Model Report

Class Model

Version ● Proposed

Date/Time Generated:	9/5/2021 1:15:37 μμ
Author:	pantz

EA Repository : D:\PROGRAMS\Eclipse\eclipse-java-2020-12-R-win32-x86_64\workspace\tic-tac-toe\tic-tac-toe.EAP

Table of Contents

1 Class Model

Package in package 'Model'

Class Model
Version Phase 1.0 Proposed
me created on 3/5/2021. Last modified 3/5/2021

1.1 main

Package in package 'Class Model'

main
Version 1.0 Phase 1.0 Proposed
pantz created on 9/5/2021. Last modified 9/5/2021

1.1.1 tictactoe

Package in package 'main'

tictactoe
Version 1.0 Phase 1.0 Proposed
pantz created on 9/5/2021. Last modified 9/5/2021

1.1.1.1 tictactoe diagram

Class diagram in package 'tictactoe'

tictactoe
Version 1.0
pantz created on 9/5/2021. Last modified 9/5/2021

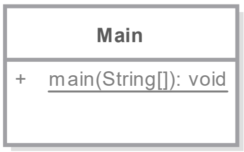

Figure 1: tictactoe

1.1.1.2 ai

Package in package 'tictactoe'

ai
Version 1.0 Phase 1.0 Proposed
pantz created on 9/5/2021. Last modified 9/5/2021

1.1.1.2.1 ai diagram

Class diagram in package 'ai'

ai
Version 1.0
pantz created on 9/5/2021. Last modified 9/5/2021

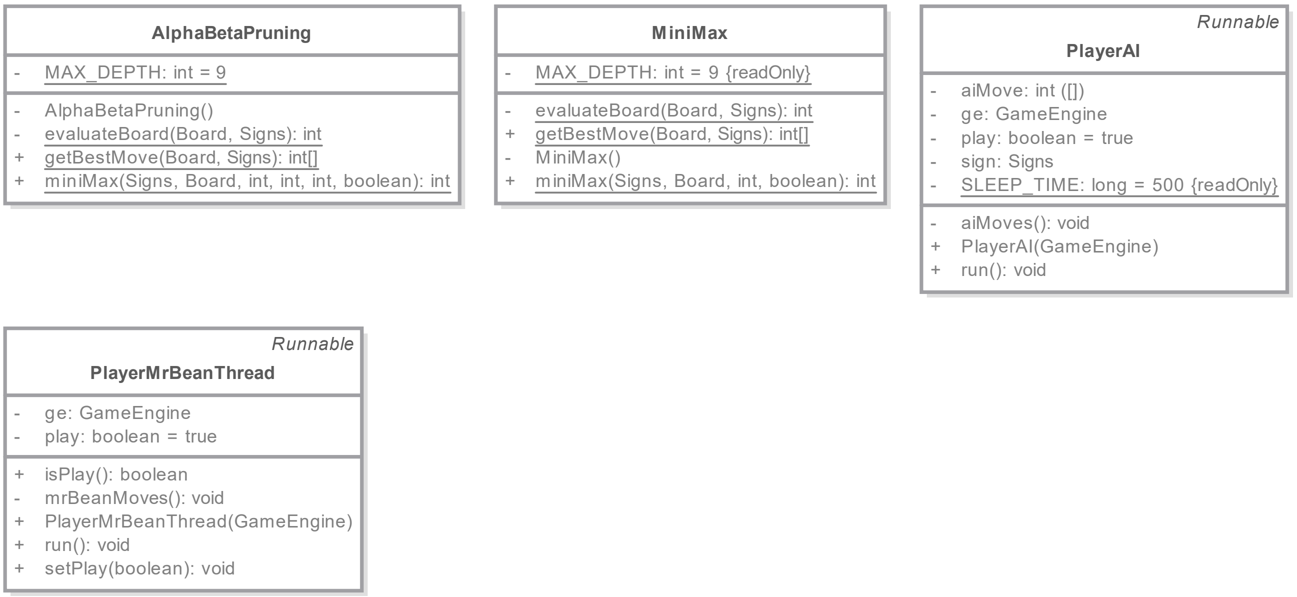

Figure 2: ai

1.1.1.2.2 AlphaBetaPruning

Class in package 'ai'

AlphaBetaPruning
Version 1.0 Phase 1.0 Proposed
pantz created on 9/5/2021. Last modified 9/5/2021

ATTRIBUTES

MAX_DEPTH : int Private = 9

[Is static True. Containment is Not Specified.]

OPERATIONS

AlphaBetaPruning () : Private

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

evaluateBoard (board : Board , sign : Signs) : int Private

Evaluate the given board from the perspective of the X player, return 1 if a winning board configuration is found, -1 for a losing one and 0 for a draw.
@return value of the board

[Is static True. Is abstract False. Is return array False. Is query False. Is synchronized False.]

getBestMove (board : Board , sign : Signs) : int Public

Evaluate every legal move on the board and return the best one.
@return Coordinates of best move

[Is static True. Is abstract False. Is return array True. Is query False. Is synchronized False.]

OPERATIONS

◆ miniMax (sign : Signs , board : Board , depth : int , alpha : int , beta : int , isMax : boolean) : int Public

Play moves on the board alternating between playing as X and O analysing the board each time to return the value of the highest value move for the player. Use variables alpha and beta as the best alternative for the maximising player (sign) and the best alternative for the minimising player (opponent) respectively, do not search descendants of nodes if player's alternatives are better than the node. Return the highest value move when a terminal node or the maximum search depth is reached.
@return Value of the board

[Is static True. Is abstract False. Is return array False. Is query False. Is synchronized False.]

1.1.1.2.3 MiniMax

Class in package 'ai'

MiniMax
Version 1.0 Phase 1.0 Proposed
pantz created on 9/5/2021. Last modified 9/5/2021

ATTRIBUTES

◆ MAX_DEPTH : int Private Const = 9

[Is static True. Containment is Not Specified.]

OPERATIONS

◆ evaluateBoard (board : Board , sign : Signs) : int Private

[Is static True. Is abstract False. Is return array False. Is query False. Is synchronized False.]

◆ getBestMove (board : Board , sign : Signs) : int Public

Evaluate every legal move on the board and return the best one.
@return Coordinates of best move

[Is static True. Is abstract False. Is return array True. Is query False. Is synchronized False.]

◆ MiniMax () : Private

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

◆ miniMax (sign : Signs , board : Board , depth : int , isMax : boolean) : int Public

Play moves on the board alternating between playing as X and O analysing the board each time to return the value of the highest value move for the (sign) player. Return the highest value move when a terminal node or the maximum search depth is reached.
@return Value of the board

[Is static True. Is abstract False. Is return array False. Is query False. Is synchronized False.]

1.1.1.2.4 PlayerAI

Class in package 'ai'

The AI thread class It checks if it is his turn every 500 milliseconds and depending on the side it plays accordingly. The AI is using the alpha beta pruning algorithm.

PlayerAI
Version 1.0 Phase 1.0 Proposed
pantz created on 9/5/2021. Last modified 9/5/2021

Extends Runnable

ATTRIBUTES

aiMove : int Private

[Is static False. Containment is Not Specified.]

ge : GameEngine Private

[Is static False. Containment is Not Specified.]

play : boolean Private = true

[Is static False. Containment is Not Specified.]

sign : Signs Private

[Is static False. Containment is Not Specified.]

SLEEP_TIME : long Private Const = 500

[Is static True. Containment is Not Specified.]

ASSOCIATIONS

Association (direction: Source -> Destination)

Source: Public (Class) PlayerAI Target: Private ge (Class) GameEngine

OPERATIONS

aiMoves () : void Private

Properties:
throws = InterruptedException
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

PlayerAI (ge : GameEngine) : Public
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

run () : void Public

Properties:
annotations = @Override
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

1.1.1.2.5 PlayerMrBeanThread

Class in package 'ai'

PlayerMrBeanThread
Version 1.0 Phase 1.0 Proposed
pantz created on 9/5/2021. Last modified 9/5/2021
Extends Runnable

ATTRIBUTES

ge : GameEngine Private

ATTRIBUTES

[Is static False. Containment is Not Specified.]

play : boolean Private = true

[Is static False. Containment is Not Specified.]

ASSOCIATIONS

Association (direction: Source -> Destination)

Source: Public (Class) PlayerMrBeanThread

Target: Private ge (Class) GameEngine

OPERATIONS

isPlay () : boolean Public

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

mrBeanMoves () : void Private

Properties:
throws = InterruptedException

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

PlayerMrBeanThread (ge : GameEngine) : Public

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

run () : void Public

Properties:
annotations = @Override

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

setPlay (play : boolean) : void Public

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

1.1.1.3 controller

Package in package 'tictactoe'

controller
Version 1.0 Phase 1.0 Proposed
pantz created on 9/5/2021. Last modified 9/5/2021

1.1.1.3.1 controller diagram

Class diagram in package 'controller'

