

Inleiding

In dit bestand beschrijf ik het proces uit dat ik heb doorlopen voor mijn stageproject. Zoals eerder uitgelegd, heb ik samen met Bram besloten om een 'kinetic lights' installatie te maken omdat we beide deze kant op wilden.

De kinetic lights installaties vond ik erg interessant omdat je niet alleen speelt met digitale elementen maar ook fysieke objecten. Juist de samenwerking van digitaal en deze technische fysieke objecten vormen voor mij een magisch gevoel omdat je speelt met beweging da

Mijn project binnen mijn stage bij Motion Experience zal het onderzoeken, bedenken, creëren en installeren zijn van een toepassing voor de Foyer om deze ruimte interessanter te maken voor de bezoekers.

Inhoud

Inleiding.....	1
Probleemanalyse en aanpak.....	3
Doel	3
Aanpak.....	3
Onderzoek en inspiratie	3
Foyer onderzoek	4
Inspiratie.....	4
Conclusie.....	12
Experimenteren	12
Alleen aansturing realiseren.....	13
Video uitlezen vanuit Unity	14
Eigen applicatie schrijven	16
Unity simulatie van foyer.....	18
C# applicatie ontwerp	19
Foyer simulatie als test toepassing	21
Unity & C# connectie.....	21
Concepting.....	23
Moving storyboard.....	23
Realisatie	31
DMX512	31
Vereisten Decoder applicatie	33
Feedback	34
Reflectie.....	34

Probleemanalyse en aanpak

Voordat ik ga beginnen met onderzoeken en experimenteren, ga ik eerst definiëren hoe ik dit ga aanpakken. Het is belangrijk om eerst een doel te stellen van mijn project. Ik heb als allereerste een moodboard gemaakt van licht installaties die mij aanspreken om uit te werken of te proberen. Het doel is samen met Tim, Gijs en Bram vastgesteld, we hebben besproken wat zij ieder verwachten van mijn project. Zij gaven bijvoorbeeld aan dat zij iets met de foyer wilden doen aangezien deze ruimte in hen ogen nog te saai is. Ik wilde graag iets doen met een soort 'kinetic lights' installatie, deze installaties heb ik vaker langs zien komen en ook in mijn moodboard verwerkt. Dit zou dus een mooie opvulling kunnen zijn voor de foyer. Toen ik dit voorlegde waren ze erg enthousiast en vertelden ze dat ze hier ook eerder aan gedacht hadden. Deze 'kinetic lights' installatie in de foyer was nog niet helemaal zeker dus heb ik voor mezelf een breder doel opgezet dat ook relevant is.

Bram heeft eerder aangegeven dat Motion nog niet de juiste doelgroep heeft bereikt. Dat zou hij graag willen veranderen. Nu komen er veel tiktokkers en ouders met jonge kinderen en Bram zoekt nu naar wat serieuzer publiek. Dit 'serieuzer publiek' zijn mensen die meer naar een achterliggende betekenis kijken en dit ook interessant vinden en niet per se mensen die alleen komen voor een foto. Hij denkt dit te kunnen te veranderen door meer kunstzinnige installaties te introduceren. Dit spreekt mij ook aan omdat ik die betekenis achter een installatie ook interessant vind. Daarom zijn we tot het volgende doel gekomen.

Doel

Het doel van deze installatie is om Motion meer de kunst richting op te sturen omdat Motion daarin wat mist. De foyer interessanter te maken voor de bezoekers van Motion en Dinner in Motion. Deze beleving moet de bezoekers positief beïnvloeden.

Aanpak

Nadat we het doel gesteld hebben ben ik gaan kijken naar hoe ik dit kan aanpakken. Zoals in mijn vorige semesters ga ik beginnen met inspiratie opdoen en onderzoek. Het is verstandig om eerst inzicht te krijgen op wat de foyer mist, dit kan ik doen door medewerkers te vragen wat zij missen in de ruimte. En wat de foyer meer kunstzinnig kan maken. Maar ook ga ik zelf de foyer goed bekijken en bedenken wat ik zou kunnen doen om het interessanter te maken.

Onderzoek en inspiratie

Ik heb verschillende manieren van onderzoek gebruikt; observatie, enquête, onlineonderzoek en experimenterend onderzoek. Observatie en enquête heb ik allebei gebruikt voor het onderzoeken van de huidige situatie van de foyer. Onlineonderzoek voor inspiratie opdoen en het zoeken naar technische aanpakken. En experimenterend onderzoek tijdens het realiseren.

Foyer onderzoek

Bezoekers van Motion Experience komen tijdens hun bezoek langs de foyer om hier even te zitten, drinken en iets te eten.

Dinner in Motion ontvangt de bezoekers kort voor de show ook in de foyer, hier krijgen ze een introductie van de komende show en geeft de kok een korte demonstratie van zijn/haar kook competenties.

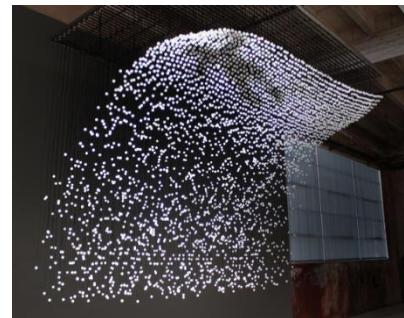
Door aan de medewerkers van Motion te vragen hoelang en wat voor bezoekers blijven zitten in de foyer kan ik mogelijk inzicht krijgen in wat de foyer zou kunnen missen. Hieruit kwam dat als er mensen gaan zitten ze er tussen een kwartier tot een uur blijven zitten. Soms drinken ze alleen koffie en gaan ze gelijk door. Dit zijn vooral ouders met kinderen en jongeren. Volgens de medewerkers is de reden dat er niet zoveel mensen komen zitten omdat de ruimte vaak leeg is en dit dan niet mensen uitnodigt om te gaan zitten. Ook verwachten ze niet dat ze bij Motion kunnen komen zitten en drinken.

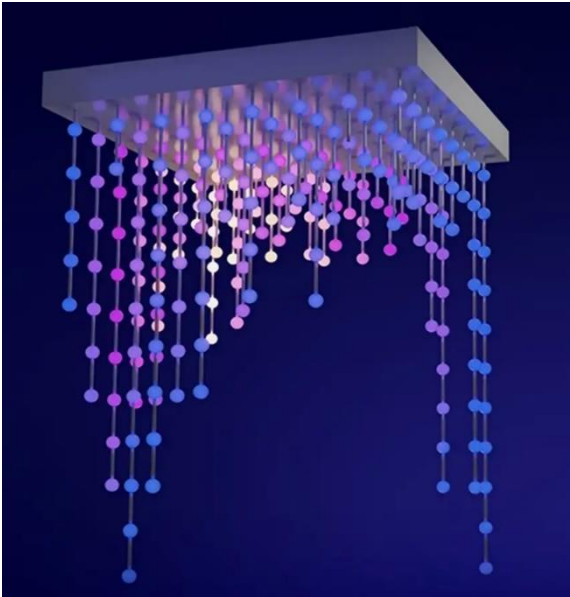
De foyer is niet zo populair als de andere ruimtes van Motion blijkt uit de antwoorden van de medewerkers. Met de opgedane informatie kan ik concluderen dat de foyer een meer uitnodigende uitstraling kan gebruiken. En om die uitstraling te zoeken ga ik hieronder inspiratie opdoen van bestaande installaties.

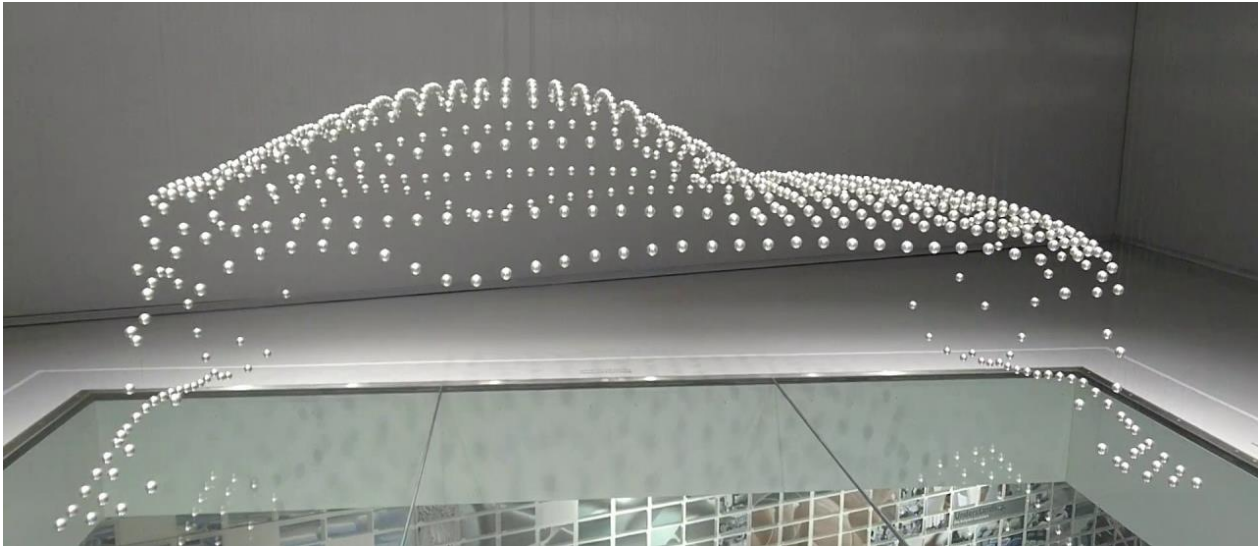
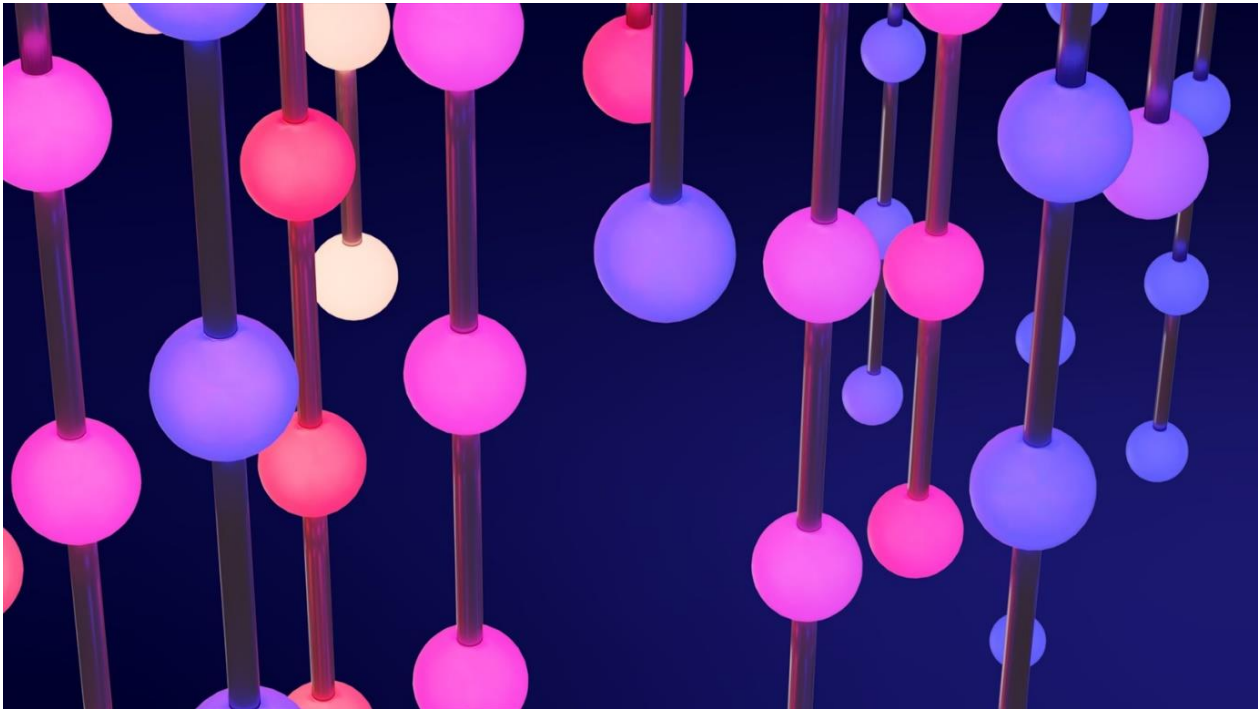
Inspiratie

