

Udvikling af en prisbillig bilsimulator

Samt statiske og dynamiske objekters indflydelse på fornemmelsen for tilstedeværelse



Institut for Datalogi
Softwareingeniør
Aalborg University
Efterår 2010

Titel

Udvikling af en prisbillig bilsimulator - samt statiske og dynamiske objekters indflydelse på fornemmelsen for tilstedeværelse.

Semestertema

Informationssystemer - HCI

Projectperiode

SW9, efterår 2010

Projektgruppe

Gruppe d511a, rum 5.2.38

Vejleder

Jan Stage

Synopsis

De fleste undersøgelser lavet af bilsimulatorer i dag, har ikke fokus på hvilke faktorer der giver følelsen af tilstedeværelse. Når følelsen af tilstedeværelse bliver nævnt, er det oftest som en del af selvkritikken om hvorfor resultater kan være usikre, da brugere ikke føler sig på samme måde til stede i en simulator som i en rigtig bil. Det kunne derfor være interessant at undersøge om det er muligt at udvikle en bilsimulator for begrænsede omkostninger med fokus på spørgsmålet om, hvorvidt statiske og dynamiske objekter i det virtuelle miljø er variable der har indflydelse på følelsen af tilstedeværelse. Et eksperiment blev udført på 16 testpersoner som alle gennemkørte 2 scenarier á 4 conditions hvor manipulation med statiske og dynamiske objekter blev foretaget. Resultatet af eksperimentet støttede ikke op om hypoteserne og kun få af værdierne var signifikante. Dog var de signifikante resultater interessante da de belyste nogle problemstillinger som var oplagte at fokusere på i en fremtidig fortsættelse af studiet om følelsen af tilstedeværelse i simulatorer.

Forfattere

Kasper Guldbrand, Rene Bach Gustafson, Kaspar Lyngsie & Kim Fiedler Vestergaard

Meta

Sideantal: 93, Appendikser: 2, Antal kopier: 6, Slutdato: 6. januar 2011

The content of this report is available for free, but publication (with source) may only occur with agreement from the authors.

Forord

Denne rapport dokumenterer udviklingen af et semesterprojekt lavet af softwarestuderende i gruppen d511a på Institut for Datalogi på Aalborg Universitet, i perioden efterårssemesteret SW9 2010, med projektstart den 1. september 2010. Denne rapport dækker udviklingen af en prisbillig bilsimulator med det formål at undersøge hvorvidt manipuleringen med visse elementer i det virtuelle miljø kan påvirke en bilist fornemmelse for tilstedeværelse. Projektvejlederen dette semester var Jan Stage, som vi gerne vil takke for sin hjælp gennem forløbet. Vi vil ydermere gerne takke samtlige testdeltagere der deltog i eksperimentet. Denne rapport fokuserer på emner relateret til softwareingeniørarbejde og det er antaget at læsere af denne rapport er videnskabsmæssigt på niveau med en softwareingeniørstuderende på 9. semester. Når der refereres til citeret materiale vil der fremstå som et nummer i hårde parenteser efter en sætning eller et ord, således: [17]. Atypiske forkortelser eller betegnelser er beskrevet yderligere i fodnoter, og når der refereres til elementer fra implementation er disse fremhævet i **typewriter**-formatet. Når der gøres brug af kodeeksempler er disse indkapslet som listings ligesom listing 1:

Listing 1: *sit/til/fil* - kodeEksempel functionen.

1 `Kodeeksempel`

Bibliografien findes i kapitel 10.

Underskrifter

Kim Fiedler Vestergaard

Rene Bach Gustafson

Kasper Guldbrand

Kaspar Lyngsie

Indholdsfortegnelse

1	Introduktion	6
2	Litteraturstudie	8
2.1	Begrænsede omkostninger	8
2.1.1	Starte fra bunden	9
2.1.2	Finde et open source projekt	9
2.1.3	Bygge videre på eksisterende virtuelle verdener	10
2.2	Forbedring af enkelte dele	10
2.3	Større opsætning	11
2.4	Sammenligning af bilsimulatorer	12
2.4.1	Andet litteraturstudie	13
2.5	Kørselsopsætning - Simulator vs. rigtig kørsel	13
2.6	Validering af bilsimulatorer	14
2.7	Fokus for projektet efter litteraturstudiet	14
3	Krav	16
3.1	Krav til den fysiske opsætning	16
3.2	Krav til udviklingsmiljø	17
4	Teknologier	19
4.1	Valg af udviklingsmiljø	19
4.1.1	Introduktion	19
4.1.2	Undersøgelse af muligheder	21
4.1.3	Konklusion	22
4.2	Skabe 3D model af virtuelt miljø	23
5	Simulator	27
5.1	Unity 3	27
5.1.1	Unity 3 som udviklingsplatform	27
5.1.2	Unity 3 scripting editor	28
5.1.3	Disse ting er allerede implementeret i Unity Studios bilspilprojekt	29
5.1.4	Disse ting skulle tilføjes og implementeres	29
5.2	Implementering af rat og pedal-sæt, samt ændre bilfysik	30
5.3	Trafik	33
5.3.1	Waypoints	34
5.3.2	Hastighed	35
5.3.3	Retning	37
5.3.4	Trafiksikkerhed og sidste udvej	39
5.4	Implementation af fodgænger	43
5.5	Logning	44
5.6	Tidsforbrug og linjer kode	46

6	Eksperiment	47
6.1	Introduktion	47
6.2	Opsætning	48
6.3	Deltagere	49
6.4	Software	50
6.5	Procedure	50
6.6	Databehandling	51
6.6.1	Bilens hastighed og placering	52
6.6.2	Øjefokuseringer	52
6.6.3	Svar på spørgsmål	54
7	Resultater	55
7.1	Hastighedsfejl	55
7.2	Vejbanefejl	56
7.3	Øjefokusering	57
7.4	Interview	57
7.4.1	Forudsætninger	58
7.4.2	Overblik	59
7.4.3	Sammenfatning	59
7.4.4	Opfattelse af simulatoren	60
7.4.5	Følelse af tilstedeværelse	62
7.4.6	Forslag til forbedringer	64
8	Diskussion	65
8.1	Fejlgrupper	65
8.1.1	Forskel imellem forskellige scenarier	65
8.1.2	Sværhedsgraden af scenarierne	66
8.1.3	Behandlingen af de stillede opgaver	66
8.2	Fremtidigt arbejde	66
8.2.1	Simulator	66
8.2.2	HUD - Head-Up Display	68
9	Konklusion	69
10	Bibliografi	71
A	Appendikser	74
A.1	Interviews	74
A.1.1	Drenge	74
A.1.2	Piger	87

1

Introduktion

Når man arbejder inden for verden af transportmidler, bliver der ofte gjort brug af simulatorer som normalt har til formål at efterligne transportmidlets adfærd i et kontrolleret miljø. Eksempelvis er det normalt for studerende der uddanner sig til pilot, først at gennemgå øvelser i en simulator før de sætter sig ind i et rigtigt fly. Ligeledes kan man også finde simulatorer hvis man uddanner sig til styrmand på et fragtskib, men også til brug af mindre maskiner som lastbiler, almindelige biler og alle mulige andre transportmidler[2]. Ofte er grunden til brug af simulatorer forbundet med økonomiske og sikkerhedsmæssige spørgsmål, såsom prisen på brændstof og faren for ulykker ved brug af en uerfaren fører af transportmidlet.

Simple bilsimulatorer er til rådighed for almindelige mennesker, primært fra spillekonsolmarkedet, hvor et bilspil sælges sammen med rat og pedaler som sættes til fjernsynet eller computeren, hvilket ifølge producenterne[17] skulle forhøje følelsen af tilstedeværelse for kunden. Bilsimulatorer i transportindustrien er komplekse og kostbare, i Japan har Toyota siden 2007 haft rekorden for den mest avancerede simulator i verden, der måler over 550 m^2 og vejer næsten 78 tons[8]. Men hvordan afgør man, hvad det er der gør en bilsimulator god eller dårlig? Det kunne være interessant at undersøge mulighederne, udfordringerne og problemstillingerne i at udvikle en simpel bilsimulator for begrænsede midler, og undersøge om et sådan produkt til trods for begrænsningerne stadig kan give brugbare resultater i et kontrolleret testmiljø.

Brugen af en simulator kan medvirke som løsning på problemer som de ovenstående, men omvendt kan det også åbne for nye spørgsmål. I artiklen "Visual Longitudinal and Lateral Driving Assistance in the Head-Up Display of Cars"[18] fra 2007, benytter Tönnis et al. en bilsimulator for at undersøge om det er muligt at forbedre billisters overholdelse af hastighedsbegrænsningen og placering på vejbanen ved hjælp af et Head-Up Display. Det fremgik i resultaterne, at en testperson generelt var villig til at køre hurtigere jo sikrere personen følte sig, og der bliver i diskussionen blandt andet diskuteret, hvorvidt forsøgspersonerne havde en falsk følelse af sikkerhed fordi forsøget ikke blev udført i en rigtig bil. Artiklen er interessant grundet dens opsætning og brug af en bilsimulator, men ydermere grundet dens

manglende besvarelse af spørgsmålet om, hvad der kunne være med til at påvirke denne følelse af sikkerhed i en bilsimulator. Ligeledes i ”Evaluating Driver Attention and Driving Behaviour: Comparing Controlled Driving and Simulated Driving” [4] fra 2008, sammenligner Bach et al. forsøgspersoners kørselsadfærd i et henholdsvis simuleret og virkeligt miljø. Her tales der om en følelse af tilstedeværelse i henhold til fysiske og funktionelle faktorer (hvor meget simulatoren føltes og lød som en rigtig bil og hvordan den virtuelle verden lignede den rigtige verden) fra det virkelige og simulerede miljø som faktorer der kan påvirke forsøgspersoners kørsel. Artiklen er interessant, fordi der bliver stillet spørgsmålstegn ved hvorvidt disse faktorer kan have indflydelse på testpersonernes kørsel, men eftersom artiklens fokus var på et andet eksperiment, bliver spørgsmålet ikke besvaret. Artiklen nævner, at simulatormiljøet kan have bidraget positivt til testpersonens følelse af tilstedeværelse, hvorimod manglen på det rummelige visuelle indtryk og bevægelser fra styretøjet kan have modvirket følelsen af tilstedeværelse.

Da det fortsat er uklart, hvilke faktorer der har indflydelse på en testpersons følelse af tilstedeværelse i en bilsimulator, kunne netop dette være et interessant spørgsmål at undersøge, og hvordan disse kan manipuleres til at påvirke og muligvis forbedre en testpersons køreegenskaber i et simuleret miljø. Yderligere er det nærliggende at undersøge udfordringerne i at udvikle bilsimulator for begrænsede midler, der giver mulighed for at manipulere med de faktorer som vi, på baggrund af et litteraturstudie, mener har indflydelse på følelsen af tilstedeværelse.

På baggrund af denne introduktion, kan vi stille følgende spørgsmål som vi ønsker at undersøge nærmere:

- Hvilke faktorer påvirker testpersoners følelse af tilstedeværelse i en bilsimulator?
- Hvordan udvikler man en prisbillig bilsimulator, der har mulighed for at manipulere med disse faktorer?
- Hvilke udfordringer, begrænsninger og problemstillinger er der ved at udvikle en prisbillig bilsimulator?

2

Litteraturstudie

I dette kapitel vil der tages udgangspunkt i et udpluk af artikler som kan relateres til arbejdet med udviklingen af en bilsimulator hvad enten udviklingen af den starter fra bunden, bygges på et open source projekt eller bygges indenfor en eksisterende virtuel verden. Desuden er der også en opdeling i hvorvidt artiklerne har fokus på det at udvikle simulatorer hurtigt og for begrænsede omkostninger eller projekter i den anden ende hvor omkostningerne skubbes i baggrunden og fokus i stedet er på at højne følelsen af tilstedeværelse i simulatoren. Foruden at fokusere på følelsen af tilstedeværelse i selve simulatoren er der også inddraget artikler hvor der er fokuseret på at forbedre enkelte dele af simulatoren for at den overordnede følelse af tilstedeværelse hos brugeren af simulatoren optimeres. Udover dette bliver der inddraget artikler med sammenligning som det gennemgående tema hvor der i en artikel sammenlignes forskellige klasser af simulatorer og i en anden artikel med forskellige bilsimulatorer inddraget kigges på følelsen af tilstedeværelse samt fornemmelsen for farts indflydelse på dette. Som en del af sammenligningerne er der også en artikel der bygger på et litteraturstudie af en række artikler hvor der kigges på hvilke målinger der kan bruges til at bestemme graden af tilstedeværelse hos testpersonerne, samt i hvilke kørselsopsætninger artiklerne har udført deres eksperimenter. Dette bringer det videre til det næste tema omkring kørselsopsætningen hvor der i en artikel kigges på problemstillingen om der skal testes i en simulator eller i en rigtig bil på lukket bane og hvordan disse kan være et alternativ til rigtig kørsel. Til sidst kigges der på en artikel hvori der er fremstillet en alternativ metode der sammen med følelsen af tilstedeværelsen kan være med til at validere bilsimulatorer på brugervenligheden og dermed også give en idé om hvor god simulatoren er.

2.1 Begrænsede omkostninger

De følgende artikler omhandler alle udviklingen af bilsimulatorer for begrænsede omkostninger. Først er omtalt artikler hvor simulatoren er udviklet fra bunden af, dernæst en artikel hvor simulatoren af baseret på et open source bilspil og til sidst en artikel hvor simulatoren er bygget i en eksisterende

virtuel verden.

2.1.1 Starte fra bunden

Udviklingen af bilsimulatorer fra bunden af kan findes omtalt i "A PC-based Driving Simulator Using Virtual Reality Technology" [9] og "Effects of Heightened Sensory Feedback to Presence and Arousal in Virtual Driving Simulators" [14]. I [9] er simulatoren bygget på en konstrueret database som er i stand til at afvikle grafik for rimelige omkostninger på PC'er. Fra denne database genereres de forskellige dele af simulatoren i form af veje, huse, trafik, osv. løbende fra en overordnet model alt efter hvor i modellen bilen befinder sig på det givne tidspunkt. I tilfældet med [14] er simulatoren opbygget ved hjælp af open source API'er (Application Program Interfaces) som giver udviklerne adgang til en række data og tjenester inden for både den grafiske og lydmæssige del af simulatoren. Lee og Kim pointerer også at ens indlevelse og opfattelse af omgivelserne er en stærkt medvirkende faktor i evnen til at bedømme og vurdere ens muligheder i omgivelserne. Det er samtidig også denne indlevelse og opfattelse af omgivelserne som Lee og Kim betegner som følelsen af tilstedeværelse og som de beskriver det måles det i hvilken grad man føler sig "inde" i den virtuelle verden som i dette tilfælde er en bilsimulator. I begge omtalte artikler ovenfor er der på trods af valget med at starte udviklingen af simulatoren fra bunden af stadig fokus på at udvikle simulatorer med lave omkostninger og simulatorer der forholdsvis hurtigt kan færdiggøres til brug i testøjemed.

2.1.2 Finde et open source projekt

I stedet for at udvikle en bilsimulator fra bunden, kan man med fordel også tage udgangspunkt i eksisterende open source simulatorer eller bilspil. Ved at modificere koden eller lave tilføjelser til det eksisterende har man en billig metode til at designe sin bilsimulator som man ønsker det. Denne metode har de eksempelvis brugt i "Developing a Low-Cost Driving Simulator for the Evaluation of In-Vehicle Technologies" [27] hvor Weinberg og Harsham forsøger at skabe en virkelighedstro simulator for få midler og hovedsageligt baseret på masseproduceret billigt hardware. Sammen med denne billige hardware bruges et eksisterende bilspil hvor man selv kan designe baner, biler og indstille kameravinkler og det er desuden muligt at logføre data omkring position og fart med et plug-in API. Brugen af et bilspil lægger begrænsninger på friheden, da den lukkede kode eksempelvis forhindrer manipulering med trafik og tilførelse af nye elementer som vejskilte og trafiklys. Trods dette, betegner Weinberg og Harsham det som en hurtig og billig måde at konstruere en bilsimulator på, som kan benyttes til at teste køretøjsmonteret udstyr så længe man er opmærksom på dens begrænsninger.

2.1.3 Bygge videre på eksisterende virtuelle verdener

Udover artikler om brugen af eksisterende bilspil som bilsimulatorer eller valget af at udvikle en simulator fra bunden af, har man også forsøgt sig med at udvikle bilsimulatorer i eksisterende virtuelle verdener for at kunne bygge ovenpå disse frem for at udvikle en ny behovstilpasset bilsimulator. Dette gør sig eksempelvis gældende i "Using Second Life to Demonstrate a Concept Automobile Heads Up Display" [7], hvor man har konstrueret en fungerende bilsimulator i den online virtuelle 3D-verden, Second Life ¹. Ved at bruge Second Lifes grafikmotor indeholdende data om tyngdekraft og kollision mellem genstande i 3D-universet, er det muligt at scripte en bil der kan manøvrere i den virtuelle verden og samtidig overholde de fysiske love. Desuden kan både statiske objekter, såsom skilte og dynamiske objekter, såsom trafik i form af andre biler, let tilføjes og bruges som en del af bilsimulatoren, da de allerede findes som objekter i den virtuelle verden. Dette giver ifølge Chu og Joseph tilsammen en hurtigere og mindre omkostningsfuld måde at udvikle bilsimulatorer på, der kan bruges i testsituationer hvor det der skal testes, ikke kræver en dyrere og mere realistisk simulator end det som Second Life tilbyder.

2.2 Forbedring af enkelte dele

Det er dog ikke kun det at udvikle en bilsimulator fra bunden af eller gøre det ved hjælp af eksisterende bilsimulatorer og/eller eksisterende virtuelle verdener der har relevans i forbindelse med at udvikle en bilsimulator der giver brugerne en følelse af tilstedeværelse. Der er også fokus på det at optimere små dele af simulatoren, så bilsimulatoren som helhed fremstår mere i overensstemmelse med virkeligheden. Dette gør sig eksempelvis gældende i "Developing a 3D Sound Environment for a Driving Simulator" [19] hvor Maurant og Refsland har forsøgt at forbedre en bilsimulator på lydsiden. For eksempel, har de tilføjet motorstøj alt efter hvor meget motoren arbejder og hvinende dæk både på ens egen bil og de andre biler der optræder i simulatoren. Derudover har de også tilføjet larmen fra vind, horn, blinklys, kollisioner og endda sirener fra udrykningskøretøjer som er modelleret til at overholde reglerne om dopplereffekten. Tilføjjelsen af disse lyde er ifølge Maurant og Refsland med til at give et helhedsbillede af en bilsimulator der gør at følelsen af tilstedeværelse forbedres om end de anerkender at lydsiden selvfølgelig ikke er det eneste man kan og bør forbedre sig på.

I en anden artikel, "A Framework for Modeling Human-Like Driving Behaviors for Autonomous Vehicles in Driving Simulators" [1], beskriver Al-Shihabi og Maurant hvordan de har udviklet et framework der kan modellere menneskelignende kørsel blandt de biler der udgør trafikken i bilsimulato-

¹<http://secondlife.com/whatis/?lang=en-US>

ren. Modellen er delt op i 4 dele som hver især står for de dele der ville have betydning for en bilists beslutninger og handlinger i trafikken. Først opfanges de forskellige sanser som bilisten oplever i øjeblikket og dernæst beskrives det i modellen hvordan bilisten rent følelsesmæssigt reagerer på dette i sine omgivelser. Derefter søger modellen efter beslutninger der kan tilfredsstille bilistens følelsesmæssige krav. Til sidst prøver modellen så at udføre de beslutninger så snart det er sikkert nok i forhold til omgivelserne. Udover disse forskellige stadier har bilisterne også forskellige menneskelige karaktertræk som vil have betydning i de forskellige stadier af modellen. Ved at gøre bilerne mere menneskelignende ved f.eks. både at indeholde kørslen for aggressive og forsigtige bilister og også at kunne simulere alkoholpåvirkede eller trætte bilister vil simulatoren ifølge forfatterne kunne fremstå mere uforudsigelig og dermed også give brugerne en større følelse af tilstedeværelse ved at modellere den trafik som vi omgiver os med i en rigtig kørselssituation.

2.3 Større opsætning

I modsætning til artiklerne omhandlende udviklingen af en bilsimulator med lave omkostninger, har man også forsøgt sig med større opsætninger af en bilsimulator som i "A High-fidelity Immersive Cluster-based Driving Simulator for Transportation Safety Research" [15] hvor fokus i langt højere grad er at gøre simulatoren så realistisk som mulig og give en høj grad af følelsen af tilstedeværelse, så den kan bruges til at teste spørgsmål om vejsikkerhed. Simulatoren blev udviklet til brug af U.S. Department Of Transportation til at sikkerhedsteste forskellige ting lige fra opmærkning af veje og fortov til innovative apparater designet til at kontrollere trafik. På grund af denne brug af simulatoren var kravene at både det visuelle og lydmæssige skulle matche kørsel som det ville foregå i den virkelige verden og kravene var generelt sat højt mens omkostningerne med udviklingen af simulatoren var sat i baggrunden. Derfor havde Liao til mål at udvikle en kompleks bilsimulator fra bunden af der skulle være i stand til at modellere en kompleks vejbane og vejside geometri bygget på topologi fra virkelighedens verden sammen med realistiske bygninger der er med til at give en rigtig bedømming af afstande. Den skulle bl.a. indeholde realistisk vegetation, skilte, fortov og kunne håndtere realistiske kørselsscenarier både om natten og dagen, hvor der ved natkørslen også var taget højde for realistisk brug af forlygter samt genskin på skilte og andre objekter. Desuden var der stillet høje krav til at systemet var i stand til at køre både grafikken og lydsporet realtime i overensstemmelse med bilens bevægelse og kunne gengive alle lyde som bilen og trafik producerer i et komplekst lydsystem. Foruden disse krav er bilsimulatoren baseret på et multi projektor output, således at det med flere projektorer kan repræsentere et bredt kurvet billede af bilistens syn og at man derved

også kan se ud af bilens sideruder. For at kunne bruge alle disse output og samtidig kunne trække bilsimulatoren er det hele sat sammen i et cluster af flere PC'er. Til sidst for at validere simulatoren er det Liao's hensigt at bruge vejdata fra virkeligheden og sammenligne det med vejdata fra simulatoren for at se i hvilken grad disse data stemmer overens med hinanden.

2.4 Sammenligning af bilsimulatorer

For at kunne se på fordele og ulemper ved forskellige typer af bilsimulatorer både hvad angår hvordan de er udviklet og med hvilket formål de er udviklet kan der kigges på artikler hvor forskellige bilsimulatorer sammenlignes. En sådan artikel er "Driving Simulators For Commercial Truck Drivers - Humans In The Loop" [2], hvor der i et studie kigges på kørselspræstationen i et system der bruges som læringsværktøj til lastbilchauffører der er i gang med at tage kørekort. Dette system testes på fire forskellige klasser af simulatorer fra helt dyre og meget komplekse lastbilsimulatorer bygget på platforme med bevægelse i, automat og manuelt gear, luftbremser, bakspejl og 270 graders visuel synsvinkel (level 4) i den ene ende. Ydermere til mindre omkostningsfulde og simple simulatorer uden gear og luftbremse og med kun 180 graders synsvinkel der er bygget til at kunne køres fra en enkelt PC (level 1) i den anden ende. Midt imellem placerer sig så level 3 og level 2 simulatorerne, hvor level 3 har en kabine svarende til en rigtig lastbil samt 180 graders synsfelt og skifte mellem luftbremser og manuel/automat gear, mens level 2 simulatoren er realistisk med hensyn til gear og luftbremser men kun kan levere mindre end de 180 graders synsfelt. Resultaterne fra eksperimentet viste at klassen med de dyre level 4 simulatorer gav de bedste kørepræstationer, men at der ikke var nogen signifikant forskel mellem level 3 og level 2 simulatorerne. Sidst viste det sig at level 1 simulatorerne klarede sig væsentligt bedre end forventet og Allen og Tarr påpeger vigtigheden af dette grundet den store prisforskel mellem de forskellige klasser af simulatorer. Allen og Tarr slutter derfor af med at konkludere at alle klasser af simulatorer har sin berettigelse, så længe simulatoren bliver brugt til at teste det den er designet til at teste.

