



CHALMERS



SHARING AUGMENTED REALITY

Examensarbete inom Data- och Informationsteknik

Anders Huynh
Mehdi Adelzadeh

Institutionen för Data- och Informationsteknik
CHALMERS TEKNISKA HÖGSKOLA
GÖTEBORGS UNIVERSITET
Göteborg, Sverige 2017

EXAMENSARBETE

Delning av 3D-objekt mellan Microsoft HoloLens

Anders Huynh
Mehdi Adelzadeh

Institutionen för Data- och Informationsteknik
CHALMERS TEKNISKA HÖGSKOLA
GÖTEBORGS UNIVERSITET
Göteborg 2017

Delning av 3D-objekt mellan Microsoft HoloLens

Anders Huynh

Mehdi Adelzadeh

© Anders Huynh, Mehdi Adelzadeh, 2017

Examinator: Peter Lundin

Institutionen för Data- och Informationsteknik
Chalmers Tekniska Högskola / Göteborgs Universitet
412 96 Göteborg
Telefon: 031-772 1000

The Author grants to Chalmers University of Technology and University of Gothenburg the non-exclusive right to publish the Work electronically and in a non-commercial purpose make it accessible on the Internet.

The Author warrants that he/she is the author to the Work, and warrants that the Work does not contain text, pictures or other material that violates copyright law.

The Author shall, when transferring the rights of the Work to a third party (for example a publisher or a company), acknowledge the third party about this agreement. If the Author has signed a copyright agreement with a third party regarding the Work, the Author warrants hereby that he/she has obtained any necessary permission from this third party to let Chalmers University of Technology and University of Gothenburg store the Work electronically and make it accessible on the Internet.

Institutionen för Data- och Informationsteknik
Göteborg 2017

FÖRORD

Detta examensarbete utgör 15 högskolepoäng som har genomförts under dataingenjörsprogrammet som omfattar 180 högskolepoäng vid Chalmers Tekniska Högskola.

Vi vill tacka handledaren Sakib Sisteck för möjligheten att kunna utföra examensarbetet under hans handledning. Vi skulle även vilja tacka företaget Stena IT för möjligheten att kunna testa arbetets resultat på deras hårdvara.

SAMMANFATTNING

Den här studien är utförd på Chalmers tekniska högskola och syftar till att undersöka möjligheterna för användning av augmented reality inom ett utbildningssyfte. I dagsläget är det inte alltid lätt att förstå vad föreläsaren ritar på tavlan. Med hjälp av augmented reality kan visualisering av olika idéer underlätta förståelsen. Den skapade demoapplikationen gör det möjligt för användarna att se samt integrera med samma 3D-objekt i realtid. Därmed kan den bidra till bättre förståelse för studenterna. Applikationen är utvecklad för Microsoft HoloLens. Till följd av studiens omfattning har avgränsningar gjorts vad gäller skapande av hologram, vilka måste laddas in från en server. Säkerhetsfrågan har inte prioriterats i detta arbete vilket därmed är av intresse vid en vidareutveckling samt förbättring av applikationen.

Nyckelord: AR, augmented reality, Microsoft HoloLens, hologram, visualisering, undervisning

ABSTRACT

This study is conducted at Chalmers University of Technology and aims at exploring the possibilities for using augmented reality for an educational purpose. At the moment, it is not always easy to understand what the lecturer draws on the board. With the help of augmented reality, visualization of different ideas can facilitate understanding. The created demo application allows users to see and integrate with the same 3D object in real time. Thus, it can contribute to a better understanding of the students. The application is developed for Microsoft Hololens. Due to the extent of the study, delimitations have been made regarding the creation of a hologram, which must be loaded from a server. The security issue has not been prioritized in this work, which is therefore of interest in further development and improvement of the application.

Keywords: AR, augmented reality, Microsoft Hololens, hologram, visualization, education

Innehåll

FÖRORD	iv
SAMMANFATTNING.....	v
ABSTRACT.....	vi
1 INLEDNING	1
1.1 Bakgrund.....	1
1.2 Syfte	2
1.3 Mål2	
1.4 Avgränsningar.....	2
1.5 Frågeställningar.....	2
2 TEKNISK BAKGRUND.....	3
2.1 Augmented Reality	3
2.2 Microsoft HoloLens	3
2.2.1 HoloLens komponenter.....	4
2.2.2 Fördefinierade funktionaliteter	4
2.2.2.1 Gaze	4
2.2.2.2 Gester	5
2.2.2.3 Röstkommandon	5
3 Mjukvara och utvecklingsverktyg.....	7
3.1 Visual Studio.....	7
3.2 Unity®	7
3.2.1 GameObjekt	7
3.2.2 World Anchor	7
3.2.3 Scen.....	7
3.2.4 AssetBundle	7
3.3 Azure molnplattform.....	7
3.4 Microsoft HoloLens emulator.....	7
3.5 HoloTool-kit	8
3.5.1 Input	8
3.5.2 Sharing	9
3.5.2.1 Sharing Service	9
3.5.2.2 Sharing Manager	9
4 METOD.....	10
4.1 Möte med handledare.....	10
4.2 Arbetsmetod.....	10

4.3 Förstudie	10
4.4 Parprogrammering	10
4.5 Verktyg	10
4.6 Testning och verifikation	10
5 GENOMFÖRANDE.....	11
5.1 Förstudie	11
5.2 Utveckling av applikationen	11
5.2.1 Uppsättning av Unity®	12
5.2.2 Inladdning av objekt under körtid.....	12
5.2.2.1 Konstruktion av modell.....	12
5.2.2.2 Assetbundle.....	13
5.2.2.3 Bundle downloader	13
5.2.2.4 Positionering av modell	14
5.2.2.5 Interaktiv modell	14
5.2.3 Delning mellan flera enheter.....	14
5.2.3.1 Kommunikationslänk.....	15
5.2.3.2 Coordinator	15
5.2.3.3 Anchor manager	16
5.2.3.4 Remote head manager.....	16
5.2.3.5 Strömning av interaktioner	17
6 RESULTAT.....	18
6 DISKUSSION	20
7 SLUTSATS OCH VIDAREUTVECKLING	21
REFERENSER.....	22

1 INLEDNING

Rapporten redovisar för ett projekt som sträcker sig över 10 veckor och har genomförts på Chalmers Tekniska Högskola. I detta kapitel finns en översiktlig introduktion till projektet. En bakgrund till arbetet och varför detta har utförts. Även frågeställningar som ska besvaras under projektets gång.

1.1 Bakgrund

Föreläsning är en undervisningsmetod som i huvudsak består av en muntlig presentation med en föreläsare ämnad för en grupp studenter. Föreläsningarna utförs med hjälp av någon slags visualisering som bildspel, lösblad, en bild eller film. Visualisering kan även ske genom användning med en whiteboardtavla för att betona viktiga delar av föreläsningen.

Dagens föreläsningens metoder kan anses vara ensidiga. Föreläsaren dikterar vilken information studenterna tilldelas. Dessa har liten eller ingen möjlighet att ge sin egen personliga inmatning, eller protesterar mot den information som lämnas utan att behöva avbryta föreläsaren. De visualiseringar som används för att öka förståelsen kring idén kan upplevas som att figuren saknar ett djup. Föreläsningen kan även betraktas som passivt för studenterna. Inte vara aktivt inblandad i diskussionen angående vad som föreläsningen presenterar kan verka onödig för studenterna. Kort sagt, kan studenterna bli uttråkade av materialen som introduceras då de inte kan ha inflytande i diskussioner och på så sätt kan det även uppfattas som svårt.

För att ge en tydligare uppfattning angående visualisering av objekt kan man tänka sig att använda de nyligen introducerade visualiseringsverktyget Microsoft HoloLens. HoloLens är ett verktyg som gör det möjligt att realisera Augmented Reality(AR). Man kan beskriva AR som en kombination av verklighet och ett visuellt lager av data. Detta kallas även "Förstärkt verklighet". Med hjälp av AR kan visualisering ske i form av 3D-modeller som visualiseras i det verkliga synfältet och som kan kombineras tillsammans med tal. Denna möjlighet kan bidra till en mer interaktiv föreläsning. Tekniken kan i nuläget inte introduceras i samband med undervisning på grund av enhetens kostnad, ungefär 30 000 kr. Men troligtvis kommer kostnaden för detta verktyg att sjunka och ha ett pris i samma klass som dagens mobiltelefoner.

