Høgskulen på Vestlandet

Fakultet for ingeniør- og naturvitskap

Institutt for datateknologi, elektroteknologi og realfag

Verktøy for behandling av MR-fobi, gjennom eksponeringsterapi i VR

Tool for treatment of MRI-phobia, through exposure therapy in VR

Systemdokumentasjon

Versjon <2.0>

REVISJONSHISTORIE

Dato	Versjon	Beskrivelse	Forfatter
07.05.2022	1.0	Første iterasjon	Øystein Kvilhaugsvik Vebjørn Vårdal Alexander Øen
15.05.2022	2.0	Andre iterasjon	Øystein Kvilhaugsvik Vebjørn Vårdal Alexander Øen

INNHOLDSFORTEGNELSE

1 INNLEDNING **1**

2 ARKITEKTUR **2**

3 KODESTRUKTUR **3**

3.1 Scene 3

3.2 Assets 5

4 KLASSEDIAGRAM **6**

5 DATABASEMODELL **7**

6 SERVER-TJENESTER **8**

7 SIKKERHET **9**

8 INSTALLASJON OG KJØRING **10**

9 DOKUMENTASJON AV KILDEKODE **11**

10 KONTINUERLIG INTEGRASJON OG TESTING **12**

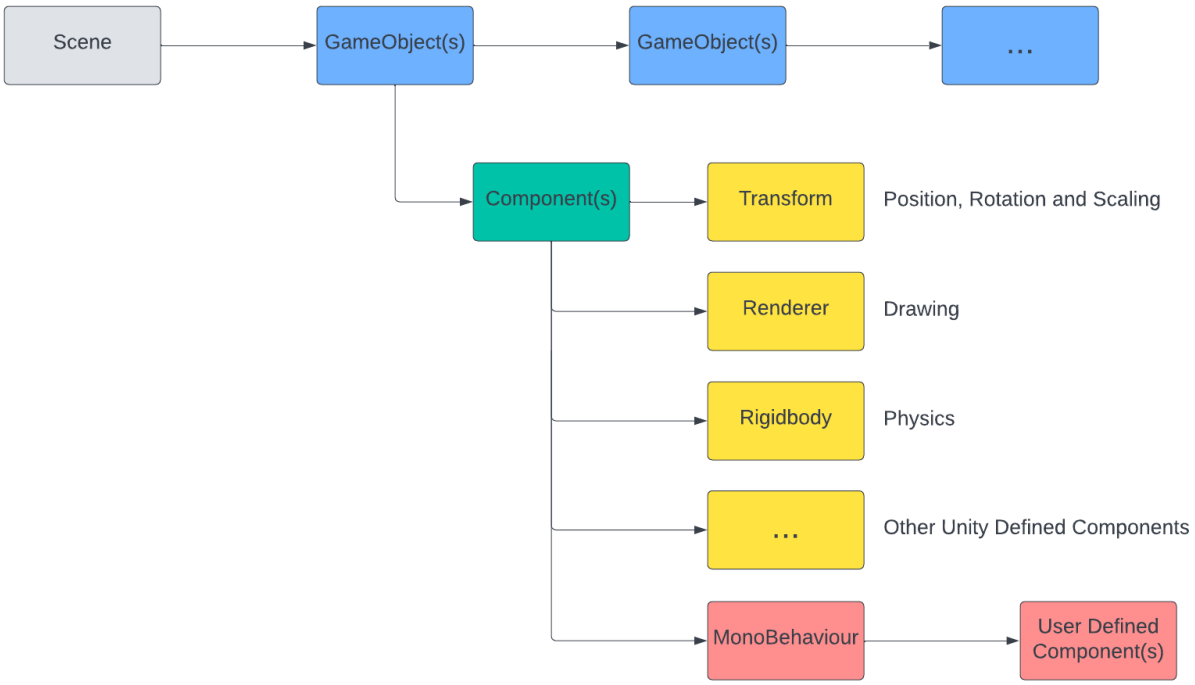
11 REFERANSER **13**

1 INNLEDNING

Dette dokumentet er skrevet i forbindelse med utførelsen av et bachelorprosjekt hos HVL. Det tar for seg arkitektur, kodestruktur, installasjon og kjøring og dokumentasjon av kildekode for prosjektet. Hensikten med dette dokumentet er å gi leseren et overblikk over designet til applikasjonen og hvordan den er laget.

