



Eksamensprojekt

Digital Design & Udvikling

Spilform: Intelligente fjender

Titel: The Horror of the Maze

Navn: Jens B. H. Issa,
Zacharias A. Nielsen, Anders B. Senger
& Emil Christiansen.

Vejleder: Esben Øvland (ESOE)

Valghold: Teknikfag - DDU A

Afleveringsdato: 28/04/2020

Projektperiode: Uge 5 - 18

Forord

Dette eksamensprojekt er udarbejdet på basis af et oplæg, som der blev delt ud i Digital Design og Udvikling. Emnet som dette projekt indeholder, er AI eller nærmere kendt som Intelligente Fjender. Det vil sige, at der skal skulle kreeres et spil, der havde noget at gøre med en AI NPC. Rapporten er derfor udarbejdet på baggrund af dette tema, og i de næste sider, vil man få en beskrivelse af vores udvikling af produktet, og hvordan vi valgte en problemstilling.

Gruppemedlemmerne der har medvirket i dette projekt, er følgende:

Zacharias A. Nielsen (3.E - Programmering & Matematik) Grafiker og Skribent
Jens B.H. Issa (3.E - Programmering & Matematik) Programmering og Skribent
Anders B. Senger (3.E - Programmering & Matematik) Primær Programmør
Emil Christiansen (3.E - Programmering & Matematik) Grafiker og Designer

Undersøgelserne vi har foretaget, har primært været kvantitative med fokus på unge personer, der har interesse til spil og IT. Vi har under ugenes forløb, taget kontakt til en del personer, og fået deres respons på vores produkt og derfor vil vi gerne uddele en tak til de medvirkende personer.

Rapporten læses kronologisk, og bilag m.m. findes i bunden af dokumentet sammen med indholdsfortegnelsen. Words egen figurer anvendes gennem rapporten. Litteratur kan ses i fodnoter, og i litteraturafsnittet, kan alt litteratur læses i alfabetisk orden.

Indholdsfortegnelse

Forord	2
Indholdsfortegnelse	2
Projektbeskrivelse	5
Projektoplæg	5
Problemidentifikation og analyse.....	7
Problemformulering	8
Produkt-/processprincip	8
Projektafgrænsning	9
Ide	9
Fokusområde/specifik afgrænsning	10
Teststrategi	11
Projekttidsplan/SCRUM	12
Projektrealisering	12
Indledning	12

Problemanalyse	13
Produktprincip	16
Krav til vores produkt	16
Kravuddybning	17
Idegenerering	18
Produktudformning	21
Præsentation af spillet	21
Spilleren	21
Spillerens bevægelse & lommelygte	21
Pick-up, notecounter og UI-tekst (signifiers og affordances)	24
Den progressive AI	26
General intelligens (pathfinding, waypoints, åbne døre)	26
Progressiviteten og interaktionen med spilleren	28
Design	31
Bane	31
Grafik	31
Oprettelse af AI	31
Noterne	34
Textures	34
Lyd	35
Implementering af bane	35
Implementering af banens udformning	35
Implementering af dørene	36
Implementering af lysene	36
Exit-door	37
Menu	37
Menu visualiseret	37
How-to	38
Tutorial	39
Spillet cyklus og tilstande for AI og spiller (Flowdiagrammer og tilstandsdiagrammer)	40
Realisering og optimering	41
User-cases (AI-sprint)	41
Test	42

THE HORROR OF THE MAZE

White Box-test	43
Black Box-test.....	43
Sammenligning af Black Box-test & White box-test.....	44
SUS (System Usability Scale)	44
Vurdering af testene	45
Konklusion:	46
Evaluerig	48
Litteraturliste:	49
Med forfatter.....	49
Uden forfatter	50
Bilag:	51
Bilag 1, spørgeskema:.....	51
Bilag 2, White Box spørgsmål & svar	55
Bilag 3, Black Box spørgsmål & svar.....	56
Bilag 4, SUS spørgsmål & svar.....	57
Bilag 5, SUS tabel 2 og 3.	60
Bilag 6, BurndownChart.....	61
Bilag 7, AI Struktur, knogler og andre features.	62
Bilag 8, noterne	63
Bilag 9, Releasebacklog + produktbacklog.....	63
3D-modellering sprint + textures sprint - 55timer	63
Pick-up objekter programmering: 10 timer	65
Bane - 32 timer	65
Lyd: 6 timer.....	65
Spiller: 6 timer.....	66
Døre-programmering: 4 timer	66
AI-modelering: 3 timer	66
Animation: 5 timer.....	67
Lys: 5 timer	67
Bilag 10, assets/3d-modellering	68

Projektbeskrivelse Projektoplæg

Eksamenscases i DDU

Generelt for alle temaer

Anbefalet gruppestørrelse er 3-4 personer. Hvis der er tungtvejende grunde til det, kan man få lov til at lave projektet individuelt. Der skal altid indhentes tilladelse hertil.

Der skal vælges ét af de følgende temaer. Der udarbejdes en projektbeskrivelse, som er en kort beskrivelse af, hvad I vil lave. Her skal der være:

- En problemformulering
- Særligt fokusområde for projektet
- Afgrænsninger
- Tidsplan/overordnet burndownchart

Denne afleveres og skal godkendes af Esben senest d. 21/2 2020

Det digitale produkt skal afleveres i et format, der kan køres af andre på en standard PC. Dette afleveres på Onedrive.

Rapporten skal dokumentere jeres proces, og ligger til grund for bedømmelsen af kvaliteten i jeres arbejde. Rapporten må fylde 15-30 sider for det første medlem af gruppen, og 5-15 sider per øvrige medlemmer. For eksempel vil en gruppe med 3 medlemmer skulle lave en rapport på 25-60 sider. Rapporten afleveres i 2 eksemplarer. I skal selv sørge for indbinding og medier. Med sider menes der fysiske sider.

Produkt og rapport skal afleveres den 28. april kl. 11 ved dueslaget (lærerværelset)

Tema 1: Læringsspil

I skal lave et spil der understøtter læring i fag i folkeskolen eller gymnasiet. I skal undersøge relevansen og behovet.

Eksempler

- Et VR spil hvor du laver kemiforsøg. Målet kan være at forudse hændelse eller bestemme kemikalier ud fra farver/reaktioner
- Puzzle game til redox reduktioner
- Simulering af biologiforsøg fx fotosyntese

Tema 2: Spil for fokusområder

Der er mange emner i dag, hvor forskellige organisationer anvender spil til at skabe fokus på særlige problemstillinger. Som eksempel kan nævnes "Dumb ways to die".

I skal lave et spil som skaber fokus på et selvvalgt område. Emnet skal være et reelt problem/debatområde

Eksempler

- Affaldssortering i VR (med online highscore)
- Simulatorspil af en bys forurening (støj, bilos, skrald)
- Bilspil, hvor det gælder om at køre optimalt i forhold til forurening, men stadig som racerløb
- Administrer et hospital. Hvad kræver det?

Tema 3: Intelligent fjender

Det er altid en udfordring at lave spil med AI. En af de største udfordringer er at lave progressiv sværhedsgrad. Hvordan laves NPC (non player characters) kunstigt intelligente, så de bliver gradvist sværere og sværere, og dermed udfordrer spilleren? Hvordan opretholdes balancen, så det ikke bliver for svært? Hvad kunne udfordringen være ved at lave udviklende friendly NPCs?

I skal lave et spil der har en udviklende sværhedsgrad blandt NPC.

Bedømmelseskriterier (pkt. 4.3 i læreplanen for faget)

Bedømmelsen er en vurdering af, i hvilket omfang eksaminandens præstation lever op til de faglige mål, som de er angivet i pkt. 2.1. Der lægges særlig vægt på:

Generelt

- evne til at arbejde problemorienteret
- evne til at kombinere teori og praktisk arbejde i et projekt
- perspektivering til relevante emner inden for teknikfaget

Rapportens form og indhold

- bearbejdning af projektets problemstillinger
- planlægning og vurdering af projektførelsen
- dokumentations- og kommunikationsværdi, herunder overskuelighed, sammenhæng, kildehenvisninger og teknisk dokumentation
- fordybelsesgraden
- specificerede krav til produktet
- en fagligt begrundet argumentation for de foretagne valg

Produktet/procesforløbet

- omhu og professionalisme ved fremstilling
- kvalitet i forhold til de opstillede krav
- argumentation for til- og fravalg

Mundtlig eksamination

- den mundtlige præsentation af projektet
- redegørelse for de valgte løsninger
- demonstration af ejerskab i forhold til projektets indhold
- besvarelse af uddybende og supplerende spørgsmål.

Problemidentifikation og analyse

¹AI, også kaldet "Artificial intelligence" eller på dansk kunstig intelligens er en teknologi hvorved man får såkaldte NPC (non-player characters) til at efterligne menneskelige tendenser. Denne form for teknologi har været brugt i videospil siden 1940² hvor det første spil som anvendte AI-teknologien var kaldet Nim. Nim er et strategispil, som omhandler om at fjerne objekter fra bunker. Ideen er at hver spiller skal fjerne mindst et objekt fra bunken og må fjerne så mange objekter som muligt, så længe de er i samme bunke. En maskine ved navn "Nimatron" blev præsenteret ved konferencen "1940 New York World's Fair Westinghouse". Nimatron var en AI-maskine, som kunne efterligne menneskelige tendenser og var særledes god til dette. Kun få personer kunne slå Nimatronmaskinen i spillet "Nim" gennem dens seks ugers prøvetid.

Gennem de efterfølgende årtier har AI-teknologien forbedret sig væsentligt fra at kunne spille et simpelt strategispil som Nim der havde mulighed for at slå mennesker, til at kunne skabe AI's der har mulighed for at lære selv gennem spillets forløb ved reaktion til spillerens aktioner. Men findes der konkrete eksempler på dette? Ja, et eksempel på dette er Google's version af AI, som hedder AlphaZero. Denne AI lærte sig selv at spille skak på under 4 timer ved at simulere flere millioner af spil³. Efter at have lært sig selv at spille skak, havde AlphaZero mulighed for at slå den tidligere verdensmester i skak. AlphaZero kunne gennem 100 spil enten slå den tidligere verdensmester, eller stå lige med vedkommende.

Problemstillingen med en AI, som AlphaZero er at det ikke er særlig interaktivt for spilleren at være imod en AI som spilleren ikke kan slå, da rent logisk vil en computer altid have mulighed for at kunne lagre mere information end et menneske nogensinde ville kunne, og derfor vil AI'en altid kunne slå spilleren. Derfor hedder det "Artificial Intelligence" fordi det ikke er et naturligt objekt, men nærmere noget der er fremstillet gennem maskinkode. Som nævnt i artiklen "Artificial intelligence in Video games"⁴ er en af de vanskeligste opgaver i forhold til at kunne implementere en NPC som bliver progressivt sværere, er at den ikke bliver så svær at den ligefrem ødelægger spiloplevelsen for spilleren, og at udviklerne bag spillet simpelthen mister kontrollen over selve spillet og dens sværhedsgrad. Derfor er et af de vigtigste elementer i en spilopbygning hvori en progressiv sværhedsgrad er tilstede er at finde en balance mellem ikke at være for simpel at slå, men heller ikke være en AI som AlphaZero hvori selv den absolutte bedste skakspiller ikke har mulighed for at slå AI'en.

¹ Encyclopedia of philosophy: Artificial Intelligence - af Hauser, Larry

² The Newyorker: IT - af Grant, Eugene F. & Lardner, Rex (Artikel)

³ The Guardian: AlphaZero AI beats champion chess after teaching itself in four hours - af Gibbs, Samuel (Artikel)

⁴ TowardsDataScience: AI in video games - af Maass, Laura E shummon (Artikel)

Derfor er ideen med en NPC som er opbygget omkring en AI at finde balancen mellem at være så svær at det ligefrem ødelægger spiloplevelsen, men heller ikke være så let at det bliver kedeligt. Hvad en NPC som er i et spil skal fuldføre er, at den skal kunne skabe "Meaningful play". "Meaningful play" er et begreb som bliver anvendt i Salen og Zimmerman's spilteori. Denne spilteori opbygger på, hvordan et spil rent faktisk kan klarificeres som et reelt spil.

"Meaningful play" er målet bag et spil ifølge Zimmerman og Salen's spilteori⁵. Ifølge teorien opnås dette, når spilleren har mulighed for at interagere med systemet, som er spillet og systemet har mulighed for at interagere med spilleren. Hvis man indsætter dette i forhold til en NPC, skal NPC'en have mulighed for at ændre hvorledes spilleren interagere med spillet, og spilleren skal have mulighed for at interagere med NPC'en. Men hvad kan AI betyde for vores proces og projekt? Først og fremmest, er noget AI de fleste kender til, da næsten alle moderne spil har indbygget forskellige intelligente fjender i sig. Dette er nok fordi, at de understøtter spillet i store omgange, og at det er nemmere for udvikleren at få andre intelligente objekter, til at bearbejde spillet. Den anden grund, er at en AI kan justeres. Hvis denne AI, var en fjende, ville man kunne få den til at udføre de relevante ting, som man ønskede. Dette ville man ikke kunne gøre hvis fjenden var en levende player. Dette er en af de fordele der er ved AI, og simpelthen kunne oprette sin egen spillestil, inden for sin egne rammer.

Problemformulering

AI er en teknologi, som bliver anvendt i mange forskellige industrier, og dette gør sig også gældende indenfor spilindustrien. I nutidens videospil er der i dag et specielt fokus på hvorledes man har muligheder for at lave en progressiv fjendtlig AI. I dette projekt, vil der være speciel fokus på at lave et 3D spil hvori fjenden som man støder ind i 3D spillet udvikler sig gennem spillets forløb i en progressiv udviklingsmodel. Hvilke forskellige problemstillinger vil man kunne have i et spil, som bygger på at spillet har en progressiv AI?

- Hvordan kan man få fjenden (AI'en) til at have en stigende sværhedsgrad gennem spilforløbet?
- Hvilket tema kan vi lave som understøtter en fjende der stiger i sværhedsgrad således at det hænger sammen med den grafiske betydning?
- Hvordan kan man give spilleren en følelse af at vedkommendes spillestil har betydning for spillets forløb?
- Hvordan kan man konstruere fænomenet "Meaningful play" og anvende det i spillet på en relevant måde?

Produkt-/processprincip

I teknikfaget Digital Design og Udvikling, ligger man vægt på hvordan et produkt kan fremstilles, inden for de tekniske rammer som DDU har. Altså med relevante krav.

⁵ Book: "Rules of play" - af Salen, Katie & Zimmerman, Eric (PDF)

Det er vigtigt og nødvendigt, at hvis man under produktfasen, holder disse såkaldte møder og opstiller krav for hver møde. Det er her den agile metode ⁶SCRUM bliver anvendt. Scrum metoden er som nævnt en del af ⁷Agile Processsing, som er måden hvorpå man typisk behandler et software-udviklingsforløb. Meningen med det er at det skal selvfølgelig indeholde flere gruppemedlemmer. Disse medlemmer, skal så holde deres SCRUM-møder, og forklare hvordan det ser ud på deres individuelle roller og om de har nået deres mål. Dette bliver gjort gennem sprints, user cases og andre relevante agile faktorer. Udover SCRUM-delen, er det vigtigt at et team, som arbejder på et produkt, holder sig til tidsplanen. Her kan man så anvende forskellige tidsplanlægning programmer, og følge op på om man har overskud eller er tidspresset. Tidsplanlægning er essentielt, ellers ved man faktisk ikke hvor langt man er, og om man kan nå i mål. I dette projekt bliver ⁸"Burndown chartet" brugt. Ydermere, er det vigtigt at når man så skal starte et projekt op, at man vælger en såkaldt "SCRUM MASTER" Det er ham, der sørger for at gruppen bliver fulgt op med de relevante ideer, problemer eller sørger for at de krav som der er bliver overholdt. Derudover, er det også vigtigt at fordele de roller som er vigtige ud på de individuelle gruppemedlemmer. Det vil sige at det vil være smart at fordele programmeringsarbejdet ud på ham/hende der har mest erfaring med faget. Dette medfører overskud, og til en vis grad også bedre produktkvalitet. For eksempel, kan det være nødvendigt at opstille krav til programmeringsdelen. Da programmering er et fag, der kan støde ind i hvilket som helst problem, kan det være svært for programmøren at følge med de andre i gruppen. Derfor vil det være en fordel at opstille nogle krav for programmøren, så han ved hvad han skal nå de forskellige dage. Dette resulterer også i at det forhåbentligt bliver et godt stykke kode, som er let tilgængelig for alle i gruppen. Det vil sige at hvis der skal oprettes en AI som bliver lavet visuelt på en tirsdag, skal programmøren fra søndag af finde ud af hvordan man kan få den AI til at åbne døre, bevæge sig imod et objekt eller andre relevante ting inden tirsdag. Dette gør at der nu er krav for programmøren, og han ikke på selve tirsdagen skal finde ud af hvad han skal lave. Alt dette medfører bare bedre overskud og mindre problematik for gruppen. Alt dette giver et hint af at teknikfaget DDU skal have krav, for at hver enkel rolle i temaet kan følge med.

Projektafgrænsning

Ide

Hvis vi i dette projekt havde en uendelig mængde af tid, og havde uendelig mængde af ressourcer, vil følgende være det ideelle spil som vi ville arbejde på.

⁶ Agilealliance: What Is Scrum - af Skribent på hjemmesiden (Artikel)

⁷ Agilealliance: Agile Glossary - af Skribent på hjemmesiden (Artikel)

⁸ Projectmanager: Burndown Chart: What is it & How do I use it? - Af Blackburn, Megan (Artikel)

Vores spil er som beskrevet i indledningen et spil som baserer sig på at der findes en progressiv NPC, som bliver mere udfordrende gennem spilforløbet.

Spillet skal være et 3D-baseret spil, hvori spilleren eller spillerne befinder sig i et mentalhospital. Spillet skal omhandle om at vedkommende skal finde et dokument, som er blevet revet i stykker.

Spillet er et single-player baseret spil, hvori NPC'en som bliver indlagt i spillet systemet skal have en progressiv sværhedsgrad, hvor den bliver sværere og sværere at slå. En videreudvikling af spillet kunne være, hvis man f.eks. lavede det således at man kunne samarbejde med en anden spiller for at slå spillet.

Spillet skal være således, at spillere skal igennem flere antal baner, hvori man går dybere og dybere ned i mentalhospitalet. Hertil vil der blive spawned flere NPC'er, som hver har deres egen karakteristisk i form af at man allerførst starter med en NPC, som følger efter en og bliver sværere og sværere, men hvorhenne i senere levels at der dukker flere NPC'er op, som har deres egen form for angreb. Videreudviklingen af NPC'en kunne være således, at NPC'erne ud fra spilleren eller spillernes aktioner lærer gennem spilforløbet.

Hertil ville spilleren eller spillerne kunne interagere med de forskellige NPC'er ved hjælp af forskellige måder, ved angreb eller lignende.

Fokusområde/specifik afgrænsning

Da vi i dette projekt ikke har uendelig tid og uendelige mængder af ressourcer, vil vi i dette projekt fokusere på, hvordan vi kunne skabe EN enkel NPC, som skaber en form "Meaningful play" for spilleren. Et spil hvori spillere har mulighed for at integrere med systemet og systemet har mulighed for at interagere med spilleren, og at denne form for interaktion er indbygget i systemet (spillet).

Vores spil vil fokusere på at være et 3D-baseret spil single-player spil, hvori spilleren bliver indlagt i en labyrinth, hvor spilleren så skal finde en vej ud af labyrinten, mens NPC'en, som vil være et frygtindgydende væsen, følger efter spilleren indtil spilleren finder en vej ud af labyrinten og har indsamlet alle objectives.

Da vores tema baserer sig på hvorledes man vil kunne lave et spil, der har en progressiv AI, skal der være nogen kriterier for hvornår NPC'en bliver sværere, og på hvilke måder NPC'en bliver progressiv sværere.

I dette spil er der også kun fokus på at implementere en enkel NPC, som har en stigende sværhedsgrad. Da vores spil rent tematisk foregår i et mentalhospital, skal den progressive udvikling i spillet følge dette koncept. Spillet skal have en form for objekt, som påvirker måden hvorpå monsteret interagerer med spillet. I spillet med en uendelig mængde af ressourcer, var den oprindelige ide at lave et massivt spil, hvori man kunne have endnu mere meaningful play med at kunne interagere med NPC'en, men i dette spil har vi tænkt os at fokusere på færre baner, og fokusere på selve interaktionen af den progressive NPC i forhold til sværhedsgraden.

Spilleren skal stadig have mulighed for at interagere med NPC'en, stå stille, således den ikke kan høre spilleren.

Efter nærmere diskussion med gruppen, kom vi frem til følgende historie. Historien bag spillet skal være sådan, at spilleren har af job at finde et dokument, men spilleren

⁹ Techtargert: 3D Gaming - af Rouse, Margaret (Artikel)

opdager at når vedkommende ankommer til mentalhospitalet, er det dokument spilleren skal finde revet i mange stykker som er spredt udover hospitalet. Målet med spillet er så, at spilleren skal indsamle dokumentet og finde en vej ud af hospitalet. Den progressive sværhedsgrad hos NPC'en bliver italesat, når spilleren begynder at opsamle disse stumper af det revne dokument. For hver eneste stumpe af det revne dokument bliver opsamlet, bliver AI'en progressivt sværere. Måden hvorpå AI'en bliver sværere er, at den har mulighed for at kunne høre lyd længere væk, og derved vil have lettere ved at finde spilleren. Spilleren skal så være mere stille, for at sørge for at monsteret ikke ved hvorhenne spilleren er. Dette forklares også i vores tutorial.

Teststrategi

Teststrategi i henhold til dette projekt, er baseret på ideen at spillet allerførst vil blive testet af udviklerne bag software og et par testpersoner. Udviklerne og testpersonerne vil herefter blive bedt om at svare på de samme spørgsmål. Disse spørgsmål vil være sat i forhold til de krav som spillet er blevet påsat og ift. Afgrænsningen. Derefter udføres der en ¹⁰SUS-test på de testere, som ikke har noget kendskab til den bagliggende software i projektet.

I den første del af vores teststrategi, vil vi anvende ¹¹White-box & black-box testmetoden. Whitebox test er en testmetode, hvori testerne har kendskab til spillets bagliggende software. Denne metode har sine fordele i, at testeren har mulighed for at kunne teste spillet i en dybere forstand, og ville kunne lokalisere fejlene i software, hvis fejl findes i spillet. Black-Box metoden er en metode, hvor en person som ingen kendskab har til spillet afprøver spillet og giver feedback baseret på dette. Denne metode har relevans i dette projekt, i og med at det er vigtigt at teste

Anden del af teststrategien ville være med anvendelse af SUS (System usability Scale). Denne testning tester et systems brugbarhed ved hjælp af et pointbaseret skema. Heri vil der anvendes mennesker udenfor projektet, som ingen kendskab har til dette.

Det samlede mål for denne teststrategi, ville være at ved allerførst kunne lokalisere fejlene ved at udviklerne tester spillet, og herefter vurdere ud fra SUS spørgeskemaerne, om disse lokaliserede fejl har nogen betydning for hvor attraktivt og værdifuldt systemet er. Tidshorisonten for testning vil være 5 timer, fordelt ud på to dage.

¹⁰ Usability: System Usability Scale (SUS) - af Skribent på hjemmesiden (Artikel)

¹¹ GeeksforGeeks: Differences between Black Box testing and White Box testing - af Jain, Mahak (Artikel)

Projekttidsplan/SCRUM

Projekttidsplanen for dette projekt, kan findes nede i bilag 6. Der er det komplette diagram over tidsforbruget, i forhold til de roller som vi uddelte. For at forklare den, kan man se til venstre side (Product Backlog) og derunder de titler, som der skal udfyldes af gruppen. Det vil sige alt fra 3D modellering til UI-programmering. Derefter, har de fået deres beginning balance, altså antal timer som arbejdet kræver at udføre. Det vil sige at ved f.eks. Texture, har den fået 10 timer. Det vil sige at designeren har 10 timer til at få lavet textures til spillet. Dette kan så fordeles ud over alle de 13 uger, som vi har i eksamensprojektforløbet. Så kan designeren, selv plotte ind hvilke uger han har lavet hvad, og hvis han havde andre ting han skulle lave, kunne han bare plotte timer ind for de andre roller. Imens alle så er i gang med at lave deres arbejde, kan man se på grafen om man falder bagud i forhold til tiden, og antal timer i det individuelle arbejde. For eksempel, kan man se at vi som gruppe faldt bagud i starten, men i slutningen kom på gode ben igen, og blev hurtigere færdig end planlagt. Se kurven i bilag 6, for nærmere forståelse. Alt i alt, har tidsplanen/burndownchartet gjort os opmærksomme på hvad vi mangler, og hvad vi skal arbejde mere på. Dette giver gruppen mere overblik, over arbejdet og hvad der kan forbedres på de individuelle roller.

