

Digital Teknik Eksamen

Af Alexander, Natasja og Mads

[Projekt information: 3](#_Toc34305472)

[Projektbeskrivelse: 4](#_Toc34305473)

[Beskrivelse: 4](#_Toc34305474)

[Brugsscenarier og formål: 4](#_Toc34305475)

[Målgruppe: 5](#_Toc34305476)

[Produkt titel og slogan: 6](#_Toc34305477)

[Story: 6](#_Toc34305478)

[Problemidentifikation og -analyse: 7](#_Toc34305479)

[Tekniske problemer: 7](#_Toc34305480)

[Værktøjer/Teknologier: 7](#_Toc34305481)

[Software: 7](#_Toc34305482)

[Hardware: 9](#_Toc34305483)

[Brug af VR: 10](#_Toc34305484)

[Kravspecifikation: 11](#_Toc34305485)

[Minimum: 11](#_Toc34305486)

[Forventet: 12](#_Toc34305487)

[Håbet: 13](#_Toc34305488)

[Illustrationer: 15](#_Toc34305489)

[Kilder: 17](#_Toc34305490)

# Projekt information:

**Github link:**

<https://github.com/Mfknudsen/DigitalTeknikEksame>

**Github brugernavne:**

Alexander - nissemand243

Natasja - NatasjaDamsbjerg

Mads - Mfknudsen

**Opgave beskrivelse:** <https://digitalteknik.it.slotshaven.dk/?p=4001>

**Godkendt af Erik:**

D. 05-03-2020

Kl: 12:42

# 

# Projektbeskrivelse:

## Beskrivelse:

VR spil baseret på det tidligere VR projekt, hvor vi lavede en kasse med en VR spiller, spiller 1, som havde muligheden for at placere flere nøgler i nogle nøglehuller, hvor den rigtige kombination, som spiller 2 kunne se via en sekundær computer, ville få spillet til at resultere i “Du vandt” og tænde en Philips Hue lampe.

I dette VR spil, vil en VR spiller opleve en verden, hvor de skal komme igennem en eller flere rum, hvor der er forskellige forhindringer, i f.eks. form af puzzles. Alle disse forhindringer kræver at en spiller, spiller 2, i den virkelige verden sender genstande ind i spillet eller forandre den virtuelle verden med f.eks. nye døre eller våben.

De to spillere skal arbejde sammen for at vinde spillet. Spiller 2 vil kunne bestemme hvor disse genstande er placeret og roteret i forhold til spiller 1 i den virtuelle verden, hvor spiller 1 kan se hvor og hvordan ting er placeret i den virtuelle verden, som spiller 2 ikke kan.

**Tematik:**

Escape Room og samarbejde mellem den virtuelle og den fysiske verdens spillere.

## Brugsscenarier og formål:

For spiller 1, der er spilleren i VR, kan spilleren færdes i VR universet, hvoraf denne kan interagere med objekter og andre ting inde i spillet.

Spiller 2, der er i den fysiske verden, vil det være sådan at denne kan kommunikere med spilleren i den virtuelle verden. Det vil ske ved at de er i samme lokale.

Spiller 2 har rollen som spiller 1's bedste ven og fiskemakker, der sejler rundt udenfor denne bygning, som spiller 1 er fanget i (uddybet i selve vores story til spillet).

Spiller 2 vil få nogle virtuelle objekter, som kan gives til spiller 1 i den virtuelle verden ved hjælp af egen computer, således, at det kan hjælpe spiller 1 med at slippe fri.

Når spiller 1 når et bestemt sted i spillet, vil spiller 2 få adgang til flere funktioner for at øge sværhedsgraden og øge niveauet af samarbejde i spillet. Det vil være i form af adgang til sikkerhedssystemet i bygningen, men da der er en fejl i dette som følge af stormen, vil det kun være muligt at ændre 1 ting i sikkerhedssystemet af gangen. Dette kan være at åbne for en dør, der er låst, det kan også være at en trappe er aflåst. Dette element vil gøre at spiller 2 har stor betydning for spillets gang og dermed er mere inkluderet i spillet.

