



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

— **TELECOM** ESCUELA
TÈCNICA **VLC** SUPERIOR
DE **UPV** INGENIEROS
DE TELECOMUNICACIÓN

EXPANSIÓ DEL SISTEMA SOUNDPOOL PER A DONAR SUPORT A VIDEOCREACIÓ COL·LABORATIVA

Elena Rita Pelejero Ibáñez

Tutor: Ximo Cerdà Boluda

Cotutor: Jaime Serquera Peyró

Treball de Fi de Grau presentat a l'Escola Tècnica Superior d'Enginyers de Telecomunicació de la Universitat Politècnica de València per a l'obtenció del Títol de Graduat en Enginyeria de Tecnologies i Serveis de Telecomunicació.

Curs 2017-18

València, 3 de desembre de 2017

Resum

La creació col·laborativa és el contingut més revolucionari de la cultura digital. Consisteix en el desenvolupament de pràctiques creatives participatives enteses com a activitat social i cultural i com a forma de gestió i producció descentralitzada, trencant així amb les jerarquies i els models dominants en els sistemes culturals tradicionals. El que la distingeix de la cultura tradicional és la possibilitat de construir una cultura realment col·lectiva.

Soundcool és un sistema que permet apropar aquesta forma de veure el procés creatiu a una gran diversitat de públic. A grans trets consisteix en un conjunt de mòduls interconnectables capaços de sintetitzar i transformar qualsevol tipus de so a temps real entre molts individus. El sistema *Soundcool* és un sistema d'educació i creació col·laborativa basat en la interacció entre mòbils, tauletes gràfiques i Kinect amb un ordinador central a temps real.

Al llarg d'aquest treball es proposaran quatre dissenys de nous mòduls per a complementar el sistema *Soundcool* amb una nova funcionalitat enfocada en la vídeo creació. Aquests quatre mòduls s'ocuparan de la càrrega de vídeos, el canvi dels seus colors, el canvi de la seua reproducció i la visualització del seu contingut. D'aquesta manera no només es podrà crear col·laborativament contingut àudio sino que a més al mateix temps es podrà crear contingut visual amb la mateixa metodologia.

Resumen

La creación colaborativa es el contenido más revolucionario de la cultura digital. Consiste en el desarrollo de prácticas creativas y participativas entendidas como actividad social y cultural y como forma de gestión y producción descentralizada, rompiendo así con las jerarquías y los modelos dominantes en los sistemas culturales tradicionales. Lo que la distingue de la cultura tradicional es la posibilidad de construir una cultura realmente colectiva.

Soundcool es un sistema que permite acercar esta forma de ver el proceso creativo a una gran diversidad de público. A grandes rasgos consiste en un conjunto de módulos interconnectables capaces de sintetizar y transformar cualquier tipo de sonido a tiempo real entre muchos individuos. El sistema *Soundcool* es un sistema de educación y creación colaborativa basado en la interacción entre móviles, tablets y Kinect con un ordenador a tiempo real

A lo largo de este trabajo se proponen cuatro diseños de nuevos módulos para complementar el sistema *Soundcool* con una nueva funcionalidad enfocada en la video creación. Estos cuatro módulos se ocuparán de la carga de vídeos, el cambio de sus colores, el cambio en su reproducción y la visualización del contenido. De esta forma no sólo se podrá crear colaborativamente contenido audio sino que además al mismo tiempo se podrá crear contenido visual con la misma metodología.

Abstract

Collaborative creation is the most revolutionary content in digital culture. It consists in the development of creative and participative practices as social and cultural activity and as form of management and decentralized production, breaking this with the hierarchies and the dominant models in the traditional cultural systems. What distinguishes it from the traditional culture is the possibility of constructing a real collective culture.

Soundcool is a system that allows to bring over this way of seeing the creative process to a wide range of people. It consists of a set of interconnectable modules capable of synthesizing and transforming any type of sound in real time among many individuals. *Soundcool* is a system of educational and collaborative creation based on the interaction among mobile phones, tablets and Kinect with a computer in real time.

Throughout this work, four new modules have been designed in order to complement Soundcool system with a new functionality focused on video creation. These four modules will deal with video loading, color changing, the change in its reproduction and the visualization of the content. In this way not only content will be able to create audio collaboratively but in addition in addition it will be possible to create visual content with the same methodology.

INDEX

Capítol 1.	Introducció	7
1.1	Objectius	7
1.2	Motivació	10
Capítol 2.	Context	13
2.1	Estat de l'art de la creació col·laborativa	13
2.2	Exemples previs en l'àmbit de la creació col·laborativa	15
2.2.1	The File Room.....	15
2.2.2	Life in a Day.....	16
2.3	Programari emprat per a desenvolupar <i>Soundcool</i>	16
2.3.1	Max/Msp/Jitter	16
2.3.2	<i>Open Sound Control</i>	19
2.3.3	Unity.....	19
2.4	El sistema <i>Soundcool</i>	20
2.1.1	Moduls que introdueixen el so.....	21
2.4.1	Mòduls que modifiquen el so	24
2.1.2	Mòduls que analitzen el so	27
2.1.3	Mòduls que mesclen, graven o reproduïxen el so	28
Capítol 3.	Metodologia	30
3.1	Distribució de tasques	30
3.1.1	Funcionament bàsic de Max i <i>Soundcool</i>	30
3.1.2	Millores i correcció d'errors.....	32
3.1.3	Connexió entre vídeos	33
3.1.4	Exclusivitat de connexió àudio-àudio i vídeo-vídeo	33
3.1.5	Vscreen.....	33
3.1.6	Vswitcher	33
3.1.7	VSampler.....	34
3.1.8	VColor.....	34
3.2	Diagrama temporal.....	35
Capítol 4.	Desenvolupament	36
4.1	Aprenentatge	36
4.1.1	<i>Soundcool</i> i Max.....	36
4.1.2	Correcció d'errors	42
4.2	Solucions comunes a tots els mòduls	45
4.2.1	Interconnexió entre mòduls de vídeo	45

4.2.2	Exclusivitat de connexió vídeo-vídeo i àudio-àudio	47
4.3	Disseny de nous mòduls	48
4.3.1	VScreen	49
4.3.2	VSwitcher.....	50
4.3.3	VSampler.....	52
4.3.4	VColor.....	56
Capítol 5.	Resultats/Test/Pressupost	58
5.1	Correccions, resultats i pressupost.....	58
5.2	Resultats	60
5.2.1	Gent familiaritzada amb <i>Soundcool</i>	60
5.2.2	Gent externa a <i>Soundcool</i>	61
5.3	Pressupost.....	62
Capítol 6.	Conclusions i treball futur	63
6.1	Conclusions	63
6.2	Treball futur.....	63

TAULA D'IL·LUSTRACIONS

Il·lustració 1: Funcionament de <i>Soundcool</i> a un ordinador i control amb mòbils	7
Il·lustració 2: El sistema <i>Soundcool</i> a les aules	8
Il·lustració 3: Representació de La Mare dels Peixos a València	9
Il·lustració 4: Utilització de <i>Soundcool</i> a mòbils i tauletes gràfiques	10
Il·lustració 5: Pàgina d'inici de The File Room.....	15
Il·lustració 6: Life in a Day, exemple de pel·lícula col·laborativa.....	16
Il·lustració 7: Entorn de Max	17
Il·lustració 8: Patcher i subpatcher	17
Il·lustració 9: Max en mode edició (esquerra) i en mode execució (dreta).....	18
Il·lustració 10: Exemple de ús de MSP (esquerra) i de Jitter (dreta)	18
Il·lustració 11: Creació de l'app per a Android i iOS en Unity.....	20
Il·lustració 12: Player	21
Il·lustració 13: SamplePlayer	22
Il·lustració 14: Direct Input.....	22
Il·lustració 15: Signal Generator	22
Il·lustració 16: Keyboard	23
Il·lustració 17: Sequencer.....	23
Il·lustració 18: Transposer	24
Il·lustració 19: Pitch.....	24
Il·lustració 20: Signal Generator	24
Il·lustració 21: Delay.....	25
Il·lustració 22: VST.....	25
Il·lustració 23: Pan	25
Il·lustració 24: Envelope	26
Il·lustració 25: Filter	26
Il·lustració 26: Oscilloscope.....	27
Il·lustració 27: Spectroscope.....	27
Il·lustració 28: Routing	28
Il·lustració 29: Mixer	28
Il·lustració 30: Record.....	29
Il·lustració 31: Speakers.....	29
Il·lustració 32: Fragment del curs MOOC de Max	30
Il·lustració 33: Mòdul bàsic, amb els diferents blocs funcionals marcats.....	31

Il·lustració 34 Exemple de [caixa numérica] de nombres enters (esquerra) i decimals (dreta) ..	37
Il·lustració 35: Exemple de la caixa de missatge	37
Il·lustració 36: Exemple de l'objecte [bang].....	38
Il·lustració 37: Exemple de l'objecte [loadbang].....	38
Il·lustració 38: Exemple de l'objecte [toggle].....	39
Il·lustració 39: Exemple d'ús de l'objecte [+]	39
Il·lustració 40: Exemple de l'objecte [delay].....	40
Il·lustració 41: Exemple dels objectes [send] i [receive]	40
Il·lustració 42: Exemple de l'objecte [if]	41
Il·lustració 43: Exemple de l'objecte [select].....	41
Il·lustració 44: Exemple per a la seqüència d'execució de Max.....	42
Il·lustració 45: Quadre d'arrossegament inicial (esquerra) i final (dreta).....	42
Il·lustració 46: Secció de Direct Input modificada	43
Il·lustració 47: Secció de MLoad_Config modificada	44
Il·lustració 48: Secció de MVPlayer i MVDirectInput modificada	45
Il·lustració 49: Enviament d'àudio (esquerra) i vídeo (dreta).....	46
Il·lustració 50: Recepció d'àudio (esquerra) i vídeo (dreta)	46
Il·lustració 51: Mòdul de vídeo exclusiu per a vídeo.....	47
Il·lustració 52: Mòdul d'àudio exclusiu per a àudio	48
Il·lustració 53: Finestra de presentació de VScreen.....	49
Il·lustració 54: Secció del mòdul VScreen.....	50
Il·lustració 55: Finestra de presentació de VSwitcher.....	50
Il·lustració 56: Secció del mòdul VSwitcher	51
Il·lustració 57: Secció de la funció selectora del mòdul VSwitcher	52
Il·lustració 58 : VSampler en mode frame (esquerra) i mode loop (dreta)	53
Il·lustració 59: Secció de gravació de frames de VSampler	53
Il·lustració 60 : Secció de reproducció de frames de VSampler	54
Il·lustració 61: VSampler frame en mode rec (esquerra) i play (dreta)	55
Il·lustració 62: VSampler loop en mode rec (esquerra) i play (dreta).....	55
Il·lustració 63: Disseny de VSampler per a la funció visual	56
Il·lustració 64: Finestra de presentació del mòdul VColor	57
Il·lustració 65: Secció del mòdul VColor	57
Il·lustració 66: Correcció del so als vídeos.....	58
Il·lustració 67: Correcció del so a Direct Input.....	58
Il·lustració 68: Versió antiga del mòdul VSwitcher.....	59
Il·lustració 69: Versió antiga del mòdul VSampler.....	59
Il·lustració 70: Mostra dels mòduls al festival Kikk de Bèlgica	60

Il·lustració 71: Launcher de <i>Soundcool</i>	64
Il·lustració 72: Finestra de presentació del prototip del mòdul VBlender	64

AGRAÏMENTS

Començant per la base vull agrair als meus pares la seua implicació en aquest treball des de la rereguarda, des dels gestos més xicotets com el 'ja arpleguem nosaltres, tu ves i redacta' fins als gestos més grans que han fet i estan fent per mi. A la meua germana i al meu germà perquè sé que sempre vaig a tindre el seu recolzament i als meus nebots per ser la meua alegria.

Als meus companys de matins i rialles al laboratori, especialment a Jaime Serquera el qual m'ha guiat i m'ha ajudat en el meu aprenentatge tractant-me d'igual a igual des del principi d'aquest treball. A tot l'equip de Soundcool en general i a Jorge Sastre en particular que m'ha donat l'oportunitat de formar part d'aquest projecte.

Han sigut tantes i tantes tutories que Ximo Cerdà ja em reconeix per la meua forma de tocar a la porta del seu despatx. Vull agrair-li-ho tot des del primer correu: la seua implicació en el meu treball, el temps que ha tingut per a mi malgrat no tindre'n i els seus ànims i energies.

Estiguen lluny o prop, en qualsevol moment que ho necessite puc tindre un lloquet on estar a gust amb les meues amigues Agnès Andrea, Marta, Perla o Esther. Però no al mercat, elles em coneixen. A Irene i Paco per seguir compartint amb mi allò que fa que siguem com som: la música i el voluntariat.

A Claudio.

Capítol 1. Introducció

Aquest primer capítol servirà com a presentació del projecte *Soundcool*, sobre el qual està basat aquest treball de fi de grau. S'explicarà en què consisteix, les vessants on hi actua, els mòduls que el formen i els objectius del treball de fi de grau, a més d'explicar la motivació personal per haver elegit aquest projecte.

1.1 Objectius

Soundcool és un sistema d'educació i creació col·laborativa basat en la interacció entre mòbils, tauletes gràfiques i Kinect amb un ordinador central a temps real [1]. A la Il·lustració 1 es pot veure com els dispositius mòbils poden controlar aquest ordinador central. Consisteix a grans trets en un conjunt de mòduls interconnectables capaços de sintetitzar i transformar qualsevol tipus de so. *Soundcool* ha sigut adoptat en centres d'educació primària i secundària encara que degut a la seua versatilitat s'està introduint també en escoles de música, en conservatoris i fins i tot a nivell universitari.



Il·lustració 1: Funcionament de *Soundcool* a un ordinador i control amb mòbils

Soundcool és un entorn que es desenvolupa en tres plans principals: el pla tècnic, el creatiu i finalment el pla pedagògic [2]:

- El primer pla, i que li dona suport, és el plànol tècnic. És necessari tenir en compte l'estat actual dels smartphones i les tauletes gràfiques així com la seua constant i ràpida evolució.
- Un segon pla és el creatiu, on aquests elements nous del nostre temps poden dinamitzar tot el potencial de l'ésser humà, sent en el nostre àmbit el relacionat amb la música, el so i la imatge.
- Finalment, el tercer pla és el pedagògic molt estretament vinculat amb els dos anteriors. I és que resulta fonamental trobar metodologies d'aprenentatge que s'adeqüen als nous mitjans tecnològics facilitant la intencionalitat i l'expressió dels qui els utilitzen.

Avui dia l'educació musical més popular a les aules d'educació primària o secundària està basada en un aprenentatge passiu: llibres de text, llenguatge musical tradicional, instruments com ara la flauta o com a molt el xilòfon, etc. Açò passa mentre a la resta d'assignatures l'aprenentatge està començant a ser més dinàmic utilitzant les noves tecnologies: continguts en vídeo, exercicis i activitats multimèdia, etc.



Il·lustració 2: El sistema *Soundcool* a les aules

Així doncs la incorporació de les noves tecnologies en la creació, edició i producció del so per a aquests tipus de contingut educatiu fa que la manera d'ensenyar i crear música es renove i s'amplie en gran mesura [3]. Un exemple de noves tecnologies en les aules de musica el tenim a la Il·lustració 2. D'aquesta manera es pot millorar el procés d'aprenentatge i d'ensenyament, fent que els alumnes es senten més motivats i més familiaritzats amb aquests nous sistemes.

Cal remarcar que el fet d'incorporar les noves tecnologies no implica deixar a banda la llibreta de pentagrama ni l'ensenyament amb els instruments de sempre, sinó fusionar-ho amb les noves possibilitats que ens ofereixen per tal de millorar l'experiència als centres tant per part de l'alumnat com del professorat.

A part de la vessant a l'ensenyament, *Soundcool* també participa en projectes purament artístics com ara concerts des del seu primer *Off_hercios*, un circuit itinerant d'Art Sonor [4] [5] a "Humans vs Insects" o l'òpera de "La Mare dels Peixos" (Il·lustració 3) basada en una rondalla d'Enric Valor representada a València al complet i a Mèxic i Romania el primer acte.

El sistema *Soundcool* consisteix a grans trets en un conjunt de mòduls interconnectables capaços de sintetitzar i transformar qualsevol tipus de so. El funcionament d'aquests és simple: Cadascun dels mòduls es pot controlar mitjançant un dispositiu Smartphone o tauleta gràfica bé siga Android o IOS connectat a un ordinador pare via wifi. A més es pot utilitzar la interfície de videojocs Kinect de Xbox que captura el moviment de l'usuari per a controlar també els mòduls [6]. El so pot provenir per exemple de llibreries de sons virtuals, arxius MIDI (*Musical Instrument Digital Interface*), àudios propis o so directe des d'un dispositiu de gravació. Tots aquests mòduls funcionen en un ordinador i poden ser controlats pels dispositius mòbils via wifi.



Il·lustració 3: Representació de La Mare dels Peixos a València al Palau de les Arts Reina Sofia de València

Més avant s'explicarà el funcionament complet de tots els mòduls que el componen. Ara en la introducció simplement es classificaran els diferents tipus de mòduls per a tindre una idea general. La funció dels mòduls es pot dividir en quatre grups:

- Els mòduls que introdueixen el so:
 - Player
 - Sampler Player
 - Direct Input
 - Keyboard
 - Signal Generator

- Els mòduls que modifiquen el so:
 - Transposer
 - Pitch
 - Signal Generator
 - Delay
 - VST
 - Pan
 - Envelope
 - Filter

- Els mòduls que analitzen el so:
 - Oscilloscope
 - Spectroscope

- Els mòduls finals, que o bé mesclen el so, el graven, o el reproduïxen:
 - Routing
 - Mixer
 - Record
 - Speakers

L'objectiu d'aquest treball és ampliar el sistema *Soundcool* creant una nova rama d'imatge amb funcionalitats similars als mòduls de so, és a dir, crear almenys un mòdul d'imatge que introduisca la imatge, un que les mescle i un que reproduisca les imatges.



II·lustració 4: Utilització de *Soundcool* a mòbils i tauletes gràfiques

La última versió de *Soundcool*, a part dels mòduls de so com els de la II·lustració 4, ja compta amb dos mòduls d'imatge: amb *VideoDirectInput* i amb *VideoPlayer*. El primer permet mostrar imatges directament gravades des de la webcam i el segon permet reproduir qualsevol vídeo guardat a l'ordinador.

Cal remarcar que per a aconseguir crear aquests nous mòduls abans s'haurà de fer possible les connexions entre vídeos ja que el sistema està optimitzat per a so. També haurem d'aconseguir que els mòduls d'àudio no es puguin connectar als de vídeo i que els de vídeo no es puguin connectar als de àudio.

- El primer objectiu que es marcarà serà crear un *VideoScreen* que permet visualitzar el contingut en una altra pantalla per tal de poder projectar el contingut. Es provarà el correcte funcionament d'aquest mòdul amb els mòduls ja creats com *VideoDirectInput* o *VideoPlayer*.
- Després es crearà un *VideoSwitcher* on podrem elegir quin vídeo passar al *VideoScreen*. Una vegada fet, comprovem el funcionament del mòdul amb els mòduls de visualització ja creats.
- L'últim mòdul que es crearà serà un *VideoSampler* on es podrà gravar i elegir a partir de quina frame es reproduirà el vídeo. També es podrà elegir un fragment del vídeo per a poder reproduir-lo en mode *Loop* tant en sentit normal com al revés o en mode palíndrom.

1.2 Motivació

Soundcool és un sistema d'educació musical i creació col·laborativa mitjançant mòbils, tauletes gràfiques, Kinect, OSC (*Open Sound Control*) i Max/MSP/Jitter creat en la Universitat Politècnica de València (UPV) i un equip d'investigadors que combina perfils artístics i científics agrupant a músics, enginyers, pedagogs i experts en comunicació audiovisual i investigació d'Arts Performatives i Tecnologia (*PerformingARTech*), grup interdisciplinari UPV [1].

Aquest projecte d'investigació, dirigit per Jorge Sastre, va començar en 2013 amb financiació de la Universitat Politècnica de València (UPV) i un equip d'investigadors que combina perfils artístics i científics agrupant a músics, enginyers, pedagogs i experts en comunicació audiovisual i investigació. A més recentment aquest projecte ha guanyat el NEM Art competition del NEM Summit 2017 a Madrid. Els components que formen aquest projecte són els següents:

Direcció

- Dr. Jorge Sastre, compositor, enginyer i matemàtic, director Grupo PerformingARTEch, dels cursos de Música Electrónica y Vídeo Creació (MEVIC) i professor del Màster Universitari en Música UPV (MUM).

Vessant tècnica y artística

- Dra. Nuria Lloret, directora del MUM, responsable de comunicació, webs i xarxes socials del projecte i professora del Dpto. de Comunicació Audiovisual Documentació i Història del Art UPV
- Dr. Adolfo Muñoz, responsable gràfic de *Soundcool*, professor del Dpto. de Comunicació Audiovisual Documentació i Història de l'Art UPV
- Dr. Carlos Hernández, director del àrea de *Soundcool* per a estudiants de diversitat funcional
- Dr. Luisa Tolosa, àrea de *Soundcool* per a estudiants de diversitat funcional
- Dr. José Ramón Díaz, àrea de *Soundcool* per a estudiants de diversitat funcional
- Montse Briceño, àrea de *Soundcool* per a estudiants de diversitat funcional, grup EMOSONS
- Dr. Stefano Scarani, videoartista, compositor, especialista en sistemes interactius i professor MEVIC.
- Dr. Ximo Cerdà, coordinador MEVIC.

Vessant pedagògica

- Dr. Adolf Murillo (Responsable pedagògic), assessor del Centro de Formació Innovació i Recursos Educatius (CEFIRE) i professor de la UPV.
- Dra. Elizabeth Carrascosa, professora de la Universitat de València i del CEIP 9 d'Octubre (Alcàsser, València), i directora del projecte europeu Erasmus+ KA2 para la introducció de *Soundcool* en el sistema educatiu europeu.
- Dr. Remigi Morant, director del Dpto. De Didàctica de la Expressió Musical, Plàstica i Corporal de la Universitat de València, vicepresident de la Federació de Societats Musicals de la Comunitat Valenciana (FSMCV).
- Dra. María Elena Riaño Galán, professora de didàctica de la Expressió Musical del Dpto. de Educació en la facultat de Educació de la Universitat de Cantàbria.
- Dra. Noemy Berbel Gómez. Vicedecana de la Facultat d'Educació i coordinadora de mobilitat. Departament de Pedagogia i Didàctiques específiques. Àrea de didàctica de l'expressió Musical. Universitat de les Illes Balears.

Desenvolupament

- Dr. Jaime Serquera (12/2015-actualitat)
- Elena Pelejero (12/2016- actualitat)
- Xiaoqian Huang (1/2017-5/2017)

- Rafael García (9/2014-7/2015)
- José Enrique Serrano Comes (1/2013-3/2014).
- Deng Han (1/2014-4/2014).

