

Cahier de recherches

Coralie PICARD, Maël JALLAIS,
Martin FOURNY, Enora JAFFRE

I2A



Le Commencement...

Tout à commencer un lundi matin. Nous étions tous très motivés pour ce nouveau workshop Code Créatif. Au moment de faire les groupes Maël, Enora, Martin et Coralie se sont regardés. Tous les quatre très motivés par le fait de faire de la réalité augmenté, et par le fait de travailler ensemble, ont décider de créer un groupe. Une équipe qui étant amis sauraient se supporter malgré les échecs, obstacles et le stress qui augmenterai au fur et à mesure des jours. Et des problèmes et tensions, il y en aura...



Notre méthodologie

Pour un travail efficace il faut une bonne organisation et des techniques de motivation !

Nous avons mis en place un vote à la majorité pour les prises de décision et des listes d'objectifs par jour pour suivre notre avancée. Pour soutenir les plus découragés notre mascotte la petite peluche Totoro nous soutenait dans les méandres de nos lignes de codes. Mais ce qui nous a gardé motivés c'est notre tableau des réussites. A chaque fois qu'on réussit quelque chose, quoi que ce soit, on le marques. Ainsi, même dans les moments de désespoir, toutes les réussites du projet sont sous nos yeux. Et nous rappellent qu'on est capable de grandes choses.



Sommaire

Journée 1

Recherche du sujet	5
Pistes	6
Choix du sujet	7-8
Recherche d'aide pour le code	9
Premiers tests	10-11
Ressources d'aide graphiques	12
Positionnements sur la créature	13
Définition des caractères	14
Recherches de texture	15-18
Réalisation des réactions Tél./PC	19-22
Compte rendu journée 2	23

Journée 3

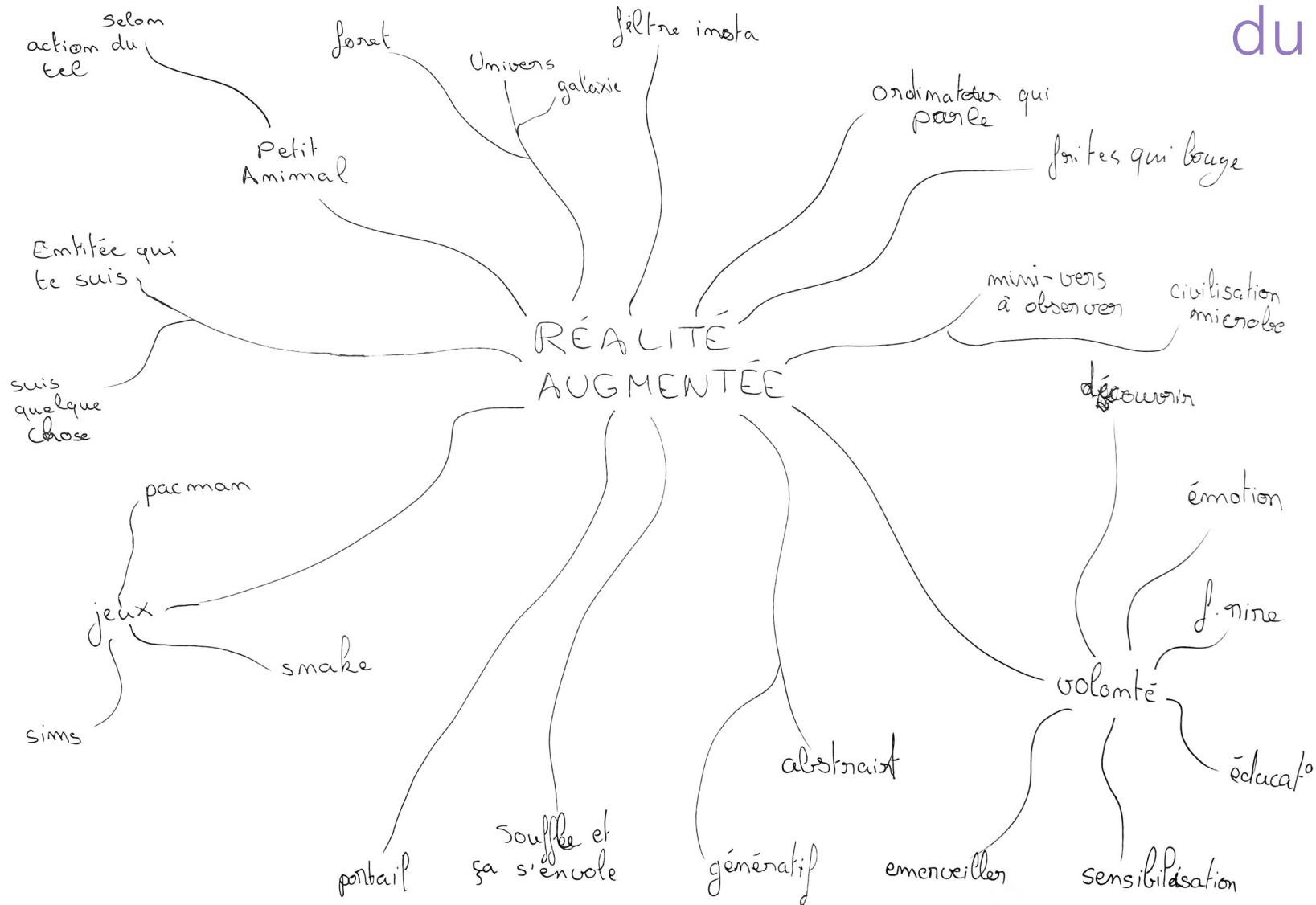
Choix d'animations	24
Essais textures	25-26
Apparence finale de Kyū	27
Animations en cours	28
Planche Univers	29-30
Tentatives de fonds	31
Fond final	32
Fond final avec Kyū	33
Un bug ••	34
Compte rendu journée 3	35

Journée 4

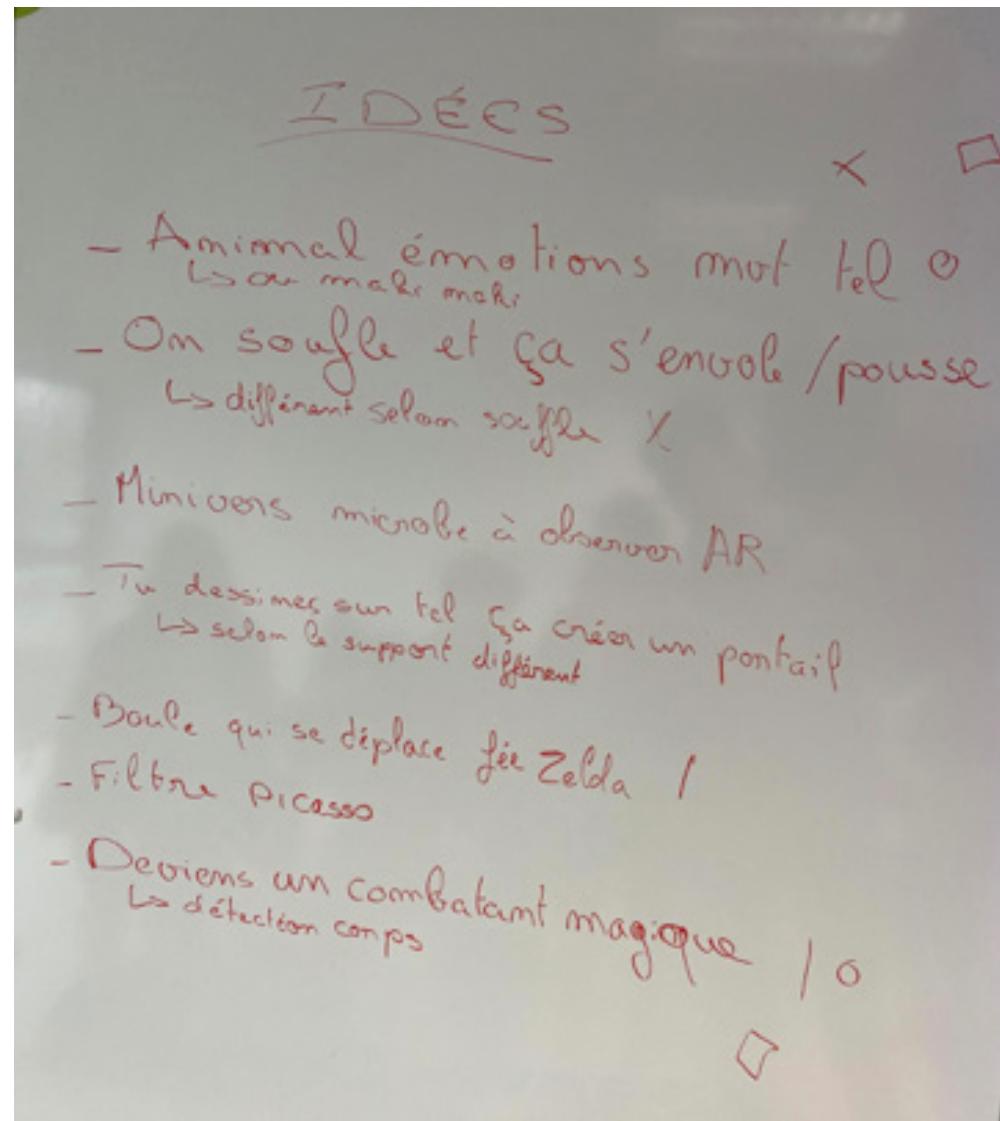
Texture de Kyu sur le fond	36
Enregistrement des sons	37
Travail des sons	38
Test final	39
Compte rendu de la journée	40
Tableau des réussites	41

