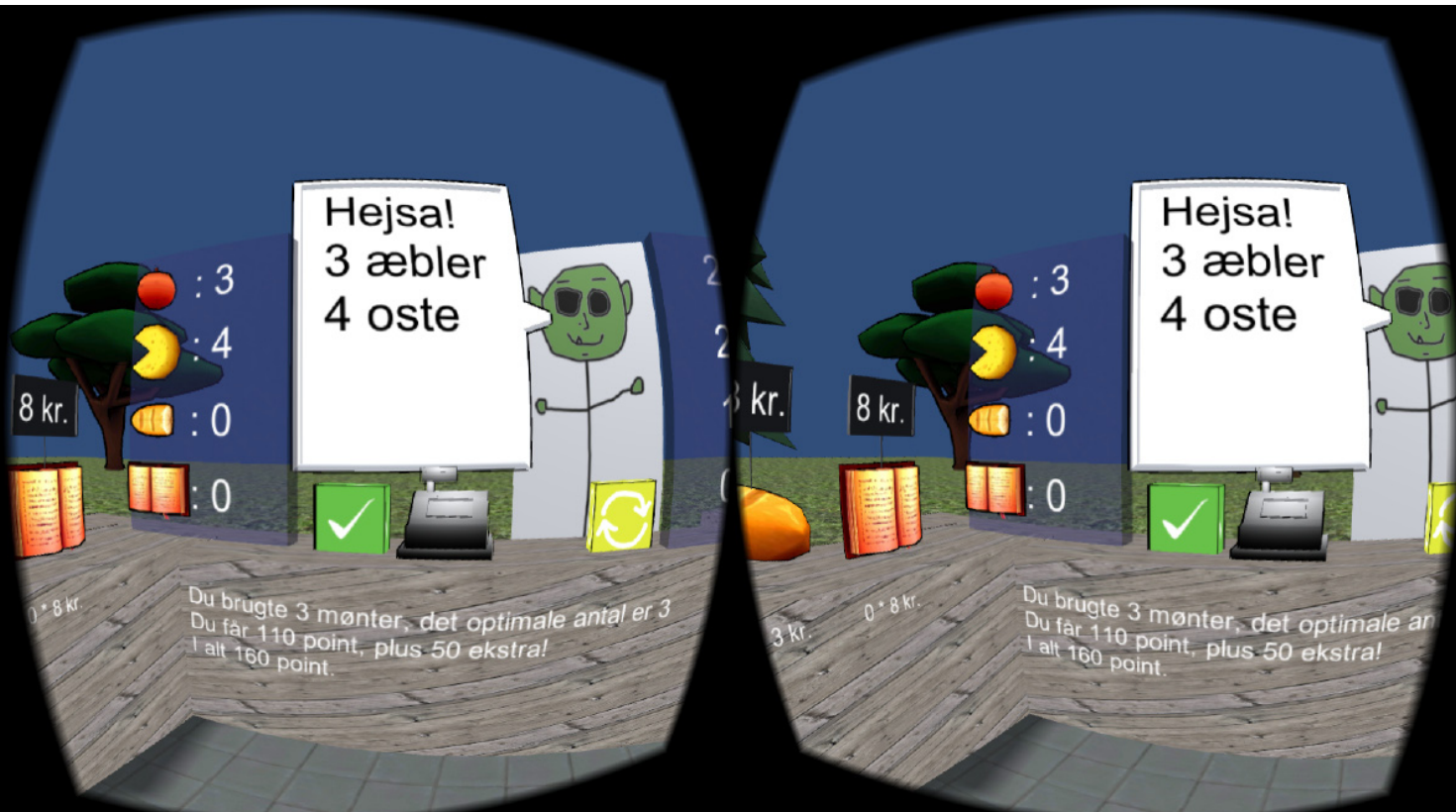


Bachelorprojekt F2016

Udvikling af virtual reality spil til læringsformål

Development of a virtual reality learning game



af Emil Lindgren Ipsen
Lærings- og Oplevelsesteknologi
emips12@student.sdu.dk

Afleveret onsdag den 1. juni 2016

Antal tegn: 64980
27 sider á 2400 tegn

Vejledere: Lasse Juel Larsen
Klaus Holth

Bachelorprojekt (OB-BAP6-U1-1-F16)

Teknisk Fakultet
Mærsk Mc-Kinney Møller Institut
Syddansk Universitet



Emil Lindgren Ipsen

Dato

Formalia

Forord

Denne opgave er skrevet som den afsluttende opgave på bachelordelen af Lærings- og Oplevelsesteknologi studiet. Opgaven omhandler de tanker og handlinger der står bag udviklingen af et virtual reality spil til Google Cardboard (herefter kaldt Cardboard), skabt i Unity. Bachelorprojekt beskrivelsen der indeholder opgavens grundlæggende rammer kan findes i bilag **Bachelorprojekt beskrivelse**.

Opgavens vejledere har været Lasse Juel Larsen (idégenerering og teori) og Klaus Holth (udvikling og planlægning) fra SDU.

Tak til mine vejledere, familie og venner for generelt feedback under opgavens forløb.

Derudover et stort tak til Google og Unity for gode produkter med en veludført dokumentation, og tak til brugerne af Stack Overflow, der i de fleste tilfælde allerede havde svaret på mine spørgsmål.

Alle links i opgaven, både i bilag og fodnoter, var tilgængelige den 1. juni 2016.

En stor tak til alle der har testet prototyperne!

Filerne kan findes på følgende webadresser (minimum indtil primo juni 2017) og på det tilhørende USB stik (1 stk udleveret med opgaven).

Android version

https://www.dropbox.com/s/7kfm114vy3crpba/Emils_BA_V0-5b-final.apk?dl=0

PC Version

https://www.dropbox.com/s/x8wx75gt5vg3nx/EmilsBA_PCPort_V0-5b-Final.rar?dl=0

Unity project

<https://www.dropbox.com/s/hgngasatvad9h8n/UnityFiler.rar?dl=0>

Abstract (dansk)

Formålet med denne opgave er at dokumentere udviklingsforløbet af et virtual reality læringsspil. Udviklingsforløbet bygger på en kombination af teori (hovedsageligt af James Paul Gee og Keith Devlin), analyse af målgruppen og mindre brugertests.

Resultatet af udviklingsforløbet er en prototype, der skal støtte læringen af matematik i 1.-3. klasse. Dette gøres ved at placere brugeren i en bod, hvori de skal beregne den samlede pris på et antal varer. En analyse af tilgængelige teknologier, og perspektivering til hvordan de kunne understøtte prototypen, er også gennemgået i opgaven. En perspektivering hvordan prototypen kunne udvikles videre, er også at finde i perspektiveringsafsnittet.

Opgaven er et bachelorprojekt, skrevet af Emil Lindgren Ipsen, studerende på Syddansk Universitets Tekniske Fakultet, Lærings- og Oplevelsesteknologi, afleveret den 1. juni 2016.

Abstract (english)

The purpose of this paper is to document the process of developing a virtual reality learning game. The development process is based on a combination of theory (mainly by James Paul Gee and Keith Devlin), analysis of the target audience and a few user tests.

The outcome of the development process is a prototype designed to support the learning of mathematics in the 1st-3rd grade. This is achieved by placing the user in a booth in which they must calculate the total price on a number of items. An analysis of available technologies and perspectives to how they could support the prototype, are also discussed in the paper. Thoughts on how the prototype could be developed further, can also be found in perspectivation section.

The paper is a bachelor thesis, written by Emil Lindgren Ipsen, student at the University of Southern Denmark's Faculty of Engineering, Learning- and Experience Technology, turned in 1st of June, 2016.

Indholdsfortegnelse

Formalia.....	1
Forord.....	1
Abstract (dansk)	2
Abstract (english).....	2
Indholdsfortegnelse.....	3
Indledning.....	5
Problemformulering.....	5
Planlægning.....	6
Tekniske forudsætninger.....	6
Læringsmål i undervisningen.....	7
Natur/Teknologi	7
Matematik	7
Valg af målgruppe og idégenereringsprocess	8
Idégenereringsprocessen	8
Caseidéer og målgruppe.....	9
Delkonklusion.....	12
Teori.....	14
Virtual Reality.....	14
Brugerens indlevelse.....	14
Immersion	14
Hypermediacy/Immediacy	15
Flow	15
James Paul Gee	16
Keith Devlin.....	17
Street Smarts.....	18
Perfect Medium	19
Key Features of Gaming	21
Udvikling	23
Tekniske aspekter ved Google Cardboard	23
Teknologianalyse af lignende teknologier	25
Samsungs Gear VR	25

Oculus Rift.....	25
HTC Vive.....	27
Google Daydream	28
Udvikling generelt	28
Første prototype	29
Første iteration	32
Brugertest.....	32
Udvikling	33
Anden iteration.....	35
Brugertest.....	35
Udvikling	36
Endelige prototype	38
Prototypens software.....	39
Struktur i Unity.....	39
Gennemgang af kodestykker	39
Afrunding	41
Konklusion.....	41
Perspektivering	41
Teknologien	41
Prototypen.....	42
Anbefalinger	44
Appendix	45
Litteraturliste.....	45
Bilag.....	45
Bachelorprojekt beskrivelse.....	45
Brugertests	46
Kode	49
Anerkendelser - billeder og figurer	58
Anerkendelser - materiale til software	58

The fact that all of this was happening in virtual space made no difference.
Being virtually killed by virtual laser in virtual space is just as effective as the
real thing, because you are as dead as you think you are.

(Adams, 1992)

Indledning

Bachelorprojektets tema var *Udvikling af virtual reality spil til læringsformål*¹, og der er derfor i denne skriftlige aflevering redegjort for planlægningsfasen, brugte og relevante teorier, og udviklingsfasen. Til sidst afrundes den med en reflekterende konklusion og en perspektivering.

Der er taget udgangspunkt i teorier gennemgået i uddannelsen, teoretikere foreslået af vejlederne, og relevant materiale, der er blevet fundet og undersøgt i løbet af projektets forløb.

Problemformulering

Denne opgave vil undersøge, hvordan man kan udvikle et virtual reality spil til læringsformål, indenfor det naturvidenskabelige- og/eller matematiske læringsområde - og samtidig forsøge at skabe en prototype der afspejler resultaterne. Dette vil blive gjort ved hjælp af en analyse af de *Fælles Mål*², og ved at undersøge teori relevant for emnet. Dette skal danne grundlag for en prototype, der kan gennemgå mindre brugertests (af brugere i målgruppen, og relevante fagfolk).

¹ Projektbeskrivelsen kan findes i bilaget **Bachelorprojekt beskrivelse**

² <http://www.emu.dk/modul/introduktion-til-forenklede-f%C3%A6lles-m%C3%A5l>

Planlægning

I dette afsnit vil projektets planlægningsproces blive gennemgået. Først vil de tekniske forudsætninger blive gennemgået, derefter vil der blive lavet en oversigt over folkeskolens Fælles Mål, der vil blive brugt som referencepunkter til afsnittet, der redegør for idégenereringsprocessen og bestemmelse af målgruppen.

Tekniske forudsætninger

De tekniske forudsætninger for produktet blev specificeret i bachelorprojekt beskrivelsen. Projektets prototype produkt, herefter refereret til som *produktet*, skulle udvikles til at fungere på en tidssvarende virtual reality (VR) platform³, og ligeledes være udviklet i et tidssvarende relevant udviklingsmiljø⁴. Produktets formål var at være et læringsspil, der minimum skulle håndtere en enkelt case inden for det matematiske/naturvidenskabelige felt. En case i dette sammenhæng består af en form for interaktiv udfordring, som produktets bruger kan løse.

Valget af VR-platform faldt på Cardboard⁵, da det var en prisvenlig og lettilgængelig platform, der kan køre på de fleste mobiltelefoner - en længere forklaring på dette er beskrevet afsnittene **Tekniske aspekter ved Google Cardboard** og **Teknologianalyse af lignende teknologier**, under **Udvikling**.



Billede 1: Google Cardboard

³ F.eks. Google Cardboard eller Oculus Rift

⁴ F.eks. Unity eller Unreal Engine

⁵ <https://vr.google.com/cardboard/index.html>

Valget af udviklingsplatform faldt på Unity⁶ - både grundet et tidligere kendskab til platformen, men også platformens versatile natur, god dokumentation for Cardboard integration, mulighed for eksportering af software til flere forskellige platforme⁷ og den gratis *Personal Edition* der giver adgang til alle platformens grundfunktioner.

Læringsmål i undervisningen

For at finde en relevant case inden for det matematiske/naturvidenskabelige felt, kan man kigge på de Fælles Mål⁸ som Ministeriet for Børn, Undervisning og Ligestilling, samt en række eksperter, har opsat som en grundlæggende basis for hvad eleverne skal lære inden for de forskellige læringsområder, og i hvornår de cirka skal lære det. Hvert fagområde består af fire kompetenceområder, der under hvert klassetrin består af en yderlig inddeling af kompetenceområdet i faser. Herudover er der også eksempler på undervisningsforløb og faglig inspiration/læringsaktiviteter.

I dette afsnit skabes et overblik over områderne Natur/Teknologi og Matematik - dette gøres for at skabe en let overskuelig referenceramme for idégenerering og bestemmelse af målgruppe.

Natur/Teknologi

Natur/Teknologi er inddelt i 1.-2. klasse, 3.-4. klasse og 5.-6. klasse - derefter bliver faget delt ud i tre andre fag; Fysik/kemi, Geografi og Biologi. I denne analyse bliver der dog kun kigget på Natur/Teknologi læringsområdet.

De fire kompetenceområder, fordelt på alle klassetrin, er: **Undersøgelse**, at eleven skal kunne udføre/gennemføre/designe undersøgelser; **Modellering**, at eleven skal kunne anvende og designe modeller; **Perspektivering**, at eleven skal kunne forstå natur og teknologi både i sin egen hverdag, i forhold til andre kontekster og i omverdenen generelt; **Kommunikation**, at eleven skal kunne beskrive deres opdagelser og undersøgelser, teknologiske problemer og om natur og teknologi generelt.

Matematik

Matematik er inddelt i 1.-3. klasse, 4.-6. klasse, 7.-9. klasse og 10. klasse.

De fire kompetenceområder, fordelt på alle klassetrin, er: **Matematiske kompetencer**, at eleven skal kunne handle hensigtsmæssigt med overblik i situationer med matematik; **Tal og algebra**, at eleven skal udvikle metoder til beregning af tal, og benyttelsen af reelle tal og algebraiske udtryk; **Geometri og måling**, at eleven skal kunne anvende og forstå geometriske

⁶ <https://unity3d.com/>

⁷ Herunder PC, Android og iPhone

⁸ <https://www.uvm.dk/Uddannelser/Folkeskolen/Laering-og-laeringsmiljoe/Laeringsmaalstyret-undervisning/Undervisning-med-udgangspunkt-i-laeringsmaal>

begreber, og kunne anvende dem i omverdenen; **Statistik og sandsynlighed**, at eleven skal kunne udføre og vurdere statistik og sandsynlighed.

Valg af målgruppe og idégenereringsprocess

Ud fra de tekniske forudsætninger og et overblik over de Fælles Mål, blev det muligt at starte idégenereringsprocessen og dermed også komme tættere på at finde en målgruppe at udvikle et produkt til.

