

Louis, l'étoile tombée du ciel

Cahier de recherches

Jour 1 : Phase d'idéation et premiers tests

Univers de jeu

Inspirations de jeux :

- Ori and the blind forest
- Azur et Asmar
- Princes et Princesses
- Journey
- Monument Valley
- Alto Odyssey

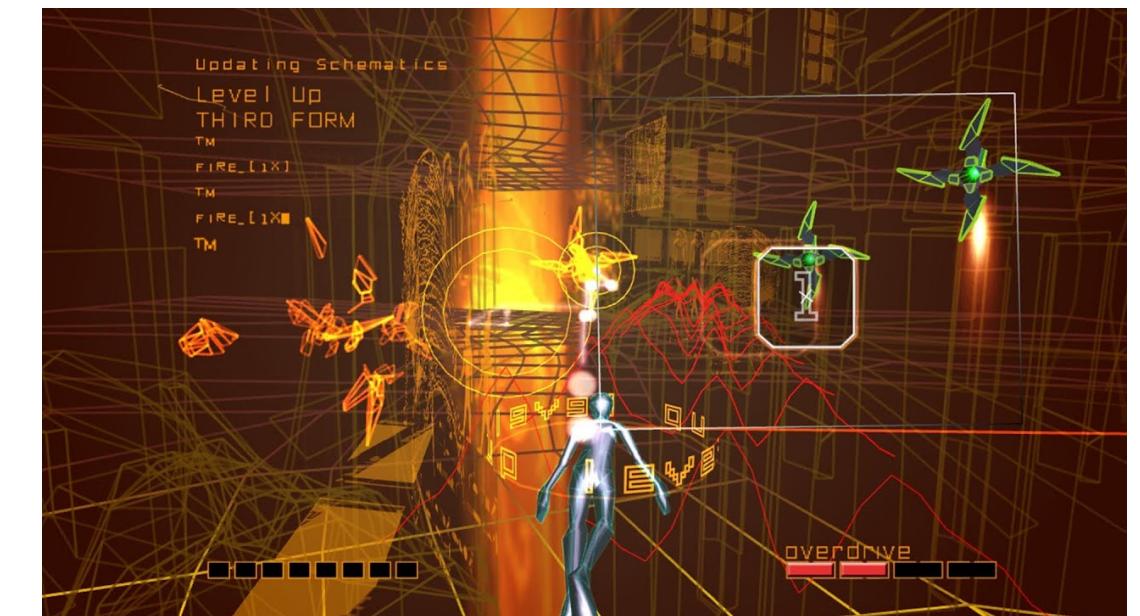


Princes et Princesses - Michel Ocelot

Changer l'ambiance mais pas le fond

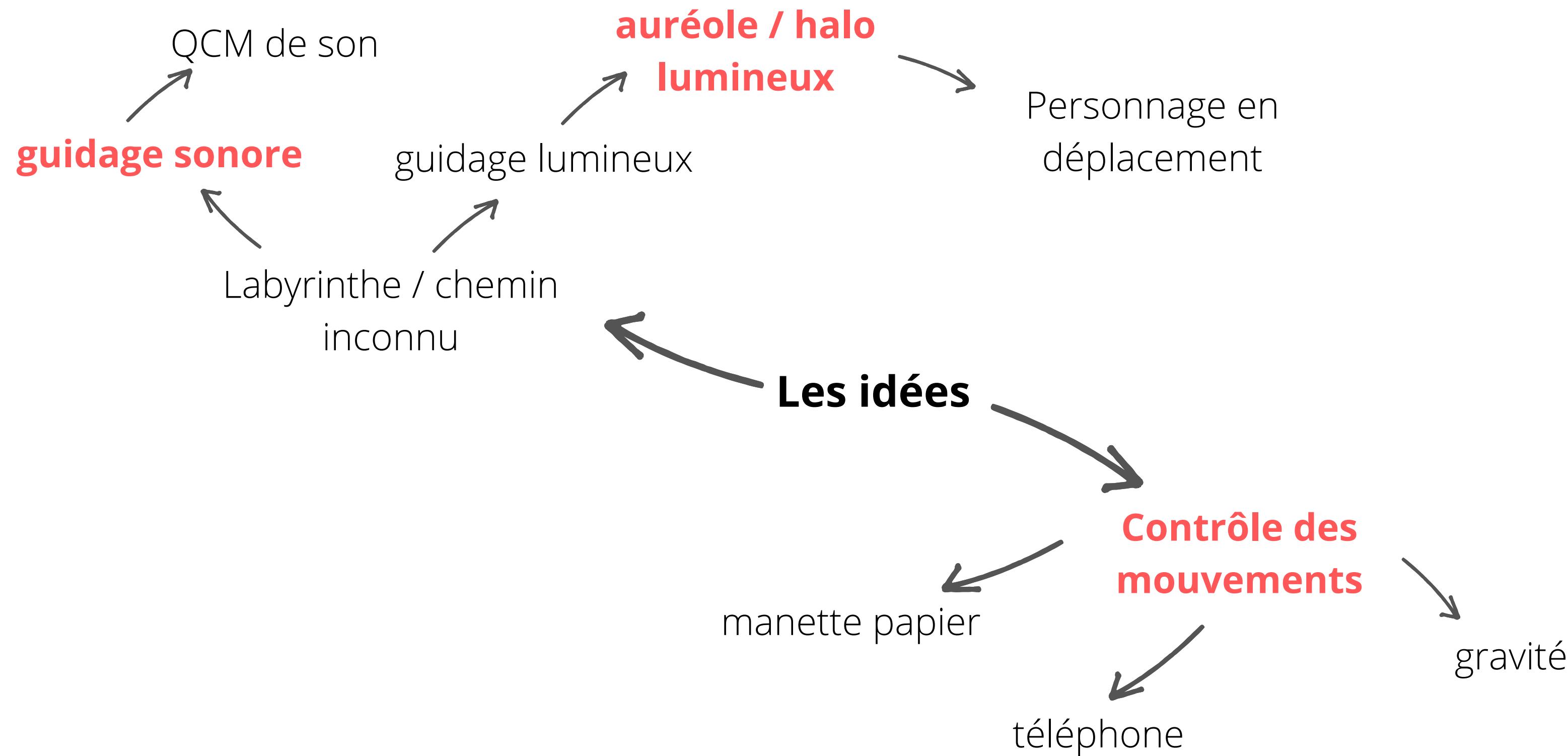


Journey - Thatgamecompany

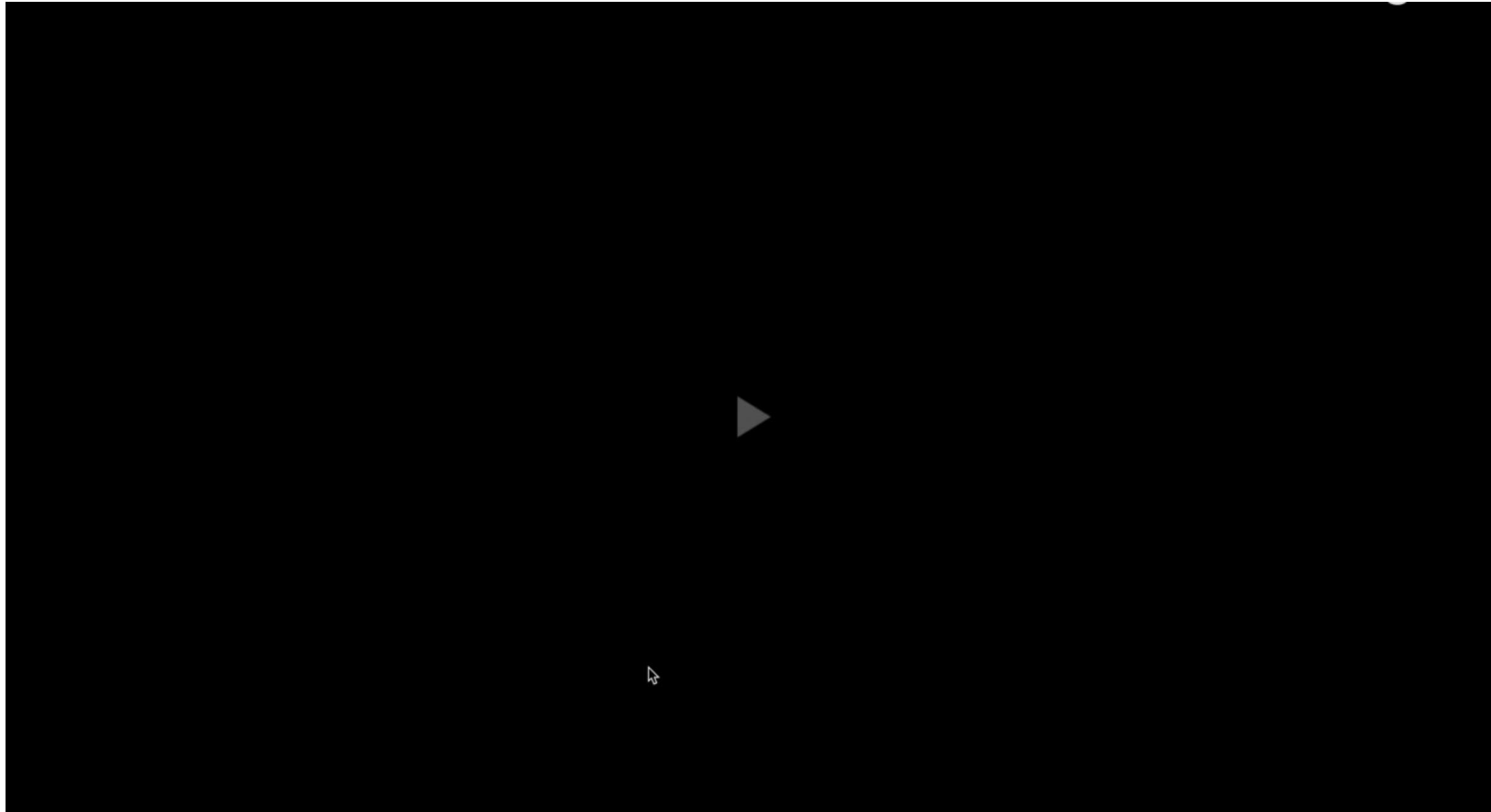


Rez Infinite - United Game Artists

Brainstorming



Inspirations



Vidéo exemple d'Ori and the Blind Forest

Présence d'une balle avec une traînée lumineuse et un halo autour. Elle se déplace avec légartée dans l'espace.

