

# Programming Assessment

The purpose of this exam is to provide a better understanding of your design, programming, and analytical skills. As with any programming project, you will need to balance the amount of time spent on development against the quality of your result. We ask you not to expend more than **a week**.

Please email a Github, link with your solution and a readme file explaining if it is required to follow any additional step to run it. It would be great to run your solution as quickly as possible with very few additional steps.

You have to use .NET Core for the back-end and SQL Server for the database structure. Usage of Entity Framework or Dapper are both allowed.

For the front-end you can use ReactJS or Angular. If you have previous experience with React, we encourage you to go down this path.

You can add any external component you need.

# Evaluation

We are aware that the time for developing the solution is short so please consider that for this instance your work will be evaluated in this order:

* Back-end
  + Architecture quality of the solution
  + Usage of design and architecture patterns for each scenario
  + Good responsibility decisions
  + Code reusability
  + Scalability
  + Optional
    - Dependency Injection
    - Azure DB and WebAppService
    - UoW or similar patterns
* Front-end
  + Architecture
  + Responsive design
  + Good usage of ES6
  + Optional
    - Preprocessed styling
    - Production builds
* Unit testing
* Easy startup

# Game of drones game specification

In *Game of Drones* there are two players trying to conquer each other.   
  
Players take turns to make their move, choosing Paper, Rock or Scissors. Each move beats another, just like the game “Paper, rock, scissors”.

Like so:

* Paper beats Rock
* Rock beats scissors
* Scissors beat Paper

The first player to beat the other player 3 times wins the battle.

The website must have the following behavior:

1. Inputs for each player to enter his name. (Only two players) and a start button to begin the game.
2. Once the game begins, each player choose one of the possible moves.
   1. First, player1 pick his move, then player2.
   2. The system computes the result of the play.
      1. (The game happens on the same computer for both players. It is not required to create a true online game. Both players share the computer, and the system asks each player for their move assuming the other player looks away while the other selects the move)
   3. The result of each round should be displayed somewhere in the screen, so that players can know the game score while they are playing.
3. Step #2 repeats until one of the players wins three times. This player will be the winner of the game.
4. Once the game has finished, a Play Again button shows to start a new game.

**Game statistics**

The result of each game should be stored to keep track of games won by each player. We would like to know how many games a player has won and show them in a page.

Implementing this considering that there could be a large number of results will be a plus.

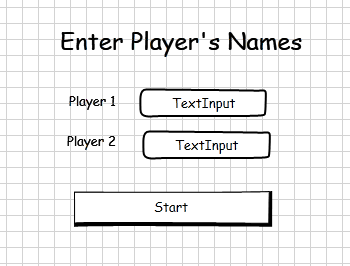
**Logging**

We would like to keep track of any action as well as any failure throughout the application.

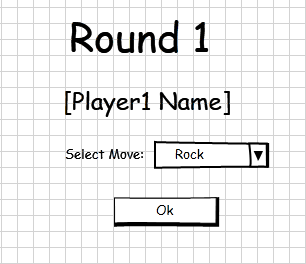
# Mocks

Here is a mock of each screen. You are free to modify the look and feel of the screen as you please.

To start a game, the system asks for the name of each player.



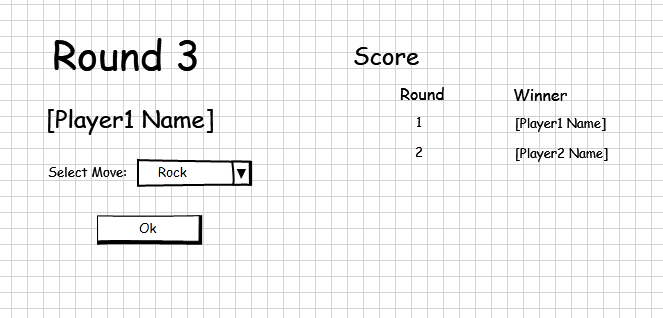
The game starts with Round1. The system asks Player1 for a move. Replace [Player1 Name] with player’s name.



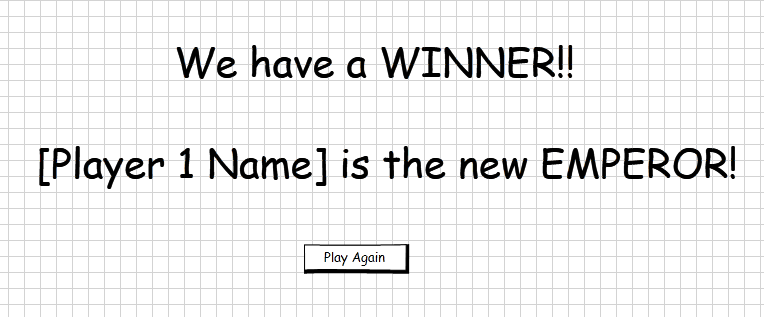
After Player1 selects a move, then the system asks Player2 for a move (same round).



Winners of each round are displayed somewhere on the screen. Here we see Round #3, and to the right the score of the previous rounds:



When a player reaches three wins, he is the winner, and the following screen displays:



Clicking the Play Again button, the systems takes you to the first screen.