

Augmented reality: En applikation som underlättar visningen av 3D-modeller i sjukvården

Jesper Lundin (BME-19)

Sammanfattning—I denna rapport undersöks möjligheterna vid användning av HoloLens-2 i sjukvården. Rapporten beskriver en applikation som kan användas för att läsa in .obj-filer och visa dessa som 3D-modeller i augmented reality med hjälp av en HoloLens-2. 3D-modellerna som visas med hjälp av lösningen bidrar med en bättre helhetsbild än vad 2D-bilder kan göra då de kan skalas, roteras och positioneras av användaren i 3D-rummet. Denna typ av applikation kan bland annat användas som hjälpmedel vid hjärkateterisering och andra komplicerade ingrepp samt som hjälpmedel för läkare vid ronder eller multidisciplinära konferenser. Applikationen kan även användas som ett verktyg för att förklara komplicerade anatomier och procedurer för anhöriga och patienter med medfött hjärtfel. Rapporten tar även upp hur prestandan i HoloLens-2 påverkas av filernas storlek och vad detta har för konsekvenser för användarupplevelsen. I rapporten diskuteras framtida utvecklingsmöjligheter för lösningen och möjliga metoder för hur dessa kan implementeras.

I. INTRODUKTION

UTVECKLINGEN inom augmented reality (AR) kan användas vid effektivisering och förbättring av sjukvården. AR innebär att den verkliga världen kombineras med den virtuella med syftet att förbättra användarens upplevelse [1]. AR skiljer sig från virtual reality i den aspekt att den verkliga världen alltid visas i användarens vy [1]. Mixed reality (MR) innebär att den virtuella världen påverkas av den fysiska. Skillnaden mellan AR och MR är att den virtuella världen i AR kompletterar den verkliga världen medan MRs virtuella del förändras av den verkliga världen [2]. Då distinktionen mellan AR och MR inte är helt tydlig används ofta AR som samlingsnamn för bägge termer.

Flera stora företag använder sig av AR- och ibland även MR-teknik för att assistera vid insatser inom industrin och inom sjukvården. Företaget Medivis som använder Microsofts HoloLens-2 har utvecklat verktyg inom kirurgin som kallas SurgicalAR [3]. Deras programvara har assisterat vid kirurgi i ryggraden, vid reparation av nyckelben samt vid kosmetiska ingrepp [4]. Medivis programvara visar upp 3D-modeller av patientens anatomi. Neurokirurger från John Hopkins universitet har använt sig av AR vid spinalfusion och avlägsning av tumörer från ryggraden [5]. Neurokirurgerna använde sig av en programvara i AR-glasögonen som visade 3D-modeller av ryggkotorna vilket hjälpte dem att genomföra kirurgin [5]. HoloLens-2 har även användningsområden inom industrin.

Biltillverkaren Toyota Motor North America & kosmetikatillverkaren L’Oreal använder HoloLens-2 i utbildningssyfte och som hjälpmedel för att visa ritningar eller instruktioner [6]. Tillverkaren Lockheed Martin använder också HoloLens-2 för att visa ritningar [7].

HoloLens-2 tillåter funktioner som klassas in i både MR och AR, därmed kommer HoloLens-2 även benämnas som MR-glasögon i rapporten. Den viktigaste funktionen en HoloLens-2 har är att den kan visa upp virtuella 3D-modeller i det verkliga rummet, vilket ligger till grunden för både AR och MR. MR-glasögonen kan även använda sig av högtalare för att kommunicera med användaren. Vid användning placerar användaren sin HoloLens-2 på huvudet. När HoloLens-2 är igång läser den av användarens händer och de gester användaren gör med händerna. Detta kan, beroende på applikation, användas för att till exempel styra placeringen av objekt i AR- eller MR-miljön. HoloLens-2 mäter även av rummet, rörelserna från användarens ögon samt användarens röst. Ögonrörelserna mäts med infraröda sensorer och rösten mäts av med mikrofoner placerade på MR-glasögonen [8]. Detta medför att HoloLens-2 även ger användaren möjlighet att använda sig av ögon- och röststyrning i de applikationer som stödjer detta.

Då AR och MR möjliggör tekniken att visa upp 3D-modeller samtidigt som användaren kan överskåda rummet öppnar det upp för flera möjligheter i sjukvården. En av dessa möjligheter är att i vissa fall ersätta 3D-printing som verktyg då sjukvårdspersonal vill visa upp patientspecifika anatomiska modeller både för kollegor och patienter. En applikation som är skapad för att visa komplexa 3D-modeller med en HoloLens-2 kommer inte att kräva samma förberedelsestid som en komplex 3D-printad modell gör. En minskning i förberedelsearbete kan medföra att sjukvårdspersonal väljer att använda 3D-modeller som verktyg för att diskutera patientfall i större utsträckning. Om denna ökning i användning resulterar i att sjukvårdspersonalen får en bättre kommunikation och förståelse för sina patientfall kan det i slutändan rädda liv. Dessutom kan en sådan applikation även användas som ett komplement till de broschyrer som patienter får med information om ingrepp och procedurer och på sätt ge patienter bättre förståelse för den vård de kommer få ta del av.

II. SYFTET

Syftet med kandidatarbetet är att utveckla ett sätt att visa upp 3D-modeller i AR som sjukhuspersonal kan ha som hjälpmedel vid tillfällen såsom: under hjärkateterisering och andra komplicerade ingrepp; vid visning av anatomi och procedur för anhöriga och barn med medfött hjärtfel; vid ronder eller multidisciplinära konferenser.

