

Læringsteknologi 2

# WE ARE IN SPACE

4. Semesterprojekt:

## We Are In Space

Udarbejdet af:

**Anthøn Pedersen, Caroline Hansen,  
Emil Rimer og Rasmus Ploug**



## Abstract

Virtual reality is a relevant tool with the potential of being a template for learning technology and applications. This project aims at exploring how it is possible to complement the physics topic in the Danish elementary school grade 7-9 with a learning technology. It will also explore the potential of substituting some of the academic reading in class with more dynamic ways of learning. In addition, this project focuses on the theme of astronomy and our solar system in coherence with the læseplan for fysik/kemi 7-9.

Structured usability-tests have been conducted in partnership with a physics teacher from Nymarkskolen in Svendborg. These usability-tests were prepared and conducted in teams of 4 test students over the course of two visits covering the entire participant class twice. In addition, the partnered physics teacher was also interviewed, thus gaining a professional perspective on the curriculum and teaching structure in said specific class.

Learning principles and design in this project is based on relevant theory from *Læringsteknologi 2*. This project uses supported claims on why and how it is possible to optimize the students' learning abilities using this learning technology.

Our results show a positive learning experience from the test students. This result is derived from a theory-based design on learning abilities in correlation to the conducted usability tests. The overall usability-tests result shows a positive attitude towards the application with an above medium being able to see themselves using this technology as a supplement to their physics class. The results also indicated a more engaged learning environment for especially dyslexic students. Hence the finished prototype proved that it is possible to positively boost learning experiences within a class using a VR learning technology.

## Abstract (Dansk)

*We Are In Space* tager eleverne i folkeskolens udskoling på en rejse gennem solsystemet. Udviklet som VR-applikation i Unity har eleverne muligheden for at finde informationer om solsystemets planeter. Dette læringsmiddel er udviklet i samarbejde med en fysiklærer fra Nymarkskolen og giver eleverne en dynamisk oplevelse af opdagelse og teori, som supplement til fysikundervisningen. Hertil er det tænkt at eleverne efter brug af applikationen kan udarbejde præsentationer om hvad de har fundet frem til på hver deres rejse gennem

solsystemet. Hele projektet tager udgangspunkt i brugertest og relevant teori fra uddannelsen Spiludvikling og Læringsteknologi på SDU.

## Indledning

De naturvidenskabelige fag i folkeskolen er ikke nemt for alle elever. Den gode kombination af svær teori og matematik er som reelt opbygget af praktiske forsøg og læsning. Derfor vil vi gerne designe en læremiddel, som skal bidrage til Fysikfagets faglige undervisning i folkeskolens udskoling.

I samarbejde med en fysiklærer på Nymarkskolen i Svendborg har vi fået bekræftet, at flere elever finder læsningen om solsystemet kedsommeligt. Særligt de ordblinde elever, som ellers er matematisk dygtige, finder problemer med den læselige teori.

Derfor har vi udviklet en applikation i virtual reality, som tager eleverne med på en opdagelse i rummet. Den ellers boglige teori er gjort mere dynamisk, nu hvor eleverne selv skal undersøge og finde relevant teori. Vores endelige prototype består af et rumskib, hvori eleven kan udforske rummet ved hjælp af knapper, håndtag og skærme. Eleven kan udføre missioner, baseret på teori fra pensum.

Et link til video af projektet kan ses i Bilag 5.

## Organisering af projekt

Organiseringen af dette projekt er ret ligetil. Vi har fra sidste semestre erfaret, at en god tilgang til organisering af større projekter er en simpel, men effektiv struktur, der bygger på god kommunikation og individuelt ansvar. Vi har derfor holdt et fast ugentligt møde á 4 timer, hvor vi gennemgår vores iterative design og diskuterer evt. fremtidige ændringer til produktet. De specifikke detaljer til dette møde er uddybet i Fremgangsmetode. Derudover fordelte vi ugentlige arbejdsopgaver, hvor resultatet løbende blev diskuteret i gruppen og evalueret til hvert møde. Med denne metode, kunne vi dække op til fire opgaver samtidig. Dog var det vigtigt, at det individuelle arbejde stemte overens med forventningerne fra mødet.

Udover arbejdsfordeling, fungerede møderne også som planlægning til vores brugertests. Dette semester har vi haft særligt fokus på at organisere og planlægge vores brugertests, for at kunne få et så konstruktivt udbytte af feedback som muligt.

## Problemformulering

I dette projekt vil vi gerne undersøge, hvordan man kan supplere den faglige fysikundervisning i folkeskolen med en læringsteknologi i virtual reality. Læringsteknologien skal fokusere på astronomi med et nyt aktivt aspekt, som kan erstatte faglig læsning.

I forbindelse med dette, vil vi også inddrage relevant teori for området og implementere det i et design til at fremme læring. Til slut vil vi gerne udforske, hvordan man i tæt kontakt med en målgruppe kan udvikle et produkt til deres behov baseret på brugertests og feedback.

## Indholdsfortegnelse

Abstract .....	1
Abstract (Dansk) .....	1
Indledning .....	2
Organisering af projekt .....	2
Problemformulering .....	3
Baggrund .....	6
Teori .....	7
Introduktion til teoriafsnit .....	7
Good Video Games as Learning Machines .....	7
Co-design .....	8
Identity .....	9
Manipulation and Distributed Knowledge .....	9
Information ‘just in time’ and ‘on demand’ .....	10
Fish Tanks .....	10
Five Stage Model of Skill Acquisition .....	11
A Dynamic Theory of Organizational Knowledge Creation .....	13
Den individuelle elev .....	13
Model for konvertering af viden .....	14
At styrke tavs viden med dialog .....	15
Fejllæring .....	16
Teknologisk platform .....	17
Related Work .....	17
Hvordan vi bygger videre på State of Art .....	19
Metode .....	19
Fremgangsmetode .....	20
Undersøgelser .....	21

1. Iteration - Konceptudvikling.....	22
2. Iteration - Papirprototype.....	24
3. Iteration - Vertical Slice.....	25
4. Iteration - Færdig Prototype.....	28
Teknologisk Udvikling .....	31
Arbejde med VR .....	31
Håndtering af data.....	32
Datamonitor .....	34
Missionsmonitor .....	36
Knapper og håndtag .....	37
Knap.....	38
Håndtag.....	40
Diskussion.....	42
Om samarbejde med folkeskole.....	42
Om risiko for bias .....	43
Om VR i undervisningen .....	44
Om realistisk visualisering af rummet .....	44
Om arbejde med VR .....	45
Opsamling og Konklusion .....	46
Perspektivering .....	46
Referencer .....	48
Bilagsliste.....	48
Emnefordeling.....	49

## Baggrund

Baggrunden for den faglige viden i vores projekt er baseret på Fysikpensum i 7 – 9 klasse i den danske folkeskole. Før vi kontaktede en skole, undersøgte vi hvorvidt deres pensum matchede vores grundlæggende projekt idé. Derfor hentede vi dokumentet *Læseplan for faget fysik/kemi*, fra Undervisningsministeriet (Undervisningsministeriet, 2018), som senest var opdateret maj 2018.

I dokumentet er der et emne, som hedder *Jorden og Universet*. Dette emne er delt op i fire kategorier:

- *Undersøgelse*
- *Modellering*
- *Perspektivering*
- *Kommunikation*

Hver af disse kategorier beskriver hvordan underviseren skal planlægge sin undervisning. Vi besluttede os daværende for, at vi ville kunne bidrage indenfor kategorierne: *undersøgelse* og *modellering*.

I kategorien *Undersøgelse* ville vi kunne bidrage inden for – ” *Senere fokuserer undervisningen på Jordens systemer, hvor elevernes egne og andres målinger af temperaturer og tryk er centrale for at forstå bevægelser, herunder vand i kredsløb, vindretning og vindhastighed. Eleverne stifter bekendtskab med begrebet energistrømme og skal selv opsamle atmosfæriske data ved hjælp af relevant udstyr, herunder elektronisk dataopsamling i form af fx dataloggere*” – (Undervisningsministeriet, 2018, p. 5)

I kategorien *Modellering* ville vi kunne bidrage indenfor – ” *Trinforløbet tager udgangspunkt i elevernes forståelse af Jordens bevægelse, rotation, hældningsakse og atmosfære. Eleverne skal kunne udvælge og anvende modeller til beskrivelse af Solsystemet, herunder digitale simuleringer eller fysiske planetmodeller. Eleverne skal kende til Solsystemets placering i Mælkevejen og Universet og skal ved brug af digitale medier kunne navigere på stjernehimlen*” – (Undervisningsministeriet, 2018, p. 7)

Med en bekræftelse i, at vores idé matchede nogle grundlæggende, tematikker i folkeskolens læseplan, kunne vi nu kontakte og indgå dialog med en fysiklærer om projektet. Her var vi så heldige at få startet et samarbejde med Johnny Steen, som er fysiklærer på Nymarkskolen.

Vores første besøg med ham er beskrevet senere i rapporten under 3. *Iteration - Vertical Slice*. Men ift. dette afsnit af rapporten gennemgår vi vores tanker omkring læseplanen.

Johnny bekræftede os i, at ikke bare ville vi kunne supplere fysiktimerne, men vores projekt ville også kunne erstatte nogle af de mere jævne arbejdsprocesser for eleverne. Bl.a. ville vi kunne erstatte noget af deres læsestof med aktive opgaver. Derfor ville den empiri af deres fysik pensum, som vores applikation ville erstatte eller supplere, være teorien omkring solsystemets planeter. Ud fra de tidligere kategorier af modellering og undersøgelse, ville Johnny gerne have ekstra fokus på, at eleverne først og fremmest:

- Kunne undersøge solsystemets planeter og forholdet mellem planeternes opbygning og dimensioner.
- Kunne opdage information og senere hen perspektivere deres viden videre til andre elever.

Som konklusion på dette afsnit, har vi ift. baggrund gjort følgende. Undersøgt de faglige mål gennem undervisningsministeriets læseplan for fysik/kemi i 7-9 klasse. Her har vi bekræftet at vores projekt vil være relevant inden for kategorierne *Undersøgelse* og *Modellering*. Vi har herefter i dialog med en fysiklærer, indskærpet kategorierne. Vores applikation ville derfor kunne bidrage positivt med relevant arbejde og teori for eleverne, ud fra Johnnys fortolkning af læseplanen for Fysik/kemi. En del af den faglige viden er dog baseret på simple placeholders (Universe Today, 2022), som sagtens ville kunne uddybes med videre arbejde af vores applikation.

## Teori

### Introduktion til teoriafsnit

Teoriafsnittet i denne rapport er baseret på anvendt teori fra 4. semesters undervisning i læringsteknologi. Formålet for dette afsnit er at give en gennemgående forklaring af, hvordan vi har anvendt og tolket relevant teoretisk viden inden for vores problemstilling. Teoriafsnittet vil gennemgå flere forskellige læringsteorier fra semesteret, bl.a. brugen af tavs og eksplicit viden, implementering af læring i videospil, og fejllæring.

### Good Video Games as Learning Machines

Vi har valgt at benytte teori fra *Learning by Design: Good Video Games as Learning Machines* af James Paul Gee (Gee, 2005), med fokus på hvordan man bedst designer lærerige og engagerende computerspil. Gee introducerer sin teori med påstanden, at børn og unge generelt



set ikke nyder at løse svære og komplekse opgaver. For at få dem til at løse disse opgaver er man derfor nødsaget til enten at tvinge børnene til læring eller at nedsætte kompleksiteten af opgaverne. I begge tilfælde findes der dog markante ulemper. Tvang er en dårlig katalysator for læring, og nedsat kompleksitet kan formindske kvaliteten af den resulterende viden. Gee advokerer derfor for brugen af diverse spil- og læringskoncepter, bl.a. i undervisningen, for potentielt at øge motivationen for de lærende til at løse disse komplekse opgaver. Spildesigneres opgave er derfor at gøre selve læringsprocessen til den implicitte motivationsfaktor for brugeren. Et godt citat fra artiklen er som følgende:

*“Good game designers are practical theoreticians of learning, since what makes games deep is that players are exercising their learning muscles, though often without knowing it and without having to pay overt attention to the matter.” (Gee, 2005, p. 5)*

Altså er det designeres opgave at inkludere diverse læringsprincipper i spillet, uden at gøre det indlysende for spilleren. Gee nævner 12 læringsprincipper. Disse læringsprincipper er fordelt over 3 sektioner: *Empowered Learners*, *Problem Solving* og *Understandig*. Bemærk, at vi ikke har inkluderet alle af Gee's principper. Dette skyldes bl.a. at vores læringsteknologi ikke direkte er blevet udviklet som et spil, men i stedet som en gamificeret læringsplatform. Resten af dette afsnit vil derfor fokusere på de læringsprincipper, som er relevante for vores læringsteknologi.

#### Co-design

Under aktiv læring er det vigtigt, at den lærende føler sig som en handlekraftig agent, frem for en passiv modtager af viden. Dette kan være svært i klasseundervisning, hvor underviseren er den primære aktuator for, at læring finder sted. Computerspil løser dette problem, ved at placere den lærende i et interaktivt miljø, hvor de er tilskyndet til selv at handle. I forbindelse med dette har vi sørget for at designe VR-oplevelsen for vores læringsteknologi således, at brugeren selv får lov til at interagere med verden omkring en, frem for at få fortalt hvad der skal gøres. Eksempelvis har vi inkluderet forskellige funktionaliteter, bl.a. at man kan rejse til forskellige planeter, skære planeterne op, samt vise forskellige informationer omkring planeten. Brugeren kan altså selv vælge hvad de vil gøre, alt afhængigt af hvad deres intentioner er. Spillets missionssystem skal derfor ses som et vejledende springbræt til udforskningen af resten af vores læringsunivers. Således vil hver brugers oplevelse være individuel.

## Identity

Læring er, ifølge Gee, mest effektiv når man kan fordybe sig i et emne over længere tid. Den mest optimale måde hvorpå man kan lade denne fordybelse finde sted, er at give den lærende mulighed for at identificere sig i en relevant rolle. Eksempelvis vil en elev have lettere ved at investere energi i et videnskabsprojekt, hvis eleven identificerer sig selv som en videnskabsmand. Dette princip strækker sig dog ikke kun over undervisning i folkeskolen, men også hvad angår ens arbejde og hobbyer. Mange computerspil har også implementeret dette princip, ved at give spilleren mulighed for at projektere deres egne fantasier, lyster og fornøjelser på spilkarakteren. Således kan spilleren identificere sig med karakteren og dermed få en dybere forståelse for spillets tema, verden, budskab, osv.