Når spillet er gennemført, er det meningen at et fysisk objekt, eksempelvis en Philips HUE, skal ændres. Det kan bl.a. ske ved netop at en Philips HUE skifter farve, men dette element diskuteres stadig.

## Målgruppe:

Vores målgruppe er unge af en alder på 12 til og med voksne på en alder af 50 år. Alle individer som er med i vores målgruppe, skal møde følgende krav:

* **Venner:**

Personerne i vores målgruppe skal have et minimum på én ven, som også selv møder de andre krav.

* **Fysiologi:**

Spilleren skal have de fysiske evner til at bruge de forskellige redskaber, som skal bruges for at spille vores spil.

* **Mentalitet:**

Spilleren skal have den mentale kapacitet til at kunne samarbejde med en anden spiller om at løse et eller flere mål på den samme tid.

Spilleren skal også have den mentale kapacitet til at kunne stoppe med at spille efter længere tid, så afhængighed ikke bliver dannet.

* **Helbred:**

Spilleren skal have et fysisk og mentalt godt helbred, så de ikke tager skade ved at bruge noget af udstyret til vores spil, som f.eks. motion-sickness fra at bruge en HTC Vive.

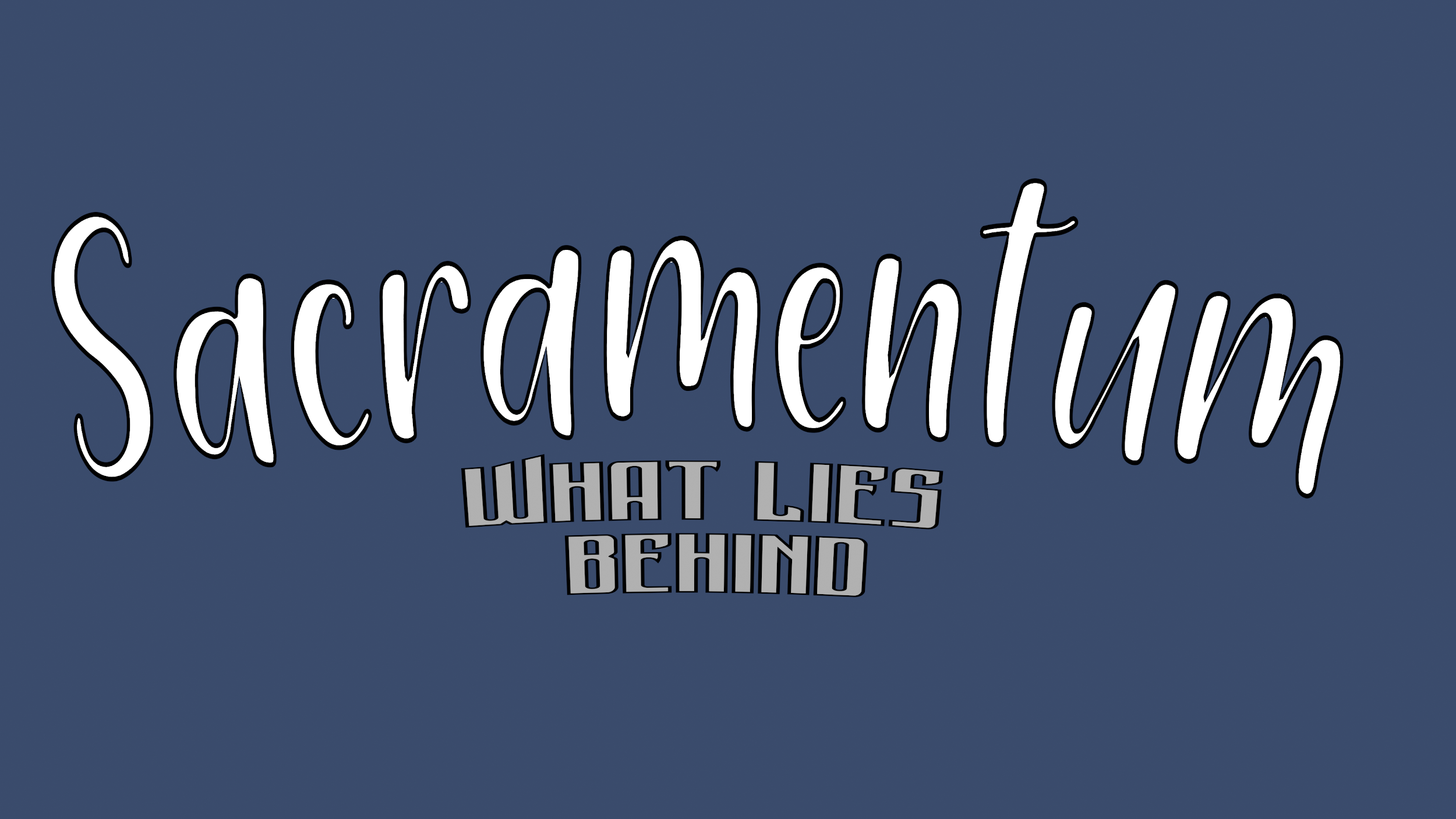
## 

## Produkt titel og slogan:

Sacramentum

What lies behind

Sacramentum er latin for blandt andet mysterium ([Sac](https://en.wiktionary.org/wiki/sacramentum)) og sloganet går ind på hvordan mysterium altid gemmer på noget og at det kunne være direkte forbundet med spillets tema.

’

## Story:

Regnen har taget overhånd i Danmark. En aften trækker det op til voldsom storm. Det regner ekstremt voldsomt og ting begynder at ryste og flyve rundt. Du (spiller 1) søger tilflugt i din kælder indtil stormen stilner af. Da du vågner morgenen efter, er det første du hører en høj alarm. Du ved hvad alarmen betyder. Det er dit sikkerhedssystem der er gået i gang, hvilket vil sige at du er fanget. Sikkerhedssystemet er lavet for at gøre det næsten umuligt for en tyv at komme ud. Du kigger dig omkring og spotter til din store frygt, at du ikke kan se andet end vand ude foran det lille vindue i væggen, og at der er små revner i glasset. Du kigger ud ad vinduet for at tjekke vandstanden, men lysets skær er ikke til at spotte. Det betyder at, hvis vinduet smadrer helt, vil kælderen fyldes med vand og du vil drukne.   