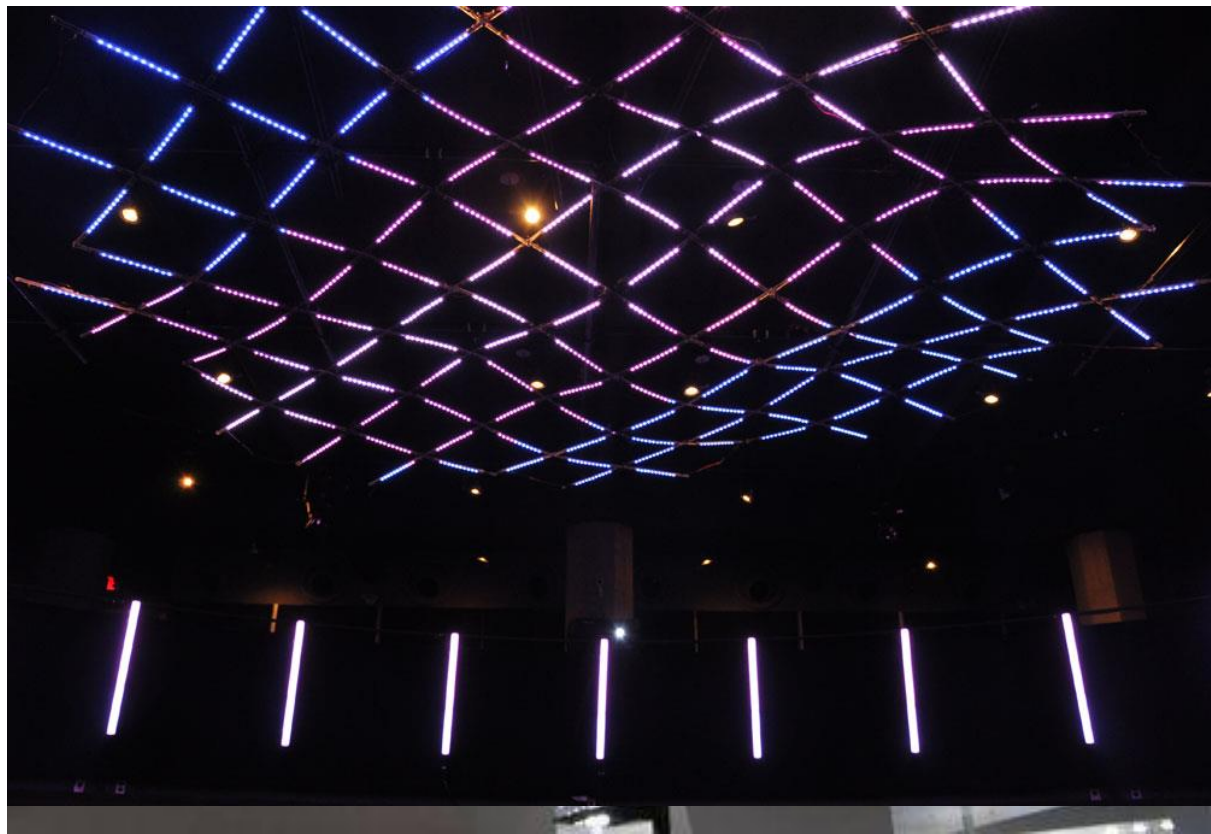
In het begin van de stage is mij gevraagd om een moodboard te maken van licht installaties die mij interessant leken, een foto hiervan was een installatie van lichtgevende bolletjes (hier rechts). Ik ben naar soortgelijke installaties gaan zoeken om zo een inzicht te krijgen hoe deze werken. Deze afbeeldingen hebben mijn verbeelding geactiveerd waarmee ik de 'ideation' fase begin.

Ik ben opzoek gegaan naar inspirerende voorbeelden om zo mijn eigen voorstellingsvermogen en verbeelding op gang te brengen. Hieronder een aantal voorbeelden van installaties die mij geïnspireerd hebben om mijn creatieve proces in gang te zetten.

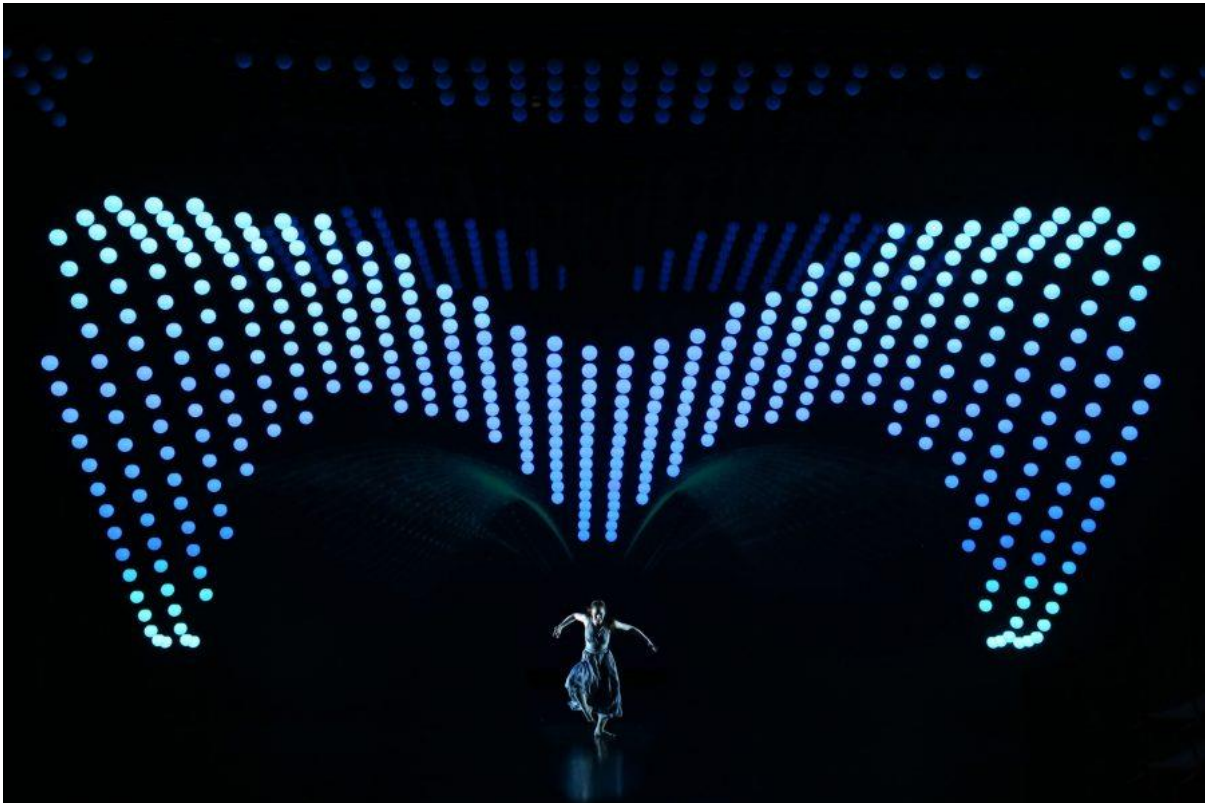














Een paar video's hebben mij ook geïnspireerd, deze heb ik hier onderstaan.

<https://www.whitevoid.com/living-sculpture-3d-module/>

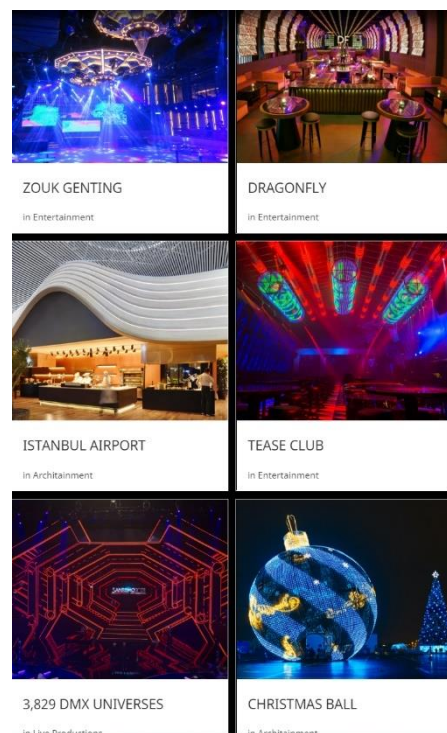
<https://www.stage-kinetik.de/en/>

<https://www.youtube.com/watch?v=PwXXqifJyEo>

<https://www.youtube.com/watch?v=3UocGg-prbA>

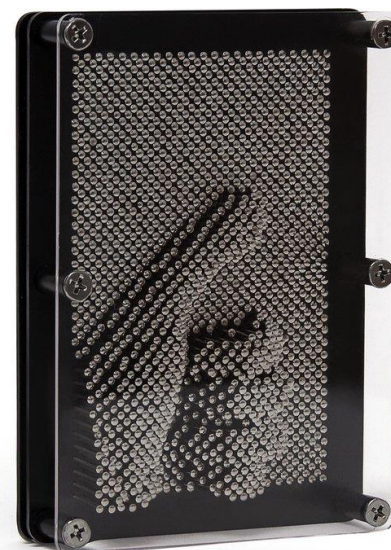
In de inspiratie fase heb gekeken naar soort gelijke installaties en wat deze gebruiken om dit aan te sturen. Tijdens het onderzoeken, kwam ik langs Madrix. Madrix is een bedrijf dat zich specialiseert in het aansturen van verlichting en is bekend in de evenementen industrie. Ze verkopen onder andere software om deze aansturing te bereiken. Dit wordt gebruikt voor vele grote evenementen zoals op de site beschreven. Typische klanten zijn; nachtclubs, hotels en vliegvelden. Omdat ik ook een aansturingsprogramma wil gaan schrijven is het voor mij gunstig om Madrix te onderzoeken en te kijken hoe zij dit aanpakken.

