Pedagogical Possibilities for the Dice Game Pig

Todd Neller, Clif Presser – *Gettysburg College*
Ingrid Russell – *University of Hartford*
Zdravko Markov – *Central Connecticut State University*

This work was sponsored in part by NSF DUE CCLI-A&I Award Number 0409497.

Simple Examples are Teaching Treasures

- Concise specification – Simply explained, avoiding distracting details
- Elegant solution – Simply solved, highlighting target concepts
- Engages student – Sparks curiosity, nurturing intrinsic motivations

Pig

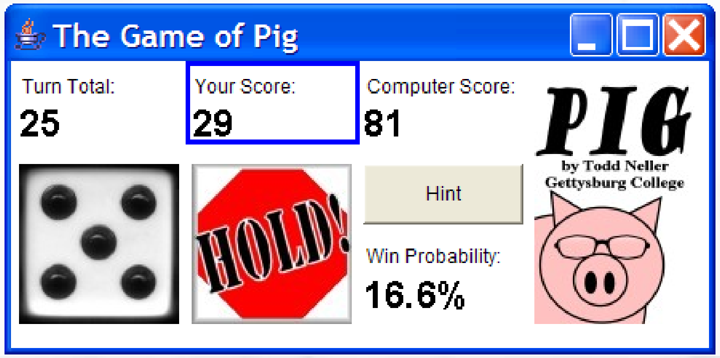
- One of the most fun, extremely simple dice games
  - “Extremely simple” – described in two sentences
  - “Fun” – spawned commercial variants, e.g. Pass the Pigs (a.k.a. Pigmania, 1977)
- Used extensively by math educators
- CS can benefit as well!

Outline

- Pig Rules
- Pedagogical Possibilities
  - CS1
  - Artificial Intelligence
  - Machine Learning
  - Graphics
  - Networking

Pig Rules

- The first player reaching **100** points wins.
- On each turn, a player rolls a die as many times as desired until either the player **holds** and scores the **sum of the rolls**, or **rolls a 1** and scores **nothing**.
- Let's play!



CS1 – Algorithms

Exercises:

1. “Hold at 20” turn
2. “Hold at 20” turn outcome distribution
3. “Hold at 20 or goal” turn
4. “Hold at 20 or goal” solitaire game
5. Average turns for 1,000,000 games of (4)
6. “Hold at 20 or goal” 2-player game
7. First-player advantage for 1,000,000 games of (6)
8. Human vs. computer game

“Hold at 20” Turn Comments

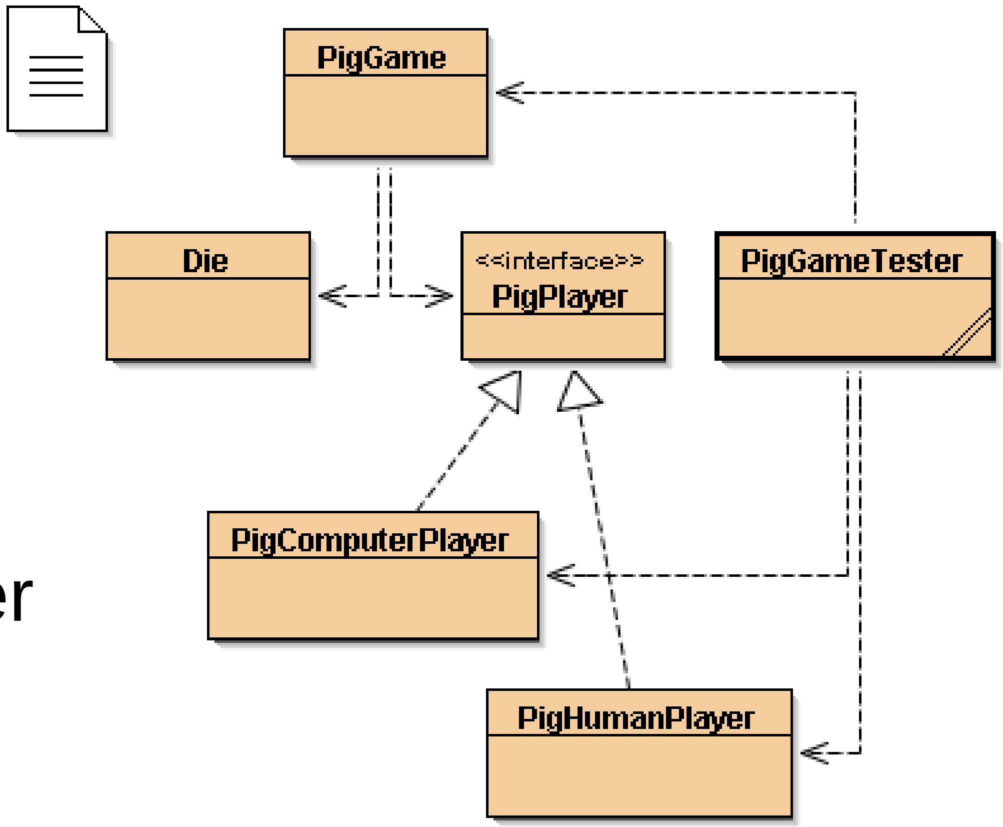
```
// Initialize variables

while () { // Turn not over
    // Roll die
    if () { // 1 rolled
        // Reset turn total and end turn
    }
    else {
        // Add roll to turn total
    }
}
```