Teknik av denna sort kommer troligen att kunna ha många användningsområden bland annat som en förstärkning av beskrivningar och presentationer inom exempelvis undervisning, seminarier och guidade turer. Det existerar applikationer som använder sig av 3D-modeller för att öka förståelsen för användaren. Ett exempel är Human Anatomy Atlas. Denna applikation visualiserar den mänskliga kroppen, användaren kan zooma på en specifik del av kroppen för att få information[1]. Det finns även andra liknande applikationer som utnyttjar samma princip. Bland dessa applikationer upptäcktes en gemensam brist, att de består av hårdkodade modeller.

En avgörande faktor i samtliga av ovan nämnda applikationer är möjligheten att hantera gemensamma visualiserade 3-D modeller som en eller flera kan påverka på ett sätt som alla kan se, dvs att man kan dela en modell inom en grupp.

Uppdraget att undersöka möjligheten med AR i presentationer och undervisning är tilldelad av Sakib Sisteck från Chalmers Tekniska Högskola och arbetet går ut på att skapa en demonstrationsapplikation som tar del av denna nya teknologi inom ett utbildningssyfte.

1.2 Syfte

Rapporten avser att undersöka möjligheterna att använda AR med stöd av Microsoft HoloLens i samband med undervisning, seminarier, presentationer mm.

1.3 Mål

Målet med projektet är att utveckla en demonstrationapplikation till Microsoft HoloLens som ska kunna hämta och ladda in 3D-objekt från en server under körtid samt dela objektet mellan olika HoloLens enheter i realtid.

1.4 Avgränsningar

Under projektets gång måste vi göra några klara avgränsningar så att vi håller oss inom ramen för projektet. Dessa avgränsningar är:

- Säkerheten för applikationen kommer att bortses.
- Applikationen ska inte tillföra egna hologram.
- Delningen ska inte ske globalt
- Användargränssnitt prioriteras ej

1.5 Frågeställningar

- Hur kan man skapa delning av visualiserade objekt för flera HoloLens enheter?
- Vilka möjligheter till interaktion avseende påverkan på modellerna finns mellan flera enheter?

2 TEKNISK BAKGRUND

I detta kapitel redogörs nödvändig förståelse för de tekniska system och tjänster som kommer att användas för att fullfölja projektet.

2.1 Augmented Reality

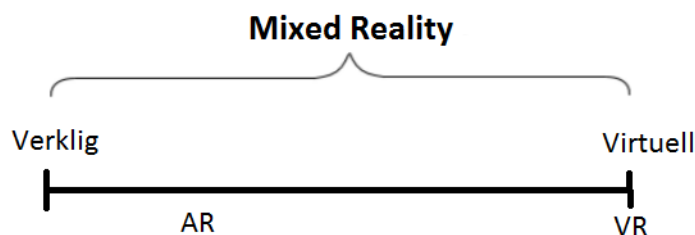
Augmented Reality är en förstärkt verklighetsbild som integrerar virtuella objekt till den riktiga världen. Till skillnad från Virtual Reality (VR), som skapar en fullständig konstgjord miljö, kombinerar Augmented Reality den verkliga världen med exempelvis hologram.

Man kan hitta AR inom områden såsom mobilapplikationer och datorspel. Det finns olika användningsområden där Augmented Reality utnyttjas som ett medium idag, till exempel mobila enheter. Då spelar kameran in omgivningen och med hjälp av ett Software Development Kit (SDK) visas ett virtuellt objekt upp på bildskärmen. Denna teknologi har implementerats för användning av Augmented Reality. Ett exempel på detta är Google glasses.

För att klassificera någonting som Augmented Reality ska tre kriterier vara uppfyllda. Först och främst ska sinnesintrycket kom från verkligheten och det virtuella på samma gång. Sedan måste båda världarna överensstämma med varandra geometriskt. Dessutom måste allt detta ske i realtid[2].

2.2 Microsoft HoloLens

Microsoft HoloLens är ett par Mixed Reality smarta glasögon som är utvecklad och tillverkad av Microsoft. Mixed Reality är ett samlingsnamn för en skala mellan verklig miljö och virtuell miljö, som illustreras enligt figur 2.1 nedan. Här ingår AR och VR. HoloLens ligger under kategorien AR, även detta illustreras i figuren. Antaganden som kan göras från figuren är: desto längre till vänster på skalan desto mer av verkligheten finns tillgänglig och tvärtom.



Figur 2.1: Bilden illustrerar en skala mellan verklig och virtuell miljö där AR kan hittas.

Microsoft HoloLens är en holografisk dator inbyggd inuti ett headset. Headsetet är en självständig enhet som inte är beroende av uppkoppling till en dator. Denna enhet tillåter HoloLens-användaren att kunna se, höra samt interagera med hologram. HoloLens använder sig av en högupplöst display samt en teknik kallad spatial sound för att skapa en interaktiv

upplevelse för användaren[3]. Spatial sound är ett sätt för användare att få en uppfattning av omgivningen genom användning av ljud, när objekt inte är inom synfält. Detta möjliggörs genom HoloLens inbyggda högtalare som kan generera ljud i 3D format vilket kan göra det möjligt att höra ljud från olika riktningar[4].

HoloLens har även en halvgenomskinlig holograms display som genererar multidimensionella hologram. Detta innebär att hologrammen endast kan visas upp för HoloLens användare. Detta betyder att endast användaren kan se dessa datorgenererade hologram samtidigt som den kan se omvärlden kring en.

Ett intressant projekt skapat av Microsoft som kallas "HoloPortation" tillåter HoloLens användare att 3D-teleporteras in i en annan användares synfält. Men i nuläget kräver denna process användning av flera kameror, men detta visar potentialen som den fysiska komponenten HoloLens har i framtiden. [5]

2.2.1 HoloLens komponenter

HoloLens hårdvaruspecifikation är väldigt komplext därför nämns det enbart utvalda komponenter nedan.[6]

- Genomskinliga linser som ger användaren möjligheten att projicera hologram.
- 4 kameror som används till att registrera användarens omgivning.
- 1 kamera som mäter och registrerar djupet i ett hologram.
- 4 mikrofoner som registrerar användarens röstkommandon.
- Högtalare
- Wi-Fi för tillgång till internet.



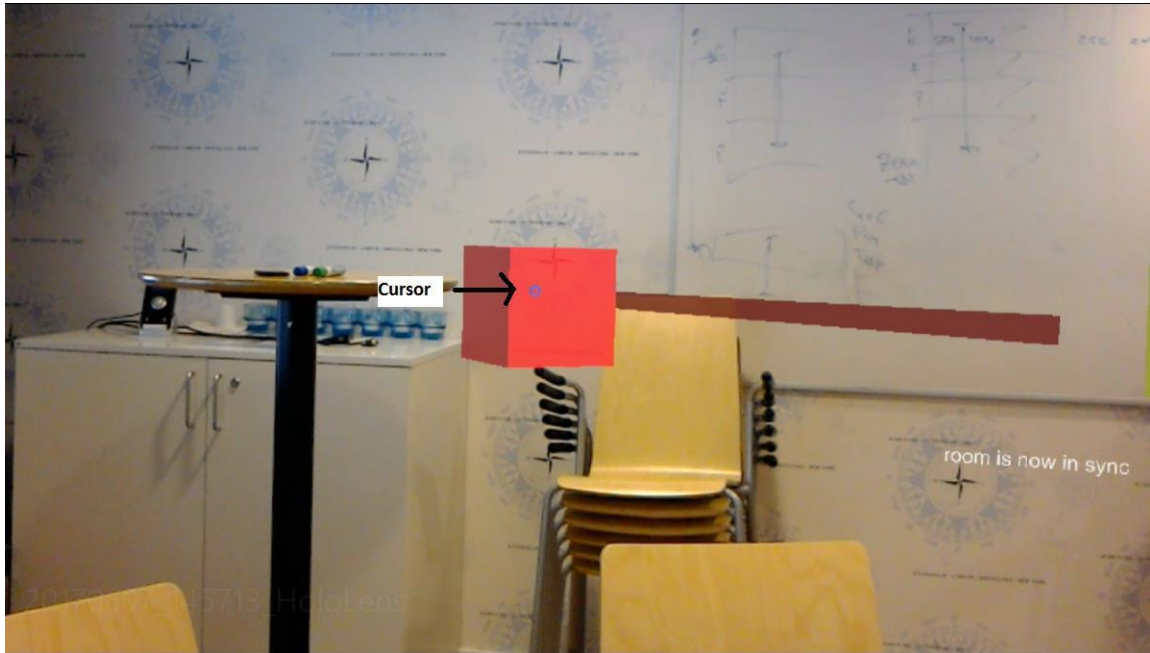
Figur 2.2: *Microsoft HoloLens*

2.2.2 Fördefinierade funktionaliteter

För att erbjuda HoloLens-användare en verklig upplevelse vad gäller styrning, innehåller enheten fördefinierade funktionaliteter som beskrivs nedan.