Du går hen til fyret og råber efter hjælp gennem skorstensgangen i håb om, at nogen kan høre dig. Til din lettelse hører du en genkendelig stemme gennem skorstenen. Det er din bedste ven du normalt fisker med (spiller 2). Han sejlede i en lille jolle, da han hørte din stemme. Han forklarer at vandstanden er steget flere meter og at det eneste synlige af huset er taget.   
Det er nu jeres opgave at få dig ud af kælderen, inden vandstanden stiger, og videre ud af huset. I kan lige nu kommunikere ved at råbe til hinanden gennem skakten, og det er muligt for din ven at kaste ting ned til dig gennem skakten.

# Problemidentifikation og -analyse:

## Tekniske problemer:

* Hvordan kan vi få spiller 2 til at styre et objekt inde i VR-rummet.
* Hvordan kan vi lave et spil, som kræver lige meget fra begge spillere.

* Hvordan kan vi lave et system som modtager og sender information mellem de to spillere.

* Hvordan får vi fysiske objekter til at påvirke den virtuelle verden og omvendt.

## Værktøjer/Teknologier:

### Software:

**Unity:**

Til vores projekt bruger vi Unity, som er et gratis game engine, hvor vi har muligheden for at bruge forskellige værktøjer inden for lys, lyd, mm, for at opbygge en virtuel verden, hvori vores spil vil foregå i.

Unity er en platform hvorfra man kan skabe 2D, 3D, VR og AR spil og andre former for simulationer.

Unity er delt op i forskellige vinduer, som hver har deres egen funktionerne. Vi vil bruge:

* **Projekt**, som indeholder de forskellige objekter, scripts, mm, som vi kan bruge i vores spil.
* **Console**, som hjælper os med at finde fejl og med at forbedre vores kode.
* **Inspector**, hvor vi kan se alle detaljerne om et objekt, script, mm.
* **Scene**, hvor vi kan se den virtuelle verden, som spillerne vil kunne opleve når spillet er færdigt med at bygge.
* **Game** som er det samme som *Scene*, men gennem et kamera.
* **Asset Store** hvor vi f.eks. kan hente de nødvendige filer for at kunne bruge HTC Vive og for at bruge Pun 2.

