

Navn: Anders, Astrid, Mikkel B, Mikkel E, Rasmus
Vejledere: Kasper Skov og Magnus Håkon
Skole: Viden Djurs HTX

Fag: Digital Design og Udvikling
Forløb: Arkadehallen
Afleveringsdato: 27-11-2022

Projekt Sandkasse



Skole: Viden Djurs HTX

Fag: Digital Design og Udvikling

Vejledere: Kasper Skov og Magnus Håkon

Forløb: Arkadehallen

Afleveringsdato: 27 – 11 – 2022

Af: Anders Larsen, Astrid Petersen, Mikkel Blom, Mikkel Erikstrup, Rasmus Riis

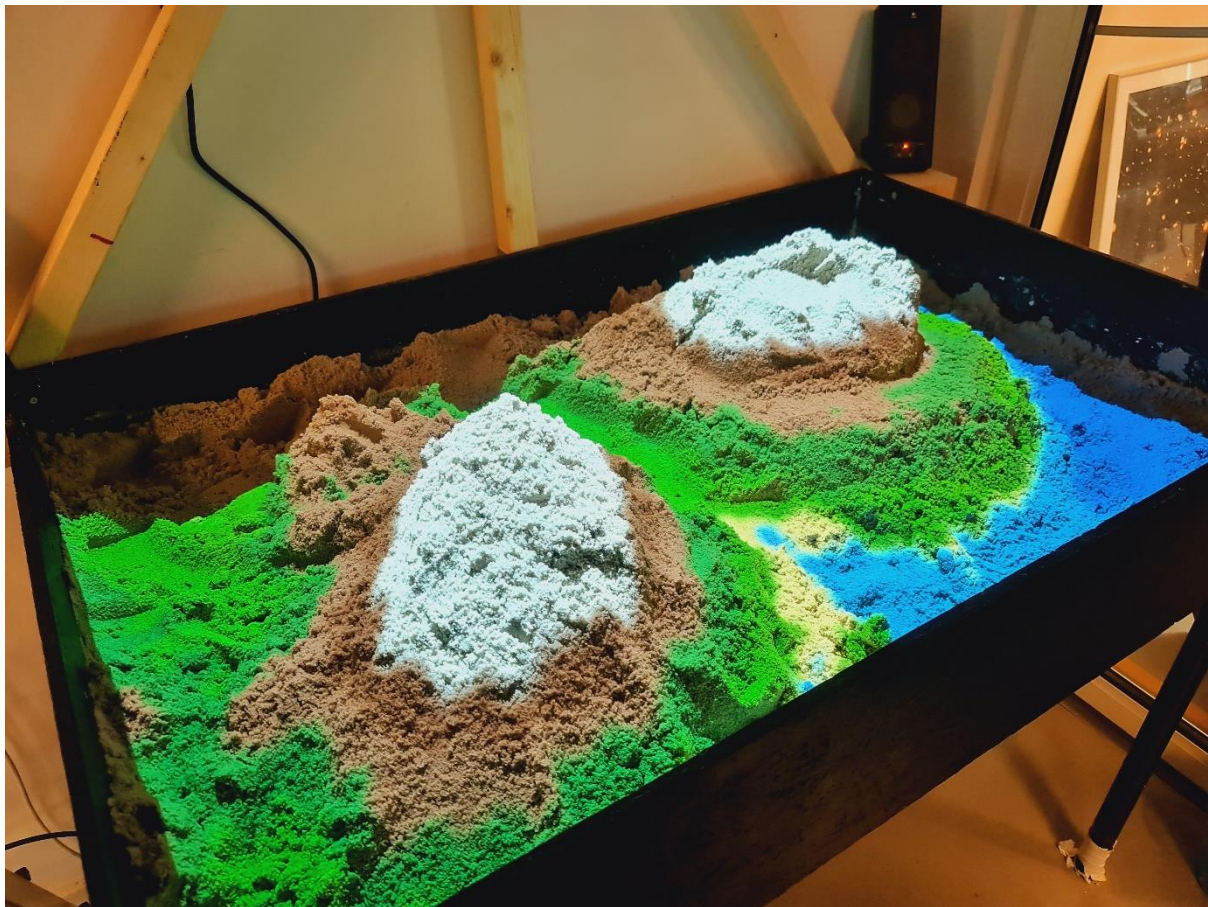
Indholdsfortegnelse

Indledning	1
Teori	2
Spil- og spillercentrisk	2
Spillets mekanikker	2
Interaktionsdesign	3
Produktudvikling	3
Idégenerering.....	3
Brainstorm	4
Evaluering af brainstorm og idegenerering	4
Koncept/Design.....	5
Moodboards.....	6
Prototyper	8
Prototype 1	8
Prototype 2	9
Prototype 3	10
Projektstyring.....	11
Produktion	12
Bord og reelt design	12
Musik.....	18
Designmål.....	18
Tekniske overvejelser.....	18
Komposition og produktion af soundtracket.....	18
Instrumentering	19
Musikteori	20
Kode	20
Kvalitetssikring	26
Brugernes vurdering af visuelle indtryk	26
Brugernes vurdering af sandhøjde.....	26
Brugertest	27
Mulige forbedringer	28
Konklusion.....	29
Bibliografi	30
Bilag.....	31

Figur 1: Billede af det endelige produkt i aktion.....	1
Figur 2: Screenshot af u-vægtet kravmatrix	5
Figur 3: Screenshot af vægtet kravmatrix.....	5
Figur 4: Moodboard af "Hvid minimalisme"	6
Figur 5: Moodboard af "Sort minimalisme"	7
Figur 6: Moodboard af "Rustik natur"	7
Figur 7: Billede af test på snorsystem til op hejsning af projektor	8
Figur 8: Skitse af projektorens projektionsvinkel.....	8
Figur 9: Billede af test og opmåling af projektoren på bordpladen.....	9
Figur 10: Billede af test af kinect sammen med projektor.....	10
Figur 11: Render af første 3D mockup af bordet	12
Figur 12: Billede af sandvinkeltest på 55 grader.....	13
Figur 13: Billede af sandvinkeltest på 10 grader	13
Figur 14: Render af andet mockup af bordet, samt ben design	14
Figur 15: Render af tredje mockup af bordet, samt projektorholder.....	15
Figur 16: Skitse af bordets metalben set oppefra.....	16
Figur 17: Billede af afvigelsen i højde fra bordene	17
Figur 18: Skitse af monteringsmulighederne for lægterne mellem bordpladen og bordbenene	17
Figur 19: Screenshot af de forskellige soundtracks	19
Figur 20: Skitse af hvordan kommunikationen mellem programmer skal forløbe.....	21
Figur 21: Skitse af hvordan information mellem kinecten og Unity bevæger sig.....	21
Figur 22: Screenshot af IsFileLocked funktionen	22
Figur 23: Screenshot af arduino kode	23
Figur 24: Screenshot af programmets læsning af arduino potentiometre værdier	23
Figur 25: Screenshot af ParseData funktionen	24
Figur 26: Screenshot af konvertering af data værdier	24
Figur 27: Screenshot af terraintype klassen	24
Figur 28: Screenshot af terrænregionerne	24
Figur 29: Screenshot af generate_map funktionen	25
Figur 30: Screenshot af terrængenerationen	25
Figur 31: Screenshot af udregning af terrænregion procentdel udgørelse af mappet	26
Figur 32: Billede af dybdetest måling	27

Indledning

Vi har gennem denne opgave skulle skabe en arkademaskine med et fysisk interaktionsinterface med formålet at vække interesse hos forbigående HHX/HTX-elever til et åbent hus-arrangement. Vi har i opgaven valgt at lave en arkademaskine, som vækker en følelse af magi, vidunder, og følelsen af at skabe noget. For at opnå det, har vi produceret en interaktiv sandkasse, med en dynamisk lydoplevelse, som former sig til brugerens interaktioner med sandet i sandkassen. Projektet har haft mange problematikker, som vi, gennem en iterativ designproces, har gennemarbejdet og løst. Mange elementer skulle samarbejde på tværs af forskellige programmer for at få projektet kunne konstrueres, hvorefter det skulle opsættes, testes, kalibreres, og fremvises. Alt dette har ført frem til vores endelige produkt (Se figur 1).



Figur 1: Billede af det endelige produkt i aktion

Teori

Spil- og spillercentrisk

Game design kan blandt andet ses fra to forskellige synsvinkler, spilcentrisk og spillercentrisk. En spilcentrisk synsvinkel vil fokusere på selve spillet, så som hvad deti bund og grund består af, hvilke elementer der er i spillet, og hvilken struktur det har. Det er bare nogle af de elementer som kan arbejdes med fra en spilcentrisk synsvinkel. Den spillercentriske synsvinkel derimod har mere fokus på spilleren, og hvilken oplevelse spillet giver. Oplevelsen er vigtig både før, under og efter spillet. Uenigheder i hvordan spil bør defineres stammer ofte i en forskel på, om der bliver brugt en spil- eller spillercentrisk synsvinkel. Spilcentriske definitioner vil oftest være fokuseret på at definere det formelt, hvorimod en spillercentrisk definition vil være ligeglad med hvad spillet indeholder, så længe spilleren opfatter det som et spil (Larsen, To perspektiver på computerspil, 2022). Under udviklingen af projekt sandkasse har gruppen haft et spillercentrisk synspunkt på udviklingen. Målet med spillet har været at give spilleren en oplevelse af vidunder, opdagelse og magi.

Spillets mekanikker

Der er forskellige definitioner på, hvad et spil er, men langt de fleste er enige om, at spilleren skal kunne interagere med spilverden. De muligheder spilleren har for at interagere med spilverden kaldes spilmekanikke, og de kan ses fra både et spil- og en spillercentrisk synsvinkel. Fra den spilcentriske synsvinkel handler det om, hvordan spilleren rent formelt interagerer med spillet. En spillercentrisk synsvinkel derimod, kigger på spillerens oplevelse ved at interagere med spillet. Der er flere måder at kategorisere spilmekanikker på, en af dem er ved at dele dem op i primære, sekundære og tertiære spilmekanikker. Primære spilmekanikker bruges konstant, og de er nødvendige for, at spilleren kan klare spillets udfordringer og nå målet. De sekundære spilmekanikker bruges en gang imellem eller ofte, men som ikke er nødvendige for at nå spillets mål. Til sidst er der de tertiære spilmekanikker, det er dem som sjældent bliver brugt, og som muligvis kun er tilgængelige for spilleren i bestemte situationer, eller ved bestemte steder i spilverdenen (Larsen, Spilmekanikker, 2022).

Spillet har som sagt flere mekanikker, men kun én af dem er kernemekanikken, som de andre mekanikker er med til at understøtte. Kernemekanikken er én enkelt mekanik. Det er den centrale mekanik som spillet er baseret på. I forlængelse af kernemekanikken er der kerneformålet. Kerneformålet er grunden til at bruge kernemekanikken. Pointen med at fokusere på en kernemekanik i spiludvikling er at sikre sig at spillet forbliver simpelt og letforståeligt for spillere. (Larsen, Spilmekanikker, 2022).

I projekt sandkasse er der kun tre spilmekanikker. Den første er sandformnings mekanikken, som er kernemekanikken i spillet og går ud på at give spilleren mulighed for at forme sandet. Sandet kan formes både ved at lave bunker, huller og alle andre mulige konstruktioner af sandet. Spilleren kan ombygge spilverden ved at benytte sig af kernemekanikken. Sand bunkerne vil blive vist som bakker og bjerge, alt efter højde. Huller i sandet vil blive vist som floder, søer og lignende. Da det at forme sandet er en meget stor del af spillet og hovedsageligt det spilleren kan gøre for at ændre landskabet, er denne mekanik en primær mekanik, det er også spillets eneste primær mekanik. Spilleren har fri mulighed til at forme sandet, de eneste restriktioner er fysikkens love.

Mekanikken er udviklet med en forhåbning om at når spilleren fysisk er nødt til at forme landskabet med deres egne hænder vil oplevelsen føles mere personlig og spilleren vil leve sig mere ind i verdenen. En spiller som selv har opbygget et landskab og har en personlig forbindelse til det, vil

forhåbentligt også kunne mærke følelser af magi og vidunder. Mere simpelt, så er kerneformålet i spillet at spilleren bygger sit eget personlige landskab.

De to sidste mekanikker er begge sekundære og meget ens. Mekanikkerne giver henholdsvis spilleren adgang til at justere landskabets temperatur og vandstand. Ved at indstille landskabets temperatur kan spilleren gøre landskabet enten koldere eller varmere. Visuelt er dette repræsenteret ved at farverne på landskabet ændres til bedre at fremstå som den aktuelle temperatur. Ændringen af vandstanden er som det lyder, spilleren kan justere hvor højt vandstanden er i landskabet. Begge parametre justeres ved brug af drejeknapper på siden af sandkassen. Knapperne er ikke markerede i håb om at spilleren vil opdage dem naturligt under spillet.

