CS 348
Introduction to Artificial Intelligence

Mike Rubenstein
Rubenstein@northwestern.edu

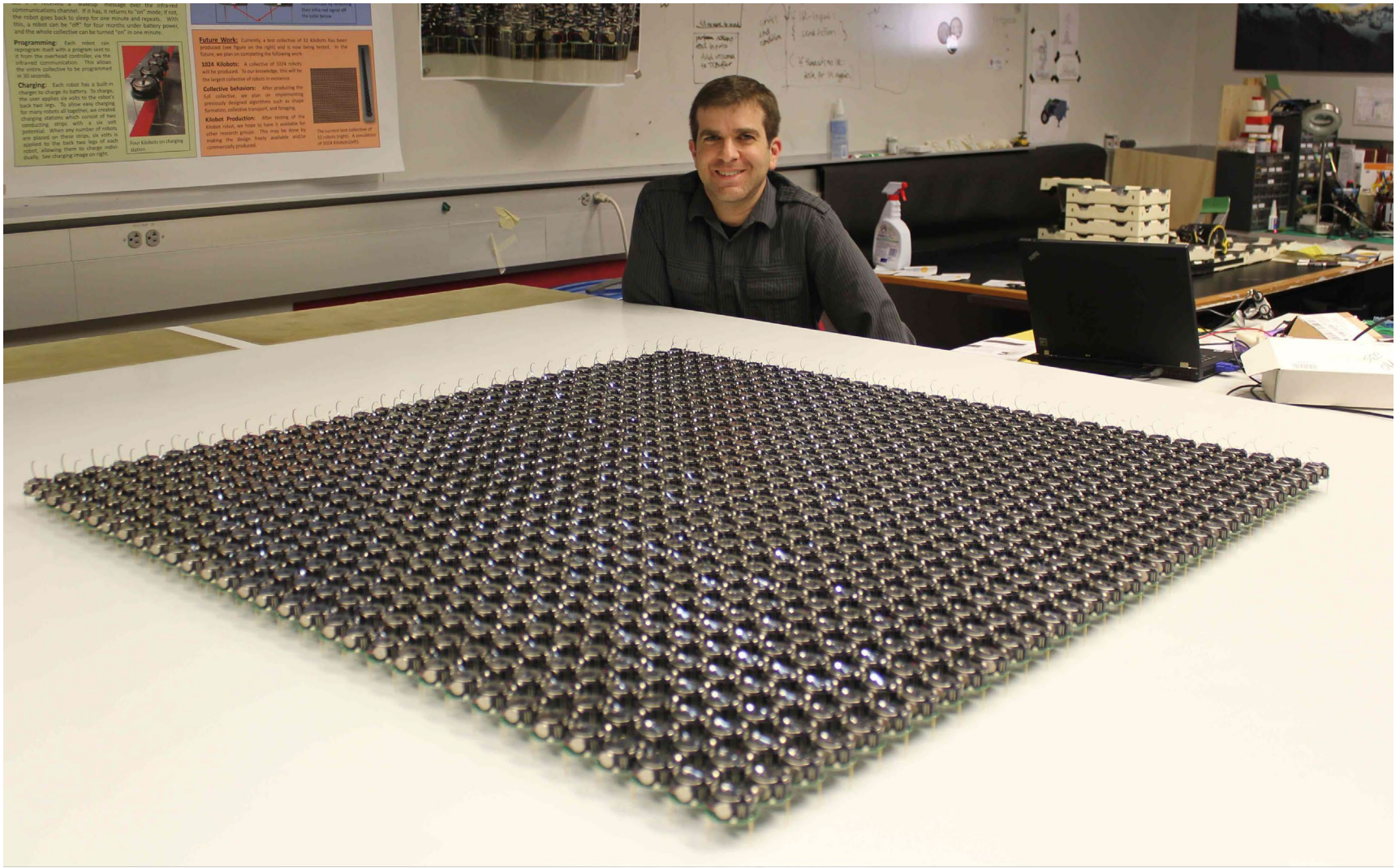
Spring 2022
Tuesday, Thursday 2:00-3:20
Room: Tech LR3

Today:

- Intro
- Agents and Environments
- Class business
  - Syllabus
  - Lab 0

- Note: all slides will be available on canvas at start of class

Briefly about me



Briefly about me

About this course:

- Goals
  - Introduce to main ideas of AI
    - AI is a HUGE discipline, we cant cover everything
    - High level survey of key techniques and algorithms from AI
      - Not in-depth in any one topic
  - Gain experience building simple examples of these main ideas

Why take this class?

