

The Parameterized Poker Squares EAAI NSG Challenge

What is the EAAI NSG Challenge?

- Goal: a fun way to encourage good, faculty-mentored undergraduate research experiences that includes an option for peer-reviewed paper publication and presentation at a major CS conference.
- Students may work independently or in teams with a faculty mentor to meet the challenge.
- Challenge submissions and associated papers would be submitted at the following EAAI paper submission deadline (late-summer, one month before AAAI submission deadline).
- At the next EAAI (February 2016): challenge results, accepted paper presentations, announcement of next NSG Challenge
- EAAI 2016 NSG Challenge: **Parameterized Poker Squares**

Poker Squares (Original)

- Materials:
 - shuffled standard (French) 52-card 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 given 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

Let's Play

- Web: <http://games.emantranet.com/PokerSquare.aspx>
- Other platforms: See <http://tinyurl.com/pokersqrs>

New Game

20 20 75

5

2

50

10

Hand Type	Points
Royal Flush	100
Straight Flush	75
Four of a kind	50
Full House	25
Flush	20
Straight	15
Three Of A Kind	10
Two Pair	5
One Pair	2

Poker Square Solitaire Copyright © www.emantranet.com

Parameterization of Poker Squares

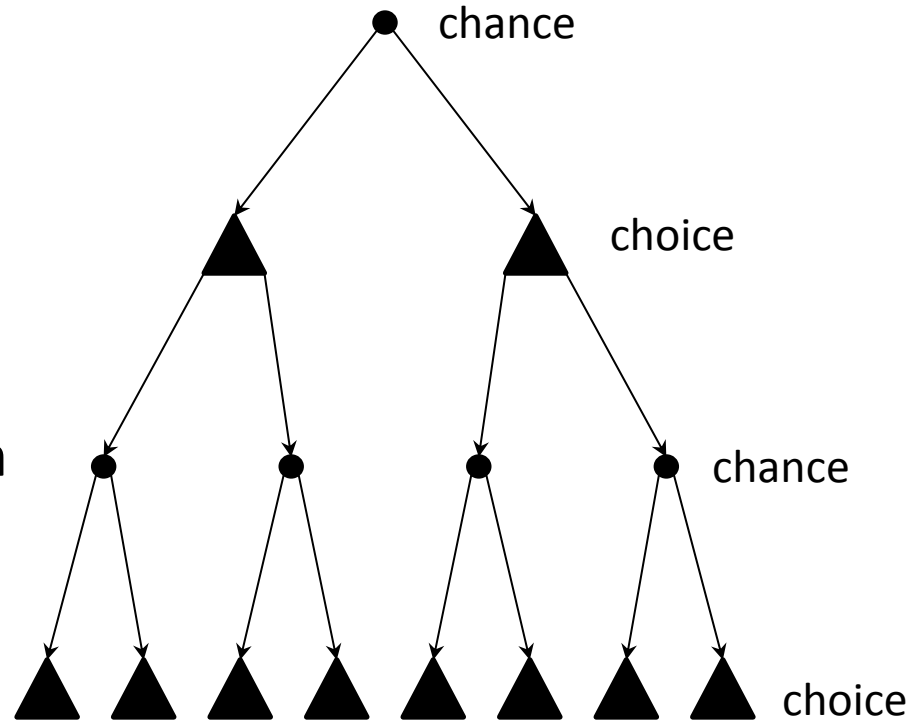
- The American Point System (0, 2, 5, 10, 15, 20, 25, 50, 75, 100) is based on hand rank in Poker.
- The British Point System (1, 3, 6, 12, 5, 10, 16, 30, 30) is based on the difficulty of forming the hands in Poker Squares. (a.k.a. English Point System)
- For our challenge, AI players will be *given the scoring system at play time* with points in the range [-128, 127].
Examples:
 - Ameritish point systems: random variations on American and British systems
 - Single Hand: 1 point for one hand type, 0 points otherwise
 - Hypercorners: all 1 or -1 score values
 - Random: random score system in range [-128, 127]