Jour 1 : Phase d'idéation

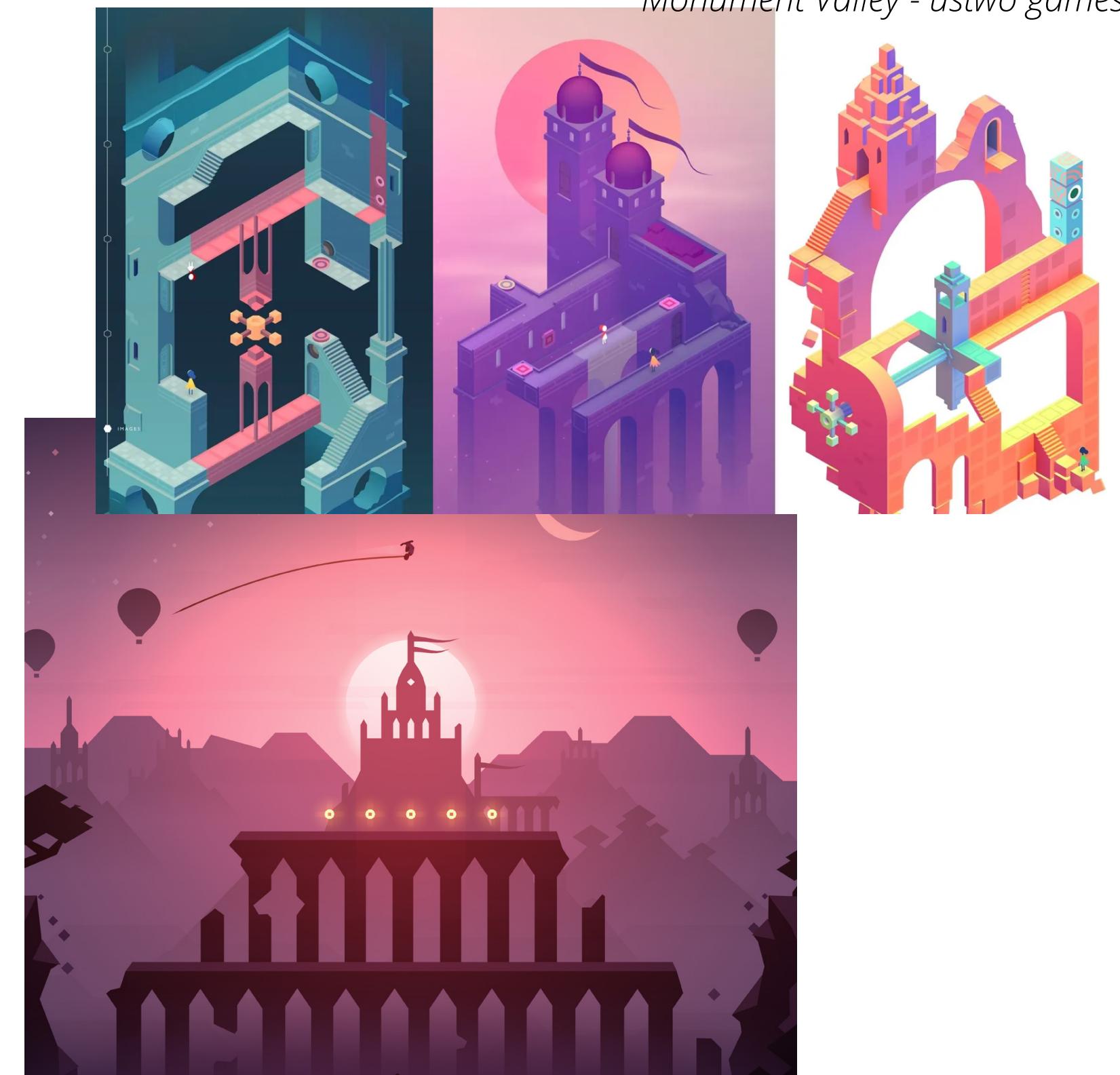
Evolution d'un élément dans un univers

En fonction de l'avancement dans le parcours,
l'ambiance change mais pas le fond.

Les éléments graphiques resteraient **similaires**
mais avec un changement d'ambiance montrant à
l'utilisateur qu'il **évolue**.

