CS 182 Lecture 1: Introduction

Professors: Ariel Procaccia and **Stephanie Gil**

Email: sgil@seas.harvard.edu

Prof. Gil Office hours: Wednesdays 2:30-3:30p

Your Teaching Team

- Professor Stephanie Gil (SEC 4.211)
  - Robotics research
    - Multi-robot coordination and control
    - Applications:
      - Coordinated drone delivery
      - Robotic search and rescue
      - Autonomous car fleet control
- Professor Ariel Procaccia (SEC 5.411)
  - AI algorithms, economics, and society

Your Teaching Team (cont.)

TF Team

- Max Guo (lead TF)
- Lauren Cooke
- Catherine Cui
- Eric Helmhold
- Mujin Kwun
- Christopher Lee
- Janani Sekar
- Sanjana Singh

Communication

- **Class Ed Discussion!** Please use this forum to post your questions, we will be closely monitoring this.
  - Participation in class and on this forum will count towards your participation grade.
  - See course website for sign up instructions
- **Email** posted on the course website
- **Office hours** posted on the course website

Resources

- **Textbook:** “Artificial Intelligence: A Modern Approach” by Stuart Russell and Peter Norvig, *Fourth Edition*
- **Poll everywhere:** used for in-class polls. Instructions for sign up are on course website
- **Course slides, refs, and videos:** posted on the course website
- **Sections:** led by the TF team and will cover the lecture material from that week
  - Will cover the material taught each week
  - Start on Sept 8th (schedule posted on the course website)
  - Section attendance is optional (meant to provide additional support and review)

Course Structure

Attendance	**10%**
Problem sets (5)	**50%**
Midterm	**15%**
Final Exam	**25%**

Course Topics (Full list on course website)

- Uninformed search
- Informed search
- Motion planning
- Constraint satisfaction problems
- Multi-robot systems

Search and Planning
Lecturer: Gil

- Intro to optimization
- Game theory
- AI game playing
- Stackelberg security games
- Bayesian networks

Optimization and Games
Lecturer: Proccacia

- Markov Decision Processes
- Reinforcement learning
- Decision trees
- Linear classification
- Neural networks
- Ethics

Learning and Uncertainty
Lecturer: Proccacia

Course Pre-requisites

- Computer programming experience and fluency with Python coding is expected.

- Students must have previously taken Statistics 110 (Probability) or an equivalent course.

- Experience with Python programming and a good understanding of time complexity (including big O notation) are assumed.

- *Note on working together:*
  - Submissions must be individual to receive grade
  - Exams are closed book, no collaboration (unless otherwise stated)

What is Artificial Intelligence?

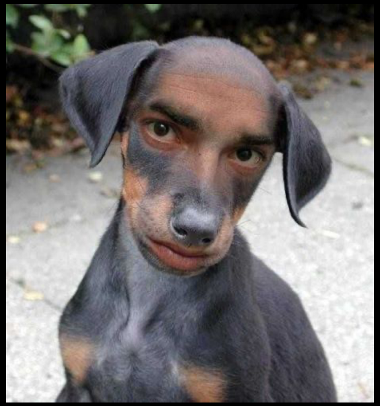
Example: Vision and Object Recognition

Clap once every time you see a dog, and only a dog
You only have 0.8 seconds per picture…