Op de site van Madrix stond een beschrijving van hoe de software werkt en hoe het technische aspect van de besturing in elkaar zit. Ik heb hier visuele inspiratie opgedaan voor de applicatie ik later in dit bestand heb gemaakt.

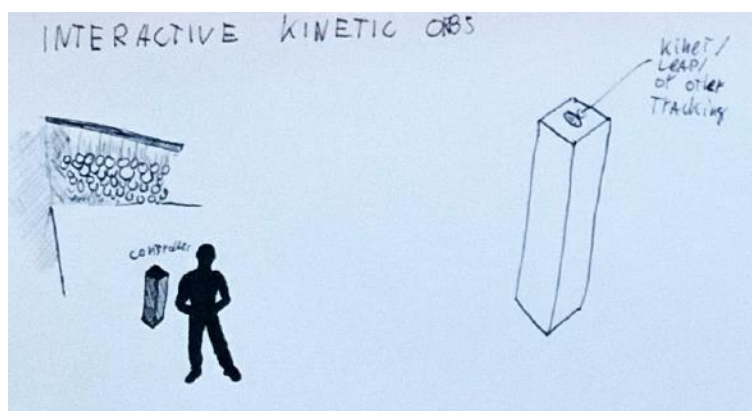
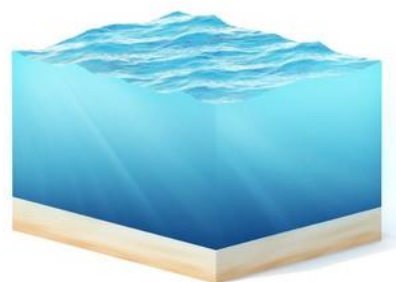


Tussen het inspiratie onderzoek door zijn Luca, Gijs en ik bij elkaar gaan zitten om ideeën te brainstorm. Dit kwam er toen uit. Kinetische licht installaties zijn niet iets nieuws maar interactiviteit is hiermee nog niet gecombineerd. Om deze interactiviteit te behalen heb ik onderzoek gedaan naar sensoren. Ik kwam langs de dieptecamera en tijdens het brainstormen met Luca en Gijs kwam ik met het idee op om een diepte camera te gebruiken als een soort spijkermat zoals hiernaast. De spijkers zijn de bollen en de hand, in dit voorbeeld, is de output van de dieptecamera. Dus je zou dan in principe je gezicht boven deze diepte camera kunnen houden en dit precies zo terug zien in het plafond.

Omdat ik hieronder praat over 'Unity', leg ik kort uit wat dit is. Unity is een game-engine bedoelt voor de ontwikkeling van computer spellen.



Een ander concept was om in Unity een soort bak met water te maken. Op het water drijven de bollen die ook in de foyer kunnen hangen. Deze bollen zijn gelimiteerd om alleen te mogen bewegen op de Y as, verticaal. Een andere sensor, zoals een Leap hier rechtsonder, die handbewegingen kan opnemen kan dan een virtuele hand besturen in Unity. Deze hand kan het water manipuleren en door hard met je hand op en neer te bewegen zou je dan golven kunnen maken waardoor de bollen ook gaan golven. Het idee hiervan is om de directe interactiviteit wat abstracter te maken, zodat je niet gelijk door hebt wat je precies doet. Hiermee hoop ik mensen nieuwsgierig te maken zodat ze langer bezig zijn met de installatie.



Conclusie

Met de inspiratie die ik heb opgedaan heb ik een beter beeld van hoe kinetische licht installaties werken.

Een 'kinetic lights' installatie maakt gebruik van lieren. Deze worden aangestuurd door een applicatie. De objecten die aan deze lieren hangen kunnen lichtgevende cirkels, staven of bollen zijn. Het gebruik van de bollen triggerde mij het meest want met deze vorm kun je veel andere vormen maken. Ook straalt de vorm van een losse bol op zichzelf al, naar mijn mening, neutraliteit uit. Door dan met de kleur van de bollen te spelen kan er een emotie of toon worden uitgestraald. Daarom heb besloten om bollen aan het plafond te hangen.

De beweging van een kinetische licht installatie maakt het juist zo uniek. En daar kan ook mee gespeeld worden. Daarom ga ik verder met het concept dat ik tijdens de inspiratie fase heb bedacht, het concept met de dieptecamera.

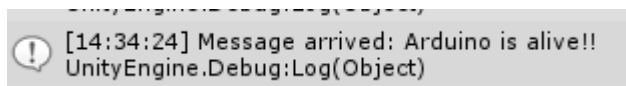
Experimenteren

Op dit moment heb ik genoeg inspiratie en voorbeelden om te beginnen met experimenteren en 'proof of concepts' maken. Ik ga verder met het dieptecamera concept, dat is bedacht in de inspiratie fase. Het plan is om dit hele systeem zelf te maken, dus ook de lieren waar de bolletjes aanhangen. Deze lieren wil ik vanaf een Arduino of iets dergelijks aansturen. Dus een programma dat de dieptecamera uitleest en dat dan doorstuurt naar een Arduino die de uitgelezen data uitvoert.

Ik heb gekozen om Unity te gebruiken omdat er makkelijk in geprogrammeerd kan worden met directe feedback. En omdat ik weet dat je via Unity ook kan communiceren met een Arduino. Unity is een game engine voornamelijk bedoeld om games in te creëren. Een UI dat makkelijk te gebruiken is en veel documentatie van is.

Mijn plan is om de ballen in 3D in Unity te plaatsen zoals ze in het echt zouden hangen. Deze ballen krijgen dan wat logica en regels, een bal mag dan bijvoorbeeld niet dichterbij dan 1cm van een ander object komen, mag niet hoger of lager dan een bepaald punt en kan alleen bewegen op de Y as. Op iedere frame stuurt Unity dan naar de Arduino op welke hoogte welke ballen moeten hangen. Met deze regels zou ik dus bijvoorbeeld een 3D model van een auto door de ballen slepen en dan heb je hetzelfde resultaat als de BMW-expositie (6^{de} plaatje in inspiratie). Dit 3D model kan dan gegenereerd worden van de output van de dieptecamera.

Om Unity te laten communiceren met een Arduino ben ik als eerst opzoek gegaan naar tutorials. Ik kwam uiteindelijk uit op 'Ardity', een Unity plugin waarmee je bidirectioneel kan communiceren via COM ports vanuit Unity. Na twee dagen van "trial and error" heb ik eindelijk een bericht van de Arduino in de Unity editor gekregen.



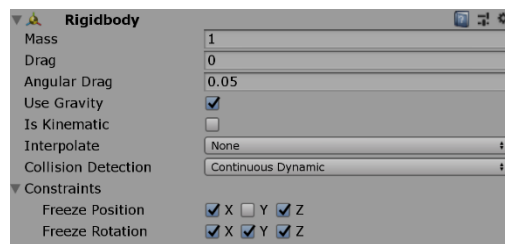
Deze tekst wordt vanuit de Arduino verstuurd dus er is een vorm van communicatie.

De volgende stap is een servo aansturen vanuit Unity, hiervoor heb ik alleen dit om te draaien, dus vanuit Unity naar de Arduino een bericht te sturen. Ik heb een 180 graden servo aan de Arduino aangesloten en stuur vanuit Unity een bericht met het aantal graden naar de Arduino. Ik liep tegen een paar problemen aan met het bewegen van de servo, dit kwam hoogstwaarschijnlijk door een benodigde stroomvoeding die ik niet gebruikt heb. Bij de servo zat een driver shield voor op de Arduino en dit gaf niet genoeg stroom voor de servo, ik heb een losse stroomvoorziening geprobeerd maar dit loste het probleem niet op.

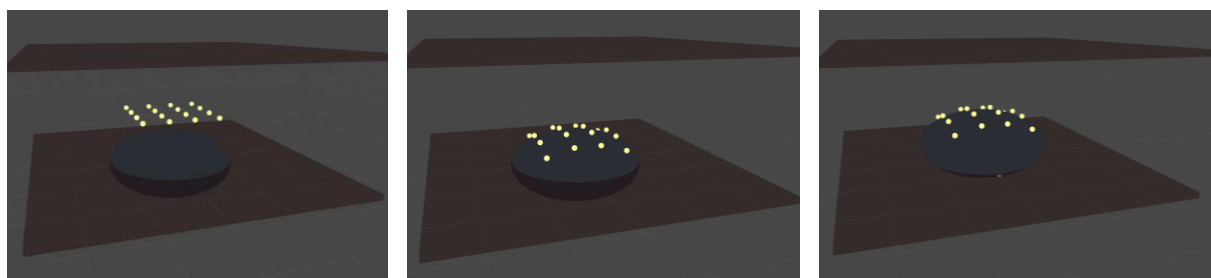
Alleen aansturing realiseren

Hierna heb ik met Tim gepraat over dit project en hoever ik ben gekomen tot nu. Tim vertelde dat maken van de bollen zelf en het systeem misschien te veel tijd zou kosten en dat het inkopen van deze bollen makkelijker zou zijn en goedkoper. Online worden deze bollen veel aangeboden en deze werken via DMX. Hiervoor kun je dan een externe software voor gebruiken, zoals Madrix, dat speciaal is gemaakt voor licht shows. Mijn doel van dit project is ook niet per se om dit allemaal zelf te maken, maar juist om de foyer interessanter te maken voor bezoekers. Ik kan me dan beter focussen op het schrijven van de software hiervoor en hoe ik bepaalde technieken het beste kan toepassen om dit einddoel te behalen. Door zelf de software te schrijven krijg ik meer vrijheid in de toepassing van deze DMX bollen. Deze software moet makkelijk zijn om te gebruiken en er moet makkelijk een nieuwe show in worden gemaakt.

Om te beginnen met Unity en het manipuleren van bollen heb ik een klein 'grid' gemaakt. Het plan hiermee is een plaat eronder en erboven te plaatsen die het plafond en de grond moeten voorstellen. De bollen hebben een 'Rigidbody' waardoor ze door ze met een variabele zwaartekracht naar de grond willen vallen. En ook zijn er bepaalde posities en rotaties vergrendeld zodat de bollen alleen op de Y as kunnen bewegen, of te wel alleen naar boven en beneden.



