

Distribuerad förstärkt verklighet för att stödja kommunikation mellan arkitekt och beställare



Raafat Hussamadin

Civilingenjör, Arkitektur
2018

Luleå tekniska universitet
Institutionen för samhällsbyggnad och naturresurser

Titel: Distribuerad förstärkt verklighet för att stödja kommunikation mellan arkitekt och beställare

Författare: Raafat Hussamadin

Datum: 2018-08-30

Publikation: Examensarbete

Utbildning: Civilingenjör Arkitektur, Husbyggnad

Universitet: Luleå Tekniska Universitet

Institution: Institutionen för samhällsbyggnad och naturresurser

Avdelning: Avdelningen för industriellt och hållbart byggande

Examinator: Gustav Jansson

Handledare: Erik Hidman

Förord

Detta examensarbete utgör den avslutande delen för examen i Civilingenjör Arkitektur med inriktning Husbyggnad vid Luleå tekniska universitet. Examensarbetet omfattar 30 högskolepoäng och har utförts under våren 2018.

Jag skulle vilja tacka alla som har medverkat och hjälp med utförande av detta examensarbete. Ett stort tack till:

- Gustav Jansson
- Peter Törlind
- Anton Pålstam
- Ahmad Hussamadin
- Mohammed Maarooof
- Fadya Al-Naqeeb
- Kristin Westlund

Luleå, 2018

Raafat Hussamadin

Sammanfattning

Collaborative design inträffar när flera aktörer samarbetar för ett gemensamt mål eller intresse som inte kan åstadkommas ensamt. Collaborative design sker antingen i ett co-located möte "ansikte mot ansikte" eller distribuerat möte "distansmöte". Distribuerade möten, oftast via ett nätverkssystem, sker då olika aktörer av ett projekt inte befinner sig i närheten av varandra. Målet med distribuerad kommunikation är att kunna återskapa det fysiska mötet i ett virtuellt möte.

Syftet med denna studie är utveckling av en kommunikationsmetod som med hjälp av förstärkt verklighet kan visualisera och överföra information mellan arkitekt och beställaren i ett distribuerat möte. Metoden baseras på mjukvaruutveckling i spelmotorer vilka har kapacitet för skapande av realtidsvisualiseringar i kombination med flera skript som kan utföra många komplexa funktioner, inklusive synkron dataöverföring. Då examensarbetet fokuserar på distribuerad kommunikation har mjukvaruutvecklingen begränsats till endast smarta mobiltelefoner, detta på grund av stora tillgängligheten till enheten i allmänheten. Ett ytterligare syfte är undersökning om hur överföring av BIM kan simplificeras och automatiseras till spelmotorn Unity.

Utvecklingen av demonstratorn visar att BIM-överföring kan ske via scheman skapade i BIM-mjukvaror såsom Autodesk Revit som inläses av skript i Unity. Överföringsmetoden visualiserar BIM i realtid och även skapandet av funktioner som integrerar BIM, inklusive våningsplanväljare och lagervisningsväxlare. Systemet för nätverkssynkronisering möjliggör synkron kommunikation mellan alla aktörer genom användande av funktioner och flera skript som möjliggör positionssynkning, skärmdelning och objektindelning.

Resultatet från studien gällande förstärkt verklighet har kombinationen av fördelar och nackdelar varierade beroende på aktörernas erfarenhet. Den utvecklade mjukvaran med tillhörande metod visar på potential för användning gällande distribuerad kommunikation.

Abstract

Collaborative design occurs when individuals work together to achieve a common goal or interest that cannot be accomplished alone. Collaborative design take place either in a co-located meeting “face to face” or a distributed meeting “distance meeting”. Distributed meetings, often via a networking system, occur when individuals of a project don’t have the opportunity for a co-located meeting. The goal of distributed communication is to be able to recreate the physical confrontation in a virtual meeting.

The purpose of this study is to develop a communication software that, with addition of augmented reality, can visualize and share information between architects and clients in a distributed meeting. For software development game engine Unity is used, which has the capacity to create a real-time visualization in combination with scripts that can add features such as synchronous data transfer. As the thesis focuses on distributed communication a limitation has been put on used devices to only consider smartphones, this is due to the high level of availability of the device in general. An additional purpose of the thesis is to simplify and automate BIM transfer to Unity game engine.

The development of a demonstrator shows that BIM transfer can be done via usage of schedules that can be created in BIM software such as Autodesk Revit, which is later read by scripts in Unity. The software visualizes BIM in real time and supports creation of functions that can integrate with BIM, including plan selector “våningsplanväljare” and layer manager “lagervisningsväxlare”. The network synchronization system enables synchronous communication between all users, using functions such as position sync, screen sharing and object sharing.

There has been a combination of advantages and disadvantages for augmented reality that varied depending on individual experience. The developed software with methods has potential for usage in use of distributed communication.

Innehållsförteckning

Förord.....	i
Sammanfattning.....	ii
Abstract.....	iii
Innehållsförteckning	iv
Figurförteckning.....	vi
Beteckningar och begrepp	viii
Inledning	1
Bakgrund	1
Syfte och mål.....	2
Forskningsfrågor	2
Avgränsning.....	2
Förväntade resultat.....	3
Förväntade problem	3
Teori	4
BIM	4
BIM mjukvara	5
Projektering.....	6
Arkitektonisk kommunikation med Collaborative design	7
Co-located arkitektonisk kommunikation	7
Distribuerad arkitektonisk kommunikation.....	8
Förstärkt verklighet (AR).....	9
Markörbaserad förstärkt verklighet	9
Markörlös förstärkt verklighet.....	10
Spelmotor.....	12
Prestanda optimering	12
Metod.....	13
Litteraturstudie	13
Intervju.....	13
Hårdvara.....	14
BIM via Autodesk Revit	15

Spelmotor.....	16
Överföringsflöde till spelmotor.....	16
Arbetsflöde för prestandaoptimering.....	16
Mjukvaruutveckling	17
Utveckling demonstrator	18
Användargränssnitt.....	18
Integration av BIM till spelmotor.....	20
Våningsplanväljare och lagervisningsväxlare	24
Objektmarkör	28
Förstärkt verklighet.....	30
Kamerastyrning.....	31
Nätverkssynkronisering	36
Realtidsskuggnings kontroll	40
Diskussion och slutsats	41
Hur kan spelmotorer möjliggöra förstärkt verklighet och nätverkssynkroniserad distribuerad kommunikation?.....	41
Hur kan metoden för en förstärkt verklighets miljö beskrivas för distribuerad möte?	41
Hur används nätverkssynkronisering i metoden för förstärkt verklighet?	41
Distribuerade kommunikationsverktyg i spelmiljöer	42
BIM-integration och visualisering.....	43
Förstärkt verklighet.....	43
Förslag till fortsatta studier.....	44
Referenser.....	46

Figurförteckning

Figur 1 Produktivitetsflöde vid användning av BIM i jämförelse med 2D CAD (Graphisoft, 2018).	4
Figur 2 Schema för element typen Vagg, med valda parametrar.	5
Figur 3 Projekteringsprocessens delprocesser (Hansson, o.a., 2015).	6
Figur 4 Markörbaserad förstärkt verklighet för ett hus (Augment, 2018).	9
Figur 5 fåtölj i markörlös förstärkt verklighet runt omkring fysiska modeller (Stichnoth, 2018).	10
Figur 6 Ljuset påverkar objektets skuggning i realtid, skript som är beroende av ljusintensitet kan skapas (Unity, 2018).	11
Figur 7 Visualisering av ARCore:s distinktioner och upptäckta punkter.	11
Figur 8 Modifikation av Figur 6 för tydliggörning av upptäckta punkter.	11
Figur 9 Frame spike som uppstår för en kort period vid aktivering av en funktion kan visualiseras av Profiler.	12
Figur 10 Schema för elementtypen Vagg, med valda parametrar.	15
Figur 11 Flödesschema för överföring från Revit till spelmotorn Unity.	16
Figur 12 Flödesschema för prestandaoptimeringsprocessen.	17
Figur 13 Användargränssnittet och numrering av huvudfunktioner.	18
Figur 14 Visualisering av list "parameterNamn" och "värde" för ett objekt i Unity Editor.	20
Figur 15 Exporterat schema från Revit i txt-format.	21
Figur 16 Flödesschema för integration av BIM i spelmotorn Unity.	21
Figur 17 Flödesschema för visualisering av BIM. Grön representerar funktion som körs i varje frame.	22
Figur 18 Textfältet visar data från schema för klickat objekt (grönt).	22
Figur 19 Flödesschema för lagervisningsväxlare.	25
Figur 20 Flödesschema för lagerknapp som växlar synlighet. Grön representerar funktion som körs i varje frame.	26
Figur 21 Lagervisningslistan, gul är synligt, transparent icke synligt, grå är delvis synligt.	26
Figur 22 Flödesschema för våningsväxlare. Grön representerar funktion som körs i varje frame.	27
Figur 23 Våningsplanväljare används för att kunna se andra våningsplanen.	27

Figur 24 Markören till vänster representerar alternativ 1, till höger är det alternativ 2.	28
Figur 25 Flödesschema för objektmarkör placering. Grön representerar funktion som körs i varje frame.	29
Figur 26 Flödesschema för ARCore. Grön representerar funktion som körs i varje frame.	30
Figur 27 Representation av "centerPoint" och dess lokala axel. Den övre, vänstra bilden är startpunkten.	32
Figur 28 Flödesschema för kamerastyrning i BIM-läge, pinch för zoom, pan för förflyttning och klick för position specificering	33
Figur 29 plus samt bockikon som används för positionering samt klart i förstärkt verklighet.	34
Figur 30 Flödesschema för kamerastyrning i förstärkt verklighet, pinch för zoom, pan för rotation och touch för objektplacering.	35
Figur 31 Röda 3D-modeller på smarta mobiltelefoner representerar positionen för fjärranvändare.	36
Figur 32 Lista på fjärranvändare som är tillgängliga för synkning.	37
Figur 33 Flödesschema för integration av synkron nätverkssynkronisering.	38
Figur 34 Flödesschema för realtidsskuggnings kontroll.	40

Beteckningar och begrepp

BIM	Building Information Modelling/Model
MEP	EI/VVS
AR	Förstärkt Verklighet, Augmented Reality
FPS	Bilder per sekund, "Frames per second"
SDK	Utvecklingsverktyg, "Software Development Kit"
APK	Applikationsprogram för Android

Inledning

Bakgrund

Kommunikationen mellan arkitekt och beställare är en avgörande faktor för ett lyckat projekt. En effektivare kommunikation leder till en lägre total kostnad, minskar risk för byggfel och störningar samtidigt som det leder till bättre tidshållning, ökad lönsamheten och bättre samarbetsklimat (Svensk Byggtjänst, 2016). En arkitekt kommunicerar oftast sina idéer med hjälp av planritningar som kan vara en utmaning att förstå, särskilt för oerfarna beställare. Därför produceras fotorealistiska renderingar, animationer och filmer som ger beställaren en bättre förståelse, dock från en vy bestämd av arkitekten som leder till att beställaren inte får en fullständig förståelse över byggnaden (Kitchens & Shiratuddin, 2007). Trots att dessa visualiseringsmetoder är mycket användbara så räcker de oftast inte i större och mer komplexa projekt, och därför skapar man även fysiska modeller. Målet med dessa är att få en bättre uppfattning om rumsstrukturen, designs och allmän skala (Kitchens & Shiratuddin, 2007). Produktion och transport av fysiska modeller kräver tid, kostnad, utrymme och oftast även yrkesverksamma engagemang, till följd av det byggs fysiska modeller oftast i sena skeden av projektering (Kitchens & Shiratuddin, 2007).

Förstärkt verklighet (Augmented Reality, AR) ger möjligheten till att producera och visa datorgenererade modeller samt digital information framför tittaren i samma fysiska värld. Detta gör det möjligt att kunna få nya uppfattningar i form av syn, hörsel och känslor som tidigare inte var möjliga. Förstärkt verklighet gör det möjligt för en beställare att se deras drömhus från olika perspektiv precis som om det var färdigt byggt i den verkliga världen, eller förflytta byggnaden i tomtmarken. (Craig, 2013).

Förstärkt verklighet baserat på BIM-modeller och genom smarta mobiltelefoner har potentialen att ersätta fysiska modeller med syftet att skapa mer ekonomiska och effektivare designprocesser. Detta tack vare den snabba utvecklingen som sker i mobilteknologin som gör det möjligt att köra allt mer komplicerade 3D-modeller och beräkningar på smarta mobiltelefoner. Den stora tillgängligheten av smarta mobiltelefoner gör det möjligt för många att kunna använda sig av teknologi för förstärkt verklighet. Enligt (Davidsson & Thoresson, 2017) äger 85 procent av Sveriges befolkning en smart mobiltelefon.

BIM, Building Information Modelling, är en informationshanteringsmetodik för samverkan mellan alla parter inom konstruktion, integrerande design och underhåll. Detta sker genom en digital beskrivning av ett byggprojekt där informationen är baserad på objektsorienterade modeller. BIM möjliggör automatisering av många processer inklusive simuleringar, beräkningar och tidsplanering.

Spelmotorer som effektivisering vid spelutveckling har generellt hög renderingskapacitet och som möjliggör realtidsrendering, vilket leder till att de blir användbara inom arkitektur. Spelmotorer har även möjligheten till uppbyggande av nätverkssystem där syftet är skapandet av flerspelarspel "multiplayer game". Dessa teknologier kan användas för skapandet av distribuerade synkrona kommunikationsverktyg inom projektering och planering.

Syfte och mål

Syftet med denna studie är att utveckla en mjukvara, med hjälp av förstärkt verklighet och spelmotorer, för visualisering och informationsöverföring mellan arkitekten och beställaren i ett distribuerad möte. Genom att utveckla metoder för ett fleranvändarsystem som nyttjar internet för koppling mellan användare kan spelmotorn möjliggöra dataöverföring via internet samtidigt som arkitektoniska analyser kan tillämpas genom skuggstudier i realtid, förstärkt verklighet, BIM visualisering och planlösningar.

Målet är att utveckla en demonstrator som visar på möjlighet att:

- Tillgängliggöra för parter att få tillgång till samma BIM genom distansbaserad teknologi.
- Möjliggöra information visualisering via distansbaserad teknologi och förstärkt verklighet.

Syftet med demonstratorn är att skapa en fungerande prototyp av mjukvaran som visar på konceptet samt möjligheten för framtida mjukvaror.

Automatisering av processen för skapandet av mjukvaran i spelmotorn är ett av syften, där målet med detta är minimering av potentiell påverkan på befintliga arbetsflöden som existerar då mjukvaran inte används.

Forskningsfrågor

För att identifiera behoven, kraven och möjligheten som behövs för att uppnå syftet med skapandet av en demonstrator som kommunikationsverktyg mellan arkitekt och beställare har tre forskningsfrågor valts ut för undersökning.

- Hur kan spelmotorer möjliggöra förstärkt verklighet och nätverkssynkroniserad distribuerad kommunikation?
- Hur kan metoden för en förstärkt verklighets miljö beskrivas för distribuerad möte?
- Hur används nätverkssynkronisering i metoden för förstärkt verklighet?

Avgränsning

Detta examensarbete avgränsar sig genom att fokusera på endast ett multifunktionellt höghus som är ritat av en erfaren arkitekt. Avgränsning sker också på enheten som den förstärkta verkligheten körs i, där endast smarta mobiltelefoner tas hänsyn till. Detta på grund av den stora tillgängligheten till smarta mobiltelefoner för allmänheten som enligt Davidsson & Thoresson (2017) ligger på 85 procent. Dessutom är majoriteten vana vid användningen av smarta

mobiltelefoner i jämförelse med andra enheter såsom Microsoft HoloLens och Google Glass. Kommunikationen avgränsas genom att endast ta hänsyn till distribuerad kommunikation, detta för att visa distribuerade synkrona kommunikationsmetoders potential i områden där traditionella metoder brister på.

Förväntade resultat

Det är förväntat att denna mjukvara ska stödja traditionella hjälpmedel som är bristande i distanskommunikation. Detta genom att säkerställa att alla parter har tillgång till samma data, dokumentversioner och 3D-modell samt göra det möjligt att kommunicera vidare sina åsikter, idéer och problem via fleranvändarsystemet.

Med hjälp av förstärkt verklighet och smarta mobiltelefoner ska man kunna ta med sig modellen samt alla nödvändiga data var man vill utan att behöva transportera något extra med sig. Ändringar som arkitekten utför kan enkelt föras vidare till beställaren. Det är förväntat att denna mjukvara kommer resultera i minskad förvirring och missförstånd mellan arkitekter och beställare.

Förväntade problem

Förstärkt verklighet-teknologin som används i det här arbetet, ARCore, kräver operativsystemet Android 7.0 (Nougat) eller nyare och har en begränsad lista av enheter som stöds officiellt av Google (Google, 2018). Detta innebär att Androids befintliga smarta mobiltelefoner som inte har tillgång till Android 7.0 inte kommer att kunna utnyttja sig av ARCore. Detta är dock ett temporärt problem då andelen användare av Android 7.0 eller nyare ökar.

Prestandaproblem är förväntade under utvecklingskedet, mest troligt kommer dessa att bero på inställningar i spelmotorn samt olika skript. Detta kan minimeras genom optimering av 3D-modellen, användning av felsökningsverktyg i spelmotorn och prövning vid förändring av skript.

På grund av att ARCore är en relativt ny teknologi med begränsad kompatibilitet till Androids smarta mobiltelefoner förväntas fel uppstå vid användningen av förstärkt verklighet.

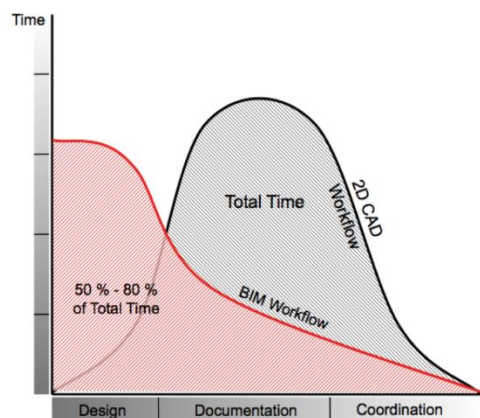
Teori

BIM

BIM, Building Information Modelling, är en metodik för integrerande design, konstruktion och underhåll av ett projekt i samverkan med alla parter. Detta utförs genom digitalt datautbyte som resulterar i en komplett digital beskrivning av ett byggprojekt. Informationshantering är baserad på objektorienterade modeller istället för linjer. Där egenskaper kan beskriva ett objekts dimensioner, material med mera. Objektets geometri beskriver hur objektet ser ut och status kan beskriva om objektet är beställt eller granskat med mera. (BIM Alliance Sweden, 2017).

BIM kan användas i många olika typer av byggnationsprojekt som till exempel broar, vägar, tunnlar, byggnader eller även komplex av byggnader där det är tillgängligt under hela byggprojektets livscykel. (openBIM, 2011; Hansson, o.a., 2015).

Många fördelar finns med integration av BIM i ett projekt. BIM-användning kan leda till färre felkostnader genom användning av effektivare projektstyrningsverktyg till exempel kollisionskontroll, som är svårt att upptäcka i CAD ritningar. Då BIM är objektorienterat och har möjligheten att ta in all nödvändig data kan simuleringar, beräkningar, tidsplanering och ekonomistyrning ske utav en basmodell. Detta istället för att behöva producera nya modeller för varje unik simulering, vilket garanterar att informationen blir konsekvent. BIM gör det möjligt för automatisk uppdatering av simuleringar, beräkningar, tidsplanering och ekonomi när basmodellen ändras, vilket resulterar i att problem och fel upptäcks effektivare. Ett effektivare arbetsflöde öppnar möjligheten för byggande av mer avancerade och korrekt utförda byggnationer, se Figur 1. (Hansson, Olander, Landin, Aulin, & Persson, 2015).



Figur 1 Produktivetsflöde vid användning av BIM i jämförelse med 2D CAD (Graphisoft, 2018).

BIM är ett kraftfullt verktyg för kommunikation med beställaren, att kunna se byggnaden innan den är byggd kan skapa medverkan i processen vilket är viktigt för att säkerställa att beställaren är nöjd (BIM Alliance Sweden, 2017).

