# SYSC3310 Lab Group 16

# Documentation

Author: Gilles Myny – 101145477

**Game Class:**

The Game class was designed by Frank Dow and creates a Board object and an ArrayList of Piece objects. The Piece object names and numbers are user inputted for customizability. As in the normal game, each Piece take their turn by calling the nextTurn() method until each Piece has had a turn or quits before the start of their turn. All possibilities of the user's Piece are covered by the numerous if-else statements in the nextTurn() method. i.e., the Piece can only buy a Property object if nobody else owns it and if they have sufficient funds; if these conditions are not met a list of options are given to the user to continue. Delegation is used in the Game class to keep the nextTurn() method easy-to-read and clean by calling other methods that deal with payment (pay() method), purchasing a Property object (purchaseProperty() method), printing information (printState() method), and moving their piece through the board (movePiece() method).

**Board Class:**

The Board class was designed by Gilles Myny and creates a Map to group the Property objects and their position adequately to allow other classes to utilize this class with ease. The Board constructor creates all 22 property objects and adds them with their respective position on the Board to the Map. This class also allows other classes to easily access Tile information such as a Tile object and Tile placement on the board with the getTile() and getTilePosition() methods.

**Piece Class:**

The Piece class was designed by Jeremy Trendoff and models the business logic each Piece the player controls. Each Piece object has a String for its name, an Integer for its balance, current Tile object (Tile), a Boolean for bankruptcy state (isBankrupt), and a List of tiles owned (ownedTiles). Originally the Piece class split up its owned Tiles by type (Property, Utility, Railroad) but for performance benefits this wasn't incorporated as its advantages did not outweigh its complexity. The Piece class has a single constructor where its name, balance, and currentTile position is initialized. It also contains getters for each of its attributes listed above. The updatePosition() method is used to update the currentTile attribute of the Piece object. Furthermore, the balance can be updated using the addFunds() and deductFunds() methods, it can also add or remove Tile objects from the ownedTiles list and specify each individual Tile object with the getOwnedTile() method. Finally, the Piece object can print out its currently owned tiles. In the future it could make sense for the Piece object to print out its current state.

**Tile Class:**

The Tile class was designed by Joshua Gatto and models the several types of Tiles in the game on monopoly. An abstract Tile class was used to model the basics of each specific tile type and can easily be expanded for future milestones. For this milestone we decided as a team to have the following Tile subclasses: a Property Tile, a Utility Tile, and a Railroad Tile. However, for this milestone we are only using the Property tiles and will put the other tiles and some unadded ones to use. The Property class in particular keeps track if its been owned with a Boolean, and which Piece object is the owner of the Property. Along with simple getters and setters are a few overridden methods for specific usages.