

Документация на проекта "Sudoku Game"

Проектът "Sudoku Game" е приложение за игра на Судоку, създадено с използването на JavaFX за графичен потребителски интерфейс и RMI (Remote Method Invocation) за комуникация между клиентската и сървърната части.

Структура на проекта

Проектът е структуриран в различни пакети:

client: Съдържа класове, свързани с клиентската част на приложението, като главния клас Client, отговаря за стартирането на играта.

server: Съдържа сървърната част на приложението, включително основния клас GameServer, който предоставя логиката за играта и методи за обработка на статистика. Главния клас Server отговаря за стартирането на сървъра,

types: Съдържа класове, представляващи различни типове обекти, използвани в проекта, като SudokuPuzzle, Stat и други.

resources: Съдържа ресурсни файлове, като FXML файлове за графичния интерфейс на приложението.

Възможности за подобрене:

Board.fxml може да се направи динамично оразмеряване спрямо големината на прозореца.

Судокутата може да се генерират по по-интелигентен начин използвайки по-добри алгоритми.

Добавяне на нагледен начин за проверка на статистика от клиента.

Клас SudokuPuzzle

Класът SudokuPuzzle се използва за съхранение на Sudoku пъзел и неговото решение. Използва се като обект изпращан от сървъра към клиента при генерация на нов пъзел. Класът е Serializable.

puzzle: Двумерен масив от цели числа, представляващ оригиналния Sudoku пъзел.

solution: Двумерен масив от цели числа, представляващ решението на Sudoku пъзела.

SudokuPuzzle(): Създава нов празен обект от тип SudokuPuzzle.

SudokuPuzzle(int[][] puzzle, int[][] solution): Създава нов обект от тип SudokuPuzzle с предварително зададен пъзел и решение.

printPuzzle(): Извежда на конзолата оригиналния пъзел. Използвано с тестови цели.

printSolution(): Извежда на конзолата решението на пъзела. Използвано с тестови цели.

Включва и гетъри и сетъри за всички член данни.

Клас SudokuMove

Класът SudokuMove се използва за съхранение на хода, направен от играча в играта Sudoku.

x: Цяло число, което представлява реда, на който е направен хода.

y: Цяло число, което представлява колоната, на която е направен хода.

value: Цяло число, което представлява стойността, която е поставена в клетката от играча.

Конструктор:

SudokuMove(int x, int y, int value): Създава нов обект от тип SudokuMove със зададени координати на ред и колона, както и стойност.

Включва и гетъри и сетъри за всички член данни.

Клас Stat

Клас Stat има за цел да съдържа информацията изпращана от клиента към сървъра при успешно завършен пъзел. Класът е Serializable.

name: Стринг, представляващ името на играча.

score: Цяло число, което отразява резултата на играча.

time: Цяло число, което указва времето, в което играчът е завършил играта.

toString(): Преобразува статистиката на играча във форматиран стринг.

Включва и гетъри и сетъри за всички член данни.

Клас Sudoku

Целта на клас Sudoku е да поддържа backend-а на играта. В този клас се съдържа логиката (правилата) на sudoku и е връзката между RMI сървър и frontend-а (BoardController)

Методи:

Sudoku(): Създава обект от тип Sudoku и инициализира игралната дъска с начална конфигурация.

startup(String difficulty): Инициализира играта "Судоку", като получава загадка от сървър в зависимост от избраната трудност. Нулира състоянието на играта и настройва игралната дъска с получената загадка.

sendStats(int time, String name): Изпраща статистиката на играча (име, време, резултат) към сървър. Извиква се, когато играта приключи.
Управление на дъската

solve(): Решава загадката "Судоку" и актуализира игралната дъска с решението.

addMove(SudokuMove move): Добавя ход към игралната дъска, ако е валиден. Ако ходът е невалиден, увеличава броя на грешните ходове.

undoMove(): Отменя последния ход на играча и го добавя към списъка с отменени ходове.

redoMove(): Повтаря последния отменен ход и го добавя отново към списъка с ходове.

removeMove(SudokuMove move): Премахва ход от игралната дъска и от списъка с ходове.

reset(): Нулира играта чрез изчистване на всички ходове и нулиране на броя на грешните ходове.

isSolved(): Проверява дали загадката "Судоку" е решена.

moveCount(): Връща броя на ходовете, направени от играча.

undoneMoveCount(): Връща броя на отменените ходове.

getBoard(): Връща текущото състояние на игралната дъска.

getScore(int time): Изчислява и връща резултата на играча в зависимост от броя на направените ходове, броя на грешните ходове и времето за завършване на играта.

