

Player(entity) <div>AI; human</div>	
<ul style="list-style-type: none">String nameint ID	

Interface <div>Piece(entity)<div>King; Queen; Rook; Bishop; Knight; Pawn;</div></div>	
<ul style="list-style-type: none">boolean isValidMove(int x, int y, int z, int w)	

King <div>Piece</div>	
<ul style="list-style-type: none">Player playerisValidMove()	<ul style="list-style-type: none">Board

Queen <div>Piece</div>	
<ul style="list-style-type: none">Player playerisValidMove()	<ul style="list-style-type: none">Board

Rook <div>Piece</div>	
<ul style="list-style-type: none">Player playerisValidMove()	<ul style="list-style-type: none">Board

Bishop <div>Piece</div>	
<ul style="list-style-type: none">Player playerisValidMove()	<ul style="list-style-type: none">Board

Knight		Piece
<ul style="list-style-type: none"> • Player player • isValidMove() 		<ul style="list-style-type: none"> • Board

Pawn		Piece
<ul style="list-style-type: none"> • Player player • isValidMove() 		<ul style="list-style-type: none"> • Board

ModifiedPiece		Piece
<ul style="list-style-type: none"> • \\ We have not decided on rules yet. 		

Abstract Board(use case)		
<ul style="list-style-type: none"> • \\Create a new Chess Board when playing a new game. • Player p1 • Player p2 • Player currentPlayer • Piece[][] board • abstract addPiece() • abstract removePiece() • abstract Board() 		<ul style="list-style-type: none"> • ChessPiece;

OrignialBoard		Board
<ul style="list-style-type: none"> • Player p1 • Player p2 • Player currentPlayer • Piece[][] board • boolean isGameEnd() 		

ModifiedBoard		Board
• \\ We have not decided on rules yet.		

BoardManager(controller & presenter)		
• \\ control the board and update the information about UI		
• void movePiece()		
• getBoardUpdate()		

UI		
• \\ We have not figured out how to make UI yet.		