Vores læringsteknologi gør brug af dette identitetsprincip, bl.a. ved at introducere brugeren som en astronaut, som skal på en rummission. Dette skaber en mere autentisk oplevelse for brugeren når de skal udforske universet, og er desuden med til at skabe 'immersion'. Målet er at gøre den relevante viden så fordøjelig så mulig for brugeren. Vi har altså skabt en sammenhæng mellem læringsteknologiens kontekst og brugerens identitet, for dermed at gøre hele læringsprocessen mere effektiv.

## Manipulation and Distributed Knowledge

Kognitiv forskning foreslår, at menneskers opfattelse og deres handlinger er dybt indbyrdes forbundet (Gee, 2005, p. 8). Dette vil også være sandt ved handlinger, som ikke sker i personens umiddelbare nærhed, f.eks. ved styring af en fjernstyret bil på afstand. Generelt set føler mennesker sig magtfulde ved manipulering af komplekse værktøjer udenfor deres direkte omfangsareal. Dette er især sandt når der er tale om computerspil, skønt dette primært sker virtuelt. I mange spil kan spilleren manøvrere igennem verden via spilkarakteren, samt interagere og manipulere med objekter i verden for at udføre spillets mål. VR har en arvelig fordel hvad dette princip angår, idet man, i modsætning til normale computerspil, får lov til at fordybe sig i spillets omgivelser via 'immersion'. Vores læringsteknologi implementerer forskellige måder, hvorpå brugeren kan interagere og manipulere med omgivelserne på. For eksempel kan spilleren interagere med diverse knapper og håndtag for bl.a. at ændre planet eller stjernehimmel. Derudover kunne brugeren, i en tidligere iteration af læringsteknologien, styre rumskibet. Disse interaktioner er med til at skabe en forbindelse mellem brugerens fysiske krop og spillets virtuelle avatar. Med andre ord tillader vi den lærende at udvide deres sind ind i en ny kontekst, for at skabe et mere effektivt læringsmiljø.

### Information 'just in time' and 'on demand'

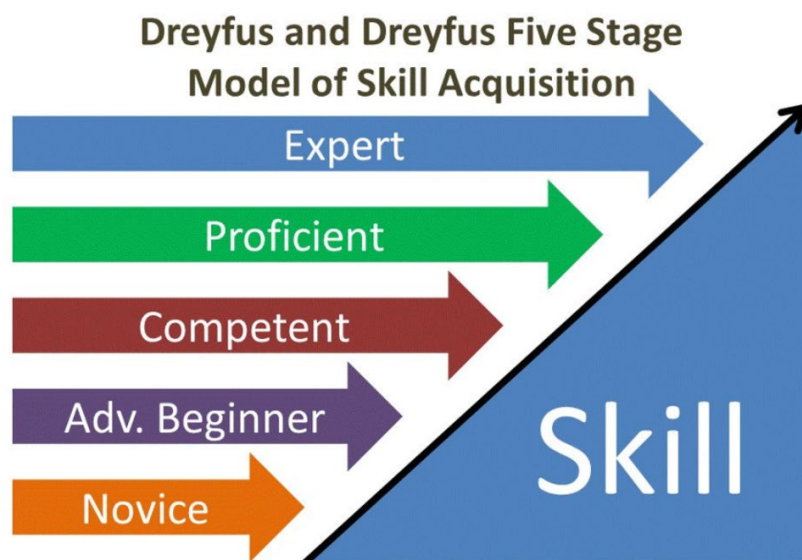
Mennesker kan have svært ved at bearbejde større mængder af informationer på én gang ude af kontekst. Den mest optimale måde at bruge specifik viden i praksis på, er derfor enten at få de relevante informationer formidlet på én gang (just in time), eller selv at anskaffe informationerne, når der er behov for det (on demand). Computerspil er mestre i begge aspekter. Mange spil implementerer eksempelvis små tutorials på specifikke punkter undervejs, alt afhængigt af hvornår bestemte informationer bliver relevante for spilleren. Samtidigt kan spilleren ofte frit tilgængeligt få adgang til tidligere informationer de har lært hen ad vejen, hvis dette skulle være nødvendigt. Vores læringsteknologi gør især brug af 'on demand' princippet. For at gennemføre missionerne i vores læringsteknologi, skal brugeren besvare forskellige spørgsmål i form af en multiple-choice quiz. Alle informationerne kan findes ved at rejse til den specifikke planet og dernæst søge efter informationerne via en af skærmene. Det vil sige, at eleverne gradvist finder den relevante viden når det er nødvendigt for at klare en mission, frem for at få informationerne formidlet ude af kontekst.

### Fish Tanks

Fish-tank-princippet forklarer en måde, hvorpå man kan opbygge en forenklet struktur af et ellers relativt komplekst økosystem. Ved at fokusere på de mest essentielle interaktioner og variable i et specifikt system, kan den lærende lettere forstå og tilegne sig viden om emnet, uden at det bliver unødvendigt kompliceret. Vores læringsteknologi kan ses som værende et lille økosystem for sig selv. For eksempel har vi valgt at fokusere på nogle meget håndgribelige aspekter af vores solsystem, for ikke at overvælde brugerne med ligegyldige informationer. Bl.a. fokuserer vi på at formidle en mere overordnet forståelse af solsystemets opbyggelse (antal planeter, planeternes placering i forhold til solen, planeternes størrelse i forhold til hinanden, osv.). Således bombarderer vi ikke brugeren med viden som, i kontekst af deres læring, ikke umiddelbart er relevant.

## Five Stage Model of Skill Acquisition

Dette teoriarsnit tager udgangspunkt i artiklen *Expertise in Real World Context* af Hubert L. Dreyfus og Stuart E. Dreyfus (Dreyfus & Dreyfus, 2005). Artiklen omhandler hovedsageligt *The Five Stage Model of Skill Acquisition*, se Figur 1. Denne model er inddelt i fem niveauer: *Novice*, *Advanced Beginner*, *Competent*, *Proficiency* og *Expert*.



Figur 1 - Five Stage Model of Skill Acquisition

Hver af disse niveauer refererer til et specifikt punkt i en persons individuelle færdighedsudvikling, altså hvor kompetent personen er til den specifikke færdighed. De fleste børn lærer bestemte færdigheder i en ung alder via 'trial and error' og imitation, oftest ubevidst. Som voksen er man dog i de fleste tilfælde nødsaget til bevidst at udvikle nye færdigheder gennem planlagte instrukser. Denne proces starter på 'Novice'-niveau, hvorefter man kan arbejde sig igennem niveauerne til et potentielt 'Expert'-niveau. Hver af disse niveauer er karakteriseret ud fra nogle bestemte aspekter:

1. *Novice*: Som 'Novice' har man blot en basal forståelse for de regler og principper, som færdigheden indebærer. Instruktionsprocessen går derfor primært ud på at operere og imitere disse principper i praksis. En nybegynder til skak vil f.eks. lære de numeriske værdier for hver unik skakbrik, samt hvordan hver skakbrik kan bevæge sig på skakbrættet.
2. *Advanced Beginner*: I takt med at nybegynderen får mere erfaring, vil personen udvikle en bedre forståelse for den relevante kontekst, og samtidig få indblik i nye meningsfulde perspektiver. Derudover får personen en bedre forståelse for specifikke procedurer. Eksempelvis vil en avanceret nybegynder til skak opdage forskellige strategiske

svagheder og styrker i modstanderes skakbrikker, selvom han/hun måske ikke har fuld forståelse for, hvorfor dette specifikt er tilfældet.

3. *Competence*: I dette niveau vil mængden af potentielle procedurer og elementer, som personen kan genkende, blive uoverskuelige. Dette niveau er præget af et øget emotionelt engagement. Dvs. at færdigheden bl.a. kan vække frustrationer og udmattelse, men også glæde og eufori. For at bearbejde denne proces og gøre beslutningstagningen lettere, vælger mange derfor at indskrænke antallet af muligheder til en mere kompakt mængde. Skakspilleren vil bl.a. fokusere på bestemte taktiker, og vil på forhånd begynde at tænke fremad for potentielt at kunne forudsige modstanderes næste træk.
4. *Proficiency*: Dette punkt i færdighedsudviklingen er præget af et fald i informationsindtag og en stigning i personlig involvering. Den lærende har altså i bund og grund lært alle essentielle aspekter og kan derfra fokusere på at bruge og finpudse disse aspekter i praksis. Således vil positive følelsesmæssige oplevelser styrke succesfulde responser, og omvendt. Skakmesteren vil kunne genkende et stort repertoire af muligheder for sit næste træk alt afhængigt af skakbrikkernes positioner, og kan dernæst bestemme sig for det træk, som vil give det bedste udkom. Med andre ord er der tale om en mere intuitiv tankeproces.
5. *Expert*: Som ekspert er der tale om en internalisering af færdighedens aspekter. Eksperten har altså en så god fundamental forståelse for aspekterne, at han/ hun ikke umiddelbart ved hvordan dette skal videreformidles til andre. Man ved at det er sådan det fungerer, men ved ikke hvordan man kan forklare hvorfor dette er tilfældet. En ekspert i skak kan for det meste lave lynhurtige træk ved hjælp af intuition alene.

Idet fysikfaget bliver introduceret for folkeskoleelever i 7. klassetrin (Undervisningsmisteriet, 2018), kan det forventes at færdighedsniveauet for eleverne starter på 'Novice', og undervisningens pensum vil derfor reflektere dette. Målet for underviseren er altså at give et grundlæggende indblik i universets opbygning, samt at introducere de fundamentale principper og regler indenfor emnet, for potentielt at lade eleverne opnå 'Advanced Beginner'-niveau når de kommer i 8.-9. klasse. Dette indebærer bl.a. forståelse for planeternes kontekst i solsystemet og en bedre grundlæggende forståelse for astrofysik. I optimale sammenhæng kan enkelte elever i de højstes klassetrin opnå 'Competent'-niveau. 'Proficiency'- og 'Expert'-niveau forventes ikke at blive opnået, idet der er tale om undervisning på folkeskoleniveau. En væsentlig faktor som skal pointeres, er dog at vi ikke har undersøgt, hvorvidt dette rent faktisk

er sandt. Den realistisk forventning for vores læringsteknologi er derfor, at eleverne kan opnå 'Advanced beginner'-niveau.

### A Dynamic Theory of Organizational Knowledge Creation

Som teori til dette projekt har vi anvendt artiklen *A Dynamic Theory of Organizational Knowledge Creation* (Nonaka, 1994). Artiklen, som er udarbejdet af Ikujiro Nonaka omhandler et paradigme, til hvordan viden organiseres og skabes i et samarbejde mellem tavs og eksplicit viden. Artiklen beskriver også en konceptualisering af dette paradigme i forskellen mellem læring for individer, grupper og organisationer. Vores projekt anvender teorien fra denne artikel til først og fremmest at definere tavs og eksplicit viden og derefter opbygge nogle af vores læringsteknologis grundlæggende principper ud fra dette videns-mønster. Disse grundlæggende principper er elevernes individuelle interaktion med applikationen og den efterfølgende mulighed for dialog med andre elever. Derfor er resultatet af disse principper i vores læringsteknologi med fokus på behandling og konceptualisering af viden med belæg fra Nonakas teori.

Vi starter med at forstå tavs og eksplicit viden som værende følgende definitioner.

- Tavs viden: underforstået viden, som kan være svært at formidle uden for individets egen forståelse. Den tavs viden ligger hos individet og opbygges gennem personlige oplevelser og udfordringer. Fordi tavs viden kommer af et personligt perspektiv, er den svær at dele med andre.
- Eksplicit viden: direkte viden som kan formidles og fortolkes mellem individer. Denne viden kan dokumenteres, vises og formidles uden yderligere komplikationer i kontrast til tavs viden.

### Den individuelle elev

At læring fremmes af dialog, betyder dog ikke, at den individuelle elev ikke får viden ud af deres egen tid med læringsteknologien. Uden opbyggelse af den tavs viden, ville det være en kort efterfølgende dialog mellem eleverne. Tværtimod mener Nonaka at opbyggelse af tavs videns kvalitet er påvirket af to faktorer.

Den første faktor er variationen af individets oplevelse og om oplevelsen er begrænset af rutine. Hvis variationen er lav og oplevelsen repetitiv falder kvaliteten af indlæring.

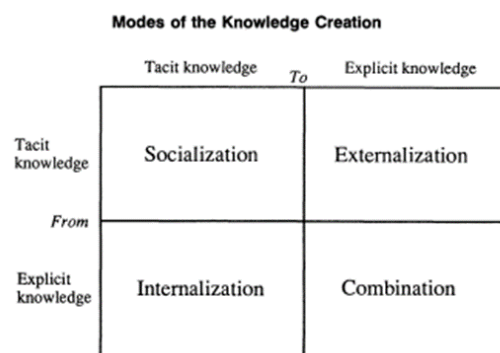
*"Routine tasks mitigate against creative thinking and the formation of new knowledge"* - (Nonaka, 1994, p. 21)

Dette problem har vi kunne forebygge ret nemt, ved at gøre applikationen til et bidrag til undervisningen frem for et permanent læringsmiddel. Det er ikke tænkt, at eleverne skal gentage processen ift. det projekt de udarbejder omkring rummet, medmindre de tilbagevender for gentagende information.

Den anden faktor er det Nonaka kalder *Knowledge of experience* (Nonaka, 1994, p. 22). *Knowledge of experience* indbefatter, at individets læring er præget, er en indre intention om selv aktivt at indgå i oplevelsen. Med andre ord er det vigtigt, at individets fysiske og psykiske samarbejde under indlæringen er til stede. Ift. vores projekt prøver vi at styrke dette princip via VR, så eleven føler sig både psykisk og fysisk til stede i rumskibet. Dertil har vi indført en god mængde dynamiske håndtag og knapper, som øger elevens følelse af immersion. Denne immersion hjælper til illusionen af rent faktisk at være til stede i rumskibet og derved også styrke samarbejdet mellem brugerens psykiske og fysiske tilstedeværelse.

Model for konvertering af viden

Fordi vi har anvendt Nonakas principper om viden og læring har det også været relevant at få defineret hvilken konvertering af viden vi har arbejdet med. Formålet med dette har været at forstå elevernes interaktion med læringsteknologien baseret på hvilken form for konvertering af viden de gennemgår.