Col·laboren

- Elena Robles Mateo, especialització en cultura audiovisual y estètica. Comunicació i disseminació internacional i disseny (11/2015-actualitat)
- Narce Dalia Ruiz Guzmán, directora de programació del FIC Monterrey. Community Manager (2/2017- 6/2017)
- Prof. Roger Dannenberg, director del grup de Computer Music, Carnegie Mellon University (Pittsburgh, EEUU)
- Sara Masters, grup de Computer Music, Carnegie Mellon University (Pittsburgh, EEUU) (2013-actualitat)
- Giannino Clemente titulat MEVIC i estudiant de Berklee (2013-actualitat).
- Angélica Rodríguez, doctoranda del MUM (2017-actualitat)
- Álvaro Forner Durá (Desenvolupament de Tutorials *Soundcool*-Audacity, Treball de Fi de Grau Eng. Telecomunicació, 2017)
- Dr. David Picó, professor del MEVIC UPV.
- Gregorio Jiménez, professor del MEVIC UPV.
- Lucía García Talavera (Tutorials *Soundcool*, Treball de Fi de Grau Eng. Telecomunicació, 2016)
- Marcelo Estornell Boscà. Professor de Música (ESO) i Professor de Trompeta (Institut de Música d'Andorra la Vella)

Com es pot veure, el perfil de tots els components és o bé tecnològic, artístic o pedagògic. Aquest projecte ens ha atret des del primer moment justament per aquesta raó, perquè estem interessats en les tres principals vessants que fan possible *Soundcool*:

- En primer lloc la vessant tecnològica òbviament perquè hem cursat el estudis del Grau en Enginyeria de Tecnologies i Serveis de Telecomunicació, la especialitat de so i imatge.
- Musicalment estem interessats perquè hem obtingut el Grau Mitjà de Música en especialitat de Violoncel i seguim rebent classes d'instrument i formant part d'orquestres. A part també estem interessats en la mescla de música i tecnologia fent cursos de Producció Musical amb ProTools o tallers com d'Art Sonor i Producció de Música Electrònica.
- Finalment està la part pedagògica, en la qual hem participat mesclant pedagogia i tecnologia al Centre Ocupacional La Torre donant classes d'informàtica durant un any. A part també formem part de l'equip de monitors d'activitats de temps lliure des dels 18 anys a Atalaia, una agrupació cultural, recreativa i excursionista que presta la seua oferta d'activitats fonamentada a partir d'uns objectius específics a xiquets d'entre 8 i 17 anys.

Formar part de l'equip de *Soundcool* creguem que és una experiència enriquidora en molts sentits per a nosaltres tant tecnològicament com musical i pedagògica, al mateix temps que podem aportar al projecte visions molt diferents en els tres àmbits.

Capítol 2. Context

Després d'explicar el projecte *Soundcool* a grans trets es farà un breu resum de l'estat actual de la creació col·laborativa en l'àmbit multimèdia i del *software* lliure relacionat. S'aprofundirà en el sistema *Soundcool* i en el funcionament del programari que el forma i després d'això s'explicarà cadascun dels mòduls que existeixen actualment.

2.1 Estat de l'art de la creació col·laborativa

Un ampli ventall de les produccions audiovisuals digitals contemporànies ja no es construeixen de forma individualista, sinó de forma col·lectiva i plural [7]. Sorgeixen de plantejaments creatius basats en la participació de la producció de l'obra, i de que els mitjans digitals possibiliten com mai abans. Nombrosos grups i col·lectius implementen diàriament una autoria compartida, fruit de la interrelació de sistemes i continguts de procedència i naturalesa divers. Açò dona pas a una nova forma d'entendre l'acció comunicativa en la qual el creador deixa de ser únic i solitari, per passar a convertir-se en un autor plural, treballant germanat con una col·lectivitat de co-participants de l'acte creatiu.

“La creació col·lectiva és el contingut més revolucionari de la cultura digital, allò que millor facilita la seva distinció de la cultura tradicional és la possibilitat de construir una cultura realment col·lectiva” [8]. L'expressió *Networking* (fer xarxa) és la seua síntesi conceptual. Consisteix en el desenvolupament de pràctiques creatives participatives i col·laboratives enteses alhora com a activitat social i cultural, i com a forma de gestió i producció descentralitzada, trencant així amb les jerarquies i els models dominants en els sistemes culturals tradicionals i de poder.

Les estratègies creatives apropiacionistes resulten pioneres i emblemàtiques d'aquesta col·lectivització creativa pròpia d'internet. Ja des de mitjan anys 90', són exemples habituals de *networking* la creació de fòrums i BBS (*bulletin board system*, 'sistema de tauler d'anuncis'), amb xarxes d'afinitat vinculades al seu torn a plataformes electròniques més àmplies: plataformes web d'organitzacions socials, polítiques i d'informació independent [9]. També n'és un exemple la distribució d'informació sobre determinats fets entre persones que no tenen cap accés a aquesta informació a través d'altres mitjans, com per exemple notícies oblidades pels mitjans de comunicació corporatius o que han estat ofertes tan sols de forma tendenciosa. Per últim també ho és la reapropiació del contingut del determinades pàgines web a fi i efecte de subvertir la seva funció comercial, qüestionant així la política de drets d'autor i altres restriccions habituals per a l'expressió cultural al món comercial-privatiu.

Esprémer totes les manifestacions possibles de la comunicació audiovisual digital té en el *software* lliure a un de de els seus majors aliats, així com el seu ús i la vitalitat de les comunitats virtuals existents afins resulten avui un clar exponent dels processos de redefinició de rols en el camp audiovisual als quals estem assistint. Diàriament nous projectes participatius s'afegeixen, mentre uns altres s'adapten o canvien en funció de nous interessos, fruit de la condició radicalment canviant del present de la comunicació audiovisual.

L'evolució tecnològica i els processos de creació i producció de projectes 'estil Basar' [10] proporcionen noves oportunitats per a la creació i la difusió audiovisual amb un elevat grau d'autonomia i d'independència. En l'intensiu procés de canvi i substitució de models i de conceptes clau al qual assistim, la producció d'aplicacions audiovisuals de naturalesa 'lliure' viu actualment una fase de recerca de les seves possibilitats. L'actiu món del *software* lliure fomenta una manera diferent de veure, percebre, viure i valorar les formes audiovisuals, d'assenyalar les problemàtiques i també les oportunitats de les incerteses, una manera que ja s'assumeix com a pròpia per part de les joves generacions que s'identifiquen com a usuaris dels mitjans de comunicació, i ja no com a mers receptors d'aquests.

En el camp concret del *software* audiovisual resulta difícil i car adaptar-se en nombroses ocasions a les principals solucions propietàries estàndard. Les empreses de *software* solen treballar, fins i tot en les seves solucions per a entorns docents, amb llicències d'ús individual i limitat, que suposen un enorme sobrecoast per als estudiants i usuaris emprenedors. Junt a aquestes solucions, l'opció per l'ús de *software*

lliure per a la creació i la producció audiovisual permet als usuaris un major grau d'adaptació, de creació col·lectiva, d'iniciativa en la configuració i en la definició de la pròpia eina.

La preferència per l'expressió '*software* lliure' en detriment d'altres properes com a '*software* de codi obert' o '*software* de domini públic' ha d'assumir implicacions socials i polítiques perquè el seu ús no tinga poca importància. D'acord amb la caracterització establerta per la mateixa *Free Software Foundation* (l'organització internacional creada en 1985 dedicada al desenvolupament i la promoció del *software* lliure), un programa és *software* lliure sempre que els usuaris d'aquest tinguen garantides quatre llibertats bàsiques: la llibertat d'executar el programa siga quin siga el seu propòsit, la llibertat d'estudiar i modificar el programa per ajustar-ho a les seues necessitats, la llibertat de redistribuir còpies, siga de forma gratuïta o a canvi del pagament d'un preu, i la llibertat de millorar el programa i després fer-ho públic pel bé de tota la comunitat.

En contra del que resulta una accepció errònia habitual, l'ús de *software* lliure no implica forçosament la seva gratuïtat. Qualsevol programa lliure està disponible per al seu posterior ús, desenvolupament o distribució comercial -sempre que respecti les quatre llibertats bàsiques a les quals hem fet referència-. *Software* lliure no vol dir software gratis, *software* de preu zero. Hem d'evitar interpretar el seu ús i promoció com una mera qüestió de preu. És i suposa molt més que això. El *software* lliure implica, abans de res, fomentar i garantir que tots els usuaris d'un programa tinguen la llibertat de cooperar amb uns altres en utilitzar-ho, i poder així compartir i fer extensibles les seues millores a la resta de la comunitat, "(el seu ús) tindria un avantatge social, en permetre als usuaris cooperar, i una ètica, en respectar la seua llibertat" [11].

Implica assumir de forma plena i coherent una perspectiva comunitària, una ideologia cooperativa i responsable pel bé comú, així com una ètica hacker del treball [12]. L'aposta pel *software* lliure comporta doncs treballar en honor d'una societat on la informació estiga veritablement a la disposició dels seus ciutadans, resultant per a aquests usable i accessible, facilitant i promocionant el desenvolupament de programes que la gent pugua adaptar, millorar i compartir, i ja no solament executar.

En afinitat amb la perspectiva ètica comunitària que domina el desenvolupament del conjunt del *software* lliure, el desenvolupament de projectes audiovisuals lliures i/o de codi obert, suposa i persegueix fomentar i garantir la possibilitat de participació col·lectiva en projectes de creació audiovisual, així com la possibilitat de compartir i fer ús dels continguts audiovisuals resultants per part del conjunt de la societat, des de la creença ètica en comú i compartida de que els continguts audiovisuals han de ser en la nostra societat una xarxa dels usuaris i per als usuaris.

La creació i producció de projectes audiovisuals col·laboratius implica així afavorir l'enfortiment d'una cultura lliure, en la qual s'assumisca de forma plena i coherent una perspectiva cooperativa i responsable pel bé cultural comú, d'ús, accés i domini públic, en afinitat amb la inspiració de les pròpies eines lliures que han permès, afavorit o inspirat la seva creació. És a dir, treballar per tal d'aconseguir una societat on la informació i els continguts de naturalesa audiovisual estiguen verdaderament a la disposició dels seus ciutadans, resultant per a aquests plenament usables i accessibles, sense les tan habituals limitacions i restriccions d'ús a les quals la cultura audiovisual comercial ens té avui tan habituats [13].

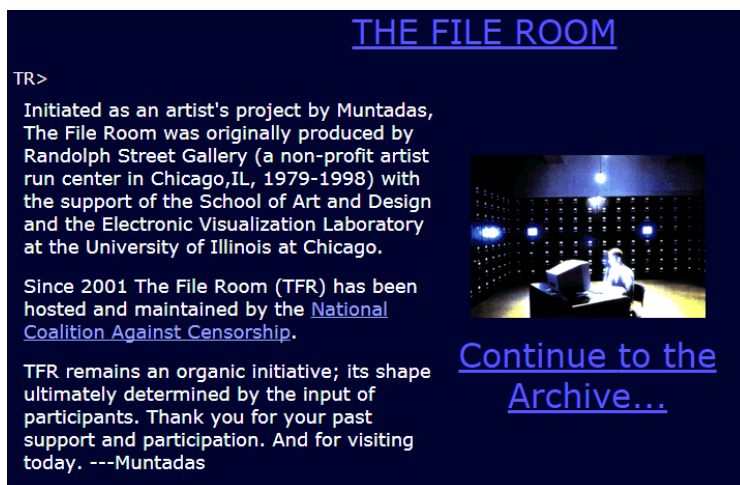
2.2 Exemples previs en l'àmbit de la creació col·laborativa

En aquest apartat explicarem breument dos projectes de creació col·laborativa. El primer és un projecte fotogràfic creat en 1994 i que segueix a l'actualitat, i el segon és un conegut llargmetratge estrenat en 2011.

2.2.1 The File Room

A l'any 1994 Antoni Muntadas [14] inicia el projecte *The File Room*, una obra audiovisual col·laborativa. Segueix activa actualment i és resultat del compromís social i de l'ètica col·laborativa assenyalada en l'acció comunicativa a internet.

The File Room [15] consisteix en un arxiu audiovisual accessible i col·laboratiu on-line que creix dia a dia amb noves dades sobre la censura artística i cultural que es practica al món. Podem vore la seua pàgina d'inici a la Il·lustració 5. Qualsevol persona connectada a la red té la possibilitat d'accedir a les imatges prohibides, a la informació recopilada sobre els autors i les seues circumstàncies, així com de proposar i incorporar participativament noves obres i casos.

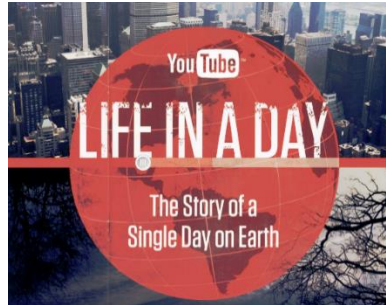


Il·lustració 5: Pàgina d'inici de The File Room

El propi autor d'aquesta obra audiovisual col·laborativa explica que "Des del seu inici, aquest projecte ha tingut en compte les qüestions suscidades per la interactivitat, l'audiència i el rol social de l'art. Mentre s'estava elaborant *The File Room*, els problemes plantejats per les noves tecnologies respecte a la censura, les fonts de l'obra, la seua accessibilitat, el seu llenguatge, la seua traducció, el seu control i les seues motivacions van ser cercades: tots aquests problemes són al seu torn elements constitutius i conseqüències de l'obra i de la seua temàtica."

2.2.2 Life in a Day

Un projecte més recent i conegut pot ser la pel·lícula *Life in a Day* [16], mostrat a la Il·lustració 6. Aquesta és una pel·lícula documental projecte de YouTube, anunciat el 6 de juliol de 2010. Els usuaris de YouTube havien d'enviar un vídeo amb la condició d'estar gravat únicament el 24 de juliol. Posteriorment el director Kevin Macdonald va editar els vídeos dels contribuents per a fer la pel·lícula.



Il·lustració 6: *Life in a Day*, exemple de pel·lícula col·laborativa

El llargmetratge complet va ser estrenat en el Festival de Cinema de Sundance en 2011 i mostrat en directe per YouTube. Tots els autors dels vídeos triats han sigut acreditats com co-directors al ser una pel·lícula creada col·laborativament.

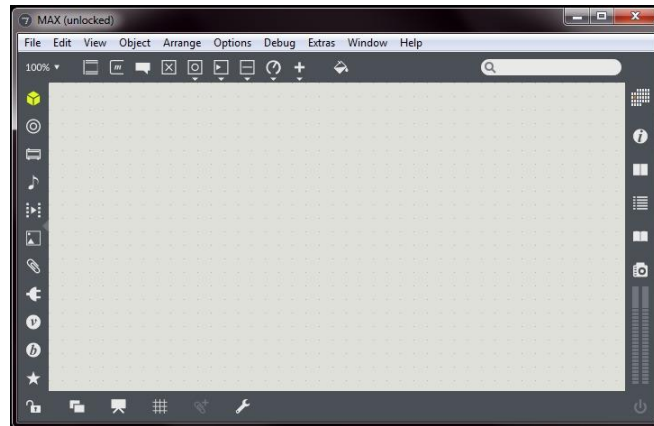
2.3 Programari emprat per a desenvolupar *Soundcool*

En aquest apartat s'explicarà el programari bàsic que fa possible *Soundcool*: Max/Msp/Jitter, *Open Sound Control* (OSC) i Unity. Max serà el programa que es descriu amb més profunditat en aquest capítol per ser el més utilitzat de tots els que es van a explicar.

2.3.1 Max/Msp/Jitter

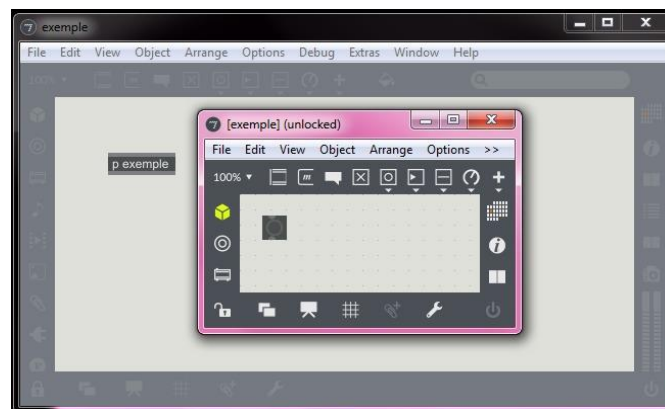
Max és un entorn de desenvolupament gràfic utilitzat en l'àmbit musical i multimèdia desenvolupat per Cycling '74 [17]. El programa ha sigut utilitzat durant més de quinze anys per compositors, artistes i dissenyadors de programes interessats en la creació de programes interactius.

Max és un entorn modular, i la majoria de les rutines formen part d'una biblioteca compartida. La API (Interfície de Programació d'Aplicacions) permet el desenvolupament de noves rutines per a terceres persones. Podem veure un exemple de la pantalla principal de Max en la Il·lustració 7. Max té una base modular de codi amb externs o objectes que són utilitzats com a blocs de construcció per a programes escrits en el *software*. Açò fa el programa arbitràriament extensible a través d'una API pública, i ajuda als desenvolupadors a afegir les seues pròpies rutines d'àudio i control, ja siga en el llenguatge de programació C o, amb l'ajuda d'altres externs, en Python, Javascript, Ruby, i potencialment altres llenguatges també.



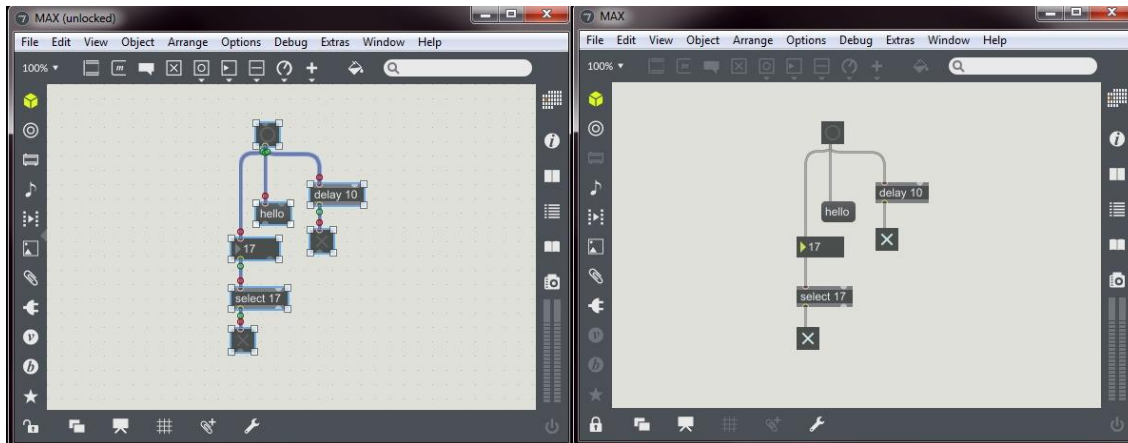
II·lustració 7: Entorn de Max

En realitat Max és un llenguatge de programació en si mateix [18]. Unitats de codi modulars són usades com a programes independents i compartits lliurement entre la comunitat d'usuaris de Max. Els seus *patches* consten de diferents objectes interconnectats entre ells. En la seua part superior trobarem les entrades, on se'ls enviaren valors numèrics o altres tipus de dades, i en la inferior l'eixida d'aquests. També existeix la possibilitat de crear *patches* secundaris coneguts com *subpatches*, que estan dins del *patch* principal, mostrats a la II·lustració 8. Clicant sobre ells se'ns obri la finestra on trobem el codi del nostre *subpatcher*.



II·lustració 8: Patcher i subpatcher

El programa té dos estats en els quals es pot trobar l'usuari. En manera d'edició o en manera d'execució, com es pot veure a la II·lustració 9. Per a canviar d'un estat a un altre teclejem Ctrl+E. Quan estem en el mode edició, podem modificar el contingut de les caixes, o la connexió entre elles. En el mode d'execució tenim la possibilitat d'activar tot el *patcher*, i anar modificant valors durant la seua reproducció o quan aquest estiga parat.



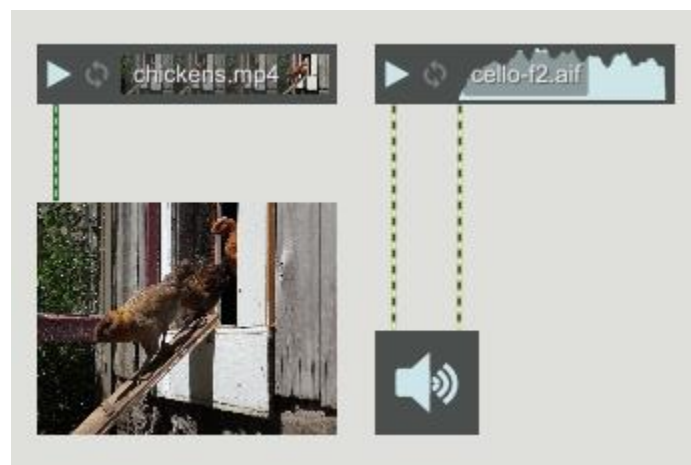
II·lustració 9: Max en mode edició (esquerra) i en mode execució (dreta)

A causa del seu disseny extensible i interfície gràfica, Max és considerat per molts com la llengua franca per al desenvolupament de programes de música interactiva. El programa compta amb la versió completa amb llicència i amb una versió d'execució anomenada *Max Runtime*. Els nostres mòduls han sigut pensats per a ser executats en aquesta última versió, ja que s'adapta perfectament a les necessitats del projecte. D'una banda, a diferència de la versió completa, la versió *Runtime* és gratuïta. A més, permet únicament l'execució de *patchers* i no la seua modificació, la qual cosa és ideal tenint en compte que les persones que van a utilitzar els mòduls no són usuaris ni molt menys experts de Max. No obstant açò, com és lògic, el desenvolupament dels mòduls s'ha realitzat amb la versió completa.

Els objectes que acabem de descriure són per al que es coneix com a dades tipus Max. En contraposició a les dades tipus Max, tenim els senyals MSP [17], com per exemple l'àudio. Aquestes senyals les podem detectar perquè viatgen a través de cables groc-verdosos amb ratlles negres en lloc del típic cable gris de les dades tipus Max. Sobre els objectes per a senyals MSP, aquests porten sempre el símbol ~, en al·lusió a una ona sinusoidal. En el cas dels operadors matemàtics, les versions MSP dels quals acabem de comentar serien +~, -~, ~ i /~. Encara que no tots els objectes disposen de versions per a dades Max i senyals MSP, molts d'ells sí les tenen i es diferencien per portar les de MSP el símbol ~ en el seu nom.

