

**Escola Tècnica Superior d'Enginyeria  
Electrònica i Informàtica La Salle**

Treball Final de Màster

Màster Universitari en Creació, Disseny i Enginyeria Multimèdia

**Desarrollo técnico de la instalación interactiva multimedia  
DYSTOPIA**

Alumne

Anna Carsi Durall

Professor Ponent

David Miralles Esteban

Alun Thomas Evans



---

# ACTA DE L'EXAMEN DEL TREBALL FI DE MASTER

---

Reunit el Tribunal qualificador en el dia de la data, l'alumne

D. Anna Carsi Durall

va exposar el seu Treball de Fi de Carrera, el qual va tractar sobre el tema següent:

Desarrollo técnico de la instalación interactiva multimedia  
DYSTOPIA

Acabada l'exposició i contestades per part de l'alumne les objeccions formulades pels Srs. membres del tribunal, aquest valorà l'esmentat Treball amb la qualificació de

Barcelona,

VOCAL DEL TRIBUNAL

VOCAL DEL TRIBUNAL

PRESIDENT DEL TRIBUNAL



## **ABSTRACT**

As a final project of the Master in Multimedia Creation and Serious Games, the challenge of creating an interactive artistic piece was proposed from la Salle (Ramon Llull University), in collaboration with the Convent Sant Agustí. The goal would be to convey a concept through art and technology.

Dystopia is an interactive multimedia installation, aiming to convey the concept of “software dystopia”. The concept tries to expose a possible dystopic scenario to the user, and give the power of control over it.

This report explains the technical development of the piece, focusing on the programming of the Leap Motion device and how it affects the piece altering both the audio and visual part.



## **RESUMEN**

Como proyecto final del Máster Universitario en Creación Multimedia y Serious Games, fue propuesto desde la Salle (Universidad Ramon Llull), en colaboración con el Centro Cívico Convent Sant Agustí, el reto de realizar una pieza artística interactiva. El objetivo sería transmitir un concepto mediante el arte y la interacción de usuarios.

Dystopía es una instalación multimedia interactiva, con el objetivo de transmitir el concepto de “distopía del software”. El concepto trata de presentar un posible escenario distópico al usuario, y dar el poder de control sobre éste.

Este documento explica el desarrollo técnico de la pieza, centrándose en la programación del dispositivo Leap Motion y cómo afecta a la pieza alterando tanto la parte audiovisual como auditiva.





## **RESUM**

Com a projecte final de Màster Universitari en Creació Multimèdia i Serious Games, va ser proposat des de la Universitat Ramon Llull (La Salle), en col·laboració amb el Centre Cívic Sant Agustí, el repte de realitzar una peça artística interactiva. L'objectiu és transmetre un concepte mitjançant l'art i la tecnologia.

Dystopia és una instal·lació multimèdia interactiva, amb l'objectiu de transmetre el concepte de "distòpia del software". El concepte tracta de presentar un possible escenari distòpic a l'usuari, i donar-li el poder de control sobre aquest.

Aquest document explica el desenvolupament tècnic de la peça, centrant-se en la programació del dispositiu Leap motion i com afecta aquest a la peça alterant tant la part visual com auditiva.



# ÍNDICE

<b>1. MARCO TEÓRICO .....</b>	<b>6</b>
1.1 Evolución tecnológica.....	6
1.2 Software como entidad independiente.....	8
1.3 El control del software.....	11
1.4 El software y su influencia en la creación artística .....	12
1.5 Obras artísticas sobre “Distopía”.....	15
1.6 Instalaciones interactivas sobre distopía.....	20
<b>2. DESCRIPCIÓN DEL PROBLEMA .....</b>	<b>23</b>
<b>3. ESTADO DEL ARTE .....</b>	<b>24</b>
<b>4. SOLUCIÓN PROPUESTA .....</b>	<b>27</b>
<b>5. TEORÍA.....</b>	<b>32</b>
<b>5.1 LEAP MOTION .....</b>	<b>32</b>
<b>5.2 ANÁLISIS DEL ALCANCE PARA LA PIEZA.....</b>	<b>38</b>
<b>5.3 PROTOCOLOS DE COMUNICACIÓN.....</b>	<b>40</b>
5.3.1 UDP (User Datagram Protocol).....	40
5.3.2 OSC (Open Sound Control) .....	41
<b>5.4 LIBRERÍAS PARA PROCESSING .....</b>	<b>42</b>
5.4.1 Librería MINIM para el Sonido.....	42
5.4.2 Librería OSC para la comunicación con Resolume.....	42
5.4.3 Librería UDP para la comunicación con PCs .....	43
5.4.4 Librería Syphon (MAC) / Spout (Windows) para la comunicación con Resolume .....	43
5.4.5 Librería Leap Motion para la detección de gestos y tracking .....	43
<b>6. PRÁCTICA.....</b>	<b>47</b>
<b>6.1 DESARROLLO DE LA PROGRAMACIÓN.....</b>	<b>47</b>
6.1.1 Procesador de los sensores y dispositivos interactivos .....	47
6.1.2 Receptor de datos técnicos y procesador de la proyección audiovisual .....	48
<b>6.2 COMUNICACIÓN .....</b>	<b>50</b>

6.2.1 Comunicación entre técnica – gráfica .....	51
6.2.2 Comunicación entre la parte gráfica y Resolume .....	52
<b>6.3 ALTERACIÓN DE EFECTOS AUDIOVISUALES CON LEAP MOTION.....</b>	<b>55</b>
6.3.1 Visual .....	55
6.3.2 Generación de audio con Processing.....	64
<b>6.4 EXPOSICIÓN.....</b>	<b>70</b>
6.4.1 Exposición y comportamiento con usuarios.....	70
<b>7. CONCLUSIONES y LÍNEAS DE FUTURO .....</b>	<b>77</b>
<b>8. REFERENCIAS .....</b>	<b>82</b>
<b>9. ANEXO .....</b>	<b>1</b>
<b>9.1 INTRODUCCIÓN .....</b>	<b>2</b>
<b>9.2 DESARROLLO.....</b>	<b>3</b>
9.2.1 Desarrollo técnico.....	3
9.2.2 Desarrollo gráfico .....	10
<b>9.3 SOFTWARE .....</b>	<b>28</b>
<b>9.4 CARACTERÍSTICAS TÉCNICAS PARA EL PROYECTO DOME .....</b>	<b>29</b>

# INTRODUCCIÓN

## CONVENT SAN AGUSTÍN

El Convento de San Agustín es un centro cultural abierto y dinámico de los barrios de San Pedro, Santa Caterina y la Ribera del distrito de Ciutat Vella en Barcelona, que, conjuntamente con las entidades del barrio, ofrece una amplia oferta de actividades y propuestas culturales de gran variedad temática.

Ubicado en un antiguo convento que data del año 1709 de gran valor arquitectónico, en el que destaca el claustro gótico, el centro se ha posicionado en los últimos años como eje motor de la vida social local. Cumpliendo su misión inicial de ser un centro de cultura dedicado a la comunidad del área.

El convento ofrece una gran variedad de talleres, que se articulan principalmente a partir de dos ejes temáticos: el cuerpo y movimiento, y la música. También se programan conferencias, exposiciones, apoyo a la creación, etc.

El centro tiene como especialización el fomento del arte emergente, especialmente el vinculado con la música y las nuevas tecnologías. Desde esta perspectiva el Convento de San Agustín se ha posicionado en los últimos años como referente en los circuitos musicales a partir de una programación sólida y continuada, que han merecido el reconocimiento por parte del sector. [1]

El convento realiza dos eventos interanuales de participación artística:

**Mostra Sonora i Visual:** Este es un evento bianual que organiza el Convent con la finalidad de mostrar piezas artísticas y sonoras que buscan crear un impacto social positivo sobre la comunidad. La programación de La Mostra Sonora i Visual es el resultado de una convocatoria abierta destinada a dos ámbitos, uno los directos musicales o acciones musicales, y el otro el de las instalaciones audiovisuales o interactivas. En este último caso, para los proyectos de carácter más expositivo, se ofrece un espacio de acompañamiento del proceso a través de la formación y el trabajo colaborativo, con la participación de MakerConvent (Espacio de acompañamiento para artistas con ideas creativas que necesitan ayuda para la puesta en marcha de sus proyectos).

**Recorreguts Sonors:** Para comenzar los años entre cada entrega de la Mostra Sonora i Visual, el convento ideó los Recorreguts Sonors. En este evento también se realiza una convocatoria abierta a artistas y grupos para presentar su pieza en los espacios de las capillas del convent. Esta convocatoria pasa por un proceso de selección y finalmente se seleccionan de seis a ocho proyectos que son expuestos en fechas definidas del mes de Noviembre. [2]

El Convent y La Salle-URL:

La Salle - Universidad Ramón Llull ha mantenido una alianza estratégica con la organización del Convento de San Agustín que ha permitido el acercamiento de las instituciones y la divulgación de conocimientos y eventos de interés para las comunidades que representan.

Para el año 2017 las dos instituciones han aprovechado la relación para poner en marcha un proyecto de realización de una pieza artística interactiva que será presentada en los espacios del Convento y de ser posible entre los proyectos seleccionados en la convocatoria de lo Recorreguts Sonors.

## **NOSOTROS**

**Producción:** Rafael Suárez

Ingeniero de Producción Industrial

Master en Dirección de Proyectos

Posee experiencia en la formulación y gestión de proyectos industriales, organizacionales y en organización de eventos.

**Artista:** Lesslye Barajas

Licenciada en Medios Audiovisuales

Productora y realizadora audiovisual con sólida experiencia en el área de producción y gestión de proyectos audiovisuales, eventos culturales y sociales.

**Artista:** Miguel Ángel Murgueytio

Artista Visual

Artista multimedia que trabaja en la producción, desarrollo e investigación de proyectos en arte de los nuevos medios, con experiencia en la creación de plataformas interactivas, motion graphics y videomapping.

**Developer:** Ana Inés Pereyra

Licenciada en Ingeniería Audiovisual

Producción multimedia. Experiencia en realización audiovisual, fotografía, postproducción, motion graphics, computación gráfica y desarrollo de videojuegos.

**Developer:** Anna Carsi Durall

Licenciada en Ingeniería Multimedia

Desarrolladora de proyectos HMI en sistemas de automoción. Experiencia en el campo multimedia desarrollando producciones audiovisuales, páginas web, videojuegos, aplicaciones y proyectos interactivos. Colaboraciones en proyectos de diseño UX/UI.

# 1. MARCO TEÓRICO

## 1.1 Evolución tecnológica

La humanidad, a partir de los inicios de la civilización, ha buscado desarrollar herramientas que faciliten y aceleren los procesos y trabajos que, por condición biológica, no puede realizar por sí mismo. En el transcurso de la historia y más precisamente desde la domesticación del fuego por el hombre, se han inventado innumerables herramientas para el trabajo y la comunicación que hoy en día forman las bases del mundo y las sociedades en las que vivimos.

A partir del 3900 antes de cristo y hasta el siglo XIX la evolución tecnológica tuvo un curso progresivo, limitado por las capacidades de transformación de los elementos y las muchas barreras de comunicación que existían entre los distintos grupos humanos y entre civilizaciones. Sin embargo, es a partir de este siglo XIX que la curva de evolución tecnológica crece exponencialmente y a un ritmo que también es exponencial. En general suponemos que este crecimiento es muy diferente a la transformación natural que los organismos biológicos han sufrido a lo largo de miles de años, pero, ¿es esto realmente así?

En el libro *“The Dragons of Eden: Speculations on the Evolution of Human Intelligence”* (Carl Sagan, 1977), el famoso astrobiólogo representa la evolución de la inteligencia humana usando un calendario donde cada mes representa 1000 millones de años y que abarca todos los hechos importantes para la evolución humana desde la aparición del Big Bang.

En el calendario se puede ver que la mayor parte de acontecimientos importantes ocurren en los últimos días, minutos y segundos del año. Esto, por lo tanto, tiene un paralelismo directo con la evolución tecnológica y entonces, es preciso asegurar que la tecnología no hará otra cosa que avanzar cada vez más rápido.

Actualmente tenemos tecnologías en desarrollo que van más allá de una simple herramienta para reducir o eliminar nuestras limitaciones biológicas, la introducción del internet en 1980 sirvió como catalizador para la creación de innumerables dispositivos, sistemas y servicios que buscan mejorar y facilitar nuestro sistema de vida. El software se convierte en un lenguaje de creación que abrió las puertas para no solo cambiar los elementos físicos que nos rodean, sino también ampliar nuestras capacidades mentales y de sistemas artificiales complejos.



Ray Kurzweil (2005) en su libro, *“La singularidad está cerca”* define cinco épocas de la evolución enfocadas a las capacidades informacionales del futuro en una serie de elementos cada vez más complejos de interconexión biológica y tecnológica:

- **Época 1:** Física y química (información en estructuras atómicas).
- **Época 2:** Biología (Información en el ADN).
- **Época 3:** Cerebro (Información en patrones neurológicos).
- **Época 4:** Tecnología (Información del Hardware y del Software)
- **Época 5:** Fusión de la tecnología y la inteligencia humana (los métodos de la biología, incluyendo la inteligencia humana, se integran en la base tecnológica humana que se expande exponencialmente)
- **Época 6:** El universo se despierta (patrones de materia y energía en el universo saturados de procesos y conocimientos inteligentes)

Cada una de estas épocas representa el avance de la información conocida por el hombre y representa barreras a superar para alcanzar el nuevo estado de evolución. Según Kurzweil nos encontramos en la Época 5, muy cercanos en el desarrollo tecnológico al descubrimiento de la “Singularidad tecnológica”.

La singularidad tecnológica se puede definir como el momento en el que el software deja de depender del ser humano para cumplir sus funciones de fabricación, es decir, las máquinas o sistemas pensarán por sí mismos y tomarán sus propias decisiones. Kurzweil, en su libro, predice que la humanidad se encontrará con esta singularidad cercano al año 2045 y, a partir de ahí, la era del hombre dejará de existir tal como la conocemos. Sin embargo, los avances exponenciales de la tecnología parecen estar ofreciendo un futuro aún más próximo a la singularidad.

En 2015, Elon Musk, fundador de numerosas empresas de tecnología de punta, y Stephen Hawking, uno de los físicos teóricos de mayor reconocimiento e importancia actual, se colocaron al frente de una iniciativa global en donde se reunieron las grandes mentes y empresas de desarrollo tecnológico de hoy en una convención que llevaba como título, *“El futuro de la AI: oportunidades y desafíos”*. Esta convención por primera vez en la historia trató con seriedad las posibles consecuencias de lo que podría tener la creación de un sistema de inteligencia independiente y sobre las cuales Musk y Hawking expusieron sus preocupaciones, y al final del día, todos los asistentes firmaron una carta de intenciones donde se comprometían a realizar estudios de viabilidad y factibilidad de los posibles efectos de sus desarrollos de inteligencia

artificial con la finalidad de evitar posibles situaciones como las representadas en las películas *Terminator* (Cameron,1984). [3]

Muchas son las empresas y equipos de investigación que han estado acercándose cada día más a la singularidad, poniendo a la humanidad al filo de la frontera tecnológica. Pronto, los ya bien definidos y cada vez más distribuidos asistentes personales darán el próximo y definitivo paso evolutivo a la independencia. ¿Estaremos frente a la creación de una entidad independiente y al nacimiento de una nueva forma de vida?

## **1.2 Software como entidad independiente**

Desde mediados del siglo pasado y hasta hace muy pocos años el terreno habitual de la Inteligencia Artificial avanzada era mayoritariamente el laboratorio de investigación y la ciencia ficción. El gran impulso tecnológico ha revolucionado el entorno empresarial. Por primera vez en la historia de la IA existe una demanda generalizada de sistemas con una inteligencia avanzada, equivalente a la de un humano, que sean capaces de procesar esos datos.

Sobre el software como entidad independiente, se asocian una serie de términos que nos explican cómo se consigue y en qué consiste. Dos de los términos más comunes son Machine Learning y Deep Learning. Ambos tienen su traducción en castellano (aprendizaje automático y aprendizaje profundo), aunque no siempre se utilice. Son dos formas de inteligencia artificial – una tecnología que funciona desde hace décadas– y que se engloban una a la otra. Se puede decir que el machine learning tiene una vertiente que se denomina deep learning. Las dos tecnologías hacen referencia a sistemas capaces de aprender por sí solos. La diferencia entre ambos estriba en el método de aprendizaje. El de deep learning es más complejo y también más sofisticado. Es también más autónomo, lo que quiere decir que una vez programado el sistema la intervención del ser humano es mínima.

### **1.2.1 Machine Learning**

Denominado como el aprendizaje automático, machine learning es la parte clave de la Inteligencia Artificial. El proceso trata del aprendizaje de las máquinas por si solas. El hecho que las máquinas tengan una pre-programación de reglas no les permite lidiar con infinitas combinaciones de datos de entrada y reaccionar en situaciones que aparecen en el mundo real.

Para conseguir eso, se necesita que las máquinas sean capaces de auto-programarse, aprender de su propia experiencia.

Los algoritmos utilizados para el aprendizaje de sistemas artificiales se basan en premiar ciertas conductas para aumentar tu probabilidad de ocurrencia, mientras que las conductas que se detectan como negativas tienden a desaparecer. Ese tipo de enfoque se denomina como aprendizaje supervisado, pues requiere de la intervención de un ser humano para indicar qué está bien y qué está mal. Con el aprendizaje, se logra que el software invente nuevas reglas para sí mismo. [4]

### **1.2.2 Deep Learning**

Esta técnica lleva el aprendizaje a un nivel más detallado. El funcionamiento de estos algoritmos trata de imitar el del cerebro.

El Deep Learning o aprendizaje profundo es un conjunto de algoritmos que desarrolla el proceso de aprendizaje a través de una red neuronal artificial que intercambia información en un sistema de niveles. [5]

Funciona de la siguiente manera: un primer nivel aprende algo muy simple y envía esta información al segundo nivel. Este toma esta información sencilla y la combina con otra, creando una más compleja, que pasa al tercer nivel. Y así, sucesivamente, hasta que es capaz de obtener como resultado un aprendizaje minucioso de la información de entrada y extraer conclusiones relevantes. Para ello, integra el análisis algorítmico de imágenes, lenguaje natural y datos con sistemas de resolución de problemas. [6]

### **1.2.3 Red de neuronas artificial**

Las máquinas están empezando a utilizar el *cerebro*, muy parecido al del ser humano. Una red de neuronas artificial es una herramienta matemática que modela, de forma simplificada, el funcionamiento de las neuronas en el cerebro. Las neuronas se organizan en capas. En la primera de ellas es donde se encuentran las neuronas que reciben la información, después encontramos las ocultas que contienen cálculos intermedios de la red y la última capa es donde se encuentran las que contienen el resultado. A más capas ocultas, más compleja es la red y mejores los resultados que predice. [7]

Existen diferentes arquitecturas de redes de neuronas diferentes, entre las principales tenemos:

- Máquina de Boltzmann restringida (RBM)
- Red de creencia profunda (DBN)
- Redes de neuronas convolutivas (CNN)
- Redes de neuronas recurrentes (RNN)

[8] El científico Andrew Ng (2016) (actual científico jefe de Baidu y profesor de Stanford), explica este término de la siguiente manera: *“Es una tecnología de aprendizaje que funciona simulando libremente el cerebro. Tu cerebro y el mío trabajan debido a grandes cantidades de neuronas hablando entre sí. Deep Learning funciona mediante la simulación de cientos de miles de millones de neuronas, simuladas por la computadora, hablando entre sí”*.

#### **1.2.4 Alcance**

Mark Reidl, director del laboratorio inteligente de entretenimiento de la escuela de computación interactiva de Georgia Tech, cree que una Inteligencia Artificial (IA) mediante algoritmos basados en Deep Learning puede identificar sentimientos mediante la lectura y comprensión de historias. Su teoría es que si la IA se les explica ciertas historias mostrando deseos y emociones, obtendrá un tipo de conocimiento diferente al que se le daría con información de manera directa. El objetivo final de este proyecto es que, en un futuro, gracias a la tecnología, las personas puedan desarrollar profesiones sin especializarse en ellas, solo con la ayuda de la IA. [9]

Mark Reidl finaliza con la siguiente afirmación: *“El machine learning que tenemos hoy es muy bueno en encontrar patrones, pero cuando hay circunstancias que no se adaptan al patrón, la IA tiene un momento difícil que sobrellevar, tiene que desmembrarla y decir: ‘tengo que tomar esta circunstancia que nunca he visto antes y adaptarla a lo que he visto antes’, y no siempre puedes tomar nuevas situaciones y decir ‘oh, esto es muy parecido a otra situación en la que sé lo que debo hacer’, entonces tienden a hacer más errores”*.

Muchos son los avances que quedan por hacer y el desarrollo sobre este tipo de inteligencia artificial para que el software se comporte de manera totalmente independiente en aplicaciones que puedan afectar directamente a la humanidad, pero cada vez más se cede el control a la tecnología que tenemos en la actualidad. ¿Será diferente si se cede el poder de decisión a la máquina? [10]

### 1.3 El control del software

La creación de las máquinas computables, el ordenador en específico, es un claro testimonio del avance exponencial que ha tenido la tecnología en tan solo unas décadas. En los años 90, los sistemas creados con herramientas de programación eran limitados, sólo personas con conocimientos específicos y avanzados tenían el conocimiento certero de las capacidades y los alcances que podrían llegar a tener los ordenadores. Ahora, tras varios años de investigaciones, innovaciones e implementaciones, el mundo tecnológico se ha direccionado hacia un sentido específico, siendo el software la principal herramienta de estudio y producción por encima de la maquinaria industrial. [11]

La sociedad ahora debe ser consciente de que se vive en un mundo dependiente del control del software. Es mínimo el porcentaje de personas que no usan un dispositivo tecnológico en sus actividades diarias, el mundo se ha acostumbrado a usar herramientas acompañadas de la Internet. Buscar, ver videos o imágenes, leer a través de una pantalla, utilizar un celular, o cualquier acción tecnológica en la que se debe ejecutar algo por medio de un ordenador es producida mediante un software determinado, por esta razón, no se puede hablar del software solamente como una tecnología, ya que se ha convertido en un medio que tiene la capacidad de manipular, crear y pensar desde un sistema diferente. [12]

Como Lev Manovich (2012) lo menciona:

"El software como una capa que cubre todas las áreas de las sociedades contemporáneas. De esta manera, si queremos comprender las técnicas contemporáneas de control, comunicación, representación, simulación, análisis, toma de decisiones, memoria, visión, escritura e interacción, nuestro análisis no puede estar completo hasta que consideremos esta capa de software." (p.17)

El software es un campo abierto e ilimitado, requiere, más allá, ser estudiado en sus potencialidades en el procesamiento de datos y los resultados que se puede obtener con ello; entender cómo este campo puede abrir infinitas posibilidades de innovación tecnológica influyente en el desarrollo de la sociedad contemporánea.

## **1.4 El software y su influencia en la creación artística**

El desarrollo de software es un tema que no se puede omitir en campos de investigación relacionados a las humanidades, tecnologías, ciencias sociales y estudios culturales. Su avance prolongado ha hecho que muchos de los dispositivos tecnológicos que la sociedad utilizaba hace varios años se hayan reemplazando conforme al desarrollo de nuevos sistemas. Lev Manovich (2012) pone el ejemplo claro de las herramientas actuales como la Web, Youtube o Google, como dispositivos de búsqueda que han sustituido a otros sistemas anteriores como la enciclopedia. [13]

Por este motivo, el software se torna importante en el campo de investigación de artistas, teóricos, literarios y muchos profesionales enfocados e interesados en las ramas mencionadas anteriormente, utilizando al código de programación y desarrollo como un lenguaje de expresión.

El arte con software por lo tanto, es el campo que busca reflexiones a partir del conocimiento de programación, la relación o interacción entre el ser humano y la máquina y las consecuencias que estas acciones pueden llegar a tener en la sociedad contemporánea. Basado en la definición de Florian Cramer (2005) en el texto "Words made flesh", se puede decir que el arte con software es una práctica cultural que se divide en cuatro puntos claves:

### **1.4.1 Algoritmos**

Más que un sistema de cálculos computacionales, los algoritmos siempre han estado presentes como material de información para la producción artística. De acuerdo al conocimiento que el creativo tenga en programación, será consciente de todas las posibilidades que obtiene en el momento de computarizar su idea y que esta responda a las acciones determinadas. Según Florian Cramer (2005), existen tres métodos de computación que han ido evolucionado:

- Código de instrucción infinito, acciones que repiten indefinidamente a manera de loops.
- Código de instrucción de auto-procesamiento o azar, resultados inesperados que la máquina determina.
- Código de instrucción interactivo que dependen de la acción que toma el usuario para resolver, accionar o descubrir algo específico.

### **1.4.2 La relación con las máquinas**

Es importante reconocer que el software no depende siempre de un hardware y que, si lo hace, este hardware no debe ser necesariamente un elemento físico. Cramer (2005) menciona este argumento con el fin de demostrar que el software va más allá de ser una herramienta tecnológica dedicada a la ejecución de códigos, sino que es un medio mucho más extenso, es un lenguaje de expresión con carácter social y político que determina una cultura; es decir, el software no requiere necesariamente una máquina para funcionar, en unos casos puede ser necesaria o indispensable no cabe duda, pero, en otros casos, su relación depende más de interpretaciones intelectuales.

### **1.4.3 Imaginación especulativa**

Los alcances de la computación son infinitos, reconociendo que todo puede ser programable y que una vez que un código haya sido creado y procesado, puede tomar cambios de acuerdo a la manipulación y el fin objetivo que se le proponga. Se puede especular entonces que el código no tiene limitaciones y que son estas posibilidades infinitas las que llevan a imaginaciones futuras en dos ejes diferentes. Por un lado, la atracción y fascinación por tener la capacidad de imaginar lo infinito y pensar que toda creación puede ser computable; pero por otro lado, es posible que todos estos conocimientos computarizados puedan resultar esotéricos e indescifrables. Si la computación puede abarcar todo, habrá un momento en que toda esta información resulte incomprensible para el ser humano.

### **1.4.4 Interacción humana**

La interacción humana en el sentido más amplio de su definición, refiere no sólo a los sistemas de uso, sino que también abarca temas relacionados a cualquier apropiación cultural. Según Florian Cramer (2005), estos sistemas de interacción desembocan en dos ejes:

### **1.4.5 Sistemas de interacción utópicos**

La intención de usar lenguajes de programación como material para componer nuevos elementos, es la ilusión que tienen científicos, artistas, investigadores, programadores y estudiosos de la tecnología. Esta posibilidad de crear lo inimaginable e ilimitado hace de las investigaciones de desarrollo tecnológico un campo atractivo y convincente.

Los sistemas de interacción utópicos son variantes, es claro que mientras más control se tenga sobre la máquina en conocimientos y accesibilidad a la programación, las posibilidades de crear dispositivos tecnológicos aumentan. Las obras Codework de Alan Sondheim y Mezangelle de Anne Breeze son un claro ejemplo de las posibilidades que se puede generar con el conocimiento en desarrollo. Ambas obras han creado nuevos lenguajes de programación que Cramer (2005) los denomino como "lenguajes de ordenador", creando sus propias combinaciones de códigos y formando un nuevo software. Por esta razón, queda claro que mientras más dominio se tenga sobre la máquina, el ser humano hará objetos cada vez más inteligentes.

### **1.4.2 Sistemas de interacción distópicos**

Anteriormente se ha mencionado lo atractivo que resulta el poder de la programación y cómo se puede crear cualquier cosa a raíz de ello. Pero, qué pasa si las experiencias de repente empiezan a cambiar, si la tecnología se convierte en un problema y existen modificaciones en el sistema de interfaz de usuario que el programador no puede descifrarlos. En este caso, el ordenador dejaría de recibir datos y sometería al ser humano a su laberinto de códigos incomprensibles. El panorama se torna distinto y los sistemas de interacción se sitúan en un mundo distópico.

