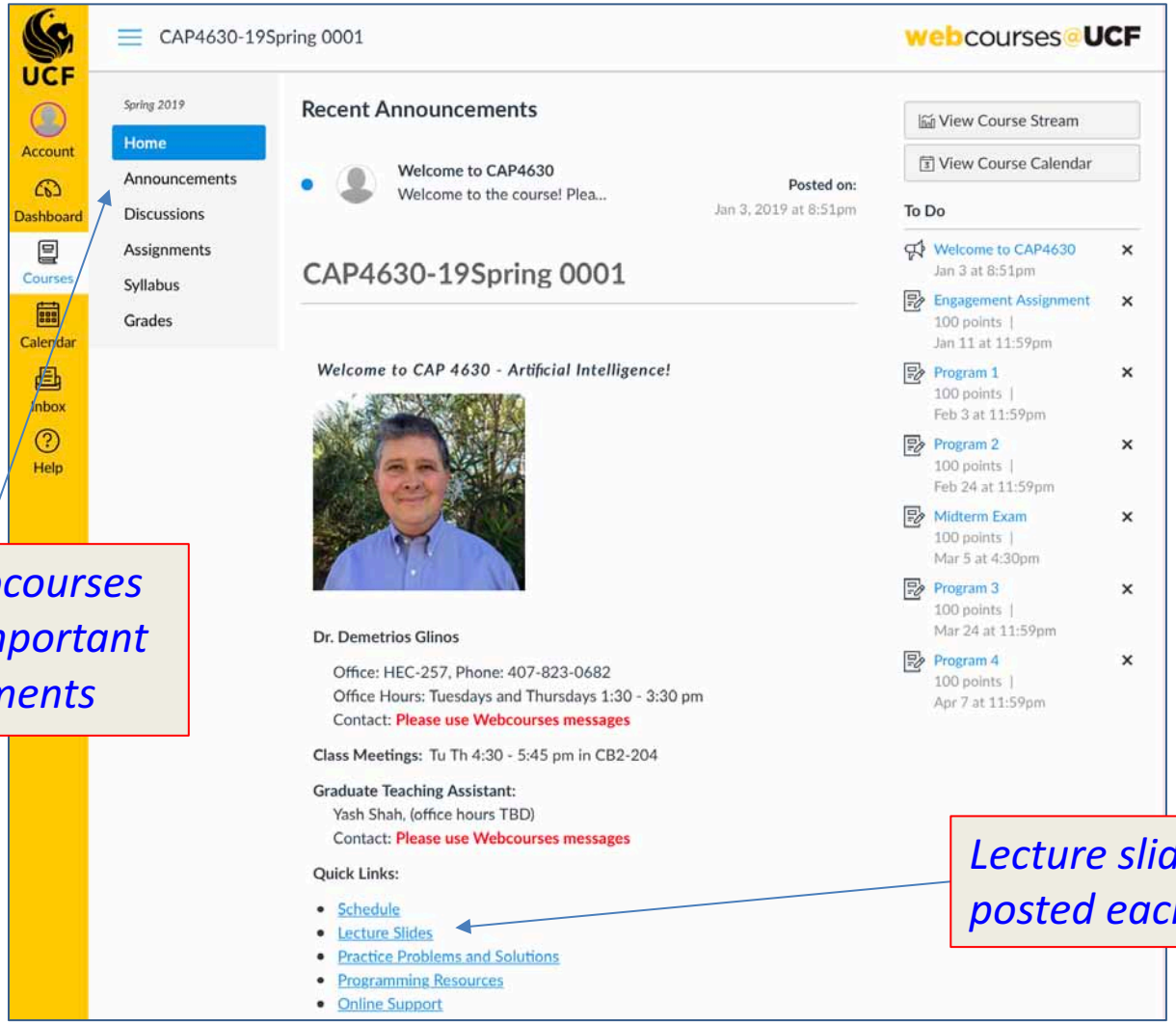
CAP4630 – Artificial Intelligence
Course Introduction

Dr. Demetrios Glinos
University of Central Florida

Spring 2019

Course Home Page



Check Webcourses daily for important announcements

Lecture slides will be posted each week

Course Schedule

- For the complete schedule, use the "Quick Link" from course home page

Weeks/Dates	Topics Covered	Readings/Assignments
Week 1 (1/7 - 1/11)	Course Introduction Uninformed Search	RN Ch. 1 and 3.1 - 3.4 Engagement assignment due Friday (or when join class)
Week 2 (1/14 - 1/18)	Heuristic Search Program 1 Preview Local Search and Genetic Algorithms	RN 3.5 - 3.6 RN 4.1
Week 3 (1/21 - 1/25)	Constraint Satisfaction Problems Adversarial Search	RN 6.1 - 6.5 RN 5.1 - 5.3
Week 4 (1/28 - 2/1)	Uncertainty and Utilities Program 2 Preview Markov Decision Processes	RN 16.1 - 16.3 RN 17.1 - 17.3 Program 1 due Sunday 2/3

Course Information

- Prerequisite
  - COP 3503C (CS II) and COT3960 (Foundation Exam)
  - Data structures
  - Programming in Java
  - Compiling and running from a terminal (command) window
- Work and Grading (see syllabus for details)
  - Engagement assignment (0 points)
  - 4 Programming assignments (drop lowest, 45 pts):
  - Mid-term exam (25 pts)
  - Final Exam (30 pts, not cumulative)
  - Bonus Assignment (5 pts, Team Pac-Man Tournament)
  - Fixed scale: ABCDF, with +/- (see next slide)
  - Exam/Program grades will not be curved
  - Final grade will be rounded to the nearest whole number
  - UCF's Golden Rule applies

Grading

Assessment	Point Value
Engagement Assignment	0
Program Assignments	45
Mid-Term Exam	25
Final Exam	30
Bonus (optional)	5
Total	105

The lowest program grade will be dropped

Letter Grade	Points
A	90 and above
A -	88 - 89
B+	86 - 87
B	80 - 85
B -	78 - 79
C +	76 - 77
C	70 - 75
C -	68 - 69
D +	66 - 67
D	60 - 65
D -	58 - 59
F	57 and below

- Note the relaxed grading scale
- Grading scale also includes rounding of final grade
- No other grades will be curved or rounded