En el cas de Jitter [19] els cables que uneixen els seus elements són verds amb ratlles negres. Jitter serveix per a treballar en vídeos en temps real, en vectors i en gràfiques 2D i 3D, i efectes que es poden aplicar a aquests elements. Els objectes que utilitzen Jitter normalment comencen amb un jit. seguit de la funció de l'objecte, com per exemple jit.+ jit.- jit.convolve, i molts més.

Vegem un exemple tant de MSP com de Jitter a la II·lustració 10.



II·lustració 10: Exemple de ús de MSP (esquerra) i de Jitter (dreta)

2.3.2 Open Sound Control

Open Sound Control és un protocol de comunicacions que permet comunicar instruments de música, ordinadors i altres dispositius multimèdia com per exemple mòbils o PDA's [20]. És fonamental per al projecte *Soundcool* ja que està pensat per a compartir informació musical en temps real sobre una xarxa. Apareix com a reemplaçament del MIDI, perquè és molt superior en característiques i capacitats. Les diferències principals entre ambdós són:

- MIDI defineix una interfície de hardware (el cable de MIDI), però OSC es transmet normalment sobre ethernet.
- MIDI utilitza un format de missatge binari compacte amb un número limitat de camps. OSC té una secció de dada binària, una adreça d'estil URL
- OSC permet transmetre molts tipus d'informació múltiple com per exemple *integers*, *floats* o *strings*. MIDI només transmet *integers*.
- OSC inclou un *timestamp* d'alta precisió amb una resolució de picosegons que permet que els missatges d'OSC siguin transmesos amb un jitter mínim. En MIDI la resolució és més baixa: té una precisió de l'ordre de pocs mil·lisegons en el millor dels casos.

OSC va ser desenvolupat en un principi a la *UC Berkeley Center for New Music and Audio Technology* (CNMAT) i hi continua ser un tema de recerca actual. El CNMAT és un centre d'investigació interdisciplinari que està dins del Departament de Berkeley de l'UC de Música. CNMAT és conegut pel seu programa dinàmic i educatiu, l'actuació i la recerca programa enfocar en la interacció creativa entre música i tecnologia. Les característiques principals del protocol OSC són:

- És ampliable i dinàmic. Compta amb un esquema de noms simbòlics tipus URL
- Les seues dades numèriques són simbòliques i d'alta resolució
- Té un llenguatge de coincidència de patrons (*pattern matching*) per a especificar múltiples receptors d'un únic missatge
- Té marques de temps (*estafe tags*) d'alta resolució.
- Els seus missatges són empaquetats per als esdeveniments que han d'ocórrer simultàniament
- Té un sistema d'interrogació per trobar les capacitats d'un OSC i obtenir documentació.
- Pot ser transportat per diversos protocols, però comunament s'usa UDP.

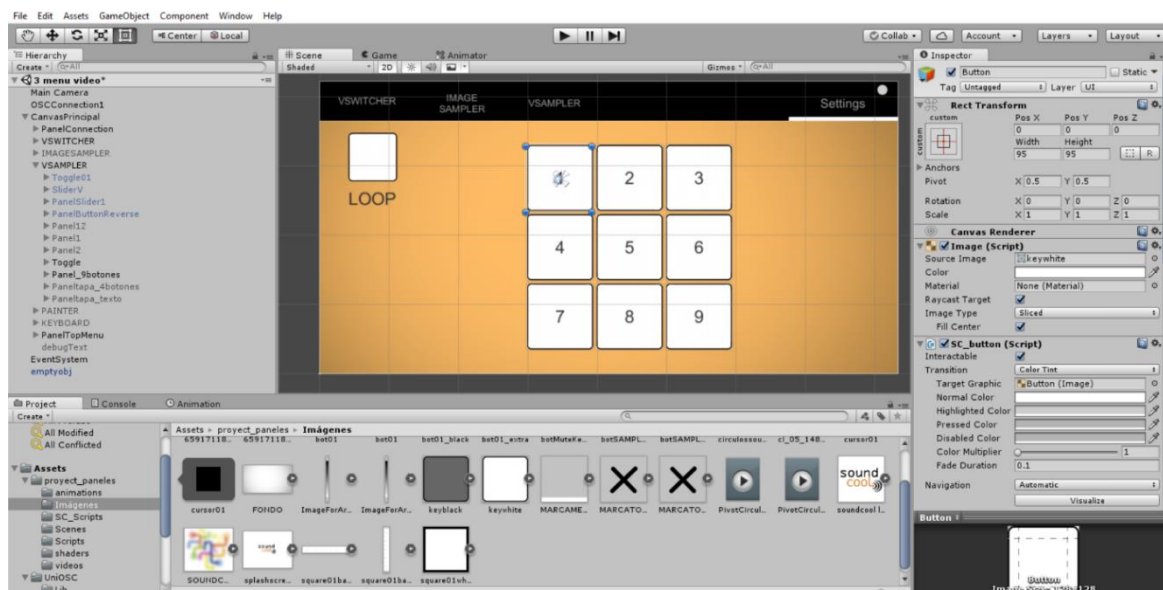
Alguns projectes coneguts que tenen la seua implementació són per exemple REAPER, ChuckK, CSound, Max/MSP, Pure Data, Reaktor, SuperCollider, Squeak, VVVV.

2.3.3 Unity

Unity és un motor de videojoc multiplataforma creat per Unity Technologies que està disponible com a plataforma de desenvolupament per a Microsoft Windows, OS X, i Linux [21]. La plataforma de desenvolupament té suport de compilació amb diferents tipus de plataformes. Hem utilitzat Unity en *Soundcool* per a la creació de l'aplicació per a telèfons mòbils i tauletes gràfiques com es pot veure a la

Il·lustració 11

El motor gràfic utilitza *OpenGL* (en Windows, Mac i Linux), *Direct3D* (solament en Windows), *OpenGL ES* (en Android i iOS), i interfícies propietàries (Wii). Amb açò nosaltres hem fet l'aplicació compatible tant per a Android com per a iOS.



II-lustració 11: Creació de l'app per a Android i iOS en Unity

S'usa el llenguatge ShaderLab per a la creació de *shaders*. Es poden escriure en tres formes diferents: com Surface *shaders*, com Vertex and Fragment *shaders*, o com *shaders* de funció fixa. Un *shader* pot incloure múltiples variants i una especificació declarativa de reserva, la qual cosa permet a Unity detectar la millor variant per a la targeta de vídeo actual i si no són compatibles, recórrer a un *shader* alternatiu que pot sacrificar característiques per a una major compatibilitat.

2.4 El sistema *Soundcool*

Soundcool és un sistema d'educació musical i creació col·laborativa mitjançant mòbils, tauletes gràfiques, Kinect, OSC (*Open Sound Control*) i Max/MSP/Jitter creat en la Universitat Politècnica de València (UPV) i un equip d'investigadors que combina perfils artístics i científics agrupant a músics, enginyers, pedagogs i experts en comunicació audiovisual i investigació d'Arts Performatives i Tecnologia (PerformingARTech), grup interdisciplinari UPV [1]. *Soundcool* ha sigut adaptat en centres d'educació primària i secundària encara que degut a la seua versatilitat s'està introduint també en escoles de música, conservatoris i universitats. Els projectes clau relacionats amb *Soundcool* són:

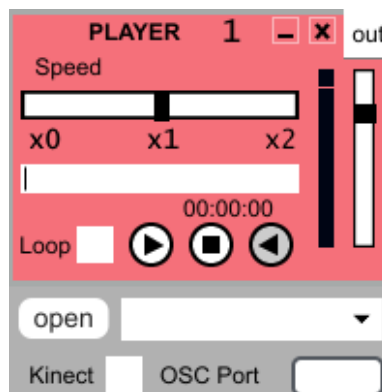
- Fundació Daniel y Nina Carasso. Projecte 16-AC-2016 (9/2016-8/2019): Desenvolupament de sistemes de Videoart 2016-2019. Centrat en el desenvolupament de sistemes interactius de vídeo art, noves apps OSC, nous desenvolupaments per a estudiants amb necessitats especials, i increment de la difusió nacional i internacional.
- Comissió Europea. Projecte Erasmus + 2015-1-ES01-KA201-016139 (9/2015-7/2017): "Tecnología al servicio del aprendizaje y la creatividad: tejiendo redes europeas a través de la creación musical colaborativa", 2015-2017. Centrat en la difusió educativa a Europa.
- Generalitat Valenciana. Projecte AICO/2105/120 (1/2015-12/2016): "*Soundcool*: Nuevas Tecnologías para la Educación Musical y la Creación Sonora" (2015-2016). Dedicat al desenvolupament de la part d'àudio del *software* de *Soundcool* i a l'inici de les tasques centrades en estudiants amb necessitats especials.
- UPV. Projecte PAID-05-12-SP20120470 (2013): "*Soundcool*: Nuevas Tecnologías e Interfaces para la Educación Musical y la Creación Sonora" (2013).

La funció dels mòduls de *Soundcool* es pot dividir en quatre grups: Els mòduls que introdueixen el so, els mòduls que el modifiquen, els mòduls que l'analitzen i els mòduls que el mesclen. El funcionament de connexió entre mòduls és simple i intuïtiu. Aquests tenen a la seua part superior una entrada representada amb el quadrat *in* i una eixida representada com a *out*. La manera de connectar-los entre sí serà fer clic primerament en el botó *out* del mòdul que es vulga connectar i després fer clic en el botó *in* del mòdul al que es vulga connectar. Els mòduls que introdueixen el so només tindran la opció de *out*, els d'efectes i els mescladors tindran les dos opcions, i els que analitzen, els mescladors, reproductor i gravador només tindran la opció de *in*.

Ara explicarem la funció de cada mòdul per a que s'entenga millor el funcionament de *Soundcool*. Cada explicació va referenciada a la il·lustració que te al davall.

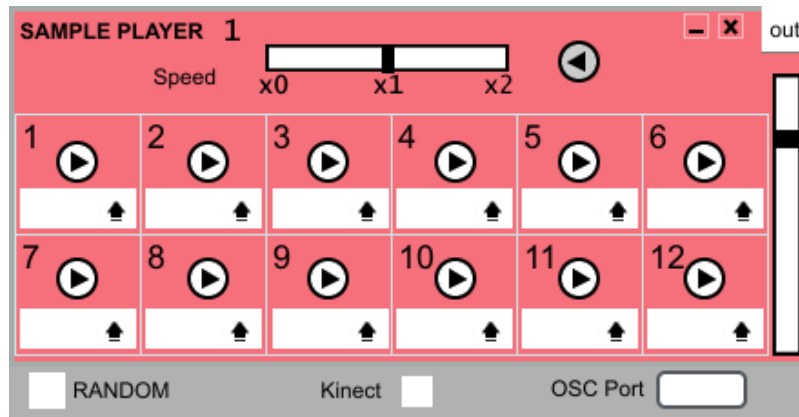
2.4.1 Moduls que introdueixen el so

- **Player:** Aquest mòdul permet reproduir arxius d'àudio. Mitjançant el botó *open* es carrega l'arxiu d'àudio. Es pot reproduir en bucle activant la casella *Loop*, es pot reproduir o pausar en el botó de *Play*, parar amb el *Stop*, i reproduir-lo a l'inrevés amb el tercer botó del mòdul. Es pot controlar la velocitat a la barra de *Speed*, i el volum a la barra vertical de la dreta.



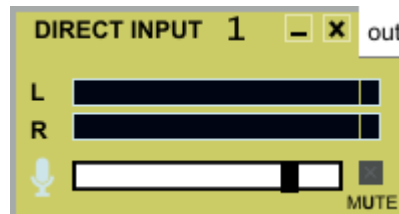
Il·lustració 12: Player

- **SamplePlayer:** A l'igual que el Player, tenim els controls de velocitat, de reproducció/pausa, de volum i de reproducció inversa. Aquest mòdul està pensat per a reproduir de forma ràpida els diferents sons que es vulguen carregar en les 12 posicions disponibles. També es poden reproduir aleatòriament fent clic en la casella *Random*



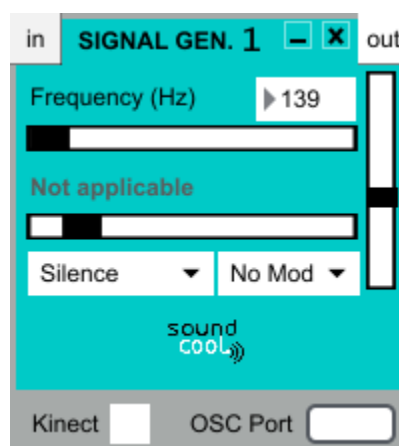
Il·lustració 13: SamplePlayer

- DirectInput: Aquest mòdul te a la seua eixida la senyal captada directament d'un dispositiu d'entrada de l'ordinador com el micròfon. Te un *slider* per a controlar el volum de la gravació i un *toggle* per a mutejar la entrada



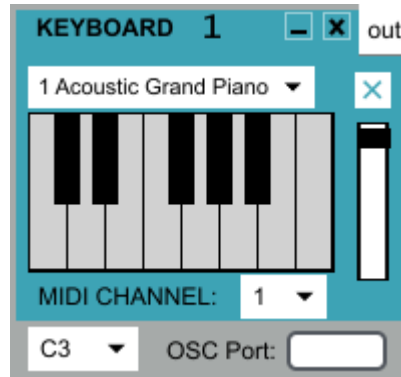
Il·lustració 14: Direct Input

- Signalgenerator: Aquest mòdul permet generar senyals de so. La freqüència es pot ajustar en el *slider Frequency* o introduint-la en la casella numèrica. Podem elegir senyals sinusoidals, triangulars, quadrades, de dent de serra o soroll blanc o rosa. A part de generar també pot modificar un so d'entrada, ho explicarem en l'apartat corresponent de mòduls d'efectes.



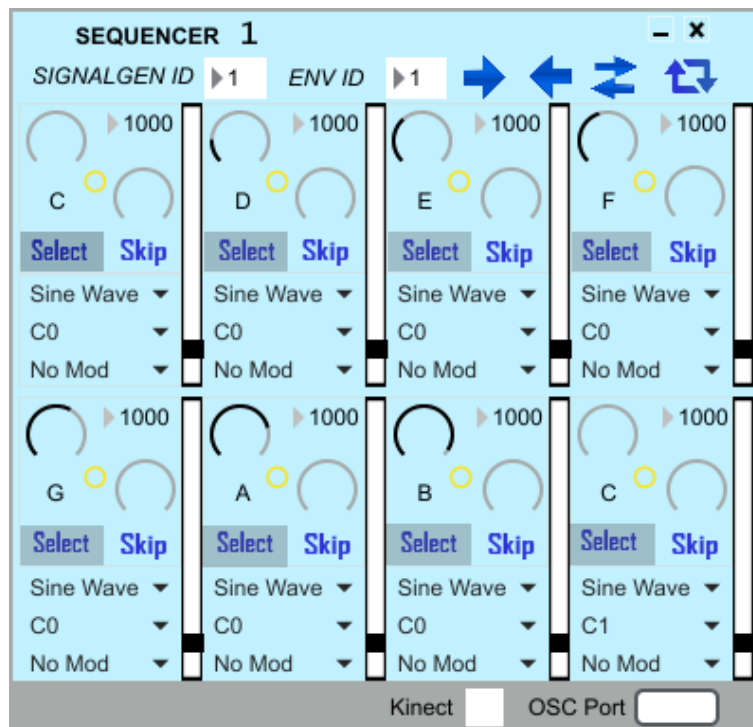
Il·lustració 15: Signal Generator

- Keyboard: Aquest mòdul te dos maneres de funcionament. La primera és polsant el *toggle* que tenim baix de *out*. Al fer-ho les teclat del piano s'enviarà a l'eixida la informació de nota i velocitat MIDI. Açò pot ser enviat a un mòdul de VST on s'utilitzarà un instrument virtual. Si el *toggle* es queda desactivat funcionarà el sintetitzador intern de l'ordinador produint so sense necessitat de connectar-se a cap altre mòdul. Es pot elegir l'instrument MIDI en la caixa de selecció, la octava en la part esquerra inferior, i el canal midi baix les teclat.



Il·lustració 16: Keyboard

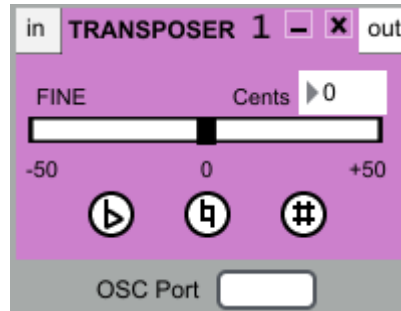
- Sequencer: Té la mateixa funció que un seqüenciador clàssic que controla un generador de funcions, en aquest cas el mòdul ja explicat de SignalGen. A la opció SignalGen Id elegim quin mòdul es desitja controlar, és a dir, posar el número del mòdul SignalGen corresponent. Tenim també huit requadres independents on es pot controlar el tipus d'ona a reproduir, la octava, la freqüència i duració de la nota, la modulació, a part d'elegir si es reproduceix la nota o silenci. Les seqüències es llancen amb les fletxes blaves de la part superior, directament, inversa, palíndrom o *loop* respectivament.



Il·lustració 17: Sequencer

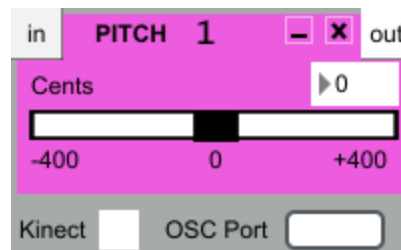
2.4.2 Mòduls que modifiquen el so

- Transposer: Amb aquest mòdul podrem fer més agut o més greu el so que es connecte a la seua entrada. Amb el símbol bemoll i sostingut podrem pujar o baixar mig to el nostre so, i després amb el *slider* podrem fer un ajust fi entre -50 i 50 centèsimes de to. Per a tornar a la posició inicial farem clic en el símbol del becaire.



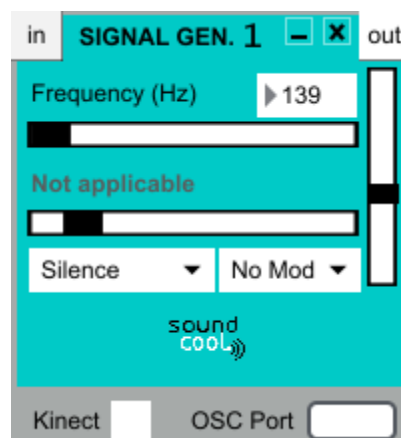
Il·lustració 18: Transposer

- Pitch: Aquest mòdul també modifica el so, però amb més marge. Mentre que els botons del Transposer anaven de semitò en semitò, el *slider* del mòdul Pitch pot fer un recorregut des de dos tons per baix (-400cents) fins a dos tons per dalt (400cents)



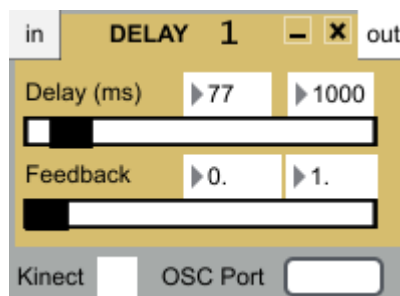
Il·lustració 19: Pitch

- Signalgenerator: A part de generar senyals, amb aquest mòdul també podem modular el àudio d'entrada. Si volem fer-ho haurem d'elegir entre una modulació RM, AM (podent elegir el seu factor MI) o FM (podent elegir el seu paràmetre FD, el seu índex de modulació i el seu rati harmònic)



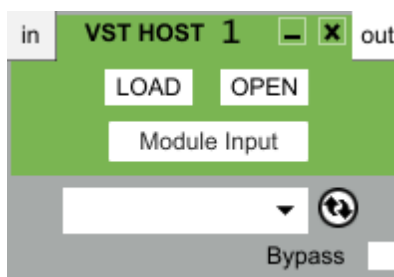
Il·lustració 20: Signal Generator

- Delay: També podem afegir un retràs en el so. El *slider* superior és per a controlar el retràs amb el qual la senyal va a ser presentada a la eixida del mòdul respecte a la senyal original. El *slider* inferior és el *feedback* que indica la proporció de senyal retardada que es tornarà a enviar a l'entrada per a produir nous retards.



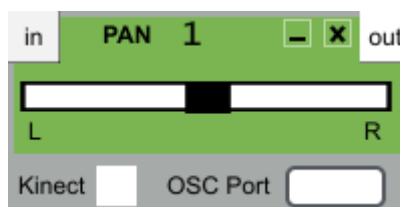
Il·lustració 21: Delay

- VST: Amb aquest mòdul es poden utilitzar instruments o efectes VST. Si te àudio connectat a l'entrada aleshores es podran utilitzar els efectes, si tenim un instrument MIDI connectat a l'ordinador o el mòdul Keyboard aleshores s'utilitzaran els instruments VST. Fent clic a *Load* es carreguen els efectes o instruments. La caixa de llistat inferior és per a elegir diferents tipus d'instruments MIDI, i el botó de refrescar per a renovar la llista d'instruments connectats.



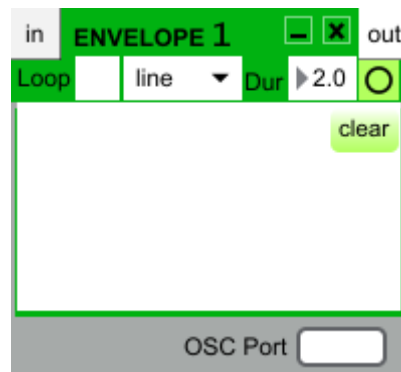
Il·lustració 22: VST

- Pan: Aquest mòdul consta d'un *slider* horitzontal per a controlar la distribució dreta/esquerra de l'àudio d'entrada, és a dir, la proporció de so que es desitja portar al canal dret i al canal esquerra.



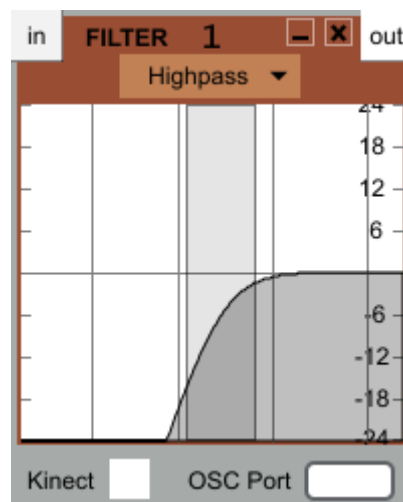
Il·lustració 23: Pan

- Envelope: Ens permet crear un control envoltent del volum mitjançant una successió de punts que s'elegeixen fent clic amb el ratolí. A mesura que es van creant s'uneixen automàticament per una línia de color negre. La coordenada horitzontal equival al temps i la y al volum. Clear es per a borrar tots els punts, la casella de *Loop* per a activarla o que només es reproduisca una vegada. La duració en segons també es pot elegir, al igual que el tipus d'envolvent entre *Curve* o *Line*.