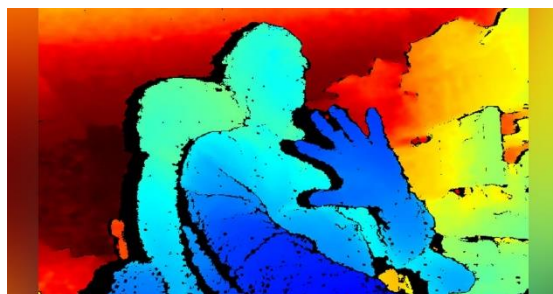
Door een object onder dit 'grid' te plaatsen vallen dan de bollen hierop door de 'Rigidbody'. Als test gebruik ik een eivormige bol.



Video uitlezen vanuit Unity

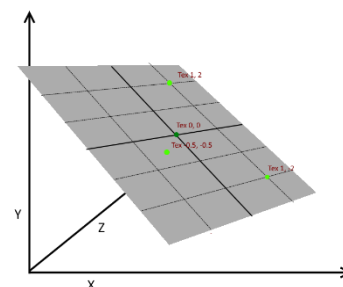
Tijdens het testen hiervan kwam ik al snel tegen wat problemen zoals de moeilijkheid van het maken van een show, omdat dit systeem erg fragiel is. Bijvoorbeeld het omhoog bewegen van de eivormige bol niet altijd de bollen mee omhoog neemt, soms vallen ze erdoorheen. Om dit op te lossen zou ik in de logica van de 'Rigidbody' moeten kijken of eventueel zelf deze logica specifiek voor deze toepassing schrijven. Omdat er nog een andere problemen zijn met dit programma zoals het verplaatsen van een object op de X en Z as en dat dit niet de bollen omhoog beweegt, denk ik dat dit niet de juiste aanpak is.

Na verder te brainstormen, moest ik denken aan hoe een dieptecamera nou eigenlijk werkt. De output van een dieptecamera gebruikt kleur om de diepte te uiten. Lichtblauw is bijvoorbeeld dichterbij dan donkerrood. Dit kan ik ook gebruiken met grijswaardes, wit is volledig naar beneden en zwart is volledig naar boven. Als ik zo'n diepte video kan gebruiken om de bollen te manipuleren kan er makkelijk nieuwe content voor worden gemaakt.



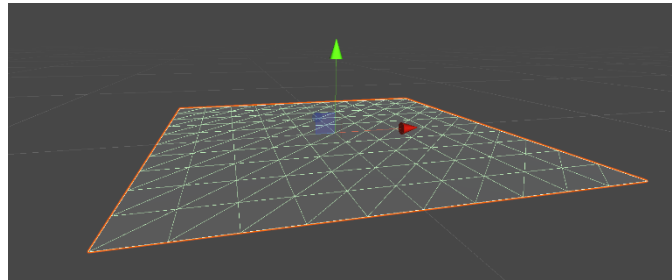
Een video zou dan de hoogte kunnen bepalen en de andere de kleur waardes van de bollen. Omdat er in Unity een soort gelijk iets wordt gedaan met UV/ hoogtemap om diepte te krijgen in bijvoorbeeld tegels van straten in spellen. Dan zou ik deze techniek kunnen toepassen door een hoogtemap video als input en de bollen in Unity als output kunnen gebruiken.

Van Gijs heb ik gehoord dat dit mogelijk via Unity's Shader Graph kan werken. Omdat ik niet bekend ben met Shader Graph heb ik eerst een paar tutorials gevolgd. Hierin heb ik geleerd hoe je vertices



(vertices/vertex zijn hoekpunten van bijvoorbeeld een kubus, hier rechts zijn dat ook de hoeken van de vierkanten) kan manipuleren via een formule.

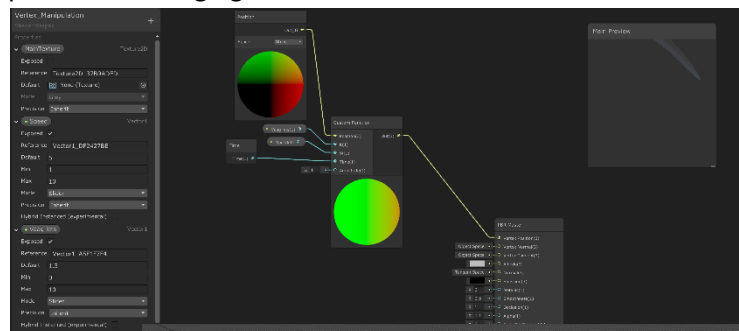
Ik gebruik een 'plane' met vierkanten (of wel 'faces') erin, iedere vertex van deze vierkanten symboliseert een bol. Hiermee wil ik deze 'plane' animeren met de Shader Graph, iedere keer dat het programma dit update moet de bol de locatie van zijn vertex volgen. En dan kan de bol zijn locatie doorsturen naar de daadwerkelijke hangende bol via Art-Net of DMX.



Ik begon met een Shader Graph tutorial waar je een vlag leert animeren. Hier rechts staat een afbeelding hoe het resultaat van de tutorial eruit zou moeten zien. Dit sluit goed aan met wat ik wil proberen dus daarom heb ik deze tutorial gekozen.

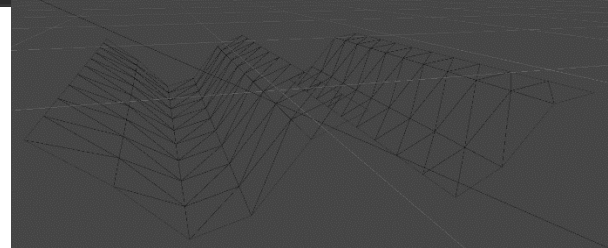


De meeste nodes heb ik overgenomen, een paar heb ik achterwegen gelaten zodat ik een simpele op en neer beweging heb. Hier rechts is de Shader Graph editor te zien met de gebruikte nodes. De Tutorial die ik heb gebruikt is van codinBlack:



<https://www.codinblack.com/vertex-manipulation-using-shader-graph-in-unity3d/>

De shader werkt en de 'plane' animeerd in een golvende beweging. De volgende stap is om aan de vertices bollen te hangen. Ik heb een klein script geschreven om de bollen te spawnen.



Hieronder de code om de bollen te spawnen op de vertex posities:

```
void Start()
{
    MeshFilter mf = gameObject.GetComponent<MeshFilter>();
    Matrix4x4 localToWorld = transform.localToWorldMatrix;

    for (int i = 0; i < mf.mesh.vertices.Length; ++i)
    {
        Vector3 world_v = localToWorld.MultiplyPoint3x4(mf.mesh.vertices[i]);
        Instantiate(OrbObject, world_v, Quaternion.identity);
        Debug.Log("We did it!");
    }
}
```

```
}

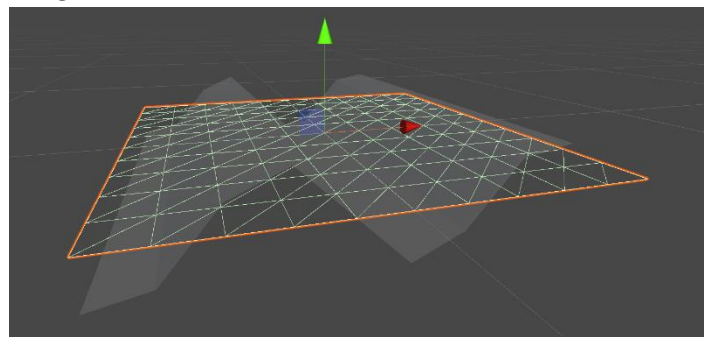
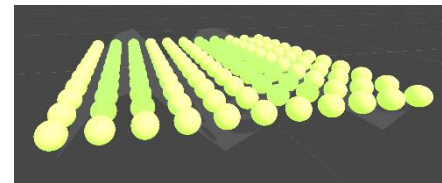
```

En hieronder de code om ze per update te verplaatsen:

```
void Update()
{
    MeshFilter mf = gameObject.GetComponent<MeshFilter>();
    Matrix4x4 localToWorld = transform.localToWorldMatrix;

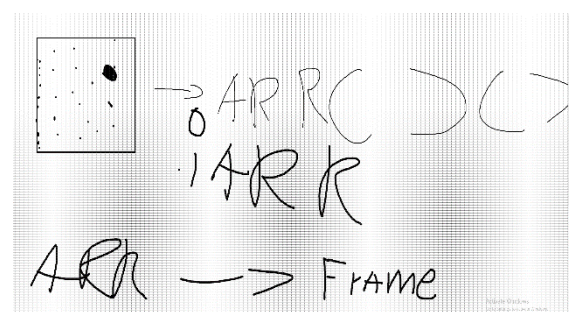
    for (int i = 0; i < mf.mesh.vertices.Length; ++i)
    {
        Vector3 world_v = localToWorld.MultiplyPoint3x4(mf.mesh.vertices[i]);
        test = localToWorld.MultiplyPoint3x4(mf.mesh.vertices[0]);
    }
    Debug.Log(test);
}
```

De bollen worden op de juiste positie geplaatst zoals te zien is in de afbeelding rechts. Echter bewegen de bollen niet mee, ik dacht dat er een fout in de code zat dus met Debug.Log log ik de positie van de eerste vertex om te kijken wat de locatie hiervan is. Maar de locatie blijft hetzelfde. Na veel googelen en het vragen aan mensen met meer ervaring met Unity, kwam ik erachten dat de Shader Graph niet de fysieke vertices kan manipuleren maar alleen een visueel effect creëert. Zoals hiernaast te zien is beweegt de 'plane' wel maar in de 'wireframe view' zie je dat de 'plane' zelf nog op de originele positie is. Dus ook deze aanpak is niet de juiste.

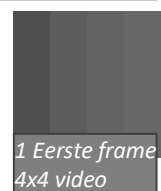


Eigen applicatie schrijven

Na weer wat brainstormen bedacht ik me dat ik dit via C# beter kan doen. Als ik een programma schrijf dat per frame van een video iedere pixel bekijkt en de kleur hiervan opslaat, dan heb ik een doel al bereikt. Ik weet dat je via C# makkelijk ieder pixel van een afbeelding kan uitlezen, dus het uitlezen van een video zou ook niet lastig moeten zijn. En eventueel zou de video in plaats van een AVI of MP4 als 'image sequence' (een reeks foto's) moeten worden geëxporteerd. Zo heb ik dit eerst geprobeerd, ik heb een zwart wit video van Gijs gekregen dat een resolutie heeft van 4 bij 4 pixels. Deze video heb ik omgezet naar een 'image sequence' en heb een kort programma geschreven dat door alle foto's in een bepaalde map looped. Het programma checkt per pixel de kleurwaarde en laat dit zien in een log. Hieronder is het log te zien.