Projektrealisering

Indledning

AI er et fænomen, hvori tankerne tit vil gå hen til dræberrobotter og menneskets udslettelse. Men i game design er dette fænomen dog en smule anderledes, og AI er en af de vigtigere komponenter til at kunne have et komplet spil.

I dette projekt vil vi undersøge hvorledes det vil være muligt at fremstille et spil i game enginen Unity. Heri har vi anvendt forskellige analyserende teknikker, således vi har kunnet zoome ind på vores problemstilling med AI og kunne kortlægge hvad præcist de mest vigtige elementer er for opbyggelse af et spil hvori der er en intelligent fjende.

Herefter vil der blive opstillet krav ud fra vores afgrænsning og problemanalyse, og disse vil blive sat op mod vores løsningsforslag, hvor faldet blev på en mental asylum/horror intelligent fjende spil.

Spillet blev herefter realiseret, og dette gennem gæes trin for trin. Der påvises derefter et eksempel på et sprint med flere user-cases. Vores produkt testes herefter ved anvendelse af black-box, white-box og SUS metoderne, og herefter bliver der vurderet på testene ift. De væsentlige krav og konkluderes til sidst.

Problemanalyse

For at kunne opstille krav til spillet, kan der analyseres relevant faglig viden og data, som kigger på progressive AI's og hvordan et spils tema kan bære langt i forhold til succes.

Skaberen af Amnesia, Thomas Grip, forklarede i vice.com artiklen "The Do's and Don'ts of Horror Game-making"¹² at de mest vigtige komponenter i et horrorgame, var ikke at overkomplicere de funktionalister som skal opbygges i spillet. De nyere, mere populære horrorspil såsom Slender og Amnesia er alle spil som har et meget simpelt koncept. Det er spil hvori gameplayet ikke er særlig vanskeligt, og hvor fjenden ikke gør noget specielt vildt, men samtidig er det to af de mest skræmmende spil ude på markedet. Som Grisp fortæller i artikelen vist tidligere *"Slender came about and showed a larger audience that a game could be terrifying with a bare minimum of features."* Dette citat viser, samtidig med reelle data om de mest populære horrorspil, at det ikke nødvendigvis er det mest komplekse og mest feature-rige horrorspil, som er det mest skræmmende spil, men faktisk meget tit er et spil, som er simpelt i sin funktionalitet, men skræmmende i sin artistisk syn og tema. Spil som Slender har vist, at succes kan opnås ved et simpelt tema, med en intelligent fjende som bliver progressiv sværere vha. af variabler som spilleren har mulighed for at ændre ved hjælp af dennes spillestil.

Men selvom en intelligent fjende har vist at være et tema i succesfulde spil hvorhen main-fjenden er en progressiv AI, kan der dybere analyseres hvorledes dette har belæg i den reelle verden og dette gøres vha. af en kvantitativ undersøgelse, i form af et spørgeskema som er lagt ud i gaming facebookgrupper (vedlagt i bilag 2).

Spørgsmålene i den kvantitative analyse blev stillet ud fra et pointskema, hvor hver respondent kunne give en score fra 1-10 i forhold til vigtigheden i elementet.

I denne kvantitative analyse, blev der adspurgte hvor stor betydning enkelte elementer i et spil havde for den generelle spiloplevelse. Da de adspurgte blev spurgt om hvorvidt det var vigtigt for dem, at en fjende i et spil var progressivt stigende i dens sværhedsgrad i en skala fra 1-10, var størstedelen enig i (<80%) at det var yderst vigtigt at fjenden blev sværere gennem spillets progression. Dette viser, at progressivt AI spil har appel til spillere, som er i målgruppen til videospil. Men heri kan det også være interessant at diskutere, hvor markant sværhedsgraden på denne progressive egentlig skal være og hvor meget denne skal stige? Ifølge et videnskabeligt studie fra et universitet baseret i Maastricht, er der videnskabelig bevis for, at en spiller vil nyde spillet mere, hvis de fjender som spilleren møder i spillet, er baseret på spillerens skill

¹² Vice: "The Do's and Don'ts of Horror Game-making" - Af Porter, Matt (Artikel)

i spillet¹³, og hvor dette hverken overstiger spillerens skill i spillet, eller gør det for let for spilleren. Dette er en AI power scaling, som man skal balancere for at gøre spillet nydeligt, og dette er også en tendens som er set i vores kvantitative analyse, hvori at der var en generel tendens til at spørgeskemasbesvarelser viste at spillere gerne vil have, at en AI starter fra et standpunkt, hvori at de kan følge med, og derefter progressivt udvikler sig til en sværhedsgrad, som vil være mere udfordrende.

Men hertil er det også relevant at analysere, HVOR udfordrende skal en progressiv AI være, således den stadig bibeholder den motiverende faktor for en spiller at fuldføre spillet? I artiklen "Do gamers really like hard games?" udgivet af gameinformer.com, diskuteres der hvorledes gamers i realiteten kan lide svære spil, og om hvilken sværhedsgrad gamers egentlig kan lide i en bred forstand og HVORFOR gamers godt kan lide at blive udfordret i spil. I artiklen beskriver Dr. Scott Rigby hvordan det normale menneske tænker, når de skal være succesfulde i et spil " [...]/*We like it because success at a challenge satisfies this basic need to feel mastery and growth – a need that has energized us for thousands of years to keep exploring, innovating, and tackling new frontiers*"¹⁴ I dette citat beskriver doktoren, at grunden til at gamers godt kan lide at møde nye udfordringer, og bekæmpe intelligente som bliver gradvist stærkere ift. Til deres niveau, er fordi at dette har mennesker gjort gennem hele deres levetid. At bekæmpe nye udfordringer og blive gradvist bedre til den bestemte aktivitet de har valgt at blive bedre i. Dette ses også reflekteret i den kvantitative undersøgelse, som er blevet udstedt hvor flertallet af de adspurgte (<70%) har specificeret at samtidig med at de gerne vil have progressiv fjende på deres eget niveau, er der også et behov for at den progressive fjende skal stige i en grad, således det er uden for dens oprindelige niveau i et mærkbart spring. Dette er fordi, at som dr. Rigby beskriver, at mennesker har en tendens til at ville afprøve deres egen kapacitet i en aktivitet til det yderste. Dog er det også vigtigt ikke at overspringe denne kapacitet, således at motivationen til rent faktisk at udføre en aktivitet ikke er til stede længere, som Dr. Rigby beskriver i dette citat fra artikel også: **"At the same time, if we feel a challenge is not within our reach – if it is pushing us too hard – we will disengage"**

Men selvom det er vigtigt, at spillet følger en progression som følger spilleren udvikling uden at det bliver for hårdt, er det så ikke også vigtigt at spilleren rent faktisk ved hvad objektet vha. en begynder-tutorial? Ifølge vores kvantitative analyse, er dette yderst vigtigt, da de adspurgte i en overstigende grad har besvaret at de

¹³ Spronck: "Difficulty scaling of game AI" - af Spronck, Pieter, Sprinkhuizen-Kuyper, Ida and Postma, Eric (Videnskabelig artikel)

¹⁴ Gameinformer: "Do gamers really like hard games" af Reeves, Ben (Artikel)

allerførst gerne vil vide hvad målet i et spil er, og at de mener at en tutorial vil kunne hjælpe med dette. Dette er også en brik i game design, som har kørt over de mest succesfulde spil som er blevet udgivet. I det ældste Mario spil, som blev udgivet, blev der sat ekstrem fokus på hvorledes at man kunne lave en simpel tutorial, som viste de mest basale funktioner i et spil ved at spilleren selv uden at være bevidst om de havde lært det.¹⁵

Nu da vi har kunne analysere os frem til, at det er essentielt for et spil, at det har en indbygget progression som matcher spillerens niveau, er det også vigtigt at analysere andre elementer i et spil. Grafikken i et spil er også et element, som er vigtigt i den forstand, at der ikke er noget spil uden noget grafik således man visuelt kan se spillet, men hvor vigtigt er det at grafikken er GODT? Ifølge vores respondenter, var grafikken et nedprioriteret emne ift. Funktionaliteten i spillet. Dette argumenteres også for i artiklen ”How important are graphics to games” hvori der nævnes, at grafik er vigtigt, men at det samtidig ikke skal override den funktionalitet som er i spillet. Der nævnes også videre i artiklen, at grafikken ikke nødvendig gør spillet sjovere, men at det godt kan skabe en illusion for dette.

Til sidst, er det også vigtigt at finde ud af, hvor vigtigt en spillers input i et spil er i forhold til hvorledes denne udvikler sig. I vores kvantitative analyse blev der besvaret af respondenterne, at det er et vigtigt element i et spil at man som spiller har mulighed for at ændre måden hvorpå spillets forløb foregår på. Dette indgår også i Salen og Zimmerman’s ”Rules of Play” teori som før beskrevet, hvori at det pointeres at det er vigtigt at alt som spilleren gør, har af betydning for systemet og at systemet har betydning for spilleren for at opnå ”Meaningful play”. Dette ses også i artiklen ”Story telling and interactivity in gaming video”¹⁶, hvori det analyseres at interaktivitet mellem spiller og system (spillet) har kæmpe betydning for, hvorledes et spil kan være succesfuldt. I artiklen fremstilles der bl.a., at for at storytelling i et spil kan være succesfuldt, skal systemet kunne respondere til spillerens aktioner i en måde hvorpå at det giver mening og at en måde hvor på at spilleren følger at der er en progression i spillet. Dette er også blevet pointeret i den kvantitativ analyse som vi har fremstillet.

¹⁵ Youtube: “Design Club - Super Mario Bros: Level 1-1 - How Super Mario Mastered Level Design” - Af Emmons, Dan (Video)











¹⁶ The Artifice: “Story telling and interactivity in gaming video” - Af deadeyemcduck (Artikel)

THE HORROR OF THE MAZE

Produktprincip

Krav til vores produkt

Ud fra vores problemanalyse, vil vi opsætte nogle krav for vores produkt, og herefter videreudbyde dem. Hernede under er der en opstilling af 5 hårde krav (Skal som vidt muligt eksistere i produktet) og fem "bløde krav" (nice to have ting).

	Hårde krav		Bløde krav
1.	Den intelligente fjende(r) skal gennem spillet udvikle sig og være mere udfordrende for spilleren gennem spillets progression. 	1.	Der skal være minimalt med bugs. 
2.	Signifiers og affordances skal være tydelige, således spilleren ved hvad målet og objektet i spillet er. 	2.	Komplekst miljø, nem at lære. 
3.	Spillerens aktioner skal have betydning for hvordan spillet reagerer. 	3.	Grafikken skal være stilrent og have sammenhæng med temaet. 
4.	Den intelligente fjende(r) skal udover være progressive, også have basisfunktioner som vidner om dens intelligens. 	4.	Produktet skal have en begynder tutorial, for at lære de basiske funktioner i produktet. 
5.	Det skal være motiverende for spilleren at spille spillet, ikke kun i starten, men videre i spillets forløb. 	5.	Alle elementer i spillet skal have en reel betydning i spillet. 

Alle figurer, er implementeret i word, og er derfor fri-tilgængelige.

THE HORROR OF THE MAZE

Kravuddybning

	Hårde krav uddybning		Bløde krav uddybning
1.	Grunden til at den intelligente fjende, skal udvikle sig progressivt er fordi, det er et krav, men også fordi at hvis en fjende blev ved med at gøre det samme, ville spilet blive for nemt at gennemskue, og derved for let at gennemføre. Derfor skal "AI" udvikle sig, og gøre spillet mere udfordrende og spændende.	1.	Et spil med bugs, er knap så godt. Men selv større spil kan have små bugs, som forstyrrer playthrough processen. Derfor ligger de fleste spil online, så de kan fikses. Men det er yderst nødvendigt at sørge for at bugs ikke er tilstede, og som ikke ødelægger hele spillets mening eller proces.
2.	Signifiers, er et "Must Have" i et spil, fordi man ligesom skal vise at der er noget interagerbart i spillet, og der derved sker en forbindelse mellem player og spillet. Udover dette, skal der være affordances, så player kan fortolke den signifier, og give sit bud på hvad der skal ske derefter.	2.	Miljøet er noget man ikke tænker så meget over, men det er stadigvæk nødvendigt at sørge for at det er grafisk appellerende for spilleren at spille spillet. Dette kan ske, ved at anvende textures, objectives eller andre relevante ting, som gør spillet mere visuelt appellerende eller bare mere visualiserende.
3.	Et spil, indeholder mange forskellige ting. Derved skal disse "ting" også kunne have betydning for spillet. Dette gør at hvis man nu har fundet et objective, medfører det en eller anden konsekvens, som både kan være positiv eller negativ. Dette gør bare spillet, mere interessant og "playable".	3.	Grafik som nævnt før er vigtig, men det kræver også at grafikken ikke ligner noget som ikke har relevans for temaet. Det vil sige at hvis man nu kørte et "horror" theme, vil det ikke give mening at bruge blomster og lyse farver i sin bane, men nærmere mørke farver, som passede til temaet, og dens visuelle fremstilling.
4.	Med en "AI" skal man kunne forstå dens handlinger til en vis grad. Dette vil sige, at hvis man nu er nær enden i et spil, går man vel ud fra at fjenden er ved at nå sit toppunkt, og er stærkere end nogensinde før. Dette kan ske gennem bedre "hørelse", stærkere "syn" eller mere "damage". Dette er alle relevante faktorer, som AI udvikler gennem spillers play proces.	4.	Alle spil der er konstitueret, er næsten aldrig det samme, så det vil sige at hvis man prøver et nyt spil, vil det være en fordel, at få en kort "tutorial" i hvordan man eventuelt kan spille spillet, og forstå dens vigtigste egenskaber og funktioner. Derfor kan en tutorial ikke kun hjælpe playeren, men også med at forstå meningen bag spillet, og hvad det går ud på.
5.	Underholdning er et af de begreber, som mennesker kender mest til. Derfor skal underholdningen i vores spil, også være vigtig at forstå. Dette kan ske, ved at implementere nogle objects, som gør spillet mere konsekvent, enten i sværhedsgrad eller i at det bliver nemmere at klare spillet. Begge dele kan virke, så længe at spilleren føler sig motiveret, og klar på nye udfordringer.	5.	Som mennesker kan vi godt lide hvis ting hænger sammen, og deler samme betydning. Dette gælder sig også i spil. Det giver altså mening at koble et "labyrint spil" sammen med små "hints", for at man ligesom kan finde ud af labyrinten. Dette viser bare at man har forstået spillets sammenhæng, og at de mindste ting, også kan have betydning for de største ting.

Idegenerering

Efter problemanalysen, er der nu oprettet en generel ide om hvad spillet skal indebære. Først og fremmest skal spillet være bestående af den såkaldte "AI" som skal fungere som meningen med spillet. For at finde den mest relevante løsning, er der nedenunder opstillet en liste af (AI NPC) spil.

Ideer indenfor AI

1. Skoleskyderi spil (AI NPC)
2. Horror spil (AI NPC)
3. Labyrint spil (AI NPC)
4. Tower defense spil (AI NPC)

Disse ovenstående ideer/løsninger, skal nu vurderes i forhold til problemanalysens opnåede viden, de pågældende krav og ved brug af forskellige teknikker. For at gøre det overskueligt, vil gruppens medlemmer, først vurdere spillene gennem et point skema, og derefter sammenligne favoritspillene med kravene, samt den opnåede viden. Se skema nedenunder.

1-5 Point	Jens	Zacharias	Anders	Emil	Total:
Spil 1	4	2	1,5	2,5	10
Spil 2	4	3	4	5	16
Spil 3	1	3	3	3	10
Spil 4	4	4	3	4	15

Som set ovenover er spil 2 ¹⁷(Horror Spillet) og spil 4 ¹⁸(Tower Defense Spillet) de mest vægtede spil, som gruppen finder relevant. Dog, skal disse spil nu vurderes i forhold til kravene og derefter gives feedback på det valgte tema. Der vil først blive oprettet et ¹⁹P&V skema, hvor der uddeles point i forhold til de to temaer, og til kravene. Derefter vil der blive yderligere uddybet hvorfor kravene har relevans for temaet nedenunder. Det skal dog påpeges, at de hårde krav, vil have større relevans, og der kun kort snakkes om de bløde krav.

¹⁷ Kotaku: Does Survival Horror Really Still Exist? - Af Alexander, Leigh (Artikel)

¹⁸ Gameranx: Best Tower Defense Games of all time - Af Reece, Damon (Artikel)

¹⁹ Vthoroe "Teknologi Håndbog" - af Thorø, Valle (E-bog)

THE HORROR OF THE MAZE

Der må maks gives 1-5 point, til hver krav. Det vil sige at hvis man føler at kravet, tilegner sig temaet, giver man den højest point (5), og hvis man syntes at kravet ikke tilpasser sig temaet, giver man den lavest point (1)

Z = Zacharias A. Nielsen

J = Jens B.H. Issa

A = Anders B. Senger

E = Emil Christiansen

P&V Skema	Hårdt krav 1.	Hårdt krav 2.	Hårdt krav 3.	Hårdt krav 4.	Hårdt krav 5.	Total:
Horror tema	Z = 5 J = 4 A = 3 E = 4	Z = 4 J = 5 A = 5 E = 4	Z = 3 J = 4 A = 5 E = 5	Z = 5 J = 4 A = 2 E = 3	Z = 3 J = 4 A = 5 E = 2	Z = 20 J = 21 A = 20 E = 19 Sum = 80
Tower Defense tema	Z = 3 J = 4 A = 4 E = 2	Z = 1 J = 3 A = 3 E = 2	Z = 2 J = 4 A = 3 E = 3	Z = 2 J = 2 A = 3 E = 4	Z = 3 J = 1 A = 2 E = 4	Z = 11 J = 14 A = 15 E = 15 Sum = 55

På baggrund af P&V skemaet, ser vi at i forhold til kravene, opnår vi bedst potentiale i at udarbejde et spil i "Horror" temaet. Se fordybning, at hvorfor kravene ikke passer eller passer til temaet nedenunder.

Horror spillet i forhold til de hårde krav.

- Krav nr.1
 - o Horrorspillet ville kunne fungere fint med en AI, men her vil der nok blive anvendt en "Intelligent Fjende" som havde til formål at søge hen mod playeren, på baggrund af horrortemaet.
- Krav nr.2
 - o Horrorspillet kunne sagtens have forskellige objectives og signifiers, da playeren nok skulle bevæge sig og interagere med objects eller andre fysiske ting.
- Krav nr.3
 - o Horror spillet, ville til en stor grad have betydning for spillets proces. Altså at jo længere man kom igennem banerne, jo sværere blev "AI" at dræbe for eksempel.

THE HORROR OF THE MAZE

- Krav nr.4
 - o Horror spillet, ville her understøtte AI, men nye egenskaber. Såsom bedre høreelse eller større damage parameter, jo længere spilleren kom gennem spillet.
- Krav nr.5
 - o Horror spillet, ville sørge for at det nok bliver mere ”skræmmende” og spille spillet, som også gør det mere opmærksomt for playeren, og at deres næste træk kan have store betydninger
- Alle bløde krav generelt
 - o Horror spillet vil selvfølgelig også indgå i de bløde krav, men det er her at det er svært at vurdere, fordi de bløde krav, kan strække sig i mange mængder. Man kan jo ikke sige, om dette tema, kommer til at have mange bugs, eller grafiske problemer, og derfor er der ikke den store vurdering.

Tower Defense spillet i forhold til de hårde krav.

- Krav nr.1
 - o Tower Defense spillet, vil her også understøtte ideen om at ”AI’en” udvikler sig. Dette kunne ske, hvis man nu havde tjent penge nok til at få en større base, og derfor skulle AI’en også være klar over dette, og derved sende stærkere og større styrker imod basen. Men dette vil være sværere at inkorporere i spillet.
- Krav nr.2
 - o Tower Defense spillet, vil også have signifiers og affordances, dog ville det nok ikke være så visuelt appellerende, fordi det nok foregår i 2D.
- Krav nr.3
 - o Tower Defense spillet vil også have betydning for processen i spillet, og at der kommer nye funktioner hen afvejs.
- Krav nr.4
 - o Tower Defense spillet vil også have en enemy, der udvikler sig i takt med at spilleren udvikler sig. F.eks. gennem nye opgraderinger af towers.
- Krav nr.5
 - o Tower Defense spillet, vil også motivere spilleren, til at opnå et højt niveau indenfor sit Tower Defense, og ligesom blive stærkere end fjenden.
- Alle bløde krav generelt
 - o Tower Defense vil selvfølgelig også indgå i de bløde krav, men det er her at det er svært at vurdere, fordi de bløde krav, kan strække sig i mange mængder. Man kan jo ikke sige, om dette tema, kommer til at have mange bugs, eller grafiske problemer, og derfor er der ikke den store vurdering.

Fra Ide → Horror Tema

På baggrund af de temaer, der lige er behandlet ovenover i forhold til kravene, kan der nok ses mest potentiale i ”Horror Spillet”, både fordi krav 2 er bedre til ”Horror Spillet” og at gruppen har valgt at foretrække horror spillet i point skemaet. Derudover ser vi bedst arbejds potentiale i et 3D produkt, da vi har stiftet bekendtskab til det førhen. Det skal dog siges at begge temaer godt kunne have opfyldt

problemstillingens krav, og derfor også have relevans for opgaven. Men efter overvejelser, over både point skemaet, de opstillede krav og egen interesse ser vi at der bedst kan laves et spil i "horror" temaet, og det appellerer mere mod os, og det gør også at produktet bliver mere interessant at oprette og konstruere.

Produktudformning

I denne del af rapporten, ville alt funktionaliteten i vores spil, de forskellige signifiers og affordances i spillet, koden og tankerne bag de forskellige designvalg gennemgås. Alt dette vil blive sat op mod vores krav, problemformulering og kriterier løbende i beskrivelserne.

Præsentation af spillet

Spillet "The Horror of the Maze" bundler op i temaet om at en mand er blevet luret ned til et forladt mentalhospital, i den tro at vedkommende skulle udføre nogle tjenester for en kammerat. Hvad vedkommende så i stedet realiserer når han indtræder på hospitalet er, at dørene til hospitalet bliver låste bag ham, og han kun kan komme ud af hospitalet, hvis han indsamler alle noterne. Han realiserer også samtidig, at der er en (blind) fjende som er ude efter ham, og denne fjende bliver både hurtigere og kan høre ham længere væk for hver note manden indsamler. Fjende kan godt som sagt kun HØRE ham og har ikke mulighed for at fange manden, hvis denne står stille.

Spilleren

Spilleren har forskellige muligheder for at bevæge sig rundt i spillet i forskellige hastigheder, og møder gennem spillet forskellige signifiers, affordances og feedback som hjælper spilleren gennem spillet således at målet for spilleren gennem spilforløbet er klar.

