



11-9-2017

The Birds of a Feather Research Challenge

Todd W. Neller
Gettysburg College

Follow this and additional works at: <https://cupola.gettysburg.edu/csfac>

 Part of the [Computer Sciences Commons](#), [Game Design Commons](#), and the [Science and Mathematics Education Commons](#)

Share feedback about the accessibility of this item.

Neller, Todd W. "The Birds of a Feather Research Challenge." Presented at the Gettysburg College Computer Science Colloquium, Gettysburg, PA, November 2017.

This is the author's version of the work. This publication appears in Gettysburg College's institutional repository by permission of the copyright owner for personal use, not for redistribution. Cupola permanent link: <https://cupola.gettysburg.edu/csfac/45>

This open access conference material is brought to you by The Cupola: Scholarship at Gettysburg College. It has been accepted for inclusion by an authorized administrator of The Cupola. For more information, please contact cupola@gettysburg.edu.

The Birds of a Feather Research Challenge

Abstract

Neller presented a set of research challenges for undergraduates that allow an excellent formative experience of research, writing, peer review, and potential presentation and publication through a top-tier conference. The focus problem is the analysis of a newly-designed solitaire card game, Birds of a Feather, so potentials for discovery abound. Open access talk slides, research code, solvability data sets, research tutorial videos, and more are also available at <http://cs.gettysburg.edu/~tneller/puzzles/boaf>.

Keywords

undergraduate research, games, Birds of a Feather, artificial intelligence

Disciplines

Computer Sciences | Game Design | Science and Mathematics Education

Comments

This talk was originally presented at the Gettysburg College Computer Science Colloquium on November 9th, 2017. A follow-up research workshop was held for Gettysburg students at our Computer Science Colloquium on February 1st, 2018. It was also presented at the Eighth Symposium on Educational Advances in Artificial Intelligence 2018 on February 3rd, 2018 and Dickinson College on February 27th, 2018.

The Birds of a Feather Research Challenge

Todd W. Neller
Gettysburg College
November 9th, 2017

Outline

- Backstories:
 - Rook Jumping Mazes
 - Parameterized Poker Squares
 - FreeCell
- Birds of a Feather
 - Rules
 - 4x4 Single Stack Play
 - Experiments
 - Brainstorming

Rook Jumping Maze Design

- [Rook Jumping Mazes](#) - logic mazes with simple rules based on Chess rook moves
- Few maze designers in history had the skill to create these.
- We worked together to create a metric to rate the quality of mazes and performed combinatorial optimization to [generate high quality mazes](#).

Example Maze

- ▶ **Specification:** grid size, start state (square), goal state, jump numbers for each non-goal state.
- ▶ **Jump number:** Move *exactly* that many squares up, down, left, right. (*Not* diagonally.)
- ▶ **Objectives:**
 - Find a path from start to goal.
 - Find the shortest of these paths.

3	4	1	3	1
3	3	3	G	2
3	1	2	2	3
4	2	3	3	3
4	1	4	3	2

Publication

Rook Jumping Maze Design Considerations

Todd W. Neller¹, Adrian Fisher², Munyaradzi T. Choga¹,
Samir M. Lalvani¹, and Kyle D. McCarty¹

¹ Gettysburg College, Dept. of Computer Science, Gettysburg, Pennsylvania, 17325, USA,
tneller@gettysburg.edu,

WWW home page: <http://cs.gettysburg.edu/~tneller>

² Adrian Fisher Design Ltd., Portman Lodge, Durweston, Dorset, DT11 0QA England,
adrian@adrianfisherdesign.com,

WWW home page: <http://www.adrianfisherdesign.com>

Abstract. We define the Rook Jumping Maze, provide historical perspective, and describe a generation method for such mazes. When applying stochastic local search algorithms to maze design, most creative effort concerns the definition of an objective function that rates maze quality. We define and discuss several maze features to consider in such a function definition. Finally, we share our preferred design choices, make design process observations, and note the applicability of these techniques to variations of the Rook Jumping Maze.

Parameterized Poker Squares

- Materials:
 - shuffled standard (French) 52-card deck,
 - paper with 5-by-5 grid, and
 - pencil
- Each turn, a player draws a card and writes the card rank and suit in an empty grid position.
- After 25 turns, the grid is full and the player scores each grid row and column as a 5-card poker hand according to a point system.