- Touches on a large number of fields
  - Math, Philosophy, Neuroscience, Psychology, Economics, etc.
  - Get to play with and build fun* algorithms
  - Get to think about the future
  - Material has potentially large impact
- Why not?
  - Not easy - Upper level CS class
  - Programming required
  - You'll have to like/tolerate Python programming
    - Weekly programming assignments

Topic overview:

- Search
- Game playing
- Constraint satisfaction
- Logic and agents
- Probabilistic reasoning
- Machine learning
- Reinforcement learning
- Perceptron/Neural nets
- Genetic algorithms
- Computer Vision

Draft Schedule

day	topic	homework intro	Homework due
March 31	Intro	lab 0	
April 5	uninformed search	lab 1	lab 0
April 7	informed search	lab 2	
April 12	informed/adversarial search	lab 3	
April 14	games	lab 4	lab 1
April 19	Constraint Satisfaction	lab 5	
April 21	MDP1		Lab 2
April 26	MPD2		
April 28	Reinforcement learning	lab 6	Lab 3
May 3	Computer Vision	lab 7	
May 5	Classificaion and Perceptron		Lab 4
May 10	Classificaion and Perceptron	lab 8	
May 12	Decision trees and neural nets		Lab 5
May 17	Probablility		
May 19	---No Class---		Lab 6
May 24	bayes		
May 26	---No Class---		Lab 7
May 31	logic 1		
June 2	logic 2		Lab 8

What is Artificial Intelligence?

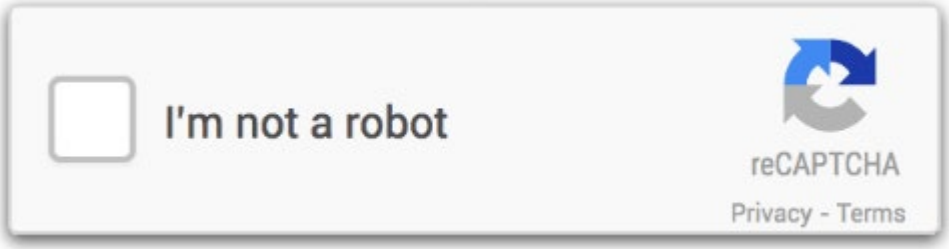
- Does this use AI?
  - Self-driving car
  - Elevator
  - Spellchecker
  - Spam filter
  - Vending machine
- In general what is AI?
- Your thoughts?

No Perfect Answer!!

Acting humanly

- Turing test
- “interrogator” poses questions to subject
- Subject passes test if interrogator cannot tell if the response is from a human or computer
- Needs to know a lot of things:
  - Language, knowledge, reasoning, learning, etc..
- Objections?

Alan Turing



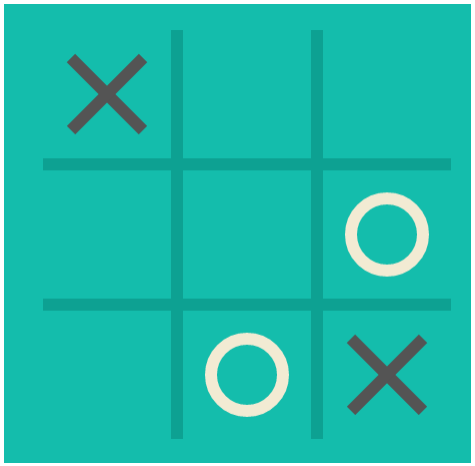
Thinking Humanly

- Find precise theory of mind
- Express that as a computer program.

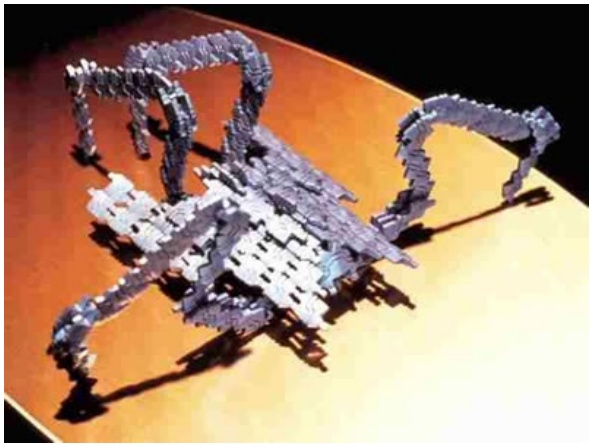
Simon and Newell

Definition for this class

- "Artificial agent that makes rational decisions"
  - Rational: "does the **right thing** given what it knows"



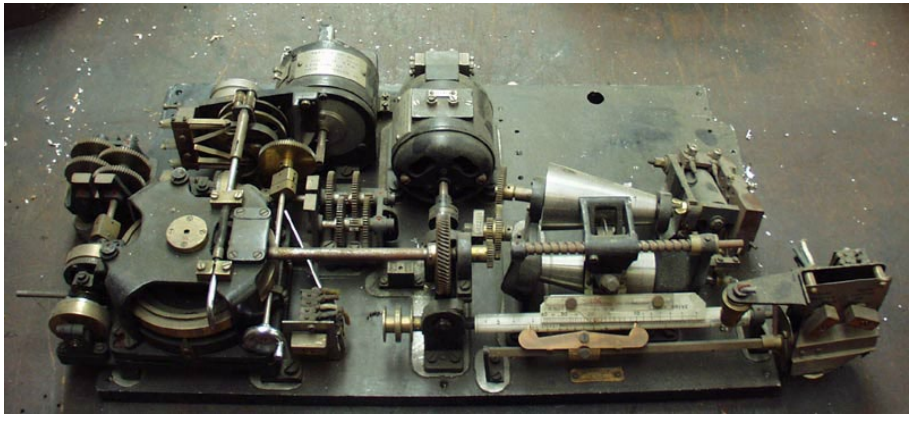
  - Note:
    - Rational is not Omniscience!



