Something for everyone:

A.I. lab assignments that span learning styles and aptitudes

Christopher League
CCSC/NE
12 April 2008
Something for everyone:
wide range of educational backgrounds, learning styles, aptitudes, and time/energy constraints
“the goal of every university teacher should be to realize the potential of each student”

— Lister & Leaney
Bloom, *Taxonomy of educational objectives*, 1956
Bloom, *Taxonomy of educational objectives*, 1956

**Knowledge**
- define, describe, identify, label, list, match, name
- outline, recall, recognize, reproduce, state

**Comprehension**
- convert, estimate, explain, generalize, exemplify
- infer, interpret, paraphrase, summarize, translate

**Analysis**
- compare, contrast, deconstruct, differentiate
- distinguish, illustrate, infer, relate, separate

**Application**
- change, compute, demonstrate, discover
- operate, predict, prepare, show, solve, use

**Synthesis**
- arrange, compile, compose, create, devise
- design, extend, generate, modify, plan, write

**Evaluation**
- appraise, conclude, criticize
- critique, defend, justify, support
“IT academics place premature emphasis on the higher levels of the taxonomy”
— Lister & Leaney
Common LISPcraft

by Robert Wilensky
“Implement a constraint solver... due on Tuesday”
Knowledge
Comprehension
Analysis
Synthesis
Application
Comprehension
Knowledge
Evaluation
Workbook-style lab assignments that interleave lecture notes and software demos with a series of questions, tasks, and projects at multiple levels of Bloom’s taxonomy.
Topic Outline
1. **Philosophical background**, strong vs. weak AI, Turing test, chat-bots

2. **Machine learning by example**: classification problems, decision trees, entropy, ID3

3. **Machine learning by evolution**: optimization problems and genetic algorithms

4. Planning using **uninformed and heuristic search**: breadth-first, depth-first, and A* algorithm

5. **Constraint propagation** and satisfaction with AC3

6. **Adversarial search** with minimax & heuristics

7. **Knowledge representation**, logic, expert systems, common sense
(define my-robot
 (if s6 'north 'south))
Dr Scheme
List the values of the robot’s sensors at its current location on this map:
• Carefully compute the truth values of the following conditional expressions:

\[
(\text{or } s1 \text{ s2}) \\
(\text{not } s4) \\
(\text{and } s3 \ (\text{not } (\text{or } s4 \text{ s5}))) \\
\ldots
\]
Which direction will the robot attempt to go, if using this controller?

```
(define my-robot
  (if s6 'north 'south))
```
• Mark all the squares from which your robot should move north.
  - What features distinguish those squares from all the others?
Synthesis

- Compose and test your own robot controller
Determine how the map is specified, then design and test your own room

(define room-1
 '(
   "           xxx"
   "           xxx"
   "           xxx"
   "           xxx"
   "  xxxxx      
   "  xx  xx      
   "  xx  xx      
   "           xxx"
   "           xxx"
   "     xxx   xxx"
   "     xxx   xxx"
   "       xxxxxx  "
   "     xx   xx   "
   "     xx   xx   "
   "     xx   xx   "
   "           xxx"
   "           xxx"
   "           xxx"
   "           xxx"
   "           xxx"
   "           xxx"
   "           xxx"
   "           xxx"
   "           xxx"
   "           xxx"
   "           xxx"
   "           xxx"
))
What are some limitations of a stateless stimulus/response system?
What are some limitations of a stateless stimulus/response system?
;; Here is the planning engine.
(define visitation-plan%)
(class searchable%)
  (override start goal? state?)
  (init-field (state #f))
(define (start) (cdr (assoc state next state)))
(define (goal? p) (elements-match p (goal)))
(define (state-eq? p q) (elements-eq? p q))
(define (successors-of p) (let ((r (car p)))
  ; grid world

Welcome to DrScheme, version 371 [3m].
Language: Graphical (MrEd, includes MzScheme)
Constraint satisfaction
Constraint satisfaction
1. C’s card has **higher** rank than B’s card

2. The **sum** of C’s card with D’s card is more than 8

3. B’s card is a **black** suit (clubs or spades)

4. E’s card is **not clubs**

5. A’s card is not the same suit as C’s card
- Identify the unary constraints
- Identify the binary constraints
• Apply the unary constraints to the hand you were dealt

• Draw a graph showing the binary relationships
How many arcs are in the graph?

