

**Team Members:** Tianyuan(Rainer)Yuan, Theodore Montalbano, Jerry Yu, Shashwat Srivastava, Griff Zhang

## Class Summary

**Team class:** stores all the information such as name and score, seating and round.

**Match class:** stores the information of matching team and winning score

**Tournament class:** driver class. Contain array heap

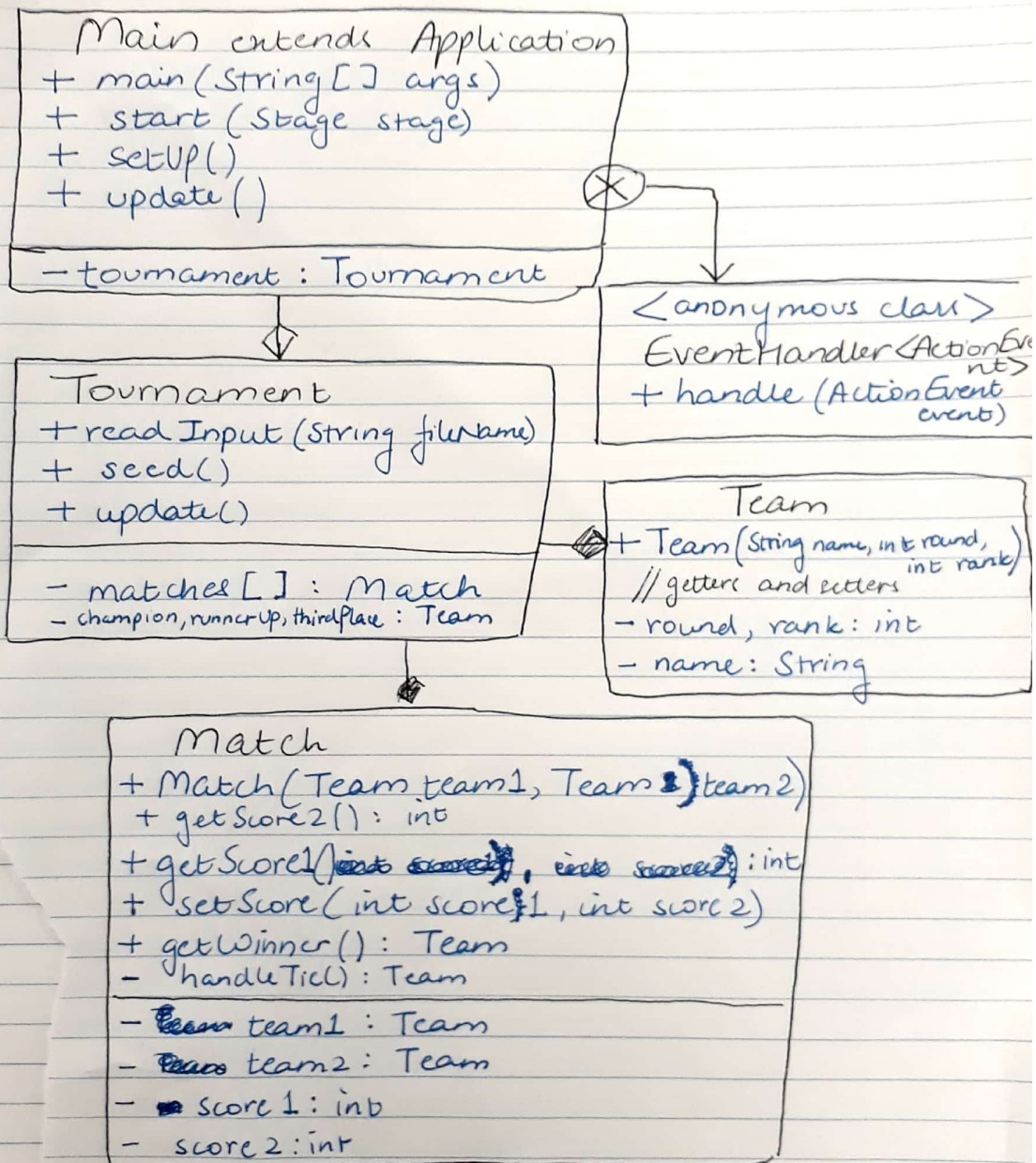
**Main:** launch GUI

Class	Types	Description
Team	String name int rank int round	The Team class represents each team in the tournament. It contains the name and rank (for seeding) of the team.
Match	Team teamOne Team teamTwo int scores	The Match class represents each match played between two teams. It stores the information of two matching teams and winning score.
Main	Tournament tournamentName	Acts as the driving class for the GUI. Therefore, it handles all the GUI elements of the Tournament. This also contains a Tournament object and inputs the file path..
Tournament	String name Match[] matches Team champion Team second Team third	The Tournament class represents the tournament that is being played. The tournament class contains the name of the tournament, Team class instances (which represent each team), an array of matches to be played (which acts like our main data structure) and specific instances of the