controller
Version 1.0
pantz created on 9/5/2021. Last modified 9/5/2021

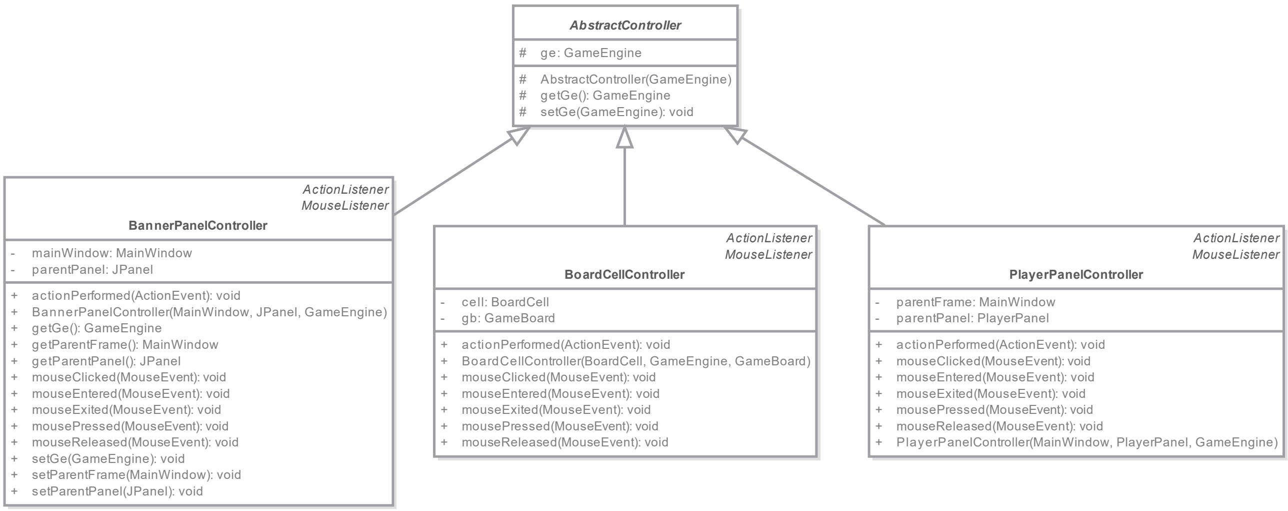
AbstractController
# ge: GameEngine
# AbstractController(GameEngine)
# getGe(): GameEngine
# setGe(GameEngine): void
ActionListener
MouseListener
BannerPanelController
- mainWindow: MainWindow
- parentPanel: JPanel
+ actionPerformed(ActionEvent): void
+ BannerPanelController(MainWindow, JPanel, GameEngine)
+ getGe(): GameEngine
+ getParentFrame(): MainWindow
+ getParentPanel(): JPanel
+ mouseClicked(MouseEvent): void
+ mouseEntered(MouseEvent): void
+ mouseExited(MouseEvent): void
+ mousePressed(MouseEvent): void
+ mouseReleased(MouseEvent): void
+ setGe(GameEngine): void
+ setParentFrame(MainWindow): void
+ setParentPanel(JPanel): void
ActionListener
MouseListener
BoardCellController
- cell: BoardCell
- gb: GameBoard
+ actionPerformed(ActionEvent): void
+ BoardCellController(BoardCell, GameEngine, GameBoard)
+ mouseClicked(MouseEvent): void
+ mouseEntered(MouseEvent): void
+ mouseExited(MouseEvent): void
+ mousePressed(MouseEvent): void
+ mouseReleased(MouseEvent): void
ActionListener
MouseListener
PlayerPanelController
- parentFrame: MainWindow
- parentPanel: PlayerPanel
+ actionPerformed(ActionEvent): void
+ mouseClicked(MouseEvent): void
+ mouseEntered(MouseEvent): void
+ mouseExited(MouseEvent): void
+ mousePressed(MouseEvent): void
+ mouseReleased(MouseEvent): void
+ PlayerPanelController(MainWindow, PlayerPanel, GameEngine)

Figure 3: controller

1.1.1.3.2 AbstractController

Class in package 'controller'

This is an abstract controller class which gets the GameEngine instance. All classes extend this class must set the GameEngine object.

AbstractController
Version 1.0 Phase 1.0 Proposed
pantz created on 9/5/2021. Last modified 9/5/2021

INCOMING STRUCTURAL RELATIONSHIPS
Generalization from BoardCellController to AbstractController [Direction is 'Source -> Destination'.]
Generalization from PlayerPanelController to AbstractController [Direction is 'Source -> Destination'.]
Generalization from BannerPanelController to AbstractController [Direction is 'Source -> Destination'.]

ATTRIBUTES

ge : GameEngine Protected

[Is static False. Containment is Not Specified.]

ASSOCIATIONS

Association (direction: Source -> Destination)

Source: Public (Class) AbstractController Target: Protected ge (Class) GameEngine

OPERATIONS

AbstractController (ge : GameEngine) : Protected
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

getGe () : GameEngine Protected
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

setGe (ge : GameEngine) : void Protected
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

1.1.1.3.3 BannerPanelController

Class in package 'controller'

This is the controller class for the top banner component which has three buttons. Handles the request for the Add Player command. Handles the request for the Quit Button (Program ends) Handles the request for the Done Button.

BannerPanelController
Version 1.0 Phase 1.0 Proposed
pantz created on 9/5/2021. Last modified 9/5/2021
Extends AbstractController, ActionListener, MouseListener

OUTGOING STRUCTURAL RELATIONSHIPS

Generalization from BannerPanelController to AbstractController

[Direction is 'Source -> Destination'.]

ATTRIBUTES

mainWindow : MainWindow Private

[Is static False. Containment is Not Specified.]

parentPanel : JPanel Private

[Is static False. Containment is Not Specified.]

ASSOCIATIONS

Association (direction: Source -> Destination)

Source: Public (Class) BannerPanelController Target: Private mainWindow (Class) MainWindow

ASSOCIATIONS

Association (direction: Source -> Destination)

Source: Public (Class) BannerPanel	Target: Private bannerPanelController (Class) BannerPanelController

OPERATIONS

actionPerformed (e : ActionEvent) : void Public

Actions coming from BannerPanel

Properties:
annotations = @Override
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

BannerPanelController (parentFrame : MainWindow , parentPanel : JPanel , ge : GameEngine) : Public

Constructor
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

getGe () : GameEngine Public
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

getParentFrame () : MainWindow Public
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

getParentPanel () : JPanel Public
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

mouseClicked (e : MouseEvent) : void Public

Properties:
annotations = @Override
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

mouseEntered (e : MouseEvent) : void Public

Properties:
annotations = @Override
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

mouseExited (e : MouseEvent) : void Public

Properties:
annotations = @Override
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

mousePressed (e : MouseEvent) : void Public

Properties:
annotations = @Override
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

mouseReleased (e : MouseEvent) : void Public

OPERATIONS

Properties:
annotations = @Override
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

setGe (ge : GameEngine) : void Public
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

setParentFrame (parentFrame : MainWindow) : void Public
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

setParentPanel (parentPanel : JPanel) : void Public
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

1.1.1.3.4 BoardCellController

Class in package 'controller'

This is the controller class for the BoardCell view class and the GameBoard class.

BoardCellController
Version 1.0 Phase 1.0 Proposed
pantz created on 9/5/2021. Last modified 9/5/2021
Extends AbstractController, ActionListener, MouseListener

OUTGOING STRUCTURAL RELATIONSHIPS

Generalization from BoardCellController to AbstractController
[Direction is 'Source -> Destination'.]

ATTRIBUTES

cell : BoardCell Private
[Is static False. Containment is Not Specified.]

gb : GameBoard Private
[Is static False. Containment is Not Specified.]

ASSOCIATIONS

Association (direction: Source -> Destination)

Source: Public (Class) BoardCellController Target: Private gb (Class) GameBoard

Association (direction: Source -> Destination)

Source: Public (Class) BoardCellController Target: Private cell (Class) BoardCell

OPERATIONS

actionPerformed (e : ActionEvent) : void Public

OPERATIONS

Properties:
annotations = @Override
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

BoardCellController (cell : BoardCell , ge : GameEngine , gb : GameBoard) : Public

Constructor
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

mouseClicked (e : MouseEvent) : void Public

Properties:
annotations = @Override
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

mouseEntered (e : MouseEvent) : void Public

Properties:
annotations = @Override
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

mouseExited (e : MouseEvent) : void Public

Properties:
annotations = @Override
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

mousePressed (e : MouseEvent) : void Public

Properties:
annotations = @Override
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

mouseReleased (e : MouseEvent) : void Public

Properties:
annotations = @Override
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

1.1.1.3.5 PlayerPanelController

Class in package 'controller'

This is the controller class for the PlayerPanel view class.

PlayerPanelController
Version 1.0 Phase 1.0 Proposed
pantz created on 9/5/2021. Last modified 9/5/2021
Extends AbstractController, ActionListener, MouseListener

OUTGOING STRUCTURAL RELATIONSHIPS

Generalization from PlayerPanelController to AbstractController
[Direction is 'Source -> Destination'.]

ATTRIBUTES

parentFrame : MainWindow Private

[Is static False. Containment is Not Specified.]

parentPanel : PlayerPanel Private

[Is static False. Containment is Not Specified.]

ASSOCIATIONS

Association (direction: Source -> Destination)

Source: Public (Class) PlayerPanelController

Target: Private parentFrame (Class) MainWindow

Association (direction: Source -> Destination)

Source: Public (Class) PlayerPanelController

Target: Private parentPanel (Class) PlayerPanel

Association (direction: Source -> Destination)

Source: Public (Class) PlayerPanel

Target: Private playerPanelController (Class) PlayerPanelController

OPERATIONS

actionPerformed (e : ActionEvent) : void Public

Properties:
annotations = @Override

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

mouseClicked (e : MouseEvent) : void Public

Properties:
annotations = @Override

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

mouseEntered (e : MouseEvent) : void Public

Properties:
annotations = @Override

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

mouseExited (e : MouseEvent) : void Public

Properties:
annotations = @Override

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

mousePressed (e : MouseEvent) : void Public

Properties:
annotations = @Override

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

mouseReleased (e : MouseEvent) : void Public

OPERATIONS

Properties:
annotations = @Override

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

◆ PlayerPanelController (parentFrame : MainWindow , parentPanel : PlayerPanel , ge : GameEngine) : Public

Constructor

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

1.1.1.4 io

Package in package 'tictactoe'

io
Version 1.0 Phase 1.0 Proposed
pantz created on 9/5/2021. Last modified 9/5/2021

1.1.1.4.1 io diagram

Class diagram in package 'io'

io
Version 1.0
pantz created on 9/5/2021. Last modified 9/5/2021

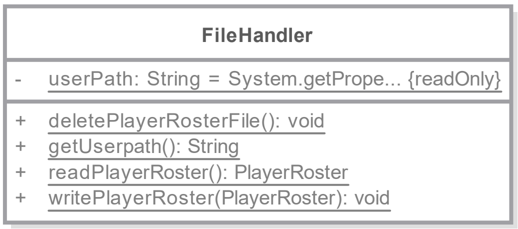
FileHandler
- userPath: String = System.getPrope... {readOnly}
+ deletePlayerRosterFile(): void
+ getUserpath(): String
+ readPlayerRoster(): PlayerRoster
+ writePlayerRoster(PlayerRoster): void

Figure 4: io

1.1.1.4.2 FileHandler

Class in package 'io'

This class is the responsible class for reading/writing (serializing/deserializing) the PlayerRoster class, which holds all the game data and players. It consists of static methods in order to be accessed by all classes independently.

FileHandler
Version 1.0 Phase 1.0 Proposed
pantz created on 9/5/2021. Last modified 9/5/2021

ATTRIBUTES

userPath : String Private Const   = System.getProperty("user.home") + File.separator + "tictactoe.ser"

The user's profile path from the OS

[Is static True. Containment is Not Specified.]