Inlämnat den 6 juni 2022

Emejladress: {jesperolundin@gmail.com}

Teknisk handledare: Einar Heiberg, Klinisk fysiologi och nuklearmed., SUS

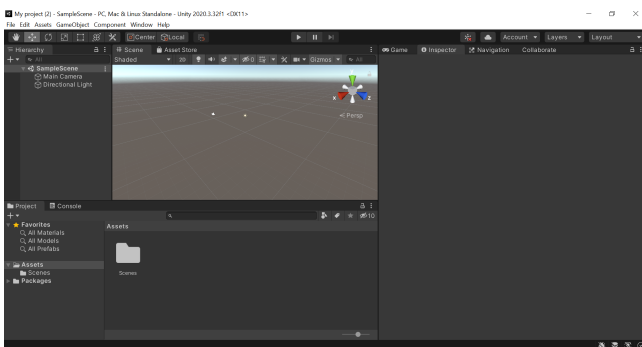
Klinisk handledare: Erik Ekbohm, 3D-centrum, SUS

Vid hjärkateterisering ska applikationen uppfylla syftet att visa upp en detaljerad modell av patientens anatomi i AR för att hjälpa användaren att lätt lokalisera punkter av intresse i anatomin. MR-glasögonen ska ge användaren en mer effektiv insyn i anatomin än vad 2D-bilder kan erbjuda före och under ingreppet. När applikationen används vid visning av anatomi och procedur för anhöriga och barn med medfött hjärtfel ska applikationen fungera som ett hjälpmedel till sjukhuspersonalen när de förklarar hur patientens anatomi ser ut och hur proceduren kommer att gå till. Applikationen i MR-glasögonen ska ge en bättre helhetssyn till patienten än vad 2D-bilder kan göra. Vid ronder ska applikationen vara ett smidigt sätt att visa punkter av intresse i patienternas anatomier. Applikationen ska även fungera som ett hjälpmedel vid multidisciplinära konferenser, där specialister från olika specialiteter diskuterar svåra patientfall och tillsammans gör en behandlingsplan.

I den *Tekniska dokumentationen* beskrivs det i steg hur en applikationen som har möjlighet att underlätta vid ingrepp och kommunikation med patienter kan återskapas. I *Metoden* beskrivs stegen i den tekniska dokumentationen endast översiktligt. Figurer som visar applikationen från den *Tekniska dokumentationen* kommer att redovisas i *Resultat* och förbättringsmöjligheter som kan underlätta för sjukvårdspersonal framgår i *Diskussionen*.

III. METOD

Det finns flera olika verktyg som kan användas för att skapa applikationer till HoloLens-2 såsom: Unity Hub; Unreal Engine; JavaScript; OpenXR [9]. För utvecklingen av denna applikation valdes Unity Hub då Unity erbjuder störst utbud av eftertraktade och för applikationen lämpliga funktioner som också möjliggör fortsatt vidareutveckling när kandidatarbetet avslutas. Unity Hub är en utvecklingsmiljö vilket innebär att det är ett datorprogram som erbjuder hjälpmedel för att skapa programvaror, se Figur 1. De funktioner som applikationen ska ha i HoloLens-2 införs i Unity Hub. Unity använder i sin tur Visual Studio som kompilator för att överföra applikationen till ett format (.exe) som kan köras av HoloLens-2.



Figur 1. Utvecklarens vy i utvecklingsmiljön Unity Hub.

Metoden som användes för att samla in information om hur applikationer utvecklas i Unity var att granska dokumentationen som fanns att tillgå på Microsofts hemsida. Därmed kommer majoriteten av informationen som användes

för att skapa projektet från Microsofts egen dokumentation. Även information från forum som gick att tillgå användes. Information om AR, MR, HoloLens-2 samt hur sjukvården kan använda sig av dessa granskades kritiskt och metodiskt. Testkörning av applikationen utfördes med en HoloLens-2 på 3D-centrum på Skånes Universitetssjukhus i Lund. För varje testkörning skapades en ny variant av programvaran i Unity som sedan lades över på en HoloLens-2 via Visual Studio. Vid testkörning användes avidentifierade segmenterade bilder från datortomografi samt andra modeller som framställts med hjälp av CAD-program. Mer än 100 versioner av programvaran testkördes. Motiveringen kring de val av funktioner som gjorts diskuteras i *Resultat* samt i *Diskussion*. Resterande del av metoden behandlar steg för steg hur den slutgiltiga applikationen återskapas och följer de rubriker som finns i den *Tekniska dokumentationen*.

A. Materiel

- Microsofts HoloLens-2 med tillhörande laddare
- Dator med nödvändig programvara

B. Programvara

- Windows 10 eller 11
- Visual Studio 2022 med inställningar för utveckling av applikationer i Unity [10]
- Unity Hub, version 2021.3.0f1
- Microsoft Reality Feature Tool

C. Filer

- Valfri 3D-modell av typen .obj

D. Skapandet av applikationen

Efter att alla programvaror hade laddats ner påbörjades utvecklingen av applikationen. Unity Hub anpassades till att skapa applikationer för HoloLens-2 och *Microsoft Reality Toolkit* (MRTK) lades till i projektet. MRTK medför att applikationen kan utnyttja funktionerna som finns i HoloLens-2 och erbjuder diagnosticeringsverktyg. Sedan inaktiverades inställningen *Spatial Perception Capability* som sköter MR-glasögonens inläsning av det verkliga rummet. Därefter redigerades ljussättningen av 3D-modellen för att ta bort att modellen belystes från en specifik vinkel. Efter att ljussättningen hade redigerats importerades 3D-modellen (i formatet .obj) in i projektet och placerades i programvyn. Vid import av objektet redigerades inställningen *Generate Colliders* så att modellen fick fysiska egenskaper i utvecklingsmiljön. Utan denna inställning uppfattar inte programmet om användaren pekar på objektet eller ej vilket medför att det inte kan interageras med. När 3D-modellen hade placerats i vyn skalades det till en storlek som var lämplig för HoloLens-2. Därefter lades kodfilen *Object Manipulator* till som medförde att objektet kunde skalas, roteras och förflyttas av användaren från avstånd i 3D-rymden. I den *Tekniska dokumentationen* beskrivs varje enskilt steg för processerna i denna och resterande rubriker i *Metoden*.

E. Funktion som låser kärmodellen med ett knapptryck

Efter att 3D-modellen hade ställts in infördes en kodfil i projektet som hanterade huruvida modellen kunde interageras med eller ej. Därefter skapades ett *Canvas-objekt* och en knapp lades till som var underordnad *Canvas-objektet*. Bägge skalades och placerades på lämpliga ställen för applikationen. Knappen var till för att styra om modellen kunde interageras med eller ej och *Canvas-objektet* var till för att knappen skulle kunna ärva funktioner från en kodfil som lades till i Canvas. När kodfilen implementerades i Canvas kunde knappen ärva funktionen *switchState()* från kodfilen som hanterar om modellen är låst eller ej.

