Concours Cgénial 2019

Hologramme 3D interactif

Un projet du club informatique
du lycée François Magendie

Réalisé par :

Elisa BROSSU
Sarah HEURTEVENT
Romane SPAMPINATO
Théo BENSOUSSAN--PERRIAT
Alban CORBEL
Isaac MAJAU-PAGOUAPE
Audran MEYRIGNAC

Sous la supervision de :

Frédéric Placin
Enseignant de sciences physiques

Lycée François Magendie
10 rue des Treuils, 33000 BORDEAUX

Résumé

Nous avons comme objectif de modifier ou déplacer un objet virtuel holographique à l'aide d'une baguette magique. L'image holographique est obtenue à l'aide d'une vitre positionnée à 45° d'un écran d'ordinateur placé horizontalement. L'observateur a l'impression que l'image est en 3D et située derrière la vitre. Une LED infrarouge au bout d'une baguette permet à l'observateur de déplacer ou modifier l'objet. Les positions de la LED sont calculées à partir des images de deux webcams placées à 90° l'une de l'autre, chaque caméra repère le point le plus lumineux de son image. Actuellement nous pouvons déplacer un objet virtuel, et nous avons comme objectif de pouvoir le déformer et l'imprimer avec une imprimante 3D.

Summary

Our goal is to modify or to change the position of a virtual holographic object thanks to a magic wand. The holographic image is obtained thanks to a windowpane placed at 45° from a computer screen positioned horizontally. The observer thinks that the image is in 3D and is located behind the windowpane. An infrared LED at the extremity of the wand allows the observer to modify the object. The positions of the LED are calculated based on the images of two cameras placed at 90° from each other, each camera locating the brightest spot on its image. At the moment, we are able to move a virtual object and our goal is to modify its shape and to print it with a 3D printer.

Problématique

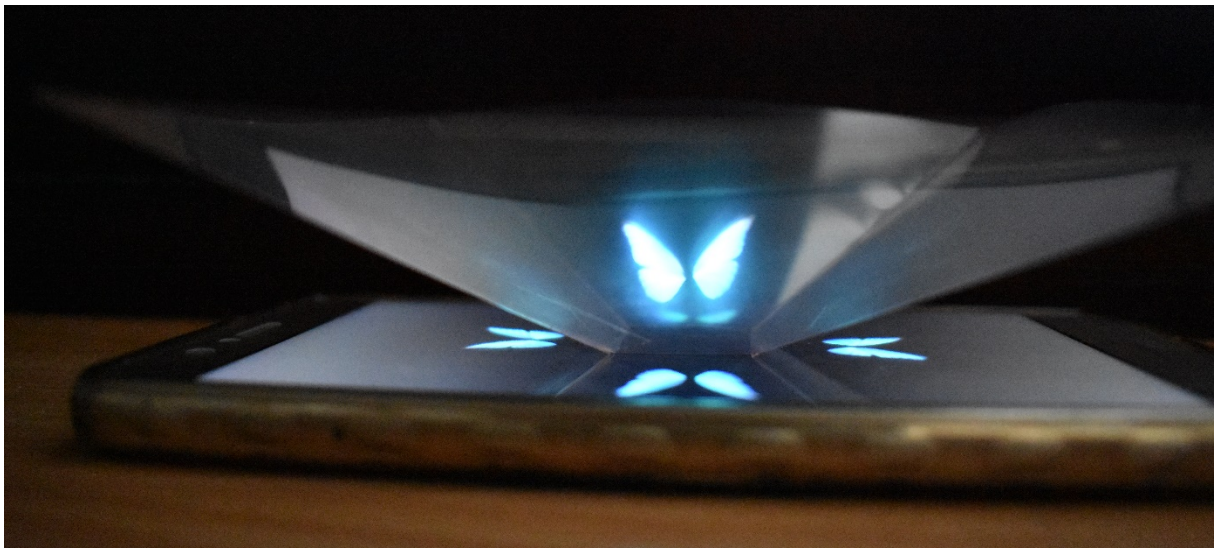
Nous avons comme objectif de modifier ou déplacer un objet virtuel holographique à l'aide d'une baguette magique.