I et andet studie, "Evaluating perception in driving simulation experiments" [10], kigges der på testpersonernes opfattelse af sanseindtryk i en række forskellige bilsimulatoreksperimenter og hvilke sanseindtryk der er vigtigst og har størst indflydelse på kvaliteten i at gengive kørsel og give en følelse af tilstedeværelse i en simulator. Kemeni og Panerai fremhæver fornemmelsen for fart og afstand til andre biler eller objekter i omgivelserne som en vigtig og vital egenskab i ens kørsel da det konstant bruges til at manøvrere bilen, som f.eks. hvornår og hvor hårdt der skal bremses for at undgå forhindringer. Det påpeges at nogle studier har vist at farten i bilsimulatorer med et tilstræk-

2.5. KØRSELSOPSÆTNING - SIMULATOR VS. RIGTIG KØRSEL

keligt stort synsfelt kan bedømmes tilfredsstillende ud fra det visuelle alene. Andre studier derimod har afsløret at vestibulære signaler, som har med ligevægt og balance at gøre, er vigtige i opfattelsen af distancer og fart, hvilket dermed har betydning når der skal manøvreres i et bilsimulator eksperiment. Generelt anbefaler Kemeni og Panerai at man i bilsimulatoreksperimenter gør brug af et stort visuelt synsfelt og gør brug af motionparallakser for at gøre følelsen af bevægelse mere mærkbar og dermed også højne følelsen af tilstedeværelse for testpersonerne.

2.4.1 Andet litteraturstudie

I artiklen der er skrevet på baggrund af et litteraturstudie, "Interacting with In-Vehicle Systems: Understanding, Measuring and Evaluating Attention" [5], har man kigget på 100 eksisterende artikler for at undersøge hvordan og under hvilke forhold køretøjsmonterede systemer evalueres og hvordan graden af opmærksomhed og tilstedeværelse måles i brugen af disse systemer. For at sortere artiklerne har Bach et al. opstillet 4 forskellige kategorier for scenerierne artiklerne gør brug af: Uden kørsel, simuleret kørsel, kontrolleret kørsel (køreteknisk anlæg) og kørsel i rigtig trafik. Udover denne opdeling i scenerier bliver opmærksomhedsmålingerne også inddelt i 5 kategorier: Primær opgaveløsning, sekundær opgaveløsning, øjefokusering, fysiske målinger og subjektive vurderinger. Litteraturstudiet giver et overblik over hvordan og i hvilke omgivelser man tester systemer til biler og hvilke ting man kan måle på når bilsimulatorer skal testes for graden af tilstedeværelse. Denne kategorisering af artiklerne viste at de fleste studier var udført i simulatorer og kørsel i rigtig trafik, mens hvor vidt man holder vejbanen og farten samt øjefokuseringer er de mest brugte målemetoder i eksperimenterne.

2.5 Kørselsopsætning - Simulator vs. rigtig kørsel

Til spørgsmålet om hvorfor det i hele taget er brugbart at gøre brug af bilsimulatorer i forhold til kørsel i en rigtig bil har man i "Evaluating Driver Attention and Driving Behaviour: Comparing Controlled Driving and Simulated Driving" [4] opstillet et eksperiment hvor man sammenligner det at teste på en lukket bane (køreteknisk anlæg) med test i en bilsimulator. Målet med eksperimentet er at finde forskelle og ligheder mellem disse to metoder, samt kigge på omkostningerne der er forbundet med dem og hvorvidt det kan betale sig at teste systemer i en rigtig bil frem for kun i simulatoren. Resultaterne af eksperimentet viste at de to forskellige metoder ledte til en lang række ens resultater, men at fokuseringer med øjet afveg fra vejen flere gange og i længere tid i kørslen på den lukkede bane og der opstod flere kørefejl i kørslen med bilsimulatoren. Bach et al. mener at de længerevarende fokuseringer væk fra vejen skyldes, at testpersonerne var mere i kontrol med situationen og derfor tillod sig selv at holde øjnene væk fra vejen i

længere tid. Med hensyn til det større antal kørefejl i bilsimulatoren mener Bach et al., at dette kan skyldes manglen på sanseindtryk i forbindelse med kørslen hvilket gør sig gældende i deres evne til at holde farten og føle sig tilstede under kørslen. Udover disse testresultater viste det sig, at metoden med at teste på en lukket bane havde større omkostninger både hvad angår tid og penge i forberedelsen og den efterfølgende analyse.

2.6 Validering af bilsimulatorer

I mange af de ovenstående artikler har følelsen af tilstedeværelse været et vigtig element i det at designe en bilsimulator og dermed også med til at validere hvad der gør en færdig simulator god. Men selvom dette er en parameter der kan være med til at validere en bilsimulator er der ikke nogen fast metode der overskueliggør hvordan man validerer en simulator. En sådan metode har de forsøgt at opsætte i "Developing a Validation Methodology for Educational Driving Simulators and a Case Study" [24], hvor H. Sancar et al. har udviklet en alternativ metode til at validere bilsimulatorer med fokus på brugervenligheden. Den udviklede metode blev brugt til at validere en lastbilsimulator som var designet til træne lastbilchauffører i at køre med bedre brændstoføkonomi og resultaterne fra eksperimenter med dette system blev opsamlet ved hjælp af interviews og observation. For at validere simulatoren for brugervenlighed kiggede H. Sancar et al. i deres model på brugen og genkendelsen af simulatorens dele, mens de opstillede formålet med design af simulatoren, simulatorklassen og demografiske karakteristika som de faktorer der kunne påvirke resultatet af brugen og genkendelsen. Så som det fremgår af denne metode kan genkendelsen og brugen af simulatorens dele (rat, pedaler, gear, osv.) som en brugervenligheds-parameter, ligesom følelsen af tilstedeværelse, være en indikator for kvaliteten af simulatoren og ifølge H. Sancar et al. være med til at validere en bilsimulator.

2.7 Fokus for projektet efter litteraturstudiet

Når man har med kørselsrelaterede problemstillinger at gøre indenfor HCI regi, så er det store spørgsmål ofte hvordan eksperimentet skal udføres, samt hvorvidt der skal gøres brug af en bilsimulator eller forsøg ude i felten. Som det også fremgår af [5] er forsøg i felten hvad enten det er kontrolleret kørsel eller kørsel i rigtig trafik ikke nær så hyppigt forekommende som eksperimenter der udføres i en bilsimulator. Dette kan hænge sammen med en række forskellige faktorer som gør kørsel i en rigtig bil problematisk som f.eks. det rent sikkerhedsmæssige perspektiv og et eventuelt erstatningsansvar hvis der skulle opstå ulykker. Men udover den sikkerhedsmæssige fordel ved at udføre eksperimentet i en simulator er den største fordel ifølge Weinberg og Harsham at kunne gentage eksperimentet, og måle på de samme variabler

2.7. FOKUS FOR PROJEKTET EFTER LITTERATURSTUDIET

hver gang, ved at bibeholde scenariet fra testperson til testperson under ens forudsætninger i et laboratorium [27]. Et andet argument for at afprøve forskelligt køretøjsmonteret udstyr i en simulator fremfor en bil er de omkostninger både i tid og penge der er forbundet med det at lave en fysisk prototype der skal monteres fremfor først at skabe en virtuel prototype der kan testes i simulatoren og tilpasses løbende før den produceres i fysisk stand [9]. Så generelt kan det siges at bilsimulatorer kommer til sin ret når nye idéer skal testes hurtigt, uden store omkostninger og under gentagelsesbare forhold som f.eks. i den indledende fase af et produkts udvikling eller når selve testen af produktet er sikkerhedsmæssigt uforsvarligt at teste i en rigtig bil og/eller i rigtig trafik.

Litteraturstudiet har vist at mange artikler omkring bilsimulatorer har haft følelsen af tilstedeværelse som et vigtigt mål for den færdige simulator og som det er sammenfattet i [10] har fornemmelsen for fart og afstand betydning for denne følelse af tilstedeværelse. Med udgangspunkt i dette vil vi gerne undersøge hvilke faktorer der har indflydelse på testpersoners fornemmelse for fart og dermed også følelsen af tilstedeværelse i forbindelse med forsøg i en bilsimulator.

En vigtig del af det at føle farten både i en simulator og i en rigtig kørselssituation er det visuelle aspekt og især hastigheden og størrelsesforholdene på omgivelserne der passerer under kørslen bruges til at bedømme afstande og den fart hvormed man selv bevæger sig [10] [15]. Som en del af omgivelserne optræder både statiske objekter, eksempelvis bygninger, træer og vejskilte, og dynamiske objekter, eksempelvis fodgængere og trafik i form af andre biler, og det kunne derfor være interessant at undersøge hvilken indflydelse disse to typer af omgivelser har på tilstedeværelsen hos en testperson.

3

Krav

Dette afsnit beskriver nogle af de krav vi på forhånd havde til vores simulator, inden vi tog vores endelige valg af udviklingsmiljø. De krav vi har sat til vores simulator er lavet ud fra at vi gerne ville bygge en bilsimulator for begrænsede omkostninger, samtidigt med at den skulle forsøge at afspejle virkeligheden så godt som muligt.

Vores krav vil være splittet op i to dele, som vil være krav til den fysiske opsætning, og dernæst krav til selve udviklingsmiljøet.

3.1 Krav til den fysiske opsætning

Vores krav til den fysiske opsætning var at det skulle gøres realistisk, samtidigt med at det skulle være forholdsvis billigt at opstille. Vores liste af krav til den fysiske opsætning var følgende:

Opstilling

- Opsætningen skal gøre brug af et bilsæde.
- Bilsædet skal være justerbart.

Funktion

- Rat og pedal-sæt med tilhørende gearstang.
- Der skal være en fungerende kobling.
- Gearstangen skal gøre brug af seks trin.

Visuelt

- Brug af skærm til forrude.
- Brug af skærme der kan agere sideruder.

Dataopsamling

- Kamera der optager nærbillede af testpersons øjne.
- Kamera der optager opgaveløsningen.
- Det kørende skærbillede skal optages.

Hele denne opsætning skulle kunne sættes op ved hjælp af billige materialer. Vi fandt det vigtigt at der kunne gøres brug af et rat og pedal-sæt med tilhørende kobling, samt 6 trins gearstang, da det tilsvarende den mest gængse opsætning i danske biler. Samtidigt ville vi simulere brug af sideruder ved hjælp af ekstra skærme, da det er vigtigt med et bredt synsfelt[27].

3.2 Krav til udviklingsmiljø

Vores krav til udviklingsmiljøet blev set ud fra at vi helst ville have så mange muligheder som muligt, for at ændre i vores bilsimulator. Derfor var det vigtigt at udviklingsmiljøet gav mulighed for at lave nemme og hurtige ændringer. Vores liste af krav til udviklingsmiljøet var følgende:

Udviklingsmiljø

- Man skal kunne ændre i alt.
- Billigt og hurtigt at udvikle i.

Dataindsamling

- Kunne indbygge automatisk logning af data.

Krav til eksisterende kode

- Mulighed for at benytte en relativ færdig bilfysik og omgivelser.
- Overskuelighed af eksisterende, samt ny kode.
- Bruger et højniveau kodesprog vi kender godt.

Virtuelle miljø

- Fuld kontrol over miljøet.
- Mulighed for at lave bykørsel.
- Kunne tilføje trafik som er selvstyret.

KAPITEL 3. KRAV

Grænseflade

- Skal kunne udvikle styring til rat og pedal-sæt med kobling.

Med kravene fastlagt for både den fysiske opsætning og udviklingsmiljøet, kunne vi herfra gå videre med at finde det rette udviklingsmiljø og dernæst begynde på selve udviklingen af simulatoren.

4

Teknologier

Dette kapitel omhandler overvejelserne der ligger bag valget af et udviklingsmiljø og valg af software til generering af et virtuelt miljø.

4.1 Valg af udviklingsmiljø

Denne sektion indeholder information omkring valg af udviklingsmiljø. Først vil der være en beskrivelse af hvilke valg vi har, dernæst en liste af de forskellige udviklingsmiljøer vi har kigget på, og til sidst en begrundelse for det valg vi har taget. Vores valg undervejs er blevet taget ud fra de krav der tidligere er beskrevet i afsnit 3.

4.1.1 Introduktion

Som beskrevet i litteraturstudiet, se kapitel 2, er der flere muligheder vedrørende valg af udviklingsmiljøer, når man skal udvikle en bilsimulator med lave omkostninger.

Disse er som følger:

- Starte fra bunden.
- Bygge videre på eksisterende virtuelle verdener.
- Finde et open source projekt.

Fordelene ved hver af disse er meget forskellige, derfor bør vi overveje de muligheder vi har, således at de passer sammen med de krav vi stiller, samtidigt med at det skal kunne realiseres, inden for den tid vi har til rådighed.

Starte fra bunden

Hvis man vælger at lave en simulator fra bunden er det klart at man ikke får alle fordelene ved at benytte sig af noget eksisterende. Det betyder dog at man helt frit kan vælge hvordan man vil bygge sin simulator, og at man ikke på nogen måde er begrænset af hvad man kan lave.

Det store spørgsmål man bør stille sig selv, når man vælger at starte fra bunden, er om man har tiden til det. Det kræver mere tid at starte fra bunden, da man ikke har mulighed for at springe over de mere trivielle udviklingsopgaver, der skal danne grundlaget for en bilsimulator. Derfor skal der bruges tid på at udvikle selve omgivelserne og hvordan bilfysikken skal fungere, før man kan begynde at lave ændringer i det man reelt set har et ønske om at teste i sin simulator.

Bygge videre på eksisterende virtuelle verdener

Når man tager en allerede eksisterende virtuel verden, og vil tilføje ting således at det passer sammen med de behov man måtte have, er fordelene at man som regel kun behøver at udføre et mindre stykke arbejde før man har det ønskede resultat. Det kommer af, at der i den eksisterende virtuelle verden allerede findes de fleste mekanismer og værktøjer, man skal bruge for at danne sit realistiske miljø. Samtidigt ved man, at alle disse ting på forhånd er testet af slutbrugere, hvilket betyder at man som udgangspunkt kun behøver at bekymre sig om at teste sine tilføjelser.

Ulempen ved at benytte en eksisterende virtuel verden, er at mulighederne for at tilføje nyt materiale kan være begrænsede, og hvis der er mulighed for at tilføje noget nyt, kan der være restriktioner på hvilke ting man har lov til at tilføje. Det betyder at man skal prøve at finde en eksisterende virtuel verden som ligger meget tæt op af de krav man har, således at mængden af ændringer er minimale og realistiske i forhold til de muligheder der er. Samtidigt er det vigtigt at de krav man har, ikke pludselig kan ændre sig undervejs, sådan at man kan ende i en situation, hvor man midt i en udviklingsperiode, finder ud af at et nyopkommet krav ikke er muligt at realisere, fordi udviklingsmiljøet ikke understøtter det. Med andre ord, skal man altså sikre sig at der er lav usikkerhed i de krav man har opsat på forhånd, hvis man har tænkt sig at bygge videre på en eksisterende virtuel verden.

Finde et open source projekt

Fordelen ved at vælge et open source projekt, er at kildekoden er frit tilgængelig og at der derfor ikke er nogen begrænsninger for hvilke ændringer der kan laves. Der er også rig mulighed for at få hjælp med sin udvikling, da der ved open source projekter oftest er tilsluttet et stort community af udviklere, som alle har ekspertise i netop udvikling på projektet.

Udfordringen ved open source projekter er dog at de tit er mindre færdiggjorte og man har en tendens til at støde på flere fejl og mangler.

Vælger man et open source projekt, må man regne med at det kræver mere tid til udvikling i forhold til at bygge videre på eksisterende virtuelle verdener. Men samtidigt er open source projekter langt mere tolerante overfor ændringer i kravspecifikationen til simulatoren, da man hurtigt kan få

kvalificeret hjælp fra communityet til at lave sine ændringer.

4.1.2 Undersøgelse af muligheder

Vi har fundet frem til nogle forskellige bud på udviklingsmiljøer, der kunne være interessante i forhold til de krav vi stiller. Først vil vi undersøge Second Life, som passer under kategorien *eksisterende virtuelle verdener*. Dernæst undersøger vi to *open source projekter*, Torcs og Dolphinity Racer. Til sidst undersøger vi en kombination af de to kategorier *open source projekt og starte fra bunden*, som er Unity 3 sammen med et bilspilprojekt fra Unity Studios.

Second Life

I vores litteraturstudie, blev det nævnt at der tidligere var lavet eksperimenter med en bilsimulator i Second Life. Fordelen ved at gøre brug af Second Life, er at vi derved kan gøre brug af en allerede eksisterende verden, hvor der allerede er understøttelse for statiske/dynamiske objekter som huse, vejskilte, mennesker og biler. Dog er mulighederne for at udvikle ting i Second Life begrænsede, da der bliver gjort brug af et scripting sprog kaldet LSL (Linden Scripting Language). LSL kan bl.a. bruges til at manipulere objekters bevægelse, hvilket eksempelvis kan være at få en bil til at køre, men Second Life sætter samtidigt nogle restriktioner for hvad man kan lave, med henblik på sikkerhed men også ydeevne.

Second Life understøtter ved brug af LSL at man kan lave målinger på objekters bevægelse, hvilket betyder at man ved brug af Second Life har mulighed for at logge bevægelses data[16].

Torcs

The Open Racing Car Simulator, er et open source bilsimulatorprojekt. Torcs har support for AI styrede biler, og bliver bl.a. brugt som research platform da det indeholder flere simulationsfunktioner som kollisioner, egenskaber for dæk og hjul, aerodynamik etc. Kildekoden er skrevet i C++ og frigivet under GPL (General Public License), hvilket vil sige at vi har fri mulighed for at ændre i koden så den svarer til de krav vi måtte have. Torcs er et projekt der hovedsageligt sætter fokus på racerløb og har derfor ikke umiddelbart understøttelse for bykørsel[25].

Dolphinity Racer

Racer er et open source bilsimulatorprojekt, som forsøger at simulere bilfysik med en realistisk føling. Simulatoren forsøger at opnå realisme ved at fokusere meget på det grafiske aspekt, ved at bruge en grafikmotor med

understøttelse for bl.a. røgpartikler, reflektering, skygger og lysskær. Samtidigt har Racer support for at konvertere logfiler fra kørslen, således at de kan analyseres i Matlab, der gør det muligt at lave en mere dybdegående analyse af resultaterne. Der er i Racer mulighed for at tilpasse en masse ting vedrørende biler og baner, ved at ændre på mindre ASCII filer, som gør det muligt at lave mindre ændringer uden specifikt at ændre i kildekoden. Kildekoden er ligeledes tilgængelig, dog kun på nuværende tidspunkt i en ældre version, da der stadigvæk arbejdes på at udgive en nyere og opdateret udgave af kildekoden[22].

Unity 3 og Unity Studios bilspilprojekt

Unity 3 er et spiludviklingsværktøj som sætter fokus på at gøre det nemt at udvikle spil. Den primære måde at udvikle spil i Unity er ved brug af grafiske redskaber, og sekundært er der mulighed for scripting i Mono, som er en open source implementation af .NET frameworket. At Unity 3 gør brug af Mono betyder også at der er mulighed for at udvikle i bl.a. C# og JavaScript, hvilket er med til at kunne lette udviklingsopgaver betydeligt.

At udvikle en bilsimulator i Unity 3 fra bunden ville dog være et større projekt, og derfor findes der også såkaldte miniprojekter som kan benyttes. En af disse er et bilspil udviklet af Unity Studios, og ligger frit tilgængeligt til brug sammen med Unity 3. Bilspilprojektet indeholder egenskaber og styring af en bil, samt en prækonstrueret bane til at køre på.

Fordelen ved at bruge Unity 3 til udvikling af en bilsimulator, er at der ikke sættes nogen begrænsninger for hvilke tilføjelser der kan laves. Dog er der begrænsede muligheder til rådighed fra start af, og der kan derfor være en større mængde arbejde i at få tilføjet alle de mangler der måtte være[26].

4.1.3 Konklusion

Vi var fra start af klar over, at der skulle være rig mulighed for at tilpasse vores simulator efter vores behov, derfor lagde vi fokus på at undersøge mulighederne indenfor open source markedet. Vores valg af udviklingsmiljø blev i sidste ende Unity 3, hvilket der er flere grunde til.

Fravalg af eksisterende virtuel verden (Second Life)

Med Second Life blev vi klar over, at mulighederne vi havde for selv at tilpasse en bilsimulator efter vores behov ville være meget begrænsede. Der er i Second Life mange begrænsninger på hvad man har lov til at konfigurere, derfor turde vi ikke tage chancen, skulle vi pludselig stå i en situation hvor udvikling af noget vigtigt ikke ville være muligt. Ser vi på vores krav i afsnit 3, er det bl.a. muligheden for at have fuld kontrol over verdenen, som ikke kan overholdes i Second Life.

Fravalg af de to open source projekter (Torcs/Dolphinity Racer)

Fælles for flere af de open source bilsimulatorer som vi kiggede på, herunder bl.a. Torcs og Dolphinity Racer, var at de alle virkede meget uoverskuelige og forholdsvis svære at tilpasse, selvom det var muligt at tilgå kildekoden. Flere af disse bilsimulatorer var også henvendt til racerløb, frem for almindelig hverdagskørsel som ligger nærmere vores krav.

Vi så hurtigt at skulle vi tilpasse disse open source projekter så de tilsvarende vores krav, skulle vi først til at bruge rigtig lang tid på at sætte os ind i projektet og dets muligheder. Samtidigt var kildekoden til de forskellige projekter ikke altid lige nem at modificere, og der ville være en lang indlærings tid før vi ville kunne starte med at udvikle. De krav fra afsnit 3 som ikke bliver overholdt, er bl.a. muligheden for at kunne lave ændringer hurtigt, samt at den eksisterende kode skal være overskuelig.

Valg af Unity 3

Vi valgte at gøre brug af Unity 3, hvor den primære grund er at mulighederne for at ændre og tilpasse bilsimulatoren efter vores behov er gode. Samtidigt er det også betydeligt lettere for os at udvikle i Unity 3, da det giver os mulighed for at ændre i bilsimulatoren ved hjælp af grafiske værktøjer. Unity 3 understøtter ligeledes udviklingssprog som C# og JavaScript, som vi har erfaring med fra tidligere studier.

Vores valg af Unity 3 er også begrundet ved at vi kunne drage nytte af fordelene ved to forskellige udviklingsscenarier, nemlig ved at kombinere udvikling fra bunden, sammen med noget eksisterende som er open source. Vi fik derfor fordelene ved at kunne udvikle i præcis det miljø hvor vi ville have de mest optimale betingelser, sammen med et eksisterende open source projekt fra Unity studios, som indeholder en masse af startbetingelserne for en bilsimulator, herunder bilfysik med tilhørende grafik.

