

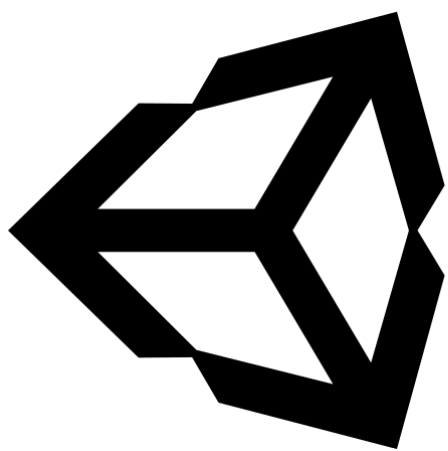
Intro til API (HTC-VIVE, C# og Unity

Af Frederik Foss Nielsen



VIVE

htc | STEAM® VR



unity

Indholdsfortegnelse

Indholdsfortegnelse	2
Opgavebeskrivelse	3
Selve Øvelsen	3
Ekstra Øvelsen	3
Installationsvejledning	3
Udførelse	3
Øvelse 1	3
Ekstra Øvelse	3
Hvad er det?	4
HTC-VIVE	4
UNITY	4
Konklusion	4
Bilag	4
Pause.cs (Script som der bliver brugt i Ekstra øvelsen)	4
Billede af Player indstillingerne	5
Billede af Video Objektet	6
Billede af Unity Færdig applikation	6

Opgavebeskrivelse

Selve Øvelsen

I skal nu programmere en 360 grader video afspiller men før i starter på projektet, skal der installeres Visual studio, Unity, Steam og HTC Vive på Lenovo 90H2. Det er en god idé at starte med at lave simple projekter som en videoafspiller i C# inde i stater med at programmere en 360 grader video afspiller (VR video afspiller).

Ekstra Øvelsen

Det er mulige at prøve ekstra kræfter med HTC-Vive controlleren f.eks start og stop video med HTC-Vive controlleren.

Installationsvejledning

Efter HTC-Vive var sat til pc'en startede man steam hvorefter man startede steamvr. Da det var oppe og køre skulle man køre rum opsætningen. Når det var gjort var alt installeret og det virkede. Normalt ville man også skulle installere den software der følger med vive og installere steam. men da det allerede var gjort var det ikke noget der skulle gøres her.

Udførelse

Øvelse 1

Man laver et project i unity hvorefter man følger denne guide

<https://docs.google.com/document/d/1JjOQ0dXTYPFwg6eSOIAdgyPo6QMLgh-PETwxf8ZVD8/edit>

Når man har gennemgået guiden skal man slå vr support til i unity. Det gør man ved at gå ind på player menuen Edit/Project Settings/Player hvorefter man slår Virtual reality support til i xr menuen. der er et billede af menuen nede i bilag

Ekstra Øvelse

Når man har lavet den første øvelse skal man få en af controllerne til at kunne pause og afspille videoen og man skal også kunne afspille videoen i forskellige hastigheder.

For at kunne gøre det skal man først installere et SteamVR plugin. Man gør det ved at gå ind på Asset store, som man gør ved at klikke på følgende /Window/Asset Store. Så søger man på SteamVR og så burde det gerne være den første der kommer frem. Når man så har installeret den vil der komme en popup, men der skal man bare klikke på ignore all.

Herefter skal man lave et script som kan gøre alle de nødvendige ting. Først laver man et script. derefter hiver man det op på video objektet som man lavede ved at følge guiden. Når

det er gjort skal man klikke på video objektet og så klikke på add component, hvorefter man så søger efter Steam_VR_Tracked_Controller og tilføjer det. Så skal man sætte controller index til at være 3 eller 4 da det er de 2 Controllers index. Man tilføjer derefter den kode som ligger nede i bilag til scriptet man har lavet. Når koden er gemt i scriptet skal man hive video objektet ind i scriptets tomme plads som der er kommet (Se bilag for billede). Det gør man for at scriptet kan starte og stoppe videoen samt bestemme hastigheden.

Hvad er det?

HTC-VIVE

HTC-VIVE er et vr headset som man kan bruge til at spille vr spil med. Men udover at man kun kan spille kan man bruge det til at designe et rum og så vise det og gå rundt og se det som om det var i virkeligheden. det virker ved at der er 2 base stations som kan ved at udsende noget lys finde ud af hvor headsettet er og controllerne er.

UNITY

Unity er en spil engine som man kan bruge til at lave 3D og 2D spil. Det er lavet af Unity Technologies som er et danskejet firma. Unity kan lave spil til mange platforme såsom Windows, Mac Os, Xbox, Playstation og flere. Man kan programmere spillene i C# og det kan gøres i enten monodevelop eller i visual studio.

