



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA

**TELECOM** ESCUELA  
TÉCNICA **VLC** SUPERIOR  
DE **UPV** INGENIEROS  
DE TELECOMUNICACIÓN

## **SOUNDCOOL: OPEN SOUND CONTROL, SMARTPHONES, TABLETS Y OTROS INTERFACES PARA LA CREACIÓN MUSICAL Y VISUAL**

**Saúl Moncho Moncho**

**Tutor: Jorge Sastre Martínez**

**Cotutor: Jaime Serquera Peyró**

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 2017-18

Valencia, 11 de septiembre de 2018



## Resumen

*Soundcool* es un proyecto innovador pensado para la educación musical y la creación colaborativa por medio de móviles, tabletas y la herramienta Kinect de la Xbox 360. El sistema completo está formado por dos partes, un ordenador donde se ejecutan los módulos y el dispositivo móvil que los controla remotamente. Ambos elementos se conectan por una red Wifi utilizando el protocolo Open Sound Control. Actualmente se está desarrollando la parte visual del sistema con la creación de 5 módulos nuevos. A lo largo de este trabajo se explicará el diseño de las 5 superficies de control de los módulos para móviles y tabletas. El objetivo es realizar una aplicación con un menú principal para la conexión que envíe las instrucciones de los objetos táctiles en tiempo real. Se ha utilizado una metodología de desarrollo ágil que divide el proyecto en iteraciones y establece la duración de cada una. En conclusión, se ha realizado una programación compleja y la aplicación para móviles y tabletas está lista para utilizarse en centros de enseñanza.

## Resum

*Soundcool* és un projecte innovador pensat per a l'educació musical y la creació col·laborativa per mitjà de mòbils, tabletas i la ferramenta Kinect de l'Xbox 360. El sistema al complet està format per dues parts, un ordinador on s'executen els mòduls i el dispositiu mòbil que els controla de manera remota. Ambdós elements es connecten per una xarxa Wifi utilitzant el protocol Open Sound Control. Actualment s'està desenvolupant la part visual del sistema amb la creació de 5 mòduls nous. Al llarg d'aquest treball s'explicarà el disseny de les 5 superfícies de control dels mòduls per a mòbils i tabletas. L'objectiu es realitzar una aplicació amb un menú principal dedicat a la connexió que envie les instruccions dels objectes tàctils en temps real. S'ha utilitzat una metodologia de desenvolupament àgil que divideix el projecte en iteracions y estableix la duració de cada un d'ells. En conclusió, s'ha realitzat una programació complexa i l'aplicació per a mòbils i tabletas està llesta per utilitzar-se als centres d'ensenyança.

## Abstract

*Soundcool* is an innovative project made for musical education and collaborative creation by usage of smartphones, tablets and Kinect of the Xbox 360. The complete system is made of two parts, one part is a computer where is execute the modules and other one is the mobile dispositive that controls remotely. Both elements are connected to a Wifi network using the protocol Open Sound Control. The visual part of the system is now being programmed, with the creation of 5 news modules for smartphones and tablets. The objective is to create an application with a menu for the connection that sending the instructions of the remote objects in real time. The project uses a methodology that divides the work in iterations and establish the duration of each one. In conclusion, the project uses a complex programation and the application for smartphones and tablets is ready to be used in teaching centres.



## **PALABRAS CLAVE**

Soundcool

Dispositivos móviles

Superficies de control

Comunicación OSC

Creación colaborativa

## **PARAULES CLAU**

Soundcool

Dispositius mòbils

Superficies de control

Protocol OSC

Creació col·laborativa

## **KEY WORDS**

Soundcool

Mobile dispositives

Control Surface

OSC Protocol

Collaborative Creation

# ÍNDICE

ÍNDICE DE IMÁGENES.....	3
ÍNDICE DE TABLAS .....	5
Capítulo 1. Introducción.....	6
1.1 Motivación .....	6
1.2 Justificación del trabajo.....	6
1.3 Objetivos .....	7
Capítulo 2. Contexto .....	8
2.1 Control Surface .....	8
2.2 Soundcool.....	9
Capítulo 3. Metodología.....	12
3.1 Gestión del proyecto.....	12
3.2 Distribución del TFG en tareas. ....	12
3.3 Distribución del proyecto en iteraciones.....	13
3.4 Diagrama temporal.....	14
Capítulo 4. Tecnologías empleadas en el desarrollo del proyecto .....	18
4.1 Software .....	18
4.1.1 MAX/MSP/JITTER .....	18
4.1.2 Unity3D.....	19
4.1.3 Componentes de Unity3D utilizados en el proyecto y características.....	21
4.1.4 Visual Studio Basic .....	24
4.1.5 Protocolo OSC.....	24
4.1.6 UniOSC .....	25
4.2 Hardware .....	27
Capítulo 5. Desarrollo de las superficies de control de los módulos.....	29
5.1 Switcher.....	29
5.2 ImageSampler .....	32
5.3 Sampler .....	36
5.4 Color.....	40
5.5 Blending .....	43
5.6 Panel conexión .....	46
Capítulo 6. Diseño de la aplicación.....	51
6.1 Exportar la aplicación desde Unity3D para Android .....	51

6.2 Versión 1 .....	53
6.3 Versión 2 .....	54
Capítulo 7. Comprobación del funcionamiento de la herramienta .....	56
Capítulo 8. Conclusiones.....	59
8.1 Generales.....	59
8.2 Dificultades.....	59
8.3 Competencias transversales aplicadas con la realización del TFG .....	60
8.4 Presupuesto .....	61
Capítulo 9. Trabajo futuro.....	62
9.1 Efectos Blending .....	62
9.2 Painter .....	63
9.3 Módulos de entrada de video y audio.....	63
9.4 Creación de tutoriales para los cursos de <i>Soundcool</i> .....	64
9.5 Control de la iluminación con DMX.....	65
Capítulo 10. Referencias .....	66

## ÍNDICE DE IMÁGENES

Ilustración 1: <i>Soundcool</i> en las aulas.....	9
Ilustración 2: Evento en la plaza de toros de Alicante .....	10
Ilustración 3: Diagrama de Gantt inicial .....	15
Ilustración 4: Diagrama de Gantt final .....	17
Ilustración 5: Ventana principal de MAX.....	18
Ilustración 6: Ejemplo de un patcher o programa gráfico de MAX.....	19
Ilustración 7: Ventana principal de Unity3D .....	20
Ilustración 8: Objetos del elemento “Text” .....	21
Ilustración 9: Objetos del elemento “Image” .....	22
Ilustración 10: Objetos del elemento “Button” .....	22
Ilustración 11: Objetos del elemento “Slider” .....	23
Ilustración 12: Objetos del elemento “Toggle” .....	23
Ilustración 13: Ventana principal del Visual Studio.....	24
Ilustración 14: Partes del mensaje OSC.....	25
Ilustración 15: Script Uni OSC Connection .....	26
Ilustración 16: Script Uni OSC Event Dispatcher Button .....	26
Ilustración 17: Script Uni OSC Event Dispatcher Slider .....	27
Ilustración 18: Script Uni OSC Event Dispatcher Toggle.....	27
Ilustración 19: Dispositivos electrónicos utilizados durante el proyecto.....	28
Ilustración 20: Módulo Switcher .....	30
Ilustración 21: Superficie de control del módulo Switcher .....	30
Ilustración 22: Algoritmo del módulo Switcher .....	31
Ilustración 23: Switcher a) botón en el estado “normal” b) botón en el estado “pressed” .....	31
Ilustración 24: Event Trigger del módulo Switcher .....	32
Ilustración 25: Módulo ImageSampler .....	33
Ilustración 26: Superficie de control del módulo ImageSampler .....	33
Ilustración 27: Algoritmo de las teclas del piano del módulo ImageSampler .....	34
Ilustración 28: Algoritmo del texto de las teclas del módulo ImageSampler .....	34
Ilustración 29: Construcción del piano del módulo ImageSampler .....	35
Ilustración 30: Slider del módulo ImageSampler .....	35
Ilustración 31: ImageSampler a) toggle desactivado b) toggle activado .....	35
Ilustración 32: Módulo Sampler a) Modo Frame b) Modo Loop .....	37
Ilustración 33: Superficie de control del Sampler modo Frame.....	38
Ilustración 34: Superficie de control del Sampler modo Loop.....	38
Ilustración 35: Algoritmo del módulo Sampler .....	39
Ilustración 36: Sampler a) botón en el estado “normal” b) botón en el estado “pressed” .....	39
Ilustración 37: Sampler a) toggle desactivo b) toggle activo .....	40
Ilustración 38: Módulo Color.....	41
Ilustración 39: Superficie de control del módulo Color .....	41
Ilustración 40: Algoritmos del módulo Color .....	42
Ilustración 41: Slider del módulo Color.....	42
Ilustración 42: Color a) botón en el estado “normal” b) botón en el estado “pressed” .....	42
Ilustración 43: Función del slider del módulo Color .....	43



Ilustración 44: Módulo Blending .....	43
Ilustración 45: Superficie de control del módulo Blending .....	44
Ilustración 46: Slider del módulo Bleding .....	44
Ilustración 47: Blending a) botón en el estado “normal” b) botón en el estado “pressed” .....	45
Ilustración 48: Panel del menú conexión .....	46
Ilustración 49: Algoritmo del menú conexión .....	47
Ilustración 50: Algoritmo para mostrar el texto de los mensajes OSC del panel conexión .....	48
Ilustración 51: Cajas de texto de los elementos de conexión .....	49
Ilustración 52: Toggle activador de la visualización de los mensajes OSC .....	49
Ilustración 53: Vista de los mensajes OSC en la aplicación .....	49
Ilustración 54: Jerarquía del panel de conexión .....	50
Ilustración 55: Opción exportar escenas de Unity3D .....	51
Ilustración 56: Opciones de la aplicación en Unity3D .....	52
Ilustración 57: Menú principal de la aplicación versión 1 .....	53
Ilustración 58: Jerarquía de la aplicación versión 1 .....	53
Ilustración 59: Varios diseños de iconos para la aplicación a) versión 1 b) versión 2 c) versión 3 .....	54
Ilustración 60: Menú principal de la aplicación versión 2 .....	54
Ilustración 61: Barra superior menú Audio .....	54
Ilustración 62: Barra superior menú Video .....	54
Ilustración 63: Jerarquía de la aplicación versión 2 .....	55
Ilustración 64: Launcher de Soundcool para el ordenador .....	56
Ilustración 65: Módulos de video de Soundcool .....	57
Ilustración 66: Aplicación conectada al ordenador .....	57
Ilustración 67: Sistema completo en funcionamiento .....	58
Ilustración 68: Módulo Blending con todas las opciones .....	62
Ilustración 69: Superficie de control del Blending actualizada .....	63
Ilustración 70: Boceto superficie de control del Painter .....	63
Ilustración 71: Boceto de la superficie de control del módulo cámara .....	64
Ilustración 72: Boceto de la superficie de control del módulo micrófono .....	64
Ilustración 73: Boceto de la superficie de control del módulo DMX .....	65

## ÍNDICE DE TABLAS

Tabla 1: Mensajes OSC del módulo Switcher .....	32
Tabla 2: Mensajes OSC del módulo ImageSampler.....	36
Tabla 3: Mensajes OSC del módulo Sampler.....	40
Tabla 4: Mensajes OSC del módulo Color .....	43
Tabla 5: Mensajes OSC del módulo Blending.....	45
Tabla 6: Estados del botón connect .....	48
Tabla 7: Presupuesto.....	61

## Capítulo 1. Introducción

Este primer capítulo será la introducción al proyecto y se expondrán las influencias que ha habido para llevarlo a cabo. También se presentaran los objetivos y la motivación que hay detrás de la realización del TFG.

### 1.1 Motivación

Son dos las principales razones por las que se elige este trabajo, la oportunidad de formar parte del equipo de *Soundcool* [1] y el perfil técnico y artístico del profesor. En las asignaturas del grado que hemos coincidido ha logrado transmitir la pasión y la ambición de crear arte con esta herramienta. El tutor es ingeniero y músico y me siento personalmente identificado con él.

Por un lado tenemos la música y las instalaciones audiovisuales. Siendo músico con Grado Elemental de trompeta/fiscorno y formando parte de varias bandas, tengo un amplio conocimiento musical. También he estado como técnico de sonido en un grupo de pop para enriquecer mis conocimientos de mezcla y sonido en directo. Este periodo como técnico me ha ayudado a adquirir experiencia sobre la utilización de superficies de control y con el DAW (Digital Audio Workstation) Cubase [2].

Por otro lado tenemos la parte técnica. Como futuro graduado en ingeniería de telecomunicaciones, con este proyecto puedo aplicar y expandir los conocimientos desarrollados durante los años de carrera. Se ha realizado una aplicación nueva para smartphone sin conocimientos previos de programación en Android.

Finalmente, la posibilidad de poder trabajar con un equipo en un proyecto real como se realiza en una empresa y poder aportar mi opinión. Es decir, formar parte de un grupo y dedicar el tiempo a trabajar en una aplicación para la educación musical que utilizará mucha gente en varios países. También tener en cuenta los comentarios de los usuarios y actualizar la herramienta con nuevas funciones.

### 1.2 Justificación del trabajo

El sistema *Soundcool* se implementa como mejora de la enseñanza musical tradicional utilizando las nuevas tecnologías audiovisuales emergentes. Las herramientas utilizadas para motivar y despertar interés de los estudiantes son dispositivos móviles como tabletas y smartphones. Todo ello sumado a los sistemas de creación colaborativa facilita enormemente el proceso de aprendizaje.

*Soundcool* tiene desarrollados y en funcionamiento los módulos y la aplicación para dispositivos móviles de la parte musical. Desde la organización se desea potenciar también la parte visual y se han creado 5 nuevos módulos. La realización de este trabajo surge por la necesidad de proporcionar una aplicación con superficies táctiles que pueda controlar de manera remota los módulos de imagen.

El sistema completo consta de dos partes principales: un ordenador Mac o PC donde se ejecuta un launcher y un dispositivo móvil (superficies de control), que se conectan mediante una red Wi-Fi. El software utilizado en la creación del Launcher del ordenador ha sido MAX/MSP/JITTER [3]. Se trata de un software creado para músicos e ingenieros audiovisuales basado en programación gráfica y modular. Existen también herramientas gratuitas con características funcionales parecidas como es PureData [4].

Por otro lado, el programa utilizado en el desarrollo de la aplicación móvil es el Unity3D [5]. Se trata de una herramienta creada para la realización de juegos tanto 2D como 3D. Este sistema tiene una versión básica gratuita siempre y cuando el usuario no genere beneficios con la APP creada y otras condiciones a cumplir según la licencia. También existen programas con funcionalidades parecidas como Android Studio [6], Eclipse [7], o TouchOSC[8]; con los que se podría generar una aplicación similar.

### 1.3 Objetivos

El objetivo principal de este trabajo es diseñar una aplicación para dispositivos móviles con superficies de control de los nuevos módulos de la parte visual. El diseño se realizará respetando las características tanto de los módulos creados como el sistema ya existente de *Soundcool*.

Los objetivos secundarios que se relacionan con la creación de la aplicación son:

- Un periodo de formación relacionada con las herramientas y conceptos que se van a utilizar en la creación de la herramienta. Obtener conocimientos del software MAX y Unity3D realizando algunos ejemplos básicos supervisados por los profesores del grupo de investigación.
- Comprensión del funcionamiento de todos los módulos generados por los compañeros. Esta tarea consiste en presentar las características de cada módulo y planificar un calendario con las tareas y las fechas de finalización.
- Diseño e implementación de código de las superficies de control. Inicialmente un trabajo de diseño individual imitando las características visuales de cada módulo utilizadas en las superficies de control de los módulos ya existentes en *Soundcool* y la configuración de los mensajes OSC.
- Generar una aplicación que contenga todas las superficies programadas. Se implementará una barra superior de menú para navegar entre todas las opciones.
- Comprobación del funcionamiento de la aplicación programada con la ayuda de compañeros, profesores y usuarios.

## Capítulo 2. Contexto

En este capítulo se describe el estado actual de las superficies de control y su evolución en los últimos años. También se describirá el proyecto *Soundcool* incluyendo su historia así como el funcionamiento y financiación.

### 2.1 Control Surface

Una superficie de control es un dispositivo electrónico asociado a un DAW (Digital Audio Workstation) y envía instrucciones de control de manera remota en tiempo real [9]. Está compuesta por elementos táctiles como faders, knobs, botones y/o pads. La colocación de los elementos en el dispositivo está relacionada con el programa que controla. Cuando el usuario acciona un objeto se envía una instrucción de control con su valor al ordenador donde se ejecuta el software.

La idea de crear este tipo de dispositivos surgió en los estudios de grabación para hacer más eficiente la producción musical. Inicialmente se realizaba todo con un ordenador, un teclado de letras y un ratón. El especialista tenía que seleccionar uno a uno los objetos mediante el ratón y esto limitaba los movimientos y la fluidez. Como solución a este problema surgieron las primeras superficies de control, que contaban con todas las principales instrucciones del programa dispuestas en elementos táctiles.

Históricamente las superficies de control se han actualizado frecuentemente influenciadas por la aparición de las nuevas tecnologías. Inicialmente, era un dispositivo electrónico fabricado por las compañías creadoras de DAWs con precios muy elevados. Este tipo de elementos solo se los podían permitir los mejores estudios con grandes presupuestos. Con el tiempo, han evolucionado los diseños reduciéndose su tamaño y sus opciones de control. Actualmente, se pueden encontrar por menos de 100€ tablas de control de las empresas AKAI [10] y NOVATION [11]. También se sigue produciendo equipo profesional orientado a un software concreto como la empresa Avid con Pro Tools [12], que desarrolla dispositivos muy avanzados con precios que rondan los 5.000€. Sin ir más lejos, las consolas utilizadas por los DJs son superficies de control con elementos táctiles adaptados para que ellos realicen sus mezclas.

Por el contrario, las empresas que fabrican mesas de mezclas digitales también están integrando sistemas de control remoto [13] de manera que los técnicos de sonido pueden controlar con una tableta todas las opciones. Esto está generando que el cliente opte por el mercado de las mesas de mezcla y no por los dispositivos de control. La razón es que en una mesa de mezcla tienes todo el estudio integrado (grabación, mezcla, efectos...) mientras que una superficie de control es un complemento del software.

Por otro lado, con la aparición de los smartphones y tabletas táctiles y sus grandes pantallas, el mercado de las superficies de control se ha visto ampliado. Existen empresas específicas en el desarrollo de aplicaciones para el móvil donde el usuario puede crear y cargar sus propias superficies. El funcionamiento es el mismo que los dispositivos electrónicos, sólo que con este tipo de aplicaciones se ahorra mucho dinero y se genera un proyecto propio. Una de las compañías que se dedica a la producción de estas herramientas es TouchOSC.

Las principales ventajas de la utilización de las superficies de control son:

- Posibilidad de tener todos los controles del programa de manera táctil.
- Poder controlar varios elementos a la vez con respuesta inmediata.

- Mayor precisión sobre las pistas de audio con los elementos de tipo “fader”.

Las desventajas:

- Un precio muy elevado para las superficies de control hardware.
- Muchos de estos dispositivos hardware llegan a tener un gran tamaño.
- No se pueden generar archivos de audio o imagen desde la superficie de control, los archivos se generan en el propio ordenador.

## 2.2 Soundcool

*Soundcool* es un sistema innovador y gratuito de creación colaborativa mediante ordenadores, dispositivos móviles y la Kinect de Xbox 360 [14]. Consiste en un conjunto de módulos relacionados con el audio y el sonido como players, instrumentos virtuales, efectos, mezcladores, etc; que funcionan en un ordenador y pueden conectarse entre sí. Estos módulos se pueden controlar de manera remota con tabletas y smartphones android e iOS conectando todos los dispositivos a la misma red WIFI y/o con el sistema de cámaras USB Kinect. Pero en los móviles y tablets no se reproducirá ningún sonido, son superficies de control y únicamente envían instrucciones al ordenador.



**Ilustración 1: *Soundcool* en las aulas**

*Soundcool* surge de la necesidad de unir dos líneas como la ingeniería y la música, siendo la idea fundamental del proyecto la creación sonora y audiovisual colaborativa. A parte de ser una aplicación gratuita, en su página web se pueden encontrar multitud de recursos como tutoriales, instrumentos virtuales, librerías de sonidos, un curso online de introducción al sistema [15], etc. Funciona mediante el protocolo Open Sound Control (OSC) [16], el cuál conecta la interfaz del ordenador y la superficie de control que se ejecuta en el smartphone o/y la Tablet. El sistema funciona en ordenadores Mac o PC permitiendo realizar creaciones colaborativas, ya que los participantes pueden controlar diferentes módulos del ordenador. Así se han llegado a realizar creaciones hasta con 100 participantes.

