Math-en-Jeans - 2021/2022

Conception d'un filtre facial

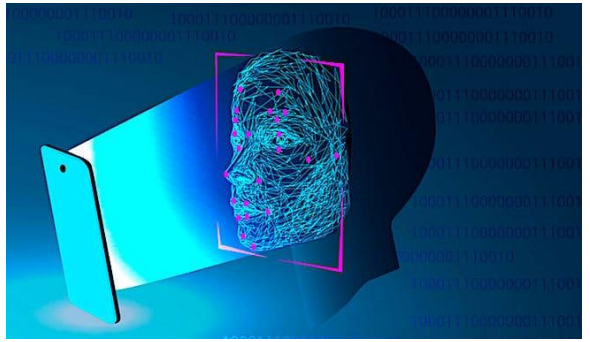
Nour A. ; Laure T. ; Dorian A. ; Myrte G. ; Anastasiia B. ; Clement K. ; Yann S. ; Julie L. ; Sarah R.G. et Quentin L.

Chercheur : Jordan Frecon, chercheur en intelligence artificielle à l'université de Rouen.
https://jordan-frecon.com/

Sommaire

Introduction

Conception d'un filtre facial :



- Très présente dans notre quotidien
- Développement d’un algorithme pour identifier le visage d’une personne
- Développer des filtres

Objectifs :

- Développer des techniques efficaces pour identifier le visage d’une personne sur une photographie
- Extraire les attributs du visage et les traiter
- Étudier comment étendre ce formalisme aux vidéos
- **Problématique : *Comment identifier le visage d'une personne sur une photographie ?***

Evolution du 1°groupe

Dans le premier groupe, nous avons eu différentes pistes de recherche. Tout d'abord, nous avons commencé à étudier la définition d'une ellipse, et pour ce faire, nous avons développé et factorisé son équation. Ensuite nous avons réalisé plusieurs programmes, un qui balaye la photographie et l'autre qui s'adapte à la forme du visage.

Nous avons également cherché à déterminer le contour du visage à l'aide d'une forme géométrique et à trouver le visage en changeant les contrastes et les couleurs. Après nous avons eu un appel avec le chercheur Jordan Frecon, qui nous a permis de trouver de nouvelles idées. Pour finir, nous avons étudier les zones d'ombres du visage.

Evolution du 2°groupe

Quant au second groupe, nous avons dans un premier temps, recherché les caractéristiques importantes du visage. Suite à l'appel avec le chercheur Jordan Frecon, nous avons eu une nouvelle approche. Elle était de créer un programme de balayage qui donnera l'opportunité de trouver le visage et de comprendre les propriétés d'une ellipse. Nous étions avant cela sur une fausse piste qui était de trouver le milieu d'une ellipse dans un repère cartésien. Pour finir nous avons mis en commun les différents travaux de recherches.

Sous-groupe 1: Etude de l'ellipse

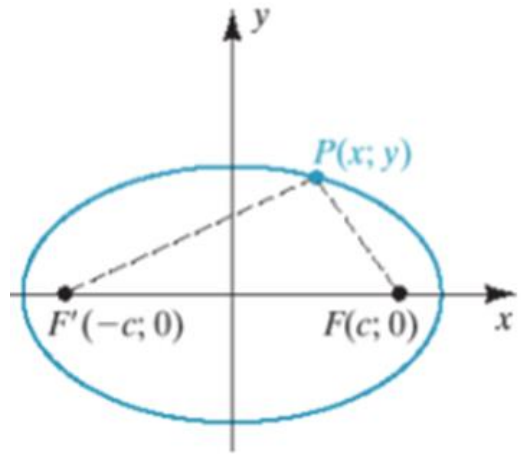
La forme la plus proche de celle du visage étant l'ellipse, nous avons commencé par étudier ses caractéristiques. Nous avons retrouvé l'équation cartésienne d'une ellipse centrée à l'origine (coordonnées : (0;0)) d'un repère orthonormé: $\frac{x^2}{a^2} - \frac{y^2}{b^2} = 1$

Nous avons observé qu'une ellipse est caractérisée par ses deux foyers F et F' respectivement de coordonnées (-c;0) et (c;0). Ainsi, un point P(x;y) appartient à l'ellipse si et seulement si d(P,F) + d(P,F') = 2a .

Problème soulevé :

Sur le visage, à quoi peuvent correspondre les deux foyers de l'ellipse ?

L'origine du repère ?