Lien vers notre vidéo

https://dai.ly/x74395h

Introduction

Le but initial de notre projet était d'améliorer une technologie déjà existante. Certains ont déjà entendu parler des pyramides holographiques. Ces pyramides permettent d'afficher une image ou une vidéo sur un smartphone par exemple. Cette image se reflète sur les côtés de la pyramide et donne à l'image une profondeur. On a l'impression que l'image est en 3 dimensions et qu'elle flotte.



Pyramide fabriquée par une élève

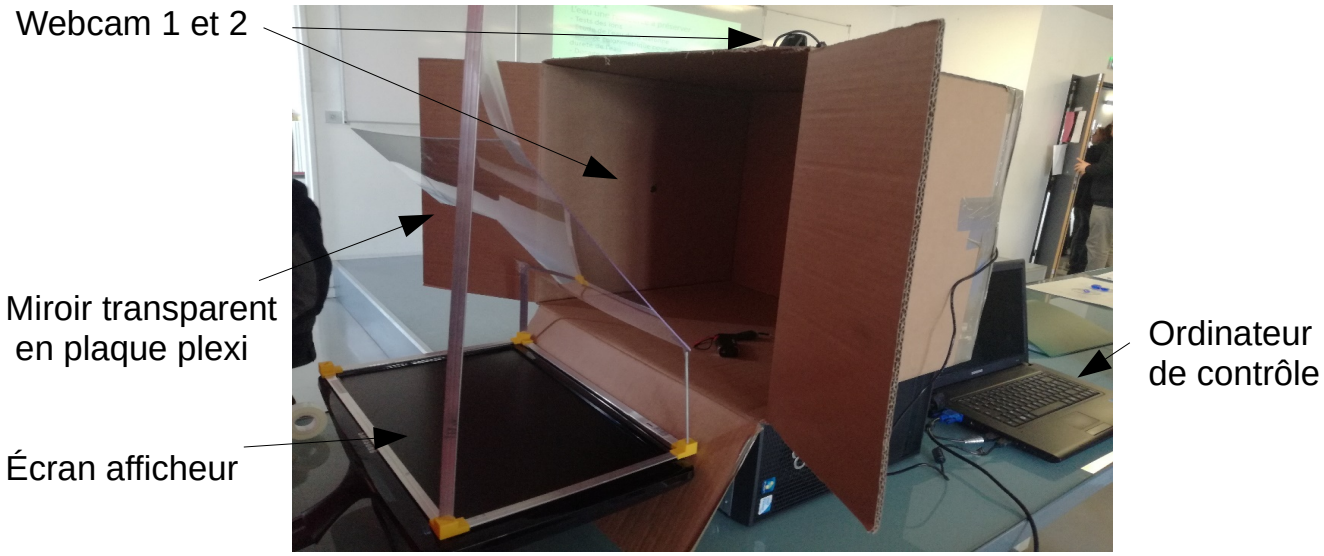
Nous avons décidé de recréer un système similaire d'une taille plus conséquente. Dans un second temps, l'idée était de profiter de cette impression de profondeur pour créer de véritables images en 3D déformables. Par exemple, en projetant un cube, l'utilisateur pourrait le manipuler pour lui donner une toute autre forme. Il était alors plus simple de ne réaliser qu'une seule face de la pyramide afin d'avoir un espace de travail libre à l'arrière.

Avec deux caméras, il est possible de connaître la position (x, y, z) d'un point dans l'espace si on place ces caméras à 90° l'une de l'autre. Les caméras les moins chères n'ont pas certains filtres et peuvent percevoir une diode infrarouge. Nous avons donc profité de cela pour fabriquer une « baguette magique ». La baguette a à son extrémité une diode infrarouge qui lorsqu'elle est allumée est repérée par les caméras. En plaçant la baguette à l'arrière de la projection de l'image, le but est de saisir une partie de l'objet projeté et de pouvoir le déplacer ou le déformer.