Nos expériences	42
Merci !	43

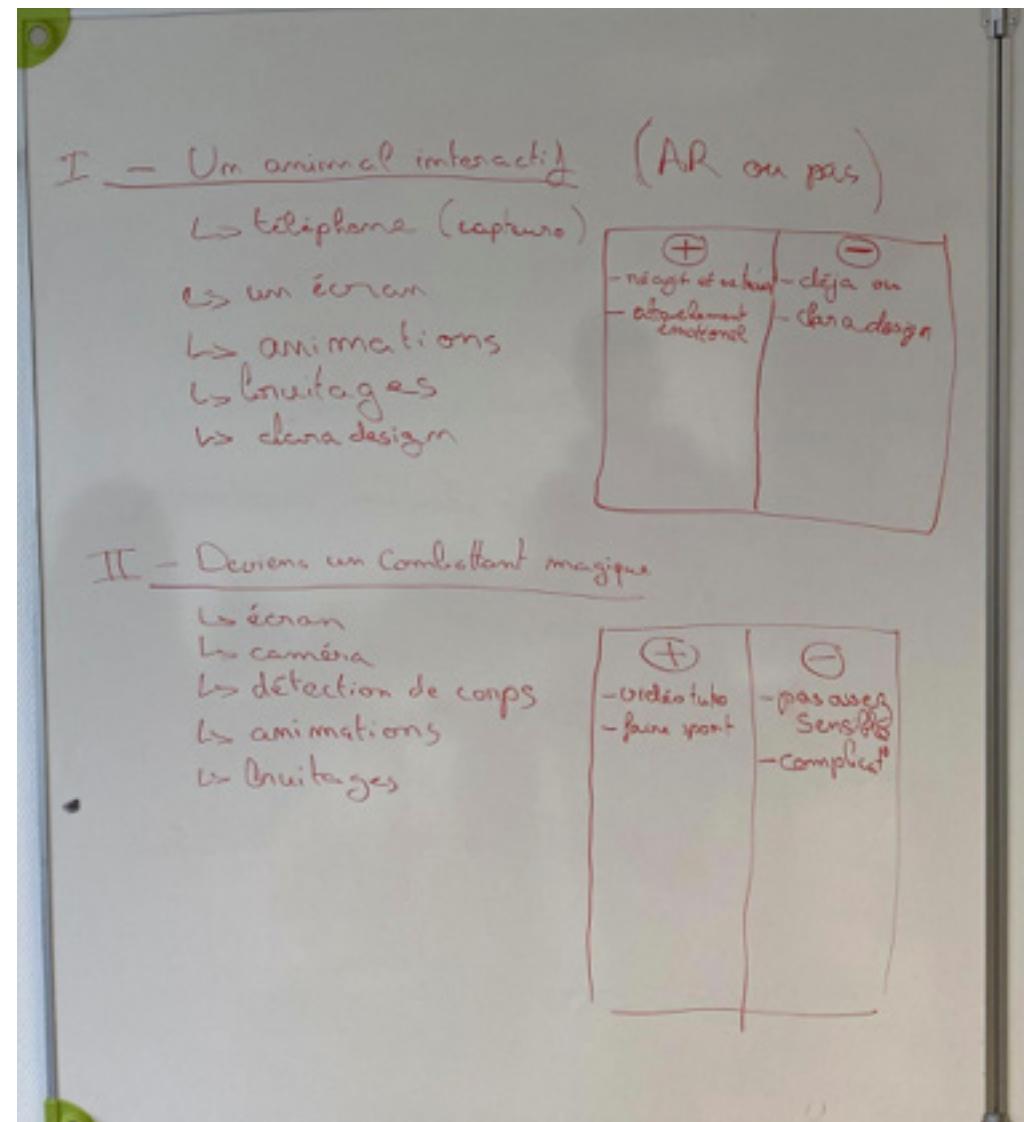
Recherche du sujet



Pistes

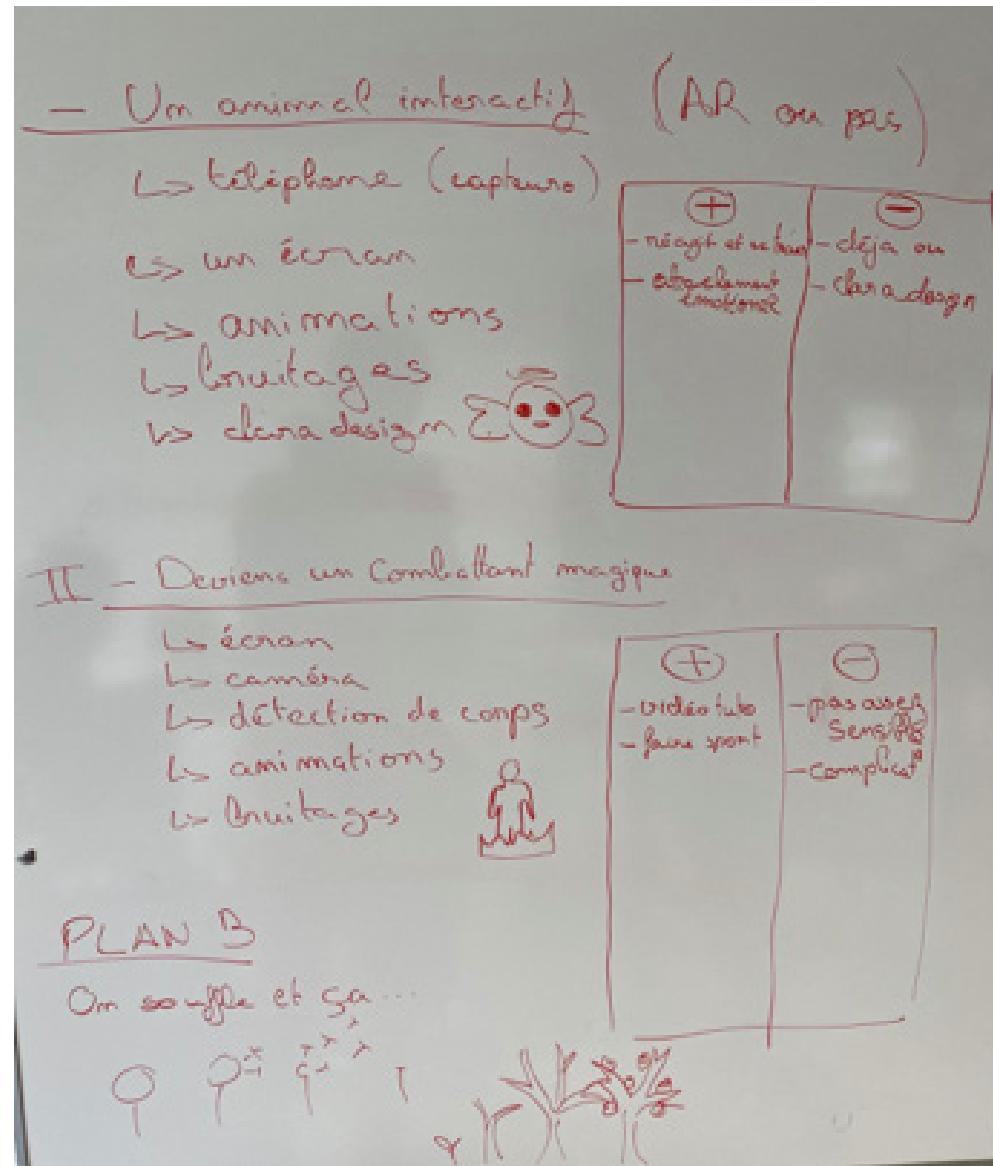


Choix du sujet



Choix du sujet

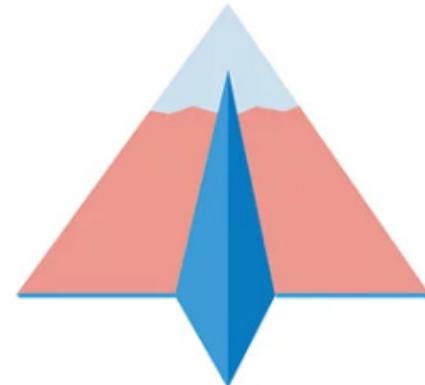
Étape 2



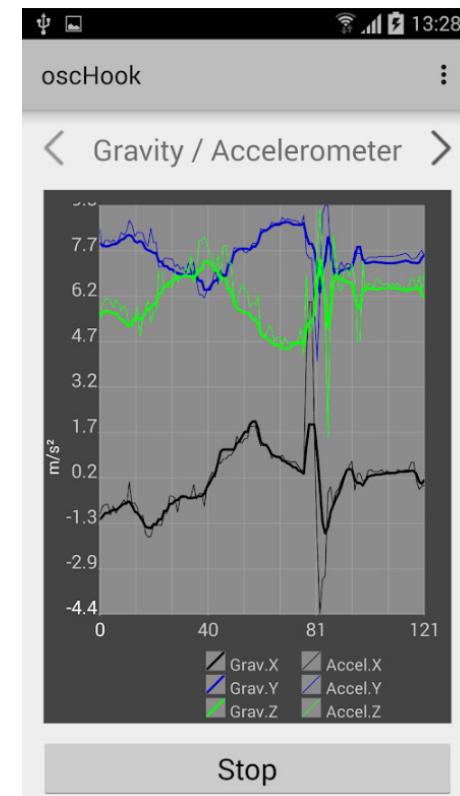
Le sujet numéro 1 gagne faceau
numéro 2, le Plan B est lui aussi choisi
dans le cas où le plan A échouerait.

27 Janvier 2020 — Après-midi

Ressources d'aide pour le code



Tramontana



oscHook

Premiers tests

The image shows a screenshot of the Processing 3.5.4 IDE. On the left, a code editor window titled "sketch_200202a" displays the following Java code:

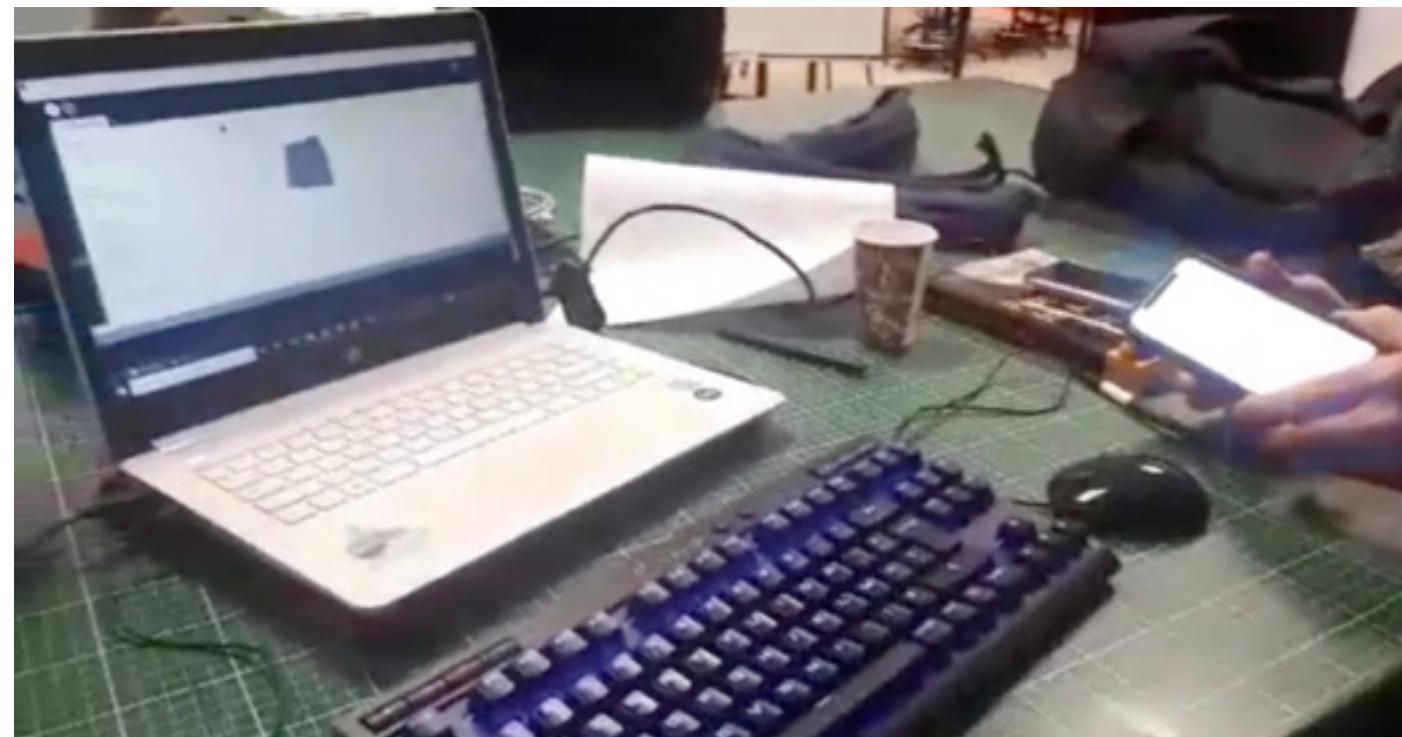
```
1 /**
2  * oscP5sendreceive by andreas schlegel
3  * example shows how to send and receive osc messages.
4  * oscP5 website at http://www.sojamo.de/oscP5
5 */
6
7
8 import netP5.*;
9 import oscP5.*;
10 OscP5 oscP5;
11 NetAddress myRemoteLocation;
12
13 float r1, r2, r3, r4;
14
15 void setup() {
16     size(400, 400);
17     frameRate(25);
18     /* start oscP5, listening for incoming messages at port 12000 */
19     oscP5 = new OscP5(this, 12000);
20
21     /* myRemoteLocation is a NetAddress. a NetAddress takes 2 parameters,
22      * an ip address and a port number. myRemoteLocation is used as parameter in
23      * oscP5.send() when sending osc packets to another computer, device,
24      * application. usage see below. for testing purposes the listening port
25      * and the port of the remote location address are the same, hence you will
26      * send messages back to this sketch.
27      */
28     myRemoteLocation = new NetAddress("10.130.18.187", 12000);
29 }
30
31 void draw() {
32     background(0);
33 }
34
35
36 void mousePressed() {
37     /* in the following different ways of creating osc messages are shown by example */
38     OscMessage myMessage = new OscMessage("callibrageon");
39
40     myMessage.add(123); /* add an int to the osc message */
41
42     /* send the message */
43     oscP5.send(myMessage, myRemoteLocation);
44 }
45
46
47
48 void oscEvent(OscMessage theOscMessage) {
49     /* check if theOscMessage has the address pattern we are looking for. */
50
51     if (theOscMessage.checkAddrPattern("/rotation_vector/r1") == true) {
```

