

# MICROSOFT KINECT XBOX 360



Por:

*Cristhian Andrés Apolo Cevallos*

MANUAL INSTALACIÓN





Originalmente conocido por el nombre en clave “Project Natal”, **Kinect** es un controlador de juego libre y entretenimiento creado por Alex Kipman, de la empresa PrimeSense, desarrollado para Microsoft para la videoconsola Xbox 360, y desde junio del 2011 para PC a través de Windows 7 y Windows 8.

Kinect permite a los usuarios controlar e interactuar con la consola sin necesidad de tener contacto físico con un controlador de videojuegos tradicional, mediante una interfaz natural de usuario que reconoce gestos, comandos de voz y objetos e imágenes. El dispositivo tiene como objetivo primordial aumentar el uso de la Xbox 360, más allá de la base de jugadores que posee en la actualidad.

## Descripción General De Dispositivo

Xbox Kinect se compone de los siguientes dispositivos USB:

- Eje genérico
- NUI Audio
- NUI cámara
- Motor NUI

### NUI Audio

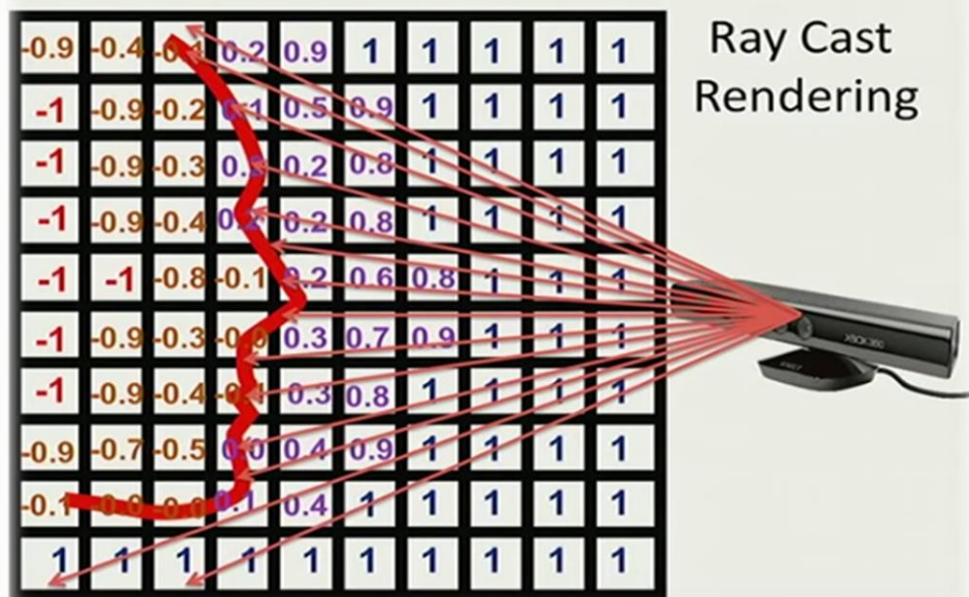
El dispositivo de audio proporciona audio combinada de cuatro micrófonos de Kinect. También proporciona el hardware de cancelación de ruido restando audio del juego de la televisión y lo que representa para el diseño 3D de la habitación.

El dispositivo de audio tiene los siguientes criterios de valoración utilizados para la transferencia de datos:

- Punto de llegada 0x81 - EN, a granel, 512 bytes
- Punto final 0x01 - OUT, a granel, 512 bytes
- Punto final 0x82 - EN, isócrono, 524 bytes
- Punto final 0x02 - hacia fuera, Isócronas, 72 bytes

## ✚ Cámara NUI

El dispositivo más interesante y útil, las cámaras proporcionan, tanto imágenes en color, como en profundidad. La cámara de infrarrojo (IR) produce matrices de profundidad las cuales se separan mediante fórmulas matemáticas. Los valores de profundidad de brillo es una matriz unidimensional.



