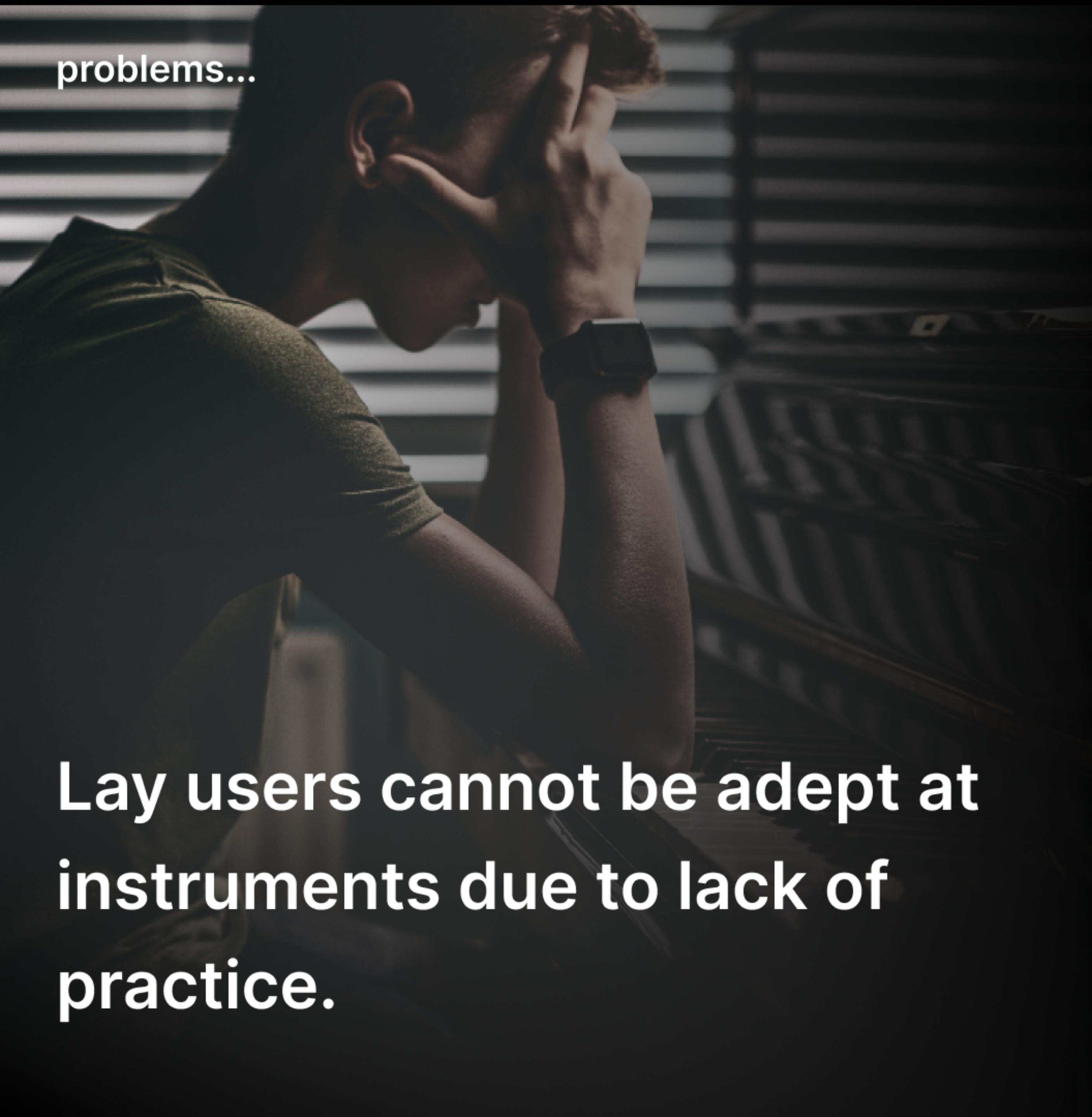


Dbox

a new playful wearable device to incorporate dance, music, and light



Dbox is a playful instrument to integrate dance, music and light simultaneously. It addressed the issue that lay people cannot express the music through their understanding and practice on music, rhythm and beat.

A dark, moody photograph of a person from the side, wearing a striped shirt and a black wristband. They are seated at a keyboard, with their right hand resting near their face in a contemplative or thoughtful pose. The background is blurred, showing horizontal lines of what might be window blinds.

problems...

Steep learning 01. curve

Music production requires long-term learning.

Limited creative 02. freedom

Strict music theory and performance norms confine creative boundaries.

Lay users cannot be adept at instruments due to lack of practice.

Disconnected 03. natural embodiment

musical expressions are deliberate but not natural.



“My body disagreed with the rhythm”

When Bodies Rebel Against Beat

When people follow the rhythm, **their involuntary movement might differ from how a practitioner performs**: they lack the knowledge of pairing the muscle, making a good pose, connect the posture across the beats, which results in “physical dissonance”.

“My movement disagreed with my mind”

