Model Report

Class Model

Version • Proposed

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 3/5/2021. Last modified 3/5/2021

1.1.1 tictactoe

Package in package 'main'

tictactoe
Version 1.0 Phase 1.0 Proposed
pantz created on 3/5/2021. Last modified 3/5/2021

1.1.1.1   tictactoe diagram

Class diagram in package 'tictactoe'

tictactoe
Version 1.0
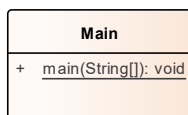pantz created on 3/5/2021. Last modified 3/5/2021

Figure 1: tictactoe

1.1.1.2   controller

Package in package 'tictactoe'

controller
Version 1.0 Phase 1.0 Proposed
pantz created on 3/5/2021. Last modified 3/5/2021

1.1.1.2.1   controller diagram

Class diagram in package 'controller'

controller
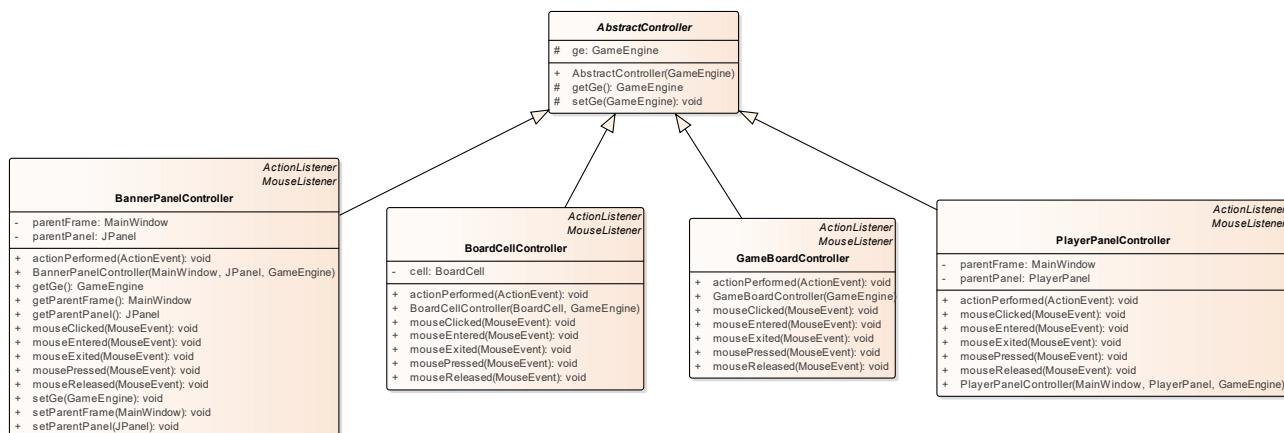
Figure 2: controller

1.1.1.2.2   AbstractController

Class in package 'controller'

This is an abstract controller class which gets the GameEngine instance. All classes extend this class are getting the GameEngine object instantiated

INCOMING STRUCTURAL RELATIONSHIPS

➡ Generalization from BannerPanelController to AbstractController

[Direction is 'Source -> Destination'.]

➡ Generalization from PlayerPanelController to AbstractController

[Direction is 'Source -> Destination'.]

➡ Generalization from BoardCellController to AbstractController

[Direction is 'Source -> Destination'.]

➡ Generalization from GameBoardController 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) : Public

[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.]

1.1.1.2.3   BannerPanelController

Class in package 'controller'

BannerPanelController
Version 1.0 Phase 1.0 Proposed
pantz created on 3/5/2021. Last modified 3/5/2021
Extends AbstractController, ActionListener, MouseListener

OUTGOING STRUCTURAL RELATIONSHIPS

⬅ Generalization from BannerPanelController to AbstractController

[Direction is 'Source -> Destination'.]

ATTRIBUTES

🔷 parentFrame : 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 parentFrame (Class) MainWindow

✏. Association (direction: Source -> Destination)

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

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.]

OPERATIONS

BannerPanelController (parentFrame : MainWindow , parentPanel : JPanel , ge : GameEngine) : 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.]

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

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.]

1.1.1.2.4   BoardCellController

Class in package 'controller'

<div align="right">

BoardCellController
Version 1.0 Phase 1.0 Proposed
pantz created on 3/5/2021. Last modified 3/5/2021
Extends AbstractController, ActionListener, MouseListener

</div>

OUTGOING STRUCTURAL RELATIONSHIPS

⬅ Generalization from BoardCellController to AbstractController

[Direction is 'Source -> Destination'.]

ATTRIBUTES

🔷 cell : BoardCell Private

[Is static False. Containment is Not Specified.]

ASSOCIATIONS

✏. Association (direction: Source -> Destination)

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

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.]

🔷 BoardCellController (cell : BoardCell , ge : GameEngine) : 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.]