On the right, a "Java Examples" browser window is open, showing a tree structure of examples. The "oscP5" category is expanded, and the "oscP5multicast" example is selected.

Code avec OSCHook

27 Janvier 2020 — Après midi

Premiers tests



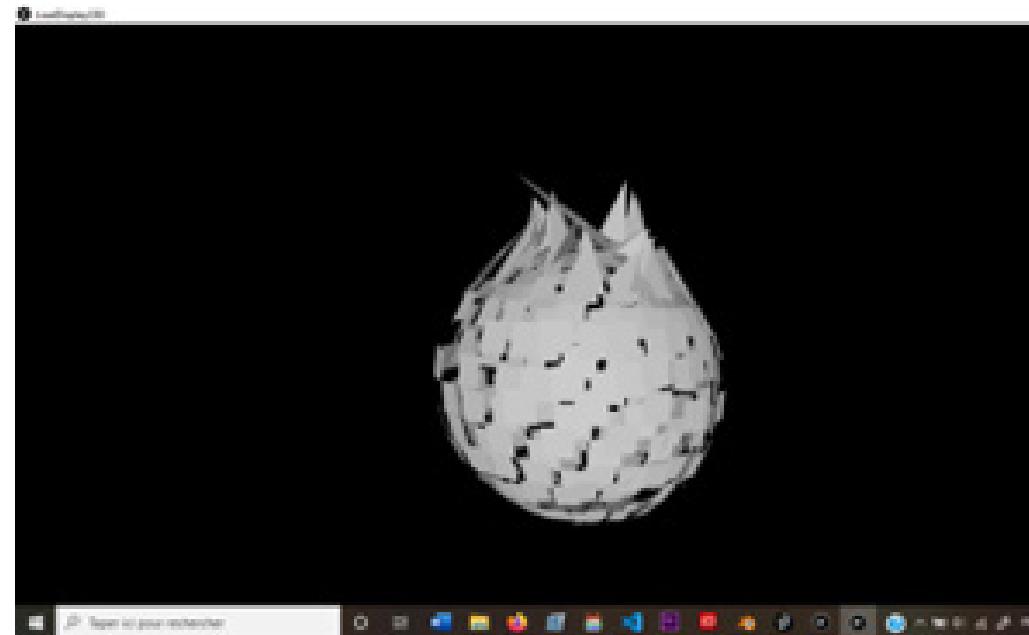
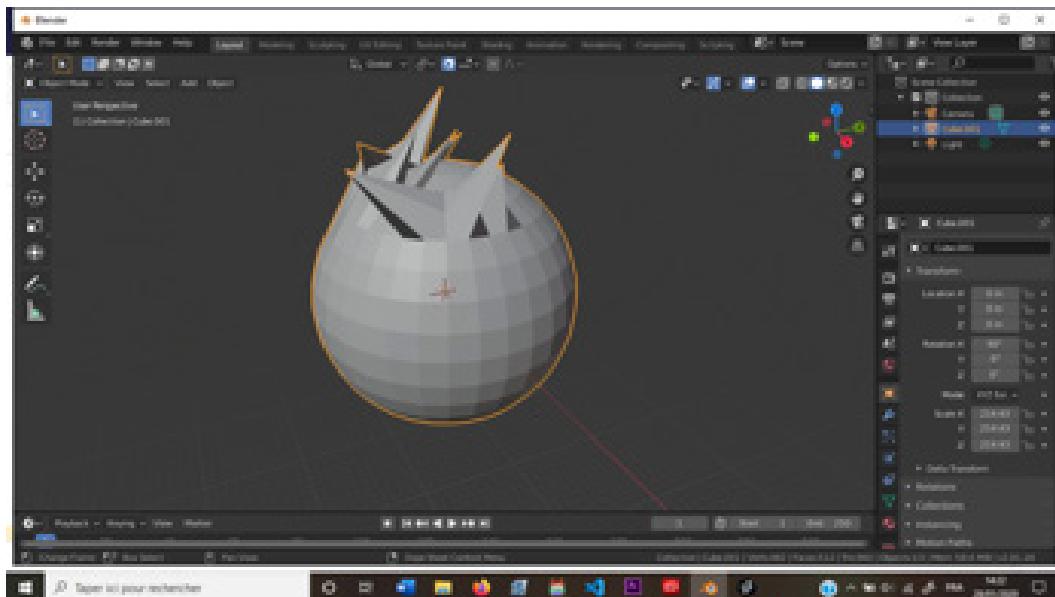
Utiliser l'accéléromètre d'un téléphone pour afficher la rotation sur Processing.

28 Janvier 2020 — Matin

Ressources d'aide graphiques

Test d'un objet 3D sur
Processing se décomposant et
se recomposant

Réalisé avec Blender



Recherche des caractères, de l'apparence générale et des actions de la créature



Pendant 30min nous avons écrit tous ce qui nous passait par la tête pour imaginer (enfin) la créature que nous allions animer.

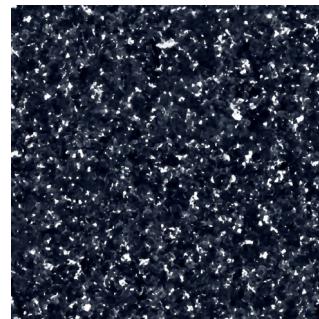
Définition des caractères

Après délibérations notre créature sera une sphère mignonne qui fera des actions tel que roule ou se diviser.



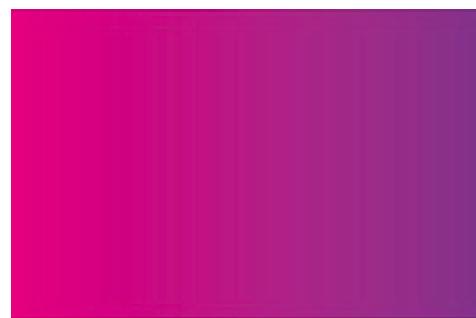
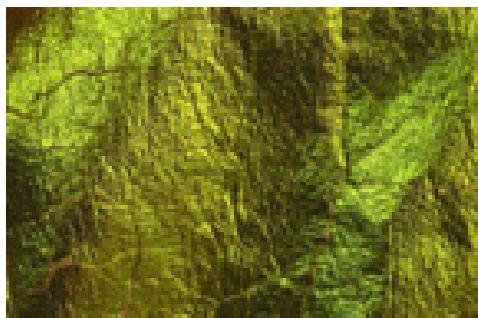
28 Janvier 2020 — Après-midi

Recherche de textures de la créature



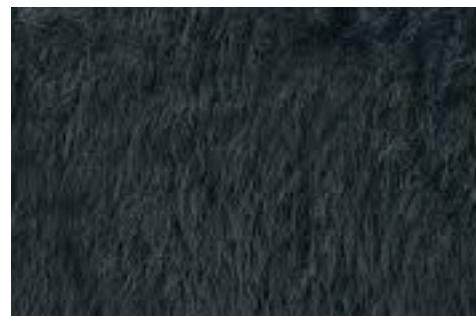
28 Janvier 2020 — Après-midi

Recherche de textures de la créature



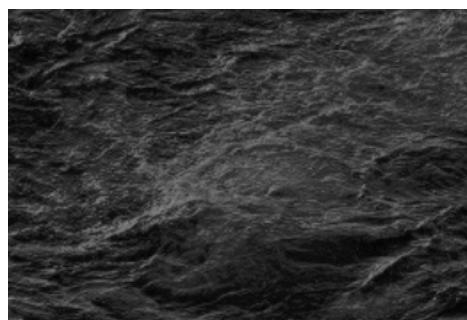
28 Janvier 2020 — Après-midi

Recherche de textures de la créature



28 Janvier 2020 — Après-midi

Recherche de textures de la créature



Réalisation des réactions téléphone - ordinateur

```
/*
 * oscP5sendreceive by andreas schlegel
 * example shows how to send and receive osc messages.
 * oscP5 website at http://www.sojamo.de/oscP5
 */

import netP5.*;
import oscP5.*;
OscP5 oscP5;
NetAddress myRemoteLocation;

float lux=500;

float shake;
float shakeMax;
float shakeMin;

float turn;

float horiz;

int etat_background = 0;
long declenchement_secoue = 0;

void setup() {
  size(400, 400);
  background(255);

  frameRate(25);
  /* start oscP5, listening for incoming
  messages at port 12000 */
  oscP5 = new OscP5(this, 12000);

  myRemoteLocation = new NetAddress(<<192.168.13.111>>,
12000);
}

void draw() {
  if (shake>0.2) {
    shake= shakeMax;
  }
  if (shake<-0.3) {
    shake= shakeMin;
  }

  // Tester les différents cas qui définissent
  l'état du background
  etat_background = 0;
  if (lux<10 == true) etat_background = 2;
  if ((shakeMax-shakeMin)>7== true) {
    etat_background = 1;
    declenchement_secoue = millis();
  }
  if (turn<-8) etat_background = 3; //
  Tourner à droite

  if (turn>8) etat_background = 4; //
  Tourner à gauche

  if (horiz>9) etat_background = 5; //
  Téléphone horizontal écran haut

  if (horiz<-9) etat_background = 6; //
  Téléphone horizontal écran bas

  // Appliquer la couleur de fond
  switch(etat_background) {

    case 0:
      background(255);
      break;
    case 2:
      background(0);
      break;
    case 3:
      background(222, 150, 0);
      break;
    case 4 :
      background(255, 250, 0);
      break;
    case 5 :
      background(255, 210, 210);
      break;

    case 6 :
      background(0, 200, 255);
      break;
    default:
      background(255);
      break;
  }
  if (millis() - declenchement_secoue < 2000) background(255,
255, 255);
}
}
```

