TCP2201 Project

Trimester 1, 2020/2021

By Ilovik

Team Leader:

Adi Zafri Bin Mohd Tahir, 0165923294,

1181101286@student.mmu.edu.my

Team Members:
Aina Afrina Binti Mohd Nasir , 0193764620 ,
1181101517@student.mmu.edu.my
Iffah Nadhrah Binti Rofie Azizi , 0186608694 ,
1181101285@student.mmu.edu.my
Puteri Nurul Izzah Binti Abdul Hadi , 0138115584 ,
1181100771@student.mmu.edu.my
Siti Aisyah Binti Zali , 0138549762 ,
1181101232@student.mmu.edu.my

1. Installation & Documentation

1.1   For Windows users:

     a. Open file manager.

     b. Find directory of the downloaded zip folder containing source codes.

     c. Run command prompt from the directory by entering "cmd" in the top bar.

     d. Type "javac Ilovik.java".

     e. Type "java Ilovik" and the program should run.

     f. To close the program, click the close (X) button on the top right of the window.

1.2   For LINUX users:

     g. From Terminal install open jdk sudo apt-get install openjdk-7-jdk.

     h. Now to compile use this command from the terminal : javac Ilovik. java. ...

     i. To run the program, type the command in the terminal: java Ilovik.

To view the source code documentation, click on "allclasses-frame.html" or "allclasses-noframe.html".

2. UML Class Diagram

Ilovik
+main(args : String[]) : void

Game
-saveFile : File
-fileName : String = "webalesave.txt"
<<Property>> -board : Board
+Game(b : Board)
+Game()
+loadExists() : boolean
+load() : Board
-validLine(s : String) : boolean
-createSaveFile() : FileWriter
+save() : void
+checkLoad() : boolean
-validLine1or2(s : String) : boolean
-validLine3(s : String) : boolean
-validPieceLine(s : String) : boolean

-game 1

Controller
<<Property>> -view : View
-game : Game
-player1 : Player
-player2 : Player
-board : Board
-start : Square
-end : Square
-sq : Square
-pieceClick : boolean = false
+Controller(view : View, game : Game, player1 : Player, player2 : Player, board : Board)
+initController() : void
-createNewGame() : void
-loadGame() : void
-saveGame() : void
+actionPerformed(e : ActionEvent) : void
-movePiece(i : int, j : int) : void
+main(args : String[]) : void
+changeTurns() : void
-setPlayerTurn() : void
-swapPiece() : void
+setViewBoardIcons() : void
+enableBoardButtons(condition : boolean) : void
+enableSaveButton(condition : boolean) : void
+emptyViewBoardIcons() : void
-resetClick() : void

View
-menuBar : JMenuBar
<<Property>> -newGameMenu : JButton
<<Property>> -loadMenu : JButton
<<Property>> -saveMenu : JButton
<<Property>> -player1Name : JLabel
<<Property>> -player2Name : JLabel
-boardPanel : JPanel
-home : JPanel
+squares : JButton[][] = new JButton[8][7]
-instruction : JLabel
+View()
-createMenu() : void
-addMenuBarMenus() : void
+getSquareButton() : JButton[][]
+addIcon(row : int, col : int, iconFile : String) : void
+getIcon(file : String) : ImageIcon
+emptyIcon() : void

-view 1

1 -board

Board
<<Property>> +squareList : Square[][] = new Square[8][7]
<<Property>> -p1 : Player
<<Property>> -p2 : Player
+Board()
+initBoard() : void
-startingArrangement() : void
+loadArrangement(p : Piece, row : int, col : int) : void
+debug() : void
+Board(p1 : Player, p2 : Player, load : boolean)
+emptyAllSquares() : void
+flip() : void
+isClearPathway(start : Square, end : Square) : boolean
+swapPieces() : void
-forwardClear(start : Square, end : Square) : boolean
-backwardClear(start : Square, end : Square) : boolean
-rightClear(start : Square, end : Square) : boolean
-leftClear(start : Square, end : Square) : boolean
-pathwayClear1(start : Square, end : Square) : boolean
-pathwayClear2(start : Square, end : Square) : boolean
-pathwayClear3(start : Square, end : Square) : boolean
-pathwayClear4(start : Square, end : Square) : boolean

