Contents

Readme for Magical Arena

Prateekshya Priyadarshini

23-06-2024

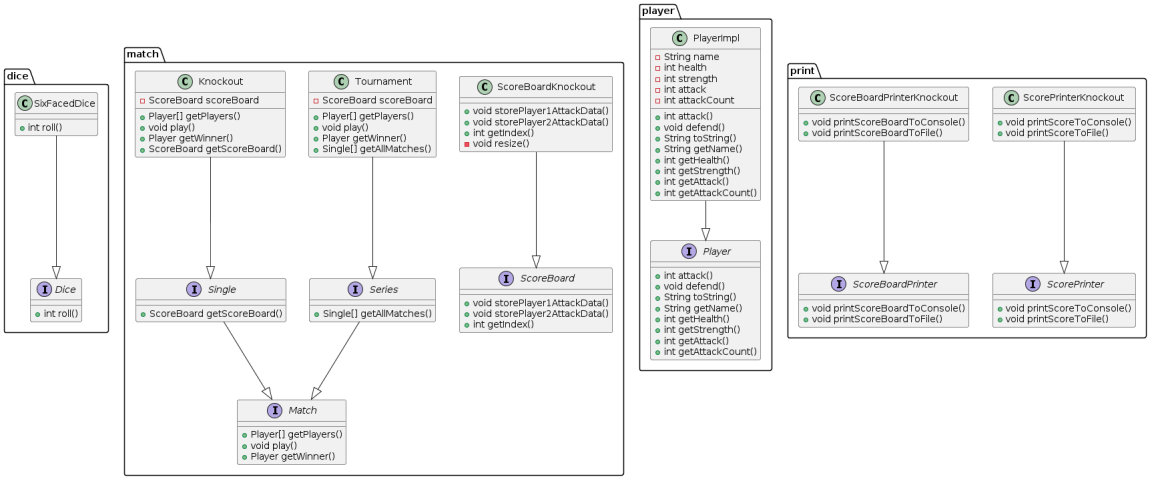
1 Introduction

A Magical Arena is designed where every player is defined by a “health” attribute, “strength” attribute and an “attack” attribute - all positive integers. The player dies if his health attribute touches 0.
Any two players can fight a match in the arena. Players attack in turns. Attacking player rolls the attacking dice and the defending player rolls the defending dice. The “attack” value multiplied by the outcome of the attacking dice roll is the damage created by the attacker. The defender “strength” value, multiplied by the outcome of the defending dice is the damage defended by the defender. The damage created by attacker which is in excess of the damage defended by the defender will reduce the “health” of the defender. Game ends when any player’s health reaches 0.
Player with lower health attacks first at the start of a match.

2 Class structure

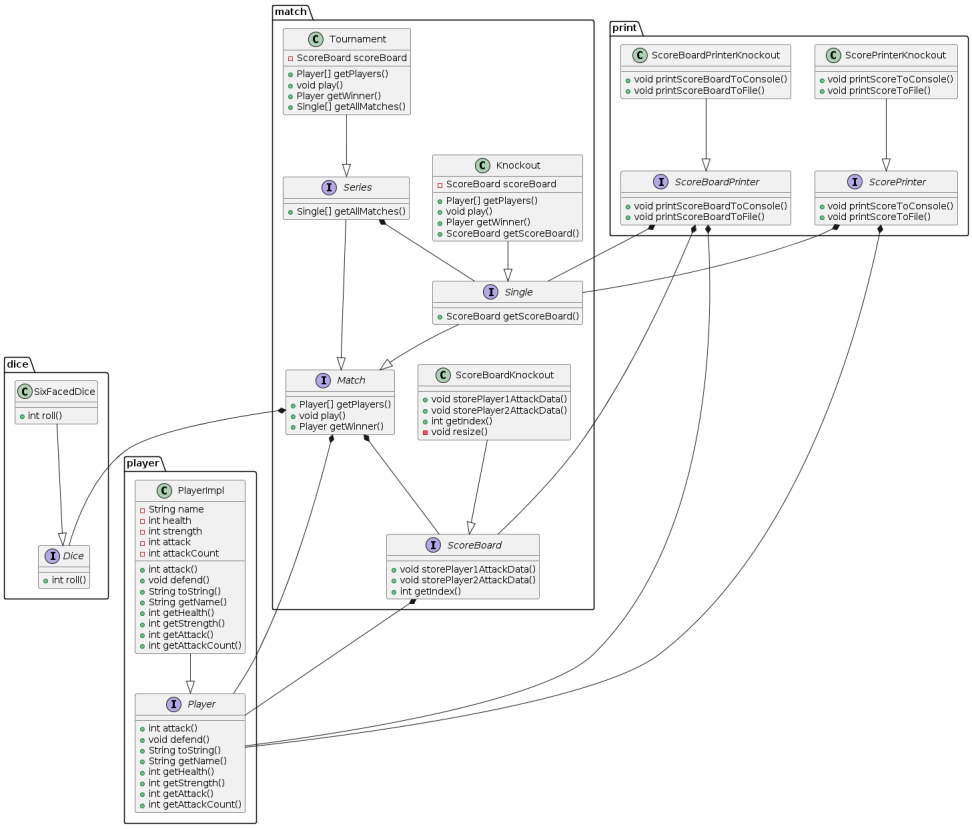
2.1 Class Diagram showing inheritance

The ”Classes.png” in the zip shows the structure.



