



Carnet de recherches et
d'avancement de projet

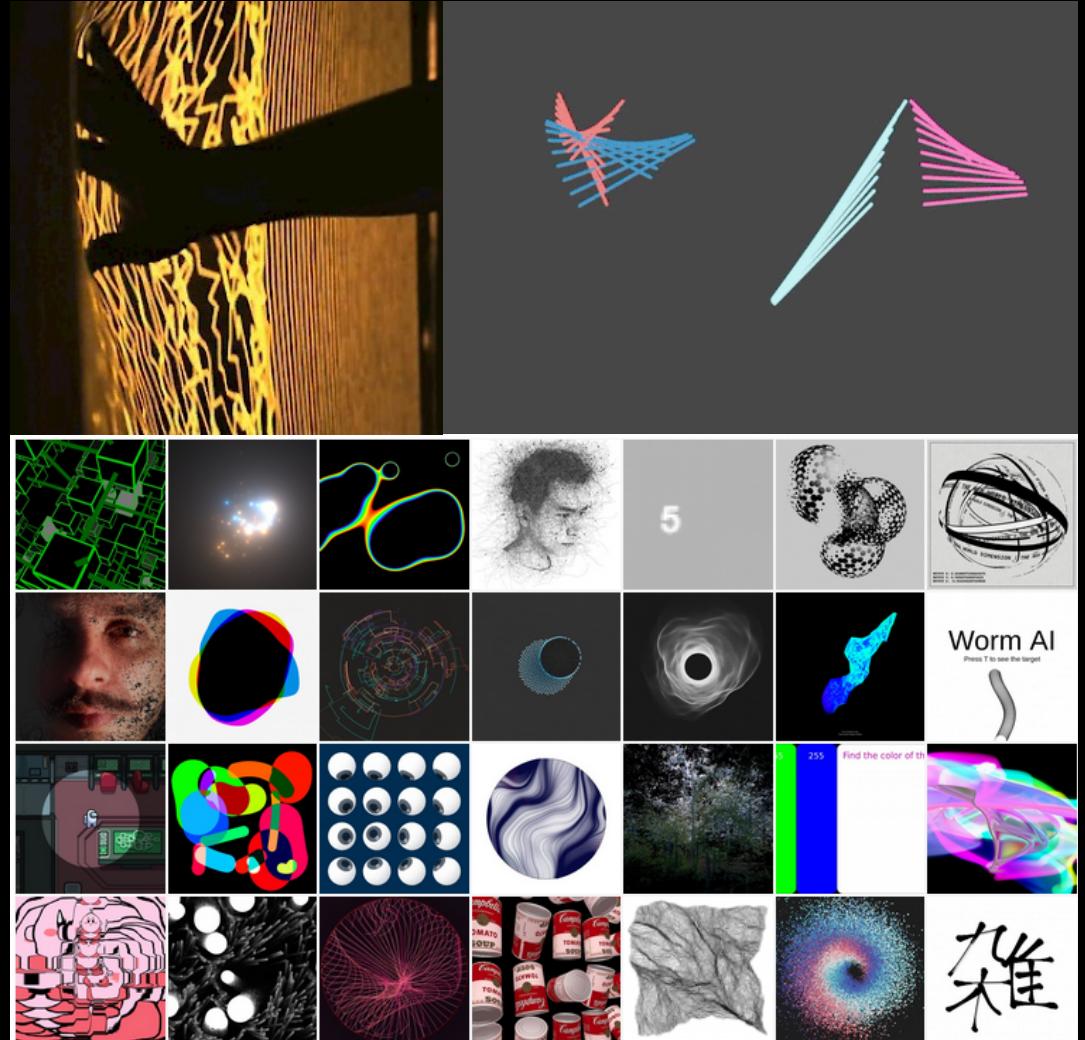
Workshop code créatif
20-21

Lundi

Matin

Recherches

- Prise de connaissance du sujet
- Découverte des inspirations (du brief)
- Les travaux qui m'ont le plus intéressés sont les codes génératif et les mapping interactif pour leurs rendus graphique que je trouve très immersif et captivant.