Kodfilen blir inte aktiv i programmet då den skapas, då det krävs att kodfilen fästs som en komponent på ett objekt i scenen för att programmet ska använda kodfilen. Koden bakom *switchState()* är simpel och gör att objektet låser sig om det inte är låst, och låser upp sig om det är låst. Ytterligare en funktion som heter *changeText()* hanterar texten på knappen för att visa användaren om 3D-modellen kan interageras med. Funktionen *switchState()* anropar *changeText()* automatiskt. Kodfilen skapade jag själv.

F. Överföring av applikationen till HoloLens-2

Efter att kodfilen implementerats döptes projektet till ett lämpligt namn. Därefter packades den specifika scenen i projektet in i en *.sln*-fil från Unity och kördes i Visual Studio. Från Visual Studio distribuerades applikationen till en HoloLens-2 som en *.exe*-fil.

IV. RESULTAT

Kandidatarbetet på 3D-centrum resulterade i ett projekt i Unity som kunde läsa in en *.obj*-fil i ett fåtal steg och sedan överföra detta till en applikation. Efter att projektet skapats och färdigstälts en första gång kunde tiden för inläsning av resterande *.obj*-filer minimeras till cirka 10–15 minuter. Denna process kunde däremot ta cirka 60 minuter för en användare som inte tidigare genomfört processen. 3D-modellen kunde sedan visas upp i AR med hjälp av en HoloLens-2. När projektet skapats och färdigstälts en första gång behövde endast stegen i punkt-listan i *Framtida Utvecklingsmöjligheter* i *Diskussion* genomföras om ytterligare 3D-modeller skulle läsas in. Användarens vy från den resulterande applikationen visas i Figur 2 samt i Figur 3. Applikationen kunde användas vid följande tillfällen:

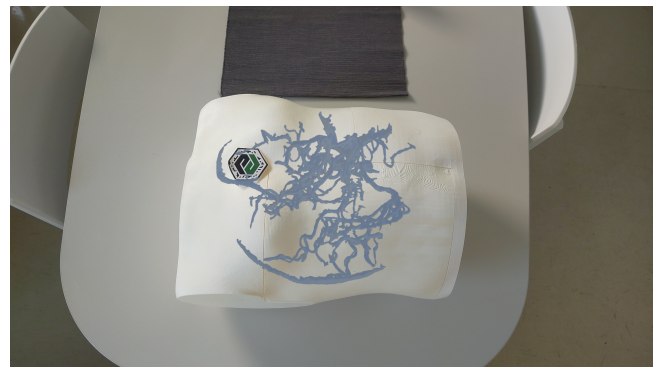
- Vid hjärkateterisering och andra komplicerade ingrepp.
- Vid visning av anatomi
- Vid beskrivning av tillvägagångssätt för planerad operation för anhöriga och barn med medfött hjärtfel
- Som hjälpmedel vid visualisering under ronder och vid multidisciplinära konferenser

Användningen under hjärkateterisering hade kunnat se ut likt bilden i Figur 3 eller likt Figurerna 7 och 8. Däremot hade läkaren förmodligen placerat 3D-modellen vid sidan om patienten under ingreppet för att snabbt kunna överskåda anatomin. Vid visning av anatomi och procedur för anhöriga och barn med medfött hjärtfel hade applikationen kunnat laddas

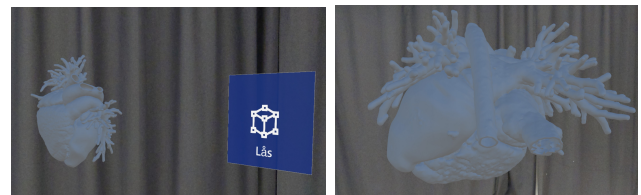
över till en HoloLens-2 och patienten hade sedan kunnat använda applikationen. Den hade kunnat användas likt bilderna i Figur 4. Vid ronder och multidisciplinära konferenser kan 3D-modeller användas för att förklara patienters anatomi eller för att planera komplicerade ingrepp. Applikationen hade även då kunnat ha figurer som liknar de i Figur 4.



Figur 2. Användarens vy i den slutgiltiga versionen av applikationen. 3D-modellen och knappen med låsfunktionen visas i vyn. Kärmodellen stämmer ej överens med de kärl som finns i bröstkorgen.



Figur 3. Användarens vy i den slutgiltiga versionen av applikationen. Här visas endast 3D-modellen då knappen ligger utanför användarens vy. Kärmodellen stämmer ej överens med de kärl som finns i bröstkorgen.

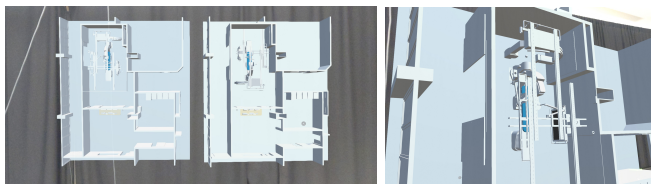


Figur 4. Vänster: Ett exempel på hur patientens vy kan se ut då patienten får använda en HoloLens-2 under genomgång av procedur vid medfött hjärtfel. Höger: En förstorad modell av 3D-modellen.

A. En applikation som kan visa byggnadsritningar

Vid testkörning av applikationen upptäcktes det att den fungerade bra för ett nytt användningsområde: det gick att visa planritningar till nybyggnationer smidigt med hjälp av applikationen (se Figur 5). Även då detta användningsområde

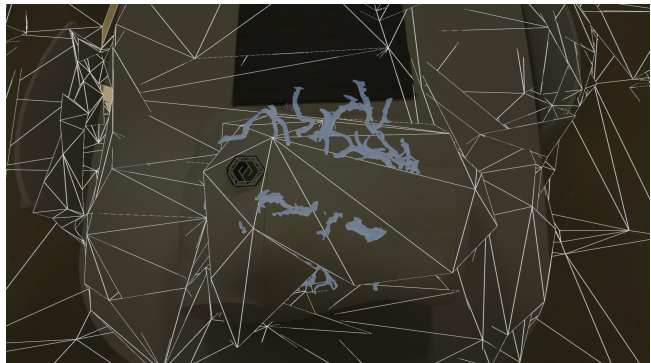
inte var något som var planerat vid skapandet av applikationen blev visning av ritningar den första praktiska användningen av applikationen under kandidatarbetet.