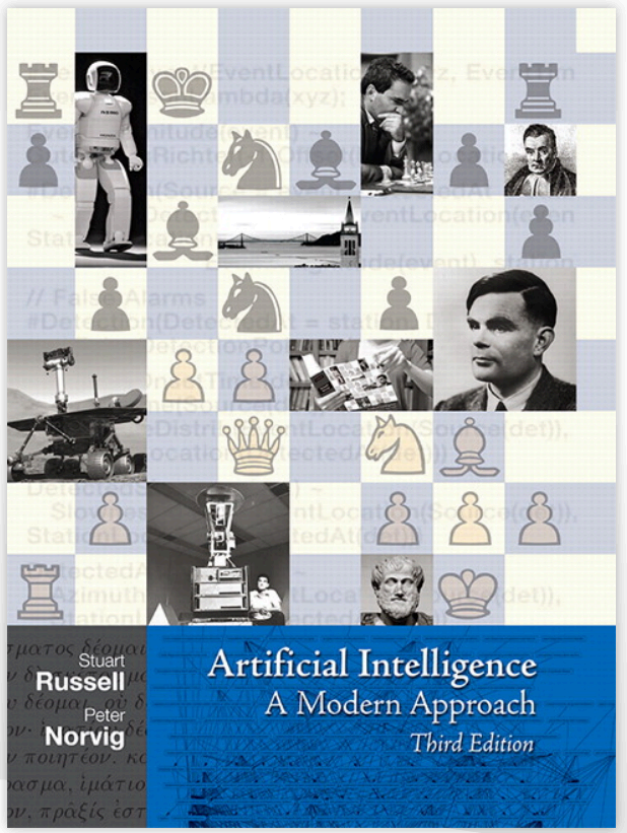
Course Rules

- Program assignments
  - Required programming languages: Java (3 pgms + bonus), Prolog (1 pgm)
  - All source files must contain header in prescribed format indicating author(s)
  - Program assignments may be done individually or in teams of 2 students
  - If teaming:
    - both teammates must submit the same file(s)
    - program file(s) must contain header identifying both authors
    - see syllabus for deductions for inconsistent team submissions
  - Bonus assignment can be done as a team
- Exams
  - May use calculator and one 8.5” x 11” personal note sheet; can use both sides
  - Must be in your own handwriting
  - Must be original (no photocopy)
  - Cannot contain anything computer generated or printed

Course Rules (cont'd)

- Class attendance
  - Required
- Missed exams
  - No make-up for mid-term exam except for truly exceptional circumstances
  - Final exam cannot be made up (university policy)
- Late program submissions
  - Lose 1 pt per minute past the deadline
- Office hours
  - For answering particular questions about the course
  - Be prepared to show what you have done so far

Recommended Course Textbook (1)



Artificial Intelligence: A Modern Approach | Edition: 3

Author: Stuart Russell, Peter Norvig

ISBN: 9780136042594

Publication Date: 12/14/2009

Publisher: Pearson

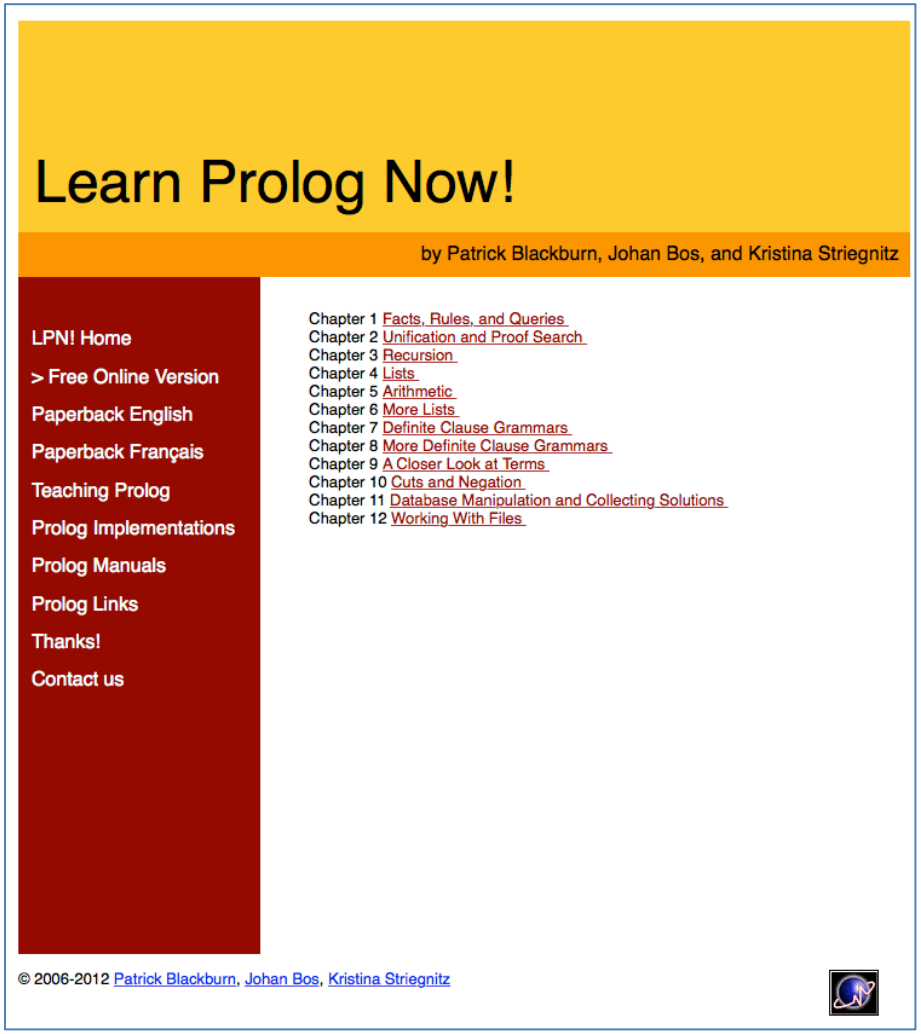
Estimated Student Price **(Savings based on Print, New)**

- Print, new: $199.00
- Print, used: $149.25 **(25% Savings)**
- Print, new rental: $135.30 **(32% Savings)**
- Print, used rental: $93.55 **(53% Savings)**
- eBook, buy: $119.00 **(40% Savings)**
- eBook, rent (for 180 days): $40.00 **(80% Savings)**

Student use of this title is Required

("RN" reading assignments in schedule)

Recommended Course Textbook (2)



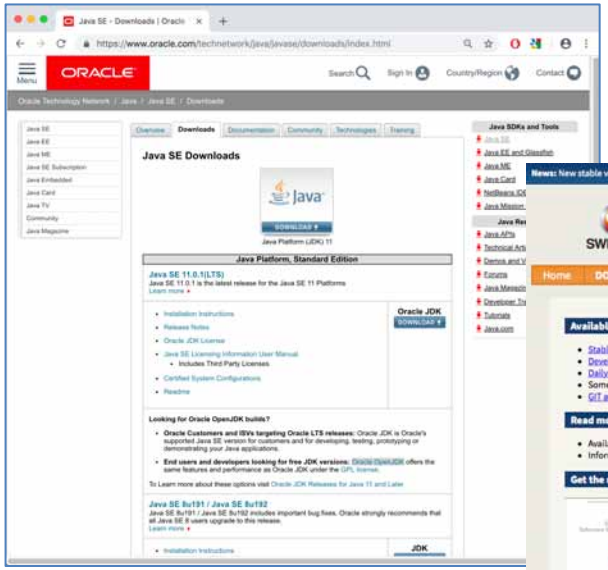
http://www.learnprolognow.org/lpnpage.php?pageid=online