Réalisation des réactions téléphone - ordinateur

```
20, 220);  
}  
void oscEvent(OscMessage theOscMessage) {  
    /* check if theOscMessage has the address pattern we are  
     * looking for. */  
    //LUMIERE  
    if (theOscMessage.checkAddrPattern("/light") == true) {  
        /* check if the typetag is the right one. */  
        if (theOscMessage.checkTypetag("f")) {  
            /* parse theOscMessage and extract the values from the osc  
             * message arguments. */  
            lux = theOscMessage.get(0).floatValue();  
            println("### reçu /light " + lux);  
            return;  
        }  
    }  
    //SECOUER  
    if (theOscMessage.checkAddrPattern("/accelerometer/linear/  
x") == true) {  
        /* check if the typetag is the right one. */  
        if (theOscMessage.checkTypetag("f")) {  
            /* parse theOscMessage and extract the values from the osc  
             * message arguments. */  
            shakeMax = theOscMessage.get(0).floatValue();  
  
            println("### reçu /accelerometer/linear/x " + shake);  
            return;  
        }  
    }  
    //TOURNER COTES  
  
    if (theOscMessage.checkAddrPattern("/accelerometer/raw/  
x") == true) {  
        /* check if the typetag is the right one. */  
        if (theOscMessage.checkTypetag("f")) {  
            /* parse theOscMessage and extract the values from the osc  
             * message arguments. */  
            turn = theOscMessage.get(0).floatValue();  
            println("### reçu /accelerometer/raw/x " + turn);  
            return;  
        }  
    }  
    //METTRE A L'HORIZONTALE  
  
    if (theOscMessage.checkAddrPattern("/accelerometer/raw/  
z") == true) {  
        /* check if the typetag is the right one. */  
        if (theOscMessage.checkTypetag("f")) {  
            /* parse theOscMessage and extract the values from the  
             * osc message arguments. */  
            horiz = theOscMessage.get(0).floatValue();  
            println("### reçu /accelerometer/raw/z " + horiz);  
            return;  
        }  
    }  
}
```

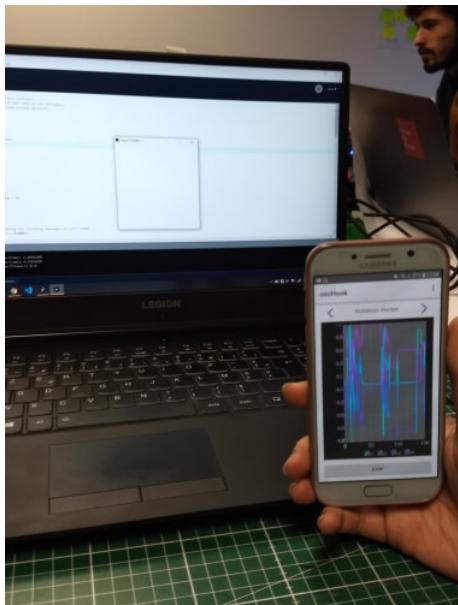


Voir le code sur Github

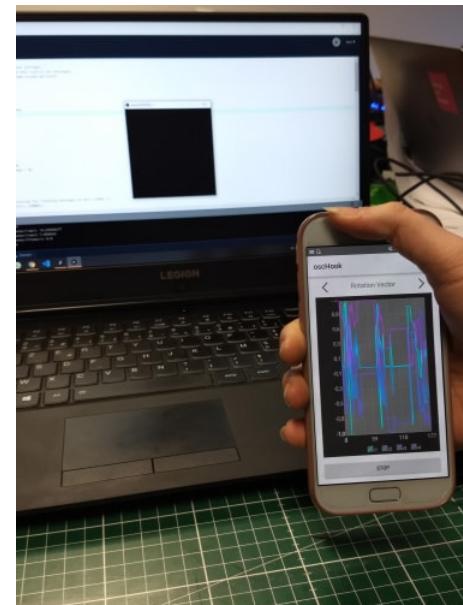
28 Janvier 2020 — Après-midi

Réalisation des réactions téléphone - ordinateur

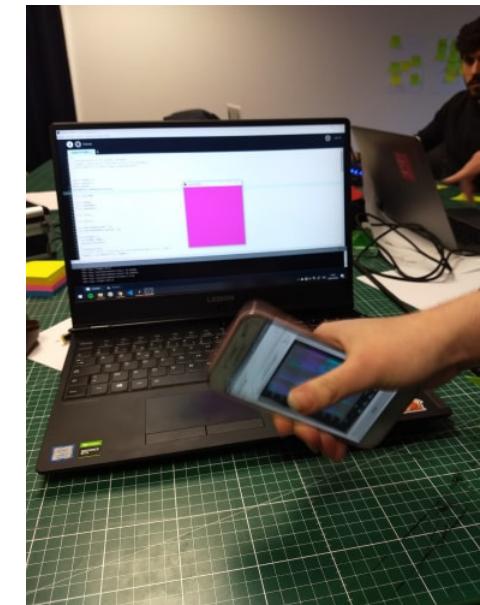
Détermination des positions que l'on utilisera pour faire bouger la créature et création du code fondateur



Ne fait rien



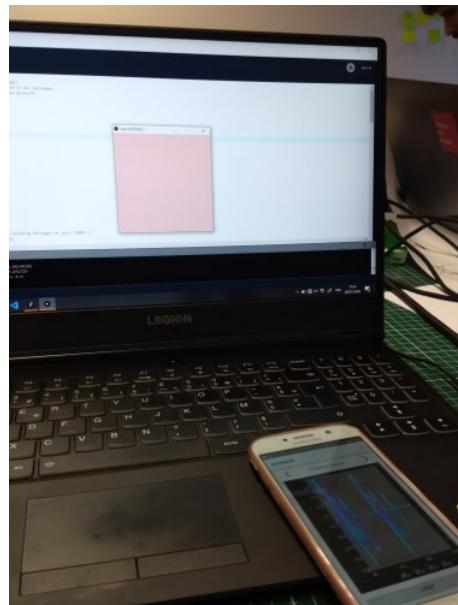
Capteur de luminosité caché



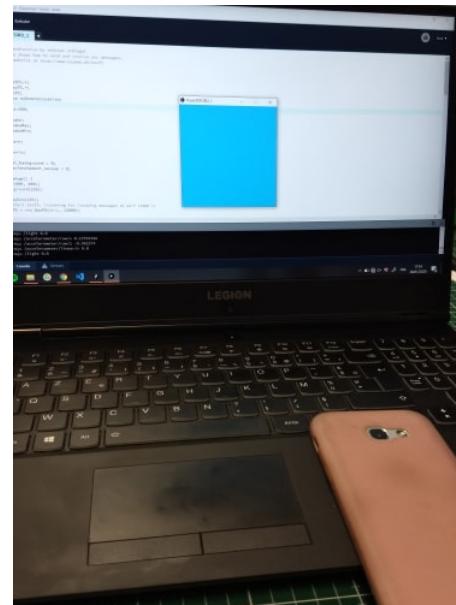
Secoué

28 Janvier 2020 — Après-midi

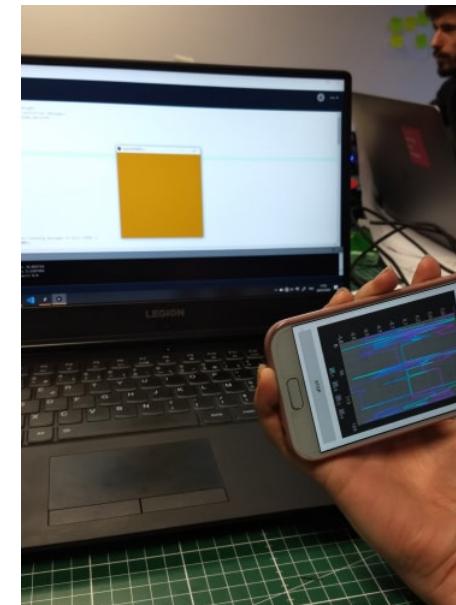
Réalisation des réactions téléphone - ordinateur



