

IMPLEMENTACIÓN DE UN THEREMIN UTILIZANDO LEAP MOTION

Javier Cervera Gomez

Tutor: Joaquín Cerdà Boluda

Trabajo Fin de Grado, presentado en la Escuela Técnica Superior de Ingenieros de Telecomunicación de la Universitat Politècnica de València, para la obtención del Título de Graduado en Ingeniería de Tecnologías y Servicios de Telecomunicación

Curso 2015-16

Valencia, 23 de mayo de 2016

Resumen

El objetivo del proyecto es evaluar las características y el funcionamiento del sensor Leap Motion de la empresa Leap Motion Inc., un dispositivo de reciente creación, el cual está teniendo una gran repercusión en el campo de la interactividad, en concreto, en el campo de la realidad virtual, ya que detecta las manos sin necesidad de llevar nada en ellas. Para ello se creará una aplicación web que consiste en un Theremin virtual que puedes modular el tono del sonido con la mano izquierda y el volumen con la mano derecha. Para la creación de esta aplicación web se va a utilizar JavaScript principalmente acompañado de HTML5 y CSS3 para su correcta visualización en cualquier navegador web, y de esta forma que se pueda utilizar desde cualquier sistema operativo, que soporte el dispositivo.

Resum

L'objectiu del projecte és avaluar les característiques i el funcionament del sensor Leap Motion de l'empresa Leap Motion Inc., un dispositiu de recent creació, el qual està tenint una gran repercussió al camp de la interactivitat, en concret, al camp de la realitat virtual, ja que detecta les mans sense portar res en aquestes. Per això es crearà una aplicació web, la qual consisteix en un theremin virtual, al qual pots modular el to del so amb la mà esquerra i el volum amb la mà dreta. Per crear aquesta aplicació web es va a utilitzar JavaScript principalment acompanyat de HTML5 i CSS3 per a la correcta visualització en qualsevol navegador web, i d'aquesta manera que es pugui utilitzar des de qualsevol sistema operatiu, que suporti el dispositiu.

Abstract

The objective of the project is to assess the characteristics and operation of the Leap Motion sensor of the company Leap Motion Inc., It is a recently created device, which is having a major impact on the field of interactivity, specifically in the field of virtual reality as it detects hands without needing to take anything in them. This requires a Web application will be created, which is a virtual Theremin, which can modulate the pitch and volume of the sound with the left hand and right hand. For the creation of this web application it is to be used mainly JavaScript accompanied by HTML5 and CSS3 for proper display in any web browser, and thus it can be used from any operating system that supports the device.

Índice

Capítulo 1. Introducción.....	3
1.1 Motivaciones	3
1.2 Objetivos	3
1.3 Estructura de la obra.....	3
Capítulo 2. Estado del arte	5
2.1 El Theremin.....	5
2.2 Dispositivos de reconocimiento de gestos	6
2.2.1 Leap Motion	7
Capítulo 3. Metodología de trabajo	10
3.1 Gestión del proyecto.....	10
3.1.1 API y plugins del Leap Motion.....	10
3.1.2 Librerías de sonido	10
3.1.3 Librerías de diseño gráfico	11
3.2 Organización del proyecto	12
Capítulo 4. Desarrollo y resultados del trabajo	14
4.1 Creación de la interfaz gráfica.....	14
4.2 Inclusión del sonido en la aplicación	19
4.3 Añadir funcionalidades extra	22
4.4 Creación de la segunda versión de la aplicación.....	26
Capítulo 5. Pliego de condiciones	28
5.1 Presupuesto	28
Capítulo 6. Conclusiones y propuesta de trabajo futuro	29
6.1 Conclusiones	29
6.2 Trabajo futuro.....	29
Capítulo 7. Manual del usuario	31
7.1 Instalación	31
7.2 Modo de empleo.....	31
7.2.1 Funcionalidades extra de las dos versiones.....	32
7.2.2 Funcionalidades extra de la primera versión.....	33
7.2.3 Funcionalidades extra de la segunda versión	34
Capítulo 8. Bibliografía.....	35
Capítulo 9. Glosario	36

Índice de figuras

Figura 1. León Theremin tocando el theremin.....	5
Figura 2. Etherwave Plus de la empresa Moog Music.....	6
Figura 3. Parte hardware del Leap Motion	7
Figura 4. Área de interacción del Leap Motion	7
Figura 5. Sistema de coordenadas del Leap Motion	8
Figura 6. Falanges que detecta el Leap Motion	9
Figura 7. Diagrama de Gantt de la planificación previa del proyecto	12
Figura 8. Diagrama de Gantt de la planificación terminado el proyecto	13
Figura 9. Modelo 3D del theremin	18
Figura 10. Modelo 3D del theremin junto con el modelo 3D de las manos	19
Figura 11. Imágenes de los tipos de onda en el menú de selección de tipo de onda	23
Figura 12. Cómo controlar el volumen de la aplicación	31
Figura 13. Cómo controlar el tono de la aplicación.....	32
Figura 14. Menú de selección de tipo de onda.....	32
Figura 15. Botón de silenciar cuando está activo y cuando no lo está	33
Figura 16. Cuadros de texto para elegir los límites de la frecuencia.....	33
Figura 17. Cuadro en el que se muestra la frecuencia del sonido que suena.....	33
Figura 18. Menú para elegir la nota de referencia	34

Capítulo 1. Introducción

1.1 Motivaciones

Una de las motivaciones que ha llevado a realizar este proyecto ha sido la oportunidad de trabajar en el campo de los sensores y en sí trabajar con el Leap Motion, un sensor enfocado al mundo de la realidad virtual y detecta las manos del usuario en tres dimensiones, se va a intentar ver el funcionamiento del Leap Motion a través de una aplicación web e intentar encontrar las limitaciones de este.

Otras de las motivaciones que promovieron a hacer este proyecto es el hecho de trabajar en una aplicación musical, ya que la aplicación va a simular un instrumento musical, el theremin, el cual es considerado uno de los primeros instrumentos electrónicos, esto hace que la aplicación tenga una utilidad en el campo de la música.

La última motivación es que al tratarse de una aplicación web, todo el mundo que tenga en su poder un Leap Motion, lo podrá utilizar sin necesidad de tener un sistema operativo en concreto, gracias a esto no se limita el software a ningún tipo de usuario.

1.2 Objetivos

El objetivo principal es crear una aplicación web que simule el funcionamiento de un theremin, el cual se trata de un instrumento que emite una nota, a un determinado volumen dependiendo de la proximidad de cada una de las dos manos del interprete, con las dos antenas que se hayan en el theremin.

Entre los objetivos de este trabajo esta valorar el funcionamiento del Leap Motion e intentar encontrar fallos en él, a través de la utilización de la aplicación.

Otro objetivo es conseguir crear una aplicación gráfica en la cual se muestren las manos del usuario, captadas a través del Leap Motion, en tiempo real. Además, que las manos que se muestran puedan controlar algún aspecto del programa en tiempo real también.

1.3 Estructura de la obra

En el segundo capítulo, se hablará sobre la historia, la utilización y la actualidad del instrumento al que se quiere simular, el theremin. También se informará sobre los dispositivos de reconocimiento de gestos, centrándose en el dispositivo que se gastará para el proyecto, el Leap Motion.

En el tercer capítulo se indicará la gestión de proyecto, más concretamente la selección de librerías y APIs que se utilizaran en él. Además, se indicará como se ha distribuido el proyecto y se marcaran las metas temporales a las que se quiere aspirar.

El capítulo cuatro trata el desarrollo de la aplicación web, los pasos que se han seguido desde el momento de empezar a programar la aplicación y como se ha dividido las tareas en cuatro principalmente, construir la interfaz gráfica, añadirle el sonido y con ello el funcionamiento básico de la aplicación, añadir funcionalidades extra para que el usuario pueda elegir distintos parámetros del theremin y crear una segunda versión.

En el quinto capítulo se enumerarán los gastos previstos para la realización del proyecto y se creará el presupuesto.

En el sexto capítulo se nombrará las conclusiones sobre el proyecto y los elementos que conforman este y se indicará cuál es el trabajo futuro para que evolucione el proyecto.

En séptimo capítulo se indicará cómo realizar la instalación del dispositivo y de la aplicación y cómo emplear la aplicación.

En el octavo capítulo se indica de dónde se ha sustraído la información que se encuentra en el trabajo.

El último apartado está reservado para definir conceptos técnicos para que todo lector pueda comprender el texto.

Capítulo 2. Estado del arte

2.1 El Theremin

El theremin es un instrumento musical inventado por el doctor en física ruso Lev Serguéyevich Termén, más conocido como León Theremin. En 1920 fue la primera vez que se mostró este invento a un grupo de científicos soviéticos bajo el nombre de ‘Etherphone’.