“Hold at 20” Turn Code

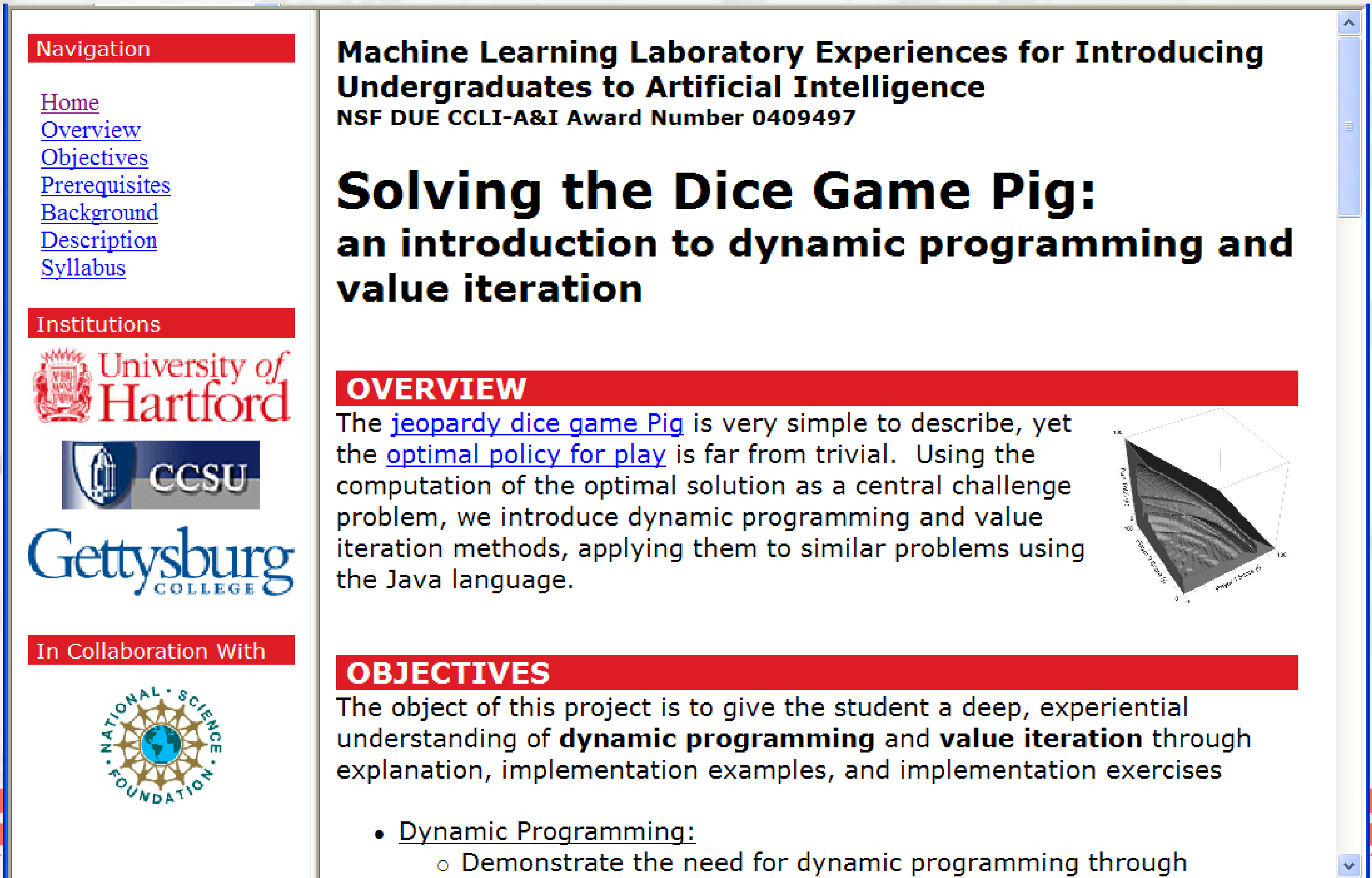
```
Random random = new Random();
int turnTotal = 0;
boolean pigRolled = false;
while (!pigRolled && turnTotal < 20) {
    int roll = random.nextInt(6) + 1;
    System.out.println("Roll: " + roll);
    pigRolled = (roll == 1);
    if (pigRolled)
        turnTotal = 0;
    else
        turnTotal += roll;
}
System.out.println("Score gain: " + turnTotal);
```