OPERATIONS

deletePlayerRosterFile () : void Public

[Is static True. Is abstract False. Is return array False. Is query False. Is synchronized False.]

getUserpath () : String Public

[Is static True. Is abstract False. Is return array False. Is query False. Is synchronized False.]

readPlayerRoster () : PlayerRoster Public

Reads the file tictactoe.ser from the user's profile If it is not found writes an error to console.
@return PlayerRoster instance

OPERATIONS

[Is static True. Is abstract False. Is return array False. Is query False. Is synchronized False.]

◆ writePlayerRoster (playerRoster : PlayerRoster) : void Public

Writes an instance of PlayerRoster class to a file in user's profile.

[Is static True. Is abstract False. Is return array False. Is query False. Is synchronized False.]

1.1.1.5 model

Package in package 'tictactoe'

model
Version 1.0 Phase 1.0 Proposed
pantz created on 9/5/2021. Last modified 9/5/2021

1.1.1.5.1 model diagram

Class diagram in package 'model'

model
Version 1.0
pantz created on 9/5/2021. Last modified 9/5/2021

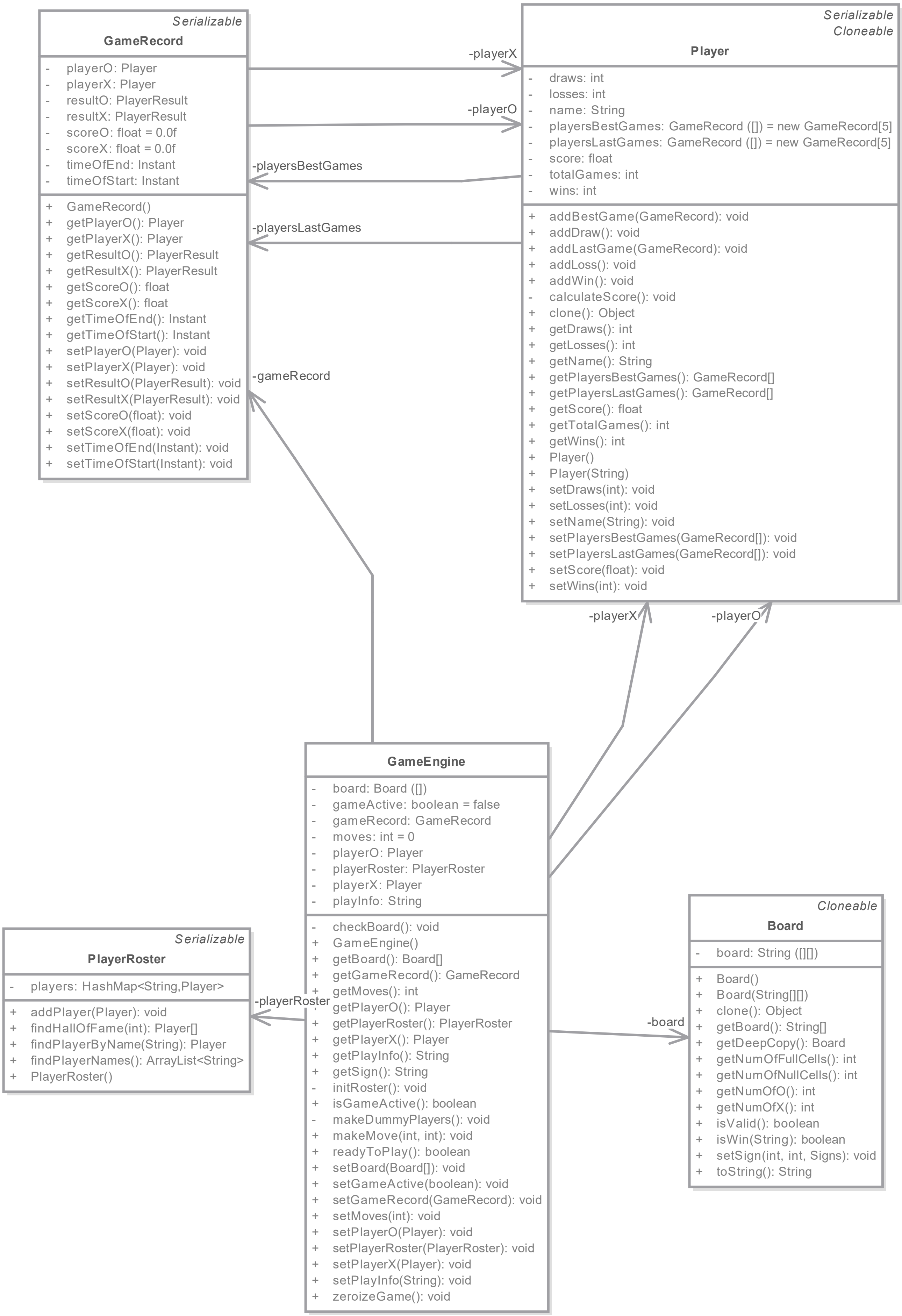

Figure 5: model

1.1.1.5.2 enums

Package in package 'model'

enums
Version 1.0 Phase 1.0 Proposed
pantz created on 9/5/2021. Last modified 9/5/2021

1.1.1.5.2.1 enums diagram

Class diagram in package 'enums'

enums
Version 1.0
pantz created on 9/5/2021. Last modified 9/5/2021

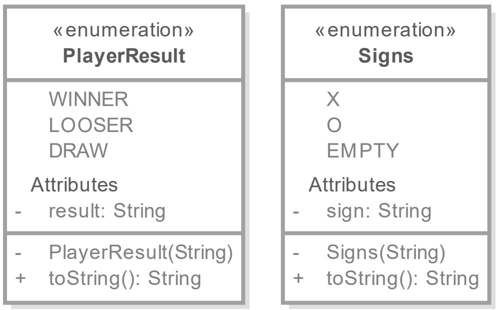

Figure 6: enums

1.1.1.5.2.2 PlayerResult

Enumeration in package 'enums'

PlayerResult
Version 1.0 Phase 1.0 Proposed
pantz created on 9/5/2021. Last modified 9/5/2021

ATTRIBUTES

WINNER : Public

Properties:
arguments = "WINNER"

[Stereotype is «enum». Is static False. Containment is Not Specified.]

LOOSER : Public

Properties:
arguments = "LOOSER"

[Stereotype is «enum». Is static False. Containment is Not Specified.]

DRAW : Public

Properties:

ATTRIBUTES
arguments = "DRAW" [Stereotype is «enum». Is static False. Containment is Not Specified.]
result : String Private [Is static False. Containment is Not Specified.]

OPERATIONS
PlayerResult (result : String) : Private [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
toString () : String Public Properties: annotations = @Override [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

1.1.1.5.2.3 Signs

Enumeration in package 'enums'

Signs
Version 1.0 Phase 1.0 Proposed
pantz created on 9/5/2021. Last modified 9/5/2021

ATTRIBUTES
X :   Public Properties: arguments = "X" [Stereotype is «enum». Is static False. Containment is Not Specified.]
O :   Public Properties: arguments = "O" [Stereotype is «enum». Is static False. Containment is Not Specified.]
EMPTY :   Public Properties: arguments = "_" [Stereotype is «enum». Is static False. Containment is Not Specified.]
sign : String Private [Is static False. Containment is Not Specified.]

OPERATIONS
Signs (result : String) : Private [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

OPERATIONS

toString () : String Public

Properties:
annotations = @Override

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

1.1.1.5.3 Board

Class in package 'model'

The Board class is an immutable class. It represents the tic tac toe game board logically.

Board
Version 1.0 Phase 1.0 Proposed
pantz created on 9/5/2021. Last modified 9/5/2021
Extends Cloneable

ATTRIBUTES

board : String Private

[Is static False. Containment is Not Specified.]

ASSOCIATIONS

Association (direction: Source -> Destination)

Source: Public (Class) BoardTest Target: Private b3 (Class) Board

Association (direction: Source -> Destination)

Source: Public (Class) GameEngine Target: Private board (Class) Board

Association (direction: Source -> Destination)

Source: Public (Class) AlphaBetaPruningTest Target: Private b3 (Class) Board

Association (direction: Source -> Destination)

Source: Public (Class) BoardTest Target: Private b2 (Class) Board

Association (direction: Source -> Destination)

Source: Public (Class) MiniMaxTest Target: Private b (Class) Board

Association (direction: Source -> Destination)

Source: Public (Class) BoardTest Target: Private b1 (Class) Board

OPERATIONS

Board () : Public

Constructor

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

OPERATIONS

Board (b : String[][]) : Public

Getters and Setters

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

clone () : Object Public

Clone method

Properties:
annotations = @Override
throws = CloneNotSupportedException

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

getBoard () : String Public

[Is static False. Is abstract False. Is return array True. Is query False. Is synchronized False.]

getDeepCopy () : Board Public

Gets a deep copy of the Board
@return a copy of the Board class

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

getNumOfFullCells () : int Public

Returns the number of filled cells of a board
@return int

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

getNumOfNullCells () : int Public

Returns the number of empty cells of a board
@return int

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

getNumOfO () : int Public

Returns the number of O's
@return int

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

getNumOfX () : int Public

Returns the number of X's
@return int

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

isValid () : boolean Public

Returns true if a Board is valid, according to the regulations of the game
@return boolean

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

isWin (s : String) : boolean Public

Ths method takes as parameter a String (X or O) and checks if its the winner

OPERATIONS

@return boolean

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

setSign (row : int , col : int , sign : Signs) : void Public

Sets the sign param to the corresponding position

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

toString () : String Public

Properties:
annotations = @Override

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

1.1.1.5.4 GameEngine

Class in package 'model'

This class is the logic heart of the game it decides which player is turn by the number of moves Typically in tictactoe the maximum moves of a game are 9 Taking into consideration that X always moves first. so move0 -> X move1 -> O move2 -> X and so on. This means even number of moves means tha X plays and odd number of moves the O plays.

GameEngine
Version 1.0 Phase 1.0 Proposed
pantz created on 9/5/2021. Last modified 9/5/2021

ATTRIBUTES

board : Board Private

[Is static False. Containment is Not Specified.]

gameActive : boolean Private = false

[Is static False. Containment is Not Specified.]

gameRecord : GameRecord Private

[Is static False. Containment is Not Specified.]

moves : int Private = 0

[Is static False. Containment is Not Specified.]

playerO : Player Private

[Is static False. Containment is Not Specified.]

playerRoster : PlayerRoster Private

[Is static False. Containment is Not Specified.]

playerX : Player Private

[Is static False. Containment is Not Specified.]

playInfo : String Private

[Is static False. Containment is Not Specified.]