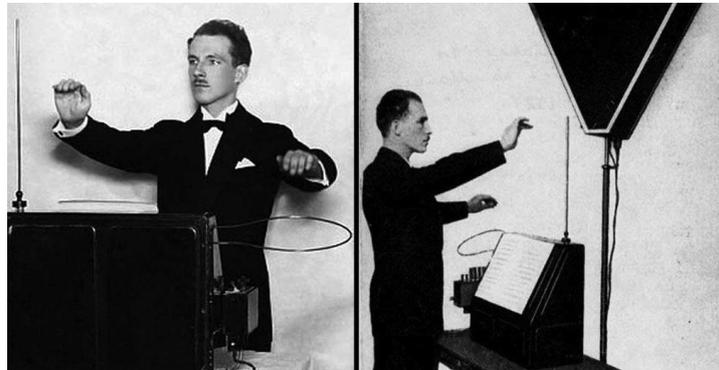


Figura 1. Léon Theremin tocando el theremin

Este instrumento se basa en la obtención de sonidos de tono variable, utilizando una corriente alterna de frecuencia variable, la antena vertical que tiene el instrumento crea pequeñas ondas electromagnéticas alrededor de ella, cuya intensidad es muy débil. Cuando se aproxima una mano hace que se varíe el campo electromagnético que rodea la antena, ya que la mano es un conductor eléctrico, esto afecta a la capacitancia de la antena, lo que afecta a la frecuencia de la corriente alterna creada por el theremin. Esto hace que cuanto más cerca este la mano de la antena vertical el tono que suene sea más agudo. Del mismo modo se encuentra otra antena de forma circular, la cual regula la intensidad del sonido que produce el instrumento, en este caso cuanto más cerca se encuentre la mano de esta antena menor será el volumen, hasta llegar a ser inaudible.

El sonido del theremin, se produce gracias al principio heterodino, en el cual un oscilador de frecuencia variable afecta la oscilación de un oscilador de frecuencia fija, haciendo que la frecuencia resultante caiga dentro del rango audible para los humanos y reproduciendo esta por un altavoz.

El instrumento se ha utilizado en canciones de grupos como Led Zeppelin, Beach Boys, o Pink Floyd. [1]

También ha sido utilizado en muchas bandas sonoras de películas, de miedo y de ciencia ficción como hizo Bernard Herrmann en la BSO de la película “Ultimátum a la tierra” (Robert Wise 1951), como hizo también Miklos Rozsa en la BSO de la película “Recuerda” (Alfred Hitchcock), o como utilizo el español Roque Baños en la película “El Maquinista” (Brad Anderson 2004). [2]

Actualmente la empresa más relevante en la venta del theremin es Moog Music, que distribuyen el modelo Etherwave Plus, aparte de este modelo, se comercializan otros modelos de otros fabricantes como el B3 Pro diseñado por Dan Burns. [3]

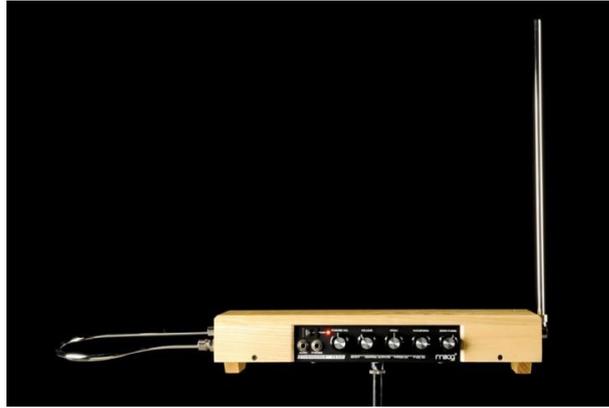


Figura 2. Etherwave Plus de la empresa Moog Music

En la actualidad, también existen modelos de theremin, tanto profesionales como DIY (Do It Yourself), que incorporan la tecnología MIDI (Musical Instrument Digital Interface). Además, en el apartado software también existen aplicaciones que simulan el funcionamiento de este instrumento, manejándola con periféricos como con un ratón del ordenador, con una pantalla táctil o con un sensor externo como va a suceder en la aplicación creada en este trabajo.

2.2 Dispositivos de reconocimiento de gestos

En la actualidad, la interacción entre las tecnologías (los ordenadores, las videoconsolas y la domótica entre otras) y las personas está creciendo considerablemente, y en este aspecto el reconocimiento de gestos puede considerarse que sigue el camino para aumentar aún más esta interacción. El reconocimiento de gestos es el proceso por el cual el receptor reconoce los gestos que realizan los dedos, las manos, los brazos, la cabeza, la cara o el cuerpo del usuario, con la intención de transmitir información significativa al receptor.

Estos gestos pueden clasificarse en estáticos (el usuario asume una cierta pose) o dinámicos (se detectan los movimientos del usuario). En ocasiones los gestos pueden tener las dos características, ya que puede que el gesto empiece y acabe en un gesto estático y haya una transición dinámica. Además de esta clasificación se puede realizar otra, según de que parte del usuario se van a recoger los datos, estos son, los gestos del cuerpo que se utilizan por ejemplo para la rehabilitación médica y el entrenamiento atlético, los gestos de la cabeza y la cara que se pueden utilizar para detectar la dirección de la mirada o el momento en el que empieza el usuario hablar, por último están los gestos de las manos y los brazos que estos se emplean para el reconocimiento de las posiciones de las manos para aplicaciones de entretenimiento.[4]

Para detectar estos gestos es necesario tener el dispositivo adecuado para cada tipo. El primer dispositivo que más fama tubo en este aspecto fue el Kinect de Microsoft, el cual es para reconocer gestos del cuerpo y la primera finalidad que tubo fue la de detectar los movimientos del cuerpo para que los reprodujera los personajes de los videojuegos de Xbox 360.

Actualmente, la mayoría de los dispositivos de reconocimiento de gestos se están centrando en detectar los gestos de las manos y los brazos, en este apartado podemos encontrar el dispositivo Myo de la empresa Thalmic Labs Inc. que se trata de un brazalete que se conecta al ordenador o al smartphone mediante Bluetooth, este brazalete contiene electrodos superficiales que mediante electromiografía (EMG) recogen la actividad muscular del brazo, esto junto a una unidad de

medición inercial que mide la velocidad, orientación y fuerzas gravitacionales del aparato, se recrean la posición de la mano y del brazo.[5][6]

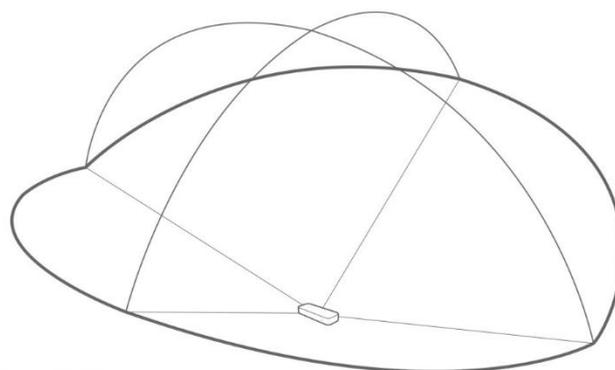
2.2.1 Leap Motion

El Leap Motion es el dispositivo de reconocimiento de gestos que se utilizará en este trabajo, se trata de un dispositivo que detecta los gestos de las manos y los brazos. En la parte hardware el dispositivo trata de dos cámaras infrarrojas con un angular de 120° junto a tres LEDs infrarrojos. Estos sensores trabajan en una longitud de onda de 850 nanómetros, lo que hace que trabaje en un espectro no visible por el ojo humano.



Figura 3. Parte hardware del Leap Motion

El área de interacción se encuentra en la parte superior al dispositivo (si este se encuentra apoyado en una superficie horizontal con las cámaras apuntando hacia arriba) y forma una pirámide invertida con vértice en el dispositivo, con un ángulo de visión de 150° y con una altura máxima de 60 centímetros, si está apoyado el dispositivo sobre una superficie, la cual aumenta hasta los 80 centímetros si se utiliza el software Orion que está enfocado a las aplicaciones de realidad virtual, esta altura está limitada por la luz que propagan los LEDs.[7]



Interaction Area
2 feet above the controller, by 2 feet wide on each side
(150° angle), by 2 feet deep on each side (120° angle)

Figura 4. Área de interacción del Leap Motion

El dispositivo se conecta al ordenador mediante una conexión USB. El dispositivo es compatible con los sistemas operativos Windows, Linux y Mac, además, la empresa distribuye APIs para poder programar en Python, Java, C++, C#, Objective-C, y JavaScript, además también distribuye plugins para los motores gráficos Unreal Engine y Unity.

El Leap Motion emplea el sistema Cartesiano de coordenadas para trabajar. El origen se encuentra centrado en la parte superior del controlador. Los ejes 'x' y 'z' se encuentran en el plano horizontal, con el eje 'x' en dirección paralela al lado más largo del dispositivo y el sentido del eje 'z' apuntando hacia el usuario. El eje 'y' corresponde al eje vertical con sentido ascendente.

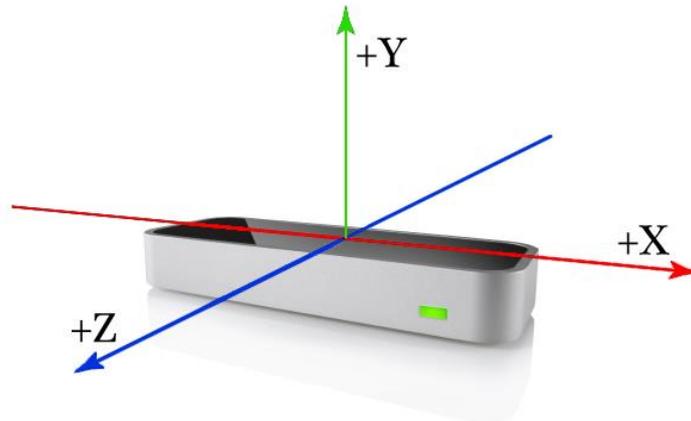


Figura 5. Sistema de coordenadas del Leap Motion

El controlador estudia cada imagen en busca de manos y dedos, envía un conjunto de datos o frames al ordenador. Cada objeto frame contiene datos sobre las manos, los dedos, también contiene el movimiento que describen las manos y detecta algunos gestos predeterminados que hay en cada imagen. Estos datos son el resultado de la comparación de las imágenes con un modelo interno que tiene el Leap Motion.