- Required (free online version)
- "LPN" reading assignments in schedule

Software You Will Need

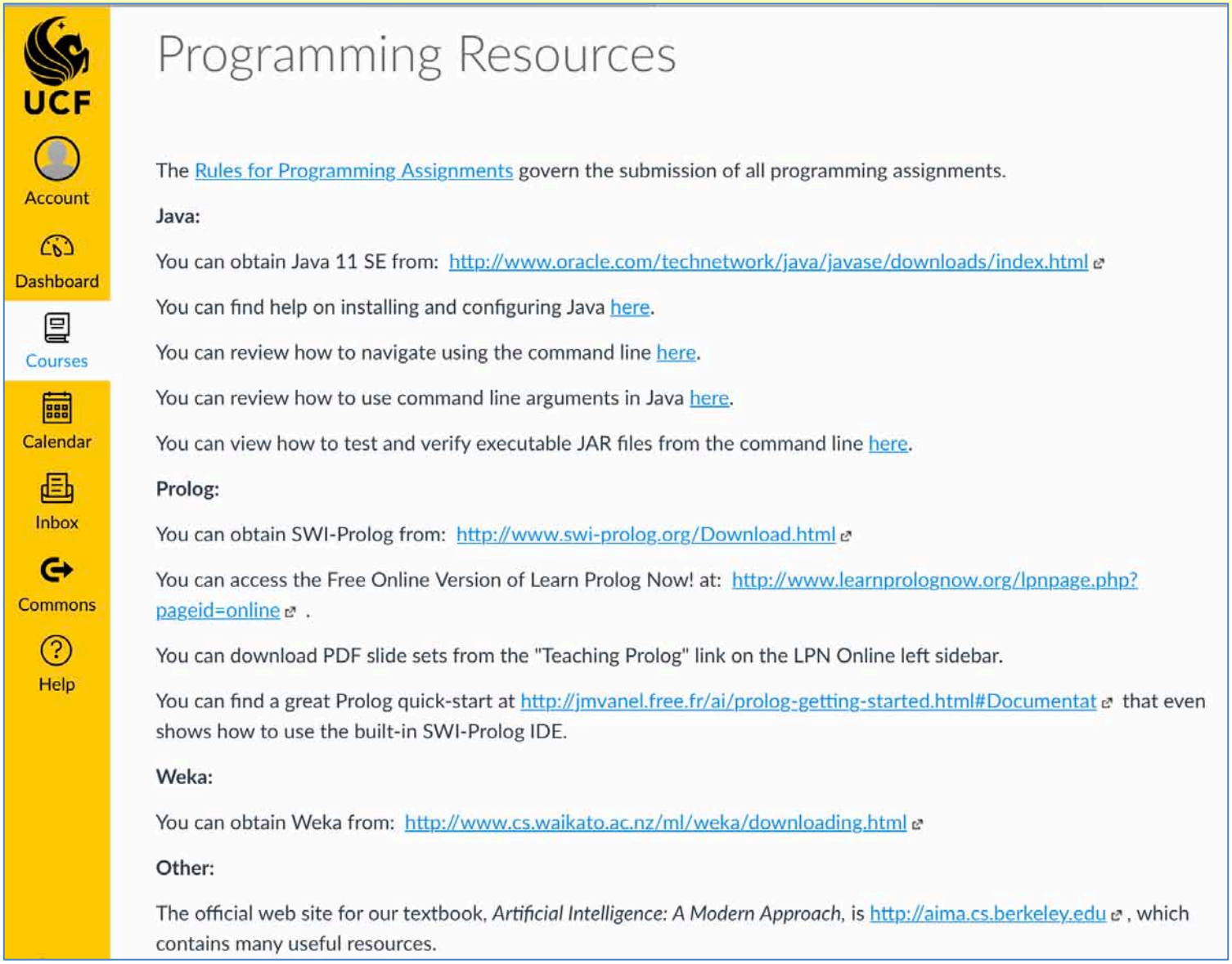
Download links are on Programming Resouces page

Java 11



SWI-Prolog

Weka



Programming Resources

The Rules for Programming Assignments govern the submission of all programming assignments.

Java:

You can obtain Java 11 SE from: http://www.oracle.com/technetwork/java/javase/downloads/index.html

You can find help on installing and configuring Java here.

You can review how to navigate using the command line here.

You can review how to use command line arguments in Java here.

You can view how to test and verify executable JAR files from the command line here.

Prolog:

You can obtain SWI-Prolog from: http://www.swi-prolog.org/Download.html

You can access the Free Online Version of Learn Prolog Now! at: http://www.learnprolognow.org/lpnpage.php?pageid=online .

You can download PDF slide sets from the "Teaching Prolog" link on the LPN Online left sidebar.

You can find a great Prolog quick-start at http://jmvanel.free.fr/ai/prolog-getting-started.html#Documentat that even shows how to use the built-in SWI-Prolog IDE.

Weka:

You can obtain Weka from: http://www.cs.waikato.ac.nz/ml/weka/downloading.html

Other:

The official web site for our textbook, *Artificial Intelligence: A Modern Approach*, is http://aima.cs.berkeley.edu , which contains many useful resources.

Outline

- What is AI ?
- What Can AI Do ?
- What this course will cover

AI in Pop Culture

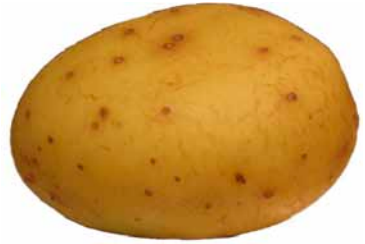
Who or What Has Intelligence?

We must be able to recognize intelligence when we see it.

So, let's consider a few examples.

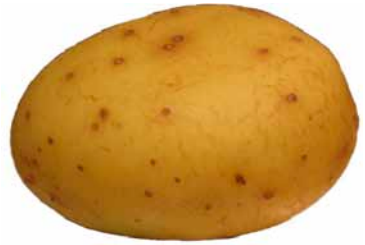
Who or What Has Intelligence?

Potato ?

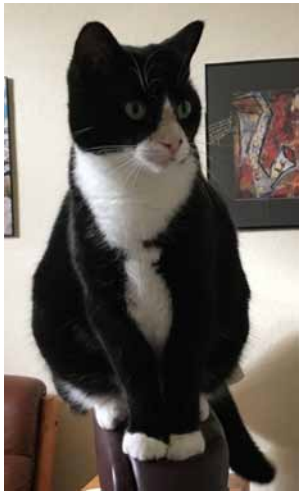


Who or What Has Intelligence?

Potato ?

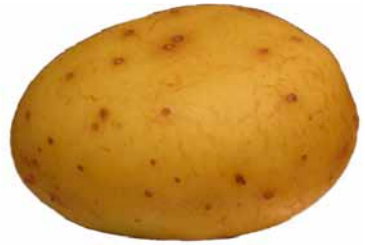


Kitten ?

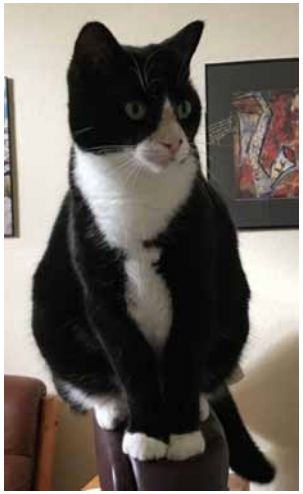


Who or What Has Intelligence?

Potato ?



Kitten ?