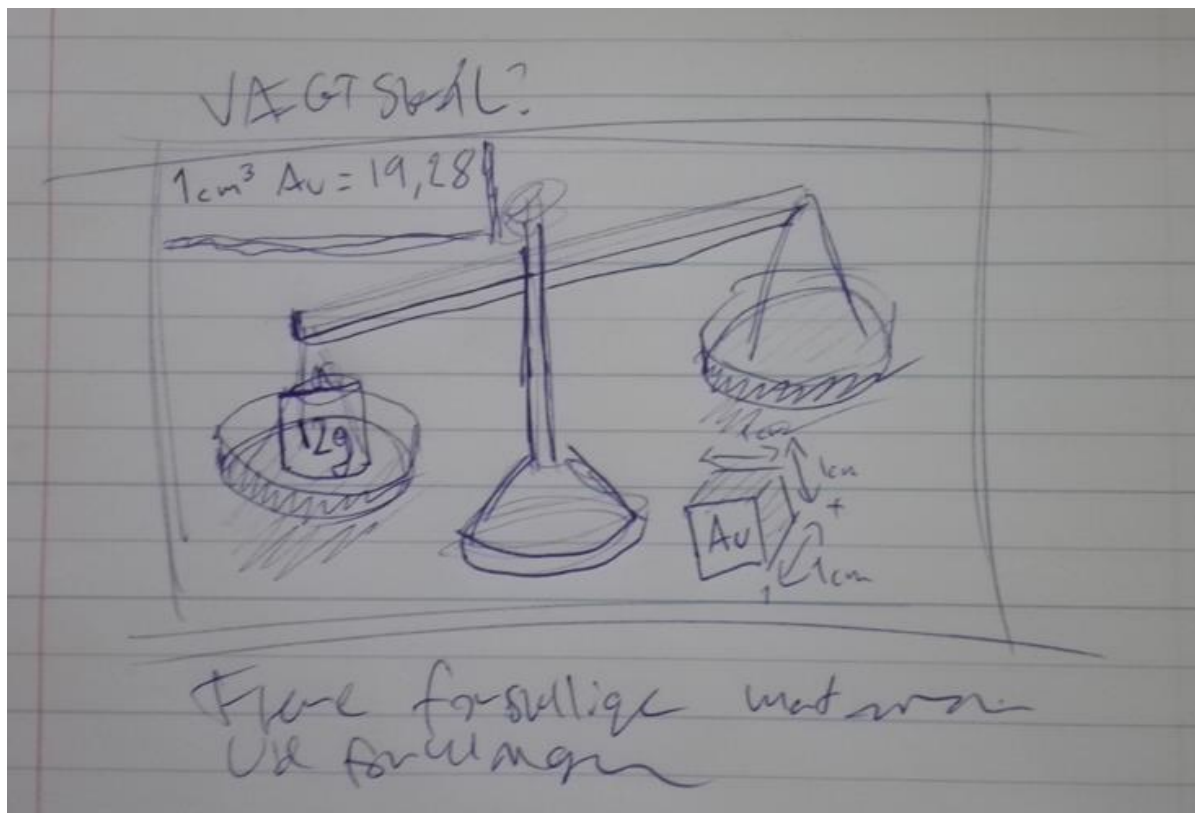
I starten af idégenereringsprocessen var den overordnede idé at lave en læringsspilsplatform der kunne håndtere flere mindre cases - produktet ville dermed kunne henvende sig til flere forskellige målgrupper inden for både matematik og natur/teknologi. Grundet tidsbegrænsninger indeholder prototypen kun én enkelt case - flere case koncepter er dog blevet gennemgået inden den endelige case blev valgt.

Idégenereringsprocessen

Den indledende idégenereringsprocess startede allerede ved det første vejledermøde, hvor de første idéer begyndte at tage form, og efterfølgende er der blevet lavet mindre mindmaps over disse idéer og de Fælles Mål, der har resulteret i fire, mere eller mindre, konkrete cases.

Caseidéer og målgruppe

Case 1: Vægtskål



Billede 2: Skitse af vægtskåls case

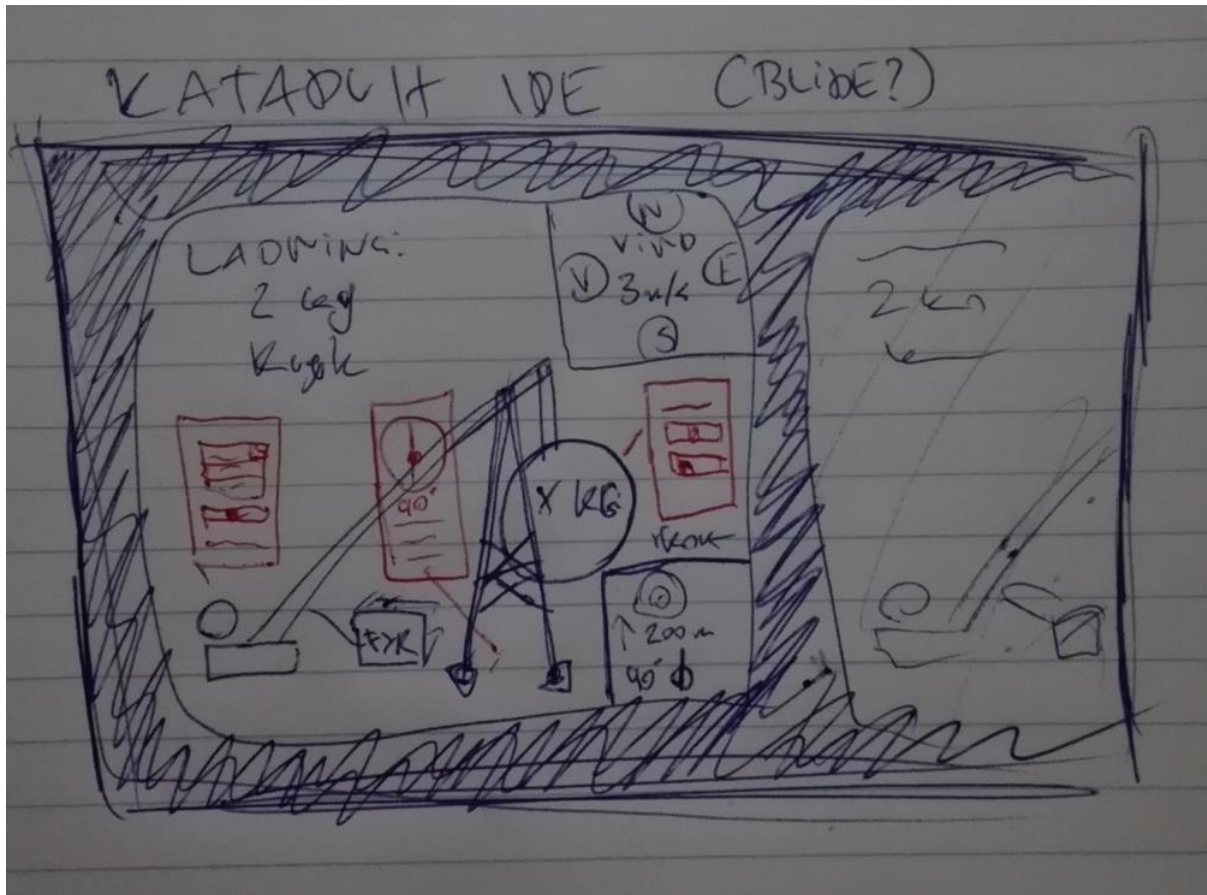
Denne case ville hovedsageligt forholde sig til *Geometri og måling (matematik 1.-6. klasse)* og vidensområdet *Teknologi og ressourcer (natur/teknologi 3.-4. klasse)*. Idéen gik ud på at brugeren skulle kunne forstå forholdet mellem vægt og massefylde af forskellige materialer.

En vægtskål (se billede 2) skulle være hovedfokus, og brugeren skulle have mulighed for at afstemme vægten af en række materialer, mod en række faste vægte. Dette kunne både gøres ved at der på vægten allerede var en række materialetyper med størrelsesforhold tilknyttet, og brugeren så skulle afstemme dette ved hjælp af en række tilpassede vægte, men også omvendt, så brugeren skulle placere en vis mængde materialer for at afstemme vægte allerede placeret på vægten.

En sandbox model kunne også være tilgængelig, så brugeren selv kunne opstille sine egne undersøgelser. Den ideelle målgruppe ville være 3. og 4. klasse, da de her har fået lidt forståelse for geometri og størrelsesforhold i matematik, og i natur/teknologi har haft fokus på materialekendskab.

Case 2: Katapult (blide)

Denne case ville hovedsageligt forholde sig til vidensområdet *Teknologi og ressourcer* (natur/teknologi 5.-6. klasse) og matematik 4.-6. klasse generelt. Idéen gik ud på at brugere skulle forholde sig til de matematiske og fysiske aspekter der er i brugen af en blide-katapult.



Billede 3: Skitse af blide case

En blide skulle være tilgængelig for brugeren, samtidig med en række forskellige former for ammunition med forskellige kvaliteter, såsom vægt, størrelse, vindmodstand. Brugeren skulle så have mulighed for at indstille de forskellige aspekter på bliden, heriblandt rotation, vægt og længde. De skulle så gå efter at ramme et område på en bane, og få point alt efter hvor ammunitionen så landede. Igen kunne en sandbox model give mening, da brugerne så havde mulighed for at lege med de forskellige aspekter. En multiplayer version, hvor det gælder om at nedlægge en anden spillers tårn, ville også kunne skabe et incitament for at tilpasse værdierne til at være korrekte.

Den ideelle målgruppe ville være drenge fra 5.-6. klasse, da de så ville kunne bruge viden fra både natur/teknologi og matematik, til at forstå den bagvedliggende fysik bag bliden.

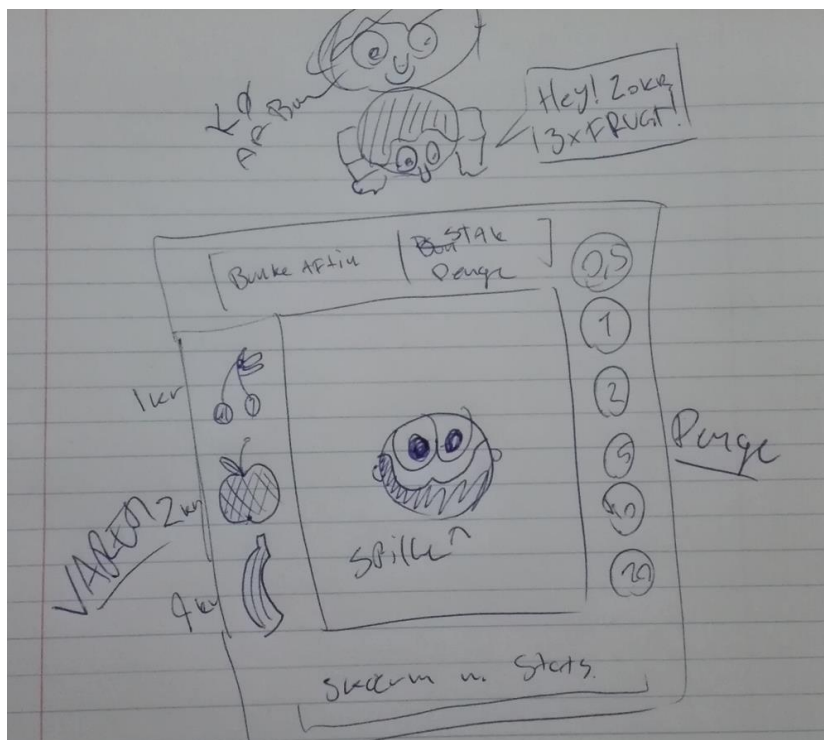
Case 3: Tredimensionelle vektorer

Denne case ville, til forskel for de andre cases, forholde sig til mere avanceret matematik (9.-10. klasse og ungdomsuddannelsesniveau). Idéen gik ud på at brugeren skulle kunne arbejde med tredimensionelle vektorer i et tredimensionelt rum, hvilket kunne hjælpe brugeren med at visualisere problemstillinger, der er svære at illustrere todimensionelt.

Denne case var mere tænkt som et virtuelt miljø, der var tilpasset til at illustrere tredimensionelle vektorer, frem for et decideret spil. Idéen var dog at brugeren skulle kunne modtage problemstillinger de så kunne løse ved hjælp af teknologien, og dermed skabe sig en visuel forståelse for denne type matematik. Den ideelle målgruppe ville være elever fra de højeste klassetrin i folkeskolen, og elever på ungdomsuddannelsesinstitutioner.

Case 4: Bod

Denne case ville udelukkende forholde sig til de tre matematiske (1.-3. klasse) vidensområder : *matematiske kompetencer, tal og algebra og statistik og sandsynlighed*. Idéen gik ud på at brugeren skulle stå i en bod, og servicere kunder.



Billede 4: Skitse af bod

Dette skulle gøres ved at en kunde kom til butikken og bad om et sæt varer - brugerne skulle så vælge det ønskede antal varer (der hver især havde en oplyst værdi), og derefter vælge en række mønter (med samme værdier som de danske mønter) der svarer til prisen på varerne. Brugeren skulle så godkende købet, og alt efter hvor tæt det valgte beløb var på varernes beløb, bliver brugeren belønnet med point.

I tilfælde af at beløbene var meget forskellige, ville handlen ikke blive gennemført, og brugeren ville blive informeret - hvis der kun var en mindre forskel på beløbene skulle handlen gå igennem, men brugeren ville modtage en mindre straf, enten på kundetilfredsheden eller i bodens status (alt afhængigt og beløbet var større eller mindre end varernes værdi).

Den ideelle målgruppe for denne case ville være elever fra 1.-3. klasse, da de derved kunne op- og viderebygge på deres grundlæggende matematiske forståelse, både i forhold til økonomisk forståelse (handel), statistik (afvigelse fra den ideelle handel), og regnestykker (både i form af objekter, og matematiske udtryk).

Samtidig bygger denne case også på en undersøgelse af sælgerbørn fra Sydamerika, der har nemt ved at regne prisen på en handel ud, men har sværere ved at løse de samme beregninger på papir. Mere om dette kan læses i teoriafsnittet "Street smarts - Keith Devlin".

Nævneværdige idéer

Andre idéer, der ikke blev undersøgt nærmere i forhold til de Fælles Mål og målgrupper, var: rumudforskning, for at hjælpe elever med at forstå solsystemet; biologi, både i forhold til menneskets indre anatomi, mikrobiologi (celler, DNA, etc.), og økosystemer; visualisering af fysik.

Delkonklusion

For at tage en beslutning om hvilken case der ville være mest hensigtsmæssigt at give sig i kast med, var det nødvendigt at overveje hvor meget udviklingsarbejde der lå bag de forskellige cases, og hvilke cases der ville være hensigtsmæssige at arbejde med i forhold til både vidensområder, og lærings- og spilteori.

I tabel 1 kan en oversigt over det forventede udviklingsarbejde og relevante teori ses - alle fire cases skal håndtere en form for valg af objekter, *juiciness*-spilteori, og generel læringsspilsteori.

Udviklingsarbejde		Teori
Case 1 Vægtskål	Meget afhængig af Unitys <i>physics engine</i>	Korrekt viden omkring materialer
Case 2 Blide	Meget afhængig af Unitys <i>physics engine</i>	Korrekt viden omkring blide-fysik
Case 3 Vektor	Kodning af tredimensionelt gitter Håndtering af vektorer	Læringsteori til ældre elever Udfordrende vektor opgaver
Case 4 Handel	Matematiske formler Sammenligning af værdier	" <i>Street smarts</i> "-teori

Tabel 1: Case oversigt

Efter nøje overvejelser faldt valget på case 4. En prototype ville være nem at udvikle, da repræsentation af elementer ville være let at opstille i Unitys brugervenlige interface, og der var relevant teori tilgængeligt til at understøtte casen.

Case 1 og 2 kræver meget korrekt viden om de tilhørende vidensområder, og en nøje tilpasning af elementerne i Unity ville også være nødvendig for ikke at skabe et urealistisk billede af de modeller som produktet skulle visualisere.

En gennemarbejdet implementering af case 3 ville, ud over at præsentere og godkende udfordrende vektor opgaver, kræve et dynamisk tredimensionelt gitter, som brugeren skulle kunne interagere frit med - dette ville for det første kræve et meget omfattende udviklingsarbejde, og navigeringen i det virtuelle miljø ville ikke være optimal med Cardboards begrænsede input muligheder.

Resten af denne rapport, indtil afsnittet **Afrunding**, omhandler derfor teori relevant for case 4, og en gennemgang af den udviklingsprocess der førte til den endelige prototype. Case 4 vil herefter blive refereret til som "casen".

Teori

I dette afsnit vil teori relevant for casen blive gennemgået, samtidig med at der vil blive draget linjer til, hvordan teorien kan implementeres i produktet - disse linjer skal ses som forslag, og betyder ikke at det omtalte teori vil blive implementeret i prototypen.

Først vil Virtual Reality som medieteknologi blive gennemgået, hvorefter der vil komme et afsnit med korte gennemgange af teorier og udtryk relevant for brugerens indlevelse. Derefter vil der blive kigget på generel læringsspilsteori af James Paul Gee, hvorefter teori af Keith Devlin, der ser på læringsspil ud fra et matematisk perspektiv, vil blive undersøgt - der vil blandt andet blive reflekteret over hans gennemgang af "Street Smart"-studiet og gennemgået et par lister.

Virtual Reality

Virtual reality, på dansk virtuel virkelighed, dækker over teknologi der, ved hjælp af computerkraft, simulerer en verden og brugerens interaktion deri. I det sidste århundrede har idéen om virtual reality udviklet sig, og inden for de seneste år er teknologien skudt i vejret. En gennemgang af relevante teknologier kan findes i afsnittene **Tekniske aspekter ved Google Cardboard** og **Teknologianalyse af lignende teknologier**.

Brugerens indlevelse

Immersion

Immersion, *fordybelse* på dansk, er et begreb der bruges til at beskrive brugerens involvering i en aktivitet. Immersion kan, ifølge Dourish (2001), beskrives med tre underbegreber, der, i en stigende grad, hver især beskriver brugerens følelsesmæssige tilknytning til aktiviteten.

