Boat Racing Game

You are required to build a simple game 'Boat Race' in Java program that fulfil the below requirements. Analyze and develop the Java program as per described using the Object-Oriented design. You should design your program for optimum maintainability and reusability with the best practices of object-oriented techniques you have learnt. You also need to document your design using the UML class and class relationship diagrams.

The game rules:

- The game is a two players game. At the beginning of the game, each player will be allocated with a boat. During the game, the players take turn to throw the dice (you can use the random function to generate the random dice number) to decide how many steps should the boat move forward.
- The river can be visualised as 100-columns track as below, which is filled with random number of traps(#) and currents(C).

Start End

		C			#		C	#			C				#		…..		C	#			#		C		C	

- Once the game started, all the traps and currents will be scattered randomly in the river. Some currents are stronger than the others, so as the traps. The stronger current or trap will make the boat moves more steps forward or backward. When boat hits the trap, the boat will need to move backward x number of steps, when the boat hits the current, it will move forward x number of steps. The boat should not be allowed to move beyond the river's boundary.

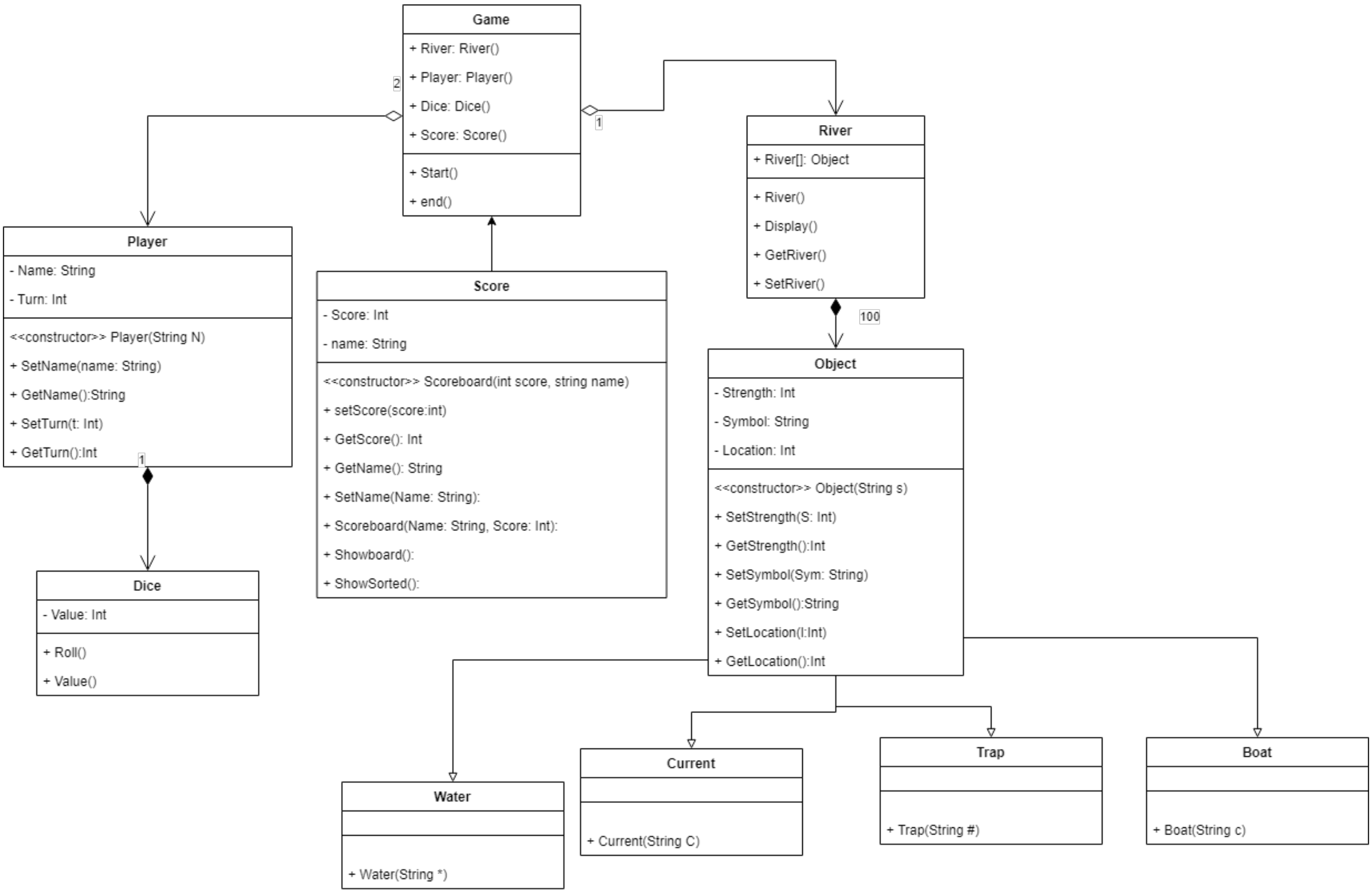
- Game will end when either player's boat reaches the end of the river. Display the location of the boats after every move.

