Pedagogical Possibilities for the 2048 Puzzle Game

Todd W. Neller

Gettysburg College
Computer Science
Outline

• Introduction to the 2048 Game
• Sample 2048 Assignments
• The Importance of Sharing Assignments
• Wanted: High Fun/SLOC Assignments
What is 2048?

• 4-by-4 square grid partially filled with tiles labeled with powers of 2.
• Primary goal: merge tiles in order to create a tile labeled $2^{11}$.
• Secondary goal: achieve higher scores and higher tiles.
• Initial board has 2 randomly generated tiles.
  – Uniformly random position
  – Values 2 (with probability .9) or 4.
2048 Turn

- Move left, right, up, or down, if it results in a *change* to the grid.
  - Change can be from tile sliding, and/or
  - merging same number tiles by sliding them together along a row/column line.
- No legal move $\rightarrow$ game over.
- Player chooses a direction along a line:
  - All tiles merge and slide in that direction if possible.
  - From the front to the back, successive identical tile pairs merge into a new tile with their sum.
  - (Merged tiles cannot merge again in the same movement.)
  - All tiles slide as far as possible in the direction of motion.
Downward Move Example
How can 2048 be useful to CS Ed?

- CS1, CS2: Modeling
  - 1D, 2D array, iteration exercise
  - Queue exercise
- CS1/CS2: MC Simulation
- AI
  - Heuristic evaluation
  - Search (Expectimax, etc.)
Modeling

- Given a 4-by-4 puzzle grid,
  - Is the game over?
  - Can the player legally move in a given direction?
  - What are the legal directions of motion?

- Given a single line of tiles, compute the result of a move towards the front of that line.

- Given a 4-by-4 puzzle grid,
  - Compute the result of a move in a given direction.
Heuristic Evaluation

• Score one’s preference for a given board.
• For non-AI students, the instructor can supply search code using student heuristics.
• How much should one value:
  – ... having our maximum value tile in a corner?
  – ... monotonicity along a snake-like path?
  – ... monotonicity along any line?
  – ... the maximum tile of a line along an edge?
  – ... adjacent identical tiles?
  – ... empty cells?
• Such open-ended assignments encourage creativity and deeper problem-solving engagement.

[Image of a 4x4 grid with numbers and arrows indicating paths]
Simple Game Play

- Greedy – Pick the move that immediately (before random tile generation) yields the grid with the maximum heuristic evaluation

- Monte Carlo:
  - After each possible play,
    - For a fixed number of trials (e.g. 1000):
      - Simulate \( n \) (e.g. 2) moves (after \( n \) random tile generations) and accumulate the heuristic evaluation of the final resulting state.
      - Divide by the number of trials to get an avg. heuristic value.
  - Choose move with the maximum avg. heuristic value.
Advanced Game Play

• Choice node and chance nodes?
  – \(\rightarrow\) Expectimax
  – Maximize at choice nodes, average at chance nodes
  – Evaluate heuristic at depth cutoff
  – Chance sampling: Average over a \textit{sample} rather than all possible chance events

• Monte Carlo Tree Search
## Performance Results

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Median Score</th>
<th>Mean Score</th>
<th>Std. Dev.</th>
<th>Max. Tile</th>
<th>Median Tile</th>
</tr>
</thead>
<tbody>
<tr>
<td>random</td>
<td>1028</td>
<td>1075</td>
<td>512</td>
<td>256</td>
<td>128</td>
</tr>
<tr>
<td>simple greedy</td>
<td>3326</td>
<td>3620</td>
<td>1708</td>
<td>1024</td>
<td>256</td>
</tr>
<tr>
<td>simple Monte Carlo (depth 3)</td>
<td>14586</td>
<td>14295</td>
<td>6592</td>
<td>2048</td>
<td>1024</td>
</tr>
<tr>
<td>expectimax (depth 2)</td>
<td>14398</td>
<td>14241</td>
<td>6717</td>
<td>2048</td>
<td>1024</td>
</tr>
<tr>
<td>expectimax (depth 3)</td>
<td>27132</td>
<td>25270</td>
<td>10429</td>
<td>4096</td>
<td>2048</td>
</tr>
<tr>
<td>10-sample expectimax (depth 2)</td>
<td>14496</td>
<td>14623</td>
<td>7096</td>
<td>4096</td>
<td>1024</td>
</tr>
<tr>
<td>10-sample expectimax (depth 3)</td>
<td>27052</td>
<td>24883</td>
<td>10106</td>
<td>4096</td>
<td>2048</td>
</tr>
</tbody>
</table>

- Simple Monte Carlo can achieve the 2048 tile.
- Chance sampling provides significant speedup with comparable performance.
Why is Nifty Nifty?

• Nifty Assignments sessions (http://nifty.stanford.edu) and spin-offs, e.g. Model AI Assignments (http://modelai.gettysburg.edu)
• Experiential learning is deep learning
• Model, well-designed, applied assignments form the foundation of experiential learning.
• Model assignments take significant time, effort, expertise, and revision to develop.
• Sharing such assignments through repositories aids the efficient advancement of CS education.
What Not to Assign

• Motivating question #1: Why isn’t a Chess game a good intro CS assignment?
  – Complex rules (e.g. moves, en passant, castling) are often a distraction from the lesson.

• Motivating question #2: Why isn’t Tic-tac-toe my favorite intro CS assignment?
  – Tic-tac-toe is boring once you know the strategy.
Wanted: High Fun/SLOC Assignments

• The best game/puzzle assignments:
  – Require few source-lines-of-code to model (simple)
  – Are fun and interesting, offering an experience worth revisiting (satisfaction) and sharing (promotion)

• Examples: Breakthrough, Connect6, Pig, Nim Games (Subtraction, 3-Pile Nim, Chomp), 2048

• I would be interested in your best examples as well!
Conclusion

• 2048 offers many good exercises for CS1 through advanced AI.
  – Modeling assignments are suitable for CS1.
  – Simple AI yields satisfying results.

• Sharing exercises for deep, experiential learning is worth our effort.

• Wanted: High Fun/SLOC Assignments!