

emeri la zabal zazu



Universidad del País Vasco  
Euskal Herriko Unibertsitatea

BILBOKO INGENIARITZA ESKOLA  
ESCUELA DE INGENIERÍA DE BILBAO

INDUSTRIA INGENIARITZA TEKNIKOKO ATALA

SECCIÓN INGENIERÍA TÉCNICA INDUSTRIAL

--

Sinadura DATA	Sinadura DATA
------------------	------------------

# AURKIBIDEA

1. Programazioa.....	2
1.1. MATLAB programazioa.....	2
1.1.1. Proiektua.....	2
1.1.2. Aukeraketa.....	3
1.1.3. Erabilera.....	4
1.1.4. Automatikoa .....	6
1.1.5. Eskuzkoa.....	22
1.1.6. Lehentasunak .....	38

# 1. Programazioa

## 1.1. MATLAB programazioa

### 1.1.1. Proiektua

```
+ function varargout = Proiektua(varargin) ...
% Matlabek sortzen dau, EZ EDITATU %

% Proiektua irekitzean egiten dena adierazten da %
- function Proiektua_OpeningFcn(hObject, eventdata, handles, varargin)
% Choose default command line output for Proiektua
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);
set(gcf, 'currentaxes', handles.fondo);
a=imread('piezak.jpg'); %Fondo bezala paisajea jarriko dugu
image(a);
axis off; %ardatzak kenduko dira

set(gcf, 'currentaxes', handles.uni);
a=imread('UNI.png'); %Uniko logoa jarriko dugu
image(a);
axis off; %ardatzak kenduko dira

% Matlabek sortzen dau, EZ EDITATU %
+ function varargout = Proiektua_OutputFcn(hObject, eventdata, handles) ...

% Hasi botoia sakatzean ejekutatzen da %
- function hasi_Callback(hObject, eventdata, handles)
Aukeraketa;
close Proiektua;
```

Programazio zati honetan aplikazioari hasiera emango zaio. Hasi botoia sakatzerakoan programa hasi egingo da eta hurrengo pantailara egingo da salto.

## 1.1.2. Aukeraketa

```

+ function varargout = Aukeraketa(varargin) [...]
    % Matlabek sortzen dau, EZ EDITATU %

    % Aukeraketa irekitzean egiten dena adierazten da %
- function Aukeraketa_OpeningFcn(hObject, eventdata, handles, varargin)

    % Choose default command line output for Aukeraketa
    handles.output = hObject;

    % Update handles structure
    guidata(hObject, handles);
    set(gcf,'currentaxes',handles.fondo);
    a=imread('piezak.jpg'); %Fondo bezala paisajea jarriko dugu
    image(a);
    axis off; %ardatzak kenduko dira

    set(gcf,'currentaxes',handles.uni);
    a=imread('UNI.png'); %Uniko logoa jarriko dugu
    image(a);
    axis off; %ardatzak kenduko dira

    % Matlabek sortzen dau, EZ EDITATU %
+ function varargout = Aukeraketa_OutputFcn(hObject, eventdata, handles) [...]

    % Formetan botoia sakatzean ejekutatzen da %
- function formetan_Callback(hObject, eventdata, handles)

    forma=1;
    kolore=0;
    save ('forma.mat','forma');
    save ('kolore.mat','kolore');
    Erabilera;
    close Aukeraketa;

    % Koloreetan botoia sakatzean ejekutatzen da %
- function koloretan_Callback(hObject, eventdata, handles)

    forma=0;
    kolore=1;
    save ('forma.mat','forma');
    save ('kolore.mat','kolore');
    Erabilera;
    close Aukeraketa;

    % Amaitu botoia sakatzean ejekutatzen da %
- function amaitu_Callback(hObject, eventdata, handles)

    close all; %Dena zarratuko dugu
    clear forma;
    clear kolore;

```

Aukeraketa izeneko honetan zein modutan egin nahi den sailkapena aukeratuko da. Formen araberako sailkapena egiten denean *forma* aldagaia batean jarriko da. Koloreen

araberako sailkapena adierazten baldin bada *kolore* aldagaia batean jarriko da. Amaitu botoia sakatzerakoan ordea, aplikazioa itxi egingo da.

### 1.1.3. Erabilera

```
+ function varargout = Erabilera(varargin) ...
    % Matlabek sortzen dau, EZ EDITATU %

    % Erabilera irekitzean egiten dena adierazten da %
- function Erabilera_OpeningFcn(hObject, eventdata, handles, varargin)
    % Choose default command line output for Erabilera
    handles.output = hObject;

    % Update handles structure
    guidata(hObject, handles);
    set(gcf, 'currentaxes', handles.fondo);
    a=imread('piezak.jpg'); %Fondo bezala paisajea jarriko dugu
    image(a);
    axis off; %ardatzak kenduko dira

    set(gcf, 'currentaxes', handles.uni);
    a=imread('UNI.png'); %Uniko logoa jarriko dugu
    image(a);
    axis off; %ardatzak kenduko dira

    % Matlabek sortzen dau, EZ EDITATU %
+ function varargout = Erabilera_OutputFcn(hObject, eventdata, handles) ...

    % Automatikoa botoia sakatzean ejekutatzen da %
- function automatikoa_Callback(hObject, eventdata, handles)

    Automatikoa;
    close Erabilera;

    % Manuala botoia sakatzean ejekutatzen da %
- function manuala_Callback(hObject, eventdata, handles)

    Manuala;
    close Erabilera;

    % Atzera botoia sakatzean ejekutatzen da %
- function atzera_Callback(hObject, eventdata, handles)

    close all; %Dena zarratuko dugu
    clear forma;
    clear kolore;
    Aukeraketa;
```

Erabilera programazioan automatikoan lan egitea edo eskuzkoan lan egitea aukeratuko da. Automatikoa botoia sakatzen denean automatikoa pantailara egingo da salto. Eskuzkoa sakatzen bada aldiz, eskuzko pantailara egingo da salto.

## 1.1.4. Automatikoa

```

function varargout = Automatikoa(varargin) ...
% Matlabek sortzen dau, EZ EDITATU %

% Automatikoa irekitzean egiten dena adierazten da %
function Automatikoa_OpeningFcn(hObject, eventdata, handles, varargin)
global vid
global a

% Choose default command line output for Manual
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);
set(gcf, 'currentaxes', handles.Bideo);
axis off; %Ardatzak kenduko dira
set(gcf, 'currentaxes', handles.Irudi);
axis off; %Ardatzak kenduko dira
set(gcf, 'currentaxes', handles.Irudi);
axis off; %Ardatzak kenduko dira
set(gcf, 'currentaxes', handles.Detektatu);
axis off; %Ardatzak kenduko dira
set(gcf, 'currentaxes', handles.fondo);
b=imread('piezak.jpg'); %Fondo bezala paisajea jarriko dugu
image(b);

load ('forma');
load ('kolore');
if(forma==1)
    set(handles.pGorri, 'Visible', 'off');
    set(handles.pUrdin, 'Visible', 'off');
    set(handles.pBerde, 'Visible', 'off');
    set(handles.aukeratu, 'Visible', 'off');
    set(handles.textKolore, 'Visible', 'off');
end
if(kolore==1)
    set(handles.pZir, 'Visible', 'off');
    set(handles.pTri, 'Visible', 'off');
    set(handles.pKar, 'Visible', 'off');
    set(handles.aukeraketa, 'Visible', 'off');
    set(handles.textForma, 'Visible', 'off');
end

a=arduino('COM5'); %Arduinoa aktibatuko da

a.configureDigitalPin(2, 'output'); %pos 1 digital sarrera
a.configureDigitalPin(3, 'output'); %pos 2 digital sarrera
a.configureDigitalPin(4, 'output'); %pos 3 digital irteera
a.configureDigitalPin(5, 'output'); %pos 4 digital irteera
a.configureDigitalPin(6, 'output'); %pieza=2 eta Gorria digital irteera
a.configureDigitalPin(7, 'output'); %pieza=3 eta Berdea digital irteera
a.configureDigitalPin(8, 'output'); %pieza=4 eta Urdina digital irteera
a.configureDigitalPin(9, 'input'); %pos 1 digital sarrera
a.configureDigitalPin(10, 'input'); %pos 2 digital sarrera
a.configureDigitalPin(11, 'input'); %pos 3 digital sarrera
a.configureDigitalPin(12, 'input'); %pos 4 digital sarrera

%%% Irteerako PIN denak 1-ean jarriko dira Robotak 0 bezal identifikatzen
a.writeDigitalPin(2,1);
a.writeDigitalPin(3,1);
a.writeDigitalPin(4,1);
a.writeDigitalPin(5,1);
a.writeDigitalPin(6,1);
a.writeDigitalPin(7,1);
a.writeDigitalPin(8,1);

%%% Bideoa momentuan bertan ikusten ari dena agertzeko %%%
vid=videoinput('winvideo',1,'YUY2_640x480'); %Bideorako kanala irekitzen da
erresoluzioa = get(vid, 'VideoResolution'); %Erresoluzioa neurtzen da
altuera = erresoluzioa(1); %Erresoluzioaren altueraren zatia
zabalera = erresoluzioa(2); %Erresoluzioaren zabalera zatia
ZenbBanda = get(vid, 'NumberOfBands'); %Banda kopurua
hImage = image( zeros(zabalera, altuera, ZenbBanda), 'Parent', handles.Bideo);
preview(vid,hImage) %Ikusten dena agertzen da GUI-an

```

Automatikoa irekitzen denean ARDUINO txartela konektatzen da eta bere pinen konfigurazioa egin. Ondoren, kamera konektatu eta interfazean bistaratuko da.

Ondoren erabiltzaileak interfazean daukan menuarekin hartu nahi dituen pieza kopurua aukeratuko du eta formen edo koloreen aukeraketaren arabera programazioa desberdina da:

```
% Matlabek sortzen dau, EZ EDITATU %
+ function varargout = Automatikoa_OutputFcn(hObject, eventdata, handles) ...

% Menua izeneko menua sakatzean ejekutatzen da %
- function menua_Callback(hObject, eventdata, handles)

    global vid
    global a

    global Irudia
    global Pieza_kop
    global L
    global Pieza

    global Zirkulu_Pieza
    global Triangelu_Pieza
    global Karratu_Pieza
    global Zirkulu_Pieza2
    global Triangelu_Pieza2
    global Karratu_Pieza2
    global Posizio_Pieza

    global Zenb_Gorri
    global Zenb_Urdin
    global Zenb_Berde
    global Zirkulu_Gorri
    global Karratu_Gorri
    global Triangelu_Gorri
    global Zirkulu_Urdin

    global Karratu_Urdin
    global Triangelu_Urdin
    global Zirkulu_Berde
    global Karratu_Berde
    global Triangelu_Berde
    global Posizio_Gorri
    global Posizio_Urdin
    global Posizio_Berde

    Zirkulu_Gorri2=0;
    Karratu_Gorri2=0;
    Triangelu_Gorri2=0;
    Zirkulu_Urdin2=0;
    Karratu_Urdin2=0;
    Triangelu_Urdin2=0;
    Zirkulu_Berde2=0;
    Karratu_Berde2=0;
    Triangelu_Berde2=0;

    balioa=get(handles.menua,'value'); %Menuan aukeratutakoa lortzen da
    load ('forma');
    load ('kolore');

    if(forma==1)
        l=1;
        x=1;
        %%% Lortutakoaren arabera zer egin jakiteko %%%
        switch balioa
```



```

##### 1.aukera #####
case 2
    Pieza=2;
    a.writeDigitalPin(6,0);
    pause(70)
    a.writeDigitalPin(6,1);

    Irudia=getsnapshot(vid);    %Argazkia egingo da
    Irudia=ycbcr2rgb(Irudia);    %Argazkia RGB ingurura igaroko da
    axes(handles.Irudi);        %Argazkia erakutsiko da
    imshow(Irudia);

    Irudi_gris=rgb2gray(Irudia);    %Argazkia grisetara igaroko da
    [m,n]=size(Irudi_gris);        %Argazkiaren neurria lortuko da
    Irudi_binario=ones(m,n);        %Irudi binarioaren fondoa guztian 1-ak jarriko dira
    umbral=100;                    %Argazkia binarizatzeko Umbrala aukeratuko da

    %%% Irudia BINARIZATUKO da %%%
    for i = 1:m
        for j = 1:n
            if (Irudi_gris(i,j)>umbral)
                Irudi_binario(i,j)=0;
            else
                Irudi_binario(i,j)=1;
            end
        end
    end
end

Irudi_binario=medfilt2(Irudi_binario,[15,15]);
Irudi_binario=bwareaopen(Irudi_binario,900);    %100 baino txikiagoko pixelak borratuko dira
[L, Pieza_kop]=bwlabel(Irudi_binario,8);        %Objetuak lortuko dira baita kopurua ere
set(handles.Pieza,'String',Pieza_kop);        %Pieza kopurua GUI-an ageriko da

axes(handles.Detektatu);
imshow(Irudia);    %Irudia ageriko da detektatutako pieza guztiekin
hold all;

if (Pieza_kop>0)
    for i=1:Pieza_kop

        Stats_Pieza=regionprops(L,'Area','Perimeter','Centroid','BoundingBox');

        Area_Pieza=[Stats_Pieza.Area];                %Area balioa gordeko da
        Perimetro_Pieza=[Stats_Pieza.Perimeter];      %Perimetro balioa gordeko da
        Zentro_Pieza=[Stats_Pieza.Centroid];

        K_Pieza(1)=(Perimetro_Pieza(1)^2)/Area_Pieza(1);    %Konstantea lortuko da

        l=l+1;
    end
    for ObjectPieza=1:length(Stats_Pieza)
        Kaja_Pieza=Stats_Pieza(ObjectPieza).BoundingBox;    %For bat sortuko da errektanguloak sortzeko
                                                              %Kaxaren balioa

