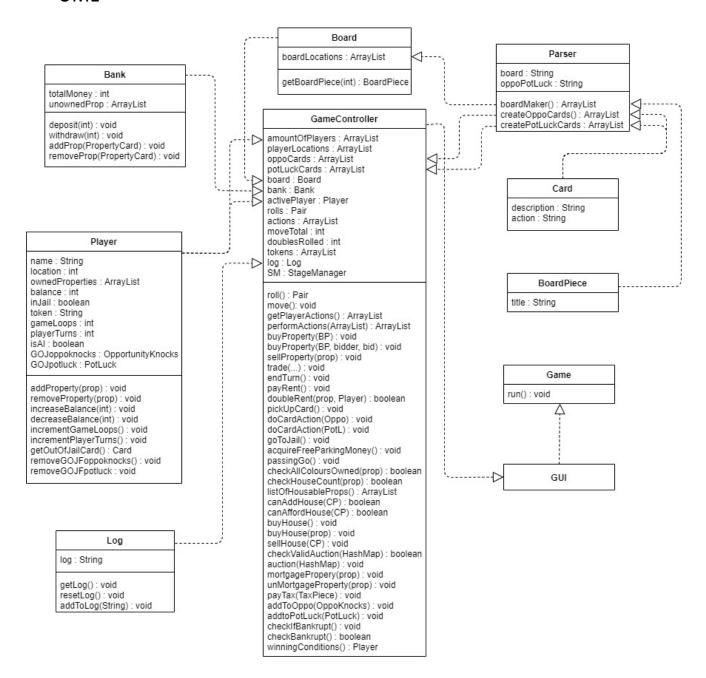
Design Document

UML



GameController class structure

GameController

amountOfPlayers : ArrayList playerLocations : ArrayList oppoCards : ArrayList potLuckCards : ArrayList

board : Board bank : Bank activePlayer : Player

rolls : Pair actions : ArrayList moveTotal : int doublesRolled : int tokens : ArrayList

log: Log

SM: StageManager

roll(): Pair move(): void

getPlayerActions(): ArrayList performActions(ArrayList): ArrayList

buyProperty(BP): void

buyProperty(BP, bidder, bid): void sellProperty(prop): void

trade(...): void endTurn(): void payRent(): void

doubleRent(prop, Player) : boolean

pickUpCard(): void doCardAction(Oppo): void doCardAction(PotL): void goToJail(): void

acquireFreeParkingMoney(): void

passingGo(): void

checkAllColoursOwned(prop): boolean checkHouseCount(prop): boolean listOfHousableProps(): ArrayList canAddHouse(CP): boolean canAffordHouse(CP): boolean

buyHouse() : void buyHouse(prop) : void sellHouse(CP) : void

checkValidAuction(HashMap): boolean

auction(HashMap) : void mortgagePropery(prop) : void unMortgageProperty(prop) : void payTax(TaxPiece) : void addToOppo(OppoKnocks) : void addtoPotLuck(PotLuck) : void

checklfBankrupt(): void checkBankrupt(): boolean winningConditions(): Player

Timed

amountOfPlayers: ArrayList playerLocations: ArrayList oppoCards: ArrayList potLuckCards: ArrayList

board : Board bank : Bank activePlayer : Player rolls : Pair

actions : ArrayList moveTotal : int doublesRolled : int tokens : ArrayList log : Log

SM : StageManager

+t : Timer +task : TimerTask

roll(): Pair move(): void

getPlayerActions(): ArrayList performActions(ArrayList): ArrayList

buyProperty(BP): void

buyProperty(BP, bidder, bid) : void

sellProperty(prop) : void trade(...) : void

endTurn(): void payRent(): void

doubleRent(prop, Player) : boolean

pickUpCard(): void doCardAction(Oppo): void doCardAction(PotL): void qoToJail(): void

acquireFreeParkingMoney(): void

passingGo(): void

checkAllColoursOwned(prop): boolean checkHouseCount(prop): boolean listOfHousableProps(): ArrayList canAddHouse(CP): boolean canAffordHouse(CP): boolean

buyHouse(): void buyHouse(prop): void sellHouse(CP): void

checkValidAuction(HashMap): boolean

auction(HashMap): void mortgagePropery(prop): void unMortgageProperty(prop): void payTax(TaxPiece): void addToOppo(OppoKnocks): void addtoPotLuck(PotLuck): void

