2048 Game

Manasa Tallapaka

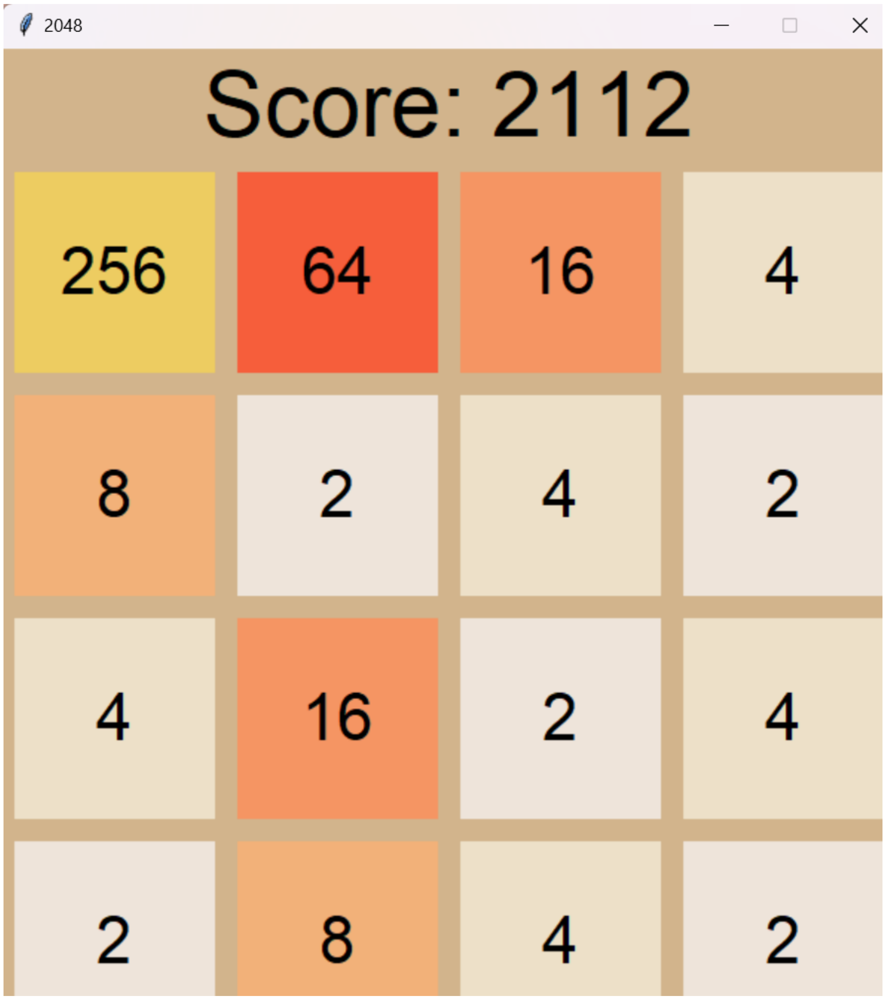
During testing and evaluation of MyAgent.py, I found that the **min-max strategy consistently outperforms the greedy approach** in score. This is expected due to the fundamental difference in how the two strategies evaluate future moves.

The **greedy agent** evaluates only the immediate reward (i.e., the score gained from the current move) and makes no consideration for the board's future state. As a result, it often makes locally optimal but globally harmful decisions, such as combining tiles too early or pushing valuable tiles into difficult corners.

In contrast, the **min-max agent**, although it does not model randomness correctly (it treats new tiles as adversarial rather than stochastic), performs **iterative deepening** and evaluates deeper branches of the game tree. This allows it to **choose actions that maintain flexibility** in the board and preserve high-value tile positions. It may not always choose the highest scoring move immediately, but it leads to better long-term results.

