

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАВЧАЛЬНО-НАУКОВИЙ КОМПЛЕКС
«ІНСТИТУТ ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ»
НАЦІОНАЛЬНОГО ТЕХНІЧНОГО УНІВЕРСИТЕТУ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

КАФЕДРА МАТЕМАТИЧНИХ МЕТОДІВ СИСТЕМНОГО
АНАЛІЗУ

Проект
З дисципліни «Комп'ютерні мережі»

Виконав: студент 3-го курсу

гр. КА-71

Столяр М. В.

Прийняв: *Кухарев С.О*

Київ 2020р.

Постановка задачі

Мета:

Розробити програмний продукт, що наглядно демонструватиме роботу клієнт-серверної архітектури, а саме – браузерну гру «крокодил».

Логіка гри:

Гра базується на основі всім відомої гри крокодил, тобто людині загадують слово, вона його демонструє рухами. Учасники повинні вгадати загадане слово. Діджиталізуємо гру.

Кожен учасник повинен авторизуватись, увівши своє ім'я, що буде відображатись іншим. Кожен має право створювати кімнати для гри. Виберемо рандомним чином одного з учасників, що приєднались до деякої кімнати. Йому приходить повідомлення з загаданим словом, і вибраний учасник демонструє його іншим, малюючи його на борді. Всі учасники можуть спілкуватися у чаті. Якщо хтось напише в чаті загадане слово, то він переможе в поточному раунді, і буде рандомно вибрано наступного учасника, що буде демонструвати загадане йому слово.

Реалізація

Засоби:

Надамо перелік технологій, що будуть використані для написання продукту.

Клієнт. Для клієнта будемо використовувати: html – розмітка, css – стилі, js – скрипти.

Сервер. Для серверу використовуватимемо мову програмування java. Для того, щоб реалізувати спілкування між клієнтами на сервером використаємо WebSockets (додамо його до проекту за допомогою Maven).

Перелік підзадач:

Розіб'ємо завдання на підзадачі, та до деяких надамо можливі особливостями реалізації.

1. Створити авторизацію клієнта.

Усім клієнтам відомо, хто зараз онлайн. Якщо зареєстрований клієнт покинув гру, перервав з'єднання (зактив усі вкладки), то при повторному переході у головне меню (головну кімнату), йому не потрібно буде знов авторизуватись.

2. Будь-який клієнт має право на створення кімнати для гри.

Усі клієнти можуть приєднуватись до будь-якої створеної кімнати.

3. Першому, хто зайшов до кімнати, надходить повідомлення зі загаданим словом.

Дане повідомлення бачить лише один учасник (хто малює).

4. Усі учасники мають змогу спілкуватись в чаті кімнати.

5. Повідомлення про вхід/вихід з кімнати

Усі учасники мають змогу бачити, хто ввійшов, чи вийшов з кімнати. Дане повідомлення буде відображатись у чаті.

6. Право на малювання.

Право на малювання має лише той, кому відомо загадане слово. Усі інші учасники не можуть малювати.

7. Віщання на всі з'єднання.

Якщо учасник має декілька відкритих вкладок з кімнатою, то на кожну вкладку надходять повідомлення.

8. Повідомлення з переможцем.

Якщо учасник вгадує загадане слово, то надсилається кожному ім'я переможця та рандомним чином вибирається наступний гравець, хто буде малювати.

Опис класів та функцій:

Клієнт. Для реалізації клієнта застосовувались класичні методи. Окремо, необхідно відміти – для реалізації «малювання» використано Canvas.

Сервер. Надамо опис класів та їх методів.

Класс Room

Поля

Id – Id кімнати

connectedUsers – юзери, що в кімнаті

canvasData – борд для малювання

GameStatus – чи почалась гра

Word – секретне слово

User – обраний юзер)))

Методи

Використано стандартні методи (гетери/сетери).

Класс User

Поля

Name – ім'я юзера

connectionName – ідентифікатор юзера
roomId – до якої кімнати належить юзер

Методи

Використано стандартні методи (гетери/сетери).

Клас GameStarted

Поля

allRegisteredUsers – усі зареєстровані учасники
usersOnline – учасники, що в онлайн
allConnections – скільки в учасника відкритих з'єднань
sockets – з'єднання учасника
createdRooms – створені кімнати.

Методи

GameServer(int port) – ініціалізація сервера
onOpen(WebSocket, ClientHandshake) – виклик при створенні з'єднання
userConnectedNotify(WebSocket) – повідомлення, що юзер онлайн, в головній кімнаті.
showAllUsers() – інформація про усіх юзерів (для розробника)
getUsersConnectionsInRoom(String roomId) – повертає юзерів, що в кімнаті
onMessage(WebSocket conn, String message) – виклик, при надходженні повідомлення з клієнта
sendRooms(String roomId) – повідомляє, які юри онлайн
sendUsers(String roomId) – повідомляє, які кімнати вже створені

Результати роботи:

Було створено ітеративну гру «крокодил», що наглядно демонструє роботу клієнт-серверної архітектури.