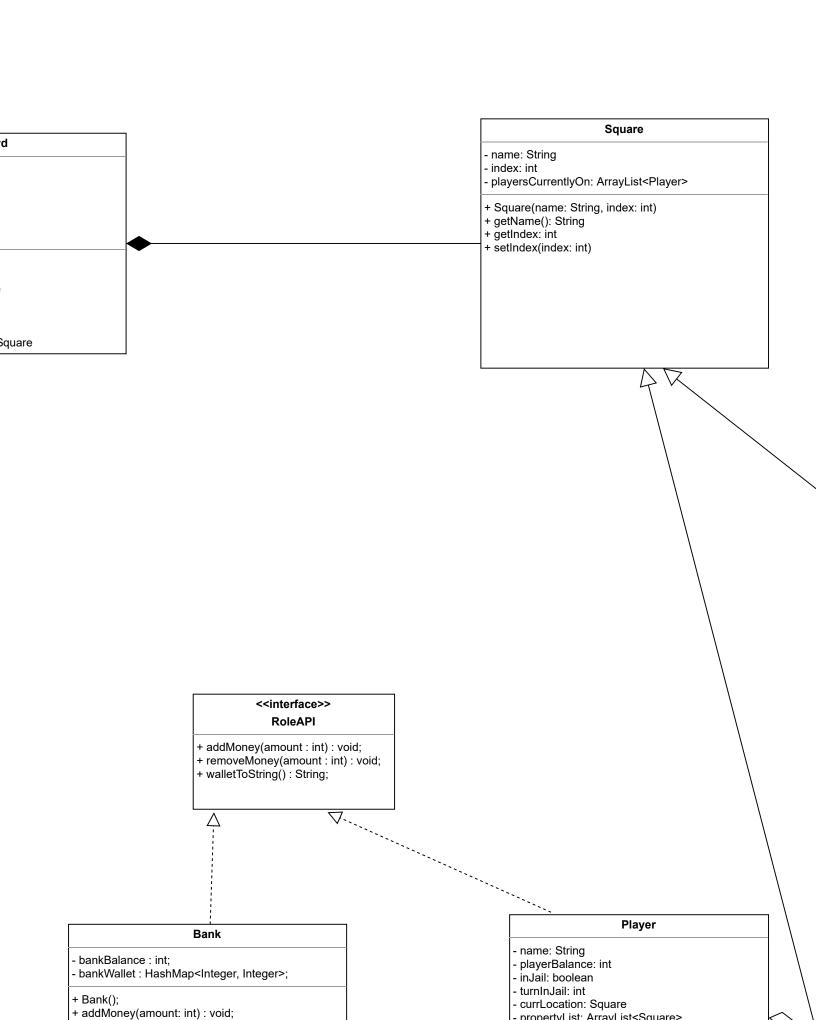
# Boar

- LENGTH: int
   aBoard: Array<Square>
   goToJail: Square
   jail: Square
   go: Square
   freeParking: Square

- + Board()
  + getGoToJail(): Square
  + getFreeParking(): Square
  + getLENGTH(): int
  + getGo(): Square
  + getJail(): Square
  + getSQUARE(index: int): S



Ngo Huu Gia Bao 101163137 Zakaria Ismail 101143497 Yuguo Liu 101142730 Gabriel Benni Kelley Evensen 101119814

# Extends

# PrivateProperty

- price: int - owner: Player - isOwned: boolean
- + PrivateProperty(name: String, index: int, price: int)
- + setOwner(player: Player) + removeOwner(): void
- + getOwner(): Player
- + getPrice(): int
- + isOwned(): boolean

# ColorGroup - countRed: int - countBlue: int - countOrange: int - countGreen: int - countBlack: int - count: int - countRed: int - countRed: int + ColorGroup() + isOwningASet(ArrayList<privateProperty>): boolean