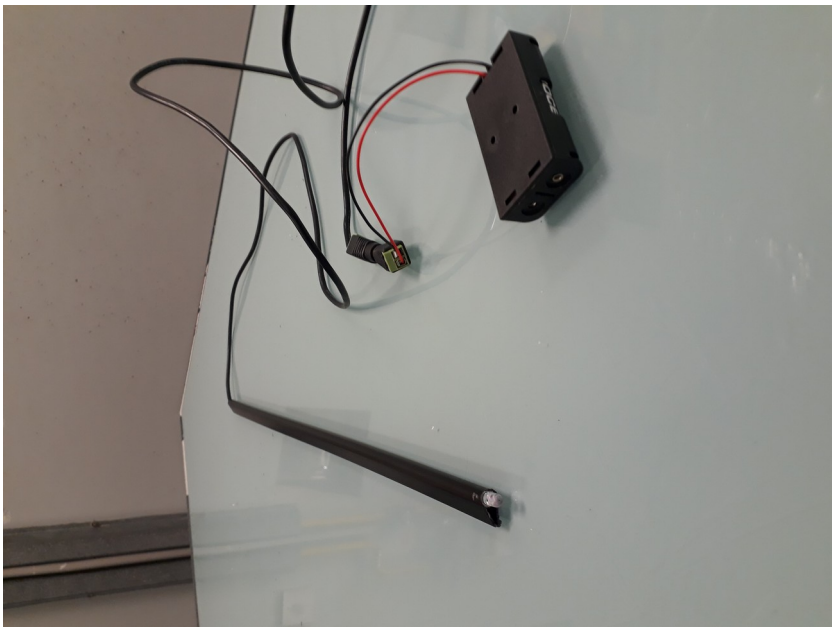
Enfin, il fallait fabriquer une structure pour faire tenir la plaque de plexiglass qui servait pour créer l'hologramme à l'angle souhaité, placer le dispositif dans l'obscurité et laisser de la place pour pouvoir agir avec la baguette tout en ayant des caméras fixes.

Présentation de l'expérience

Nous avons opté pour un carton pour reproduire des conditions d'obscurité pour améliorer la qualité de l'hologramme. Nous avons également choisi une plaque de plexiglass de 2mm d'épaisseur pour éviter que l'image ne soit double. La plaque est inclinée à 45° et placée au dessus de l'écran d'affichage. Nous avons fixé au carton deux caméras premier prix afin de pouvoir profiter de l'absence de filtres et détecter l'infrarouge. La baguette est constituée d'une LED infrarouge fixée à l'extrémité d'un bâton.



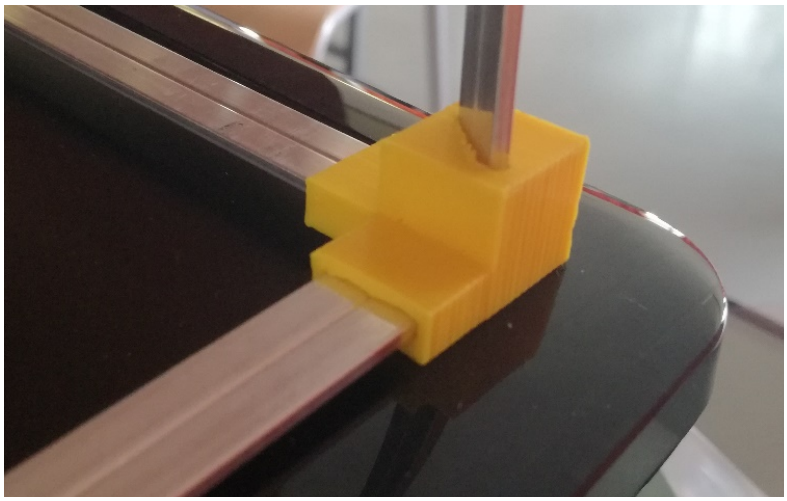
Notre premier montage



Notre baguette magique

Notre montage en détail

Pour la structure qui permet de faire tenir la plaque de plexiglass au-dessus de l'écran, nous avons scié des baguettes d'aluminium et nous les avons reliées avec des petites structures imprimées en 3D grâce à l'imprimante de notre établissement.