ASSOCIATIONS

Association (direction: Source -> Destination)

Source: Public (Class) GameEngine	Target: Private gameRecord (Class) GameRecord

Association (direction: Source -> Destination)

Source: Public (Class) GameEngine	Target: Private board (Class) Board

Association (direction: Source -> Destination)

Source: Public (Class) GameEngine	Target: Private playerO (Class) Player

Association (direction: Source -> Destination)

Source: Public (Class) GameEngine	Target: Private playerX (Class) Player

Association (direction: Source -> Destination)

Source: Public (Class) GameEngine	Target: Private playerRoster (Class) PlayerRoster

Association (direction: Source -> Destination)

Source: Public (Class) MainWindow	Target: Private ge (Class) GameEngine

Association (direction: Source -> Destination)

Source: Public (Class) AbstractController	Target: Protected ge (Class) GameEngine

Association (direction: Source -> Destination)

Source: Public (Class) PlayerMrBeanThread	Target: Private ge (Class) GameEngine

Association (direction: Source -> Destination)

Source: Public (Class) PlayerAI	Target: Private ge (Class) GameEngine

Association (direction: Source -> Destination)

Source: Public (Class) GameEngineTest	Target: Private ge (Class) GameEngine

Association (direction: Source -> Destination)

Source: Public (Class) AbstractPanel	Target: Protected ge (Class) GameEngine

OPERATIONS

checkBoard () : void Private

Method which calls the Board methods to decide if the game is Active. If there is a winner or draw, sets the gameActive value to false. This method also sets the GameRecords to the involved players.

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

OPERATIONS

◆ GameEngine () : Public

Constructor

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

◆ getBoard () : Board Public

[Is static False. Is abstract False. Is return array True. Is query False. Is synchronized False.]

◆ getGameRecord () : GameRecord Public

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

◆ getMoves () : int Public

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

◆ getPlayerO () : Player Public

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

◆ getPlayerRoster () : PlayerRoster Public

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

◆ getPlayerX () : Player Public

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

◆ getPlayInfo () : String Public

[Is static False. Is abstract False. Is return array True. Is query False. Is synchronized False.]

◆ getSign () : String Public

This method returns the sign who's turn is now. The sign is defined by the number of movement 0-9 If the number is even then the sign is X If the sign is odd the sign is O
@return String

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

◆ initRoster () : void Private

Initializes the PlayerRoster class If it is found on file system uses the file If it is not found, creates dummy players.

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

◆ isGameActive () : boolean Public

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

◆ makeDummyPlayers () : void Private

This method makes players with random scores and saves the data to the file. It also creates the Hal and Mr. Bean players.

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

◆ makeMove (row : int , col : int) : void Public

This method is the basic method to make a move it the game It takes two parameters row and col which represents the point of movement. If it is the first move, it creates an empty Board class. If it is the 2nd-9th move it gets a copy of the previous Board and makes a copy of that including the present movement.

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

◆ readyToPlay () : boolean Public

OPERATIONS

If the game has two players instances set, then it is ready to play.
@return true or false

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

setBoard (board : Board[]) : void Public

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

setGameActive (gameActive : boolean) : void Public

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

setGameRecord (gameRecord : GameRecord) : void Public

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

setMoves (moves : int) : void Public

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

setPlayerO (playerO : Player) : void Public

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

setPlayerRoster (playerRoster : PlayerRoster) : void Public

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

setPlayerX (playerX : Player) : void Public

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

setPlayInfo (playInfo : String) : void Public

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

zeroizeGame () : void Public

Initializes a new game

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

1.1.1.5.5 GameRecord

Class in package 'model'

GameRecord class, which holds each game information

GameRecord
Version 1.0 Phase 1.0 Proposed
pantz created on 9/5/2021. Last modified 9/5/2021
Extends Serializable

ATTRIBUTES

playerO : Player Private

[Is static False. Containment is Not Specified.]

playerX : Player Private

[Is static False. Containment is Not Specified.]

ATTRIBUTES

resultO : PlayerResult Private

[Is static False. Containment is Not Specified.]

resultX : PlayerResult Private

[Is static False. Containment is Not Specified.]

scoreO : float Private = 0.0f

[Is static False. Containment is Not Specified.]

scoreX : float Private = 0.0f

[Is static False. Containment is Not Specified.]

timeOfEnd : Instant Private

[Is static False. Containment is Not Specified.]

timeOfStart : Instant Private

[Is static False. Containment is Not Specified.]

ASSOCIATIONS

Association (direction: Source -> Destination)

Source: Public (Class) GameRecord

Target: Private playerX (Class) Player

Association (direction: Source -> Destination)

Source: Public (Class) GameRecord

Target: Private playerO (Class) Player

Association (direction: Source -> Destination)

Source: Public (Class) Player

Target: Private playersLastGames (Class) GameRecord

Association (direction: Source -> Destination)

Source: Public (Class) GameEngine

Target: Private gameRecord (Class) GameRecord

Association (direction: Source -> Destination)

Source: Public (Class) Player

Target: Private playersBestGames (Class) GameRecord

OPERATIONS

GameRecord () : Public

Constructor

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

getPlayerO () : Player Public

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

OPERATIONS
getPlayerX () : Player Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
getResultO () : PlayerResult Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
getResultX () : PlayerResult Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
getScoreO () : float Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
getScoreX () : float Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
getTimeOfEnd () : Instant Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
getTimeOfStart () : Instant Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
setPlayerO (playerO : Player) : void Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
setPlayerX (playerX : Player) : void Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
setResultO (resultO : PlayerResult) : void Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
setResultX (resultX : PlayerResult) : void Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
setScoreO (scoreO : float) : void Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
setScoreX (scoreX : float) : void Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
setTimeOfEnd (timeOfEnd : Instant) : void Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
setTimeOfStart (timeOfStart : Instant) : void Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

1.1.1.5.6 Player

Class in package 'model'

The tic tac toe Player class.

Player
Version 1.0 Phase 1.0 Proposed
pantz created on 9/5/2021. Last modified 9/5/2021
Extends Serializable, Cloneable

ATTRIBUTES

draws : int Private
[Is static False. Containment is Not Specified.]

losses : int Private
[Is static False. Containment is Not Specified.]

name : String Private
[Is static False. Containment is Not Specified.]

playersBestGames : GameRecord Private = new GameRecord[5]
[Is static False. Containment is Not Specified.]

playersLastGames : GameRecord Private = new GameRecord[5]
[Is static False. Containment is Not Specified.]

score : float Private
[Is static False. Containment is Not Specified.]

totalGames : int Private
[Is static False. Containment is Not Specified.]

wins : int Private
[Is static False. Containment is Not Specified.]

ASSOCIATIONS

Association (direction: Source -> Destination)

Source: Public (Class) Player

Target: Private playersLastGames (Class) GameRecord

Association (direction: Source -> Destination)

Source: Public (Class) Player

Target: Private playersBestGames (Class) GameRecord

Association (direction: Source -> Destination)

Source: Public (Class) PlayerTest

Target: Package p1 (Class) Player

Association (direction: Source -> Destination)

Source: Public (Class) PlayerRosterTest

Target: Package p5 (Class) Player

ASSOCIATIONS

Association (direction: Source -> Destination)

Source: Public (Class) PlayerRosterTest Target: Package p6 (Class) Player

Association (direction: Source -> Destination)

Source: Public (Class) PlayerRosterTest Target: Package p4 (Class) Player

Association (direction: Source -> Destination)

Source: Public (Class) FileHandlerTest Target: Package p4 (Class) Player

Association (direction: Source -> Destination)

Source: Public (Class) FileHandlerTest Target: Package p3 (Class) Player

Association (direction: Source -> Destination)

Source: Public (Class) FileHandlerTest Target: Package p2 (Class) Player

Association (direction: Source -> Destination)

Source: Public (Class) PlayerRosterTest Target: Package p3 (Class) Player

Association (direction: Source -> Destination)

Source: Public (Class) PlayerRosterTest Target: Package p1 (Class) Player

Association (direction: Source -> Destination)

Source: Public (Class) FileHandlerTest Target: Package p1 (Class) Player

Association (direction: Source -> Destination)

Source: Public (Class) GameEngine Target: Private playerO (Class) Player

Association (direction: Source -> Destination)

Source: Public (Class) GameRecord Target: Private playerX (Class) Player

Association (direction: Source -> Destination)

Source: Public (Class) GameRecord Target: Private playerO (Class) Player

Association (direction: Source -> Destination)

Source: Public (Class) PlayerTest Target: Package p2 (Class) Player

Association (direction: Source -> Destination)

Source: Public (Class) GameEngine Target: Private playerX (Class) Player

ASSOCIATIONS

Association (direction: Source -> Destination)

Source: Public (Class) PlayerRosterTest   Target: Package p2 (Class) Player

Association (direction: Source -> Destination)

Source: Public (Class) FileHandlerTest   Target: Package p5 (Class) Player

Association (direction: Source -> Destination)

Source: Public (Class) FileHandlerTest   Target: Package p6 (Class) Player

Association (direction: Source -> Destination)

Source: Public (Class) PlayerTest   Target: Package p (Class) Player

Association (direction: Source -> Destination)

Source: Public (Class) PlayerTest   Target: Package p3 (Class) Player

OPERATIONS

addBestGame (gr : GameRecord) : void Public

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

addDraw () : void Public

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

addLastGame (gr : GameRecord) : void Public

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

addLoss () : void Public

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

addWin () : void Public

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

calculateScore () : void Private

Score calculator

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

clone () : Object Public

Properties:
throws = CloneNotSupportedException

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

getDraws () : int Public

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

getLosses () : int Public

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

OPERATIONS

getName () : String Public

Getters and Setters

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

getPlayersBestGames () : GameRecord Public

[Is static False. Is abstract False. Is return array True. Is query False. Is synchronized False.]

getPlayersLastGames () : GameRecord Public

[Is static False. Is abstract False. Is return array True. Is query False. Is synchronized False.]

getScore () : float Public

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

getTotalGames () : int Public

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

getWins () : int Public

public void setTotalGames(int totalGames) { this.totalGames = totalGames; }

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

Player () : Public

Default empty constructor

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

Player (name : String) : Public

Constructor of the class, which uses the setter method

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

setDraws (draws : int) : void Public

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

setLosses (losses : int) : void Public

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

setName (name : String) : void Public

This setter removes leading and trailing spaces from the name and ensures that only the first 20 characters are kept.