Approfondissement

- J'ai donc décidé d'après mes recherches de travailler sur une installation interactive avec du mapping qui intégrerais du code génératif... ou en tout cas avoir une notion importante de l'aléatoire dans mon travail.
- J'ai donc réfléchis a quelles systèmes peuvent utiliser des projection et du aléatoire pour créer la liste ci-contre -->

Aléatoire

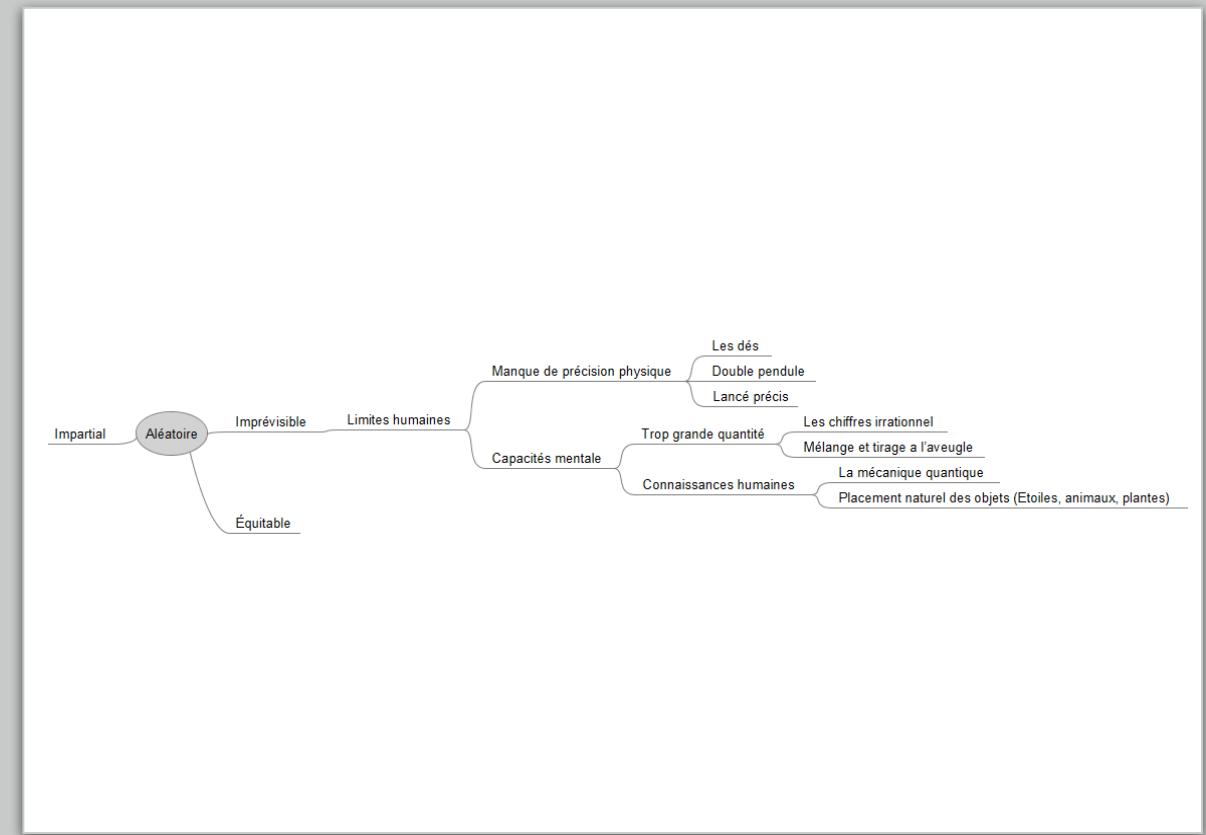
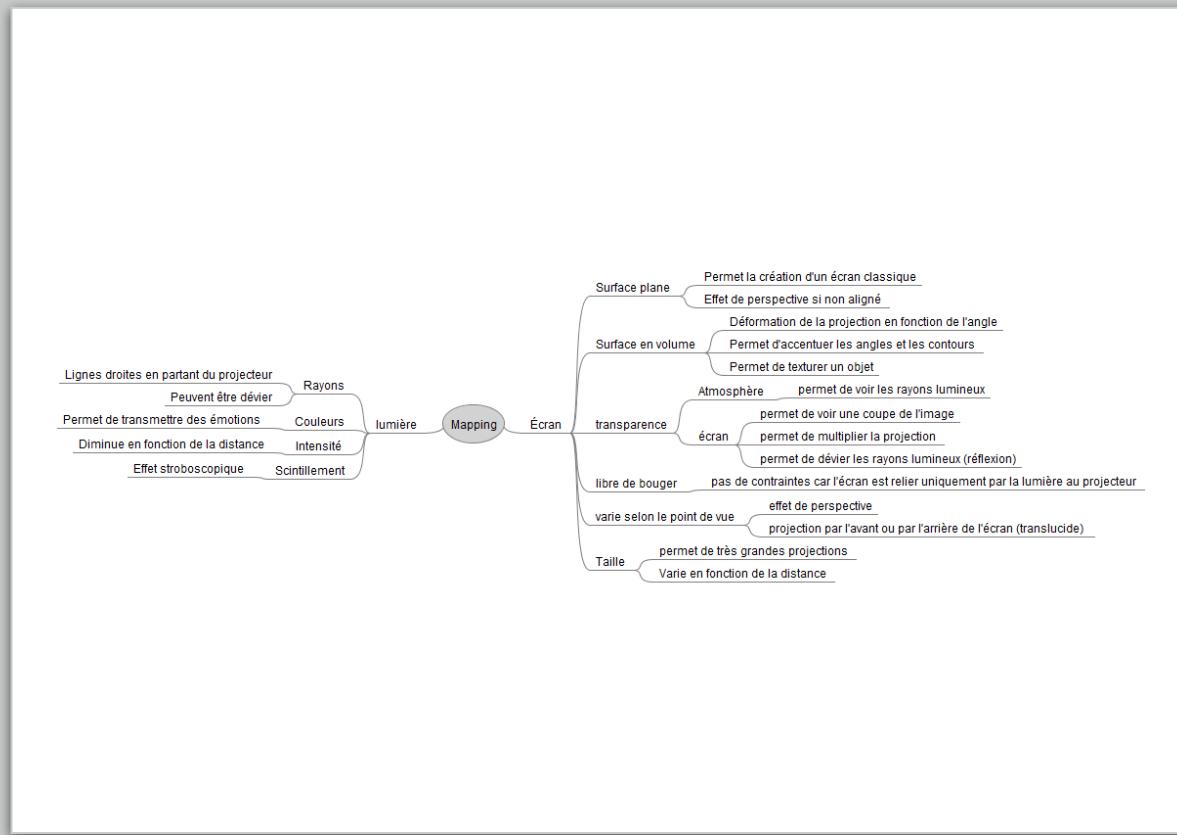
- **Les dés** (aléatoire car très dur de dosé précisément la force de la main qui les lance et les rebonds exactes de l'objet)
- **Les chiffres irrationnel** (écriture décimale ni finie, ni périodique)
- **Mélange et tirage a l'aveugle** (principe du jeux de carte ou tirage au sort)
- **Systèmes chaotiques** (Système tellement instable que la précision humaine ne permet pas de prévoir sa réaction. Exemple : double pendule)
- **La mécanique quantique** (Système physique encore incompris par l'homme qui semble être aléatoire)