```

```

        rectangle('Position',Kaja_Pieza,'EdgeColor','b','LineWidth',2) %Kaxak margotuko dira
        plot(Zentro_Pieza(1),Zentro_Pieza(2),'-g+') %Errektanguloak eta zentroak irudikatuko dira
    end

    %%% Lortutako konstantearen arabera forma lortuko da %%%
    for x=1:Pieza_kop

        Zirkulu_Pieza(x)=0;
        Karratu_Pieza(x)=0;
        Triangelu_Pieza(x)=0;

        if ((10<K_Pieza(x)) && (K_Pieza(x)<13.3)) %Zirkulua izango da

            Zirkulu_Pieza(x)=1;
            Karratu_Pieza(x)=0;
            Triangelu_Pieza(x)=0;

        end

        if ((13.3<K_Pieza(x)) && (K_Pieza(x)<17)) %Karratua izango da

            Zirkulu_Pieza(x)=0;
            Karratu_Pieza(x)=1;
            Triangelu_Pieza(x)=0;

        end

        if ((17<K_Pieza(x)) && (K_Pieza(x)<22)) %Triangelua izango da

            Zirkulu_Pieza(x)=0;
            Karratu_Pieza(x)=0;

            Triangelu_Pieza(x)=1;

        end

    end

    %%% Piezen posizioa lortuko da %%%
    if (Zentro_Pieza>0)

        k=1;

        for i=1:2:Pieza_kop*2

            if (Zentro_Pieza(i)<320) && (Zentro_Pieza(i+1)<240) % 1.posizioan dagoen ikusteko

                Posizio_Pieza(k)=1;
            end

            if (Zentro_Pieza(i)>320) && (Zentro_Pieza(i+1)<240) % 2.posizioan dagoen ikusteko

                Posizio_Pieza(k)=2;
            end

            if (Zentro_Pieza(i)<320) && (Zentro_Pieza(i+1)>240) % 3.posizioan dagoen ikusteko

                Posizio_Pieza(k)=3;
            end

            if (Zentro_Pieza(i)>320) && (Zentro_Pieza(i+1)>240) % 4.posizioan dagoen ikusteko

                Posizio_Pieza(k)=4;
            end

            k=k+1;

        end

    end
end
end

```

```

Zirkulu_Pieza2=sum(Zirkulu_Pieza);
set(handles.Zir,'String',Zirkulu_Pieza2);    %Zirkulu kopurua agertuko da GUI-an
Triangelu_Pieza2=sum(Triangelu_Pieza);
set(handles.Tri,'String',Triangelu_Pieza2);    %Zirkulu kopurua agertuko da GUI-an
Karratu_Pieza2=sum(Karratu_Pieza);
set(handles.Kar,'String',Karratu_Pieza2);    %Karratu kopurua agertuko da GUI-an

Pieza_Oker=0;

if(Pieza_kop~=(Zirkulu_Pieza2+Triangelu_Pieza2+Karratu_Pieza2))

    Pieza_Oker=(Pieza_kop)-(Zirkulu_Pieza2+Triangelu_Pieza2+Karratu_Pieza2);

end

set(handles.PiezaOker,'String',Pieza_Oker);
Pieza=Pieza-Pieza_Oker;

end

```

Aurreko aukeraketa pantailan formen araberako sailkapena eskatu baldin bada honaino egingako programazioa sortzen da. Bertan argazkia egingo da eta irudia binarizatu ondoren piezak detektatuko dira. Behin piezak detektatuta daudela beraien ezaugarriak lortuko dira ondoren forma eta posizioa zein den jakiteko. Azkenik emaitza horiek interfazean bistaratuko dira. Programazio zati hau errepikatu egiten da aukeratutako pieza kopuruen arabera, beraz, ez da berriro jarriko.

Orain koloreen sailkapena eskatu bada:

```

if(kolore==1)

    %%% Lortutakoaren arabera zer egin jakiteko %%%
    switch balioa

        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        case 2

            Pieza=2;
            a.writeDigitalPin(6,0);
            pause(70)
            a.writeDigitalPin(6,1)

            Irudia=getsnapshot(vid);    %Argazkia egingo da
            Irudia=ycbcr2rgb(Irudia);    %Argazkia RGB ingurura igaroko da
            axes(handles.Irudi);    %Argazkia erakutsiko da
            imshow(Irudia);

            Irudi_gris=rgb2gray(Irudia);    %Argazkia grisetara igaroko da
            [m,n]=size(Irudi_gris);    %Argazkiaren neurria lortuko da
            Irudi_binario=ones(m,n);    %Irudi binarioaren fonda guztian 1-ak jarriko dira
            umbral=100;    %Argazkia binarizatzeko Umbrala aukeratuko da

            %%% Irudia BINARIZATUKO da %%%
            for i = 1:m
                for j = 1:n
                    if (Irudi_gris(i,j)>umbral)
                        Irudi_binario(i,j)=0;
                    end
                end
            end
        end
    end
end

```

```

        else
            Irudi_binario(i,j)=1;
        end
    end
end

Irudi_binario=bwareaopen(Irudi_binario,900);    %100 baino txikiagoko pixelak borratuko dira
[L, Pieza_kop]=bwlabel(Irudi_binario,8);      %Objetuak lortuko dira baita kopurua ere
set(handles.Pieza, 'String', Pieza_kop);      %Pieza kopurua GUI-an ageriko da

if Pieza_kop>0

    gorri=imsubtract(Irudia(:,:,1),rgb2gray(Irudia)); %Irudian pixel gorriak isolatuko dira
    gorri=medfilt2(gorri,[15,15]);              % [15,15]-eko iragazki bat erabiliko da
    gorri=im2bw(gorri,0.26);                   %Irudia Grisetatik binarioa igaroko da
    gorri=bwareaopen(gorri,300);              %300 baino gutxiagoko pixelak borratuko dira

    berde=imsubtract(Irudia(:,:,2),rgb2gray(Irudia)); %Irudian pixel berdeak isolatuko dira
    berde=medfilt2(berde,[15,15]);            % [15,15]-eko iragazki bat erabiliko da
    berde=im2bw(berde,0.03);                 %Irudia Grisetatik binarioa igaroko da
    berde=bwareaopen(berde,5000);           %5000 baino gutxiagoko pixelak borratuko dira

    urdin=imsubtract(Irudia(:,:,3),rgb2gray(Irudia)); %Irudian pixel urdinak isolatuko dira
    urdin=medfilt2(urdin,[9,9]);              % [9,9]-eko iragazki bat erabiliko da
    urdin=im2bw(urdin,0.18);                 %Irudia Grisetatik binarioa igaroko da
    urdin=bwareaopen(urdin,500);             %500 baino gutxiagoko pixelak borratuko dira

    [LGorri, Zenb_Gorri]=bwlabel(gorri,8);    %Objetua gorriak lortuko dira baita kopurua ere
    [LBerde, Zenb_Berde]=bwlabel(berde,8);    %Objetua berdeak lortuko dira baita kopurua ere

    [LUrdin, Zenb_Urdin]=bwlabel(urdin,8);    %Objetua urdinak lortuko dira baita kopurua ere

    set(handles.ZenbGorri, 'String', Zenb_Gorri); %Zenbat pieza gorri dauden ageriko da GUI-an
    set(handles.ZenbBerde, 'String', Zenb_Berde); %Zenbat pieza berde dauden ageriko da GUI-an
    set(handles.ZenbUrdin, 'String', Zenb_Urdin); %Zenbat pieza urdin dauden ageriko da GUI-an
    Zentro_Gorri=[];
    Zentro_Berde=[];
    Zentro_Urdin=[];

    axes(handles.Detektatu);
    imshow(Irudia); %Irudia ageriko da detektatutako pieza guztiekin
    hold all;

    %%% Pieza GORRIAK aztertuko dira %%%
    if Zenb_Gorri>0

        Stats_Gorri=regionprops(LGorri, 'BoundingBox', 'Centroid'); %Pieza Gorrien ezaugarriak lortuko dira

        for ObjectGorri=1:length(Stats_Gorri)
            Kaja_Gorri=Stats_Gorri(ObjectGorri).BoundingBox; %For bat sortuko da errektanguloak sortzeko
            Zentro_Gorri=Stats_Gorri(ObjectGorri).Centroid; %Kaxaren balioa
                                                    %Zentroidearen balioa

            rectangle('Position',Kaja_Gorri,'EdgeColor','b','LineWidth',2) %Kaxak margotuko dira
            plot(Zentro_Gorri(1),Zentro_Gorri(2),'-g+') %Errektanguloak eta zentroak irudikatuko dira
        end

        Zentro_Gorri=[Stats_Gorri.Centroid]; %Zentroaren balioa gordeko da array bate
    end
end

```

```

end

%%% Pieza BERDEAK aztertuko dira %%%
if Zenb_Berde>0

    Stats_Berde=regionprops(LBerde, 'BoundingBox', 'Centroid'); %Pieza Berdeen ezaugarriak lortuko dira

    for ObjectBerde=1:length(Stats_Berde) %For bat sortuko da errektanguloak sortzeko
        Kaja_Berde=Stats_Berde(ObjectBerde).BoundingBox; %Kaxaren balioa
        Zentro_Berde=Stats_Berde(ObjectBerde).Centroid; %Zentroidearen balioa

        rectangle('Position',Kaja_Berde,'EdgeColor','b','LineWidth',2) %Kaxak margotuko dira
        plot(Zentro_Berde(1),Zentro_Berde(2),'-g+') %Errektanguloak eta zentroak irudikatuko dira
    end

    Zentro_Berde=[Stats_Berde.Centroid]; %Zentroaren balioa gordeko da array baten

end

%%% Pieza URDINAK aztertuko dira %%%
if Zenb_Urdin>0

    Stats_Urdin=regionprops(LUrdin, 'BoundingBox', 'Centroid'); %Pieza Urdinen ezaugarriak lortuko dira

    for ObjectUrdin=1:length(Stats_Urdin) %For bat sortuko da errektanguloak sortzeko
        Kaja_Urdin=Stats_Urdin(ObjectUrdin).BoundingBox; %Kaxaren balioa
        Zentro_Urdin=Stats_Urdin(ObjectUrdin).Centroid; %Zentroidearen balioa

        rectangle('Position',Kaja_Urdin,'EdgeColor','r','LineWidth',2) %Kaxak margotuko diras
        plot(Zentro_Urdin(1),Zentro_Urdin(2),'-g+') %Errektanguloak eta zentroak irudikatuko dira
    end

    Zentro_Urdin=[Stats_Urdin.Centroid]; %Zentroaren balioa gordeko da array baten

end

%%% Pieza GORRIEN posizioa lortuko da %%%
if Zentro_Gorri>0

    j=1;

    for i=1:2:Zenb_Gorri*2

        if ((Zentro_Gorri(i)<320)&&(Zentro_Gorri(i+1)<240)) % 1.posizioan dagoen ikusteko
            Posizio_Gorri(j)=1;
        end

        if (Zentro_Gorri(i)>320)&&(Zentro_Gorri(i+1)<240) % 2.posizioan dagoen ikusteko
            Posizio_Gorri(j)=2;
        end

        if (Zentro_Gorri(i)<320)&&(Zentro_Gorri(i+1)>240) % 3.posizioan dagoen ikusteko

```

```

        Posizio_Gorri(j)=3;
    end

    if (Zentro_Gorri(i)>320) && (Zentro_Gorri(i+1)>240) % 4.posizioan dagoen ikusteko
        Posizio_Gorri(j)=4;
    end

    j=j+1;

end

else
    Posizio_Gorri=[];
end

%%% Pieza BERDEEN posizioa lortuko da %%%
if Zentro_Berde>0

    j=1;

    for i=1:2:Zenb_Berde*2

        if (Zentro_Berde(i)<320) && (Zentro_Berde(i+1)<240) % 1.posizioan dagoen ikusteko
            Posizio_Berde(j)=1;
        end

        if (Zentro_Berde(i)>320) && (Zentro_Berde(i+1)<240) % 2.posizioan dagoen ikusteko
            Posizio_Berde(j)=2;
        end

        if (Zentro_Berde(i)<320) && (Zentro_Berde(i+1)>240) % 3.posizioan dagoen ikusteko
            Posizio_Berde(j)=3;
        end

        if (Zentro_Berde(i)>320) && (Zentro_Berde(i+1)>240) % 4.posizioan dagoen ikusteko
            Posizio_Berde(j)=4;
        end

        j=j+1;

    end

else
    Posizio_Berde=[];
end

%%% Pieza URDINEN posizioa lortuko da %%%
if Zentro_Urdin>0

    j=1;

    for i=1:2:Zenb_Urdin*2

```

```

        if (Zentro_Urdin(i)<320) &&(Zentro_Urdin(i+1)<240)    % 1.posizioan dagoen ikusteko
            Posizio_Urdin(j)=1;
        end
        if (Zentro_Urdin(i)>320) &&(Zentro_Urdin(i+1)<240)    % 2.posizioan dagoen ikusteko
            Posizio_Urdin(j)=2;
        end
        if (Zentro_Urdin(i)<320) &&(Zentro_Urdin(i+1)>240)    % 3.posizioan dagoen ikusteko
            Posizio_Urdin(j)=3;
        end
        if (Zentro_Urdin(i)>320) &&(Zentro_Urdin(i+1)>240)    % 4.posizioan dagoen ikusteko
            Posizio_Urdin(j)=4;
        end
        j=j+1;
    end
else
    Posizio_Urdin=[];
end

```

```

%%% Pieza GORRIEN forma lortuko da %%%
if Zenb_Gorri>0
    j=1;
    g=1;
    Area_Gorri=[];
    Perimetro_Gorri=[];
    K_Gorri=[];
    for i=1:Zenb_Gorri
        Stats_Gorri=regionprops(LGorri,'Area','Perimeter');%Pieza gorrien area eta perimetroa lortuko da
        Area_Gorri=[Stats_Gorri.Area];                    %Area balioa gordeko da
        Perimetro_Gorri=[Stats_Gorri.Perimeter];          %Perimetro balioa gordeko da
        K_Gorri(j)=(Perimetro_Gorri(j)^2)/Area_Gorri(j);  %Konstantea lortuko da
        j=j+1;
    end
    %%% Lortutako konstantearen arabera forma lortuko da %%%
    for g=1:Zenb_Gorri
        Zirkulu_Gorri(g)=0;
        Karratu_Gorri(g)=0;
        Triangelu_Gorri(g)=0;
    end
end