Interaktionsdesign

Interaktionsdesign går ud på at udvikle brugbare produkter. Et brugbart produkt er nemt at lære, og nemt og effektivt at bruge samt det er behageligt for brugeren at udnytte. Godt interaktions design er meget afhængigt af produktets brugergruppe. En teknisk brugergruppe, f.eks. brugere af styresystemet Linux, som har brug for mange funktioner, muligheder og frihed til at bruge produktet kræver en mere avanceret måde at interagere med produktet. En mere simpel brugergruppe, f.eks. brugere af Apples iPad, har ikke brug for så mange avancerede indstillinger og interaktions muligheder. I det tilfælde er det derfor bedst at holde interaktionen simpel, let forståelig og dette kan bl.a. gøres ved at skære ned på produktets funktionalitet, så det bedre passer til brugergruppens formål med at bruge produktet. Under processen af interaktionsdesign er det vigtigt at stille spørgsmålet: "Hvordan optimeres brugernes interaktion med et system, miljø eller produkt, på sådan en måde at det understøtter brugernes aktiviteter på effektive, nyttige, brugbare og behagelige måder?" (Sharp, Preece, & Rogers, 2019).

Visionen med projekt sandkasse er et arkadespil, som ikke kræver en introduktion. Interaktionen skal derfor være så simpel og selvindlysende som muligt. Når spilleren former sandet i forskellige højder får de straks visuelt feedback, der viser at sandets højde i sammenspil med projekteres lys resulterer i et topografisk landskab. Knapperne til at justere temperatur og vandstand, har som sagt ikke nogen symboler. Ingen symboler gør at det er sværere for spilleren at forstå hvad de gør ved første øjekast, hvilket gør interaktionen mere avanceret. Symbolerne er fravalgt på trods af den mere avancerede interaktion, for at videre understøtte spillets opdagelses og nysgerrigheds elementer. Selvom intet viser hvad knapper gør, er der en forhåbning om at spillerens nysgerrighed vil frembringe en interaktion. Som sagt før, så når spilleren interagerer med knapperne, vil de blive mødt af både visuelt feedback, enten i form af vandstanden der stiger, eller farverne i landskabet der enten bliver koldere eller varmere, samt musikken i baggrunden vil ændre sig til bedre at passe til landskabets tema.

Produktudvikling

Idégenerering

Første fase af vores projekt, var idégenerering. Vi skulle finde en ide, som opfyldte alle de formelle krav for opgaven, samtidig med at opfylde gruppens personlige krav om at blive udfordret fagligt. Vi startede derfor med en fælles brainstorm, hvor vi fandt på alle de ideer vi kunne. På dette tidspunkt i processen tænkte vi ikke på kravene endnu, da vi gerne ville have et bredt grundlag for vores ideer, samt ville vi ikke begrænse vores ideer inden de blev fremlagt. Når først der tænkes krav ind i en brainstorm, bliver det dårligt for processen, da ideer allerede bliver frasorteret inden de

præsenteres for gruppen. Vi ville undgå denne form for begrænsning, eftersom at tilsyneladende dårlige ideer godt kan videreudvikles. Det var også vigtigt for vores proces, at vi heller ikke begrænsede os selv ift. hvor realistisk en ide var, da det igen kunne resultere i at gode ideer frasorteres.

Efterfølgende tog vi et tættere kig på alle vores ideer og prøvede så at identificere om de levede op til de formelle krav som målgruppen, interaktionsinterfacet, og interessen. Eftersom at vi selv var en del af målgruppen, som var: "HHX/HTX-elever på skolen", vurderede vi primært appellen til målgruppen ud fra os selv.

Brainstorm

Under vores brainstorm endte vi ud med 6 grundlæggende ideer, som vi mente kunne opfylde kravene for opgaven (Se bilag 1). Den første ide var en vippemaskine, som brugeren selv har kontrol over. Altså brugeren styrer ved hjælp af et håndtag, om maskinen vipper til højre, venstre, op eller ned. Vores anden ide var en sandkasse simulation, hvor brugeren kan grave i sand for at skabe floder, stable det for at lave bjerge, eller noget helt tredje. Udstyret med eventuelle sliders eller lignende til at lege med temperatur, vandstand, årstid eller andet. Den tredje ide gik ud på at lave en robot, som er styret af høje og lave toner. F.eks. lave toner får den til at dreje til højre, høje toner for den til at køre fremad. Robotten skal styres igennem en labyrint. Så var den fjerde ide, at lavet et "grid" med distancemålere, som måler afstanden til en hånd. Afstandene bestemmer hvilke lyder der bliver lavet, som så også bliver visuelt repræsenteret af en robotohånd. Og den femte ide var at lave et hologram af en masse projekter, som projekterer en skulptur. Skulpturen skal designes af brugeren. Sidste ide var en robotohånd styret af en handske. Altså en handske der kan tages på, og en robotohånd som efterligner ens egne håndbevægelser.

Derudover opstillede vi også nogle yderligere krav til ideerne, som vi i gruppen bestemte.

Vi havde en overordnet ide om hvilken retning vi gerne ville have projekt skulle føres. Forinden projektets opstart lavede vi en indledende brainstorm øvelse, hvor vi skulle finde eksempler på arkademaskiner og beskrive oplevelser og unikke aspekter ved arkademaskine. Vi valgte derfor nogle oplevelser og unikke aspekter, som vi ønskede at medføre i vores projekt. Sammen med det og nogle udvalgte krav, vi i gruppen havde til projektet, opstillede vi derfor følgende krav: Det skal føles magisk at bruge, være nemt at lære, være underholdende, have et passende ambitionsniveau, indeholde interaktivt lyddesign, være velegnet til arkade.

Evaluerings af brainstorm og idegenerering

Vi brugte kravene til at afgøre hvilken ide var bedst til projektet. Vi satte ideerne op imod hinanden i en kravmatrix og gav dem en værdi mellem 1 og 5. 1 værende "Opfylder kravet meget dårligt" og 5 værende "Opfylder kravet meget godt". Vi valgte at vægte kravene, så "Velegnet til arkade" kravet bliver vægtet med 2, fordi det er en af de formelle krav i opgaven og dermed også er vigtigst for projektet. Derefter kom "Det skal føles magisk at bruge". Dette krav en af de krav vi i gruppen var blevet enige om, skulle fylde en stor del af projektet, samt var det også en af de udvalgte oplevelser og unikke aspekter fra den indledende brainstorm, og det blev derfor vægtet med 1,8. Dernæst var det "Det skal være underholdende" kravet, som er essentielt for at fange interessen af målgruppen, som igen er et af opgavens krav, så det fik vægtningen 1,6. Derudover var der også kravet om "Det skal indeholde interaktivt lyddesign", som var en af gruppens største krav til opgaven, hvilket også afspejles i mængden af ideer der indeholder lyddesign. Dette krav blev vægtet med 1,4. Så var der kravet om "Det skal være nemt at lære", som hænger sammen med den spilteori vi endte med at gå med, samt resultaterne fra den indledende brainstorm. Den blev derfor vægtet med 1,2. Til sidst er

der kravet om "Det skal have et passende ambitionsniveau", som blev vægtet med 1, fordi det ikke er ligeså vigtigt som flere af de andre krav, men stadig er vigtigt ift. realisering af produktet. Dog viste det sig at vægtningen ikke gjorde meget ved det endelige resultat. Den eneste umiddelbare ændring, er definering af en 4. og 5. plads, frem for en delt 4. plads af hhv. tone robotten og handske robotten (Se figur 2 og 3)

Score 1-5	Vippemaskine	TUI	Tone robot	Lyd board	Hologram	Handske
Det skal føles magisk at bruge	3	5	2	4	5	3
Nemt at lære	5	4	3	3	1	5
Underholdende	3	3	5	2	3	2
Passende ambitionsniveau	3	5	2	3	1	4
Interaktivt lyd design	3	5	5	5	2	1
Velegnet til arkade	5	4	1	3	3	3
I ALT	22	26	18	20	15	18

Figur 2: Screenshot af u-vægtet kravmatrix

Vægtning	Score 1-5	Vippemaskine	TUI	Tone robot	Lyd board	Hologram	Handske
1,8	Det skal føles magisk at bruge	5,4	9	3,6	7,2	9	5,4
1,2	Nemt at lære	6	4,8	3,6	3,6	1,2	6
1,6	Underholdende	4,8	4,8	8	3,2	4,8	3,2
1	Passende ambitionsniveau	3	5	2	3	1	4
1,4	Interaktivt lyd design	4,2	7	7	7	2,8	1,4
2	Velegnet til arkade	10	8	2	6	6	6
	I ALT	33,4	38,6	26,2	30	24,8	26

Figur 3: Screenshot af vægtet kravmatrix

Dog har vi efterfølgende kunne konkludere at der måske havde været en lidt forvrænget ide om det passende ambitionsniveau, eftersom at vi havde anset TUI ideen som en 5 ud af 5, hvilket viste sig ikke at passe. Dog havde det ikke ændre resultatet af valget, eftersom TUI-ideen stadig ville vinde, selvom det passende ambitionsniveau havde været 1. Dette er dog kun sandt for den vægtet version af kravmatrixen, hvilket er med til at vise hvordan vægtningen af kravene faktisk kunne have gjort en forskel, hvis vi havde haft et realistisk syn på TUI-ideens omfang.

Koncept/Design

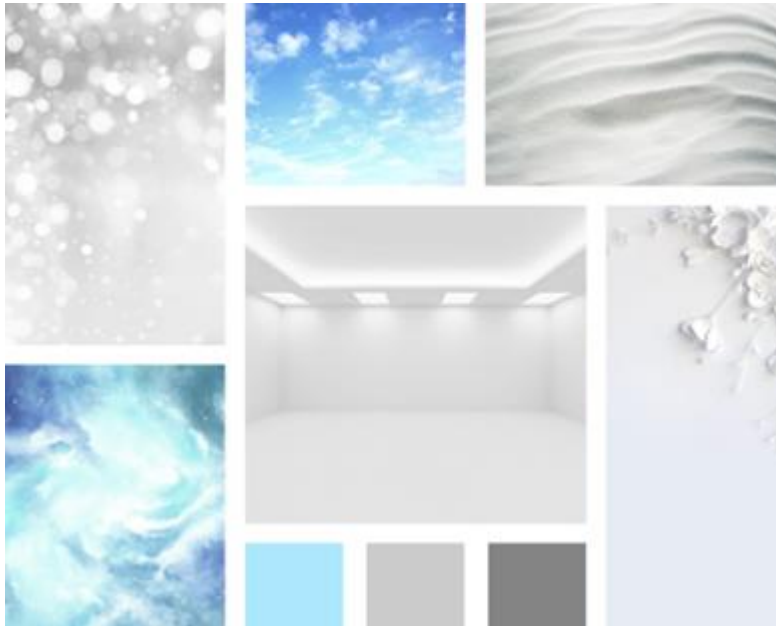
Det endelige koncept endte med at blive et tangible user interface (TUI), hvor højden på sand blev målt. Ned på sandet blev projekteret et landskab efter de målte højder, sådan at brugeren kunne lave et landskab i sandet. Der blev også planlagt at lave en beholder med et visuelt organisk udtryk, et dynamisk lyd design, sådan at lyden skiftede efter hvordan brugeren indstillede vandhøjden og temperaturen.

Hvis der var tiden til det, blev det yderligere planlagt at lave simuleringer af dyre- og planteliv på landskabet, som også skulle være afhængig af målingerne og de nuværende indstillinger. Dette samt et organisk visuelt indtryk blev dog ikke implementeret.

Moodboards

Der blev lavet tre moodboards, for at fange forskellige udgaver af den ønskede stemning. Der blev i gruppen talt om flere idéer til, hvor overordnet blev talt om henholdsvis en hvid minimalistisk, en sort minimalistisk og en rustik/naturfarvet stil. Moodboardene er lavet i Canva, alle med tre valgte temafarver, og indsat i den samme skabelon. Dette er for at minimere indtrykket på brugeren fra skabelonen selv. Dog nåede det visuelle indtryk ikke for alvor at blive implementeret i projektet grundet tidsmangel.

