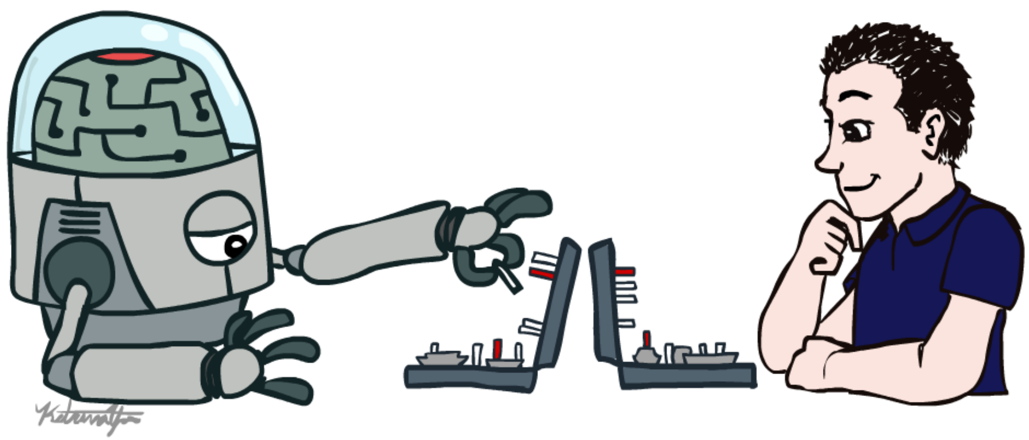
CS 188: Artificial Intelligence

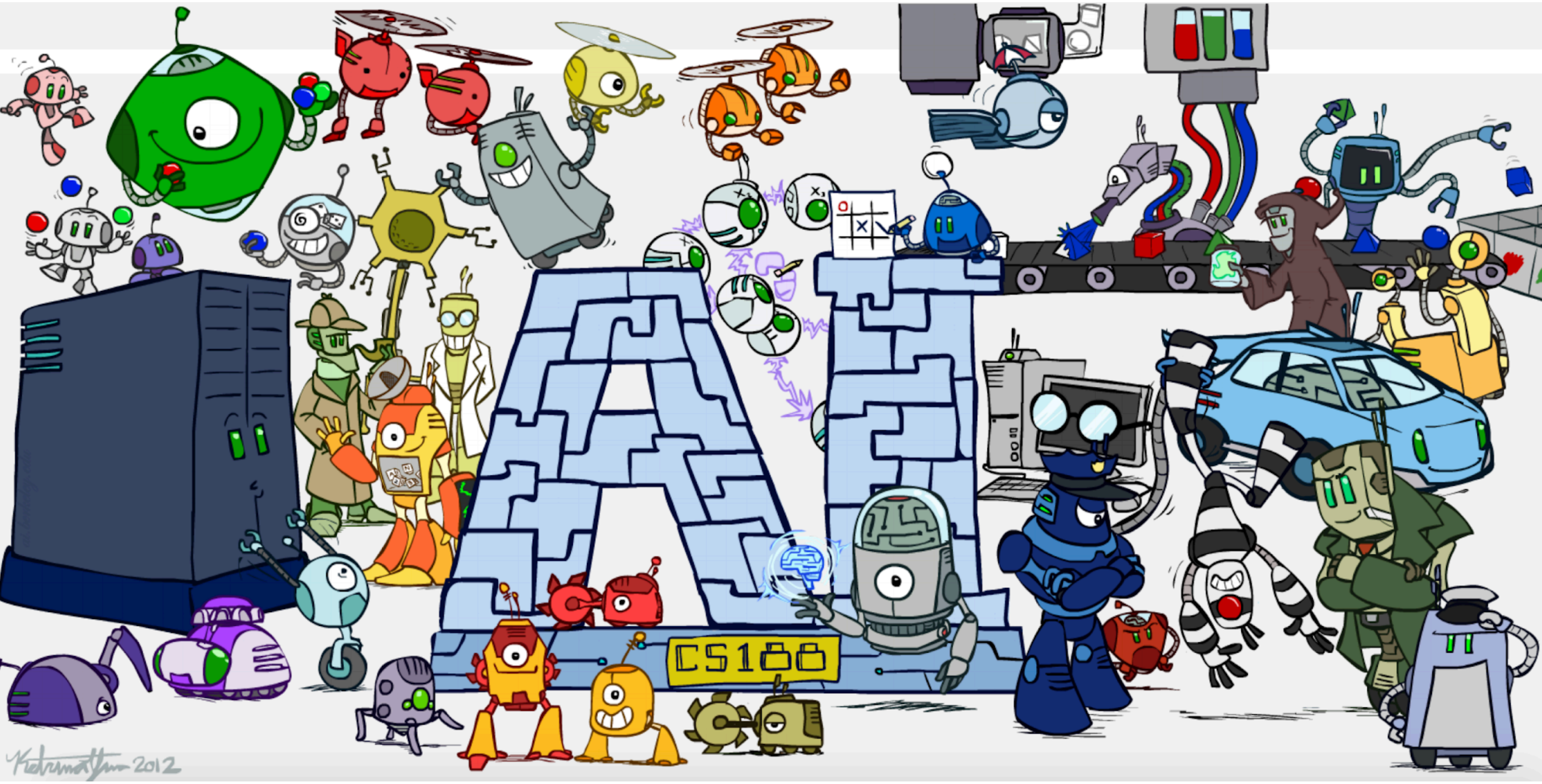
Introduction and Rationality



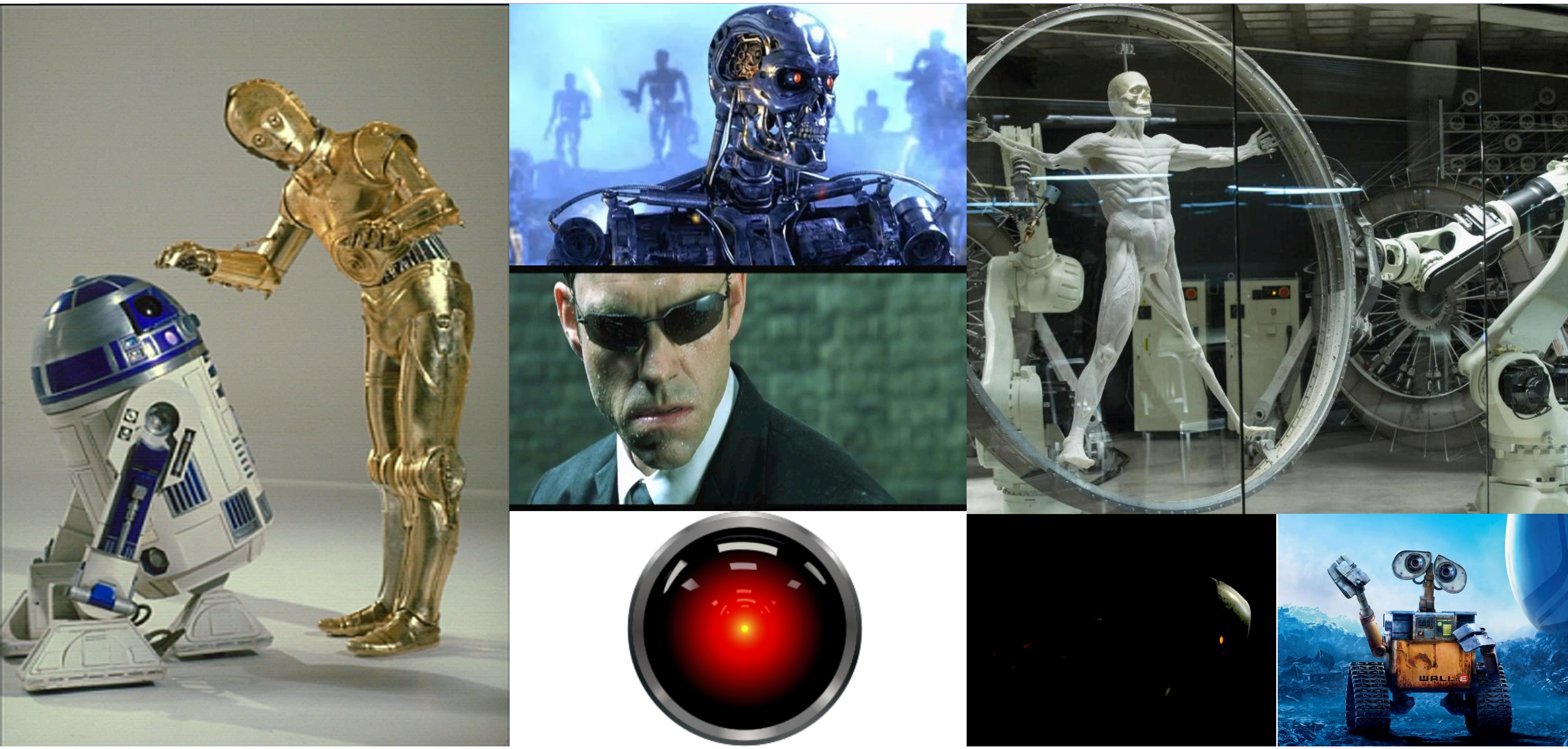
Instructors: Saagar Sanghavi, Nicholas Tomlin

University of California, Berkeley

(slides adapted from Dan Klein, Pieter Abbeel, Anca Dragan, Stuart Russell)


CS188

AI




TUG
CAUTION
MAY CONTAIN
CHEMOTHERAPY DRUG
CAUTION
MAY CONTAIN
CHEMOTHERAPY DRUG

SS

What is artificial intelligence?

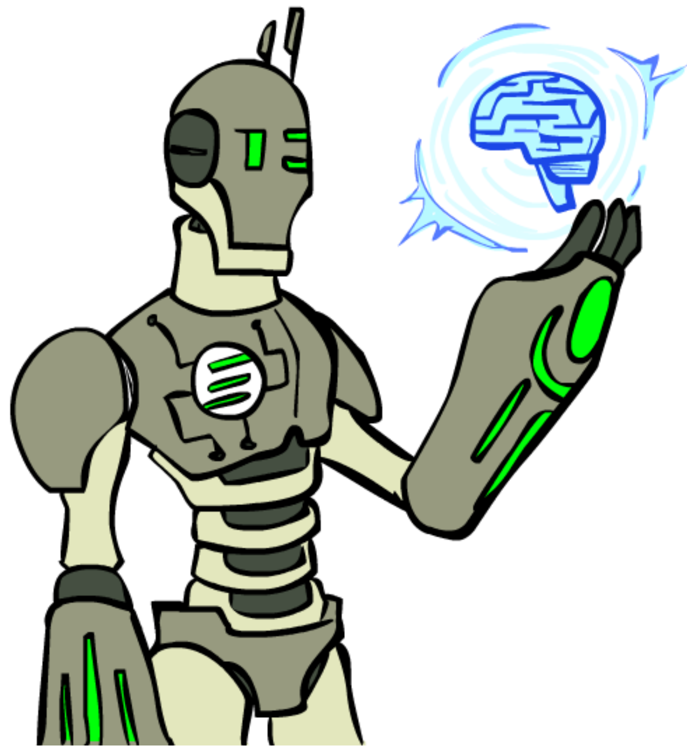
Artificial intelligence (AI) refers to the simulation of human intelligence in machines that are programmed to think and learn like humans. It is a broad field of computer science that focuses on creating intelligent machines capable of performing tasks that typically require human intelligence, such as visual perception, speech recognition, decision-making, problem-solving, and language translation.

AI encompasses various subfields and techniques, including machine learning, natural language processing, computer vision, expert systems, and robotics. These approaches enable AI systems to acquire knowledge, process information, reason, and make predictions or decisions based on the available data.

Machine learning, a key component of AI, involves training algorithms to recognize patterns in large amounts of data and make predictions or take actions without being explicitly programmed. This ability to learn from experience and adapt to new situations is what sets AI apart from traditional software systems.

In these slides…

- What is artificial intelligence?
- Where did it come from/What can AI do?
  - What should we and shouldn't we worry about?
- Course Logistics
- Utilities and Rationality



What is AI?

The science of making machines that:

Rational Decisions

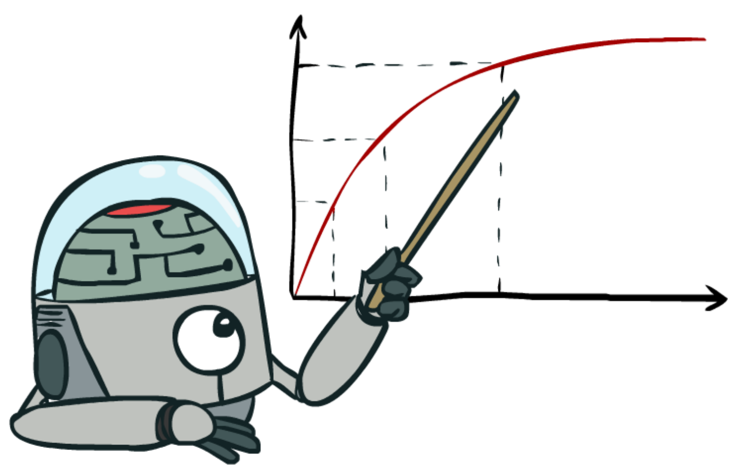
We'll use the term **rational** in a very specific, technical way:

- Rational: maximally achieving pre-defined goals
- Rationality only concerns what decisions are made
  (not the thought process behind them)
- Goals are expressed in terms of the **utility** of outcomes
- Being rational means **maximizing your expected utility**

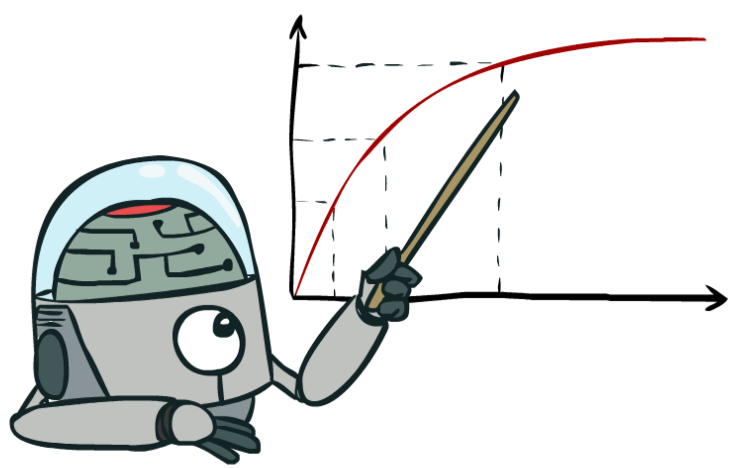
A better title for this course would be:

Computational Rationality

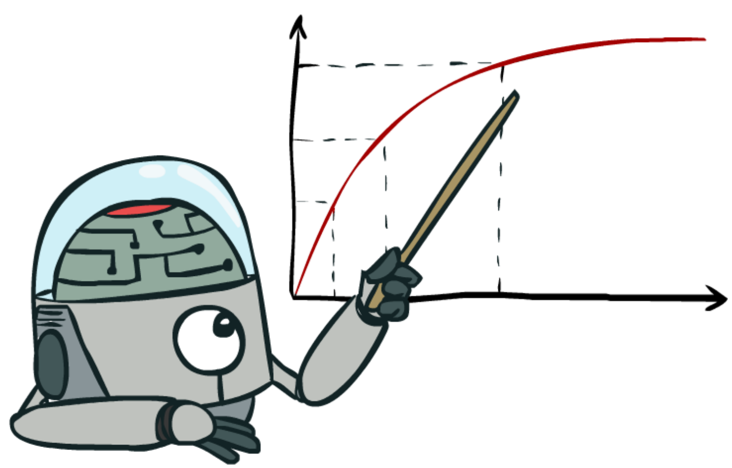
Maximize Your Expected Utility



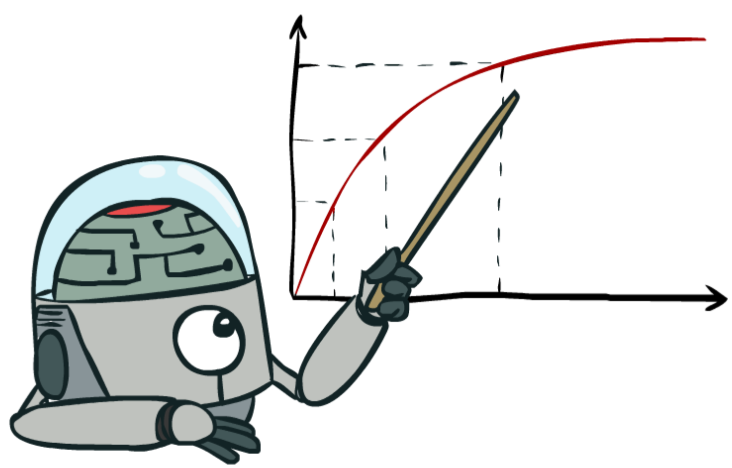
Maximize Your Expected Utility



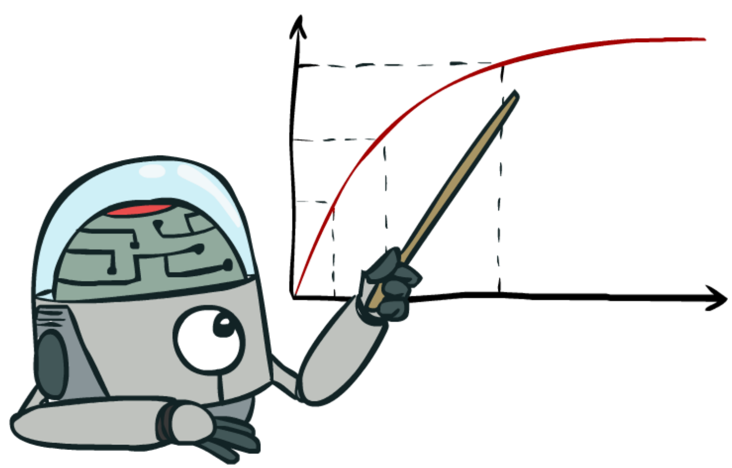
Maximize Your Expected Utility



Maximize Your Expected Utility



Maximize Your Expected Utility

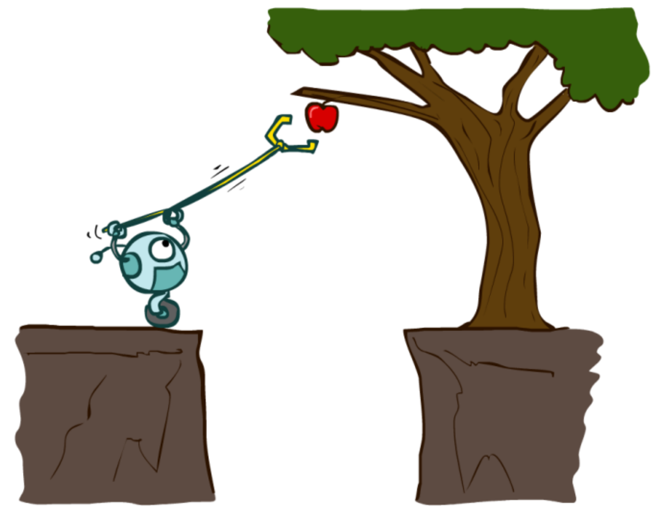


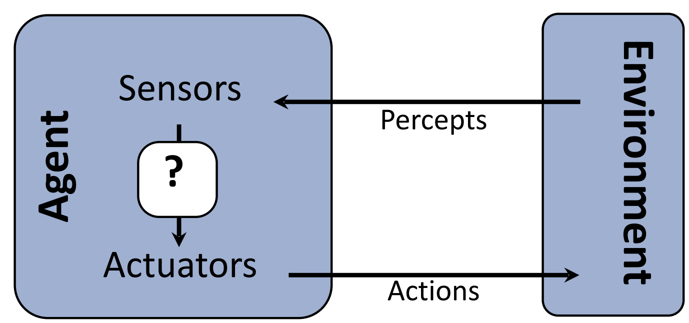
What About the Brain?

- Brains (human minds) are very good at making rational decisions, but not perfect
- Brains aren't as modular as software, so hard to reverse engineer!
- "Brains are to intelligence as wings are to flight"
- Lessons learned from the brain: memory and simulation are key to decision making

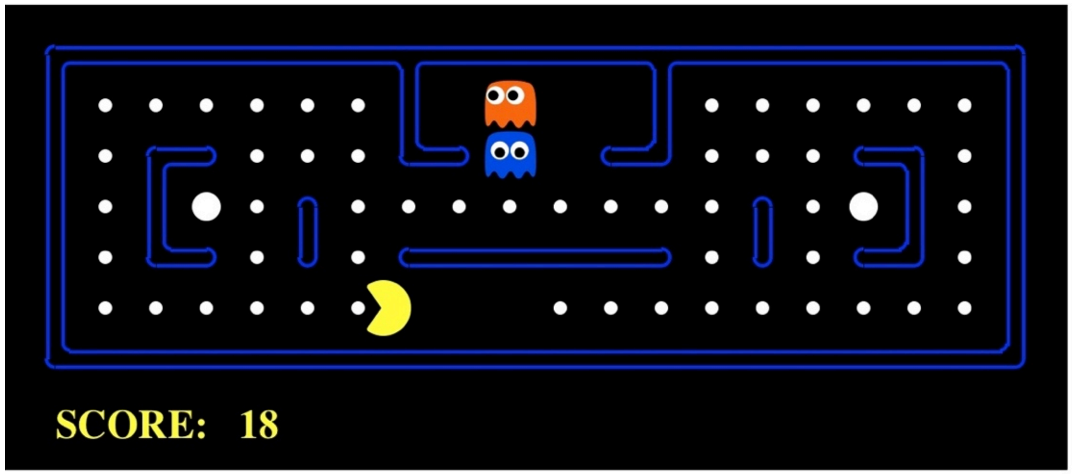
Designing Rational Agents

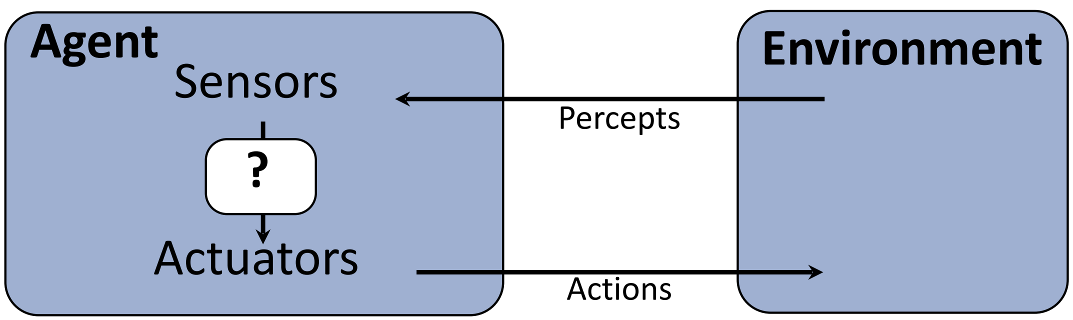
- An **agent** is an entity that *perceives* and *acts*.
- A **rational agent** selects actions that maximize its (expected) **utility**.
- Characteristics of the **percepts, environment,** and **action space** dictate techniques for selecting rational actions
- **This course** is about:
  - General AI techniques for a variety of problem types
  - Learning to recognize when and how a new problem can be solved with an existing technique





Pac-Man as an Agent

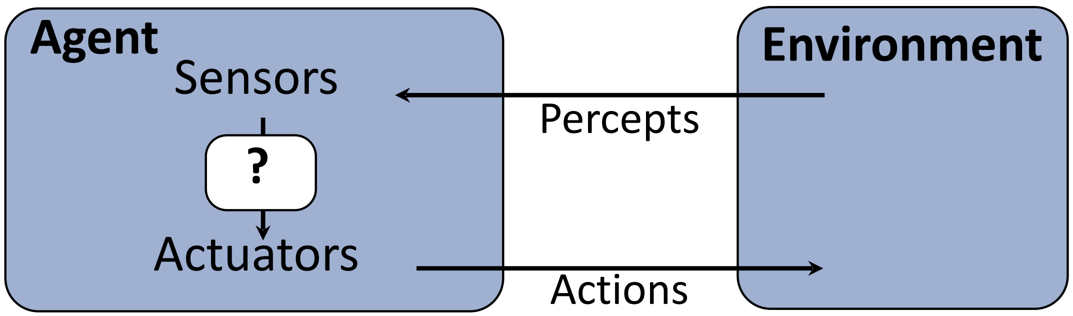




Pac-Man is a registered trademark of Namco-Bandai Games, used here for educational purposes

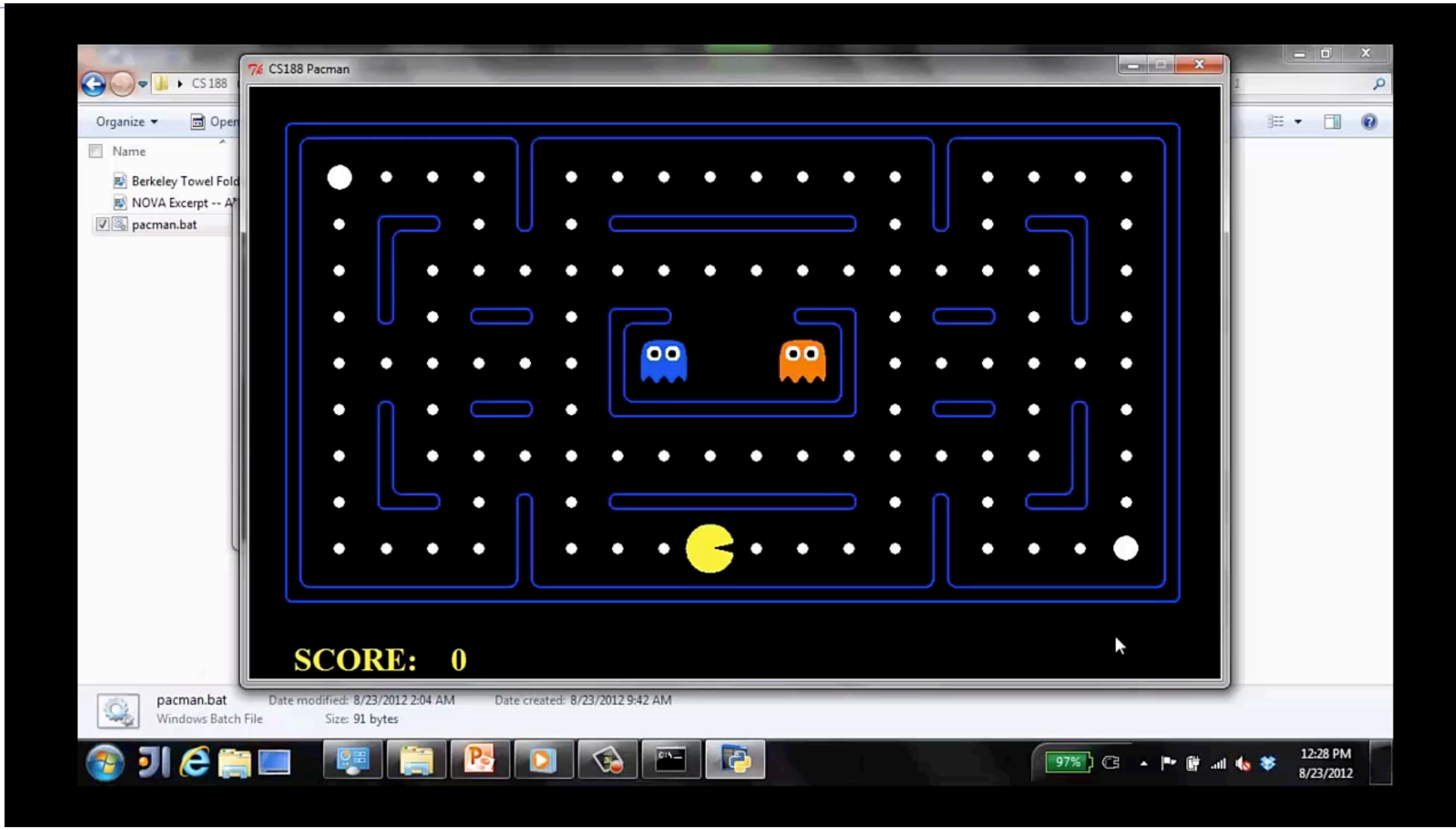
Demo1: pacman-l1.mp4

Agents and environments



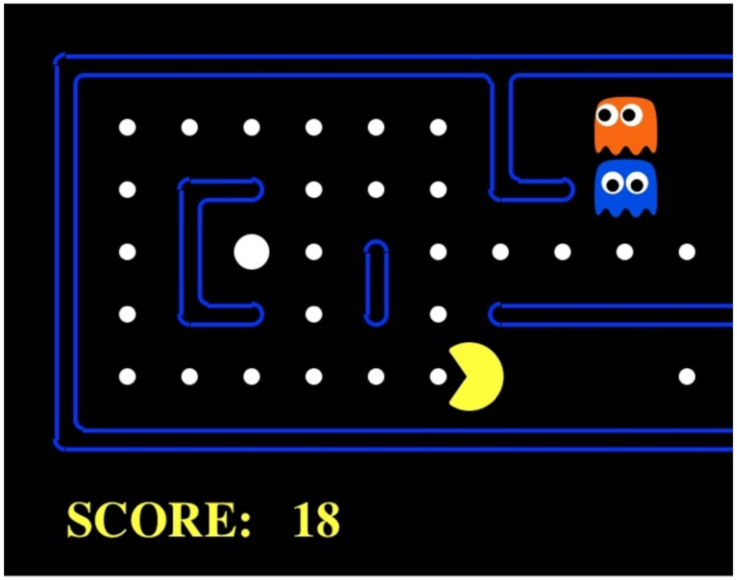
- An agent ***perceives*** its environment through ***sensors*** and ***acts*** upon it through ***actuators*** (or *effectors*, depending on whom you ask)
- The *agent function* maps percept sequences to actions
- It is generated by an ***agent program*** running on a ***machine***