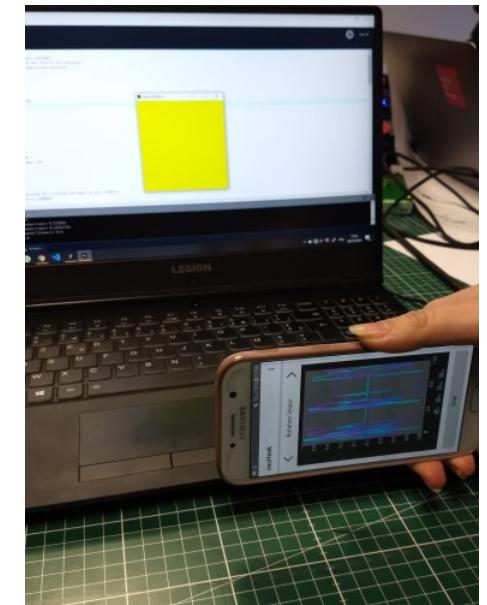
Couché sur le dos



Couché sur l'écran

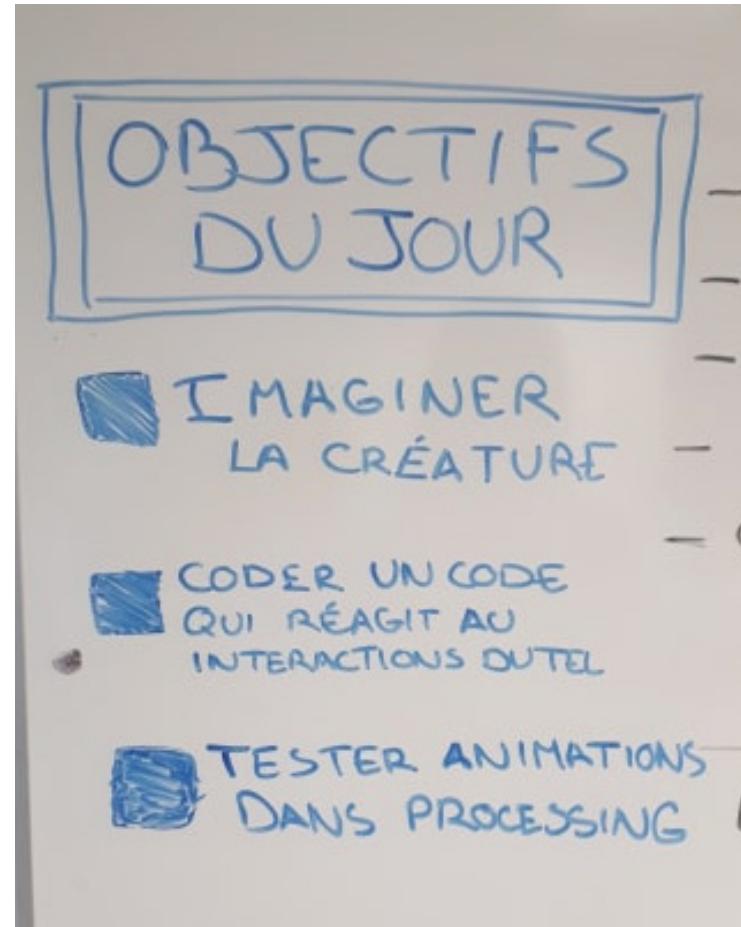


Rotation vers la droite

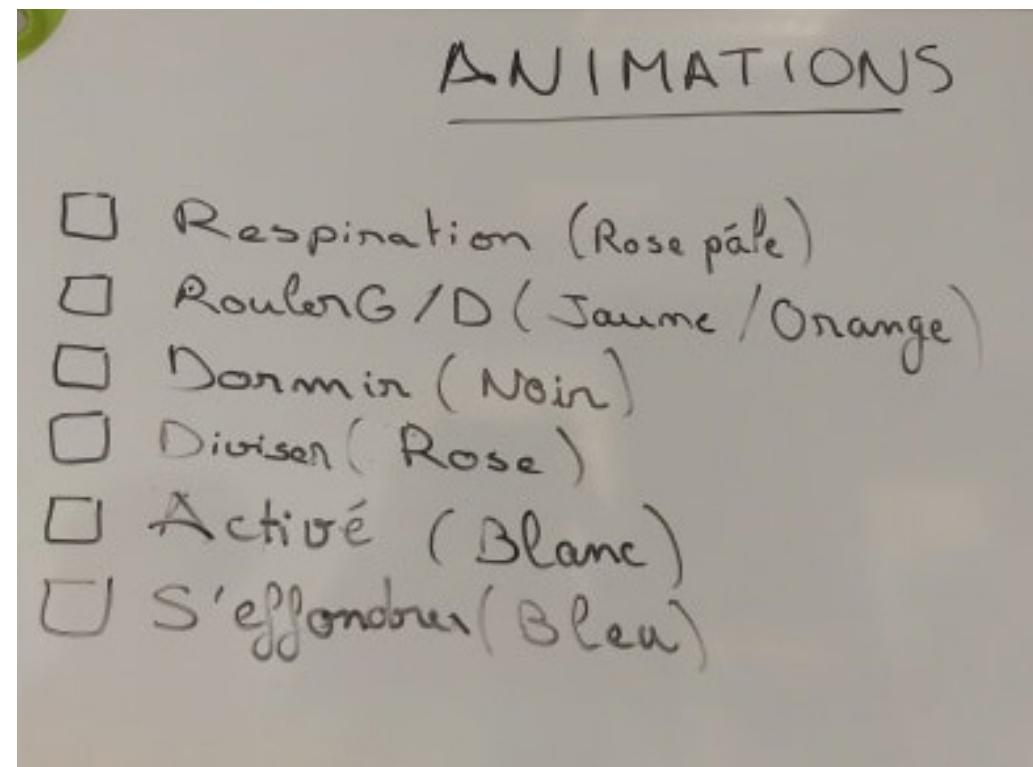


Rotation vers la gauche

Compte rendu de la journée

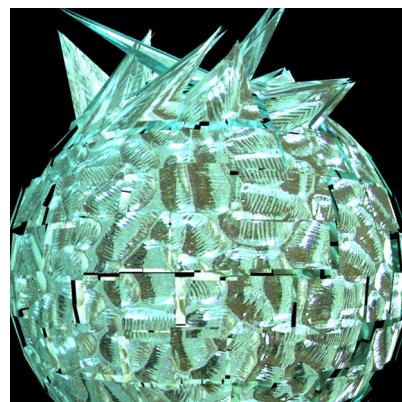
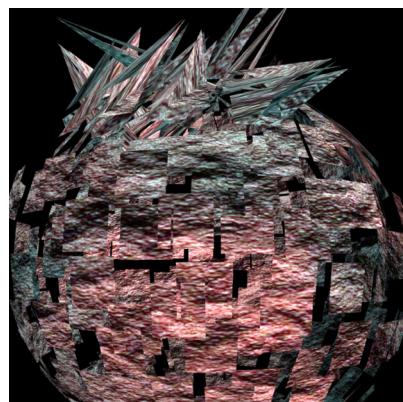
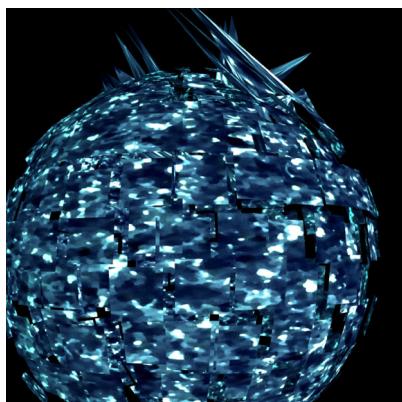
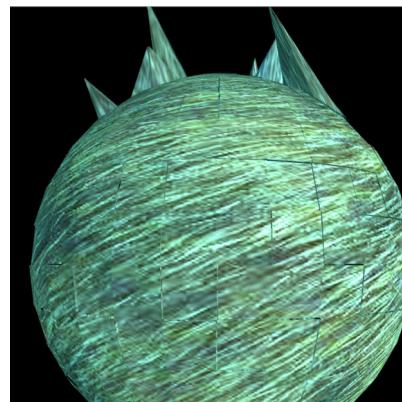
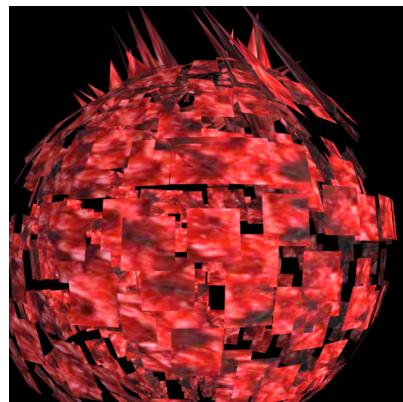
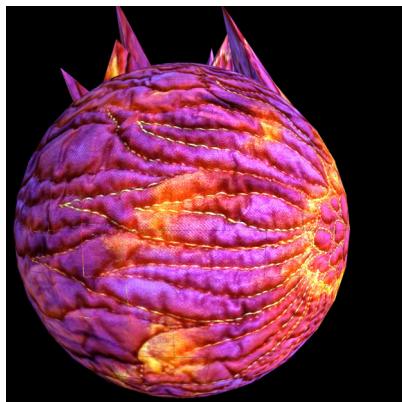
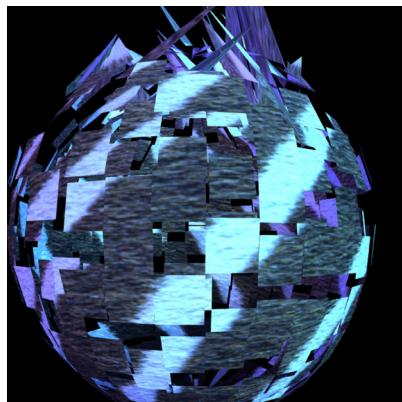


Choix des animations



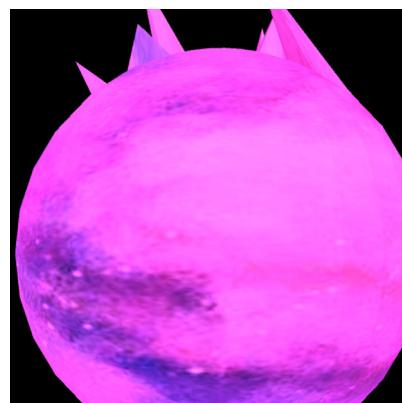
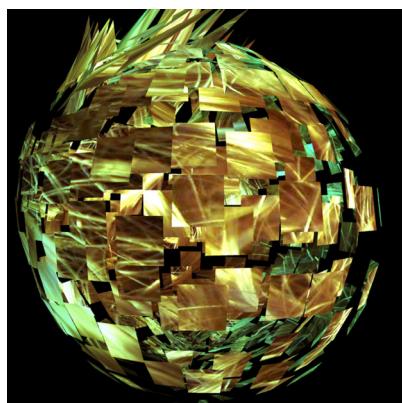
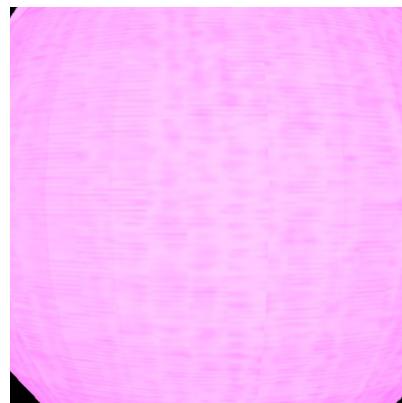
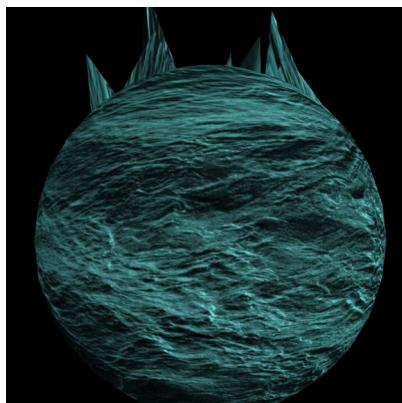
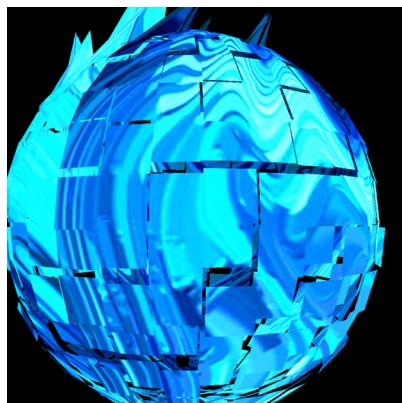
29 Janvier 2020 — Matin

Essais de textures



29 Janvier 2020 — Matin

Essais de textures

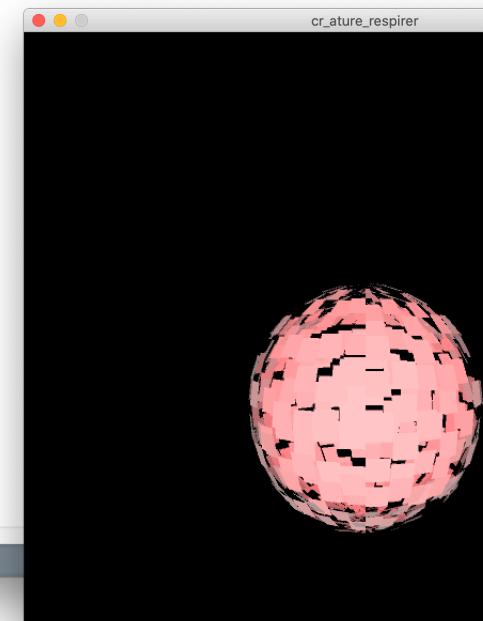


29 Janvier 2020 — Après-midi

Apparence finale de Kyū (la créature)



Animations en cours



The image shows the Processing 3.5.4 software interface. On the left, the code editor window displays a Processing sketch named "cr_ature_respirer". The code uses PShape to load an OBJ file ("untitled.obj") and applies three sinusoidal deformations (deformation1, deformation2, deformation3) to each vertex of the shape's children. The resulting 3D object is a pinkish-red sphere with visible texture and deformation. On the right, the preview window shows the rendered 3D scene.

