+33 (0)6 01 62 60 75

https://www.loukail.xyz

hello@loukail.xyz
lou.kail@edu.esiee.fr

Étudiant en Cycle Ingénieur à ESIEE Paris
Filière Informatique et Applications - Ingénierie 3D et Technologie des Médias

Lou Kail

Portfolio

Loustoc

Sommaire

Sentiers Aveugles - - -

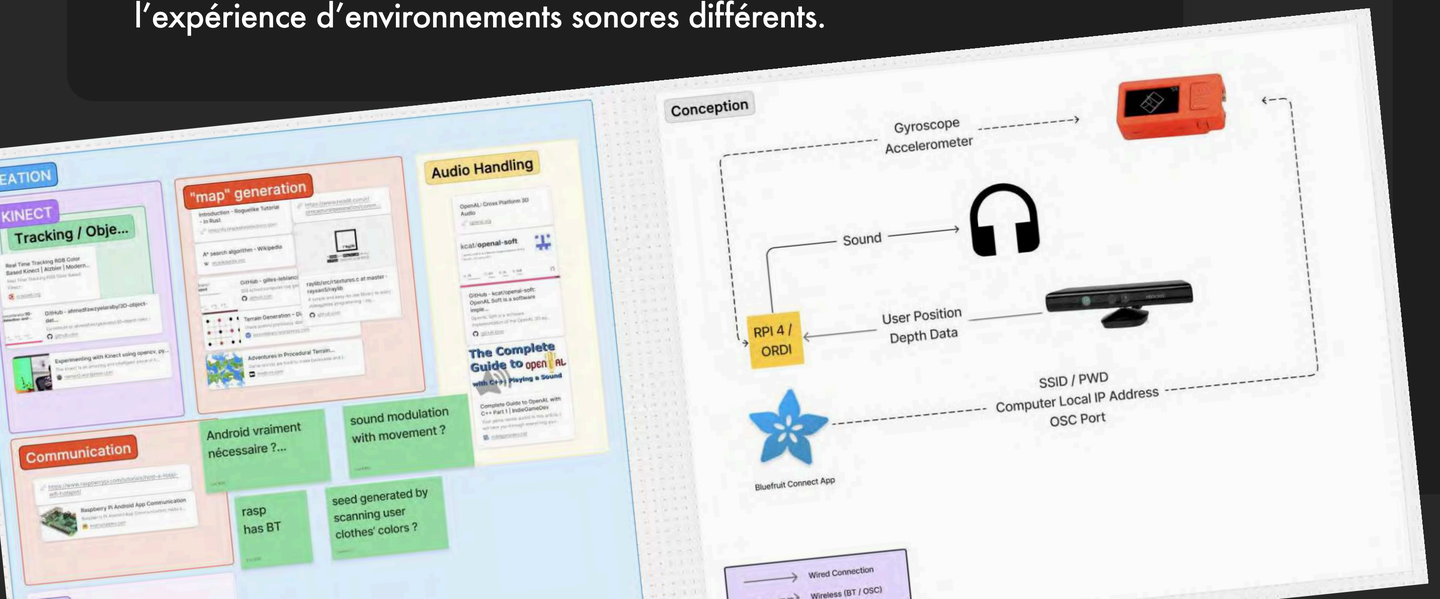
Sentiers Aveugles

KINECT

Sentiers Aveugles

Sentiers Aveugles est mon projet de fin de deuxième année de DNMADe, suivant le thème imposé *"Les 4 éléments et les 5 sens"*.
J'ai fait le choix, en cohérence avec mes appétences, de réaliser un dispositif tangible et interactif.

À l'aide d'une Kinect (v.1) et d'un gyroscope, la position de l'utilisateur, qui porte un casque audio sur la tête, fait varier des effets sonores directionnels, ce qui lui permet de se déplacer pour faire l'expérience d'environnements sonores différents.

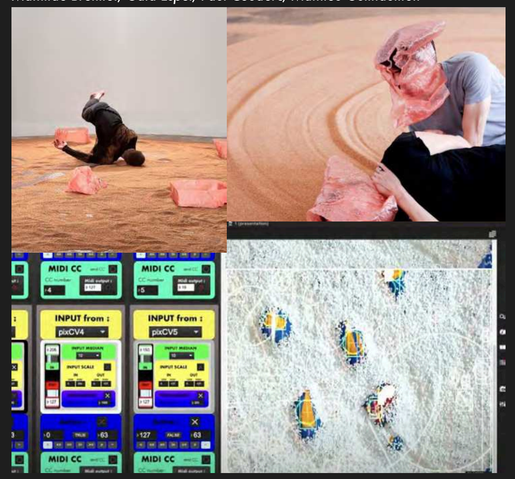


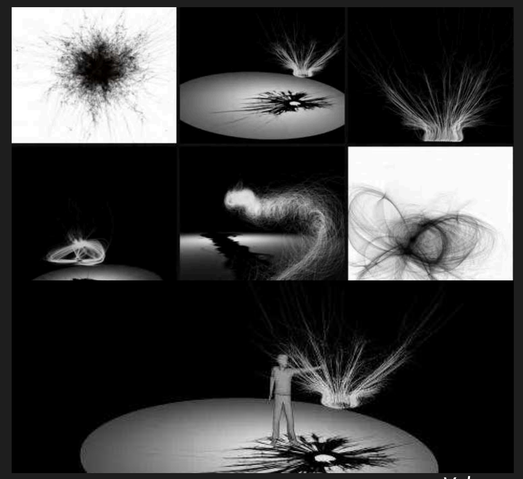
J'ai été inspiré par de multiples références faisant le lien entre les sens, et tout particulièrement entre le son et la gestuelle.
Deux réalisations tangibles en particulier ont été source de réflexion et m'ont permis de développer mon propre projet.
Elles ont en commun la liberté donnée à l'utilisateur d'évoluer dans un environnement donné, lequel semble alors réagir à ses déambulations.

