Gameplay API for Frontend Integration

1. REST Endpoints

Join a Public Lobby

POST /game/joinByUsername/{username}

- Returns: {"message": "joined game", "lobbyId": "..."} or {"error": "..."}.

Join a Club Lobby

POST /game/club/joinGame/{clubId}/{username}

- Returns: {"message": "joined club lobby", "lobbyId": "..."} or {"error": "..."}.

Get Lobby Game State

GET /game/state/{lobbyId}

- Returns: "LOBBY", "COUNTDOWN", "RUNNING", or "FINISHED".

Get Players in Lobby

GET /game/players/{lobbyId}

- Returns: List of players with userId, score, playerIdentifier, username.

Get Most Recent Running Game Lobby

GET /game/spectate/recent

Returns: {"lobbyId": "..."} if a public game is running, otherwise {"error": "no running games"}.
This will always exclude club-based games.

Get Random Running Game Lobby

GET /game/spectate/random

Returns: {"lobbyId": "..."} if a public game is running, otherwise {"error": "no running games"}.
Randomly selects from currently active public games.

- Example Success Response:

```
{
  "lobbyId": "public-4a6b8c29-8231-4ff3-ae18-429f7309aa89"
}
```

- Example Error Response:

```
{
  "error": "no running games"
}
```

- Example Success Response:

```
{
  "lobbyId": "public-1f9d3e7b-212a-489d-a4f4-213a0e5c8f92"
}
```

- Example Error Response:

```
{
  "error": "no running games"
}
```

2. WebSocket Endpoints

Player WebSocket Connection

Connect to: /gameSocket/{lobbyId}/{username}

- Send "hit" to attempt ball capture.
- Errors: {"error": "Hit on cooldown"}, {"error": "User not found"}, etc.

Spectator WebSocket Connection

Connect to: /spectatorSocket/{lobbyId}/{username}

- Watch games in real-time.

3. WebSocket Server Broadcast Events

Player Assignments Broadcast

- Sent after 6 seconds during countdown.
- {"players": {"player1": "John", "player2": "Anna", ...}}

Ball Positions Broadcast

- Sent every ~200ms during gameplay.
- {"balls": [{"x": 512.0, "y": 200.0}, {"x": 400.5, "y": 600.2}, ...]}

Game Snapshot Broadcast

- After each hit.
- {"scores": [{"playerIdentifier": "player1", "score": 12}, ...]}

Final Results Broadcast

- At the end of the game.
- {"finalResults": [{"username": "JohnDoe", "score": 15, "isWinner": true}, ...]}

4. Special Notes

- - Players have a 1.5-second enforced cooldown between hits.
- - Lobby IDs are dynamic and must be tracked by frontend after joining.
- - Frontend should listen for "error" messages and handle gracefully.