Spilleren har følgende controls til anvendelse i spillet:

W - Fremad

S - Bagud

D - Højre

A - Venstre

Control - Crouch

Shift - Run

E - Åbne døre/Pick-up noter

F - Tænde lommelygte

Spillerens bevægelse & lommelygte

Da temaet er et mentalhospital, er banen helt mørk, og der er ingen naturlig lys der bryder igennem hospitalet. Derfor er der anvendt et flashlight, som ses herunder:

Koden som bliver anvendt til dette ses forinden:

THE HORROR OF THE MAZE

```
5 public class Flashlight : MonoBehaviour
6 {
7     private bool flashLightEnabled;
8     public GameObject flashLight;
9
10    // Start is called before the first frame update
11    void Start()
12    {
13    }
14
15    //hvis keypress F sker, virker child spotlight lys og bliver enabled i spillet
16    // Update is called once per frame
17    void Update()
18    {
19        if (Input.GetKeyDown(KeyCode.F))
20        {
21            flashLightEnabled = !flashLightEnabled;
22        }
23        if (flashLightEnabled)
24        {
25            flashLight.SetActive(true);
26        }
27        else
28        {
29            flashLight.SetActive(false);
30        }
31    }
32 }
33
```

Figur 1 - Flashlight kode

I denne kode ses der, at der anvendes en if-sætning til at sørge for, at hvis keypress down F sker, så er flashlightet(spotlightet) aktiveret. Dette kan ske, da der til playerobjektet er blevet sat et spotlight child på, som aktivieres ved keypress F.

Som før set, er spilleren kontrolleret identificeret ved forskellige movements, hvor han kan løbe i alle 4 retninger og samtidig har crouch/sprint funktionen tildelt. Koden herunder er "mouse look" scriptet, som styrer musen i forhold til bevægelse:

```
Miscellaneous Files - MouseLook
3 using UnityEngine;
4
5 public class MouseLook : MonoBehaviour
6 {
7     //tjekker mouses sensitivity
8     public float mouseSensitivity = 100f;
9
10    public Transform playerbody;
11
12    float xRotation = 0f;
13    //sørger for at musen ikke er visible
14    // Start is called before the first frame update
15    void Start()
16    {
17        Cursor.lockState = CursorLockMode.Locked;
18        Cursor.visible = false;
19    }
20
21    //Ganger mousesensitivity med hastighed i spillet
22    //Gør det muligt at rotere
23    // Update is called once per frame
24    void Update()
25    {
26        float mouseX = Input.GetAxis("Mouse X") * mouseSensitivity * Time.deltaTime;
27        float mouseY = Input.GetAxis("Mouse Y") * mouseSensitivity * Time.deltaTime;
28
29        xRotation -= mouseY;
30        xRotation = Mathf.Clamp(xRotation, -90f, 90f);
31
32        transform.localRotation = Quaternion.Euler(xRotation, 0f, 0f);
33        playerbody.Rotate(Vector3.up * mouseX);
34    }
35 }
```

Figur 2 - Mouselook

Koden foroven styrer basalt, hvorledes musen reagerer på hastigheden af den bevægelse som spilleren foretager sig. Spillerens bevægelse er defineret i scriptet "Player movement", som ses forinden:

THE HORROR OF THE MAZE

```
public class PlayerMovement : MonoBehaviour
{
    public CharacterController controller;
    //alle variabler i forhold til speed og lignende
    public float speed = 4f;
    public float gravity = -9.82f;
    public float jumpHeight = 0.1f;
    public KeyCode runKey = KeyCode.LeftShift;
    public KeyCode crouchKey = KeyCode.LeftControl;
    public KeyCode tiltLeft = KeyCode.Q;
    public KeyCode tiltRight = KeyCode.E;

    public Transform groundCheck;
    public float groundDistance = 0.4f;
    public LayerMask groundMask;

    Vector3 velocity;
    bool isGrounded;
    // Update is called once per frame
    void Update()
    {
        //tjekker om playeren er grounded
        isGrounded = Physics.CheckSphere(groundCheck.position, groundDistance, groundMask);

        if (isGrounded && velocity.y < 0)
        {
            velocity.y = -2f;
        }
    }
}
```

Figur 3 viser variabler til playermovement

I den foroven kode bliver der defineret de forskellige variabler anvendt i movement scriptet, og hvor hurtig disse skal være. Heri bliver der også defineret keycodesne, og i Update metoden bliver der defineret om spilleren er grounded.

```
//x og y aksens bevægelse
float x = Input.GetAxis("Horizontal");
float z = Input.GetAxis("Vertical");

Vector3 move = transform.right * x + transform.forward * z;

//jump funktionen
if (Input.GetButtonDown("Jump") && isGrounded)
{
    velocity.y = Mathf.Sqrt(jumpHeight * -2f * gravity);
}

controller.Move(move * speed * Time.deltaTime);
//w key til fremad og shift key til løb, speed variablen sat op når der bliver løbt
if (Input.GetKey(KeyCode.W) && Input.GetKeyDown(runKey))
{
    speed = 7f;
}
if (Input.GetKeyUp(KeyCode.W))
{
    speed = 4f;
}
```

Figur 4 viser koden til bevægelse i playermovement

Der er i scriptet ovenover defineret hvordan spilleren skal bevæge sig i x og z retningen, og der er blevet instantseret en keycode til jump funktionen. Der er også blevet påsat, at jumpfunktionen (og andre bevægelsesfunktioner) kan ske, når spilleren er "grounded".

Der er i koden foroven også defineret keycode w og keycode til at løbe (shift) i en if-sætning med en konstant variabel for speeden til løb. Denne form for kode gentages også i spilleren andre bevægelsesmønstre, bare med forskellige variabler og keycodes.

Pick-up, notecounter og UI-tekst (signifiers og affordances)

Spillerens mål med spillet, er at kunne samle alle noterne og derefter nå ud til exit-døren før monstret fanger vedkommende ved hjælp af lyd. Men for at kunne realisere dette, har der været en nødvendighed at bygge en counter, som kunne have kunne tælle hvor mange af disse noter som kalkulerer hvor mange objekter af tagget "noter" spilleren har samlet op. Disse objekter af tagget "noter" har også betydning i forhold til hvor progressivt sværere AI'en er, men dette redegøres for mere grundigt ved "Den progressive AI". Inventory score er således:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class InventoryScript : MonoBehaviour
{
    public List<string> Items = new List<string>();

    // Start is called before the first frame update
    void Start()
    {
    }

    // Update is called once per frame
    void Update()
    {
    }

    public void AddItem()
    {
        Items.Add("item 1");
        print(Items.Count);
    }
}
```

Figur 5 - Koden der viser inventory scriptet

Scriptet ovenover, er en simpel string, som holder alle vores objekter af tagget "note" styr på inde i en liste, og at der er et tomt "item" script, som anvendes til at adde i denne liste

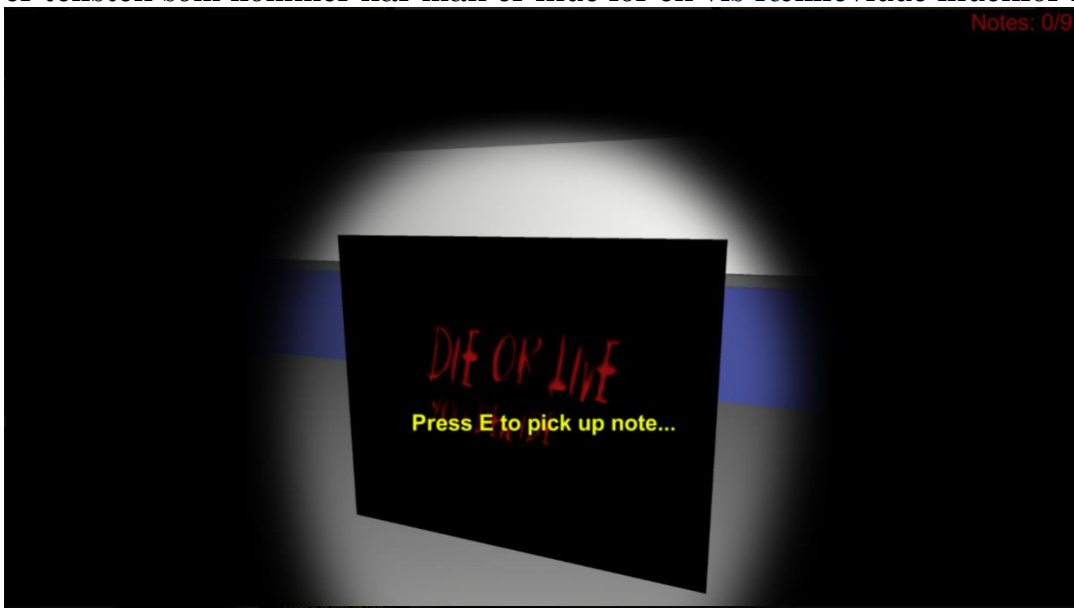
Indholdet i denne liste kommer ved hjælp af de objekter som spilleren indsamler gennem spillet. Dette gøres ved hjælp af følgende script, som hedder "PickUpObject", som vises nedenunder:

THE HORROR OF THE MAZE

```
17 //find tag af spiller, disable UI teksten således det ikke kommer i starten
18 // Inventory = GameObject.FindWithTag("Inventory");
19 Player = GameObject.FindWithTag("Player");
20 PressText = GameObject.Find("notePressE").GetComponent<Text>();
21 PressText.enabled = false;
22
23 //når spilleren når inde i collideren, kommer UI tekst frem
24 void OnTriggerEnter(Collider other)
25 {
26     PressText.enabled = true;
27 }
28
29 //når spilleren går ud af collideren, kommer UI tekst frem
30 void OnTriggerExit(Collider other)
31 {
32     PressText.enabled = false;
33 }
34
35 void Update()
36 {
37     if ((Player.transform.position - this.transform.position).sqrMagnitude < 2 * 2)
38     {
39         //Ødelægger noterne når key press down E
40         print("enabled");
41         if (Input.GetKeyDown(KeyCode.E))
42         {
43             Destroy(gameObject);
44             PlayerManager.AddToInventory();
45             PressText.enabled = false;
46         }
47     }
48 }
49
50
51
```

Figur 6 - Viser koden til noterne

Heri ses der hvorledes at når keypress down E sker ødelægges noterne og der adderes en note ind i inventoryen. Ovenover ser man også en signifier blive programmeret, som er teksten som kommer når man er inde for en vis rækkevidde indenfor noten:



Figur 7 - Viser signifiers og notecounteren

Heri ses der, at der er en signifier når spilleren ankommer ind for en vis rækkevidde indenfor noten. Når spilleren trykker "E" destrueres noten, som skrevet i scriptet, kommer der en affordance, og der adderes en til noterne oppe i højre hjørne (som ses i billederne). Dette tydeliggør kravet om, at spilleren ved hvad målet i spillet er, og selv har en interaktivitet til at ændre spillets forløb, da hver note gør AI'en stærkere (specificeres længere nede)

Den progressive AI

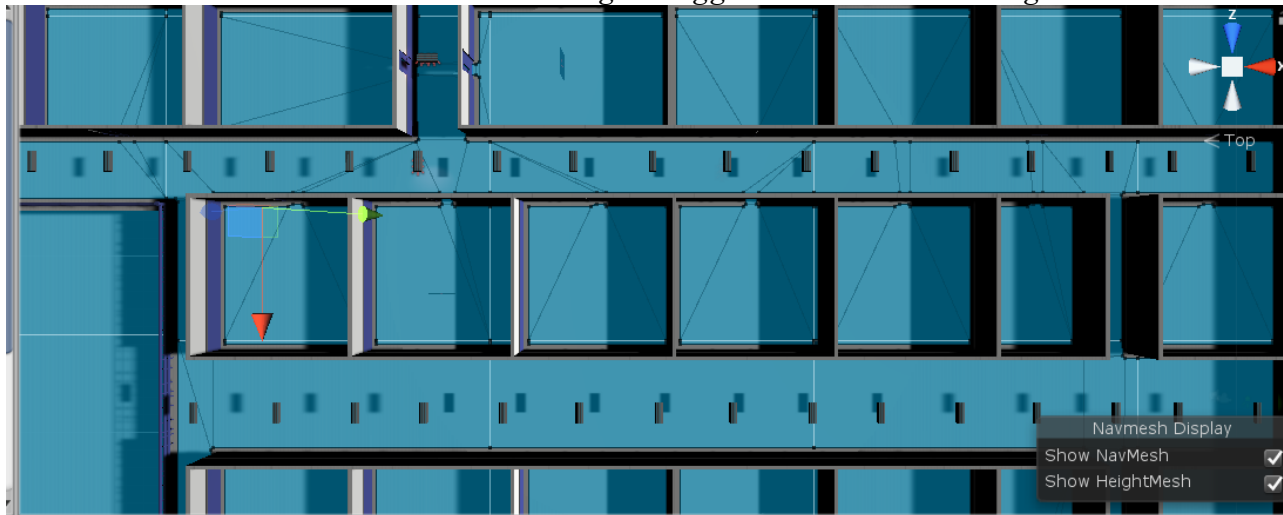
Den progressive AI er den fjende, som spilleren skal sørge for at undgå gennem hele spillet. Den intelligente fjendes egenskaber har af relation til kravene for selve oplægget og de krav som vi selv har stillet til vores spil. Den progressive fjende skal være progressiv og udvikle sig gennem spillerens aktioner, har en basis intelligens og være motiverende for spilleren at have imod.

AI'en i vores spil har nogle forskellige egenskaber som er følgende. Disse egenskaber vil blive diskuteret og vist kodeeksempler til forneden:

- AI'en kan IKKE se spilleren når denne står stille
- AI'en bliver hurtigere og radiussen hvor den kan "høre" spilleren bliver større for hver note som spilleren samler op
- AI'en har en basis intelligens, der gør at den kan pathfinde gennem hele banen via waypoint og undgå vægge
- AI'en kan åbne døre, således det ikke er for let for spilleren
- AI'en har en signifier i og med at den afspiller lyd når den er inde for en vis distance for spilleren

General intelligens (pathfinding, waypoints, åbne døre)

AI'ens generelle intelligens baserer sig ud fra det faktum, at den kan identificere hvor den skal hen, og sørge for at den ikke går ind i andre objekter i banen. Dette gøres ved hjælp af et indbygget unity komponent, som har ved navn NavMesh surface, som kan automatisk diktere hvorhenne der er nogle vægge i en baneudforming. Ses forneden:



Figur 8 - Viser navmeshagent surface

Alt det blå signalerer til den progressive AI hvorhenne i banen den kan gå på, og hvorhenne den IKKE kan gå på. Hertil er der også et indbygget off-mesh til dørene, som fungerer på samme måde som resten af NavMesh surface komponentet. AI'en kan

THE HORROR OF THE MAZE

bevæge sig gennem banen uden at se spilleren ved hjælp af såkaldte "waypoints", som er bygget puttet manuelt gennem banen. Ses ved koden nedenunder:

```
//får monsteren til at gå rundt til waypoint og tjekker om den har nået waypoint
void WalkToWayPoint()
{
    if (destArrived)
    {
        SetNextWaypoint();
        agent.SetDestination(nextWaypoint.transform.position);
        destArrived = false;
    }

    float distance = Vector3.Distance(nextWaypoint.transform.position, agent.transform.position);
    if (distance < 5)
    {
        print("arrived");
        destArrived = true;
    }
}

//vælger en random waypoint
void SetNextWaypoint()
{
    var rand= Random.Range(0, waypoints.Length - 1);
    if(rand == nextWaypointNum)
    {
        //do it again because you found the same endpoint
        SetNextWaypoint();
    }
    nextWaypoint = waypoints[nextWaypointNum];
    nextWaypointNum = rand;
}
```

Figur 9 - Kode til waypoints for AI'en

Heri ses der at ved hjælp af metoden "WalkToWayPoint" er der en if-sætning, som sørger for at AI'en går hen til waypointsne (som er tomme gameobjekter, og som er tagget "waypoints") ved anvendelse af transform.position og monsteret vælger et random waypoint ved hjælp af metoden "SetWayPoint". Monsteret har også skulle have en basis intelligens i, at da banen er fyldt med mange døre, skulle vi sørge for at der ikke var en for let sværhedsgrad, og sørge for at spilleren ikke bare kunne gennem sig i en af dørene ved hjælp af følgende script:

```
private void OnTriggerEnter(Collider other)
{
    if (other.gameObject.name.Contains("UnlockedDoor"))
    {
        if (other.gameObject.GetComponent<UnlockedDoor>().animator.GetBool("Open") == false)
        {
            print("open door");
            other.gameObject.GetComponent<UnlockedDoor>().doDoorOpen();
        }
    }
}
```

Figur 10 - Collider til unlocked door for monsteret

Dette gøres vha. en OnTriggerEnter metode, hvorhenne at der sørges for at animationen for dørene i vores bane bliver aktiveret når monsteret går tæt på denne.

THE HORROR OF THE MAZE

Progressiviteten og interaktionen med spilleren

Da vi ud fra vores problemformulering og krav skulle have en AI, som har en progressiv sværhedsgrad, og dens niveau skulle stige efter hvor godt spilleren klarer sig, er der blevet lavet nogle funktioner til dette. Hertil går AI'en efter "lyd", hvor at den ikke kan se spilleren, når denne står stille. Det er ikke rigtig programmeret efter lyd, men det er mere en illusion af lyd, hvori der er blevet anvendt lookRadius, hvor enemy går efter spilleren når denne er inde for lookRadius. Men enemy kan ikke se spilleren når denne ikke bevæger sig, dvs. den kan ikke se spilleren når denne ikke laver "lyd".

Den mest basale interaktion der er mest spilleren, er at når AI'en hører(ser) spilleren, kan denne løbe efter ham. Dette realiseres i koden nedenunder:

```
//tjekker om spilleren bevæger sig med metoden isPlayerMoving
//tjekker om spilleren er inde for "høreradiusen". Hvis den er inde for Target, så vil den løbe efter vedkommende hvis playeren laver lyd
// Update is called once per frame
void Update() {

    float distance = Vector3.Distance(target.position, transform.position);
    PlayMonsterSound(distance);

    if (distance <= lookRadius && IsPlayerMoving())
    {
        destArrived = true;
        agent.SetDestination(target.position);
        FaceTarget();

        if (distance <= agent.stoppingDistance)
        {
            FaceTarget();
        }
    }
    else
    {
        WalkToWayPoint();
    }

    UpdateLookRadius();
    UpdateSpeed();
}
```

Figur 11 - Om monsteret er inde for høreadiusen, og tjekker om spilleren bevæger sig

Den kode foroven er en udsnit af "EnemyController" scriptet. Update() metoden styrer hvorledes at når AI'en ser target (spilleren) og går efter target vha. SetDestination() metoden går efter spilleren. Hvis enemy så mister synet("hørelsen") af playeren, går den videre til waypoints. Den mister synet på spilleren, når denne ikke bevæger sig længere (dvs. ikke afgiver lyd længere) som ses i nedenstående script:

```
//checker om player bevæger sig ved at tjekke gamle position er lige med nye position
public bool IsPlayerMoving()
{
    curPos = target.transform.position;
    if (curPos == lastPos)
    {
        return false;
    }
    lastPos = curPos;
    return true;
}
```

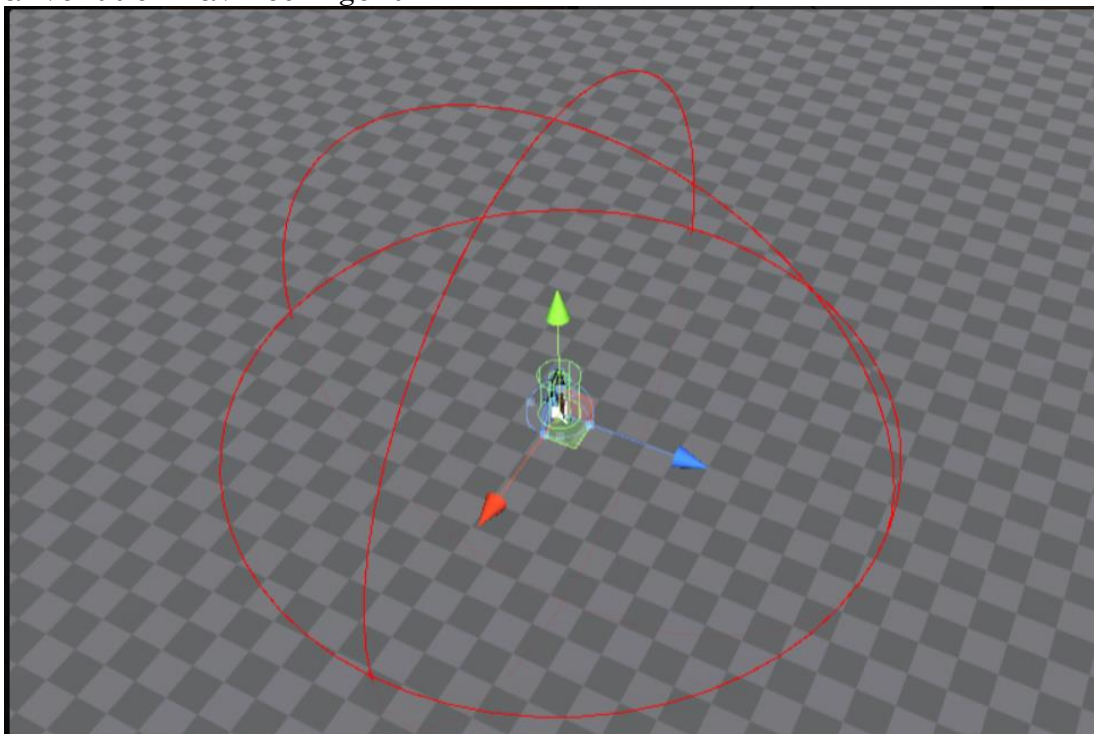
Figur 12 - checker om spilleren har bevæget sig

THE HORROR OF THE MAZE

I dette kode, checkes der om playeren havde samme position som forrige, og hvis denne ikke havde samme position går AI'en mod spilleren og hvis playeren havde samme position som før, dvs. står stille, og ikke afgiver lyd, går AI'en ikke imod en.

AI'en har også en progressivt gennem spillet, som selvfølgelig var et core tema i dette produkt. Progressiviteten i spillet afhænger af hvorledes spilleren klarer sig i spillet, og jo bedre spilleren klarer sig, jo sværere blive AI'en og mere udfordrende. AI'ens lookRadius ("høre radius", dvs der hvor AI'en går efter spilleren når denne bevæger sig) og speed (løbehastighed) bliver højere, for hvert eneste noteobjekt som bliver picked op.

Nedenunder bliver der set hvorledes radiusen ser ud. Komponentet i unity som er anvendt er NavMeshAgent:



Figur 13 - Navmeshagent radius

Denne radius og speeden bliver progressivt større ved hjælp af følgende kode:

```

}
//radiusen og speeden bliver større for hver note
void UpdateLookRadius()
{
    lookRadius = (2 * PlayerManager.GetInventory()) + 6;
}

void UpdateSpeed()
{
    agent.speed = PlayerManager.GetInventory() + 4f;
}

```

Figur 14 - Progressivitet

Disse to update metoder til både lookRadius og speed gør således, at for hver gang at inventory klassen får et objekt adderet til dens liste, gøre det således at den radius som bliver, vist i det billede før bliver større, og det gør således at den bliver hurtigere. Denne form for progression gør således, at spilleren har indvirkning til hvordan monsteret bliver sværere. Monsteret har også en signifier, i og med at indenfor en konstant radius, kan spilleren høre monsteret growle, som ses i koden heri:

```

//checker om spiller er i radius for at monster lyden skal spilles
void PlayMonsterSound(float distance)
{
    if (distance <= soundRadius) {
        //play sound hvis lyden ikke allerede spiller
        if (!monsterSound.isPlaying)
        {
            monsterSound.Play();
        }
    }
    else
    {
        monsterSound.Stop();
    }
}

```

Figur 15 - Signifier AI

Dette er en signifier for spilleren, som giver spilleren mulighed for at spille strategisk ift. Monsteret i og med at spilleren ved hjælp af lyd kan identificere hvor monsteret er henne og kan stå stille for at få monsteret til ikke at gå imod spilleren. Monsteret opfylder derved ønsket om at denne bliver progressivt sværere for spilleren, at spillerens interaktion med systemet kan ændre hvorledes monsteret opfører sig og at monsteret har en basis intelligens, i og med at den selv kan føre sig igennem banen ved hjælp af waypoints og navmesh surface.

Design

Bane

Spillet design er meget simpel. Det gik egentligt ud på at lave de byggeblokke der var nødvendige for at bygge vores bane. Der er ikke blevet gjort det store i forhold til Terrain, da vores spil kun foregår indenfor. Det vil sige de assets vi har lavet, består af simple vægge, døre osv. De assets vi selv har lavet, ser således ud. Billederne er screenshots fra "Autodesk fusion 360".



Figur 16: Screenshots af assets lavet i Autodesk fusion 360.

Da vi startede projektet, gik vi i gang med at planlægge spillet, hvad skulle det handle om, hvordan skulle det laves, og hvilke assets havde vi brug for. Efter den første planlægnings periode, gik vi i gang med at lave vores egne assets. Der er kun blevet gjort brug af egne assets til vores spil. Alle assets er blevet lavet i tre forskellige programmer: "Autodesk fusion 360", "Blender" og "Photoshop". Vi modellerede de forskellige assets der var behov for, da der var begrænset tid, var det vigtig kun at lave det mest nødvendige. Man kunne derefter begynde at lave assets til dekoration. Banen er bygget på 9 forskellige størrelse af plader, da der skulle bruges UV maps for lyset skulle virke korrekt. Der er blevet gjort brug af "Progrids" i Unity for at kunne placere væggene. Kanterne indenfor består af en pæl med samme textures som væggene. Banen består af to etager, begge er delt op i rum, hvor man skal finde noter igennem banen for at undslippe. Noterne er blevet lavet i "Photoshop". Se note afsnittet.

Billeder af alle assets ligger i bilag.

Grafik

I nedestående afsnit, vil man få et indblik i grafik fasen af projektet. Ikke alt grafik vil blive dokumenteret, da der er så meget. De vigtigste grafik elementer, vil blive forklaret nedenunder.

Oprettelse af AI

Papir fasen

Efter vi bestemte at det skulle være et horror spil, skulle vi fremstille en form for intelligent fjende. Denne fjende, skulle selvfølgelig have et skræmmende udseende, for at tilpasse sig temaet og resten af spillet. Helt grafisk, skulle der derfor fremstilles et monster, som ville kun være i stand til at have en animation, movement og textures på sig. Ideen til hvordan den intelligente fjende skulle se ud, blev gjort gennem nogle tegne skitser.

Se de tre (grove) skitser herunder.

Til højre ses skitse 1, af eksempel på det et monster (Delta).

Dette er egentligt bare en repræsentation af et mennesker, der er omformet nogle steder.

Igen til højre kan man se skitse 2 af monster (Beta), som egentligt er en slags "hund" der har store arme, og som har en kraftig struktur.

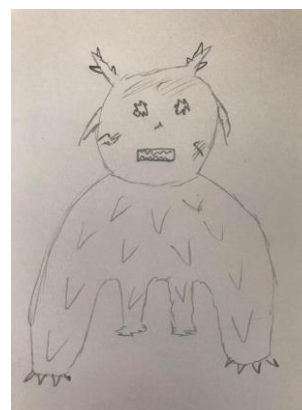
Til sidst, kan man se monster (Alpha) til højre, som er et monster, i menneske form, der har uddybninger på kroppen, men også skarpe pikke rundt omkring sin krop.

Efter gruppens vurdering, valgte vi at gå med monster (Alpha) da vi syntes at det havde den mest optimale struktur i forhold til temaet "horror" og også at det var visuelt appellerende at arbejde med.

Efter vi havde valgt den pågældende skitse, skulle der nu udformes et fysisk objekt. Dette blev gjort gennem 3D builderen "Blender", som er et program, hvor man kan oprette 3D builds, og skabe sin egen 3D modeller.



Figur 17 viser monster (Delta)



Figur 18 viser monster (Beta)



Figur 19 viser monster (Alpha)

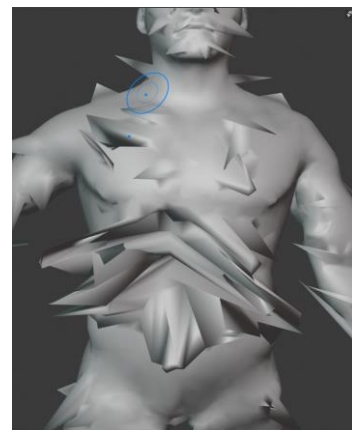
3D modellerings fasen

Som nævnt før, skulle skitse (Alpha) nu oprettes rent visuelt og grafisk. Det hele startede ud med, at der blevet lavet en base model, som egentligt bare lignede et menneskeligt (hankøn) opbygning. Efter dette, blev der tilføjet forskellige features, såsom de forskellige pikke, der stikkede ud rundt omkring kroppen. Der blevet også lavet en udhuling omkring dens mave, for at gøre AI'en mere usædvanlig og skræmmende.

Derudover, har resten af kroppen, selvfølgelig også have relevante features, forskellige model principper og en visuel fremstilling der passede til temaet. For at se alle sider af AI, dens struktur og knogleskelet og patteren henvises der til bilag 7, for at se alle vinkler af AI.

For at hurtig vise AI'ens struktur, kan man se på figur 10, hvordan den ser ud i bevægelse, altså en normal gå gang.

Som man kan se er AI, af menneskelig form, og har menneskelige træk, men den har usædvanlige former nogle steder, som sørger for at der er diversitet mellem playeren og AI. For at gøre AI mere skræmmende, blev der tilføjet et patteren, som findes nedenunder. Dette patteren erstatter hele kroppen, så AI fremstår som et rødt og hvidt/grå og sort monster. Grunden til at den har forskellige farver, er fordi banen er sort, og komplet mørk, og derfor ville det give mening, hvis man kunne spotte AI fra en vis distance. For at vise den sidste visuelle præsentationen af AI, kan man se det færdige visuelle produkt på figur 8.

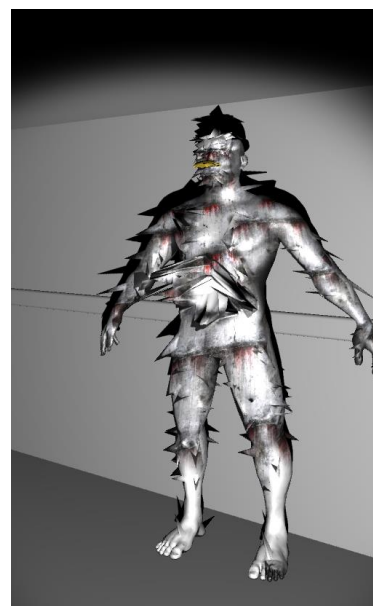


Figur 20 viser AI's mave struktur



Figur 21 viser AI i bevægelse

Den Intelligente Fjende →

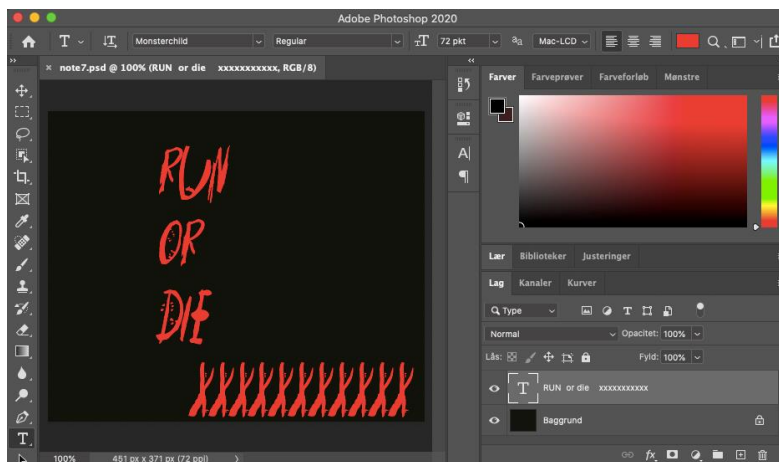


Figur 22 viser AI ENEMY

THE HORROR OF THE MAZE

Noterne

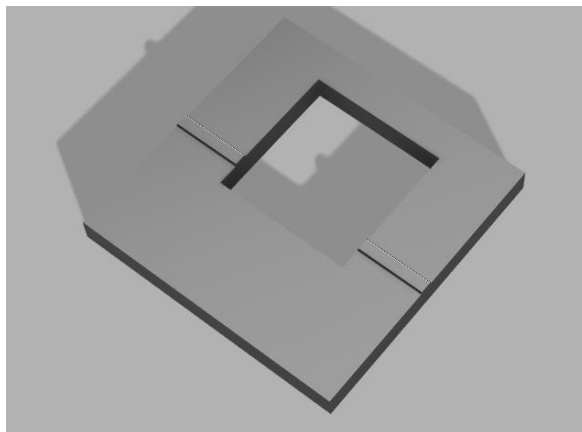
Noterne som er det objective man skal finde i spillet, blev oprettet gennem Photoshop. Det er i alt 9 noter, som man skal finde. Disse noter har hver deres egen udseende, som gør sig gældende i spillet. Det var ikke det helt store, at fremstille disse noter. Alle noter kan findes nede i bilag 8. For at hurtig gennemgå en af noterne, kan der tages udgangspunkt i note 7 som set nedenunder. Først blev der opstillet et sort baggrund, og derefter blev der hentet en "horror" font, for at det tilpassede temaet. Denne font, blev så brugt til at fremstille teksten på noten. Disse noter blev derefter implementeret i programmet som objectives.



Figur 23 viser note 7 (PS)

Textures

Til teksturerne er der blevet gjort brug af "Autodesk fusion 360" funktion med materiale og appearance. Der er brugt de indbyggede funktioner til teksturerne. Alle 3D modellerne har fået en simpel tekstur for at give spillet mere personlighed. Til test faserne af spillet brugte vi den standard tekstur "Satin Steel". Denne tekstur ville ikke passe til vores spil, da alt ville være gråt.

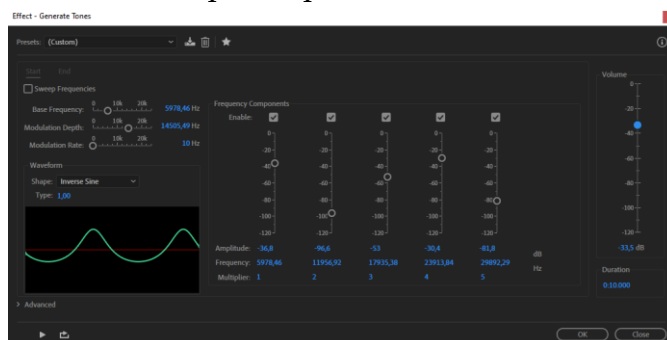


Figur 24: Screenshot af et asset der ikke blev brugt, med det standard materiale (Satin Steel)

Lyd

Lyd, musik og toner er relevante når man snakker computerspil. Især hvis formålet er at konstruere noget gennem lyd. Hertil kan man oprette baggrundsmusik, som sætter stemningen for miljøet eller andre relevante faktorer.

Baggrundsmusikken og alle de andre lyde, som er en essentiel del for spillet, blev lavet ved brug af forskellige toner. Disse toner, blev konstrueret i programmet "Adobe Audition". Adobe Audition er et lyd oprettelses program, hvor man både kan optage og konstruerede lyde eller toner.



Figur 25 viser adobe audition

I Adobe Audition, blev der derudover anvendt forskellige synthesizers, autotunes osv. Efterhånden, blev der lavet et par toner, så de kunne blev sat sammen. Disse toner havde selvfølgelig overensstemmelse med programmets tema "AI og HORROR"

Da lydene så var oprettet, skulle det mixes sammen, som blev gjort gennem hjemmesiden "Soundtrap.com" Altså, tonerne, blev sat sammen, i forskellige egenskaber. Soundtrap har mange egenskaber såsom om det skulle være dystret, altså en dybere tilgang til tonen eller om der skulle noget bass ind over. Endnu en egenskab, kunne være hvordan lydene skulle afrundes. Altså at inden tonen sluttede, skulle den afrunde med et mindre lydniveau end hvad det tidligere havde været. Dette kan man høre, hvis den "AI" som er i spillet finder og dræber dig. Altså lyden som også er "Jumpscaren".

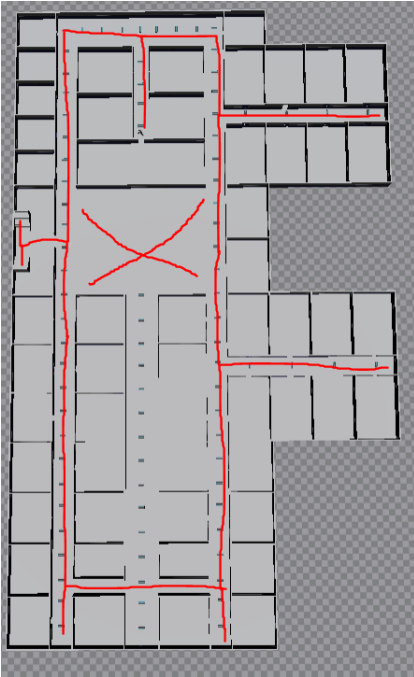
Desuden, skulle lydene også implementeres i programmet. Selve lydfilen, blev omdøbt til en ".mp3" fil, fordi det er den mest optimale lydfil i Unity. Alle lydene bliver anvendt i programmet på et eller andet tidspunkt, og det hele afhænger af hvor lang man er nået i spillet.

Implementering af bane

Implementering af banens udformning

Banens udformning blev konstrueret ud fra en hurtig skitse der blev lavet i starten af forløbet. Hele banen er bygget op af prefabs der er blevet lavet af gruppen selv. Banen er blevet bygget med blindgyder, smutvej og "chokepoints" for at gøre det nemmere eller sværere for spilleren i visse tilfælde.

THE HORROR OF THE MAZE



På billedet kan man se gange (røde linjer) og et åbent område (rødt kryds) hele bygningen er i to etager og hænger sammensat med et par trapper som laver et "Chokepoint"

Figur 26 viser kort over banen

Implementering af dørene

Dørene i banen er lavet blå for at passe ind med resten af banens visuelle stil samtidig når man går i nærheden af "oplåste" døre så vil der komme en gul tekst op som siger "Press E to Use Door" som en signifier. Alle døre som der ikke er "oplåste" der vil komme en besked op i rød som siger "The Door Seems Locked" for at sigde til spilleren at han ikke kan komme derind



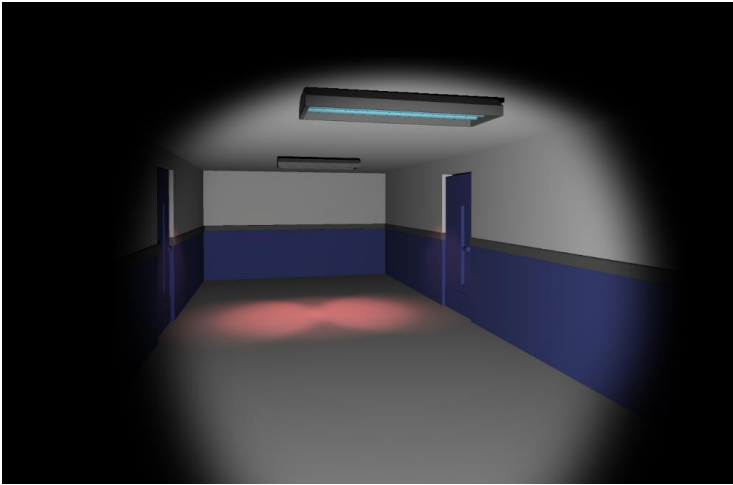
Figur 27 viser UI når man kommer tæt på oplåste døre

Implementering af lysene

Lyset i banen er der næsten intet af men det som er i banen, er rødt lys. Dette røde lys betegner at der er en note i nærheden og dette bruges som en signifier. Dette er også

THE HORROR OF THE MAZE

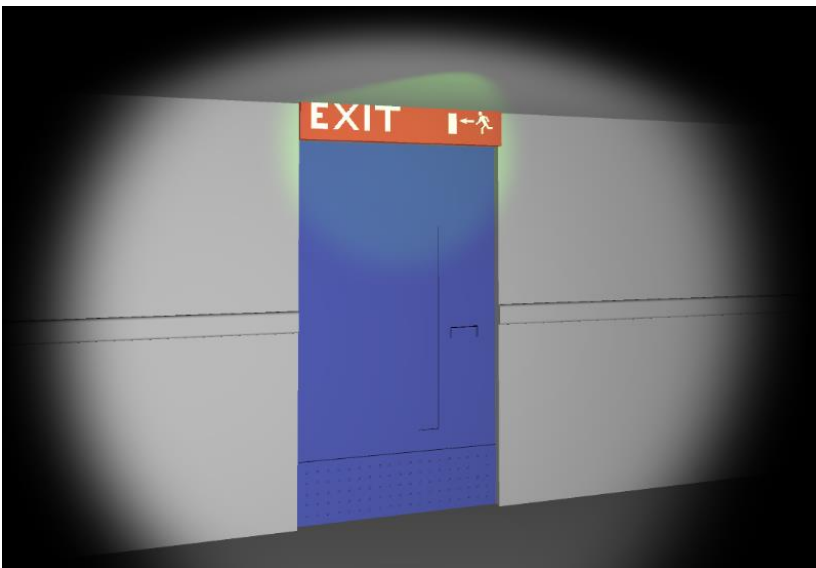
noget som spillere selv kunne finde frem til, men det er også en ting som står i tutorialen til spillet.



Figur 28 viser det røde signifier lys

Exit-door

Exit døren er lavet på den måde at når man har samlet alle noter så tænder et grønt lys på Exit skiltet over døren der virker som en affordance for spilleren.



Figur 29 viser når man har samlet alle noter og det grønne lys tænder

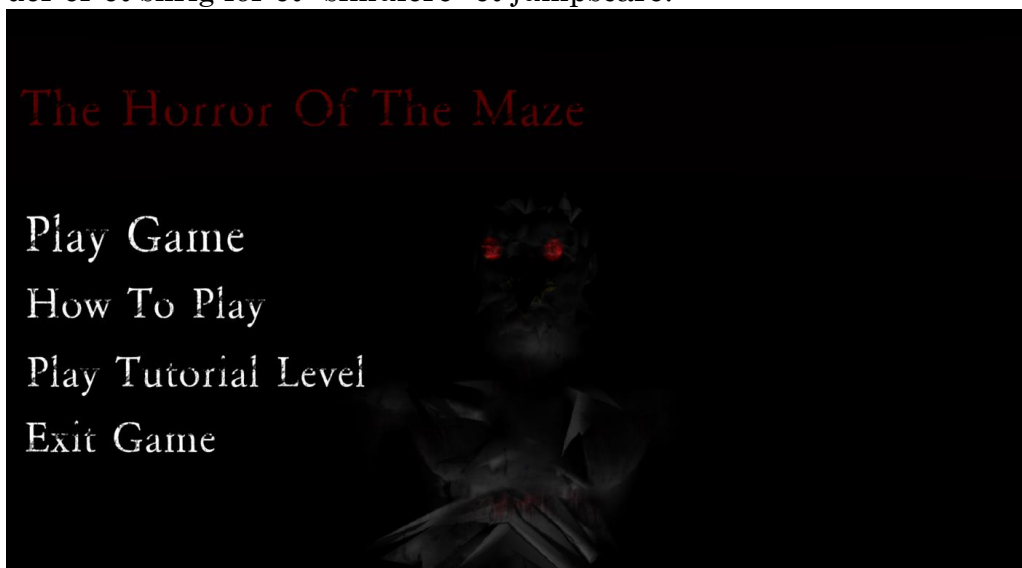
Menu

Menu visualiseret

Menuerne I spillet er blevet lavet på en måde så det passer til temaet til spillet og på samme tid er der også et billede af monsteret for at give en ide om hvad man er oppe