Références

+ Versatilité de MaxMSP
+ Intrication ouïe - gestuelle - vue
- Encombrant

Mon île
Mathilde Bretillot, Gala Espel, Paul Couderc, Mathieu Guilhaumon





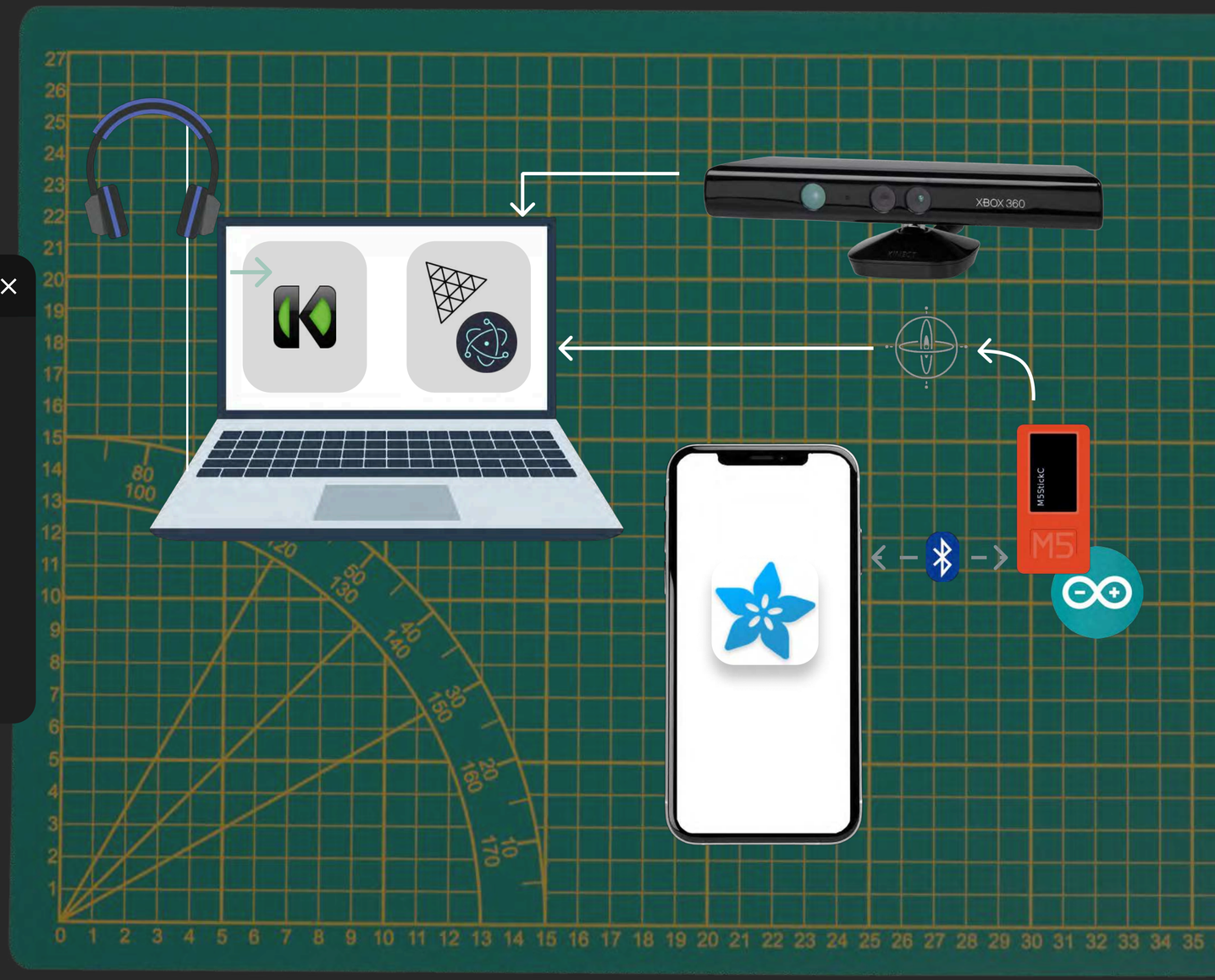
Volumes
Mathieu Chamagne

+ Interactions entre utilisateurs
+ Intrication ouïe - vue
- Dispositif Coûteux

Sentiers Aveugles

KINECT

Un M5Stick-C, une objet programmable pourvu d'un accéléromètre / gyroscope, est fixé sur au sommet d'un casque audio, porté par l'utilisateur.
Celui-ci se tient devant une Kinect, qui transmet des données de profondeur et de position à une application Electron.JS via le protocole Open Sound Control, et à l'aide de la librairie LibFreenect.
Celle-ci, grâce à l'objet **PositionalAudio**, interprète la position de l'utilisateur dans l'environnement sonore, permettant au casque d'émettre des sons directionnels, qui réagissent à sa position comme à ses mouvements de tête.



Sentiers Aveugles

KINECT

Lecture des données gyroscopique depuis le M5Stick-C

```
float readGyroData() {
    float gyroX, gyroY, gyroZ;
    M5.IMU.getGyroData(&gyroX, &gyroY, &gyroZ);

    unsigned long currentTime = millis();
    float dt = (currentTime - lastTime) / 1000.0;
    lastTime = currentTime;

    angleZ += gyroZ * dt;

    angleZ = fmod(angleZ, 360.0);
    if (angleZ < 0) {
        angleZ += 360.0;
    }
    return angleZ;
}
```