BIM mjukvara

Exempel på BIM-mjukvaror är Revit, utvecklat av Autodesk, samt ArchiCAD, utvecklad av Graphisoft, vilka används inom arkitektur, konstruktion, el och VVS. BIM-mjukvaror är objekt- och elementorienterat och består av till exempel väggar, dörrar och fönster. Där projektens utföring och visualisering kan ske i flera olika vyer, inklusive 3D-modellvy, planvyer och sektionsvyer (Autodesk, 2018).

Dessa mjukvaror har parametriska egenskaper som lagrar objekts- och elementsrelaterad information i en enda databas, inklusive mått, material och absorptionsförmåga. Dessa kan användas inom tillämpningar bland annat visualisering, kostnadsberäkningar, funktionalitet samt simuleringar. För många objekt och element finns det redan befintliga standardparametrar men det är också möjligt att tillägga egna parametrar. Lagringen av parametreringsinformation i en enda databas möjliggör automatisk uppdatering vid någon ändring överallt i ett projekt, inklusive modell vy, ritningar, tabeller, förteckningar, sektioner och planer. Detta leder till att arbetsflödet blir mer effektivt och därför även mer ekonomiskt gynnsamt. (Jeppsson CAD Center, 2014).

Schema eller "Schedule" är en tabellformad visningsmetod som oftast är tillgängligt med BIM. Informationen extraheras från befintliga elements och objekts parametriska egenskaper till ett schema. Scheman för en eller flera typer av element och objekt kan skapas, se Figur 2. Scheman uppdateras automatiskt när en ändring sker som är relaterad till själva scheman. (Autodesk, 2018).

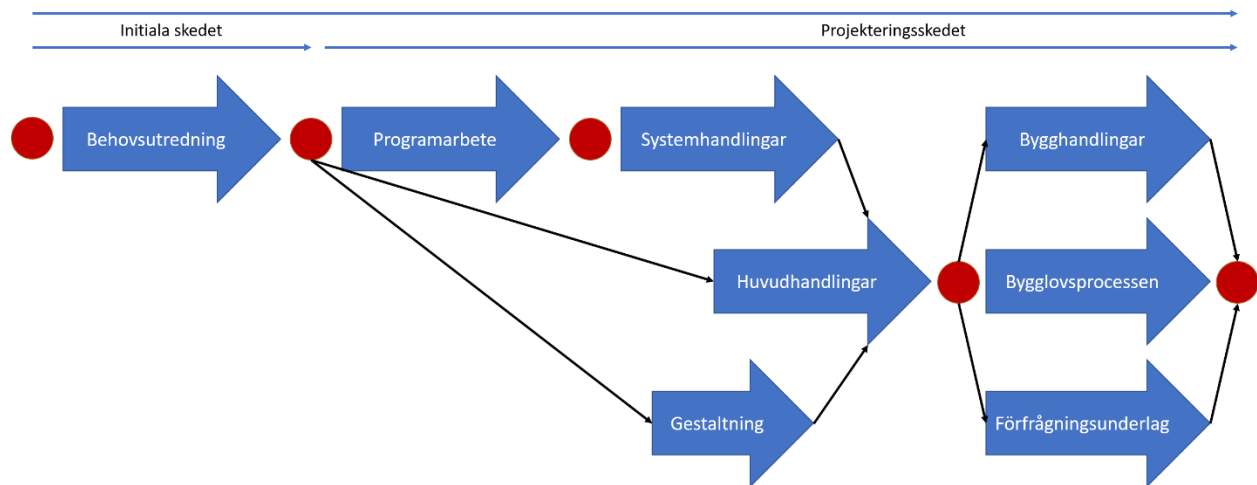
För att BIM mjukvaror ska kunna identifiera och skilja mellan olika element används Element ID. Element ID är en sifferkombination som är unik för varje element. Denna sifferkombination ändras aldrig och kan presenteras i ett schema. (Autodesk, 2014).

<Wall Schedule>									
A	B	C	D	E	F	G	H	I	J
Element ID	Level	Base Offset	Function	Length	Width	Height	Phase Created	Structural Usage	Type
303287	Level 0	0.00	Exterior	3.00	0.30	3.50	New Construction	Bearing	Col 0.3
303288	Level 0	0.00	Exterior	2.90	0.30	3.50	New Construction	Bearing	Col 0.3
303289	Level 0	0.00	Exterior	2.90	0.30	3.50	New Construction	Bearing	Col 0.3
303290	Level 0	0.00	Exterior	3.00	0.30	3.50	New Construction	Non-bearing	Col 0.3
303291	Level 0	0.00	Exterior	2.90	0.30	3.50	New Construction	Non-bearing	Col 0.3

Figur 2 Schema för element typen Vägg, med valda parametrar.

Projektering

Projekteringsprocessen börjar med en idé och ett behov som kan lösas av en byggnad och processen avslutas med att behovet i detalj är definierat av ritningar och beskrivningar. Projekteringsprocessen delas upp i två skeden, det initiala skedet och projekteringskedet, se Figur 3. Det initiala skedet, även kallat tidiga skedet, har behovsutredning och programarbete som delprocesser, det är här utredning och precisering av behov, tidskrav och preliminär budgetering sker. (Hansson, Olander, Landin, Aulin, & Persson, 2015).



Figur 3 Projekteringsprocessens delprocesser (Hansson, Olander, Landin, Aulin, & Persson, 2015).

I projekteringskedet utformas och ritas handlingarna till en byggnad. Gestaltningsskedet är centralt i projekteringskedet på grund av att arkitekten behöver ta fram förslag till planer och fasader. Detta sker oftast genom skissning där de olika alternativen studeras och identifieras, utifrån dessa upprättas förslagshandlingar. Övriga konsulter kan då börja med sina insatser. Arbetsflödet i projekteringsprocessen är iterativt för att möjliggöra förbättringar och anpassning efter nya krav och ändringar som beställare eller andra aktörer begär. (Hansson, Olander, Landin, Aulin, & Persson, 2015).

Projekteringen och arkitektens kostnad under tidiga skeden är liten i förhållande till totala investeringskostnaden. Detta i kombination med de stora möjligheter som dessa aktörer har att påverka utformningen motiverar beställaren att satsa mer för att säkerställa slutresultatet. Detta är viktigt för att kunna identifiera fel i tidiga skeden då kostnader för att åtgärda fel är lägre i tidiga skeden av projekt i jämförelse med sena skeden. (Hansson, Olander, Landin, Aulin, & Persson, 2015).

Arkitektonisk kommunikation med Collaborative design

Collaborative design involverar flera aktörer med olika kunskap och erfarenhet som vill uppnå ett gemensamt mål eller intresse som en individ inte kan uppnå ensam. Collaborative design används huvudsakligen i gestaltningskedet men är användbart i andra skeden. Collaborative design, oavsett om det sker co-located eller distribuerat "distansmöte", går igenom cykler av individuellt arbete och samverkan. (Törlind, 2015).

"The biggest problem with communication is the illusion that it has occurred. We think when we express ourselves that, because we generally understand what we think, the person that we're expressing it to generally understands it in the same way. When you're creating something, you have to recognize that it's the interaction that will allow everybody to come to a fundamental understanding of what it's supposed to do, how it's going to be made. We should always be striving to have an environment that allows those interactions to happen." - (Sabbagh, 1996).

Co-located arkitektonisk kommunikation

Co-located kommunikation, möte ansikte mot ansikte, är den mest naturliga kommunikationsmetoden som inte är begränsad av teknologi. Denna metod kombineras och integreras med många kommunikationsverktyg inklusive skissning, fysiska modeller, dokument, renderingar och ritningar. Skissning, utförs även digitalt nuförtiden tack vare utvecklingen av pekskrävar och aktiva stylus pennor. (Törlind, 2015).

Projektet börjar med enkla skisser som definieras och utvecklas vidare under projektets genomgång till mer utvecklade förslag (Törlind, 2015). I dagsläget förses beställaren oftast med information i form av presentationer där arkitekten väljer vilken information som ska visas (Edwards, Li, & Wang, 2015). Detta leder till att beställaren får tillgång till information som arkitekten vill visa vilket inte nödvändigtvis är det som beställaren är intresserad av, beställaren får då en begränsad uppfattning om byggnaden som leder till att dennas bidrag i projekteringsprocessen begränsas (Edwards, Li, & Wang, 2015).

Vid realtidsrendering används en virtuell kamera som kan röras fritt i en virtuell värld. Den virtuella kameran omvandlar kontinuerligt en 3-dimensionell virtuell miljö till 2-dimensionella bilder. Antalet renderade bilder per sekund, FPS, behöver vara tillräckligt hög för att realtidsrendering ska bli smidigt och interaktivt. Ett lågt FPS-värde minskar känslan av kontinuerlig rörelse. En god realtidsrendering ger användaren känslan av rörelse som om det är den fysiska världen. (Johansson, 2016).

I mer komplexa och stora projekt används även fysiska modeller (Mateus, o.a., 2015). Fysiska modeller är viktiga i tidiga skeden för att skapa en gemensam uppfattning av projektet (Törlind, 2015). Fysiska modeller är statiska och kan vara resurskrävande beroende på designen. En ombearbetning på fysiska modeller kan medföra stora kostnader och förseningar (Maing, 2012). Därför uppbyggs fysiska modeller ofta i slutet av ett projekt och används oftast för

marknadsföring. Storleken av fysiska modeller kan leda till transportproblem som i sin tur leder till svårigheter med att använda dem.

Distribuerad arkitektonisk kommunikation

Co-located kommunikation har många fördelar, inklusive möjligheten till interaktion med hjälp av skrivtavlor, skisser, bärbara datorer, utskrivna dokument, fysiska modeller och även kroppsliga språket (Törlind, 2015). Dock är det inte möjligt att utföra alla möten ansikte mot ansikte och därför är det viktigt med effektiva distribuerade kommunikationsmetoder.

Distribuerad kommunikation används då olika aktörer av ett projekt inte befinner sig i närheten av varandra och inte kan hålla co-located möten. Då används ofta internetbaserade kommunikationsmetoder (Törlind, 2015). Distansen mellan olika aktörer spelar roll för kommunikation, där en högre distans leder till att mindre kommunikation sker och kommunikationen minskar så hastigt att även ett avstånd på 30 meter är nästan samma sak som flera mil (Allen, 1984).

Målet med distribuerad kommunikation är att kunna återskapa det fysiska mötet i ett virtuellt möte över internet. Videokonferens och telefonkonferens är lösningar som ofta används för kommunikation. De kombineras ofta med mjukvaror som möjliggör presentation och webbkonferenser, till exempel Adobe Connect. Problemet med denna lösning är att den inte är anpassad till behoven som alla olika aktörer kan ha i en projekteringsprocess. Särskilt i gestaltningsskedet blir detta tydligt, där mycket av kommunikationen sker genom skissning, skrivtavlor, utskrivna dokument, olika renderingar och fysiska modeller. (Törlind, 2015).

I distribuerad kommunikation sker distinktion mellan synkront och asynkront. Synkron kommunikation är kommunikation som sker omedelbart mellan olika aktörer som inte befinner sig i närheten av varandra, med asynkron kommunikation sker kommunikationen istället vid olika tidpunkter (Hrastinski & Keller, 2007). För att kunna effektivt samarbeta och utnyttja aktörernas kompetens behöver distribuerade möten ske så nära till ett synkront möte som möjligt (Törlind, 2015).

Skärmdelning är ett synkront kommunikationsverktyg som används oftast i ett distribuerat möte. I skärmdelning tittar alla aktörer på samma vy, metoden är tillräckligt god för presentation dock begränsar den möjligheten för interaktion. (Törlind, 2015).

Objektdelning är ett synkront kommunikationsverktyg som låter olika deltagare vara delaktiga i samma projekt samtidigt genom exempelvis delning av dokument med hjälp av Google Docs eller delning av 3D-modeller. Skillnaden med skärmdelning är att i denna metod har varje användare möjligheten till att välja sin egen vy och även interagera interaktivt med 3D-modellen. Denna metod används allmänt mycket, dock är den fortfarande inte väl integrerad i arkitektoniska sammanhang. (Törlind, 2015).

Förstärkt verklighet (AR)

Förstärkt verklighet är en visualiseringsmetod som möjliggör visualisering av datorgenererade modeller, inklusive bilder, videor och 3D-modeller för användaren som om de är i den fysiska världen istället för en virtuell värld (Zhou, Duh, & Billinghurst, 2008). Förstärkt verklighet strävar efter att förenkla livet med hjälp av virtuell information som man får tillgång till utan att kopplas bort från den fysiska världen, detta möjliggör ökad uppfattning och interaktion (Furht, 2011). Tillgången till portabel förstärkt verklighet ökar, idag är det tillgängligt i smarta mobiltelefoner, surfplattor och förstärkt verklighet head mounted smartglasses såsom Microsoft HoloLens (Heimgartner, 2016).

Som nämndes tidigare, så är det arkitekten som väljer data och vyer som presenteras till beställaren vilket är en begränsande faktor för beställarens uppfattning. Förstärkt verklighet kan leda till ett naturligare kommunikationsflöde. I en co-located kommunikation kan förstärkt verklighet kombinera den fysiska världen med den virtuella världen för att kunna interagera mellan fysiska och virtuella objekt vilket möjliggör ökad förståelse mellan aktörer. Förstärkt verklighet kan förbättra en distribuerad kommunikation via användande av 3D-ljudsverkyg samt användande av visuella signaler som representerar dem övriga aktörer. (Billinghurst & Kato, 2002).

Markörbaserad förstärkt verklighet

Markörer är utskrivna bilder som kan vara från en simpelt svartvit bild till ett komplex planlösning. Markörens skala, position och rotation relativt till den virtuella 3D-modellen är känd av en mjukvara som när den upptäcker markören via en kamera placerar den valda virtuella 3D-modellen i skärmen relativt till upptäckta markören, se Figur 4. Förflyttning och rotering av markören påverkar virtuella modellen i realtid. Denna metod är enkel att använda och relativt problemfri, dock behöver man ha markörbilden med sig konstant. (Prabhu, 2017).



Figur 4 Markörbaserad förstärkt verklighet för ett hus (Augment, 2018).

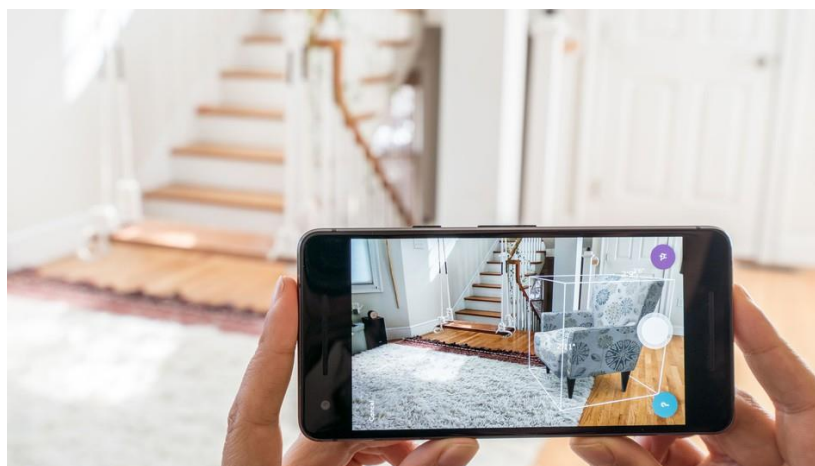
Markörlös förstärkt verklighet

Markörlös förstärkt verklighet, se Figur 5, använder sig av olika lokalisering och kartläggningstekniker som använder sig av sensorer såsom kamera och accelerometer i kombination med algoritmer. Markörlös förstärkt verklighet används i ARKit och ARCore som är utvecklade för iPhone respektive Android smarta mobiltelefoner. (Prabhu, 2017).

SDK, Software Development Kit, är ett programutvecklingsverktyg som möjliggör skapandet av mjukvaror som är baserade på ett system, hårdvara eller även en mjukvara (Shamsee, o.a, 2015). ARCore SDK är ett fritt tillgängligt verktyg som möjliggör utvecklingen av mjukvaror som baserar sig på markörlös förstärkt verklighets teknologin, ARCore, för Android plattformen (Google, 2018). ARCore använder sig av flera olika teknologier för att kunna integrera det virtuella med den fysiska världen som användaren ser genom smart mobiltelefonens kamera (Google, 2018).

ARCore använder sig av rörelsespårning "Motion tracking". Genom smarta mobiltelefonens kamera upptäcker ARCore visuella distinktioner och placerar punkter på respektive distinktion, se Figur 7 och Figur 8. Punkterna i kombination med rörelsesensorerna beräknar positionsändring vid rörelse. Om ARCore hittar en grupp av punkter som ligger i en gemensam horisontell yta, till exempel ett bord eller golv så skapar den ett virtuellt plan som man kan placera 3D-modellen på. Punkterna uppdaterar sig konstant då ARCore får ny information vid rörelse som leder till att skapade planens precision ökar konstant. På grund av precisionsökning används ankare på 3D-modellen som ser till att den anknyts till ett plan. ARCore använder sig också av ljus uppskattningsverktyg, genom bilderna från kameran kan ARCore få en uppskattning på både intensitet och riktning av ljus som den då använder för att modifiera ljuset i realtid, se Figur 6. (Lanham, 2018).

ARCore:s beroende av kameran leder till att den inte detekterar ytor utan texturer lika bra relativt till ytor med texturer samt att den inte detekterar alls transparenta ytor inklusive glasbord. (Mira, 2018).



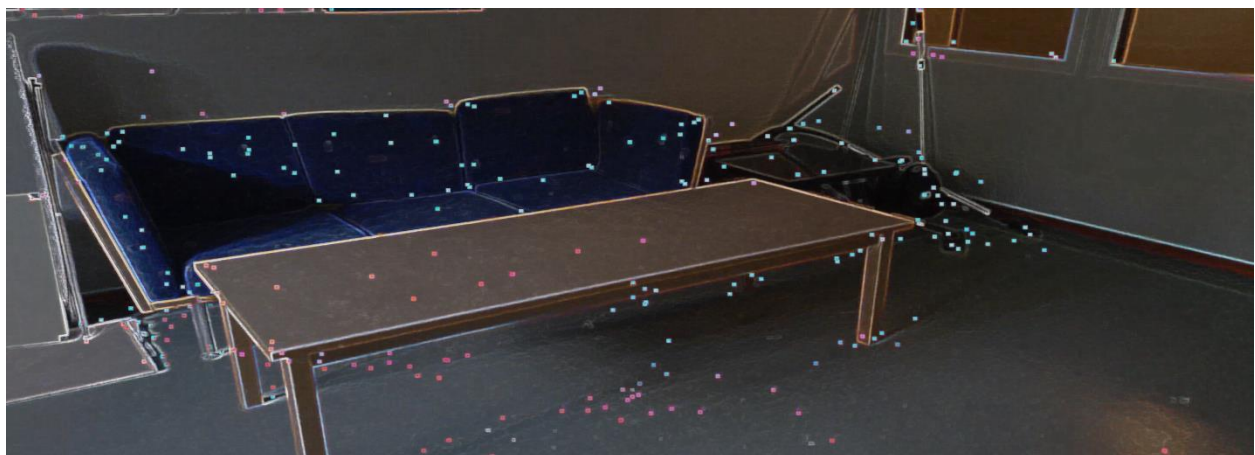
Figur 5 fätölj i markörlös förstärkt verklighet runt omkring fysiska modeller (Stichnoth, 2018).



Figur 6 Ljuset påverkar objektets skuggning i realtid, skript som är beroende av ljusintensitet kan skapas (Unity, 2018).



Figur 7 Visualisering av ARCore:s distinktioner och upptäckta punkter.



Figur 8 Modifikation av Figur 6 för tydliggörning av upptäckta punkter.

Spelmotor

Spelmotorer är mjukvaror som är utvecklade och utformade för effektivisering av arbetsflödet för spelutveckling. Spelmotorer inklusive Unity och Unreal Engine 4, har många funktioner inklusive hög presterande renderingskapacitet, fysiksystem, nätverkssystem och skriptning.

