

AiVDJ

intelligent generative audio/visual system.

Alex Wolfe | Jake Lyons | Melissa Alleyne

Downloading AiVDJ

(Instructions on how to download and run AiVDJ.)

WINDOWS

A compiled version of AiVDJ is included in the handin zip under Work/AiVDJ_executable.zip. Install the drivers for the kinect (described below) and then simply unzip and run AiVDJ_debug.exe.

INSTALLING KINECT DRIVERS

Whether running the exe or downloading for development, you will need to correct kinect drivers. If you already have other kinect drivers installed, you will need to uninstall them and install the included libfree drivers. After uninstalling any previous versions and plugging in your kinect, it will show up in the device manager as an unrecognized human interface device. You will need to update the motor driver first. Under the driver properties, click update driver and manually select to directory location to the `kinect_drivers/win/inf/xbox nui motor/`. Next, unrecognized camera and audio kinect devices will show up. Repeat the instructions above, changing the path names to `kinect_drivers/win/inf/xbox nui camera/` and `kinect_drivers/win/inf/xbox nui audio/` respectively.

COMPILING THE SOURCE

if you would you like to download/compile the source code it is available at

<https://github.com/alexiswolfish/AiVDJ/tree/movedData>

you can download the repository as a zip from the site or clone it using git.

AiVDJ was built using openFrameworks, a cross platform open source toolkit for creative coding in C++. To compile the project, you can download openframeworks from [their github](#) or their [downloads page](#). After downloading the additional libraries and placing them in your openFrameworks/addons directory, the simplest way to create a project is to use the project generator found under `apps/devApps/projectGenerator/`, selecting the appropriate addons, and replacing the src and bin/data folder with ours. More detailed instructions can be found on the [openFrameworks site](#)

Additional Libraries

ofxColourTheory

(alex's color library, based off of toxiclibs color library for processing)

<https://github.com/alexiswolfish/ofxColourTheory>

ofxCv

(kyle mcdonald's vision library, used for contour finding/blob tracking)

<https://github.com/kylemcdonald/ofxCv>

ofxKinect

(kinect support for OpenFrameworks)

<https://github.com/ofTheo/ofxKinect>

ofxUI

<https://github.com/rezaali>

(included in your openFrameworks download)

ofxOpenCv

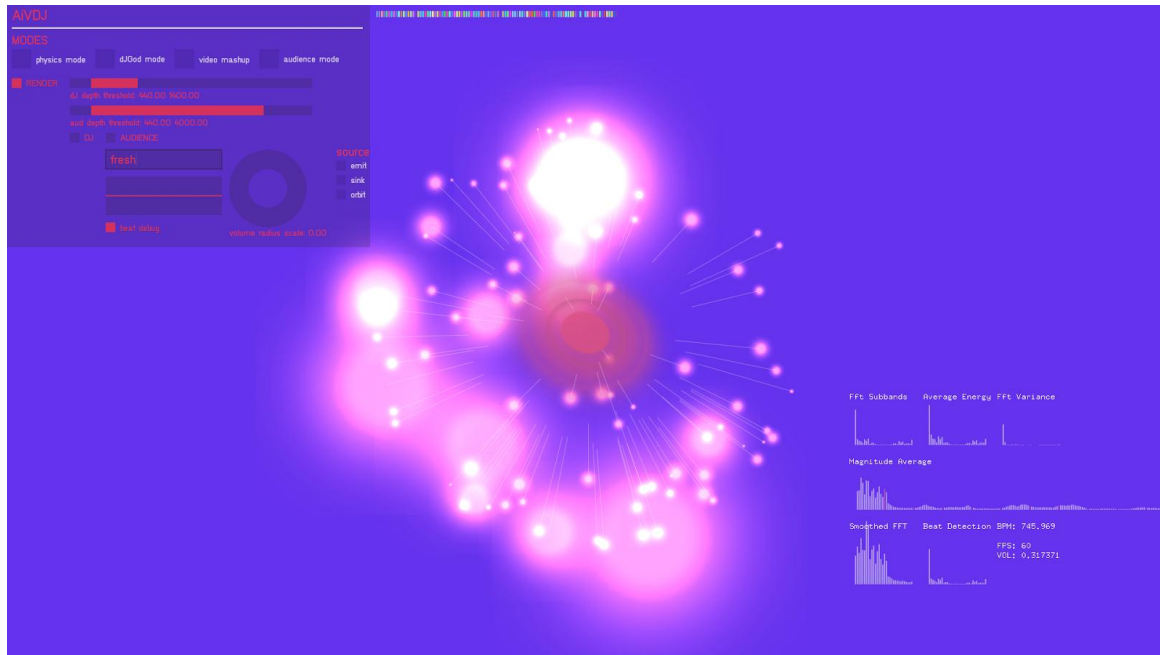
vision library for working with images/video