Boucle envoyant en OSC la "matrice de position" à l'application Electron.JS

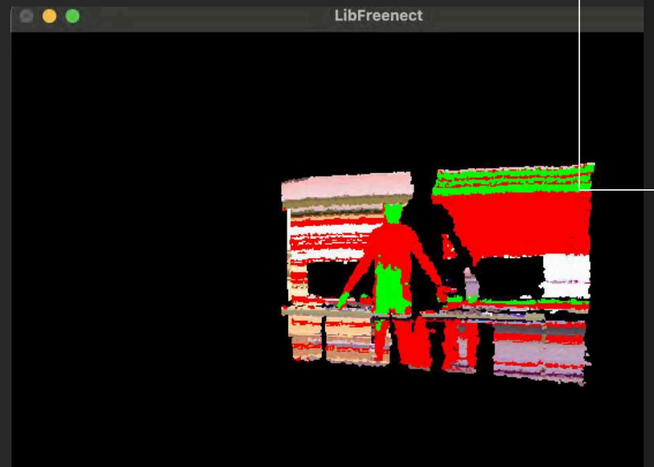
```
for (int i = 0; i < SUBDIV; ++i) {
      for (int j = 0; j < SUBDIV; ++j) {
        bodyPerc[i][j] = (float)map[i][j] / (float)totalBodyPoints;
        if (totalBodyPoints > 2000) {
          printf("%f", (bodyPerc[j][i]));
          printf(" ");
          int len = tosc_writeMessage(buffer, sizeof(buffer), "/data", "fff",
                                      (float)i, (float)j, bodyPerc[i][j]);
          if (sendto(oscSendSocket, buffer, len, 0, (struct sockaddr *)&oscAddr,
                     sizeof(oscAddr)) < 0) {
            perror("Failed to send OSC message");
            close(oscSendSocket);
            exit(1);
          }
        }
        printf(" ");
      }
```

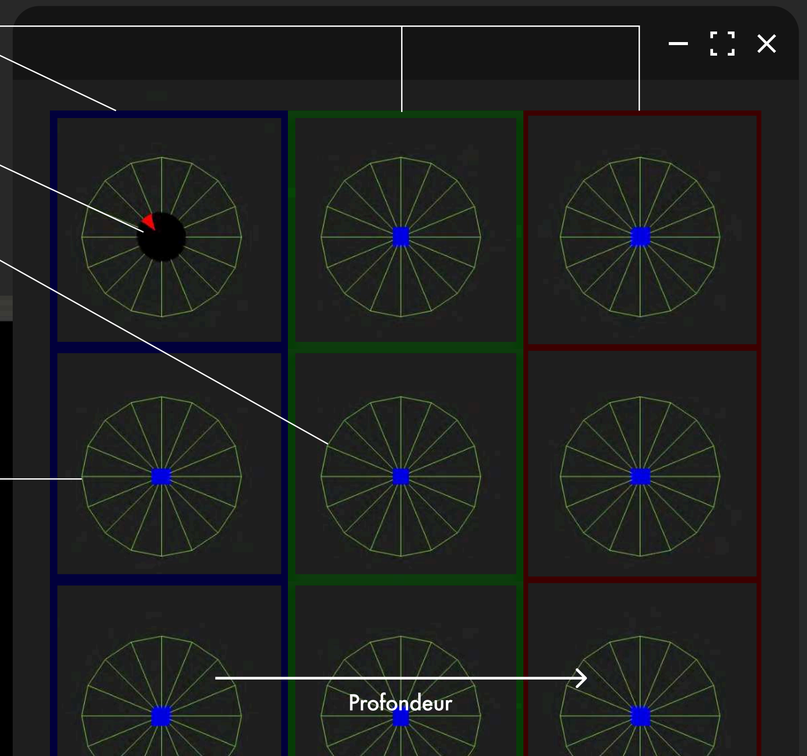
L'environnement est subdivisé en 9 zones, chacune d'entre elles étant associée à un son. La Kinect détermine alors dans quelle zone se trouve l'utilisateur, et envoie une matrice (3x3) de position, en pondérant cette dernière en fonction de la position de l'utilisateur.
(Il peut ainsi se trouver entre plusieurs cases.)
L'application Electron interprète ensuite cette matrice, et place l'utilisateur dans l'environnement sonore réalisé à l'aide de Three.JS.

Quadrillage de l'environnement

Utilisateur (Le segment rouge indique la position des yeux)

Meshes "émetteurs" de son



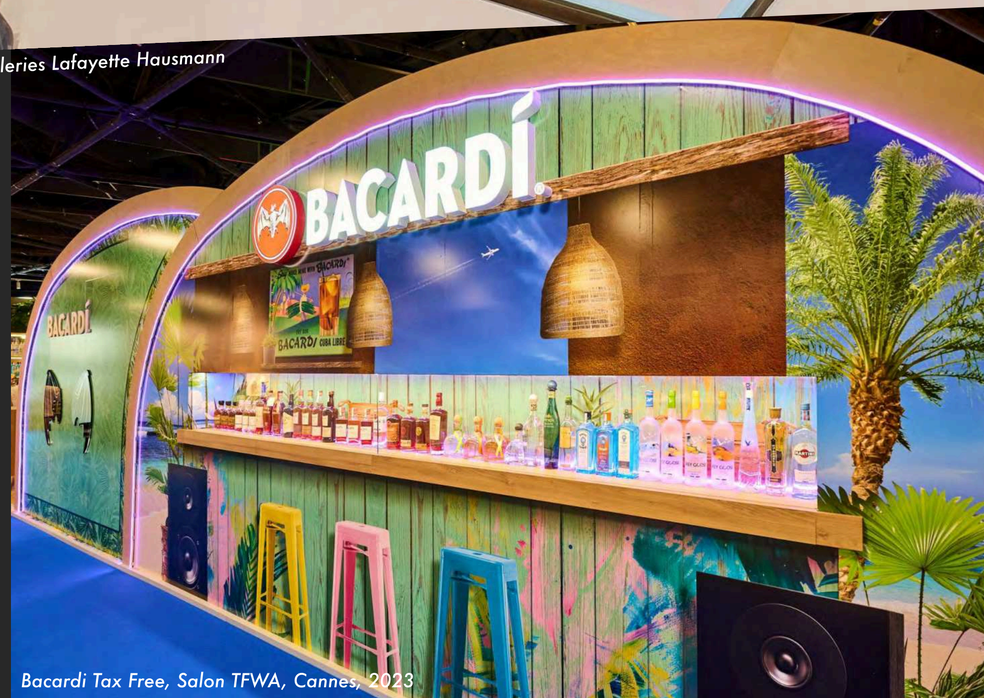


Freelance

Ayant créé ma micro-entreprise en Août 2023, j'ai depuis réalisé plusieurs missions pour l'entreprise m'ayant accueilli en stage la même année : Le Club Sandwich Studio.

