Splendor – OOP Design

\*This implementation of Splendor will use random cards that are within the same range as the actual cards – maybe this will change if think of easy way to implement cards, but I kind of like the idea of random cards.

\*Try to think of places where we can implement data structures that will help us store/retrieve data and/or play the game.

Classes

1. Player
   1. Private Variables
      1. Permanent Resources – (5 separate data variables or together within a struct?)
         1. Black
         2. White
         3. Red
         4. Blue
         5. Green
   2. Public Variables – Q: will non-member functions need to access these resources? – Situation I can think of is if you want to see the number of resources of all other players. But then will permanent resources need to be public as well?
      1. Temporal Resources
         1. Black
         2. White
         3. Red
         4. Blue
         5. Green
         6. Yellow
      2. Reserved Cards – can this just be a vector of pointers to cards?
   3. Functions
      1. Helper Functions – interact with data variables
         1. addResources() – (will we need function for each type of resource?)
            1. addResources will increment the temporal resources after drawing
         2. removeResources – should it be it’s own function or part of buyCard?
         3. checkNobles
      2. Actions – things player can do in a round
         1. buyCard
            1. calls removeResources and adds permenant resources to player
            2. deallocates memory for card that was bought
         2. getTokens
            1. calls addResources to change data vars
         3. reserveCard
            1. reserves a card
         4. endTurn
            1. calls checkNobes to see if any can be claimed.
            2. Passes control to next player/bot
2. Card – All vars need to be const.
   1. Public Variables
      1. Level
      2. Points
      3. Cost – struct
         1. Total
         2. Black
         3. White
         4. Red
         5. Blue
         6. Green
3. gameBoard
   1. Public Variables
      1. Vector of pointers to cards – Level1Cards
      2. Vector of pointers to cards – Level2Cards
      3. Vector of pointers to cards – Level3Cards
      4. Vector of pointers to Nobles – Tiles
         1. Should nobles just be level 0 cards?
      5. Vector of pointers to players
   2. Functions
      1. displayBoard
      2. replaceCard

Functions

1. Initialize Game
2. Generate Players
3. Generate Card – if we generate randomly each time we only have to have 4 cards created at a time
4. Play Game