CS1 – Objects

- Objects:
  - Game
  - Die
  - Player
  - Human player
  - Computer player



MLExAI: Solving the Dice Game Pig



Navigation

- Home
- Overview
- Objectives
- Prerequisites
- Background
- Description
- Syllabus

Institutions

University of Hartford

CCSU

Gettysburg College

In Collaboration With

National Science Foundation

Machine Learning Laboratory Experiences for Introducing Undergraduates to Artificial Intelligence
NSF DUE CCLI-A&I Award Number 0409497

Solving the Dice Game Pig:
an introduction to dynamic programming and value iteration

OVERVIEW

The jeopardy dice game Pig is very simple to describe, yet the optimal policy for play is far from trivial. Using the computation of the optimal solution as a central challenge problem, we introduce dynamic programming and value iteration methods, applying them to similar problems using the Java language.

OBJECTIVES

The object of this project is to give the student a deep, experiential understanding of **dynamic programming** and **value iteration** through explanation, implementation examples, and implementation exercises

- Dynamic Programming:
  - Demonstrate the need for dynamic programming through

Optimality Equations

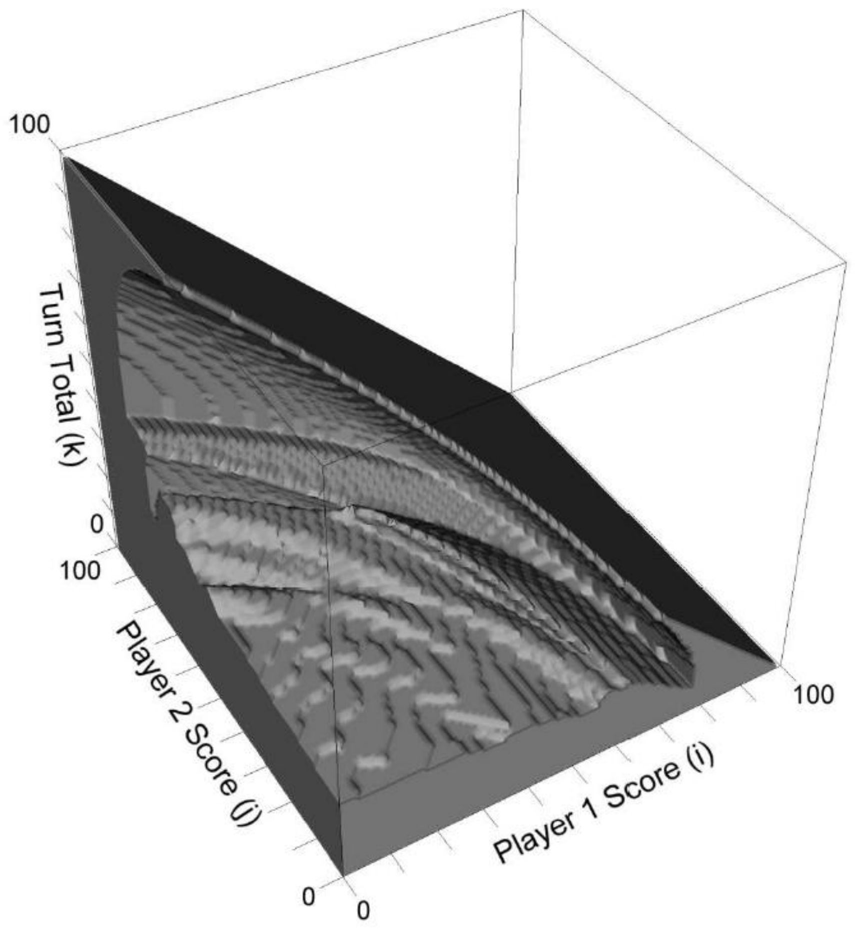
- i = player score, j = opponent score, k = turn total