2.2.2.1 Gaze

Gaze är den fördefinierade funktionaliteten i enheten som registrerar användarens huvudrörelse för att bestämma vart användaren tittar. [7] I den verkliga världen är det vanligt att man tittar på det objekt som man vill interagera med och detta motsvarar gaze i HoloLens. Gaze används även för att peka mot objektet som användaren vill interagera med och interaktionen utförs med hjälp av handgester eller röstkommandon. För att underlätta för användaren vart gaze pekar på införs en markör.



Figur 2.3: Den blåa cirkeln är en markör som representerar gaze

2.2.2.2 Gester

Inmatning av gester i form av en handgest möjliggör interaktion mellan användare och hologram. [8] Sådana gester identifieras av HoloLens genom att användaren utför gesten inom enhetens synfält. För att exempelvis flytta på hologram kombineras handgester tillsammans med “gaze”. “Select” är en handgest inom användning av HoloLens som är det vanligaste sättet att samspela med hologram. Denna gest utförs genom att användaren utför en nyp-gest inom enhetens synfält.



Figur 2.4: Denna nyp-gest motsvarar Select i HoloLens

2.2.2.3 Röstkommandon

Ett annat sätt att göra en inmatning är genom användning av rösten. Handgesten Select kan

exempelvis ersättas med ett röstkommando. Utförandet sker genom att användaren placerar pekaren på ett hologram för att sedan uttala kommandot Select, därefter registreras detta kommando och användaren meddelas genom att enheten spelar upp ett ljud. [9]

3 Mjukvara och utvecklingsverktyg

I detta kapitel kommer det tas upp om den mjukvara och dess utvecklingsmiljöer. Sedan även vilka verktyg som använts.

3.1 Visual Studio

Visual Studio är en utvecklingsmiljö inom programmering utvecklad av Microsoft. Detta är ett väldigt flexibelt verktyg som kan användas för att skapa applikationer för mobila enheter till exempel Windows, Android och IOS. [10] Utöver mobila enheter kan användaren även utveckla datorprogram. Programmeringsspråket som används mest inom detta verktyg är C# men har även stöd för andra språk såsom C, C++, Python mm. För detta arbete kommer programmeringsspråket C# att användas.

3.2 Unity®

Unity® är en spelmotor som kan användas för att utveckla både 2D- och 3D spel. Denna spelmotor är en av de populäraste i sin marknad och används för att utveckla spelscener för mobila enheter, persondatorer, webbsidor samt konsoler. [11]

3.2.1 GameObject

Ett "GameObject" är en kontainer för lagring av objekt. En karaktär eller något annat kan skapas genom att bitar adderas till spelobjektet, varje bit kallas en "component". Beroende på vad förslags objekt som ska skapas adderas olika kombinationer av komponenter till ett spelobjekt[12].

3.2.2 World Anchor

Detta är en komponent som representerar en länk mellan en exakt punkt i den fysiska världen och föräldern spelobjektet av "World Anchor". Ett spelobjekt med en "World Anchor" förblir låst på samma plats i den riktiga världen. Denna kan användas för till exempel fixera ett hologram till en vägg. World Anchor används när det finns ett spelobjekt eller ett träd av spelobjekt som ska fästas på en fysisk plats. [13]

3.2.3 Scen

Applikationer och spel utvecklade i Unity® är uppbyggda av scener där utvecklaren binder objekt och skript till scenen. Dessa scener kan till exempel vara menyn för applikationen eller spelets nivåer. [14]

3.2.4 AssetBundle

Assetbundles är en samling av filer som skapas i Unity® och samlingen kan till exempel innehålla modeller, material eller scener. I detta arbete består samlingen av 3D-modeller. Användning av Assetbundle ger applikationen möjligheten att kunna strömma objekt från nätet under kortid. [15]

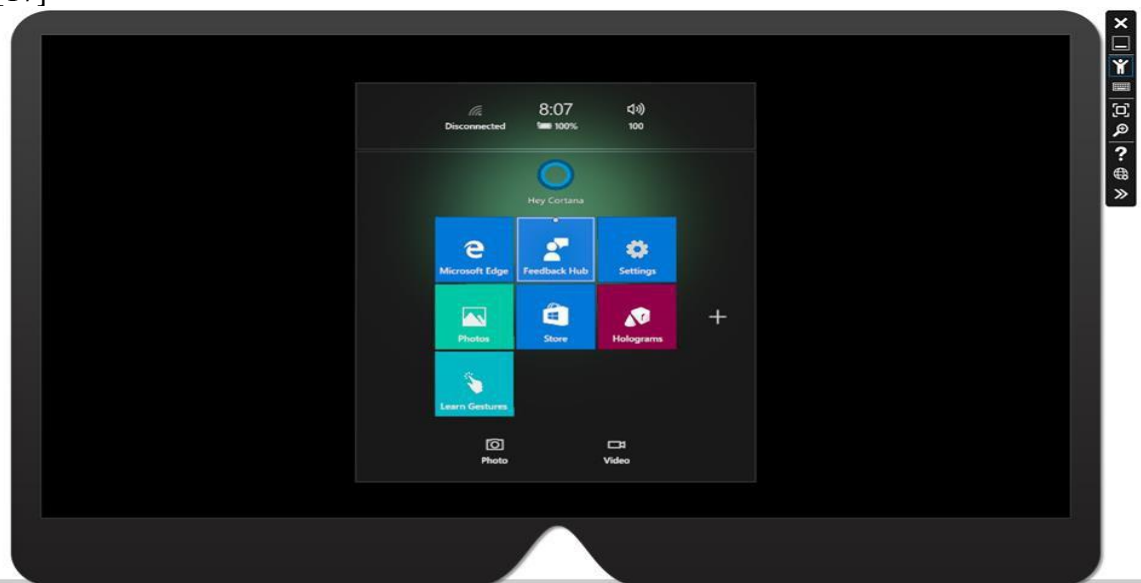
3.3 Azure molnplattform

Azure är en molnplattform som har ett stort utbud av stöd för flera olika ramverk, operativsystem, databaser, programmeringsspråk mm. Under detta projekt används Azure som en plats för lagring av objekt, detta öppnar möjligheten för applikationen att kunna ladda in valda objekt. [16]

3.4 Microsoft HoloLens emulator

HoloLens emulator är ett verktyg utvecklad av Microsoft. Emulatorn ger möjligheten för

utvecklare att testa holografiska applikationer på en PC utan att behöva den fysiska enheten. Den fysiska enhetens sensorer som läser av människans rörelser och omgivning är ersatt av simulerade signaler som består av inmatning från tangentbord, mus eller Xbox handkontroll. [17]



Figur 3.1: *HoloLens Emulator*

3.5 HoloTool-kit

HoloTool-kit är ett öppet bibliotek utvecklad av Microsoft och i detta bibliotek innehåller det färdiga skripten för att underlätta utvecklingen av applikationer för Microsoft HoloLens. [18] Skripterna i detta bibliotek är organiserade efter rubriker. Delar som används i detta arbete kommer att beskrivas kortfattat.

Namn	Senast ändrad	Typ	Storlek
Build	2017-05-08 02:54	Filmapp	
CrossPlatform	2017-05-08 02:54	Filmapp	
Input	2017-05-08 02:54	Filmapp	
Sharing	2017-05-08 02:54	Filmapp	
SpatialMapping	2017-05-08 02:54	Filmapp	
SpatialSound	2017-05-08 02:54	Filmapp	
SpatialUnderstanding	2017-05-08 02:54	Filmapp	
UI	2017-05-08 02:54	Filmapp	
Utilities	2017-05-08 02:54	Filmapp	

Figur 3.2: *Innehållet i HoloTool-kit*

3.5.1 Input

Input är en av bibliotekets rubriker och en del av innehållet har implementerats i detta arbete. Under denna rubrik innehåller det skripten som ansvarar applikationens inmatning, exempelvis gaze och markör.

3.5.2 Sharing

Innehållet under rubriken Sharing ger applikationen möjligheten att kunna kommunicera med andra enheter som kör samma applikation. Kommunikationen sker i realtid oberoende av vart användaren befinner sig, dvs i samma rum eller på distans.

3.5.2.1 Sharing Service

Sharing service agerar en lokal punkt som enheter kan koppla sig till för att kommunicera med varandra. Denna server är en del av Sharing biblioteket och servern exekveras via kommandotolken i Windows och använder sig av datorns ip adress som enheter kan koppla sig till. All kommunikation sker via UDP och servern har ansvaret att försäkra leverans av meddelanden.

