Netcode-Mechaniken in Echtzeitanwendungen: Analyse von Client-Side Prediction, Interpolation und Lag Compensation am Beispiel einer Unity-basierten Zielumgebung

Alexander Seitz

14.05.2025

Contents

1 Einleitung

Moderne Echtzeitanwendungen, insbesondere im Bereich der Computerspiele, stellen hohe Anforderungen an die zugrunde liegende Netzwerkarchitektur. Die Synchronisation von Spielzuständen, die Minimierung von Latenzzeiten sowie ein robuster Umgang mit Paketverlust und Jitter sind entscheidend für eine positive Nutzererfahrung. Multiplayer-Systeme, wie sie etwa mit Unity entwickelt werden, müssen deshalb nicht nur funktional, sondern auch leistungsfähig und fehlertolerant sein.

Die vorliegende Arbeit beschäftigt sich mit den theoretischen und technischen Grundlagen der Netzwerkkommunikation in Mehrspielerumgebungen. Anhand eines prototypischen Projekts in Unity werden exemplarisch zentrale Mechanismen wie Reconciliation, Interpolation sowie die Kommunikation im Client-Server-Modell umgesetzt und analysiert.

Im Folgenden werden zunächst die Motivation und Zielsetzung der Arbeit erläutert. Anschließend wird der strukturelle Aufbau der Arbeit vorgestellt.

3 C# Beispiel: Unity Netcode

Das folgende Beispiel zeigt, wie man eine einfache `NetworkManager`-Klasse in Unity erstellen kann, die bei der Verbindung von Server und Client hilft. Moderne Echtzeitanwendungen stellen hohe Anforderungen penis an die Netzwerkarchitektur [1].

```
using Unity;
using Unity.Netcode;

public class SimpleNetworkManager : NetworkBehaviour
{
    public override void OnNetworkSpawn()
    {
        base.OnNetworkSpawn();

        //kek
        if (IsServer  )
        {
            Debug.Log("Server gestartet");
        }
        else
        {
            Debug.Log("Client verbunden");
        }
    }

    [ServerRpc]
    public void SendMessageServerRpc(string message)
    {
        Debug.Log("Nachricht vom Client empfangen: " + message);
    }


    [ClientRpc]
    public void SendMessageClientRpc(string message)
    {
        Debug.Log("Nachricht vom Server empfangen: " + message);
    }
}
```

Listing 1: Unity Netcode Example

4 Literaturverzeichnis

References

[1] Vercel. *What is Next.js?* n.d. URL: https://nextjs.org/docs#what-is-nextjs (visited on 01/16/2025).