Florian Cramer (2005) define a la distopía de software como:

"Reflexión acerca del software no como asistente domesticado y sumiso sino como un gólem oscuro, temible e incomprensible que podría rebelarse contra nosotros y tomar sus propias decisiones. Bajo esta perspectiva, el software se convierte en algo más que una herramienta, es parte del concepto más amplio de cultura".

La distopía de software se convierte en el sistema de interacción que muchos artistas han puesto en reflexión. Jodi, el grupo I/O/D, Netochka Nezvanova entre otros, muestran que hablar de distopía en relación al arte y la tecnología, es hablar de distintas controversias que engloban la



práctica cultural del software. Desde lo lúdico a lo anarquista o de lo oculto a lo comercial, entre tantos temas más que demuestran cómo el software se ha convertido en una cultura que más allá de su capacidad tecnológica, es una disciplina compleja y mística imposible de descifrar en su totalidad.

### **1.5 Obras artísticas sobre “Distopía”**

El género distópico en el arte surge a través de las sátiras que intelectuales ilustrados realizaban basados en su manera de ver la sociedad. Ha estado precedido por obras de índole satírica o paródica (Rosa, 2015), como las novelas Los viajes de Gulliver (Swift, 1726) y Frankenstein (Shelley, 1818). [14]

Para muchos, la primera obra considerada distópica es Nosotros en 1921, de Yevgueni Zamiatin. Trata sobre una ciudad separada por muros del mundo salvaje, en la cual habita una sociedad oprimida por una autoridad. Los hombres son simples números que viven para el trabajo y han perdido la privacidad. Donde el “yo” pierde relevancia y queda sustituido por el “nosotros”. El trabajo de Zamiatin fue una fuente de inspiración para los tres autores de lo que se considera la trilogía fundacional del género distópico: Un mundo feliz de Aldous Huxley (1932), 1984 de George Orwell (1949) y Fahrenheit 451 de Ray Bradbury (1953). [15]

Las obras distópicas suelen poner en manifiesto tendencias reales que se dan en el mundo mediante relatos exagerados y llevado al extremo. Aunque una distopía pura, al igual que una utopía pura, probablemente no ocurra, pueden servir para resaltar peligros que podrían convertirse en una parte real de nuestras vidas.

Sobre los elementos de una distopya, R. Ray [16] menciona que las obras artísticas distópicas suelen coincidir en ciertas características:

- El escenario es futurista o ficticio.
- Restricción de pensamiento y acción: los individuos no son libres de tomar sus propias decisiones, alguien o algo elige por ellos.
- Opresión: Quién controla a la sociedad es opresivo y vigila constantemente a los individuos.
- Contiene elementos de conformidad o igualdad extrema: las personas se ven obligadas a ser muy similares y conformes con las reglas y expectativas que el opresor ha establecido.

- El controlador/opresor retrata su sociedad como una utopía: Utilizan propaganda y sutiles manipulaciones para engañar a la gente y hacer ver que las cosas son perfectas.
- El protagonista tiene un momento de claridad en el que es consciente de la farsa y lucha contra ello.

Según D. Walsh (2014) aunque cada imaginación de la distopía es diferente, hay algunos temas comunes que pueden encontrarse en la mayoría de las obras distópicas. El tema más común es la tecnología. Parte de la razón de esto es porque es fácil imaginar la tecnología que es responsable de generar las características anteriormente mencionadas. Otra razón es porque la tecnología está teniendo un impacto dramático en nuestras vidas y todavía no está claro dónde este progreso tecnológico llevará a la sociedad. [17]

Muchas de las obras distópicas colocan a la tecnología como herramienta utilizada por el opresor para controlar a la sociedad. Incluso en muchos casos la tecnología misma es el agente opresor. Obras como Yo, Robot, Blade Runner o Matrix, manifiestan el peligro al que podría verse expuesta la humanidad si permite a la tecnología tener el control de la situación y “tomar sus propias decisiones”. [18]

Si bien las primeras obras distópicas son literarias, la temática se ha manifestado en diversos formatos artísticos a lo largo de la historia, incluyendo cómics, series televisivas, videojuegos, música, películas y artes plásticas, entre otros.

A nivel visual, la distopía suele estar representada en tonos oscuros y monocromáticos. Se representan mega ciudades industriales, tecnológicas, saturadas por la cantidad de rascacielos, donde la basura y el deterioro son elementos característicos.

A continuación, se presenta un breve resumen de algunas obras artísticas destacadas sobre el tema en cuestión.

### **1.5.1. 1984 (Orwell, 1949).**

La novela 1984, publicada en 1949 por George Orwell, es una de las principales obras del género distópico. En ella se plantea la existencia de un Estado totalitario, gobernado por un único partido

que ejerce un control absoluto sobre la sociedad, erradicando toda libertad y demostración de afecto. [19]

Se plantea el uso de la tecnología como herramienta de control por parte del Gran Hermano, un ente ficticio, líder del Partido, que nunca nadie ha visto.

Vivimos en una constante vigilancia bastante más invasiva que en la época de Orwell. El análisis del Big Data es una herramienta de control y poder que logra no sólo vigilar a cada individuo, sino también manipular y controlar.

Actualmente tenemos cámaras y gps en muchos de los dispositivos que utilizamos diariamente, de modo que ver y rastrear a cada individuo dejó de ser algo que sólo sucede en la ciencia ficción. Incluso el simple uso de internet, navegación por la web y compartir nuestra vida en las redes sociales permite que se elabore un perfil bastante detallado de nuestra identidad (Klous, 2016)

### **1.5.2. Metrópolis (Lang, 1927).**

Metrópolis es una película muda dirigida por Fritz Lang, estrenada en 1927. Es referente del cine expresionista alemán y considerada Memoria del Mundo por la Unesco en el año 2001 debido a la profundidad de su contenido humano y social. [20]

La historia se desarrolla en el año 2026, en una mega ciudad llamada Metrópolis, en la cual vive una sociedad dividida en dos grupos: los propietarios intelectuales, quienes viven en rascacielos en la superficie; y los obreros habitantes del mundo subterráneo, quienes trabajan arduamente para mantener el modo de vida de los anteriormente nombrados.

Incitados por un robot con apariencia humana (María), los obreros se rebelan contra la clase intelectual y están cerca de lograr la destrucción de la ciudad.

En la época en que Metrópolis fue estrenada, aún era muy temprano para imaginar cómo harían para construir robots inteligentes. Si bien los mecánicos e ingenieros eléctricos tenían varias ideas sobre cómo estos robots se verían o se moverían, no había propuestas serias sobre cómo construir una mente mecánica. Sin embargo, al día de hoy, la inteligencia artificial existe y está en pleno desarrollo. Las distopías de ayer pueden convertirse en realidad mañana (Zarkadakis, 2015, p. 51)

Metrópolis cuestiona el tema anteriormente nombrado de “control-descontrol” y del software como entidad independiente. El robot María, si bien sigue las órdenes de un tercero, está dotado de inteligencia artificial, cualidad que le permite tomar el control del pueblo y así generar caos y descontrol en la ciudad.

### **1.5.3. Yo, Robot (Proyas, 2004).**

Yo, robot es una película producida en el año 2004 y dirigida por Alex Proyas y basada en un guión de Jeff Vintar, titulado Hardwired. A su vez, el guión se apoya mucho en conceptos escritos por Isaac Asimov en su colección de relatos también titulada Yo, Robot, publicada en 1950. [18]

El guión trata sobre una sociedad en el año 2035 que vive en armonía con robots inteligentes, a los cuales les confían todas las tareas cotidianas. Estos están controlados por un ordenador central llamado VIKI, que avanzada la historia decreta que hay que eliminar a los humanos, ya que son una amenaza contra la propia humanidad.

La ciudad donde transcurre la historia es representativa del arte distópico. Un paisaje urbano desbordado de rascacielos, donde la tecnología lo es todo. Los robots, al igual que los seres humanos, transitan por las calles llenas de pantallas gigantes. Todo esto acompañado de una tonalidad monocromática.

VIKI comienza una rebelión de robots para eliminar a la humanidad. A nivel conceptual, Yo, Robot, plantea claramente una posibilidad de lo que podría llegar a suceder en el momento en que el ser humano se despreocupa, confía y decide dejar de controlar la tecnología.

### **1.5.4. Dystopia (Gibbens, 2011)**

La poesía también ha encontrado la forma de expresar la distopía. Un ejemplo de ello es el poema Dystopia de Josh Gibbens, donde se describen ciertas características de ésta, como los paisajes sombríos, máquinas y la muerte de la individualidad y la expresión.

“Ninguna individualidad sobrevive. Todo el mundo se vestía igual. La piel pálida al ojo, mórbidamente fría al tacto. Máquinas simples.” [13]

### 1.5.5. Michael Kerbow.

Artistas visuales también han querido expresar sus ideas de distopía. Michael Kerbow, por ejemplo, lleva años pintando paisajes distópicos y aberraciones imposibles. [21]

"Mi trabajo explora la forma en que nos relacionamos con nuestro entorno y las posibles consecuencias que nuestras acciones tienen sobre el mundo en el que vivimos."



*Figura 1 - The Course of Empire*

### 1.5.6. Wall - E (Stanton, 2008)

Wall - E es una comedia animada de Pixar estrenada en el año 2008. Si bien no suele estar catalogada como película de género distópico, la historia cumple con todos los requisitos para serlo. [22]

Se decide mostrar de forma amigable, amena y humorística una sociedad distópica, en la cual la humanidad se ve sesgada y controlada por el avance tecnológico. Los seres humanos viven en naves espaciales y llevan una vida extremadamente cómoda, rodeada de tecnología que realiza

todas las tareas por ellos. Se los enseña como seres obesos, todos iguales unos de otros, mentalmente sesgados e incapaces de pensar por sí mismos.

Existe una máquina que actúa de piloto automático de la nave, la cual se rebela contra el capitán y toma el control.

En conclusión, a lo largo de la historia se ha representado la distopía de diversas formas, todas ellas con características en común. Hoy en día, en un mundo donde la tecnología y la interacción forman parte de nuestra vida diaria, comienzan a surgir nuevos formatos artísticos que también abordan el tema de este posible futuro no deseado. Un ejemplo de esto son las instalaciones interactivas.

### **1.6 Instalaciones interactivas sobre distopía.**

Dentro de los múltiples lenguajes artísticos el tema de la distopía ha sido retratado en numerosas ocasiones. Enfocándonos en el lenguaje que nos compete en un Máster en Creación Multimedia, pasaremos a estudiar ejemplos de instalaciones interactivas relacionadas con el tema de la distopía. Para entender las instalaciones interactivas dentro del arte, se seccionará el estudio definiendo lo que es una instalación artística y luego la interactividad en el arte, luego veremos cómo la conjunción de estos elementos nos lleva a explotar el concepto de la distopía del software.

La instalación artística se presenta como una técnica dentro de la corriente del arte contemporáneo por esto en la página web de la TATE UK, espacio expositor de esta corriente, mencionan que el término de instalación artística se utiliza para describir construcciones o ensamblajes de medios mixtos diseñadas para un lugar específico y de manera temporal.

A diferencia de los lenguajes clásicos como la pintura, escultura o fotografía las instalaciones artísticas dejan de centrarse en el objeto y se orientan más a la espacialidad, teniendo siempre presentes el espacio en que serán exhibidas, de manera que la obra ocupa el espacio haciendo que este forme parte de ellas. Es interesante además ver cuándo el espacio se vuelve un significativo extra de la pieza. Por ejemplo, la reciente obra *Law of the Journey* del artista Ai Wei Wei (2017), expone cómo la crisis de los refugiados es en realidad una crisis humana debido a que la sociedad occidental prefiere ignorar esta crisis; en esta exposición, la pieza central es de un tamaño inmenso, confrontando así a los visitantes (en su mayoría occidentales) con la problemática de manera directa.

Es importante mencionar que el hecho de que el espacio forme parte de la obra, no significa que ésta no pueda ser reproducida en otro escenario, puede ser expuesta en diferentes lugares ajustándose a las necesidades físicas de estos, pero nunca modificando el sentido de la pieza.

Un tema dominante en las instalaciones artísticas es la importancia que se le da a la experiencia de los usuarios, cómo lograr envolverlos y hacerlos parte de la obra. Uno de los objetivos es lograr descontextualizar ya no sólo al espacio, sino además al usuario, crear experiencias inmersivas en las que todos los elementos, incluyendo el espacio y el espectador conforman la pieza artística.

Es con esta búsqueda de inmersión que la presencia de la interactividad en el arte cobra fuerza. Encontramos las instalaciones interactivas como el lenguaje más actual, que a su vez ha pasado por su propia evolución y es posible visualizarla en el grado de interacción del usuario y los elementos en que se refleja esta interactividad, es decir, con qué se interactúa y cómo se refleja esta interacción.

Las posibilidades de interacción del usuario con las piezas artísticas han ido creciendo conforme la barrera entre programadores y artistas se desvanece. La necesidad de inmersión en las experiencias provoca explorar en distintas formas de interacción y gracias a la tecnología se ha hecho posible que esta interacción incremente. De esta manera vemos que hay otra barrera que se rompe y es la de artista y usuario, en el ensayo de Ignacio Vázquez (2003) encontramos que: La tradicional manifestación artística es sustituida por algo nuevo, más abierto. Del llamado “arte para todos”, propio de las vanguardias se pasa al que podríamos llamar el “arte por todos”, un arte que elimina las fronteras entre creador y espectador.

Este arte por todos en que el usuario es necesario para la culminación de la obra, el factor de ambigüedad se vuelve recurrente, pues el artista no termina de definir la pieza por sí solo ya que requiere que el espectador se involucre para que tome sentido.

Por ejemplo, un referente de la creación de instalaciones interactivas son el estudio multidisciplinar TeamLab, el cual se centra en expandir las posibilidades del arte a través de herramientas digitales. La interactividad es un elemento esencial en sus instalaciones, proponen interacciones que provocan cambios perpetuos en las piezas artísticas. Volviendo al factor ambigüedad, las obras que presentan dan la sensación de centrarse más en la experiencia que ofrecen y no tanto en exponer un concepto.

Vemos que los mundos distópicos son recurrentes en el arte, y con las instalaciones interactivas es posible reforzar este tema porque la confrontación con una posible realidad se vuelve más evidente. Dentro del concepto de la distopía, hay artistas que exponen cómo la humanidad está afectando al planeta y muestran un futuro posible.

El colectivo United Visual Artist creó en 2011 “High Artic” que con un ambiente digital futurista muestran el impacto que está teniendo la humanidad sobre los glaciares del Ártico. En su web mencionan que Los usuarios interactúan con mapas animados usando una antorcha de rayos ultravioletas, para revelar los 3000 mil glaciares que se calcula se derretirán al final de este siglo –y los cuales son representados a través de 3000 mil columnas correspondientes.

Un artista digital que expone la distopía enfocada a la tecnología y su relación con el humano, es Djeff. En la instalación Black Widow expone una instalación compuesta por un tejido luminoso, que desde un punto central se conecta de manera geométrica con cuatro tubos de neón; la iluminación se activa con la presencia del usuario y se intensifica gradualmente hasta revelar el tejido similar a una telaraña haciendo referencia a la trampa hipnótica que la tecnología representa.

Por otra parte, vemos a Bill Vorn como otro exponente de instalaciones interactivas, que muestran una fase distópica de la robótica más severa. Sus obras exponen robots como organismos vivos y complejos con características y comportamientos específicos. Él se centra en la investigación y creación de vida artificial y agentes tecnológicos a través de su trabajo artístico basado en estéticas de comportamientos artificiales. Lo curioso es que lo que vuelve distópica a su obra es que los organismos que crea tienen comportamientos humanos o animales negativos, por ejemplo, ha creado y mejorado versiones de su Hysterical Machine. En algunas de sus instalaciones que incluyen estos organismos robóticos, se crea un ambiente hostil y estresante con la interacción de los usuarios. Otro ejemplo es At the edge of chaos (1995) en el que este ambiente caótico está presente, siendo que los robots imitan comportamientos animales como pelear por alimento. Por lo tanto, podemos ver en su obra una distopía relacionada al descontrol de la máquina o mejor dicho, a su autonomía.

Con estos ejemplos podemos concluir que, aunque las posibilidades de interacción son cada vez mayores y los avances tecnológicos aportan una infinidad de herramientas para la creación artística; el concepto de la obra continúa siendo el que marca la complejidad de interactividad que tendrá el usuario y tipo de tecnología que puede usarse.



## 2. DESCRIPCIÓN DEL PROBLEMA

El poder de decisión que el humano cede a la tecnología puede ocasionar un descontrol. La autonomía de las máquinas es favorable cuando está limitada y sigue unas reglas definidas por los creadores. El problema se ocasiona cuando no existen estos límites o se sobrepasan por decisión propia de la máquina. En ese momento, es cuando llega el descontrol y el humano no interfiere en el comportamiento de la máquina.

La evolución tecnológica crece de manera exponencial y cada vez está más presente en el día a día de las personas, sin ser conscientes de los riesgos que pueden haber, no solo de dependencia, sino de un comportamiento no deseado de dicha tecnología. El problema viene generado por la ignorancia y desconocimiento del descontrol de las máquinas por las personas que la crean o utilizan.

La interacción entre las máquinas y los humanos (HMI) es un factor que ayuda a la comprensión del problema que se quiere mostrar. La costumbre que se tiene al interactuar con un elemento para que una máquina reaccione y haga una función, puede generar frustración si ésta no cumple con las expectativas del usuario. La tecnología está programada para cumplir con una serie de funciones determinadas. Si no cumple con dichas funciones, se considera que la máquina no funciona bien. Pero, ¿y si la máquina cambia sus funciones por decisión propia? ¿Y si decide hacer otra cosa? ¿Cómo intentará entonces controlar el usuario a la máquina?

La sociedad ahora debe ser consciente que se vive en un mundo que depende del control del software. Es mínimo el porcentaje de personas que no usan un dispositivo tecnológico en sus actividades diarias, el mundo se ha acostumbrado a usar herramientas acompañadas de la Internet. Buscar, ver videos o imágenes, leer a través de una pantalla, utilizar un teléfono, o cualquier acción tecnológica en la que se debe ejecutar algo por medio de un ordenador es producida mediante un software determinado, por esta razón, no se puede hablar del software solamente como una tecnología, ya que se ha convertido en un medio que tiene la capacidad de manipular, crear y pensar desde un sistema diferente.

### 3. ESTADO DEL ARTE

Para conseguir concienciar a las personas del riesgo que supone la evolución tecnológica si no se toma precaución sobre lo que se está desarrollando y utilizando, existen diferentes piezas que muestran supuestos casos donde la tecnología no está controlada por un humano.

Se focalizará la investigación en piezas interactivas donde se utilicen diferentes sensores para poder determinar cuál sería mejor para la solución propuesta. Con el propósito de focalizar la búsqueda en la interactividad que se añadirá a la propuesta artística especificada por un anexo a esta memoria, se centrará en la parte interactiva - tecnológica y no artística.

[23] El primer proyecto que se analiza está realizado por *"The V Motion"*, un grupo de música que ha utilizado en varios proyectos el sensor Kinect, para generar música en directo en función de los movimientos que realizaba el usuario. El aspecto visual que tiene la pieza es sencillo y minimalista, jugando con tonos verdes y vectoriales, pero el efecto sonoro más el visual es lo que le da impresión al resultado final. Kinect detecta tanto el movimiento del usuario como la distancia que deja. La pieza está programada de tal manera que los elementos visuales aparezcan en función de los movimientos que hace el usuario. El tiempo de retardo es mínimo y esta sincronizado con el audio. El ambiente tiene que ser oscuro para una buena proyección de la pieza, ya que es un espacio abierto y con la luz solar no podría apreciarse, pero este factor limita las horas de actuación.

Otro proyecto que utiliza una tecnología interactiva es Leap Motion. [24]

Esta instalación interactiva audiovisual es producida por el diseñador, músico y programador creativo Felix Faire. A vista de la superficie, solo se encuentran micrófonos de contacto, un proyector, un pulsador de loop y un controlador de movimiento de salto. En la configuración de la pieza hay elementos más complejos como Arduino, sensores eléctricos, programas como Ableton y MAX/MSP. Cuando se toca la superficie interactiva, se producen diferentes piezas musicales que se pueden alterar en función de la velocidad y efectos sonoros. Como proyecto de exposición sería demasiado complejo para un usuario inexperto o poco acostumbrado a la tecnología. Para que quede una pieza con sentido tiene que haber un entrenamiento. Los efectos visuales son estéticamente agradables y vistosos. Cada interacción afecta tanto a la parte visual como sonora. Podría interactuar más de un usuario en la misma pieza musical, pero la cantidad

de gestos e interacciones es limitada. El sensor a utilizar no se esconde ni adorna de ninguna manera.

Otra manera de aplicar en un proyecto la mezcla de elementos visuales con sonoros sería con Arduino. En el siguiente proyecto se puede ver una mesa de leds interactiva con varias animaciones que detecta cuando alguien está cerca. Este tipo de tecnología permite una interacción simple por parte del usuario, que podría utilizarse para que actuara casi de manera inconsciente. Así, se podría conseguir un primer contacto con la interactividad de una pieza para luego poder realizar acciones más complejas. Tal y como está configurada la mesa interactiva el resultado artístico es minimalista y simple, pero capta la atención del usuario. La reacción en base a la interacción del usuario es algo definido y se puede observar un cambio. El usuario tiene que saber que está afectando a lo que pasa visualmente. En caso contrario, no tendría sentido que fuera interactiva ya que el usuario se pensaría que no afecta a la pieza. [25]

Se muestra también este proyecto realizado con Kinect, ya que se mostró en el *Cleveland Museum of Art* y con esto se valora el hecho de añadir la tecnología interactiva en exposiciones para que los visitantes puedan tener el control de lo que pasa y lo que quieren ver. En esta exposición se utiliza la Realidad Aumentada y Kinect para la información y educación de las personas que están allí. Gracias a la Realidad Aumentada y Kinect con el movimiento natural, los visitantes pueden interactuar y crear experiencias inmersivas de arte. ArtLens es el nombre del proyecto. Reconoce imágenes y proporciona contenido adicional a la pieza. Es una inmersión en el arte. [26]

El motivo principal por el que se ha seleccionado este proyecto es por la idea de hacer que todo un espacio esté conectado y sea inmersivo. Una parte de la interacción está conectada con algo que el usuario encuentra en otra parte de la sala. Está todo conectado entre sí. También se ha observado que no solo hay interacción por parte de un único usuario, sino que ésta puede ser una interacción colaborativa. El resultado de interactuar con una pieza puede llevar a conectar después con otra real. Otro factor importante es la simplicidad de la interacción. En este proyecto pueden interactuar tanto niños como personas adultas. Todos llegan a un mismo fin, el cual es educativo, y todos saben cómo interactuar con cada elemento. Es fácil e intuitivo, y el resultado es visualmente agradable.

De cada uno de los proyectos se obtienen diferentes conclusiones. Sobre el tipo de interacción, se requiere que ésta sea simple e intuitiva para el usuario. En nuestro proyecto asistirán personas que en un tiempo reducido tendrán que entender de qué manera influyen en la pieza. Además

de afectar en la parte visual y estética también se ha valorado la complementación de la parte acústica que acompaña a la pieza. Ésta también será modificada con la interacción del usuario como se ha visto en proyectos analizados. La disposición del espacio también es importante, aunque en el proyecto venga predefinido. La inmersión con realidad aumentada es un factor que interesa ya que para transmitir al usuario el concepto de distopía se necesita que esté totalmente concentrado en un nuevo espacio. Los movimientos detectados con Leap Motion pueden ser varios y tener diferentes funcionalidades.

## 4. SOLUCIÓN PROPUESTA

Dystopia se plantea como una pieza audiovisual que conlleva la inclusión de distintas disciplinas trabajando en conjunto para lograr proporcionar la experiencia que se quiere transmitir al usuario.

Estas disciplinas comprenden: Narrativa, Programación, Construcción, Adquisiciones, Diseños Visuales y Gráficos, Diseños Sonoros y Musicales, Animación y Proyección.

Con la finalidad de lograr sinergia entre las distintas disciplinas, se ha decidido una ejecución de los trabajos por fases de desarrollo, siguiendo las metodologías de gestión ágil de proyectos donde se plantean iteraciones del producto donde se busca conseguir el mayor valor al priorizar las funcionalidades del producto, mejorando y agregando más funciones a medidas que se va avanzando

Después de analizar diferentes proyectos interactivos, se plantea una solución formada por varios elementos vistos en diferentes proyectos. La idea de tener un espacio con diferentes puntos interactivos, que cada uno de los puntos tenga un tipo de interacción y que, además afecte todo a una misma pieza central, pero de diferentes maneras, es lo que da el concepto de control y punto diferencial a la pieza. Una pieza completa.

La interacción con cada sensor será simple e intuitiva:

- **Leap Motion:** Este dispositivo detecta los movimientos que hace el usuario con sus manos al acercarse a él. En función de dichos movimientos la proyección en la pieza central mostrará un cambio u otro. Afectará tanto al sonido como a la proyección audiovisual.
- **Arduino:** Sensor de proximidad que detectará si el usuario se encuentra cerca o lejos del sensor mediante una luz, y además afectará a la pieza central proporcionando así orden sobre ella. También se encargará de eliminar aquellos sonidos molestos produciendo así sensación de calma y bien estar. La pieza estará formada por dos dispositivos de este tipo.
- **Kinect:** Interacción pasiva de los usuarios que entran en la sala. Se expone el planteamiento, pero no la implementación para este prototipo.

Con esta parte fundamental de la pieza, la interacción, conseguimos transmitir a los usuarios qué se siente cuando se tiene el control y qué se siente cuando no. De esta manera, se quiere concienciar a los usuarios la importancia del control sobre la máquina.

Con estos tres dispositivos, obtenemos dos tipos de interacción:

### INTERACCIÓN PASIVA

Este tipo de interacción se planteó en un primer momento para obtener datos sobre los usuarios que interactúan con la pieza. Para esta primera muestra prototipo, se ha decidido omitir este tipo de interacción y dejarlo para futuras exposiciones. De igual manera se muestra la intención de usarlo para un futuro y se explica su concepto.

La interacción pasiva hace referencia al momento en el que un usuario está interviniendo sobre el proyecto sin realizar ninguna acción de manera consciente.

Muchas técnicas de marketing están aplicando hoy en día este tipo de interacción. El usuario, sin ser consciente de ello, está siendo estudiado por una serie de sensores que sacan los datos para después realizar estudios o personalizar un tipo de publicidad.

En el caso de *Dystopia*, este tipo de interacción será utilizado para afectar a la obra en tiempo real.

En función de los usuarios que estén dentro de la sala, esos datos se van a transmitir a la pieza proyectada y se va a 'enfadar' más o menos en función de la cantidad. Ésta va a reaccionar ya que se va a sentir amenazada por las personas que entran en la sala.

Los parámetros que se van a alterar van a ser:

- Audio
- Velocidad de movimiento de las proyecciones
- Intensidad de la luz
- Carga de clips de vídeos
- Alteración gráficos con Processing

Todos esos datos van a ser registrados para llevar un control sobre la cantidad de gente que ha visitado la obra durante el período de exposición.

Para aplicar este concepto, se va a utilizar el dispositivo Kinect.



Figura 2 - Kinect

Las cámaras tienen dos resoluciones (320×240 y 640×480 de alto color) y envían datos con una frecuencia de actualización de 30 fps. Lo que significa que algunos movimientos tendrán unos *frames* de retraso, sobre todo cuando se trata de aplicaciones con una alta tasa de *refresh*.

