FairKalah: Fair Mancala Competition

Todd W. Neller

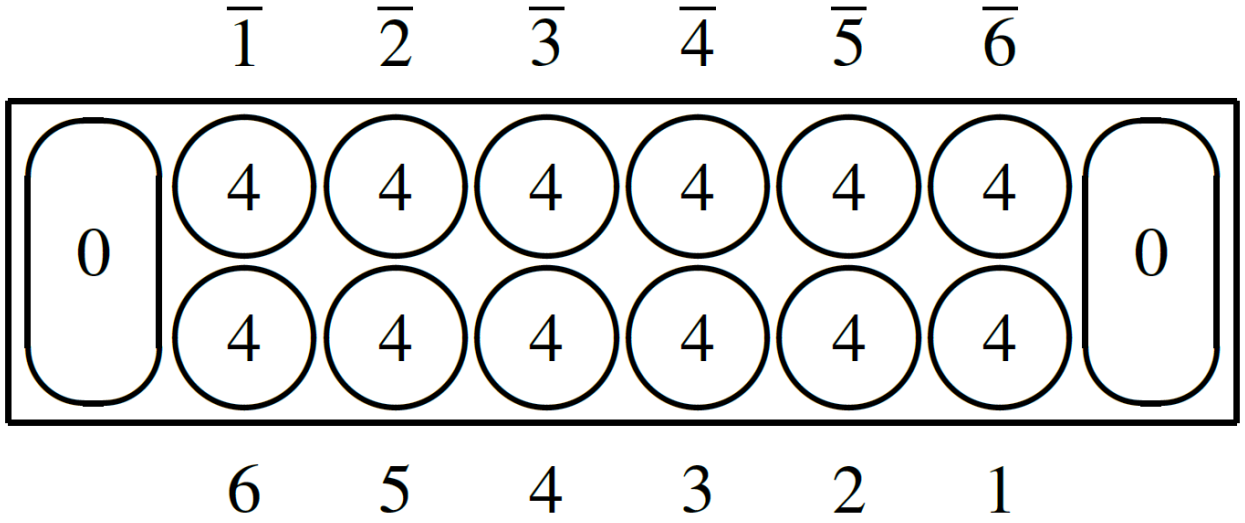
Gettysburg COLLEGE

Overview

- Java, Python, and Ludii General Game System support for fair Mancala game competition to teach:
  - Heuristic evaluation
  - Alpha-beta pruning
  - Time management for real-time constraints
- Outline:
  - Mancala rules
  - FairKalah
  - Code provided
  - Future improvements