2 Schets gebruik van een multidimensionale array




```
row 4 column 1 height:140 color:Color [A=255, R=151, G=0, B=255]
row 4 column 2 height:138 color:Color [A=255, R=149, G=0, B=255]
row 4 column 3 height:136 color:Color [A=255, R=148, G=0, B=255]
row 4 column 4 height:135 color:Color [A=255, R=147, G=1, B=254]
row 4 column 5 height:135 color:Color [A=255, R=147, G=1, B=254]
row 4 column 6 height:134 color:Color [A=255, R=145, G=0, B=254]
row 4 column 7 height:132 color:Color [A=255, R=143, G=0, B=253]
row 4 column 8 height:129 color:Color [A=255, R=142, G=0, B=252]
row 4 column 9 height:129 color:Color [A=255, R=144, G=1, B=253]
row 4 column 10 height:128 color:Color [A=255, R=143, G=1, B=254]
row 4 column 11 height:127 color:Color [A=255, R=142, G=2, B=254]
row 4 column 12 height:127 color:Color [A=255, R=140, G=1, B=252]
row 4 column 13 height:125 color:Color [A=255, R=140, G=1, B=252]
row 4 column 14 height:125 color:Color [A=255, R=140, G=0, B=253]
row 4 column 15 height:128 color:Color [A=255, R=139, G=1, B=252]
row 5 column 0 height:141 color:Color [A=255, R=151, G=0, B=255]
row 5 column 1 height:139 color:Color [A=255, R=150, G=1, B=255]
row 5 column 2 height:138 color:Color [A=255, R=148, G=0, B=255]
row 5 column 3 height:136 color:Color [A=255, R=148, G=0, B=255]
row 5 column 4 height:134 color:Color [A=255, R=147, G=0, B=255]
row 5 column 5 height:132 color:Color [A=255, R=146, G=1, B=255]
row 5 column 6 height:132 color:Color [A=255, R=143, G=0, B=253]
row 5 column 7 height:131 color:Color [A=255, R=143, G=0, B=253]
row 5 column 8 height:128 color:Color [A=255, R=142, G=0, B=252]
row 5 column 9 height:128 color:Color [A=255, R=143, G=1, B=254]
row 5 column 10 height:127 color:Color [A=255, R=142, G=1, B=254]
row 5 column 11 height:126 color:Color [A=255, R=142, G=1, B=254]
row 5 column 12 height:127 color:Color [A=255, R=139, G=1, B=252]
row 5 column 13 height:124 color:Color [A=255, R=139, G=2, B=252]
```

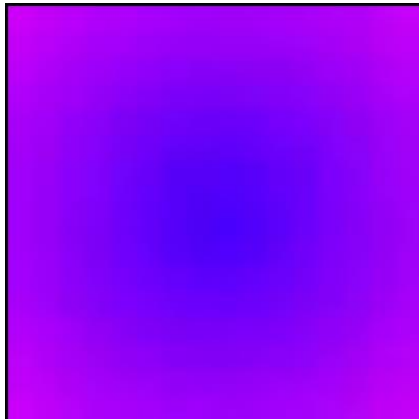
Omdat interactiviteit nog een mogelijke route is voor dit project wil ik de video uitlezen en live verwerken en door sturen naar de bollen. Hierom moet het uitlezen van de video niet te lang duren, aangezien er dan een vertraging zit tussen de frames en dit dan de afspeelsnelheid kan beïnvloeden. Als een frame er langer over doet dan de andere kan dit ook ongelijk ogen tijdens de show. Om het afspelen geleidelijk te laten verlopen moet het programma een frame rate aanhouden. Dit kan worden gedaan door te meten hoeveel tijd de huidige frame heeft gekost om uit te lezen en deze tijd te vergelijken met de bedoelde tijdspan. Als de gemeten tijd lager is moet het programma de resterende tijd wachten. Om inzicht te krijgen van de gemeten frame-looptijd, meet ik bij iedere frame hoelang dit duurt. Dit voeg ik allemaal toe aan een lijst en kan zo het gemiddelde meten, de snelste en de langzaamste frames. Ook heb ik wat andere informatie weergeven.

Ik heb berekend dat voor een video van 25fps, een frame 40 milliseconde heeft. Zoals hier rechts te zien is, komt dit er minstens een keer bovenuit. En het gemiddelde is net onder het doel. Van Luca heb ik gehoord dat dit mogelijk via CUDA sneller kan, dit is een technologie van Nvidia bedoeld voor programmeurs om functies op de grafische kaart te laten uitvoeren waardoor dit sneller gaat.

```
Depth video:
Codec: H264
Framerate: 25
Height: 16
Width: 16
Total frames: 100
Total pixels: 25600
Original video duration: 4 seconds

Color video:
Codec: H264
Framerate: 25
Height: 16
Width: 16
Total frames: 100
Total pixels: 25600
Original video duration: 4 seconds

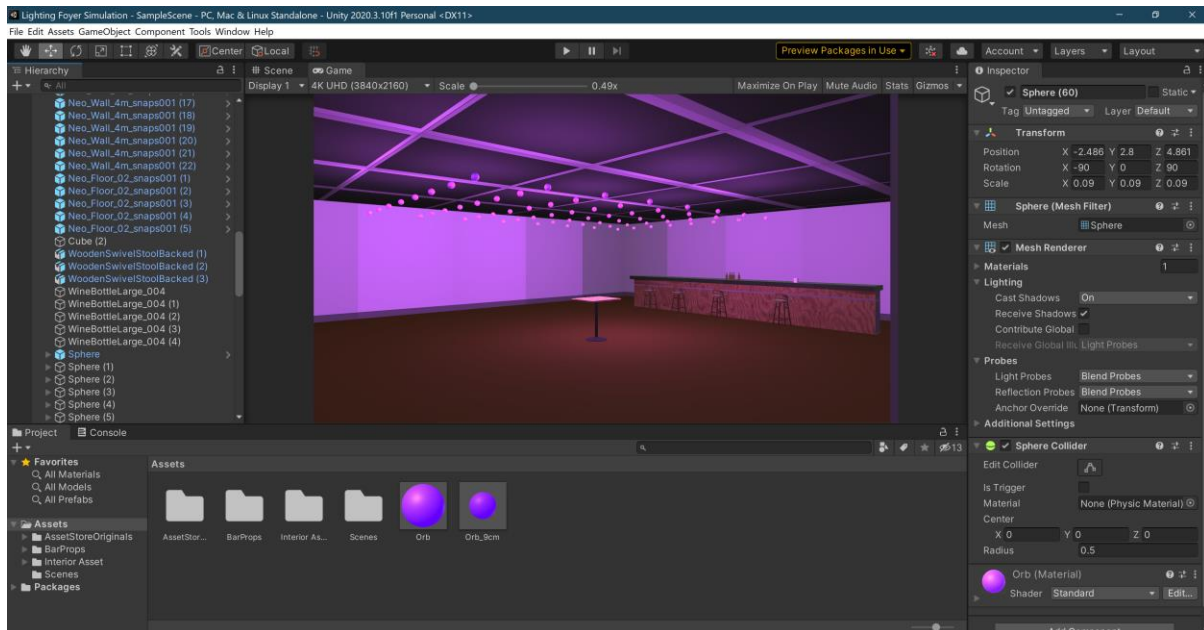
Average frame run time: 38.13 ms
Slowest frame run time: 110 ms
Fastest frame run time: 6 ms
Target frame run time: <40 ms
RunTime 00:00:04.20
```



3 Frame kleur video 32x32

Unity simulatie van foyer

Omdat Bram zei dat hij wat meer visuele uitkomsten wilde zien heb ik de foyer nagemaakt om daarmee te laten zien hoe vet deze kinetische bollen kunnen zijn. Dit heb ik via Unity gedaan omdat ik bekend ben met de software, bij nader inzien was Blender of een specifiek 3D programma beter geweest. In Unity kun je namelijk niet makkelijk alles op een bepaalde schaal brengen. Hieronder de uitkomst.



C# applicatie ontwerp

De C# applicatie moet worden gebruikt door het personeel dus daarom moet het design makkelijk zijn om te gebruiken en overzichtelijk. Ik begon eerst met de eisen van de applicatie, dit heb ik ook aan Gijs gevraagd. Eigenlijk een basis CRUD-systeem; Create, Read, Update en Delete. Je moet bijvoorbeeld een playlist kunnen aanmaken en daarin een animatie toevoegen.

Playlist

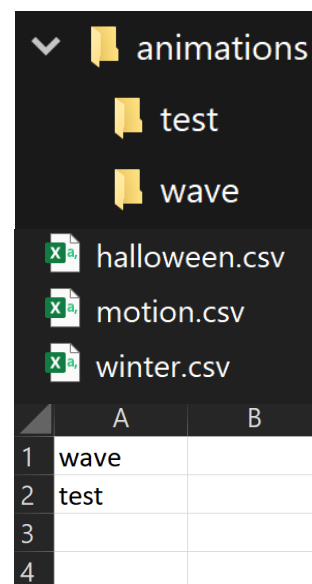
- Naam
- Animaties

Animatie

- Naam

Het aanmaken van een SQL-database zou een optie zijn om deze data op te slaan maar ik heb besloten om dit via een mappen structuur te doen zodat dit makkelijk te lezen is voor iedereen die bekend is met een computer. Ook is het aanmaken van een SQL-database extra werk bij het installeren van de applicatie op een nieuwe computer en een extra ding dat fout kan gaan.

Iedere animatie is map in de map 'animations', zo hoeft de applicatie hoeft alleen de locatie van de 'animations' map te weten. Ook is er een 'playlists' map waarin alle playlists staan als CSV-document, ik heb gekozen voor CSV omdat dit vergelijkbaar is met een SQL-database en dit makkelijk te lezen is met C#. In een CSV-document scheid je data met komma's, in een Excel bestand is dit per vakje. Deze data kunnen in een C# opgehaald worden als een array. Via een foreach loop ga ik door iedere lijn en kan ik de eerste van de array selecteren (`String animationName = columns[0]`). Wat in dit geval de naam van de animatie is, op deze manier kan er meer informatie mee gestuurd worden.