#### Dice

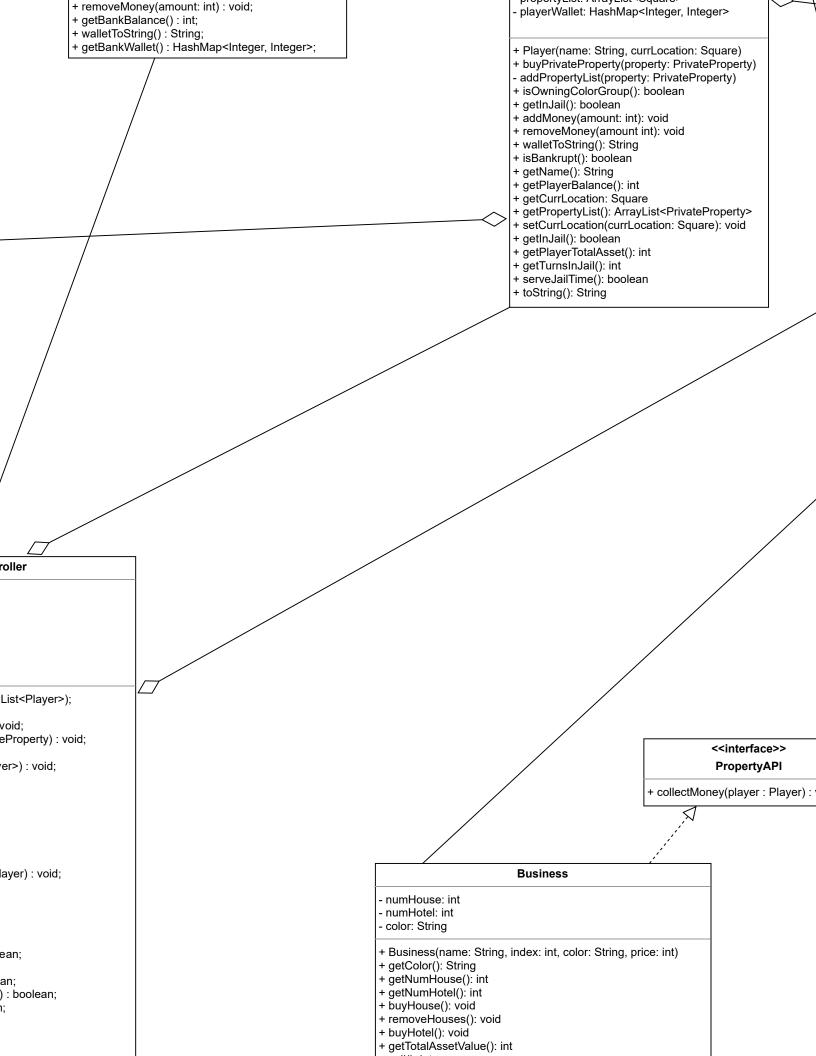
- SIZE : int;
- NUM\_SIZE: int;
- dice : int[];
- rand : Random;
- + Dice();
- + roll(): void
- + isDouble() : boolean;