Projections (mapping)

- **Cinéma** (permet une taille et qualité d'écran très grande avec un faible prix par rapport aux écrans)
- **Présentations** (diaporamas / tableaux géants)
- **Evènement musicaux** (permet des décors relativement simple a mettre en place et a animer)
- **Installations immersives** (support différent et possibilité de jouer avec les rayons lumineux avant qu'ils arrivent sur l'écran. En faisant passer quelque chose devant par exemple)
- **Éclairage de bâtiments** (précision et animation de l'éclairage et / ou projection de publicités)

Brainstorming

- Après ces recherches je me suis rendu compte que plusieurs aspects ressortait souvent j'ai donc fait des minde maps pour organiser mes idées et voir ce qui en ressort :



Choix

- L'idée que l'on puisse modifier l'image avant quelle arrive sur l'écran m'as beaucoup plus j'ai donc pris un papier et un crayon pour chercher différents moyen de déformer l'image de façons plus ou moins aléatoire.

Lundi

Après-midi

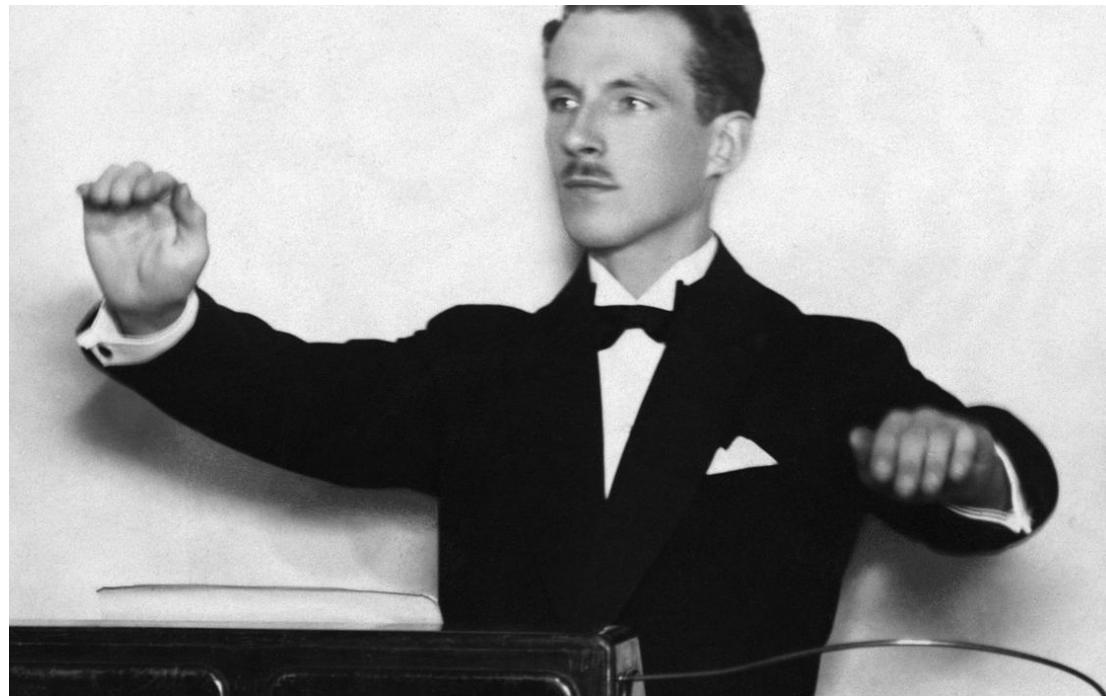
Recherches d'idées et croquis

Mardi

Matin

Instrument contrôlable avec une caméra

On décide donc de reprendre cette idée en la développant en contrôlant mes synthétiseurs





• Matériel

- Korg EMX
- Korg Minilogue XD
- Behringer TD-3

Mardi

Après-midi



Le MIDI

- Musical Instrument Digital Interface

Mercredi

Matin

The MidiBus

Plug-in processing



```
Available MIDI Devices:  
-----Input-----  
[0] "minilogue xd 1 MIDI IN"  
[1] "minilogue xd 1 KBD/KNOB"  
[2] "Steinberg UR22mkII -1"  
[3] "Real Time Sequencer"  
-----Output-----  
[0] "Gervill"  
[1] "Microsoft MIDI Mapper"  
[2] "Microsoft GS Wavetable Synth"  
[3] "minilogue xd 1 MIDI OUT"  
[4] "minilogue xd 1 SOUND"  
[5] "Steinberg UR22mkII -1"  
[6] "Real Time Sequencer"
```

OscP5 0.9.9 infos, comments, questions at <http://www.soja>

Mercredi

Après-midi

Réalisation du code :

Pour la réalisation du code j'ai récupérer le code d'un tutoriel youtube que j'ai ensuite adapter.

