

## **Mastermind:**

The process of making the design for this game was largely educated by watching videos about how to play the game and reading the manual for how to play. In general, as it is an object oriented analysis, breaking the game down into the constituent parts, in this case, the specific pegs, as well as the board, which will be made of 10 rows. The pegs will be required to hold a colour value, and will exist in arrays defined within the Row class, which will handle the behavior of ordering and storing the guess rows. The Row class will also be important with it's integration with the Check class which will handle all operations pertaining to checking the correctness of each row, and reporting the necessary statistics like the number of correct guesses per row. The check class will also be necessary for calling the victory condition, when the positions and colours of all 4 Pegs within a singular Row match the master code.

### **Peg Class**

- **Responsibilities**
  - **Will be a skeleton object that holds a color**
- **Collaborators**
  - **4 of these objects will exist in each instance of a Row object**
  - **The methods of getting data from the Peg object will be used in the Check class**

### **Row Class**

- **Will create 4 unique Peg objects each instance, and hold those 4 objects within an array. This will be important for holding, displaying and receiving each guess.**
- **Responsibilities**
  - **Will have all the information about one guess of the code**
  - **Creation and display of each guess**
- **Collaborators**
  - **Each line will be a list of 4 Peg objects**
  - **Will work with the Check class to check the similarities between Lines**

### **Check Class**

- **Will check the correctness of each individual guess against the master code. Will also show the number of correct position and color guesses**
- **Responsibilities**
  - **Will return statistics about the guess made when compared to the master**
  - **Will be responsible for triggering the win condition in the game when the code is correctly guessed**
- **Collaborators**
  - **The input for each Check is two line objects, one the master code, and the other the guess code**

### **Game State Enum**

- **Will be an enum that holds game states**
- **Responsibilities**
  - **Differentiating different parts of the game**
  - **Win and Loss conditions**
- **Collaborators**

- Will be set initially by the instancing of a Board object
- Will be changed by the result of the Check class central method

#### **Board Class**

- Will initialize the array in which the game is played, as defined by the number of guesses the user wants
- Responsibilities
  - Creating and holding an List of the Line objects
  - Reporting past guesses
- Collaborators
  - Will be made up of a List of line objects

#### **Mastermind Class**

- Will be responsible for the actual playing of the game, initializing and starting games
- Responsibilities
  - Taking user input to start playing the game and exiting once the game completes
- Collaborators
  - All classes within the code