# Design Backend Classes to support a chess game

## Development Scenario

Your team is developing a chess game that can be played between two players on a single computer. This is a desktop application. Your colleague is writing the UI and you have been asked to write the backend classes. You have to write the classes and test that the functionality is working. However, you don’t need to write any front end code.

## Objectives of the Exercise

Write Backend classes to support playing the chess game. We will evaluate your ability to create an object oriented design given a problem space.

Following are the actions a user can perform and the related actions the UI application will need to do. It’s up to you to tell your colleague what class to create and which functions to call.

## Detailed Use Cases

|  |  |  |  |
| --- | --- | --- | --- |
| # | Action | UI response | Backend responsibility |
| 1 | User starts the application | Create a class and/or call initialization functions | Initializes the game and anything else needed |
| 2 | User makes a move | Call a function on a class with the details of the move like coin/location on board/user | 1. Validate that the piece being moved belongs to the user that is supposed to move next 2. Validates that the move is allowed for the game piece. In this exercise, we will only implement validation for a bishop. A bishop is only allowed to move diagonally 3. Validate that the path is clear unless this is a knight (horse). For example, if there is a pawn (Soldier) in the way, bishop can’t move ahead of that |

Notes:

1. Further checks like who won, killing another piece, etc. do not need to be implemented
2. The game does not have any database and doesn’t store anything