AI in fiction

History of AI

Shakey



Calculators

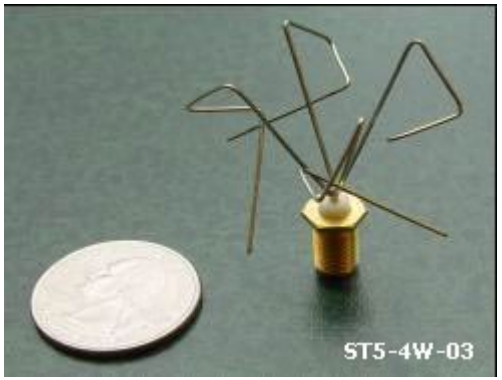
Deep blue

Modern AI

Games/education

Recommendation

Design

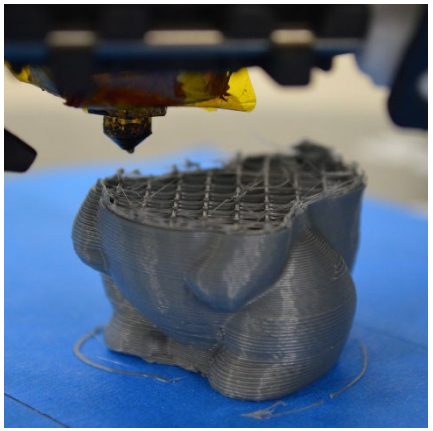


Translation

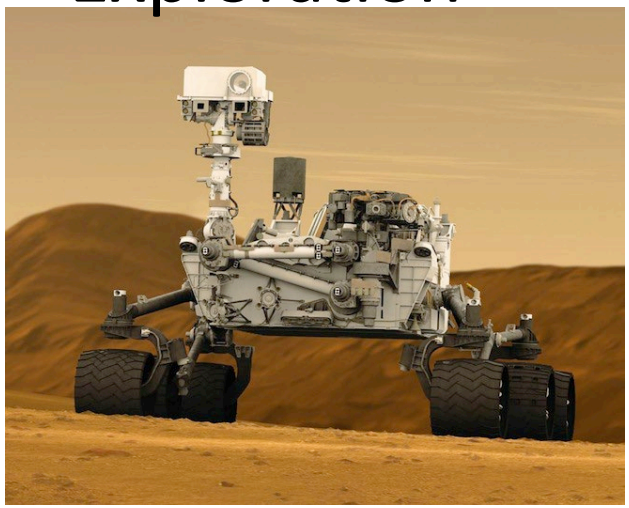
Personal assistants

Autonomous vehicles

Future of AI?