When your hand changes, which arcs are added to the work list?
Same process for 8-queens, but we follow through to implementation.
Topic Outline
Topic Outline

1. **Philosophical background**, strong vs. weak AI, Turing test, chat-bots

2. **Machine learning by example**: classification problems, decision trees, entropy, ID3

3. **Machine learning by evolution**: optimization problems and genetic algorithms

4. Planning using **uninformed and heuristic search**: breadth-first, depth-first, and A* algorithm

5. **Constraint propagation** and satisfaction with AC3

6. **Adversarial search** with minimax & heuristics

7. **Knowledge representation**, logic, expert systems, common sense
Connect 4

```scheme
;; Is there space for another piece at the end of this vector?
(define (space-in-vec? v)
 (= 0 (vector-last v)))

;; Drop a piece into the end of this vector.
(define (drop-in-vec player v)
 (do ((i (- (vector-length v) 1) (- i 1))
 (or (< i 0)
 (not (= 0 (vector-ref v i))))
 (vector-replace v (+ i 1) player)))

;; Is there space for another piece in column #c?
(define (space-in-column? board c)
 (space-in-vec? (vector-ref board c)))

;; Drop a piece into the top of column #c.
(define (drop-in-column player board c)
 ....)
```

> (play xy-heuristic)
#(struct:object:c4-game% ...)

Programming language: MrEd, includes MzScheme

Graphical (MrEd, includes MzScheme)
• Results: seems to work, more students submitting than usual

• “⭐⭐⭐⭐⭐”
• Start with working software
• ‘Lower’ cognitive exercises explicitly part of assignment (for credit)
• Lab work time during class hours
<table>
<thead>
<tr>
<th>Name</th>
<th>Size</th>
<th>Rev</th>
<th>Age</th>
<th>Last Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>ac3algo.scm</td>
<td>7.4kB</td>
<td>4</td>
<td>10 minutes</td>
<td>league: copy code from Fall 2007 tag into trunk, as starting point</td>
</tr>
<tr>
<td>adventure.scm</td>
<td>7.0kB</td>
<td>4</td>
<td>10 minutes</td>
<td>league: copy code from Fall 2007 tag into trunk, as starting point</td>
</tr>
<tr>
<td>connect4-brain.scm</td>
<td>9.2kB</td>
<td>4</td>
<td>10 minutes</td>
<td>league: copy code from Fall 2007 tag into trunk, as starting point</td>
</tr>
<tr>
<td>connect4-model.scm</td>
<td>6.6kB</td>
<td>4</td>
<td>10 minutes</td>
<td>league: copy code from Fall 2007 tag into trunk, as starting point</td>
</tr>
<tr>
<td>connect4-view.scm</td>
<td>5.3kB</td>
<td>4</td>
<td>10 minutes</td>
<td>league: copy code from Fall 2007 tag into trunk, as starting point</td>
</tr>
<tr>
<td>connect4.scm</td>
<td>9.7kB</td>
<td>4</td>
<td>10 minutes</td>
<td>league: copy code from Fall 2007 tag into trunk, as starting point</td>
</tr>
<tr>
<td>gene-algo.scm</td>
<td>5.1kB</td>
<td>4</td>
<td>10 minutes</td>
<td>league: copy code from Fall 2007 tag into trunk, as starting point</td>
</tr>
<tr>
<td>gene-bot.scm</td>
<td>2.0kB</td>
<td>4</td>
<td>10 minutes</td>
<td>league: copy code from Fall 2007 tag into trunk, as starting point</td>
</tr>
<tr>
<td>gene-knapsack.scm</td>
<td>2.1kB</td>
<td>4</td>
<td>10 minutes</td>
<td>league: copy code from Fall 2007 tag into trunk, as starting point</td>
</tr>
<tr>
<td>gene-prog.scm</td>
<td>6.7kB</td>
<td>4</td>
<td>10 minutes</td>
<td>league: copy code from Fall 2007 tag into trunk, as starting point</td>
</tr>
<tr>
<td>gene-tree.scm</td>
<td>4.0kB</td>
<td>4</td>
<td>10 minutes</td>
<td>league: copy code from Fall 2007 tag into trunk, as starting point</td>
</tr>
<tr>
<td>gene-vec.scm</td>
<td>1.5kB</td>
<td>4</td>
<td>10 minutes</td>
<td>league: copy code from Fall 2007 tag into trunk, as starting point</td>
</tr>
<tr>
<td>grid-main.scm</td>
<td>122 bytes</td>
<td>4</td>
<td>10 minutes</td>
<td>league: copy code from Fall 2007 tag into trunk, as starting point</td>
</tr>
<tr>
<td>grid-model.scm</td>
<td>8.5kB</td>
<td>4</td>
<td>10 minutes</td>
<td>league: copy code from Fall 2007 tag into trunk, as starting point</td>
</tr>
</tbody>
</table>