Tutoriel youtube :

Live Music Visualization in Processing
https://www.youtube.com/watch?v=3_tjd4CjlH8&ab_channel=ThomasCastleman

```
1 import themidibus.*;
2 import java.util.*;
3 import oscP5.*;
4 import netP5.*;
5 
6 OscP5 oscP5;
7 MidiBus bus;
8 MidiBus sendbus;
9 NoteManager nm;
10 
11 float b = 0;
12 float g = 0;
13 float a = 0;
14 
15 int channel = 2;
16 int number = 43;
17 
18 void setup() {
19   size(700,700);
20   frameRate(48);
21   background(0);
22 
23   MidiBus.list();
24   bus = new MidiBus(this, 2, 5);
25   bus = new MidiBus(this, 3, 6);
26   sendbus = new MidiBus(this, 0, 4);
27   nm = new NoteManager();
28   oscP5 = new OscP5(this, 12000);
29 }
30 
31 
```

```
7 class Note {
8 
9   int channel, velocity, pitch; // store the channel, velocity
10  int lifespan; // lifespan of note, in frames
11  boolean isReleased; // whether or not the note has
12  float x, y;
13  float size;
14  color col;
15 
16 
17 // constructor for new Note object
18 Note(int channel_, int pitch_, int velocity_) {
19   this.channel = channel_;
20   this.pitch = pitch_;
21   this.velocity = velocity_;
22   this.lifespan = 7; // temps que reste une note
23   this.isReleased = true;
24 
25   this.x = map(pitch, 20, 70, 0, width); // placer les notes en hauteur
26   this.y = height / 2 + (channel * 20); // placer les notes en largeur
27 
28   this.size = map(velocity * velocity / 100 , 0, 127, 5, 100);
29 
30   this.col = color(pitch*2+20,0,channel*10+20);
31 }
32 
33 // update note properties
34 void update() {
35   void oscEvent(OscMessage theOscMessage) {
36     if (theOscMessage.checkAddrPattern("/orientation/beta")==true) { // transforme l'angle en beta
37       b = theOscMessage.get(0).intValue();
38     }
39     if (theOscMessage.checkAddrPattern("/orientation/gamma")==true) {
40       g = theOscMessage.get(0).floatValue();
41     }
42     if (theOscMessage.checkAddrPattern("/orientation/alpha")==true) {
43       a = theOscMessage.get(0).floatValue();
44     }
45 
46     println("### received an osc message, with address pattern "+theOscMessage.addrPattern());
47   }
48 
49   void controllerChange(int channel, int number, int value) {
50     println("Controller Change: " + channel + ", " + number + ", " + value);
51   }
52 
53   void draw() {
54     background(0,0,random(0,50));
55     fill(255);
56     ellipse(width*.5, height*.5, g*.5, g*.5);
57 
58     nm.track(); // intégrer la note dans la liste
59 
60     int channel = 15;
61 
62     int value = 2*(abs(int(g)));
63     delay(1);
64 
65     sendbus.sendControllerChange(channel, number, value);
66 
67   }
68 
```

```
10 class NoteManager {
11 
12   public ArrayList<Note> notes = new ArrayList<Note>();
13   private ArrayList<Note> notesToAdd = new ArrayList<Note>();
14   private ArrayList<Note> release = new ArrayList<Note>();
15   private ArrayList<Note> notesToRelease = new ArrayList<Note>();
16 
17   // construct a new NoteManager object
18   public NoteManager() {
19 
20   }
21 
22   // add a new note to tracked notes
23   void addNote(Note n) {
24     // add note to list of notes that will be tracked in the next frame
25     notesToAdd.add(n);
26   }
27 
28   // remove a note from tracked notes
29   void releaseNote(Note n) {
30     // add note to list of notes that will be released in the next frame
31     notesToRelease.add(n);
32   }
33 
34   // add new notes to tracked notes, remove old notes from tracked notes
35   void track() {
36     release.addAll(notesToRelease); // add every note we want to release
37     notesToRelease.clear(); // remove everything from list of notes
38   }
39 
```

```
40 // display note on canvas
41 void display() {
42   strokeWeight(0);
43   fill(this.col);
44   ellipse(this.x, this.y, this.size, this.size);
45 
46 }
47 
```

```
75 void noteOn(int channel, int pitch, int velocity) {
76   nm.addNote(new Note(channel, pitch, velocity));
77   println("Note on: " + channel + ", " + pitch + ", " + velocity);
78 }
79 
80 void noteOff(int channel, int pitch, int velocity) {
81   nm.releaseNote(new Note(channel, pitch, velocity));
82   println("Note off: " + channel + ", " + pitch + ", " + velocity);
83 }
```