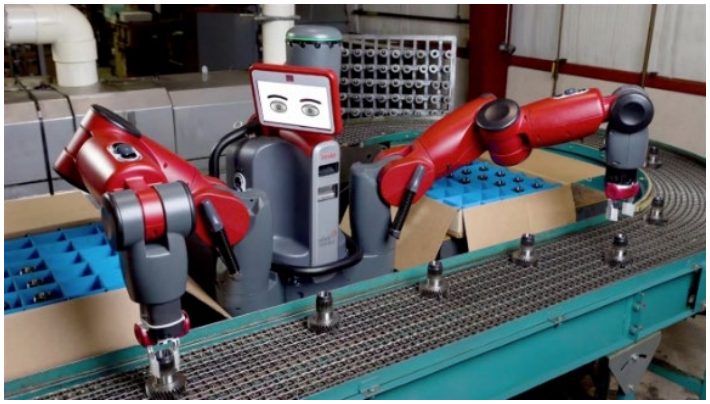


Exploration



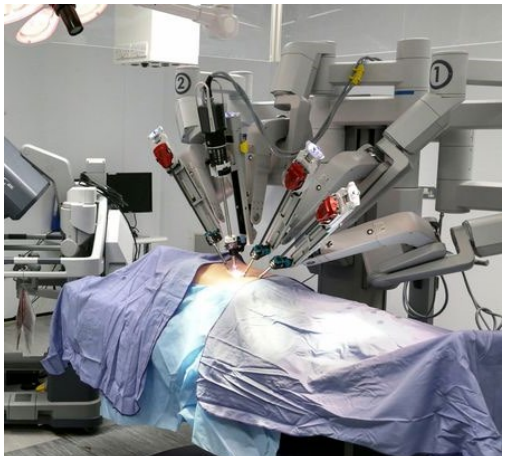
Manufacturing

Jobs



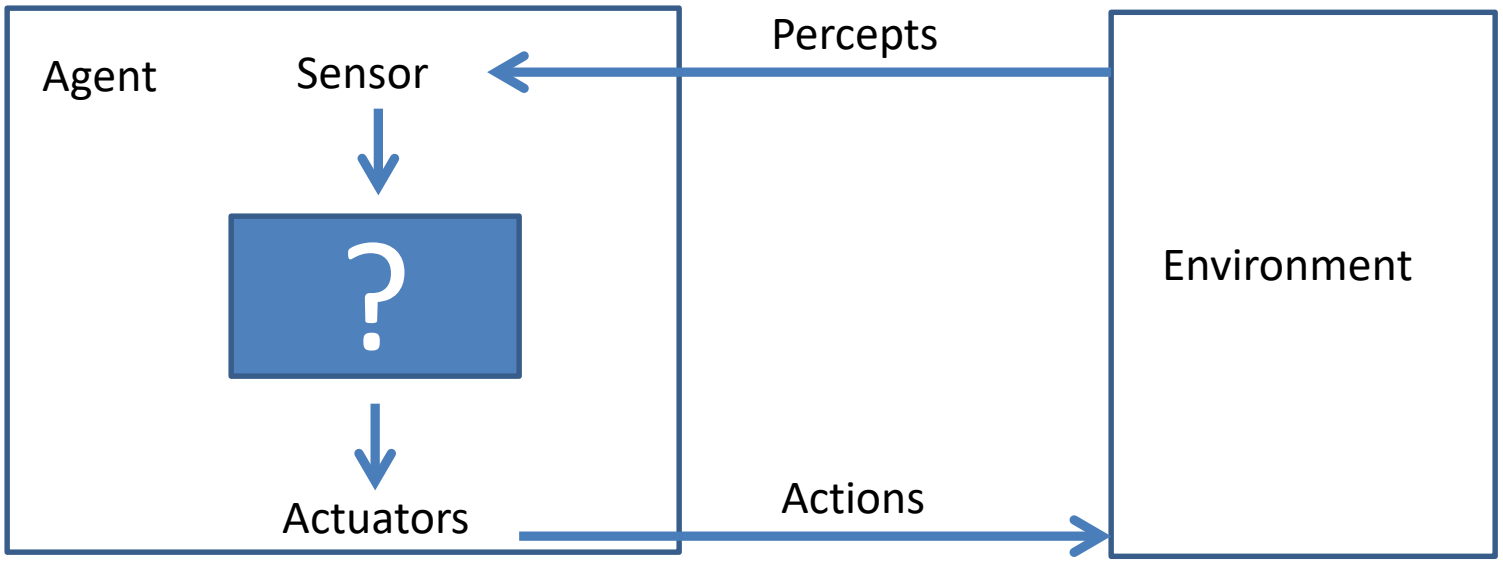
Economy

Medicine

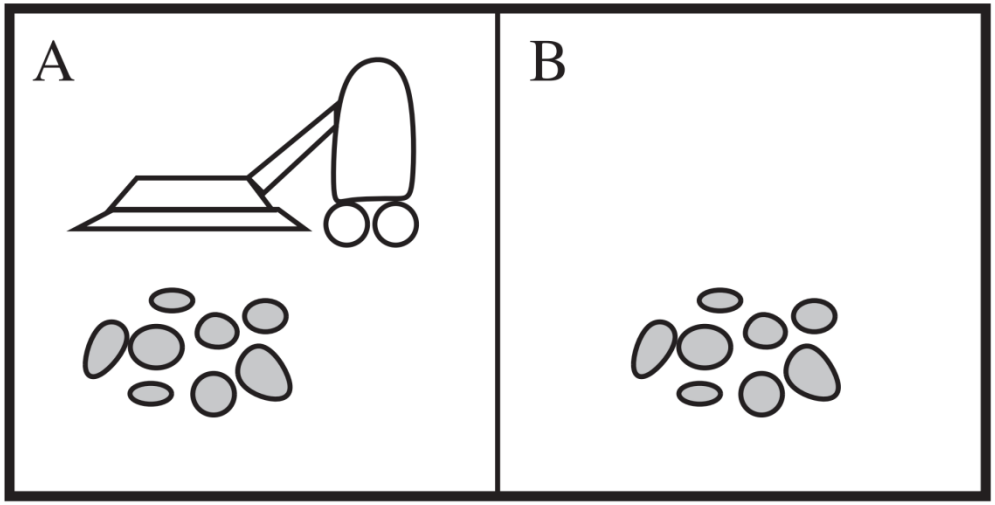


Warfare

Agents and Environment