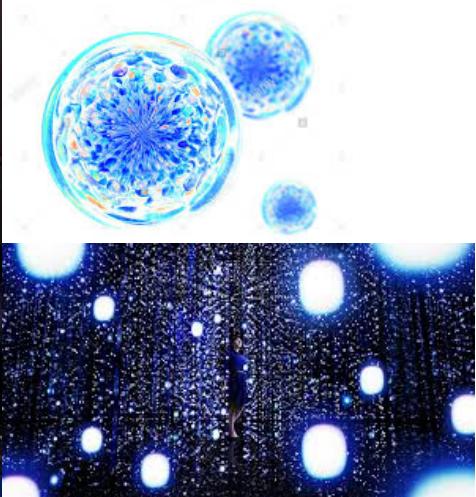
```
9
10 PShape s;
11 float ry;
12
13 public void setup() {
14     size(640, 560, P3D);
15
16     s = loadShape("untitled.obj");
17     println( s.getChildCount());
18 }
19
20 public void draw() {
21     background(0);
22     lights();
23     //ambientLight(224,8,127);
24     spotLight(255, 0, 0, width/2, height/2, 400, 0, 0, -1, PI/4, 2);
25
26     for (int j = 0; j < s.getChildCount(); j++) {
27         PShape child = s.getChild(j);
28
29
30         for (int i = 0; i < child.getVertexCount(); i++) {
31             float deformation1 = sin (frameCount*15.+j)/2.5;
32             float deformation2 = sin (frameCount/20.+j )/2.5;
33             float deformation3 = sin (frameCount/40.+j )/2.5;
34             PVector v = child.getVertex(i);
35             v.x += deformation1 ;
36             v.y += deformation2;
37             v.z += deformation3 ;
38             child.setVertex(i, v);
39         }
40     }
41     translate(width/2, height/2 + 100, -200);
42     rotateZ(PI);
43     //rotateY(ry);
44     shape(s);
45
46     ry += 0.02;
47 }
48 }
```

29 Janvier 2020 — Après-midi

Planche Univers



Détachement du fond

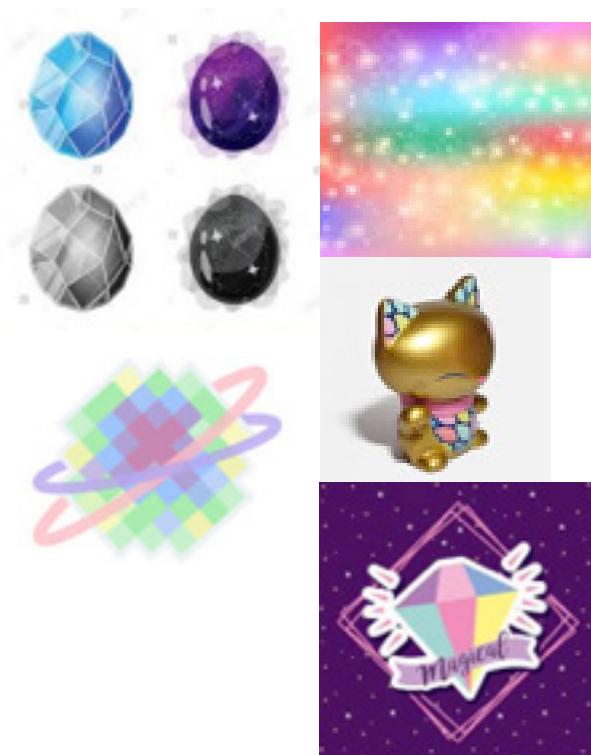


Monstueux coloré



29 Janvier 2020 — Après-midi

Détachement du fond



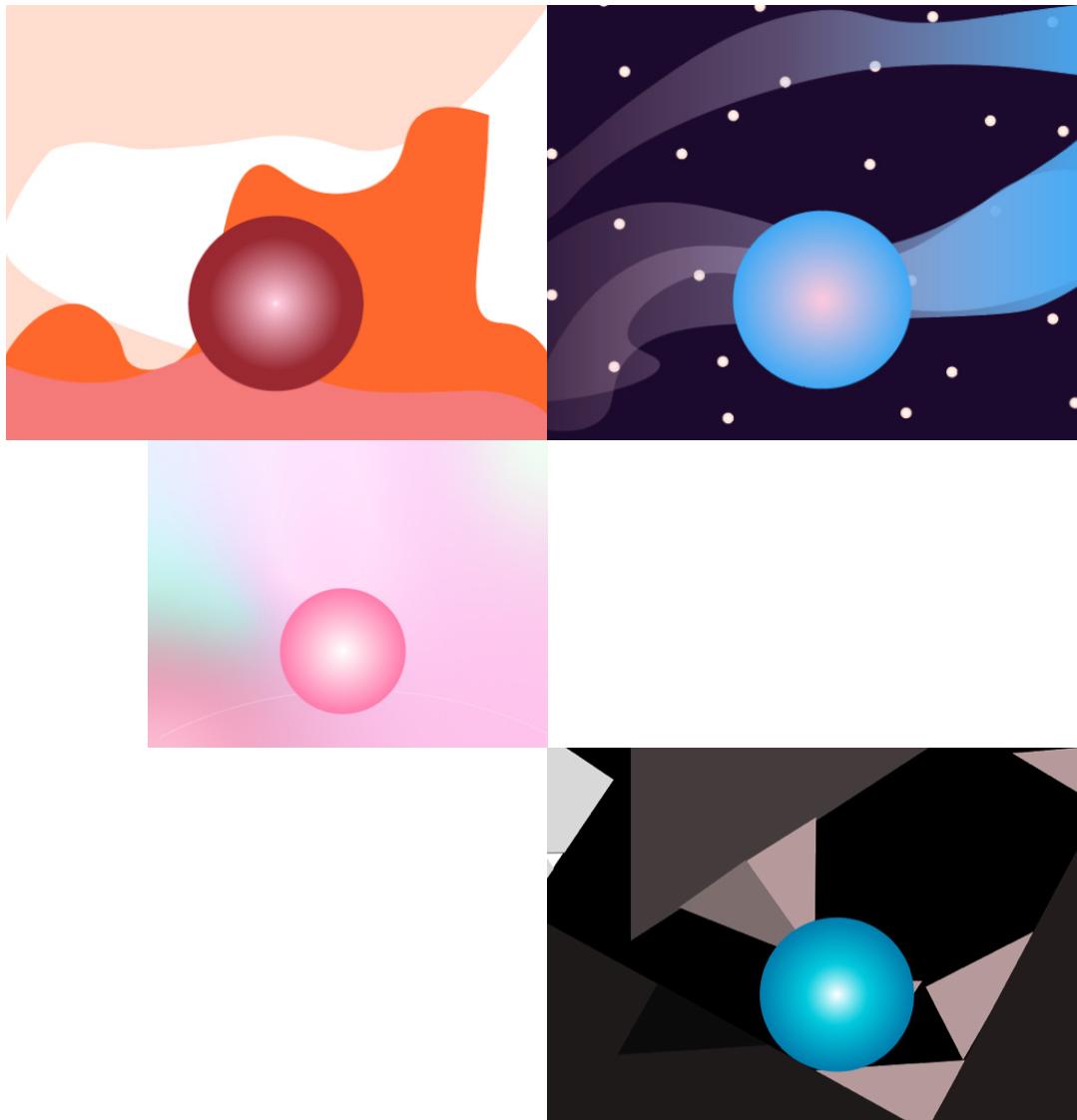
Pastel Fantaisiste



Peluche dans la chambre

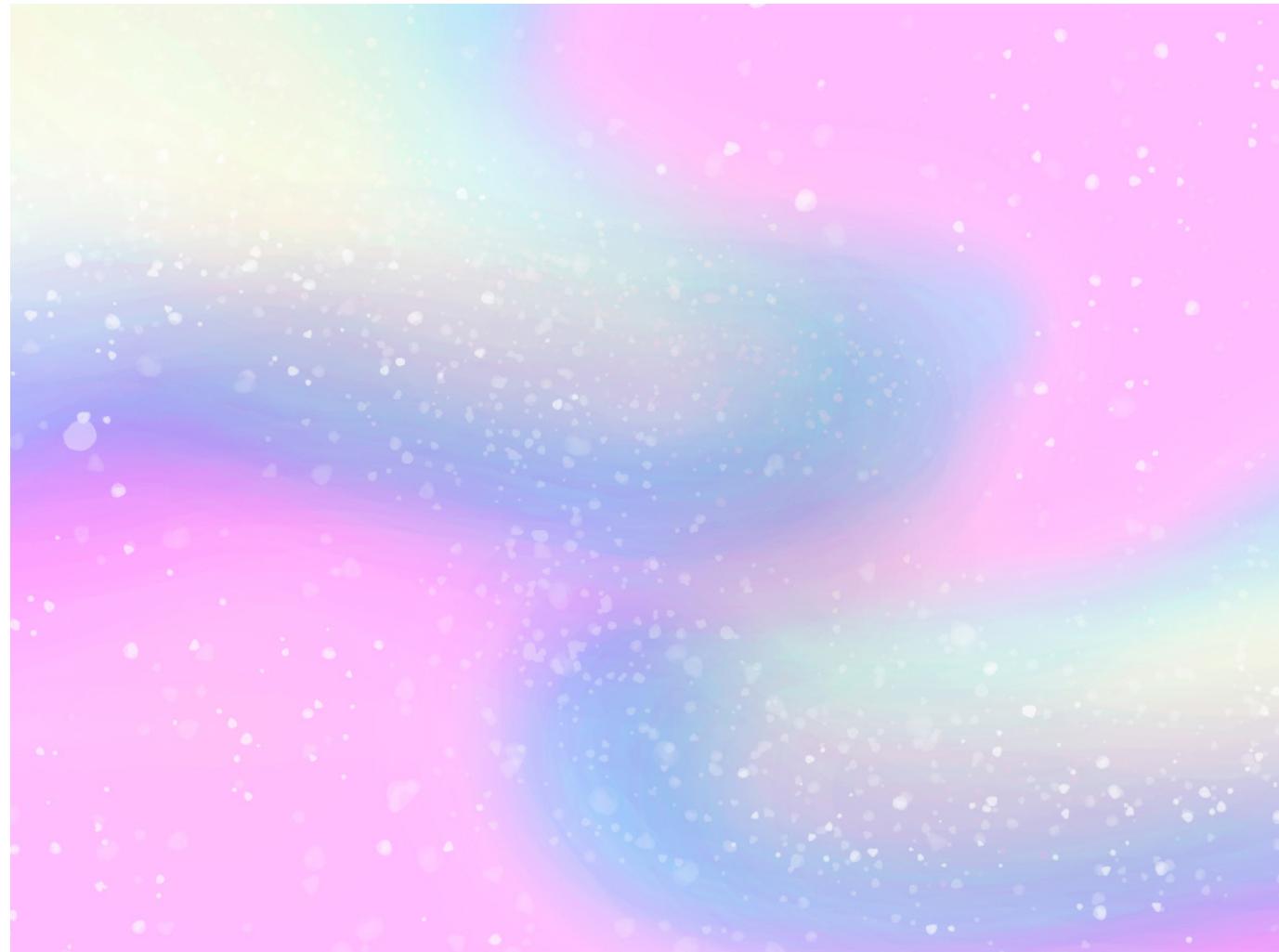
Tentatives de fonds

Essais sur Illustrator pour trouver la bonne ambiance.