OPERATIONS

◆ mouseReleased (e : MouseEvent) : void Public

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

1.1.1.2.5   GameBoardController

Class in package 'controller'

GameBoardController
Version 1.0 Phase 1.0 Proposed
pantz created on 3/5/2021. Last modified 3/5/2021
Extends AbstractController, ActionListener, MouseListener

OUTGOING STRUCTURAL RELATIONSHIPS

⬅ Generalization from GameBoardController to AbstractController

            [Direction is 'Source -> Destination'.]

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.]

◆ GameBoardController (ge : GameEngine) : 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.]

OPERATIONS

◆ mouseReleased (e : MouseEvent) : void Public

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

1.1.1.2.6   PlayerPanelController

Class in package 'controller'

PlayerPanelController
Version 1.0 Phase 1.0 Proposed
pantz created on 3/5/2021. Last modified 3/5/2021
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.]

OPERATIONS

◆ 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.]

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

1.1.1.3   io

Package in package 'tictactoe'

<div align="right">

io
Version 1.0 Phase 1.0 Proposed
pantz created on 3/5/2021. Last modified 3/5/2021

</div>

1.1.1.3.1   io diagram

Class diagram in package 'io'

<div align="right">

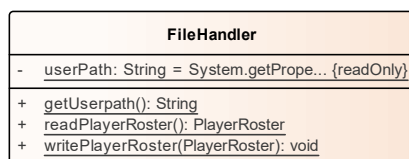io
Version 1.0
pantz created on 3/5/2021. Last modified 3/5/2021

</div>

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

Figure 3: io

1.1.1.3.2   FileHandler

Class in package 'io'

<div align="right">

FileHandler
Version 1.0 Phase 1.0 Proposed
pantz created on 3/5/2021. Last modified 3/5/2021

</div>

ATTRIBUTES

  ◆ userPath : String Private Const = System.getProperty("user.home") + File.separator + "tictactoe.ser"
                                          [Is static True. Containment is Not Specified.]

OPERATIONS

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

  ◆ readPlayerRoster () : PlayerRoster Public
                              [Is static True. Is abstract False. Is return array False. Is query False. Is synchronized False.]

  ◆ writePlayerRoster (playerRoster : PlayerRoster) : void Public
                              [Is static True. Is abstract False. Is return array False. Is query False. Is synchronized False.]

1.1.1.4   model

Package in package 'tictactoe'

<div align="right">

model
Version 1.0 Phase 1.0 Proposed
pantz created on 3/5/2021. Last modified 3/5/2021

</div>

1.1.1.4.1   model diagram

Class diagram in package 'model'

<div align="right">

model
Version 1.0
pantz created on 3/5/2021. Last modified 3/5/2021

</div>

Package in package 'tictactoe'

GameRecord

- players: Player ([])
- results: PlayerResult ([])
- scores: float ([])
- timeOfEnd: Instant
- timeOfStart: Instant

+ GameRecord()
+ getPlayers(): Player[]
+ getResults(): PlayerResult[]
+ getScores(): float[]
+ getTimeOfEnd(): Instant
+ getTimeOfStart(): Instant
+ setPlayers(Player[]): void
+ setResults(PlayerResult[]): void
+ setScores(float[]): void
+ setTimeOfEnd(Instant): void
+ setTimeOfStart(Instant): void

-gameRecord

-playersBestGames

-players

Serializable
Cloneable

Player

- draws: int
- losses: int
- name: String
- playersBestGames: GameRecord ([]) = new GameRecord[5]
- playersLastGames: GameRecord ([]) = new GameRecord[5]
- score: float
- totalGames: int
- wins: int

+ addDraw(): void
+ addLoss(): void
+ addWin(): void
- calculateScore(): void
+ clone(): Object
+ getDraws(): int
+ getLosses(): int
+ getName(): String
+ getPlayersBestGames(): GameRecord[]
+ getPlayersLastGames(): GameRecord[]
+ getScore(): float
+ getTotalGames(): int
+ getWins(): int
+ Player()
+ Player(String)
+ setDraws(int): void
+ setLosses(int): void
+ setName(String): void
+ setPlayersBestGames(GameRecord[]): void
+ setPlayersLastGames(GameRecord[]): void
+ setScore(float): void
+ setWins(int): void

-playerXrO

Board

{leaf}

- board: String ([][]) {readOnly}

+ Board()
+ Board(String[][])
+ getBoard(): String[]
+ getNumOfFullCells(): int
+ getNumOfNullCells(): int
+ getNumOfO(): int
+ getNumOfX(): int
+ isValid(): boolean
+ isWin(String): boolean
+ toString(): String

-board

Serializable

PlayerRoster

- players: HashMap<String,Player>

+ addPlayer(Player): void
+ findHallOfFame(int): Player[]
+ findPlayerByName(String): Player
+ findPlayerNames(): ArrayList<String>
+ PlayerRoster()