4.2 Skabe 3D model af virtuelt miljø

Som vi har skrevet i afsnit 3 omkring krav til vores simulator, så skulle vi have lavet to forskellige scenarier der kunne køres i, et med bykørsel og et med landevej. For at skabe disse to miljøer undersøgte vi først hvilke muligheder der var til rådighed, hvorefter et valg blev taget ud fra disse kriterier:

- Skal være nemt at konstruere omgivelserne
- Hurtigt at lave ændringer på miljøet
- Må ikke have for store omkostninger
- Skal generere et miljø med lav kompleksitet

Blandt disse kriterier er det vigtigste især at det skulle være nemt at konstruere omgivelserne, da vi ikke ville bruge for lang tid på at skabe de forskellige scenarier, samtidigt med at det helst skulle være let at ændre i dem. Det er ligeledes vigtigt at kompleksiteten af miljøet er i bund, da vi skal have projiceret vores simulator op på en stor skærm, som kræver en høj opløsning og derfor sætter store krav til den maskine der skal køre vores applikation.

Som beskrevet i afsnit 4.1, fulgte der sammen med bilspilprojektet fra Unity Studios, en bane med som kunne benyttes. Men vi endte med ikke at bruge denne, da vi ville have svært ved at skabe to scenarier (by og landevej), som svarede til hinanden i sværhedsgrad ud fra banen der fulgte med.

Vi undersøgte derfor hvilke muligheder der var af værktøj til at skabe et virtuelt miljø i Unity 3, der kunne hjælpe med at lette arbejdet, frem for at udvikle et virtuelt miljø fra bunden.

De forskellige muligheder vi overvejede, var følgende:

- Skabe et virtuelt miljø fra bunden.
- Bruge 3D model af Aalborg fra Aalborg Kommune.
- CityEngine.
- CityScape.

Følgende giver en kort beskrivelse af de forskellige muligheder vi havde, og derefter vil der komme en begrundelse for vores endelige valg.

Skabe et virtuelt miljø fra bunden

Ved brug af Unity 3 er der rig mulighed for at lave et virtuelt miljø fra bunden af, hvor man helt selv bestemmer detaljegraden af det man laver. Man vil så selv kunne vælge om man vil tegne alle bygninger og objekter fra bunden af, i 3D programmer som eksempelvis Blender¹, eller om man vil gøre brug af objekter lavet på forhånd af tredjepart. Om man vælger det ene eller det andet, vil det tage længere tid end hvis man gjorde brug af et værktøj der kunne autogenerere en by. Fordelen ved selv at konstruere miljøet, er at man er inde over alle aspekter og har 100% kontrol over hvordan miljøet skal se ud. Ulempen er at man selv skal være meget striks med at sørge for at kompleksiteten er i bund, samtidigt med at det kan være meget tidskrævende, da der i teorien kan bruges uger på blot at lave et enkelt objekt som eksempelvis en bygning, da det afhænger af den ønskede detaljegrad.

¹<http://blender.org>

Bruge 3D model af Aalborg

Aalborg Kommune har lavet en 3D model af Aalborg midtby, som frit kan downloades, og benyttes af Teknik- og Miljøforvaltningen i det daglige arbejde til vurdering af kommende projekter.

For at anvende denne model skal man benytte sig af AutoCad, SketchUp eller 3D Studio Max, og da Unity 3 har mulighed for import af bl.a. data fra 3D Studiomax, har vi mulighed for at importere hele modellen ind i Unity 3 og derved gøre brug af modellen.

Da vi undersøgte modellen fandt vi dog ud af at detaljegraden af modellen var alt for stor. Det ville simpelthen ikke være muligt at gøre brug af modellen sammen med vores simulatorprojekt, som kræver at der kan navigeres flydende i modellen. Der var alt for mange polygoner på de enkelte objekter, samtidigt med at modellen var utrolig stor, hvilket gjorde at selv en kraftfuld maskine ville have svært ved at behandle modellen tilfredsstillende[11].

CityEngine

CityEngine er et værktøj til at skabe 3D modeller på en nem måde, ved bl.a. at give mulighed for import af data fra kortinformationer gemt i eksempelvis GIS ².

CityEngine er ikke et gratis værktøj, og deres mindste udgave koster omkring 2.700 kr. hvilket ikke helt passer sammen med vores krav om at det skulle være en simulator bygget for begrænsede omkostninger. Dog har CityEngine en demo-version, hvor en mindre del af funktionerne er tilgængelige. Da vi undersøgte denne demo, fandt vi ud af at de modeller af bygninger som de gjorde brug af, igen var meget detaljerede, men dette kunne vi løse ved at importere nogle andre modeller med en mindre detaljegrad.

Problemet med demoen var dog endvidere at vi ikke kunne komme til at eksportere vores projekt til Unity 3, da man så skulle have den købte udgave af programmet. Dette gjorde at vi helt måtte fravælge at gøre brug af CityEngine[21].

CityScape

CityScape er som CityEngine et værktøj til at skabe 3D modeller på en nem måde. CityScape er som CityEngine ikke noget gratis værktøj, men igen her er der mulighed for at gøre brug af en demo version med begrænsede funktioner. Denne demo har dog mulighed for at eksportere til Unity 3, samtidigt med at den gør brug af nogle mindre komplekse modeller af bygninger, hvilket betyder at det vil køre bedre i Unity 3.

I CityScape har man mulighed for selv at tegne sit vejnet, hvorefter man kan vælge at den selv skal generere bygninger langs et ønsket stykke vej. Det-

²<http://www.procedural.com/cityengine/>

KAPITEL 4. TEKNOLOGIER

te gør det utroligt let at skabe sit miljø, dog skal man være påpasselig med ikke at fylde alt for mange bygninger på, da flere objekter selvfølgelig betyder at det stiller større krav til maskinen der skal visualisere modellen[20].

Vores valg

Vi valgte at gøre brug af CityScape. Da vi fra start af havde et ønske om at benytte et værktøj til at generere vores 3D model, var der til sidst ikke så mange valgmuligheder, da CityScape simpelthen var det eneste værktøj der kunne eksporteres til Unity 3, samtidigt med at det ikke havde for komplekse modeller.

5

Simulator

Dette kapitel omhandler de forskellige skridt i udviklingen af software-delen af bilsimulatoren.

5.1 Unity 3

Dette afsnit vil give et indblik i hvad Unity 3 er for en udviklingsplatform, samtidigt med at det vil give svar på hvilke ting vi allerede har at arbejde videre med, og hvilke ting vi skal have udviklet før at alle vores krav er opfyldt.

5.1.1 Unity 3 som udviklingsplatform

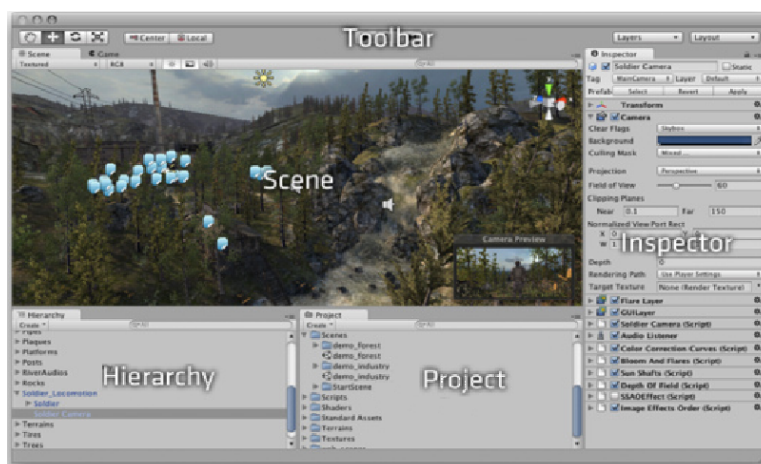
Unity 3 er et stykke spiludviklingssoftware, som er specifikt designet med henblik på at udvikling af spil skal være nemt. Det gør den ved at give en kombineret mulighed for både at ændre i spilfysikken ved hjælp af en editor, samtidigt med at der er mulighed for at ændre i spilfysikken ved at manipulere med spillet på det grafiske plan.

Det hele er opsat på en måde så det er nemt at skifte imellem at tilføje nyt til koden, ændre lidt i det grafiske miljø, tilføje nye objekter fra biblioteket osv. Figur 5.1 giver et overblik over det interface som Unity 3 benytter sig af.

Som man kan se er der overordnet fem forskellige vinduer, heraf **Toolbar**, **Scene**, **Project**, **Hierarchy** og **Inspector**.

Ser vi på det første vindue som er **Toolbar**, så indeholder dette vindue de funktioner som skal bruges til at play/pause selve scenen man er ved at udvikle på, samtidigt med at den indeholder funktioner til at ændre på kameraet der viser scenen der afspilles.

Scene vinduet giver en grafisk repræsentation af dit spil. Her kan du bevæge dig frit rundt i dit spil, og har mulighed for ved hjælp af bl.a. musen at flytte på de objekter du har tilføjet. Samtidigt er det også i scenen at du kan se hvordan dit spil bliver afviklet når du trykker på play knappen i **Toolbar** vinduet.



Figur 5.1: Overblik over Unity 3.

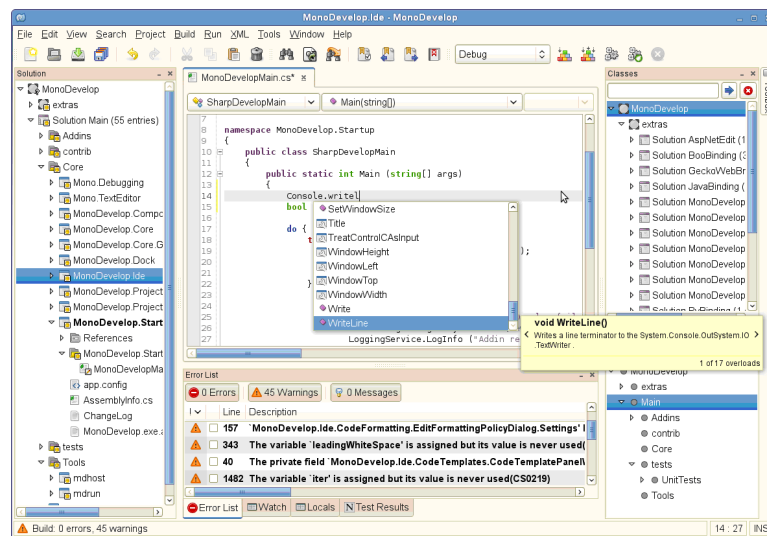
Hierarchy vinduet indeholder alle de objekter man gør brug af i sin scene. Det kan eksempelvis være en model af en bil som man tilføjer til scenen.

Project vinduet fungerer som et bibliotek over alle de ting man kan tilføje til sin scene. Skal man eksempelvis tilføje en model til sin scene, kan man blot trække denne model fra **Project** vinduet over til **Hierarchy** vinduet. **Project** vinduet indeholder også alle de scripts som man gør brug af til eksempelvis bevægelse af en model, og disse kan ændres i en editor blot ved at dobbeltklikke på filen.

Inspector vinduet giver en oversigt over forskellige attributter som kan ændres ved det objekt som man har markeret. Her kan man eksempelvis ændre på højden, bredden og længden af sit objekt, og har man tilføjet scripts til sit objekt er der også mulighed for at ændre på de definerbare attributter her.

5.1.2 Unity 3 scripting editor

Det er som skrevet muligt at udvikle scripts i Unity 3, hvilket gør det muligt eksempelvis at programmere hvordan modeller skal styres. Unity 3 gør brug af Mono, som er en open source implementation af .NET Frameworket. Det betyder samtidigt at Unity 3 giver mulighed for at skrive scripts i MonoDevelop, som er et IDE designet til bl.a. C# og andre .NET sprog. Det at der er mulighed for at bruge Mono, betyder at vi har mulighed for at benytte højniveausprog som eksempelvis C#, som vi igennem vores studie har god erfaring med at bruge. MonoDevelop-editoren giver ligeledes fordele som syntax highlighting, og autocomplete af funktionsnavne, som kan lette udviklingstiden markant.



Figur 5.2: MonoDevelop editor.

5.1.3 Disse ting er allerede implementeret i Unity Studios bilspilprojekt

Som tidligere nævnt under valg af udviklingsmiljø, så valgte vi at gøre brug af et bilspilprojekt fra Unity Studios, som giver os den fordel at der allerede er implementeret nogle grundlæggende ting som er nødvendige i vores bilsimulator. På listen nedenunder kan man se en oversigt over ting der allerede er implementeret.

- Bilfysik der kan simulere en bils bevægelse.
- Mulighed for at ændre på denne bilfysik.
- Styring af bil.
- Gearing af bil (dog uden kobling).
- Indeholder få modeller af biler, som vi kan gøre brug af.

Alle disse ting var implementeret sammen med en lille testbane, således at det faktisk allerede var muligt at køre med bilen ved brug af keyboard. Alt dette gav os noget grundlæggende at arbejde ud fra, hvilket gjorde at vi kunne koncentrere os om at udvikle de ting, som var mere specifikke for hvordan vores bilsimulator skulle fungere.

5.1.4 Disse ting skulle tilføjes og implementeres

Selvom vi havde mulighed for at gøre brug af et færdigudviklet projekt fra Unity Studios, var der stadigvæk mange ting som skulle ændres, før at det

KAPITEL 5. SIMULATOR

opfyldte de krav vi stillede i afsnit 3. På listen nedenunder kan man se en oversigt over de ting som skulle implementeres.

- Tilføje styring af rat og pedal-sæt.
- Muliggøre brug af kobling og 6 trins gearstang.
- Skabe en repræsentation af en by.
- Skabe en repræsentation af en landevej.
- Tilføje fodgængere.
- Tilføje selvstyrende biler til både bykørsel og landevej.
- Tilføje logning af kørsel.

Udviklingen af alle disse ting fra listen vil blive beskrevet nedenfor i vores implementeringsafsnit.

5.2 Implementering af rat og pedal-sæt, samt ændre bilfysik

Dette afsnit vil forklare implementeringen af rat og pedal-sættet, samt implementering af koblingen sammen med 6 trins gearing. Der vil også være en beskrivelse af hvordan bilfysikken blev tilpasset, således følelsen af at køre fungerede mere som en almindelig personbil.

Rat og pedal sæt

Til det Unity Studios projekt som vi brugte fulgte der allerede et `CarController.cs` script, som indeholdte alle de keyboardtaster som kunne bruges til at styre bilen med. Vi tilpassede dette script således at det havde en boolean attribut som hed `KeyboardControl`, denne skulle så kunne sættes til og fra, sådan at rat og pedal-sættet var aktivt når `KeyboardControl` var sat til `false`.

Hernæst lavede vi i `CarController.cs` nogle ændringer til de værdier som hed `steering`, `brake`, `clutch`, `throttle` og benyttede `drivetrain.setGear()` ved gearskift.

De forskellige inputs som var på rat og pedal-sættet blev tilgået ved hjælp af `Input.GetAxis()` funktionen. Vi lavede selv en sammenkobling mellem joysticket og Axisværdierne i Unity 3, da det kun var værdierne `Horizontal` og `Vertical` der virkede til at starte med.

For at finde ud af hvor meget der blev drejet på rattet gjorde vi som i listing 5.1.

Listing 5.1: *scripts/CarController.cs* - Find værdien for rattets drejning.

```
1 steering = Input.GetAxis("Horizontal");
```

5.2. IMPLEMENTERING AF RAT OG PEDAL-SÆT, SAMT ÆNDRE BILFYSIK

Da der var flere ting der skulle testes nu hvor vi skulle benytte et rat og pedal-sæt, manglede vi nogle gange muligheden for at se hvor meget henholdsvis kobling, speeder og bremsen blev trykket i bund. Derfor lavede vi et nyt script, der kunne vise værdierne for kobling, speeder og bremsen, men også hvilket gear den stod i, samt hvor meget rattet blev drejet til hver side. Dette script skulle hjælpe os med at implementere koblingen, sådan at den fungerede realistisk, men skulle også hjælpe til med at tilpasse bilfysikken til rat og pedal-sættet.

For at vise værdierne blev der lavet en boks med forskellige sliders der viste hver værdi ved hjælp af bl.a. at bruge funktionerne `GUI.Box`, `GUI.Label`, `GUI.HorizontalSlider` og `GUI.VerticalSlider`.

I listing 5.2 er et uddrag af koden som viser hvordan selve boksen bliver tegnet.

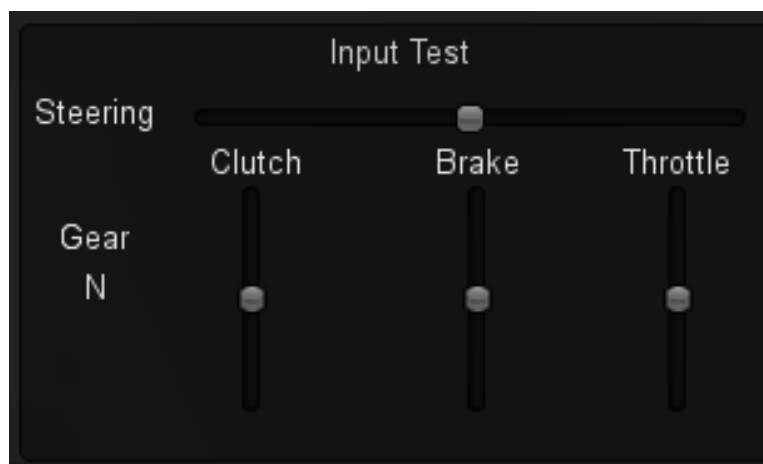
Listing 5.2: *scripts/InputTest.cs* - Tegn en box der viser værdier for rat og pedal-sættet.

```
1 GUI.BeginGroup(Rect (Screen.width - boxWidth, Screen.height -  
2 boxHeight, boxWidth, boxHeight));  
3 GUI.Box(Rect(0,0,boxWidth,boxHeight), "Input Test");  
4  
5 GUI.Label(Rect(5,25,100,30), "Steering");  
6 GUI.HorizontalSlider (Rect (70, 32, 220, 30), steeringValue, -1.0,  
7 1.0);  
8  
9 GUI.Label(Rect(25 + 50, 45, 100, 30), "Clutch");  
10 GUI.VerticalSlider (Rect (87, 65, 100, 90), clutchValue, -1.0,  
11 1.0);  
12  
13 GUI.Label(Rect(80 + 85, 45, 100, 30), "Brake");  
14 GUI.VerticalSlider (Rect (87 + 90, 65, 100, 90), brakeValue, 1.0,  
15 0.0);  
16  
17 GUI.Label(Rect(125 + 115, 45, 100, 30), "Throttle");  
18 GUI.VerticalSlider (Rect (87 + 170, 65, 100, 90), throttleValue,  
19 1.0, 0.0);  
20  
21 GUI.Label(Rect(15, 75, 100, 30), "Gear");  
22 GUI.Label(Rect(25, 95, 100, 30), gear);  
23  
24 GUI.EndGroup();
```

Som man kan se bliver boksen tegnet ved hjælp af funktioner fra GUI biblioteket. De forskellige sliders bliver sat ved hjælp af værdierne `steeringValue`, `clutchValue`, `brakeValue` og `throttleValue`. Derudover bliver der sat en label med en numerisk værdi ud fra værdien fra `gear` variabelen, der får et nummer alt efter hvilket gear som gearstangen står i. På figur 5.3 kan der ses et billede af hvordan `InputTest` boksen ser ud.

Implementering af kobling

Efter vi havde konfigureret `CarController.cs` klassen, således at den kunne tage imod input fra rat og pedal-sættet, kunne vi begynde at implementere



Figur 5.3: Model af InputTest boksen.

koblingen. Koblingen blev sat ud fra den værdi vi fik fra `Input.GetAxis("Clutch")` og denne værdi brugte vi til at finde ud af hvornår koblingen var i brug. Når koblingspedalen var i bund fik `clutch` attributten værdien -1, og når koblingspedalen var helt fri fik den værdien 1.

Koden der viser hvordan koblingen er implementeret er vist i listing 5.3.

Listing 5.3: *scripts/CarController.cs* - Implementering af kobling.

```

1  if(clutch < -0.1f && clutch > -0.4f)
2  {
3      drivetrain.maxTorque = 50 / -clutch;
4  }
5  else if (clutch < -0.4f)
6  {
7      drivetrain.maxTorque = 1;
8  }
9  else if (clutch > -0.1f)
10 {
11     drivetrain.maxTorque = 100;
12
13     // Stop engine if throttle is low and car is not moving < 6 km/
        t and gear is not neutral
14     if (drivetrain.gear != 1)
15     {
16         float speedKMt = rigidbody.velocity.magnitude * 3.6f;
17
18         if (throttle < 0.01 && speedKMt < 6)
19         {
20             SoundController.engineOn = false;
21         }
22     }
23 }
```

På linie 1, ser vi efter om koblingen ligger på koblingspunktet. Hvis dette er tilfældet sætter vi `drivetrain.maxTorque` til en værdi der bliver større dets mere koblingen har fat. Dvs. at når koblingsværdien går fra -0.4 imod -0.1 vil `drivetrain.maxTorque` blive større, og bilen vil få mulighed for at

køre.

På linie 5, kan man se at `drivetrain.maxTorque` bliver sat til 1, hvilket betyder at bilen ikke kan køre, hvis koblingsværdien er mindre end -0.4.

På linie 9, som er punktet hvor koblingen får helt fat, tjekker vi om bilen får nok gas ved hjælp af værdien `throttle` eller om bilen kører over 6 km/t. Hvis dette ikke er tilfældet vil bilen gå i stå fordi vi sætter `SoundController.engineOn = false;`. Når `SoundController.engineOn = false;` betyder det at motoren bliver slukket, da vi sørger for at funktionen `FixedUpdate()` i `Drivetrain` klassen ikke kan køre, samtidigt med at motorlyden bliver slukket i `SoundController` klassen.

Tilpasning af bilfysik

Projektet fra Unity Studios og deres tilhørende bil, er tilpasset således at den både lyder og reagerer som en racerbil. Da vi gerne vil prøve at simulere hvordan man under normale omstændigheder kører rundt i trafikken, vil vi derfor gerne have tilpasset bilen således den reagerer mere som en almindelig personbil.

Dette er muligt ved hjælp af nogle forskellige variableværdier der kan ændres på. Alle disse variable ligger gemt i klassen kaldet `Drivetrain`, som indeholder alle de funktioner der styrer selve bilfysikken.

Vi ændrede på `drivetrain.maxTorque`, som beskriver det maksimale drejningsmoment i nm (Newton meter), og `drivetrain.maxRPM` som beskriver det maksimale RPM (omdrejninger pr. minut).

Efter at rat og pedal-sættet var tilføjet til projektet, stødte vi på et problem som vi ikke havde oplevet før. Dette var at bilen pludselig ville blive ved med at accelererer hvis man gav gas og i samme øjeblik satte den i frigear. Grunden til at problemet ikke var opstået før, var at det ikke på samme måde var nemt at fremtvinge fejlen ved hjælp af et keyboard. Vi fandt en løsning til problemet, ved at tilføje et tjek i `drivetrain.FixedUpdate()` funktionen, som sørgede for at hjulene ville spinde ned, selvom at der blev givet gas når den var i frigear.

5.3 Trafik

Vores krav, se kapitel 3, dikterede, at der skulle være en mulighed for selvstyrrende biler i miljøet. Implementeringen af en kunstig intelligens som skulle styre biler autonomt, åbnede op for en afvejning af kompleksitet kontra behov. For at give miljøet liv og aktivitet, var der et fokus på at der skulle eksistere trafik i byen, der ville give testpersonen en følelse af aktivitet og udfordring, samtidig med at trafikken skulle kunne køre autonomt og uden at skabe problemer. Især uforudsigeligheden af testpersonens handlinger udgjorde en udfordring for implementationsmetoden.

KAPITEL 5. SIMULATOR

Medmindre andet er angivet, er implementationen i dette afsnit baseret på gruppens egne løsningsmetoder.



