# The Tennis Game

We want you to develop an application to help digitalize the Tennis sport. The association of chair umpires has asked you to develop a simple application to manage the scoring of a Tennis game.

Tennis is a ball and racket sport that is scored in an interesting way. The scoring system of a tennis *match* is based on *points,* games and *sets.* We will focus on the scoring of *points* inside a single *game* for the scope of this exercise (excluding the *set* & *match* management).

The rules that we want you to consider are the ones found in the section "Game Score" of the wikipedia page of Tennis Scoring System

## Exercise:

### Feature 1 - Scoring engine

We want you to develop an application used by the Tennis Umpire that can be used to score a game in real time, so we can use it for all of the tennis related endeavors we plan to undertake in the future. To begin with, we're going to need a way to update the score when a player wins a point, see what the current score is after each service, and see if there is a winner based on the current score and the rules above.

**Winning a Point Increases Score Correctly**

**As a** tennis chair umpire
**I want** the score to increase according to Tennis rules when a player wins a point during the first 3 points of each player
**So that** I can easily read and announce the current score to the players and the crowd

Given the score is 0:0
When the server wins a point
Then the score is 15:0

Given the score is 15:15
When the receiver wins a point
Then the score is 15:30

Given the score is 30:30
When the server wins a point
Then the score is 40:30

Deuce and Advantage are Scored Correctly

**As a** tennis chair umpire
**I want** deuce and advantage to be scored correctly
**So that** I can easily read and announce the current score to the players and the crowd

Given the score is 40:40
When the receiver wins a point
Then the score should be 40:A

Given the score is A:40
When the receiver wins a point
Then the score should be 40:40

**Winning Points are Scored Correctlly**

**As a** tennis chair umpire
I **want** the winning point to be scored correctly
So that I can easily read and announce the current score to the players and the crowd

Given the score is 40:30
When the server wins a point
Then the server should win

Given the score is 40:A
When the receiver wins a point
Then the receiver should win

## Feature 2 - Umpire User Interface

We want you to create a simple User Interface to fulfill 2 main purposes:

1- Display the current score to the players and the crowd
2- Let the chair umpire update the score when a player scores

Note: If you are applying for a **Frontend** job position , we will expect a web interface implemented **with** Java FX or SWING . If you are applying for **backend** position, you can either choose a web interface or a command line interactive interface.

**Display the score**

**As a** tennis chair umpire
I **want** to see a scoreboard with 2 labels for each player (Server, Receiver) and below each of them, their current score
So that both players and the crowd can see the current score at any moment

When the game hasn't started
Then the display should show 0:0

When the umpire presses the button "point server"
Then the display will update the score according to the rules in Feature 1

When the umpire presses the button "point receiver"
Then the display will update the score according to the rules in Feature 1

When the server wins the game
Then the display will show "Game Server" and the score will be set to 0:0 again

When the server wins the game
Then the display will show "Game Receiver" and the score will be set to 0:0 again

## Exercise evaluation

If you are applying for a backend job, We expect the application to be written in Java. For the UI framework you can use the framework of your choice (Example : Swing or JavaFX).

What are we going to consider when evaluating your exercise:

- The acceptance criteria for each user story are met
    - The modelization & design of your library and its components
    - Good software development principles and practices  clean code, SOLID principles
    - The existence of an automated test suite that ensures the code does what it should
    - Easiness to understand, evolve & maintain
    - The tools & libraries that you choose to use

Other tips

- Proper dependency management will be appreciated
- Dockerization will not add points, we don't want you to dedicate time on that
- Use of database persistence is out of the scope also and will not add points
- Deailng  with concurrency is out of scope of this exercise
- Functional Programming principles are taken into account positively