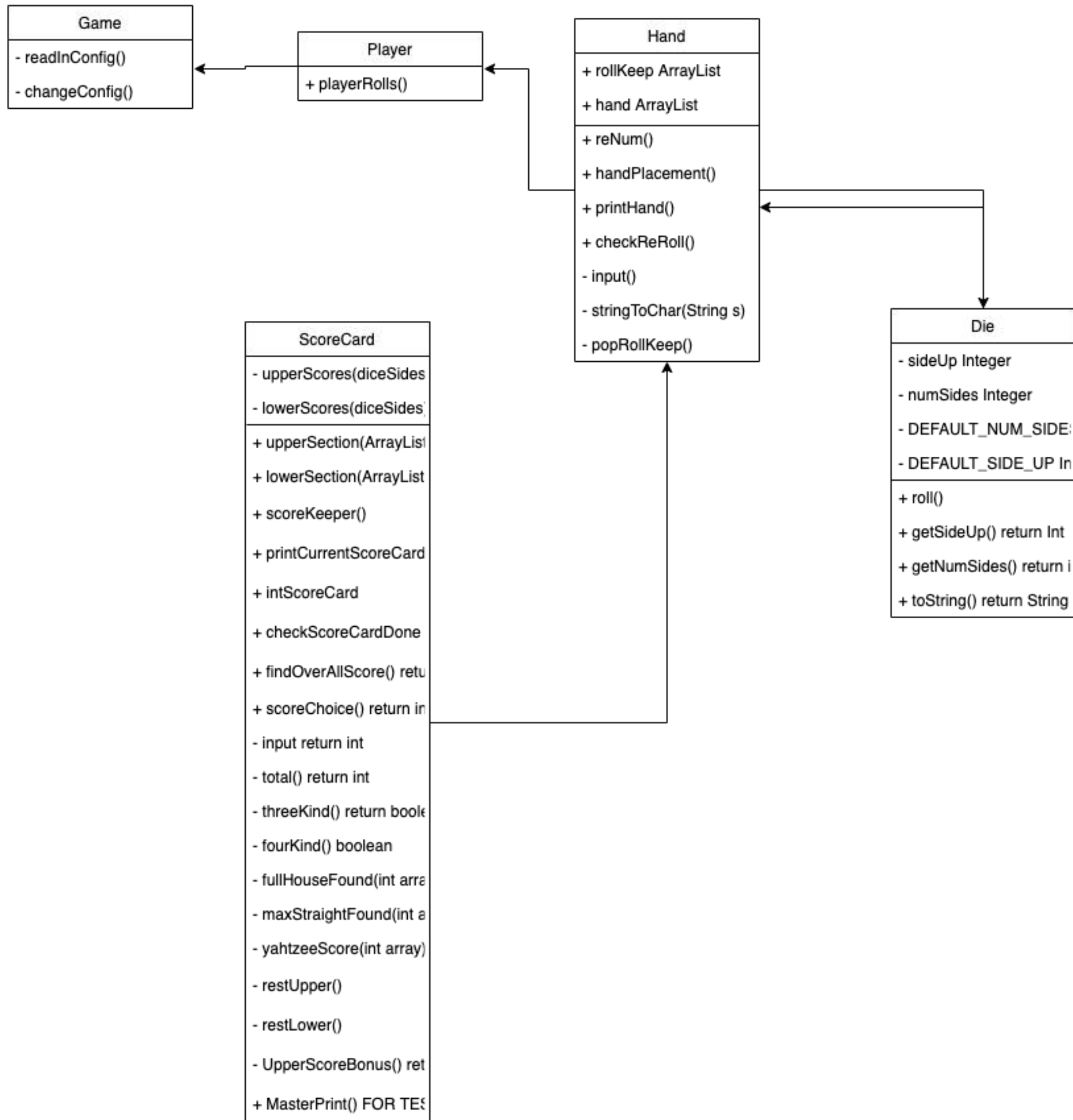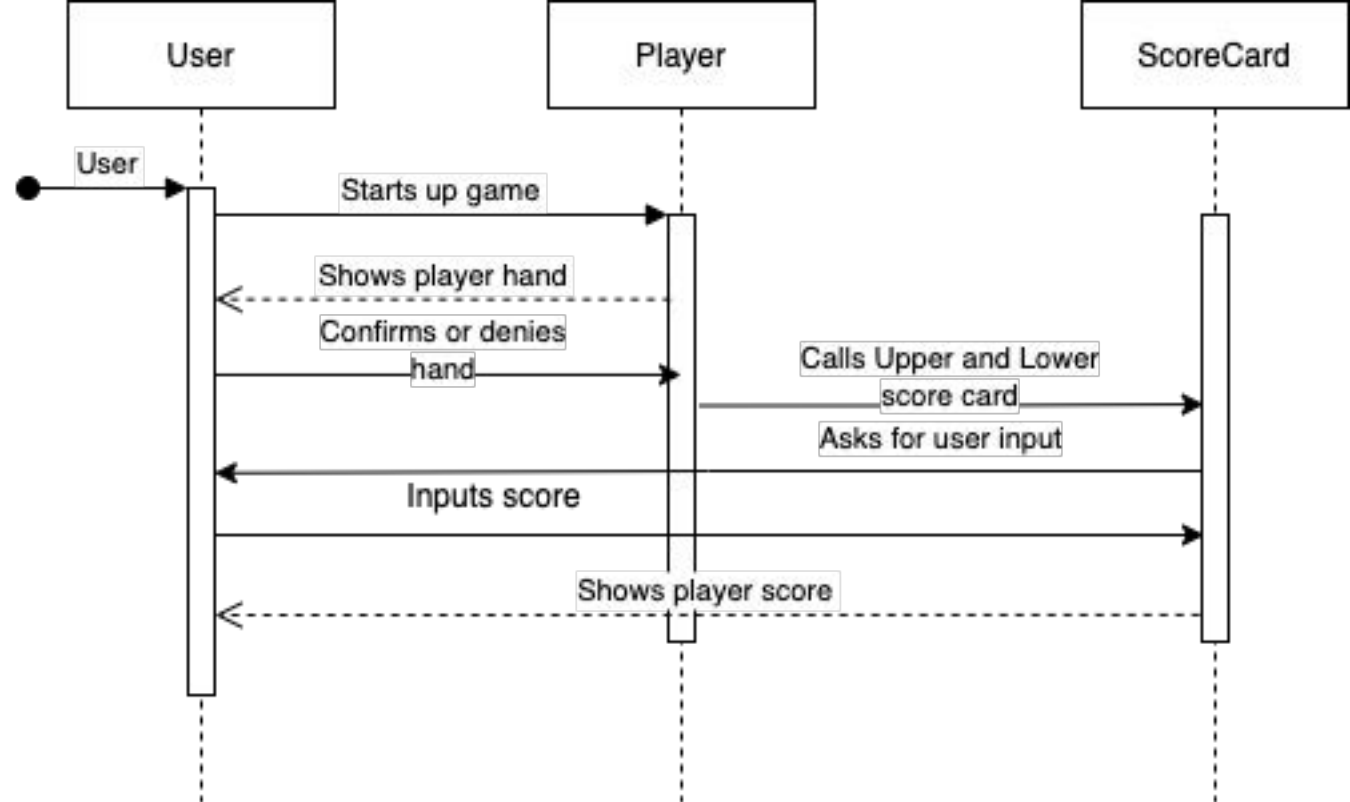# Fully Playable Single Player Yahtzee