American Point System

<u>Poker Hand</u>	<u>Points</u>	<u>Description</u>	<u>Example</u>
Royal Flush	100	A 10-J-Q-K-A sequence all of the same suit	10♣, J♣, Q♣, K♣, A♣
Straight Flush	75	Five cards in sequence all of the same suit	A♦, 2♦, 3♦, 4♦, 5♦
Four of a Kind	50	Four cards of the same rank	9♣, 9♦, 9♥, 9♠, 6♥
Full House	25	Three cards of one rank with two cards of another rank	7♠, 7♣, 7♦, 8♥, 8♠
Flush	20	Five cards all of the same suit	A♥, 2♥, 3♥, 5♥, 8♥
Straight	15	Five cards in sequence; Aces may be high or low but not both	8♣, 9♠, 10♥, J♦, Q♣
Three of a Kind	10	Three cards of the same rank	2♠, 2♥, 2♦, 5♣, 7♠
Two Pair	5	Two cards of one rank with two cards of another rank	3♥, 3♦, 4♣, 4♠, A♣
One Pair	2	Two cards of one rank	5♦, 5♥, 9♣, Q♠, A♥
High Card	0	None of the above	2♦, 3♣, 5♠, 8♥, Q♦

Scoring Examples

PySol - Poker Square

File Select Edit Game Assist Options Help

tneller

0

Royal Flush	100	1
Straight Flush	75	0
Four of a Kind	50	0
Full House	25	2
Flush	20	3
Straight	15	0
Three of a Kind	10	1
Two Pair	5	1
One Pair	2	2

WON

Total: 229

100 20 20 20 2

1:38 25/25 147: 89/58

PySol - Poker Square

File Select Edit Game Assist Options Help

tneller

0

Royal Flush	100	0
Straight Flush	75	1
Four of a Kind	50	2
Full House	25	0
Flush	20	2
Straight	15	0
Three of a Kind	10	1
Two Pair	5	0
One Pair	2	2

WON

Total: 229

20 0 75 20 0

1:37 25/25 151: 92/59

Competition Results

Parameterized Poker Squares Results

Players Mean Scores by Point System

	American	Ameritish	British	Hypercorner	Random	High Card	One Pair	Two Pair	3 of a Kind	Straight	Flush	Full House
BMO_V2	125.27	105.54	54.50	1.10	437.77	9.37	9.12	4.46	3.20	2.97	3.43	1.82
DevneilPlayer	14.36	15.27	7.51	-9.52	-86.92	5.22	4.10	0.45	0.21	0.04	0.05	0.03
Gettysburg	123.94	110.28	53.38	1.24	429.89	9.37	9.17	4.47	3.02	2.71	3.46	1.93
SRulerPlayer	51.83	55.39	30.29	-5.10	242.85	9.34	8.84	4.04	2.10	1.58	1.98	0.61
JoTriz	116.75	109.03	53.59	-0.78	351.07	9.31	9.15	4.59	3.03	2.59	3.36	1.67
xRandomRolloutPruningPlayer	116.12	111.26	53.92	-2.20	411.78	9.35	9.16	4.52	2.89	2.94	3.41	1.82
MonteCarloTreePlayer	15.47	15.31	7.61	-9.30	-86.83	4.80	4.53	0.45	0.20	0.05	0.02	0.00
RandomPlayer	14.25	15.67	7.71	-9.66	-106.80	5.20	4.31	0.42	0.23	0.01	0.01	0.01
Max	125.27	111.26	54.50	1.24	437.77	9.37	9.17	4.59	3.20	2.97	3.46	1.93
Min	14.25	15.27	7.51	-9.66	-106.80	4.80	4.10	0.42	0.20	0.01	0.01	0.00