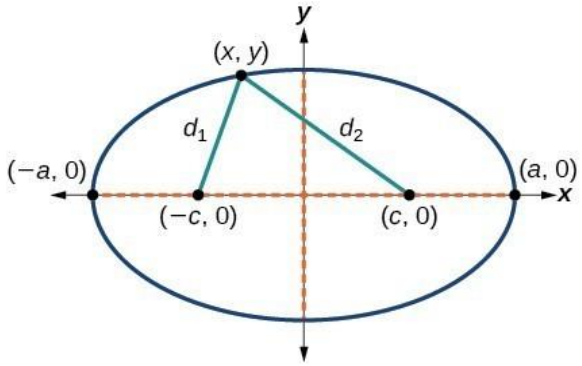


Sous-groupe 1

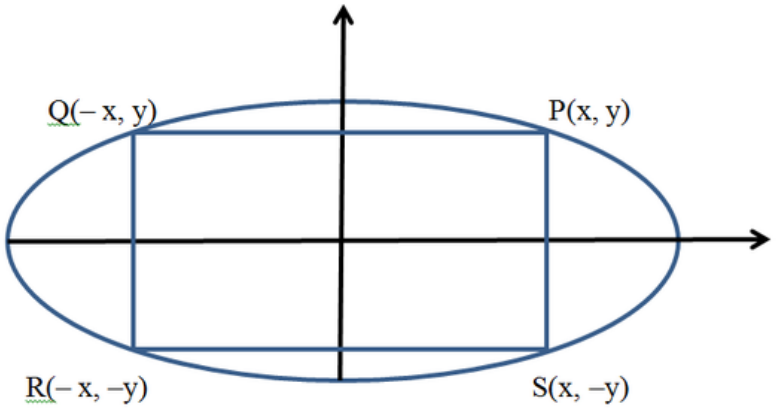
Nous avons choisi la forme de l'ellipse car c'est celle qui se rapproche le plus du visage.

- Pour comprendre le fonctionnement et la création d'une ellipse, nous avons réalisé des calculs, et nous avons essayé de reproduire la forme par coups d'essai sur des logiciels géométriques tel que géogebra

- Ensuite, nous avons étudié des proportions de visage en dessin afin de trouver tous les points clefs et pertinents que nous devrons identifier sur le visage.



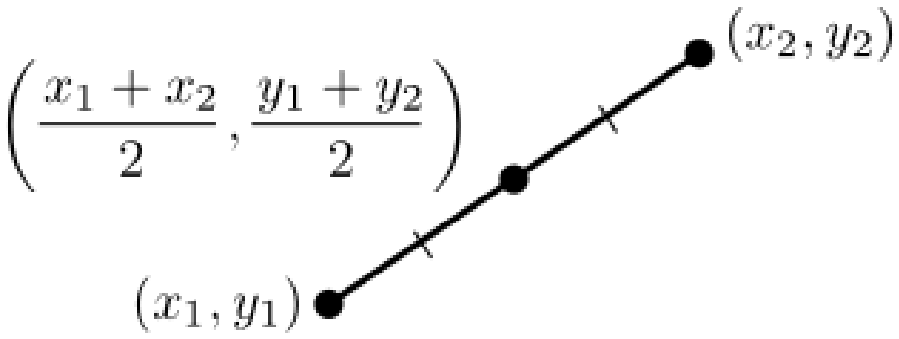
On cherche des calculs de rectangles inscrits dans les ellipses car nous avons réalisé que le rectangle passe par tous les points clefs du visage.

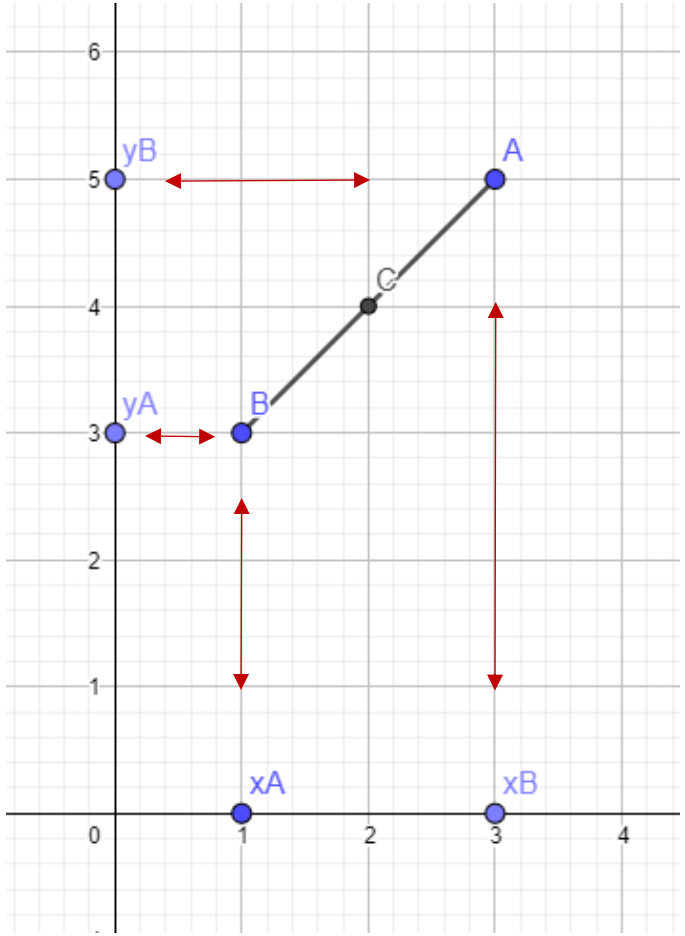


Sous-groupe 1: Le centre du visage

Le nez étant le point le plus important à identifier sur le visage car il est son centre, nous cherchons à trouver le centre du visage:

Pour cela nous allons trouver les coordonnées du milieu d'un segment dans un repère orthonormé. On fait ces calculs pour les deux axes du visage. Ils sont perpendiculaires l'un à l'autre.