```

```

        if ((10<K_Gorri(g)) && (K_Gorri(g)<14))    %Zirkulua izango da

            Zirkulu_Gorri(g)=1;
            Karratu_Gorri(g)=0;
            Triangelu_Gorri(g)=0;

        end

        if ((16<K_Gorri(g)) && (K_Gorri(g)<18))    %Karratua izango da

            Zirkulu_Gorri(g)=0;
            Karratu_Gorri(g)=1;
            Triangelu_Gorri(g)=0;

        end

        if ((18<K_Gorri(g)) && (K_Gorri(g)<28))    %Triangelua izango da

            Zirkulu_Gorri(g)=0;
            Karratu_Gorri(g)=0;
            Triangelu_Gorri(g)=1;

        end

    end

    Zirkulu_Gorri2=sum(Zirkulu_Gorri);
    set(handles.ZirGorri,'String',Zirkulu_Gorri2);
    Karratu_Gorri2=sum(Karratu_Gorri);
    set(handles.KarGorri,'String',Karratu_Gorri2);
    Triangelu_Gorri2=sum(Triangelu_Gorri);
    set(handles.TriGorri,'String',Triangelu_Gorri2);

```

```

%%% Pieza BERDEEN forma lortuko da %%%
if Zenb_Berde>0

    k=1;
    b=1;
    Area_Berde=[];
    Perimetro_Berde=[];
    K_Berde=[];

    for i=1:Zenb_Berde

        Stats_Berde=regionprops(LBerde,'Area','Perimeter');%Pieza berdeen area eta perimetroa lortuko da
        Area_Berde=[Stats_Berde.Area];                    %Area balioa gordeko da
        Perimetro_Berde=[Stats_Berde.Perimeter];          %Perimetro balioa gordeko da

        K_Berde(k)=(Perimetro_Berde(k)^2)/Area_Berde(k);  %Konstantea lortuko da

        k=k+1;
    end

%%% Lortutako konstantearen arabera forma lortuko da %%%
for b=1:Zenb_Berde

    Zirkulu_Berde(b)=0;
    Karratu_Berde(b)=0;
    Triangelu_Berde(b)=0;

```



```

        if ((10<K_Berde(b)) && (K_Berde(b)<13.7)) %Zirkulua izango da

            Zirkulu_Berde(b)=1;
            Karratu_Berde(b)=0;
            Triangelu_Berde(b)=0;

        end

        if ((13.7<K_Berde(b)) && (K_Berde(b)<20)) %Karratua izango da

            Zirkulu_Berde(b)=0;
            Karratu_Berde(b)=1;
            Triangelu_Berde(b)=0;

        end

        if ((20<K_Berde(b)) && (K_Berde(b)<28)) %Triangelua izango da

            Zirkulu_Berde(b)=0;
            Karratu_Berde(b)=0;
            Triangelu_Berde(b)=1;

        end

    end

    Zirkulu_Berde2=sum(Zirkulu_Berde);
    set(handles.ZirBerde,'String',Zirkulu_Berde2);
    Karratu_Berde2=sum(Karratu_Berde);
    set(handles.KarBerde,'String',Karratu_Berde2);
    Triangelu_Berde2=sum(Triangelu_Berde);
    set(handles.TriBerde,'String',Triangelu_Berde2);

```

```

%%% Pieza BERDEEN forma lortuko da %%%
if Zenb_Urdin>0

    l=1;
    c=1;
    Area_Urdin=[];
    Perimetro_Urdin=[];
    K_Urdin=[];

    for i=1:Zenb_Urdin

        Stats_Urdin=regionprops(LUrdin,'Area','Perimeter');%Pieza berdeen area eta perimetroa lortuko da
        Area_Urdin=[Stats_Urdin.Area]; %Area balioa gordeko da
        Perimetro_Urdin=[Stats_Urdin.Perimeter]; %Perimetro balioa gordeko da

        K_Urdin(l)=(Perimetro_Urdin(l)^2)/Area_Urdin(l); %Konstantea lortuko da

        l=l+1;
    end

    %%% Lortutako konstantearen arabera forma lortuko da %%%
    for c=1:Zenb_Urdin

        Zirkulu_Urdin(c)=0;
        Karratu_Urdin(c)=0;
        Triangelu_Urdin(c)=0;
    end

```

```

        if ((10<K_Urdin(c)) &&(K_Urdin(c)<14))    %Zirkulua izango da

            Zirkulu_Urdin(c)=1;
            Karratu_Urdin(c)=0;
            Triangelu_Urdin(c)=0;

        end
        if ((14<K_Urdin(c)) &&(K_Urdin(c)<19))    %Karratua izango da

            Zirkulu_Urdin(c)=0;
            Karratu_Urdin(c)=1;
            Triangelu_Urdin(c)=0;

        end
        if ((19<K_Urdin(c)) &&(K_Urdin(c)<28))    %Triangelua izango da

            Zirkulu_Urdin(c)=0;
            Karratu_Urdin(c)=0;
            Triangelu_Urdin(c)=1;

        end
    end

    Zirkulu_Urdin2=sum(Zirkulu_Urdin);
    set(handles.ZirUrdin, 'String', Zirkulu_Urdin2);
    Karratu_Urdin2=sum(Karratu_Urdin);
    set(handles.KarUrdin, 'String', Karratu_Urdin2);
    Triangelu_Urdin2=sum(Triangelu_Urdin);
    set(handles.TriUrdin, 'String', Triangelu_Urdin2);
    %Zirkulu urdin guztiak gordeko dira
    %Zirkulu kopurua agertuko da GUI-an
    %Karratu urdin guztiak gordeko dira
    %Karratu kopurua agertuko da GUI-an
    %Triangelu urdin guztiak gordeko dira
    %Triangelu kopurua agertuko da GUI-an

```

```

Pieza_Oker=0;

if(Pieza_kop~=(Zirkulu_Gorri2+Triangelu_Gorri2+Karratu_Gorri2+Zirkulu_Berde2+Triangelu_Berde2+Karratu_Berde2

    Pieza_Oker=(Pieza_kop-(Zirkulu_Gorri2+Triangelu_Gorri2+Karratu_Gorri2+Zirkulu_Berde2+Triangelu_Berde2+Ka

end
if(Pieza_kop~=(Zenb_Gorri+Zenb_Urdin+Zenb_Berde))

    Pieza_Oker=(Pieza_kop)-(Zenb_Gorri+Zenb_Urdin+Zenb_Berde);

end
if(Pieza_kop~=(Zirkulu_Gorri2+Triangelu_Gorri2+Karratu_Gorri2+Zirkulu_Berde2+Triangelu_Berde2+Karratu_Berde2

    Pieza_Oker=(Pieza_kop)-(Zirkulu_Gorri2+Triangelu_Gorri2+Karratu_Gorri2+Zirkulu_Berde2+Triangelu_Berde2+Ka

end

set(handles.PiezaOker, 'String', Pieza_Oker);
Pieza=Pieza-Pieza_Oker;

end

```

Koloreen bidezko sailkapena aukeratzen denean irudia ateratzen da lehenengo. Ondoren, piezak detektatzen dira koloreen arabera. Orduan detektatutako objektuak koloreen arabekoak izango dira. Ondoren objektu bakoitzari ezaugarriak ateratzen zaizkio. Ezaugarri horien arabera objektuen forma eta posizioa lortzen da eta azkenik emaitza horiek bistaratzen dira. Programazio zati hau aukeratutako pieza kopuru guztiakin errepikatzen da beraz, ez da berriro jarriko.

Behin objektuen forma eta posizioa jakinda, erabiltzaileak zer sailkatu nahi duen aukeratu du. Aurretik aukeraketa formetan izan bada:

```
% Formetan sailkatu menua sakatzean ejekutatzeko da %
function aukeraketa_Callback(hObject, eventdata, handles)

global Zirkulu_Pieza
global Triangelu_Pieza
global Karratu_Pieza
global Zirkulu_Pieza2
global Triangelu_Pieza2
global Karratu_Pieza2
global Posizio_Pieza
global pos_bidali
global pos_itzuli

global a
global Pieza

balioa2=get(handles.aukeraketa,'value'); %Menuan aukeraturakoa lortzen da

%%% Lortutakoaren arabera zer egin jakiteko %%%
switch balioa2
    case 2 %%%Forma denak hartuko ditu%%
        kasua=1;
        if (Zirkulu_Pieza2>0)&&(Triangelu_Pieza2>0)&&(Karratu_Pieza2>0)
            [ pos_bidali , pos_itzuli ] = Lehenetasunak2( a , Posizio_Pieza , kasua , Zirkulu_Pieza2 , Triangelu_Pieza2 ,
                clear all;
                close Automatikoa;
                Erabilera;
            else
                h1 = errordlg('Hiru formak ezin dira hartu!!!');
            end
        case 3 %%%Zirkuluak hartuko ditu%%
            kasua=2;
            if (Zirkulu_Pieza2>0)
                [ pos_bidali , pos_itzuli ] = Lehenetasunak2( a , Posizio_Pieza , kasua , Zirkulu_Pieza2 , Triangelu_Pieza2 ,
                    clear all;
                    close Automatikoa;
                    Erabilera;
                else
                    h2 = errordlg('Zirkuluak ezin dira hartu!!!');
                end
            case 4 %%%Triangeluak hartuko ditu%%
                kasua=3;
                if(Triangelu_Pieza2>0)
                    [ pos_bidali , pos_itzuli ] = Lehenetasunak2( a , Posizio_Pieza , kasua , Zirkulu_Pieza2 , Triangelu_Pieza2 ,
                        clear all;
                        close Automatikoa;
                        Erabilera;
                    else
                        h3 = errordlg('Triangeluak ezin dira hartu!!!');
                    end
                case 5 %%%Karratuak hartuko ditu%%
                    kasua=4;
                    if(Karratu_Pieza2>0)
                        [ pos_bidali , pos_itzuli ] = Lehenetasunak2( a , Posizio_Pieza , kasua , Zirkulu_Pieza2 , Triangelu_Pieza2 ,
                            clear all;
                            close Automatikoa;
                            Erabilera;
                        else
                    end
                end
            end
        end
    end
end
```

```
        h4 = errordlg('Karratuak ezin dira hartu!!!');
    end
case 6 %%Zirkuluak eta Triangeluak hartuko ditu%%
kasua=5;
if(Zirkulu_Pieza2>0)&&(Triangelu_Pieza2>0)
    [ pos_bidali , pos_itzuli ] = Lehtasunak2( a , Posizio_Pieza , kasua , Zirkulu_Pieza2 , Triangelu_Pieza2 ,
        clear all;
        close Automatiko;
        Erabilera;
    else
        h5 = errordlg('Zirkuluak eta Triangeluak ezin dira hartu!!!');
    end
case 7 %%Zirkuluak eta Karratuak hartuko ditu%%
kasua=6;
if(Zirkulu_Pieza2>0)&&(Karratu_Pieza2>0)
    [ pos_bidali , pos_itzuli ] = Lehtasunak2( a , Posizio_Pieza , kasua , Zirkulu_Pieza2 , Triangelu_Pieza2 ,
        clear all;
        close Automatiko;
        Erabilera;
    else
        h6 = errordlg('Zirkuluak eta Karratuak ezin dira hartu!!!');
    end
case 8 %%Triangeluak eta Karratuak hartuko ditu%%
kasua=7;
if(Triangelu_Pieza2>0)&&(Karratu_Pieza2>0)
    [ pos_bidali , pos_itzuli ] = Lehtasunak2( a , Posizio_Pieza , kasua , Zirkulu_Pieza2 , Triangelu_Pieza2 ,
        clear all;
        close Automatiko;
        Erabilera;
    else
        h7 = errordlg('Triangelu eta Karratuak ezin dira hartu!!!');
    end
end
```

Zati honetan erabiltzaileak sailkatu nahi duena adieraziko da. Non ondoren Lehtasunak2 funtzioarekin aukeratutakoaren arteko lehtasunak emango dira eta robotari bidaliko zaio informazioa.