ofxXmlSettings

for saving GUI settings

Running the system

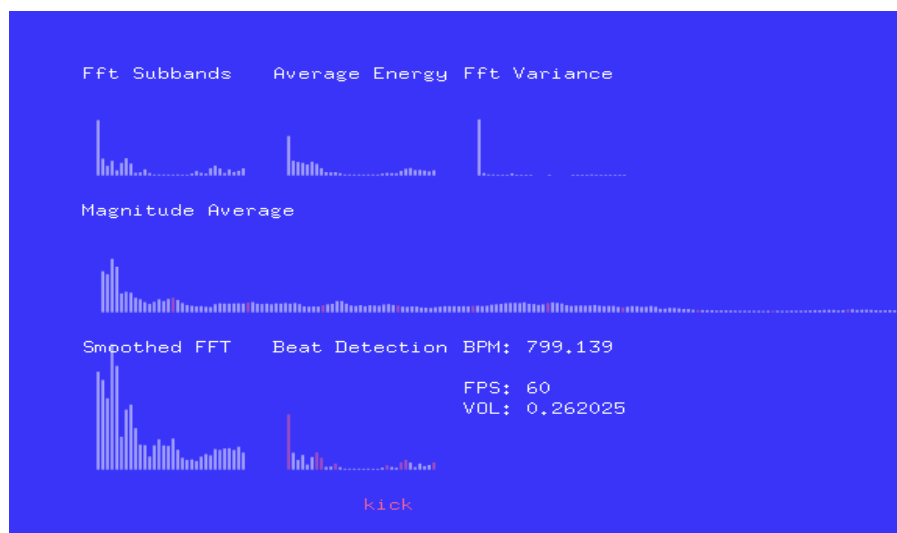
AiVDJ is comprised of four modes, able to be selected by four toggles located on the top. The app works without additional hardware, but to generate visuals for Audience and DJ modes, you'll need an Xbox Kinect.



Physics mode is setup to run by default, and reacts to music picked up from your laptop's speakers. You can enter in a word like "fresh" or "cool" into the "describe your set" box and the visuals will change accordingly.

Physics mode demonstrates two things: that the system is capable of working without any extra ports attached such as the kinect or arduino; and that the DJ can change the entire look of the system with simple key phrases.

You can toggle on and off the beat and kinect debuggers by pressing the "beat debug" and can toggle off the GUI by pressing the '~' key. The beat debugger breaks down the various variables from the sound visually, so you can get a feeling for how the system is working at a glance.



Video Mode

Video mode is designed to run without any user input. The video is sped up and slowed down to match the BPM of the song, opacity is determined by the volume of various frequencies, and the visuals clear on each drumbeat.

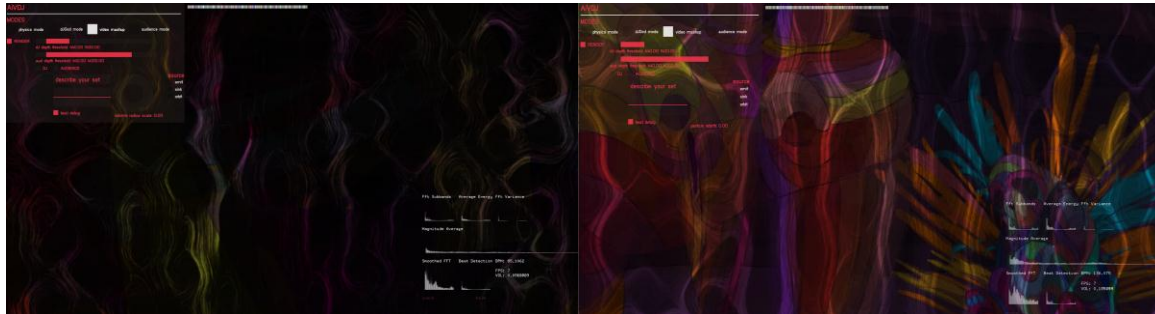
keys

'1'

toggles on and off the particle overlay, displaying the raw speed adjusted video

'2'