**Visualstudio:**

Unity bruger forskellige værktøjer til at redigere kode, og vi har af disse muligheder, valgt at bruge Visualstudio. Med Visual Studio kan vi redigere, debug og bygge kode. Visual Studio indeholder flere værktøjer i sig selv, som gør det muligt for folk at skabe deres egen udvidelser, som de har udgivet. F.eks. Format Dokument (C#), som vi har taget brug af for at gøre vores arbejdsproces nemmere.

**C#:**

C# eller C Sharp, som det også er kendt for, er et generelt formåls multi-paradigm programmeringssprog, som blandt andet omfatter stærk skrivning.

Vi bruger dette programmeringssprog, da det er et af de to understøttede af Unity, hvor det anede er Javascript, og fordi vi mener at det generelt er bedre end Javascript.

**Pun 2:**

Pun 2 står for **P**hoton **U**nity **N**etworking **2,** som er er real-time multiplayer framework, der er hurtigt, nem og fleksibel.

Det fungerer ved, at man laver en bruger på deres officielle hjemmeside, hvorefter man laver en API, som vil fungere som ens server og bliver skabt med en nøgle, som senere hen skal bruges i Unity. Når man har sin nøgle går man til Asset Store og finder Pun 2(Free) og importere den til ens projekt. Når den er klar, vil et nyt vindue komme op og bede om en nøgle, som svarer til den nøgle man fik tidligere da man lavede en API. Man indsætter nøglen og klikker “Setup Project” og hvis nøglen passer, så er man klar til at bruge Pun 2.

Det kræver derefter kun et par linjer kode for at forbinde ens spil til den tilhørende API, og derefter vil de forskellige komponenter, som følger med Pun 2(Free), automatisk holde styr på alle forbindelser og optimere dem for den bedste oplevelse.

Vi har valgt at bruge denne pakke fordi det kan skabe en stabil forbindelse mellem forskellige enheder og fordi det ikke bliver stoppet af at forbindelserne er mellem en computer og en telefon.

**SteamVR Plugin:**

SteamVR Plugin er den officielle udgivelse fra Valve, som også er producenten bag HTC Vive, og indeholder alt man skal bruge i Unity, for at få forbindelse til en HTC Vive, og til at tolke og bruge de forskellige typer af input, som en HTC Vive vil sende.

De forskellige knapper på kontrollerne kan give forskellige typer af data, fra float, bool og vector 2 og indeholder også flere funktioner, som spilleren indirekte påvirker, som f.eks. placeringen og rotationen af kontrollerne og headsettet.

SteamVr Plugin medbringer også flere scripts som bliver brugt til at udregne position og rotationen af HTC Vive headsettet.

Vi bruger denne pakke fordi den gør det meget nemmere at arbejde med HTC Vive, som vi senere hen gør rede for.

**Blender 3D:**

Blender er et gratis open-software program, som går ind for at give brugeren mulighed for at.

* **Modellere** et 3D objekt ved brug af vertices, edges og faces, som opbygger alle virtuelle 3D objekter,
* **Rigging**, som er den måde hvorpå et 3D objekt kan kontrolleres uden at ændre på den originale form, som så giver muligheden for at.
* **Animeringer** som er når et eller flere objekter påfører sig en eller flere handlinger over tid.
* **Simulationer** som er den metode hvorpå f.eks. realistisk vand bliver lavet og/eller at forskellige objekter, som er påvirket af forskellige former for fysik, som mimere fysikkens love, og påvirker den virtuelle verden omkring dem.
* **Render** hvor alle de tidligere emner bliver samlet til at danne et overstående billede af den virtuelle verden.
* **Compositing** hvor man laver ændringer til ens render, som ikke kunne blive lave sammen med alle objekter.

Vi bruger dette 3D værktøj til at lave visuelle modeller og fordi et af vores gruppemedlemmer har brugt det før, samt at Unity har en funktion hvor man kan lave et Blender projekt og derefter importere det til Unity, hvor der bliver lavet en kopi, som kan redigeres og vil derefter opdatere i Unity, hver gang det blive gemt i den nye Blender kopi.