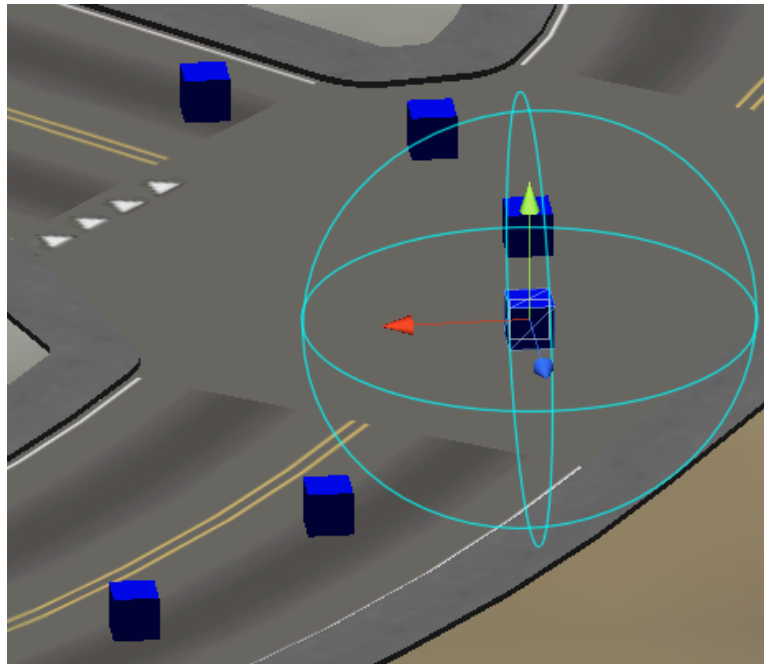
Figur 5.4: Tre selvkørende biler i et lyskryds.

Helt essentielt for opgaven, var behovet for en algoritme der kunne finde vej gennem en rute, en såkaldt *pathfinding*-algoritme. Der blev lagt nogle timer i at undersøge mulighederne for at benytte offentlige tilgængelige ressourcer for pathfinding i bil-simulatorer som vi kunne bruge, men ultimativt var den nemmeste løsning at lave vores egen fra bunden, da det ville tage længere tid at tilpasse noget eksisterende til vores miljø, end det ville at lave det selv. Især integrationen med den eksisterende bilfysik, se afsnit 5.1, ville have været et problem, men da denne alligevel skulle tilpasses en del for at fungere sammen med det eksterne rat, pedaler og gear, se afsnit 3, lå det os nært samtidig at skrive nogle metoder som den kunstige intelligens kunne benytte sig af. Metoder der gav adgang til bilens acceleration, bremses og styring blev implementeret i bilens `carController`. Disse metoder var henholdsvis `setThrottle`, `setBrake` og `setSteering`, hvor `setThrottle` var en metode der modtog en floating-point værdi mellem 0 og 1, hvor 0 var ækvivalent med at gaspedalen var uberørt, og 1 betød at den var trykket i bund. Det samme gjorde sig gældende for `setBrake`. Slutteligt var `setSteering` en metode der modtog en floating-point værdi mellem -1 og 1, hvor -1 betød at hjulene var drejet maksimalt mod venstre, 1 betød maksimalt til højre, og når værdien var 0 pegede hjulene ligeud. Disse metoder manipulerede med den kode, som den åbne bilfysik havde leveret. Det krævede nogle timers nærstudie af bilfysikkens kode før at vi havde mulighed for at manipulere tilstrækkeligt med den.

5.3.1 Waypoints

Det blev besluttet, at den bedste løsning på pathfinding, der lå inden for en passende tidshorisont, var et paradigme om at trafikken blev givet en række rutepunkter, *waypoints*, den skulle navigere efter. Det var herefter op til de enkelte waypoints at informere biler som kørte efter det, om hvilken

hastighedsgrænse der var i det pågældende område. Altså ville et waypoint der lå i et skarpt sving eksempelvis have en lavere hastighedsgrænse end et der lå på en lige strækning. Yderligere blev et waypoint givet en radius, der bestemte hvor præcist et område bilen skulle ramme, før den blev sendt videre til det næste waypoint.



Figur 5.5: En serie af waypoints i et sving, med radiussen af et waypoint synligt.

Da der var flere forskellige biler til rådighed, og bilfysikken reagerede forskelligt alt efter om det eksempelvis var en tung bil eller en let bil, var det ikke til at forudsige en helt præcis bane en bil ville befinde sig i, eksempelvis et sving. Af grunde som denne var det nødvendigt at give waypoints en større radius i skarpe sving end på blødere kurver eller lige vejbaner, da tunge biler ville dreje i en anden bane end lette biler. I Unity var waypoints en `Transform`, der er et universalt komponent som bestemmer rotationen, størrelsen og positionen på et objekt. For overskuelighedens skyld blev de, som det fremgår af figur 5.5, visualiseret som blå kasser, men i den endelige version var de usynlige.

5.3.2 Hastighed

Koden til bilfysikken var åben, og denne gav os adgang til at aflæse bilens hastighed på et hvert givent tidspunkt i afviklingen af programmet. Denne variabel, sammen med det pågældende waypoints hastighedsgrænse, gjorde det muligt at udvikle nogle funktioner for hastighedsforøgelse og -sækelse.

KAPITEL 5. SIMULATOR

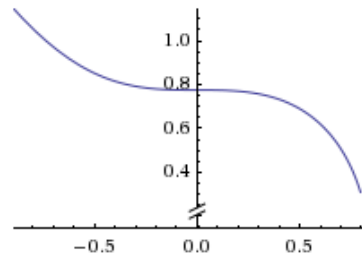
Eksempelvis skulle en bil bremse, hvis den i øjeblikket kørte 60 km/t, men kørte efter et waypoint der dikterede 40 km/t og accelerere hvis det omvendte var tilfældet. Men det var ikke en holdbar løsning at give `setThrottle` eller `setBrake` en værdi af 1 indtil hastigheden nåede grænsen, og derefter give den 0. For at tage udgangspunkt i accelerationen, er det ikke realistisk at trykke speederen i bund hvis man kører 40 km/t for at nå 45 km/t, men i stedet forøger man typisk hastigheden gradvist og i et mere kontrolleret tempo. Yderligere var flere af bilerne i simulatoren baghjulstrukne, hvilket betød at fuld acceleration ville bevirke at bilerne skred ud, selv ved lavere hastigheder. Omvendt var det heller ikke holdbart at den computerstyrede trafik ville blokere bremserne hver gang hastigheden kom det mindste over fartgrænsen, men ligeledes ville en nedbremsning kræve en kontrolleret effekt, og spørgsmål som bremselængde og andre biler i nærheden havde også en indflydelse på opbremsningen. Se sektion 5.3.4.

Listing 5.4: *scripts/aiSteer/SteerForTarget.cs* - `adjustThrottle` i `SteerForTarget`.

```
1 void adjustThrottle(float currentSpeed, Waypoint wp) {
2     if (brake) {
3         carController.setBrake(1.0f);
4         carController.setThrottle(0);
5     }
6     else {
7         carController.setBrake(0);
8
9         if ((currentSpeed / wp.speedLimit) > 1.0f) {
10            float distance = Vector3.Distance(transform.position, wp.
11                transform.position);
12            // Easing function 1 / Max of distance or 0.1f
13            carController.setBrake(1 / (Mathf.Max(distance, 0.1f)));
14        }
15        else {
16            // Easing function sqrt(1-x^3) - 0.6f
17            float throttle = (float)Math.Sqrt(Math.Max(0.6f - Math.Pow
18                ((currentSpeed / wp.speedLimit), 3), 0));
19            carController.setThrottle(throttle);
20        }
21    }
22 }
```

Listing 5.4 viser kildekoden der styrer bilens hastighed. Som løsning på de ovenstående udfordringer blev der udviklet nogle tilpasningsfunktioner, eller *easing functions*, der ud fra nogle variabler returnerede en værdi (typisk) mellem 0 og 1 som `setBrake` og `setThrottle` kunne forstå. Til at bremse, var funktionen $1/\max(\text{distance}, 0.1)$, altså 1 over afstanden mellem bilen og det waypoint det kørte imod. `Max()`-funktionen blev tilføjet for at undgå division med 0. Formålet med denne var, at sørge for en bil ville bremse mest effektivt op, alt efter hvor lang afstand der var til det næste waypoint. Hvis afstanden til næste waypoint var meget kort, ville opbremsningen være stærkere end hvis afstanden var lang.

Hastigheden blev styret med easing-funktionen $\sqrt[2]{0.6 - x^3}$ hvor x = bilens nuværende hastighed divideret med fartgrænsen, se figur 5.6.



Figur 5.6: $\sqrt[3]{0.6 - x^3}$.

Formålet med funktionen var at give bilen en kontinuerlig acceleration på $\sqrt[3]{0.6}$ indtil hastigheden var næsten ved fartgrænsen. En funktion der lod accelerationen være direkte proportional med tilnærmelsen af hastighedsgrænsen gav i forsøget en alt for langsom og urealistisk acceleration, hvor det i virkeligheden er normalt med et konstant pres på gaspedalen indtil man er tæt på den ønskede hastighed. Da det er sjældent man accelererer med speederen helt i bund, og givet problemerne med de føromtalte baghjulstrukne biler, blev den maksimale acceleration sat ned til $\sqrt[3]{0.6 - 0^3} = 0.77$ og ikke 1.

5.3.3 Retning

På samme måde som hastigheden, blev bilens styring kontrolleret af en easing funktion. Hvis en bils retning ligger langt fra det målsatte, er det normalt at dreje skarpere, end hvis bilens retning og målets retning er tæt på hinanden. Ligeledes er det normalt at svinge blødere ved højere hastigheder og skarpere ved lave hastigheder, se listing 5.5.

Listing 5.5: *scripts/aiSteer/SteerForTarget.cs* - `adjustSteering` i `SteerForTarget`.

```

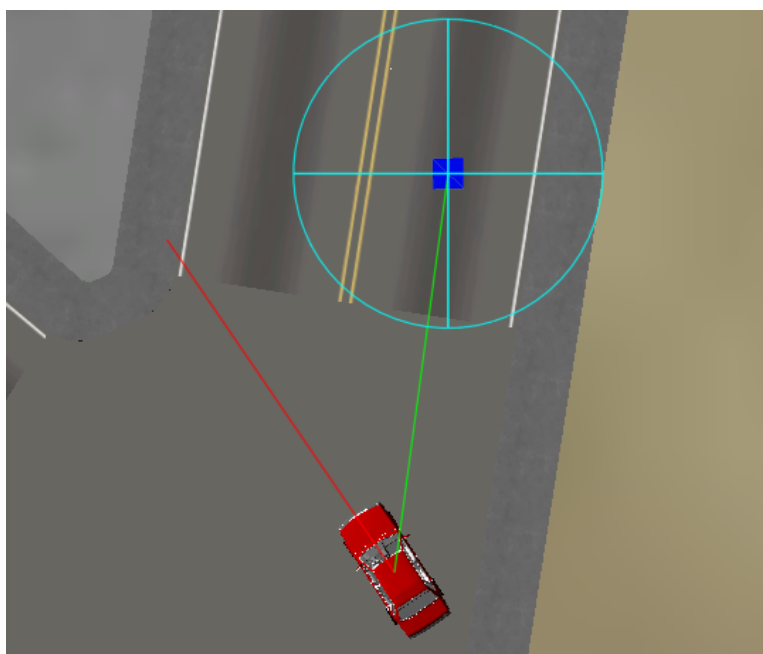
1 void adjustSteering(float currentSpeed, Waypoint wp) {
2     /*
3     angle > 0 && angle < 180 = Right
4     angle < 0 && angle > -180 = Left
5     */
6     Vector3 targetDirection = wp.transform.position - transform.
        position;
7
8     float angleCar = Mathf.Atan2(transform.forward.x, transform.
        forward.z) * Mathf.Rad2Deg;
9     float angleTarget = Mathf.Atan2(targetDirection.x, targetDirection
        .z) * Mathf.Rad2Deg;
10    float angle = Mathf.DeltaAngle(angleCar, angleTarget);
11
12    // The faster it goes, the smoother it should turn - 150 km/h =
        max speed
13    float steerSpeedAdjust = Math.Max(1.0f - (currentSpeed / 150), 0.1
        f);
14    // Easing function sin(x/2)*3
15    float steerAdjust = Mathf.Sin((angle * Mathf.Deg2Rad)/2)*3*
        steerSpeedAdjust;

```

KAPITEL 5. SIMULATOR

```
16 |  
17 |     carController.setSteering(steerAdjust);  
18 | }
```

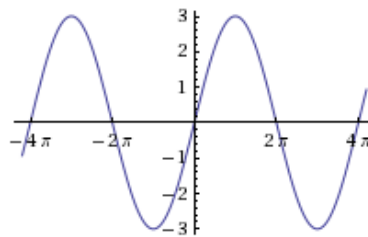
For at udregne hvilken retning bilen skulle køre, blev der gjort brug af nogle af Unitys indbyggede matematikfunktioner. *Mathf.Atan2* er en funktion der returnerer en vinkel i radianer hvor tangens er y/x . I henhold til bilen, er x og y er henholdsvis x- og z-koordinatet på **transform.forward**, en "flad" (3D-koordinat uden nogen y-værdi) retningsvektor der peger fremad fra bilen. Den samme procedure gentages for x- og z-koordinatet på retningsvektoren for det pågældende waypoint. Vinklen mellem retningen fra bilens forparti og retningen mod waypointet udregnes så fra *Mathf.DeltaAngle* der returnerer en pæn værdi mellem -180 og 180 i stedet for 0-360, se figur 5.7.



Figur 5.7: Visualisering af udregningen af vinklen mellem bilens forparti og waypointets position.

Easing-funktionen benyttet til at styre intensiteten af svinget tog udgangspunkt i en sinuskurve til x divideret med 2, ($\sin(x/2)$), hvor x = vinklen i radianer og blev støttet af en justerings-funktion, se figur 5.8. Altså bevirkede funktionen at bilen drejede skarpere jo længere fra målet den befandt sig, med lidt efterfølgende justeringer. Ligesom med hastighedsfunktionen, var en ren proportional svingning for langsom i forhold til både funktionaliteten og realismen i en bils drejning. Selvom **steerAdjust** ikke kunne dreje skarpere end 1, blev sinusfunktionen ganget med 3 for at give den et maksimalt udsving mellem -3 og 3, for derved at lade svinget

være maksimalt i en længere periode til trods for at vinklen nærmede sig 0. For yderligere at styre intensiteten i svinget i forhold til bilens hastighed, blev funktionen ganget med en `steerSpeedAdjust`-værdi, for herved at lade bilen dreje langsommere ved højere hastigheder.



Figur 5.8: $\sin(x/2) * 3$.

Hastighedsfunktionen og styrefunktionen udgjorde logikken i `steerForTarget`-objektet. Dette blev nedarvet i `steerForPath` som forstod at tolke en række af waypoints og derved give bilen disse i den rigtige rækkefølge, så bilen foregav at følge en vej på en naturlig måde. Yderligere funktionalitet inkluderede muligheden for at bilen selv kunne finde det waypoint den var tættest på og hvilken retning bilen drejede, for herved at effektivisere den handling at indsætte flere biler på den samme rute, forskellige steder i byen. Endvidere blev de selvkørende bilers bremse- og blinklys styret automatisk, da dette blev anset for at være vigtigt for testpersonens opfattelse og reaktioner i den virtuelle verden.

5.3.4 Trafiksikkerhed og sidste udvej

Selvom bilernes evne til at overholde fartgrænsen og navigere efter mål var opfyldt, var det stadig nødvendigt at sikre, at bilerne også kunne undgå at køre forkert eller kolliderer med enten andre selvkørende biler eller testpersonen. Yderligere skulle bilerne overholde gældende danske regler for vigepligt og lyskryds som begge var tilstede i byen. For at opnå en acceptabel løsning blev bilerne udstyret med en `SphereCast`-funktion, der er en Unity-metode til at opfange om et givent objekt kolliderer med en `collider`. Som navnet antyder, er en `collider` et objekt i Unity som der kan kolliderer med andre objekter, og i tilfældet returnere en række værdier på baggrund af kollisionen.

Listing 5.6: `scripts/aiSteer/DetectCollisions.cs` - Udpluk af kollisionsdetektionen i `DetectCollisions.cs`.

```

1 float stoppingDistance() {
2     float distance = 0f;
3     float v0 = rigidbody.velocity.magnitude;
4     float mu = 1.0f;
5     float tr = 1.2f; // A little earlier than perscribed safety
      distance

```

KAPITEL 5. SIMULATOR

```
6     float g = -Physics.gravity.y;
7
8     distance = tr * (Mathf.Pow(v0, 2) / 2*mu*g);
9
10    // Below 3.0 is inside the car
11    return Mathf.Max(distance/100, 3.0f);
12 }
13
14 if(Physics.SphereCast(ray, detectionRadius, out hitInfo,
15    detectionDistance, ((1 << carLayer) | (1 << trafficLightLayer) |
16    (1 << giveWayLayer))) {
17
18    int layer = hitInfo.transform.gameObject.layer;
19    // Hit a car
20    if (layer == carLayer) {
21        sft.brake = true;
22    }
23    (...)
24    // We're out of the collision
25    else {
26        sft.brake = false;
27    }
28 }
```

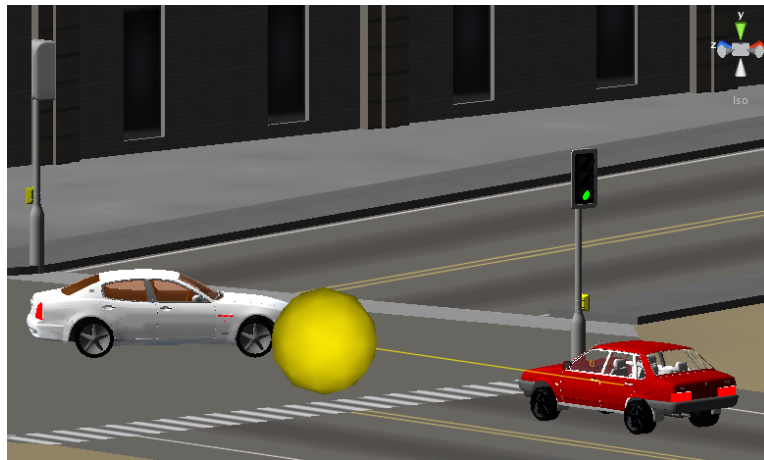
Listing 5.6 viser et udpluk af funktionen, der opfanger om bilens forparti er på vej til et frontalsammenstød. *SphereCast* skal opfattes som en usynlig ”kugle” der skydes ud fra forpartiet af bilen, i en afstand bestemt af **stoppingDistance**, der er den almindelige formel¹ for at beregne bremselængde baseret på en række variabler som hastighed og vejgreb, se figur 5.9. Denne funktionalitet gør sig nyttig i flere sammenhænge, da det herved er muligt i Unity at få informationer om præcis hvad det er, som bilen er på vej til at køre ind i. I ovenstående eksempel bliver der på linje 18 undersøgt, om det der opfanges kommer fra laget **carLayer**. I forbindelse med kollisioner tilbyder Unity at tildele objekter forskellige lag, hvorved alle biler i miljøet er tildelt et fælles lag. På den måde er der mulighed for at diskriminere mellem om bilen eksempelvis er på kollisionskurs med et lyskryds eller en fremmed bil, og herefter handle derefter. Ovenstående eksempel beskytter bilen mod at kolliderer med andre biler, ved at aktivere bremserne når der opdages en anden bil. Dette giver automatisk også en sikkerhedsafstand på vejene, hvis to eller flere biler kører efter hinanden.

Lyskryds

Udover at afholde biler fra at kolliderer med hinanden, sørger kollisionsfunktionen også for at kontrollere bilers adfærd når et lyskryds opdages forude. Et lyskryds er en serie af fire **colliders** der er placeret på alle fire veje der leder ind i et lyskryds, se figur 5.4. På samme måde som foreskellige biler er tilknyttet et bestemt lag i Unity, er lyskryds-**colliders** tilknyttet deres eget lag. Ved hjælp af et Singleton² design pattern, sørger en kontrollerende

¹<http://www.wolframalpha.com/input/?i=stopping+distance>

²http://en.wikipedia.org/wiki/Singleton_pattern



Figur 5.9: Visualisering af et SphereCast. I realiteten strækker kuglen sig dog i sin fulde størrelse langs den gule linje, og ikke kun ude for enden.

funktion for at skifte farve i alle lyskryds samtidig i et bestemt interval.

Listing 5.7: *scripts/aiSteer/DetectCollisions.cs* - Trafiklys-delen af *DetectCollisions.cs*.

```

1  else if (layer == trafficLightLayer) {
2      trafficLight = hitInfo.transform.gameObject.GetComponent<
        TrafficLight>();
3      facing = Vector3.Dot(trafficLight.transform.forward, transform.
        forward);
4
5      if (facing < 0) {
6          // Red light
7          if (!trafficLight.hasGreenLight) {
8              sft.brake = true;
9          }
10         // Green light
11         else {
12             sft.brake = false;
13         }
14     }
15 }

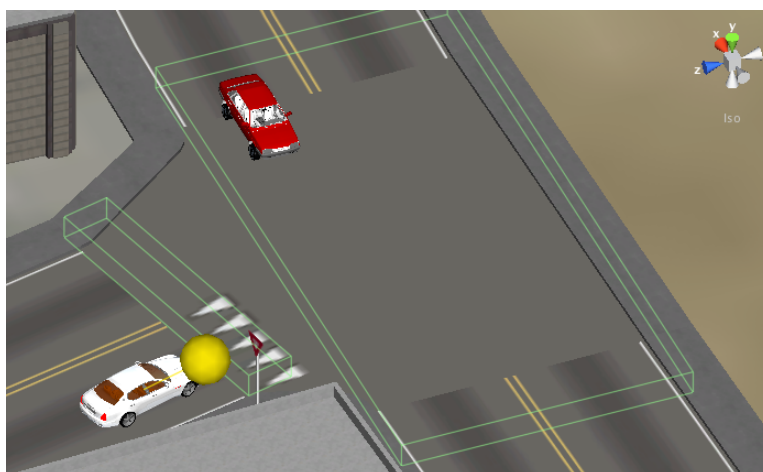
```

Funktionalitetsmæssigt tilbyder Unity en funktion der kan ignorere kollisioner mellem to lag, da Unitys indbyggede fysikregler ellers ikke tillader at en **collider** kan gå igennem en anden. De selvkørende biler har derfor mulighed for at opdage når de nærmer sig en **collider**, men ikke blive standset af dem, når de skal køre over for grønt. På samme måde som med funktionaliteten der stopper en bil fra at køre ind i anden, vil en bil der opdager et rødt lys forude begynde at bremse indtil bilen holder stille foran lyskrydset, eller indtil der bliver grønt, se listing 5.7. Når to biler der kører ud i et kryds samtidig, hvor den ene skal svinge og den anden kører ligeud, vil den svingende bil stoppe og afvente fri bane så snart den opdager en anden bil i sin svingbane, som beskrevet i begyndelsen af afsnittet. Ved at tage prikproduktet af bilens og trafiklysets fremadrettede vektorer, er det

KAPITEL 5. SIMULATOR

muligt at undersøge om bilen er på vej ind eller ud af et lyskryds. På den måde vil en bil der eksempelvis svinger til højre i et lyskryds, ikke stoppe når den støder på den højre vejbanes lyskryds, der i det tilfælde vil lyse rødt.

Vigepligt



Figur 5.10: Visualisering af et T-kryds med vigepligt.