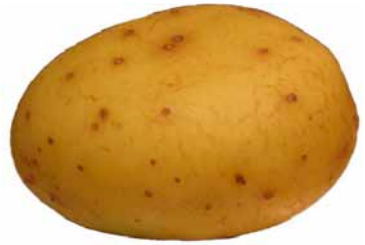


Program to play tic-tac-toe ?

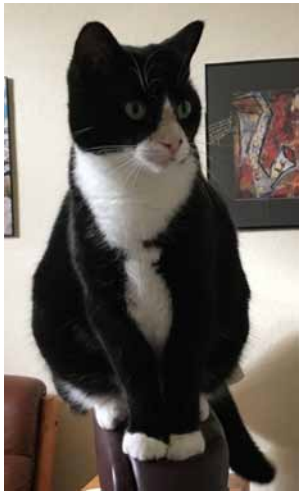
X		X
	O	X
O	X	O

Who or What Has Intelligence?

Potato ?



Kitten ?



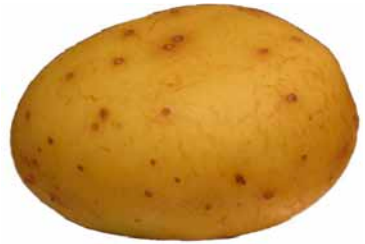
Program to play tic-tac-toe ?

X		X
	O	X
O	X	O

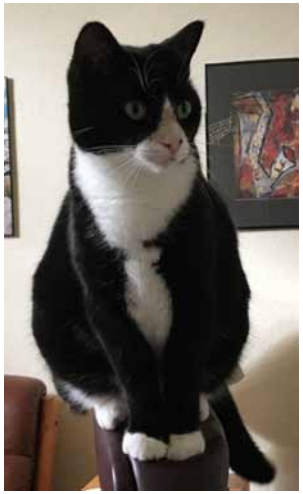
Self-driving car ?

Who or What Has Intelligence?

Potato ?



Kitten ?



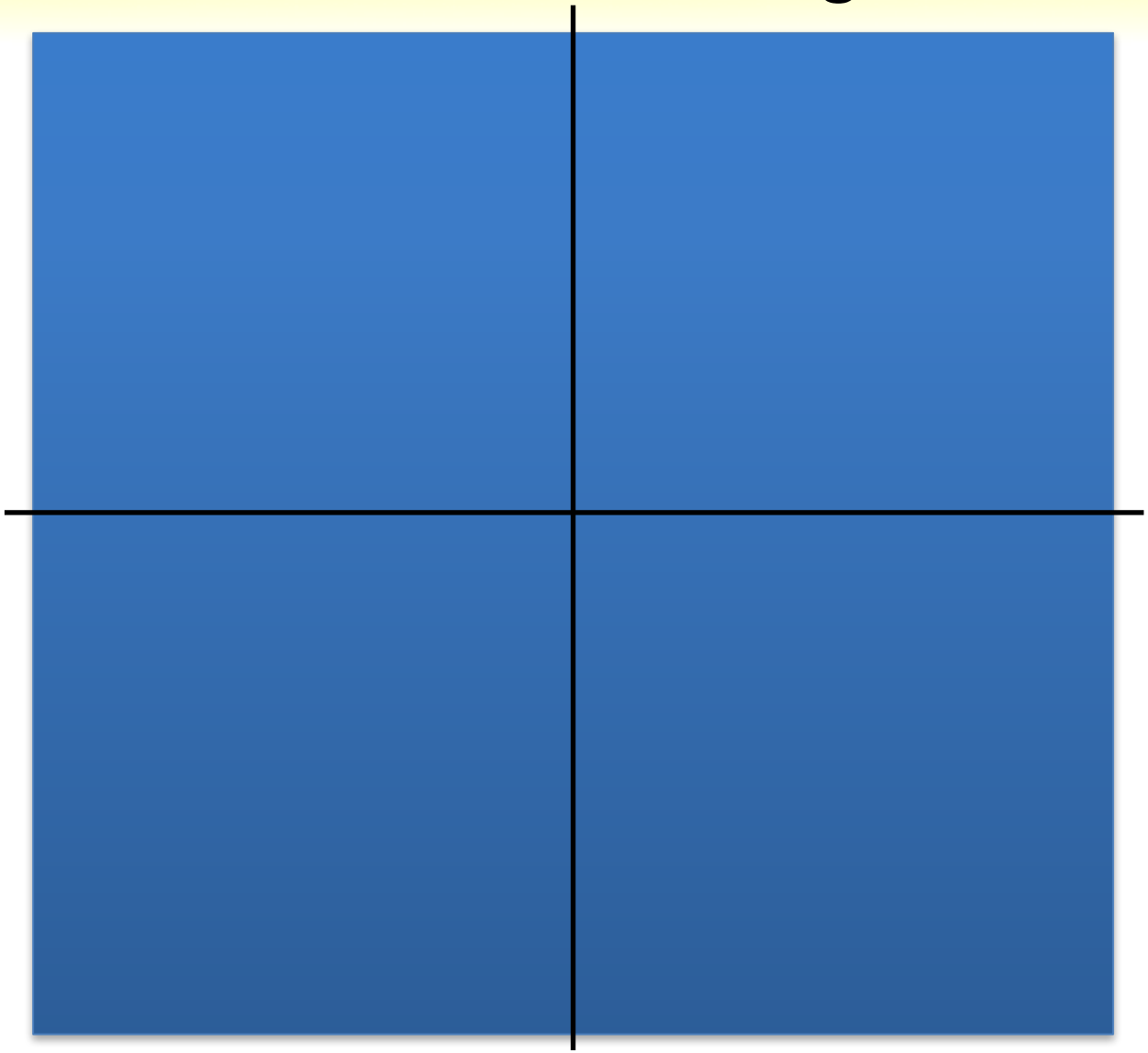
Program to play tic-tac-toe ?

X		X
	O	X
O	X	O

Self-driving car ?

Q: What do intelligent entities have in common?

What is “artificial intelligence” ?



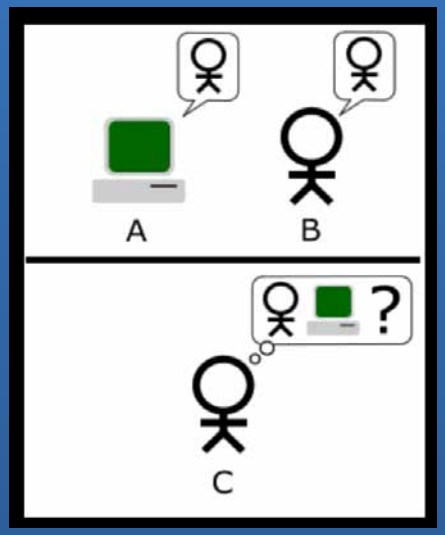
What is “artificial intelligence” ?

Think like a human

What is “artificial intelligence” ?