Dentro de estos datos existen diferentes clases, la primera es la clase 'hand' en la cual están los datos sobre qué mano es (izquierda o derecha), la posición, la dirección e incluso el porcentaje de coincidencia de la mano detectada con el modelo interno que contiene el Leap Motion. En esta clase existen otras subclases, una de ellas es la clase 'arm' en la que existen datos sobre las orientaciones de los distintos brazos, sus longitudes, sus anchuras y donde finaliza cada uno de ellos, otra subclase es la referente a cada dedo, la clase 'finger', en la que se encuentran los datos sobre la dirección de los dedos, la posición de cada falange, el tamaño de cada una y el ángulo de estas

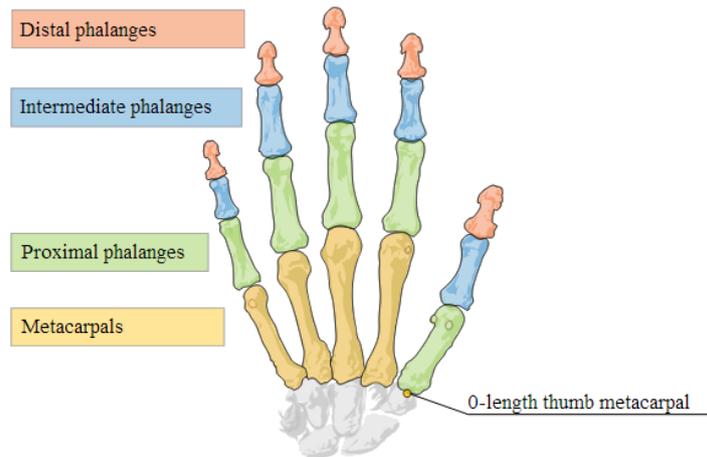


Figura 6. Falanges que detecta el Leap Motion

Además de estos datos de la mano, el dispositivo también es capaz de detectar algunos gestos como bien se ha dicho antes, estos gestos son, hacer círculos con el dedo índice, un movimiento lineal continuo con la mano, mover el dedo índice como si estuvieras apretando un pulsador y mover el dedo índice como si estuvieras traspasando un plano vertical paralelo al eje 'x' del dispositivo. [8]

Capítulo 3. Metodología de trabajo

3.1 Gestión del proyecto

Para la creación de una aplicación web que se simule el funcionamiento del instrumento theremin, hay que tener en cuenta que la posición de las manos, con respecto al dispositivo que se va a utilizar para la captura de los gestos, es muy relevante y que la respuesta de la aplicación dependerá de esta. Para saber esta posición y representarla en la pantalla se utilizará la API y los plugins que proporciona la empresa Leap Motion Inc. ya que para recoger los datos se utilizará el sensor Leap Motion. Además, se ha de recurrir a librerías gráficas y a librerías de sonido, ya que al tratarse de la simulación de un instrumento musical es muy importante el sonido que reproduzca la aplicación y como se trata de un instrumento en el que es muy importante la posición de las manos con respecto al instrumento, también es importante crear una referencia visual en la aplicación y para ello se necesita utilizar un interfaz gráfico.

La principal herramienta que se va a utilizar en la aplicación web, es el lenguaje de programación JavaScript, acompañado de HTML y CSS3 para el correcto funcionamiento de la aplicación. Se va a utilizar estos lenguajes ya que casi todos los navegadores actuales son compatibles con ellos y haciendo que la aplicación sea multiplataforma y se pueda utilizar en cualquier dispositivo electrónico.

3.1.1 API y plugins del Leap Motion

En la aplicación se va a utilizar la API proporcionadas por el fabricante de los cuales se ha hablado antes, que proporcionan datos sobre, entre otras cosas, las posiciones de las manos con respecto al dispositivo, el movimiento y los gestos. A parte de la API, también se utilizarán algunos de los plugins que proporciona el fabricante en su página web para desarrolladores. [9]

LeapJS Plugins es la colección más básica de plugins que comparte la empresa y entre sus funcionalidades se encuentran, por ejemplo, obtener la posición de la mano en la pantalla que se extrapola de la posición en la que encuentra en el espacio de interacción del Leap Motion o hacer una translación, una rotación o un escalado de los datos proporcionados por el sensor rápidamente. [10]

JS Rigged Hand Plugin se trata de la segunda librería de plugins que se va a utilizar en la aplicación, para utilizar esta librería se necesita utilizar la librería gráfica *Three.js*, que se hablara posteriormente y la librería *LeapJS-Plugins* la cual se ha descrito en el párrafo anterior. Esta librería se utiliza para añadir la imagen de las manos en cualquier escena *Three.js*. [11]

Además de estas librerías la empresa comparte otras librerías de plugins como es *LeapJS 3D Widgets*, que sirve para añadir la funcionalidad de apretar botones y arrastrar planos con gestos recogidos por el Leap Motion.

3.1.2 Librerías de sonido

En este apartado, la W3C no ha encontrado la solución para crear una librería de audio que contente a los desarrolladores, por este motivo han ido apareciendo varias librerías de audio independientes, con distintas características para satisfacer los diferentes casos que pueden ocurrir, como cada librería está enfocada a objetivos diferentes, no se pueden considerar librerías para uso general, entre estas librerías podemos encontrar las siguientes:

- **<audio>**: No se trata de una librería de JavaScript, es una etiqueta HTML y es la solución que el W3C ha encontrado para poder reproducir audio en cualquier página web, el problema es que no es compatible con todos los navegadores. Soporta los archivos de audio MP3, Ogg y Wav y aquí surge otro problema de la etiqueta ya que cada navegador soporta la etiqueta no admite todos los tipos de archivo que la etiqueta soporta. Para manejar el reproductor se hace mediante JavaScript utilizando unos métodos sobre el objeto DOM de la etiqueta.
- **Buzz!**¹: Librería de JavaScript, creada por Jay Salvat que utiliza la etiqueta antes nombrada <audio> para cargar y manejar archivos de sonido, las limitaciones son las mismas que para la etiqueta que utiliza y si no es soportada la etiqueta se queda en silencio.
- **SoundJS**²: Librería creada por la plataforma CreateJS que soporta la etiqueta <audio> de HTML, el reproductor Flash player y el reproductor Cordova para reproducir archivos de audio, gracias a esto tiene más navegadores compatibles. Esta librería esta liberada bajo licencia MIT.
- **jPlayer**³: Plugin para jQuery, con licencia MIT/GPL creada por la empresa Happyworm Ltd, que reproduce archivos de audio sobre HTML5 y sobre Flash.
- **Timbre.JS**⁴: Librería que permite la creación de sonidos mediante un simple sintetizador, creada por el usuario de GitHub mohayonao y con licencia MIT.
- **MIDI.js**⁵: Librería que contiene un reproductor de archivos MIDI para páginas web, trabaja sobre la Web Audio API del W3C y los navegadores que no soportan este API, trabaja sobre métodos convencionales basados en plugins.
- **Web Audio API**⁶: API de alto nivel para el procesamiento y sintetizar audio, creado por el W3C. Esta API incluye la generación de un oscilador con el tipo de onda que desea el usuario.

Evaluando las características de cada librería de sonido antes numeradas y primero descartando las que solo se encargaban de reproducir archivos de audio y no de sintetizar audio a partir de un oscilador al cual el desarrollador puede cambiar la frecuencia de oscilación, se decidió utilizar la API Web Audio API, ya que para hacer este proyecto simplificaba mucho el trabajo gracias a su fácil funcionamiento y a sus características que permiten la creación de un oscilador que el desarrollador le puede modificar el tipo de onda y la frecuencia a la que trabaja y suena.

3.1.3 Librerías de diseño gráfico

En este apartado, JavaScript tiene varias librerías de diseño gráfico 3D, aunque hay una que es de referencia para la mayoría de desarrolladores del lenguaje, pero aun así se tendrán en cuenta alguna librería más por si hubiera alguna que se adaptara mejor a la aplicación.

- **Three.js**⁷: Esta es la librería de referencia para los que buscan crear objetos 3D en sus páginas web, ya que es muy completa y tiene una comunidad muy activa. A diferencia de las demás librerías esta recibe actualizaciones de forma frecuente y existen muchos ejemplos en su página web. Esta librería está desarrollada por el español Ricardo Cabello.

