# Theoretisches Vorgehen

## Anforderungsanalyse

Wie in der Einführung (siehe 1.1) dargestellt, ist das Ziel die Schaffung einer multimodalen Telepräsenz zwischen einem menschlichen Benutzer und einem entfernten Roboter. Dabei soll eine Virtuelle Umgebung geschaffen werden, in welcher der Benutzer einerseits Steuerungseingaben an den Roboter senden und andererseits Bild- und Toninformationen vom entfernten Roboter erhält. Über die eine Virtual Reality Brille soll der Benutzer in die Umgebung eintauchen können und darüber interagieren können.

Die Virtuelle Umgebung soll folgende Features bereitstellen:

* Die Verbindung zu einem beliebigen Nao aufbauen
* Der Nao soll über den Virtuellen Raum angesteuert werden können, um Aktionen wie Hinsetzen, Aufstehen oder Laufen zu ermöglichen.
* Das Kamerasignal des Nao soll im Virtuellen Raum angezeigt werden. Dadurch kann der Benutzer durch die Umgebung navigieren, in der sich der Nao befindet.
* Das Tonsignal des Nao soll in den Virtuellen Raum übertragen werden
* Die Sprachausgabe auf dem Nao soll über das Eingeben von einfachen Sätzen im Virtuellen Raum ermöglicht werden.

Fehlt noch etwas bzw. ausführlicher machen? So gesehen deckungsgleich mit Kapitel 1.1, besser aufteilen oder entfernen?

## Geplantes Vorgehen

Die Anforderungen werden mit dem Team besprochen, um Unklarheiten frühzeitig zu beseitigen. Für die Entwicklung wird ein Projektplan entworfen und wichtige Meilensteile festgelegt.

Für den frühzeitigen Entwurf der Anwendung werden Designentwürfe (Mockups) entwickelt, um die unterschiedlichen Vorstellungen im Team besser visualisieren zu können. Diese können einfacher diskutiert werden und sollen Missverständnissen vorbeugen. Anhand der Mockups wird ein Modell für den Virtuellen Raum konzipiert, der die Steuerung für den Roboter enthält. (Telepräsenz?)

Die gewählte Architektur soll die Anwendung in Komponenten untergliedern, so dass diese unabhängig voneinander verändert und angepasst werden können. Gleichzeitig ermöglicht die Architektur eine bessere Übersicht der Anwendung. Die zum Planungszeitpunkt feststehenden Funktionen werden in Komponenten eingeteilt. Darüber hinaus werden Schnittstellen zwischen den Komponenten festgelegt, so dass diese durch Erweiterungen oder Rückschläge nachträglich verändert werden können.

Die Aufgaben aus dem Projektplan werden in kleinere Aufgaben zerlegt, so dass diese den Komponenten zugeordnet werden können.

Jeder Entwicklungsstand wird in ein Git-Repository hochgeladen, so dass der Projektfortschritt bei jedem Teammitglied immer auf dem aktuellen Stand ist. Dies soll die Parallelentwicklung fördern und dem Datenverlust vorbeugen.

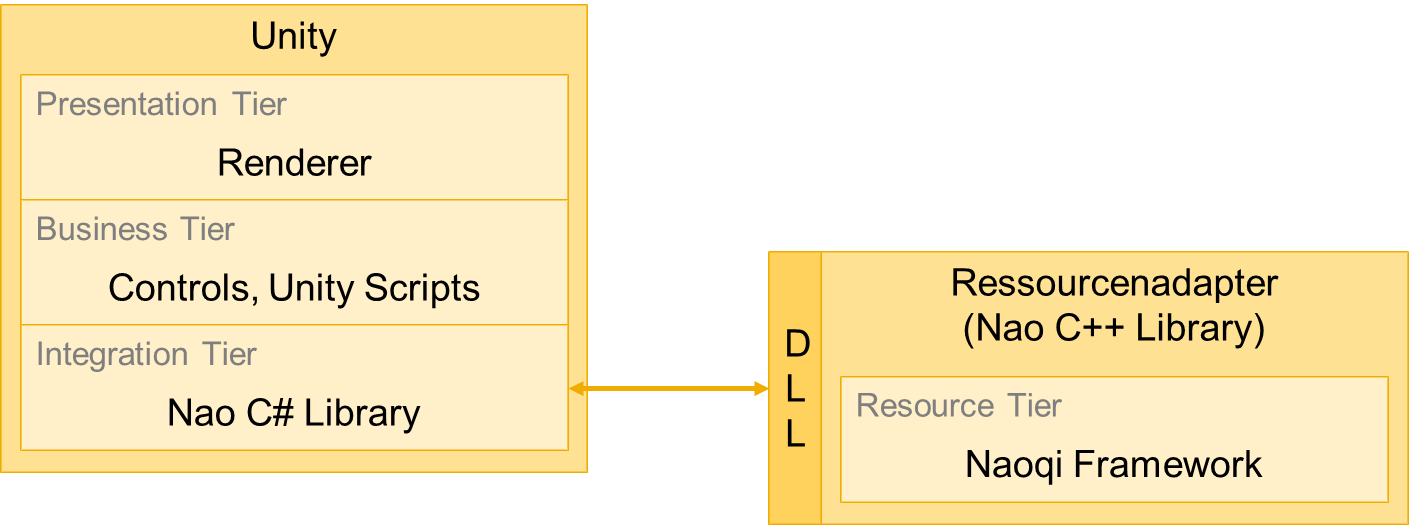
Durch regelmäßige Teammeetings soll über den aktuellen Entwicklungsfortschritt kommuniziert werden. Jedes Teammitglied sollte über die aktuellen Entwicklungen informiert werden. Durch die Treffen können auch Ratschläge und Verbesserungen durch das Team eingeholt werden.

Dokumentation?

# Konzeption

## Aufbau der Anwendung

Die Anwendung wurde nach dem 5-Schichtenmodell (siehe Kapitel x.x) konzipiert. In der Nachfolgenden Abbildung ist die Architektur zu sehen.



**Resource Tier**

**Die Ressourceschicht (Resource Tier) ist über einen Ressourcenadapter für das Naoqi Framework integriert. Dieser Adapter realisiert eine DLL-Schnittstelle, über welche Aktionen auf dem Nao gestartet werden können. Diese sind u.a. Laufen, Gehen oder Sprechen. Über den Adapter wird auch auf die Kamera und das Mikrofon des Nao zugegriffen.**

**Integration Tier**

**Die Integrationsschicht (Integration Tier) ist für die Kommunikation mit dem Ressourcenadapter zuständig. Dabei soll die Einbindung des Frameworks in Unity sichergestellt werden. Funktionen in dieser Schicht rufen Routinen in der DLL auf und fungieren als Erweiterung von Unity.**

**Business Tier**

**Die Businesslogik (Business Tier) übernimmt die Datenlogik und Steuerung in Unity. Speziell entwickelte Skripte in dieser Schicht ermöglichen die Interaktion des Benutzers mit der in der Integrationsschicht implementierten Erweiterung von Unity.**

**Presentation Tier**

**Die Präsentationsschicht (Presentation Tier) wird über Unity realisiert und enthält sowohl sämtliche Objekte zur Generierung des Virtuellen Raumes als auch Elemente für die Grafische Benutzeroberfläche (GUI).**

**Client Tier**

**Die Clientschicht (Client Tier) übernimmt die Darstellung der in der Präsentationschicht generierten Oberfläche und wird über die HTC-Vive Brille realisiert.**