Engagement repræsenterer brugerens lyst til at benytte sin tid og energi til at udføre en aktivitet. Det indebærer kun en overfladisk tilknytning til aktiviteten, hovedsageligt kun fysisk.

Engrossment, *optagethed* på dansk, dækker over en tilstand hvor spillet direkte påvirker brugerens følelser. Det indebærer mere end bare den fysiske tilknytning.

Total Immersion, *fuldkommen fordybelse* på dansk, beskriver en tilstand hvor brugerens indlevelse i aktiviteten er så stor, at det er det eneste brugeren forholder sig til - der kan til dette drages paralleller til Mihaly Csikszentmihalyis teori omkring *flow*.

Hypermediacy/Immediacy

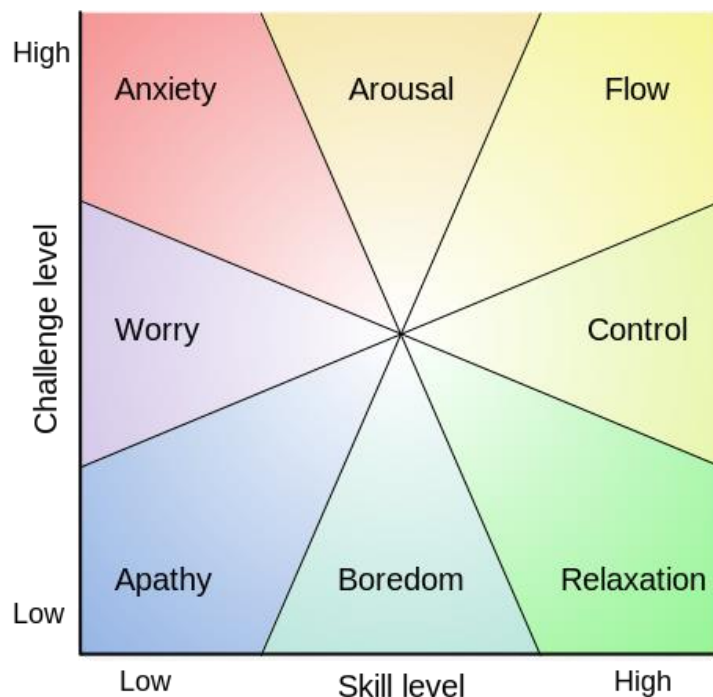
Hypermediacy og *immediacy* (Bolter og Grusin, 2000) er to modsatte ender af spektrum, der beskriver hvor transparent et medie er - ved transparens menes hvor meget brugeren er opmærksom på at der bruges et medie.

Hypermediacy repræsenterer den ende af spektret hvor mediet er tydeliggjort, hvor **immediacy** derimod repræsenterer det transparente medie, altså hvor brugeren ikke er opmærksom på mediet.

En høj grad af immediacy gør det derfor lettere for brugeren at nå et dybere niveau af immersion.

Flow

Begrebet *flow* blev opfundet af den ungarske psykolog *Mihaly Csikszentmihalyi* (1991). Flow er en tilstand der forekommer når en bruger er engageret i en aktivitet, der er tilpas udfordrende, i forhold til brugerens evner inden for den pågældende aktivitet. På figur 1 ses en oversigt over de forskellige stadier en bruger bevæger sig i, afhængigt af evne (x-aksen) og udfordring (y-aksen) - flow kan ses øverst til højre, hvor både evne- og udfordring niveauet er højt.



Figur 1: Oversigt over stadier

Flow repræsenterer en tilstand hvori brugeren er opslugt af en aktivitet, hvilket gør at brugeren oplever at "tiden flyver", samtidig med at aktiviteten skaber en stor tilfredshedsfølelse. At skabe et produkt der fremkalder flow hos brugeren, kan være et stort skridt på vejen til at skabe total immersion.

James Paul Gee

James Paul Gee er en amerikansk forsker, med hovedområder inden for psykolingvistik, læsefærdigheder, tosproget undervisning, og computerspil og læring. Netop hans teori angående computerspil og læring er relevant for dette projekt.

I artikelen *Good Video Games, the Human Mind, and Good Learning* (Gee, 2013), snakker Gee om hvordan videospil, det menneskelige sind, og læring spiller sammen. Den er delt ind i to dele - den første er en snak om hvordan spil kaster lys på, og afspejler, det menneskelige sind, og den anden del er en opstillet liste, der indeholder principper, der er med til at analysere hvor meget potentiale et spil har, i forhold til læring. En oversigt over listen, med korte forklaringer af de forskellige principper, kan ses herunder.

I. Empowered Learners

- » **Co-design:** brugerne er aktive agenter, deres handlinger har betydning. De er ikke bare passive modtagere af information.
- » **Customize:** materiale bør tilpasses til brugernes læringsstile, men stadig opfordre til at prøve andre stile.
- » **Identity:** indlevelse i en aktivitet, som en påtaget identitet, kan hjælpe brugeren med at engagere sig, og derved skabe bedre læring.
- » **Manipulated and Distributed Knowledge:** manipulering af objekter og viden kan hjælpe brugeren med at føle at deres krop og sind strækker over et større område.

II. Problem Solving

- » **Well-Ordered Problems:** ved at skabe udfordringer i en tilpasset rækkefølge, kan brugeren forberedes til sværere udfordringer.
- » **Pleasantly Frustrating:** læringspotentialet er højest, når man har løsbare, men udfordrende, opgaver - se "flow".
- » **Cycles of Expertise:** ekspertise skabes ved at brugeren lærer til de mestrer en færdighed, hvorefter færdigheden bliver udfordret. Brugeren skal så tænke og lære igen - dette gentages.
- » **Information "On Demand" and "Just in Time":** i stedet for at præsentere brugeren med en masse information ude af kontekst, kan det hjælpe at præsentere et stykke information når der er brug for det, eller når brugeren selv mener at de har brug for det.
- » **Fish Tanks:** virtuelle miljøer der simplificere dele af en aktivitet, hvilket gør det lettere for brugere at forholde sig til det, i mere komplekse sammenhænge.

- » **Sandboxes:** virtuelle miljøer hvor brugerne kan eksperimentere og udforske uden store konsekvenser.
- » **Skills as Strategies:** det er nemmere at få brugere til at øve en færdighed, hvis de kan se færdigheden som et middel til at nå et mål.

III. Understanding

- » **System Thinking:** hvis brugeren kan se det store perspektiv i systemet, er det nemmere at lære de mindre perspektiver.
- » **Meaning as Action Image:** god forståelse kræver at brugeren kan reflektere over oplevelser og erfaring.

I **Empowered Learners** kan man i casen især gøre brug af begreberne: **Identity**, ved at skabe en handelsmand-identitet til brugeren, og **Manipulated and Distributed**, ved give brugeren evner til at interagere med objekter i en virtuel verden.

I **Problem Solving** kan gøre brug af begreberne: **Well-Ordered Problems**, ved at have en række lette problemstillinger klar til brugeren i starten af spillet; **Pleasantly Frustrating**, ved at tilpasse sværhedsgraden alt efter brugerens præsentation, og **Cycles of Expertise**, ved at lade sværhedsgraden stige, og eventuelt præsentere brugeren for en form for test/boss, inden det næste niveau nås.

I **Understanding** er det mest brugbare begreb **Meaning as Action Image**, da brugeren er nødt til at have en vis forståelse for matematik og økonomi, for at forstå casen.

Keith Devlin

Keith Devlin er en britisk matematiker, der udover at specialisere sig i matematik, også har skrevet en bog der omhandler matematik og læringsspil (Devlin, 2011). Bogen titel er *Mathematics Education for a New Era*, og har undertitlen *Video Games as a Medium for Learning*.

I dette afsnit vil en håndfuld af disse kapitler blive opsummeret. Der vil hovedsageligt blive lagt fokus på kapitlerne om udvikling af læringsspil, frem for kapitlerne om hvordan det kan påvirke skolesystemet - både fordi hovedfokus på dette projekt er udviklingen, men også fordi at Devlin hovedsageligt har fokus på det amerikanske skolesystem (der givetvis har en del fællestræk med det danske, på mange af hans kritikpunkter).

Street Smarts

Street Smarts er kapitel 2 af bogen, og omhandler Devlins gennemgang og refleksion over et studie af unge gadesælgere i Recife, Brasilien - studiet blev lavet tre forskere i 1993⁹.

Gennemgangen starter med to eksempler på transaktioner mellem forskerne og sælgerne. I den første transaktion beder forskeren om 10 kokosnødder af en værdi på 35. Sælgeren regner derefter prisen ud med følgende udsagn: “*Three will be 105; with three more, that wil be 210. (Pause) I need four more. That is... (Pause) 315 ... I think it is 350*” (Devlin, 2011, s. 7). Den anden transaktion gør ud på at regne prisen ud på et køb, hvor sælgeren skal trække 75 fra 243, hvilket gøres med følgende udsagn: “*You just give me two hundred. I’ll give you twenty-five back. Plus the fort-three that you have, the hundred and forty-three, that’s one hundred and sixty-eight*” (Devlin, 2011, s. 7). Devlin forsøger så at redegøre for sælgernes udregninger, der afviger fra de udregninger en person med kendskab til matematik på folkeskoleniveau ville gøre sig,

Den første sælger kunne, med kendskab til matematik, bare have ganget med 10 (altså tilføjet et 0). Sælgeren regner dog prisen ud ved at bruge de værdier han normalt bruger (som nok er to eller tre kokosnødder af gangen), og lægger så værdierne sammen,

til det svarer til værdien af ti kokosnødder.

Den anden sælger, som Devlin dog forklarer sagde forkert (han sagde 200 i stedet for 100). Han deler så regnestykket op i mindre stykker - så han ender med at regne udregningen: $100 + 43 + (100 - 75) = 168$.

Efter forskerne havde “testet” en række af sælgere på gaden, udførte de en række stille og rolige test på dem, hvor de fik en række spørgsmål præsenteret der ligner de regnestykker de udfører på markedet. Resultatet viste sælgerne på markedet havde en præcision på over 98%, men med mundtlige opgaver faldt den til 74%, og når opgaverne kom ned på papir, som regnestykker, faldt præcisionen helt ned til 37%.

Et lignende studie¹⁰ blev foretaget på voksne i californiske supermarkeder - en test viste at de voksne havde en præcision på 98% i supermarkedet, men kun 59% præcision på en decideret test (med samme type beregninger).

Disse test viser en tydelig korrelation mellem resultaterne når brugeren står over for matematisk udfordring i virkeligheden, frem for når de bliver præsenteret for de samme udfordringer som tests. Devlin mener, at dette er fordi de testede, i stedet for at tilgå

⁹ Nunes, T., Schliemann, A. og Carraher, D. (1993). *Street Mathematics and School Mathematics*, Cambridge University Press, Cambridge, UK

¹⁰ Lave J. (1988). *Cognition in Practice: Mind, Mathematics, and Culture in Everyday Life*. Cambridge University Press, Cambridge, UK

udfordringen som de ville gøre hvis de mødte det i virkeligheden, forsøger at hidkalde matematiske procedurer de (måske ikke) har lært i skolen.

Casen bygger løst på disse studier, da den sætter brugeren i en situation, hvori de skal agere som en sælger, der skal udføre en handel korrekt. Ud over at blive præsenteret for en oversigt over de valgte varer, kan brugerne også se hvordan transaktionen som regnestykke vil se ud. Hypotesen ved dette, er at brugeren skal have mulighed for at sætte en sammenhæng mellem varerne, og den matematiske notation af et opdelt multiplikation- og additions stykke.

Perfect Medium

The Perfect Medium er kapitel 3 af bogen, og forsøger at forklare hvorfor Devlin mener at computerspil er det perfekte medium for matematisk læring.

I første del redegøres der for hvordan vi, som mennesker, gør brug af matematik. En kort opsummering af denne del, er at vi ikke er lavet til at forstå matematik som tegn, men at det er en evne der må læres - og indtil starten af 1300-tallet var matematik ikke noget man bare lige lærte; det var en profession der blev lært via mesterlære. Det var først da Leonardo af Pisa, også kendt som Fibonacci, i 1202 fik færdiggjort en bog kaldet *Liber abbaci*, hvor i symbolsk matematik blev forklaret, at folk havde mulighed for at lære matematik ved at læse - de samme metoder bliver stadigvæk brugt den dag idag, til at lære folk matematik.

Devlin skriver at vi har brugt symbolsk matematik til at håndtere og distribuere matematik længe, af den grund, at det var den mest hensigtsmæssige måde at gøre det på - men spørger så, hvordan man ville designe en alternativ metode at distribuere det på? Han henviser her til computerspil, og stiller derefter en liste af 11 principper et ideelt læringsmiljø ville indeholde. En oversigt over denne liste, med korte forklaringer til hvordan det kan relateres til casen, vil herefter blive opstillet.

1. It would have to be a “real-world environment”

- » Ved at skabe en form for marked, vil brugeren nemmere kunne drage paralleller mellem spillet og virkeligheds situationer,.

2. The learning environment should be as similar as possible to the environment in which people will use what they learn

- » Ved at benytte mønster med “virkelige” værdier, kan brugeren nemmere se hvordan de skal bruges i virkeligheden.

3. It is important to provide an unlimited number of learners with exactly the same learning environment

- » Ved at skabe en applikation flere kan bruge på én gang, kan man tilbyde dem det samme miljø - hvilket sørger for at det er de samme de lærer, og på samme vilkår.

4. **The learning environment should be designed to allow a learner to repeat an experience facilitating a post mortem analysis of what took place**
 - » Ved at give brugeren mulighed for at se det beregningerne for den forrige handel, og hvor præcist der var svaret, ville brugeren kunne se eventuelle fejl - og reflektere derafter.
5. **The environment should be designed to facilitate variation along a single dimension or a given selection of dimensions**
 - » Ved at skabe kontrolleret variation af transaktionerne, kan brugeren lære at regne med forskellige tal.
6. **There should be some uniform means of assessing students' performance**
 - » Ved at skabe en form for statistik over brugernes præsentationer, ville en underviser kunne sammenligne elever. Dette kunne eventuelt gøres via et webinterface.
7. **The environment should store and present the students with pre-planned learning experiences, some of them in a particular order**
 - » Ved at præsentere brugeren for en række forudbestemte transaktioner på diverse tidspunkter - så som i starten, inden en "test", eller tilfældigt - der indeholder en række relaterbare transaktioner, kan man øve brugeren i en bestemt type beregninger.
8. **It should be possible for the student to explore new concepts and practice new techniques at his or her own pace**
 - » Ved at give brugeren mulighed for at arbejde i deres eget tempo, kan brugerne eksperimentere med forskellige metoder (for eksempel forskellige beregningsstrategier).
9. **The student should be given immediate positive (and ideally public) feedback for any success that is commensurate with that student's current level of attainment**
 - » Ved at give brugeren øjeblikkelig positiv feedback, både lyd- og grafiskmæssigt (point, gode lyde, bifald etc.), belønner man brugeren. Ved at implementere en måde så feedbacken bliver offentlig, kan det effektivisere feedbacken.
10. **There should be a sufficient "cost" to getting something wrong to motivate correction, but not so great that it leads to the student losing heart and giving up**
 - » Ved at "straffe" brugeren med en mild straf ved små afvigelser (såsom en forringet status bar), og ikke tillade brugeren at fortsætte ved store afvigelser, kan man sørge for at brugeren kan prøve igen hvis det er helt ved siden af. Dette kan motivere brugeren til at skabe et perfekt resultat, for at vedligeholde en positiv status.

11. The learning environment should, if possible, provide an enjoyable and stimulating experience

- » Ved at skabe et rart miljø, med god stemning (behageligt musik, rare figurer, behagelige omgivelser) og god playability, kan man skabe en god og stimulerende oplevelse for brugeren.

Ydermere kommer Devlin med fire argumenter for, hvorfor han mener at man skal bruge videospil i matematikundervisning:

- I. spils immersive miljø er ideelt for at lære hverdagsmatematik, da der er mulighed for at designe til flere forskellige læringssituationer,
- II. spillet kan give struktur til læringen, da det kan præsentere brugeren for en række variationer af samme problem,
- III. spil kan motivere brugeren til at fortsætte med at spille (og derved lære),
- IV. både spillet og det virtuelle miljø kan være behageligt.

Key Features of Gaming

The Key Features of Gaming er kapitel 7 i bogen, og omhandler en gennemgang af en liste lavet af John Beck og Mitchell Wade¹¹, der fokuserer på de evner og oplevelser spil kan give brugerne (blandt andet sociale færdigheder, multitasking og beslutningstagning).

Herunder ses en oversigt over listen, med korte beskrivelser af hvorfor Devlin mener at det er brugbart i udviklingen af spil, og en kort beskrivelse af hvordan det kan relateres til udviklingen af casen.