3.5.2.2 Sharing Manager

Sharing manager är ett hjälpmedel som visar information om servern som är igång samt vilka enheter som är kopplade mot servern. Informationen visas i ett användargränssnitt och detta ger en överblickande bild över serverns session. Sharing manager kan även agera klient för att ansluta till servern. På så sätt kan information om vad som skickas mellan olika enheter läsas av.

4 METOD

Här beskrivs planeringen samt arbetsmetoden som ska följas under projektets gång.

4.1 Möte med handledare

Till en början kommer möten med handledare att ske veckovis för att kunna diskutera idéer och fastställa en projektplan angående om vad som skulle kunna vara möjligt att skapa.

4.2 Arbetsmetod

Då det kommer att vara svårt att göra en tidsuppskattning av de olika delar som krävs för att utföra arbetet valdes inte Scrum som arbetsmetod. Arbetet kommer istället att ske agilt som är baserad på Scrum, genom att arbeta agilt kommer kontinuerliga möten och diskussioner hållas för att fatta nya projektbeslut.

4.3 Förstudie

Microsoft HoloLens är relativt ny i arbetsmarknaden och har begränsade resurser. Detta innebär att en stor del av projektets tid kommer att användas till instudering om vad som är möjligt att utveckla samt om det finns liknande applikationer som vi kan ta del av. Förstudier görs med hjälp av Googles sökmotor samt Microsoft egna dokumentation angående applikationsutveckling för Microsoft HoloLens.

4.4 Parprogrammering

Utvecklingen av projektet kommer att utföras genom parprogrammering. Tanken bakom parprogrammering är att man turas om att skriva kod. Under tiden den ena parten skriver kod skall den andra observera koden. Utöver kommer den som observerar även att bidra med idéer och förslag. Genom att använda denna metod håller paret en kontinuerlig kommunikation vilket kan leda till att programmerarnas kunskap delas och hjälper varandra under utvecklingen.

4.5 Verktyg

Utvecklingen av applikationen kommer att ske i utvecklingsmiljön Microsoft Visual Studio 2017 med programmeringsspråket, C#. Visual Studio har stöd för exekvering av Hololens-emulator och den fysiska komponenten. För att utveckla scener och designa applikationens utseende kommer Unity® att användas. Även skapandet av 3d-objekt kommer att ske med hjälp av Unity®.

4.6 Testning och verifikation

På grund av begränsad tillgänglighet av den fysiska enheten kommer större delar av testningen av applikationen att ske på HoloLens emulator. Testning kommer även att ske under senare delen av utvecklingsprocessen när den fysiska enheten blir tillgänglig. Detta berodde på att Chalmers inte kunde förse gruppen med hårdvaran. Gruppen diskuterade detta problem med andra grupper som också jobbade med en HoloLens projekt. Dessa kunde förse oss med hårdvaran under en begränsad tid under projektets slutstadium.

5 GENOMFÖRANDE

Detta kapitel handlar om hur man gick tillväga för att skapa en Microsoft HoloLens applikation. Även vilka inställningar i Unity® som krävs för att applikation skall fungera. Här beskrivs även mer utförligt hur applikationens funktioner och modeller skapades.

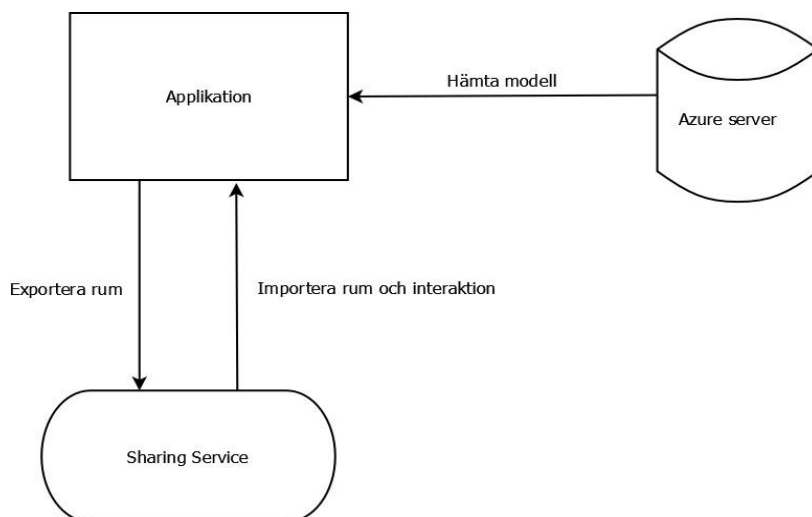
5.1 Förstudie

Innan applikationen skapades började gruppen att göra en förstudie angående hur man utvecklar en applikation till HoloLens samt vad en sådan applikation består av. Microsoft har tydlig information i form av en guide som en utvecklare kan följa för att skapa en enkel applikation till HoloLens. En del av förstudien användes till att studera andra applikationer för att finna idéer kring vad som var möjligt att utveckla samt få en bild av vilka brister applikationerna har. Bland de applikationer som studerades fann vi en gemensam brist, att dessa applikationer består av hårdkodade modeller, vilket var funktionaliteten som skulle undvikas. Även applikationer innehållande funktionaliteten att kunna dela objekt mellan flera enheter var svårt att hitta och detta berodde på att utvecklingen av delning inte är tillräckligt välutvecklad än.

Förstudien ledde till en ökad förståelse kring utveckling av AR applikationer. En sådan applikation består av 3D-modeller som styrs av tillhörande skript, dessa skript bestämmer över modellernas beteende samt vad som skall ske vid olika interaktioner. Förstudien gav även idéer och tankar kring hur de nämnda bristerna kan undvikas. Även en fördjupad kunskap inom C# samt hantering av verktyget Unity® skaffades.

5.2 Utveckling av applikationen

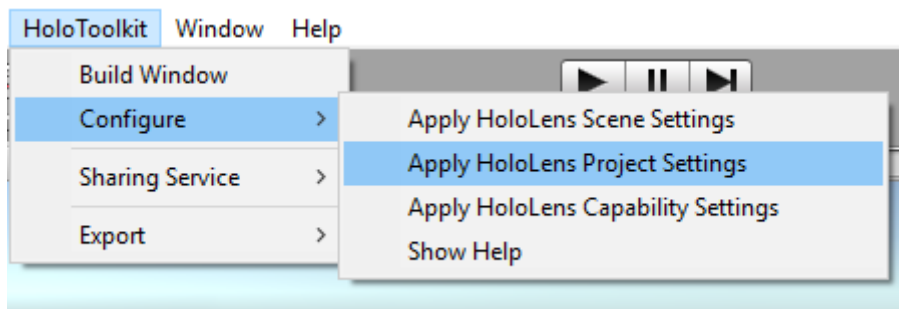
Utvecklingen av applikationen beskrivs i detta delkapitel. Förstudien resulterade i att en primär version av en systemarkitektur ritades upp. Systemarkitektur består av olika byggstenar för detta arbete samt beskriver kommunikationen sinsemellan. Tanken var att applikationen ska ladda in modeller från en extern server samt att delningen mellan olika enheter sker via en server kallad Sharing Service. Detta resulterade i att applikationen delades upp i två mindre rubriker kallad “Inladdning av objekt” och “Delning mellan flera enheter”.



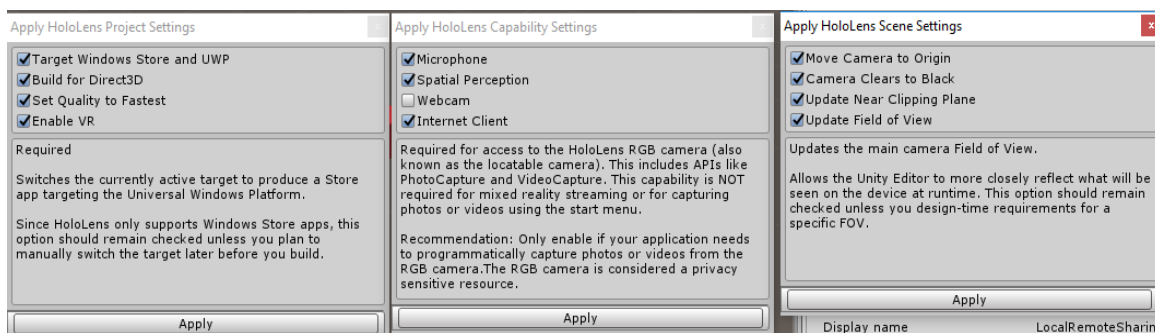
Figur 5.1: Primär version av systemarkitekturen för detta arbete

5.2.1 Uppsättning av Unity®

För att utveckla en applikation till Microsoft HoloLens med hjälp av Unity® krävs det att rätt inställningar är applicerade. HoloToolkit som är ett verktyg av Microsoft användes för att underlätta appliceringen av dessa inställningar, verktyget ger tillgång till en fördefinierad checklista där utvecklaren kan bocka i de inställningar som ska appliceras. För detta arbete applicerades inte alla inställningar som detta verktyg erbjuder utan enbart de användbara för detta arbete valdes.