Normalized Scores

	American	Ameritish	British	Hypercorner	Random	High Card	One Pair	Two Pair	3 of a Kind	Straight	Flush	Full House	Total
BMO_V2	1.00	0.94	1.00	0.99	1.00	1.00	0.99	0.97	1.00	1.00	0.99	0.94	11.821
DevneilPlayer	0.00	0.00	0.00	0.01	0.04	0.09	0.00	0.01	0.00	0.01	0.01	0.02	0.190
Gettysburg	0.99	0.99	0.98	1.00	0.99	1.00	1.00	0.97	0.94	0.91	1.00	1.00	11.763
SRulerPlayer	0.34	0.42	0.48	0.42	0.64	0.99	0.93	0.87	0.63	0.53	0.57	0.32	7.149
JoTriz	0.92	0.98	0.98	0.81	0.84	0.99	1.00	1.00	0.94	0.87	0.97	0.87	11.170
xRandomRolloutPruningPlayer	0.92	1.00	0.99	0.68	0.95	1.00	1.00	0.98	0.90	0.99	0.99	0.94	11.334
MonteCarloTreePlayer	0.01	0.00	0.00	0.03	0.04	0.00	0.08	0.01	0.00	0.01	0.00	0.00	0.192
RandomPlayer	0.00	0.00	0.00	0.00	0.00	0.09	0.04	0.00	0.01	0.00	0.00	0.01	0.153

[Publications](#)

Monte Carlo Approaches to Parameterized Poker Squares

Todd W. Neller¹(✉), Zuozhi Yang¹, Colin M. Messinger¹, Calin Anton²,
Karo Castro-Wunsch², William Maga², Steven Bogaerts³,
Robert Arrington³, and Clay Langley³

Proceedings of the Sixth Symposium on Educational Advances in Artificial Intelligence (EAAI-16)

Proceedings of the Sixth Symposium on Educational Advances in Artificial Intelligence (EAAI-16)

BeeMo, a Monte Carlo Simulation Agent for Playing Parameterized Poker Squares

Karo Castro-Wunsch, William Maga, Calin Anton

MacEwan University, Edmonton, Alberta, Canada

karoantonio@gmail.com, magaw@mymacewan.ca, antonc@macewan.ca

Using Domain Knowledge to Improve Monte-Carlo Tree Search Performance in Parameterized Poker Squares

Robert Arrington, Clay Langley, and Steven Bogaerts

Department of Computer Science

DePauw University

Greencastle, IN, USA

{robertarrington.2015, claylangley.2017, stevenbogaerts}@depauw.edu

Abstract

Parameterized Poker Squares is a two-player, perfect information, game playing agent. We organized the search space into three dimensions: partial hand patterns, search algorithms, and partial hand utilities. In this paper, we describe the dimension we implemented in BeeMo, our final product. BeeMo uses Monte-Carlo search. The search is based on hand patterns utility through an iterative improvement Monte-Carlo simulations and optimi-

Proceedings of the Sixth Symposium on Educational Advances in Artificial Intelligence (EAAI-16)