La historia del grupo de investigación tiene su origen en 2011 con la creación del título EU en Música Electrónica, Electroacústica y Video Creación de la UPV [17]. En 2012 surgió la idea de crear una herramienta para la educación musical y la creación sonora utilizando las nuevas tecnologías. En 2013 el proyecto “Nuevas Tecnologías Audiovisuales e Interfaces para la Educación Musical y Creación Sonora” dirigido por Jorge Sastre fue financiado por la UPV y supuso el nacimiento de *Soundcool*. En octubre de ese mismo año se empezó a probar el sistema en el Instituto Arabista Ribera de Carcaixent (Valencia). Desde entonces el proyecto no ha parado de crecer, recibiendo premios como NEM Art, dos premios Bankia al talento musical, el premio SIMO a la mejor experiencia en programación y robótica, etc.

A partir de 2015 la enseñanza musical con esta herramienta se extendió a través del proyecto Erasmus+ a países europeos como Italia, Portugal y Rumanía, con excelentes resultados por parte de profesores y alumnos. En España también se extendió su utilización a escuelas de la Federación de Sociedades Musicales de la Comunidad Valenciana (FSMCV). El 2016 se realizó el estreno de la ópera “La Mare dels Peixos”, composición propia del equipo de *Soundcool*, en el Palau de les Arts (Valencia). También se realizó un preestreno de la misma (el Acto I) en el Colegiul de Are Baia Mare (Rumanía). Ya en 2017 se realizó el preestreno de la obra (Acto I) en el Instituto Tecnológico de Estudios Superiores de Monterrey. Pero el evento más importante de la historia de *Soundcool* fue el pasado 12 de mayo en Alicante, donde se batió un récord Guinness al impartir la mayor lección de música hasta el momento titulada “Introducción a Soundcool” [18]. Participaron 100 “soundcoolers” junto a 4130 estudiantes de las escuelas de música de la FSMCV tocando sus instrumentos acústicos.



**Ilustración 2: Evento en la plaza de toros de Alicante**

El sistema fue creado en la Universidad Politécnica de Valencia y está financiado por:

- Fundación Daniel y Nina Carasso (16-AC-2016) en el desarrollo de sistemas de VideoArte.

- Cátedra Telefónica de la UPV en el desarrollo de un demostrador tecnológico de *Soundcool* ayudando en la adquisición de dispositivos móviles Android y iOS.
- Generalitat Valenciana en el proyecto “Soundcool: nuevas tecnologías para la educación musical y la creación sonora” (AICO/2015/120) liderado por Jorge Sastre.
- Ministerio de Educación, Cultura y Deporte beca Salvador de Madariaga para estancias de profesores e investigadores séniores en centros extranjeros PRX12/00557.
- Proyecto de la Universidad Politécnica de Valencia PAID-05-12-SP20120470 “Soundcool: nuevas tecnologías e interfaces para la educación musical y la creación sonora”.

Soundcool ha sido adoptado en dos proyectos Erasmus+ para su uso en varios países de Europa:

- Comisión Europea Proyecto Erasmus+ “Tecnología al servicio del aprendizaje y la creatividad: tejiendo redes europeas a través de la creación musical colaborativa” con países involucrados Italia, Rumanía, Portugal y España (2015-1-ES01-KA201-016139).
- Music & Technology con países involucrados Italia, España, Rumanía, Francia, Turquía, Malta, Croacia y Georgia (2017-2-IT03-KA105-011802).

El director Jorge Sastre cuenta, para el desarrollo del proyecto, con un equipo de investigadores de perfiles multidisciplinares como músicos, ingenieros, pedagogos, expertos en comunicación audiovisual y licenciados en Bellas Artes. Por el equipo también han pasado muchos colaboradores, alumnos de ingeniería y grado de Telecomunicación, alumnos de doctorado y máster, etc., y profesores que han proporcionado nuevos puntos de vista y conocimientos para mejorar la herramienta.

En un primer momento, todo el proyecto está basado en módulos referentes a efectos de audio como módulos para la captación, grabación y reproducción de audio, lanzadores de loops, modificadores de la envolvente, filtros, plugins VST, mesas de mezclas, moduladores FM y AM, etc. La programación de la primera aplicación de control para móviles y tabletas se realizó con TouchOSC. TouchOSC tenía dos inconvenientes: por una parte no es gratuito, suponiendo un cierto coste por dispositivo a utilizar, y por otra no era todo lo versátil que se requería, no permitiendo algunas funcionalidades que se precisaban. Así, finalmente se optó por realizar una app propia en Unity3D, añadiendo mayor versatilidad al diseño, y permitiendo con un solo desarrollo cubrir de forma sencilla varias plataformas, como Android e iOS.

Actualmente se está expandiendo *Soundcool* al mundo visual creando módulos para imagen y vídeo, siendo la referencia para la realización de este TFG. La idea es programar nuevos módulos con funcionalidades similares a las de audio. Para ello cuenta con módulos de reproducción de vídeo, de presentación imágenes, mezcladores de vídeos e imágenes, efectos sobre imágenes y vídeo grabados o en directo, etc.



## Capítulo 3. Metodología

En este capítulo se describe como se ha gestionado el proyecto utilizando una metodología de desarrollo ágil llamada Scrum[19-21], que es muy utilizada en los proyectos relacionados con la programación. Este método basado en procesos iterativos se desarrolla partiendo de una idea general del producto, numerando las características que el propietario quiere, y dividiendo la producción en iteraciones llamadas “sprints” para tener un mejor control del tiempo y un seguimiento de los resultados.

### 3.1 Gestión del proyecto

La idea general del proyecto es lanzar una aplicación para dispositivos móviles que controle los módulos que se ejecutan en el ordenador mediante una conexión Wireless. La aplicación debe poder controlar dos partes, el bloque de audio que ya está en funcionamiento y el bloque nuevo de video. El diseño tendrá un menú principal desde el cual se introduzcan los parámetros de conexión y se pueda acceder a cada bloque.

En el proyecto, aparte del equipo de trabajo formado por alumnos y profesores de la UPV, tenemos dos figuras muy importantes. Por un lado Jorge Sastre, director del proyecto *Soundcool*, se encarga de tomar las decisiones finales sobre las funcionalidades de todo el sistema completo, obtener financiación y decidir el día del lanzamiento. Por otro lado el Scrum Master es Jaime Serquera, responsable del control y seguimiento de mi proyecto garantiza que se implementan todas funcionalidades en la fecha establecida. También he recibido ayuda del profesor Adolfo Muñoz, creador de la aplicación para móviles *Soundcool* Audio, quien me ha explicado con detalle las características del diseño y del código para realizar este proyecto.

Durante la producción se han realizado varias reuniones de planificación, actualización y revisión. Las reuniones de planificación servían para plantear una idea y entre todos los miembros elaborar el diseño, por eso estas reuniones duraban dos o más días para poder resolver todas las dudas que iban surgiendo. Las reuniones de actualización eran breves y en ellas se presentaba el estado del proyecto, se fijaba fecha de presentación y se añadían las mejoras si era el caso. Finalmente, la revisión consistía en presentar el proyecto terminado a todos los miembros para realizar un testeo; de estas reuniones se solían hacer 2 en un plazo de una semana para tener margen de corrección de errores y lanzar la aplicación final.

La herramienta utilizada en la gestión de las tareas y su duración ha sido el diagrama de Gantt [22], se trata de un gráfico con el orden de la sucesión de las tareas y su fecha de inicio y fin. Por otro lado, para el intercambio de los archivos entre todos los miembros del grupo de trabajo se ha utilizado el correo de la UPV junto con la plataforma intercambios UPV. También se ha utilizado el cloud de Unity3D [23] para enviar los proyectos a Adolfo, se trata de un sistema de almacenamiento en la nube. Y por último, se ha utilizado la aplicación móvil del calendario para tener en cuenta las fechas de entrega y que se envíen notificaciones.

### 3.2 Distribución del TFG en tareas.

En la primera reunión con el tutor se debaten las características del TFG y se divide en 4 tareas principales: estudio del software para el desarrollo de la herramienta, presentar la idea general del diseño, desarrollo de los módulos por separado y generar una aplicación android que integre todos los módulos en un mismo proyecto.

#### *TAREA 1. Formación básica del software.*

En primer lugar se estudiará el entorno MAX, sus características y sobre todo la programación de los módulos creados por los compañeros de la organización para comprender el funcionamiento.

Estuve un periodo de 4 meses estudiando y realizando diferentes ejemplos, con la ayuda de tutoriales, referentes a efectos de audio y a la conexión OSC.

En segundo lugar se empezó con Unity3D a hacer unos ejemplos base que vienen guiados en tutoriales en su página oficial y que fueron supervisados por el Scrum Máster. Una vez familiarizado con el entorno de trabajo se estudiará la aplicación de móviles y tabletas de *Soundcool* ya desarrollada con el profesor que la creó. Se revisará la programación, los objetos utilizados en el diseño y como se vinculan unos con otros; pero sobre todo los parámetros de conexión.

### **TAREA 2. Estudiar la idea general del proyecto.**

En este apartado se pretende obtener la idea general para la nueva aplicación y decidirla por consenso entre todos los miembros del equipo en una reunión. El tutor me propuso que pensara y desarrolle varias opciones que cumplan con los requisitos. Mi primera opción fue pensar en una única aplicación con 2 menús, uno para audio y otro para video/imagen. La segunda opción era generar una nueva aplicación únicamente para video/imagen, de esta manera se tendrían 2 y habría que diferenciarlas por el icono. La tercera y última opción era poner un botón dentro de cada superficie para modo audio o modo video, pero este diseño se descartó por no ser las superficies de control exactamente las mismas ni iguales para audio que para video.

El día de la reunión se eligió la opción de una única aplicación con 2 menús. Pero para facilitar la programación y el posterior testeo, se propuso primero generar una aplicación sólo con video/imagen y luego fusionarla en una única para audio y video.

### **TAREA 3. Diseñar las superficies de control.**

El proceso de desarrollo será extraer de cada módulo la información necesaria para el control remoto desde el dispositivo móvil. Una vez claro el plan de acción se empezará a diseñar los diferentes paneles. La versión de video contará con 5 módulos y se desarrollarán siguiendo este orden las superficies de control: Switcher, ImageSampler, Sampler, Color y Blending.

### **TAREA 4. Generar y probar una aplicación para dispositivos móviles.**

Con todas las superficies terminadas, el último paso será generar una aplicación que incluya todos los paneles desarrollados y probar su funcionamiento desde el smartphone. Se diseñará un “menú principal” donde se introducirán los datos necesarios de la conexión y una barra superior de botones para poder navegar por las diferentes opciones.

Cuando todos los errores sean corregidos, se fusionarán los dos bloques (audio y video) en una única aplicación. También se tendría que probar esta nueva herramienta antes de lanzarse.

### **TAREA 5. Redacción del TFG.**

Como en cualquier proyecto, el último paso es documentar todo el proceso de implementación. En esta tarea se detalla y explica todo el proceso empleado para generar la aplicación de *Soundcool* y toda la información referente.

## **3.3 Distribución del proyecto en iteraciones.**

A continuación, se detallan los sprints o iteraciones en las que se fragmentó el proyecto para optimizar su producción.

### **Sprint 0**

Reunión con todo el equipo de *Soundcool* y presentación del proyecto.

### ***Sprint 1***

Desarrollo y testeo de la superficie de control Switcher. (Nota: se explicará más adelante la funcionalidad de cada una de las superficies de control).

### ***Sprint 2***

Desarrollo y testeo de la superficie de control ImageSampler.

### ***Sprint 3***

Desarrollo y testeo de la superficie de control Sampler.

### ***Sprint 4***

Desarrollo y testeo de la superficie de control Color.

### ***Sprint 5***

Desarrollo y testeo de la superficie de control Blending.

### ***Sprint 6***

Implementación de la primera aplicación (solo video) y del icono que la distinga de la otra versión.

### ***Sprint 7***

Implementación de la segunda aplicación unificando los bloques de audio y video.

## **3.4 Diagrama temporal**

La herramienta utilizada para la distribución temporal de las tareas ha sido un Diagrama de Gantt. Este tipo de gráfico viene con el listado de tareas ordenadas por su aparición en el desarrollo del proyecto y tienen definida su fecha de inicio, su fecha de fin y el tiempo que se ha tardado en realizarlas (orientativo). Con este diagrama también se puede analizar que tareas son independientes y cuales dependen de su precedente.

Inicialmente, se propuso un periodo de 1 mes para el diseño de cada módulo con posibilidad de ampliación en el caso de no ser suficiente. En este tiempo se haría el diseño, la programación, la configuración de los mensajes OSC y la comprobación de su correcto funcionamiento con varios compañeros en el laboratorio.



El planteamiento inicial se modificó a los pocos días de comenzar el proyecto. Se comenzaron a realizar las superficies de control de los módulos y se vio que el diseño con las funciones básicas de comunicación OSC estaban listas a los pocos días (como máximo 1 semana). Entonces, se optó por crear las superficies de control la vez que los módulos en MAX de vídeo que funcionaban en el ordenador. Debido a ello, cuando se realizaba cualquier ajuste en los módulos se disponía de la app de móvil para aplicar inmediatamente la modificación. Como por ejemplo, se crea un objeto de tipo toggle que activa o desactiva un panel y después de probarlo se decide que envíe su estado por comunicación OSC al ordenador.

Resumiendo, con la modificación explicada anteriormente se realizaba el diseño de las superficies que controlan los módulos de vídeo realizados en MAX inmediatamente que se tenían terminados por la programadora de dichos módulos (Elena Pelejero, miembro del equipo). Esto supuso una reducción del tiempo, pero al proponer varias ideas sobre la funcionalidad que debía tener cada módulo se aumentó el tiempo de programación de cada módulo. Si la aplicación se quería para diciembre del 2017, ahora se tendría para febrero de 2018. También, con una programación abierta, se podía realizar ajustes del funcionamiento de cada superficie de control en cualquier momento, incluso después de tener todas juntas en una misma aplicación.

Debido a actualizaciones necesarias de los diferentes módulos, la presentación del TFG se ha retrasado hasta la convocatoria actual (Sept. 2018) hasta tener una aplicación con todas las mejoras aplicadas.

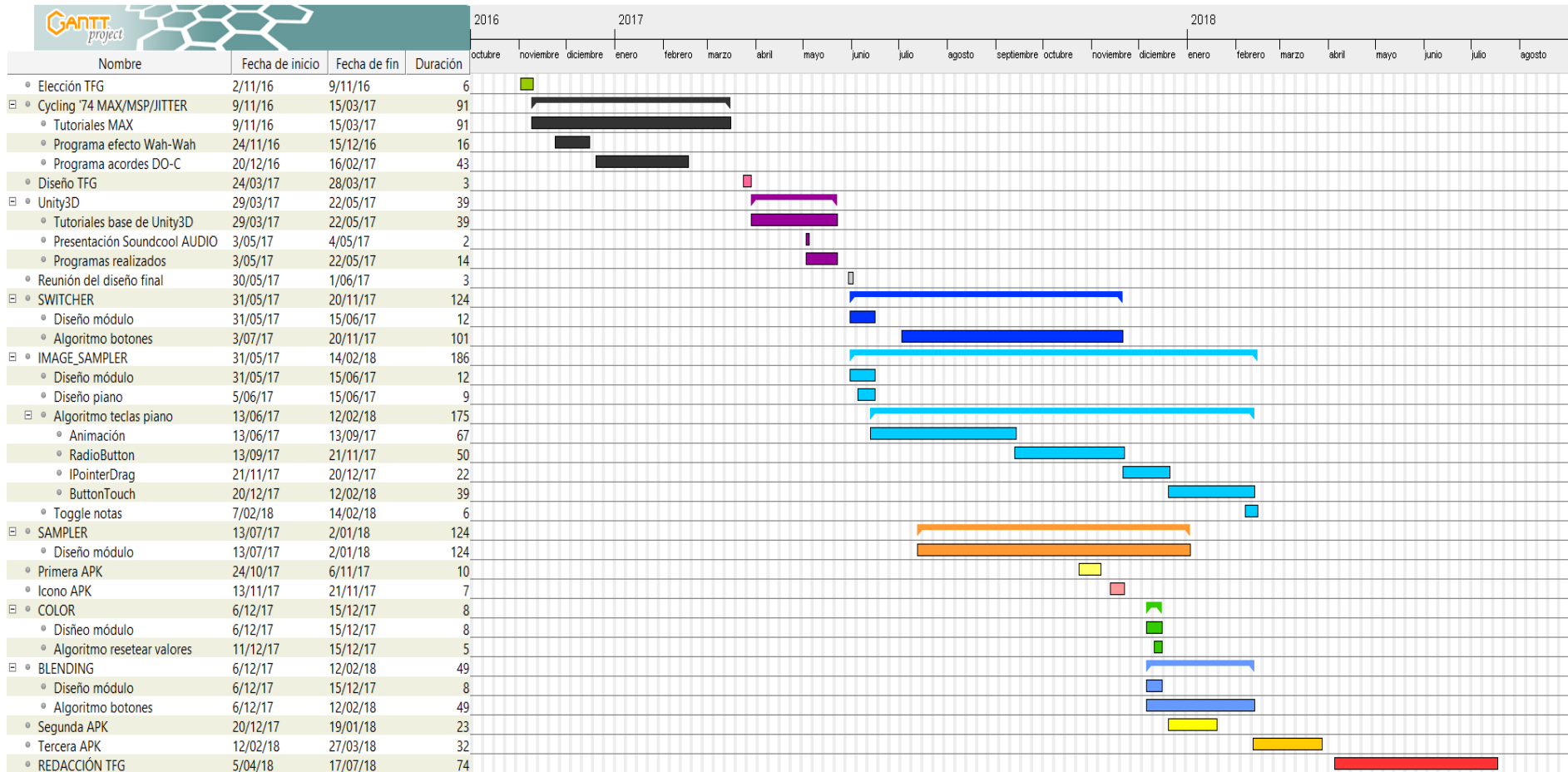


Ilustración 4: Diagrama de Gantt final

## Capítulo 4. Tecnologías empleadas en el desarrollo del proyecto

En este capítulo se introducirán los programas utilizados en la creación de la herramienta *Soundcool*, como son MAX/MSP/JITTER y Unity3D. También se describirán las principales características del protocolo OSC (Open Sound Control) utilizado en la comunicación. Por último se presentarán el hardware utilizado en la programación y testeo de la aplicación.

### 4.1 Software

#### 4.1.1 MAX/MSP/JITTER

Se trata de un software de desarrollo gráfico creado por la empresa Cycling '74. Está diseñado para la producción de contenido multimedia, tanto musical como visual, de un modo profesional con posibilidad de trabajar en tiempo real y de forma interactiva. También cuenta con una interfaz de desarrollo para programadores basada en API de JavaScript, C++ y el lenguaje Gen exclusivo para DSP. El entorno no solo se limita a proyectos audiovisuales, se pueden desarrollar programas para dispositivos electrónicos con sensores. MAX presenta posibilidades para conectar toda clase de periféricos como USB, tarjetas de sonido, micrófonos, Kinect, controladoras MIDI/OSC, sintetizadores, proyectores, etc. Por la parte electrónica permite controlar placas basadas en componentes de Arduino [24] con multitud de sensores. También se puede utilizar como una plataforma para la investigación avanzada a través de las API de Javascript y C++, el lenguaje Gen exclusivo para DSP y efectos visuales y miles de extensiones de asociados.

Además, es un entorno de desarrollo modular, en otras palabras, está basado en pequeños bloques con entradas y salidas que se conectan entre sí mediante “cables virtuales”. Cada bloque realiza una función única, por ejemplo sumar 2 señales; pero esto no limita la creación porque se pueden combinar tantos bloques como quiera el usuario. Supone un salto en cuanto a la producción, ya que no existe el hándicap de revisar líneas y líneas código de texto en busca de los errores. Con la reducción del tiempo de programación, el usuario puede dedicar mucho más tiempo a plasmar sus ideas y generar objetos más completos.

La ventana principal de MAX es un cuadro de diálogo donde aparecen las líneas de comandos que se están ejecutando y los errores en caso de que existan.

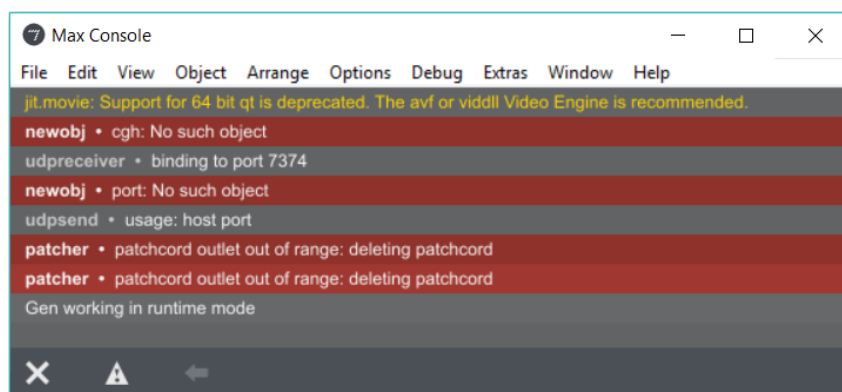


Ilustración 5: Ventana principal de MAX

Desde este menú principal accedemos a la opción de crear un nuevo proyecto en “File” y “New Patcher”. Se abre una nueva ventana en blanco donde disponemos de todas las herramientas para programar nuestro módulo. En la barra superior dispondremos de todos los elementos gráficos como botones, toggles y sliders. La barra lateral izquierda contiene todos los bloques de programación disponibles separados en categorías: audio, imagen y video. En el otro lado, la barra

lateral derecha referencia toda la información del objeto seleccionado, Por último, la barra inferior tiene las opciones de ejecución y visualización de módulo completo.