¹ Página web de la librería Buzz! < <http://buzz.jaysalvat.com/>>

² Página web de la librería SoundJS < <http://www.createjs.com/soundjs>>

³ Página web de la librería Jplayer < <http://jplayer.org/>>

⁴ Página web de la librería timbre.js < <http://mohayonao.github.io/timbre.js/>>

⁵ Página web de la librería MIDI.js < <http://www.midijs.net/>>

⁶ Página web oficial de la API Web Audio API < <https://www.w3.org/TR/webaudio/>>

⁷ Página web de la librería Three.js < <http://threejs.org/>>

Una de las características que ayudan a esta librería a ser referente en este campo es el editor gratuito que hay en su página web y eso junto a que puedes importar modelos de otros editores gráficos como Blender o Maya.

- **3difier⁸**: Esta librería está también bastante extendida en muchísimos sitios web, aunque con ella no se logran efectos 3D puros, a cambio es muy sencilla de utilizar para el desarrollo. El desarrollador es Owen Galen y los requisitos que necesita es HTML5 con soporte de objetos <canvas> y jQuery.
- **Oimo.js⁹**: Es un motor 3D que imita las propiedades físicas de una escena real, gracias a esto se pueden crear complejas animaciones que responden a leyes físicas. Esta librería está desarrollada por el francés Laurent Thillet. Una de las ventajas que tiene esta librería es que trabaja combinando sus funcionalidades con las de otras librerías gráficas 3D como la anteriormente comentada Three.js.

Después de ver las ventajas de cada una y las desventajas, se utilizará la librería Three.js, ya que se requiere esta librería en los plugins distribuidos por el fabricante de Leap Motion, además por su gran cantidad de ejemplos, los cuales ayudan al rápido aprendizaje de cómo utilizar la librería.

3.2 Organización del proyecto

La aplicación web se puede dividir en varias tareas para hacer, la primera es aprender a utilizar los idiomas de programación que se van a utilizar para este proyecto (HTML, CSS3 y JavaScript), después esta la búsqueda de las librerías que se van a utilizar en el proyecto, que ya está hecho en el apartado 2.1 de esta memoria, a continuación, se creará la parte gráfica de la aplicación, para seguidamente incluir la parte de sonido, después de haber terminado esta parte se le añadirán funcionalidades extra a la aplicación web y por último se escribirá este documento.

Ahora que ya se saben las tareas a realizar, hay que gestionar el tiempo en el que se van a hacer estas tareas, esto está plasmado en la figura 7.



Figura 7. Diagrama de Gantt de la planificación previa del proyecto

⁸ Página web de la librería 3difier < <https://github.com/owengalenjones/3difier>>

⁹ Página web de la librería Oimo.js < <https://github.com/lo-th/Oimo.js>>

Después de haber terminado el proyecto, la planificación ha cambiado, debido a problemas externos, ya que el autor estuvo trabajando en un ayuntamiento los meses de julio y agosto, y al tiempo para realizar cada tarea que en algunos momentos se alargó o se acortó y se añadió una tarea más, ya que se realizaron dos versiones de la aplicación por una cuestión surgida durante la creación de la aplicación web. Después de haber terminado el proyecto el diagrama de Gantt queda como se ve en la figura 8.



Figura 8. Diagrama de Gantt de la planificación terminado el proyecto

Capítulo 4. Desarrollo y resultados del trabajo

4.1 Creación de la interfaz gráfica

Para la creación de la interfaz gráfica de la aplicación web, como se indicó anteriormente, se utilizará la librería Three.js por dos motivos, que es la librería de referencia para la mayoría de desarrolladores en JavaScript y porque es una librería necesaria para implementar los plugins distribuidos por los fabricantes del Leap Motion.

La primera tarea dentro de la interfaz gráfica es crear un modelo 3D del theremin, para ello se utilizará el editor¹⁰ online que proporciona la página web oficial de la librería. El modelado del instrumento lo forman un prisma central, con una textura simulando a la madera, en la parte derecha se encuentra un toroide incrustado en este prisma, lo que simulara la antena del volumen del sonido, por último, en la parte izquierda se encuentran dos cilindros formando un ángulo de noventa grados. Después de haber realizado el modelizado se descarga del editor el archivo JSON, pero se ocurre un problema cuando se intenta cargar este archivo en el navegador Google Chrome y por ello se empieza otra vez a diseñar desde cero, utilizando los métodos que incluye la librería que introducen uno a uno los dibujos gráficos en un elemento <canvas>, que es una etiqueta de HTML5 que sirve como un lienzo, para dibujar cualquier cosa. Esta parte del código al final quedo de la siguiente forma:

```
WIDTH = window.innerWidth*0.9; // Ancho de pantalla
HEIGHT = window.innerHeight; // Alto de pantalla

window.lienzo = new THREE.WebGLRenderer();// Lienzo u objeto
encargado del renderizado
lienzo.setClearColor( 0x262525 ); //Color de fondo
lienzo.setPixelRatio( window.devicePixelRatio );
lienzo.setSize(WIDTH,HEIGHT);//Establecemos las dimensiones del
lienzo
document.body.appendChild(lienzo.domElement);//Añadimos el
lienzo a la página
window.scene = new THREE.Scene();//Creamos la scene

//Cubo
var geocube = new THREE.BoxGeometry(
2.50, //Dimensiones en x
2.50, //Dimensiones en y
2.50 //dimensiones en z
); //Creamos un prisma
```

¹⁰ El editor gráfico de la librería Three.js se encuentra en la siguiente página web <http://threejs.org/editor/>

```

    var texture = new THREE.TextureLoader().load(
"Texture/madera.gif" );
//Cargo la textura de la madera
    var apcube = new THREE.MeshBasicMaterial({color:0x441A00,map:
texture});
//Creamos la apariencia

var cube = new THREE.Mesh(geocube,apcube);//Generamos el prisma y
le aplicamos color
cube.scale.set(3.9,1,1);
scene.add(cube);//Añadimos el cubo a la scene

//Toridio
var geotor = new THREE.TorusGeometry(
5.38,//Radio
0.32,//Diametro del tubo
30,//Segmentos Radiales
180,//Segmentos tubulares
5.18//Arco
);//Creamos un toridio
var aptor = new THREE.MeshStandardMaterial({
    color: 0xffffffff,
    emissive: 0x000000
}); //Creamos la apariencia
var torus = new THREE.Mesh(geotor,aptor);//Generamos el toridio y
le aplicamos color
torus.position.set(6.7,0.08,-0.16);//Ponemos el toridio en su
posicion
torus.rotation.set(1.56,0.04,-2.62);//Rotamos el toridio
torus.scale.set(0.44,0.44,0.48);//Escalamos el toridio
scene.add(torus);//Añadimos el toridio a la scene

//Cilindro vertical
var geocill = new THREE.CylinderGeometry(
1,//Radio de arriba
1,//Radio de abajo
4,//altura
32,//Segmentos Radiales
1//Segmentos de altura

```

```

); //Creamos un cilindro
var apcil1 = new THREE.MeshStandardMaterial({
    color: 0xffffffff,
    emissive: 0x000000
}); //Creamos la apariencia
var cil1 = new THREE.Mesh(geocil1, apcil1); //Generamos el cilindro
y le aplicamos color
cil1.position.set(-6.44, 4.33, 0.08); //Ponemos el cilindro en su
posicion
cil1.scale.set(0.1, 2.18, 0.10); //Escalamos el cilindro
scene.add(cil1); //Añadimos el cilindro a la scene

//Cilindro horizontal
var geocil2 = new THREE.CylinderGeometry(
1, //Radio de arriba
1, //Radio de abajo
4, //altura
32, //Segmentos Radiales
1 //Segmentos de altura
); //Creamos un cilindro
var apcil2 = new THREE.MeshStandardMaterial({
    color: 0xffffffff,
    emissive: 0x000000
}); //Creamos la apariencia
var cil2 = new THREE.Mesh(geocil2, apcil2); //Generamos el cilindro
y le aplicamos color
cil2.position.set(-5.69, -0.07, 0.01); //Ponemos el cilindro en su
posicion
cil2.rotation.set(2.68, -0.02, 1.56); //Rotamos el cilindro
cil2.scale.set(0.1, 0.42, 0.10); //Escalamos el cilindro
scene.add(cil2); //Añadimos el cilindro a la scene

window.camera = new
THREE.PerspectiveCamera(75, (WIDTH/HEIGHT), 0.1, 100);
//Creamos la camara
camera.position.x=0;
camera.position.y=3;
camera.position.z=10;

```

```

scene.add(camera);//Añadimos la camera a la scene

// Creamos una par de focos de luz
var luz1 = new THREE.PointLight(0xffffff); //Punto de luz
luz1.position.set(
    120, // Posición en eje X
    260, // Posición en eje Y
    100 // Posición en eje Z
);

var luz2 = new THREE.DirectionalLight( 0xffffff, 1 ); //Luz
direccional
luz2.position.set(
    5.0, // Posición en eje X
    10.0, // Posición en eje Y
    7.5 // Posición en eje Z
);

// Añadimos las luces
scene.add(luz1);
scene.add(luz2);

lienzo.render(scene,camera);

window.addEventListener( 'resize', function () {
    camera.aspect = window.innerWidth*0.9 / window.innerHeight;
    camera.updateProjectionMatrix();
    lienzo.setSize( window.innerWidth*0.9, window.innerHeight );
    lienzo.render(scene,camera);
} );//Para que se ajuste a la pantalla completa

```



Figura 9. Modelo 3D del theremin

Después de haber terminado el modelo del theremin como antes hemos dicho, creando el modelo 3D solo con código en un elemento <canvas>, pero siguiendo las referencias del modelo creado en el editor, se introduce el modelo de las manos que viene implementado en el plugin del Leap Motion *JS Rigged Hand Plugin*, este lo que hace es coger el elemento <canvas> sobre el que se quiere trabajar e incluye el modelo de las manos, el problema es que si se utilizaba el elemento <canvas> del modelo del theremin, el modelo de las manos salía en otra perspectiva y con un escalado el cual no se podía modificar para que las dimensiones de los elementos concordaran. Por ello la solución fue primero poner el elemento <canvas> del modelo del theremin y después invocar otro elemento <canvas> este sin fondo y con otra cámara para que hubiera concordancia entre todos los elementos.