Il·lustració 24: Envelope

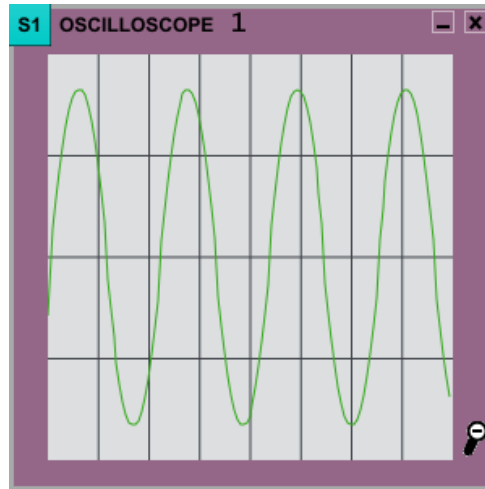
- Filter: Aquest mòdul permet filtrar die diferents formes el que rep per la seua entrada. En el menú desplegable es pot modificar el tipus de filtre que es desitja utilitzar. Es pot elegir entre filtre pas baix, pas alt, pas banda, elimina banda, filtre *peak*, *low* i *highshelf*, resonant i passa tot. En els corresponents tipus de filtres es pot també manipular el factor *Q* i la ganància.



Il·lustració 25: Filter

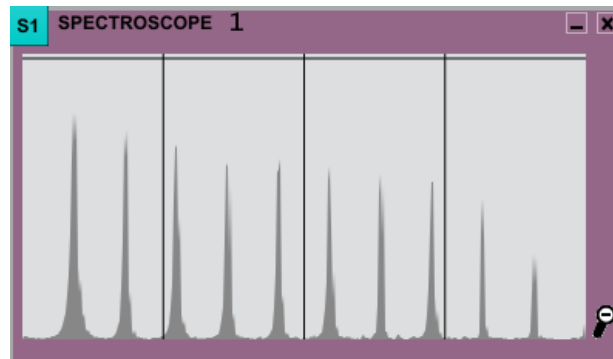
2.4.3 Mòduls que analitzen el so

- Oscilloscope: Es tracta d'un mòdul que funciona com un oscil·loscopi, és a dir, un aparell per a visualitzar les variacions de les senyals d'àudio al llarg del temps.



Il·lustració 26: Oscilloscope

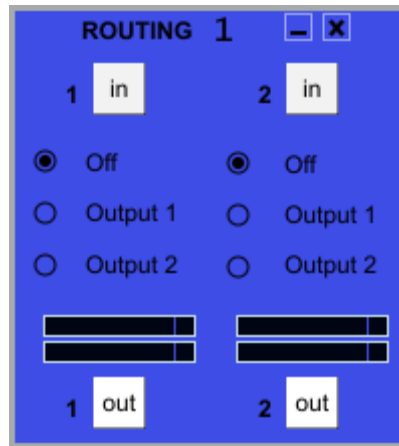
- Spectroscope: Aquest mòdul funciona com un espectroscopi, que es caracteritza per analitzar la senyal que rep i descomposar-la en funció de l'espectre de freqüències.



Il·lustració 27: Spectroscope

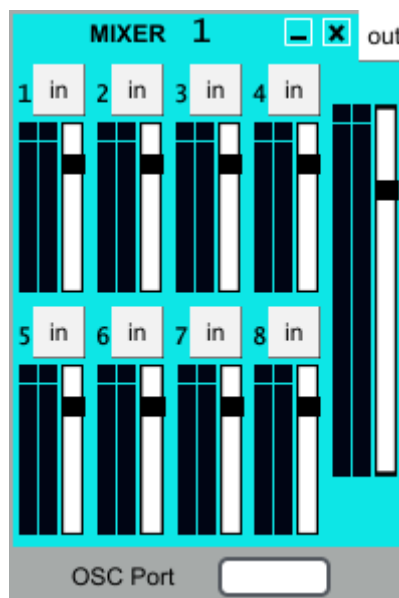
2.4.4 Mòduls que mesclen, graven o reproduïxen el so

- **Routing:** Aquest mòdul te dos entrades i dos eixides d'àudio. Cada una de les dues entrades que es volen connectar poden ser direccionades a l'eixida *Output 1*, a l'eixida 2 o bé deixar-les desconnectades en la posició *Off*. Cadascuna de les dos eixides te associada un *slider* que mostra el volum d'aquell que està connectat a la seua respectiva eixida.



Il·lustració 28: Routing

- **Mixer:** Aquest mòdul compta amb huit entrades d'àudio. Cadascuna d'elles te un control de volum individual i la suma de totes elles és la que ix a *out*, que al mateix temps te també un control principal de volum. Aquest mòdul és l'adequat per a portar més d'una senyal a l'eixida de àudio del nostre ordinador, al mòdul Speakers



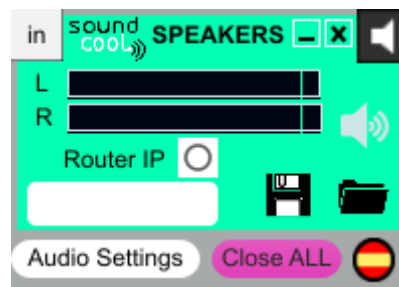
Il·lustració 29: Mixer

- Record: Aquest mòdul permet gravar el so que arriba a la seua entrada. Es pot elegir el temps màxim de gravació o parar la gravació en qualsevol moment. Després d'elegir on guardar l'arxiu de gravació i el seu nom es podrà fer clic a un botó *rec* per a començar a gravar.



Il·lustració 30: Record

- Speakers: Aquest és el mòdul indispensable per a poder reproduir el so, ja que correspon a l'eixida de àudio del ordinador. El botó de l'altaveu es pot polsar per a desactivar la eixida d'àudio. Es pot també guardar la configuració de mòduls disposada en eixe moment, i carregar l'arxiu *.Soundcool* guardat fent clic a les icones respectives. Es pot elegir idioma i paràmetres de l'àudio. Amb el botó de *Close ALL*, es tancaran tots els mòduls. Finalment podem saber la direcció IP de l'ordinador fent clic al botó de *Router IP*.



Il·lustració 31: Speakers

Capítol 3. Metodologia

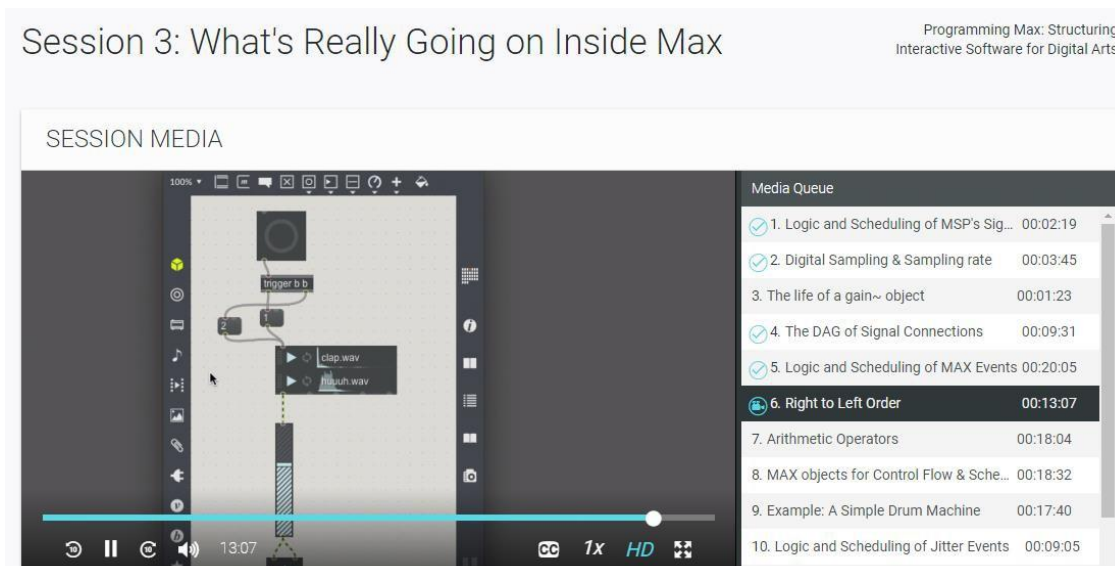
Una vegada explicat el projecte de *Soundcool*, el programari amb el qual està format i el funcionament dels seus mòduls operatius podem passar a parlar dels objectius d'aquest treball amb més coneixements. En aquest capítol els objectius es distribuïran en xicotetes tasques i s'explicaran les propostes per a afrontar-les, juntament amb el diagrama temporal que en un primer moment s'espera del treball.

3.1 Distribució de tasques

Per tal d'aconseguir dissenyar els nous mòduls de vídeo necessitarem uns coneixements previs del programa Max que anem a utilitzar per a programar-ho. Aquesta serà la primera tasca, la d'aprenentatge. Després aplicarem estos coneixements per a començar a solucionar xicotets errors en els mòduls de *Soundcool* ja creats. Una vegada fet açò començarem a fer els requisits previs per als nostres mòduls de vídeo que són la connexió entre ells i la exclusivitat de connexió àudio-àudio i vídeo-vídeo. Finalment podrem començar amb el disseny del mòdul VScreen, VSwitcher, VSampler i VColor.

3.1.1 Funcionament bàsic de Max i *Soundcool*

El primer pas que hem de donar per a arribar a poder crear nous mòduls és saber i entendre el funcionament bàsic de Max. Per tal d'aconseguir-ho, el que farem serà apuntar-nos a un curs MOOC (*Massive Online Open Courses*) impartit per Matt Wright [22] on s'explicarà el seu funcionament i els objectes des dels més bàsics com els explicats a l'apartat anterior fins als més complexos i concrets. Aquest curs, disponible en la pàgina web de *Kadenze* es nomena *Programming Max: Structure Interactive Software for Digital Arts* [23] i és el corresponent a la Il·lustració 32. Aquesta tasca tenim prevista que tinga una durada de com a màxim tres mesos, encara que a partir del primer mes ja es podrà anar corregint xicotets errors amb els coneixements que s'aniran adquirint.



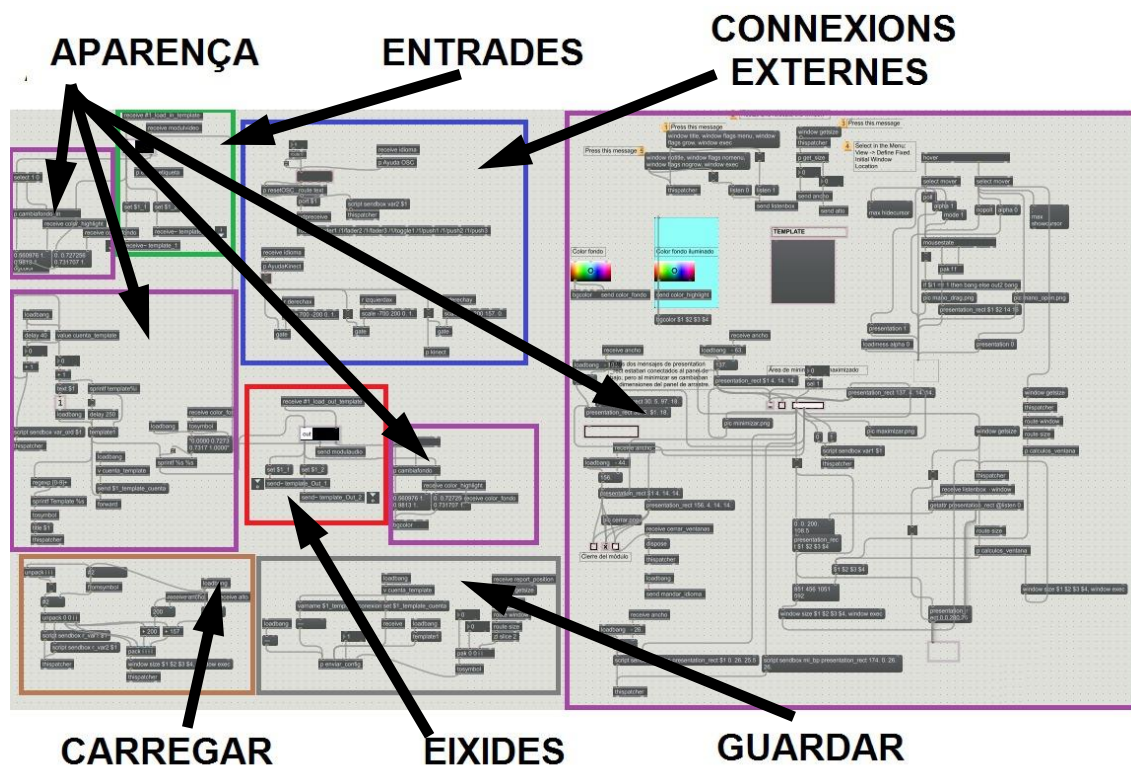
Il·lustració 32: Fragment del curs MOOC de Max

Durant tot el procés hem fet un total de deu sessions d'aprenentatge dividides en diversos vídeos curts on s'ha explicat tots els continguts. Aquestes sessions han sigut les següents:

- *Session 1: Introduction*
- *Session 2: Fundamental Elements*
- *Session 3: What's Really Going On Inside Max*
- *Session 4: Getting Mathematical: Arithmetic, Logic, Matrices*
- *Session 5: Structuring Non-Trivial Patches*
- *Session 6: Abstractions, Files, And How Max Works*
- *Session 7: Data Types In Depth*
- *Session 8: Interactivity*
- *Session 9: Max Programming Paradigms*
- *Session 10: Integrating MAX With Other Software And Languages*

Paral·lelament anirem aprenent la forma general dels mòduls de *Soundcool*. Per fer-ho existeix un mòdul base anomenat Template (vore la II·lustració 33). Aquest conté els elements bàsics per a el correcte funcionament de qualsevol mòdul. Està format per quatre grans blocs:

- El bloc que determina l'aparença del mòdul, marcat en morat a la II·lustració 33
- El bloc que fixa les entrades *in* en verd i les eixides *out* en roig, marcats a la II·lustració 33
- El bloc per a connexions de Kinect i dispositius mòbils, marcat en blau a la II·lustració 33
- El bloc per a guardar, i carregar les configuracions del mòdul, marcats en gris i marró respectivament a la II·lustració 33



II·lustració 33: Mòdul bàsic, amb els diferents blocs funcionals marcats

A grans trets, el bloc de l'aparença consta de diverses caixes de missatge que s'envien a l'objecte [thispatcher] el qual s'encarrega de fer modificacions visuals a la finestra del propi patcher. Canvia el color del mòdul, tant si no està seleccionat com si es selecciona (que aleshores es ressaltava d'un altre color). També fa uns càlculs per a determinar quantes finestres del mateix mòdul estan obertes per tal de numerar-lo.

El bloc per a les entrades i eixides consta de diversos [send] i [receive] els quals s'encarreguen d'enviar i rebre el so que passa de mòdul en mòdul. Amb açò tindrem un problema i és que si obrim dues finestres d'un mateix mòdul, els [send] i [receive] tindran els mateixos arguments, és a dir, encara que una senyal d'àudio només s'envia a un dels dos mòduls, en realitat ho rebran els dos ja que tindran el mateix [receive]. Açò es soluciona amb el valor #0 que és una variable especial en Max que resulta útil per a aquests casos. Aquesta variable prendrà un valor de quatre xifres únic entre tots els mòduls oberts en eixe moment. Açò farà que els missatges d'enviament i recepció siguin únics per a cada mòdul obert, encara que s'obriga més d'una finestra del mateix mòdul. D'aquesta manera els missatges no interferiran uns amb altres.

Per a les connexions amb dispositius mòbils l'objecte més important és [udpreceive] que serveix per a rebre missatges transmesos a través d'una xarxa utilitzant UDP (*User Datagram Protocol*). El número de port s'especifica al mateix mòdul i també s'haurà d'escriure al dispositiu mòbil amb el qual es vulga controlar el mòdul. Per al control mitjançant Kinect es té per defecte el port número 12345. Tant per a mòbils com per a Kinect, disposem de l'objecte [route] que s'encarrega de direccionar els missatges rebuts per al correcte funcionament del mòdul.

Finalment tenim el bloc per a guardar i carregar les configuracions. Per a carregar-les també disposem de caixes de missatge que envien informació a l'objecte [thispatcher] esmentat abans per tal de carregar informació important com la posició de la finestra o els mòduls connectats a l'entrada o eixida. Per a guardar la configuració, emmagatzemem aquesta mateixa informació i l'enviem mitjançant un [send] a un altre mòdul encarregat de convertir tota aquesta informació en un arxiu *.Soundcool*.

3.1.2 Milliores i correcció d'errors

Una vegada après com funcionen els mòduls de *Soundcool* i el funcionament bàsic de Max podem a poc a poc fer-los xicotetes millores. Aquesta tasca la començarem a fer passat el mes d'aprenentatge i durarà fins al final del projecte, ja que contínuament es revisen i es proposen millores per als mòduls. En aquest apartat s'esmentaran les primeres millores i els coneixements que es van adquirir al fer-les.

- Comentarem, per exemple, l'arrossegament de la finestra al complet. En un principi la finestra de qualsevol mòdul es podia arrossegar només fent clic al nom del mòdul, i la millora va ser que ho poguera fer fent clic a qualsevol part del mòdul.
- També hem solucionat detalls en el mòdul Direct Input per a afegir la opció de mutejar el so entrant, cosa que ens va obligar també a canviar l'aspecte visual del mòdul.
- Vam tindre que solucionar un error als mòduls de càrrega i guardat anomenats MLoad_Config; si des del mòdul Speakers es feia clic al botó de guardar o carregar i no s'elegia cap arxiu, un panell informatiu es quedava obert sense possibilitat de tancar-se.
- Per últim vam llevar el so del mòdul VPlayer, ja que aquest es tracta d'un mòdul únicament de vídeo i tot el so ha d'eixir pel mòdul Speakers

3.1.3 Connexió entre vídeos

En aquest punt ja comencem a treballar directament per a fer els mòduls finals. La primera tasca que anem a realitzar serà la de permetre la connexió entre diferents mòduls de vídeo. Açò és necessari perquè la connexió entre mòduls està optimitzada només per a so i amb aquesta els vídeos no es poden enviar.

El primer problema que tindrem que solucionar, doncs, serà el de la transmissió de vídeos. Açò farà que hagem de crear dos mòduls de Template, un de àudio i un altre de vídeo. La solució per a aquest problema va ser senzilla encara degut la inexperiència es va trobar després de tres setmanes de treball.

3.1.4 Exclusivitat de connexió àudio-àudio i vídeo-vídeo

Una vegada solucionat el problema de l'enviament de vídeo tindrem un problema en les connexions. Si fem clic en la eixida de un mòdul de àudio la resta de mòduls amb entrada canviaran de forma per a indicar que se'ls pot connectar aquest primer mòdul. Açò és un problema perquè no volem que s'envie àudio a les entrades de mòduls de vídeo com tampoc volem que s'envie vídeo a les entrades de mòduls d'àudio.

Com ja hem estudiat a fons la estructura bàsica dels mòduls amb l'anomenat Template hem considerat la forma més senzilla d'arreglar aquest problema en tan sols uns dies.

3.1.5 Vscreen

Quan aquest últim problema estiga resolt aleshores ja podrem començar amb el disseny del primer mòdul. Necessitarem en primer lloc un mòdul final per a visualitzar els vídeos anàleg al mòdul Speakers en la part de so de *Soundcool*

Aquest mòdul no tindrà un *out* ja que ell mateix és el final de la cadena de vídeo. Comptarà amb un botó de pantalla completa, i unes dimensions precises que seran obligatòries a tots els vídeos o imatges que es vulguen reproduir, ja que els objectes de visualització de Max no compten amb cap element de auto size.

Provarem el mòdul Screen amb els mòduls ja creats anteriorment, és a dir VDirect Input i el VPlayer.

3.1.6 Vswitcher

Per tal de seleccionar imatges o vídeos sense canviar els elements *in* i *out* i fer-ho ràpidament, necessitarem un mòdul que s'encarregue d'elegir el mòdul que enviarà el vídeo o la imatge al mòdul final Screen.

VSwitcher comptarà amb huit entrades i una sola eixida, i fent clic a un dels huit selectors es podrà canviar entre diferents fonts d'entrada ràpidament.

Aquest mòdul es provarà amb els mòduls creats anteriorment com el VDirect Input i el VPlayer per a les entrades, i amb el mòdul VScreen per a l'eixida.

3.1.7 VSampler

Després se'ns plantejarà dissenyar el mòdul més exigent de tots. Amb els coneixements adquirits de totes les tasques anteriors es farà un mòdul capaç de guardar diverses posicions de frame i fragments de *loop* de qualsevol vídeo i de disparar-los en qualsevol moment

Aquest mòdul comptarà amb 9 botons per guardar les posicions i 9 per a disparar-les. També comptarà amb 4 per a guardar el *loop* i 4 més per a disparar-lo

Les proves en aquest mòdul, com al VSwitcher, es faran amb els mòduls creats anteriorment com el VDirect Input i el VPlayer per a les entrades, i amb el mòdul VScreen per a l'eixida.

3.1.8 VColor

Quasi al final de l'entrega del Treball se'ns planteja dissenyar un últim mòdul capaç de canviar aspectes de la imatge com la lluminositat, contrast, tonalitat del color i saturació.

Quatre *sliders* faran la funció del canvi d'aquests paràmetres i un botó per tal de reestablir els paràmetres inicials seran els components d'aquest mòdul.

Amb els quatre mòduls anteriors VSampler, VScreen VPlayer i VDirect Input es podrà provar el funcionament d'aquest últim mòdul.

3.2 Diagrama temporal

El diagrama temporal plantejat per a totes aquestes tasques s'ha plasmat a la Taula 1.

L'aprenentatge del programa com a única activitat durarà uns tres mesos però durant tot el temps que estem treballant seguirà aquest procés d'aprenentatge. Com hem dit abans, la tasca de correcció d'errors també durarà fins el final ja que tenim un gran feedback i sempre hem tingut propostes noves de millora. La connexió entre vídeos durarà poc menys d'un mes i la exclusivitat de connexions també. El plantejament que tenim de temps per a cada mòdul nou és de dos mesos per mòdul. Durant els mesos de Agost i Setembre es planteja l'escriptura del Treball de Fi de Grau ja que en eixos mesos no estarem treballant com a becària. Finalment els últims dies se'ns ha proposat fer un últim mòdul que ha coincidit amb la part final de l'escriptura del Treball. El diagrama d'aquestes tasques ha resultat així:

	Desembre	Gener	Febrer	Març	Abril	Maig	Juny	Juliol	Agost	Setembre	Octubre	Novembre	Desembre
Aprenentatge	X	X	X	X	X	X	X	X			X	X	X
Correcció d'errors			X	X	X	X	X	X			X	X	X
Connexió entre vídeos				X									
Exclusivitat de connexió					X								
Vscreen					X	X							
Vswitcher						X	X						
Vsampler							X	X			X	X	
Vcolor													X
Redacció TFG									X	X		X	X

Taula 1: Diagrama temporal del nostre treball

Capítol 4. Desenvolupament

En aquest capítol explicarem detalladament tots els processos que hem seguit per a realitzar totes les tasques esmentades en l'anterior capítol de metodologia.

