Game

- board:Board
- players:Player[ ]
- numberOfPlayers:int
- roundCount: int
- turn:int
- dice:Die[2]

+ startGame( ):void
- makePlayCurrentPlayer():void
- nextPlayerTurn():void
- checkBankruptcy():void
- roundCheck(int):void

Board

- squares:Square[40]
- ccCards: CommunityChestCard[16]
- cCards: ChanceCard[16]
- ownersOfBrown: Player[2]
- ownersOfBlue: Player[3]
- ownersOfPink: Player[3]
- ownersOfOrange: Player[3]
- ownersOfRed: Player[3]
- ownersOfYellow: Player[3]
- ownersOfGreen: Player[3]
- ownersOfDarkblue: Player[2]
- ownersOfUtility: Player[2]
- ownersOfRailroad: Player[4]

+ setSquareWithInfoFromFile:void
+ addOwnerToColorGroups(LotSquare,Player):void
- findGroup(LotSquare):Player[ ]
+ isOwnerHasAllSquareInTheSameGroup(LotSquare,Player):boolean
+ isOwnerHasAllUtility(UtilitySquare):boolean
+ addOwnerToUtilityGroup(Player):void
+ howManyRailroad(RailroadSquare):int
+ addOwnerToRailroad(Railroad,Player):void
+ findNewLocation(Square,int,String):Square

Player

- name:String
- token: Token
- location:Square
- money:int
- sumOfDice:int
- inJail:boolean
- numberOfRoundsOnJail:int
- hasJailFreeCard:boolean
- ownedCCCard:CommunityChestCard
- ownedCCard:ChanceCard

+ rollDice(Die,Die):int
+ whereWillBeTokenLocated(Board,String):void
+ addMoney(int):void
+ subtractMoney(int):void
+ toString():void
- toStringGoSquare(Player):void

Die

- faceValue:int
- numberOfFaces:int

+ rollDie( ):int
+ toString():void

Token

- tokenId:int
- currentSquare:Square

+ toStringGoSquare():void

Square {Abstract}

- squareName:String
- squareNo:int
- totalTokenNumber:int
- tokens:Token[ ]

+ addToken(Token):void
+ removeToken(Token):void
+ *doOperation(Player,Board):void {Abstract}*

IncomeTaxSquare

- incomeTax:int

+ doOperation(Player,Board):void
- toStringPayIncomeTax(Player):void

LuxuryTaxSquare

- luxuryTax:int

+ doOperation(Player,Board):void
- toStringPayLuxuryTax(Player):void

RegularSquare

+ doOperation(Player,Board):void

CommunityChestSquare

- cards:CommunityChestCard[16]
- turnOfCommunityChestCards:Int

+ drawACard:CommunityChestCard
+ controlCard:int
+ doOperation(Player,Board):void

ChanceSquare

- cards:ChanceCard[16]
- turnOfChanceCards:Int

+ drawACard:ChanceCard
+ controlCard:int
+ doOperation(Player,Board):void

PropertySquare {Abstract}

- owner:Player
- priceOfProperty:int
- rent:int
- mortgageValue:int

+ buyProperty(Player):void
+ *setTitleDeed(Scanner):void {Abstract}*
+ doOperation(Player,Board):void
+ toStringBuyProperty():void
+ toStringPayRent(int,Player):void

GoSquare

+ doOperation(Player,Board):void
+ toStringGoSquare(Player):void

JailSquare

+ doOperation(Player,Board):void
+ toStringJailSquare(Player):void

LotSquare

- colorGroup:String
- rentWithOneHouse:int
- rentWithTwoHouse:int
- rentWithThreeHouse:int
- rentWithFourHouse:int
- rentWithHotel:int
- eachHouseCost:int
- eachHotelCost:int

+ setTitleDeed(Scanner):void
+ doOperation(Player,Board):void

RailroadSquare

- rentWithTwoRailroad:int
- rentWithThreeRailroad:int
- rentWithFourRailroad:int

+ setTitleDeed(Scanner):void
+ doOperation(Player,Board):void

UtilitySquare

- rentWithTwoUtility:int

+ setTitleDeed(file:Scanner):void
+ doOperation(Player,Board):void

**<< interface >>
CommunityChestCard**

+ getCardNo():int
+ cardOperations(Player,Board):void
+ getText():String
+ isCardBlocked():boolean
+ setCardBlocked(boolean):void

ChanceCard {Abstract}

+ *getCardNo():int {Abstract}*
+ *cardOperations(Player,Board):void {Abstract}*
+ *getText():String {Abstract}*
+ *setText(String):void {Abstract}*
+ *isCardBlocked():boolean {Abstract}*
+ *setCardBlocked(boolean):void {Abstract}*

LoseMoneyCard

- cardNo:int
- amount:int
+ isCardBlocked: boolean
- text:String

+ getCardNo():int
+ getAmount():int
+ setAmount(int):void
+ getText():String
+ setText(String):void
+ isCardBlocked():boolean
+ setCardBlocked(boolean):void
+ cardOperations(Player,Board):void

GainMoneyCard

- cardNo:int
- amount:int
+ isCardBlocked: boolean
- text:String

+ getCardNo():int
+ getAmount():int
+ setAmount(int):void
+ getText():String
+ setText(String):void
+ isCardBlocked():boolean
+ setCardBlocked(boolean):void
+ cardOperations(Player,Board):void

MoveForwardCard

- cardNo:int
- squareToGo:int
- isCardBlocked: boolean
- text:String

+ getCardNo():int
+ getSquareToGo():int
+ setSquareToGo(int)
+ getText():String
+ setText(String):void
+ isCardBlocked():boolean
+ setCardBlocked(boolean):void
+ cardOperations(Player,Board):void

MoveBackwardCard

- cardNo:int
- squareToGo:int
- isCardBlocked: boolean
- text:String

+ getCardNo():int
+ getSquareToGo():int
+ setSquareToGo(int):void
+ getText():String
+ setText(String):void
+ isCardBlocked():boolean
+ setCardBlocked(boolean):void
+ cardOperations(Player,Board):void

JailFreeCard

- cardNo:int
- isCardBlocked: boolean
- text:String

+ getCardNo():int
+ getText():String
+ setText(String):void
+ isCardBlocked():boolean
+ setCardBlocked(boolean):void
+ cardOperations(Player,Board):void