-playerRoster

GameEngine

- board: Board ([])
- gameActive: boolean = false
- gameRecord: GameRecord
- moves: int = 0
- playerO: Player
- playerRoster: PlayerRoster
- playerX: Player

- checkBoard(): void
+ checkGameStatus(): boolean
+ GameEngine()
+ getBoard(): Board[]
+ getGameRecord(): GameRecord
+ getMoves(): int
+ getPlayerO(): Player
+ getPlayerRoster(): PlayerRoster
+ getPlayerX(): Player
- getSign(): String
- initRoster(): void
- makeDummyPlayers(): void
+ makeMove(int, int): void
+ readyToPlay(): boolean
+ setBoard(Board[]): void
+ setGameRecord(GameRecord): void
+ setMoves(int): void
+ setPlayerO(Player): void
+ setPlayerRoster(PlayerRoster): void
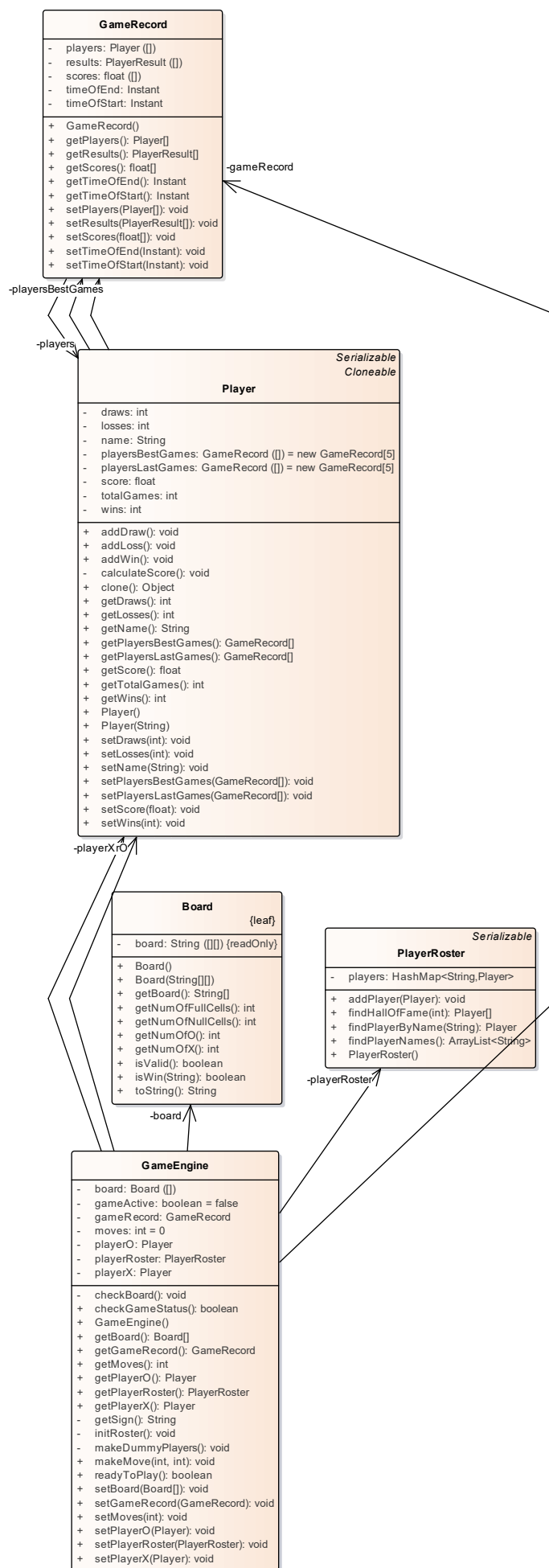+ setPlayerX(Player): void

Figure 4: model

1.1.1.4.2 enums

Package in package 'model'

<div align="right">

enums
Version 1.0 Phase 1.0 Proposed
pantz created on 3/5/2021. Last modified 3/5/2021

</div>

1.1.1.4.2.1 enums diagram

Class diagram in package 'enums'

<div align="right">

enums
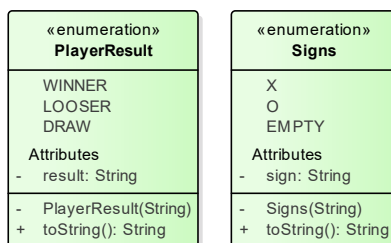Version 1.0
pantz created on 3/5/2021. Last modified 3/5/2021

</div>

```
«enumeration»          «enumeration»
  PlayerResult             Signs
─────────────────    ─────────────────
  WINNER                X
  LOOSER                O
  DRAW                  EMPTY
─────────────────    ─────────────────
  Attributes            Attributes
-   result: String    -   sign: String
─────────────────    ─────────────────
-   PlayerResult(String)  -   Signs(String)
+   toString(): String  +   toString(): String
```

Figure 5: enums

1.1.1.4.2.2 PlayerResult

Enumeration in package 'enums'

<div align="right">

PlayerResult
Version 1.0 Phase 1.0 Proposed
pantz created on 3/5/2021. Last modified 3/5/2021

</div>

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.]