Vacuum world example



Percepts->square, dirty

Actions->suck, move

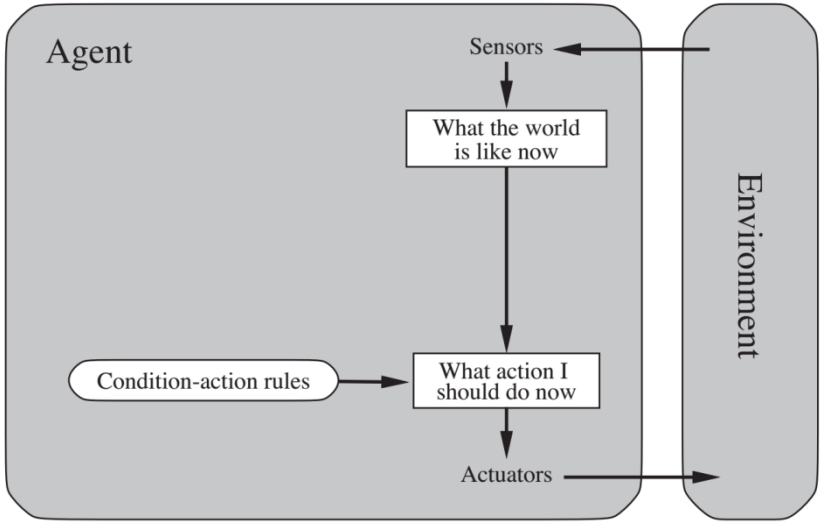
Agent program(simple reflex):

If(current_square==dirty)

suck

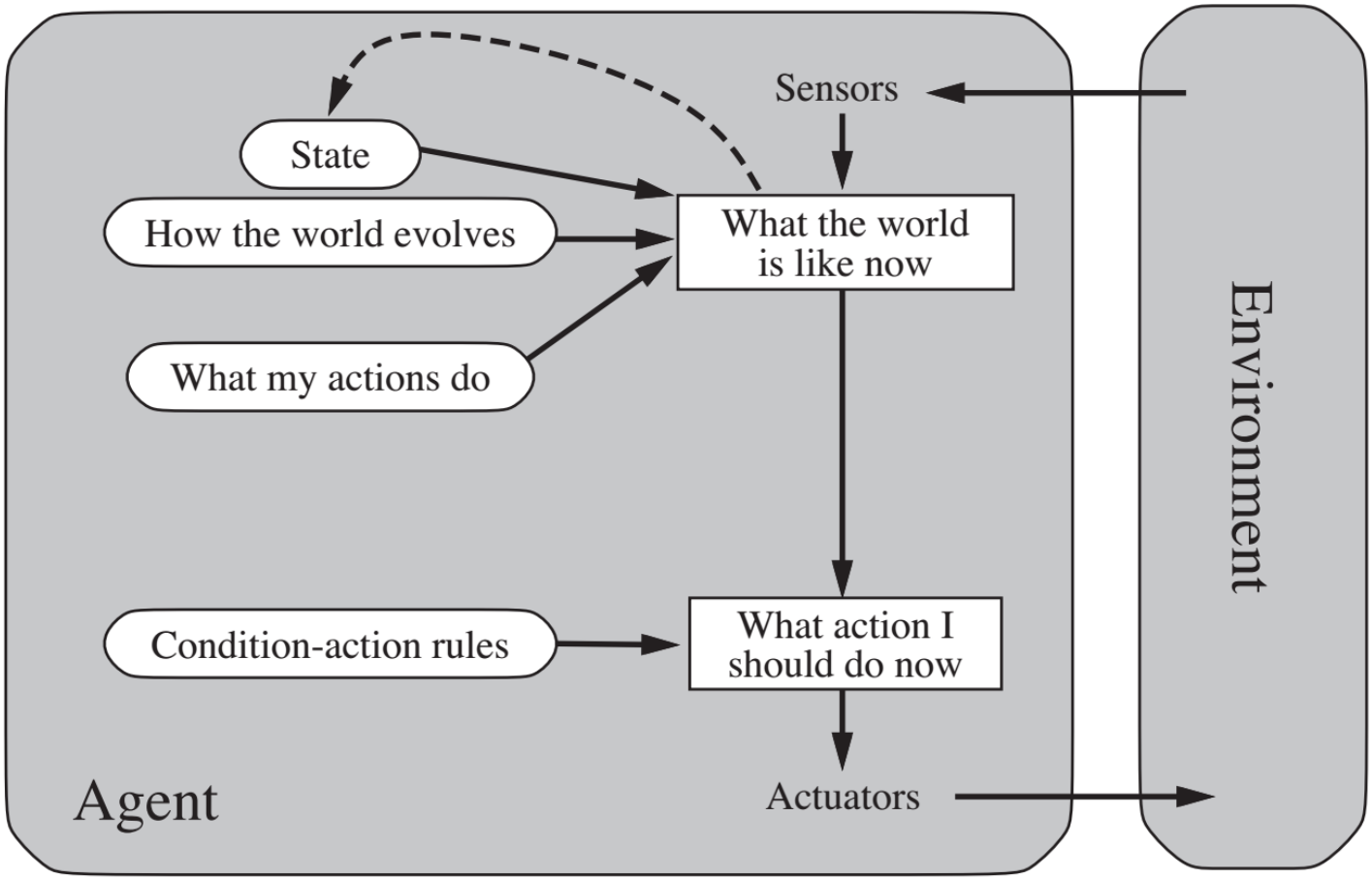
Else

move



Memory??