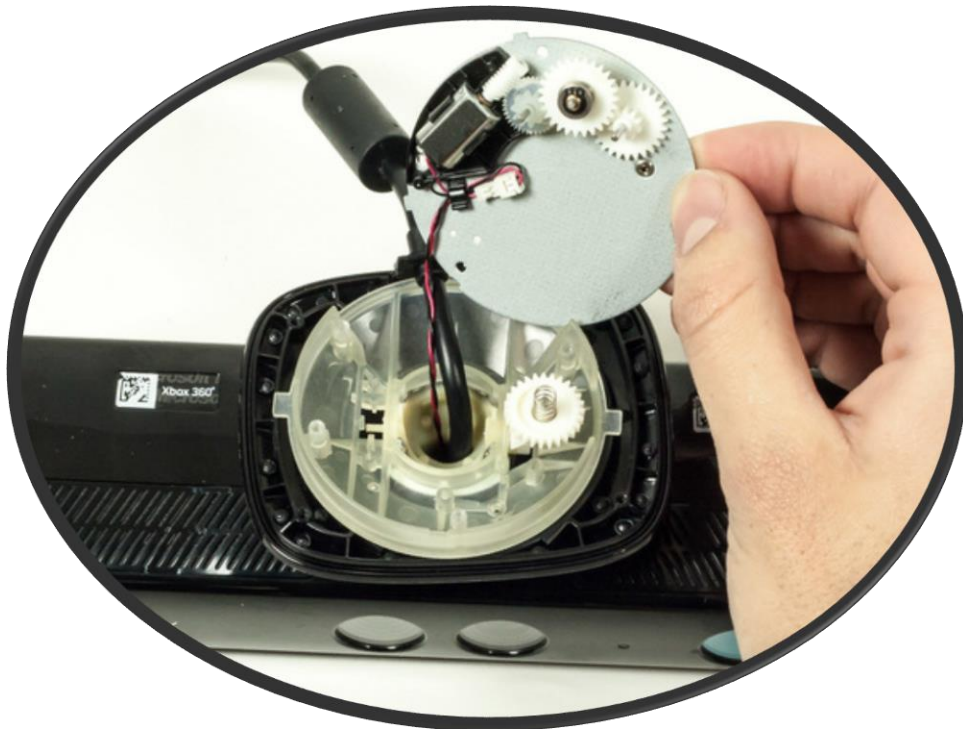
Además de la variable de control 0x00 predeterminada, el dispositivo tiene la cámara:

- Punto final 0x81 - IN, Isócronas, 1920 bytes
- Punto final 0x82 - EN, isócrono, 1920 bytes

La captura de imágenes RGB el sensor lo interpreta como matrices (x, y) de pixeles.

## **Motor NUI**

El dispositivo controla el motor real del motor para inclinar el Kinect. Además, el dispositivo motor permite la lectura de los datos de 3 ejes de acelerómetro incorporados. El dispositivo utiliza sólo el motor de punto final de control para toda la comunicación de datos.



# INSTALACIÓN KINECT EN MAC OSX 10.12

## *Instalación Xcode*

1. Descargar e instalar Xcode (versión 7.3) a través de App Store.

Es el entorno de desarrollo integrado (IDE) de Apple Inc. y se suministra gratuitamente junto con Mac OS X. Xcode trabaja conjuntamente con Interface Builder, una herencia de NeXT, una herramienta gráfica para la creación de interfaces de usuario.

2. Descargar e instalar Command Line Tools para OS X 10.12 (*commandlinetoolsox10.9forxc6.2.dmg*).

Son librerías y comandos extras para Xcode.

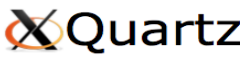


## *Adiciones*

1. Descargar e instalar XQuartz (versión 2.7.11).

Es una versión de código abierto del sistema X Window X.org para Mac. (<https://www.xquartz.org/releases/archive.html>)