Figur 2 - Modes of the Knowledge Creation

Ifølge Figur 2, kan viden gennem dialog konverteres på fire forskellige måder.

- Tavs til tavs
- Tavs til Eksplicit
- Eksplicit til eksplicit
- Eksplicit til Tavs

Den første konvertering er eksplicit viden til tavs viden. Når eleverne anvender læringsteknologien, går vi ud fra at de allerede er blevet præsenteret til emnet og pensum af deres underviser. Her vil eleven have opbygget viden i form af eksplicit viden fra underviseren, som nu skal anvendes i praksis. Ifølge Nonaka vil denne konvertering omdanne eksplicit viden

til tavs viden, som en naturlig del af praksis. Resultatet af dette, er en udvidelse af elevens tavse viden baseret på den eksplicite viden fra underviseren. Det er også her eleverne har haft muligheden for at stille spørgsmål til underviseren. Derudover er læringsteknologien designet således, at eleverne forventes præsenteret for grundlæggende begreber om rummet inden brug.

Næste konvertering er fra tavs til eksplicit, hvor eleven forventes at få italesat den tavse viden de har opbygget gennem læringsteknologien. Præmissen er her, at den tavse og eksplicite viden komplimenterer hinanden. Denne komplimenterne effekt sker i sammenhæng med internalisering og eksternalisering af viden. Her vil eleven først have mulighed for at internalisere ny viden og derefter viderebygge denne viden gennem en eksternaliserende proces.

Dette projekts anvendelse af denne model er derfor også baseret på følgende: at motivere processen i elevens konvertering af viden ud fra Nonakas model. Resultatet af dette er en positiv konvertering af viden fra tavs til eksplicit, hvis omstændighederne er optimale. Derfor har vi valgt at styrke dette princip med fokus på dialog.

At styrke tavs viden med dialog

Denne del er baseret på Nonakas artikel kapitel 2, *Basic Concepts and Models of Organizational Knowledge Creation*.

Her mener Nonaka, at der skal være et samarbejde mellem de to videns typer for at nye ideer og koncepter kan blive skabt og, at denne proces finder sted når individer indgår i en dialog.

*" It embraces a continual dialogue between explicit and tacit knowledge which drives the creation of new ideas and concepts. Although ideas are formed in the minds of individuals, interaction between individuals typically plays a critical role in developing these ideas." - (Nonaka, 1994, p. 15)*

Og så er spørgsmålet - hvordan kan vi anvende dette argument til vores læringsteknologi?

Nonakas begreb, *Communities of interaction*, er hovedprincippet her. Da vi indgik en dialog med vores kontaktperson og fysiklærer Johnny, gjorde han det klart, at det er vigtigt, at eleverne efter deres tid med applikationen, har muligheden for at indgå i dialog og snakke med de andre elever om hvad de har undersøgt. Dette kunne f.eks. gøres gennem en præsentation for klassen. Altså var det vigtigt, at børnene gennem egen interaktion med læringsteknologien først opbyggede et individuelt niveau af viden, som derefter kunne fremhæves i dialog.



*"That is to say, "communities of interaction" contribute to the amplification and development of new knowledge."* (Nonaka, 1994, p. 15).

Formålet med dette ville derfor, ifølge Nonaka, være et større udbytte af den samlede viden eleverne har opnået.

For at kompensere dette ønske om dialog mellem eleverne, har vi derfor designet en form for opgavetemplate som eleven skal løse når de anvender læringsteknologien. Når de så har udfyldt de angivne opgaver, har eleven derefter mulighed for at anvende deres resultater til at opbygge en præsentation for resten af klassen. På denne måde anvender alle eleverne deres egen individuelle erfaring og viden fra læringsteknologien. Her kan eleverne diskutere deres resultater og samtidig praktisere formidlingen af den tavse viden de har taget til sig.

### Fejllæring

Der er primært to forskellige måder at visualisere vores solsystem på. Den første måde er ved at skalere planeterne til de rigtige størrelsesforhold. Ved denne metode viser du dog ofte alle planeterne side om side, så man rigtig kan se forholdene mellem dem. Den anden måde at visualisere planterne, er ved at skalere efter afstandsforholdet mellem dem. Her vælger man dog at skalere planeterne op, så man bedre kan se dem. Med dette i tankerne, havde vi to forskellige problemer at tage stilling til.

Første problemstilling var, om man skulle kunne se naboplaneterne, når man var ved en planet. Eksempelvis om man skulle kunne se Venus og Mars som små planeter, når man besøgte Jorden. Vi ville mene at dette gav en bedre forståelse af planeterne og deres placering i solsystemet. Dog ville det ikke være realistisk, da dette ikke er tilfældet i virkeligheden.

Den anden problemstilling omhandlede planeternes relative størrelsesforhold. Da Solens radius er betydeligt større end selv den største planet, stod vi med to valgmuligheder. Enten skulle solen nedskaleres, for at kunne ses ordentligt, ellers skulle alle planeterne nedskaleres, for at solen fik en acceptabel størrelse.

Begge disse problemstillinger gjorde, at vi stod med et problem der kunne give anledning til fejllæring hos eleverne.

*"Fejllæring drejer sig om læring, der indholdsmæssigt ikke modsvarer det tilsigtede eller det, der er blevet formidlet"* - (Illeris, 2006, p. 166)

Denne læringsbarrierer vil opstå, da eleverne ville få et forkert syn på rummet, hvilket er præcis det vi ville rette op på med vores applikation. Derfor er det vigtigt for os at vise og fortælle brugeren, hvad de rigtige forhold er, på en anden måde. Dette vil blive taget op igen i *Om realistisk visualisering af rummet*.

## Teknologisk platform

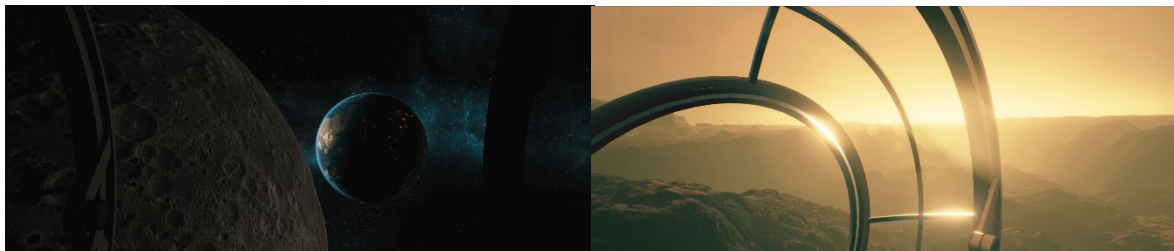
Udviklingen af vores læringsteknologi er baseret på hardware og software fra undervisningen i *Virtual og Augmented Reality I*. Til softwareudviklingen har vi gjort brug af programmerne *Unity Game Engine*, *Microsoft Visual Studios* og *Blender*. *Unity* er den oplagte platform til udviklingen af vores produkt, idet vores læringsteknologi er baseret på grafiske interfaces og udnyttelsen af forskellige spilelementer. I forbindelse med *Unity* har vi desuden skrevet forskellige C# scripts med programmet *Microsoft Visual Studios*, for at give de forskellige elementer i læringsteknologien den ønskede funktionalitet. De grafiske aspekter af vores læringsteknologi er primært udviklet i programmet *Blender*, hvor man nemt kan skabe og gengive forskellige 3D-objekter.

Det primære hardware i vores læringsteknologi er *Oculus Quest*. *Oculus Quest* er et Virtual Reality headset, som gør det muligt at skildre et computergenereret miljø, hvori brugeren kan interagere med verden omkring en. Således vil brugeren føle sig som en integreret del af teknologien, i stedet for at føle sig som en ekstern aktør.

## Related Work

For at kunne udvikle en velfungerende prototype har vi kigget på nogle VR-applikationer, som har kunnet hjælpe os på vej. Vi har haft en sammensat idé om hvad vi ville opnå og en ramme for hvordan vi ville udføre det. Her har vores referencer spillet en part i at raffinere produktet undervejs. Der er blevet taget eksempler som vi mente virkede godt og som vi dermed har fokuseret på, for at forøge kvaliteten af prototypen.

En af de referencer er en anden VR-applikation kaldet *SOLAR SYSTEM VR*, som også er udviklet til at være en læringsapplikation omkring rummet. Denne applikation tager brugeren ud på en rumrejse og er mest story teller baseret. Brugeren rejser igennem rummet hvor de kommer forbi diverse planeter i solsystemet og får viden tilført igennem en fortæller. Man får alt den praktiske viden at vide angående de forskellige planeter, hvordan de er opbygget, hvordan de fungerer osv.



Figur 3 - Solar System VR (Steam, 2022)

I applikation kommer man helt tæt på overfladerne på planterne, så man som rejsende kan se teksturen og tage ind, hvordan de ser ud i virkeligheden. Opbyggelsen af rejsen er som en immersive version af en dokumentarfilm og kan dermed virke meget mere optagende end at skulle læse i en bog eller se en film. Grafikken er blevet skabt til at være mere realistisk og gør det mere relaterbart, hvilket er en faktor som applikationen har fået gode anmeldelser på.

Vi tog dog ikke kun referencer fra applikationer hvor der er udviklet til læring af universet, men også fra VR spil generelt. En af disse er *Job Simulator* som er et meget velkendt VR spil for de fleste som har oplevet VR. Det er et spil hvor man, som navnet siger, bliver sat ind i nogle forskellige job simulationer. Spillet har fire forskellige jobmuligheder hvor man kan vælge imellem at være en mekaniker, en kassedame, en kok eller en kontorarbejder. Som spiller bliver man givet et sæt af opgaver som man skal udføre, hvor der er mange forskellige kombinationer af udførelser. Spillet er en god øvelse til at være i VR, hvor man kan tage tingene med ro og har mange muligheder for hvad man kan interagere med, se Figur 4.



Figur 4 - Job Simulator (Steam, 2022)

## Hvordan vi bygger videre på State of Art

De relaterede applikationer som vi fandt, har givet en inspiration i hvordan vi derefter eller løbende igennem vores proces har udviklet vores egen prototype. Inspirationerne kom i mange forskellige størrelser som fx i *Job Simulator*, hvor vi gerne ville gengive den glæde man får som bruger af de forskellige muligheder man aktivt kan gøre i spillet. Alle knapperne, håndtagene, lågerne osv. er noget som vi har taget fokus i at give til brugerne i vores egen applikation. Der er ikke kun taget inspiration i, at der skal være nok med muligheder, men også hvordan det føles at gøre diverse handlinger i spillet og hvilken haptisk feedback det giver.

For selve idéen bag vores applikation har det hjulpet at kigge på lignende teknologier, hvor det mest udbredte vi har fundet er baseret på story telling. Det vi har noteret, er derved, at der ikke er blevet implementeret en interaktiv oplevelse og at det dermed er mere en immersive filmoplevelse. I vores eget projekt har vi ville fokusere på det interaktive aspekt, som skulle medbringe en erfaringsbaseret viden i brugeren. I det at de aktivt er inde og kontrollere den viden de indtager, samt de erfaringer de gør sig, håbes der at blive opnået en længere oplagring af den faglige viden. Konceptet med en fortæller er dog noget vi ville have i baghovedet, ved muligheden for at implementere en tutor/mentor i vores applikation, som skulle hjælpe brugeren på vej.

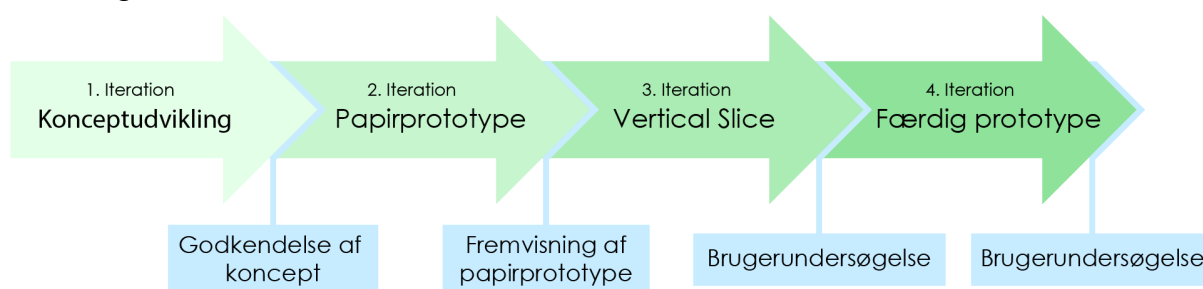
En ting som stod meget til diskussion i begyndelsen af designfasen, var hvorvidt vi ville give et realistisk syn på solsystemets størrelsesomfang, eller om vi skulle skalere det ned til en mere forstående kontekst. Det ses i *Solar System VR*, at de har arbejdet på at give en højkvalitets grafikoplevelse af solsystemet, som virker godt til en ikke-interaktiv applikation. Da vores demografi var baseret på unge i folkeskolen, valgte vi at gå efter den mere enkelte grafik. Denne grafik tog bl.a. udgangspunkt i *Job Simulators* low-poly stil. Vi har prøvet at samle det bedste fra begge verdener og derved forhåbentligt give brugeren en god oplevelse som er lærerig, interaktiv og omfattende.

## Metode

I dette afsnit vil der blive beskrevet hvordan vi har arbejdet med dette projekt. Herunder vil vores fremgangsmetode blive forklaret, som bl.a. indeholder vores ugentlige møder, samt en plan for vores iterationer. Derudover vil der blive forklaret hvordan vi vil teste vores projekt igennem forløbet ved hjælp af forskellige evalueringsmetoder.

## Fremgangsmetode

Før vi begyndte at udvikle applikationen, besluttede vi at lægge en plan for projektet. Dette gjorde vi bl.a. pga. tidligere erfaring med projektarbejde. At have en plan med deadlines og milestones, giver alle noget at arbejde hen i mod. Vi havde to faste deadlines, sat af vores vejledere. Dette var godkendelse af konceptet, og den endelige deadline. Derudover vidste vi også, at vi skulle brugerteste mindst to gange i projektet. Med dette i tankerne, lavede vi et processkema, se Figur 5, som vi arbejdede ud fra. Vores projekt blev essentielt delt op i fire forskellige iterationer, som hver især mandede ud i en form for konklusion.



Figur 5 - Processkema

Den første iteration var konceptudviklingen. Her var det vigtigt at få skabt et godt fundament at bygge applikationen på. Denne iteration lød på for-analyser og analysering af det eksisterende marked. Denne iteration endte med at skulle godkendes af vores vejledere, for at vi kunne arbejde videre til næste iteration.

