HouseAndHotelController
- model : MonopolyModel

MonopolyModel
- board : Board
- players : List<Player>

DiceController

- model : MonopolyModel

+ actionPerformed(ActionEvent e) : void

InfoController

- model : MonopolyModel

+ actionPerformed(ActionEvent e) : void

MonopolyBoardController

- model : MonopolyModel

+ actionPerformed(ActionEvent e) : void

+ actionPerformed(ActionEvent e) : void

- players : List<Player>
- die : Dice
- playerTurn: int
- view : MonopolyView

+ addMonopolyView(MonopolyView mv) : void
+ addPlayer(String name) : void
+ buyProperty(int selection) : boolean
+ checkProperty() : boolean
+ getBoard() : Board
+ getPlayers() : Player
+ getPlayer(): Player
+ getPropertyInfo(): String
+ getPropertyOwner(): Player
+ handleEmptyProperties() : boolean
+ payRent(): void
+ playTurn(int rollValue) : void
+ removePlayer(): void
+ roll(): int
+ addAI(String name) : void
+ AITurn() : void
+ goToJail() : void
+ inJail() : void
+ isDoubles() : boolean
+ setRailroadRent() : void
+ setUtilityRent(int roll) : void

Dice

- dice1: int
- dice2: int
- doubles : boolean

+ roll(): void
+ checkDoubles() : void
+ isDoubles() : boolean

Board

- properties : List<Property>

+ getProperty(int position): Property

Player

- name: String
- money : int
- position : int
- propertiesOwned: List<Property>
- positionTracker : int
- jailed : boolean
- BOARDSIZE : int;

+ getName(): String
+ getPosition(): int
+ getJailed() : boolean
+ setJailed(boolean jailed) : void
+ addPosition(int position): void
+ getMoney(): int
+ addMoney(int money): void
+ updatePositionTracker() : void
+ getPositionTracker() : int
+ buy(Property property): boolean
+ rent(Property property): void
+ removeProperties(): void
+ getProperties(): String
+ toString(): String
+ getPropertiesForController() : string
+ getAmountofRailroads() : int
+ getAmountOfUtilities() : int
+ getPropertiesOwned(): List<Property>
+ getPropertiesArray(): List<Property>
+ buyHouse(String propertyName) : void

Property

- name : String
- cost : int
- rent : int
- owner : Player
- hasHouse : boolean
- hasHotel : boolean

+ setOwner(Player player): void
+ getOwner(): Player
+ getCost(): int
+ updateRent(int rent) : void
+ getRent(): int
+ getName(): String
+ toString(): String
+ setHasHouse (boolean hasHouse) :
+ hasHouse() : boolean
+ setHasHotel (boolean hasHotel) : v
+ hasHotel() : boolean

MonopolyView

- model : MonopolyModel
- pane : Container
- turnLabel : JLabel
- propertyButtons : JButton
- BOARDLENGTH : int
- BOARDSIZE : int

+ AIBuy(String playerName, String propertyName
+ checkAvailability() : void
+ playerEliminated() : void
+ playerWin() : void
+ playerSetup() : void
+ updateStatus() : void
+ createBoard() : void
+ main(String[] args) : void

) : void

FreeParking

-name : String
- cost : int
- rent : int

Go

-name : String
- cost : int
- rent : int

+ toString() : String

+ buyHotel(String propertyName) : void

AI

-name : String

Railroad

-name : String
- cost : int
- rent : int

+ updateRent(int amount) : void
+ toString() : String

Utilities

-name : String
- cost : int
- rent : int

+ toString() : String

Jail

- jailedPlayers : List<Player>
- jailedTurns : int[]

+ inJail(Player player, boolean isDoubles)
+ addToJailList(Player player) : void

GoToJail

- jail : Jail
- GOTOJAILSPACES : int

+ goToJail(Player player) : void