A parte de la solución que se hizo para no tener problemas con los tamaños de los elementos, se encontró otro problema referido con el espacio de interacción del Leap Motion, que es muy pequeño para los movimientos que se deseaba realizar para manejar el instrumento, y por ello el centro de referencia se movió hacia la derecha, ya que según el funcionamiento que se buscaba, la mano derecha solo realiza movimientos en el eje vertical y la mano que daba problemas en este caso era la mano izquierda, ya que para que cambie el tono se tiene que mover en el eje horizontal y aquí era donde no había suficiente espacio, además de esto, también se hizo que el modelo de la mano hiciera el doble de recorrido en la pantalla del ordenador con respecto a la realidad, esto incluyo un efecto de más amplitud en la zona de interacción del Leap Motion. La inclusión de las manos en el código queda de la siguiente forma:

```
var controller;
window.controller = controller = new Leap.Controller({
  background: true
});
var escena = new THREE.Scene();

controller.use('handHold', {})
```

```

controller.use('handEntry', {})
controller.use('riggedHand');
scope = controller.plugins.riggedHand; //variable para contolar
las características del plugin
scope.scale = 0.05;
scope.positionScale = 0.875;
scope.camera.position.x=-30;
scope.camera.position.y=270;

controller.connect();//Conectamos el Leap Motion

controller.on('frame', function(frame) {
    //Este es el lugar donde se pondra la funcionalidad de las
manos en el programa
})//Aqui recibimos los frames de cada mano

```

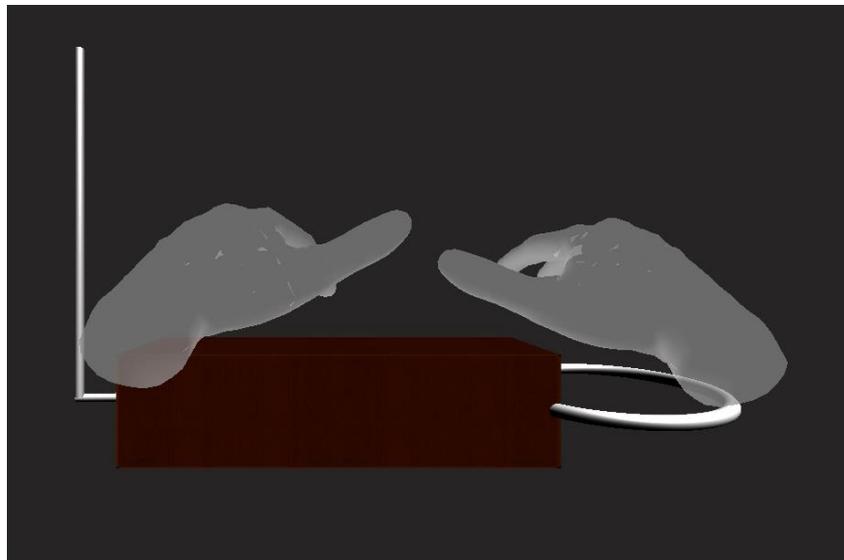


Figura 10. Modelo 3D del theremin junto con el modelo 3D de las manos

4.2 Inclusión del sonido en la aplicación

A continuación de haber integrado tanto el modelo de las manos como el modelo del theremin en la aplicación web, el siguiente paso a seguir, es incluir el sonido que se producirá al manejar la aplicación, para ello se va a utilizar la API proporcionada por el W3C, Web Audio API, en esta se va a crear un oscilador, que en principio su salida es un tren de pulsos cuadrados. Para comprobar el funcionamiento en primera instancia se hace que suene un tono de 1000 Hz cuando el dispositivo Leap Motion detecte dos manos. Después de comprobar el correcto funcionamiento de la API, se empieza a implementar la función que la mano derecha tendrá sobre el sonido, la función que se desea es que cuando se encuentre esta mano sobre la antena circular en la pantalla,

se pueda manejar el volumen del tono que varía dependiendo de la altura a la que se encuentra la mano, cuanto más alta se encuentre la mano, mayor será el volumen del tono que sonará. El problema que surge con esta mano es que no se sabe en qué coordenadas de la zona de interacción del Leap Motion se encuentra la antena circular, para ello se modifica la aplicación para que por consola salgan las coordenadas, en las que se encuentra el dedo índice de la mano derecha y mediante pruebas se encuentra la localización de esta antena, para que si la mano no está encima de la antena no suene nada.

Con la funcionalidad de la mano derecha ya implementada, toca implementar la funcionalidad de la mano izquierda, que es la de manejar el tono, en este caso al tratarse de una antena vertical, la que se toma de referencia, el movimiento que maneja el tono es un movimiento horizontal, haciendo que contra más cerca de la antena se encuentre la mano el tono de salida sea más grave, en definitiva, que la frecuencia del oscilador sea más alta. En esta implementación hay dos problemas, el primero es el mismo que ha surgido en la implementación de la funcionalidad de la mano derecha, es que no se saben las coordenadas donde se encuentra la antena vertical, para ello se utiliza la misma solución que en el caso anterior, lo que pasa que en este caso saldrá por la consola de JavaScript las coordenadas del dedo índice de la mano izquierda, tras unas pruebas se localiza la antena vertical. El segundo problema es que relación crear entre movimiento y frecuencia, ya que la diferencia de frecuencia que hay entre las notas musicales no es constante y esto hace que crear una relación lineal entre el movimiento y la frecuencia no cree una relación lineal entre notas musicales y el movimiento. En la primera versión de la aplicación se ha optado porque el movimiento y la frecuencia tengan una relación lineal, así pudiendo reproducir frecuencias que hay entre las notas musicales con más facilidad. Finalmente, el resultado de introducir esta parte en el programa queda así:

```
var oscillator, gainNode, mute, maxFreq, minFreq;
//Inicializar la libreria de audio
var      AudioContext      =      window.AudioContext      ||
window.webkitAudioContext;
var audioCtx = new AudioContext();
oscillator = audioCtx.createOscillator();//Creas el oscilador
gainNode = audioCtx.createGain();//Creas el amplificador
oscillator.connect(gainNode);
gainNode.connect(audioCtx.destination);

//Opciones del oscilador
var maxVol = 0.2;
var initialFreq = 1000;
var initialVol = 0;

controller.on('frame', function(frame) {
    var hand, handMesh, offsetDown, offsetForward, pos, dis;
    if (!frame.hands[1]) {//Si hay menos de 2 manos (frames 0 y
1)
        gainNode.gain.value = 0 * maxVol;
        document.getElementById("nota").innerHTML = "###";
```

```

        return;
    }
    if(frame.hands[1]){//Si hay 2 manos o mas
        if(mute==0){//La variable mute se añade pensando en las
funcionalidades extra
            hand=frame.hands
            for(i=0;i<frame.hands.length;i++){
                if(hand[i].type=='right'){
                    derecha = hand[i];//Para Saber que mano
es izquierda o derecha
                }else{
                    izquierda = hand[i];
                }
            }

            derpo=derecha.palmPosition
            if(derpo[0]>79){
                if (derpo[1]<300 && derpo[1]>90) {
                    gainNode.gain.value = maxVol *
(derpo[1]-90) /300 ;
                }else if(derpo[1]>300){
                    gainNode.gain.value = maxVol;
                }else{
                    gainNode.gain.value = maxVol/300 ;
                }
            }else{
                gainNode.gain.value = 0;
            }

            var min_dis = 200;

            var pos_man =
[izquierda.palmPosition,izquierda.fingers[0].carpPosition,
izquierda.fingers[0].mcpPosition,izquierda.fingers[0].dipPositio
n,
izquierda.fingers[0].pipPosition,izquierda.fingers[1].carpPositi
on,
izquierda.fingers[1].mcpPosition,izquierda.fingers[1].dipPositio
n,
izquierda.fingers[1].pipPosition,izquierda.fingers[2].carpPositi
on,

```

```

izquierda.fingers[2].mcpPosition,izquierda.fingers[2].dipPositio
n,
izquierda.fingers[2].pipPosition,izquierda.fingers[3].carpPositi
on,
izquierda.fingers[3].mcpPosition,izquierda.fingers[3].dipPositio
n,
izquierda.fingers[3].pipPosition,izquierda.fingers[4].carpPositi
on,
izquierda.fingers[4].mcpPosition,izquierda.fingers[4].dipPositio
n,
izquierda.fingers[4].pipPosition];//Todas las posiciones de la
mano y los dedos izquierdos
        pos_man.forEach(function(name) {
            dis=Math.sqrt(Math.pow(-220-
name[0],2)+Math.pow(name[2],2));

            if(min_dis>dis){

min_dis=Math.abs(Math.floor(dis));//Calculo de la distancia
minima

            }

        })
        oscillator.frequency.value = eval(((200-min_dis)/200) *
(maxFreq-minFreq) + minFreq);
        //Valor del oscilador
    }

});

```

4.3 Añadir funcionalidades extra

Ya con la aplicación en funcionamiento, se le añade nuevas funcionalidades para tener más opciones de uso.