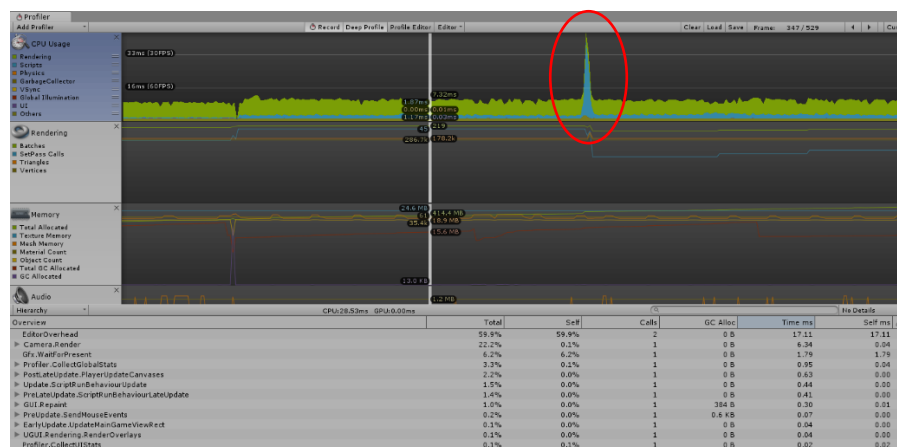
Renderingskapaciteten möjliggör realtidsrenderingar/visualiseringar av byggnaden samt ljussättning och skuggning. Fysiksystemet möjliggör utförande av avancerade simuleringar där Rüppel & Schatz (2011) har utförd simuleringar för brandevakuering med hjälp av spelmotorer. Nätverkssystemet möjliggör distribuerad kommunikation både genom synkron och asynkron informationssynkning som kan inkludera positionsdata och statussynkning. Skriptning möjliggör skapande av interaktiva funktioner och spellogik, detta kan användas till exempel för rörelsespårning och ytdetektion som sker vid användning av förstärkt verklighet.

Unity kan bygga mjukvaror för flera olika plattformar, detta innebär att man oftast bara behöver ett projekt som kan byggas vidare till många olika plattformar inklusive Windows, Android och IOS (Unity, 2018). Både C# och JavaScript fungerar för skriptning i Unity.

Prestanda optimering

Optimering av mjukvaror är särskilt viktigt för smarta mobiltelefoner, där en icke väl optimerad mjukvara kan leda till sämre batteritid, högre temperaturer och kan leda till låga värden på FPS i form av lag och "frame spikes" som kan ses i Figur 9. 60 FPS är målet som man vill uppnå konstant, dock ses 30 FPS ofta som acceptabelt. (Unity, 2018).

För att hitta flaskhalsen för en mjukvaras prestanda i Unity används ett inbyggt verktyg som kallas för Profiler. Profiler mäter många olika aspekter som kan påverka prestanda i realtid, där det är möjligt att få detaljerad information om var flaskhalsen är, som oftast är relaterat till antingen processorn eller grafikprocessorn. Det är också möjligt att veta om det är ett specifikt skript eller funktion som är orsaken till prestandaproblem. Beroende på problemet kommer olika åtgärder behöva vidtas, Profiler är ett viktigt verktyg för att identifiera problem och säkerställa att dem är lösta. (Unity, 2018; Schell & Turner, 2017).



Figur 9 Frame spike som uppstår för en kort period vid aktivering av en funktion kan visualiseras av Profiler.

Metod

Mjukvaruutvecklingen har genomförts genom distribuerad kommunikation, där tre deltagare var till hjälp i utveckling och prövning. Mjukvaran har kunnat testas emellan olika orter, där två deltagare befann sig i Stockholm, medan den tredje befann sig i Linköping. Detta har även gjort det möjligt att testa prestanda i olika enheter. Första deltagaren som befinner sig i Stockholm är en arkitekt med över 25 års erfarenhet, det är också från denna arkitekt som ritningsmaterial tagits. Andra deltagaren, som också befinner sig i Stockholm, har arbetat och varit lärare inom design. Tredje deltagaren befinner sig i Linköping och har inte erfarenhet relaterad till byggande eller arkitektur. Totalt har mjukvaran testats vid 64 olika tillfällen med tredje deltagaren, där 7 inkluderade första deltagaren och 2 även den andra deltagaren. Testen gjordes i syfte att samla kritik från deltagarna och således utveckla mjukvaran. Många av testförsöken i tidiga skeden utfärdades i samarbete med tredje deltagaren då denne hade tillgång till smart mobiltelefon som kunde använda sig av ARCore.

Litteraturstudie

Litteraturstudie har utförts för att skapa en teoretisk grund om projektering, förstärkt verklighet, BIM och collaborative design. Litteraturen baseras på böcker och forskarartiklar i kombination med presentationsvideor. Sökandet av litteratur, böcker samt artiklar genomfördes via Luleå tekniska universitetsbiblioteket (Libris) och Google Scholar. Förstärkt verklighet samt prestandaoptimering kompletterades via presentationsvideor från Unite Austin Live som är en årlig konferens för och av utvecklare. Litteraturen har delvis valts utifrån rekommendationer från professorer inom ämnen som är relevanta för det här arbetet. Källor relaterade till arbete och arbetsflöden i Unity har valts utifrån trovärdigheten hos författaren samt huruvida de uppfyllt de eftersökta funktionerna.

Intervju

Intervjuerna har genomförts som semistrukturerade intervjuer. För utförande av semistrukturerad intervju uppsätts ett intervjutema med möjligheten till varierande ordning på de frågor som ställs, detta möjliggör insamlingen av kvalitativa data (Saunders, Lewis, & Thornhill, 2009). För att få så mycket information som möjligt utav testpersonerna är det viktigt att intervjun inte leds av intervjuaren (Hedin & Martin, 2011). För att säkerställa att intervjun inte leds, skickas först versionen på APK (mjukvaran i fråga) till testpersonerna tillsammans med en förklaring av nyligen adderade samt modifierade funktioner. Testpersonerna testar mjukvaran och sedan utförs intervjuer i form av mobilsamtal men även via meddelanden och inspelade videor, som utfärdats av testpersonerna, i vilka dem beskriver deras åsikter. För prestandaproblem behövdes dock oftast collaborative kommunikation.

Syftet med intervjun har varit insamlingen av kvalitativa data utifrån perspektiven av både erfarna och icke erfarna personer inom byggande och arkitektur. Kvalitativa data inkluderade fördelar, nackdelar, problem och utvecklingsmöjligheter för varje funktion och dess process.

Deltagare 1 och 3 har varit med sedan början av projektet, dessa deltagare har valts då de har relevant erfarenhet, äger smarta mobiltelefoner som kan köra mjukvaran samt geografisk distans dem emellan. Deltagare 2 kom med i slutet av projektet för att få ett mer objektiva perspektiv på kombinationen och samspelen av alla konstruerade funktioner.

Hårdvara

Mjukvaruutvecklingen samt prövningen av mjukvaran utfördes i en stationär dator som behövde klara körningen av spelmotorn Unity samtidigt som minst 3 mjukvaror byggda i Unity körs, detta för att kunna testa nätverks- och distribuerade kommunikationssystemet. Stationära datorn har följande specifikationer:

- Operativsystem: Windows 10 Pro
- Processor: AMD Ryzen 7 1700
- Grafikkort: GeForce GTX 1080 Ti
- RAM: 16 GB

För att testa distribuerade kommunikationssystemet i förstärkt verklighet mellan fyra olika aktörer har följande smarta mobiltelefoner använts:

- Samsung Galaxy S9 Plus (Android 8.0)
- Samsung Galaxy S8 (Android 8.0)
- Samsung Galaxy S7 edge (Android 7.0)
- Samsung Galaxy S7 (Android 7.0)

BIM via Autodesk Revit

För modellering lånades en förslagsritning för ett punkthus som är utförd i AutoCad, av en arkitekt. Förslagsritningen används som ritningsmall i Revit istället för att planera och rita ett punkthus från början vilket gör det möjligt att utföra modelleringen snabbare.

Modellering i Revit utfördes med en standard arkitektonisk mall. Parametriska egenskaper för dem olika typerna av väggar, fönster, dörrar och golv skrevs och kontrollerades. Detta inkluderade parametrar för:

- Strukturell användningen, bärande eller icke bärande.
- Material med deras egenskaper samt texturer.
- Funktionalitet, exteriör eller interiör
- Bredd, längd och tjocklek
- Operationstyp, som beskriver gångjärnplacering

Modellering av entréplanen samt förstavåningen utfärdades med hjälp av olika typer av element och ritningsmallen. Därefter kopierades förstavåningen vidare till och med tionde våningen. Efter färdigställning av byggnadsmodellen skapades en ny parameter som innehåller identitetsnummer för alla element. Detta kan fyllas in manuellt, dock är det tidskrävande, i detta fall skulle det ha behövts göras för 2500 element och därför användes istället tilläggsprogrammet ShowID som automatiserar denna process.

För att överföra parametriska data till spelmotorn Unity skapas scheman för kategorier av elementtypen vägg, fönster, dörr och golv som innehöll nödvändiga parametriska data, se Figur 10.

<Wall Schedule>									
A	B	C	D	E	F	G	H	I	J
Element ID	Level	Base Offset	Function	Length	Width	Height	Phase Created	Structural Usage	Type
303287	Level 0	0.00	Exterior	3.00	0.30	3.50	New Construction	Bearing	Col 0.3
303288	Level 0	0.00	Exterior	2.90	0.30	3.50	New Construction	Bearing	Col 0.3
303289	Level 0	0.00	Exterior	2.90	0.30	3.50	New Construction	Bearing	Col 0.3
303290	Level 0	0.00	Exterior	3.00	0.30	3.50	New Construction	Non-bearing	Col 0.3
303291	Level 0	0.00	Exterior	2.90	0.30	3.50	New Construction	Non-bearing	Col 0.3

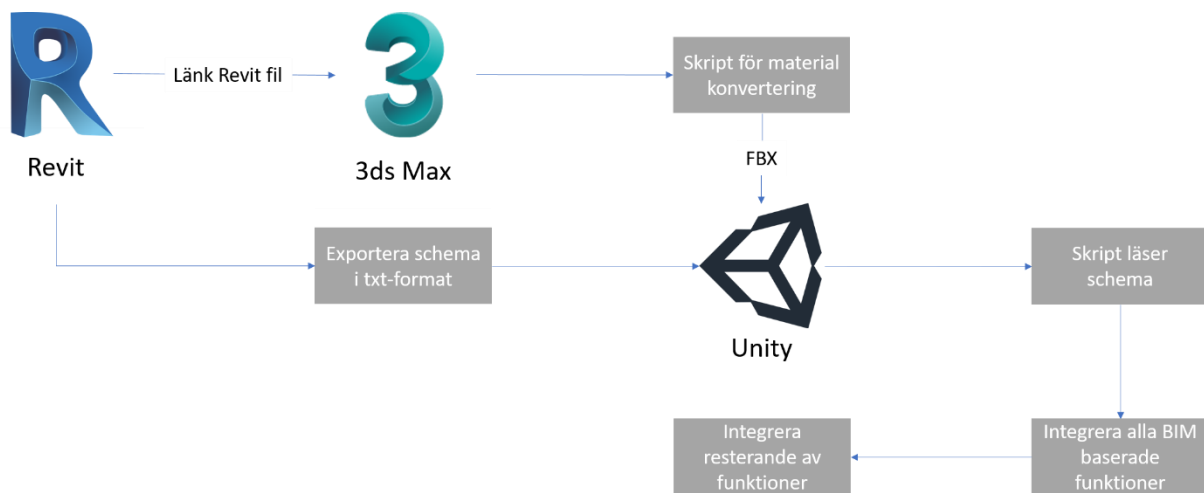
Figur 10 Schema för elementtypen Vägg, med valda parametrar.

Spelmotor

Överföringsflöde till spelmotor

En Revit-modell kan inte överföras direkt till Unity som en FBX-fil, detta på grund av att Revit:s material och texturer inte är kompatibla i Unity vilket resulterar i att materialen inte syns alls i Unity. För att gå runt detta används en överföringsmetod, se Figur 11. Överföringen börjar genom att först importera/länka Revit-modellen till 3ds Max där inställningen "utan att kombinera element" väljs och därefter används ett skript i 3ds Max för materialkonvertering. I detta fall används ett skript som är skrivet av Jani Mukkavaara, doktorand på Luleå tekniska universitet. Därefter importeras modellen vidare från 3ds Max till Unity. (Liu, o.a., 2016; Callewaert, 2013).

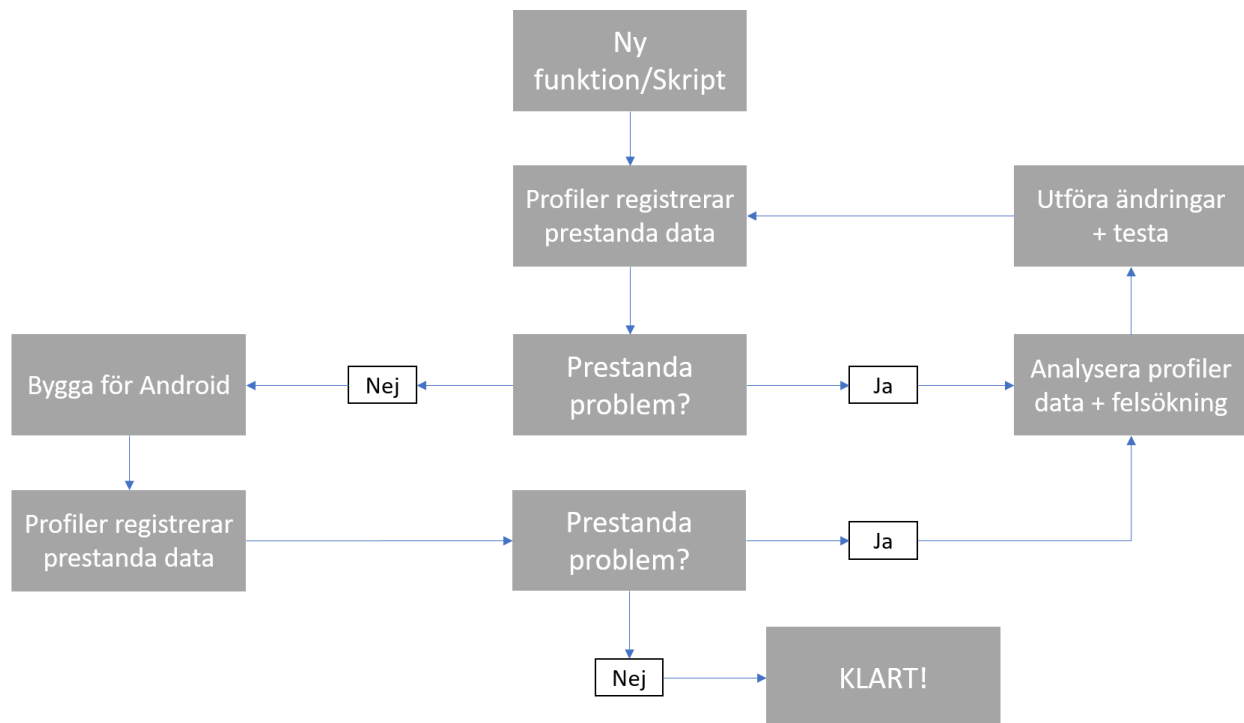
Skapade scheman i Revit exporteras i form av ett txt-fil vidare till Unity, där ett skript automatiskt läser txt-filer och sorterar informationen så att det blir möjligt att automatiskt skapa interaktiva BIM-funktioner i spelmotorn Unity, detta arbetsflöde är inspirerat av Boeykens (2014).



Figur 11 Flödesschema för överföring från Revit till spelmotorn Unity.

Arbetsflöde för prestandaoptimering

För att säkerställa god prestanda så har alla huvudfunktioner samt skript testats med denna metod som nyttjar Unity:s skriptgranskare, Profiler, för att hitta möjliga prestandaproblem och/eller utvecklingsmöjligheter, se Figur 12. Profiler sätts i inspelningsläge samtidigt som olika funktioner och skript testkör. När problem upptäcks så utförs en analys av tillgängliga data från profiler samt felsökningar vilket oftast sker genom användande av debug.log som kan visualisera problemen genom meddelanden skickade till Unity:s konsol. Lösningen och ändringen varierar från problem till problem som därefter testas igen genom profiler, detta är en sluten slinga tills problemet är löst. När problemet är löst testas det istället i Android smarta mobiltelefoner som oftast har visat nya problem och då används samma metod återigen för att upptäcka och lösa problemen.



Figur 12 Flödesschema för prestandaoptimeringsprocessen.

Mjukvaruutveckling

Vid test av mjukvaran observerades användbarheten för funktioner samt användargränssnittet och diskussion fördes med testpersonerna. Testen har utförts i en liten grupp, därför är det troligt att resultaten kommer att variera om arbetet utförs igen.

Detta var första gången för alla deltagare att testa förstärkt verklighet, det var intresseväckande att pröva och efter utvecklingen av knappar samt ikoner blev navigationen enklare och naturligare. Deltagarna hade inga problem med att navigera sig omkring på samma gång som man studerade byggnaden.

Unity kan vara en komplex mjukvara, skriptning i C# kan göra det ännu mer komplicerat, därför har stora delar av arbetsflödet automatiserats vilket gör det mer lätthanterligt. Detta inkluderar enkel aktivering och inaktivering av funktioner, filtreringsmöjligheter samt variabelbyte som påverkar funktioner snabbt och enkelt för exempelvis kamerarörelse och zoom-hastighet.

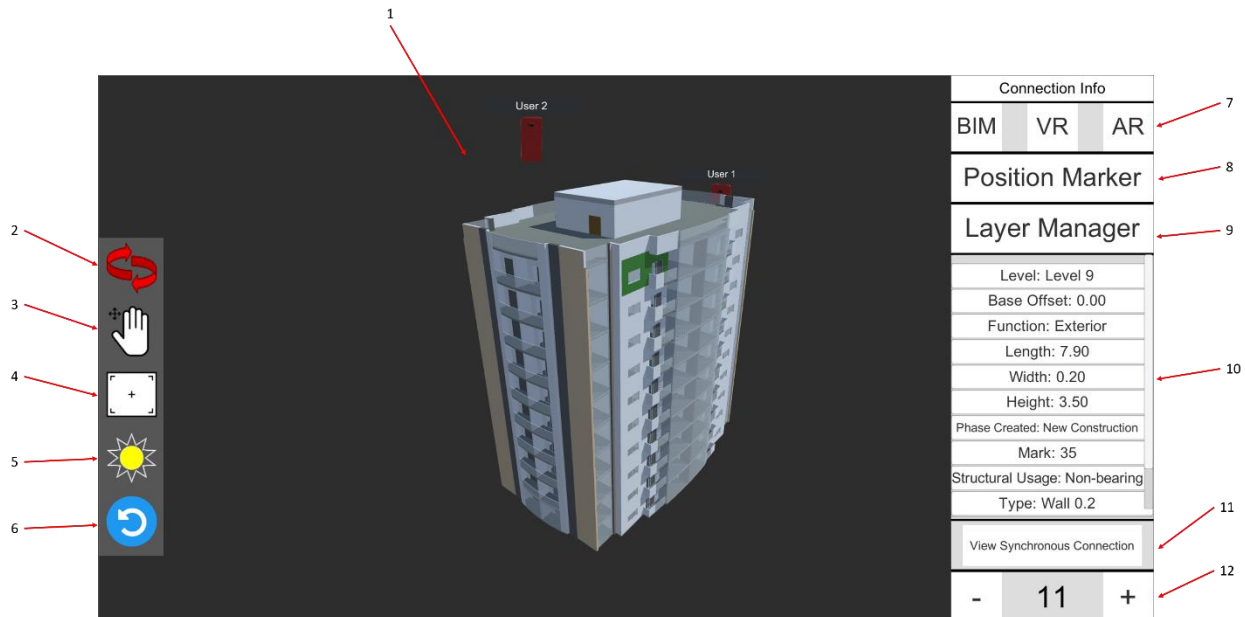
Utveckling demonstrator

Användargränssnitt

Användargränssnittet är uppbyggt av tre delar, modellvisare, kamerakontroller och Andra/BIM funktioner. Med detta i åtanke skissades förslagsritningar fram och slutligen valdes designen som ses i Figur 13.