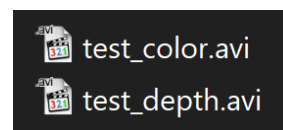


```

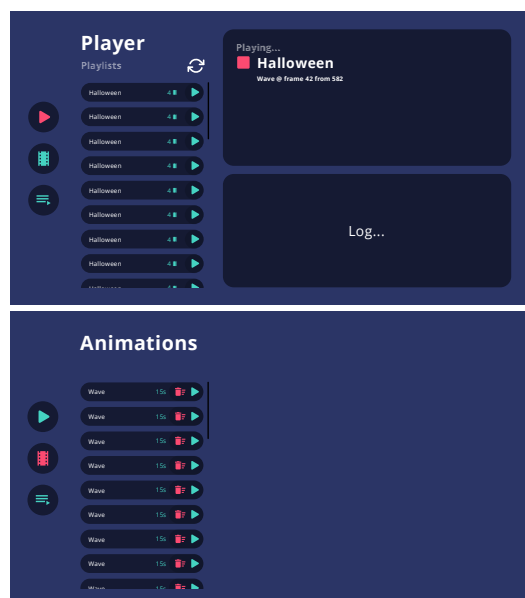
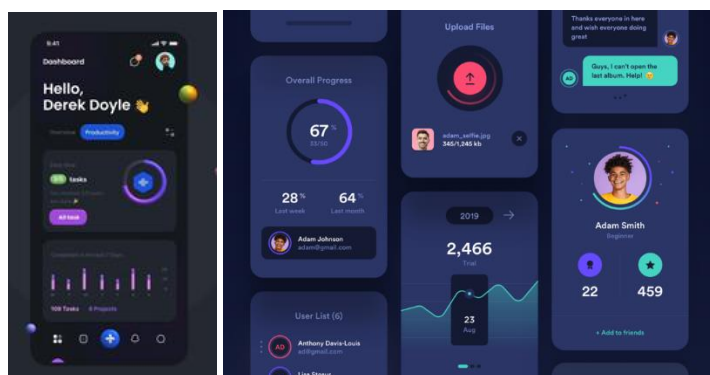
List<String> notFoundVideos = new List<String>();
string[] lines = System.IO.File.ReadAllLines(playlistFolder + "/" + playlist + ".csv");
foreach (string line in lines)
{
    string[] columns = line.Split(',');
    String animationName = columns[0];

    if (!File.Exists(animationFolder + "/" + animationName + "/" + animationName + "_depth.avi"))
    {
        notFoundVideos.Add(animationFolder + "/" + animationName + "/" + animationName + "_depth.avi" + enter);
    }
    if (!File.Exists(animationFolder + "/" + animationName + "/" + animationName + "_color.avi"))
    {
        notFoundVideos.Add(animationFolder + "/" + animationName + "/" + animationName + "_color.avi" + enter);
    }
}
if (notFoundVideos.Count > 0)
{
    this.logBox2.AppendText("The following videos could not be found: " + enter + string.Join(enter, notFoundVideos)+enter);
    return false;
}
else
{
    return true;
}
    
```

Hierboven staat een screenshot van de functie die checkt of de animaties in de gegeven playlist bestaan. De variabele 'playlistFolder' en 'animationFolder' zijn globaal en door de bekende mappenstructuur kan de applicatie zo bij de animaties en playlists. Animatie map bevat twee video's, de kleuren video en de diepte video. Deze wordt voorgevoegd door de animatie naam, zoals in de code te zien wordt zo de video opgehaald.



Hierna had ik het design gemaakt in Adobe XD. Hier rechts het gemaakte design, ik wilde gaan voor een 'dark theme' omdat het rustig is voor de ogen en ik dit persoonlijk mooi vind. Online heb ik wat referentie foto's gezocht.



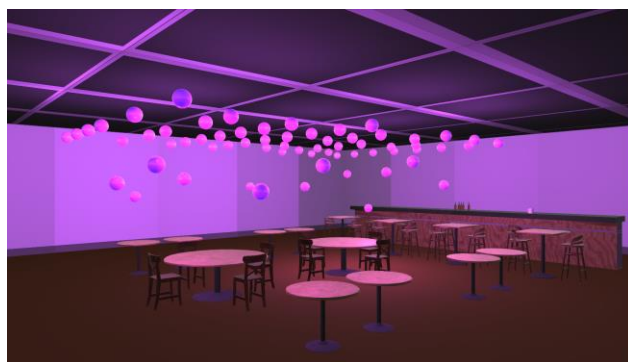
Dit design heb ik voorgelegd bij Gijs en hij was er blij mee. Aangezien Gijs voorlopig de enige gebruiker van deze applicatie zal zijn, ben ik toen begonnen met het realiseren van dit design in Visual Studio.



Foyer simulatie als test toepassing

Omdat ik op dit moment niet kan testen met echte bewegende lichtgevende bollen heb ik nagedacht aan een mogelijkheid om met de applicatie niet alleen de bollen aan te sturen maar ook de foyer simulatie in Unity. Omdat ik weet dat ik via een Arduino met Unity kan praten ben ik ervan overtuigd dat dit vanaf een andere C# applicatie ook kan. Het makkelijkst en het meest logisch zou zijn om de applicatie als script in Unity zelf te plaatsen en dit heb ik geprobeerd. Echter zijn de 'packages' van Accord niet beschikbaar voor Unity. Accord.NET is een 'machine learning' .NET framework geschreven in C#, waaronder image processing. Image/video processing wordt bijvoorbeeld ook gebruikt bij het maken van filters en kan foto's/video's per pixel manipuleren. Zonder deze framework werkt mijn programma niet dus het gebruiken als script in Unity gaat niet werken.

Door de Unity als simulatie te gebruiken kan ik al testen met welke animaties wel of niet werken. Zo kan er zonder de hardware al in gezien worden hoe het er ongeveer in werkelijkheid uit gaat zien.



4 Foyer simulatie

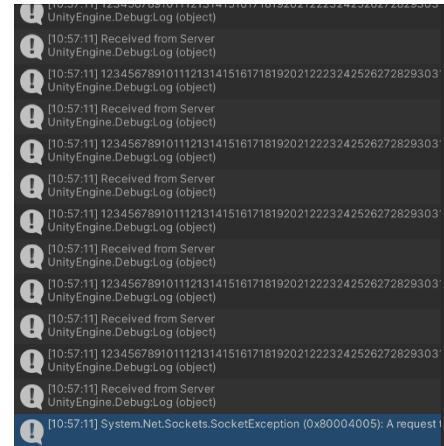


5 Foyer tijdens "winter wonderland"

Unity & C# connectie

Online heb ik onderzoek gedaan hoe er vanaf een losstaand C# programma met Unity kan worden gecommuniceerd. Ik kwam veel pagina's tegen die verwijzen naar het gebruik van 'sockets', hiermee kan er zelfs van twee verschillende computers worden gecommuniceerd. Deze communicatie gaat tussen een 'client' programma en een 'server' programma. Unity zal de server zijn en de C# applicatie de client, de client stuurt steeds data naar de server. Deze data zijn de posities en kleuren van de bollen, de server vangt dit dus op en moet dit verdelen op de bollen.

Online heb ik een script gevonden dat een socket server aanmaakt in C#, ook heb ik een socket client script gevonden. Dit koste flink wat tijd om aan de praat te krijgen aangezien ik geen ervaring heb met sockets. Zoals hier rechts te zien is, krijgt Unity van alles binnen. Voor het testen verstuur ik iedere frame het nummer van het frame door. Dus hier rechts zou eigenlijk steeds 1, 2, 3 etc. moeten komen staan. Maar zoals er te zien is, gebeurt dit niet. De nummering is goed alleen ziet Unity deze niet los per bericht. Ik las de code opnieuw en zag dat de socket server alle data bij elkaar voegt tot dat er "<EOF>" in de tekst staat en dit iedere milliseconde in de console schrijft (het console scherm is het beeld hier rechts). "<EOF>" is een afkorting voor "End Of File". In dit voorbeeld gaan ze ervan uit dat er aan het einde van het bericht "<EOF>" staat, dit is blijkbaar (na onlineonderzoek) normaal bij sockets omdat het programma nooit zelf kan weten wanneer het hele bericht binnen is.



Ik dacht dat dit makkelijk op te lossen was door "<EOF>" achter ieder bericht te zetten maar dit loste het niet gelijk op. Er kwam een nieuw probleem wat me wat **1<EOF>2<EOF>** tijd heeft gekost om te begrijpen. Ieder bericht dat wordt verstuurd wordt eerst omgezet naar een byte en dan doorgestuurd, een byte is een array, of te wel reeks, van bits. Daarom wordt dit in het voorbeeld bij elkaar opgeteld tot er "<EOF>" in het bericht staat en daarna weer omgezet van een byte array/reeks naar leesbare tekst. Omdat ik iedere 40 milliseconden een lang bericht stuur, denk ik dat het programma de volgende data ook al binnenkrijgt. Dit is lastig uit te leggen, maar hier rechts een afbeelding van wat ik denk dat het programma binnen krijgt. De tekst wordt denk ik verkort en zo staat er bijvoorbeeld een deel van het volgende bericht in het huidige bericht.

inkomendedata<EOF>inkome
ndedata<EOF>in

Toen ik het programma starten op mijn andere pc had ik hier om een of andere reden geen last meer van. En ik weet niet waarom. Het werkte uiteindelijk ook op mijn laptop.

Toen ik de communicatie tussen het C# programma en Unity had afgerond. Kon ik mijn eerste echte visuele test doen. Hier onder het resultaat.

https://youtu.be/S8pIW1TFt_8

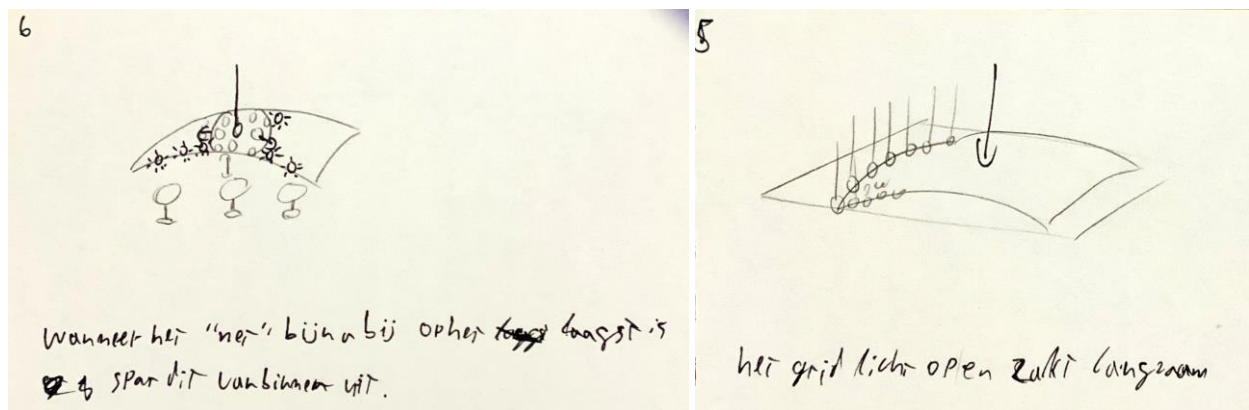
Het was een heerlijk gevoel dat het programma werkt zoals ik wilde, maar snel zag ik een mogelijk probleem. De video die gelezen wordt heeft een soort golf van binnenuit en ik zie dat niet terug in de simulatie. Het kan natuurlijk zijn dat de grijswaarde niet ver genoeg uit elkaar zitten en dat het daarom niet zichtbaar is in de simulatie. Daarom heb ik Gijs gevraagd om een video te maken waar het verschil tussen zwart en wit wat groter is zodat ik dit terug kan zien in de simulatie.