Efter godkendelsen, begyndte vi at visualisere midtpunktet af vores projekt, ufoen. Her byggede vi en papirprototype, som skulle visualisere hvor forskellige funktioner skulle være i forhold til hinanden. Denne papirprototype skulle fremvises til vores medstuderende, som resulterede i god konstruktiv feedback.

Ved tredje iteration begyndte vi at bygge på vores virtuelle applikation. Dette ville være vores vertical slice og her var vigtigt at vi fik hovedelementerne med. Vi havde kontakt til Nymarkskolen og lavede en aftale med Johnny d. 25/3, og et par dage senere med hans klasse. Dette ville være vores første rigtige brugerundersøgelse, med elever i vores valgte målgruppe. Her ville målet været at få feedback for det generelle koncept, samt forslag til forbedringer.

Den sidste iteration omhandlede viderebygning af vores projekt på baggrund af egne mål, samt feedback fra den første brugerundersøgelse. Her gjaldt det om at få udarbejdet så mange funktioner som muligt, så vi kunne vise en færdig prototype til vores sidste brugertest. Denne brugertest var sat en måned efter den første, som gav os en smal ramme for udvikling. Med

denne brugertest kunne vi afprøve vores fulde applikation på de samme elever, for at kunne sammenligne resultater fra sidste brugerundersøgelse.

Igennem alle iterationer, valgte vi at have et ugentligt møde, hvor vi viser vores fremskridt. Med dette holder vi os konstant i gang. Derudover har vi løbende samtaler med hinanden. Til organisering af opgaver, har vi brugt GitKraken Boards. Dette værktøj er suverænt til at holde overblik over igangværende, kommende og afsluttede opgaver, og har hjulpet os gevaldigt.

## Undersøgelser

Ved tredje og fjerde iteration, har vi testet på vores målgruppe. Vi valgte, at begge disse tests skulle være lignende, for nemmere at kunne sammenligne resultaterne efterfølgende. Disse tests fandt sted i et uformelt miljø, som skulle give anledning til en mere afslappet samtale med eleverne. Vi valgte at teste på fire elever ad gangen. Vi mente dette var en god midte hvor eleverne følte sig trykke sammen, men stadig tog undersøgelsen seriøst.

Ifølge Jenny Preece, (Preece, 2002) er der tre forskellige typer af evaluering, herunder *Controlled settings involving users*, *Natural settings involving users* og *Any settings not involving users*. I dette projekt valgte vi at bruge en blanding af de første to typer. Måden vi gjorde dette på, var at kombinere dem. Begge typer inkluderede brugeren, hvor forskellen mellem dem lå i omgivelserne og om de var styret eller ej. Det var vigtigt for os, når først brugeren blev hjulpet i gang med applikationen, at de selv udforskede vores teknologi, uden guidning fra os. Vi motiverede eleverne til at gøre hvad de havde lyst til. På denne måde havde vi ingen kontrol over deres handlinger, dog havde de altid muligheden for at spørge os til råds. Dette mente vi, var en god blanding mellem at have et kontrolleret miljø, hvor vi derefter gav frie tøjler til at prøve applikationen.

Efter testene, gav vi alle eleverne en usability test, i form af SUS (Brooke, 1995). Denne test skulle vise hvor tilgængelig vores applikation var. Eleverne blev præsenteret for et stykke papir med 10 sætninger på. Her havde de muligheden for at sætte et kryds mellem et og fem, alt efter hvor enige det var med sætningen. Ved at have to brugertests planlagt, havde vi muligheden for at sammenligne de to tests. Dette ville vise, om vi havde forbedret os. Derudover havde vi også forberedt et interview med nogle formative spørgsmål, som skulle hjælpe os med at viderebygge på projektet.

## 1. Iteration - Konzeptudvikling

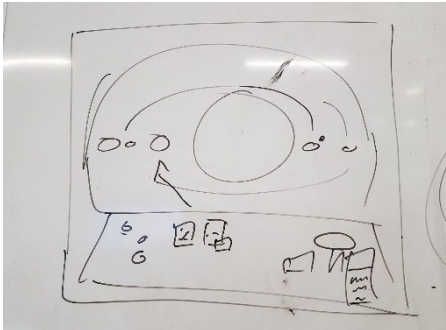
Det første punkt i udviklingen af vores læringsteknologi var konceptudviklingen. Med andre ord skulle vi, vha. forskellige idégenereringsmetoder, komme frem til et overordnet koncept for produktet. Udgangspunktet for idégenereringen var bl.a. baseret på teori fra fagene *Spildesign 1* og *Prototyper og Animationer til Computerspil 1*, fra 1. semester i *Spiludvikling og Læringsteknologi*.

Vores initiale fremgangsmåde gik ud på at brainstorme diverse potentielle idéer. Her skulle vi være opmærksomme på, at disse idéer skulle implementere enten virtual reality eller augmented reality. I starten var vi uenige om, hvilke af disse teknologier vi ønskede at arbejde med. En af vores tidligste idéer var en AR-læringsteknologi med fokus på matematik for folkeskoleelever. En anden idé, også med udgangspunkt i AR, gik ud på at lære folk om naturens økosystemer. Vi følte dog, at der var mere potentiale i at arbejde med VR. Efter at have afprøvet diverse VR-applikationer for inspiration (her iblandt *Job Simulator*), blev vi som gruppe enige om at udvikle vores læringsteknologi med fokus på VR. Efter en længere diskurs kom vi desuden frem til at udvikle en VR-oplevelse i rummet, med det formål at lære elever i folkeskolen om solsystemet. Vi begyndte dernæst at udarbejde forskellige skitser, for at konceptualisere konkrete aspekter for hinanden. Denne proces var desuden med til at skabe konsensus i gruppen, hvis der skulle være uenighed eller tvivl angående nogle af disse aspekter.

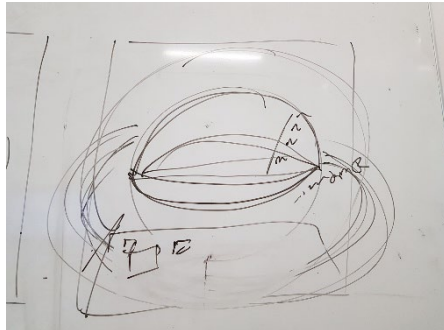
Præmissen for læringsteknologien skulle være som følger: brugeren skulle kunne flyver rundt til planeterne i solsystemet vha. et rumskib. Dette rumskib skulle designes som en ufo, da dette ville give brugeren mulighed for at se 360 grader omkring dem selv, se Figur 7 og Figur 11. Derudover skulle ufoens interiør designes med diverse knapper, håndtag og skærme, således at brugeren ville få muligheden for interagere med teknologien på mange forskellige måder, se Figur 6 og Figur 13. Vi ville desuden gerne kunne skære planterne op, for at se hvad de består af, se Figur 13

Et yderligere aspekt som skabte uenighed i gruppen var, hvordan vi ønskede at vise planeternes placering i rummet: skulle alle planeter være synlige for brugeren uafhængigt af ens egen placering, se Figur 12, eller skulle hver planet kun være synlig, når brugeren befandt sig i nærhed af den, se Figur 13. Denne uenighed var bl.a. baseret på det faktum, at man i virkeligheden ikke kan se alle planeter samtidigt, idet distancen mellem planeterne er for stor. Omvendt ville hele rumoplevelsen blive mere fascinerende for brugeren, hvis de kan se alle planeter på én gang. Dette vil samtidigt give et bedre indblik i størrelsesforholdet mellem

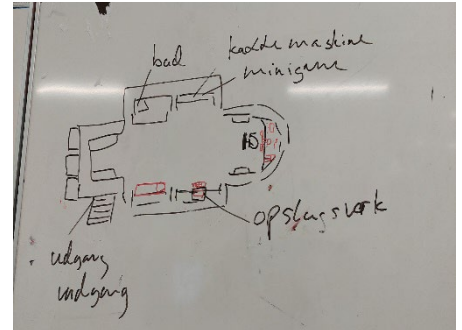
planeterne. Med andre ord skulle vi altså beslutte os om vi ville fokusere på gøre oplevelsen mere realistisk eller visuel. Vi valgte i sidste ende at gå den mere realistiske vej, bl.a. grundet potentiel fejllæring.



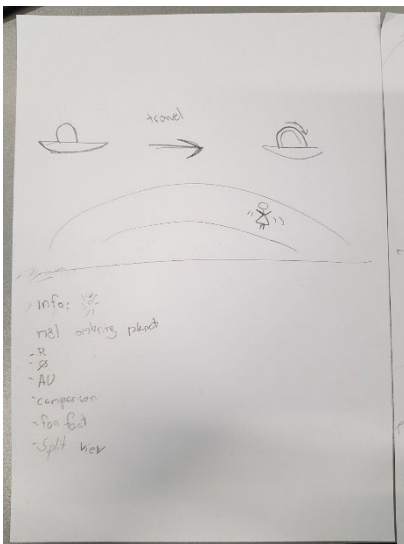
Figur 6 - Skitse 1



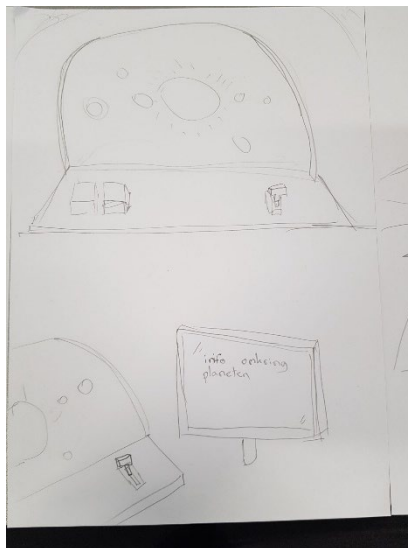
Figur 7 - Skitse 2



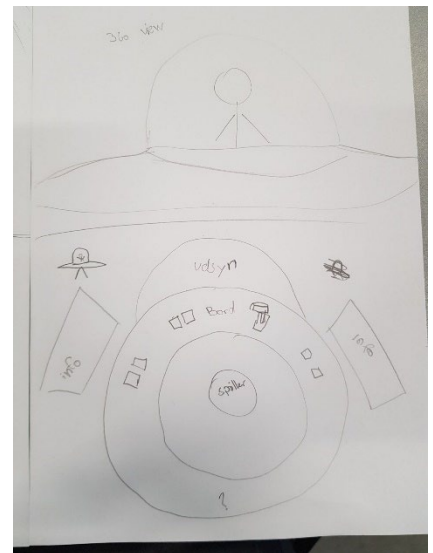
Figur 8 - Skitse 3



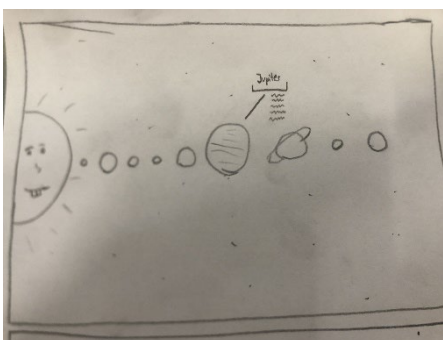
Figur 9 - Skitse 4



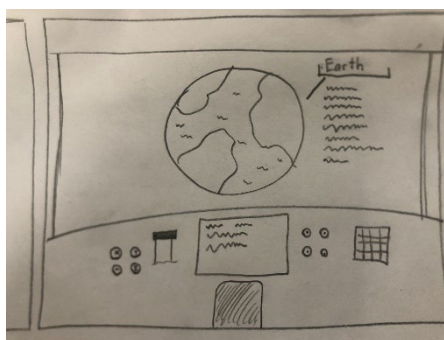
Figur 10 - Skitse 5



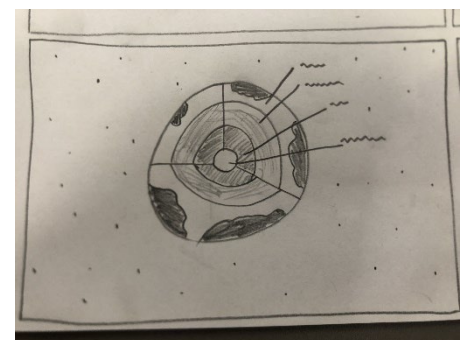
Figur 11 - Skitse 6



Figur 12 - Skitse 7



Figur 13 - Skitse 8



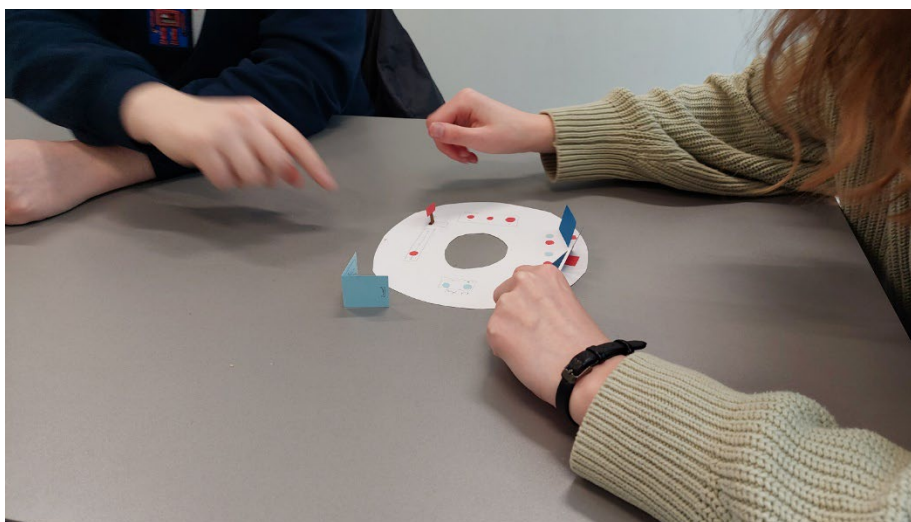
Figur 14 - Skitse 9



## 2. Iteration - Papirprototype

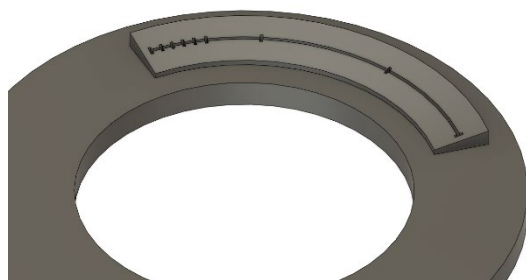
Anden iteration gik ud på at udvikle low-fidelity prototyper af vores produkt, for at give potentielle testpersoner indblik i projektets hensigt og design. Derudover var denne del af processen med til at skabe konsensus i gruppen angående eventuelle uenigheder blandt diverse designaspekter.

