Yahtzee

+ image: Image
+ gameWindow: JFrame
+ introScreen: IntroScreen
+ settingsScreen: SettingsScreen
+ gameScreen: GameScreen

+ main

---

1   Has a

1   Has a

1 Has a

GameScreen

- backgroundColor: Color
- lightRed: Color
- darkGreen: Color
- darkBlue: Color
- turnLabel: GameLabel
- rollButton: JButton
- hand: Hand
- handPanel: JPanel
- scoreCard: ScoreCard
- scoreCardPanel: JPanel
- goToHomeScreen: JButton
- turnCount: int
- rollsPerTurn: int
- rollCount: int
- scoringMode: boolean
- gameInfo: FileCabinet
- finishGame: GameButton
- explanationPossibleScores: GameLabel
- PossibleScoreColor: GameLabel

+ setSwitchButtons
+ setBeginningGameStats
+ initializeComponents
+ setComponentLocations
+ setColorSettings
+ configureView
+ setMechanics
+ setScoringMode
+ scoreTurn
+ updateLabels

IntroScreen

- introScreenContent: JPanel
- yahtzeeTitle: JLabel
- footer: JPanel
- goToSettings: GameButton
- goToGame: GameButton

+ paintComponent
+ setSwitchButtons

SettingsScreen

- backgroundColor: Color
- lightBackgroundColor: Color
- darkBrown: Color
- lightRed: Color
- gameData: FileCabinet
- titleStrip: GameLabel
- title: GameLabel
- numDiceLabel: GameLabel
- numDiceOptions: Integer[]
- numDice: int
- numDiceBox: JComboBox<Integer>
- numSidesLabel: GameLabel
- numSidesOptions: Integer[]
- numSides: int
- numSidesBox: JComboBox<Integer>
- numRollsLabel: GameLabel
- numRollsOptions: Integer[]
- numRolls: int
- numRollsBox: JComboBox<Integer>
- goToIntroScreen: GameButton

+ configureComponentViewSettings
+ setComponentBounds
+ addComponents
+ setSwitchButtons
+ getGameData
+ updateValues

FileCabinet

- numSides: int
- numDice: int
- numRolls: int
- file: File
- fileOpen: boolean

+ checkIfFileOpened
+ updateValues
+ internalUpdateValues
+ getNumSides
+ getNumDice
+ getNumRolls

---

1   Has a   **1**

1 Has a **1**

1   Has a

1 Has a

ScoreCard

1   Has a   **1**

ScoreTable

Hand

1   Has a   ★

Die

JButton

Is a

GameButton
- darkBrown: Color
- lightRed: Color
- darkRed: Color
- lightGreen: Color
- darkGreen: Color
- lightBlue: Color
- darkBlue: Color
- mainColor: Color
- clickedColor: Color
+ setGameType
+ getClickedColor

JLabel

Is a

GameLabel
- darkBrown: Color
+ setTextSize

JRadioButton

Is a

GameRadioButton
- defaultIcon: Icon
- rolloverIcon: Icon
- pressedIcon: Icon
- selectedIcon: Icon
+ GameRadioButton

ScoreCard

- backgroundColor: Color
- darkBrown: Color
- lightRed: Color
- darkBlue: Color
- darkGreen: Color
- possibleScoreColor: Color
- toPresent: JPanel
- scoreTable: ScoreTable
- scoreTablePanel: JPanel
- scoringMenuButtons: JRadioButton[]
- scoringMenuButtonsGroup: ButtonGroup
- scores: Integer[]
- explanationScoringMenu1: GameLabel
- explanationScoringMenu2: GameLabel
- explanationScoringMenuWidth: int

+ setColorSettings
+ configureView
+ configureScoringMenuView
+ setScoringMenuButtonLocations
+ configurePanel
+ getTotalLength
+ getTotalWidth
+ clearScores
+ setScoreMenuMechanics
+ setScoreMenuVisibility
+ setScore
+ resetScore
+ showScoringOptions
+ calcFinalPoints
+ calcUpperScoreLinePoints
+ calcOfAKindPoints
+ calcFullHousePoints
+ calcStraightPoints
+ getTotalNumScoringRows
+ getScoringMenu
+ checkAvailableSpaces
+ getAppearance
+ getPossibleScoreColor

ScoreTable

- backgroundColor: Color
- darkBrown: Color
- lightRed: Color
- darkRed: Color
- lightGreen: Color
- darkGreen: Color
- lightBlue: Color
- darkBlue: Color
- toPresent: JPanel
- upperScoreCardContents: GameLabel[][]
- lowerScoreCardContents: GameLabel[][]
- rowWidth: int
- totalLength: int
- totalWidth: int

+ setViewSettings
+ setUpperScoreCardRows
+ configureView
+ configureUpperScoreCardViews
+ configureLowerScoreCardViews
+ setColumnCenterAligned
+ setCellBorders
+ configurePanel
+ setUpperCellValue
+ setLowerCellValue
+ getUpperScoreValue
+ getLowerScoreValue
+ getNumUpperScoringRows
+ getNumLowerScoringRows
+ getTotalNumScoringRows
+ getTotalLength
+ getTotalWidth
+ getRowHeight
+ getAppearance

1   Has a   1

Hand

- backgroundColor: Color
- toPresent: JPanel
- numRollsLabel: GameLabel
- rollingExplanationLabel: GameLabel
- dice: Die[]
- dicePanels: JPanel[]
- rollButton: GameButton
- numDice: int
- totalLength: int

+ configureView
+ setComponentLocations
+ setDiceLocations
+ configurePanel
+ setRollLabel
+ setExplanationLabelScoringMode
+ resetExplanationLabelText
+ hideDiceButtons
+ revealDiceButtons
+ hideRollingExplanation
+ revealRollingExplanation
+ calcHandSum
+ getRollLabel
+ getRollButton
+ getDice
+ getDie
+ getSize
+ getAppearance

Die

- toPresent: JPanel
- dieImage: JLabel
- dieButton: GameRadioButton
- sideUp: Integer
- numSides: Integer
- sixSidedDiceSides: ImageIcon[]
- eightSidedDiceSides: ImageIcon[]
- twelveSidedDiceSides: ImageIcon[]

+ configureView
+ roll
+ updateDieAppearance
+ getDieButton
+ getNumSides
+ getSideUp
+ getAppearance
+ compareTo

1   Has a   ★