**Board***Current LCOM: 0.60*

**Board (ArrayList Players)**

* *Initialises board object*
* *Creates dealer instance*
* *Initialises player positions array and current turn variable*
* *Dealer object creates card object*
* *Create tile object and puts them into ArrayList Tiles*

**continueGame()**

* *Continuous do while loop until there’s only one player remaining i.e. game will end when winner is established*

**selectChoice**

* *Uses switch statement to call different methods based on player choice*
* *Menu options shown are:   
  Buy Property  
  Build House  
  Initiate trade  
  Sell House   
  Mortgage Property*

**getTile()**

* *tile number parameter, will return tile object*

**getPlayer**

* *player number parameter, will return player object*

**nextTurn**

* *Gets player object from players array list of index current turn*
* *It will then trigger select choice method from player class*
* *After player turn is done, increments current turn variable++*
* *Check if current turn is equal to length of player array list -1*
* *After player turn, sets current turn to 0*

**movePlayer(Index of player in player array list)**

* *Player rolls dice*
* *Board object gets instance current player position from player positions array, adds the random number from dice to player position array, gets tile object of that player’s new position*
* *Call to checkPlayerPosition*
* *After check is done, goes to next player’s turn*

**checkPlayerPosition(Tile landedTile)**

* *Method call, used getTileType method from tile abstract class*
* *If property tile, will trigger methods*
* *If wildcard tile, used getWildCardType method from the wildcard tile class*
* *If wildcard tile is security, dealer.drawsecuritycard, vice versa for attack*

**Player***Current LCOM: 0.69*

**Player**

* *Initialises each player that was created*
* *Includes name, security level, money, and an array list of owned properties*

**getName()**

* *Standard getter that returns player name*

**incrementAttribute(String parameter type e.g security/money, int Value)**

* *Increments specific player attribute based on which type of card is drawn*
* *i.e., if a security card is drawn from the bonus card pile, the appropriate parameter will be passed in from the bonus card subclass*

**decrementAttribute(String parameter type e.g security/money, int Value)**

* *decrements specific player attribute based on which type of card is drawn*
* *i.e., if an attack card is drawn from the damage card pile, the appropriate parameter will be passed in from the damage card subclass*

**checkMoney**

* *Method that performs a check of a player’s money*

**checkSecurity**

* *Method that performs a check of a player’s security level*

**purchaseProperty(Tile property)**

* *Allows a player to purchase property they’ve landed on after the dice roll*
* *Takes tile parameter from Tile class*

**buildHouse**

* *Builds house(server) on selected property*
* *Property tile will be passed in as a parameter from the PropertyTile class*

**initiate Trade   
TBC**

**sellHouse**

* *This method will allow the selling of a house(server)*
* *E.g. if a player wishes to mortgage a property, this method will be used to sell whatever houses they have built on the specified property, in exchange for half of what it was originally bought for*

**mortgageProperty  
TBC**

**Tile (abstract)***Current LCOM: 0.50*

**Tile()**

* *Initialises each tile*
* *Assigns a tile number, tile name, and tile type*

**getTileNumber**

* *Standard getter that returns tile number*

**getTileName()**

* *Standard getter that returns tile name*

**getTileType()**

* *Standard getter that returns tile type*

**WildCardTile***Current LCOM: 0.00*

**wildCardType(String tileType)**

* *Initialises wildcard tile based on the parameter*

**getWildCardType()**

* *Standard getter than returns the wildcard type*

**JailTile**

**IncrementJailTurn(int playerNumber)**

* *Increments player turn number in jail until they’ve reached a max of 3 turns*

**resetTurnCounter(int index)**

* *Once a player has reached the required jail turns or they roll a double six, this method will reset the turn counter to 0 after the player has left in jail status*

**Property Tile***Current LCOM: 0.67*

**PropertyTile()**

* *Initialises each property tile with property cost, player house level, linked properties*

**getOwner(): String**

* *Standard getter that returns the owner of a property*

**setOwner(): void**

* *Once a property has been purchased by a player, this method will set the player who bought the property as the owner*

**checkGroupOwnership(Player player)**

**increaseHouseLevel()**

* *If a player wishes to build a house(server), this method will increment the house level by 1*

**triggerRent()(Player player)**

**Card(abstract)***Current LCOM: 0.44*

**Card()**

* *Initialises card with the card name, description of the card and targetSelf, which is a boolean, if the card can be applied to the player that has drawn the card i.e. bonus security*

**getCardName()**

* *Standard getter that returns the card name*

**getCardDescription**

* *Standard getter than returns the card description*

**triggerEffect()**

* *Method that will trigger the effect of the card*

**CardHandler***Current LCOM: 0.33*

**cardHandler()**

* Constructor

**drawAttackCard()**

* *This method will draw an attack card from the attack card array list*

**drawSecurityCard()**

* *This method will draw a security card from the security card array list*

**createAttackCard**

* *Creates attack card and adds it to attack card array list*

**createSecurityCard()**

* *Creates security card and adds it to the security card array list*

**shuffleCard(Card drawnCard)**

* *Method will be called when a card is pulled from the deck*
* *This method will then check the array list and insert it at the end of the array list*

**DamageCard***Current LCOM: 1.00*

**DamageCard()**

* *Initilises damage card with the attack value and what type of attack is associated with each card*

**triggerEffect()**

* *Trigger effect method will initiate the effects of bonus card*

**BonusCard()***Current LCOM: 1.00*

**BonusCard()**

* *Initialises bonus card with the bonus value and what type of bonus is associated with each card*

**triggerEffect()**

* *Trigger effect method will initiate the effects of bonus card*

**UpgradePropertyCard***Current LCOM: 0.00*

**UpgradePropertyCard()**

* Constructor

**triggerEffect()**

* Trigger effect method will upgrade a property to the next server level