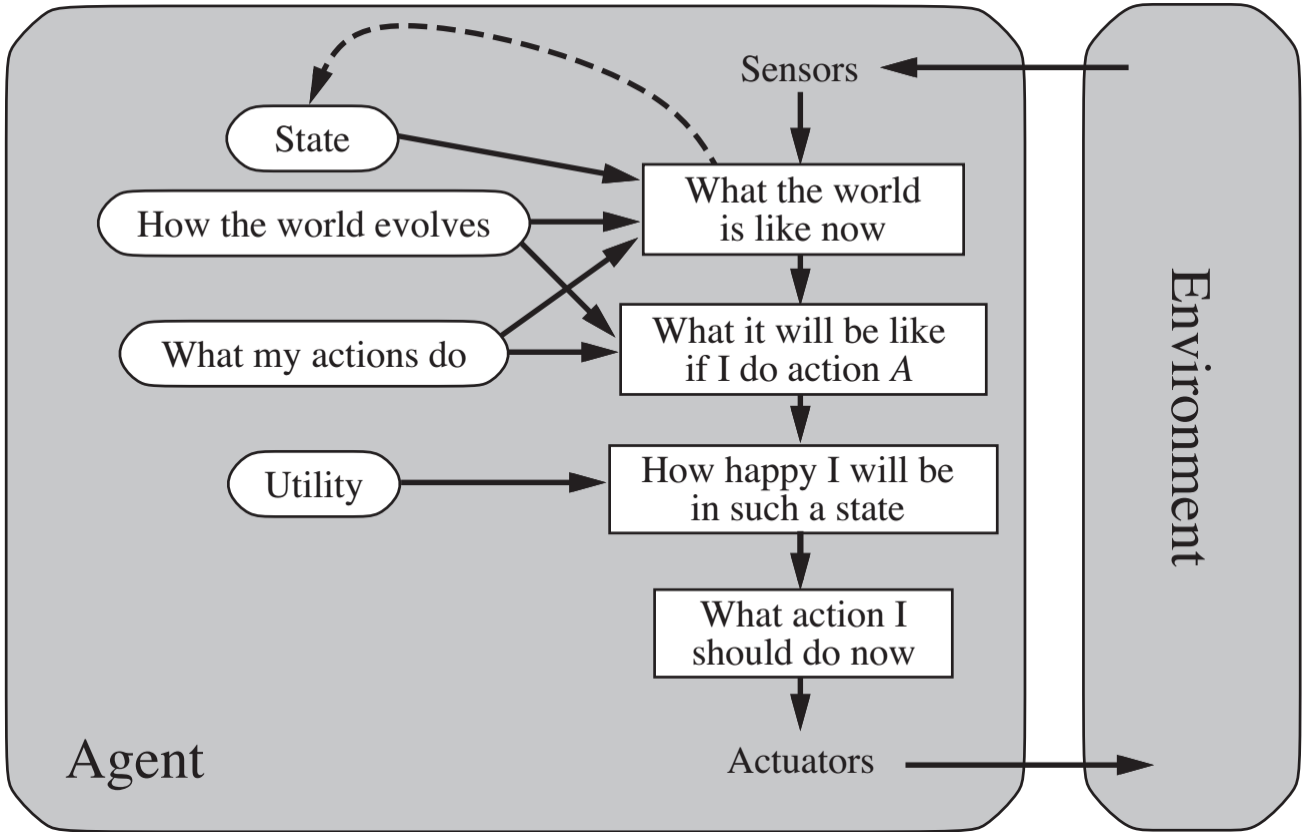
Model-based



Keep track of world it cant see right now

Utility-based

Utility: measure of “happiness” or goal proximity



Task Environments

- Performance measure
- Environment
- Actuators
- Sensors

- Self-driving car?
- Software agent (medical diagnostic system)?

Environment types

- Observable
  - Fully – agent senses all relevant aspects
    - example: chess
      - What are the sensors?
    - Memory required?
  - Partially
    - Example: poker
  - Unobservable?
    - Example?

Environment types

- Deterministic
  - Next state=current state + agent action
  - Example: chess
- Stochastic
  - Next state=current state + agent action + randomness
  - Example: Backgammon

- Static vs dynamic
  - Can outcome change while agent is deliberating?
  - Examples: poker? Autonomous car?
  - What happens as agent deliberation speed increases (i.e. faster computer)?
  - What if agent thinks fast but actuators are slow?

