Four in a row online

Project of Foundation of Cybersecurity

Barsanti Nicola, Tumminelli Gianluca

2019-2020

Contents

1 Requirement Analysis

This paper will document the development of the game *Four in a row online*. The application is a multiplayer online game accessed by a graphic interface where each user must be authenticated before access the application and his communication must be confidential and secure. When logged the user could choose another player, start a match and then they could talk each other and play the game. The application will have a score table where there will be printed general statistics about all the users of the application.

1.1 Specification Document

- The user will access the application through a GUI
- The user will access the application remotely
- The user must authenticate to access the application
- The user messages must be secure,confidential and authenticated
- **The user could log-out from the application**
  - The user will use a username and password to login
  - The user will use an authtoken to login
- **The user could see all the available players and interact with them**
  - The user will have a list of all the free users available
  - The user will select an adversary and send him a challenge
  - The user will see all the pending challenges
  - The user could withdraw a previous sended challenge
  - The user could accept or reject a challenge
- **The user could play a match with other players**
  - The user will have a dedicated window for play a match
  - A match is played by two users
  - A match is composed by rounds
  - In each round the control is assigned to the opposite player
  - In the first round the control is assigned to the player who has sent the challenge
  - In each round the user in charge selects an available column of the gameboard
  - Each round lasts a maximum of 15s

  - The first user who put four tokens in a row win
  - If all the gameboard is full without a winner the match ends with a tie
  - The user could logout from a match
  - The user who left a game automatically loses
- **During a match users could talk**
  - The user during a match became unavailable for challenges and reject automatically all the pending requests
  - Into the game window there will be a chat
  - The user during a match could always read messages from the chat
  - The user during a match could always write a message into the chat
- **The application has a rank table**
  - The user could see the ranks of all the users
  - A rank will be defined as $\langle TotalMatch, WonMatch, LostMatch, TieMatch \rangle$

1.2 System Requirements

- The application is composed by a server and several clients which communicate remotely
  - A client-server protocol is adopt for the communication between clients and server
  - A peer-to-peer protocol is adopt for the communication between clients
  - All the messages must be sanified before used by the application
  - There will be a symmetric session key for each sended message
  - Each user will have a key for a public key encryption protocol
  - The server will have a key for a public key encryption protocol
  - The server will store all the PKE public keys of the users
  - PKE key will be used by users to cipher the peer-to-peer session keys and authenticate them
  - PKE key will be used by the server to cipher the client-server session keys and authenticate them
- the GUI is composed by four windows:
  - a Login window
  - a Main window
  - a Game window

  - a Rank window

- The first window showed is the login window
  - The user could login with a username and password
  - The user could register giving a username and password
  - The user could login with an authtoken

- After a successfull login will be shower the main window
  - Into the main window there will be a list of the current player
  - The user could choose a user and send him a challenge
  - The user could make only a challenge a time
  - The user could withdraw the current sended challenge
  - The user could see a list of all the pending received challenges
  - The user could reject a pending challenge
  - The user could accept a pending challenge
  - The user could reject all the pending challenge directly
  - The user could logout from the application
  - The user could change the current window to the rank window

- The rank window is accesible only from the main page
  - The user could see statistics of all the registered users
  - The statistics are defined as $\langle TotalMatch, WonMatch, LostMatch, TieMatch \rangle$
  - The user could change the current window to the main window

- When a challenge is accepted the current window is changed to the game window whatever page the user was seeing
  - The game window has a chat
    - The user during a match could always write into the chat
    - The chat during a match is updated instantly upon receipt of a message
  - The game window has a matrix of 6x7 as gameboard
  - The match is composed by round
  - During a round only one user could insert one token into the gameboard
  - The player who could insert the token is changed after each round
  - The first player who insert four tokens in a row wins
  - If the matrix is full with no winner the match ends with a tie
  - Each round lasts a maximum of 15s
  - The user could logout from a match and return to the main window
  - If a user logout he loses automatically

1.3 System Specification Document

- The application will be implemented in C++ with Secure Coding
- The application will use OpenSSL library for crypto algorithms
- The application will use .NET library for the GUI