



**Escola Politècnica Superior
de Castelldefels**

UNIVERSITAT POLITÈCNICA DE CATALUNYA

TREBALL DE FI DE CARRERA

TÍTOL DEL TFC: Vídeo interactiu amb Processing

TITULACIÓ: Enginyeria Tècnica de Telecomunicació, especialitat Telemàtica

AUTOR: Marc Gálvez Font

DIRECTOR: Eulàlia Barriere Figueroa

DATA: 23 de octubre de 2008

Títol: Video interactiu amb Processing

Autor: Marc Gálvez Font

Director: Eulàlia Barriere Figueroa

Data: 23 d'octubre de 2008

Resum

Aquest document conté la memòria del projecte de fi de carrera "Vídeo interactiu amb Processing". Processing és un programa de codi obert creat per Ben Fry i Casey Reas amb l'objectiu d'ajuntar art i programació. Amb aquest programa es poden crear dibuixos i animacions de forma fàcil i ràpida. A més, permet crear interaccions dels gràfics amb un ratolí, un teclat, àudio o comandaments sense fils com el Wiimote.

Els objectius d'aquest projecte són la connexió del comandament de la consola Wii (Wiimote) amb el PC i amb el programa Processing, la interacció d'aquest comandament amb gràfics i la creació d'una peça final on es pogués apreciar aquesta interacció. Aquesta memòria està dividida en 3 capítols: Processing, Wiimote i Sketch i àudio. En el primer capítol es parla de Processing, programa amb el qual es creen els gràfics i que ens permet crear la interacció del Wiimote amb dibuixos i animacions. En el segon capítol s'expliquen tots els passos a fer per connectar el Wiimote amb el PC i amb Processing, com està dissenyat el Wiimote i les llibreries pròpies del comandament. El tercer i últim capítol, explica totes les proves fetes amb gràfics, amb el Wiimote i amb àudio i també la peça final. A més, també s'explica la llibreria Minim, la qual ens ha permès interaccionar amb àudio.

La part del projecte que ha donat més problemes ha estat tota la part explicada en el capítol 2: el tema de la connexió del Wiimote amb el PC i amb Processing utilitzant l'entorn de Java amb la seva versió correcta i les llibreries adequades. Per aquest motiu s'ha dedicat molt de temps a tota la part de connexió del Wiimote amb el PC i amb Processing i no s'ha pogut aprofundir més en la part gràfica ni en la interacció d'àudio i gràfics. Tot i els problemes que hi ha hagut durant l'execució del projecte, finalment s'han aconseguit els objectius inicials.

Title: Interactive video with Processing

Author: Marc Gálvez Font

Director: Eulàlia Barriere Figueroa

Date: October, 23th 2008

Overview

This document contains the memory of my Final Project “Interactive video with Processing”. Processing is an open source program created by Ben Fry and Casey Reas. Its objective is to mix art and programming. With this program, you can do pictures and animations easily and fast. Moreover, the program allows the interaction between the graphics and a mouse, a keyboard, audio or wireless control devices, such as a Wiimote.

The aims of this project are the connection of the Wiimote with the PC and with the Processing software, the interaction of this with graphics and the creation of a final sketch where you can appreciate this interaction. This memory is divided on 3 chapters: Processing, Wiimote and Sketch and Audio. The first chapter defines and talks about the Processing, a program used to create graphics and that allows the interaction between the Wiimote and the drawings and animations. The second chapter contains the steps that you must follow in order to connect the Wiimote with the PC and with Processing, it also explains how the Wiimote is designed and the libraries of the controller device. The third and last chapter details, with some graphics, all the tests done with the Wiimote and the audio, and also the final sketch. In addition, this chapter includes a section dealing with the Minim library, that is the one that permitted me the interaction with the audio.

The most problematic part of the Project have been all the things explained in chapter 2: most of the problems were related with the connection of the Wiimote with the computer and Processing, using the right version of the Java environment and the appropriate libraries. For that reason, I devoted a lot of time to this part and I haven't deepened as much as I would like to with the graphics part or with the interaction of audio and graphics.

Although all the problems occurred during the execution of the project, I finally reached all the initial objectives.

ÍNDEX

INTRODUCCIÓ	1
CAPÍTOL 1. PROCESSING	2
1.1. Què és Processing?	2
1.2. Descripció	4
CAPÍTOL 2. EL WIIMOTE I LES LLIBRERIES	9
2.1. El comandament de la Wii	9
2.1.1. Introducció	9
2.1.2. Funcionament	10
2.1.3. Comunicació	11
2.2. Connexió del Wiimote amb Processing	11
2.2.1. Què es necessita	12
2.2.2. Quins passos s'han seguit	12
2.3. Descripció de les llibreries	14
2.3.1. Wrj4P5	14
2.3.2. Loc	17
2.3.3. WiiremoteJ	18
2.4. Variables i funcions que s'han utilitzat del Wiimote	19
3. CAPÍTOL 3. SKETCH I ÀUDIO	20
3.1 Proves del Wiimote	21
3.2 Proves de gràfics	23
3.3 Proves amb la llibreria Minim	25
3.4 Peça final	26
BIBLIOGRAFIA	30

INTRODUCCIÓ

El treball de fi de carrera Vídeo interactiu amb Processing té com a objectius principals la connexió d'un Wiimote amb el PC i amb Processing, la interacció del comandament amb gràfics i la creació d'una peça final on es pugui veure aquesta interacció. La peça final és la culminació de tot el projecte, representa la unió de totes les parts d'aquest amb un vídeo interactiu creat amb Processing on interaccionen àudio, gràfics i el Wiimote. El projecte està dividit en 3 capítols.

El primer capítol tracta del programa Processing. Aquest capítol explica què és aquest programa, a qui va dirigit, com funciona, l'entorn de desenvolupament i el sistema de carpetes. També explica l'estructura bàsica d'un sketch i les funcions bàsiques pròpies dividides per tipus.

El segon capítol fa referència al comandament de la Wii. Explica com està format, tant l'exterior com l'interior; la comunicació amb el PC i el funcionament intern. També explica quins són els passos a seguir per connectar-lo amb el PC i amb Processing i les llibreries necessàries per aconseguir-ho.

El tercer capítol parla de les proves fetes amb Processing: gràfiques, àudio i Wiimote. També inclou una explicació de la llibreria d'àudio Minim, que és la que s'ha utilitzat per a la peça final i les funcions bàsiques d'aquesta llibreria. Aquest capítol acaba amb una explicació de la peça final. S'explica com funciona i com es poden dividir les interaccions. A l'última part d'aquest capítol hi ha un esquema molt bàsic del codi d'aquesta peça.

En un principi semblava que la interacció del comandament de la Wii amb Processing seria una feina més o menys senzilla, ja que tot i haver-hi poques coses fetes, hi havia programes, com per exemple el GlovePie, que permetien connectar el Wiimote amb el PC de forma gairebé automàtica. El problema era que per connectar el Wiimote amb Processing, no només s'havia de connectar amb el PC, sinó que s'havien d'instalar unes versions determinades del software, unes versions que fossin compatibles amb les llibreries que permetien la connexió i alhora, que fossin compatibles amb el receptor de Bluetooth. Després de molt treballar-hi, buscar informació per fòrums i parlar amb altres persones que ja havien aconseguit aquesta connexió, es va aconseguir. A partir d'aquell moment, es van començar a fer proves amb gràfics, amb el Wiimote i amb àudio per arribar a aconseguir la peça final.

CAPÍTOL 1. PROCESSING

1.1. Què és Processing?

Processing és un programa dissenyat per treballar de forma molt fàcil amb imatges, moviment i amb la interacció de dispositius externs. Aquest programa va ser creat per ensenyar els fonaments de programació mitjançant un context visual. Processing és un projecte obert creat entre el 2001 i el 2002 per Ben Fry i Casey Reas com una evolució de les idees explorades per el grup d'Estètica i Computació en el MIT Media Lab de Nova York. A la dècada dels 90, les universitats formaven estudiants artístics o estudiants tecnològics, la única forma d'aconseguir un híbrid entre artista i tecnòleg era anant a estudiar al MIT on John Maeda era professor. L'objectiu del grup de John Maeda en el MIT era aplicar la tecnologia a l'art. El fet d'acostar la programació a dissenyadors i artistes va fer possible la creació de Processing. Tant Ben Fry com Casey Reas venien del món del disseny, però tots dos tenien certs coneixements tecnològics. Un cop van entrar a formar part de l'equip del MIT d'Estètica i Computació, juntament amb altres alumnes, es van posar a treballar amb un projecte de disseny amb nombres que els va permetre aprendre molt sobre què esperaven de la programació els dissenyadors i artistes. En aquell moment van començar a veure algunes semblances amb els esbossos i prototips que els dos havien fet anys abans a la universitat i d'aquí en van sortir les primeres idees per arribar al Processing.

Processing està format per un llenguatge de programació i un entorn de desenvolupament. En un principi, el programa va ser dissenyat per crear esbossos i prototips, actualment, té moltes altres aplicacions orientades a la producció. Processing s'utilitza per fer anuncis on es treballa amb elements en 3 dimensions i amb sistemes de partícules, per fer projeccions en discoteques o concerts de música electrònica, per a la visualització de dades... Hi ha una gran quantitat de programes que permeten treballar amb vídeo interactiu, però la grandesa de Processing és que es pot aconseguir crear animacions de forma molt ràpida i simple.

Processing és un conjunt de llibreries Java que permet oblidar-nos del sistema de classes de les llibreries, encara que se'n poden utilitzar i crear, i treballar directament amb les funcions pròpies d'aquestes. Mentre que en la majoria d'entorns de programació és complicat crear llibreries, amb Processing no s'ha de fer res, el programa ho fa tot automàticament. Les llibreries Java que formen Processing estan totalment orientades a generar i modificar imatges i animacions.

Les persones que s'inicien en el camp de la programació poden realitzar el seu propi programa en pocs moments degut a la simplicitat de Processing; per altra banda, els que tenen un cert nivell de programació, poden utilitzar llibreries per aconseguir sketchos més complexos. Processing facilita la generació algorítmica d'imatges i simplifica la interacció d'altres dispositius com el teclat, el ratolí, àudio o llibreries externes amb el programa.