La primera funcionalidad que se le añade es para que el usuario pueda elegir entre el tipo de onda que suena, las opciones que se proponen son las mismas que ofrece la API de sonido, que son: onda cuadrada o tren de pulsos, onda triangular, onda de sierra y onda sinusoidal. Para que el usuario tenga disponible el elegir cualquier tipo de onda se crea un menú en la parte izquierda de la aplicación, con cuatro imágenes, cada una de ellas representa los tipos de onda que hay disponibles, en tres imágenes el dibujo del tipo de onda será rojo y en una será verde, este sistema señalizara cual se está utilizando, ya que cuando se seleccione alguna la que estaba en verde se convertirá en roja y la seleccionada por el usuario se coloreará en verde. Para crear ese efecto

hace falta ocho imágenes tipo SVG, seis de estas han sido creada desde un editor¹¹ online de imágenes SVG, con las cuales se ha creado la imagen roja de las ondas triangular, cuadrada y de sierra, después de tenerlas terminadas y guardadas se ha hecho una copia y con un editor de texto¹² solo se cambia el color de las imágenes por verde en una copia, así se obtienen las seis primeras imágenes, para la imagen de la onda sinusoidal, se ha hecho una cosa distinta, ya que la forma de esta es más compleja de crear en el editor, lo que se ha hecho es descargarse la imagen SVG de la forma de onda sinusoidal del banco de imágenes de Wikipedia¹³ y esta modificarla en el editor de imágenes hasta tener los resultados que se busca.



Figura 11. Imágenes de los tipos de onda en el menú de selección de tipo de onda

La siguiente funcionalidad es añadir un botón de mute, el cual se encuentra en el menú creado al lado izquierdo de la aplicación, se trata de un botón en el que pone el texto MUTE y cuando se presiona con el ratón del ordenador el texto del botón cambia por SOUND y en la aplicación no se reproduce sonido, para hacer que vuelva a sonar la aplicación hay que volver a presionar el botón con el ratón del ordenador y volverá a funcionar como al principio.

La tercera funcionalidad extra que existe en la primera versión, es que el usuario pueda elegir la frecuencias más aguda y más grave a la que llegue el instrumento, para ello se han creados dos cuadros de texto en el menú derecho de la aplicación, en los que están indicados los límites que se le han puesto por defecto en Hercios. La frecuencia más aguda o como está indicado en la aplicación frecuencia mínima por defecto es 20 Hz y la frecuencia más grave o frecuencia máxima es 3000 Hz. Si se cambia el valor de alguno de estos parámetros hace que se cambie la relación entre el movimiento y la frecuencia la cual sigue la ecuación 1.

$$\begin{aligned}
 Frecuencia_{salida} &= (Frecuencia_{Maxima} - Frecuencia_{Minima}) * \%distancia \\
 &+ Frecuencia_{Minima} \\
 0 \leq \%distancia &\leq 1
 \end{aligned}
 \tag{1}$$

¹¹ Página del editor online de imágenes SVG < <http://vectorpaint.yaks.co.nz/>>

¹² Para el proyecto se ha utilizado el editor de texto Notepad++

¹³ La imagen tomada de referencia para el tipo de onda sinusoidal se encuentra en la siguiente dirección web < <https://es.wikipedia.org/wiki/Sinusoide#/media/File:Sin.svg>>

Todas estas funcionalidades quedan plasmadas en JavaScript de la siguiente manera:

```
var CU,TR,SI,DS;

function forma(sel){ //Funcionalidad de cambiar de tipo de onda
    if(sel==1){ //Onda tipo cuadrada
        oscillator.type = 'square';
        CU.attr("src", "images/CU_green.svg");
        TR.attr("src", "images/TR_red.svg");
        DS.attr("src", "images/DS_red.svg");
        SI.attr("src", "images/SI_red.svg");
        return

    }else if(sel==2){ //Onda triangular
        oscillator.type = 'triangle';
        CU.attr("src", "images/CU_red.svg");
        TR.attr("src", "images/TR_green.svg");
        DS.attr("src", "images/DS_red.svg");
        SI.attr("src", "images/SI_red.svg");
        return

    }else if(sel==3){ //Onda de sierra
        oscillator.type = 'sawtooth';
        CU.attr("src", "images/CU_red.svg");
        TR.attr("src", "images/TR_red.svg");
        DS.attr("src", "images/DS_green.svg");
        SI.attr("src", "images/SI_red.svg");
        return

    }else if(sel==4){ //Onda sinusoidal
        oscillator.type = 'sine';
        CU.attr("src", "images/CU_red.svg");
        TR.attr("src", "images/TR_red.svg");
        DS.attr("src", "images/DS_red.svg");
        SI.attr("src", "images/SI_green.svg");
        return
    }
}
```

```

    }
};

function mut(){ //Funcion silencio
    if(mute==0){
        mute=1;
        document.getElementById("mute").innerHTML="SOUND";
    }else{
        mute=0;
        document.getElementById("mute").innerHTML="MUTE";
    }
};

function frq(){ //Para cambiar los límites de frecuencia
    minFreq=1*+document.getElementById("fmin").value;
    maxFreq=1*+document.getElementById("fmax").value;
};

//Inicializar el programa
$(function(){
    CU = $("#CU");
    TR = $("#TR");
    DS = $("#DS");
    SI = $("#SI");

    $("#fmin").val(20);
    $("#fmax").val(3000);
    minFreq = 20;
    maxFreq = 3000;
    mute=0;

});

```

4.4 Creación de la segunda versión de la aplicación

Cuando ya se ha finalizado la primera versión de la aplicación web, se empieza a crear la segunda versión, ya que se quiere cambiar el enfoque de la relación entre el movimiento de la mano izquierda y la distancia entre notas musicales, haciendo esta relación lineal. Para ello se cambia la ecuación que se emplea para la obtención de la frecuencia, por las ecuaciones 2, 3 y 4.

$$Octava = \frac{50 - Distancia}{12} \quad (2)$$

$$Nota = (50 - Distancia) - (Octava * 12) + 1 \quad (3)$$

$$Frecuencia = FrqRef * \sqrt[12]{2}^{(Octava+2-OctRef)*12+(Nota-NotRef)} \quad (4)$$

$$0 < Distancia < 50;$$

$$FrqRef = 440 ; OctRef = 4 ; NotRef = 10$$

Siendo en estas ecuaciones la variable “**octava**”, que octava va a sonar, si está el valor es 0 sonará la segunda octava, si en 1 sonara la tercera octava y así sucesivamente. La variable “**nota**” marca la nota que sonará si el valor es 1 sonara un do, si es 2 sonara un do#, si es 3 sonara un re y así sucesivamente. Las variables “**FrqRef**”, “**OctRef**” y “**NotRef**” son los valores de referencia que se toman para calcular las frecuencias de cada nota, en el caso de por defecto se toma de referencia la nota en la cuarta octava que suena a una frecuencia de 440 Hz. Para esto se sustituye la ecuación de la variable **oscillator.frequency.value** del código que se muestra en el apartado 4.2 y se introduce el siguiente código:

```
var oscillator, gainNode, mute, refFreq, refNot, refOct, ocxy, ntxy;

ocxy=Math.floor((50-Math.floor(min_dis/4))/12);
ntxy=(50-Math.floor(min_dis/4))-(ocxy*12)+1;
ocxy=ocxy+2;
oscillator.frequency.value =
refFreq*Math.pow(Math.pow(2,(1/12)),((ocxy-refOct)*12)+(ntxy-
refNot));
```

Después de hacer este cambio la funcionalidad de la frecuencias máxima y mínima queda anulada, ya que en la ecuación no existen unas variables que marquen la frecuencia máxima a alcanzar ni la frecuencia mínima. Por esta razón se va a cambiar esta funcionalidad por otra, la cual tratara de cambiar los valores de referencia que se toman para calcular la notas. Para esto hay que modificar un poco el código HTML para cambiar el número inputs de texto, además del título de estos inputs, después de esto se hace que cuando cambie el valor de estas cajas, se guarde el valor de estas en las variables que antes se han expuesto, que eran para tomar los valores de referencia y ya se ha acabado la segunda versión de la aplicación. Esta funcionalidad queda plasmada en el siguiente código:

```
function frq(){
    refFreq=1*+document.getElementById("fref").value;
    refNot=1*+document.getElementById("nref").value;
    refOct=1*+document.getElementById("oref").value;
};
```

Cuando estas estén terminadas se subirán a un servidor gratuito¹⁴ para que cualquier persona pueda utilizar la aplicación, si disponen del dispositivo Leap Motion.

Por ultimo como se han creado dos versiones y están las dos disponibles en internet, se crea un botón que contiene un hiperenlace de una versión a la otra, para que los usuarios tengan el acceso de una versión a la otra mucho más fácil.

