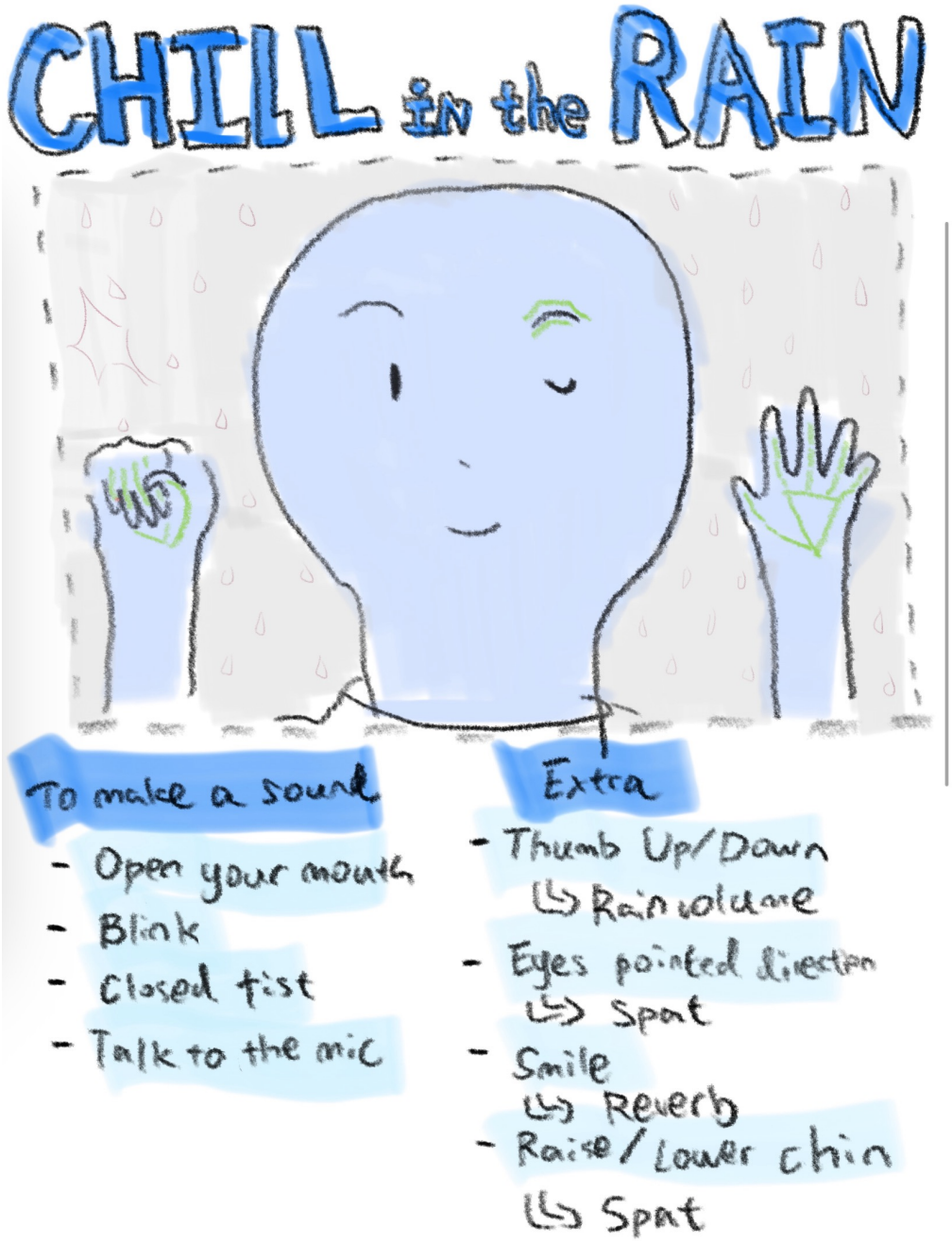
Interactive music making with facial and hand-gesture recognition——

Chill in the Rain

Ganling Zhou



*This photo is the user manual presented to the audience.

