

```

function color=cameracolor(cam)

% Fonction pour detecter les colors
%
% Usage : color= cameracolor
% Entrée : cam : le nom de caméra
% Sortie : color:les valeur de color (0, 1, 2, 3)
%
% Définition des variables :
%     img      : Photo pris par caméra
%     n        : Le nombre de pixels utilisé chaque photos
%     R        : La photo dans le channel rouge
%     G        : La photo dans le channel vert
%     B        : La photo dans le channel bleu
%     value_R  : La valeur moyenne d'éléments de la photo au channel rouge
%     value_G  : La valeur moyenne d'éléments de la photo au channel vert
%     value_B  : La valeur moyenne d'éléments de la photo au channel blue
%     y        : Le différence entre value_R et value_G
%     th       : le seuil pour définir Yellow
%
% Valeurs de color:
%     0 : Yellow, ssi (value_R>value_B) && (value_G>value_B) && (y<th)&&(value_R>20) ✓
&& (value_G>20)
%     1 : Red, ssi (value_R>value_G) && (value_R>value_B) && (y>th) && (value_R>20)
%     2 : Green, ssi (value_G>value_R) && (value_G>value_B) && (y>th) && ✓
(value_G>20)
%     3 : Blue, ssi (value_B>value_R) && (value_B>value_G) && (y>th) && (value_B>20)

img = snapshot(cam); % Prendre un photo

color=666; % L'initialisation de color
n=2500; % On compte les 50*50 pixels centraux dans les photos

figure
subplot(2,2,1)
imshow(img); % Afficher la photo origine
title('image')

subplot(2,2,2)
R=img(:,:,1);
imshow(R);% Afficher la photo dans le channel rouge
value_R=sum(sum(img(220:269,300:349,1)))/n;
title('r')

subplot(2,2,3)
G=img(:,:,2);
value_G=sum(sum(img(220:269,300:349,2)))/n;
imshow(G); % Afficher la photo dans le channel vert
title('g')

subplot(2,2,4)
B=img(:,:,3);
value_B=sum(sum(img(220:269,300:349,3)))/n;
imshow(B); % Afficher la photo dans le channel bleu
title('b')

```

```
y=abs(value_R-value_G);
th=60; % le seuil pour definir Yellow

if (value_R>value_G) && (value_R>value_B) && (y>th) && (value_R>20)
    color=1; % Red

    elseif (value_G>value_R) && (value_G>value_B) && (y>th) && (value_G>20)
    color=2; % Green

    elseif (value_B>value_R) && (value_B>value_G) && (y>th) && (value_B>20)
    color=3; % Blue

    elseif (value_R>value_B) && (value_G>value_B) && (y<th) && (value_R>20) &&
(value_G>20)
    color=0; % Yellow
end
end
```