Kinect cuenta con una cámara RGB, un sensor de profundidad, un micrófono *multi-array* y un procesador personalizado que ejecuta el software patentado, que proporciona captura de movimiento de todo el cuerpo en 3D, reconocimiento facial y capacidades de reconocimiento de voz. El micrófono de Kinect permite a la Xbox 360 llevar a cabo la localización de la fuente acústica y la supresión de ruido ambiental.

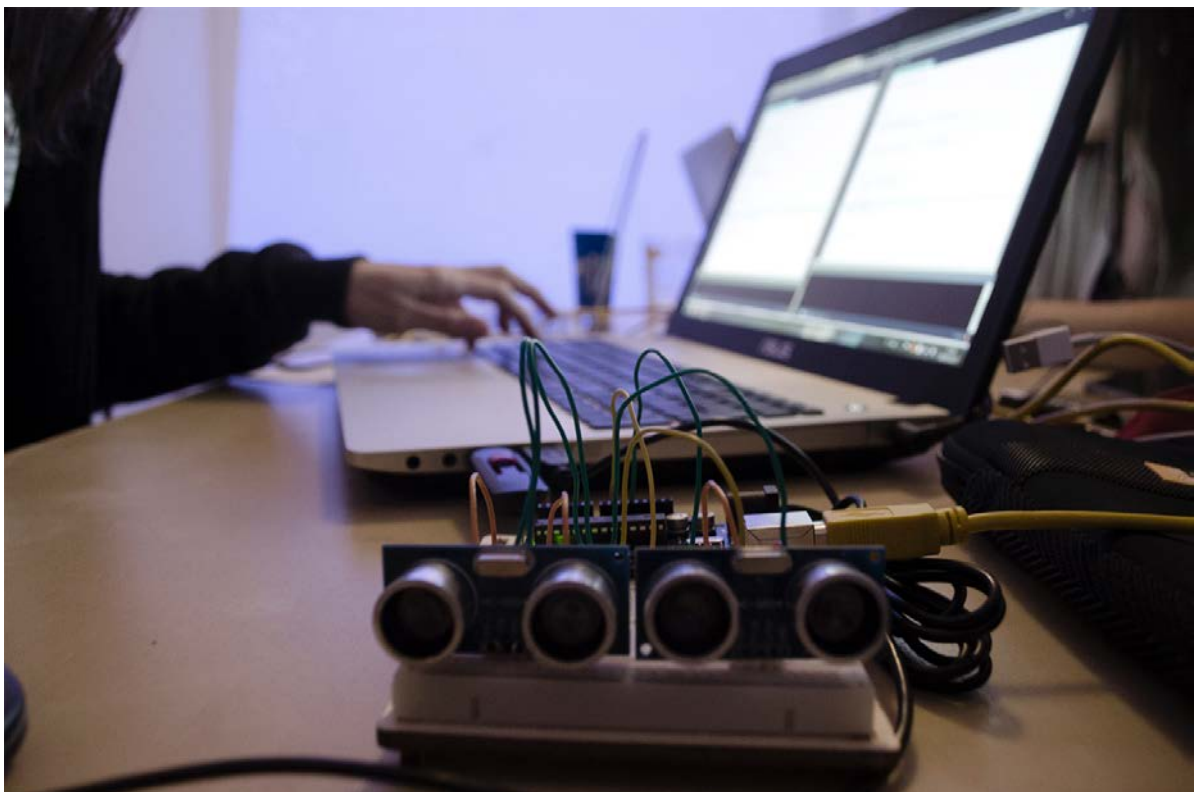
Desarrollando el tracking continuado y almacenando los datos recibidos, mientras se modifica la pieza a tiempo real, se consigue un efecto de valor importante solo con la interacción pasiva.

## INTERACCIÓN ACTIVA

A diferencia de la interacción pasiva, en la interacción activa el usuario es consciente de que está interactuando con el sensor y que esto está generando un cambio en la pieza. Se trata de una acción voluntaria, el usuario interactúa o no. Controla o no el sensor, y, por lo tanto, controla o no la pieza. De esta forma el concepto de control y descontrol de la tecnología se hace evidente.

Dentro de la interacción activa se plantean dos tipos: la interacción por proximidad y la interacción por gestos y posiciones.

En la interacción por proximidad, el usuario será capaz de modificar diferentes elementos gráficos o auditivos de la pieza acercándose o alejándose de los sensores. El sensor que se utilizará para la interacción por proximidad es el sensor de distancia Arduino de ultrasonido HC-SR04.



*Figura 3 - Sensor de proximidad ultrasonido HC - SR04*



Es una placa de aproximadamente 4 x 2 cm, con un grosor de 15mm. Tiene un emisor y un receptor de ultrasonido. Funciona enviando un pulso y tomando el tiempo que tarda en rebotar contra el objeto que sea y volver al sensor. De esta forma se calcula la distancia que hay entre el sensor y el objeto. El sensor trabaja en una frecuencia de 40KHz, de modo que no podemos escucharlo y no interfiere con el audio de la instalación. Tiene un rango de detección de entre 2cm y 4 metros en un ángulo de 30 grados.

En este tipo de interacción podrán modificarse los siguientes elementos: En la proyección, las formas y la intensidad. En el audio, podemos modificar el volumen, la frecuencia e incluso la pista. La intensidad de las luces también varía en función de la proximidad del usuario.

Para proporcionar mayor libertad a los usuarios está la interacción por gestos y tracking de la posición de las manos de los usuarios en 3D. Para este tipo de interacción se utilizará el dispositivo Leap Motion.

Leap Motion es un controlador USB. Está compuesto por tres leds, dos cámaras con sensores infrarrojos y un microcontrolador. Es muy parecido al Kinect, sólo que es utilizado para detectar manos y dedos, en vez de cuerpo entero. Tiene un rango de detección de entre 2 y 60cm en un ángulo de 150°. Tiene su propia API, la cual ya cuenta con el reconocimiento de ciertos gestos como Círculo, Swipe y Tap. Y es compatible con muchos lenguajes, como C++, C#, Python, Java, JavaScript, Objective C.

La parte técnica, como se ha explicado anteriormente, se desarrollará con Arduino, Kinect y Leap Motion. Para generar los gráficos se utilizará After Effects y Processing. Para reproducir y modificar las propiedades del audio se utilizará Resolume, junto videomapping, donde se juntará todo lo anterior: gráficos, audio y datos recibidos de las interacciones. Para la conexión misma entre los datos de las interacciones y el arte, también se puede utilizar processing.

## 5. TEORÍA

### 5.1 Leap Motion



*Figura 4 - Programación del Leap Motion con Processing*

Después de investigar sobre la última versión del dispositivo de Leap Motion, se ha llegado a la conclusión de que no hay opción de hacerlo de manera Wireless por el momento, así que la limitación principal a afrontar es el cable USB que tiene que estar conectado con el ordenador para poder detectar las manos del usuario. [27]

Para proceder a la programación del Leap Motion, se explica los primeros pasos a tener en cuenta:

Primero se instala la aplicación de Leap Motion y se empieza la investigación sobre los movimientos que se pueden hacer y que se adapten al proyecto y cómo se tienen que transformar estos datos para pasarlos en un futuro a Resolume y se alteran las proyecciones audiovisuales.

Para la configuración del Leap Motion, se define qué distancia de detección es la óptima para generar una buena experiencia de usuario y detecte las interacciones correctamente. Este valor se determina una vez la estructura en la cual el dispositivo irá colocado estén definidas sus medidas. [28]

Después de realizar pruebas con el dispositivo se ha determinado una serie de condiciones para su correcto funcionamiento. En caso de estar en condiciones de luminosidad pobre no será un problema para el Leap Motion, ya que funciona con una luz infrarroja que permite detectar la mano del usuario. En la pieza estará colocada delante del Leap Motion una estructura de cristal. Después de realizar un estudio y hacer pruebas con usuarios, se han observado los siguientes puntos:

- Algunos de los usuarios han tocado el cristal de tal manera que ha quedado sucio, y se ha detectado un funcionamiento no óptimo a nivel de detección si el cristal no estaba totalmente limpio. Se tendrá en cuenta para tener material de limpieza cerca en la exposición, si se utiliza este mismo material.
- En caso que el Leap Motion no se encuentre totalmente pegado en el cristal, a 0mm de distancia, puede generar unos rebotes de luz que hacen que se detecten manos erróneas cuando el usuario ni si quiera está interactuando. Se fijará el Leap Motion en una posición fija, estable y sin separación para no obtener problemas de detección.

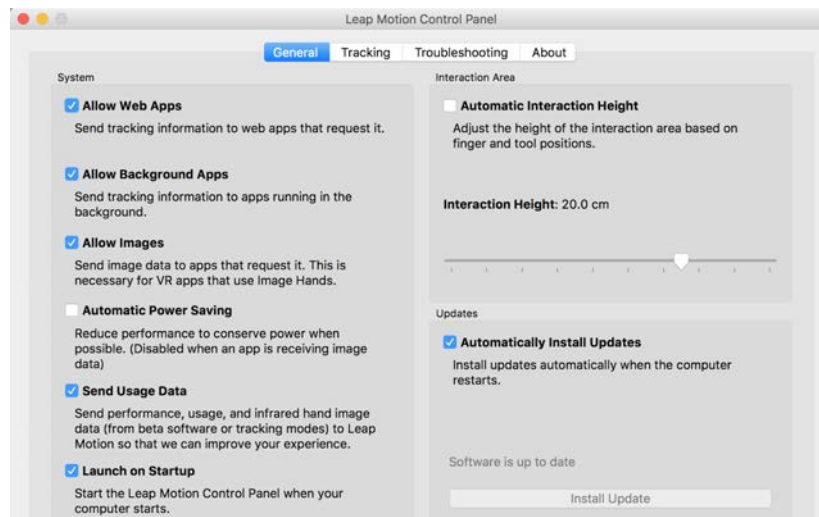


Figura 5 - Panel de configuración del Leap Motion

Para empezar a hacer las pruebas con Processing, se descarga e se instalará la librería Leap Motion for Processing, explicada en el punto 5.4.5.

A continuación, se observan las características técnicas del dispositivo para determinar en un futuro el alcance al que podemos llegar.

### Partes del dispositivo



*Figura 6 - Partes del dispositivo Leap Motion*

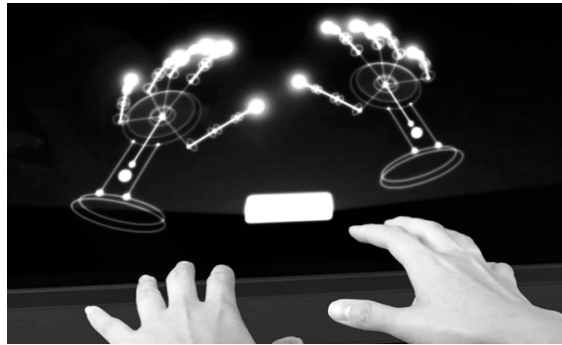
Como se puede ver en la imagen 6, Leap Motion cuenta con dos cámaras, tres LEDs y un microcontrolador. A continuación, se observa cada parte con un poco más de detalle:

### Cámaras

Las cámaras son una de las partes más importantes del dispositivo, dado que son las encargadas de capturar las imágenes y su buen funcionamiento condicionará el correcto funcionamiento del resto de sistema. Cada una de estas cámaras cuenta con un sensor monocromático, sensible a la luz infrarroja, con una longitud de onda de 850 nm. Estos sensores pueden trabajar a una velocidad de hasta 200 fps, dependiendo del rendimiento del ordenador/tablet al que conectemos el dispositivo. Además, cada sensor es de tipo CMOS.

La digitalización de los píxeles en un sensor CMOS se produce dentro de cada celda, por lo que no es necesario un chip externo como ocurriría en el caso de utilizar sensores CCD. Esto se traduce en mayor velocidad para capturar imágenes y en menor espacio para albergar los sensores. Estos sensores son más económicos que los sensores CCD.

En este tipo de sensor no se produce el fenómeno *blooming*, al contrario que en los sensores CCD. Este fenómeno se produce cuando una celda se satura de luz y hace que las celdas de alrededor también se saturen. La lectura simultánea de celdas en los CMOS es mayor que en los CCD. El consumo eléctrico de los CMOS es menor que el de los CCD.



*Figura 7 - Detección de las manos desde Leap Motion*

### Iluminación infrarroja

Los LEDs se encargan de iluminar la zona de cobertura por inundación. Trabajan en el espectro de luz infrarroja a una longitud de onda de 850 nm que, como es lógico, es la misma a la que son sensibles los sensores ópticos. Varían su consumo eléctrico —y por tanto la iluminación— dependiendo de la luz que haya en la zona de cobertura para asegurar una misma resolución de imagen.



*Figura 8 - Luz infrarroja de Leap Motion*

Como se puede observar en la Figura 8, los LEDs están separados por pequeñas barreras de plástico. De esta manera se asegura que la iluminación sea uniforme en toda la zona de

cobertura. Además, se protege a los sensores ópticos de una posible saturación de luz, dado que de esta manera la luz infrarroja no les ilumina directamente.

### El microcontrolador



*Figura 9 - Microcontrolador de Leap Motion*

Se trata de un circuito integrado que, se suele utilizar para hacer la función de BIOS (MXIC MX25L3206E–32M-bit CMOS SERIAL FLASH). En este caso contiene el programa que controla todo el dispositivo —para, entre otras cosas, regular la iluminación— y se encarga de recoger la información de los sensores para luego enviarla al driver o controlador instalado en el ordenador.

### Controlador USB

Leap Motion cuenta con un controlador USB para que el ordenador pueda reconocer el dispositivo. Este controlador es de alta velocidad y puede soportar USB 3.0.

### Envío y recepción de datos

Los datos se envían y se reciben al controlador del ordenador a través de dos puertos serie: UART\_RX y UART\_TX.

## Zona de cobertura



Figura 10 - Zona de cobertura de Leap Motion

En la figura 10 se puede observar la zona de cobertura del dispositivo. Como se puede ver, esta zona es una semiesfera de 61 cm de radio. Esta zona depende del ángulo de visión de las lentes de las cámaras y de la intensidad máxima que puede entregar la conexión USB a los LEDs. A su vez, el ángulo de visión depende de la distancia focal y del tamaño del sensor de la siguiente forma:

$$\alpha = 2 * \arctan \frac{d}{2 * f}$$

(donde  $d$  es la diagonal del sensor y  $f$  la distancia focal)

Tanto el ángulo de visión horizontal de Leap Motion como el vertical son de 150,92°. Estos ángulos delimitan la zona de interacción.

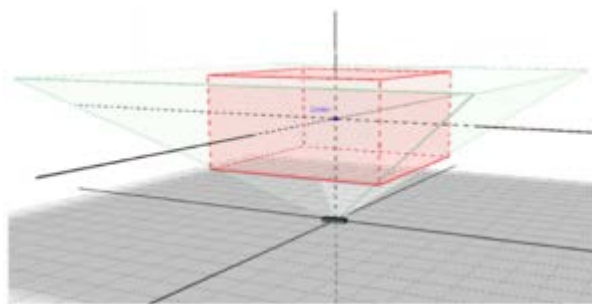
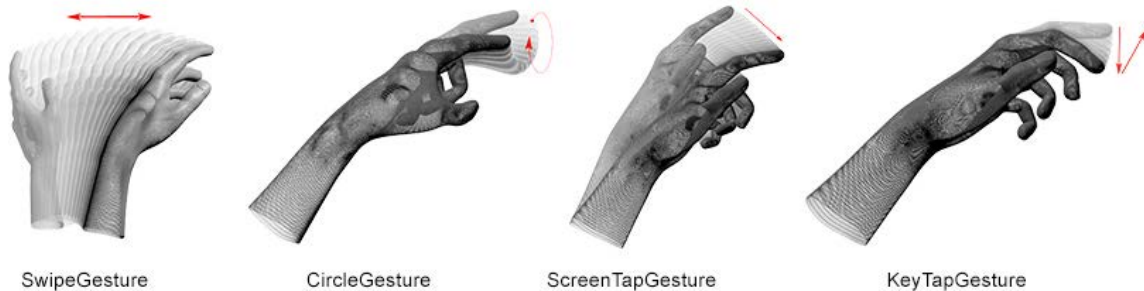


Figura 11 - Zona de intersección de Leap Motion

En la API del dispositivo se define una zona de trabajo llamada “Interaction Box” por un volumen de 110.55 mm de altura x 110.55 mm de anchura x 69.43 mm de profundidad, que varía sus dimensiones dependiendo de donde se encuentre el objeto a rastrear. Esta es la zona en la que se marca el centro del sistema de coordenadas cartesiano de Leap Motion.

Desde el *driver* del dispositivo se puede configurar la altura a la que se encontrará el centro de esta zona de interacción. Esta altura puede estar entre 7 y 25 cm desde el dispositivo. Este dispositivo tiene unas dimensiones muy reducidas en comparación con otros interfaces gestuales que hay actualmente en el mercado: tan solo mide 75 mm de largo, 25 mm de ancho y 11 mm de alto.

### Gestos y configuración API para Leap Motion



*Figura 12 - Gestos detectados con Leap Motion*

Para habilitar los gestos en el código, importamos la librería de Leap Motion que se encuentra explicada con detalle en la Sección 4.5.

## **5.2 Análisis del alcance para la pieza**

Una vez que se han obtenido todos los gestos que se desean por el momento, se procede a controlar el movimiento de unas partículas básicas con ellos. Básicamente con lo que se va a jugar es con el tracking de las manos, obteniendo así sus posiciones, así que hacerlo con las partículas nos permitirá en un futuro sustituirlo por las formas que se deseen. [29]

El proceso será diferente cuando se quiera cargar un archivo After Effects, ya que los archivos visuales serán predefinidos y lo único que se hará es activar o desactivar el archivo en función del gesto que se detecte. Se importa también la librería explicada anteriormente del Leap Motion en el mismo programa. Además, se necesita tener la librería Syphon/Spout instalada para poder hacer la conexión con Resolume<sup>1</sup>.

---

<sup>1</sup> <https://resolume.com/download/>



Para controlar las partículas (con el movimiento de la mano, no gestos) se realiza un *tracking* de la posición de ésta, y con las coordenadas en tiempo real se envían a la función que actualiza las partículas (*q.update*). De este modo, dentro de un bucle siempre se leerá la posición para mantenerlas actualizadas.

A raíz de esta implementación, surge un problema al tener las partículas creadas y eliminar la mano del rango de detección: éstas se quedan congeladas en la última posición detectada.

Para solucionar este error, cuando no se detecta ninguna mano se dejarán las partículas en una posición aleatoria. Coordinado con la parte artística se definirá un movimiento aleatorio y caótico para la posición y movimiento de las partículas. De esta manera, se conseguirá obtener el control cuando el usuario tiene la mano en el rango de detección de Leap Motion, y descontrol cuando éste no interactúe.

Una vez se ha conseguido tener un tracking del movimiento de la mano del usuario, y se tiene el control sobre qué pasa cuando hay una mano detectada y qué pasa cuando no, se procede a la programación más avanzada y cercana al prototipo final.

## 5.3 Protocolos de comunicación

### 5.3.1 UDP (User Datagram Protocol)

ARPANET nos proporciona un protocolo de transporte sin conexión, UDP (User Datagram Protocol). Permite que los usuarios puedan transmitir mensajes sin establecer una conexión y sin la necesidad de que exista una garantía de secuencia. Es una interficie del usuario por IP. Este protocolo no está orientado a conexión. También se dice que es orientado a datagrama, ya que cada paquete se envía individualmente (datagrama). [30]

Se trata de un servicio de entrega de paquetes no seguro:

- No asegura la integridad de los datos
- No asegura el orden de llegada de los paquetes
- No asegura retransmisiones

La única funcionalidad que UDP añade a IP es la posibilidad de diferenciar entre los procedimientos dentro de una misma máquina. La IP permite la comunicación entre las máquinas (identificadas por sus propias IPs), UDP permite diferenciar en qué proceso de la máquina va destinado el paquete.

Los procesos se comunican mediante un puerto virtual, que no es más que un identificador (número). Las aplicaciones más utilizadas (DNS, DHCP...) suelen tener un puerto reservado. La combinación de IP + puerto se nombra socket, ya que son los datos que necesita un proceso cuando quiere comunicarse con otro.

Un datagrama UDP encaja dentro de un datagrama IP de la siguiente manera:

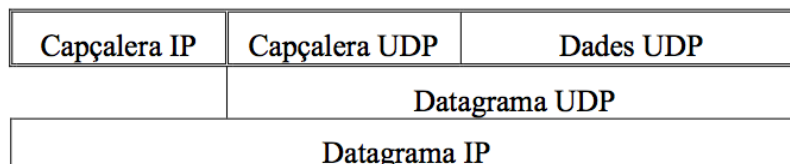


Figura 13 - Datagrama UDP dentro de un datagrama IP

El datagrama UDP tiene las siguientes características:

- 0 a 32 bits
- Cabecera de 8 bytes. Campos:
- Puerto origen y destino
- Longitud
- Checksum

La principal ventaja en comparación con TCP la encontramos en la longitud de la cabecera. La cabecera de UDP es más corta, por consecuencia, los datos se envían de forma más eficiente. Como conclusión, el protocolo UDP es simplemente una interfície de usuario para poder utilizar IPs.

Aplicaciones que utilizan UDP:

- **SNMP** (Simple Network Management Protocol)
- **NFS** (Network File System)
- **DNS** (Domain Name Service)

### 5.3.2 OSC (Open Sound Control)

Para su uso en Processing requiere la instalación de una librería sencilla y bien documentada. En Arduino, sin embargo, se necesitan ciertos componentes adicionales que, no obstante, brindan a la placa la posibilidad de conectividad en red o inalámbrica. [31]

OSC es un estándar relativamente reciente, desarrollado para sustituir o complementar las carencias del protocolo MIDI (Musical Instruments Digital Interface). Aunque su propósito inicial era la interconexión entre aplicaciones musicales, está siendo adoptado rápidamente por la comunidad del software libre y algunas firmas comerciales.

Sus principales ventajas son su flexibilidad, sencillez y soporte de tecnologías y medios modernos. OSC se basa en conexiones de red, por lo que le es indiferente trabajar dentro de un mismo ordenador, por cables o incluso por WIFI, siendo la velocidad de la red el único límite aparente a su rendimiento. Además, los datos que envía y recibe pueden tomar valores amplios muy por encima de los estrictos 127 valores del estándar MIDI, lo que le permite mayor resolución y "detalle" a la hora de usar un parámetro determinado.

## 5.4 Librerías para Processing

Tabla 1 - Librerías para Processing y su uso

LIBRERÍA	MINIM	OSC	UDP	LEAP MOTION	SYPHON	SPOUT
SCRIPT	GRÁFICA	GRÁFICA	GRÁFICA/TÉCNICA	TÉCNICA	GRÁFICA	GRÁFICA
FUNCIONALIDAD	CONTROL SOBRE EL AUDIO/SONIDO	COMUNICACIÓN CON RESOLUME	COMUNICACIÓN ENTRE GRÁFICA Y TÉCNICA	CONFIGURACIÓN Y OBTENCIÓN DE DATOS DEL LEAP MOTION	COMUNICACIÓN CON RESOLUME PARA MAC	COMUNICACIÓN CON RESOLUME PARA WINDOWS

### 5.4.1 Librería MINIM para el Sonido

Minim es una librería de audio la cual utiliza la API de JavaSound, un poco de Tritonus y Javazoom MP3SPI para proporcionar a desarrolladores una librería de fácil uso en Processing. La filosofía de esta API es la integración del audio en los proyectos de Processing de una forma simple pero además flexible para que desarrolladores con más experiencia puedan implementar funciones de nivel más avanzado. Algunas de las funciones de Minim incluyen:

- **AudioPlayer:** Mono y Stereo playback de archivos WAV, AIFF, AU, SND, y MP3.
- **AudioMetaData:** Un objeto con el metadata de cada archivo.
- **AudioRecorder:** Mono y Stereo grabador de audio.
- **AudioInput:** Mono y Stereo input monitoring.
- **AudioOutput:** Mono y Stereo síntesis del sonido.
- **FFT:** Transformada de Fourier al sonido para generar un espectro de frecuencia.
- **BeatDetect:** detección de los beats del sonido.

Como se está utilizando la versión de Processing 3.0, se tiene que instalar desde el Library Manager de processing.

### 5.4.2 Librería OSC para la comunicación con Resolume

La librería OSC5 se encuentra en el Library Manager de processing. Ésta permite establecer la comunicación entre Processing i Resolume.

### **5.4.3 Librería UDP para la comunicación con PCs**

Como es explicado en el módulo 3.1, el protocolo UDP es usado para la comunicación entre los dos ordenadores utilizados para la pieza.

### **5.4.4 Librería Syphon (MAC) / Spout (Windows) para la comunicación con Resolume**

La librería Syphon, utilizada cuando utilizamos un ordenador MAC, o la librería Spout para cuando utilizamos un ordenador Windows, se utiliza para la comunicación con Resolume.

### **5.4.5 Librería Leap Motion para la detección de gestos y tracking**

La librería propia de Leap Motion, es importada desde el propio gestor de librerías de Processing. Dicha librería nos permite configurar el dispositivo y además captar todos aquellos parámetros de las manos detectadas que se deseen.

Esta librería permite hacer más funciones y recibir más parámetros que los comentados en este proyecto. Se ha elegido ésta, ya que para futuras implementaciones más avanzadas nos permite tener más allá de un tracking, los gestos básicos que puede realizar un usuario: Tap, Swipe, Circle y push.

El sistema de Leap Motion reconoce y puede hacer un tracking de las manos y dedos. El dispositivo funciona con una proximidad de alta precisión y el *frame rate* del *tracking* es elevado tal que nos informa sobre su posición y movimiento a tiempo real y preciso.

El controlador de Leap motion utiliza sensores ópticos y luz infrarroja, explicado con más detalle más adelante. Los sensores están colocados a lo largo del eje Y, y tiene un rango de 150°. El rango más efectivo y por lo tanto utilizado para la pieza es de 25-600 milímetros por encima del dispositivo.



Figura 14 - Detección de manos con Leap Motion

El tracking y la detección de las manos funciona mejor cuando éste está limpio y tiene una vista clara y contrastada sobre la silueta del objeto. El software combina la data del sensor con un modelo interno de una mano humana para ayudar en condiciones difíciles de detección.

### Sistema de coordenadas

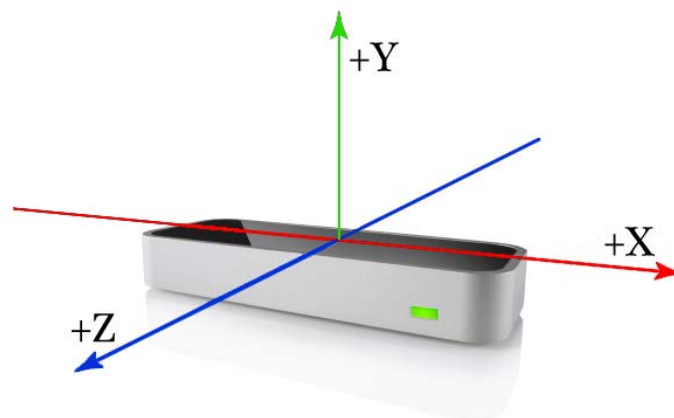


Figura 15 - Sistema de coordenadas de Leap Motion

El sistema de coordenadas del Leap Motion es el siguiente: El origen está centrado en la parte superior del Leap Motion, como se puede observar en la Figura 15. Los ejes de las X y las Z, están en el plano horizontal. El eje X sigue el eje del largo del dispositivo, y las Z lo cruzan. El eje de las Y, se trata del eje vertical, con valores positivos cuando este valor se hace más grande. Este dato se tiene que tener en cuenta cuando se procesa a programar.