1. Failure doesn't hurt

- » I typisk matematikundervisning stabler fejlene sig bare oven på hinanden, hvilket giver brugeren ubehag.
- » Ved at kun at straffe brugeren minimalt ved små fejl, og hjælpe brugeren ved store fejl, kan dette ubehag forhindres.

2. Risk is a part of the game

- » For at lære er man nødt til at fejle, det samme gælder ved spil hvis man skal lære.
- » Dette kan relateres til punkt 1 - brugeren risikerer at regne forkert, men skal ikke straffes så meget at de frygter at fejle.

3. Feedback needs to be immediate

- » Øjeblikkelig positiv feedback hjælper læring.
- » Dette kan relateres til punkt 9 i listen i *Perfect Medium*-afsnittet.

¹¹ Beck, J. og Wade, M. (2004). *Got Game: How the Gamer Generation Is Reshaping Business Forever*, Harvard Business Press

4. Used to being the “star”

- » Oftest i klasser er der en eller to der er meget gode, og får en masse positiv feedback - dette sker på bekostning af de andre elever.
- » Ved at skabe feedback til hvert individ, får alle samme behandling for deres resultater.

5. Trial and error is almost always the best plan

- » Sammen med erfaring, er det at prøve sig frem (som regel) den bedste løsning - men i matematikundervisningen er dette ikke understøttet godt.
- » Hvis man giver brugeren mulighed for at prøve sig frem og fortryde tidligere valg, kan de få eksperimentere med løsninger.

6. There’s always an answer

- » Der er altid en løsning på dagligdags matematik.

7. I can figure it out

- » Selvom matematikproblemer som regel kan løses, er det ikke altid i skoler at eleverne føler dette.
- » Sørg for at problemerne er til at finde ud af, så brugerne ikke opgiver.

8. Competition is fun and familiar

- » Konkurrence i klasserne er oftest set som en opdeling mellem de dygtige elever, og de knap så dygtige. Dette kan forhindre de knap så dygtige i at deltage, fordi de ikke vil føle sig dårlige - og det kan forhindre de dygtige elever fra at deltage, fordi de ikke vil skille sig ud.
- » Ved at skabe et kontrolleret multiplayer miljø, kan man sørge for at begge typer elever har lyst til at deltage.

9. Bosses and rules are less important

- » Elever kan nemt føle sig restrikeret af regler og autoriteter.
- » I et spil er der mere mulighed for at bryde reglerne uden store konsekvenser - dette mindsker presset fra regler og autoriteter.

10. Used to group action and conflict

- » Gruppearbejde er normalt i klasseundervisningen, men sjældent bliver konflikter brugt til undervisningen.
- » Ved at skabe en form for konflikt i et spil, kan man motivere eleverne til at gennemføre sværere opgaver.

Udvikling

I dette afsnit vil udviklingsprocessen blive gennemgået. Først vil der være et afsnit omkring prototypens tekniske aspekter, i forhold til Cardboard som virtual reality teknologi, efterfulgt af en teknologianalyse, hvor konkurrerende teknologier vil blive analyseret. Derefter vil der være et kort afsnit omkring udviklingsprocessen generelt, samt et par tips til udvikling af Virtual Reality fra hjemmesiden Gamasutra. Så vil der blive kigget på processen der ledte til den første prototype, hvorefter de forskellige brugerundersøgelser og tilhørende udviklingsarbejde blive gennemgået, hvilket leder til et afsnit omkring den endelige prototype. Til sidst vil der være et afsnit omhandlende prototypens software, der blandt andet vil indeholde en oversigt over produktets struktur i Unity, og gennemgang af koden.

Tekniske aspekter ved Google Cardboard

Google Cardboard¹² er en virtual reality enhed, der ved hjælp af en smartphone, giver brugerne mulighed for at opleve virtual reality. Cardboard, der betyder pap på dansk, er givet enheden, da den hovedsageligt er lavet af pap. Ud over pap består Cardboard af to linser og en form for input (en magnet på den gamle version, og en knap der trykker på skærmen på den nye version).



Billede 5: Google Cardboard i brug

¹² <https://vr.google.com/cardboard/index.html>

Teknologien fungerer ved at brugeren starter en kompatibel applikation op på sin smartphone, og placerer den i Cardboard holderen. Brugeren holder så enheden op foran sine øjne, og kan derefter (ved hjælp af smartphonens indbyggede sensorer, heriblandt gyroskopet) kigge rundt i et virtuelt univers. Enheden reagerer ikke på at brugeren går frem eller tilbage i den fysiske verden.

Cardboard har, som tidligere nævnt, kun en enkelt inputmetode. I den "gamle" version af Cardboard (som denne prototype er udviklet til) fungere input ved at smartphonen opfanger ændring i magnetfeltet - hvilket skabes ved at bevæge magneter på Cardboard enheden. I den nyere version, kaldet Cardboard V2, registrerer enheden et input, ved at brugeren trykker på en knap, der så trykker på skærmen. Dette gør det nemmere at kode til, da man derfor bare skal lytte efter en berøring, frem for ændringer i et magnetfelt.

Valget faldt på Cardboard, da teknologien er billig - 69 DKK for den gamle udgave på Komplett.dk¹³, omkring 100 DKK for V2 fra iamcardboard.com¹⁴, og endnu billigere for off-brand versioner, bestilt fra asien - hvilket gør det muligt for institutioner at købe en masse enheder billigt, samtidig med at eleverne kan gøre brug af deres egne smartphones. En undersøgelse af YouGov¹⁵ gjort i 2015, viste at 90% af eleverne i 4.-6. klasse, og 96% af eleverne i 7.-9. klasse havde smartphones. Ud over deres egen smartphones, kunne man forestille sig at institutionerne indkøbte en række smartphones, specielt købt til dette formål, når de bestilte Cardboard enheder.

Dette ville give et stort antal af elever muligheden for at benytte sig af simple læringsspil, udviklet til bestemte formål. Det negative ved at bruge Cardboard ville være at applikationerne kun kunne benytte sig af et enkelt input (med mindre applikationerne understøttede eksterne inputs, såsom gamepads), og at applikationerne grafik- og performancemæssigt er begrænset af telefonernes hardware.

I det næste afsnit vil en række af lignende teknologier blive gennemgået - både i forhold til hvordan de fungerer som virtual reality teknologi, og hvordan de kunne bruges til undervisning.

¹³ Set den 27. maj 2016

¹⁴ Set den 27. maj 2016, til \$14.99, kurs \$1 = 6,690 DKK.

¹⁵ Her refereret til at Nordjyske: <http://nordjyske.dk/nyheder/naesten-alle-boern-har-en-smartphone/b0b1e8d5-4626-4a3d-843c-afd314428701/112/1513>

Teknologianalyse af lignende teknologier

Samsungs Gear VR

Gear VR ligner på bundlinjen Cardboard - det er en en virtual reality platform, der ved hjælp af smartphones giver brugeren mulighed for at opleve virtual reality. Forskellen¹⁶ kan dog ses når man tager et kig på det hardwaremæssige. Gear VR har indbyggede sensorer i headsettet, specialudviklet til at håndtere virtual reality oplevelsen - dette er med til at skabe en meget mere flydende oplevelse for brugeren. Selve kvaliteten af materialerne er også højere end Cardboard (der jo består af pap), og Gear VR er udstyret med indbyggede ventilationsskakte, tilbage- og lydknapper og en touchpad.

Da Gear VR er udviklet til Samsung telefoner, er det dog kun en begrænset række af telefoner der er kompatible med Gear VR - dette giver dog mulighed for at optimere software, og tilbyde eksklusive applikationer til brugerne. Samtidig er prisen på Gear VR også en del højere end Cardboard.

Grundet den begrænsede tilgængelighed af kompatible enhedstyper, ville det ikke være optimalt at udvikle læringsspil applikationer til Gear VR - dette ville kræve at den potentielle målgruppe skulle anskaffe sig en bestemt type telefon. Og selvom prisen på Gear VR alene ikke nærmer sig prisen på teknologier som Oculus rift, så bliver prisen inklusiv en kompatibel telefon tæt nok på, til at det ville kunne betale sig at anskaffe sig andre teknologier.

Det skal dog også siges at Cardboard applikationer, ved hjælp af en tredjeparts applikation¹⁷, fungerer på Gear VR, så hvis teknologien i forvejen er tilgængelig, kan en smartphone lærings applikation benytte de fordele der er.

Oculus Rift

Oculus Rift¹⁸ (herefter kaldt Rift) er et virtual reality headset, udviklet af Oculus VR, der i 2014 blev opkøbt af det sociale medie Facebook. Rift er resultatet af en vellykket Kickstarter kampagne, der har affødt en række af prototype - den kommercielle version af Rift, blev udgivet den 28. marts 2016.

¹⁶ <http://www.androidcentral.com/google-cardboard-vs-samsung-gear-vr>

¹⁷ <https://play.google.com/store/apps/details?id=com.youtopiavr.cbenabler>

¹⁸ <https://www.oculus.com/>



Billede 6: Oculus Rift

Rift samarbejder med brugerens computer om at skabe en virtual reality oplevelse for brugeren. Riften sender signaler til computeren om hvad der sker, og computeren sender så et videosignal tilbage til brugeren, optimeret til Riften.

Rift består af et display med et 2160x1200 display, hvilket betyder at hvert øje har en opløsning på 1080x1200 tilgængelig - til sammenligning har en iPhone 6 en opløsning på 750x1334, en HTC One M9 en opløsning på 1080x1920 og en Samsung Galaxy S7 (der er udviklet til at være kompatibel med Gear VR) en opløsning på 1440x2560. Den mest normale opløsning, ifølge en undersøgelse af DeviceAtlas¹⁹ i 2015, er dog 640x1136 - dette betyder at hvert øje har en 640x564 opløsning tilgængelig. Skaberen af Rift, Palmer Luckey, har dog udtalt sig om²⁰ at en 8K opløsning til hvert øje, muligvis vil være nok til at lave en virtual reality oplevelse uden synlig anti-alias og pixels - og opløsningen skal være endnu større før at det er helt perfekt.

Rift har potentiale til læringsspil - men med en pris på næsten 4000 kroner²¹ pr enhed er det en dyr fornøjelse hvis mange elever skal benytte teknologien samtidig. En institution med et lille håndfuld enheder, kunne med fordel dele elever op i mindre hold, der så på skift kunne prøve.

¹⁹ <https://deviceatlas.com/blog/most-popular-smartphone-screen-resolutions-2015>

²⁰ <http://arstechnica.com/gaming/2013/09/virtual-perfection-why-8k-resolution-per-eye-isnt-enough-for-perfect-vr/>

²¹ Oculus Rifts hjemmeside, 25. maj 2016. \$599, kurs \$1 = 6,671 DKK.

HTC Vive

HTC Vive (herefter kaldt Vive) er et virtual reality headset udviklet af mobil- og tablet fabrikanten HTC og spiludviklerne Valve (der også står bag distributionsplatformen Steam).



Billede 7: HTC Vive

Opløsningsmæssigt har Vive den samme opløsning som Rift. Den fungerer sammen med motion controllere og base stations, der sammen gør det muligt for brugeren at gå rundt i et designeret område, på 4.6 kvadratmeter, og spille. Der er også monteret et frontkamera, der gør det muligt for enheden at advare brugeren, hvis de er ved at gå ind i en genstand, eller en person bevæger sig ind i synsfeltet.

Den nuværende pris²² på Vive er 7299 kroner, hvilket placerer den i en højere prisklasse end Rift - dette gør det til en meget dyr fornøjelse, hvis der skal købes enheder ind til en hel skoleklasse. Vive giver dog brugeren mulighed for mange flere ting end de andre nævnte teknologier, hvilket kan give et bedre indblik i, hvad vi kan forvente af virtual reality i fremtiden.

²² Elgiganten.dk, d. 25. maj 2016

Google Daydream

Google Daydream er Googles nyeste skud på deres virtual reality-stamme. Teknologien blev præsenteret på Googles I/O event, den 18. maj 2016, og vil være klar til kommerciel brug efteråret 2016.

På samme måde som Cardboard, vil Google Daydream benytte sig af brugerens telefon, men tilhørende til headsettet følger en motion controller, der giver brugeren mulighed for mere interaktion med applikationen - et eksempel fra deres præsentation er en bruger, der svinger en tryllestav. Dette placerer Google Daydream et sted mellem Cardboard og Vive, da den lader brugeren benytte sit eget hardware, men samtidig tilbyder udvidet interaktion med en motion controller.

Der er endnu ikke blevet annonceret en pris for enheden, men da Google samarbejder med Samsung (der står bag Gear VR, og fortsat vil blive ved med at supportere dette), kan det tænkes at prislejet bliver der omkring.

Den spekulerede pris, forbedrede teknologi og mulighed for bedre brugerinput, vil gøre Google Daydream til en god platform for læringsspil, og det bliver kun interessant at se hvordan der bliver taget imod teknologien, når den udkommer.

Udvikling generelt

Udviklingen af prototypen foregik i Unity 5.3.4, ved brug af Googles *VR SDK for Unity* (herefter kaldt SDKen), der gør det muligt at lave applikationer virtual reality applikationer til Googles platforme (Cardboard og Daydream). Under udviklingsprocessen er der blevet gjort brug af Unitys Asset Store, der gør det muligt at downloade assets - både gratis og betalte. Der er også brugt lyd fra tredjeparter - en oversigt med links kan findes i bilagene.

Grundet SDKen var det muligt at teste applikationen på både computer og telefon - dette gjorde udviklingsarbejdet en del lettere, da det derfor ikke var nødvendigt at installere applikationen på telefonen for hver eneste ændring.

Første prototype

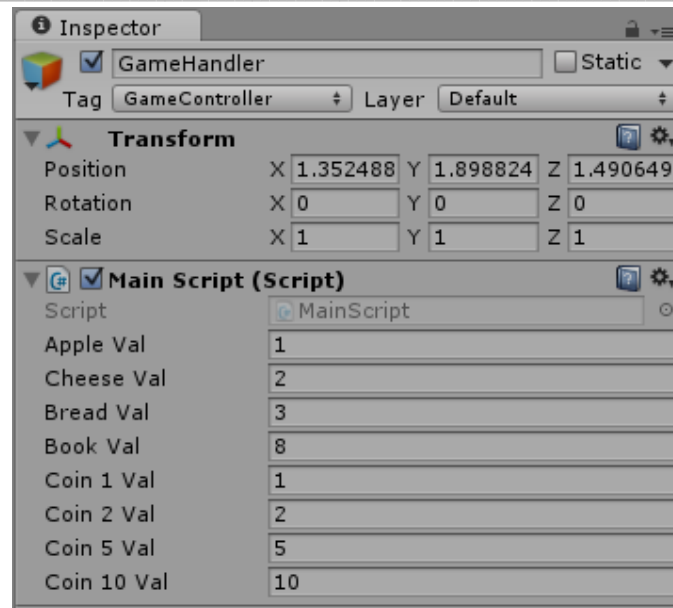
Den første prototype blev udviklet på baggrund af den indledende idégenereringsprocess. Denne prototype blev udviklet i en række mindre iterationer. Først blev et spiller objekt placeret i midten af en overflade, hvorefter der blev lavet en bod med fire forskellige objekter (æble, ost, brød og bog), to typer valuta (guldmønt og guldbar), og en placeholder figur (en grøn "tændstikmand" på hvid baggrund). Derefter blev der udviklet en *Raycast*²³ funktion, så de forskellige objekter kunne markeres ved at kigge på dem. Objekterne skiftede farve når spilleren kiggede på dem, til grøn som standard, men klikkede spilleren på objekterne, så skiftede farverne. Dette gjordes for at teste spiller input og objekt håndtering. Et screenshot af denne iteration kan ses på billede 8.