Клас BoardControler

Клас BoardControler има за цел да свързва fxml (board.fxml) файла със backend-а (Клас Sudoku). В него се намират и елементарни логики използвани с козметична цел за подобряването на визуалната част на приложението.

Методи:

initialize(URL url, ResourceBundle rb): Инициализира контролера след като кореновият му елемент е бил напълно обработен. Задава началното състояние на игралната дъска, деактивира бутоните за отмяна/повторение на ходове и стартира таймера на играта.

timerStart() Стартира таймера на играта, като актуализира етикета за време на всяка секунда.

updateTimeLabel(): Актуализира етикета за време, за да покаже изминалото време в минути и секунди.

findButtonAtPosition(GridPane gridPane, int row, int col) : Намира бутона на зададена позиция в решетката на игралната дъска.

undoRedoAction(ActionEvent event): Обработва действията за отмяна и повторение на потребителя, като актуализира дъската съответно.

boardMove(ActionEvent event): Обработва ходовете на потребителя върху игралната дъска, като актуализира състоянието на дъската, проверява за завършване на играта и управлява изпращането на резултатите.

covertName(String name) :Преобразува името на играча във валиден формат, премахвайки всички символи, които не са буквено-цифрови, и ограничавайки дължината му.

openInputDialog(Stage primaryStage) Отваря диалогов прозорец, за да подкани потребителя да въведе своето име.

difficultyChange(ActionEvent event): Обработва промените в трудността на играта, като нулира игралната дъска, таймера и избраното число.

numberSelect(KeyEvent event): Позволява на потребителя да избере число, което да постави на игралната дъска, или да избере опцията за премахване на число.

updateBoard(): Актуализира визуалното представяне на игралната дъска, за да отрази текущото състояние на играта.

Клас GameServerInterface

Клас GameServerInterface е класът реализираш интерфейлната част на RMI връзката.

Методи:

getPuzzle(int K): Метод за получаване на пъзел от игралния сървър.

Параметър K определя сложността на пъзела. Хвърля RemoteException при възникване на проблем със свързването към сървъра.

cleanUp(): Метод за почистване на ресурсите на сървъра след приключване на играта.

Хвърля RemoteException при възникване на проблем със свързването към сървъра.

receiveStat(Stat stat): Метод за получаване на статистика за играта от клиента и запазването ѝ на сървъра.

Параметър stat съдържа информация за статистиката на играта.

Хвърля RemoteException при възникване на проблем със свързването към сървъра.

Board.fxml

Файлът е XML декларация за създаване на интерфейс за потребителското взаимодействие в JavaFX. Той включва различни елементи от графичния потребителски интерфейс, като бутони, лейбъли и панели. XML структурата е базирана на разположението на елементите на екрана и техните атрибути.

Освен това, XML декларацията включва връзка към външен CSS файл, който дефинира стила на интерфейса.

Клас GameServer

Клас GameServer има за цел да поддържа функционалностите на сървъра. Той генерира пъзели и поддържа статистиките изпращани му обратно

RecordsFile: Стринг, представляващ пътя до файлът за записване на статистиката.

StatisticsFile: Стринг, представляващ пътя до файлът за записване на обобщената статистика.

cellsToKeep: Цяло число, указващо колко клетки да се запазят след генерирането на пъзела.

main(String[] args): Главният метод, изпълняващ генерирането на Sudoku пъзел и извеждането му в конзолата за тестване.

receiveStat(Stat stat): Метод за приемане на статистика от клиентите и записване в съответния файл.

generateStatsFile(): Приватен метод за генериране на обобщена статистика от файла със записи за статистиката.

getPuzzle(int K): Метод за генериране на Sudoku пъзел с даден брой запазени клетки.

cleanUp(): Метод за изчистване на полетата grid и solution след генериране на пъзела.

GameServer(): Конструктор за създаване на нов обект от тип GameServer.

shuffleArray(int[] array): Приватен метод за разбъркване на масив от цели числа по алгоритъма на Fisher-Yates.

Класът включва и множество от помощни методи за генериране на sudoku използвайки backtracking

<https://geeksforgeeks.org/program-sudoku-generator/>