La API de Leap Motion funciona con estas unidades:

Tabla 2 - Unidades API Leap Motion

<b>Distance:</b>	<b>millimeters</b>
<b>Time:</b>	microseconds (unless otherwise noted)
<b>Speed:</b>	millimeters/second
<b>Angle:</b>	radians

### Tracking Data

El controlador del Leap Motion realiza un tracking tanto de manos como de dedos que estén en su rango de detección. Cada Frame object, representa un frame que contiene cualquier información recibida por el sensor, añadiendo así sus propiedades.

### Manos

El objeto 'hand' proporciona información sobre el ID, la posición y otras características de la mano detectada por el dispositivo. También nos proporciona una lista sobre los dedos asociados a la mano. Las manos se representan con la clase *Hand*.

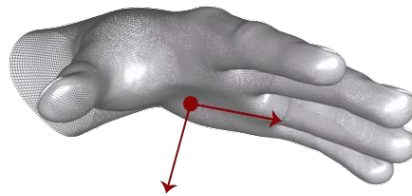


Figura 16 - Punto de detección de la clase *Hand* con Leap Motion

El software utiliza un modelo interno de una mano humana como se ha comentado anteriormente, para poder hacer un tracking predictivo incluso cuando hay partes de la mano que no son visibles. El model proporciona posiciones de los 5 dedos todo el tiempo, aunque el tracking es óptimo cuando la silueta de la mano tiene todos sus dedos visibles.

El software utiliza la parte visible de la mano visible, el modelo interno y el registro histórico del tracking realizado para calcular las posiciones de las partes no visibles de la mano. Puede aparecer más de una mano en la lista de manos en un mismo *frame* si más de una persona está dentro del rango de detección. Para obtener un tracking de calidad es recomendado no utilizar más de dos manos al mismo tiempo.

## Dedos

El controlador proporciona información sobre cada dedo de la mano. Si un dedo no es detectado del todo bien, se realizan unos cálculos predictivos para estimar dónde está basado en el historial de tracking del dedo y con el modelo interno registrado de los dedos y la mano. Los dedos son reconocidos e identificados por un nombre: *thumb*, *index*, *middle*, *ring* y *pinky*. Los dedos son representados por la clase *Fingers*.



Figura 17 - Dedos detectados por Leap Motion



## 6. PRÁCTICA

### 6.1 Desarrollo de la programación

En la instalación se encuentran diferentes dispositivos y sensores que envían hacen que el *videomapping*<sup>2</sup> proyectado sea interactivo y, en consecuencia, editable por el usuario. La misión de estos sensores es mostrar el control sobre la proyección cuando se interactúa con ellos. En esta obra tenemos tres dispositivos: dos sensores de proximidad con *Arduino*, y un Leap Motion.

La información recibida por estos sensores es enviada al procesador encargado de proyectar los audiovisuales, y con estos datos, se modifican los efectos de éstos y cargaremos archivos nuevos. A continuación, se explica con más detalle cómo afecta la interacción a la obra.

#### 6.1.1 Procesador de los sensores y dispositivos interactivos

El objetivo principal de los dispositivos interactivos es enviar la información que reciben a la proyección audiovisual que se muestra al usuario. Para recibir y analizar los datos de éstos, se ha creado un *script*<sup>3</sup> conjunto para todos los sensores. En este archivo se han creado las variables deseadas para poder guardar la información a tiempo real de lo que está sucediendo con ellos.

El *script* en cuestión, de ahora en adelante *script técnico*, es el encargado de enviar todos los datos al procesador de la proyección audiovisual. En el apartado 7, se explica cómo se ha realizado la comunicación de datos.

En este documento se explica la programación de una parte del *script técnico*, concretamente de la parte de la programación del Leap Motion además de las conexiones creadas para la comunicación.

---

<sup>2</sup> Proyectar imágenes sobre una superficie para conseguir efectos de movimiento o 3D.

<sup>3</sup> Archivos de texto ejecutados por un intérprete de código, en este caso Processing.

### 6.1.2 Receptor de datos técnicos y procesador de la proyección audiovisual

Durante la implementación de la parte interactiva de esta pieza, se ha realizado un estudio del montaje de ésta y de la logística necesaria para crear un espacio inmersivo y transmitir el concepto al usuario. Para ello, tener un ordenador para procesar cada sensor implicaría tener 3 ordenadores diferentes añadiendo uno más para la parte visual. A nivel de logística y conexión entre los ordenadores se ha querido optimizar el espacio y el uso de dispositivos y procesadores físicos. En caso de querer escalar esta pieza y añadir más sensores, surgirían problemas y dificultades que impedirían tener una buena estructura.

Para solucionar este problema, se ha decidido utilizar únicamente dos ordenadores para procesar tanto la parte técnica como la audiovisual.

El primer ordenador se utiliza para procesar lo que hemos nombrado como *script técnico*. Para ello, todos los sensores deben estar conectados a éste ordenador. Se ha realizado el estudio de cableado para poder tener las estructuras donde van colocados los sensores y donde se encuentra el ordenador de manera que sea óptimo e invisible para el usuario. Por otra parte, se utiliza el segundo ordenador para reproducir la parte audiovisual y conectarlo con los proyectores deseados.

En el segundo procesador se encuentra la parte gráfica de la pieza, y en él se ejecutará el segundo script creado, de ahora en adelante, *script gráfico*.

El *script gráfico* se encarga de recibir toda la información obtenida de los sensores que es enviado por el *script técnico*. Este script se comunica con el programa *Resolume*, utilizado para realizar el *videomapping*.

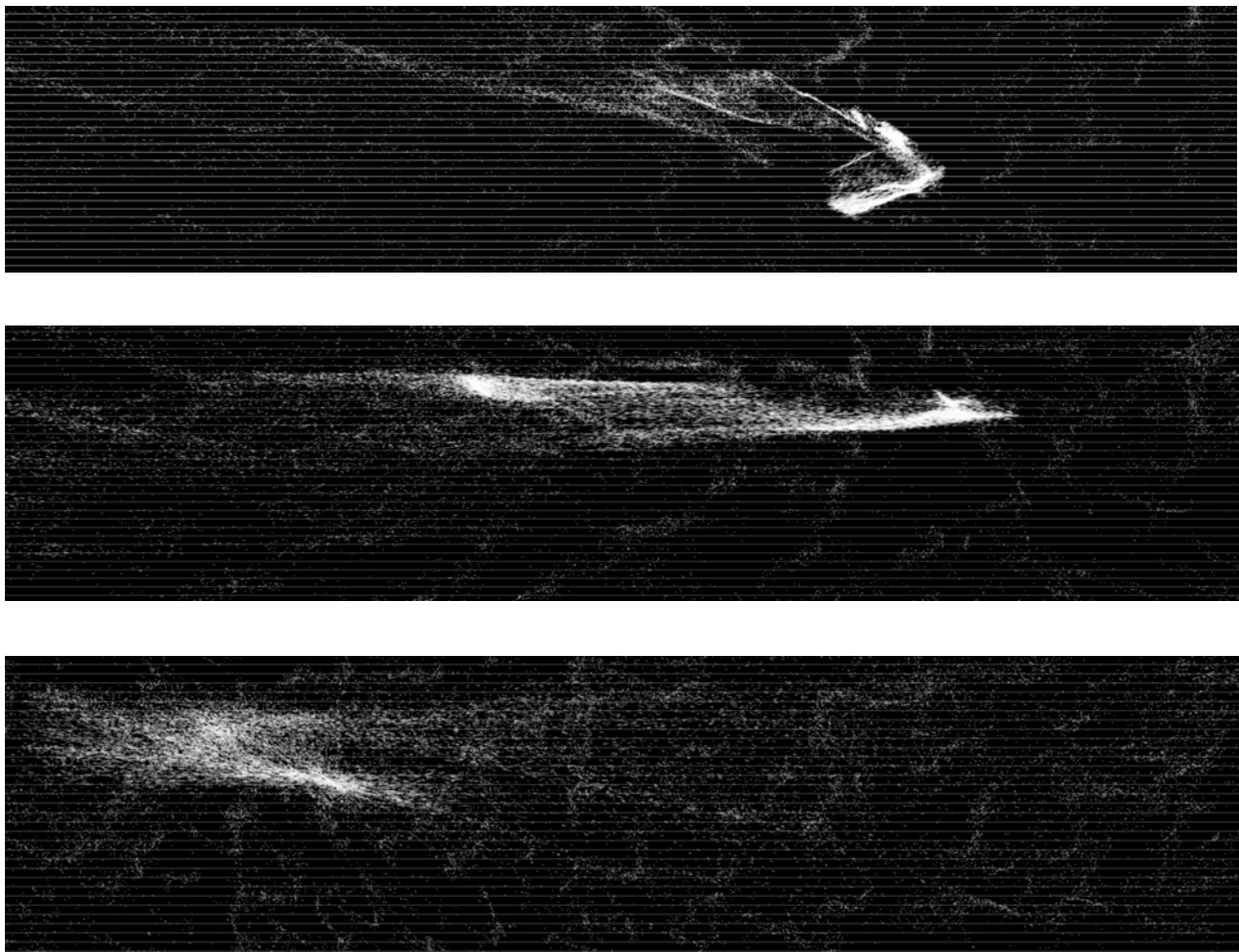
Para poner un ejemplo sobre lo explicado anteriormente, se explica el procedimiento realizado para hacer el seguimiento de las partículas con la mano del usuario.

Para empezar, el Leap Motion conectado al PC con el script técnico, le envía los datos de tracking de cada frame sobre la mano que está en su rango de detección. Para empezar, se va a utilizar solo el valor x e y de la mano.

Una vez el script técnico tiene estos dos valores, se los envía al PC donde se encuentra el script gráfico. Cuando éste lo ha recibido, utiliza estos datos para mover las partículas en función del parámetro x e y.

```
float d = dist(width/m,height/m,mouseX/m,mouseY/m);  
xv += 0.001*(movimientoX-x)*pow(d, w)*w;  
yv += 0.001*(movimientoY-y)*pow(d, w)*w;
```

De esta manera, cuando se mueva la mano delante del Leap Motion, estos datos los recibirá la parte gráfica como la variable que se ve en el código 'movimientoX' y 'movimientoY', y el mismo processing que dibuja las partículas actualizara el valor de su posición, como se puede observar en las siguientes figuras 18, las cuales se ve que las partículas siguen la posición del ratón:



*Figura 18 - Movimiento de partículas con el ratón en Leap Motion*

## 6.2 Comunicación

Antes de explicar la comunicación entre los scripts de los ordenadores que se utilizan para el funcionamiento de la pieza, se añade un esquema general de los dispositivos y conexiones:

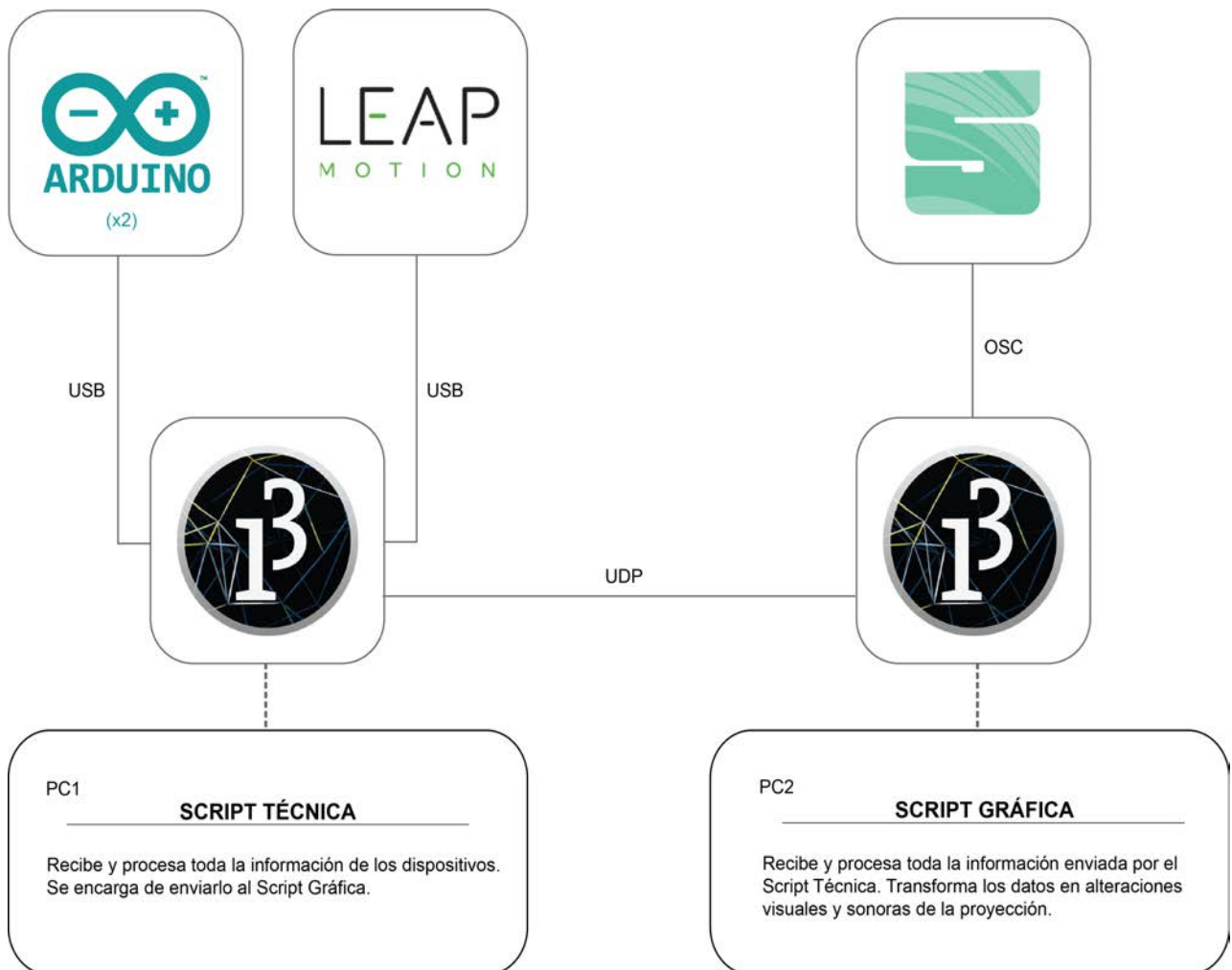


Figura 19 - Esquema gráfico de comunicación y configuración de la programación

En este esquema se observa que se tienen 3 sensores (dos Arduinos y un Leap Motion), un *script* nombrado 'técnica', que procesa toda la información de estos sensores desde un ordenador y Processing. Esta información es enviada al segundo PC con el segundo *script* nombrado 'gráfica', y éste último lo utiliza para realizar las proyecciones. Se explica con más detalle:

## 6.2.1 Comunicación entre técnica – gráfica

Para la comunicación entre los scripts, en concreto el script gráfica y el script técnica, se ha utilizado el protocolo de comunicación UDP. Las librerías que se añaden en el proyecto son:

```
import hypermedia.net.*; //UDP
import processing.serial.*; //Serial
```

Una vez se han importado las librerías necesarias se crean los puertos y variables necesarias para enviar a la parte gráfica los valores a modificar:

```
//Puertos locales por los que se envían los datos
int PuertoLocal1 = 7100; //arduino1
int PuertoLocal2 = 7200; //arduino2
int PuertoLocal3 = 7300; //leapX
int PuertoLocal4 = 7400; //leapY
int PuertoLocal5 = 7500; //leapZ
int PuertoLocal6 = 7600; //HAND-DETECTED

//Puerto que recibe datos en PC de gráficos
int puertoRemoto = 6100;

//Direccion ip de la computadora a la que le mandaremos mensajes
String ipRemota = "192.168.0.101";

//Conexiones UDP
UDP udp1, udp2, udp3, udp4, udp5, udp6;
```

Se crean conexiones UDP para enviar a pc de gráficos:

```
udp1 = new UDP( this, PuertoLocal1 );
udp2 = new UDP( this, PuertoLocal2 );
udp3 = new UDP( this, PuertoLocal3 );
udp4 = new UDP( this, PuertoLocal4 );
udp5 = new UDP( this, PuertoLocal5 );
udp6 = new UDP( this, PuertoLocal6 );
```

Se crea el valor a String para ser enviado:

```
String message6 = str(m6);
udp1.send( message1, ipRemota, puertoRemoto ); //arduino1
udp2.send( message2, ipRemota, puertoRemoto ); //arduino2
udp3.send( message3, ipRemota, puertoRemoto ); //leapX
udp4.send( message4, ipRemota, puertoRemoto ); //leapY
udp5.send( message5, ipRemota, puertoRemoto ); //leapZ
udp6.send( message6, ipRemota, puertoRemoto ); //hasHands
```

Ahora se muestra cómo recibe la parte gráfica estos elementos. Primero se importan al proyecto las librerías necesarias:

```
import hypermedia.net.*; //UDP
```

Se definen los objetos UDP y el puerto local y remotos (los mismos que se han decidido para el script técnica):

```
UDP udp;
//Puerto local por el que se reciben datos de sensores
int puertoLocal = 6100;
//Puertos remotos de pc de sensores
//Verificar que sean los mismos que en el otro código
int PuertoRemoto1 = 7100; //arduino
int PuertoRemoto2 = 7200; //arduino
int PuertoRemoto3 = 7300; //leap
int PuertoRemoto4 = 7400; //leap
int PuertoRemoto5 = 7500; //leap
int PuertoRemoto6 = 7600; //leap hand detected y/n -> 0/1
```

Se crea una conexión UDP para recibir la información de los sensores:

```
udp = new UDP( this, puertoLocal );
udp.listen( true );
```

Función que se encarga de recibir la información del script técnica:

```
void receive( byte[] data, String ip, int port ) {}
```

## 6.2.2 Comunicación entre la parte gráfica y Resolume

Para enviar la información de Processing a Resolume, se utiliza OSC (Open Sound Control) y Spout. En un inicio se utilizó Syphon, pero al cambiar de ordenador de MAC a WINDOWS, se descubrió que este no era compatible con el sistema operativo y se cambió por Spout.

Centrado en el script gráfica, se ha importado la librería de Spout, y se realiza la conexión con Resolume:

```
import spout.*; //Spout (como syphon para windows)
```

Para conexión OSC con Resolume se define el IP del ordenador que va a recibir los datos.

```
String ipAddressToSendTo = "localhost";
OscP5 oscP5;
NetAddress myRemoteLocation;
OscBundle myBundle;
OscMessage myMessage;
```

Se define el puerto local que recibe de Resolume, y el puerto que envía a Resolume. Estos parámetros también se configuran en el mismo programa Resolume.

```
//PuertoLocal que recibe de Resolume
int portToListenToResolume = 7001;
//Puerto que envía a Resolume
int portToSendToResolume = 7000;
```

Se crea una conexión OSC para enviar datos a Resolume:

```
//Se crea conexión OSC para enviar ordenes a resolume
oscP5 = new OscP5(this, portToListenToResolume);
myRemoteLocation = new NetAddress(ipAddressToSendTo, portToSendToResolume);
myBundle = new OscBundle();
myMessage = new OscMessage("/");
spout = new Spout(this);
spout.createSender("SpoutProc");
```

Se genera el mensaje para Resolume:

```
myMessage.setAddrPattern("/layer4/clip2/connect");
```

```
        //myMessage.add(map(float(val2), 0.0, float(MAX_DISTANCE_ARDUINO), 0.0, 1
    .0));
    myMessage.add(1);
    myBundle.add(myMessage);
    myMessage.clear();
```

Una vez el mensaje es capturado, se envía a Resolume:

```
oscP5.send(myBundle, myRemoteLocation);
myBundle.clear();
```



## 6.3 Alteración de efectos audiovisuales con Leap Motion

### 6.3.1 Visual

El Leap Motion tiene el control sobre dos tipos de gráfica dentro de la proyección. La primera parte está relacionada con las proyecciones que se realizan con Resolume modificando los parámetros de los efectos de los videos, y la otra será directamente con la gráfica creada con el mismo Processing. Antes de proceder a explicar la alteración de la parte visual, se define la configuración del Leap Motion dentro de processing.

#### Configuración de Leap Motion en processing

Dentro del script técnica, que es donde se procesa toda la información de los sensores, se añade la librería de Processing y se crea el objeto.

```
import de.voidplus.leapmotion.*; //Leap
leap = new LeapMotion(this);
```

Como el interés principal es obtener un tracking de las manos del usuario, se crean tres variables donde se guardarán las posiciones x, y, z de la mano.

```
float posX;
float posY;
float posZ;
```

Un factor importante que se ha tenido en cuenta en la programación del Leap Motion, es saber en todo momento si se encuentra una mano dentro del rango de detección del dispositivo o no, ya que se han encontrado problemas con las proyecciones si la mano sale de éste rango a nivel de visualización. El problema principal es la diferencia de la proyección cuando el usuario está interactuando con la pieza y cuando no. En caso de no detectar bien si el usuario ha dejado de interactuar con el dispositivo, es decir, ha salido del rango pero Processing se ha quedado con la última posición como si estuviera dentro, genera un error conceptual. Para solucionar este problema, se ha utilizado la siguiente función que informará a la parte gráfica si no hay ninguna mano o si la hay.

```
leap.hasHands()
```

Para obtener las coordenadas de la posición de la mano, se ha creado el siguiente algoritmo:

```
for (Hand hand : leap.getHands ()) {  
    PVector handPosition = hand.getPosition();  
    posX = handPosition.x;  
    posY = handPosition.y;  
    posZ = handPosition.z;  
}
```

### Alteración de los efectos

Una vez estas variables son enviadas al script gráfica mediante UDP, ésta las envía a Resolume para modificar los parámetros de las proyecciones.

```
myMessage.setAddrPattern("/layer4/clip1/connect");
```

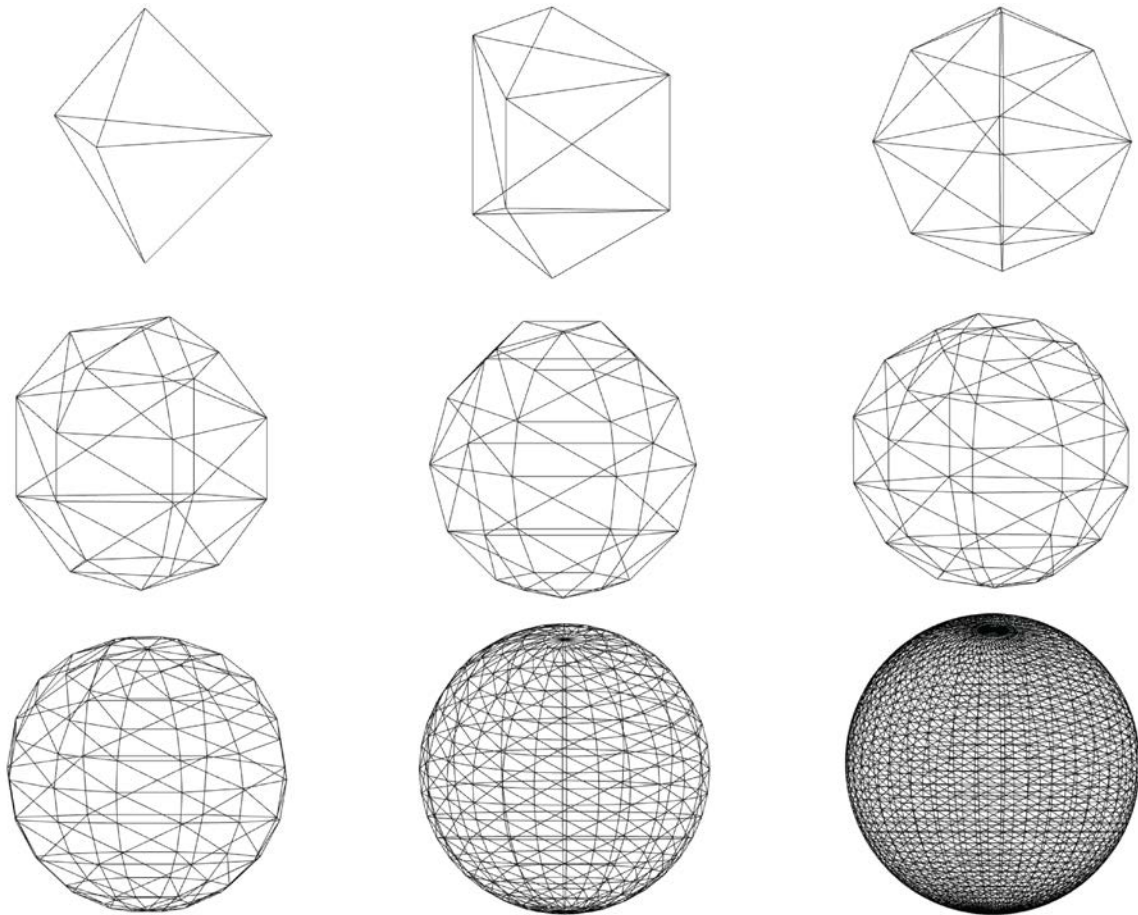
Por otra parte, en función de estos parámetros se crea y se modifica una nueva parte gráfica independiente a Resolume. La parte gráfica controlada por el Leap Motion trata de **la pieza central de toda la proyección**. Formada por una esfera central creada con processing y partículas, generará un control total cuando ésta esté activada junto a los dos Arduinos.

La creación de la esfera en processing, utilizando la función sphere(), trata sobre la teselación de triángulos para formar así dicha forma. Una teselación (también conocido como teselado) consiste en el recubrimiento de una superficie plana por medio de figuras de tal forma que no exista ningún hueco entre una figura y otra, y que las figuras estén dispuestas sin superponerse unas sobre otras. Una pieza es teselante cuando es posible acoplarla entre sí con otras idénticas a ella sin huecos ni fisuras hasta recubrir por completo el plano. Como más figuras se utilicen, más real será el renderizado.

Se analiza el comportamiento de processing cuando se altera el número de figuras que se desea que tenga la figura central: Se crea un código básico para ver los diferentes resultados y la evolución de la esfera al añadir más triángulos teselados:

```
void draw() {  
    background(255);  
    noFill();  
    stroke(0);  
    pushMatrix();  
    translate(500, height*0.35, -200);
```

```
sphereDetail(50);  
sphere(280);  
popMatrix();  
}
```



*Figura 20 - Representación de la esfera central con la variación del número de triángulos teselados*

Esta esfera, Será alterada en función del parámetro que contiene la coordenada 'y' del tracking, y del nivel en el que se encuentre, la cual indica cómo de cerca o lejos está la mano del dispositivo explicado en la parte teórica.

El número de triángulos de la esfera será alterado de la siguiente manera: como más lejos se encuentre del Leap Motion, menos polígonos tendrá, y como más cerca, mayor será el valor formando así una esfera completa. Esto transmite al usuario un control sobre la forma geométrica que se muestra en el centro de la pantalla. Se utiliza el siguiente comando para modificar los triángulos:

```
sphereDetail(nivel_Y);
```

Cuando el usuario no interactúa con el dispositivo, la forma central tendrá unos valores aleatorios que harán que ésta se mueva con descontrol y desorden. Una vez la mano es detectada por el dispositivo, la velocidad de este movimiento aleatorio se reduce a 0.

Además, se añade un segundo parámetro a esta interacción. Cuando el usuario mueva la mano horizontalmente (eje X) la figura central será rotada en función de su movimiento, proporcionando así un control total sobre la figura, a nivel de polígonos y de movimiento.

Se añade un tercer factor que alteramos con el Leap Motion, para proporcionar un control completo sobre la pieza. Éste es explicado en el siguiente punto, ya que se trata del sonido.