**OSC:**

OSC står for **O**pen **S**ound **C**ontrol og er en protokol for kommunikationen mellem computere, samt en lyd synthesizer og andre multimedieenheder som er optimeret til moderne netværksteknologier. Det giver os mulighed for at sende float værdier fra en computer til en andet via de forskellige porte som kan findes på en computer, samt give dem en adresse, som kan bruges til at bestemme hvad beskederne skal bruges til efter de er modtaget på en anden computer eller et andet program på den samme computer.

Vi vil bruge OSC til at sende information fra de fysiske, som f.eks. kunne være opfanget med brug af sensorer, og sende det til Unity, som så f.eks. kunne påvirke den virtuelle verden.

### Hardware:

**HTC Vive:**

Vi bruger HTC Vive til at skabe en forbindelse til det virtuelle univers og dermed muliggør at en spiller kan opleve og færdes i den virtuelle verden. Denne hardware bruges sammen med SteamVR Plugin, således at opsætningen og konfigurationen af controllerne sker automatisk, når spillet køres.

Vi bruger en HTC Vive i stedet for f.eks. en mobil, da en HTC Vive kan, udover at være en skærm indtil den virtuelle verden for spilleren, også give denne spiller flere metoder hvorpå de kan påvirke den virtuelle verden, ved brug af de medfølgende controllers, som udover at give spilleren mulighed for at f.eks. samle ting op i den virtuelle verden, så kan vi programmere forskellige typer af input, som de giver, til at påføre forskellige handlinger, som giver os flere muligheder for at lave forskellige typer puzzles.

**Computer:**

Computeren er nødvendig da denne skal køre selve spillet, den skal køre vores plugins, samt trække VR-setuppet og den skal kunne forbinde, via plugins, til andre enheder således at der kan opstå en multiplayer platform. Det gode ved at bruge computere er at vi kan benytte os af fysiske objekter, der kræver en kablet opkobling og dermed udvider vi det spektrum, der er af objekter vi kan benytte os af til dette projekt.

Spilleren som bruger denne computer, vil være spiller 1 og den anden enhed, som også kan være en computer, dog uden VR-setuppet forbundet, vil blive markeret som spiller 2.

**Andre enheder:**

Vi vil bruge forskellige fysiske enheder, som på forskellige måder, som f.eks. sensorer, vil omdannet fysiske objekter til data, der bliver brugt til at forbinde fysiske objekter med VR-spillet.

## Brug af VR:

Projektet gør brug af fagets emne om VR og den fysiske verden, der skaber en forbindelse mellem den virtuelle og den fysiske verden.

Vi forestiller os at spillet skal sættes op et sted hvor der allerede udbydes fysiske Escape Rooms, da der også skal bruges fysiske objekter for at kunne klare spillet. Disse objekter er ikke nogle man normalt har derhjemme, og det vil derfor give mening, at det ikke er til hjemmebrug.

Det fungerer ved, at der er en spiller i VR og en i den fysiske verden. Der skal her være nogle nøglefunktioner for gennemførelsen af spillet placeret i den fysiske verden, således, at man er nødt til at inddrage den fysiske verden for at gennemføre spillet. Der skal også være en kommunikation mellem spilleren i den virtuelle og spilleren i den fysiske verden for, at spillet skal kunne gennemføres og sikrer, at spillet ikke kun er ensidigt. Der skal desuden også være nogle deciderede ting spilleren udenfor VR skal have til rådighed, således at spilleren også føler sig som en del af spillet. Disse interaktioner skal have signifikant betydning for spilleren i VR-verdenen og spillets gennemførsel generelt.