1.1.1.4.2.3   Signs

Enumeration in package 'enums'

<div align="right">

Signs
Version 1.0 Phase 1.0 Proposed
pantz created on 3/5/2021. Last modified 3/5/2021

</div>

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.]

1.1.1.4.3   Board

Class in package 'model'

<div align="right">

Board
Version 1.0 Phase 1.0 Proposed
pantz created on 3/5/2021. Last modified 3/5/2021

</div>

ATTRIBUTES

🔷 board : String Private Const

              [Is static False. Containment is Not Specified.]

ASSOCIATIONS

🖊. 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

🖊. 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

OPERATIONS

🔶 Board () : Public

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

🔶 Board (b : String[][]) : Public

              [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.]

🔶 getNumOfFullCells () : int Public

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

🔶 getNumOfNullCells () : int Public

OPERATIONS
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
🔹 getNumOfO () : int Public
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
🔹 getNumOfX () : int Public
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
🔹 isValid () : boolean Public
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
🔹 isWin (s : String) : boolean 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.]

1.1.1.4.4   GameEngine

Class in package 'model'

GameEngine
Version 1.0 Phase 1.0 Proposed
pantz created on 3/5/2021. Last modified 3/5/2021

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.]

ASSOCIATIONS

✎. 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 board (Class) Board

✎. Association (direction: Source -> Destination)

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

✎. Association (direction: Source -> Destination)

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

✎. Association (direction: Source -> Destination)

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

✎. 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

OPERATIONS

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

🔷 checkGameStatus () : boolean Public
                    [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

🔷 GameEngine () : Public
                    [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

OPERATIONS

[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

Getters And Setters

[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.]

◆ getSign () : String Private

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

◆ initRoster () : void Private

[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.

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

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

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

◆ readyToPlay () : boolean Public

[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.]

◆ 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.]

1.1.1.4.5   GameRecord

Class in package 'model'

GameRecord
Version 1.0 Phase 1.0 Proposed
pantz created on 3/5/2021. Last modified 3/5/2021

ATTRIBUTES

◆ players : Player Private

[Is static False. Containment is Not Specified.]

◆ results : PlayerResult Private

[Is static False. Containment is Not Specified.]

◆ scores : float Private

[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 players (Class) Player

✎. Association (direction: Source -> Destination)

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

✎. 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

OPERATIONS

◆ GameRecord () : Public

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

◆ getPlayers () : Player Public

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

◆ getResults () : PlayerResult Public

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

OPERATIONS
🔷 getScores () : float Public
[Is static False. Is abstract False. Is return array True. 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.]
🔷 setPlayers (players : Player[]) : void Public
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
🔷 setResults (results : PlayerResult[]) : void Public
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
🔷 setScores (scores : 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.]

1.1.1.4.6   Player

Class in package 'model'

Player
Version 1.0 Phase 1.0 Proposed
pantz created on 3/5/2021. Last modified 3/5/2021
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.]

ATTRIBUTES

◆ 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 playersBestGames (Class) GameRecord

✎. Association (direction: Source -> Destination)

Source: Public (Class) Player

Target: Private playersLastGames (Class) GameRecord

✎. Association (direction: Source -> Destination)

Source: Public (Class) PlayerRosterTest

Target: Package p5 (Class) Player

✎. Association (direction: Source -> Destination)

Source: Public (Class) FileHandlerTest

Target: Package p5 (Class) Player

✎. Association (direction: Source -> Destination)

Source: Public (Class) GameRecord

Target: Private players (Class) Player

✎. 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) PlayerRosterTest

Target: Package p3 (Class) Player

✎. Association (direction: Source -> Destination)

Source: Public (Class) PlayerRosterTest

Target: Package p2 (Class) Player

✎. Association (direction: Source -> Destination)

Source: Public (Class) PlayerTest

Target: Package p3 (Class) Player

ASSOCIATIONS

✏. Association (direction: Source -> Destination)

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

✏. Association (direction: Source -> Destination)

Source: Public (Class) PlayerRosterTest   Target: Package p6 (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 p1 (Class) Player

✏. Association (direction: Source -> Destination)

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

✏. Association (direction: Source -> Destination)

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

✏. Association (direction: Source -> Destination)

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

OPERATIONS

◆ addDraw () : 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.]

OPERATIONS

🔶 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.]

🔶 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

OPERATIONS
[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.]

🔶 setWins (wins : int) : void Public
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

1.1.1.4.7   PlayerRoster

Class in package 'model'

PlayerRoster
Version 1.0 Phase 1.0 Proposed
pantz created on 3/5/2021. Last modified 3/5/2021
Extends Serializable

ATTRIBUTES

🔶 players : HashMap<String,Player> Private
[Is static False. Containment is Not Specified.]