**Home**

**Releases**

**Support**

**Contributing**

**Bug Reporting**

**GitHub**


## Releases Archive

### Installer Packages

- [XQuartz 2.7.11](#) - 2016-10-29
- [XQuartz 2.7.11\\_rc2](#) - 2016-10-26 - **Pre-Release**
- [XQuartz 2.7.11\\_rc1](#) - 2016-10-24 - **Pre-Release**
- [XQuartz 2.7.10](#) - 2016-10-22
- [XQuartz 2.7.10\\_rc5](#) - 2016-10-07 - **Pre-Release**
- [XQuartz 2.7.10\\_rc4](#) - 2016-09-27 - **Pre-Release**
- [XQuartz 2.7.10\\_rc3](#) - 2016-09-27 - **Pre-Release**
- [XQuartz 2.7.10\\_rc2](#) - 2016-09-11 - **Pre-Release**
- [XQuartz 2.7.10\\_rc1](#) - 2016-09-08 - **Pre-Release**
- [XQuartz 2.7.10\\_beta2](#) - 2016-05-31 - **Pre-Release**
- [XQuartz 2.7.10\\_beta1](#) - 2016-05-07 - **Pre-Release**
- [XQuartz 2.7.9](#) - 2016-05-05
- [XQuartz 2.7.9\\_rc2](#) - 2016-01-07 - **Pre-Release**
- [XQuartz 2.7.9\\_rc1](#) - 2016-01-06 - **Pre-Release**
- [XQuartz 2.7.9\\_beta2](#) - 2015-12-28 - **Pre-Release**
- [XQuartz 2.7.9\\_beta1](#) - 2015-10-29 - **Pre-Release**
- [XQuartz 2.7.8](#) - 2015-10-17 - **First release supported on El Capitan**
- [XQuartz 2.7.7](#) - 2014-08-18 - **First release supported on Yosemite**
- [XQuartz 2.7.6](#) - 2014-05-17
- [XQuartz 2.7.5](#) - 2013-11-10 - **First release supported on Mavericks**
- [XQuartz 2.7.4](#) - 2012-09-27
- [XQuartz 2.7.3](#) - 2012-08-27
- [XQuartz 2.7.2](#) - 2012-06-01 - **First release supported on Mountain Lion**
- [XQuartz 2.7.1](#) - 2012-02-20
- [XQuartz 2.7.0](#) - 2011-11-04 - **First release supported on Lion**
- [XQuartz 2.6.3](#) - 2011-07-20 - **Final Leopard Release**
- [XQuartz 2.6.2](#) - 2011-04-30 - **No Leopard Release**

## 2. Descargar e instalar CMake (*cmake-3.7.1-Darwin-x86\_64.dmg*).

Es un sistema de automatización de construcción de plataforma cruzada.  
(<https://cmake.org/download/>)








[About](#) [Resources](#) [Developer Resources](#) [Download](#)

---

### Get the Software

You can either download binaries or source code archives for the [latest stable](#) or [previous](#) release or access the [current development](#) (aka nightly) distribution through Git. This software may not be exported in violation of any U.S. export laws or regulations. For more information regarding Export Control matters please go to [http://kitware.com/export\\_control/index.html](http://kitware.com/export_control/index.html).

-  [Join the mailing list](#)
-  [CMake success stories](#)
-  [Attend a training course](#)
-  [Buy the book](#)
-  [Purchase support](#)