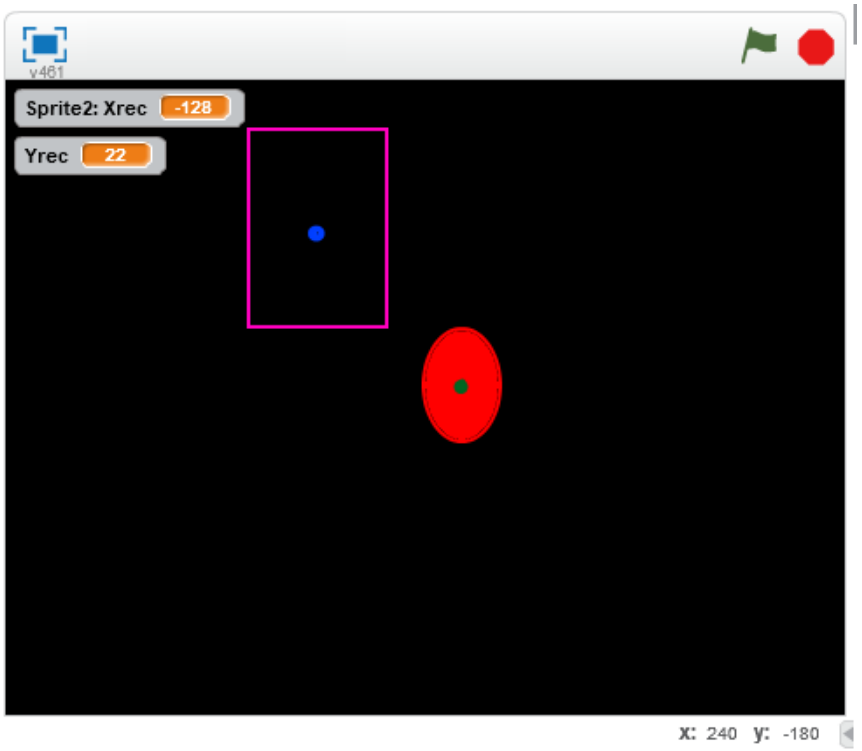




Sous-groupe 2: Le balayage

Afin de réaliser le balayage, nous avons tout d'abord réalisé un premier programme qui n'a malheureusement pas fonctionné puisque il était trop rapide ce qui a aboutis à un échec.

Dans un second temps nous avons pensé à un autre programme qui était un encadrement de l'ellipse. Afin de le réaliser nous avons utilisez un point de couleur pour représenter le nez et utilisé un rectangle qui entoure le point de sorte que le point soit au centre du rectangle.



Plusieurs programmes

1° essai

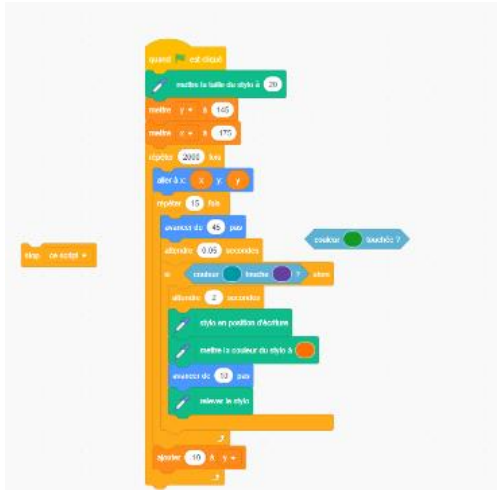
Afin de réaliser ce projet, on a dû réaliser plusieurs programmes. On a commencé avec un programme de balayage qui fonctionne sur le système d'un capteur. Il y a alors un point rouge qui va de gauche à droite sur l'écran. Dès qu'il touche une zone noire, le stylo est mis "en position d'écriture" et un point orange est placé (les lignes sont le résultat de la succession de points).

Avant d'aboutir à ce résultat final, on pensait que le capteur devait détecter une couleur en particulier afin de, par la suite, tracer les contours de la forme. Cependant, le capteur ne détectait pas toujours le point violet avec précision. Ce programme n'a donc pas abouti.

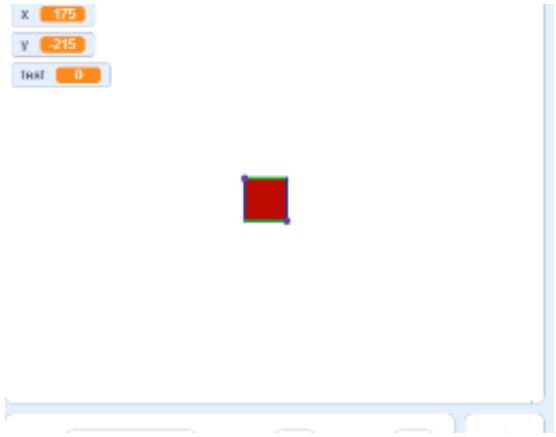
1° essai

Résultat du programme de balayage.

Script du programme de balayage.



Programme et résultat du programme qui n'a pas abouti.



2° essai : Programme sans suite

Pendant ce temps, on a cherché à compléter le programme de balayage avec un programme qui permettrait d'adapter la taille du capteur (si représenté par un rectangle/carré) à la forme détectée par le programme précédent.

