

# Bachelorthesis

## Konzeption und Umsetzung von Matchmaking Architekturen in Online-Spielen

zur Erlangung des akademischen Grades

Bachelor of Science

eingereicht im Fachbereich Mathematik, Naturwissenschaften und Informatik an der  
Technischen Hochschule Mittelhessen

von

Leon Schäfer

2. Dezember 2021

Referent: Dr. Dennis Priefer Korreferent: < noch nicht festgelegt >



# Erklärung der Selbstständigkeit

Hiermit versichere ich, die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie die Zitate deutlich kenntlich gemacht zu haben.

Gießen, den 2. Dezember 2021

Leon Schäfer



Abstract:

Der Einstieg in die Multiplayer-Spieleprogrammierung gestaltet sich für angehende Entwickler als schwierig. Ein erstes eigenes Projekt umzusetzen erfordert viel Einarbeitung und es gibt viele Einstiegshürden. Insbesondere der Teil des Matchmakings ist ohne Budget oft schwierig umzusetzen. Es gibt schlichtweg zu wenig Einstiegshilfen in diesem Teil, da jedes Spiel sehr individuelle Anforderungen mit sich bringt.

Das Ziel dieser Arbeit ist, eine Einstiegshilfe für das Matchmaking in Multiplayer-Projekten zu geben. Hierzu wird eine abstrakte Vorlage konstruiert, welche auf viele Multiplayer Use-Cases anwendbar ist.

Dazu wurden abstrahierte Konzepte erstellt, welche das Matchmaking in Multiplayer-Spielen abbilden und auf möglichst viele Spiele anwendbar sein sollen. Diese Konzepte bestehenden aus einzelnen Software-Komponenten, welche selbst noch keine konkrete Implementierungsvorgabe sind, sondern lediglich das Konzept der Komponente abstrakt beschreiben.

Als Proof of Concept wurden die beschriebenen Konzepte anhand einer Implementierung in einem Multiplayer-Spiel in der Unity Engine, mithilfe des Mirror Frameworks implementiert.

Ideensammlung für Gliederung::

## 1. Einführung:

Matchmaking -> Moderne Matchmaking-Architekturen/Algorithmen von heute (Beispiele) -> Elo-Zahl basiertes Matchmaking (Rangliste) + Lobby -> Generell: Parameter-basiertes Matchmaking -> Server Browser -> Lobbys -> ... -> Implementierungsbeispiele der "realen Welt"

## 2. Ableitbare Konzepte:

-> Offline/Online Verwaltung durch Network-Manager -> verwaltet u.a. Abbruchbedingungen (Manuell / Automatisch) für Client und Server -> Lobby-Manager (Verwaltung der Spieler in Lobby-Szenen) (optional) -> Prepare-InGame-Manager (Initialisierung v. Spawn Punkten, Objektspawn v. abh. Objekten) -> In-Game-Player-Manager (Verwaltung der Spieler in Spiel-Szenen) -> In-Game-Object-Manager (Verwaltung v. Objekt und Spieler-Respawn) -> In-Game-Progress-Manager (Verwaltung v. Spielfortschritt, u.a. zuständig für Spielbeendigung) -> Dieser ist ebenfalls dafür zuständig, nach Beendigung einer Spiel-Session die Spieler in die offline-Szene oder zurück in ihre "Lobby-Szene" zu

---

befördern. Sollte die Implementierung eine Online-Lobby unterstützen, übernimmt im Anschluss wieder der "Pre-Game-Player-Manager"

-> HTTP Server / REST API / Dedicated Server -> Unterscheidung Hosting (Client Host / Dedicated Server Host) -> Einordnung der generalisierten Konzepte für das Matchmaking

### 3. Implementierung 1 :

Genutzte Frameworks / Technologien

Grundlagen des Mirror Frameworks: -> worauf basiert es eigentlich? -> Auszug Features | diese erklären. -> Network-Manager -> NetworkManager Callbacks -> Network Identity / Network Behavior / Network Transform -> Network Behavior Callbacks -> Server und Clientcode in einer Datei -> Dedicated Server vs. Self-Host -> Synchronization -> Remote Actions -> Player Game Objects -> Anticheat -> Transports

### 4. Implementierung 2:

-> Beispiel Implementierung Matchmaking System (Hide n seek) 5. Abschluss:

# Inhaltsverzeichnis

1	Einführung	1
1.1	Problembeschreibung . . . . .	1
1.2	Motivation . . . . .	1
1.3	Ziele dieser Arbeit . . . . .	2
1.4	Abgrenzung . . . . .	3
1.5	Vorgehensweise . . . . .	3
1.6	Struktur der Arbeit . . . . .	3
2	Hintergrund	5
3	Konzepte	7
4	Realisierung	9
5	Zusammenfassung	11
5.1	Fazit . . . . .	11
5.2	Auswertung . . . . .	11
5.3	Weitere Ansätze . . . . .	11
5.4	Nächste Schritte . . . . .	11
5.5	Ausblick . . . . .	11
	Literaturverzeichnis	13
	Abkürzungsverzeichnis	14
	Abbildungsverzeichnis	15
	Tabellenverzeichnis	17
	Listings	19
A	Anhang 1	21
B	Anhang 2	23





# 1 Einführung

## 1.1 Problembeschreibung

Konzeption und Entwicklung von Online-Multiplayerspielen stellt angehende Entwickler vor große Herausforderungen. Die Einstiegshürden sind groß, der Dschungel an Technologien und Frameworks unübersichtlich. Einheitliche Vorgehensmodelle und Best-Practises gibt es nicht oder beziehen sich stets auf eine spezielle Art von Spieltyp. Das Thema Hosting ist ebenfalls ein komplexer Aspekt eines solchen Projekts, welcher von Anfängern oft nicht sofort durchschaut wird. Durch erste Projekterfahrung gewinnen die Entwickler nach und nach das Know-How, ohne ein erfahrenes Team sind die Chancen auf Misserfolg jedoch hoch.

### Why Multiplayer is Hard

One of the key differences between a multiplayer and single player game is that in a multiplayer experience the game exists independently of the player. This isn't required in a single player experience, where the game and the player's 'view' of the game are often the same thing. The only bits we care about in a single player game are the bits your player can see right now, the world only exists to serve that player's needs.

Some of the 'hard' problems are there at development time. For example, sending messages across the internet takes an amount of time, so you need to work out how to asynchronously communicate what is happening back and forth, and deal with any cases where your clients and servers get a bit out of sync with what is happening. Just the logistics of running a server on your development PC, running a local version of any other services you're using (matchmaking, save data storage, etc) then running multiple clients to test a feature isn't always easy or fun. [?]

### 1.2 Motivation

Die Spieleindustrie ist im Wandel. Reine Singleplayer-Spiele, also Spiele, die ohne andere menschliche Mitspieler auskommen werden immer seltener entwickelt, da die Erfolgsaussichten um ein vielfaches schlechter erscheinen. Die drei bestplatzierten Titel der meistverkauften PC-und Konsolenspiele 2019 (FIFA 2020, Call of Duty: Modernware und Mario Kart 8 Deluxe) in Deutschland enthalten einen großen Multiplayer-Anteil.[gam20]

Deutlicher wird der Trend, wenn die Statistik der meistgespielten Titel auf der Online Spiele-Vertreibsplattform Steam betrachtet wird. Das mit Abstand meistgespielte Spiel 2020 ist das Multiplayer-First-Person-Shooter Spiel "Counter-Strike: Global Offensive" gefolgt von "DOTA 2" auf Platz 2. Beide Spiele sind reine Multiplayer-Titel.[Git21]

Einer internationalen Umfrage zufolge geben 60% der Befragten an, seit Beginn der Corona Pandemie mehr Multiplayer-Videospiele zu spielen. [Sim20]

Wegen der Marktentwicklung ist es für angehende Entwickler, welche in der Spieleindustrie Fuß fassen möchten ist es auf lange Sicht eine gute Entscheidung, sich im mit dem Bereich Multiplayer-Spieleentwicklung auseinanderzusetzen.

Angetrieben wurde diese Arbeit durch den eigenen Wunsch professionell ein Multiplayer-Spielprojekt umzusetzen und den Grundstein für ein einheitliches Vorgehen für zukünftige Projekte zu legen. Von diesem Grundstein sollen möglichst viele weitere Projekte profitieren.

### 1.3 Ziele dieser Arbeit

Um einen einfacheren Einstieg in die Multiplayer-Spieleentwicklung zu gewährleisten, beschäftigt sich diese Arbeit mit einigen Grundlagen, die Entwicklern einen Vorteil bei der Gestaltung ihrer Software-Architektur bringen soll.

Ein Entwickler soll außerdem nach dem Studium dieser Arbeit eine konkrete Vorstellung davon besitzen, welche Voraussetzungen ein Framework oder eine Game-Engine erfüllen muss, damit die Konzepte möglichst leicht umzusetzen sind.

Die Arbeit soll eine Blaupause liefern, welche andere Entwickler nutzen können, um einen leichteren Einstieg in ein Multiplayer-Projekt zu gewährleisten. Diese Blaupause soll allerdings derart generisch sein, dass sie unabhängig von einem konkreten Implementierungskontext funktioniert. Die beschriebenen Konzepte sollen auf möglichst viele Szenarien anwendbar sein. Der Fokus hier soll auf dem Matchmaking liegen. Der Begriff Matchma-

king wird später noch genauer erläutert. [Dennis, ist spätere Erklären von Begriffen in Ordnung?]