THE HORROR OF THE MAZE

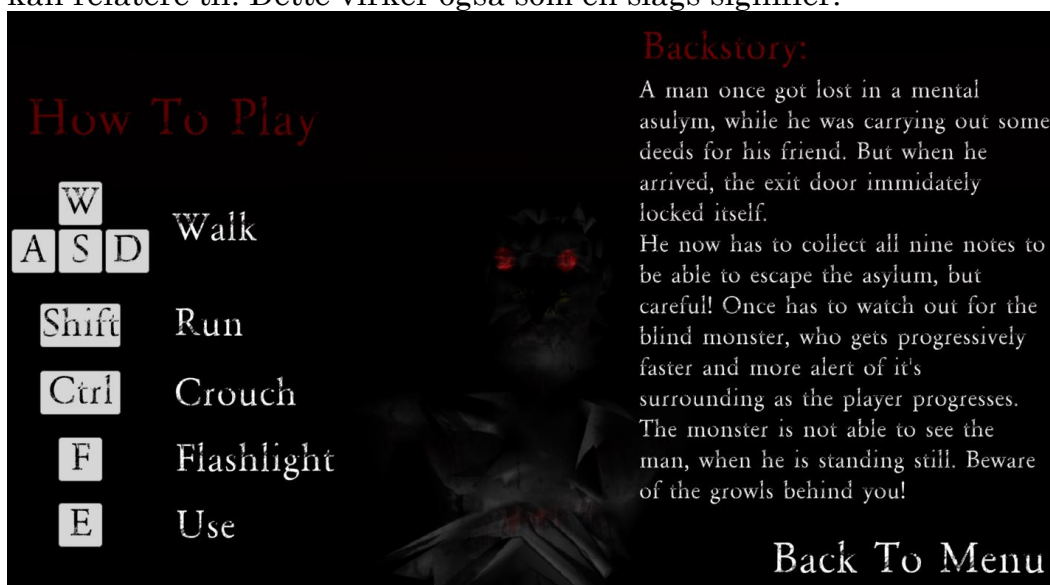
imod. Der er også en Win og Lose skærm som er identiske udover at i Lose skærmen der er et skrig for et "simulere" et jumpscare.



Figur 30 viser hovedmenuen

How-to

How to menuen er lavet på en måde så man visuelt kan se hvilke knapper man skal trykke på da der er blevet billeder som ligner taster på et tastatur som spilleren så kan relatere til. Dette virker også som en slags signifier.



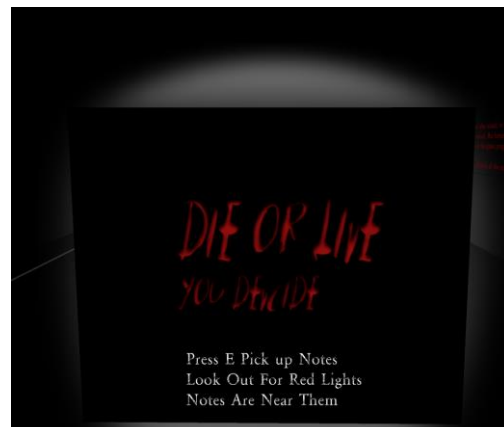
Figur 31 viser menuerne og controls

THE HORROR OF THE MAZE

Tutorial

Del 1:

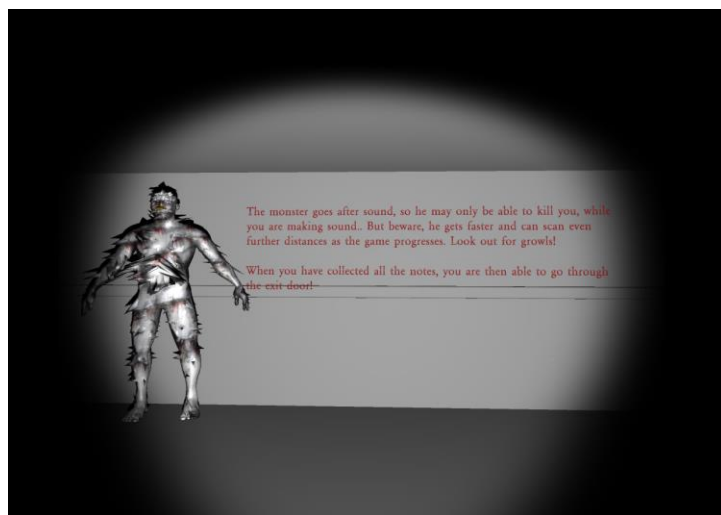
Spillet tutorial er en vigtig del af spillet, siden den fungerer som introduktion til spillet, og i sin helhed, er starten på spillet. Det ikke alle der gør brug af en tutorial, selvom man egentligt burde, og derfor er vores tutorial gjort så spændene som muligt, ved at få den til at ligne spillet i stor stil. Selve tutorialen starter ud med at man åbner en dør, og derved bliver ledt ind til introduktions "rummet". Her ser man den første "note" som er dem man skal samle op i spillet.



Figur 32 viser en note, med introduktions tekst.

Del 2:

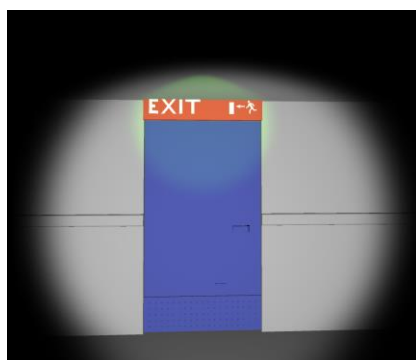
Efter at man har trykket "E" på noten, kan man begive sig ud i resten af rummet. Her lægger man nok mærke til den røde tekst på væggen og selve monsteret "AI'en". Se figur 22. Der står noget tekst markeret med rød farve, for at oplyse spilleren om at dette er vigtigt at læse. Efter man så har læst teksten, og forstået spillets sammenhæng, kan man bevæge sig hen imod EXIT døren, for ligesom af afslutte/vinde spillet.



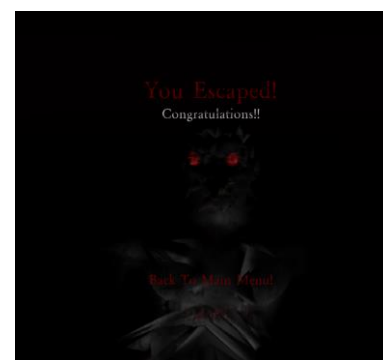
Figur 33 viser den røde tekst, som oplyser spilleren om fjenden og hvordan man vinder.

Del 3:

Som set på figur 15, ser man den dør, som fungerer som det sidste stykke vej inden spillet er slut. Hvis man så "hypotetisk set" har alle notes, kan man åbne EXIT døren, og slippe ud i live og i god behold. Derefter popper "Win" screen op, som set på figur 23, og man har derved klaret spillet. Det eneste du mangler nu, er bare at gennemføre det rigtige spil, og finde alle 9 noter i labyrinten, uden af træffe vores intelligente fjende.



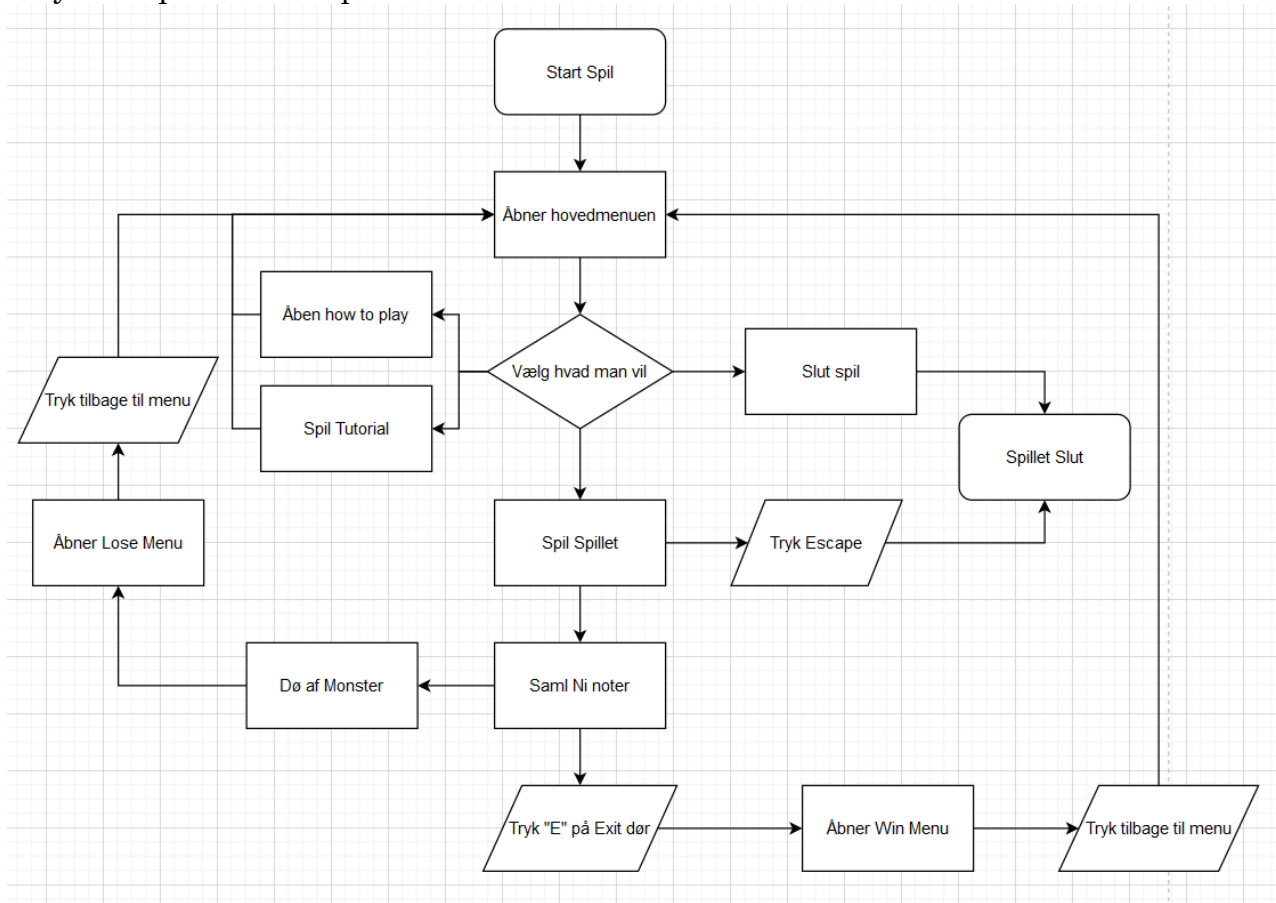
Figur 35 viser EXIT døren



Figur 34 viser "win" screen

Spillet cyklus og tilstande for AI og spiller (Flowdiagrammer og tilstandsdiagrammer)

I ²⁰flowdiagrammet betyder rektanglen med rundede kanter start eller slut, rektanglen betegner en process, diamanten betegner et valg og paralellogrammet betyder input eller output



Figur 36 flowdiagram over spillet

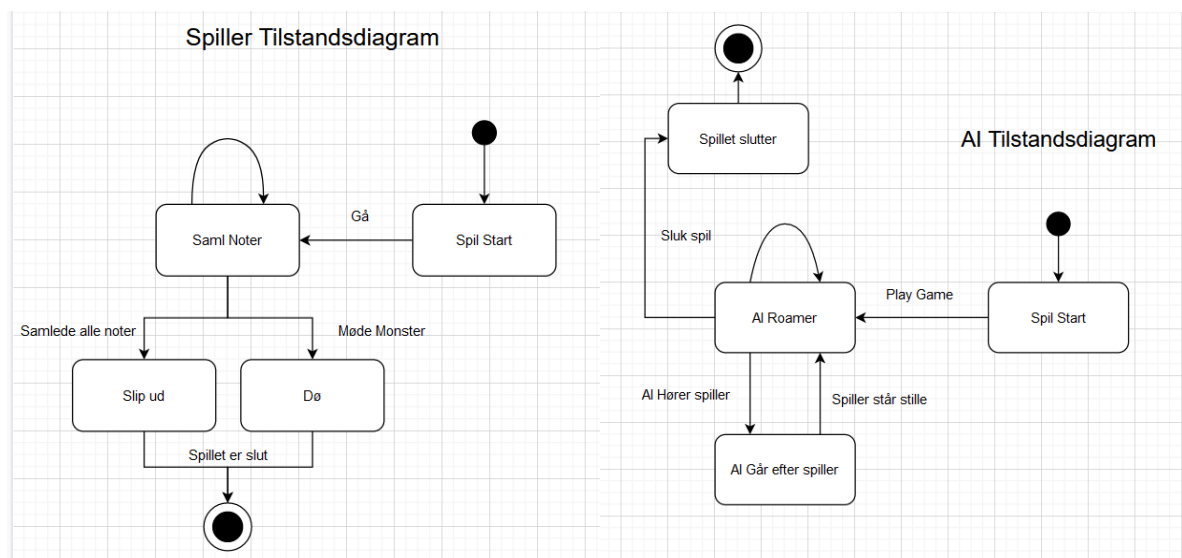
²¹Spillerens tilstandsdiagram virker på den måde at man starter spillet så går man som spiller hen for at samle noter og det gør man indtil man enten slipper ud eller bliver dræbt af monstret, derefter slutter spillet.

AI'ens tilstandsdiagram virker på den måde at spillet starter og AI'en begynder at roame imellem de waypoints som der er lagt ud på banen og det gør den indtil at spilleren kommer ind i den høre radius og genoptager roamingen når spilleren står stille og så slukker spillet så når at AI'en enten dræber spilleren eller spilleren slipper ud.

I tilstandsdiagramene betegner den sorte cirkel start og den sorte cirkel med en ring som slut.

²⁰ Lucidchart: What is a flowchart - Af skribent på Lucidchart (Artikel)

²¹ DavidLindholm.info/Books: Bogen om Java "Tilstandsdiagram" - Af Lindholm, David (E-bog)



Figur 37 tilstandsdiagram over spilleren og AI

Realisering og optimering

User-cases (AI-sprint)

Igennem projektet har vi lavet flere forskellige sprints, og user-cases til disse sprints for at holde styr på hvor langt vi var nået ved hvert enkelt sprint. Vi har indsat disse user-cases nede i bilag for hvert enkelt sprint i vores produktlog, og i denne realisering vil vi anvende vores vores AI-sprint, hvor vi vil gennem de forskellige user-cases som gør således at AI'en i produktet blev en realisering.

Som programmøre vil vi have en progressiv AI der følger efter spilleren når han er i nærheden af AI'en.

Dette er opnået ved at indsætte spilleren som "tag" navngivet player, og herefter anvendt lookRadius strukturen i Unity komponentet NavMeshAgent²² til at se om spilleren var inde for denne radius, og dette vil så få AI'en til at gå imod spilleren ved brug af SetDestanation() metoden.

- **Tidshorisont:** 8 timer

Som programmør vil jeg have en progressiv AI der kan "roame" som for den til at føltes mere levende, og opfylde kravet for at få den til at føle sig som om at den har en basis intelligent.

Dette blev opnået ved at placere "waypoints" (empty gameobjekter) som AI'en går hen til i en tilfældig rækkefølge. Dette er gjort muligt ved at tilføje et navmesh surface til

²² Docs.Unity: NavMeshAgent - Af Skribent på Unity (html)

banen som fortæller AI'en hvor han kan gå og ikke kan gå. Til dette blev vi nød til at importere et navmesh surface²³ komponent fra et andet projekt²⁴, i og med at Unity har fjernet dette komponent. Men da unity stadig understøttede surface komponentet, var det mulig for os at importere det, og dette løste også en anden problemstilling i og med at AI'en ikke kunne gå igennem væggene.

- **Tidshorison**t: 12 timer

Som programmør vil jeg have en progressiv AI der bliver sværere jo længere ind i spillet man kommer for at give spilleren en udfordring.

Dette blev opnået ved at forøge AI'ens hastighed (speed og høreradius (lookRadius) variablerne for hver note der bliver samlet op. Dette skabte også interaktion mellem spilleren og systemet.

- **Tidshorison**t: 4 timer

Som programmør vil jeg have en progressiv AI der stopper med at følge efter spilleren når han står stille for at give en strategi til spilleren for at overkomme AI'en.

Dette er blevet opnået ved at kalkulere om hvorvidt spilleren har bevæget sig ud fra sin sidste position, og hvis spilleren ikke havde gjort dette, valgte AI'en at følge sine waypoints igen.

- **Tidshorison**t: 6 timer

Samlet tid i alt: 30 timer

Test

Gennem dette afsnit, vil spillet blive testet, ud fra tre forskellige testmetoder. Disse tests er henholdsvis White Box-metoden, Black Box-metoden samt system usability scale metoden. White Box-metoden, vil blive sammenlignet med Black-box metoden, for at opnå et bedre overblik over resultaterne af de tests som nu indgår, og til sidst vil der være en vurdering ud fra testerne resultater for at se om dette opfylder de krav som blev påsat.

²³ Docs.Unity: NavMeshSurface - Af Skribent på Unity (html)

²⁴ Youtube: Unity NavMesh Tutorial Basics - Af Brackeys (Video)

White Box-test

White box-metoden er måden hvorpå testeren kan se koden og danne sig et perspektiv udefra såvel som indefra. Det er typisk software udviklere der tester gennem White Box metoden, fordi de har en anden tidligere programmerings viden.

På baggrund af white box metoden, er der opstillet et spørgeskema til spil udviklere. Hvis man tager udgangspunkt i bilag 2 ser man de spørgsmål, der er blevet sendt til en software udvikler. Personen, har efterfølgende besvaret de spørgsmål der havde relevans, og givet sin professionelle mening på dem. Nogle af spørgsmålene bliver blive inddraget, gennem denne test. For at tage udgangspunkt i nr.3 spørgsmål, som set i bilag 2, svarer udvikleren dette: ***"Ja det er nemt at finde noterne også selvom man ikke spiller tutorial så finder man efter et stykke tid ud af at noterne er ved rødt lys"*** Dette svar er yderst relevant, da det viser at udvikleren sætter pris på at objektet/målet med spillet er tydeligt og relevant. Derfor kan der nu vises at den grafiske side, har haft en valid betydning for spillet, og dens opbygning. Udover dette kan der også ses på den lidt kritiske side, fra udvikleren, da det også er vigtig at vise hvad der kan forbedres i spillet. Her ser man at udvikleren svarer følgende til spørgsmål 5: ***"Der var bugs mest med UI hvor at "Press E to use door" blinkede på skærmen men også at på et tidspunkt kan man ikke længere crouch men det havde ikke en stor betydning for spillet"*** Her ser man, at spillet har mindre fejl, såsom at man ikke kunne "crouch" altså dukke sig ned, men udvikleren nævner at det ikke havde den helt store betydning for spillet. Dette er også et vigtigt punkt, som er rigtig god respons, fordi der i dette tilfælde kræves forbedring, og yderligere udvikling af softwaren.

Der er nu gjort rede for nogle af de spørgsmål, som udvikleren gav feedback på, og disse vil blive sammenlignet med svarene fra Black-Box metoden efter.

Black Box-test

Black Box-metoden er måden hvorpå man undersøger spillet fra et ydre perspektiv, uden at kende spillets kode og dens kernefunktionalitet. Derfor er der i denne test, blevet delt et spørgeskema ud til personer, som ikke har nogen form for software erfaring. Se bilag 3 for spørgeskemaet.

For først at tage udgangspunkt i et af spørgsmålene, kan man se på følgende svar fra personen der ingen erfaring havde her i spørgsmål 2: ***"Ja, det skulle overvejes hvor man placerede sig, og hvilke noter man valgte først, hvis ikke man ville spænde ben for sig selv, når han senere fik en forøget hastighed."*** Her ser man at den uerfarne testperson indenfor software har sagt at det giver mening at forholde sig til AI og ikke bare gå random rundt. Det er her man kan lave en strategi, for at gennemføre spillet nemmere. Udover dette spørgsmål kan man også se på spørgsmål 7, som brugeren også har besvaret. Han svarer følgende: ***"Det skal måske grafisk specificeres bedre grafisk, at man befinder sig i et mentalhospital, da der det mest af alt føles som en grå gang. Dette kunne måske arbejdes bedre med, men det føles klart som et horrorpil."*** Her nævner testpersonen, at man måske kunne grafisk forbedre banen, og få det til at ligne et sindssygehospital lidt bedre. Dette feedback, er også relevant, fordi det grafiske også er en stor del af spillet, og ikke kun dets mechanics. Men udover dette,

THE HORROR OF THE MAZE

nævner han at det føles som et horrorspil, som er et af de krav, som derved også giver sig tilkende i brugerens svar her.

Sammenligning af Black Box-test & White box-test

For at se på hvad forskelle de to forskellige tests havde, kan man se på forskellen mellem udviklernes tankegang og den uerfarne brugers tankegang. Dette gør at vi kan specificere vores produkt endnu mere, og sørge for at den også rammer den rigtige målgruppe. For eksempel, kan man se rent sprogligt at udvikleren kender til fænomenet "UI" som set i svaret til spørgsmål 5 under White Box-testen. UI eller User Interface, er måden hvorpå man opfatter noget helt grafisk og visuelt. Derfor kan vi godt regne med at udvikleren selv har kendskab til området, og spillets form. Derfor er dernedunder opstillet et skema, over hvordan vi syntes de forskellige spørgsmål er besvaret, og om vi er enige med dem. Se bilag 2 & 3 for de resterende spørgsmål og svar.