When Bodies Rebel Against Fantasies



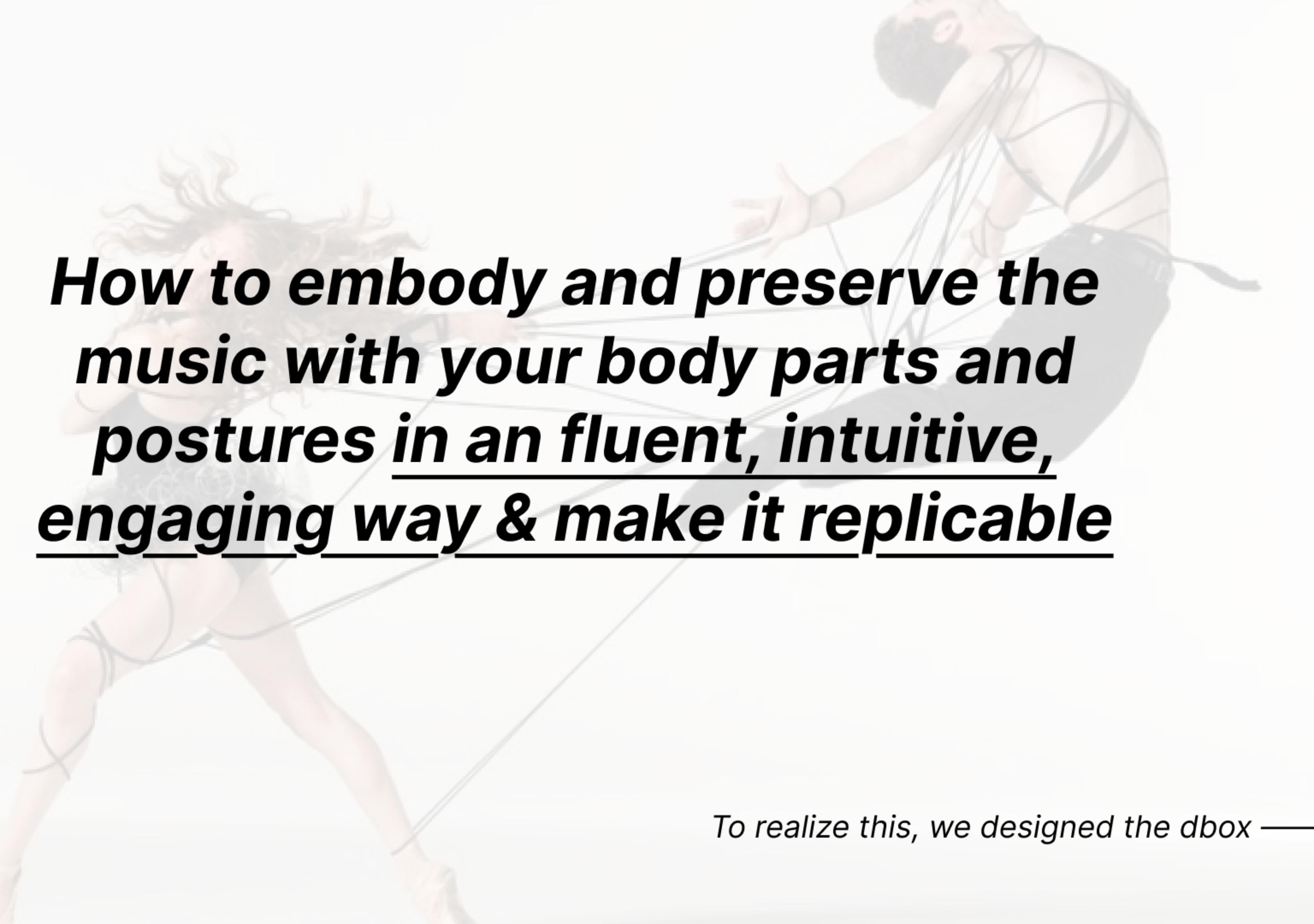
Expectation

V/S



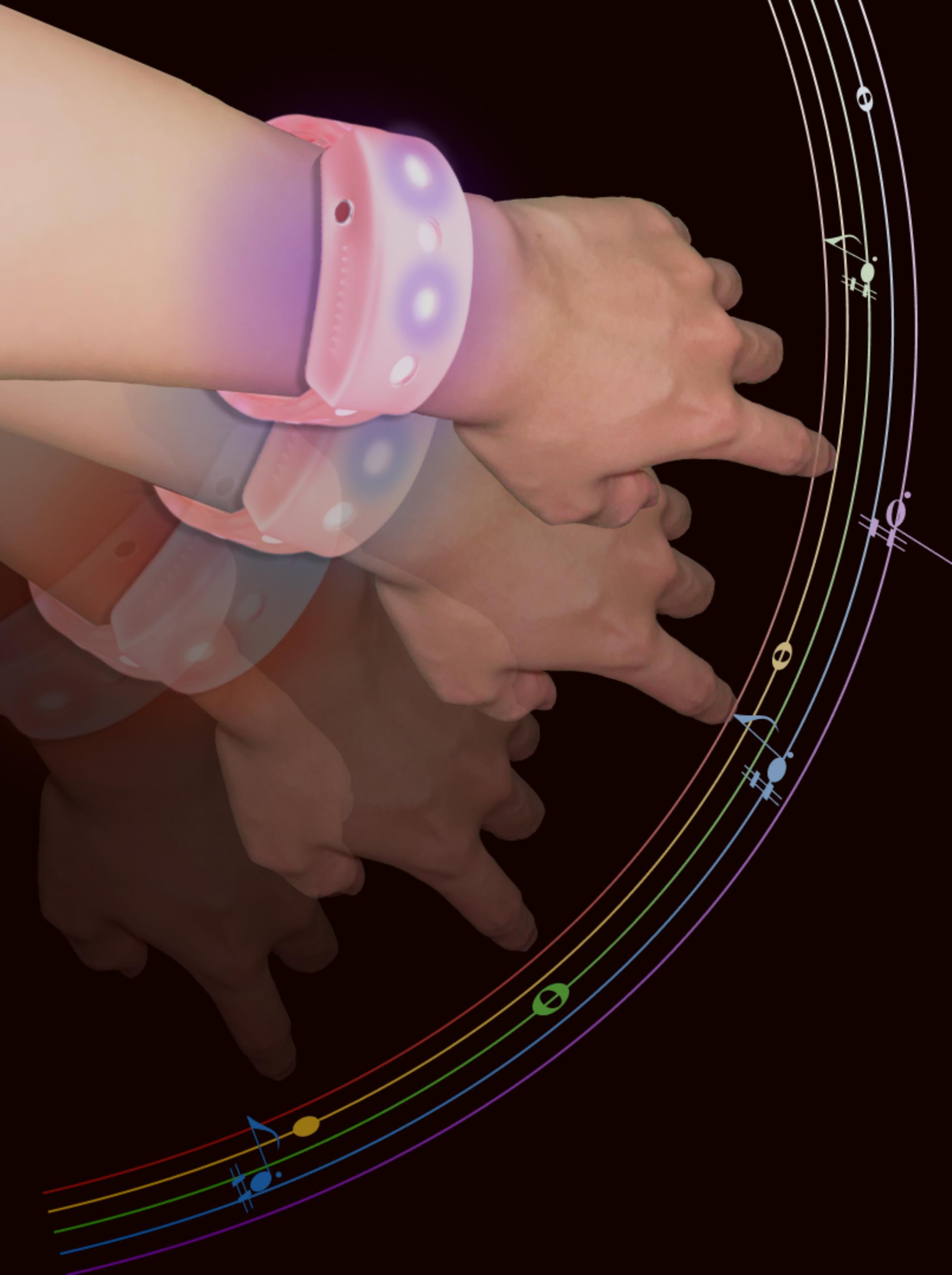
Reality

Ironically, these people cannot be creative or even playful - laypeople can **neither perform creatively as gorgeously as they imagine, nor can they be sparkling with decorative**, performative media to improve what they dance.



How to embody and preserve the music with your body parts and postures in an fluent, intuitive, engaging way & make it replicable

To realize this, we designed the dbox —



*Body as “Orchestra”, **Dbox embodies** **the music with the beat***

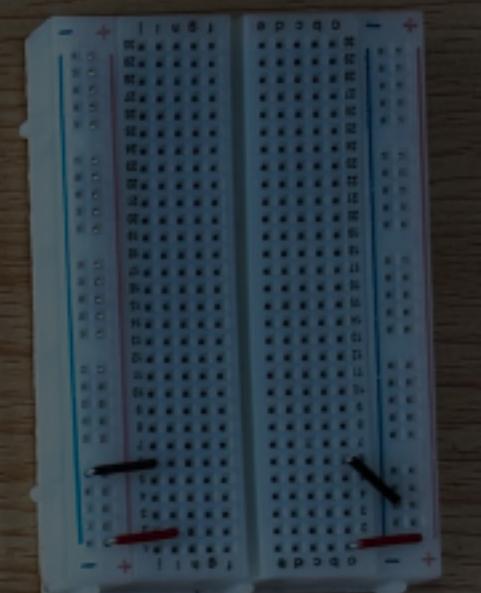
Enabling users to "play" their limbs like musical instruments

Through motion-sound-sight mapping, their body movement is naturally transformed into performative music and visual light art, providing any different types of users the same playing experiences to produce whatever they want.

USB Cable

Transfer data and codes between the computer and device

Breadboard



used for fixation of other materials

M-M Dupont wires

Materials...