Figur 5. Vänster: Användarens vy vid uppvisning av två varianter av samma nybyggnation. Höger: Användarens vy då en nybyggnation skalats upp och betraktas på kort avstånd.

B. Spatial Perception Capability

I de första versionerna av programmet var *Spatial Perception Capability* aktiverat, vilket innebar att HoloLens-2-glasögonen läste av rummets geometrier och visade sin uppfattning av dessa i realtid i användarens vy. Denna funktion medförde att HoloLens-2-glasögonen kunde avgöra om den inlästa 3D-modellen låg bakom ett objekt och valde i sådant fall att inte visa 3D-modellen. Funktionen var inte önskvärd då den medförde att 3D-modellen lätt försvann bakom verkliga objekt och därmed inte blev tillgänglig för användaren att se eller interagera med. Avbildningen av rummets geometrier kunde även framstå som distraherande för användaren varpå *Spatial Perception Capability* inaktiverades. Användarens vy då funktionen är aktiverad visas i Figur 6.



Figur 6. Vyn från applikationen i HoloLens-2 då *Spatial Perception* var aktiverad. Kärnmodellen göms inuti den verkliga modellen av bröstkorgen. Inläsningen av rummet täcker mycket av synfältet. Mönstret blir intensivare på bild än vad det är i användarens vy vid körning av programmet.

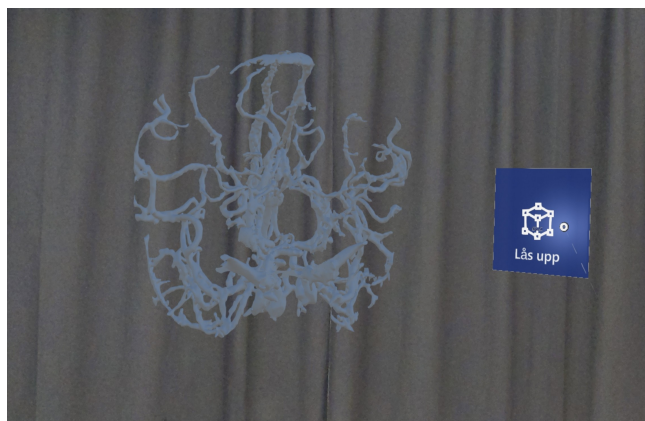
C. Funktion till att låsa fast modellen i 3D-rummet

Efter att en tidig version av programmet visades för två läkare, en från Kranskärskliniken samt en på Neuroradiologi i Lund, kom läkarna med återkopplingen att modellen lätt förflyttades vid tillfällena då den redan hade placerats i sin önskade position. Läkarna önskade ett sätt att låsa fast modellen i 3D-rummet när den nått önskvärd position för att förbättra användarupplevelsen. Efter denna feedback lades en funktion till som läste fast modellen om användaren tryckte

på en knapp i användarens vy. Denna knapp visas i Figur 7 och i Figur 8 och är samma knapp som anropar funktionen *switchState()*. För att öka användarvänligheten lades även kod till för att ändra på texten på knappen för visa användaren att modellen blivit låst och den sedan kunde låsas upp igen.



Figur 7. Användarens vy då en låsknapp har lagts till. På knappen står texten "Lås" vilket visar att modellens position kan fixeras i 3D-rummet. Strecket och punkten visar att användaren interagerar med 3D-modellen.

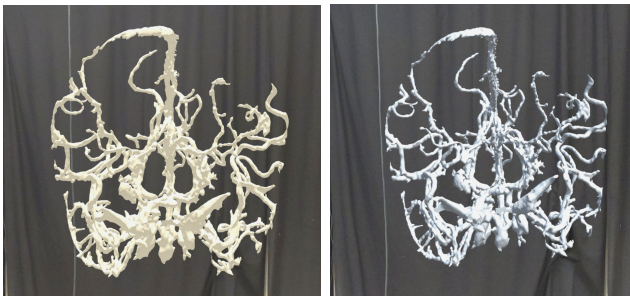


Figur 8. Användarens vy då en låsknapp har lagts till. På knappen står texten "Lås upp" vilket visar att modellens position kan fixeras i 3D-rummet och nu kan låsas upp med ett knapptryck. Pekaren visar att knappen vid detta tillfälle är markerad.

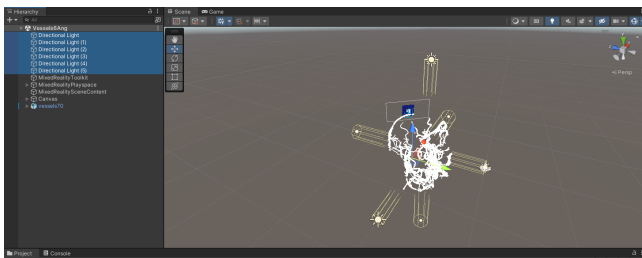
D. Ljussättning

Vid testkörning av applikationen på 3D-centrum upptäcktes det att ljussättningen på 3D-modellen var annorlunda i olika vinklar. Detta uppstod då standardprojektet som skapas via *Microsoft Reality Toolkit* har en riktad ljuskälla *Directional Light*. En oönskad effekt av detta var att skuggor uppstod på 3D-modellen som resulterade i att alla dess ytor inte visades lika tydligt för användaren. Om den riktade ljussättningen nollställdes medförde detta modeller som var mer behagliga att se på då de inte lyste lika starkt. Detta ligger till grund till beslutet att nollställa den riktade ljuskällan i steg 2 i *Skapandet av applikationen* i *Teknisk dokumentation*. Två alternativ till

att nollställa ljuskällan visas i Figur 9. Det vänstra alternativet i Figur 9 uppnåddes med hjälp av att skapa sex ljuskällor (se Figur 10).