¹⁴ El servicio que se ha utilizado en este proyecto es el servicio gratuito ofrecido por Neocities cuya página web es <https://neocities.org/>

Capítulo 5. Pliego de condiciones

5.1 Presupuesto

En este apartado se va a estimar el presupuesto que debería de haber costado el proyecto, contando todos los factores a tener en cuenta, entre ellos el dinero invertido en personal, el dinero pagado por los equipos que se utilizan para la creación del proyecto, además del dinero que se tendría que haber invertido en una oficina.

El personal que se ha encargado de la aplicación web se compone por un solo programador que ha estado alrededor de un mes y medio para hacer la aplicación, después de haber consultado cuánto cobra se ha llegado a la cifra de 1000€ al mes¹⁵ y teniendo en cuenta que este ha trabajado un mes y medio el coste de personal ascenderá a los 1500 €.

Para el proyecto hace falta un equipo informático sobre el que trabajar, al igual que utilizar software específico para trabajar. Los softwares con los que se ha trabajado principalmente son el editor de texto Notepad++ y los navegadores Google Chrome y Mozilla Firefox que son todos gratuitos. El equipo informático que se ha utilizado en el proyecto son el sensor Leap Motion, cuyo precio es de 90€ y un ordenador portátil con las siguientes características:

- Procesador: Intel Core i5-5200U
- Tarjeta gráfica: NVIDIA GeForce 820M
- Memoria RAM: 8GB DDR3
- Disco duro: 1000 GB HDD

Este ordenador portátil tiene un precio de 550€, que sumado al sensor hace que el coste del equipo informático ascienda a 640€.

Por ultimo queda por añadir al presupuesto el alquiler de la oficina, para reducir estos costes se va a utilizar la técnica de trabajo llamada coworking la cual se basa en compartir el espacio de trabajo entre varias empresas, como para este proyecto no se necesita mucho espacio, es la manera más económica que se ha encontrado para no inflar el presupuesto. El alquiler de una oficina con esta técnica de trabajo de un tamaño sobre los 11m² está sobre los 100€ al mes¹⁶ y como el proyecto va a tardar en programarse un mes y medio se pagarán dos meses lo que hace que el alquiler de la oficina aumente 200€ el presupuesto.

Después de haber calculado todos los costes el presupuesto del proyecto se ajustará a lo indicado en la tabla 1.

Coste de personal	1500€
Coste de equipo informático y de software	640€
Coste de alquiler de oficina	200€
Coste total	2340€

Tabla 1. Presupuesto del proyecto

¹⁵ El sueldo medio del programador de JavaScript al año es de 1408€ que si se divide en 12 pagas más 2 pagas extra queda un total aproximado de 1000€ cada paga, esto ha sido consultado el día 26 de agosto de 2016 en el siguiente enlace < <http://espana.jobtonic.es/salary/26526/83174.html?currency=EUR>>

¹⁶ Este precio se ha calculado después de comparar precios en la siguiente página web < <https://www.idealista.com/alquiler-oficinas/valencia-valencia/con-metros-cuadrados-menos-de-60/>>

Capítulo 6. Conclusiones y propuesta de trabajo futuro

6.1 Conclusiones

Después de probar el dispositivo Leap Motion en la aplicación web creada en el proyecto se ha llegado a las siguientes conclusiones.

El dispositivo contiene muchos fallos, primero el dispositivo no se puede utilizar si le da directamente la luz del sol, ya que el sol emite entre otras, ondas electromagnéticas con una longitud de onda de 850 nanómetros y esto afecta a la imagen que recibe el dispositivo, siendo imposible su tratamiento y así haciendo que no funcione correctamente. Otro fallo del dispositivo es por cómo se captan los datos, esto se hace mediante dos cámaras de infrarrojo, estas, si no se usan para la realidad virtual donde irían acopladas a las gafas, se sitúan en una superficie y captan el movimiento de las manos desde la parte baja de las manos, esto hace que los movimientos de los dedos que se encuentren en la parte superior, si la mano no se encuentra abierta en paralelo al plano de la superficie en la cual esta apoyado el dispositivo, este problema se hace muy visible con la aplicación, ya que la mano izquierda hace que la frecuencia del tono que suene dependa de la proximidad de cualquier parte de la mano con unas coordenadas, pues por este problema, si el usuario intenta controlar esto con el dedo índice, colocando la mano de manera que el dedo pulgar extendido forme parte de una línea perpendicular al plano formado por la superficie en la cual se encuentra el dispositivo, el dispositivo no detecta este movimiento del dedo índice. Además de esto en la parte software tiene fallos cuando introduces más de una mano en la zona de interacción ya que muchas veces confunde de que mano se trata, muchas veces cuando una persona pone las dos manos detecta dos manos izquierdas o dos manos derechas, en este apartado también los fallos se multiplican si el usuario trabaja en las partes laterales de la zona de interacción, en esa zona los datos del dispositivo contienen muchísimos errores y en cualquier aplicación el modelo de las manos hace cosas extrañas. Por ultimo en este apartado nombrar la limitación del Leap Motion a solo dispositivos Linux, Windows, e iOS, lo cual limita la utilización del Leap Motion a ordenadores, aunque el dispositivo Leap Motion este enfocado a la realidad virtual y los fabricantes y desarrolladores móviles estén desarrollando formas de realidad virtual utilizando principalmente el sistema operativo Android y la conexión se pudiera hacer a estos dispositivos gracias a que tienen un puerto nano USB o un puerto USB tipo C, de los cuales existen convertidores de USB 2.0 a estos tipos de USB.

A parte del dispositivo, otra conclusión que se saca de este trabajo es la facilidad de utilizar los lenguajes HTML, CSS y JavaScript, para alguien que ya ha programado en cualquier idioma de programación como C, Java, C# entre otros. Y la versatilidad de estos idiomas ya que creando una aplicación en una plataforma automáticamente se hace multiplataforma, esto no quiere decir que las páginas web estén optimizadas para cualquier plataforma, pero sí que se pueden utilizar en la mayoría de sistemas operativos. También añadir, la cantidad de recursos que existen sobre todo para JavaScript lo cual facilita mucho el trabajo de los desarrolladores que utilizan este idioma de programación para trabajar.

6.2 Trabajo futuro

El trabajo futuro trataría de mejorar la segunda versión de la aplicación, haciendo por ejemplo, que el sonido fuese continuo y no discontinuo como sucede en la segunda versión ya que solo suena los tonos de las notas musicales, además de esto también se le podría añadir a la aplicación, compatibilidad con otros dispositivos de reconocimiento de gestos como podría ser el dispositivo MYO. También se podría añadir que no solo sonaran tipos de onda si no que, con el reproductor MIDI, que contienen algunos de los navegadores, hacer que la aplicación reprodujera cualquier

instrumento musical, el cual fuera elegido por el usuario y controlado con el mismo mecanismo del theremin.

Otro camino a seguir en este trabajo, sería desarrollar la aplicación con algún motor gráfico, que permitiera programar la aplicación en JavaScript y que tuviera un motor de sonido potente para mejorar el sonido que produce la aplicación.

Capítulo 7. Manual del usuario

7.1 Instalación

Si es la primera vez que se va a utilizar en el ordenador el dispositivo Leap Motion, lo primero que se requiere es el software controlador del dispositivo, este se encuentra en la siguiente dirección web, <https://www.leapmotion.com/setup>. Después de haber instalado el software el ordenador ya reconocerá el Leap Motion, este funcionará correctamente. Si ya se ha utilizado el dispositivo Leap Motion con anterioridad en el ordenador que se va a utilizar no es necesario realizar este paso.

A continuación, lo único que tiene que hacer el usuario es abrir el navegador, preferiblemente Google Chrome o Mozilla Firefox e ir a la página web donde se encuentra la aplicación web. Si se quiere ir a la primera versión de la aplicación se tiene que visitar la dirección web <http://thereleap.neocities.org/>. Si se quiere visitar la segunda versión se tiene que visitar la siguiente página web <http://thereleap.neocities.org/v2/>.

Siguiendo estos pasos el usuario ya podrá disfrutar de la aplicación. Esta aplicación web está hecha para pantallas con relación 16/9 en disposición horizontal de ordenador a pantalla completa, si se utiliza en una ventana reducida se pide que se conserve la proporción del tamaño de la pantalla, si se utiliza en pantallas con disposición vertical o en una ventana que no conserve la proporción de una pantalla horizontal no funcionará correctamente.

7.2 Modo de empleo

Para empezar, se va a explicar el funcionamiento básico de la aplicación. Para que la aplicación empiece a funcionar, se tienen que encontrar las dos manos del usuario dentro de la zona de interacción del dispositivo Leap Motion y encontrarse el dibujo en la pantalla de la mano derecha, en la parte superior del dibujo de la antena con forma redonda. Cuando el usuario ya tiene las manos como se acaba de indicar, el usuario utilizará la mano derecha para manejar el volumen del sonido de la aplicación, como se puede ver en la siguiente imagen.



Figura 12. Cómo controlar el volumen de la aplicación

El usuario utilizará la mano izquierda para manejar el tono, esta mano maneja el tono dependiendo de su posición en el eje horizontal, como se puede observar en la siguiente imagen.

