This project implemented the game Reversi in Java with the following three game modes:

1. Human vs Pure Monte Carlo search tree AI.

2. Human vs Heuristic search AI.

3. Pure Monte Carlo search tree AI vs Heuristic search AI.

**­­Pure Monte Carlo Tree search AI:**

This choice of AI is implemented with 10,000 random playouts for each valid move. The result of each random playouts is recorded and used to decided how promising that move is. The value of each move is computed by the formula:

the number of wins + the number of draws - the number of losses in those random playouts.

The efficiency of the Pure Monte Carlo Tree search is very promising as it takes around a second to decide a move. This means the program is able to complete at least several tens of thousands of random playouts in one second.

**Heuristic search AI:**

This is a combination of positional strategies and absolute disks. About the first 40 moves after the game starts, the heuristic is to bias a location based on a weight associated with each location. And when the game is approaching the last 25 moves, this search bias toward the locations that leads to the maximum flips. This heuristic is also very efficient as it does involve massive computations at all. A move is always calculated within a second.