A human agent in Pacman



The task environment - PEAS

- Performance measure
  - -1 per step; + 10 food; +500 win; -500 die; hit scared ghost
- Environment
  - Pacman dynamics (incl ghost behavior)
- Actuators
  - Left Right Up Down or NSEW
- Sensors
  - Entire state is visible (except power pellet duration)



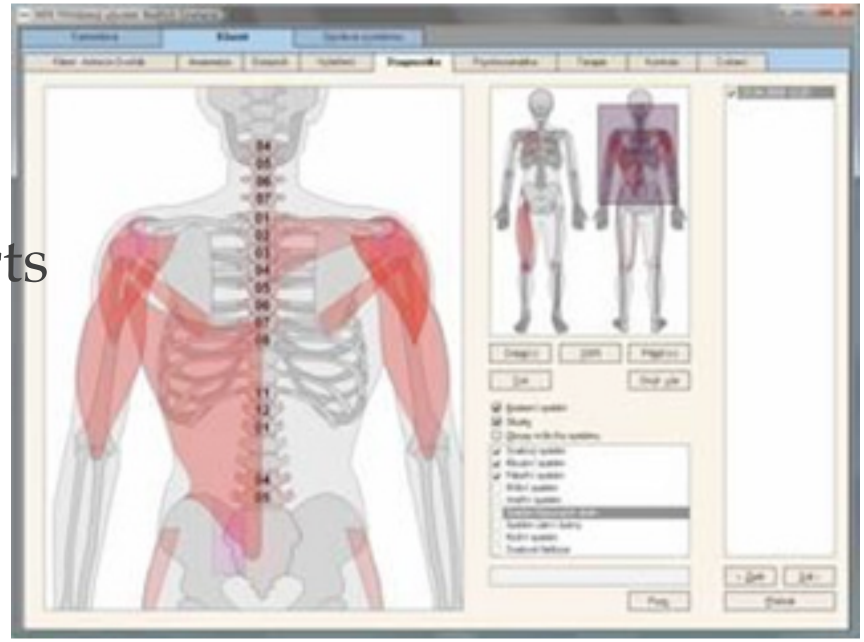
PEAS: Automated taxi

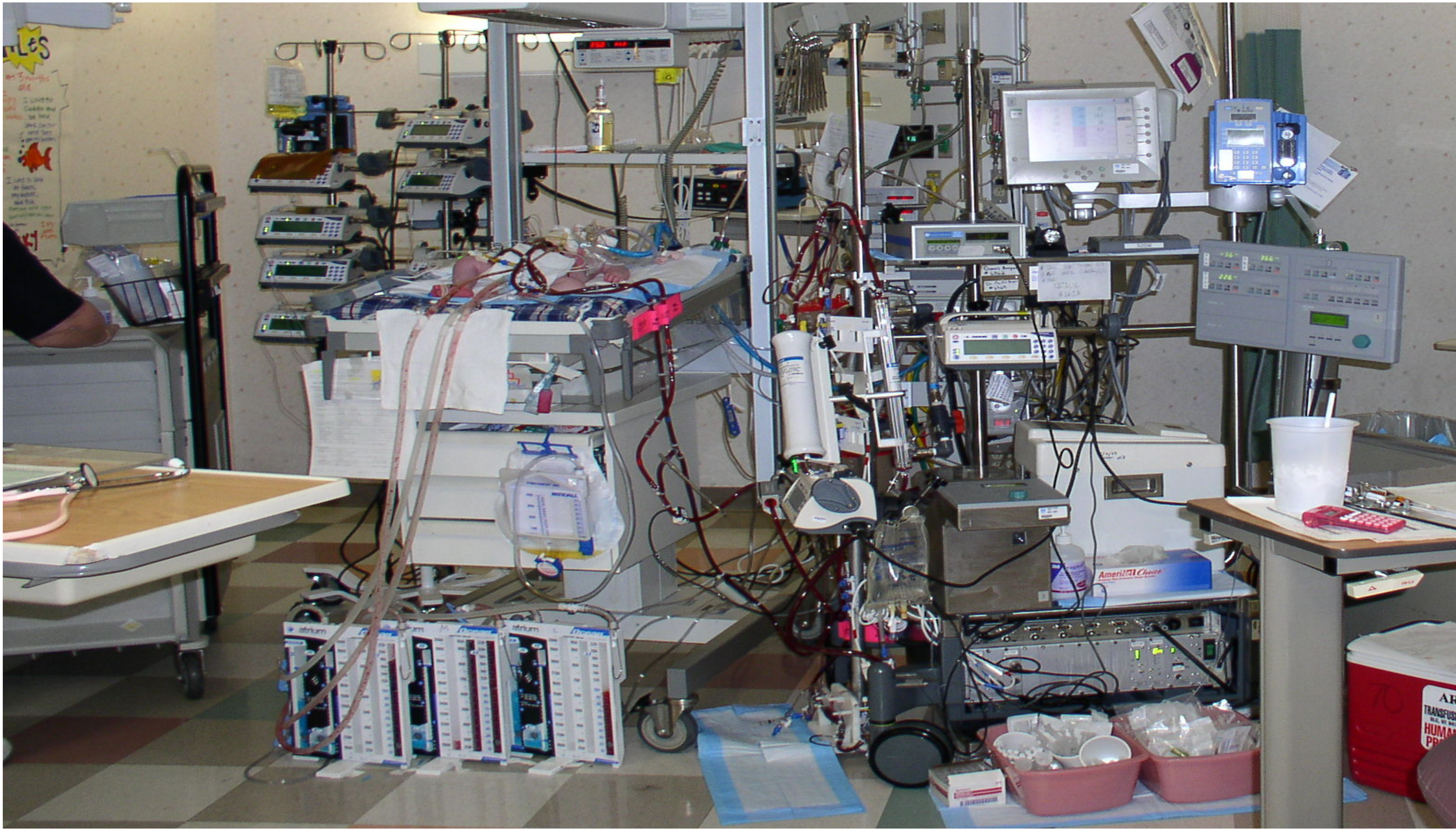
- Performance measure
  - Income, happy customer, vehicle costs, fines, insurance premiums
- Environment
  - US streets, other drivers, customers, weather, police…
- Actuators
  - Steering, brake, gas, display/speaker
- Sensors
  - Camera, radar, accelerometer, engine sensors, microphone, GPS

Image: http://nypost.com/2014/06/21/how-google-might-put-taxi-drivers-out-of-business/

PEAS: Medical diagnosis system

- Performance measure
  - Patient health, cost, reputation
- Environment
  - Patients, medical staff, insurers, courts
- Actuators
  - Screen display, email
- Sensors
  - Keyboard / mouse




atrium

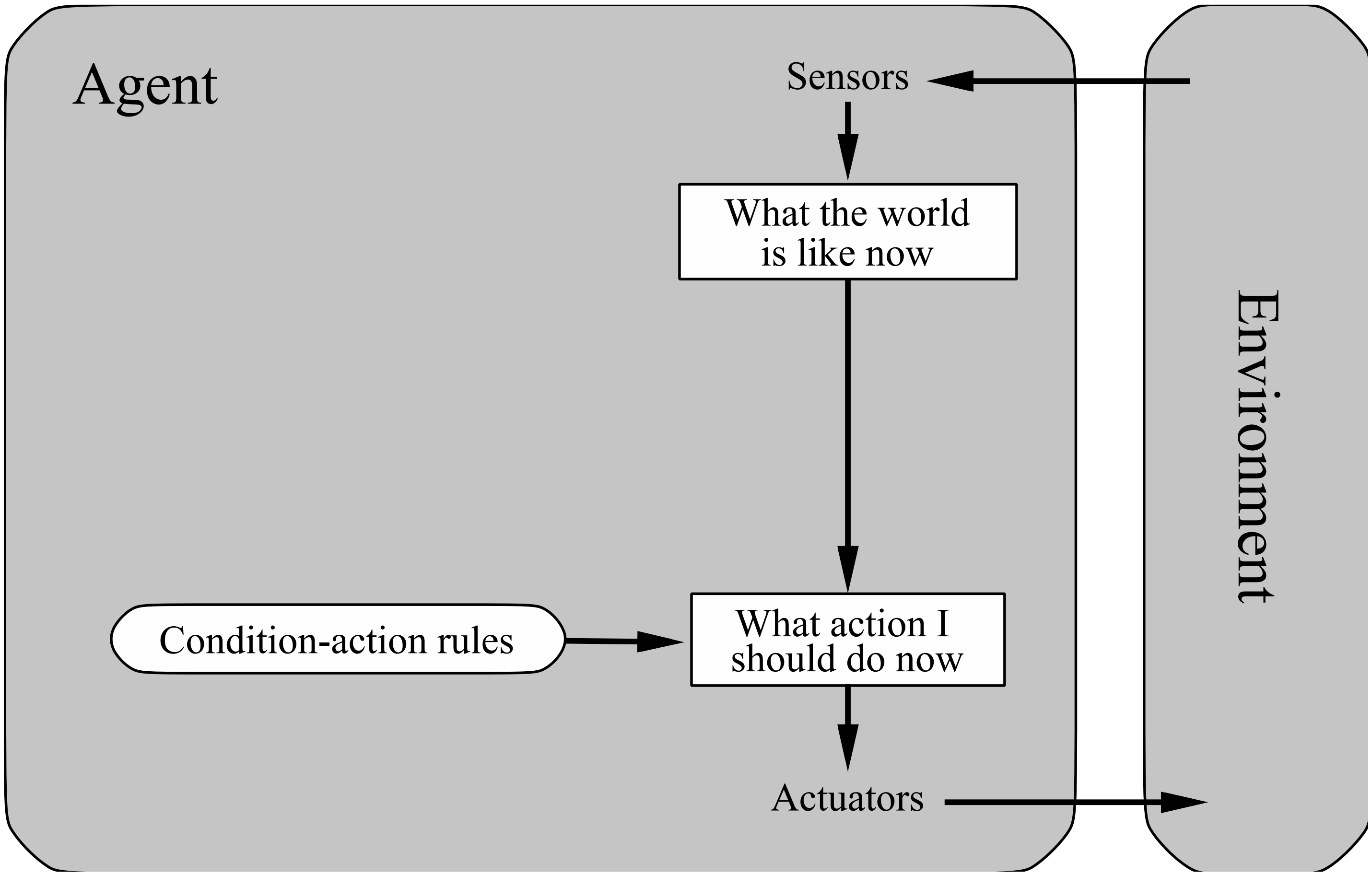
Environment types

	Pacman	Diagnosis	Taxi
Fully or partially observable			
Single-agent or multiagent			
Deterministic or stochastic			
Static or dynamic			
Discrete or continuous			
Known physics?			
Known perf. measure?			

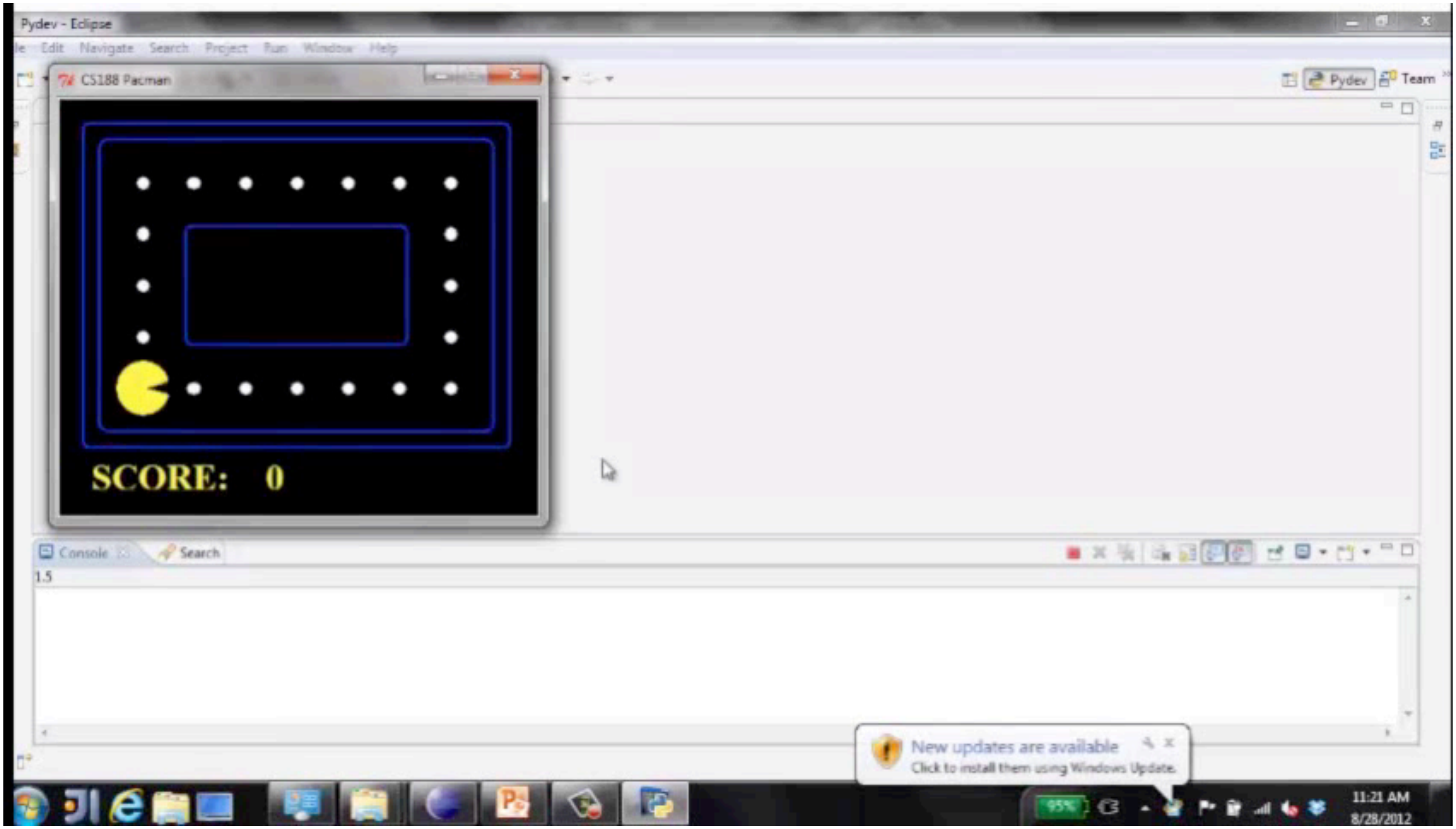
Agent design

- **The environment type largely determines the agent design**
  - ***Partially observable*** => agent requires ***memory*** (internal state)
  - ***Stochastic*** => agent may have to prepare for ***contingencies***
  - ***Multi-agent*** => agent may need to behave ***randomly***
  - ***Static*** => agent has time to compute a rational decision
  - ***Continuous time*** => continuously operating ***controller***
  - ***Unknown physics*** => need for ***exploration***
  - ***Unknown perf. measure*** => observe/interact with ***human principal***