Koloreen araberako aukeratu bada:

```
% Koloretan sailkatu menua sakatzean ejetutatzen da %
function aukeratu_Callback(hObject, eventdata, handles)

    global a
    global Pieza

    global Zenb_Gorri
    global Zenb_Urdin
    global Zenb_Berde
    global Zirkulu_Gorri
    global Karratu_Gorri
    global Triangelu_Gorri
    global Posizio_Gorri
    global Posizio_Urdin
    global Zirkulu_Urdin
    global Karratu_Urdin
    global Triangelu_Urdin
    global Zirkulu_Berde
    global Karratu_Berde
    global Triangelu_Berde
    global Posizio_Berde

    global pos_bidali
    global pos_itzuli

    balioa=get(handles.aukeratu,'value'); %Menuan aukeratutakoa lortzen da

    %%% Lortutakoaren arabera zer egin jakiteko %%%
    switch balioa

        case 2 %%%Hiru koloreak hartuko ditu%%
            kasua=1;
            if (Zenb_Gorri>0)&&(Zenb_Urdin>0)&&(Zenb_Berde>0)
                [ pos_bidali , pos_itzuli ] = Lehetasunak( Zenb_Gorri , Zenb_Urdin , Zenb_Berde , Zirkulu_Gorri , Triangelu
                    clear all;
                    clear forma;
                    clear kolore;
                    close Automatikoa;
                    Erabilera;
            else
                h1 = errorldg('Hiru Koloreak ezin dira hartu!!!');
            end
        case 3 %%%Gorriak hartuko ditu%%
            kasua=2;
            if(Zenb_Gorri>0)
                [ pos_bidali , pos_itzuli ] = Lehetasunak( Zenb_Gorri , Zenb_Urdin , Zenb_Berde , Zirkulu_Gorri , Triangelu
                    clear all;
                    clear forma;
                    clear kolore;
                    close Automatikoa;
                    Erabilera;
            else
                h2 = errorldg('Gorriak ezin dira hartu!!!');
            end
        case 4 %%%Berdeak hartuko ditu%%
            kasua=3;
```

```
if(Zenb_Berde>0)
    [ pos_bidalari , pos_itzuli ] = Lehentasanak( Zenb_Gorri , Zenb_Urdin , Zenb_Berde , Zirkulu_Gorri , Triangelu
clear all;
clear forma;
clear kolore;
close Automatiko;
Erabilera;
else
    h3 = errorldg('Berdeak ezin dira hartu!!!');
end

case 5 %%Urdinak hartuko ditu%%
kasua=4;
if(Zenb_Urdin>0)
    [ pos_bidalari , pos_itzuli ] = Lehentasanak( Zenb_Gorri , Zenb_Urdin , Zenb_Berde , Zirkulu_Gorri , Triangelu
clear all;
clear forma;
clear kolore;
close Automatiko;
Erabilera;
else
    h4 = errorldg('Urdinak ezin dira hartu!!!');
end

case 6 %%Gorriak eta urdinak hartuko ditu%%
kasua=5;
if(Zenb_Gorri>0)&&(Zenb_Urdin>0)
    [ pos_bidalari , pos_itzuli ] = Lehentasanak( Zenb_Gorri , Zenb_Urdin , Zenb_Berde , Zirkulu_Gorri , Triangelu
clear all;

clear forma;
clear kolore;
close Automatiko;
Erabilera;
else
    h4 = errorldg('Gorri+Urdin ezin da hartu!!!');
end

case 7 %%Gorriak eta Berdeak%%
kasua=6;
if(Zenb_Gorri>0)&&(Zenb_Berde>0)
    [ pos_bidalari , pos_itzuli ] = Lehentasanak( Zenb_Gorri , Zenb_Urdin , Zenb_Berde , Zirkulu_Gorri , Triangelu
clear all;
clear forma;
clear kolore;
close Automatiko;
Erabilera;
else
    h5 = errorldg('Gorri+Berde ezin da hartu!!!');
end

case 8 %%Urdinak eta Berdeak%%
kasua=7;
if(Zenb_Urdin>0)&&(Zenb_Berde>0)
    [ pos_bidalari , pos_itzuli ] = Lehentasanak( Zenb_Gorri , Zenb_Urdin , Zenb_Berde , Zirkulu_Gorri , Triangelu
clear all;
clear forma;
clear kolore;
close Automatiko;

-----
Erabilera;
else
    h6 = errorldg('Urdin+Berde ezin da hartu!!!');
end

end
```

Zati honetan erabiltzaileak sailkatu nahi duena adieraziko da. Non ondoren Lehentasanak funtzioarekin aukeratutakoaren arteko lehentasanak emango dira eta robotari bidaliko zaio informazioa.

## 1.1.5. Eskuzkoa

```

function varargout = Manuala(varargin) ...
% End initialization code - DO NOT EDIT

% --- Executes just before Manuala is made visible.
function Manuala_OpeningFcn(hObject, eventdata, handles, varargin)
global vid
global a

% Choose default command line output for Manual
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);
set(gcf,'currentaxes',handles.Bideo);
axis off; %ardatzak kenduko dira
set(gcf,'currentaxes',handles.Irudi);
axis off; %ardatzak kenduko dira
set(gcf,'currentaxes',handles.Irudi);
axis off; %ardatzak kenduko dira
set(gcf,'currentaxes',handles.Detekttatu);
axis off; %ardatzak kenduko dira
set(gcf,'currentaxes',handles.fondo);
b=imread('piezak.jpg'); %Fondo bezala paisajea jarriko da
image(b);

load ('forma');
load ('kolore');
if(forma==1)
    set(handles.pGorri,'Visible','off');
    set(handles.pUrdin,'Visible','off');
    set(handles.pBerde,'Visible','off');
    set(handles.aukeratu,'Visible','off');
    set(handles.textKolore,'Visible','off');
end
if(kolore==1)
    set(handles.pZir,'Visible','off');
    set(handles.pTri,'Visible','off');
    set(handles.pKar,'Visible','off');
    set(handles.aukeraketa,'Visible','off');
    set(handles.textForma,'Visible','off');
end

a=arduino('COM5'); %Arduinoa aktibatuko da

a.configureDigitalPin(2,'output'); %pos 1 digital sarrera
a.configureDigitalPin(3,'output'); %pos 2 digital sarrera
a.configureDigitalPin(4,'output'); %pos 3 digital irteera
a.configureDigitalPin(5,'output'); %pos 4 digital irteera
a.configureDigitalPin(6,'output'); %pieza=2 eta Gorria digital irteera
a.configureDigitalPin(7,'output'); %pieza=3 eta Berdea digital irteera
a.configureDigitalPin(8,'output'); %pieza=4 eta Urdina digital irteera
a.configureDigitalPin(9,'input'); %pos 1 digital sarrera
a.configureDigitalPin(10,'input'); %pos 2 digital sarrera
a.configureDigitalPin(11,'input'); %pos 3 digital sarrera
a.configureDigitalPin(12,'input'); %pos 4 digital sarrera
%%% Irteerako PIN denak 1-ean jarriko dira Robotak 0 bezal identifikatzen dituelako %%%%
a.writeDigitalPin(2,1);
a.writeDigitalPin(3,1);
a.writeDigitalPin(4,1);
a.writeDigitalPin(5,1);
a.writeDigitalPin(6,1);
a.writeDigitalPin(7,1);
a.writeDigitalPin(8,1);

%%% Bideoa momentuan bertan ikusten ari dena agertzeko %%%
vid=videoinput('winvideo',1,'YUY2_640x480'); %Bideorako kanala irekitzen da eta erresoluzioa aukeratu
erresoluzioa = get(vid, 'VideoResolution'); %Erresoluzioa neurtzen da
altuera = erresoluzioa(1); %Erresoluzioaren altueraren zatia
zabalera = erresoluzioa(2); %Erresoluzioaren zabalera zatia
ZenbBanda = get(vid, 'NumberOfBands'); %Banda kopurua
hImage = image( zeros(zabalera, altuera, ZenbBanda),'Parent',handles.Bideo); %Ezaugarri guztiak batuko dira
preview(vid,hImage) %Ikusten dena agertzen da GUI-an

```

Eskuzkoa irekitzen denean ARDUINO txartela konektatzen da eta bere pinen konfigurazioa egin. Ondoren, kamera konektatu eta interfazean bistaratuko da.

```
% --- Executes on selection change in menua.
function menua_Callback(hObject, eventdata, handles)

global a
global Pieza

balioa=get(handles.menua,'value'); %Menuan aukeratutakoa lortzen da

%%% Lortutakoaren arabera zer egin jakiteko %%%
switch balioa
    case 2 %%%2 pieza hartuko dira%%
        Pieza=2;
        a.writeDigitalPin(6,0);
        pause(7)
        a.writeDigitalPin(6,1)
    case 3 %%%3 pieza hartuko dira%%
        Pieza=3;
        a.writeDigitalPin(7,0);
        pause(7)
        a.writeDigitalPin(7,1)
    case 4 %%%4 pieza hartuko dira%%
        Pieza=4;
        a.writeDigitalPin(8,0);
        pause(7)
        a.writeDigitalPin(8,1)
end
```

Erabiltzaileak pieza kopurua aukeratzeko duenean irudiaren ikusten den programazioa sortuko da. Informazioa ARDUINO bidez robotera bidalita.

```
% --- Executes on button press in argazkia.
function argazkia_Callback(hObject, eventdata, handles)

global vid
global Irudia
global Pieza_kop
global L

Irudia=getsnapshot(vid); %Argazkia egingo da
Irudia=ycbcr2rgb(Irudia); %Argazkia RGB ingurura igaroko da
axes(handles.Irudi); %Argazkia erakutsiko da
imshow(Irudia);

Irudi_gris=rgb2gray(Irudia); %Argazkia grisetara igaroko da
[m,n]=size(Irudi_gris); %Argazkiaren neurria lortuko da
Irudi_binario=ones(m,n); %Irudi binarioaren fondoa guztian 1-ak jarriko dira
umbral=100; %Argazkia binarizatzeko Umbrala aukeratu da

%%% Irudia BINARIZATUKO da %%%
for i = 1:m
    for j = 1:n
        if (Irudi_gris(i,j)>umbral)
            Irudi_binario(i,j)=0;
        else
            Irudi_binario(i,j)=1;
        end
    end
end

Irudi_binario=medfilt2(Irudi_binario,[15,15]);
Irudi_binario=bwareaopen(Irudi_binario,900); %900 baino txikiagoko pixelak borratuko dira
[L, Pieza_kop]=bwlabel(Irudi_binario,8); %Objektuak lortuko dira baita kopurua ere
set(handles.Pieza,'String',Pieza_kop); %Pieza kopurua GUI-an ageriko da
```



Argazkia botoia sakatzen denean argazkia aterako da. Ondoren argazki hori binarizatu egingo da eta dauden objektuak detektatuko dira. Azkenik, emaitza horiek interfazean bistaratuko dira.

```
% --- Executes on button press in Prozesatu.
function Prozesatu_Callback(hObject, eventdata, handles)

    global Pieza_kop
    global Irudia

    global Zenb_Gorri
    global Zenb_Urdin
    global Zenb_Berde
    global Zirkulu_Gorri
    global Karratu_Gorri
    global Triangelu_Gorri
    global Zirkulu_Urdin
    global Karratu_Urdin
    global Triangelu_Urdin
    global Zirkulu_Berde
    global Karratu_Berde
    global Triangelu_Berde
    global Posizio_Gorri
    global Posizio_Urdin
    global Posizio_Berde
    global Pieza
    global Zirkulu_Pieza2
    global Triangelu_Pieza2
    global Karratu_Pieza2
    global L
    global Posizio_Pieza
    global Zirkulu_Pieza
    global Triangelu_Pieza
    global Karratu_Pieza

    Zirkulu_Gorri2=0;
    Karratu_Gorri2=0;
    Triangelu_Gorri2=0;
    Zirkulu_Urdin2=0;
    Karratu_Urdin2=0;
    Triangelu_Urdin2=0;
    Zirkulu_Berde2=0;
    Karratu_Berde2=0;
    Triangelu_Berde2=0;

    load ('forma');
    load ('kolore');
```

```

if Pieza_kop>0
    if(forma==1)

        l=1;
        a=1;
        Area_Pieza=[];
        Perimetro_Pieza=[];
        Zentro_Pieza=[];
        K_Pieza=[];
        Posizio_Pieza=[];

        axes(handles.Detektatu);
        imshow(Irudia);          %Irudia ageriko da detektatutako pieza guztiekin
        hold all;

        for i=1:Pieza_kop

            Stats_Pieza=regionprops(L,'Area','Perimeter','Centroid','BoundingBox');

            Area_Pieza=[Stats_Pieza.Area];          %Area balioa gordeko da
            Perimetro_Pieza=[Stats_Pieza.Perimeter]; %Perimetro balioa gordeko da
            Zentro_Pieza=[Stats_Pieza.Centroid];    %Zentro balioa gordeko da

            K_Pieza(1)=(Perimetro_Pieza(1)^2)/Area_Pieza(1); %Konstantea lortuko da

            l=l+1;

        end
        for ObjectPieza=1:length(Stats_Pieza)          %For bat sortuko da errektanguloak sortzeko
            Kaja_Pieza=Stats_Pieza(ObjectPieza).BoundingBox; %Kaxaren balioa

            rectangle('Position',Kaja_Pieza,'EdgeColor','b','LineWidth',2) %Kaxak margotuko dira
            plot(Zentro_Pieza(1),Zentro_Pieza(2),'-g+') %Errektanguloak eta zentroak irudikatuko dira
        end

        %%% Lortutako konstantearen arabera forma lortuko da %%%
        for x=1:Pieza_kop

            Zirkulu_Pieza(x)=0;
            Karratu_Pieza(x)=0;
            Triangelu_Pieza(x)=0;

            if ((10<K_Pieza(x)) && (K_Pieza(x)<13.3))          %Zirkulua izango da

                Zirkulu_Pieza(x)=1;
                Karratu_Pieza(x)=0;
                Triangelu_Pieza(x)=0;
            end
            if ((13.3<K_Pieza(x)) && (K_Pieza(x)<17))          %Karratua izango da

                Zirkulu_Pieza(x)=0;
                Karratu_Pieza(x)=1;
                Triangelu_Pieza(x)=0;
            end
            if ((17<K_Pieza(x)) && (K_Pieza(x)<22))          %Triangelua izango da

                Zirkulu_Pieza(x)=0;
                Karratu_Pieza(x)=0;
                Triangelu_Pieza(x)=1;
            end
        end
    end
end

```

```
    %%% Piezen posizioa lortuko da %%%
    if (Zentro_Pieza>0)

        k=1;
        for i=1:2:Pieza_kop*2

            if (Zentro_Pieza(i)<320)&&(Zentro_Pieza(i+1)<240)    % 1.posizioan dagoen ikusteko

                Posizio_Pieza(k)=1;
            end

            if (Zentro_Pieza(i)>320)&&(Zentro_Pieza(i+1)<240)    % 2.posizioan dagoen ikusteko

                Posizio_Pieza(k)=2;
            end

            if (Zentro_Pieza(i)<320)&&(Zentro_Pieza(i+1)>240)    % 3.posizioan dagoen ikusteko

                Posizio_Pieza(k)=3;
            end

            if (Zentro_Pieza(i)>320)&&(Zentro_Pieza(i+1)>240)    % 4.posizioan dagoen ikusteko

                Posizio_Pieza(k)=4;
            end

            k=k+1;
        end
    end
end
```

```
Zirkulu_Pieza2=sum(Zirkulu_Pieza);
set(handles.Zir, 'String', Zirkulu_Pieza2);    %Zirkulu kopurua agertuko da GUI-an
Triangelu_Pieza2=sum(Triangelu_Pieza);
set(handles.Tri, 'String', Triangelu_Pieza2);    %Zirkulu kopurua agertuko da GUI-an
Karratu_Pieza2=sum(Karratu_Pieza);
set(handles.Kar, 'String', Karratu_Pieza2);    %Karratu kopurua agertuko da GUI-an

Pieza_Oker=0;

if(Pieza_kop~=(Zirkulu_Pieza2+Triangelu_Pieza2+Karratu_Pieza2))

    Pieza_Oker=(Pieza_kop)-(Zirkulu_Pieza2+Triangelu_Pieza2+Karratu_Pieza2);

end

set(handles.PiezaOker, 'String', Pieza_Oker);
Pieza=Pieza-Pieza_Oker;
end
```

Prozesatu botoia sakatzen denean aurretik ikusitako programazioa egingo da. Formetan sailkapena aukeratu baldin bada aurretik detektatu diren objektuen informazioa lortuko da eta ondoren ezaugarri horien arabera forma eta posizioa lortuko da. Azkenik, emaitzak bistaratuko dira.