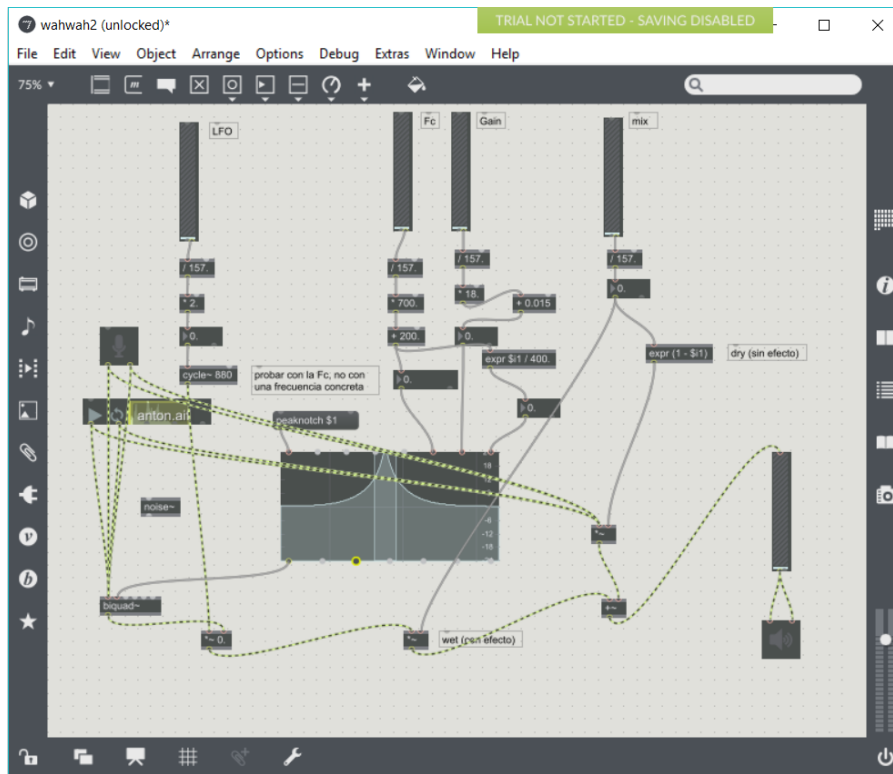


Ilustración 6: Ejemplo de un patcher o programa gráfico de MAX

Este software está creado solo para las plataformas Windows y Mac OS. Su versión completa para programadores se tiene que comprar y el equipo *Soundcool* tiene varias licencias. Max Player, que permite utilizar los módulos de *Soundcool* sin tener la licencia completa de MAX, y que será la que utilizarán los usuarios. MAX permite realizar aplicaciones standalone que llevan integrado Max Player, y esta es la opción final por la que se ha optado con *Soundcool*, realizando una aplicación denominada “Launcher de *Soundcool*” que funciona tanto en Mac como en PC.

Por último, comentar que el programa tiene dos bloques diferenciados como son MSP y JITTER. La parte MSP es la utilizada para las señales como es el caso del audio. Y la parte de JITTER está diseñada para trabajar con imágenes y videos en tiempo real.

#### 4.1.2 Unity3D

Unity3D ha sido la herramienta utilizada en el diseño de las superficies de control para dispositivos móviles del proyecto. El principal motivo es porque Unity permite programar una sola vez y compilar la aplicación para diferentes plataformas como son Android o iOS. En el presente proyecto se ha realizado la compilación en Android. La de iOS es sencilla a partir de la anterior. Además se contaba con dos profesores del equipo con experiencia en Unity3D.

Es un producto de Unity Technologies y ofrece una plataforma para crear juegos y App en 2D, 3D, virtual reality (VR) y augmented reality (AR). Consta de un poderoso motor gráfico basado en OpenGL [25] y Direct3D y un editor muy intuitivo para programadores. Se conecta con facilidad al ordenador, Internet, dispositivos móviles, sistemas para el hogar o visualizadores de 3D.

Esta empresa cuenta con una extensa plataforma web donde poder encontrar toda la información sobre sus librerías y clases de programación. También tiene accesibles numerosas guías y



tutoriales en video explicando cómo utilizar el editor y sus funciones asociadas. Además, tiene un foro donde los usuarios pueden dejar sus dudas y comentarios junto con un repositorio de proyectos disponibles para todo el mundo.

Este software se puede instalar solamente en los sistemas operativos Mac OS y Windows. Dispone de una versión básica totalmente gratuita y una versión de pago para el mundo profesional. Después, desde el mismo programa, se pueden generar aplicaciones para multitud de plataformas como Windows, Mac, Linux, Android, e iOS.

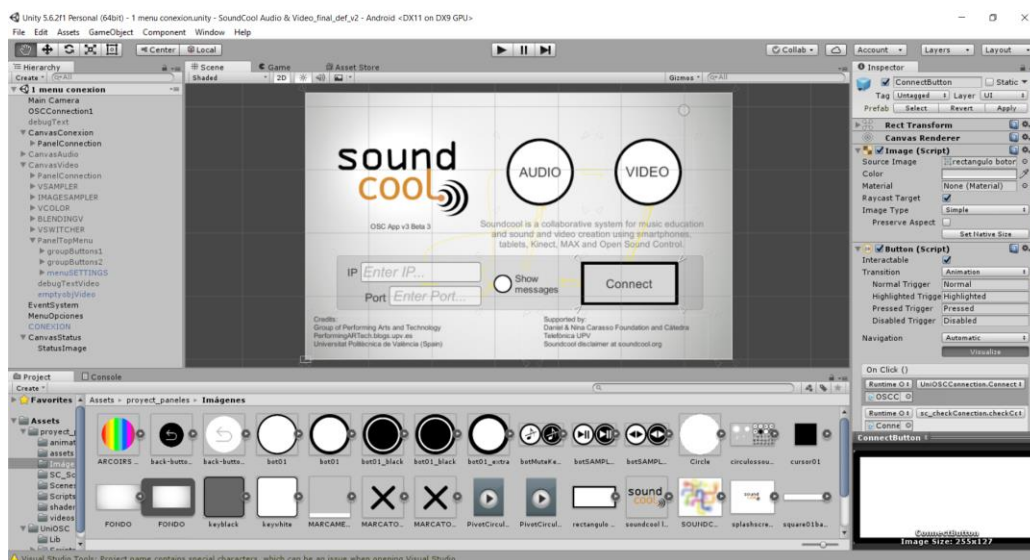


Ilustración 7: Ventana principal de Unity3D

Tal como se muestra en la ilustración 7, el programa está dividido en estas 5 secciones:

- **Barra de herramientas:** situada en la parte superior de la ventana y consta de botones para manipular objetos, ajustar tamaños, lanzar el programa, pausar el programa y menús de capas.
- **Ventana de jerarquía:** situada en la parte izquierda, es una representación de texto jerárquico de cada objeto de la escena. Desde esta ventana se puede hacer “parenting”, que es hacer padre o hijo a un objeto de otro.
- **Vista de escena:** situada en la parte central muestra la disposición visual de los objetos de la escena. Su principal función es seleccionar y manipular los objetos de una forma rápida y puede mostrarse en perspectiva 2D o 3D dependiendo de la aplicación generada. También cuenta con Vista de juego, con la que se puede ver el aspecto del proyecto finalizado.
- **Ventana de inspector:** situada en la parte derecha, se utiliza para visualizar y editar todas las propiedades del objeto seleccionado. Teniendo en cuenta que diferentes objetos tienen propiedades distintas existen parámetros modificables generales como posición y tamaño, y propios del objeto como tipo, scripts, eventos o animaciones.
- **Ventana de proyecto:** situada en la parte inferior, muestra todos los assets que contiene el proyecto. Estos assets se organizan en carpetas por parte del usuario para tener una buena organización. Los tipos de assets que solemos encontrar son: imágenes, videos, clips de sonido, texturas, materiales de objetos, scripts y animaciones.

### 4.1.3 Componentes de Unity3D utilizados en el proyecto y características.

En este apartado se van a describir los componentes utilizados en la programación de las funciones y en el diseño de las superficies de control. Se utilizarán dos tipos de componentes, visuales y de iteración.

Los componentes visuales son los utilizados en la creación del diseño de las superficies de control. Estos elementos permiten al programador modificar las propiedades visuales de la aplicación como crea necesario. Tenemos el componente “Text” que permite mostrar un texto introducido por el usuario. También se utilizará “Image” en la visualización de todos los objetos visuales relacionados con los de iteración.

Los componentes de iteración son los encargados de capturar eventos del mouse, del teclado o táctiles en la pantalla. Estos objetos no son visibles si no se combinan con elementos visuales. También tienen una funcionalidad integrada compartida para visualizar transiciones entre estados. En el proyecto se utilizaran los componentes de iteración “Button”, “Toggle” y “Slider”.

A continuación, se describirán brevemente todos los componentes utilizados.

#### Text

El componente “Text”, también conocido como “Label”, tiene definida un área donde ingresar el texto a mostrar. En la ventana “Inspector” aparecen todas las propiedades editables de los objetos utilizados dentro de este componente visual. Únicamente se explicaran los objetos de los componentes utilizados en el proyecto.

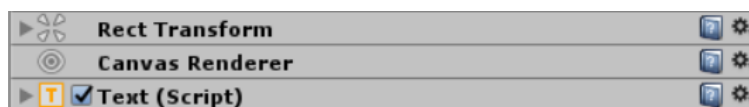


Ilustración 8: Objetos del elemento “Text”

Las propiedades de los objetos mostrados en la ilustración 8 son:

- **Rect Transform:** define un área rectangular donde se colocará el componente elegido, en este caso “Texto”. Se utiliza en el posicionamiento del objeto respecto a los otros objetos del proyecto. Los parámetros con los que cuenta son la modificación de la posición (componentes X, Y, Z), altura y anchura, rotación y escalado. El elemento “Rect Transform” aparecerá por defecto en todos los componentes utilizados en el proyecto.
- **Canvas Renderer:** este componente es utilizado para renderizar un objeto gráfico contenido dentro de un “Canvas”. No tiene propiedades expuestas en el inspector y en el proyecto estará presente en todos los componentes porque está contenido dentro de un “canvas”.
- **Text:** tiene definidas variables de letra como tipo de fuente, estilo, tamaño y espaciado. También tiene propiedades de párrafo como alineamiento, desbordamiento horizontal y vertical y la opción encaje perfecto, el texto se adapta al espacio disponible. A parte tenemos las opciones básicas de color y tipo de material.

## Image

Este elemento se ha utilizado principalmente para hacer visibles todos los componentes del proyecto. También se ha incorporado en las opciones del objeto “Panel” para crear los “Background” de las superficies de control.

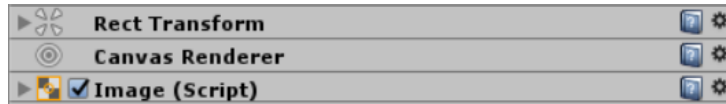


Ilustración 9: Objetos del elemento “Image”

El objeto que diferencia este elemento de los demás es:

- **Image**: cuenta con propiedades como seleccionar la imagen de un archivo, modificar el color y la transparencia, seleccionar el material y el tipo de escalado de la imagen dentro del área del proyecto. Estas características aparecen por defecto al añadir los objetos de tipo imagen.

## Button

El botón se ha utilizado para capturar los eventos correspondientes a un click sobre la pantalla del dispositivo móvil. Su función es iniciar una acción cuando se detecta la pulsación y finalizar la acción cuando se interrumpe la pulsación.

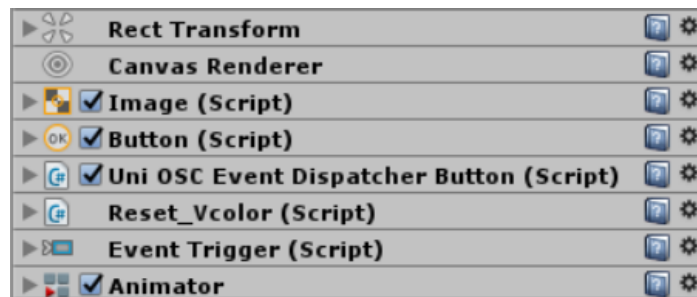


Ilustración 10: Objetos del elemento “Button”

El botón puede tener estos elementos configurados para su funcionamiento:

- **Button**: tiene como característica principal el tipo de transición entre los estados “Normal”, “Pressed”, “Highlighted” y “Disabled”. Son 3 los tipos de transición existentes: “Color Tint”, “Sprite Swap” y “Animation”. *Color Tint* selecciona un color para cada estado. *Sprite Swap* selecciona una imagen para cada estado. Y por último, el *Animator* carga una animación creada por el usuario entre los diferentes estados. También cuenta con un lanzador de eventos llamado “OnClick” donde el usuario puede seleccionar que funciones se ejecutan cuando se pulsa el botón. Esta característica existe por defecto en todos los objetos de tipo “Button”.
- **Uni OSC Event Dispatcher Button**: es un script perteneciente a la librería UniOSC que implementa la comunicación mediante el protocolo Open Sound Control. En él se definen parámetros como la URL de salida, la conexión, y los valores de “Up” y “Down”. Este objeto ha sido añadido por el usuario para este proyecto en particular.
- **Reset\_Vcolor**: es un script que ha sido programado y añadido por el usuario para controlar ciertos efectos cuando se acciona un elemento concreto perteneciente a la superficie.

- **Event Trigger:** se trata de un disparador de eventos, según el evento que se detecte ejecutará una función diferente. Se han utilizado 4 tipos de eventos: “Pointer Enter”, “Pointer Exit”, “Pointer Down” y “Pointer Up”. *Pointer Enter* y *Pointer Down* se ejecutarán cuando se pulse sobre el botón. *Pointer Up*, en cambio, se ejecutará cuando se deje de pulsar el botón. Por último, *Pointer Exit* se ejecutará cuando se pulse sobre la pantalla pero fuera del botón concreto.
- **Animator:** es un motor de animación utilizado para crear y definir una animación sobre el objeto seleccionado. En este caso, anima la transición entre los diferentes estados del botón.

### Slider

Más conocido como deslizador, el slider tiene un indicador dentro de un área definida y el usuario puede variar el valor de salida desplazando dicho indicador por toda la superficie. Consta de 3 elementos, el background o imagen de fondo, fill area o area seleccionada y el handle o indicador.

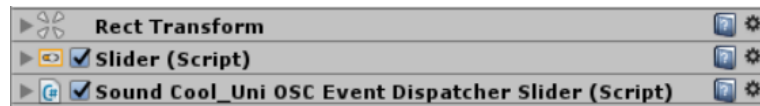


Ilustración 11: Objetos del elemento “Slider”

Objetos asociados al componente durante el proyecto:

- **Slider:** contiene los campos para definir el valor mínimo y máximo de salida que puede tomar. Se puede seleccionar la dirección de los valores dependiendo de la orientación del objeto. Cuenta con un indicador del último valor seleccionado. También tiene un lanzador de eventos llamado “OnValueChanged” donde el usuario puede elegir que funciones se ejecutan cuando cambia el valor.
- **Sound Cool\_Uni OSC Event Dispatcher Slider:** es el script de la librería UniOSC asociado al objeto “slider”.

### Toggle

Este elemento funciona como un conmutador, cuando se pulsa se activa y se mantiene activo hasta que vuelve a ser pulsado. Se diferencia el estado activo del estado normal con una “checkmark”, para este proyecto será una “X”. Se ha utilizado para hacer visibles paneles con objetos como texto o botones.



Ilustración 12: Objetos del elemento “Toggle”

El “Toggle” cuenta con los siguientes objetos:

- **Toggle:** este objeto tiene la opción “Is On” que establece si el elemento está activo al iniciar la aplicación o no. Contiene un campo para enlazar la imagen de la “checkmark” y otro para seleccionar el “toggle group” al que pertenece. También cuenta con el evento “OnValueChanged” donde se pueden añadir las funciones que se ejecutaran al ser accionado.

- **Uni OSC Event Dispatcher Toggle:** script de la librería UniOSC asociado a “Toggle”.

#### 4.1.4 Visual Studio Basic

Visual Studio [26] es el software utilizado en la creación del código de programa con las funciones definidas por el usuario. Es una herramienta de programación desarrollada por Microsoft. Este entorno permite crear proyectos utilizando lenguajes como C#, VB, C++, JavaScript y Python. El lenguaje de programación que se utilizará en el proyecto es C#. La herramienta funciona en los sistemas operativos Linux, Windows y MAC.

En el proyecto se utilizará Visual Studio asociado a Unity3D, significando que el código se probará y ejecutará desde Unity3D. Las funciones del Visual Studio que se utilizarán serán las básicas, abrir archivos, guardar, buscar instrucciones y crear las funciones de código.

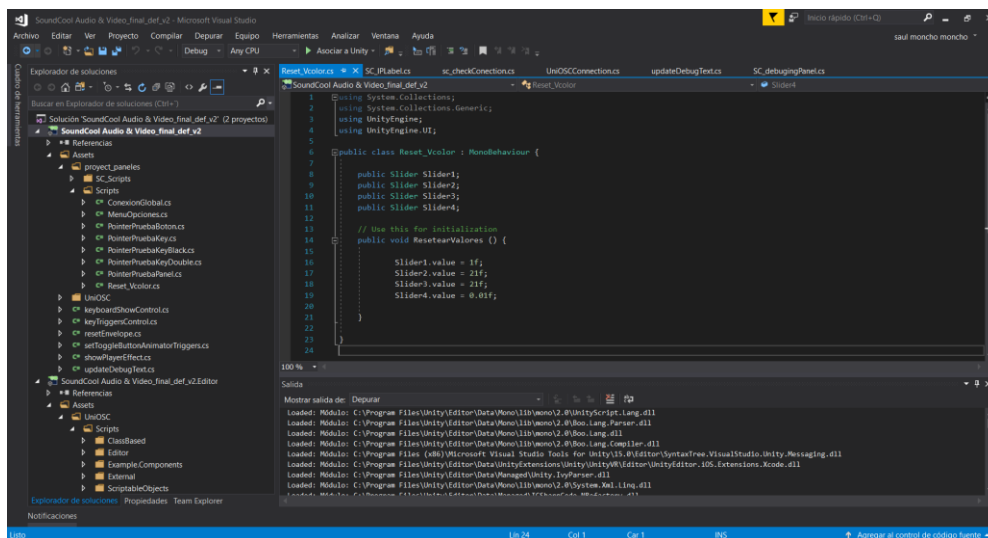


Ilustración 13: Ventana principal del Visual Studio

En la ilustración 13 se visualiza la ventana principal del programa dividida en 4 secciones:

- Barra herramientas del programa. Situada en la parte superior tiene todas las opciones del Visual Studio.
- Ventana principal. Situada en la parte central derecha contiene todas las líneas de código que el usuario escribe.
- Ventana de ejecución. Situada en la parte inferior contiene todo el código que se ejecuta del programa. También se mostraría los errores en caso de haber.
- Cuadro de herramientas. Situado en la parte izquierda tiene dos opciones, el explorador y las propiedades.

#### 4.1.5 Protocolo OSC

El protocolo Open Sound Control fue desarrollado por el CNMAT (Center for New Music and Audio Technologies) para sistemas distribuidos de música y es una clara alternativa al protocolo MIDI. Se utiliza para la comunicación entre computadoras, sintetizadores de sonido y otros dispositivos multimedia, y no solo es una alternativa al MIDI, sino que mejora todas sus características. Envía instrucciones de control en una red de telecomunicaciones entre emisor y receptor y no necesita conexión a Internet para trabajar.

Las principales características de este protocolo son:

- Protocolo dinámico y extensible en tiempo real.
- Los datos son encapsulados con el protocolo UDP, el cual agiliza la comunicación, pero también se pueden usar otros protocolos como el TCP.
- Los mensajes se forman por el usuario siguiendo un esquema de nombres simbólicos de tipo URL.
- Pueden llevar contenido en formato entero, flotante o cadena de caracteres.
- Permite una comunicación simultánea entre varios dispositivos con un único mensaje utilizando “pattern matching”.
- Posibilidad de empaquetar varios mensajes para aquellas instrucciones que se tengan que ejecutar en el mismo instante.