Environment types

- Discrete vs Continuous
  - Refers to the environment state
  - Finite number of states or infinite?
  - Examples: chess, autonomous car?

Next class

- Uninformed search

Now for some business

- Syllabus
- Lab 0
  - On canvas now
  - Due Tues April 5 at 7:00 pm (CST)

ZOOM recordings

Zoom recordings:

- I plan on recording lectures, which can be viewed until the end of quarter
- BUT not 100% guaranteed, due to technical problems.
- I will not monitor zoom for questions.
- Recordings will be available on canvas->zoom

Getting help

- First place to get help is Piazza
  - You should have received an invite, if not, email me
  - No sharing of code.
  - Do not expect instant replies.
  - Helping classmates is encouraged*.
- Next place to get help is office hours
  - 59 person/hours a week, 2 TAs, 9PMs, (still < 18min/student)
    - TAs: Manuel Valentin, Samuel Hill
    - PMs: Aura Ordonez, Garphy Tam, Sydney Smith, Tergel Myanganbayar, Ben Blakeway-Webb, Jamie Lee, Brian Gleason, Julia Greenberg
  - Times and locations vary, listed on canvas and day 1 slides
  - Don’t wait till last minute to start labs
  - Note on help: We can’t help you on coding problems, you need to know how to write and debug code.

Draft office hours

Time	Monday Mudd 3532	Tuesday Tech EG 20	Wednesday Tech EG 20	Thursday Tech EG 20	Friday Mudd 3532
9:30 - 10:00		1			
10:00 - 10:30		1		1	
10:30 - 11:00				2	
11:00 - 11:30	1	1	2	2	1
11:30 - 12:00	2	2	3	3	1
12:00 - 12:30	1	1	3	3	1
12:30 - 1:00	1	1	3	3	1
1:00 - 1:30	1	1	4	4	1
1:30 - 2:00		1	2	3	1
2:00 - 2:30			3		1
2:30 - 3:00			3		
3:00 - 3:30	1		2		
3:30 - 4:00	1		2	3	
4:00 - 4:30	1	1	4	4	
4:30 - 5:00	1	1	4	4	
5:00 - 5:30		1	4	4	
5:30 - 6:00		1	4	4	
6:00 - 6:30				4	

Grades

- Labs 100% of final grade
  - lab 0 is worth 2.5%, remainder is equally distributed across labs 1-8
- Up to 5% extra credit for participation
  - In class (participate, ask questions, etc..)
  - Piazza participation
- Grade Scale:

  A>= 91%>A->=90>B+>=89>B>=81>B->=80>C+>=79>C>=71>C->=70>D>=60>F

- No Curve

Labs

- Labs 100% of grade
- python3 programming assignments (no python2)
- 9 labs
- Done individually
- Approx. 4 hours each ~100 lines of code
- Submit on canvas
- Strict submission deadline.
- The last project submitted will be the one graded.
- No additional modules can be used (e.g. you cant import numpy)

Lab topics

0. Sorter (python test)
1. Maze solver (uninformed search)
2. Smart maze solver (informed search)
3. N-queens game (greedy search)
4. Tic-tac-toe (adversarial search)
5. Sudoku (constraint satisfaction)
6. Delivery planner (MDP planning)
7. Circle/line detector (computer vision)
8. Linear classifier (perceptron)

Labs

- All grading for labs will be done automatically, there will be no subjective component to the labs grades. This means myself or the TA will not change grades.
- Labs with compile or runtime errors will receive a 0, so make sure the code you submit runs correctly on assigned Northwestern computers. Labs with runtime errors will receive no points for that portion of the lab, so make sure you use memory safely, check for out of bounds inputs, etc..
  - Common problem to avoid: code works on your computer, but not on the assigned Northwestern computers
- Your responsibility that you submit the right files, after submission, download the submitted file and run it and check.
  - Common problem to avoid: accidentally submitted lab 1 instead of lab 2

Lab submission timeline

- T=0 hours: homework due (STRICT deadline)
- T=+24 hours (1 day): initial grades released
- T=+120 hours (5 days): First resubmission due (STRICT deadline)
- T=+143 hours: -One-time-per-quarter no-questions-asked penalty-free late submission due (STRICT deadline). Will be graded based on initial grading cases
- T=+144 hours (6 days): First resubmission grades released, grading cases released
- T=+240 hours (10 days): Second resubmission due (STRICT deadline)

Lab grading

- Initial submission:
  - Init_submit_score=autograder_score
- First resubmission:
  - first resubmission due 5 days after due date, STRICT deadline
  - first resubmission has a 20% penalty
  - Resubmit_1_score=resubmision_autograder_score*0.8
- Second resubmission:
  - Second resubmission due 10 days after due date, STRICT deadline
  - Final resubmission has a 40% penalty
  - Resubmit_2_score=resubmision_autograder_score*0.6
- Final score= Max(Init_submit_score,Resubmit_1_score,Resubmit_2_score)

