

# Audio/Visual Resampler

Built by Sam Smith, Nachiketa Gargi, Kaysen Chown, and Julian Cianciolo

## Introduction

This project was prompted off an 18 hours hake-  
thon with a focus on music and technology. We  
aimed to create a program that generated visuals  
off of a continous stream of audio from a perform-  
er. This idea involved into a fully immersive expere-  
ince where music was also resampled to create new  
compositions based off a performance. Previously,  
the practice of sampling was limited by control-  
lers such as MPCs or DAWs. Our project showed  
the possibility of real time sampling via an alter-  
native controller, an instrument. This controller  
allows for musical expression in a non traditional  
sense, where the notes played on the instrument  
(program controller) correlated to a response not  
typically possible through traditional instruments.

## Audio Processing

The program's interface allows the user to choose  
a musical piece to be resampled. After select-  
ing a piece, the program then uses a Fast Fourier  
Transform (FFT) on short chunks from the piece  
and stores the "footprint" in memory. Then, the  
user plays an instrument. In real time, the pro-  
gram performs an FFT on the audio signal and  
matches it to a footprint stored in memory. The  
program then plays that chunk on repeat until a  
better match is found. We prototyped this con-  
cept in python using PyAudio but quickly learned  
it would not be fast enough for real time process-  
ing. We then built it in JUCE (C++) to allow for  
a more friendly interface and faster processing.

## Visuals

The visual element to this program was built in Max/  
MSP. We used a [playlist] object to store many differ-  
ent short clips of video. Max then recieves a stream  
of audio from the output of the audio processing  
and uses a [snapshot~] object to measure the am-  
plitude of the signal. This amplitude is then used to  
quickly select between the videos within the playlist  
and also allows for modification such as play back  
speed and direction. To add one more layer of com-  
plexity we layered two of these video playback sys-  
tems together. This created complex visual patterns.

## Hakethon Success

At the end of the 18 hour "Music Makethon", we  
presented our project to all the attendees and a  
panel of 3 judges (all professionals in audio soft-  
ware/hardware). Our team of college freshman  
ended up winning first prize over upper-class-  
man and grad students in audio technology.  
We were extremmely encouraged by the results  
and feedback. While our project was novel, it  
proved that sampling and audio reactive visu-  
als were possible without having to spend hours  
editing and animating in a traditional sense.

## More Info

Featured in this EECS Department Article - <https://bit.ly/3f9aGae>

Source Code - <https://github.com/nacgarg/music-makeathon>