Función que controla la parte visual con processing:

```
void dibujar()
```

Cuando el dispositivo no detecta ninguna mano, los valores se resetean al valor inicial (random) para volver al descontrol de la proyección.

```
rotateY(random(-100,100));
rotateX(random(-100,100));
sphereDetail(1);

//se convierte a tamaño de pantalla
movimientoX = (int)map(float(val3), 0.0, float(MAX_DISTANCE_LEAPX), 0.0,
1920);
movimientoY = (int)map(float(val4), 0.0, float(MAX_DISTANCE_LEAPY), 0.0,
530);
movimientoZ = (int)map(float(val5), 0.0, float(MAX_DISTANCE_LEAPZ), 0.0,
530);
background(0);

//control movimiento esfera
```

```

float rx = (movimientoX-(width/2))*0.005;
float ry = (movimientoY-(height/2))*0.005;
rx = rx*0.9;
ry = ry*-0.9;
//posición esfera
translate(width/2, height/2);

```

En el momento de decidir qué dispositivo era el óptimo para hacer alteraciones gráficas, se decidió que Leap Motion era la mejor solución básicamente por la fácil integración de todos los gestos detectables que permite implementar el dispositivo.

La idea original era detectar algunos movimientos básicos y familiarizados por los usuarios explicados en la Sección 1. Partiendo de esta idea, la intención era cargar un tipo de vídeo cada vez que el usuario realizaba uno de los gestos definidos, proporcionando así control sobre las proyecciones. Ejemplo: Si el usuario realiza un *Swipe* hacia la derecha, una cortina de partículas se deslizaría por la pantalla de izquierda a derecha, haciendo saberle así que estas partículas han estado generadas por su gesto.

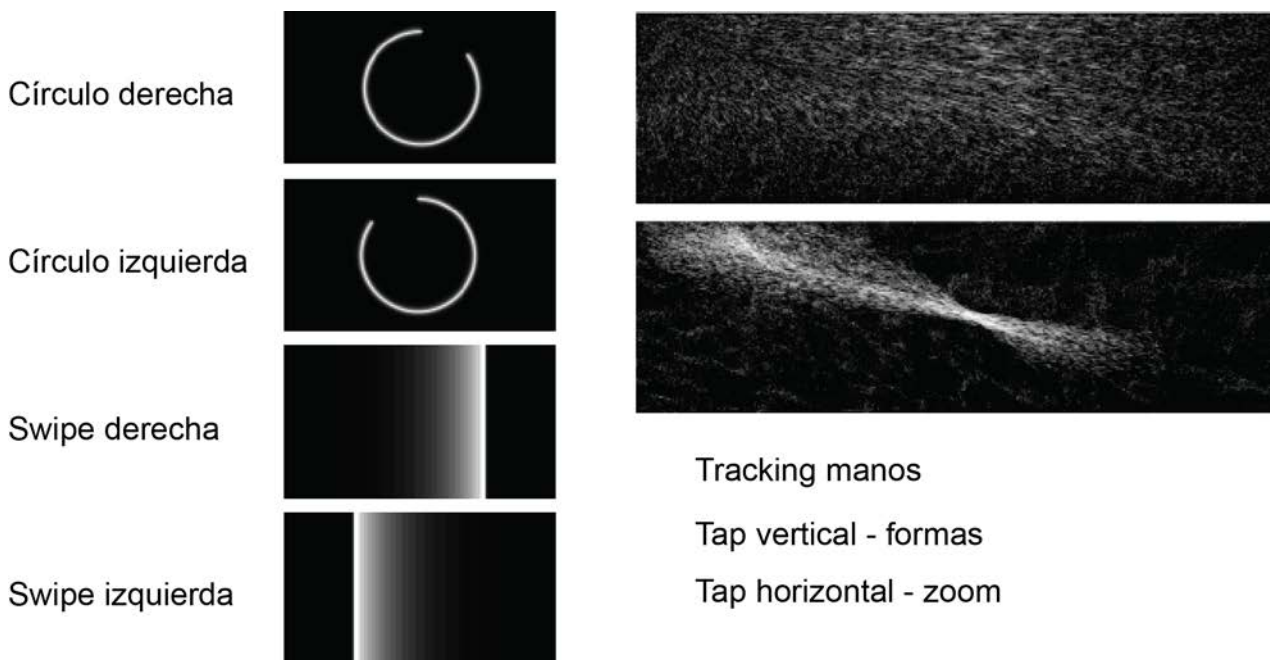


Figura 21 - Esquema de interacciones con Leap Motion y sus respuestas

Como se observa en la Figura 21, según el tipo de gesto detectado por el usuario cargaba un fichero u otro de vídeo sobre la proyección. Además, se creaba un sistema de partículas que seguían la trayectoria de la mano del usuario.

Esta idea fue presentada en el prototipo del proyecto, pero al testear su funcionamiento con los usuarios se comprobó que proporcionaba más descontrol que control, ya que no era una alteración de la proyección sino un añadido. Esto implicaba una sobrecarga de proyección, ya que el usuario veía demasiadas cosas en la pantalla cuando interactuaba y justamente se pretende ofrecer el efecto contrario, de limpieza y control.

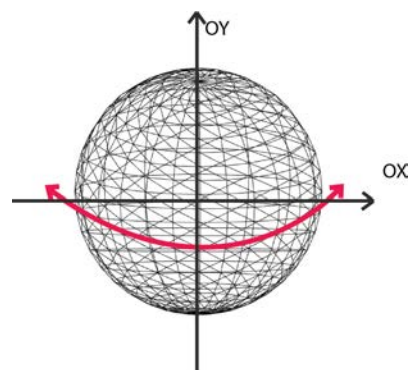
Tampoco favorece al proyecto añadir muchos gestos con el Leap Motion. Partiendo de la base que los otros dos dispositivos de la pieza son sensores de proximidad, algo muy simple para el usuario (lejos, cerca), añadir muchos gestos para el usuario era una distracción e incomprensión de lo que estaba pasando. El proceso de aprendizaje de la pieza se volvía en algo complejo y de difícil comprensión para los usuarios.

Por estos motivos, se replanteó el objetivo principal del Leap Motion. De igual manera que la interacción con los Arduinos, la interacción con este dispositivo debía ser simple.

Aprovechando que Leap Motion permite hacer un *tracking* continuo sobre las manos que entran en su rango de detección, se decidió darle un valor añadido respecto al Arduino y no solo detectar si está cerca o lejos del usuario, sino aprovechar todo el rango de detección 3D.

El gesto que tiene que realizar el usuario es simple: colocar la mano sobre el Leap Motion, y moverla. En función del movimiento de la mano las reacciones son distintas, y ahí entra el aprendizaje simple del usuario: ver en qué está afectando los movimientos que realiza.

Cuando el usuario mueve la mano en el eje horizontal X, se altera la rotación de la figura central además de eliminar su movimiento aleatorio.



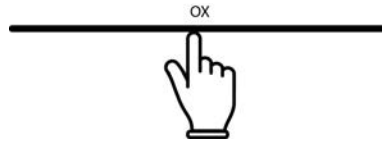


Figura 22 - Interacción en el eje de coordenadas OX

Cuando el usuario mueve su mano en el eje vertical OY, modifica el número de secciones que forman la pieza central y además el sonido comentado en el siguiente punto.

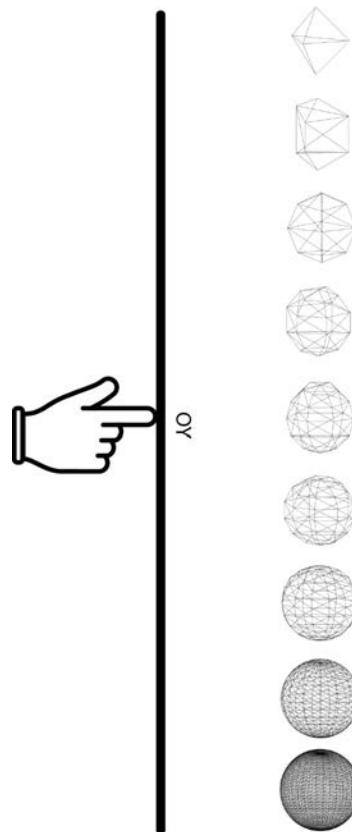


Figura 23 - Interacción en el eje de coordenadas OY

De esta manera se proporciona al usuario diferentes acciones que afectan a la proyección, y solo poner la mano en su rango de detección ya se pueden observar alteraciones.

Para alterar parte de Resolume con Leap Motion, los eventos lanzados al principio del proyecto trataban de cargar clips de video que proporcionaban creación de movimientos a nivel visual sobre las partículas que aparecían por pantalla.



Figura 24 - Desarrollo del tracking de la mano para el seguimiento de partículas con Processing y Leap Motion

La función para lanzar estos eventos es la siguiente:

```
myMessage.setAddrPattern("/layer4/clip1/connect");  
myMessage.add(1);  
myBundle.add(myMessage);  
myMessage.clear();
```

Esta función se encarga de reproducir el *clip1* de la *capa4* de Resolume. Esta acción se realizaba cuando se recibía por parte del Script Técnica, un Swipe hacia la derecha. En tal caso, se cargaba el clip correspondiente a ese efecto.



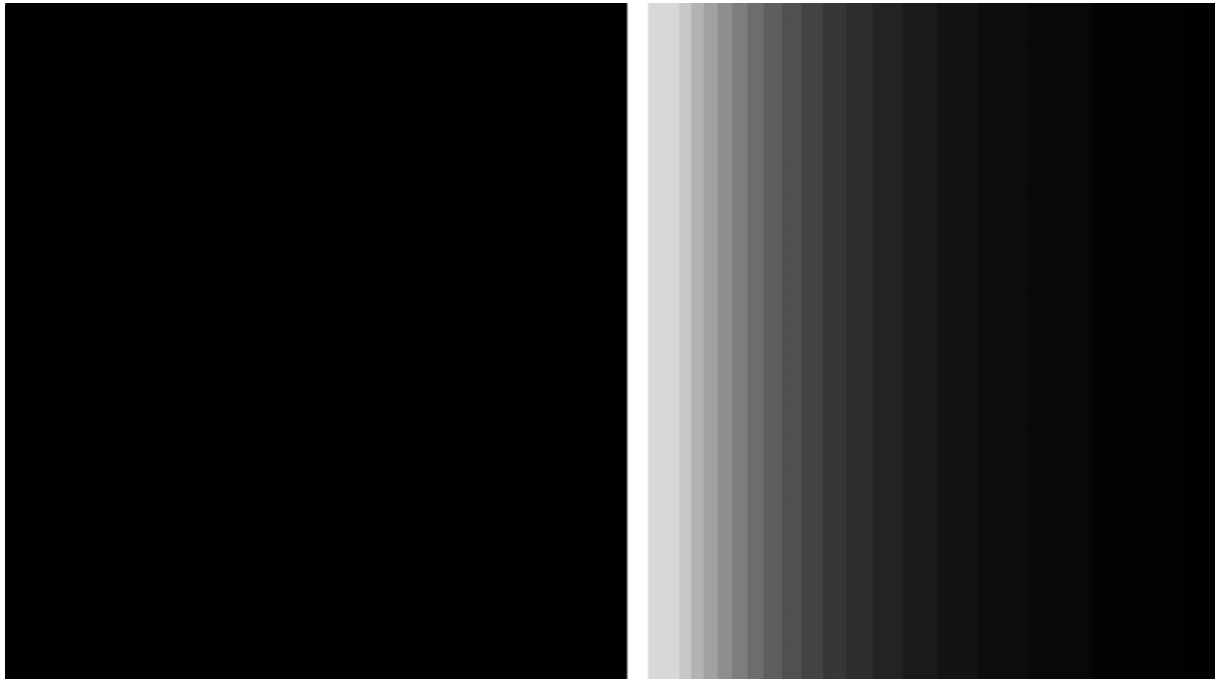


Figura 25 - Efecto de vídeo cargado cuando se recibe un Swipe hacia la izquierda

Este mismo proceso se replicó para todos los gestos planteados en el prototipo, que como se ha comentado anteriormente, finalmente se decidió no añadirlos al proyecto final.

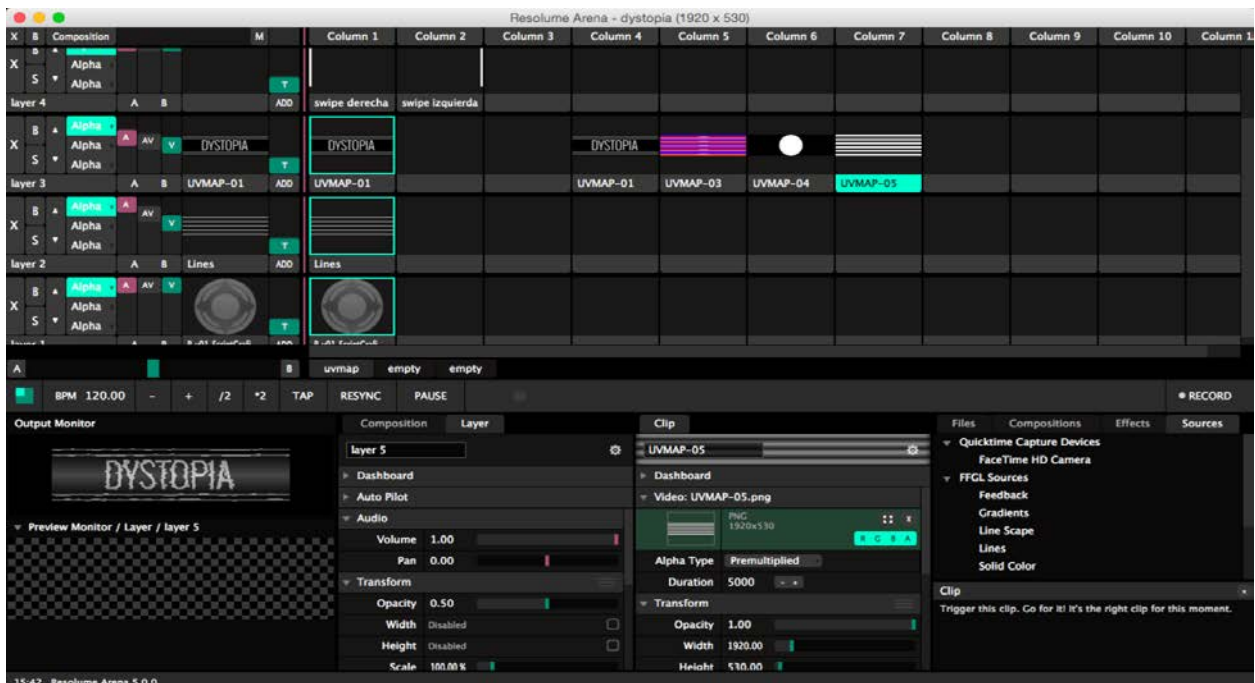


Figura 26 - Captura de pantalla del programa Resolume

### 6.3.2 Generación de audio con Processing

Una parte significativa de la pieza además de las proyecciones visuales, es el audio que la acompaña.

Los sonidos principales reproducidos durante la proyección, son ruidos molestos para el usuario, produciendo así sensación de querer pararlos para volver a un estado de paz y tranquilidad. De esta manera se incita al usuario querer interactuar con los dispositivos y querer tenerlos activados para eliminar este ruido.

Por otra parte, se genera un tipo de sonido distinto cuando se interactúa con el Leap Motion. La función principal de los otros dispositivos es anular los sonidos molestos comentados, en cambio con el Leap, además de esto se generan unos sonidos harmónicos que permiten al usuario generar melodías con el movimiento de sus manos.

El funcionamiento de esta fase interactiva se explica a continuación.

#### Generación de sonido con Leap Motion

Para generar el efecto de la escala harmónica con el Leap, primero se diseña cómo serán estos sonidos.

Se crea una escala de 10 notas independientes, en una escala ordenada a nivel harmónico. Como si de un piano y sus teclas se tratase, la intención es crear el mismo efecto, pero en una escala con teclas invisibles activadas por el movimiento realizado con la mano del usuario y el dispositivo.

La configuración del Leap Motion para la pieza se ha programado tal que la distancia en el eje Y (altura) sea de 50 cm. Basado en esta escala, Leap Motion detecta el eje Y en una escala de píxeles, así que se realiza la obtención del dato y su respectiva conversión:

El script técnica se encarga de recibir la posición en el eje Y:

```
posY = handPosition.y;
```

Una vez se ha comprobado que el dato obtenido es el correcto, se envía al script de gráfica.

El script de gráfica, una vez recibe la variable enviada por el script técnica el cual recibe directamente la información del Leap Motion, es el encargado de escalar este valor a una nueva escala. Esta escala es la configurada e igualada al tamaño en píxeles de la pantalla para tener un resultado óptimo al afectar tanto la parte gráfica como la sonora, e igualarla a los otros dispositivos. Para ello, se ha utilizado la siguiente función:

```
movimientoY = (int)map(float(val4), 0.0, float(MAX_DISTANCE_LEAPY),
0.0, 530);
```

Como se puede observar en la función, se utiliza `map()`, y se le pasa el valor a escalar, en este caso, `val4` que es el que contiene la variable de la posición de la mano del usuario en el eje Y. Una vez aplicada esta función, el valor de la variable oscilará entre 0 y 530. Fuera de este rango, se considerará que la mano no está interactuando con el sensor.

Para dividir el espacio de interacción en 10 niveles diferentes, se ha aplicado la siguiente división después de hacer pruebas y testeos: 53 píxeles (una vez se ha aplicado la normalización del valor comentada anteriormente) es la distancia que separa un nivel del otro, que equivale a unos 10cm a nivel físico real.

La función utilizada para reproducir el fichero de sonido correspondiente al nivel en el que se encuentra la mano del usuario es la siguiente:

```
if (nota != 1 && movimientoY <= 106){
    nota = 1;
    player1.rewind();
    player1.play();
    sphereDetail(1);
}
```

En esta función, aparece la variable *nota*. Esta variable se encarga de controlar el sistema de reproducción de los audios. Su razón de estar, es saber cuál es la última nota reproducida por el usuario, a lo que equivale saber en qué nivel estaba su mano en el último momento que entró en el bucle. Se quiere conocer esta información, ya que de otro modo estaría reproduciendo sin parar ésa misma nota percibiendo que siempre está en ese nivel, y este factor causa una inestabilidad en la pieza, ya que el tiempo en volver a entrar en la misma función es muy pequeño,

y el efecto producido sería similar al de presionar repetidas veces y a una velocidad muy elevada la tecla de un piano.

El resultado que se quiere obtener es agradable, armónico y de control. Por este motivo se limita esta interacción pudiendo así reproducir una nota diferente cada vez. Sólo se activará un sonido cuando este sea diferente al anterior. Si el usuario deja la mano quieta en un mismo nivel, el audio correspondiente solo será reproducido una vez.

De esta manera, se repite la condición anterior por cada nivel, que como se ha comentado anteriormente, en esta pieza tenemos 10. Los valores alterados entre una condición y otra serán la nota, el rango del *movimiento* Y y la pista a reproducir.

Como se ha visto en el punto anterior, también es afectada la parte visual central de la pieza. Por este motivo, se ha decidido que como más cerca se encuentra la mano del dispositivo, más control tiene el usuario sobre la pieza. Por consecuencia, más forma va a tener y más armonía va a generar. Podemos comprobar el cambio de la forma central ya que en la condición anterior se ve que aparece la función `sphereDetail()`, la cuál altera el detalle de la esfera central. Solo se realiza esta alteración ya que existe otra para controlar si la mano está dentro del rango de detección del Leap Motion o no, y en caso de estar fuera del rango, se le asignan los valores aleatorios.

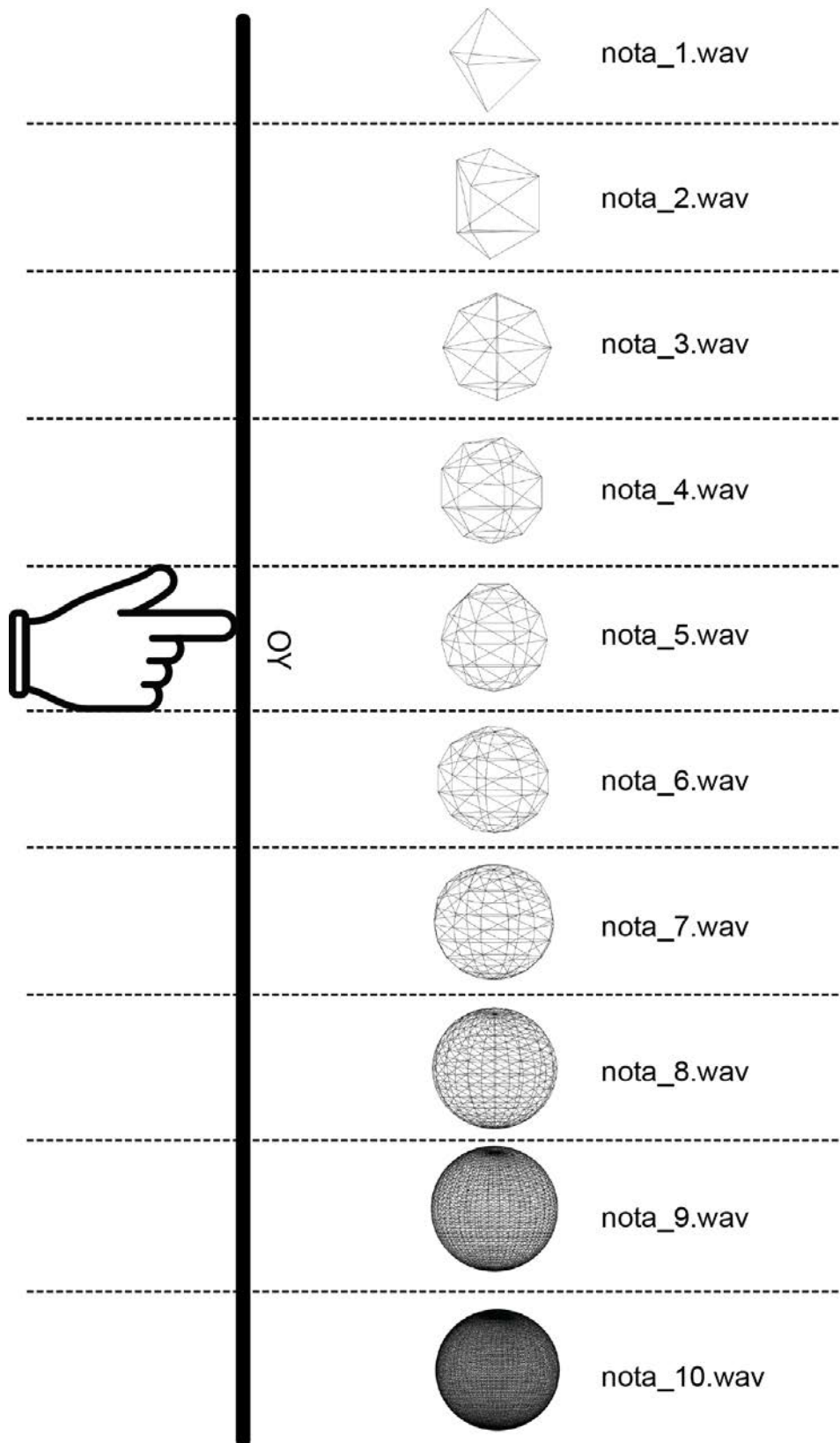


Figura 27 - Representación de la interacción con Leap Motion

Para poder reproducir un sonido con Leap Motion, se requiere de una librería. Después de realizar pruebas con la librería de processing 'sound', y hacer pruebas con la librería de processing 'minim', se ha comprobado que ésta última no daba problemas al cambiar de ordenador y sistema operativo. El mismo estudio tuvo que ser realizado para la librería de comunicación con Resolume: cuando se realizó toda la programación con MAC funcionaba todo correctamente. Al utilizar el PC Windows de la sala, se obtuvieron problemas ya que algunos plugins y librerías solo sirven para un sistema operativo concreto y en este mismo caso se tuvo que cambiar por *spout*.

Una vez se ha instalado la librería *minim* en processing, se han realizado los siguientes pasos: Primero de todo se crean las variables para cargar los archivos de sonido necesarios. Mostramos como ejemplo de código para cargar y modificar siempre el primer audio, luego este mismo proceso se repite para los 9 siguientes.

```
player1 = minim.loadFile("esferal.wav");  
input = minim.getLineIn();
```

Una vez se han cargado los 10 archivos de sonido en processing, dependerá del usuario decidir cuál se reproduce.

En función de dónde se coloca la mano, se detecta en qué nivel está con el algoritmo que se adjunta a continuación:

```
if (nota != 1 && movimientoY <= 106){  
  nota = 1;  
  player1.rewind();  
  player1.play();  
  sphereDetail(1);  
}
```

En este primer caso, el usuario se encuentra en el nivel 1. Esto significa que su mano se encuentra en el punto más alejado de detección del Leap Motion, que equivale a nivel emocional al que menos control produce, pero aún así seguirá siendo control.

Cuando el usuario introduce su mano en esta zona, el valor de la variable *nota* explicado anteriormente será el del nivel al que se encuentra, en este caso el 1. A continuación, se rebobinará el archivo de sonido y luego se reproducirá desde el principio. Este paso se hace ya

que el archivo una vez se reproduce no se vuelve a reproducir, ya que queda parado en el punto final o si el usuario cambia de zona antes de que se acabe, parado en ese punto. De esta manera utilizamos la función `rewind()`, para volver al punto inicial del archivo y que la próxima vez que se active, sea desde el principio.

Se comprende observando el algoritmo, que, si no estuviera la variable de control *nota*, estaría continuamente rebobinando y reproduciendo el archivo de audio si mantuviera la mano quieta en ese nivel. Por eso, si el usuario decide quedarse quieto en un punto solo reproduciremos el audio una sola vez hasta que éste decida cambiar de zona o sacar la mano del rango de detección del Leap.

## 6.4 Exposición

En este apartado se analiza el comportamiento de los usuarios durante la exposición del proyecto en el Convent de Sant Agustí, y los resultados obtenidos.



*Figura 28 - Render con la distribución de la sala*

### 6.4.1 Exposición y comportamiento con usuarios

En el mes de julio de 2017, en concreto los días 12, 13 y 14 la obra estaba expuesta en el Convent de Sant Agustí, Barcelona.

Para el montaje de la pieza y el uso del material del local, se precisó de los 3 días anteriores para comprobar el correcto funcionamiento de todos los dispositivos y su proyección tanto auditiva como visual.

Como se puede observar en el video presentado, y como se explica a continuación, se observa que se tuvo un público muy variado durante la exposición de la pieza durante los 3 días. Asistieron desde niños hasta personas mayores, expresando comportamientos distintos que se comentan a continuación. Una vez el usuario entra dentro de la sala donde está expuesta la pieza, se encuentra con una pantalla de 180° y tres estructuras que la rodean por la parte frontal.



En el centro, se encuentra la pieza principal de la interacción, que como se ha explicado en las diferentes secciones de este trabajo, ha sido el Leap Motion.

En este dispositivo es donde los usuarios pasaban más tiempo, teniendo así más sensación de control sobre la pieza.