ASSOCIATIONS

🖊. Association (direction: Source -> Destination)

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

🖊. Association (direction: Source -> Destination)

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

🖊. Association (direction: Source -> Destination)

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

OPERATIONS

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.
@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.]

1.1.1.5   utils

Package in package 'tictactoe'

<div align="right">

utils
Version 1.0 Phase 1.0 Proposed
pantz created on 3/5/2021. Last modified 3/5/2021

</div>

1.1.1.5.1   utils diagram

Class diagram in package 'utils'

<div align="right">

utils
Version 1.0
pantz created on 3/5/2021. Last modified 3/5/2021

</div>

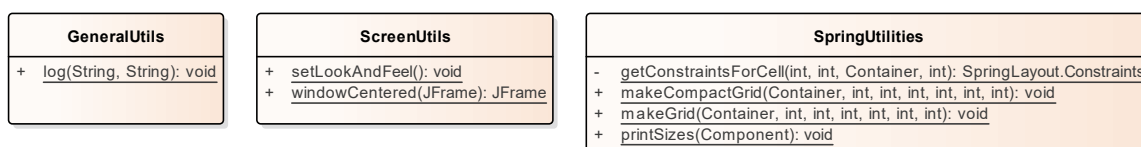| GeneralUtils	ScreenUtils	SpringUtilities
+ log(String, String): void	+ setLookAndFeel(): void + windowCentered(JFrame): JFrame	- getConstraintsForCell(int, int, Container, int): SpringLayout.Constraints + makeCompactGrid(Container, int, int, int, int, int, int): void + makeGrid(Container, int, int,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

Figure 6: utils

1.1.1.5.2   GeneralUtils

Class in package 'utils'

<div align="right">

GeneralUtils
Version 1.0 Phase 1.0 Proposed
pantz created on 3/5/2021. Last modified 3/5/2021

</div>

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

1.1.1.5.3   ScreenUtils

Class in package 'utils'

<div align="right">

ScreenUtils
Version 1.0 Phase 1.0 Proposed
pantz created on 3/5/2021. Last modified 3/5/2021

</div>

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.]

1.1.1.5.4   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.

<div align="right">

SpringUtilities

Version 1.0 Phase 1.0 Proposed

pantz created on 3/5/2021. Last modified 3/5/2021

</div>

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.]

1.1.1.6 view

Package in package 'tictactoe'

view
Version 1.0 Phase 1.0 Proposed
pantz created on 3/5/2021. Last modified 3/5/2021

1.1.1.6.1 view diagram

Class diagram in package 'view'

view
Version 1.0
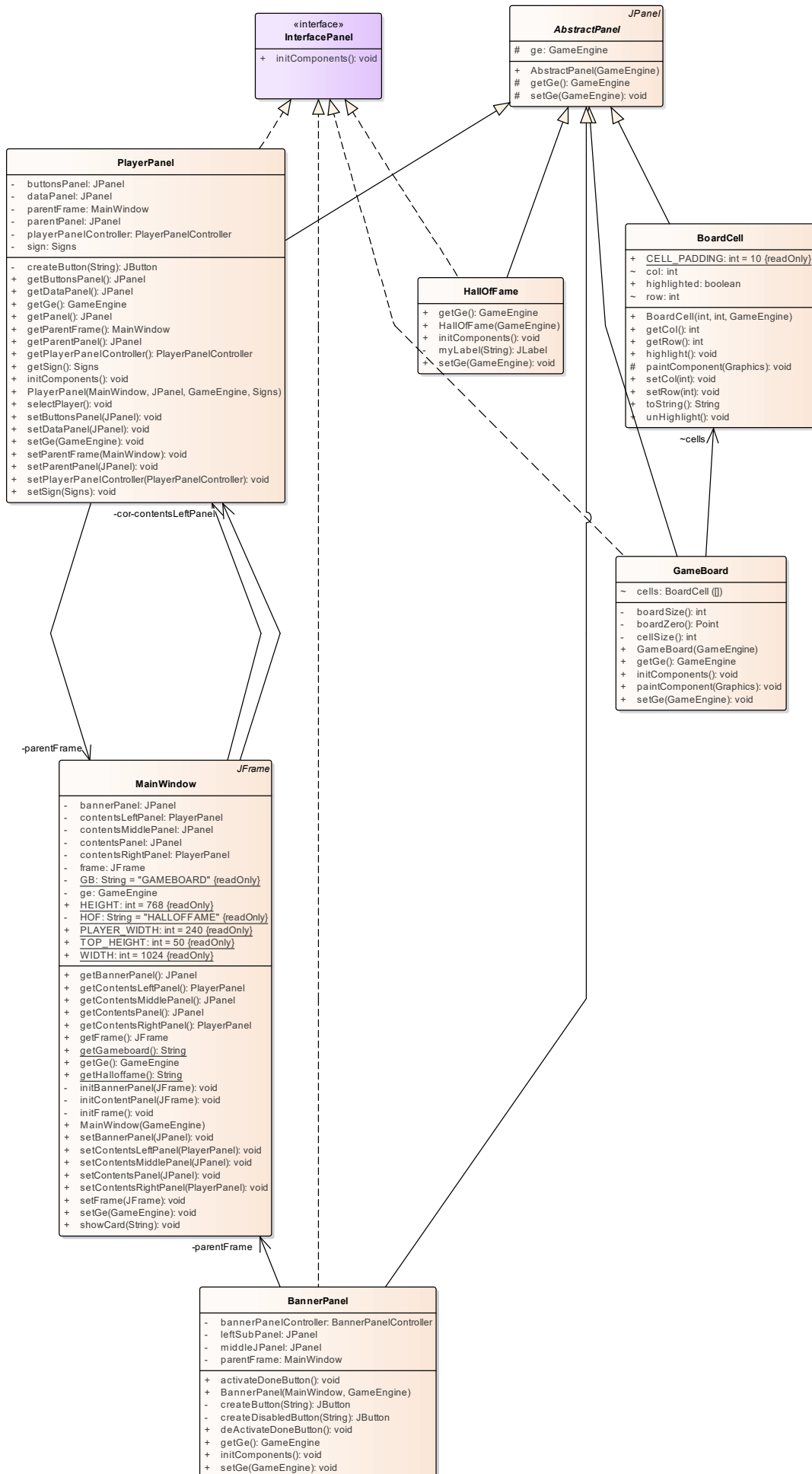pantz created on 3/5/2021. Last modified 3/5/2021