Koloreen araberako sailkapena aukeratu bada, prozesatu botoia sakatzean:

```

if(kolore==1)

    gorri=imsubtract(Irudia(:,:,1),rgb2gray(Irudia)); %Irudian pixel gorriak isolatuko dira
    gorri=medfilt2(gorri,[15,15]); % [15,15]-eko iragazki bat erabiliko da
    gorri=im2bw(gorri,0.26); %Irudia Grisetatik binariora igaroko da
    gorri=bwareaopen(gorri,300); %300 baino gutxiagoko pixelak borratuko dira

    berde=imsubtract(Irudia(:,:,2),rgb2gray(Irudia)); %Irudian pixel berdeak isolatuko dira
    berde=medfilt2(berde,[15,15]); % [15,15]-eko iragazki bat erabiliko da
    berde=im2bw(berde,0.03); %Irudia Grisetatik binariora igaroko da
    berde=bwareaopen(berde,5000); %5000 baino gutxiagoko pixelak borratuko dira

    urdin=imsubtract(Irudia(:,:,3),rgb2gray(Irudia)); %Irudian pixel urdinak isolatuko dira
    urdin=medfilt2(urdin,[9,9]); % [9,9]-eko iragazki bat erabiliko da
    urdin=im2bw(urdin,0.18); %Irudia Grisetatik binariora igaroko da
    urdin=bwareaopen(urdin,500); %500 baino gutxiagoko pixelak borratuko dira

    [Lgorri, Zenb_Gorri]=bwlabel(gorri,8); %Objetua gorriak lortuko dira baita kopurua ere
    [LBerde, Zenb_Berde]=bwlabel(berde,8); %Objetua berdeak lortuko dira baita kopurua ere
    [LUrdin, Zenb_Urdin]=bwlabel(urdin,8); %Objetua urdinak lortuko dira baita kopurua ere

    set(handles.ZenbGorri,'String',Zenb_Gorri); %Zenbat pieza gorri dauden ageriko da GUI-an
    set(handles.ZenbBerde,'String',Zenb_Berde); %Zenbat pieza berde dauden ageriko da GUI-an
    set(handles.ZenbUrdin,'String',Zenb_Urdin); %Zenbat pieza urdin dauden ageriko da GUI-an
    Zentro_Gorri=[];
    Zentro_Berde=[];
    Zentro_Urdin=[];

    axes(handles.Detekttatu);
    imshow(Irudia); %Irudia ageriko da detektatutako pieza guztiekin
    hold all;

    %%% Pieza GORRIAK aztertuko dira %%%
    if Zenb_Gorri>0
        Stats_Gorri=regionprops(LGorri,'BoundingBox','Centroid'); %Pieza Gorrien ezaugarriak lortuko dira

        for ObjectGorri=1:length(Stats_Gorri) %For bat sortuko da errektanguloak sortzeko
            Kaja_Gorri=Stats_Gorri(ObjectGorri).BoundingBox; %Kaxaren balioa
            Zentro_Gorri=Stats_Gorri(ObjectGorri).Centroid; %Zentroidearen balioa

            rectangle('Position',Kaja_Gorri,'EdgeColor','b','LineWidth',2) %Kaxak margotuko dira
            plot(Zentro_Gorri(1),Zentro_Gorri(2),'-g+') %Errektanguloak eta zentroak irudikatuko dira
        end

        Zentro_Gorri=[Stats_Gorri.Centroid]; %Zentroaren balioa gordeko da array baten
    end

    %%% Pieza BERDEAK aztertuko dira %%%
    if Zenb_Berde>0
        Stats_Berde=regionprops(LBerde,'BoundingBox','Centroid'); %Pieza Berdeen ezaugarriak lortuko dira

        for ObjectBerde=1:length(Stats_Berde) %For bat sortuko da errektanguloak sortzeko
            Kaja_Berde=Stats_Berde(ObjectBerde).BoundingBox; %Kaxaren balioa
            Zentro_Berde=Stats_Berde(ObjectBerde).Centroid; %Zentroidearen balioa

            rectangle('Position',Kaja_Berde,'EdgeColor','b','LineWidth',2) %Kaxak margotuko dira
            plot(Zentro_Berde(1),Zentro_Berde(2),'-g+') %Errektanguloak eta zentroak irudikatuko
        end

        Zentro_Berde=[Stats_Berde.Centroid]; %Zentroaren balioa gordeko da array baten
    end
end

```

```
%%% Pieza URDINAK aztertuko dira %%%
if Zenb_Urdin>0

    Stats_Urdin=regionprops(LUrdin,'BoundingBox','Centroid'); %Pieza Urdinen ezaugarriak lortuko dira

    for ObjectUrdin=1:length(Stats_Urdin) %For bat sortuko da errektanguloak sortzeko
        Kaja_Urdin=Stats_Urdin(ObjectUrdin).BoundingBox; %Kaxaren balioa
        Zentro_Urdin=Stats_Urdin(ObjectUrdin).Centroid; %Zentroidearen balioa

        rectangle('Position',Kaja_Urdin,'EdgeColor','r','LineWidth',2) %Kaxak margotuko diras
        plot(Zentro_Urdin(1),Zentro_Urdin(2),'-g+') %Errektanguloak eta zentroak irudikatuko
    end

    Zentro_Urdin=[Stats_Urdin.Centroid]; %Zentroaren balioa gordeko da array baten

end
```

```
%%% Pieza GORRIEN posizioa lortuko da %%%
if Zentro_Gorri>0
    j=1;
    for i=1:2:Zenb_Gorri*2

        if (Zentro_Gorri(i)<320) && (Zentro_Gorri(i+1)<240) % 1.posizioan dagoen ikusteko

            Posizio_Gorri(j)=1;
        end
        if (Zentro_Gorri(i)>320) && (Zentro_Gorri(i+1)<240) % 2.posizioan dagoen ikusteko

            Posizio_Gorri(j)=2;
        end
        if (Zentro_Gorri(i)<320) && (Zentro_Gorri(i+1)>240) % 3.posizioan dagoen ikusteko

            Posizio_Gorri(j)=3;
        end
        if (Zentro_Gorri(i)>320) && (Zentro_Gorri(i+1)>240) % 4.posizioan dagoen ikusteko

            Posizio_Gorri(j)=4;
        end
        j=j+1;
    end
else
    Posizio_Gorri=[];
end
```

```
    %%% Pieza BERDEEN posizioa lortuko da %%%
    if Zentro_Berde>0
        j=1;
        for i=1:2:Zenb_Berde*2

            if (Zentro_Berde(i)<320) && (Zentro_Berde(i+1)<240) % 1.posizioan dagoen ikusteko

                Posizio_Berde(j)=1;
            end
            if (Zentro_Berde(i)>320) && (Zentro_Berde(i+1)<240) % 2.posizioan dagoen ikusteko

                Posizio_Berde(j)=2;
            end
            if (Zentro_Berde(i)<320) && (Zentro_Berde(i+1)>240) % 3.posizioan dagoen ikusteko

                Posizio_Berde(j)=3;
            end
            if (Zentro_Berde(i)>320) && (Zentro_Berde(i+1)>240) % 4.posizioan dagoen ikusteko

                Posizio_Berde(j)=4;
            end
            j=j+1;
        end
    else
        Posizio_Berde=[];
    end
end
```

```
    %%% Pieza URDINEN posizioa lortuko da %%%
    if Zentro_Urdin>0
        j=1;
        for i=1:2:Zenb_Urdin*2
            if (Zentro_Urdin(i)<320) && (Zentro_Urdin(i+1)<240) % 1.posizioan dagoen ikusteko

                Posizio_Urdin(j)=1;
            end
            if (Zentro_Urdin(i)>320) && (Zentro_Urdin(i+1)<240) % 2.posizioan dagoen ikusteko

                Posizio_Urdin(j)=2;
            end
            if (Zentro_Urdin(i)<320) && (Zentro_Urdin(i+1)>240) % 3.posizioan dagoen ikusteko

                Posizio_Urdin(j)=3;
            end
            if (Zentro_Urdin(i)>320) && (Zentro_Urdin(i+1)>240) % 4.posizioan dagoen ikusteko

                Posizio_Urdin(j)=4;
            end
            j=j+1;
        end
    else
        Posizio_Urdin=[];
    end
end
```

```

    %%% Pieza GORRIEN forma lortuko da %%%
    if Zenb_Gorri>0

        j=1;
        a=1;
        Area_Gorri=[];
        Perimetro_Gorri=[];
        K_Gorri=[];

        for i=1:Zenb_Gorri

            Stats_Gorri=regionprops(LGorri,'Area','Perimeter');
            Area_Gorri=[Stats_Gorri.Area];
            Perimetro_Gorri=[Stats_Gorri.Perimeter];

            K_Gorri(j)=(Perimetro_Gorri(j)^2)/Area_Gorri(j);

            j=j+1;
        end

        %%% Lortutako konstantearen arabera forma lortuko da %%%
        for a=1:Zenb_Gorri

            Zirkulu_Gorri(a)=0;
            Karratu_Gorri(a)=0;
            Triangelu_Gorri(a)=0;
            if ((10<K_Gorri(a)) && (K_Gorri(a)<14))
                %Zirkulua izango da

                Zirkulu_Gorri(a)=1;
                Karratu_Gorri(a)=0;
                Triangelu_Gorri(a)=0;

            end
            if ((16<K_Gorri(a)) && (K_Gorri(a)<18))
                %Karratua izango da

                Zirkulu_Gorri(a)=0;
                Karratu_Gorri(a)=1;
                Triangelu_Gorri(a)=0;

            end
            if ((18<K_Gorri(a)) && (K_Gorri(a)<28))
                %Triangelua izango da

                Zirkulu_Gorri(a)=0;
                Karratu_Gorri(a)=0;
                Triangelu_Gorri(a)=1;

            end

            end

            Zirkulu_Gorri2=sum(Zirkulu_Gorri);
            set(handles.ZirGorri,'String',Zirkulu_Gorri2);
            Karratu_Gorri2=sum(Karratu_Gorri);
            set(handles.KarGorri,'String',Karratu_Gorri2);
            Triangelu_Gorri2=sum(Triangelu_Gorri);
            set(handles.TriGorri,'String',Triangelu_Gorri2);
        else

            Zirkulu_Gorri=0;
            Karratu_Gorri=0;
            Triangelu_Gorri=0;
        end
    end

```

```

    %%% Pieza BERDEEN forma lortuko da %%%
    if Zenb_Berde>0

        k=1;
        b=1;
        Area_Berde=[];
        Perimetro_Berde=[];
        K_Berde=[];

        for i=1:Zenb_Berde

            Stats_Berde=regionprops(LBerde, 'Area', 'Perimeter');
            Area_Berde=[Stats_Berde.Area];
            Perimetro_Berde=[Stats_Berde.Perimeter];

            K_Berde(k)=(Perimetro_Berde(k)^2)/Area_Berde(k);

            k=k+1;
        end

        %%% Lortutako konstantearen arabera forma lortuko da %%%
        for b=1:Zenb_Berde

            Zirkulu_Berde(b)=0;
            Karratu_Berde(b)=0;
            Triangelu_Berde(b)=0;
            if ((10<K_Berde(b)) && (K_Berde(b)<13.7))
                %Zirkulua izango da

                Zirkulu_Berde(b)=1;
                Karratu_Berde(b)=0;
                Triangelu_Berde(b)=0;

            end
            if ((13.7<K_Berde(b)) && (K_Berde(b)<20))
                %Karratua izango da

                Zirkulu_Berde(b)=0;
                Karratu_Berde(b)=1;
                Triangelu_Berde(b)=0;

            end
            if ((20<K_Berde(b)) && (K_Berde(b)<28))
                %Triangelua izango da

                Zirkulu_Berde(b)=0;
                Karratu_Berde(b)=0;
                Triangelu_Berde(b)=1;

            end

        end

        Zirkulu_Berde2=sum(Zirkulu_Berde);
        set(handles.ZirBerde, 'String', Zirkulu_Berde2);
        Karratu_Berde2=sum(Karratu_Berde);
        set(handles.KarBerde, 'String', Karratu_Berde2);
        Triangelu_Berde2=sum(Triangelu_Berde);
        set(handles.TriBerde, 'String', Triangelu_Berde2);
    else

        Zirkulu_Berde=0;
        Karratu_Berde=0;
        Triangelu_Berde=0;
    end