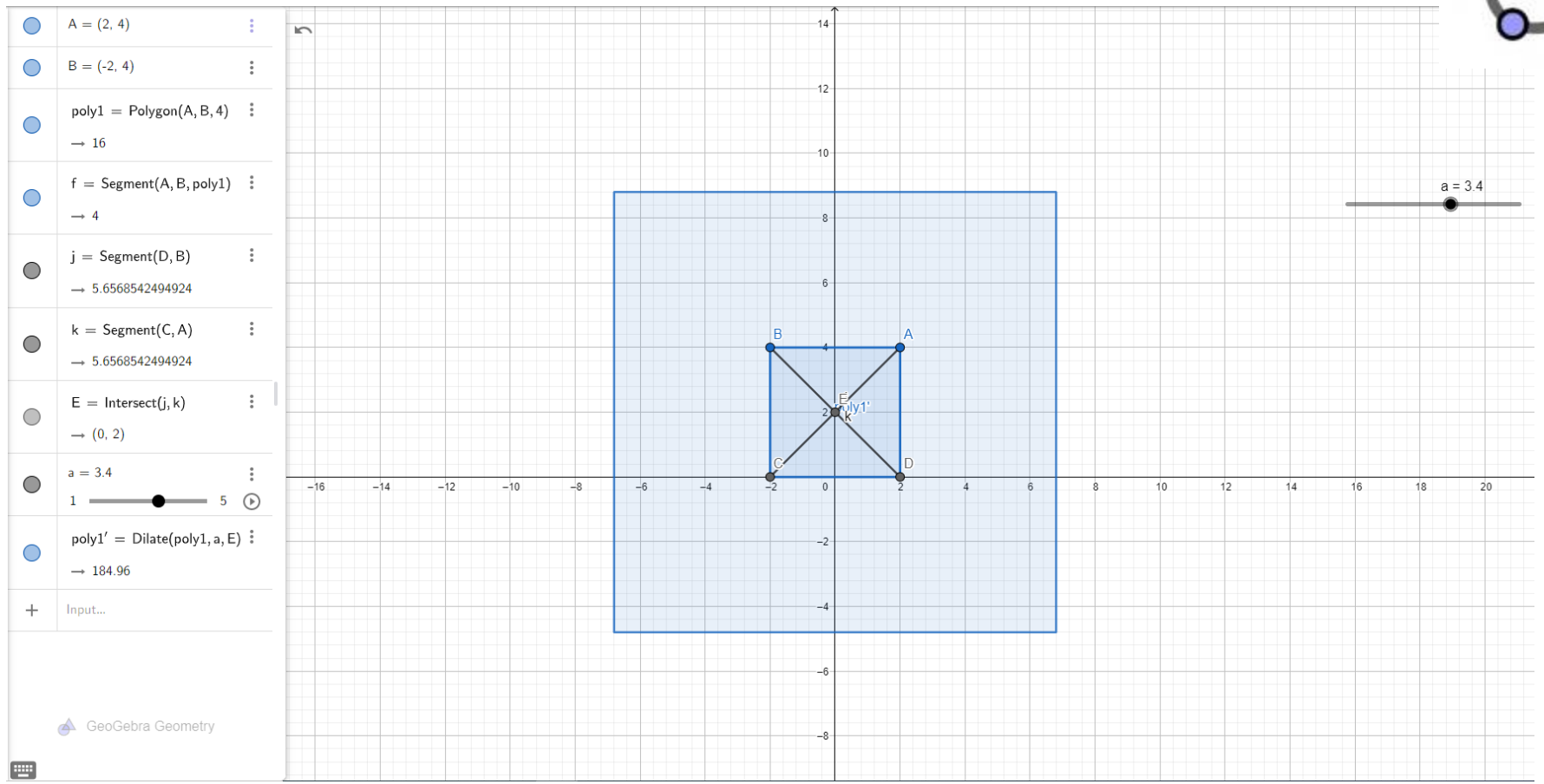
Pour cela, on a modélisé sur Geogebra le visage à détecter par un carré (par souci de simplicité) et le capteur par un carré également de côté a, modifiable par un curseur qu'on faisait défiler.

Problème soulevé : Ce programme nécessiterait que le capteur soit déjà centré sur le carré, et donc, le visage. Comment ?

Objectif : On a décidé de finaliser le programme de balayage.

Finalement, ce programme d'adaptation n'a pas abouti. On a privilégié d'"encadrer" le visage à l'aide de droites partant des bordures de l'écran Scratch.