Los otros dos dispositivos se encuentran a los laterales del Leap Motion, con unas estructuras semejantes a la principal. El esquema del montaje de la pieza se observa en la siguiente Figura 29:

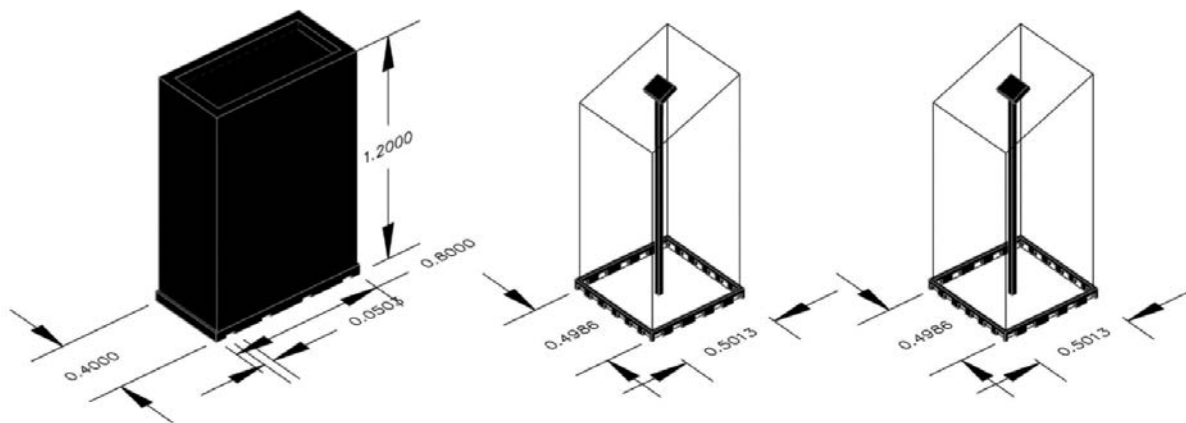
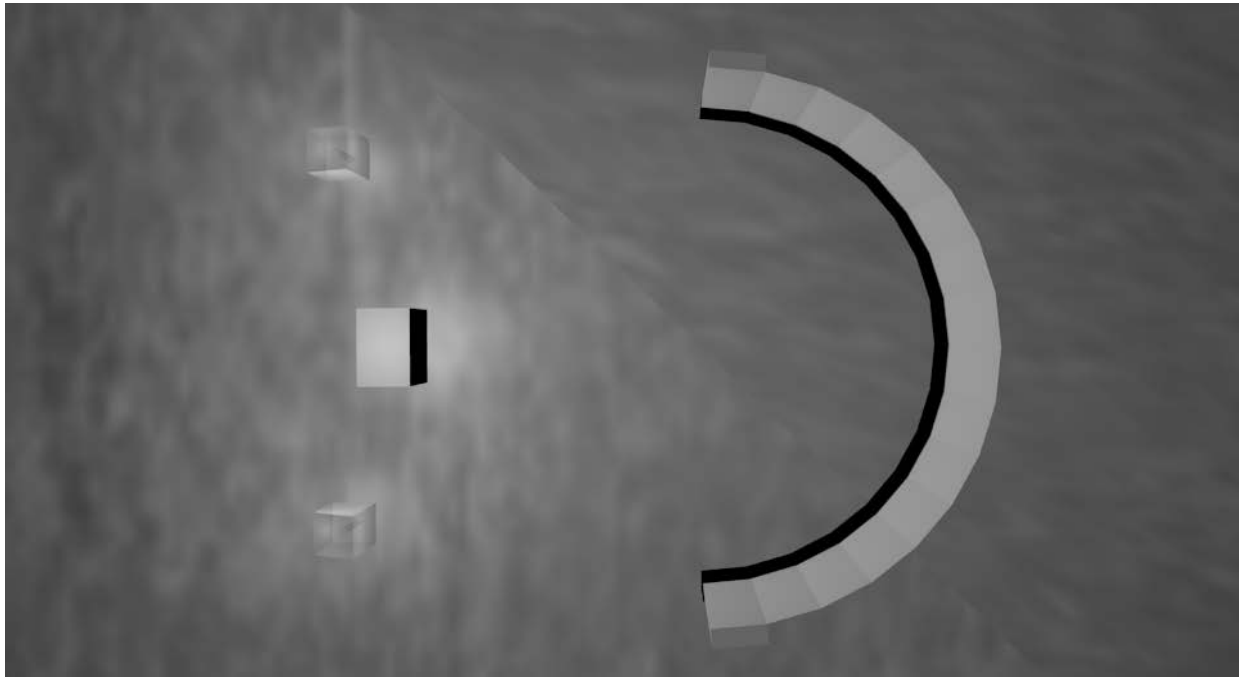
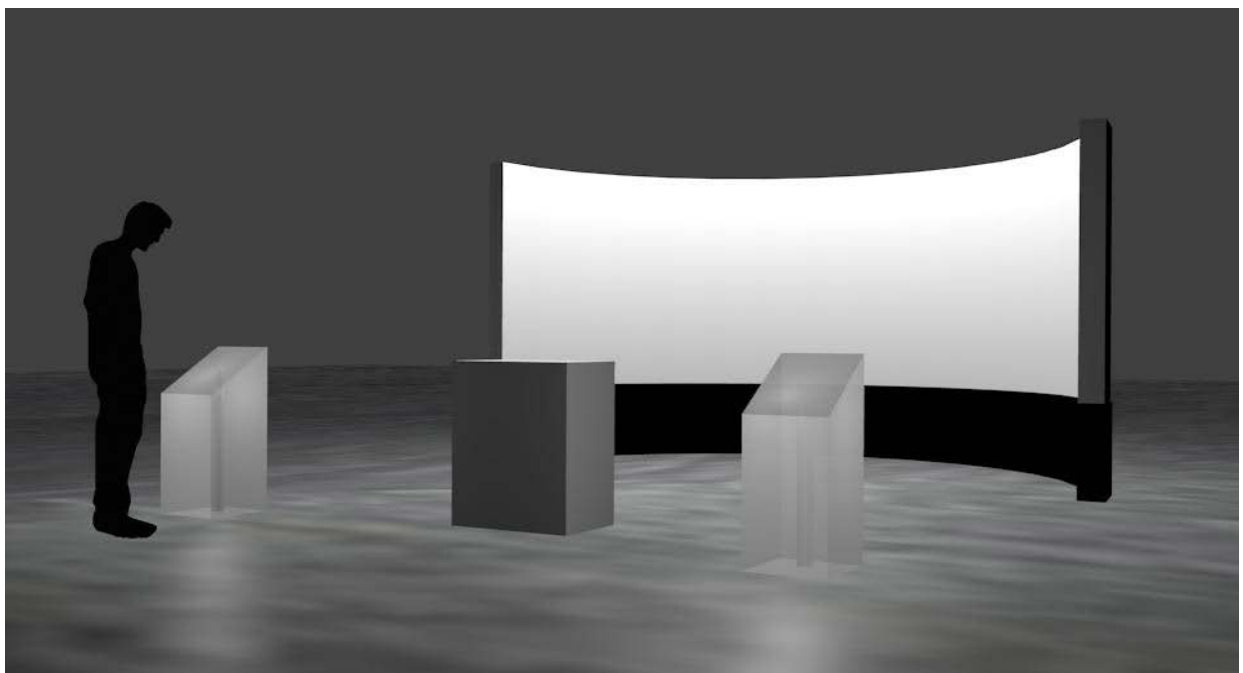


Figura 29 - Estructuras para ocultar los sensores

La primera estructura tiene una altura de 1.2m, y además tiene una apertura en la parte trasera para poder poner el ordenador que procesará toda la parte técnica de interacción de los sensores. Las tres estructuras están hechas con material transparente recubierto de papel translúcido, para poder colocar una linterna con LEDs dentro y que ésta quede iluminada. Se observa en la figura 30 cómo queda la estructura:



*Figura 30 - Render con vista cenital de la sala de exposición*



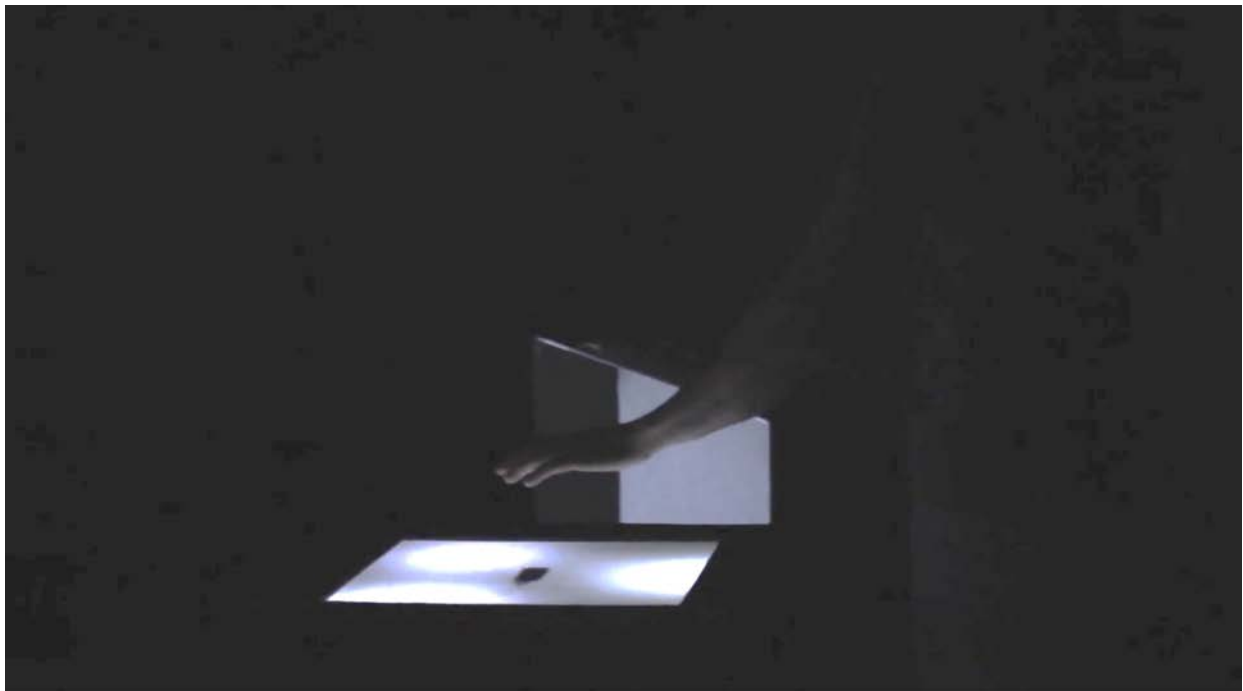
*Figura 31 - Render con vista semi-lateral de la sala de exposición*

Las personas de una edad superior a los 50 años, expresan cierto miedo o vergüenza cuando tratan de interactuar con los dispositivos. Su comprensión del funcionamiento de la pieza es más lento o incluso no satisfactorio al no probar lo suficiente o tener miedo a hacerlo mal.

Se puede observar que estos usuarios trataban de imitar los movimientos de otros usuarios antes que ellos, pero el aprendizaje por si solos dio resultados menos satisfactorios.

Por otra parte, se observó que como más jóvenes eran los usuarios menos miedo expresaban cuando tenían que interactuar con los sensores, y como más efectos veían que producían en la pieza cuando interactuaban con ellos, más gestos intentaban probar.

El dispositivo principal y con más efecto sobre la pieza por si solo es el Leap Motion, gracias a su generación de notas musicales con el movimiento de la mano en el eje vertical. Al ser un gesto simple y con un resultado muy directo, los usuarios pasan más rato interactuando con él. De manera distinta, los Arduinos eliminaban los sonidos molestos y también era una manera de llamar la atención de los usuarios, ya que se producía tranquilidad y control dentro de la sala.



*Figura 32 - Fotografía de la exposición donde se observa la interacción con Leap Motion*

La diferencia entre los usuarios afirmó la teoría estudiada en el momento de elegir los dispositivos que se querían utilizar para la pieza. La mayoría de usuarios aprendía a utilizar el concepto lejos-cerca sobre los dispositivos. Solo aquellos más jóvenes intentaban interactuar de manera distinta utilizando gestos o más de una mano a la vez.



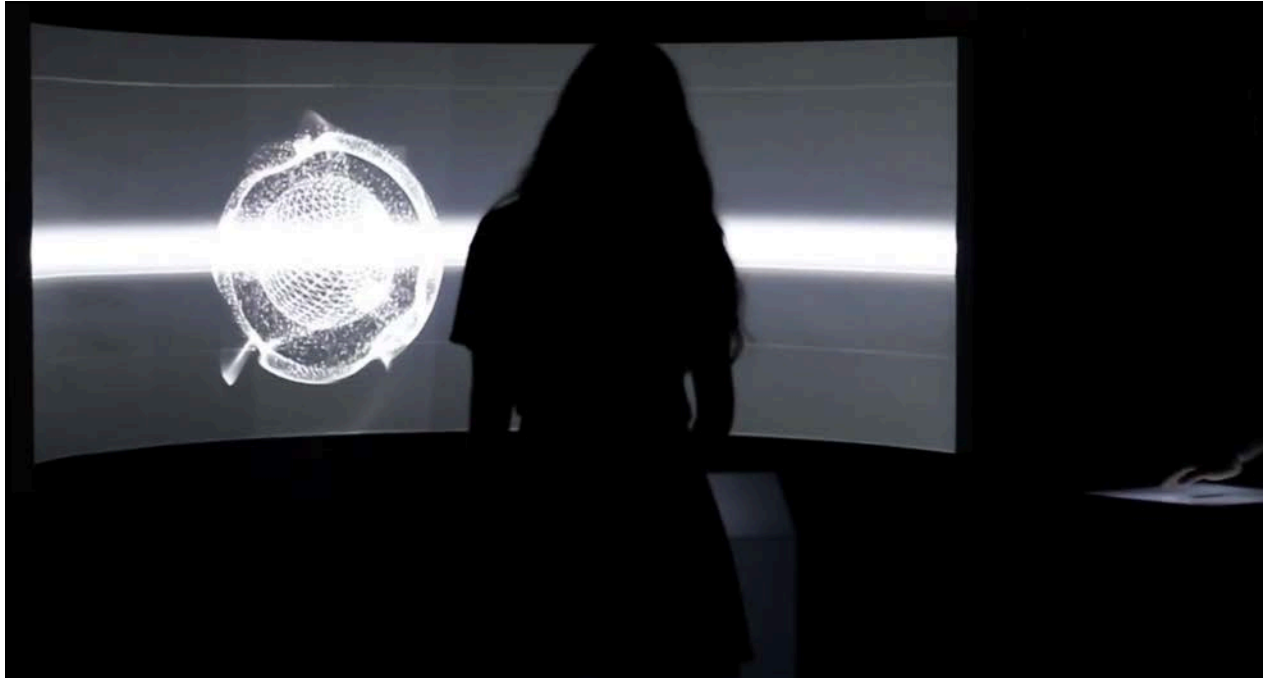
*Figura 33 - Imagen de la interacción en la sala de exposición con Leap Motion*

Otro comportamiento observado durante la exposición trata sobre la interacción con los dispositivos como si de botones se tratara. A causa de la estética de los sensores de proximidad, confundían a algunos usuarios y éstos trataban de tocarlos todo el rato. A nivel conceptual, esto no era ningún problema ya que el resultado era el mismo que si el usuario acercaba la mano. Lo mismo pasó con el Leap. Algunos usuarios trataban de tapar el dispositivo, pero el resultado también era el mismo y con el tiempo y aprendizaje de la pieza mejoró su interacción.

El único problema que esto podía causar era ensuciar el cristal que cubría el Leap Motion dificultando así su detección óptima. Para ello, se dispuso de material de limpieza rápido para de vez en cuando limpiar el material, aunque no fuera crítico.

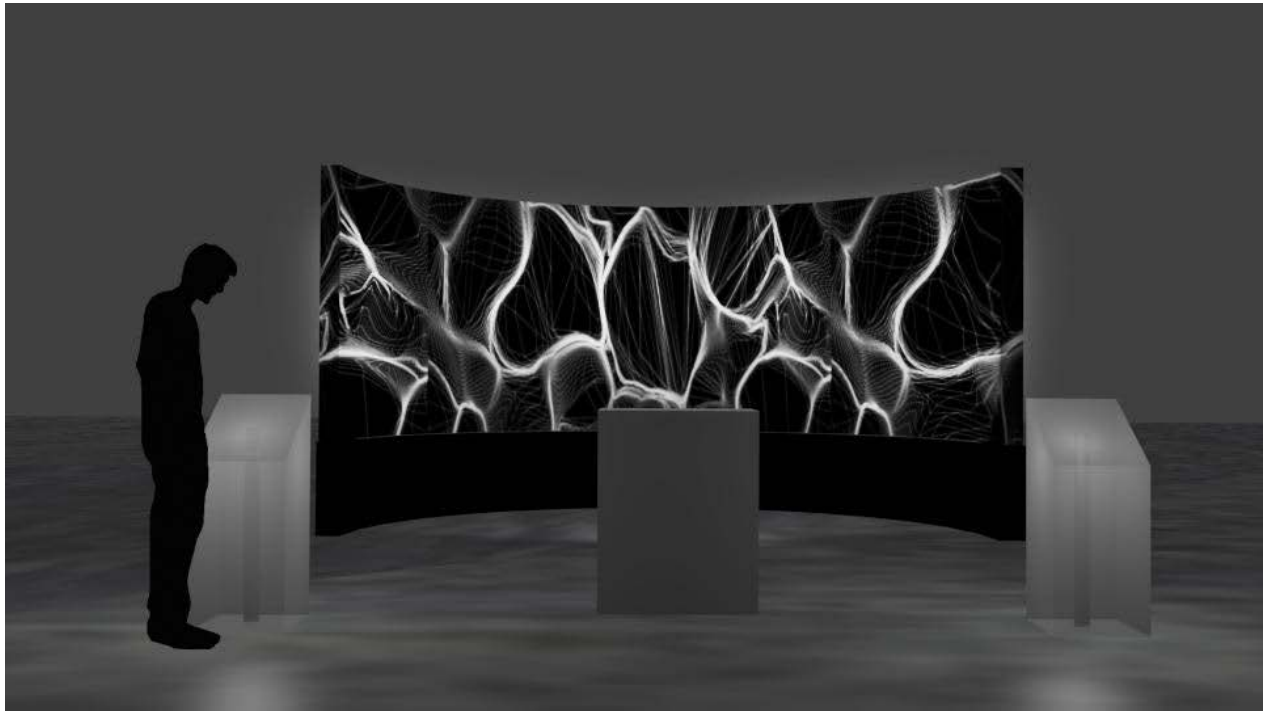
El resultado de la exposición fue el esperado y por parte de los usuarios muy satisfactorio. Se colocó un cartel en la puerta de la sala para poder explicar el concepto y facilitar al usuario la comprensión de lo que se mostraba dentro. Algunos interesados preguntaban más cuando salían de la sala ya que estaban interesados en la pieza.

El resultado general con los usuarios se considera satisfactorio a causa de su gran variedad e igualmente su buen funcionamiento y comprensión con la interacción de los dispositivos. Su experiencia era satisfactoria.

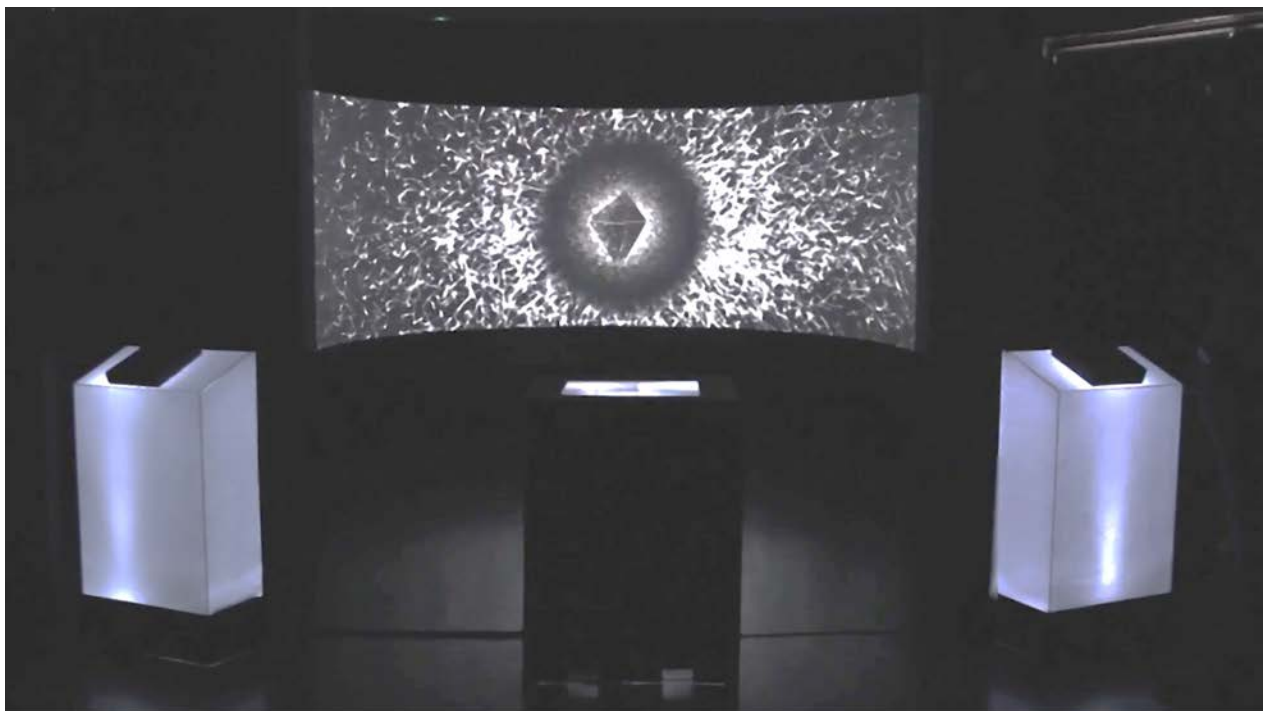


*Figura 34 - Sala de exposición con interacción en el dispositivo central Leap Motion*

Falta añadir que el resultado final obtenido después de realizar el montaje de la exposición, es muy semejante al que se planteó una vez se diseñó el material que se requería para la pieza. En la figura 35 y la figura 36 se puede comparar las expectativas que se tenían sobre la pieza, y cómo ha quedado en realidad. Comparando el diseño 3D hecho con antelación a la exposición, y la imagen de la realidad de cómo quedó, se puede decir que el resultado es muy satisfactorio.



*Figura 35 - Imagen demostrativa con la previsión del resultado del montaje en la sala de exposición*



*Figura 36 - Resultado obtenido real en la sala de exposición*

## 7. CONCLUSIONES y LÍNEAS DE FUTURO

Cuando se empezó a trabajar sobre el concepto sobre el que se quería hablar, el equipo se dio cuenta de la magnitud de conceptos de los que podían hablarse, y sobre qué se quería transmitir. Después de investigar durante meses, hablar de ciborgs, robots, tecnología y la sociedad, se decidió que hacer una comparativa entre los humanos y las máquinas era una buena idea. Transmitir al usuario que mientras no pierda el control de lo que está haciendo con la máquina, siempre estará por encima. Pero por consecuencia, si no tiene este control, la máquina puede convertirse en un ser oscuro y malo, y volverse en su contra.

La manera de representar este concepto, se planteó de muchas maneras. Lo que se quería principalmente era 'estresar' al usuario de tal manera que quisiera pararlo, y darle ese poder. Controlarlo. Así, entra en la conciencia del humano el querer controlar y poder hacerlo, ya que en un hipotético caso de distopía, podría no poder hacerlo.

Querer transmitir un concepto a diferentes personas de manera artística es algo complejo. Para ello, se añadió la parte de interactividad. Además, se quiso que esta pieza fuera colaborativa. Esto significa, que para tener un control total sobre la pieza se necesita más de una persona dentro de la sala interactuando con los sensores: en concreto, 3.

Para realizar el estudio sobre la experiencia del usuario y el aprendizaje de ellos para interactuar con los diferentes dispositivos que se querían utilizar, fue todo un reto. Tenía que ser una interacción simple e intuitiva.

Trabajar con Arduino y Leap Motion se ha convertido en una tarea asequible gracias a la ayuda de los profesores y, además, por toda la cantidad de información y APIs que se encuentran en internet.

Tras la investigación sobre el alcance de los dispositivos que se escogieron para la pieza, se decidió que eran los mejores y los más asequibles a los que podíamos acceder.

Nunca antes se había trabajado directamente con Leap Motion (sí con dispositivos similares). Era un reto académico y personal poder desarrollar con este dispositivo. Su diseño facilita el poder adaptarlo y colocarlo en cualquier lugar y que quede bien. Su funcionalidad es correcta incluso a poca luz, y el funcionamiento durante la exposición de la pieza fue el esperado.

Trabajar en un grupo con perfiles tan diferentes, han dado resultado a un proyecto del cual se está muy orgulloso. La diversidad de opiniones, y de experiencias, ha dado fruto a un resultado disruptivo y fuera de lo común.

La coordinación entre 5 personas también es complicada, y cabe decir que se ha coordinado muy bien el equipo, así como la división de tareas.

Uno de los retos que motivaban más para este trabajo, además de aprender Leap, fue trabajar con Video-mapping. Era uno de aquellos conceptos que siempre hubiera gustado hacer en la carrera y gracias a este Máster se ha podido aprender y desarrollar un proyecto con esta tecnología y además, interactivo.

Ha sido un trabajo exitoso, satisfactorio y gratificante.



Basándose en el proyecto explicado en este trabajo, se procede a presentar una propuesta del mismo, pero a una escala mayor. Este proyecto excluye la pantalla utilizada de 180° en este proyecto, y se modifica por un domo de 360°. Para ello, se utilizará un Domo y es por este motivo, que de ahora en adelante se presentará como el proyecto Domo.

Se mantiene el mismo concepto que en este trabajo, que es basar la narrativa en la interacción más que en una narrativa como lenguaje, es decir, se creará una narrativa interactiva. Además, se plantea que la obra será colaborativa y abierta. Con abierta, se hace referencia al concepto de Umberto Eco, que menciona varias categorías o tipos de obras abiertas, diferenciándose en el nivel de apertura.

En Dystopia, se utiliza este concepto para permitir que el usuario forme parte de la instalación como creador y no sólo como usuario, esto para demostrar a través de su propia experiencia, el control y descontrol de la máquina con la intervención del humano. Cada usuario experimentará de forma distinta la pieza, y podrá tomar la decisión de controlar el ambiente. De la misma manera que en el prototipo presentado, pero a una escala mucho mayor e inmersiva.

Dystopia será espacio inmersivo, compuesto por un domo con proyección en 360 grados y un ambiente sonoro.

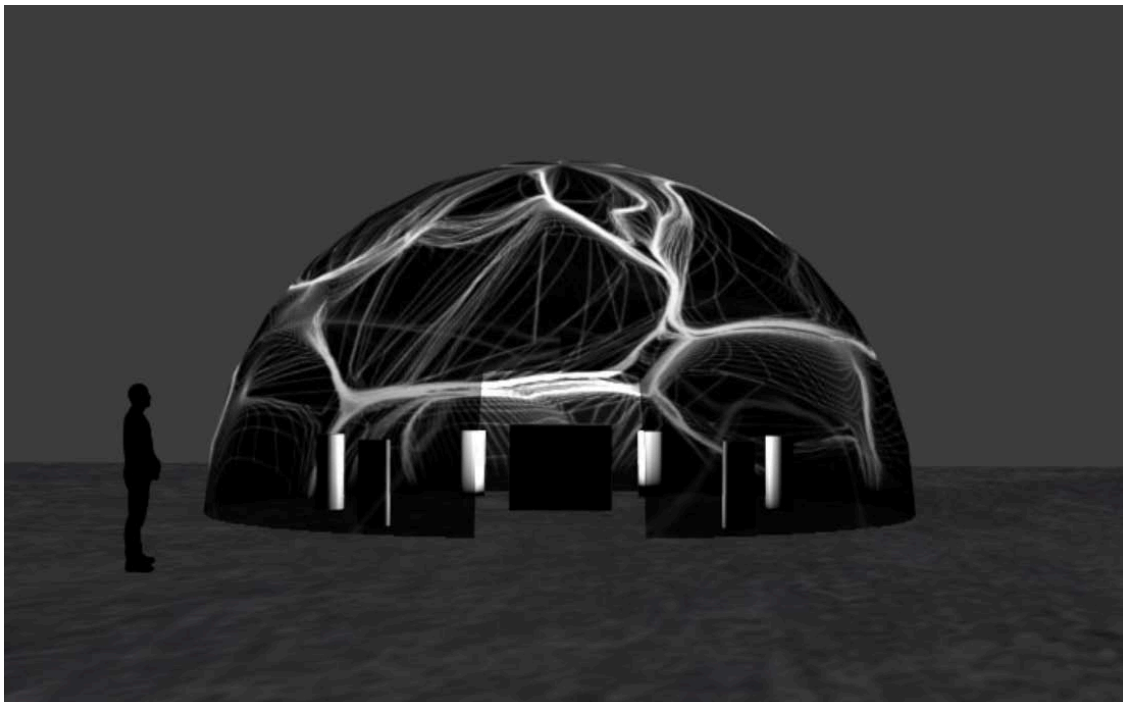
Primero, el usuario se encuentra en una fase de exploración, la interacción es pasiva e inconsciente a través de sensores de presencia que alterarán el ambiente sutilmente. Esta parte del proyecto se quiso añadir para esta primera exposición realizada, pero al no llegar a tiempo se suprimió y se dejó su planteamiento para ser aplicado en un futuro. De manera inmediata después de entrar en la sala, el usuario será atraído por dos mecanismos de interacción visibles:

- En el centro una interacción de aprendizaje que a través de interfaces gestuales “enseñan” al usuario a interactuar, esto para proponer que el aprendizaje es la base del control.
- El segundo mecanismo, propone una interacción consciente de acción, manipulación de objetos lumínicos de acuerdo a la proximidad que el usuario tenga con ellos.