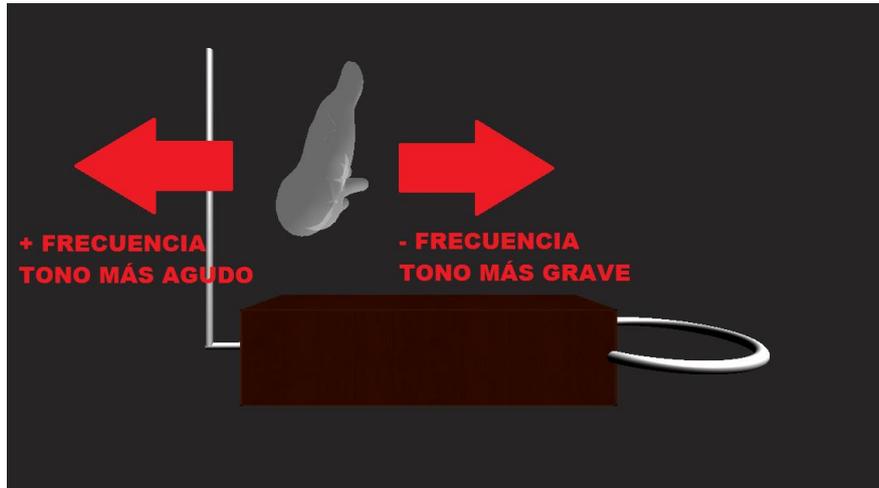


Figura 13. Cómo controlar el tono de la aplicación

7.2.1 Funcionalidades extra de las dos versiones

En las dos versiones de la aplicación existen funcionalidades extra que comparten ambas, la primera de ellas es elegir el tipo de forma de onda que quiere el usuario que suene. Esto el usuario lo puede seleccionar en el menú que hay en la parte derecha, en el cual se encuentran cuatro opciones correspondientes a las imágenes que contiene el menú, la primera opción es la que está seleccionada por defecto y hace que el sonido de la aplicación sea una onda cuadrada, la segunda opción selecciona el tipo de onda triangular, la tercera opción hace que el sonido sea una onda con diente de sierra y la cuarta y última opción hace que suene una onda sinusoidal. Para saber qué tipo de onda está seleccionada, la opción que está seleccionada se encuentra de color verde mientras que las otras son de color rojo.



Figura 14. Menú de selección de tipo de onda

La otra funcionalidad que comparten las dos versiones es el botón de silenciar el sonido, que se encuentra en el menú de la parte derecha de la pantalla, este tiene un texto en el que indica si está en modo silencio o si no lo está, si en el botón pone MUTE entonces no está activado el modo silencio, si en el botón pone SOUND entonces le modo silencio está activo.



Figura 15. Botón de silenciar cuando está activo y cuando no lo está

7.2.2 Funcionalidades extra de la primera versión

En la primera versión de la aplicación, se incluye la funcionalidad de elegir la frecuencia máxima del tono del sonido de la aplicación y la frecuencia mínima de esta. Para ello en el menú de la derecha de la pantalla, se encuentran unos cuadros de texto en los cuales, el usuario puede escribir la frecuencia en hercios que se quiere como límite superior e inferior, por defecto estas frecuencias son 3000 Hz y 20 Hz respectivamente.

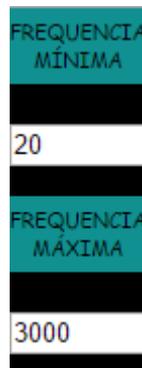


Figura 16. Cuadros de texto para elegir los límites de la frecuencia

Otra funcionalidad que contiene la primera versión de la aplicación, es que muestra la frecuencia del sonido que suena en la aplicación, este apartado se encuentra justo debajo de los cuadros de texto, en los cuales eliges las frecuencias limites, en este apartado cuando no suena nada pone ### y si suena algún sonido pone la frecuencia a la que suena el sonido en hercios.

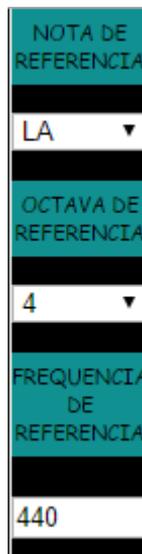


Figura 17. Cuadro en el que se muestra la frecuencia del sonido que suena

Por último, se encuentra un botón en el menú del lado derecho de la imagen para ir a la segunda versión de la aplicación, en este pone IR V2.

7.2.3 Funcionalidades extra de la segunda versión

La segunda versión de la aplicación tiene la funcionalidad de poder elegir la nota de referencia para poder calcular las frecuencias de las demás notas, para ello se encuentran dos cuadros de selección en los cuales el usuario puede seleccionar la nota y la octava que se va a tomar de referencia y un cuadro de texto en el que el usuario puede seleccionar a que frecuencia se encuentra la nota de referencia, por defecto la nota de referencia es un LA de la cuarta octava y a una frecuencia de 440 hercios.



The image shows a vertical menu with three sections. The first section is titled 'NOTA DE REFERENCIA' and has a dropdown menu with 'LA' selected. The second section is titled 'OCTAVA DE REFERENCIA' and has a dropdown menu with '4' selected. The third section is titled 'FRECUENCIA DE REFERENCIA' and has a text input field containing '440'.

Figura 18. Menú para elegir la nota de referencia

Otra funcionalidad es que en este caso en vez de mostrarse la frecuencia que suena, como sucede en la primera versión, se muestra la nota que está sonando en ese momento en un cuadro. Por ultimo también hay un botón por si el usuario quiere volver a la primera versión en el cual pone IR V1.

Capítulo 8. Bibliografía

[1] Ingeniatic, "Theremin"

<http://ingeniatic.euitt.upm.es/index.php/tecnologias/item/631-theremin> [Online]

[2] Estrada, V. "El theremin: Compositores destacados"

<http://www.victorestrada.com/2011/11/22/el-theremin-compositores-destacados/> [Online]

[3] Estrada, V. "El theremin: Fabricantes"

<http://www.victorestrada.com/2011/11/21/el-theremin-fabricantes/> [Online]

[4] Mitra, S.; Acharya, T. "Gesture Recognition: A Survey", *IEEE Transactions on Systems, Man, and Cybernetics-Part C: Applications and Reviews*, vol. 55, no. 3, pp. 311-324, May 2007.

[5] Thalmic Labs Inc. "Myo Gesture Control Armband | Wearable Technology by Thalmic Labs"

<https://www.myo.com/> [Online]

[6] Thalmic Labs Inc. "Myo armband Product Specs"

<https://support.getmyo.com/hc/en-us/articles/202648103-Myo-aramband-Product-Specs> [Online]

[7] Colgan, A. "How Does the Leap Motion Controller Work?"

<http://blog.leapmotion.com/hardware-to-software-how-does-the-leap-motion-controller-work/> [Online]

[8] Leap Motion Inc. "API Overview"

https://developer.leapmotion.com/documentation/v2/javascript/devguide/Leap_Overview.html [Online]

[9] Leap Motion Inc "Javascript | Leap Motion Developers"

<https://developer.leapmotion.com/javascript#plugins> [Online]

[10] Ehrlich, P. "LeapJS Plugins"

<https://github.com/leapmotion/leapjs-plugins> [Online]

[11] Ehrlich, P. "JS Rigged Hand Plugin"

<https://github.com/leapmotion/leapjs-rigged-hand> [Online]

Capítulo 9. Glosario

API: *Application Programming Interface*, Conjunto de subrutinas, funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

BSO: *Banda Sonora Original*, Música utilizada en una película.

CSS: *Cascading Style Sheets*, Lenguaje usado para definir y crear la presentación de un documento estructurado escrito en HTML o XML.

DIY: *Do It Yourself*, Objetos hechos por el usuario, manualidad.

DOM: *Document Object Model*, Interfaz de plataforma que proporciona un conjunto estándar de objetos para representar documentos HTML, XHTML y XML.

EMG: *Electromiografía*, Técnica de registro gráfico de la actividad eléctrica producida por los músculos esqueléticos.

Hardware: Conjunto de elementos físicos o materiales que constituyen una computadora o un sistema informático.

HTML: *HyperText Markup Language*, Lenguaje de marcado para la elaboración de páginas web.

IMU: *Unidad de Medición Inercial*, dispositivo electrónico que mide e informa acerca de la velocidad, orientación y fuerzas gravitacionales de un aparato, usando una combinación de acelerómetros y giróscopos.

jQuery: Biblioteca de JavaScript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web.

Métodos: Subrutina cuyo código es definido en una clase.

MIDI: *Musical Instrument Digital Interface*, estándar tecnológico que describe un protocolo, una interfaz digital y conectores que permiten que varios instrumentos musicales electrónicos, computadoras y otros dispositivos relacionados se conecten y comuniquen entre sí.

MIT: *Massachusetts Institute of Technology*, Universidad privada localizada en Cambridge, Massachusetts (Estados Unidos)

Plugin: Aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy específica. Esta aplicación adicional es ejecutada por la aplicación principal por medio de la API.

Software:

SVG: *Scalable Vector Graphics*, Especificación para describir gráficos vectoriales bidimensionales tanto estáticos como animados, en formato XML.

W3C: *World Wide Web Consortium*, Comunidad internacional que desarrolla estándares que aseguran el crecimiento de la Web a largo plazo.

Wikipedia: Enciclopedia libre, poliglota y editada colaborativamente y es administrada por la Fundación Wikipedia.