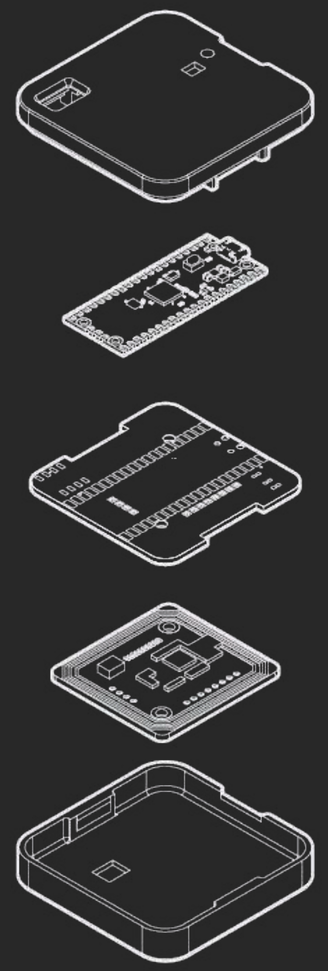
J'ai réalisé pour eux plusieurs applications Electron.JS, destinées à être déployées en boutique (ou sur des stands d'exposition dans le milieu du retail).
Celles-ci s'inscrivent dans une démarche d'activation produit et s'articulent autour de cartes programmables produites par le Club Sandwich : Le MiM (et le MiM NFC)
Ces cartes sont pourvues respectivement de boutons poussoir et d'un Capteur NFC, et sont entièrement modulables et paramétrables.
Ainsi, les applications permettent la modification à la volée des expériences ainsi que leur paramétrage.

GUCCI Galeries Lafayette Hausmann



Bacardi Tax Free, Salon TFWA, Cannes, 2023

Freelance



MiM NFC

Pour le compte de Gucci, j'ai participé à l'élaboration d'un dispositif interactif faisant la promotion de leur nouvelle gamme de parfum.
J'ai ainsi réalisé une application **Electron.JS** communiquant avec un lecteur NFC conçu par le Club Sandwich (le MiM NFC).

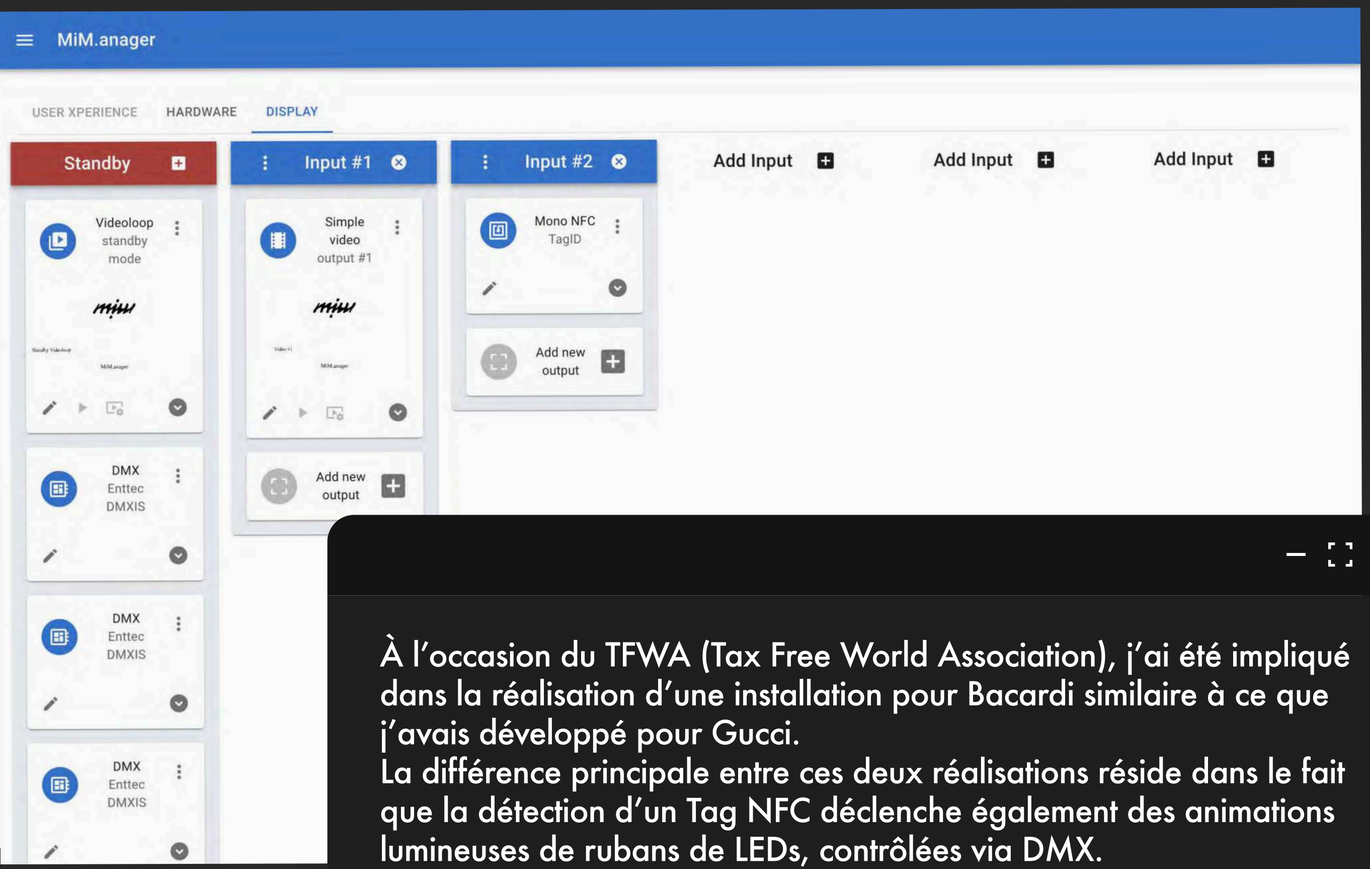
À l'aide d'un fichier de configuration de Tags NFC modifiables via une interface dédiée, différentes animations et informations s'affichent selon le modèle de parfums posé sur le socle NFC.