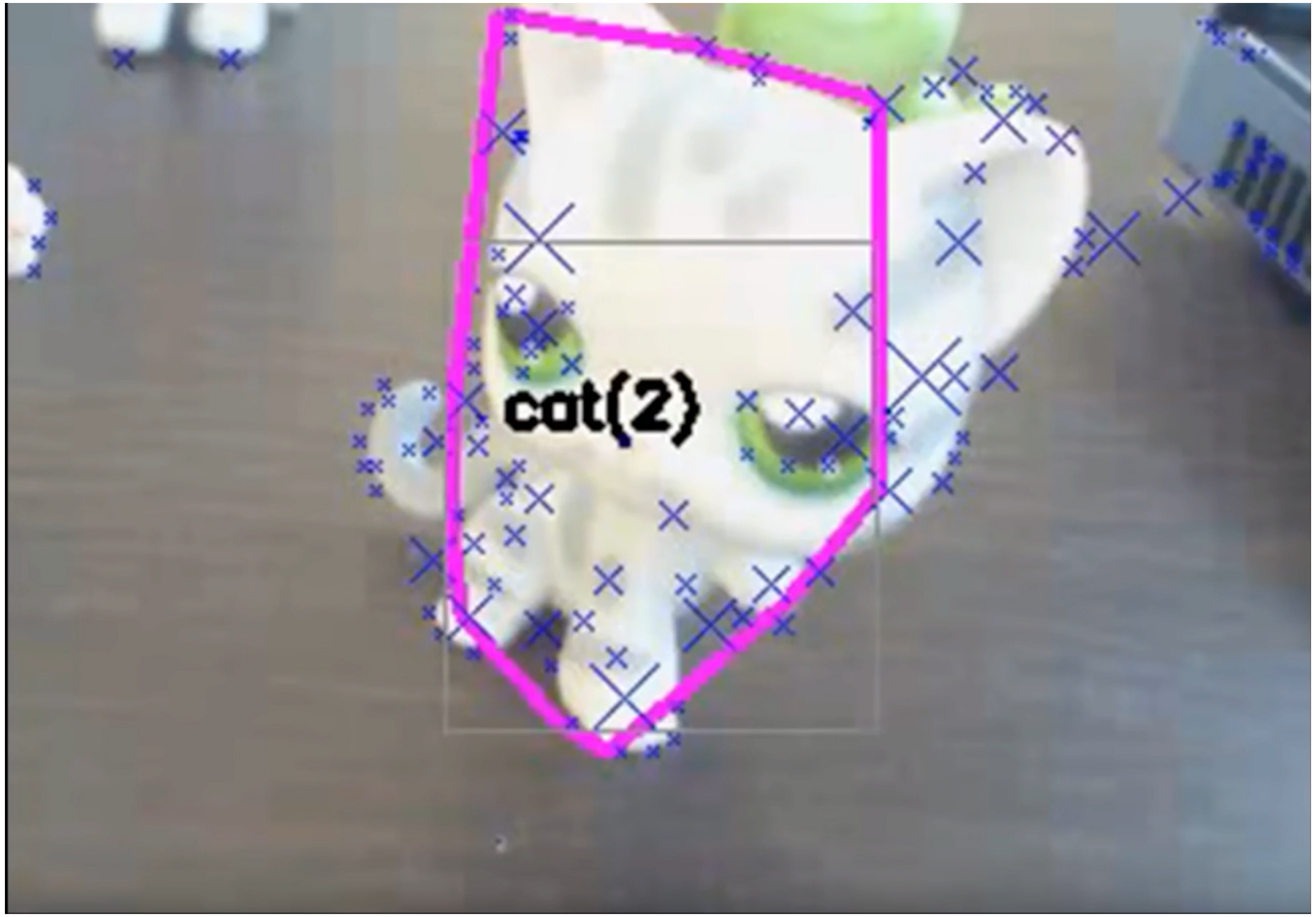
Ready

Set

Go



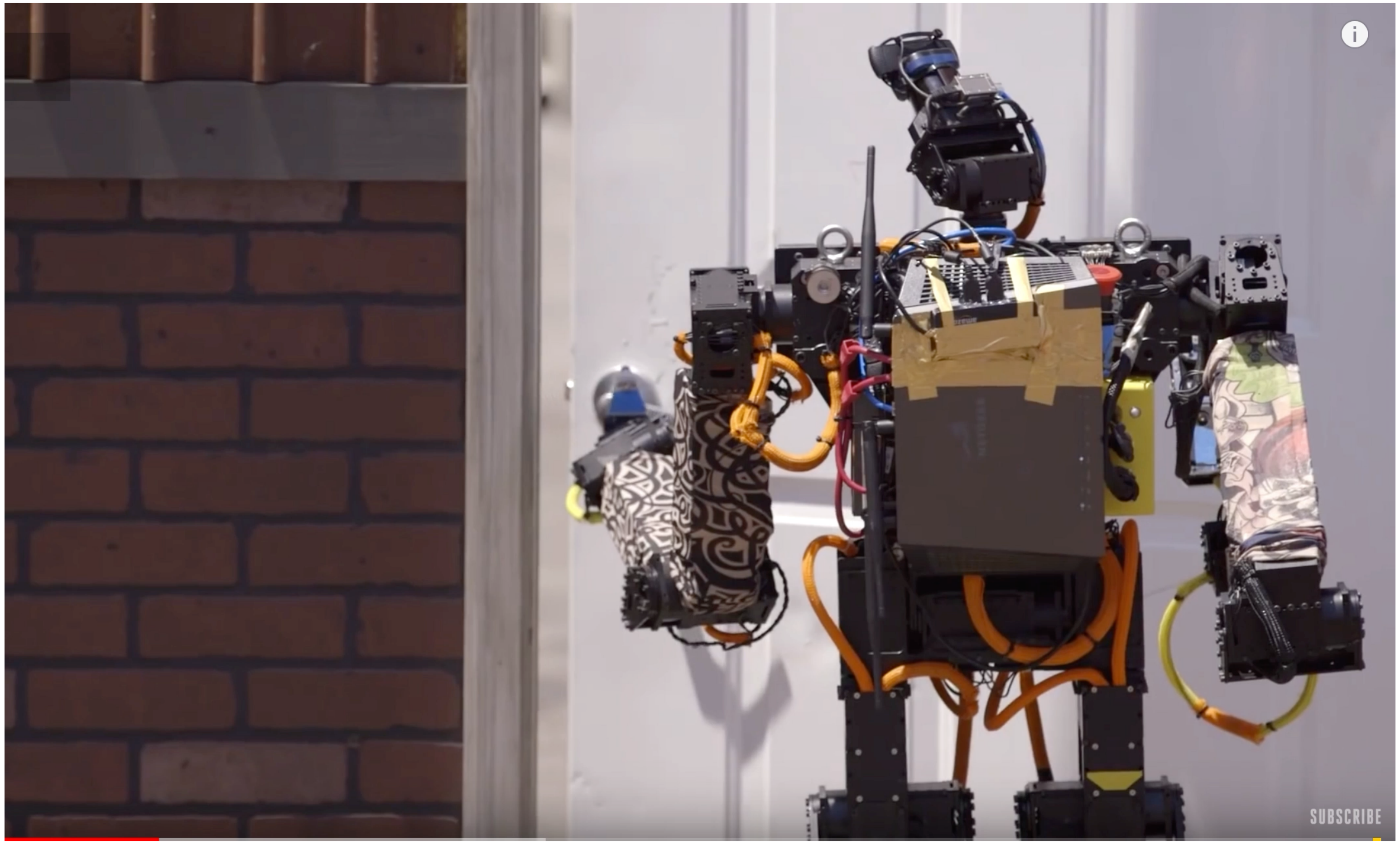
Example: Vision and Object Recognition



Video Excerpt from: Berkeley CS 188 Lecture 1

Example: Object Recognition and Decision on Action

- DARPA Robotics Challenge

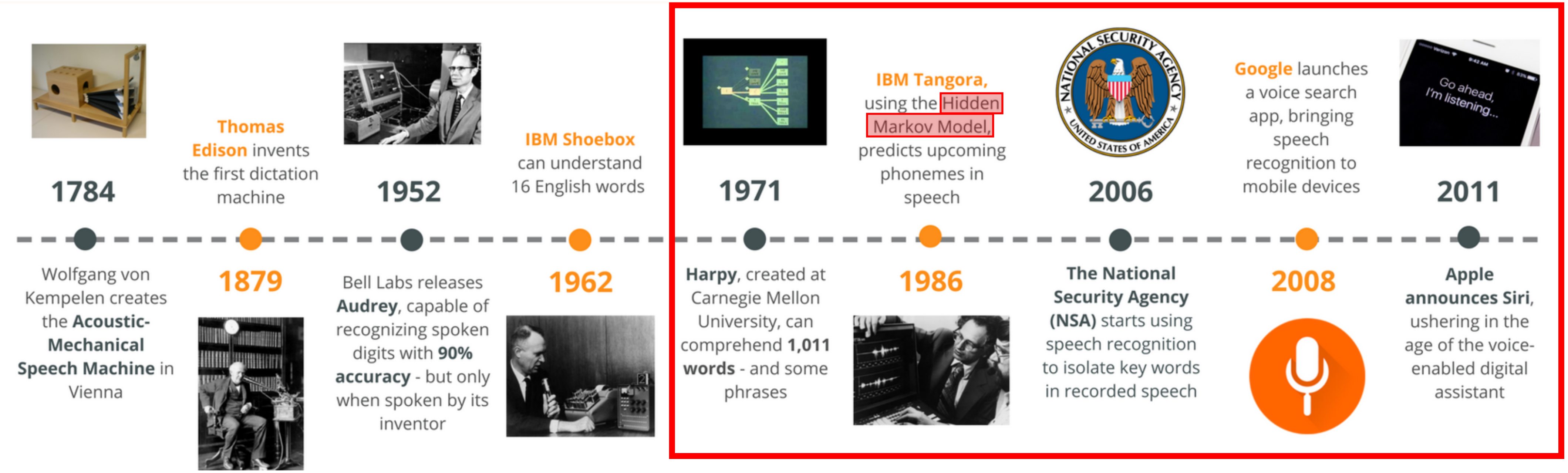


Video credit: The Verge

Example: Natural Language

The Past, Present, and Future of Speech Recognition Technology

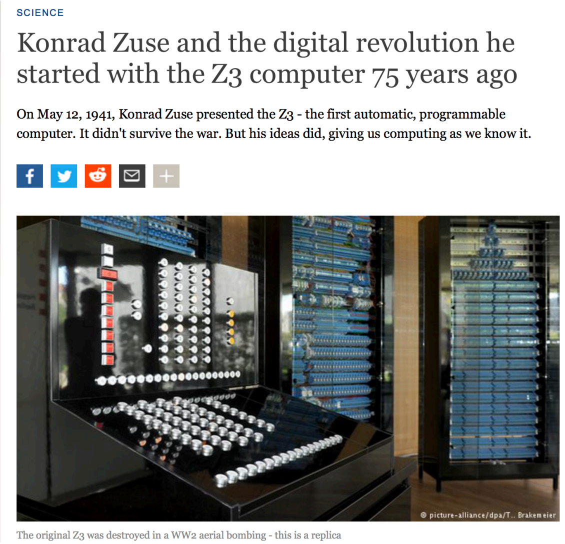
Voice is the future. The world's technology giants are clamoring for vital market share, with ComScore projecting that "50% of all searches will be voice searches by 2020."



Source: TheStartup.com

Example: Games

- Game playing was one of the first tasks undertaken by AI
- Notable scientists working on games through history:
  - **Konrad Zuse** - the inventor of the first programmable computer and first programming language



SCIENCE

Konrad Zuse and the digital revolution he started with the Z3 computer 75 years ago