Billede 8: Tidlig version af den første prototype

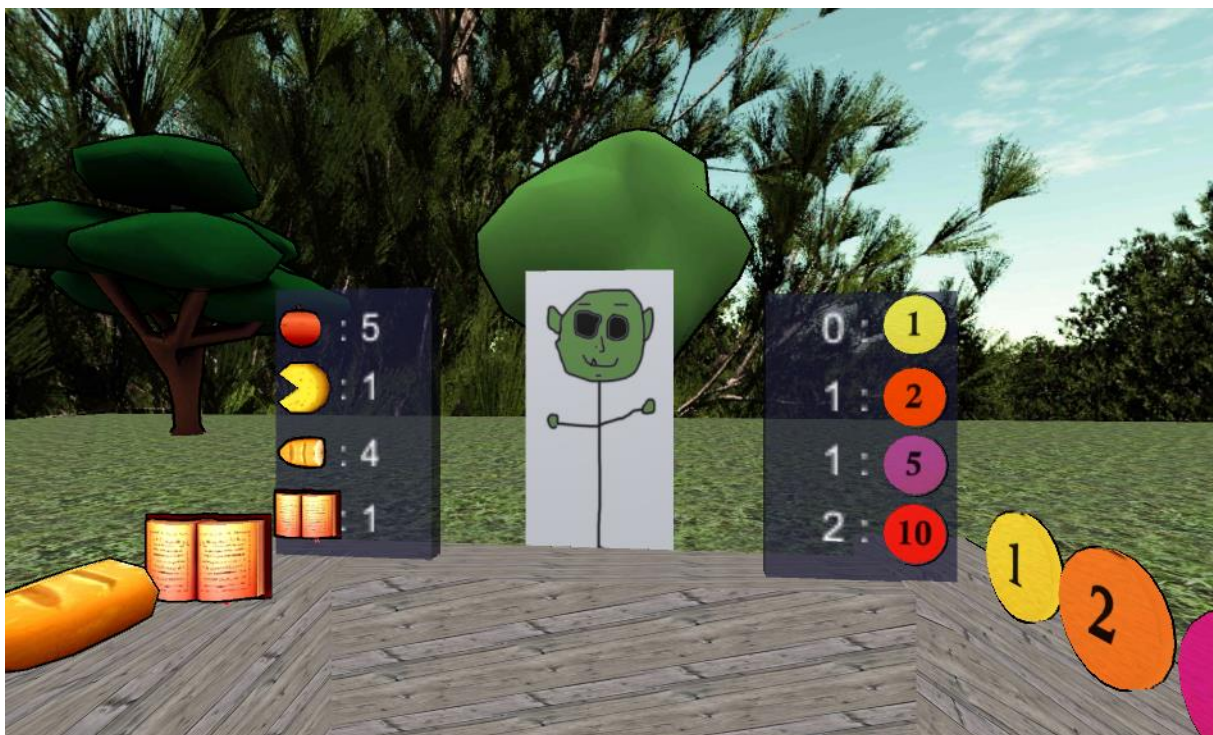
Der blev derefter rettet på koden, så objekterne der blev klikket på, blev tilføjet til et array. Guldmønten og -baren blev skiftet ud med fire mønter, for at afspejle det danske møntsystem, med værdierne: 1, 2, 5 og 10 (20'eren blev undladt, da værdierne i spillet var lave, for at holde niveauet på 1-cifrede udregninger og 10'ere). Array-metoden blev dog, grundet effektivisering, lavet om til at objekterne fik tilegnet en specifik værdi i MainScriptet, hvilket gjorde det muligt hurtigt at ændre værdierne i Unity. En oversigt over dette kan ses på billede 9.

²³ <http://docs.unity3d.com/ScriptReference/Physics.Raycast.html>



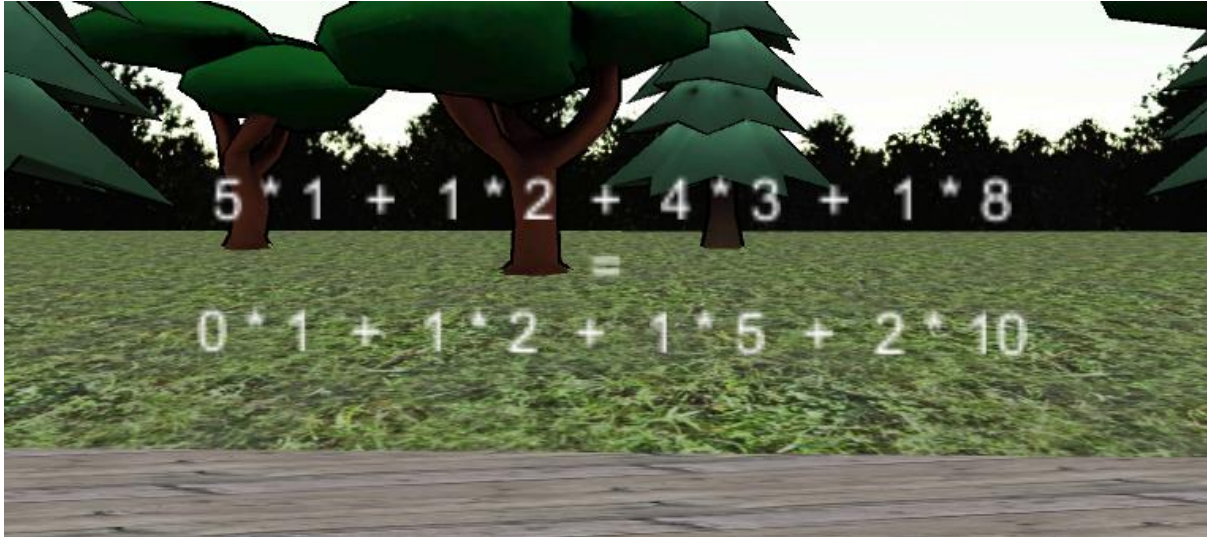
Billede 9: GameHandler objekt, med MainScript tilføjet

Derefter blev der tilføjet to objekter, der hver især indeholdt nedskalerede modeller af objekterne, med tilhørende tekst til. Denne tekst bruges til at vise hvor mange objekter brugeren har valgt, og kan bruges til at skabe en hurtig oversigt. Et screenshot af dette kan ses på billede 10 .



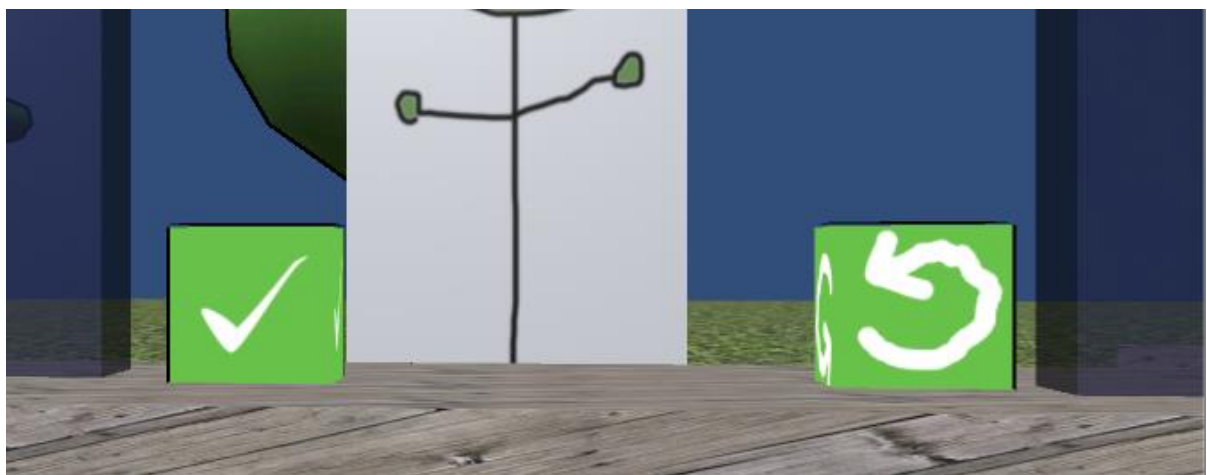
Billede 10: Prototype med tilrettede teksturer og oversigt over valgte enheder

Derefter blev der tilføjet en tekst, der viste et regnestykke, hvor den bruger antallet af valgte objekter, og deres værdi. Resultatet af stykkerne kan ikke ses, da det er meningen at brugeren selv skal regne dette ud, og derefter sammenligne. Dette kan ses på billede 11.



Billede 11: Screenshot af udregningerne

Der blev derefter tilføjet to knapper mere; en knap til at godkende (rette tegn), der skal tjekke om transaktionen er korrekt, og en reset-knap (cirkel, med en pil mod uret), der skal resette de valgte objekter. Disse to knapper kan ses på billede 12. Kun koden for reset-knappen blev implementeret i denne iteration af prototypen. Ydermere blev der tilføjet lyd til prototypen. Når der klikkes på en vare afspilles en kort, blød klik lyd, når der klikkes på en mønt afspilles en mønt lyd²⁴, knappen til at godkende afspiller en kasse register lyd, og reset knappen spiller en . Rolig musik blev også tilføjet som baggrundslyd - dette musikstykke blev valgt, da det samtidig med at være roligt, stadig var varierende.



Billede 12 : En knap til at godkende (til venstre) og en knap til at resette (til højre)

²⁴ En lyd der minder om den klassiske mønt lyd fra *Super Mario Bros*

Det var denne prototype der blev testet med i brugertesten der ledte til den første iteration. Der var stadig mindre fejl tilstede, men idéen var til stede, og brugeren havde næsten (se afsnittet **Brugertest**, under **Første iteration**) mulighed for både at kigge på, og vælge objekter.

Første iteration

Brugertest

Grundet en fejl i koden der skulle integrere magnet funktionen i Cardboard, var det desværre ikke muligt for brugeren at vælge objekter med magneten. En hotfix til dette blev lavet ved at bruge et stykke sølvpapir til at trykke på skærmen med - det blev placeret på siden af enheden (modsat magneten). Da den ene af brugertestene blev foretaget via email, var det desværre ikke en mulighed at bruge hotfixet der. Brugertesten var både en test på teknologien, og en evaluering af idéen.

Michael M

Michael er E-læringsudvikler ved HF+VUC FYN (herefter kaldt VUC), og står blandt andet for håndteringen af Cardboard enheder på VUC. Efter en mailkorrespondance modtog han den første prototype, så han kunne teste det på sin egen Cardboard enhed. Hans feedback kan ses i bilaget **Michael M** under **Brugertests**.

Michaels feedback var hovedsageligt på idéen, og den blev både kaldt “god” og “spændende”. Som nævnt tidligere var der desværre en fejl på magneten, og han havde derfor ikke mulighed for at prøve spillet som planlagt. Det blev aftalt at sende applikationen til ham igen, når problemet blev løst²⁵.

Hannah D

Hannah går i 0. klasse, og skal starte i 1. klasse efter sommerferien - hun er derfor lige inde i den spekulerede målgruppe (men dog stadig i den yngre del). Hendes svar på en række spørgsmål kan finde i bilaget **Hannah D** under **Brugertests**.

²⁵ Michael fik sendt en version af prototypen et sted under udviklingen af anden iteration. Han nåede desværre ikke at vende tilbage med feedback inden projektet skulle indleveres.



Billede 13 og 14 : Hannah tester prototypen

Hannah prøvede produktet, lige efter at hun blev hentet af sin far i SFO. Hun havde umiddelbart ikke de store problemer med at forstå at hun kunne kigge rundt igennem Cardboard - også selvom hun lige skulle finde ud at man ikke flyttede sig i den virtuelle verden, bare fordi man gjorde det i den virkelige verden.

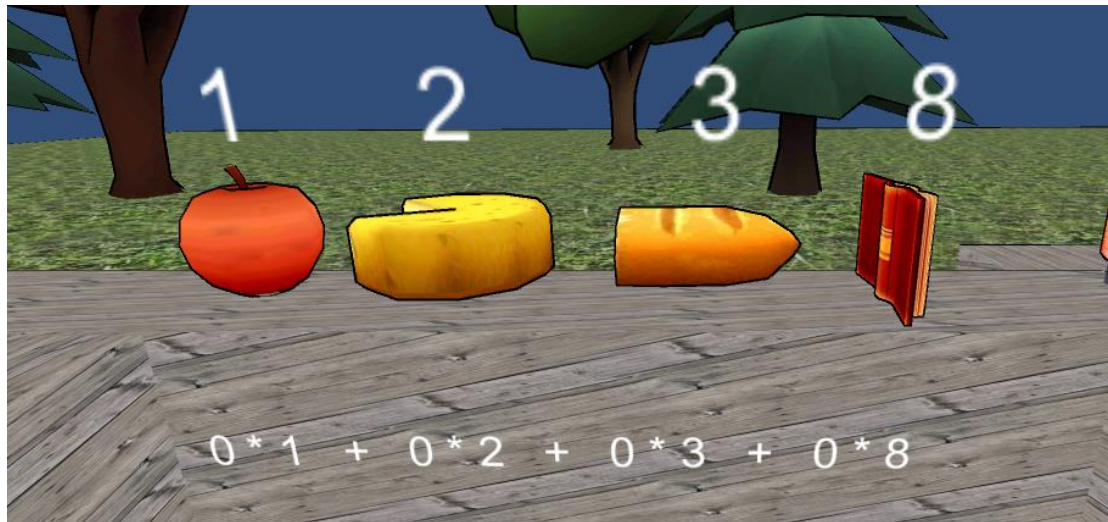
Hannah havde en masse kommentarer til både produktet og teknologien. Om selve produktet, mente hun at musikken gerne måtte være lidt anderledes - enten noget der var lidt hurtigere, eller noget med sang til. Dette udsagn står meget i kontrast til idéen bag musikken. Hun kommenterede også på at det kunne være nemmere hvis man kunne tage fat i varerne, og så give dem til kunden. Hun mente også at det ville være sjovt hvis man kunne se ned, og så se "sig selv" - dette er ikke noget der tidligere er blevet overvejet, men det giver god mening i forhold til at skabe en identitet for brugeren. Hannah ville også foretrække en mere "virkelig" repræsentation af verdenen - også selvom hun synes det var sjovt at varerne var meget store. Hun troede dog også at osten på tal-displayet var en PACMAN, da hun først prøvede prototypen.

Selve teknologien synes hun var rigtig sjovt, men at det ville gøre det lidt nemmere hvis den sad fast med et bånd (hvilket er en mulighed, hvis materialerne er tilgængelige). Hun mente også at det ville være smart med en form for sensor der alarmerede brugeren, hvis de var ved at gå ind i ting. Dette er dog ikke en nødvendighed, da Cardboard er en stationær teknologi, hvor brugeren kun skal bevæge hovedet og rotere om sig selv, og ikke gå rundt.

Udvikling

Efter disse brugertests blev der udviklet lidt videre på prototypen. Først blev der implementeret kode, der gjorde det muligt for spillet at tjekke om værdierne, på de varer og

mønter brugeren havde valgt, stemte overens - der blev også tilføjet lyde til resultaterne. Derefter blev der sat tal ind over varerne, så brugeren kan se prisen på varerne. Beregningerne blev også flyttet hen under de objekter de repræsenterer. Et screenshot af dette kan ses på billede 15.



Billede 15 : Tal viser varernes værdi og tilhørende beregning

Der blev også lavet en oversigt, hvor brugeren kan se sine resultater, samtidig med at der blev programmeret tilfældige forespørgsler fra kunden - dette kan ses på billede 16.



Billede 16 : Brugerens resultater og en forespørgsel fra kunden

I denne prototype får brugeren 100 point for at regne rigtigt, og -10 point for at regne forkert. Der bliver genereret en ny forespørgsel efter hvert gæt. Gættet bliver dog verificeret, og brugeren er nødt til at vælge mindst én mønt og én vare.

Anden iteration

Brugertest

David D

David er en medstuderende på Læring- og Oplevelsesteknologi uddannelsen, og er far til Hannah der også blev testede prototypen. David prøvede ved samme lejlighed også produktet, og sendte efterfølgende en mail med lidt feedback, både til idé og produktet. Mailen med feedbacken kom først efter at den nye prototype var sendt ud, og derfor bliver det først medtænkt i anden iteration. Hans mail kan ses i bilaget **David D** under **Brugertests**.



Billede 17 : David tester prototypen, mens Hannah kigger på

David synes at teknologien (Cardboard) virkede interessant, men han havde dog lidt bekymringer angående det sociale aspekt - altså at man er lukket inde i verdenen alene. Han foreslog at man eventuelt kunne ændre konceptet, så det opfordrer brugerne til at skiftes om enheden.

Han var positiv stillet over for den virtuelle verden, og sammenlignede det med *Pixeline* og andre spil skabt til målgruppen. Selve idéen var han også begejstret for, selvom der i prototypen manglede noget der fortalte brugerne hvad de skulle gøre - han foreslog her et kasseapparat, der hvor der skal betales.

Lærings- og Oplevelsesteknologi studerende

Denne brugertest foregik på tre medstuderende; **Martin K, Trine T og Nicklas F** (der alle kan findes i bilagene under **Brugertests**). De fik alle tre beskrevet konceptet, og fik så lov til at prøve prototypen. De fik også tilsendt en PC version af prototypen til at teste derhjemme, inden de gav feedback.

Deres feedback var generelt positivt stemt, og de kom med en del forslag til hvordan prototypen kunne forbedres, eksempelvis: et pengeglas der ligeså stille fyldes op, i takt med at man får korrekte svar; visuel forbedring af knapperne, og et kasseapparat til at indikere handelsområdet.