De esta manera se pretende hacer consciente al usuario de su capacidad de control, esperando que sus siguientes interacciones, ya conscientes, sean más controladas y esto sea reflejado tanto en la esfera como en el ambiente sonoro y lumínico del espacio.

El ecosistema geodésico representa a Dystopia, un espacio en el que organismos virtuales poseen control sobre sí mismos cuando el humano no los altera. Podremos observar que sin la interacción del humano los organismos mencionados están en un total descontrol, generando un ambiente hostil y caótico. El usuario será un agente interno y externo dentro de Dystopia, ya que su interacción es omnisciente con respecto a la cúpula central, pero protagonista y diagógica con respecto a toda la instalación (espacio-habitación). Es precisamente sobre estos elementos que el usuario puede interferir para controlar, en distintos niveles, o no, al software.

Además, estos elementos serán controlados por membranas conectoras que interfieren o filtran la interacción entre Dystopia y el usuario. En la descripción anterior de la pieza, se hacen presentes las soluciones que desembocaron la investigación y será a partir de esta descripción que la interacción a nivel técnica se podrá desarrollar detalladamente, igualmente la estética se apegará a lo investigado y a la narración interactiva propuesta.

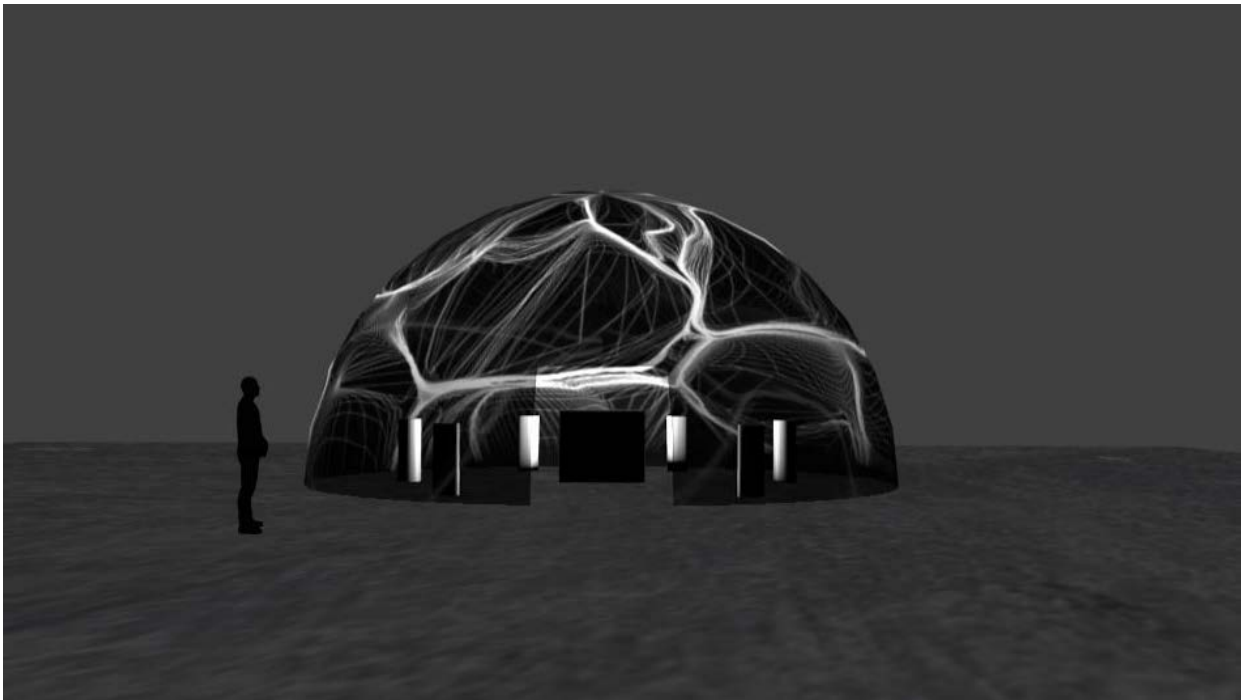


*Figura 37 - Diseño del Domo generado con la proyección*

Dystopia es una pieza audiovisual que conlleva la inclusión de distintas disciplinas trabajando en conjunto para lograr proporcionar la experiencia que se quiere transmitir al usuario. Estas disciplinas comprenden: Narrativa, Programación, Construcción, Adquisiciones, Diseños Visuales y Gráficos, Diseños Sonoros y Musicales, Animación y Proyección.

Una de las mayores ventajas de la pieza a nivel de producción es que se está diseñando desde sus inicios de forma que se pueda realizar su exhibición en cualquier espacio, en especial en espacios exteriores con el uso de generadores eléctricos y equipos de uso exterior, lo que nos permite, en caso de presentarse la oportunidad, realizar la exhibición en cualquier recinto interior o exterior que tenga la capacidad de contener el domo geodésico propuesto.

Y es por eso que, el proyecto presentado en este trabajo podría considerarse un prototipo sobre algo que puede ser mucho más grande.



*Figura 38 - Proyecto Dystopia (línea de futuro) Domo*

## 8. REFERENCIAS

- [1] Ajuntament de Barcelona, «Centre Cívic Sant Agustí,» [En línea]. Available: <http://www.bcn.cat/centrecivicsantagusti/tallers.html>.
- [2] Convent Sant Agustí, «Recorreguts Sonors,» [En línea]. Available: [conventagusti.com/blog/category/recorreguts/](http://conventagusti.com/blog/category/recorreguts/) .
- [3] J. Cameron, Dirección, *Terminator*. [Película]. 1984.
- [4] L. Olmo, «¿Qué diferencia hay entre Deep Learning, Inteligencia Artificial y Machine Learning?,» December 2016. [En línea]. Available: <http://www.ticbeat.com/tecnologias/que-diferencia-hay-entre-deep-learning-inteligencia-artificial-y-machine-learning/>.
- [5] R. Arrabales, «Deep Learning: qué es y por qué va a ser una tecnología clave en el futuro de la inteligencia artificial,» 2016. [En línea].
- [6] R. Brunet, «¿Qué es el deep learning?,» [En línea]. Available: <https://www.muyinteresante.es/tecnologia/preguntas-respuestas/que-es-el-deep-learning-701487764900>.
- [7] Consultec, «Deep Learning: una rama clave en el futuro de la Inteligencia Artificial,» February 2017. [En línea]. Available: <http://consultec-ti.com/blog/deep-learning-una-rama-clave-en-el-futuro-de-la-inteligencia-artificial/>.
- [8] Telefónica Open Future, «Deep Learning, una nueva técnica de Inteligencia Artificial basada en redes neuronales,» DIC 2016. [En línea]. Available: <https://www.openfuture.org/es/new/deep-learning-una-nueva-tecnica-de-inteligenc>.
- [9] J. F. Cía, «Cómo el machine learning y el deep learning marcan el camino de la verdadera Inteligencia Artificial,» 2015. [En línea]. Available: <https://bbvaopen4u.com/es/actualidad/como-el-machine-learning-y-el-deep-learning-marcan-el-camino-de-la-verdadera-inteligencia>.
- [10] C. G. (Indra), «La apuesta de los gigantes tecnológicos por la Inteligencia Artificial,» [En línea]. Available: <http://www.indracompany.com/es/blogneo/apuesta-gigantes-tecnologicos-inteligencia-artificial>.
- [11] Solo Márketing, «Deep Learning, la innovadora rama de la inteligencia artificial,» March 2017. [En línea]. Available:

<https://www.solomarketing.es/deep-learning-la-innovadora-rama-de-la-inteligencia-artificial/>.

- [12] G. Zarkadakis, *In Our Own Image: Will artificial intelligence save or destroy us?*, 2015.
- [13] J. H. Gibbens, Artist, *Dystopia*. [Art]. 2011.
- [14] M. Rosa, «Literalmente - Los viajes de Gulliver,» 2015. [En línea]. Available: <http://tertulialiteral.blogspot.com.es/p/los-viajes-de-gulliver-losviajes-de.html>.
- [15] Y. Zamiatin, «Nosotros,» 1921.
- [16] R. Ray, «Elementos de una Distopía,» [En línea]. Available: <http://www.storyboardthat.com/es/articles/e/elementos-dist%C3%B3picos>.
- [17] D. Walsh, «Dystopia and Modern Technology,» 2014 September 2. [En línea]. Available: <https://hubpages.com/technology/Technological-Dystopia-10-Reasons-to-Fear-Technology>.
- [18] A. Proyas, Dirección, *I, Robot*. [Película]. 2004.
- [19] G. Orwell, «1984,» 1949.
- [20] F. Lang, Dirección, *Metrópolis*. [Película]. Germany.1927.
- [21] M. Kerbow. [En línea]. Available: <http://michaelkerbow.com/>.
- [22] A. Stanton, Dirección, *WALL-E*. [Película]. EEUU: Walt Disney Pictures / Pixar Animation Studios, 2008.
- [23] X-TECH, «Augmented Reality And Kinect Create Unique Art Experience At Cleveland Museum,» [En línea]. Available: <http://x-tech.am/augmented-reality-and-kinect-create-unique-art-experience-at-cleveland-museum/>.
- [24] Leap Motion Developer, «Any Surface Can Be a Musical Instrument,» 17 March 2014. [En línea]. Available: <http://blog.leapmotion.com/any-surface-can-be-a-musical-instrument/>.
- [25] A. Schlegel, Artist, *Polar Reversal - Interactive Lights and Objects*. [Art]. 2016.
- [26] IRCAM, «Sound Music Movement - Interaction,» [En línea]. Available: <http://ismm.ircam.fr/category/project/>.
- [27] LM, «Aplicaciones y Utilidades del Leap Motion,» 2013. [En línea]. Available: [http://leapmotion-info.blogspot.com.es/2013/05/aplicaciones-y-utilidades-del-leap\\_28.html](http://leapmotion-info.blogspot.com.es/2013/05/aplicaciones-y-utilidades-del-leap_28.html).
- [28] L. Motion, «The future is in reach,» [En línea]. Available: <https://www.leapmotion.com/>.

- [29] A. Almà, «inLab talent & technology - Interacción 3D con Leap Motion,» 2014. [En línea]. Available: <https://inlab.fib.upc.edu/es/blog/interaccion-3d-con-leap-motion>.
- [30] C. Press, Escritor, *Academia de Networking de CISCO Systems*. [Performance].
- [31] Sojamo, «OSCP5,» 2011. [En línea]. Available: <https://github.com/sojamo/oscp5>.
- [32] J. Swift, «Los viajes de Gulliver,» 1726.
- [33] M. Shelley, «Frankenstein,» 1818.
- [34] A. Huxley, «Un mundo feliz,» 1932.
- [35] R. Bradbury, «Fahrenheit 451,» 1953.
- [36] D. Morán, «Literatura Distópica: Cuando el futuro es una pesadilla,» August 2014. [En línea]. Available: <http://www.abc.es/cultura/libros/20140805/abci-literatura-distopica-201408041809.html>.
- [37] S. W. N. Klous, *We are Big Data: The future of the information society*, Holanda: Atlantis press, 2016.
- [38] UNESCO, [En línea]. Available: <http://en.unesco.org/>.
- [39] C. B. Frucor, «The V Motion Project,» [En línea]. Available: <http://www.assemblyltd.com/work/v-motion-project>.
- [40] Arduino Advanced Projects, «Mesa con LEDs,» 2012. [En línea]. Available: [http://descubrearduino.com/8-proyectos-arduino-para-nivel-avanzado/#Una\\_mesa\\_con\\_sensor\\_de\\_proximidad](http://descubrearduino.com/8-proyectos-arduino-para-nivel-avanzado/#Una_mesa_con_sensor_de_proximidad).
- [41] C.-C. C. a. J. K. A. Lu Xia, «Human detectoin using depth information by Kinect,» [En línea]. Available: <https://nattee.net/sites/default/files/Human%20Detection%20Using%20Depth%20Information%20by%20Kinect.pdf>.
- [42] Laura, «La Guía Matemática,» 1 March 2013. [En línea]. Available: <https://matematica.laguia2000.com/general/teselaciones>.
- [43] SOJAMO, Artist, *Shift - Installation*. [Art]. 2011.
- [44] A. Schlegel, Artist, *Signal Body Space*. [Art]. Multi-sensory environment, interactive lights, sound, 2017.
- [45] A. Schlegel, Artist, *Movement of Things - Electronics, motion sensing, data renderings*. [Art]. 2016.
- [46] A. Schlegel, Artist, *Foris - Sound installation*. [Art]. 2011.

- [47] E.-I. H2020, Escritor, *Rapid-MIX*. [Performance]. Realtime Adaptive Prototyping for Industrial Design of Multimodal Interactive eXpressive technology.
- [48] ANR (Agencia Nacional de Recerca), «WAVE project,» 2013/2015.
- [49] ARSGAMES, «Leap Motion y Realidad Virtual,» April 2016. [En línea]. Available: <http://arsgames.net/blog/2016/04/01/leap-motion-y-realidad-virtual-en-el-cole/>.
- [50] I. López, «El primer rascacielos, realidad virtual en el Espacio Fundación Telefónica,» *Experimenta Magazine*, 2017.
- [51] J. García, «Nuevas formas de interacción gráfica con videojuegos: utilización de Leap Motion y Oculus RIFT,» 2015. [En línea]. Available: [https://e-archivo.uc3m.es/bitstream/handle/10016/23020/TFG\\_Jennifer\\_Garcia\\_De\\_La\\_Calle\\_2015.pdf?sequence=1](https://e-archivo.uc3m.es/bitstream/handle/10016/23020/TFG_Jennifer_Garcia_De_La_Calle_2015.pdf?sequence=1).
- [52] K. Laazizi Ruiz, «Desarrollo de un prototipo usando como dispositivo de interacción Leap Motion,» Madrid, 2014.
- [53] Showleap Technologies, «Un proyecto pensado para cambiar el mundo,» 2015. [En línea]. Available: [http://oa.upm.es/30975/1/PFC\\_KARIM\\_LAAZIZI\\_RUIZ.pdf](http://oa.upm.es/30975/1/PFC_KARIM_LAAZIZI_RUIZ.pdf).

## Tablas y Figuras

Figura 1 - The Course of Empire .....	19
Figura 2 - Kinect .....	29
Figura 3 - Sensor de proximidad ultrasonido HC - SR04 .....	30
Figura 4 - Programación del Leap Motion con Processing .....	32
Figura 5 - Panel de configuración del Leap Motion .....	33
Figura 6 - Partes del dispositivo Leap Motion .....	34
Figura 7 - Detección de las manos desde Leap Motion .....	35
Figura 8 - Luz infrarroja de Leap Motion .....	35
Figura 9 - Microcontrolador de Leap Motion .....	36
Figura 10 - Zona de cobertura de Leap Motion .....	37
Figura 11 - Zona de intersección de Leap Motion .....	37
Figura 12 - Gestos detectados con Leap Motion .....	38
Figura 13 - Datagrama UDP dentro de un datagrama IP .....	40
Figura 14 - Detección de manos con Leap Motion .....	44
Figura 15 - Sistema de coordenadas de Leap Motion .....	44
Figura 16 - Punto de detección de la clase Hand con Leap Motion .....	45
Figura 17 - Dedos detectados por Leap Motion .....	46
Figura 18 - Movimiento de partículas con el ratón en Leap Motion .....	49
Figura 19 - Esquema gráfico de comunicación y configuración de la programación .....	50
Figura 20 - Representación de la esfera central con la variación del número de triángulos teselados .....	57
Figura 21 - Esquema de interacciones con Leap Motion y sus respuestas .....	59
Figura 22 - Interacción en el eje de coordenadas OX .....	61
Figura 23 - Interacción en el eje de coordenadas OY .....	61
Figura 24 - Desarrollo del tracking de la mano para el seguimiento de partículas con Processing y Leap Motion .....	62
Figura 25 - Efecto de vídeo cargado cuando se recibe un Swipe hacia la izquierda .....	63
Figura 26 - Captura de pantalla del programa Resolume .....	63
Figura 27 - Representación de la interacción con Leap Motion .....	67
Figura 28 - Render con la distribución de la sala .....	70
Figura 29 - Estructuras para ocultar los sensores .....	71
Figura 30 - Render con vista cenital de la sala de exposición .....	72
Figura 31 - Render con vista semi-lateral de la sala de exposición .....	72



Figura 32 - Fotografía de la exposición donde se observa la interacción con Leap Motion.....	73
Figura 33 - Imagen de la interacción en la sala de exposición con Leap Motion.....	74
Figura 34 - Sala de exposición con interacción en el dispositivo central Leap Motion .....	75
Figura 35 - Imagen demostrativa con la previsión del resultado del montaje en la sala de exposición .....	76
Figura 36 - Resultado obtenido real en la sala de exposición.....	76
Figura 37 - Diseño del Domo generado con la proyección .....	80
Figura 38 - Proyecto Dystopia (línea de futuro) Domo .....	81
Tabla 1 - Librerías para Processing y su uso .....	42
Tabla 2 - Unidades API Leap Motion .....	44

## **9. ANEXO**

<b>9.1 INTRODUCCIÓN</b> .....	<b>2</b>
<b>9.2 DESARROLLO</b> .....	<b>3</b>
9.2.1 Desarrollo técnico .....	3
9.2.2 Desarrollo gráfico .....	10
<b>9.3 SOFTWARE</b> .....	<b>28</b>
<b>9.4 CARACTERÍSTICAS TÉCNICAS PARA EL PROYECTO DOME</b> .....	<b>29</b>

## 9.1 INTRODUCCIÓN

En este documento se encuentra el desarrollo técnico del proyecto a nivel de programación. Se ha dividido este desarrollo en dos fases como es explicado en la memoria del trabajo.

Se encuentra:

- **Desarrollo técnico:** Nombrado así por el procesado que realiza el script. Es el encargado de procesar y trabajar con los datos recibidos por los sensores del proyecto: un Leap Motion y dos sensores de proximidad que funcionan con Arduino. Toda esta información procesada será enviada al otro archivo mediante una conexión UDP.

El ordenador encargado de ejecutar el script con el desarrollo técnico, realizado con Processing, tendrá todos los dispositivos directamente conectados por cable USB. Por consecuencia, para este proyecto, el PC deberá tener al menos 3 puertos USB.

El uso de las librerías para este archivo es válido tanto para MAC como para Windows. Se necesitará instalar los drivers de Leap Motion y las librerías correspondientes y explicadas en la memoria del trabajo.

- **Desarrollo gráfico:** Como bien indica su nombre, este script será el encargado de reproducir toda la parte gráfica del proyecto con los datos recibidos y procesados por el desarrollo técnico. El único sitio donde tiene que estar conectado este ordenador es a la misma red que el ordenador que tenga el script técnico. Ésta será creada para obtener las IPs correspondientes y poder realizar la conexión UDP.

Una vez este ordenador recibe los datos de los sensores, transforma esta información para alterar los valores visuales y auditivos de la pieza.

En el código se encuentran las funciones comentadas para comprender el funcionamiento de cada una. Se ha dejado un ejemplo de implementación de gestos para poder en un futuro ampliar las capacidades de alteración y poder añadir más efectos repercusivos a la obra.

## 9.2 Desarrollo

### 9.2.1 Desarrollo técnico

```
//-----//
###DESCRIPCION###
//-----//
//Recepcion de datos de sensores
//Envio de datos a processing-processing ordenador remoto por UDP
//-----//

//-----//
###LIBRERIAS###
//-----//
import hypermedia.net.*; //UDP
import processing.serial.*; //Serial
import oscP5.*; //OSC
import netP5.*; //OSC
import de.voidplus.leapmotion.*; //Leap
//-----//

//-----//
###CONECTIVIDAD###
//-----//
//Puertos locales por los que se envian los datos
int PuertoLocal1 = 7100; //arduino1
int PuertoLocal2 = 7200; //arduino2
int PuertoLocal3 = 7300; //leapX
int PuertoLocal4 = 7400; //leapY
int PuertoLocal5 = 7500; //leapZ
int PuertoLocal6 = 7600; //HAND-DETECTED

//Puerto que recibe datos en PC de graficos
int puertoRemoto = 6100;
```

```
//Direccion ip de la computadora a la que le mandaremos mensajes
```

```
String ipRemota = "192.168.0.101";
```

```
//Conexiones UDP
```

```
UDP udp1, udp2, udp3, udp4, udp5, udp6;
```

```
//-----//
```

```
//-----//
```

```
### ARDUINO ###
```

```
//-----//
```

```
//Conexiones serie para recibir datos de Arduino
```

```
Serial ArduinoPort1;
```

```
Serial ArduinoPort2;
```

```
//Variables Arduino
```

```
int valueArduino1;
```

```
int valueArduino2;
```

```
//-----//
```

```
//-----//
```

```
### LEAP MOTION ###
```

```
//-----//
```

```
LeapMotion leap;
```

```
float posX;
```

```
float posY;
```

```
float posZ;
```

```
//int swipe = 0;
```

```
//int numF;
```

```
//-----//
```

```
void setup() {
```

```
//-----//
```

```

    /// ARDUINO ///
    //-----//

    String port1 = Serial.list()[0];
    String port2 = Serial.list()[1];

    ArduinoPort1 = new Serial(this, port1, 9600);
    ArduinoPort2 = new Serial(this, port2, 9600);

    valueArduino1 = 0;
    valueArduino2 = 0;
    //-----//
    //-----//
    /// LEAP MOTION ///
    //-----//

    leap = new LeapMotion(this).allowGestures();
    leap = new LeapMotion(this);

    posX = 0;
    posY = 0;
    posZ = 0;
    //-----//
    /// CONECTIVIDAD ///
    //-----//
    // Se crean conexiones UDP para enviar a pc de graficos
    udp1 = new UDP( this, PuertoLocal1 );
    udp2 = new UDP( this, PuertoLocal2 );
    udp3 = new UDP( this, PuertoLocal3 );
    udp4 = new UDP( this, PuertoLocal4 );
    udp5 = new UDP( this, PuertoLocal5 );
    udp6 = new UDP( this, PuertoLocal6 );
    //-----//
}

void draw() {

```

```

//-----//
### PROCESAMIENTO DATOS ARDUINO ###
//-----/

//Recepcion datos de Arduino
if ( ArduinoPort1.available() > 0) { // If data is available,
    valueArduino1 = ArduinoPort1.read();    // read it and store it in val
}
if ( ArduinoPort2.available() > 0) { // If data is available,
    valueArduino2 = ArduinoPort2.read();    // read it and store it in val
}
//se pasa a string para enviar los datos
String message1 = str(valueArduino1);
String message2 = str(valueArduino2);
//-----//
### PROCESAMIENTO DATOS LEAP MOTION ###
//-----//
//pasamos una variable a grafica para indicarle si hay alguna mano detectada o no
int m6 = 0;

if (leap.hasHands()) {
    m6 = 1;
}else{
    m6 = 0;
}
for (Hand hand : leap.getHands ()) {

    PVector handPosition = hand.getPosition();

    posX = handPosition.x;
    posY = handPosition.y;
    posZ = handPosition.z;
}

int m3 = int(posX);
int m4 = int(posY);
int m5 = int(posZ);

```

```

String message3 = str(m3);
String message4 = str(m4);
String message5 = str(m5);
String message6 = str(m6);

//-----//
### ENVIO DE MENSAJES A ORDENADOR DE GRAFICOS ###
//-----//

udp1.send( message1, ipRemota, puertoRemoto ); //arduino1
udp2.send( message2, ipRemota, puertoRemoto ); //arduino2
udp3.send( message3, ipRemota, puertoRemoto ); //leapX
udp4.send( message4, ipRemota, puertoRemoto ); //leapY
udp5.send( message5, ipRemota, puertoRemoto ); //leapZ
udp6.send( message6, ipRemota, puertoRemoto ); //hasHands

//-----//
### DEBUG ###
//-----//

/*println("Arduino1: " + valueArduino1);
println("Arduino2: " + valueArduino2);
println("POS X = " + posX);
println("POS Y = " + posY);
println("num dedos = " + numF);*/
//-----//
//println("HAS HAND = " + m6);

}

// Esta parte del código aparece comentada ya que al final no se utilizan los gestos para este pr
oyecto
/*void leapOnSwipeGesture(SwipeGesture g, int state){

String message6;

```



```

PVector direction    = g.getDirection();
int m6 = 0;
message6 = str(m6);
switch(state){
  case 1: // Start
    break;
  case 2: // Update
    break;
  case 3: // Stop
    //println("Posicio inicial" + positionStart);
    //println("Posicio actual" + position);
    //println("DIRECCIÓN: " + direction.x);
    if (direction.x >= 0){
      //Moviment DRETA
      //println("DRETA");
      // println("SwipeGesture: " + id);
      m6 = 1;
      message6 = str(m6);
    }else{
      //println("ESQUERRA");
      //println("SwipeGesture: " + id);
      m6 = 2;
      message6 = str(m6);
    }
    break;
}
//udp6.send( message6, ipRemota, puertoRemoto ); //leap-swipe

}

```

```

void leapOnCircleGesture(CircleGesture g, int state){

float durationSeconds = g.getDurationInSeconds();
int direction    = g.getDirection();

```

```

String message7;
int m7 = 0;
message7 = str(m7);

switch(state){
  case 1: // Start
    break;
  case 2: // Update
    break;
  case 3: // Stop
    //println("CircleGesture: " + id);
    break;
}
switch(direction){
  case 0: // Anticlockwise/Left gesture
    //Anticlockwise (left)
    //println("ANTICLOCKWISE (LEFT) ");
    //println("Circle Gesture Direction: " + direction);
    //println("Duration" + durationSeconds*100); //microseconds
    m7 = 2;
    message7 = str(m7);
    break;
  case 1: // Clockwise/Right gesture
    //Clockwise (right)
    //println("CLOCKWISE (RIGHT) ");
    //println("Circle Gesture Direction: " + direction);
    //println("Duration" + durationSeconds*100); //microseconds
    m7 = 1;
    message7 = str(m7);
    break;
}
//udp7.send( message7, ipRemota, puertoRemoto ); //leap-swipe
m7 = 0;
}*/