Structure imprimée en 3D

L'image de l'écran horizontal se reflète sur la plaque transparente et semble provenir de l'intérieur du carton.

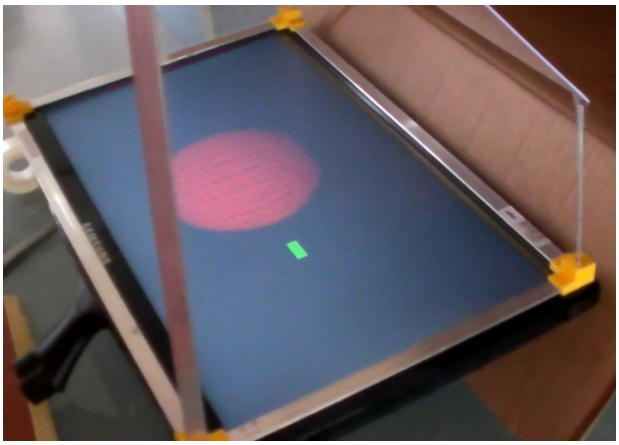


Image sur l'écran horizontal

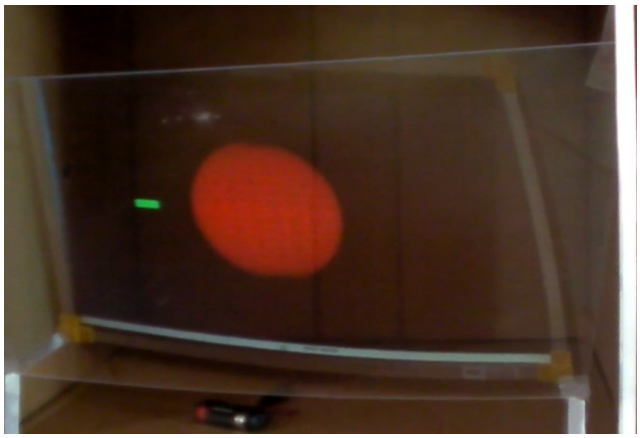
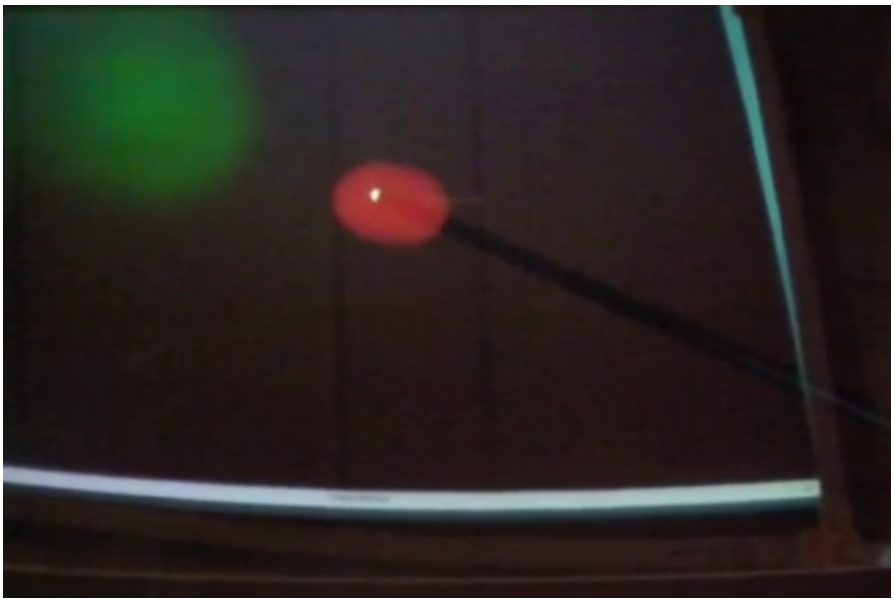


