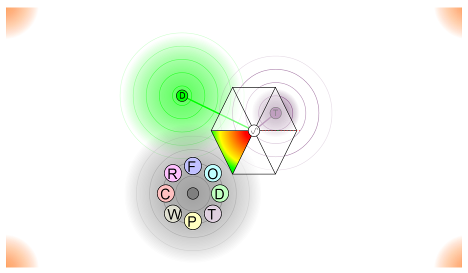
Final Documentation

Web-audio - Visual Mixer



Short description of the application

Our application is a instrument with dynamic audio generation with effects that can be added and adjusted on fly. The application works on destop with mouse or via touch screen (i.e. the AaltoWindow) . All the instruments and sound effects are represented with 2D graphical representations to allow easy adjustment and interaction. This naturally comprises two elements of HTML5: WebAudio and Canvas/2D graphics.

UI interaction was decined to fit the limitations of the hardware to be used (the AaltoWindow touchscreen and its resolution of the detected gestures had to be taken into account).

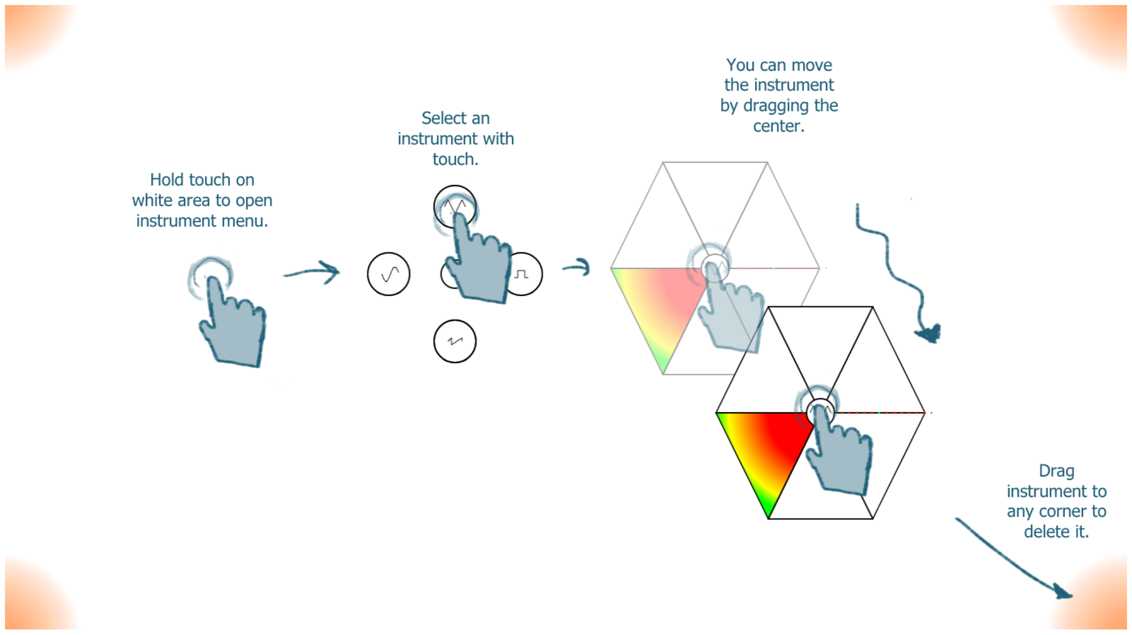
The final scope of the project was accomodated to fit the course scope in terms of workload. The interaction isn't as fluid as we would have liked it to be and there are still some bugs with the multi-touch interface and the added effects aren't that impressive, but we got the app working and all the planned functionalities done.

Functionalities

Creating and deleting instruments

Instruments are used to play sounds in the app. The app has 4 different kinds of instruments: SINE, SQUARE, TRIANGLE and SAWTOOTH. Each instrument represent basic sound created by an oscillator.

Instructions for creating instruments in the image below..