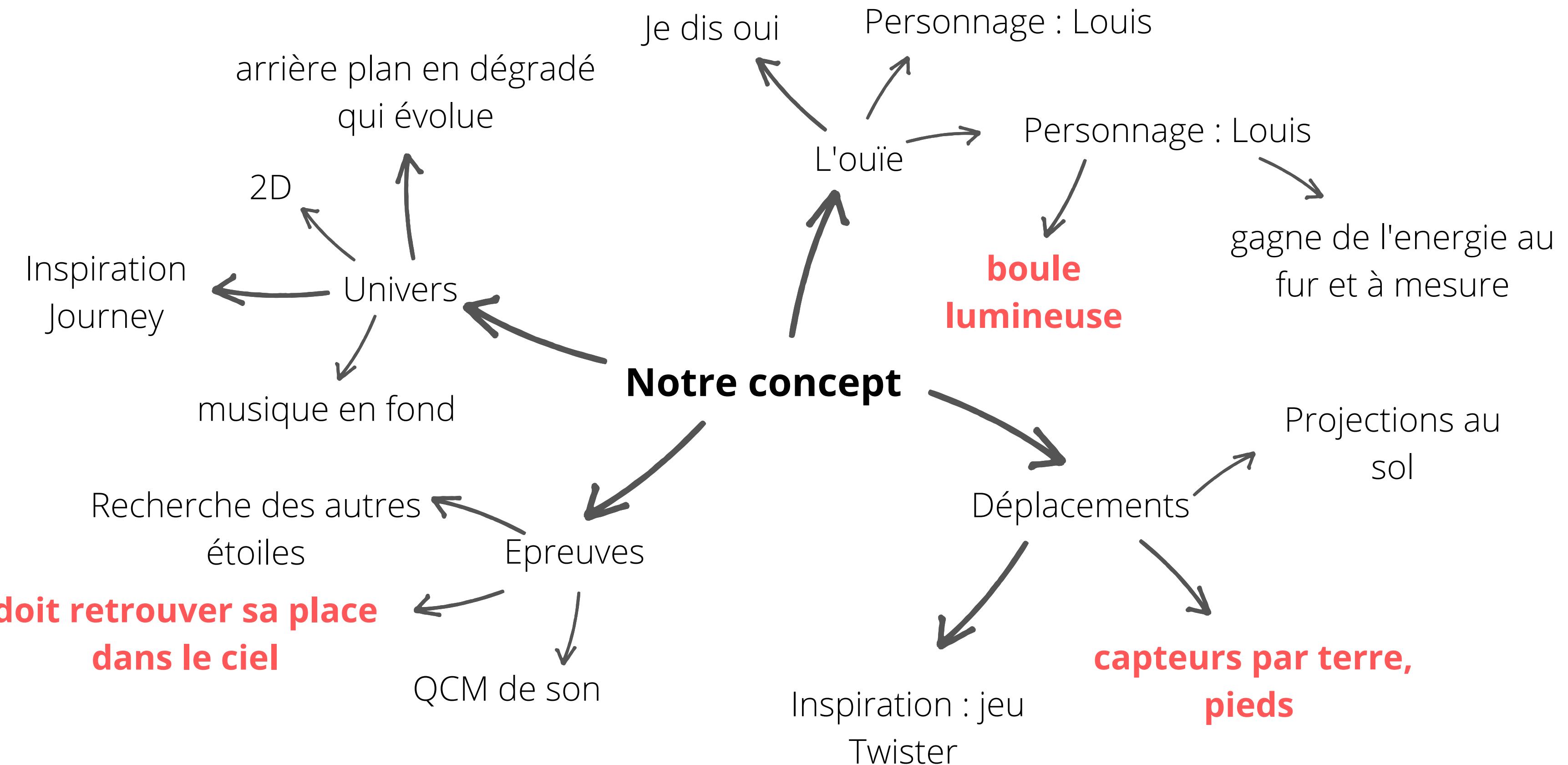


New Super Mario Bros U Deluxe - Nintendo



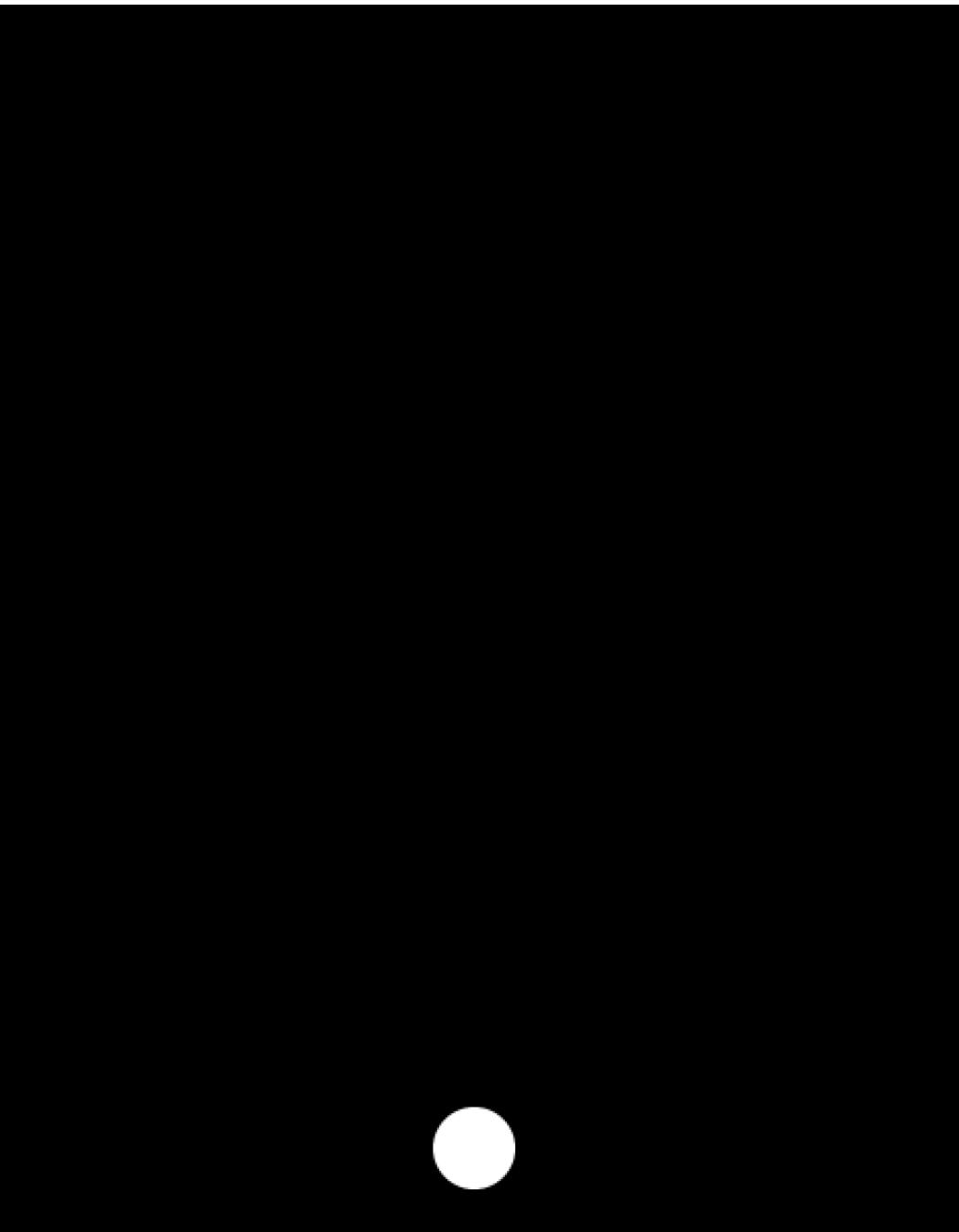
Alto Odyssey - Team Alto

Jour 1 : Phase d'idéation



Jour 1 : Premiers tests de code

Balle en mouvement



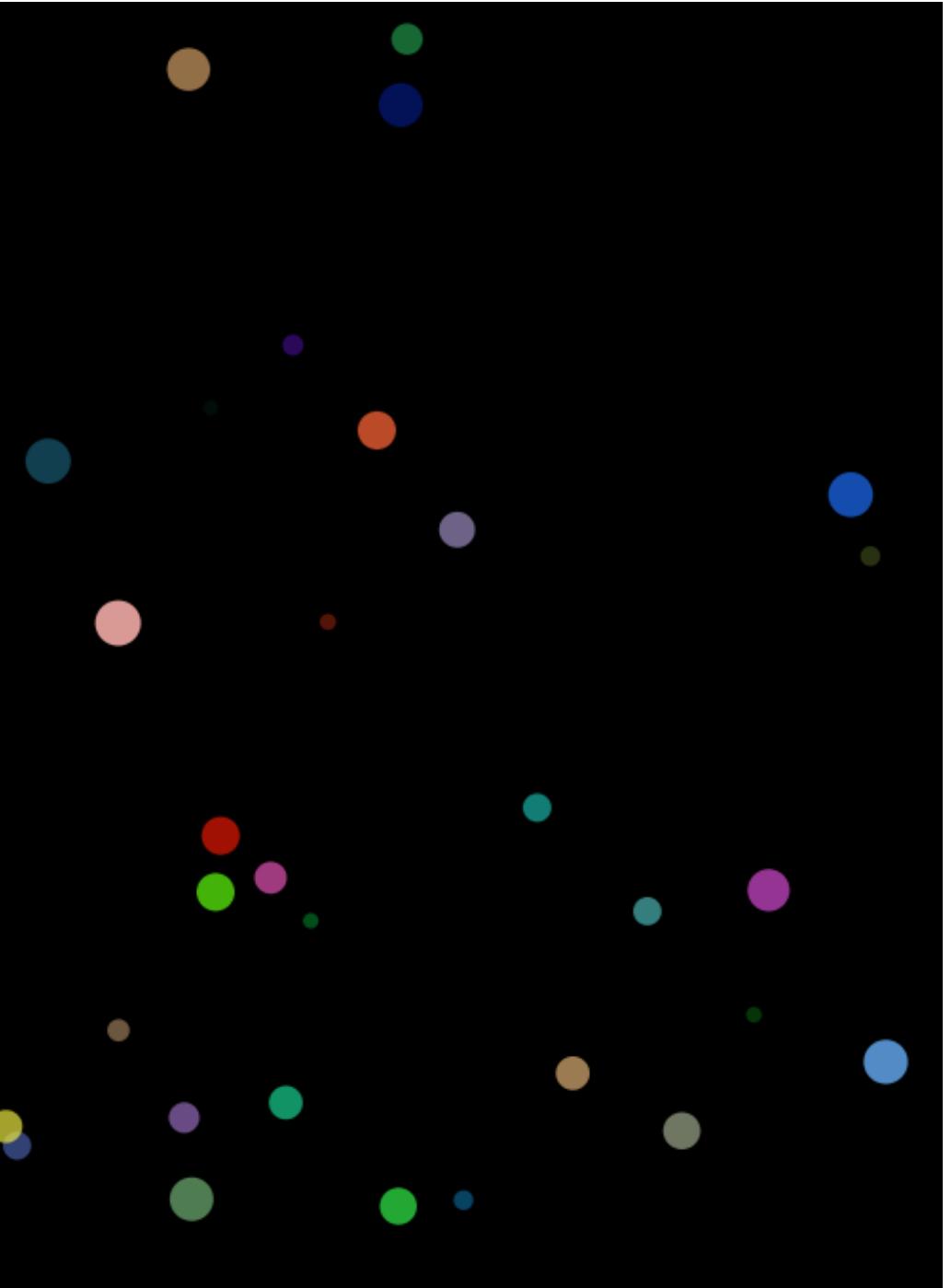
Capture d'écran de l'animation

Code

```
sketch_210104b | ▾  
1 Balle maBalle = new Balle(300, 500, color(255));  
2  
3 void setup() {  
4     size(600, 600); //Taille de la fenêtre  
5 }  
6  
7 void draw() {  
8     background(0);  
9     noStroke();  
10  
11    //Affichage des différents éléments  
12    maBalle.bouge(); //Affichage déplacement de la balle  
13    maBalle.display(); //Affichage de la balle  
14  
15    key=0;  
16 }  
17  
18 class Balle {  
19     float x;  
20     float y;  
21     float vitesseY;  
22     color couleur;  
23  
24     Balle (float nouvX, float nouvY, color nouvCouleur) {  
25         x      = nouvX;  
26         y      = nouvY;  
27         couleur = nouvCouleur;  
28         vitesseY = 2;  
29     }  
30  
31     void display() {  
32         fill(couleur);  
33         ellipse(x, y, 40, 40);  
34     }  
35  
36     void bouge() {  
37         y = y + vitesseY;  
38     }  
39  
40     void keyPressed() {  
41         if (key == CODED) {  
42             if (keyCode == UP) {  
43                 vitesseY=-2;  
44             }  
45         }  
46     }  
47 }  
48 }
```

Jour 1 : Premiers tests de code

Background



Capture d'écran du fond

```
grand_espace | Boule ▾
/*
déplacement dans un espace virtuel plus grand que la surface de la fenêtre
workshop I2, 23 janvier 2019
*/
// Définir les dimensions de l'espace virtuel
int espace_largeur = 5000;
int espace_hauteur = 5000;

// Dimensions de la fenêtre
int fenetre_largeur, fenetre_hauteur;
int origine_x, origine_y; // origine de la fenêtre dans l'espace virtuel

// Définir la structure de donnée qui contient les objets
ArrayList<Boule> boules;
int max_boules = 2000; // nombres de boules

void setup() {
    size(500, 1000);
    fenetre_largeur = width;
    fenetre_hauteur = height;

    origine_x = 2500;
    origine_y = 2500;

    // créer les objets sur l'espace virtuel
    boules = new ArrayList<Boule>();
    for (int i=0; i < max_boules; i++) {
        boules.add(new Boule(espace_largeur, espace_hauteur));
    }
    textSize(10);
}

void draw() {
    background(0);

    // Afficher uniquement les objets qui sont visibles dans la fenêtre
    for (int i=0; i < boules.size(); i++) {
        Boule b = boules.get(i);
        b.deplacer();
        //if (b.visible(origine_x, origine_y, fenetre_largeur, fenetre_hauteur));
        b.dessiner(origine_x, origine_y);

        // Afficher quelques infos utiles
        fill(0);
        text("origine de la fenêtre : " + origine_x + " / " + origine_y, 10, 40);
    }

    /* ****
    Contrôle du mouvement au clavier
    **** */
}

void keyPressed() {
    if (key == CODED) {
        if (keyCode == UP) {
            origine_y -= 10;
        }
        if (keyCode == DOWN) {
            origine_y += 10;
        }
    }
}
```

```
grand_espace | Boule ▾
class Boule {
    float x, y; // coordonnées dans l'espace virtuel
    float d; // diamètre de l'objet
    float opacite;
    float vx, vy; // déplacement sur les deux axes
    color c;

    Boule(int _largeur, int _hauteur) {
        x = random(_largeur);
        y = random(_hauteur);
        d = random(8, 24);
        vx = random (-10, 10); //déplacement au hasard
        vy = random (-10, 100);
        opacite = d*10;
        c = color(random (255),random (255), random (255), opacite);
    }

    void deplacer() {
        x+= vx;
        y+= vy;

        if ((x < 0) ||(x> _largeur)) vx = -vx;
        if ((y < 0) ||(y>_hauteur)) vy =-vy;
    }

    // Cet élément apparaît il dans la fenêtre ?
    // _x, _y : origine de la fenêtre d'affichage dans l'espace virtuel
    // _l, _h : largeur et hauteur de la fenêtre d'affichage
    boolean visible(int _x, int _y, int _l, int _h) {
        if ( (x >= _x) && (x < (_x + _l)) && (y >= _y) && (y < (_y + _h)) ) {
            return true;
        } else return false;
    }

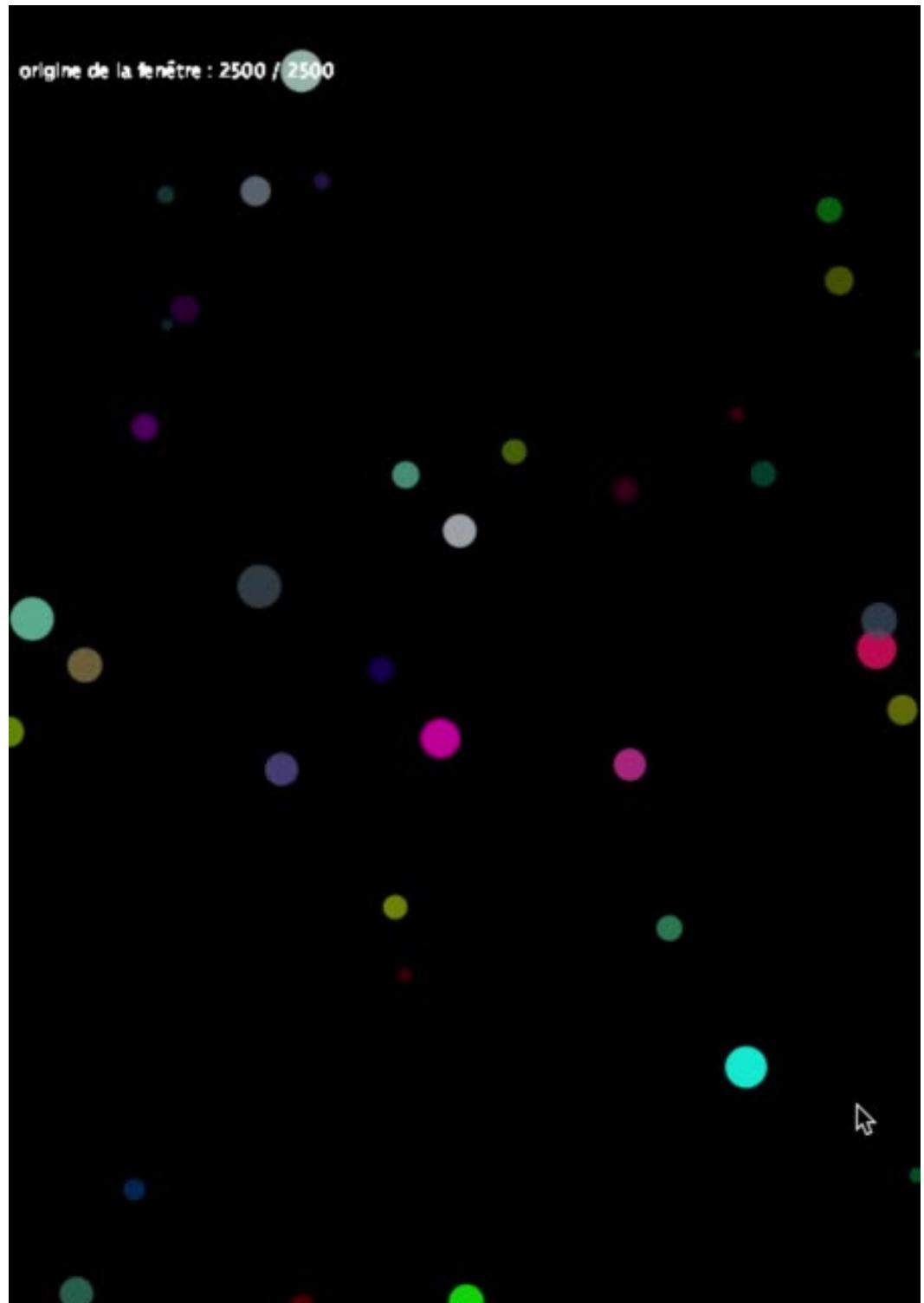
    // Afficher l'objet en fonction de la position de l'origine de la fenêtre d'
    void dessiner(int _x, int _y) {
        noStroke();
        fill (c);
        ellipse(x - _x, y - _y, d, d);
    }
}
```

Code

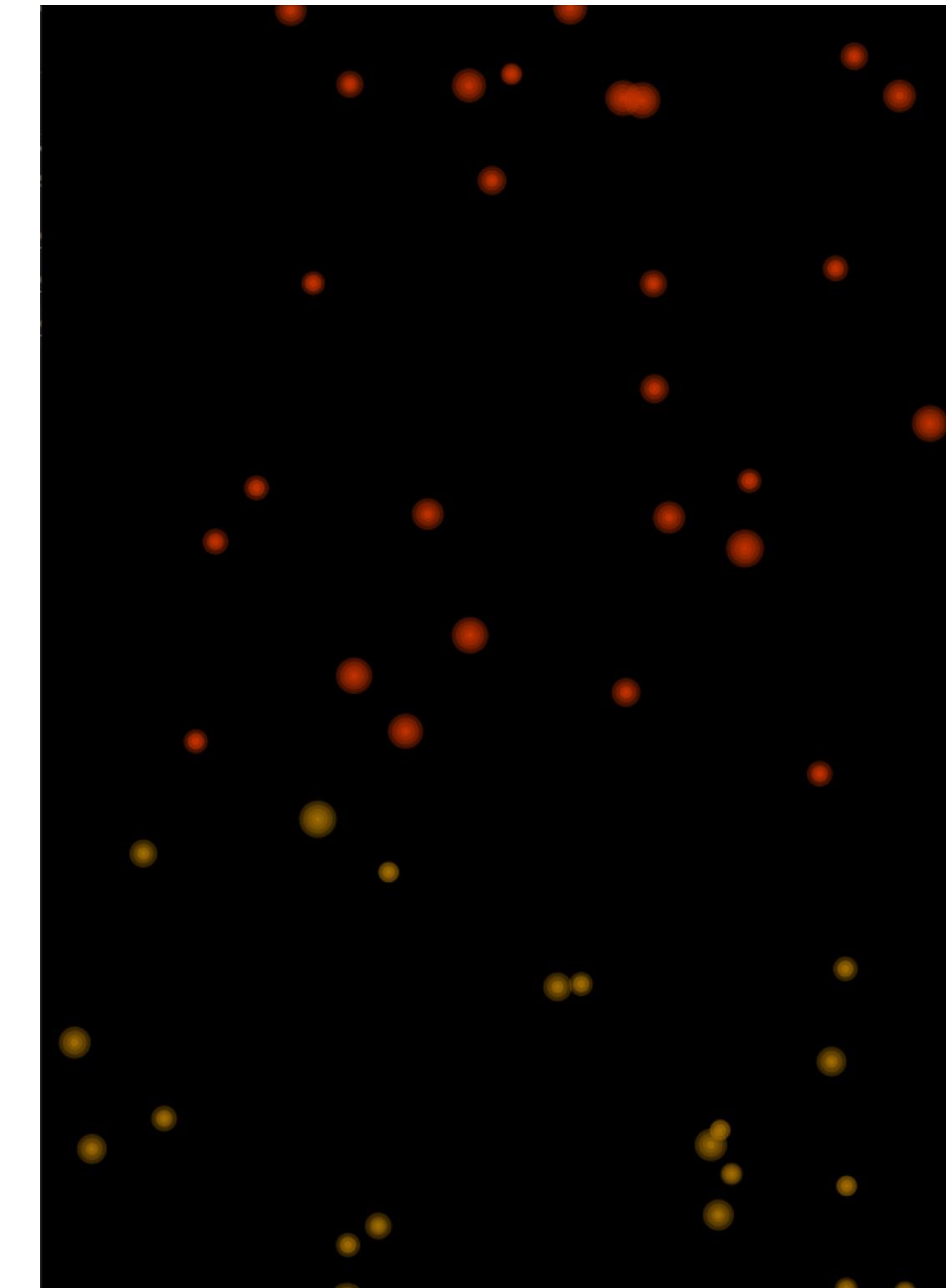
Jour 2 : Affinage des pistes et réalisation du décor

