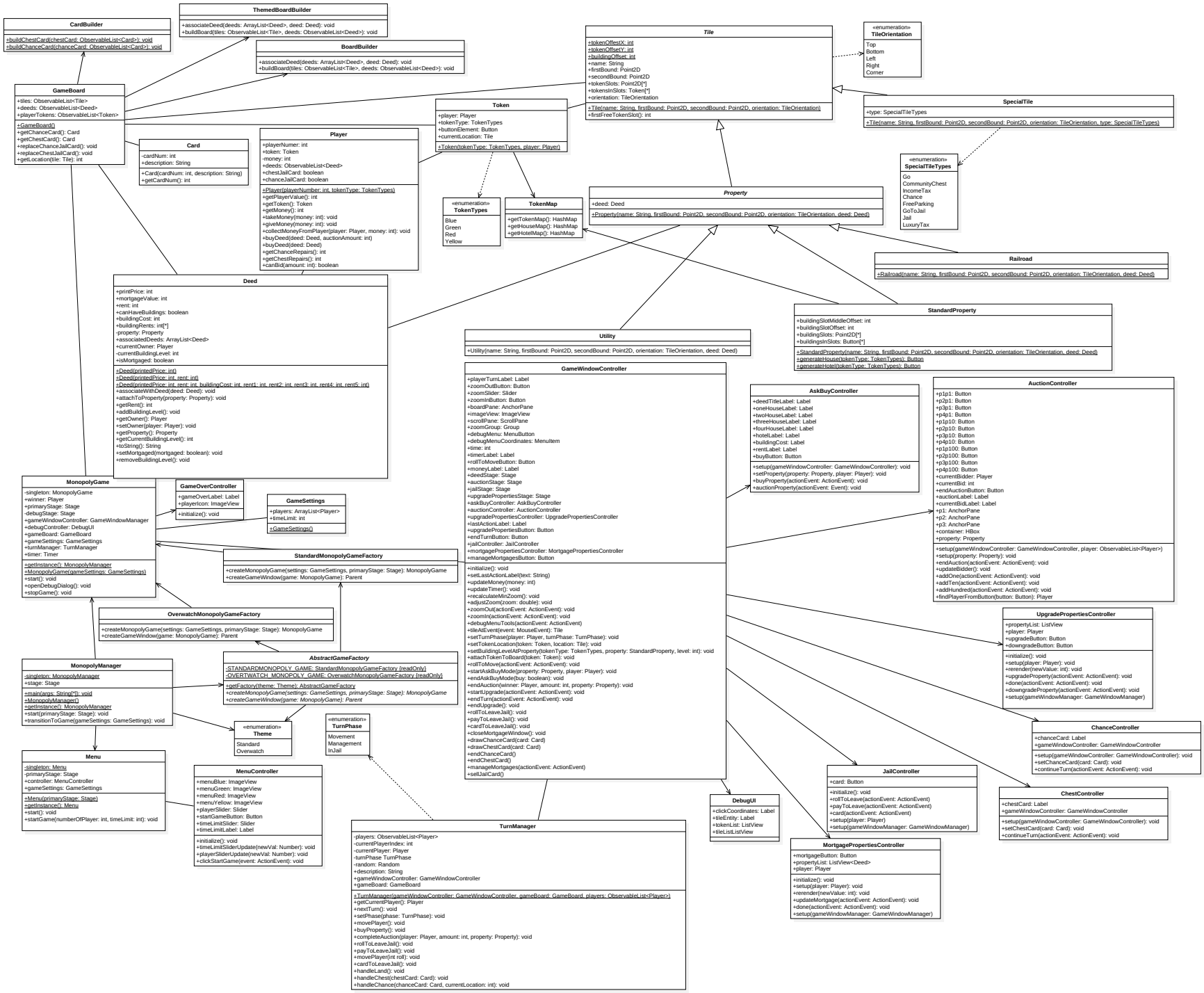
Model::Main


CardBuilder
+buildChestCard(chestCard: ObservableList<Card>): void
+buildChanceCard(chanceCard: ObservableList<Card>): void
ThemedBoardBuilder
+associateDeed(deeds: ArrayList<Deed>, deed: Deed): void
+buildBoard(tiles: ObservableList<Tile>, deeds: ObservableList<Deed>): void
BoardBuilder
+associateDeed(deeds: ArrayList<Deed>, deed: Deed): void
+buildBoard(tiles: ObservableList<Tile>, deeds: ObservableList<Deed>): void
GameBoard
+tiles: ObservableList<Tile>
+deeds: ObservableList<Deed>
+playerTokens: ObservableList<Token>
+GameBoard()
+getChanceCard(): Card
+getChestCard(): Card
+replaceChanceJailCard(): void
+replaceChestJailCard(): void
+getLocation(tile: Tile): int
Tile
+tokenOffestX: int
+tokenOffsetY: int
+buildingOffset: int
+name: String
+firstBound: Point2D
+secondBound: Point2D
+tokenSlots: Point2D[*]
+tokensInSlots: Token[*]
+orientation: TileOrientation
+Tile(name: String, firstBound: Point2D, secondBound: Point2D, orientation: TileOrientation)
+firstFreeTokenSlot(): int
«enumeration»
TileOrientation
Top
Bottom
Left
Right
Corner
SpecialTile
+type: SpecialTileTypes
+Tile(name: String, firstBound: Point2D, secondBound: Point2D, orientation: TileOrientation, type: SpecialTileTypes)
Token
+player: Player
+tokenType: TokenTypes
+buttonElement: Button
+currentLocation: Tile
+Token(tokenType: TokenTypes, player: Player)
Card
-cardNum: int
+description: String
+Card(cardNum: int, description: String)
+getCardNum(): int
Player
+playerNumer: int
+token: Token
-money: int
+deeds: ObservableList<Deed>
+chestJailCard: boolean
+chanceJailCard: boolean
+Player(playerNumber: int, tokenType: TokenTypes)
+getPlayerValue(): int
+getToken(): Token
+getMoney(): int
+takeMoney(money: int): void
+giveMoney(money: int): void
+collectMoneyFromPlayer(player: Player, money: int): void
+buyDeed(deed: Deed, auctionAmount: int)
+buyDeed(deed: Deed)
+getChanceRepairs(): int
+getChestRepairs(): int
+canBid(amount: int): boolean
«enumeration»
SpecialTileTypes
Go
CommunityChest
IncomeTax
Chance
FreeParking
GoToJail
Jail
LuxuryTax
«enumeration»
TokenTypes
Blue
Green
Red
Yellow
TokenMap
+getTokenMap(): HashMap
+getHouseMap(): HashMap
+getHotelMap(): HashMap
Property
+deed: Deed
+Property(name: String, firstBound: Point2D, secondBound: Point2D, orientation: TileOrientation, deed: Deed)
Railroad
+Railroad(name: String, firstBound: Point2D, secondBound: Point2D, orientation: TileOrientation, deed: Deed)
Deed
+printPrice: int
+mortgageValue: int
+rent: int
+canHaveBuildings: boolean
+buildingCost: int
+buildingRents: int[*]
-property: Property
+associatedDeeds: ArrayList<Deed>
+currentOwner: Player
-currentBuildingLevel: int
+isMortgaged: boolean