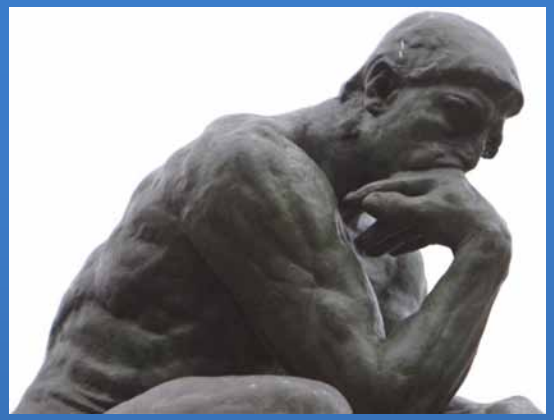
Think like a human

Act like a human



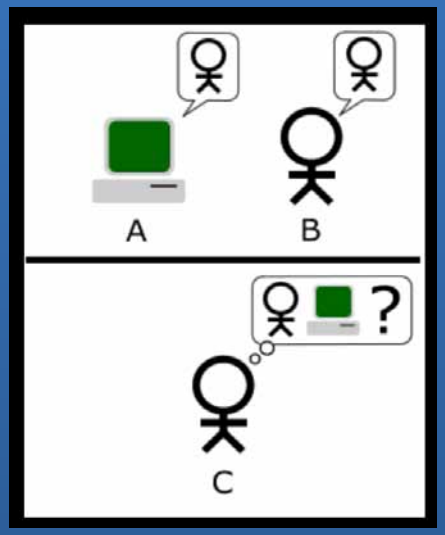
What is “artificial intelligence” ?

Think like a human



Think rationally

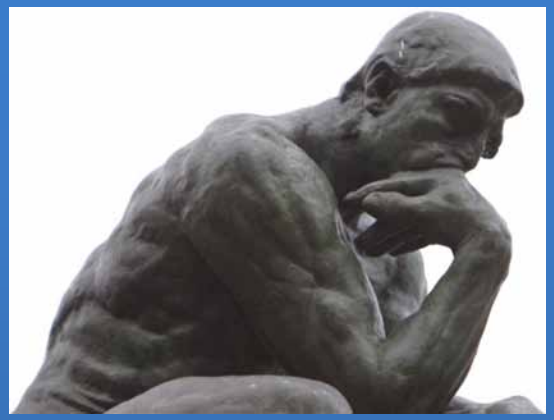
Act like a human



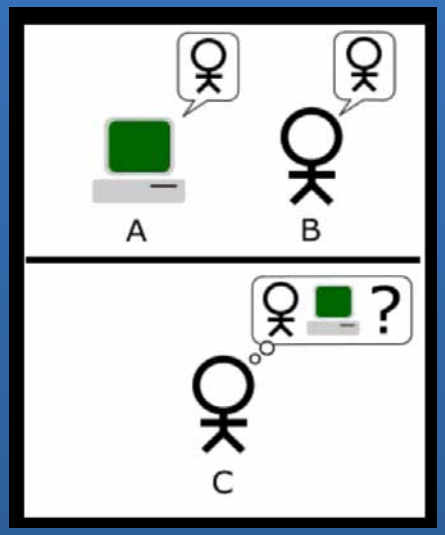
What is “artificial intelligence” ?

Think like a human

Think rationally



Act like a human



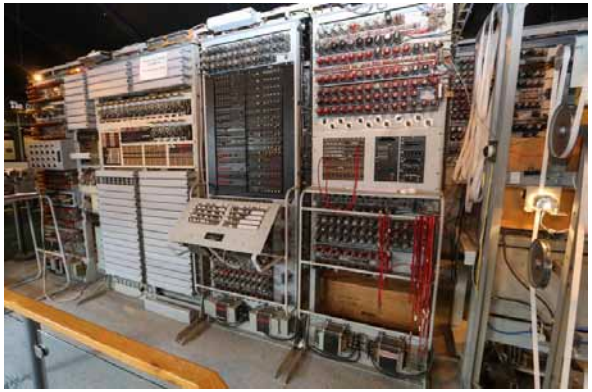
Act rationally

What *we* mean by AI

- For us, AI is the science of making machines that **act rationally**
  - Achieving ***goals*** in an ***optimal*** way
  - Goals are expressed in terms of ***utility***
  - Rationality is concerned with the ***decisions***, not the thought process
    - Chess players think in terms of "controlling the center", "passed pawns", "connected rooks", "open files", etc.
    - An ideal chess-playing strategy would be just a lookup table (impractical, but conceptually simple).
  - **Acting rationally means acting to maximize your expected utility**
  - **Our goal is to design rational agents (actors)**

Brief History of AI

- ***1940-1950: First steps***
  - 1943: McCulloch & Pitts: Artificial neuron
  - 1950: Alan Turing's paper: "Computing Machinery and Intelligence"
    - Computers based on tubes and relays, room-sized
    - World War II context: code-breaking, anti-aircraft aiming, etc.
    - Transistor invented in 1947



- ***1950-1970: Initial excitement: "Look Ma, no hands!" period***
  - 1956: Dartmouth Conference: the term "artificial intelligence" is coined
  - 1950s: Newell and Simon's Logic Theorist, and later, General Problem Solver
    - First integrated circuits in early 1960s
    - Computers were expensive mainframes

Brief History of AI

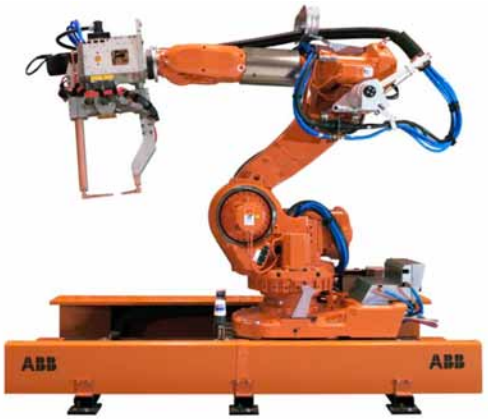
- ***1970-1990: Knowledge-based approaches***
  - 1970s: Early KBS successes: DENDRAL, MYCIN, etc.
  - 1980s: Expert systems industry booms: DEC's R1
  - 1988-1993: Expert systems industry busts: "AI Winter"
    - Pre-Internet age
    - Small hand-crafted data sets

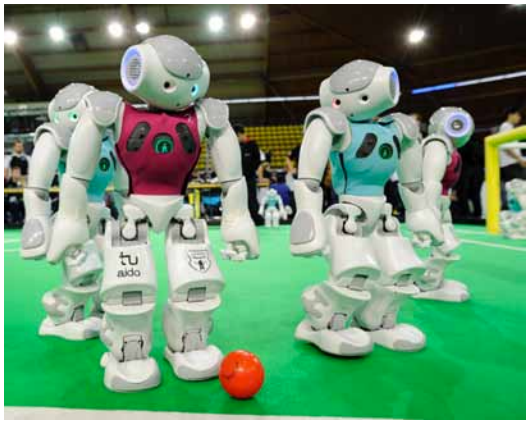
- ***1990-present: Statistical methods and uncertainty***
  - Connectionist models: neural networks
  - Large data sets: data mining
  - Focus on learning
    - Cluster computing
    - Data mining