Figur 5.2: Sökväg till Unity inställningar



Figur 5.3: Unity® inställningar för detta arbete

Utöver inställningarna som nämnts ovan behövs ytterligare två inställningar appliceras för att applikationen ska vara körbar. Under File -> Build Settings och rubriken Debugging ska Unity® C# Projects vara markerad. Sedan ska Edit-> Project Settings -> player och i underrubriken capabilities ska PrivateNetworClientServer vara markerad

5.2.2 Inladdning av objekt under körtid

I detta delkapitel beskrivs konstruktionen av modellen som applikationen använder sig av samt hur inladdning av objekt under körtid har genomförts.

5.2.2.1 Konstruktion av modell

3D-modellen som används av applikationen är väldigt enkel då designen inte spelade en viktig roll för applikationens slutändliga mål. Konstruktionen av modellen gjordes i Unity®. Fyra separata objekt i form av kuber skapades med olika färger för att enkelt kunna se skillnad på objekten. Dessa skalades ner till 0.25 meter och distansen mellan objekten var 1 meter vilket formade en kvadratisk figur. Vi använde oss av Unity® box-collider för att binda samman dessa kuber till ett enda objekt. Modellen är även inställd att placeras 1 meter framför HoloLens-användaren.



Figur 5.4: Konstruktionen av modellen i Unity®

5.2.2.2 Assetbundle

För att tackla bristen angående hårdkodade objekt implementerades AssetBundle som är en teknik som stöttes på under förstudien. Tekniken bygger på att konvertera modeller till en enda stor filsamling som lagras lokalt i datorn. Dessa filer används av applikationen för att kunna ladda in den valda modellen under körtid. Implementationen av Assetbundle kunde ske på två olika sätt. Det ena sättet var att med hjälp av Unity® dokumentation, bygga en egen konverterare. Det andra sättet var att använda Assetbundle Manager som är ett verktyg som finns på Unity® Store. För att underlätta arbetet valdes det färdiga verktyget. För att kunna använda verktyget lades det i Unity® projektets Assets mapp. Assetbundle Manager var en utdaterad version då den var skapad för Unity® 4.0.0. och detta arbete är utvecklad i Unity® 5.6. Detta orsakade kompilieringsfel när Assetbundle testades. Orsaken till detta berodde på att verktyget inte kunde identifiera applikationens build-target. Problemet löstes ganska enkelt genom att manipulera Utility skriptet. Nya byggmål lades till, i detta fall Windows Store applikationer vilket motsvarar applikationer för HoloLens.

```

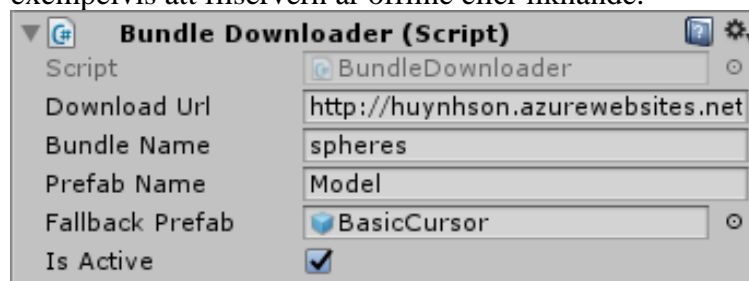
case RuntimePlatform.WebGLPlayer:
    return "WebGL";
case RuntimePlatform.OSXWebPlayer:
case RuntimePlatform.WindowsWebPlayer:
    return "WebPlayer";
case RuntimePlatform.WindowsPlayer:
    return "Windows";
case RuntimePlatform.OSXPlayer:
    return "OSX";
case RuntimePlatform.WSAPlayerARM:
case RuntimePlatform.WSAPlayerX64:
case RuntimePlatform.WSAPlayerX86:
    return "Windows";
  
```

Figur 5.5: Justering i Utility skript, den inramade rektangeln är koden som implementerades

5.2.2.3 Bundle downloader

En webbserver skapades med hjälp av Azure molnplattform vilket blev en global lagringsplats för Assetbundle filerna. Till följd av detta undviker applikationen hårdkodade modeller och filer. För att hämta modeller från webbservern implementerades ett skript, kallad Bundle Downloader. Bundle downloader ansvarar för nedladdning av en utvald modell i Assetbundle

som är lagrad på webbservern. Skriptet kräver att användaren definierar nedladdningslänk till servern, samlingens namn samt modellens namn. En fallback prefab implementerades, detta är en förutbestämd modell som skall laddas in när applikationen misslyckas med nedladdning av modell från servern. Detta implementerades för att inte orsaka hinder, exempelvis att filservern är offline eller liknande.



Figur 5.6: Definiering av nedladdningslänk, bundle-namn samt namnet på modellen

5.2.2.4 Positionering av modell

Ett problem uppstod vid placering av modellen. Problemet var att modellen placerades olika varje gång applikationen startades. Exempelvis en modell som ska ha koordinaterna (3,3,3) kunde placeras ut annorlunda i den verkliga världen. Detta berodde på att koordinatsystemet ompositioneras efter användaren då koordinaterna är relativa till den specifika användaren. Detta löstes genom att implementera World Anchor.

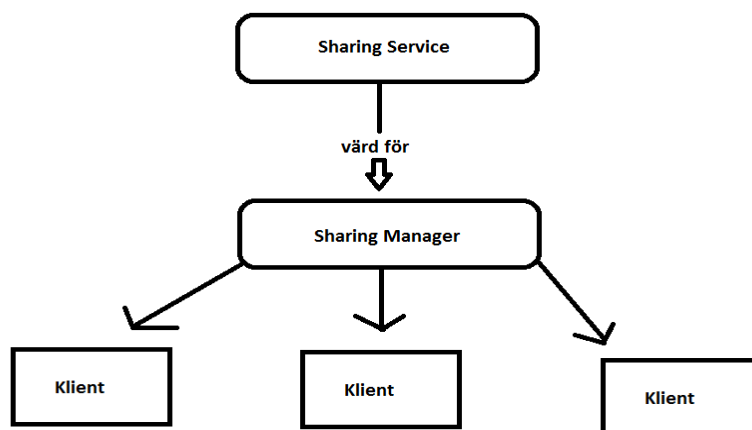
5.2.2.5 Interaktiv modell

Efter skapandet av modellen behövdes funktionaliteten att kunna interagera med modellen ~~att~~ implementeras. Ett skript kallad UserMovable implementerades och det första ~~som~~ prioriterades var hur förflyttning av modellen skulle ske. Tanken var att användaren placerar sin gaze på modellen och med hjälp av nyp-gesten Select, kunna greppa tag i modellen för att kunna rotera samt förflytta modellen. För att släppa eller placera modellen utförs samma handgest. Att interagera modeller på ett sådant sätt kunde orsaka problem då applikationen inte alltid fångade upp användarens handgest. Lösningen till detta var att med hjälp av HoloToolkit, implementera Input Manager som används för att ta hand om all inmatning från användaren. Efter att ha lyckats med implementationen angående förflyttning av modell, implementerades ett röstkommando "lock" vars ~~i~~ uppgift är att signalera applikationen att modellens position är låst och kan skickas vidare. Valet att implementera detta redan nu gjordes med hänsyn till delningen som kommer att implementeras.

