

Giriş: Multiplayer Shooter Game Protokolü

Protokolün Amacı ve Kapsamı

Multiplayer Shooter Game protokolü, çok oyunculu bir çevrimiçi oyun ortamında istemci (Client) ve sunucu (Server) arasında veri alışverişini sağlamak amacıyla tasarlanmıştır. Bu protokol, çok oyunculu oyun deneyimini geliştirmek ve optimize etmek için tasarlanmış olup, oyuncular arasında kesintisiz ve güvenilir iletişim sağlar.

Protokolün Kapsamı

Protokol, aşağıdaki iletişim senaryolarını kapsar:

1. **Lobi Yönetimi:** Oyuncuların oyun lobileri oluşturmaları, mevcut lobileri listelemesi ve belirli bir lobiye katılması.
2. **Oyun Yönetimi:** Oyunların başlatılması, oyun içi durum güncellemeleri ve oyun sonu durumlarının iletilmesi.
3. **Hareket ve Eylem Yönetimi:** Oyuncuların oyun içi hareketlerini ve eylemlerini (örneğin, ateş etme) sunucuya iletilmesi.
4. **Sohbet Yönetimi:** Oyuncuların sohbet mesajları gönderip alması.
5. **Bağlantı Yönetimi:** İstemcilerin sunucuya bağlanması ve bağlantıyı kesmesi durumunda gerekli işlemlerin gerçekleştirilmesi (örneğin, oyuncuyu lobiden çıkarma, diğer oyunculara bildirme).
6. **Hata Yönetimi:** İstemci veya sunucu tarafında oluşabilecek hataların (bağlantı hatası, geçersiz mesaj formatı vb.) tespit edilmesi ve uygun hata mesajlarının iletilmesi.

Protokolün Temel Özellikleri ve Avantajları

1. **JSON Formatında Mesajlaşma:** Protokol, veri alışverişini basit ve okunabilir hale getiren JSON formatını kullanır. Bu, mesajların kolayca oluşturulmasını ve ayrıştırılmasını sağlar.
2. **TCP/IP Tabanlı İletişim:** Protokol, güvenilir ve sıralı veri iletimini sağlayan TCP/IP protokolünü kullanır. Bu, oyun içi iletişimin kesintisiz ve hatasız olmasını sağlar.
3. **Gerçek Zamanlı İletişim:** Protokol, düşük gecikme süresi ile gerçek zamanlı iletişim sağlar. Bu, oyuncuların hareketlerinin ve eylemlerinin anında sunucuya iletilmesini ve diğer oyunculara aktarılmasını mümkün kılar.
4. **Esneklik ve Genişletilebilirlik:** JSON formatı ve modüler yapı, protokolün kolayca genişletilmesine ve yeni özelliklerin eklenmesine olanak tanır.
5. **Kapsamlı Mesaj Tipleri:** Protokol, oyun içi ihtiyaçları karşılamak için çeşitli mesaj tipleri sunar. Bu mesaj tipleri, lobi yönetiminden oyun içi hareketlere ve sohbet mesajlarına kadar geniş bir yelpazeyi kapsar.
6. **Güvenilir Veri İletimi:** TCP/IP tabanlı iletişim, verilerin güvenli ve doğru bir şekilde iletilmesini sağlar. Bu, oyun içi veri bütünlüğünü ve oyuncu deneyimini artırır.

