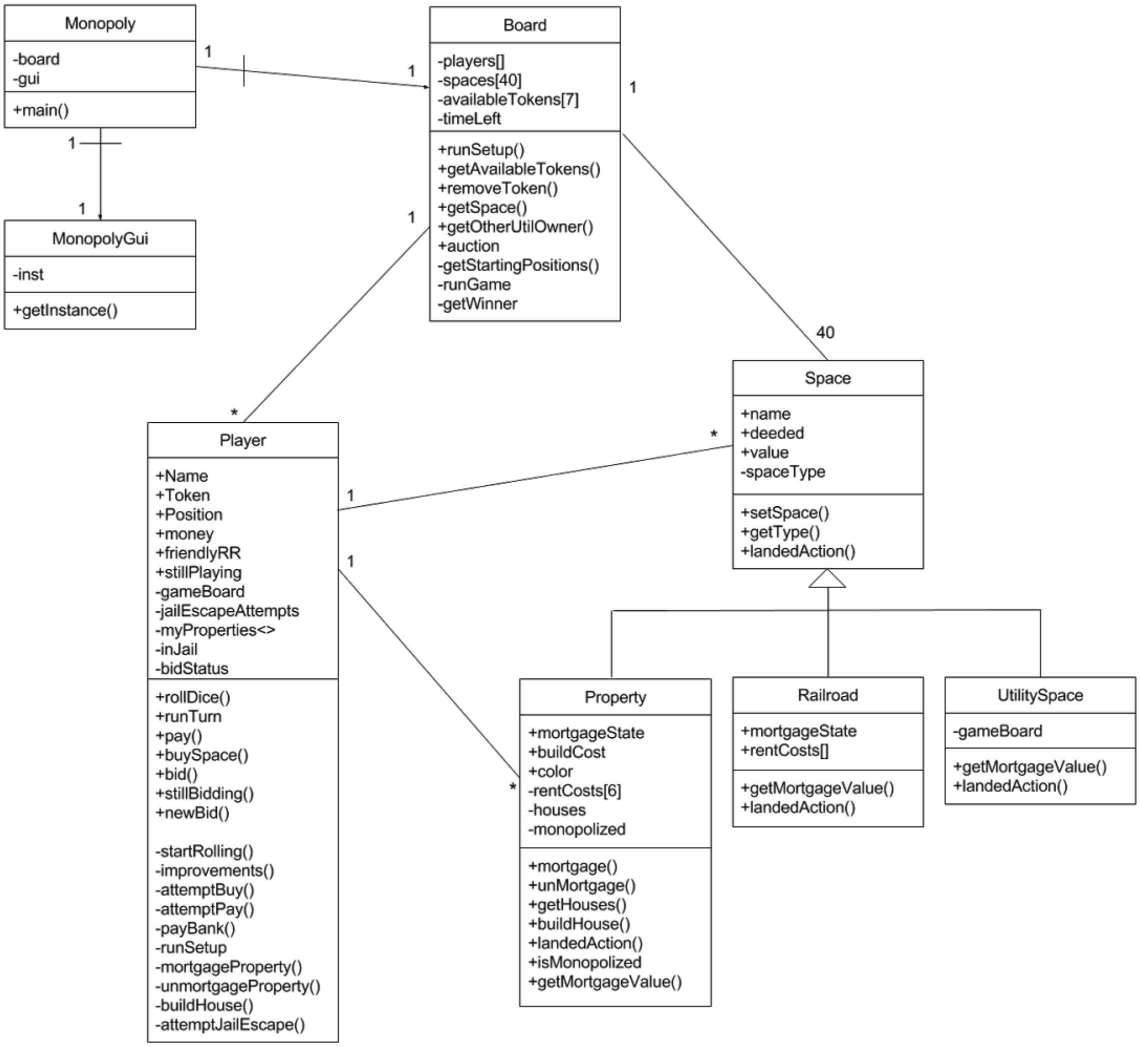

UML Description:

Monopoly: Our driver class that runs the program and interacts between the board and the GUI.

MonopolyGui: Class to control all of the graphical functionality of the game.

- getInstance(): This function returns an instance of the class so it can be used to draw graphics.

Board: Class that is in primary control of the game mechanics and facilitates the interactions between the players and the game spaces.

- runSetup(): Builds the board and calls for the players to be created.
- getAvailableTokens(): Returns an array of the unchosen tokens so a player may choose one.
- removeToken(): Removes a token from the array so that other players can’t choose it.
- getSpace(): Returns a reference to a space from its position on the board.

- getOtherUtilOwner(): Returns the owner of the utility not sent into the function so that the multiplication factor of a utility's cost can be calculated.
- auction(): Begins an auction for a property, railroad, or utility that a player chose not to buy (or couldn't afford).
- getStartingPositions(): Has each player roll a die and then sorts their play order based on descending order.
- runGame(): The main game loop that runs until the turn counter reaches 40.
- getWinner(): Determines the winner of the game based on the players values.

Space: This is the superclass for every space on the board.

- setSpace(): This function can be used to set up a space without using the constructor.
- getType(): Returns the type of the space.
- landedAction(): Runs the necessary code for when the space is landed on and returns the cost of landing on the space.

UtilitySpace:

- getMortgageValue(): Returns the value of mortgaging this utility.
- landedAction(): Overwrites the space function for how the utility acts when it is landed on.

Railroad:

- getMortgageValue(): Returns the value of mortgaging a railroad.
- landedAction(): Overwrites the space function for how a railroad acts when it is landed on.

Property:

- mortgage(): Mortgages the property giving the player money and invalidating it's landedAction function.
- unMortgage(): Reactivates the space.
- getHouses(): Returns the current number of houses on the property.
- buildHouse(): Adds a house or hotel to the property
- landedAction(): Overwrites the space function for how the property acts when it is landed on.
- isMonopolized(): Returns true if all the properties of this group are owned by the same player.
- getMortgageValue(): Returns the value of mortgaging this property.

Player:

- rollDice(): Returns an integer of a fake dice roll between 1 and 12 including a check for if each dice rolled the same value.
- runTurn(): Runs a single turn for the player.
- pay(): Allows the player to send a specified amount of money to another player.
- buySpace(): Transfers the deed of a space to the player while subtracting the cost from their money.
- bid(): Allows the player to bid on an auctioned property.
- stillBidding(): Determines if the player is still bidding or not.
- newBid(): Resets the bid status of the player.
- startRolling(): Runs the roll and subsequent payments of the player.

- improvements(): Handles the player improving a property.
- attemptBuy(): Determines if the user can buy a property and does so if they can.
- attemptPay(): Determines if the user can pay another player and mortgages properties or bankrupts the player if they can't.
- payBank(): Used when the player needs to pay for income or luxury tax.
- runSetup(): Sets up a player with a name and token.
- mortgageProperty(): If the property can be mortgage this function does so and gives the player the correct amount of money.
- unmortgageProperty(): Reverts the mortgage process if the player can afford it.
- buildHouse(): Builds a house or hotel on a property if this is a valid option.
- attemptJailEscape(). If the user is in jail this function runs an attempt at escaping once the user chooses not to pay the fee. Handles if the user has tried three times already.

Google Drive link: https://docs.google.com/document/d/1sgWjmz8Cyji7whmintFNVI2Hcf9_OcPIg66ooi8refg/edit?usp=sharing