Processing és un llenguatge orientat a objectes, per això es poden crear classes independents de l'sketch general, però relacionades amb aquest tal i com es veu al capítol 3. Una de les qualitats més importants que té Processing és permetre crear moviment de forma molt fàcil. Normalment els dibuixos es representen sense moviment, però Processing permet crear animacions i interactuar amb aquestes. Facilita molt el treball amb imatges en 2 i 3 dimensions.

Per una banda, aquest programa pot interessar a artistes que són capaços de crear amb molta facilitat, però que no estan acostumats a fer-ho a partir de codi. Processing els hi permet aprendre a programar de forma molt visual, amb molt poc aprenentatge són capaços de veure reflectits els seus avenços en la programació. Molts dels artistes que utilitzen Processing ja havien barrejat anteriorment art i tecnologia, però mai havien arribat a crear dibuixos o animacions a partir de codi picat per ells mateixos. A més, l'única forma de variar la peça creada és a partir del codi. Els artistes saben què crear ràpidament, però no saben com fer-ho. Per altra banda, pot interessar a enginyers. A aquests els hi és molt fàcil programar, el problema més gran el tenen amb la imaginació o creativitat, saben com fer-ho, però no tenen clar què fer. Els enginyers fan tota la part més tècnica: creació de llibreries, connexió de dispositius externs amb el programa i creació de tutorials o cursos d'aprenentatge. Tal i com s'acaba de dir, una de les principals raons que fa que Processing interessi als enginyers és que es pot aconseguir la interconnexió de dispositius externs (ratolins, teclats, comandaments de la Wii, joysticks) amb animacions i dibuixos. Tot el tema de comunicació, ja sigui mitjançant plaques com la Wiring, l'Arduino o la connexió d'altres dispositius amb fils, o sense fils, són raons que fan que Processing sigui interessant per als enginyers i sobretot, enginyers de telecomunicacions.

Processing està en fase de proves, és una versió beta. Un dels factors que fa evolucionar el programa ràpidament és la col·laboració de tota la gent que treballa amb aquest per fer-l'ho cada dia millor. Tant enginyers com artistes, poden contribuir a millorar el programa creant noves llibreries, exposant les seves peces i compartint-les amb altres usuaris, responent questions o creant cursos sobre Processing. Per contribuir en la millora de Processing es pot fer des de la pròpia pàgina web o mitjançant el fòrum que s'hi pot accedir des de la mateixa pàgina a l'apartat *Discourse*. La millor manera per millorar el programa és que els dos perfils de persones que l'utilitzen interactuïn entre si, treballin juntes i comparteixin coneixements.

La pàgina web oficial del programa és <http://www.processing.org> i conté tota la informació necessària sobre Processing: instal·lació per a diferents sistemes operatius, totes les llibreries necessàries, un fòrum, un lloc on penjar sketchos, totes les contribucions per part de persones anònimes. El fòrum és una de les parts més potents de la pàgina web, ja que està molt actiu degut a la gran quantitat de gent que hi entra cada dia i al feedback constant que hi ha entre usuaris.

1.2. Descripció

En aquest apartat s'explica com funciona Processing. Està format per 3 subapartats: Entorn de desenvolupament, Estructura simple d'un sketch i Funcions bàsiques

1.2.1 Entorn de desenvolupament

A continuació s'explica l'entorn de desenvolupament del programa. Primer s'explica el funcionament del programa i en el punt següent l'estructura de carpetes.

Executar: botó per executar el codi. Obre una nova finestra on es representa l'sketch.

Stop: Atura l'animació.

Arxiu nou: Crea un nou arxiu.

Obrir: Obre sketchos guardats en el PC.

Guardar: Guarda l'sketch actual.

Exportar: Exporta l'sketch com a applet.

Nova Classe: Crea una nova classe associada a l'sketch.

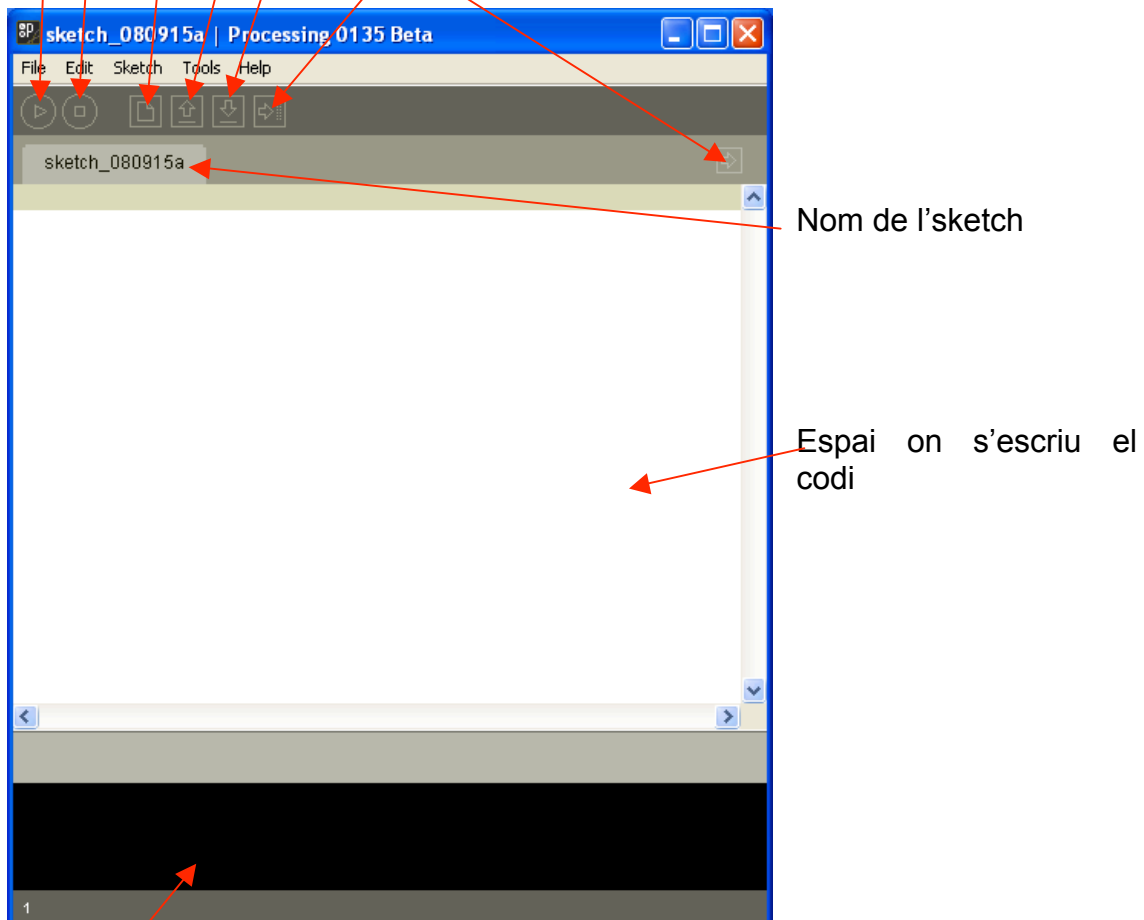


Figura 1.1 Entorn de programació.

Consola: espai on apareixen els missatges d'error i els que l'usuari vol.

En aquest apartat s'explica el sistema de carpetes de Processing. Es descarrega el programa en una carpeta comprimida de nom: "processing_versió". Quan es descomprimeix la carpeta, automàticament el programa crea una altra carpeta a dins de Mis Documentos amb el nom de Processing. Aquí només s'hi guarden els sketchos.

Un sketch es guarda com una carpeta amb el nom de l'sketch, on hi ha el fitxer en format "pde", les classes i fitxers de dades, si n'hi ha.

La carpeta descomprimida, que és la pròpia del programa, està formada per 5 carpetes, l'arxiu que executa el programa i un arxiu de text amb les últimes revisions. Les 5 carpetes són: libraries, exemples, reference, tools i lib. D'aquestes la carpeta més important és la libraries. A continuació s'explica els passos a seguir per instal·lar una llibreria externa en Processing.

- Crear una carpeta dins la ruta: processing_versió/libraries amb el nom de la llibreria externa que volem importar.
- A dins d'aquesta carpeta crear-ne una altre de nom library
- La carpeta library ha de contenir l'arxiu "nom de la llibreria".jar

Si hi ha documents i fitxers font de les classes de la llibreria es poden copiar a processing_versió/libraries/nomdelallibreria

1.2.2 Estructura d'un sketch

L'estructura d'un sketch pot ser de dos tipus diferents: en mode bàsic i en mode continuu. El mode bàsic consisteix en un seguit de comandes, una sota de l'altre, sense cap mena d'organització. El mode continuu està format per dues funcions, void setup() i void draw().

void setup()

```
{  
    Inicialitzacions de les variables, de la mida de la finestra emergent, color  
    dels objectes i dels fons en l'inici...  
}
```

void draw()

```
{  
    Bucle que s'executa 30 cops per segon. A dins d'aquesta funció és on hi  
    ha les comandes que fan canviar les variable. És la que permet crear  
    animacions de forma fàcil.  
}
```

1.2.3 Funcions i variables bàsiques

Les funcions i variables bàsiques estan dividides en 5 subapartats depenent del tipus.

1.2.3.1 Paràmetres del dibuix.

size(resolució x, resolució y)

Marca la mida de la finestra on es representarà l'sketch. El primer paràmetre és l'amplada i el segon, l'alçada. Quan es treballa en 3D, tal i com es veurà més endavant, hi ha un tercer paràmetre que pot valdre P3D o OPENGL depenent de quin sigui el motor de 3D que s'utilitzi.

background(color)

Pinta el fons del color que es vulgui.

stroke(color)

Deixa escollir el color del contorn del dibuix, funciona igual que el fill() i el background().

noStroke()

Aquesta funció no pinta les línies ni els contorns de les figures.

fill(color)

Pinta la figura o figures que vénen després d'aquesta funció.

noFill()

Aquesta funció deixa la figura transparent.

smooth()

Aquesta funció activa l'antialiasing, és a dir, permet veure els objectes de dos dimensions sense que estiguin pixelats.

