M1 : Advanced UML

EFREI – M1 SIA

Written Exam

(2 hours, all documents allowed)

(Barème indicatif sur 20)

1. Questions de cours [5 Pts]

Reply with precision and concision, and **justify** your answers.

Q1.1 : Is UML a software engineering methodology ? **(1.5 points)**

Q1.2 : We wish to implement a package offering support for basic arithmetic expressions with integer variables. Which design pattern(s) would you suggest to use? Explain your answer with a small class diagram. **(1.5 points)**

Q1.3 : We have developed a system that interacts with a relational DB. Each time we need to query the value of an object, we create a connection to the database, send the query, get the response then disconnect. However, because of the frequent queries needed and that establishing a connection is quite long, this implementation is very slow. Suggest a design pattern and how to apply it to solve this issue. **(2 points)**

2. Problem: Analysis of MyChess [15 Pts]

A company offering a Web site dedicated to the game of Chess (articles, tutorials, …) wishes to put online a chess application. The goal is to allow registered users to quickly find an adversary and play a game. The registration of users and their authentication is already implemented. Each game outcome is recorded, and used to compute and update the ELO ranking –which is a standard indication of player strength- of the users, according to rules which will not be detailed here. The system is to be developed as a rich client application that can be executed directly from a web browser with no installation.

Chess is a game for two players, designated as White and Black. The game is played on a square chequered chessboard with 64 squares arranged in an eight-by-eight grid. At the start, each player (one controlling the white pieces, the other controlling the black pieces) controls sixteen pieces: one king, one queen, two rooks, two knights, two bishops, and eight pawns. The object of the game is to checkmate the opponent's king, whereby the king is under immediate attack (in "check") and there is no way to remove it from attack on the next move.

Each piece has specific rules governing its possible movements which will not be detailed here. The game is finished when one player checkmates his opponent (victory), when a draw situation is reached (draw), or when one of the players abandons (opponent victory). A draw can also be proposed to the opponent, who can accept it or continue playing.

The game of chess, particularly online, is often limited in length by a game clock. The clock limits the amount of time available to play moves. A **game clock** consists of two adjacent decrementing clocks and allows to stop one clock while starting the other, such that the two component clocks never run simultaneously. The purpose is to keep track of the total time each player takes for his own moves. When a player's clock reaches 0, he loses the game. The

game duration can be configured when setting up the game, and is usually less than 10 minutes per player. Of course, the game can also finish normally before the clock runs out.

The players can challenge other players to a game. Sending a challenge consists in configuring the game parameters (clock duration) and then selecting from the list of connected players one or more other players to send the challenge to. For ease of use, a player can also challenge all connected players that are not currently playing a game. Challenges received by a player will be presented in a specific continuously updated zone of the GUI, as long as he is not currently playing a game. The player will have the option of accepting or refusing a challenge. The first player that accepts the challenge will immediately start a game against the challenger.

Players can also access their profile information, which presents various statistics on past games. In particular, the player will be able to graphically display the evolution of his ELO score, and his game record history. The moves of each game will by default be stored only for 4 hours, but the user can decide to archive a game (save a replay). A GUI to display game replays has already been implemented for other sections of the website, and will be reused. It uses a standard textual exchange format to describe the moves of a game: each move is described by a start and arrival cell.

Question 2.1 : Produce a use case diagram for this system. Annotate the diagram with comments, or explain by a short text what each identified use case denotes. **(4 points)**

Question 2.2 : Build an analysis class diagram (only inheritance, simple associations with cardianlity, and no operations) for this system. Be as precise as possible. **(5 points)**

Question 2.3 : When going to design, we define three main components in the system. The **player GUI**, which runs in the user's browser, the **server component** that the users connect to, and the **history component** that stores information on past games in an efficient relational DB. **(4 points)**

Build a sequence diagram describing the interactions necessary to challenge a single player, up to the point where the challenged player accepts the challenge. In these interaction level sequence diagrams, we will only represent the lifelines of the system components (no actors).

Question 2.4 : Deduce from the previous question the offered and required component interfaces (for this interaction), and represent them on a diagram (specify operation signatures as precisely as possible). **(3 points)**