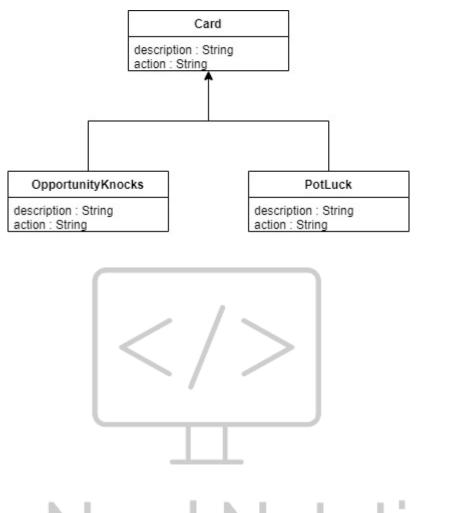
checkIfBankrupt(): void checkBankrupt(): boolean +fullTurn(): boolean

+winningConditions(): Player

Player class structure

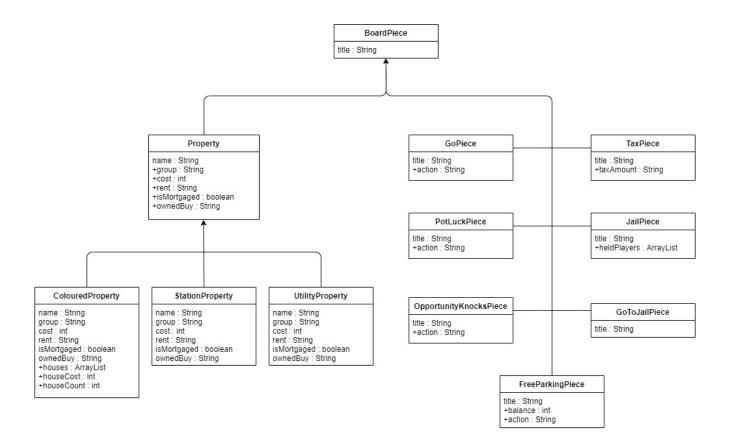
Player name: String location : int ownedProperties : ArrayList balance: int inJail: boolean token : String gameLoops: int playerTurns: int isAl: boolean addProperty(prop): void removeProperty(prop): void increaseBalance(int): void decreaseBalance(int): void incrementGameLoops(): void incrementPlayerTurns(): void AiPlayer name: String location : int ownedProperties : ArrayList balance : int inJail: boolean token: String otation gameLoops: int playerTurns: int isAl : boolean addProperty(prop): void removeProperty(prop): void increaseBalance(int) : void decreaseBalance(int): void incrementGameLoops(): void incrementPlayerTurns(): void +DoesAiBuy(): boolean +AiAuctionValue(prop): int +tryBuyHouse(): boolean +trySellProperty(): boolean