Nota: Per representar el color de dins de les figures o contorn hi ha 4 formes de fer-ho. Les dues primeres formes utilitzen el mode escala de grisos, mentre que les dues últimes utilitzen el mode r,g,b. Amb un sol paràmetre (d'aquesta manera és pinta amb un color de l'escala de grisos, colors que tenen la mateixa quantitat de vermell, verd i blau). D'aquesta manera, la funció fill(255), pintaria de color blanc; fill(0), pintaria negre i fill(150), pintaria gris. Amb dos paràmetres (el primer paràmetre funciona igual que l'anterior i el segon dóna el % de transparència). Amb 3 paràmetres (el primer paràmetre dóna la quantitat de vermell, el segon, la de verd i el tercer, la de blau). I la quarta forma és amb 4 paràmetres (els tres primers funcionen igual que l'anterior i el quart paràmetre és el % de transparència).

1.2.3.2 Programació contínua

Apart de les funcions setup() i draw(), que s'expliquen a l'apartat anterior, hi ha algunes funcions que permeten controlar paràmetres de l'animació.

frameRate(int a)

Canvia la velocitat de l'animació. El paràmetre indica el nombre de vegades que s'executa el bucle draw() cada segon. Per defecte són 30.

1.2.3.3 Variables del sistema

width

height

Aquestes dues variables del sistema són l'amplada i l'alçada de la finestra emergent que s'han inicialitzat amb la funció size().

1.2.3.4 Primitives de dibuix

ellipse(centre de x, centre de y, diàmetre de x, diàmetre de y)

Dibuixa una el·lipse. Els dos primers paràmetres corresponen al centre de l'el·lipse i els dos últims, al diàmetre en X i en Y. Si es vol aconseguir un cercle, aquests dos radis han de ser iguals.

rect(punt inicial x, punt inicial y, costat x, costat y)

Dibuixa un rectangle. Els dos primers paràmetres són la posició en X i en Y del vèrtex superior esquerra del rectangle. Els altres dos paràmetres, les dimensions del rectangle.

line(inici x, inici y, final x, final y)

Dibuixa un segment. Els paràmetres són les coordenades del punt inicial i del punt final.

point(x, y)

Dibuixa un punt en les coordenades indicades pels paràmetres.

1.2.3.5 Altres funcions

k=((int)random(a,b));

La funció random crea un número aleatori. Hi ha 3 maneres de cridar a la funció random:

- random(); Crea un nombre aleatori entre 0 i 1.
- random(x); Crea un nombre aleatori entre 0 i x.
- random(x,y); Crea un nombre aleatori entre x i y.

En principi els nombres són reals, però afegint-hi "(int)" davant de la funció s'aconsegueix que sigui enter.

saveFrame("nom_del_fitxer");

Aquesta funció crea un fitxer en format "tif" o "jpg" de la imatge que hi ha aquell moment a la finestra cada cop que es crida.

pushMatrix()

Guarda la matriu que defineix l'origen de coordenades i l'orientació de l'escena. Per defecte, en un principi aquesta matriu és la matriu identitat. Sempre va acompanyada de la funció `popMatrix()`. És una manera d'aïllar el codi que hi ha entre el `pushMatrix()` i el `popMatrix()` de la resta, que utilitzarà una nova matriu definida per les funcions `translate()` i `rotate()`.

popMatrix()

Recupera la matriu guardada per l'últim `pushMatrix()`.

translate(x,y)

Aquesta funció aplica un canvi de coordenades. El punt (0,0), que per defecte és la cantonada superior esquerra, en el moment que es fa un `translate(x,y)`, el punt (0,0) passa a ser el punt (x,y).

rotate(angle)

Gira els eixos de coordenades segons l'angle indicat pel paràmetre.

CAPÍTOL 2. EL WIIMOTE I LES LLIBRERIES

2.1. El comandament de la Wii

2.1.1. Introducció

El comandament de la Wii, més conegut com “Wiimote”, és el dispositiu principal de control de la consola de Nintendo Wii. La seva funció principal és la d’interpretar els moviments reals dels jugadors d’aquesta consola mitjançant una sèrie de sensors.

La seva forma és semblant a la d’un comandament de televisió. Les seves mides són: 14,6x3,5x3,1 cm, i presenta els botons següents: Power, creu digital, A, B, 1, 2, Home, + i –, tal i com es pot veure en el dibuix:

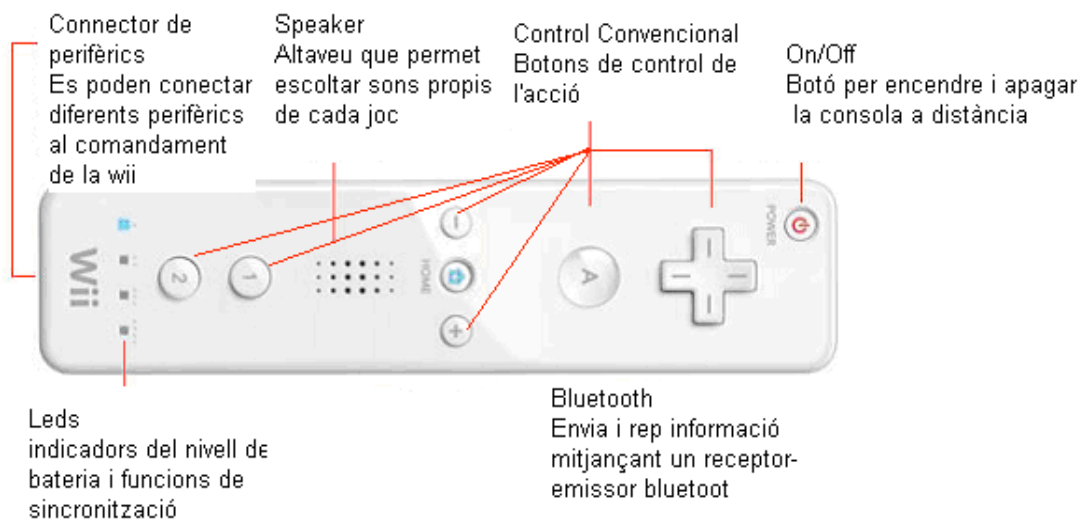


Figura 2.1 Comandament de la Wii

El Wiimote utilitza dues piles AA com a font d’energia que duren una mitjana de 60 hores fent servir només l’acceleròmetre i 30 hores utilitzant l’acceleròmetre i el punter (IR càmera). Actualment s’està investigant la possibilitat de fer un comandament amb bateries recarregables. La duració de les piles del comandament depèn de l’ús de la funció d’apuntar (IR camera), perquè als moviments s’hi ha d’afegir també tots els càlculs i l’emissió de dades del sistema òptic.

El comandament de la Wii té un xip de 16 KB EEPROM dels quals una secció de 6 KB pot ser llegida i escrita de forma lliure per l’usuari. Part d’aquesta memòria està disponible per emmagatzemar fins a 10 perfils de Mii (jugadors de Wii). El Wiimote també porta incorporat un xip Bluetooth de la marca Broadcom Technologies que permet la comunicació amb un PC que porti receptor de Bluetooth o amb la consola Wii. El comandament també està format per una IR camera (càmera d’infrarojos) que fa la funció d’apuntar. La programació de la funció de detecció de la IR càmera requereix la barra

sensora de la consola Wii o un hardware equivalent. En aquest projecte no s'han implementat aquestes funcionalitats.

2.1.2. Funcionament

El Wiimote té la capacitat de detectar l'acceleració i el sentit dels moviments al llarg dels tres eixos a través de la utilització d'un acceleròmetre. Un acceleròmetre és un instrument la funció del qual és mesurar l'acceleració, detectar i mesurar vibracions o mesurar l'acceleració deguda a la gravetat (inclinació). Aquest tipus de dispositiu es coneix com MEMS (Sistema Micro Electrico-Mecànic).

El xip MEMS (fabricat per Analog Devices) per el Wiimote és el model ADXL330 (4x4x1,45 mm) i és de molt baix consum. El seu rang de detecció és de $\pm 3,6$ gravetats i té una sensibilitat de 300 mV/g.

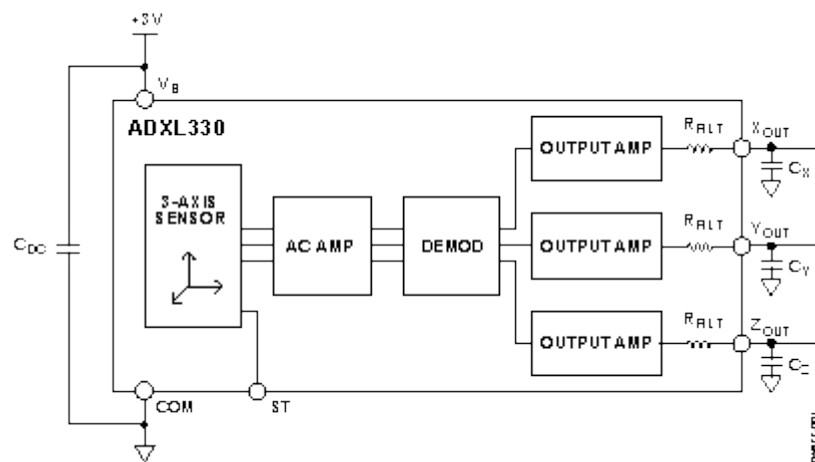


Fig 2.2. Acceleròmetre ADXL330

La part del xip que detecta els moviments és el sensor, que és una peça de silici que està subjectada a un extrem i col·locada entre el camp elèctric que generen dos capacitors. És l'encarregada de transmetre moviment. Quan es desplaça el comandament, la peça queda més aprop d'algun dels capacitors, i això provoca que el camp elèctric canviï. Quan es detecta aquest canvi es tradueix en una acció a la pantalla. En el dibuix es pot veure:

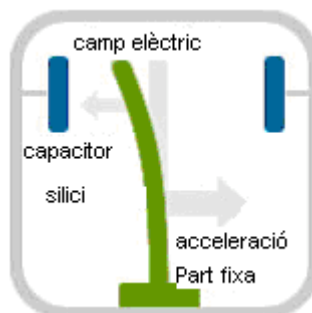


Fig2.3 moviment de la peça de silici

Aquests xips mesuren tant l'acceleració com la direcció del moviment comprovant els canvis en els electrons del seu interior. Imaginem dues plaques

minúscules dins del sensor. Una es manté fixa, però l'altra es mou. Els electrons que envolten les plaques es mouen amb aquest moviment, i mesurant la capacítancia, el sensor és capaç de capturar les dades necessàries. D'aquesta forma el dispositiu pot enviar dades de moviments relatius en tots els eixos de coordenades, a més de les dades d'acceleració, detectant girs, moviments en l'aire, inclinacions... Aquestes dades s'envien al xip Broadcom que és l'encarregat d'enviar-les a la consola o al PC.