Alberte P

Alberte er matematikudlært lærerstuderende - hvilket betyder at hun har gennemført matematiklærer delen af læreruddannelsen. Det var derfor rigtig interessant at modtage feedback fra hende, da hun kunne bidrage med et fagligt blik på idéen og prototypen. Hun fik sendt en PC version af prototypen. Hendes feedback kan læses i bilaget **Alberte P** under **Brugertests**.

Alberte var rigtig positivt stemt over for prototypen, men hun havde dog et par kommentarer til, hvordan det kunne forbedres. Hun foreslog en form for prisliste, og en form for enheder på tallene, så der ikke er tvivl om tallenes betydning. Hendes andet forslag var at dele den visuelle beregning ud i mindre stykker, da målgruppen endnu ikke har lært så store regneudtryk - altså i stedet for $1*2 + 3*2 + \dots$, så $| 1*2 | + | 3*2 | + \dots$ for at gøre det mere forståeligt for dem. Hun mente dog at niveauet var passende, men at det kunne hjælpe at skjule kompleksiteten af matematikken for brugeren, da de endnu ikke kan overskue så store udtryk.

Hun nævnte også at eleverne er rigtig glade for at arbejde med IT, og at prototypen har et hverdagsaspekt, der kan være meget behjælpeligt for eleverne.

Udvikling

Efter at have modtaget feedback fra brugere testene, blev der igen udviklet videre på prototypen. Det første der blev ændret, var det visuelle ved transaktionsområdet - godkend- og resetknappen fik lavet sorte kanter, reset knappen ændrede farve og symbol, der blev tilføjet en meget simpel pengekasse og taleboble. Dette kan ses på billede 18.



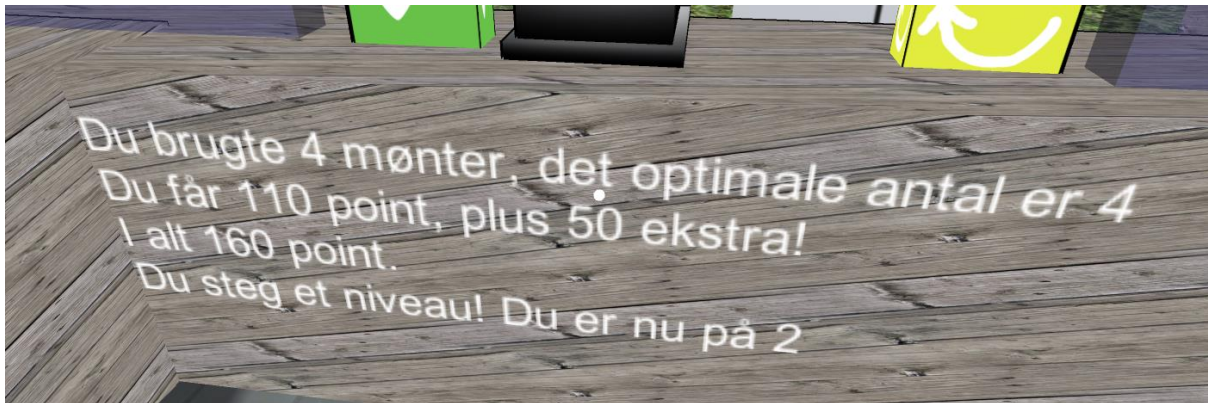
Billede 18 : Opdateret grafik og nye modeller

Derefter blev tilføjet enheder efter prisen, samtidig med at der blev lavet et skilt. Beregningen under objekterne blev også delt ud, samtidig med at der kom enheder på - dette gjorde det nemmere at adskille regnestykkerne, og gøre det mere overskueligt. Billede 19 er et screenshot af ændringerne.



Billede 19: Prisskilte, og opdelte beregninger

Derudover blev der udviklet videre på koden til at håndtere pointgivning. For at tjekke om brugeren er berettiget til en bonus, er der en algoritme der beregner den optimale løsning - altså det færreste antal mønter, der giver det rigtige resultat. Rammer brugeren dette resultat, uddeles der 50 ekstra point. Derudover blev der tilføjet en beskrivende tekst under transaktionsområdet, der viser resultatet af handlen, hvilket kan ses på billede 20. Der blev også tilføjet et sværhedsniveau, der stiger for hvert tusinde point - det påvirker det maksimale antal varer kunden kan bede om, og mængden af point brugeren får.

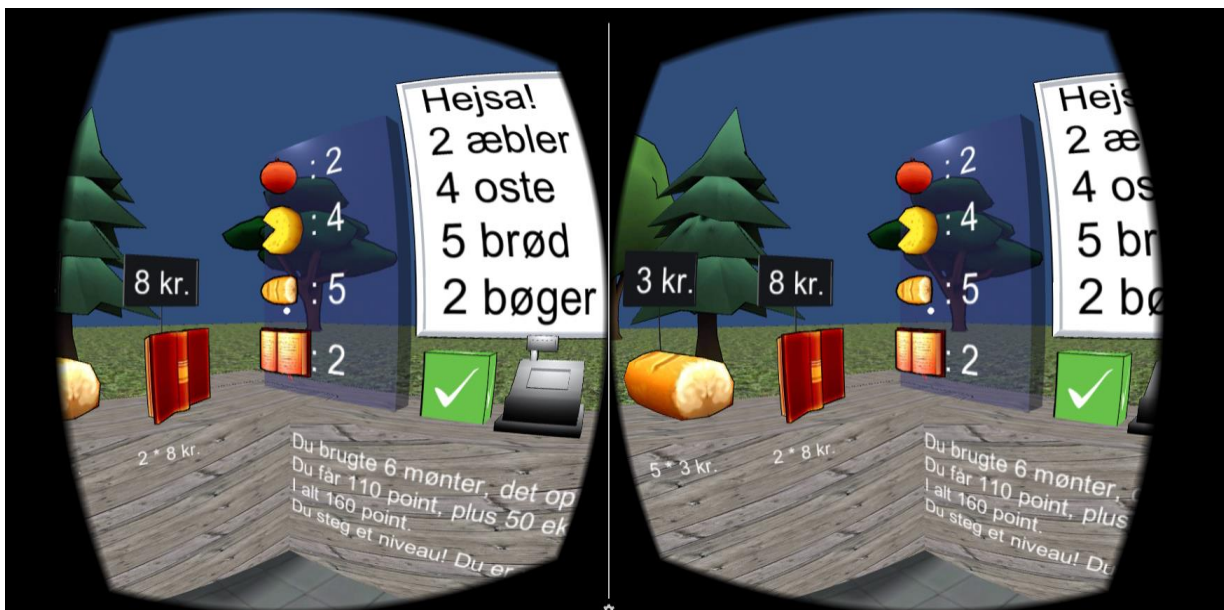


Billede 20: Statustekst og point information.

Endelige prototype

Der blev ikke udført flere brugertests efter den anden iteration. Der var en dialog kørende med en uddannelsesinstitution om en brugertest, men grundet en presset tidsfrist fra begge parter, blev det desværre ikke muligt at udføre en afsluttende brugertest.

Den “endelige” prototype er kun endelig i den forstand, at det ikke var muligt at videreudvikle mere på den inden for projektets ramme. Der er rig mulighed for videreudvikling af både koncept og software, hvilket der vendes tilbage til i afsnittet **Perspektivering**.

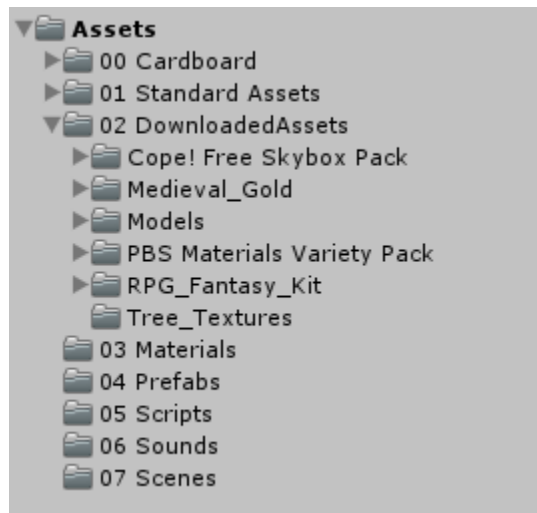


Billede 21: Prototypen som den ser ud på smartphone

Prototypens software

Struktur i Unity

For at skabe en overskuelig struktur i Unity, blev prototypens filer delt ind i forskellige mapper, en oversigt over disse kan ses på billede 22.



Billede 22: Mapestruktur i Unity

Strukturen er skabt for at give et nemt overblik i udviklingsprocessen: mappe 00 og 01 er faste, og skal ikke ændres på; eksternt hentede assets bliver placeret i 02; materialer og teksturer i 03; objekter, og grupper af objekter bliver gemt i 04; i 05 er koden placeret; alle lydfiler er placeret i 06, og scener bliver gemt i 07.

Gennemgang af kodestykker

Den endelige prototype bestod af en række scripts. Koden bag disse scripts kan findes i bilagene. Der vil i dette underafsnit blive gennemgået . De gennemgåede kodefiler kan findes i bilagene under **Kode**. Kun interessante dele af koden vil blive gennemgået. Der vil blive refereret til linjenumrene.

Cardboard Magnet kode filerne er ikke med, da de er hentet eksternt - et link kan ses i **Anerkendelser - materiale til software**. Den eneste modificering i filerne er, at der er fjernet et par linjer debug kode, og den kalder *clickingSomeway*-metoden i *RaycastPlayer.cs*. *SlowlyRotate.cs* er ikke tilføjet, da den består af en enkelt skrevet linje kode, der roterer objektet. Kode fra Cardboard SDKen og Unity Standard Assets er heller ikke tilføjet.

MainScript.cs

Linje 120-140: `generateRequest()` metoden defineres her. For hver af de fire objekter udregnes en tilfældig værdi (indenfor en rækkevidde, der er bestemt af niveauet). Derefter sammensættes en string, der så fortæller brugeren hvilke varer der bliver anmodet om (uden

at vise de objekter der har 0 værdier). Er summen af alle fire anmodninger 0, betyder det at der ikke bliver spurgt om varer, og metoden kører igen.

Linje 142-292: Heri `addObjectToArray()` metoden (som ikke har tilføjet objekter til arrays siden prototypen), der holder styr på hvormange varer og mønter brugeren har valgt, samtidig med at den vedligeholder statistik og statustekst. Det er også her at der bliver tjekket om mønternes og varernes værdi stemmer overens. For at give en mere overskuelig oversigt, deles denne op i to stykker.

Linje 144-255: Her køres tre if-sætninger. Først tjekkes der om der klikkes på `resetBtn`, hvilket sætter de valgte værdier til 0. Derefter tjekkes der om der klikkes på `verifyBtn`, hvilket betyder at brugeren prøver at godkende transaktionen. Der tjekkes først om antallet af de forskellige varer er korrekt (linje 153-162). Derefter tjekkes der hvor vist at prisen stemmer overens eller ej. Er prisen forkert (linje 198-209) mister spilleren 10 point, der afspilles en lyd, statistikken opdateres, de valgte objekter nulstilles og der genereres en ny anmodning. Er prisen korrekt (linje 164-197) sker der stort set det samme som hvis prisen var forkert, ud over at spilleren får point istedet for at miste dem. Der bliver også tjekket om det er den optimale løsning, og der bliver givet point ud fra dette. Efter det, tjekkes det om spilleren skal stige et niveau. Er det hverken `reset`- eller `verifyBtn`, betyder det, at det er et objekt, og der køres derfor en switch case, der øger antallet af den valgte vare i systemet.

Linje 257-292: Her håndteres det meste af teksten i spillet, det bliver opdateret på baggrund af de variabler der sættes i den tidligere if sætning

Linje 346-369: Her defineres `Change` metoden, der er en lettere modificeret kode lånt fra *dotnetperls.com*. Algoritmen finder den optimale løsning af mønter, i forhold til at bruge færrest antal, og stadig få det rigtige resultat.

RaycastPlayer.cs

Linje 29-55: Her tjekkes om `rayCast` funktionen (der blev skabt på linje 26) rammer noget, og om det den rammer, er taget `selectableObj` - i dette tilfælde sættes dens `selected` værdi til true (hvilket gør at den skifter farve, se hvorfor i `ObjectHandler.cs`). Samtidig lytter den efter et klik fra musen (til PC versionen). Hvis den ikke rammer et `selectable` objekt sætte den `selected` og `isHit` til false.

Linje 58-67: Her er metoden `clickingSomeway()` defineret - den kaldes af det magnetiske input, og ændrer de samme ting som et klik med musen.

ObjectHandler.cs

Linje 13-17: I `Start()`-metoden sættes default materialet. `whiteDefMat` bliver ikke benyttet.

Linje 21-43: I Update() funktionen tjekkes der om objektets *selected* værdi er blevet ændret til true. Er den true, skiftes materialearven til grøn, for at markere objektet. Er den ikke true, sættes farven til default materialets farve. timeLeft benyttes ikke i prototypen, men blev brugt i forbindelse med debugging. Den gjorde at objekterne skiftede farve efter en vis mængde tid var gået.

Afrunding

Konklusion

Ved at benytte relevante teorier og analysere læringsmål, var det muligt at skabe en fungerende prototype på et virtual reality læringsspil. Selvom valget af Cardboard teknologien limiterede kompleksiteten af prototypen, blev der udviklet et velfungerende spil til en let tilgængelig - både for brugere og uddannelsesinstitutioner - platform (der perspektiveres til hvordan prototypen kunne have draget nytte af mere avancerede teknologier, i perspektiveringsafsnittet).

De Fælles Mål gav et indblik i potentielle målgrupper og læringsmateriale, og James Paul Gees og Keith Devlins teorier hjalp med at designe en prototype der støttede sig op af ekspertviden. Brugertestene hjalp med at forme prototypen til et mere fuldkomment værk - også selvom alle planlagte tests desværre ikke blev udført.

Prototypen har, på baggrund af teori og læringsmål, potentiale til at blive en del af folkeskolens undervisning. Og da det er en prototype, er der stadig en masse uudforsket udviklingspotentiale, hvilket der også kan læses mere om i det efterfølgende perspektiveringsafsnit.

For at vende tilbage til det indledende citat af Douglas Adams (omkring effektiviteten af lasere i virtuelle rum), kan man sammenligne med læring: læring i virtuelle rum er ligeså effektivt som læring i virkeligheden - ligegyldigt om man lærer i en virtuel verden eller i virkeligheden, **så lærer man dét man lærer.**

Perspektivering

Da det endelige produkt stadig er en prototype, er det relevant at se på hvilke muligheder der er for videreudvikling.

Teknologien

Først er det værd at tage et kig på de tilgængelige teknologier, som beskrevet i afsnittet **Teknologianalyse.**

Cardboard er, som tidligere beskrevet, en teknologi der giver mulighed for at nå ud til en bred brugergruppe billigt - den har dog visse svagheder; heriblandt en medfødt mulighed for avancerede input, og software begrænset af brugerens telefon. De to teknologier der minder mest om Cardboard er Gear VR og Daydream. Begge er dog begrænsede af telefonens hardware, men har adgang til mere avancerede inputs, som begge vil optimere brugeroplevelsen; Gear VR har en touchpad og Daydream har en form for motion controller. Daydream er dog den, af de to teknologier, som prototypen ville drage mest nytte af. Der er dog ikke særlig meget information om Daydream endnu, men hvis teknologien bliver billig og tilgængelig på mange telefoner, kan den danne en solid bund for en videreudvikling.

Det er dog mere spændende at kigge på mulighederne ved de to andre teknologier: Rift og Vive. Selvom disse enheder rent prismæssigt ikke egner sig til hele klasse af gangen, giver de, med deres avancerede teknologi, mulighed for mere spændende løsninger - hvilket kan give en mere immersive oplevelse for brugeren: Rift er stationær, hvor Vive giver mulighed for brugeren at bevæge sig; Rift har mulighed for forskellige input typer, heriblandt tastatur/mus, controller og Leap Motion²⁶, og Vive har bevægelse og to motion controllere. Især muligheden for Leap Motion til Rift er værd at kigge på - dette ville give brugerne mulighed for at bruge hænderne til samle både varer og mønter op, og aflevere dem til kunden, hvilket ville gøre spillet endnu mere virkelighedsnært. På samme måde kan Vive give en bedre immersion, da brugeren så ville kunne bevæge sig rundt i boden og bruge begge motion controllere til at flytte ting med.

Som beskrevet i teknologianalysen er begge disse teknologier væsentligt dyrere end Cardboard (især når der skal indkøbes hardware der kan køre spillene i ordentlig kvalitet), men alt fra én enkelt til en lille håndfuld enheder ville åbne op for mange muligheder.