Förklaring av Figur 13 numreringar:

- | | |
|------------------------------|---|
| 1. 3D-modell | 7. Bytt visualiseringsmetod |
| 2. Omloppera "Orbit" kameran | 8. Objektmarkör |
| 3. Pan-kameran | 9. Lagervisningsväxlare |
| 4. Ny mittpunkt | 10. BIM-visare |
| 5. Omloppera solen | 11. Nätverkssynkronisering |
| 6. Återställ kameran | 12. Våningsplanväljare + nuvarande våning |



Figur 13 Användargränssnittet och numrering av huvudfunktioner.

I användargränssnittet har deltagare 1 och 2 tyckt att den högra delen, från nummer 2 till och med 6 i Figur 13, är tydliga och att knappen hade god storlek. För den vänstra delen så tyckte deltagare 1 att den krävde större knappar med tydligare ikoner som beskriver dessa funktioner samt att ha en bakgrundsfärg för dem. Detta för att de är svåra att skåda vid användandet av förstärkt verklighet, utifrån dessa rekommendationer har användargränssnittet utvecklats vidare. Figur 13 illustrerar resultatet efter uppmaning från deltagarna.

Klickade objekt färgas temporärt grönt som indikation. Samma metod används för alla växlingsknappar men istället för grön färg används en röd färg, nummer 3 i Figur 13, röd färg indikerar att knappen/funktionen är aktiv.

I användningen av förstärkt verklighet sker en slags blindhet för användargränssnittet, där användarna missar många av tillgängliga knappar och 2d-bilder. Detta sker på grund av att fokuset är lagt på den fysiska världen som ses genom smarta mobiltelefonens kamera. (Mira, 2018).

För att minimera blindhet för användargränssnittet har knapparna grupperats i två paneler med tydliga gränser som representerats genom panelens bakgrundsfärg. För vidareutveckling av användargränssnittet behövs text och animationer "UI tutorials" som klargör funktionerna i förstärkt verklighet till nya användare. Det är också viktigt att möjliggöra gömning av knapparna för maximering av skärmutrymmet.

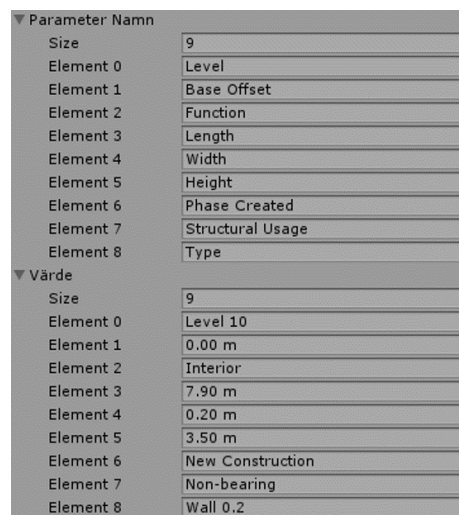
Integration av BIM till spelmotor

Skripten för integration av BIM till Unity är uppdelad till två skript. Första skriptet är för integration och inläsning av scheman från Revit till Unity, se Figur 14 och Figur 16. Den andra skriptet visar parametriska egenskaper genom att redovisa resultatet av första skriptet, se Figur 17 och Figur 18.

Första skriptet börjar med inläsning av exporterade scheman från Revit. Där varje inläst rad separeras och sedan adderas till list "radSepareradList". När additionen är klart, läses list "radSepareradList" kontinuerligt från det första till sista raden. All text i första raden som befinner sig mellan två citattecken delas upp, se Figur 15, där var och en av dem läggs i en egen rad i list "parameterNamn". För återstående rader i list "radSepareradList" sker en likadan uppdelningsmetod men läggs i en temporär list "värde", se Figur 14. För att koppla rätt objekt till rätt lista görs en uppsökning på element-ID i alla objektsnamn i Unity, detta sker genom användandet av första raden i "värde" list som innehåller ID-numret. Vid upptäckande av ett kompatibelt objekt kopieras list "parameterNamn" och list "värde" till själva objektet, se Figur 14. Detta möjliggör inläsningen av parametrarna för ett specifikt objekt via andra skriptet i senare skede.

Alla upptäckta objekt adderas till en global list "revitObjekt" som gör sökandet efter dessa objekt snabbare samt förenklar användningen av dem i andra skript/funktioner. För alla objekt skapas även meshcollider som möjliggör kollision och kollisiondetektering.

När skriptet har gått igenom alla rader kontrollerar den om ytterligare scheman existerar, om inte så tar den slut.



▼ Parameter Namn	
Size	9
Element 0	Level
Element 1	Base Offset
Element 2	Function
Element 3	Length
Element 4	Width
Element 5	Height
Element 6	Phase Created
Element 7	Structural Usage
Element 8	Type
▼ Värde	
Size	9
Element 0	Level 10
Element 1	0.00 m
Element 2	Interior
Element 3	7.90 m
Element 4	0.20 m
Element 5	3.50 m
Element 6	New Construction
Element 7	Non-bearing
Element 8	Wall 0.2

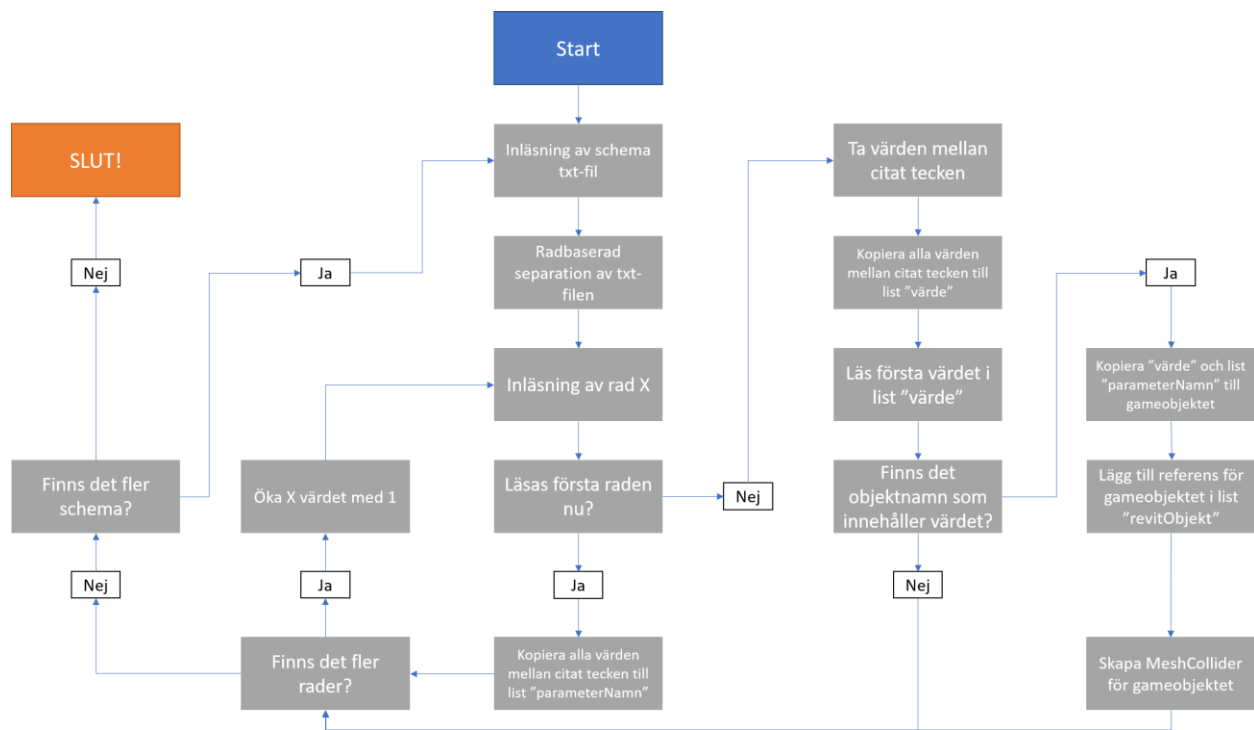
Figur 14 Visualisering av list "parameterNamn" och "värde" för ett objekt i Unity Editor.

```

"Element ID","Level","Base Offset","Function","Length","Width","Height","Phase Created","Structural Material","Structural Usage","Type","Area","Volume"
"303287","Level 0","0.00","Exterior","3.00","0.30","3.50","New Construction","Betong C90","Bearing","Col 0.3","11 m²","3.15 m³"
"303288","Level 0","0.00","Exterior","2.90","0.30","3.50","New Construction","Betong C90","Bearing","Col 0.3","11 m²","3.15 m³"
"303289","Level 0","0.00","Exterior","2.90","0.30","3.50","New Construction","Betong C90","Bearing","Col 0.3","10 m²","3.05 m³"
"303292","Level 0","0.00","Exterior","1.75","0.30","3.50","New Construction","Betong C90","Non-bearing","Col 0.3","6 m²","1.84 m³"
"303293","Level 0","0.00","Exterior","1.75","0.30","3.50","New Construction","Betong C90","Non-bearing","Col 0.3","6 m²","1.84 m³"
"303295","Level 0","0.00","Interior","1.25","0.40","3.50","New Construction","","Non-bearing","Wall 0.4","4 m²","1.75 m³"
"303296","Level 0","0.00","Exterior","7.90","0.20","3.50","New Construction","","Non-bearing","Wall 0.2","20 m²","3.99 m³"
"303297","Level 0","0.00","Interior","0.20","0.30","3.50","New Construction","","Non-bearing","Wall 0.3","1 m²","0.21 m³"
"303299","Level 0","0.00","Interior","3.03","0.10","3.50","New Construction","","Non-bearing","Wall 0.1","8 m²","0.85 m³"
"303300","Level 0","0.00","Interior","3.03","0.10","3.50","New Construction","","Non-bearing","Wall 0.1","9 m²","0.89 m³"
"303301","Level 0","0.00","Interior","2.35","0.10","3.50","New Construction","","Non-bearing","Wall 0.1","6 m²","0.59 m³"
"303302","Level 0","0.00","Exterior","3.25","0.20","3.50","New Construction","","Non-bearing","Wall 0.2","11 m²","2.10 m³"
"303303","Level 0","0.00","Exterior","7.00","0.20","3.50","New Construction","","Non-bearing","Wall 0.2","25 m²","4.97 m³"
"303304","Level 0","0.00","Exterior","0.75","0.20","3.50","New Construction","","Non-bearing","Wall 0.2","3 m²","0.53 m³"
"303305","Level 0","0.00","Exterior","6.35","0.20","3.50","New Construction","","Non-bearing","Wall 0.2","19 m²","3.73 m³"
"303306","Level 0","0.00","Interior","2.50","0.10","3.50","New Construction","","Non-bearing","Wall 0.1","6 m²","0.62 m³"
"303307","Level 0","0.00","Interior","0.48","0.30","3.50","New Construction","","Non-bearing","Wall 0.3","2 m²","0.50 m³"
"303308","Level 0","0.00","Interior","2.60","0.10","3.50","New Construction","","Non-bearing","Wall 0.1","9 m²","0.82 m³"
"303309","Level 0","0.00","Interior","2.85","0.10","3.50","New Construction","","Non-bearing","Wall 0.1","10 m²","1.00 m³"
"303310","Level 0","0.00","Interior","1.25","0.10","3.50","New Construction","","Non-bearing","Wall 0.1","2 m²","0.22 m³"
"303311","Level 0","0.00","Interior","1.25","0.10","3.50","New Construction","","Non-bearing","Wall 0.1","4 m²","0.39 m³"
"303312","Level 0","0.00","Exterior","0.30","0.20","3.50","New Construction","","Non-bearing","Wall 0.2","1 m²","0.21 m³"
"303313","Level 0","0.00","Interior","1.35","0.10","3.50","New Construction","","Non-bearing","Wall 0.1","4 m²","0.44 m³"
"303331","Level 0","0.00","Interior","7.80","0.10","3.50","New Construction","","Non-bearing","Wall 0.1","23 m²","2.27 m³"
"303332","Level 0","0.00","Interior","2.45","0.10","3.50","New Construction","","Non-bearing","Wall 0.1","6 m²","0.59 m³"
"303333","Level 0","0.00","Interior","1.30","0.10","3.50","New Construction","","Non-bearing","Wall 0.1","1 m²","0.12 m³"
"303334","Level 0","0.00","Interior","3.95","0.10","3.50","New Construction","","Non-bearing","Wall 0.1","13 m²","1.33 m³"
"303335","Level 0","0.00","Interior","0.35","0.10","3.50","New Construction","","Non-bearing","Wall 0.1","1 m²","0.12 m³"
"303336","Level 0","0.00","Interior","0.60","0.10","3.50","New Construction","","Non-bearing","Wall 0.1","2 m²","0.18 m³"
"303346","Level 0","0.00","Interior","1.55","0.30","3.50","New Construction","","Non-bearing","Wall 0.3","5 m²","1.44 m³"
"303377","Level 0","0.00","Exterior","0.75","0.20","3.50","New Construction","","Non-bearing","Wall 0.2","3 m²","0.53 m³"
"303378","Level 0","0.00","Exterior","1.45","0.20","3.50","New Construction","","Non-bearing","Wall 0.2","5 m²","1.02 m³"
"303379","Level 0","0.00","Exterior","0.75","0.20","3.50","New Construction","","Non-bearing","Wall 0.2","2 m²","0.39 m³"
"303391","Level 0","0.00","Interior","0.24","0.10","3.50","New Construction","","Non-bearing","Wall 0.1","1 m²","0.10 m³"

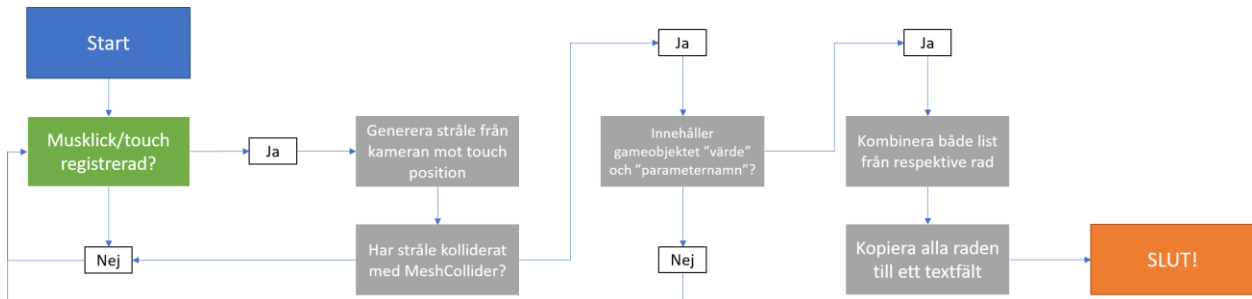
```

Figur 15 Exporterat schema från Revit i txt-format

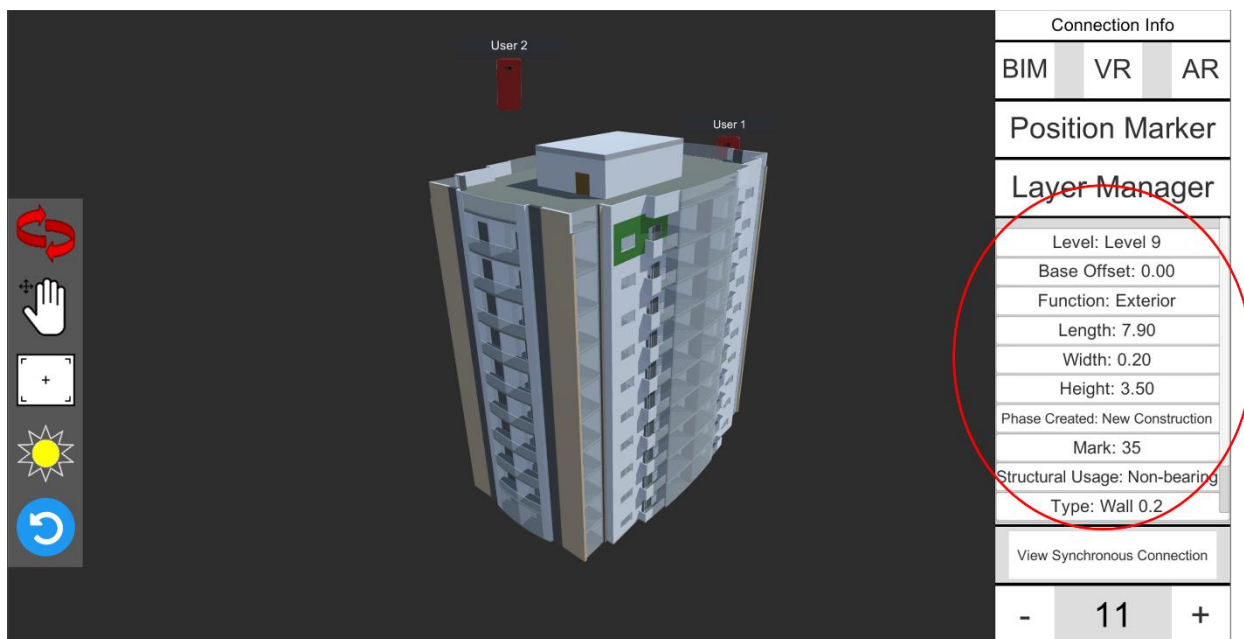


Figur 16 Flödesschema för integration av BIM i spelmotorn Unity.

Andra skriptet väntar konstant på klick från användaren, antingen i form av musklick eller touch. När ett klick registreras genereras en stråle som skickas från riktningen av kameran mot punkten i skärmen som klickades. Strålen har anvisningar om att referera till det första objekt, med en MeshCollider, som den kolliderar med. Vid kollision kontrolleras om objektet som har kolliderats med innehåller list "parameterNamn" och "värde". Om dessa list upptäcks så kombineras värdena från respektive rad och visas sedan för användaren i ett rullbart textfält, se Figur 18.



Figur 17 Flödesschema för visualisering av BIM. Grön representerar funktion som körs i varje frame.



Figur 18 Textfältet visar data från schema för klickat objekt (grönt).

Första integration av BIM i ett projekt till spelmotorn Unity tar cirka 10 minuter, vilket ses i Tabell 1. För modifieringar i senare skede behövs oftast inga nya scheman då dessa automatiskt uppdaterar sig. Omlänkning av Revit-fil till 3ds Max krävs heller inte, en manuell omladdning krävs dock. Det krävs heller ingen ny exportering från 3ds Max till spelmotorn Unity då Unity:s automatiska konverteringsverktyg upptäcker att dessa ändringar har utförts och därefter konverterar den nya filen och uppdaterar 3D-modellen. Detta möjliggör en tidsminskning för integrationen av BIM till cirka 3 minuter, samtidigt som det säkerställs att samtliga funktioner fungerar inklusive förstärkt verklighet, nätverkssynkronisering och objektmarkör.

Tabell 1 Krävd tid för att utföra alla processer för BIM integration. * tyder på att processen behöver utföras endast en gång.

Process	Krävd tid
Skapa Element ID	15 sekunder
Skapa scheman	4 minuter*
Länk av Revit till 3ds Max samt	2 minuter*
Materialkonvertering	15 sekunder
Export av 3ds Max till Unity	1 minut*
Export av schema till Unity	30 sekunder
Export/Bygga från Unity	2 minuter
Totalt	≈ 10 minuter

Integrationsmetoden möjliggör visualisering av BIM i realtid med hög grad automation då alla nödvändiga data kommer från den befintligt gjorda BIM. Detta resulterar i att processen är värd att användas även i tidiga skeden av projekteringsprocessen. Användningen av schema för dataextraktion ger användaren flexibilitet, där användaren kan inkludera samt exkludera parametrar från redan befintlig lista för att anpassa det till projektens behov. Användaren kan också skapa egna parametrar vid behov. Misslyckande i att visualisera BIM-data i realtid kan minska förmånen som realtidsvisualisering annars ger vid användandet (Roupé, Johansson, & Bosch-Sijtsema, 2015).

Ett utvecklingsförslag från examinatorn är att användaren ska kunna välja mellan fördefinierade data filter som är anpassade för olika aktörer. Till exempel för beställaren kan uppvisad information om konstruktion gömmas.