Jour 2: Afinage des pistes

Background

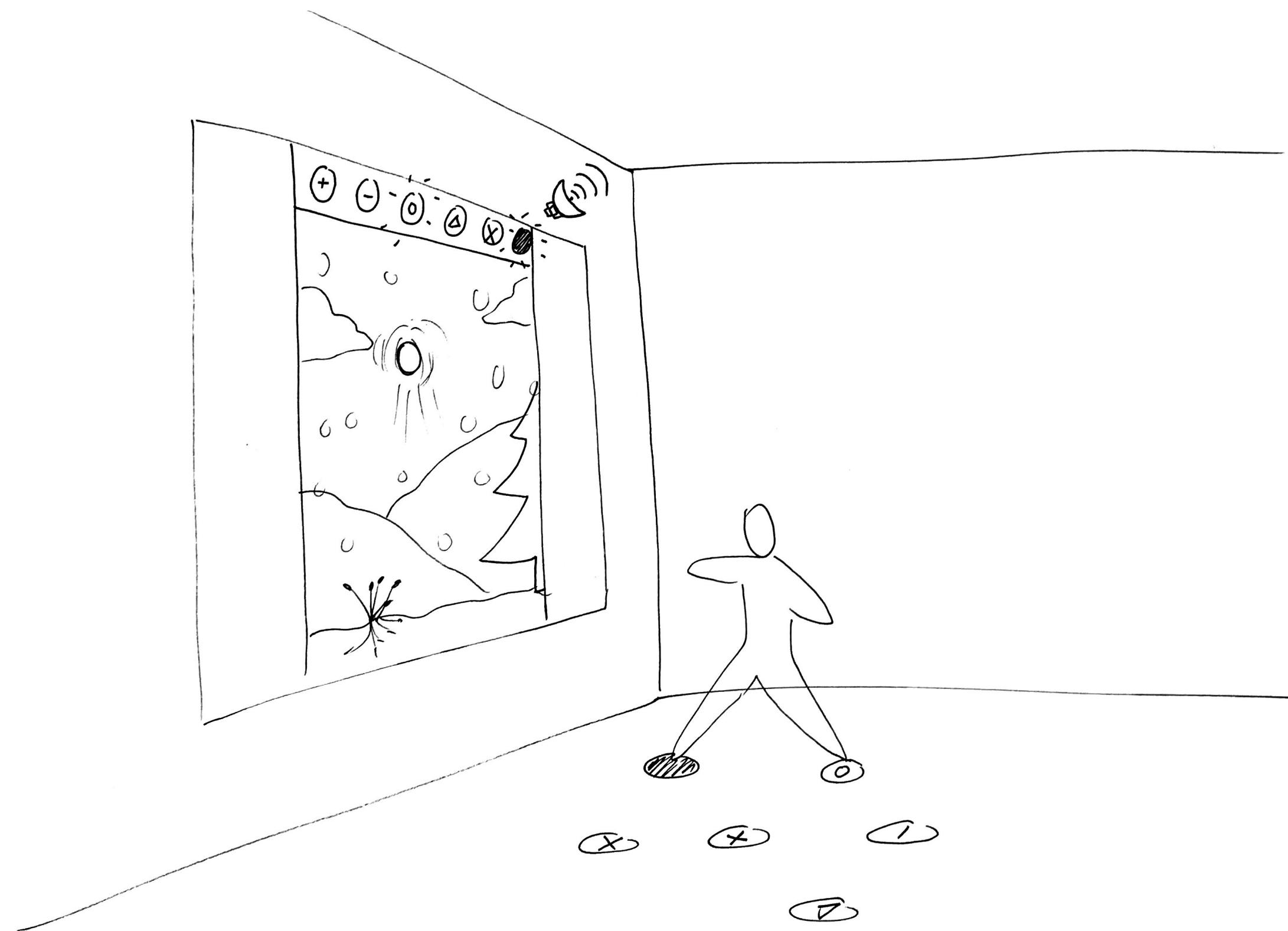


Les balles colorées flottent et se déplacent lentement dans l'espace



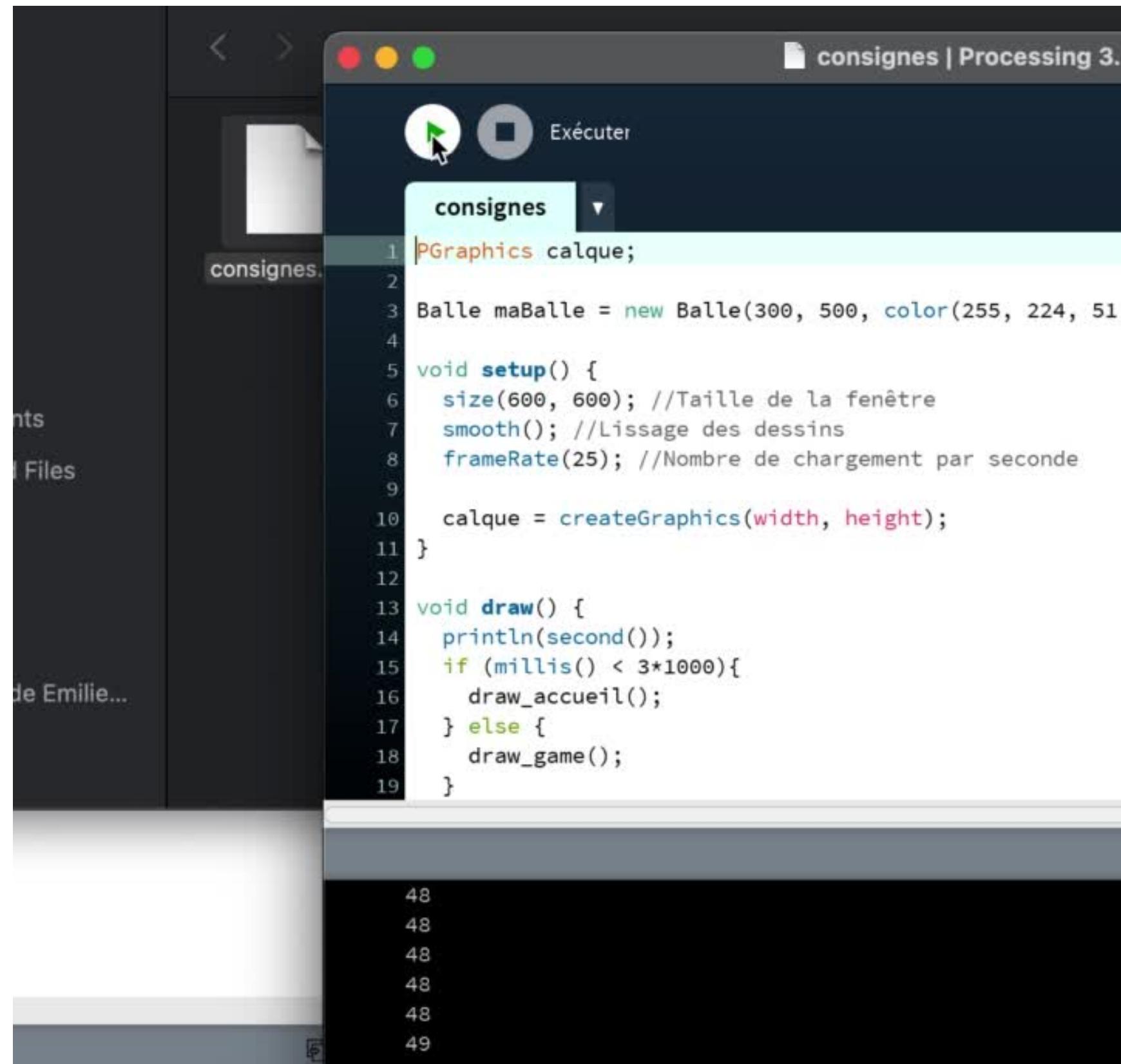
Boules floues et dégradé

Schéma explicatif



Balle en mouvement

La balle est présente dès la page de consigne puis une fois que les consignes disparaissent, la balle se met en mouvement suite à l'activation de la touche UP.



```
consignes
PGraphics calque;

Balle maBalle = new Balle(300, 500, color(255, 224, 51),

void setup() {
    size(600, 600); //Taille de la fenêtre
    smooth(); //Lissage des dessins
    frameRate(25); //Nombre de chargement par seconde

    calque = createGraphics(width, height);
}

void draw() {
    println(second());
    if (millis() < 3*1000){
        draw_accueil();
    } else {
        draw_game();
    }
}
```

Jour 2: Afinage des pistes

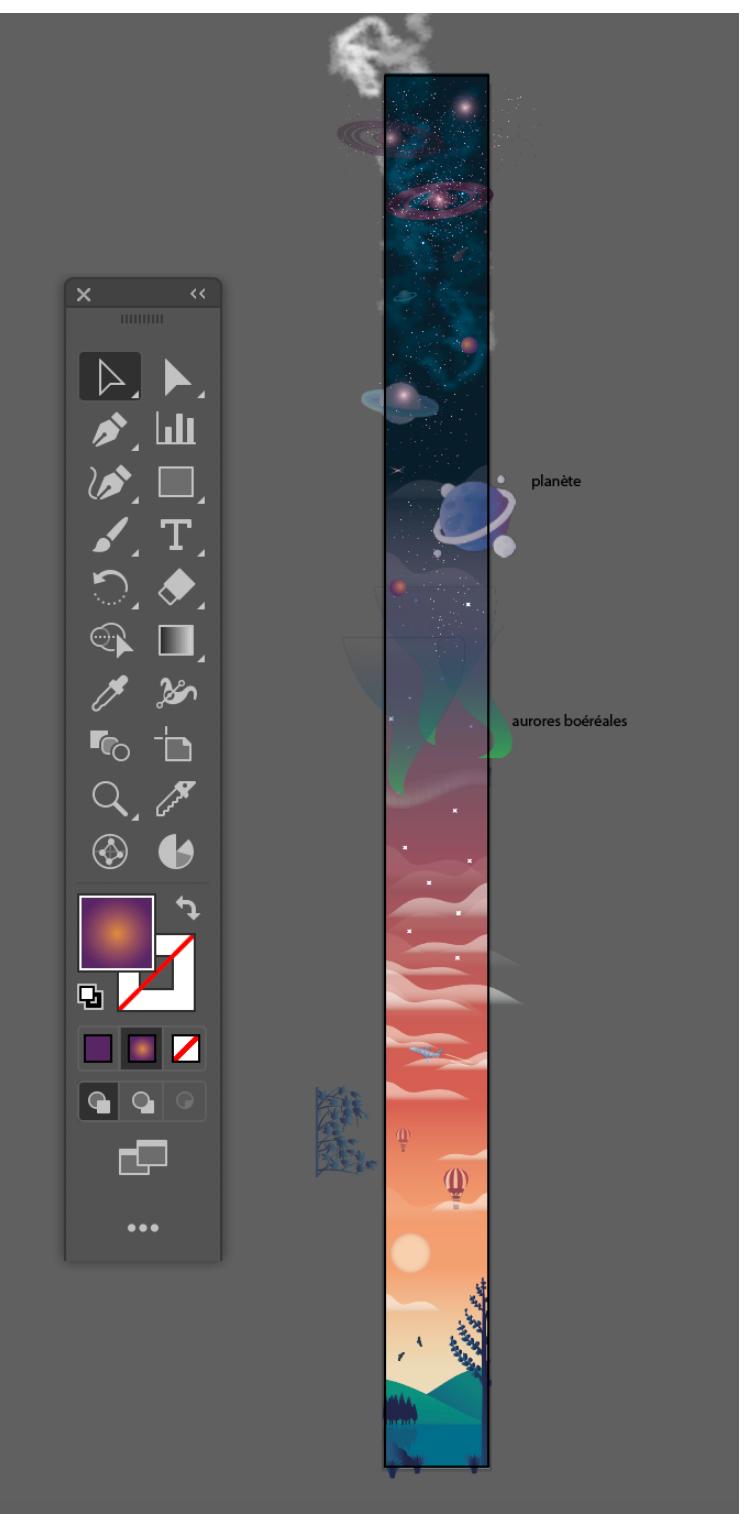
Travail sur la forme de l'étoile

Nous avons voulu retravailler la forme de l'étoile pour qu'elle soit plus en mouvement.

