Wesley Muehlhausen  
CPSC 224  
Feb 28, 2020  
Homework 3

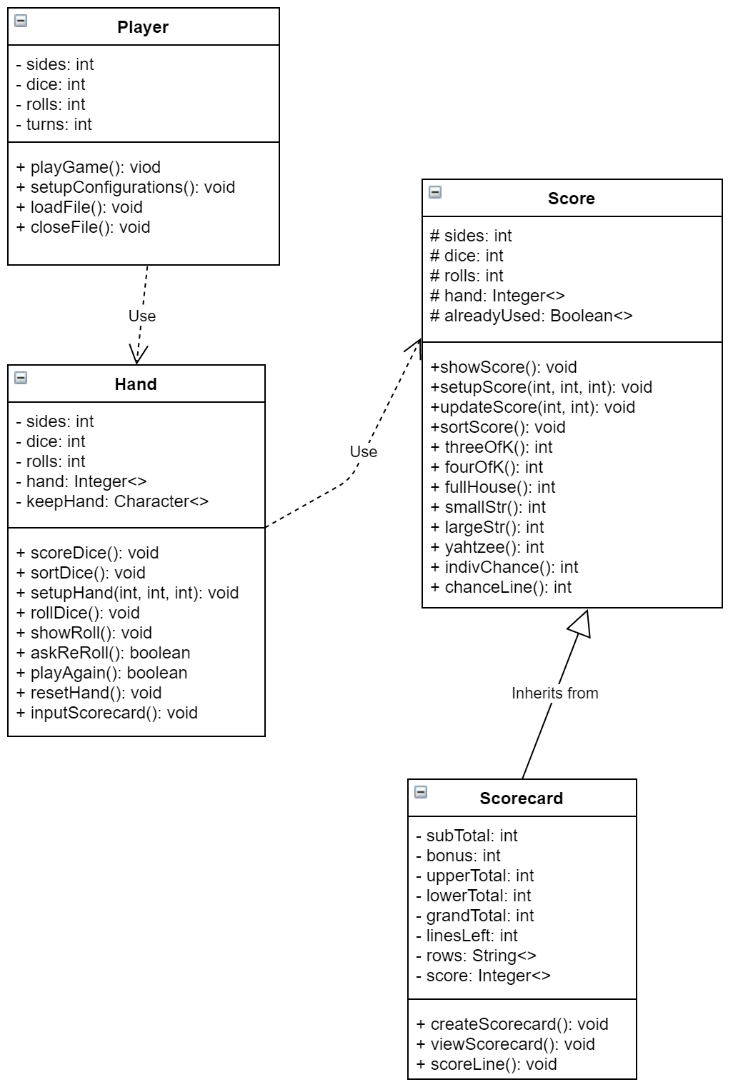
**A summary of the goal or purpose of the program in your own words**

In this project, we created the Yahtzee game. This game works for a single person and shows the score of each hand and keeps the scores of more than one time rolling through i.e. a scorecard. The basic run through of this program is that it starts by stating what the game configuration is, it then asks if you want to change it. Then it runs though the game and scores the game. There are a few changes from last homework including having the scoreboard option after every roll. The scoring also adjusted to this change.

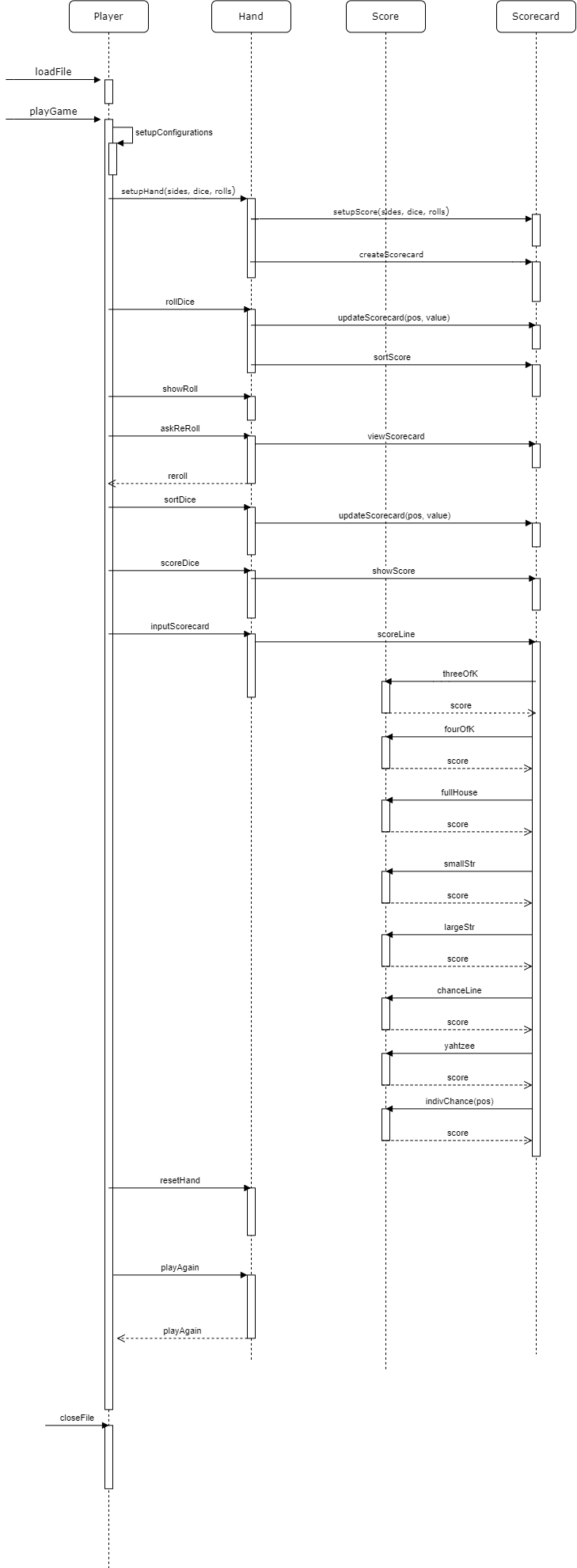
**An overview of the general design you chose for your program**

The design I chose for this was pretty simple. In the main file I call create an object that runs the Player class, which runs the file I/O as well as calls the Hand class to run. Inside the hand class, all of the mechanics of the game happen like rolling the hand of dice and rerolling etc. called inside of that function is the score class which does all the scoring. There is a new class this time which is inherited from the score class which is called Scoreboard which handles all of the scoreboard operations using the Score functions.

**A UML Class Diagram**

****

**UML Sequence Diagrams**

****

**A description of any major design and/or programming issues**

The main problem I had was a very small detail but was catastrophic to my scoring methods. All of my scoring methods worked perfectly but the output would be all of the scores as zero which was incorrect. Initially I thought that the values of dice, rolls, and sides were not being passed through the classes so the arrays would have a size as zero. Eventually I found out that he reason why I was having problems with scoring was because the hand of dice was only sorted once so after more than one turns, the dice would stop sorting which messed up all of the methods in Score. After I fixed that everything worked well.

**A retrospective of what you would have done diﬀerently if you had more time**

If I have more time, I would definitely have made the protocol unbreakable which is what is expected in the next assignment to where the program would correct you if you messed up an input instead of just crashing.