Hvid minimalisme

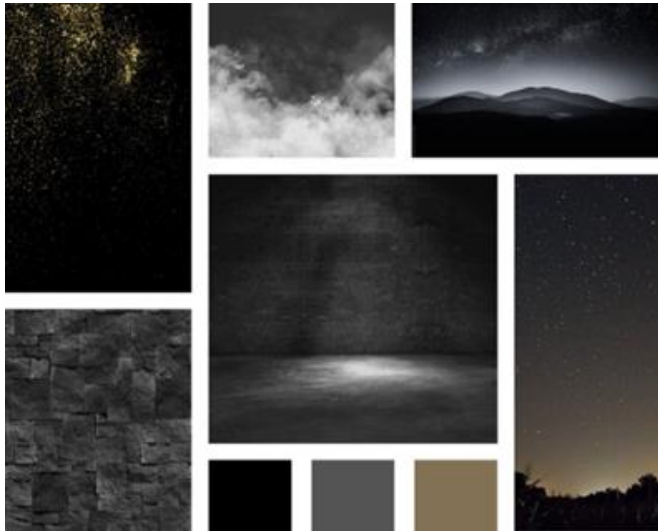


Figur 4: Moodboard af "Hvid minimalisme"

I det hvide minimalistiske moodboard (Se figur 4), er der forsøgt at indkapsle en følelse af guddommelighed, og en følelse af, at der kreeredes noget på et blankt kanvas. Her er målet, at brugeren skal blive påvirket mindst muligt i forhold til, hvordan de skal udfolde deres kreativitet, men i stedet sætte et fokus på, hvad der er bygget i sandkassen.

Derfor er der blevet valgt farverne hvid, grå med highlights af blå, til at skabe associationer til himlen, som den typisk bliver fremstillet i populærkultur. Yderligere da målet er at skabe en minimalistisk følelse, så billederne valgt til moodboardet er med få, simple motiver, som er unison farvede.

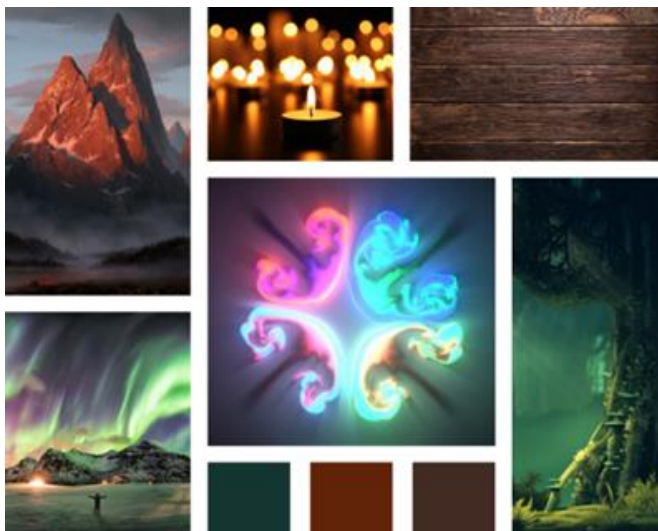
Sort minimalisme



Figur 5: Moodboard af "Sort minimalisme"

I dette design blev der fokuseret på sort minimalisme med guld highlights (Se figur 5). Der blev her fokuseret på at give en luksuriøs følelse, dog stadig med fokus på, at Sandkassen er i fokus, og ikke skal overvældes af andre indtryk. Som temafarver er der valgt sort, grå og guld, og billederne er fokuseret på sorte sten, for at give en anelse rå fornemmelse. Der er også blevet valgt billeder af nattehimmeler for at associationer til universet og det magiske der kan være i, at se på en stjernehimmel.

Rustik natur



Figur 6: Moodboard af "Rustik natur"

Moodboardet med temaet rustik natur (Se figur 6), er det en følelse af natur og magi, som er i fokus. Her er der taget temafarverne mørkegrøn, rød og brun, og der er brugt billeder af store naturoplevelser som bjerge, nordlys og skove. Ud over det, er der taget inspiration fra den mystiske magitradition. Billederne er ikke så strikte i farvevalget her, da det er meningen, det skal give et mere kaotisk indtryk.

Prototyper

Der blev undervejs lavet en række prototyper af forskellige dele af projektet. Overordnet blev der lavet tre prototyper. To forsøg til at teste hvordan projektoren og Kinekten skulle indstilles, og en til at test, om højdemappet på Kinekten virkede.

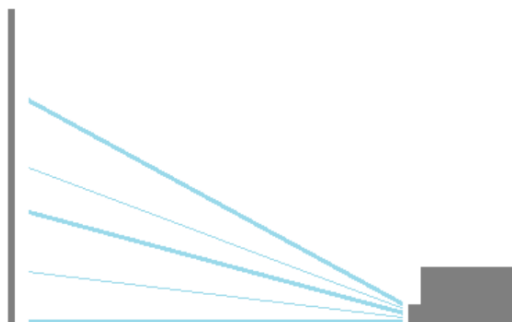
Prototype 1

I den første prototype, blev det forsøgt at finde et system, hvor positionen af projektoren ville kunne kalibreres løbende, også efter systemet blev sat op. Dette blev forsøgt med et snorsystem, hvor projektoren blev stabiliseret med reb. I forsøget blev projektoren spændt ud i hvert sit hjørne for at stabilisere projektoren, som set på figur 7. Den blev lavet ved at lasercutte henholdsvis to plader, til at fastne projektoren med 6 mentalstænger. Yderligere blev der lasercuttet en skive for at trække snore igennem, for at opnå en større stabilitet, ved at have et udspreddt tyngdepunkt. Yderligere blev der brugt reb til at binde det hele sammen.



Figur 7: Billede af test på snorsystem til op hejsning af projektor

Dog viste det sig, at dette var mere ustabil end først forudset, og at projektoren svang mere i rebet end forventet, og derfor blev løsningen ikke implementeret i det endelige produkt. Dog blev det på denne måde fundet ud af, at projektoren ikke skød ned på centeret af bordet, men ned på kanten og op, som vist på figur 8. Yderligere blev det opdaget, at projektoren skulle være omkring 1.85 meter fra bordet.



Figur 8: Skitse af projektorens projektionsvinkel

Prototype 2

Efterfølgende blev der lavet en anden test, for at måle afstanden mellem projektor og skærm. Her blev kinekten også første gang sat sammen med projektoren. Her blev bordpladen, som var planlagt at bruge til projektet, sat op mod en væg, hvor projektoren blev placeret sådan, at den fyldte mest muligt af bordpladen. Herefter blev afstanden målt mellem projektoren og bordet, og der blev markeret på bordet, hvor meget der skulle skæres af. Dog blev der her ikke taget hensyn til, at sandet i kassen på bordet også havde en højde, hvilket betød, at projektoren ikke udfyldte hele billedet i det endelige produkt.

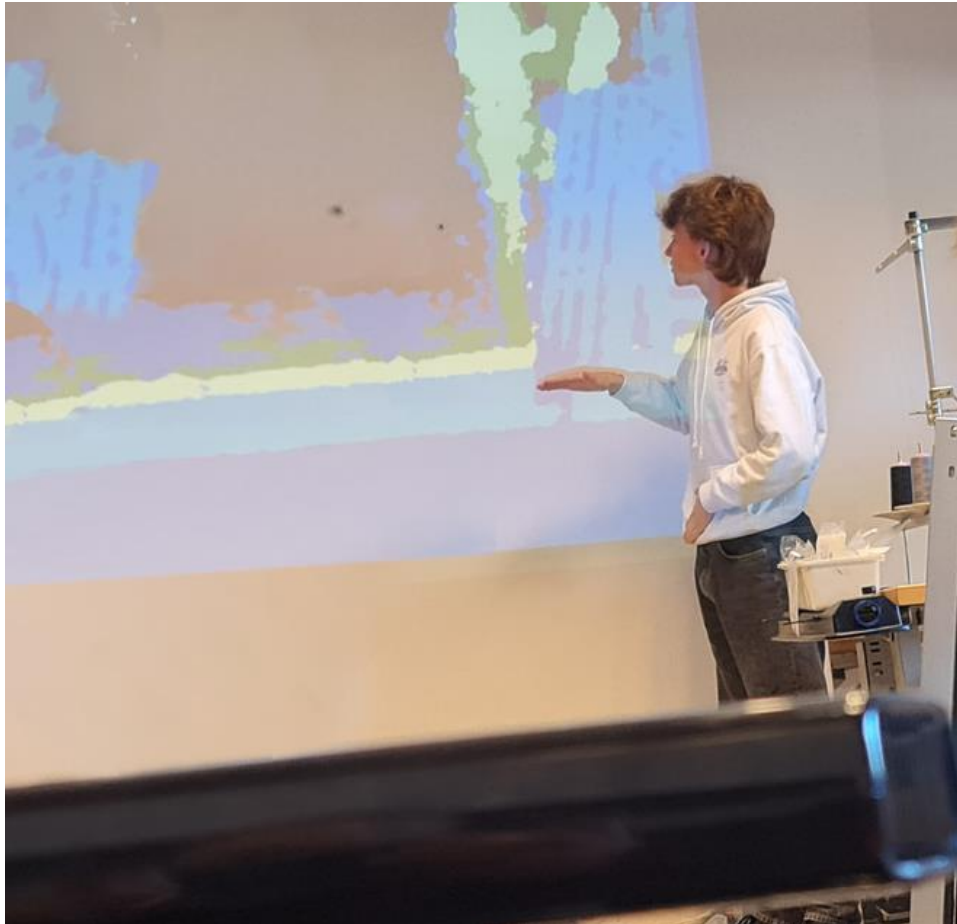
Der blev også fundet ud af, at kinekten skulle stå så tæt på projektorens linse som muligt. Dette problem blev senere løst ved at skrive programmet sådan, at det var muligt at flytte på det projekterede stykke af billedet (Se figur 9).



Figur 9: Billede af test og opmåling af projektoren på bordpladen

Prototype 3

Der blev yderligere lavet en koncepttest for at undersøge, hvordan kinecten virkede med en tidlig udgave af unityprojektet. Dette blev gjort ved at skyde projektoren op mod en væg med kinecten ovenpå, og herefter blev der testet med forskellige afstande, for at se hvor hurtigt den opdaterede billedet på dette tidspunkt, hvor store afstande den målte, hvor præcist den målte dem, hvor den målte i forhold til sensorens placering, osv. På dette tidspunkt viste det sig, at den kun kunne opdatere én gang i sekundet, hvilket betød programmet skulle optimeres, før brugeren fik fornemmelsen af, at det opdaterede i real time (Se figur 10).



Figur 10: Billede af test af kinect sammen med projektor

Projektstyring

Fra projektets start fastlagde vi nogle roller, der beskrev hvert gruppemedlems primære opgaver, hvor der blev lagt vægt på medlemmernes individuelle kompetencer:

Astrid	Visuelt design
Anders	Konstruktion og hjælp efter behov
Mikkel Blom	Konstruktion og programmering af Arduino
Mikkel Erikstrup	Lyddesign og udvikling af sound engine
Rasmus	Programmering i Unity og kompatibilitet med Kinect

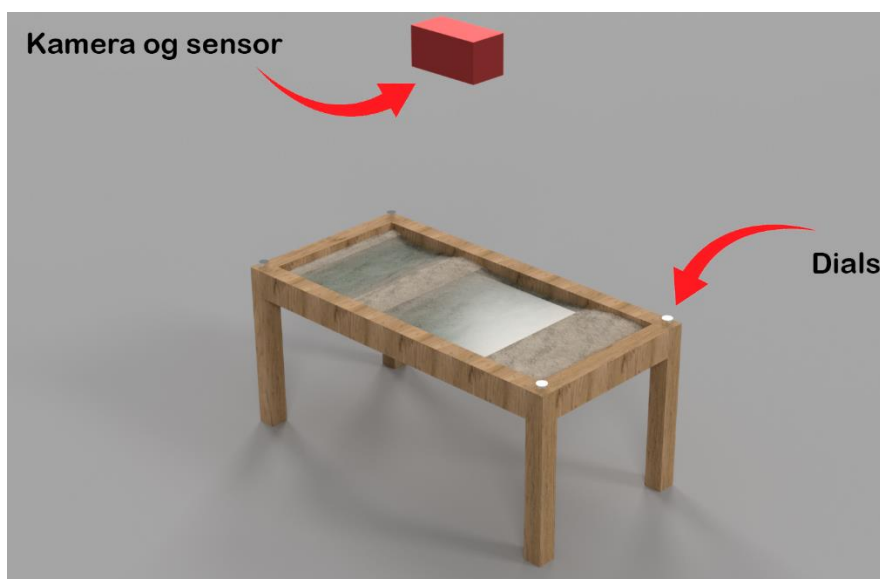
Rollerne fungerede godt og gav en god ramme for hvad der var forventet af hvert gruppemedlem uden at virke begrænsende - ved behov eller overskud skiftede vi rundt til de opgaver, der var mest hensigtsmæssige. Hvis noget skulle ændres i rollefordelingen, ville det nok være fornuftigt at justere omfanget af visse roller løbende; et konkret eksempel er Rasmus' opgave med programmering i Unity og kompatibilitet med Kinecten, der hurtigt viste sig at være en større opgave end først antaget. Omvendt var rollen med visuelt design forholdsvis lille, og Astrid endte med at arbejde på andre opgaver, hvor behovet var større.