Començarem parlant dels objectes bàsics utilitzats en Max i després passarem a explicar les parts que tenen en comú tots els mòduls de *Soundcool* tals com l'aparença, les entrades i eixides, el guardat i les connexions amb Kinect i dispositius mòbils.

Després explicarem amb més detalls les solucions que hem trobat als problemes que tenia *Soundcool* en eixe moment. Parlarem de l'arrossegament de la finestra de tots els mòduls al complet, de la opció mute del mòdul Direct Input, d'un problema que teníem amb el carregat del conjunt de mòduls i de la eliminació del so als mòduls de vídeo.

Una vegada hagem après a utilitzar Max començarem a fer els primers passos que ens ajudaran a poder dissenyar correctament els mòduls de vídeo. Farem que es pugui passar la informació de qualsevol vídeo entre mòdul i mòdul, ja que *Soundcool* està optimitzat per a so i el vídeo no es pot transmetre, i després farem que els mòduls d'àudio no es puguin connectar amb els de vídeo i viceversa.

Finalment dissenyarem i provarem els quatre mòduls proposats: VScreen per a la visualització dels vídeos, VSwitcher per a commutar totes les entrades de vídeo que es connecten a ell, és a dir, per a elegir quina entrada serà la que s'envie al VScreen. El mòdul VSampler per a reproduir un vídeo des d'una frame específica i per a elegir i reproduir *Loops* al vídeo, i finalment VColor per a canviar els paràmetres del vídeo com el contrast, el to de color, la saturació i la lluminositat.

4.1 Aprenentatge

En aquest primer apartat explicarem els objectes bàsics utilitzats en Max i després passarem a explicar les parts que tenen en comú tots els mòduls de *Soundcool* tals com l'aparença, les entrades i eixides, el guardat i les connexions amb Kinect i dispositius mòbils.

Després explicarem amb més detalls les solucions que hem trobat als problemes que tenia *Soundcool* en eixe moment. Parlarem de l'arrossegament de la finestra de tots els mòduls al complet, de la opció mute del mòdul Direct Input, d'un problema que teníem amb el guardat i carregat del conjunt de mòduls i de la eliminació del so als mòduls de vídeo.

4.1.1 *Soundcool* i Max

Max és un entorn de desenvolupament gràfic utilitzat en l'àmbit musical i multimèdia desenvolupat per Cycling '74. Programació visual significa que, a diferència dels llenguatges de programació tradicional on es programa mitjançant codi, programarem mitjançant la interconnexió de caixes o objectes (com són coneguts en Max), que realitzen una determinada funció. En aquesta secció introduïrem aquells objectes que resulten bàsics per a la programació en Max [24]. No obstant açò, aquells objectes que apareguen únicament de forma puntual en els mòduls, seran introduïts en la descripció d'aquest mòdul.

Abans de començar a explicar els principals objectes farem una observació sobre la notació. Quan parlem d'objectes de Max parlem de unes caixes que al seu interior tenen escrit el nom de l'objecte que es vol utilitzar. Per a referir-se a aquests objectes en llocs de consulta com fòrums es fa posant el nom de l'objecte entre els símbols [], que pot representar visualment la caixa que el conté. Així doncs a partir d'ara tots els objectes de Max estaran representats amb aquesta notació.

4.1.1.1 [Caixa numèrica]

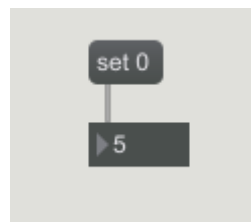
Les caixes numèriques serveixen per a enviar nombres a altres objectes. Poden ser de nombre enter o decimal, com s'indica a la Il·lustració 34. Aquesta característica pot detectar-se pel punt decimal, que apareixerà sempre en les caixes numèriques de nombre decimal. El seu valor pot modificar-se per mitjà de missatges, fent clic sobre elles i introduint un valor seguit de la tecla enter, o fent clic sobre elles i arrossegant el ratolí cap amunt o cap avall.



Il·lustració 34 Exemple de [caixa numèrica] de nombres enters (esquerra) i decimals (dreta)

4.1.1.2 Caixa de missatge

La caixa de missatge no és pròpiament un objecte, és a dir, no exerceix cap funció per sí sola. Utilitzem la caixa de missatge per a comunicar-nos amb altres objectes.

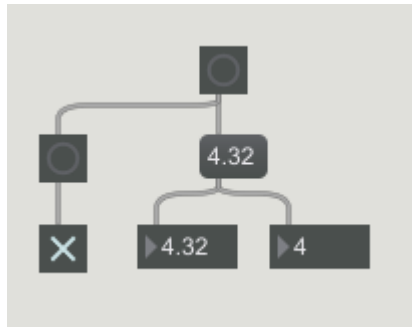


Il·lustració 35: Exemple de la caixa de missatge

Les caixes de missatge ens permeten manar missatges, número, llistes, etc. als objectes de Max. Per exemple, a la Il·lustració 35 s'envia el missatge set 0 a l'objecte [number], la qual cosa farà que en lloc del número 5 estiga el número 0 escrit dins d'ell. Finalment, cal dir que les caixes de missatge compten amb una entrada en el seu costat esquerra el qual ens permet enviar el missatge que contenen. Fer clic damunt del missatge també fa la mateixa funció. La seua entrada dreta serveix per a canviar el contingut escrit dins d'elles.

4.1.1.3 [bang]

El [bang] és un objecte molt utilitzat en Max que serveix per a enviar un impuls, encara que també s'empra el terme bang per a denominar aquests impulsos. La seua funció és la d'activar objectes, produir l'enviament d'un missatge o el valor d'una [caixa numèrica], etc. El seu efecte pot variar segons l'objecte sobre el qual s'aplique. En fer clic sobre un bang, aquest s'il·lumina. Qualsevol acció que arribe a un bang, farà els efectes d'un clic sobre bang i produirà un altre bang a la seua eixida.

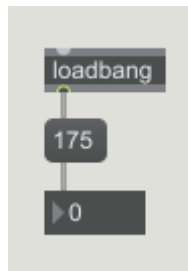


Il·lustració 36: Exemple de l'objecte [bang]

A la Il·lustració 36 podem veure un exemple d'ús de [bang]. Si fem clic en el que hi ha a la part superior, aquest enviarà un bang a la caixa de missatge i al [bang] inferior. El missatge serà enviat a les caixes numèriques de nombres enters i decimals, com es pot veure. D'altra banda, l'objecte [bang] situat més avall enviarà un altre bang a l'objecte [toggle], que es queda marcat.

4.1.1.4 [loadbang]

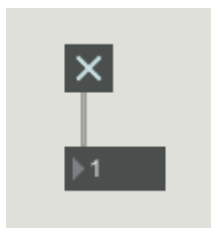
L'objecte [loadbang] produirà un bang en obrir-se el patcher, açò ens serà d'utilitat quan vulguem forçar una determinada situació inicial en obrir el patcher. A l'exemple de la Il·lustració 37 es pot veure com la [caixa numèrica] té el valor inicial 0. Si guardem aquest patch i el tornem a obrir, el loadbang enviarà un bang a la caixa de missatge que farà que la [caixa numèrica] passe a tindre un valor de 175.



Il·lustració 37: Exemple de l'objecte [loadbang]

4.1.1.5 [toggle]

El [toggle] o commutador és un objecte que ens proporciona una interfície gràfica senzilla per a commutar entre els valors 1 ó 0 (encés o apagat). Té l'aparença d'un quadrat buit quan està desactivat, i d'un quadrat amb una x en el seu interior quan està activat. El seu valor pot canviar-se per mitjà de missatges amb els valors 1 o 0, mitjançant bangs a la seua entrada, o fent clic sobre el propi interruptor. El canvi d'estat en un interruptor produeix també un bang a la seua eixida (siga un 0 o un 1, segons siga el seu estat).

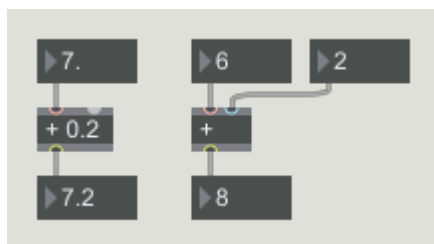


Il·lustració 38: Exemple de l'objecte [toggle]

A la Il·lustració 38 es mostra un [toggle] connectat a una [caixa numèrica]. Aquesta caixa mostra el valor 1, que és l'associat a un toggle encès. Podem veure, a més, la x a l'interior del [toggle].

4.1.1.6 Objectes d'operacions matemàtiques bàsiques

Freqüentment emprarem operacions matemàtiques tals com suma, resta, multiplicació i divisió. Max ens permet també realitzar operacions més complexes. Els objectes d'operacions matemàtiques tenen tots les mateixes característiques: compten amb dues entrades, que es corresponen amb els dos operands que intervenen en l'operació. El seu nom és igual al símbol matemàtic de l'operació a la qual representen, és a dir, +, -, * i /. Per a traure el resultat de l'operació, ha d'enviar-se un valor per la seua entrada esquerra. El canvi de valors en la seua entrada dreta no produirà resultat algun fins que s'envie algun valor per l'entrada esquerra, moment en el qual es prendrà l'últim valor que va arribar a l'entrada dreta.

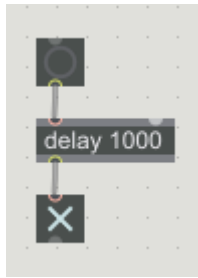


Il·lustració 39: Exemple d'ús de l'objecte [+]

A la Il·lustració 39 tenim dos exemples d'aquest objecte. A l'exemple de l'esquerra tenim el número 0.2 dins la caixa d'operació cosa que fa que, en activar el número 7, la caixa d'operació envie la suma d'aquests dos nombres. A l'exemple de la dreta està el resultat de la suma 6+2. El resultat canviarà sempre i quan canvie la caixa on està el número 6 i mai quan canvie la caixa del 2, com hem explicat abans.

4.1.1.7 [delay]

L'objecte [delay] ens permet retardar una certa quantitat de temps un bang que arribe a la seua entrada. Aquesta quantitat, s'especifica com a argument de l'objecte en mil·lisegons.

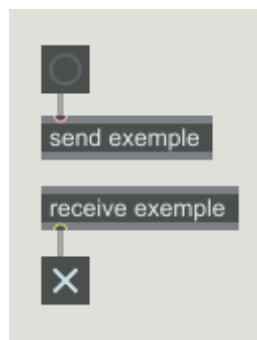


Il·lustració 40: Exemple de l'objecte [delay]

A l'exemple de la Il·lustració 40 podem comprovar que el [toggle] s'activa després d'1 segon d'haver fet clic al [bang] superior.

4.1.1.8 [send] i [receive]

Els objectes [send] i [receive] serveixen per a enviar dades d'un punt a un altre d'un *patcher* sense necessitat de connectar-los mitjançant cables, així com per a enviar dades entre diferents *patchers* i *subpatchers* que estiguen oberts en un mateix moment. S'identifiquen mitjançant noms, de manera que un [receive] de nom "x" rebrà tot allò que s'envia a través d'un [send] amb el mateix nom.



Il·lustració 41: Exemple dels objectes [send] i [receive]

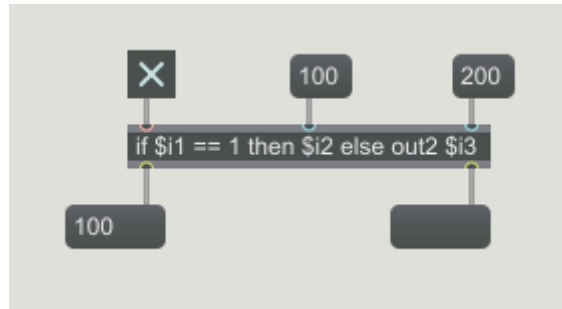
A la Il·lustració 41 veiem com en fer clic al [bang] que es troba sobre l'objecte [send], un bang s'ha rebut en l'objecte [receive]. Açò s'ha enviat sense la necessitat de connectar-los mitjançant cables ja que estan identificats amb el mateix nom "exemple"

4.1.1.9 [if]

Com en qualsevol llenguatge de programació, en Max també disposem de la tradicional sentència *if*, la qual ens permetrà programar decisions condicionades. La sintaxi d'aquest objecte és:

- if condició then acció else acció

En la formació de condicions es farà ús de variables preses des de les dues entrades de l'objecte [if]. Observem el següent exemple.



Il·lustració 42: Exemple de l'objecte [if]

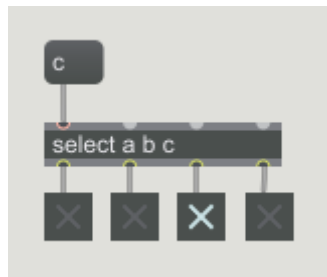
A la Il·lustració 42 tenim la condició:

- “si l’enter que entra per l’entrada 1 és igual a 1, aleshores trau per la primera entrada el número 100. En cas contrari trau per la segona entrada el número 200”

Ací vegem com hem fet clic una vegada a [toggle] i tenim al missatge connectat a la primera entrada el número 100, i al segon missatge res. Si férem clic una altra vegada, aquest segon missatge tindria el número 200 al seu interior.

4.1.1.10 [select]

L'objecte [select] ens permet traure bangs a les seues eixides segons la coincidència amb valors establerts com a arguments. Un objecte [select] tindrà tantes eixides com a arguments introduïm, a més d'una eixida per a traure un bang quan no es coincidisca amb cap dels arguments. Els valors entraran sempre per l'entrada de l'esquerra, mentre que la resta d'entrades serviran per a canviar els valors per defecte dels patrons de coincidència. Observem un cas típic en el següent exemple:



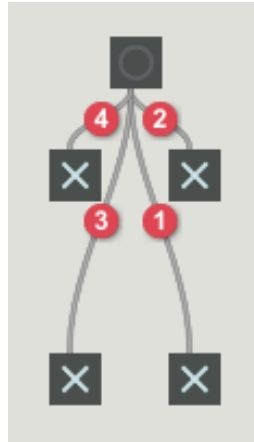
Il·lustració 43: Exemple de l'objecte [select]

En el nostre exemple de la Il·lustració 43, tenim un [select] que buscarà coincidències amb les lletres a b i c. Si es troba una coincidència amb la lletra a, s'enviarà un bang per la primera eixida (ja que la a s'ha col·locat com a argument en primera posició) mentre que si arriba una b o c, s'enviarà un bang per la segona eixida o per la tercera respectivament. En aquest cas, s'ha connectat a l'entrada del [select] un missatge amb una c que arriba a l'entrada del [select], es produeix la coincidència amb el seu tercer argument i, per tant, es mana un bang a la segona eixida.

4.1.1.11 Seqüència d'execució de Max

L'últim concepte que comentarem en aquesta breu introducció a Max és la seqüència d'execució de missatges en Max. En Max, els missatges s'executen de forma seqüencial, tot i que en ocasions ens puga

semblar que s'executen de forma simultània. Així doncs, l'ordre d'execució natural en Max és de dreta a esquerra i d'a baix a dalt. Observem la següent figura.



Il·lustració 44: Exemple per a la seqüència d'execució de Max

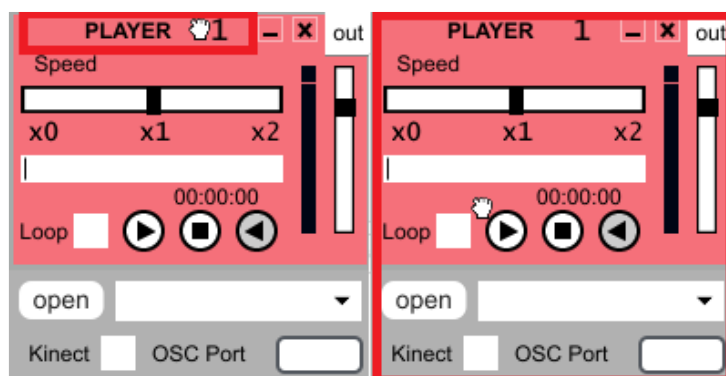
En aquest exemple de la Il·lustració 44 podem veure un [bang] connectat a quatre [toggle]. Cada cable que els uneix està numerat. Els [toggle] 3 i 4 estan situats en la mateixa línia un damunt de l'altre, així com els 1 i 2. L'ordre en el que el [bang] enviarà el seu bang a la resta d'objectes serà el marcat pels cables: 1,2,3,4 ja que el [toggle] 1 és el que està situat més avall i més a la dreta, el 2 és el següent més a la dreta, el 3 és el que queda més avall d'entre ell i el 4 i finalment s'activarà el 4.

4.1.2 Correcció d'errors

Amb el funcionament de Max clar, ja podem solucionar alguns errors. En aquest apartat explicarem els problemes més importants que ens van sorgir abans de començar a treballar en els nous mòduls.

4.1.2.1 Arrossegament de finestra

Comentem l'arrossegament de la finestra al complet. En un principi la finestra de qualsevol mòdul es podia arrossegar només fent clic al nom del mòdul com es mostra a la Il·lustració 45, però se'ns va proposar que el mòdul es poguera arrossegar arrastrant des de qualsevol punt de la finestra on no hi haguera un element actiu.



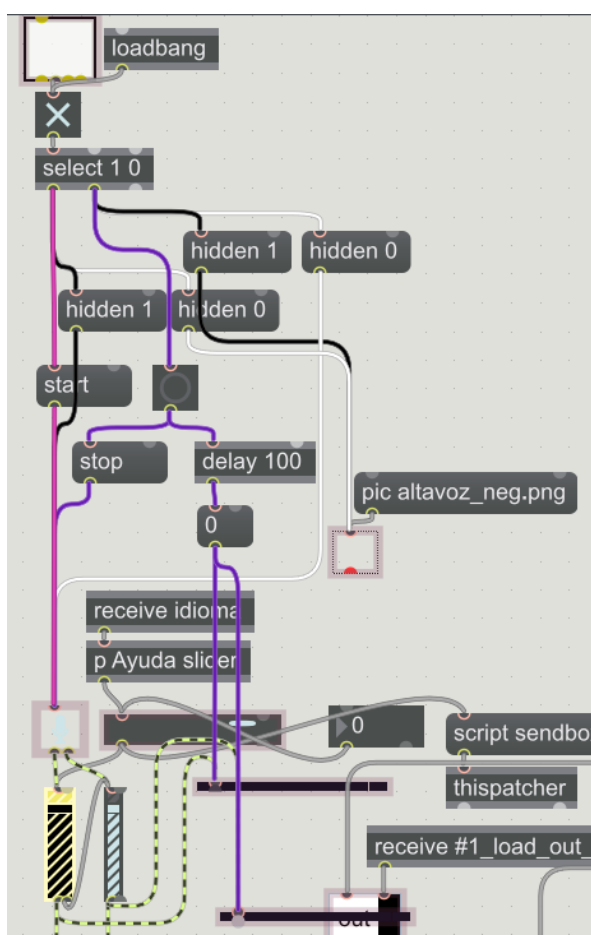
Il·lustració 45: Quadre d'arrossegament inicial (esquerra) i final (dreta)

Dins del mòdul podem trobar l'objecte [panel] que és l'encarregat d'aquesta acció d'arrossegament. El que farem serà canviar les seues mesures per a que coincidisquen amb les de la finestra completa. Açò portarà alguns problemes ja que en un primer moment els botons, *toggle* i *sliders* no reaccionaran als clics, ja que el panell d'arrossegament estarà al pla més avançat. Canviarem la distribució de plans amb les opcions “*Send to back*” i “*Bring to front*” fins que es pugui arrastrar tota la finestra al complet menys els elements actius.

D'altra banda també llevarem alguns objectes [loadbang] que s'encarregaven de posar les mesures del nostre objecte [panel] com estaven en un primer moment. Açò passava tant a l'obrir el mòdul com al minimitzar-lo i tornar-lo a expandir, ja que eren uns mecanismes de seguretat per a que el mòdul no canviara mai de forma.

4.1.2.2 Opció mute a Direct Input

També hem solucionat detalls en el mòdul Direct Input per a afegir la opció de mutejar el so entrant, cosa que ens va obligar també a canviar l'aspecte visual del mòdul ja que el canvi de mute a so es farà fent clic sobre la imatge. A la Il·lustració 46 els cables que uneixen els objectes tenen diferents colors per a diferenciar-se i millorar la explicació.



Il·lustració 46: Secció de Direct Input modificada

Comencem amb un [loadbang] perquè el mòdul s'obri amb opció de gravació. En fer clic en l'objecte [ubutton], el [toggle] indicarà si passem a la manera *grab* (select 1) o mute (select 0).

Tenim quatre missatges que comencen amb la paraula *hidden*. Els utilitzarem per a amagar (amb l'argument 1) o mostrar (amb l'argument 0) objectes de la presentació.

Els cables morats són els corresponents al mode *mute* on s'amagarà la imatge de l'altaveu i es mostrarà la del micròfon amb els missatges *hidden* explicats abans.

Enviam un missatge *Stop* a l'objecte [ezadc] per a parar la gravació i després d'açò enviem un 0 perquè en els objectes [live.ficar] no es quede gravat l'últim valor de volum.

Els cables rosa són els corresponents al mode de gravació, on s'amagarà la imatge del micròfon i es mostrarà la de l'altaveu. S'activarà el missatge *start* a l'objecte

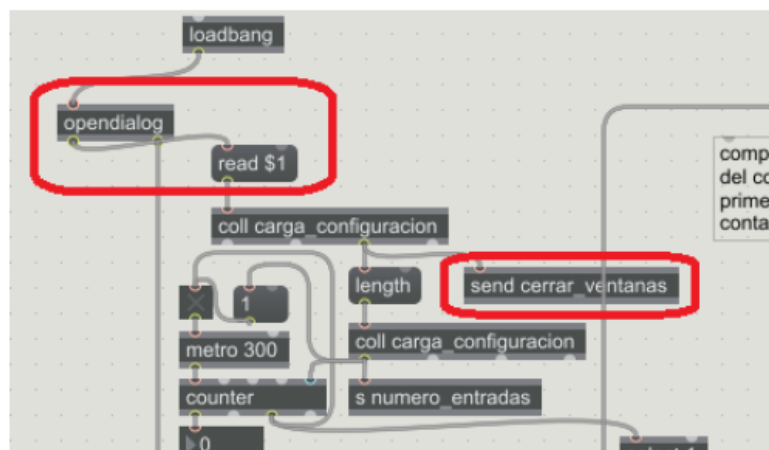
De totes formes aquesta no va ser la solució finalment adoptada. Parlarem de la solució final al capítol cinc, a l'apartat de millores i correccions.