-board 1

Player
<<Property>> -side : char
<<Property>> -name : String
<<Property>> -turn : boolean
<<Property>> -moves : int
<<Property>> -pieceList : Piece = new ArrayList<Piece>()
+Player(name : String, side : char)
+assignPlayerPieces(side : char) : void
+piecesCaptured(captured : boolean) : void
+isWinner() : boolean
+updateMoves() : void
+swapPiece() : void
+getPieceList() : ArrayList<Piece>
+isTurn() : boolean
+debug() : void

-player1
-player2

-p2
-p1

Square
<<Property>> -x : int
<<Property>> -y : int
<<Property>> -occupied : boolean
<<Property>> -piece : Piece
-fileName : String = "Sun.png"
+isNearDestination(end : Square) : boolean
+Square(squareBuilder : Builder)
+isOccupied() : boolean

+squareList
-end 1
-start 1
-sq 1

Piece
<<Property>> -name : String
<<Property>> -iconFile : String
<<Property>> -side : char
<<Property>> -captured : boolean
+Piece(side : char, iconFile : String)
+setName() : void
+move(start : Square, end : Square) : boolean
+isCaptured() : boolean

-pieceList
-piece
-piece

Arrow
-fileName : String = "Arrow.png"
<<Property>> -rotate : boolean
+Arrow(side : char, rotate : boolean)
+isRotating() : boolean
+move(start : Square, end : Square) : boolean

Chevron
-fileName : String = "Chevron.png"
+Chevron(side : char)
+move(start : Square, end : Square) : boolean

Plus
-fileName : String = "Plus.png"
+Plus(side : char)
+move(start : Square, end : Square) : boolean

Builder
-x : int
-y : int
-occupied : boolean
-piece : Piece
+x(X : int) : Builder
+y(Y : int) : Builder
+piece(P : Piece) : Builder
+occupied(O : boolean) : Builder
+build() : Square

Triangle
-fileName : String = "Triangle.png"
+Triangle(side : char)
+move(start : Square, end : Square) : boolean

Sun
+Sun(side : char)
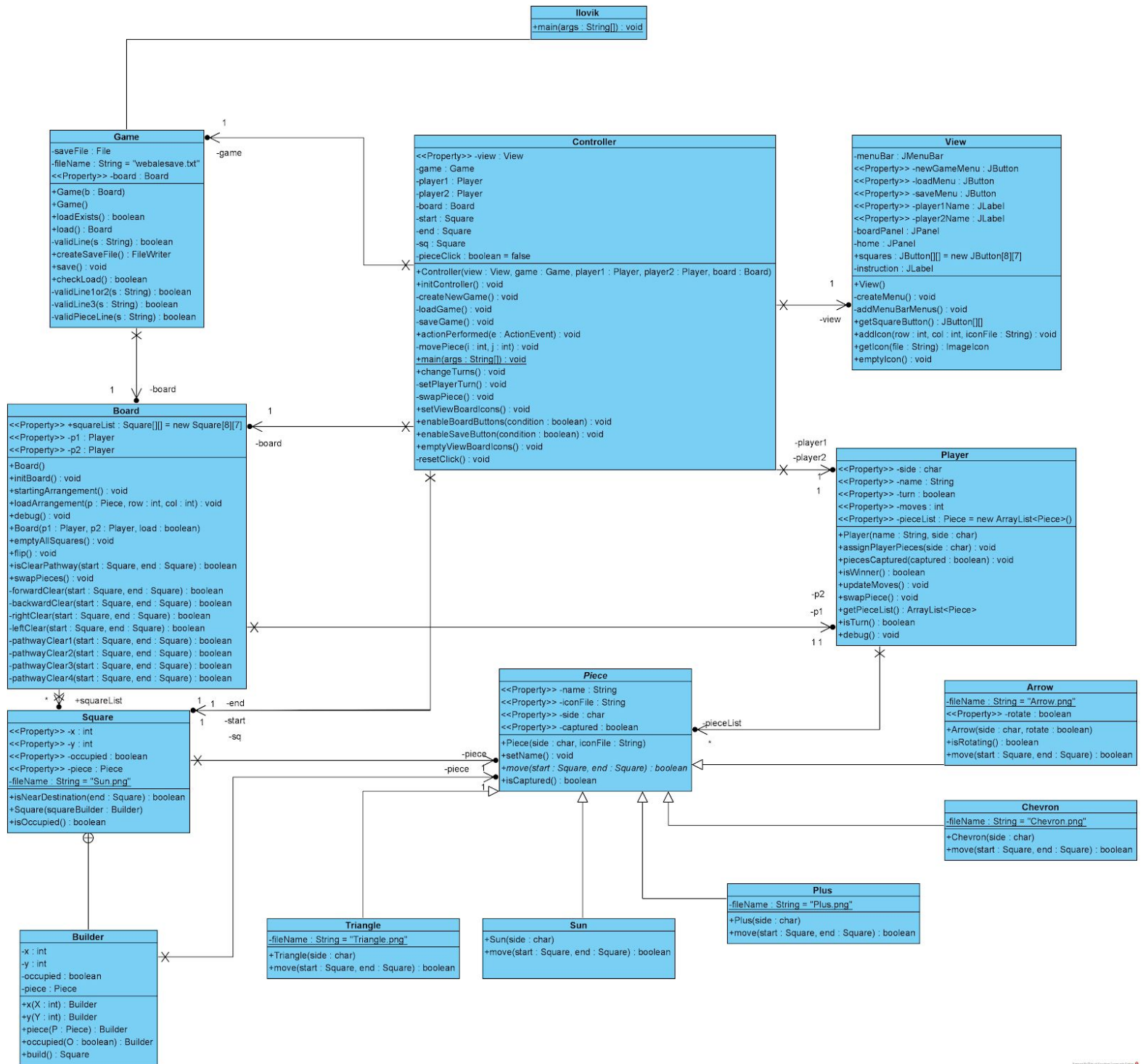+move(start : Square, end : Square) : boolean