Attempt1:

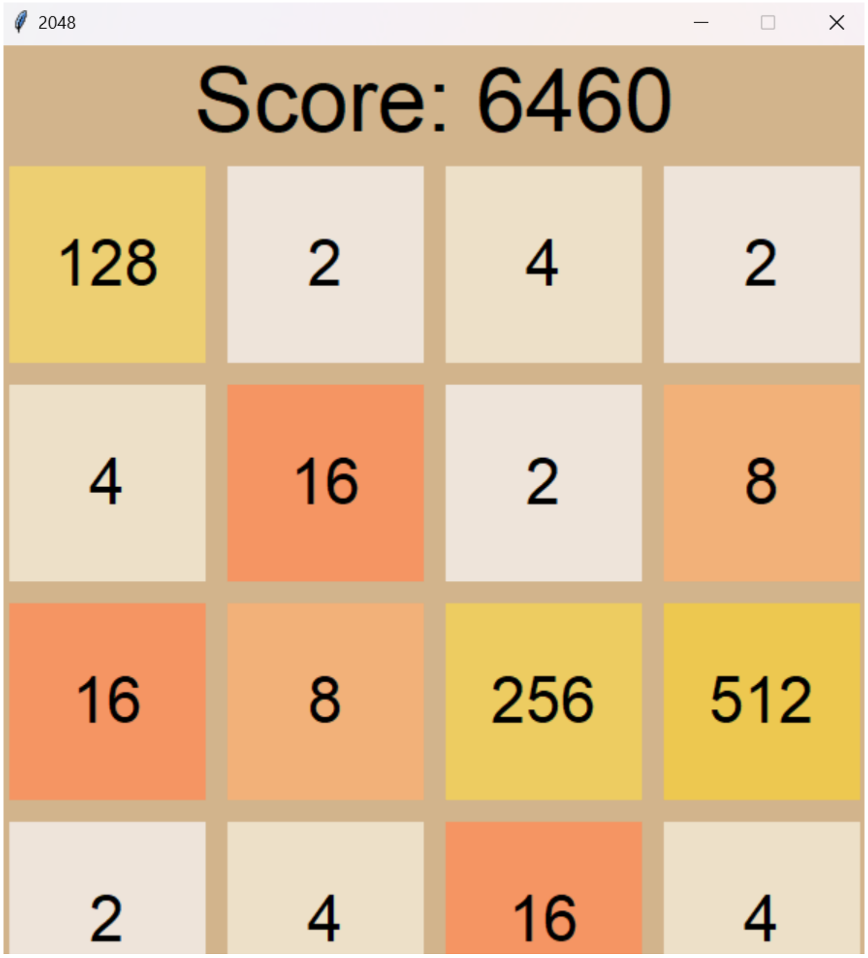
python Play.py MyAgent 1.5 -g 600



Attempt2:

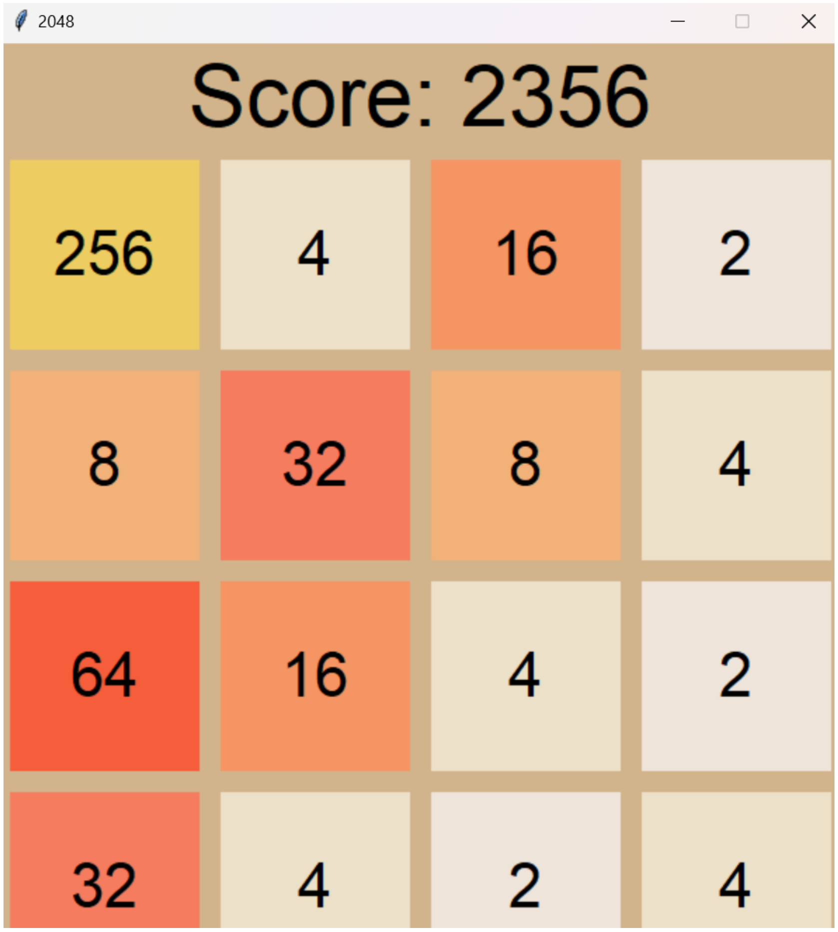
python Play.py MyAgent 1.5 -g 500

python Play.py MinMax 2.0 -g 600



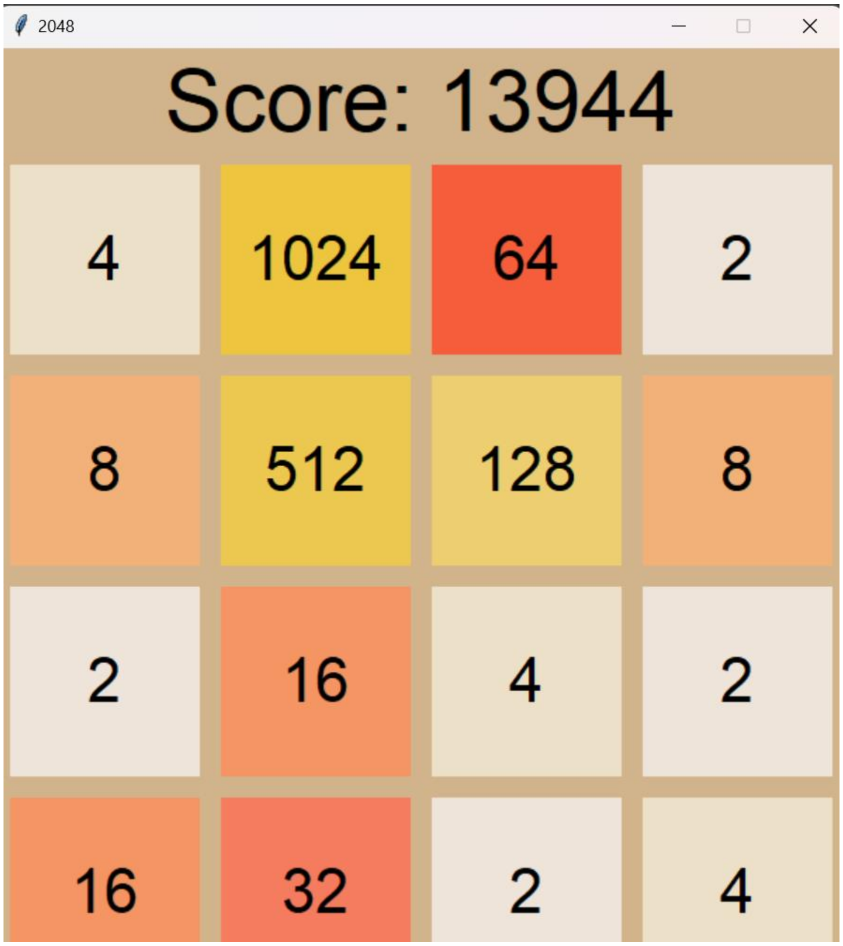
Greedy

python Play.py Greedy.py 2.0 -g 600



python Play.py MinMax 2.0 -g 600


2048
Score: 11112
8
2
8
2
32
1024
16
4
8
32
256
8
4
2
64
4


2048
Score: 13944
4
1024
64
2
8
512
128
8
2
16
4
2
16
32
2
4