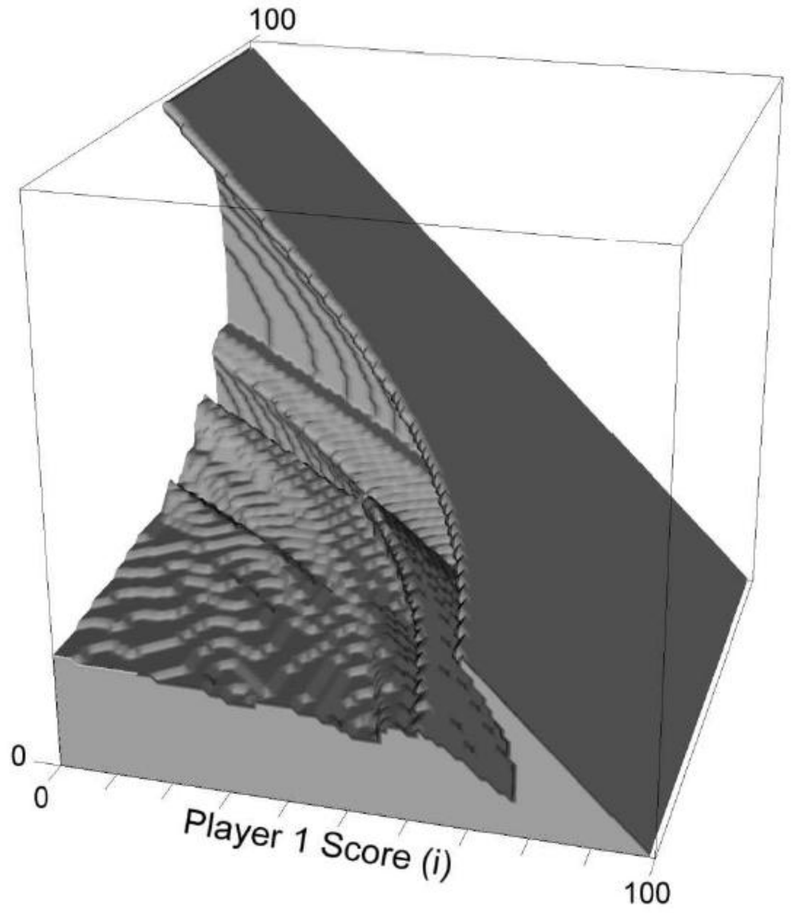
$$P_{i,j,k} = \max\left(P_{i,j,k,roll}, P_{i,j,k,hold}\right)$$

$$P_{i,j,k,roll} = \frac{1}{6}\left((1 - P_{j,i,0}) + \sum_{r=2}^{6} P_{i,j,k+r}\right)$$