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

setPlayersBestGames (playersBestGames : GameRecord[]) : void Public

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

setPlayersLastGames (playersLastGames : GameRecord[]) : void Public

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

setScore (score : float) : void Public

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

OPERATIONS

setWins (wins : int) : void Public

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

1.1.1.5.7 PlayerRoster

Class in package 'model'

PlayerRoster
Version 1.0 Phase 1.0 Proposed
pantz created on 9/5/2021. Last modified 9/5/2021
Extends Serializable

ATTRIBUTES

players : HashMap<String,Player> Private

[Is static False. Containment is Not Specified.]

ASSOCIATIONS

Association (direction: Source -> Destination)

Source: Public (Class) PlayerRosterTest Target: Package roster (Class) PlayerRoster

Association (direction: Source -> Destination)

Source: Public (Class) FileHandlerTest Target: Package roster (Class) PlayerRoster

Association (direction: Source -> Destination)

Source: Public (Class) GameEngine Target: Private playerRoster (Class) PlayerRoster

OPERATIONS

addPlayer (p : Player) : void Public

This method adds a player to the HashMap<String,Player>. Before saving, checks if the given name already exists.

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

findHallOfFame (n : int) : Player Public

This method returns the best n players of the Players Roster.
@return Player[]

[Is static False. Is abstract False. Is return array True. Is query False. Is synchronized False.]

findPlayerByName (name : String) : Player Public

Finds a player by its name.
@return Player

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

findPlayerNames () : ArrayList<String> Public

This method returns an array with all the names of players.

OPERATIONS
@return ArrayList<String> [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ PlayerRoster () : Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

1.1.1.6 utils

Package in package 'tictactoe'

utils
Version 1.0 Phase 1.0 Proposed
pantz created on 9/5/2021. Last modified 9/5/2021

1.1.1.6.1 utils diagram

Class diagram in package 'utils'

utils
Version 1.0
pantz created on 9/5/2021. Last modified 9/5/2021

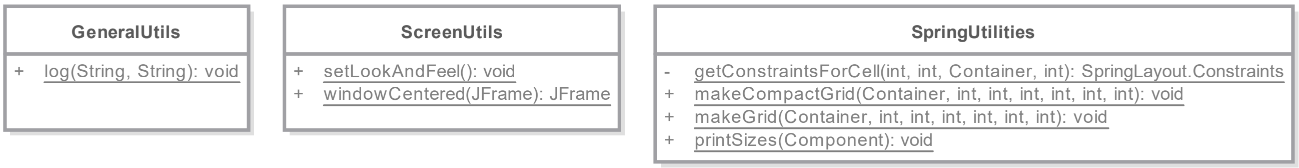
GeneralUtils
+ log(String, String): void
ScreenUtils
+ setLookAndFeel(): void
+ windowCentered(JFrame): JFrame
SpringUtilities
- getConstraintsForCell(int, int, Container, int): SpringLayout.Constraints
+ makeCompactGrid(Container, int, int, int, int, int, int): void
+ makeGrid(Container, int, int, int, int, int, int): void
+ printSizes(Component): void

Figure 7: utils

1.1.1.6.2 GeneralUtils

Class in package 'utils'

GeneralUtils
Version 1.0 Phase 1.0 Proposed
pantz created on 9/5/2021. Last modified 9/5/2021

OPERATIONS
log (className : String , log : String) : void Public [Is static True. Is abstract False. Is return array False. Is query False. Is synchronized False.]

1.1.1.6.3 ScreenUtils

Class in package 'utils'

ScreenUtils
Version 1.0 Phase 1.0 Proposed
pantz created on 9/5/2021. Last modified 9/5/2021

OPERATIONS
setLookAndFeel () : void Public [Is static True. Is abstract False. Is return array False. Is query False. Is synchronized False.]
windowCentered (frame : JFrame) : JFrame Public [Is static True. Is abstract False. Is return array False. Is query False. Is synchronized False.]

1.1.1.6.4 SpringUtilities

Class in package 'utils'

A 1.4 file that provides utility methods for creating form- or grid-style layouts with SpringLayout. These utilities are used by several programs, such as SpringBox and SpringCompactGrid.

SpringUtilities
Version 1.0 Phase 1.0 Proposed
pantz created on 9/5/2021. Last modified 9/5/2021

OPERATIONS

◆ getConstraintsForCell (row : int , col : int , parent : Container , cols : int) : SpringLayout.Constraints Private

Used by makeCompactGrid.

[Is static True. Is abstract False. Is return array False. Is query False. Is synchronized False.]

◆ makeCompactGrid (parent : Container , rows : int , cols : int , initialX : int , initialY : int , xPad : int , yPad : int) : void Public

Aligns the first <code>rows</code> * <code>cols</code> components of <code>parent</code> in a grid. Each component in a column is as wide as the maximum preferred width of the components in that column; height is similarly determined for each row. The parent is made just big enough to fit them all.

[Is static True. Is abstract False. Is return array False. Is query False. Is synchronized False.]

◆ makeGrid (parent : Container , rows : int , cols : int , initialX : int , initialY : int , xPad : int , yPad : int) : void Public

Aligns the first <code>rows</code> * <code>cols</code> components of <code>parent</code> in a grid. Each component is as big as the maximum preferred width and height of the components. The parent is made just big enough to fit them all.

[Is static True. Is abstract False. Is return array False. Is query False. Is synchronized False.]

◆ printSizes (c : Component) : void Public

A debugging utility that prints to stdout the component's minimum, preferred, and maximum sizes.

[Is static True. Is abstract False. Is return array False. Is query False. Is synchronized False.]

1.1.1.7 view

Package in package 'tictactoe'

view
Version 1.0 Phase 1.0 Proposed
pantz created on 9/5/2021. Last modified 9/5/2021

1.1.1.7.1 view diagram

Class diagram in package 'view'

view
Version 1.0
pantz created on 9/5/2021. Last modified 9/5/2021

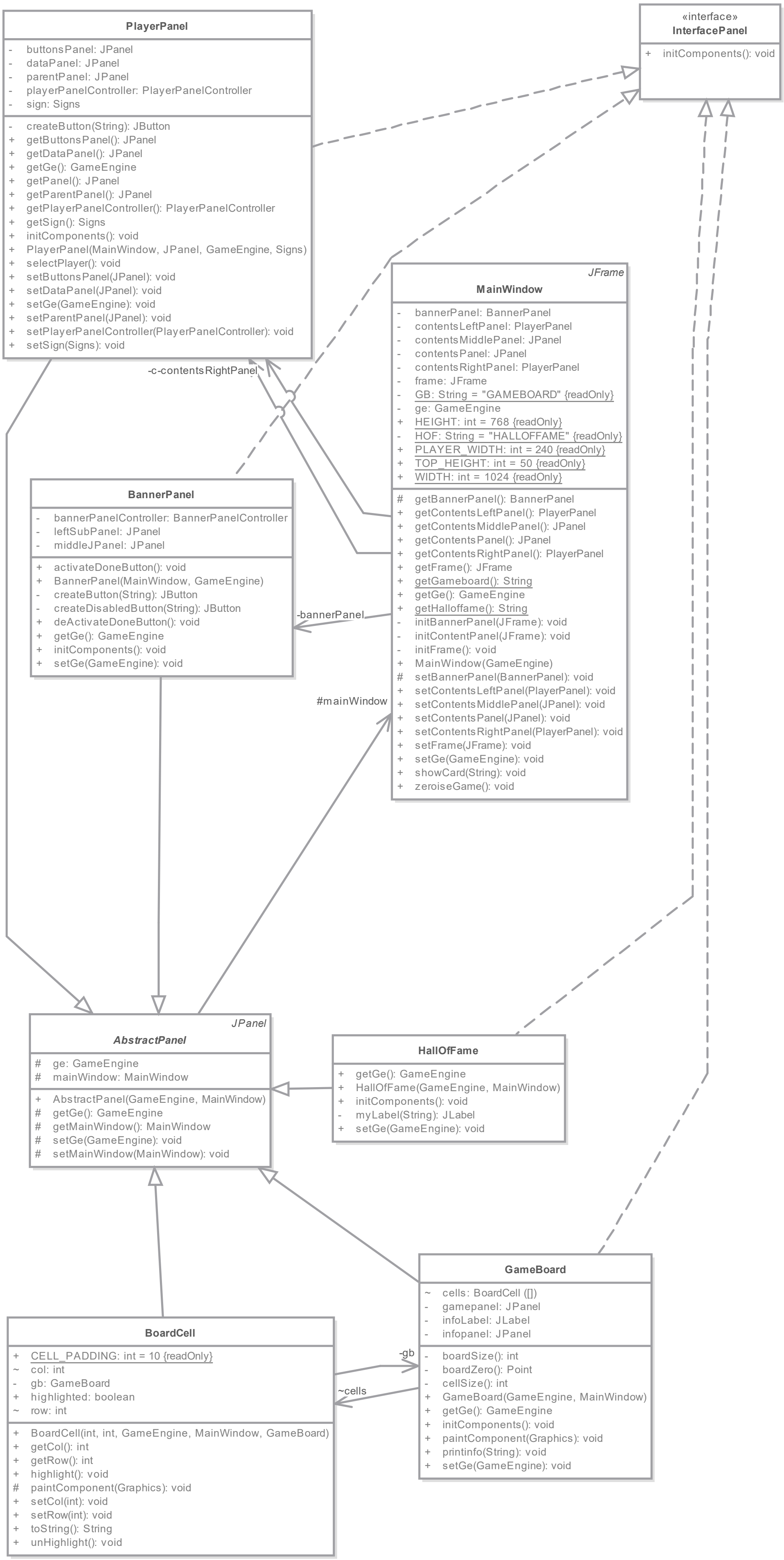
PlayerPanel
- buttonsPanel: JPanel
- dataPanel: JPanel
- parentPanel: JPanel
- playerPanelController: PlayerPanelController
- sign: Signs
- createButton(String): JButton
+ getButtonsPanel(): JPanel
+ getDataPanel(): JPanel
+ getGe(): GameEngine
+ getPanel(): JPanel
+ getParentPanel(): JPanel
+ getPlayerPanelController(): PlayerPanelController
+ getSign(): Signs
+ initComponents(): void
+ PlayerPanel(MainWindow, JPanel, GameEngine, Signs)
+ selectPlayer(): void
+ setButtonsPanel(JPanel): void
+ setDataPanel(JPanel): void
+ setGe(GameEngine): void
+ setParentPanel(JPanel): void
+ setPlayerPanelController(PlayerPanelController): void
+ setSign(Signs): void
«interface»
InterfacePanel
+ initComponents(): void
-c-contentsRightPanel
JFrame
MainWindow
- bannerPanel: BannerPanel
- contentsLeftPanel: PlayerPanel
- contentsMiddlePanel: JPanel
- contentsPanel: JPanel
- contentsRightPanel: PlayerPanel
- frame: JFrame
- GB: String = "GAMEBOARD" {readOnly}
- ge: GameEngine
+ HEIGHT: int = 768 {readOnly}
- HOF: String = "HALLOFFAME" {readOnly}
+ PLAYER_WIDTH: int = 240 {readOnly}
+ TOP_HEIGHT: int = 50 {readOnly}
+ WIDTH: int = 1024 {readOnly}
# getBannerPanel(): BannerPanel
+ getContentsLeftPanel(): PlayerPanel
+ getContentsMiddlePanel(): JPanel
+ getContentsPanel(): JPanel
+ getContentsRightPanel(): PlayerPanel
+ getFrame(): JFrame
+ getGameboard(): String
+ getGe(): GameEngine
+ getHalloffame(): String
- initBannerPanel(JFrame): void
- initContentPanel(JFrame): void
- initFrame(): void
+ MainWindow(GameEngine)
# setBannerPanel(BannerPanel): void
+ setContentsLeftPanel(PlayerPanel): void
+ setContentsMiddlePanel(JPanel): void
+ setContentsPanel(JPanel): void
+ setContentsRightPanel(PlayerPanel): void
+ setFrame(JFrame): void
+ setGe(GameEngine): void
+ showCard(String): void
+ zeroiseGame(): void
BannerPanel
- bannerPanelController: BannerPanelController
- leftSubPanel: JPanel
- middleJPanel: JPanel
+ activateDoneButton(): void
+ BannerPanel(MainWindow, GameEngine)
- createButton(String): JButton
- createDisabledButton(String): JButton
+ deActivateDoneButton(): void
+ getGe(): GameEngine
+ initComponents(): void
+ setGe(GameEngine): void
-bannerPanel
#mainWindow
JPanel
AbstractPanel
# ge: GameEngine
# mainWindow: MainWindow
+ AbstractPanel(GameEngine, MainWindow)
# getGe(): GameEngine
# getMainWindow(): MainWindow
# setGe(GameEngine): void
# setMainWindow(MainWindow): void
HallOfFame
+ getGe(): GameEngine
+ HallOfFame(GameEngine, MainWindow)
+ initComponents(): void
- myLabel(String): JLabel
+ setGe(GameEngine): void
GameBoard
~ cells: BoardCell ([])
- gamepanel: JPanel
- infoLabel: JLabel
- infopanel: JPanel
- boardSize(): int
- boardZero(): Point
- cellSize(): int
+ GameBoard(GameEngine, MainWindow)
+ getGe(): GameEngine
+ initComponents(): void
+ paintComponent(Graphics): void
+ printinfo(String): void
+ setGe(GameEngine): void
-gb
~cells
BoardCell
+ CELL_PADDING: int = 10 {readOnly}
~ col: int
- gb: GameBoard
+ highlighted: boolean
~ row: int
+ BoardCell(int, int, GameEngine, MainWindow, GameBoard)
+ getCol(): int
+ getRow(): int
+ highlight(): void
# paintComponent(Graphics): void
+ setCol(int): void
+ setRow(int): void
+ toString(): String
+ unHighlight(): void