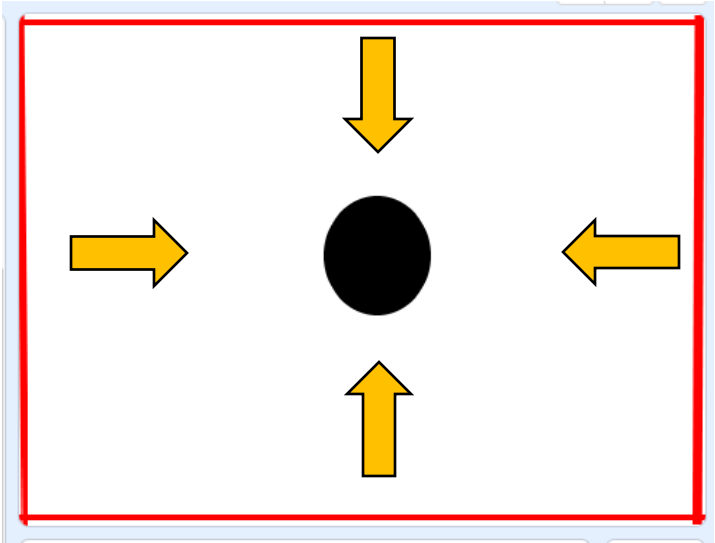
2⊙ essai : Programme sans suite



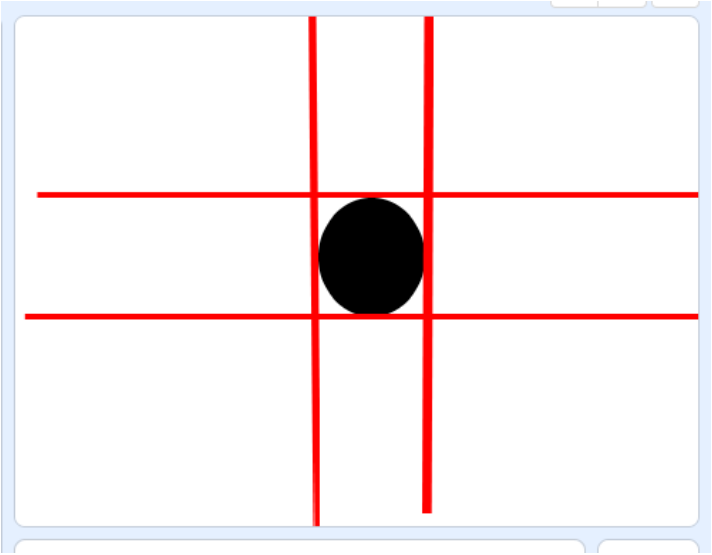
3© essai : Encadrer et définir le visage avec des droites

Notre troisième essai consistait à créer un programme Scratch qui va pouvoir reconnaitre et encadrer la forme du visage grâce à quatre droites. Au centre se trouve une ellipse représentant la tête. Sur les côtés se trouvent des droites rouges qui vont au fur et à mesure encadrer le visage. Cet encadrement est possible grâce à la détection du changement de couleur.

Début :

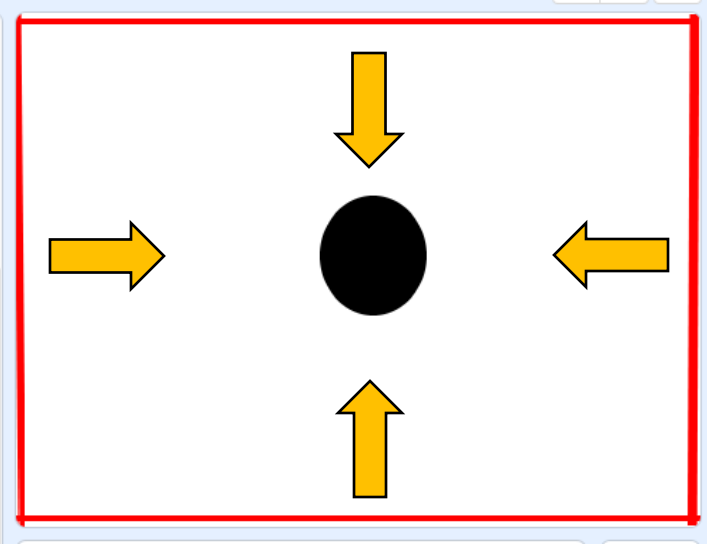


Fin :

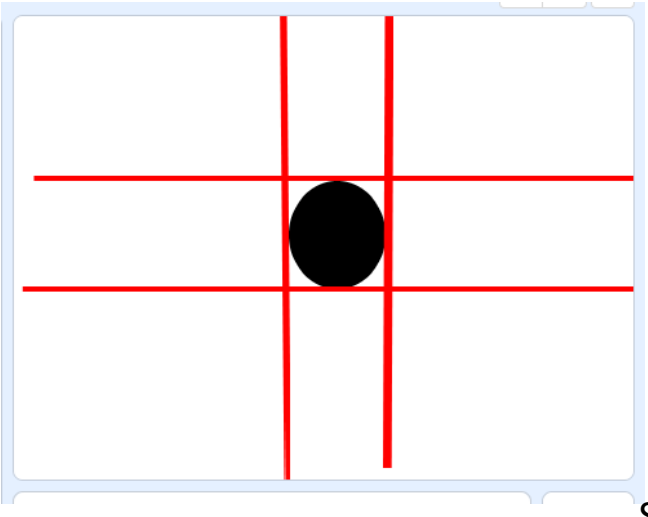


3° essai : Encadrer et définir le visage avec des droites

Début :

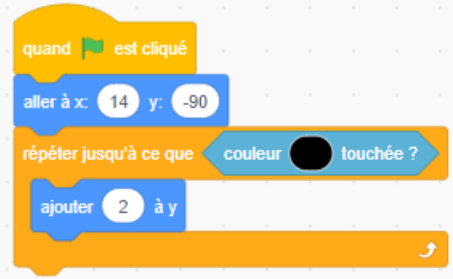


Fin :