5.2.3 Delning mellan flera enheter

Förstudien resulterade i grundläggande kunskap inom delning samt gav en bild av hur delning mellan flera enheter kunde implementeras. Ena sättet var att implementera egna skript där enheterna sköter all kommunikation sinsemellan utan någon mellanhand. Det vill säga att enheter agerar både server och klient och på så sätt kunna strömma data till enheter som är kopplad till den enheten som agerar server. Genom att implementera delning på detta sätt ger det utvecklaren möjligheten att bestämma hur och vad för data som skall skickas mellan enheterna. Det andra sättet var att implementera delning med hjälp av HoloToolkit-verktygets underkatalog Sharing som är utvecklad av Microsoft. En viktig sak att notera vid användning av verktyget är att 3D objekten inte strömmas över nätverket utan renderas av varje enhet och

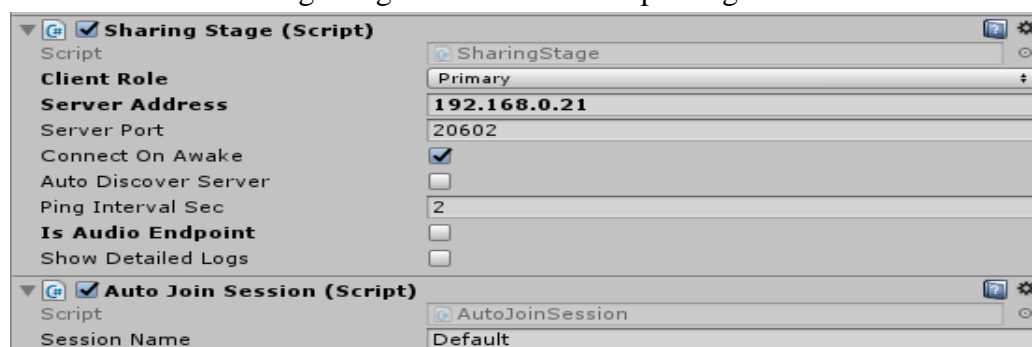
endast interaktionen strömmas. Med andra ord ändrar objektets beteende och positionering efter interaktionen som strömmas. Verktøget innehåller en server kallad Sharing Service, ett hjälpmedel kallad Sharing Manager och färdigskrivna skriptet inom delning. Detta verktyg är inte fullt utvecklad men är lanserad av Microsoft för att underlätta arbetet för en utvecklare. För enkelhetens skull valdes implementering av delning med hjälp av HoloToolkit-verktyget, valet är även baserad på anpassning för tidsramen för detta arbete.



Figur 5.7: En överblick på hur Sharing Service kommunicerar med klienter

5.2.3.1 Kommunikationslänk

En kommunikationslänk mellan klienten och Sharing Service skapas med hjälp av Sharing Stage skriptet som är en del av HoloToolkit-verktyget där den viktigaste parametern är Server Address. För att undvika extra belastning för applikationen implementerades Auto Join Session. Detta skript har i uppgift att automatiskt koppla upp applikationen mot Sharing Service vid start av applikationen. En viktig sak att notera, ett fel kan uppstå när användaren anger serveradressen då Sharing Service kan bestå av flera IP-adresser. Detta kan bero på att datorn som kör Sharing Service har emulatorer eller virtuella maskiner installerade med olika IP-adresser. Det är viktigt att kontrollera, vilken av dessa adresser som är datorns egna adress. En sådan kontroll kan göras genom kommandot ipconfig i kommandotolken.



Figur 5.8: Här sätts Sharing Service IP-adress

5.2.3.2 Coordinator

Efter att ha implementerat delningsdelen skapades ett skript kallad Coordinator. Skriptet hade i uppgift att kontrollera att en kommunikationslänk hade skapats mellan applikationen och

servern. Det var svårt att kontrollera om en anslutning hade gjorts då applikationen inte hade någon form av status text. Detta ledde till att en Text Mesh implementeras som huvudsakligen visar upp text, exempelvis olika tillstånd eller liknande. Efter att ha lyckats att koppla applikationen mot Sharing Service, expanderades Coordinator ytterligare. Tanken bakom expansionen var att skriptet även skulle ha hand om applikations flödet i form av tillstånd:

1. Väntar på att en kommunikationslänk är skapad
2. Väntar på att allt är färdigladdat
3. Kontrollerar om det finns ett rum i Sharing Server annars skapas ett rum
4. Om det finns ett rum, går applikationen med i rummet och laddar ner dess world anchor och ändrar modellens position utifrån det som laddas ner från servern
5. Om det inte finns ett rum, skapas en world anchor utifrån modellens position och sedan exporterar applikationen detta till Sharing Service

Efter att ha expanderat Coordinator enligt ovan uppstod det ett problem, vid det andra tillståndet användes RoomManager's GetRoomCount funktion för att kontrollera om det fanns ett rum på servern. Detta kunde ibland returnera 0 även om det fanns ett rum på Sharing Service. Lösningen till detta var att skapa en fördröjning i detta tillstånd för att försäkra att applikationen faktiskt är färdigladdad.

För att kunna binda samman delning och inladdning av objekt från en extern webbserver ändrades Coordinator, två extra tillstånd lades till:

1. Skapa en kommunikationslänk mellan applikationen och Sharing Service vid start av applikationen
2. Ladda in objekt från webbservern
3. Väntar på att en kommunikationslänk är skapad
4. Väntar på att applikationen är färdigladdad
5. Kontrollera om det finns ett rum i Sharing Server, annars skapa ett rum
6. Om det finns ett rum går applikationen med i rummet och laddar ner dess world anchor och ändrar modellens position utifrån det som laddas ner från Sharing Service
7. Om det inte finns ett rum, skapas en world anchor utifrån modellens position och sedan exporterar applikationen detta till Sharing Service

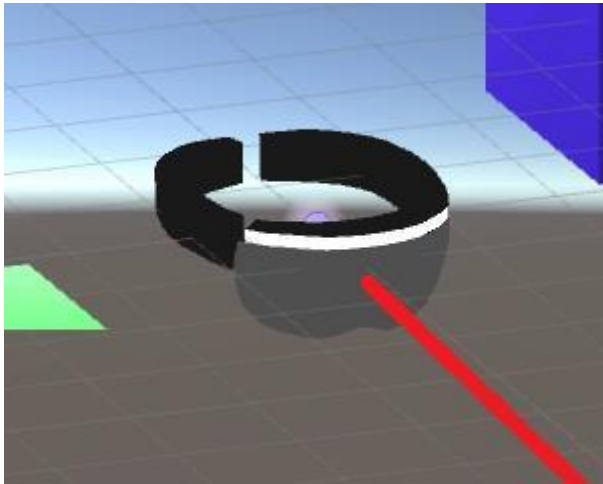
5.2.3.3 Anchor manager

ImportExportAnchorManager är en del av Sharing biblioteket som ansvarar för importering och exportering av interaktioner på en modell. För att kunna strukturera upp användningen av detta delades skriptet upp i två nya skript, ImportAnchorManager och ExportAnchorManager. Namnen indikerar att ena skripten tar hand om importering och den andra exportering. En del av robustheten förlorades från den original skripten till följd av förenkling. Detta resulterade att irrelevanta funktioner filtrerades bort och funktionaliteten blev mer anpassad för detta arbete. Dessa skript antar att en koppling till sharing service redan har skett och Room manager har använts för att montera ett rum.

5.2.3.4 Remote head manager

För att kunna identifiera positionen på andra enheter som är anslutna till samma rum används CustomMessage och RemoteHeadManager. CustomMessage ansvarar för delning av enheternas huvudposition sinsemellan via sharing service. RemoteHeadManager ansvarar för visualisering av dessa enheter och huvudpositioner presenteras i form av hololens modell. För att kunna se vart andra enheter tittar justerades skriptet. Gazen för enheterna motsvaras i form

av en linje som pekar mot vart enheterna tittar mot. Remote head implementerades huvudsakligen för att kunna bekräfta för hololens användarna att andra användare har kopplat sig till rummet.



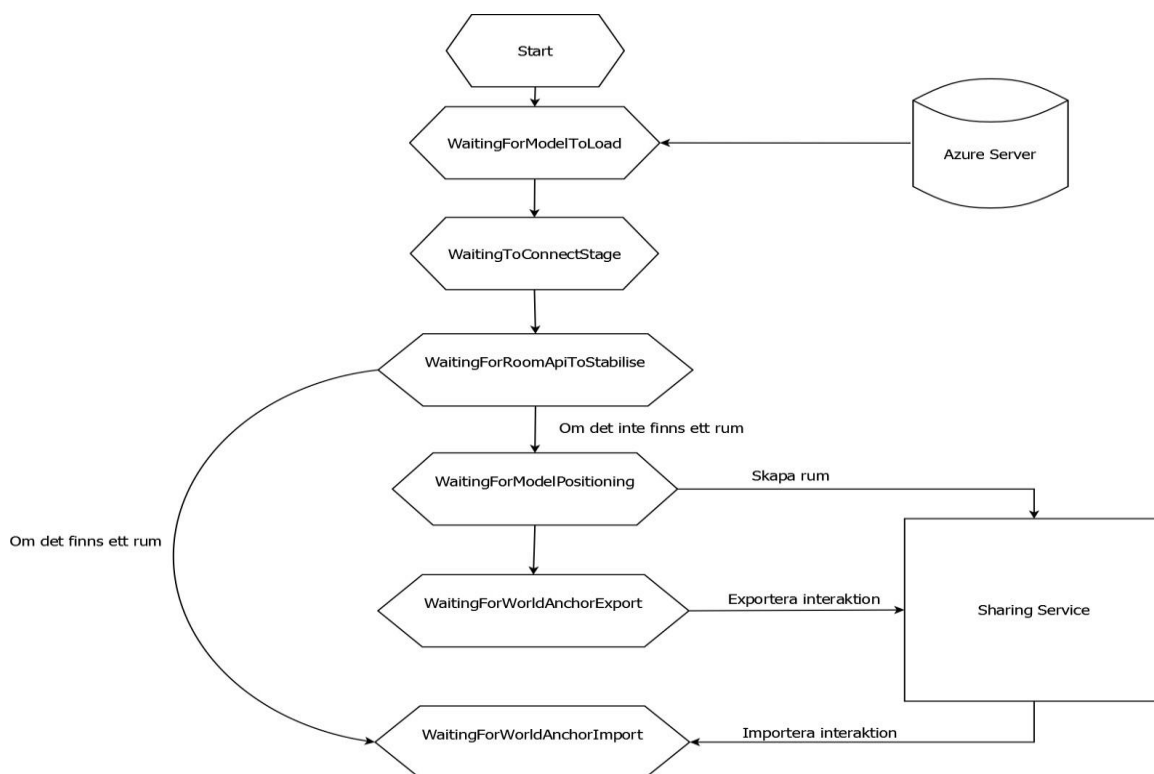
Figur 5.9: HoloLens modell tillsammans med en gaze