- ***Today: Are we in an "AI Spring" ?***

What Can AI Do: Robotics

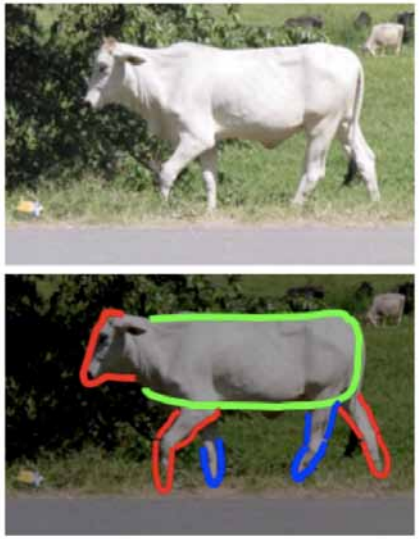
- Robotics combines
  - AI
  - Mechanical Engineering
- Physical reality introduces new uncertainties
- Technologies
  - Industrial robots
  - Hazmat
  - Google cars
  - Robocup
- Our focus: decision making
  - Planning
  - Control

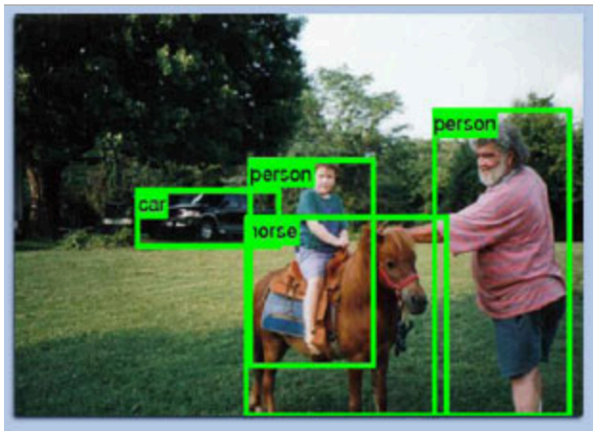


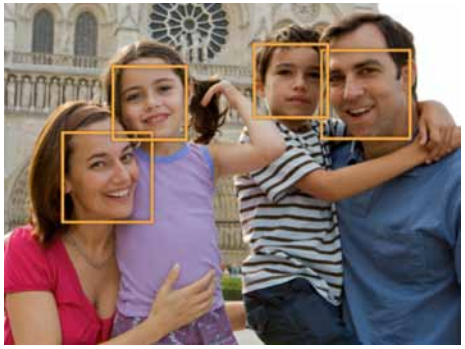


What Can AI Do: Computer Vision

- Image segmentation
- Object detection
- Face detection
- Not restricted to human capabilities

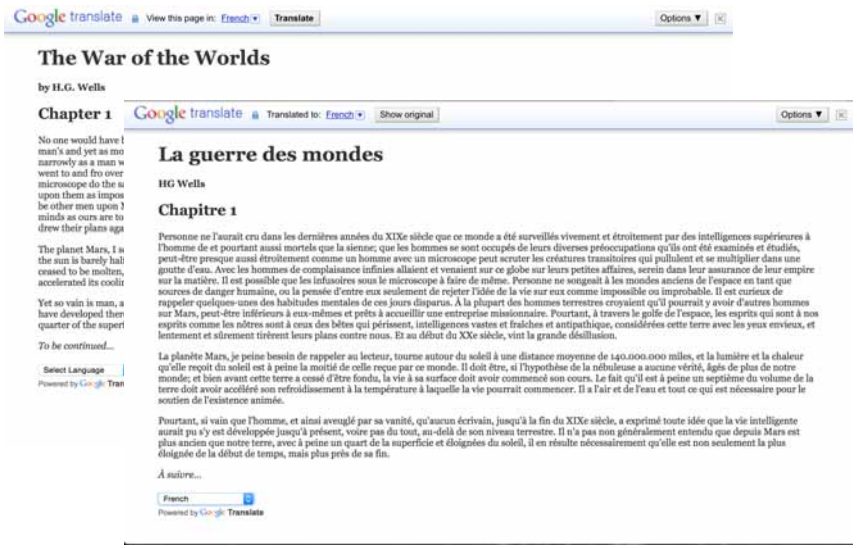






What Can AI Do: Natural Language

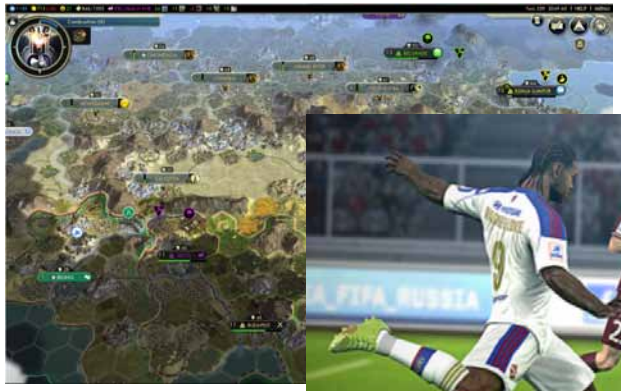
- Speech technologies
  - Automatic speech recognition (ASR)
  - Text-to-speech (TTS)
  - Dialog systems
    - Siri, etc.



- Language technologies
  - Machine translation
  - Question answering
  - Text classification
  - Named entity recognition
  - Web search engines

What Can AI Do: Game Playing

- Deep Blue v. Kasparov
  - 1996: Kasparov wins
  - 1997: Deep Blue wins
  - Tournament conditions
- Video games
  - Computer-generated opponents
  - First person shooters
  - Games of strategy



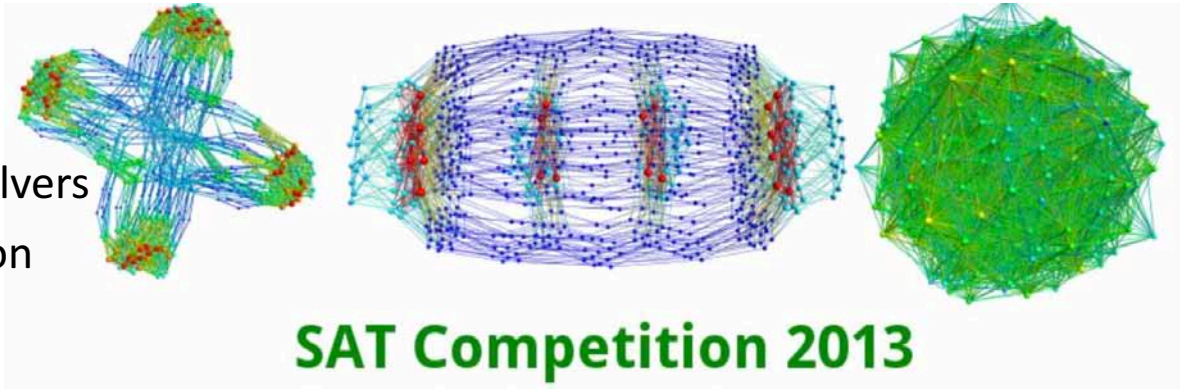
What Can AI Do: Logic and Analysis

- Theorem provers
  - Logic Theorist (1956)
  - Work continues
- NASA spacecraft fault diagnosis
- Methods:
  - Deduction systems
  - Satisfiability (SAT) solvers
  - Constraint satisfaction