+Deed(printedPrice: int)
+Deed(printedPrice: int, rent: int)
+Deed(printedPrice: int, rent: int, buildingCost: int, rent1: int, rent2: int, rent3: int, rent4: int, rent5: int)
+associateWithDeed(deed: Deed): void
+attachToProperty(property: Property): void
+getRent(): int
+addBuildingLevel(): void
+getOwner(): Player
+setOwner(player: Player): void
+getProperty(): Property
+getCurrentBuildingLevel(): int
+toString(): String
+setMortgaged(mortgaged: boolean): void
+removeBuildingLevel(): void
StandardProperty
+buildingSlotMiddleOffset: int
+buildingSlotOffset: int
+buildingSlots: Point2D[*]
+buildingsInSlots: Button[*]
+StandardProperty(name: String, firstBound: Point2D, secondBound: Point2D, orientation: TileOrientation, deed: Deed)
+generateHouse(tokenType: TokenTypes): Button
+generateHotel(tokenType: TokenTypes): Button
Utility
+Utility(name: String, firstBound: Point2D, secondBound: Point2D, orientation: TileOrientation, deed: Deed)
GameWindowController
+playerTurnLabel: Label
+zoomOutButton: Button
+zoomSlider: Slider
+zoomInButton: Button
+boardPane: AnchorPane
+imageView: ImageView
+scrollPane: ScrollPane
+zoomGroup: Group
+debugMenu: MenuButton
+debugMenuCoordinates: MenuItem
+time: int
+timerLabel: Label
+rollToMoveButton: Button
+moneyLabel: Label
+deedStage: Stage
+auctionStage: Stage
+jailStage: Stage
+upgradePropertiesStage: Stage
+askBuyController: AskBuyController
+auctionController: AuctionController
+upgradePropertiesController: UpgradePropertiesController
+lastActionLabel: Label
+upgradePropertiesButton: Button
+endTurnButton: Button
+jailController: JailController
+mortgagePropertiesController: MortgagePropertiesController
+manageMortgagesButton: Button
+initialize(): void
+setLastActionLabel(text: String)
+updateMoney(money: int)
+updateTimer(): void
+recalculateMinZoom(): void
+adjustZoom(zoom: double): void
+zoomOut(actionEvent: ActionEvent): void
+zoomIn(actionEvent: ActionEvent): void
+debugMenuTools(actionEvent: ActionEvent)
+tileAtEvent(event: MouseEvent): Tile
+setTurnPhase(player: Player, turnPhase: TurnPhase): void
+setTokenLocation(token: Token, location: Tile): void
+setBuildingLevelAtProperty(tokenType: TokenTypes, property: StandardProperty, level: int): void
+attachTokenToBoard(token: Token): void
+rollToMove(actionEvent: ActionEvent): void
+startAskBuyMode(property: Property, player: Player): void
+endAskBuyMode(buy: boolean): void
+endAuction(winner: Player, amount: int, property: Property): void
+startUpgrade(actionEvent: ActionEvent): void
+endTurn(actionEvent: ActionEvent): void
+endUpgrade(): void
+rollToLeaveJail(): void
+payToLeaveJail(): void
+cardToLeaveJail(): void
+closeMortgageWindow(): void
+drawChanceCard(card: Card)
+drawChestCard(card: Card)