Image reflétée

Sur le carton, deux caméras placées à 90° l'une par rapport à l'autre sont reliées à l'ordinateur. Si on place une LED infrarouge au centre du carton, les caméras en captent la lumière et permettent le calcul des positions en 3D de la LED. Le logiciel recherche sur chaque flux des caméras l'endroit le plus lumineux.

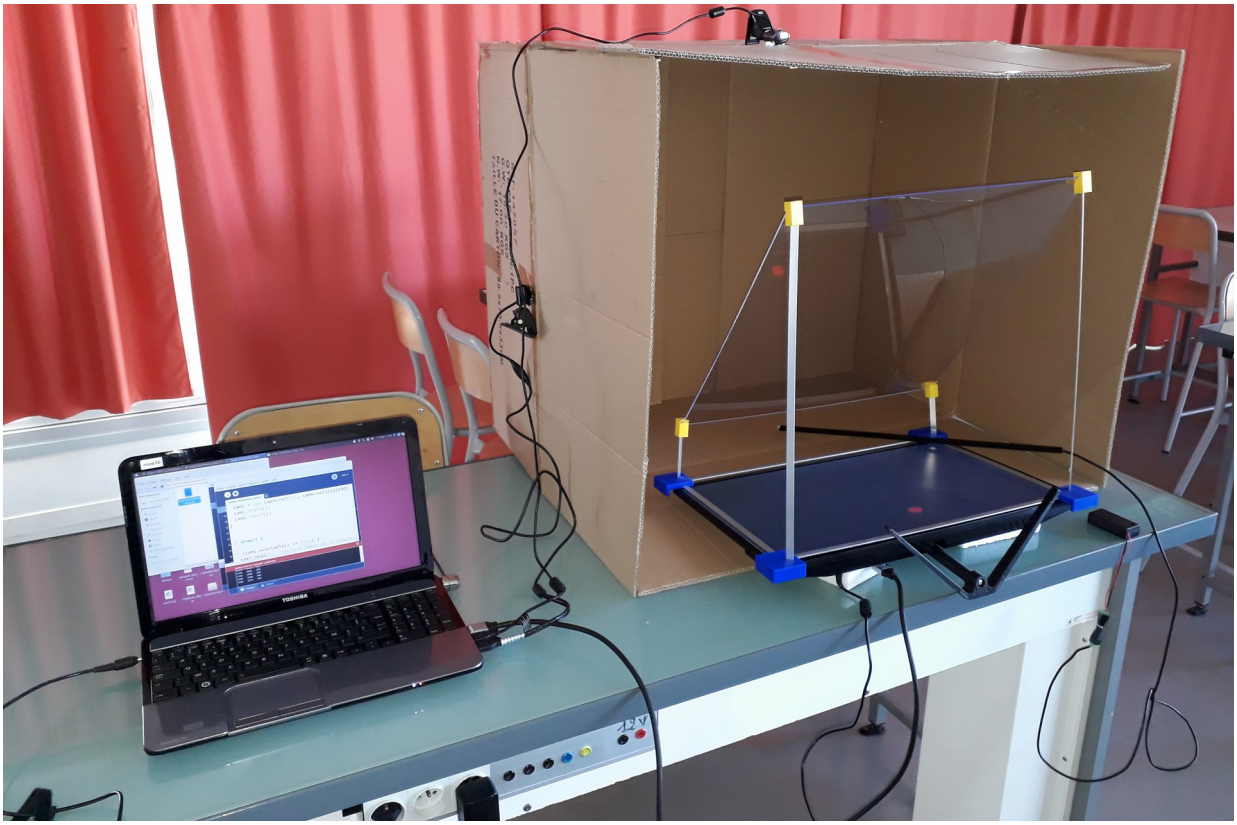
Positions de la LED capturées par les caméras

Ensuite nous avons cherché à faire déplacer une sphère virtuelle grâce à une LED , les premiers essais ont été un peu compliqués... Après un étalonnage, le résultat est assez concluant malgré les quelques soubresauts. La LED placée au bout d'une baguette permet de pouvoir facilement jouer avec la sphère.