WS2812B Programming RGB led

Play the lighting
Fixing device on hand

Insert instrument audio
USB

Wires

FlexiForce A401

Open the device and control some function

Mini MP3 Player

Micro SD

Restore info



Play the beat
(built-in 9 axis sensors)

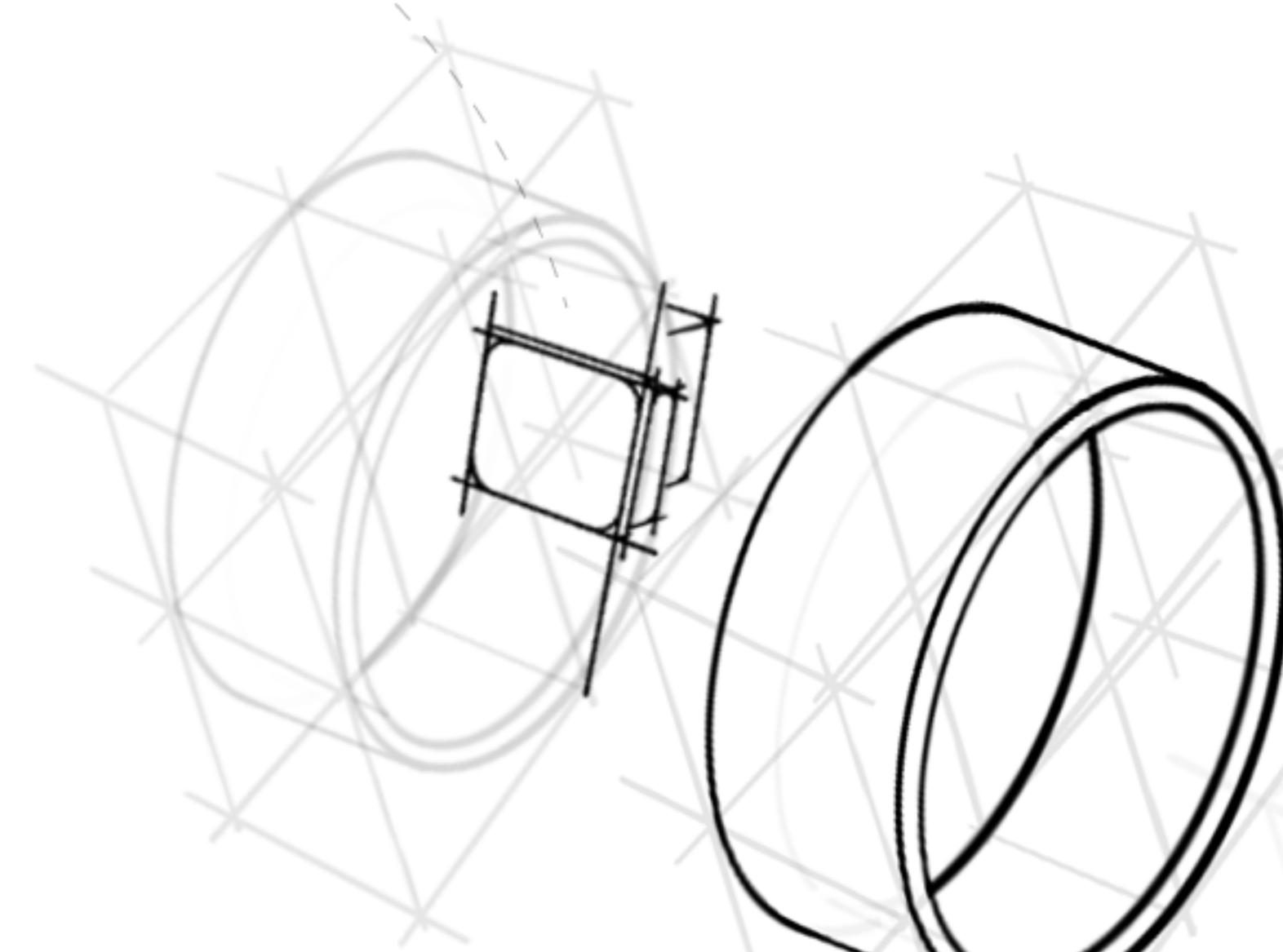
Pui Audio w250



small size, easy to carry on

clip

flexible for every person's wrist size



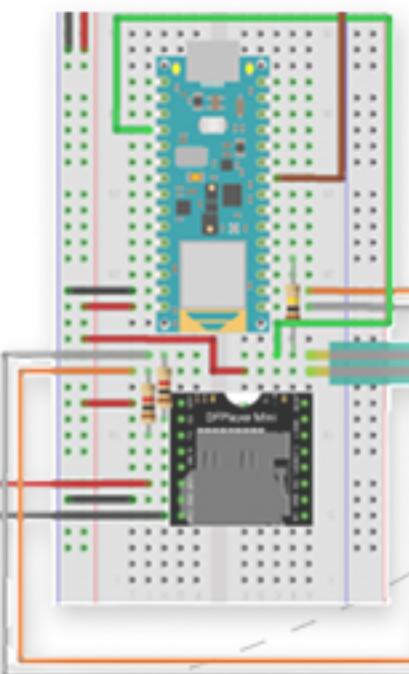
silicon covered Led String

WS2812B Programming RGB led



central analysis district

**Audio
speaker**



Cpu+sensor

**Pressure
sensor**

silica made

easy to be worn

circular structure

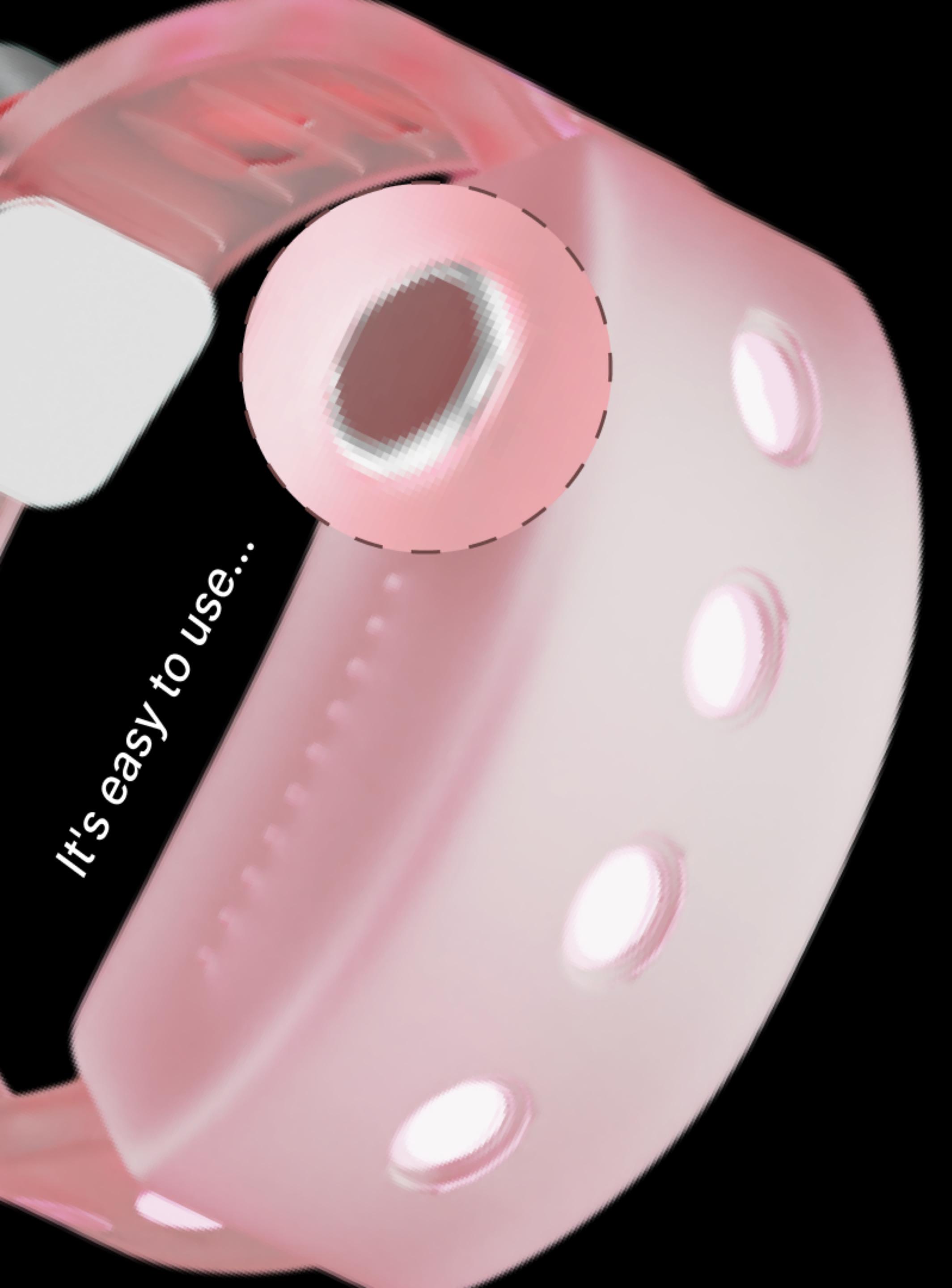




Wear Anywhere

The background features a woman with long brown hair dancing joyfully, with motion blur effects around her arms and legs. The scene is set against a dark, moody background with glowing blue and purple light rays emanating from the center.

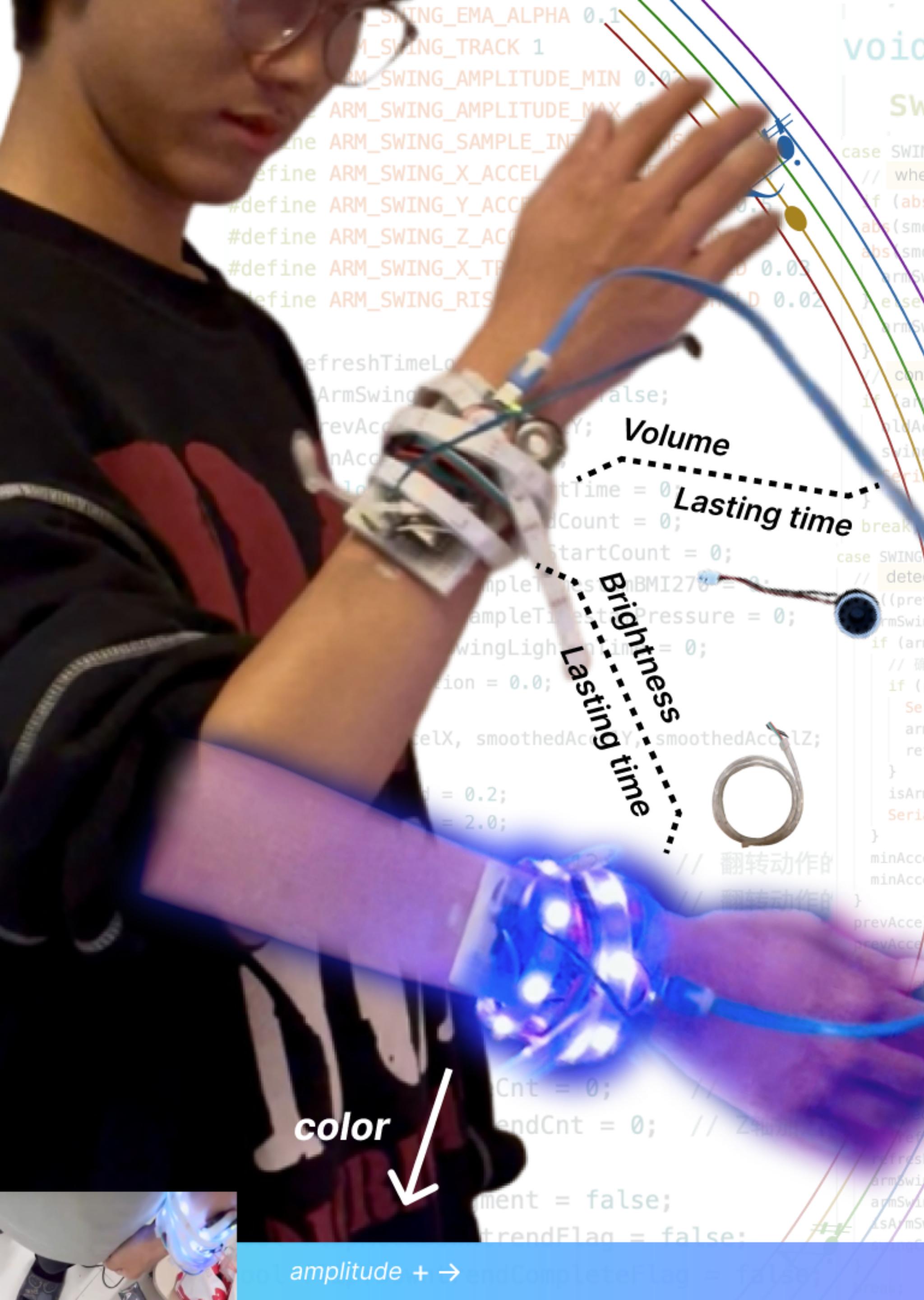
Arms, legs, waist
sensors can match sounds to all body part.



One button No clutter

Just one touch button for opening the device and changing modes.
No confusing controls.

Real-time Synthesis Translation



```
void detectArmSwing() {
```

```
    switch (armCurrentState) {
```

```
        case SWING_IDLE_STATE:
```

```
            // whether the limb is static
```

```
            if (abs(smoothedAccelX) > ARM_SWING_X_ACCEL_MIN_THRESHOLD &&
```

```
                abs(smoothedAccelY) < ARM_SWING_Y_ACCEL_MAX_THRESHOLD &&
```

```
                abs(smoothedAccelZ) < ARM_SWING_Z_ACCEL_MAX_THRESHOLD) {
```

```
                    armSwingWaveStartCount++; // 增加垂直状态计数
```

```
                } else {
```

```
                    armSwingWaveStartCount = 0; // 重置计数
```

```
                }
```

```
                // confirmed vertical
```

```
                if (armSwingWaveStartCount == 50) {
```

```