# 

# Kravspecifikation:

## Minimum:

|  |  |  |
| --- | --- | --- |
| Krav | Forklaring | Test |
| Fungerende VR-verden | VR skal virke så spillere, der bruger VR, ikke har nogle ubehageligheder i spillet og så VR spilleren kan spille med og gennemføre spillet. | **Kvantitativ:**  Vi vil sikre mod dette ved at teste vores spil gennem arbejdsprocessen og en VR indstilling pakke, som vil huske hvordan hvilke knapper og andre funktioner, vil reagere med spillets scripts. |
| Multiplayer | Multiplayer skal virke sådan at de to spilleres applikationer aktivt kan kommunikere og have en stabil forbindelse, så der ikke sker afbrydelser under spillet. | **Kvantitativ:**  Vi vil udsætte spillet for forskellige scenarier hvor ved forbindelse vil blive teste. |
| Regler | Begge spillere skal kende reglerne for spillet og vide hvad de kan og ikke kan gøre i spillet. | **Kvalitativ:**  Vi vil ved hjælp af en brugertest, observere om reglerne er til at forstå, uden at få hjælp ude fra. |
| Flere puzzles | Der skal være flere forskellige typer puzzles, så hvordan de kan løses og hvor svære de er, ikke er det samme og give en bedre oplevelse for spilleren. | **Kvantitativ:**  Vi vil tælle hvor mange forskellige puzzles som er lavet og som bliver brugt i det færdige spil. |
| Forbundet | Alle puzzles er forbundet og påvirker hinanden enten direkte eller indirekte. | **Programmatisk:**  Tjekke de forskellige scripts og tjekke de danner et net af forbindelser mellem dem alle. |
| Historie påvirker gameplay | Historien bag setup, opbygger og danner rammerne for et tema. Spillet følger så dette tema. | **Kvalitativ:**  Teste og hører mening om hvorvidt testpersonerne mener at alle objekter passer til historien. |
| Samarbejde | Spillet kan kun gennemføres gennem samarbejde mellem flere spillere. | **Programmatisk:**  Spillet skal være programmeret så det kun kan gennemføres gennem samarbejde mellem flere spillere. |
| Brug af titel, slogan og evt. logo | Vi sætter vores titel, slogan og logo, ind i vores spil som enten en loading skærm i enten starten når spillet starter eller mellem scener. | **Programmatisk:**  Er det programmeret til at komme frem enten ved starten af spillet eller når der skal loades. |

## Forventet:

|  |  |  |
| --- | --- | --- |
| Krav | Forklaring | Test |
| Flere rum | Der skal være flere forskellige rum som hver indeholder et nyt puzzle. De forskellige rum vil have forskelligheder inden for det funktionelle og det grafiske. | **Kvantitativ:**  Vi vil tælle hvor mange dele af den virtuelle verden som vil tælle som deres egne rum. |
| Livlig verden | VR spilleren skal kunne påvirke flere objekter i den virtuelle verden sådan at det ikke kun er objekterne, der er løsningerne til et puzzle. | **Programmatisk:**  Er der programmeret flere objekter, som ikke er nødvendig for et eller puzzel, som kan VR spilleren kan interagere med. |
| Bevægelighed | VR-spilleren skal kunne bevæge sig rundt i den virtuelle verden, så VR-spilleren ikke skal gå rundt i den virkelige verden eller teleportere, for at komme rundt i den virtuelle verden. | **Programmatisk:**  Er der programmeret en funktion som kan bevæge VR spilleren i den virtuelle verden, uden brugen af teleportering funktionen. |
| Fysik | Den virtuelle verden skal fungere ved brug af fysik, så objekter ikke falder gennem gulvet når VR spilleren giver slip på et objekt. | **Programmatisk:**  Er de forskellige objekter som opbygger den virtuelle verden programmeret til at følge de basale fysiske love. |
| Lyd | Der skal være lyde i spillet, sådan at spiloplevelsen bliver mere realistisk | **Programmatisk:**  Er der programmeret så spillet afspiller lyde. |