> Il n'y a de réalité que dans l’action.
>
> — **Jean-Paul Sartre, *Existentialism is a Humanism***

Introduction

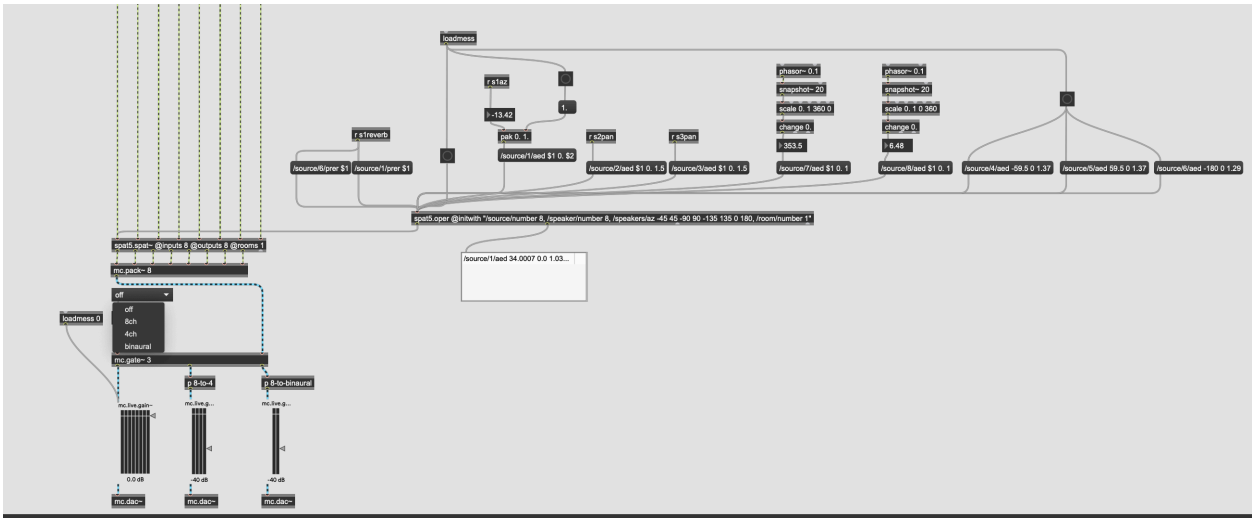
Running on Max, also technically supported by Google MediaPipe and Spat5, ***Chill in the Rain*** is an interactive sound game for therapeutic purposes.

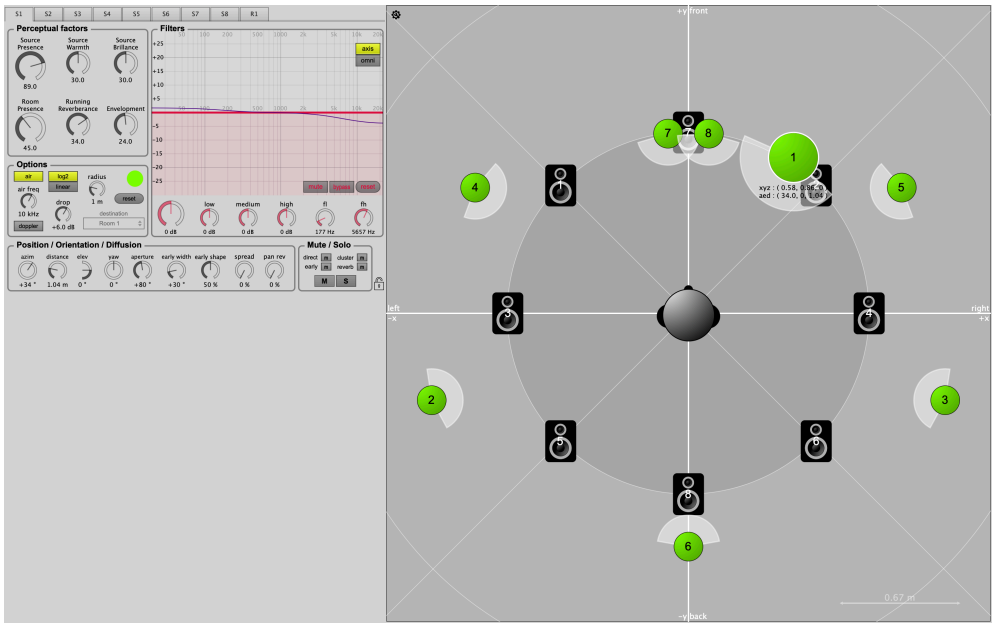
Surrounding by a rain ambient background, the players may enjoy triggering sounds with they hand gestures and facial expressions, listen to their own sounds made with extensive echos, and these sounds are all designed to spatialized according no matter if you are in a setting with two, four, or even eight speakers.