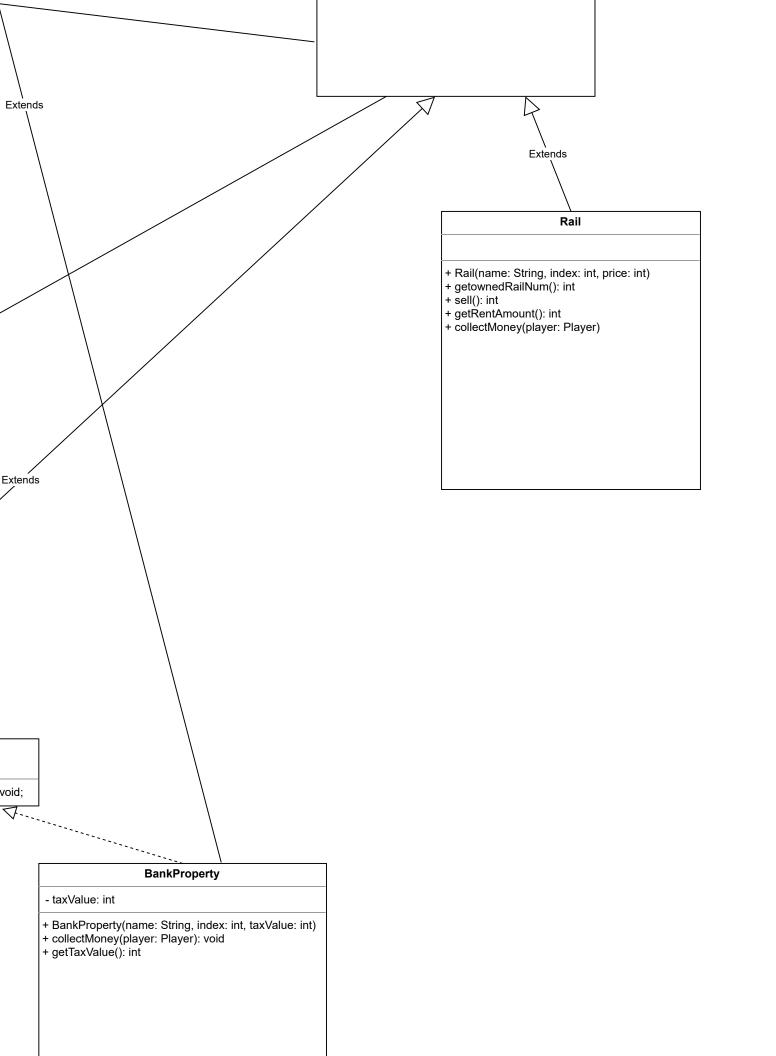
+ getNUM DICE(): int

- + getTotal(): int;
- + getDice(): int[];

# MonopolyCont

- GO\_REWARD : int;
- players : ArrayList<Player>;
- board : Board;
- bank : Bank;
- die : Dice;
- currentPlayer : Player;
- consecutiveDoubles : int;
- + MonopolyController(players : Array + sellProperty(property : Rail) : void;
- + sellProperty(property : Business) :
- + purchaseProperty(property: Privat
- + getPlayers() : ArrayList<Player>;
- + setPlayers(players : ArrayList<Play
- + getBoard(): Board;
- + setBoard(board : Board) : void;
- + getBank() : Bank;
- + setBank(bank : Bank) : void;
- + getDie() : Die;
- + setDie(dice : Die) : void;
- + getCurrentPlayer() : Player;
- + setCurrentPlayer(currentPlayer : P
- + getNextPlayer() : Player;
- + rollDie() : int[];
- + isSpeeding() : boolean;
- + sendCurrentPlayerToJail(): void;
- + moveCurrentPlayer() : boolean;
- + currentPlayerIsOnGoToJail(): bool
- + getGO\_REWARD() : int;
- + currentPlayerIsOnParking(): boole
- + currentPlayerIsOnOwnedProperty(
- + currentPlayerIsOnBlank(): boolear + determineWinner() : Player;
- + isGameEnded(): boolean





+setNumSide(numSide: int): void

### MonopolyGUIView

- controller: MonopolyController

- board: Board

- mainPanel: JPanel

- gb: GridBagLayout

- c: GridBagConstraints

- squares: ArrayList<JPanel>

- textPanel: JPanel - textLabel: JLabel

- showStatsBtn: JButton

- rollBtn: JButton

- buyBtn: JButton

- sellBtn: JButton

- endTurnBtn: JButton

- paxTaxBtn: JButton - feePaid: boolean

- diceRolled: boolean

- roll: int[]