+endChanceCard()
+endChestCard()
+manageMortgages(actionEvent: ActionEvent)
+sellJailCard()
AskBuyController
+deedTitleLabel: Label
+oneHouseLabel: Label
+twoHouseLabel: Label
+threeHouseLabel: Label
+fourHouseLabel: Label
+hotelLabel: Label
+buildingCost: Label
+rentLabel: Label
+buyButton: Button
+setup(gameWindowController: GameWindowController): void
+setProperty(property: Property, player: Player): void
+buyProperty(actionEvent: ActionEvent): void
+auctionProperty(actionEvent: Event): void
AuctionController
+p1p1: Button
+p2p1: Button
+p3p1: Button
+p4p1: Button
+p1p10: Button
+p2p10: Button
+p3p10: Button
+p4p10: Button
+p1p100: Button
+p2p100: Button
+p3p100: Button
+p4p100: Button
+currentBidder: Player
+currentBid: int
+endAuctionButton: Button
+auctionLabel: Label
+currentBidLabel: Label
+p1: AnchorPane
+p2: AnchorPane
+p3: AnchorPane
+container: HBox
+property: Property
+setup(gameWindowController: GameWindowController, player: ObservableList<Player>)
+setup(property: Property): void
+endAuction(actionEvent: ActionEvent): void
+updateBidder(): void
+addOne(actionEvent: ActionEvent): void
+addTen(actionEvent: ActionEvent): void
+addHundred(actionEvent: ActionEvent): void
+findPlayerFromButton(button: Button): Player
MonopolyGame
-singleton: MonopolyGame
+winner: Player
+primaryStage: Stage
-debugStage: Stage
+gameWindowController: GameWindowManager
+debugController: DebugUI
+gameBoard: GameBoard
+gameSettings: GameSettings
+turnManager: TurnManager
+timer: Timer
+getInstance(): MonopolyManager
+MonopolyGame(gameSettings: GameSettings)
+start(): void
+openDebugDialog(): void
+stopGame(): void
GameOverController
+gameOverLabel: Label
+playerIcon: ImageView
+initialize(): void
GameSettings
+players: ArrayList<Player>
+timeLimit: int
+GameSettings()
StandardMonopolyGameFactory
+createMonopolyGame(settings: GameSettings, primaryStage: Stage): MonopolyGame
+createGameWindow(game: MonopolyGame): Parent
OverwatchMonopolyGameFactory
+createMonopolyGame(settings: GameSettings, primaryStage: Stage): MonopolyGame
+createGameWindow(game: MonopolyGame): Parent
UpgradePropertiesController
+propertyList: ListView
+player: Player
+upgradeButton: Button
+downgradeButton: Button
+initialize(): void
+setup(player: Player): void
+rerender(newValue: int): void
+upgradeProperty(actionEvent: ActionEvent): void
+done(actionEvent: ActionEvent): void
+downgradeProperty(actionEvent: ActionEvent): void
+setup(gameWindowManager: GameWindowManager)
MonopolyManager
-singleton: MonopolyManager
+stage: Stage
+main(args: String[*]): void
+MonopolyManager()
+getInstance(): MonopolyManager
+start(primaryStage: Stage): void
+transitionToGame(gameSettings: GameSettings): void
AbstractGameFactory