**Ilustración 14: Partes del mensaje OSC**

El mensaje está compuesto por una cadena de bytes dividida en 3 partes como se puede ver en la ilustración 14:

- La primera parte del mensaje corresponde a la dirección y se trata de una cadena de caracteres. Utiliza el símbolo “/” para los objetos en diferentes carpetas como un sistema de tipo URL de internet permitiendo la navegación en una estructura jerárquica.
- La segunda parte define el tipo de datos que contiene con una letra: “i” para enteros, “s” para una cadena de caracteres, o “f” para un número con coma flotante. Durante el trabajo no se actuará sobre esta parte del mensaje.
- Por último tenemos los datos que transporta el mensaje que serán del tipo entero o decimal, no se utilizarán cadenas en nuestro proyecto. Corresponderán con los valores de salida que toman los diferentes objetos de la aplicación.

En el proyecto se utilizará una librería comprada basada en el protocolo OSC y llamada UniOSC[27]. Contiene todos los objetos necesarios para establecer la conexión y generar todos los tipos de mensajes OSC. Únicamente se tienen que rellenar los campos de los scripts de la librería con los datos necesarios para su correcto funcionamiento.

#### 4.1.6 UniOSC

UniOSC ha sido desarrollada por Unity utilizando una versión modificada de la librería OSCsharp. En el proyecto, se utilizará esta librería para generar y enviar los mensajes OSC a otros dispositivos conectados vía Wi-Fi. La librería está creada para importar directamente al proyecto y desde éste poder acceder a todos los objetos.

En cuanto a los elementos que se utilizarán en el proyecto tenemos dos tipos. Por un lado tendremos un script asociado a la conexión con los datos de IP y Puerto. Y por otro lado se dispondrá de un script vinculado a cada elemento de control (Button, Toggle o Slider) que ahora se describirán brevemente.

#### *Uni OSC Connection*

El script está dividido en dos partes que se pueden ver en la ilustración 15; una referenciada a los mensajes de entrada y otra a los mensajes de salida. En el proyecto descartaremos la parte de mensajes de entrada “OSC IN” porque únicamente utilizamos el canal de salida para mandar las instrucciones de control. Los parámetros modificables que disponemos son: opción de auto conectar con el receptor al ejecutar la aplicación, elección del puerto, tipo de transmisión (unicast, multicast o broadcast) y una textbox para ingresar la IP del receptor.

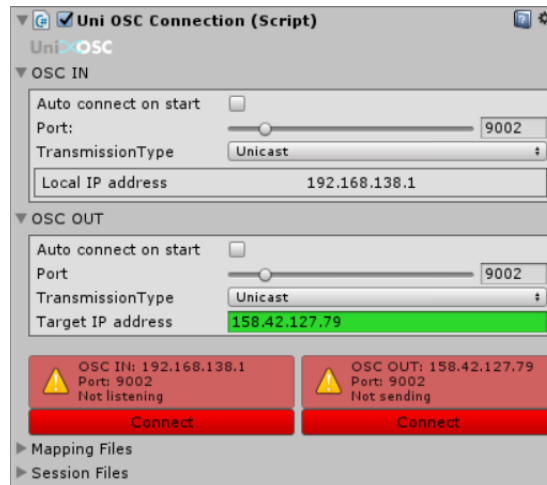


Ilustración 15: Script Uni OSC Connection

En este script se tienen que enviar los datos manualmente con una única IP y un único Puerto. Estos datos se programarían dentro de la aplicación y serían los mismos en cada ejecución por lo que nuestro proyecto no tendría un funcionamiento dinámico. La solución ha sido crear y vincular un script (“SC\_IP Label”) que modifique los valores cuando el usuario crea conveniente.

### *Uni OSC Event Dispatcher Button*

Se trata de un script vinculado a los objetos de tipo “button”. En el envío de mensajes OSC utiliza dos tipos de estado, el correspondiente a la acción de pulsar (Down) y el estado dejar de pulsar (Up). Este script tiene un cuadro de texto para especificar la ruta URL del elemento. También tiene una opción donde nosotros hemos asociado el script “OSCConnection” que contiene los datos de conexión. De no hacerlo de esta manera se tendría que definir para cada objeto la IP y el Puerto. Por último tenemos los datos que se envían. Como el botón tiene dos estados, “Up” y “Down” se enviará 0 y 1 respectivamente. La opción “Show GUI” no la utilizaremos.

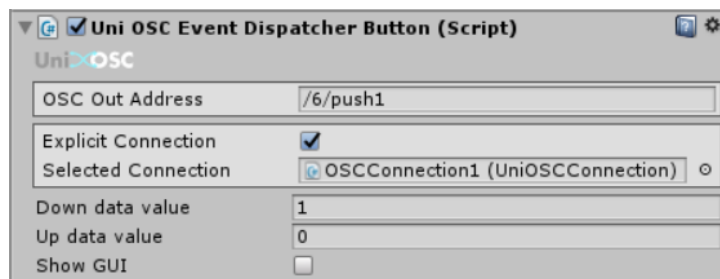


Ilustración 16: Script Uni OSC Event Dispatcher Button

### *SoundCool\_Uni OSC Event Dispatcher Slider*

Este script hace referencia a los objetos de tipo “slider”. Una característica que se puede elegir es su orientación, vertical u horizontal; un dato que a priori parece ser poco útil para el envío de

datos OSC pero que es tan necesario como todos. A continuación se define el rango de valores del componente, mínimo y máximo. Luego siguen las opciones de las posiciones que en este proyecto no se utilizan. El tamaño del slider siempre lo hemos dejado por defecto en 100. La ruta URL del slider y por último los parámetros de la conexión definidos en el script “OSCConnection”.



Ilustración 17: Script Uni OSC Event Dispatcher Slider

### Uni OSC Event Dispatcher Toggle

Se vincula a los elementos de tipo “toggle”. Básicamente tiene las mismas opciones que el botón. El valor que se envía en cada estado de “on” y “off”, la ruta URL del protocolo OSC y los datos de conexión.

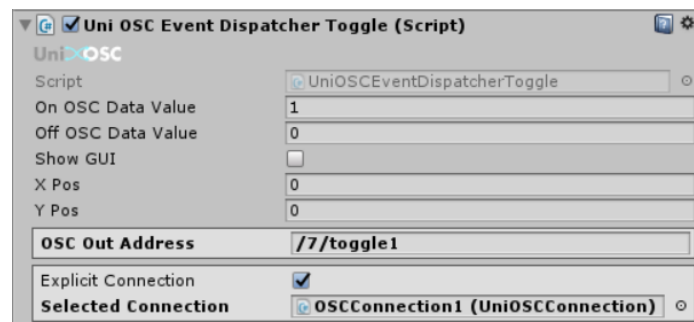


Ilustración 18: Script Uni OSC Event Dispatcher Toggle

## 4.2 Hardware

Para el presente proyecto se ha utilizado el siguiente hardware:





Ilustración 19: Dispositivos electrónicos utilizados durante el proyecto

### PC ASUS ROG STRIX GL553VD DM078

- Procesador: Intel Core i7-7700HQ (4 Núcleos, 2.8GHz hasta 3.8GHz, 6M Cache).
- Controlador gráfico: NVIDIA GeForce GTX1050 con 4GB GDDR5 VRAM.
- Memoria RAM: 24GB (16GB + 8GB) tipo DDR4 a 2400MHz.
- Disco duro: 1TB 7200rpm SATA + 256GB Samsung SSD 960 EVO.
- Display: 15.6" LED retroiluminado, Full HD 1920x1080 (16:9), 200 nits y NTSC: 45%.
- Sistema Operativo: Windows 10 Pro (64bits).

### Smartphone BQ Aquaris E5 4G

- CPU: Qualcomm Snapdragon 410 Quad Core A53 hasta 1.2GHz.
- GPU: Qualcomm Adreno 306 hasta 400MHz.
- Memoria RAM: 1GB.
- Disco duro: 16GB.
- Display: 5" IPS qHD/HD OGS 450 nits, resolución HD 720x1280 (16:9).
- Sistema Operativo: Android 5.0.2 (Lollipop).

### Tablet Teclast X98 pro

- Procesador: Intel Atom Z8500-X5 Quad Core de 1.44GHZ hasta 2.24GHZ.
- GPU: Graphics Intel HD Gen8.
- Memoria RAM: 4GB.
- Disco Duro: 64GB.
- Display: 9.7" IPS con resolución 2048x1536 QXGA.
- Sistema Operativo: Windows 10 (64bits) + Android 5.1.

## Capítulo 5. Desarrollo de las superficies de control de los módulos

En este capítulo se describirá todo el proceso de diseño de las superficies de control de los módulos del MAX. Se van a tener en cuenta los elementos, su disposición, el funcionamiento de cada uno y los mensajes OSC asociados.

La idea general del proyecto es diseñar y programar las superficies de control que enviarán instrucciones a los módulos que se ejecutan en el ordenador. En ningún momento vamos a poder cargar archivos desde el propio dispositivo móvil en la aplicación. Tampoco se podrán enviar archivos desde el ordenador al móvil y viceversa.

Se ha tomado como referencia la aplicación ya existente de *Soundcool* para los módulos de audio y los propios módulos de la parte de video. Por lo tanto, el nuevo diseño y los objetos deberán guardar relación con los antiguos ya desarrollados.

La descripción de los módulos se realizará siguiendo los siguientes pasos:

- Primero se realizará una descripción general del módulo diseñado en MAX. Principalmente se analizará el funcionamiento y la disposición de los elementos.
- Luego se mostrará el módulo creado para el dispositivo móvil teniendo en cuenta el apartado anterior y la aplicación anterior.
- A continuación se describirá el algoritmo de funcionamiento para el envío de mensajes OSC.
- Después enumeraremos los elementos utilizados y su función. También se hará el listado de los elementos en la jerarquía, ya que dependiendo de dónde se sitúe el elemento puede variar su función.
- Por último pondremos un listado con todos los mensajes OSC que envían los distintos objetos.

### 5.1 Switcher

El módulo Switcher programado con el MAX tiene 8 conectores de entrada junto con una pantalla para visualizar el contenido seleccionado. El usuario puede conectar un módulo de reproducción de imágenes, vídeo grabado, vídeo en directo de una cámara, a cada entrada y ver el contenido multimedia. En la parte inferior tiene una pantalla más grande que sirve para ver el video seleccionado y corresponde con la salida del bloque.

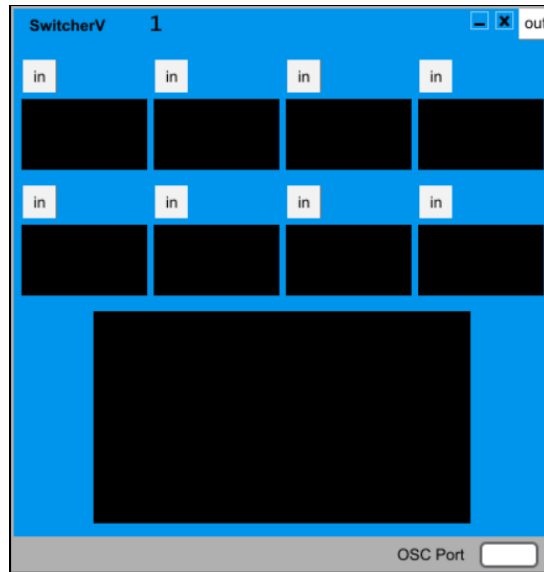


Ilustración 20: Módulo Switcher

El funcionamiento es sencillo, se trata de un conmutador de 8 canales donde el usuario puede elegir cuando y que video visualizar pinchando con el ratón en la pantalla deseada del módulo en el ordenador. Como la acción que se realiza es pulsar sobre la pantalla, se ha pensado en botones para la superficie de control. Dispondremos 8 botones en una cuadrícula de 2 filas y 4 columnas. También pondremos una etiqueta con el número correspondiente de cada elemento. El color del panel de fondo será azul.

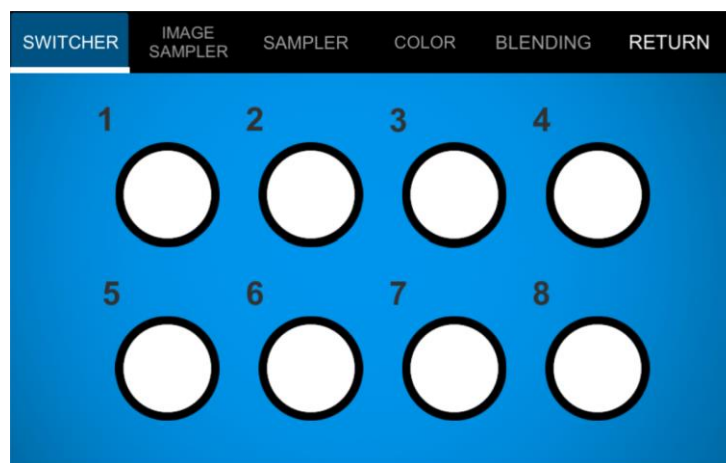


Ilustración 21: Superficie de control del módulo Switcher

### *Funcionamiento de la superficie de control*

La idea inicial era poner un botón con dos estados, “normal” y “accionado”. Es decir, el botón está en “normal” hasta que el usuario pulsa sobre él y pasa a estado “accionado”. Cuando se deja de pulsar el botón, éste vuelve a “normal”. Después de realizar varias pruebas en el laboratorio, nos dimos cuenta que existía la necesidad de saber cuál era el último botón seleccionado. El objeto que más se acercaba a este comportamiento era un “toggle”, pero esto implicaba cambiar toda la programación del módulo en MAX.

A pesar del inesperado problema llegué a una solución aparentemente sencilla para no tener que modificar nada del módulo. Añadimos sobre cada pulsador una imagen en estado “accionado” y

la dejamos en modo invisible. Cuando activamos un botón, la imagen correspondiente se hace visible y las del resto de botones se fuerzan a modo invisible. De esta manera conseguimos tener el último elemento pulsado diferenciado de los demás.

En cuanto a la programación, no se ha necesitado generar líneas adicionales de código. Simplemente se han utilizado las funciones que vienen por defecto en un objeto de tipo botón como es `OnClick` [28].

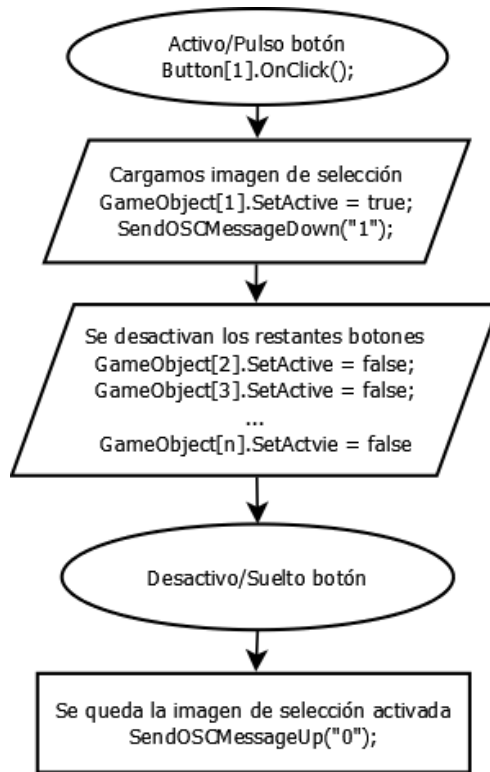


Ilustración 22: Algoritmo del módulo Switcher

### Elementos de control

Los elementos de los que consta el módulo son 8 botones y 8 etiquetas de texto con el número referente a cada uno. Los pulsadores tomarán forma circular, con fondo blanco y un borde negro bastante ancho. Para indicar el estado “accionado” explicado anteriormente será la misma imagen pero con fondo negro. En la ilustración 23 se pueden diferenciar los dos estados.

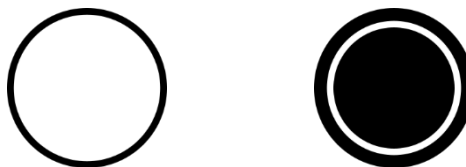


Ilustración 23: Switcher a) botón en el estado “normal” b) botón en el estado “pressed”

La imagen de la izquierda, de color blanco, hace referencia al estado “normal” y, a su vez, la imagen de la derecha, de color negro, es la que tomará el botón en estado “accionado”.

Para tener los botones dispuestos en una cuadrícula 2x4 y con el mismo tamaño se han creado varios paneles. En primer lugar, hemos creado “Panel\_buttons” con un tamaño de 400x800 puntos. Luego se han creado dentro 8 paneles de tamaño 200x200. Por último, se han añadido los botones de tamaño 100x100, la imagen del botón accionado de 100x100 y la etiqueta de texto de

50x50. Con esta forma de enlazar objetos si modificamos el tamaño del panel padre, todos los elementos se reajustarán de manera inmediata respetando sus parámetros iniciales.

### Mensajes OSC

Los mensajes OSC asociados a este panel tendrán el “5” como número de identificación. En la siguiente tabla se puede ver el listado de los mensajes OSC de todos los elementos y su construcción:

SWITCHER			
Number of module	Element	Value	OSC Message
6	push1	0 or 1	/6/push1 0
6	push2	0 or 1	/6/push2 0
6	push3	0 or 1	/6/push3 0
6	push4	0 or 1	/6/push4 1
6	push5	0 or 1	/6/push5 0
6	push6	0 or 1	/6/push6 0
6	push7	0 or 1	/6/push7 0
6	push8	0 or 1	/6/push8 0

Tabla 1: Mensajes OSC del módulo Switcher

En este módulo solo tenemos botones, los mensajes serán idénticos exceptuando el número que diferencia a cada uno. El valor OSC que puede mandar cada botón es “1” o “0”. El “1” va asignado a la acción de pulsar el botón (“Pointer Down”) y a la función SendOSCMessageDown(). El “0” va asignado a la acción de levantar el dedo del botón (“Pointer Up”) y a la función SendOSCMessageUp(). Estas acciones se configuran con un elemento llamado “Event Trigger” o lanzador de eventos. Este objeto cuando detecta que ocurre una acción programada lanza su función asociada.

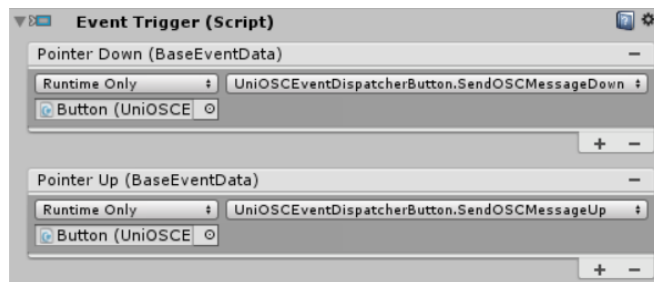


Ilustración 24: Event Trigger del módulo Switcher

## 5.2 ImageSampler

El módulo ImageSampler funciona como un selector de imágenes. Consta de 13 botones dispuestos en forma de piano y se vincula una imagen a cada uno mediante el botón “LOAD”. Por otro lado tenemos 2 toggles, uno llamado “N” que activa o desactiva el nombre de las notas en su tecla correspondiente y el otro “Rand”, que se utiliza para presentar las imágenes de manera aleatoria. También consta de dos sliders, uno controla el fundido a negro en la transición entre

imágenes y el otro controla la velocidad del cambio entre imágenes. La elección de un teclado de piano para la selección de las imágenes se realizó por razones didácticas para permitir relacionar notas musicales e imágenes en el aula.

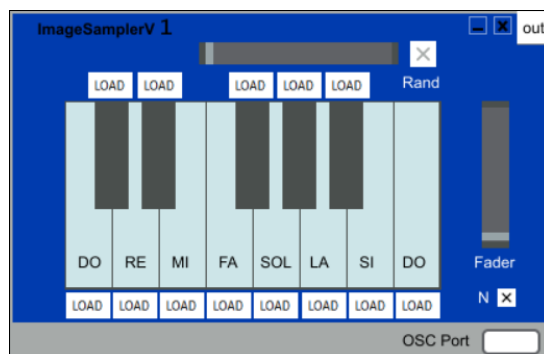


Ilustración 25: Módulo ImageSampler

El usuario puede cargar hasta un total de 13 imágenes asociadas a cada tecla del piano y elegir el tipo de presentación, manual o aleatoria. En el modo manual, el usuario elige una imagen pulsando su tecla asociada. El modo aleatorio, se accede activando el toggle “Rand” y las imágenes se presentan de forma aleatoria.

El diseño para dispositivos móviles se realiza teniendo en cuenta los siguientes elementos: 2 toggles, 2 sliders y 13 botones dispuestos de manera que forman un piano.