Figur 9. Vänster: Användarens vy då sex riktade ljuskällor används. Höger: Användarens vy då en riktad ljuskälla används. I den vänstra bilden har majoriteten av skuggorna försvunnit.



Figur 10. Exempel på orientering av 6 riktade ljuskällor kring en 3D-modell i utvecklingsmiljön. Detta uppnås genom att duplicera *Directional Light* fem gånger och ändra dess värden för rotation och positionering.

E. Observationer

Vid testkörningar av applikationen upptäcktes begränsningar i prestandan för HoloLens-2 då 3D-modeller med filstorlekar som översteg cirka 10000 kilobyte (kB) användes. Med begränsningar menas att MR-glasögonens bilduppdateringsfrekvens (eng. frames per second, förkortat FPS) stundvis minskade till den grad att det påverkade användarupplevelsen genom att programmets bild frös fast eller då inläsningen av användarens gester försvann. Detta undersöktes vidare genom en testkörning med tre storlekar på 3D-modeller på följd. Applikationens bilduppdateringsfrekvens noterades vid skalning, positionering och rotering av 3D-modellen med hjälp av diagnostiseringsverktyget *Visual Profiler* som tillhandahålls med *Microsoft Reality Toolkit*:

- Kärmodellen med storleken 7739 kB påverkade inte användarvänligheten. I enstaka fall förlorade applikationen cirka 5 FPS. Genomsnittlig FPS var ej påverkad avsevärt.
- Kärmodellen med storleken 10909 kB hade viss påverkan på användarvänligheten. Detta märktes av då applikationen vid enstaka tillfällen förlorade cirka 20 FPS. Genomsnittlig FPS var inte påverkad avsevärt.
- Kärmodellen med storleken 15874 kB hade stor påverkan på användarvänligheten. Under körning av applikationen hängde bilden sig och MR-glasögonen misslyckades vid många tillfällen att läsa av händernas gester.

Applikationen hade vid många tillfällen förluster på kring 30 FPS och dess genomsnittliga FPS sjönk till ca 45 FPS.

Bilduppdateringsfrekvens vid normal prestanda är 60 FPS. Det observerades generellt att applikationerna inte presterade bra vid stora filstorlekar och att detta ledde till förluster i användarupplevelsen. Det observerades även att MR-glasögonens möjlighet att följa handrörelser begränsades vid en stor förlust av FPS som skedde i samband med stora filstorlekar. Ytterligare ett test utfördes vilket var att inkludera *Spatial Perception Capabilities* i projektet vid körning av applikationerna. Inkluderingen av inställningen hade inte särskilt stor påverkan på prestandan förrän det användes vid stora filstorlekar och försämrade då prestandan ytterligare.

V. DISKUSSION

Resultatet av projektet blev en lösning där man snabbt kan skapa applikationer med specifika 3D-modeller. Förloppet från inläsning av en .obj-fil till skapandet av en användbar applikation redo för inläsning i HoloLens-2 är med lite erfarenhet en kort process. Däremot kan det ta mycket tid att ladda in en 3D-modell om vanan inte finns där. För att motverka detta skrevs en detaljerad teknisk dokumentation. Vikten av att det går att utföra processen från inläsning av modell till applikation relativt snabbt är stor då det kommer medföra att användningen av applikationerna inte bortses från i lika hög grad. Då en snabbare inläsning av 3D-modeller förmodligen leder till en högre grad av användning för applikationerna är det viktigt att effektivisera denna process.

A. Framtida utvecklingsmöjligheter

1) *Smidigare inläsning av 3D-modeller*: Den viktigaste förbättringsmöjligheten som finns i applikationen är att underlätta inläsningen av objekt. Efter att ha skapat ett fullständigt projekt i enlighet med instruktionerna i *Metod* behöver sjukhuspersonalen i nuläget göra om flera av stegen vid inläsningen av nya modeller. Denna process är något som tar viktig tid från personalen på sjukhuset. Processen medför att varje ny modell behöver läsas in som en ny applikation i HoloLens-2. Detta medför även att applikationer ackumuleras på MR-glasögonen vid skapandet av nya applikationer vilket leder till att gamla applikationer behöver avinstalleras från MR-glasögonen vid jämna mellanrum. Stegen från den *Tekniska dokumentationen* som behöver göras vid inläsning av en ny modell följer:

- Radering av föregående modell
- Steg 3-5 i *Skapandet av applikationen*
- Vänsterklicka på **Canvas**. Dra 3D-modellen som finns i **Hierarchy** till rutan bredvid **Model** i komponenten **Låsfunktion** i **Inspector**.
- Alla steg i *Överföringen av applikationen till HoloLens-2*

För att åtgärda detta problem utvecklades kod för ytterligare en knapp som skulle sköta inläsning av nya filer från mappar som finns i HoloLens-2. Tanken var att öppna upp utforskaren i HoloLens-2 samtidigt som applikationen kördes vilket skulle tillåta användaren att läsa in filer direkt från HoloLens-2 egna mappar. Därefter skulle .obj-filer redigeras automatiskt

då de placerades i programmet. Om det fungerade skulle det medföra att endast en applikation hade behövts laddas över på MR-glasögonen och att .obj-filerna endast skulle behöva överföras till MR-glasögonen. Det hade minskat tiden för att förbereda en visning i applikationen avsevärt, möjligtvis ner till ca tre minuter.

2) *Tvårsnittsplån*: En utvecklingsmöjlighet som hade hjälpt vid förklaring av anatomi eller procedurer hade varit införandet av ett tvårsnittsplån. Kod hade behövts införas för att få planet att tillfälligt radera allt från användarens vy på en sida om planet. På sådant sätt hade applikationen kunnat visa olika tvårsnittsplån av modellerna.