2.1.3. Comunicació

El Wiimote es comunica amb la consola o PC via Bluetooth. La consola utilitza la llum infraroja per detectar un grup de LEDs a la barra sensora i d'aquesta manera el control adquireix ubicació i orientació per fer més reals tots els moviments (funció única de la IR càmera). D'aquesta manera el Wiimote és capaç de detectar moviments diferents en 3D que tenen resultats precisos a la pantalla. El comandament es comunica amb la consola sense fils a través de Bluetooth de curt abast en el medi ràdio, amb el qual és possible connectar fins a quatre controladors a 10 metres de la consola. Tot i això, si s'utilitza la IR càmera els comandaments han d'estar com a molt a 5 metres de la barra sensora.

Totes les dades de detecció de moviment i de la funció d'apuntar són enviades directament al xip Bluetooth de Broadcom integrat a la placa del comandament. Aquest xip, que funciona via ràdio, és l'encarregat d'enviar instantàniament el conjunt de dades a la consola o al PC on ho rep un altre xip Bluetooth. El xip que ha d'anar al PC no té perquè ser de la marca Broadcom, el que sí que ha de ser Broadcom és el software per fer funcionar aquest xip. La comunicació en els dos sentits utilitza la banda de 2.4GHz i proporciona un ample de banda per a dades de 2.1 Mbps. Broadcom ha afegit millores que permeten una latència molt baixa entre consola i Wiimote, intentant que la resposta sigui pràcticament la d'un comandament cablejat. Cal afegir que els paràmetres Bluetooth ja tenen un consum realment baix, d'aquesta manera s'aconsegueix que les piles del comandament durin més temps.

2.2. Connexió del Wiimote amb Processing

El primer pas a fer en aquest projecte ha estat connectar el comandament de la Wii amb el PC i posteriorment amb el programa Processing.

En aquest capítol s'explica com connectar el comandament de la Wii amb Processing. Les proves s'han fet amb el sistema operatiu Windows XP Service Pack 2.

2.2.1. Què es necessita

- Un Wiimote (comandament sense fils de la consola de Nintendo Wii).
- Un receptor Bluetooth per a PC (en el cas que el PC no porti incorporat aquest dispositiu). En aquest cas el dispositiu és el 15542 de la marca Trust.
- Un programa controlador de Bluetooth per a PC. En aquest cas el software és la versió 5.0.1.801 del software de Broadcom, Widcomm per a Bluetooth.
- Les llibreries adequades per a la correcta interacció del comandament amb Processing. Aquestes llibreries són: Wrj4P5, WiiremoteJ, Loc i BlueCove 2.0.2.

2.2.2. Quins passos s'han seguit

2.2.2.1. *Instal·lar el receptor Bluetooth en el PC*

Connectar un receptor de Bluetooth en el port USB del PC i instal·lar el software/driver per detectar els dispositius Bluetooth. Aquest software ha de suportar el protocol L2CAP, perquè la llibreria Wrj4P5 només suporta aquest protocol. Amb un altre protocol s'aconseguiria connectar el Wiimote amb el PC, però no es podria aconseguir que el comandament interactués amb Processing. Això limita a fer servir el controlador Widcomm, perquè és l'únic que suporta aquest protocol. Exactament la versió correcta del software controlador és la 5.0.1.801. Totes les altres versions donen problemes: són massa recents i no detecten el comandament de la Wii, o són massa antigues i donen errors amb la llibreria Wrj4P5.

2.2.2.2. *Instal·lar la versió correcta de Java.*

Si es tenen instal·lades diferents versions de Java, Processing, per defecte, treballa amb la versió 1.4 o inferior. Per fer funcionar la llibreria WiiremoteJ es necessita la versió de Java 1.5. Per això, s'han de desinstal·lar totes les versions de Java existents en el PC i instal·lar el Java 1.5, tant el JRE1.5 com el JDK1.5. D'aquesta manera s'assegura que tot funcionarà correctament. Si s'utilitza la versió de Java 1.6, la més recent, no s'obre el programa. Si s'utilitza la versió de Java 1.4 no funciona correctament amb les llibreries Wrj4P5 ni amb la WiiremoteJ. Tots aquests problemes amb el Java són deguts a que Processing actualment encara no ha arribat a la versió definitiva (1.0), sinó que és una versió Beta.

2.2.2.3. *Instal·lar Processing i les llibreries necessàries*

El primer pas a fer és descarregar i instal·lar Processing, la versió sense Java, ja que amb la versió amb Java, la llibreria WiiremoteJ no funciona correctament.

Un cop es té instal·lada aquesta versió de Processing s'han de descarregar i instal·lar les llibreries que permetran connectar el Wiimote amb Processing.

Les llibreries a descarregar són:

- Loc
- Wrj4P5
- WiiremoteJ

La informació sobre aquestes 3 llibreries es troba en el fòrum de Processing. Dins del fòrum en el tema que es parla de llibreries i eines de desenvolupament, exactament en el post que té per títol: Wrj4P5, WiiRemote P5Library.

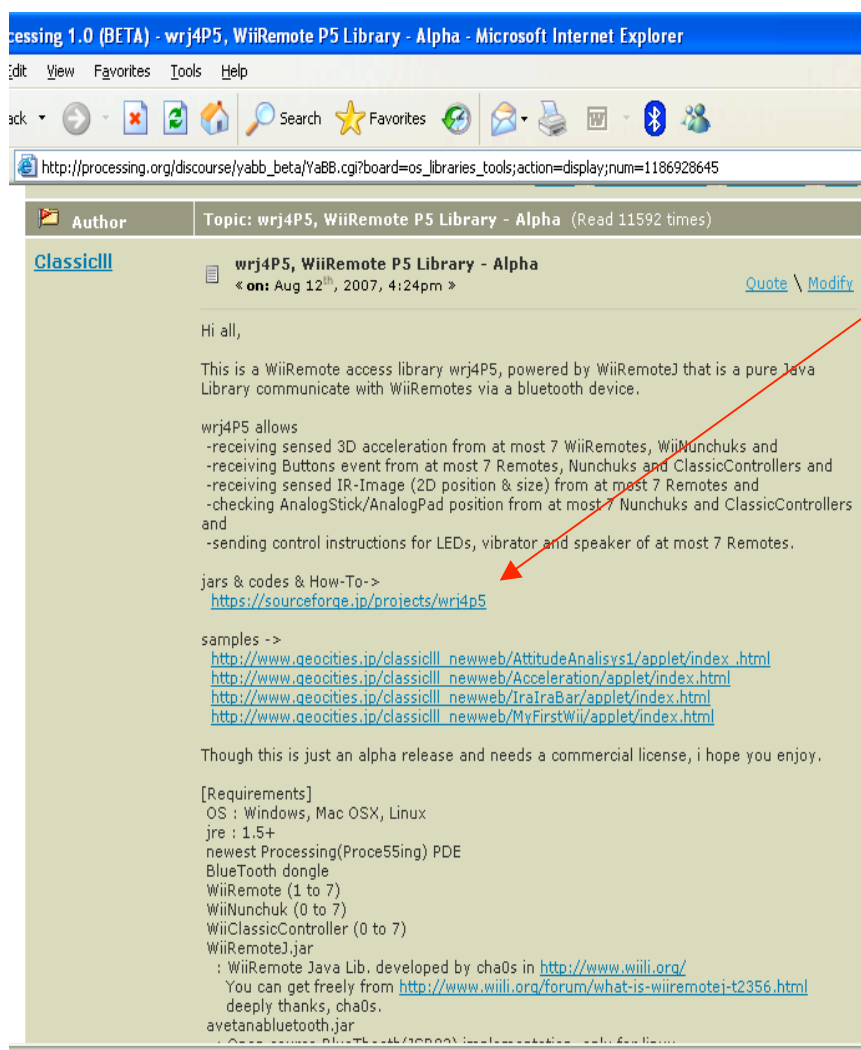


Figura 2.2 Tema wrj4p5, WiiRemote P5 Library-Alpha del fòrum oficial de Processing

La llibreria Loc està formada per l'arxiu Loc.jar (el fitxer propi de la llibreria).

La llibreria Wrj4P5 necessita la llibreria Bluecove. S'han de descarregar els arxius Wrj4P5.jar i BlueCove2.0.2.jar

La llibreria BlueCove connecta el dispositiu Bluetooth amb el controlador de Wiidcomm. S'ha d'utilitzar la versió 2.0.2 de la llibreria BlueCove. La llibreria Wrj4P5 també depèn de la WiiremoteJ.

La llibreria WiiremoteJ s'ha de descarregar a partir d'un buscador Web com per exemple Google. La versió 1.3 de WiiremoteJ és la correcta perquè tot funcioni.

2.2.2.4. *Fer funcionar el Wiimote amb el Processing.*

Primer s'ha d'obrir el programa Processing. Executar un exemple on es faci servir la llibreria Wrj4P5 o una peça pròpia. És important que aquest exemple tingui importades les llibreries Loc i Wrj4P5. Per carregar les llibreries dins del programa s'ha d'escriure a la capçalera de l'sketch: `import nomdelllibreria.*`

A l'executar l'exemple de la Wrj4P5 en Processing apareix el missatge: "BlueCove 2.0.2 on widcomm." a la consola. Seguidament el programa busca el comandament de la Wii i un cop pitjat el botó de sincronització, el programa connecta el comandament amb el Processing mitjançant les llibreries. Totes les funcions utilitzades per programar el comandament de la Wii s'expliquen a l'apartat "Descripció de les llibreries" d'aquest mateix capítol.

2.3. **Descripció de les llibreries**

En aquest apartat es descriuen les llibreries que han permès interconnectar el comandament de la Wii amb Processing: la Wrj4P5, la Loc i la WiiremoteJ.

2.3.1. **Wrj4P5**

Wrj4P5 és la llibreria principal, la que controla totes les funcions del Wiimote i altres dispositius externs que es poden connectar a la Wii. Aquesta llibreria ha estat creada per ClassicIII, que mitjançant el fòrum de la pàgina de Processing resol qüestions de forma bastant ràpida a altres usuaris que volen connectar el Wiimote amb Processing. Aquesta llibreria està formada per 17 classes, però per a la interconnexió del Wiimote amb Processing només se n'utilitzen 3: *RimokonEvent*, *WiiRimokon* i *Wrj4P5*. En el projecte només s'utilitzen algunes funcions del Wiimote, que són les que s'expliquen aquí.