Figure 8: view

1.1.1.7.2 AbstractPanel

Class in package 'view'

This is an Abstract Class which inherits all the attributes of a JPanel. All methods have access modifier protected in order to be used only by classes which extend this class.

AbstractPanel
Version 1.0 Phase 1.0 Proposed
pantz created on 9/5/2021. Last modified 9/5/2021
Extends JPanel

INCOMING STRUCTURAL RELATIONSHIPS	
Generalization from HallOfFame to AbstractPanel	[Direction is 'Source -> Destination'.]
Generalization from PlayerPanel to AbstractPanel	[Direction is 'Source -> Destination'.]
Generalization from BannerPanel to AbstractPanel	[Direction is 'Source -> Destination'.]
Generalization from BoardCell to AbstractPanel	[Direction is 'Source -> Destination'.]
Generalization from GameBoard to AbstractPanel	[Direction is 'Source -> Destination'.]

ATTRIBUTES	
ge : GameEngine Protected	[Is static False. Containment is Not Specified.]
mainWindow : MainWindow Protected	[Is static False. Containment is Not Specified.]

ASSOCIATIONS	
Association (direction: Source -> Destination)	
Source: Public (Class) AbstractPanel	Target: Protected ge (Class) GameEngine
Association (direction: Source -> Destination)	
Source: Public (Class) AbstractPanel	Target: Protected mainWindow (Class) MainWindow

OPERATIONS

OPERATIONS

◆ AbstractPanel (ge : GameEngine , mainWindow : MainWindow) : Public

Constructor with one argument of GameEngine instance.

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

◆ getGe () : GameEngine Protected

Getters and Setters.

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

◆ getMainWindow () : MainWindow Protected

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

◆ setGe (ge : GameEngine) : void Protected

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

◆ setMainWindow (mainWindow : MainWindow) : void Protected

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

1.1.1.7.3 BannerPanel

Class in package 'view'

BannerPanel
Version 1.0 Phase 1.0 Proposed
pantz created on 9/5/2021. Last modified 9/5/2021
Extends AbstractPanel

OUTGOING STRUCTURAL RELATIONSHIPS

Generalization from BannerPanel to AbstractPanel

[Direction is 'Source -> Destination'.]

Realization from BannerPanel to InterfacePanel

[Direction is 'Source -> Destination'.]

ATTRIBUTES

◆ bannerPanelController : BannerPanelController Private

[Is static False. Containment is Not Specified.]

◆ leftSubPanel : JPanel Private

[Is static False. Containment is Not Specified.]

◆ middleJPanel : JPanel Private

[Is static False. Containment is Not Specified.]

ASSOCIATIONS

Association (direction: Source -> Destination)

ASSOCIATIONS

Source: Public (Class) BannerPanel

Target: Private bannerPanelController (Class) BannerPanelController

Association (direction: Source -> Destination)

Source: Public (Class) MainWindow

Target: Private bannerPanel (Class) BannerPanel

OPERATIONS

activateDoneButton () : void Public
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

BannerPanel (parentFrame : MainWindow , ge : GameEngine) : Public
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

createButton (text : String) : JButton Private
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

createDisabledButton (text : String) : JButton Private
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

deActivateDoneButton () : void Public
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

getGe () : GameEngine Public
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

initComponents () : void Public

Properties:
annotations = @Override
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

setGe (ge : GameEngine) : void Public
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

1.1.1.7.4 BoardCell

Class in package 'view'

BoardCell
Version 1.0 Phase 1.0 Proposed
pantz created on 9/5/2021. Last modified 9/5/2021
Extends AbstractPanel

OUTGOING STRUCTURAL RELATIONSHIPS

Generalization from BoardCell to AbstractPanel
[Direction is 'Source -> Destination'.]

ATTRIBUTES

ATTRIBUTES

CELL_PADDING : int Public Const = 10
[Is static True. Containment is Not Specified.]

col : int Package
[Is static False. Containment is Not Specified.]

gb : GameBoard Private
[Is static False. Containment is Not Specified.]

highlighted : boolean Public
[Is static False. Containment is Not Specified.]

row : int Package
[Is static False. Containment is Not Specified.]

ASSOCIATIONS

Association (direction: Source -> Destination)

Source: Public (Class) BoardCell Target: Private gb (Class) GameBoard

Association (direction: Source -> Destination)

Source: Public (Class) GameBoard Target: Package cells (Class) BoardCell

Association (direction: Source -> Destination)

Source: Public (Class) BoardCellController Target: Private cell (Class) BoardCell

OPERATIONS

BoardCell (row : int , col : int , ge : GameEngine , mainWindow : MainWindow , gb : GameBoard) : Public
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

getCol () : int Public
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

getRow () : int Public
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

highlight () : void Public
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

paintComponent (g : Graphics) : void Protected

Properties:
annotations = @Override
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

setCol (col : int) : void Public
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

OPERATIONS

setRow (row : int) : void Public

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

toString () : String Public

Properties:
annotations = @Override

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

unHighlight () : void Public

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

1.1.1.7.5 GameBoard

Class in package 'view'

GameBoard
Version 1.0 Phase 1.0 Proposed
pantz created on 9/5/2021. Last modified 9/5/2021
Extends AbstractPanel

OUTGOING STRUCTURAL RELATIONSHIPS

Generalization from GameBoard to AbstractPanel

[Direction is 'Source -> Destination'.]

Realization from GameBoard to InterfacePanel

[Direction is 'Source -> Destination'.]

ATTRIBUTES

cells : BoardCell Package

[Is static False. Containment is Not Specified.]

gamepanel : JPanel Private

[Is static False. Containment is Not Specified.]

infoLabel : JLabel Private

[Is static False. Containment is Not Specified.]

infopanel : JPanel Private

[Is static False. Containment is Not Specified.]

ASSOCIATIONS

Association (direction: Source -> Destination)

Source: Public (Class) GameBoard Target: Package cells (Class) BoardCell

Association (direction: Source -> Destination)

ASSOCIATIONS

Source: Public (Class) BoardCell Target: Private gb (Class) GameBoard

Association (direction: Source -> Destination)

Source: Public (Class) BoardCellController Target: Private gb (Class) GameBoard

OPERATIONS

boardSize () : int Private

Returns the internal size of the real board

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

boardZero () : Point Private

Returns the upper left corner of the board

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

cellSize () : int Private

Computes the cell size along with paddings (min Panel dimension / 5)

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

GameBoard (ge : GameEngine , mainWindow : MainWindow) : Public

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

getGe () : GameEngine Public

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

initComponents () : void Public

Properties:
annotations = @Override

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

paintComponent (g : Graphics) : void Public

Properties:
annotations = @Override

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

printinfo (info : String) : void Public

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

setGe (ge : GameEngine) : void Public

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

1.1.1.7.6 HallOfFame

Class in package 'view'

HallOfFame

Version 1.0 Phase 1.0 Proposed
pantz created on 9/5/2021. Last modified 9/5/2021
Extends AbstractPanel

OUTGOING STRUCTURAL RELATIONSHIPS
Generalization from HallOfFame to AbstractPanel [Direction is 'Source -> Destination'.]
Realization from HallOfFame to InterfacePanel [Direction is 'Source -> Destination'.]