Vores low-fidelity prototype blev konstrueret af forskellige stykker farvet pap og papir, og skulle illustrere ufoen, hvori brugeren befinder sig, se Figur 15. Prototypen inkluderede diverse knapper, håndtag og skærme, med den hensigt at formidle de forskellige funktioner, f.eks. at man kan trække i håndtaget for at rejse til en planet, eller at man kan vise forskellige informationer på skærmene.



Figur 15 - Test af papirprototype

I forbindelse med denne low-fidelity papirprototype, begyndte vi desuden at udvikle en digital prototype i *Blender* med udgangspunkt i papirprototypen. Formålet med disse prototyper var at finde ud af om vores idé rent faktisk ville fungere i praksis, især hvad VR angår. Hertil begyndte vi at designe ufoens ydre udseende, samt at designe den indre panelstruktur for ufoen. Vi lagde især meget vægt i det indre design, idet panelet er den essentielle faktor for, at brugeren kan interagere med teknologien. Nogle forskellige iterationer af dette design kan ses på Figur 16.



Figur 16 - Ufoens paneldesign

Planeterne skulle også designes og implementeres. Dette var relativt enkelt, idet vi blot kreerede forskellige sfærer med hver deres individuelle materiale, for at give dem det ønskede udseende.

Til slut fokuserede vi på at inkludere de relevante VR-biblioteker i Unity, med det formål at opstille en sammenhæng mellem de forrige grafiske aspekter og selve VR-oplevelsen.

### 3. Iteration - Vertical Slice

I vores tredje iteration, begyndte vi at have et testbart produkt. I denne version havde vi netop lavet ufoen og dens panel, samt placeret solen og jorden over for hinanden. Dette betød at man nu kunne være i rumskibet og kigge rundt på både jorden og solen, men også se på ufoens interface. Ufoen havde stjerneknapper og planethalveringsknapper, hvor kun de første virkede. Det var heller ikke muligt at bevæge hverken gearstangen eller håndtaget til at flyve til andre planeter. Informationsskærmen havde kun data omkring jorden, og havde ikke noget at skifte i mellem. Ufoens og brugerens position var også låst, så man ikke kunne bevæge sig. Til sidst havde vi også udarbejdet hænder i form af rektangulære kasser, som brugeren kunne trykke på knapperne med. Alt dette rundede ud i, at eleverne ikke havde meget at gøre i applikationen. Denne opsætning skulle dog give essensen af, hvordan vores læringsteknologi ville føles.

Vores mål med denne iteration, var at vise både eleverne og læreren, hvad vores læringsteknologi gik ud på. Vi håbede at de ikke havde brug for alle funktionerne, til at kunne forestille sig hvordan man skulle flyve til forskellige planeter, og hvordan man skar planeterne op.

En anden ting vi ville teste var, om nutidens folkeskoleelever var velkendt med virtual reality. Da virtual reality er et relativt nyt produkt i det private marked, var vi ikke sikre på om alle eleverne havde kendskab til det. Dette kunne resultere i, at vi automatisk ville få lavere points på vores SUS tests.

Den første test vi lavede, var med Johnny. Johnny var vores primære kontakt på Nymarkskolen. Det var også hans klasse vi senere fik lov at lave vores anden test af applikationen på.

Vi udnyttede en blanding af kontrollerede og naturlige opstillinger på brugerne på sådan en måde, at der blev taget små grupper ud af gangen i løbet af en undervisningstime, for at teste vores produkt. Grupperne bestod omtrent af fire personer ad gangen som testede vores produkt samtidigt med, at vi observerede og kunne hjælpe dem hvis der gik noget galt eller hvis de havde spørgsmål. Dette kombinerede fieldstudies i form af observationen af hvordan de naturligt reagerede på vores produkt. Efter at de blev testet, blev eleverne sat til at svare på en SUS evaluering, som blev efterfulgt af et opfølgende semistruktureret formel interviewform, for at kunne få så meget brugbart feedback som muligt.

Der blev testet primært på elever, med undtagelse af et par lærer, der ønskede at prøve teknologien. I det at der blev testet på lærerne, fik vi muligheden for at høre deres meninger om idéen i helhed, samt hvis teknologien skulle blive aktuelt tilgængeliggjort til undervisning.



*Figur 17 - Elever i gang med at teste vores Vertical Slice*

Vi har fået en masse blandede testresultater, som hver især har kunne hjælpe os i den rette retning og give en bedre forståelse for, hvordan vores arbejde skulle udfolde sig. Resultaterne kan ses i Bilag 1, hvor svarende fra interviewet er blevet sammensat.

SUS testens resultater blev skemalagt, hvor man kan se det endelige resultat på Figur 18. Der er her blevet taget et udkast af resultatet hvor resten kan ses på Bilag 2.

Number	VR XP	Score	SUS Score
1	x	35	87,5
...	...	...	...
22	x	31	77,5
<b>VR XP result</b>	<b>18/22</b>	<b>Average score</b>	<b>81,99</b>

Figur 18 - Resultater af SUS

Der skal tages i betragtning at denne SUS evaluering er noteret til at være lidt for kompliceret for den demografi den er blevet brugt på, og dermed kan de endelige resultater der vises, godt varierer i deres aktualitet. Dette er noget som uddybes i diskussionen.

Resultatet af den gennemsnitlige score ses som x ud af 100 hvilket sætter vores resultater i den positive ende. Der har dog været gentagende steder i nogle svarresultater, hvor vi har kunnet gennemskue en forvirring, relateret til spørgsmålene, da der var givet meget modsigende svar. Overordnet ses der en relation med den feedback der ses fra interviewet med de SUS resultater vi har noteret.

Efter vores første brugertests, havde vi fået langt mere feedback end vi havde drømt om. Både Johnny og hans elever gav utrolig værdifuld feedback, som hjalp os videre med vores udvikling.

Den første samtale med Johnny, gav os svar på, om vores applikation overhovedet ville passe ind i folkeskolens pensum. Dette mente Johnny sagtens at den ville. Han havde et par forslag, såsom at tilføje en quiz, som eleverne ville kunne viderebringe til klassen og fremlægge deres rummission. Derudover gav det os håb, da en voksen mand, der aldrig havde prøvet virtual reality før, nemt kunne finde ud af at bruge vores applikation, uden rigtige instrukser.

Resultaterne fra eleverne var også overvældende positive. Stort set alle elever havde det sjovt, imens de prøvede applikationen, og da de blev spurgt, om de kunne se dem selv bruge applikationen i undervisningen, var svaret entydigt.

Af al vores feedback, kom også nogle gode idéer, som vi valgte at arbejde videre på. Én af disse idéer, var at tilføje text-to-speech på vores informationsskærm. Denne idé kom fra en af klassens lærerinder. Hun sagde at mange i klassen var ordblinde, og havde derfor svært ved at læse informationerne. En TTS-funktion ville både gøre det nemmere for brugere med

ordblindhed, men også give muligheden for at lave andre ting, imens informationerne blev læst højt. Dette var en ting vi ville have lavet til næste brugertest. En anden idé de kom med, var at kunne flyve rundt mellem planeterne. Dette ville give muligheden for at se planeterne tættere på, samtidig med at give en helt ny dimension til applikationen.

Til næste iteration, ville vi fokusere på at få alle ting på dashboardet til at virke. Dette vil gøre, at man kan rejse til de forskellige planeter, i stedet for at være fanget ved solen og jorden. Derudover ville man også kunne skære alle planeterne op og se hvordan de er opbygget. Overordnet set, ville det give brugeren meget mere at trykke på og trække i.

En anden ting vi ville arbejde på, var at udarbejde en slags tutorial, som introducerede brugeren til rumskibet. Ved denne test, hjalp vi en del med at vise dem hvad de kunne. Denne tutorial ville gøre det automatisk ved en slags rundvisning af rumskibet og dets knapper.

#### 4. Iteration - Færdig Prototype

I 4. iteration var vores testbare produkt klar. Der var nu ikke længere håndtag og knapper uden funktioner og vi havde samtidig tilføjet en masse elementer til ufoen baseret på brugerfeedback. Derudover havde vi bl.a. fået implementeret et quizsystem, funktionelt dashboard, funktionel informationssskærm samt en introduktion til læringsmidlet, som vi var spændt på at få testet.

Derfor var det grundlæggende mål for 4. iteration at teste vores færdige prototype og de ændringer og tilføjelser vi havde implementeret siden sidste besøg på skolen. Vores mål var også at samle information til hvad vi ville kunne videreudvikle.

Vores introduktion i applikationen fik en positiv respons. Vi havde via text-to-speech designet en pseudo AI ved navn Mads. Mads forklarer brugeren hvad de forskellige elementer i rumskibet funktion er. Samtidig anvendes Mads også til oplæsning af informationer til de forskellige planeter. Denne tilføjelse var baseret på feedback fra 3. iteration, hvor der havde været efterspørgsel på hjælp til de ordblinde elever. Den eneste negative ting omkring vores implementering af vores guide Mads, var den mekaniske måde han oplæser teksten på. Med mere fokus på lyd-design i applikationen, ville det helt klart være optimalt at samarbejde med en oplæser.

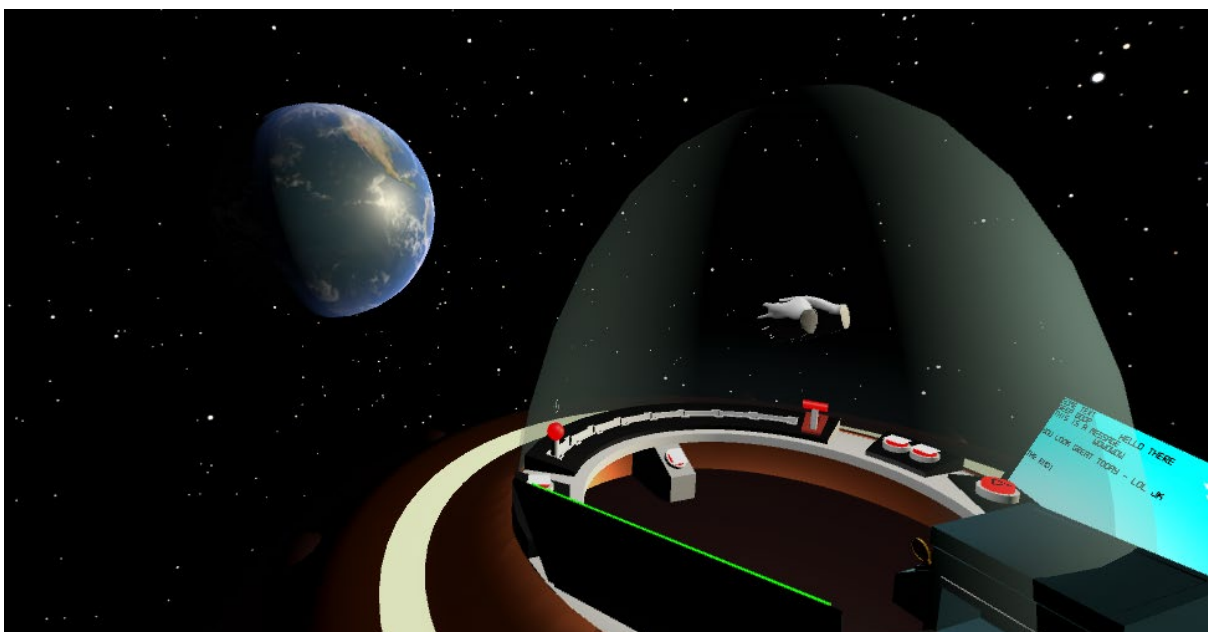
Vores 4. iteration af projektet inkluderede også et funktionelt dashboard. Nu var det muligt for eleverne at rejse mellem de forskellige planeter. Af hensyn til cybersickness var designet af denne mekanik dog simpel og planeterne bliver bare udskiftet med den nye valgte planet ved

aktivering. Som refleksion til denne mekanik, kunne vi godt tænke os at have udarbejdet en mere visuel effekt ved planetskifte.

Vi havde også implementeret et quizsystem. Dette quizsystem var relativt simpelt og havde til funktion at udfordre eleverne til at besvare spørgsmål undervejs i deres rumrejse. Som refleksion til denne del, ville det være en god idé at udarbejde quizen til først og fremmest at være mere dybdegående og dernæst tilpasse den til det mere konkrete fysikpensum.

Den sidste større tilføjelse til vores 4. iteration, er skærmene. Her kan eleverne bl.a. finde fakta om de forskellige planeter. Under tests observerede vi, at eleverne anvendte infoskærmen til at finde svar til quizen, hvilket var positivt. I sammenhæng med refleksionerne ift. vores quizdesign, ville det være optimalt at tilføje meget mere information til infoskærmen. I 4. iteration havde vi kun implementeret en minimal mængde relevant fakta om de forskellige planeter. I fremtidige iterationer, kunne der tilføjes meget mere fagmateriale.

En ekstra ting, som vi havde tilføjet baseret på feedback var, at man kunne flyve rundt i rumskibet. Flere elever havde ønsket, at de gerne vil have kontrol over rumskibet og derved også kunne flyve rundt om planeterne. Det viste sig dog ret hurtigt, at det fremprovokerede cybersickness, fordi man sidder ned uden fremdrift ved brug af applikationen. Derfor valgte vi under 4. iteration at fjerne muligheden for at flyve rundt af hensyn til vores testpersoner. Derudover havde vi også oplevet problemer med, at hinge-komponenterne i *Unity* ikke følger med rumskibet, når det ændrer position. Derfor ville man ikke kunne trække i nogen håndtag efter flyvning.



Figur 19 - Billede fra *We Are In Space*



Figur 20 - Billede af ufoens interior

Som afslutning på 4. iteration var resultatet følgende. Vores endelige prototype var færdig og klar til at blive testet. Vi havde fået implementeret nye funktioner ud fra 3. iterations feedback. Disse nye funktioner var relative simple, men oplagte til videreudvikling.

