These are the UML class diagrams for Milestone 1 of SYSC3110Monopoly
Student Name: Wahab Ansari
Student ID: 101060704
Group 39
Milestone 1

Jail

-interviewJailTurn: int
-timeAtJail: int = 3

+Jail(name: String, rent: int, price: int, color: Color, location: int)
+timeAtJail(player: Player, turnount: int): void
+timePass(player: Player): void

Board

-reader: Scanner
-boardConstructor: BoardConstructor
<<static>> -players: ArrayList<Player>
-propertiesArrayList: ArrayList<Properties>
-isValid: boolean
-saveInputSource: String

+Board(saveInputSource: String)
+getUserInputSource(): String
+getProperties(arrayList): ArrayList<Properties>
+setIsValid(valid: boolean): void
+setPlayers(playersFromBoard: ArrayList<Player>): void
<<static>> +newBoard(saveInputSource: String, gameNewNew: boolean): Board

Utilities

+Utilities(name: String, rent: int, owner: Player, price: int, colour: Color, location: int)
+getRent(): int

GO

+Go()
+giveMoney(player: Player): void

Properties

-name: String
-rent: int
-owner: Player
-price: int
-numHouses: int
-numHotels: int
-color: Color
-location: int

+Properties(name: String, color: Color, location: int)
+Properties(name: String, price: int, rent: int, color: Color , location: int)
+getNumTax(): int
+getRent(): int
+payRent(payingPlayer: Player): void
+getName(): String
+getLocation(): int
+getOwner(): Player
+setOwner(owner: Player): void
+getPrice(): int
+getColor(): Color
+setName(name: String): void
+getNumHouses(): int
+setNumHouses(numHouses: int): void
+getNumHotels(): int
+setNumHotels(numHotels: int): void
+getHousePrice(): int

LuxuryTax

+takeMoney(player: Player): void

IncomeTax

+IncomeTax()

Railroads

+Railroads(name: String , rent: int, owner: Player, price: int, colour: Color, location: int)
+getRent(): int

Game

<<static>> -MAX_PLAYERS: int = 8
-players: ArrayList<Player>
-propertiesArrayList: ArrayList<Properties>
-gameState: Enumeration
-colorPropertiesArrayList: HashMap<Color, ArrayList<Properties>>
-currentPlayer: Player
-reader: Scanner
-endTrue: Boolean
-jail: Jail
-DRAWER: Color
-board: Board
-boardConstructor: BoardConstructor

+Game()
+play(): void
+determineTruePlayer(): int
<<static>> +isNegative(s: String): boolean
-welcomeMessage(): void
+operateCommand(command: String): boolean
+addPlayer(player: Player): void
+setCurrentPlayer(player: Player): void
+getCurrentPlayer(): void
+getPlayerTurn(): void
+addProperty(property: Properties): void
+roll(): boolean
+purchaseProperty(): void
+purchaseWasaOwned(property: Properties): void
+checkNumOfActivePlayers(): Boolean
<<static>> +main(args: String[]): void

Player

-controlledProperties: List<Properties>
-ownsEntirelyB: Boolean
-balance: int
-ownsVrtrans: int
-name: String
-position: int
-color: Color
-inJail: Boolean
-turnsInJail: int
-bankruptStatus: Boolean

+Player(name: String, color: Color, startingMoney: int)
+getColor(): Color
+getName(): String
+getOwnsVrtrans(): int
+setOwnsVrtrans(ownsVrtrans: int): void
+getPosition(): int
+setInJail(): Boolean
+setInJail(inJail: Boolean): void
+getTurnsInJail(): int
+setTurnsInJail(turn: int): void
+setPosition(newPosition: int): void
<<static>> +newPlayer(name: String, color: Color, startingMoney: int): Player
+gainTerritory(property: Properties): void
+removeTerritory(property: Properties): void
+getControlledProperties(): List<Properties>
+removeAllControlledProperties(): void
+getOwnsBothUtil(): Boolean
+setOwnsBothUtil(ownsBothUtil: Boolean): void
+getBalance(): int
+setBalance(balanceamount: int): void
+increaseBalance(amount: int): void
+getBankruptStatus(): Boolean
+setBankruptStatus(status: Boolean): void

BoardConstructor

-board: Board

<<static>> +validateUtilSchemes(pathPath: String, errPath: String): boolean
+loadBoardFromMapFile(board: Board): boolean

<interface> boardInput

<<abstract>> +getString(prompt: String, defaultValue: String): String
<<abstract>> +getInt(prompt: String, min: int, max: int): int
<<abstract>> +getOption(prompt: String, options: Object[]): int
<<abstract>> +getString(input: String): boardInput