On May 12, 1941, Konrad Zuse presented the Z3 - the first automatic, programmable computer. It didn't survive the war. But his ideas did, giving us computing as we know it.

The original Z3 was destroyed in a WW2 aerial bombing - this is a replica

Example: Games

- Game playing was one of the first tasks undertaken by AI
- Notable scientists working on games through history:
  - **Konrad Zuse** - the inventor of the first programmable computer and first programming language
  - **Claude Shannon** – the inventor of information theory
  - **Norbert Weiner** – the creator of modern control theory
  - **Alan Turing** – widely considered the father of theoretical computer science and artificial intelligence

Some Important Games in AI History

- Checkers
- 1994, an AI beat Tinsely, the 40 year undefeated world champion

Some Important Games in AI History

- Chess

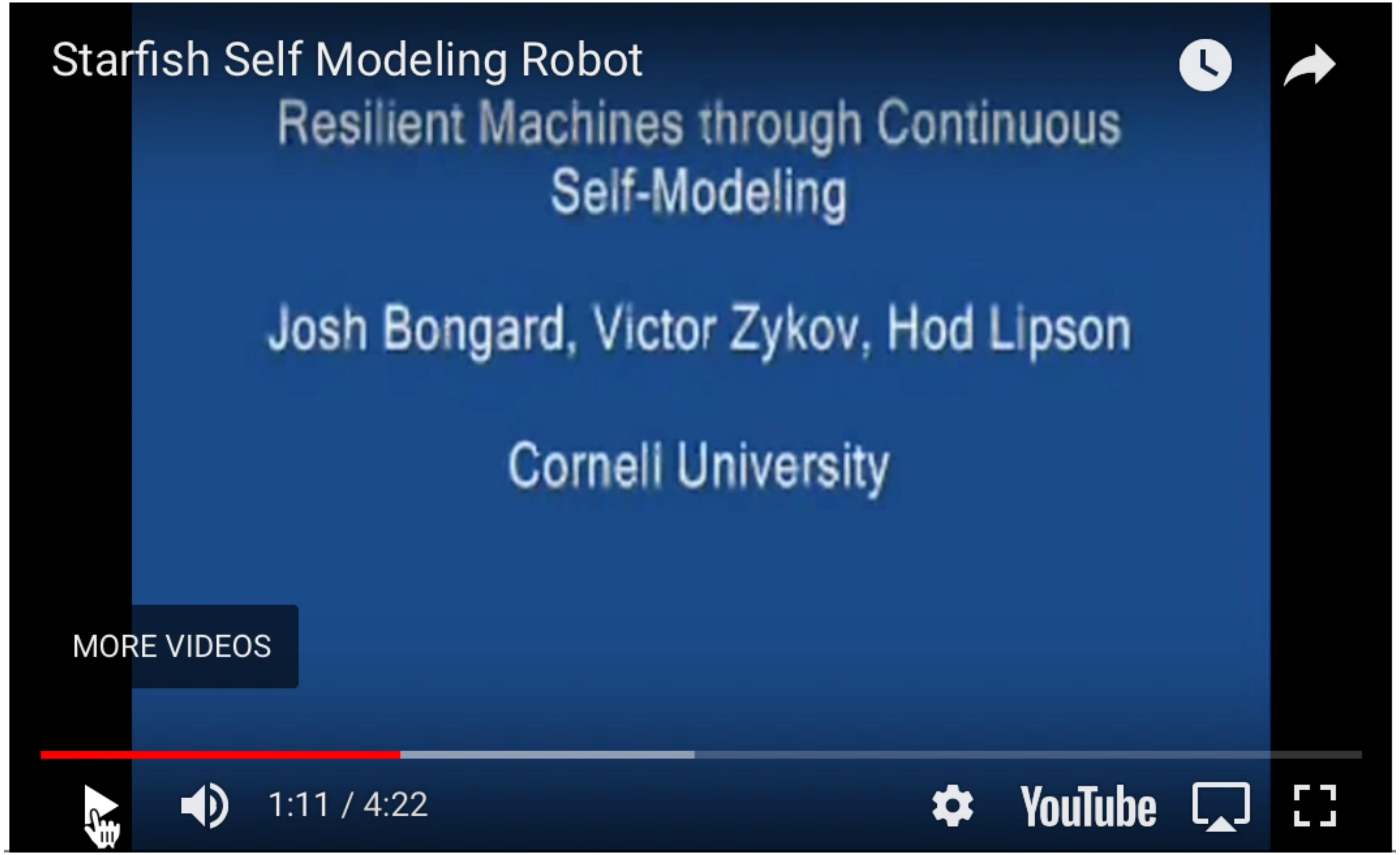
IBM's Deep Blue examined 200 million moves every second!