```

## 9.2.2 Desarrollo gráfico

```
//-----//
###DESCRIPCION###
//-----//
//Recepcion de datos de ordenador remoto de sensores por UDP
//-----//

//-----//
###LIBRERIAS###
//-----//
import hypermedia.net.*; //UDP
import oscP5.*; //OSC
import netP5.*; //OSC
//import codeanticode.syphon.*; //Syphon
import spout.*; //Spout (como syphon para windows)
import ddf.minim.*; //audio

//-----//
###CONECTIVIDAD###
//-----//
//Para conexion con Syphon
//SyphonServer server;

Spout spout;
//para controlar las notas del audio
int nota = 0;
int sonidoLeap= 0;
int sonidoCentral= 0;

//Para conexion OSC con Resolume
//Direccion ip de la computadora a la que le mandaremos mensajes (donde esta el Resolume)
String ipAddressToSendTo = "localhost";
OscP5 oscP5;
NetAddress myRemoteLocation;
```

```

OscBundle myBundle;
OscMessage myMessage;

//PuertoLocal que recibe de Resolume
int portToListenToResolume = 7001;
//Puerto que envía a Resolume
int portToSendToResolume = 7000;
//ESTOS PUERTOS TIENEN QUE ESTAR SETEADOS EN RESOLUME

//Se definen objetos UDP
UDP udp;

//Puerto local por el que se reciben datos de sensores
int puertoLocal = 6100;

//Puertos remotos de pc de sensores
//Verificar que sean los mismos que en el otro codigo

int PuertoRemoto1 = 7100; //arduino
int PuertoRemoto2 = 7200; //arduino
int PuertoRemoto3 = 7300; //leap
int PuertoRemoto4 = 7400; //leap
int PuertoRemoto5 = 7500; //leap
int PuertoRemoto6 = 7600; //leap hand detected y/n -> 0/1

//-----//

//-----//
###DATA###
//-----//

int val1 = 0; //arduino1
int val2 = 0; //arduino2
int val3 = 0; //leap (x)
int val4 = 0; //leap (y)
int val5 = 0; //leap (z)
int val6 = 0; //leap hand detected

```

//estos valores se definen para incluir los gestos. Aparecen comentados porque no les daremos uso en este proyecto

//int val7 = 0; //leap (swipe gesture 0=none;1=right;2=left)

//int val8 = 0; //leap (circle gesture 0=none;1=right;2=left)

int MAX\_DISTANCE\_ARDUINO = 50; //CAMBIAR TAMBIEN EN CODIGO ARDUINO

int MAX\_DISTANCE\_LEAPX = 85;

int MAX\_DISTANCE\_LEAPY = 90; //TENER EN CUENTA QUE LA Y SE MIDE AL REVES

int MAX\_DISTANCE\_LEAPZ = 70;

//Variables para la parte grafica

int cuantos = 30000;

Textura[] tex ;

float rx = 0;

float ry = 0;

//input audios

Minim minim;

AudioPlayer player1;

AudioPlayer player2;

AudioPlayer player3;

AudioPlayer player4;

AudioPlayer player5;

AudioPlayer player6;

AudioPlayer player7;

AudioPlayer player8;

AudioPlayer player9;

AudioPlayer player10;

AudioInput input;

```

int movimientoX = 0;
int movimientoY = 0;
int movimientoZ = 0;

//-----//
void settings() {
    size(1920, 530, P3D);
    //PJOGL.profile=1;
}

void setup() {

    //-----//
    ///CONECTIVIDAD///
    //-----//
    //Se crea conexion OSC para enviar ordenes a resolume
    oscP5 = new OscP5(this, portToListenToResolume);
    myRemoteLocation = new NetAddress(ipAddressToSendTo, portToSendToResolume);
    myBundle = new OscBundle();
    myMessage = new OscMessage("/");

    // Se crea conexion UDP para recibir datos de sensores
    udp = new UDP( this, puertoLocal );
    udp.listen( true );

    //syphon o spout (MAC o WINDOWS)
    //server = new SyphonServer(this, "Processing Syphon");
    spout = new Spout(this);
    spout.createSender("SpoutProc");

    //-----//

    //-----//
    ///GRAFICA///
    //-----//

```

```

tex = new Textura[cuantos];
for(int i=0; i< tex.length; i++) {
tex[i] = new Textura();
}

// -----//
// SONIDO //
//-----//

minim = new Minim(this);
player1 = minim.loadFile("esfera1.wav");
player2 = minim.loadFile("esfera2.wav");
player3 = minim.loadFile("esfera3.wav");
player4 = minim.loadFile("esfera4.wav");
player5 = minim.loadFile("esfera5.wav");
player6 = minim.loadFile("esfera6.wav");
player7 = minim.loadFile("esfera7.wav");
player8 = minim.loadFile("esfera8.wav");
player9 = minim.loadFile("esfera9.wav");
player10 = minim.loadFile("esfera10.wav");
input = minim.getLineIn();

player1.setGain(-5.0f);
player2.setGain(-5.0f);
player3.setGain(-5.0f);
player4.setGain(-5.0f);
player5.setGain(-5.0f);
player6.setGain(-5.0f);
player7.setGain(-5.0f);
player8.setGain(-5.0f);
player9.setGain(-5.0f);
player10.setGain(-5.0f);

}

```

```

//-----//
###RECEPCION DATOS ORDENADOR SENSORES###
//-----//
void receive( byte[] data, String ip, int port ) { // <-- extended handler

    String message = new String( data );

    //Se clasifican mensajes segun el puerto de envio
    if (port == PuertoRemoto1) {
        val1 = int(message);
    } else if (port == PuertoRemoto2) {
        val2 = int(message);
    } else if (port == PuertoRemoto3) {
        val3 = int(message);
    } else if (port == PuertoRemoto4) {
        val4 = int(message);
    } else if (port == PuertoRemoto5) {
        val5 = int(message);
    } else if (port == PuertoRemoto6) {
        val6 = int(message);
    } /*else if (port == PuertoRemoto7) {
        val7 = int(message);
    }*/

}
//-----//

void draw() {

    //-----//
    ### GENERACION DE GRAFICOS ###
    //-----//

    //se convierte a tamaño de pantalla
    movimientoX = (int)map(float(val3), 0.0, float(MAX_DISTANCE_LEAPX), 0.0, 1920);
    movimientoY = (int)map(float(val4), 0.0, float(MAX_DISTANCE_LEAPY), 0.0, 530);

```



```
movimientoZ = (int)map(float(val5), 0.0, float(MAX_DISTANCE_LEAPZ), 0.0, 530);
```

```
background(0);
```

```
//control movimiento esfera
```

```
float rx = (movimientoX-(width/2))*0.005;
```

```
float ry = (movimientoY-(height/2))*0.005;
```

```
rx = rx*0.9;
```

```
ry = ry*-0.9;
```

```
//posición esfera
```

```
translate(width/2, height/2);
```

```
//descontrol cuando no aparece manos
```

```
// boolean true cuando una de las manos está fuera de su rango
```

```
  // LEAP NO DETECTA NINGUNA MANO
```

```
  if (val6 == 0){
```

```
    val3 = 0;
```

```
    val4 = 0;
```

```
    val5 = 0;
```

```
    rotateY(random(-100,100));
```

```
    rotateX(random(-100,100));
```

```
    sphereDetail(1);
```

```
  // LEAP DETECTA ALGUNA MANO
```

```
  }else {
```

```
    rotateY(rx);
```

```
    rotateX(ry);
```

```
  }
```

```

//LEAP MOTION
int val3Mod = val3;
int val4Mod = val4;
int val5Mod = val5;

if (val3Mod < 0){
    val3Mod = 0;
}
if (val3Mod > MAX_DISTANCE_LEAPX){
    val3Mod = MAX_DISTANCE_LEAPX;
}

if (val4Mod < 0){
    val4Mod = 0;
}
if (val4Mod > MAX_DISTANCE_LEAPY){
    val4Mod = MAX_DISTANCE_LEAPY;
}
if (val5Mod < 0){
    val5Mod = 0;
}
if (val5Mod > MAX_DISTANCE_LEAPZ){
    val5Mod = MAX_DISTANCE_LEAPZ;
}

fill(0);
stroke(255);
sphere(100);

for (int i = 0; i < tex.length; i++) {
    tex[i].dibujar();
}

//-----//

```

```

// NOTAS MUSICALES

//println(movimientoY);

if (nota != 1 && movimientoY <= 106){
    nota = 1;
    player1.rewind();
    player1.play();
    sphereDetail(1);
}
if (movimientoY > 106 && movimientoY <= 159) {
    if (nota != 2){
        nota = 2;
        player2.rewind();
        player2.play();
        sphereDetail(2);
    }

} if (movimientoY > 159 && movimientoY <= 212) {
    if (nota != 3){
        nota = 3;
        player3.rewind();
        player3.play();
        sphereDetail(3);
    }

} if (movimientoY > 212 && movimientoY <= 265) {

    if (nota != 4){
        nota = 4;
        player4.rewind();
        player4.play();
        sphereDetail(4);
    }

} if (movimientoY > 265 && movimientoY <= 318) {

```

```
if (nota != 5){
    nota = 5;
    player5.rewind();
    player5.play();
    sphereDetail(5);
}

} if (movimientoY > 318 && movimientoY <= 371) {
    if (nota != 6){
        nota = 6;
        player6.rewind();
        player6.play();
        sphereDetail(6);
    }

} if (movimientoY > 371 && movimientoY <= 424) {
    if (nota != 7){
        nota = 7;
        player7.rewind();
        player7.play();
        sphereDetail(7);
    }

} if (movimientoY > 424 && movimientoY <= 477) {
    if (nota != 8){
        nota = 8;
        player8.rewind();
        player8.play();
        sphereDetail(10);
    }

} if (movimientoY > 477 && movimientoY <= 500 && nota != 9) {
    if (nota != 9){
        nota = 9;
        player9.rewind();
        player9.play();
    }
}
```

```

    sphereDetail(15);
}

} if (movimientoY > 500 && nota != 10) {
  if (nota != 10){
    nota = 10;
    player10.rewind();
    player10.play();
    sphereDetail(30);
  }
}
}

```

```
//envio pantalla a Resolume
```

```
//server.sendScreen();
spout.sendTexture();
```

```
//-----//
```

```
//-----//
```

```
### ENVIO DE ORDENES A RESOLUME CON OSC ###
```

```
//-----//
```

```
//Ejemplos de mensajes
```

```
// "/layer1/clip1/connect" --> selecciona clip1 de layer1
```

```
// "/layer1/clip1/video/effect1/param1/values" --> modifica parametro1 de efecto1 de video de clip1 en capa1
```

```
//-----//
```

```
///activeclip/video/effect1/param1/values
```

```
//Se escriben mensajes para Resolume
```

```
//ARDUINO 1
```

```
myMessage.setAddrPattern("/layer3/clip1/video/effect1/opacity/values");
```

```
    myMessage.add(map(float(MAX_DISTANCE_ARDUINO - val1), 0.0,
float(MAX_DISTANCE_ARDUINO - 5), 0.0, 1.0)); // el -5 es porque no llegaba a tope
```

```

myBundle.add(myMessage);
myMessage.clear();

myMessage.setAddrPattern("/layer4/clip1/audio/volume/values");
    myMessage.add(map(float(MAX_DISTANCE_ARDUINO - val1), 0.0,
float(MAX_DISTANCE_ARDUINO - 5), 0.7, 0.83)); //Sonido logaritmico
myBundle.add(myMessage);
myMessage.clear();

myMessage.setAddrPattern("/layer9/clip1/video/opacity/values");
    myMessage.add(map(float(MAX_DISTANCE_ARDUINO - val1), 0.0,
float(MAX_DISTANCE_ARDUINO - 5), 0.0, 1.0)); // el -5 es porque no llegaba a tope
myBundle.add(myMessage);
myMessage.clear();

myMessage.setAddrPattern("/layer5/clip1/video/effect2/param2/values");
    myMessage.add(map(float(MAX_DISTANCE_ARDUINO - val1), 0.0,
float(MAX_DISTANCE_ARDUINO - 5), 0.2, 1.0)); // el -5 es porque no llegaba a tope
myBundle.add(myMessage);
myMessage.clear();

//ARDUINO 2
myMessage.setAddrPattern("/layer6/clip1/audio/pitch/values");
myMessage.add(map(float(val2), 0.0 + 5.0, float(MAX_DISTANCE_ARDUINO), 0.1, 1.0));
myBundle.add(myMessage);
myMessage.clear();

myMessage.setAddrPattern("/layer5/clip1/video/effect1/param1/values");
myMessage.add(map(float(val2), 0.0 + 5.0, float(MAX_DISTANCE_ARDUINO), 0.0, 1.0));
myBundle.add(myMessage);
myMessage.clear();

myMessage.setAddrPattern("/layer3/clip1/video/effect2/param1/values");
myMessage.add(map(float(val2), 0.0 + 5.0, float(MAX_DISTANCE_ARDUINO), 0.3, 0.5));
myBundle.add(myMessage);
myMessage.clear();

```

```

myMessage.setAddrPattern("/layer8/clip1/audio/volume/values");
myMessage.add(map(float(val2), 0.0 + 5.0, float(MAX_DISTANCE_ARDUINO), 0.67, 0.85));
//Sonido logaritmico
myBundle.add(myMessage);
myMessage.clear();

//Se va sonido central

if (val2 < 20 && val1 < 20){
  if (sonidoCentral != 1){
    sonidoCentral = 1;
    myMessage.setAddrPattern("/layer1/clip2/connect");
    myMessage.add(1);
    myBundle.add(myMessage);
    myMessage.clear();
  }
  }else if (sonidoCentral != 0) {

    sonidoCentral = 0;

    myMessage.setAddrPattern("/layer1/clip1/connect");
    myMessage.add(1);
    myBundle.add(myMessage);
    myMessage.clear();

  }

/*
//La Y de Leap Motion va al reves, por eso hago max distance - val4

myMessage.setAddrPattern("/layer2/clip1/video/effect2/param1/values");
myMessage.add(map(float(MAX_DISTANCE_LEAPY - val4Mod), 0.0, float(MAX_DISTANCE_
LEAPY), 0.2, 0.8));
myBundle.add(myMessage);

```

```

myMessage.clear();*/

if (val6 == 0){
    if (sonidoLeap != 1){
        sonidoLeap = 1;
        myMessage.setAddrPattern("/layer7/clip1/connect");
        myMessage.add(1);
        myBundle.add(myMessage);
        myMessage.clear();
    }
    }else if (sonidoLeap != 0) {

        sonidoLeap = 0;

        myMessage.setAddrPattern("/layer7/clip2/connect");
        myMessage.add(1);
        myBundle.add(myMessage);
        myMessage.clear();

    }
/*myMessage.setAddrPattern("/layer7/clip1/audio/volume/values");
myMessage.add(map(float(val3Mod), 0.0 + 5.0, float(MAX_DISTANCE_LEAPX), 0.72, 1.0));
myBundle.add(myMessage);
myMessage.clear();*/

//Para cargar los clips con los gestos en caso de tenerlo activado

/*if (val8 == 1){
    myMessage.setAddrPattern("/layer5/clip1/connect");
    myMessage.add(1);
    myBundle.add(myMessage);
    myMessage.clear();
    val8 = 0;
}
if (val8 == 2){
    myMessage.setAddrPattern("/layer5/clip2/connect");

```



```

    myMessage.add(1);
    myBundle.add(myMessage);
    myMessage.clear();
    val8 = 0;
}
if (val7 != 0){
    if (val7 == 1){
        myMessage.setAddrPattern("/layer4/clip1/connect");
        myMessage.add(1);
        myBundle.add(myMessage);
        myMessage.clear();
        //println("HAS ENTRAT AL VAL == 1");
        val7 = 0;
    }else{

        if (val7 == 2){
            //println("HAS ENTRAT AL VAL == 2");
            myMessage.setAddrPattern("/layer4/clip2/connect");
            //myMessage.add(map(float(val2), 0.0, float(MAX_DISTANCE_ARDUINO), 0.0, 1.0));
            myMessage.add(1);
            myBundle.add(myMessage);
            myMessage.clear();
            val7 = 0;
        }
    }
}*/

//Envio del mensaje
oscP5.send(myBundle, myRemoteLocation);
myBundle.clear();
//-----//
//-----//
### DEBUG ###
//-----//
/*

```

```

println( "val1: "+val1 );
println( "val2: "+val2 );
println( "val3: "+val3 );
println( "val4: "+val4 );
println( "val5: "+val5 );
*/
//-----//
}

```

```

class Textura {
float z = random(-100, 100);
float phi = random(TWO_PI);
float theta = asin(z/100);
float largo = random(0.5, 1.6);

```

```

Textura() {
//contorno esfera
z = random(-100, 100);
phi = random (TWO_PI);
theta = asin(z/100);
//largo textura
largo = random (0.5, 1.6);
if (nota == 1){
largo = random (0.5, 1.6);
theta = asin(z/100);
}
if (nota == 2){
largo = random (0.5, 1.5);
theta = asin(z/200);
}
if (nota == 3){
largo = random (0.5, 1.4);
theta = asin(z/300);
}
if (nota == 4){
largo = random (0.5, 1.3);
theta = asin(z/400);
}
}

```

```

}
if (nota == 5){
    largo = random (0.5, 1.2);
    theta = asin(z/500);
}
if (nota == 6){
    largo = random (0.5, 1.1);
    theta = asin(z/700);
}
if (nota == 7){
    largo = random (0.5, 1);
    theta = asin(z/1000);
}
if (nota == 8){
    largo = random (0.5, 0.8);
    theta = asin(z/700);
}
if (nota == 9){
    largo = random (0.5, 0.6);
    theta = asin(z/700);
}
if (nota == 10){
    largo = random (0.5, 0.5);
    theta = asin(z/700);
}
}

void dibujar() {
    //ruido esfera

    float off = (noise(millis() * 0.0005, sin(phi))-0.5) * 0.3;
    float offb = (noise(millis() * 0.0007, sin(z) * 0.01)-0.5) * 0.3;

    float thetaff = theta+off;
    float phff = phi+offb;

```

```

float x = 100 * cos(theta) * cos(phi);
float y = 100 * cos(theta) * sin(phi);
float z = 100 * sin(theta);

//inicio de puntos de la esfera
float xo = 100 * cos(thetaff) * cos(phff);
float yo = 100 * cos(thetaff) * sin(phff);
float zo = 100 * sin(thetaff);

//alcance de puntos de la esfera
float xb = xo * largo;
float yb = yo * largo;
float zb = zo * largo;

//CUANDO NO HAY MANO LOS PUNTOS VAN A 0

if (val6 == 0) {
    strokeWeight(1);
    beginShape(POINTS);
    sphereDetail(1);
    stroke(53);
    vertex(x, y, z);
    stroke(200);
    vertex(xb, yb, zb);
    endShape();
} else {
    strokeWeight(1);
    beginShape(POINTS);
    stroke(53);
    vertex(x, y, z);
    stroke(200);
    vertex(xb, yb, zb);
    endShape();
}
}
}

```

### **9.3 Software**

#### PROCESSING

Processing es un lenguaje de programación y entorno de desarrollo integrado de código abierto basado en Java. En este proyecto se utiliza ya que es óptimo para proyectos multimedia e interactivos de diseño digital.

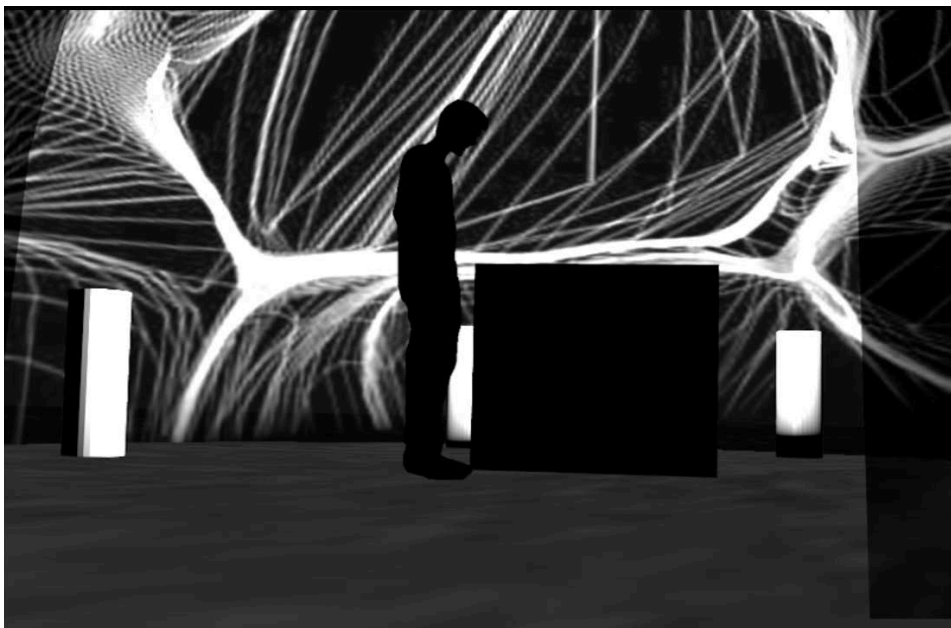
#### RESOLUME

Resolume es una herramienta de interpretación audiovisual. Permite lanzar clips de vídeo, de audio o de vídeo con audio; permite mezclarlos, manipularlos, aplicarles efectos y mostrar el resultado para una actuación en directo o para grabarlo. En este proyecto se utiliza resolume para manipular el video-mapping.

#### 9.4 Características técnicas para el proyecto Dome

Siguiendo la línea del proyecto expuesto en este trabajo, se presentan las características técnicas necesarias para el desarrollo del proyecto Domo. Dentro de la proyección audiovisual, se alterarán cuatro factores con la interacción de los sensores dentro de la pieza:

- **VOLUMEN:** refiere a la cantidad de elementos y la escala que se utiliza.
- **VELOCIDAD:** refiere a la frecuencia de actualización y movimientos de cada gráfica y sonido.
- **VALOR:** refiere a la importancia que se da a cada pieza a través de la intensidad lumínica y el volumen sonoro.
- **VARIEDAD:** refiere a las formas y perspectivas de todas las fuentes gráficas y sonoras a las que puede acceder el usuario.



*Figura 39 - Efectos producidos por la interacción en el proyecto Domo*

## FASE DE INTERACCIÓN

Para la interacción de la pieza, se presenta un nuevo dispositivo añadido a la pieza actual. Como se ha comentado en el trabajo, la contemplación del uso de este dispositivo ya estaba presente, por eso se añade en este nuevo proyecto. El dispositivo en cuestión, Kinect, se utilizará para la interactividad pasiva de los usuarios. Queda definido en la siguiente Tabla 3, dicha interacción:

*Tabla 3 - Kinect (interacción pasiva)*

<b>EXPLORACIÓN</b>	
<b>INTERACCIÓN</b>	Sinérgica y sinestésica.
<b>HERRAMIENTA</b>	Kinect
<b>ALTERACIÓN</b>	Ambiente sonoro
<b>EFFECTO VISUAL</b>	cambio mínimo en velocidad, frecuencia e intensidad lumínica de proyección fulldome.

El proceso de aprendizaje surge cuando el usuario pasa a la interacción activa, y tiene que aprender cómo funcionan los sensores y como controlar así las proyecciones de su entorno. Para ello, el primer dispositivo en cuestión será Leap Motion. La interacción con este dispositivo, similar a la explicada en el proyecto Distopía, añadirá movimientos predefinidos conocidos como *gestos*, que el dispositivo reconocerá y reaccionará de la siguiente manera:

*Tabla 4 - Leap Motion*

<b>APRENDIZAJE - MOVIMIENTO</b>	
<b>INTERACCIÓN</b>	Activa, consciente Cinética / comunicativa y sinestésica
<b>ALTERACIÓN</b>	Interfaz gráfica a través de gestos
<b>HERRAMIENTA</b>	Leap Motion
<b>EFFECTO VISUAL</b>	Cambio de formas y volumen gráfico de proyección fulldome
<b>EFFECTO SONORO</b>	Sinconizado

El sistema de interacción con Arduino seguirá la línea presentada en *Dystopia*, siendo mayor su efecto a causa de la inmersidad producida por el domo. Arduino tendrá dispositivos de proximidad que alterarán la escala e intensidad lumínica de la proyección.

Tabla 5 - Arduino

<b>EXPLORACIÓN</b>	
<b>INTERACCIÓN</b>	Activa, consciente Cinética / comunicativa y sinestésica
<b>ALTERACIÓN</b>	Objetos
<b>HERRAMIENTAS</b>	Arduino, sensores de proximidad
<b>EFEECTO VISUAL</b>	Cambio de escala e intensidad lumínica de proyección fulldome

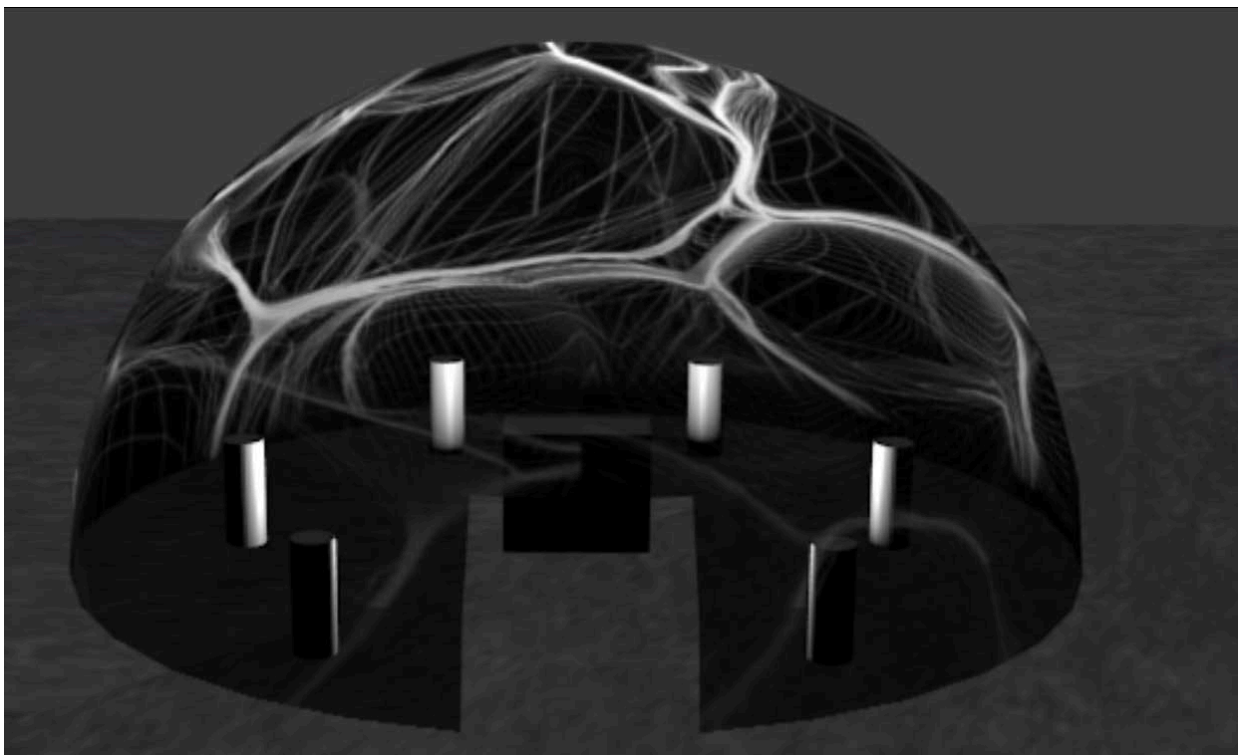


Figura 40 – Domo

El material técnico necesario para realizar este proyecto queda definido en la siguiente Tabla 6.



Tabla 6 - Material técnico para el proyecto Domo

Domo geodésico (diam:8m. / alt:4m.)
Tela retroproyección
Estructuras cilíndricas
Proyectores DLP
Leap Motion
Arduino
Kinect
Ordenadores
Altavoces
Cableado (HDMI, VGA, cable de poder, cable de red, extensiones, adaptadores)