Als Proof-of-Concept soll die eigene Implementierung eines Hide-and-Seek Multiplayer-Spiels dienen, welches sich an den abstrahierten Konzepten orientiert, bzw. diese implementiert.

Folgende Forschungsfragen soll die Arbeit beantworten:

- Ist es durch abstrakte Konzepte möglich, Entwicklern beim Einstieg in das Themenfeld zu helfen?
- Welche Einstiegshürden gibt es überhaupt? Wobei würden sich angehende Entwickler mehr Unterstützung wünschen?

## 1.4 Abgrenzung

Die Arbeit ist jedoch keine Schritt-für-Schritt Anleitung. Implementierungsdetails müssen vom Benutzer selbstständig konzipiert und umgesetzt werden. Es wird außerdem lediglich Rücksicht genommen auf Spielkonzepte, welche in der heutigen Industrie gängig sind. Es ist durchaus denkbar, dass zukünftige Multiplayer / Matchmaking Architekturen nicht mehr mit den hier beschriebenen Konzepten umzusetzen sind.

Es ist außerdem nicht final bewiesen, dass die beschriebenen Methoden tatsächlich einen Mehrwert bei unterschiedlichen Projekten bieten. Die Konzepte wurden lediglich an einem eigenen Projekt getestet und validiert. Sie basieren auf der individuellen Erfahrung des Autors. Um zu testen, ob die Konzepte auch praxistauglich für verschiedene Arten von Projekten sind, muss eine weitergehende Untersuchung stattfinden.

Die erarbeiteten Konzepte sind außerdem unabhängig von einer Game-Engine oder einem Framework. Durchaus ist es aber möglich, dass ein bestehendes Framework oder eine Game-Engine Hilfestellung bei der Implementierung der Konzepte bietet.

## 1.5 Vorgehensweise

Aus der bisher gesammelten Praxiserfahrung werden Konzepte abstrahiert, welche jeweils einen bestimmten Einsatzzweck erfüllen sollen. Hierbei wurde auf Implementierungsdetails verzichtet und lediglich ein generelles Konzept erarbeitet, an welchem der Entwickler sich während des Implementierungsprozesses orientieren kann.

Diese Konzepte behandeln Probleme, welche im Entwicklungsprozess von Multiplayer-Spielen auftreten können.

Jedes dieser Konzepte wurde in einer Beispielimplementierung umgesetzt, welche im Kapitel "Realisierung" näher vorgestellt wird.

### 1.6 Struktur der Arbeit

Dies Arbeit besteht aus der Einführung, in welcher die generelle Problemstellung erläutert, und die Forschungsfragen gestellt wurden. Ebenfalls wurde die Relevanz des Themengebiets verdeutlicht und eingeordnet.

Im Anschluss wird der Hintergrund erklärt, dort sind alle Informationen zu finden, um die späteren Konzepte zu verstehen. Hier werden ebenfalls Begrifflichkeiten und Grundlagen erläutert [Dennis, kann ich das so machen?]

Das Kapitel "Konzepte" behandelt nun die erarbeiteten Konzepte, welche in Summe die oben beschriebene Blaupause abbilden.

Anschließend werden die beschriebenen Konzepte in einer Beispielimplementierung umgesetzt, diese bildet das Kapitel "Realisierung". In der Zusammenfassung werden die Ergebnisse der Arbeit beleuchtet und ein Ausblick aufgezeigt.

## 2 Hintergrund



## 3 Konzepte





## 4 Realisierung



## 5 Zusammenfassung

### 5.1 Fazit

### 5.2 Auswertung

### 5.3 Weitere Ansätze

### 5.4 Nächste Schritte

### 5.5 Ausblick



## Literaturverzeichnis

- [gam20] GAME - VERBAND DER DEUTSCHEN GAMES-BRANCHE E.V.: Jahresreport der deutschen Games-Branche 2020 (2020), URL <https://www.game.de/wp-content/uploads/2020/08/game-Jahresreport-2020.pdf>
- [Git21] GITHYP: Most played games on Steam in 2020, by hourly average number of players (February 2021), URL <https://www.statista.com/statistics/656319/steam-most-played-games-average-player-per-hour/>
- [Sim20] SIMON-KUCHER & PARTNERS: Impact of COVID-19 on the frequency of playing multiplayer video games worldwide as of June 2020 (2020), URL <https://www.statista.com/statistics/1188549/covid-gaming-multiplayer/>



## Abbildungsverzeichnis





## Tabellenverzeichnis



## Listings

.



## A Anhang 1



## B Anhang 2