1.1.1.6 view

Package in package 'tictactoe'

«interface»
InterfacePanel

+ initComponents(): void

JPanel
AbstractPanel

\# ge: GameEngine

+ AbstractPanel(GameEngine)
\# getGe(): GameEngine
\# setGe(GameEngine): void

PlayerPanel

- buttonsPanel: JPanel
- dataPanel: JPanel
- parentFrame: MainWindow
- parentPanel: JPanel
- playerPanelController: PlayerPanelController
- sign: Signs

- createButton(String): JButton
+ getButtonsPanel(): JPanel
+ getDataPanel(): JPanel
+ getGe(): GameEngine
+ getPanel(): JPanel
+ getParentFrame(): MainWindow
+ getParentPanel(): JPanel
+ getPlayerPanelController(): PlayerPanelController
+ getSign(): Signs
+ initComponents(): void
+ PlayerPanel(MainWindow, JPanel, GameEngine, Signs)
+ selectPlayer(): void
+ setButtonsPanel(JPanel): void
+ setDataPanel(JPanel): void
+ setGe(GameEngine): void
+ setParentFrame(MainWindow): void
+ setParentPanel(JPanel): void
+ setPlayerPanelController(PlayerPanelController): void
+ setSign(Signs): void

HallOfFame

+ getGe(): GameEngine
+ HallOfFame(GameEngine)
+ initComponents(): void
- myLabel(String): JLabel
+ setGe(GameEngine): void

BoardCell

+ CELL_PADDING: int = 10 {readOnly}
~ col: int
+ highlighted: boolean
~ row: int

+ BoardCell(int, int, GameEngine)
+ getCol(): int
+ getRow(): int
+ highlight(): void
\# paintComponent(Graphics): void
+ setCol(int): void
+ setRow(int): void
+ toString(): String
+ unHighlight(): void

~cells

GameBoard

~ cells: BoardCell ([])

- boardSize(): int
- boardZero(): Point
- cellSize(): int
+ GameBoard(GameEngine)
+ getGe(): GameEngine
+ initComponents(): void
+ paintComponent(Graphics): void
+ setGe(GameEngine): void

-cor-contentsLeftPanel

-parentFrame

JFrame
MainWindow

- bannerPanel: JPanel
- 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(): JPanel
+ 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(JPanel): void
+ setContentsLeftPanel(PlayerPanel): void
+ setContentsMiddlePanel(JPanel): void
+ setContentsPanel(JPanel): void
+ setContentsRightPanel(PlayerPanel): void
+ setFrame(JFrame): void
+ setGe(GameEngine): void
+ showCard(String): void

-parentFrame

BannerPanel

- bannerPanelController: BannerPanelController
- leftSubPanel: JPanel
- middleJPanel: JPanel
- parentFrame: MainWindow

+ activateDoneButton(): void
+ BannerPanel(MainWindow, GameEngine)
- createButton(String): JButton
- createDisabledButton(String): JButton
+ deActivateDoneButton(): void
+ getGe(): GameEngine
+ initComponents(): void
+ setGe(GameEngine): void

Figure 7: view

1.1.1.6.2   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 3/5/2021. Last modified 3/5/2021
Extends JPanel

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

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

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

OPERATIONS
◆ AbstractPanel (ge : GameEngine) : 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.