Cliquez ici pour en voir plus !

GUCCI GENEVA (showroom)

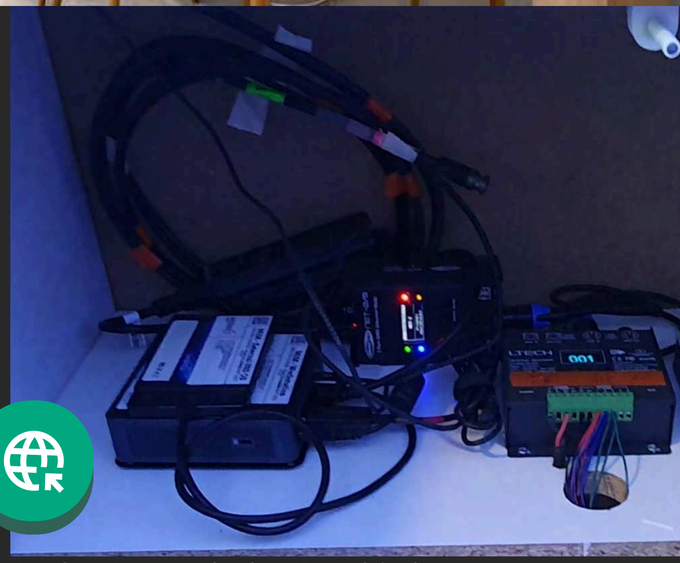
GUCCI Galeries Lafayette Hausmann

Freelance



À l'occasion du TFWA (Tax Free World Association), j'ai été impliqué dans la réalisation d'une installation pour Bacardi similaire à ce que j'avais développé pour Gucci.
La différence principale entre ces deux réalisations réside dans le fait que la détection d'un Tag NFC déclenche également des animations lumineuses de rubans de LEDs, contrôlées via DMX.
L'application que j'ai conçue permet ainsi de personnaliser les animations selon le Tag détecté, qu'il s'agisse de la couleur, du délai, ou encore du rythme.

Bacardi Tax Free, Salon TFWA, Cannes, 2023



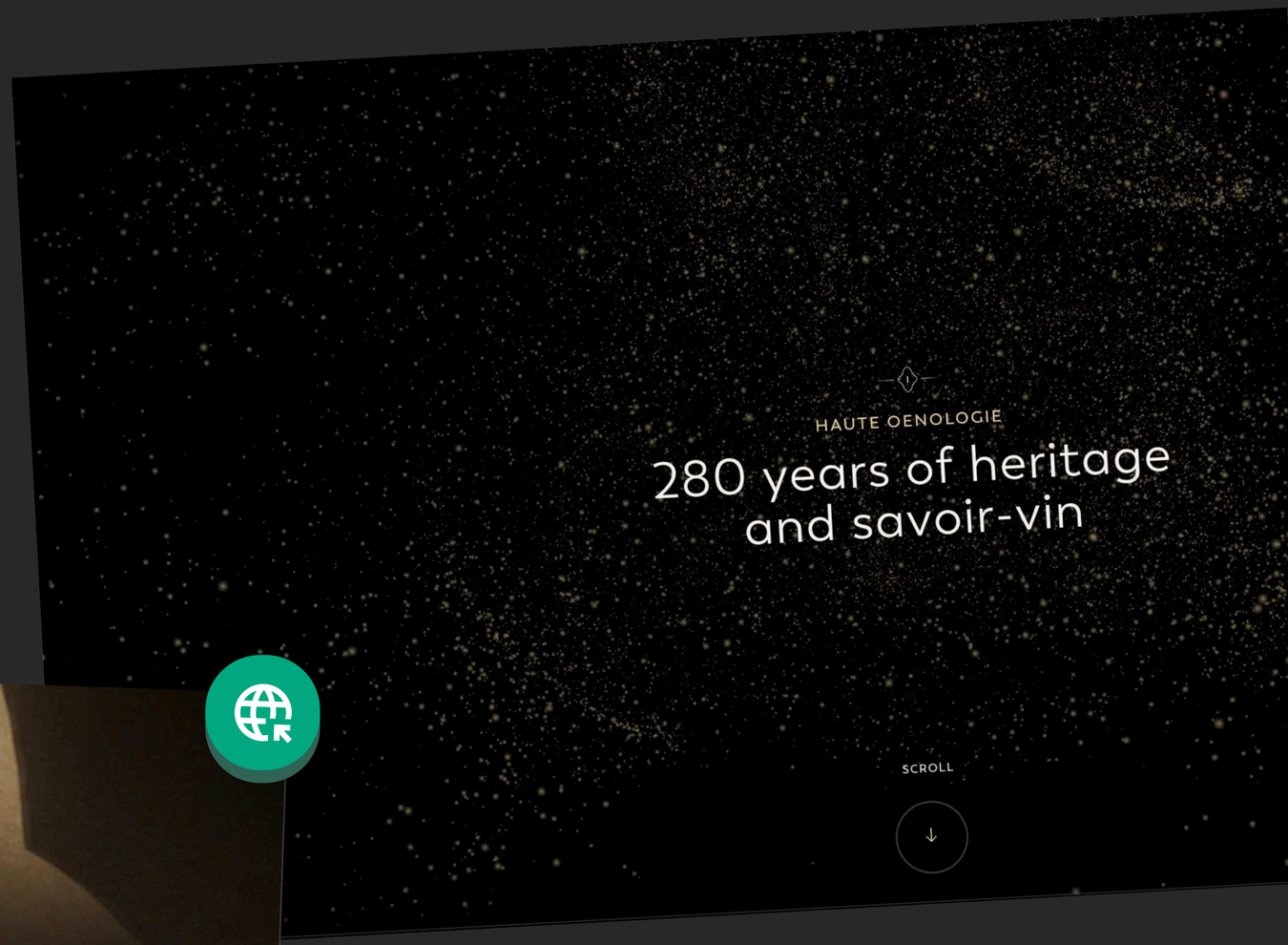
Décodeurs DMX contrôlant les LEDs au plafond

Alternance

AKQA

Durant mon alternance à AKQA, j'ai eu l'opportunité de prendre part à la conception et au développement de plusieurs sites web, reposant sur des stacks techniques diverses.