2.3.1.1. *Wrj4P5*

La classe Wrj4P5 és la bàsica per manejar el Wiiremote i les extensions d'aquest. Conté totes les funcions necessàries per extreure informació d'aquests dispositius i comunicar-los amb l'sketch de Processing.

Atributs

public static WiiRimokon rimokon;

Objecte del tipus WiiRimokon que, tal i com es pot veure més endavant, és una altra classe de la llibreria Wrj4P5.

private static WiiRimokon[] rims = new WiiRimokon[7];

Vector de 7 objectes WiiRimokon, que és el nombre màxim de comandaments que es poden connectar amb el PC.

Funcions

public Wrj4P5 connect()

Aquesta funció connecta el Wiimote.

public void wiiRemoteDiscovered(WiiRemoteDiscoveredEvent evt)

Aquesta funció busca els Wiimotes que hi ha a suficient distància i activats per a ser connectats i els connecta amb el PC. Imprimeix per consola el número del Wiimote connectat: si només hi ha un comandament a la consola hi surt el missatge següent: "Discover called #0".

public void findFinished(int numberFound)

Si troba un Wiimote o més, iguala la variable rCount al nombre de Wiimotes connectats.

public WiiRimokon rimokon(int rid)

Aquesta funció retorna el nombre de Wiimotes connectats o null, si no n'hi ha cap de connectat.

public void disconnected(int rid)

Aquesta funció detecta quan es desconecta algun Wiimote per error i imprimeix un error per consola.

2.3.1.2. *WiiRimokon*

La classe *WiiRimokon* implementa el *WiiRemoteListener* (detecta els senyals del *WiiRemote*) i actua com un *WiiRemoteController* (controla els senyals del *WiiRemote*). Aquesta classe conté totes les funcions que controlen el comandament de la Wii. És la classe pròpia del Wiimote.

Atributs

public long tNow = 0;

Variable tNow de tipus long, inicialment igual a 0. Aquesta variable s'utilitza per a funcions d'aquesta classe.

public Loc sensed = new Loc();

Variable sensed de tipus Loc, més endavant s'expliquen els objectes tipus Loc.

public Loc acc = new Loc();
Variable acc de tipus Loc.

Les variables sensed i acc són les variables a les quals s'assigna la sensibilitat i l'acceleració del Wiimote respectivament.

Funcions

public void disconnected()
Aquesta funció desconnecta el Wiimote.

public void accelerationInputReceived(WRAccelerationEvent evt)
Aquesta és una de les funcions més importants de la llibreria. Detecta els senyals que l'acceleròmetre del Wiimote envia a Processing, els senyals d'entrada. Detecta el moviment del comandament de la Wii.

public void buttonInputReceived(WRButtonEvent evt)
Aquesta funció detecta tots els senyals que els botons del Wiimote envien a Processing. Cada cop que s'apreta un botó del comandament, aquest envia un senyal que la llibreria interpreta mitjançant aquesta funció i això permet que mitjançant Processing es pugui treballar amb aquests senyals.

public void disconnect()
Desconnecta el Wiimote.

private boolean isJustPressed(WRButtonEvent evt)
Aquesta funció retorna el botó que ha estat premut. Les variables que representen els botons són booleans, valen 1 si estan premuts i 0 en cas contrari.

private boolean isJustReleased(WRButtonEvent evt)
Retorna el botó que s'ha deixat anar, fa la funció contrària que el mètode anterior.

2.3.1.3. *RimokonEvent*

La classe *RimokonEvent* té com a funció principal simplificar el nom dels botons i comprova l'estat d'aquests en cada instant.

Atributs

public final static int ONE = WRButtonEvent.ONE;
public final static int TWO = WRButtonEvent.TWO;
public final static int B = WRButtonEvent.B;
public final static int A = WRButtonEvent.A;
public final static int MINUS = WRButtonEvent.MINUS;
public final static int HOME = WRButtonEvent.HOME;
public final static int LEFT = WRButtonEvent.LEFT;

```
public final static int RIGHT = WRButtonEvent.RIGHT;  
public final static int DOWN = WRButtonEvent.DOWN;  
public final static int UP = WRButtonEvent.UP;  
public final static int PLUS = WRButtonEvent.PLUS;
```

Variables del tipus enter per a cada botó del comandament.

Funcions

```
public boolean isAnyPressed(int button)
```

Retorna cert si hi ha algun botó premut.

```
public boolean isOnlyPressed(int button)
```

Retorna cert si només hi ha un botó premut.

```
public boolean isPressed(int button)
```

Retorna cert si el botó està premut.

```
public boolean wasOnlyPressed(int button)
```

Retorna cert si només hi ha un botó que ha estat premut.

```
public boolean wasPressed(int button)
```

Retorna cert si el botó ha estat premut.

```
public boolean wasReleased(int button)
```

Retorna cert si el botó ha estat deixat anar.

2.3.2. Loc

La llibreria **Loc** està formada per una sola classe que té el mateix nom.

2.3.2.1. Loc

La classe *Loc* conté totes les funcions que tenen relació amb angles, distàncies, girs, rotacions i les coordenades de l'acceleració del comandament: x, y i z.

Atributs

```
public float x,y,z:
```

Variables de tipus real x, y i z. Són els 3 paràmetres d'acceleració del comandament.

Funcions

```
public Loc()
```

```
public Loc(float lx, float ly, float lz)
```

```
public Loc(Loc l)
```

```
public Loc(float lng, float ltt)
```

La funció `Loc()` està sobrecarregada, pero en totes les versions la seva funció és la d'assignar les variables `x`, `y` i `z` en funció dels paràmetres.

public Loc move(float dx, float dy, float dz)

public Loc move(Loc pos)

La funció `move()` està sobrecarregada, en les seves dues versions permet que qualsevol objecte es pugui moure per la finestra a partir de les acceleracions de `x`, `y` i `z`.

public Loc rotateZ(float pan)

Aquesta funció gira els objectes respecte l'eix `z`.

public Loc rotateY(float swing)

Aquesta funció gira els objectes respecte l'eix `y`.

public Loc rotateX(float tilt)

Aquesta funció gira els objectes respecte l'eix `x`.

public Loc mul(float fact)

public Loc mul(Loc fact)

Les funció `mul()`, també sobrecarregada, retorna un objecte `Loc` de tres variables (que són les 3 variables: `x`, `y` i `z`). Depenent de si li passen un objecte `Loc` o `float` fa un canvi de variable o no.

2.3.3. WiiremoteJ

La llibreria `WiiremoteJ` conté totes les funcions necessàries per a la connexió del `Wiimote` amb el `PC` via `Bluetooth` i d'aquest amb `Processing`. També connecta els `sketchos`, les llibreries, `Processing`, el `PC`, el comandament de la `Wii` amb el `Java`. És una llibreria de connexió. No hi ha cap funció que es faci servir directament en els `sketchos`, perquè ja es fan servir les funcions de les altres llibreries. Aquesta llibreria conté algunes funcions que no s'utilitzen ja que les llibreries `Wrj4P5` o la `Loc` ja les tenen.

2.3.3.1. WiiremoteJ

La classe `WiiremoteJ` disposa de comandes per al control de totes les característiques del `WiiRemote` i conté totes les funcions que permeten obtenir els senyals dels acceleròmetres i botons del `Wiimote`, del sensor d'infrarojos i del port d'entrada. Aquesta llibreria també permet controlar les extensions del comandament.

Quan el `Wiimote` està desconnectat (ja sigui a través d'alguna funció de desconnexió o per algun error del `Wiimote`), el comandament esborra totes les variables dels punters per alliberar memòria. Per defecte, els missatges d'error de connexió s'envien mitjançant la sortida estàndard i es veuen per la consola.

2.4. Variables i funcions que s'han utilitzat del Wiimote

Variables

wii.rimokon.senced.x;

wii.rimokon.senced.y;

wii.rimokon.senced.z;

Són les 3 variables on es guarda l'acceleració en x, y i z que envia el Wiimote i a partir de les quals es pot treballar. Aquestes variables són de tipus Loc (x, y i z). L'objecte wii és de tipus Wrj4P5, el rimokon és un dels atributs de la classe Wrj4P5, i és del tipus WiiRimokon i l'objecte senced és un dels atributs de la classe WiiRimokon, i és del tipus Loc.

Funcions

wii.connect();

La funció *wii.conect()* serveix per connectar el comandament de la Wii amb Processing.

wii.disconnect();

Aquesta funció desconnecta el Wiimote de Processing quan es tanca l'sketch.

void buttonPressed(RimokonEvent evt, int rid)

Aquesta funció iguala a 1 la variable assignada al botó que ha estat premut i a 0 als altres botons. També imprimeix per consola el botó que està premut del comandament.

CAPÍTOL 3. SKETCH I ÀUDIO

Sketch en anglès significa dibuix ràpid, el programa busca que l'usuari aconsegueixi fer dibuixos de forma ràpida, prototips i esbossos. El programa guarda l'sketch en format propi (*.pde, Processing Data Environment). El lloc on es guarda el codi que s'ha escrit, amb el codi inclòs, és l'sketch.

El projecte té 2 objectius principals: la interacció del Wiimote amb Processing i la creació d'una peça final. Aquesta peça consisteix en la interacció del Wiimote amb gràfics creats amb Processing on, a més, s'hi ha afegit àudio per fer-la més espectacular.

Hi ha 3 llibreries candidates per treballar amb música i Processing: Sonia, Ess i Minim. La llibreria escollida va ser la Minim, ja que era la més fàcil de treballar i d'entendre i, a més, permetia treballar amb mp3 sense gaires complicacions. Les altres dues llibreries no deixen treballar amb àudio en format mp3 de forma fàcil i eren més complicades. La filosofia de la llibreria Minim és integrar àudio en els sketchos de forma tan simple com sigui possible alhora que proporciona una certa flexibilitat per a usuaris més avançats. Hi ha un seguit de funcions bàsiques d'aquesta llibreria. D'aquestes funcions s'expliquen les que s'han utilitzat per fer proves i per la peça final. La llibreria Minim està formada per 64 classes diferents dividides en: 34 classes dedicades a funcions generals, 4 classes dedicades a l'anàlisi d'àudio, 8 classes dedicades a efectes i 8 classes dedicades a diferents tipus de senyals. Com que s'han utilitzat només 3 classes diferents no s'han separat els mètodes per classes.