- diceLabel1: JLabel

- diceLabel2: JLabel

# + MonopolyView()

- SquaresLayout(): void - addSquareToBoard(): void

- handleShowStatsBtn(e: ActionEvent): void

handleBuyPropertyBtn(): voidhandlePayTaxBtn(): void

- handleEndTurnBtn(): void

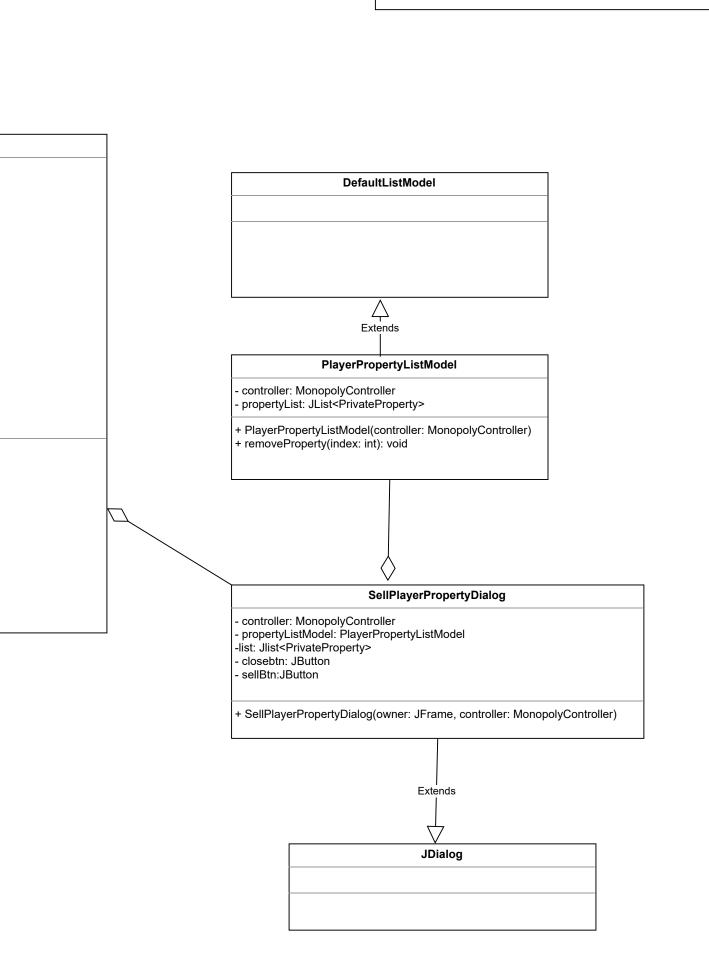
- handleRollDiceBtn(): void

- handleSellBtn(): void

- addButtonToBoard(): void

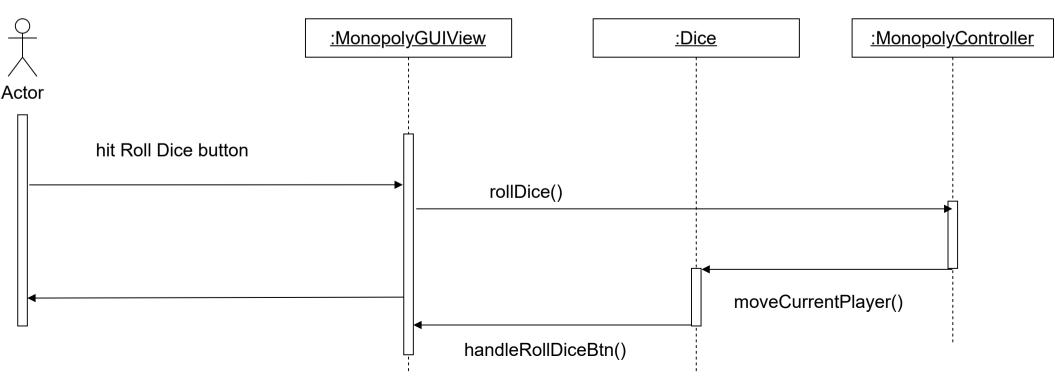
+ displayGUI: void + actionPerformed(e: ActionEvent): void

+ main (args: String[])



+ sell(): int + getRentAmount: int + collectMoney(): void + toString(): String

Ngo Huu Gia Bao 101163137 Zakaria Ismail 101143497 Yuguo Liu 101142730 Gabriel Benni Kelley Evensen 101119814



Tax Prompt (bankrupt) example:

Player X is currently at Square Y.

Player X rolls the dice. A 3 and 4 are rolled.

Player X has landed at Tax Square.

Player X, you have 5\$ cash, your net worth is 20\$, and the tax to

the bank is 50\$.

Player X cannot afford to pay the tax.