Réalisé à l'occasion des 280 ans de Moët & Chandon, ce site web retrace l'histoire de la maison, et fait la promotion de la Collection Impériale, leur dernière cuvée.
J'ai notamment participé à la création des shaders de texte, ainsi qu'à la réalisation des diverses animations qui y figurent en GSAP.

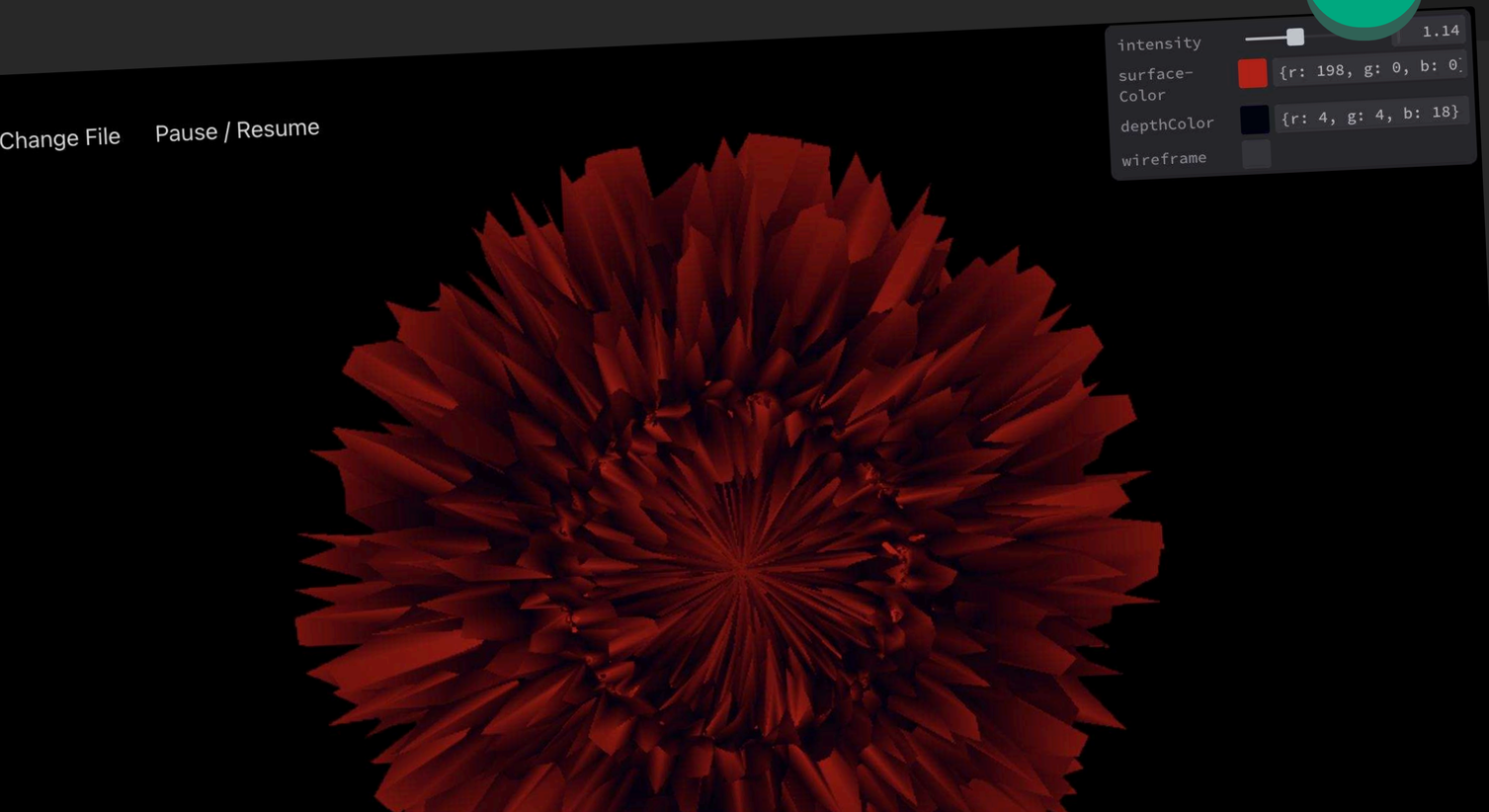


Laboratoire

Les prochaines réalisations sont des sites web ayant été développés dans le cadre d'un workshop **Three.JS** *et* **WebAudio API**.
L'objectif principal était alors de produire, en un temps réduit, une ou plusieurs expériences interactives faisant usage de ces technologies, et témoignant d'une démarche d'appropriation de ces dernières.

La première expérience fait usage de l'API WebAudio; cette dernière est employée pour exploiter un filtre basse fréquences appliquée sur un fichier audio.