When the game starts, display the Top 5 scores and ask the player for the name (short name with one word). You should count the total turns that each player takes in the games. When the game ended and the score of the player is within the top 5 scores, store the player's score and name in the 'TopScore.txt' text file. The list should be ordered by score in ascending order.

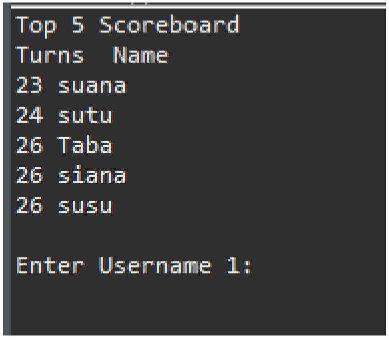
Tips: You can add any additional attributes to the objects in this game which you see fit

Additional Functionality:
Design and develop one additional function that may help to improve the game you have developed above.

Class and class relationship diagram for all the classes in the system

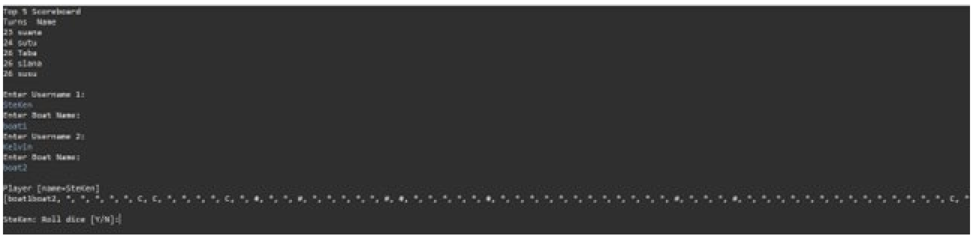


Screenshots with descriptions to demonstrate the test results



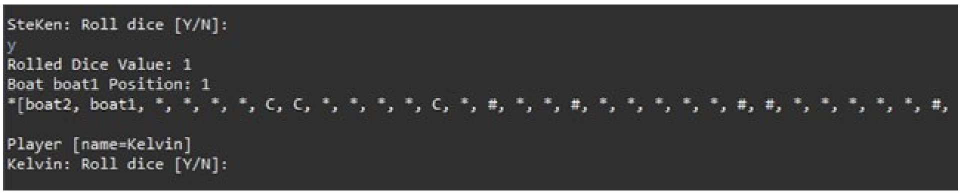
```
Top 5 Scoreboard
Turns  Name
23 suana
24 sutu
26 Taba
26 siana
26 susu

Enter Username 1:
```

When the program run, the program will read the text file which stores all the score of past players and displays out. Then, the program will prompt the user to enter their name and also their boat name.



