Listening Loop — interaction flow

Storyboard v2 · IAT 806 final project. How a visitor moves through the instrument and what the system does at each step.

Cards are states (the sketch is what the screen shows). Arrows are the trigger that moves to the next state. Diamonds are decisions. Numbers match the pseudocode.

visitor action (trigger)
system step (automatic)
takes effect on the next downbeat
takes effect instantly
alternative input, same actions
decision (IF … THEN …)

cyan = what you put in · magenta = what the model writes · amber = playhead / pending

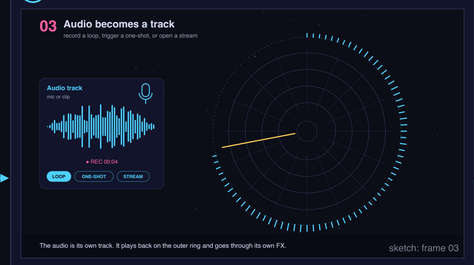
1 Idle — attract



VISITOR walks up. Nothing has been played yet.
SYSTEM SEES mic level, taps on the screen, dropped files.
SYSTEM DOES empty rings, slow noise-driven particles, prompt "give it a sound".

WHEN
mic > −30 dB
or taps LOOP / ONE-SHOT / STREAM
or drops a file

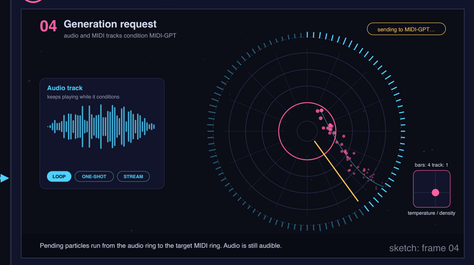
2 Audio becomes a track



VISITOR records a loop, triggers a one-shot, or opens a live stream.
SYSTEM SEES input mode + the audio buffer.
SYSTEM DOES plays it in the mix through its own FX; waveform fills the outer ring.

WHEN
taps GENERATE
or one audio loop
has passed (auto)

3 Generation request



VISITOR picks target track, bars, temperature / density (defaults work).
SYSTEM SEES audio features + any existing MIDI tracks as conditioning.
SYSTEM DOES POST to MIDI-GPT; audio keeps playing; particles flow from the audio ring to the target ring.

YES → swap in on next downbeat

MIDI back?

NOT YET → keep looping the audio, particles pending
> 8 s → play a cached variation

4 Loop plays



VISITOR listens.
SYSTEM SEES the shared clock: bar and beat.
SYSTEM DOES synth renders the notes; pitch → colour, velocity → size; playhead sweeps the rings.

BOUNCE: becomes a new audio track (feedback)
ADD: generate for the new track
infill request: bar range + pinned notes
temperature / density: pending → regenerate
synth / FX / mix: instant

8 Other ways to steer

VISITOR waves a hand, leans in, turns a hardware knob, holds text up to the camera.
SYSTEM SEES ml5 hand / body pose, MIDI CC or serial sensors, OCR.
SYSTEM DOES maps each input onto the same parameters as the on-screen controls (5–7).

7 Layer and bounce

VISITOR adds a MIDI track (drums, bass, chords, lead) or taps BOUNCE on a track.
SYSTEM SEES which button, which track.
SYSTEM DOES ADD → new ring, generate for it (→3). BOUNCE → render to audio, becomes a new audio track that can condition the model (→2).

6 Infill a region

VISITOR drags across an arc of a ring; long-presses notes to pin them.
SYSTEM SEES arc start/end angle → bar range; pinned note ids.
SYSTEM DOES regenerates only that range (→3); old notes fade to ghosts, new ones grow in.

5 Steer the sound

VISITOR turns synth / FX / mix knobs or moves the temperature / density pad.
SYSTEM SEES which control moved.
SYSTEM DOES IF pipeline control → apply now (stay in 4). IF generation control → mark pending, regenerate on the downbeat (→3).

every bar: what does the visitor do?

touch → take over

feeds the same decision
adds / bounces a track
drags an arc
turns a control
IF no input for 60 s

Pseudocode — the same flow as the diagram

```
state = IDLE                                                          // 1  attract screen
on mic > −30 dB  or tap(LOOP | ONE_SHOT | STREAM)  or dropFile:
    audio = capture(mode); outerRing.show(audio)                      // 2  plays through its own FX
on tap(GENERATE)  or audio.loopsPlayed >= 1:
    req = { audio: features(audio), midi: midiTracks, bars: 4, temp, density }
    pending = post('/generate', req); particles.flow(audioRing -> target)   // 3
on downbeat:                                                          // shared clock
    if pending.done:          swap(pending.midi); state = PLAYING      // 4
    elif pending.age > 8 s:   swap(cache.next())                       // fallback
    if queued: pending = post('/generate', queued); queued = {}       // 5  next-loop params
every bar while PLAYING:                                              // decision
    if moved(synth | fx | mix):   apply(control)                      // 5  instant
    if moved(temp | density):     queued += control                   // 5  pending
    if arcSelected:   pending = post('/infill', track, barsIn(arc), pinned)   // 6
    if addTrack(kind): pending = post('/generate', newTrack)          // 7
    if bounce(track):  audioTracks.add(render(track))                 // 7  feedback
    gestures, sensors, OCR -> same calls as the controls              // 8
    if noInputFor(60 s): state = EXHIBITION                           // 9
in EXHIBITION:  every 8 bars generate(randomTrack)                    // 9
    on touch: state = PLAYING;   if emptyFor(5 min): reset(); state = IDLE   // → 4 / → 1
```

How it's built

FRONT END p5.js: radial sequencer, particles, all controls. One clock drives sound and visuals.
MODEL MIDI-GPT HTTP server: track generation and bar infilling (CORS enabled).
SOUND audio track → FX → mix; MIDI → synth → FX → mix. Max/MSP, RAVE, Stable Audio optional.

9 Exhibition mode

VISITOR nobody is there, or a passer-by touches the screen.
SYSTEM SEES no input for 60 s; then any touch or sound.
SYSTEM DOES auto-generates every 8 bars with ghost trails. Touch → hand control back (→4). Empty 5 min → RESET (→1).

nobody for 5 min → RESET to 1

Sketches drawn with AI assistance (Claude); ideas and direction are mine.