Prototypen

På baggrund af teori og brugertests, er der stadig mange muligheder for videreudvikling af prototypen, heriblandt:

- » **Menu**

- » En menu kunne implementeres, hvilket ville give mulighed for at ændre en række af spillets aspekter, og se informationer

- » **Indledende skærm/knap med instruktioner**

- » Meningen med dette ville være at skabe en brugervenlig instruktion i, hvordan brugeren spiller spillet. Denne information kunne enten dukke op i starten af spillet, eller når brugeren valgte den på menuen (*information on-demand*)

²⁶ En controller der aflæser brugerens hænder og fingre: <https://www.leapmotion.com/>

» **Web interface med database**

- » En form for web interface med tilhørende database, kunne bruges til at gemme brugernes fremskridt. Dette ville kræve en form for login i applikationen, men ville give for både lærere, forældre, og eleven selv, mulighed for at se forbedringer på længere sigt

» **Dybdegående statistik**

- » Dette hænger sammen med web interfacet, men det ville også give mening at implementere det i applikationen, som et menupunkt.

» **Implementering af tæt-på point**

- » Som beskrevet tidligere, var tanken at brugerne ikke skulle kunne gennemføre handlen hvis forskellen mellem varer og mønter var for stor, men derimod skulle have point hvis de kun undveg en smule fra det korrekte resultat. Tanken bag dette var at undgå at straffe brugeren for meget, og stadig give point for at være tæt på. En grundig analyse af hvordan dette kunne påvirke læringen, ville dog være optimal at have, inden implementering.

» **Justerbart niveau**

- » Et justerbart niveau ville gøre det muligt for brugeren at skræddersy sin oplevelse, hvis de fandt den for svær. Et højere niveau skulle selvfølgelig afspejles i antallet af point brugerne får.

» **Responsivt niveau**

- » Niveaue i prototypen stiger lineært med antallet af point brugerne får. En optimering af dette ville være et responsivt niveau, der tilpassede sig alt efter hvordan brugeren svarede. Ved at se hvilke aspekter brugeren har problemer med, kan anmodningerne ændres, eller der kan give højere bonus for optimal gennemførsel.

» **Highscore**

- » En form for highscore kunne hjælpe med at motivere eleverne. Men for ikke at skabe en for stor skildring mellem “de dygtige” og “de mindre dygtige”, ville det være nødvendigt at overveje nøje hvordan dette kunne implementeres. Eksempelvis kunne det vise de bedste procentmæssige stigninger af niveau, eller dem der har klaret sig bedre end forventet.

» **Udfordringer**

- » En form for udfordringer inde i spillet. Som foreslået i brugertestene, så kunne man evt. fjerne nogle af mønterne. Andre muligheder kunne være: ekstra bonus hvis det blev løst korrekt inden for noget tid, eller at brugeren skulle gå efter at ramme et bestemt antal mønter.

» **Achievements**

- » Achievements er en motiverende faktor for mange spil. Ved at implementere den i spillet, kunne man belønne brugeren både for faglig og udforskende adfærd. Achievements kunne samtidig løse op for tilpasning af verdenen - eksempelvis kunne brugeren skifte varernes modeller og teksturer, låse op for andre lyde, og skifte markerings-farven.

Mulighederne er mange, og da prototypen bare er en enkelt case, er det vigtigt at holde det simpelt, for ikke at overvælde brugeren. Men hvis man udvider konceptet til en læringsplatform med flere cases, kan de fleste af forslagene også tilpasses, så de kan benyttes der.

Der er masser af plads til gode læringsspil i undervisningen, og ved at benytte de nyeste teknologier, kan man skabe spil der både har et stort læringspotentiale, og er underholdende for brugeren at benytte.

Anbefalinger

For yderlige interesse i læringsspil kan jeg anbefale *Good Video Games + Good Learning* af James Paul Gee, og *Mathematics education for a new era* af Keith Devlin - begge kan findes i litteraturlisten. Ydermere vil jeg anbefale *What video games have to teach us about learning and literacy*²⁷, der ikke bliver brugt som direkte litteratur, men er refereret til flere gange i den læste teori.

²⁷ Gee, J. (2007). *What video games have to teach us about learning and literacy*. New York: Palgrave Macmillan.

Appendix

Litteraturliste

- Adams, D. (1992). *Mostly harmless*. New York: Harmony Books.
- Bolter, J. og Grusin, R. (2000). *Remediation*. Cambridge, Mass: MIT Press.
- Csikszentmihalyi, M. (1991). *Flow*. New York: HarperPerennial.
- Devlin, K. (2011). *Mathematics education for a new era*. Natick, Mass.: A K Peters.
- Dourish, P. (2001). *Where the action is*. Cambridge, Mass.: MIT Press.
- Gee, J. (2013). *Good Video Games + Good Learning*. 2nd ed. New York: Peter Lang Publishing.

Bilag

Bachelorprojekt beskrivelse

Udvikling af virtual reality spil til læringsformål (*Development of a virtual reality learning game*)

Bachelorprojektet vil omhandle udviklingen af en virtual reality (VR) læringsspil-plattform, der skal kunne fungere på en (eller flere) VR-platforme (Google Cardboard, Oculus Rift, etc.). Udviklingen sker i et relevant udviklingsmiljø (såsom Unity, Unreal Engine, etc.), hvoraf der vil blive udarbejdet en prototype af læringsplatformen. Prototypen skal kunne vise minimum én case inden for det matematiske/naturvidenskabelige læringsområde²⁸.

Den færdige case skal indeholde en form for udfordring, der er relevant inden for det gældende fagområde - brugeren skal så kunne interagere med udfordringen ved hjælp af VR-miljøet (og eventuelle relevante redskaber). Formålet med platformen, og de tilhørende cases, er at hjælpe brugeren med at indleve sig i en case, hvor problemer kan visualiseres, og eventuelle fejl ikke har en reel betydning for brugeren.

Projektets humanistiske område vil hovedsageligt være baseret på brugercentreret interaktionsdesign, læringsteori og spilteori. Projektets tekniske område vil være baseret på programmering i et relevant udviklingsmiljø.

²⁸ Se <http://www.emu.dk/omraade/gsk-l%C3%A6rer>

Brugertests

Michael M

E-læringsudvikler ved HF & VUC FYN (aka Cardboard-Guru)

Jeg synes din tanke er god. Og spændende om ikke andet.

Nej det virker heller ikke på min telefon, altså magneten.

Det er jo kun en prototype som du skriver. Det er vigtigt at du tænker en eller andet start sekvens ind, der gør det meget meget let at forstå hvad man skal...

Jeg kan godt li din idé. Send endelig en ny version når du har noget der kan mere end bare markere... :)

Hannah D

6 år, 0. klasse, snart 1. klasse

Hvad synes du om musikken? *Det måtte gerne gå lidt hurtigere, eller nogen der synger*

Hvad synes du om idéen? *Det ville være lidt nemmere, hvis han kommer over til mønten, og så tager han mønten, f.eks 1 krone, så kunne han give den til damen, hvis der var en dame. Og måske kunne der også være et bord. Det kunne være sjovt hvis man kunne se sig selv, det kunne være sjovt hvis man kiggede ned og så havde man bare lige så tynde ben som manden.*

Tror du det kunne være sjovt at bruge til at lære? *Ja, det kunne det godt.*

Hvis du skulle bruge den i klassen, hvordan kunne du så bruge den? *Man kunne måske bruge den til at tage den med til legetøjsdag. Måske så kunne man lege inde i et spil i en butik.*

Teknologien, altså selve den man kigger i, hvad synes du om den? *Det er rigtig sjovt, måske kunne man lige gøre så der kunne sidde et bånd rundt, så den sad fast. Og så kan alle bruge den hvis man nu kan rykke stroppen.*

Var det sjovt at gå rundt og forestille sig at man var inde i spillet? *Ja, jeg kunne godt prøve at se hvis man gik hen, og hvis der var noget så kunne den lige en lyd (så man ikke går ind i væggen). Det måtte gerne være lidt mere virkeligt*

David D

Lærings- og Oplevelsesteknologi studerende, 6. semester, og forældre

Google Cardboard(GC): Jeg synes GC virker rigtig spændende. Det er lidt sjovt at man kan kigge rundt inde i den verden du har lavet. Jeg kan godt være lidt nervøs for det sociale aspekt af din idé når du bruger GC. Man er meget lukket inde i sin egen verden når man har

GC på. Kunne man måske udvide konceptet så det virker helt naturligt at give den videre eller deles om den? Måske kunne scenariet skifte så man fik af vide at nu er det køberens / sælgerens tur.

Din verden: Er rigtig fin! Jeg kan godt lide den måde du har lavet din verden på. Den virker lys, varm og indbydende.

Lidt finpudsning her og der og så tror jeg sagtens den kan bruges. Jeg føler mig i hvert fald inde i en verden der minder meget om noget jeg ville forvente i et Pixeline/"noget til den aldersgruppe" spil. Måske kunne man udvide den med bevægelse så man kan gå lidt rundt, men jeg ved ikke helt om GC kan det.

Handelskonceptet: Jeg synes idéen om en virtuel "købmand" virker ret sjov. Jeg savnede dog lidt visuelle spor i interfacet der fortæller mig, hvad jeg skal. Nu kan jeg ikke helt huske dit interface, men jeg mener ikke der var nogen tekst eller ikoner andet end de penge og varer der var. Måske kunne et ikon med et kasseapparat fortælle at her betaler man osv...

Til sidst: Jeg synes idéen er rigtig god, men jeg er lidt usikker på, hvad den giver til købmandslegen. Jeg tænker lidt at der mangler noget, som gør at man vil bruge den fremfor fysiske papirsedler og dåser.

Men uanset, hvad så virkede det til at Hannah synes det var sjovt, så måske er det bare en voksenbekymring :)

Martin K

Lærings- og Oplevelsesteknologi studerende, 6. semester

Jeg synes at selve konceptet af dit projekt er rigtigt godt. Jeg synes det er et rigtig godt og afslappende spil. Jeg kan godt lide det, at der er en tracker som holder øje med ens svar om de rigtige eller forkerte. I starten brugte jeg lidt lang tid på at finde ud af hvad man skulle gøre, med henholdvis tingene og pengene, i starten kunne jeg ikke lige finde ud af hvor meget det kostede i det hele osv. Men helt sikkert et fed spil når man først er inde i det med god potentiale. Det kunne være rigtig fedt, at nu man står i en bod og skal betjene kunder, at det er nogle nye mennesker der kommer frem hvergang de er blevet betjent, samtidig med at man eksempelvis ser et glas blive fyldt op med penge lidt efter lidt, når en kunde er blevet betjent

Trine T og Nicklas F

Lærings- og Oplevelsesteknologi studerende, 6. semester

Computeren bliver utrolig varm haha. Den arbejder på højtryk :-D

Synes ikke der behøves at være de ting, som han ikke skal købe ved manden. F.eks. 0* brød.

Måske nogle sorte kanter på de to grønne kasser så man kan se det er 3D.

Priserne kan måske sættes som nogle skilte foran varerne. Og så kan dér hvor prisen står nu, være antallet af varer man har valgt.

Måske et andet symbol til at fortryde og anden farve end grøn.

Fint med feedback på varerne når musen er over.

Fint at varerne roterer så der ikke er stille.

Måske tilføje 20kronen så alle mønterne er der.

Man kan vælge 1-kroner når man trykker på mønterne på pladen med mønter man har valgt.

Grineren med valg af mand. Han kunne måske selv også stå og rykke på sig. Måske gå hen mod "disken" og gå igen bagefter.

Måske noget visuel feedback på når man klikker på varerne.

Fint med baggrundsmusik.

Underlig lyd ved forkert.

Manden kan komme med en indkøbsseddel istedet for rød tekst.

En udvidelse kunne være at nogle af mønterne er væk, så man er tvunget til at betale med mønter, man måske ikke ville have brugt.

Det øverste tal ved de valgte mønter er gråt - er det med vilje?

Mht. streger på kasserne, som vi skrev først, mener vi, ligesom på varerne - det er godt!

Måske skulle mønterne ligne rigtige mønter.

Alberte P

Matematikudlært lærerstuderende

Synes virkelig det er godt fundet på med spillet, de ting, som jeg tænke over da jeg spillede var:

- Priser: kunne man evt. forestille sig en prisliste? Husk enheder (ex. kroner), således at der ikke er nogen tvivl om hvad tallene betyder

- Regneudtrykket der står på gulvet: Jeg tror at det bliver for svært for eleverne, jeg har ikke oplevet en 3. klasse der har stiftet bekendtskab med så stort et regneudtryk. Jeg tror at det ville være lettere for dem at regne:

| i en kasse | | i en kasse |
1 x (billede af ost) + 2x (billede af bog) osv.

| 2 x æble | + | 3 x ost | = 2 kr. + 4 kr.

eksempelvis :-) det tror jeg er mere forståeligt for dem :-D

og så et prisskilt, hvorpå de kunne finde priserne på de forskellige ting. For niveauet er egentlig passende, man skal noglegange bare kamuflere selve matematikken i det :-) ... de skal stadigvæk nok kunne koble billederne til matematik, men det går nok lidt, før de kan overskue store udtryk som disse :-)

Men ellers synes jeg det er super godt, for eleverne elsker at arbejde med IT, og det har det hverdagsselement som er en kæmpe hjælp for dem (hvilket også nævnes under modellering i FFM) :-)

Kode

Som beskrevet i afsnittet **Gennemgang af kodelystykker**, er det kun tre af kodefilerne der er at finde her. De andre er hentet eksternt, eller fylder kun en enkelt linje.

MainScript.cs

```
C:\Users\Public\Documents\Unity Projects\BAPCVersion\Assets\05 Scripts\MainScript.cs 1
1 using UnityEngine;
2 using System.Collections;
3 using System.Collections.Generic;
4 using System;
5
6 public class MainScript : MonoBehaviour
7 {
8
9     public int gameDifficulty = 1;
10    ArrayList objectList = new ArrayList();
11    public float appleVal = 1;
12    public float cheeseVal = 2;
13    public float breadVal = 3;
14    public float bookVal = 8;
15    public float coin1Val = 1;
16    public float coin2Val = 2;
17    public float coin5Val = 5;
18    public float coin10Val = 10;
19    float transactionVal = 0;
20
21    int appleCount = 0;
22    int cheeseCount = 0;
23    int breadCount = 0;
24    int bookCount = 0;
25
26    int coin1Count = 0;
27    int coin2Count = 0;
28    int coin5Count = 0;
29    int coin10Count = 0;
30
31    float transactionResult = 0;
32
33    public int points = 0;
34
35    GameObject appleText;
36    GameObject cheeseText;
37    GameObject breadText;
38    GameObject bookText;
39
40    GameObject coin1Text;
41    GameObject coin2Text;
42    GameObject coin5Text;
43    GameObject coin10Text;
44
45    GameObject calcText;
46    GameObject goodieCalcText;
47    GameObject coinCalcText;
48    GameObject optimalBonus;
49
50    public GameObject requestText;
51
52    float coinCalc;
53    float goodiesCalc;
54
55    int statTries = 0;
56    int statCorrect = 0;
57    int statWrong = 0;
58
59    double statusPercentage = 0;
60    int lowestAmountOfCoins = 10000;
61
62    int reqApple = 0;
63    int reqCheese = 0;
64    int reqBread = 0;
65    int reqBook = 0;
66    public int maxReq = 5;
67    bool correctAmount = false;
68
69    GameObject appleCalcText;
70    GameObject cheeseCalcText;
71    GameObject breadCalcText;
72    GameObject bookCalcText;
73
74    GameObject coin1CalcText;
```

C:\Users\Public\Documents\Unity Projects\BAPCVersion\Assets\05 Scripts\MainScript.cs