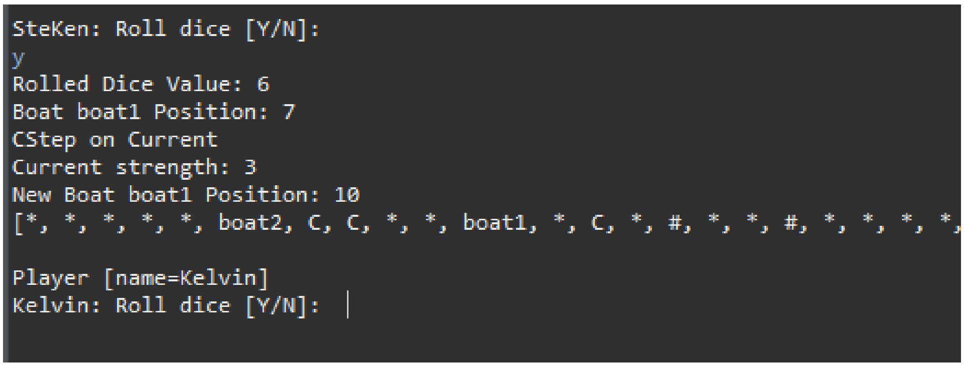
```
Top 5 Scoreboard
Turns  Name
23 suana
24 sutu
26 Taba
26 siana
26 susu

Enter Username 1:
SteKen
Enter Boat Name:
boat1
Enter Username 2:
Kelvin
Enter Boat Name:
boat2

Player [name=SteKen]
[boat1boat2, *, *, *, *, *, C, C, *, *, *, *, C, *, #, *, *, #, *, *, *, *, *, #, #, *, *, *, *, *, #, *, *, *, *, *, *, *, *, *, *, *, *, *, #, *, *, *, #, *, *, *, *, *, *, *, *, *, *, *, *, *, *, C, *

SteKen: Roll dice [Y/N]:
```

After the user entered their username and boat name, a river with a range of 100 will be displayed and prompted the user to start rolling the dice.



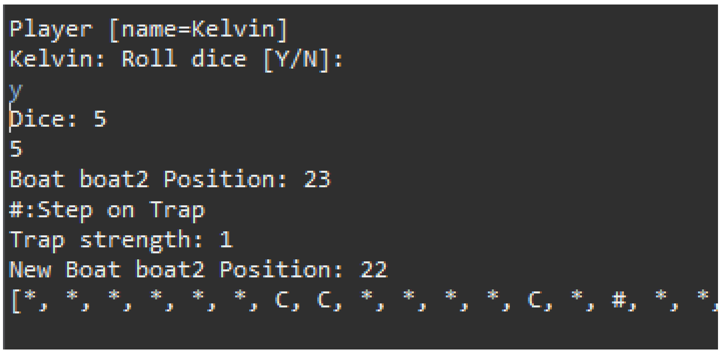
```
SteKen: Roll dice [Y/N]:
y
Rolled Dice Value: 1
Boat boat1 Position: 1
*[boat2, boat1, *, *, *, *, C, C, *, *, *, *, C, *, #, *, *, #, *, *, *, *, *, #, #, *, *, *, *, *, #,

Player [name=Kelvin]
Kelvin: Roll dice [Y/N]:
```

The user can enter either ‘Y’ or ‘N’ (Non-case sensitive). When ‘Y’ is returned, the program will roll the dice with a random integer for the user. The new location of the boat (initial position + dice integer) will be allocated in the river displayed. When ‘N’ is returned, the program will return a statement of thank you. For any alphabet other than ‘Y’ or ‘N’, the program will print an error message to remind the user to only enter ‘Y’ or ‘N’.



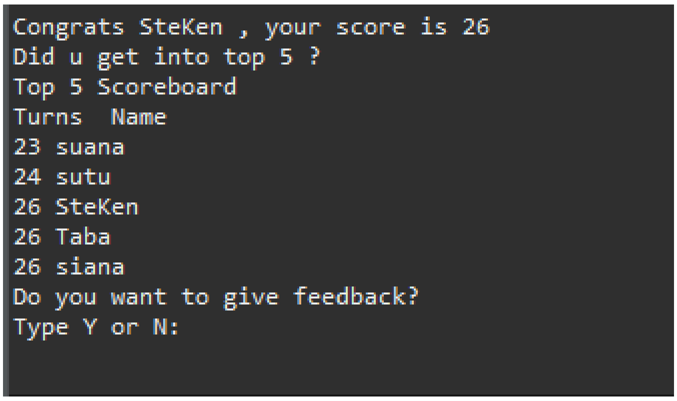
```
SteKen: Roll dice [Y/N]:
y
Rolled Dice Value: 6
Boat boat1 Position: 7
CStep on Current
Current strength: 3
New Boat boat1 Position: 10
[*, *, *, *, *, boat2, C, C, *, *, boat1, *, C, *, #, *, *, #, *, *, *, *,

Player [name=Kelvin]
Kelvin: Roll dice [Y/N]:
```

When the new boat location of user steps on a current, the program will print out the initial boat location, the strength of the current that the boat encountered, and the new boat location (initial boat location + strength of current).