## Håbet:

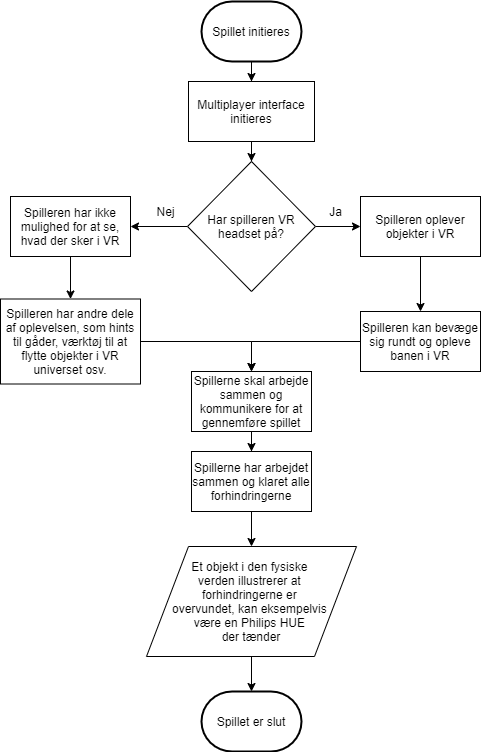
|  |  |  |
| --- | --- | --- |
| Krav | Forklaring | Test |
| Fra fysisk verden til virtuel verden | At implementere en teknologi, som gør det muligt for spilleren i den fysiske verden at sende objekter ind i den virtuelle verden, og har kontrol over position og rotation af objekter når det bliver sendt ind. | **Programmatisk:**  Er der programmeret så der kommer input og/eller output til og/eller fra spillets virtuelle verden til den fysiske verden. |
| Tid | Vi vil implementere en timer, der tæller ned, sådan at man skal klare alle puzzles inden tiden løber ud.  Timeren skal tydeligt kunne forstås af begge spillere. | **Programmatisk og kvantitativ:**  Er der programmeret en klokke, og ved brug af spørgeskema, kan klokken tydeligt ses og forstås. |
| Surround Sound | At lydende i spillet bliver afspillet baseret på dets placering i forhold til VR spilleren. | **Kvantitativ og kvalitativ:**  Mener testpersoner at de kan finde lydens origin point i spillet ud fra hvordan det bliver afspillet. |
| Intro film | At når begge spillerne er klar til at starte spillet, skal en lille minifilm starte, som fortæller forhistorien til spillet. | **Kvalitativ og programmatisk:**  Er filmen lavet til at starte og kan spillerne forstå hvad filmen fortælle i forhold til den virtuelle verden de spiller i. |
| VR figur omtaler sine omgivelser | At den figur som VR spilleren kontrollere, vil omtale sine omgivelser. | **Programmatisk:**  Er der sat lydfiler ind og er figuren programmeret til at tage brug af dem. |

# 

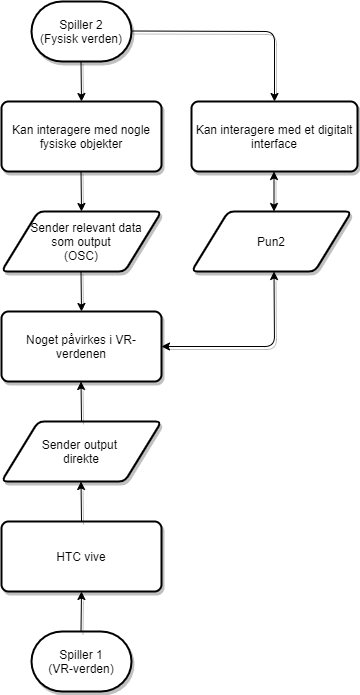
# Illustrationer:

**Flowchart:**

Beskriver spillernes rolle og funktioner i spillet, og den foreløbig gennemgang i spillet:



Beskriver input og output relationerne mellem spillerne og Unity spillet:



# Kilder:

Sacramentum (Sac):

<https://en.wiktionary.org/wiki/sacramentum>

Hvad er Unity (U):

<https://en.wikipedia.org/wiki/Unity_Technologies>

Hvad er Visual Studio (VS):

<https://docs.microsoft.com/en-us/visualstudio/get-started/visual-studio-ide?view=vs-2019>

Hvad er C# (C):

<https://en.wikipedia.org/wiki/C_Sharp_(programming_language)>

Hvad er Pun 2 (P2):

<https://doc-api.photonengine.com/en/pun/v2/>

Hvad er SteamVR Plugin (S-VR):

<https://github.com/ValveSoftware/steamvr_unity_plugin>

Hvad er Blender 3D (B):

<https://www.blender.org/about/>

Hvad er OSC (OSC):

<http://opensoundcontrol.org/introduction-osc>