DASS Assignment 3 New Features

Team - Group AE (31 Row)

1. Kishan Sairam Adapa - 2018101026
2. Akshat Goyal - 20181010
3. Siddireddy Akash Reddy* (Didn't respond and hasn't contributed anything)

UML Class Diagram

Description

First Part

1. maxPatronsPerParty variable changed to 6 in drive file

Second Part

Sequence Diagram

1. **Scores** button is added in the button panel of control desk
2. **ScoreView.java** file is made which is responsible for opening the window on clicking the score button.
3. Score Window has 3 parts.
   1. First part is player panel which shows the list of all scores, max, min score(depends on the button clicked in button panel) of the selected party.
   2. Second part is the leaderBoard which shows the list of all players along with their highest score.
   3. Third part is the button panel which contains 3 buttons.
      1. All score button shows all the scores of the player.
      2. Max score button shows the max score of the player.
      3. Min score button shows the min score of the player.

4. sortByScore class is made in the scoreHistoryFile which sorts the list of type Score according to score in descending order.

5. LeaderBoard and player panel updates automatically while players keep playing .

6. Various other functions implemented to get the aforementioned things.

Third Part

Sequence Diagram

1. **Pause** button is added in front of every lane to pause and resume the game.
2. **Load Game** button is added in the button panel of control desk
3. **LoadGameView.java** file is made which is responsible for opening the window on clicking the load game button.
4. Load Game Window has three parts.
   1. First part is bowler panel which shows the list of all bowlers of the selected party.
   2. Second part is the party panel which shows the list of all paused game's party name and time of pause
   3. Third part is the button panel which contains two buttons.
      1. On clicking **Bowlers** button, all the bowlers of the selected party is shown in the bowler panel.
      2. On clicking **Resume** button, selected party is assigned a lane, window closes and game starts.
5. **LaneJsonFile.java** file is made which stores the state of game on pausing the game in Lane.json file and loads the variables from Lane.json file on resuming the game
6. Various other functions implemented to get the aforementioned things.