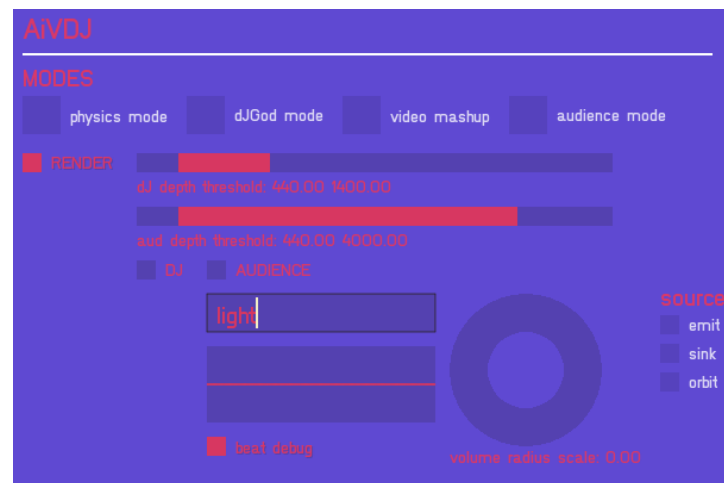
manual refresh if you are playing a song without very much bass



For the following modes make sure the kinect is properly plugged in and connected to your laptop. It also takes a moment for the kinect to initialize itself and for you to see the results on screen.

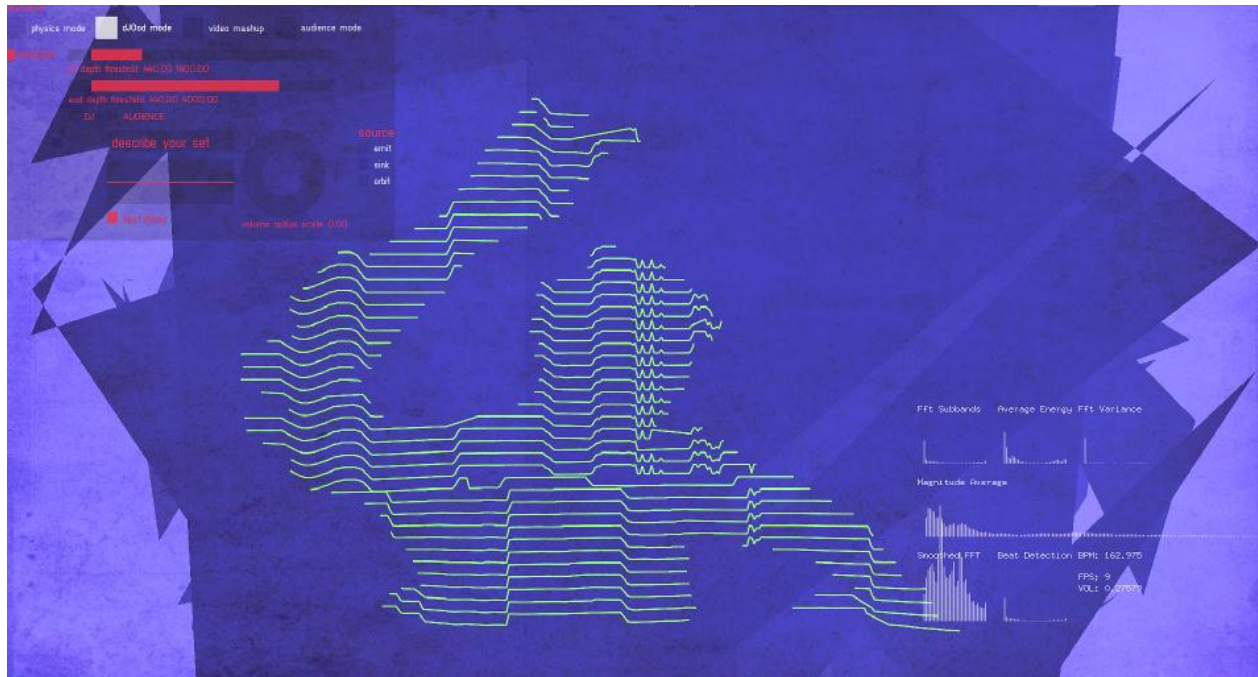
DJ Mode

Adjust the kinect to focus on you. (the menu will be in the top left corner, when not hidden by pressing the '~' key)



The kinect has a motor which can be used to tilt the kinect up or down. This can be triggered by pressing the up/down keys on your keyboard. There is a dj depth threshold slider on the menu, which is used to determine the distance from the DJ to the kinect. This is vital to ensure that things too close and background items will not be detected as part of the DJ. To see the raw kinect data, very useful for initial setup, click the square next to "DJ" directly under the two sliders.

Play music and enjoy.