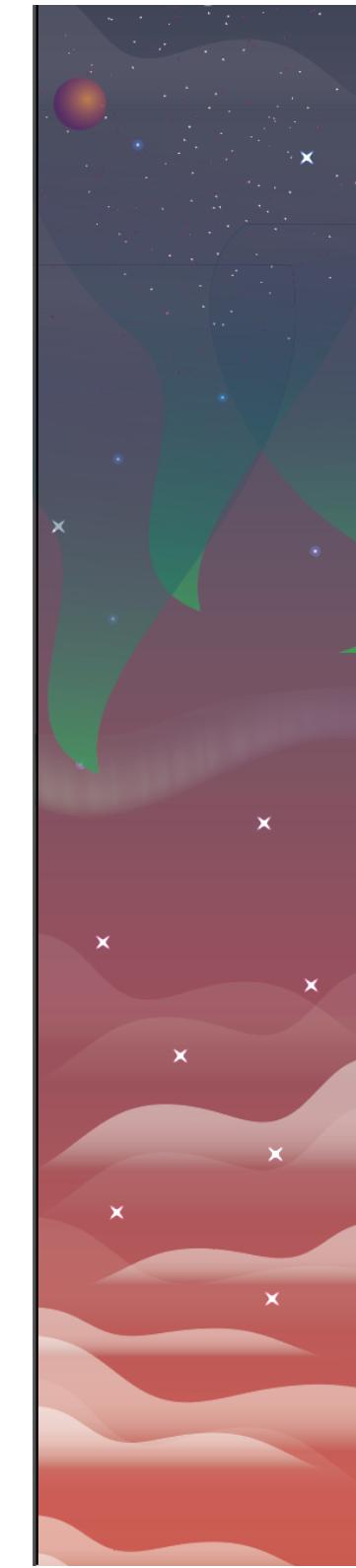


Jour 2: Afinage des pistes

Réalisation du décor



Réalisation du décor sur Illustrator

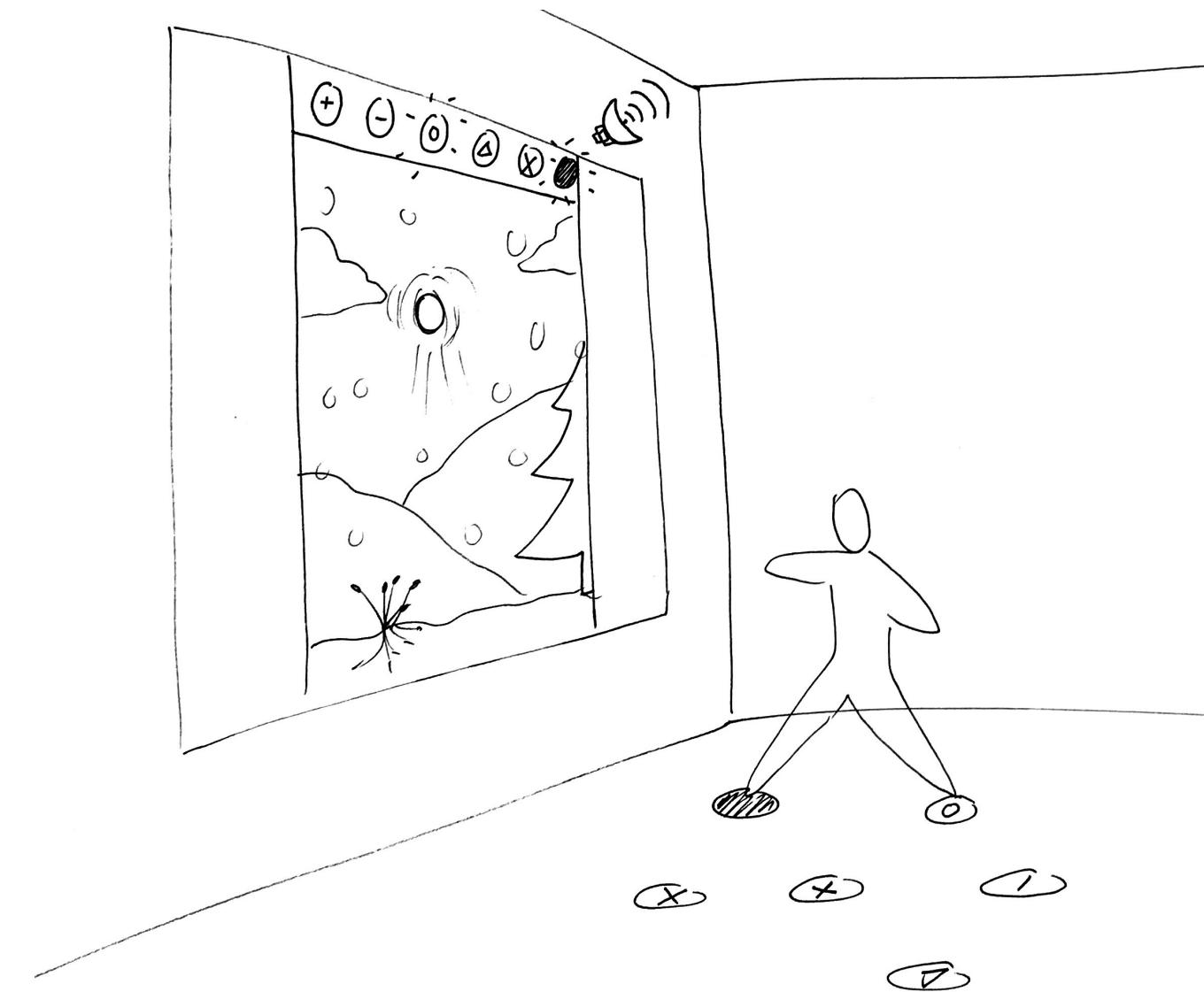


Jour 3 : Validation du projet, choix de sons et tests

Jour 3: Validation du projet

Création de l'interaction entre les pastilles qui s'allument et les boutons sur lesquels le joueur doit appuyer.

Lorsqu'une pastille s'allume à l'écran, le joueur dispose de X temps pour appuyer sur le bouton qui correspond au sol.
S'il met trop de temps, Louis chute.



Inspiration

Jeu *Simon* de Hasbro

But: reproduire la combinaison donnée par la machine



Choix de la musique de fond

"Schizo" de Hicham Chahidi

C'est une musique calme avec des notes qui se démarquent .

Elle évoque la légèreté et correspond à l'avancée de Louis qui gravit peu à peu les différents niveaux de l'univers.