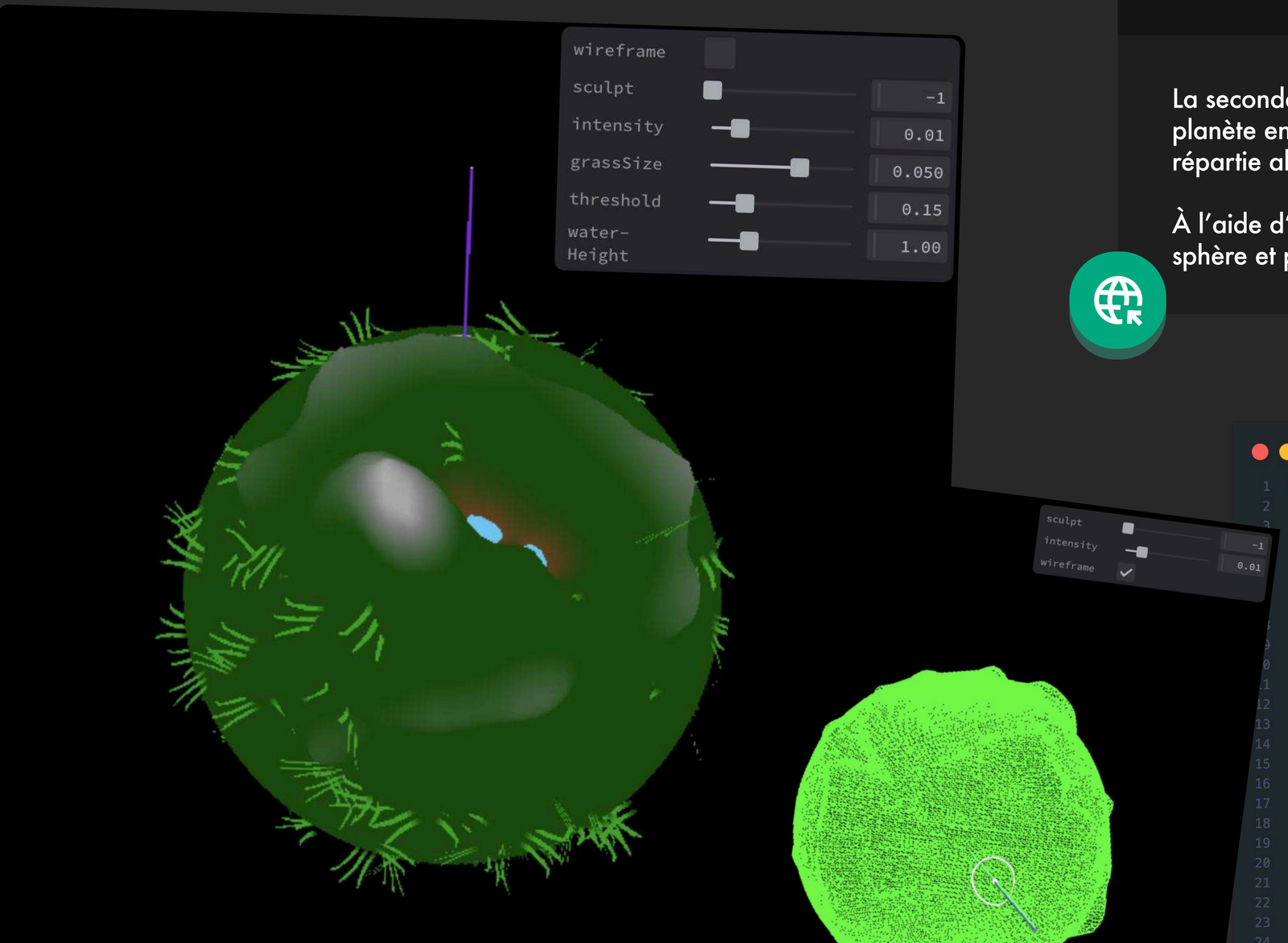
Les basses fréquences sont ensuite liées aux vertices d'un mesh 3D sur Three.JS, lesquels pulseront au rythme de la musique.



Fonction associant la force des basses aux vertices

```
function animate() {
    planetMesh.rotation.y += 0.001;
    if (bassData) {
      let acc = Array.from(bassData).reduce((acc, curr) => {
        return acc + curr;
      }, 0);
      acc /= bassData.length;
      for (
        let i = 0;
        i < planetGeometry.attributes.position.array.length / 20;
        i++
      ) {
        const index = Math.floor(
          Math.random() * (planetGeometry.attributes.position.array.length - 1)
        );
        const vertIndex = index - (index % 3);
        const vert = new THREE.Vector3(
          planetPositionsMem[vertIndex],
```

Laboratoire



La seconde expérience, **Artlas**, permet le modelage d'une petite planète en ligne, permettant la création de relief et la tonte de l'herbe répartie aléatoirement sur la sphère lors de sa génération.

À l'aide d'un Raycaster, la position du curseur épouse la forme de la sphère et permet un modelage localisé.

Boucle déterminant l'intensité de la déformation en fonction de la distance au curseur