3) *Ljussättningen*: Då den vänstra 3D-modellen i Figur 9 har sex olika ljuskällor resulterar det i en modell som lyser i ett starkare ljus. Detta händer då applikationen uppfattar 3D-modellen som ljusare. Att modellen lyser starkt kan uppfattas som störande av användaren. Detta hade gått att justera genom att sänka intensiteten på det riktade ljuset.

Likt hur en HoloLens-2 hanterar användarens vy då den kör en applikation hade en kodfil kunnat fästas som komponent på den riktade ljussättningen *Directional Light*. Kodfilen hade givit *Directional Light* samma koordinater och vinkel som HoloLens-2 använder för användaren. På sådant sätt hade den riktade ljuskällan följt användaren och lyst upp modellen likt en pannlampa i det virtuella rummet. Denna funktion hade varit intressant i de sammanhang då skuggor på modellerna gärna inkluderas.

4) *Användargränssnitt*: Inom användargränssnittet finns det många möjliga utvecklingsmöjligheter. En meny kan göras för att ge fler funktioner till användaren, en viktig sådan vore en funktion för att stänga ner applikationen med hjälp av en knapptryckning. En annan funktion som hade varit användbar är att kunna förflytta låsknappen medan applikationen körs. För tillfället placeras knappen likt 3D-modellen orienterade efter var användaren startade applikationen. I kombination med att applikationen inte kan startas om på ett enkelt sätt medför detta ytterligare problem för användaren om användaren inte startar applikationen inom det område som det är tänkt att den ska öppnas på.

För vissa modeller som har mycket hålrum hade det varit bra med en ruta runt 3D-modellen, likt en bounding box, som användaren kan använda för att styra modellens position, rotation och skala med hjälp av då modeller med hålrum är svårare att greppa tag i på avstånd. 3D-modeller som har mycket hålrum är exempel modeller av kärl i en viss kroppsdel.

B. Hållbarhet och etik

Inom ämnet hållbarhet uppfyller applikationen kravet för ekologisk hållbarhet då den minskar antalet 3D-modeller som används som hjälpmedel som behöver 3D-printas. Dessa modeller kan i stället för att 3D-printas läsas in i applikationen och tas med under ronder eller vid konferenser. Om applikationen används som hjälpmedel för planering av nybyggnationer och uppfattas som ett mer intuitivt sätt att kommunicera planritningarna mellan parter kommer detta att medföra att ombyggnationer kan minimeras vilket i sin tur resulterar i att mindre material behöver ödas.

När det kommer till komplicerade och farliga ingrepp såsom hjärtfel är det viktigt att patienten och dess anhöriga har möjligheten att förstå innebörden av ingreppet. En applikation som kan visa upp 3D-modeller som hjälpmedel för att förklara de procedurer som händer vid ingrepp kan medföra att patienten känner sig mer säker kring den sjukvården som utövas. Detta kan i sin tur resultera i att patienter inte upplever stress i lika hög grad inför ingrepp. En applikation som på ett enkelt sätt visar hur sjukvårdsproceduren går till bidrar även till en mer jämlik sjukvård och upprätthåller därmed den sociala aspekten i hållbarhet.

VI. SLUTSATSER

Slutsatsen från projektet är att det går att skapa applikationer som visar 3D-modeller i AR och att det finns flera olika användningsområden för detta. Vidare utveckling hade behövts för att införa fler funktioner som hade bidragit till användarvänligheten i applikationen. Den viktigaste funktionen att införa hade varit filinläsning i själva HoloLens-2 då det hade ökat användarvänligheten avsevärt. I det nuvarande skedet uppfyller applikationen sin uppgift med varierande kvalitet beroende på användningsområde & kvalitet samt storlek på digitala modeller.

VII. EFTERORD

Tack till mina handledare Einar Heiberg och Erik Ekbohm som givit kontinuerlig feedback under arbetets gång samt kommit fram till idén att skapa applikationen. Jag tackar även Muris Imširović på 3D-centrum som upptäckte att ljussättningen i programmet skilde sig i olika vinklar och som bidrog med feedback. Jag riktar även ett tack till läkarna på kranskärskliniken och neuroradiologi i Lund som kom med viktig feedback. Författare är Jesper Lundin.

VIII. TEKNISK DOKUMENTATION

A. Skapandet av applikationen

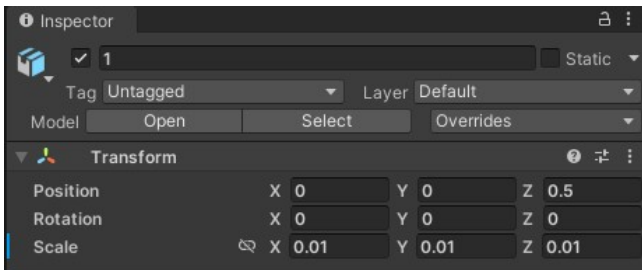
Genomför stegen till och med *Unit 4 i Introduction to the Mixed Reality Toolkit-Set Up Your Project and Use Hand Interaction* [11]. Börja därefter med stegen nedanför.

1) *Inaktivering av Spatial Perception Capability*: Inställningen nås via **Edit – Project Settings – Player - Publishing Settings – Capabilities – Spatial Perception**. När inställningen funnits ska rutan som är markerade checkas ur.

2) *Redigering av ljussättning*: Denna funktion tas bort genom att vänsterklicka på **Directional Light**. Fönstret till höger i projektet kallas **Inspector**. Ändra **Intensity** till 0 för komponenten (eng. component) **Light i Inspector**.

3) *Inläsning av 3D-modell*: Dra först in den valfria .obj-filen i mappen **Assets** som finns i fönstret **Project** i nedre delen av projektet. Vänsterklicka på .obj-filen i **Assets**. Checka ur **Convert Units** samt checka i **Generate Colliders i inspector**. Klicka sedan på knappen **Apply** längst ner i **Inspector**. Fönstret till vänster i projektet kallas **Hierarchy**. Dra den nu redigerade .obj-filen in i scenen genom att placera den i **Hierarchy**.