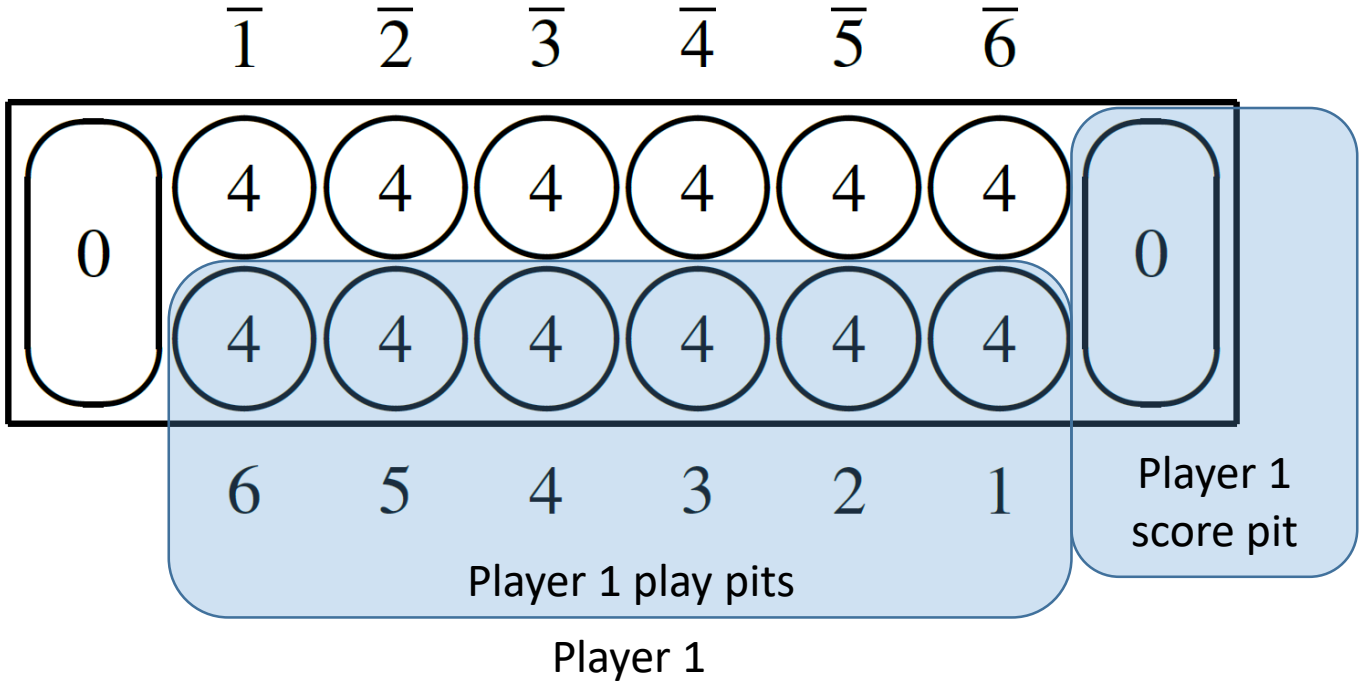
Mancala (a.k.a. Kalah) Materials

- Board with
  - 6 play pits per side for each player
  - 2 score pits, one to the right end of the board for each player
- 48 pieces initially distributed 4 per play pit in standard game



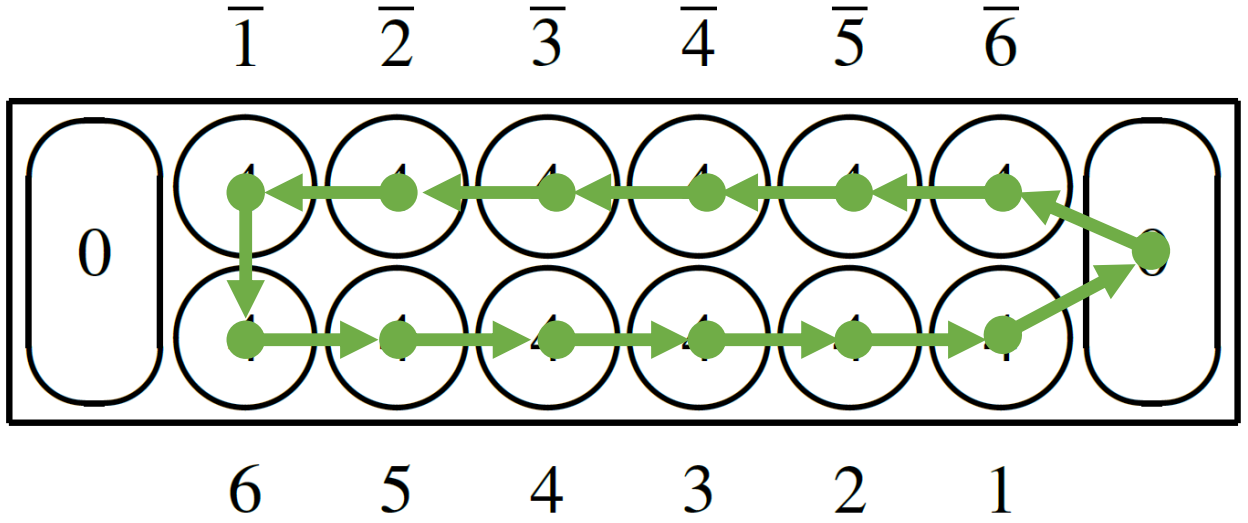
Mancala (a.k.a. Kalah) Materials

- Board with
  - 6 play pits per side for each player
  - 2 score pits, one to the right end of the board for each player
- 48 pieces initially distributed 4 per play pit in standard game