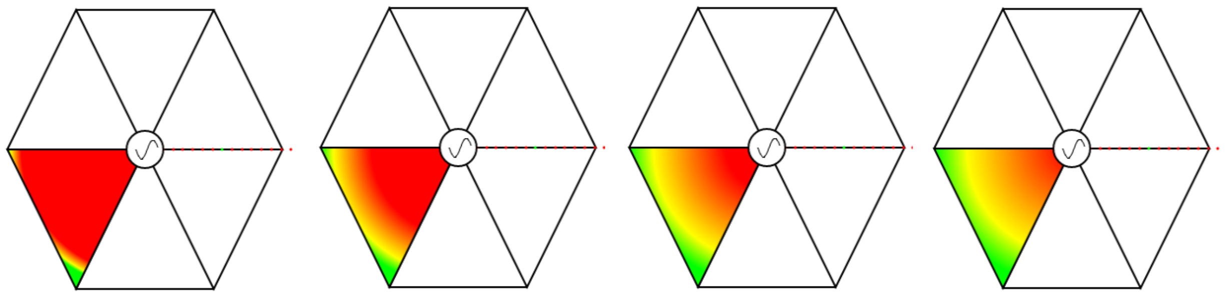

Hold touch on white area to open instrument menu.
Select an instrument with touch.
You can move the instrument by dragging the center.
Drag instrument to any corner to delete it.

Playing sounds and adjusting volume

Instrument are used to play sounds in the app. Touching white area inside the instrument creates a sound. Hold will produce longer sounds and taps short. The pitch of the sound is based on the distance of the center of the instrument - close to center low and far from the center high. Instrument basic sound depends from the sound type (SINE, SQUARE, TRIANGLE and SAWTOOTH).

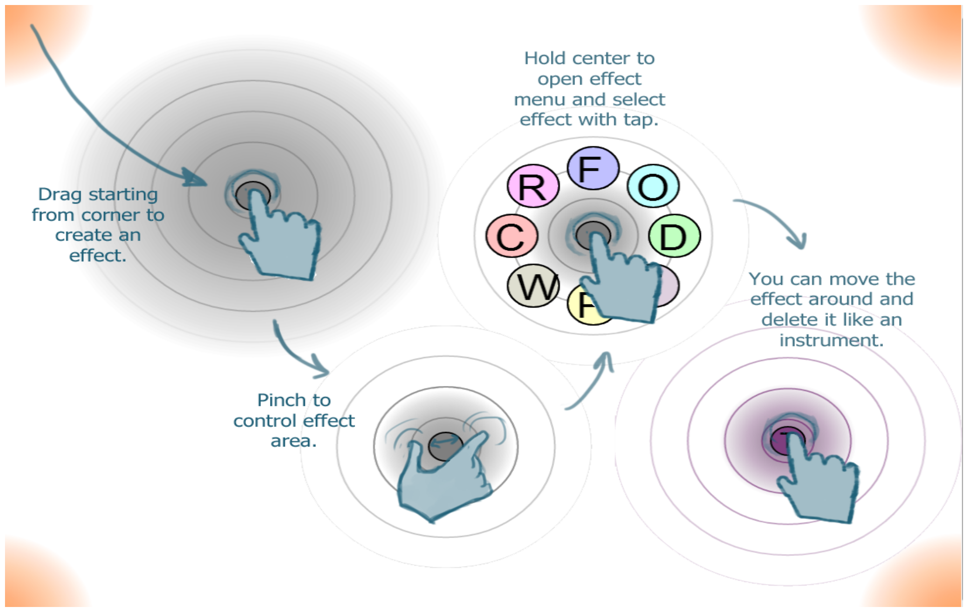
The volume of the instrument can be adjusted by tapping to the colored area of the instrument. Tapping close the center will decrease the volume. Tapping far from the center increases the volume. Current volume of the instrument is represented in the instrument with colors - more red means more volume. Volume affects the gain node of the instrument oscillator.

Because the instrument play area is limited, the app should in theory be playable by multiple users at once using the same insrument or different instruments at the same time.