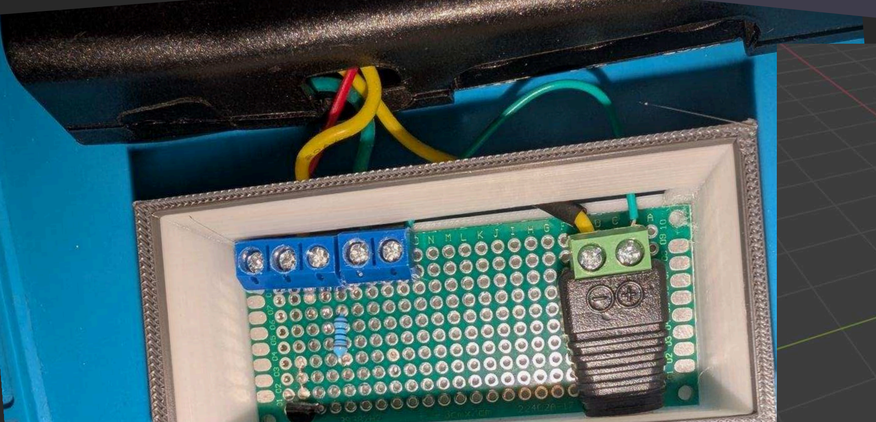
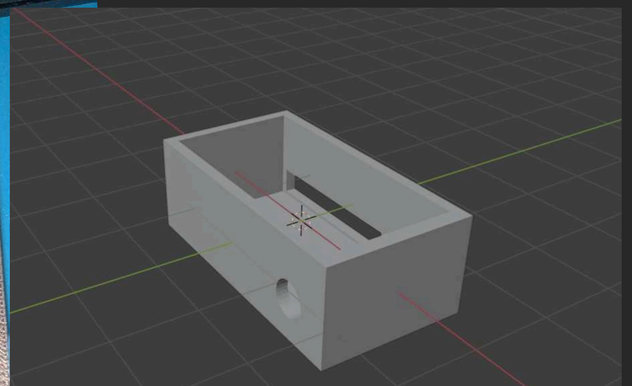
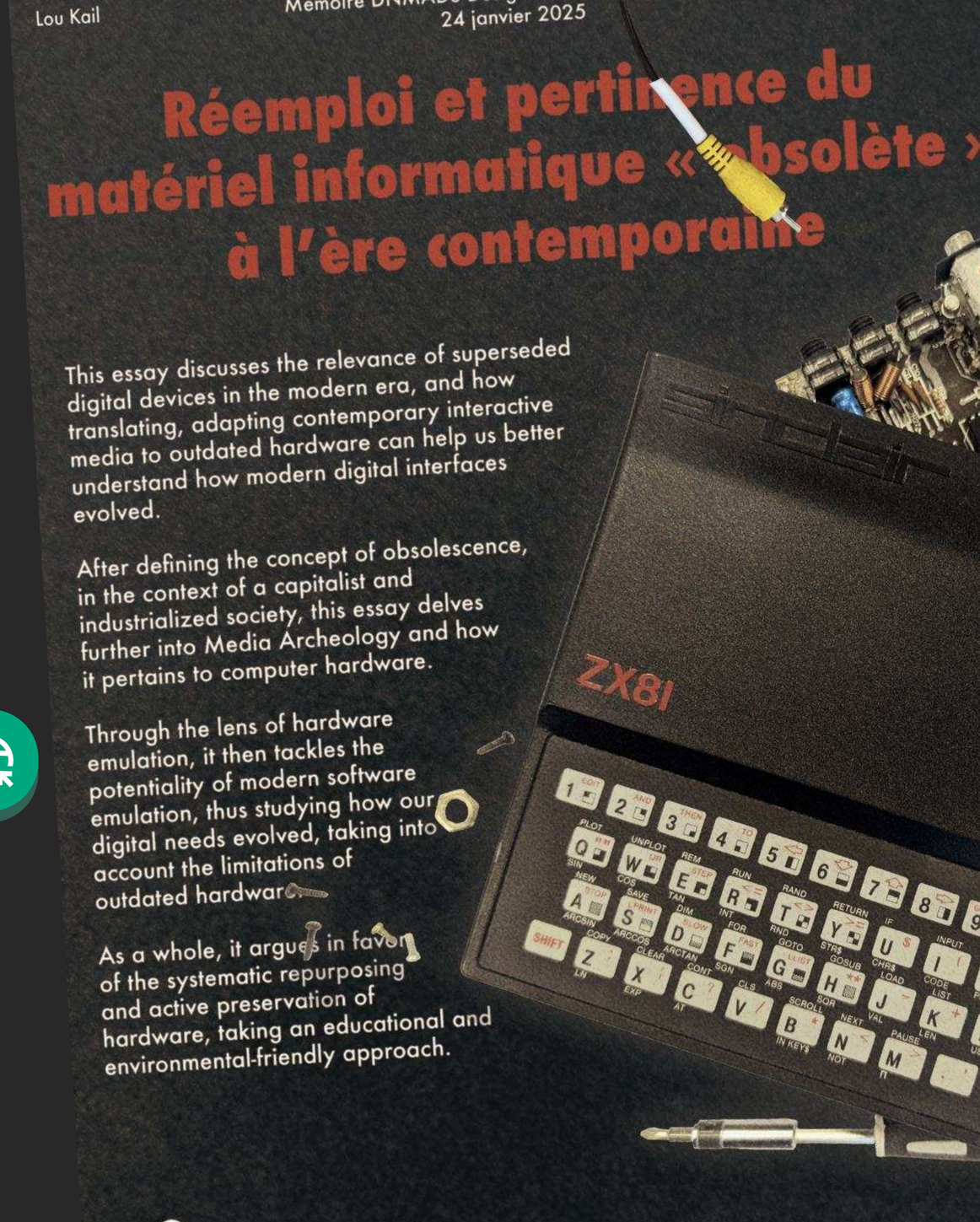
```
for (let i = 0; i < planetGeometry.attributes.position.array.length; i += 3) {
        const neighbourVertex = new THREE.Vector3(
          planetGeometry.attributes.position.array[i],
          planetGeometry.attributes.position.array[i + 1],
          planetGeometry.attributes.position.array[i + 2]
        );

        const distance = neighbourVertex.distanceTo(vertex);
        if (distance < PARAMS.threshold) {
          const dist2Center = neighbourVertex.distanceTo(new THREE.Vector3(0, 0, 0));
          let factor = (dist2Center + PARAMS.sculpt * PARAMS.intensity) / dist2Center;
          factor = 1 + PARAMS.sculpt * Math.abs(1 - factor) * (1 - distance / PARAMS.threshold);
          if (lawnMowerMode) {
            instancedMesh.setMatrixAt(i / 3, new THREE.Matrix4());
            instancedMesh.instanceMatrix.needsUpdate = true;
          } else {
            sculptArray[i / 3] = sculptArray[i / 3] * factor;
          }
        }
        planetGeometry.attributes.sculptAttribute.array.set(sculptArray);
        instancedBufferGeometry.attributes.sculptAttribute.array.set(sculptArray);
        planetGeometry.attributes.sculptAttribute.needsUpdate = true;
        instancedBufferGeometry.attributes.sculptAttribute.needsUpdate = true;
      }
```

Mémoire

Dans le cadre de mon DNMADe à Gobelins - Paris, j'ai rédigé un mémoire traitant de la pertinence du matériel informatique dit "obsolète" à l'ère contemporaine.

J'y souligne l'importance de la préservation fidèle des expériences tangibles d'origine, tout en mettant en valeur des démarches diverses et créatives de préservation du patrimoine informatique :
demoscene, modding, ajout de fonctionnalités...

Pour accompagner mon mémoire, j'ai moi-même "moddé" un ordinateur personnel des années 1980, le Sinclair ZX81, en altérant la sortie vidéo d'origine pour le rendre compatible avec le standard composite.

Composite mod