Academic honesty and plagiarism

- Plagiarism is unacceptable.
  - Code you submit must be your own. No copying, adapting, or submitting code you did not create is allowed. Working together and presenting variants of the same file is not acceptable. Here are some specific guidelines to make sure you don't cross the line:
    - Do not exchange programs or program fragments in any form on paper, via e-mail, photos, or by other means.
    - Do not copy solutions from any source, including the web or previous quarters' students.
    - Do not discuss code with other students at the level of detail that will lead to identical programs or program fragments.
  - Ask me if you are uncertain.

- **All violations of the academic honesty will be immediately referred to the dean's office.**

- **Plagiarism detection will be applied to all code. It is very good and will catch you!! (20% of a class referred to dean!!)**

Class recordings

- Unauthorized student recording of classroom or other academic activities (including advising sessions or office hours) is prohibited. Unauthorized recording is unethical and may also be a violation of University policy and state law. Students requesting the use of assistive technology as an accommodation should contact AccessibleNU. Unauthorized use of classroom recordings — including distributing or posting them — is also prohibited. Under the University's Copyright Policy, faculty own the copyright to instructional materials — including those resources created specifically for the purposes of instruction, such as syllabi, lectures and lecture notes, and presentations. Students cannot copy, reproduce, display or distribute these materials. Students who engage in unauthorized recording, unauthorized use of a recording or unauthorized distribution of instructional materials will be referred to the appropriate University office for follow-up.

Book

- No book is required, however majority of the class will follow the Russel-Norvig 3rd edition of “Artificial Intelligence: A Modern Approach”

Labs framework

Select a server

- Look for a Wilkinson machine from one of these two lists with a number equal to the last digit of your student ID (it doesn't matter which list):

0 - alfred.eecs.northwestern.edu
1 - bane.eecs.northwestern.edu
2 - batgirl.eecs.northwestern.edu
3 - batman.eecs.northwestern.edu
4 - clayface.eecs.northwestern.edu
5 - cobblepott.eecs.northwestern.edu
6 - freeze.eecs.northwestern.edu
7 - gordon.eecs.northwestern.edu
8 - gotham.eecs.northwestern.edu
9 - harley.eecs.northwestern.edu

0 - huntress.eecs.northwestern.edu
1 - hush.eecs.northwestern.edu
2 - joker.eecs.northwestern.edu
3 - killercroc.eecs.northwestern.edu
4 - madhatter.eecs.northwestern.edu
5 - nightwing.eecs.northwestern.edu
6 - poisonivy.eecs.northwestern.edu
7 - ras.eecs.northwestern.edu
8 - riddler.eecs.northwestern.edu
9 - robin.eecs.northwestern.edu

- Or
  - murphy.wot.eecs.northwestern.edu
  - hanlon.wot.eecs.northwestern.edu
  - finagle.wot.eecs.northwestern.edu
  - moore.wot.eecs.northwestern.edu

NOTE: you need to be on the Northwestern network or VPN into it in order to access these machine

Verify credentials

- Use the EECS Self Service portal to validate your credentials:

  https://selfserv.eecs.northwestern.edu/
- If you have any problems contact help@eecs.northwestern.edu

Get a ssh client

- We recommend MobaXterm
  http://mobaxterm.mobatek.net/download.html
- Linux and Mac have native ssh clients (type “ssh –help” on a terminal window to get more info)

Lab 0 sort

-Designed to familiarize everybody with environment and submission process.

-Designed to make sure you are comfortable with using python -- If you struggle with Lab0, you don't have the python coding experience needed for 348. If this is the case I would recommend dropping the class.

-Modify student_code.py so when the function order(data) is called, it sorts the data in that array (smallest numbers at lowest index, highest at highest index)

e.g. changes this [-1, 3, 2, 8, 9, 11, 100, 4] to this: [-1, 2, 3, 4, 8, 9, 11, 100]

-data array can have any size >= 0

-data array will only have integers, ranging from -1000 to +1000 (inclusive)

-cannot use the python built-in list function sort() or sorted(). Code that uses this will receive a 0.

-cannot include any additional modules. Code that uses this will receive a 0.

Lab 0 details

- Download files on canvas “lab 0” folder
- Create a directory in one of the NU computers and put these files there.
- Modify student_code.py
- Execute and test by running: python3 main.py
  - This tests your code against some test cases, but passing all test cases does not guarantee you will pass all grading cases. Try your own test cases too! Think about corner cases!

Lab 0

- This is the full description of Lab 0!
- Lab 0 due 7:00pm April 5

- For lab 0, TAs and PMs will help you get the environment running, but will not help with the logic of the lab. This is supposed to test your Python programming proficiency.

- -Show demo:

Questions?