Logic Theorist

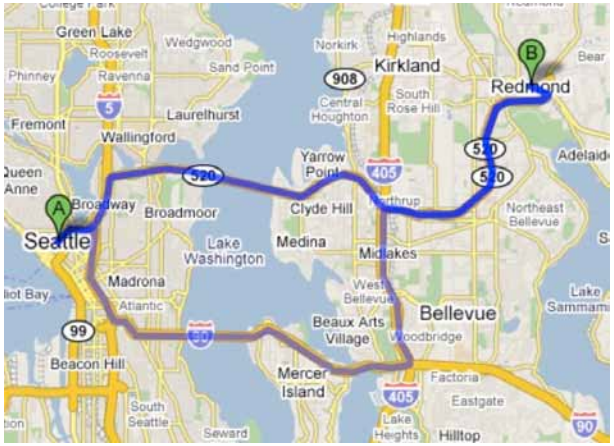
In about 12 minutes LT produced, for theorem 2.45:

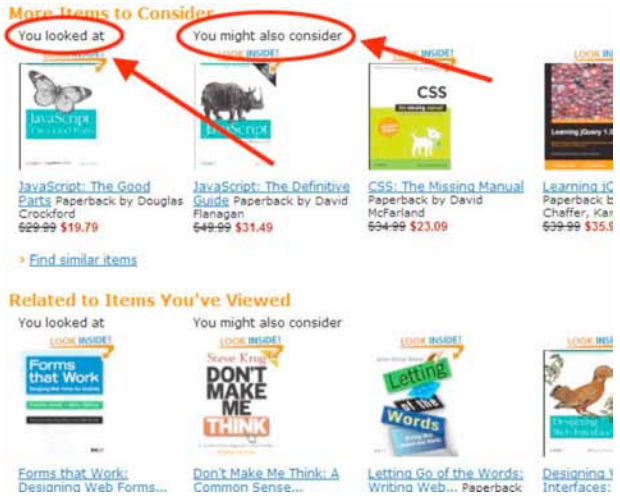
	$\neg(p \vee q) \rightarrow \neg p$	(Theorem 2.45, to be proved.)
1.	$A \rightarrow (A \vee B)$	(Theorem 2.2.)
2.	$p \rightarrow (p \vee q)$	(Subst. p for A, q for B in 1.)
3.	$(A \rightarrow B) \rightarrow (\neg B \rightarrow \neg A)$	(Theorem 2.16.)
4.	$(p \rightarrow (p \vee q)) \rightarrow (\neg(p \vee q) \rightarrow \neg p)$	(Subst. p for A, $(p \vee q)$ for B in 3.)
5.	$\neg(p \vee q) \rightarrow \neg p$	(Detach right side of 4, using 2.)
	Q. E. D.	



What Can AI Do: Decision Making

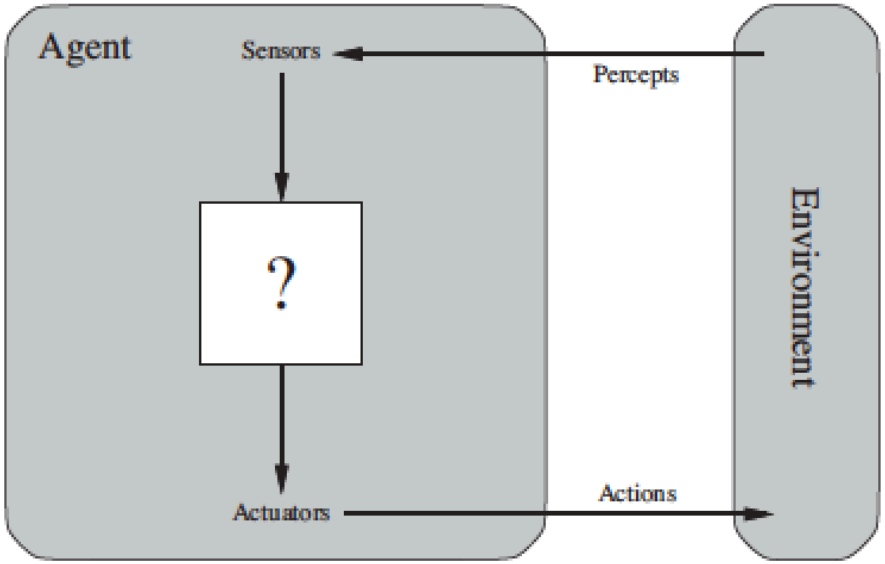
- AI algorithms are used in many kinds of applications
  - Airline routing
  - Military logistics planning
  - Google maps
  - Medical diagnosis
  - Spam classifiers
  - Automated help desks
  - Fraud detection
  - Web search engines
  - Product recommendations





What This Course is About

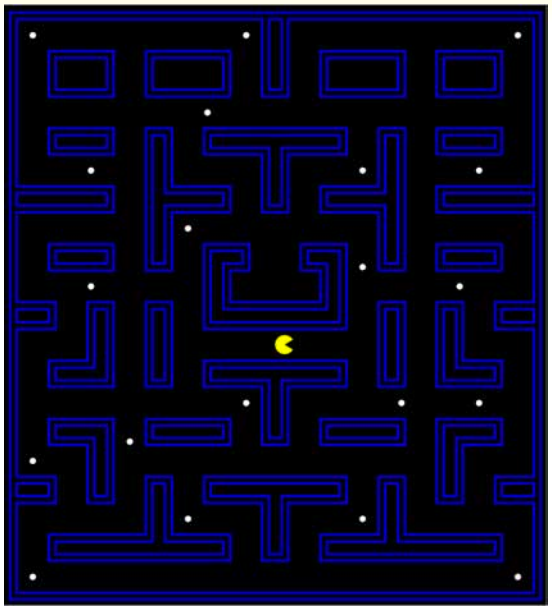
- Designing rational agents
  - Agent: perceives and acts
  - Rational: maximizes expected utility
- A rational agent makes good decisions based on:
  - percepts,
  - environment, and
  - state space (possible choices)