Terminoloji: Multiplayer Shooter Game Protokolü

Protokol Boyunca Kullanılan Terimler ve Kısaltmalar

1. **İstemci (Client):** Oyunu oynayan oyuncunun cihazında çalışan uygulama. İstemci, sunucuya bağlanır ve oyunla ilgili bilgileri gönderir/alır.
2. **Sunucu (Server):** Oyunun merkezi yönetimini sağlayan uygulama. Sunucu, istemcilerden gelen bilgileri işler ve gerekli güncellemeleri tüm istemcilere iletir.
3. **Lobi (Lobby):** Oyuncuların bir araya geldiği, oyun başlamadan önceki bekleme alanı. Lobiler, oyunların organize edilmesine yardımcı olur.
4. **JSON (JavaScript Object Notation):** Veri değişimi için kullanılan hafif bir veri formatı. Mesajların JSON formatında iletilmesi, veri yapılarının kolayca okunup yazılmasını sağlar.
5. **TCP/IP (Transmission Control Protocol/Internet Protocol):** İnternet üzerinden veri iletimi için kullanılan iletişim protokolü. Bu protokol, verilerin güvenilir ve sıralı bir şekilde iletilmesini sağlar.
6. **FPS (Frames Per Second):** Saniye başına kare sayısı, oyunun akıcılığını belirler. Protokolde oyun güncellemeleri genellikle FPS cinsinden ifade edilir.
7. **UUID (Universally Unique Identifier):** Bir lobi veya oyuncunun benzersiz kimliğini temsil etmek için kullanılan 128-bitlik bir tanımlayıcı. UUID'ler, çakışma olasılığını minimize ederek farklı oyun oturumları ve oyuncuları ayırt etmeyi kolaylaştırır.

Mesaj Formatı

Mesajların Genel Yapısı

Multiplayer Shooter Game protokolündeki mesajlar, JSON formatında düzenlenir. Her mesaj, iki ana bileşenden oluşur: başlık (header) ve içerik (content).

- **Başlık (Header):** Mesajın türünü belirler ve "type" anahtar kelimesi ile tanımlanır.
- **İçerik (Content):** Mesajın içeriğini barındırır ve "content" anahtar kelimesi ile tanımlanır.

Mesaj Türleri ve Ayrıntılı Formatları

İstemci Mesajları:

1. Connect

İstemcinin sunucuya bağlanma isteğini ileten mesajdır.

- **Mesaj Türü:** Connect
- **İçerik:**
 - **playerName:** Lobi oluşturmak isteyen oyuncunun adı. (String, UTF-8 kodlamasında, uzunluk: 1-20 karakter)

Mesaj Örneği:

```
{
  "type": "connect",
  "content": {
    "playerName": "Player1"
  }
}
```

2. Disconnect

İstemcinin sunucudan bağlantısını kesme isteğini ileten mesajdır.

- **Mesaj Türü:** disconnect
- **İçerik:**
 - **playerName:** Bağlantıyı kesmek isteyen oyuncunun adı. (String, UTF-8 kodlamasında, uzunluk: 1-20 karakter)
 - **lobbyId** (isteğe bağlı): Oyuncunun ayrıldığı lobi kimliği. (String, UUID formatında, uzunluk: 36 karakter)

Mesaj Örneği:

```
{
  "type": "disconnect",
  "content": {
    "playerName": "Player1",
    "lobbyId": "unique_lobby_id"
  }
}
```

3. CreateLobby

Yeni bir lobi oluşturma isteğini ileten mesajdır.

- **Mesaj Türü:** createLobby
- **İçerik:**
 - **playerName:** Lobi oluşturmak isteyen oyuncunun adı. (String, UTF-8 kodlamasında, uzunluk: 1-20 karakter)

Mesaj Örneği:

```
{
  "type": "createLobby",
  "content": {
    playerName: "Player1"
  }
}
```

4. GetLobbies

Mevcut lobilerin listesini talep etme isteğini ileten mesajdır.

- **Mesaj Türü:** getLobbies
- **İçerik:** Boş bırakılabilir.

Mesaj Örneği:

```
{
  "type": "getLobbies",
  "content": {}
}
```

5. JoinLobby

Belirtilen lobiye katılma isteğini ileten mesajdır.

- **Mesaj Türü:** joinLobby
- **İçerik:**
 - **lobbyId:** Katılmak istenen lobi kimliği. (String, UUID formatında, uzunluk: 36 karakter)
 - **playerName:** Lobiye katılmak isteyen oyuncunun adı. (String, UTF-8 kodlamasında, uzunluk: 1-20 karakter)

Mesaj Örneği:

```
{
  "type": "joinLobby",
  "content": {
    lobbyId: "unique_lobby_id",
    playerName: "Player1"
  }
}
```

```
}  
}
```

6. StartGame

Oyunun başlatılma isteğini ileten mesajdır.

- **Mesaj Türü:** startGame
- **İçerik:**
 - **lobbyId:** Başlatılacak oyunun lobi kimliği. (String, UUID formatında, uzunluk: 36 karakter)

Mesaj Örneği:

```
{  
  "type": "startGame",  
  "content": {  
    "lobbyId": "unique_lobby_id"  
  }  
}
```

7. Move

Oyuncunun gemisinin hareket bilgilerini sunucuya ileten mesajdır.