Anden testplan var bygget videre fra første iteration, hvor spørgsmålene var blevet tilpasset til den nye prototype. Meningen med denne test var at se hvorvidt eleverne og lærerne mente, at det gik i den rigtige retning og for at sammenligne resultaterne fra sidste test med den nye. Her ville vi kunne se om der foregik en forøgelse af positive reaktioner og derved om der var potentiale i vores produkt. I Bilag 3 kan ses en sammensat tabel af svar fra andet interviewet. Der blev også lavet en SUS evaluering, hvorefter vi kunne se om vores samlede score er gået op. En opsummering på resultaterne kan ses i Figur 21. Alle resultaterne kan ses i Bilag 4.

Som forrige gang var der tegn på at resultaterne ikke kunne være helt aktuelle, men når der sammenlignes med første gangs SUS evaluering, ses en stigning af scoren.

Number	VR XP	2nd visit	Score	SUS Score
1		x	37	92,5
...	...	...	...	...
19		x	38	95
<b>VR XP result</b>	<b>/19</b>	<b>/19</b>	<b>Average score</b>	<b>85,72</b>

Figur 21 - Resultater fra SUS

## Teknologisk Udvikling

Den teknologiske udvikling af vores projekt har især taget udgangspunkt i undervisningsteori fra *3D programmering 1, 2 og 3*, samt *Virtual og Augmentet Reality 1*. Dette afsnit vil gennemgå bestemte kode highlight, og give et overblik over udvalgte detaljer, som vi føler er essentielle til forståelsen af teknologiens udvikling og opbygning.

### Arbejde med VR

Når man arbejder med VR i *Unity* er der en lille tjekliste man skal igennem. *Unity* gør det muligt at downloade/importere nogle bestemte packages til sit projekt, som opsætter ens workspace med de værktøjer der skal til for, at VR kan virke.

De packages som skal installeres er henholdsvis *XR Plugin Management*, som er det der gør det muligt at arbejde med mixed reality, og *XR interaction toolkit* som gør, at vi kan få standard inputs uden selv at skulle programmere dem. Derudover er der to samples som vi har importeret hvilket er *Default Input Actions* og *XR Device Simulator* som er praktisk i form af, at den gør det muligt at simulere et headset så der kan testes uden.

Når alt det så er gjort har vi inde i *XR Plugin Management* sat det til at vi udvikler til *Oculus*, så *Unity* ved hvilket system vi arbejder med.

Alt dette gør, at vores workspace giver os muligheden for at opsætte et VR-spil og indsætte interaktive objekter. Blandt andet hvis man har et objekt som man gerne vil kunne tage i hånden via ens kontroller. *XR Interaction Toolkit* indeholder en komponent man tilføjer til objektet kaldet *XR Grab Interactable*, som gør, at man kan tage objektet op i hånden. En anden komponent vi har brugt, er *HingeJoint* som sætter et interval i grader for hvor meget det er muligt at trække i et objekt. Dette ses f.eks. på vores to håndtag i rumskibet.

Angående test af applikationen har man to muligheder. Man kan enten builde programmet til VR-brillerne og så køre spillet, eller man kan koble sig på sin desktop og direkte køre applikation igennem *Unity*.

At builde applikationen og derefter køre den på VR-brillerne giver en bedre stabilitet i programmet og er også mindre rodet i form af, at der ikke er lige så mange ting man skal tage højde for. Der er her kun fokus på en ting og det er den applikation man kører. Denne metode er dog langtrukken og kan være alt for tidskrævende hvis det er små ændringer i programmet som skal testes ad gangen.



Man kan tilkoble sine VR-briller til sin computer og dermed gå i desktop mode. Dette gør, at du med dine VR-briller på, kan kontrollere din computer inde i VR og dermed køre programmet direkte fra *Unity*. Denne metode gør det betydeligt hurtigere og nemmere at teste små ændringer, men er dog mere ustabil og giver derved ikke altid den helt optimale oplevelse.

Disse to metoder har vi brugt igennem vores proces, hvor vi har buildet til at teste på skolen og en kombination af begge til at teste derhjemme.

## Håndtering af data

Hver planet i vores læringsteknologi er defineret ud fra bestemt data. For at holde styr på denne data, har vi gjort brug af *Unity's ScriptableObjects*. Et *ScriptableObject* fungerer som en databeholder uafhængig af klasseinstanser. Med andre ord refererer klassen til dataen frem for at kopiere værdierne, som på længere sigt reducerer hukommelsesbrug. Denne måde at lagre

```
[CreateAssetMenu(fileName = "New Planet", menuName = "Planet")]
public class PlanetData : ScriptableObject
{
    //Empiri
    [Tooltip("The name of the planet")]
    public string planetName;
    [Tooltip("The type of planet (if it is a gas planet or a rock planet)")]
    public string PlanetType;
    [Tooltip("The time it takes for the planet to rotate around itself (in days)")]
    public float rotationSpeed;
    [Tooltip("The time it takes for the planet to orbit the sun (in days)")]
    public float orbitTime;
    [Tooltip("The radius of the planet (in km)")]
    public float radius;
    [Tooltip("The distance to the sun (in 100.000 km)")]
    public float distanceToSun;
    [Tooltip("The mass of the planet (in 10^24 kg)")]
    public float mass;
    [Tooltip("the amount of moons that orbit the planet")]
    public int amountOfMoons;
    [Tooltip("A sprite of the planet")]
    public Sprite planetImage;
    [Tooltip("soundclips")]
    public AudioClip[] SpeechClips;

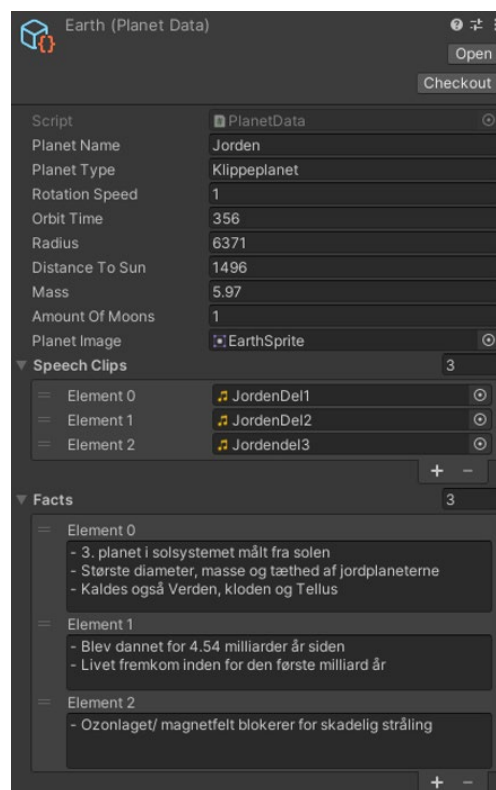
    //Fun facts
    [TextArea(3, 10)]
    public string[] facts;

    //the string that contains the data from the planet
    public string GetData() {
        return "Type: \t\t\t\t" + PlanetType +
            "\nRotationstid:\t\t" + rotationSpeed + " dage" +
            "\nKredsløbstid:\t\t" + orbitTime + " dage" +
            "\nRadius:\t\t\t\t" + radius + " km" +
            "\nKM fra solen:\t\t" + distanceToSun + "00000 km" +
            "\nMasse:\t\t\t\t" + mass + "*10^24 kg" +
            "\nAntal måner:\t\t" + amountOfMoons;
    }
}
```

Figur 22 - PlanetData

data på er især brugbar ved håndtering af data, som ikke skal ændres (som f.eks. empiri for specifikke planeter). En yderligere fordel ved *ScriptableObjects* er deres flexibilitet, idet man let kan tilføje og ændre værdierne på dataen under udvikling af projektet. Desuden kan *ScriptableObjects* let genereres. Hvis man altså ønsker at tilføje en ny planet til projektet, kan man let oprette et tilsvarende *ScriptableObject* uden at skulle lave et helt nyt *MonoBehaviour* script.

Vi har således oprettet et *ScriptableObject* ved navn *PlanetData* til håndtering af værdierne. Dataen er inddelt i forskellige variabelfelter, som hver især indeholder en specifik værdi for den pågældende planet. Et billede af selve scriptet kan ses på Figur 22. Vores *ScriptableObjects* består primært af simple variabeltyper, såsom *string*, *float* og *int*, for at referere til specifik empiri for planeten, som f.eks. planetens navn, masse, rotationstid, antal måner osv. Hertil er der givet et felt, som indeholder et sprite af planeten. Derudover indeholder vores *ScriptableObjets* to arrays, som respektivt indeholder nogle “fun facts” om planeten, samt nogle tilhørende lydklip. Til slut i scriptet definerer vi metoden *GetData()*, som returnerer en lang string af den samlede data for planeten. Denne metode kaldes i forbindelse med vores datamonitor, som vi går i dybden med i næste afsnit af rapporten. På Figur 23 kan man se et eksempel på hvordan vores *ScriptableObject* ser ud i inspectoren.



Figur 23 - *ScriptableObject* i inspectoren

## Datamonitor

For at formidle viden omkring planterne til brugeren, har vi designet og programmeret en in-game monitor. Monitoren består af en skærm og tre knapper, se Figur 24. Skærmen fungerer som et display, som viser forskellige data for den pågældende planet. Knapperne i højre og venstre side bruges til henholdsvis at iterere frem og tilbage igennem planetens data og “fun facts”. Den midterste knap bruges i forbindelse med oplæsning af disse “fun facts” hvis dette ønskes.



Figur 24 - Informationsskærm i applikationen

I *MonoBehaviour* scriptet “*Monitor*” har vi defineret funktionaliteten for datamonitoren. De tre vigtigste metoder i denne klasse er *GetDataText()*, *SetDataText()* og *TypeText()*. Ved gennemgang af disse funktioner bruger vi desuden tre grundlæggende variabler:

- *planetData*: en reference til dataen for den respektive planet.
- *monitorText*: teksten som skal vises på skærmen.
- *dataIndex*: indekset for at afgøre, hvilke informationer som skal vises på skærmen.

På Figur 25 kan man se et udklip af disse funktioner og variabler fra “*Monitor*” scriptet.

Metoden *GetDataText()* returnerer en *string* med informationer afhængigt af værdien for *dataIndex*. Hvis *dataIndex* er mindre eller lig med 0, kalder *planetData* på metoden *GetData()*, som returnerer en lang *string* med empiri fra *PlanetData* (se tidligere afsnit om datahåndtering). Hvis *dataIndex* derimod er større end 0, returnerer *planetData* i stedet en *string*-værdi fra “fun fact”-arrayet. Bemærk, at *dataIndex* inkrementeres og dekrementeres vha. knapperne som nævnt tidligere i afsnittet.

Metoden *SetDataText()* henter *planetData* for den nuværende planet og nulstiller teksten fra *monitorText*-variablen. Hvis *planetData* for planeten eksisterer, sættes *string*-variablen *fullText* til den returnerede værdi fra *GetDataText()*.

Til slut har vi metoden *TypeText()*, hvis formål er at vise teksten på skærmen. For at give en mere æstetisk effekt, har vi programmeret denne funktion til at printe ét bogstav ad gangen. For at gøre dette har vi defineret metoden som en *Coroutine*, som gør det muligt at sprede en funktions opgaver over flere frames. Efter et selvdefineret delay på få millisekunder (*yield return new WaitForSeconds(delay)*), kontrollerer vi antallet af bogstaver, som allerede er blevet printet på skærmen. Hvis antallet af bogstaver er mindre end længden af *fullText*-variablen, vil et nyt bogstav blive tilføjet efter dette delay. Således bliver teksten vist et bogstav ad gangen.

```
public PlanetData planetData;    //instance of the planet data ScriptableObject
public Text monitorText;        //text to be displayed on the monitor
public int dataIndex;          //index reference to the current data

//returns the data for the current planet
public string GetDataText()
{
    if (dataIndex <= 0)
    {
        return planetData.GetData();
    }
    else
    {
        return planetData.facts[dataIndex - 1];
    }
}

//(re)set the fullText string variable
public void SetDataText() {
    planetData = GameManager.instance.currentPlanetData;
    stringIndex = 0;
    monitorText.text = "";
    if (planetData != null) {
        fullText = "Planet: " + planetData.planetName + "\n\n\n" + GetDataText();
    }
}

//type the text one character at a time
IEnumerator TypeText() {
    while (true) {
        if (stringIndex < fullText.Length) {
            stringIndex++;
        }
        currentText = fullText.Substring(0, stringIndex);
        monitorText.text = currentText;
        planetImage.sprite = planetData.planetImage;
        yield return new WaitForSeconds(delay);
    }
}
```

Figur 25 - Monitor script

## Missionsmonitor

Vores anden monitor er missionsmonitoren. Denne monitor består også af en skærm og tre knapper, se Figur 26, og har nogle af de samme funktionaliteter som datamonitoren (se tidligere afsnit, Datamonitor). Formålet med denne monitor er at give brugerne mulighed for at teste deres viden om solsystemet. Dette gør vi vha. såkaldte missioner, som tager udgangspunkt i de informationer og empiri, som vi formidler i løbet af læringsteknologien.



Figur 26 - Missionsmonitoren

Missionerne fungerer som en multiple choice quiz. Ligesom vores *PlanetData*, har vi struktureret missionerne som *ScriptableObjects*. Dette skyldes bl.a. at vi meget let kan tilføje og referere til nye missioner, samtidigt med at gøre selve strukturen mere fleksibel (se tidligere afsnit Håndtering af Data). Hver mission består af et spørgsmål og nogle svarmuligheder. Et udsnit af scriptet "*MissionMonitor*" er angivet foruden, se Figur 27.

I "*MissionMonitor*"-scriptet gemmer vi alle missioner i listen *allMissions*. Funktionen *NewMission()* beregner et tilfældigt indeks i listen vha. *Random.Range*-metoden, og sætter dernæst variabelen *currentMission* til den tilsvarende mission i listen. Denne mission vil dernæst blive fjernet fra listen, for at man ikke gentager den samme mission flere gange. Spørgsmålene og svarmulighederne vil dernæst blive printet på skærmen med *SetText()* funktionen.

Hvis ikke der er flere missioner tilbage i *allMissions* listen (altså at alle missioner er blevet besvaret), bliver der i stedet printet ud, hvor mange af spørgsmålene der blev besvaret korrekt.

