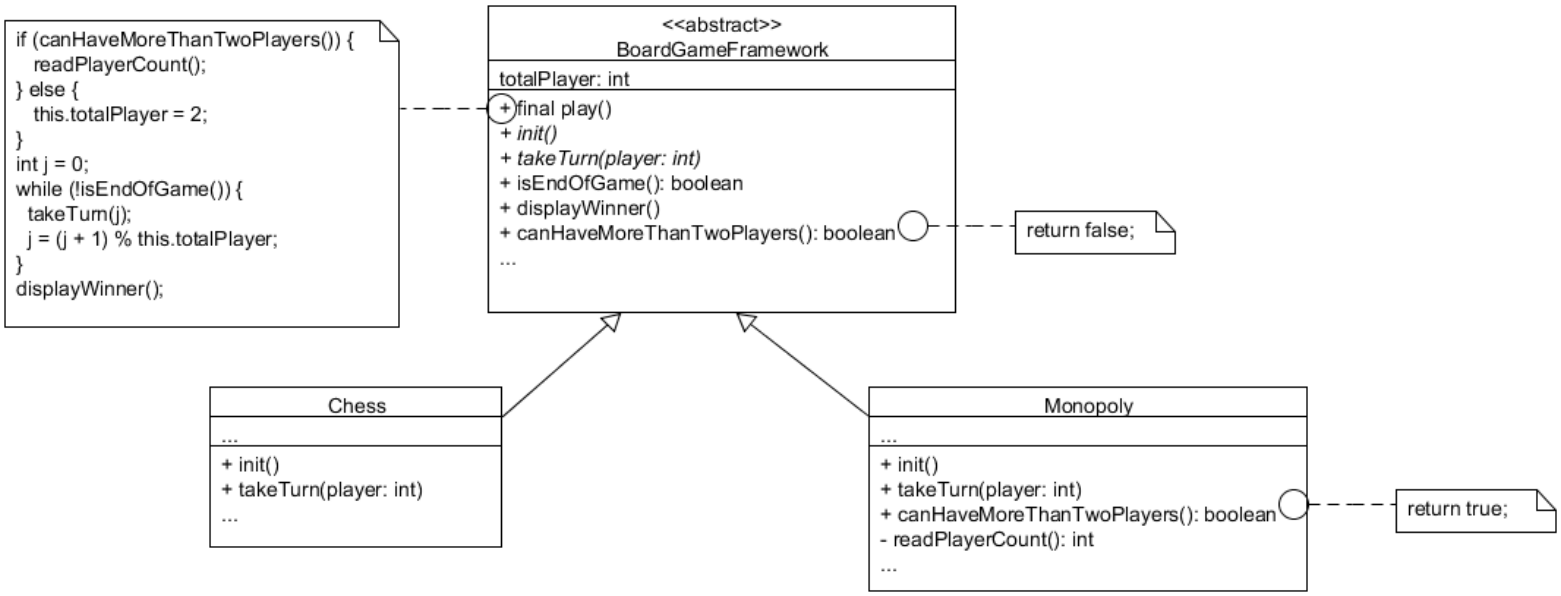
1.

The Monopoly and Chess games have a lot of code duplication and some of the methods are slightly different. The Monopoly methods and their corresponding Chess methods are listed below.

Monopoly Method	Chess Method
init()	initializeBoard()
makePlay(int player)	takeTurn(int player)
boolean gameEnded()	boolean isEndOfGame()
showWinner()	displayWinner()
playMonopoly(int totalPlayer)	play()

This results in a lot of code duplication and makes the system harder to maintain/update. For example, trying to implement a new board game would be pretty hard to do since you have to write several lines of code for initializing the board, showing who won, playing the game, taking turns, etc.

2.



3.

Each board game extends the abstract BoardGameFramework. As you can see, play() is the template method and canHaveMoreThanTwoPlayers() is the hook. By default, canHaveMoreThanTwoPlayers() returns false so this.totalPlayer is set to 2. Board games that can have 2+ players will prompt the user to input how many players there will be. This value is then set as this.totalPlayer. This design gets rid of a lot of the code duplication from before while also relying on the subclasses only when necessary, implementing the Hollywood Principle. It also makes it much easier to implement a new board game

because the skeleton of the game is already in place. The design patterns I implemented were the template method including hooks as well as the Hollywood Principle.