SUS (System Usability Scale)

Derved bliver der nu opstillet fra forskellige (SUS) skemaer for hver person der har besvaret skemaet. Den første kan du se nedenunder, de resterende 2 SUS findes i bilag 5.

SUS 1. Se spørgsmålene fra (1-10) som blev stillet i bilag 4.

SUS 1	Meget uenig	Uenig	Neutral	Enig	Meget enig	Beregning	Point
Spg.1	✓					1-1	0
Spg.2	✓					5-1	4
Spg.3					✓	5-1	4
Spg.4	✓					5-1	4
Spg.5				✓		4-1	3
Spg.6	✓					5-1	4
Spg.7					✓	5-1	4
Spg.8	✓					5-1	4
Spg.9					✓	5-1	4
Spg.10					✓	5-5	0

Derved regner vi point sammen, og ganger det med 2,5

$$Point_{SUS_1} = 0 + 4 + 4 + 4 + 3 + 4 + 4 + 4 + 4 + 0 = 31$$

$$Point_{total} = 31 \cdot 2,5 = 77,5$$

Derved har vores første SUS et total på 77,5 ud af 100 maksimal point. Dette er meget positivt og tilstrækkeligt godt. På baggrund af dette, kan vi også se at der er nogle punkter/spørgsmål som kræver forbedring senere hen.

Vurdering af testene

Overordnet set, har vores test på spillet, været tilstrækkeligt gode. Vi har fået en masse feedback, som der kan bruges i stor grad. Både i forhold til problemstillingen, men også efter de behov som spørgeskemaets besvarelser har. Vi kan vurdere ud fra disse testresultater hvorledes hvor succesfuldt vores oprindelige mål set i forhold til både problemformulering og kravene, i og med at whitebox og blackbox spørgsmålene var specifikt til input fra brugerne om hvorvidt kravene blev overholdt, og SUS testning kan anvendes til at vurdere hvor succesfuldt systemet var.

En af de største kritikpunkter, som var angivet til produktet var dens bugs. Det ser man på baggrund af de svar, som findes nede i bilag 2 og 3, hvori at testpersonerne gjorde det klart at spillet havde nogle mindre væsentlige bugs, men dette var dog en af vores blødere krav og det var ikke spilødelæggende, da vores SUS test viser at systemet ifølge vores testere fungerede udmærket. Dette vil sige, at der i spillet er plads til forbedringer og rettelser af eventuelle småfejl som i bugs, men at det ikke påvirkede systemet på en måde hvor på at det var altødelæggende.

Derudover, ser det ud som om at der ikke er de store kommentar, både fra udvikleren eller den uerfarne testperson, om spillets sammenhæng, og om hvorvidt målet i spillet var klar. Dette opfylder vores krav i forhold til at gøre affordances, signifiers og feedback tydelige for brugerne og objektet i spillet har ifølge testpersonerne været klare.

Det viser sig også at ifølge de System Usability Scales test som vi har, at vi har forholdsvis høje point, og at det selvfølgelig ikke er i top, men på rette kurs i stedet for. Dette viser at systemet generelt fungerede, og da systemet baserede sig ud fra en progressiv AI som prøvede at fange spilleren, må dette betyde at systemet fuldførte dette mål i at få det til at føle som om at AI'en blev progressivt sværere. Hvis der skulle arbejdes på projektet i forhold til den intelligent fjende, kunne det være at ikke bare at gøre hans variabler større, men eventuelt også gøre ham mere intelligent og måske kunne få ham til at udvikle sig mere gennem spillets forløb hvis dette var et fuldt spil. Hertil kunne sværhedsgraden også stiges, hvis man som vi proklamerede med at kunne tilføje flere fjender i løbet af spillet hvis man skulle gøre spillet mere omfattende.

Et andet vurderingspunkt som der var ud fra kravene, var at spillets tema og måden hvorpå at spillets føltes som om skulle være klare og tydelige for spilleren. I vores test er der blevet fortalt til os, at spillets tema var klar i forhold til måden menuen og tutorialen er sat op på, men når spilleren selve er inde i spillet, kan grafikken føltes lidt lackluster. Følelsen af at det er en horror er der ifølge vores testpersoner, men hvis det skulle være mere uddybende, kunne der ifølge de testpersoner vi har afprøvet det på være mere detaljeret grafik og gå dybere ind i temaet "mentalsygehus". En af de mest væsentlige krav har også været om hvorvidt om spilleren følte at måden hvorpå at de havde ejerskab om hvor svært spillet progressivt har været er lykkedes, i

og med at testpersonerne har givet udtryk for, at dette krav lykkedes, og de følte at jo mere de kendte til spillet og jo længere fremad i progressionen af spillet de var, følte de også at det passede mere til deres niveau. Dette gives både udtryk i, i white og black box testene og også i SUS-testene, hvor at spillerne følte at de havde styr på systemets virkemåde og hvorledes dette fungerede.

Der er nogle af spørgsmålene, som ikke har den bedste respons ift. SUS, men dette betyder også bare at der kræves forbedring, på spillet og hver enkelt mands individuelle arbejde. Alt i alt, har vi opnået et bredere forståelse for vores spil, på grund af at vi har fået andre til at teste spillet, og givet deres mening på det vi arbejder med. De svar vi har fået, har vi taget til os, og prøvet at forbedre i så stor grad som overhovedet muligt. Hertil har motivationsfaktoren i spillet for spillerne også været væsentlig, og ifølge de SUS tests at vi har kunne få forbedringer i forhold til om spillerne vil anvende systemet i en længere periode og vil været motiveret til at spille det. Dertil er dette dog også noget der hører til horror-genren, da man tit ikke vil genspille et horror-spil 2 gange. Generelt kan det vurderes, at de mest hårde krav er ud fra testene udfyldt og at vi har kunnet fremstille et spil, hvori at systemet interagere med spilleren på en måde som er generelt opfyldende for spillerne som spiller spillet, og at den intelligente fjende virker på en måde hvorpå at den progressivt bliver sværere gennem spillets forløb, og denne progression følger spillerens evner i og med at AI'en kun bliver sværere ud fra en aktion fra spilleren.

Konklusion:

Vi kan ud fra vores analytiske metoder konkluderer, at vores ønskelige løsning til vores produkt var et horrorspil baseret på en intelligent AI, som gennem spillet blev progressivt sværere. Dette blev fundet ud af, ved hjælp af at fremstille en problemanalyse, hvori vi anvendte faglig viden og egen analytiske data som vi frembragte til hovedmålgruppen til vores projekt (gamers på vores alder).

Vha. af disse analytiske metoder, kunne vi opstille hårde og bløde krav som tilpassede den projektafgrænsning som vi formulerede i vores projektafgrænsningen. Heri konkluderede vi bl.a. at funktionaliteten i spillet skulle fylde mere end de grafiske elementer, og hvorledes AI'en skulle interagere med spilleren. Kilderne vi anvendte til at kunne opstille disse hårde og bløde krav, var bl.a. vores spørgeskema som havde 31 respondenter. Troværdigheden i dette spørgeskema er ikke optimal, da der var en forholdsvis lille svarbase på kun 31 respondenter, og dette kan man ikke konkludere det specifikke ud fra. Dog kunne tendenserne i analysen støttes op af andre mere troværdige kilder, hvori vi anvendte kilder fra bl.a. Amnesia's skaber, vi anvendte et studie fra Maastricht universitet og forskellige troværdige kilder såsom geeksforgeeks.com og lignende Tech hjemmesider.

Udover disse kilder kan det konkluderes at kilderne i rapporten generelt været troværdige, i og med at kilderne vi anvendt til at kunne finde ud af hvordan navmesh komponenterne virkede, var baseret på hjemmesider som Unity ejede.

Vi kan konkludere ud fra den arbejdsmetode (SCRUM) som vi har valgt, at vi kunne nå alting som vi satte os for i vores releasebacklog. Timerne som blev anvendt til de enkelte sprints blev dog varieret gennem selve produktionen af dem, da noget af sprintsne tog kortere tid end andre sprints. Det kan også konkluderes at dette var en hensigtsmæssig måde at gå frem på, i og med at alle sprints i releasebacklog blev udført.

Vi kan ud fra vores test konkludere, at vores fastlagte mål blev nået med enkelte afvigelser. Vores mål med at skabe en AI som fulgte efter en spiller og dette blev gjort på lyden. Afvigelsen heri lå måden hvorpå vi gjorde det. I stedet for at lave en AI der rent teknisk reagerer på lyd, lavede vi en AI som har en lookRadius, som bliver progressivt større og således at monsteret ikke kan reagere på en spiller, hvis den ikke laver lyd (eller rettere står stille). Vi kan konkludere heri, at vi opnåede målet med AI'en, dog på andre måder end tidligere tænkt. Vi kan også ud fra vores testning konkludere, at det generelle system med vores intelligente AI fungerede, i og med at der ud fra vurderingen i testning kan konkluderes at kravene om at AI'en bliver progressivt sværere for spilleren og at systemet interagerer med spilleren er fuldført. Dette skabte også en form for "Meaningful play", da spillets aktioner har betydning for hvordan systemet reagerer på dette og systemet reagerer på spillerens aktioner, og at dette fuldfører også kravet med at spillets aktioner skal have en betydning for hvorledes systemet reagerer.

Dertil kan det også konkluderes, at signifiers, affordances og feedback også fungerer optimalt, da alle vores testpersoner erklærede at systemet var let at finde rundt i og var brugbart. Dog kan det som en nedprioritering også konkluderes, at det temaet i spillet ikke var optimalt sat ind i temaet "mental hospital" ifølge vores testresultater. Dette blev dog også fra starten af i kravene angivet, at dette var en nedprioritering i forhold til selve funktionaliteten i spillet og den progressive AI.

De fastlagte test som vi har anvendt kan konkluderes til at være en succes, i og med at vi kunne anvende testene til at vurdere vores spil ud fra de mest væsentlige krav til hvorledes vores produkt opfyldte de mål som vi fik påsat i både problemformuleringen, problemafgrænsningen og krav. Det kan også konkluderes, at testene viste os at der var visse ting i vores system (produkt) som kunne forbedres hvis dette skulle videreudvikles. Det kan også konkluderes at resultaterne fra testene kunne vurderes til at være troværdige, i og med at spillet blev testet af både mennesker uden kendskab til softwaren og med kendskab til softwaren hvori der så kunne fås en sammenflydende sammenligning. Dertil blev systemets brugbar ved hjælp af SUS også testet af endnu flere personer uden kendskab til softwaren.

Det kan også konkluderes at denne rapport kan anvendes til at få et indgående indblik i hvordan vores produkt (spil) blev sat sammen og hvorledes tankeprocessen bag selve opbygningen af spillet fremgik.

Til slut kan det konkluderes, at spillet var en succes med visse mén så som at der skal være en større afklaring med temaet og at der ud fra testvurderingerne kan finpudses i måden hvor på den intelligente fjende opfører sig på.

Evaluerings