Våningsplanväljare och lagervisningsväxlare

För att BIM-visualisering ska kunna fungera så behövs det verktyg som assisterar användaren till att hitta/se det dem vill samt exkludera det dem inte vill se. För detta används två funktioner som är integrerade från BIM, nämligen våningsplanväljare och lagervisningsväxlare. Våningsplanväljaren gör det möjligt att gömma våningar som man inte är intresserad av med en knapp vilket ger användaren ett enkelt men kraftfullt filtreringsverktyg, se Figur 23. Detta möjliggör studerande av den våning man är intresserad av samtidigt som den förstärker BIM-visualiseringsmetoden som är baserad på objektklickning för informationsextraktion.

Lagervisningsväxlare är ett mer avancerat filtreringsverktyg, där användaren kan vara mer specifikt på sina val, se Figur 21. Detta ger användaren högre tillgänglighet till detaljer, så som balkar, kolumner med mera. Detta filtreringsverktyg förenklar även användandet av förstärkt verklighet vid studerande av ett objekt i större/verklig skala.

Dessa funktioner är beroende av tre olika skript. Första skriptet, se Figur 19, skapar alla nödvändiga lager. Med lager menas knappen som innehåller en liten solikon samt namnet på parametern eller typen av objekt, se Figur 21. Det är i första skriptet som all nödvändig beräkning sker och alla list:s och lager skapas. Andra skriptet nyttjar sig av skapade list:s i första skriptet för växling av synligheten, se Figur 20. Tredje skriptet, se Figur 22, nyttjar sig också av list skapade i första skriptet, men istället används dem för växling mellan olika våningsplan, se Figur 23.

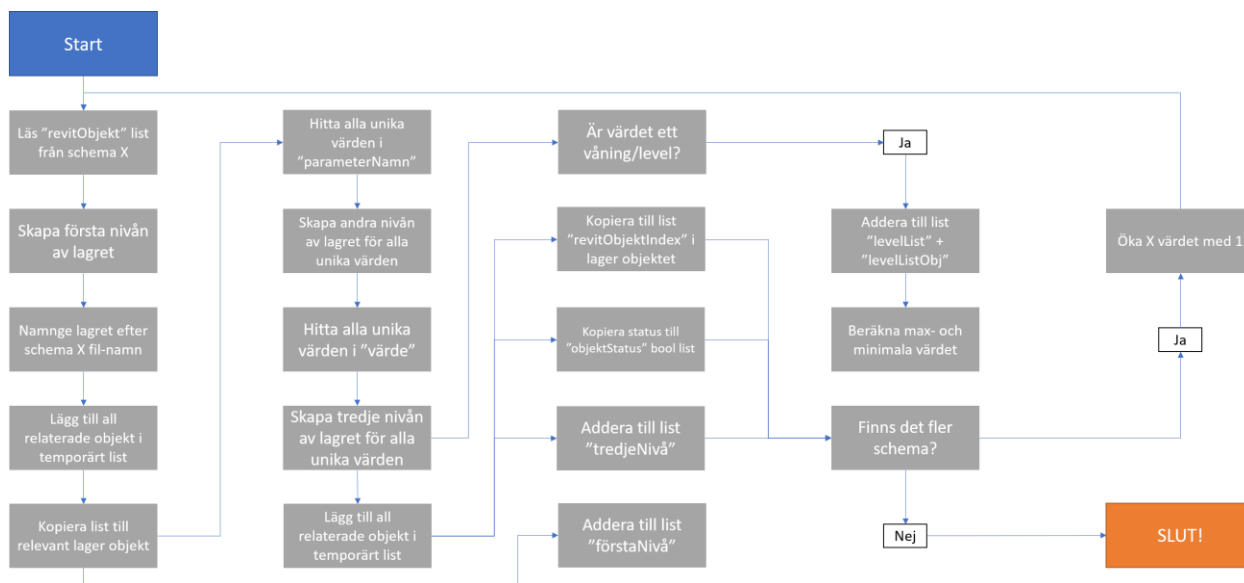
Första skriptet inhämtar data från alla list "revitObject" som skapades via första skriptet i föregående rubrik. Vid upptäckande av skripten skapas första lagret i första nivån som namnges efter namnet på schema-filen från Revit, det är text som visas till användaren, i detta fall är det "Wall" som ses i Figur 21. Referens till lagret och dess status som alltid börjar med "true" läggs till i list "förstaNivå" och "objektStatus", dessa möjliggör nätverkssynkning i senare skeden. Objekten som påverkas av första lagret söks upp i "revitObjekt", där dess indexvärde, radnummer, adderas till temporära listan "revitObjektIndex" som vidare överförs till själva lagret, det vill säga till andra skriptet.

För skapandet av första lagret i andra nivån sker sökning efter alla unika värden i list "parameterNamn" från schemaskripten. För varje upptäckt unikt värde skapas ett nytt lager som placeras i andra nivån, där lagret namnges efter texten på det unika värdet, exempelvis "Level", "Function". Mer än detta görs inte i andra nivån och istället börjar skapandet av tredje nivån.

För skapandet av tredje nivån en sökning sker på alla unika värden för varje parameter i list "värde". Varje upptäckt unikt värde leder till skapandet av ett nytt lager som placeras i tredje nivån och namnges efter texten på det unika värdet, exempelvis "Interior" eller "Exterior". För varje skapat lager i tredje nivån läggs en referens till list "tredjeNivå" och dess status till "objektStatus". Objekten som påverkas av varje unikt lager i tredje nivån söks upp i "revitObjekt", där dess indexvärde, radnummer, adderas till den temporära listan "revitObjektIndex" som vidare överförs till lagren. Detta kan efterliknas till tömning av en lådas innehåll till en annan låda, där målet är omsortering och organisation. Om sökta parametern tillhör gruppen för våning

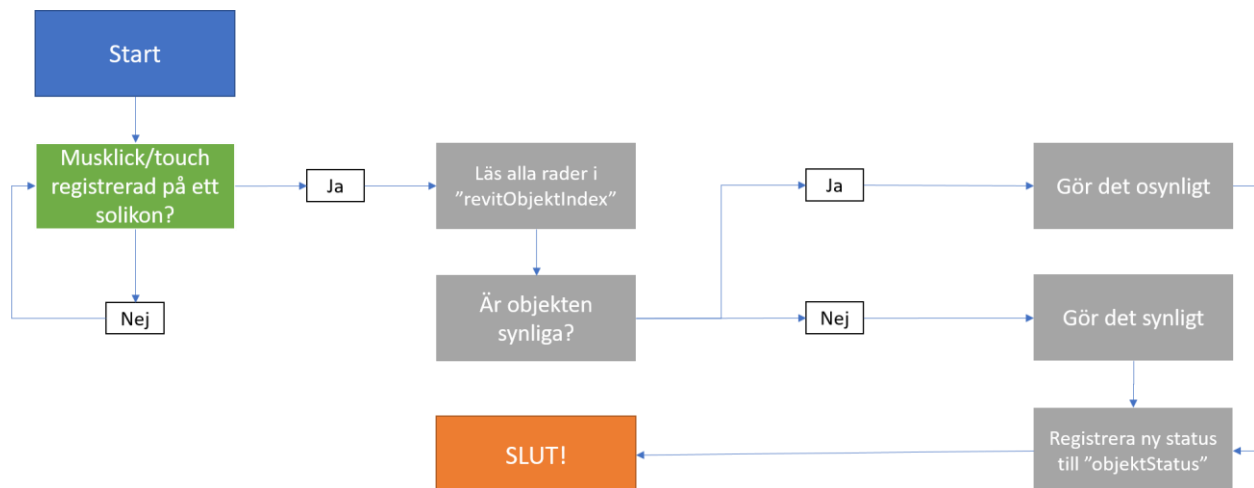
”Level” läggs även referens till objektet i list ”levelListObj” och dess våningsplanssiffra till ”levelList”. Maximala samt minimala värdet på listan ”levelList” kontrolleras, detta görs för att kunna få värdet på högsta samt lägsta våning som byggnaden kan ha. Skriptet kontrollerar nu om ytterligare schemaskript existerar, om inte så tar den slut.

Våningsplanväljare samt lagervisningsväxlare kombineras i första skriptet för minimering av risken för funktionskollision, där både fokuserar på växlande av synlighet för ett eller flera objekt. Detta medför krav på att dessa funktioner måste kunna kommunicera med varandra.

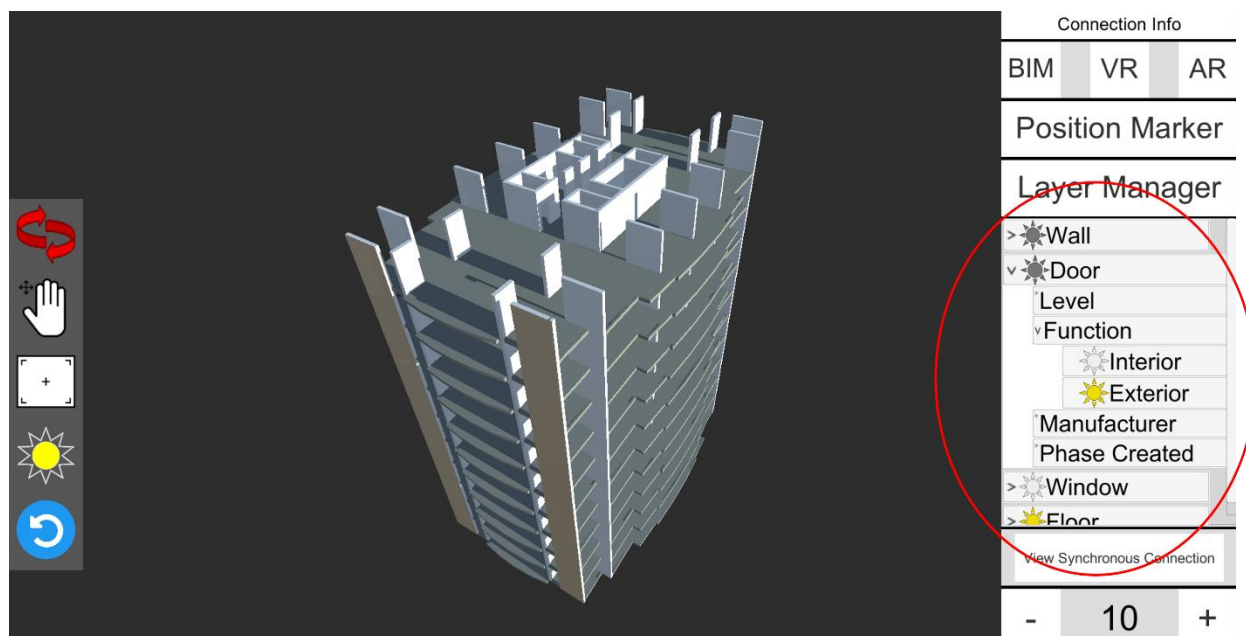


Figur 19 Flödesschema för lagervisningsväxlare.

Andra skriptet börjar med att den konstant väntar på klick från användaren på en av ”solikonerna” i lagren, se Figur 21, antingen i form av musklick eller touch. Klickandet resulterar i inläsning av list ”revitObjektIndex” och ”objektStatus” som finns i solikonens lager. Indexvärden i list ”revitObjektIndex” används för upphittande av alla relaterade objektens status i list ”objektStatus” och referens från list ”revitObjekt”. Beroende på statusen växlar objektet antingen till synligt eller osynligt, därefter registreras ändringen som har utförts på synligheten till list ”objektStatus”.



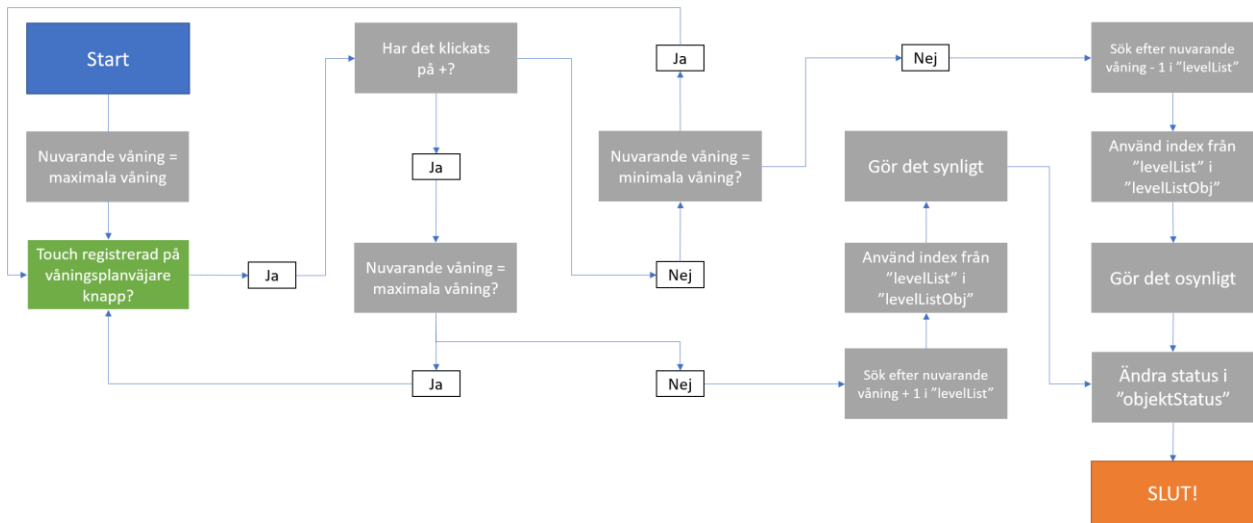
Figur 20 Flödesschema för lagerknapp som växlar synlighet. Grön representerar funktion som körs i varje frame.



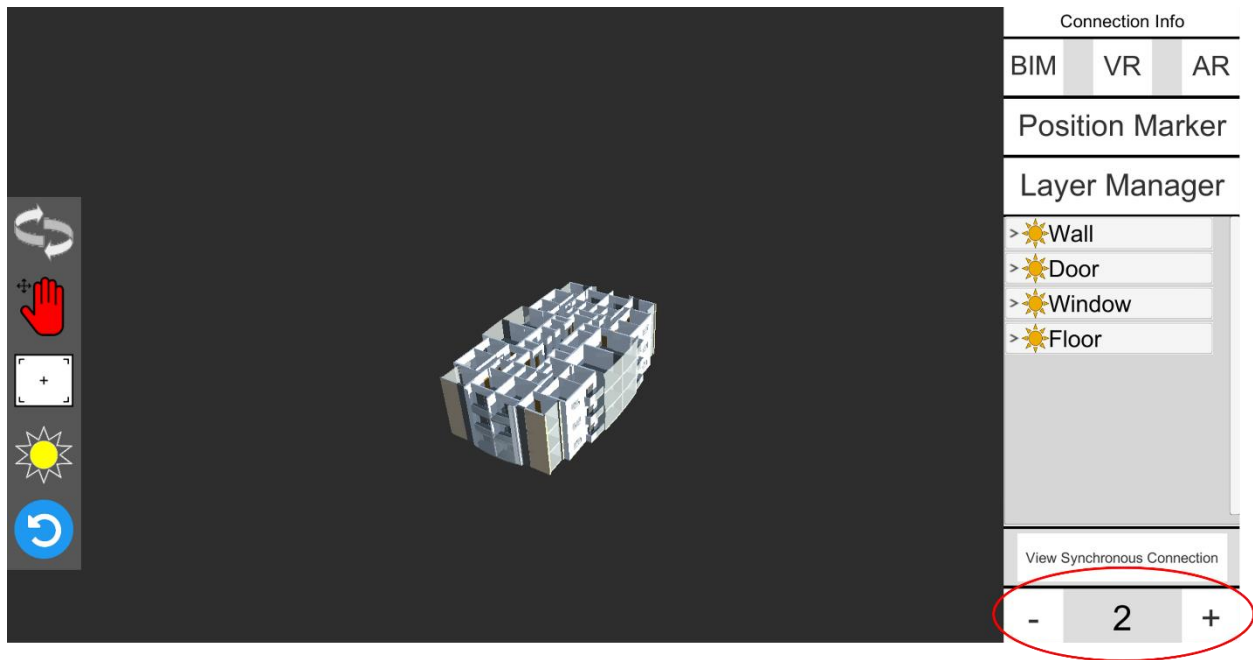
Figur 21 Lagervisningslistan, gul är synligt, transparent icke synligt, grå är delvis synligt

Tredje skriptet använder sig av beräknade värden på högsta våning för att anta värdet på nuvarande våning, därefter väntar skriptet på klick på ena av knapparna - eller +, se Figur 23. Vid klick på knappen + sker en jämförelse mellan värden för högsta våning med nuvarande våning, om dem är lika, det vill säga att ingen mer våning finns så gör skriptet ingenting. När istället nuvarande våningsvärde är lägre än högsta våningsvärde utförs en sökning av värdet på nuvarande våning plus ett i list "levelList" för upphittandet av indexen till relaterade objekt. Indexen används i list "levelListObj" för att hitta själva objekten som synliggörs. Ändringen på synligheten meddelas till första skriptets list "objektStatus".

Vid klick på knappen -, se Figur 23, sker en jämförelse mellan värden för lägsta våning med nuvarande våning, om dem är lika så gör skriptet ingenting. När istället nuvarande våningsvärde är högre än lägsta våningsvärde utförs en sökning av värdet på nuvarande våning minus ett i list "levelList" för upphittandet av indexen till relaterade objekt. Indexen används i list "levelListObj" för att hitta själva objekten som osynliggörs. Ändringen på synligheten meddelas till första skriptets list "objektStatus".



Figur 22 Flödesschema för våningsväljare. Grön representerar funktion som körs i varje frame.



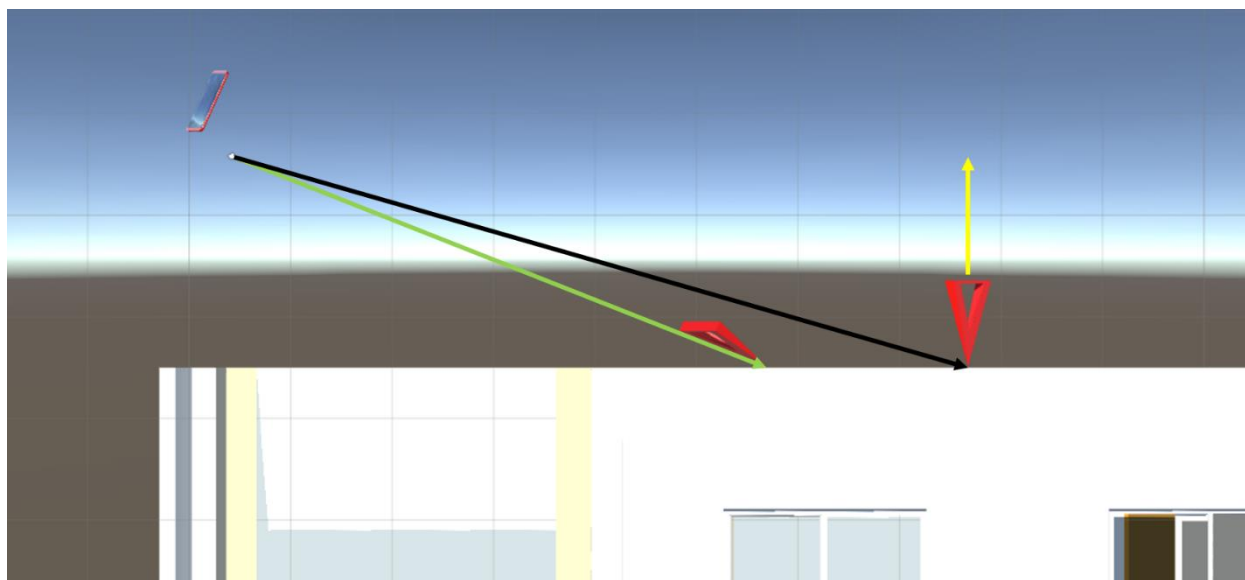
Figur 23 Våningsplanväljare används för att kunna se andra våningsplanen.

Funktionen våningsplanväljare är i tidiga skeden, där utvecklingsmålet är att möjliggöra visualisering av planlösningar med hjälp av samma BIM-visualiseringsmetod som beskrivits. Planlösningen ska även kunna visa alla nödvändiga mått.