5.2.3.5 Strömning av interaktioner

I det nuvarande stadiet av utvecklingen tillåter applikationen användaren att positionera ut modellen i ett rum, sedan exportera modellens position med hjälp av röstkommandot “lock”. Detta tillåter andra HoloLens-användare att koppla sig till samma rum för att sedan kunna se samma objekt. Detta ledde till att modellerna inte var interaktiva efter användning av röstkommandot “lock”. Detta ansågs vara en begränsning, att inte kunna interagera med modellen. För att tackla detta problem användes en teknik som stöttes på under förstudien, tekniken bygger på att applikationen dynamiskt laddar/raderar world anchor från applikationen och på sätt kan applikationen strömma modellens position. Denna teknik applicerades till “Coordinator” skriptet som synkroniserar/raderar world anchor från “UserMoveable”. Det här öppnade upp möjligheten för applikationen att inte låsa modellens position vid användning av “lock”. För att kunna synkronisera modellens rörelse i realtid behövdes en komponent som övervakar rörelser, och sedan strömma detta över Sharing Service. Detta ledde till en ytterligare förändring i Coordinator skriptet, en komponent implementerades som i huvudsak övervakar rörelser som öppnar möjligheten för realtid strömning av modeller i applikationen.

6 RESULTAT

Arbetet resulterade i en fungerande demo applikation för Microsoft HoloLens som hämtar och laddar objekt från en server under körtid samt dela objekt mellan olika HoloLens enheter i realtid. Applikationens slutgiltiga systemarkitektur blev densamma som ritades upp i början av arbetet.



Figur 6.1: Applikationens flödesschema

Flödesschemat består av sju olika tillstånd:

Start

Applikationen kommer automatiskt in i detta tillstånd vid start och en kommunikationslänk mellan applikationen och Sharing Service skapas. Ett meddelande visas “waiting for model to load” som i sin tur hoppar vidare till nästa tillstånd.

WaitingForModelToLoad

Under det andra tillståndet laddar applikationen in den utvalda modellen från webbservern med hjälp av Bundle Downloader skriptet. Om nedladdning av modellen misslyckas laddar applikationen in den förbestämda modellen fallback prefab som definieras i Bundle Downloader.

WaitingToConnectStage

I detta tillstånd kontrollerar applikationen om det finns en kommunikationslänk till Sharing Service.

WaitingForRoomApiToStabilise

Applikationen hamnar sedan i detta tillstånd där den väntar på att allt är färdigladdat och en

stadig kommunikationslänk mellan applikationen och servern är skapad. Därefter kontrollerar applikationen om det finns ett rum på servern.

WaitingForModelPositioning

Applikationen hamnar enbart i detta stadie om det inte finns existerande rum i servern. Sedan får användaren placera objektet med hjälp av nyp-gesten samt låsa objektets position genom att använda röstkommandot "lock". Därefter skapas ett rum och applikationen går vidare till nästa tillstånd.

WaitingForWorldAnchorExport

Efter att användaren har använt sig utav röstkommando "lock" synkroniseras det skapade rummet med Sharing Service, med andra ord laddar användaren upp sitt befintliga rum till servern som andra enheter på samma nätverk kan hitta.

WaitingForWorldAnchorImport

Om det existerar ett rum i Sharing Service hamnar användaren i detta tillstånd där applikationen synkroniseras och laddar ner det rum som finns på servern, därefter kan denna användare se samma objekt som ägaren av rummet samt se alla interaktioner som görs på objektet av ägaren.

De frågeställningar som ställdes upp i början av arbetet besvaras nedan:

- Hur kan man skapa delning av visualiserade objekt för flera HoloLens enheter?
Arbetet resulterade i att applikationen kan skapa delning av visualiserade objekt för fler HoloLens enheter. Detta åstadkom med hjälp av en server kallad Sharing Service, detta öppnar upp möjligheten för flera enheter att kunna kommunicera sinsemellan. De olika HoloLens enheter renderar modellen var för sig, vilket leder till att delningen är baserad på modellens position som skickas sinsemellan. Detta gör att modellerna har samma position i varje enhet som kommunicerar med varandra.
- Vilka möjligheter till interaktion avseende påverkan på modellerna finns mellan flera enheter?
I detta arbete har delningen av interaktion på modeller utförts med hjälp av att dynamiskt ladda/radera World Anchor. Detta resulterar i att applikationen strömmar interaktionerna på modellen sinsemellan i realtid.

6 DISKUSSION

Inom ett utbildningsperspektiv har användning av augmented reality en väldigt stor potential. Genom att visualisera modeller och objekt i 3D kan det ge en tydligare beskrivning på idéer. Utvecklingen av applikationen har gett en inblick på vad som är möjligt att åstadkomma med denna teknologi. Interaktioner mellan användare och modeller går att dela mellan olika enheter i realtid vilket medför att en grupp studenter kan visualisera objekt för varandra. På grund av att HoloLens som enhet har inbyggda högtalare samt väldigt högupplösta hologram, ger detta en förstärkt beskrivning och visualisering av figurer.

Exempel på ett scenario kan vara en grupp läkarstudenter som arbetar tillsammans i ett labb. Där kan de studera hjärtat genom att ta del av augmented reality och visualisera hjärtat i form av en 3D-modell. Detta skulle kunna ge studenterna en klarare bild av hjärtats delar.

I dagsläget är det väldigt svårt att kunna ta del av AR överlag då de fysiska komponenter som stödjer denna teknologi är väldigt kostsamma samt väldigt svåra att få tag på. För att detta ska kunna nyttjas av alla så krävs det att den fysiska komponenten blir betydligt billigare samt att de blir mer lättillgängliga. Utöver det nämnda krävs det även att utveckling av applikationer inom AR blir enklare, detta för att öppna upp möjligheten för fler utvecklare att utveckla denna teknolgi.

Grundidéen med applikationen var att användarna skulle kunna kommunicera med varandra antingen genom röst eller ett kommentarssystem. Dessa prioriterades bort under projektets genomförande. Detta berodde på att en del funktioner tog längre tid att implementera än vad som förväntades.

Begränsningar fanns i utvecklingsverktyget Unity som användes för utveckling av applikationen och även i Hololens Emulator som användes för att testköra applikationen. Då nya versioner av Unity släpps ändras även programvarans bibliotek. Detta kan resultera i att applikationen som skapades i nuvarande version kan få kompileringsfel i kommande Unity versioner. Emulatorn var väldigt krävande på datorn vilket medförde lång väntetid under uppstart samt att verktyget kunde frysa vilket eventuellt ledde till krasch. Bortsett från emulatorns brister fungerade verktyget väldigt bra som ersättning av den fysiska enheten.

Under arbetsprocessen blev det svårt för gruppen att följa gantt-schemat som skapades under arbetets planeringsfas. Detta berodde på brist av tidigare erfarenheter och kunskap inom detta fält samt hantering av verktyg för att kunna skapa en sådan applikation. Förstudien underskattades vilket resulterade i att gruppen ändrade programmerings strategier för att tackla problem som uppstod.

Den agila arbetsmetod som följdes var väldigt givande för gruppen, genom att hålla en vardaglig kommunikering i gruppen har det varit enkelt gällande beslutstagande och diskussion om diverse problem och resultat som har uppnåtts. På grund av begränsad tillgänglighet av den fysiska enheten har det varit svårt att kunna testa applikationen kontinuerligt och mycket antagande om att det skulle funka gjordes istället.