- World Champion Kasparov lost to IBM's Deep Blue in 1997

Example: Learning to Walk

- Possible actions are learned (model)
- Driven by a reward function

Example: Learning to Walk



Video credit: Hod Lipson research group Cornell University

Example: Learning how to Fly

Video credit: Andrew Ng research group

What was Common to all of these Examples?

- Agent
  - Software
  - Robot

  The Decision-making agent

- Decision Making
  - Correctly recognize object
  - Diffuse dangerous situation
  - Win game
  - Move a far distance

  Utility
  "AI should maximize the expected utility"

- Environment
  - Other player
  - Friction, gravity
  - Noise

  Uncertainty
  Affects the decision making

Designing Rational Agents

- An **agent** is an entity that *perceives its environment* and *acts upon that environment*

- A **rational agent** selects actions that maximize its expected utility

- The *percepts, environment*, and *action space* dictate techniques for selecting rational actions

- **This course:**
  - How to *model* the environment (utility, actions, state)
  - Techniques for making decisions that maximize utility

This Course

- We will not focus on applications (we will use these as motivating examples)

- We will focus on creating a *framework* for decision making that maximizes expected utility

- You will get some hands-on practice on the theory as well as some programming experience through the problem sets (HW)

Intro to Polls Everywhere

- Q: What really is this course all about?

- It’s all about neural networks!
- It’s all about encoding *human intelligence*
- It’s about creating a framework for intelligent (rational) decision making

Agent and Environment in AI

- Agent and Environment
  - Agent - anything that obtains information about the environment and acts upon the information
  - Environment – the “world” in which the agent exists and which externally influences the agents’ behavior
- Example

What do we care to optimize?

What affects decisions, behavior, cost?

What actions can the agent take?

What feedback goes to the agent?

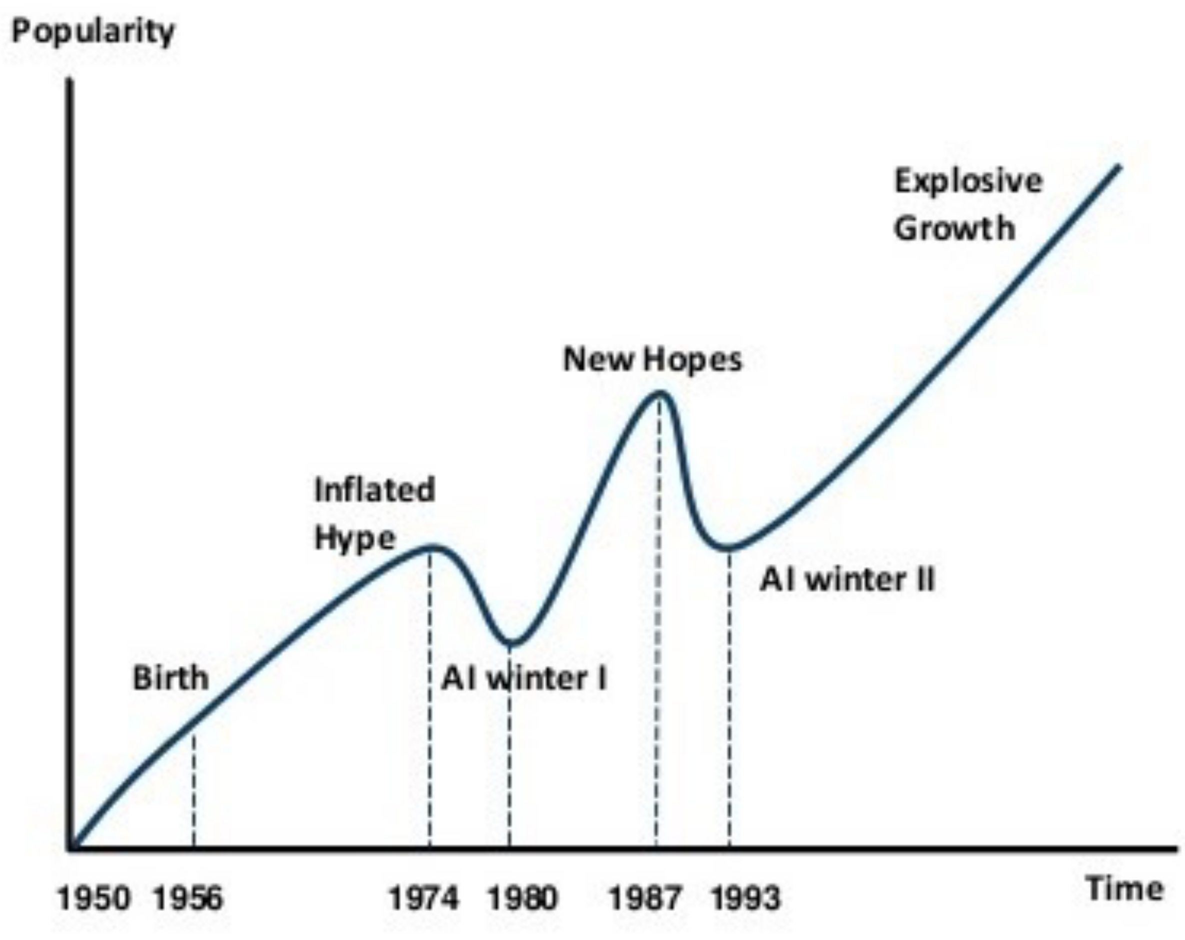
Agent Type	**Performance Measure**	**Environment**	**Actuators**	**Sensors**
Taxi driver	Safe, fast, legal, maximize profits	Roads, other traffic, customers	Steering, accelerator, brake, horn	Cameras, GPS, speedometer