$$P_{i,j,k,hold} = 1 - P_{j,i+k,0}$$

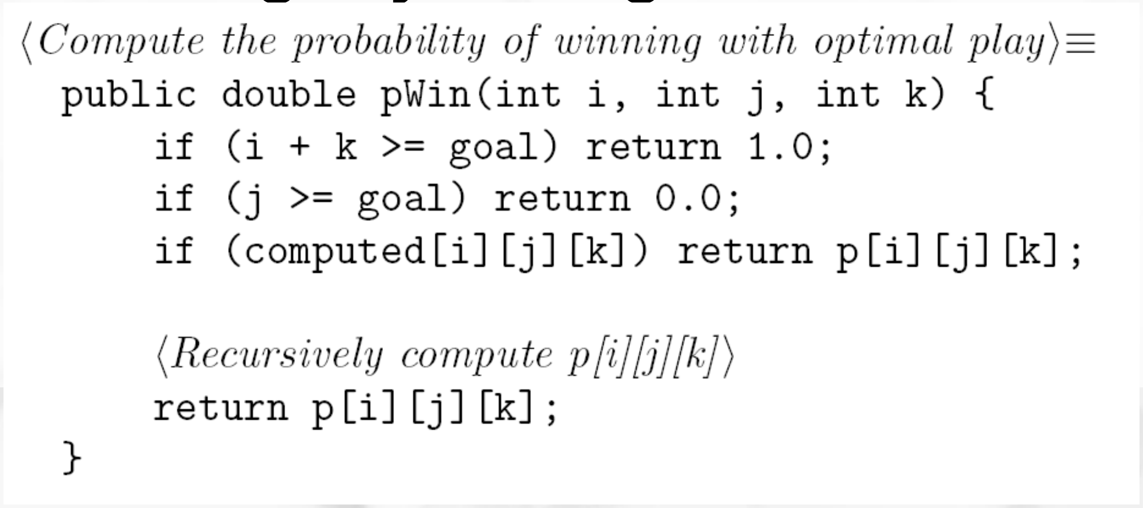
Optimal Play





Dynamic Programming

- Fibonacci example
- Progressive Pig: score at least 1 per turn → acyclic state space
- Worked example using Knuth's literate programming style, e.g.



```
⟨Compute the probability of winning with optimal play⟩≡
  public double pWin(int i, int j, int k) {
       if (i + k >= goal) return 1.0;
       if (j >= goal) return 0.0;
       if (computed[i][j][k]) return p[i][j][k];

       ⟨Recursively compute p[i][j][k]⟩
       return p[i][j][k];
  }
```

Dynamic Programming (cont.)

- Exercises:
  - Pig solitaire: reach goal score *g* in *n* turns
  - Pig solitaire 2: maximize score in *n* turns
  - THINK solitaire: 2-dice Pig variant in 5 turns
  - Advanced projects:
    - Risk board game analysis
    - Simple Yahtzee variant analysis