4) *Storlek och placering*: Blir 3D-modellen för stor eller för liten redigeras inställningarna genom att vänsterklicka på 3D-modellen i **Hierarchy**. Därefter redigeras **Scale** i komponenten **Transform** i **Inspector**. Alternativt redigeras **Scale Factor** i **Inspector** innan den placeras i **Hierarchy** i föregående steg. Placeringen av objektet ändras genom att redigera **Position** under **Transform** i **Inspector**. I Figur 11 nedan följer ett exempel på inställningar för den kärldmodell som visas i *Resultat*.



Figur 11. Exempel på inställningar för kärld modellen som används i *Resultat*. I detta fall befinner sig objektet i Z = 0.5 och skalan är 0.01 i alla led.

5) *Lägg till kodfiler*: Vänsterklicka på 3D-modellen genom att markera den i **Hierarchy**. Lägg till komponenten **Object Manipulator** genom att trycka på **Add Component** i **Inspector** och sök efter komponenten i rutan som kommer upp. Komponentens **Constraint Manager** läggs automatiskt till då **Object Manipulator** läggs in.

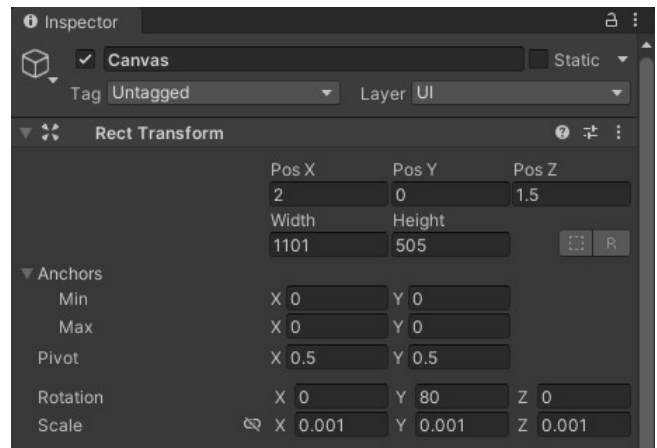
B. Funktion som låser kärld modellen med ett knapptryck

1) *Skapa kodfil*: Öppna upp den elektroniska bilagan och importera denna in i projektet. Låsfunktionen med tillhörande textredigeringsfunktion ska nu finnas i mappen **Assets** i projektet.

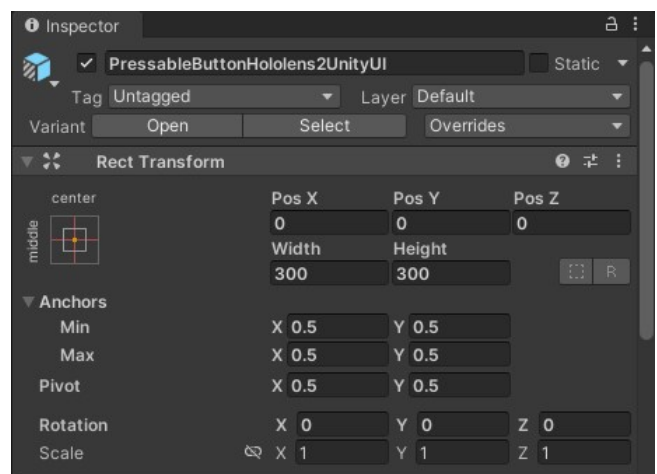
2) *Skapa och ställ in ett Canvas-objekt*: Högerklicka i **Hierarchy** och välj **UI > Canvas** för att skapa ett **Canvas**-objekt. Vänsterklicka sedan på **Canvas** i **Hierarchy**. Vänsterklicka på knappen **Convert to MRTK Canvas** och **Add NearInteractionTouchableUnityUI** i **Inspector**. Redigera sedan skalan och placeringen av **Canvas** för att det ska passa in i projektet likt steget *Storlek och placering*. I Figur 12 nedan följer ett exempel på inställningar för det **Canvas**-objekt som används i programmet som visas i *Resultat*. I det fjärde steget läggs kodfilen som tidigare skapats in i scenen.

3) *Skapa och ställ in en knapp*: Vänsterklicka på **All Prefabs** i mappen **Projekt** nere till vänster i fönstret. Finn **PressableButtonHoloLens2UnityUI.prefab** i listan. Dra sedan in denna fil i **Canvas** i **Hierarchy**. Expandera **PressableButtonHoloLens2UnityUI** samt **IconAndText** genom att klicka på pilen vänster om namnet i **Hierarchy**. Redigera sedan storleken för att få en bra placering i **Canvas**. I Figur 13 samt Figur 14 nedan finns exempel på den skalning och orientering för **PressableButtonHoloLens2UnityUI** samt **IconAndText** som används i programmet i *Resultat*.

4) *Implementera kodfilen i projektet*: Vänsterklicka på **Canvas** i **Hierarchy**. Vänsterklicka sedan på **Add Component** i **Inspector** och lägg till kodfilen **Låsfunktion** i sökrutan



Figur 12. Exempel på inställningar för **Canvas**. I detta fall är skala, rotation samt position ändrade från standardinställningar.



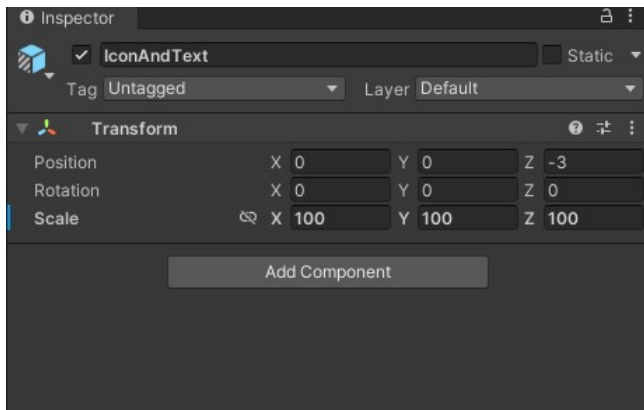
Figur 13. Exempel på inställningar för **PressableButtonHoloLens2UnityUI**. I detta fall är bredden och höjden ändrad från standardinställningarna.

som visas. Dra 3D-modellen som finns i **Hierarchy** till rutan bredvid **Model** i komponenten **Låsfunktion** i **Inspector**. Dra sedan **TextMeshPro** som finns under **PressableButtonHoloLens2UnityUI > IconAndText** i **Hierarchy** till rutan höger om **Btn Text** i komponenten **Låsfunktion**. Vänsterklicka på knappen **PressableButtonHoloLens2UnityUI** i **Hierarchy**. Navigera ner i **Inspector** till **Interactable > Events > On Click()**. Under rutan **Runtime Only** finns en tom ruta. Dra in **Canvas** från **Hierarchy** i den tomma rutan. Lägg därefter till den funktion som knappen ska utföra genom att klicka i rutan höger om **Run Time Only**. Välj **Låsfunktion > switchState()**.