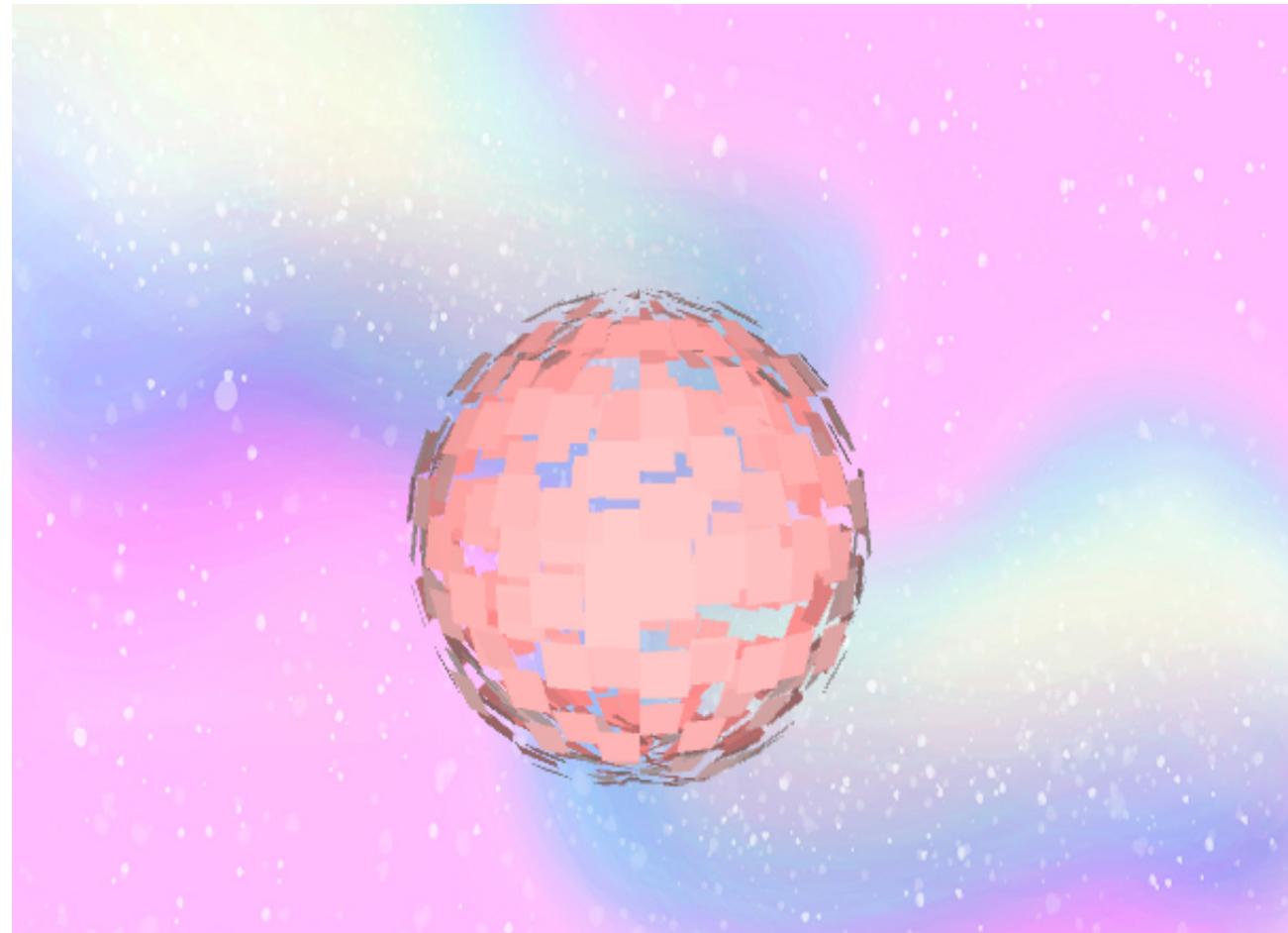
Fond final

Notre choix final s'est posé sur un fond inspiré de la série animée Star Butterfly pour ses couleurs pastel et son inspiration galactique



29 Janvier 2020 — Après-midi

Fond final
avec Kyū

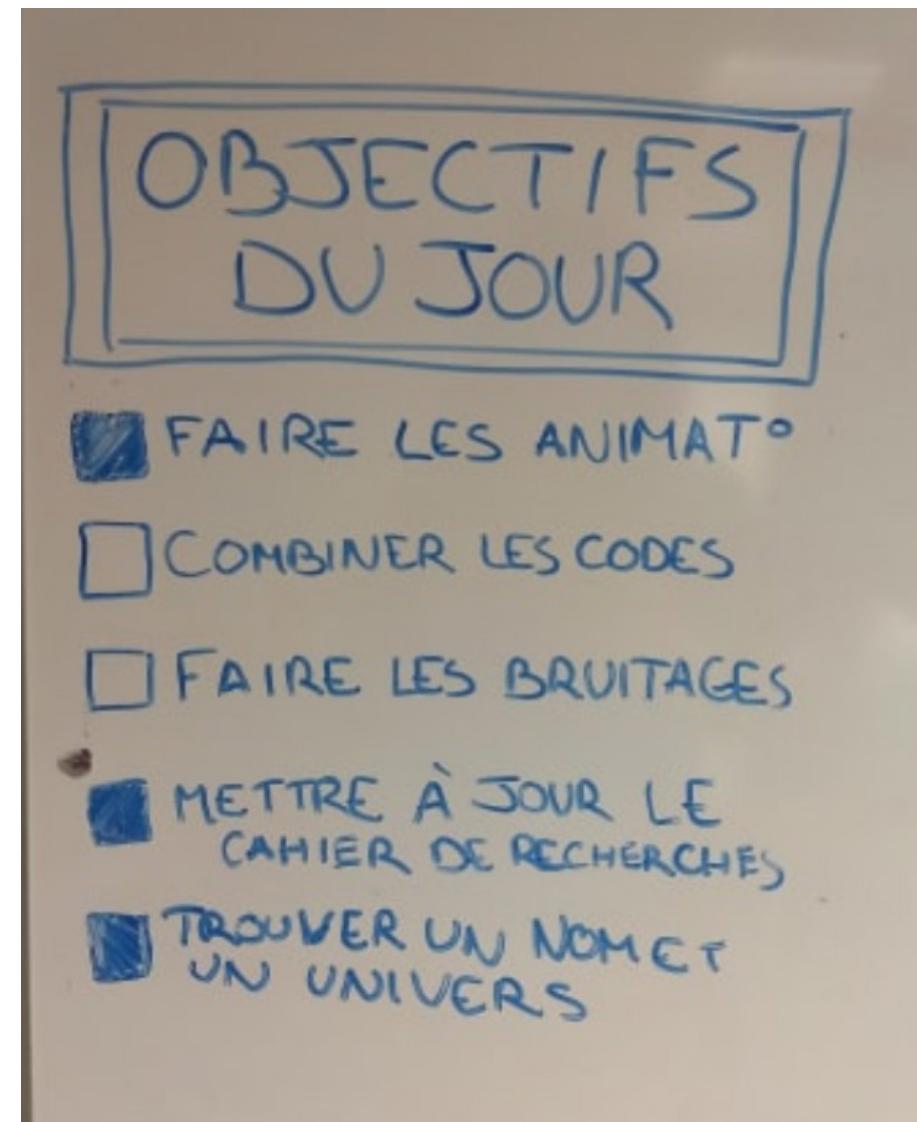


Gneu..
un bug •--•



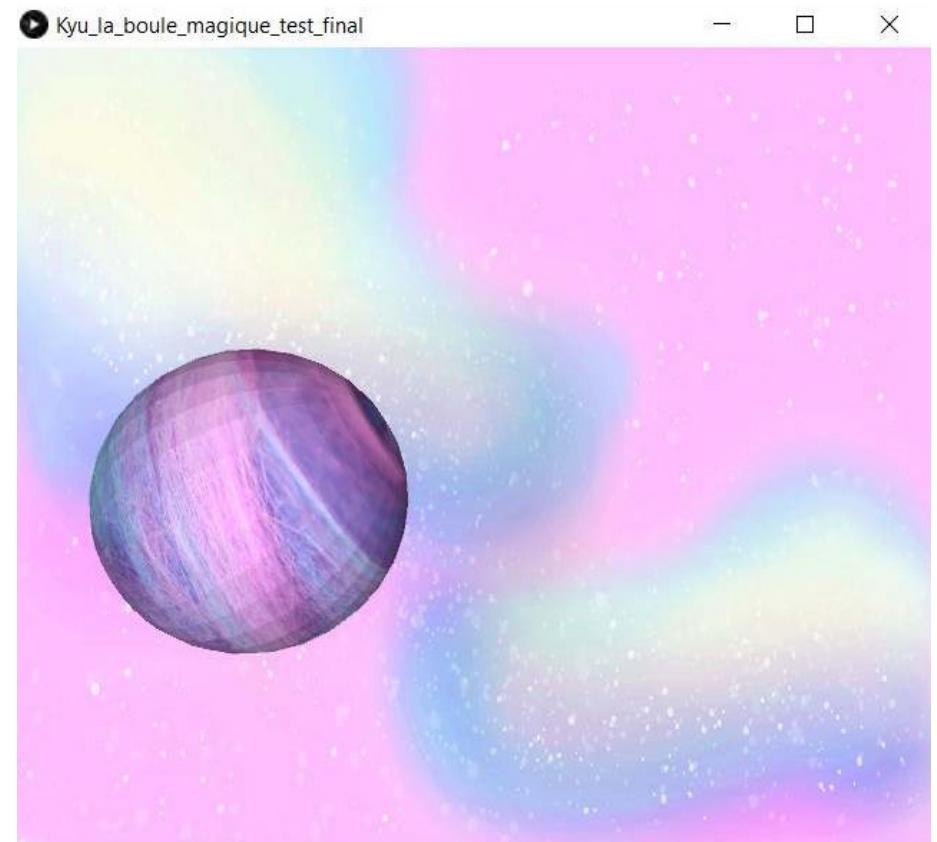
La sphère ne revenait
pas à l'état normale et
restait séparé en deux