Value Iteration

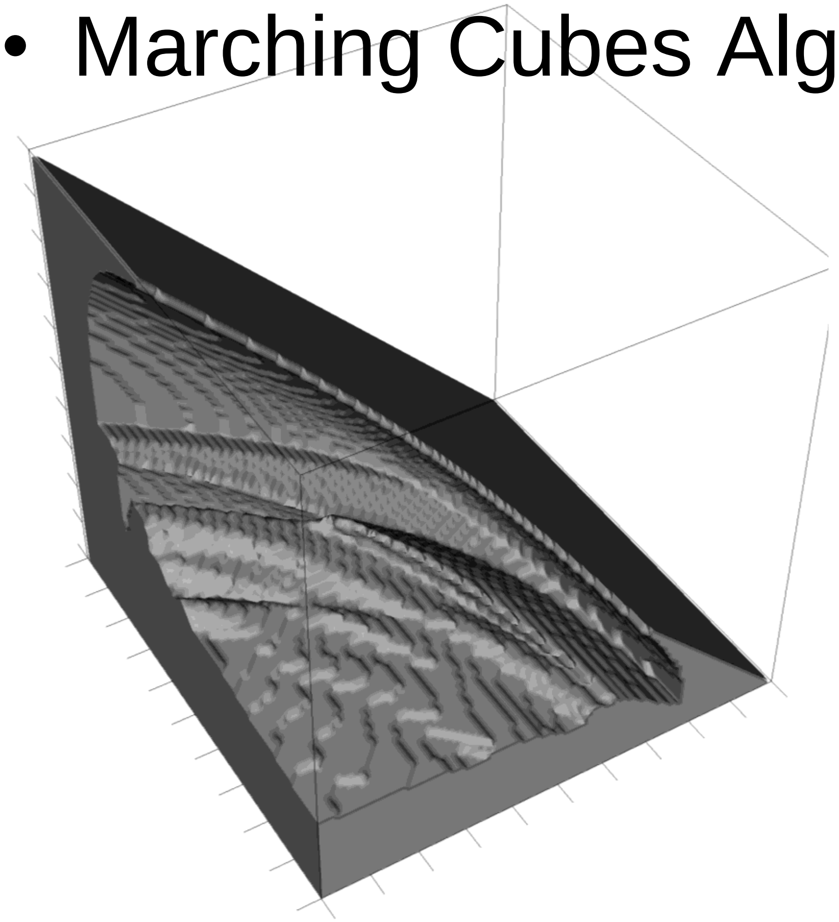
- Worked example: Piglet
  - Pig with a coin and goal score of 10
  - Score heads flipped or nothing if a tail is flipped
- Exercises:
  - Pig
  - Pig Solitaire 3: minimize turns to reach goal score *g*
  - Pass the Pigs
  - Advanced projects:
    - Hog: Pig with single throw of as many dice as desired
    - 10,000: jeopardy dice game

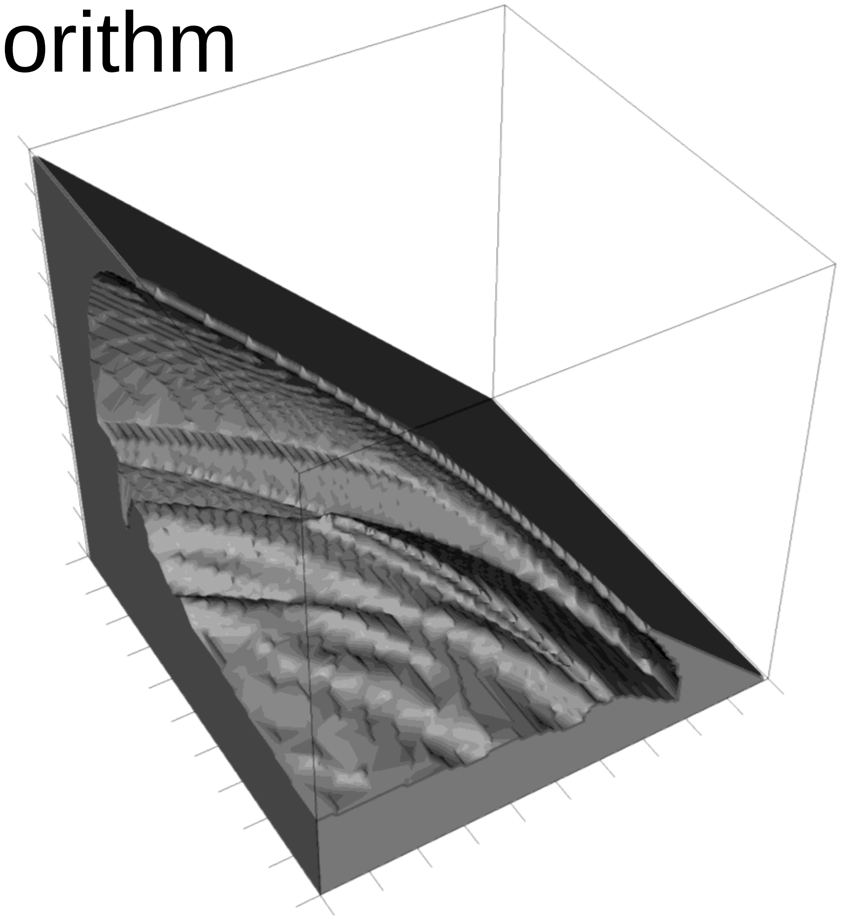
Machine Learning

- Comparative study of RL algorithms:
  - Useful as simple (but not trivial) benchmark
  - Precise optimal policy/state values known through value iteration
  - Monte-Carlo/TD learning: model free, but slower convergence for infrequent states
  - Transition to more complex Markov games