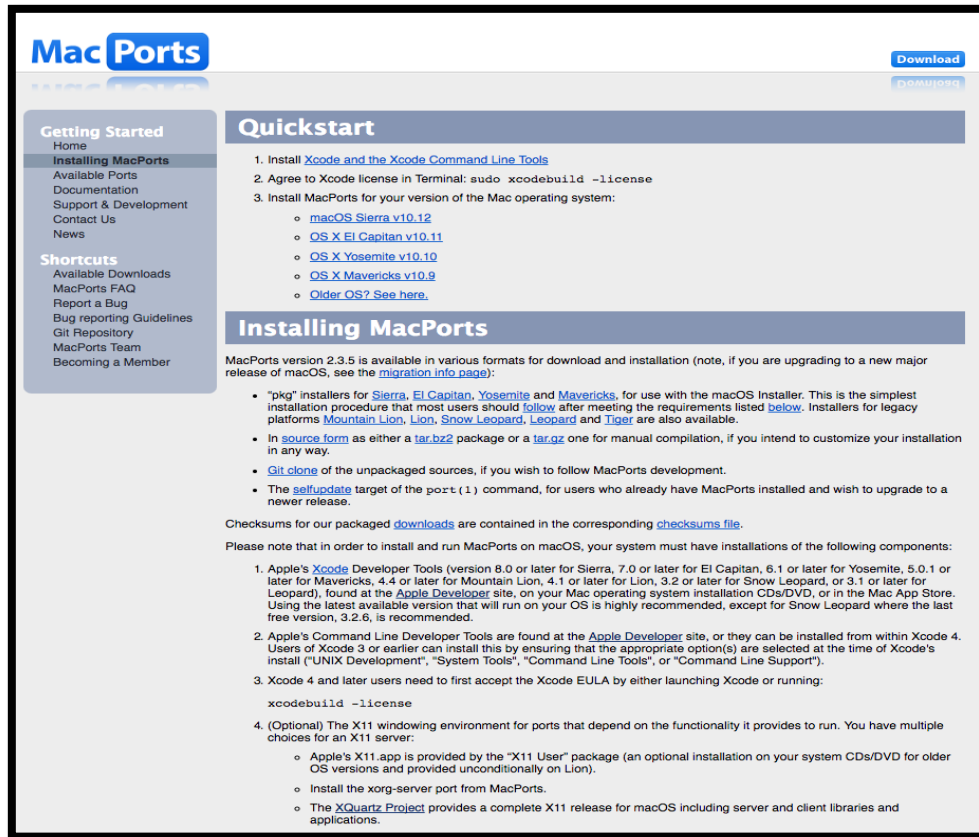
### Latest Release (3.7.2)

The release was packaged with CPack which is included as part of the release. The .sh files are self extracting gzipped tar files. To install a .sh file, run it with /bin/sh and follow the directions. The OS-machine.tar.gz files are gzipped tar files of the install tree. The OS-machine.tar.Z files are compressed tar files of the install tree. The tar file distributions can be untared in any directory. They are prefixed by the version of CMake. For example, the Linux-x86\_64 tar file is all under the directory cmake-Linux-x86\_64. This prefix can be removed as long as the share, bin, man and loc directories are moved relative to each other. To build the source distributions, unpack them with zip or tar and follow the instructions in README.txt at the top of the source tree. See also the [CMake 3.7 Release Notes](#). Source distributions:

Platform	Files
Unix/Linux Source (has \n line feeds)	<a href="#">cmake-3.7.2.tar.gz</a>
	<a href="#">cmake-3.7.2.tar.Z</a>
Windows Source (has \r\n line feeds)	<a href="#">cmake-3.7.2.zip</a>

### 3. Descargar e instalar MacPorts (*MacPorts-2.3.5-10.12-Sierra*)

MacPorts es un sistema fácil de usar para compilar, instalar y gestionar el software de código abierto. (<https://guide.macports.org/#installing.shell>)



## Dependencias

Librerías para código abierto para trabajar con Kinect.

1. Abrir Terminal
2. Escribir:
  - `sudo port install libtool`  
(<http://mirror.cedia.org.ec/gnu/libtool/>)
  - `sudo port install libusb +universal`

O instalar mediante HomeBrew:

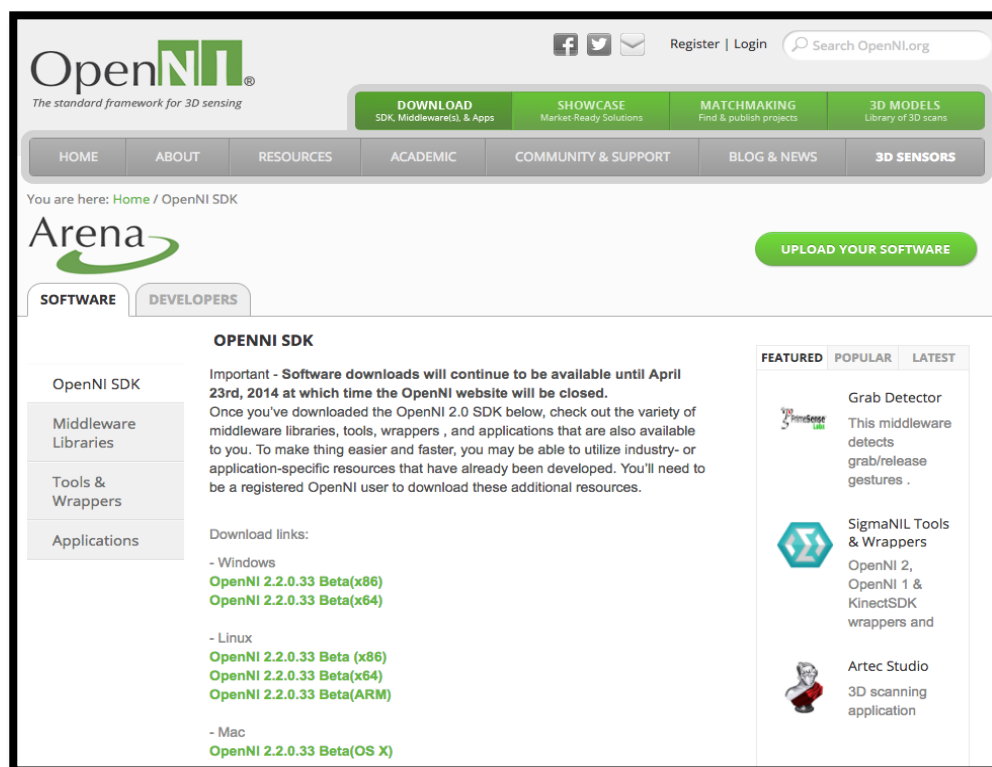
1. Abrir Terminal
2. Escribir:
  - `brew install libtool`
  - `brew install libusb --universal`

Para continuar, cree una carpeta llamada preferiblemente “**Kinect**” donde se va a proceder a guardar todo lo que descarguemos:

### ***OpenNI, Sensor Kinect y NITE***

1. Instalación pre formulas
  - 1) Abrir terminal
  - 2) Escribir:
    - brew tap homebrew/science
    - brew tap totakke/openni
2. Instalación OpenNI, SensorKinect, y NITE
  - brew install oppeni

O puede descargarlo de la página oficial (<http://openni.ru/openni-sdk/index.html>)



Después de descomprimir el archivo descargado, moverlo a la carpeta de trabajo “Kinect” creada para el proyecto.

- 1) Abrir el terminal
- 2) Dirigirse a la carpeta OpenNi-Bin copiada anteriormente:  
`cd ../../Kinect/OpenNI-MacOSX-x64-2.2`
- 3) `sudo sh ./install.sh`



- brew install sensor-kinect

O puede descargarlo del repositorio GitHub <https://github.com/avin2/SensorKinect>.

Para instalarlo se descomprime la carpeta descargada dentro de la carpeta Kinect del proyecto. Entramos dentro de la carpeta Bin del Sensor y descomprimos el .tar.bz2 para MAC.

- 1) Abrir terminal
- 2) Nos dirigimos a la carpeta descomprimida  
`cd ../../Kinect/SensorKinect-unstable/Bin/Sensor-Bin-MacOSX-v5.1.2.1`
- 3) `sudo sh ./install.sh`

- brew install nite

O puede descargarlo del repositorio GitHub <https://github.com/desirino/spark/tree/master/kinect/NITE-Bin-Dev-MacOSX-v1.5.2.21>

Para instalarlo se descomprime la carpeta descargada dentro de la carpeta Kinect del proyecto.

- 1) Abrir el terminal
- 2) Dirigirse a la carpeta OpenNi-Bin copiada anteriormente.  
`cd ../../Kinect/OpenNI-MacOSX-x64-2.2`
- 3) `sudo sh ./install.sh`

\*Es importante seguir el orden de los pasos anteriores, en el caso que diera algún error o no se lo hizo ordenadamente pueden volver a desinstalar todo y volverlo a instalar.