Figure 1.1 UML Class Diagram
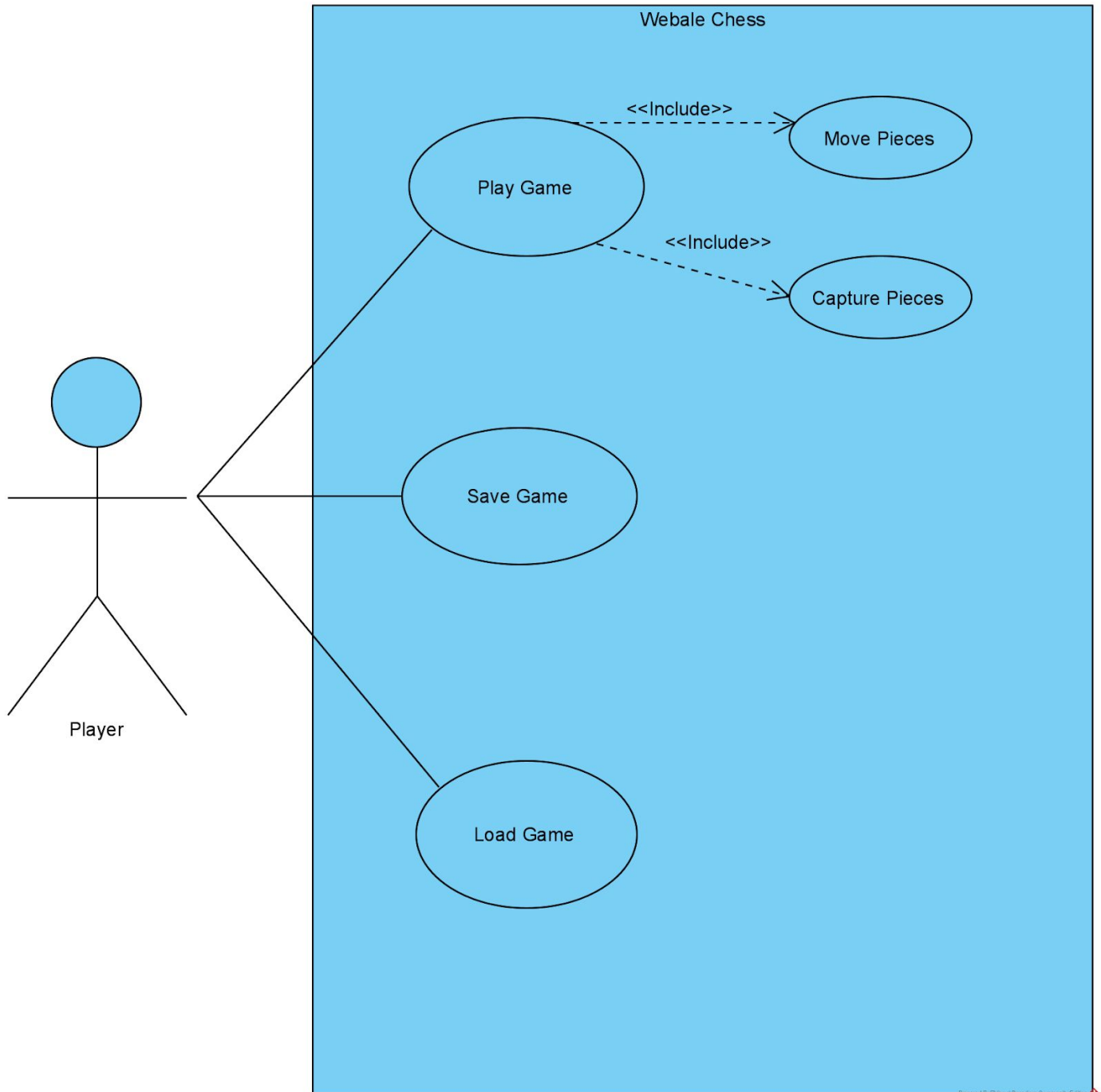
3.   Use Case Diagram