                    oldAccX = abs(smoothedAccelX); // 记录初始X轴加速度
```

```
                    armCurrentState = SWING_VERTICAL_COMPLETED; // 切换到垂直完成状态
```

```
                    Serial.println("Hand vertical ready");
```

```
                }
```

```
            case SWING_VERTICAL_COMPLETED:
```

```
                // detecting the hand motion
```

```
                if ((prevAccelX - abs(smoothedAccelX)) > ARM_SWING_X_TREND_DELTA_THRESHOLD) {
```

```
                    armSwingTrendCount++; // 增加趋势计数
```

```
                if (armSwingTrendCount >= 3) {
```

```
                    // confirm hand action start
```

```
                    if (refreshTimeLock) {
```

```
                        Serial.println("Record time start");
```

```
                        armSwingStartTime = millis(); // 记录挥手开始时间
```

```
                        refreshTimeLock = false; // 锁定时间记录
```

```
                    }
```

```
                    isArmSwingRisingPhase = true; // 标记进入上升阶段
```

```
                    Serial.println("Waving~");
```

```
                }
```

```
                minAccelX = prevAccelX; // 记录最小X轴加速度
```

```
                minAccelY = prevAccelY; // 记录最小Y轴加速度
```

```
            }
```

```
            prevAccelX = abs(smoothedAccelX); // 更新前一次X轴加速度
```

```
            prevAccelY = abs(smoothedAccelY); // 更新前一次Y轴加速度
```

```
        } // when the motion ends
```

```
        if (!isArmSwingRisingPhase && (abs(smoothedAccelX) > minAccelX)
```

```
            && (abs(smoothedAccelY) - minAccelY) >= ARM_SWING_RISING_Y_DELTA_THRESHOLD) {
```

```
            armCurrentState = SWING_HAND_COMPLETED; // 切换到挥手完成状态
```

```
        }
```

```
    } // about detection (no motion in 2 secs)
```

```
    if (refreshTimeLock && (millis() - armSwingStartTime >= 2000)) {
```

```
        refreshTimeLock = true;
```

```
        armSwingWaveStartCount = 0;
```

```
        armSwingTrendCount = 0;
```