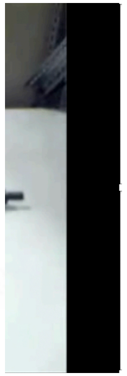
Watershed in AI

- Two element lvability of a problem in a

Compu

IBM's Deep Blue examined 200 million moves every second





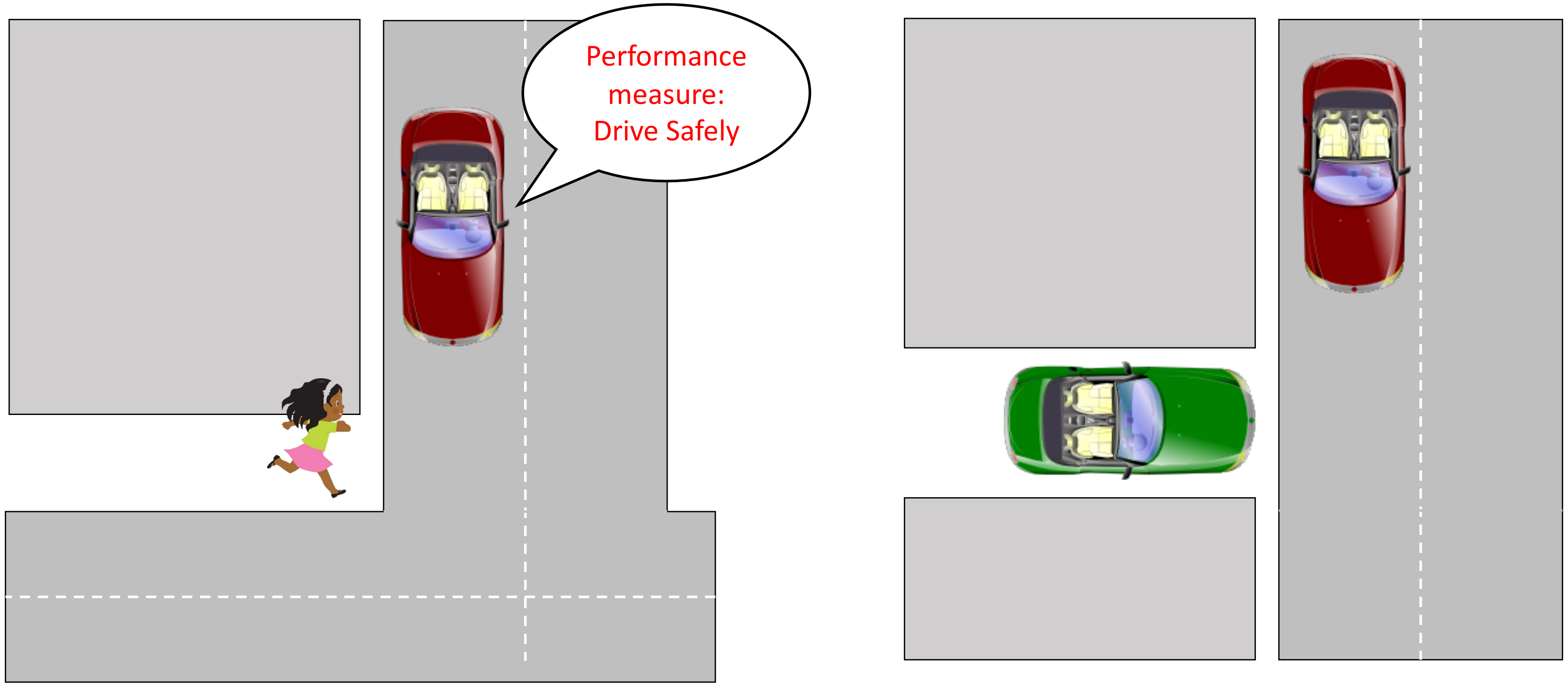
Types of Problems and Environments

Important! This will largely influence the type of model, search, and solution strategy available to us:

- Fully vs. partially observable
- Deterministic vs. stochastic
- Episodic vs. sequential (i.e. what role does *history* play)
- Static vs. dynamic
- Discrete vs. continuous
- Single-agent vs. multiagent

Types of Problems and Environments

- Fully vs. partially observable



What information do I have about the world? My state? The success of my action?