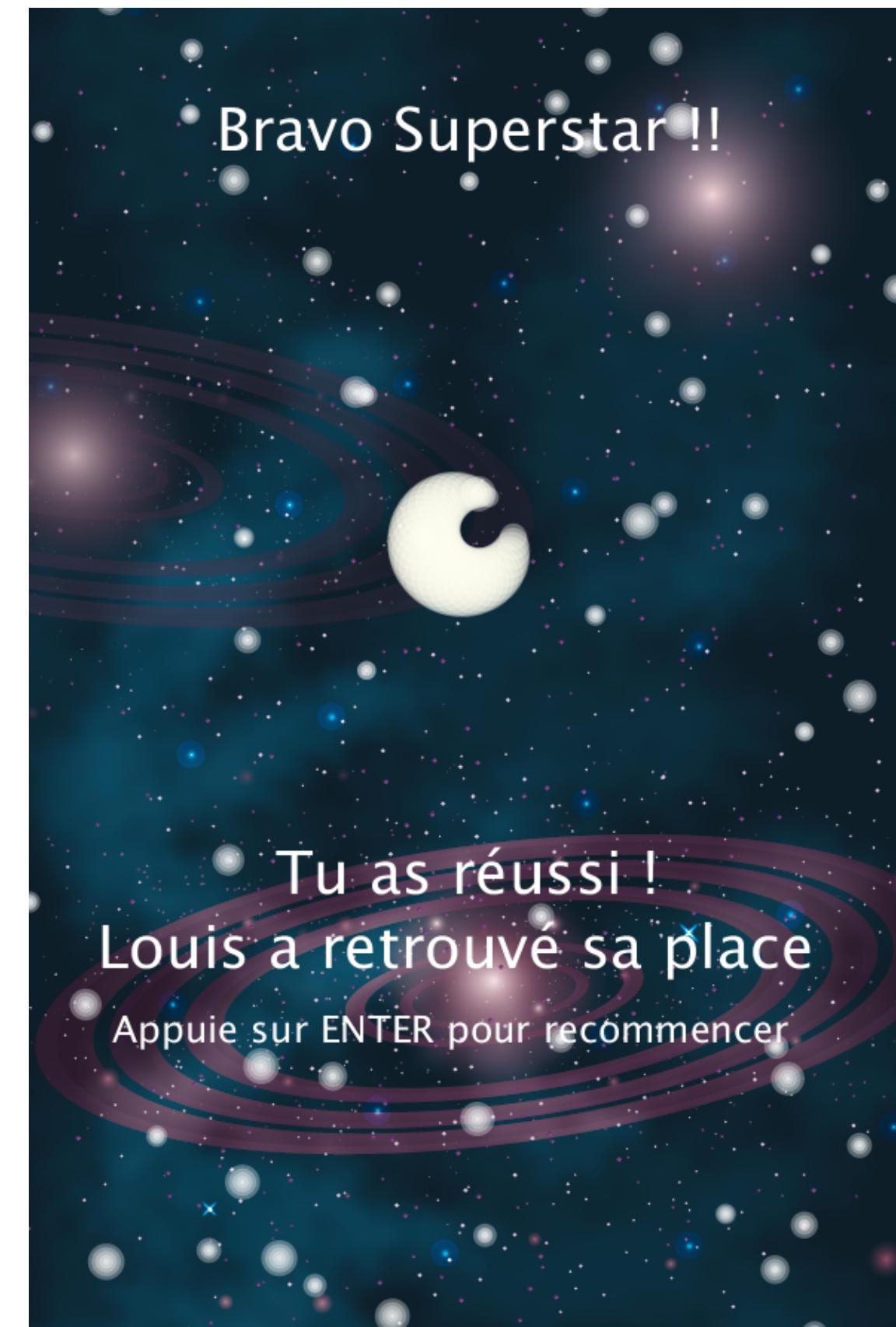


Jour 3: Début et fin

Animations de début et de fin

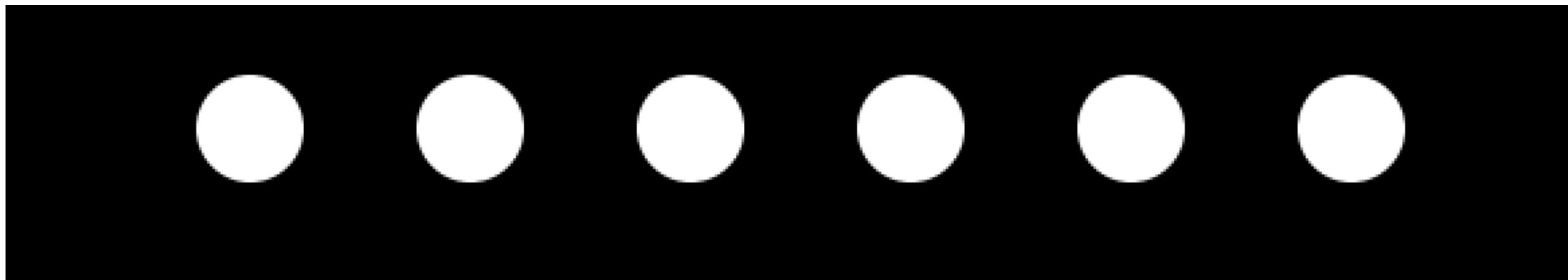


Animation de début

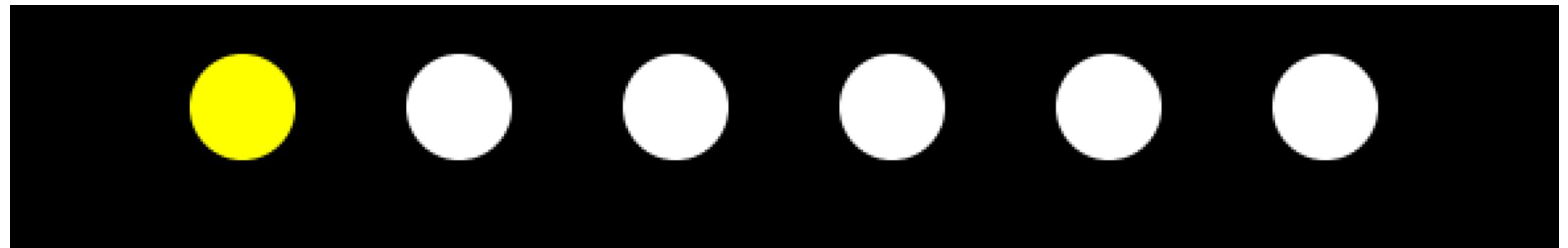


Animation de fin

Jour 3: Tests interaction



Nous avions commencé par codé 6 boutons dont l'un s'allumait lorsqu'on appuyait sur la touche UP.



Jour 3: Tests interaction

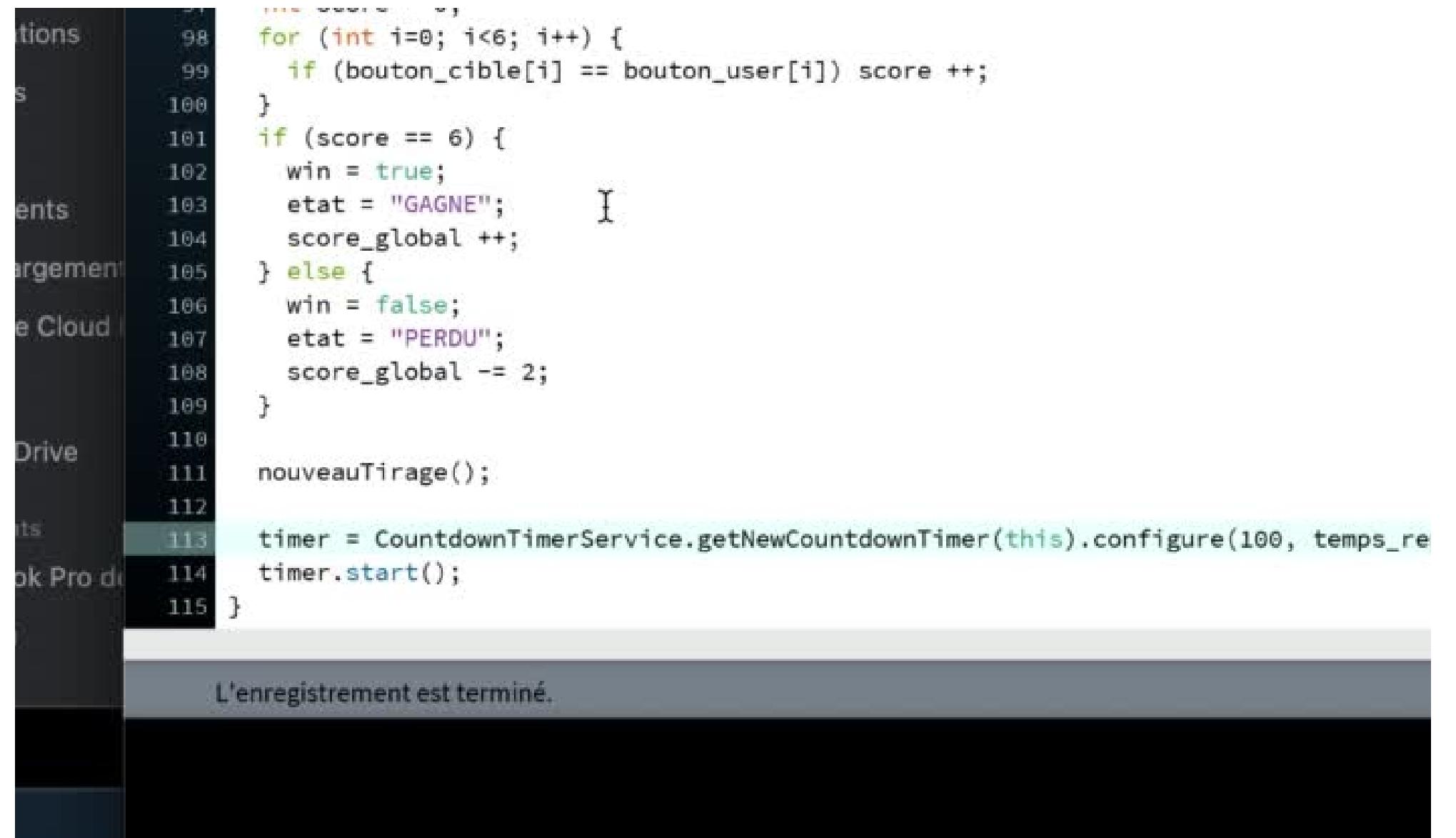
Les pastilles s'allument et laissent X temps au joueur pour appuyer sur la bonne touche du clavier.

Tous les 10 points, une touche allumée se rajoute (jusqu'à 4 boutons maximum) :

10pts > 2 boutons

20 pts > 3 boutons

Après des tests, nous avons diminué le score à atteindre pour avoir plusieurs boutons

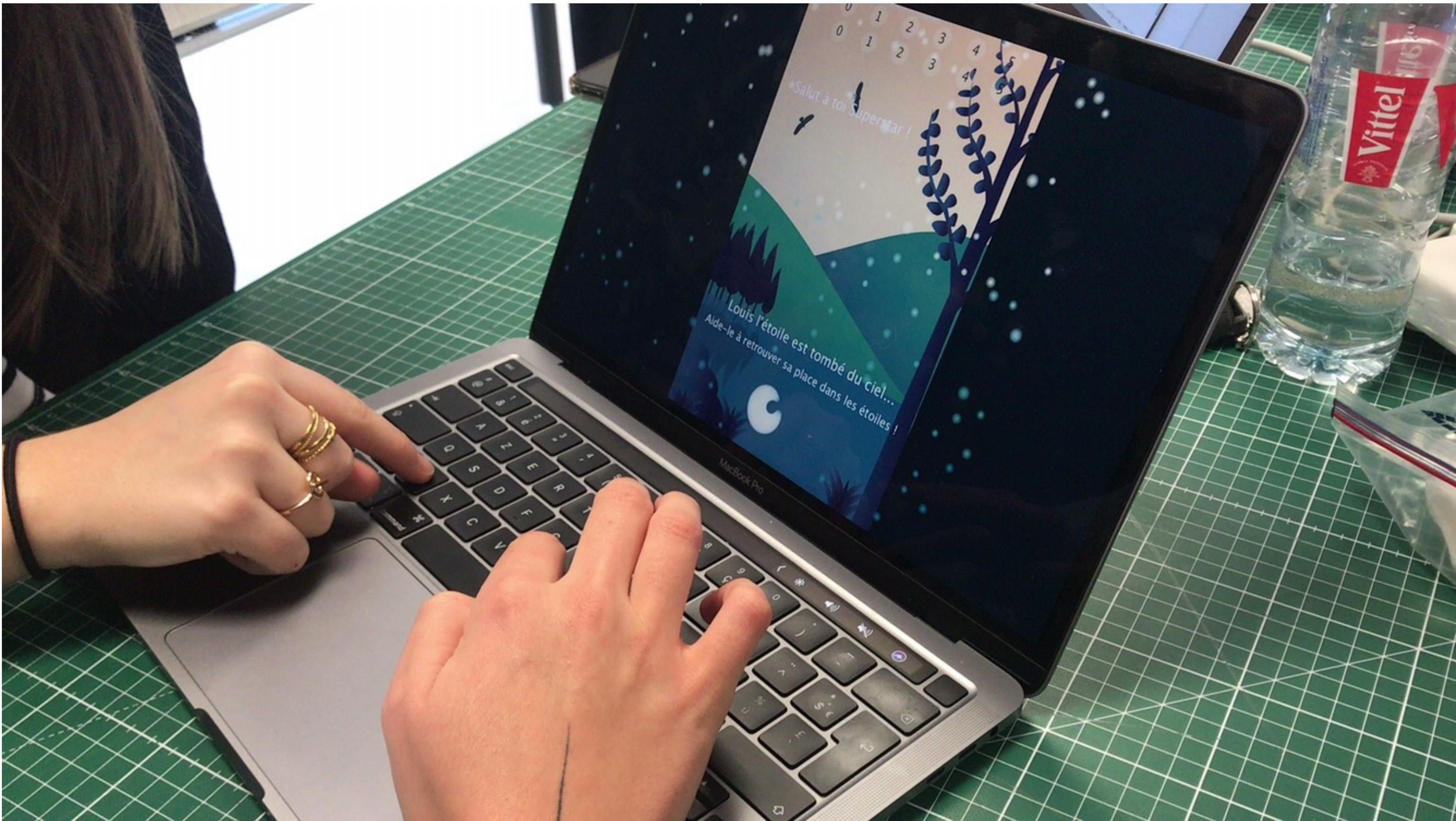


```
    ...
    for (int i=0; i<6; i++) {
        if (bouton_cible[i] == bouton_user[i]) score++;
    }
    if (score == 6) {
        win = true;
        etat = "GAGNE";
        score_global++;
    } else {
        win = false;
        etat = "PERDU";
        score_global -= 2;
    }
    nouveauTirage();
}
timer = CountdownTimerService.getNewCountdownTimer(this).configure(100, temps_re
ok Pro d
    timer.start();
}
```

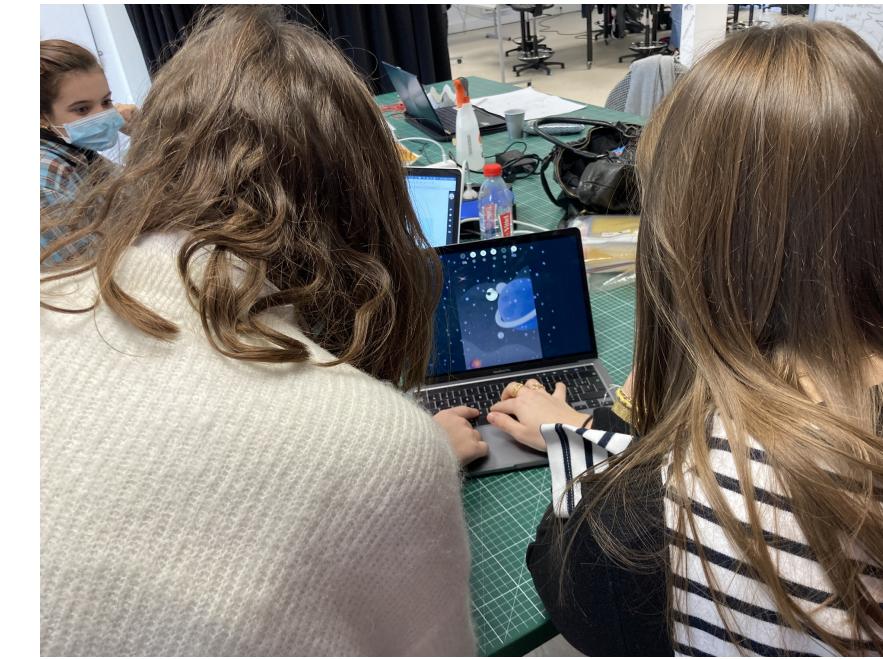
L'enregistrement est terminé.