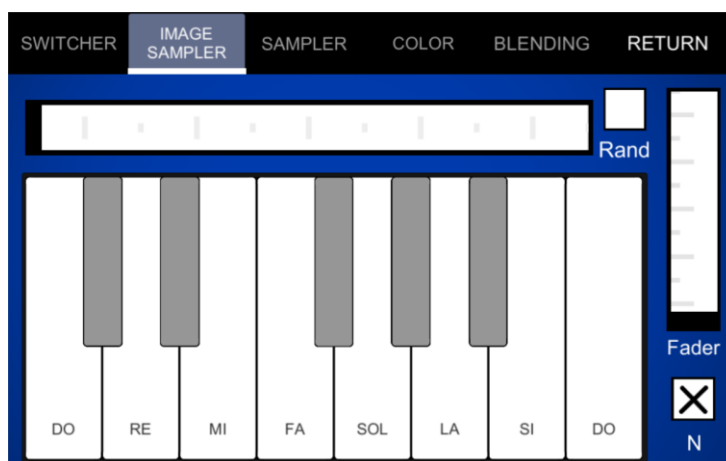


Ilustración 26: Superficie de control del módulo ImageSampler

### *Funcionamiento de la superficie de control*

El teclado funciona de manera monofónica, es decir, que solo es posible pulsar una única tecla. Después de pulsar la tecla se queda diferenciada de las demás hasta que se vuelve a activar otra tecla distinta. Como pasaba con el módulo anterior, para esta idea sería más sencillo utilizar toggles en lugar de botones, pero tendrían un comportamiento distinto al que se pretende.

En el problema del teclado monofónico la solución pensada ha sido desactivar todas las teclas distintas de la pulsada mediante una función. Cuando el usuario pulse varias teclas a la vez siempre quedará activada la última. El segundo problema era destacar la última tecla seleccionada y se ha utilizado el “Event Trigger” y una animación para controlar el estado “Pressed” del botón. A parte, también se han programado dos funciones para poner el objeto no seleccionado en estado “Normal”.

Todas estas soluciones se han incluido en un único script llamado “UniOSCEventDispatcherButtonSC”, de manera que todas las teclas tendrán asociado el mismo.

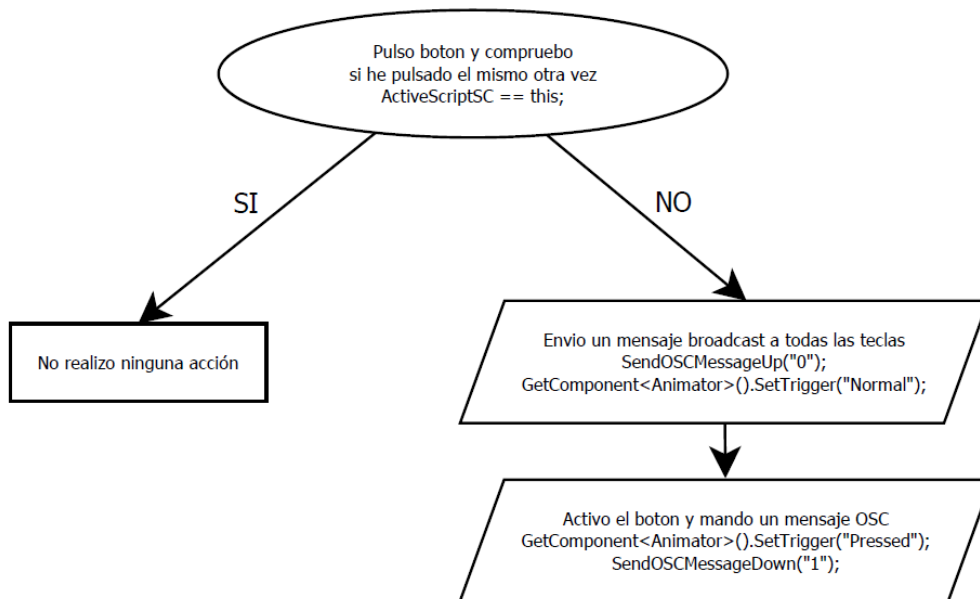


Ilustración 27: Algoritmo de las teclas del piano del módulo ImageSampler

El funcionamiento de las etiquetas con los nombres de las notas es sencillo. Se trata de un toggle que cuando se activa hace visible el texto en cada nota, y al desactivarse vuelve a ocultarlo.

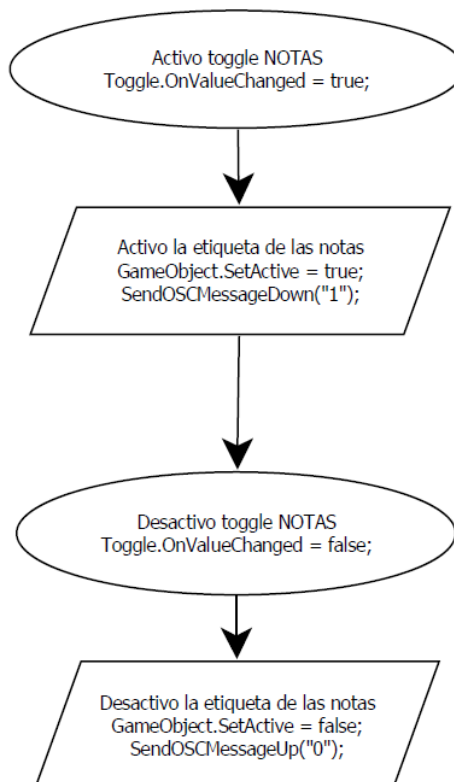


Ilustración 28: Algoritmo del texto de las teclas del módulo ImageSampler

### Elementos de control

El piano ha sido la parte más difícil del diseño ya que en Unity3D no existe ningún objeto que emule un teclado de piano, por lo se ha tenido que realizar utilizando botones y paneles. En primer lugar se ha dispuesto un panel de 800x200 puntos y 8 botones blancos de la medida 100x200. Después se han añadido los botones oscuros con un tamaño de 50x100, respetando el tamaño de los espacios entre teclas y que tengan el mismo ancho. Por último se han creado las etiquetas de texto con el nombre de las notas de tamaño 50x25.

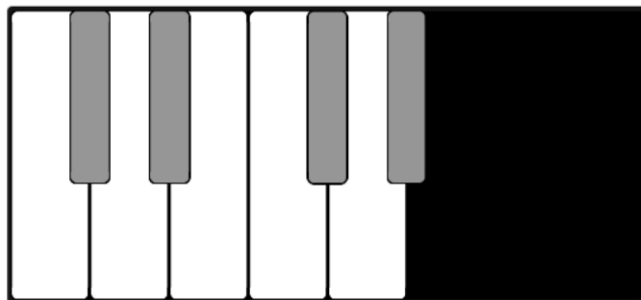


Ilustración 29: Construcción del piano del módulo ImageSampler

Tenemos 2 sliders, uno horizontal y otro vertical. Presentan los dos la misma imagen de fondo, blanca con líneas grises. El área seleccionada tiene un color azul claro relacionado con el color del panel. Por último, el handle es un rectángulo de color negro.



Ilustración 30: Slider del módulo ImageSampler

También hay dispuestos 2 toggles idénticos, con un aspecto cuadrado y con una “X” como checkmark.



Ilustración 31: ImageSampler a) toggle desactivado b) toggle activado

### Mensajes OSC

Los mensajes OSC asociados a este panel tendrán el “3” como número de identificación y están detallados en la tabla 2. Este número es el mismo que tiene el módulo Keyboard del bloque de Audio, por lo que se podrán controlar los dos al mismo tiempo.

En el ImageSampler tenemos los 3 tipos de elementos: 2 sliders, uno horizontal y uno vertical con valores entre 0 y 1; 2 toggles con valor 0 en estado normal y 1 cuando está activo; y 13 botones correspondientes a las teclas del piano, con valor 0 o 1.



IMAGESAMPLER			
Number of module	Element	Value	OSC Message
3	fader1	$0 \leq X \leq 10$	/3/fader1 0.235
3	fader2	$0 \leq X \leq 10$	/3/fader2 0.775
3	toggle1	0 or 1	/3/toggle1 1
3	toggle2	0 or 1	/3/toggle2 1
3	push1	0 or 1	/3/push1 0
3	push2	0 or 1	/3/push2 0
3	push3	0 or 1	/3/push3 0
3	push4	0 or 1	/3/push4 0
3	push5	0 or 1	/3/push5 0
3	push6	0 or 1	/3/push6 1
3	push7	0 or 1	/3/push7 0
3	push8	0 or 1	/3/push8 0
3	push9	0 or 1	/3/push9 0
3	push10	0 or 1	/3/push10 0
3	push11	0 or 1	/3/push11 0
3	push12	0 or 1	/3/push12 0
3	push13	0 or 1	/3/push13 0

Tabla 2: Mensajes OSC del módulo ImageSampler

### 5.3 Sampler

El módulo Sampler está pensado para realizar distintas acciones sobre la reproducción de un video. Tiene dos modos llamados FRAME y LOOP. Con ellos se puede guardar un punto concreto de la reproducción y pulsando un botón acceder a él. O también se puede guardar un fragmento y reproducirlo en bucle.

El diseño de los modos FRAME y LOOP está dividido en dos mitades como se aprecia en la ilustración 32. La mitad superior es la encargada de tomar acciones sobre el video, como abrirlo, visualizarlo, reproducirlo, detenerlo, repetirlo en bucle y fijar el punto de reproducción. Por otro lado, la mitad inferior es la encargada de guardar los instantes de reproducción del video en los botones. Para la programación de la superficie de control del móvil o tablet tendremos en cuenta solo la mitad inferior de cada módulo.

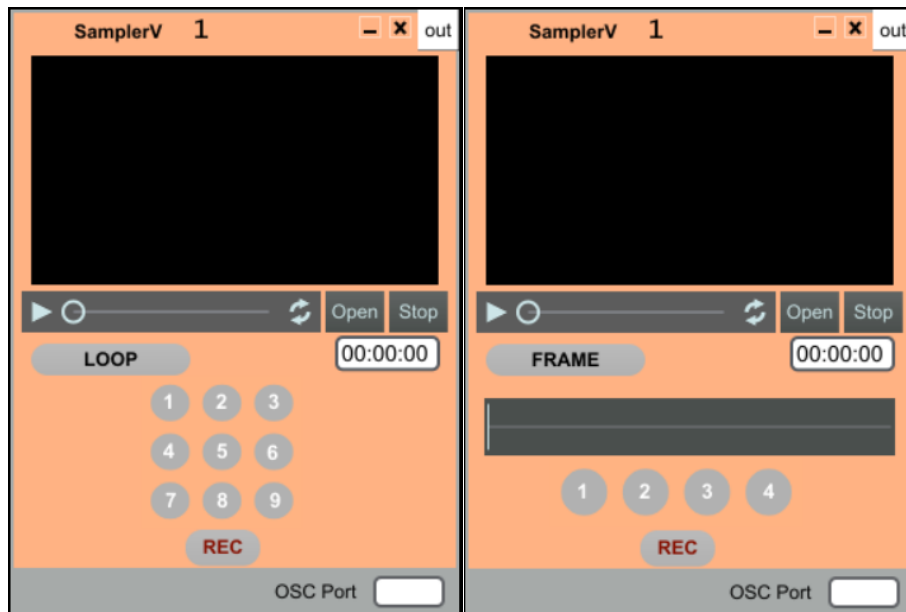


Ilustración 32: Módulo Sampler a) Modo Frame b) Modo Loop

El menú FRAME se utiliza para grabar instantes concretos del video a partir de los cuales empezará la reproducción cuando se seleccionen. Consta de 9 botones numerados y otro llamado REC que es el encargado de activar el menú de grabado. El funcionamiento consiste en cargar un video y con el modo REC guardar en cada botón, del 1 al 9, el instante desde el cual empezar la reproducción.

Por otro lado, el menú LOOP se utiliza para grabar regiones concretas del video las cuales se reproducirán en bucle al seleccionarse. Consta de 4 botones numerados, un botón de REC y una barra que indica la duración del periodo del video seleccionado. El funcionamiento es el mismo que el módulo anterior, pero en este caso reproduce un fragmento del vídeo en bucle.

El diseño para los dispositivos móviles se realizará teniendo en cuenta sólo los botones numerados. La parte de grabar los instantes de reproducción le corresponde a la persona que controle los módulos desde el ordenador. Se creará un panel para cada opción y se dispondrá de un elemento de tipo “toggle” para navegar entre ellos. El color del panel de “background” será amarillo anaranjado.

En primer lugar diseñamos la superficie de control FRAME, que consta de una cuadrícula 3x3 con 9 botones. El toggle se situará en la parte izquierda del panel y pondremos una etiqueta de texto con el nombre del menú al que se accede cuando se activa. En este caso nos encontramos en el modo FRAME y al activar el toggle abriremos el modo LOOP.

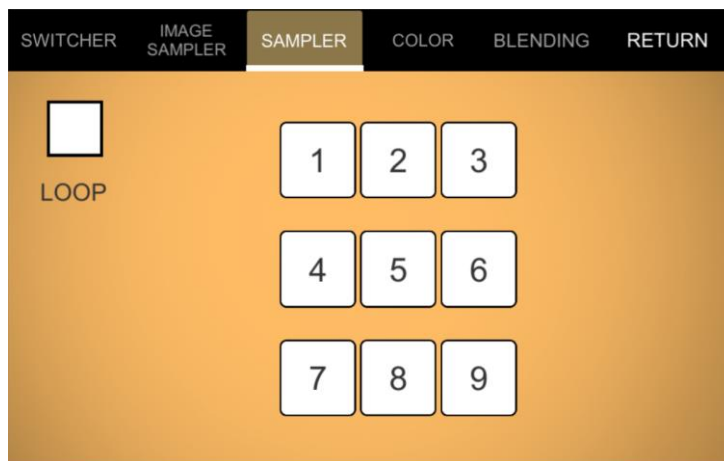


Ilustración 33: Superficie de control del Sampler modo Frame

En segundo lugar diseñamos el modo LOOP, que consta de 4 botones. En este caso la etiqueta de texto situada debajo del toggle es “FRAME” que es el modo al que se accede cuando se desactiva el toggle.

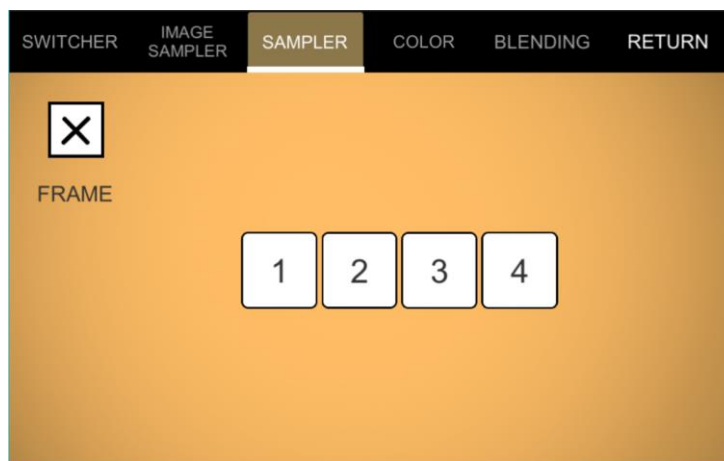
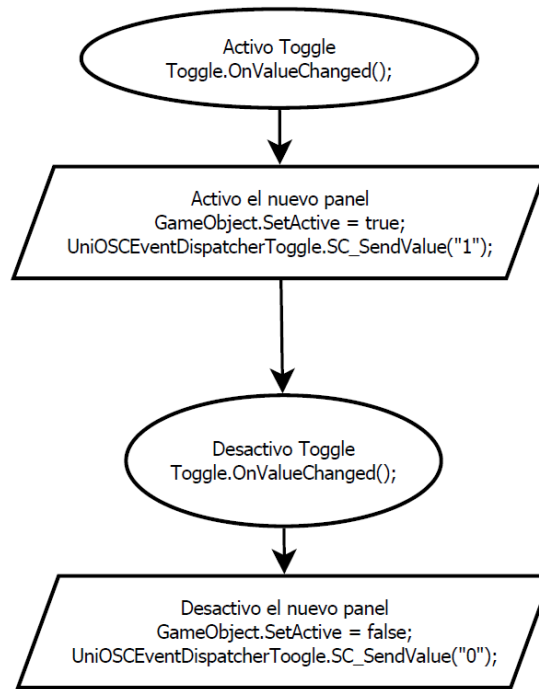


Ilustración 34: Superficie de control del Sampler modo Loop

### *Funcionamiento de la superficie de control*

Este módulo no tiene ningún Script asociado, funciona con los parámetros internos que tiene el toggle. Al activar el toggle se hace visible el panel de LOOP. Del mismo modo, al desactivar el toggle el panel se vuelve a hacer invisible. El algoritmo de funcionamiento es:



**Ilustración 35: Algoritmo del módulo Sampler**

### Elementos de control

El submenú FRAME cuenta con 9 botones dispuestos en una cuadrícula 3x3. Se ha creado un panel llamado “Panel\_9botones” de tamaño 600x600 puntos. Este panel se ha dividido en 9 paneles de tamaño 200x200 dispuestos en cuadrícula. Dentro de estos se ha creado el botón de tamaño 150x150 y la etiqueta de texto con el número.

Por otra parte, el submenú LOOP tiene 4 botones dispuestos en una única fila. Se ha creado un objeto de tipo imagen llamado “Paneltapa” de tamaño 200x800 puntos cuya función es ocultar el contenido del submenú que no esté activo. Éste a su vez se ha dividido en 4 paneles de tamaño 200x200 y se han colocado en su centro los botones de tamaño 150x150 como anteriormente.

Los botones tienen un diseño cuadrado con los bordes redondeados. Tienen un borde fino de color negro y en su estado “default” tienen un color blanco de fondo. Cuando pulsamos sobre ellos adquieren un tono gris-amarillo relacionado con el color del panel.



**Ilustración 36: Sampler a) botón en el estado “normal” b) botón en el estado “pressed”**

Únicamente tenemos un toggle, presenta un aspecto cuadrado y su checkmark es una cruz “X”.



Ilustración 37: Sampler a) toggle desactivo b) toggle activo

### Mensajes OSC

Los mensajes OSC asociados a este panel tendrán el “8” como número de identificación, tanto para el panel de LOOP como el panel de FRAME. En la siguiente tabla se puede ver el listado de los mensajes OSC de todos los elementos y su construcción:

SAMPLER			
Number of module	Element	Value	OSC Message
8	toggle1	0 or 1	/8/toggle1 1
8	push1	0 or 1	/8/push1 0
8	push2	0 or 1	/8/push2 0
8	push3	0 or 1	/8/push3 0
8	push4	0 or 1	/8/push4 0
8	push5	0 or 1	/8/push5 0
8	push6	0 or 1	/8/push6 0
8	push7	0 or 1	/8/push7 0
8	push8	0 or 1	/8/push8 1
8	push9	0 or 1	/8/push9 0
8	push10	0 or 1	/8/push10 0
8	push11	0 or 1	/8/push11 0
8	push12	0 or 1	/8/push12 0
8	push13	0 or 1	/8/push13 0

Tabla 3: Mensajes OSC del módulo Sampler

## 5.4 Color

El módulo Color está pensado para modificar las propiedades de la imagen como son el brillo, la saturación, el contraste y el matiz de color. Estas propiedades se modifican con 4 sliders horizontales y se pueden ver el resultado en tiempo real. Por defecto, el módulo se inicia con unos valores preestablecidos en cada slider para que la representación de la imagen o vídeo sea la original. También incluye un botón para restablecer esos valores iniciales.

Desde otro punto de vista, este módulo no puede cargar ni visualizar las imágenes resultantes. Depende de un módulo para abrir imágenes y de otro de salida para poder ver el resultado. Aunque es un punto que no afecta en el diseño de la superficie de control, si se tiene que tener en cuenta para cuando se realice el montaje del sistema al completo.



Ilustración 38: Módulo Color

El diseño para los dispositivos móviles contará con los 4 sliders horizontales junto con el botón de reseteo de valores. Cada elemento incluido tendrá su propia etiqueta de texto indicando el parámetro que modifica.



Ilustración 39: Superficie de control del módulo Color

### *Funcionamiento de la superficie de control*

Esta superficie de control lleva programados dos algoritmos. Uno referente a los valores cargados por defecto al iniciar la aplicación. El relacionado con un botón para reiniciar los valores. Siempre que pulsemos el botón los faders tomarán sus valores por defecto. En definitiva se trata de un código sencillo de entender y fácil de programar.