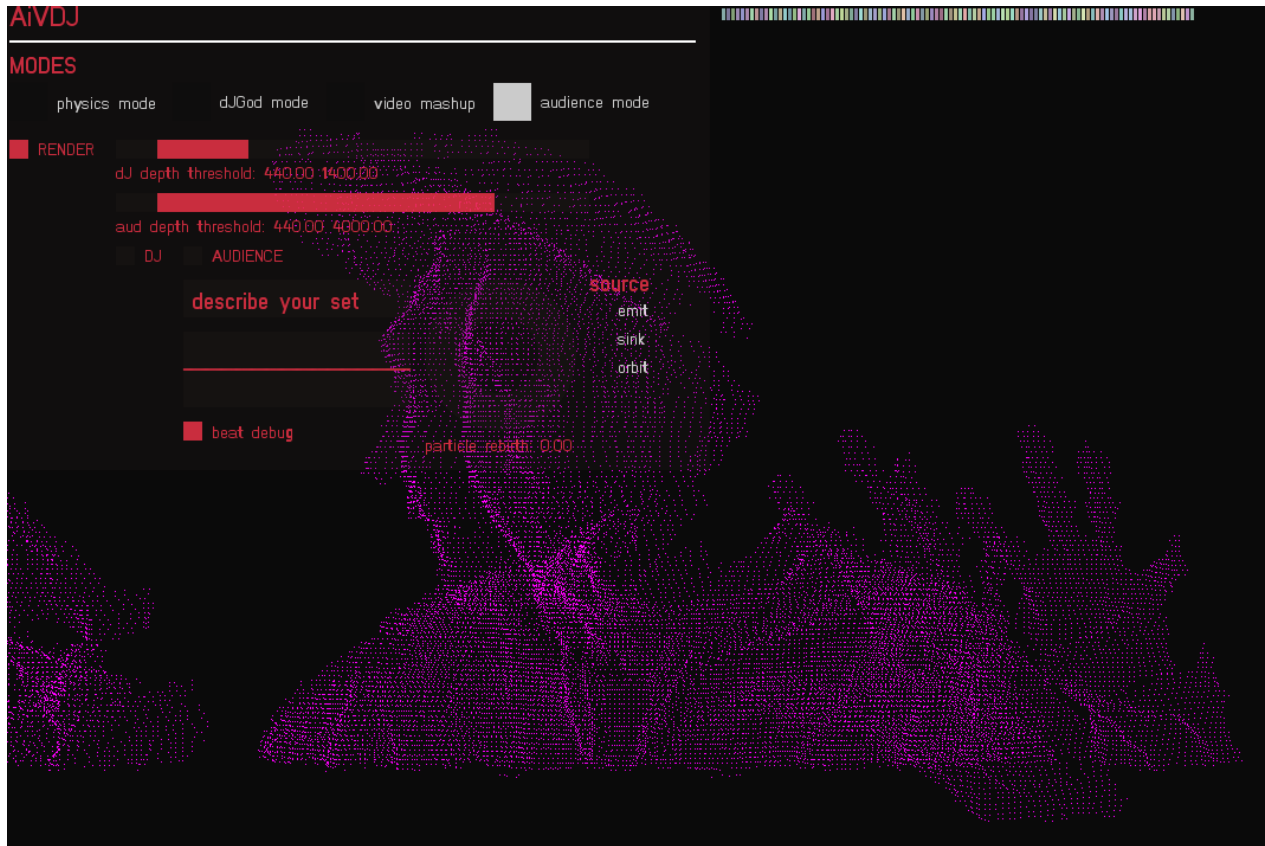
Typical ways to impress the crowd are dancing, fist pumping, or the simple but classic head bob.

You can change the color scheme the same way as in the physics mode, by typing in words like “fresh” or “cool” into the “describe your set” box, or at random by pressing the spacebar. As you will notice, the color of the DJ changes on every beat of the song that is playing.

The lines that make up the DJ projection are based off the changing volume of the song. The background is affected by the tempo, volume, and the distance of the DJ to the kinect.

This mode demonstrates intelligent switching. If the DJ is out of shot, the mode will immediately switch to physics mode.