Med T-kryds er der potentiel risiko for sammenstød, hvis en tværgående bil kører forbi i høj fart, mens en anden bil er på vej ud på den tværgående vej, se figur 5.10. Vigepligt implementeres funktionelt på næsten samme måde som lyskryds, hvor den vej der leder ind på den tværgående vej har en *collider* tilknyttet, som teknisk vil ”lyse rødt” hvis en modkørende bil befinder sig i nærheden på den tværgående vej, se listing 5.8.

Listing 5.8: *scripts/aiSteer/DetectCollisions.cs* - Vigepligt-delen af *DetectCollisions.cs*.

```
1 if (layer == giveWayLayer) {
2     giveWayEntry = hitInfo.transform.gameObject.GetComponent<
        GiveWayEntry>();
3
4     facing = Vector3.Dot(giveWayEntry.transform.forward, transform.
        forward);
5
6     if (facing < 0) {
7         if (giveWayEntry.giveWayField.isOccupied) {
8             sft.brake = true;
9         }
10        else {
11            sft.brake = false;
12        }
13    }
14 }
```


Sidste udvej

Skulle uforudsete konsekvenser alligevel sende en bil på afveje, implementeredes en funktion til at rette op på eventuelle fejl der ville forstyrre eller ødelægge eksperimentet. For at undgå selvkørende biler endte et sted hvor de ikke kunne komme nogen steder, eller på anden måde pludselig mistede evnen til at fortsætte sin programmerede rute, blev bilens position undersøgt hver 2. sekund. Ud fra en række variabler kunne det afsløres, om en bil der burde bevæge sig ikke gjorde det, og hvis dette var tilfældet, ville programmet simplethen tvinge bilen hen til det nærmeste waypoint der befandt sig på bilens rute. Skulle en anden bil uheldigvis befinde sig på det sted, ville den blive rykket til den næst-nærmeste, og så videre, indtil en ledig plads blev fundet.

5.4 Implementation af fodgænger

Et af de dynamiske objekter som vi også havde tænkt os at have i vores bilsimulator var fodgængere. Da vi ikke havde tænkt os at bruge for meget tid på at tilføje disse fodgængere fandt vi igen et projekt fra Unity Studios, hvor der på forhånd var en model med tilhørende animationer for bevægelse, hvoriblandt der bl.a. var en animation til at få modellen til at gå.



Figur 5.11: Model af fodgænger.

På figur 5.11 kan man se hvordan den model vi har valgt ser ud. Denne model er lavet med et lavt antal polygoner, hvilket betyder at vi har mulighed for at tilføje flere modeller på samme tid, uden at det sætter for store krav til afviklingen af simulatoren.

For at få fodgængereren til at bevæge sig lavede vi et script kaldet `Walk.cs`, som indeholder to vigtige funktioner, en `moveForward()` funktion og en `TurnAround()` funktion.

KAPITEL 5. SIMULATOR

Koden for `moveForward()` funktionen kan ses i listing 5.9

Listing 5.9: *scripts/CharacterControl/Walk.cs* - Få modellen til at bevæge sig.

```
1 void moveForward ()
2 {
3     charDirection.x = 0;
4     charDirection.y = 0;
5     charDirection.z = walkSpeed;
6
7     charDirection = transform.TransformDirection(charDirection);
8     charDirection.y -= gravity * Time.deltaTime;
9
10    characterController.Move(charDirection * (Time.deltaTime *
11        walkSpeed));
12    animation.CrossFade("walk");
13 }
```

Efter linie 3 sætter vi den vektor som skal bruges til at beskrive den retning som vores model skal bevæge sig efter. Z-koordinaten bliver sat til `walkSpeed` som er hastigheden på modellen, herefter bliver `transform.TransformDirection()` kaldt på vektoren, hvilket transformere den lokale vektor position til en global vektorposition i forhold til hele det virtuelle miljø. Hernæst bliver y-koordinaten sat ud fra en `gravity` værdi, for at bestemme hældningen på vektoren hvis modellen bevæger sig op langs en bakke. På linie 10-11 bliver modellen sat i bevægelse ved hjælp af `characterController.Move()` funktionen som får retningsvektoren som input, hernæst sættes animationen der skal bruges ved hjælp af `animation.CrossFade("walk")`.

I `Walk.cs` klassen bliver der sat to globale positioner, som er endepunkter modellen bevæger sig imod. Hver gang `turnAround()` funktionen bliver kaldt, vil modellen vende sig imod den modsatte globale position, og derfor vil fodgængereren blive ved med at gå imellem de 2 positioner.

Både `moveForward()` og `TurnAround()` funktionerne bliver kaldt inde i `Update()` funktionen der ligger i `Walk.cs` klassen. Denne `Update()` funktion vil blive kaldt ved hver billede frame, og sørger bl.a. for at kalde `TurnAround()` hver gang modellen er kommet hen til den ene af de globale positioner.

5.5 Logning

En af kravene til vores simulator var at vi ville kunne automatisere indsamling af data i løbet af eventuelle eksperimenter. Denne data skal være i et format som ikke gør det besværligt at lave analyse på det. Der skal være mulighed for at udvide logning hvis der er nogle nye værdier man er interesseret i. Vi havde tre valgmuligheder indenfor hvilket format nemlig CSV(Comma-Separated Values), XML(Extensible Markup Language) og JSON(JavaScript Object Notation).

CSV er et meget simpelt format hvor man laver en liste af værdier separeret af komma-tegn. Man laver så en ny linje for hver måling med værdierne. Man skal holde styr på hvilke værdier der er på hvilke pladser på hver linje.

XML er et struktureret format som kan bruges til en lang række opgaver. Der er indbygget værktøjer i .NET til at læse og skrive XML. Med XML vil man kunne opbygge et træ af målinger hvor hver måling indeholder et element for hver værdi.

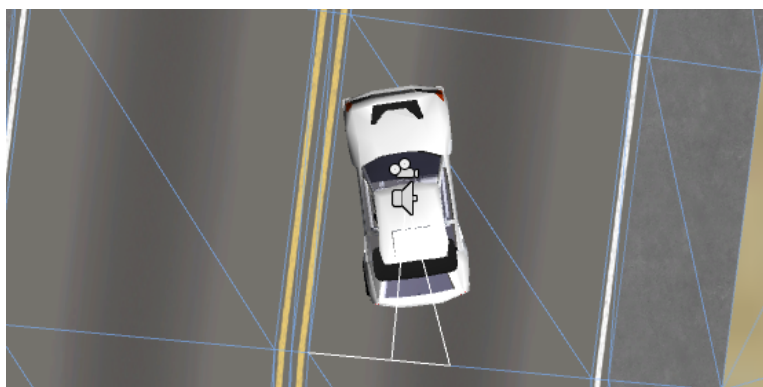
JSON er et format som har den samme syntaks som man bruger når man programmerer i JavaScript. Der findes udviklingsbiblioteker til en lang række programmeringssprog der gør det nemt at både læse og skrive formatet heriblandt C#. ³ Med JSON kan man lave et objekt for hver måling som indeholder par af navne og værdier.

Vi har valgt at bruge JSON da den bedst tilfredsstiller vores krav om at det ikke er besværligt at lave analyse på det, og det er simpelt at udvide logningen. Vi fravalgte CSV fordi man skal holde styr på værdiernes position og det gør at der nemmere kan opstå menneskeskabte fejl når man skal udvide logningen og et eventuelt tilhørende analyseredskab. En af grundene til at XML blev fravalgt er at det kan være besværligt at skrive det løbende til en fil imens målingerne sker og stadig lave korrekt XML.

De ting vi har valgt at tage med i logfilen i første omgang er tid, fart, afstand til midten af vejbanen, værdierne fra rat og pedaler, position i det virtuelle miljø, condition og hvilket segment af vejen bilen befinder sig. De fleste af disse værdier kan aflæses direkte uden nogen beregning. Når man skifter i mellem conditions sættes værdien i loggeren så den kan skrive ved hver måling. Vejen kan inddeles i en række segmenter, og loggeren gemmer en værdi som bliver skrevet sammen med hver måling. Når bilen kører fra et segment til et andet kører den over en **Trigger** som har navnet på det næste segment. Denne værdi bliver gemt i loggeren. Den mest besværlige værdi at beregne er afstanden til midten af vejbanen.

På figur 5.12 ses linjerne der bliver brugt til at beregne afstanden. Modeller som bruges i Unity er inddelt i trekanter og/eller firkanter, og vi finder den trekant der er i vejen under midten af bilen. Vi finder så den kant af trekanten der er 5 enheder lang fordi at vi ved at vejen er 5 enheder bred. Herefter beregner vi ved hjælp af linjens ligning linjen der går fra midten af bilen og vinkelret ind på kanten som vi lige har fundet. Så finder vi skæringen mellem den nye linje og kanten. Herefter finder vi afstanden fra skæringen til midten af kanten. Fordi vi også mente at der måske kunne være situationer hvor det kunne være interessant at vide om det var i højre side eller venstre side af vejen der blev overskredet ville vi gerne have at hvis man var

³<http://json.org/>



Figur 5.12: Beregning af afstand til midten af vejbanen.

til venstre for midten ville tallet være negativt og hvis man er til højre for midten ville tallet være positivt. Trekanterne er i et array på modellen og vi fandt ud af at der var en sammenhæng mellem deres position i arrayet og om det var den højre eller den venstre trekant nemlig at hvis trekantens position i arrayet er et lige tal så er trekanten til venstre og hvis det er et ulige tal er trekanten til højre.

5.6 Tidsforbrug og linjer kode

I vores oprindelige krav var det defineret, at udviklingen af simulatoren skulle være for begrænsede omkostninger. Ydermere pointerede kravene til selve simulatoren i kapitel 3 også, at simulatoren blandt andet skulle være billig og hurtig at udvikle i, har relevans at belyse tidsforbruget og mængden af ændringer lavet på simulatoren.

Af tilføjelser til simulatoren udgjorde vores egen implementation omkring 1400 nye linjer kode. Det skal dog tages med i betragtningerne, at Unity er mere end ren kode, da modellering af verden samt opsætning af variable værdier gøres i selve programmet og ikke eksisterer som tekstbaseret kode hvor linjerne kan tælles. Altså skal der påregnes noget ekstra tidsforbrug udover de 1400 tilføjede linjer. Fra udviklingen i Unity startede til koden blev frosset var der gået næsten 2 måneder, men i denne periode skal også medregnes tidsforbrug til de andre aspekter af projektet som ikke nødvendigvis var direkte udvikling af software-delen af simulatoren.

6

Eksperiment

Ud fra den viden vi opnåede i litteraturstudiet i kapitel 2, de krav der blev defineret i kapitel 3 samt erfaringerne gjort under implementationen af simulatoren i kapitel 5, kan vi nu opstille et eksperiment som har til formål at undersøge vores hypoteser.

6.1 Introduktion

Givet hvad vi lærte i litteraturstudiet, har eksperimentet til formål at undersøge, hvorvidt en testpersons fornemmelse for tilstedeværelse og fart kan manipuleres ved introduktion af statiske og dynamiske objekter til miljøet. Vi vil på baggrund af tidligere forsøg, koncentrere os om testpersoners reaktion på forskellige conditions. Samtidig udsættes de for at udføre en række opgaver imens de kører, for her at undersøge deres følelse af tilstedeværelse i simulatoren. Mere herom i afsnit 6.2. Successkriterierne for udførelsen af eksperimentet vil omhandle testpersonens evne til at overholde fartgrænsen samt deres evne til at placere bilen i midten af kørebanen, mere herom i afsnit 6.6.

Hypoteser

På baggrund af ovenstående, kan vi hermed udlede vores hypoteser, som dette eksperiment vil undersøge:

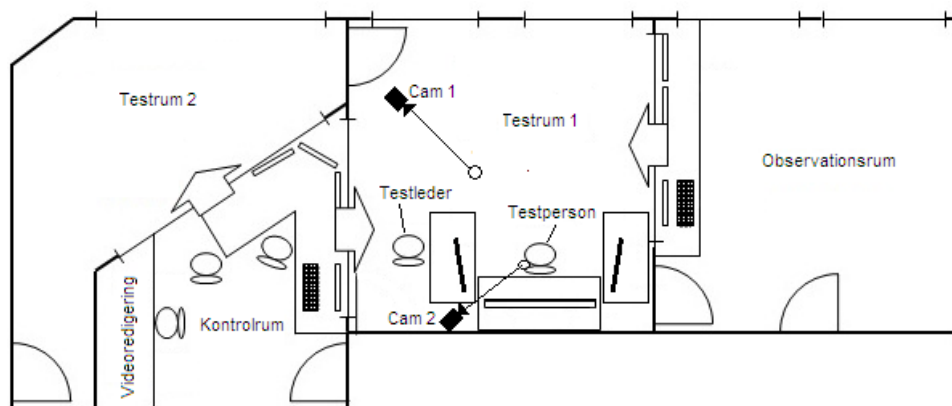
- Ved introduktion af statiske objekter i miljøet (bygninger), vil testpersonen være bedre til at overholde fartgrænsen, end uden tilstedeværelsen af statiske objekter i samme miljø.
- Ved introduktion af statiske objekter i miljøet (bygninger), vil testpersonen være bedre til at holde placeringen på vejen, end uden tilstedeværelsen af statiske objekter i samme miljø.
- Ved introduktion af dynamiske objekter i miljøet (trafik og fodgængere), vil testpersonen være bedre til at overholde fartgrænsen, end uden tilstedeværelsen af statiske objekter i samme miljø.

KAPITEL 6. EKSPERIMENT

- Ved introduktion af dynamiske objekter i miljøet (trafik og fodgængere), vil testpersonen være bedre til at holde placeringen på vejen, end uden tilstedeværelsen af statiske objekter i samme miljø.

6.2 Opsætning

Vores opsætning af eksperimentet er baseret på de krav der på forhånd er stillet i afsnit 3. Vi valgte at opstille simulatoren i HCI-laboratoriet på Aalborg Universitet Cassiopeia, og selve opsætningen af laboratoriet kan ses på figur 6.1.



Figur 6.1: Overblik over laboratoriet.

Vi valgte at gøre brug af testrum 1, samt kontrolrummet. I testrum 1 havde vi vores simulator opstillet som vist på figuren, hvor vores testleder var placeret således at han havde et godt overblik over selve testen og testpersonen.

Vi gjorde brug af to kameraer, hvor det ene var sat således at det gav et billede af opgaveløsningen på iPod Touch, og det andet kamera var fokuseret direkte imod ansigtet af testpersonen for at opfange øjefokuseringer. Fra hovedskærmen havde vi et output kabel der gik til kontrolrummet, sådan at vi kunne optage det, der skete på selve simulatorskærbilledet.

I kontrolrummet var vi tre personer, hvoraf en person havde ansvar for at styre optagelsen af testen og de sidste to personer overvågede testudførelsen.

Selve konstruktionen af simulatoren var lavet ved hjælp af EURO paller, hvor et bilsæde var boltet fast til den øverste palle, således at justering af bilsædet var fuldt funktionelt. På figur 6.2 er vist et billede af selve opstillingen, hvor man ser hele konstruktionen. Der var placeret en stor fladskærm forud for bilsædet, som skulle fungere som forruden. Vi brugte denne løsning, da vi havde svært ved at få billedet fra en projektor placeret tæt nok på testpersonen, uden at testpersonen selv ville stå i vejen for projekteringen. Derudover



Figur 6.2: Billede af simulator.

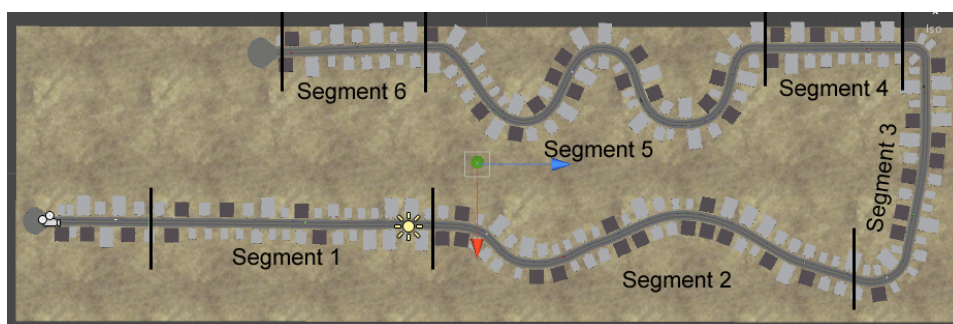
gav brug af fladskærmen et mere tydeligt og klarere billede, end de projektorer vi havde til rådighed fra AAU. Der var også opstillet to fladskærme der skulle agere sideruder, hvilket ligeledes var et krav fra tidligere, som skulle give et bredere synsfelt til testpersonerne. Rattet var spændt fast til bordet foran testpersonen, og pedalerne var placeret langs væggen nedenunder bordet. Gearstangen var placeret således, at den stod til højre for bilsædet, i en højde der svarer til hvad der forventes i en almindelig personbil.

6.3 Deltagere

Til udførelse af eksperimentet, forsøgte vi at gøre brug af en så objektiv gruppe som muligt, som beskrevet i kapitel 14 i [13]. Det blev besluttet at vi ville finde 16 testpersoner, 8 drenge og 8 piger, da der er en mulighed for at interessante resultater forbigås ved grupper på mindre end 15-20 [13]. 8 drenge og 2 piger som gik på studiet blev fundet, og med de resterende 6 forsøgte vi at finde individer der så vidt som muligt passede på de andre testpersoners data, altså universitetsstuderende i samme aldersgruppe. Resultatet blev 16 testpersoner, alle med kørekort, i alderen 20-25 år, fra naturvidenskabelige, humanistiske eller samfundsvidenskabelige studieretninger.

6.4 Software

Efter vi havde bygget byen som skulle bruges som simulatorens testmiljø, gjorde vi os den erkendelse, at brugen af en by ville producere for mange variabler som der blev ændret på. Vi ville ikke kunne garantere at alle testpersoner holdt den samme fart i de samme situationer, og generelt ville brugen af en by ikke introducere de samme forhold for alle. Ved i stedet at bygge en lige landevej, hvor alle testpersoner skulle køre den samme vej med samme hastighed, ville vi kun have to variabler at ændre på, når vi introducerede bygninger, trafik eller begge dele.



Figur 6.3: Banens opbygning med segmenter.

Som det fremgår af figur 6.3, bestod testmiljøet af en lige vejbane opdelt i en række segmenter. Disse segmenter havde til formål at udsætte testpersonen for en stigende sværhedsgrad af kørebanen, ved at starte ud med en lige landevej der begyndte at svinge mere og skarpere som banen blev gennemkørt.

Vi udviklede et controller-script til testlederen, så denne nemt og uden for meget spildtid for testpersonen, kunne skifte mellem de forskellige conditions, fjerne speedometeret og genstarte simulatoren. Altsammen ved enkelte tastetryk på tastaturet.

6.5 Procedure

Eksperimentet blev udført ved at testpersonen blev budt velkommen af testlederen og introduceret til forsøget. Personen blev sat i bilsædet og der blev lavet justeringer af sædets placering for at imødekomme benlængde i forhold til pedalerne. For at give testpersonerne en mulighed for at vænne sig til bilen og styretøj, fandt vi nytte i den kasserede by ved at lade dem øve sig i at køre bilen rundt i byen, indtil de følte de havde fået styr på bilen.

For at tilfældiggøre rækkefølgen af conditions og stillede opgaver, havde vi udviklet et script der ved hjælp af en random-generator havde produceret 16 dokumenter, en til hver testperson, som hver især havde en tilfældig

rækkefølge af conditions og opgaver som testpersonen skulle udføre. Alt efter hvilken condition der stod først på dokumentet, blev denne hentet først ind i simulatoren, og testpersonen blev informeret om at formålet med de fire første øvelser var at køre strækningen, holde en konstant fart på 60 km/t samt udføre de opgaver som testlederen stillede hvert 30. sekund, når testpersonen selv følte at det var forsvarligt at løse dem. Efter de fire første conditions var blevet udført i tilfældig rækkefølge, blev testpersonen informeret om, at eksperimentet nu blev gentaget, men denne gang uden et synligt speedometer, og testpersonen skulle hermed holde en fart denne selv fandt forsvarlig. Formålet med de to forskellige hastigheder var at undersøge, om der kunne spores en forskel i testpersonens fornemmelse for fart, hvis denne ikke vidste hvor hurtigt bilen kørte.

Opgaver der blev stillet under forsøget:

- Play/Pause.
- Skift sang (fremad/tilbage).
- Skru op / Skru ned.
- Sig sangens navn.

For at undgå, at testpersonen kendte sangen der blev spillet, og derved kunne løse opgaven uden at kigge, havde vi ændret sangenes titler på iPod'en.

De fire conditions var:

1. Landevej uden bygninger og uden trafik og fodgængere.
2. Landevej med bygninger og uden trafik og fodgængere.
3. Landevej uden bygninger og med trafik og fodgængere.
4. Landevej med bygninger og med trafik og fodgængere.

Alle disse conditions blev gennemkørt på den samme bane som vist på figur 6.3. Efter de 8 gennemkørsler var overstået, blev testpersonen takket for sin deltagelse og interviewet kortvarigt for her at afklare deres subjektive holdning til simulatoren, og deres egen vurdering af hvorvidt de havde en følelse af tilstedeværelse og fornemmelse for fart under eksperimentet. For en sammenfatning af interviewets resultater se sektion 7.4 og for en oversigt over spørgsmål og svar se appendiks A.1.

6.6 Databehandling

Til at behandle den samlede data blev der benyttet forskellige metoder alt efter hvilken data der var tale om. Der blev behandlet data for:

- Bilens hastighed og placering på vejbanen i simulatoren.
- Øjefokuseringer.
- Svar på spørgsmål under interviewet.

6.6.1 Bilens hastighed og placering

Ved hjælp af det udviklede logging-script beskrevet i implementationsafsnittet sektion 5.5, havde vi ved eksperimentets afslutning en stor mængde JSON-filer som indeholdt data for bilens fart og placering hvert halve sekund. Vi udviklede et program til at konvertere JSON-dataen så den kunne forstås af statistikprogrammet R¹, for herved at kunne køre efterfølgende analyser på tallene.

6.6.2 Øjefokuseringer

Det følgende afsnit er foruden de eksplicit angivende kilder baseret på [3].

Øjefokuseringer er ofte brugt når der skal måles på tilstedeværelse i en bilsimulator og er en accepteret måleenhed for sammenhængen mellem den visuelle opmærksomhed og kørselspræstationen. Vi målte på både antallet samt varigheden af de øjefokuseringer en testperson har væk fra vejen under deres kørsel.

Når der kigges på varigheden af øjefokuseringer kan de med fordel inddeles i 3 kategorier: Mindre end 0,5 sekund, mellem 0,5 og 2,0 sekunder og over 2 sekunder. Denne opdeling er valgt fordi øjefokuseringer hvor øjet dvæler ved noget er typisk over 0,5 sekunder og det kan dermed bruges som en indikator for hvor testpersonens visuelle opmærksomhed befinder sig. Derudover har forskning vist at føreren af en bil nødtigt vil fortsætte uden information fra vejen i mere end 2 sekunder, hvilket også er i overensstemmelse med tommelfingerreglen om at holde afstand på 2 sekunder til forankørende for at kunne nå at reagere på hastighedsændringer. Derfor benyttes grænseværdierne 0,5 sekund og 2 sekunder som skillelinje mellem de tre kategorier, så testpersonernes øjefokuseringer kan kategoriseres i forhold til hvor vidt øjefokuseringen er ubetydelig (under 0,5 sekund, hvor øjet ikke dvæler ved noget), betydelig (0,5 - 2 sekunder, hvor øjet fokuserer på noget andet end vejen) eller den er alvorlig (over 2 sekunder, hvor øjet fokuserer på andet i så lang tid at der er fare for at man ikke når at reagere på information fra vejen).