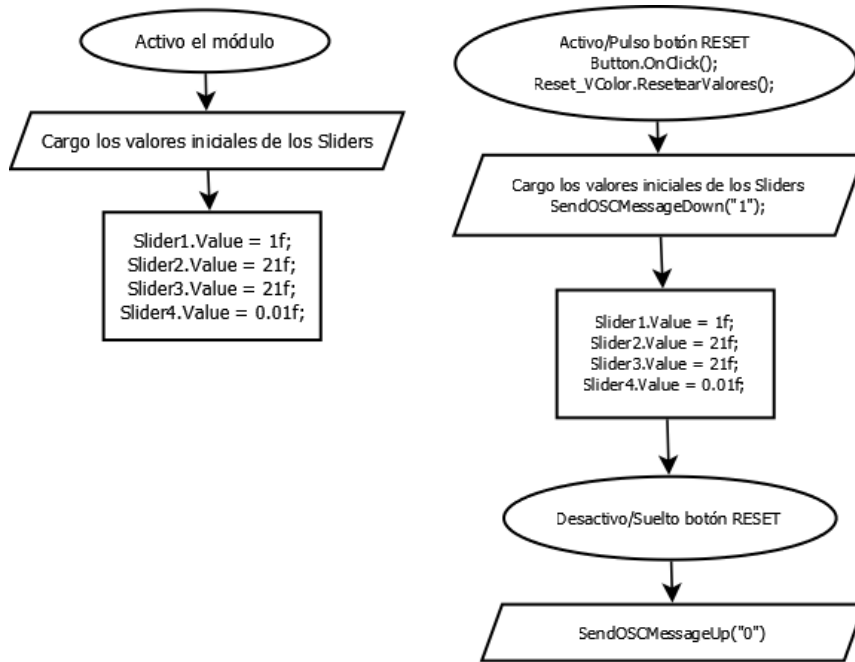


Ilustración 40: Algoritmos del módulo Color

### Elementos de control

Los sliders se han dispuesto horizontalmente en 4 filas. Se ha creado un panel principal llamado “Panel\_Sliders” de tamaño 600x800 puntos. Dentro de este se han añadido 4 paneles, uno para cada slider, de tamaño 150x800. En la parte inferior se ha situado el botón de tamaño 75x75 puntos.

Los sliders tienen una imagen blanca de fondo con unas barras grises difuminadas. El área seleccionada se verá de color verde claro, guardando relación con el color de fondo. El “Handle” presenta un aspecto rectangular y es de color negro.



Ilustración 41: Slider del módulo Color

El botón presenta un aspecto cuadrado con los bordes redondeados. Tiene un borde de color oscuro y el fondo blanco por defecto. Cuando se pulsa se anima de un color verde oscuro.



Ilustración 42: Color a) botón en el estado “normal” b) botón en el estado “pressed”

### Mensajes OSC

Los mensajes OSC asociados a este panel tendrán el “9” como número de identificación. Los datos que se envían en los mensajes y a modo de ejemplo son:

COLOR			
Number of module	Element	Value	OSC Message
9	slider1	$0 \leq X \leq 10$	/9/slider1 5.639
9	slider2	$0 \leq X \leq 40$	/9/slider2 31.266
9	slider3	$0 \leq X \leq 40$	/9/slider3 19.488
9	slider4	$0 \leq X \leq 360$	/9/slider4 255.831
9	push1	0 or 1	/9/push1 0

Tabla 4: Mensajes OSC del módulo Color

El módulo Color dispone de un botón de “Reset”, que como todos los botones envía el valor 1 o 0 dependiendo de su estado. En este caso los sliders tienen valores diferentes, en consecuencia se tienen que rutar bien los elementos o no funcionarán correctamente.

Continuamos utilizando un EventTrigger para enviar los mensajes “Down” y “Up” del botón en las acciones de “Pointer Down” y “Pointer Up” respectivamente. Como también enviamos los datos de los Sliders utilizamos la propia función del objeto OnValueChanged [29] que llama, a su vez, a la función SC\_SendValue().

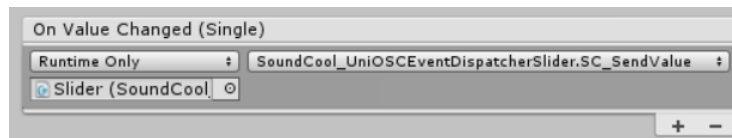


Ilustración 43: Función del slider del módulo Color

## 5.5 Blending

El módulo Blending está diseñado para aplicar efectos a imágenes o videos. Puede haber una o dos imágenes o uno o dos videos de entrada. El módulo dispone de un slider horizontal que controla la cantidad de efecto aplicado y 20 botones que controlan el tipo de efecto que se aplica a los elementos de entrada. También dispone de dos pantallas de entrada y una pantalla de salida para visualizar los elementos y el efecto final generado.

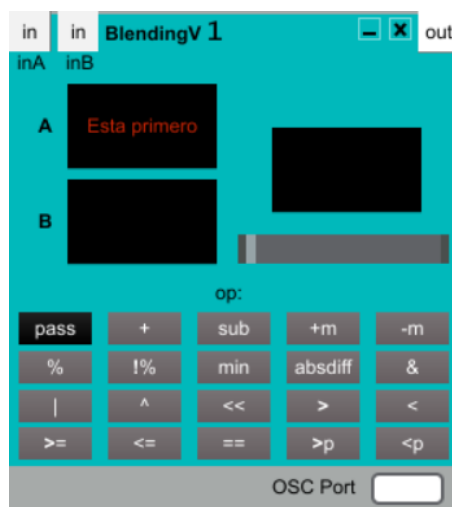


Ilustración 44: Módulo Blending



Al igual que el módulo Color, este no tiene ningún elemento para poder cargar las imágenes desde el propio módulo, pero cuenta con dos entradas “in” y una salida “out” situadas en las esquinas de la parte superior. Este módulo se tiene que conectar con otro para poder cargar las imágenes o videos y las pantallas A y B muestran el contenido de cada elemento. En cuanto a la salida “out” se cargan los datos del elemento final con los efectos aplicados y esto aparecerá en la pantalla de la derecha.

El diseño para la aplicación móvil se ha realizado teniendo en cuenta sólo el slider y los botones, las pantallas se han eliminado porque no son elementos para una superficie de control y el usuario no tiene que ver el resultado en el dispositivo sino en una pantalla conectada con el ordenador.

El slider se ha dispuesto horizontalmente en la parte superior del panel y por debajo se ha creado un panel asociado a una cuadrícula de 4 filas por 5 columnas con los 20 botones tal y como se muestra en la imagen:



Ilustración 45: Superficie de control del módulo Blending

Se han mantenido los colores de la aplicación de MAX para el ordenador, un panel de fondo de color azul, el slider con un color gris y los botones con un color gris como color por defecto y un color negro para cuando son seleccionados.

#### *Funcionamiento de la superficie de control*

Como ya hemos dicho, con el slider controlamos la cantidad de efecto añadido y con los botones la operación que se le aplica a los elementos seleccionados. El efecto viene descrito con una etiqueta de texto encima de cada botón para que el usuario pueda elegir correctamente su opción.

En la programación del algoritmo de los botones se utilizó el mismo script que el ImageSampler porque la funcionalidad es la misma, sólo puede activarse un único botón y este se debe quedar coloreado hasta que se pulse otro diferente.

#### *Elementos de control*

Se ha dispuesto un slider horizontal con un fondo blanco, la fill area tiene un color azulado y el handle es rectangular y de color negro.



Ilustración 46: Slider del módulo Bleding

Los botones tienen un color gris y cuando se seleccionan pasan a estar de color negro.

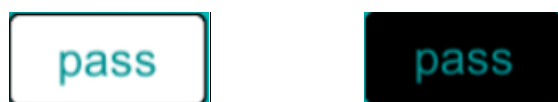


Ilustración 47: Blending a) botón en el estado “normal” b) botón en el estado “pressed”

### Mensajes OSC

Los mensajes OSC asociados a este panel tendrán el “10” como número de identificación.

BLENDING			
Number of module	Element	Value	OSC Message
10	fader1	$0 \leq X \leq 1$	/10/fader1 0.548
10	buttonPad	0 or 1	/10/buttonPad 0
10	buttonPad	0 or 2	/10/buttonPad 0
10	buttonPad	0 or 3	/10/buttonPad 0
10	buttonPad	0 or 4	/10/buttonPad 0
10	buttonPad	0 or 5	/10/buttonPad 0
10	buttonPad	0 or 6	/10/buttonPad 0
10	buttonPad	0 or 7	/10/buttonPad 0
10	buttonPad	0 or 8	/10/buttonPad 0
10	buttonPad	0 or 9	/10/buttonPad 0
10	buttonPad	0 or 10	/10/buttonPad 0
10	buttonPad	0 or 11	/10/buttonPad 0
10	buttonPad	0 or 12	/10/buttonPad 12
10	buttonPad	0 or 13	/10/buttonPad 0
10	buttonPad	0 or 14	/10/buttonPad 0
10	buttonPad	0 or 15	/10/buttonPad 0
10	buttonPad	0 or 16	/10/buttonPad 0
10	buttonPad	0 or 17	/10/buttonPad 0
10	buttonPad	0 or 18	/10/buttonPad 0
10	buttonPad	0 or 19	/10/buttonPad 0
10	buttonPad	0 or 20	/10/buttonPad 0

Tabla 5: Mensajes OSC del módulo Blending

En el Blending se dispone de un “slider” horizontal, su valor está comprendido entre 0 y 1, y 20 botones. Estos botones están configurados de una manera diferente al resto porque todos ellos tienen la misma etiqueta de nombre para los mensajes OSC y lo único que los diferencia es el valor que se envía cuando este es pulsado. Se ha optado por hacer como valor de activación asociado al mensaje “OSCSendMessageDown” el número correspondiente al botón en el diseño.

## 5.6 Panel conexión

El panel conexión no es una superficie de control porque no envía mensajes OSC ni dispone de un módulo análogo creado en el Launcher de MAX. Este elemento ha sido diseñado y programado exactamente igual que en la otra aplicación referente a los bloques de audio. Su principal función es recoger los datos de conexión que introduce el usuario y vincularlos a los objetos. También provee de la información con los textos referentes al equipo que ha creado la aplicación.

La pantalla se ha diseñado en dos secciones. En la parte izquierda del panel tenemos una imagen con el logo, la versión de la aplicación, un enlace a la página oficial y una pequeña descripción del sistema. En la parte derecha tenemos las entradas de texto de IP y Puerto, el toggle de visualización de mensajes OSC, el botón para iniciar la comunicación y los créditos referentes a *Soundcool*.

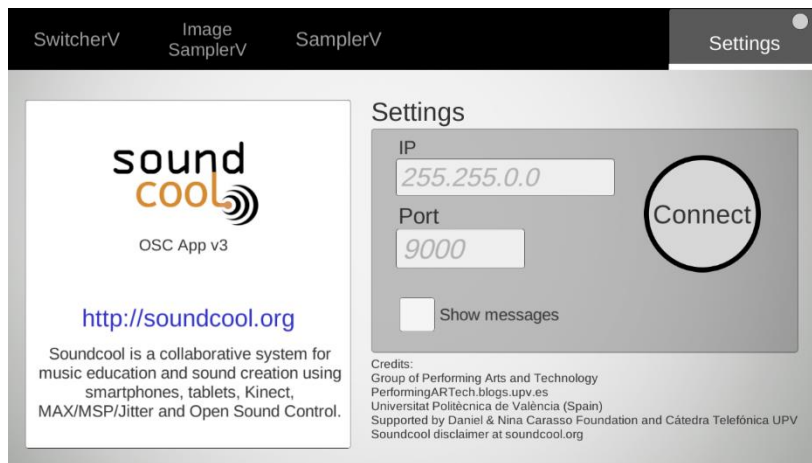


Ilustración 48: Panel del menú conexión

### Funcionamiento del panel

Antes de nada hablaremos de cómo funciona este panel de cara al público. El supervisor facilitará las direcciones de IP y Puerto en red del ordenador donde se ejecuta el Launcher. El usuario introducirá cada dato en su correspondiente campo y luego pulsará el botón “Connect” para realizar la conexión.

El script llamado “UniOSCConnection”, perteneciente a la librería UniOSC, es el encargado de recoger los datos de conexión y procesarlos. También genera y envía los mensajes OSC correspondientes a los botones, sliders y toggles. El problema que apareció con este script es que los datos se introducían manualmente en la programación. Debido a este contratiempo se tuvieron que programar varios códigos en diferentes partes de la aplicación:

- **SC IPLabel (Script):** contiene los objetos de tipo “TextField” referentes a la IP y el Puerto. Un TextField es una caja de texto para introducir por teclado los datos, símbolos y palabras. Luego tenemos “PlaceholderTextIP” y “PlaceholderTextPort” que son etiquetas de texto que muestran en la pantalla los datos introducidos en el TextField. Por último, referencia al objeto “OSCConnection” al cual le envía los datos introducidos en cada campo de texto. Este script también tiene la función de cargar datos almacenados en el caso de que la aplicación hubiese sido utilizada anteriormente. Por el contrario carga valores por defecto.
- **sc\_checkConnection (Script):** este código está relacionado con la animación del color de los objetos botón “Connection” y “Status Image”. Primero se hace un estudio de la conexión para saber si es buena, aceptable o nula. En este estudio se tienen en cuenta

parámetros como tiempo de respuesta a un “ping”, si la conexión ha fallado, el wifi está desactivado, etc. Después se colorean los dos objetos según el estado.

- sc\_changeoscOutIPAdress (función en el script OSCConnection): la función carga cada vez que se ejecuta un nuevo valor de “IP”.
- sc\_changePort (función en el script OSCConnection): esta función realiza la misma acción que la anterior, pero en este caso con el dato “Puerto”.

En resumen, se han tenido en cuenta todos los elementos anteriores para lograr que el usuario pueda introducir manualmente los datos de conexión y actualizarlos cuando lo crea conveniente. Existirá una relación entre los elementos “InputField IP”, ingresar datos, y la función sc\_changeoscOutIPAdress”, que actualizará la conexión con el nuevo valor. Otro ejemplo sería la acción de pulsar el botón “Connect” e inmediatamente ejecutar el script “sc\_checkConection”, encargado de analizar el estado de la conexión. Todas estas relaciones se muestran en el algoritmo de la ilustración 49.

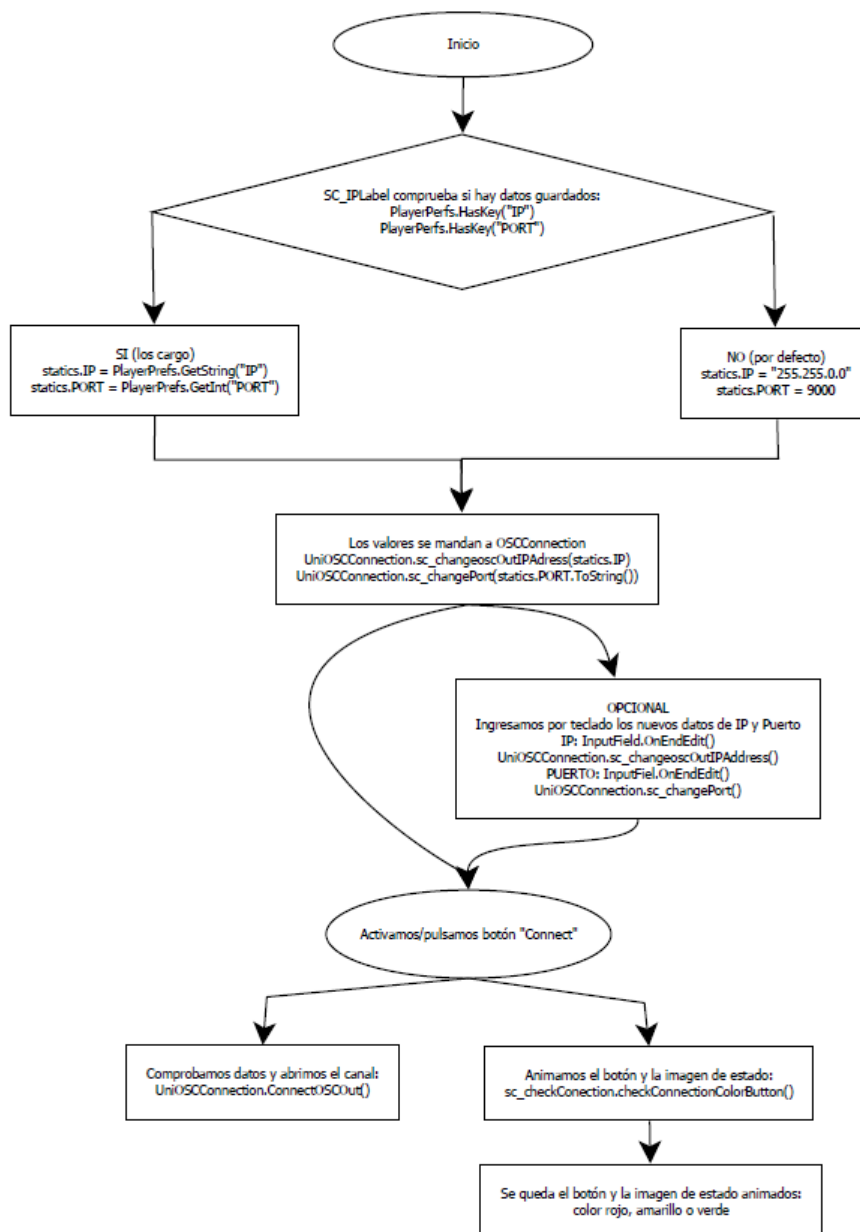


Ilustración 49: Algoritmo del menú conexión

Opcionalmente, se dispone de un toggle con la etiqueta “Show messages” para hacer visibles los mensajes OSC que se envían en la transmisión. En este mensaje se puede ver la ruta del objeto activado y el valor que tiene.

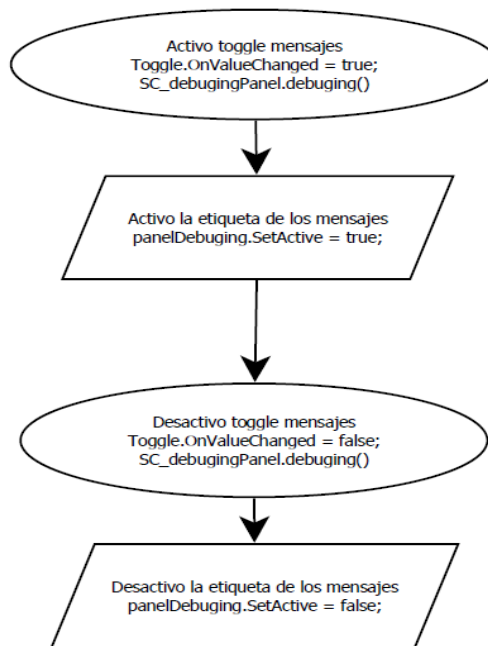


Ilustración 50: Algoritmo para mostrar el texto de los mensajes OSC del panel conexión

### Elementos de control

Dispondremos de un botón circular con borde ancho negro y fondo blanco. Tendrá una etiqueta de texto con la palabra “Connect” para que el usuario identifique rápidamente su función. También contará con una animación del color de fondo dependiendo del estado de la conexión.




	<p><b>Estado correcto:</b> color verde y significa que la conexión ha sido chequeada correctamente y funciona. Se ha recibido la contestación al “ping” enviado dentro de los tiempos correctos.</p>
	<p><b>Estado semi-correcto:</b> se anima de color amarillo y representa que se ha recibido contestación al “ping” pero no en el tiempo correcto. A pesar de todo, la conexión puede funcionar.</p>
	<p><b>Estado incorrecto:</b> el botón queda rojo significando que no se ha podido establecer la conexión. Las consecuencias posibles son que el router esté apagado y no exista ninguna red, que no se tenga el Wifi encendido, o que algún dato ingresado es incorrecto.</p>

Tabla 6: Estados del botón connect

Por otro lado también tenemos las llamadas “cajas de texto”. Se trata de campos de entrada de datos para la IP y Puerto. A pesar de ser entradas de texto, si no se generase un objeto de tipo texto y se vinculara a ellas, el usuario no podría ver los datos escritos. El diseño es rectangular y el tamaño ha sido ajustado para que se muestre todo el texto correctamente dentro de cada caja.

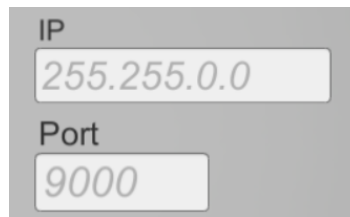


Ilustración 51: Cajas de texto de los elementos de conexión

Por último tenemos el toggle, de aspecto cuadrado y la checkmark se representa con un tick afirmativo.

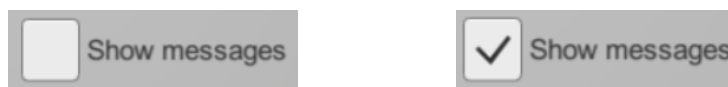


Ilustración 52: Toggle activador de la visualización de los mensajes OSC

La función de este objeto es mostrar el mensaje OSC enviado al ordenador. Este texto ha sido ubicado en la parte superior izquierda y estará presente en todos los paneles como se muestra en la ilustración 53.

