edu.monash
game
actions
<<Interface>>
Action
+ isValid(game : Game, player : Player): boolean
+ excuteOn(Game game): boolean
LoadAction
- filepath: String
- deserializer: Deserializer
+ isValid(game : Game, player : Player): boolean
+ excuteOn(Game game): boolean
- load() : void
MoveAction
- move : Move {read only}
+ MoveAction(player : Player, from : Integer, to : Integer)
+ isValid(game : Game, player : Player): boolean
+ excuteOn(Game game): boolean
SaveAction
- filePath : String
- serializer : Serializer
+ isValid(game : Game, player : Player): boolean
+ excuteOn(Game game): boolean
- save() : void
UndoAction
+ isValid(game : Game, player : Player): boolean
+ excuteOn(Game game): boolean
1
creates
1
Deserializer
- filePath: String
- load(): String
1
1
creates
1
Serializer
- filePath: String
- save(): String
applies to
1
«enum»
Piece
1
Board
- NUM_RINGS : int {read only}
- NUM_POSITIONS_PER_RING : int {read only}
- postions : List<Position> {read only}
+ Board()
- createBoardStructure() : List<Position>
+ getPosition(id : int) : Position
- previousOfY(y : int) : int
- nextOfY(y : int) : int
- previousOfX(x : int) : int
- nextOfX(x : int) : int
- toPositiveIndex(index : int, length : int) : int
has
1
24
has neighbours
1
2...4
Position
- idCounter : int {read only}
- id : int {read only}
- neighbourUp : Position
- neighbourDown : Position
- neighbourLeft : Position
- neighbourRight : Position
- OccupiedBy : Piece
+ Position()
+ getId() : int
+ withUpNeighbour(neighbourUp : Position): Position
+ withDownNeighbour(neighbourDown : Position): Position
+ withLeftNeighbour(neighbourLeft : Position): Position
+ withRightNeighbour(neighbourRight : Position): Position
+ canPieceBePlaced() : boolean
+ canPieceBeMoved(player : Player) : boolean
+ canPieceBeRemoved(player : Player) : boolean
+ getPiece() : Piece
+ setPiece(piece : Piece) : void
+ isInHorizontalMill() : boolean
+ isInVerticalMill() : boolean
+ isHoriziontalAnchor() : boolean
+ isVerticalAnchor() : boolean
1
1
executes on
is played on
0...*
1
Move
- piece : Piece {read only}
- from : Integer {read only}
- to : Integer {read only}
+ Move(piece : Piece,from : Integer,to : Integer)
+ getFrom() : Integer
+ getTo(): Integer
+ executeOn(board : Board) : void
0...*
creates
1
Game
- board : Board
- player1 : Player
- player2 : Player
- currentPlayer : Player
- trunCount : int
- movesPlayed : Stack<Move>
+ Game()
+ getBoard() : Board
+ excute(action : Action) : void
+ storePlayerMove(move : Move) : void
+ initializeNewGame() : void
+ switchPlayer() : void
+ getCurrentPlayer() : Player
1
has
2
player
<<Interface>>
PlayerPhase
+ validate(board : Board, move : Move): Boolean
1
has
1
Player
- pieceColour : Piece {read only}
- phase : PlayerPhase
+ Player(pieceColour : Piece, initialPhaseConstructor : Function<Player, PlayerPhase>)
+ getPieceColour() : Piece
+ getPhase() : PlayerPhase
+ setPhase(phaseConstructor : Function<Player, PlayerPhase>) : void
JumpPhase
- player : Player {read only}
+ JumpPhase(player : Player)
+ validate(board : Board, move : Move): boolean
SlidePhase
- player : Player {read only}
+ SlidePhase(player : Player)
+ validate(board : Board, move : Move): boolean
PlacePhase
- player : Player {read only}
+ PlacePhase(player : Player)
+ validate(board : Board, move : Move): boolean
ViewController
- GRID_PANE_CELL__FORMAT : DateFormat = new DateFormat(...) {read only}
- game : Game
- boardGridChildren : ObservableList<ImageView>
- stage : Stage
- boardGrid : GridPane
- blackGrid : GridPane
- whiteGrid : GridPane
+ ViewController()
+ setStage(stage : Stage) : void
- newGame() : void
+ quitGame() : void
+ showPromptDialog(title : String, header : String, content : String, yesCallBack : Runnable, noCallBack : Runnable) : void
- boardMapping : Integer[][] = {{...}} {read only}
+ getPositionID(x: int, y: int) : Integer
- setUpDragAndDrop(gridPane : GridPane) : void
- onDragDetectedHandler(child : Node, event : MouseEvent) : void
- onDragOverHandler(child : Node, event : MouseEvent) : void
- onDragDroppedHandler(child : Node, event : MouseEvent) : void
- enableBlur() : void
- disableBlur() : void
- initialize() : void
GUIController
- boardGridChildren : ObservableList<ImageView>
- stage : Stage
- boardGrid : GridPane
- leftTitleGrid : GridPane
- rightTitleGrid : GridPane
+ setStage(stage : Stage) : void
- newGame() : void
+ quitGame() : void
- promptDialogAndRestartGame(title : String, header : String,contentText : String, id : int) : void
- handleAbout() : void
- showInvalidMoveDialog(context : String) : void
- enableBlur() : void
- disableBlur() : void
- initialize() : void
Main
+ start(stage : Stage) : void
+ main(args : String[]) : void