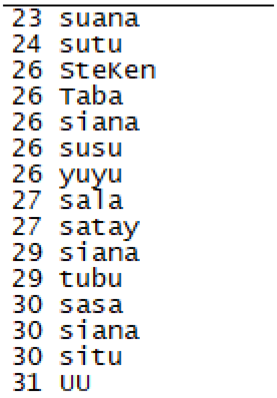
```
Player [name=Kelvin]
Kelvin: Roll dice [Y/N]:
y
Dice: 5
5
Boat boat2 Position: 23
#:Step on Trap
Trap strength: 1
New Boat boat2 Position: 22
[*, *, *, *, *, *, C, C, *, *, *, *, C, *, #, *, *,
```

When the new boat location of user steps on a trap, the program will print out the initial boat location, the strength of the trap that the boat encountered, and the new boat location (initial boat location – strength of trap).



```
Congrats SteKen , your score is 26
Did u get into top 5 ?
Top 5 Scoreboard
Turns  Name
23 suana
24 sutu
26 SteKen
26 Taba
26 siana
Do you want to give feedback?
Type Y or N:
```

Picture1

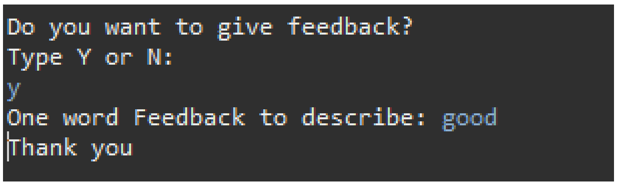


```
23 suana
24 sutu
26 SteKen
26 Taba
26 siana
26 susu
26 yuyu
27 sala
27 satay
29 siana
29 tubu
30 sasa
30 siana
30 situ
31 UU
```

Picture2

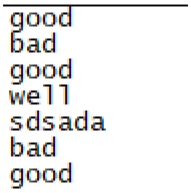
(Picture 1) Whoever reaches the index 100 in the river first, the program will print out a message where containing the name and total turns of player to congrats the player. The total turns will be acted as the score of the player.

(Picture 2) The program will display the top 5 players who have the highest scores from the text file. In the situation of the picture, the program only displayed the following players: Suana, Sutu, Steken, Taba, Siana.



```
Do you want to give feedback?
Type Y or N:
y
One word Feedback to describe: good
Thank you
```

Picture3



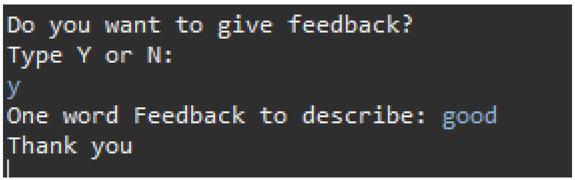
```
good
bad
good
well
sdsada
bad
good
```

Picture4

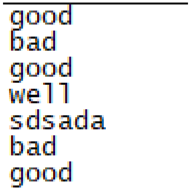
(Picture 3) The user can give feedback after the game. The user can enter either ‘Y’ or ‘N’ to decide whether to give feedback or not. When a ‘Y’ is returned, the program will print a new statement ask the user to give feedback in the size of one word. When ‘N’ is returned, the program will print a “Thank You” statement. Any other alphabet will return an error message.

(Picture 4) All the feedbacks from the user will be stored in a new text file named as “userfeedback.txt”

Explanation or justification of the additional functionality



```
Do you want to give feedback?
Type Y or N:
y
One word Feedback to describe: good
Thank you
```



```
good
bad
good
well
sdsada
bad
good
```

Our additional function is asking for feedback for this boat game and the feedback will be stored in the text file as shown above. The feedback allowed the programmer to improve the design and provide better user experience. Moreover, the feedback will be stored in anonymous for protecting the privacy of the user.