Artificial Intelligence (CS 4253/6613)
Project: Game Tree Search (Minimax, α-β, Monte Carlo)

In this project, you will modify the provided Python codebase to implement game tree search procedures. In particular, you will be implementing the minimax, α-β, and Monte Carlo tree search algorithms, described in Figures 5.3, 5.7, and 5.10, respectively, in the Russell & Norvig textbook (Fourth Edition).

The primary domain of application of game tree search is a discretized and stylized, completely observable two-person soccer game. Consult the README file in the enclosed zipped repository for details of the game, including the actions available from any state. The game ends when either player scores a goal and wins or if a particular state, given by the positions of the two players and the ball on the field, is repeated, which corresponds to a draw.

You should play the game interactively to have a better understanding of the soccer domain, which should assist you in developing a competitive evaluation function. You should experiment with varying search depths (minimax) and the maximum runouts or time limits (Monte Carlo) and with simple versus more sophisticated evaluation functions to form an understanding of the relative advantages of deeper domain knowledge and the ability to search further ahead in the game.

In your report, you need to explain your rationale for constructing your evaluation function.

You should also compare the performance of your system with the provided agent, both going first and second. The lookahead for the minimax game-tree search should be limited to 5 moves.

Include an analysis of how α-β pruning helps reduce search over minimax in a game tree for this particular domain. Also elaborate on your experience in designing a good evaluation function: tradeoffs between deep search and sophisticated evaluation functions.

Your grade break-up for the project is as follows:

Minimax	20%
α-β	20%
Monte Carlo	20%
Board Evaluation	20%
Analysis and write-up	20%
Total	100%