

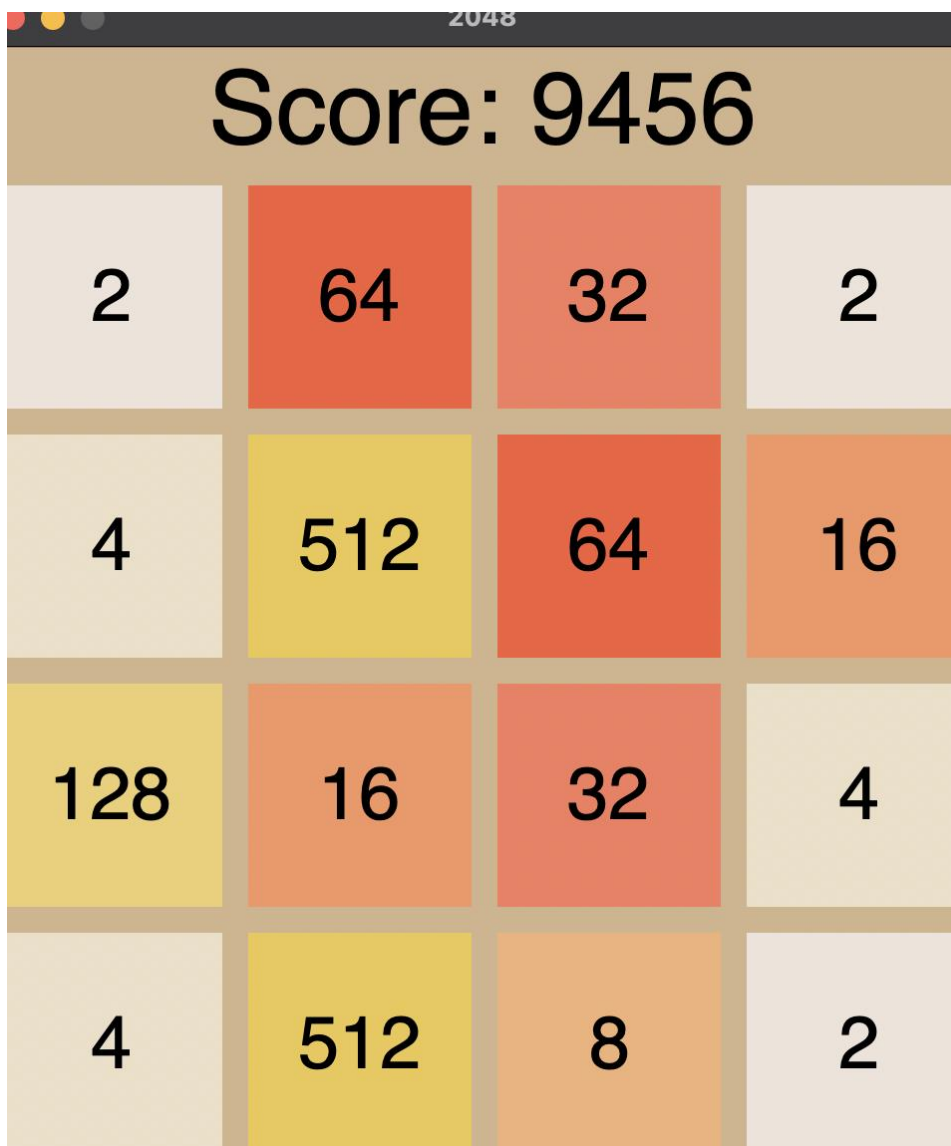
2048 Game

Saranya Roy

1st Version :

In this version, I implemented a basic MinMax agent for the 2048 game, utilizing a heuristic based on the game score .

Testing result :



A screenshot of a 2048 game interface. At the top, the title '2048' is visible. Below it, the score 'Score: 9456' is displayed in a large, bold, black font. The game board is a 4x4 grid of tiles. Each tile contains a number representing its value. The tiles are colored based on their value: 2 (light beige), 4 (light tan), 8 (light orange), 16 (orange), 32 (red-orange), 64 (red), 128 (yellow-orange), 256 (yellow), 512 (gold), and 1024 (dark gold). The grid is as follows:

Row	Col 1	Col 2	Col 3	Col 4
1	2	64	32	2
2	4	512	64	16
3	128	16	32	4
4	4	512	8	2

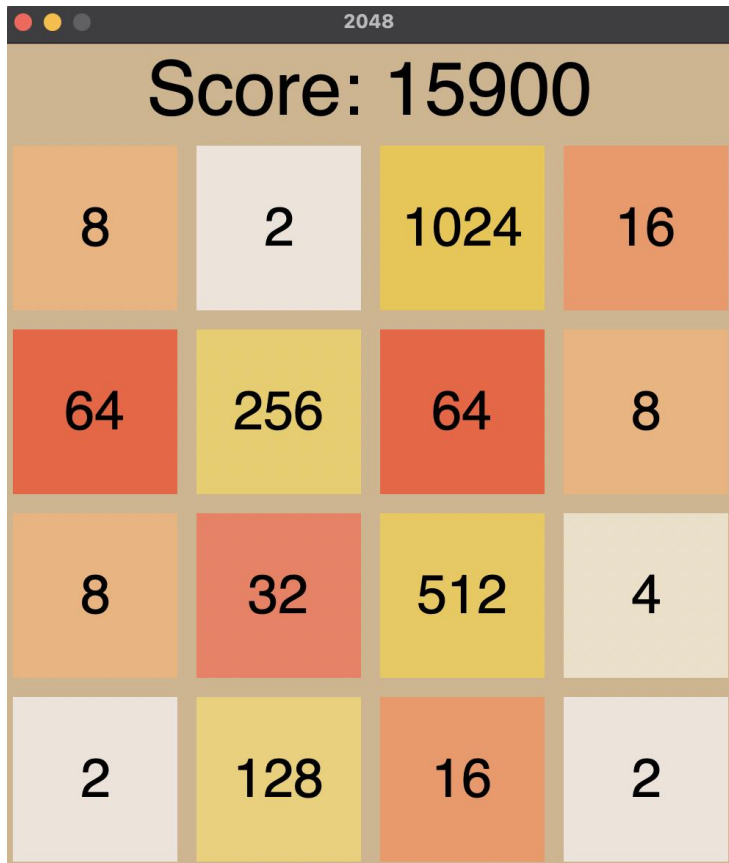
- Final Score: 6040
- Time-limit:0.1
- Search depth: 10

Observations :

- 1) The current MinMax agent successfully avoids invalid moves and performs reasonably well.
- 2) It primarily favors right and down moves.
- 3) The scoring is inconsistent, typically ending around 2000–4000 points.

2nd Version : MinMax with improved heuristic

Testing results:



2048

Score: 11972			
4	2	8	2
64	256	16	64
2	1024	128	4
8	32	8	2

Maximum Score: 15900

- Average Score: 8000

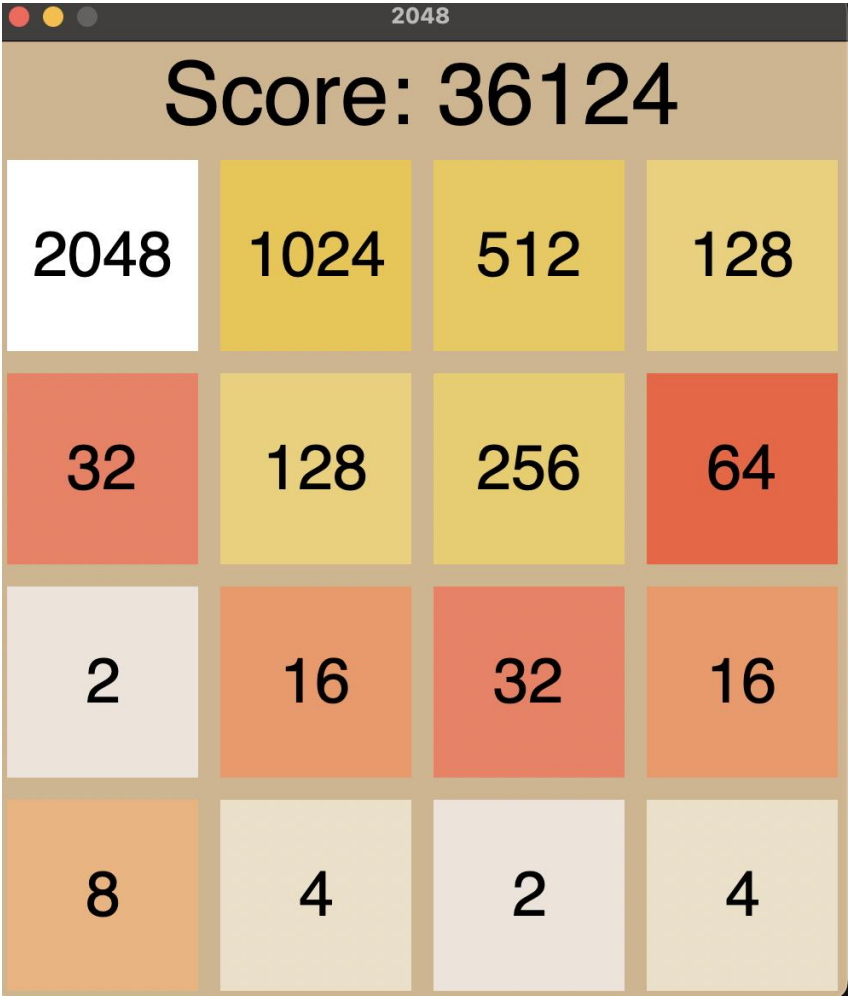
Observations:

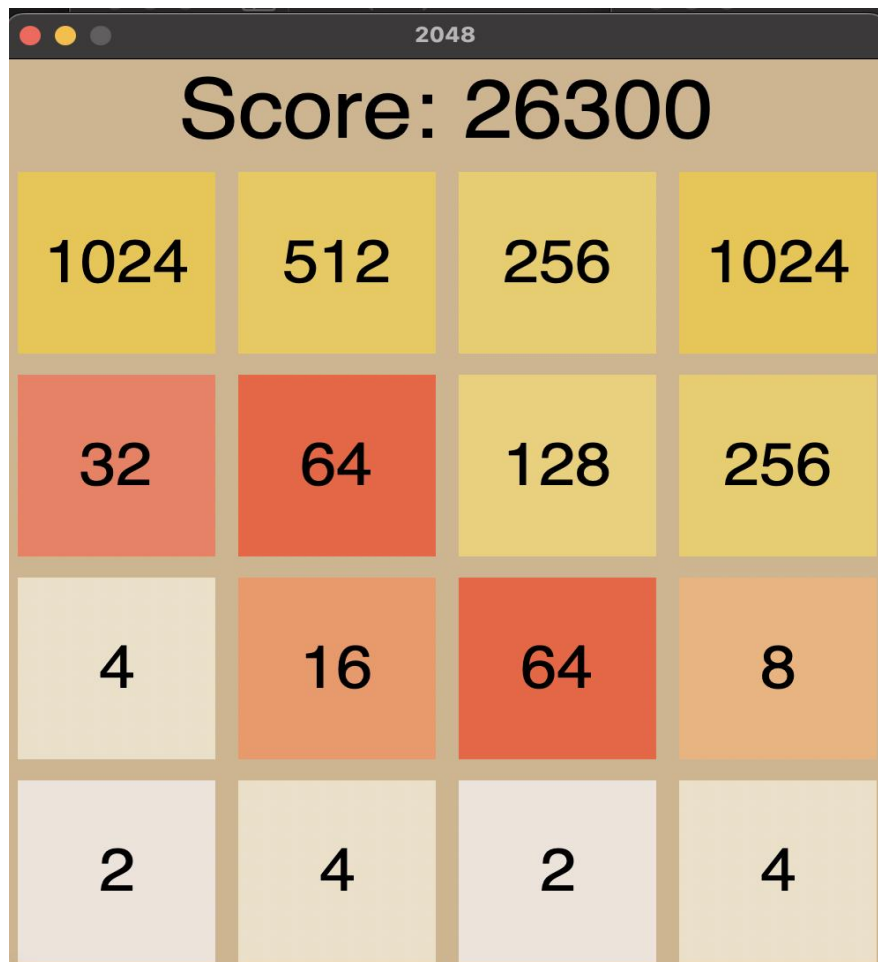
- The board consistently maintains a higher number of empty tiles.
- Overall, the score is more stable and reliable compared to version 1.

Final Version : **Expectimax with monotonic and smoothing heuristic**

In this version I used Expectimax instead of Minimax, I used :

Testing results:





Maximum Score: 58298

Average Score: 26000

Observations :

- 1) Expectimax gave better scores than MinMax.
- 2) This version was significantly more stable producing better median scores.

- 3) In this version, the highest tile consistently stayed in a corner — a key strategy in 2048.

Conclusion:

Across all three versions, my agent showed steady improvement over time.

- **Version 1** used a basic score-based approach for decision-making, but it didn't account for tile positioning, leading to subpar performance.
- **Version 2** introduced a smarter strategy by attempting to keep the highest tile in a corner, which resulted in more effective moves.
- **Version 3** marked the most significant improvement. By implementing Expectimax and incorporating advanced heuristics—such as maintaining board smoothness, anchoring the max tile in a corner, and using a snake-pattern layout—the agent achieved higher and more consistent scores. This version proved to be the most stable and successful overall.