For at tælle øjefokuseringer blev der benyttet videoredigeringssoftwaren VirtualDub² der havde mulighed for at give information om nummeret på den enkelte frame der var ved at blive afspillet. Alle DVD'er som blev optaget under eksperimentet blev herefter gennemgået for øjefokuseringer enkeltvis

¹<http://www.r-project.org/>

²<http://www.virtualdub.org/>

ved at observere det antal frames som der passerede fra testpersonen fjernede blikket fra kørebanen til denne kiggede tilbage igen, se figur 6.4.



Figur 6.4: Screendump af en testperson i færd med at udføre en opgave.

Selve kategoriseringen af øjefokuseringer foregik ved at to af forfatterne uafhængigt analyserede sig igennem alle videoerne fra hver af testpersonerne og noterede alle deres øjefokuseringer væk fra vejen. Længden af dem blev bestemt ved at registrere det frame hvor øjets bevægelse væk fra vejen startede (start-frame) samt det frame hvor øjets fokusering igen var tilbage på vejen (slut-frame). Dette start- og slut-frame blev så brugt til at kalkulere det antal frames øjefokuseringen havde taget, hvilket så kunne omregnes til sekunder og øjefokuseringen kunne herefter placeres i en af de tre kategorier. Efter de to forfattere havde registreret alle øjefokuseringer til en af de tre kategorier blev resultaterne sammenlignet med Kappa-koefficienten som er et udtryk for pålideligheden mellem de to sæt af øjefokuseringer som er registreret af to forskellige observatører. Da observatørerne ikke på forhånd kendte antallet af øje fokuseringer der bliver tildelt hver kategori betragtes distributionen af dem som fri og derfor er det anbefalet at man bruger en "free-marginal Kappa" [6]. For at udregne overensstemmelsen benytter vi os af "Free-Marginal Multirater Kappa" af Randolph[23], hvor vi har udregnet

overensstemmelsen mellem de to observatører for hver af testpersonerne. Den højeste værdi vi fik for en testperson var $\kappa_{free} = 0,99$ og laveste værdi for en testperson var $\kappa_{free} = 0,74$, og samlet set for alle testpersoner fik vi et gennemsnit på $\kappa_{free} = 0,88$. Med udgangspunkt i gennemsnitsværdien på 0,88 er graden af overensstemmelse nærmest perfekt ifølge Landis og Koch, mens den laveste værdi på 0,74 stadig betragtes som en omfattende overensstemmelse[12]. De uoverensstemmelser af kategoriseringer der forekom mellem de to observatører blev efterfølgende gennemgået i fællesskab for at træffe en fælles beslutning om kategoriseringen.

6.6.3 Svar på spørgsmål

Testpersonernes svar på interviewet blev afskrevet ved at tage videoerne der blev optaget under eksperimentet, og dernæst ordret udfylde et skema der indeholdt de spørgsmål, de var blevet stillet under interviewet, se appendiks A.1.1. Da visse besvarelser nogle steder overlappede hinanden og andre spørgsmål ikke blev svaret på, er visse felter efterladt tomme.

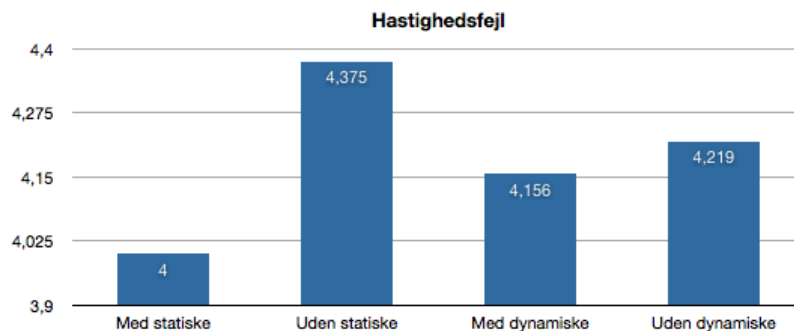
7

Resultater

Dette kapitel handler om resultaterne vi fik fra vores automatiske logning, øjefokuseringsanalyse og interview.

7.1 Hastighedsfejl

Vi så på antal hastighedsfejl der skete henholdvist med og uden dynamiske objekter, samt med og uden statiske objekter, se figur 7.1. Vi definerede en hastighedsfejl som at testpersonen kørte under 55km/t eller over 65km/t i de scenarier hvor de blev bedt om at køre 60km/t. Generelt set blev der lavet færre hastighedsfejl når der var statiske objekter end når der ikke var, og færre hastighedsfejl når der var dynamiske objekter end når der ikke var, men resultatet var ikke signifikant.

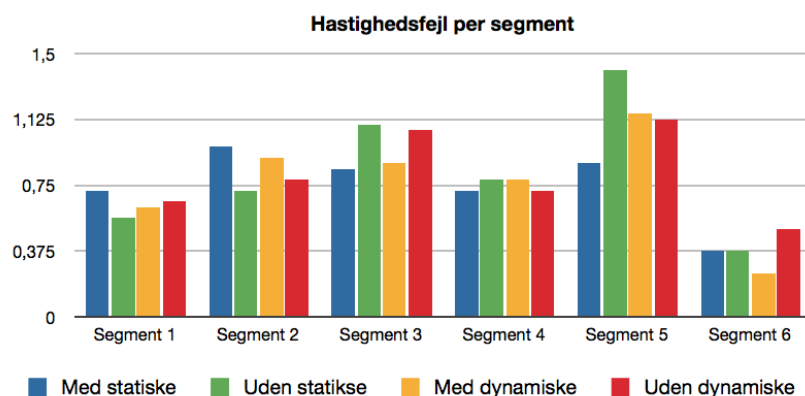


Figur 7.1: Gennemsnitlig hastighedsfejl.

Vi så på hvert segment af vejbanen for sig, se figur 7.2. Her viste det sig at der i de første to segmenter blev lavet flere fejl hvis der var statiske objekter end når der ikke var, mens det i segmenterne 3,4, og 5 var omvendt. I segment seks var der ingen forskel om der var statiske objekter eller der ikke var. Resultatet fra segment 2 og segment 5 var signifikante ($p < 0.03$). Segment 5 er den del af vejen der svinger mest. Det kan altså betyde at de statiske objekter hjælper med at give en fornemmelse for fart når vejen er svær at

KAPITEL 7. RESULTATER

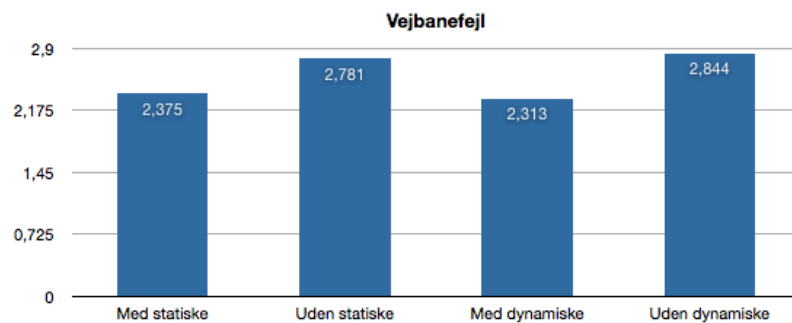
køre på. I segment 2 svinger vejen en smule men altså ikke nær så meget som i segment 5. Selvom vi har to signifikante resultater modsiger de hinanden og vi kan derfor ikke konkludere noget ud fra dette. Det viste sig også at der i halvdelen af segmenterne var der færre der lavede hastighedsfejl når der var dynamiske objekter end når der ikke var, og omvendt for den anden halvdel. Den halvdel hvor der blev lavet færre fejl indeholdte allesammen en stor del lige strækning. Her er det igen modsigende resultater og derfor kan vi ikke konkludere noget ud fra dette heller.



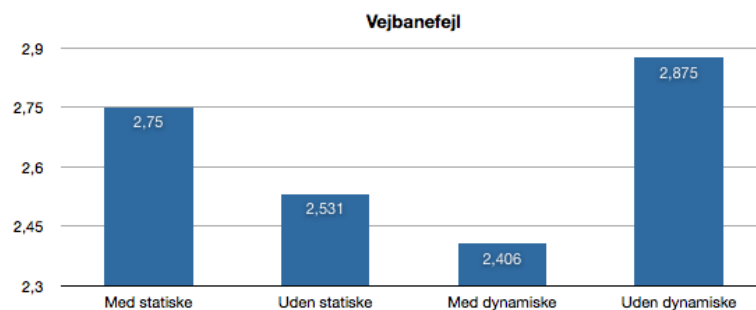
Figur 7.2: Gennemsnitlig hastighedsfejl pr segment.

7.2 Vejbanefejl

Kun to af de 16 forsøgspersoner kørte helt af banen, men vi så på hvor mange gange de kørte ud fra deres egen vejbane, f.eks ved at køre op på fortov eller over i den modkørende vejbane, se figur 7.3 og 7.4. Her var resultaterne modsigende. I scenarierne hvor forsøgspersonerne fik at vide at de skulle køre 60km/t blev der gennemsnitlig lavet færre vejbane fejl hvis der var statiske objekter end hvis der ikke var statiske objekter. Det kan skyldes at de nemmere kan se vejens forløb. Det samme gjaldt for de dynamiske objekter. Derimod blev der lavet flere fejl hvis der var statiske objekter end hvis der ikke var i scenarierne hvor forsøgspersonener kørte uden speedometer. Det kan skyldes at de bedre kunne se vejens forløb og de derfor fik en falsk tryghedsfølelse. Her blev der lavet færre fejl hvis der var dynamiske objekter end hvis der ikke var. Dette kan betyde at forsøgspersonerne følte et større pres for ikke at køre over i den anden vejbane da der var mulighed for at køre ind i en modkørende. Ingen af disse gennemsnitlige værdier var signifikante.



Figur 7.3: Gennemsnitlig vejbanefejl med 60km/t.



Figur 7.4: Gennemsnitlig vejbanefejl uden speedometer.

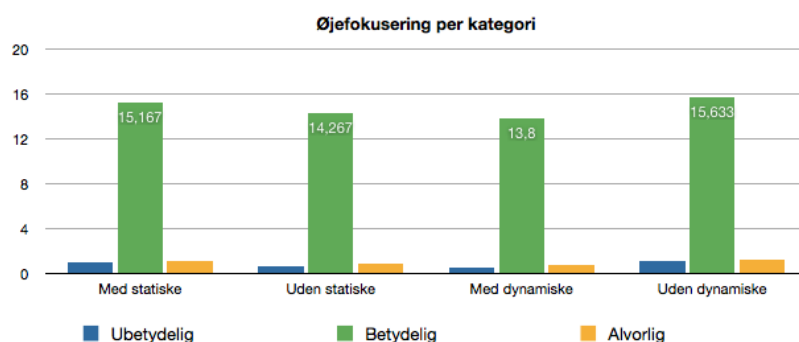
7.3 Øjefokusering

Vi kiggede på øjefokuseringer af 7 drenge og 8 piger, da DVD'en til den sidste mandlige testperson var beskadiget og dermed ikke mulig at lave efterbehandling på. Der var langt flest betydelige øjefokusering både hvor forsøgspersonen skulle køre 60 km/t og hvor de kørte uden speedometer. Der blev lavet færre øjefokuseringer med dynamiske objekter end uden dynamiske objekter. Dette kan betyde at forsøgspersonerne følte sig utrygge ved at kigge væk fra vejen når der var modkørende trafik. I scenariet hvor forsøgspersonerne skulle køre 60km/t er der flere øjefokuseringer med statiske objekter end uden, men i scenariet hvor de kørte uden speedometer var det omvendt. Ingen af disse gennemsnitlige værdier var signifikante.

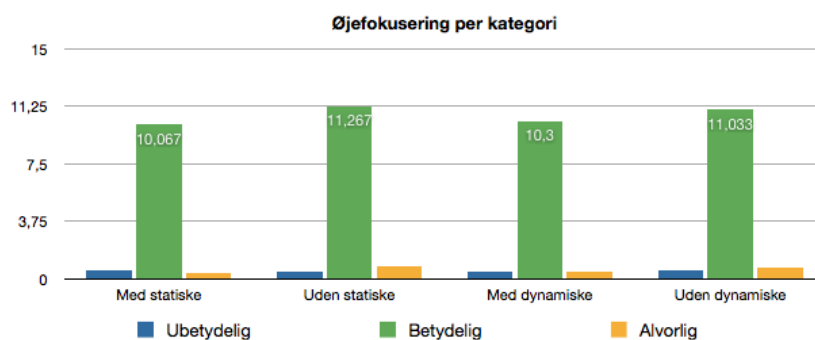
7.4 Interview

Under det efterfølgende interview blev testpersonerne stillet en række spørgsmål, som havde til formål at afdække hvordan den generelle oplevelse af simulatoren havde været, og hvilke kritikpunkter som kunne udpeges til fremtidige

KAPITEL 7. RESULTATER



Figur 7.5: Gennemsnitlig øjefokuseringer med 60km/t.



Figur 7.6: Gennemsnitlig øjefokuseringer uden speedometer.

forbedringer. Yderligere kunne testpersonernes svar give os et indtryk af, om vi havde udformet opgaverne tilstrækkeligt, og i hvor høj grad misforståelser og tvetydigheder var forekommet.

Emnerne som spørgsmålene afdækkede var:

- Testpersonens forudsætninger for deltagelse i eksperimentet.
- Testpersonens opfattelse af simulatoren.
- Testpersonens følelse af tilstedeværelse.
- Forslag til forbedringer.

Tabellerne for interviews med drenge og piger kan ses henholdsvis i bilag A.1.1 og A.1.2.

7.4.1 Forudsætninger

Som det fremgår i tabellerne, kørte størstedelen af dem der blev spurgt ikke bil hver dag. For de fleste var det i gennemsnit en gang om måneden.

Ingen af testpersonerne havde prøvet at køre i en decideret bilsimulator, men mange havde erfaringer med bilspil og 2 ud af de 15 testpersoner havde erfaringer med brugen af iPod Touch. Testpersonerne var i aldersgruppen 20-25 år og alle universitetsstuderende på naturvidenskabelige, humanistiske og samfundsvidenskabelige linjer.

7.4.2 Overblik

Underemner til nedenstående spørgsmål er følgende:

- Samfatning af udvalgte tal.
- Testpersonens opfattelse.
 - Generel opfattelse af softwaredelen.
 - Lyd og grafik.
 - Fysisk setup.
- Følelse af tilstedeværelse.
 - Naturlig eller kunstig oplevelse.
 - Udførelse af opgaver.
 - Foretrukne scenarier.
- Forslag til forbedringer

7.4.3 Sammenfatning

For at give et overblik over besvarelsesfordelingen af nogle af de spørgsmål, der bliver behandlet i nedenstående afsnit, er her præsenteret nogle tabeller over forskellige spørgsmål samt antallet af testpersoner med et givent svar. Alle tal er angivet i antal testpersoner.

Generel opfattelse af softwaredelen Antal testpersoner der henholdsvis fandt simulatoren tilfredsstillende og mangelfuld.

Tilfredsstillende	Mangelfuld
7	8

Benyttelse af sidevinduer Antal testpersoner der henholdsvis brugte, ikke brugte, og var usikre på deres brug, af simulatorens sideruder.

Ja	Nej	Usikker
4	7	4

KAPITEL 7. RESULTATER

Foretrukne scenarie Det foretrukne testscenarie som testpersonerne bedst syntes om at køre i. Henholdsvis: Med bygninger og med trafik, uden bygninger og uden trafik, med trafik og uden bygninger samt med bygninger og uden trafik.

Begge dele	Uden noget	Kun trafik	Kun bygninger
4	5	1	5

Fornemmelse for fart Antal testpersoner der henholdsvis følte, ikke følte og var usikre på om de følte en fornemmelse for fart under eksperimentet.

Ja	Nej	Usikker
4	6	5

Gennemgående forbedringsforslag Antal testpersoner der alle udtalte sig om en given problemstilling:

Problem	Antal testpersoner
Implementation af gearskifte	6
Hakkende/dårlig grafik	5
Fladt landskab	3
Styretøj	9

7.4.4 Opfattelse af simulatoren

Generel opfattelse af softwaredelen

Adspurgte om den generelle opfattelse af software-delen af simulatoren, var der en grundlæggende enighed om funktionaliteten fungerede godt. Vi spurgte til testpersonernes generelle indtryk og herunder også hvordan deres opfattelse af hvordan grafikken og lydsiden var. Testpersonernes generelle indtryk var blandet, en testperson udtalte eksempelvis *"Jeg syntes det var vildt fedt, rigtig godt!"*, hvor en anden udtalte mere kritisk at *"Jeg synes den gør det ret godt. Jeg følte dog, eksempelvis at det var sådan at man bare kunne køre i et helt forkert gear som man havde lyst til. Der var ikke den helt samme følelse mellem gear og fart som i en rigtig bil"*. Mange udtalte problemer med selve styringen af bilen, som på nogen virkede unaturlig især i svingene. For nogle af testpersonerne forekom ratslør og modstanden i rattet ikke som det gjorde i en rigtig bil.

Lyd og grafik

Adspurgte mere konkret til lyd- og grafiksiden fik vi i visse tilfælde mere konkret respons. Flere udtalte et problem med, at grafikken ikke kørte flydende især når der var bygninger til stede. Der blev flere gange nævnt at grafikken *"hakkede"*, blandt andet udtalte en af testpersonerne at *"[grafikken] var ikke*

noget at skrive hjem om. En meget flad verden uden bakker og den slags". Netop det faktum at vi ikke havde nogle forskelle i det virtuelle miljøes niveauer var et problem flere af testpersonerne belyste. De følte især problemet i sving, hvor de ikke havde mulighed for i god nok tid at se hvilken vej der skulle drejes. En testperson udtalte blandt andet "Det var svært at se langt frem i forhold til hvor radikale svingene var. Det var svært at vurdere hvor hurtigt man kørte og hvor skarpe svingene der kom var" og "Det er svært at se fremad hvordan svingene er, i forhold til hvad man kan i virkeligheden (...)".

Lydsiden var der ligeledes delte meninger om, og selvom et par af de adspurgte talte positivt om den, var der flere der påpegede at lyden ikke gik i harmoni med gearskiftet. En enkelt udtalte at *"Lyden var fin nok, jeg kunne høre når jeg skiftede gear."*, hvor en anden deltager konkret udtrykte sit problem med lyden i forhold til gearskiftet: *"Man kan ikke rigtig høre på motoren når man skiftede gear, så når man så sætter den i 4. så er det ligemeget om koblingen er nede, så hakker den bare op i omdrejninger, lyden er svær at forholde sig til"*. Flere omtalte lyden som *"monoton"*, og et par deltagere havde modstridende opfattelser af lydsidens indvirkning på gearskiftet: *"Jeg synes den gør det ret godt. Jeg følte dog, eksempelvis at det var sådan at man bare kunne køre i et helt forkert gear som man havde lyst til"* og *"Motorlyden var lidt skør. Ved ikke om den var realistisk. Syntes den lød meget som om man ikke kørte i det rigtige gear, men måske var det fordi jeg ikke gjorde det"*.

Fysisk setup

Angående det fysiske setup, herunder rat, gear og pedaler, brugen af bilsæde og flere skærme, var de fleste enige om at bilsædet fungerede godt. Der var uenighed omkring brugen af styretøj, hvor nogen udtalte at det fungerede godt. Blandt andet udtalte en af testpersonerne: *"Jeg synes det er ret godt. Rattet er lidt lille, men ellers er det godt"*, og *"Rattet fungerede meget godt. Det føltes godt at der var lidt slør i rattet og at den trak lidt i det"*. I forlængelse af lydsidens påpegede problemer udtalte en anden at *"Gearskiftet var svært at vurdere. Pedalerne var udmærket når man lige lærte følelsen af hvor meget man skulle trykke. Rattet fungerede også meget godt"*. Gearstangen og pedalerne blev kun nævnt af et fåtal af de adspurgte, hvor nogen påpegede at gearstangen var for lille i forhold til en normal bil, og at den sad for løst. En testperson omtalte pedalerne som *"meget plastic-agtige"* og at hvis de havde siddet bedre fast havde det givet en bedre oplevelse. Slutteligt påpegede en af deltagerene at *"Man skal altid lige vænne sig til en ny bil"*.

Nogle af deltagerne udtalte at de ikke havde kigget på andet end den midterste skærm, og slet ikke havde benyttet de to mindre skærme i hver side af simulatoren. En af testpersonerne udtalte at han helt havde glemt de var der indtil de kom til den 5. gennemkørsel. En testperson udtrykte

irritation ved skærmenes tilstedeværelse: *"Jeg brugte dem ikke rigtig, jeg syntes det var forvirrende når man kørte inde i byen at man hele tiden havde noget mørke i øjenkrogene"*. Dog påpegede et par af testpersonerne, at skærmene muligvis ubevidst har bidraget til helhedsindtrykket: *"Nej, det er kun den store jeg kigger på. Det kan være sideskærmene har påvirket mig ubevidst"*, *"Jeg kiggede på dem for at holde øje med hvor jeg var henne på vejen. Men jeg kiggede mest ud af øjenkrogen"*. En deltager udtrykte eksplicit et positivt indtryk af skærmenes tilstedeværelse: *"Jeg synes man fik et rigtig godt overblik. Selvom man ikke kigger direkte på skærmene så fik man alligevel indtrykket med"*.

7.4.5 Følelse af tilstedeværelse

Testpersonerne blev stillet en række spørgsmål med henblik på deres egen følelse af tilstedeværelse, herunder om de følte at de havde en fornemmelse for hastigheden. Vi spurgte dem om de følte at det var som at køre i en rigtig bil, eller om de havde en kunstig følelse under eksperimentet, og yderligere om hvordan de følte sig tilpas under udførelsen af de stillede opgaver.

Naturlig eller kunstig oplevelse

Adspurgte om de følte at oplevelsen var naturlig, observerede vi i flere tilfælde at testpersonerne havde svært ved at sætte præcise ord på deres følelser. Personerne kom frem til blandede konklusioner hvor nogen mente at de havde en *"meget god"* følelse af tilstedeværelse, samt at det *"faktisk føltes rimeligt naturligt"*. En testperson udtrykte også en følelse af gradvis tilstedeværelse og udtalte: *"[Det] føles som at køre en bil, men rattet 'hang' i det. Det begyndte at føles lidt naturligt til sidst"*. Der forekom også direkte modstridende udtalelser såsom *"Jeg syntes det føltes naturligt"* og *"Det føltes meget kunstigt"*. Især to testpersoner kom med mere fyldstgørende udtalelser omkring deres følelser omkring tilstedeværelse:

"Det føltes som noget andet. Nogle af tingene i simulatoren går igen i en rigtig bil, man skal være meget opmærksom, man skal korrigere, man skal fokusere, man kan ikke kigge væk fordi så ender det med at man kører ud i den ene side. På den måde går der ting igen. Men der er et stykke vej igen, det synes jeg. Hastigheden havde jeg svært ved at bedømme. Man får ikke den rigtige fornemmelse af hvilken hastighed der er farlig at køre med. Fordi det er en simulator, man føler ikke at man bliver rykket fysisk".

"Det føltes mærkeligt at tage et næsten 90-graders sving med 60 km/t, dem ville jeg normalt have taget med meget lavere hastighed. Mange af tingene virkede meget naturligt, når man kørte på lige strækninger så kunne man sidde og slappe af, men ligeså snart der kom et sving, så blev det svært. På grund af det med lyden forsøgte jeg såvidt at undgå at skifte gear. Det var svært at vurdere".