2 ARKITEKTUR

Arkitekturen til systemet er forhåndsdefinert, ettersom det utvikles i Unity, som er et ferdig utviklingsrammeverk. I figur 2.1 kan man se et overblikk av arkitekturen som et tatt i bruk.



Figur 2.1: *Objektdiagram som viser Unity sin komponentbaserte programvareutvikling*

En scene består av et eller flere spillobjekt, hvor hvert spillobjekt kan ha hver sine komponenter. To spillobjekt kan enten være strukturert side ved side eller som forelder og barn. Dersom et spillobjekt er et barn av et annet spillobjekt, vil barnet arve posisjon, rotasjon og skalering fra forelderen.

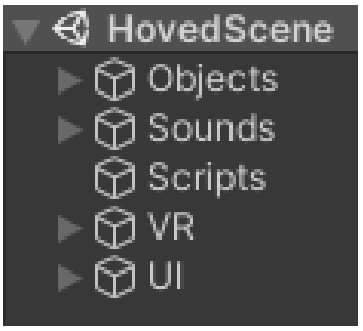
Komponentene til spillobjektene sørger for funksjonaliteten i scenen. Unity kommer med forhåndsdefinerte komponenter, som vist i gult på figur 2.1, men brukere kan også definere egne komponenter ved å skrive skript i programmeringsspråket C#. Skripting er essensielt for programutvikling i Unity (Unity Documentation, 2021).

3 KODESTRUKTUR

Kodestrukturen til prosjektet kan deles opp i to deler; Scene og Assets. Scene tar for seg strukturen til scenen og Assets tar for seg strukturen til ressursene til scenen.

3.1 Scene

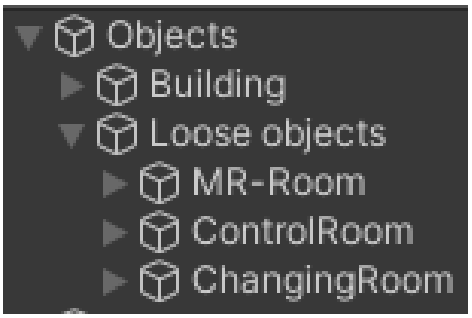
Strukturen til scenen er delt inn i hovedgruppene: Objects, Sounds, Scripts, VR og UI. I figur 3.1 ser du en oversikt over strukturen i Unity.



Figur 3.1: *Strukturen til scenen*

3.1.1 Objects

Her ligger alt du kan se i scenen. Alt fra vegger og gulv til stoler og MR-maskin. Objektene er videre strukturert hvor bygningen og løse objekter er hver for seg, og løse objekter er delt inn i hvilket rom de befinner seg i, som vist i figur 3.2.



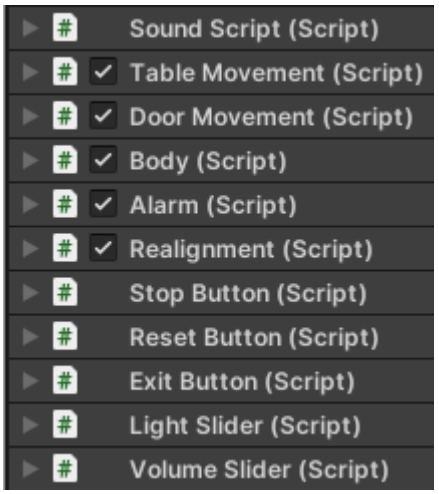
Figur 3.2: *Strukturen til synlige objekter*

3.1.2 Sounds

Her finner du alt av lyder du hører i scenen.

3.1.3 Scripts

Her er omtrent alle selvlagde skript festet som komponenter. I figur 3.3 kan du se alle skriptene som er festet.