Learning and Using Hand Abstraction Values for Parameterized Poker Squares

Todd W. Neller and Colin M. Messinger and Zuozhi Yang

Gettysburg College

tneller@gettysburg.edu

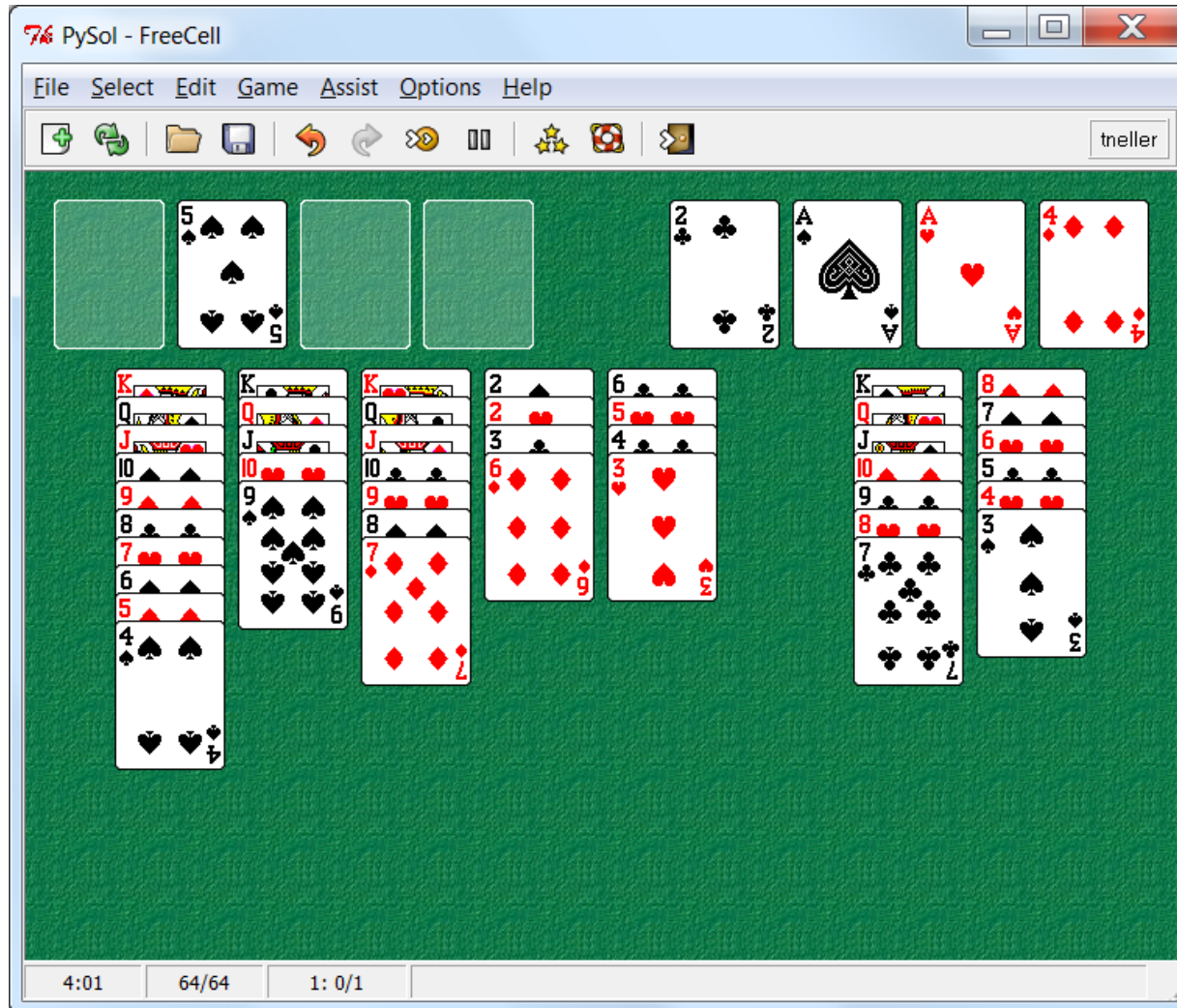
Abstract

five rows of five cards each. The object is to make as high a total score as possible, in the ten Poker hands

Table 1: Example scoring s

Hand Type	Am.	Brit.	E.I.
royal flush	100	30	.
straight flush	75	30	.
4 of a kind	50	16	.
straight	25	10	.
full house	20	5	.
3 of a kind	15	12	.
flush	10	6	.
2 pair	5	3	.
1 pair	2	1	.
high card	0	0	.

FreeCell Solitaire Card Game



FreeCell Characteristics

- Randomly generated, but no chance after face-up deal (perfect information) → Combinatorial game
- Self-generating puzzle that is solvable with high probability
- Invited many interesting research questions posed and solved by skilled enthusiasts
 - See Michael Keller's FreeCell FAQ:
<http://solitairelaboratory.com/fcfaq.html>
- Would have been great for undergraduate research, but largely harvested.
- To gain new low-hanging fruit, plant a new tree!