Dette projekt kan evalueres til at være en succes, i mente med at vi har fuldført det oprindelige mål som blev afsat i projektafgrænsingen og i problemformuleringen. Det gik rigtig godt med at kunne fuldføre alle de store mål vi havde om spillet, som var at give det en illusion at enemy AI'en følger spilleren efter lyd, at kunne opretholde et spændene miljø som spilleren kan bevæge sig rundt af og at skabe en form for "meaningful play" hvor at spillerens aktioner påvirkede spillet og hvor systemet påvirkede måden spillerens spillestil blev udformet på.

Der var dog nogle enkelte skavanker i forhold til at kunne få produktet til at fuldføre temaet om et "mental hospital" helt. Dette var et resultat af at funktionerne i spillet blev vægtet mere end selve temaet, da vores mainfokus var den fjendtlige AI og derfor blev nogle af grupperessorucerne re lokaliseret af den grund. Tidsplanen blev overholdt i den grad, at alle sprintsne som var i backloggen blev fuldført på den ene eller anden måde.

Litteraturliste:

Med forfatter

Blackburn, Megan. (2019, 13. Feb). *Burndown Chart: What is it & How do I use it?*. ProjectManager. <https://www.projectmanager.com/blog/burndown-chart-what-is-it> besøgt d. 20/02/2020

Dead, Eyemcduck. (2017, 26. Februar). *Storytelling and interactivity in gaming video*. The Artifice. <https://the-artifice.com/video-gaming-story-telling-interactivity/> besøgt d. 25/04/2020

Emmons, Dan. (2014, 5. Juni). *Design Club - Super Mario Bros: Level 1-1 - How Super Mario Mastered Level Design* [Video]. Youtube. <https://www.youtube.com/watch?v=ZH2wGpEZVgE> besøgt d. 19/04/2020

Grant, Eugene. & Lardner, Rex. (1952, 26. July). IT. *The Newyorker*. <https://www.newyorker.com/magazine/1952/08/02/it> besøgt d. 19/02/2020

Gibbs, Samuel. (2017, 7. December). *AlphaZero AI beats champion chess after teaching itself in four hours*. Guardian. <https://www.theguardian.com/technology/2017/dec/07/alphazero-google-deepmind-ai-beats-champion-program-teaching-itself-to-play-four-hours> besøgt d. 19/02/2020

Hauser, Larry. (2019). *Artificial Intelligence*. Encyclopedia of philosophy. <https://www.iep.utm.edu/art-intel/> besøgt d. 22/02/2020

Jain, Mahak. (s.d.). *Differences between Black Box testing and White Box testing*. GeeksforGeeks. <https://www.geeksforgeeks.org/differences-between-black-box-testing-vs-white-box-testing/> besøgt d. 21/02/2020

Leigh, Alexander. (2008, 29. September). *Does Survival Horror Really Still Exist*. Kotaku. <https://kotaku.com/does-survival-horror-really-still-exist-5056008> 21/04/2020

Lindholm, David. (2019). *Java Bogen: Tilstandsdiagram*. Creative Commons. <http://www.davidlindholm.info/Books> 26/02/2020

Maass, Laura E shummon . (2019, 1. Juli). *AI in video games*. TowardsDataScience. <https://towardsdatascience.com/artificial-intelligence-in-video-games-3e2566d59c22> besøgt d. 22/02/2020

Porter, Matt. (2014, 20. Oktober). *The Do's and Don'ts of Horror Game-making*. Vice. <https://www.vice.com/da/article/gq83dm/the-dos-and-donts-of-horror-game-making-323> 20/02/2020

Reece, Damon. (2015, 27. April). *Best Tower Defense Games of all time*. Gameranx. <https://gameranx.com/features/id/13529/article/best-tower-defense-games/> 14/02/2020

Reeves, Ben. (25, 2014. September). *Do gamers really like hard games*. Gameinformer. <https://www.gameinformer.com/b/features/archive/2014/09/25/why-we-like-hard-games.aspx> 07/02/2020

Rouse, Margaret. (s.d.). *3D Gaming*. TechTarget. <https://whatis.techtarget.com/definition/3D-gaming> besøgt d. 22/02/2020

Salen, Katie. & Zimmerman, Eric. (2003). *Rules of play*. MIT PRESS. besøgt d. 23/04/2020

Spronck, Pieter., Sprinkhuizen-Kuyper, Ida. & Postma, Eric. (s.d.). *Difficulty scaling of game AI*. Spronck. <https://spronck.net/pubs/SpronckGAMEON2004.pdf> 02/04/2020

Thirslund, Asbjørn. (2018, 14. Marts). *Unity NavMesh Tutorial Basics* [Video]. Youtube. <https://www.youtube.com/watch?v=CHV1ymlw-P8&feature=youtu.be> 12/04/2020

Thorø, Valle. (2020). *Teknologi Håndbogen*. http://www.vthoroe.dk/Teknologi/Projektarbejde/Teknologi_haandbog.pdf 18/04/2020

Uden forfatter

Agile Glossary. (s.d.). Agilealliance. <https://www.agilealliance.org/agile101/agile-glossary/> besøgt d. 22/02/2020

NavMashSurface. (2020, 22. April). Docs.Unity. <https://docs.unity3d.com/ScriptReference/AI.NavMeshAgent.html> 18/04/2020

NavMesh Surface. (2017, 14. September). Dosc.Unity. <https://docs.unity3d.com/Manual/class-NavMeshSurface.html> 18/04/2020

System Usability Scale (SUS). (s.d.). Usability. <https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html> besøgt d. 22/02/2020

What is a Flowchart. (s.d.). Lucidchart. <https://www.lucidchart.com/pages/what-is-a-flowchart-tutorial> 18/04/2020

What is Scrum. (s.d.). Agilealliance. <https://www.agilealliance.org/glossary/scrum/> besøgt d. 22/02/2020

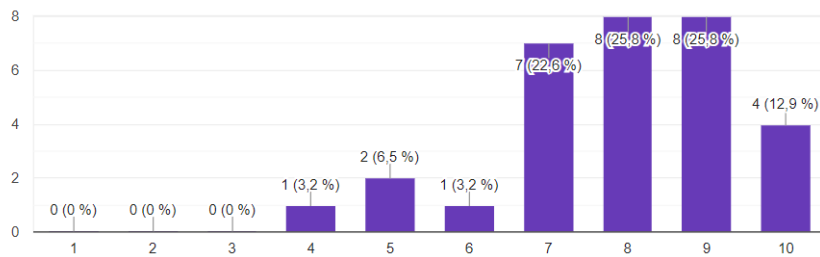
THE HORROR OF THE MAZE

Bilag:

Bilag 1, spørgeskema:

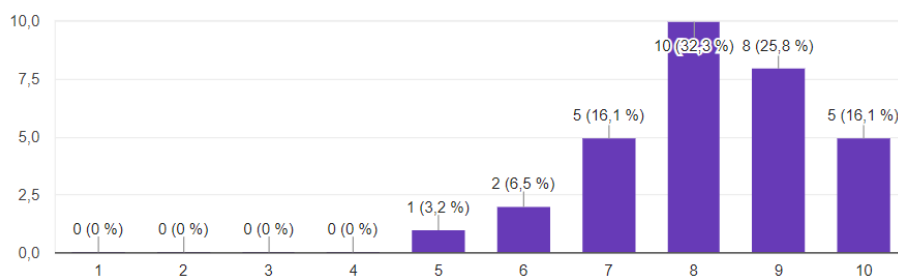
Er det vigtigt for spillets udvikling, at fjenden(erne) i spillet udvikler gennem spillets progression?

31 svar



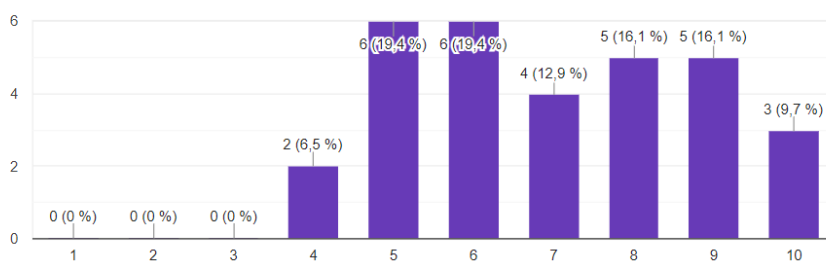
Hvor vigtigt er funktionaliten i et spil?

31 svar



Hvor svær skal en progressiv fjende (fjende der bliver sværere) være fra starten af. (1 er let, 5 er dit niveau, 10 er hardcore fra starten)

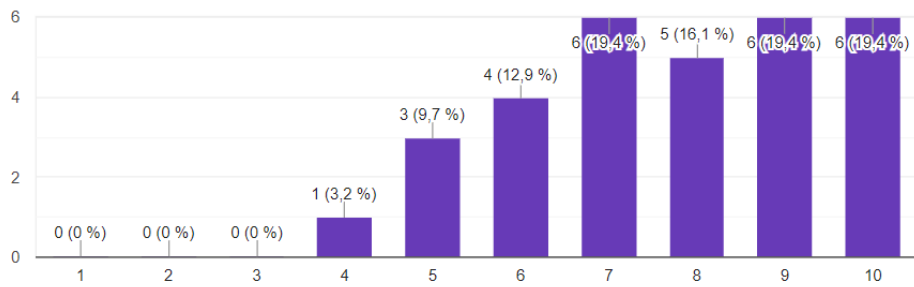
31 svar



THE HORROR OF THE MAZE

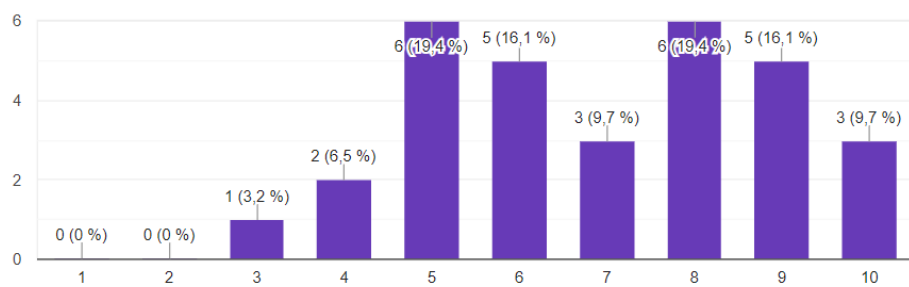
Hvor vigtigt er en tutorial i et spil?

31 svar



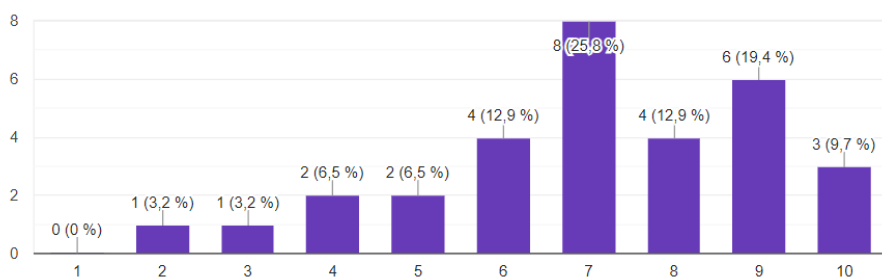
Hvor irriterende er (mindre) bugs i et spil? (1 er ikke irriterende overhovedet, 10 er SUPER irriterende)

31 svar



Hvor kompleks skal en bane være?

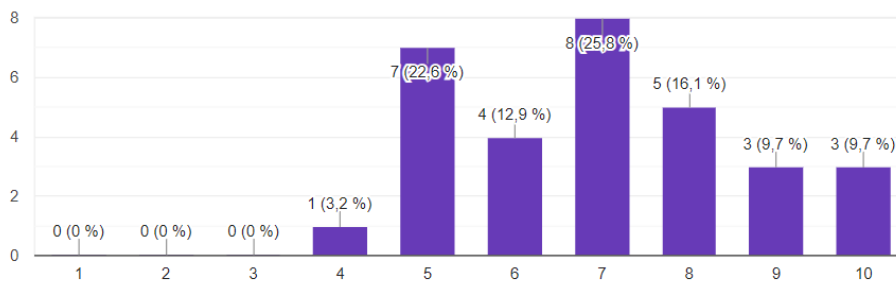
31 svar



THE HORROR OF THE MAZE

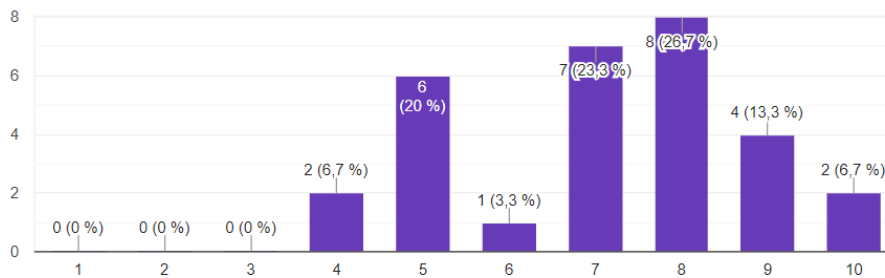
Hvor vigtigt er det, at alle elementer i spillet har af betydning for gameplay?

31 svar



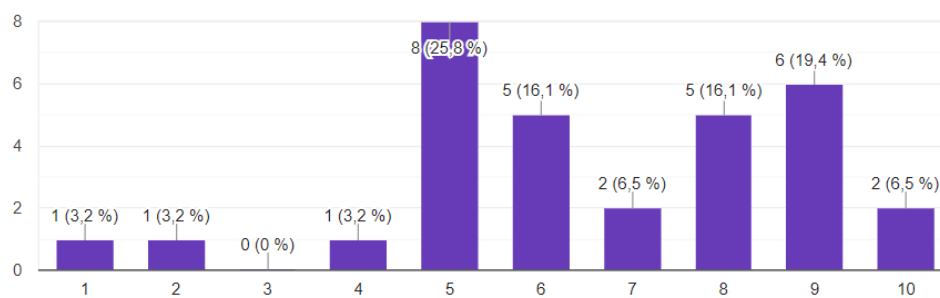
Hvor svær skal en progressiv fjende BLIVE (samme skala som det forrige spørgsmål)

30 svar



Hvor vigtigt er det, at spillets grafik har sammenhæng med spillets tema?

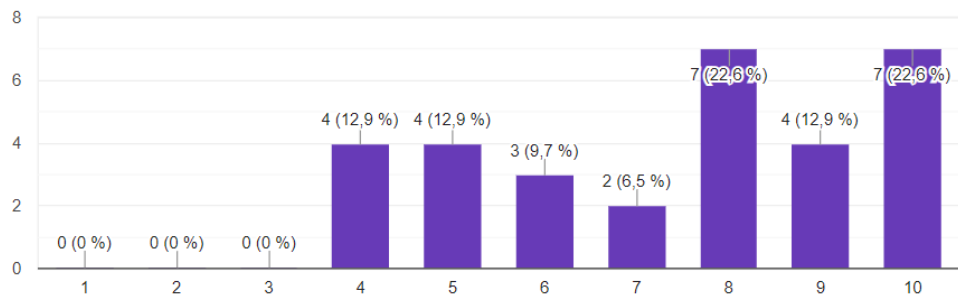
31 svar



THE HORROR OF THE MAZE

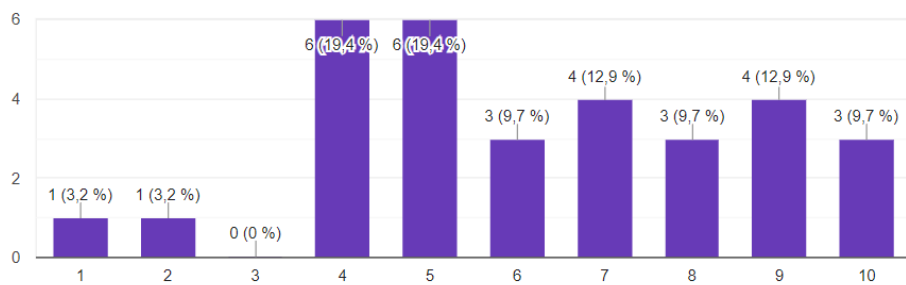
Er det vigtigt for dig at vide, hvad målet i et spil er?

31 svar



Hvor vigtigt er grafikken i et spil?

31 svar



Bilag 2, White Box spørgsmål & svar

Spørgsmål 1:

Har du det indtryk, at fjenden i spillet bliver sværere og mere intelligent som spillet progressivt udvikler sig?

Ja det gør den

Spørgsmål 2:

Føler du, at dine aktioner i spillet havde betydning for hvordan den intelligente fjende progressivt udviklede sig?

Ja føler den bliver hurtigere og kan høre længere

Spørgsmål 3:

Var det ved hjælp af tutorial og de forskellige grafiske elementer i spillet (signifiers og affordances) nemt at finde objektet/målet i spillet, eller burde disse have været mere tydelige?

Ja det er nemt at finde noterne også selvom man ikke spiller tutorial så finder man efter et stykke tid ud af at noterne er ved rødt lys

Spørgsmålet 4:

Var spillet motiverende at spille, og blev det mere spændene gennem spillets forløb?

Ja spillet blev mere fastpaced

Spørgsmål 5:

Hvor mange bugs var der i spillet, og føler du de havde special betydning for spilleoplevelsen?

Der var bugs mest med UI hvor at "Press E to use door" blinkede på skærmen men også at på et tidspunkt kan man ikke længere crouch men det havde ikke en stor betydning for spillet

Spørgsmål 6:

Var banen nem og lære, og gjorde de grafiske elementer det lettere at finde rundt i banen?

Ja efter man havde spillet den lidt så finder man ud af hvor forskellige veje fører hen

Spørgsmål 7:

Føler du at spillet føles som et horrorspil, og efterlever temaet at være siddet fast i et mentalt hospital?

Ja da der er meget tomt er det et meget skræmmende miljø og de mange rum kunne godt indikere noget der ville ligne et sindssygt hospital

Bilag 3, Black Box spørgsmål & svar

Spørgsmål 1:

Har du det indtryk, at fjenden i spillet bliver sværere og mere intelligent som spillet progressivt udvikler sig?

Han bliver hurtigere og får længere range, men om han bliver mere intelligent, er en smule diskutabel. Han bliver ofte stuck, når hans hastighed bliver for høj. Sværhedsgraden bliver dog højere, som hans range og hastighed stiger.

Spørgsmål 2:

Føler du, at dine aktioner i spillet havde betydning for hvordan den intelligente fjende progressivt udviklede sig?

Ja, det skulle overvejes hvor man placerede sig, og hvilke noter man valgte først, hvis ikke man ville spænde ben for sig selv, når han senere fik en forøget hastighed.

Spørgsmål 3:

Var det ved hjælp af tutorial og de forskellige grafiske elementer i spillet (signifiers og affordances) nemt at finde objektet/målet i spillet, eller burde disse have været mere tydelige?

Ja, det var nemt og lige til.

Spørgsmålet 4:

Var spillet motiverende at spille, og blev det mere spændene gennem spillets forløb?

Det spændende, og der var en grad af spænding som steg lineært, som spillets forløb udviklede sig.

Spørgsmål 5:

Hvor mange bugs var der i spillet, og føler du de havde special betydning for spilleoplevelsen?

Der var en bug mht. monsterets bevægelse, eftersom han blev stuck på flere tidspunkter, hvor han "campede" en, og gik i streger omkring en. Hithboxes for døre skal derudover ordnes, da den kun kan åbnes på en meget specifik overflade, og dette er en smule væsentligt.

Spørgsmål 6:

Var banen nem og lære, og gjorde de grafiske elementer det lettere at finde rundt i banen?

Ja, banen var rimelig nem at forstå, og lamperne gjorde det nemt at finde ens noter.

Spørgsmål 7:

Føler du at spillet føles som et horrorspil, og efterlever temaet at være siddet fast i et mentalt hospital?

Det skal måske grafisk specificeres bedre grafisk, at man befinder sig i et mentalhospital, da der det mest af alt føles som en grå gang. Dette kunne måske arbejdes bedre med, men det føles klart som et horrorspil.

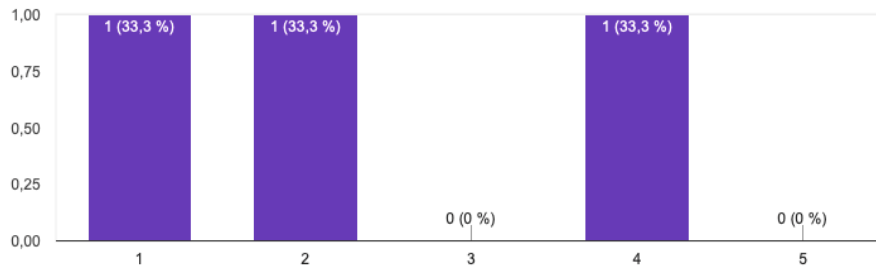
Bilag 4, SUS spørgsmål & svar

Spørgsmål 1:

Jeg tror at jeg ville bruge system ofte



3 svar

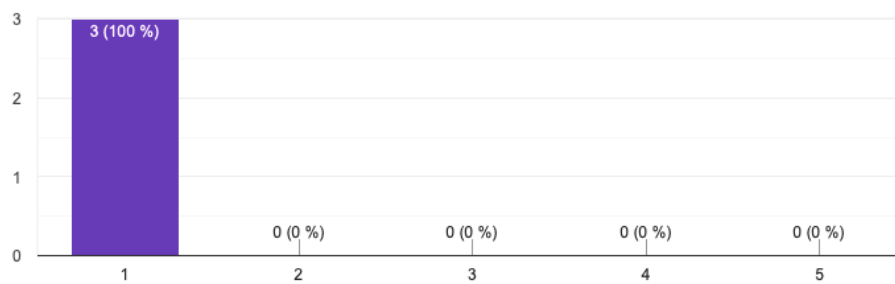


Spørgsmål 2:

Jeg synes, at systemet er unødvendigt komplekst



3 svar

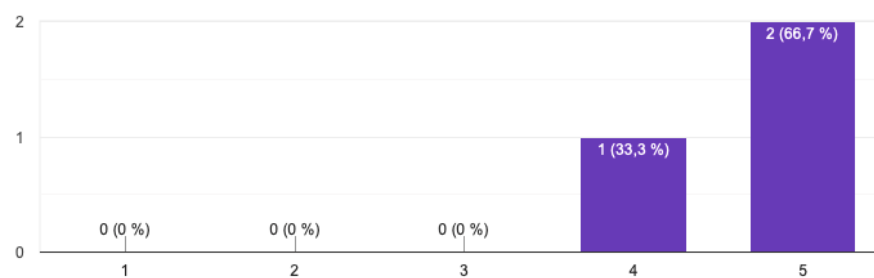


Spørgsmål 3:

Jeg synes, systemet er nemt at bruge



3 svar



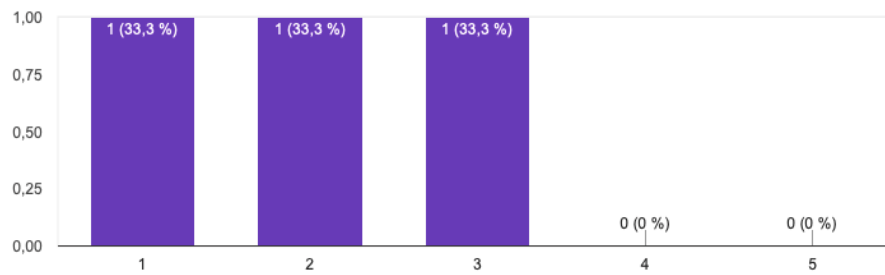
Spørgsmål 4:

THE HORROR OF THE MAZE

Jeg tror, at jeg har brug for hjælp af en ekspert til at bruge systemet



3 svar

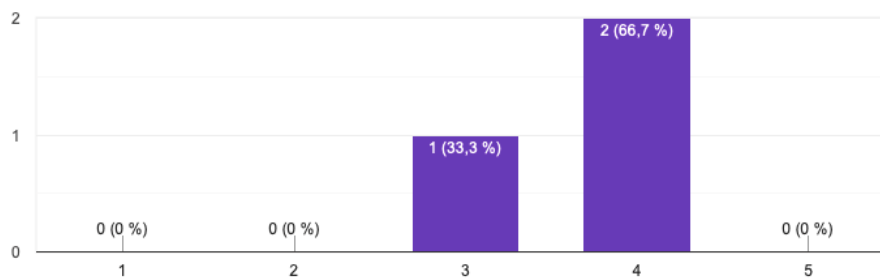


Spørgsmål 5:

Jeg synes, de forskellige dele af systemet er godt integrerede



3 svar

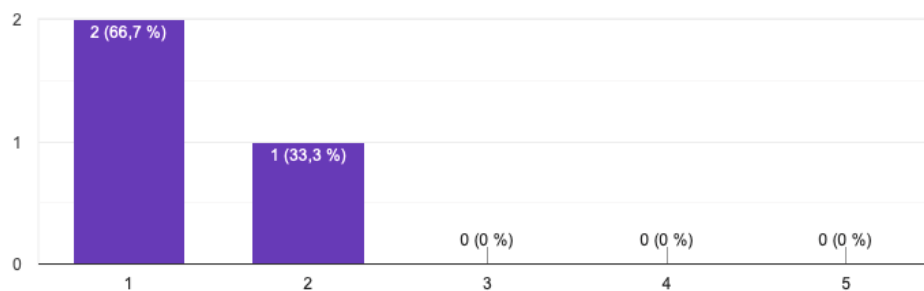


Spørgsmål 6:

Jeg synes, der er for mange ting, der er usammenhængende i systemet



3 svar



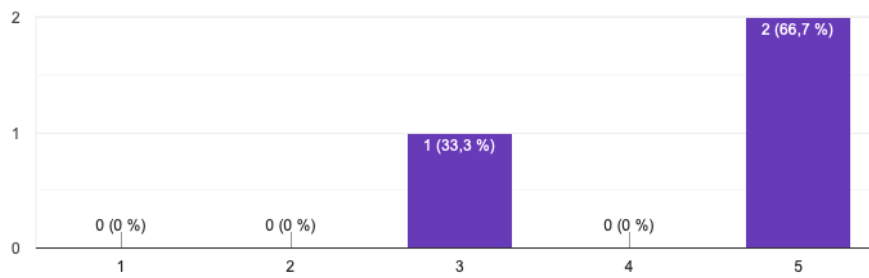
Spørgsmål 7:

THE HORROR OF THE MAZE

Jeg vil tro, at de fleste mennesker nemt ville lære at bruge systemet



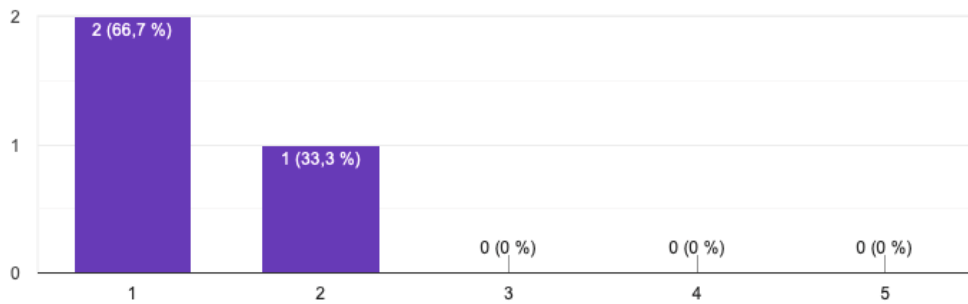
3 svar



Spørgsmål 8:

Jeg synes, at systemet er besværligt at bruge

3 svar

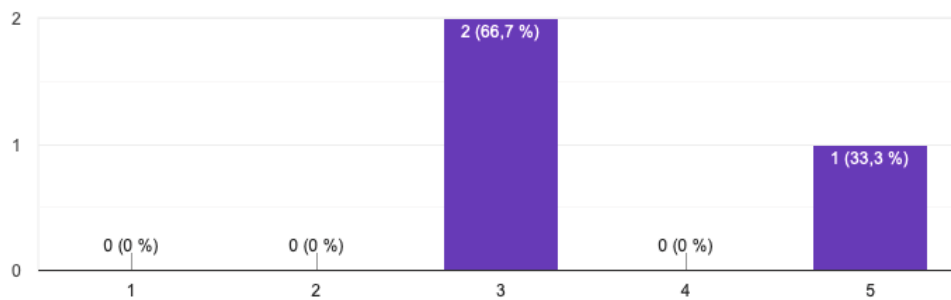


Spørgsmål 9:

Jeg synes, at jeg er fortrolighed med systemets virkemåde