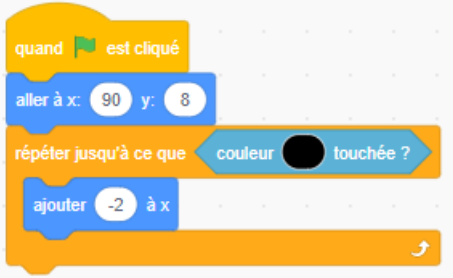
Script de la droite du haut :

Script de la droite de bas :



Script de la droite à droite :

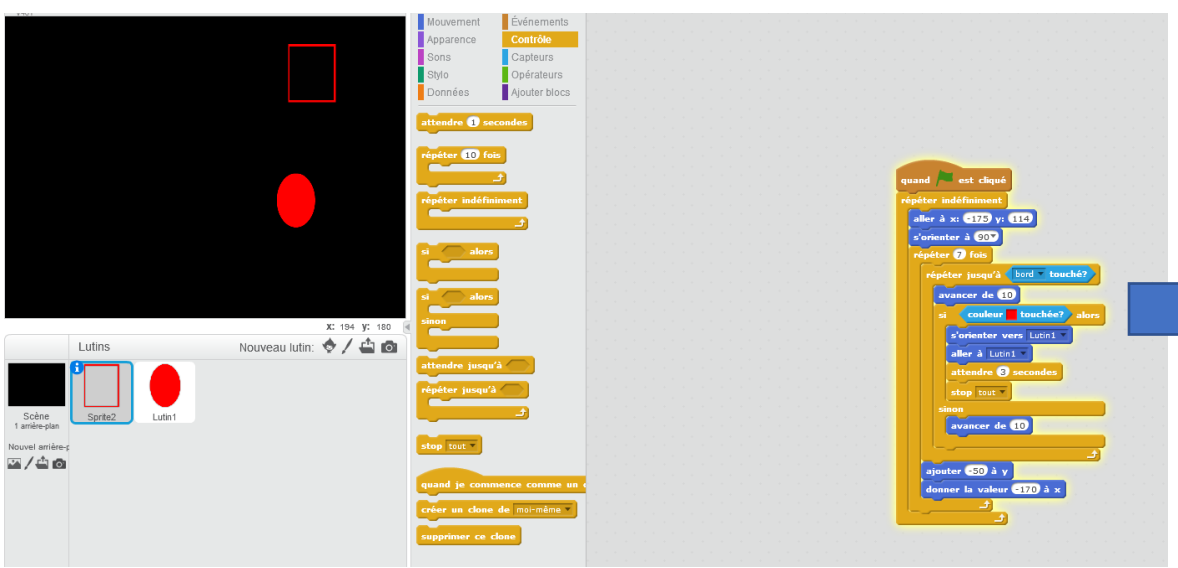
Script de la droite à gauche :

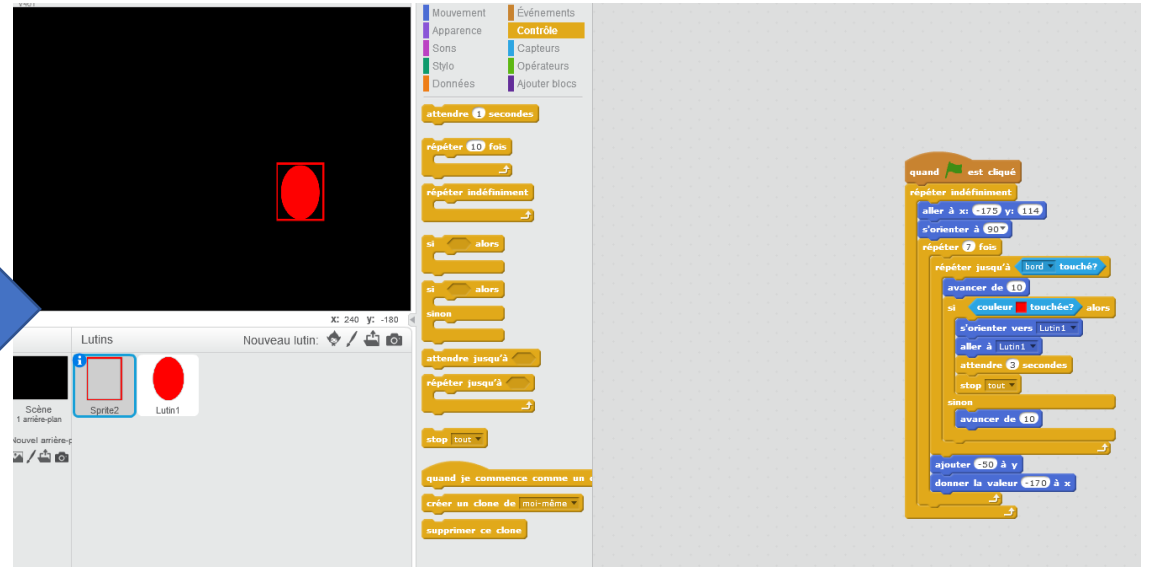


- Ellipse noire = la tête
- Droites rouges permettent d'encadrer le visage.
- Flèches jaunes représentent les mouvements des droites.