Figure 2.1 Use Case Diagram
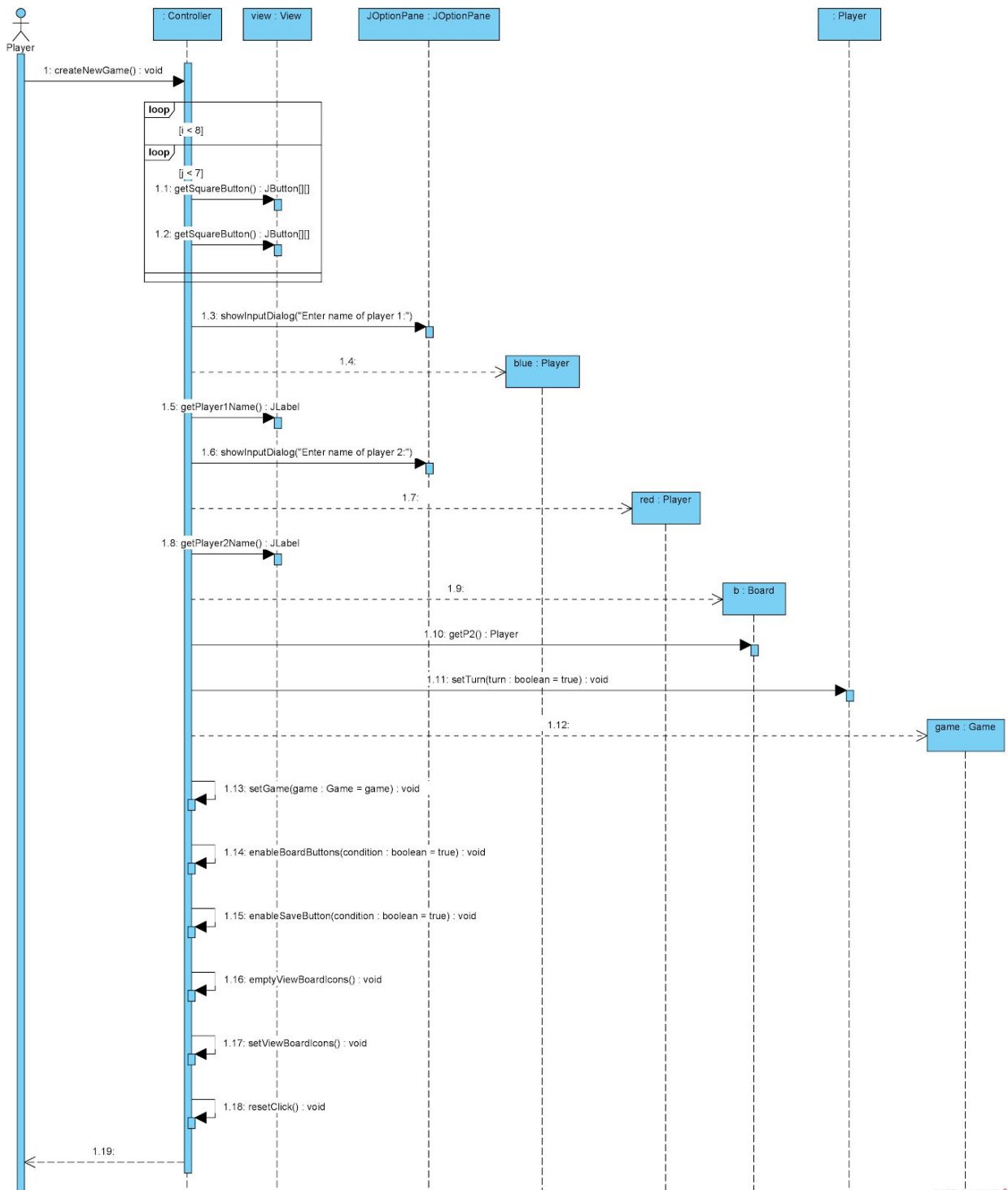
4.   Sequence Diagram