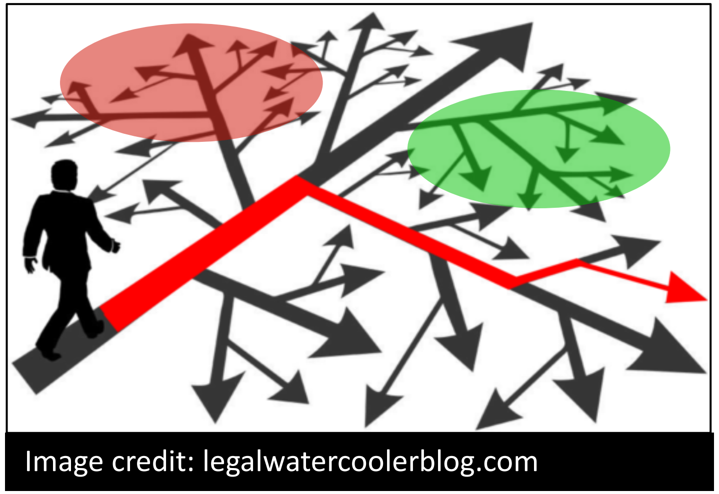
Types of Problems and Environments

- Fully vs. partially observable
- Deterministic vs. stochastic

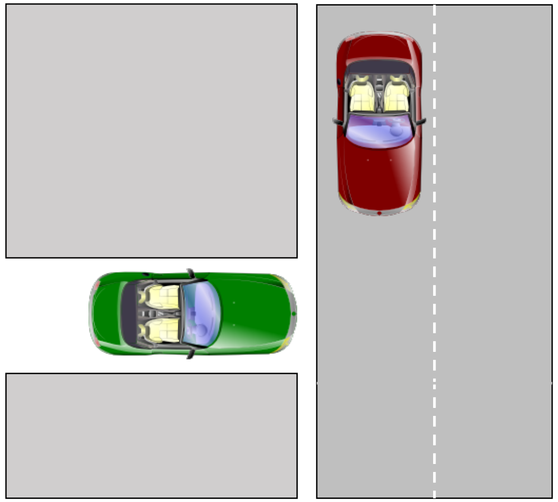
Is there a chance that Tesla gets negative press?

Is there a chance that government adds incentives?

Is there a chance that my breaks could fail?



Stochasticity in the next state given my current state



Stochasticity in my action space

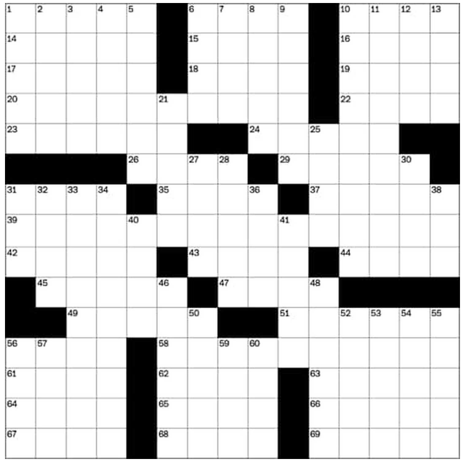
Types of Problems and Environments

- Fully vs. partially observable
- Deterministic vs. stochastic
- Episodic vs. sequential

  (i.e. what role does my *history* play)

Types of Problems and Environments

- Fully vs. partially observable
- Deterministic vs. stochastic
- Episodic vs. sequential (i.e. what role does *history* play)
- Static vs. dynamic
  - Is the environment changing while the agent makes a decision?



Types of Problems and Environments

- Fully vs. partially observable
- Deterministic vs. stochastic
- Episodic vs. sequential (i.e. what role does *history* play)
- Static vs. dynamic
- Discrete vs. continuous
    - What does my action space look like?
    - What about my state space?
    - Examples?

Types of Problems and Environments

- Fully vs. partially observable
- Deterministic vs. stochastic
- Episodic vs. sequential (i.e. what role does *history* play)
- Static vs. dynamic
- Discrete vs. continuous
- Single-agent vs. multiagent
  - Does one agents' decisions rely on another agents' decisions?

Types of Problems and Environments

- Fully vs. partially observable
- Deterministic vs. stochastic
- Episodic vs. sequential (i.e. what role does *history* play)
- Static vs. dynamic
- Discrete vs. continuous
- Single-agent vs. multiagent

The type of problem and environment will largely determine what models, methods, and solutions are available for us

➢ Often determines the *solvability* of a problem

Types of Agents

- Reflex agents – agent acts according to a rule whose condition matches the current state

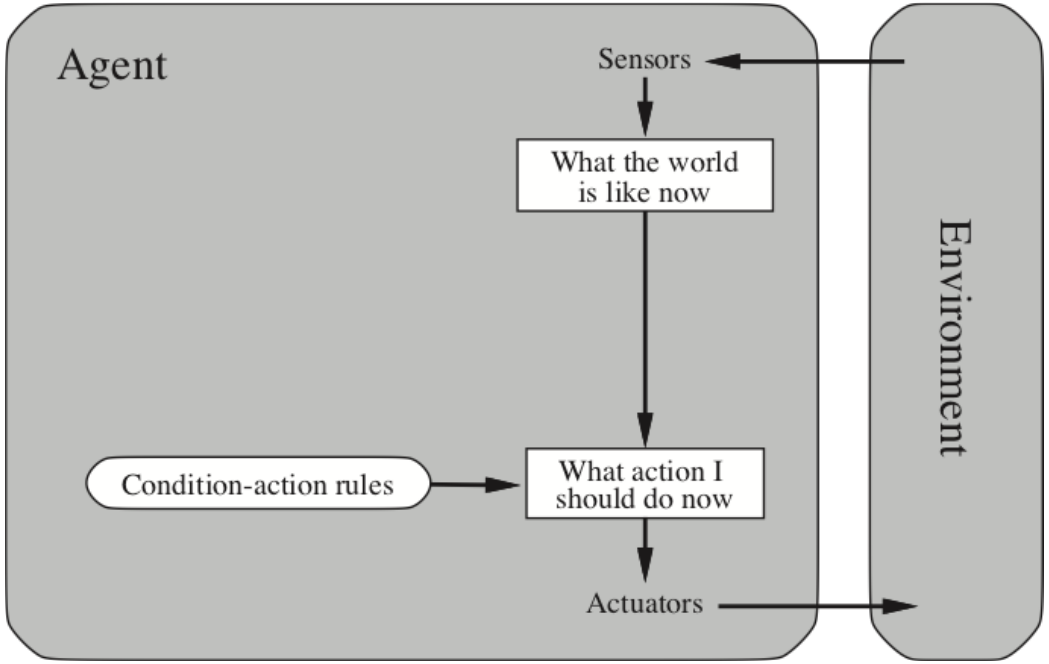


Fig 2.9 from Russell et. al.