Parprogrammering var en metod som fungerade väldigt bra då utvecklarna turades om att skriva kod, då den andra partnern agerade observatör kunde denne komma med idéer och lösningsförslag till problem som uppstod.

7 SLUTSATS OCH VIDAREUTVECKLING

I dagsläget existerar det enstaka applikationer som används inom ett utbildningssyfte. Dessa applikationer är väldigt beroende av hårdkodade modeller som levereras tillsammans med applikationen och detta medför en brist som begränsar applikationer att hålla sig till ett specifikt syfte. En ytterligare brist för dessa applikationer är att det inte finns någon möjlighet för flera enheter att kunna interagera med varandra. Den lösning som har utvecklats under arbetsgången har tacklat dessa problem genom att ladda in modeller från en extern plats och delningen har löst genom användning av Microsofts delningsverktyg. Den skapade lösningen är i nuläget inte en användbar applikation men är en stabil grund för vidareutveckling.

Vidareutveckling

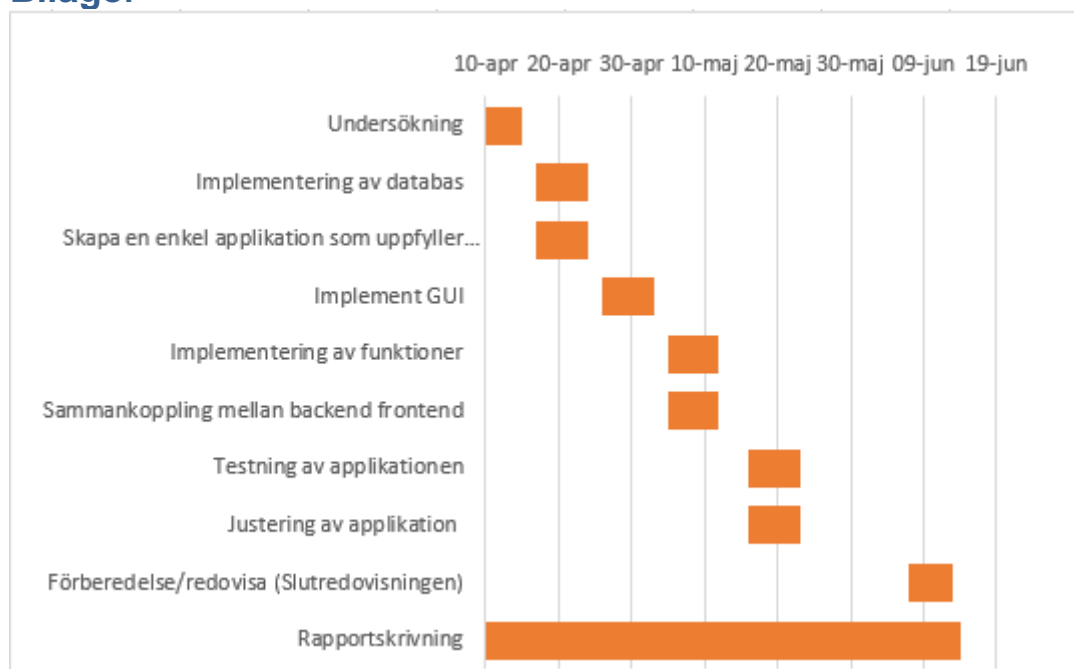
Applikationens användargränssnitt är väldigt begränsat i nuvarande stadiet, då det inte har varit en prioritet för projektets mål. För att kunna ge användaren en möjlighet att välja vilket objekt som skall laddas in krävs en implementation av ett mer utvecklat användargränssnitt. En annan implementering kan vara röstkommunikation. Detta ger användare som inte befinner sig i samma rum möjligheten att kunna kommunicera med varandra i realtid. En ytterligare funktion som skulle kunna implementeras är ett kommentarsystem som ger användaren möjlighet att kunna ställa en fråga i form av en kommentar utan att behöva avbryta personen som håller i föreläsningen.

Om applikationen vidareutvecklas till en färdig produkt kan en sådan applikation användas i samband med ett utbildningssyfte. En sådan användning kan vara att ersätta dagens föreläsningar med en digital föreläsning. Där modeller visualiseras i form av 3D-objekt och sådana föreläsningar kan uppfattas mer interaktivt mellan föreläsaren och eleven. Detta kan även ge en tydligare uppfattning om vad föreläsaren vill poängtera i sina föreläsningar.

REFERENSER

- [1] H. Body and A. Publishing, "Human Anatomy Atlas – 3D Anatomical Model of the Human Body på App Store", App Store, 2017. [Internet]. Hämtad från: <https://itunes.apple.com/se/app/human-anatomy-atlas-3d-anatomical-model-of-the-human-body/id446207961?mt=8>. [Hämtad 20 juni 2017].
- [2] "Augmented Reality", Augmentedreality.se, 2017. [Internet]. Hämtad från: <http://augmentedreality.se/>. [Hämtad: 19 juni 2017].
- [3] "Microsoft HoloLens". Microsoft HoloLens [Internet]. 2017. Hämtad från: <https://www.microsoft.com/en-us/hololens>. [Hämtad: 18 apr, 2017].
- [4] "Spatial sound", Developer.microsoft.com, 2017. [Internet]. Hämtad från: https://developer.microsoft.com/en-us/windows/mixed-reality/spatial_sound. [Hämtad: 18 juni. 2017].
- [5] "Holoportation - Microsoft Research", Microsoft Research, 2017. [Internet]. Hämtad från: <https://www.microsoft.com/en-us/research/project/holoportation-3/>. [Hämtad: 19 juni 2017].
- [6] Microsoft. HoloLens hardware details [Internet]. 2017. Hämtad från: https://developer.microsoft.com/en-us/windows/mixed-reality/hololens_hardware_details. [Hämtad 25 apr, 2017].
- [7] Microsoft. Gaze [Internet]. 2017. Hämtad från: <https://developer.microsoft.com/en-us/windows/mixed-reality/gaze>. [Hämtad 18 apr, 2017].
- [8] Microsoft. Gestures [Internet]. 2017. Hämtad från: <https://developer.microsoft.com/en-us/windows/mixed-reality/gestures>. [Hämtad 18 apr, 2017].
- [9] Microsoft. Use your voice with HoloLens [Internet]. 7 apr, 2017. Hämtad från: <https://support.microsoft.com/sv-se/help/12647/hololens-use-your-voice-with-hololens>. [Hämtad 18 apr, 2017].
- [10] Microsoft. Visual Studio IDE [Internet]. 2017. Hämtad från: <https://www.visualstudio.com/vs/>. [Hämtad: 18 apr, 2017].
- [11] Unity Technologies. Unity – Game engine, tools and multiplatform [Internet]. 2017. Hämtad från: <https://unity3d.com/unity>. [Hämtad 20 apr, 2017].
- [12] U. Technologies, "Unity - Manual: GameObjects", Docs.unity3d.com, 2017. [Internet]. Hämtad från: <https://docs.unity3d.com/Manual/GameObjects.html>. [Hämtad: 21 juni, 2017].
- [13] U. Technologies, "Unity - Manual: Anchors", Docs.unity3d.com, 2017. [Internet]. Hämtad från: <https://docs.unity3d.com/Manual/windowsholographic-anchors.html>. [Hämtad: 18 juni, 2017].
- [14] Unity Technologies. Scenes [Internet]. 2017. Hämtad från: <https://docs.unity3d.com/Manual/CreatingScenes.html>. [Hämtad 02 maj, 2017].
- [15] Unity Technologies. AssetBundles [Internet]. 2017. Hämtad från: <https://docs.unity3d.com/Manual/AssetBundlesIntro.html>. [Hämtad: 02 maj, 2017].
- [16] Microsoft. Vad är Azure? [Internet]. 2017. Hämtad från: https://azure.microsoft.com/sv-se/overview/what-is-azure/?&WT.srch=1&wt.mc_id=AID607369_SEM_aVauylXJ. [Hämtad 5 maj, 2017].
- [17] Microsoft. Using the HoloLens emulator [Internet]. 2017. Hämtad från: https://developer.microsoft.com/en-us/windows/mixed-reality/using_the_hololens_emulator. [Hämtad 21 apr, 2017].
- [18] Microsoft. HoloToolkit-Unity [Internet]. Github; 2017. Hämtad från: <https://github.com/Microsoft/HoloToolkit-Unity>. [Hämtad: 25 apr, 2017].

Bilagor



A1: Tidsestimering för detta projekt är redovisat i form av ett Gantt-schemat.