Vi fravalgte i gruppen at lægge en fast tidsplan, da vi ikke mente det ville være hensigtsmæssigt i vores tilfælde - i stedet besluttede vi at have en flydende tidsplan, så der også var plads til andre af gruppemedlemmernes projekter, der lå oveni sandkasseprojektet. For at nå den endelige deadline mødtes vi i fritiden og arbejdede på projektet, og aftalte lektier mellem modulerne i skolen.

Produktion

Bord og reelt design

Produktionen af bordet gennemgik en iterativ design og konstruktionsproces. Med det menes der, at vi gennemgik flere forskellige designs af bordet, før vi endte med det endelige produkt. Vi benyttede denne metode fordi vi fik ny viden hver gang vi kom op med en koncept. Denne nye viden gjorde det muligt at gennemtænke en ny og bedre måde at udføre konstruktionen. Før vi overhovedet begyndte at bygge noget, brugte vi meget tid på at planlægge konstruktionen via. Programmet Fusion 360. Faktisk begyndte vi ikke at bygge produktet før vi var nået til vores endelige 3D model. Vores første iteration så sådan her ud (Se figur 11). På dette tidspunkt i processen havde vi ikke nogen ide om hvordan projektet skulle laves, hvor mange dials der skulle bruges, hvordan projektoren skulle monteres, eller hvordan vi skulle måle højden. Denne iteration var med til at få styr på flere af disse problematikker.



Figur 11: Render af første 3D mockup af bordet

Det første vi tog os til efter at have lavet denne mockup, var at lave en MoSCoW model. MoSCoW står for: Must Have, Should Have, Chould Have, og Will Not Have. Dog lavede vi ikke en "Will Not Have" kategori, da vi ikke mente det var nødvendigt at skrive alle de ting ned, vi ikke havde tænkt os at lave. Vi nedbrød så projektet til dets mest minimalistiske tilstand, hvor det stadig ville opfylde kravene. I "Must Have" havde vi følgende: Vi skulle have et bord, som vi kunne projektere på, Vi skulle have en projektor der kunne projekter et landskab på bordet, Vi skulle have en måde at måle afstanden fra sensor til bordet, vi skulle have sand som kunne modelleres med, vi skulle have et program, som behandler dataen og generere et topografisk billede ud fra den data, og til sidst skulle vi have en form for lyddesign. Disse elementer skulle være på plads, før vi kunne tilføje andre elementer til projektet. Se bilag 2, for resten af MoSCoW modellen.

For at opfylde de minimumskrav vi havde sat, skulle vi lave et bord med en projektor og sensor. Vi begyndte derfor at udforme det reelle design for bordet. Allerede ud fra første mockup, blev vi enige om at nedskalere, på grund af mængden af arbejdsopgaver vi havde. I første mockup, var bordet 1x2 meter. Det besluttede vi os for, var for stort. Vi tog derfor inspiration fra undervisningsbordene i på skolen, som er 1,4x0,7 meter indvendigt. Det virkede som en nogenlunde størrelse, så brugeren kunne nå det meste af bordet, samtidig med at der kunne være flere omkring bordet på en gang.

Derudover var undervisningsbordene også en god højde til at alle i gruppen nemt kunne nå bordfladen uden at skulle bukke sig eller strække sine arme for meget. Derfor valgte vi at bordet skulle have lignende dimensioner med undervisningsbordene, både i højde, bredde, og længde. Ud fra dette begyndte vi så at fremstille en ny 3D model til bordet. Da vi nu vidste hvor stort bordet kom til at være, begyndte vi at regne på hvor meget sand vi skulle bruge til at fylde bordet. Det eneste vi manglede til at udregne rumfanget af bordet, var dybden på sandet. For at finde ud af hvad den bedste højde var, udførte vi en hurtig test. Læs mere om dette under kvalitetssikring. I testen fandt vi frem til at en ideel højde er omkring 10 centimeter. Ud fra dette, kunne vi regne $1,4 * 0,7 * 0,1 = 0,098 m^3$. Det svarer til 98 liter. Eftersom at vi havde nogle overvejelser omkring omkostningen af projektet, valgte vi at finde det billigste sandkassesand vi kunne finde. Der var overvejelser om at bruge andre former for modellerbart materiale, som f.eks. magnetisk sand, men vi fravalgte det og endte med at gå med normalt sandkassesand. Det sand vi fandt, var 12 liter pr. 20 kg. Det svarer til at vi skulle bruge lidt over 160 kg sand til at fylde bordet til den betragtede højde. Dette fik os til at overveje stressen bordet ville gennemgå af den store vægt. Noget andet vi også overvejede, var tabet af sand. Her kom vi op med ideen om en skrå kant, der stak ud over bordet, som kunne fange sand der skulle blive slynget ud fra bordets omfang. Til det testede vi flere forskellige vinkler til den eventuelle skrå kant, for at se hvilken vinkel, der bedst ville lade sandet falde tilbage i kassen, samtidig med at være mindst i vejen for brugeren.

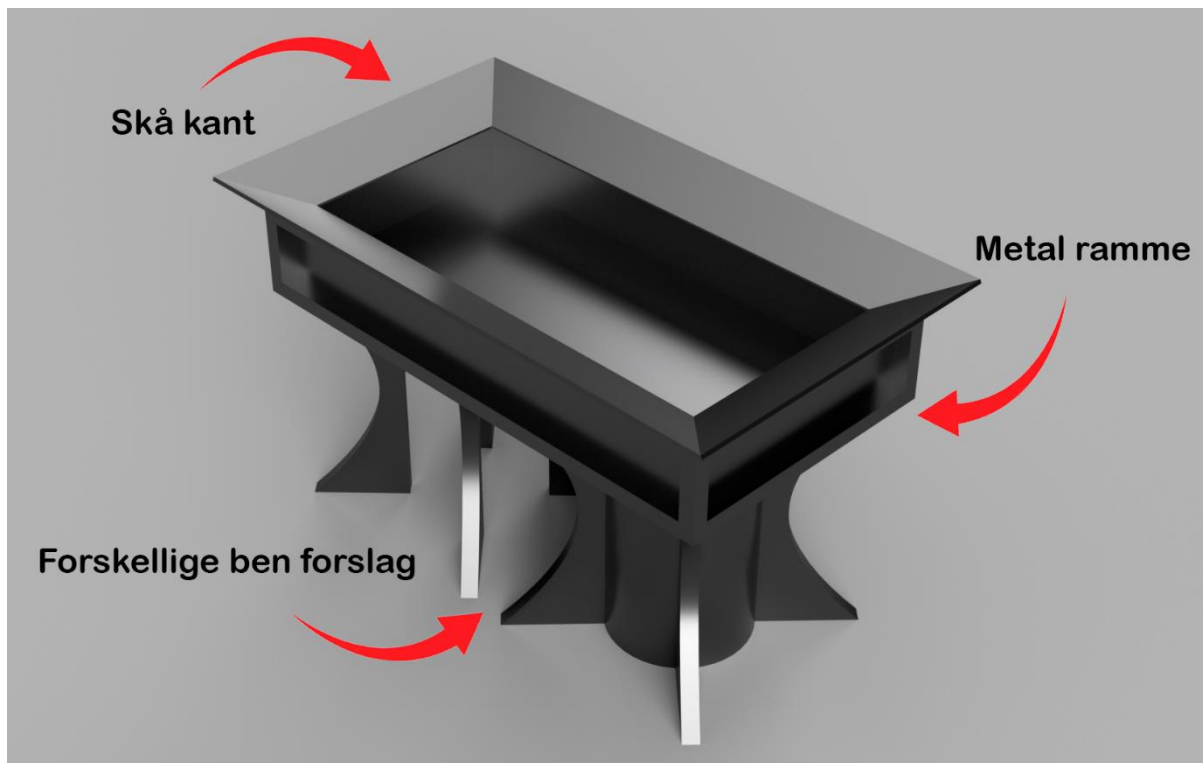


Figur 13: Billede af sandvinkeltest på 10 grader



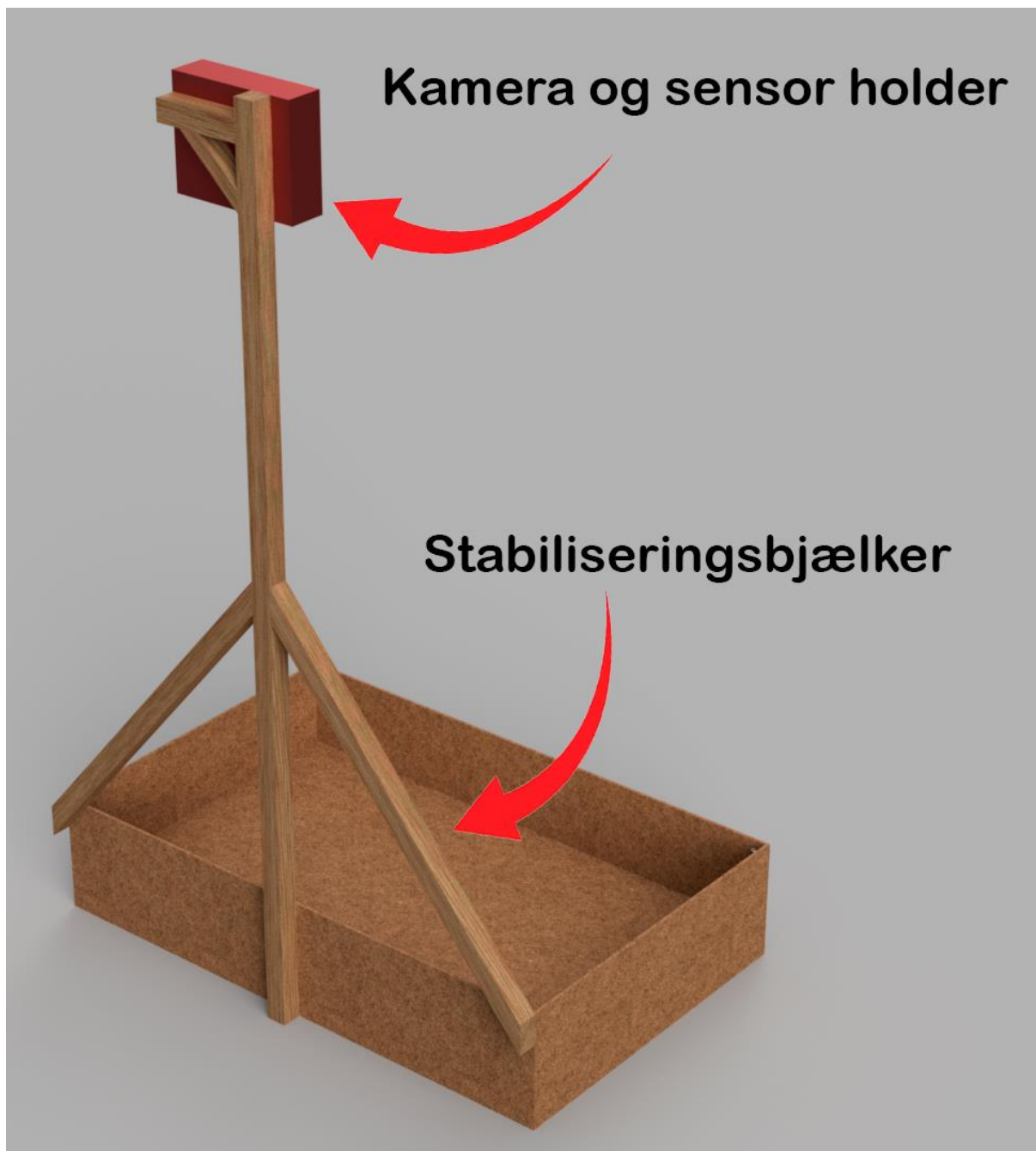
Figur 12: Billede af sandvinkeltest på 55 grader

Vi testede vinklerne mellem 10 grader (Se figur 13) og 55 grader (Se figur 12), hvor vi øget vinklen med 5 grader for hver gang. Derudover dryssede vi både langsom og hurtigt sand på pladen, samt bankede vi lidt på plade, for at se hvor nemt det ville falde tilbage i spanden. Vi endte med at finde frem til en vinkel på 25 grader, som var bedst til at lade sandet falde tilbage, samtidig med den ikke var for upraktisk at sætte på og stå ved. Så ideen med tilbageholdelsen af sandet, samt den førhen ukendte vægt, resulterede så i endnu en iteration af bordet, nu med en mere detaljeret model (Se figur 14).