<https://youtu.be/kTa-1GZBXpM>

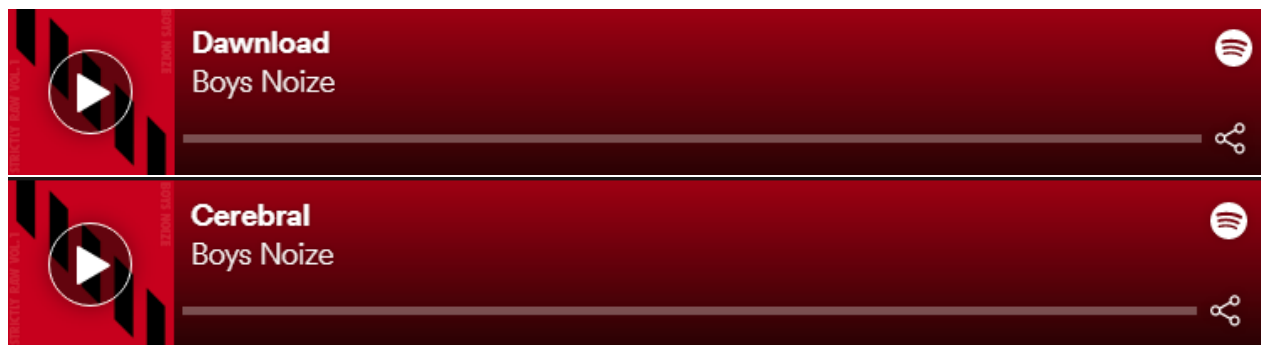
Concepting

Om een show te concepten heb ik Gijs gevraagd hoe ik dit het beste kan aanpakken. Gijs heeft mij uitgelegd hoe hij dit creatieve proces oppakt. Hij begint met een storyboard en schetst daarmee een geanimeerd verhaal. Hierbij gebruikt Gijs muziek die hij vindt passen bij dit verhaal.

Deze aanpak ga ik ook proberen te gebruiken bij het bedenken van de show. Ik begin met storyboards en ga zoveel mogelijk opschrijven. Normaal blokkeert een witte pagina mijn gedachtegang en bekritiseer ik mijn ideeën voor ik ze kan opschrijven, dus dit is ook een goede oefening om de openheid van mijn creativiteit te vergroten.



Het voelde fijn om bijpassende muziek te gebruiken tijdens het storyboarden omdat ik me al beter kon voorstellen hoe mijn ideeën eruit konden zien. Vooral de volgende twee tracks hebben me geïnspireerd.

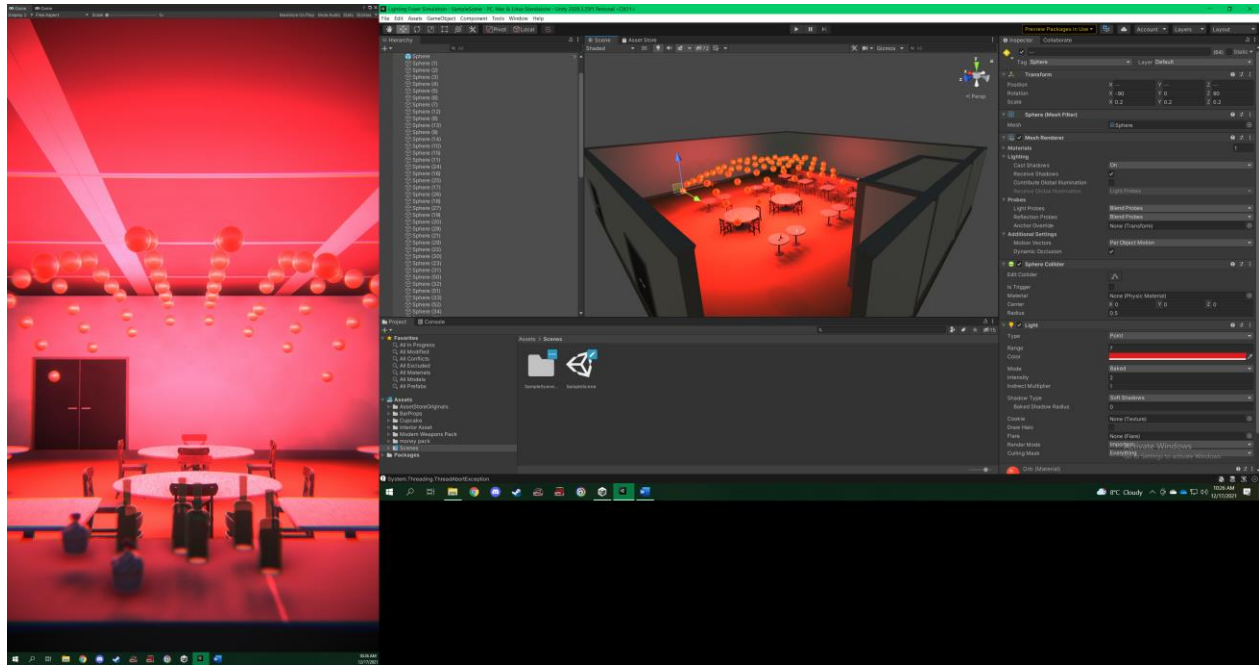


Moving storyboard




Nu ik voor een groot deel klaar ben met het maken van mijn geschetste storyboard kan ik beginnen met een "moving storyboard". De in Unity gemaakte simulatie van de foyer kan ik hier goed voor gebruiken. Door de schetsen uit te werken in deze simulatie kan er al een betere 'feeling' komen van hoe het er uiteindelijk uit gaat zien.




Gijs heeft mij wat tips gegeven over camera-instellingen voor Unity, via de Post-Processing van Unity kan ik de camera's filters geven om zo een beter effect te creëren. Unity heeft een eigen documentatie over licht en die heb ik gebruikt om het effect van het licht te verbeteren.

https://docs.unity3d.com/Manual/Lighting.html?_ga=2.128639116.1203230437.1639657670-1206786857.1605258171



Wegens wat tijd en software gebrek heb ik helaas niet een bewegend storyboard kunnen maken. Hieronder heb ik de tot zoverre uitgewerkte situaties in volgorde neergezet met een uitleg.

	<p>In het begin hangen de bollen boven aan het plafond.</p>
	<p>Dan zakker er een paar bollen naar beneden als een soort sprankelende sterren.</p>
	<p>Ze zakken steeds lager en sterven dan af (door uit te gaan).</p>

	<p>Alle bollen gaan weer naar boven.</p>
	<p>Alle bollen gaan tegelijk aan met een beat.</p>
	<p>Dan gaan ze allemaal uit en zakken ze een stuk.</p>



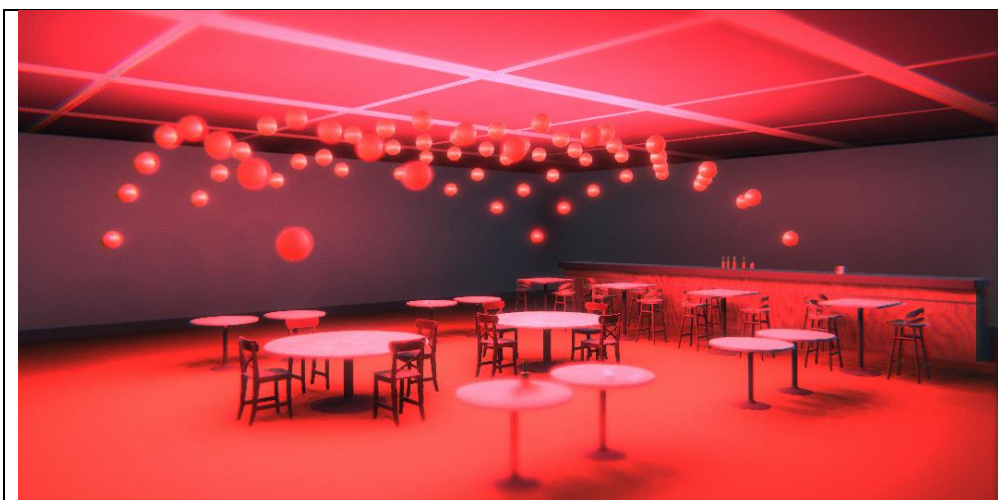
Weer met een beat gaan alle bollen aan maar in een andere kleur.



En weer uit en zakken een stuk.



Alle bollen gaan weer aan op een beat met een andere kleur.






De bollen blijven aan en veranderen langzaam van kleur en de hoeken zakken langzaam lager zodat het een soort net vormt.



Het midden sterft af en de rest blijft langzaam doorzakken.



Het midden dat afgestorven is gaat weer naar boven. Dit 'midden' verspreid zich langzaam en steeds meer bollen sterven af.

	<p>Dit 'verspreiden' versneld een beetje en hierna zijn alle bollen dan afgestorven en naar boven.</p>
	<p>In combinatie met stormachtige muziek maken de bollen bliksem animaties.</p>
	<p>Twee bollen zakken snel naar beneden.</p>



Een bol daarnaast gaat ook mee en dan wisselen ze om. Zodat dit horizontale beweging imiteert



Deze bollen gaan ook snel naar beneden.

Realisatie

Om te testen of het concept van het aansturen van de bollen kan lukken heb ik met Bram besloten om vanuit Motion Experience een bol en een lier te bestellen.

Nu de bol eindelijk binnen is kan ik hiermee testen. Tim heeft een achtergrond in de evenementen wereld en dus ook kennis van DMX en Art-Net. Hij stelde voor om simpel te beginnen, zodat ik eerst kan begrijpen hoe de bol werkt. Tim heeft een licht tafel gepakt en aangesloten en ik heb de bol en lier opgehangen. Met het meegeleverde boekje kon ik makkelijk inzicht krijgen hoe de kanalen werken.



DMX512

“DMX512 (ook wel DMX genoemd) wordt gebruikt als seriële communicatiebus tussen verschillende apparaten in de theater- en lichttechniek. Vaak is het sturende apparaat een lichttafel met enkele dimmerpacks of andere meer ingewikkelde armaturen. Elk apparaat heeft een ingang en een uitgang zodat de bus tussen de verschillende apparaten doorgelust kan worden.”

“Het protocol zelf is bijzonder eenvoudig: na een korte wachttijd worden tot maximaal 512 8-bits kanaalwaarden achter elkaar verzonden. Er kunnen dus 512 dimmerkanalen aangestuurd worden met één DMX-kabel, waarbij elke dimmer tussen de 0 en 255 geregeld kan worden. Daarbij is een waarde van 255 gelijk aan een volledig ingeschakelde lamp en 0 gelijk aan een uitgeschakelde lamp. Op elk dimmerpack kan het adres waar het naar luistert ingesteld worden, het ontvangende apparaat hoeft na het detecteren van de zendpauze alleen maar af te tellen tot zijn waarde langskomt. Op de verschillende apparatuur die op DMX aangesloten kunnen worden is door middel van een menu, RDM, via dip- of rotary switches het desgewenste startadres in te stellen. Men dient daarbij te letten op het feit dat de adressen elkaar niet mogen overlappen als een apparaat meerdere parameters heeft.”