Este link les ayudará como hacerlo: <http://blog.nelga.com/how-to-uninstall-microsoft-kinect-drivers-xquartz-cmake-macports-openni-nite-sensorkinect-on-your-mac/>

Si quieres controlar si se instalaron los drivers de Kinect, abre el terminal y escribe:

```
system_profiler SPUSBDataType
```

Muestra todos los puertos USB utilizados, si sale el 3.0 y los sensores de audio y cámara de Kinect, entonces se instaló correctamente.

# PROCESSING



## ¿Qué es Processing?

Procesamiento es un lenguaje de programación de código abierto y un medio para las personas que quieren crear imágenes, animaciones e interacciones. Inicialmente desarrollado para servir como un cuaderno de bocetos de software y para enseñar los fundamentos de la programación de computadoras dentro de un contexto visual, procesamiento también se ha convertido en una herramienta para la generación de trabajo profesional terminada.

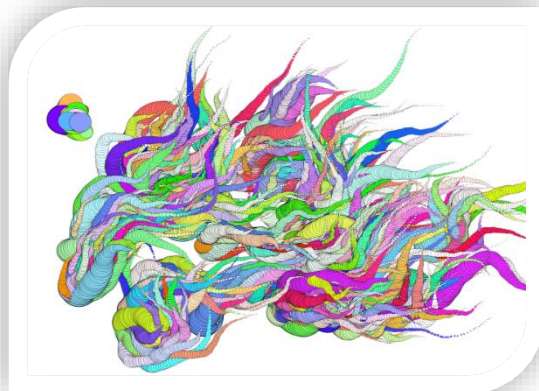
Hoy en día, hay decenas de miles de estudiantes, artistas, diseñadores, investigadores y aficionados que utilizan el procesamiento para el aprendizaje, la creación de prototipos y producción.

Algunos ejemplos de proyectos como:

[EJEMPLO 1](#)

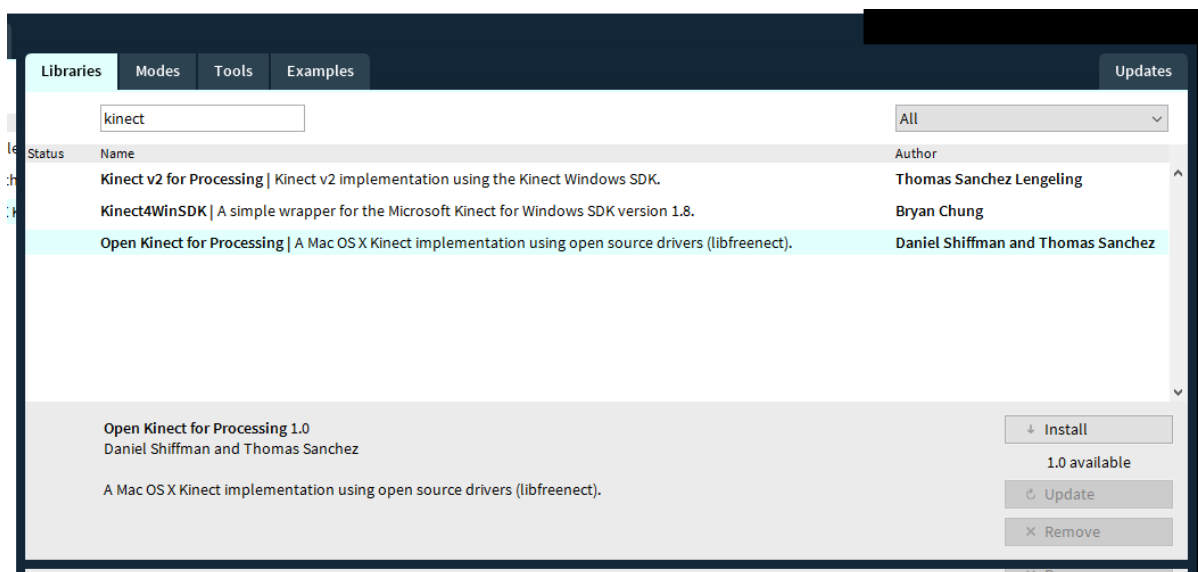
[EJEMPLO 2](#)

[EJEMPLO 3](#)



## Instalación Processing + Kinect

1. Descargar Processing  
<https://processing.org/download/?processing>
2. En la barra de menú de Processing, nos dirigimos a “Sketch” → Importar Librería → Añadir Biblioteca. Y buscamos en el buscador:
  - Buscamos “Kinect” y descargamos la librería de Daniel Shiffman y Thomas Sanchez.