Figur 14: Render af andet mockup af bordet, samt ben design

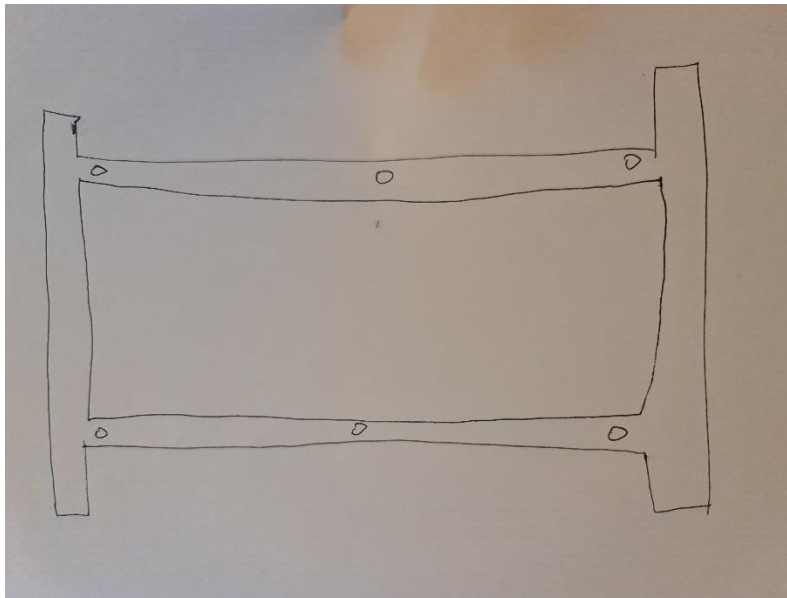
Eftersom at vægten var meget højere end vi originalt havde tænkt, valgte vi at efterspørge en faglærer ift. hvordan en ramme for bordet bedst kunne konstrueres. Faglærer Lotte Dahl Andersen, mente at, pga. vægten, vi skulle konstruere bordet i metal. Dog efter eftertanke fra gruppe, valgte vi ikke at forfølge dette forslag yderligere, eftersom at vi i gruppen mente at det blev for besværligt at skulle bestille et metal arbejdet version af bordet, blandt andet grundet vores iterative designproces, som lagde op til at vi kunne blive nødt til at ændre modellen. Og når det kommer til at skulle lave ændringer i hhv. træ eller metal, er det nemmest at skulle modificere en træversion. Derfor besluttede vi os for at designe fremtidige modeller med træ som hovedmaterialet. Dog mente vi stadig at det ville blive svært for benene at klare stressen fra alt den vægt, hvis vi lavede dem ud af trælægter. Umiddelbart holdt vi os til at bruge 45x45mm lægter i træ, eftersom de var billige, nemme at skære, og vi havde allerede flere lægter på lager. Dog er 45x45mm læger ikke stærke nok til at kunne holde den vægt, hvis det stadig så pænt ud. På dette tidspunkt i processen havde vi stadig flere overvejelser omkring udseendet af bordet, eftersom vi endnu ikke var nået i tidsvanskeligheder. Derfor ville det kræve at købe nye, tykkere, stolper ind til ben. Det var ikke en ideel mulighed, eftersom vi prøvede at holde omkostningerne nede. Derfor valgte vi at gå med en alternativ løsning, hvor vi ville genbruge et bord, der ikke længere blev brugt. Dette bord kom med en 2 cm tyk bordplade, samt metalben. Dette var mere end nok til at holde vægten fra sandet, projektoren, sensoren, og holderen til det hele. Dette valg sparede os både for at skulle konstruere ben, støtterammer, og bordplade. Det gjorde så, at vi kunne fokusere på hvordan projektoren skulle monteres på bordet. Indtil nu, havde tankegangen været, at bordet skulle kunne tilgås fra alle sider, men vi havde stadig ikke fundet ud af hvordan projektoren skulle monteres. Det ledte så til vores tredje og sidste mockup (Se figur 15).



Figur 15: Render af tredje mockup af bordet, samt projektorholder

I vores sidste iteration af bordet, havde vi lavet nogle ændringer ift. designet. For det første besluttede vi os for at fjerne de skrå kanter på bordet. Dette gjorde vi delvis fordi vi var begyndt at komme tæt på afleveringsdatoen og havde stadig ikke påbegyndt produktion af bordet. Derudover mente vi også at den skrå kant ville blokere for projektorens lys, samt ville den bare samle sand, i stedet for at hjælpe med at holde sandet inde. Derudover valgte vi også at gøre så det ikke længere var muligt at tilgå bordet fra alle sider. Dette gjorde vi både på grund af muligheden for at stille bordet op ad væggen, hvilket ville hjælpe med stabilisering af projektor holderen, men også for at kunne montere stabiliseringsbjælkerne som kan se på figur 15. Stabiliseringsbjælkerne er med til at holde projektorholderen stille, så den ikke forstyrres for meget af interaktionen med bordet. Dog var der opstået et par problemer ved at genbruge et bord. For det første var dimensionerne på bordet

lidt anderledes. I stedet for 1,4x0,7 meter, var det nu 1,23x0,762 meter. Dette var ikke det største problem, da bordet nu kun skulle bruge omkring 156 kg sand i stedet for 163 kg sand. Men derudover var der også en afvigelse i højden på bordet. Vi var blevet enige om en højde på det samme som et normalt undervisningsbord. Dette kunne vi dog fikse, samtidig med at fikse et andet problem. Fordi vi havde et problem med at sætte bordpladen fast direkte til metalbenene, eftersom det ville kræve stor præcision at borde direkte gennembordpladen til hullet i metallet. Metalbenene så nemlig sådan her ud (Se figur 16).



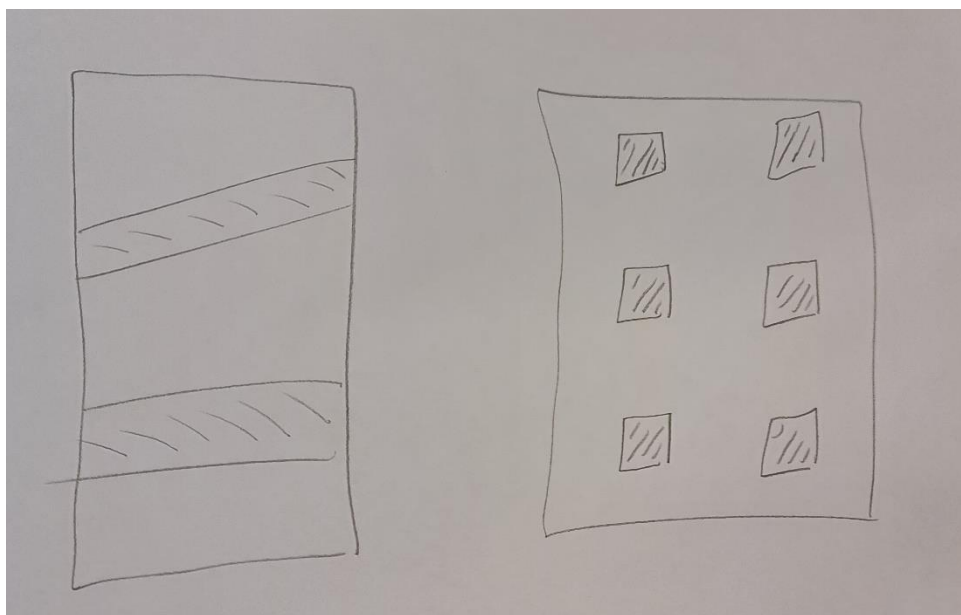
Figur 16: Skitse af bordets metalben set oppefra

Problemet med højden og problemet med montering af bordpladen, kunne dog løses på en gang. Fordi vi fandt ud af at højdefvigelsen næsten var det samme som bredden på en lægte (Se figur 17).



Figur 17: Billede af afvigelsen i højde fra bordene

Fordi, hvis lægterne kunne monteres ovenpå hullerne, kunne bordpladen blive bordet fast til lægterne efterfølgende. Denne løsning kom dog med sine egne overvejelser. Igen, fordi vi tænker over materialebruget og omkostningerne, overvejede vi to forskellige måder at påmontere lægterne til hullerne. Den første løsning (Se figur 18, til venstre) ville det være lægter hele venen over midten, hvilket ville give større fejlmulighed til montering af bordpladen, men også give bedre fordeling af vægt, dog med den ulempe, at det ville koste mere. Og så var der den anden mulighed (Se figur 18, til højre), som ville være billigere, men det ville påvirke vægtfordelingen negativt, samt gøre det svære at montere bordpladen. Vi valgte derfor første mulighed.



Figur 18: Skitse af monteringsmulighederne for lægterne mellem bordpladen og bordbenene

Herefter kunne sidepladerne skæres ud til de rette mål, som endte med at blive doublehøjden af sandet, så altså 20 centimeter, plus 57mm ekstra ift. bredden af bordet og lægterne. Monteringen af siderne blev gjort ved at lime og skrue nogle 45x45x45mm klodser fast rundt i kanten af bordet. Eftersom det ville blive dyrt at lave hele kanten til lægte, lavede vi nogle klodser, som blev fordelt ligeligt ud på bordpladens længder. Der var tre klodser på den korte længde, altså en i hvert hjørne, samt en i midten. Og ud fra fordelingsforholdet tre klodser for ca. 76 cm (Bredden af bordet). blev der regnet frem til at der skulle være 5 klodser på ca. 124 cm (Længden af bordet).

Efter siderne blev sat på, blev der monteret holderen til kameraet og sensoren. Den blev skåret i længden, så den ramte gulvet og førte projektoren op i en højde på 185 cm. Lægten rammer gulvet, for at give yderligere støtte. Dette forstærkes yderligere ved at montere stabiliseringsbjælkerne, som er skåret med en vinkel på 50 grader, eftersom det var det højeste der kunne skæres på rundsaven. Hvis vinklen havde været højere, kunne lægten været ført højere op, og kunne have givet mere støtte, dog var dette ikke muligt med den præcision som var nødvendig for udskæringen.

Til sidst blev sensoren monteret til projektoren med monteringslim, og projektoren monteres på projektor holderen med noget patentbånd, for nemt at kunne kalibrere og justere projektoren.

Derudover laserskæres der yderligere en kasse til montering af dials, som monteres i venstre side af forreste plade, for ikke at komme i vejen for oplevelsen.

Musik

Til vores projekt med at konstruere en interaktiv sandkasse har vi besluttet at arbejde med lyd-design som ét af vores punkter i interaktionen.

Designmål

- Et lydbillede der ændrer sig når brugeren interagerer med sandkassen.
 - o Det er vigtigt at lyden til gengæld holder sig forholdsvis stilstående, så længe der ikke foregår en interaktion. Brugeren skal få indtrykket af, at det udelukkende er deres handlinger, der driver både det visuelle og lydlige billede, for at opnå en større taktil fornemmelse i oplevelsen.
- Lydfladen skal være afslappende - akkordskifter skal foregå langsomt, og melodier skal være simple og spredt ud over lang tid.
- Inddrag organiske lyde - f.eks. samplede naturlyde, rigtige instrumenter og levende percussion. For at lydsporet kan passe med naturtemaet, ville det være hensigtsmæssigt at bruge mere "naturlige" lyde, og ikke kun synthesizers.

Tekniske overvejelser

Hele lydenginen kan køre i FL Studio. Hvis der opstilles nogle basistracks med forskellige lyde, kan der fades mellem dem med automation. Det samme kan gøre sig gældende for effekter på de individuelle lydspor. Automationen kan kontrolleres via Unity og et par C# libraries, der kan åbne en virtuel MIDI kanal og dermed bruge Unity som en MIDI controller.

Komposition og produktion af soundtracket

Soundtracket bliver produceret i tre forskellige versioner, der har den samme melodiske og harmoniske struktur, men er bygget op af forskellige elementer. De forskellige versioner bygges ovenpå en underlægning af en kontrabas og en subbas (*se de to gule lag i nedenstående figur 19*). Musikkens tempo er forholdsvis langsomt på 81 BPM (*beats per minute*), for at fremme den afslappende stemning der blev fastsat som designmål.