C. Överföring av applikationen till HoloLens-2

1) *Utvecklarläge och parkoppling*: Följ Microsofts instruktion i *Using Visual Studio to deploy and debug* för hur utvecklarläge och parkoppling ställs in för enheten [12].

2) *Ställ in applikationens namn*: Vänsterklicka på **Edit - Project Settings** uppe till vänster i projektet. Se till så att **Player** är markerat till vänster. Ändra namnet i rutan bredvid **Product Name** för att sätta in valfritt nytt namn på



Figur 14. Exempel på inställningar för **IconAndText**. I detta fall är skalan ändrad från standardinställningar.

applikationen. Klicka sedan på **X** i övre högra hörnet för att stänga ner fönstret.

3) *Skapa Visual Studio-filen:* Vänsterklicka på **File - Build Settings**. Avmarkera alla scener under **Scenes In Build** i **Build Settings**. Vänsterklicka sedan på knappen **Add Open Scenes**. Navigera ner och tryck på knappen **Build**. Skapa en ny mapp med valfritt namn och välj att placera filerna i mappen.

4) *Distribuera applikationen:* Efter att projektet har byggt färdigt filerna lyser mappen upp i aktivitetsfältet. Dubbelklicka .sln-filen i mappen för att öppna upp den i Visual Studio. Ändra parametrarna **Debug** och **x64** till **Release** samt **ARM64** i övre delen av fönstret genom att klicka på respektive inställning. Ändra **Remote Machine** till **Device** på samma sätt. Logga in på HoloLens-2-glasögonen och koppla in dem i datorn. Vänsterklicka på den ihålliga **Play**-ikonen höger om **Device** i Visual Studio. Vänsterklicka sedan på **Yes** i dialogrutan. Därefter startar applikationen automatiskt upp på den parkopplade HoloLens-2-glasögonen.

BILAGA

Kodfilen finns som elektronisk bilaga.

REFERENSER

- [1] Wikipedia.org. 2022. *Augmented reality* - Wikipedia. [online] Available at: [https://en.wikipedia.org/wiki/Augmented\\$reality](https://en.wikipedia.org/wiki/Augmented$reality) [Accessed 22 May 2022].
- [2] Wikipedia.org. 2022. *Mixed reality* - Wikipedia. [online] Available at: https://en.wikipedia.org/wiki/Mixed_reality [Accessed 22 May 2022].
- [3] Medivis. *SurgicalAR* - Medivis, [online] Available at: <https://www.medivis.com/surgicalar> [Accessed 23 May 2022].
- [4] Palladino, T., 2019. *Medical Software Maker Medivis Closes \$2.3 Million in Funding to Bring HoloLens to Surgical Platform* - Next Reality, [online] Available at: <https://hololens.reality.news/news/medical-software-maker-medivis-closes-2-3-million-funding-bring-hololens-surgical-platform-0193743/> [Accessed 23 May 2022].
- [5] Witham, T., 2021. *Johns Hopkins Performs Its First Augmented Reality Surgeries in Patients* - Neurologic Winder, [online] Available at: <https://www.hopkinsmedicine.org/news/articles/johns-hopkins-performs-its-first-augmented-reality-surgeries-in-patients> [Accessed 22 May 2022].
- [6] Jenkins, R., 2022. *Manufacturers are mitigating disruptions with HoloLens 2 and Dynamics 365 mixed reality solutions* - Microsoft Dynamics 365 Blog, [online] Available at: <https://cloudblogs.microsoft.com/dynamics365/bdm/2022/03/24/manufacturers-are-mitigating-disruptions-with-hololens-2-and-dynamics-365-mixed-reality-solutions/> [Accessed 22 May 2022].
- [7] Endicott, S., 2020. *HoloLens 2 is 'helpful, time-saving, and fun' for people working on Orion spacecraft* - Windows Central, [online] Available at: <https://www.windowscentral.com/hololens-2-helpful-time-saving-and-fun-those-working-orion-spacecraft> [Accessed 22 May 2022].
- [8] Microsoft Store. *HoloLens 2: Find Specs, Features, Capabilities & More* - Microsoft, [online] Available at: <https://www.microsoft.com/en-us/d/hololens-2/91pnzznzwp?activetab=pivot:techspecstab#tab189249f48-9de4-4fce-84f6-2bab6dfe6e9c> [Accessed 22 May 2022].
- [9] Docs.microsoft.com. 2022. *Välja motor* - Microsoft, [online] Available at: <https://docs.microsoft.com/sv-se/windows/mixed-reality/develop/choosing-an-engine?tabs=unity> [Accessed 22 May 2022].
- [10] Docs.microsoft.com. 2022. *Installera verktygen* - Microsoft, [online] Available at: <https://docs.microsoft.com/sv-se/windows/mixed-reality/develop/install-the-tools> [Accessed 22 May 2022].
- [11] Docs.microsoft.com. 2022. *Introduction to the Mixed Reality Toolkit- Set Up Your Project and Use Hand Interaction* - Microsoft, [online] Available at: <https://docs.microsoft.com/en-us/learn/modules/learn-mrtk-tutorials/> [Accessed 18 May 2022].
- [12] Docs.microsoft.com. 2022. *Using Visual Studio to deploy and debug* - Microsoft [online] Available at: <https://docs.microsoft.com/en-us/windows/mixed-reality/develop/advanced-concepts/using-visual-studio?tabs=hl2> [Accessed 22 May 2022].