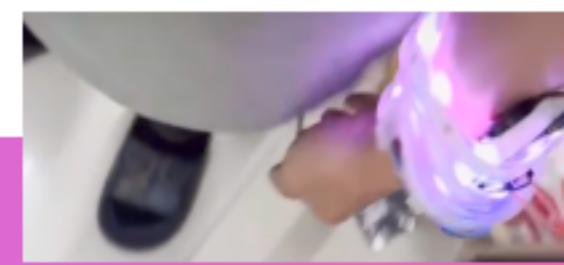
```
        isArmSwingRisingPhase = false;
```

```
        Serial.println("SWING_IDLE STATE");
```

```
    }
```

When you move, Dbox hit the beat as the limb waves fast and slow, left to right.

When the users wave their limbs, the cpu (arduino nano 33) will collect the value of acceleration and displacement collected from the body movement and translate them into the speaker's volume, the led's brightness, and their lasting time as well through codes.



Smart Force Adapting

When you move, Dbox learns your kinetic signature, scales your effort to 100% manifestation.

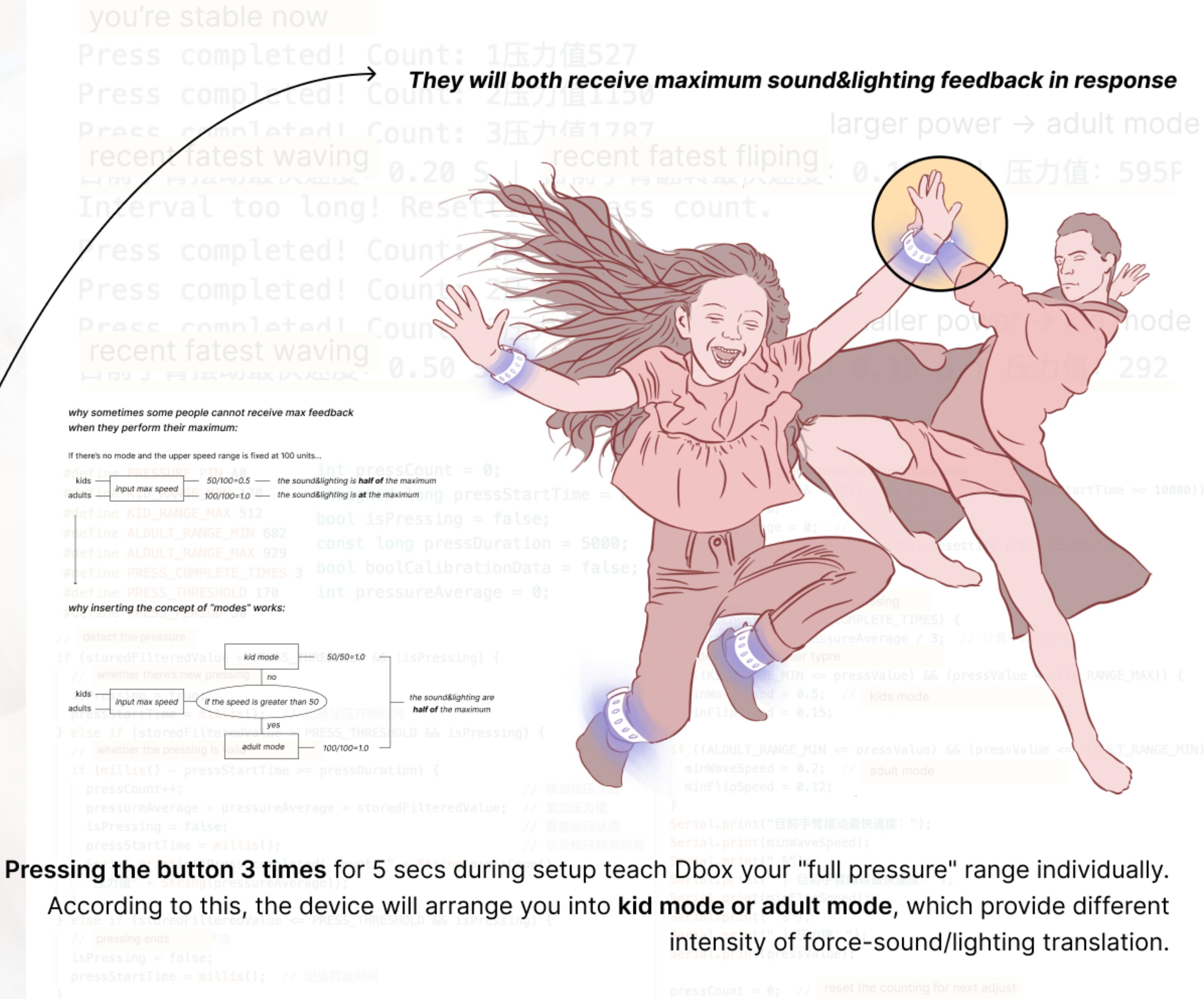


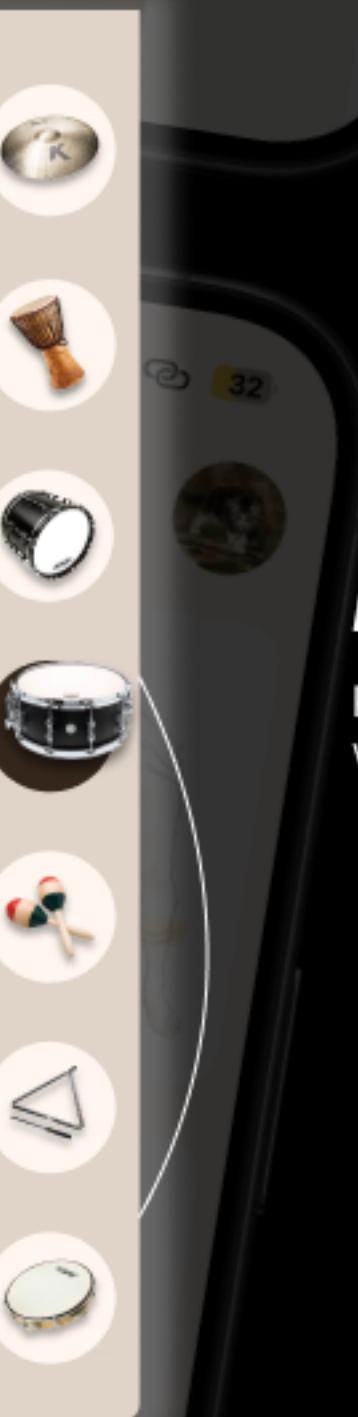
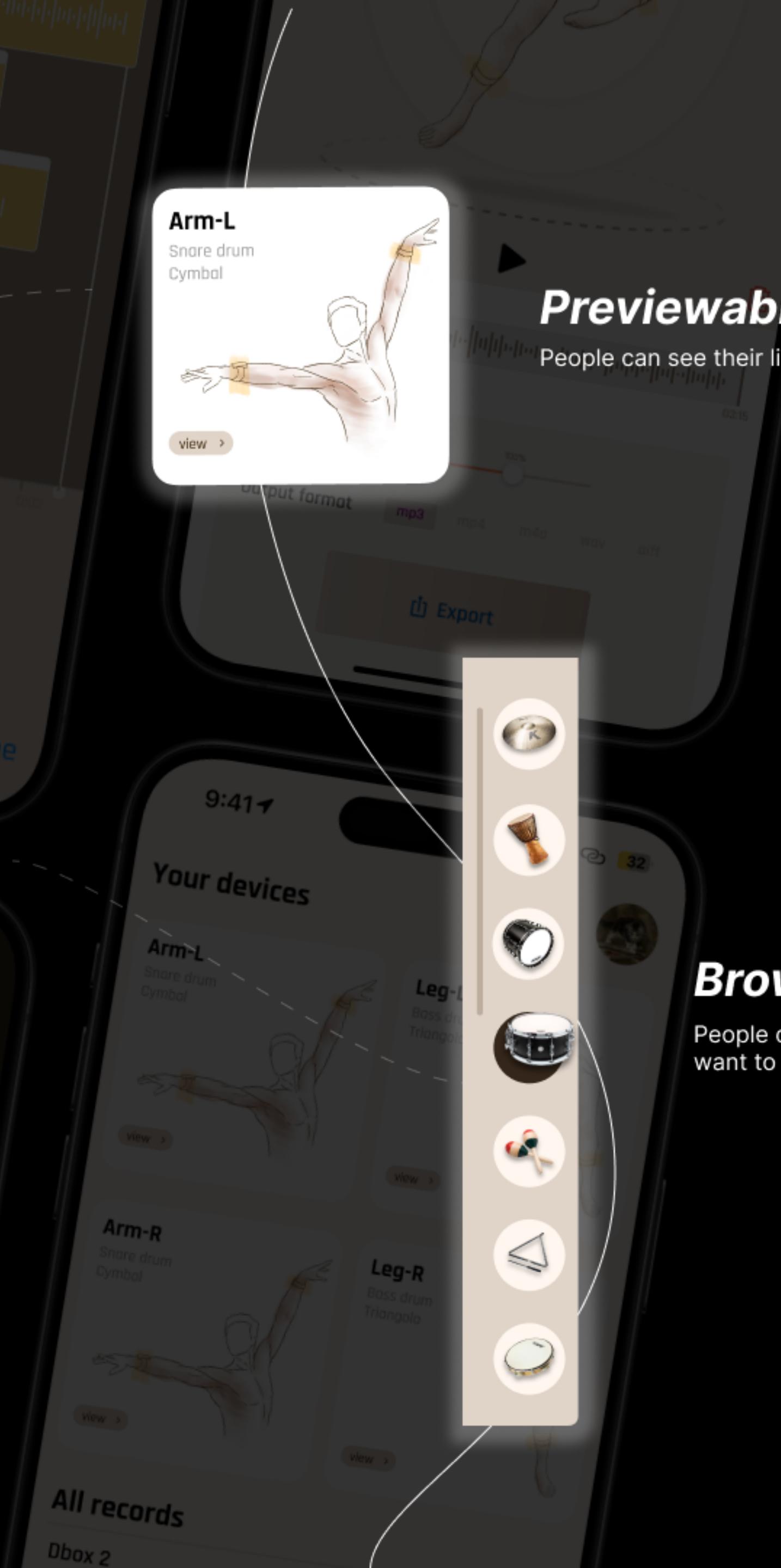
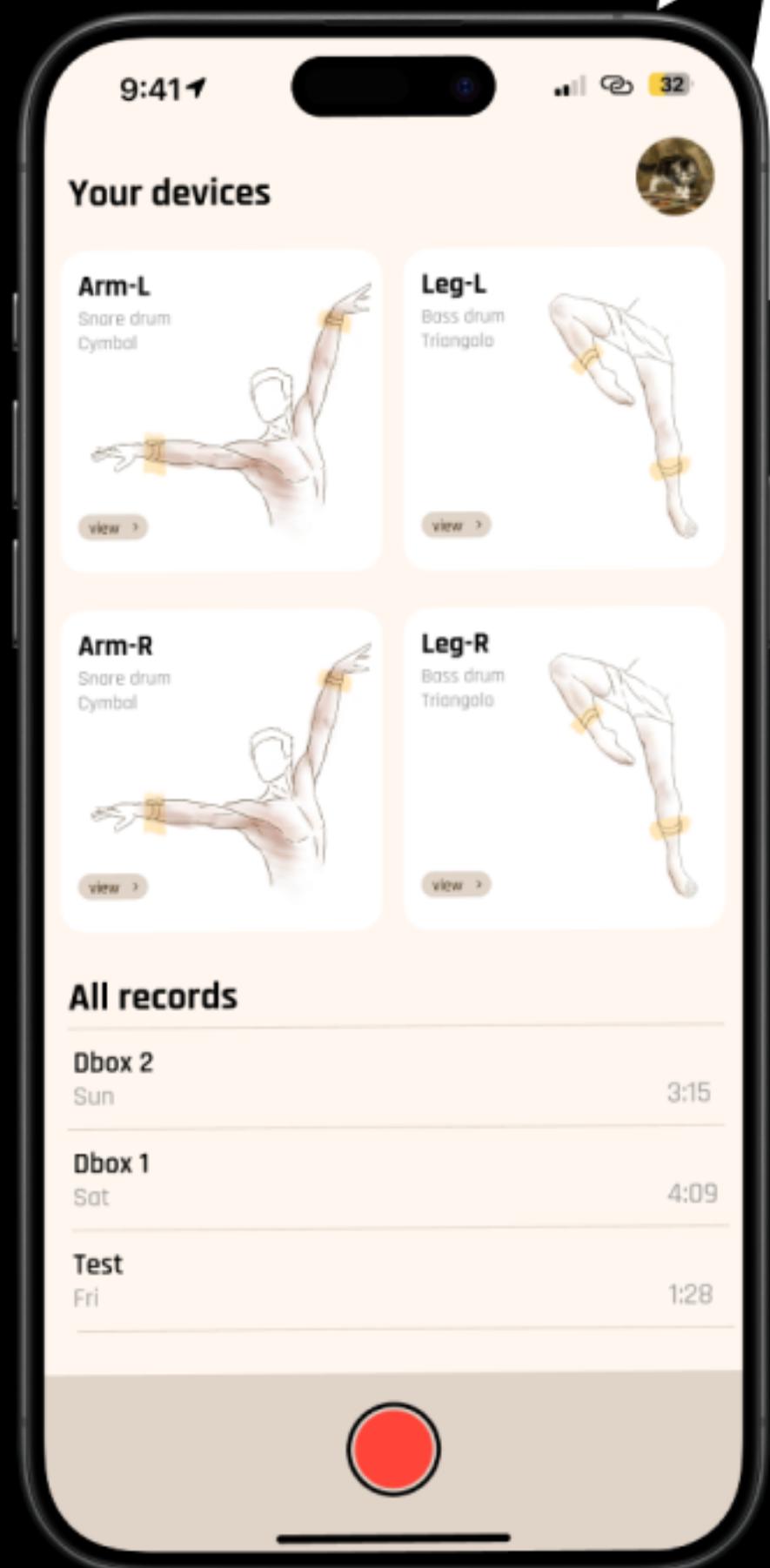
When people perform at their maximum speed, no matter how fast is it, once they reach their own max...



Pressing the button 3 times for 5 secs during setup teach Dbox your "full pressure" range individually.