Atributs

ÀudioPlayer song

La variable song és del tipus AudioPlayer i és on es carrega la cançó.

BeatDetect beat

La variable beat és de tipus BeatDetect. Dins de la classe BeatDetect hi ha un mètode que detecta tots els beats de l'àudio.

float eRadius

La variable eRadius és un nombre que augmenta cada cop que detecta un beat i augmenta segons l'amplitud d'aquest.

public static void start(PApplet pro)

Aquesta funció de la classe Minim inicialitza la llibreria.

public static AudioPlayer loadFile(String filename)

Aquesta funció de la classe Minim carrega l'arxiu d'àudio que es vol posar a l'sketch.

public static void stop()

Aquesta funció de la classe Minim atura la llibreria.

public void play()

Aquesta funció de la classe AudioPlayer inicialitza l'arxiu d'àudio.

public void close()

Aquesta funció de la classe AudioPlayer tanca la cançó.

public void detect(AudioBuffer ab)

Aquesta funció de la classe BeatDetect permet analitzar trossos d'àudio a partir de pulsos. Detecta els beats de la cançó. A més, també diferencia si detecta els beats en mode freqüència o en mode energia.

public boolean isOnset()

Aquesta funció de la classe BeatDetect retorna la variable isOnset, que només pot ser true o false perquè és boleana. Aquesta variable val true si detecta un beat, mentre que val false si no el detecta.

Aquestes 2 últimes comandes s'executen quan es tanca l'sketch.

Per arribar a l'sketch final hi ha hagut una evolució que es pot dividir en 4 tipus de proves. Tenint en compte que la part que va ocupar més temps va ser la interconnexió del Wiimote amb el PC i Processing, l'evolució cronològica de les proves és la següent: proves del Wiimote, proves de gràfics, proves de la llibreria Minim i Sketch final.

3.1 Proves del Wiimote

Un cop connectat el Wiimote amb el PC i amb Processing, les primeres proves va ser crear unes gràfiques on es representés a la finestra l'acceleració en x, y i z al llarg del temps i es guardés en el disc dur per a futures aplicacions. S'havia de buscar quina era la funció o amb quin objecte s'obtenien les 3 acceleracions (x, y i z). Un cop es tenien s'havien de dibuixar a la finestra al llarg del temps i seguidament guardar-les.

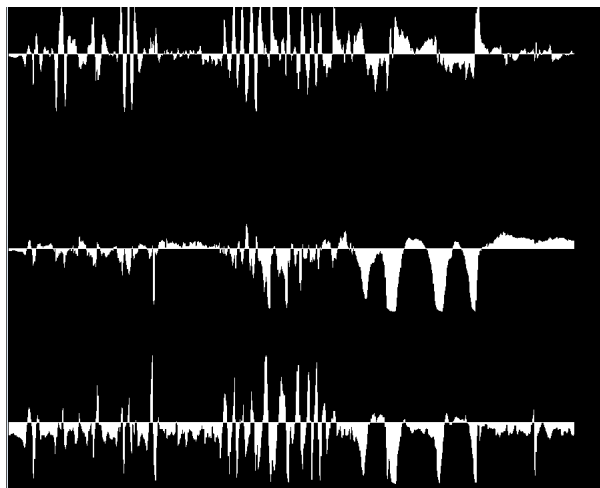


Figura 3.1. Gràfiques de les acceleracions de x, y i z

La gràfica superior és l'acceleració de X, la del mig l'acceleració de Y i la inferior la de Z. Cada 30 segons es guarda la gràfica en el disc dur en format "tif" i es fa un esborrat de pantalla per tornar a dibuixar a la finestra durant 30 segons més les acceleracions. Aquests passos es repeteixen fins que l'usuari atura l'sketch.

La prova següent va ser dibuixar a la finestra amb el comandament de la Wii com si fos un llapis. A més, també es van programar els botons 1,2 i + per canviar els colors del "llapis" i el botó - per esborrar la finestra.

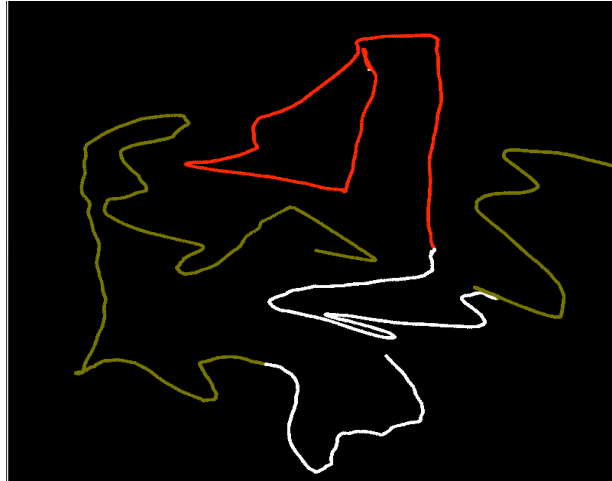
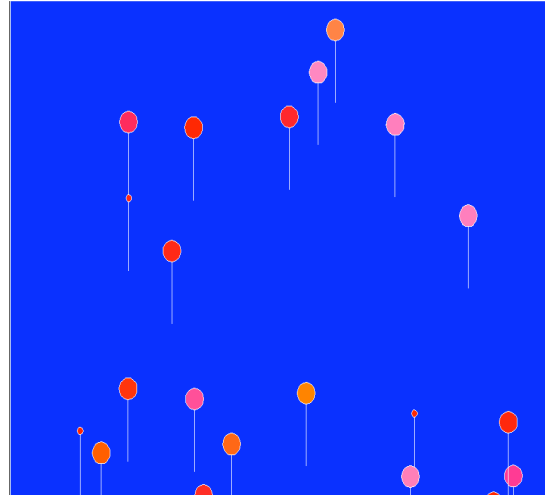
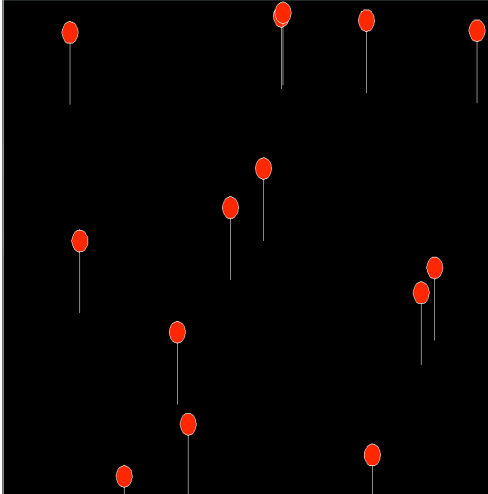


Figura 3.2 Prova del llapis de color

Una prova totalment diferent per treballar amb el Wiimote, va ser l'sketch dels xupa-xups aleatoris. La prova consistia en què el programa anava dibuixant xupa-xups de forma aleatoria per tota la finestra i, en funció de l'acceleració amb la qual es movia el comandament, el color del xupa-xup canviava. Si no hi havia acceleració, o aquesta era molt baixa, el xupa-xup era vermell mentre que, com més alta fos l'acceleració, cada cop el xupa-xup agafava un to més groguenc o rosat (depenent de si l'acceleració més alta provenia de x o y). A més, amb els botons A i B s'augmentava o disminuïa el diàmetre del xupa-xup, amb el botó 1 es pintava tota la pantalla de color negre i amb el 2 es pintava de color blau.

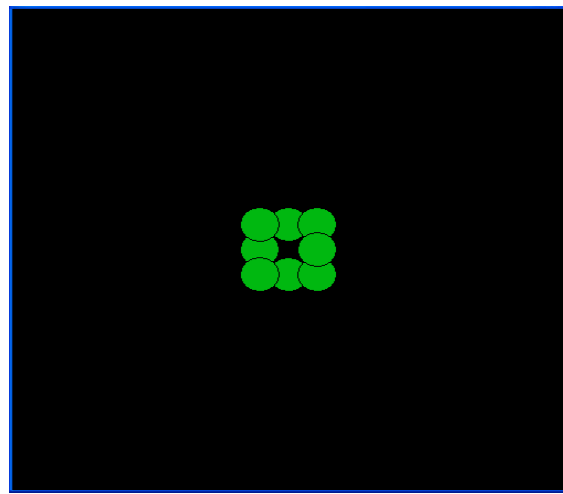
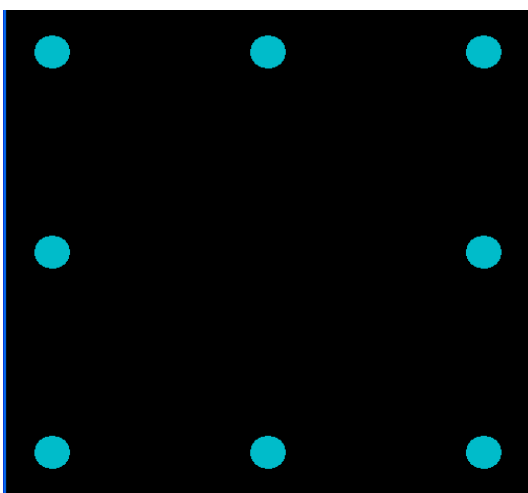


Figures 3.3 i 3.4 Proves dels xupa-xups aleatoris

3.2 Proves de gràfics

Com que no es coneixia el programa es va començar fent proves amb gràfics molt simples per anar agafant més experiència. Les primeres proves eren dibuixos senzills i sense moviment a partir de rectangles, línies, el·lipses i punts.

Una de les primeres proves de gràfics va ser aconseguir una animació sense la interacció del comandament. La prova consistia en 9 rodones que sortien des del centre de la pantalla i s'expandien cap als extrems. Quan arribaven als extrems rebotaven i tornaven al mig. A més, com més lluny estaven del centre de la finestra el seu color es tornava blau, mentre que al punt central eren de color verd. És una prova molt senzilla per treballar amb els rebots de les el·lipses i amb la varietat de colors. En aquest sketch no hi ha cap tipus d'interacció amb l'usuari.