Midi Note NoteManager ▾

```
1 import themidibus.*;
2 import java.util.*;
3 import oscP5.*;
4 import netP5.*;
5
```

```
0
0
Controller Change: 4, 74, 44
Controller Change: 1, 74, 60
Note on: 9, 42, 125
0
```

```
Note off: 9, 42, 64
Note on: 0, 32, 80
Note off: 0, 33, 64
0
```

```
0
0
Note off: 0, 32, 64
0
0
```

```
Controller Change: 4, 74, 31
Controller Change: 1, 74, 41
0
```

```
Note on: 0, 25, 80
0
0
```

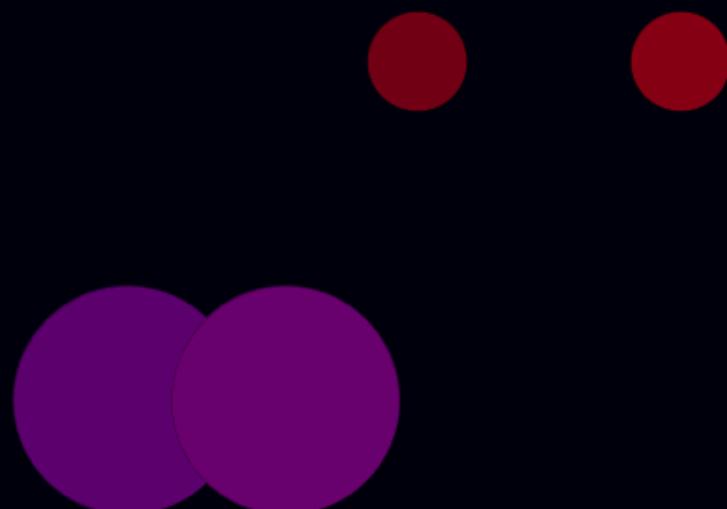
```
Note off: 0, 25, 64
0
0
```

```
Controller Change: 4, 74, 30
Controller Change: 1, 74, 38
0
```

```
Note on: 0, 57, 80
```