Mancala Move

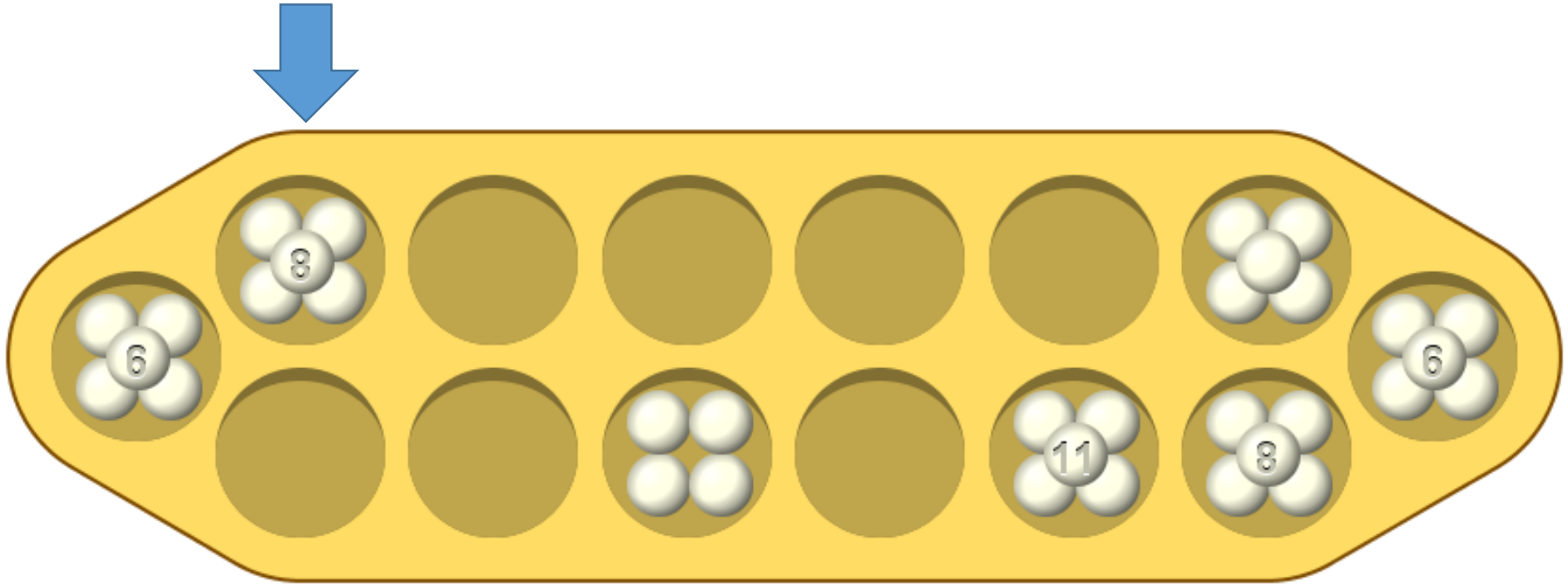
- A player's move in Mancala consists of
  - Selecting their non-empty play pit,
  - Picking up all pieces from that pit,
  - And "sowing" them counter-clockwise, one per pit, skipping the opponent's score pit.



Player 1 sowing pattern

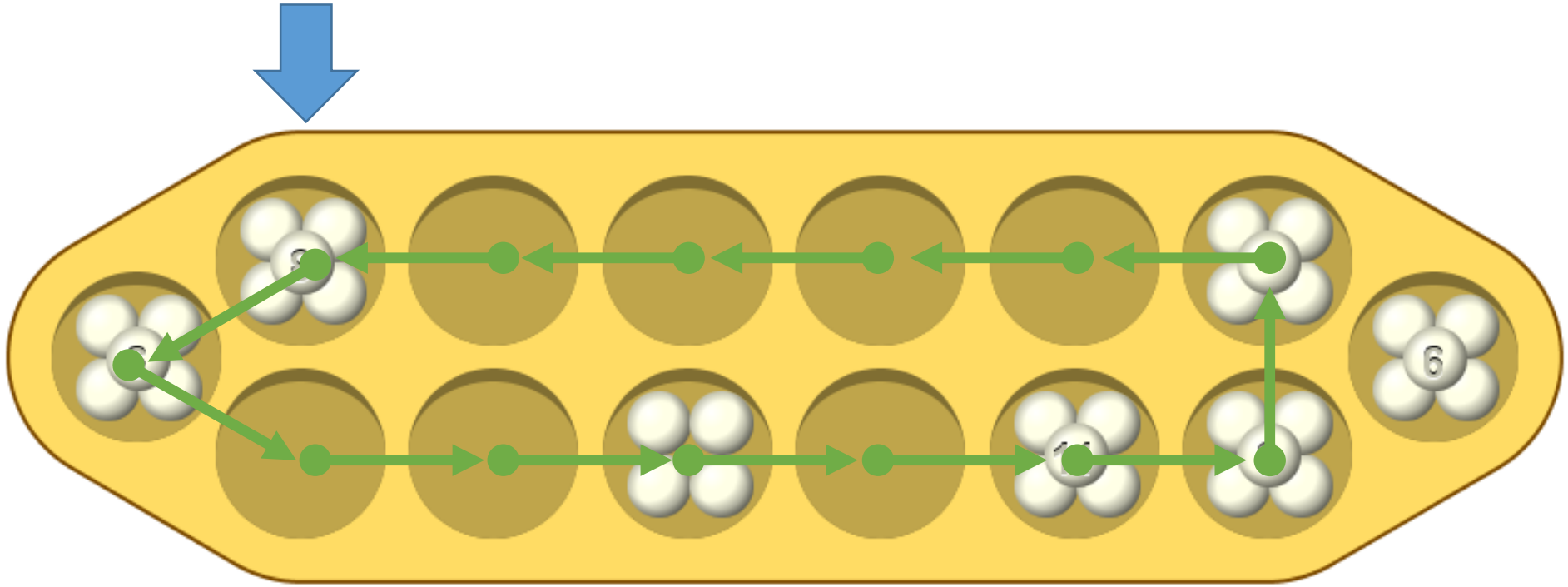
Mancala Move Example

- Player 2 (top) plays 8 pieces from upper-leftmost play pit:



Mancala Move Example

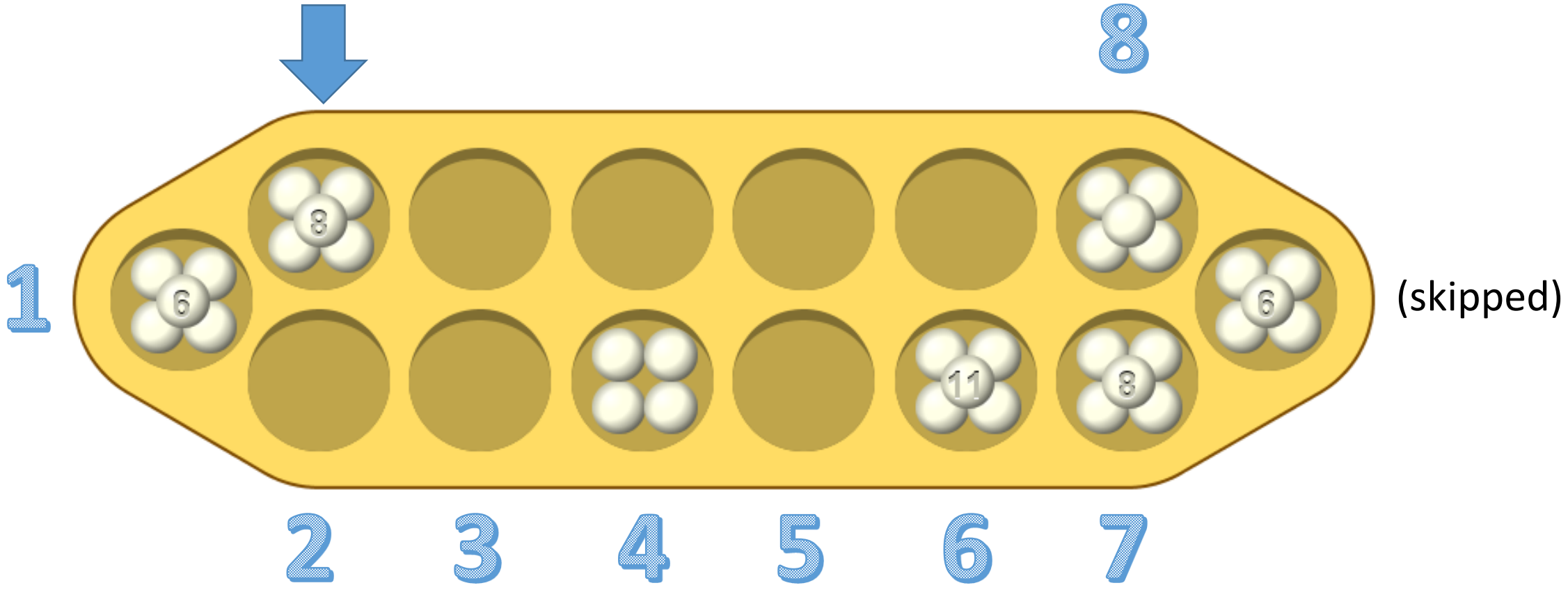
- Player 2 (top) plays 8 pieces from upper-leftmost play pit:



Player 2 sowing pattern

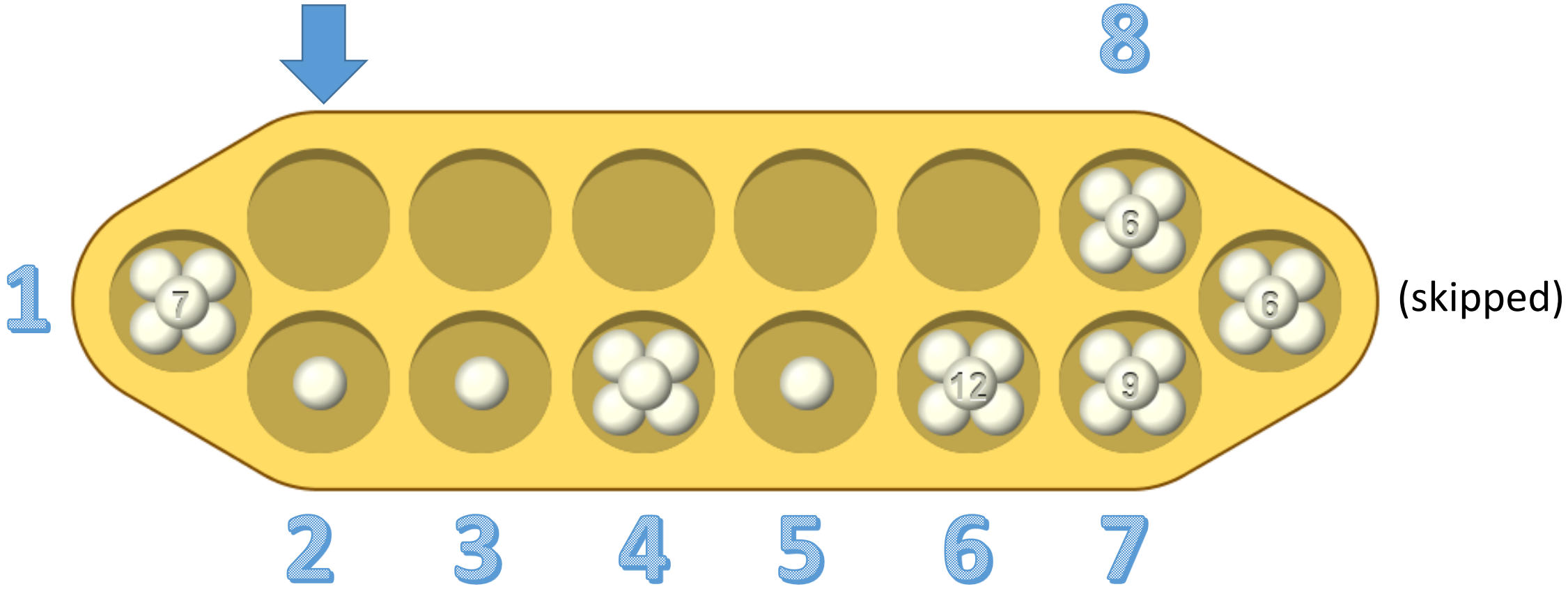
Mancala Move Example

- Player 2 (top) plays 8 pieces from upper-leftmost play pit:



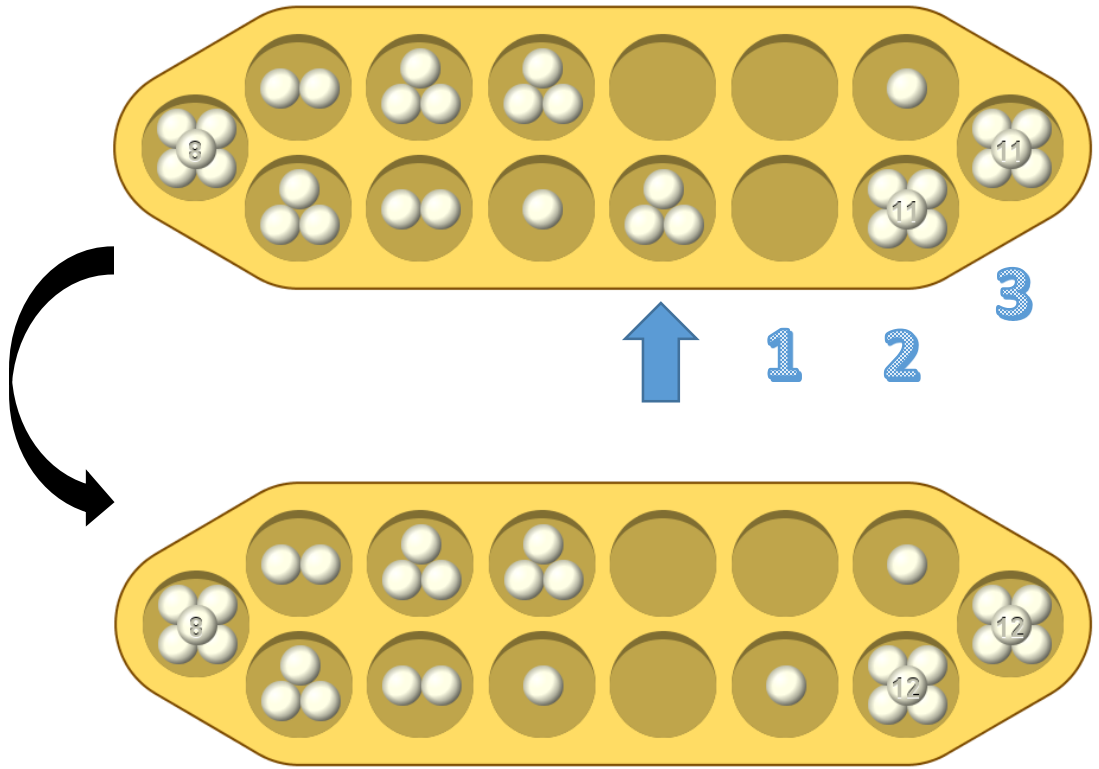
Mancala Move Example

- Player 2 (top) plays 8 pieces from upper-leftmost play pit:



Mancala Free Move

- If your last piece sown is to your score pit, take another turn.