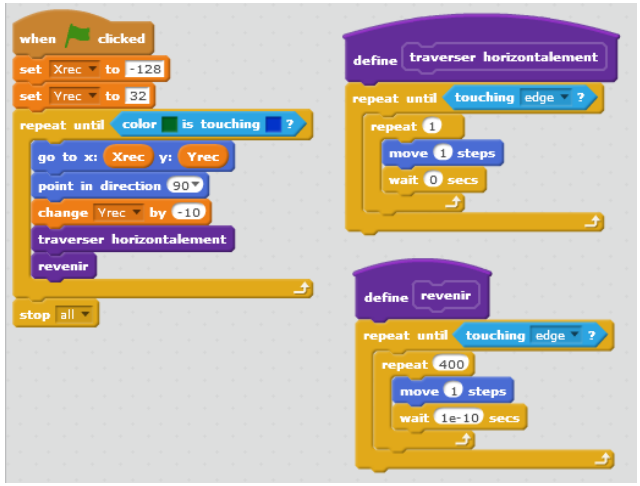
Premier prototype

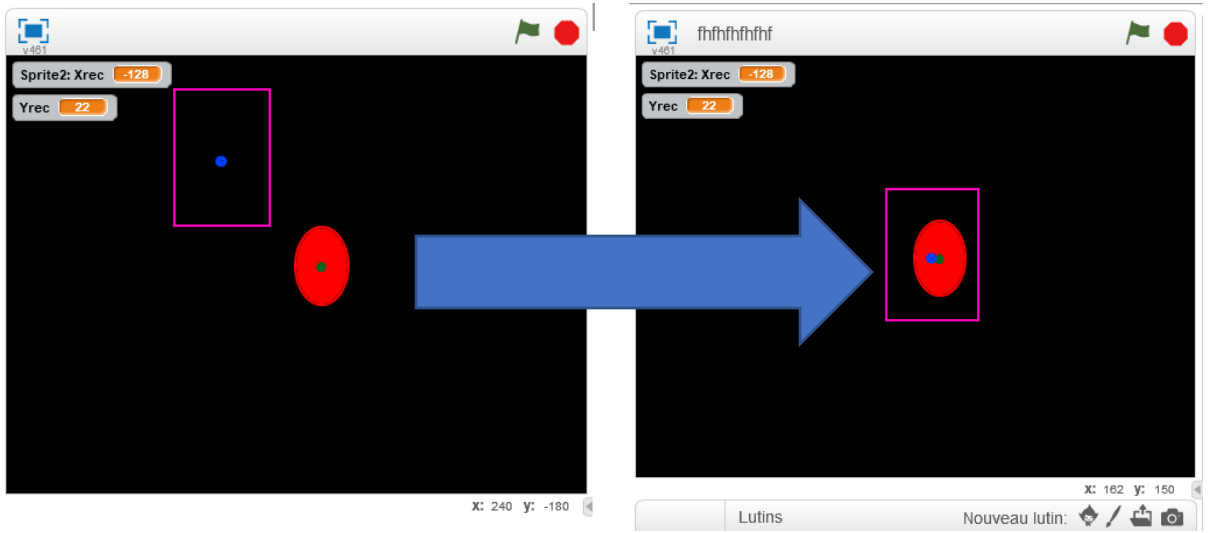
Le rectangle rouge balaye l'écran jusqu'à ce qu'il trouve le visage (cercle rouge) et s'y accroche. S'il ne trouve pas de visage, il recommence jusqu'à en trouver un.





Deuxième prototype





Le centre du rectangle (point bleu) balaye l'écran jusqu'à toucher le centre de l'ellipse (point vert). Cela permet d'avoir plus de précision, puisque le rectangle s'arrête presque au milieu de l'ellipse.