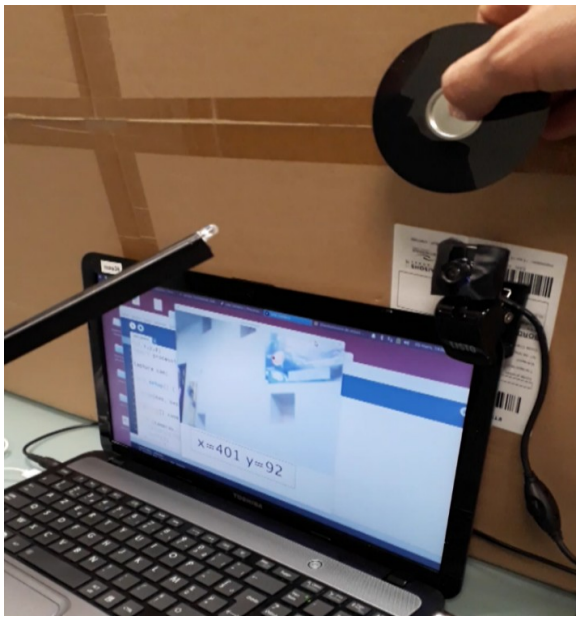
Sphère virtuelle rouge suivant la position de la LED

Après l'achat récent d'un écran plus grand, nous avons du refaire le montage avec un carton plus grand et une nouvelle structure.

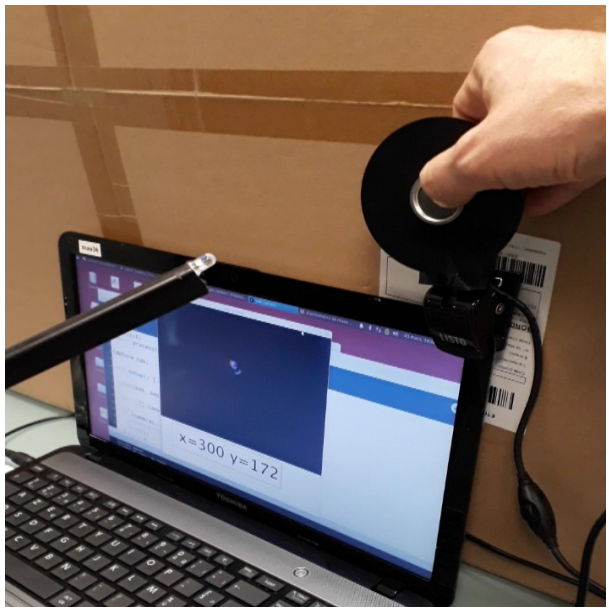


Notre dernier montage

Nous essayons de limiter l'influence de la lumière extérieure sur les caméras en utilisant le disque d'une vieille disquette. Ce disque stoppe la lumière visible et ne laisse passer que les infrarouges. Ce procédé rend le montage beaucoup moins instable.



Position de la baguette sans filtre



Position de la baguette avec filtre

Notre programmation

Nous avons utilisé Processing 3 sous linux pour coder le programme en Java. Le programme pour les caméras est le suivant :

Gestion d'une seule caméra :

```
import processing.video.*;
Capture cam ;
void setup() {
    size(1000, 800,P3D);
    String[] cameras = Capture.list();
    cam = new Capture(this, cameras[0]);      // choix de la camera
    cam.start();                              //  démarrage camera
}
void draw() {
    if (cam.available() == true) {
       cam.read();                            //lecture de l'image
      }
    float luminosite=0;
    int lumx = 0; int lumy = 0 ;
    for (int i = 0; i < 640; i++)
      for (int j = 0; j < 480; j++) {
        float lum = brightness(cam.get(i,j));  // recherche du point le plus lumineux
        if (lum>luminosite) {
            luminosite=lum;
            lumx=i;                           // lumx et lumy donne la position de la LED
            lumy=j;
         }
      }
}
```

une partie de code identique est utilisé pour la seconde caméra.