Lagervisningsväxlare kan utvecklas vidare genom att skapa funktioner som möjliggör isolering/gömning av individuella objekt. Examinatorns utvecklingsförslag på fördefinierade filter som är anpassade för olika aktörer kan även fungera väl med lagervisningsväxlare, där skapade lagren varierar beroende på användaren.

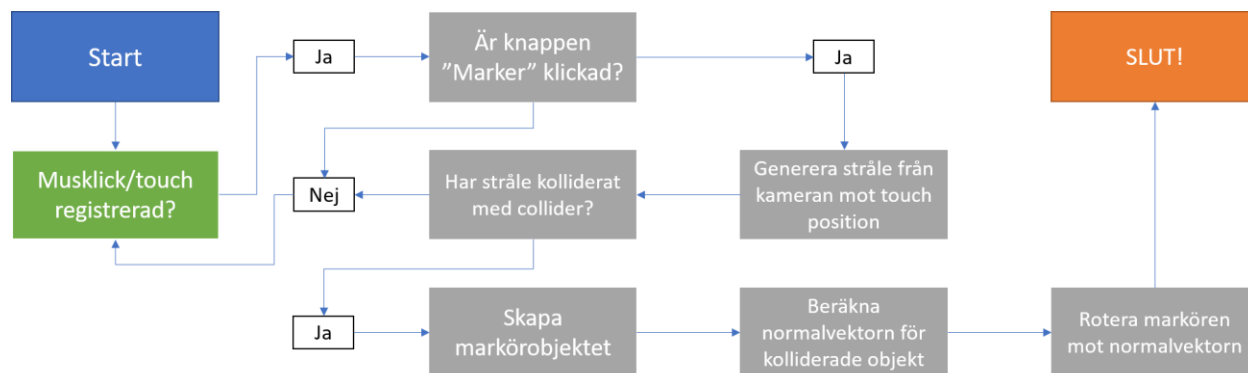
Objektmarkör

Objektmarkör-funktionen gör det möjligt att placera ut markörer genom att klicka på objekt. Markören hamnar på den punkt som användaren klickar på. Målet med denna funktion är att kunna simplificera markeringen i förstärkt verklighet men också i BIM-läge vilket kan leda till minskad förvirring och färre fel i kommunikation. Två alternativ för markörplacering samt rotation har testats, i slutändan har det andra alternativet använts då det bedömdes vara tydligare. Första alternativet gick ut på att rotera markören baserat på kamerapositionen vid klick, Figur 24 alternativ 1, detta alternativ är enklare att skriva skript för samtidigt som det ger all kontroll över markörrotation till användaren. Då en markör på exakt samma position/punkt kan leda till skapandet av två olika markörer som har olika riktningar. Det andra alternativet gick ut på att beräkna normalvektorn för det kolliderade objektet, för att bestämma dess ytas orientering, Figur 24 alternativ 2. Markörrotation kontrolleras endast av objektets geometri i punkten som användaren har klickat på vilket leder till mer tydliga och arbets säkra markörer.



Figur 24 Markören till vänster representerar alternativ 1, till höger är det alternativ 2.

Skriptet börjar med att den konstant väntar på klick från användaren. Vid registrering av klick kontrolleras om knappen "Objektmarkör", nummer 8 i Figur 13 är klickat. Om ja, så genereras en stråle som skickas från kamerariktningen mot punkten i skärmen som klickades. Vid kollision mellan strålen och en MeshCollider beräknas normalvektorn för det kolliderade objektets yta. I den kolliderade punkten skapas markörojektet som sedan roteras mot riktningen av beräknade normalvektorn.



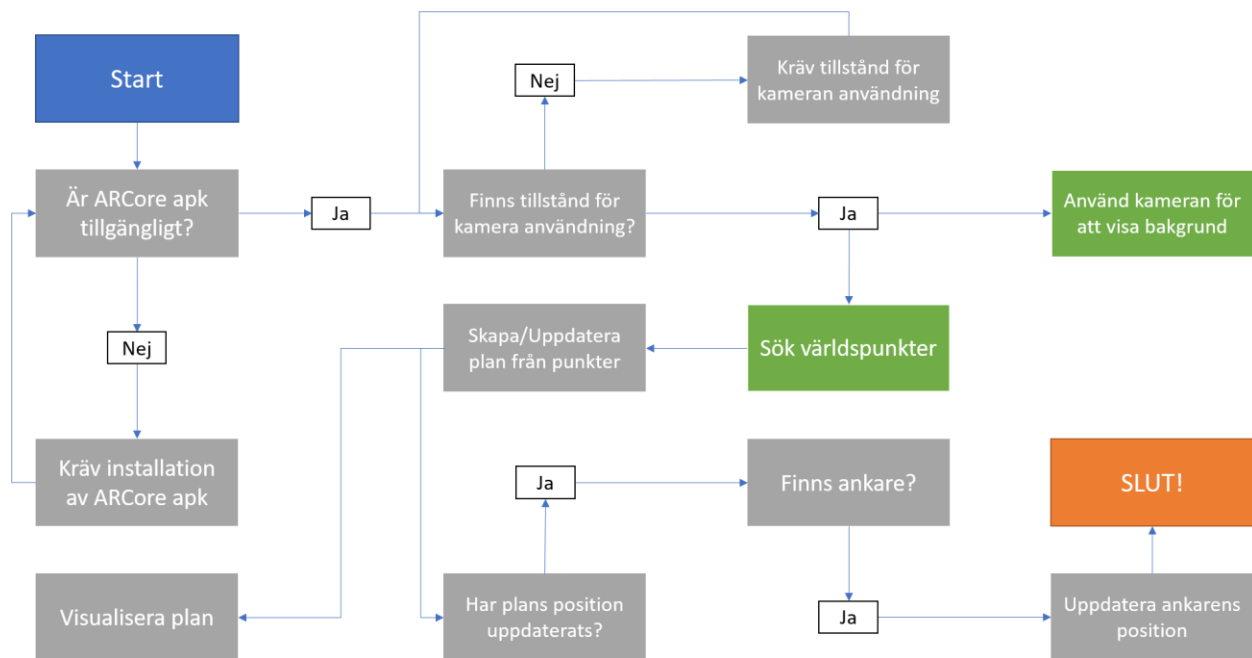
Figur 25 Flödesschema för objektmarkör placering. Grön representerar funktion som körs i varje frame.

Deltagare 3 tycker att markören kan förbättras genom möjlighet till skissning på skärmdump och kommentering som sparas och tillgängliggörs för andra aktörer att se vid klickande på själva markörojektet. Markören bör också ha unik färg för varje användare och/eller prioritet. Deltagare 3 sa:

"Jag tycker att markören bör kunna inkludera kommentarer och skärmbilder och helst möjligheten till olika markörfärger."

Förstärkt verklighet

Implementering av förstärkt verklighet sker genom användning av flera skript som är tillgängliga för användning i ARCore SDK. Ändringar har utförts på flera skript för att anpassas till projektets behov. Skripten börjar med letande efter ARCore APK, det vill säga ARCore mjukvaran i Android:s smarta mobiltelefoner, och kräver installation om ingen upptäcks. ARCore kontrollerar också om tillstånd för kameraanvändning finns och ifall det saknas frågas användaren om tillstånd. Skripten tar bilddata från kameran och överför det till bakgrundsbild via omvandling av kameradata till material i Unity. Därefter söker skripten efter världspunkter, detta sker i en konstant takt som leder till att ARCore får en ökad förståelse över den fysiska världen allt eftersom den körs. Insamlade data möjliggör identifiering av horisontala plan som visualiseras till användaren. Då världspunkter söks konstant och dess position uppdateras så sker även uppdatering konstant på de upptäckta horisontala planens positioner. Vid existens av ankare, uppdateras dess position i samband med det upptäckta planet som den är kopplat till.



Figur 26 Flödesschema för ARCore. Grön representerar funktion som körs i varje frame.

Kamerastyrning

Kamerastyrning i BIM läge

Vid skapandet av navigationsverktyget har två olika metoder testats. Första testen använde sig av en flygande kamera "Flying camera, free roaming camera" som oftast är en naturlig kamerastyrningsmetod för en 3D-modellbaserad visualiseringsmjukvara för en personlig dator, dock används den sällan för smarta mobiltelefoner. I detta fall har prövningen visat att det var en frustrerande kamerastyrningsmetod när det kombinerades med BIM-visualiseringsverktygen. Prövningen samt navigationssvårigheten i smart mobiltelefon har lett till sökandet efter en annan navigationsmetod. För att bestämma navigationsmetoden har inspiration sökts från 3D-visualiserings appar för Android:s smarta mobiltelefoner. Utifrån prövningarna valdes navigationsmetoden som orienterar sig i en sfärisk bana "omlopps bana" runt ett objekt, nämligen "orbit kamera". För att möjliggöra en navigation som är oberoende av byggnaden har ett objekt skapats i Unity "centerPoint".

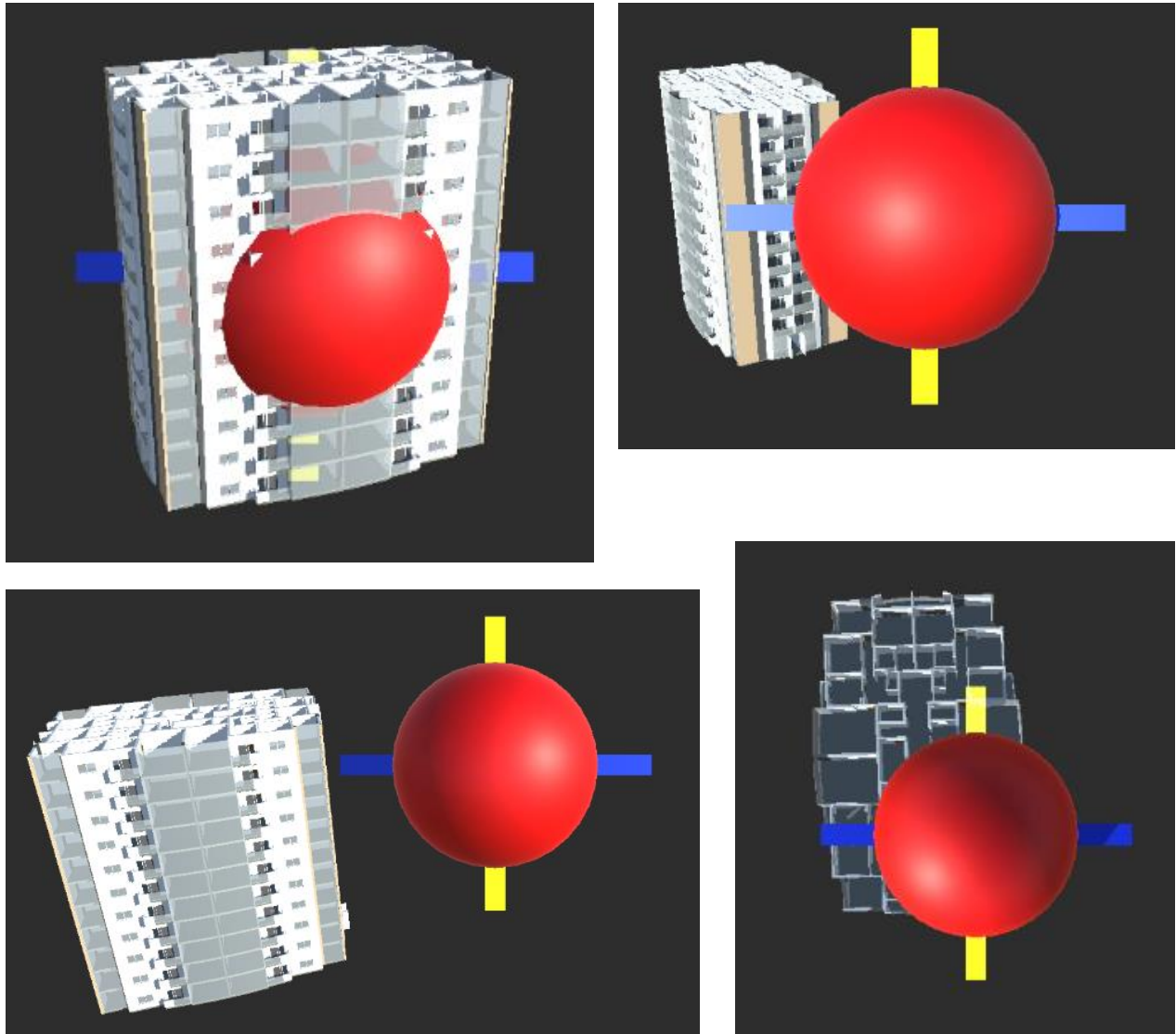
Kamerastyrningsskriptet börjar med beräkning av mittpunkten för byggnadsmodellen. För att utföra detta beräknas mittpunkten för alla individuella byggnadselement i form av vektorer som adderas ihop till en vektor. Summan av alla vektorer divideras med totala antalet byggnadselement som ger oss mittpunkten i vektorform. I den beräknade position skapas objektet "centerPoint", den övre, vänstra bilden i Figur 27 är startpunkten. Vektorn för mittpunkten sparas för användning när återställning till startposition via knappen "återställVy" används, knapp nummer 6 i Figur 13.

Kameran ser till att hålla ett oföränderligt avstånd "kameraDistans" samt riktning relativt till "centerPoint". När en touch registreras kontrollerar skriptet typen av touch som har utförts, nämligen pinch, pan eller klick. Vid pinch beräknar skriptet avståndet på pinch, det vill säga hur mycket avståndet ändrats och åt vilken riktning mellan både fingrarna. Värdet som fås används i en formel som inkluderar "kameraDistans" samt en hastighetskonstant för beräkning av nya värdet för "kameraDistans". Ändringen av "kameraDistans" flyttar kameran som leder till möjligheten för både in- och utzoomning.

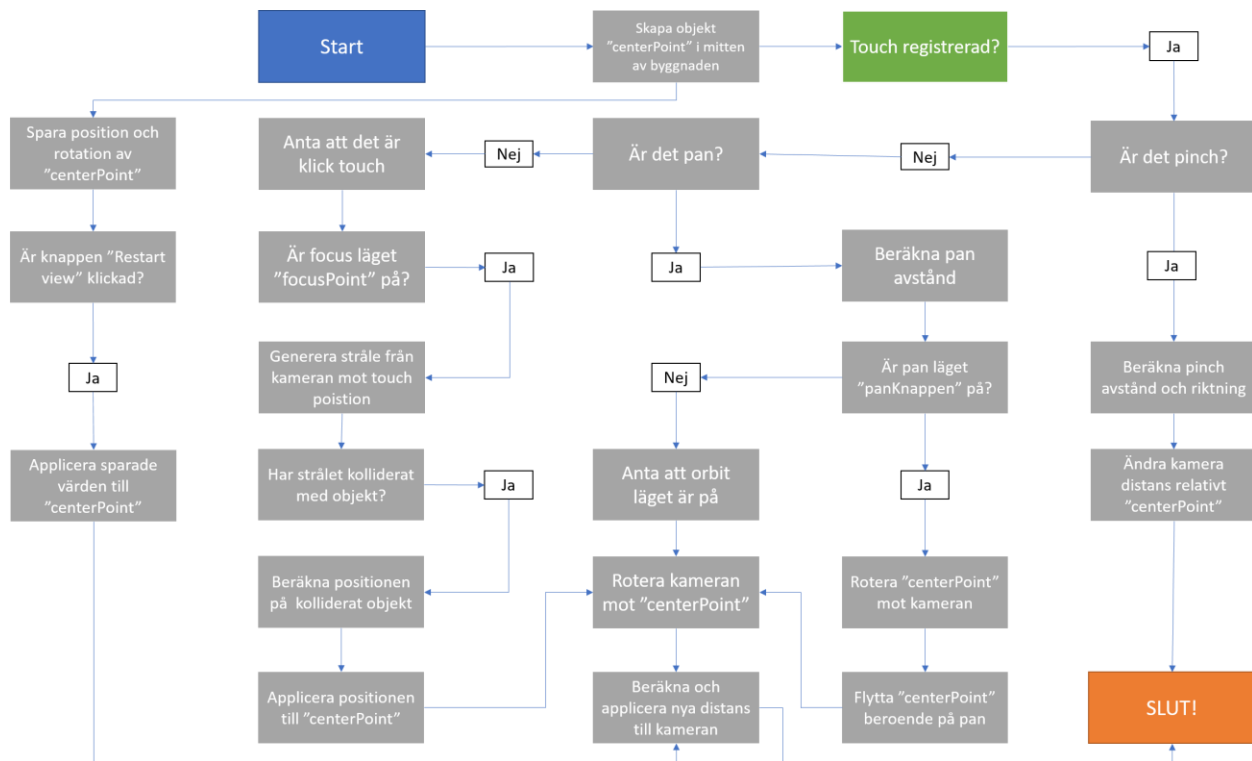
Vid pan beräknas avståndet från början till slutet av touch. Därefter kontrolleras om det är knappen "panLäget" eller "orbitLäget" som har aktiverats, nummer 3 respektive 2 i Figur 13. I "panLäget" roteras "centerPoint" mot kameran för att därefter flyttas omkring sina lokala axlar, som representeras i Figur 27 av dem blåa och gula stavarna. Detta leder till att pan åt höger riktning förflyttar objektet "centerPoint" åt sin högra sida vilket motsvarar vänstra sidan relativt till kameran. Utan rotation av "centerPoint" så tappas detta förhållande som resulterar i att pan-riktningen inte ändras beroende på kamerariktningen.

I "orbitLäget" förflyttas kameran mot riktningen av pan samtidigt som den roteras mot "centerPoint". Därefter räknas nya kamerapositionen för att säkerställa att "kameraDistans" inte ändras.

Om registrerade touchen inte är pinch eller pan görs antaganden att det är ett klick. För detta kontrolleras om knappen "Ny mittpunkt" är aktiv, nummer 4 i Figur 13. Vid ett aktivt "fokus" genereras en stråle som skickas från kamerariktningen som går mot punkten i skärmen som klickades. Vid kollision med en collider inhämtas kollisionens position som appliceras på objektet "centerPoint". Kameran roteras mot nya positionen på "centerPoint" och en ny position räknas för kameran som säkerställer att "kameraDistans" inte ändras. Detta möjliggör selektion av nya mittpunktpositioner.



Figur 27 Representation av "centerPoint" och dess lokala axel. Den övre, vänstra bilden är startpunkten.



Figur 28 Flödesschema för kamerastyrning i BIM-läge, pinch för zoom, pan för förflyttning och klick för position specificering

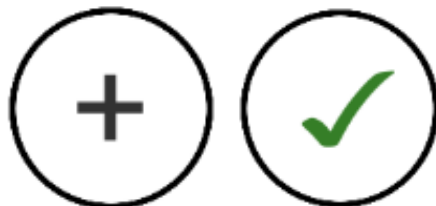
Kamerastyrningen har valts kontrolleras endast med knappar i tidiga skeden för att simplificera användningen samt utvecklingen. Detta är en utvecklingsmöjlighet då möjligheten till både knappar och gester kan förenkla samt snabba upp användningen för en bredare publik. Särskiljning kan ske mellan kort och lång klickning, där en lång klick till exempel kan selektera en ny mittpunktposition "centerPoint". Särskiljning kan också ske mellan pinch och två finger pan för kamerapanorering istället för in- och utzoomning.

Kamerastyrning i förstärkt verklighet

I denna del skapas funktionen för placering av ankare/objektet i förstärkt verklighet. Funktionen inspireras av skripten för "HelloAR" som är tillgängligt i ARCore SDK. "HelloAR" placerar ett nytt objekt, som i detta fall är en 3D-modell av Android-loggan, vid varje klickning som sker på ett detekterat plan. (Google ARCore, 2018).