Making a fully playable from start to finish single player Yahtzee game in the terminal. In making a fully playable Yahtzee game I discovered that arrays cannot do everything and sometimes you need an Array List. With that I was able to make the Yahtzee game with some tweaking to existing functions and creating a few more in the ScoreBoard class I have. By having my totalScoreCard be an ArrayList it allowed me to resize it along with setting each value to -1 when initialized. By having the values be -1 it allows 0 to be used as a score and then not show up in possible score choices and then appearing on the score card when its printed. A major design I had was to create a new ArrayList for my scorecard and then initialize it with -1. This was to check for scores that have been entered that could be zero and so forth as said above. Another major design modification was to change the function ScoreChoice and how it processes the number given in. It now checks to make sure that that number has not been entered in before. If the number was entered before it prompts the user to enter in a new user. Since I have finished the project before the due date everything I have done is done to how I would like it.

Yahtzee

Game
- readInConfig()
- changeConfig()

Player
+ playerRolls()

Hand
+ rollKeep ArrayList
+ hand ArrayList
+ reNum()
+ handPlacement()
+ printHand()
+ checkReRoll()
- input()
- stringToChar(String s)
- popRollKeep()

Die
- sideUp Integer
- numSides Integer
- DEFAULT_NUM_SIDE
- DEFAULT_SIDE_UP In
+ roll()
+ getSideUp() return Int
+ getNumSides() return i
+ toString() return String

ScoreCard
- upperScores(diceSides
- lowerScores(diceSides
+ upperSection(ArrayList
+ lowerSection(ArrayList
+ scoreKeeper()
+ printCurrentScoreCard
+ intScoreCard
+ checkScoreCardDone
+ findOverAllScore() retu
+ scoreChoice() return in
- input return int
- total() return int
- threeKind() return boole
- fourKind() boolean
- fullHouseFound(int arra
- maxStraightFound(int a
- yahtzeeScore(int array)
- restUpper()
- restLower()
- UpperScoreBonus() ret
+ MasterPrint() FOR TES

User	Player	ScoreCard

User

Starts up game

Shows player hand

Confirms or denies hand

Calls Upper and Lower score card

Asks for user input

Inputs score

Shows player score

Process repeats till totalScoreCard is full