- **Mesaj Türü:** move
- **İçerik:**
 - **lobbyId:** Oyun lobisinin kimliği. (String, UUID formatında, uzunluk: 36 karakter)
 - **x:** Oyuncunun x koordinatı. (Integer, Uç değerler oyun ekranı içerisinde olur)
 - **y:** Oyuncunun y koordinatı. (Integer, Uç değerler oyun ekranı içerisinde olur)
 - **playerName:** Hareket eden oyuncunun adı. (String, UTF-8 kodlamasında, uzunluk: 1-20 karakter)

Mesaj Örneği:

```
{  
  "type": "move",  
  "content": {  
    "lobbyId": "unique_lobby_id",  
    "x": 100,  
    "y": 200,  
    "playerName": "Player1"  
  }  
}
```

8. Shoot

Oyuncunun ateş etme eylemini sunucuya ileten mesajdır.

- **Mesaj Türü:** shoot.

- **İçerik:**
 - **lobbyId:** Oyun lobisinin kimliği. (String, UUID formatında, uzunluk: 36 karakter)
 - **x:** Oyuncunun ateş ettiği x koordinatı. (Integer, değer aralığı: 0-800)
 - **y:** Oyuncunun ateş ettiği y koordinatı. (Integer, değer aralığı: 0-600)
 - **playerName:** Ateş eden oyuncunun adı. (String, UTF-8 kodlamasında, uzunluk: 1-20 karakter)

Mesaj Örneği:

```
{
  "type": "shoot",
  "content": {
    "lobbyId": "unique_lobby_id",
    "x": 100,
    "y": 200,
    "playerName": "Player1"
  }
}
```

9. Chat

Oyuncunun sohbet mesajını sunucuya ileten mesajdır.

- **Mesaj Türü: chat**
- **İçerik:**
 - **lobbyId:** Oyun lobisinin kimliği. (String, UUID formatında, uzunluk: 36 karakter)
 - **message:** Gönderilen sohbet mesajı. (String, UTF-8 kodlamasında, uzunluk: 1-256 karakter)
 - **playerName:** Mesajı gönderen oyuncunun adı. (String, UTF-8 kodlamasında, uzunluk: 1-20 karakter)

Mesaj Örneği:

```
{
  "type": "chat",
  "content": {
    "lobbyId": "unique_lobby_id",
    "message": "Hello World",
    "playerName": "Player1"
  }
}
```

Sunucu Mesajları:

1. LobbiesList

Mevcut lobilerin listesini istemciye ileten mesajdır.

- **Mesaj Türü: lobbiesList**

- **İçerik:**
 - **lobbies:** Array of Strings, mevcut lobilerin listesi. (Her biri UUID formatında, uzunluk: 36 karakter)

Mesaj Örneği:

```
{
  "type": "lobbiesList",
  "content": {
    "lobbies": ["lobby1", "lobby2", "lobby3"]
  }
}
```

2. LobbyUpdate

Lobideki güncellemeleri ve oyuncu listesini istemciye ileten mesajdır.

- **Mesaj Türü:** lobbyUpdate
- **İçerik:**
 - **lobbyId:** String, lobi kimliği. (UUID formatında, uzunluk: 36 karakter)
 - **players:** Array of Strings, lobideki oyuncuların listesi. (Her biri UTF-8 kodlamasında, uzunluk: 1-20 karakter)

Mesaj Örneği:

```
{
  "type": "lobbyUpdate",
  "content": {
    "lobbyId": "unique_lobby_id",
    "players": ["Player1", "Player2"]
  }
}
```

3. LobbyCreated

Yeni bir lobi oluşturulduğunu istemciye bildiren mesajdır.

- **Mesaj Türü:** lobbyCreated
- **İçerik:**
 - **lobbyId:** String, yeni oluşturulan lobi kimliği. (UUID formatında, uzunluk: 36 karakter)
 - **players:** Array of Strings, lobideki oyuncuların listesi. (Her biri UTF-8 kodlamasında, uzunluk: 1-20 karakter)

Mesaj Örneği:

```
{
  "type": "lobbyCreated",
  "content": {
    "lobbyId": "unique_lobby_id",
```

```
        "players": ["Player1"]
    }
}
```

4. Chat

Bir sohbet mesajını istemciye ileten mesajdır.