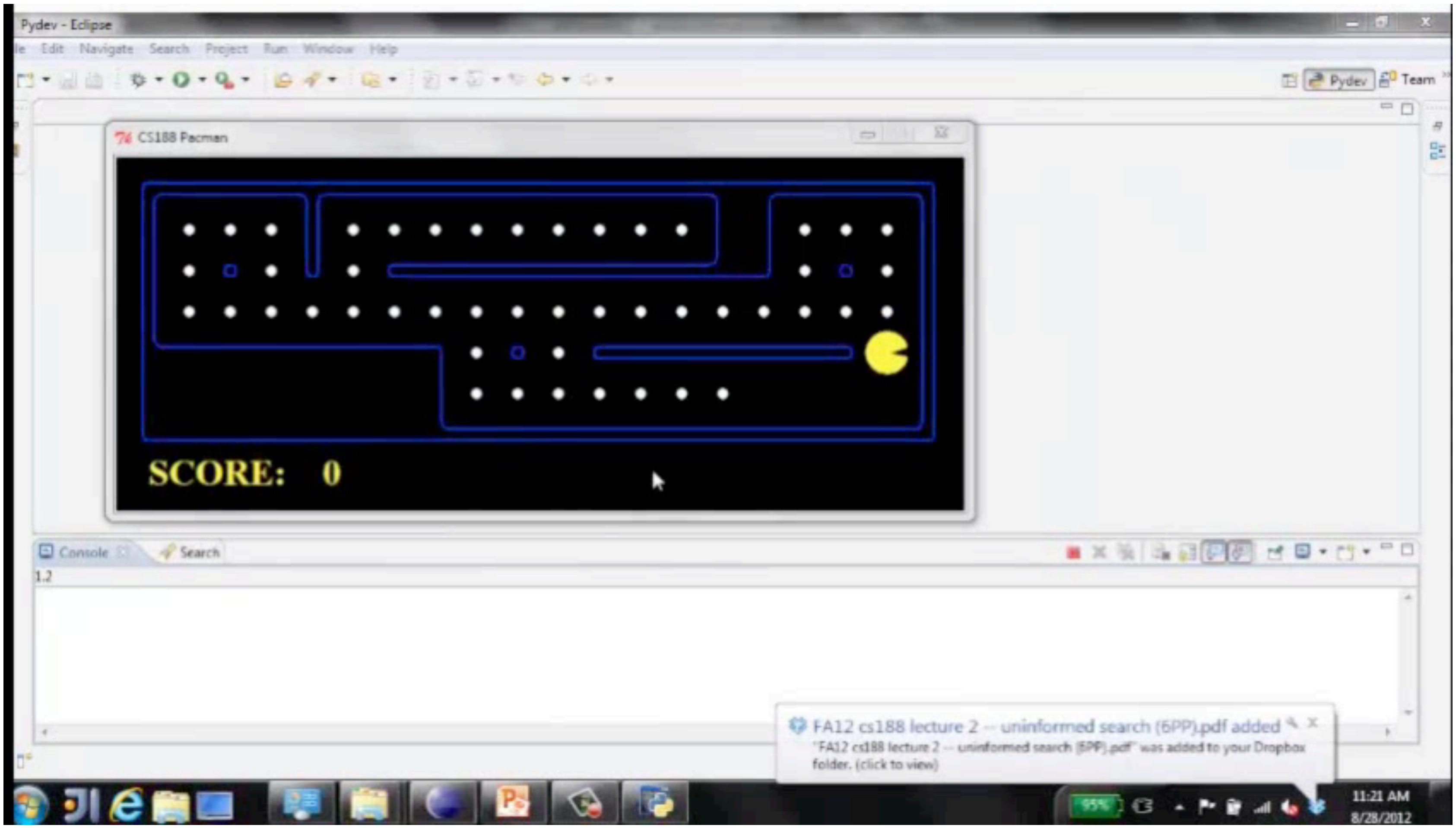
Simple reflex agents



Eat adjacent dot, if any



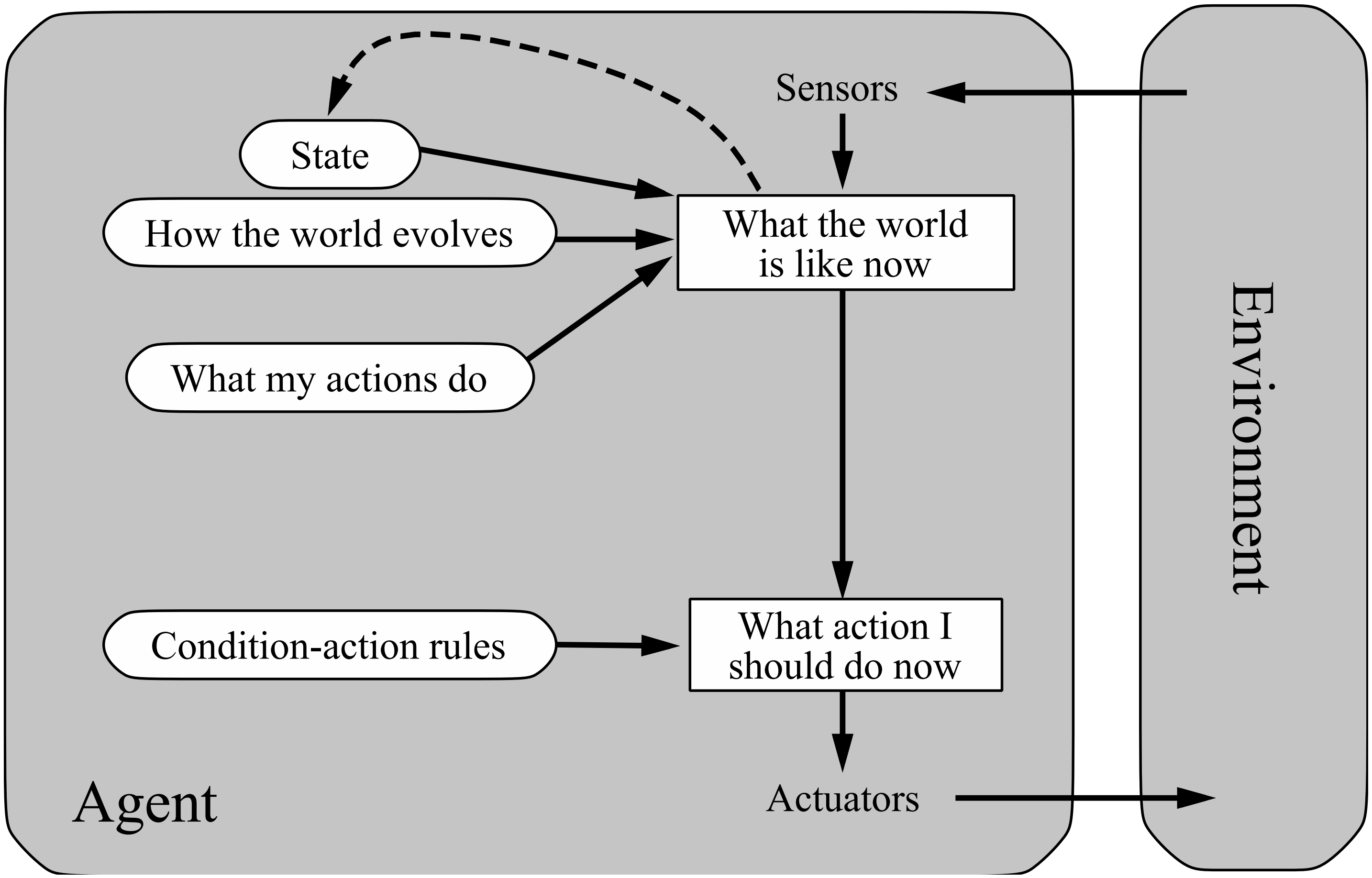
Eat adjacent dot, if any



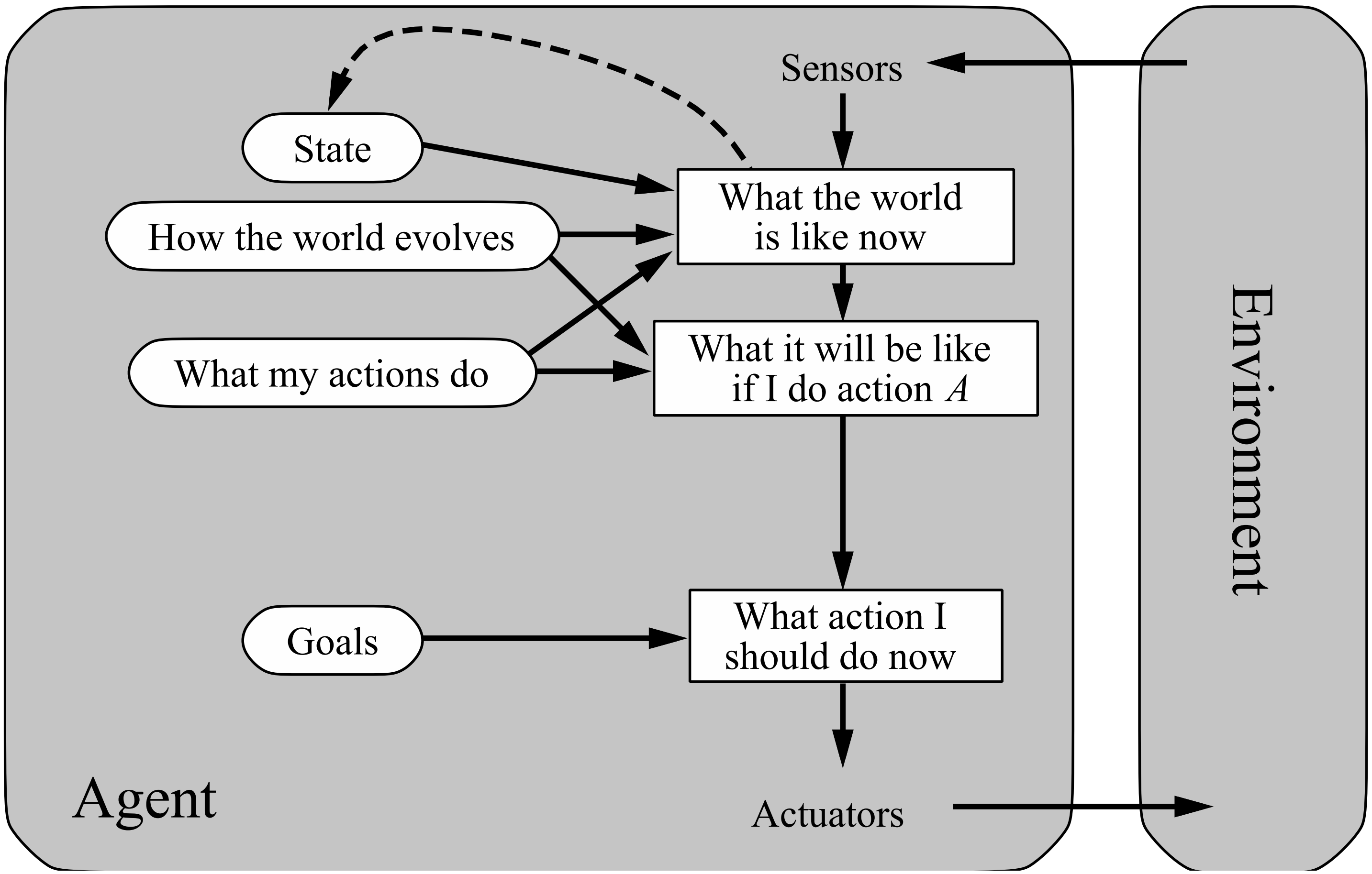
Pacman agent contd.

- Can we (in principle) extend this reflex agent to behave well in all standard Pacman environments?
  - No – Pacman is not quite fully observable (power pellet duration)
  - Otherwise, yes – we can (*in principle*) make a lookup table…..
  - *How large would it be?*

Reflex agents with state



Goal-based agents



Logistics

- Course Website, Gradescope, Edstem
- Discussion Sections in Cory 521: signup form on EdStem
- Homework
- Projects
- Exams
- Office Hours

Website

- Website: https://inst.eecs.berkeley.edu/~cs188/su23
  - tentative schedule
  - lecture slides and notes
  - course policies, etc.

EdStem

- Communication:
  - Ask and answer questions; announcements
  - Private matters – private messages
    - if your message is not answered promptly enough, here is the staff email: cs188@berkeley.edu
  - Exceptional circumstances and DSP extensions: email eecs-course-management@eecs.berkeley.edu. We will follow only what is provided in the DSP letter.

Course Format

- Lectures Mon – Thu in **Lewis 100**
  - Please show up and actively engage; participation credit
    - Raise your hand to ask questions
  - May be recorded by department
    - Slides posted after lecture
    - Course notes available on website

Course Format

- Discussion Sections
  - Daily, corresponds to each day's lecture
  - Schedule on course website, signup on SignUpGenius
  - Pick 1 to go to; show up to it consistently; participation credit
- Summer Session is DOUBLE SPEED
  - Please make sure to stay on top of the material!

Course Format (continued)

- Homework
  - Due Fridays at at 10:59pm (grace period up to 11:59pm)
  - Exercises based on class material
  - Get you comfortable with the basics
  - Solve together, submit your own
  - Autograded, multiple submissions!
  - We expect you to get 100% on autograded parts of homework
  - You get to drop one assignment. *No slip days* — use drop.