Udførelse af opgaver

Med hensyn til udførelsen af opgaver undervejs følte næsten alle testpersonerne enstemmigt at det forstyrrede deres kørsel. Til trods for at testlederen i alle tilfælde understregede overfor testpersonerne inden forsøget gik i gang, at de kun skulle udføre en opgave når de følte det forsvarligt, tyder flere af testpersonernes svar på at de troede de skulle udføre opgaven med det samme. Eksempelvis lød et svar, på spørgsmålet om opgaveløsningerne forstyrrede: *"Ikke specielt, men lidt i svingene", "Det gik ikke godt når man skulle lave opgaver midt i et sving"* og *"Jeg følte, at jeg blev tvunget til at løse opgaver på tidspunkter jeg normalt ikke ville have gjort det. Havde jeg selv kørt havde jeg ikke skiftet sang i et sving eksempelvis"*. Adspurgt om opgaveløsningerne gav testpersonerne en følelse af stress at være under pres, var flere enige om at de følte sig pressede: *"Ja, det gik udover min kørsel, der var flere gange hvor jeg kiggede væk, og så skulle jeg med det samme korrigere når jeg kiggede op igen. Jeg er meget uopmærksom på hvad der foregår når jeg kigger væk"*, og *"Ja jeg følte en lille smule stress, du var god til at spørge lige i svingene"*. Flere testpersoner gav udtryk for, at de troede testlederen med vilje bad dem løse opgaver netop i svingene, selvom realiteten var at spørgsmålene blev givet på tid, og at det altså var et tilfælde hvornår opgaverne kom. En enkelt testperson konkluderede endda at det måtte være en del af vores forsøg netop at bede folk løse opgaver specifikt i svingene: *"Jeg prøvede at udføre opgaven med det samme, da jeg antog at det var det I ville have mig til, siden du tit bad mig gøre det i et sving"*.

Fortrukne scenarier

Slutteligt spurgte vi testpersonerne, om hvilken en af de 4 scenarier se afsnit 6.1 de selv foretrak at køre i. Her sagde 5 ud af de 15 at de foretrak at køre i scenariet hvor der kun var vejen, og altså uden biler og bygninger. Kun en ud af de fem uddybede deres svar og sagde *"Uden bygninger og uden fodgængere. Jeg kunne ikke lide når der var modkørende trafik."* 4 foretrak scenariet hvor der både var trafik og bygninger, men det er muligt at det førnævnte problem med det flade landskab kan have bidraget til denne følelse, da flere nævnte at bygningerne hjalp med at give en idé om hvornår vejen drejede: *"Med bygninger, så kunne man se hvilken vej vejen gik, fordi landskabet var så flat. Trafikken var ligemeget. Man mødte jo ikke rigtig noget i sin egen kørebane"*. Opfattelsen af, at trafik og fodgængere var ligegyldigt for deres opfattelse af tilstedeværelse blev delt af flere. Et par af de adspurgte udtrykte også at bygningerne hjalp dem til at føle hvor hurtigt de kørte, hvilket var en af de vigtigste udtalelser i forbindelse med vores hypoteser: *"Det var nemmere når der var bygninger for at føle hvor hurtigt jeg kørte. Jeg kørte måske mere forsigtigt når der var mennesker og biler, men jeg ved det ikke"*.

7.4.6 Forslag til forbedringer

Som allerede nævnt, havde testpersonerne forslag til forbedringer i deres kritikpunkter udtalt i andre spørgsmål. Ting som det flade landskab, lydsiden og den hakkende grafik var steder hvor vi har noget at arbejde med. Spurgt direkte til forslag om forbedringer præciserede et par af testpersonerne deres holdninger til måden hvorpå vi har implementeret gearskiftet. *"Der føltes ingen ændringer når jeg trykkede koblingen ned"*, og mere præcist: *"Det fysiske setup er rigtig godt. Jeg ved ikke om det var muligt at den ventede med at skifte gear til man var koblet ud. Det virkede meget underligt i starten at den gjorde det. Altså når man kobled ud så gik den i neutralt, i forhold til lyden. Så sætter du den i gear, så kører den op i omdrejninger før du har løftet koblingen. Også eventuelt i sidevinduerne, sætte sidespejlene der så man har en grund til at kigge mere der. Det er sjældent jeg kigger ud af sidevinduerne, men det er tit jeg bruger spejlene, når jeg kører bil"*. En anden testperson udtrykte også i forlængelse af udtalelsen om spejlene, at han savnede et bakspejl da det at regelmæssigt kigge i et bakspejl var en central del af det at køre bil. En enkelt testperson påpegede at han følte det underligt at man var placeret midt i bilen: *"Placeringen af personen i bilen. Man sidder midt i. Det kunne være smart at man sad i den rigtige side, og at man også kunne se de 'dårlige' vinkler, blinde vinkler. Og spejle, bakspejl og sidespejl. Og en bedre lyd"*. Slutteligt udtrykte to testpersoner at de savnede en følelse af bevægelse i bilsædet og rattet: *"Der kunne godt komme flere vibrationer, hvis man kører ind i en bil eller op på fortovskanten"* og endeligt: *"Mere bevægelse i sædet. Ellers synes jeg det var lækkert at bruge"*.

8

Diskussion

Dette kapitel omhandler de fejlgrupper der var ved vores eksperiment samt forslag til fremtidigt arbejde på næste semester.

8.1 Fejlgrupper

Efter endt eksperimentudførelse og analyse af resultater, er der kommet flere ting frem, som vi har måttet konkludere, kunne have haft en negativ indflydelse på hvordan resultaterne er endt med at se ud. I dette afsnit vil vi se på hvilke fejlgrupper der kan have været i vores eksperiment, og hvorfor det forholder sig således.

Følgende ting anser vi har haft indflydelse på vores resultater:

- For lidt forskel imellem forskellige scenarier.
- Sværhedsgraden af scenarierne har været for lav.
- Behandlingen af de stillede opgaver har mangler.

8.1.1 Forskel imellem forskellige scenarier

En af de ting som vi anser for at være et problem, og sandsynligvis har haft en indflydelse på hvorfor mange af vores resultater ikke er signifikante, kan have været en for lille forskel imellem de forskellige scenarier. Når vi ser på resultaterne er det kun i begrænset omfang, at vi ser en særlig stor forskel på farten og antal af øjefokuseringer. Det har fra start af været svært for os at bedømme, hvorvidt de forskellige scenarier ville kunne frembringe nogle resultater der ville afvige fra hinanden. Vi har måttet indse at vores forskellige scenarier hvor der er ændret på antal af statiske og dynamiske objekter, ikke har været nok til at give signifikante resultater. Om dette skyldes at der i en bilsimulator ikke er stor forskel på om statiske og dynamiske objekter er til stede, eller om det er fordi vi ikke har gjort brug af nok af disse objekter, er stadigvæk uklart.

KAPITEL 8. DISKUSSION

8.1.2 Sværhedsgraden af scenarierne

Sværhedsgraden af de forskellige scenarier, har været for lav til at give store nok udsving i resultaterne. Her tænker vi især på farten der er kørt med i de forskellige scenarier. Idéen var til at starte med at lave alle scenarier på en plan overflade, uden at gøre brug af bakker. Dette gjorde vi fordi vi ville sikre os, at forskellen og den faktor vi målte på, kun var forskel på antal af statiske og dynamiske objekter. Men dette betød at vores testpersoner havde forholdsvis let ved at holde farten, da de blot behøvede at holde foden på speederen det samme sted, når de først havde opnået den ønskede fart. Sværhedsgraden burde have været ændret således at alle scenarier havde haft det samme kurvede terræn.

8.1.3 Behandlingen af de stillede opgaver

En fejlkilde som vi desværre først fandt frem til efter den udførte test, var at måden de stillede opgaver var blevet givet til testpersonerne indeholdte fejl. Her er der tale om selve udførelsen, der indbefatter at give testpersonen en ny opgave hvert 30. sekund. Problemet med denne metode er, at antallet af stillede opgaver derved afhænger af den tid, testpersonen bruger på at køre igennem det enkelte scenarie. Dette har betydet at hver gang vi har haft en testperson der har kørt langsomt igennem de forskellige scenarier, har denne testperson fået stillet flere opgaver end de personer der har kørt hurtigt. Da antallet af især øjefokuseringer har en direkte kobling til antallet af opgaver, må det antages at det har haft en indflydelse på vores resultater. Det er samtidigt heller ikke til at sige, om dette også har haft en indflydelse på antal af kørefejl, da flere opgaver sandsynligvis også har en virkning her.

8.2 Fremtidigt arbejde

I dette afsnit vil vi prøve at fremlægge de idéer og tanker vi har gjort os om det fremtidige arbejde vi står over for på tiende semester. Med udgangspunkt i simulatoren som vi har udviklet og eksperimentet vi har udført med de dertilhørende problemstillinger er vi kommet frem til to retninger som vi mener kunne være spændende at tage fat på i det videre specialeforløb.

8.2.1 Simulator

Som det fremgår af afsnit 8.1 er der en række forbedringer der kunne gøre simulatoren og dermed også eksperimentet bedre ved at give testpersonerne en forbedret følelse af tilstedeværelse. Især de manglende bakker og niveauforskelle i simulatoren, som det også er omtalt flere steder i interviewet (7.4), er et punkt vi gerne vil forbedre i en eventuel fremtidig version af simulatoren. Niveauforskellene ville udover at stemme mere overens med et virkeligt

vejforløb også give testpersonerne nemmere ved at se vejen fremad og registrere sving som flere udtrykte de havde svært ved at opdage i tide når vejen var flad og flød ud i et.

Et andet problem i relation til de manglende niveauforskelle var at sværhedsgraden og forskellen på de forskellige scenarier ikke var tilstrækkelig stor til at give os signifikante resultater der stemmer overens med vores hypoteser. Vi har f.eks. tænkt at kørsel i den by vi har designet ville give et langt sværere scenarie med en rute gennem en by med lyskryds osv. og at forskellen mellem de forskellige scenarier ville blive større med alle de forskellige statiske og dynamiske objekter der kan og i virkeligheden vil optræde i en by.

Desuden synes vi også det kunne være interessant at se hvad enkelte dele af simulatoren har af indflydelse på kvaliteten af simulatoren som helhed. Vi har f.eks. talt om at fjerne de 2 skærme i siderne for at undersøge om dette vil have en indflydelse på testpersonernes kørselspræstation. Som det fremgår af afsnit 7.4.3 var det kun 4 ud af 15 der svarede ja til at de brugte sideruderne, og vi synes derfor det kunne være interessant om det overhovedet vil have nogen betydning at vi fjernede dem. Det kunne jo f.eks. også tænkes at testpersonerne ubevidst brugte disse sidevinduer under deres kørsel og orienterede sig ud af øjenkrogen på sideruderne som man eksempelvis gør i en rigtig bil.

Som det også fremgår af interviewene er der flere testpersoner der udtaler at udstyret i simulatoren (rat, pedaler og gear) kunne gøres bedre og mere realistisk i forhold til at imitere udstyret fra en rigtig bil. Især måden hvorved gear og kobling fungerede sammen kunne forbedres således at man først skiftede gear når man havde koblet ud igen, og at lyden også opfyldte dette. Dette ville give en bedre idé om hvilket gear man er i og i stedet for den først gik op og ned i omdrejninger når foden var løftet fra koblingen igen. Et andet kritikpunkt var styretøjet og at det føles som om der var lidt slør i rattet, hvilket i en eventuelt forbedret version ville være muligt at indstille endnu bedre, så rattet kunne opføre sig som rattet i en rigtig bil.

Udover dette blev der i interviewene også udtrykt behov for spejle, både sidespejle og bakspejl, hvilket også kunne tænkes at blive implementeret i en forbedret version af simulatoren for at højne tilstedeværelsen hos testpersonerne. Et andet ønske var at sædet var placeret i den rigtige side fremfor midt for skærmen og at sæde kunne indeholde bevægelse og at det f.eks. ville reagere på bump når man kører over fortovskanter eller lignende. Den sidste del er selvfølgelig ikke nem at realisere men man kan som Weinberg og Harsham har gjort i [27] investere i et sæde der kan simulere vibrationer og hældninger i bilen, men da dette også koster omkring 15.000 US Dollars er det ikke helt i tråd med vores mål om at udvikle en prisbillig simulator.

8.2.2 HUD - Head-Up Display

Ved starten af niende semester var det interessen for et Head-Up Display og hvad et sådan display kunne bruges til i en bil der ledte os frem til at udvikle en bilsimulator. Vi fandt hurtigt ud af, at for at teste og afprøve et HUD ville det kræve en bilsimulator at teste det i, blandt andet pga. sikkerhedsmæssige og andre problemer der vil være i forbindelse med at teste et HUD-system i en rigtig bil, se afsnit 2.7. Desuden vil testen af en prototype af et HUD-system i en bilsimulator være i fin tråd med det fokus vi har haft på lave omkostninger i forbindelse med udviklingen af simulatoren.

I begyndelsen af projektforløbet havde vi møder med forskellige relevante personer, bl.a. en trafikforsker, for at finde ud af hvilke systemer der var mulige at implementere i et HUD og hvilke systemer der fra et brugersynspunkt kunne være behov for. Efter disse møder blev det hurtigt klart at vi i udviklingen af et HUD-system skulle have et system der ville være i stand til at forbedre de primære køreopgaver, dvs. kunne forbedre kørselspræstationen hos førerne af en bil, fremfor systemer med fokus på sekundære køreopgaver som f.eks. at betjene musikanlægget.

Med dette i tankerne har vi tænkt på en idé til et HUD-system kunne være et system der kunne markere det videre forløb af vejen foran bilen på bilens forrude. Dette ville gøre at føreren af bilen kunne se vejens forløb på uoplyste veje om natten eller i vejrforhold der gjorde det svært at se vejen f.eks. hvis der ligger sne eller det er tåget. I artiklen "Visual Longitudinal and Lateral Driving Assistance in the Head-Up Display of Cars" [18] har Tönnis et al. lavet et lignende HUD-system hvor de forsøger at hjælpe bilisten ved at angive bremselængden og retningen bilen vil bevæge sig i med den nuværende fart og retning. Dette system skulle hjælpe føreren af en bil til at holde den rette afstand til forankørende biler samt hjælpe med at holde den rette kurve f.eks. ind i et sving. Det kunne også tænkes at systemet fra Tönnis et al. sammen med vores idé om at markere vejens forløb kunne være et samlet HUD-system der ville være med til at forbedre kørslen hos bilister.

En anden idé til et HUD-system eller som en del af et samlet system som det ovenfor kunne være en hjælp til at tilpasse hastigheden. Vi har forestillet os et system der hele tiden viste hvilken hastighed man må køre med et pågældende sted og at systemet kunne hjælpe dig med at holde denne hastighed. Ligeledes kunne det også tænkes at systemet kunne hjælpe med at tilpasse farten ind i et sving eller på vej mod andre forhindringer så man undgår at tage disse med for høj fart.

9

Konklusion

Denne rapport skulle bl.a. give svar på hvilke faktorer, der påvirker testpersoners følelse af tilstedeværelse. Vi har gennem et litteraturstudie fået et indblik i hvilke faktorer der kan have en indflydelse på følelsen af tilstedeværelse i en bilsimulator, og har med denne viden kunne opsætte vores eksperiment for at undersøge hvilken betydning dynamiske og statiske objekter har på det.

Som forklaret i afsnit 8.1, så har der været en del fejlgrupper i udførelsen af eksperimentet, hvilket kan have haft indflydelse på vores resultater fra eksperimentet. Ser vi på resultaterne i kapitel 7, så har vi ikke kunne finde frem til signifikante resultater, der taler for at en testpersons følelse af tilstedeværelse bliver påvirket af bestemte typer objekter, hvad enten det er statiske eller dynamiske. Det kan dog se ud til at statiske objekter kan have en virkning i situationer hvor vejen er svær at følge.

Et anden vigtig problemstilling, var hvorledes der kunne udvikles en prisbillig bilsimulator, hvor der var mulighed for at manipulere med faktorer, som eksempelvis statiske og dynamiske objekter. Dette blev realiseret ved at benytte Unity 3, som opfyldte de krav beskrevet i kapitel 3, hvor det bl.a. var et krav at der skulle kunne udvikles hurtigt og koden skulle være overskuelig. Vi tog udgangspunkt i et open source projekt fra Unity Studios som indeholdte en fungerende bilfysik. Vi tilføjede herefter selv selvstyrende biler, mulighed for at styre med rat, gear og pedaler, fodgængere og automatisk logning til simulatoren. Vi byggede også to virtuelle miljøer, en by og en landevej ved hjælp af programmet CityScape.

En af udfordringerne ved at lave en prisbillig bilsimulator, var for det første at finde den rigtige udviklingsplatform, som opfyldte alle de krav vi havde. Vi ønskede en udviklingsplatform som skulle være hurtig at udvikle i, samtidig med at det skulle være let at skabe sig et overblik over udviklingsprojektet. Da det kom til udvikling af det virtuelle miljø til bykørsel og landevej, som beskrevet i afsnit 4.2, kom der igen udfordringer i forhold til at udvikle et virtuelt miljø, der ikke optog for mange systemressourcer fra den kørende maskine. En anden udfordring var det rent fysiske setup hvor vi så vidt

KAPITEL 9. KONKLUSION

muligt skulle prøve at imitere det at sidde i en rigtig bil. Med begrænsede midler til rådighed fik vi opsat et fungerende setup med et bilsæde monteret på paller, samt monteret rat, pedaler og gearstang placeret som i en rigtig bil.

Det kan altså konkluderes at de opstillede hypoteser ikke er blevet bevist ud fra vores eksperiment, dog anser vi det for muligt at resultatet kunne være anderledes med et forbedret eksperiment baseret på hvad vi har lært gennem forløbet.

10

Bibliografi

- [1] Talal Al-Shihabi and Ronald R. Maurant. A framework for modeling human-like driving behaviors for autonomous vehicles in driving simulators. *AGENTS '01 Proceedings of the fifth international conference on Autonomous agents*, 2001.
- [2] Talleah Allen and Ronald Tarr. Driving simulators for commercial truck drivers - humans in the loop. *PROCEEDINGS of the Third International Driving Symposium on Human Factors in Driver Assessment, Training and Vehicle Design*, 2001.
- [3] Kenneth M. Bach, Mads G. Jæger, Mikael B. Skov, and Nils G. Thomassen. You can touch, but you can't look: interacting with in-vehicle systems. *CHI '08 Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, 2008.
- [4] Kenneth M. Bach, Mads G. Jæger, Mikael B. Skov, and Nils G. Thomassen. Evaluating driver attention and driving behaviour: Comparing controlled driving and simulated driving. *BCS-HCI '08 Proceedings of the 22nd British HCI Group Annual Conference on People and Computers: Culture, Creativity, Interaction - Volume 1*, 2008.
- [5] Kenneth M. Bach, Mads G. Jæger, Mikael B. Skov, and Nils G. Thomassen. Interacting with in-vehicle systems: understanding, measuring, and evaluating attention. *BCS-HCI '09 Proceedings of the 23rd British HCI Group Annual Conference on People and Computers: Celebrating People and Technology*, 2009.
- [6] Robert L. Brennan and Dale J. Prediger. Coefficient kappa: Some uses, misuses, and alternatives. *Educational and Psychological Measurement* 1981, 41, 1981.
- [7] Kar-Hai Chu and Sam Joseph. Using second life to demonstrate a concept automobile heads up display (a-hud). *MobileHCI '08 Proceedings of the 10th international conference on Human computer interaction with mobile devices and services*, 2008.

KAPITEL 10. BIBLIOGRAFI

- [8] Gizmodo.com. Toyota's new driving simulator looks like fairground attraction, 19. december 2010. URL <http://gizmodo.com/gadgets/drive-safely/toyotas-new-driving-simulator-looks-like-fairground-attraction-326240.php>.
- [9] H. S. Kang, M. K. Abdul Jalil, and Musa Mailah. A pc-based driving simulator using virtual reality technology. *VRCAI '04 Proceedings of the 2004 ACM SIGGRAPH international conference on Virtual Reality continuum and its applications in industry*, 2004.
- [10] Andras Kemeny and Francesco Panerai. Evaluating perception in driving simulation experiments. *TRENDS in Cognitive Sciences Vol.7 No.1 January 2003*, 2003.
- [11] Aalborg Kommune. 3d bymodel af aalborg., 19. december 2010. URL http://www.aalborgkommune.dk/0m_kommunen/kort-over-kommunen/hoejdeinfo/Sider/3DBymodel.aspx.
- [12] J. Richard Landis and Gary G. Koch. The measurement of observer agreement for categorical data. *Biometrics*, Vol. 33, No. 1, March 1977, 1977.
- [13] Jonathan Lazar, Jinjuan Heidi Feng, and Harry Hochheiser. *Research Methods*. Wiley, 2010.
- [14] Myonghee Lee and Gerard J. Kim. Effects of heightened sensory feedback to presence and arousal in virtual driving simulators. *VRCAI '09 Proceedings of the 8th International Conference on Virtual Reality Continuum and its Applications in Industry*, 2009.
- [15] Duoduo Liao. A high-fidelity immersive cluster-based driving simulator for transportation safety research. *VRCAI '06 Proceedings of the 2006 ACM international conference on Virtual reality continuum and its applications*, 2006.
- [16] Second Life. Second life, 3d virtual world community., 19. december 2010. URL <http://secondlife.com>.
- [17] Logitech.com. Products / gaming / wheels, 19. december 2010. URL <http://www.logitech.com/en-us/gaming/wheels/devices/5184>.
- [18] Gudrun Klinker Marcus Tönnis, Christian Lange. Visual longitudinal and lateral driving assistance in the head-up display of cars. *ISMAR '07 Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, 2007.

-
- [19] Ronald R. Mourant and David Refsland. Developing a 3d sound environment for a driving simulator. 2003.
- [20] PixelActive3D. Cityscape., 19. december 2010. URL <http://pixelactive3d.com/>.
- [21] Procedural. Cityengine., 19. december 2010. URL <http://www.procedural.com/cityengine/>.
- [22] racer.nl. Dolphinity racer., 19. december 2010. URL <http://www.racer.nl>.
- [23] Justus J. Randolph. Free-marginal multirater kappa (multirater κ_{free}): An alternative to fleiss' fixed-marginal multirater kappa. *Presented at the Joensuu Learning and Instruction Symposium 2005*, 2005.
- [24] Hatice Sancar, Kursat Cagiltay, Veysi Isler, Gizem Tamer, Neslihan Ozmer, and Utkan Eryilmaz. Developing a validation methodology for educational driving simulators and a case study. *Proceedings of the 13th International Conference on Human-Computer Interaction. Part IV: Interacting in Various Application Domains*, 2009.
- [25] TORCS. The open racing car simulator., 19. december 2010. URL <http://torcs.sourceforge.net>.
- [26] unity3d.com. Unity 3, a game development tool., 19. december 2010. URL <http://www.unity3d.com>.
- [27] Garrett Weinberg and Bret Harsham. Developing a low-cost driving simulator for the evaluation of in-vehicle technologies. *AutomotiveUI '09 Proceedings of the 1st International Conference on Automotive User Interfaces and Interactive Vehicular*, 2009.