3 svar



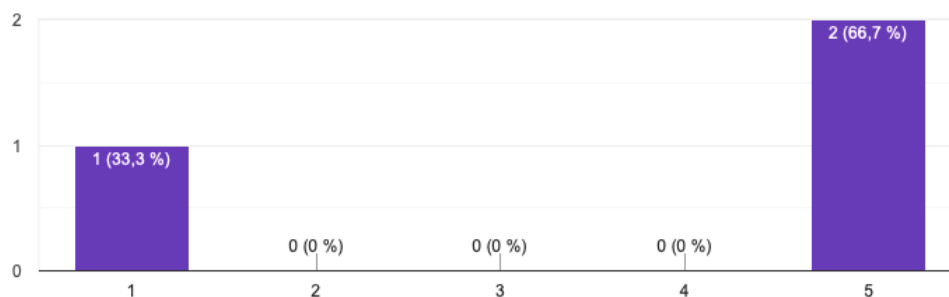
Spørgsmål 10:

THE HORROR OF THE MAZE

Der var meget at lære, før jeg kunne komme i gang med at bruge systemet



3 svar



Bilag 5, SUS tabel 2 og 3.

SUS 2	Meget uenig	Uenig	Neutral	Enig	Meget enig	Beregning	Point
Spg.1				✓		4-1	3
Spg.2	✓					5-1	4
Spg.3					✓	5-1	4
Spg.4		✓				5-2	3
Spg.5				✓		4-1	3
Spg.6		✓				5-2	3
Spg.7					✓	5-1	4
Spg.8	✓					5-1	4
Spg.9			✓			3-1	2
Spg.10					✓	5-5	0

Derved regner vi point sammen, og ganger det med 2,5

$$Point_{SUS_2} = 3 + 4 + 4 + 3 + 3 + 3 + 4 + 4 + 2 + 0 = 30$$

$$Point_{total} = 30 \cdot 2,5 = 75$$

SUS 3	Meget uenig	Uenig	Neutral	Enig	Meget enig	Beregning	Point
Spg.1		✓				2-1	1
Spg.2	✓					5-1	4
Spg.3				✓		4-1	3
Spg.4			✓			5-3	2
Spg.5			✓			3-1	2
Spg.6	✓					5-1	4
Spg.7			✓			3-1	2
Spg.8		✓				5-2	3
Spg.9			✓			3-1	2
Spg.10	✓					5-1	4

Derved regner vi point sammen, og ganger det med 2,5

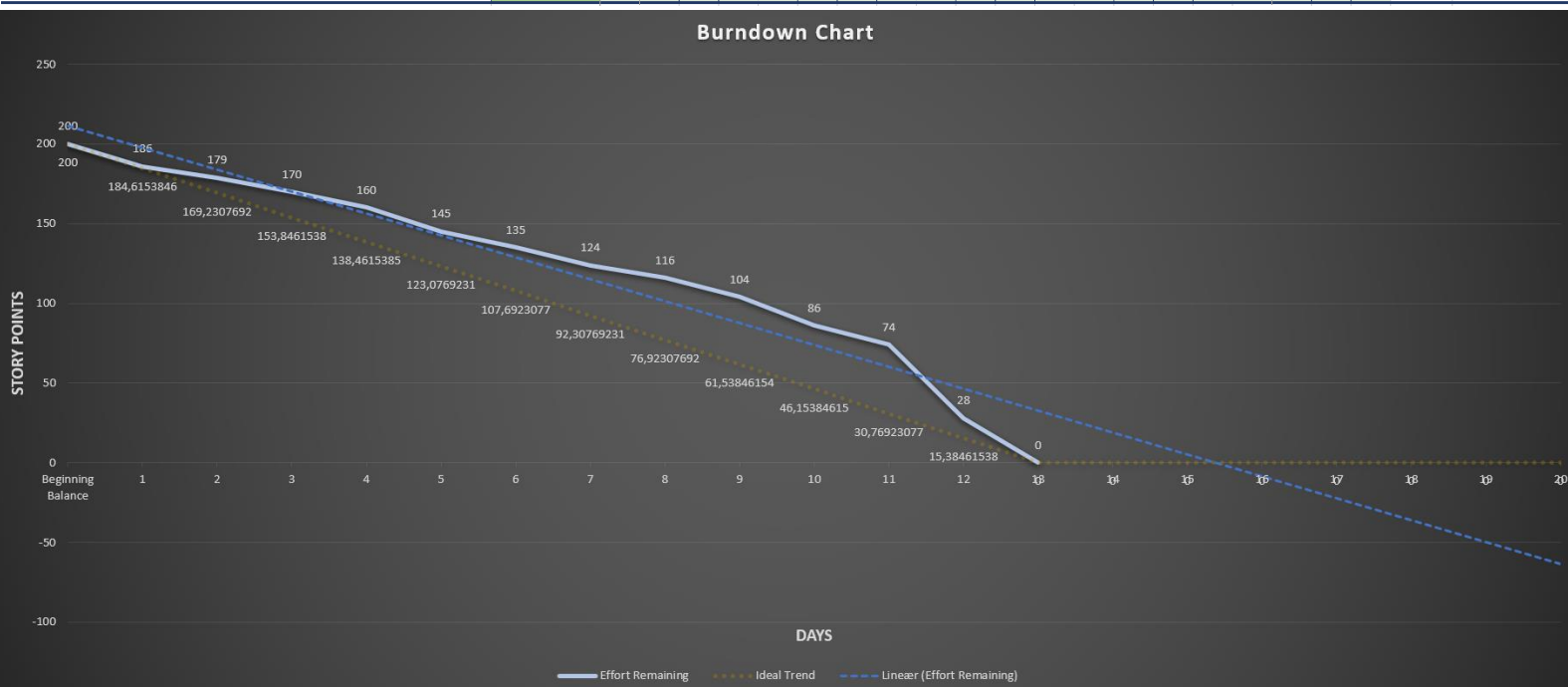
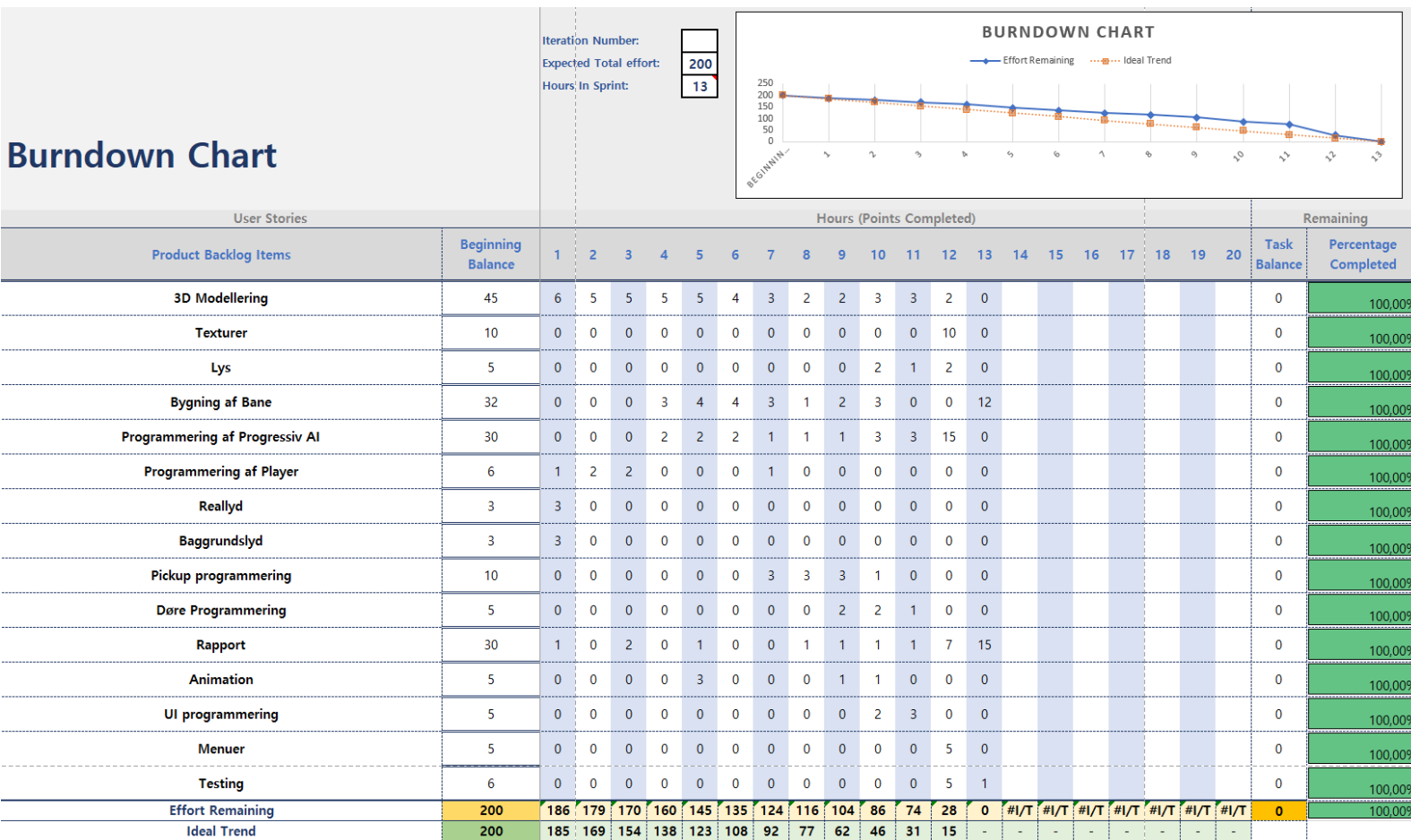
$$Point_{SUS_3} = 1 + 4 + 3 + 2 + 2 + 4 + 2 + 3 + 2 = 23$$

$$Point_{total} = 23 \cdot 2,5 = 57,5$$

THE HORROR OF THE MAZE

Bilag 6, BurndownChart

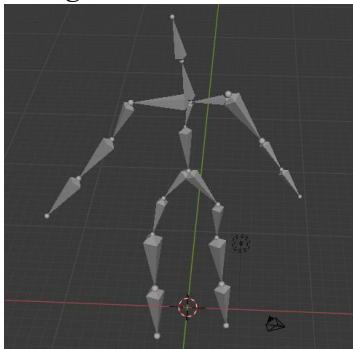
Burndown Chart



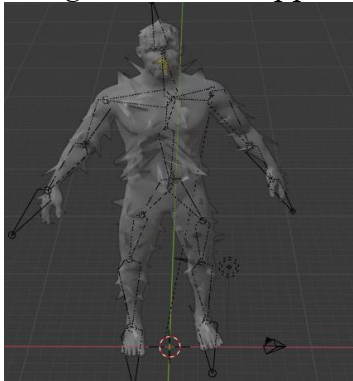
THE HORROR OF THE MAZE

Bilag 7, AI Struktur, knogler og andre features.

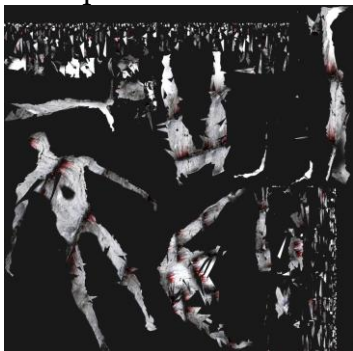
Knogleskelet:



Knogleskelet i kroppen:



Skin/patteren af AI:

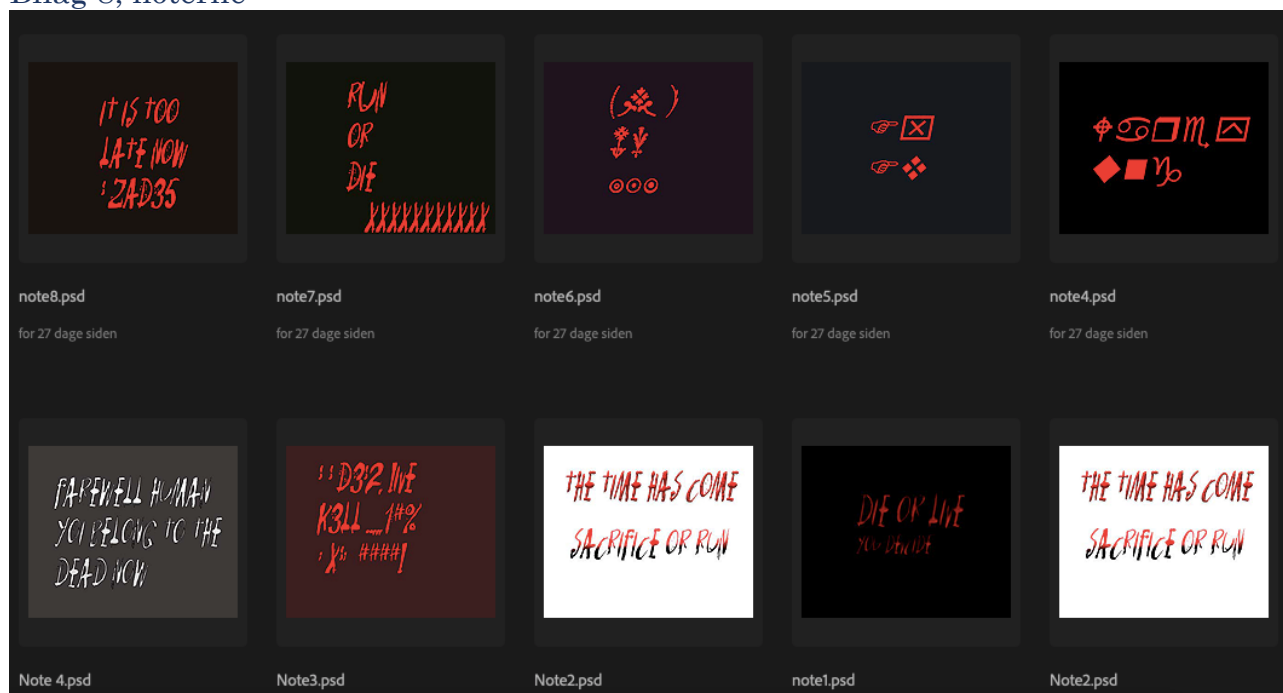


Front stilling:



THE HORROR OF THE MAZE

Bilag 8, noterne



Bilag 9, Releasebacklog + produktbacklog

3D-modellering sprint + textures sprint - 55timer

Plader til gulv og tag: 5 timer

Gulv/tag	Vi har lavet ni plader til spillet, som skulle bruges til tag og gulv. Disse plader er lavet i "Autodesk Fusion". Der var mange problemer undervejs, da der ikke kunne laves to forskellige appearances (To forskellige materialer). Dette blev løst ved at lave alle pladerne den samme farve, hvilket er derfor taget og gulvet er hvidt i vores spil.
----------	--

Vægge til banen: 6 timer

Vægge	Banen består af forskellige komponenter en af disse er en væg. Væggen er lavet i "Autodesk Fusion 360". Der var problemer med teksturerne, hvor at dele af væggene forsvandt. Dette blev løst ved at dele væggen op i tre segmenter og tildele et materiale til hvert segment.
-------	--

THE HORROR OF THE MAZE

Døre til banen: 5 timer

Døre	Døre er lavet i "Autodesk fusion 360". Døre havde samme problem, hvor materialerne ikke kunne mixes. Derfor var det nemmeste fix at give døren det samme materiale.
------	---

Lamperne: 5 timer

Lamperne	Lamperne er lavet i "Autodesk Fusion 360". Der var problemer med at eksportere filen, hvis lampen havde et materiale af metal. Den originale udgave af lampen havde materialet "poleret aluminium", men da modellen blev eksporteret, blev segmenter af lampen gennemsigtige. Dette blev løst ved at give lampen et andet materiale, samt at dele lampen i flere segmenter.
----------	---

Dørkarmene: 5 timer

Dørkarmene	Dørkarmen er lavet i "Autodesk Fusion 360". Dørkarmen blev lavet ligesom væggene og derfor blev løst med samme metode. Dørkarmen blev delt op i tre forskellige segmenter, hvilket er de tre forskellige farver.
------------	--

Rækværket: 7 timer

Rækværket	Rækværket er lavet i "Autodesk Fusion 360". Rækværket kunne ikke bruge materialet "poleret aluminium", da der var fejl i den eksporteret fil. Derfor fik rækværket et andet materiale hvorpå fejlen forsvandt.
-----------	--

Trappen: 6 timer

Trappen	Trappen er lavet i "Autodesk Fusion 360". Trappen havde ingen problemer som de andre assets. Det vigtige var at trappen havde det samme materiale som gulvet.
---------	---

THE HORROR OF THE MAZE

--	--

EXIT skilt: 6 timer

Exit skilt	Exit skiltet er lavet i "Autodesk Fusion 360". Skiltet blev opdelt i flere segmenter hvorpå, at de to forskellige materialer kunne blive anvendt.
------------	---

Pick-up objekter programmering: 10 timer

Inventory: 3 timer

Notecounter:

Spilleren skal have mulighed for at se hvor mange noter denne tog op	Dette blev realiseret vha. et tomt inventory script, som havde en liste i sig
--	---

Pick-up objekter script: 7 timer

Spilleren skal have mulighed for at indsamle objekter til notecounteren	Dette blev realiseret ved hjælp af at
---	---------------------------------------

Bane - 32 timer

Bane tog 32 timer at færdiggøre	
Bane	<p>Som designer vil jeg have en bane der kan virke både forvirrende men også er nem at komme til at kende som vil give en udfordring om at komme rundt for spilleren de første par gange han spiller spillet.</p> <p>Banen er blevet lavet med selvlavede assets sammensat en efter en</p> <p>Som designer vil jeg have en bane hvor der er blindgyder, Chokepoints og genveje som ville gøre hvert spil unikt</p> <p>Banen er blevet bygget med blindgyder og et chokepoint som er trapperne da det er den eneste måde man kan komme nedfra og op på 2 etage.</p>

Lyd: 6 timer

Lyd tog 6 timer at færdiggøre	
-------------------------------	--

THE HORROR OF THE MAZE

Lyd	<p>Som lyddesigner vil jeg have noget musik samt lydeffekter i spillet som passer til spillet dette er for at give en mere mørk tone til et tema der er meningen skal være dystert og skræmmende.</p> <p>Dette er blevet gjort i programmet Adobe Audition til at lave lydende og tonerne og derefter en hjemmeside der hedder soundtrap til at mixe lydende og musikken.</p>
-----	---

Spiller: 6 timer

Spiller tog 6 timer at færdiggøre	
Spiller	<p>Som programmør vil jeg gerne have en spiller som har nemt og bruge Controls so spilleren er velkendt med får at give en bedre spiloplevelse</p> <p>Dette er blevet gjort ved at bruge WASD til at gå, at bruge musen for at kigge rundt og bruge E til at interagere med ting samtidig F til flashlight og escape for at lukke spillet.'</p> <p>Som programmør vil jeg gerne have en spiller som kan bruge et flashlight for at give bedre syn til spilleren</p> <p>Dette blev gjort ved at placere et spotlight som child til playeren og ville skifte enabled tilstand ved at trykke på F.</p>

Døre-programmering: 4 timer

Døre tog 4 timer at færdiggøre	
Døre:	<p>Som designer vil jeg gerne have døre som der har en smooth animation som giver lidt mere visuel værdi til spilleren</p> <p>Dette er blevet udført ved at animere døren inde i Unity ved at skifte den rotation.</p>

AI-modelering: 3 timer

Karakter Modellering tog 3 timer at færdiggøre	
--	--

THE HORROR OF THE MAZE

Karakter Modellering	<p>Som designer vil jeg have et monster som der kunne ligne et eksperiment fra et sindssygehospital for at give en uhyggelig følelse fra den.</p> <p>Dette er blevet gjort ved at modellere en humanoid figur og så trække i modellen dette er blevet gjort i blender.</p>
----------------------	--

Animation: 5 timer

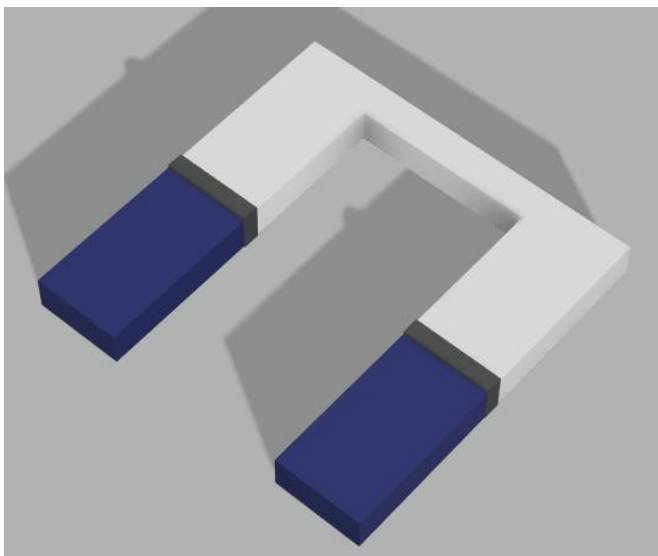
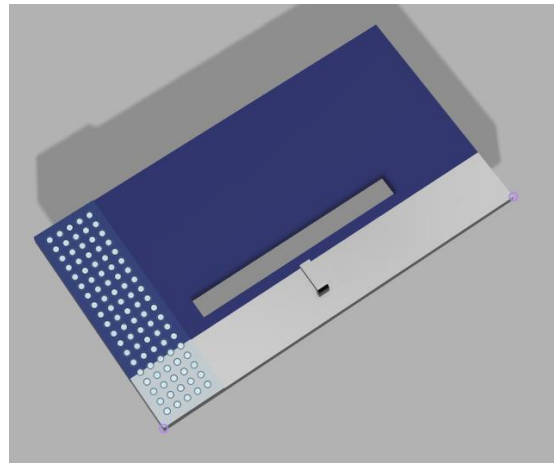
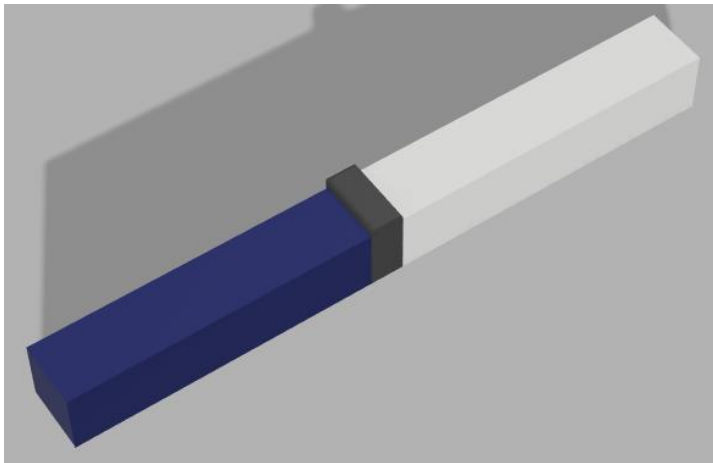
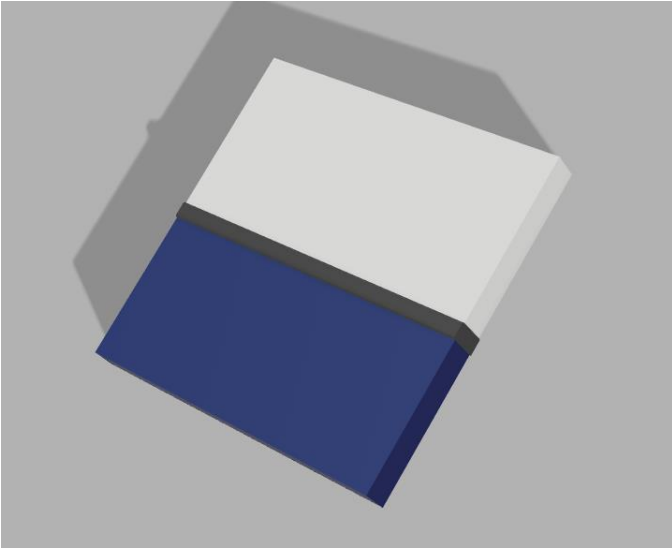
Animation tog 5 timer at færdiggøre	
Animation	<p>Som Animator vil jeg have animationer som passer ind i spillet og som ikke hakker men smooth dette ville give mere visuel værdi</p> <p>Dette er blevet gjort inde i Unity ved bare at ændre på transform til det objekt man var igang med at animere</p>

Lys: 5 timer

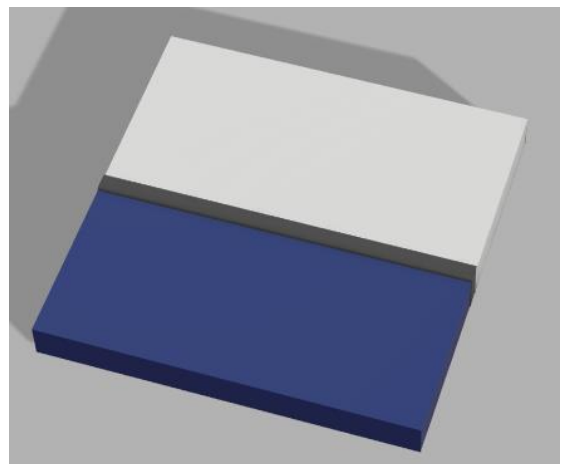
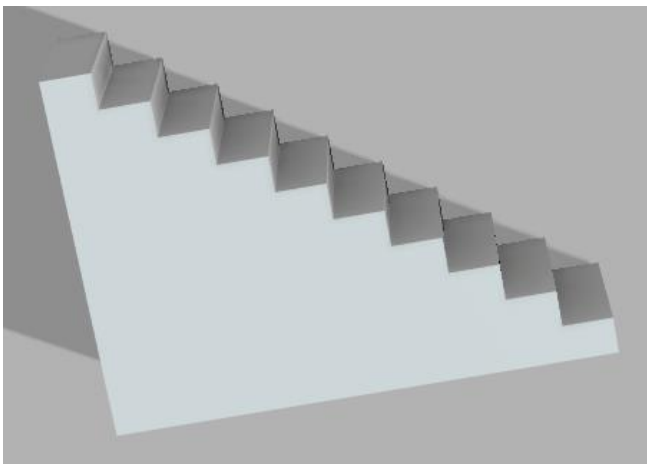
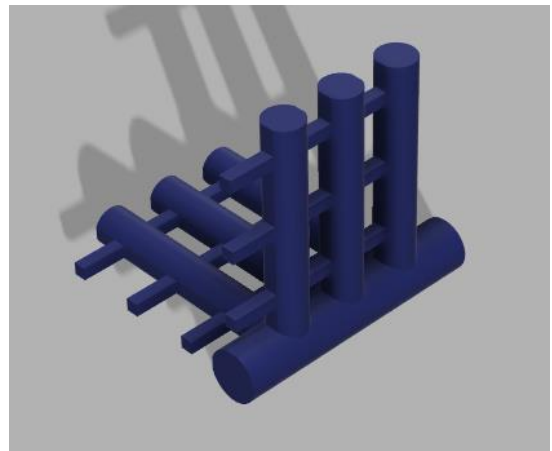
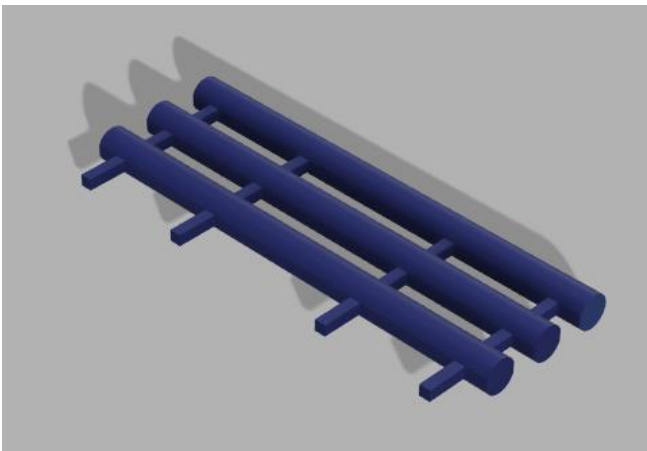
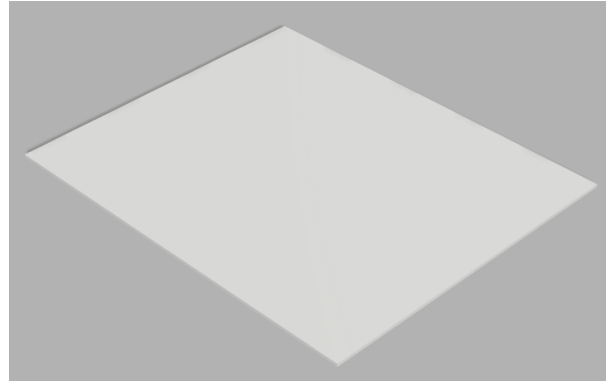
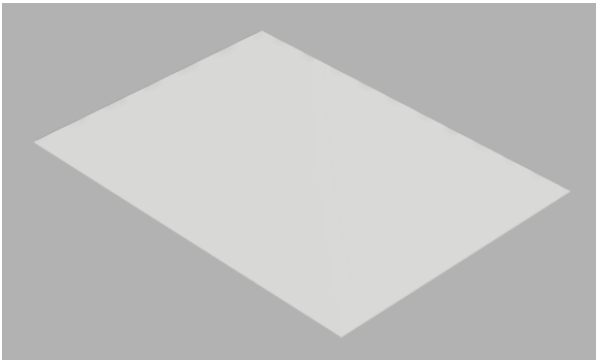
Lys tog 5 timer at færdiggøre	
Lys	<p>Som designer vil jeg have lys som der passer ind til miljøet af et uhyggeligt insane asylum for at give et mere passende miljø til teamet "Horror"</p> <p>Dette er blevet gjort ved at placere får røde "NødLys" rundt omkring i banen som på samme tid også virker som en signifier hvor noterne man skal samle er lokaliseret.</p>

THE HORROR OF THE MAZE

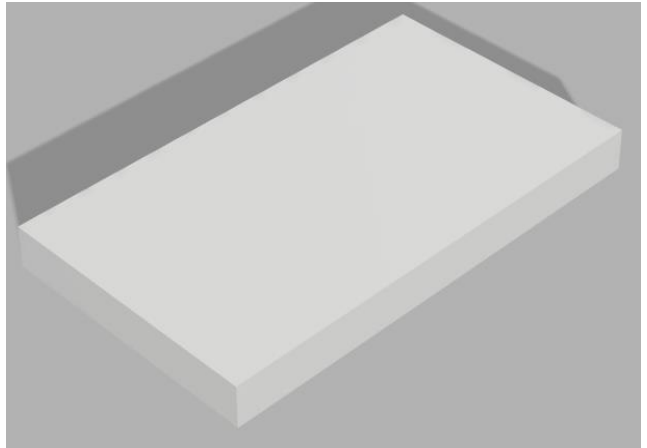
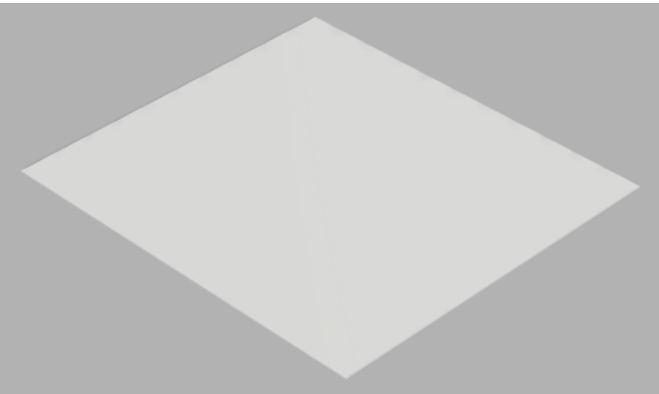
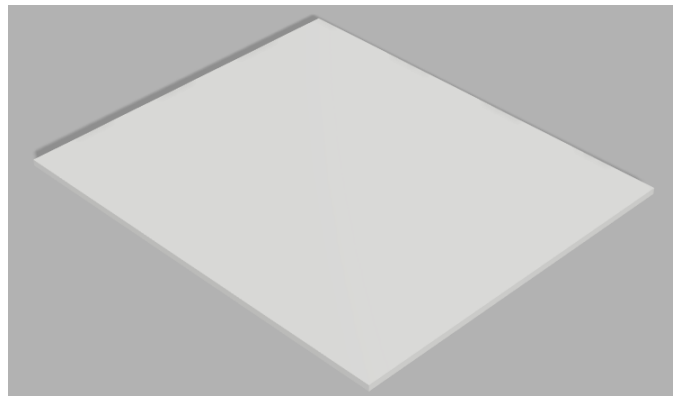
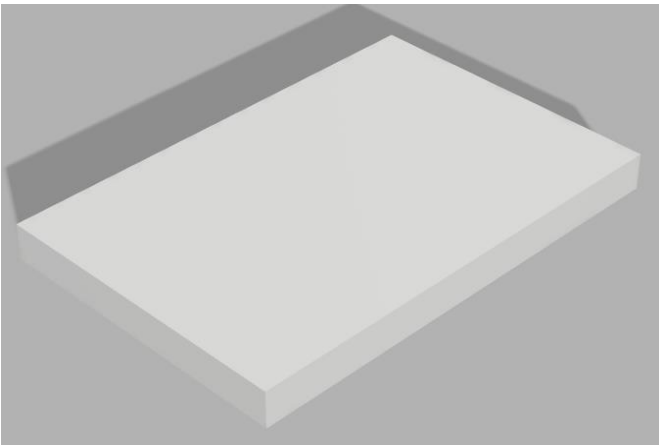
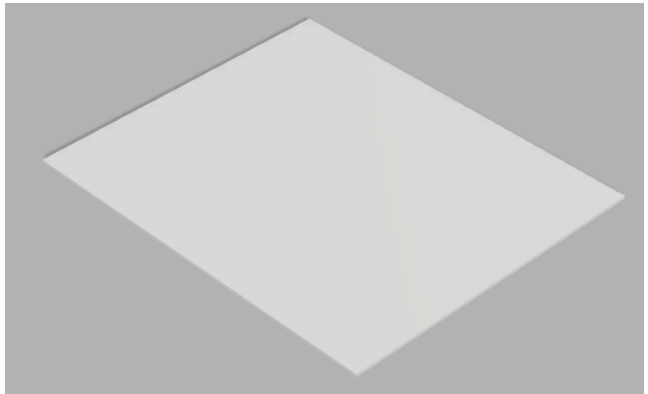
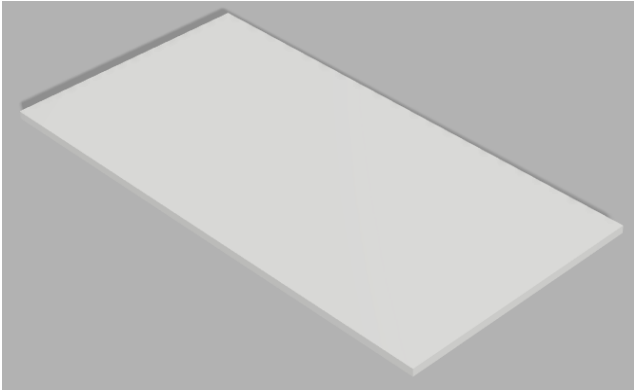
Bilag 10, assets/3d-modelling

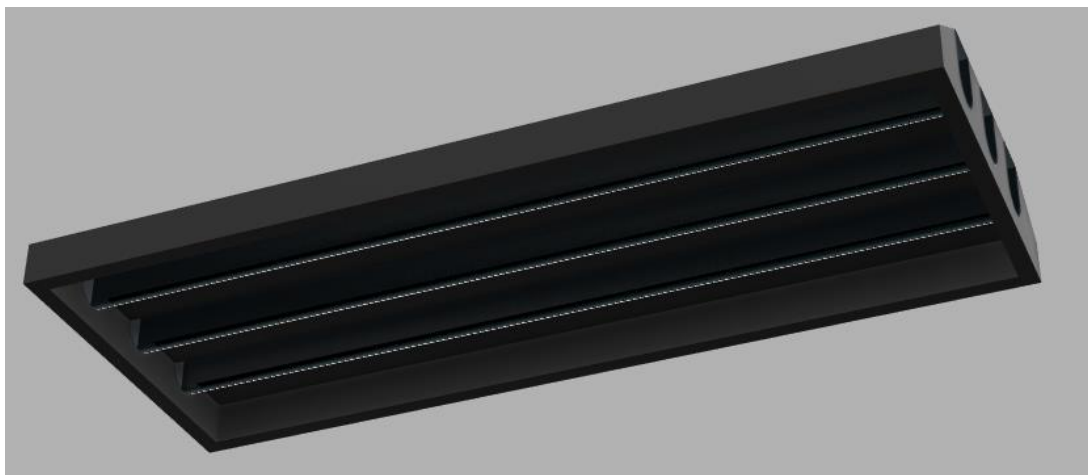


THE HORROR OF THE MAZE



THE HORROR OF THE MAZE





Indholdsfortegnelse

Forord - Zacharias	2
Indholdsfortegnelse	2
Projektbeskrivelse	5
Projektoplæg - Alle	5
Problemidentifikation og analyse Zacharias og Jens	7
Problemformulering - Alle	8
Produkt-/processprincip - Zacharias	8
Projektafgrænsning - Jens	9
Ide - Jens	9
Fokusområde/specifik afgrænsning - Jens	10
Teststrategi - Jens	11
Projekttidsplan/SCRUM - Zacharias	12
Projektrealisering	12
Indledning - Jens og Zacharias	12
Problemanalyse - Jens	13
Produktprincip	16
Krav til vores produkt - Jens	16
Kravuddybning - Zacharias	17
Idegenerering - Zacharias	18
Produktudformning - Zacharias	21
Præsentation af spillet - Jens	21

Spilleren - Jens.....	21
Spillerens bevægelse & lommelygte - Jens.....	21
Pick-up, notecounter og UI-tekst (signifiers og affordances) - Jens.....	24
Den progressive AI - Jens.....	26
General intelligens (pathfinding, waypoints, åbne døre) - Jens.....	26
Progressiviteten og interaktionen med spilleren - Jens.....	28
Design - Emil.....	31
Bane - Emil.....	31
Grafik.....	31
Oprettelse af AI - Zacharias.....	31
Noterne - Zacharias.....	34
Textures - Emil.....	34
Lyd - Zacharias.....	35
Implementering af bane - Anders.....	35
Implementering af banens udformning.....	35
Implementering af dørene.....	36
Implementering af lysene.....	36
Exit-door.....	37
Menu - Anders.....	37
Menu visualiseret.....	37
How-to.....	38
Tutorial - Zacharias.....	39
Spillet cyklus og tilstande for AI og spiller (Flowdiagrammer og tilstandsdiagrammer) - Anders.....	40
Realisering og optimering - Anders og Jens.....	41
User-cases (AI-sprint)- Anders og Jens.....	41
Test.....	42
White Box-test - Zacharas.....	43
Black Box-test - Zacharias.....	43
Sammenligning af Black Box-test & White box-test - Zacharas.....	44
SUS (System Usability Scale) - Zacharias.....	44
Vurdering af testene - Jens.....	45
Konklusion - Jens.....	46
Evaluerings - Alle.....	48

Litteraturliste - Zacharias:	49
Med forfatter.....	49
Uden forfatter.....	50
Bilag:	51
Bilag 1, spørgeskema:.....	51
Bilag 2, White Box spørgsmål & svar.....	55
Bilag 3, Black Box spørgsmål & svar.....	56
Bilag 4, SUS spørgsmål & svar.....	57
Bilag 5, SUS tabel 2 og 3.	60
Bilag 6, BurndownChart.....	61
Bilag 7, AI Struktur, knogler og andre features.	62
Bilag 8, noterne	63
Bilag 9, Releasebacklog + produktbacklog.....	63
3D-modellering sprint + textures sprint - 55timer	63
Pick-up objekter programmering: 10 timer	65
Bane - 32 timer	65
Lyd: 6 timer.....	65
Spiller: 6 timer.....	66
Døre-programmering: 4 timer.....	66
AI-modelering: 3 timer	66
Animation: 5 timer.....	67
Lys: 5 timer	67
Bilag 10, assets/3d-modellering.....	68