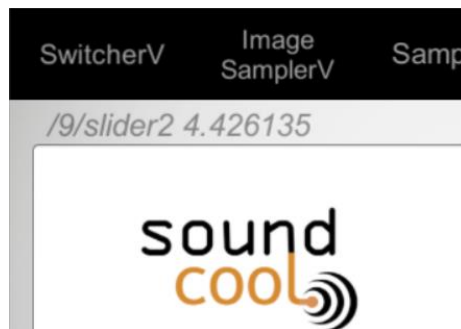
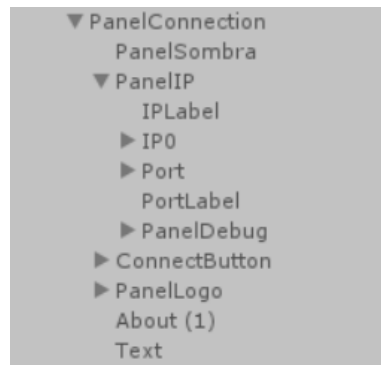


Ilustración 53: Vista de los mensajes OSC en la aplicación

La disposición de los objetos en la jerarquía es muy importante cuando se realiza el diseño. Se vinculan todos los objetos a su “padre”, es decir, el objeto que los contiene. También es importante situar en la parte inferior los elementos de mayor importancia, los visibles en todo momento. Se han creado dos paneles principales, uno con los textos e icono, y otro con los objetos de la conexión. El elemento “texto” de los mensajes OSC se ha ubicado al final de la jerarquía para que quede visible en todo momento.



**Ilustración 54: Jerarquía del panel de conexión**

## Capítulo 6. Diseño de la aplicación.

Una vez diseñadas y probadas todas las superficies de control, en este capítulo explicaremos cómo se han organizado en una misma aplicación. Se detallará el funcionamiento de la barra de menú superior.

La idea que se desarrolla es la utilizada en la aplicación anterior de *Soundcool*, por su buen funcionamiento y sencillez. Se pretende respetar la apariencia entre las distintas versiones para que el usuario no encuentre grandes diferencias y pueda utilizar ambas sin complicaciones. Las superficies de control de ambos bloques también guardan relación.

Esta idea parte de tener juntos todos los paneles en un mismo proyecto y la navegación se realiza con una barra superior de toggles. Se han dispuesto 6 toggles (uno por cada superficie de control) y se les ha dado la propiedad de “Group of toggles”. Con este tipo de propiedad logramos que solo pueda estar un único toggle activado. De esta manera, como tenemos vinculados los objetos a la superficie correspondiente, cuando se activen se mostrará el panel indicado y el resto permanecerá invisible.

Los toggles son rectangulares, están animados y cuando se activan se muestran en el color referente de la superficie de control que hace visible. La “checkmark” es una barra en color blanco que se queda visible en la parte inferior, indica que elemento esta seleccionado.

Conviene subrayar que con este tipo de programación conseguimos tener una aplicación intuitiva, funcional y con una rápida respuesta. A parte, se logra guardar el estado de los elementos para cuando el usuario decida volver a la misma superficie de control. También, centramos la atención en el panel de conexión porque si no se rellenan los datos de conexión el sistema no funcionará.

### 6.1 Exportar la aplicación desde Unity3D para Android

La aplicación se genera desde el propio Unity3D sin necesidad de software adicional. El proceso empieza por buscar en el menú de programa “File” la opción “Build Setting”. Se abrirá una ventana como la que se muestra a continuación:

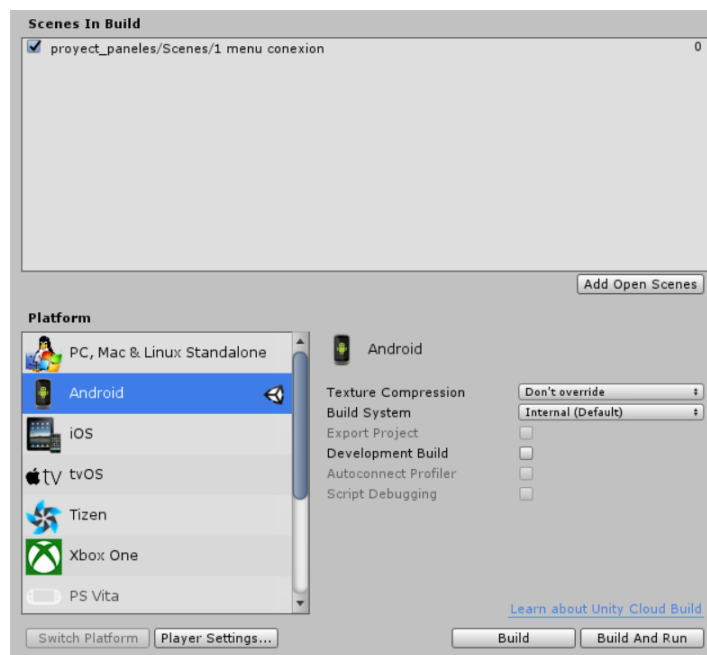


Ilustración 55: Opción exportar escenas de Unity3D



En la parte superior de la ilustración 56 aparecen las escenas del proyecto, en nuestro caso existe únicamente una. Seleccionamos Android como plataforma de salida porque el smartphone con el que realizaremos las pruebas utiliza este sistema operativo. Seguidamente pulsaremos en la opción que aparece en la parte inferior izquierda “Player Settings” y aparecerá una nueva ventana con estas opciones:

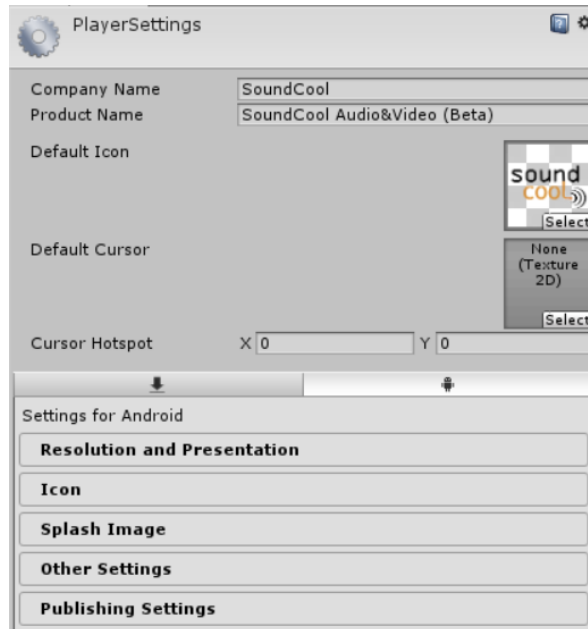


Ilustración 56: Opciones de la aplicación en Unity3D

En este apartado aparecen unos parámetros relacionados con la aplicación y la compañía que procedemos a explicar a continuación:

- **Company Name:** Se trata del nombre de la organización, en nuestro caso Soundcool.
- **Product Name:** El nombre que le pondremos a la aplicación y con el que aparecerá en el menú del móvil.
- **Resolution and presentation:** Esta opción hace referencia a la resolución y la presentación. En nuestro caso queremos que la aplicación tenga una orientación fija de tipo “Landscape Right”. Se elige alineación a la derecha porque después de probarse en Tablets era la mejor opción.
- **Icon:** Es la imagen elegida como icono de la aplicación. Se abre en este menú se generan varios tamaños automáticamente para diferentes perfiles.
- **Splash image:** Hace referencia a la imagen de fondo que se muestra al arrancar la aplicación. En nuestro caso utilizaremos un “Background” referente a Soundcool. También se ha añadido una animación referente al software Unity3D.
- **Other Settings:** En este apartado solo modificaremos la propiedad “Bundel Identifier” que hace referencia a la compañía y la aplicación. El texto se dispone como “com.company.product” y los datos de la empresa y el producto deben ser los mismos que hemos puesto anteriormente. El resultado queda “com.SoundCool.SoundCool Audio&Video”. Las demás opciones se dejarán por defecto. Eso sí, la versión mínima de Android dependerá del SDK que se instaló junto con Unity3D.
- **Publishing Settings:** en este caso pondríamos las claves del producto para subirlo al PlayStore, pero como solo estamos de pruebas se dejará por defecto.

En resumen, se han realizado 2 versiones diferentes, la primera contiene solo los nuevos módulos pertenecientes al bloque de video, y la segunda junta los bloques de audio y video en una misma aplicación.

## 6.2 Versión 1

La primera versión se ha diseñado solo con las superficies de control del bloque de video, tal y como se acordó en la reunión de planificación del proyecto. Se mostrará el panel de conexión por defecto al iniciar la aplicación para que el usuario rellene los campos de IP y Puerto en su primera acción.

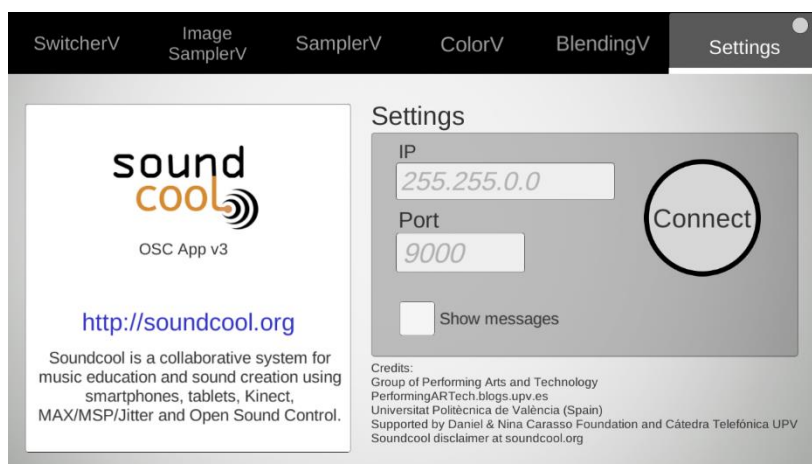


Ilustración 57: Menú principal de la aplicación versión 1

Si nos fijamos en la jerarquía, se han incluido todos los objetos visibles en el elemento “canvas” para poder renderizarse y ser visibles. El orden en el que se han dispuesto las superficies de control es el mismo que las opciones de la barra superior del menú. Se ha situado el panel de conexión en primer lugar porque es el elemento activo por defecto al iniciar la aplicación.

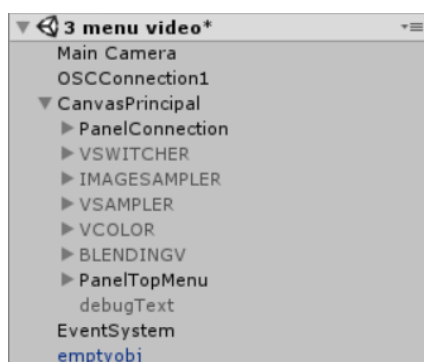


Ilustración 58: Jerarquía de la aplicación versión 1

Finalmente, como había dos aplicaciones de *Soundcool* y que resultase más fácil distinguirlas, aparte de cambiar el nombre también diseñamos diferentes iconos.



Ilustración 59: Varios diseños de iconos para la aplicación a) versión 1 b) versión 2 c) versión 3

Se ha elegido el icono de la ilustración 59 c), después de un debate con los miembros del equipo. Es el que mejor destaca el logo de Soundcool y luego el título referente al bloque en letra pequeña en la parte inferior para que el usuario no se confunda.

### 6.3 Versión 2

La versión definitiva contará con un nuevo panel principal muy parecido a la aplicación anterior con los datos de la organización y los elementos de conexión. A diferencia, el usuario dispondrá de dos botones adicionales, “AUDIO” y “VIDEO” para acceder al menú correspondiente. Los cambios visuales en el diseño se muestran en la ilustración 60.

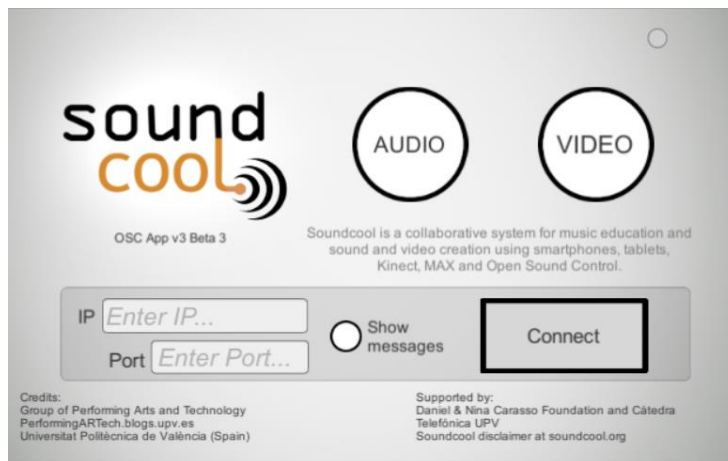


Ilustración 60: Menú principal de la aplicación versión 2

Al acceder a cada bloque, las 5 superficies de control se presentarán con la barra superior de toggles explicada en los apartados anteriores. Además, contaremos con un botón adicional en cada barra para salir al menú principal. Por defecto, al acceder a los submenús tendremos activado el panel “PLAYER/EFFECT” en Audio y el panel “SAMPLER” en Video.

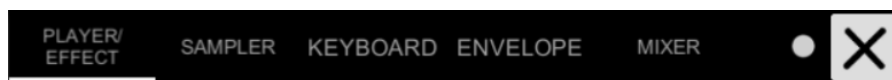


Ilustración 61: Barra superior menú Audio



Ilustración 62: Barra superior menú Video

En cuanto a la jerarquía de esta versión, Figura 63, se han generado 3 objetos de tipo “Canvas”. Uno hace referencia al panel principal de conexión con los botones para acceder a los submenús. En otro hemos puesto todas las superficies de control del menú de Audio. Y en el último hemos puesto todos los objetos referentes al bloque de Video. Se ha seguido un orden lógico, en nuestro

caso, por orden de aparición de los elementos. También se ha generado un elemento llamado “CanvasStatus” que corresponde a la imagen que indica el estado de la conexión (verde, amarillo o roja). Se ha ubicado al final de la jerarquía para que esté visible en todo momento y en todos los paneles.



**Ilustración 63: Jerarquía de la aplicación versión 2**

## Capítulo 7. Comprobación del funcionamiento de la herramienta.

Finalizada y examinada la aplicación en el dispositivo móvil sólo queda por saber si la comunicación OSC funciona correctamente. En este capítulo explicaremos paso a paso como se lleva a cabo el montaje del sistema completo: smartphone y ordeador.

En primer lugar abrimos en el ordenador el Launcher creado en MAX y que está disponible en la página de *Soundcool*. Esta es una aplicación standalone con representaciones simbólicas de los módulos de audio y vídeo, que, si las pulsamos con el ratón, abrirán el módulo correspondiente de *Soundcool*.



Ilustración 64: Launcher de Soundcool para el ordenador

Seguidamente seleccionaremos los nuevos módulos referentes al bloque de video y de los cuales hemos creado las superficies de control. Una vez abiertos, en la parte inferior donde se encuentra la caja de texto de Puerto OSC escribiremos por ejemplo el puerto.



Ilustración 65: Módulos de video de Soundcool

La comunicación OSC precisa tener conectados, tanto computadora como smartphone, a la misma red Wi-Fi. Se utilizará un router o modem router independiente, es decir, sin ningún tipo de servicio contratado para crear la red Wi-Fi. Únicamente conectaremos los dispositivos que se vayan a utilizar para *Soundcool*. A su vez, buscaremos la IP del ordenador dentro de la misma red y será guardada para su posterior utilización.

Ahora, ejecutaremos la APP en nuestro dispositivo móvil y ponemos la dirección IP del ordenador y el puerto OSC. Pulsamos el botón “Connect” y si se muestra de color verde ya tenemos el sistema listo. En caso de que el botón se anime en color amarillo puede funcionar también correctamente pero hay que probarlo para confirmar la conexión.

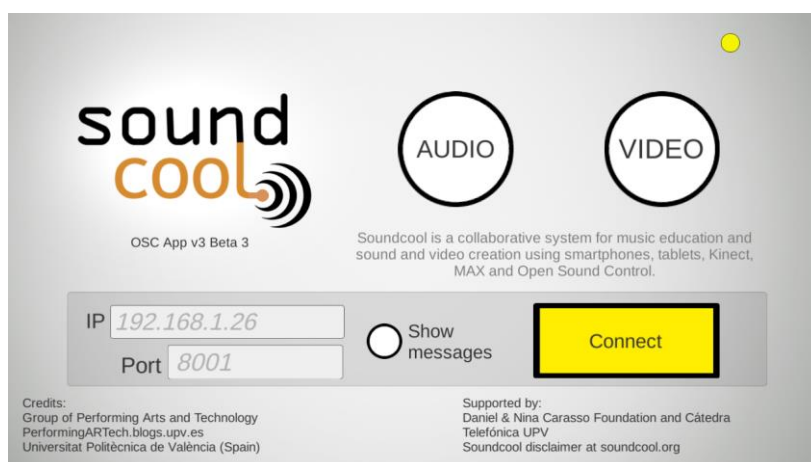
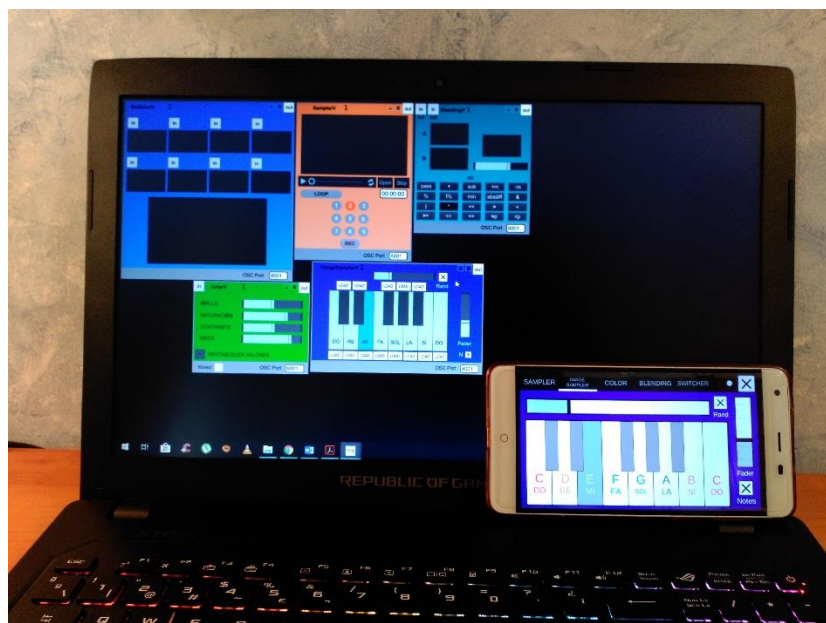


Ilustración 66: Aplicación conectada al ordenador

Finalmente, probamos uno a uno todos los elementos de cada superficie de control y vemos la respuesta inmediata de su análogo en el ordenador.



**Ilustración 67: Sistema completo en funcionamiento**

## Capítulo 8. Conclusiones

En este capítulo se describen las conclusiones finales del trabajo. También se ha añadido un apartado con las competencias transversales adquiridas por el alumno durante el desarrollo del trabajo. Por último, se ha calculado el coste de todos los elementos necesarios para realizar y testear la aplicación.

### 8.1 Generales

Una vez terminada la aplicación para dispositivos móviles se tiene que presentar a los usuarios y comprobar su correcto funcionamiento. Con los nuevos módulos y sus correspondientes superficies de control se pretende potenciar la parte visual colaborativa de *Soundcool*. Así mismo, con estas nuevas opciones se espera aumentar las posibilidades artísticas de los alumnos y crear espectáculos audiovisuales de mayor complejidad. El sistema completo ha sido probado en el laboratorio por otros miembros del equipo *Soundcool* y ha funcionado correctamente. Finalmente el sistema se ha probado en un entorno real de clase por miembros del equipo pedagógico del proyecto con excelentes resultados.

Personalmente, este trabajo ha sido la oportunidad de aplicar todos los conocimientos adquiridos durante el grado desarrollando una aplicación para dispositivos móviles. Durante estos años he ido perfilando mis conocimientos al mundo de la programación utilizando lenguajes como C, Java, Verilog, Arduino (C++) y, ahora, también C#. Pero en este proyecto he conocido un entorno de programación gráfico en el cuál la programación escrita ha quedado en segundo plano. Este software no utiliza casi código en la generación de las funciones porque ya las llevan implementadas los propios objetos. De esta manera se agiliza mucho la creación y presenta muchas más opciones gráficas para el aspecto visual de las aplicaciones.

Por último, la oportunidad de realizar el trabajo junto a un equipo de profesionales ha sido una experiencia muy positiva. El trabajo con un grupo altamente multidisciplinar como el de *Soundcool* me ha proporcionado una mentalidad mucho más abierta y un pensamiento más crítico para realizar proyectos de esta índole. También he aprendido a buscar soluciones y recursos fiables en la web.