Une partie calibrage se fait avec des touches du clavier pour centre l'objet sur la LED.

```
void keyPressed(){                        //étalonnage avec les touches du clavier zedxwq
  if (key=='z' ) zeroz=zeroz+10;
  if (key=='x' ) zeroz=zeroz-10;
  if (key=='q' ) zerox=zerox-10;
  if (key=='d' ) zerox=zerox+10;
  if (key=='w' ) zeroy=zeroy-10;
  if (key=='e' ) zeroy=zeroy+10;
  println(zerox,' ',zeroy,' ',zeroz);    }    //zerox est le correctif de la position lumx
```

Quelques difficultés

Ce que nous avons su gérer :

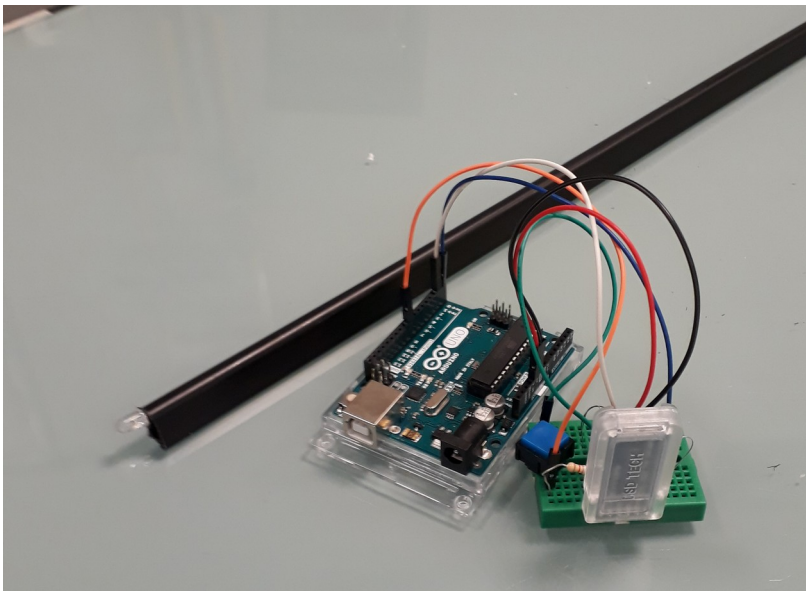
Notre principale difficulté a été d'ordre matérielle, les ordinateurs de notre lycée ne sont pas assez performants pour gérer deux caméras et une image 3D. La gestion d'une seule caméra consomme 60 % des ressources de notre ordinateur, il nous a fallu rechercher un ordinateur puissant (>i5) avec une carte graphique dédiée pour gérer chaque partie (caméra 1, caméra 2, structure 3D). De plus les webcams doivent être branchées sur un port USB 3 sinon cela ne fonctionne pas assez vite.

Notre première programmation était en langage Java, et nous avons cherché à programmer en Python. Malheureusement, au bout de deux mois d'essais, nous avons compris que ce langage consommait trop de mémoire de l'ordinateur et n'était pas adapté à notre projet. Nous avons donc repris notre programmation en Java mais nous avons perdu beaucoup de temps.

Ce que nous devons encore traiter :

Une autre difficulté est de gérer la perspective. L'observateur voit l'objet dans un champ de perspective mais les caméras donnent un champs différent. Les images au centre sont bonnes mais plus on s'éloigne du centre du carton, plus l'image virtuelle est déviée. Nous travaillerons sur ce point ces prochaines semaines. Nous ajouterons une lentille grand champ sur chaque webcam pour augmenter la zone active, et nous calculerons et corrigerons la distorsion avec le programme informatique.