2.2 Dependency

The ”Dependency.png” in the zip shows the composition relationship. Composite design pattern is used to contain a series of single matches inside the tournament (Tournament class).



3 Steps to run the code

- Install Java in your system.
- Unzip "MagicalArena.zip".
- Open terminal/command prompt inside MagicalArena folder.
- Execute "java Arena" command to run the main file.
- The first input is a text file name which will contain all the player names along with their health, strength and attack values. Sample files are given. (input.txt, inputT1.txt, inputT2.txt)
- The second input is a text file name which will contain the match details. The match can be a single knockout match or a tournament. Sample files are given. (match.txt, matchT1.txt, matchT2.txt)
- All the mentioned matches will be played and the final scores will be displayed for each match.
- You will be prompted to choose whether you want to see the detailed score card of each round of the match.
- The respective score cards will be displayed.
- To run the tests, you can execute the command, "java package_name.test_class_name" in the root folder itself. E.g. java match.TestKnockout

SOME OF THE TESTS WILL THROW EXCEPTIONS, WHICH IS EXPECTED SINCE NO TEST LIBRARIES ARE USED.

4 Description

Arena is the main class which takes all the inputs, reads the files and executes the matches.

4.1 player package

It contains the PlayerImpl class which indicates a player. It has all the required properties, attack and defend methods and toString method.

4.2 dice package

It contains the SixFacedDice class which indicates a six faced dice. It has one method to roll the dice.

4.3 match package

It contains Two types of matches i.e. Knockout and Tournament. Knockout is a single match which requires 2 players and Tournament is a series of matches which requires exactly 8 players. In a tournament, all the players will play in pairs in the first series. 4 players will remain for the semi finale and 2 will remain for finale. Tournament contains a list of Knockout objects for each match.

4.4 print package

It contains two classes to print the final score and the detailed score card respectively.

5 Workflow Examples

```
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
                          Magical Arena
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

Create a text file with the following format:
    1. The first row contains the total number of players.
    2. Each subsequent row contains a playerâ??s name, health
       , strength, and attack, separated by commas.

Example:

2
A,50,5,10
B,100,10,5

Enter the input file name:
input.txt

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
Create a text file with the following format:
     1. First line: Match type (Knockout or Tournament).
     2. Second line: Number of matches.
     3. Next lines: Player names for each match, separated by commas.
For tournaments, total number of players should be exactly 8.
You cannot mix different types of matches in one file.
You can only enter the player names whose details are already available in input.txt.

Example:

Tournament
2
A,B,C,D,G,H,J,K
B,C,D,H,J,K,M,P

Enter the input file name:
match.txt
```

```
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
Create a text file with the following format:
     1. First line: Match type (Knockout or Tournament).
     2. Second line: Number of matches.
     3. Next lines: Player names for each match, separated by commas.
For tournaments, total number of players should be exactly 8.
You cannot mix different types of matches in one file.
You can only enter the player names whose details are already available in input.txt.

Example:

Tournament
2
A,B,C,D,G,H,J,K
B,C,D,H,J,K,M,P

Enter the input file name:
match.txt

+----------------------------------+
|    A                             |
|    -----                         |
|    Total Attacks: 8              |
|    Remaining Health: 0           |
|                                  |
|    B                             |
|    -----                         |
|    Total Attacks: 8              |
|    Remaining Health: 80          |
+----------------------------------+
|    B won!                        |
+----------------------------------+
Do you want to print the complete score table? (Y/N)
y
```

```
|    Total Attacks: 8              |
|    Remaining Health: 0           |
|                                  |
|    B                             |
|    -----                         |
|    Total Attacks: 8              |
|    Remaining Health: 80          |
+----------------------------------+
|    B won!                        |
+----------------------------------+
Do you want to print the complete score table? (Y/N)
y

+-------------------------------------+-------------------------------------+
|            A's attack (5)           |            B's attack (10)          |
+-------------------------------------+-------------------------------------+
| Attacking  Damage  Defending  Opponent | Attacking  Damage  Defending  Opponent |
|      Dice              Dice    Health |      Dice              Dice    Health |
+-------------------------------------+-------------------------------------+
|         1      10          2       100 |                                     |
|                                        |         5      25          3        40 |
|         5      50          5       100 |                                     |
|                                        |         2      10          2        40 |
|         3      30          5       100 |                                     |
|                                        |         5      25          1        20 |
|         4      40          5       100 |                                     |
|                                        |         5      25          5        20 |
|         2      20          2       100 |                                     |
|                                        |         1       5          4        20 |
|         1      10          5       100 |                                     |
|                                        |         3      15          1        10 |
|         2      20          5       100 |                                     |
|                                        |         3      15          2         5 |
|         5      50          3        80 |                                     |
|                                        |         2      10          1         0 |
+-------------------------------------+-------------------------------------+
```