OPERATIONS
getGe () : GameEngine Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
HallOfFame (ge : GameEngine , mainWindow : MainWindow) : Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
initComponents () : void Public Properties: annotations = @Override [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
myLabel (text : String) : JLabel Private [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
setGe (ge : GameEngine) : void Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

1.1.1.7.7 MainWindow

Class in package 'view'

MainWindow
Version 1.0 Phase 1.0 Proposed
pantz created on 9/5/2021. Last modified 9/5/2021
Extends JFrame

ATTRIBUTES
bannerPanel : BannerPanel Private [Is static False. Containment is Not Specified.]
contentsLeftPanel : PlayerPanel Private [Is static False. Containment is Not Specified.]
contentsMiddlePanel : JPanel Private [Is static False. Containment is Not Specified.]
contentsPanel : JPanel Private

ATTRIBUTES

[Is static False. Containment is Not Specified.]

contentsRightPanel : PlayerPanel Private

[Is static False. Containment is Not Specified.]

frame : JFrame Private

[Is static False. Containment is Not Specified.]

GB : String Private Const = "GAMEBOARD"

[Is static True. Containment is Not Specified.]

ge : GameEngine Private

[Is static False. Containment is Not Specified.]

HEIGHT : int Public Const = 768

[Is static True. Containment is Not Specified.]

HOF : String Private Const = "HALLOFFAME"

[Is static True. Containment is Not Specified.]

PLAYER_WIDTH : int Public Const = 240

[Is static True. Containment is Not Specified.]

TOP_HEIGHT : int Public Const = 50

[Is static True. Containment is Not Specified.]

WIDTH : int Public Const = 1024

[Is static True. Containment is Not Specified.]

ASSOCIATIONS

Association (direction: Source -> Destination)

Source: Public (Class) MainWindow

Target: Private bannerPanel (Class) BannerPanel

Association (direction: Source -> Destination)

Source: Public (Class) MainWindow

Target: Private ge (Class) GameEngine

Association (direction: Source -> Destination)

Source: Public (Class) MainWindow

Target: Private contentsRightPanel (Class) PlayerPanel

Association (direction: Source -> Destination)

Source: Public (Class) MainWindow

Target: Private contentsLeftPanel (Class) PlayerPanel

Association (direction: Source -> Destination)

Source: Public (Class) PlayerPanelController

Target: Private parentFrame (Class) MainWindow

ASSOCIATIONS

Association (direction: Source -> Destination)

Source: Public (Class) BannerPanelController

Target: Private mainWindow (Class) MainWindow

Association (direction: Source -> Destination)

Source: Public (Class) AbstractPanel

Target: Protected mainWindow (Class) MainWindow

OPERATIONS

getBannerPanel () : BannerPanel Protected
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

getContentsLeftPanel () : PlayerPanel Public
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

getContentsMiddlePanel () : JPanel Public
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

getContentsPanel () : JPanel Public
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

getContentsRightPanel () : PlayerPanel Public
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

getFrame () : JFrame Public

Getters And Setters
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

getGameboard () : String Public
[Is static True. Is abstract False. Is return array False. Is query False. Is synchronized False.]

getGe () : GameEngine Public
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

getHalloffame () : String Public
[Is static True. Is abstract False. Is return array False. Is query False. Is synchronized False.]

initBannerPanel (frame : JFrame) : void Private
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

initContentPanel (frame : JFrame) : void Private
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

initFrame () : void Private
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

MainWindow (ge : GameEngine) : Public

OPERATIONS

Properties:
throws = HeadlessException
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

setBannerPanel (bannerPanel : BannerPanel) : void Protected
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

setContentsLeftPanel (contentsLeftPanel : PlayerPanel) : void Public
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

setContentsMiddlePanel (contentsMiddlePanel : JPanel) : void Public
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

setContentsPanel (contentsPanel : JPanel) : void Public
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

setContentsRightPanel (contentsRightPanel : PlayerPanel) : void Public
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

setFrame (frame : JFrame) : void Public
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

setGe (ge : GameEngine) : void Public
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

showCard (cardname : String) : void Public
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

zeroiseGame () : void Public
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

1.1.1.7.8 PlayerPanel

Class in package 'view'

PlayerPanel
Version 1.0 Phase 1.0 Proposed
pantz created on 9/5/2021. Last modified 9/5/2021
Extends AbstractPanel

OUTGOING STRUCTURAL RELATIONSHIPS

Generalization from PlayerPanel to AbstractPanel
[Direction is 'Source -> Destination'.]

Realization from PlayerPanel to InterfacePanel
[Direction is 'Source -> Destination'.]

ATTRIBUTES

ATTRIBUTES

buttonsPanel : JPanel Private

[Is static False. Containment is Not Specified.]

dataPanel : JPanel Private

[Is static False. Containment is Not Specified.]

parentPanel : JPanel Private

[Is static False. Containment is Not Specified.]

playerPanelController : PlayerPanelController Private

[Is static False. Containment is Not Specified.]

sign : Signs Private

[Is static False. Containment is Not Specified.]

ASSOCIATIONS

Association (direction: Source -> Destination)

Source: Public (Class) PlayerPanel	Target: Private playerPanelController (Class) PlayerPanelController

Association (direction: Source -> Destination)

Source: Public (Class) PlayerPanelController	Target: Private parentPanel (Class) PlayerPanel

Association (direction: Source -> Destination)

Source: Public (Class) MainWindow	Target: Private contentsRightPanel (Class) PlayerPanel

Association (direction: Source -> Destination)

Source: Public (Class) MainWindow	Target: Private contentsLeftPanel (Class) PlayerPanel

OPERATIONS

createButton (text : String) : JButton Private

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

getButtonsPanel () : JPanel Public

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

getDataPanel () : JPanel Public

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

getGe () : GameEngine Public

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

getPanel () : JPanel Public

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

OPERATIONS
getParentPanel () : JPanel Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
getPlayerPanelController () : PlayerPanelController Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
getSign () : Signs Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
initComponents () : void Public Properties: annotations = @Override [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
PlayerPanel (mainWindow : MainWindow , parentPanel : JPanel , ge : GameEngine , sign : Signs) : Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
selectPlayer () : void Public Properties: throws = CloneNotSupportedException [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
setButtonsPanel (buttonsPanel : JPanel) : void Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
setDataPanel (dataPanel : JPanel) : void Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
setGe (ge : GameEngine) : void Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
setParentPanel (parentPanel : JPanel) : void Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
setPlayerPanelController (playerPanelController : PlayerPanelController) : void Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
setSign (sign : Signs) : void Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

1.1.1.7.9 InterfacePanel

Interface in package 'view'

InterfacePanel
Version 1.0 Phase 1.0 Proposed
pantz created on 9/5/2021. Last modified 9/5/2021

INCOMING STRUCTURAL RELATIONSHIPS
Realization from PlayerPanel to InterfacePanel [Direction is 'Source -> Destination'.]
Realization from HallOfFame to InterfacePanel [Direction is 'Source -> Destination'.]
Realization from BannerPanel to InterfacePanel [Direction is 'Source -> Destination'.]
Realization from GameBoard to InterfacePanel [Direction is 'Source -> Destination'.]

OPERATIONS
initComponents () : void Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

1.1.1.8 Main

Class in package 'tictactoe'

This application is a Java GUI (Swing Based) application for tic-tac-toe game. This is the entry point (main) of the whole application. All comments are in English language because I hate greeklish. Sorry...

Main
Version 1.0 Phase 1.0 Proposed
pantz created on 9/5/2021. Last modified 9/5/2021

OPERATIONS
main (args : String[]) : void Public Main method [Is static True. Is abstract False. Is return array False. Is query False. Is synchronized False.]

1.2 test

Package in package 'Class Model'

test
Version 1.0 Phase 1.0 Proposed
pantz created on 9/5/2021. Last modified 9/5/2021

1.2.1 test diagram

Class diagram in package 'test'

test
Version 1.0
pantz created on 9/5/2021. Last modified 9/5/2021

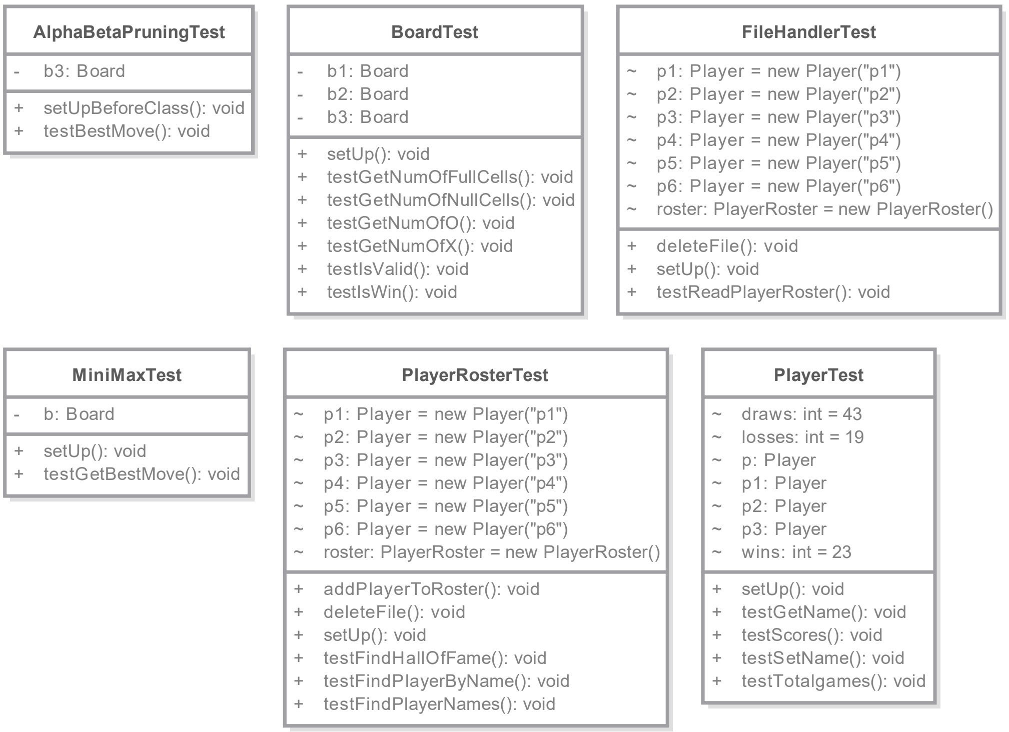
AlphaBetaPruningTest
- b3: Board
+ setUpBeforeClass(): void
+ testBestMove(): void
BoardTest
- b1: Board
- b2: Board
- b3: Board
+ setUp(): void
+ testGetNumOfFullCells(): void
+ testGetNumOfNullCells(): void
+ testGetNumOfO(): void
+ testGetNumOfX(): void
+ testIsValid(): void
+ testIsWin(): void
FileHandlerTest
~ p1: Player = new Player("p1")
~ p2: Player = new Player("p2")
~ p3: Player = new Player("p3")
~ p4: Player = new Player("p4")
~ p5: Player = new Player("p5")
~ p6: Player = new Player("p6")
~ roster: PlayerRoster = new PlayerRoster()
+ deleteFile(): void
+ setUp(): void
+ testReadPlayerRoster(): void
GameEngineTest
- ge: GameEngine
+ setUpBeforeClass(): void
+ testMakeMove(): void
MiniMaxTest
- b: Board
+ setUp(): void
+ testGetBestMove(): void
PlayerRosterTest
~ p1: Player = new Player("p1")
~ p2: Player = new Player("p2")
~ p3: Player = new Player("p3")
~ p4: Player = new Player("p4")
~ p5: Player = new Player("p5")
~ p6: Player = new Player("p6")
~ roster: PlayerRoster = new PlayerRoster()
+ addPlayerToRoster(): void
+ deleteFile(): void
+ setUp(): void
+ testFindHallOfFame(): void
+ testFindPlayerByName(): void
+ testFindPlayerNames(): void
PlayerTest
~ draws: int = 43
~ losses: int = 19
~ p: Player
~ p1: Player
~ p2: Player
~ p3: Player
~ wins: int = 23
+ setUp(): void
+ testGetName(): void
+ testScores(): void
+ testSetName(): void
+ testTotalgames(): void