2

```
75  GameObject coin2CalcText;
76  GameObject coin5CalcText;
77  GameObject coin10CalcText;
78
79  void Start()
80  {
81      appleText = GameObject.Find("AppleText");
82      cheeseText = GameObject.Find("CheeseText");
83      breadText = GameObject.Find("BreadText");
84      bookText = GameObject.Find("BookText");
85
86      coin1Text = GameObject.Find("Coin1Text");
87      coin2Text = GameObject.Find("Coin2Text");
88      coin5Text = GameObject.Find("Coin5Text");
89      coin10Text = GameObject.Find("Coin10Text");
90
91      calcText = GameObject.Find("CalcText");
92      coinCalcText = GameObject.Find("CoinCalcText");
93      goodieCalcText = GameObject.Find("GoodieCalcText");
94
95      appleCalcText = GameObject.Find("applePrice");
96      cheeseCalcText = GameObject.Find("cheesePrice");
97      breadCalcText = GameObject.Find("breadPrice");
98      bookCalcText = GameObject.Find("bookPrice");
99
100     coin1CalcText = GameObject.Find("coin1Price");
101     coin2CalcText = GameObject.Find("coin2Price");
102     coin5CalcText = GameObject.Find("coin5Price");
103     coin10CalcText = GameObject.Find("coin10Price");
104
105     optimalBonus = GameObject.Find("bonusText");
106
107     addObjectToArray("");
108     generateRequest();
109 }
110
111 void Update()
112 {
113     if (Input.GetKey("escape"))
114     {
115         Debug.Log("Escape game");
116         Application.Quit();
117     }
118 }
119
120 void generateRequest()
121 {
122     reqApple = UnityEngine.Random.Range(0, maxReq + gameDiffuculty);
123     reqCheese = UnityEngine.Random.Range(0, maxReq + gameDiffuculty);
124     reqBread = UnityEngine.Random.Range(0, maxReq + gameDiffuculty);
125     reqBook = UnityEngine.Random.Range(0, maxReq - 2 + gameDiffuculty);
126
127     string requestString = "Hejsa!\n";
128     if (reqApple == 1) { requestString = requestString + reqApple + " æble\n"; } else if (reqApple != 0) { requestString = requestString + reqApple + " æbler\n"; }
129     if (reqCheese == 1) { requestString = requestString + reqCheese + " ost\n"; } else if (reqCheese != 0) { requestString = requestString + reqCheese + " oste\n"; }
130     if (reqBread != 0) { requestString = requestString + reqBread + " brød\n"; }
131     if (reqBook == 1) { requestString = requestString + reqBook + " bog\n"; } else if (reqBook != 0) { requestString = requestString + reqBook + " bøger\n"; }
132     Debug.Log("A " + reqApple + " C " + reqCheese + " B " + reqBread + " b " + reqBook);
133
134     requestText.GetComponent<TextMesh>().text = requestString;
135     if (reqApple + reqCheese + reqBread + reqBook == 0)
136     {
137         generateRequest();
138     }
139 }
140
141
142 public void addObjectToArray(string obj)
143 {
144     if (obj == "resetBtn")
145     {
```

```
C:\Users\Public\Documents\Unity Projects\BAPCVersion\Assets\05 Scripts\MainScript.cs 3
146     resetValues();
147 }
148 else if (obj == "verifyBtn")
149 {
150     if (coinCalc != 0 && goodiesCalc != 0)
151     {
152         if (reqApple == appleCount && reqCheese == cheeseCount && reqBread == breadCount &&
153             reqBook == bookCount)
154         {
155             correctAmount = true;
156             Debug.Log("Rigtig antal varer");
157         }
158         else
159         {
160             correctAmount = false;
161             Debug.Log("Forkert antal varer");
162         }
163         statTries++;
164         if (coinCalc == goodiesCalc && coinCalc != 0 && correctAmount == true)
165         {
166             //Prisen stemmer overens
167             AudioSource audio = GameObject.Find("sound_Verify").GetComponent<AudioSource>();
168             audio.Play();
169             float pointscalc = (10 * (1 + gameDiffuculty / 10));
170             Debug.Log(pointscalc);
171             int pointsgained = 100 + (int)pointscalc;
172             points = points + pointsgained;
173             statCorrect++;
174
175             List<int> coins = new List<int>();
176             List<int> amounts = new List<int>() { 1, 2, 5, 10 };
177             lowestAmountOfCoins = 10000;
178             Change(coins, amounts, 0, 0, Convert.ToInt32(coinCalc));
179             Debug.Log("Optimale antal mønter: " + lowestAmountOfCoins);
180             int usedCoins = coin1Count + coin2Count + coin5Count + coin10Count;
181             Debug.Log("Brugte antal mønter: " + usedCoins);
182
183             if (usedCoins == lowestAmountOfCoins)
184             {
185                 points = points + 50;
186                 int pointsgainedBonus = pointsgained + 50;
187                 optimalBonus.GetComponent<TextMesh>().text = "Du brugte " + usedCoins + " mønter
, det optimale antal er " + lowestAmountOfCoins + "\nDu får " + pointsgained + " point, plus 50
ekstra!\nI alt " + pointsgainedBonus + " point.";
188             }
189             else
190             {
191                 optimalBonus.GetComponent<TextMesh>().text = "Du brugte " + usedCoins + " mønter
, det optimale antal er " + lowestAmountOfCoins + "\nDu får " + pointsgained + " point, ingen ekstra
point.";
192             }
193             resetValues();
194             generateRequest();
195         }
196     }
197     else if (coinCalc != goodiesCalc)
198     {
199         //Prisen er helt ved siden af
200         AudioSource audio = GameObject.Find("sound_NotGood").GetComponent<AudioSource>();
201         audio.Play();
202         points = points - 10;
203         statWrong++;
204         resetValues();
205         generateRequest();
206         optimalBonus.GetComponent<TextMesh>().text = "Du svarede desværre forkert.\nIngen
ekstra point :-(";
207     }
208 }
209 if (points > 1000 * gameDiffuculty)
210 {
211     gameDiffuculty++;
212     string textString = optimalBonus.GetComponent<TextMesh>().text;
213 }
```

```
C:\Users\Public\Documents\Unity Projects\BAPCVersion\Assets\05 Scripts\MainScript.cs 4
214         optimalBonus.GetComponent<TextMesh>().text = textString + "\nDu steg et niveau! Du ✓
er nu på " + gameDifficulty;
    }
216
217     }
218
219     }
220     else
221     {
222         switch (obj)
223         {
224             // Coins
225             case "coin1":
226                 coin1Count++;
227                 break;
228             case "coin2":
229                 coin2Count++;
230                 break;
231             case "coin5":
232                 coin5Count++;
233                 break;
234             case "coin10":
235                 coin10Count++;
236                 break;
237
238             // Items
239             case "apple":
240                 appleCount++;
241                 break;
242             case "cheese":
243                 cheeseCount++;
244                 break;
245             case "bread":
246                 breadCount++;
247                 break;
248             case "book":
249                 bookCount++;
250                 break;
251             default:
252                 Debug.Log("Non value item");
253                 break;
254         }
255     }
256
257     appleText.GetComponent<TextMesh>().text = ": " + appleCount;
258     cheeseText.GetComponent<TextMesh>().text = ": " + cheeseCount;
259     breadText.GetComponent<TextMesh>().text = ": " + breadCount;
260     bookText.GetComponent<TextMesh>().text = ": " + bookCount;
261
262     coin1Text.GetComponent<TextMesh>().text = coin1Count + " :";
263     coin2Text.GetComponent<TextMesh>().text = coin2Count + " :";
264     coin5Text.GetComponent<TextMesh>().text = coin5Count + " :";
265     coin10Text.GetComponent<TextMesh>().text = coin10Count + " :";
266
267     coinCalc = coin1Count * coin1Val + coin2Count * coin2Val + coin5Count * coin5Val + coin10Count *
coin10Val;
268     goodiesCalc = appleCount * appleVal + cheeseCount * cheeseVal + breadCount * breadVal +
bookCount * bookVal;
269
270     transactionResult = (float)goodiesCalc - (float)coinCalc;
271     if (statTries != 0)
272     {
273         statusPercentage = (statCorrect * 1.0f / statTries) * 100f;
274         Debug.Log("S%: " + statusPercentage);
275         calcText.GetComponent<TextMesh>().text = "Points: " + points + "\nNiveau: " + gameDifficulty ✓
+ "\nForsøg: " + statTries + "\nKorrekte gæt: " + statCorrect + "\nForkerte gæt: " + statWrong + "\n
nTilfredshed: " + statusPercentage.ToString("0.00") + "%";
276     }
277     else
278     {
279         calcText.GetComponent<TextMesh>().text = "Points: 0\nNiveau: 1\nForsøg: 0\nKorrekte gæt: 0\
nForkerte gæt: 0\nTilfredshed: x.xx%";
280     }
281
```

C:\Users\Public\Documents\Unity Projects\BAPCVersion\Assets\05 Scripts\MainScript.cs

5

```
282     appleCalcText.GetComponent<TextMesh>().text = appleCount + " * " + appleVal + " kr.";
283     cheeseCalcText.GetComponent<TextMesh>().text = cheeseCount + " * " + cheeseVal + " kr.";
284     breadCalcText.GetComponent<TextMesh>().text = breadCount + " * " + breadVal + " kr.";
285     bookCalcText.GetComponent<TextMesh>().text = bookCount + " * " + bookVal + " kr.";
286
287     coin1CalcText.GetComponent<TextMesh>().text = coin1Count + " * " + coin1Val + " kr.";
288     coin2CalcText.GetComponent<TextMesh>().text = coin2Count + " * " + coin2Val + " kr.";
289     coin5CalcText.GetComponent<TextMesh>().text = coin5Count + " * " + coin5Val + " kr.";
290     coin10CalcText.GetComponent<TextMesh>().text = coin10Count + " * " + coin10Val + " kr.";
291     Debug.Log(transactionResult);
292 }
293
294 void calculateValue()
295 {
296     transactionVal = 0;
297     foreach (string obj in objectList)
298     {
299         switch (obj)
300         {
301             // Coins
302             case "coin1":
303                 transactionVal = transactionVal + coin1Val;
304                 break;
305             case "coin2":
306                 transactionVal = transactionVal + coin2Val;
307                 break;
308             case "coin5":
309                 transactionVal = transactionVal + coin5Val;
310                 break;
311             case "coin10":
312                 transactionVal = transactionVal + coin10Val;
313                 break;
314
315             // Items
316             case "apple":
317                 transactionVal = transactionVal - appleVal;
318                 break;
319             case "cheese":
320                 transactionVal = transactionVal - cheeseVal;
321                 break;
322             case "bread":
323                 transactionVal = transactionVal - breadVal;
324                 break;
325             case "book":
326                 transactionVal = transactionVal - bookVal;
327                 break;
328         }
329     }
330     Debug.Log("Value: " + transactionVal);
331 }
332
333 void resetValues()
334 {
335     appleCount = 0;
336     cheeseCount = 0;
337     breadCount = 0;
338     bookCount = 0;
339     coin1Count = 0;
340     coin2Count = 0;
341     coin5Count = 0;
342     coin10Count = 0;
343 }
344
345 // Følgende kode er modificeret fra http://www.dotnetperls.com/change
346 void Change(List<int> coins, List<int> amounts, int highest, int sum, int goal)
347 {
348     if (sum == goal)
349     {
350         if (coins.Count < lowestAmountOfCoins)
351         {
352             lowestAmountOfCoins = coins.Count;
353         }
354     }
355     if (sum > goal)
```

C:\Users\Public\Documents\Unity Projects\BAPCVersion\Assets\05 Scripts\MainScript.cs

6

```
356     {
357         return;
358     }
359     foreach (int value in amounts)
360     {
361         if (value >= highest)
362         {
363             List<int> copy = new List<int>(coins);
364             copy.Add(value);
365             Change(copy, amounts, value, sum + value, goal);
366         }
367     }
368 }
369 }
370
371 }
372
```

ObjectHandler.cs

```
C:\Users\Public\Documents\Unity Projects\BAPCVersion\Assets\05 Scripts\ObjectHandler.cs 1
1 using UnityEngine;
2 using System.Collections;
3
4 public class ObjectHandler : MonoBehaviour {
5     public bool _selected = false;
6     Material defMat;
7     Material whiteDefMat;
8     Color selectCol = Color.green;
9     public Color defColor = Color.white;
10    public string type = "";
11
12    // Use this for initialization
13    void Start () {
14        defMat = this.GetComponent<Renderer>().material;
15        whiteDefMat = defMat;
16        whiteDefMat.SetColor("_Color", Color.white);
17    }
18
19    public float timeLeft = 1f;
20
21    // Update is called once per frame
22    void Update () {
23
24        if (_selected == true)
25        {
26            Material material = defMat;
27            material.SetColor("_Color", selectCol);
28            this.GetComponent<Renderer>().material = material;
29        }
30        else
31        {
32            if (timeLeft < 0)
33            {
34
35                this.GetComponent<Renderer>().material.SetColor("_Color", defColor);
36            }
37            else
38            {
39                timeLeft -= Time.deltaTime;
40            }
41        }
42    }
43 }
44 }
```

RaycastPlayer.cs

```
C:\Users\Public\Documents\Unity Projects\BAPCVersion\Assets\05 Scripts\RaycastPlayer.cs 1
1 using UnityEngine;
2 using System.Collections;
3
4 public class RaycastPlayer : MonoBehaviour {
5     float x ;
6     float y ;
7     float z ;
8     bool isHit = false;
9     GameObject temporaryObject = null;
10    GameObject mainScript;
11
12    void Start () {
13        x = this.transform.position.x;
14        y = this.transform.position.y;
15        z = this.transform.position.z;
16        mainScript = GameObject.FindWithTag("GameController");
17    }
18
19
20    void Update (){
21        x = this.transform.position.x;
22        y = this.transform.position.y+(1/10);
23        z = this.transform.position.z;
24        Vector3 vect = new Vector3 (x,y,z);
25        Debug.DrawRay(vect, transform.forward*2);
26        Ray ray = new Ray(vect, transform.forward*2);
27        RaycastHit hit;
28
29        if (Physics.Raycast(ray, out hit))
30        {
31            if (hit.collider.tag == "selectableObj")
32            {
33                isHit = true;
34                hit.collider.gameObject.GetComponent<ObjectHandler>()._selected = true;
35                hit.collider.gameObject.GetComponent<ObjectHandler>().timeLeft = 0.0f;
36                temporaryObject = hit.collider.gameObject;
37                if (Input.GetMouseButtonDown(0))
38                {
39                    string nameOfObject = temporaryObject.GetComponent<ObjectHandler>().type;
40                    mainScript.GetComponent<MainScript>().addObjectToArray(nameOfObject);
41                    AudioSource audio = temporaryObject.GetComponent<AudioSource>();
42                    audio.Play();
43                }
44            }
45            else if (temporaryObject != null)
46            {
47                temporaryObject.GetComponent<ObjectHandler>()._selected = false;
48                isHit = false;
49            }
50        }
51        else if (temporaryObject != null)
52        {
53            temporaryObject.GetComponent<ObjectHandler>()._selected = false;
54            isHit = false;
55        }
56    }
57
58    public void clickingSomeway()
59    {
60        if (isHit == true)
61        {
62            string nameOfObject = temporaryObject.GetComponent<ObjectHandler>().type;
63            mainScript.GetComponent<MainScript>().addObjectToArray(nameOfObject);
64            AudioSource audio = temporaryObject.GetComponent<AudioSource>();
65            audio.Play();
66        }
67    }
68 }
69
```

Anerkendelser - billeder og figurer

Billeder fra flickr er brugt under Creative Commons 2.0 licensen²⁹

Forsidebillede: Screenshot

Billede 1: <https://www.flickr.com/photos/44124339301@N01/25613941650>

Billede 2, 3, 4: Skitser

Billede 5: <https://www.flickr.com/photos/22974618@N00/14530775542>

Billede 6: <https://www.flickr.com/photos/121483302@N02/15380631096/>

Billede 7: <https://www.flickr.com/photos/39824398@N06/27073609552>

Billede 8, 9, 10, 11, 12: Screenshots

Billede 13, 14: Billeder taget ved brugertest

Billede 15, 16: Screenshots

Billede 17: Billede taget ved brugertest

Billede 18, 19, 20, 21, 22: Screenshot

Figur 1: Public domain fra Wikimedia Commons

Anerkendelser - materiale til software

Lyd af *Morgantj*: <http://www.freesound.org/people/morgantj/sounds/59628/>

Lyd af *Jobro*: <http://www.freesound.org/people/jobro/sounds/180894/>

Lyd af *JustinBW*: <http://www.freesound.org/people/JustinBW/sounds/79287/>

Lyd af *broumbroum*: <http://www.freesound.org/people/broumbroum/sounds/50561/>

Lyd skabt af *as3sfxr*-software: <http://www.superflashbros.net/as3sfxr/>

Lyd skabt af *Bfr*-software: <http://www.bfxr.net/>

Lyd skabt af *ChipTone*-software: <http://sfbgames.com/chiptone/>

Lyd af *Nihilore*: <http://nihilore.bandcamp.com/track/alone-by-the-fire>

Skyboxes af *70:30*: <https://www.assetstore.unity3d.com/en/#!/content/22252>

Objekter af *Dungeon Masters*: <https://www.assetstore.unity3d.com/en/#!/content/9994>

Træer af *Tom Poon*: <https://www.assetstore.unity3d.com/en/#!/content/49916>

Materiale af *Integrity Software & Games*:

<https://www.assetstore.unity3d.com/en/#!/content/34607>

Magnetfunktion fra *andrewnoske.com/wiki*: http://www.andrewnoske.com/wiki/Unity_-_Detecting_Google_Cardboard_Click

²⁹ <https://creativecommons.org/licenses/by-sa/2.0/>