|  |
| --- |
| Flexymove |
| Fiche de suivi n°2 |
| Nicolas Berthelot – Tom Nasri – Raphaël Daguenet  Version 1  7 Pages  24/02/2011  PROTOTYPE-CAHIER-DES-CHARGES-PLANNING |



Propriétés du document

|  |  |
| --- | --- |
| Auteur | Nicolas Berthelot – Tom Nasri – Raphaël Daguenet |
| Version | 1 |
| Nombre de pages | 12 |
| Références | PROTOTYPE-CAHIER-DES-CHARGES-PLANNING |

Historique du document

|  |  |  |  |
| --- | --- | --- | --- |
| Date de révision | Version | Auteur | Changements |
| 24/02/2011 | 1.0 | Nasri Tom |  |

Sommaire

[Introduction 4](#_Toc286824872)

[Tâche à réaliser 5](#_Toc286824873)

[Prototype 5](#_Toc286824874)

[Algorithme 5](#_Toc286824875)

[Sélection 8](#_Toc286824876)

[Déplacement du mini-Raphi 8](#_Toc286824877)

[Cahier des charges 9](#_Toc286824878)

[Partenariat 9](#_Toc286824879)

[Conclusion 10](#_Toc286824880)

# Introduction

Le présent document est le document d’avancement du 16 février au 2 mars 2011 de la PLIC Adobe.

Il regroupe le travail effectué durant cette période sur le projet flexymove et le partenariat Epita/Adobe.

# Tâche à réaliser

Les tâches à réaliser durant ces trois semaines sont tout d’abord d’écrire le cahier des charges pour notre projet technique. Le prototype que nous avions commencé à développer durant le suivie numéro 1 devra quant à lui être plus développé. Nous avions déjà pu identifier des couleurs comme le rouge, le bleu, le vert, le jaune et le rose. Le but de ce suivi est d’identifier une seul couleur (donc un seul doigt) puis de la suivre afin de pouvoir déplacer un ou plusieurs carrés.

# Prototype

## Algorithme

Notre prototype dispose d’un algorithme de détection très simple, nous avons donc cherché à l’améliorer pour que la détection soit plus performante.

Dans l’exemple ci-dessous l’algorithme détecte 3 pixels sur l’image. Il calcul ensuite le barycentre de ces trois pixels. Le barycentre est le résultat du traitement, c’est donc notre point détecté.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | Pixels détectés | | | | |
|  |  |  | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  | X | Barycentre des | | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | pixels détectés | | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

Voici le code source de l’algorithme qui consiste simplement à parcourir chaque pixel puis a déterminé si il est dans une certaine plage de couleur :

for (i = 0; I < img.width; i++ )

{

for (j = 0; j < img.height; j++)

{

RGB = img.getPixel(i, j);

R = getR(RGB);

G = getG(RGB);

B = getB(RGB);

if (r\_min <= r && r <= r\_max &&

g\_min <= g && g <= g\_max &&

b\_min <= b && <= b\_max)

img.setPixel(i, j, 0);

}

}

Cet algorithme a pu être optimisé en utilisant une technique de groupe. Avec ce nouvel exemple, le premier algorithme ne trouve pas un bon résultat :

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | Pixels détectés | | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  | X | Barycentre des | | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | pixels détectés | | | | |
|  |  |  |  | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

Avec la première méthode, on peut voir que le point détecté n’est pas à la bonne place. Il suffit qu’il y est d’autres points bleu dans l’image et le barycentre est faussé.

Le deuxième algorithme utilise une méthode de regroupement de pixels. Cette méthode utilise une matrice de la taille de l’image dans laquelle sont stockés les numéros de groupe de chacun des pixels. Ils sont initialisés à 0.

Il parcourt chaque pixels, s’il en trouve un qui match la couleur, il l’étiquette avec un numéro de groupe.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | Pixels détectés | | | | |  |
|  |  | 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  | X | Barycentre des | | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | pixels détectés | | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  | 8 | numéro de groupe | | | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

Ensuite il commence un parcours en largeur des pixels autour de lui, puis étiquette tous les pixels (qui match la couleur) du même numéro de groupe.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | Pixels détectés | | | | |  |
|  |  | 1 | # |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | # | # |  |  |  |  |  |  |  |  |  |  | X | Barycentre des | | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | pixels détectés | | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  | 8 | numéro de groupe | | | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  | # | pixel autour qui | | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | match la couleur | | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | niveau 1 | | |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

Puis continue le parcourt en largeur sur ces pixels étiquetés.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | Pixels détectés | | | | |  |
|  |  | 1 | 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | 1 | 1 | # |  |  |  |  |  |  |  |  |  | X | Barycentre des | | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | pixels détectés | | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  | 8 | numéro de groupe | | | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  | # | pixel autour qui | | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | match la couleur | | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | niveau 2 | | |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

Le parcourt en largeur s’arrête lorsqu’il n’y a plus de pixel qui match la couleur.

La parcourt de tous les pixels continu seulement sur les pixels non-étiqueté et le numéro de groupe augmente à chaque parcourt en largeur. Voici le résultat :

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | Pixels détectés | | | | |  |  |
|  |  | 1 | 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | 1 | 1 | 1 |  |  |  |  |  |  |  |  |  | X | Barycentre des | | | | |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | pixels détectés | | | | |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  | 8 | numéro de groupe | | | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  | 2 | 2 |  |  |  | 3 |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  | 3 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

Ensuit il suffit de trouver le groupe le plus grand et de prendre le barycentre des pixels de ce groupe.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | Pixels détectés | | | | |  |
|  |  | 1 | 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | 1 | X | 1 |  |  |  |  |  |  |  |  |  | X | Barycentre des | | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | pixels détectés | | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  | 8 | numéro de groupe | | | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  | 2 | 2 |  |  |  | 3 |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  | 3 |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

Grace à cet algorithme nous avons pu élargir notre plage de couleur détecté.

## Sélection

Lors de la détection d’un groupe de couleur nous avons le nombre de pixels qui forme ce groupe. Grâce à ce nombre de pixels nous avons une information sur la distance entre la webcam et l’objet coloré (le gant). Cela nous permet de simuler un clic, une première utilisation du clic nous permet de dessiner.

## Déplacement du mini-Raphi

Maintenant que nous arrivons à dessiner en simulant un clic, nous avons décidé d’ajouter une fenêtre dans laquelle nous pouvions déplacer un objet.

# Cahier des charges

Df

df

# Partenariat

Concernant le partenariat avec Adobe nous avons commencé les démarches pour organiser une réunion de présentation.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Tom Nasri | Raphaël Daguenet | Nicolas Berthelot |
| Algorithme |  |  |  |
| Sélection |  |  |  |
| mini-Raphi |  |  |  |
|  |  |  |  |

# Conclusion

Nous …..