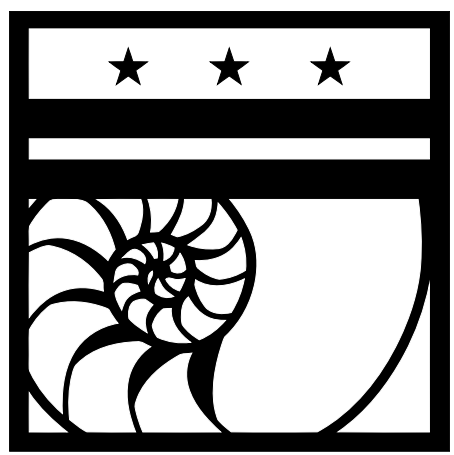
Intro to Artificial Intelligence

Natural language and ARtificial intelligence Group



Overview

- A brief history of AI
  - Beginnings
  - Winters
  - "The victory of the Neats"
- What is AI?
- breve and Braitenberg Vehicles

AI Beginnings 1950 - 70

- Turing published Computing Machinery and Intelligence which outlined the Turing Test
- First Artificial Intelligence conference is held at Dartmouth College
- Lisp is developed to allow researchers to focus on algorithms
- Natural Language Processing and Reasoning as Search are born

First Winter 1974 - 80

- Researchers underestimated the difficulty of problems within AI
- AI research funding was cut due to lack of deliverables
- Criticism of the Perceptron halts Neural Network research for 10 years

AI Boom 1980 - 90

- Expert Systems adopted by researchers and industry alike
- Neural Networks regained favor with the advent of the Backpropagation algorithm
- Lisp Machines developed to speed up the execution of complex AI code

Next Winter 1990 - 95

- Expert Systems usefulness plateau
- DARPA pulls funding from AI
- Underestimation of the difficulty of problems within AI... Again

Present 1995 - 2010

- New Agent Based intelligence gains popularity
- Field of AI fractures into Machine Learning and Artificial Intelligence
- Machine Learning gains extreme popularity and “Victory of the Neats” is declared

What is Machine Learning?

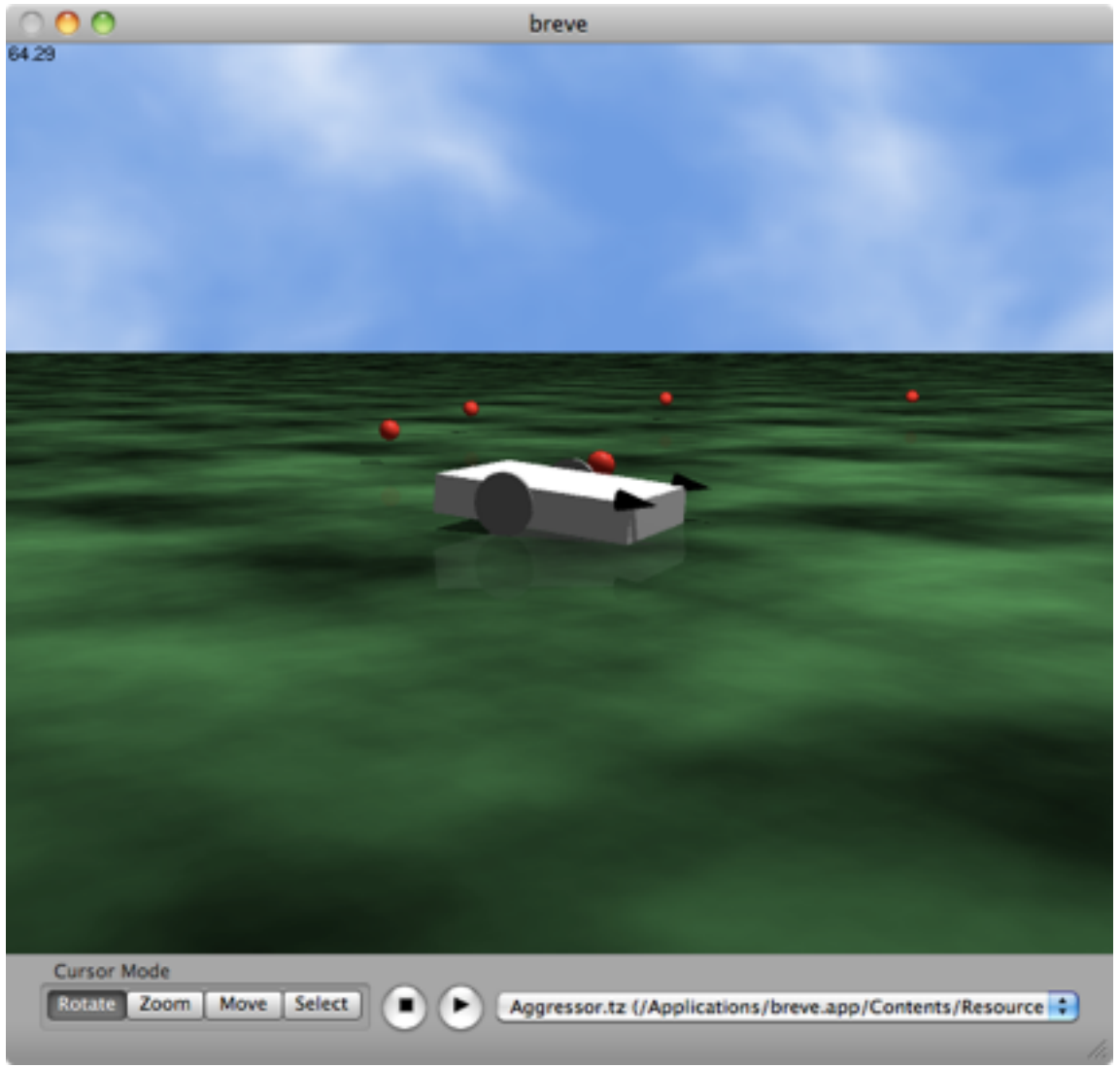
- The "Neat" approach
- Mathematical approach for dealing with problem spaces and data sets
- Often involves Data Mining, Pattern Recognition, Reinforcement Learning and Bayesian Networks
- Frequently concerned with complete data sets