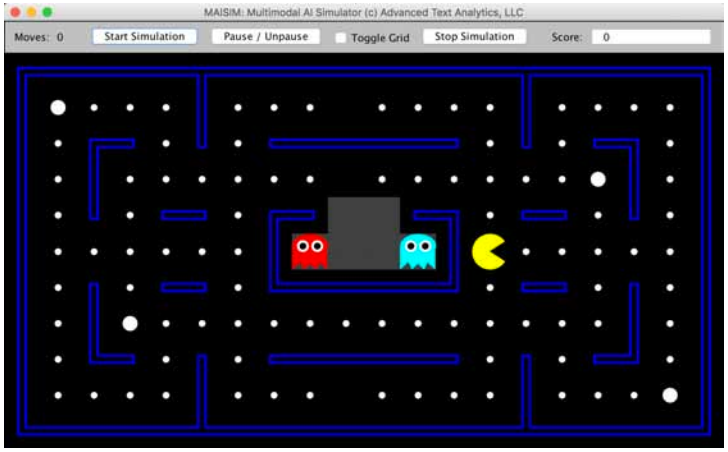


- We will learn general AI techniques and when they can be applied

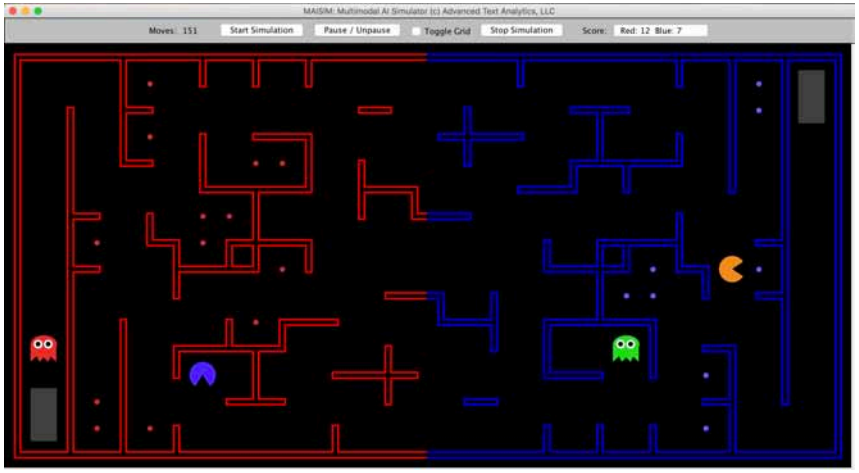
Pac-Man as Agent

- Here, we follow the lead of AI researchers Dan Klein and Peter Abbeel at UC Berkeley, and use a Pac-Man simulator (our own) to serve as a test bed for illustrating and developing AI algorithms



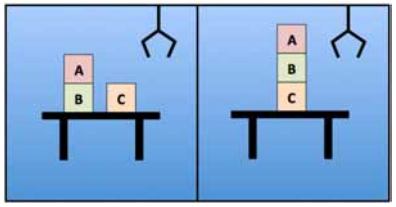


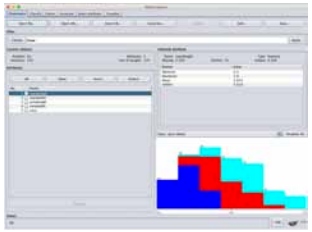
Question: Who is the "agent" in the arcade game?

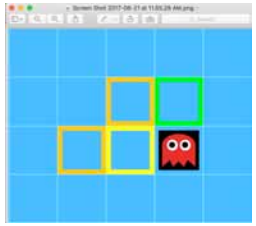


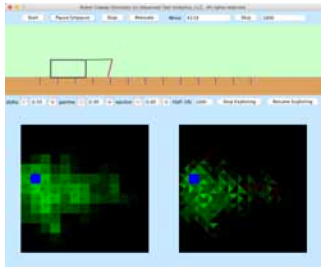
Other Tools We Will Use

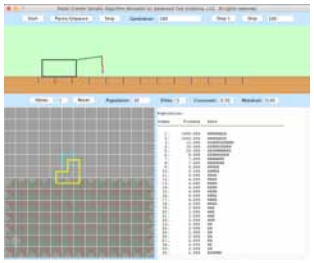
- For program assignments
  - Robot planning - Prolog
  - Classification - Weka toolkit

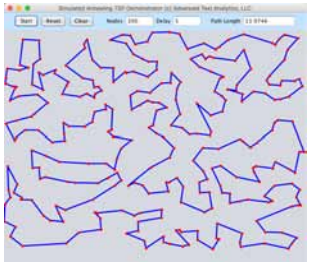




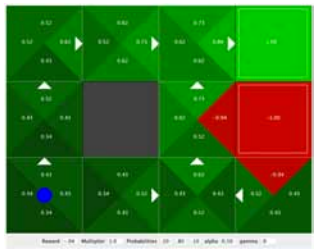


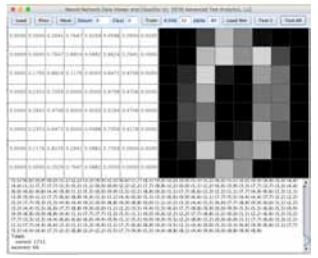


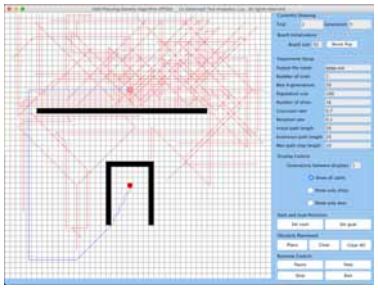


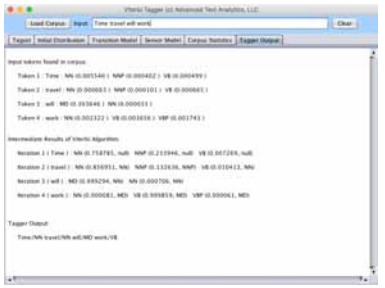


- Demonstrated in lecture
  - Ghostbusters Bayesian inferencing simulator
  - Robot Crawler, reinforcement learning and genetic algorithm versions
  - Simulated annealing demonstrator
  - Gridworld Q-learning simulator
  - Handwritten digit neural network classifier
  - Robot path planning genetic algorithm
  - Viterbi algorithm part-of-speech tagger









Course Topics

- **Part I: Intelligent Search**
  - Uninformed Search
  - Heuristic Search
  - Genetic Algorithms
  - Constraint Satisfaction
  - Adversarial Search
  - Utilities
  - Markov Decision Processes
  - Reinforcement Learning

Program 1: Pac-Man Heuristic Search (Java)

Program 2: Pac-Man Minimax Adversarial Search (Java)

- **Part II: Knowledge Representation and Reasoning**
  - Logical Agents
  - Propositional Logic
  - Prolog
  - First Order Logic
  - Classical Planning

Program 3: Two-Room Planner (Prolog)

- **[Mid-Term Exam Tuesday 3/5]**

Course Topics

- **Part III: Probabilistic Reasoning**
  - Probability and Bayes Rule
  - Markov Models
  - Hidden Markov Models
  - Bayesian Networks
  - Naive Bayes
- **Part IV: Machine Learning**
  - Decision Trees
  - Artificial Neurons and Perceptrons
  - Neural Networks
- **Part V: AI Application Areas**
  - Natural Language Processing and Understanding
  - Computer Vision and Robotics
- **[Final Exam Thursday, 4/25, 4:00 pm – 6:50 pm]**

Program 4: Perceptron Classifier (Java and Weka)

Bonus Program: Team Pac-Man (Java)

Do This Week

- Set up your software development environment
  - Java 11 (plus some IDE with a good debugger)
  - Java API documentation
  - Install SWI-Prolog
  - Install Weka

  (for above see the Programming Resources page linked from the course home page on Webcourses)
- Recommended: Get the course textbook and do Week 1 readings (see schedule)
- Read the Syllabus and Schedule
- Submit the Academic Engagement Assignment on Webcourses
  - Required for everyone, even if you are not receiving financial aid
  - Due Friday of first week of class (or as soon as you join class, whichever is later)