According to this, the device will arrange you into **kid mode or adult mode**, which provide different intensity of force-sound/lighting translation.

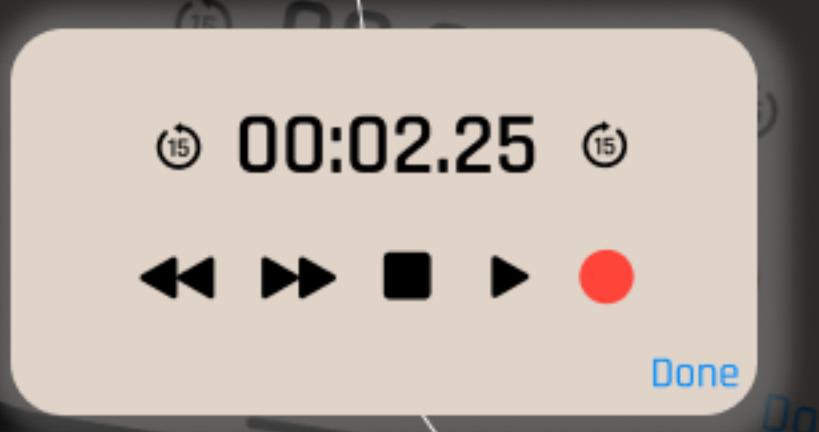
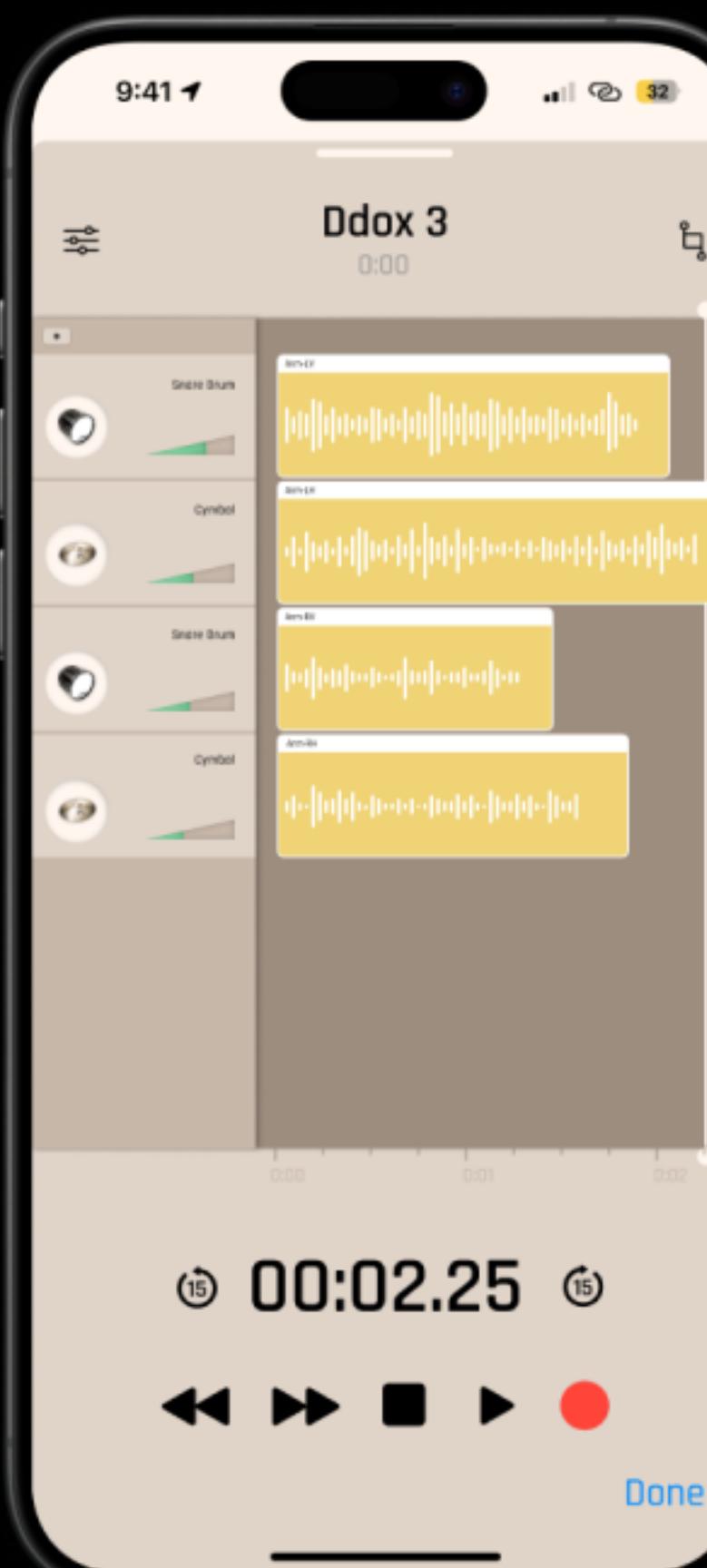




APP

APP

3



Hi Performer
Please choose the instrument

Remix

People can choose the format and volume they want to export their work here.