Konklusion

Min konklusion er at det har været virkeligt spændende og lærerigt at få lidt bekendtskab til api'er og det at få lov til at arbejde med VR har været nice. Men ville lidt have ønsket der var lidt mere i opgaven end der var, sådan som at man skulle få videoen til at afspille i forskellige hastigheder. Selvom det ikke var et krav har jeg gjort så man godt kan skifte hastigheden da det ikke er så svært at implementere når man først har lavet ekstra øvelsen. Men Lenovo pc'en kunne godt bruge noget mere ram da for mit vedkommende arbejdede pc'en meget langsomt da den ikke havde været tændt i noget tid og der derfor var mange windows updates kørende i baggrunden og Unity havde det med at crashe hver gang vinduet fik fokus efter jeg f.eks. havde set en video på nettet.

Bilag

Pause.cs (Script som der bliver brugt i Ekstra øvelsen)

```
using UnityEngine;
using UnityEngine.Video;

public class pause : MonoBehaviour {
```

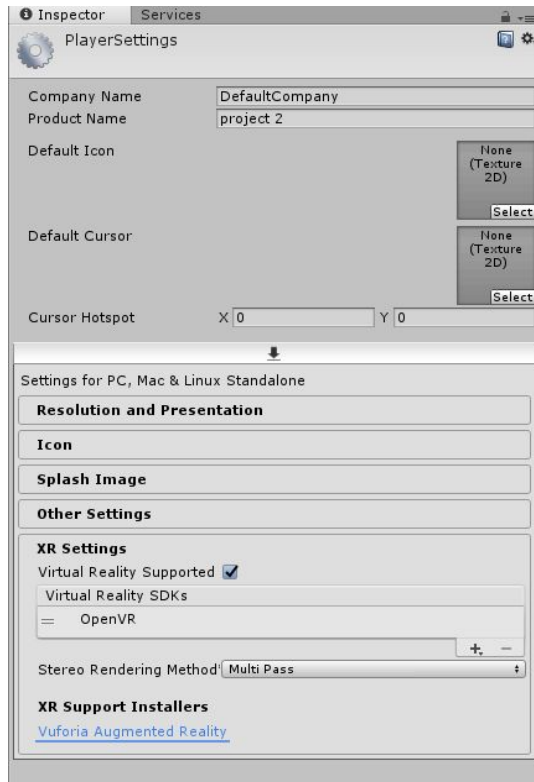
```
//det her er det tomme felt som man ser i unity editoren som er videoplayeren
public VideoPlayer player;
private SteamVR_TrackedController device;
private int count = 2;
//Det her float array bruges til at opbevare hastighederne
private float[] value = {0.1f, 0.5f, 1f, 2f, 4f, 8f };
// Use this for initialization
void Start () {

//Det her er Controller scriptet man henter ind som også er blevet tilføjet til videoen
    device = GetComponent<SteamVR_TrackedController>();
//De næste 2 linjer bruges til at oprette listeners for knapperne
    device.TriggerClicked += Device_TriggerClicked;
    device.PadTouched += Device_PadTouched;
}

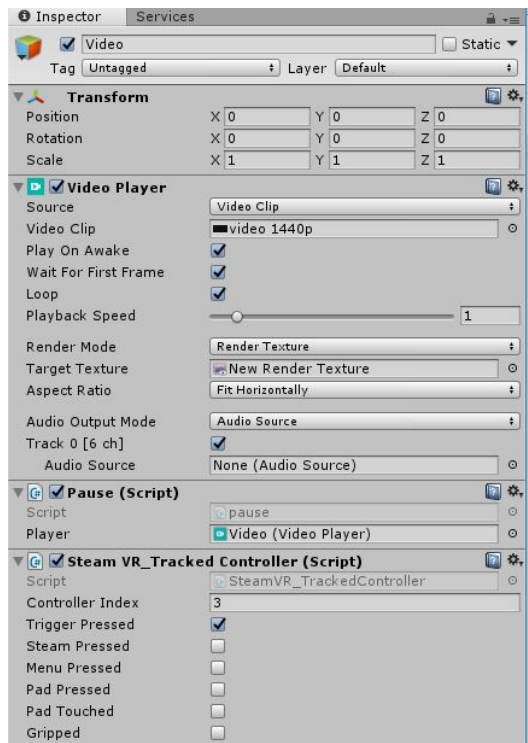
//den her metode sætter hastigheden på videoen ud fra hvor der er blevet trykket på
touchpanelet
    private void Device_PadTouched(object sender, ClickedEventArgs e)
    {

        if (e.padX < 0 && count > 0) count--;
        else if (e.padX > 0 && count < 5) count++;
        player.playbackSpeed = value[count];
    }
//den her metode starter og stopper videoen når knappen bliver klikket
    private void Device_TriggerClicked(object sender, ClickedEventArgs e)
    {
        if (player.isPlaying)
        {
            player.Pause();
            return;
        }
        player.Play();
    }
}
```

Billede af Player indstillingerne



Billede af Video Objektet



Billede af Unity Færdig applikation