Card class structure



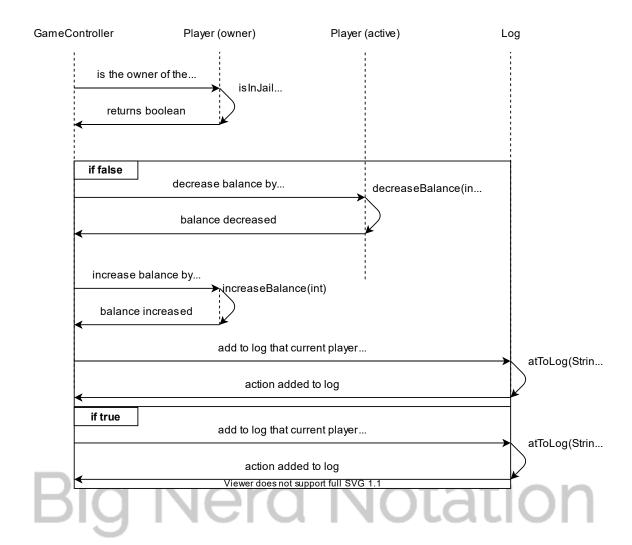
Big Nerd Notation

BoardPiece class structure

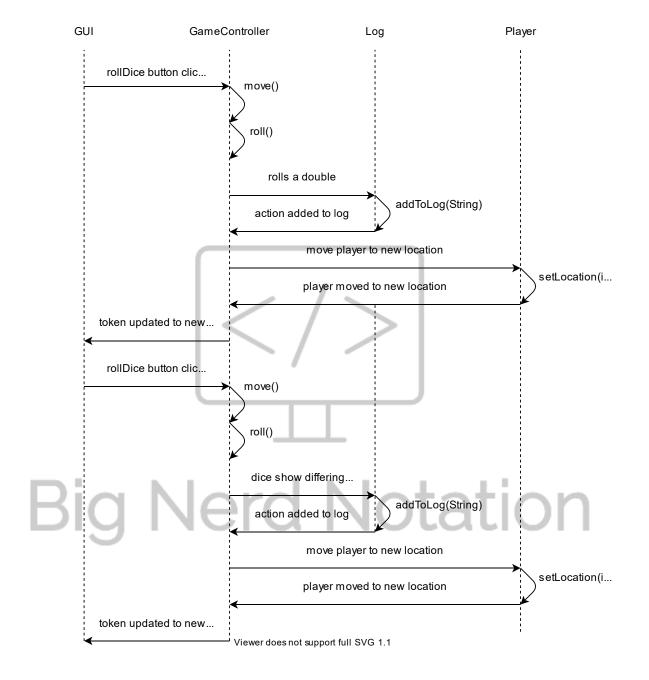


Big Nerd Notation

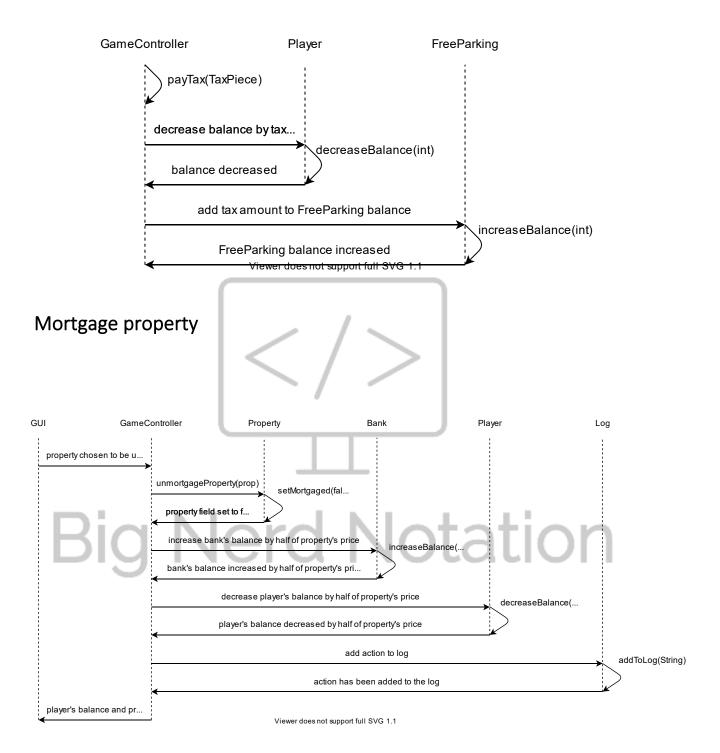
Renting



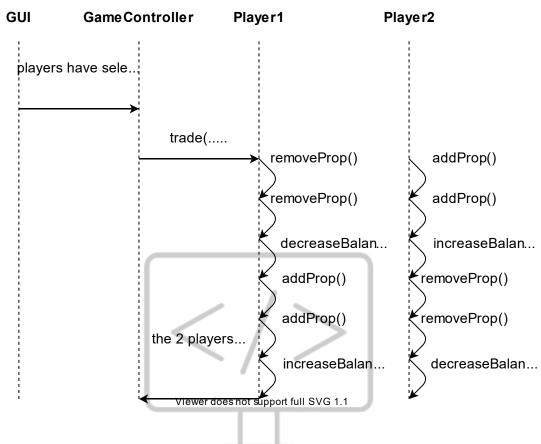
Moving



Pay tax



Trading



Ending turn