- **Mesaj Türü:** chat
- **İçerik:**
 - **lobbyId:** String, lobi kimliği. (UUID formatında, uzunluk: 36 karakter)
 - **message:** String, gelen sohbet mesajı. (UTF-8 kodlamasında, uzunluk: 1-256 karakter)

Mesaj Örneği:

```
{
    "type": "chat",
    "content": {
        "lobbyId": "unique_lobby_id",
        "message": "Player1: Hello World"
    }
}
```

5. GameState

Oyun durumunu ve güncellemelerini istemciye ileten mesajdır.

- **Mesaj Türü:** gameState
- **İçerik:**
 - **lobbyId:** String, lobi kimliği. (UUID formatında, uzunluk: 36 karakter)
 - **gameState:** Object, oyun durumu.
 - **Players:** Array of Strings, Oyundaki oyuncuların listesi.
 - **gameStateType:** GameOver, GameGoingOn, GameNotStart
 - **ships:** Array of Objects, oyuncu gemilerinin durumu.
 - **id:** Integer, geminin kimliği.
 - **x:** Integer, x koordinatı. (Değer aralığı: 0-800)
 - **y:** Integer, y koordinatı. (Değer aralığı: 0-600)
 - **health:** Integer, geminin sağlığı. (Değer aralığı: 0-100)
 - **score:** Integer, oyuncunun skoru. (Değer aralığı: 0 ve üzeri)
 - **enemies:** Array of Objects, düşmanların durumu.
 - **id:** Integer, düşmanın kimliği.
 - **x:** Integer, x koordinatı. (Değer aralığı: 0-800)

- **y**: Integer, y koordinatı. (Değer aralığı: 0-600)
- **health**: Integer, düşmanın sağlığı. (Değer aralığı: 0-100)
- **bullets**: Array of Objects, mermilerin durumu.
 - **id**: Integer, merminin kimliği.
 - **x**: Integer, x koordinatı. (Değer aralığı: 0-800)
 - **y**: Integer, y koordinatı. (Değer aralığı: 0-600)

Mesaj Örneği:

```
{
  "type": "gameState",
  "content": {
    "lobbyId": "unique_lobby_id",
    "gameState": {
      "gameStateType": GameOver
      "ships": [
        {
          "id": 1,
          "x": 100,
          "y": 200,
          "health": 3,
          "score": 10
        }
      ],
      "enemies": [
        {
          "id": 1,
          "x": 400,
          "y": 300,
          "health": 1
        }
      ],
      "bullets": [
        {
          "id": 1,
          "x": 200,
          "y": 100
        }
      ]
    }
  }
}
```

6. LobbyJoinState

İstemcinin lobiye katılım durumunu bildiren mesajdır.

- **Mesaj Türü**: lobbyJoinState
- **İçerik**:

- **message:** String, katılma durumu mesajı. (UTF-8 kodlamasında, uzunluk: 1-256 karakter)

Mesaj Örneği:

```
{
  "type": "lobbyJoinState",
  "content": {
    "message": "Lobby is full"
  }
}
```

7. ConnectionStatus

Sunucunun istemciye bağlantı durumunu bildirmesi.

- **Mesaj Türü:** connectionStatus
- **İçerik:**
 - **message:** Bağlantı durumu hakkında bilgilendirici mesaj. Connected ve Disconnected.
 - **playerName:** Mesajın gönderildiği oyuncunun adı.

Mesaj Örneği:

```
{
  "type": "connectionStatus",
  "content": {
    "playerName": "Player1"
    "message": "Connected"
  }
}
```

8. Error

İstemci veya sunucunun karşılaştığı bir hatayı bildirmesi.

Mesaj Türü: error

- **İçerik:**
 - **message:** Hata hakkında bilgilendirici mesaj. (String, UTF-8 kodlamasında, uzunluk: 1-100 karakter)

GameCouldNotBeStarted: İstemcinin **startGame** isteği yerine getirilemedi.

MessageCouldNotBeSent: İstemcinin **chat** isteği yerine getirilemedi.