Audience Mode

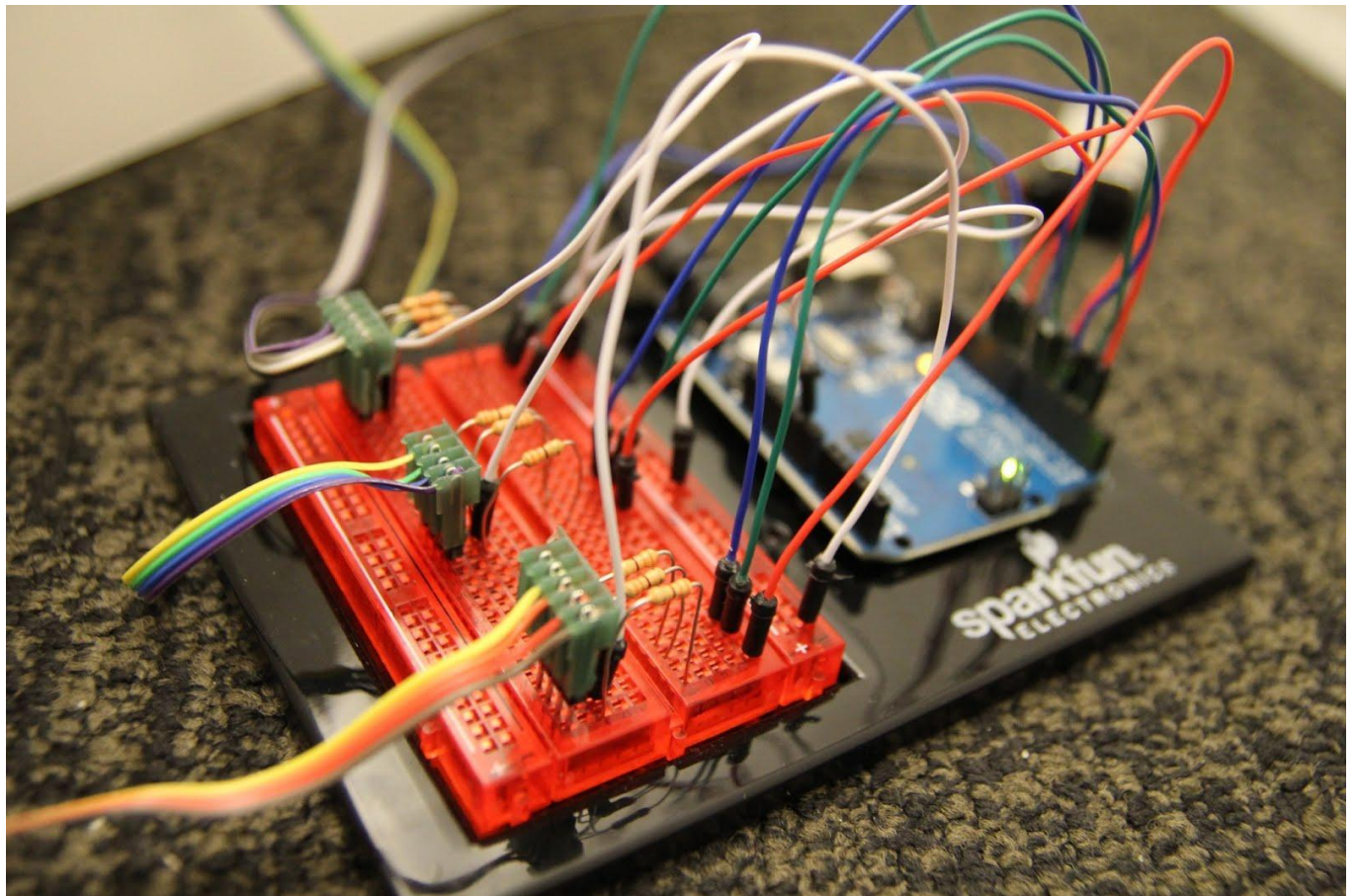


This mode will work without an arduino attached.

To attach an arduino you will need

- 1 Arduino
- 1 Breadboard
- 1 Arduino USB Power Cable
- 3 RGB LEDs
- 12 330 Ohm/0.25 W Resistors
- 14 Jumper Cables
- 3 30cm Flat Rainbow Jumper Cables (each with 4 wires)

to connect the arduino mimic the setup shown below



Connect the jumper cables to ports 2, 3, 4, 5, 6, 7, 9, 10, 11
Make sure the the jumper cables to power and ground are also connected.

Check SIK Guide for details on how to wire an arduino.

Run the arduino file called "LED.ino" (you will have to download the arduino platform from here <http://arduino.cc/en/Main/Software> to run the file) and ensure that the serial port used is the same as the one declared in audMode.cpp here:

```
/*-----arduino stuffs-----*/  
serial.enumerateDevices();  
serial.setup("/dev/tty.usbmodem411",9600);  
/*-----*/
```

This unfortunately has to be changed manually occasionally. It will either be
serial.setup("/dev/tty.usbmodem411",9600);
or
serial.setup("/dev/tty.usbmodem641",9600);

Then run Audience Mode and you should see your LEDs light up insync with the visuals on screen.