4.1.2.3 Càrrega de mòduls

Vam tindre que solucionar un error al mòdul de càrrega anomenat MLoad_Config; si des del mòdul Speakers es feia clic al botó de carregar i no s'elegia cap arxiu, un panell informatiu "LOADING" es quedava obert sense possibilitat de tancar-se i a més es tancaven totes les finestres dels mòduls oberts en eixe moment.

En un principi el procediment que es seguia era primer tancar tots els mòduls oberts i després llegir el arxiu .Soundcool. Vam canviar algunes coses per tal d'arreglar açò, com es pot vore a la Il·lustració 47

Afegirem l'objecte [opendialog] per a que ens indique quan es tanca la finestra sense haver elegit cap arxiu. Si s'elegeix un arxiu aleshores per el primer output de l'objecte eixirà la ubicació d'aquest que llegirà l'objecte [coll] per a carregar la configuració després d'enviar un [send] a tots els mòduls oberts en eixe moment per a tancar-los. Si es tanca la finestra sense haver elegit cap arxiu aleshores pel segon output de [opendialog] eixirà un bang que utilitzarem per a tancar el panell "LOADING" mitjançant el missatge *dispose* a l'objecte [thispatcher]

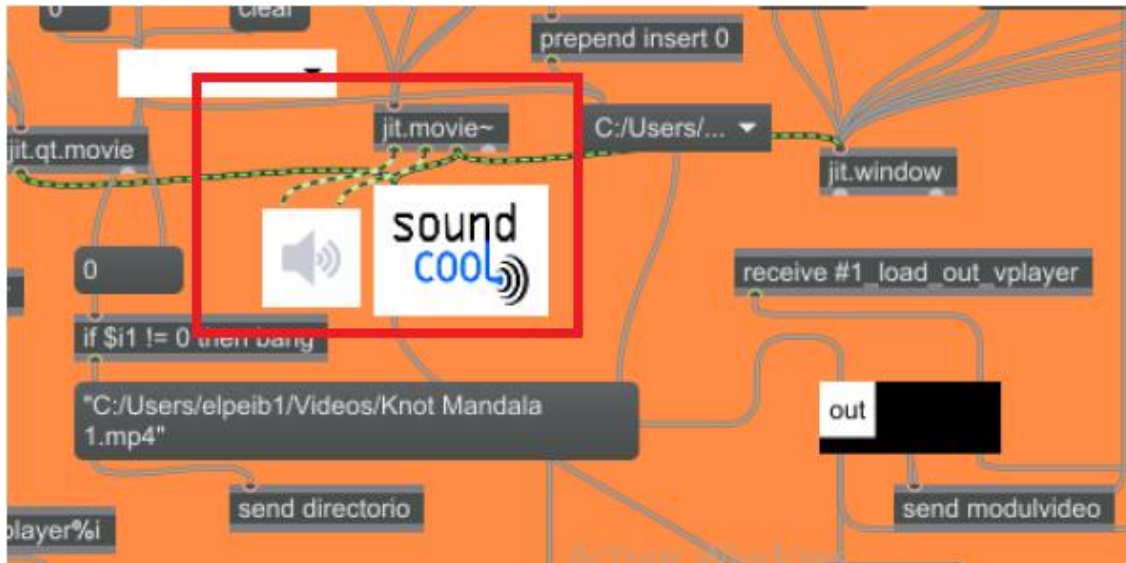


Il·lustració 47: Secció de MLoad_Config modificada

4.1.2.4 Llevar el so als mòduls de vídeo ja existents

A *Soundcool* el so es tracta i es reproduïx únicament mitjançant els mòduls de so, de la mateixa manera que els mòduls de vídeo i imatge només es tracten i reproduïxen mitjançant els mòduls corresponents.

El problema que existia era que els dos mòduls de vídeo reproduïen independentment el seu propi so. A la Il·lustració 48 podem veure la solució proposada:



Il·lustració 48: Secció de MVPlayer i MVDirectInput modificada

En un principi existia l'objecte [jit.movie] per a reproduir la imatge i el so dels dos mòduls. La solució que vam proposar va ser canviar aquest objecte per [jit.movie~] que és bàsicament el mateix però amb dues out més per on ix el so i es pot modificar. Connectarem l'objecte [ezadc] per a poder mutejar l'eixida de so.

De totes formes, a l'igual que al mòdul de Direct Input, aquesta no va ser la solució finalment adoptada. Parlarem de la solució final al capítol cinc, a l'apartat de millores i correccions.

4.2 Solucions comunes a tots els mòduls

En aquest punt ja tindrem els coneixements suficients com per a no només aprendre i corregir, sinó per a començar a dissenyar des de zero les solucions que faran que els nostres futurs mòduls funcionen correctament.

Primer tindrem que fer que la informació en forma de vídeo o imatge es pugui passar de mòdul a mòdul mitjançant [send], [receive] i la variable #0. Després posarem unes restriccions per tal que la informació de vídeos o imatges no es pugui mesclar amb la informació d'àudio.

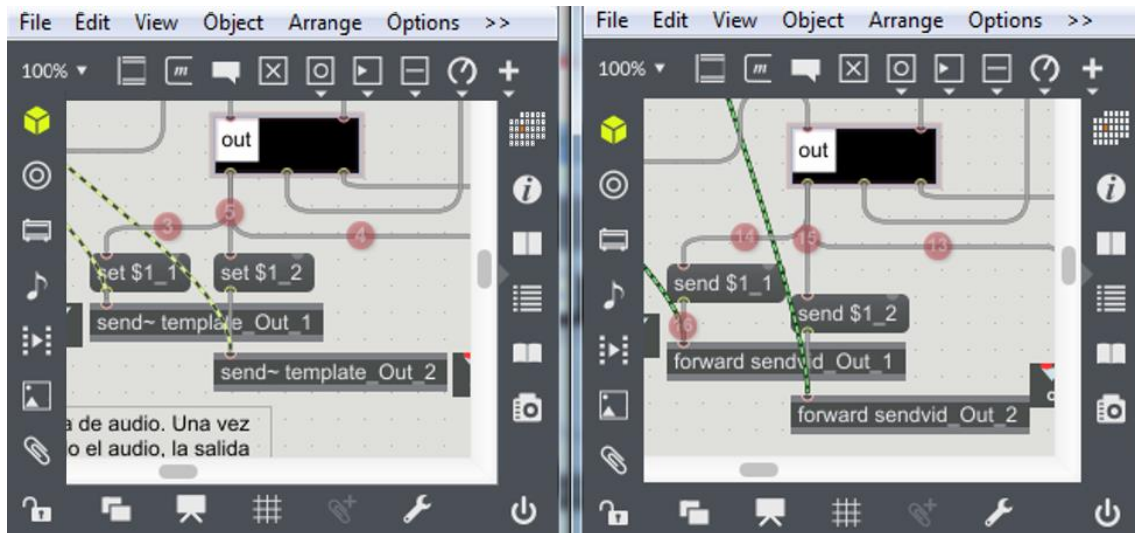
4.2.1 Interconnexió entre mòduls de vídeo

En aquest punt ja comencem a treballar directament per a fer els mòduls finals. La primera tasca que anem a realitzar serà la de permetre la connexió entre diferents mòduls de vídeo. Açò és necessari perquè la connexió entre mòduls està optimitzada només per a so i amb aquesta els vídeos no es poden enviar.

Abans d'explicar la nostra proposta de solució anem a explicar la diferència entre els objectes [send~], [send] i [forward]. El primer envia només senyals, el segon només envia dades, i el tercer envia qualsevol

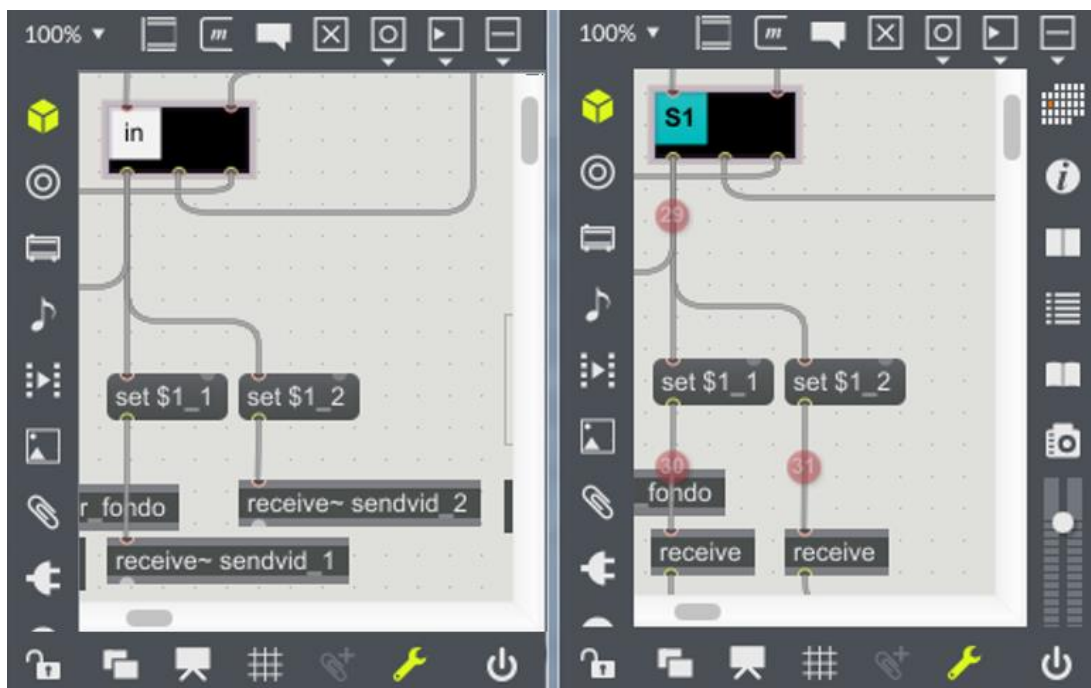
informació. No s'utilitza [forward] per a tot perquè [send~] i [send] estan optimitzats per a senyals i dades respectivament.

Per a enviar àudio com es veu a la Il·lustració 49, és a dir per a l'eixida des de *out*, existeix en un primer moment l'objecte [send~] amb una caixa de missatge amb la paraula *set* que indica on s'enviarà eixa informació. L'equivalent de vídeo, per tant, serà l'objecte [forward] amb una caixa de missatge amb la paraula *send*.



Il·lustració 49: Enviament d'àudio (esquerra) i vídeo (dreta)

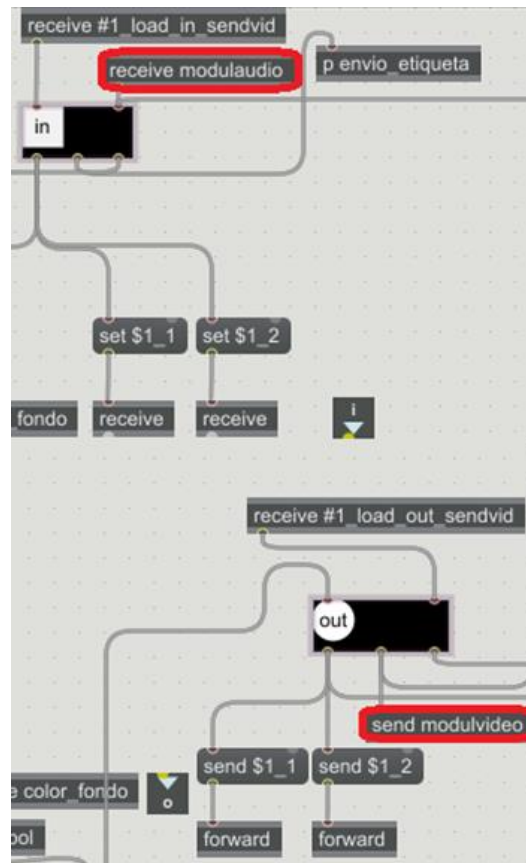
De la mateixa manera com podem veure a la següent Il·lustració 50 per a la recepció de vídeos seguirem el mateix mètode. Canviarem l'objecte [receive~] per l'objecte [receive] per a que pugui rebre la informació de vídeo i imatge correctament. Els dos objectes necessiten un missatge *set* per a saber de qui reben la informació.



Il·lustració 50: Recepció d'àudio (esquerra) i vídeo (dreta)

4.2.2 Exclusivitat de connexió vídeo-vídeo i àudio-àudio

Una vegada solucionat el problema de l'enviament de vídeo tindrem un problema en les connexions. Si fem clic en la eixida de un mòdul de àudio la resta de mòduls amb entrada canviaran de forma per a indicar que se'ls pot connectar aquest primer mòdul. Açò és un problema perquè no volem que s'envie àudio a les entrades de mòduls de vídeo com tampoc volem que s'envie vídeo a les entrades de mòduls d'àudio.



Il·lustració 51: Mòdul de vídeo exclusiu per a vídeo

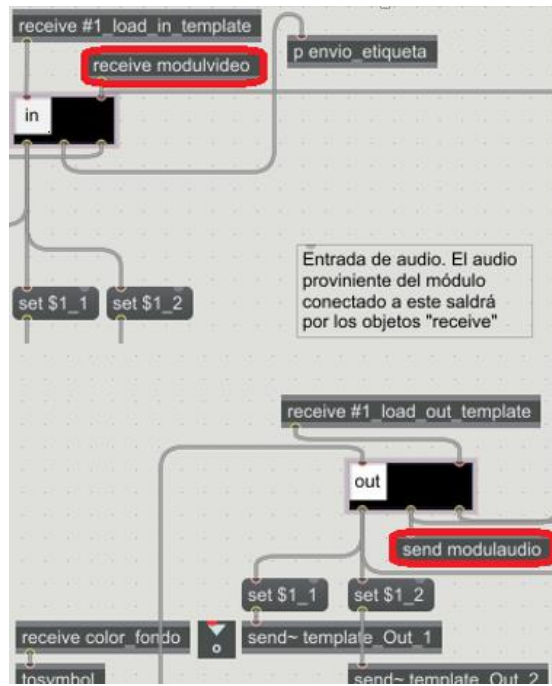
Com ja hem estudiat a fons la estructura bàsica dels mòduls amb l'anomenat Template hem considerat la forma més senzilla d'arreglar aquest problema en poc temps.

El requadre on està escrit *in* al mòdul és un *bpatcher*, açò vol dir que té la funcionalitat d'un *subpatcher* però que en lloc d'estar dins del *patcher* principal és un altre *patcher* el qual es crida. És similar a una *abstraction* en la diferència de que tenim la possibilitat de visualitzar algun dels seus components en el *patcher* principal, com en aquest cas el botó *in* i el botó *out*. Aquest *bpatcher* té una funció que fa que, si per la seua segona entrada passa el número 1, aleshores no està disponible per a cap connexió.

Açò en un primer moment serveix per a que un mòdul no es pugui connectar amb ell mateixa ja que al fer clic en *out*, de la seua segona eixida s'envia un 1 que entra a la segona entrada de *in* fent així impossible la auto connexió.

Hem aprofitat aquest funcionament per solucionar el problema de la connexió exclusiva entre mòduls de vídeo i de àudio amb l'ajuda dels objectes [send] i [receive]

La Il·lustració 51 és una secció d'un mòdul de vídeo. A la segona eixida del seu *out* tenim un [send] amb l'argument modulvídeo que indica que s'ha fet clic al *out* d'un mòdul de vídeo.



Il·lustració 52: Mòdul d'àudio exclusiu per a àudio

Amb açò el que passa és que els mòduls de àudio, com podem vore en la Il·lustració 52, reben en la seua segona entrada de *in* un 1 de l'objecte [receive] amb argument modulvídeo.

D'aquesta forma, mai un mòdul de vídeo es podrà connectar al *in* d'un mòdul d'àudio.

I de la mateixa manera, quan es vol connectar un mòdul d'àudio, s'envia mitjançant [send] modulaudio a tots els mòduls de vídeo fent que no es puga connectar a cap d'ells.

Així doncs amb quatre objectes [send] i [receive] col·locats a tots els mòduls existents hem fet que la connexió entre mòduls d'àudio i de vídeo siga exclusiva entre ells mateixa.

4.3 Disseny de nous mòduls

En aquest punt ja tenim tot el necessari per començar a dissenyar els nous mòduls. Ja estem familiaritzats amb Max i ja estan preparats els enviaments i la exclusivitat entre mòduls de vídeo.

Començarem dissenyant el més fàcil, el mòdul VScreen que serveix únicament per a rebre el contingut audiovisual i mostrar-lo en pantalla completa.

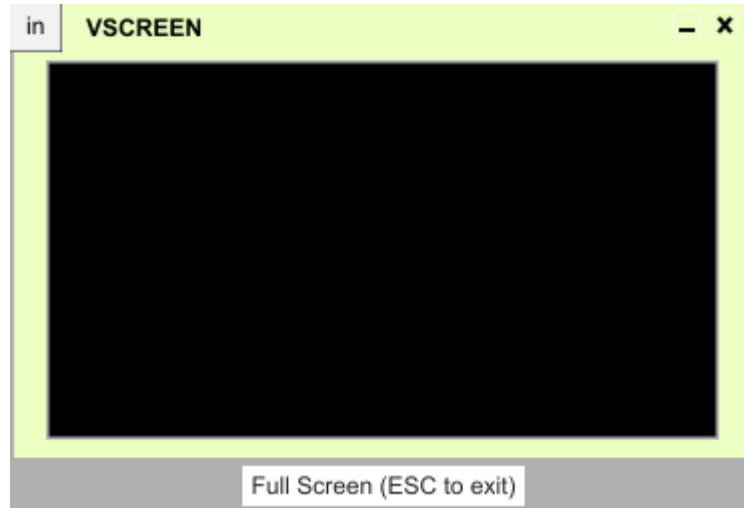
Després dissenyarem el VSwitcher que farà de commutador entre huit entrades de vídeo o imatge diferents per a passar la seleccionada al mòdul VScreen.

El mòdul VSampler serà el més complex dels quatre i servirà per a reproduir vídeos des d'una frame en concret o crear *loops* dins del propi vídeo.

Finalment el mòdul VColor s'utilitzarà per a canviar propietats de la imatge com la lluminositat, contrast, saturació i to de color.

4.3.1 VScreen

El primer dels quatre mòduls a dissenyar serà VScreen, el qual podem veure en mode presentació en la Il·lustració 53. Els elements que es poden veure són la entrada *in*, els botons de minimitzar i tancar, la pròpia pantalla de visualització i un botó per a veure-la a pantalla completa



Il·lustració 53: Finestra de presentació de VScreen

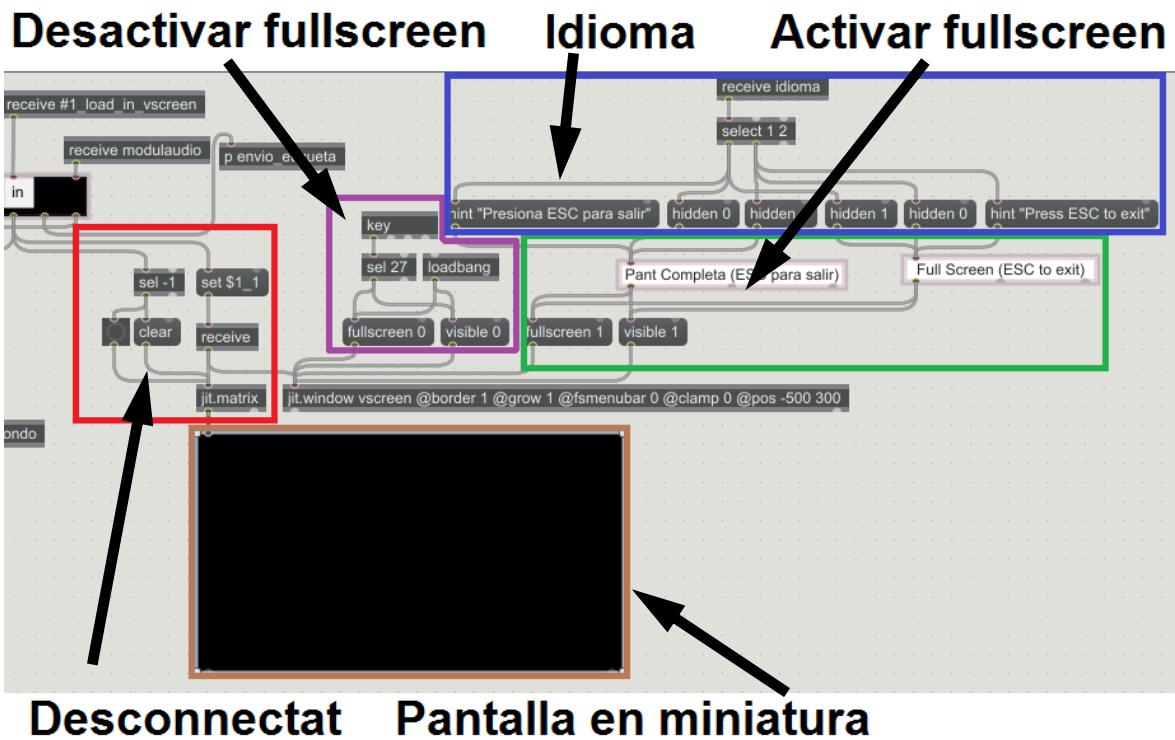
Els objectes principals que anem a utilitzar seran `[jit.pwindow]` per a visualitzar el contingut multimèdia en miniatura, és a dir, en la pròpia pantalla del mòdul requadrat en color marró a la Il·lustració 54, i l'objecte `[jit.window]` encarregat de la funció de pantalla completa.

Aquest mòdul consta a grans trets de cinc funcionalitats: Idioma, *fullscreen on*, *fullscreen off*, finestra en miniatura, i pantalla en negre al desconnectar-se.

Tenim la secció de l'idioma requadrada en blau, encarregada de que el text de pantalla completa es mostre en anglès o en castellà mitjançant l'objecte `[select]`

La secció de *fullscreen on*, en verd a la Il·lustració 54, s'encarrega d'enviar uns missatges a `[jit.window]` ordenant primerament que es pose en mode pantalla completa i després que es mostre. Per contra, la secció de *fullscreen off* fa que al pulsar la tecla `esc`, el número 27 per a l'objecte `[key]`, la pantalla completa es desactive i no es mostre.