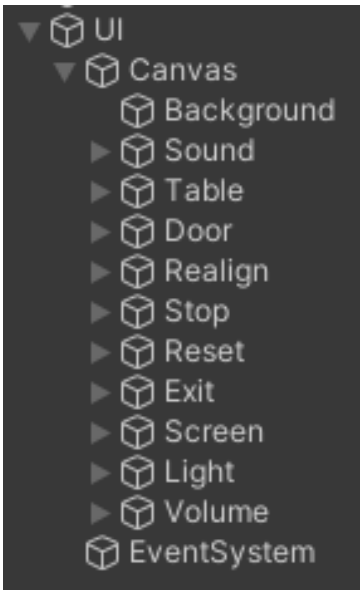
Figur 3.3: *Skript i form av komponenter som er festet til "Scripts"-objektet*

3.1.4 VR

Her er alt som omhandler VR-funksjonalitet.

3.1.5 UI

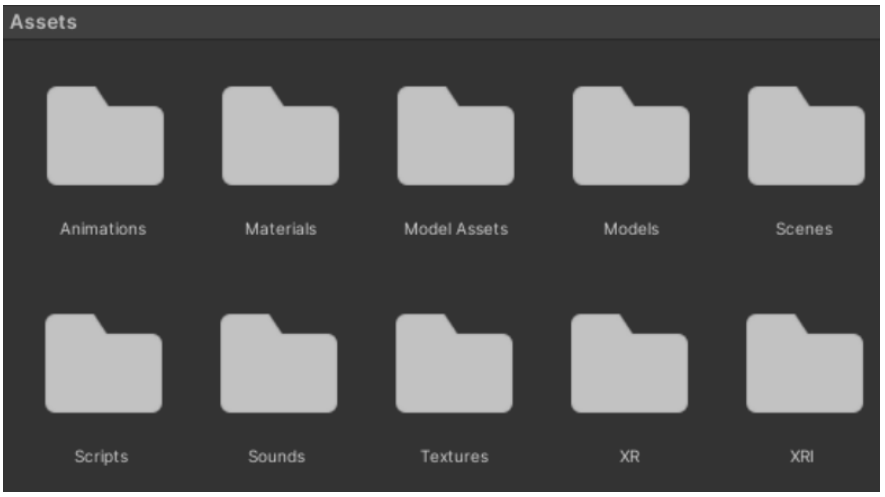
Kontrollpanelet til terapeuten finner du her. Objektene er delt inn i hver sin funksjonalitet, som vist i figur 3.4.



Figur 3.4: *Oversikt over objektene til kontrollpanelet til terapeuten.*

3.2 Assets

Ressursene er delt inn i hver sine kategorier, som vist i figur 3.5.



Figur 3.5: *Oversikt over Assets*

3.2.1 Animations

Selvlagde animasjon-filer.

3.2.2 Materials

Selvlagde material-filer.

3.2.3 Model Assets

Modell-filer og tilhørende filer prosjektgruppen har fått fra oppdragsgiver og internett.

3.2.4 Models

Selvlagde modell-filer.

3.2.5 Scenes

Selvlagde scene-filer.

3.2.6 Scripts

Selvlagde skript-filer, skrevet i programmeringsspråket C#.

3.2.7 Sounds

Selvlagde lydfiler.

3.2.8 Textures

Selvlagde teksturer.

3.2.9 XR

Nødvendige filer for OpenXR, hentet fra OpenXR-pakken.

3.2.10 XRI

Nødvendige filer for “XR Interaction Toolkit”, hentet fra “XR Interaction Toolkit”-pakken.

4 KLASSEDIAGRAM

Prosjektgruppen har unngått å lage et klassediagram for dette prosjektet, ettersom det ikke har egnet seg. Applikasjonen som utvikles i dette prosjektet har ikke klasser med attributter og relasjoner, men blir utviklet i unity med objekter og komponenter.

5 DATABASEMODELL

Det er ikke laget en databasemodell for dette prosjektet, ettersom applikasjonen ikke bruker en database.