Types of Agents

- Model-based reflex agents – agent keeps track of the part of the world that it can't see now (uncertainty, future evolution of the state, etc)

Model: *Gives predictive capability*

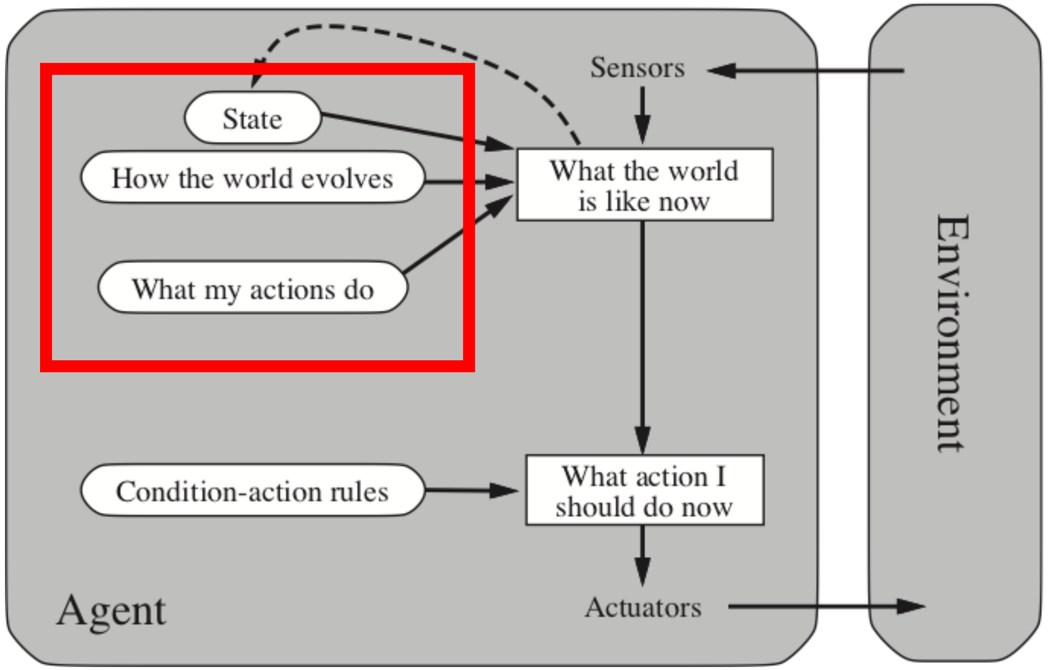


Fig 2.11 from Russell et. al.

Types of Agents

- <u>Goal-based agent</u> – Chooses actions that will eventually lead to achievement of goal

Goal drives action

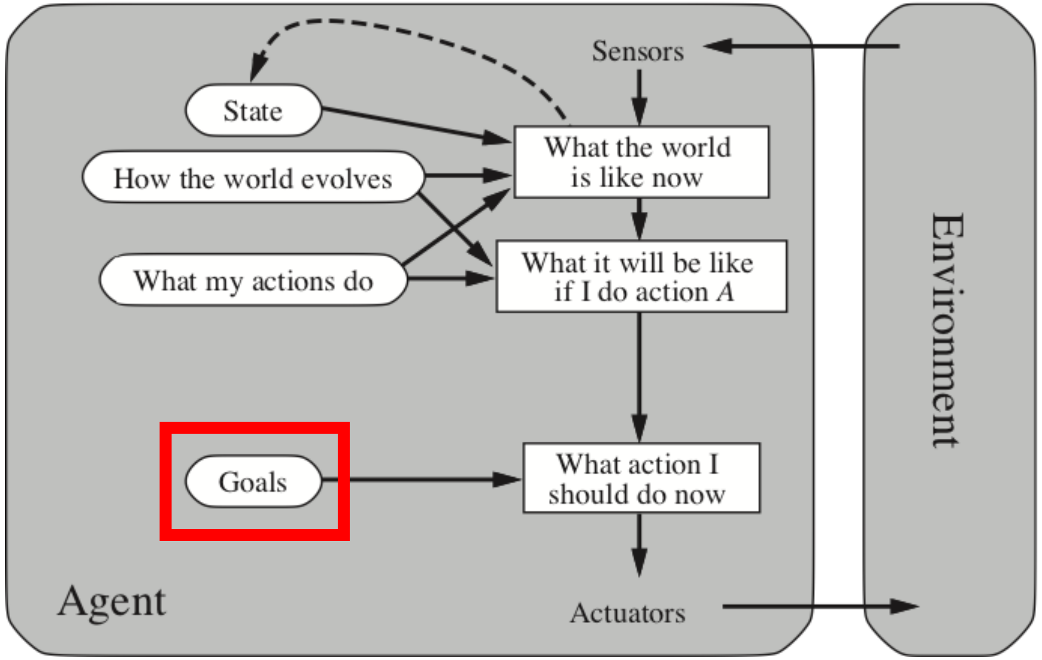


Fig 2.13 from Russell et. al.

There can be many paths, or sequences of actions, leading to a goal.

Types of Agents

- Utility-based agent – Utility function maps a state, or sequence of states, onto a real number