--GAME OVER--

Winner: Player K with a net worth of 1000\$

Prompts for rent should be similar

NOTE: there is also the Simply tell the use

Unowned property (car

Player X is currently at

Player X rolls the dice.

Player X, you have land

Player X, you have 100

Player X cannot afford

3. Display player status

Tax Pron

Player X

Player X

Player X

Player X 1. Pay th

Sell prDispla

Enter che

Unowned property (can afford) prompt example:

Player X is currently at Square Y.

Player X rolls the dice. A 3 and 4 are rolled.

Player X, you have landed at Unowned Private Property Square.

Player X, you have 100\$ cash, your net worth is 500\$, and the price of UPPS is 50\$.

- 1. Purchase UPPS and end turn
- 2. End turn without purchasing UPPS
- 3. Sell properties
- 4. Display player status

Enter choice:

Enter choice:

1. End turn

2. Sell properties

- SIZE : int - dice : int[]

+ Dice()

+ roll() : void

+ getTotal(): int

+ isDouble() : boolean

Dice

+ getDie() : int[]

#### MonopolyView

controller : MonopolyControllerplayers : ArrayList<Player>

+ MonopolyView()

+ play(): void

- promptSale(player : Player) : void

· displayStatus(player : Player) : void

# MonopolyController

- players : ArrayList<Player>

board : Boardbank : Bank

- die : Dice

- currentPlayer : Player

+ MonopolyController(players : ArrayList<Player>)

+ purchaseProperty(property : PrivateProperty) : void

+ sellProperty(property : Business) : void

npt (can afford) example:

is currently at Square Y. rolls the dice. A 3 and 4 are rolled. has landed at Tax Square.

you have 100\$ cash, your net worth is 500\$, and the tax to the bank is 50\$.

e tax and end turn

operties

y player status

oice:

e case of having enough netWorth but not enough cash. ser to sell some properties before ending their turn.

nnot afford) prompt example:

Square Y.

A 3 and 4 are rolled.

ded at Unowned Private Property Square.

0\$ cash, your net worth is 500\$, and the price of UPPS is 800\$. UPPS.

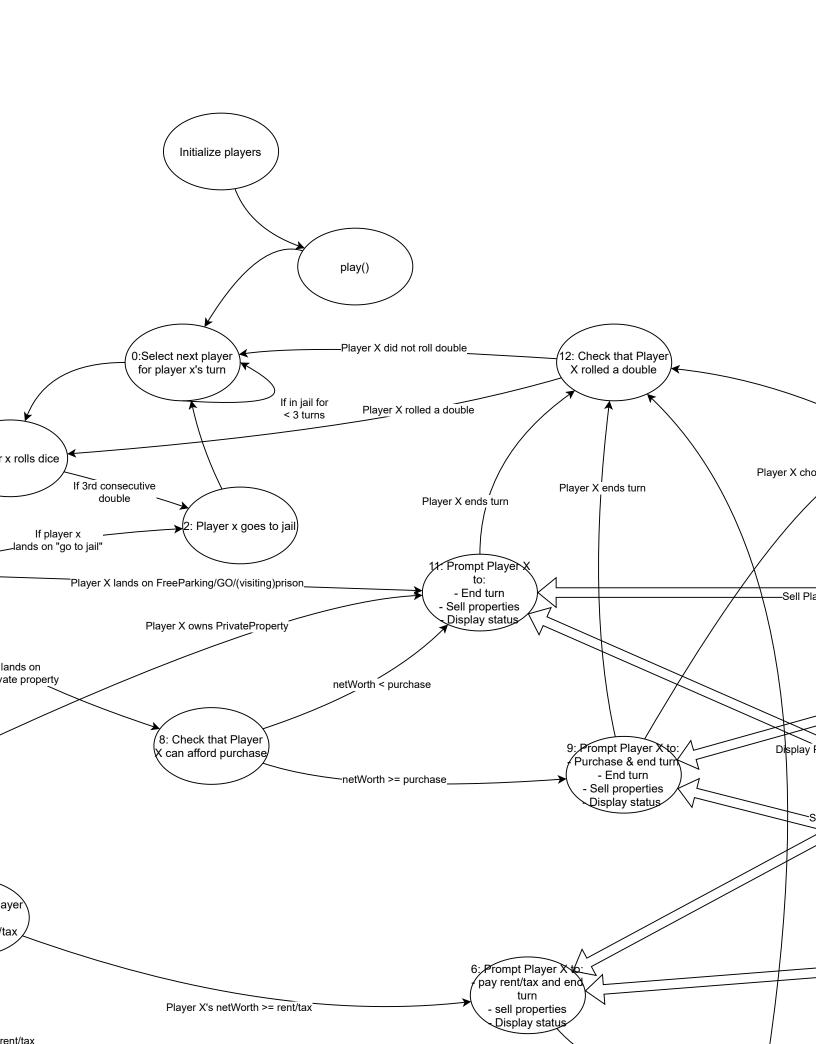
Note: I need a "visiting jail"