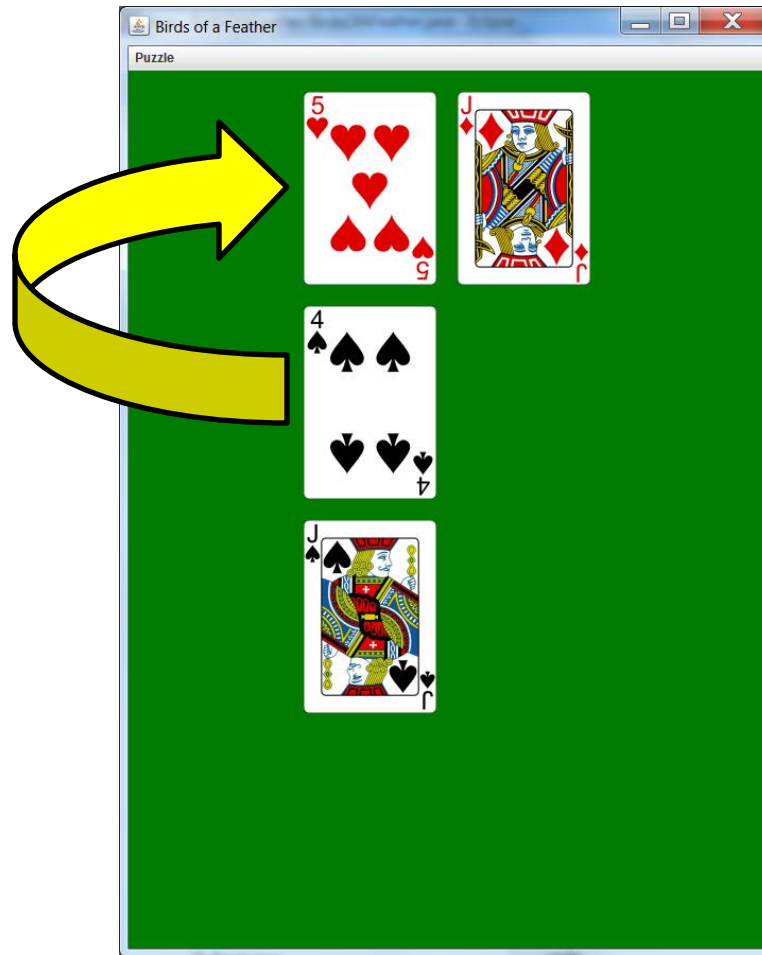
Birds of a Feather Characteristics

- Randomly generated, but no chance after face-up deal (perfect information) → Combinatorial game
- Self-generating puzzle that is solvable (for certain deal dimensions) with high probability
- Invites many interesting research questions (to be introduced later)
- But first, we introduce the game...