Course Format (continued)

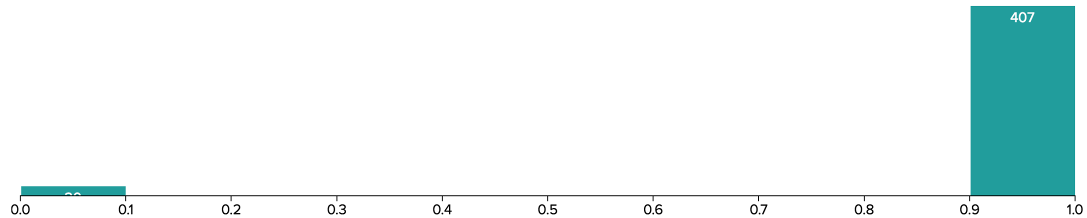
- Projects
  - Due Tuesdays at 10:59pm
    - Grace period to 11:59pm
  - 6 projects, groups of 1-2
    - Academic integrity! Copying code is easy to catch with MOSS, so please do your own work
  - Python, autograded
  - Give you hands-on experience with the algorithms
  - I expect you to get 100% on projects
  - 5 slip days
    - After that, lose 20% per day turned in late

Course Format (continued)

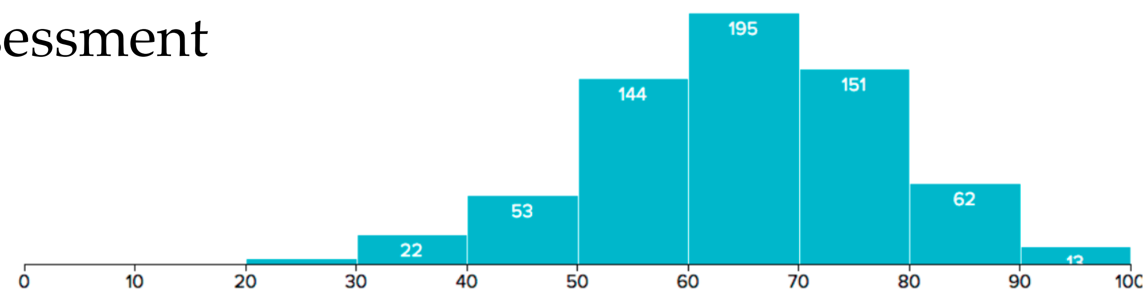
- Exams
  - Midterm: Mon, Jul 17 from 7-9pm
  - Final: Thu, Aug 10 from 7-10pm
- **Exams are the main assessment tool, so they are hard**
  - **Use the hard discussion sections to prepare!**
  - **Previous exams available on the course website, but they are mostly multiple choice (this semester's exams will be mostly free response questions)**

Some Historical Statistics

- Homework and projects: instruction (iterate/learn till you nailed it)



- Exams: assessment



Course Format (continued)

- Office hours
  - Saagar: After lecture 3:30-4:30pm for weeks 1-4
  - Nick: TBD
  - Schedule is on course website
  - GSI and uGSI: projects, homework, discussion questions, etc.
  - Saagar/Nick: concepts, high level guidance

Location subject to change!! Check Ed for updates.

Prerequisites

- Math (CS 70, and Math 1B/53 may be helpful)
  - Discrete Math and Probability Theory
  - Basic Calculus, Partial Derivatives
- Programming (CS 61A, CS 61B)
  - Basic Python (learn the syntax if you don't know it already)
  - Understanding of Data Structures, Object Oriented Programming, etc.

Announcements

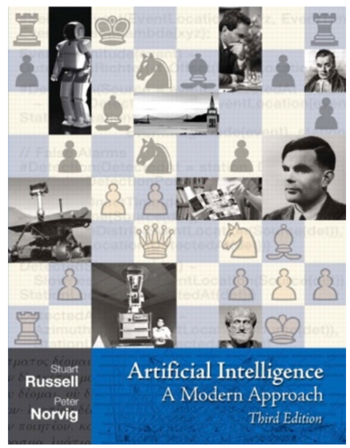
- Project 0: Python Tutorial
  - Due Fri, Jun 23 at 10:59pm
- Homework 1: Math Review and Search
  - Due Fri, Jun 23 at 10:59pm
- Project 1: Search
  - Will go out this week, best way to test programming preparedness
  - Due Tue, Jun 27 at 10:59pm
- Sections
  - Start this week. There is one section corresponding to every lecture

Important This Week

- Important this week:
  - **Register** for the class on Gradescope: enrollment code 6ZY5ZX
  - **Register** for the class on EdStem—our main resource for discussion and communication
  - **Signup** for discussion sections — link on Edstem
  - **P0: Python tutorial and HW1: Math Review and Search** are out; due this Friday
  - **P1: Search** is out; due next Tuesday
  - **Mark exam dates in your calendars**
- Also important:
  - **Sections and office hours** start today
  - **If you are wait-listed,** you might or might not get in depending on how many students drop

Textbook

- Not required, but for students who want to read more we recommend
  - Russell & Norvig, AI: A Modern Approach



  - Warning: Not a course textbook, so our presentation does not necessarily follow the presentation in the book.

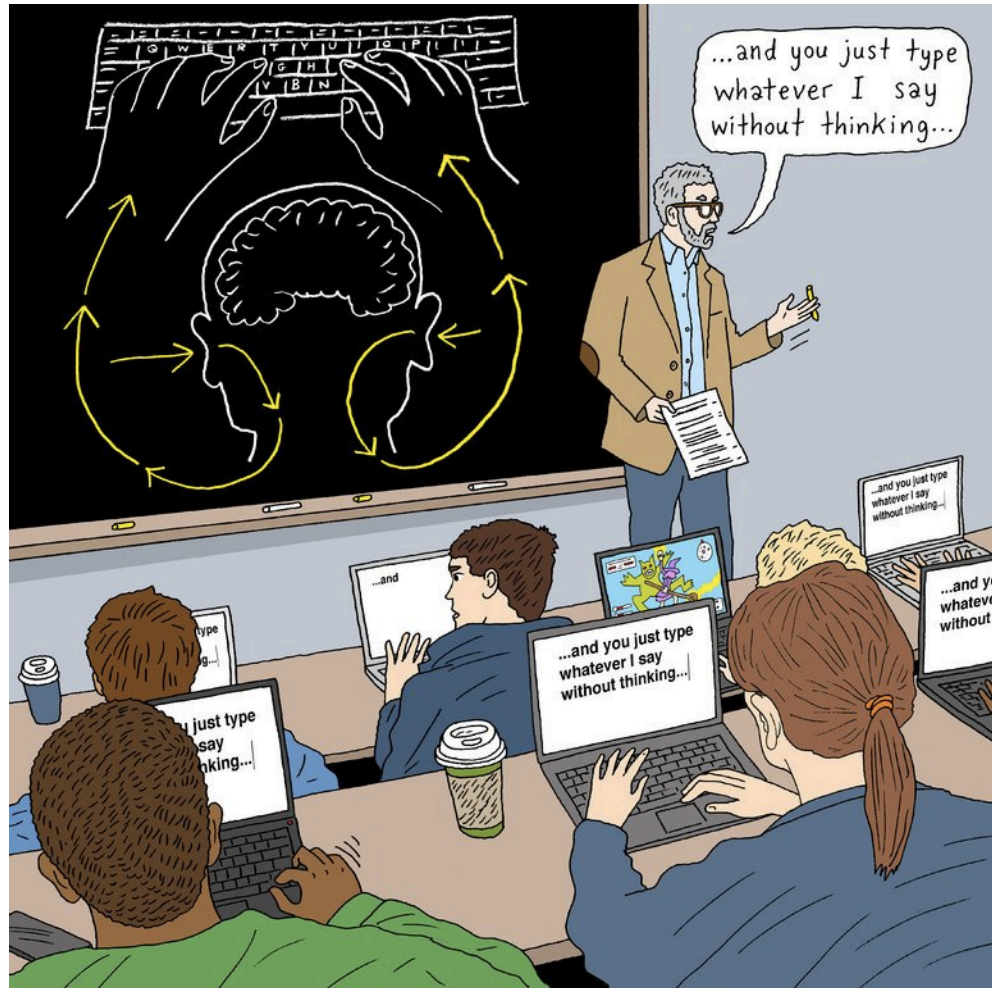
Laptops in Lecture?

The New York Times

Laptops Are Great. But Not During a Lecture or a Meeting.

Economic View

By SUSAN DYNARSKI NOV. 22, 2017



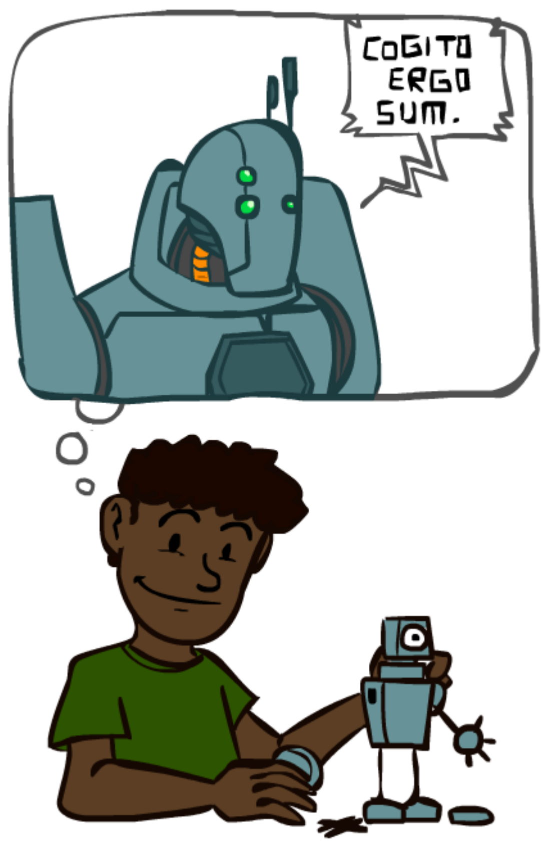
Laptops in Lecture

- Please do not use laptops or phones in lecture.
- If you want to multitask, please go home and watch the recording instead.
- (I encourage you to sit in the front so that we can have an interaction. That is the best way to learn!)
- Only exception is if there are accessibility issues (if so, please sit in the back)

Build a great community

- Help out your peers on piazza and in meetings!
- Be mindful of the tone you use – be respectful and supportive, help everyone feel at home.
- Watch out for implicit bias.
  - Someone's gender, race, ethnicity, sexual orientation, etc. do NOT have anything to do with how awesome they will be in this class.
  - Having a ton of programming experience will help with projects, but does NOT give anyone an edge on how well they can understand the material and how highly they can score on the exams.
- Please talk to me/Nick if something we or the GSIs do makes you uncomfortable!

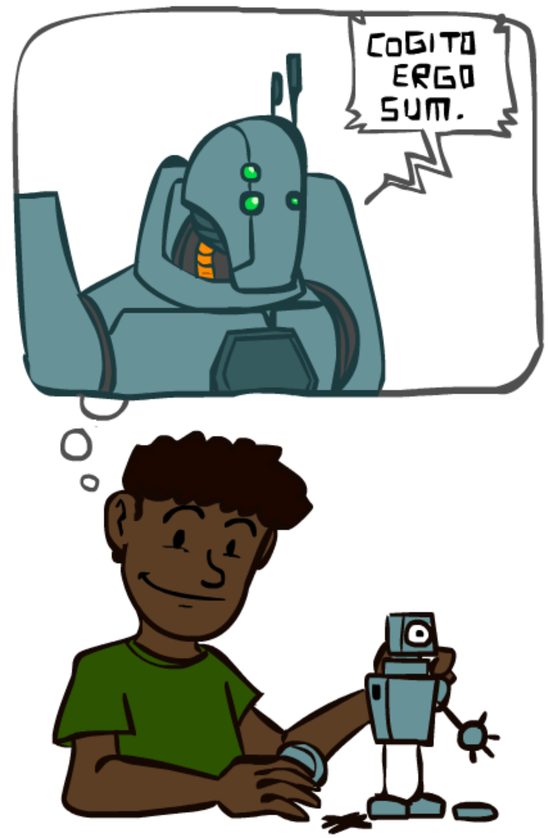
A (Short) History of AI



[redacted]

A (Short) History of AI

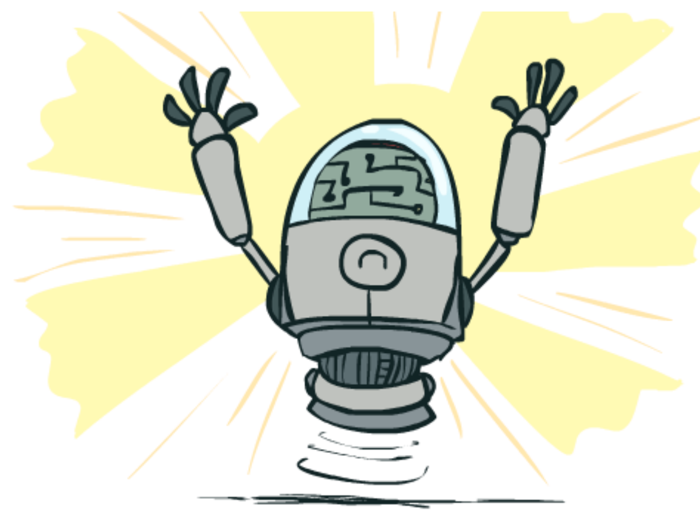
- 1940-1950: Early days
  - 1943: McCulloch & Pitts: Boolean circuit model of brain
  - 1950: Turing's "Computing Machinery and Intelligence"
- 1950—70: Excitement: Look, Ma, no hands!
  - 1950s: Early AI programs, including Samuel's checkers program, Newell & Simon's Logic Theorist, Gelernter's Geometry Engine
  - 1956: Dartmouth meeting: "Artificial Intelligence" adopted
  - 1965: Robinson's complete algorithm for logical reasoning
- 1970—90: Knowledge-based approaches
  - 1969—79: Early development of knowledge-based systems
  - 1980—88: Expert systems industry booms
  - 1988—93: Expert systems industry busts: "AI Winter"
- 1990—: Statistical approaches
  - Resurgence of probability, focus on uncertainty
  - General increase in technical depth
  - Agents and learning systems… "AI Spring"?
- 2000—: Where are we now?



What Can AI Do?