What is Artificial Intelligence?

- The "Scruffy" approach
- Deals with less mathematically rigid models of intelligence
- Often involves Neural Networks, Evolutionary Systems, Agent Based Systems
- Frequently concerned with partial or limited data sets

The breve Simulation Environment

What is breve?



Hello World

```
import breve

class myController( breve.Control ):
    def __init__( self ):
        breve.Control.__init__( self )
        self.agent = myAgent()
        print '''simulation started'''

    def iterate( self ):
        breve.Control.iterate( self )

class myAgent( breve.Mobile ):
    def __init__( self ):
        breve.Mobile.__init__( self )
        print '''created agent'''


    def iterate( self ):
        # make the agent do something interesting her
        breve.Control.iterate( self )


# instantiate the controller to initialize the simulation
myController()
```

Controller

- Drives the state of the world
- Instantiates agents and objects within the world
- Iterates the simulation

```
class myController( breve.Control ):
    def __init__( self ):
        breve.Control.__init__( self )
        self.agent = myAgent()
        print '''simulation started'''

    def iterate( self ):
        breve.Control.iterate( self )
```

Agent

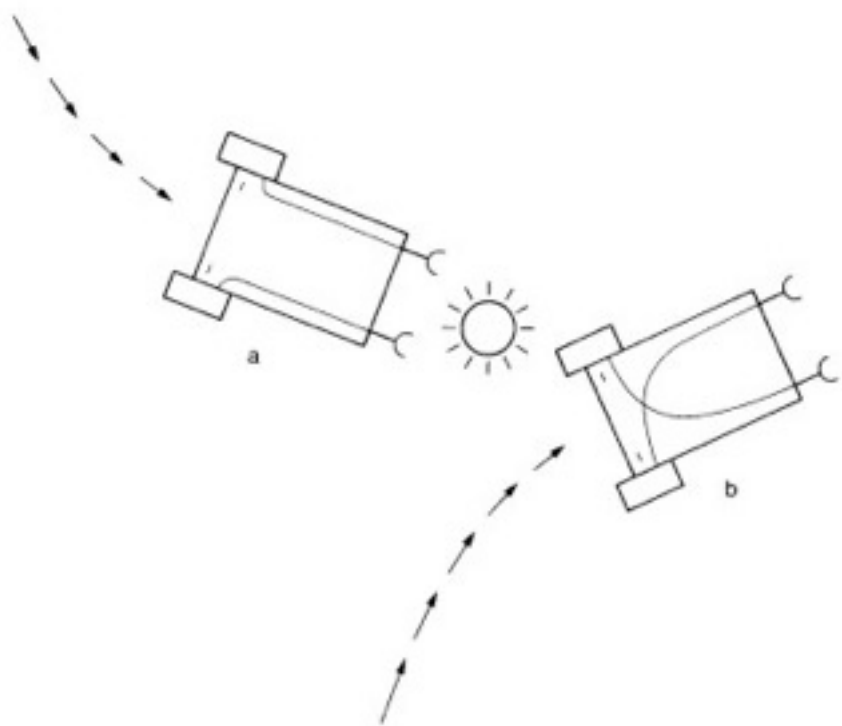
- Subclass of Mobile (means its physical and movable)
- Handles it's own iteration state

```
class myAgent( breve.Mobile ):
    def __init__( self ):
        breve.Mobile.__init__( self )
        print '''created agent'''


def iterate( self ):
        # make the agent do something
        # interesting here
        breve.Control.iterate( self )
```

Braitenburg Vehicles

- Each vehicle is comprised of three parts (chassis, actuators and sensors)
- Wiring sensors directly to actuators in different ways creates a very wide range of "intelligent" behaviors



breve Braitenburg Vehicles

- Five main classes to worry about
  - Controller (runs the simulation)
  - Vehicle (the chassis of the vehicle)
  - Sensor (measures distance to light)
  - Wheel (simply that)
  - Light (sensors read distance to this)

Demo Time