		Team class which represent the champion, runner-up and the third place holder. This class is responsible for taking in the input from the file, seeding and storing it in the main data structure.
--	--	--

## Class diagram:

### Class Diagram



## Main class

Return type	name	parameters	description
void	main	String[] args	The driver method for the class. The args[] contains command line arguments
void	setUp	-	The method sets up the GUI and the back end using the Tournament class instance
void	start	Stage stage	The method starts the GUI and displays the Tournament bracket
void	update	-	Updates the state of the tournament bracket in the backend part and also displays the updated state of the bracket

## Tournament

Return type	name	parameters	description
void	readInput	String fileName	Responsible for reading input from the file
void	seed	-	Responsible for seeding the teams according to their team ranks and store them into the data structure (Array)
void	update	-	Updates the data structure depending upon the result of a match/challenge

## Match

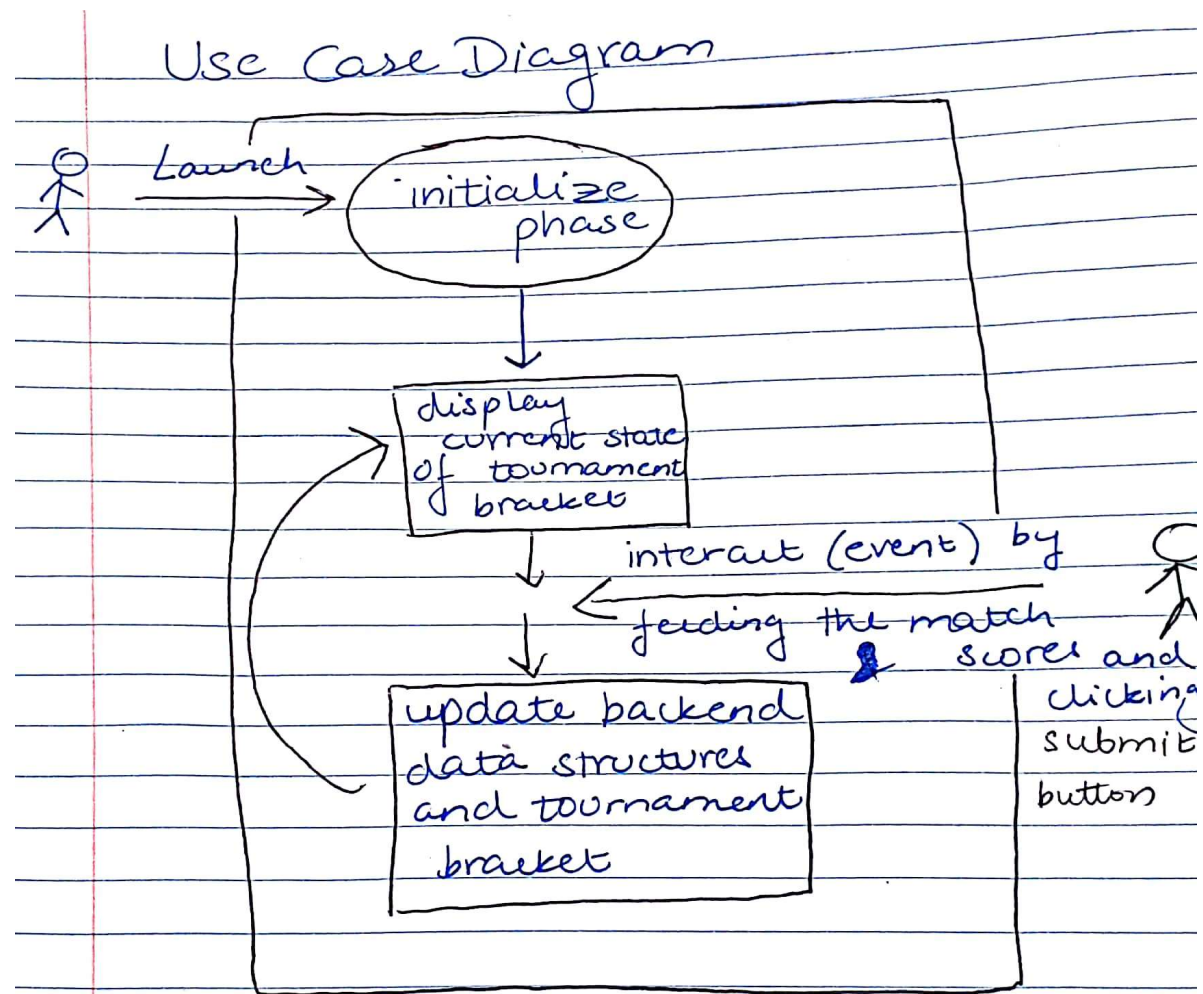
Return type	name	parameters	description
Constructor	Match	Team team1, Team team2	Constructor to initialize the instance
int	getScore1	-	Getter for score1

