General Game Playing AI

Scenario

General game playing is a field of artificial intelligence where a game playing agent is programmed to play any type of game. The agent is not aware of which game it will play until run-time, when it is provided the rules to the game, so there is no way to generate a specific algorithm for your player. (For those interested in this topic, there is a short course on Coursera -- https://www.coursera.org/course/ggp). The basic concept is to generalize the idea of a game, and to create an algorithm for determining the best move for a given type of game based on some criteria, such as timers, game complexity, etc. Note that different algorithms may be suitable for different situations (such as move selection time, board size, etc.)

Problem

Your task is to generate a general game player class, which can easily select a method of playing based on the context of the game. Your game player should be able to play a game using one of three different types of approached: *TreeSearch, Minimax,* and *MonteCarlo*. The game player should implement two public methods: *setup* and *selectMove*. The playing approach to use should be selected based on the following criteria: If the board is small (i.e., size is less than 20), then *TreeSearch* should be used. If the board is large (i.e., size is greater than or equal to 20) and the moveSelectionTime large (i.e., greater than or equal to 60), then *Minimax* should be used. Finally, if board size is large and moveSelectionTime is small, then *MonteCarlo* should be used.

A skeleton class for *GamePlayer* is provided, with the method signatures for the two methods described above. You may add attributes / methods to this class as needed, but the Game will interact with the GamePlayer through these two methods. Your solution should involve creating a set of new classes, each implementing a game playing approach. Your solution should be easily extendible (i.e., adding a new playing technique involves creating a new class). An incorrect solution involves creating a new method in *GamePlayer* for each play approach approach.

Deliverables

1. Identify the design pattern you used to solve this problem, and the participants (i.e., the roles each class takes).
2. An implementation in a language of your choice.
3. A class diagram of your solution (including existing classes), so future developers can easily see how to work with your solution.