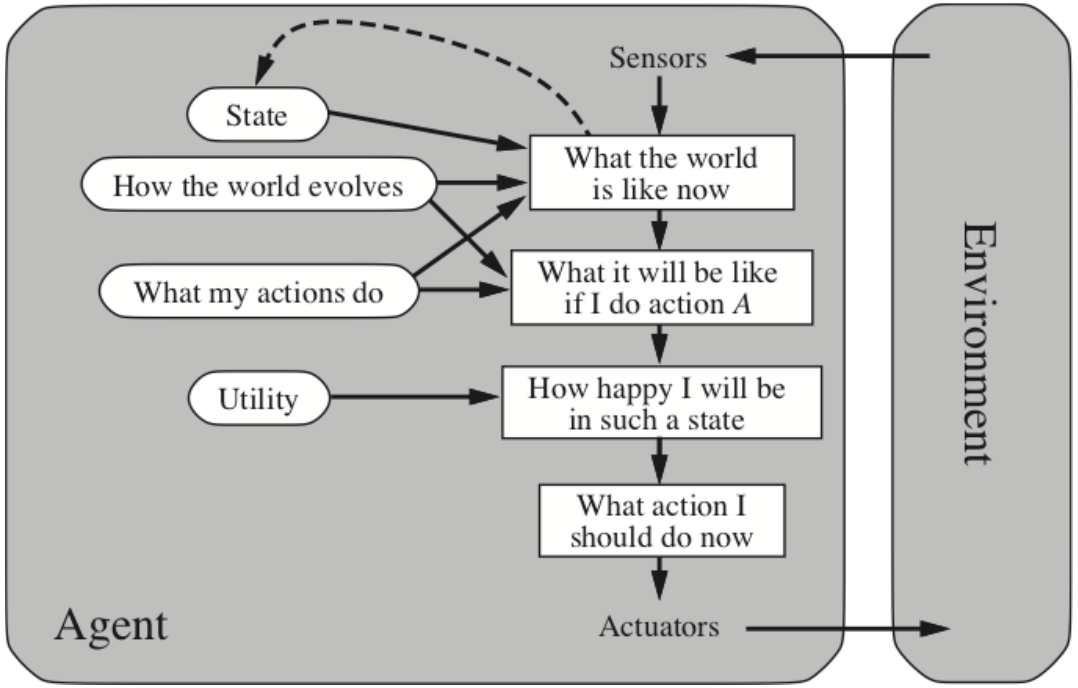


Fig 2.14 from Russell et. al.

Not only is it important that you reach your goal, but *how* you reach your goal is also important.

Types of Agents

- Learning agents– Uses feedback on the problem generation part – i.e. formulation of possible actions

Problem generation becomes *closed loop*

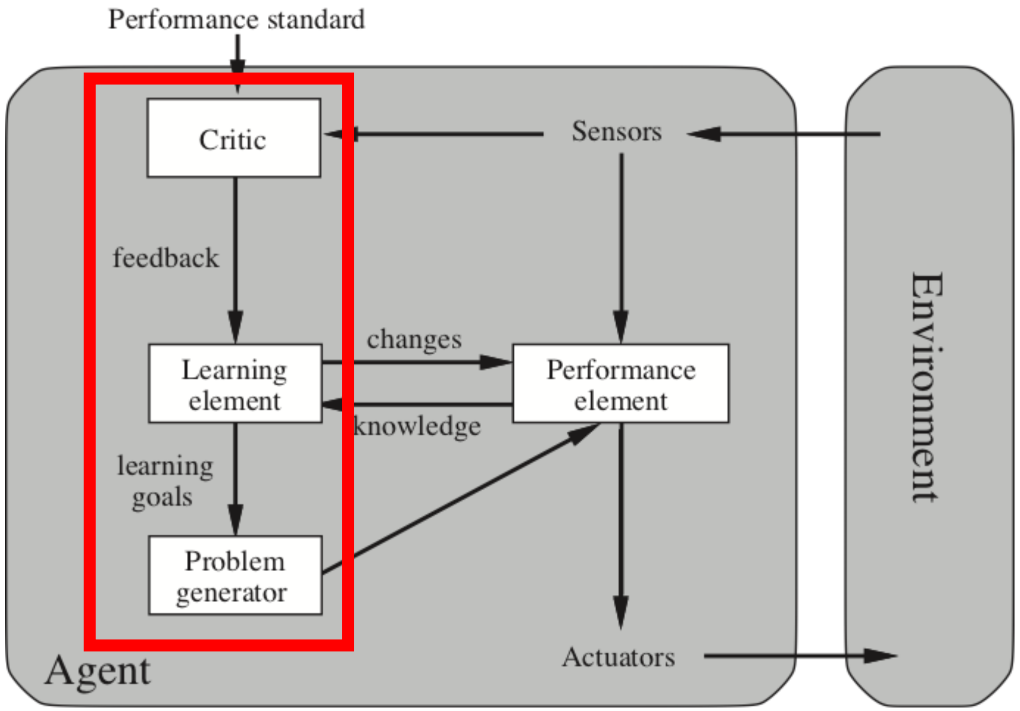


Fig 2.15 from Russell et. al.

- Uses reward (or penalty) to learn behaviors that are desirable
- How can this apply to taxi example?

Course Expectations

- Prior programming experience
  - Pset 0
    - Will test the required background for taking this course
    - Please complete this problem set *individually* in order to best assess your preparedness for the course.
- Attendance is required (tracked through in-class poll participation)

Course Policies

- Late HW (policy on website)
- Regrades
    - Must be requested within 5 days from returned assignments

Overlap with Similar Courses

- CS 181

- MIT 6.036

CS 181 (S21) VS. 182

Problem solving	Reasoning with uncertainty
Uninformed search	Bayesian networks
Informed search	Hidden Markov Models
Motion planning	Markov decision processes
Constraint satisfaction problems	**Machine learning**
Convex optimization	Reinforcement learning
Integer programming	Decision trees
Multi-agent systems	Linear classification
Game theory	Neural networks
AI game playing	Language models (Alvarez-Melis)
Wildlife protection (Tambe)	**Ethics**
Social choice	Fairness
	Value alignment
	Embedded EthiCS

Overlap

MIT 6.036 (F20) VS. 182

Problem solving	Reasoning with uncertainty
Uninformed search	Bayesian networks
Informed search	Hidden Markov Models
Motion planning	Markov decision processes
Constraint satisfaction problems	**Machine learning**
Convex optimization	Reinforcement learning
Integer programming	Decision trees
Multi-agent systems	Linear classification
Game theory	Neural networks
AI game playing	Language models (Alvarez-Melis)
Wildlife protection (Tambe)	**Ethics**
Social choice	Fairness
	Value alignment
	Embedded EthiCS

Overlap

Readings

- This lecture: Chapter 2 *Agents and Environments*
- Next lecture: Chapter 3 *Search*
- Stay ahead! Please read the materials before lectures