

## Carrom board ( multiplayer ) Application

```
// calculateNewPositions(CurrentPositions, StrikerHitDetails = (force, angle) {}
```

**MainApplication:**

```
Game game = new Game();  
game.init({numberOfPlayers: 3});
```

```
game.executeMove(playerId, move)
```

```
/play?playerId, striker(x,y), angle, force
```

### Entities

#### 1. Game

Properties:

- a. maxnumberOfPlayers
- b. PointTable pointTable
- c. Player[] players
- d. Board board;
- e. Status - INPROGRESS, PAUSE
- f. MoveHistory

Methods:

- a. init() - initialise all entities inside Game,
- b. removePlayer - remove player from array.
- c. exitGame() -
- d. executeMove() -  
Result = Board.updateBoard();  
Pointstable.updatePlayerScore(Result);  
Move created  
updateMoveHistory(Move);  
Return {  
    nextPlayer,  
    status,  
    updatedBoardConfig,  
    updatedPoints  
}

#### 2. Board

a. Properties:

- i. Striker

- ii. Pieces // interface
      - 1. Black piece
      - 2. White piece
      - 3. queen
  - b. Methods:
    - i. init() - sets the initial position of each piece.
    - ii. boardUpdate(); // update positions for each piece including striker  
//calculateScore  
//calculate Penalty
- 3. PointTable
  - a. Properties:
  - b. Map<Integer,Integer> playerIdVSScore;
    - i.
  - c. Methods:
    - i. init() - sets all players score to zero
    - ii. updatePlayerScore(playerId,Integer score); // update the map
    - iii. getScore(playerId)
    - iv. playerId getWinner();
- 4. Player
  - a. Properties:
    - i. playerId
    - ii. Name
  - b. Methods:
- 5. Move:
  - PlayerId
  - Int score
  - Int penalty
  - List<Pience> goalsScored;
  - Board currentPositonOfBoard;
- 6. MoveHistory
- 7. Properties
  - List<Move> allmovesSoFar;
- Methods:
  - AddMoves(Move);