// הוספה פשוטה לקוד השרת הקיים שלך

if (type === 'game\_data') {

// פשוט להעביר את נתוני המשחק לכל המשתתפים האחרים בחדר

rooms[room]?.sockets.forEach(s => {

if (s !== socket && s.readyState === WebSocket.OPEN) {

s.send(JSON.stringify(data));

}

});

console.log(`🎮 Game data forwarded in room ${room}`);

}

ובצד הלקוח, התלמידים יוכלו לשלוח ולקבל מצבי משחק בצורה פשוטה:

javascript

Copy

// שליחת מצב משחק

function sendGameState(gameData) {

socket.send(JSON.stringify({

type: "game\_data",

room: currentRoom,

data: gameData

}));

}

// קבלת מצב משחק

socket.onmessage = (event) => {

try {

const message = JSON.parse(event.data);

if (message.type === "game\_data") {

// עדכון מצב המשחק בממשק

updateGame(message.data);

} else {

// טיפול בהודעות צ'אט רגילות

log("📨 " + message.message);

}

} catch (e) {

log("⚠️ הודעה לא תקינה: " + event.data);

}

};

השיפור הקטן שאני מציע לקוד השרת שלך הוא להוסיף סוג הודעה חדש ייעודי למשחקים (type: 'game\_data') שיאפשר לך להפריד בין הודעות צ'אט רגילות לבין נתוני משחק.

בהשוואה לקוד הנוכחי שלך שכבר מטפל בהודעות מסוג chat ו-join, התוספת תהיה:

javascript

Copy

// בקובץ server.js, בתוך פונקציית הטיפול בהודעות

if (type === 'game\_data') {

// העברת נתוני המשחק לכל השחקנים האחרים בחדר (לא כולל השולח)

rooms[room]?.sockets.forEach(s => {

if (s !== socket && s.readyState === WebSocket.OPEN) {

s.send(JSON.stringify(data));

}

});

console.log(`🎮 Game data forwarded in room ${room}`);

}

השיפור הזה עושה שלושה דברים:

1. **הוספת סוג הודעה ייעודי למשחקים** - מפריד בין לוגיקת הצ'אט ללוגיקת המשחק
2. **העברה לכולם חוץ מהשולח** - שבדרך כלל לא צריך לקבל בחזרה את המידע שהוא עצמו שלח
3. **לוג ספציפי** - מתעד פעולות משחק בנפרד מפעולות צ'אט רגילות

זה שינוי מינימלי שיתן לך תשתית טובה יותר למשחקים שבהם הלוגיקה נמצאת בצד הלקוח והשרת רק מעביר נתונים בין השחקנים.