Figur 19: Screenshot af de forskellige soundtracks

Musikken er delt op i to sektioner. A-sektionen varer otte takter, og gentages to gange med variation i melodien og de afsluttende akkorder. B-sektionen følger en anden akkordprogression, og varer otte takter. Den såkaldte AAB-struktur ses ofte i gentagende musikstykker - i et statisk stykke musik med en fast begyndelse og slutning ville det være mere hensigtsmæssigt at følge en AABA-struktur, da musikken i så fald forløses og bliver til noget lytteren genkender i A-stykket; men da musikken her spiller på et loop, kommer forløsningsen i kraft af at loopet genstarter.

Instrumentering

Instrumenteringen af de tre forskellige versioner af soundtracket kan inddeles i fire forskellige elementer: akkorder, hovedmelodi, arpeggio og dekoration/kontramelodi. De spilles hver især af forskellige instrumenter i de forskellige versioner. Når temperaturen ændres, fades volumen imellem de forskellige versioner.

I den normale version spilles akkorderne af en forholdsvis neutral synth-pad, melodien af en stor plucky synth med ekko og rumklang, og kontramelodien af en syntetisk violin. Arpeggioen er også spillet af en synth pluck, og holder en simpel fjerdedelsrytme og giver dermed noget bevægelse i lyden. Denne version er med vilje holdt rimelig neutral i instrumenteringen, da det selvfølgelig er den version, der skal spille når temperaturen er neutral.

I den varme version spilles akkorderne af en anden synth-pad med en varmere klang; den lyder mere organisk og har et strengeinstrumentlignende præg. Hovedmelodien bæres af en harpe, og kontramelodien af en fløjte, da fløjten for mange har associationer med syden og "de varme lande". Arpeggioen spilles af synthen fra den normale version, der akkompagneres af en akustisk guitarlignende lyd. Rytmen af arpeggioen er også ændret fra kvartdele til en triol-baseret rytme, der giver mere bevægelse til lyden.

I den kolde version spilles akkorderne af en synth-pad med krystallyde i baggrunden. Hovedmelodien spilles af et klaver, og her er tilføjet et par ekstra høje toner undervejs. Kontramelodien er en simpel,

lys synth, og der er ingen arpeggio; det betyder der er mindre bevægelse i lyden, og bidrager dermed til den ønskede kolde følelse.

De sidste lydspor i soundtracket er nogle simple trommer og bølgelyde; deres lydstyrke er styret af hvor meget vand der er i sandkassen - så hvis der er meget vand, bliver trommerne og bølgelydene højere.

Musikteori

Soundtracket er overordnet skrevet i tonearten E-dur. Dur er kendt som værende en forholdsvis positivt lydende og åben toneart, men da der i dette stykke ønskedes en mere surrealistisk og uforløst stemning, anvendes der mange af de såkaldte suspenderede akkorder. Suspenderede akkorder er navngivet efter traditionen for at bruge dem inden den tilsvarende dur-akkord for at suspendere (eller forlænge) spændingen før forløsningen sker. I dette stykke er der mange af de suspenderede akkorder, der aldrig forløses - det bidrager til den svævende, æteriske stemning i musikstykket. Akkordprogressionen i A-stykket går således:

$$Esus2 [I] \rightarrow Asus2 [IV] (\times 3) \rightarrow G\#mol [iii] \rightarrow F\#sus2 [ii]$$

$$Esus2 [I] \rightarrow Asus2 [IV](\times 3) \rightarrow A [IV] \rightarrow F\#sus2/E [ii/I]$$

Den skarpe lytter vil måske bemærke, at der ikke indgår nogle akkorder med dominant funktion ($[V]$ eller $[vii^\circ]$, i E-dur altså B eller $D^\#^\circ$) - dette er gjort bevidst, da dominantakkorder opbygger en uønsket høj spænding, der meget tydeligt skal opløses tilbage til hjem-akkorden E. Dette ville give en for stærk effekt til det som sagt meget svævende musikstykke.

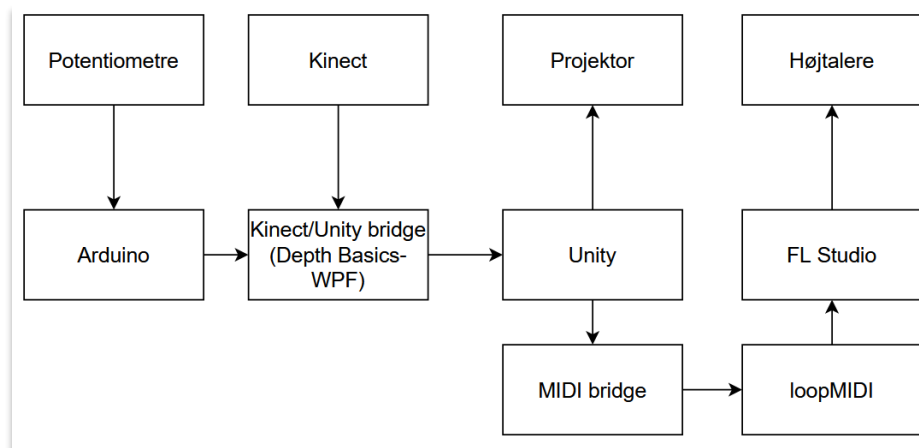
Indtil videre har alle akkorderne været diatoniske - altså tilhørende stykkets hovedtoneart E-dur - men i B-stykket sker nogle ændringer for at holde harmonien interessant:

$$D [bVII] \rightarrow C\#mol [vi] \rightarrow C [bVI] \rightarrow A6sus2 [IV]$$

Her indtræder der et par nondiatoniske akkorder lånt fra den parallelle moltoneart; E-mol. Der er tale om akkorderne D og C, der her medvirker til at bygge videre på stemningen og giver B-stykket en tydeligt anderledes, og mere tonalitetsmæssigt ambivalent lyd; med hvert akkordskifte skiftes der mellem fornemmelsen af mol og dur-tonearter.

Kode

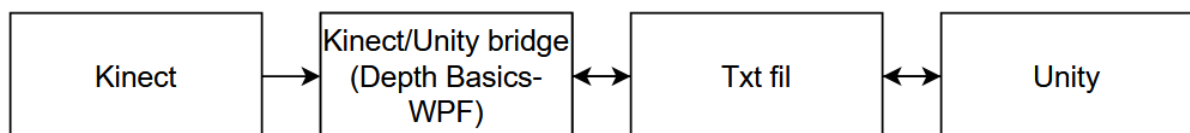
Dette produkt består af mange individuelle dele, der skal kunne kommunikere med hinanden. For at dette er muligt, bliver der nødt til at skrives kode, der kan sørge for at information kan transporteres mellem disse dele (Se figur 20). Se bilag 6 og 7, for kildekode



Figur 20: Skitse af hvordan kommunikationen mellem programmer skal forløbe

Kinect/Unity Bridge & Arduino

En Xbox Kinect 360 bruges til at måle dybden af sandet. Denne dybdeinformation skal Unity bruge. For at transportere dybdeinformationen til Unity fra Kinecten, bruges der en "bridge", altså en bro i form af et program, der både kan kommunikere med Kinecten og Unity. Grunden til at dette skal være et separat program, er at de Kinect libraries, der er designet til Unity er forældet. Microsoft, skaberne af Kinecten, har dog udviklet diverse open-source programmer, der er udviklet til at bruge Kinect data på en computer. Ved at modificere et af disse programmer, kan dybdeinformationen deles med Unity. Programmet, der tages udgangspunkt i her, er et program kaldet "Depth Basics-WPF", der fremover vil refereres til som "Kinect/Unity bridge" (Se figur 21).



Figur 21: Skitse af hvordan information mellem kinecten og Unity bevæger sig

Før programmet blev modificeret, kunne Kinect/Unity Bridge indsamle data fra Kinecten, ved at tilslutte Kinecten til en computer med en adapter. For at videresende denne information til Unity, skriver Kinect/Unity bridge alt dybdeinformationen til en tekstfil, ved brug af en StreamWriter, som Unity derefter kan læse. Dette er langt fra den mest effektive metode, at få to programmer til at kommunikere på, men det er en metode som var garanteret at virke. Det betyder dog ikke at metoden er uden problemer.

```
protected virtual bool IsFileLocked(FileInfo file)
{
    try
    {
        using (FileStream stream = file.Open(FileMode.Open, FileAccess.Read, FileShare.None))
        {
            stream.Close();
        }
    }
    catch (IOException)
    {
        //the file is unavailable because it is:
        //still being written to
        //or being processed by another thread
        //or does not exist (has already been processed)
        return true;
    }

    //file is not locked
    return false;
}
```

Figur 22: Screenshot af IsFileLocked funktionen

Hvis et af de to programmer prøver at læse eller skrive på tekstfilen, imens det andet program er i gang med at bruge tekstfilen, vil det første program crashe. For at undgå dette, er både Unity/Kinect bridge og Unity udstyret med en funktion kaldt "IsFileLocked". Denne funktion prøver at læse tekstfilen, og returnerer en boolean baseret på resultatet. Hvis StreamWriteren i funktionen returnerer en fejlkode, når tekstfilen læses, returnerer IsFileLocked true, ellers returnerer IsFileLocked false. Hver gang Kinecten sender data til Unity/Kinect bridge, bruger Unity/Kinect bridge denne funktion, og hvis tekstfilen ikke er optaget af Unity, skriver Unity/Kinect bridge alt dybdeinformationen i tekstfilen. På samme måde bruger Unity også funktionen hver gang Unity skal læse informationen fra tekstfilen (Se figur 22).

Kinect /Unity bridge har dog også endnu en separat funktion. På samme måde som at Kinecten skal kommunikere med Unity, skal Arduinoen også kommunikere med Unity. Arduinoen er tilkoblet to potentiometre, der hele tiden sender data til Arduinoen om deres rotation. Denne data skal naturligvis også transporteres til Unity. Til dette formål bruges Kinect/Unity bridge også (Se figur 23).

Hver gang Kinect/Unity bridge skriver data til tekstfilen, læser den potentiometrenes værdier via Arduinoen og tilføjer dataet til enden af tekstfilen, separeret med et "X". Dette gøres i Kinect/Unity bridge, da Unity ikke har support til at åbne en seriel port (Se figur 24).

```
if (!locked)
{
    using (serialPort = new SerialPort("COM3", 9600))
    {
        serialPort.Open();
        sw.Write("X" + serialPort.ReadExisting());
    }
    sw.Close();
}
```

Figur 24: Screenshot af programmets læsning af arduino potentiometre værdier

```
// the setup routine runs once when you press reset:
void setup() {
    // initialize serial communication at 9600 bits per second:
    Serial.begin(9600);
}

// the loop routine runs over and over again forever:
void loop() {
    // read the input on analog pin A0:
    int analogValue = analogRead(A0);
    int analogValue2 = analogRead(A1);
    // Rescale to potentiometer's voltage (from 0V to 5V):
    float voltage = floatMap(analogValue, 0, 1023, 0, 5);
    float voltage2 = floatMap(analogValue2, 0, 1023, 0, 5);

    // print out the value you read:
    Serial.print("Analog: ");
    Serial.print(analogValue);
    Serial.print(", Voltage: ");
    Serial.println(voltage);
    Serial.print("Analog2: ");
    Serial.print(analogValue2);
    Serial.print(", Voltage2: ");
    Serial.println(voltage2);
    delay(1000);
}
```

Figur 23: Screenshot af arduino kode

Unity

Unity står for at modtage dataet sendt fra Kinect/Unity bridge, og bruge det til at genererer billedet af terrænet, som projektoren skal vise.