LobbyCouldNotBeCreated: İstemcinin LobbyCreated isteği yerine getirilemedi.

Her Mesajın İletiminde Yapılacak İşlemler Listesi

Mesaj Türü	İstemci İşlemleri	Sunucu İşlemleri
Connect	İstemci connect mesajını sunucuya gönderir.	<ol style="list-style-type: none">Oyuncu adının benzersiz olup olmadığını kontrol eder.<ul style="list-style-type: none">Eğer benzersizse, bağlantıyı başarılı olarak işaretler ve connectionStatus mesajı gönderir.Eğer benzersiz değilse, bağlantıyı başarısız olarak işaretler ve connectionStatus mesajı gönderir.
Disconnect	İstemci disconnect mesajını sunucuya gönderir.	<ol style="list-style-type: none">Alınan mesajın ait olduğu istemci tüm oyunlardan çıkartılır.İstemcinin oyun durumu kontrol edilir. Ait olduğu oyundan tüm varlıkları silinir. gameStatus mesajı güncellenir ve ilgili istemcilere yollanır.connectionStatus mesaj kısmı "disconnected" olacak şekilde hazırlanır.
ConnectionStatus	<ol style="list-style-type: none">Eğer mesaj "success" ise:<ul style="list-style-type: none">Kullanıcıya bağlantının başarılı olduğu bildirilir.Lobi seçenekleri gösterilir.Eğer mesaj "fail" ise:<ul style="list-style-type: none">Kullanıcıya hata mesajı gösterilir.Eğer mesaj "disconnected" ise oyun kapatılır.	<ol style="list-style-type: none">connectionStatus mesajı hazırlanır.<ul style="list-style-type: none">Eğer bağlantı başarılı ise başarılı mesajı eklenir.Eğer bağlantı başarısız ise başarısız mesajı eklenir.
CreateLobby	Kullanıcı bir lobi oluşturmak istediğinde, createLobby mesajını oluşturur ve sunucuya gönderir.	<ol style="list-style-type: none">Mesajı alır ve yeni bir lobi oluşturur.Oyuncunun adını ve lobi kimliğini kaydeder.Oyuncuyu yeni lobiye ekler ve lobbyUpdate mesajını istemciye gönderir.Lobi oluşturulduğunu belirten bir lobbyCreated mesajını istemciye gönderir.
joinLobby	Kullanıcı belirli bir lobiye katılmak istediğinde, joinLobby mesajını oluşturur Mesaj içeriği, lobi kimliğini (lobbyId) ve oyuncunun adını (playerName) içerir.	<ol style="list-style-type: none">Sunucu, joinLobby mesajını alır.Belirtilen lobi kimliğini doğrular.<ul style="list-style-type: none">Lobi dolu değilse, oyuncuyu lobiye ekler ve lobbyJoinState mesajı ile katılım durumunu istemciye bildirir.Lobi doluysa veya lobi bulunamazsa, uygun bir hata mesajını içeren lobbyJoinState mesajını istemciye gönderir.Lobideki oyuncuların listesini günceller ve istemciye iletir.
startGame	<ol style="list-style-type: none">Kullanıcı oyunu başlatmak istediğinde, startGame mesajını oluşturur.<ul style="list-style-type: none">Mesaj içeriği, lobi kimliğini (lobbyId) içerir.startGame mesajı JSON formatında oluşturulur ve sunucuya gönderilir.	<ol style="list-style-type: none">Sunucu, startGame mesajını alır.Oyunu başlatır.Tüm lobi üyelerine oyunun başladığını bildiren bir gameState mesajı gönderir.Oyun durumunu güncelleyerek tüm istemcilere iletir.