OPERATIONS

	[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.]

1.1.1.6.3   BannerPanel

Class in package 'view'

BannerPanel
Version 1.0 Phase 1.0 Proposed
pantz created on 3/5/2021. Last modified 3/5/2021
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.]

◆ parentFrame : MainWindow Private
[Is static False. Containment is Not Specified.]

ASSOCIATIONS

✏ Association (direction: Source -> Destination)

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

✏ Association (direction: Source -> Destination)

Source: Public (Class) BannerPanel   Target: Private parentFrame (Class) MainWindow

OPERATIONS

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

OPERATIONS

◆ 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.]

1.1.1.6.4   BoardCell

Class in package 'view'

BoardCell
Version 1.0 Phase 1.0 Proposed
pantz created on 3/5/2021. Last modified 3/5/2021
Extends AbstractPanel

OUTGOING STRUCTURAL RELATIONSHIPS

⬅ Generalization from BoardCell to AbstractPanel

[Direction is 'Source -> Destination'.]

ATTRIBUTES

◆ CELL_PADDING : int Public Const   = 10

[Is static True. Containment is Not Specified.]

◆ col : int Package

[Is static False. Containment is Not Specified.]

◆ highlighted : boolean Public

[Is static False. Containment is Not Specified.]

ATTRIBUTES

🔹 row : int Package

[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)

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

OPERATIONS

🔹 BoardCell (row : int , col : int , ge : GameEngine) : 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.]

🔹 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.]

1.1.1.6.5   GameBoard

Class in package 'view'

GameBoard
Version 1.0 Phase 1.0 Proposed
pantz created on 3/5/2021. Last modified 3/5/2021
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.]

ASSOCIATIONS

✎ . Association (direction: Source -> Destination)

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

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) : 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

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.]

1.1.1.6.6   HallOfFame

Class in package 'view'

> HallOfFame
> Version 1.0 Phase 1.0 Proposed
> pantz created on 3/5/2021. Last modified 3/5/2021
> Extends AbstractPanel

OUTGOING STRUCTURAL RELATIONSHIPS

⬅ Realization from HallOfFame to InterfacePanel

> > [Direction is 'Source -> Destination'.]

⬅ Generalization from HallOfFame to AbstractPanel

> > [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) : 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.]

1.1.1.6.7   MainWindow

Class in package 'view'

> MainWindow
> Version 1.0 Phase 1.0 Proposed

pantz created on 3/5/2021. Last modified 3/5/2021
Extends JFrame

ATTRIBUTES

◆ bannerPanel : JPanel 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

[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 ge (Class) GameEngine

✎. Association (direction: Source -> Destination)

ASSOCIATIONS

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) BannerPanelController	Target: Private parentFrame (Class) MainWindow

✏. Association (direction: Source -> Destination)

Source: Public (Class) PlayerPanelController	Target: Private parentFrame (Class) MainWindow

✏. Association (direction: Source -> Destination)

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

✏. Association (direction: Source -> Destination)

Source: Public (Class) BannerPanel	Target: Private parentFrame (Class) MainWindow

OPERATIONS

🔶 getBannerPanel () : JPanel Public
              [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.]

OPERATIONS

💠 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

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

💠 setBannerPanel (bannerPanel : JPanel) : void Public
[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.]

1.1.1.6.8   PlayerPanel

Class in package 'view'

PlayerPanel
Version 1.0 Phase 1.0 Proposed
pantz created on 3/5/2021. Last modified 3/5/2021
Extends AbstractPanel

OUTGOING STRUCTURAL RELATIONSHIPS

⬅ Generalization from PlayerPanel to AbstractPanel

[Direction is 'Source -> Destination'.]

⬅ Realization from PlayerPanel to InterfacePanel

[Direction is 'Source -> Destination'.]

ATTRIBUTES

🔹 buttonsPanel : JPanel Private

[Is static False. Containment is Not Specified.]

🔹 dataPanel : JPanel Private

[Is static False. Containment is Not Specified.]

🔹 parentFrame : MainWindow 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) PlayerPanel   Target: Private parentFrame (Class) MainWindow

✏. 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 parentPanel (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.]

◆ 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.]

◆ 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 (parentFrame : 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.]

OPERATIONS

◆ 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.]

◆ 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.]

1.1.1.6.9   InterfacePanel

Interface in package 'view'

InterfacePanel
Version 1.0 Phase 1.0 Proposed
pantz created on 3/5/2021. Last modified 3/5/2021

INCOMING STRUCTURAL RELATIONSHIPS

➡ Realization from HallOfFame to InterfacePanel

[Direction is 'Source -> Destination'.]

➡ Realization from PlayerPanel to InterfacePanel

[Direction is 'Source -> Destination'.]

➡ Realization from GameBoard to InterfacePanel

[Direction is 'Source -> Destination'.]