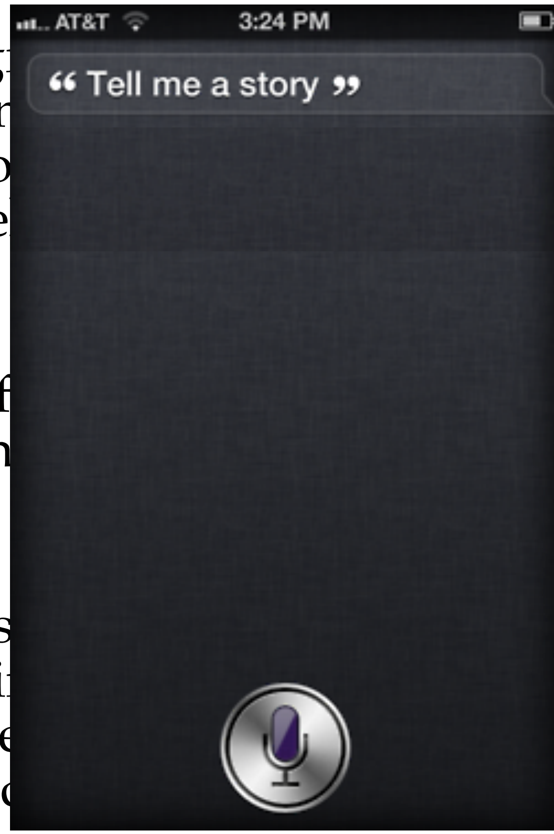
Quiz: Which of the following can be done at present?

- Play a decent game of Jeopardy?
- Win against any human at chess?
- Win against the best humans at Go?
- Play a decent game of tennis?
- Grab a particular cup and put it on a shelf?
- Unload any dishwasher in any home?
- Drive safely along the highway?
- Drive safely along Telegraph Avenue?
- Buy a week's worth of groceries on the web?
- Buy a week's worth of groceries at Berkeley Bowl?
- Discover and prove a new mathematical theorem?
- Perform a surgical operation?
- Unload a know dishwasher in collaboration with a person?
- Translate spoken Chinese into spoken English in real time?
- Write an intentionally funny story?



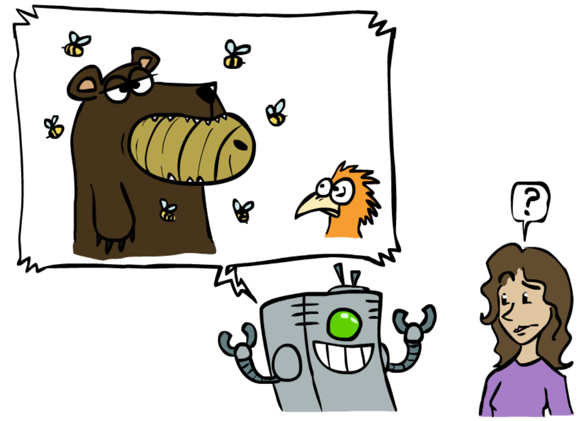
Unintentionally Funny Stories

- One day Joe Bear was hung
  Irving Bird where some hon n
  there was a beehive in the o
  the oak tree. He ate the bee



- Henry Squirrel was thirsty.
  river bank where his good f ng.
  Henry slipped and fell in th ed.
  The End.

- Once upon a time there was ain crow. One day the crow
  was sitting in his tree, holdi is mouth. He noticed that he
  was holding the piece of che , and swallowed the cheese.
  The fox walked over to the c

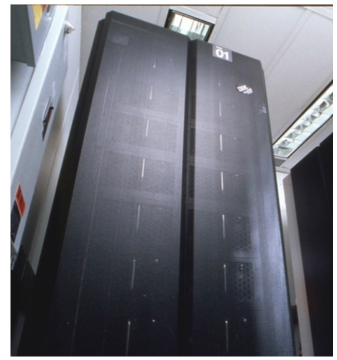


[Shank, Tale-Spin System, 1984]


. e4 c6 2. d4 d5 3. Nc3 dxe4

Game Agents

- Classic Moment: May, '97: Deep Blue vs. Kasparov
  - First match won against world champion
  - "Intelligent creative" play
  - 200 million board positions per second
  - Humans understood 99.9 of Deep Blue's moves
  - Can do about the same now with a PC cluster



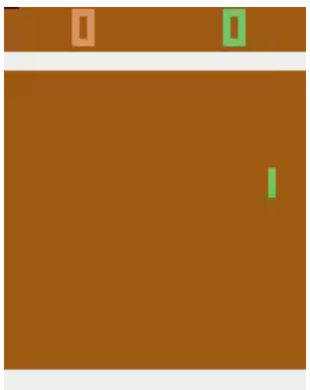
- 1996: Kasparov Beats Deep Blue
  "I could feel --- I could smell --- a new kind of intelligence across the table."

- 1997: Deep Blue Beats Kasparov
  "Deep Blue hasn't proven anything."

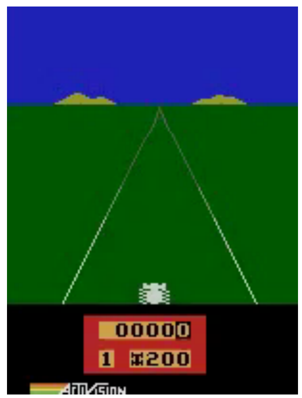
Text from Bart Selman, image from IBM's Deep Blue pages

Game Agents

- Reinforcement learning

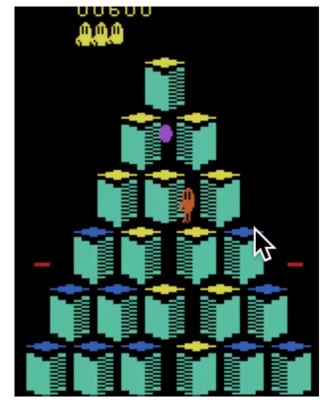


Pong



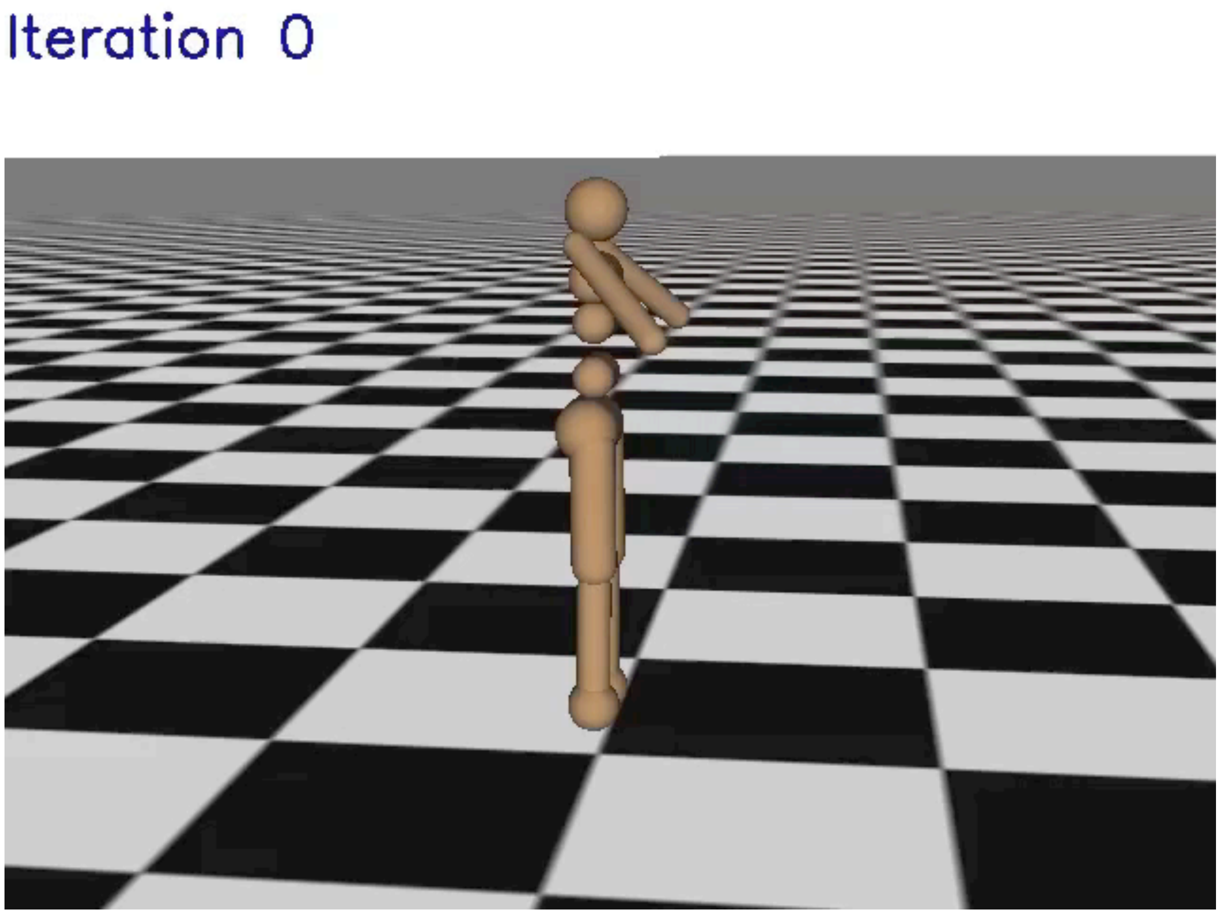
Enduro

Beamrider



Q*bert

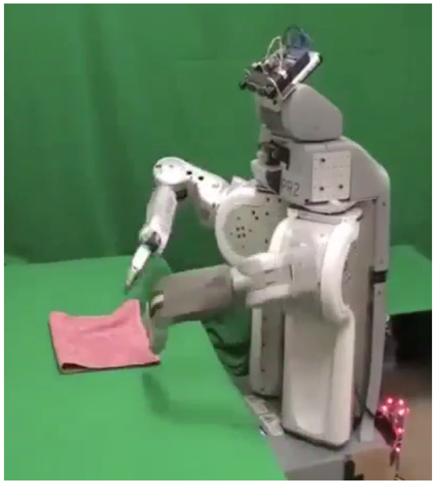
Simulated Agents

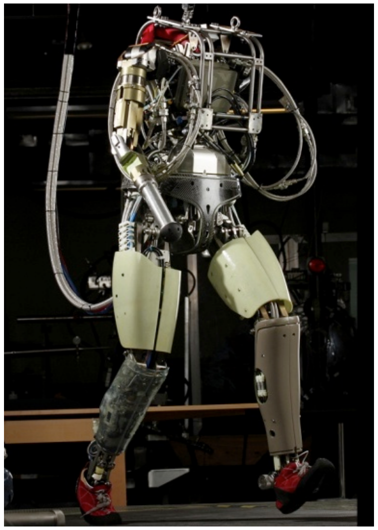


[Schulman, Moritz, Levine, Jordan, Abbeel, ICLR 2016]

Robotics

- Robotics
  - Part mech. eng.
  - Part AI
  - Reality much harder than simulations!
- Technologies
  - Vehicles
  - Rescue
  - Help in the home
  - Lots of automation…
- In this class:
  - We ignore mechanical aspects
  - Methods for planning
  - Methods for control





Robots

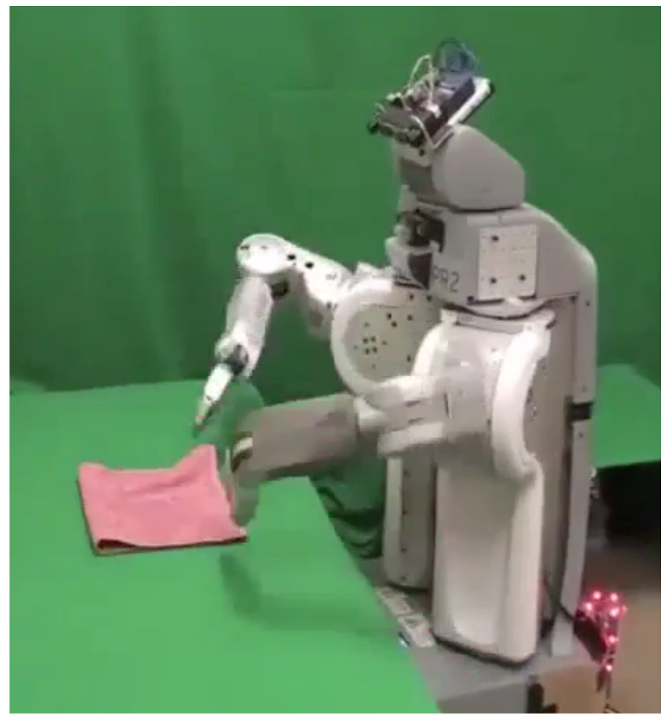
Robots

[Levine*, Finn*, Darrell, Abbeel, JMLR 2016]

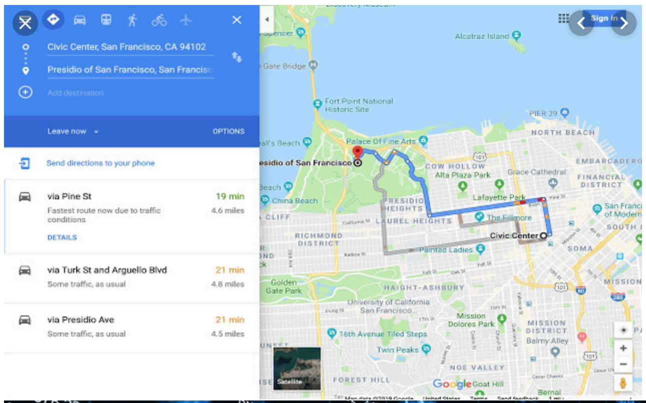
Utility?