```



```

    %% Pieza BERDEEN forma lortuko da %%
    if Zenb_Urdin>0

        l=1;
        c=1;
        Area_Urdin=[];
        Perimetro_Urdin=[];
        K_Urdin=[];

        for i=1:Zenb_Urdin

            Stats_Urdin=regionprops(LUrdin,'Area','Perimeter');
            Area_Urdin=[Stats_Urdin.Area];
            Perimetro_Urdin=[Stats_Urdin.Perimeter];

            K_Urdin(l)=(Perimetro_Urdin(l)^2)/Area_Urdin(l);

            l=l+1;
        end

        %% Lortutako konstantearen arabera forma lortuko da %%
        for c=1:Zenb_Urdin

            Zirkulu_Urdin(c)=0;
            Karratu_Urdin(c)=0;
            Triangelu_Urdin(c)=0;
            if ((10<K_Urdin(c)) && (K_Urdin(c)<14)) %Zirkulua izango da

                Zirkulu_Urdin(c)=1;
                Karratu_Urdin(c)=0;
                Triangelu_Urdin(c)=0;

            end
            if ((14<K_Urdin(c)) && (K_Urdin(c)<19)) %Karratua izango da

                Zirkulu_Urdin(c)=0;
                Karratu_Urdin(c)=1;
                Triangelu_Urdin(c)=0;

            end
            if ((19<K_Urdin(c)) && (K_Urdin(c)<28)) %Triangelua izango da

                Zirkulu_Urdin(c)=0;
                Karratu_Urdin(c)=0;
                Triangelu_Urdin(c)=1;

            end

        end

        Zirkulu_Urdin2=sum(Zirkulu_Urdin);
        set(handles.ZirUrdin,'String',Zirkulu_Urdin2);
        Karratu_Urdin2=sum(Karratu_Urdin);
        set(handles.KarUrdin,'String',Karratu_Urdin2);
        Triangelu_Urdin2=sum(Triangelu_Urdin);
        set(handles.TriUrdin,'String',Triangelu_Urdin2);
    else

        Zirkulu_Urdin=0;
        Karratu_Urdin=0;
        Triangelu_Urdin=0;
    end

```

```
    Pieza_Oker=0;

    if(Pieza_kop~=(Zirkulu_Gorri2+Triangelu_Gorri2+Karratu_Gorri2+Zirkulu_Berde2+Triangelu_Berde2+Karratu_Berde2+Zir
        Pieza_Oker=(Pieza_kop-(Zirkulu_Gorri2+Triangelu_Gorri2+Karratu_Gorri2+Zirkulu_Berde2+Triangelu_Berde2+Karrat
    end
    if(Pieza_kop~=(Zenb_Gorri+Zenb_Urdin+Zenb_Berde))
        Pieza_Oker=(Pieza_kop)-(Zenb_Gorri+Zenb_Urdin+Zenb_Berde);
    end
    if(Pieza_kop~=(Zirkulu_Gorri2+Triangelu_Gorri2+Karratu_Gorri2+Zirkulu_Berde2+Triangelu_Berde2+Karratu_Berde2+Zir
        Pieza_Oker=(Pieza_kop)-(Zirkulu_Gorri2+Triangelu_Gorri2+Karratu_Gorri2+Zirkulu_Berde2+Triangelu_Berde2+Karrat
    end

    set(handles.PiezaOker,'String',Pieza_Oker);
    Pieza=Pieza-Pieza_Oker;
end
end
```

Koloreen arabera sailkapena eskatzen denean irudia hartzen da lehenik. Irudi hori koloreetako planoetan banantzen da lehenik. Plano horiek binarizatu egiten dira eta objektuak lortzen dira koloreen arabera. Behin objektuak lortu ditugula, objektu horren ezaugarriak lortzen ditugu. Eta ezaugarri horien arabera objektu horien forma eta posizioa lortzen dugu. Azkenik, emaitza horiek interfazean bistaratzen dira.

Behin objektuen forma eta posizioa jakinda, erabiltzaileak zer sailkatu nahi duen aukeratu du. Aurretik aukeraketa formetan izan bada:

```
% Formetan sailkatu menua sakatzean ejekutatzeko da %
function aukeraketa_Callback(hObject, eventdata, handles)

global Zirkulu_Pieza
global Triangelu_Pieza
global Karratu_Pieza
global Zirkulu_Pieza2
global Triangelu_Pieza2
global Karratu_Pieza2
global Posizio_Pieza
global pos_bidali
global pos_itzuli

global a
global Pieza

balioa2=get(handles.aukeraketa,'value'); %Menuan aukeraturakoa lortzen da

%%% Lortutakoaren arabera zer egin jakiteko %%%
switch balioa2
    case 2 %%%Forma denak hartuko ditu%%
        kasua=1;
        if (Zirkulu_Pieza2>0)&&(Triangelu_Pieza2>0)&&(Karratu_Pieza2>0)
            [ pos_bidali , pos_itzuli ] = Lehtasunak2( a , Posizio_Pieza , kasua , Zirkulu_Pieza2 , Triangelu_Pieza2 ,
            clear all;
            close Automatikoa;
            Erabilera;
        else
            h1 = errordlg('Hiru formak ezin dira hartu!!!');
        end
    case 3 %%%Zirkuluak hartuko ditu%%
        kasua=2;
        if (Zirkulu_Pieza2>0)
            [ pos_bidali , pos_itzuli ] = Lehtasunak2( a , Posizio_Pieza , kasua , Zirkulu_Pieza2 , Triangelu_Pieza2 ,
            clear all;
            close Automatikoa;
            Erabilera;
        else
            h2 = errordlg('Zirkuluak ezin dira hartu!!!');
        end
    case 4 %%%Triangeluak hartuko ditu%%
        kasua=3;
        if(Triangelu_Pieza2>0)
            [ pos_bidali , pos_itzuli ] = Lehtasunak2( a , Posizio_Pieza , kasua , Zirkulu_Pieza2 , Triangelu_Pieza2 ,
            clear all;
            close Automatikoa;
            Erabilera;
        else
            h3 = errordlg('Triangeluak ezin dira hartu!!!');
        end
    case 5 %%%Karratuak hartuko ditu%%
        kasua=4;
        if(Karratu_Pieza2>0)
            [ pos_bidali , pos_itzuli ] = Lehtasunak2( a , Posizio_Pieza , kasua , Zirkulu_Pieza2 , Triangelu_Pieza2 ,
            clear all;
            close Automatikoa;
            Erabilera;
        else
        end
    end
end
```

```
        h4 = errordlg('Karratuak ezin dira hartu!!!');
    end
case 6 %%Zirkuluak eta Triangeluak hartuko ditu%%
kasua=5;
if(Zirkulu_Pieza2>0)&&(Triangelu_Pieza2>0)
    [ pos_bidali , pos_itzuli ] = Lehtasunak2( a , Posizio_Pieza , kasua , Zirkulu_Pieza2 , Triangelu_Pieza2 ,
        clear all;
        close Automatiko;
        Erabilera;
    else
        h5 = errordlg('Zirkuluak eta Triangeluak ezin dira hartu!!!');
    end
case 7 %%Zirkuluak eta Karratuak hartuko ditu%%
kasua=6;
if(Zirkulu_Pieza2>0)&&(Karratu_Pieza2>0)
    [ pos_bidali , pos_itzuli ] = Lehtasunak2( a , Posizio_Pieza , kasua , Zirkulu_Pieza2 , Triangelu_Pieza2 ,
        clear all;
        close Automatiko;
        Erabilera;
    else
        h6 = errordlg('Zirkuluak eta Karratuak ezin dira hartu!!!');
    end
case 8 %%Triangeluak eta Karratuak hartuko ditu%%
kasua=7;
if(Triangelu_Pieza2>0)&&(Karratu_Pieza2>0)
    [ pos_bidali , pos_itzuli ] = Lehtasunak2( a , Posizio_Pieza , kasua , Zirkulu_Pieza2 , Triangelu_Pieza2 ,
        clear all;
        close Automatiko;
        Erabilera;

    else
        h7 = errordlg('Triangelu eta Karratuak ezin dira hartu!!!');
    end
end
end
```

Zati honetan erabiltzaileak sailkatu nahi duena adieraziko da. Non ondoren Lehtasunak2 funtzioarekin aukeratutakoaren arteko lehtasunak emango dira eta robotari bidaliko zaio informazioa.

Koloreen araberrako aukeratu bada:

```
% Koloretan sailkatu menua sakatzean ejetutatzen da %
function aukeratu_Callback(hObject, eventdata, handles)

    global a
    global Pieza

    global Zenb_Gorri
    global Zenb_Urdin
    global Zenb_Berde
    global Zirkulu_Gorri
    global Karratu_Gorri
    global Triangelu_Gorri
    global Posizio_Gorri
    global Posizio_Urdin
    global Zirkulu_Urdin
    global Karratu_Urdin
    global Triangelu_Urdin
    global Zirkulu_Berde
    global Karratu_Berde
    global Triangelu_Berde
    global Posizio_Berde

    global pos_bidali
    global pos_itzuli

    balioa=get(handles.aukeratu,'value'); %Menuan aukeratutakoa lortzen da

    %%% Lortutakoaren arabera zer egin jakiteko %%%
    switch balioa

        case 2 %%%Hiru koloreak hartuko ditu%%
            kasua=1;
            if (Zenb_Gorri>0)&&(Zenb_Urdin>0)&&(Zenb_Berde>0)
                [ pos_bidali , pos_itzuli ] = Lehetasunak( Zenb_Gorri , Zenb_Urdin , Zenb_Berde , Zirkulu_Gorri , Triangelu
                    clear all;
                    clear forma;
                    clear kolore;
                    close Automatikoa;
                    Erabilera;
            else
                h1 = errordlg('Hiru Koloreak ezin dira hartu!!!');
            end
        case 3 %%%Gorriak hartuko ditu%%
            kasua=2;
            if(Zenb_Gorri>0)
                [ pos_bidali , pos_itzuli ] = Lehetasunak( Zenb_Gorri , Zenb_Urdin , Zenb_Berde , Zirkulu_Gorri , Triangelu
                    clear all;
                    clear forma;
                    clear kolore;
                    close Automatikoa;
                    Erabilera;
            else
                h2 = errordlg('Gorriak ezin dira hartu!!!');
            end
        case 4 %%%Berdeak hartuko ditu%%
            kasua=3;
```

```
if(Zenb_Berde>0)
    [ pos_bidalari , pos_itzuli ] = Lehentasanak( Zenb_Gorri , Zenb_Urdin , Zenb_Berde , Zirkulu_Gorri , Triangelu
clear all;
clear forma;
clear kolore;
close Automatikoak;
Erabilera;
else
    h3 = errorldg('Berdeak ezin dira hartu!!!');
end

case 5 %%Urdinak hartuko ditu%%
kasua=4;
if(Zenb_Urdin>0)
    [ pos_bidalari , pos_itzuli ] = Lehentasanak( Zenb_Gorri , Zenb_Urdin , Zenb_Berde , Zirkulu_Gorri , Triangelu
clear all;
clear forma;
clear kolore;
close Automatikoak;
Erabilera;
else
    h4 = errorldg('Urdinak ezin dira hartu!!!');
end

case 6 %%Gorriak eta urdinak hartuko ditu%%
kasua=5;
if(Zenb_Gorri>0)&&(Zenb_Urdin>0)
    [ pos_bidalari , pos_itzuli ] = Lehentasanak( Zenb_Gorri , Zenb_Urdin , Zenb_Berde , Zirkulu_Gorri , Triangelu
clear all;

clear forma;
clear kolore;
close Automatikoak;
Erabilera;
else
    h4 = errorldg('Gorri+Urdin ezin da hartu!!!');
end

case 7 %%Gorriak eta Berdeak%%
kasua=6;
if(Zenb_Gorri>0)&&(Zenb_Berde>0)
    [ pos_bidalari , pos_itzuli ] = Lehentasanak( Zenb_Gorri , Zenb_Urdin , Zenb_Berde , Zirkulu_Gorri , Triangelu
clear all;
clear forma;
clear kolore;
close Automatikoak;
Erabilera;
else
    h5 = errorldg('Gorri+Berde ezin da hartu!!!');
end

case 8 %%Urdinak eta Berdeak%%
kasua=7;
if(Zenb_Urdin>0)&&(Zenb_Berde>0)
    [ pos_bidalari , pos_itzuli ] = Lehentasanak( Zenb_Gorri , Zenb_Urdin , Zenb_Berde , Zirkulu_Gorri , Triangelu
clear all;
clear forma;
clear kolore;
close Automatikoak;

-----
Erabilera;
else
    h6 = errorldg('Urdin+Berde ezin da hartu!!!');
end

end
```

Zati honetan erabiltzaileak sailkatu nahi duena adieraziko da. Non ondoren Lehentasanak funtzioarekin aukeratutakoaren arteko lehentasanak emango dira eta robotari bidaliko zaio informazioa.

### 1.1.6. Lehentasunak

Ondoren lehentasunak azalduko dira. Funtzio bi bereizten dira lehentasun hauetan. Alde batetik, koloreen araberrako lehentasunak eta bestetik, formen araberrako lehentasunak.