En parallèle, nous essayons de fabriquer une baguette avec un ou deux boutons, comme pour une souris. Nous utilisons un Arduino et un dispositif Bluetooth pour s'affranchir d'un fil.



La nouvelle baguette Bluetooth

Perspectives

Notre projet fonctionne correctement. Nous devons encore le rendre plus esthétique en peignant en noir l'intérieur et en décorant l'extérieur.
Maintenant que la gestion 3D est opérationnelle, nous travaillons des applications comme un jeu de pêche à l'épuisette virtuelle ou encore de la création 3D où la structure que nous créons pourra être imprimée grâce à notre imprimante 3D…

Partenariats

Nous remercions Jérémy INCONSTANTE de la société Thales, pour ses discussions et conseils lors d'un échange dans le cadre des Savanturiers.
Nous remercions également le fablab EIRLAB de l'ENSEIRB-MATMECA pour la conception de pièces.

Annexe : programme final

```
//intialisation des variables en global
int x,y,z; //position de la sphère
int zerox,zeroy,zeroz; //variable de réglage
float x1; //rayon de la sphère

import processing.video.*; //bibliothèque de la caméra,

Capture cam1, cam2; //initialisation des caméras cam1 en haut cam2 à gauche

void setup() {

  size(1920, 1080,P3D); // taille adapté à l'écran HD
 //initialisation des réglages
 zerox=1260;
 zeroy=1070;
 zeroz=300;

  String[] cameras = Capture.list(); //liste des caméras disponibles
  //test des caméras
  if (cameras.length == 0) {
    println("There are no cameras available for capture.");
    exit();
  } else {
    println("Available cameras:");
    for (int i = 0; i < cameras.length; i++) {
      println(i,cameras[i]); //affiche la liste des caméras disponibles
    }
    cam1 = new Capture(this, cameras[2]); //caméra haute
    cam2 = new Capture(this, cameras[1]); //caméra gauche
    cam1.start();
    cam2.start();
  }
} //fin du setup

void draw() {

  if (cam1.available() == true) {
    cam1.read(); //lecture image de la caméra 1
     }
  //recherche du point de luminosité maximale de l'image 1
  float luminosite1=0;
  int lumx1 = 0; int lumy1 = 0 ; //initialisation de la position du point le plus lumineux
  for (int i = 0; i < 640; i++)
     for (int j = 0; j < 480; j++) {
       float lum1 = brightness(cam1.get(i,j));
       if (lum1>luminosite1) {
         luminosite1=lum1;
         lumx1=i;//position x du point le plus lumineux
         lumy1=j;//position y du point le plus lumineux
       }
     }
```

```
  if (cam2.available() == true) {
    cam2.read();
     }
  float luminosite2=0;
  int lumx2 = 0;
  int lumy2 = 0 ;
  for (int i = 0; i < 640; i++)
     for (int j = 0; j < 480; j++) {
       float lum2 = brightness(cam2.get(i,j));
       if (lum2>luminosite2) {
         luminosite2=lum2;
         lumx2=i;
         lumy2=j;
       }
     }

  //calcul des positions de la sphère
   x=zerox-lumx2;
   y=zeroy-lumy1;
   z=zeroz-lumy2;
  //affichage de la sphère
  background(0,0,0);
  translate(x,y,z);
  fill(255,0,0);
  sphere(40);
} //fin du draw

void keyPressed(){

 //réglage initial de la position de la spère
  if (key=='z' ) zeroz=zeroz+10;
  if (key=='x' ) zeroz=zeroz-10;
  if (key=='q' ) zerox=zerox-10;
  if (key=='d' ) zerox=zerox+10;
  if (key=='w' ) zeroy=zeroy-10;
  if (key=='e' ) zeroy=zeroy+10;
  println(zerox,' ',zeroy,' ',zeroz);
}
```