Figure 9: test

1.2.2 AlphaBetaPruningTest

Class in package 'test'

AlphaBetaPruningTest
Version 1.0 Phase 1.0 Proposed
pantz created on 9/5/2021. Last modified 9/5/2021

ATTRIBUTES

b3 : Board Private

[Is static False. Containment is Not Specified.]

ASSOCIATIONS	
Association (direction: Source -> Destination)	
Source: Public (Class) AlphaBetaPruningTest	Target: Private b3 (Class) Board

OPERATIONS
setUpBeforeClass () : void Public Properties: annotations = @Before throws = Exception [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
testBestMove () : void Public Properties: annotations = @Test throws = CloneNotSupportedException [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

1.2.3 BoardTest

Class in package 'test'

BoardTest
Version 1.0 Phase 1.0 Proposed
pantz created on 9/5/2021. Last modified 9/5/2021

ATTRIBUTES
b1 : Board Private [Is static False. Containment is Not Specified.]
b2 : Board Private [Is static False. Containment is Not Specified.]
b3 : Board Private [Is static False. Containment is Not Specified.]

ASSOCIATIONS	
Association (direction: Source -> Destination)	
Source: Public (Class) BoardTest	Target: Private b3 (Class) Board
Association (direction: Source -> Destination)	
Source: Public (Class) BoardTest	Target: Private b2 (Class) Board
Association (direction: Source -> Destination)	
Source: Public (Class) BoardTest	Target: Private b1 (Class) Board

ASSOCIATIONS

OPERATIONS

setUp () : void Public

Properties:
annotations = @Before
throws = Exception

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

testGetNumOfFullCells () : void Public

Properties:
annotations = @Test

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

testGetNumOfNullCells () : void Public

Properties:
annotations = @Test

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

testGetNumOfO () : void Public

Properties:
annotations = @Test

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

testGetNumOfX () : void Public

Properties:
annotations = @Test

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

testIsValid () : void Public

Properties:
annotations = @Test

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

testIsWin () : void Public

Properties:
annotations = @Test

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

1.2.4 FileHandlerTest

Class in package 'test'

FileHandlerTest
Version 1.0 Phase 1.0 Proposed
pantz created on 9/5/2021. Last modified 9/5/2021

ATTRIBUTES

ATTRIBUTES

p1 : Player Package = new Player("p1")

[Is static False. Containment is Not Specified.]

p2 : Player Package = new Player("p2")

[Is static False. Containment is Not Specified.]

p3 : Player Package = new Player("p3")

[Is static False. Containment is Not Specified.]

p4 : Player Package = new Player("p4")

[Is static False. Containment is Not Specified.]

p5 : Player Package = new Player("p5")

[Is static False. Containment is Not Specified.]

p6 : Player Package = new Player("p6")

[Is static False. Containment is Not Specified.]

roster : PlayerRoster Package = new PlayerRoster()

[Is static False. Containment is Not Specified.]

ASSOCIATIONS

Association (direction: Source -> Destination)

Source: Public (Class) FileHandlerTest Target: Package p4 (Class) Player

Association (direction: Source -> Destination)

Source: Public (Class) FileHandlerTest Target: Package p3 (Class) Player

Association (direction: Source -> Destination)

Source: Public (Class) FileHandlerTest Target: Package p2 (Class) Player

Association (direction: Source -> Destination)

Source: Public (Class) FileHandlerTest Target: Package p1 (Class) Player

Association (direction: Source -> Destination)

Source: Public (Class) FileHandlerTest Target: Package roster (Class) PlayerRoster

Association (direction: Source -> Destination)

Source: Public (Class) FileHandlerTest Target: Package p5 (Class) Player

Association (direction: Source -> Destination)

Source: Public (Class) FileHandlerTest Target: Package p6 (Class) Player

OPERATIONS

deleteFile () : void Public

Properties:
annotations = @After

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

setUp () : void Public

Properties:
annotations = @Before
throws = Exception

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

testReadPlayerRoster () : void Public

Properties:
annotations = @Test

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

1.2.5 GameEngineTest

Class in package 'test'

GameEngineTest
Version 1.0 Phase 1.0 Proposed
pantz created on 9/5/2021. Last modified 9/5/2021

ATTRIBUTES

ge : GameEngine Private

[Is static False. Containment is Not Specified.]

ASSOCIATIONS

Association (direction: Source -> Destination)

Source: Public (Class) GameEngineTest
Target: Private ge (Class) GameEngine

OPERATIONS

setUpBeforeClass () : void Public

Properties:
annotations = @Before
throws = Exception

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

testMakeMove () : void Public

Properties:
annotations = @Test

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

1.2.6 MiniMaxTest

Class in package 'test'

MiniMaxTest
Version 1.0 Phase 1.0 Proposed
pantz created on 9/5/2021. Last modified 9/5/2021

ATTRIBUTES

b : Board Private

[Is static False. Containment is Not Specified.]

ASSOCIATIONS

Association (direction: Source -> Destination)

Source: Public (Class) MiniMaxTest Target: Private b (Class) Board

OPERATIONS

setUp () : void Public

Properties:
annotations = @Before
throws = Exception

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

testGetBestMove () : void Public

Properties:
annotations = @Test

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

1.2.7 PlayerRosterTest

Class in package 'test'

PlayerRosterTest
Version 1.0 Phase 1.0 Proposed
pantz created on 9/5/2021. Last modified 9/5/2021

ATTRIBUTES

p1 : Player Package = new Player("p1")

[Is static False. Containment is Not Specified.]

p2 : Player Package = new Player("p2")

[Is static False. Containment is Not Specified.]

p3 : Player Package = new Player("p3")

[Is static False. Containment is Not Specified.]

ATTRIBUTES

p4 : Player Package = new Player("p4")

[Is static False. Containment is Not Specified.]

p5 : Player Package = new Player("p5")

[Is static False. Containment is Not Specified.]

p6 : Player Package = new Player("p6")

[Is static False. Containment is Not Specified.]

roster : PlayerRoster Package = new PlayerRoster()

[Is static False. Containment is Not Specified.]

ASSOCIATIONS

Association (direction: Source -> Destination)

Source: Public (Class) PlayerRosterTest Target: Package roster (Class) PlayerRoster

Association (direction: Source -> Destination)

Source: Public (Class) PlayerRosterTest Target: Package p5 (Class) Player

Association (direction: Source -> Destination)

Source: Public (Class) PlayerRosterTest Target: Package p6 (Class) Player

Association (direction: Source -> Destination)

Source: Public (Class) PlayerRosterTest Target: Package p4 (Class) Player

Association (direction: Source -> Destination)

Source: Public (Class) PlayerRosterTest Target: Package p3 (Class) Player

Association (direction: Source -> Destination)

Source: Public (Class) PlayerRosterTest Target: Package p1 (Class) Player

Association (direction: Source -> Destination)

Source: Public (Class) PlayerRosterTest Target: Package p2 (Class) Player

OPERATIONS

addPlayerToRoster () : void Public

Properties:
annotations = @Test

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

deleteFile () : void Public

Properties:

OPERATIONS

annotations = @After
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

◆ setUp () : void Public

Properties:
annotations = @Before
throws = Exception
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

◆ testFindHallOfFame () : void Public

Properties:
annotations = @Test
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

◆ testFindPlayerByName () : void Public

Properties:
annotations = @Test
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

◆ testFindPlayerNames () : void Public

Properties:
annotations = @Test
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

1.2.8 PlayerTest

Class in package 'test'

PlayerTest
Version 1.0 Phase 1.0 Proposed
pantz created on 9/5/2021. Last modified 9/5/2021

ATTRIBUTES

◆ draws : int Package = 43
[Is static False. Containment is Not Specified.]

◆ losses : int Package = 19
[Is static False. Containment is Not Specified.]

◆ p : Player Package
[Is static False. Containment is Not Specified.]

◆ p1 : Player Package
[Is static False. Containment is Not Specified.]

◆ p2 : Player Package
[Is static False. Containment is Not Specified.]

ATTRIBUTES

p3 : Player Package

[Is static False. Containment is Not Specified.]

wins : int Package = 23

[Is static False. Containment is Not Specified.]

ASSOCIATIONS

Association (direction: Source -> Destination)

Source: Public (Class) PlayerTest Target: Package p1 (Class) Player

Association (direction: Source -> Destination)

Source: Public (Class) PlayerTest Target: Package p2 (Class) Player

Association (direction: Source -> Destination)

Source: Public (Class) PlayerTest Target: Package p (Class) Player

Association (direction: Source -> Destination)

Source: Public (Class) PlayerTest Target: Package p3 (Class) Player

OPERATIONS

setUp () : void Public

Properties:
annotations = @Before
throws = Exception

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

testGetName () : void Public

Properties:
annotations = @Test

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

testScores () : void Public

Properties:
annotations = @Test

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

testSetName () : void Public

Properties:
annotations = @Test

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

testTotalgames () : void Public

Properties:
annotations = @Test

OPERATIONS
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]