Figures 3.6 i 3.7 Prova dels 8 cercles

En la prova següent ja hi ha la interacció de gràfics una mica més complexos amb el Wiimote. El nom de l'sketch és canonades, ja que vol emular una canonada mòbil. S'han programat els botons perquè permetin a l'usuari canviar el color de la canonada. Quan la canonada surt de la finestra torna a aquesta en una posició aleatòria cada cop diferent. És un efecte bastant espectacular, ja que es crea un efecte 3D en dues dimensions i molt senzill d'aconseguir.

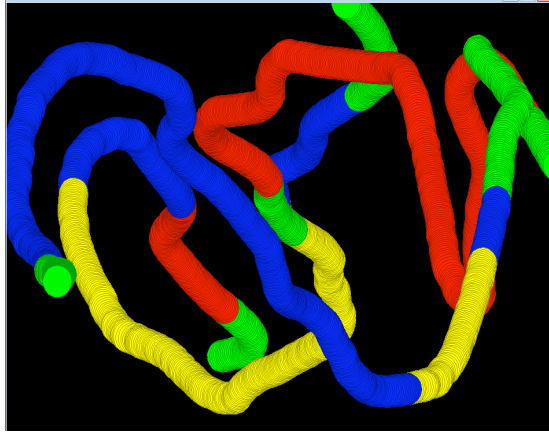


Figura 3.8 Sketch de les canonades mòbils

Totes les proves relacionades amb girs estan basades en una figura situada al mig de la pantalla que es fa girar mitjançant el comandament de la Wii, o amb diferents figures que giren sobre el seu propi eix independentment les unes de les altres. També hi ha proves on un cercle situat a un costat de la pantalla gira a partir d'un eix central causant un efecte molt interessant. Aquestes proves s'han fet per poder treballar amb rotacions mitjançant el comandament de la Wii. Hi ha diferents maneres de treballar en 3D amb Processing, les proves s'han fet amb el motor de 3 dimensions P3D perquè és més ràpid que el motor OPENGL. Un altre exemple on es treballa amb 3 dimensions, girs i so consisteix en un cub que gira i augmenta o disminueix la seva transparència en funció dels beats de la música.

Una altre prova feta amb 3D i girs consistia en un cub gran on a dins hi havia 4 cubs petits. El cub gran girava en funció de les paràmetres X i Y del ratolí, mentre que els 4 cubs petits giraven sols amb un ritme constant.

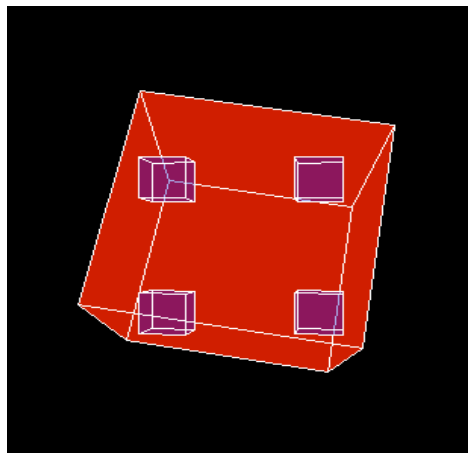


Figura 3.9 Prova dels 5 cubs

3.3 Proves amb la llibreria Minim

Després de fer proves amb gràfics i amb el Wiimote, el pas següent va ser incorporar música a l'sketch. Per això s'havien de fer proves.

La primera prova que es va fer amb la llibreria Minim consistia en posar música a una animació creada anteriorment sense que aquesta interactués amb els gràfics.

La prova següent fou crear un cercle que augmentava de mida cada cop que detectava un beat. És un sketch molt senzill però que permetia treballar amb les comandes bàsiques de la llibreria. A més, s'hi va afegir l'efecte d'esborrat que permetia veure amb més claretat quan augmentava i disminuïa l'el·lipse.

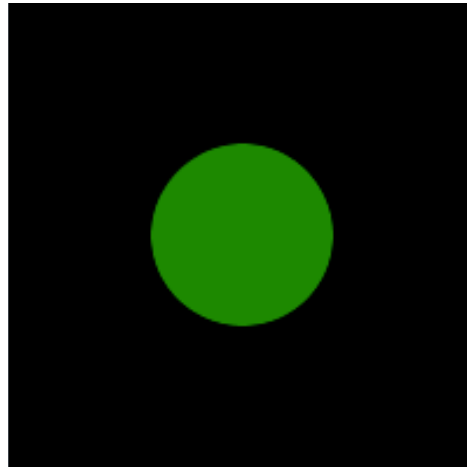


Fig 3.10 i 3.11 Prova del cercle amb àudio

Un cop s'havia comprovat que funcionava correctament aquesta llibreria amb gràfics es va decidir crear una peça on interactuessin el comandament de la Wii amb gràfics i àudio. D'aquesta interacció en van sortir dos exemples. El primer consistia en unes canonades que es movien per tota la pantalla mitjançant el comandament de la Wii i que a més, augmentava el seu radi en funció de potència del so. Els botons també es van programar perquè es pogués canviar el color de la canonada.

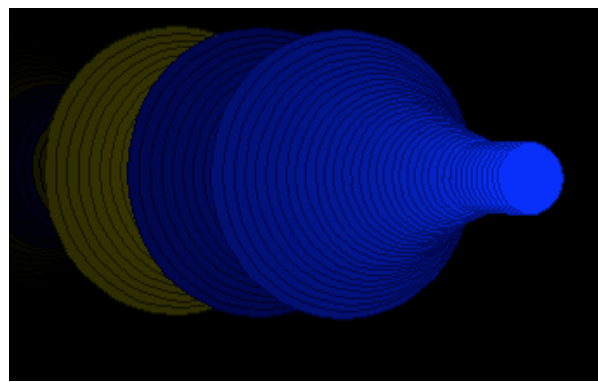
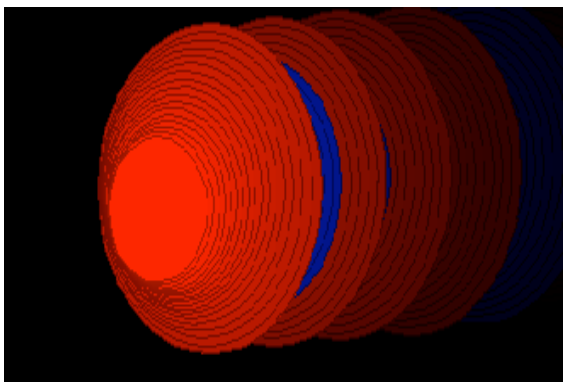


Figura 3.12 i 3.13 Prova de les canonades amb àudio

L'última prova consistia en 4 cercles dibuixats a les 4 cantonades de la finestra que augmentaven segons la música. Mitjançant el comandament de la Wii es controlava un cercle que es movia per tota la finestra. Es van programar els botons per aconseguir canviar els colors de les 4 el·lipses dels extrems. Aquest exemple ja va ser una aproximació a la peça final. A més, es va controlar també que en funció del botó que s'apretava sortissin rodones pel mig de la pantalla o es dibuixessin 2 "cortines" al mig de la pantalla tal i com es pot veure en les figures següents:

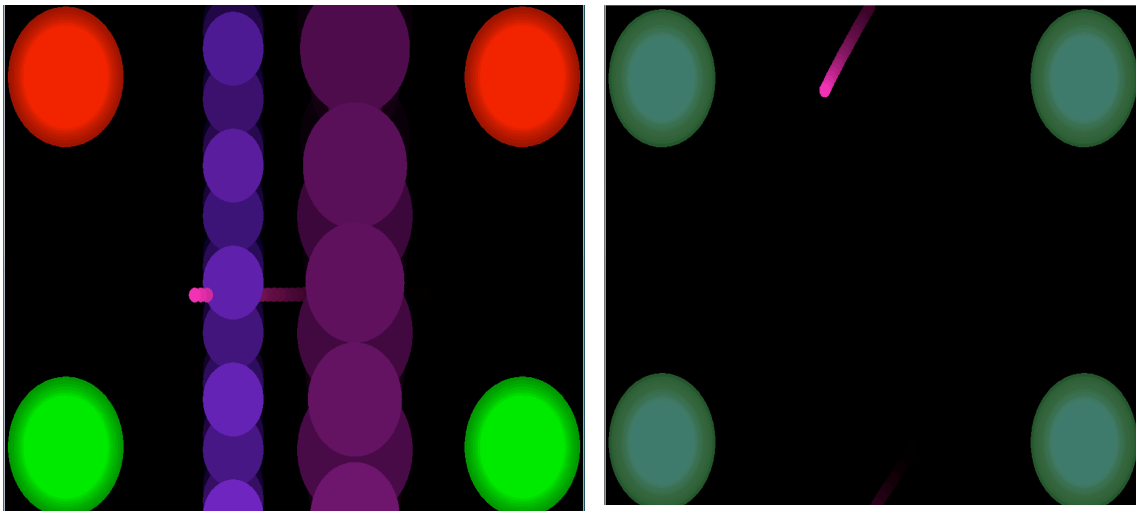


Figura 3.14 i 3.15 Últimes proves amb àudio, Wiimote i Processing.

3.4 Peça final

Un dels objectius inicials del projecte era que un cop es tingués prou documentació i s'haguessin fet prou proves, crear una peça final on interactuessin el Wiimote amb imatges en moviment. A més, també s'hi ha afegit àudio per fer més atractiva la peça final.

La peça final es pot dividir segons la interacció d'aquesta manera:

- Interacció d'àudio amb gràfics
- Interacció del Wiimote amb gràfics
 - Interacció dels botons amb gràfics
 - Interacció de l'acceleròmetre amb gràfics: moviments precisos i moviments continus.

La interacció de l'acceleròmetre amb gràfics pot ser a partir de moviments precisos que causen un efecte a la finestra o a partir de moviments continus que permeten moure un o varis objectes. A continuació hi ha dues captures de pantalla de l'sketch final.