- Importación de librería a usar y constructor para crear un objeto Kinect:

```
import org.openkinect.processing.*;

// Declaración objeto Kinect a trabajar.
Kinect kinect;
// Kinect imagen para trabajar con Processing
PImage cam = createImage(640, 480, RGB);
```

- Función setup(): Se utiliza para definir las propiedades iniciales de medio ambiente, tales como tamaño de pantalla y para cargar los medios de comunicación tales como imágenes y fuentes que se inicia el programa.

```
void setup(){
  // Dimensión pantalla
  size(650, 550);
  //Constructor
  kinect = new Kinect(this);

  // Inicia la profundidad IR
  kinect.initDepth();
}
```

- Función draw(): la función ejecuta continuamente las líneas de código contenidos dentro de su bloque hasta que el programa se detiene o noLoop() es llamado.

```
void draw() {
    background(bgColor);

    // Obtiene el deptcamh crudo como matriz de enteros
    int[] depth = kinect.getRawDepth();
    for (int x = 0; x < kinect.width; x++) {
        for (int y = 0; y < kinect.height; y++) {
            int offset = x + y * kinect.width;
            int d = depth[offset];
```

Para acceder a los datos desde el sensor Kinect. Actualmente, la biblioteca pone a disposición el uso de datos a en cinco formas:

- **PImage** (RGB) de la cámara de vídeo de Kinect.
- **PImage** (grayscale) de la cámara kinect IR.
- **PImage** (grayscale) con el brillo de cada píxel asignado a la profundidad (= más brillantes más cerca).
- **PImage** (RGB) con tonalidad de cada píxel asignado a profundidad.
- **int[]** array con los datos de profundidad primas (11 números de bits entre 0 y 2048).

Esta librería fue adaptada e implementada en Processing por este característico profesor en artes en el Programa de Telecomunicaciones Interactivas (ITP) en la Universidad de Nueva York Tisch School of the Arts .

Dispone en su página web de orientación y guías, así como un curso online de Processing, y Processing + Kinect en Youtube, para poder empezar a familiarizar con el código de la librería junto con Processing:

1. Curso Processing con Kinect  
<http://shiffman.net/p5/kinect/> // Página Web  
<https://www.youtube.com/watch?v=WH31daSj4nc&list=PLRqwx-V7Uu6bw0bVn4M63p8TMJf3OhGy8> // Curso Youtube
2. Curso Processing + Kinect  
<https://www.youtube.com/watch?v=QmVNgdapJJM&index=1&t=1s&list=PLRqwx-V7Uu6ZMIWHdcy8hAGDy6IaoxUKf>
3. Guía para Processing y Box2D  
<http://natureofcode.com/book/introduction/>

## Librerías Utilizadas

### JBOX2D for Processing

(descargado de <https://github.com/shiffman/Box2D-for-Processing>)

Box2D es un motor ++ C fuente abierta para la simulación de cuerpos rígidos en 2D. Box2D está desarrollado por Erin Catto y tiene el zlib licencia. Mientras que la licencia zlib que no requiere confirmación, te animamos a dar crédito a Box2D en su producto.

Una de las cosas fundamentales para realizar sobre Box2D es que es un verdadero motor de física. Box2D no sabe nada de gráficos por ordenador y el mundo de píxeles; se trata simplemente de una biblioteca que lleva en número y escupe más números. ¿Y cuáles son esos números? Metros, kilogramos, segundos, etc. Todas las mediciones y cálculos de Box2D son para las mediciones de sólo el mundo real de su "mundo" es un plano de dos dimensiones con la parte superior, inferior, izquierdo y bordes derecho.