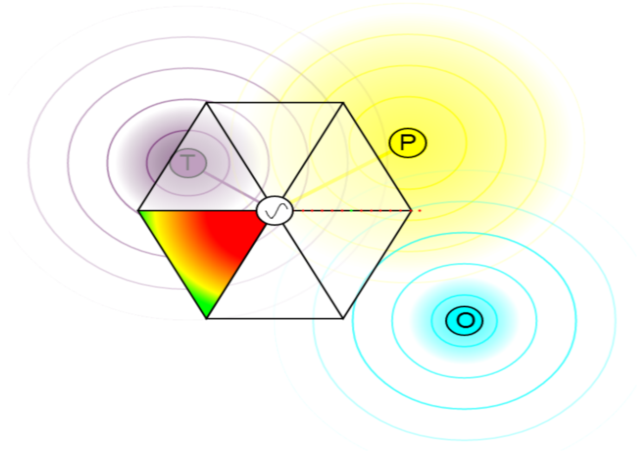
Creating, selecting and deletig effect

Effects can modulate the sound generated by the instrument oscillators. New effect can be added to the app by dragging it from corner to white app area. The effect can be selected by holding the center of an effect to open the effect select menu and tapping the wanted effect. Effect can be switched to any other effect same way.


Drag starting from corner to create an effect.
Hold center to open effect menu and select effect with tap.
R
F
O
C
D
W
P
You can move the effect around and delete it like an instrument.
Pinch to control effect area.

Adjusting effect

Effect will affect the sound based on effect size and instrument proximity. The closer and the bigger the effect is the more it will affect the sound. If multiple effects are on the instrment area, all of them will affect the sound.



Group Members, effort spent and responsibilities

Name	Student number	Effort amount	Implementation responsibilities	Other responsibilities
Miquel Puig i Pey	???	high (~50h)	• Initializing the project and the git repository • Instrument and effect menu • Oscillators and effects • Multitouch	• Project management
Arthur Hamon	415019	average (~20h)	• Tuna effects • Research and experimentations with the Web Audio API	• Final presentation preparation
Rik van der Laan	???	average (?)	• Creating interaction demo with flash.	• Preparing presentations • Researching about audio and different possible effects
Minna Turunen	77299J	average (~20h)	• Drag and drop corners • Keyboard • Connecting sound generation to keyboard • Keyboard volume area • Volume control • Few tuna effects	• Researching about processing and possible JS-libraries for presentation plan • Final document and project submission • Screencapture

Installation and executing instructions

Application can run on server or local server (such as Wamp for windows). Unzip the app and run it from “**VisualMixer.html**”-file. No other installations shouldn’t be required.

The app works in google Chrome. Web-audio API should work also on latest Firefox, Opera, Safari and some mobile applications, but the application hasn’t been tested on those platforms.

Note that the app needs to be running on localhost or otherwise the Chrome will complain about file access issues.

Used platform, frameworks, and libraries

We wanted the app to be as much standalone webpage using **HTML5-CSS3-JS** as possible. We used **Processing.js** framework to develop the application.

Since Processing.js didn’t support touch interaction or audio, we used **hammer.js** (multitouch library), **hammer.hakemultitouch.js**, and **tuna.js** (web-audio effect library) to make developlement easier. Most of the audio handling we could implement with basic web audio API.

To help handling events and dom elements **jquery** was also used.

We run into some problems, because the hammer.js didn’t support more events for more than one hand interaction. We tried to fix this by implementing our own touch interface, but the implementation is still buggy. The app works best at the moment as single user app.

Brief use case description

1. User opens the app and creates a new instrument (SINE).
2. User adds new empty effect, opens up the effect menu and sets an effect (T= TREMOLO).
3. User drags the effect to instrument area.
4. User adjusts the volume of the instrument and plays the instrument.
5. User adds more sound effects and tests how they affect the sound of the instrument.
6. User is done playing and deletes the instrument and the sound effect and closes the app.

Source code

Attached as a zip.

Video

Video of app in action and possible interaction flash demo in video’s folder.