En vigtig funktion i Unity projektets primære script, "MapGenerator", er funktionen, der står for at indlæse dataet fra tekstfilen og opstille det i en todimensionel float array. Denne funktion er kaldt "ParseData", og kaldes hver gang terrænet bliver genereret.

```
float[,] ParseData()
{
    StreamReader Sr = new StreamReader(@"C:\Users\maste\OneDrive\Skrivebord\Data.txt");
    string data = Sr.ReadToEnd();
    string[] dataString = data.Split(' ');

    string arduinoData = data.Split('X')[1];

    string[] arduinoSplitOne = arduinoData.Split(new string[] { "Voltage: " }, System.StringSplitOptions.None);
    string[] arduinoSplitTwo = arduinoData.Split(new string[] { "Voltage2: " }, System.StringSplitOptions.None);

    temperature = Remap(int.Parse(arduinoSplitOne[1][0].ToString() + arduinoSplitOne[1][2].ToString() + arduinoSplitOne[1][3].ToString()), 0, 500, -1, 1);
    heightOffset = Remap(int.Parse(arduinoSplitTwo[1][0].ToString() + arduinoSplitTwo[1][2].ToString() + arduinoSplitTwo[1][3].ToString()), 0, 500, -0.038f, 0.020f);

    Sr.Close();
}
```

Figur 25: Screenshot af ParseData funktionen

Det første ParseData gør, er at åbne tekstfilen, og opdele alle de individuelle værdier i en string array. Det er også her Arduino dataet bliver læst og sat som variablerne "temperature" og "heightOffset" (Se figur 25).

```
for (int i = 0; i < dataFloat.Length; i++)
{
    dataFloat[i] = Remap(dataIntUnmapped[i], 0, maxValue, 0f, 1f);
}
float[,] floatMap = new float[480, 640];
return Make2DArray(dataFloat, 480, 640);
```

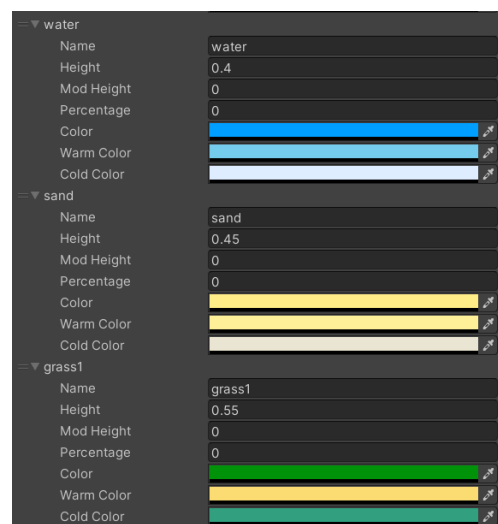
Figur 26: Screenshot af konvertering af data værdier

Herefter laver funktionen dataet om, så tallene er fra 0 til 1, i stedet for hvad det var før. Det endimensionelle array konverteres til et todimensionelt array, med højde og bredde på Kinectens infrarøde resolution (Se figur 26).

MapGenerator indeholder en class kaldt "TerrainType" (Se figur 27). Denne klasse består af de forskellige egenskaber regionerne i terrænet har. Heriblandt er der f.eks. hvilken højde terrænregionen burde være synlig ved, hvor stor procentdel af hele terrænet regionen udgør og hvilke farver terrænregionen burde have (Se figur 28).

```
[System.Serializable]
public struct TerrainType {
    public string name;
    public float height;
    public float modHeight;
    public float percentage;
    public Color color;
    public Color warmColor;
    public Color coldColor;
}
```

Figur 27: Screenshot af terraintype klassen



Figur 28: Screenshot af terrænregionerne

Den primære funktion i Unity projektet, er en funktion ved navn "generate_map" (Se figur 29). Denne funktion bliver kaldet hvert halve sekund, og står for at bruge dataet fra Kinecten, til at tegne en texture.

```
public void generate_map()
{
    bool locked = IsFileLocked(new FileInfo(@"C:\Users\maste\OneDrive\Skrivebord\Data.txt"));
    if (locked)
    {
        return;
    }

    float[,] noiseMap = ParseData();
    Color[] colorMap = new Color[width*height];

    for (int i = 0; i < regions.Length; i++)
    {
        regions[i].modHeight = regions[i].height;
        regions[i].percentage = 0;
    }
}
```

Figur 29: Screenshot af generate_map funktionen

Funktionen starter med at tjekke om tekstfilen er i brug af Kinect/Unity bridge. Hvis ikke, bliver ParseData funktionen, og resultatet gemmes i en variabel kaldt "noiseMap". Et array af farver kaldt "colorMap" bliver også oprettet med længden af antal pixels i den endelige texture. Et array af TerrainType classes kaldt "regions", gøres klar til at bruges senere i funktionen.

```
for (int y = 0; y < height; y++)
{
    for (int x = 0; x < width; x++)
    {
        float currentHeight = noiseMap[x, y];
        for (int i = 0; i < regions.Length; i++)
        {
            if (Remap(Remap(Mathf.Clamp(currentHeight + heightOffset, lowHeight, height), lowHeight, height, 0, 1), 0, 1, 1, 0) <= regions[i].modHeight){
                regions[i].percentage++;

                if (temperature < 0)
                {
                    colorMap[y * width + x] = Color.Lerp(regions[i].color, regions[i].coldColor, -temperature);
                } else if (temperature > 0)
                {
                    colorMap[y * width + x] = Color.Lerp(regions[i].color, regions[i].warmColor, temperature);
                } else
                {
                    colorMap[y*width + x] = regions[i].color;
                }
            }
            break;
        }
    }
}
```

Figur 30: Screenshot af terrængenerationen

Ved hjælp af to for-loops gennemgås alle pixelpositioner. Baseret på dybde dataet, der er associeret med pixelpositionen, bliver pixelpositionen tilføjet en farveværdi, baseret på terrænregionen, højden af vandstanden og temperaturen. Disse værdier bliver gemt i colorMap arrayet (Se figur 30).


```
for (int i = 0; i < regions.Length; i++)  
{  
    regions[i].percentage = regions[i].percentage / 306081 * 100;  
}  
  
if(regions[7].percentage >= 60)  
{  
    return;  
}  
  
MapDisplay display = FindObjectOfType<MapDisplay>();  
  
Blur blur = new Blur();  
  
display.drawTexture(blur.FastBlur(TextureGenerator.GenerateTexture(colorMap, width, height), 3, 1));
```

Figur 31: Screenshot af udregning af terrænregion procentdel udgørelse af mappet

Derefter beregnes procentdelene de forskellige regioner udgør i terrænet, og disse værdier bliver gemt i TerrainType klasserne i regions arrayet. Disse procentdele bliver brugt til musikdelen af Unity projektet. Hvis den øverste terrænregion udgør mere end 60% af hele billedet, stopper funktionen her. Dette er for at undgå et problem, hvor sne-regionen nogle gange ville overtage hele billedet uden grund, hvilket kunne være distraherende (Se figur 31).

En texture bliver herefter genereret med de korrekte farvевærdier fra colorMap, der bliver tilføjet en sløringseffekt, for at gøre billedet mindre skarpt, og texturen bliver vist til kameraret.

Kvalitetssikring

Der blev igennem projektet lavet løbende brugertests for at sikre, at forskellige komponenter var målrettet brugergruppen. Dette inkluderede henholdsvis moodboards og en test af sandhøjde. Begge tests var af typen "quick and dirty", hvilket er en type brugertests, hvor formålet er at skabe et indtryk om et produkt eller en idé på en hurtig måde.

Brugernes vurdering af visuelle indtryk

Her blev en brugertest brugt til at vurdere, hvilket moodboard best relaterede til brugergruppen. For hvert moodboard blev der spurgt, hvilke associationer testerne fik. Efterfølgende blev der spurgt, hvilket moodboard passede best til ordet "kreere". For det første moodboard (Se bilag 3) blev der hyppigst svaret "Himlen, død, roligt", for nummer to (Se bilag 4) blev der hyppigst svaret "mørkt" eller det blev beskrevet som en modsætning af det første. For det tredje moodboard (Se bilag 5) blev der hyppigst svaret "fantasy, magi". Der blev hyppigst svaret, at nummer tre moodboard passede best til at kreere.

Derfor blev moodboardet "Rustik natur" valgt som det visuelle indtryk, gruppen ønskede. Dog blev dette ikke implementeret grundet andre, højere prioriter. Dog for at fremhæve farverne i selve boksen, blev det valgt at male bordet sort.

Brugernes vurdering af sandhøjde

Grundet uenighed om den optimale sandhøjde til projektet, blev der spurgt brugere om deres mening. På daværende tidspunkt var sandhøjden vurderet til at skulle være 15-20 cm. Dog vurderede brugerne, at 10 cm var nok sand, hvilket fik os til at ændre vores planlagt design. Testen blev udført ved at bede testere om at prøve at bygge noget i en spand med omkring 8 cm jord. De

blev begge enige om, at det var en god højde, og at 15 cm faktisk ville have forringet oplevelsen, da bunden ikke ville kunne nås (Se figur 32).



Figur 32: Billede af dybdetest måling

Brugertest

Grundet det færdige produkt blev fremvist under et åbent hus-arrangement var omstændigheder ikke til at holde formelle brugertests. Gruppen observerede og adspurgte selvfølgelig stadig spillers spillere. Det fremgik meget hurtigt at knapper til at justere både temperatur og vandstand ikke blev interageret med af spillerne, medmindre de blev oplyst om både deres funktion og at det var tilladt. Eftersom knapperne af design ikke havde symboler eller lignende til at vise deres formål, blev de ofte forvekslet med knapper til at styre teknikken bag spillet. Nogle af de adspurgte spillere kunne godt have svært ved at høre musikken ændrede sig, men så snart de blev oplyst omkring det, kunne de i de fleste tilfælde identificere hvordan musikken ændrede sig. Udover forvirringen med knapperne modtog spillet en overvældende mængde ros og positive anmeldelser. Her er et par citater fra nogle af spillerne:

“Meget god ide.”

“Mega fedt koncept.”

“Mine børn kunne være i flere time i den der.”

Under fremvisningen var der nogle spillere som undlod at lave traditionelle landskaber i sandkassen, men i stedet valgte at skulptere kendte figurer fra computerspil eller visse andre, mindre børnevenlige, former. Det var tydeligt at noget af spillets charme kom i form af nostalgi i at lege i sandkasser, samt den medfødte frihed som spillet har. Selve sandet var i sig selv en oplevelse som spillerne var interesseret i. Netop det havde gruppen også tidligere observeret da sandkassen første gang fik tilført sandet. Efter sandet var kommet i gik der ikke lang tid før bordet var omringet af folk ivrige efter at lege med sandet.

Hvis både tiden og omstændighederne havde været til det, ville det have været oplagt at holde en observationstest, efterfulgt af et interview. Eftersom projekt sandkasse er et arkadespil med fokus på spillerens oplevelse vil en observationstest være bedst til at se hvordan spillerne vælger at spille, går spilleren efter at lave et sammenhængende landskab? Ændrer spilleren konstant både landskab, temperatur og vandstand? Eller noget helt tredje. Det efterfølgende interview vil være med til at skabe en forståelse for hvordan spilleren oplevede spillet og hvad de følte.

Mulige forbedringer

Vores produkt er ikke perfekt, og der er mange ting som kunne være gjort bedre. Og hvis der havde været mere tid, kunne det være at det var blevet indført. For det første, blev vi ikke færdige med at indføre ting fra vores MoSCoW model (Se bilag 2). Vi blev færdige med at indføre alle must have elementer, men blev kun delvist færdige med should have. Dette er i sig selv ikke et problem, men det er noget vi gerne ville have forbedret, hvis der var tid. En af de problemer der er i vores produkt, er problemer som kunne være undgået hvis vi havde mere tid, eller i hvert fald, hvis vi havde håndteret vores tid bedre. Et af de store problemer ved produktet, var at det tog lang tid at kalibrere og det var meget besværligt. En forbedring, som vi prøvede at finde en løsning til, men blev nødt til at droppe, var at lave et system, så projektoren nemmere kunne justeres, efter den var sat op. Derudover ville vi også gerne have set at den så lidt bedre ud. Far starten havde vi tænkt designet med i vores proces, men vi endte ikke med at få indført nogle af de design overvejelser vi havde. Et bedre design kunne have været med til at forbedre brugeroplevelsen, samt løse andre problemer. F.eks. var det et problem, at folk ikke lagde mærke til, eller vidste om de måtte bruge de dials der var på siden af bordet. En forbedring kunne være at gøre dem mere synlige f.eks. med lysdioder, eller gøre dem en del af designet. Det hænger også sammen med at folk ikke helt vidste hvad der foregik, når de først stod ved sandkassen. Der var ikke mange der lagde mærke til de helt fine detaljer ved lyden, eller dialsne. Derfor kunne det være en ide at give en form for introduktion til oplevelsen, for at gøre brugeren klar på at de skal udforske og være nysgerrige. Det er stadig meningen at brugeren selv skal opleve og udforske, men brugeren vidste ikke at det var det de skulle, så en introduktion kunne forbedre oplevelsen. Derudover kunne der også være nogle tidligere ændringer i lyden, da det til tider var svært at høre forskellen. Og yderligere kunne det være en stor forbedring med nogle flere og længere soundtracks, så der er en mere sanselig oplevelse, som varer i længere tid. Derudover var det største brud på illusionen, at programmet var langsom til at opdatere og det glitchede tit. Dette var umuligt at forbedre eller fikse på den tid vi havde, eftersom det var en enorm stor udfordring bare at få det til at virke. Men hvis der var mere tid, kunne det være med til at forbedre brugeroplevelsen en hel del. Og så er der også mindre fejl, som gjorde det svært f.eks. at montere kassen med dials. Kassen var lavet af et materiale, som var tyndt, og gik nemt i stykker, så det var svært at sætte fast til bordet. Derfor kunne den kasse f.eks. laves af krydsfiner eller et andet, mere monterbart materiale. Derudover var det også besværligt at få så mange programmer til at arbejde sammen på en computer, så at kunne sammensætte eller minimere mængde af programmer nødvendige, kunne være en stor hjælp for blandt andet performance. Det kunne f.eks. være ved at bruge en arduino leonardo til at styre dialsne, og dermed få et controllerinput i stedet for bare tal input.