Koloreen araberrako lehentasunak badira, funtzioa Lehentasunak izena edukiko du. Kasuetako bat adieraziko da bakarrik beste denak berdinak baitira:

```
function [ pos_bidali , pos_itzuli ] = Lehentasunak( Zenb_Gorri , Zenb_Urdin
if(kasua==1)
    if (Zenb_Gorri>0) && (Zenb_Urdin>0) && (Zenb_Berde>0)
        Zirkulu_Gorri2=sum(Zirkulu_Gorri);
        Karratu_Gorri2=sum(Karratu_Gorri);
        Triangelu_Gorri2=sum(Triangelu_Gorri);
        Zirkulu_Urdin2=sum(Zirkulu_Urdin);
        Karratu_Urdin2=sum(Karratu_Urdin);
        Triangelu_Urdin2=sum(Triangelu_Urdin);
        Zirkulu_Berde2=sum(Zirkulu_Berde);
        Karratu_Berde2=sum(Karratu_Berde);
        Triangelu_Berde2=sum(Triangelu_Berde);
        if(Pieza==3)
            if (Zenb_Gorri==1) && (Zenb_Urdin==1) && (Zenb_Berde==1)
                pos_bidali=(Posizio_Gorri(1))+1;
                pos_itzuli=(Posizio_Gorri(1))+8;
                while a.readDigitalPin(pos_itzuli)==0
                    a.writeDigitalPin((pos_bidali),0)
                    a.writeDigitalPin(6,0);
                end
                a.writeDigitalPin(pos_bidali,1)
                a.writeDigitalPin(6,1)
                pos_bidali=(Posizio_Urdin(1))+1;
                pos_itzuli=(Posizio_Urdin(1))+8;
                while a.readDigitalPin(pos_itzuli)==0
                    a.writeDigitalPin((pos_bidali),0)
                    a.writeDigitalPin(7,0);
                end
                a.writeDigitalPin(pos_bidali,1)
                a.writeDigitalPin(7,1)
                pos_bidali=(Posizio_Berde(1))+1;
                pos_itzuli=(Posizio_Berde(1))+8;
                while a.readDigitalPin(pos_itzuli)==0
                    a.writeDigitalPin((pos_bidali),0)
                    a.writeDigitalPin(8,0);
                end
                a.writeDigitalPin(pos_bidali,1)
                a.writeDigitalPin(8,1)
            end
        end
    end
    if(Pieza==4)
        if (Zenb_Gorri==2) && (Zenb_Urdin==1) && (Zenb_Berde==1)
            if(Zirkulu_Gorri2==2)
                A=find(Zirkulu_Gorri==1);
                pos_bidali=(Posizio_Gorri(A(1)))+1;
                pos_itzuli=(Posizio_Gorri(A(1)))+8;
                while a.readDigitalPin(pos_itzuli)==0
                    a.writeDigitalPin((pos_bidali),0)
                end
            end
        end
    end
end
```

```
        a.writeDigitalPin(6,0);
    end
    a.writeDigitalPin(pos_bidali,1)
    a.writeDigitalPin(6,1)

    pos_bidali=(Posizio_Gorri(A(2)))+1;
    pos_itzuli=(Posizio_Gorri(A(2)))+8;
    while a.readDigitalPin(pos_itzuli)==0
        a.writeDigitalPin((pos_bidali),0)
        a.writeDigitalPin(6,0);
    end
    a.writeDigitalPin(pos_bidali,1)
    a.writeDigitalPin(6,1)
end
if(Triangelu_Gorri2==2)
    A=find(Triangelu_Gorri==1);
    pos_bidali=(Posizio_Gorri(A(1)))+1;
    pos_itzuli=(Posizio_Gorri(A(1)))+8;
    while a.readDigitalPin(pos_itzuli)==0
        a.writeDigitalPin((pos_bidali),0)
        a.writeDigitalPin(6,0);
    end
    a.writeDigitalPin(pos_bidali,1)
    a.writeDigitalPin(6,1)

    pos_bidali=(Posizio_Gorri(A(2)))+1;
    pos_itzuli=(Posizio_Gorri(A(2)))+8;
    while a.readDigitalPin(pos_itzuli)==0
        a.writeDigitalPin((pos_bidali),0)
        a.writeDigitalPin(6,0);
    end
    a.writeDigitalPin(pos_bidali,1)
    a.writeDigitalPin(6,1)
end
if(Karratu_Gorri2==2)
    A=find(Karratu_Gorri==1);
    pos_bidali=(Posizio_Gorri(A(1)))+1;
    pos_itzuli=(Posizio_Gorri(A(1)))+8;
    while a.readDigitalPin(pos_itzuli)==0
        a.writeDigitalPin((pos_bidali),0)
        a.writeDigitalPin(6,0);
    end
    a.writeDigitalPin(pos_bidali,1)
    a.writeDigitalPin(6,1)

    pos_bidali=(Posizio_Gorri(A(2)))+1;
    pos_itzuli=(Posizio_Gorri(A(2)))+8;
    while a.readDigitalPin(pos_itzuli)==0
        a.writeDigitalPin((pos_bidali),0)
        a.writeDigitalPin(6,0);
    end
    a.writeDigitalPin(pos_bidali,1)
    a.writeDigitalPin(6,1)
end
if(Zirkulu_Gorri2==1) && (Triangelu_Gorri2==1)
    A=find(Zirkulu_Gorri==1);
    pos_bidali=(Posizio_Gorri(A))+1;
    pos_itzuli=(Posizio_Gorri(A))+8;
    while a.readDigitalPin(pos_itzuli)==0
        a.writeDigitalPin((pos_bidali),0)
        a.writeDigitalPin(6,0);
    end
    a.writeDigitalPin(pos_bidali,1)
    a.writeDigitalPin(6,1)

    B=find(Triangelu_Gorri==1);
    pos_bidali=(Posizio_Gorri(B))+1;
    pos_itzuli=(Posizio_Gorri(B))+8;
    while a.readDigitalPin(pos_itzuli)==0
        a.writeDigitalPin((pos_bidali),0)
        a.writeDigitalPin(6,0);
    end
    a.writeDigitalPin(pos_bidali,1)
    a.writeDigitalPin(6,1)
end
if(Zirkulu_Gorri2==1) && (Karratu_Gorri2==1)
    A=find(Zirkulu_Gorri==1);
    pos_bidali=(Posizio_Gorri(A))+1;
    pos_itzuli=(Posizio_Gorri(A))+8;
    while a.readDigitalPin(pos_itzuli)==0
        a.writeDigitalPin((pos_bidali),0)
        a.writeDigitalPin(6,0);
    end
    a.writeDigitalPin(pos_bidali,1)
    a.writeDigitalPin(6,1)

    C=find(Karratu_Gorri==1);
```



```

pos_bidali=(Posizio_Gorri(C))+1;
pos_itzuli=(Posizio_Gorri(C))+8;
while a.readDigitalPin(pos_itzuli)==0
    a.writeDigitalPin((pos_bidali),0)
    a.writeDigitalPin(6,0);
end
a.writeDigitalPin(pos_bidali,1)
a.writeDigitalPin(6,1)
end
if(Triangelu_Gorri2==1)&&(Karratu_Gorri2==1)
    B=find(Triangelu_Gorri==1);
    pos_bidali=(Posizio_Gorri(B))+1;
    pos_itzuli=(Posizio_Gorri(B))+8;
    while a.readDigitalPin(pos_itzuli)==0
        a.writeDigitalPin((pos_bidali),0)
        a.writeDigitalPin(6,0);
    end
    a.writeDigitalPin(pos_bidali,1)
    a.writeDigitalPin(6,1)
end
C=find(Karratu_Gorri==1);
pos_bidali=(Posizio_Gorri(C))+1;
pos_itzuli=(Posizio_Gorri(C))+8;
while a.readDigitalPin(pos_itzuli)==0
    a.writeDigitalPin((pos_bidali),0)
    a.writeDigitalPin(6,0);
end
a.writeDigitalPin(pos_bidali,1)
a.writeDigitalPin(6,1)
end
pos_bidali=(Posizio_Urdin(1))+1;
pos_itzuli=(Posizio_Urdin(1))+8;
while a.readDigitalPin(pos_itzuli)==0
    a.writeDigitalPin((pos_bidali),0)
    a.writeDigitalPin(7,0);
end
a.writeDigitalPin(pos_bidali,1)
a.writeDigitalPin(7,1)
end
pos_bidali=(Posizio_Berde(1))+1;
pos_itzuli=(Posizio_Berde(1))+8;
while a.readDigitalPin(pos_itzuli)==0
    a.writeDigitalPin((pos_bidali),0)
    a.writeDigitalPin(8,0);
end
a.writeDigitalPin(pos_bidali,1)
a.writeDigitalPin(8,1)
end
if(Zenb_Gorri==1)&&(Zenb_Urdin==2)&&(Zenb_Berde==1)
    pos_bidali=(Posizio_Gorri(1))+1;
    pos_itzuli=(Posizio_Gorri(1))+8;
    while a.readDigitalPin(pos_itzuli)==0
        a.writeDigitalPin((pos_bidali),0)
        a.writeDigitalPin(6,0);
    end
    a.writeDigitalPin(pos_bidali,1)
    a.writeDigitalPin(6,1)
end
if(Zirkulu_Urdin2==2)
    A=find(Zirkulu_Urdin==1);
    pos_bidali=(Posizio_Urdin(A(1)))+1;
    pos_itzuli=(Posizio_Urdin(A(1)))+8;
    while a.readDigitalPin(pos_itzuli)==0
        a.writeDigitalPin((pos_bidali),0)
        a.writeDigitalPin(7,0);
    end
    a.writeDigitalPin(pos_bidali,1)
    a.writeDigitalPin(7,1)
end
pos_bidali=(Posizio_Urdin(A(2)))+1;
pos_itzuli=(Posizio_Urdin(A(2)))+8;
while a.readDigitalPin(pos_itzuli)==0
    a.writeDigitalPin((pos_bidali),0)
    a.writeDigitalPin(7,0);
end
a.writeDigitalPin(pos_bidali,1)
a.writeDigitalPin(7,1)
end
if(Triangelu_Urdin2==2)
    A=find(Triangelu_Urdin==1);
    pos_bidali=(Posizio_Urdin(A(1)))+1;
    pos_itzuli=(Posizio_Urdin(A(1)))+8;
    while a.readDigitalPin(pos_itzuli)==0
        a.writeDigitalPin((pos_bidali),0)
        a.writeDigitalPin(7,0);
    end
    a.writeDigitalPin(pos_bidali,1)
end

```

```

a.writeDigitalPin(7,1)

pos_bidali=(Posizio_Urdin(A(2)))+1;
pos_itzuli=(Posizio_Urdin(A(2)))+8;
while a.readDigitalPin(pos_itzuli)==0
  a.writeDigitalPin((pos_bidali),0)
  a.writeDigitalPin(7,0);
end
a.writeDigitalPin(pos_bidali,1)
a.writeDigitalPin(7,1)
end
if(Karratu_Urdin2==2)
A=find(Karratu_Urdin==1);
pos_bidali=(Posizio_Urdin(A(1)))+1;
pos_itzuli=(Posizio_Urdin(A(1)))+8;
while a.readDigitalPin(pos_itzuli)==0
  a.writeDigitalPin((pos_bidali),0)
  a.writeDigitalPin(7,0);
end
a.writeDigitalPin(pos_bidali,1)
a.writeDigitalPin(7,1)

pos_bidali=(Posizio_Urdin(A(2)))+1;
pos_itzuli=(Posizio_Urdin(A(2)))+8;
while a.readDigitalPin(pos_itzuli)==0
  a.writeDigitalPin((pos_bidali),0)
  a.writeDigitalPin(7,0);
end
a.writeDigitalPin(pos_bidali,1)
a.writeDigitalPin(7,1)
end
if(Zirkulu_Urdin2==1)&&(Triangelu_Urdin2==1)
A=find(Zirkulu_Urdin==1);
pos_bidali=(Posizio_Urdin(A))+1;
pos_itzuli=(Posizio_Urdin(A))+8;
while a.readDigitalPin(pos_itzuli)==0
  a.writeDigitalPin((pos_bidali),0)
  a.writeDigitalPin(7,0);
end
a.writeDigitalPin(pos_bidali,1)
a.writeDigitalPin(7,1)

B=find(Triangelu_Urdin==1);
pos_bidali=(Posizio_Urdin(B))+1;
pos_itzuli=(Posizio_Urdin(B))+8;
while a.readDigitalPin(pos_itzuli)==0
  a.writeDigitalPin((pos_bidali),0)
  a.writeDigitalPin(7,0);
end
a.writeDigitalPin(pos_bidali,1)
a.writeDigitalPin(7,1)
end
if(Zirkulu_Urdin2==1)&&(Karratu_Urdin2==1)
A=find(Zirkulu_Urdin==1);
pos_bidali=(Posizio_Urdin(A))+1;
pos_itzuli=(Posizio_Urdin(A))+8;
while a.readDigitalPin(pos_itzuli)==0
  a.writeDigitalPin((pos_bidali),0)
  a.writeDigitalPin(7,0);
end
a.writeDigitalPin(pos_bidali,1)
a.writeDigitalPin(7,1)

C=find(Karratu_Urdin==1);
pos_bidali=(Posizio_Urdin(C))+1;
pos_itzuli=(Posizio_Urdin(C))+8;
while a.readDigitalPin(pos_itzuli)==0
  a.writeDigitalPin((pos_bidali),0)
  a.writeDigitalPin(7,0);
end
a.writeDigitalPin(pos_bidali,1)
a.writeDigitalPin(7,1)
end
if(Triangelu_Urdin2==1)&&(Karratu_Urdin2==1)
B=find(Triangelu_Urdin==1);
pos_bidali=(Posizio_Urdin(B))+1;
pos_itzuli=(Posizio_Urdin(B))+8;
while a.readDigitalPin(pos_itzuli)==0
  a.writeDigitalPin((pos_bidali),0)
  a.writeDigitalPin(7,0);
end
a.writeDigitalPin(pos_bidali,1)
a.writeDigitalPin(7,1)

C=find(Karratu_Urdin==1);
pos_bidali=(Posizio_Urdin(C))+1;
pos_itzuli=(Posizio_Urdin(C))+8;