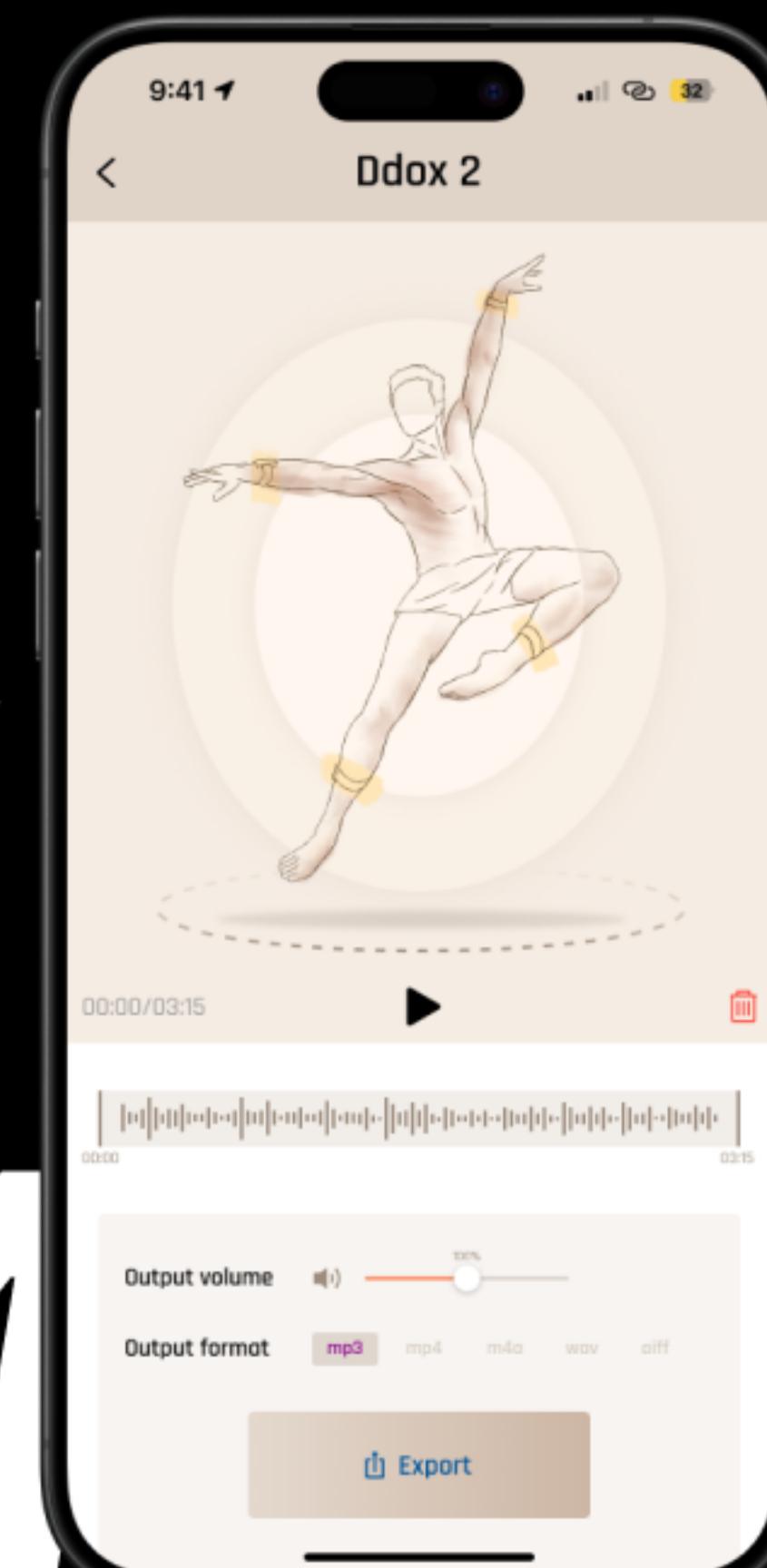
Recordable

People can record their instrument and change the volume and length of them there

Output volume
Output format
mp3 mp4 m4a wav off

Export

4

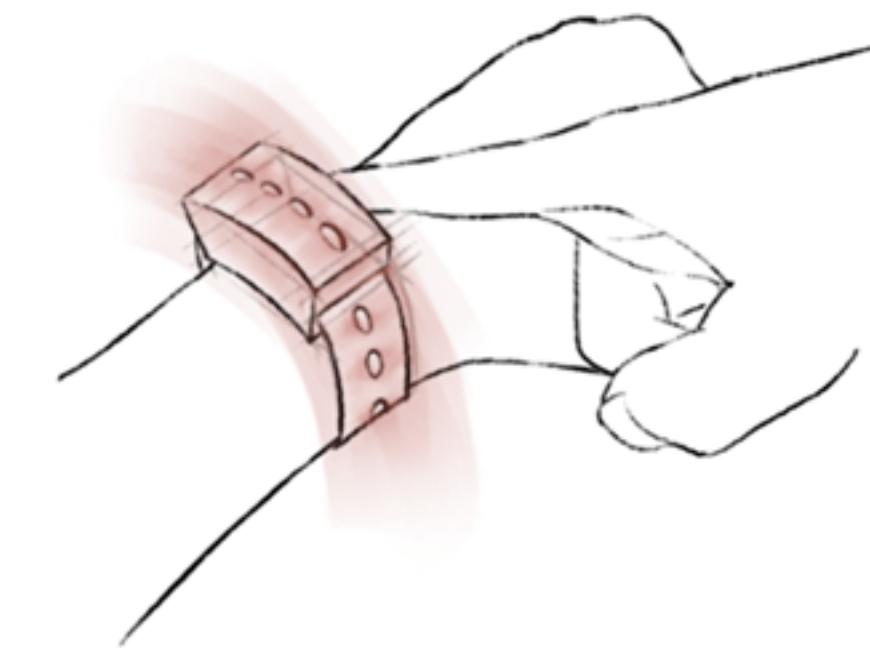


User's Flow



1. choose the instrument

the user selects the instrument
he want on app



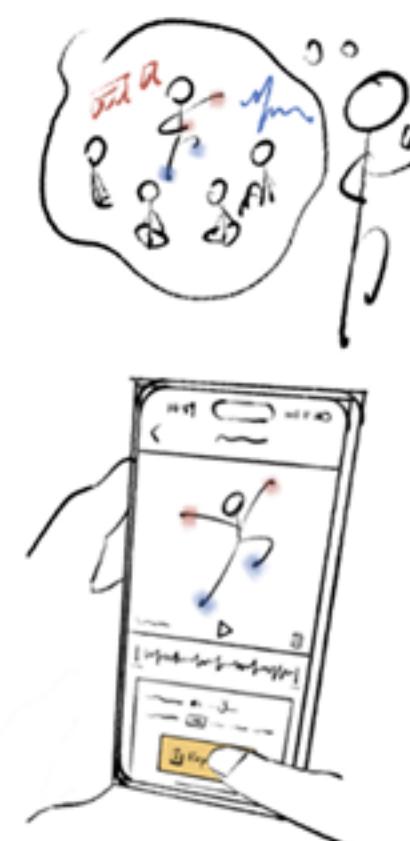
2. adjust the mode

the user is being arranged into the
proper mode for performing



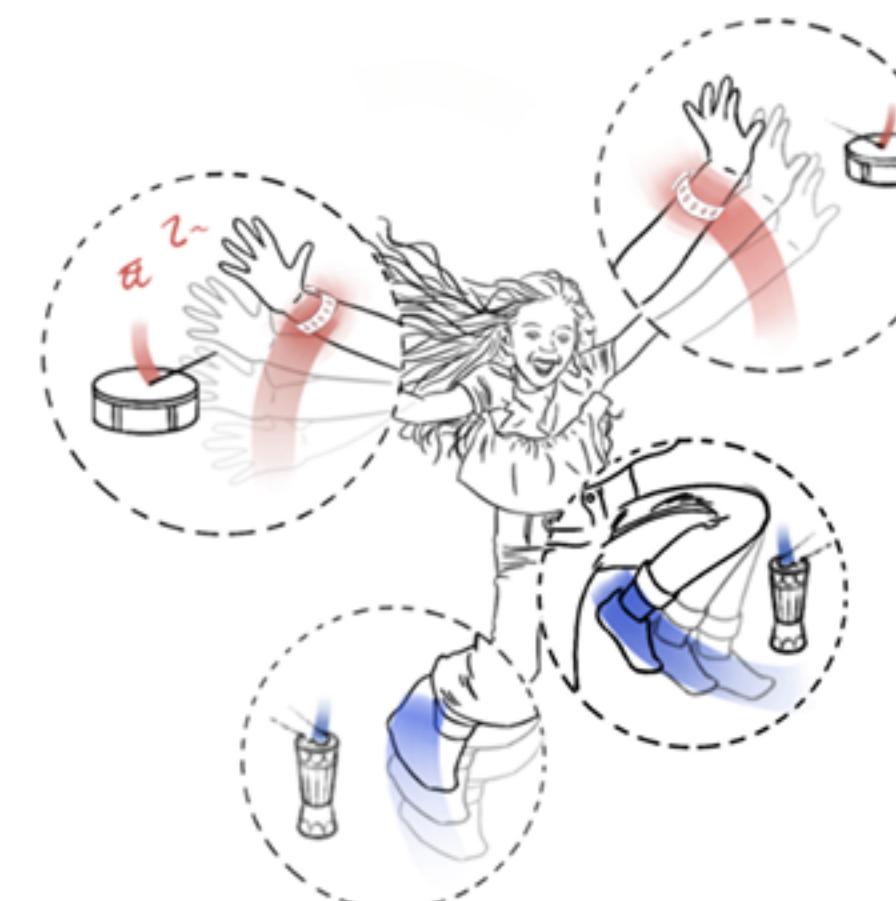
5. edit and share

the user can check their progress and
share it with others



4. perform

the user can start performing now



3. start recording

the user clicks on the button on app
to start record





Open Access Engineering Document:

<https://github.com/SandyZhang-art/Dbox-doc>