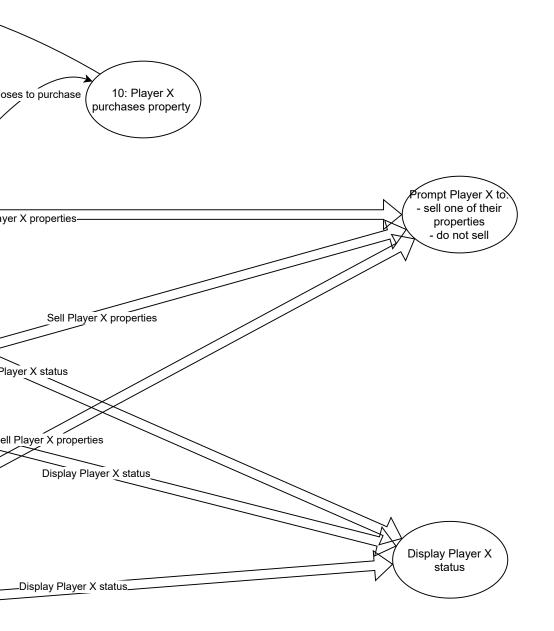
- list of Players will be defined and declared in MonopolyView via user input (customizable # players) and passed to Controller
- note: return value of functions is subject to discretion. if i cant think of one now i put void but if you think it should return then make it return
- sellProperty() is used by promptSale()
- i want a Die class so that I MonopolyView can access and present its data EDIT: nvm bobby wrote one poggers

3:Move player x forward k steps give GO money if passed Player X lands on owned private/bank propert Player X unowned priva 4:If PrivateProperty, check that Player owns the property Player X lands on owned private/bank property 5: Check that P Х can afford rent Player X's netWorth <

1:Playe

FIXME: i forgot to include case where player landsd on their own pprperty





-Use a "returnState" var to return to current stafter entering state with multiple entryways?
 - make a function that gives a prompt variant depending on parameters passed?
 - void promptPlayer(Player, Square PromptType.TYPE)
 note: PromptType is an Enum {AffordPurchas CannotPurchase, AffordRent, AffordTax}

Note: the act of paying taxes vs rent does not seem all that different to me.... Is there a way to me.... this one single function?



+ sellProperty(property : Rail) : void

Notes:-

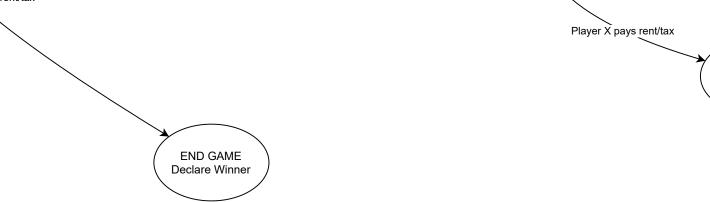
purchaseProperty does not check if the currentPlayer does already own the property.

purchaseProperty automatically gets a house for the square if a full colour set is owned by the currentPlayer

currentPlayer set to which ever Player is at the index 0 of the ArrayList<Player> (passed as a parameter in MonopolyController())

...after each turn you could put a new player at index 0

1.2)01 /10 1101110111



7: Player X pays rent/tax to Player Y/bank