<http://natureofcode.com/book/chapter-5-physics-libraries/>

//Pag Shiffman

<https://www.youtube.com/watch?v=iNHfAFx8ktQ>  
Box2D

//Curso Shiffman para

### Kinect for Processing

(descargado de processing)

Este repositorio es para para usar el sensor de Microsoft Kinect con Processing. Utiliza los controladores de código abierto desde las libfreenect y libfreenect2 proyectos.

<http://shiffman.net/p5/kinect/reference/>

//API libreria

### BlobDetection

(descargado de processing)

Esta biblioteca tiene como objetivo hacer la visión por ordenador mediante la búsqueda de 'manchas' en una imagen, es decir, las zonas cuyo brillo está por encima o por debajo de un determinado valor.

Permite calcular blobs'edges así como caja blobs'bounding. Sin embargo, esta biblioteca no realizar el seguimiento de burbuja, que sólo trata de encontrar todas las gotas cada cuadro se alimenta co

<https://www.youtube.com/watch?v=ce-2l2wRqO8&index=7&list=PLRqWX-V7Uu6bw0bVn4M63p8TMJf3OhGy8>

//Tutorial

BlobDetection Shiffman

<http://www.v3ga.net/processing/BlobDetection/index-page-home.html>

//Pág. Oficial

## Toxiclibs (descargado de processing)

Toxiclibs es una recopilación independiente, abierto fuente de la biblioteca de tareas de diseño computacional con Java & Procesamiento desarrollados por "toxi" Karsten Schmidt. Las clases se mantienen bastante genérico a propósito con el fin de maximizar su reutilización en diferentes contextos que van desde el diseño generativo, la animación, la interacción / diseño de la interfaz, visualización de datos a la arquitectura y la fabricación digital, el uso como herramienta de enseñanza y más.

<http://toxiclibs.org/docs/p5/>

//API librería



## Bibliografía

- [1] <https://www.youtube.com/watch?v=JZC6wUXBubs> //INSTALACION  
MACPORTS
- [2] [https://www.youtube.com/watch?v=ZfW\\_BYlvglk&t=1s](https://www.youtube.com/watch?v=ZfW_BYlvglk&t=1s) //INSTALACION  
KINECT EN MAC OSX 10.12
- [3] [https://openkinect.org/wiki/Hardware\\_info](https://openkinect.org/wiki/Hardware_info) //Info. hardware
- [4] <http://stackoverflow.com/questions/10335747/how-to-download-xcode-4-5-6-7-8-and-get-the-dmg-or-xip-file/10335943#10335943> //Descargar Xcode 7.3
- [5] <https://developer.apple.com/download/more/> //Xcode y Command  
Line Tools
- [6] <https://www.xquartz.org/releases/XQuartz-2.7.5.html> //XQuartz
- [7] <https://mecatronicauaslp.wordpress.com/2014/01/20/instalar-opencv-en-mac-os-x-10-9-mavericks/> //CMake
- [8] <https://guide.macports.org> // MACPORTS
- [9] <http://openni.ru/openni-sdk/index.html> //OpeniNi
- [10] <http://blog.nelga.com/how-to-uninstall-microsoft-kinect-drivers-xquartz-cmake-macports-openni-nite-sensorkinect-on-your-mac/> //Uninstall
- [11] <http://processing-spain.blogspot.com.es/> //Curso Processing
- [12] <https://processing.org/reference/> //Librerías Processing
- [13] <http://shiffman.net/p5/kinect/> //Processing con Kinect  
curso
- [14] <https://github.com/shiffman/OpenKinect-for-Processing> //Repositorio ejemplos  
Kinect GitHub Shiffman
- [15] <http://shiffman.net/p5/kinect/reference> //Referencias librería  
OpenKinect-Processing
- [16] <https://github.com/shiffman/The-Nature-of-Code-Examples> //Repositorio  
ejemplos Processing GitHub Shiffman
- [17] <https://www.openprocessing.org/> //Proyectos con  
Processing
- [18] <https://forum.processing.org/two/discussion/2625/can-kinect-physics-code-examples> // Foro Proyecto Kinect
- [19] <http://box2d.org/downloads/> //Librería Box2D
- [20] <http://natureofcode.com/book/chapter-5-physics-libraries/> //Referencias librería  
Toxiclibs