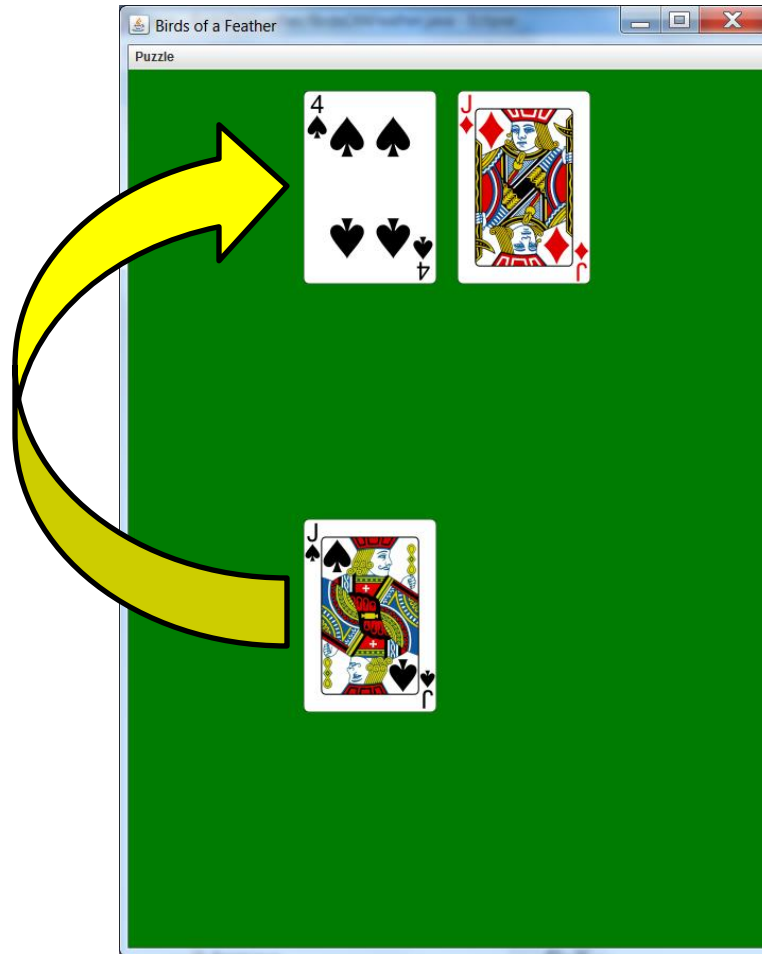
Birds of a Feather

- [“Birds of a feather flock together.”](#)
- Designed August 9, 2016
- **Materials:** a standard, shuffled 52-card deck
- **Setup:** Deal cards singly, face-up into a grid (e.g. 4-by-4).
- **Object:** Form a single stack of all cards.
- **Play:** A player may move one stack of cards onto another stack of cards in the *same row or column* if the cards on top of the stacks have either
 - (1) the *same suits*, or
 - (2) the *same or adjacent ranks*. Aces are low and not adjacent to kings, so rank adjacency is according the ordering A, 2, 3, ..., J, Q, K.