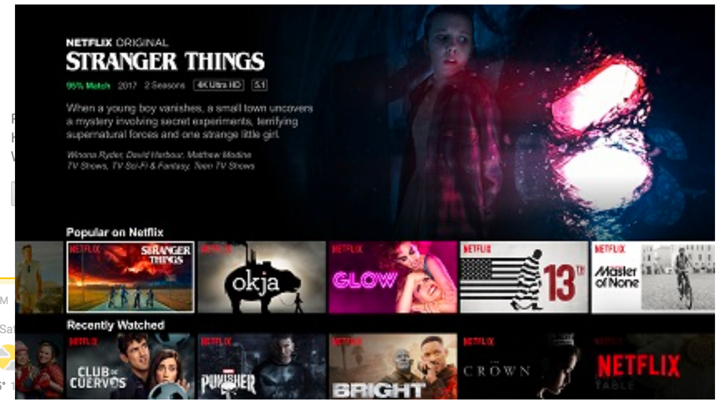
Clear utility function

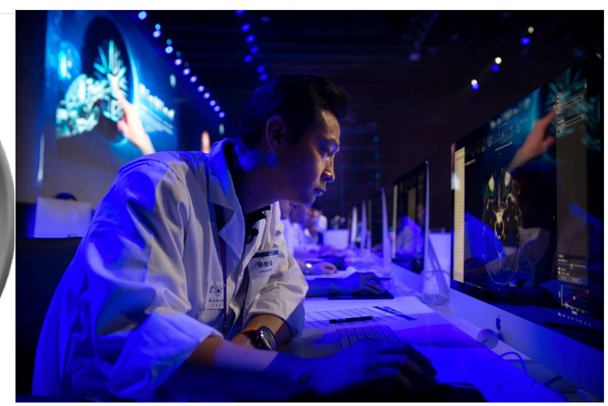
Not so clear utility function



Tools for Predictions & Decisions

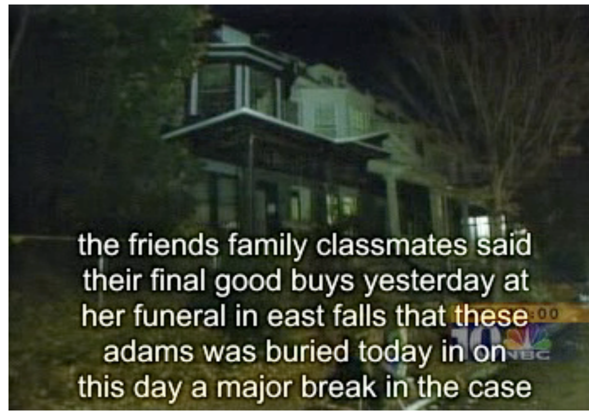






Natural Language

- Speech technologies (e.g. Siri)
  - Automatic speech recognition (ASR)
  - Text-to-speech synthesis (TTS)
  - Dialog systems



- Language processing technologies
  - Question answering
  - Machine translation

"Il est impossible aux journalistes de rentrer dans les régions tibétaines"

Bruno Philip, correspondant du "Monde" en Chine, estime que les journalistes de l'AFP qui ont été expulsés de la province tibétaine du Qinghai "n'étaient pas dans l'illégalité".

Les faits Le dalaï-lama dénonce l'"enfer" imposé au Tibet depuis sa fuite, en 1959

Vidéo Anniversaire de la rébellion

"It is impossible for journalists to enter Tibetan areas"

Philip Bruno, correspondent for "World" in China, said that journalists of the AFP who have been deported from the Tibetan province of Qinghai "were not illegal."

Facts The Dalai Lama denounces the "hell" imposed since he fled Tibet in 1959

Video Anniversary of the Tibetan rebellion: China on guard

  - Web search
  - Text classification, spam filtering, etc…

Computer Vision

"man in black shirt is playing guitar."

"construction worker in orange safety vest is working on road."

"two young girls are playing with lego toy."

"boy is doing backflip on wakeboard."

"girl in pink dress is jumping in air."

"black and white dog jumps over bar."

"young girl in pink shirt is swinging on swing."

"man in blue wetsuit is surfing on wave."

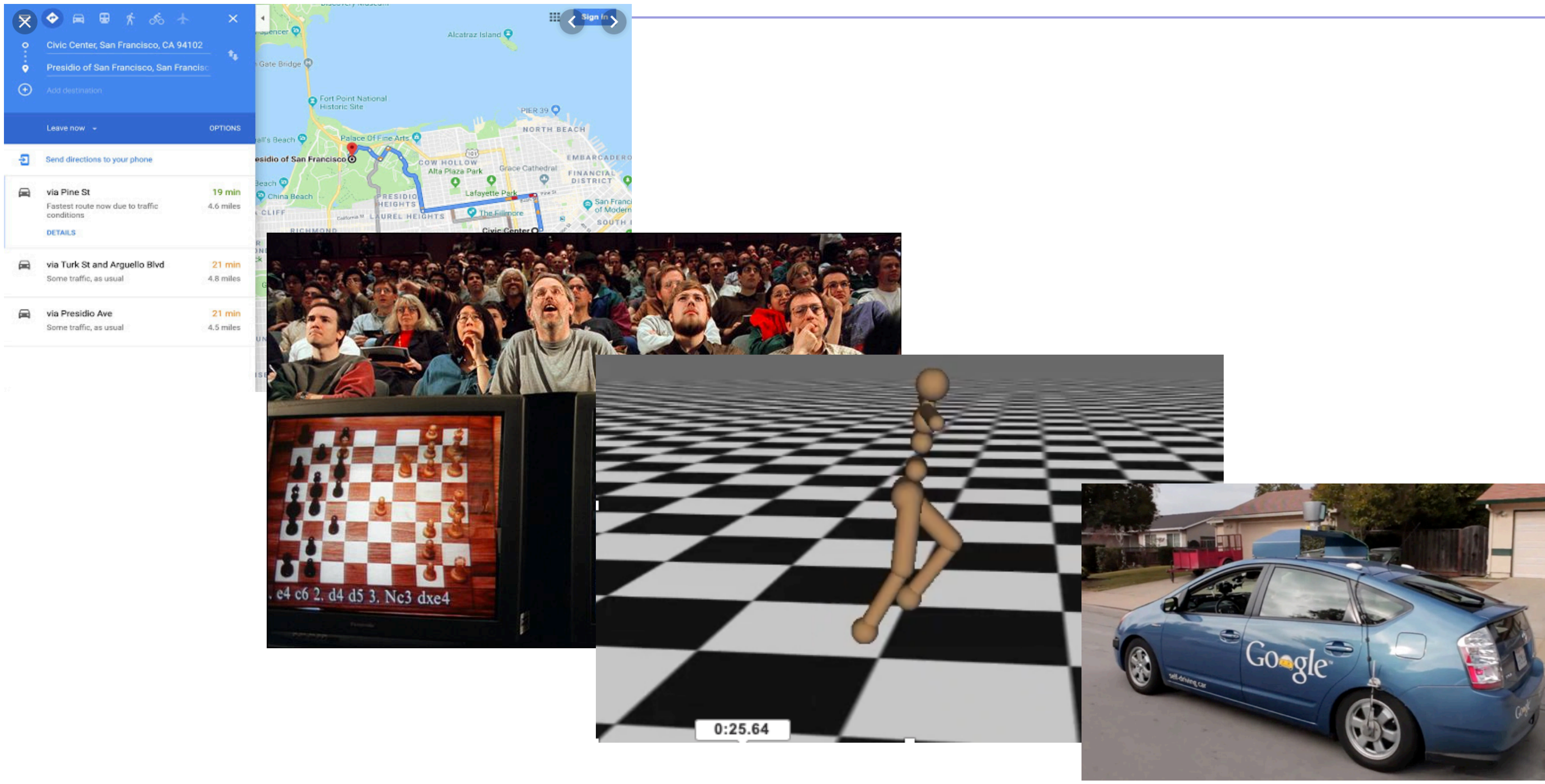
Should I take 188?

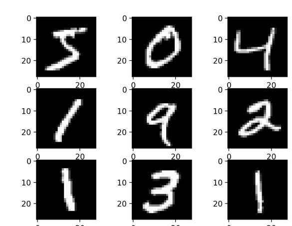
- Yes, if you want to know how to design rational agents!
  - CS 188 also teaches you a different way of thinking.
  - Powerful ideas without too much math rigor
- Disclaimer: If you're interested in making yourself more competitive for industry jobs, courses like CS 189, CS 182, and various courses in Data Science and the School of Information are a much better fit.

Topics

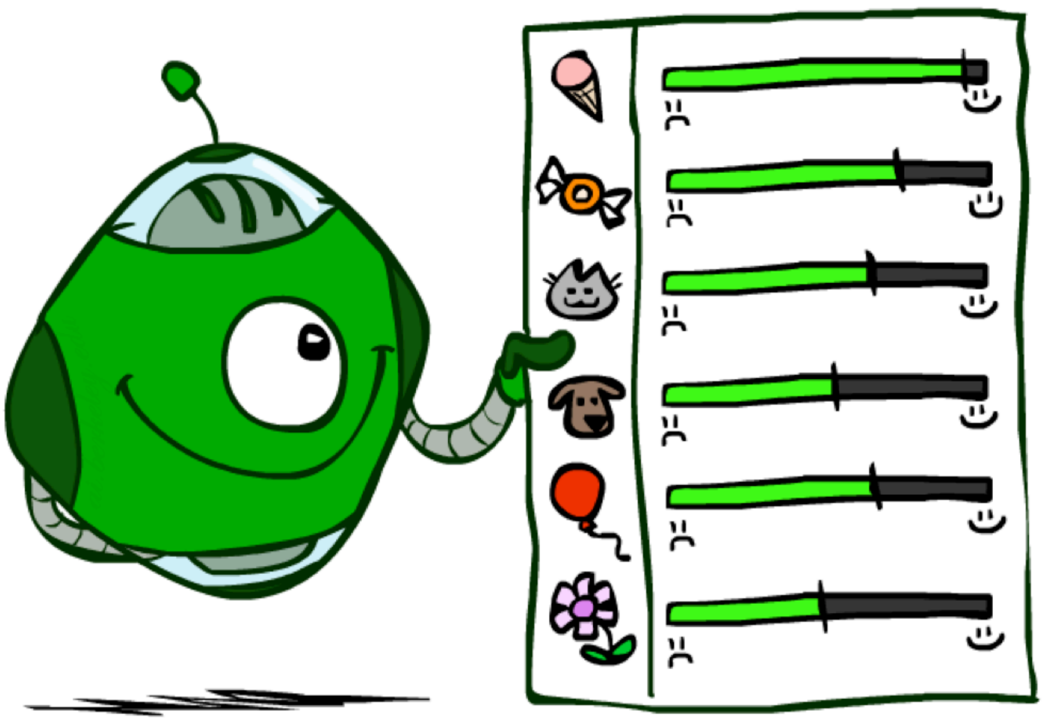
- Utilities and Rationality, Search and Planning
- Constraint Satisfaction Programming
- Game Trees, Minimax, Pruning, Monte-Carlo Tree Search
- Probabilistic Inference, Bayesian Networks, Markov Models
- Decision Networks and Value of Perfect Information
- Markov Decision Processes and Reinforcement Learning
- Supervised Machine Learning, MLE and MAP
- Optimization Theory, Neural Networks
- Survey of Modern Problems and Topics (Guest Lectures!)

The kinds of AI problems in CS 188





Utilities



Maximum Expected Utility

- Principle of maximum expected utility:
  - A rational agent should choose the action that maximizes its expected utility, given its knowledge

- Questions:
  - Where do utilities come from?
  - How do we know such utilities even exist?
  - How do we know that averaging even makes sense?
  - What if our behavior (preferences) can't be described by utilities?

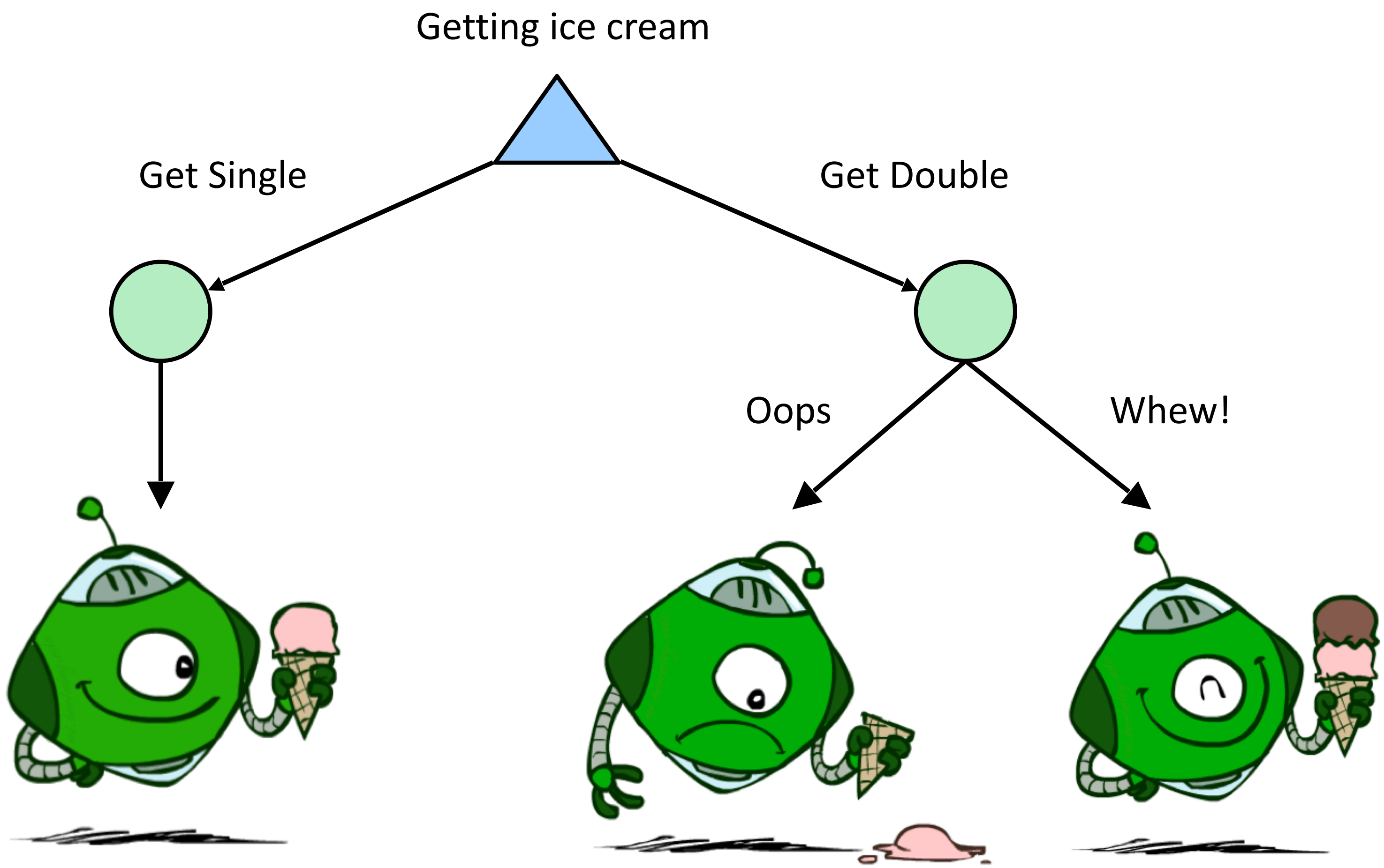
Utilities

- Utilities are functions from outcomes (states of the world) to real numbers that describe an agent's preferences

- Where do utilities come from?
  - In a game, may be simple (+1/-1)
  - Utilities summarize the agent's goals
  - Theorem: any "rational" preferences can be summarized as a utility function

- We hard-wire utilities and let behaviors emerge
  - Why don't we let agents pick utilities?
  - Why don't we prescribe behaviors?