<p>move</p>	<ol style="list-style-type: none"> 1. Kullanıcı gemisini hareket ettirdiğinde, move mesajını oluşturur. <ul style="list-style-type: none"> • Mesaj içeriği, lobi kimliğini (lobbyId), oyuncunun yeni x ve y koordinatlarını (x, y), ve oyuncunun adını (playerName) içerir. 2. move mesajı JSON formatında oluşturulur ve sunucuya gönderilir. 	<ol style="list-style-type: none"> 1. Sunucu, move mesajını alır. <ul style="list-style-type: none"> • Mesajı parse eder ve içeriğindeki lobi kimliği, oyuncunun yeni koordinatları ve oyuncunun adını alır. 2. Oyuncunun yeni pozisyon bilgilerini günceller. <ul style="list-style-type: none"> • Sunucu, ilgili oyuncunun gemisinin x ve y koordinatlarını günceller. 3. Oyun içi objelerinin çarpışmaları kontrol eder. <ul style="list-style-type: none"> • Güncellenmiş pozisyon bilgilerine göre, geminin diğer oyun içi objeler (düşman gemiler, mermiler, vb.) ile çarpışma durumu kontrol edilir. • Çarpışma tespiti durumunda: <ul style="list-style-type: none"> ○ Düşman ile Çarpışma: Oyuncunun gemisinin canı azalır. Düşmanın canı azalır. Canı sıfıra inen düşman oyun alanından kaldırılır. ○ Mermi ile Çarpışma: Oyuncunun gemisinin canı azalır. Mermi oyun alanından kaldırılır. ○ Diğer Oyuncu ile Çarpışma: Her iki oyuncunun canı azalır. 4. Güncellenen oyun durumunu içeren gameState mesajı oluşturulur. <ul style="list-style-type: none"> • Bu mesaj, lobi kimliği ve tüm oyuncuların, düşmanların ve mermilerin güncellenmiş durumlarını içerir. 5. Mesaj, JSON formatında tüm lobi üyelerine gönderilir. <ul style="list-style-type: none"> • Güncellenen oyun durumunu tüm lobi üyelerine ileten bir gameState mesajı gönderir.
<p>Shoot</p>	<ol style="list-style-type: none"> 1. Kullanıcı ateş ettiğinde, shoot mesajını oluşturur. <ul style="list-style-type: none"> • Mesaj içeriği, lobi kimliğini (lobbyId), oyuncunun ateş ettiği x ve y koordinatlarını (x, y), ve oyuncunun adını (playerName) içerir. 2. Shoot mesajı JSON formatında oluşturulur ve sunucuya gönderilir. 	<ol style="list-style-type: none"> 1. Sunucu, shoot mesajını alır. <ul style="list-style-type: none"> • Mesajı parse eder ve içeriğindeki lobi kimliği, oyuncunun ateş ettiği koordinatlar ve oyuncunun adını alır. 2. Yeni bir mermi objesi oluşturur ve oyun alanına ekler. <ul style="list-style-type: none"> • Merminin başlangıç pozisyonunu ve hızını belirler. 3. Oyun içi objelerinin çarpışmalarını kontrol eder. <ul style="list-style-type: none"> • Güncellenmiş pozisyon bilgilerine göre, merminin diğer oyun içi objeler (düşman gemiler, diğer mermiler, vb.) ile çarpışma durumu kontrol edilir. • Çarpışma tespiti durumunda: <ul style="list-style-type: none"> ○ Düşman ile Çarpışma: Düşmanın canı azalır. Canı sıfıra inen düşman oyun alanından kaldırılır. Ateş eden oyuncunun skoru artar. ○ Diğer Mermi ile Çarpışma: Her iki mermi oyun alanından kaldırılır. 4. Güncellenen oyun durumunu tüm lobi üyelerine ileten bir gameState mesajı gönderir. <ul style="list-style-type: none"> • Güncellenen oyun durumunu içeren gameState mesajı oluşturulur. • Bu mesaj, lobi kimliği ve tüm oyuncuların, düşmanların ve mermilerin güncellenmiş durumlarını içerir. • Mesaj, JSON formatında tüm lobi üyelerine gönderilir.