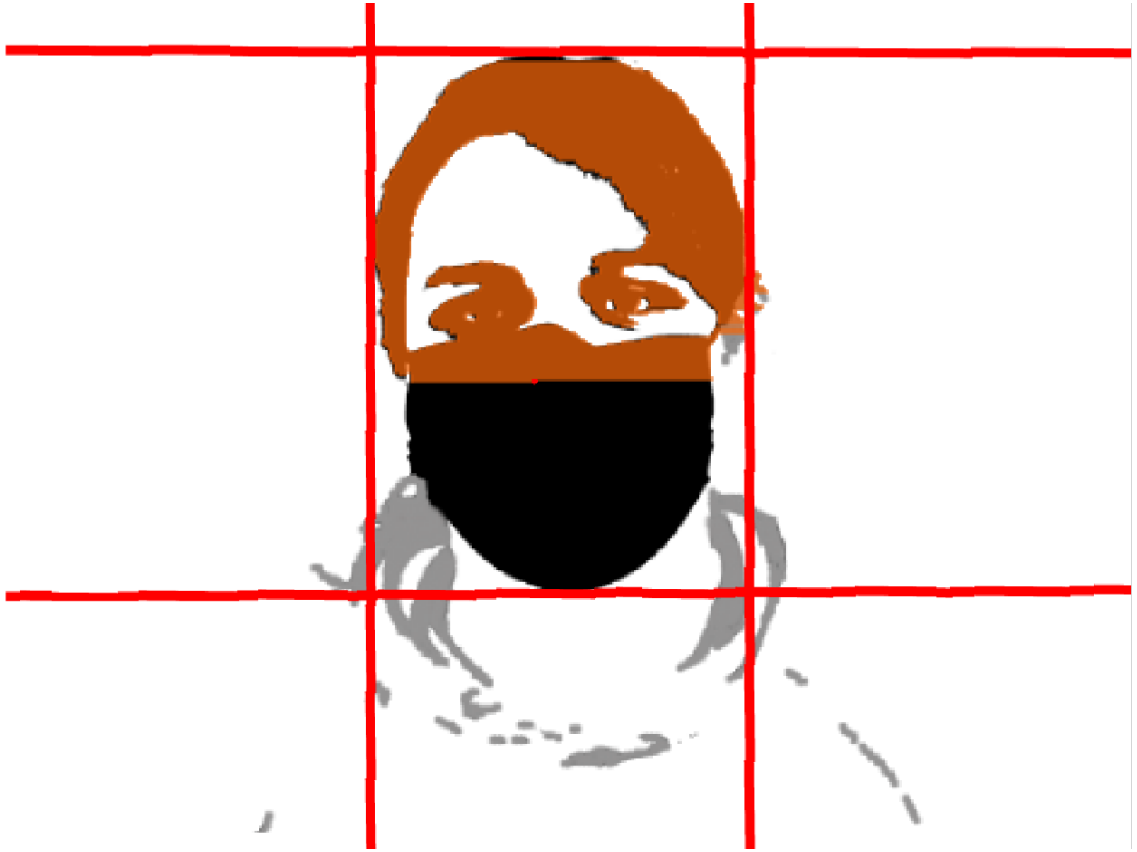
Résultat final

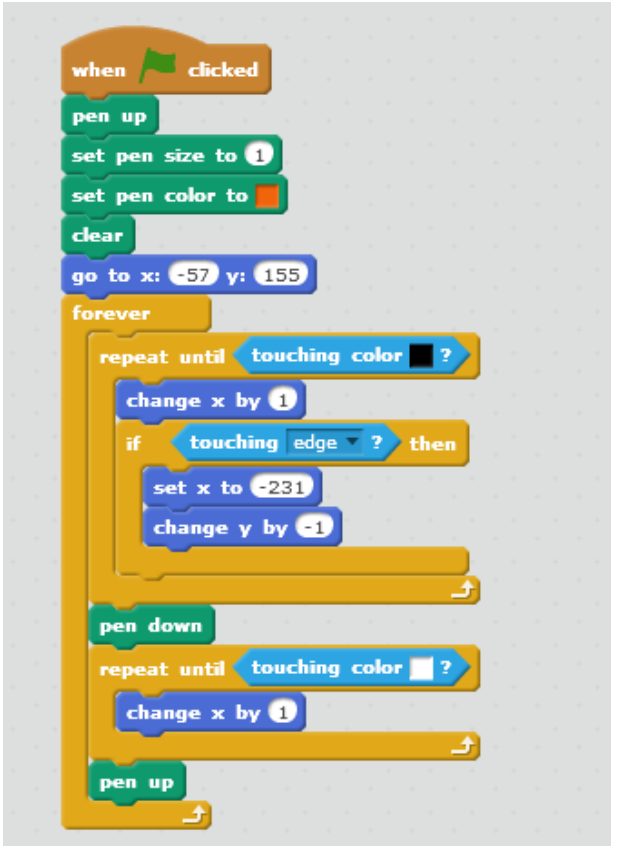
Pour notre programme final, on a décidé de combiner deux programmes: un qui délimite les contours du visage, l'autre qui "détecte" les caractéristiques faciales. On voulait utiliser Snap Berkeley pour réaliser ce programme final car utiliser du code permet une meilleure performance.

Cependant, Snap Berkeley ne proposait pas toutes les fonctionnalités nécessaires. On a donc fini par utiliser Scratch.

Ensuite, on est satisfait de notre résultat car notre programme permet de détecter le visage d'un individu ainsi que les éléments importants de ce dernier (caractérisés par des zones d'ombre).

Résultat final





Améliorations possibles

Nous avons donc imaginé diverses améliorations possibles.

Nous avons tout d'abord pensé à un premier programme qui modifie directement les contrastes de la photo. Ainsi il transformera le visage en noir et blanc pour que le programme final puisse être utilisé.

Nous avons ensuite pensé à utiliser Python plutôt que Scratch, ce qui permettrai d'accélérer la vitesse d'exécution et donc le nombre de tests possibles.

Une seconde amélioration consiste à limiter l'utilisation du contraste blanc-noir, ce qui serait en contradiction à notre première idée. Mais cela permettrait au programme d'être utilisé avec diverses photos comprenant des fonds et des objets plus ou moins complexes.

Conclusion

Afin de répondre à la question "Comment identifier le visage d'une personne sur une photographie ?", nous avons donc, au cours de cette année réalisé notre programme final. Ce dernier, utilise le contraste de la photo prise afin de détourer le visage.

Malheureusement cette méthode a deux principales limites :

- La nécessité d'avoir un arrière-plan neutre afin de ne pas avoir de problème avec le contraste.

- De plus il faut qu'il n'y ait aucun objet présent sur l'image, ce qui pourrait fausser le programme pour la même raison donnée précédemment puisque cela créera des problèmes au niveau du contraste également.