### 8.2 Dificultades.

Han sido muchas las dificultades que han aparecido durante la realización del proyecto. Estas dificultades han sido debidas principalmente a que ha sido necesaria una confluencia entre los aspectos pedagógicos, de ingeniería y de usabilidad óptimos a lo largo de su desarrollo. Así, la aplicación se ha ido probando en entornos reales y han sido necesarias diversas modificaciones y mejoras a lo largo del proyecto. Esto también ha retrasado el lanzamiento de la aplicación varios meses.

La mayor dificultad con la que me encontré fue crear el piano del módulo ImageSampler y asociar el algoritmo que controla que el teclado sea monofónico, es decir, que solo se permitiera visualizar una imagen cada vez. Esto contrasta con la superficie de control de audio correspondiente al teclado, que sí que permitía pulsar varias teclas a la vez. Y era una necesidad del equipo pedagógico realizar el módulo ImageSampler en forma de teclado, y por tanto condición de diseño. Así, el diseño se tuvo que hacer en varias partes para tener todas las teclas claras y oscuras con su mismo tamaño respectivo y que el espaciado entre ellas tuviera la misma distancia.

La búsqueda del algoritmo del ImageSampler que cumpliera con la función correcta también fue lenta. Primero se pensó en utilizar un script para cada tecla, de esta manera se aplicaba una



funcionalidad distinta para cada una. Después, pensando en reutilizar código, se estudió en generar solamente 2 scripts, uno para las teclas blancas y otro para las negras. También funcionaba de esta manera por lo que apareció la ambición de programarlo todo utilizando un único script. Finalmente se ha utilizado el mismo script en todas las teclas.

Por otro lado, asociar todos los elementos de conexión del “panel de conexión” también fue complejo. Existen scripts que recogen los datos introducidos por el usuario, otros que se encargan de construir los mensajes OSC, otro que comprueba que la conexión es correcta; y vincular todos los parámetros conlleva un cierto grado de dificultad.

Por último, se ha tenido la dificultad de terminar el proyecto en los plazos definidos inicialmente. Han surgido retrasos tanto en el diseño de los módulos del ordenador, realizados como parte de otro TFG con el que ha habido que coordinarse, como en las superficies de control. Después, también se realizaban pruebas con varias ideas para concluir cual era la mejor opción. Inicialmente se creyó que los módulos se podían terminar en 1 mes cada uno, pero las mejoras en el diseño no lo han permitido.

### 8.3 Competencias transversales aplicadas con la realización del TFG

La UPV ha definido una serie de competencias para evaluar y saber cuál es el nivel del alumno en cada una de ellas. Durante la realización del TFG se han visto mejorado el nivel en cada una de ellas como se describe a continuación:

1. Comprensión e integración: durante el proyecto se ha aprendido la programación en Unity3D de manera autónoma con tutoriales. También se aprendió a realizar pequeños proyectos con el MAX para comprender el funcionamiento de los módulos.
2. Aplicación y pensamiento práctico: se han aplicado los conocimientos adquiridos durante el grado diseñando el plan de acción para realizar la aplicación. También se han extraído conclusiones que han servido para mejorar la aplicación.
3. Análisis y resolución de problemas: durante la programación han aparecido muchos problemas, que se han resuelto de manera autónoma o buscando en caso necesario la ayuda del miembro del equipo indicado.
4. Innovación, creatividad y emprendimiento: todo ello se ha plasmado en la búsqueda de soluciones creativas a los problemas y de forma autónoma. Además la aplicación diseñada es una novedad en el mundo audiovisual y se pretende generar un gran impacto en lo que respecta a la creación colaborativa.
5. Diseño y proyecto: se ha tenido libertad en la creación de las superficies de control. El profesor explicaba la funcionalidad y el alumno generaba las funciones de manera autónoma.
6. Trabajo en equipo y liderazgo: se ha integrado al alumno dentro de un equipo de investigadores. La comunicación ha sido efectiva y se ha realizado un buen trabajo que se va a utilizar en una prueba piloto desde principios del próximo octubre en varias decenas de centros educativos de la Comunidad Valenciana, de acuerdo con Conselleria de Educación.
7. Responsabilidad ética, medioambiental y profesional: siempre se ha sido respetuoso con todos los miembros del equipo cumpliendo los compromisos adquiridos con el proyecto.
8. Comunicación efectiva: se han expresado las ideas y las dudas vía oral y escrita de manera efectiva.

9. Pensamiento crítico: el alumno ha estado centrado en cómo documentar toda la programación utilizando los mínimos recursos y disponiendo toda la información para la gente nueva que colabore en el proyecto.
10. Conocimiento de problemas contemporáneos: en este aspecto se ha realizado la aplicación en distintos dispositivos como smartphones y tabletas, y con Unity3D, una de las plataformas utilizadas en videojuegos más extendidas actualmente; y se han proporcionado diferentes opciones de mejora.
11. Aprendizaje permanente: el alumno ha estado en constante formación y de forma lo más independiente posible para solucionar todos los problemas que han aparecido en la herramienta.
12. Planificación del tiempo: el alumno ha sido capaz de dividir el trabajo en diferentes tareas para después poder alcanzarlas con solvencia y cumplirlas en el tiempo marcado con el equipo de trabajo.
13. Instrumental específica: se han utilizado los conocimientos de programación, de diseño y de desarrollo en la creación de la aplicación aprendiendo lenguajes nuevos y muy diferentes de los estudiados en el grado como son Unity3D y Max. También se han utilizado dispositivos como el ordenador, router, tablet y smartphone y se ha programado una aplicación específica que se hallará en breve en el Google Play Store para su uso público.

En cuanto al desarrollo personal, he mejorado en la búsqueda de nuevas ideas para proyectos relacionados con el TFG, he podido desarrollar mis ideas propias, ser reflexivo con mi trabajo, en cierta manera ser autónomo en la manera de programación pero realizar trabajo conjunto con mis compañeros para hacer el diseño final de la aplicación con consenso, conocer mi potencial frente a una situación desconocida y cómo puedo lograr las diferentes metas que se presentan en el proyecto.

#### 8.4 Presupuesto

Una parte importante de los proyectos es tener en cuenta el presupuesto, tanto de los elementos necesarios como del sueldo de la persona que lo ha realizado. En este proyecto se han utilizado los dispositivos electrónicos como ordenador portátil, smartphone y Tablet. En la tabla 7 se detalla el precio de cada uno junto con su vida útil y la amortización.

	Precio (€)	Vida útil	Amortización por año (€)
Ordenador Windows	1260	6	168
Móvil Android	200	4	40
Tablet Android/Windows	230	4	46
<b>Total</b>	1690		254

Tabla 7: Presupuesto

En referencia al sueldo de un ingeniero de telecomunicaciones acudimos a los informes del colegio oficial de ingenieros de telecomunicación [30]. Un ingeniero con 5 o menos años de experiencia tiene un sueldo anual de 25731€, teniendo en cuenta nuestro caso de 2 años, el importe total asciende a 51462€. Este salario me parece excesivo ya que no he dedicado 8h/día exclusivamente a la programación de la app durante estos dos años. Siendo honesto y contando con las mejoras sucesivas de la aplicación, el trabajo realizado sería de unos 6 meses a tiempo completo.

## Capítulo 9. Trabajo futuro

En este capítulo se expondrán las futuras mejoras del proyecto con las ideas que van surgiendo influenciadas por las nuevas tecnologías. Como se trata de un proyecto real en el ámbito de la educación estará en constante actualización y siempre será necesario contar con programadores.

### 9.1 Efectos Blending

El módulo Blending en MAX dispone de una cuadrícula de 20 botones vinculados a 20 efectos de imagen. Realmente hay programados 66 efectos y se requieren modificaciones en el diseño para poder elegir cualquier efecto. La idea que se está desarrollando es vincular el último botón de la cuadrícula a un menú desplegable de manera que se pueda seleccionar un efecto y al pulsar el botón se aplique sobre la imagen o vídeo. En el caso de cambiar a otro efecto lo seleccionaríamos del menú desplegable y ya estaría vinculado al último botón.



Ilustración 68: Módulo Blending con todas las opciones

En cuanto al diseño de la superficie de control del Blending quedaría igual, no se tendrían que añadir o quitar elementos. Solamente se cambiaría la etiqueta de texto del último botón por “menú” o “opción”.



Ilustración 69: Superficie de control del Blending actualizada

## 9.2 Painter

Actualmente se está desarrollando un nuevo módulo en MAX llamando “Painter”. Este módulo funciona como una herramienta de dibujar trazos sobre una imagen cargada por el usuario desde el PC y que se visualiza en una pantalla. La idea es que el usuario pueda “pintar” desde el smartphone sobre la imagen proyectada, utilizando líneas de diferentes colores y tamaños.

La idea inicial es hacer un módulo que disponga de un área dibujar los trazos. Se han incluido 3 opciones iniciales, la opción de borrar todo el contenido con un botón, la selección del tamaño del trazo y el color a través de sliders. El funcionamiento es sencillo, el programa captura la posición del toque y con una función específica dibuja el trazo teniendo en cuenta los parámetros de tamaño y color. La selección del color se realiza por la composición RGB, se debe decidir si se quiere poner un único slider en la superficie de control o los 3.

La superficie de control dispondría de una superficie de detección del toque delimitada visualmente, de un botón “clear” para borrar los trazos realizados y dos sliders verticales, uno de selección de color (o 3 para el formato RGB) y otro para el tamaño del trazo.

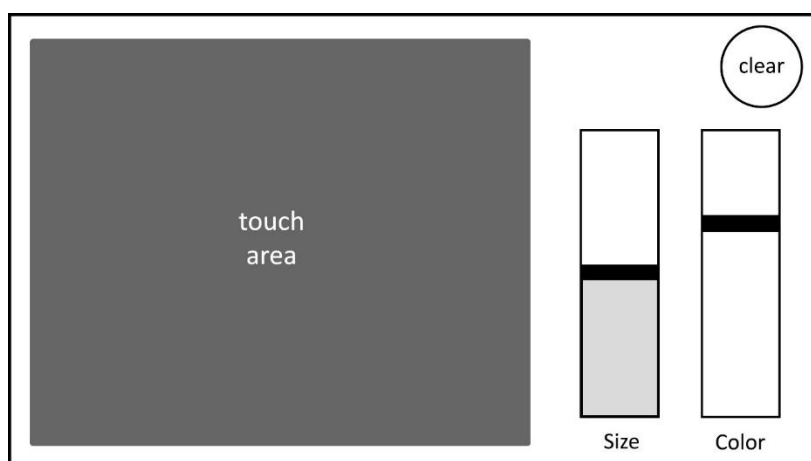


Ilustración 70: Boceto superficie de control del Painter

## 9.3 Módulos de entrada de video y audio

Un colaborador del equipo *Soundcool* ha estado trabajando en un módulo para la captura de video en tiempo real desde el dispositivo móvil. En este caso la idea sería realizar una superficie de

control relacionada con este módulo con varias opciones. También, realizar un sondeo para generar otro módulo de captura de sonido desde el propio smartphone. A modo de ejemplo ya he programado una aplicación que utiliza el micrófono del dispositivo y convierte el audio en digital para poder enviarlo al ordeandor mediante el protocolo OSC.

Por un lado, la superficie de control para la captura de video tendría estos elementos de control: dos botones principales, uno de “captura” y otro de “stop”, y varios sliders verticales para opciones como zoom, brillo, contraste, etc.

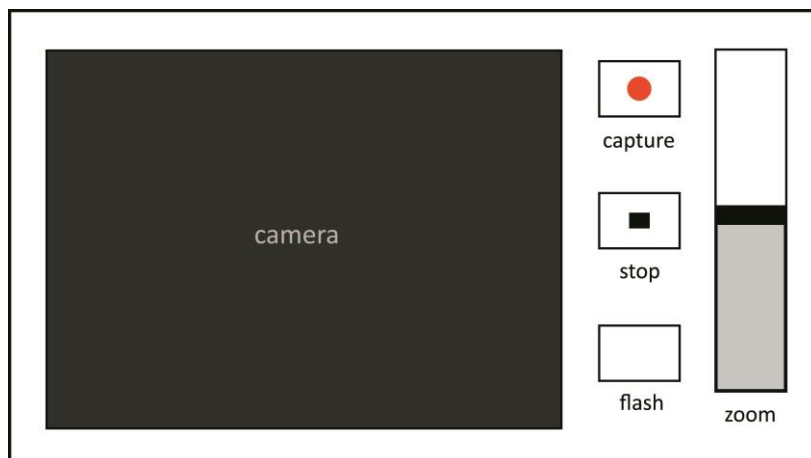


Ilustración 71: Boceto de la superficie de control del módulo cámara

Por otro lado, la superficie de control para el módulo de audio contaría también con los botones “captura” y “stop” respetando la funcionalidad y el aspecto hacia el usuario. Dispondría de una pantalla donde podría visualizarse por ejemplo el espectro del sonido dado que hay un módulo spectroscopio para visualizar espectros ya en *Soundcool*; y un slider vertical para el control del volumen.

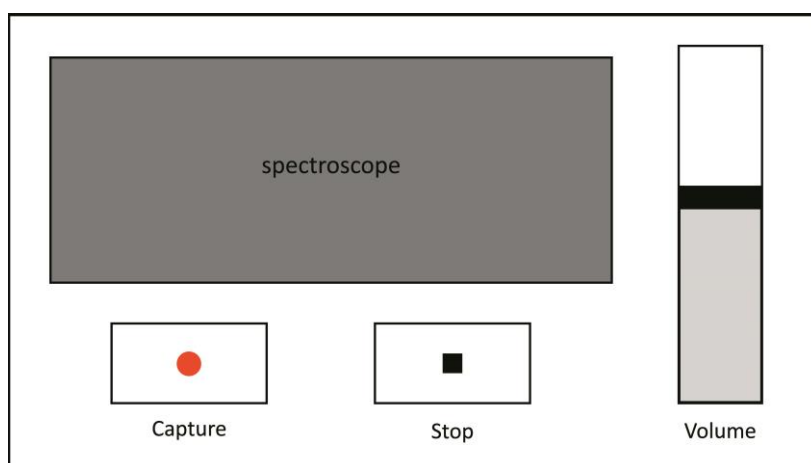


Ilustración 72: Boceto de la superficie de control del módulo micrófono

#### 9.4 Creación de tutoriales para los cursos de *Soundcool*

El equipo de *Soundcool* está realizando cursos online de formación con la herramienta desde la plataforma de la UPV. El bloque de audio que ya está en funcionamiento tiene disponibles varias guías para el alumno. La idea sería recopilar toda la información sobre los nuevos bloques y

diseñar un programa educativo para la plataforma. Estos tutoriales deberían ser perfilados tanto por los pedagogos como por los programadores que han desarrollado el sistema.

### 9.5 Control de la iluminación con DMX

En la expansión visual de *Soundcool* se han generado nuevos módulos con efectos de imagen y video. Personalmente, pienso que la iluminación del escenario en un concierto en directo también es un aspecto visual importante. Por lo tanto, la idea de este módulo sería imitar una consola para el control de la iluminación. Desde el ordenador se crearían las escenas con las luces disponibles y en el dispositivo móvil se tendría la superficie de control.

Para la creación de este módulo se utilizarían las funciones del programa MAX/MSP/JITTER asociadas al protocolo DMX [31]. También se necesitaría adquirir una interfaz USB con control DMX. Este dispositivo electrónico es necesario para convertir las instrucciones del ordenador y poder enviarlas por el cable de 3 pines que utilizan todos los sistemas DMX.

La superficie de control se diseñaría con botones para cargar una escena concreta y dispondría de sliders con los diferentes parámetros variables de cada escena. Los 4 efectos pensados son: modificación del tiempo de transición entre escena, el color de las luces, el tiempo de repetición de las escenas y el máster que controlaría la cantidad total de luz.

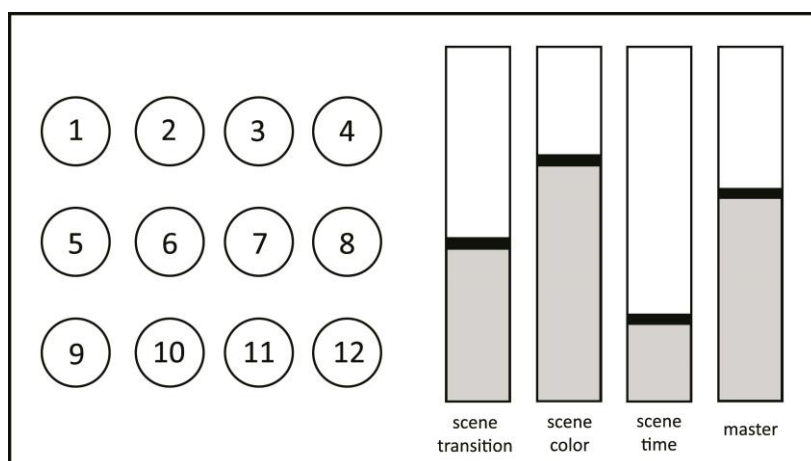


Ilustración 73: Boceto de la superficie de control del módulo DMX

## Capítulo 10. Referencias

- [1] «Soundcool,» [En línea]. Available: <http://soundcool.org>.
- [2] «DAW,» [En línea]. Available: <https://es.steinberg.net/es/productos/cubase/inicio.html>.
- [3] «Cycling74,» [En línea]. Available: <https://cycling74.com/>.
- [4] «PureData,» [En línea]. Available: <https://puredata.info/>.
- [5] «Unity,» [En línea]. Available: <https://unity3d.com/es>.
- [6] «Android Studio,» [En línea]. Available: <https://developer.android.com/studio/?hl=es-419>.
- [7] «Eclipse,» [En línea]. Available: <https://www.eclipse.org/eclipse/>.
- [8] «TouchOSC,» [En línea]. Available: <https://hexler.net/software/touchosc>.
- [9] «Soundcheck,» [En línea]. Available: <https://soundcheck.com.mx/superficies-de-control/>.
- [10] «Akai,» [En línea]. Available: <http://www.akaipro.com/products/pad-controllers>.
- [11] «Novation,» [En línea]. Available: <https://global.novationmusic.com/keys/launchkey#>.
- [12] «Avid pro tools,» [En línea]. Available: <https://www.avid.com/es/pro-tools>.
- [13] «Yamaha,» [En línea]. Available: [https://es.yamaha.com/es/news\\_events/2017/1215\\_50\\_stagemix\\_v7.html](https://es.yamaha.com/es/news_events/2017/1215_50_stagemix_v7.html).
- [14] «Kinect,» [En línea]. Available: <https://support.xbox.com/es-CO/browse/xbox-360/accessories/Kinect>.
- [15] «Creación musical con Soundcool,» [En línea]. Available: <https://www.upvx.es/courses/course-v1:soundcool+soundcool101x+2018-01/about>.
- [16] «Open Sound Control,» [En línea]. Available: <http://opensoundcontrol.org/implementation/osc-simpl-unity>.
- [17] «MEVIC,» [En línea]. Available: <http://musicaelectronica.blogs.upv.es>.
- [18] «Diario ABC,» [En línea]. Available: [https://www.abc.es/espana/comunidad-valenciana/abci-mas-4000-alumnos-baten-record-guinness-mayor-leccion-musica-historia-alicante-201805131657\\_noticia.html](https://www.abc.es/espana/comunidad-valenciana/abci-mas-4000-alumnos-baten-record-guinness-mayor-leccion-musica-historia-alicante-201805131657_noticia.html).
- [19] Vacaniti, Daniel; "The Kanban Guide for Scrum Teams". 2018.
- [20] Janoff, N.S.; Rising, L. "The Scrum Software Development Process for Small Teams". 2000.
- [21] Sutherland, Jeff; Schwaber, Ken. "Scrum Guides". 2013.
- [22] «Diagrama de Gantt,» [En línea]. Available: <https://www.obs-edu.com/es/blog-project-management/diagramas-de-gantt/que-es-un-diagrama-de-gantt-y-para-que-sirve>.
- [23] «Cloud build,» [En línea]. Available: <https://unity3d.com/es/unity/features/cloud-build>.

[24] «Arduino,» [En línea]. Available: <https://www.arduino.cc/>.

[25] «OpenGL,» [En línea]. Available: <https://www.opengl.org/>.

[26] «Visual Studio Basic,» [En línea]. Available: <https://www.visualstudio.com/es/>.

[27] «UniOSC scripts,» [En línea]. Available: <http://uniosc.monoflow.org/>.

[28] «Unity Button,» [En línea]. Available: <https://docs.unity3d.com/es/current/Manual/script-Button.html>.

[29] «Unity Slider,» [En línea]. Available: <https://docs.unity3d.com/es/current/Manual/script-Slider.html>.

[30] «Coit,» [En línea]. Available: <https://www.coit.es/informes/el-ingeniero-de-telecomunicacion-perfil-socio-profesional/acceso-al-informe>.

[31] «Protocolo DMX,» [En línea]. Available: <http://www.etsist.upm.es/estaticos/ingeniatic/index.php/tecnologias/item/558-protocolo-dmx-digital-multiplex.html>.