Scientific Visualization

- Marching Cubes Algorithm





Networking

- Illustrate network programming
  - Client-server, peer-to-peer (P2P), …
  - Simple game → focus on networking concepts, not game
  - Fun factor

What Other Courses?

- This is what has been done.
- What other potentials are there?
  - Game Programming?
  - Software Engineering?
  - GUI Design?
  - <Insert your course here>?

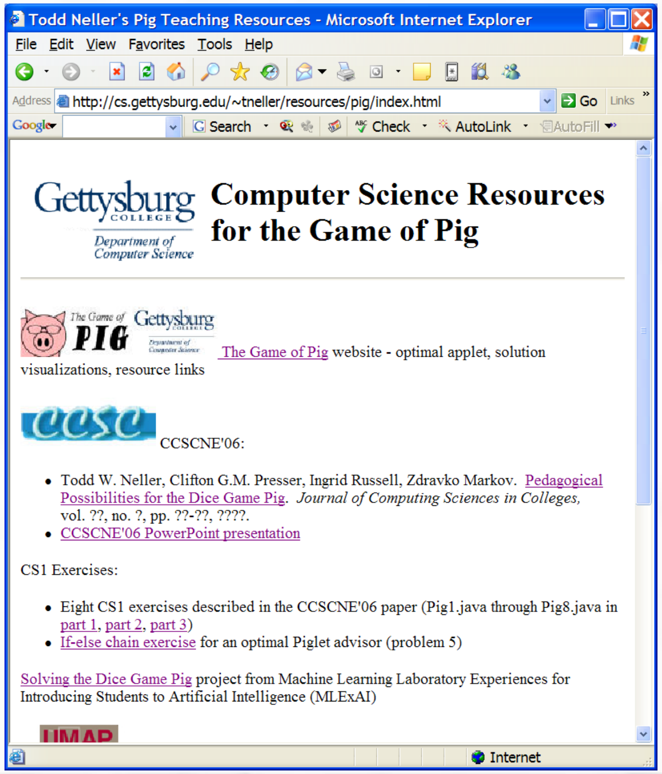
Summary

- Simple examples are teaching treasures.
- Pig is a simple, fun teaching example for:
  - CS1 (algorithm design, OO design)
  - AI (MDPs, DP, value iteration)
  - Machine Learning, Graphics, Networking
  - Etc.

Resources

- http://cs.gettysburg.edu/~tneller/resources/pig
  - Game of Pig Website
  - CCSCNE paper, PPT
  - CS1 Exercises
  - MLExAI Pig project
  - 2 UMAP Journal papers

This work was sponsored in part by NSF DUE CCLI-A&I Award Number 0409497.


Todd Neller's Pig Teaching Resources - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://cs.gettysburg.edu/~tneller/resources/pig/index.html
Go Links
Google Search Check AutoLink AutoFill
Gettysburg College Department of Computer Science
The Game of PIG
Internet

Computer Science Resources for the Game of Pig

The Game of Pig website - optimal applet, solution visualizations, resource links

CCSCNE'06:

- Todd W. Neller, Clifton G.M. Presser, Ingrid Russell, Zdravko Markov. Pedagogical Possibilities for the Dice Game Pig. *Journal of Computing Sciences in Colleges*, vol. ??, no. ?, pp. ??-??, ????.
- CCSCNE'06 PowerPoint presentation

CS1 Exercises:

- Eight CS1 exercises described in the CCSCNE'06 paper (Pig1.java through Pig8.java in part 1, part 2, part 3)
- If-else chain exercise for an optimal Piglet advisor (problem 5)

Solving the Dice Game Pig project from Machine Learning Laboratory Experiences for Introducing Students to Artificial Intelligence (MLExAI)

UMAP