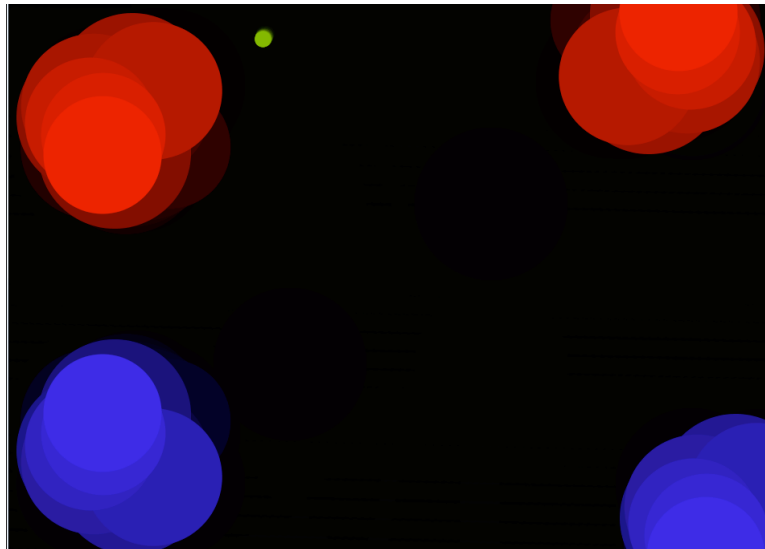


Figura 3.15 Peça final

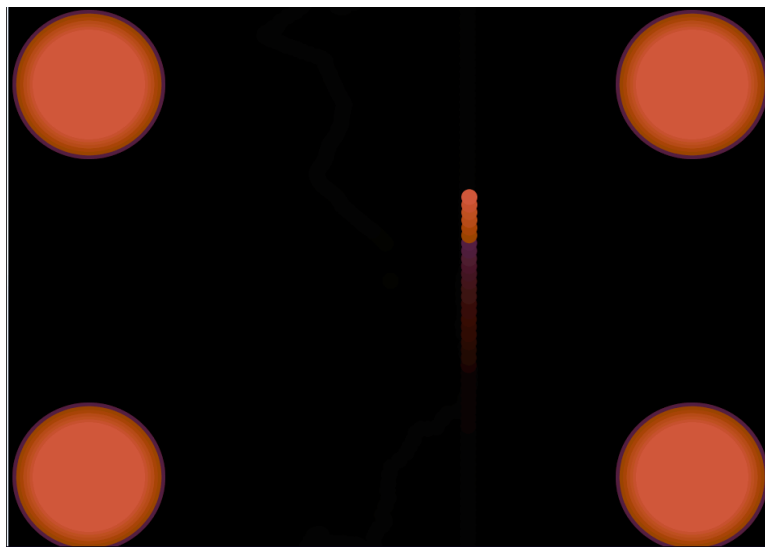


Figura 3.16 Peça final

La peça final és una evolució de les últimes proves explicades en l'apartat 3.3 on es treballa amb l'acceleració del comandament de la Wii. Es basa en un cercle, inicialment al centre de la finestra, que es mou per la pantalla mitjançant el comandament de la Wii i en 4 cercles, un a cada cantonada, que augmenten de mida a partir dels beats de l'àudio. També s'ha programat el comandament perquè per a determinats moviments creï uns efectes determinats a la finestra. Els botons estan programats per canviar el color dels cercles o per canviar-los de posició. La programació de moviments específics del Wiimote i que aquests fessin algun efecte per la finestra, ha requerit que s'estudiés primer el moviment a fer i després a partir de l'sketch abans explicat de les gràfiques de l'acceleració, s'ha gravat el moviment concret i s'ha aconseguit que aquest precís moviment causés un efecte a la finestra.

Un dels moviments que té un efecte a la pantalla està representat a la gràfica següent:

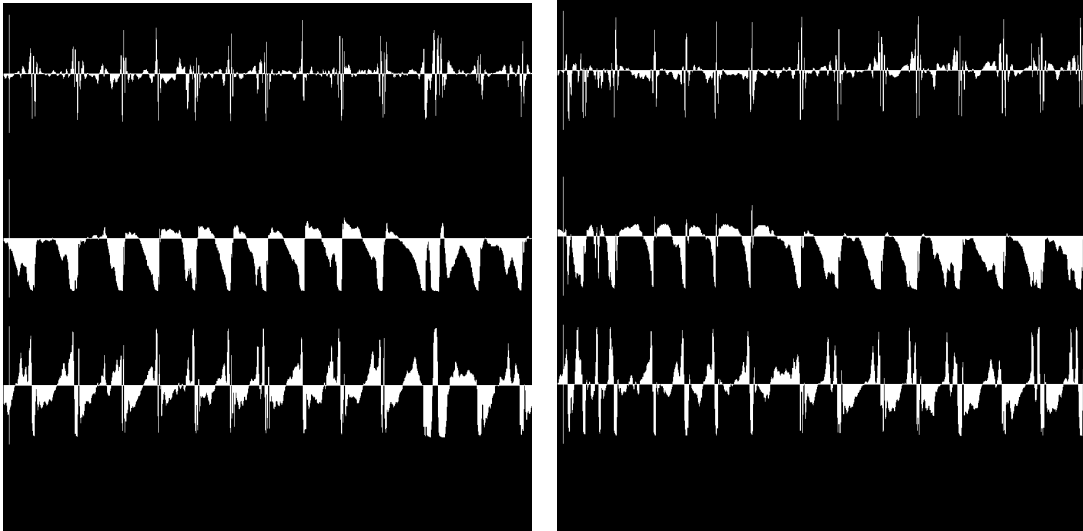


Figura 3.17 i 3.18 Moviments del comandament

No s'ha posat cap codi de les proves fetes, però si s'ha cregut convenient posar un esquema del codi de la peça final, és el següent:

S'importen les llibreries Wrj4P5, Minim i Loc.

Es declaren totes les variables i objecte que s'utilitzaran a la resta de codi.

Es declara un vector de 5 cercles.

```
void setup()
```

```
{
```

```
    S'inicialitza la finestra.
```

```
    Es connecta el comandament de la Wii.
```

```
    S'obra la llibreria Minim i es carrega la cançó.
```

```
    Es creen 5 cercles, es pinten d'un color inicial, es posicionen i s'assigna un valor al seu radi.
```

```
    S'inicialitzen variables.
```

```
}
```

```
void draw()
```

```
{
```

```
    Es simplifica el nom de les acceleracions de X, Y i Z del comandament.
```

```
    Es creen 5 variables aleatòries amb la funció random().
```

```
    Es treu el contorn de les rodones.
```

```
    Es detecten els beats de la cançó amb la funció beat.detect().
```

```
    Es creen un seguit de condicions per controlar els efectes dels botons del comandament de la Wii.
```

```
if((iy>3)&&(ix<1))
```

```
{
```

```
    Si es produeix un moviment que té una acceleració de Y superior a 3 i l'acceleració de X és inferior a 1 es crea l'efecte "confetti".
```

```
}
```

```
if((iz<1.5)&&(iy<1.5)&&(ix>2))
{
```

Si es produeix un moviment que fa que les acceleracions de Z i de Y sigui inferior a 1.5 i la de X superior a 2, augmenta la mida del cercle que es mou amb el Wiimote.

```
}
```

Es controla tot el tema de l'àudio.

Els radis dels 4 cercles de les cantonades varien en funció de l'àudio mitjançant l'assignació de la variable eRadius a la mida dels seus radis.

dibuixa tots els cercles amb la funció: c[x].dibuixa():

```
if((c[2].Cx)<0)
if(c[2].Cy<0)
c[2].Cx=(c[2].Cx+(ix*10))%width;
c[2].Cy=(c[2].Cy+(iy*10))%height;
```

Amb aquestes condicions i comandes es controla que quan el cercle surt per la part inferior o superior de la pantalla torni a aparèixer per l'altre banda.

```
if(time>2000)
```

```
{
```

Quan el temps arriba a 2000 segons els cercles es mouen de forma aleatòria

```
}
```

```
fill(0,25);
rect(0,0,width,height);
time++;
}
```

```
void buttonPressed (RimokonEvent evt, int rid)
```

```
{
```

Programació dels botons del comandament amb variables booleans.

```
}
```

```
void stop()
```

```
{
```

Parar la cançó i la llibreria Minim.

```
}
```

Durant la realització de les proves amb el Wiimote s'ha detectat un problema en les llibreries d'aquest: quan es mou molt ràpidament el Wiimote, la representació del moviment a la pantalla és la mateixa que si l'orientació d'aquest fos cap avall, per tant, la llibreria que interpreta els moviments del Wiimote no detecta correctament l'orientació del Wiimote.

A la memòria del projecte s'hi ha ajuntat un CD amb els codis de les diferents proves i de la peça final i també el codi de les llibreries utilitzades.

BIBLIOGRAFIA

Material Bàsic

- [1] Web de Processing. <http://www.processing.org>
- [2] Reas, C. and Fry,B., *Processing: a programming handbook for visual designers and artists*, Massachusetts (2007).
- [3] Greenberg, I., *Processing: Creative Coding and Computational Art* , (2007)
- [4] Tutorial bàsic en francès de Processing.
http://www.ecole-art-aix.fr/rubrique.php?id_rubrique=81

Fòrums

- [5] Fòrum oficial de la Web de Processing.
http://processing.org/discourse/yabb_beta/YaBB.cgi
- [6] Fòrum en castellà de Processing <http://foro.visualp5.net>

Wiimote

- [7] Web amb informació sobre la Wii, Wiimote i connexió <http://www.wiili.org>
- [8] Web amb informació sobre el Wiimote <http://www.revogamers.net>
- [9] Web amb informació sobre la connexió del Wiimote amb el PC.
<http://www.christianmeinke.com/2007/04/10/wiimote-communication/>

Persones importants de la comunitat Processing i Wiimote

- [10] Web de Ben Fry. <http://benfry.com/>
- [11] Web de Casey Reas <http://reas.com/>
- [12] Web de ClassicIII, creador de la classe Wrj4P5.
http://www.geocities.jp/classicIII_newweb
- [13] Web de TM. <http://sosobering.free.fr/art/>
- [14] Web de Robert Hodgins. <http://www.flight404.com>
- [15] Web de Toxi. <http://www.toxi.co.uk/p5/>
- [16] Web de Jared Tarbell.
<http://www.complexification.net/gallery/machines/citytraveler/>
- [17] Web de Amit Pitaru. <http://www.pitaru.com/>
- [18] Web de Marius Watz. <http://unlekker.net>
- [19] Web de MarlonJ <http://marlonj.darkgreenmedia.com/site/>
- [20] Web d'Alba G. Corral <http://www.albagcorral.com/>