For at besvare missionens spørgsmål skal brugeren enten trykke på knap A, B eller C. Når én af disse knapper trykkes ned, vil funktionen *AnswerQuestion()* blive kaldt. Denne funktion tager et argument *'bool answeredCorrectly'*, som angiver om man svarer korrekt eller forkert på spørgsmålet. Hvis svaret er korrekt, bliver missionen sat til at være gennemført, og omvendt. Samtidigt vil en tilsvarende "Korrekt" eller "Forkert" tekst blive vist på skærmen, hvorefter vi invoker en ny mission med funktionen *NewMission()*.

```

public List<Mission> allMissions = new List<Mission>();
public List<Mission> finishedMissions = new List<Mission>();
public Mission currentMission;

//set new mission
private void NewMission() {
    int index = Random.Range(0, allMissions.Count);
    if(allMissions.Count > 0) {
        currentMission = allMissions[index];
        allMissions.RemoveAt(index);
        SetText();
        SetHeadline();
    }
    else {
        currentMission = null;
        SetHeadline("Ikke flere missioner");
        int amountOfRightAnswers = 0;
        for(int i = 0; i < finishedMissions.Count; i++) {
            if (finishedMissions[i].answeredCorrectly) {
                amountOfRightAnswers++;
            }
        }
        SetText("Du har svaret rigtigt på " + amountOfRightAnswers + " ud af " +
finishedMissions.Count + " spørgsmål");
    }
}

//answer mission
private void AnswerQuestion(bool answeredCorrectly) {
    if(currentMission != null) {
        if (answeredCorrectly) {
            audioSource.clip = correctAnswer;
            audioSource.Play();
            currentMission.answeredCorrectly = true;
            finishedMissions.Add(currentMission);
            currentMission = null;
            SetText("Korrekt!");
            Invoke("NewMission", 3f);
        }
        else {
            audioSource.clip = wrongAnswer;
            audioSource.Play();
            currentMission.answeredCorrectly = false;
            finishedMissions.Add(currentMission);
            currentMission = null;
            SetText("Forkert");
            Invoke("NewMission", 3f);
        }
    }
}
}

```

Figur 27 - Missionmonitor script

## Knapper og håndtag

En stor del af vores applikation omhandler vores knapper og håndtag. Stort set alle vores funktioner skal enten aktiveres af et tryk på en knap, eller ved at trække i håndtag. Vi synes, det at trykke på fysiske knapper i det virtuelle space, i modsætning til et tastatur, er med til at fremme immersion. Derfor valgte vi, at alle funktioner skulle ske ved hjælp af håndgribelige genstande, såsom en gearstang til at vælge planet, et håndtag til at flyve og knapper til at skifte mellem forskellige tilstande.

## Knap

I dette afsnit vil der blive beskrevet den nødvendige kode bag disse håndgribelige genstande. Den første genstand vi lavede, var knappen. Igennem hele applikation bruges samme knapsystem, dog med ændret effekt ved tryk. Måden dette er lavet på, er ved brug af *Unity's* eventsystem. Dette gør, at man kan kalde et event på et specifikt tidspunkt, som andre scripts kan abonnerer til. Hver gang dette event bliver kaldt, vil alle dens abonnenter også blive kaldt. Under ses koden for knappen, samt dets event.

```
void Update()
{
    //Checks if pressed or released
    if (!_isPressed && GetValue() + threshold >= 1) Pressed();
    if (_isPressed && GetValue() - threshold <= 0)
    StartCoroutine(Released());
}

private float GetValue()
{
    //Gets the relative value
    var value = Vector3.Distance(_startPos, transform.localPosition) /
    _joint.linearLimit.limit;

    if(Math.Abs(value)< deadZone)
    {
        value = 0;
    }

    return Mathf.Clamp(value, -1f, 1f);
}
```

Figur 28 - Kodeuddrag fra knap

Knappen holder konstant øje med dens position relativ til dens startposition. Ved hjælp af *GetValue()* funktionen, tjekker koden om knappen er oversteget den definerede grænseværdien. Hvis dette er tilfældet, kører den funktionen *Pressed()*. Når knappen derefter bliver sluppet, kalder den på *Released()* funktionen, på baggrund af samme værdier.

```

void Pressed()
{
    _isPressed = true;

    //Calls all subscribers
    onPressed.Invoke();

    //Plays the button click sound
    ButtonClickSound.Play();

    //Vibrating the joystick in use
    StartCoroutine(VibrationManager.singleton.Vibrate(.1f, .1f, .1f,
CurrentController));

    //Disabling the box collider to prevent spamming
    BC.enabled = false;
}

private void OnCollisionStay(Collision collision)
{
    if (collision.collider.CompareTag("LeftHand"))
    {
        //Left Controller
        CurrentController = OVRInput.Controller.LTouch;
    }
    else if (collision.collider.CompareTag("RightHand"))
    {
        //Right Controller
        CurrentController = OVRInput.Controller.RTouch;
    }
}
}

```

Figur 29 - Kodeuddrag fra knap

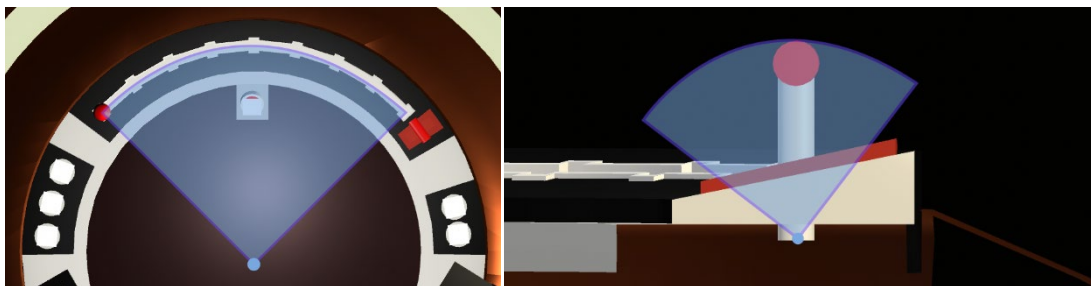
Inde i *Pressed()* funktionen, se Figur 29, sker det, at *onPressed* eventet bliver invoked, som kalder på alle abonnenter. Derudover sker afspilningen af knaplyden. Vi valgte at hardcode denne del, da alle knapperne skulle have samme lyd når man trykkede på dem. Denne afspilning kunne i princippet også have været en abonnent til eventet, hvis ikke alle knapperne havde samme tryklyd. Efter lyden, bliver der vibreret i det pågældende joystick, som du trykker med. Dette er gjort ved hjælp af colliders. Begge virtuelle hænder har enten tag'et *LeftHand* eller *RightHand*. Hvis en knap kolliderer med et objekt med en af disse tags, bliver variabelen *CurrentController*, skiftet til den pågældende hånd. Derefter ved programmet hvilket hånd den skal vibrere. Derudover er vibrationsfunktionen lavet, så vi selv kan styre hvor længe og kraftigt vibrationen skal være. Vi valgte at minimere begge parametre, da det ikke må tage fokus fra oplevelsen, men blot indikerer at du nu har trykket på knappen. Til sidst i funktionen deaktiverer vi knappens collider. Dette er for at man ikke kan holde knappen inde og derved kalde funktionerne hver frame. Collideren bliver først genaktiveret, ét sekund efter *Released()* funktionen bliver kaldt.



Derudover er det tilføjet et `ConfigurableJoint`-komponent til knappen. Denne komponent bliver brugt til to ting. Det ene er at give en grænse til hvor langt ned knappen kan blive trykket. Den anden er brugen af springs. Dette gør at knappen automatisk bliver skubbet op, når man ikke trykker på den, som om der var fjedre under knappen.

### Håndtag

I projektet er to forskellige håndtag. Det ene håndtag er den store gearstang i midten af instrumentpanelet. Dette håndtag bruges til at sætte kursen til de forskellige planeter. Efter man har valgt sin destination, trækker man dernæst i håndtaget til højre, for at flyve derhen. Begge håndtag er opbygget på samme princip, og gør brug af `HingeJoint`-komponenten. Denne komponent gør, at håndtaget er påmonteret et fiktivt hængsel, og kan derfor kun bevæge sig i en cirkulær bevægelse omkring nævnt hængsel. Derudover er der påmonteret en grænse for, hvor langt hver håndtag kan bevæge sig. På billedet under ses en visuelt representation af disse `HingeJoints` og deres effekt. På Figur 30, ses gearstangen ovenfra. Grænserne er sat ved den første og sidste destination, og på samme figur ses også håndtaget fra siden. Ligeledes her er grænserne sat, så håndtaget ikke kan gå ud over kanten.



Figur 30 - Visualisering af `HingeJoints`

Den røde top på begge håndtag, er hvor brugeren kan tage fat med sin controller. Dette er gjort ved hjælp af `XRGrabInteractable`-komponenten. Denne komponent er fra XR biblioteket, og gør det muligt at tage fat i objekter med controlleren. Derudover har denne komponent forskellige events, ligesom knappen. Disse events er eksempelvis når du tager fat i håndtaget. Ved begge håndtag, har vi gjort brug af `SelectExited` eventet, som bliver kaldt når du slipper håndtaget.

Det der sker, når du slipper gearstangshåndtaget, er at `onRelease()` funktionen i `Gearshift` scriptet bliver kaldt. Denne funktion samt den hjælpende funktion `CalculateClosestGear()` kan ses under.

```

public void OnRelease()
{
    Vector3 closestGear = new Vector3();
    closestGear = snapList[CalculateClosestGear()];
    transform.position = closestGear;
    AS.Play();
}
public int CalculateClosestGear()
{
    snapList.Clear();

    foreach (Transform child in snappingPoints.transform)
    {
        snapList.Add(child.position);
    }

    currentPos = transform.position;

    float shortestDist = Mathf.Infinity;
    Vector3 closestGear = new Vector3();

    for (int i = 0; i < snappingPoints.transform.childCount; i++)
    {
        float distToTar = Vector3.Distance(currentPos, snapList[i]);
        if (distToTar < shortestDist)
        {
            shortestDist = distToTar;
            closestGear = snapList[i];
            chosenPlanetIndex = i;
        }
    }
    return chosenPlanetIndex;
}

```

Figur 31 - Kodeuddrag fra håndtag

I *OnRelease()* funktionen starter den med at instantiere en ny *Vector3* variable, hvor den senere vil gemme det nærmeste gear. Et gear i dette sammenhæng, er hvor gearstangen kan snappe til. Gearstangen har ni gear, hvor det første er Solen, og det sidste er Neptun. Disse gear er skrevet ind i arrayet *snapList*. Næste step i *OnRelease()* funktionen kalder på funktionen *CalculateClosestGear()*. Denne funktion sørger for at finde det tætteste gear i forhold til gearstangens position da den blev sluppet. Dette gør den ved hjælp at et simpelt forloop, som looper igennem alle gear og finder distancen fra gearet til gearstangen. Hvis et gear er tættere end de forrige, bliver dette valgt som det tætteste. Når dette for-loop er ovre, indeholder variabelen *chosenPlanetIndex* det indeks i *snapList* som er tættest på gearstangen. Dernæst bruges dette indeks til at ændre gearstangens position til det tætteste gear. Efter dette, afspiller gearstangen en lyd, som indikerer at gearstangen er snappet til et gear.

Udover at snappe til det tætteste gear, vibrerer gearstangen også hver gang du bevæger den over et snapping punkt. Dette gøres ved hjælp af *vibrationscolliders*. Disse colliders er placeret

på gearenes lokationer, og når gearstangen kommer i kontakt med en collider der har tag'et *Vibrate*, vil den aktive kontroller vibrerer på samme måde som hvis du trykkede på en knap.

```
public void UpdateGearText()
{
    //Changes text to closest planet
    if(OVRInput.Get(OVRInput.Button.PrimaryHandTrigger) ||
OVRInput.Get(OVRInput.Button.SecondaryHandTrigger))
    {
        gearText.text =
GameManager.instance.planets[CalculateClosestGear()].name.ToUpper();
    }
}

private void UpdateGearTextRot()
{
    //Canvas facing the camera
    canvas.transform.LookAt(mc);
}
```

Figur 32 - Kodeuddrag fra håndtag

Over ses funktionerne *UpdateGearText()* og *UpdateGearTextRot()*. Begge funktioner bliver kaldt i *Update()* funktionen. Disse funktioner sørger for teksten over gearstangen. Denne tekst viser hvilken planet din gearstang er sat til. Teksten er simpelt genereret i *UpdateGearText()* ved hjælp af *CalculateClosestGear()* funktionen. Denne gang bruges den returnerede *int*-variabel i en liste der indeholder alle planeterne og den søgte planets navn skrives i tekst objektet. Den anden funktion, *UpdateGearTextRot()* gør, at canvas'et der indeholde denne tekst, altid vil vende mod kameraet. Dette gør at teksten er læsbar fra alle vinkler.

Håndtaget der trækkes i for at komme til den valgte planet, fungerer stort set på samme måde som gearstangen. Også her bruges der et array med to snapping points. Det der sker når håndtaget snapper til den aktive position, er at den kigger på hvilken planet er valgt af gearstangen, for derefter at deaktivere alle planeter i scenen, og aktivere den valgte. Dette giver effekten af, at man teleporterer til den valgte planet.

## Diskussion

### Om samarbejde med folkeskole

Igennem vores fire iterationer, har vi være i tæt kontakt med både vores vejledere, Johnny og hans klasse på Nymarkskolen. Dette har været fundamentet for den iterative proces vi har

arbejdet med igennem dette projekt. Denne aktive dialog har hjulpet os med at forblive på rette spor, og hjulpet os med mange nye idéer til vores projekt.

Én ting er, at kunne læse sig til fysik/kemi pensummet i folkeskolen, en anden ting er at snakke med en lærer der underviser i det. Samarbejdet med Johnny har været en hjørnesteen i vores projekt og fra dag ét, har han bidraget til hvordan man kan skabe en applikation, til elever i udskolingen. Den første spilbare iteration var Johnny med til at udvikle, og her kom han med idéen omkring en quiz, som skulle være unik til hver elev. Denne quiz skulle senere hen blive til en fremlæggelse foran klassen, for at vise hvad man har lært. Denne idé kom til livs i iterationen efter, og blev et hit. Eleverne brugte nemlig vores applikation efter intention, og begyndte at flyve til forskellige planeter for at finde svaret til de viste spørgsmål.

Ved den sidste brugertest, var alle funktioner ikke fuldstændig færdige. Printereren kunne ikke printe, og quizen var ikke helt færdiglavet. Alle elever fik de samme seks spørgsmål og selve quizskærmen var ikke koblet til det stykke papir hvor opgaverne også stod på. Dette kunne give anledning til forvirring, da eleverne ikke kunne finde ud af hvad de skulle med disse objekter.