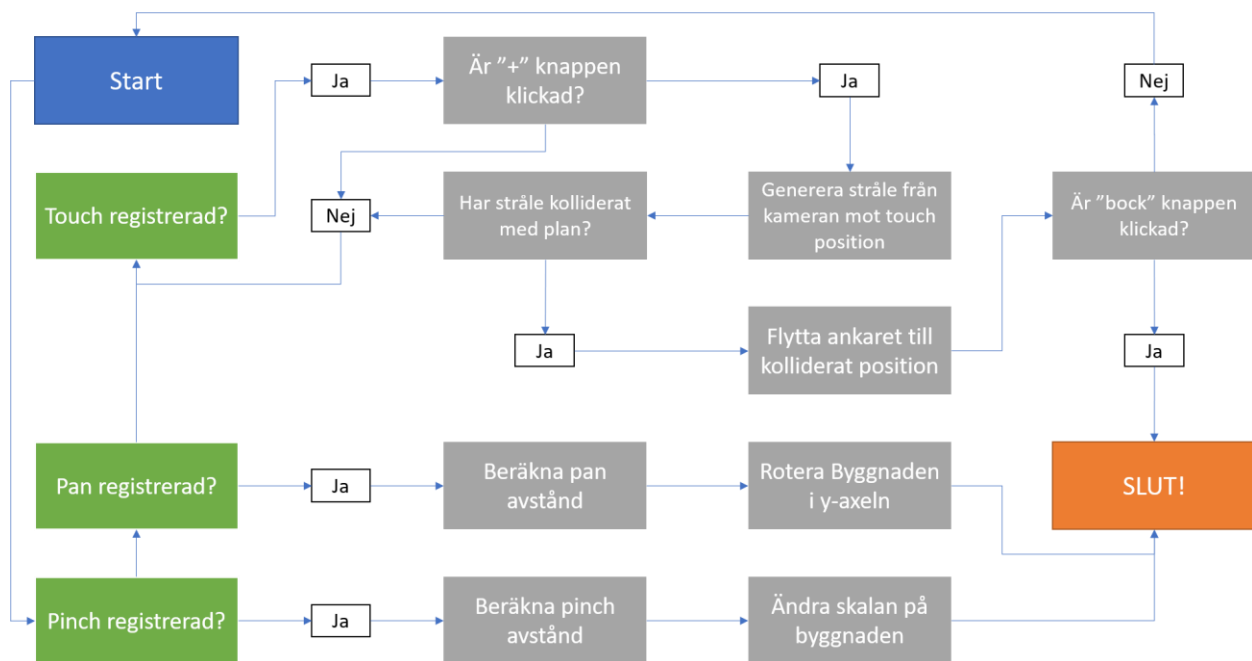
Först testades med att begränsa antalet skapade objekt "byggnadsmodell" till endast ett objekt, detta resulterar i att endast första klicket placerar byggnadsmodellen i det detekterade plan som klickningen har utförts på. Anledningen till denna begränsning är att BIM-visualiseringen använder sig av klick på byggnaden vilket resulterar i funktionell kollision. Återställningsknappen, nummer 6 i Figur 13, används för omförflyttning med samma metod. Första metoden kunde användas utan funktionella problem, dock var det inte tillräckligt smidigt då testpersonerna använde sig av återställningsknappen i början för att få en önskad vy.

För att placera byggnadsmodellen i andra testet har en + knapp, Figur 29, lagts till som blir tillgängligt när förstärkt verklighets-läget används. Efter klickning på + knappen förflyttar sig byggnaden konstant oberoende om det en klick eller pan som utförs på detekterade planen. Detta är möjligt tills användaren väljer att klicka på bockikonen, se Figur 29, som visas istället för +. Återställningsknappen används för omförflyttning med samma metod tills bockikonen klickas igen.



Figur 29 plus samt bockikon som används för positionering samt klart i förstärkt verklighet.

Vid touch och pan granskar skriptet, Figur 30, om + knappen har klickats på så genereras en stråle som skickas från kamerariktningen mot punkten i skärmen som klickas. Strålen kolliderar endast med detekterade plan och vid kollision förflyttas ankaret till kollisionspunkten. Skriptet utför detta tills bockknappen är klickad. Då ändras funktionaliteten för touch och pan samtidigt som skriptet kontrollerar touch-typen. För pinch beräknas avståndsändringen samt dess riktning mellan både fingrarna. Värdet adderas till byggnadsmodellens skala. Metoden är relativt enkel att utföra, dock har den begränsningar i form av fysisk integration då Unity inte skalar ner gravitations- och kollisionskraften. Enligt Alamparambil & Mowrer (2017) kan problemet lösas genom att istället genomföra ändringar på virtuella kamerans och detekterade planets skala och dess avstånd. Detta har visat sig vara svårt att implementera i detta projekt då problem med icke konsistenta plan samt prestandaproblem har uppstått, dock innebär detta bara att mer tid krävs för denna funktion. Vid pan beräknas avståndet mellan första och sista touchen. Värdet som fås adderas för att rotera byggnadsmodellen runt y-axeln, som är riktad uppåt.



Figur 30 Flödesschema för kamerastyrning i förstärkt verklighet, pinch för zoom, pan för rotation och touch för objektplacering.

Diskussioner har utförts på förstärkt verklighets manövrering och kontroll. Deltagare 1 tycker att förstärkt verklighet ger en naturligare navigationsmetod för smarta mobiltelefoner i jämförelse med traditionella 3D-mjukvaror. Där istället för att använda sig av gester och knappar så flyttar man sig fysiskt för att bytta vinkeln som man tittar på och detta möjliggör högre noggrannhet vid navigation. Dock att behöva konstant hålla smarta mobiltelefonen i handen i en onaturlig position är problematiskt, särskild för co-located samt långa möten. Första deltagaren sa:

"Förstärkt verklighet gör det enklare att följa och titta på detaljen som man är intresserad av. I ett distribuerat möte kan jag se att den ersätter en 3D utskrift av en modell, men i ett co-located möte så kan den begränsa kommunikation."

Deltagare 3 tycker också att användningen av förstärkt verklighet i längden blir jobbigt. Det finns också bekymmer över batteritid när förstärkt verklighet använder sig av kameran. Deltagare 3 tycker att han har fått en bättre uppfattning av byggnadsvolymen i förstärkt verklighet, detta har inte första och andra deltagarna, som är erfarna med att läsa planlösningar och 3D-modeller, märkt. Tredje deltagaren sa:

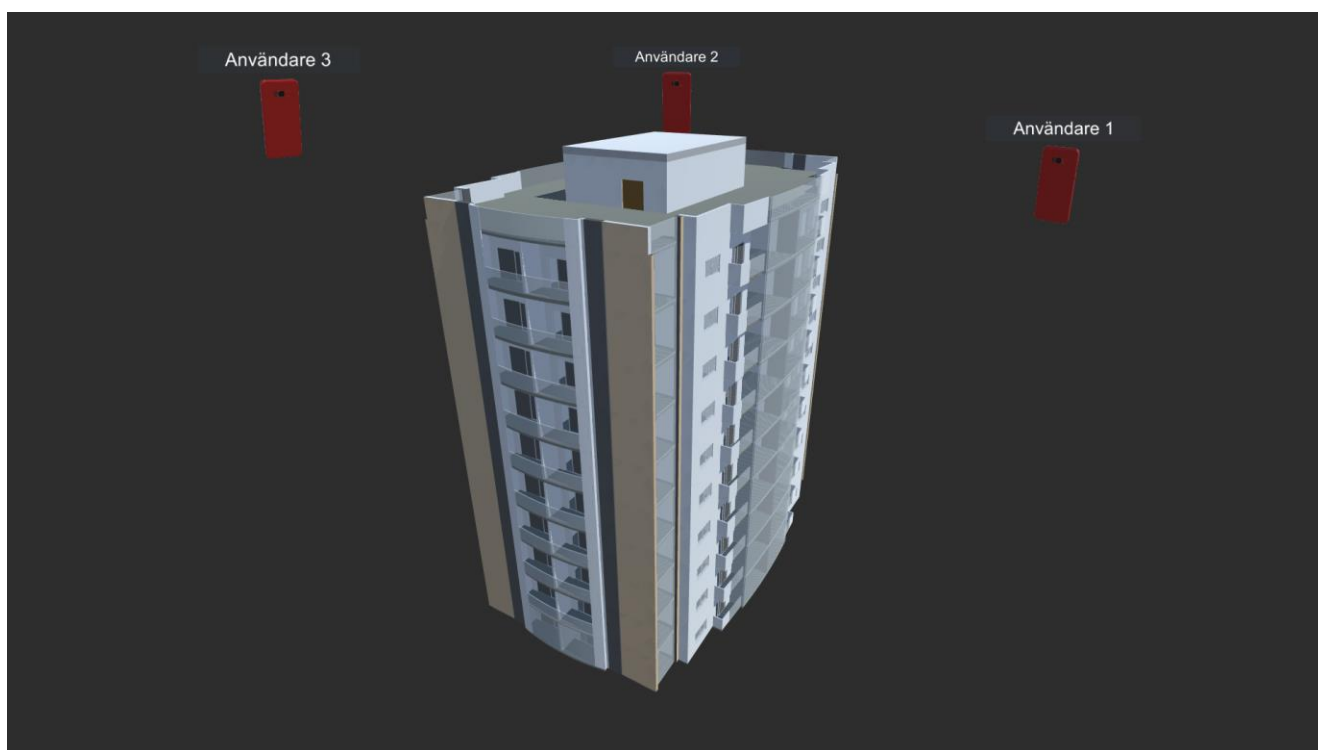
"Förstärkt verklighet ger en djuphets känsla i kombination med skuggning som inte är möjligt i en 3D vy. Jag kan föreställa mig hur byggnaden är utformad enklare. Därför kan det vara bra att använda sig av förstärkt verklighet när man inte har tillgång till en fysisk modell."

Nätverkssynkronisering

I denna delmetod skrivs skriptet som gör det möjligt att integrera föregående del i ett nätverkssynkroniserat system. Servern skapas hos en av användarna, det vill säga en användare är också nätverksvärden, som resterande användare ansluter sig till. I nuläget har nätverksvärden inga ytterligare behörigheter i jämförelse med andra klienter.

Positionsvisualisering av alla klienter utgörs av 3D-modeller i form av smarta mobiltelefoner, modell hämtad från Sketchfab (2017), se Figur 31. Dessa 3D-modeller representerar positionen samt rotationen för varje användare. Denna behöver vidareutvecklas genom färgspecifisering som är unik för varje användare.

Synkningsprocessen utförs i en tvåstegsprocess. Ett klick på knappen "Nätverkssynkronisering", nummer 11 i Figur 13, visar en popupmeny som innehåller en lista på namn för alla tillgängliga fjärranvändare, se Figur 32. I popupmenyn kan namnen klickas på vilket meddelar mjukvaran att börja synkningen till just denna användare. Synkningen kan avbrytas enkelt genom att klicka igen på knappen "Nätverkssynkronisering".



Figur 31 Röda 3D-modeller på smarta mobiltelefoner representerar positionen för fjärranvändare.



Figur 32 Lista på fjärranvändare som är tillgängliga för synkning.

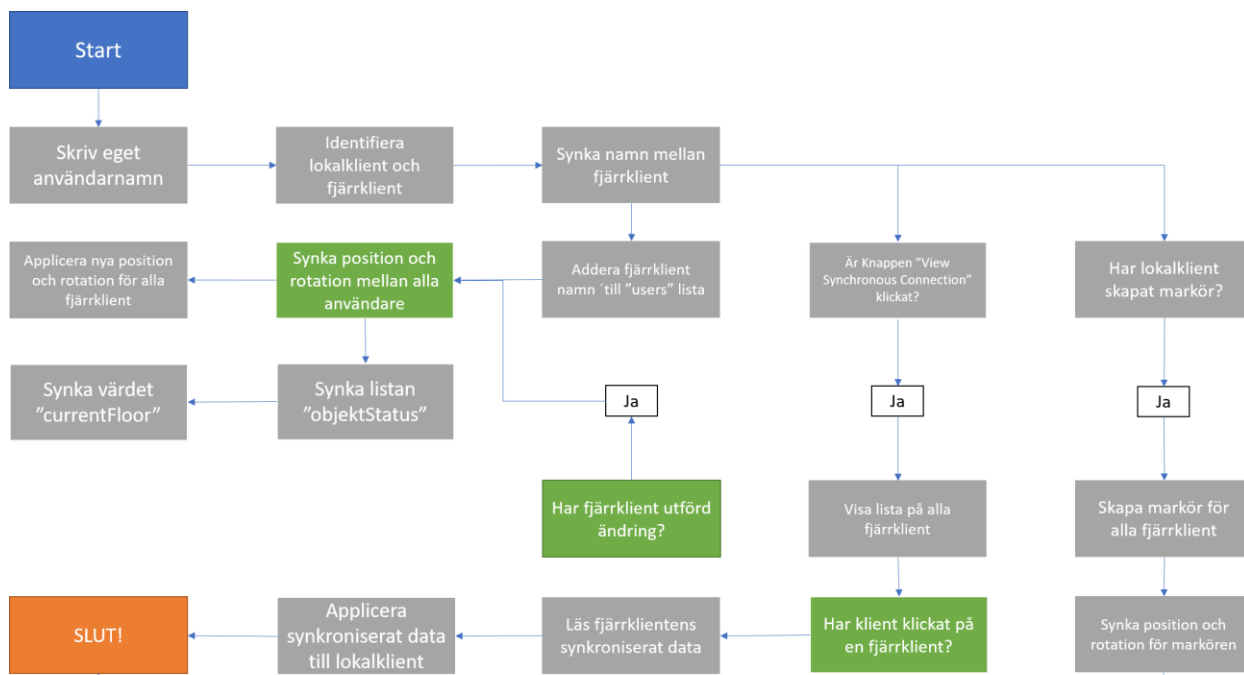
Skriptet, Figur 33, för nätverkssynkronisering börjar med att fråga efter ett användarnamn. Därefter kan man välja mellan att ansluta sig till en nätverksvärd eller själv vara nätverksvärd. I fall användaren väljer att själv vara nätverksvärd krävs också namn på servern. Namnet används bara för att identifiera och skilja mellan multipla servrar. Inne i varje användares mjukvara görs en skiljaktighet mellan lokalklient och fjärrklient. Mjukvaran på respektive smart mobiltelefon klassificerar dess användare som lokalklient samt alla övriga användare som fjärrklienter.

Det är endast lokalklienten som skickar data till servern, detta inkluderar position, rotation, "nuvarandeVåning", "objektStatus" och användarnamn. Positions- och rotationsdata skickas 29 gånger per sekund till servern, "nuvarandeVåning" och "objektStatus" skickas endast vid ändring och användarnamnet skickas endast en gång.

Data från servern synkas till klienterna och innehåller position, rotation, "nuvarandeVåning", "objektStatus" och användarnamn från alla fjärranvändare. Alla fjärrklienters namn som synkas adderas till en list "users", denna list används i popupmenyn som också inhämtar referens på objektet som äger användarnamnet. Då en klient kopplar från servern tas först klientens namn bort från list "users", sedan dess representativa 3D-modell och knapp i popupmenyn. Position och rotation för alla fjärranvändare uppdateras omedelbart med data som fås ifrån servern. Detta visualiseras via 3D-modellen för smarta mobiltelefonen som förflyttar sig omkring.

När lokalklienten skapar en ny objektmarkör skickas informationen om skapandet, inklusive position och rotation, till servern som synkar det vidare till alla fjärranvändare. Detta leder till att en ny objektmarkör skapas hos alla användare med exakt samma position och rotation.

När en lokalklient klickar på knappen "Nätverkssynkronisering", nummer 11 i Figur 13, synliggörs en popupmeny med knappar som innehåller användarnamn. Namnet som klickas på meddelar skriptet att användaren vill synka till just denna specifika fjärranvändare. Då tar skriptet fjärranvändarens position och rotation som appliceras till lokalklientens kamera, detta resulterar i att båda användarna får tillgång till samma perspektiv. Vidare jämför skriptet värden i list "objektStatus" och "nuvarandeVåning" mellan det som existerar hos lokalklienten och fjärrklienten. Vid olikhet skickar skriptet ett meddelande till skriptet "Våningsplanväljare" och "Lagervisningsväxlare" om att utföra ändringar så att värden ska stämma överens vilket resulterar i att även byggnadsmodellen ser likadan för båda klienterna.



Figur 33 Flödesschema för integration av synkron nätverkssynkronisering.

Deltagarna tycker att positionssynkningen samt visualisering av detta i förstärkt verklighet har varit intresseväckande, då man kunde se rörelsen av alla fjärranvändare i hög precision relativt till byggnadsmodellens position. I BIM ger positionssynkningen en uppfattning av vyn som andra aktörer tittar på, dock är det svårare att uppfatta vad dessa aktörer har sitt fokus på och distansen som dem tittar ifrån. Detta har inte varit problem i förstärkt verklighet, där man får en mycket bättre uppfattning över distans och position.

Deltagare 3 tycker att möjligheten till markörplacering som synkas mellan alla aktörer via nätverkssystemet är ett viktigt kommunikationsverktyg. Potentiellt kan det säkerställa en smidigare samt lättare kommunikation genom att se till att alla aktörer diskuterar om samma detalj eller byggnadsdel. Tredje deltagaren sa:

”Jag gillar markören, den är väldigt enkel att använda och är väldigt användbar. För en person som mig som inte kan byggnadsdelsnamn gör denna funktion kommunikationen med yrkesmän enklare”

Deltagare 1 tycker att möjligheten till att växla på/av skärmdelning är viktigt för att möjliggöra en interaktiv designdiskussion. Vid behov kan man slå på skärmdelningen för att presentera ett förslag/problem, annars så kontrollerar varje användare det dem vill se. Byggnadsförståelsen som fås av att se planlösningar eller 3D-modeller fritt är särskilt viktig för en icke erfaren person. Första deltagaren sa:

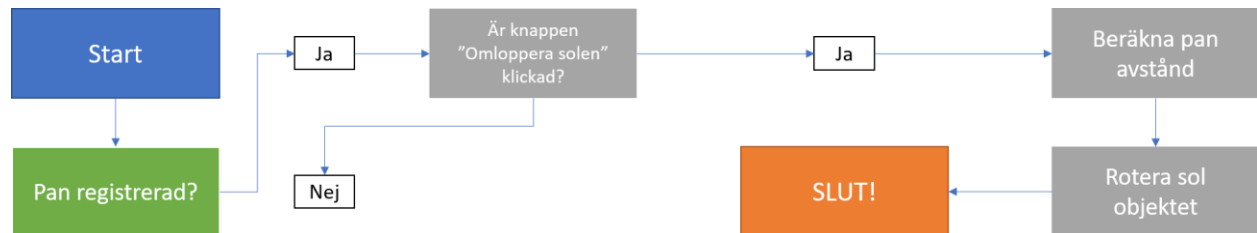
”Att kunna kontrollera och manövrera själv runt 3D-modellen ökar förståelsen som man får utifrån modellen. Kunden begränsas inte när en lösning behöver presenteras, skärmdelningen blir en simpel metod för kommunikation. Det ger oss det bästa av båda världar.”

Första deltagaren tycker att denna distribuerade kommunikationsmetod har stor potential, en utvecklingsmöjlighet för kommunikationsmetoden är integration av asynkron kommunikation vid sidan av synkron kommunikation. Detta kan ske genom addering av kommenteringsfunktionalitet där man kan skriva och spara kommentarer över molnet via användningen av markörer, chatt och skärmbilder. Röstinspelning kan också vara en möjlighet för asynkron kommunikation. Första deltagaren sa:

”Denna app har god potential i att möjliggöra distribuerat möte mellan många olika aktörer. En utvecklingsmöjlighet som jag ser är att göra det möjligt att kommunicera både synkront och asynkront.”

Realtidsskuggnings kontroll

Realtidsskuggningskriptet är simpelt, se Figur 34. När pan registreras kontrolleras om knappen "Omloppera solen", nummer 5 Figur 13, är klickat eller inte. Om "Omloppera solen" är klickat så beräknas avstånden på pan som adderas till rotationen av solobjektet som redan finns.



Figur 34 Flödesschema för realtidsskuggnings kontroll.

Skriptet är i en tidig utveckling, målet är att kunna kontrollera solpositionen manuellt genom att välja ort samt datum och tid. Denna utveckling möjliggör simulering av realtidsskuggstudie, en likadan metod används redan i många arkitektoniska mjukvaror, inklusive Revit och ArchiCAD. Solpositionssynkning ska också vara möjligt att utföra.

Diskussion och slutsats

Hur kan spelmotorer möjliggöra förstärkt verklighet och nätverkssynkroniserad distribuerad kommunikation?