DMX512. (2021, 14 april). In Wikipedia. <https://nl.wikipedia.org/wiki/DMX512>

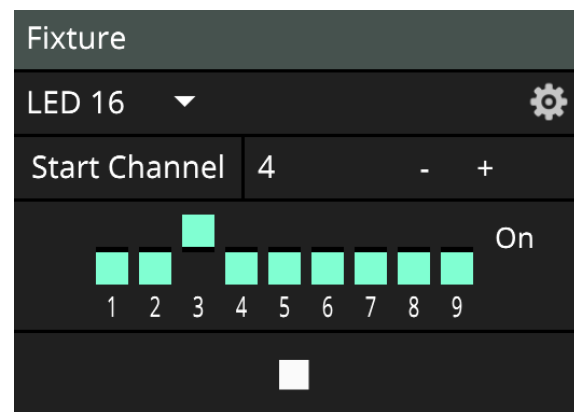
Dus via DMX kunnen er 512 kanalen worden aangestuurd en deze kanalen kun je waardes geven tussen 0 en 255. De lichttafel heeft een schuifbalkje dat dus van 0 tot 255.

Door hiermee te experimenteren begrijp ik hoe de bol werkt. Hierna ben ik begonnen met uitzoeken hoe ik deze bol kan besturen vanaf mijn laptop.

Tim heeft het eerder over een Art-Net naar DMX-kastje gehad en ik heb hem gevraagd of ik die kan gebruiken. Tim heeft deze van een andere installatie gepakt aangezien motion nu gesloten is in verband met de coronamaatregelen. Dit kastje sluit je aan via een netwerkkabel en kan twee DMX-poorten gebruiken als in- en output. In mijn geval is dat output, omdat ik alleen data wil sturen. In het begin dacht ik dat dit makkelijk zou zijn, maar hoe meer ik erover nadacht hoe moeilijker het bleek. Dit kastje is bedoeld om te gebruiken met professionele software. Waar je alleen het kastje hoeft aan te sluiten en de software de rest doet. Dat is geen goed teken aangezien ik dan 'de rest' zelf moet programmeren. Omdat het een grotere uitdaging bleek te zijn dan ik dacht, leek het me een goed idee eerst te kijken naar hoe de installatie waar het kastje vandaan komt in te duiken. Deze installatie gebruikt een programma genaamd Resolume, dit is een bekend



programma en wordt hier gebruikt om een video te weergeven op een reeks led strips. Omdat ik zelf niet veel begreep van Resolume heb ik Tim en Gijs gevraagd om mij te helpen uitleggen hoe ze dit hebben gebruikt. Hierdoor heb ik een beter beeld gekregen hoe dit werkt en ben ik gaan proberen de bol aan te sturen met Resolume. Hier detecteerde het programma het kastje gelijk en nu was het alleen een kwestie van de kanalen goed selecteren. Hier rechts is een screenshot van het selecteren van deze kanalen, ergens anders heb ik aangegeven dat ik van een wit vlak een RGB-



waarde wil krijgen. Dit zijn dus 3 kanalen (R, G, B). Omdat er ook een dimmer kanaal is voor de bol en dit op 255 moet staan om niet te dimmen, heb ik het start kanaal op 4 gezet. Kanaal 4 is dimmer, 5 is rood, 6 is groen en 7 is blauw. Op dit wit vlak is de RGB als output 255,255,255. Dus door te beginnen op kanaal 4 is het: dimmer=255, rood=255, groen=255, blauw=0. Door het kastje te selecteren gaf de bol al gelijk licht. Dus ik wist nu zeker dat het kastje het doet en data kan doorsturen. Nu is de vraag, hoe ga ik dit aansturen vanuit C#. Ik had er eerder al onderzoek naar gedaan en er is weinig documentatie van te vinden. Veel voorbeeld code waren verouderd of op een sketchy website te vinden. Ook heb ik projecten gedownload en code snippets geprobeerd en kwam daar niks uit. Maar door zoveel te zoeken en te proberen was ik erg geneigd om te code te lezen en proberen te begrijpen, ook al was dit voor mijn gevoel ver boven mijn intelligentieniveau. Normaal als ik code vind, plak ik dit in mijn document, probeer ik het en waarschijnlijk moet ik dan iets aanpassen omdat ik het niet echt heb gelezen. Deze switch van mindset was fijn om te ervaren, nu stapte ik uit mijn comfort zone in het code gebied. Wat in mijn mening alleen goed kan zijn, want hoe meer je weet dat het bestaat hoe beter je het kan opzoeken en later weer kan gebruiken.

Na twee dagen staren en proberen had ik het eindelijk voor elkaar gekregen om te bol aan te sturen via Unity. Via een soort 'packet tracer' van de uitgevers van Art-Net zelf heb ik de communicatie van Resolume naar het Art-Net-naar-DMX-kastje kunnen onderscheppen en uitlezen en heb deze parameters

gebruikt om zo een bestaand Unity script dat voor Art-Net is gemaakt te configureren om te praten met dit specifieke kastje. De communicatie gaat instant en kan zo iedere kanaal een waarde geven die gelijk worden doorgestuurd naar de bol. Hier een video van hoe dit eruit zag: <https://youtu.be/Az30TskvbuQ>.

De volgende stap is de bol laten meegaan met de show. In het Unity script wordt er een array van bytes gebruikt. Deze array is 512 waardes lang en staan standaard op 0 en kan maximaal 255 worden. Mijn plan is om deze array te updaten wanneer er nieuwe data binnen komt van het C# programma. Ik moest een formule maken om te berekenen wat de maximale verlenging is, aangezien 255 lager dan de vloer is. En in de toekomst moet er ook een minimale en maximale hoogte zijn voor bijvoorbeeld een show die van Dinner in Motion is waar iedereen moet blijven zitten, dan kan er wel vol gebruik gemaakt worden van de hoogte. Mijn wiskunde is niet al te sterk dus dit vond ik een lastige maar een hele interessante opdracht. Door aan een vriend, die een flink verstand heeft van wiskunde, hoe ik dit het beste kon doen kwamen we uit op de volgende formule. $\text{NewHeight} = \text{input} / 255 \times (\text{max} - \text{min}) + \text{min}$. Met dit laatste puzzelstukje heb ik de data vanuit Unity door kunnen sturen. Hier een video van de bol die mee doet met de show: <https://youtu.be/1QHHyK7B-Bc>.

Het Unity programma stuurt nu data door naar de bol, maar dan verwerkt het C# programma eerst data om vervolgens door te sturen naar Unity dat dan weer naar de bol data verstuurd. Ik denk dat dit niet de efficiëntste manier is om data te sturen, het gaat nu namelijk eerst via een socket server, dan wordt het in de wachtrij gezet in Unity en vervolgens weer via een socket naar de bol verstuurd. Het zou, denk ik, beter en sneller zijn als het C# programma dit direct doorstuurt naar de bol. Hiervoor moet ik dit Unity script gaan overzetten naar mijn C# programma, wat niet moeilijk zal zijn aangezien dit beide in C# geschreven is. Toen het script in het C# programma stond en ik het testen zag ik gelijk verschil. De bol gaat nu ook iets eerder dan de simulatie, wat een goed teken is want de simulatie loopt wat achter omdat mijn laptop moeite heeft met het renderen. De Art-Net data worden tegelijk verzonden met de simulatie data.

Voordat de bol binnen was had ik een mogelijk probleem ontdekt. In Unity is een frame rate van 25 nog redelijk weinig, hoe meer frames hoe vloeiender het oogt. Dit is omdat de bollen in Unity letterlijk teleporteren van positie naar positie. In het echt werkt dat natuurlijk niet zo en heeft de bol een maximale snelheid. Het kan zijn dat de lier de frequentie van data niet aan kan en dat de animatie haperig wordt. Om dit te checken heb ik de maximale hoogte op 30 gezet, zo is alle beweging wat extremer en dat zou deze theorie moeten testen. Zoals op de video te zien heeft de lier er geen moeite mee en maakt het een vloeiende beweging. <https://youtu.be/Llhp31JgtEc>.

De C# applicatie heb ik afgerond en hieronder een video van het resultaat: <https://youtu.be/kQh62svHfSw>.

Vereisten Decoder applicatie

De decoder applicatie is het uiteindelijke product van de C# applicatie waar ik het eerder in dit document over heb gehad.

Om de video te processen is er een speciale codec nodig dat te downloaden is op https://www.codecguide.com/download_k-lite_codec_pack_mega.htm.

Het IP-adres van Art-Net-naar-DMX-kastje moet in dezelfde range zijn zodat ze elkaar kunnen zien. In mijn situatie is het IP van het kastje '2.0.0.102' en ik moest het adres van mijn laptop veranderen naar '2.0.0.100'.

De simulatie moet op dezelfde computer staan. De simulatie moet ook eerst aanstaan voordat je een playlist afspeelt. Als een playlist is afgelopen moet de simulatie opnieuw worden opgestart voordat er een nieuwe playlist kan worden gestart.

Feedback

Tijdens het laten zien van de werking van de decoder applicatie heb ik van Gijs vaak gehoord dat hij het heel vet vindt dat ik dit heb kunnen maken. Ook Bram vertelde mij dat ik echt trots mocht zijn op wat ik heb gemaakt. Dit vond ik echt fijn om te horen, omdat ik zelf zo diep in de code van mijn project zat ik niet echt de tijd nam om te kijken naar wat ik nou heb neergezet.

Reflectie

Tijdens dit project heb ik natuurlijk geleerd hoe kinetische licht installaties werken. Ik kwam erachter dat ik dit soort installaties erg interessant vind, niet alleen het technische aspect maar juist ook de artistieke kant ervan. Waar je kan spelen met de ervaring van een bezoeker en bepaalde gevoelens omhoog kan brengen. Een bijna magisch idee, dat het plafond op en neer kan bewegen en daarom juist een super interessant idee voor mij. Iets waar ik later zeker weer iets mee wil doen.

Toen ik begon met het project geen kennis over DMX, Art-Net, lichttafels, C# socket communicatie, Bytes, C# Windows forms, Resolume. Toch ben ik echt blij dat ik dit allemaal heb leren kennen omdat ik juist verder wil gaan in deze richting. Het leren kennen van DMX en Art-Net heeft me alleen meer nieuwsgierigheid opgewekt en ik zou hier mee in de toekomst zeker nog iets mee willen gaan doen.

Toen ik bezig was met het begrijpen van Resolume kwam ik erachter dat juist wat ik wilde bereiken met het bouwen van mijn eigen applicatie ook in Resolume kon worden gemaakt. In Resolume kan er van een video een kleurwaarde worden uitgelezen en deze waarden kunnen per geselecteerd kanaal worden verstuurd. Hier zou je dus ook een video kunnen gebruiken dat eigenlijk twee video's aan elkaar geplakt zijn. Als ik dit project opnieuw zou moeten doen zou ik eerst mezelf in Resolume hebben laten duiken.