Jour 4 : Réalisations de tests du jeu, mise en place de
l'installation

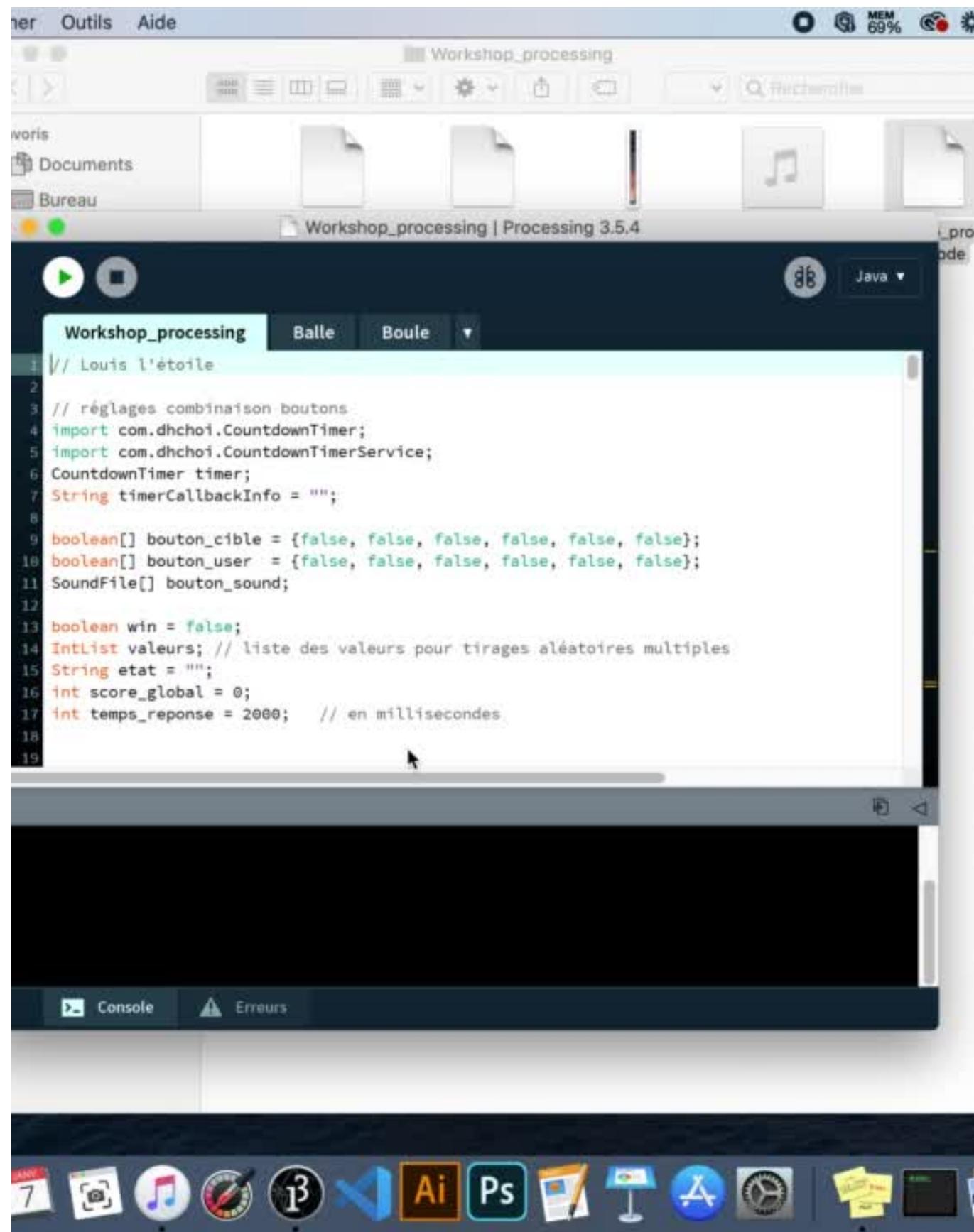
Jour 4: Vidéo test



Jour 4: Tests avec d'autres utilisateurs extérieurs au projet

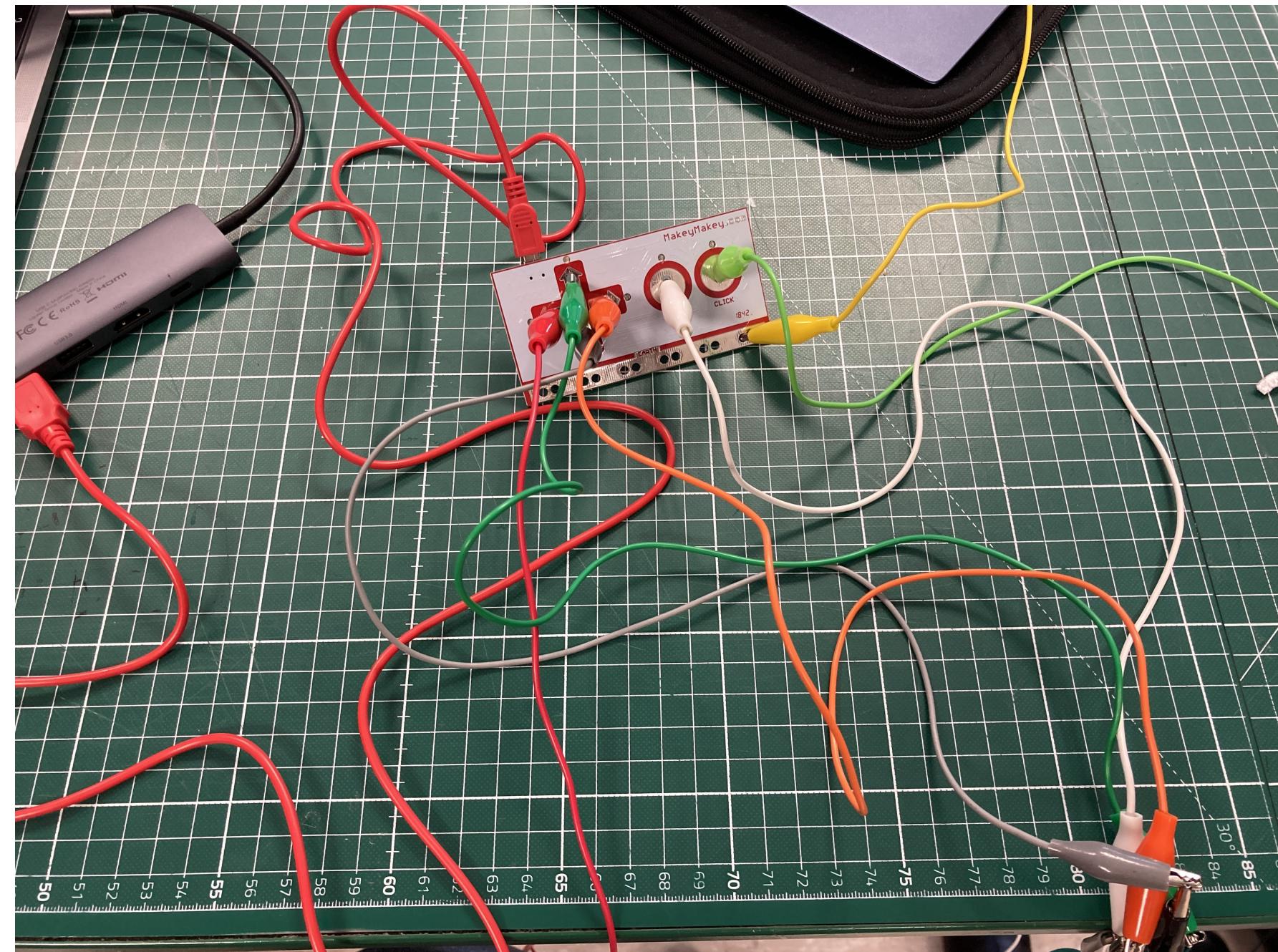


Jour 4: Vidéo du jeu après toutes modifications



Jour 4 : Tests avec le makey makey

Avant de tester le jeu avec nos pieds, il faut d'abord faire des essais avec les touches du clavier reliées au makey makey



Jour 4 : Liste du matériel nécessaire pour la présentation

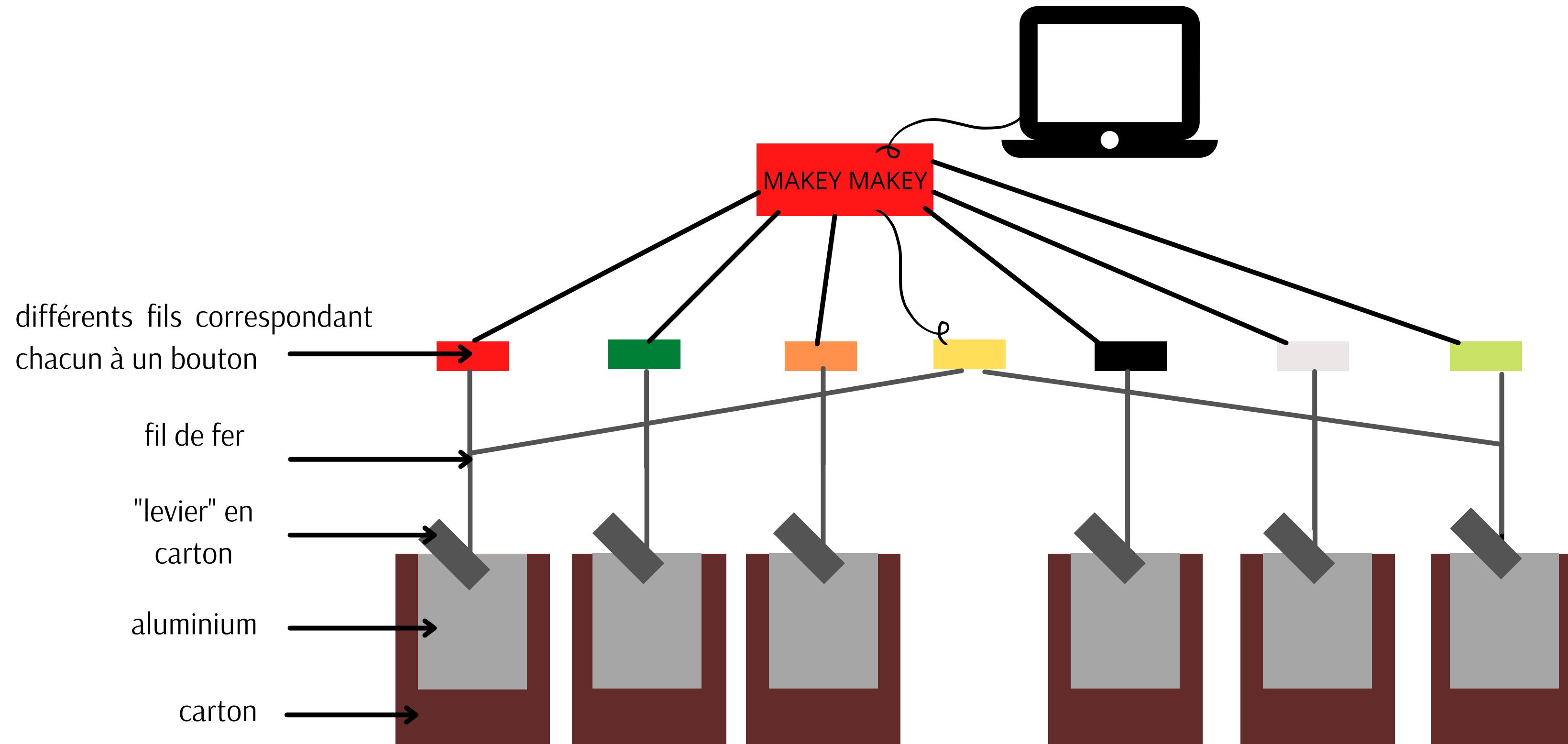
- ordinateur avec le code informatique du projet
- un support sur lequel projeter notre présentation (télé)
- makey makey (avec ses fils et pinces crocos)
- carton
- aluminium
- fil de fer