int	getScore2	-	Getter for score2
void	setScore	int score1, int score2	Setter for score1 and score2
Team	getWinner	-	Gets winner by comparing the score or handling the tie (calls a private method to do so)

## Team

Return type	name	parameters	description
Constructor	Team	String name, int round, int rank	Puts in initial values
int	getRank	-	Getter for Rank
int	round	-	Getter for Round
-	round	int round	Setter for Round
String	name	-	Getter for name of the Team

## Use-Case Diagram:



initialize phase → read data from the file and set the data structures up.

[see object diagram for initialize state of objects].

# Object Diagram

```
graph TD
    Main((Main)) -- tournament --> Tournament((Tournament))
    Main -- teamLabel1 --> Match1((Match))
    Main -- teamLabel2 --> Match2((Match))
    Main -- submitBtn1 --> Match3((Match))
    Main -- submitBtn2 --> Match4((Match))
    Main -- ... --> Match5((Match))
    Tournament -- matches --> Match1
    Tournament -- matches --> Match2
    Tournament -- matches --> Match3
    Tournament -- matches --> Match4
    Tournament -- matches --> Match5
    Match1 -- teamOne --> Team1((Team))
    Match1 -- teamTwo --> Team2((Team))
    Match2 -- teamOne --> Team3((Team))
    Match2 -- teamTwo --> Team4((Team))
    Match3 -- teamOne --> Team5((Team))
    Match3 -- teamTwo --> Team6((Team))
    Match4 -- teamOne --> Team7((Team))
    Match4 -- teamTwo --> Team8((Team))
    Match5 -- teamOne --> Team9((Team))
    Match5 -- teamTwo --> Team10((Team))
```

The diagram illustrates the following objects and their attributes:

- Main**: tournament, teamLabel1, teamLabel2, ..., submitBtn1, submitBtn2, ...
- Tournament**: name, matches (array), champion, second, third
- Match** (multiple instances): teamOne, teamTwo, scores
- Team** (multiple instances): name, rank, round

Relationships (Arrows):

- Main** to **Tournament**: tournament
- Main** to **Match** (instances): teamLabel1, teamLabel2, submitBtn1, submitBtn2, ...
- Tournament** to **Match** (instances): matches (multiple arrows to different Match instances)
- Match** (instances) to **Team** (instances): teamOne, teamTwo (multiple arrows to different Team instances)

