

Collect and display the player names

```
public void setPlayerName(String name) {  
    //sets instance variable to parameter variable  
}
```

```
public String getPlayerName(String name) {  
    //returns parameter name  
}
```

Determine who moves first and gets assigned the dark pieces

```
public Player firstMove(Player player) {  
    //chooses a player to move first  
}
```

Display a board with pieces the user can interact with if one or both players are human

```
public Checkerboard build() {  
    //builds a checkerboard with pieces  
}
```

Prevent illegal moves

```
If (/* move is legal */) {  
    //move player  
    //switch turns  
} else if (/* move is not legal */) {  
    //don't move anything  
    //display error message  
    //let the player choose another move  
}
```

Identify when there is a winner, loser, or if a draw has occurred

```
public void gameOver()
```

```
If (/* all of one player's pieces are gone */) {  
    //call gameOver() function  
}
```

Determine what moves are possible/allowable

```
If (/* there is open space diagonally */) {  
    //set boolean variable to true  
}
```

```

If (/* if there is an opponent piece diagonally */) {
    If (/* there is no wall or other piece adjacent to it going in the same direction */)
        //set boolean variable to true
    } else {
        //set boolean variable to false
    }
}

```

Determine if a move is illegal

```

public void illegalMove()

```

```

If (/* a wall is in the way */) {
    //call illegalMove()
    //don't move anything
    //don't switch turns
}
If (/* another piece is in the way */) {
    //pcall illegalMove()
    //don't move anything
    //don't switch turns
}

```

Determine a winner, loser, or a draw condition

```

If (/* one team doesn't have any pieces left */) {
    //call gameOver()
    //winning message
}

If (/* if no more moves can be made */) {
    //call gameOver()
    //draw message
}

```

Implement the algorithms (outside of the model) for the computer (AI) to determine moves

```

If (/* move is legal */) {
    If (/* opponent can be jumped over */) {
        //jump over opponent
    } else if (/* legal move can be performed */) {
        //move piece
    }
}

```

```

        //switch turns
    }
} else if (/* move is illegal */) {
    //call illegalMove()
}

```

Implement a UI representation of the game and game play

Dark Player



Light Player