Jour 4 : Début de la réalisation des plateformes d'appuie

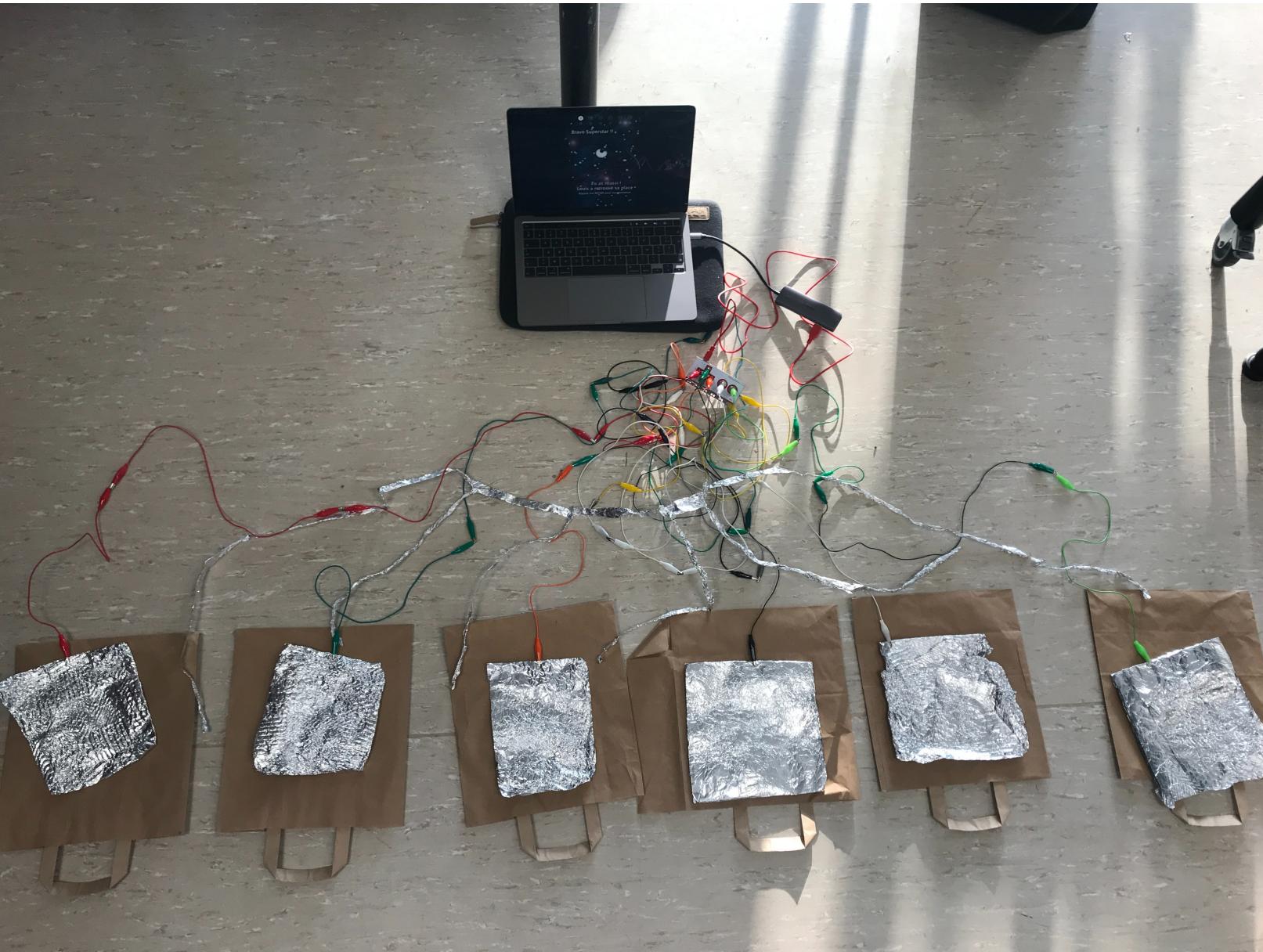


On commence par placer un bout de carton en support puis un bout d'aluminium. Le but est de faire passer le courant en placant un fil du makey makey en dessous pour actionner les boutons

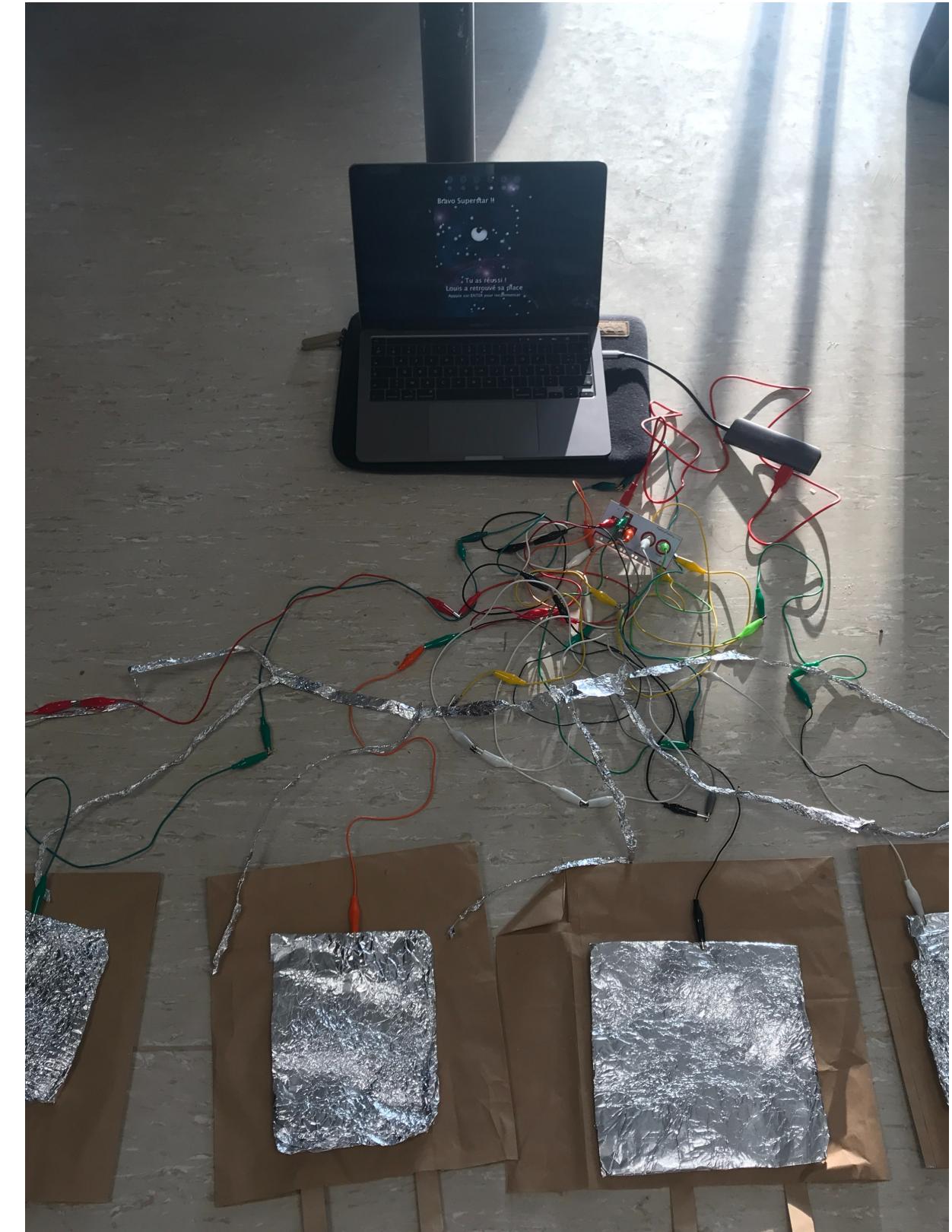
Jour 4 : Schéma de l'installation



Jour 4 : Premier test



Nous faisons es premiers tests avec les mains au lieu
d'appuyer sur les plaques d'aluminium directement
avec les pieds



Jour 4 : Photos de tests de l'installation finale

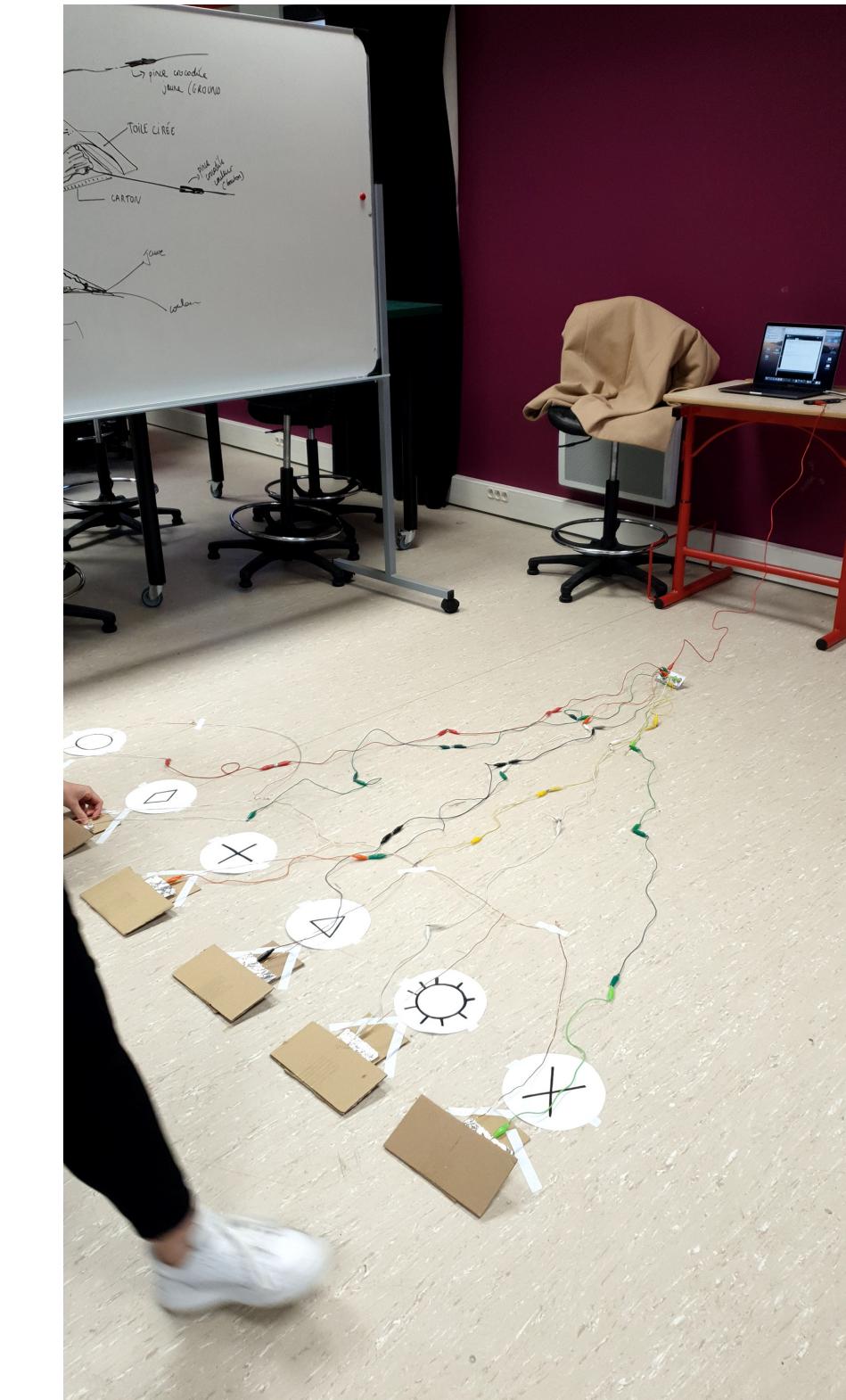


Création d'une pédale en carton



Jour 5 : Rendu du projet final

Jour 5 : Photos de l'installation finale



Nos intentions et remerciements

Nous avons voulu créer un jeu avec un univers propre pour émerger l'utilisateur. Notre but était avant tout de rendre le jeu **convivial**, le fait qu'il n'y ait pas un mais deux à trois utilisateurs à jouer simultanément renforce cet **esprit de groupe**, de plus, Louis est un jeu de **coordination** et de **cohésion** de groupe !

Merci pour votre aide votre gentillesse et votre patience tout au long de cette semaine ! Certaines d'entre nous appréhendions beaucoup ce workshop et pourtant nous sommes ravis de cette semaine et de ce que nous avons produit, nous avons également aimé travaillé avec de nouveaux dispositifs tel que le makey makey qui change par rapport à ce que l'on a l'habitude d'utiliser !

Merci encore !