Player Interface

- Java software supplied manages games, tournaments, scoring, partial scoring
- The Java PokerSquaresPlayer interface is very simple:
 - **void setPointSystem(PokerSquaresPointSystem system, long millis)** – Player is given a point system and a given amount of milliseconds (300K) to form strategy from the point system
 - **void init()** – Player initializes before a game (e.g. clear board, game history)
 - **int[] getPlay(Card card, long millisRemaining)** – Given a drawn card and the number of ms remaining in the 30 second game, return the 0-based row and column the player chooses for the Card
 - **String getName()** – return the player name

Structure of the Game

- The game is structured as an alternating sequence of *chance nodes* and player *choice nodes*.
 - Each card draw is a probabilistic event where any remaining card is drawn with equal probability.
 - Each player *action* is a commitment to a card placement.



Game Tree Size

- How big is the Poker Squares game tree?
 - Root chance node: 52 possible cards
 - 52 depth-1 choice nodes: 25 possible placements
 - 52x25 depth-2 chance nodes: 51 possible cards
 - 52x25x51 depth-3 choice nodes: 24 possible placements
 - ...
 - $52!/27! \times 25! = 52!/(27 \times 26) \cong 1.15 \times 10^{65}$ nodes
 - Although:
 - Different draw/play sequences can lead to the same state.
 - Rows/columns may be reordered without affecting score.
 - Still, we will not be able to evaluate entire expectimax trees except for much smaller end-game situations.

Possible Approaches

- Rule-based
- Reinforcement Learning
 - Abstract each (partial) hand in play according to scoring potential
 - Learn expected hand score from simulated experience
- Expectimax variations
 - E.g. Iterative-deepening, iterative-broadening
- Monte Carlo Tree Search (MCTS)
- ???

Evaluation

- Players will be evaluated using 4 to 6 point systems. For each point system:
 - Players will have 5 minutes to process the point system, forming strategy, etc.
 - Players will then play 100 games with 30 seconds of decision time per game.
 - Total scores will be linearly scaled between 0 (min. total score) and 1 (max. total score).
- The player with the maximum sum of scaled total scores is the winner.

Supplied Code Classes

- PokerSquares – manages tournaments, individual games, game sequences, etc.
- Card – models playing card
- PokerHand – classifies full/partial Poker hand
- PokerSquaresPointSystem – generates point systems, scores hands and grids, prints grids
- PokerSquaresPlayer – player interface
- Sample PokerSquaresPlayer implementations

What's Next?

1. Share Poker Squares with potential faculty mentors and undergraduate student researchers.
2. Form teams and contact the contest organizer ([Todd Neller](#)) to get on the contest mailing list for updates.
3. Download software and resources from <http://tinyurl.com/ppokersqrs>
4. Enjoy creating and experimenting with your Parameterized Poker Squares players.
5. Submit them for evaluation towards the end of the summer 2015.
6. Optional: Write up your research results with your faculty mentor and submit your paper to EAAI one month later.
 - If accepted, present your work at EAAI in February 2016.
 - Papers will appear with EAAI papers in the AAAI proceedings.

Resources and References

- Organizer email: Todd Neller <tneller@gettysburg.edu>
- Poker Squares Page: <http://tinyurl.com/pokersqrs>
 - References
 - Rules and play grids
- Parameterized Poker Squares Page:
<http://tinyurl.com/ppokersqrs>
- Monte Carlo Tree Search (MCTS):
 - C. Browne et al. [A Survey of Monte Carlo Tree Search Methods](#)
 - L. Kocsis, C. Szepesvari. [Bandit based Monte-Carlo Planning.](#)
 - <http://www.mcts.ai/?q=mcts>
 - MCTS application to similar problem: R. Lorentz. *An MCTS Program to Play EinStein Würfelt Nicht!*