6 SERVER-TJENESTER

Det er ikke brukt noen server-tjenester i dette prosjektet.

7 SIKKERHET

Prosjektet har ingen spesiell eller ekstra beskyttelse for uautorisert tilgang. Applikasjonen lagrer ingen informasjon om brukeren, og har derfor ingen data å beskytte.

8 INSTALLASJON OG KJØRING

For å kunne kjøre applikasjonen trenger man følgende:

1. En datamaskin med mus, skjerm og tastatur.
   (For kjøring av applikasjonen og for at terapeuten skal kunne styre kontrollpanelet).
2. Et VR-system med VR-briller, to VR-kontrollere og en tilkoblingskabel.
   (For at pasienten skal kunne se og bevege seg i applikasjonen)
3. Et bord på høyde mellom 0.5 og 1 meter.
   (For at pasienten skal kunne legge seg på bordet i applikasjonen)
4. En "build" av applikasjonen tilgjengelig på datamaskinen.
   (En "build" av applikasjonen er en ferdigbygget versjon, med en kjørbar fil)

Man bør også ha følgende:

1. Et kraftig nok grafikkort i datamaskinen.
   (For at applikasjonen skal kjøre fint og ikke hakkete)

For å starte applikasjonen gjøres følgende:

1. Start datamaskinen og koble til VR-systemet.
2. Åpne "builden" av applikasjonen og kjør "Anxieyeet.exe"-filen.

9 DOKUMENTASJON AV KILDEKODE

Det er skrevet dokumentasjon i form av kommentarer i alle selvlagde skript, men det er ikke opprettet noen måte å generere visning av dokumentasjonen.

I eksempelet i figur 5.1 kan man se stilen som er brukt for å skrive kommentarer.

```
// Class for any general button with 2 states in Unity.
public class GeneralButton : MonoBehaviour
{
    /*  Global variables:
     *  pressed, boolean variable representing if the button is pressed or not.
     *  pressedMessage, a user defined string that will be shown on the button when its pressed.
     *  unpressedMessage, a user defined string that will be shown on the button when its not pressed.
     */
    public bool pressed = false;
    [SerializeField] private string pressedMessage;
    [SerializeField] private string unpressedMessage;

    /*  Method ChangeState:
     *  Alters the text component of the button and the pressed variable, based on the pressed variable.
     */
    public void ChangeState() {
        if (pressed) {
            Text txt = transform.Find("Text").GetComponent<Text>();
            txt.text = unpressedMessage;
            pressed = false;
        } else {
            Text txt = transform.Find("Text").GetComponent<Text>();
            txt.text = pressedMessage;
            pressed = true;
        }
    }

    /*  Method TurnOff:
     *  Changes the value of the text component of the button into the unpressedMessage variable and sets the pressed variable to false.
     */
    public void TurnOff() {
        Text txt = transform.Find("Text").GetComponent<Text>();
            txt.text = unpressedMessage;
            pressed = false;
    }
}
```

Figur 5.1: *Eksempel av dokumentasjonsstil i GeneralButton-skriptet.*

10 KONTINUERLIG INTEGRASJON OG TESTING

Applikasjonen har ingen kontinuerlig integrasjon og er ment å være ferdig utviklet i denne omgang. Applikasjonen er åpen for å bli videreutviklet i fremtiden, basert på ønskene til eierne av kildekoden.

Applikasjonen har ingen enhetstester og er kun testet ved å kjøre-teste applikasjonen. De selvlagde skriptene kunne det blitt laget enhetstester for, men det er ikke gjort i denne omgang. Prosjektgruppen vurderte det unødvendig å lage enhetstester ettersom det var lett å se på kjøre-testene om funksjonaliteten gjorde det den skulle. Ved en eventuell videreutvikling kan det være aktuelt å lage enhetstester, ettersom funksjonaliteten antagelig blir mer avansert.

11 REFERANSER

Unity Documentation (2021) Scripting. Tilgjengelig fra: https://docs.unity3d.com/Manual/ScriptingSection.html (Hentet: 06.05.2022).