Console

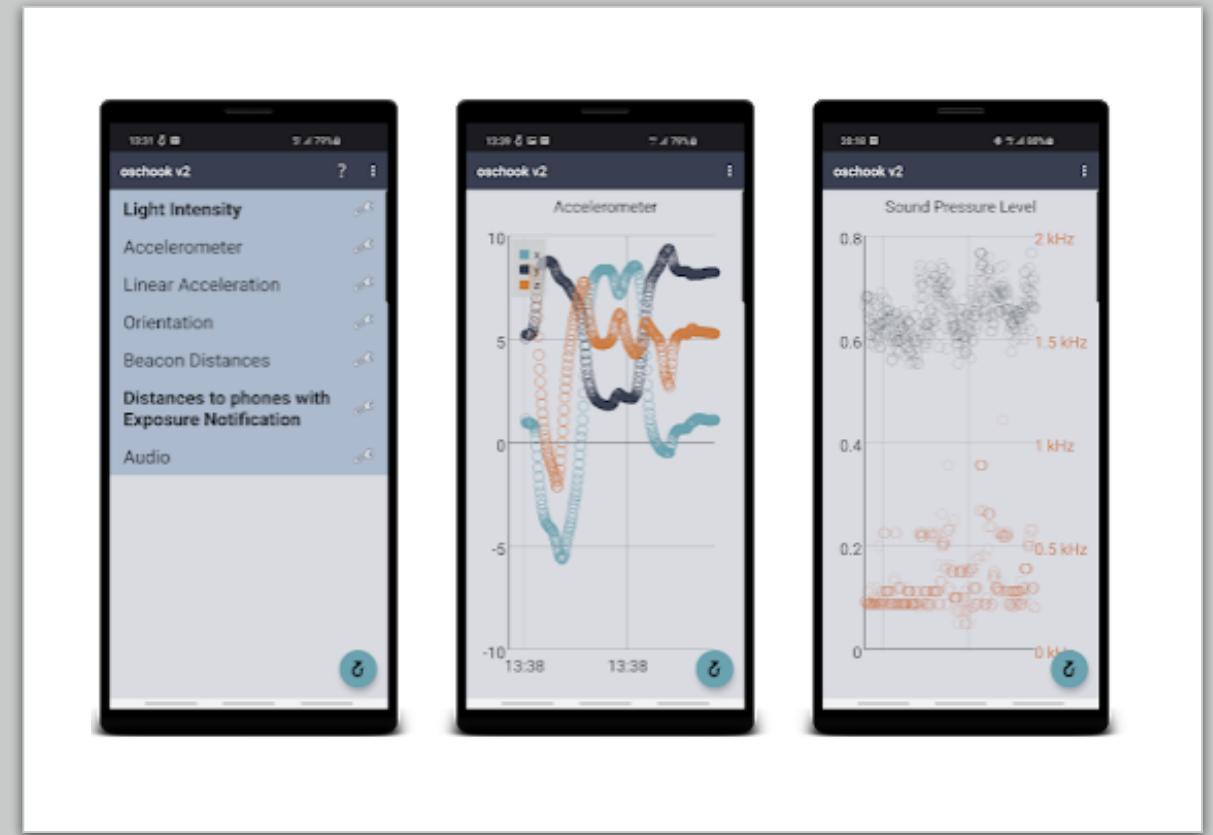
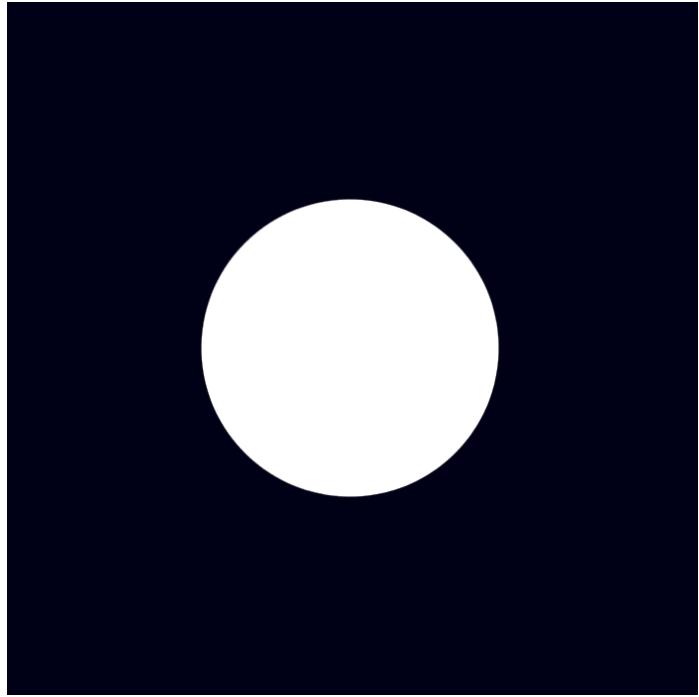
Erreurs



Jeudi

Matin

Possibilité de modification du filtre grâce au gyroscope du téléphone



Jeudi

Après-midi

Possibilité de
modification du filtre
grâce au gyroscope
du téléphone



Automatisation du filtre en fonction de l'inclinaison du
téléphone



Vendredi

Matin

Diapo et passage à l'orale

Vendredi

Après-midi