Object-Oriented Programming CSCE A222

Final Project: Monopoly

Due

Points:

Implement the game of monopoly using object-oriented programming techniques and Java. At a minimum, your solutions should have the classes: Square, Tax, BuyableSquare, Property, Utility, Railroad, Person, Player, GamePiece, Card, Monopoly, and a class with a main method. It is up to you to figure out what should go in each of the classes and their relationship with one another. **NOT** all monopoly rules will be implemented or are original. **DO NOT** upload all the classes in one java file. Avoid duplicate code. Comment your methods with a small description (no need for input/output comments). Comment anything you think might be confusing to the reader.

Implement however you wish as long as you meet the requirements noted on this document.

Create and include the UML Diagram for your program and a short write up (3 paragraphs minimum) of the design choices you made and any problems you encountered.

**Addition Requirements & Information:**

* Read in the provided txt files to set up your gameboard.
* Original win conditions do not need to be implemented. Simulate the game for 7 turns and then display the winner (most money) instead.
* Trading/Selling property does not need to be implemented.
* No bonuses need to be implemented if a player owns all properties of the same color.
* Support up to 5 players at the same time.
* Gain $200 if player lands on **OR** passes go.
* Players should start with $1500.
* Properties can only have up to 4 houses (no hotels).
* Houses cost $50.
* Jail time is 1 turn.
* All squares should be implemented with their appropriate action.
* Simple non-graphical interface with relevant options (roll, print current money, print owned squares, buy houses for owned property, etc.).
* Players traverses the board correctly (update their current location).
* Player can buy square at current spot if it is not owned & buyable.
* Landing on a square owned by another player should correctly deduct the rent from the current player’s wallet and add it to the owner’s wallet.
* Properties rent changes with the number of houses on it (numbers given in txt file).
* If a player lands on a Chance or Community Chest square, pick a random card from the relevant Card array and execute the appropriate action.
* Tax Squares should deduct the specified amount from the players wallet.

**Class Specific Notes:**

*Monopoly*

* Decide how to store your game board and store it in this class (array, linked list, etc.) along with any game specific logic.
* Should have 2 arrays for the chance & community cards.

*Square*

* You can use this class for the four corners of the board & the community and chance squares (You do not need to split up go, jail, free parking, go to jail, chance, or community chest into 6 different classes).

*Railroad*

* Railroads rent is based off of the number of railroads owned ($25, $50, $100, $200).

*Utility*

* Rent should be determined by the formula: (dice roll \* multiplier).

*Player*

* Player class should have an instance of GamePiece.

*GamePiece*

* Pick 5 types of potential game pieces of your choice and store it as a string variable in the GamePiece class.
* Your class should randomly pick one of the game piece types and assign the string variable to it.
* No duplicate pieces allowed between players.

**Text File Formats & Information:**

*community.txt*

* Advance to Go: Move player to go square.
* Doctor Fees: Subtract $100 from player’s money.
* Go to Jail: Move player to jail square & skip next turn.
* School Fees: Subtract $200 from player’s money.
* PFD: Add $350 to player’s money.

*chance.txt*

* Go Back Three Spaces: Move player 3 spaces backwards
* Advance to Illinois Ave.: Move player to Illinois Ave.
* Advance to Nearest Utility: Move player to nearest utility.
* Advance to Nearest Railroad: Move player to nearest railroad.
* Go Back Seven Spaces: Move player 7 spaces backwards.

*squares.txt*

Values are separated by commas. First number identifies what class it corresponds to.

* Square (1): Square name
* Property (2): Square name, color, price, rent, price w/ 1 house, price w/ 2 houses, price w/3 houses, price w/ 4 houses.
* Tax (3): Square name, tax price
* Railroad (4): Square name, rent
* Utility (5): Square name, price, multiplier value

**Extra Credit:**

Print the current state of the board every turn in a meaningful way.

**Notes:**

START EARLY! This is not as challenging as it may appear to be. Do not be afraid to ask for help, you have already paid for tutoring with your tuition so use them.

**Tutoring Schedule:**

https://www.uaa.alaska.edu/academics/college-of-engineering/student-opportunities/tutoring.cshtml