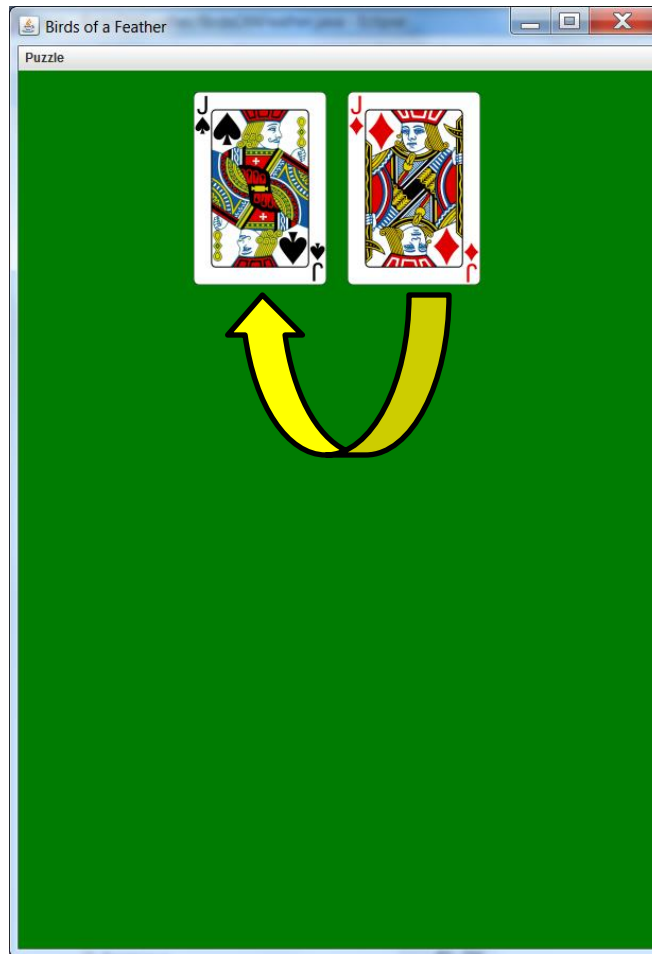
Birds of a Feather: Adjacent Rank



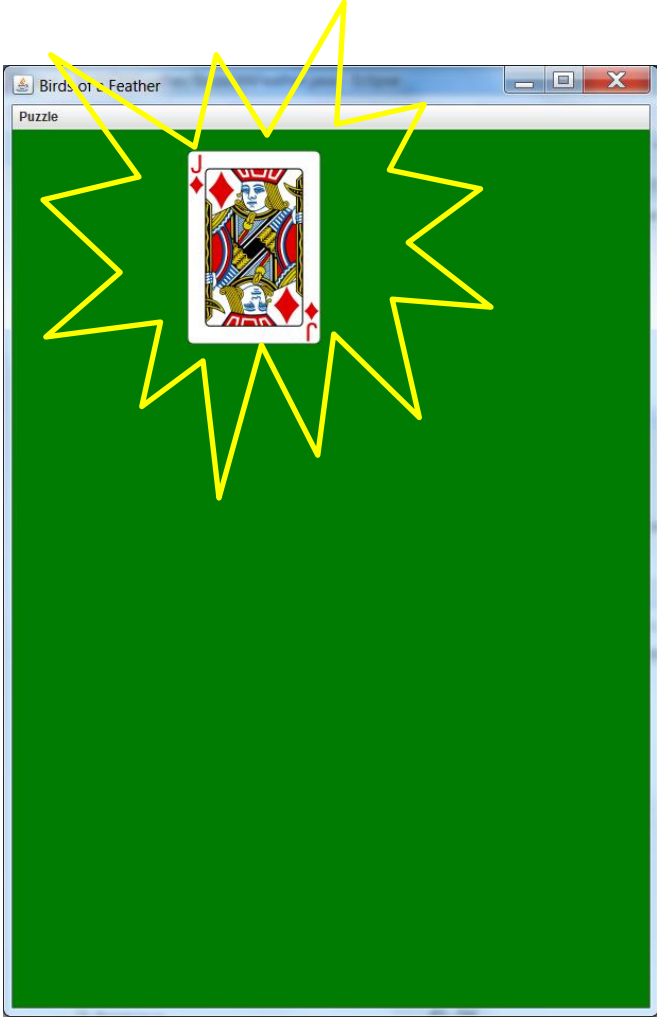
Birds of a Feather: Same Suit



Birds of a Feather: Same Rank



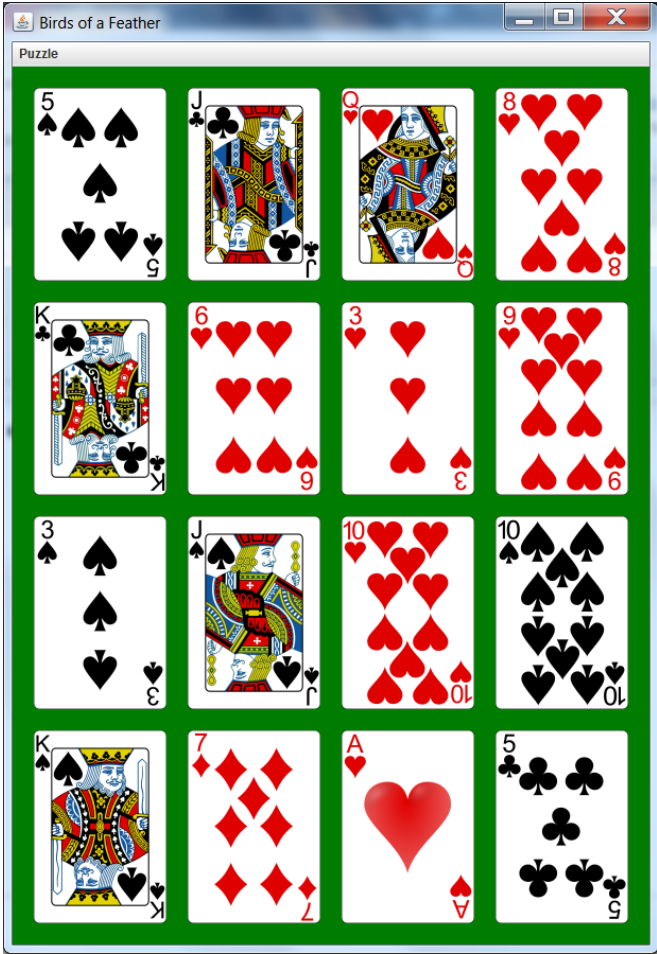
Birds of a Feather: Single Stack Goal



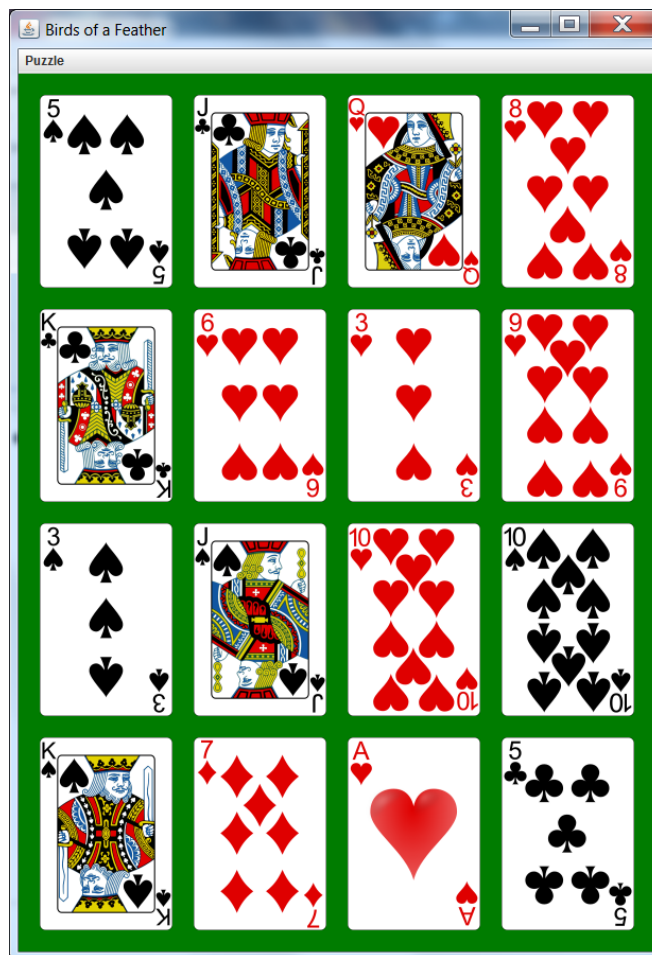
Birds of a Feather Java Project

- Import Eclipse Java project...
 - File → Import... → General → Existing Project into Workspace
 - Select “Select Archive File” radio button and click “Browse” button.
 - Select
/Courses/Colloquia/tneller171109/BirdsOfAFeather.zip
 - Click “Finish”
- Open BirdsOfAFeather.java (GUI) in project BirdsOfAFeather
- Run

Birds of a Feather Example Deal



Birds of a Feather Example Solution



TS-9H AH-TH AH-3H
 AH-QH 6H-7D JS-JC
 KS-3S KS-KC 5S-KS
 6H-5C 5S-TS AH-8H
 6H-AH 5S-6H 5S-JS

Experiment 1: Are all 4x4 deals solvable?

- Open Experiment1.java and run.
 - Program attempts to solve deal 0, 1, 2, ...
 - Are all solvable? If not, what is the first that isn't solvable?
- Press the red square by the Eclipse console window to terminate the experiment.
- Run BirdsOfAFeather and type 's' to create seed 10 puzzle.
- What makes this unsolvable?
- Type 't' to toggle connections between flockable card pairs.
- Let's call a single unflockable card an *odd bird*.

Experiment 2: Do all unsolvable deals have an odd bird?

- For efficiency, we'll divide the solving ranges and work in parallel. Wait for your input parameters.