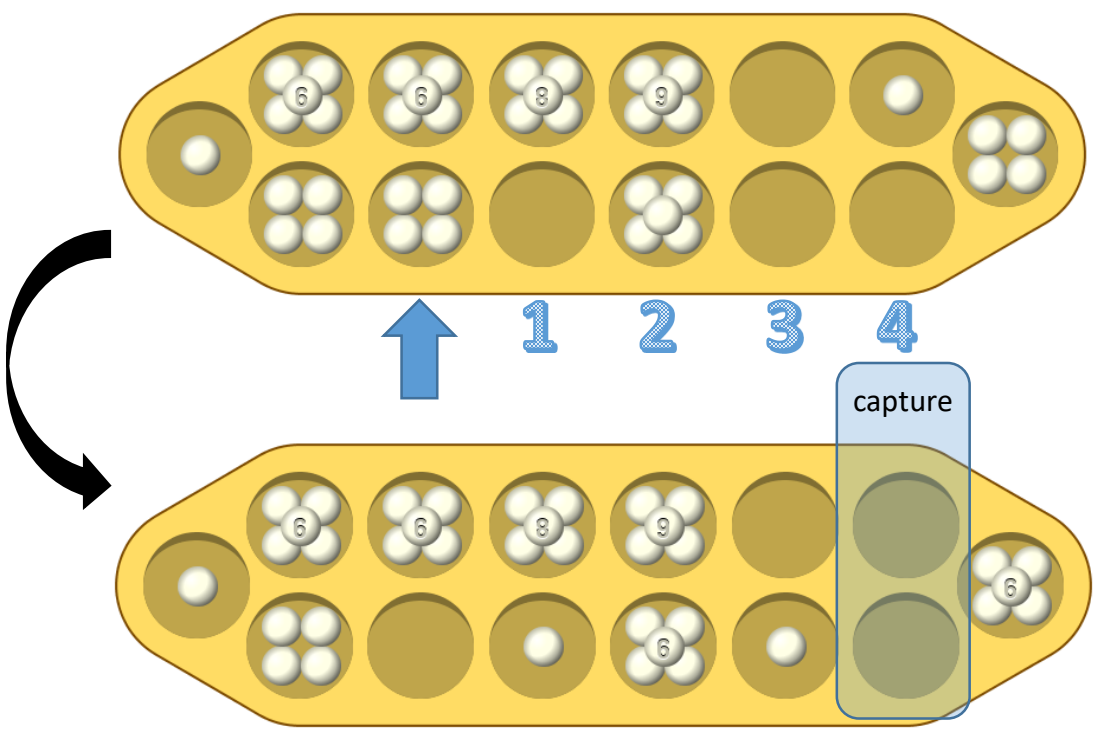


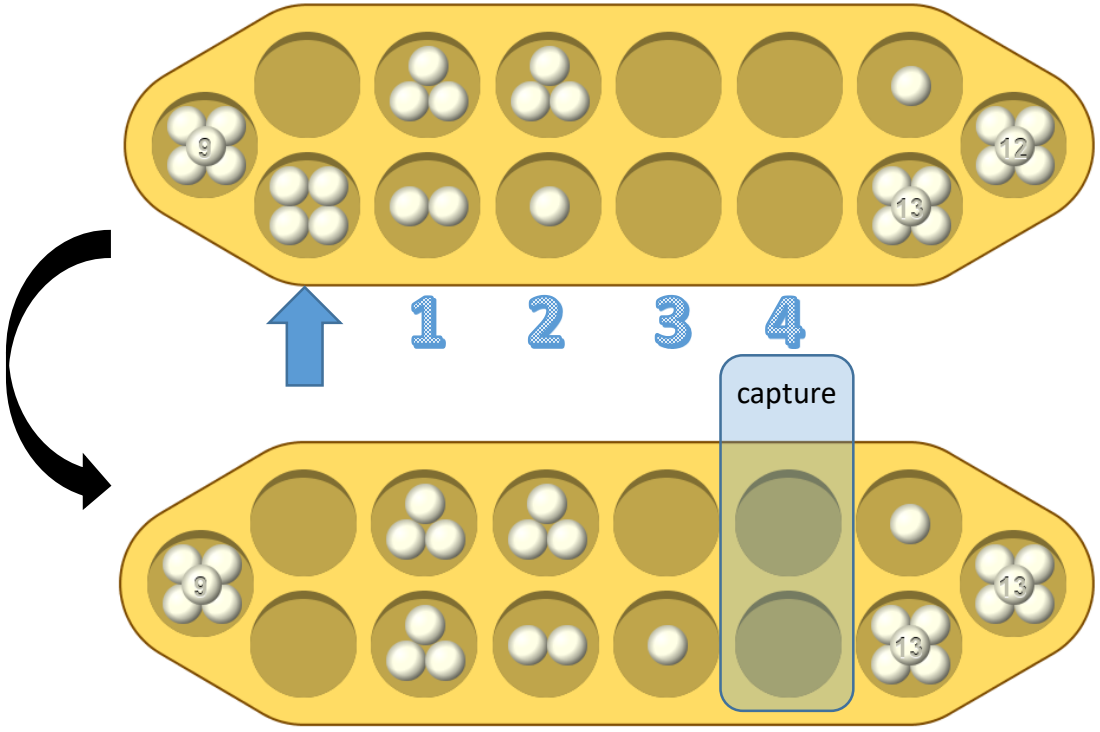
Player 1 to play

Player 1 to play again

Mancala Capture

- If your last piece sown is to an empty play pit on your side, capture that piece and any in the opponent's opposite pit (which may be empty). Captured piece(s) are placed in the player's score pit.





Mancala Game End

- “Starvation” - At the end of a turn, when no pieces remain in one player’s play pits, their opponent scores remaining play pits.
- The player that scores more pieces wins. If both players score the same number of pieces, the game is a draw (i.e. tie).

Problem: Mancala is Unfair

- The first of two perfect players will win by 10 points. (Irving, Donkers, Uiterwijk, 2000)

$m(n)$	Game value	Perfect game
4(1)	2 (W)	32-32-1
4(2)	6 (W)	3-3-230-231-1323
4(3)	8 (W)	13-1-032-2-3-0-1320-3-0-2
4(4)	2 (W)	02-2-3-0-2-21-303231-0-32-1-0
4(5)	2 (W)	0-2-1-3-0-2-1-0-32-3-021-2-0-02-2-1-3-3-313032-13031-1-2-2-3
4(6)	0 (D)	1-0-2-3-1330-01-2-2-21-201-2-23-3-23-230-32-32-12-1-0-32
5(1)	0 (D)	43-43-2-2