Konklusion

Vi kan konkludere, at vi i dette projekt har produceret en interaktiv sandkasse kombineret med en dynamisk lydoplevelse, som responderer på højderne af sandet i sandkassen. Vi har lavet et program, som sammenbinder flere forskellige lydprogrammer, Unity programmer, og arduinoinput. Alt dette spiller sammen til at skabe en magisk følelse for brugeren ved brug af denne arkademaskine. Der er blevet gået igennem flere iterationer, forslag, ideer, og nederlag, for at skabe denne oplevelse. Alt er ikke endt med at være perfekt, og der er flere ting der, ved brug af mere tid, kunne have været forbedret. Men på trods af det, har vi formået at skabe et produkt der lever op til alle opgavens krav og vores personlige forhåbninger til projektet, samtidig med at generere positiv feedback fra brugere. Det betyder, at på trods af de mangler der er for produktet og arbejdsprocessen, blandt andet indenfor brugertesten af oplevelsen, så har projektet været en succes.

Bibliografi

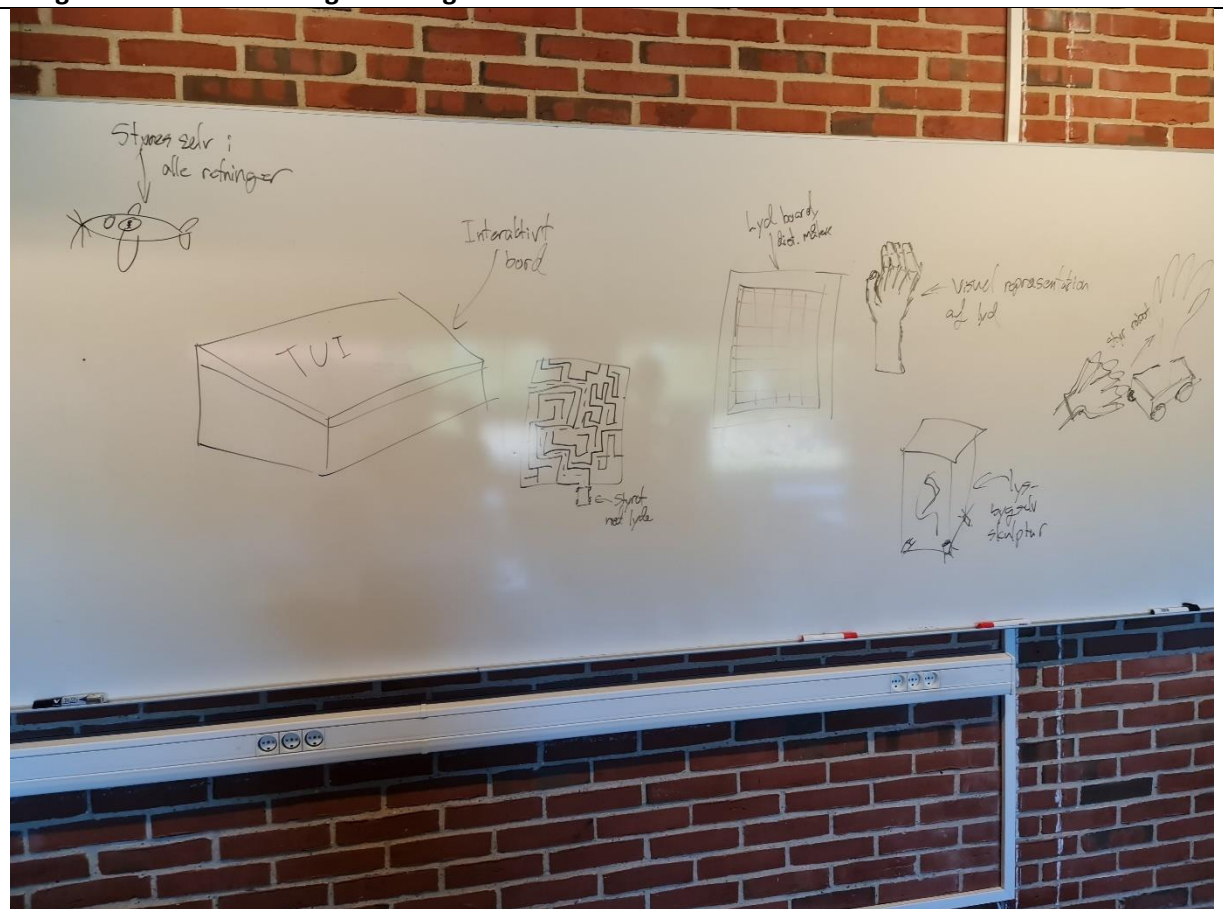
Larsen, L. J. (2022). Spilmekanikker. I L. J. Larsen, *Game Design - En Introduktion* (s. 27-32). Praxis.

Larsen, L. J. (2022). To perspektiver på computerspil. I L. J. Larsen, *Game Design - en Introduktion* (s. 10-14). Praxis.

Sharp, H., Preece, J., & Rogers, Y. (2019). I H. Sharp, J. Preece, & Y. Rogers, *Interaction Design: beyond human-computer interaction, Fifth Edition* (s. 2-9). John Wiley & Sons, Inc.

Bilag

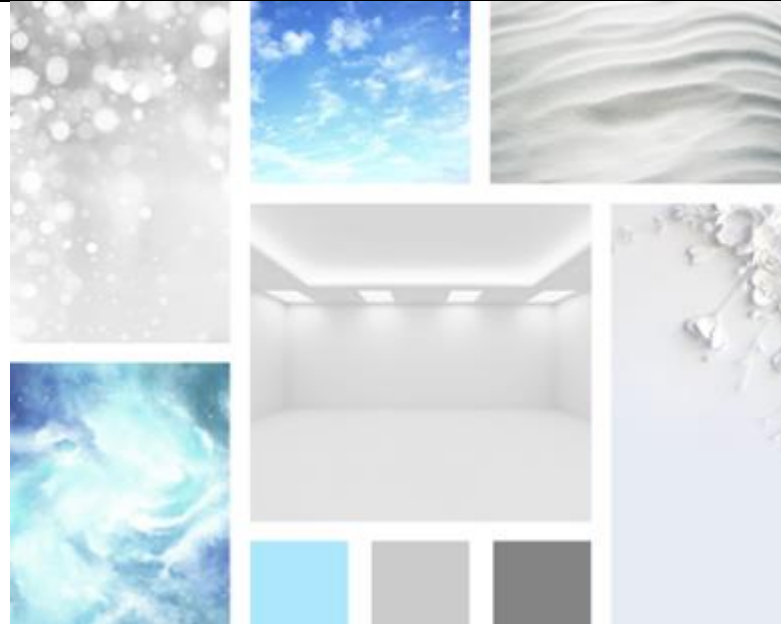
Bilag 1: Tavleskitser af idegenereringen



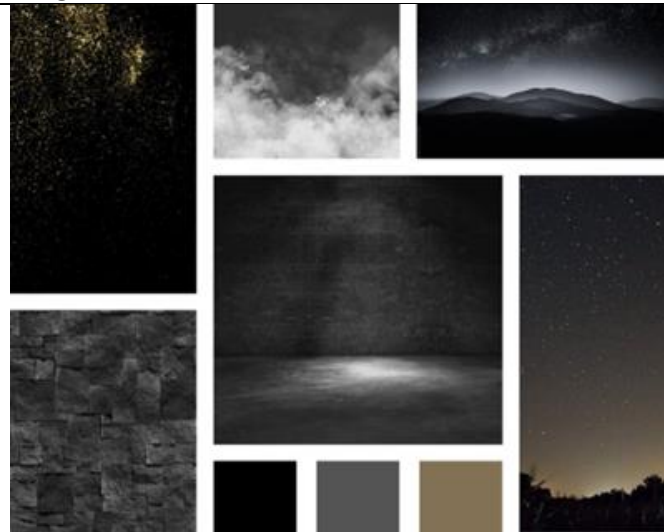
Bilag 2: Den fulde MoSCoW model

Must have	Should Have	Could Have
Et bord	Dynamisk lyd	Tegne muligheder (Til at kunne tegne tekstur på landskabet)
En projektor der kan projektere på bordet	Drejeknapper til justering af parametre (Så som, vand niveau, temperatur, årstid, dagstid, osv.)	Mulighed for eksterne elementer, som f.eks. firkanter, som bliver til bygninger osv.
En måde at måle dybden fra projektor til bordet	Specielt sand (F.eks. magnetisk sand)	
Sand til modellering af landskabet	Eventlyde (F.eks. en speciel lyd, når man først graver ned til vandet i sandet)	
Et program der kan behandle topografi dataen til projekterne	Selvstændig struktur (F.eks. at den kan stå alle steder, eller så den ikke skal sættes i stikkontakten)	
Lyddesign	Simulation af dyr eller landskab	

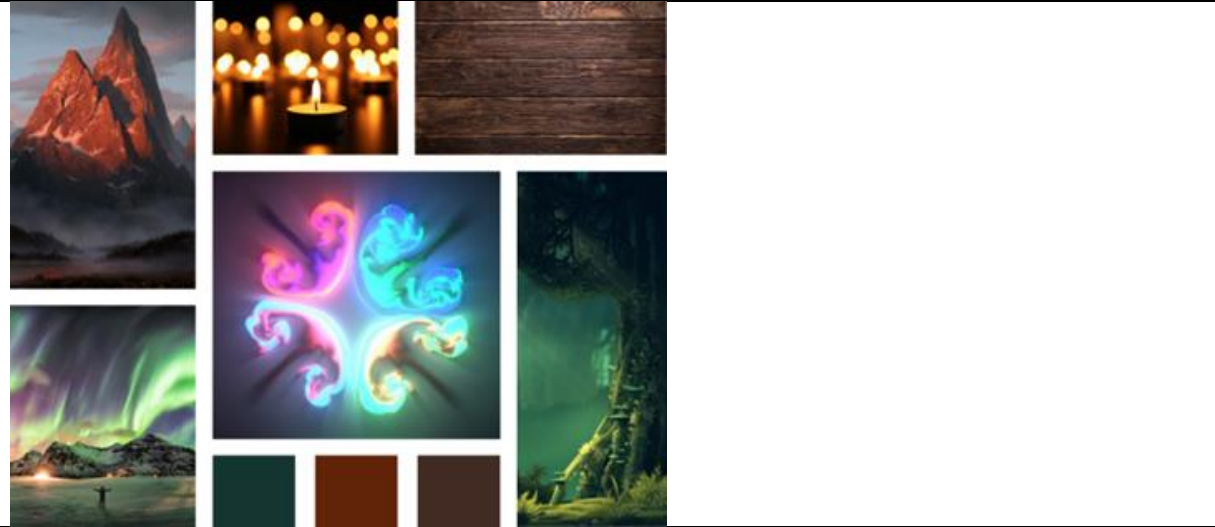
Bilag 3: "Hvid minimalisme" moodboard



Bilag 4: "Sort minimalisme" moodboard



Bilag 5: "Rustik natur" moodboard



Bilag 6: KILDEKODE

Kildekoden afleveres separat, grundet størrelsen, og det at kildekoden ikke er tilgængelig før mandag d. 28 (Undskyld)

Bilag 7: Arduino kode

```
float floatMap(float x, float in_min, float in_max, float out_min, float out_max) {  
    return (x - in_min) * (out_max - out_min) / (in_max - in_min) + out_min;  
}  
  
// the setup routine runs once when you press reset:  
void setup() {  
    // initialize serial communication at 9600 bits per second:  
    Serial.begin(9600);  
}  
  
// the loop routine runs over and over again forever:  
void loop() {  
    // read the input on analog pin A0:  
    int analogValue = analogRead(A0);  
    int analogValue2 = analogRead(A1);  
  
    // Rescale to potentiometer's voltage (from 0V to 5V):  
    float voltage = floatMap(analogValue, 0, 1023, 0, 5);  
    float voltage2 = floatMap(analogValue2, 0, 1023, 0, 5);  
  
    // print out the value you read:  
    Serial.print("Analog: ");  
    Serial.print(analogValue);  
    Serial.print(", Voltage: ");  
    Serial.println(voltage);  
    Serial.print("Analog2: ");  
    Serial.print(analogValue2);  
    Serial.print(", Voltage2: ");  
    Serial.println(voltage2);  
    delay(1000);  
}
```