- Open Experiment2.java and run. Odd bird deals will be identified and skipped.
- Every time you find an unsolvable deal (odd bird or otherwise), add it to your list of unsolvable deal numbers and note if it is an odd bird deal.
- After covering your range, examine any unsolvable deals using BirdsOfAFeather.java. If there are no unsolvable deals or they're easily understood, examine deal 1163 or 1264.

Example Research Questions

- m-by-n single-stack deal solvability/scoring:
 - What is the probability that a deal will have a single-stack solution?
 - What is the maximal score distribution of deals?
 - Which features of a puzzle can guide a *computer* efficiently to a successful solution?
 - Which algorithms work best to solve such puzzles?
 - What are characteristics of grids without single-stack solutions?
- Puzzle solving heuristics:
 - Which features of a puzzle can guide a *player* to a successful solution?
- Puzzle design:
 - How would you create a metric for a good BirdsOfAFeather puzzle?
 - Which techniques work best for designing puzzles for such a metric?
 - How can one generate a succession of gradually more difficult puzzles to guide and teach a player to solve such puzzles well?

Conclusion

- Birds of a Feather offers a new landscape for research exploration.
- Together, we can understand more about puzzle solving, puzzle design, etc. *and* gain research, writing, presentation, and publication experience in the process.
- Please contact Todd Neller (tneller@gettysburg.edu) if you're interested!