5(2)	0 (D)	31-32-24-0-0-31-3
5(3)	8 (W)	24-3-3-03-41-1-04342-3-3-2-2-4-414342-3
5(4)	12 (W)	12-04-03-2-20-41-1-42-32-3-2-40-0-1
5(5)	2 (W)	03-2-2-1-1-23-24-2-0-34414340-4240-2-14342
5(6)	2 (W)	2-0-0-3-3-0-2-1-440-4-42-2-121-12-3-1-4-20-0-34-24-324-1-3-3-1-41-43-2-2-3
6(1)	2 (W)	54-54-3-3-2
6(2)	10 (W)	42-42-30-0-1-1-4-5
6(3)	2 (W)	4-5-35-250-2-154-451535452-53-3-54-2
6(4)	10 (W)	25-10-3-3-5153-1-4-5-045-4-535452-53-4-1-3-2-0-54-1-3
6(5)	12 (W)	12-02-05-2-4-51-53-3-45-20-3-2-2-345-5-4-351-0-54-1-52-354-4-254-3-3

Table 9: Game values and perfect games for Kalah(m, n).

SOLVING KALAH

Geoffrey Irving[1] *Jeroen Donkers and Jos Uiterwijk*[2]

Pasadena, California Maastricht, The Netherlands

ABSTRACT

Using full-game databases and optimized tree-search algorithms, the game of Kalah is solved for several starting configurations up to 6 holes and 5 counters per hole. The main search algorithm used was iterative-deepening MTD(f). Major search enhancements were move ordering, transposition tables, futility pruning, enhanced transposition cut-off, and endgame databases.

