Rock, Paper, Scissors

You have been asked to create a simple application to process a game of rock, paper, scissors. The company that has asked you to do this has great plans for the game's algorithm: they are convinced they can use it in all kinds of applications for conflict resolution. As a result, your minimum viable product (MVP) is allowed to be a simple version of the game, but its architecture should be designed in such a way that it can later on easily support a number of extra gameplay additions.

Game Rules

A match takes place between 2 players and is made up of 3 games, with the overall winner being the first player to win 2 games (i.e. best of 3).

Each game consists of both players selecting one of Rock, Paper or Scissors; the game winner is determined based on the following rules:

- Rock beats scissors
- Scissors beats paper
- Paper beats rock

Requirements

Your application must support 3 types of players:

- Human Player: this player is of course a human being and the application requires this player to select from Rock, Paper, and Scissors each turn. User input is therefore required.

- Random Computer Player: this player is controlled by the computer and uses a very simple algorithm to select from Rock, Paper, and Scissors each turn: it must simply randomly select one of the options.

- Tactical Computer Player: this player is controlled by the computer but uses a slightly more complex algorithm. Its first choice must be random, its follow-up choices during a match must always be the option that would have beaten its previous choice. For example, if it played Rock in game 1, it should play Paper in game 2, and Scissors in game 3.

Because there is a human player, a basic user interface is required in order to allow this player to start games, and to select from Rock, Paper, Scissors each turn. How you implement this user interface is completely up to your own preferences. A simple console application is fine.

Extensions

In your application's architecture, you must account for the following additions later on:

- New computer player strategies. As we learn more about the game, we may add new types of computer player strategies.
- Longer matches. We may want to be able to support a "best of 5" or "best of 7" match format.
- Additional moves. We may want to be able to add additional moves to the game, like "Lizard", or "Spock". In these cases choices may also beat multiple other choices: e.g. Spock beats "Rock" and "Scissors"