import \* as Tone from 'tone';

document.body.innerHTML = `

<div class="game-container">

<h2>创作你的旋律</h2>

<p>点击音符按钮，并拖动到时间线上，排列你的旋律。</p>

<div id="note-selection">

<button onclick="selectNote('G')">选择 G</button>

<button onclick="selectNote('A')">选择 A</button>

</div>

<div id="timeline" class="timeline" ondrop="drop(event)" ondragover="allowDrop(event)"></div>

<canvas id="visualizer" width="400" height="100"></canvas>

<button onclick="playMelody()">播放</button>

<button onclick="resetMelody()">重置</button>

</div>

<style>

.game-container { text-align: center; margin-top: 20px; }

button { margin: 5px; padding: 10px; font-size: 16px; }

.timeline { margin-top: 20px; min-height: 50px; border: 2px dashed #000; padding: 10px; }

.note { display: inline-block; padding: 5px 10px; margin: 5px; background: lightblue; cursor: grab; }

canvas { margin-top: 20px; border: 1px solid #000; }

</style>

`;

const synth = new Tone.Synth().toDestination();

const analyser = new Tone.Analyser("waveform", 1024);

synth.connect(analyser);

const canvas = document.getElementById("visualizer");

const ctx = canvas.getContext("2d");

let selectedNote = null;

let melody = [];

function selectNote(note) {

selectedNote = note;

}

function allowDrop(event) {

event.preventDefault();

}

function drop(event) {

event.preventDefault();

if (selectedNote) {

let noteElement = document.createElement('div');

noteElement.className = 'note';

noteElement.innerText = selectedNote;

noteElement.dataset.note = selectedNote;

noteElement.dataset.duration = 1;

document.getElementById('timeline').appendChild(noteElement);

melody.push({ note: selectedNote, duration: 1 });

}

}

async function playMelody() {

await Tone.start();

let now = Tone.now();

melody.forEach((n, i) => {

synth.triggerAttackRelease(n.note + '4', n.duration, now + i);

});

visualize();

}

function resetMelody() {

melody = [];

document.getElementById('timeline').innerHTML = '';

}

function visualize() {

requestAnimationFrame(visualize);

let data = analyser.getValue();

ctx.clearRect(0, 0, canvas.width, canvas.height);

ctx.beginPath();

for (let i = 0; i < data.length; i++) {

let x = (i / data.length) \* canvas.width;

let y = (0.5 - data[i] / 2) \* canvas.height;

ctx.lineTo(x, y);

}

ctx.stroke();

}

visualize();