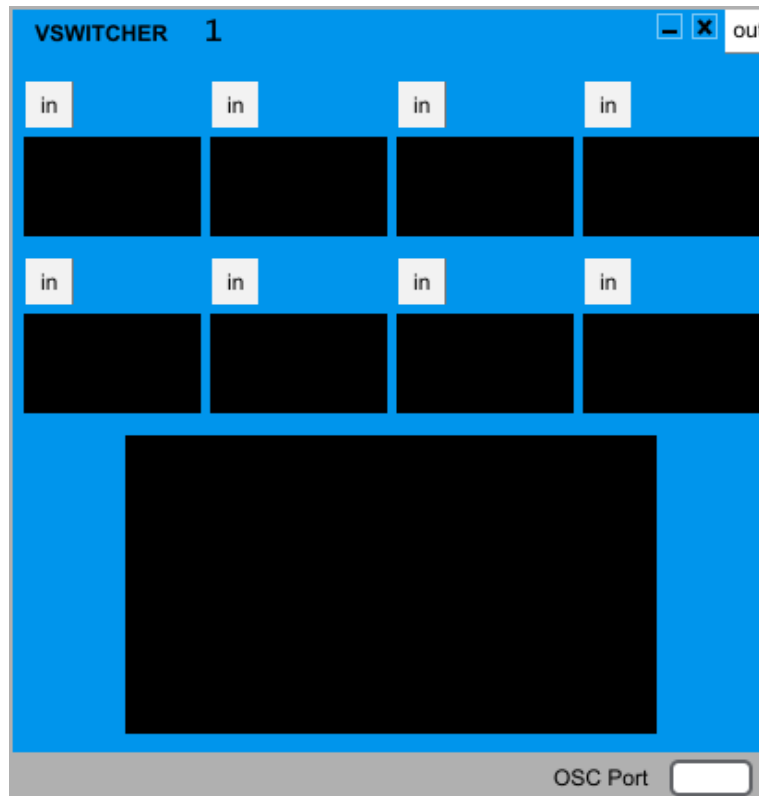
Per últim afegirem uns elements requadrats en roig per tal que, quan un mòdul es desconnecte de VScreen, la pantalla no es quede congelada amb l'últim fotograma rebut. Quan del *bpatcher in* isca el número -1 per indicar que s'han desconnectat d'ell, enviarem el missatge *clear* a `[jit.matrix]` per tal de passar la pantalla a negre



Il·lustració 54: Secció del mòdul VScreen.

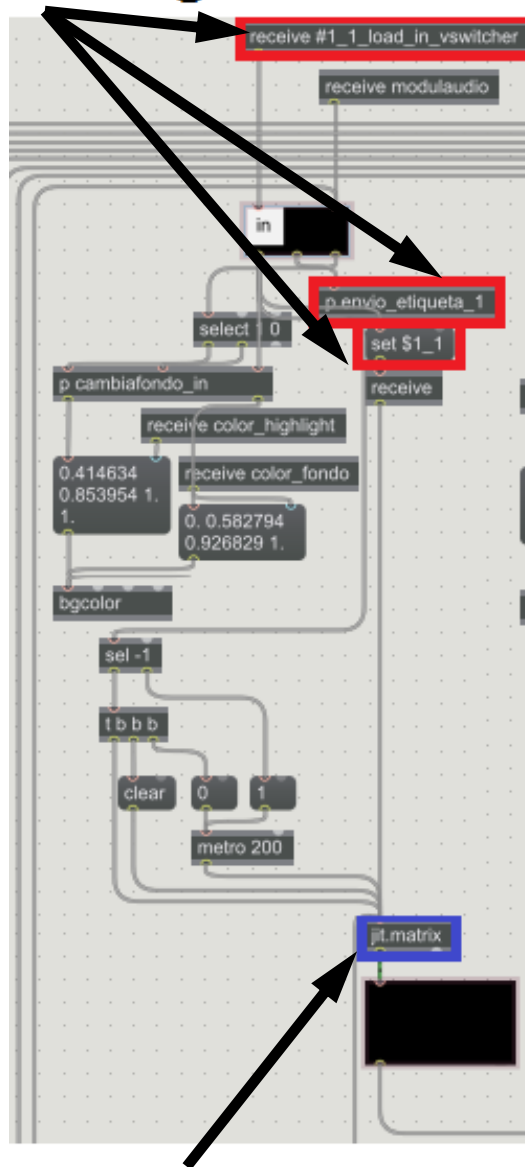
4.3.2 VSwitcher

Passem al mòdul VSwitcher, que farà de commutador entre huit entrades diferents per a passar la seleccionada al mòdul VScreen. Tenim el seu mode de presentació en la Il·lustració 55.



Il·lustració 55: Finestra de presentació de VSwitcher

Distinguir entrades



Enviar imatges

Il·lustració 56: Secció del mòdul VSwitcher

En primer lloc necessitarem insertar les huit entrades *in* al mòdul, i per a que funcionen independentment les haurem de nomenar unes diferents a altres. A l'exemple de Il·lustració 56 veiem marcat en roig les instàncies que faran aquesta funció. En aquest cas estem a la primera entrada així que tant el *subpatcher* [receive] i el missatge set acabaran amb el número 1. Els objectes de la resta d'entrades tindran un acabament diferent del 2 al 8 segons la seua posició.

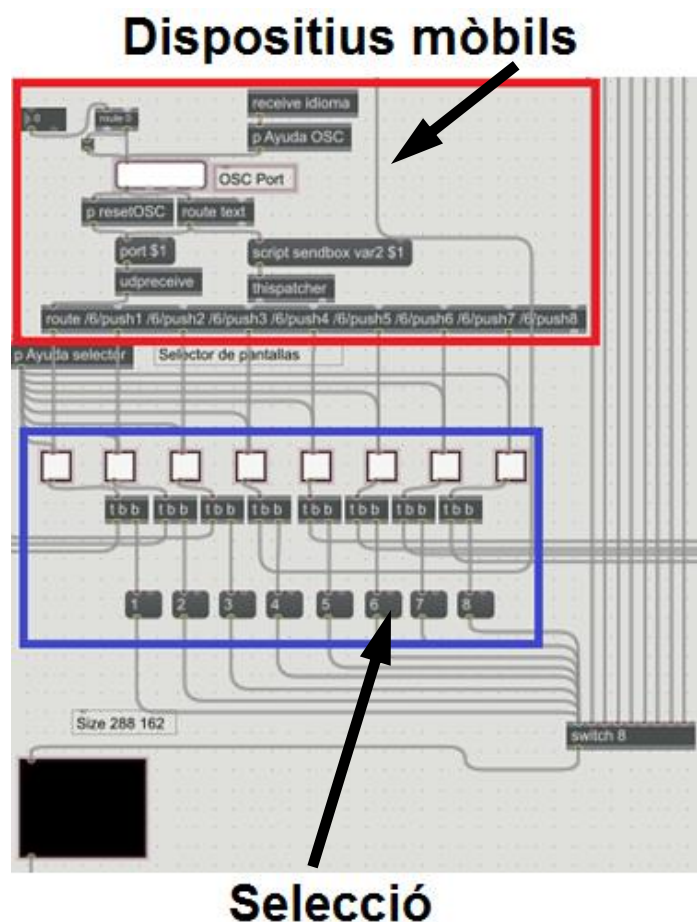
L'objecte [matrix] que es veu marcat en blau serveix per solucionar un problema que passava al carregar imatges i connectar el mòdul a VScreen. A l'enviar-se una imatge aquesta es mostra en la finestra xicoteta de [pwindow], però al seleccionar-la, el [pwindow] que la deuria mostrar no canviava. Açò passava perquè la imatge s'envia una vegada per a mostrar-se i després no es segueix enviant. Aquest problema l'hem solucionat amb l'objecte [jit.matrix] que sí és capaç de fer-ho quan rep un bang a l'entrada i ja te la imatge carregada.

D'altra banda tenim la secció del mòdul que enviarà el contingut d'una entrada o d'una altra a l'eixida *out* com es veu a la Il·lustració 57 . Aquesta secció té dues funcionalitats principals.

La primera, marcada en roig és la part que permet controlar-ho mitjançant dispositius mòbils. A l'insertar el mateix número de port que el que s'elegeix per al dispositiu mòbil, els huit botons de selecció es poden controlar amb l'objecte [route] des dels mòbils

La part blava es la part de selecció en sí. Els quadres blancs són objectes [panel] que en mode presentació es tornen transparents. D'aquesta manera, al fer clic en una de les huit imatges aquesta és la que ix seleccionada. Els passos són tres: es fa clic a [panel], s'envia el número corresponent a l'objecte [switch] per a que done pas només a la seua entrada corresponent i s'envia un bang al [matrix] corresponent per a que envie la imatge.

Finalment l'objecte [switch] s'encarrega de commutar les huit entrades i elegir només una segons el número que ha rebut anteriorment per a enviar-la a l'objecte [pwindow] de previsualització i a l'eixida *out* del mòdul.

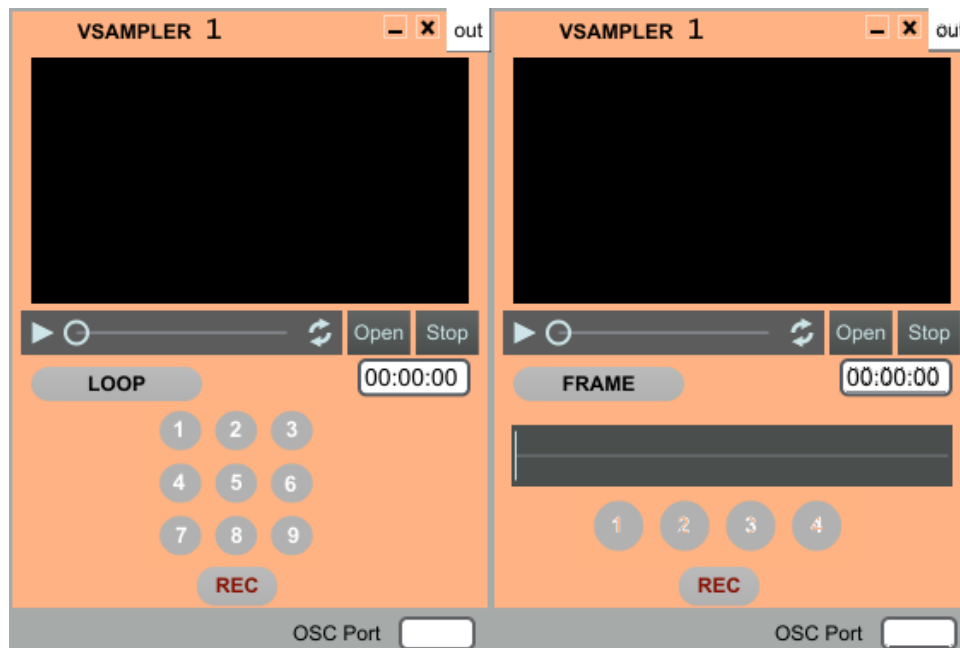


Il·lustració 57: Secció de la funció selectora del mòdul VSwitcher

4.3.3 VSampler

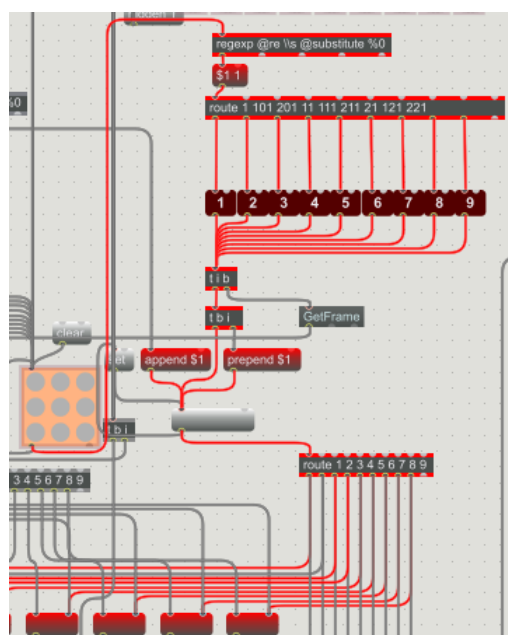
Aquest mòdul té dos modes principals que són el mode *frame* i el mode *loop*, com es mostra a la Il·lustració 58. VSampler és el mòdul que més repte ens suposa ja que per a que es puguin guardar i carregar posicions exactes i *Loops* del vídeo haurem de tindre en compte algunes coses:

Necessitarem quatre [matrixctrl], que és una espècie d'array de *buttons* que utilitzarem com a botons de memòria. Necessitem un per al mode gravació en *frames*, un per al mode reproducció en *frames*, un per al mode gravació en *loop* i un últim per al mode reproducció en *loop*.

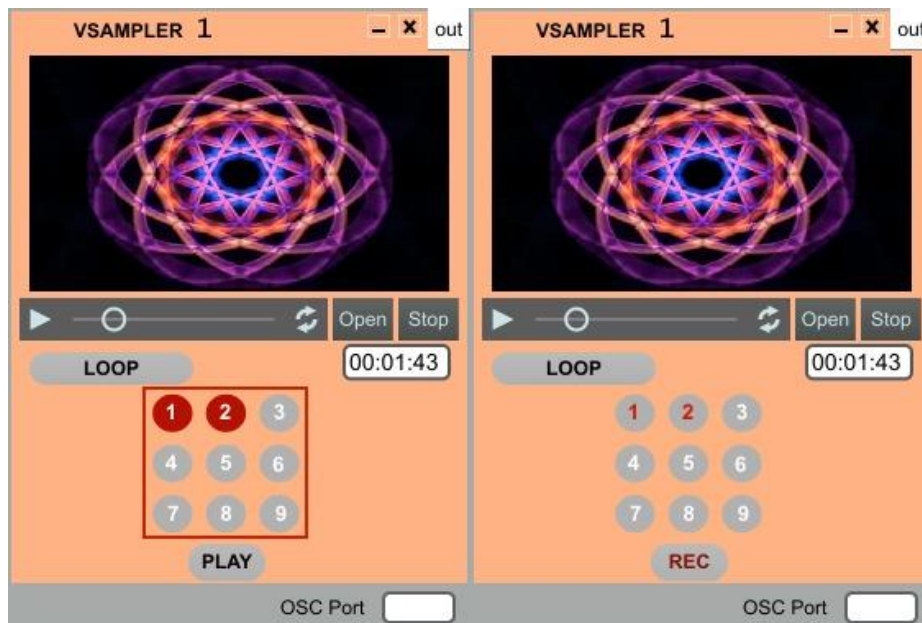


II·lustració 58 : VSampler en mode frame (esquerra) i mode loop (dreta)

El funcionament dels [matrixctrl] de gravació està a la II·lustració 59, i es pot veure el recorregut de la informació als cables i objectes de color roig. A l'apretar en un dels seus botons, l'objecte trau per la seua eixida la posició del botó i un número d'estat. Aquesta informació passa per un [regex] que elimina els espais entre els números per a que després aquest número commute amb l'objecte [route] segons el botó que s'haja polsat. A l'elegir el botó polsat es demana a l'objecte [jit.movie] el número de *frame* actual, i aquest número es guarda a la caixa de missatge corresponent per a estar llest per a quan es vulga reproduir. Per a l'apartat de gravació de *loop* es seguirà el mateix procediment però amb dos números, el d'inici i el de fi del *loop*.

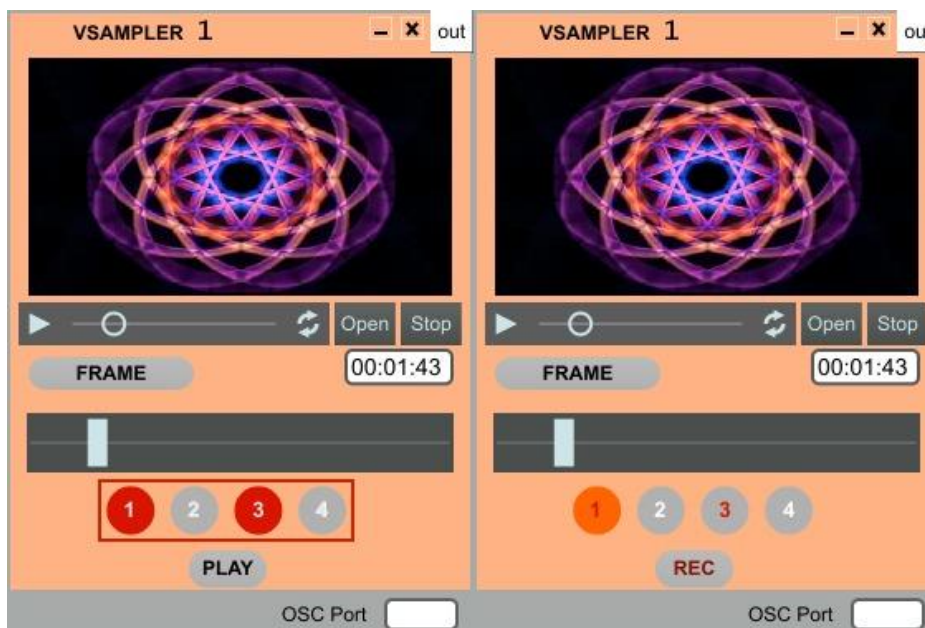


II·lustració 59: Secció de gravació de frames de VSampler



Il·lustració 61: VSampler frame en mode rec (esquerra) i play (dreta)

A la Il·lustració 62 veiem el mateix però en el mode *frame*. En aquest cas hem decidit guardar informació als botons 1 i 3. Es veu el fons roig al mode *rec* i els números rojos al mode *play*. L'últim número marcat es veu pel fons taronja i a més es pot veure en el [slider] el tram de vídeo que hem convertit en *loop*.



Il·lustració 62: VSampler loop en mode rec (esquerra) i play (dreta)

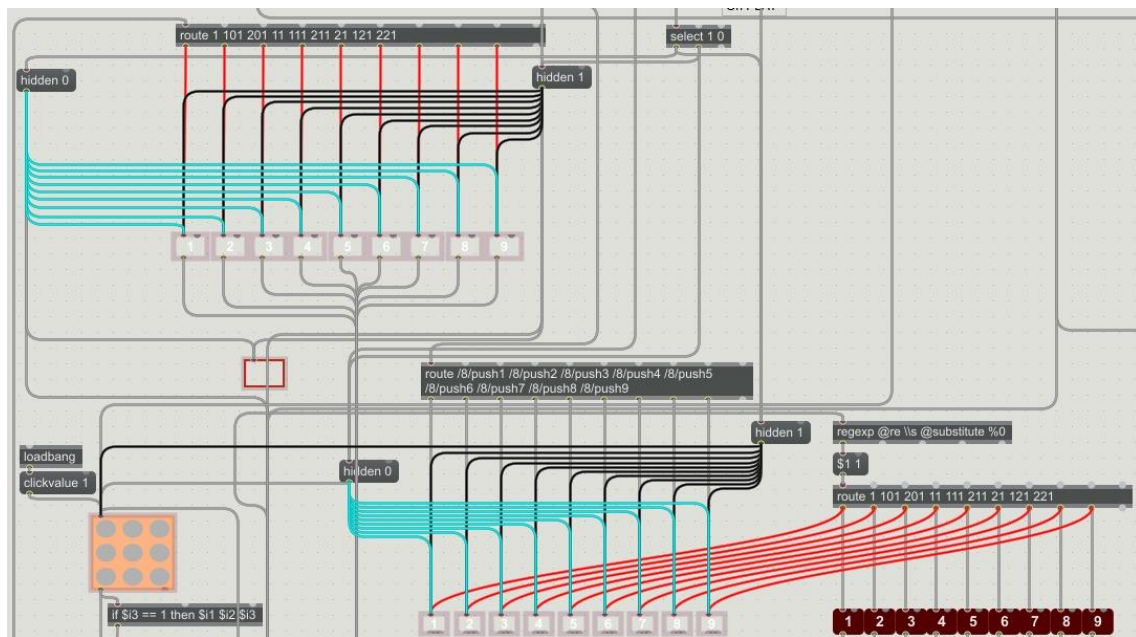
A la següent Il·lustració 63 podem veure com està dissenyat el mòdul per poder fer tota aquesta funció visual. Començem reconeguent els nostres elements: dels missatges *hidden 0* ixen cables blaus, dels missatges *hidden 1* n'ixen de negres, i dels objectes [route] ixen cables rojos. La fila de números superior són objectes [message] i la fila de números inferior són objectes [textbutton] en mode *toggle*.

El procediment que es segueix es el següent:

Quan al [matrixctrl] de rec s'apreta un dels botons, aquesta informació va al [route] inferior el qual va al [textbutton] corresponent i li envia un 1. Açò fa que el número passe de blanc a roig, propietat que hem programat a l'apartat de informació de l'objecte. A part també va als botons que tenen fons roig per a guardar la frame corresponent a una caixa de missatge.

Quan al [matrixctrl] de play s'apreta un dels botons, aquesta informació va al [route] superior el qual envia al [missatge] corresponent un bang. Açò farà que s'envie un bang al missatge on està guardada el número de frame abans guardada.

Els missatges de *hidden* s'activen o desactiven segons apremem en el [textbutton] la opció de rec o play. Amb aquests tres processos s'aconsegueix l'efecte visual que volem per a VSampler.

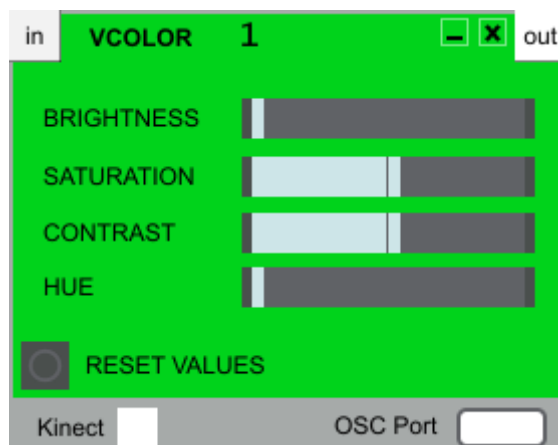


Il·lustració 63: Disseny de VSampler per a la funció visual

Aquest és el funcionament bàsic del mòdul. A part amb els telèfons mòbils i les tauletes gràfiques també es poden controlar els botons de reproducció, amb un disseny paregut a l'explicat al mòdul de VSwitcher. Finalment te una última secció important que compta amb un sistema de guardat per a les nou dades de *frames* i les huit de *loops* per tal de que es pugui carregar i guardar des del mòdul Speakers.

4.3.4 VColor

Aquest últim mòdul tindrà la funcionalitat de canviar les propietats de la imatge tals com la lluminositat, el contrast la saturació o el to de color. El mode de presentació del mòdul es pot veure a la Il·lustració 64, es pot elegir canviar aquests quatre paràmetres i a més tenim la opció de reestablir els valors per defecte, i d'utilitzar la Kinect i els dispositius mòbils.