Utilities: Uncertain Outcomes



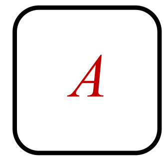
Preferences

- An agent must have preferences among:
  - Prizes: A, B, etc.
  - Lotteries: situations with uncertain prizes

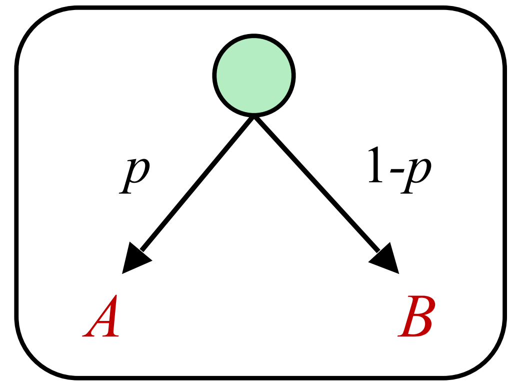
    $L = [p, A;\ (1\text{-}p), B]$

- Notation:
  - Preference: $A > B$
  - Indifference: $A \sim B$

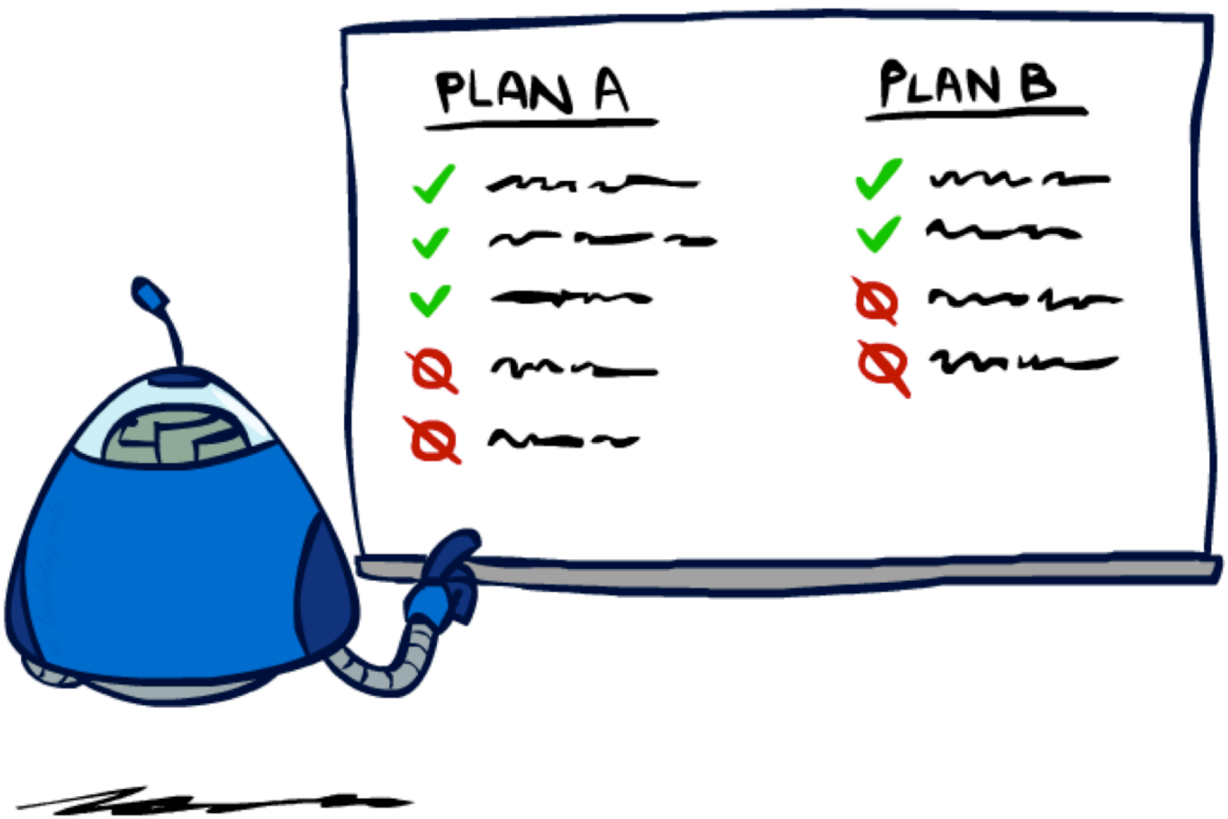
A Prize



A Lottery



Rationality

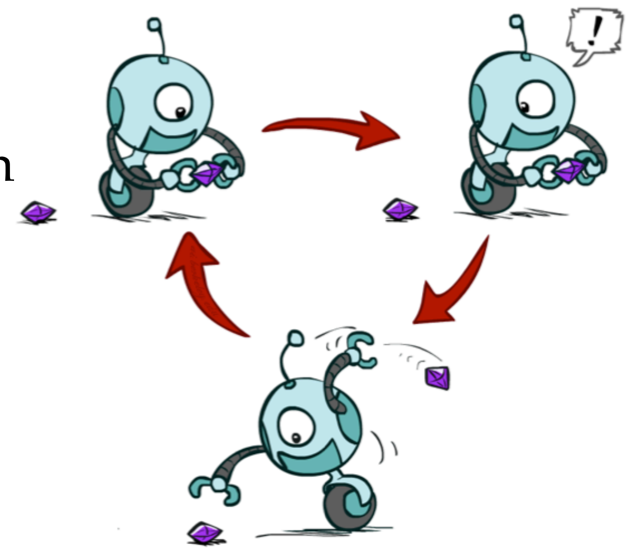


Rational Preferences

- We want some constraints on preferences before we call them rational, such as:

Axiom of Transitivity: $(A > B) \wedge (B > C) \Rightarrow (A > C)$

- For example: an agent with intransitive preferences can be induced to give away all of its money
  - If B > C, then an agent with C would pay (say) 1 cent to get B
  - If A > B, then an agent with B would pay (say) 1 cent to get A
  - If C > A, then an agent with A would pay (say) 1 cent to get C



Rational Preferences

The Axioms of Rationality

Orderability:

$$(A > B) \vee (B > A) \vee (A \sim B)$$

Transitivity:

$$(A > B) \wedge (B > C) \Rightarrow (A > C)$$

Continuity:

$$(A > B > C) \Rightarrow \exists p\ [p, A;\ 1\text{-}p, C] \sim B$$

Substitutability:

$$(A \sim B) \Rightarrow [p, A;\ 1\text{-}p, C] \sim [p, B;\ 1\text{-}p, C]$$

Monotonicity:

$$(A > B) \Rightarrow$$
$$(p \geq q) \Leftrightarrow [p, A;\ 1\text{-}p, B] \geq [q, A;\ 1\text{-}q, B]$$

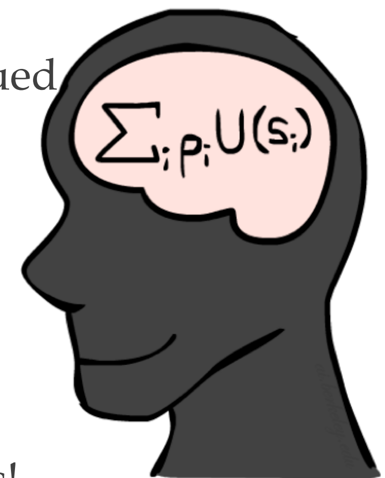
Theorem: Rational preferences imply behavior describable as maximization of expected utility

MEU Principle

- Theorem [Ramsey, 1931; von Neumann & Morgenstern, 1944]
  - Given any preferences satisfying these constraints, there exists a real-valued function U such that:

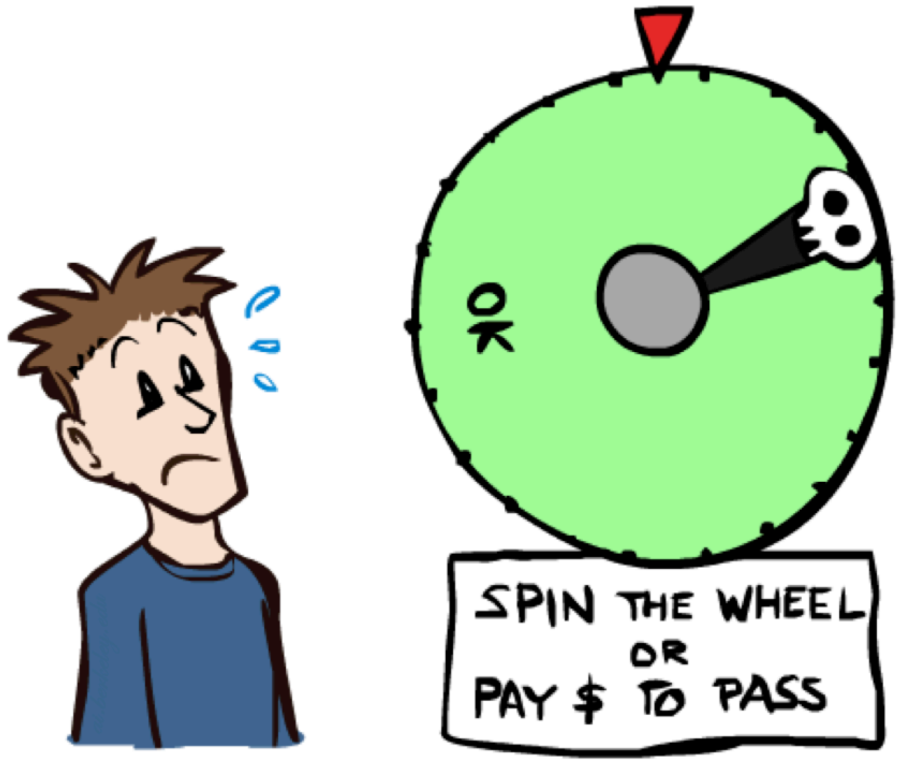
$$U(A) \geq U(B) \iff A \succeq B$$

$$U([p_1, S_1; \ldots ; p_n, S_n]) = p_1 U(S_1) + \ldots + p_n U(S_n)$$



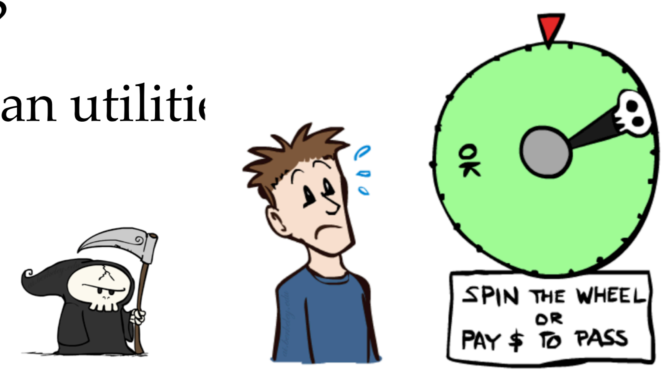
  - I.e. values assigned by U preserve preferences of both prizes and lotteries!
- Maximum expected utility (MEU) principle:
  - Choose the action that maximizes expected utility
  - Note: rationality does ***not*** require representing or manipulating utilities and probabilities
    - E.g., a lookup table for perfect tic-tac-toe

Human Utilities

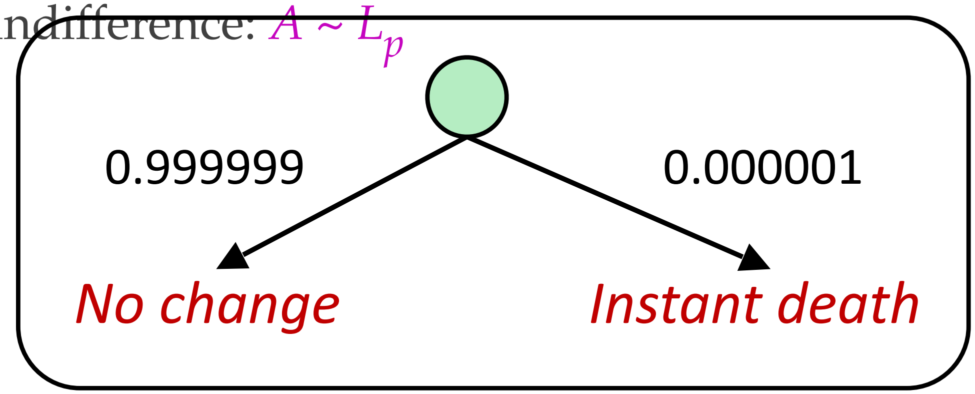


Human Utilities

- Utilities map states to real numbers. Which numbers?
- Standard approach to assessment (elicitation) of human utilities:

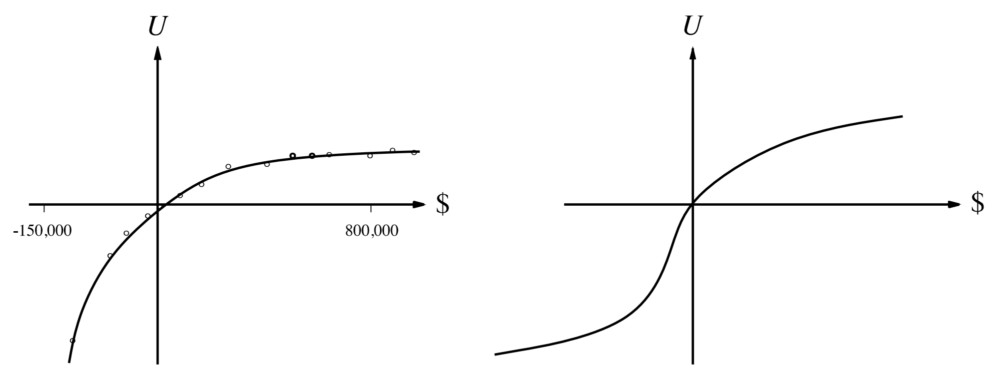


  - Compare a prize A to a ***standard lottery*** L_p between
    - "best possible prize" $u_{\top}$ with probability p
    - "worst possible catastrophe" $u_{\bot}$ with probability $1-p$
  - Adjust lottery probability p until indifference: $A \sim L_p$
  - Resulting p is a utility in $[0,1]$



Money

- Money ***does not*** behave as a utility function, but we can talk about the utility of having money (or being in debt)
- Given a lottery $L = [p, \$X; (1\text{-}p), \$Y]$
  - The ***expected monetary value*** $\mathrm{EMV}(L) = pX + (1\text{-}p)Y$
  - The utility is $U(L) = pU(\$X) + (1\text{-}p)U(\$Y)$
  - Typically, $U(L) < U(\ \mathrm{EMV}(L)\ )$
  - In this sense, people are ***risk-averse***
  - E.g., how much would you pay for a lottery ticket $L=[0.5, \$10{,}000;\ 0.5, \$0]$?
  - The ***certainty equivalent*** of a lottery $\mathrm{CE}(L)$ is the cash amount such that $\mathrm{CE}(L) \sim L$
  - The ***insurance premium*** is $\mathrm{EMV}(L) - \mathrm{CE}(L)$
  - If people were risk-neutral, this would be zero!



(a) (b)