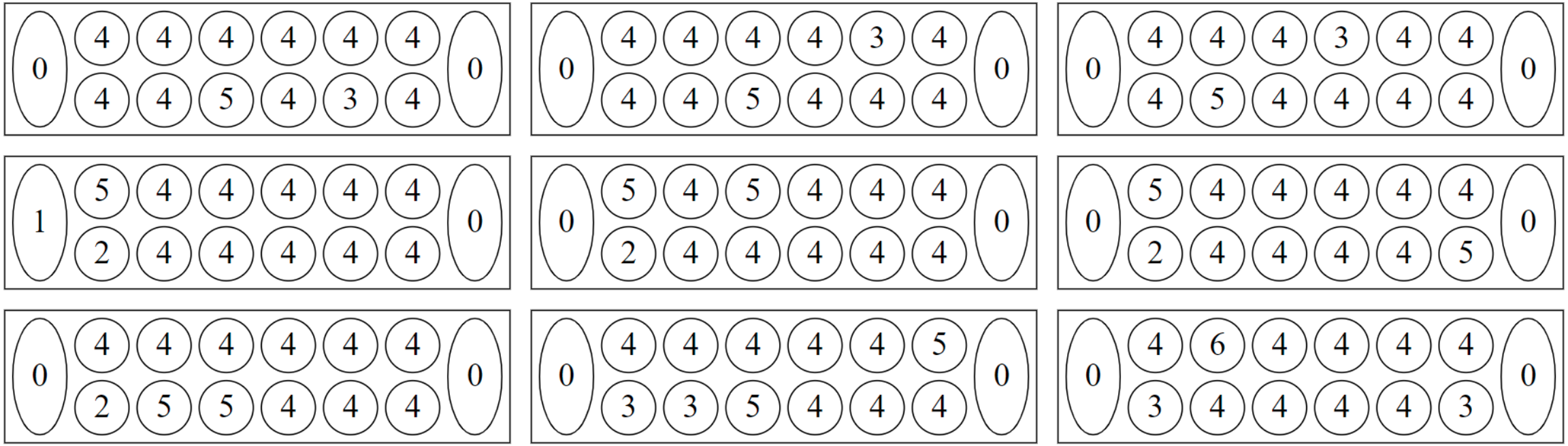
1. INTRODUCTION

Kalah is a modern, commercial variant of Mancala, introduced in the 1950s by a firm called "The Kalah Game Company" (owned by W.J. Champion). It has gained a large popularity especially in the United States and is still played in pubs and at home. In 1960, a first computerized version of the game was produced and many others followed. Remarkably, Kalah has a relatively long history in Artificial Intelligence: Bell (1968) already used Kalah to demonstrate game playing by a computer, and Slagle and Dixon (1970) used Kalah to illustrate their M & N search algorithm. Nowadays Kalah is often used as an example game in computer-science courses.

The term 'mancala' is used to indicate a large group of related games that are played almost all over the world (Murray, 1952; Russ, 2000). Mancala games (also known as 'pebble-and-pit games' or 'count-and-capture games') are played on a board that contains 2, 3 or 4 rows of holes. Sometimes these holes are simply dug in the soil or drawn on paper. Often there are two or four additional holes (called *stores*) with a special meaning. The games are usually played by two players, although one-player and three-player versions are known. Mancala games are played with a large set of equal counters. These counters can be pebbles, shells, seeds or any small round objects. The game starts with a certain distribution of the counters over the pits (usually an equal number per hole). A move is made by selecting one of the holes, lifting all counters out of it and putting back the counters one-by-one in adjacent holes in a certain direction. This is called 'sowing'. The hole in which the last counter is put determines what happens next. Sometimes a capture takes place and the turn is over, sometimes the sowing continues, and other times the player is allowed to do another move. The goal of the game is always to capture as many counters as possible.

Solution: FairKalah – fair initial board states

- We provide 254 initial states with 48 pieces arranged to be *fair*, i.e. two perfect players are proven to draw.
- This makes improvements to heuristic functions more apparent, as Mancala's unfairness obscured relative player strength.



FairKalah Project Code Provided

- Java/Python object-oriented implementations of
  - Mancala/FairKalah game tree node
  - Depth-limited minimax
  - A text-based human player interface for testing and demonstration
  - A simple, real-time player using depth-limited minimax and a score difference heuristic
  - Round-robin FairKalah tournament code that produces game transcripts and a spreadsheet summary
- Also provided: suggested readings, video presentation of rules, integration with Ludii general game system.

Project Stages

- For a two-week project, students:
  - Devise an improvement to the score difference heuristic, empirically testing performance with provided tournament code
  - Implement alpha-beta pruning, thus speeding search and allowing greater search depth limits in the same real-time limits
  - Devise improved time-management, seeing iterative-deepening as an anytime algorithm and empirically testing how to better distribute reasoning time across a game.
- For a longer (e.g. term) project, students can learn improved heuristics from play data.

Coming Soon: Optimal Play Dataset

- When a current research project is concluded, the Model AI Assignments page for FairKalah will include access to an optimal play dataset with:
  - Input state description (pieces per pit)
  - Output game value of state
  - Output Boolean indication of optimal move(s)
- From these, students can apply machine learning techniques to build better heuristics.

Conclusion

- This assignment has been refined since the fall of 2000 and has long been a student favorite.
- With fair play for the most recent iterations, students have had much greater success in discerning heuristic evaluation improvements.
- Although optimal play is known and computable, applying both tighter time and memory limits will preserve this assignment's relevance.
- Enjoy!

Questions?