Appendikser

A.1 Interviews

A.1.1 Dreng

Spørgsmål / Testperson	Testperson1	Testperson2	Testperson3
Hvor gammel er du?	25	24	24
Hvilket semester?	9	9	Kursist (9)
Hvad læser du?	Software	Software	Software
Hvor mange år har du haft kørekort?	7	6	6
Hvor tit kører du bil?	Hver dag.	1 gang hver anden måned.	Et par gange om måneden.
Har du erfaring med bilsimulatorer?	Ikke ret meget.		Ja.
Har du erfaring med iPod/iPhone?	Nej.	Nej	Ja.
Hvad synes du generelt om simulatoren?	Simulatoren var udmærket, men det var svært at vurdere hvor hurtigt man kørte.	Styringen føles urealistisk. Bilens reaktion når man drejer på rattet.	Fint nok.
- Grafikken	Det var svært at se langt frem i forhold til hvor radikale svingene var. Det var svært at vurdere hvor hurtigt man kørte og hvor skarpe svingene der kom var.		Det hakker en lille smule.
- Lyden	Man kan ikke rigtig høre på motoren når man skiftede gear, så når man så sætter den i 4. så er det ligemeget om koblingen er nede, så hakker den bare op i omdrejninger, lyden er svært at forholde sig til.		
Hvad synes du om det fysiske setup?	Det føltes ikke som en rigtig bil, men indtrykket føltes meget naturligt. Man ved selvfølgelig godt at man sidder i en simulator, men man glemmer ret hurtigt at tænke på det. Det føltes ret naturligt, det er meget godt med rat og gearskifte og sådan noget.	Sædet er godt. Pedalerne er gode. Rattet føles ikke som et rigtigt rat. Modstanden er noget anderledes end et rigtigt rat. Men ellers kan man godt få den samme fornemmelse, når man kører bil.	Ja det føles meget rigtigt.

Hvordan var følelsen, føltes det som at køre i en rigtig bil, eller var det kunstigt?	Det føltes mærkeligt at tage et næsten 90-graders sving med 60 km/t, dem ville jeg normalt have taget med meget lavere hastighed. Mange af tingene virkede meget naturligt, når man kørte på lige strækninger så kunne man sidde og slappe af, men ligeså snart der kom et sving, så blev det svært. På grund af det med lyden forsøgte jeg såvidt at undgå at skifte gear. Det var svært at vurdere.	Det føltes som noget andet. Nogle af tingene i simulatoren går igen i en rigtig bil, man skal være meget opmærksom, man skal korrigere, man skal fokusere, man kan ikke kigge væk fordi så ender det med at man kører ud i den ene side. På den måde går der ting igen. Men der er et stykke vej igen, det synes jeg. Hastigheden havde jeg svært ved at bedømme. Man får ikke den rigtige fornemmelse af hvilken hastighed der er farlig at køre med. Fordi det er en simulator, man føler ikke at man bliver rykket fysisk.	Det var en smule kunstigt. Jeg synes ikke bilen helt reagerede på den måde som jeg forventede.
Hvordan fungerede rat, gear og pedaler i forhold til en rigtig bil?	Gearskiftet var svært at vurdere. Pedalerne var udmærket når man lige lærte følelsen af hvor meget man skulle trykke. Rattet fungerede også meget godt.		Den bremsede alt for godt. Det at skifte gear havde jeg ingen følelse af.
Brugte du de to skærme ude i siderne?	Jeg kan huske ved 5. kørsel der kom jeg i tanke om at de var der, så der kiggede jeg lidt, men der var ikke så meget information der gjorde at jeg havde brug for at kigge i dem.	Nej, det er kun den store jeg kigger på. Det kan være sideskærmene har påvirket mig ubevidst.	Ikke til andet end at jeg kunne se hvis der kom et eller andet.
Hvorvidt følte du, at du lærte ruten at kende?	Man lærte det sådan lidt, selvom situationen ændrede sig så føltes det som den samme rute. Jeg kunne huske bygningerne hvor der kom et skarpt sving.	Jeg kunne godt fornemme at noget gik igen, men jeg kunne ikke genkende enkelte steder på ruten.	Jeg kunne mærke med det samme, at det var den samme rute. Jeg lærte den at kende.

Hvordan gik det med at løse opgaverne undervejs? Forstyrrede det din kørsel meget?	Det gik ikke godt når man skulle lave opgaver midt i et sving. Det var svært at oplyse navnene fordi man genkendte cover art på iPod'en og der så stod et andet navn på sangen. Så man skulle nok fjerne cover art så man ikke bliver forudindtaget.	Opgaverne forstyrrede mig rigtig meget. Man kan ikke holde fokus på begge steder. Det var meget svært når man hele tiden skal korrigere.	Ja jeg blev lidt forstyrret en gang i mellem. Men det var godt at jeg kunne vente lidt med at skifte i stedet for at gøre det med det samme.
- Havde du en følelse af stress eller at du var under pres?	Jeg prøvede at udføre opgaven med det samme, da jeg antog at det var det I ville have mig til, siden du tit bad mig gøre det i et sving.	Ja, det gik udover min kørsel, der var flere gange hvor jeg kiggede væk, og så skulle jeg med det samme korrigere når jeg kiggede op igen. Jeg er meget uopmærksom på hvad der foregår når jeg kigger væk.	Ja jeg følte en lille smule stress, du var god til at spørge lige i svingene.
Kunne du fornemme farten, da du kørte uden speedometer?	Jeg kunne kun fornemme farten fordi jeg vidste at jeg brugte 4. gear for at nå 60 km/t da jeg kørte med speedometer, så jeg vidste at hvis jeg kørte i fjerde og ikke gav for meget gad, så kørte jeg nok cirka det samme. Men i forhold til byen var det svært, visuelt at se det.	Det havde jeg meget svært ved. Jeg vidste ikke om jeg kørte 50 eller 100, det kunne jeg ikke vurdere.	Kunne overhovedet ikke mærke farten. Ikke andet end hvis jeg eksempelvis kørte i 4. gear og jeg kunne huske hvordan det lød, når jeg kørte 90 med speedometer, så havde jeg en nogenlunde idé om hvor hurtigt jeg kørte.
Hvornår kunne du bedst lide at køre?	Med bygninger, så kunne man se hvilken vej vejen gik, fordi landskabet var så flat. Trafikken var ligemeget. Man mødte jo ikke rigtig noget i sin egen kørebane.	Uden trafik og uden bygninger.	Uden trafik og uden bygninger.

Forslag til forbedringer?	Det fysiske setup er rigtig godt. Jeg ved ikke om det var muligt at den ventede med at skifte gear til man var koblet ud. Det virkede meget underligt i starten at den gjorde det. Altså når man koblede ud så gik den i neutralt, i forhold til lyden. Så sætter du den i gear, så går kører den op i omdrejninger før du har løftet koblingen. Også eventuelt i sidevinduerne, sætte sidespejlene der så man har en grund til at kigge mere der. Det er sjældent jeg kigger ud af sidevinduerne, men det er tit jeg bruger spejlene, når jeg kører bil.	Måske være noget med grafikken. Det er et stykke bagefter moderne spil som Need For Speed. Rattet kan også forbedres. Få en bedre følelse af når man accelererer. Give en bedre følelse af når man drejer.	Det kunne være dejligt hvis det ikke hakkede. Der var lige nogle gange når der kom fodgængere eller andre biler, så hakkede det en lille smule. Det var generende og man mistede føling. Ellers så ved jeg ikke lige.
----------------------------------	--	---	--

Spørgsmål / Testperson	Testperson4	Testperson5	Testperson6
Hvor gammel er du?	25	25	23
Hvilket semester?	9	9	5
Hvad læser du?	Software	Software	Software
Hvor mange år har du haft kørekort?	7	5	5
Hvor tit kører du bil?	Hver dag		
Har du erfaring med bilsimulatorer?	Ja, bilspil.	Ja, bilspil.	Nej.
Har du erfaring med iPod/iPhone?	Nej.	Nej.	Nej.
Hvad synes du generelt om simulatoren?	Jeg synes den var sjov.		Den virker meget godt.
- Grafikken	Den hakkede lidt. Gerne højere opløsning.	Den var ikke noget at skrive hjem om. En meget flad verden uden bakker og den slags.	
- Lyden		Meget monoton.	Lyden er svær at bruge til at vide hvornår man skal skifte gear.
Hvad synes du om det fysiske setup?	Ja det virkede meget autentisk.	Det var unaturligt. Force feedback virkede meget underligt. Det føltes ikke som at sidde i en rigtig bil.	Det føltes meget godt.

Hvordan var følelsen, føltes det som at køre i en rigtig bil, eller var det kunstigt?	Det føltes rimelig naturligt, bortset fra den måske var lidt svag.	Det var kunstigt.	Det føltes meget godt.
Hvordan fungerede rat, gear og pedaler i forhold til en rigtig bil?	Det var nok det tætteste man kan komme, men jeg syntes der var lidt ratslør. Pedalerne var meget plasticagtige. De flyttede sig meget. Hvis man satte dem fast ville det være bedre. Gearet var okay.	Gearstangen føltes ikke som noget der mindede om en rigtig gearstang. Jeg følte mere at jeg sad i et racerspil end i en rigtig bil.	Rattet fungerede meget godt. Det føltes godt at der var lidt slør i rattet og at den trak lidt i det.
Brugte du de to skærme ude i siderne?	Jeg kiggede på dem for at holde øje med hvor jeg var henne på vejen. Men jeg kiggede mest ud af øjenkrogen.	Nej.	Jeg kiggede kun en gang for at se om tingene også viste sig når man kørte forbi dem.
Hvorvidt følte du, at du lærte ruten at kende?	Ja jeg lærte den at kende, jeg lagde mærke til at det var den samme rute.	Jeg lagde mærke til det var den samme rute. Jeg lærte den til sidst og genkendte nogle af svingene.	Ja jeg lagde mærke til at det var den samme rute jeg kørte.

Hvordan gik det med at løse opgaverne undervejs? Forstyrrede det din kørsel meget?	Det gik fint, det var lidt svært at skrue op og ned en gang imellem. Man skulle lige ramme slideren og passe på at den ikke faldt ned.	Meget forstyrrende.	Jeg prøvede altid at skifte når du sagde det, også selvom det var i sving hvor jeg nok ikke skulle have gjort det.
- Havde du en følelse af stress eller at du var under pres?	Nej.	Jeg følte stress, og løste opgaverne når jeg fik besked på det hver gang i stedet for at vente til et bedre tidspunkt.	Ja jeg følte stress.
Kunne du fornemme farten, da du kørte uden speedometer?	Ja jeg kunne stadigvæk fornemme farten, men om det så var den rigtige fart kan jeg ikke sige.	Overhovedet ikke.	Ja jeg kunne godt følte farten.
Hvornår kunne du bedst lide at køre?	Når der var biler og fodgængere og bygninger, så var der mest naturligt for mig. Det føltes mere autentisk og naturligt.	Når der var biler og bygninger.	Når der var biler og bygninger.

Forslag til forbedringer?	Placeringen af personen i bilen. Man sidder midt i. Det kunne være smart at man sad i den rigtige side, og at man også kunne se de "dårlige" vinkler, blinde vinkler. Og spejle, bakspejl og sidespejl. Og en bedre lyd.	Et bakspejl. Det virkede underligt ikke at have det, det bruger man meget når man kører normalt. Det var underligt at rattet havde racergear når man ikke skulle bruge dem.	Ikke rigtigt, mest med lyden, så man er mere klar over hvornår man skal skifte gear.

Spørgsmål / Testperson	Testperson7
Hvor gammel er du?	23
Hvilket semester?	9
Hvad læser du?	Software
Hvor mange år har du haft kørekort?	5
Hvor tit kører du bil?	
Har du erfaring med bilsimulatorer?	Meget lidt.
Har du erfaring med iPod/iPhone?	Nej.
Hvad synes du generelt om simulatoren?	
- Grafikken	Grafikken hakkede lidt når man kørte med bygninger.
- Lyden	Lyden blev irriterende. Lidt for monoton.
Hvad synes du om det fysiske setup?	Det føltes som at sidde i en bil, tæt på.

Hvordan var følelsen, føltes det som at køre i en rigtig bil, eller var det kunstigt?	Det føltes kunstigt at dreje. Det var svært at føle når man drejede. Der var meget ratslør.
Hvordan fungerede rat, gear og pedaler i forhold til en rigtig bil?	Rattet føltes ikke naturligt.
Brugte du de to skærme ude i siderne?	Næsten ikke.
Hvorvidt følte du, at du lærte ruten at kende?	Jeg lagde mærke til at det var den samme rute, men jeg kunne ikke huske hvornår sving kom og sådan noget.

Hvordan gik det med at løse opgaverne undervejs? Forstyrrede det din kørsel meget?	Jeg følte at jeg blev tvunget til at løse opgaver på tidspunkter jeg normalt ikke ville have gjort det. Havde jeg selv kørt havde jeg ikke skiftet sang i et sving eksempelvis.
- Havde du en følelse af stress eller at du var under pres?	
Kunne du fornemme farten, da du kørte uden speedometer?	Det var ikke nemt at vurdere. Jeg ved ikke om jeg kørte hurtigere eller langsommere end med speedometer. Det var svært når der ikke var speedometer.
Hvornår kunne du bedst lide at køre?	Det var nemmere når der var bygninger for at føle hvor hurtigt jeg kørte. Jeg kørte måske mere forsigtigt når der var mennesker og biler, men jeg ved det ikke.

Forslag til forbedringer?	Hvis det ikke hakkede så meget, og bedre lydside. Der føltes ingen ændringer når jeg trykkede koblingen ned. Der kom ikke lyd fra iPod'en så det forvirrede mig lidt i starten.

A.1.2 Piger

Spørgsmål	Testperson1	Testperson2	Testperson3
Hvor gammel er du?	22	23	22
Hvilket semester?	5	9	5
Hvad læser du?	Erhvervsjura	Informatik	Datalogi
Hvor mange år har du haft kørekort?	3	5	4
Hvor tit kører du bil?		Ikke særlig tit.	Meget sjældent.
Har du erfaring med bilsimulatorer?	Nej	Bilspil, men ikke simulatorer.	Ja, har spillet lidt.
Har du erfaring med iPod/iPhone?	Nej	Nej.	Ikke særlig meget.
Hvad synes du generelt om simulatoren?		Jeg savnede fornemmelsen af at være i en bil.	
- Grafikken		Den var okay, men kunne nok godt være bedre.	Grafikken var ganske flot.
- Lyden	Hjalp med motorlyd.	Lyden var fin nok, jeg kunne høre når jeg skiftede gear.	Motorlyden var lidt skør. Ved ikke om den var realistisk. Syntes den lød meget som om man ikke kørte i det rigtige gear, men måske var det fordi jeg ikke gjorde det.
Hvad synes du om det fysiske setup?		Manglede noget bevægelse i sædet, når man skiftede gear og den slags.	Ja, men det var ikke et rigtigt rat, det kunne man godt mærke.
Hvordan var følelsen, føltes det som at køre i en rigtig bil, eller var det kunstigt?	Føles som at køre en bil, men rattet "hang i det"? Det begyndte at føles lidt naturligt til sidst. Men nu har jeg jo ikke kørt i lang tid.	Jeg skulle lige vænne mig til det.	Jeg syntes det føltes naturligt.

Hvordan fungerede rat, gear og pedaler i forhold til en rigtig bil?	Ikke umiddelbart nogen forskel.	Rattet skulle man lige gennemskue for at vide hvor meget man egentlig skulle dreje nr man kom ind i et sving.	Rattet agerede lidt anderledes end et rigtigt rat. Det var ikke realistisk.
Brugte du de to skærme ude i siderne?	Nej.	Jeg kiggede lidt for at se hvad der foregik, men kiggede næsten ikke.	Overhovedet ikke.
Hvorvidt følte du, at du lærte ruten at kende?	Lagde ikke mærke til, at det var den samme rute.	Jeg genkendte ruten og kunne genkende når vejen var ved at slutte.	Ja, jeg lagde mærke til at nogle af svingene var de samme, men det var ikke sådan at jeg lærte ruten at kende.
Hvordan gik det med at løse opgaverne undervejs? Forstyrrede det din kørsel meget?	Ikke specielt, men lidt i svingene.	Ikke rigtigt, måske lidt. Jeg skulle lige vænne mig til hvordan man skurede op og ned, men jeg lærte det efterhånden.	Der var nok nogle gange hvor jeg skulle have valgt at skifte på et andet tidspunkt.
- Havde du en følelse af stress eller at du var under pres?	Ikke umiddelbart.	Jo i svingene gjorde jeg.	En lille smule stress nogle gange, men det havde nok ikke været det samme hvis man bare skulle gøre det selv når man sad og kørte.
Kunne du fornemme farten, da du kørte uden speedometer?	Nej.	Jeg manglede fornemmelsen af at bilen bevægede sig. Så det kunne jeg ikke.	Kunne overhovedet ikke fornemme farten.
Hvornår kunne du bedst lide at køre?	Uden trafik og uden bygninger.	Når der var bygninger og uden fodgængere.	Uden trafik og uden bygninger.
Forslag til forbedringer?	Nej.	Mere bevægelse i sædet. Ellers synes jeg det var lækkert at bruge.	Ikke umiddelbart, jeg syntes det var meget sjovt.

Spørgsmål	Testperson4	Testperson5	Testperson6
Hvor gammel er du?	20	21	22
Hvilket semester?	3	3	5
Hvad læser du?	Nanoteknologi	Engelsk	Nanoteknologi
Hvor mange år har du haft kørekort?	2	3	4
Hvor tit kører du bil?	En gang om måneden.	Lidt mere end en gang om måneden.	
Har du erfaring med bilsimulatorer?	Lidt da jeg var lille.	Nej.	Nej. Lidt bilspil.
Har du erfaring med iPod/iPhone?	Nej.	Ja.	Meget lidt.
Hvad synes du generelt om simulatoren?			
- Grafikken	Det hakkede meget inde i byen.		Det var svært at se fremad. Det var meget flat.
- Lyden		Lyden virkede ikke så godt.	
Hvad synes du om det fysiske setup?	Det syntes jeg var fint. Man skal altid lige vænne sig til en ny bil.	Jeg syntes gearstangen sad unaturligt.	Gearstangen sad løst og rattet syntes jeg ikke var så godt.
Hvordan var følelsen, føltes det som at køre i en rigtig bil, eller var det kunstigt?	Man kan godt mærke at det er et spil.	Det føltes lidt kunstigt.	Det føltes meget kunstigt.

Hvordan fungerede rat, gear og pedaler i forhold til en rigtig bil?	Det er utrolig nemt at køre her. Der er ikke noget svært med gearstang eller noget som helst.	Jeg rodede meget rundt i gearene, det var svært at føle. Rat og pedaler var meget som en rigtig bil.	Rattet virkede ikke som et rigtigt rat. Det reagerede ikke som i en rigtig bil. Der var meget slør i det.
Brugte du de to skærme ude i siderne?	Jeg brugte dem ikke rigtig, jeg syntes det var forvirrende når man kørte inde i byen at man hele tiden havde noget mørke i øjenkrogene.	Nej.	Meget lidt.
Hvorvidt følte du, at du lærte ruten at kende?	Jeg lagde mærke til det var den samme rute, men jeg var ikke helt sikker. Men svingene lignede meget hinanden.	Jeg lagde mærke til den samme rute og kunne genkende nogle af svingene.	Jeg kunne genkende steder, men det var ikke sådan at jeg kunne huske når de kom. Mest sving.
Hvordan gik det med at løse opgaverne undervejs? Forstyrrede det din kørsel meget?	Ja det havde jeg ikke troet, men det gjorde det meget. Jeg var hele tiden ved at køre ud i rabatten eller en fodgænger ned.	Ja det forstyrrede lidt, jeg kom til at køre ud i siden når jeg skulle udføre opgaverne.	Det forstyrrede ikke mere end hvad man kunne regne med.
- Havde du en følelse af stress eller at du var under pres?	Jeg løste altid opgaverne lige med det samme, jeg skulle nok have ventet lidt. Du stillede mange opgaver lige når jeg var i et sving.	Ja jeg følte mig presset når jeg tænkte på at jeg skulle til at udføre en opgave.	Lidt.
Kunne du fornemme farten, da du kørte uden speedometer?	Jeg havde kørt i 4. gear før, så jeg tænkte nok det var der jeg skulle køre.	Nej.	Ja, lidt.
Hvornår kunne du bedst lide at køre?	Uden bygninger og uden fodgængere. Jeg kunne ikke lide når der var modkørende trafik.	Uden bygninger, men det var ligemeget om der var mennesker og biler eller ej.	Når der var biler og bygninger.
Forslag til forbedringer?	Et bedre billede. Der kunne godt komme flere vibrationer, hvis man kører ind i en bil eller op på fortovskanten.	Måske noget bedre placering af gearene. Det følte ikke som gearstangen på en rigtig bil.	Pedalerne skal sættes bedre fast. Hvis der er noget at lave ved rattet skulle man nok gøre det. Og så grafikken, så man kan se mere fremad.

Spørgsmål	Testperson7	Testperson8
Hvor gammel er du?	21	21
Hvilket semester?	3	3
Hvad læser du?	Nanoteknologi	Psykologi
Hvor mange år har du haft kørekort?	3	2
Hvor tit kører du bil?	En gang om måneden.	Meget sjældent.
Har du erfaring med bilsimulatorer?	Nej.	Nej.
Har du erfaring med iPod/iPhone?	Nej.	Nej.
Hvad synes du generelt om simulatoren?	Jeg synes den gør det ret godt. Jeg følte dog, eksempelvis at det var sådan at man bare kunne køre i et helt forkert gear som man havde lyst til. Der var ikke den helt samme føling mellem gear og fart som i en rigtig bil.	Jeg synes det var vildt fedt. Rigtig godt.
- Grafikken	Det er svært at se fremad hvordan svingene er, i forhold til hvad man kan i virkeligheden, at det er svært at se fremad.	
- Lyden		
Hvad synes du om det fysiske setup?	Jeg synes det er ret godt. Rattet er lidt lille, men ellers er det godt.	Der er lidt med rattet som nok ville give sig lidt mere på en rigtig bil.
Hvordan var følelsen, følte det som at køre i en rigtig bil, eller var det kunstigt?	Det med at man ikke kan se ordentligt frem, og gearene i forhold til bilen.	Det var faktisk rimelig naturligt vil jeg sige.

Hvordan fungerede rat, gear og pedaler i forhold til en rigtig bil?	Gear reagerer ikke ligesom en rigtig bil, og heller ikke rattet. Du kan ikke mærke koblingspunktet. Det føles lidt ligemeget om man lige er i 3. eller 4. gear for eksempel.	Når man har lært det at kende føltes det naturligt nok.
Brugte du de to skærme ude i siderne?	Overhovedet ikke.	Jeg synes man fik et rigtig godt overblik. Selvom man ikke kigger direkte på skærmene så fik man alligevel indtrykket med.
Hvorvidt følte du, at du lærte ruten at kende?	Jeg lagde ikke rigtig mærke til hvilken rækkefølge sving kom i og så videre. Men man lærte den lidt at kende i løbet af øvelsen.	Jeg kunne godt fornemme det var den samme rute men jeg lærte den ikke at kende.
Hvordan gik det med at løse opgaverne undervejs? Forstyrrede det din kørsel meget?	De forstyrrede min kørsel.	De forstyrrede noget. Men ikke ødelæggende.
- Havde du en følelse af stress eller at du var under pres?	Der var et par gange hvor jeg fik stress, det kan man nok også se at jeg var ved at køre galt nogle gange.	Det var lidt når man var i et sving og man skulle til at lave noget, så var det forstyrrende.
Kunne du fornemme farten, da du kørte uden speedometer?	Nej. I hvert fald ikke til at starte med. Men det er igen lidt det med at afstanden er svær at bedømme.	Kunne ikke fornemme farten, det var en meget kunstig følelse bare at trykke på speederen uden at vide det.
Hvornår kunne du bedst lide at køre?	Med bygninger kunne jeg bedre finde ud af hvor hurtigt jeg kørte. Biler og fodgængere gjorde mig ikke så meget, fordi de ikke rigtig forstyrrede i min vejbane.	Når der ikke var biler eller fodgængere, men om der var bygninger eller ej gjorde ingen forskel.
Forslag til forbedringer?	Det med at man skal kunne se bedre fremad. Og bedre gear. Hvis iPod'en skulle simulere at være en radio var det heller ikke så godt, så skulle den have været placeret anderledes.	Der var ikke så meget føling med hvilket gear det var man kørte i.