acapella

a free collaborative recording website that allows
ensemble members to create remotely

Areas: software and signal processing

Team D5 Jackson Bogomolny, Christy Lee, Ivy Ye

Use Case: Collaborative Recording

- Many groups are rehearsing remotely due to COVID
- Latency is a major issue; oftentimes video / audio falls out of sync
- Current applications that do this follow a 'one at a time' approach
  - Some also require expensive software and advanced knowledge of sound recording

Requirements

- Intuitive, easy to use
- CD quality recordings
  - 44.1 kHz, 16-bit, 2 channels (stereo) per person
- Click generator
  - Tempos from 25-200 bpm
  - Duple, triple, or quadruple meter
- Synchronization of all tracks exactly on beat
- Able to handle up to 100ms of latency
  - 100ms is just under a 16th note at 120 bpm
- Host at least 4 people recording or editing one project simultaneously
- Save the final work onto your local computer
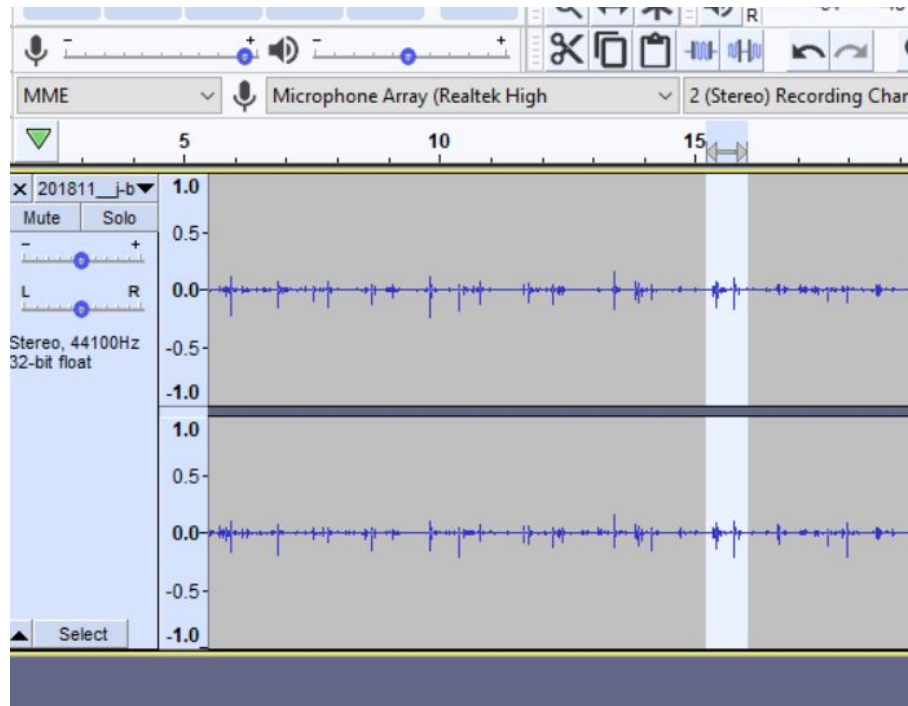
Technical Challenge: Latency

- Average / 'good' latency should be about 100 ms
  - About a 16th note at tempo = 120

- Delay will be calculated according to the clicks on a click track
  - Algorithm will then compare the click track time to the real time

- Ideally, ensemble members should hear each other, as if they are recording in the same room
  - Further research is needed-- syncing this way might not be technologically feasible

Technical Challenge: Uploading Audio

- Ideally, send small packets of audio in "real-time"
  - Possibly using websockets

- Otherwise, can send complete recorded audio after recording is complete
  - Using simple form file upload or AJAX
  - Drawbacks
    - No possibility of hearing other musicians as you record
    - Must upload large files at once

- Need to upload timing information along with the audio for synchronization
  - Which audio samples correspond to which "beat" on the click track

Technical Challenge: UI

- Must be intuitive to the all users
  - Usability is integral to our product
  - free alternative to expensive and professional software

- Audio visualization
  - Web Audio API
  - Users must be able to view the waveform in order to edit

Final Product: Back End

- Website built using Django to organize code
- AWS hosting
- Audio processing using Web Audio API and sockets, and on the server using Python DSP libraries

Final Product: Front End

- Clean, intuitive, easy-to-use UI
- Support for various audio formats
- Accurate display of the audio using different types of visualizations (Spectrogram, Waveform)
- Tools for manually adjusting volume and pitch, etc
- Fault tolerant website against security attack

Testing

- deploy the website, using AWS
- test performance & delay of the website on varying number of users
  - Limit maximum number of people in a group
  - Limit maximum number of groups that can be formed at the same time
- try breaking into security of the website
  - No crash against malicious HTTP request and cross-site request forgery
- get qualitative feedback from groups who use it
  - Is the website intuitive enough?
  - suggestion for audio editing functions?

Division of labor

Jackson	Ivy	Christy
- Audio processing - Effects - Uploading - Django views / models / urls	- Recording / performance synchronization - Editing tools - Click track generator	- Website UI - Cloud deployment - Audio visualization

Tasks	Progress	Start	Due	Start on Day*	Duration*
wordpress set up	0%	2/18/2021	2/22/2021	0	4
work on project proposal	0%	2/18/2021	2/21/2021	0	3
set up github	0%	2/18/2021	2/22/2021	0	4
----- Research -----					
figure out way to sync recording	0%	2/22/2021	2/27/2021	4	5
figure out way to sync individual members	0%	2/22/2021	2/27/2021	4	5
	0%	2/22/2021	2/27/2021	4	5
	0%	2/22/2021	2/27/2021	4	5
----- website creation -----					
Implement Group formation & Registration feature	0%	2/27/2021	3/6/2021	9	7
upload your own audio file	0%	2/27/2021	3/6/2021	9	7
audio visualizer	0%	3/6/2021	3/13/2021	16	7
make syncing algorithm	0%	2/27/2021	3/6/2021	9	7
sync performers in real time?	0%	3/6/2021	3/12/2021	16	6

create adjustable metronome	0%	2/27/2021	3/6/2021	9	7
record according to metronome	0%	3/6/2021	3/13/2021	16	7
slack time	0%	3/12/2021	3/20/2021	22	8
----- testing -----					
deploy on AWS & test performance on varying number of users	0%	3/13/2021	3/20/2021	23	7
try breaking into security of the website & resolve security issue	0%	3/13/2021	3/20/2021	23	7
get qualitative feedback from groups who use it	0%	3/13/2021	3/20/2021	23	7
----- additional website features -----					
implement Chatting	0%	3/6/2021	3/13/2021	16	7
implement sound editing function (volume, pitch)	0%	3/6/2021	3/13/2021	16	7
implement panning	0%	3/6/2021	3/13/2021	16	7