Chat	<ol style="list-style-type: none"> 1. Kullanıcı sohbet mesajı yazdığında, chat mesajını oluşturur. <ul style="list-style-type: none"> • Mesaj içeriği, lobi kimliğini (lobbyId), gönderilen mesajı (message), ve mesajı gönderen oyuncunun adını (playerName) içerir. 2. Chat mesajı JSON formatında oluşturulur ve sunucuya gönderilir. 3. Sunucudan gelen mesajlar arayüzde gösterilir. 	<ol style="list-style-type: none"> 1. Sunucu, chat mesajını alır. <ul style="list-style-type: none"> • Mesajı parse eder ve içeriğindeki lobi kimliği, mesaj ve oyuncunun adını alır. 2. Sunucu, chat mesajını işleme koyar ve lobideki diğer oyunculara iletir. <ul style="list-style-type: none"> • Mesajın lobideki tüm oyunculara iletilmesi için yeni bir chat mesajı oluşturur. • Mesaj içeriğine gönderenin oyuncu adı eklenir. 3. Chat mesajı, JSON formatında tüm lobi üyelerine iletilir.
getLobbies	<ol style="list-style-type: none"> 1. Kullanıcı mevcut lobileri görmek istediğinde, getLobbies mesajını oluşturur. <ul style="list-style-type: none"> • Mesaj içeriği boş bırakılabilir. 2. GetLobbies mesajı JSON formatında oluşturulur ve sunucuya gönderilir. 	<ol style="list-style-type: none"> 1. Sunucu, getLobbies mesajını alır. <ul style="list-style-type: none"> • Mevcut lobilerin listesini derler. • Lobi kimliklerini içeren bir liste oluşturur. 2. Sunucu, mevcut lobilerin listesini içeren bir lobbiesList mesajı oluşturur. <ul style="list-style-type: none"> • Mesaj içeriği, mevcut lobilerin kimliklerini içerir. 3. LobbiesList mesajı JSON formatında istemciye iletilir.
gameState	<ol style="list-style-type: none"> 1. Sunucudan gelen gameState mesajını alır. <ul style="list-style-type: none"> • Mesajı parse eder ve içeriğindeki bilgileri alır. • Oyun durumu bilgilerini içerir. 2. Gelen oyun durumunu farklı işlemler yapar: <ul style="list-style-type: none"> • GameNotStart: Oyun başlamamış, lobi ekranında bekletilir. • GameGoingOn: Oyun devam ediyor, oyuncu gemilerinin, düşmanların ve mermilerin yeni pozisyonlarını günceller. Oyuncuların sağlık ve skor bilgilerini günceller. • GameOver: Oyun bitmiş, oyun bitiş ekranına yönlendirir ve skorları gösterir. 	<ol style="list-style-type: none"> 1. Oyun döngüsünde her güncellemede: <ul style="list-style-type: none"> • Oyuncu gemilerinin, düşmanların ve mermilerin yeni pozisyonlarını ve durumlarını hesaplar. • Oyun durumunu belirler gameStateType(GameNotStart, GameGoingOn, GameOver). • Oyun durumu GameOver olduğunda oyuncu listesi mesaja Skorlarla birlikte ve oyun sonucunu belirleyen sırala ile eklenir. Eğer GameNotStart ve GameGoingOn durumundaysa sıralama yapılmaz. 2. Güncellenen oyun durumunu içeren gameState mesajını oluşturur: <ul style="list-style-type: none"> • Mesaj içeriği, oyuncu gemileri, düşmanlar ve mermilerin güncellenmiş durumlarını ve oyun durumunu içerir. 3. Güncellenen oyun durumunu içeren gameState mesajını tüm lobi üyelerine gönderir: <ul style="list-style-type: none"> • Mesaj, JSON formatında tüm lobi üyelerine iletilir.
disconnect	<ol style="list-style-type: none"> 1. Kullanıcı bağlantıyı kesmek istediğinde, disconnect mesajını oluşturur. <ul style="list-style-type: none"> • Mesaj içeriği, oyuncunun adını (playerName) ve lobi kimliğini (lobbyId) içerir. 2. Disconnect mesajı JSON formatında oluşturulur ve sunucuya gönderilir. 	<ol style="list-style-type: none"> 1. Sunucu, disconnect mesajını alır. <ul style="list-style-type: none"> • Mesajı parse eder ve içeriğindeki oyuncu adı ve lobi kimliğini alır. 2. Sunucu, oyuncuyu lobiden ve bağlantı listesinden çıkarır. <ul style="list-style-type: none"> • Oyuncunun lobi içindeki varlığını ve oyun içi durumunu günceller. 3. Sunucu, lobideki diğer oyunculara disconnect bildirimini iletir. <ul style="list-style-type: none"> • Oyuncunun lobiden ayrıldığını belirten bir disconnect mesajı oluşturur ve tüm lobi üyelerine iletir.

Sonlu Durum Makinesi (Finite State Machine - FSM)