Spatialization

Spatialization of this project is powered by **Spat5**. There are 8 sources in total, each of them are mono.

These 8 sources are treated as a 8 inputs into a spat space with one room and 8 speakers. There are two reasons why we want 8 speakers: a)there are 8 speakers in Studio D at IRCAM(cough cough), b)it is always easier to convert more speakers to less speakers instead of the other way around.





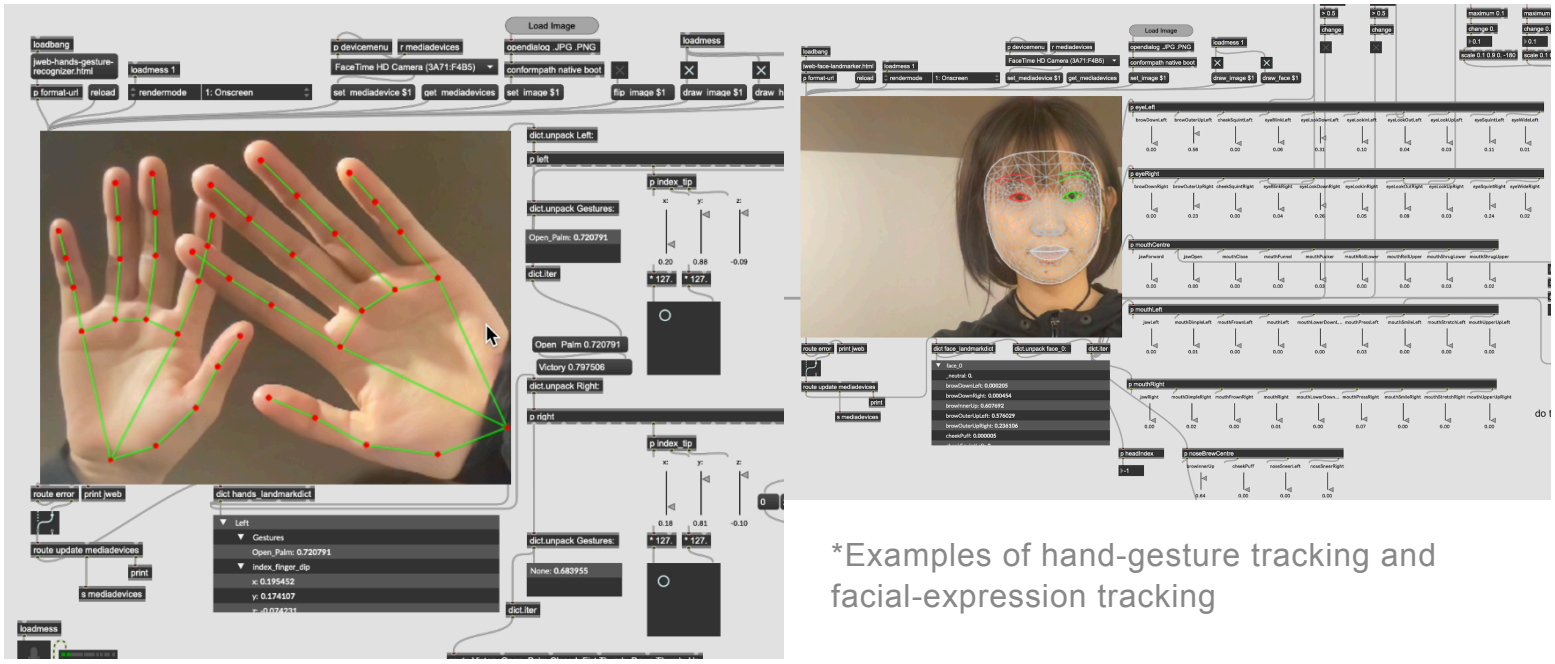
*The 8 sources are: 1) fist_synth, 2) rain_left, 3) rain_right, 4) eye_left, 5) eye_right, 6) mouth_synth, 7) echo_left, 8) echo_right.

Motion Trackings

Motion trackings of this project, including hand-gesture trackings and facial-expression trackings, are all powered by **Google MediaPipe**.

Written in JavaScript, MediaPipe is a public-resource framework that can motivate your devices into on-device machine learning. In the specific case of this project and in simpler words, MediaPipe can teach a computer in recognizing certain human movements in front of a camera. The only problem is that, you do not have an easily accessed camera in Max. However, GitHub contributors find the way: they wrote html for the computer to read real-time cameras and CSS to make it a proper format embedding in Max. Brilliant works. Really brilliant.