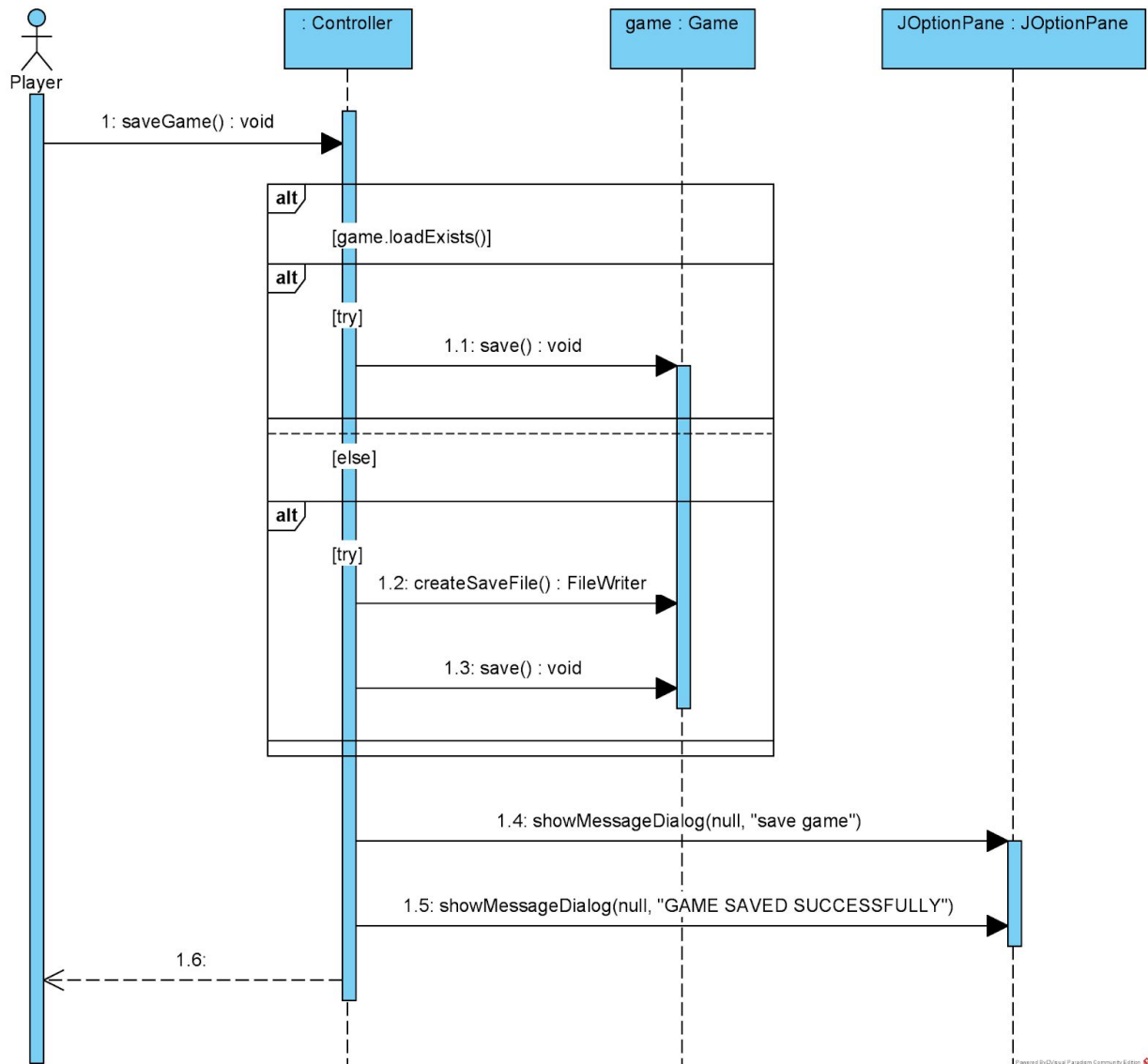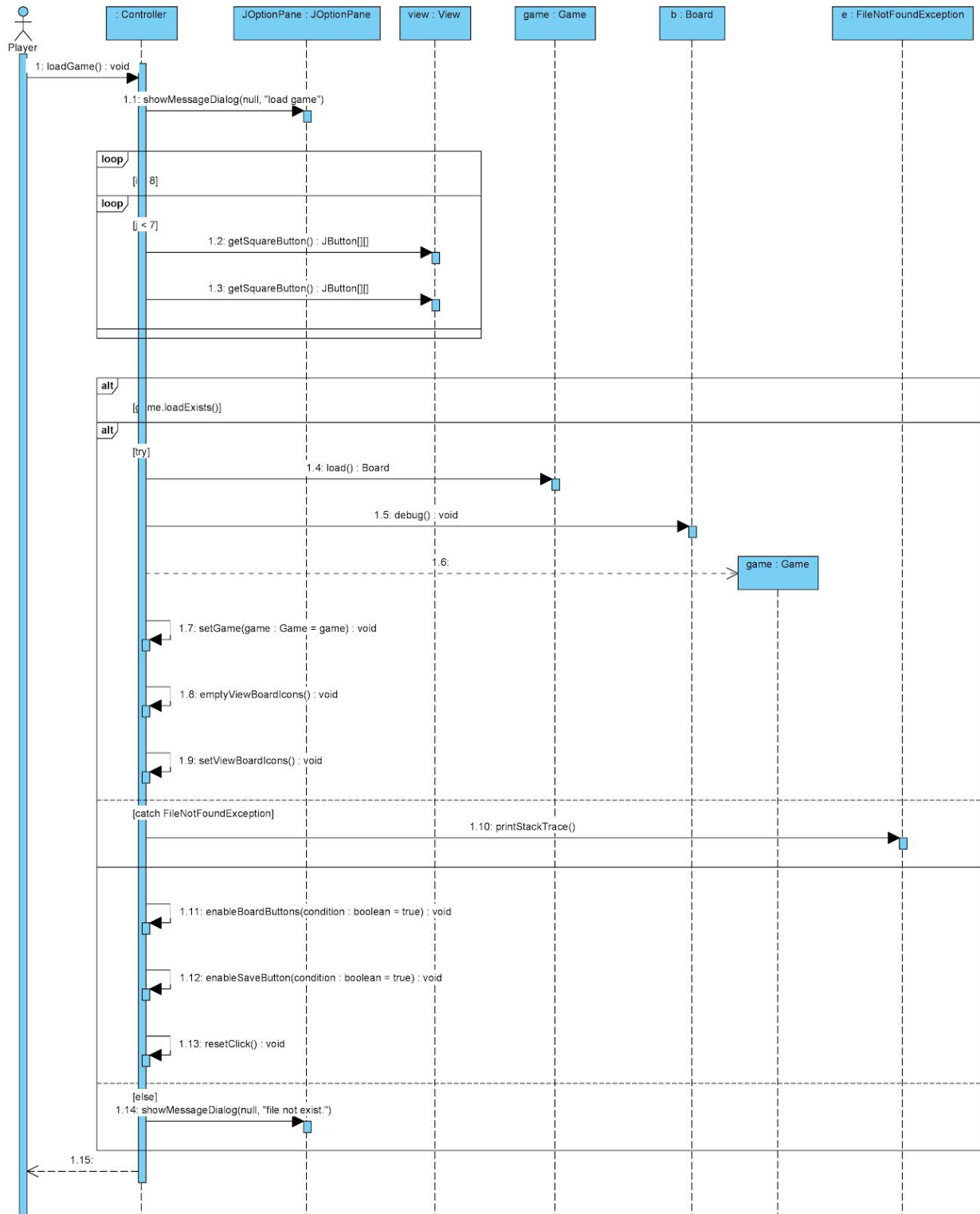

Figure 3.1 Play Game Sequence Diagram

Figure 3.2 Save Game Sequence Diagram

Figure 3.3 Load Game Sequence Diagram