Den genomförda studien bekräftar att spelmotorer såsom Unity har möjligheten till att skapa mjukvaror som utnyttjar sig av avancerade teknologin ARCore för skapandet av förstärkt verklighet. Detta i kombination med nätverksbaserade funktioner och nätverkssynkronisering via skriptning i det kraftiga skriptningsspråket C#, dessutom på relativ korttid. Realtidsvisualiseringen och nätverkssynkroniseringen som spelmotorn möjliggör ser till att den distribuerade kommunikationen sker med minimal fördröjning oberoende av distansen då data överförs mellan parter med försumbart dröjsmål. Studien visar också att smarta mobiltelefoner har nu förtiden möjligheten till att visualisera höghus i realtid utan optimering i kombination med interaktive funktioner.

Hur kan metoden för en förstärkt verklighets miljö beskrivas för distribuerad möte?

Deltagarna tycker att förstärkt verklighet ger en användare som är oerfaren med läsningen av planlösningar möjligheten för bättre förståelse över byggnadsvolymen samtidigt som det blir enklare att föreställa sig distans i jämförelse med traditionella 3d vy. Förstärkt verklighet ger användarna ett manövreringssystem som leder till att klienten kan bestämma sin vy genom att fysiskt röra sig till önskad position och riktning såsom om det är en fysisk modell man istället inspekterar. Låga kostnaden, tidskraven och ändringsmöjligheten relativt till fysiska modeller möjliggör utnyttjande av teknologin även i tidiga skede som leder till att distribuerade samarbetsmöten kan ske oftare och med mindre tid däremellan möten för ändringar. Förstärkt verklighet ger även klienten möjligheten till att kunna se bygganden i planerade byggplatsen och i verkliga skala.

Hur används nätverkssynkronisering i metoden för förstärkt verklighet?

Nätverkssynkroniseringen i detta arbete ser till att dela information såsom individuell position, riktning och många inställningar med alla parter som använder sig av systemet. Vid testet av mjukvaran beskrevs möjligheten att nätverkssynkronisera som att det kan minska risken för kommunikations fel, tack vare att distanskommunikationen blir tydligare med gränssnittet och enkelheten att utföra distanskommunikationen mellan aktörer. Deltagarna tycker också att friheten för vysynkning mellan olika användare är viktigt för att kunna växla emellan beroende på möte kraven, då olika krav finns för diskussion och presentation baserade möten. Synkning av position och riktning för alla användare har uppfattats som ett viktigt hjälpmedel för att nå målet med ett fysiskt möte på distans, dock så försvinner fortfarande en hel del information då ansiktsuttrycken inte visas alls.

Distribuerade kommunikationsverktyg i spelmiljöer

Utvecklingen av distribuerade kommunikationsverktyg baserad på synkron kommunikation är viktigt för en effektiv projekteringsprocess, som dessutom använder sig av redan befintliga verktyg för BIM. Det öppnar möjligheten för flera typer av möten inklusive designutvecklingsmöten på ett smidigt sätt. Spelmotorer såsom Unity har visat att dess användande möjliggör skapandet av interaktiva funktioner i en virtuell realtidsmiljö inklusive förstärkt verklighet. I ett experiment utfört av Billinghamurst & Kato (2002), testades skillnaden i kroppslig rörelse som sker på en videokonferens med vanlig bildskärm i jämförelse med en videokonferens i förstärkt verklighet. När bildskärmen användes lutade användaren sig framåt medan i förstärkt verklighet lämnade användaren istället ett personligt utrymme med virtuella personen som är likadant som i ett möte ansikte mot ansikte. Detta experiment visar att förstärkt verklighet kan möjliggöra en uppfattning av avstånd i distribuerade möten som utmanar den i traditionella möten. En likadan uppfattning har fått även i denna mjukvara, där deltagarna kunde identifiera dem andra aktörerna, dess position och riktningen som dem tittar på tydligare. Denna uppfattning stöds även av Rekimoto (1996) där testpersonerna effektivt kunde identifiera måtten på objekt i förstärkt verklighet samt positionen och placeringen med hjälp av virtuella skuggor. Fler test behövs för att vara säker, då utformningen av BIM-läget samt dess kontroll relativt till förstärkt verklighets kan vara orsaken till denna uppfattning.

Enligt Dong, o.a. (2013) så är den mest nödvändiga funktionen i ett distribuerat förstärkt verklighetsmöte avatarer som representerar aktörerna. Funktionens implementering för förstärkt verklighet i denna mjukvara inspirerades av co-located mötets användning av kroppsliga rörelser samt kamerariktningen. Med målet att återskapa fördelen med co-located möte i ett distribuerat möte. Därför används en smart mobiltelefons 3D-modell som följer användarens position samt riktning i fysiska världen relativt till 3D-modellen på byggnaden. Det är värt att undersöka andra alternativ till 3D-modeller, istället kan till exempel framkameran av smarta mobiltelefonen nyttjas för att visa användaren.

Distribuerade kommunikationsverktyget är inte begränsat till endast extern kommunikation såsom kommunikation mellan arkitekt och beställare, den kan lika gärna användas internt för granskning och projektering. Metoden för kommunikation är inte avgränsad till endast BIM, dock innebär detta att vissa funktioner som BIM är ett måste för inte kommer att ha normal funktion utan manuellt arbete.

BIM-integration och visualisering

Första integration av BIM i ett projekt till spelmotorn Unity har visat sig ta cirka 10 minuter. optimala målet är att det inte ska kräva några ytterligare processer eller tid, dock att tiden kan minska till cirka 3 minuter för framtida ändringar är ett steg i rätt riktning för ett helt automatiserat arbetsflöde. Att första samt resterande integrationer inte kräver särskilt mycket tid ökar potentialen till användning av mjukvaran även i tidiga skeden.

En grundläggande funktion för realtidsrendering är möjligheten till skapandet av interaktiva funktioner som kan ändras och uppdateras i realtid (Roupé, Johansson, & Bosch-Sijtsema, 2015). BIM-integrationsmetoden har visat möjligheten till skapandet av visualiseringsfunktioner i realtid som nyttjar dess data. BIM-visualiseringen gör det möjligt att visa information som för övrigt kan vara svåra att presentera, särskilt till mindre erfarna inom byggnationer vilket i sin tur kan leda till tidsbesparing och minskning av missförstånd. Det leder också till att beställaren begränsas mindre av arkitektens val och kompetens inom informationspresentation.

BIM-integrationen och visualiseringsmetoden har visat sig vara flexibel. Arkitekten har möjligheten till inkluderande, exkluderande och även skapande av parametrar som möjliggör anpassning av BIM-visualisering beroende på projektbehov. BIM-visualiseringen kan även integreras för skapandet av funktioner för just förstärkt verklighet och virtuell verklighet.

För att kommunikationsverktyget ska användas behövs effektivisering av BIM-integrationsmetoden mellan användarna. Den nuvarande integrationsmetoden kräver fortfarande uppdatering av appen/mjukvaran hos alla användare vilket kan göra BIM-visualiseringen besvärlig.

Förstärkt verklighet

Ur en arkitekts synvinkel är det önskvärt att ha kommunikationsverktyg som stödjer en förbättrad kommunikation då detta möjliggör snabbare konstruktion och granskningscykler (Broll, o.a., 2004). Förstärkt verklighet kan användas för att utveckla nya metoder för co-located och distribuerad kommunikation. Detta beror på möjligheten till att övergå smidigt mellan den fysiska världen och förstärkt verklighet samt tillgången till rumsliga signaler (Billinghurst & Kato, 2002). Den fysiska världen och förstärkt verklighet medför en naturligare navigation i jämförelse med 3D-mjukvaror, där den kroppsliga rörelsen definierar vyn som man ser vilket resulterar i en mer noggrann och användarvänlig navigation.

Uppfattning av avstånd fås redan av VR "virtuell verklighet" med potential till ännu mer interaktion tack vare möjligheten till handspårning, dock så har förstärkt verklighet fördelen när det kommer till tillgänglighet då den kräver endast smart mobiltelefon. Användningen av förstärkt verklighet isolerar inte användaren från dem andra aktörerna som är i samma fysiska rum i jämförelse med VR som blockerar den visuella kontakten genom VR-glasögon.

Förstärkt verklighet har visat sig vara en effektiv kommunikationsmetod för att sända rumsliga signaler i ett distribuerat möte, där kroppsliga rörelser samt kamerariktningen tydligt visualiserar

vad användaren gör. Anledningen till användningen av kamerariktningen istället för ögonkontakt är att användaren för det mesta har sin blick fokuserad på skärmen medan kamerans riktning är det som användaren egentligen tittar på. (Rekimoto, 1996).

Mjukvaran har kunnat visa en prototyp som är fungerande i Android smarta mobiltelefoner. Detta innebär dock inte att den är kommersiellt redo, då flera utvecklingsmöjligheter, problem samt buggar måste åtgärdas innan det blir praktiskt att användas. Detta inkluderar framsteg inom prestanda, batteriliv, BIM-integration och röstchatt. Bättre prestandaoptimering möjliggör användningen av 3D-modeller med högre detaljnivå samt mer realistiska modeller i en förstärkt verklighetsmiljö. Ingen modelloptimering har utförts på byggnadsmodellen som används i detta projekt, vilket inte bara visar den snabba utvecklingen och kapaciteten på smarta mobiltelefoner men också på att möjligheten för mer komplexa byggnationer och fysiska integrationer finns.

Realtidskuggning har visat sig vara särskilt viktigt i förstärkt verklighet för sammanlänkning och kombinerad med den fysiska världen. Ljussuppskattningsverktygen i ARCore kan förbättra sammanlänkningen med den fysiska världen genom att ha en naturligare ljussättning i virtuella modeller. Detta begränsar dock användarens möjlighet till att, på ett enkelt sätt, påverka ljussättningen och skuggningen av den virtuella modellen. Istället går det att använda en ljussättning som är oberoende av ARCore, som fortfarande kombinerar förstärkt verklighet med den fysiska världen detta fungerar dock inte lika bra i alla ljussättningar. För att kunna utföra skuggstudier behövs kontrollen finnas hos själva användaren vilket leder till att virtuell ljussättning krävs.

Förslag till fortsatta studier

Mixed reality som är kombinationen av förstärkt verklighet och virtuell verklighet kombinerar majoriteten av dess fördelar och nackdelar. Detta kan troligen utöka användningen av förstärkt verklighet där beroende på kraven för tillgänglighet och interaktion kan antingen förstärkt verklighet eller mixed reality nyttjas. För detta kan en studie på tillämpning av mixed reality i en distribuerad projektering utföras för att jämföras med detta arbete.

För att kommunikationsverktyget ska användas behövs effektivisering av BIM-integrationsmetoden mellan användarna. Där nuvarande integrationsmetoden fortfarande kräver uppdatering av appen/mjukvaran hos alla användare som kan göra BIM-visualiseringen besvärlig. Detta kan effektiviseras genom att istället använda sig av nätverksbaserade integrationsmetoder, som även kan utföras via användning av redan befintliga teknologier såsom AssetBundle. AssetBundle är en arkivfil som innehåller bland annat 3D-modeller, texturer och ljudklipp som kan läsas samt laddas ner från servern av spelmotorn Unity i körtid "runtime" (Unity, 2017). Detta är värt att studeras vidare för att se hur det utvecklar integrationen av BIM.

Utförande av simuleringar i förstärkt verklighet kan kanske möjliggöra bättre förståelse och i kombination med BIM i spelmotorer kan arbetsflödet för dess utförande förenklas.

Markörlös handspårning och gestspårning i kombination med förstärkt verklighet i smarta mobiltelefoner kan möjliggöra en bättre distribuerad kommunikation samt ökad interaktion med BIM i realtid.

Referenser

- Alamparambil, J., & Mowrer, T. (2017). *Unite Austin 2017 - How to Leverage ARKit and ARCore with Unity* [Videokonferens].
- Allen, T. (1984). *Managing the Flow of Technology: Technology Transfer and the Dissemination of Technological Information Within the R&D Organization*. Cambridge: MIT Press.
- Augment. (2018, April 24). *Better present & sell your building projects through augmented reality*. Retrieved from Augment: <http://www.augment.com/augmented-reality-architecture/>
- Autodesk. (2014, April 18). *Element ID*. Retrieved from Autodesk: <https://knowledge.autodesk.com/support/revit-products/learn-explore/caas/CloudHelp/cloudhelp/2014/ENU/Revit/files/GUID-455BA5CC-EA25-4AF1-8E6F-3B54E19AD8E1-htm.html>
- Autodesk. (2018, April 10). *About Schedules*. Retrieved from Autodesk: <https://knowledge.autodesk.com/support/revit-products/learn-explore/caas/CloudHelp/cloudhelp/2018/ENU/Revit-DocumentPresent/files/GUID-73090B70-8A13-4E12-909C-F25D724D5BA7-htm.html>
- Autodesk. (2018, Mars 5). *Building information modeling software*. Retrieved from Autodesk: <https://www.autodesk.com/products/revit/features>
- Billinghurst, M., & Kato, H. (2002). Collaborative Augmented Reality . *Communications of the ACM - How the virtual inspires the real* , 64-70.
- BIM Alliance Sweden. (2017, Februari 27). *BIM Alliance om BIM*. Retrieved from BIM Alliance Sweden: <http://www.bimalliance.se/vad-aer-bim/bim-alliance-om-bim/>
- BIM Alliance Sweden. (2017, mars 3). *BIM förbättrar kontrollen över byggprocessen*. Retrieved from BIM Alliance Sweden: <http://www.bimalliance.se/vad-aer-bim/exempel/link-arkitekter/>
- Boeykens, S. (2014, Oktober 10). *Getting BIM data into Unity (Part 1 - ArchiCAD)*. Retrieved from CAD, BIM & 3D: <http://cad-3d.blogspot.se/2014/10/getting-bim-data-into-unity-part-1.html>
- Broll, W., Lindt, I., Ohlenburg, J., Wittkämper, M., Yuan, C., Novotny, T., . . . Strothmann, A. (2004). ARTHUR: A Collaborative Augmented Environment for Architectural Design and Urban Planning. *JVRB - Journal of Virtual Reality and Broadcasting*, 1-9.
- Callewaert, C. (2013). *Unite 2013 - Architectural Visualization with Unity: From Revit to Unity to Rift* [Video].
- Craig, A. (2013). Understanding augmented reality: concepts and applications.

- Davidsson, P., & Thoresson, A. (2017). *Svenskarna och internet 2017*. Stockholm: Internetstiftelsen i Sverige.
- Dong, S., Behzadan, A., Chen, F., & Kamat, V. (2013). Collaborative visualization of engineering processes using tabletop augmented reality. *Advances in Engineering Software Volume 55*, 45-55.
- Edwards, G., Li, H., & Wang, B. (2015). BIM based collaborative and interactive design process using computer game engine for general end-users. *Visualization in Engineering 3:4*, 2.
- Furht, B. (2011). *Handbook of Augmented Reality*. New York City: Springer.
- Google. (2018, Maj 8). *ARCore Overview*. Retrieved from Google: <https://developers.google.com/ar/discover/>
- Google ARCore. (2018, Maj 11). *Quickstart for Android*. Retrieved from Google: <https://developers.google.com/ar/develop/unity/quickstart-android>
- Graphisoft. (2018, April 5). *About BIM*. Retrieved from Graphisoft: http://www.graphisoft.com/archicad/open_bim/about_bim/
- Hansson, B., Olander, S., Landin, A., Aulin, R., & Persson, U. (2015). *Byggledning - Projektering*. Lund: Studentlitteratur.
- Hedin, A., & Martin, C. (2011). *En liten lathund om kvalitativ metod med tonvikt på intervju*. Retrieved from Studentportalen: <https://studentportalen.uu.se/uusp-filearea-tool/download.action?nodeId=459535&toolAttachmentId=108197>
- Heimgartner, J. (2016, April 13). *What Is Augmented Reality and How Can Engineers and Designers Use It?* Retrieved from engineering.com: <https://www.engineering.com/DesignSoftware/DesignSoftwareArticles/ArticleID/11873/What-Is-Augmented-Reality-and-How-Can-Engineers-and-Designers-Use-It.aspx>
- Hrastinski, S., & Keller, C. (2007, Mars 21). *När är det lämpligt att stödja synkron kommunikation i nätbaserade kurser?* Retrieved from LearningNet: <http://webnews.textalk.com/learning-net/nar-ar-det-lampligt-att-stodja-synkron-kommunikation-i-natbaserade-kurser>
- Jeppsson CAD Center. (2014). *Lär dig Revit architecture 2015 grunder*. Göteborg.
- Johansson, M. (2016). From BIM to VR - The design and development of BIMXplorer. *Chalmers University of Technology*, 5-12.
- Kitchens, K., & Shiratuddin, M. (2007). Interactive home design in a virtual environment. *7th International Conference on Construction Applications of Virtual Reality*, 1-12.

- Lanham, M. (2018). *Learn ARCore - Fundamentals of Google ARCore*. Birmingham: Packt Publishing Ltd.
- Liu, Y., Tanudjaja, G., Jiang, Z., & Beck, N. (2016). Workflow of exporting Revit models to Unity. *Computer Integrated Construction Research Group - Penn state CIC research* , 3-27.
- Maing, M. (2012). Physical or Virtual? : Effectiveness of virtual mockups of building envelope systems. *Georgia Institute of Technology School of Architecture*, 15.
- Mateus, D., Sousa, M., Klerk, R., Gama, S., Jorge, J., & Duarte, J. (2015). From Τέκτων to Τέχνη: Going Back to the Classical Roots of. *Proceedings of the 33rd*, 107-109.
- Mira, L. (2018). *Unity at Game Developers Conference - Building an ARCore app: Learnings and Challenges* [Videokonferens].
- openBIM. (2011). *The Sky is the limit* [Video].
- Prabhu, S. (2017, December 18). *Types of Augmented Reality (for Me and my Business)*. Retrieved from ARreverie: <http://www.arreverie.com/blogs/types-of-augmented-reality/>
- Rekimoto, J. (1996). Transvision : a hand-held augmented reality system for collaborative design. *Proceeding of Virtual Systems and Multimedia*.
- Roupé, M., Johansson, M., & Bosch-Sijtsema, P. (2015). Real-time visualization of building information models (BIM). *Automation in Construction Volume 54*, 69-82.
- Rüppel, U., & Schatz, K. (2011). Designing a BIM-based serious game for fire safety evacuation simulations. *Advanced Engineering Informatics Volume 25, Issue 4*, 600-611.
- Sabbagh, K. (1996). *Twenty-First-Century Jet: The Making and Marketing of the Boeing 777*. New York City: Scribner.
- Saunders, M., Lewis, P., & Thornhill, A. (2009). *Research Methods for Business Students*. Harlow: Pearson Education.
- Schell, M., & Turner, K. (2017). *Unite Europe 2017 - Performance optimization for beginners* [Videokonferens].
- Shamsee, N., Klebanov, D., Fayed, H., Afrose, A., & Karakok, O. (2015). *CCNA Data Center DCICT 640-916*. Indianapolis: Cisco Press.
- Sketchfab. (2017, April 2). Galaxy S8. Eternal Realm.
- Stichnoth, P. (2018, 6 8). *Wayfair technology blog*. Retrieved from Wayfair: <https://tech.wayfair.com/2018/06/everywhere-you-look-computer-vision-at-wayfair/>

- Svensk Byggtjänst. (2016). *Besparingsmöjligheter genom effektivare kommunikation i förvaltningsprocesser*. Helsingborg: Industrifakta.
- Törlind, P. (2015). Collaborative Design. *Journal of the Indian Institute of Science*, 354-360.
- Unity. (2017, Maj 15). *AssetBundles*. Retrieved from Unity:
<https://docs.unity3d.com/Manual/AssetBundlesIntro.html>
- Unity. (2018, April 24). *Diagnosing performance problems using the Profiler window*. Retrieved from Unity : <https://unity3d.com/learn/tutorials/temas/performance-optimization/diagnosing-performance-problems-using-profiler-window>
- Unity. (2018, April 24). *Game engines—how do they work?* Retrieved from Unity:
<https://unity3d.com/what-is-a-game-engine>
- Zhou, F., Duh, H., & Billinghurst, M. (2008). Trends in Augmented Reality Tracking, Interaction and Display: A Review of Ten Years of ISMAR . *ISMAR '08 Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*, 193-202.