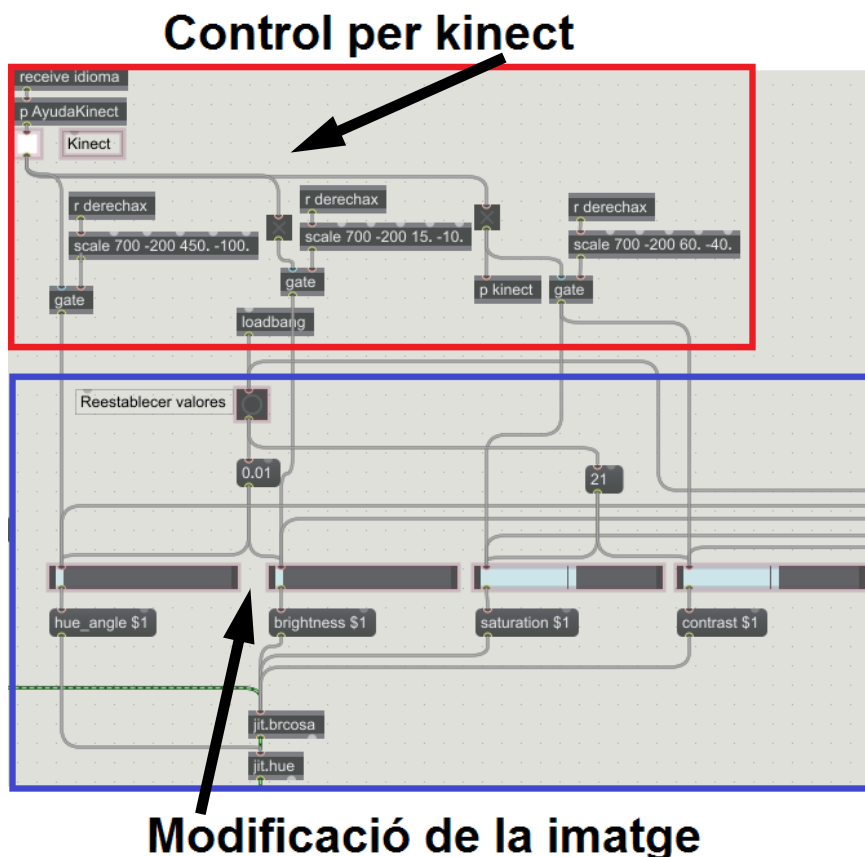


Il·lustració 64: Finestra de presentació del mòdul VColor

El funcionament d'aquest mòdul es pot dividir en la part del control per Kinect requadrat en roig a la Il·lustració 65 i la part de la modificació de la imatge requadrat en blau a la Il·lustració 65.

Al control per Kinect la part de control es troba a l'objecte [patcher]. A l'activar-se s'enviarà pel port 12345 la informació de la posició x i y de les mans dreta i esquerra que per defecte anirà entre -200 i 700. Haurem d'escalar els valors segons el rang dels nostres quatre *sliders*, ho farem amb l'objecte [scale].

Per a modificar la imatge utilitzarem l'objecte *brightness/contrast/saturation* [brcosat] i el [hue]. Els paràmetres els modificarem amb aquests quatre *sliders* que es podran controlar bé amb l'ordinador, amb dispositius mòbils o amb Kinect. A l'apretar el botó de Restablecer valores enviarà als *sliders* el número corresponent que torna la imatge als seus valors originals.



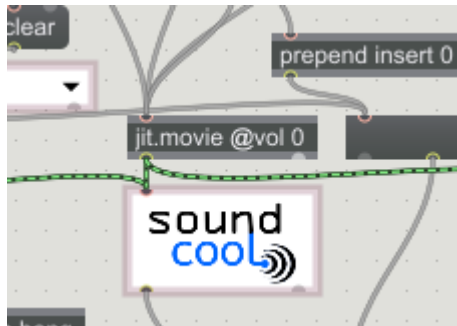
Il·lustració 65: Secció del mòdul VColor

Capítol 5. Correccions, resultats i pressupost

En este capítol tractarem les modificacions que hem fet després d'haver fet la primera versió dels mòduls. Primer parlarem del *feedback* que ens ha donat la gent de l'equip de *Soundcool* i de persones sense cap relació amb el projecte, després dels resultats finals que tenim per ara i finalment del pressupost del projecte.

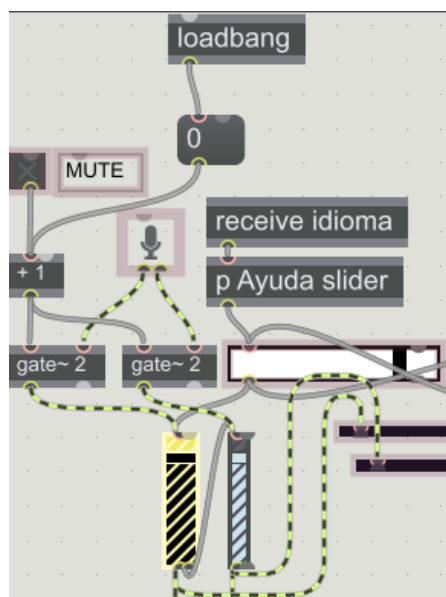
5.1 Test i correccions dels mòduls

Començarem per les correccions més senzilles com la dels mòduls de vídeo ja existents VPlayer i VDirectInput. Vam proposar una solució per a llevar la veu del contingut, però ens ho van corregir amb una altra més fàcil, com es pot veure a la Il·lustració 66. En lloc de utilitzar l'objecte [jit.movie~] i utilitzar un altre objecte per llevar el so, el que farem serà utilitzar [jit.movie] i al mateix objecte posar l'argument @vol 0 per a que faja la mateixa funció de mutejar.



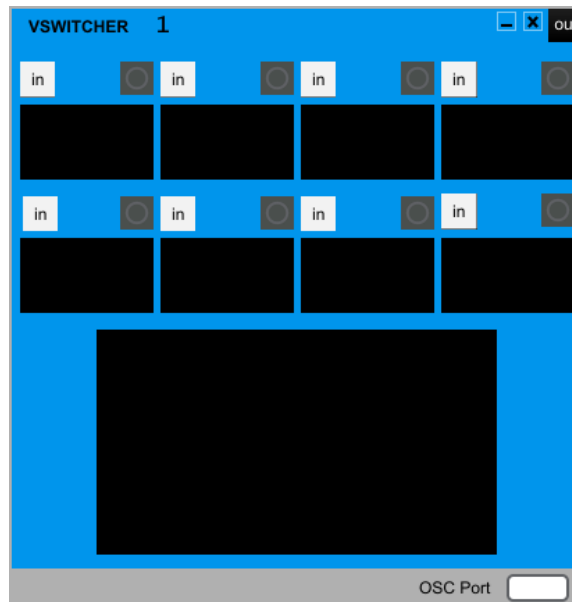
Il·lustració 66: Correcció del so als vídeos

El mòdul corregit de Direct Input no funcionava ja que al tancar el so del micròfon amb la solució proposada el que passava era que tot el so general de tots els mòduls es desactivava. Per a resoldre aquest problema es va afegir els objectes [gate~] que feien la funció de deixar passar el fluxe d'àudio o no, segons es feia clic al botó de *mute*, com es veu a la Il·lustració 67.



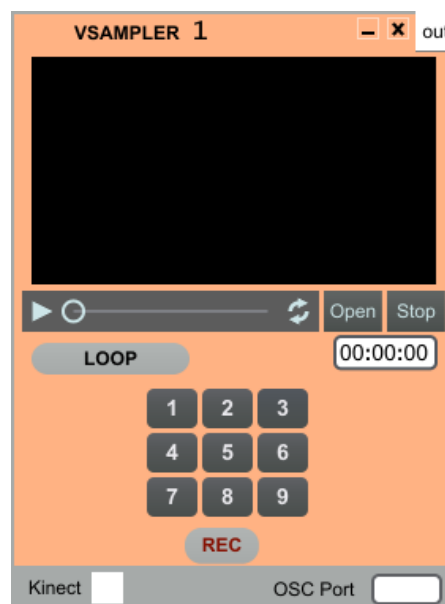
Il·lustració 67: Correcció del so a Direct Input

Respecte al mòdul de VSwitcher en un primer moment la elecció de vídeo es feia fent clic a un botó que hi havia al costat de la entrada *in* de cadascun de les huit entrades com es pot veure a la Il·lustració 68, però se'ns va proposar eliminar els botons. En la versió final el canvi d'entrada es fa fent clic en la pròpia pantalla, així doncs en lloc d'un [button] necessitarem un [panel] que el posarem a la presentació sobre cadascuna de les pantalles per a fer la selecció.



Il·lustració 68: Versió antiga del mòdul VSwitcher

L'equip de *Soundcool* veu alguns problemes al mòdul VSampler, uns els hem resolt i a altres els estem buscant solucions. Uns dels problemes resolts ha sigut els de l'objecte [matrixctrl] perquè en un primer moment hi havia només objectes [button] els quals no donaven informació sobre qui de tots tenia dades a la memòria, com es mostra a la Il·lustració 69. Aquest problema el vam solucionar introduint l'objecte [matrixctrl] i dues capes de [textbutton] amb números que passaven de color blanc a roig per a indicar que tenien informació en la seua memòria. També veuen un problema que utilitzem un [textbutton] per a la funció de guardar i un altre per a la funció de reproduir, i estem buscant les millors alternatives.



Il·lustració 69: Versió antiga del mòdul VSampler

Per a tindre feedback de gent aliena a *Soundcool* ens van proposar anar al festival Kikk [25] a presentar el prototip de la nova versió de *Soundcool*. El Kikk és un festival internacional de cultures digitals i creatives, i està enfocat en la implicació de les noves tecnologies en l'àmbit artístic i econòmic. Al Kikk hi acudeix gent de tot arreu del món des de dissenyadors, científics, artistes, arquitectes, desenvolupadors i músics.

Nosaltres estàvem a un stand com es pot vore a la II·lustració 70 ensenyant a altres investigadors, professors i artistes el projecte *Soundcool*. Comptàvem amb tot el material necessari per a fer una xicoteta demo dirigida al públic més curiós. En ella podien parlar per un micròfon i canviar paràmetres de la seua veu com el to, el retràs, o aplicar-hi filtres entre altres.

A part també vam explicar els mòduls que estàvem desenvolupant en eixe moment a la gent que venia a visitar la exposició i també a companys que estaven a stands com nosaltres per a tindre un feedback de persones no familiaritzades amb el funcionament de *Soundcool*.



II·lustració 70: Mostra dels mòduls al festival Kikk de Bèlgica

5.2 Resultats

Encara que queden algunes coses per millorar i corregir, per ara els resultats de tots els feedbacks que hem rebut són bons. Presentarem en un primer lloc les opinions de gent familiaritzada amb *Soundcool* i després la opinió de gent aliena a aquest.

5.2.1 Gent familiaritzada amb *Soundcool*

- Des de la resta de l'equip de *Soundcool* la opinió més general és que els mòduls de VScreen i VSwitcher estan llestos per a traure en la nova versió de *Soundcool* sense fer mes canvis.
- Sobre el mòdul VColor s'ha comentat un possible canvi de la funció de la Kinect. Ara està configurat per a que només amb una mà es canvien els quatre paràmetres per igual i aquesta

funcionalitat te poques combinacions i no hi hauria possibilitat de canviar els paràmetres independentment els uns dels altres.

- Sobre el mòdul VSampler, com hem dit abans, pareix ser un problema el fet que utilitzem un [textbutton] per a la funció de guardar i un altre per a la funció de reproduir i no només un per a les dues funcions.
- A part, també es proposen canvis als colors que hem elegit per al fons dels quatre mòduls.

5.2.2 Gent externa a *Soundcool*

- Les persones que han utilitzat *Soundcool* per primera vegada veuen intuïtiu i senzill el seu funcionament per normes generals.
- Sobre el mòdul VSwitcher no han destacat cap problema al ser un mòdul senzill i amb una sola opció.
- Al mòdul VSwitcher han hagut de preguntar com elegir diferents entrades perquè no queda molt clar en un primer moment.
- Al mòdul VSampler queda un poc confús el fet que si están a la pantalla de *Rec* el ròtul que es veja siga *PLAY* i quan están a la pantalla de *Play* es veja *REC*. Passa igual amb la pantalla de *LOOP* i *FRAMES*.
- Sobre el mòdul VColor no han destacat tampoc cap problema, és fàcil, senzill i intuïtiu.

5.3 Pressupost

Per a aquest apartat necessitarem saber el preu tant dels aparells que hem utilitzat per a posar en funcionament *Soundcool* tals com un ordinador, un *router*, telèfons mòbils, tabletas gràfiques, Kinect, micròfons com del sou d'un enginyer per a tots aquests mesos de treball. Cal destacar que per a provar que el nostre sistema funciona tant en iOS com en Windows i Android necessitarem tindre aparells amb tots aquests sistemes operatius.

Per a calcular el sou d'un enginyer de Telecomunicacions hem consultat els informes del col·legi oficial de enginyers de Telecomunicació [26]. De tots els informes el més actualitzat tracta de l'any 2013 així que ens podem fer una lleugera idea. El sou mitjà d'un enginyer de Telecomunicacions és de 49206€ encara que com es disposa de la informació per sexes, per al nostre cas el sou mitjà seria de 36768€ que per a 10 mesos treballats seria un total de 30640€. Si ens fixem en el sou anual guanyat per a treballadors amb cinc o menys anys d'experiència, com és també el nostre cas, serien uns 25731€ a l'any, que per a 10 mesos treballats seria un total de 21442€

	Preu unitari (€)	Vida útil	Preu repercutit amb amortització (€)
Tauleta Android	180	4	37,4
Tauleta iOS	600	4	124,5
Mòbil Android	150	3	41,5
Mòbil Ios	640	3	177,1
Ordinador Windows	1000	6	138,3
Ordinador Mac	1500	6	207,5
Router	50	4	10,4
Micròfon	100	5	16,6
Peu de micròfon	30	8	3,1
Cable Canon	10	5	1,7
Tarjeta de so	80	6	11,1
Total	4340		769,1

Taula 2: Pressupost unitari i repercutit del nostre treball

A la Taula 2 tenim la informació del preu unitari de tots els aparells que hem necessitat i també els anys estimats de vida útil d'aquests aparells. Aquesta informació l'hem emprada per a calcular el preu repercutit amb amortització, és a dir, el preu total que hem amortitzat de cada aparell tenint en compte el seu preu unitari, la seua vida útil, i els 10 mesos que hem passat treballant amb aquests.

El pressupost total d'aparells que hem necessitat per a dur a terme aquest treball és de 4340€, però si ens fixem en el seu preu amortitzat baixa a 769€ en total.

Capítol 6. Conclusions i treball futur

Per a aquest capítol final explicarem les conclusions que hem pogut traure de tot aquest treball. Parlarem sobre el *feedback* que hem rebut i quines serien les millors formes de millorar en un futur els mòduls presentats a aquest treball. També parlarem sobre possibles mòduls nous i com enfocar-los per a tindre una equivalència amb els mòduls de so ja creats

6.1 Conclusions

Una vegada realitzat tot aquest treball podem veure en perspectiva l'impacte que poden arribar a tenir tots els mòduls creats tant a l'àmbit professional com a l'àmbit pedagògic. La creació d'aquests quatre mòduls donen lloc a un ampli ventall de projectes artístics on l'usuari final serà qui marcarà els límits de les seues possibilitats.

- En primer lloc aquesta és una potencial ferramenta per a tota classe de creació col·laborativa audiovisual. Amb aquests mòduls ja es podria donar forma a un projecte visual que per donar recolzament a qualsevol espectacle augmentant així les seues possibilitats artístiques.
- I no només a espectacles sinó a les aules, donant suport així a la vessant pedagògica de *Soundcool*. Amb aquests mòduls es podria donar suport a les classes de música o també es podria començar a introduir a les classes de plàstica. Fins i tot ambdós matèries podrien treballar en un mateix projecte conjuntament.
- Aquest treball és un bon punt de partida per a fer una comunitat d'artistes, músics professionals i professors que suggerisquen millores i noves funcionalitats.
- Aquest projecte no és un projecte tancat, sinó que anirà evolucionant segons les necessitats dels usuaris i les propostes del grup de *Soundcool*.

6.2 Treball futur

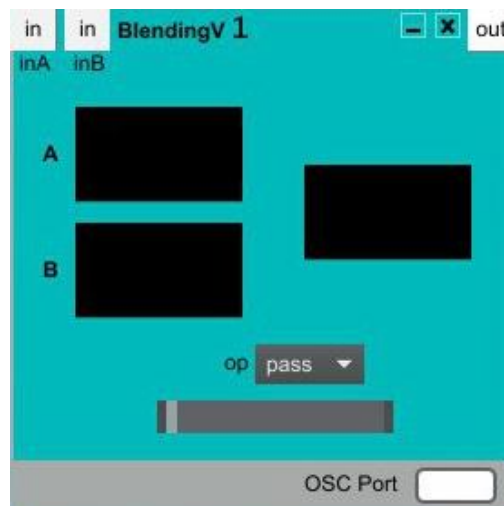
Com a últim punt per tractar parlarem de les línies de futur que te tot aquest projecte. No només haurem de millorar els mòduls ja fets, sinó que en proposarem de nous i millorarem també mòduls que estan desenvolupant-se a la resta de l'equip. En resum, per a les futures versions caldria:

- Traure un Launcher on es poguera elegir fàcilment els mòduls a obrir. El prototip que tenim per ara es el que es veu a la Il·lustració 71 on tenim per una banda els mòduls de so i per l'altra els de vídeo. Baix a la dreta de cada bloc tenim les fonts d'entrada com els Direct Input, i també els reproductors com el Speakers i el VScreen.
- Millorar els mòduls proposats en este treball. A VColor tindrem un possible canvi de la funció de la Kinect per a poder controlar independentment els quatre *sliders* que controlen els canvis de color de la imatge. A VSampler haurem de canviar la posició dels elements per tal que la manera de passar del mode *loop* a frame i de passar del mode *Play* a *Rec* siga d'alguna forma més intuïtiva i senzilla. Respecte als mòduls en general faltaria canviar el color de fons al mode de presentació.



Il·lustració 71: Launcher de Soundcool

- A part dels mòduls que estem desenvolupant, també hi ha companys com Jaime Serquera Peyró i Laura Calabuig Benítez, que per la seua part estan desenvolupant-ne de nous. D'una banda s'està dissenyant un mòdul anomenat Video Painter la funció del qual és dibuixar sobre una imatge o vídeo que es carrega al propi mòdul. Un altre mòdul que s'està creant te com a funció el crear imatges a partir d'un so que rep a la seua entrada. També hi haurà un mòdul anomenat Video Blender que es podria definir com una espècie de *mixer* entre dos vídeos. Aquest mòdul serà capaç de fer diverses operacions entre les imatges com per exemple sumar, quedar-se amb el màxim o mínim, etc. La finestra de presentació la podem veure a la Il·lustració 72



Il·lustració 72: Finestra de presentació del prototip del mòdul VBlender

En resum, aquest projecte té moltes línies de futur per a millorar i augmentar el número de mòduls. Caldrà millorar els mòduls presentats a aquest treball en els aspectes que hem assenyalat, i també caldrà avançar amb la creació de nous mòduls similars als de la part de so. Amb tot açò serà possible una usabilitat millor i més intuïtiva cap a l'usuari i tindrem com a resultat unes creacions col·laboratives millors i amb més possibilitats artístiques.

BIBLIOGRAFIA

- [1] «*Soundcool*,» [En línea]. Available: <http://Soundcool.org/es/>.
- [2] M. B. Mezquita, Aplicaciones tecnológicas para el aprendizaje musical, el desarrollo de la creatividad individual y aplicaciones musicoterapéuticas en usuarios con diversidad funcional. Proyecto *Soundcool*, València: Tesis Doctoral, 2017.
- [3] «Diari la veu,» [En línea]. Available: <https://www.diarilaveu.com/noticia/18923/%20Soundcool-tecnologia-educativa-aclamada-internacionalment-desapercebuda-a-valencia>.
- [4] «Audiotalaia,» [En línea]. Available: <http://www.audiotalaia.net/2014/02/offhz013-Soundcool-alumnos-del-ies.html>.
- [5] «YouTube,» [En línea]. Available: <https://www.youtube.com/watch?v=lzQ7GJbJiL8>.
- [6] «Synapsekinect,» [En línea]. Available: <http://synapsekinect.tumblr.com/post/6307752257/maxmspjitteer>.
- [7] «Cibersociedad,» [En línea]. Available: <http://www.cibersociedad.net/congres2009/es/coms/creacion-colaborativa-y-autoproduccion-audiovisual-iniciativas-emergentes/389/>.
- [8] D. Casacubieta, Creación colectiva. En Internet el creador es el público., Barcelona: Gedisa, 2003.
- [9] A. Roig, Industrias audiovisuales y nuevos medios, Barcelona: Publicaciones y ediciones de la Universitat de Barcelona, 2008.
- [10] E. Raymond, The Cathedral and the Bazaar. Musings on Linux and Open Source by an Accidental Revolutionary, EUA: O'Reilly Media, 1999.
- [11] R. Stallman, Software libre para una sociedad libre, Madrid: Traficantes de Sueños, 2004.
- [12] P. Himaen, La ética hacker y el espíritu del informacionalismo, Barcelona: Destino, 2001.
- [13] L. Lessig, Free Culture. How Big Media Uses Technology and the Law to Lock Down Culture and Creativity, New York: Penguin Books, 2004.
- [14] A. Muntadas, 3e Biennale d'Art Contemporain de Lyon (Catálogo), Paris: Réunion des Musées Nationaux., 1995.
- [15] «The File Room,» [En línea]. Available: <http://www.thefileroom.org/>.
- [16] «YouTube,» [En línea]. Available: <https://youtube.googleblog.com/2010/07/life-in-day.html>.
- [17] «Global.Oup,» [En línea]. Available: <https://global.oup.com/ushe/product/maxmspjitteer-for-music-9780199777679?cc=es&lang=en&>.
- [18] «Cycling74,» [En línea]. Available: <https://cycling74.com/products/max/>.
- [19] «Tangatamanu,» [En línea]. Available: <http://www.tangatamanu.com/mevic/jit.html>.

- [20] «Open Sound Control,» [En línea]. Available: <http://opensoundcontrol.org/introduction-osc>.
- [21] «Unity,» [En línea]. Available: <https://unity3d.com/es>.
- [22] «Kadenze,» [En línea]. Available: <https://www.kadenze.com/users/matt-wright>.
- [23] «Kadenze,» [En línea]. Available: <https://www.kadenze.com/courses/programming-max-structuring-interactive-software-for-digital-arts-i>.
- [24] J. E. S. Comes, Nuevas Tecnologías e Interfaces para la Educación Musical, València: Trabajo de Fin de Carrera, 2013.
- [25] «Kikk,» [En línea]. Available: <https://www.kikk.be/2017/>.
- [26] «Coit,» [En línea]. Available: <https://www.coit.es/informes/el-ingeniero-de-telecomunicacion-perfil-socio-profesional/acceso-al-informe>.
- [27] «UPV,» [En línea]. Available: <http://www.upv.es/>.
- [28] «MEVIC,» [En línea]. Available: <http://musicaelectronica.blogs.upv.es/category/Noticias/>.
- [29] «MUM,» [En línea]. Available: <https://www.upv.es/titulaciones/MUM/>.