Begge brugertests har været kæmpe succeser. Ikke kun fordi det lod til, at elever og lærere var glade for vores idé, men også for alt den feedback vi fik. På grund af vores uformelle og afslappet brugertest, havde vi fornemmelsen af, at eleverne sagde hvad de tænkte og godt turde at kritisere os.

### Om risiko for bias

Ved evaluering af vores produkt på Nymarkskolen, blev eleverne bedt om at udfylde et SUS-skema og derefter svare på spørgsmål omkring applikationen. Som forklaret tidligere, var dette i et uformelt og afslappet miljø. Vi kunne mærke på eleverne, at de var spændte på at komme ind og prøve VR-spil i skolen. Det, at VR er relativt nyt i skolen, gør at det muligvis kan blive taget ekstra godt i mod og der kan derfor være anledning til bias.

Vi har en anelse om, at eleverne måske har været lidt for positive omkring vores applikation, fordi det er nyt og spændende. Vi ville have ønsket at kunne teste vores produkt over en længere periode, og dermed indsamle data omkring deres forståelse af rummet og planeterne. På denne måde ville VR blive en mere integreret del af undervisningen, og måske ikke så spændende som før. Her ville vi også få at vide om applikationen har motiveret eleverne mere, end hvis al undervisningen foregik på papir.

Derudover stødte vi på et problem ved SUS-evalueringen. Her lød et af vores spørgsmål på følgende: ”Jeg synes der var for meget inkonsistens i applikationen”. Her var nogle elever i tvivl om hvad ordet ”inkonsistens” betød. Vi tror mange ikke forstod dette ord, men ikke turde spørge hvad det betød. Selvom vi gjorde vores bedste til at forklare hvad ordet betød, kan dette være en fejlkilde i undersøgelsen, da nogle elever måske bare har sat et tilfældigt kryds.

### Om VR i undervisningen

Virtual reality i undervisningen bliver mere og mere attraktivt, da udvalget af undervisningsmateriale i VR konstant vokser sig større. Ifølge Johnny, var der ingen tvivl om hvorvidt han ville bruge vores applikation i undervisningen, når den en dag blev færdig. Han mener at man med virtual reality kan vise mange eksperimenter og ellers abstrakte ting, som ville kunne fremme forståelsen hos elever. Han mener, at man en dag ville kunne have læringsmateriale til alle fag og at virtual reality ville kunne være en indbygget del af undervisningen. Dog er der ét problem når det kommer til dette. Johnny mente nemlig også, at udvalget af undervisningsmateriale skulle være stort, for skoler rundt om i landet, ville betale for VR headset til undervisning. Disse headset er dyre og skal holdes ved lige. Alternativet til en *Oculus Quest*, er *Google Cardboard*. Google så problemet i de dyre headset, og kreerede en brille lavet af pap, hvor man, med sin telefon, kunne se simpel virtual reality, se Figur 33. Dog lægger denne form for VR-biller op til et andet problem. Denne type headset understøtter ikke kontrollere. At kunne trykke på knapper og trække i håndtag, er en stor del af vores applikation. Derfor ville det ikke være muligt at prøve vores applikation med *Google Cardboard*.



Figur 33 - Google Cardboard

### Om realistisk visualisering af rummet

Igennem projektet har vi haft en løbende diskussion omkring, hvordan vi ville visualisere rummet. Som forklaret tidligere, bliver vores solsystem ofte illustreret på én af to forskellige måder. Vi ville gerne give eleverne en god start på astronomi. Derfor ville vi også undgå

fejllæring (Illeris, 2006) af solsystemet samt hvordan det skal opfattes. Vi valgte at skalere solen en anelse ned, da den ellers ville blive for stor at vise. Dette kan dog give anledning til fejllæring, og er noget vi er opmærksomme på. Vi valgte ikke at vise naboplaneterne omkring brugerens besøgte planet, da dette ikke er hvad ses i virkeligheden. Kun med et skarpt øje eller en kikkert, kan nogle planeter ses fra Jorden.

Hvis vores målgruppe havde været på et højere niveau, havde vi klart skulle gøre vores applikation mere præcis. Dette eksempelvis gøres ved at skalere solen til dens rigtige størrelse, og dermed skalerer de andre planeter ned i størrelse. Derudover skulle planeternes måner samt ringe have været implementeret.

En ting vi var i tvivl om under udvikling af applikationen, var hvorvidt stjerne kortene ville være præcise, når man besøgte andre planeter. Dog fandt vi hurtigt ud af, at stjernerne ikke skifter betydelig position fra alle solsystemets planeter (University of Southern Maine, 2022). Dette er fordi, af den nærmeste stjerne er mange flere lysår væk, end til den nærmeste planet.

### Om arbejde med VR

At arbejde med virtual reality har været besværligt på flere punkter. Da teknologien kræver, at man har et VR-headset, har det ikke været alle i gruppen der har haft mulighed for at teste derhjemme. Måden vi har arbejdet omkring dette på, har været ved at én person har haft headsettet til at teste hvad de andre har lavet. Derudover har vi haft ugentlige møder, hvor vi alle har kunne teste de ændringer vi havde lavet.

En anden besværlighed ved arbejde med VR, var at teste applikationen. Processen i at bygge et spil til headsettet, for hver lille ændring, tog lang tid. Dette gav udviklingen et handicap, da alt tog meget længere tid at teste. Det var først senere i projektet at vi fandt ud af *Oculus Link*, hvilket gav muligheden for at teste sin applikation igennem computeren med headsettet. Dog skulle man være opmærksom på, at når man gjorde dette, så kørte programmet via computeren. Derfor kunne man nemt overse, hvis headsettet ikke kunne klare det.

## Opsamling og Konklusion

Vores arbejde med projektet har resulteret i et unikt og lærerigt produkt. Vi står nu med en brugbar læringsteknologi, som på nogle punkter har overgået de forventninger vi havde i starten af projektet. Det skal dog stadig pointeres, at denne læringsteknologi er en prototype, og skal derfor ses som et 'Proof of Concept', frem for et fuldstændigt færdigt produkt. Der er naturligvis plads til diverse forbedringer.

Projektet bygger på viden i VR, læringsteori og programmering, som vi løbende har tilegnet os gennem dette semester, samt viden og erfaringer fra tidligere semestre og semesterprojekter. Med henblik på vores problemformulering kan vi konkludere følgende:

Vores indledende problemstilling gik ud på at undersøge, hvordan man kan supplere folkeskolens fysikundervisning med en læringsteknologi. Som udgangspunkt valgte vi specifikt at fokusere på astronomi og rummet, med henblik på at gøre selve oplevelsen så interaktiv og engagerende så mulig. Vi har således udviklet en VR-applikation, som kan erstatte og supplere den faglige læsning for folkeskoleelever, når de skal lære om solsystemet og rummet. Hertil har vi anvendt relevante læringsteorier, her bl.a. James Paul Gee's teori om udvikling af lærerige og engagerende computerspil, Ikujiro Nonaka's teori om tavs og eksplicit viden, og Knud Illeris' teori om læringsbarrierer. Under udviklingsforløbet har vi gjort brug af den iterative udviklingsproces, med fokus på at forbedre og bygge ovenpå tidligere iterationer af produktet. I forbindelse med dette, har vi desuden været i kontakt med vores ønskede målgruppe, for at udvikle produktet efter deres behov via brugertest. Målgruppen er udskolingen i folkeskolen, hvor vi, vha. kvalitative interviews og kvantitative spørgeskemaer, har akkumuleret værdifuld feedback til videreudvikling af læringsteknologien.

Til opsummering bidrager vores læringsteknologi med en alternativ tilgang til undervisningen i folkeskolen. Vi håber, at både elever og lærer kan få gavn af teknologien, og at produktet kan lyse op for det læringsteoretiske potentiale, som virtual reality og teknologi kan have i undervisningen. I fremtidig udvikling af produktet vil vi derfor stræbe efter at optimere eksisterende aspekter af læringsteknologien, samt at udvikle videre på nye måder, hvorpå man potentielt set kan forbedre læring for folkeskoleelever.

## Perspektivering

Ved resultatet af vores prototype har vi en gensidig mening om, at det er gået som vi havde håbet. Dog er der altid ting som der kan forbedres, hvor end relevante de ændringer ville have

været til at skabe en velfungerende prototype. Da vi startede, havde vi mere eller mindre en ret solid plan for hvad vi ville lave og hvordan det skulle se ud. Når vi ser tilbage på implementeringen og designet af rumskibet, kan vi godt se, at det ville have været bedre hvis vi havde udregnet et mere gennemtænkt layout. Dette layout indebærer eksempelvis hvor meget plads vi havde brug for og hvor de forskellige komponenter skulle placeres. Dette kunne have resulteret i et mere intuitivt og struktureret design for brugerne at skulle sættes ind i.

Der er nogle interaktioner planlagt til spillet, som vi ikke fik tid til at implementere og blev valgt fra da vi prioriterede de mere fundamentale aspekter af applikationen. Vi havde overvejet en måde hvorpå man let kunne sammenligne størrelsesforholdene på, blandt andet planeternes omkreds men muligt også deres afstand til hinanden.

Ved de forskellige lag på planeterne, ville vi have implementeret et system som gjorde, at brugeren kunne pege på dem og få relateret informationer.

Derudover havde vi eftertænkt muligheden for at kunne tage ned på planeterne og lave forsøg. Blandt andet i form af at kaste en bold op i "luften" og se forskellen på tyngdekraften eller tage temperaturen på planeten. Dette ville have skabt muligheder for brugeren selv at skulle finde ud af informationer frem for at søge efter planeten og derefter få informationerne givet. Ved at de selv aktivt udarbejder de missioner de får og søger efter svar, bliver der givet lidt hen ad en erfaringsbaseret viden som gør, at de nemmere kan huske de informationer de finder frem til.

Der er mange forskellige ting som vi i gruppen har snakket om, men vores endelige resultat af prototypen giver en god fornemmelse af hvad vores hensigter med applikationen var.



## Referencer

- Brooke, J. (1995, 11). *SUS: A quick and dirty usability scale*. Retrieved from ResearchGate:  
[https://www.researchgate.net/publication/228593520\\_SUS\\_A\\_quick\\_and\\_dirty\\_usability\\_scale](https://www.researchgate.net/publication/228593520_SUS_A_quick_and_dirty_usability_scale)
- Dreyfus, H. L., & Dreyfus, S. E. (2005, 06). *Expertise in Real World Contexts*. Retrieved from ResearchGate:  
[https://www.researchgate.net/publication/275011036\\_Expertise\\_in\\_Real\\_World\\_Contexts](https://www.researchgate.net/publication/275011036_Expertise_in_Real_World_Contexts)
- Gee, J. P. (2005, 03). *Learning by Design: Good Video Games as Learning Machines*. Retrieved from ResearchGate:  
[https://www.researchgate.net/publication/253367747\\_Learning\\_by\\_Design\\_Good\\_Video\\_Games\\_as\\_Learning\\_Machines](https://www.researchgate.net/publication/253367747_Learning_by_Design_Good_Video_Games_as_Learning_Machines)
- Illeris, K. (2006). *Læring*.
- Nonaka, I. (1994, 02). *A Dynamic Theory of Organizational Knowledge Creation*. Retrieved from JSTOR: <https://www.jstor.org/stable/2635068>
- Preece, J. (2002). *Interaction Design - Beyond Human-Computer Interaction*. Wiley.
- Steam. (2022, 05 29). *Job Simulator*. Retrieved from Steam:  
[https://store.steampowered.com/app/448280/Job\\_Simulator/](https://store.steampowered.com/app/448280/Job_Simulator/)
- Steam. (2022, 05 29). *Solar System VR*. Retrieved from Steam:  
[https://store.steampowered.com/app/1379970/Solar\\_System\\_VR/](https://store.steampowered.com/app/1379970/Solar_System_VR/)
- Undervisningsmisteriet. (2018, 05). *Læseplan for faget fysik/kemi*. Retrieved from Undervisningsmisteriet: <https://www.uvm.dk/-/media/filer/uvm/publikationer/folkeskolen/faghaefter/faelles-maal-2014/200813-fysik-kemi--laeseplan.pdf>
- Universe Today. (2022, 05 30). Retrieved from Universe Today:  
<https://www.universetoday.com/>
- University of Southern Maine. (2022, 03 02). *"Do the constellations look the same from other planets?"*. Retrieved from University of Southern Maine:  
<https://usm.maine.edu/planet/do-constellations-look-same-other-planets>

## Bilagliste

Bilag 1 - Resultater fra første interview

Bilag 2 - Sus resultater fra første brugerundersøgelse

Bilag 3 - Resultater fra andet interview

Bilag 4 - Sus resultater fra anden brugerundersøgelse

Bilag 5 - Link til video

## Emnefordeling

<b>Anthon</b>	<ul style="list-style-type: none"><li>• Abstract</li><li>• Abstract (Dansk)</li><li>• Indledning</li><li>• Baggrund</li><li>• Problemformulering</li><li>• Organisering af projekt</li><li>• A Dynamic Theory of Organizational Knowledge Creation</li><li>• 4. Iteration - Færdig Prototype</li></ul>
<b>Caroline</b>	<ul style="list-style-type: none"><li>• Problemformulering</li><li>• Related Work</li><li>• 3. Iteration - Evaluering og Testplan</li><li>• 4. Iteration - Evaluering og Testplan</li><li>• Arbejde med VR</li><li>• Perspektivering</li></ul>
<b>Emil</b>	<ul style="list-style-type: none"><li>• Problemformulering</li><li>• Introduktion til teoriafsnit</li><li>• Good Video Games as Learning Machines</li><li>• Five Stage Model of Skill Acquisition</li><li>• Teknologisk Platform</li><li>• 1. Iteration - Konceptudvikling</li><li>• 2. Iteration - Papirprototype</li><li>• Teknologisk Udvikling</li></ul>

	<ul style="list-style-type: none"> <li>○ Indledning</li> <li>○ Håndtering af data</li> <li>○ Datamonitor</li> <li>○ Missionmonitor</li> <li>● Opsamling og Konklusion</li> </ul>
<b>Rasmus</b>	<ul style="list-style-type: none"> <li>● Problemformulering</li> <li>● Fejllæring</li> <li>● Metode</li> <li>● 3. Iteration - Vertical Slice</li> <li>● Knapper og Håndtag</li> <li>● Diskussion</li> </ul>