EE 205 Final Project: Monopoly
Program Specifications

This idea of the final project was to create a program based on the game Monopoly. The game can be playable with 2-4 players. The following classes and functions are some of the requirements in order to make this program work:

- **Classes**
  - **BOARD**
    - **Data**
      - Entries being the different Properties, Utilities, etc.
      - Player positions
    - **Functions**
      - changePlayerPos
        - Changes a players position on the board
  - **updatePosition**
    - Changes the position of the player on the board
  - **BANKER**
    - **Property, Railroad, & Utility**
      - **Data**
        - Name, Ownership, Rent amount, House/Hotel status, Set info
      - **Functions**
        - buyHouse
          - Checks if the player has the entire set of properties, then adds a house
        - changeOwner
          - Changes the owner of the property
        - checkOwner
          - Checks who the property belongs to
        - setRent
          - Set the amount of rent that a non-owner player pays when he lands on the block
        - changeBalance
          - Updates the dollar amount of the player
        - addProperty
          - Adds a property to the list of properties owned by the player
        - subtractProperty
          - Removes a property from the list of properties owned by the player
  - **ACTION BLOCKS** (Cards drawn ???)
    - Go to Jail, Chance, Community Chest, Income Tax, etc.
    - **Data**
      - **Type of Block**
    - **Functions**
      - sendToJail
        - Changes the position of the player to the jail block
      - drawChance
        - Draws a “card” from the Chance deck
      - drawComChest
        - Draws a “card” from the Community Chest
- Draws a “card” from the Community Chest deck
  - taxPlayer
  - If the block is a tax income block, subtracts the proper amount from the player’s balance

- PLAYER
  - Data
    - Position, Balance, Properties owned
  - Functions
    - addProperty
      - Adds a property to the list of properties owned by the player
    - subtractProperty
      - Removes a property from the list of properties owned by the player

- GENERAL
  - rollDice
    - Simulates a random roll of two die
  - getNumPlayers
    - Gets the number of players who are playing the game
  - Menu functions – Trade properties, etc.

- Any other functions added by the respective teams to make this game FUN and EXCITING.

- THE GAME ENDS WHEN **ONE** PLAYER REMAINS (MEANING ALL THE OTHER PLAYERS GO BANKRUPT)
  - >=D