It is hard to tell who came out with this solution first, because each contributors recognize some other contributors for doing this first and I failed to find the origin, but my project is mostly based on lysdexic-audio's works: https://github.com/lysdexic-audio/jweb-hands-gesture-recognizer and https://github.com/lysdexic-audio/jweb-face-landmarker.



*Examples of hand-gesture tracking and facial-expression tracking

Sound Designs

Let's go over each sounds. Let's go over what they are, how they are being triggered, and how they are spatialized.

Source	Sounds	What are they?	How they are triggered?	How they are spatialized?	Special mentions
1	fist_syn	ez-synth built in Max	Each time it is detected that there is a close fist(either hand), it generates a note in D major pentatonic with a random tempo. A hand is an octave above the other one.	The direction of this sound will move according to where your eyes are currently looking at.	The higher-pitch one is more likely to have a lower tempo then the lower-pitch one. They will get a reverb when you smile.
2	rain_L	a stereo rain sound that I found in ADD library and then edited into a loop	They should be playing when you open the Max project.	When you lift up your chin, the rain sounds go to the back. Lowering your chin for having them on the front.	When a thumb-up movement is detected, the volume of the rain will turn up. Thumb-down for lowering the volume.
3	rain_R				
4	eye_L	One sound I designed with Reason, but I made them in two pitches	When a left-eye blink is detected.	On the far left.	With this trigger, I discovered that I am not capable of blinking one eye at a time…
5	eye_R		When a right-eye blink is detected.	On the far right.	
6	mouth_syn	Sound designed with Reason	When an opened mouth is detected.	On the back.	It will also get a reverb when you smile.
7	echo_L	Slap echos from sounds caught by a connected mic or the laptop mic	When any sound is caught.	Moving counterclockwise.	Promise me you are not using the mic and the speakers side by side. They should be moving symmetrically.
8	echo_R			Moving clockwise.	

An oversimplified explain in the long process in making triggers do what they should do

Long story short, I make Max do certain things using certain datas or certain analysis provided by MediaPipe.

MediaPipe provides a numerical value range from 0 to 1 to show the possibility(in the statistical wise) of a motion being done. For example, when you blink you left eye in front of the camera at the moment, MediaPipe may judge that you have a 0.86 chance of blinking your left eye at the moment. There are a lot of motions that on the list that we may use: the odds that you have jawForward, browDownRight, mouthDimpleLeft…just to make a few examples. With the and gestures, it can also tell you what the possibility for each hand being a fist, thumb-up, thumb-down, victory, and open palm. Please see screenshots on page 4 of this document or go to the patch for more details.

I send these datas directly to Max: mouthSmileLeft is scaled to 45 to 120 and then sent to “Room Presence” in source 1 and 6(fist_synth and mouth_synth) and creates a bigger reverb when the smile is bigger. For another example, when the MediaPipe sees a possibility bigger than 0.5, it bangs and plays the prepared recording.

These two examples have an essential difference: the mouthSmileLeft should send datas continuously so the player will not be tortured by weird reverbs. But you also cannot trigger a blink synth too frequently or it will give you multiple times of it played with just one blink. Thus, with the smile, you only need a clip object and a slide object to makes it smoother, but with the blink, you want to have the change object to make sure it does not send bangs to your audio player too frequently.

Thus, we actually have two ways that Max can potentially receive the datas from MediaPipe and we make sure we distinguish them: eye-looking direction for panning, continuous; hand gesture detections, one at a time; etc..

Concerns

MediaPipe is such a reliable and fast-reacting machine-learning model. The datas it provides are extremely reliable. It is also not too CPU consuming based on personal experiences. However, the fun part is, embedding a camera into Max with HTML and CSS can crash your CPU. Moreover, spat5 can crash your CPU to another extend.

Besides, MediaPipe can still be insufficient in few details: a)it seldom tells the difference between the left hand and the right hand, b)I did not find any information anywhere on either it can analyze multiple people's faces or more than two hands at the same time.

Theory and Vision

I built this project to emphasize the power of subtle movements, so ones may appreciate themselves better when they are aware that even their blinks or closed fists can lead to something beautiful. As Jean-Paul Sartre once writes, *il n'y a de réalité que dans l'action*, there is no reality except in action, but, also, there will be a reality even out of the smallest action.

It is my way to solve my existential crisis and I sincerely hope that it will help our with yours, too.