-STANDARDMONOPOLY_GAME: StandardMonopolyGameFactory {readOnly}
-OVERTWATCH_MONOPOLY_GAME: OverwatchMonopolyGameFactory {readOnly}
+getFactory(theme: Theme): AbstractGameFactory
+createMonopolyGame(settings: GameSettings, primaryStage: Stage): MonopolyGame
+createGameWindow(game: MonopolyGame): Parent
«enumeration»
Theme
Standard
Overwatch
«enumeration»
TurnPhase
Movement
Management
InJail
ChanceController
+chanceCard: Label
+gameWindowController: GameWindowController
+setup(gameWindowController: GameWindowController): void
+setChanceCard(card: Card): void
+continueTurn(actionEvent: ActionEvent): void
Menu
-singleton: Menu
-primaryStage: Stage
+controller: MenuController
+gameSettings: GameSettings
+Menu(primaryStage: Stage)
+getInstance(): Menu
+start(): void
+startGame(numberOfPlayer: int, timeLimit: int): void
MenuController
+menuBlue: ImageView
+menuGreen: ImageView
+menuRed: ImageView
+menuYellow: ImageView
+playerSlider: Slider
+startGameButton: Button
+timeLimitSlider: Slider
+timeLimitLabel: Label
+initialize(): void
+timeLimitSliderUpdate(newVal: Number): void
+playerSliderUpdate(newVal: Number): void
+clickStartGame(event: ActionEvent): void
JailController
+card: Button
+initialize(): void
+rollToLeave(actionEvent: ActionEvent)
+payToLeave(actionEvent: ActionEvent)
+card(actionEvent: ActionEvent)
+setup(player: Player)
+setup(gameWindowManager: GameWindowManager)
ChestController
+chestCard: Label
+gameWindowController: GameWindowController
+setup(gameWindowController: GameWindowController): void
+setChestCard(card: Card): void
+continueTurn(actionEvent: ActionEvent): void
DebugUI
+clickCoordinates: Label
+tileEntity: Label
+tokenList: ListView
+tileListListView
MortgagePropertiesController
+mortgageButton: Button
+propertyList: ListView<Deed>
+player: Player
+initialize(): void
+setup(player: Player): void
+rerender(newValue: int): void
+updateMortgage(actionEvent: ActionEvent): void
+done(actionEvent: ActionEvent): void
+setup(gameWindowManager: GameWindowManager)
TurnManager
-players: ObservableList<Player>
-currentPlayerIndex: int
-currentPlayer: Player
-turnPhase TurnPhase
-random: Random
+description: String
+gameWindowController: GameWindowController
+gameBoard: GameBoard
+TurnManager(gameWindowController: GameWindowController, gameBoard: GameBoard, players: ObservableList<Player>)
+getCurrentPlayer(): Player
+nextTurn(): void
+setPhase(phase: TurnPhase): void
+movePlayer(): void
+buyProperty(): void
+completeAuction(player: Player, amount: int, property: Property): void
+rollToLeaveJail(): void
+payToLeaveJail(): void
+movePlayer(int roll): void
+cardToLeaveJail(): void
+handleLand(): void
+handleChest(chestCard: Card): void
+handleChance(chanceCard: Card, currentLocation: int): void