Compte rendu de la journée



30 Janvier 2020 — Matin

Texture de Kyū
sur le fond
et dernières
animations



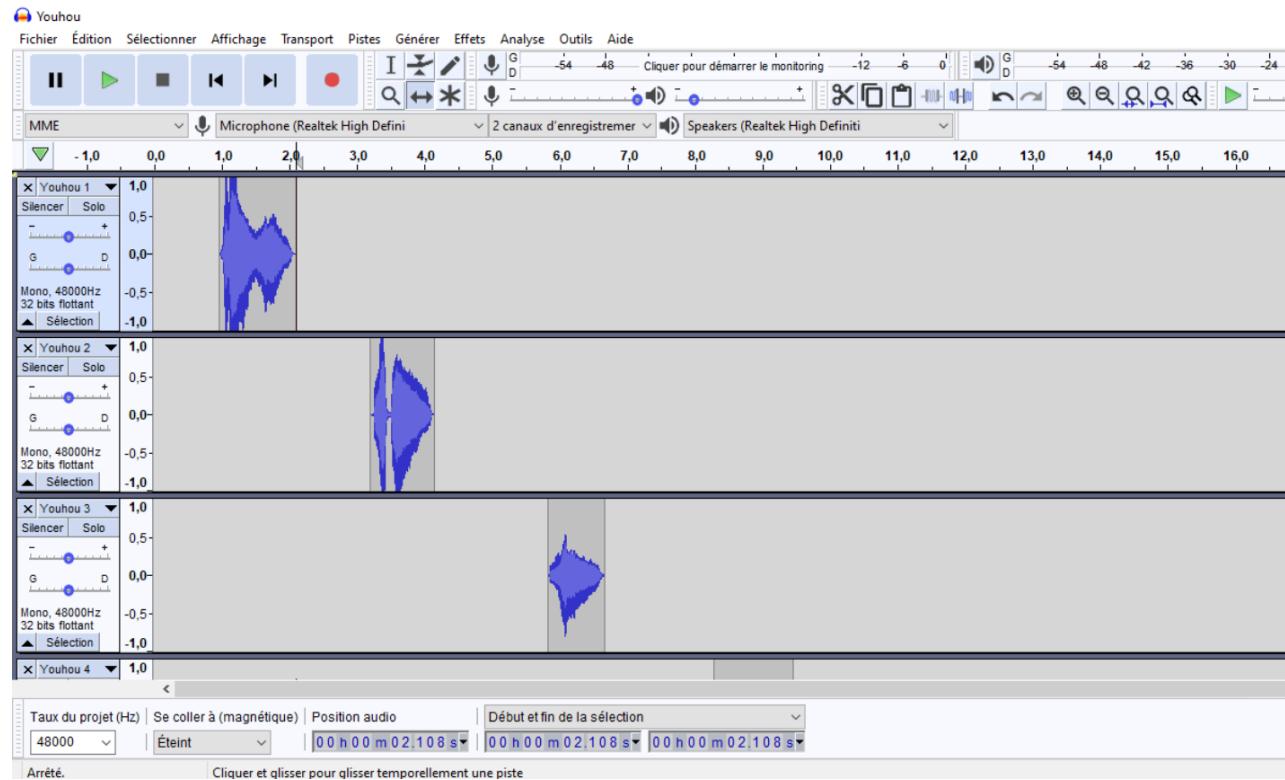
30 Janvier 2020 — Matin

Enregistrement des sons

Nous avons enregistré pleins d'essais de voix qui pourraient coller avec la personnalité de notre créature, ainsi qu'une mélodie à la guitare pour le fond sonore.



Travail des sons



Test final

Seulement une demie-heure avant le rendu et tout fonctionne enfin, les sons sont bien placés, les animations n'ont plus de problèmes et la sensibilité de la lumière à été réglée pour le lieu son dans lequel on va présenter Kyū.



Compte rendu de la journée

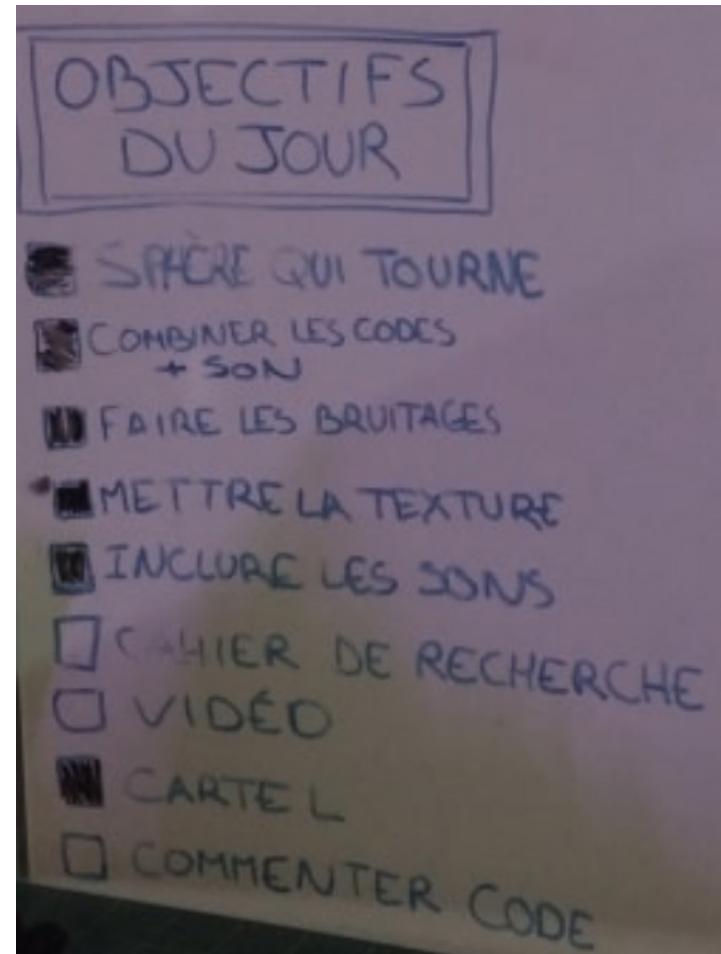
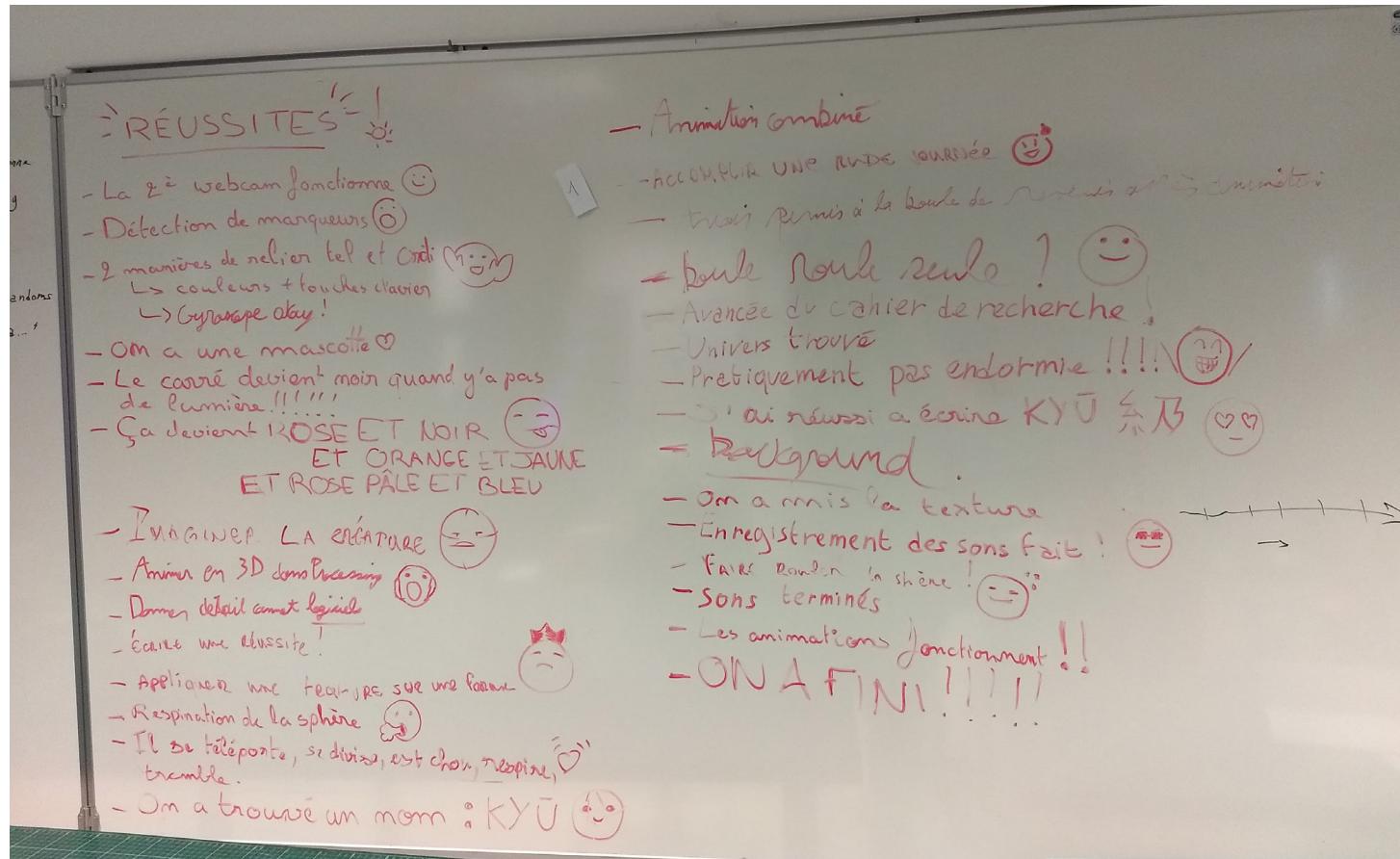


Tableau des réussites



Nos experiences

Durant ce workshop j'ai appris beaucoup de chose, autant sur l'apprentissage du code que le travail d'équipe. En réalisant Kyū et en gardant et définissant nos objectifs dès le départ cela m'a permis de tenter une nouvelle création et de sortir de ma zone de confort, comme par exemple réaliser des animations en 3D faire en sorte d'interagir avec le téléphone grâce à Processing. En ce qui concerne le travail d'équipe, j'ai appris à essayer de communiquer avec les autres de façon plus claire et d'être plus patient pour réaliser le projet. J'ai en tout cas pris beaucoup de plaisir à créer Kyū avec ce groupe d'ami.

— Martin Fourny

Ce workshop m'a permis d'apprendre à travailler en groupe dans le stress. Mais il m'a surtout permis de découvrir qu'en groupe nous sommes capable de réaliser beaucoup de choses en code qu'il y a une semain je ne croyais pas possible. J'ai découvert comment faire de la réalité augmentée, utiliser les positions du téléphone, faire du montage audio, intégrer des créations 3D sur processing et comment lire un code qu'on a jamais vu auparavant. Je suis maintenant plus confiante sur processing avec lequel j'avais quelques difficultés.

— Coralie Picard

Ce workshop m'a permis de concilier autonomie et travail d'équipe en même temps, de travailler avec des amis avec des caractères très différents, pour finalement aboutir à la création de notre petite boule rose. J'ai pu découvrir des logiciels comme OSCHook, qui me servira assurément plus tard, et comment fonctionne la réalité augmentée. En clair, ce workshop a été enrichissant et amusant à la fois.

— Enora Jaffre

Ce workshop fût une expérience enrichissante, j'ai pu me plonger de façon plus approfondie sur Processing. Chaque animations que j'essayais d'intégrer sur Processing était comme un nouveau challenge ! La deadline est bien évidemment importante, il a fallu se surpasser ce qui était pas forcément évident avec les bugs et difficultés rencontrées.

— Maël Jallais

Merci !

Après de longues heures de travail de galère, des centaines de prières, et des centaines de lignes de codes... Kyū, la créature interdimensionnelle a pris vie ! Ces sons mignons, ses couleurs enjouées et ces mouvements trop «Kyūte», nous ont fait oublier les heures de sommeils manquantes. Tous les quatre très fières de nous et de notre petite créature mouvante nous avons perfectionner notre présentation, faisant entrer le spectateur dans un univers mignon, doux et chaleureux emplis de lumières roses et de peluches. Tout cela se mêlant à la perfection à la musique de fond relaxante et à la respiration douce de Kyū.



Voir la vidéo





De la sphère à l'étape (en passant par Google traduction)

球

Définition du mot Kyū

Dans les pratiques traditionnelles japonaises, c'est une étape dans les cérémonies ou coutume, quelque d'important dans la progression ou une évolution