➡ Realization from BannerPanel 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.]

1.1.1.7   Main

Class in package 'tictactoe'

This application is aJava 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 3/5/2021. Last modified 3/5/2021

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'

<div align="right">

test
Version 1.0 Phase 1.0 Proposed
pantz created on 3/5/2021. Last modified 3/5/2021

</div>

1.2.1 test diagram

Class diagram in package 'test'

<div align="right">

test
Version 1.0
pantz created on 3/5/2021. Last modified 3/5/2021

</div>

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()
- + setUp(): void
- + testReadPlayerRoster(): 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
- + 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
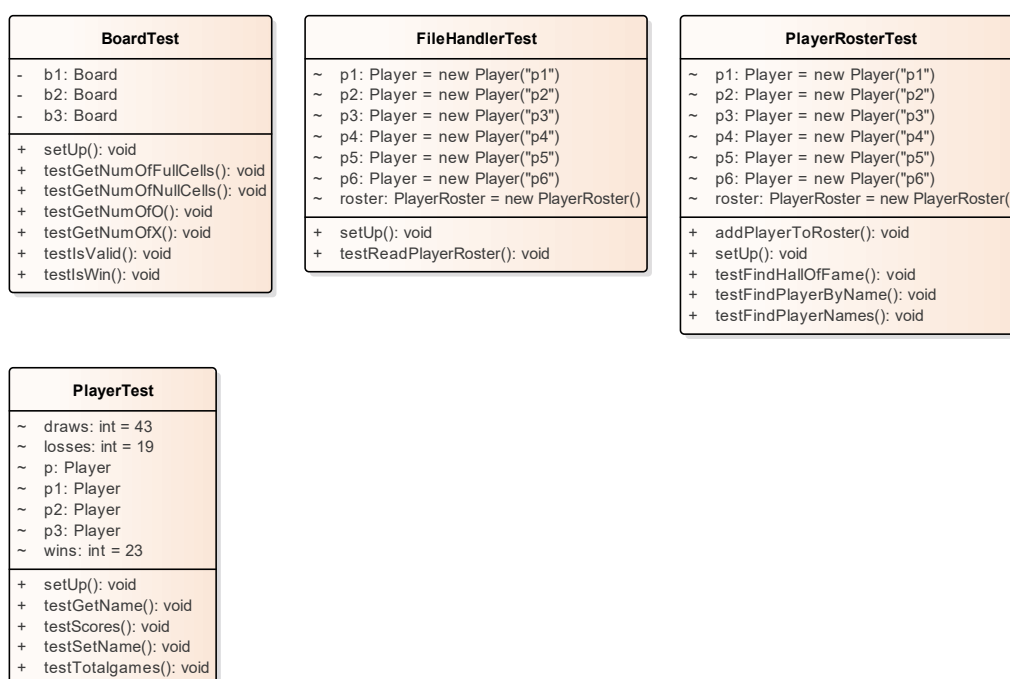
Figure 8: test

1.2.2 BoardTest

Class in package 'test'

<div align="right">

BoardTest
Version 1.0 Phase 1.0 Proposed
pantz created on 3/5/2021. Last modified 3/5/2021

</div>

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

ATTRIBUTES

🔷 b3 : Board Private

[Is static False. Containment is Not Specified.]

ASSOCIATIONS

✏️ 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

✏️ Association (direction: Source -> Destination)

Source: Public (Class) BoardTest   Target: Private b3 (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.]

🔷 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.]

OPERATIONS

◆ testIsWin () : void Public

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

1.2.3 FileHandlerTest

Class in package 'test'

FileHandlerTest
Version 1.0 Phase 1.0 Proposed
pantz created on 3/5/2021. Last modified 3/5/2021

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.]

◆ 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 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 p2 (Class) Player

ASSOCIATIONS

✏. Association (direction: Source -> Destination)

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

✏. Association (direction: Source -> Destination)

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

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.]

🔷 testReadPlayerRoster () : void Public

Properties:
    annotations = @Test

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

1.2.4 PlayerRosterTest

Class in package 'test'

PlayerRosterTest
Version 1.0 Phase 1.0 Proposed
pantz created on 3/5/2021. Last modified 3/5/2021

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.]

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

ATTRIBUTES

[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 p5 (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 p2 (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 p1 (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 roster (Class) PlayerRoster

OPERATIONS

🔷 addPlayerToRoster () : void Public

Properties:
   annotations = @Test

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

🔷 setUp () : void Public

Properties:
   annotations = @Before

OPERATIONS
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.]

1.2.5 PlayerTest

Class in package 'test'

PlayerTest
Version 1.0 Phase 1.0 Proposed
pantz created on 3/5/2021. Last modified 3/5/2021

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.]

🔷 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 p3 (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 p2 (Class) Player

✏. Association (direction: Source -> Destination)

Source: Public (Class) PlayerTest   Target: Package p1 (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
                        [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]