```

```

pos_itzuli=(Posizio_Urdin(C))+8;|
while a.readDigitalPin(pos_itzuli)==0|
a.writeDigitalPin((pos_bidali),0)|
a.writeDigitalPin(7,0);|
end|
a.writeDigitalPin(pos_bidali,1)|
a.writeDigitalPin(7,1)|
end|
pos_bidali=(Posizio_Berde(1))+1;|
pos_itzuli=(Posizio_Berde(1))+8;|
while a.readDigitalPin(pos_itzuli)==0|
a.writeDigitalPin((pos_bidali),0)|
a.writeDigitalPin(8,0);|
end|
a.writeDigitalPin(pos_bidali,1)|
a.writeDigitalPin(8,1)|
end|
if(Zenb_Gorri==1) && (Zenb_Urdin==1) && (Zenb_Berde==2)|
pos_bidali=(Posizio_Gorri(1))+1;|
pos_itzuli=(Posizio_Gorri(1))+8;|
while a.readDigitalPin(pos_itzuli)==0|
a.writeDigitalPin((pos_bidali),0)|
a.writeDigitalPin(6,0);|
end|
a.writeDigitalPin(pos_bidali,1)|
a.writeDigitalPin(6,1)|
end|
pos_bidali=(Posizio_Urdin(1))+1;|
pos_itzuli=(Posizio_Urdin(1))+8;|
while a.readDigitalPin(pos_itzuli)==0|
a.writeDigitalPin((pos_bidali),0)|
a.writeDigitalPin(7,0);|
end|
a.writeDigitalPin(pos_bidali,1)|
a.writeDigitalPin(7,1)|
end|
if(Zirkulu_Berde2==2)|
A=find(Zirkulu_Berde==1);|
pos_bidali=(Posizio_Berde(A(1)))+1;|
pos_itzuli=(Posizio_Berde(A(1)))+8;|
while a.readDigitalPin(pos_itzuli)==0|
a.writeDigitalPin((pos_bidali),0)|
a.writeDigitalPin(8,0);|
end|
a.writeDigitalPin(pos_bidali,1)|
a.writeDigitalPin(8,1)|
end|
pos_bidali=(Posizio_Berde(A(2)))+1;|
pos_itzuli=(Posizio_Berde(A(2)))+8;|
while a.readDigitalPin(pos_itzuli)==0|
a.writeDigitalPin((pos_bidali),0)|
a.writeDigitalPin(8,0);|
end|
a.writeDigitalPin(pos_bidali,1)|
a.writeDigitalPin(8,1)|
end|
if(Triangelu_Berde2==2)|
A=find(Triangelu_Berde==1);|
pos_bidali=(Posizio_Berde(A(1)))+1;|
pos_itzuli=(Posizio_Berde(A(1)))+8;|
while a.readDigitalPin(pos_itzuli)==0|
a.writeDigitalPin((pos_bidali),0)|
a.writeDigitalPin(8,0);|
end|
a.writeDigitalPin(pos_bidali,1)|
a.writeDigitalPin(8,1)|
end|
pos_bidali=(Posizio_Berde(A(2)))+1;|
pos_itzuli=(Posizio_Berde(A(2)))+8;|
while a.readDigitalPin(pos_itzuli)==0|
a.writeDigitalPin((pos_bidali),0)|
a.writeDigitalPin(8,0);|
end|
a.writeDigitalPin(pos_bidali,1)|
a.writeDigitalPin(8,1)|
end|
if(Karratu_Berde2==2)|
A=find(Karratu_Berde==1);|
pos_bidali=(Posizio_Berde(A(1)))+1;|
pos_itzuli=(Posizio_Berde(A(1)))+8;|
while a.readDigitalPin(pos_itzuli)==0|
a.writeDigitalPin((pos_bidali),0)|
a.writeDigitalPin(8,0);|
end|
a.writeDigitalPin(pos_bidali,1)|
a.writeDigitalPin(8,1)|
end

```



Formen araberako sailkapena eskatu baldin bada erabiliko den funtzioa Lehenetasunak2 izango da:

```
function [ pos_bidali , pos_itzuli ] = Lehenetasunak2( a , Posizio_Pieza , kasua , Zirkulu
if(kasua==1)%%Hiru Formak hartzen ditu%%
    if(Zirkulu_Pieza2>0) && (Triangelu_Pieza2>0) && (Karratu_Pieza2>0)
        if(Pieza==3)
            if(Zirkulu_Pieza2==1) && (Triangelu_Pieza2==1) && (Karratu_Pieza2==1)
                A=find(Zirkulu_Pieza==1);
                pos_bidali=(Posizio_Pieza(A(1)))+1;
                pos_itzuli=(Posizio_Pieza(A(1)))+8;
                while a.readDigitalPin(pos_itzuli)==0
                    a.writeDigitalPin((pos_bidali),0)
                    a.writeDigitalPin(6,0);
                end
                a.writeDigitalPin(pos_bidali,1)
                a.writeDigitalPin(6,1)

                B=find(Triangelu_Pieza==1);
                pos_bidali=(Posizio_Pieza(B(1)))+1;
                pos_itzuli=(Posizio_Pieza(B(1)))+8;
                while a.readDigitalPin(pos_itzuli)==0
                    a.writeDigitalPin((pos_bidali),0)
                    a.writeDigitalPin(7,0);
                end
                a.writeDigitalPin(pos_bidali,1)
                a.writeDigitalPin(7,1)

                C=find(Karratu_Pieza==1);
                pos_bidali=(Posizio_Pieza(C(1)))+1;
                pos_itzuli=(Posizio_Pieza(C(1)))+8;
                while a.readDigitalPin(pos_itzuli)==0
                    a.writeDigitalPin((pos_bidali),0)
                    a.writeDigitalPin(8,0);
                end
                a.writeDigitalPin(pos_bidali,1)
                a.writeDigitalPin(8,1)
            end
        end
        if(Pieza==4)
            if(Zirkulu_Pieza2==2) && (Triangelu_Pieza2==1) && (Karratu_Pieza2==1)
                A=find(Zirkulu_Pieza==1);
                pos_bidali=(Posizio_Pieza(A(1)))+1;
                pos_itzuli=(Posizio_Pieza(A(1)))+8;
                while a.readDigitalPin(pos_itzuli)==0
                    a.writeDigitalPin((pos_bidali),0)
                    a.writeDigitalPin(6,0);
                end
                a.writeDigitalPin(pos_bidali,1)
                a.writeDigitalPin(6,1)

                pos_bidali=(Posizio_Pieza(A(2)))+1;
                pos_itzuli=(Posizio_Pieza(A(2)))+8;
                while a.readDigitalPin(pos_itzuli)==0
                    a.writeDigitalPin((pos_bidali),0)
                    a.writeDigitalPin(6,0);
                end
                a.writeDigitalPin(pos_bidali,1)
                a.writeDigitalPin(6,1)

                B=find(Triangelu_Pieza==1);
                pos_bidali=(Posizio_Pieza(B(1)))+1;
                pos_itzuli=(Posizio_Pieza(B(1)))+8;
                while a.readDigitalPin(pos_itzuli)==0
                    a.writeDigitalPin((pos_bidali),0)
                    a.writeDigitalPin(7,0);
                end
                a.writeDigitalPin(pos_bidali,1)
                a.writeDigitalPin(7,1)

                C=find(Karratu_Pieza==1);
                pos_bidali=(Posizio_Pieza(C(1)))+1;
                pos_itzuli=(Posizio_Pieza(C(1)))+8;
                while a.readDigitalPin(pos_itzuli)==0
                    a.writeDigitalPin((pos_bidali),0)
                    a.writeDigitalPin(8,0);
                end
                a.writeDigitalPin(pos_bidali,1)
                a.writeDigitalPin(8,1)
            end
        end
        if(Zirkulu_Pieza2==1) && (Triangelu_Pieza2==2) && (Karratu_Pieza2==1)
            A=find(Zirkulu_Pieza==1);
            pos_bidali=(Posizio_Pieza(A(1)))+1;
            pos_itzuli=(Posizio_Pieza(A(1)))+8;
            while a.readDigitalPin(pos_itzuli)==0
                a.writeDigitalPin((pos_bidali),0)
                a.writeDigitalPin(6,0);
            end
        end
    end
end
```

```
a.writeDigitalPin(pos_bidali,1)
a.writeDigitalPin(6,1)

B=find(Triangelu_Pieza==1);
pos_bidali=(Posizio_Pieza(B(1)))+1;
pos_itzuli=(Posizio_Pieza(B(1)))+8;
while a.readDigitalPin(pos_itzuli)==0
  a.writeDigitalPin((pos_bidali),0)
  a.writeDigitalPin(7,0);
end
a.writeDigitalPin(pos_bidali,1)
a.writeDigitalPin(7,1)

pos_bidali=(Posizio_Pieza(B(2)))+1;
pos_itzuli=(Posizio_Pieza(B(2)))+8;
while a.readDigitalPin(pos_itzuli)==0
  a.writeDigitalPin((pos_bidali),0)
  a.writeDigitalPin(7,0);
end
a.writeDigitalPin(pos_bidali,1)
a.writeDigitalPin(7,1)

C=find(Karratu_Pieza==1);
pos_bidali=(Posizio_Pieza(C(1)))+1;
pos_itzuli=(Posizio_Pieza(C(1)))+8;
while a.readDigitalPin(pos_itzuli)==0
  a.writeDigitalPin((pos_bidali),0)
  a.writeDigitalPin(8,0);
end
a.writeDigitalPin(pos_bidali,1)
a.writeDigitalPin(8,1)

end
if(Zirkulu_Pieza2==1)&&(Triangelu_Pieza2==1)&&(Karratu_Pieza2==2)
  A=find(Zirkulu_Pieza==1);
  pos_bidali=(Posizio_Pieza(A(1)))+1;
  pos_itzuli=(Posizio_Pieza(A(1)))+8;
  while a.readDigitalPin(pos_itzuli)==0
    a.writeDigitalPin((pos_bidali),0)
    a.writeDigitalPin(6,0);
  end
  a.writeDigitalPin(pos_bidali,1)
  a.writeDigitalPin(6,1)

  B=find(Triangelu_Pieza==1);
  pos_bidali=(Posizio_Pieza(B(1)))+1;
  pos_itzuli=(Posizio_Pieza(B(1)))+8;
  while a.readDigitalPin(pos_itzuli)==0
    a.writeDigitalPin((pos_bidali),0)
    a.writeDigitalPin(7,0);
  end
  a.writeDigitalPin(pos_bidali,1)
  a.writeDigitalPin(7,1)

  C=find(Karratu_Pieza==1);
  pos_bidali=(Posizio_Pieza(C(1)))+1;
  pos_itzuli=(Posizio_Pieza(C(1)))+8;
  while a.readDigitalPin(pos_itzuli)==0
    a.writeDigitalPin((pos_bidali),0)
    a.writeDigitalPin(8,0);
  end
  a.writeDigitalPin(pos_bidali,1)
  a.writeDigitalPin(8,1)

  pos_bidali=(Posizio_Pieza(C(2)))+1;
  pos_itzuli=(Posizio_Pieza(C(2)))+8;
  while a.readDigitalPin(pos_itzuli)==0
    a.writeDigitalPin((pos_bidali),0)
    a.writeDigitalPin(8,0);
  end
  a.writeDigitalPin(pos_bidali,1)
  a.writeDigitalPin(8,1)

end
end
end
end
```

Aukeratutako kasua forma denak hartzekoa izan da. Horretarako zein objektu dauden jakin beharra dago eta zein posiziotan. Hori jakinda dagoenean hartu beharreko ordena adierazten da. Eta robotari bidaltzen zaio hartu beharreko horren posizioa eta izango duen forma. Robotak erantzuna bidaltzen ez duen bitartean programak ez du aurrera egingo. Behin robotak erantzuna ematen duenean hartu beharreko hurrengo objektuaren posizioa eta kolorea bidaliko zaio eta horrela eskatutako denak hartu arte.

## 1.2. ROBOCELL programazioa

Robocell programazioa hasten denean lehenik eta behin erabiltzaileak bidali dituen pieza kopuruaren arabera subrutina desberdinetara joango da non *Pieza\_Esk* aldagaian erabiltzaileak esaktutakoa gordeko den.

```
1 Open Gripper
2 Turn Off Output 1
3 Turn Off Output 2
4 Turn Off Output 3
5 Turn Off Output 4
6 Set Variable PIEZA_KOP = 0
7 If Input 5 On Call Subroutine 2PIEZA
8 If Input 6 On Call Subroutine 3PIEZA
9 If Input 7 On Call Subroutine 4PIEZA
```

```
96 Set Subroutine 2PIEZA
97 Set Variable PIEZA_ESK = 2
98 Return from Subroutine
99 Set Subroutine 3PIEZA
100 Set Variable PIEZA_ESK = 3
101 Return from Subroutine
102 Set Subroutine 4PIEZA
103 Set Variable PIEZA_ESK = 4
104 Return from Subroutine
```

Pieza kopurua zein den jakin denean piezak hartzen hasiko da robota. Horretarako zinta martxan jarriko da sentsoreak pieza detektatu arte. Sentsoreak pieza detektatzean zinta geratu egingo da eta robotak pieza posiziotara eramaten hasiko da:

```
10 HASIERA:
11 Reset Timer
12 Start Conveyor Axis 7 at Speed 17 (%) in Plus Direction
13 Jump to SENSORE_DETEKTATU
14 SENSORE_DETEKTATU:
15 If Input 8 On Jump to GELDITU
16 Set Variable ZINTA to Timer
17 If ZINTA > 100 Jump to AMAITU
18 Jump to SENSORE_DETEKTATU
19 GELDITU:
20 Stop Conveyor Axis 7
21 If PIEZA_KOP == 0 Jump to POSIZIO4
22 If PIEZA_KOP == 1 Jump to POSIZIO2
23 If PIEZA_KOP == 2 Jump to POSIZIO1
24 If PIEZA_KOP == 3 Jump to POSIZIO3
```

```
25 POSIZIO4:  
26 Go to Position 9 Speed 75 (%)  
27 Go to Position 10 Speed 75 (%)  
28 Close Gripper  
29 Go to Position 1 Speed 75 (%)  
30 Go to Position 7 Speed 75 (%)  
31 Go to Position 8 Speed 75 (%)  
32 Open Gripper  
33 Go to Position 7 Speed 75 (%)  
34 Go to Position 1 Speed 75 (%)  
35 Go to Position 9 Speed 75 (%)  
36 Set Variable PIEZA_KOP = PIEZA_KOP+1  
37 If PIEZA_KOP == PIEZA_ESK Jump to MATLAB_HASI  
38 Jump to HASIERA
```

---

```
39 POSIZIO2:  
40 Go to Position 9 Speed 75 (%)  
41 Go to Position 10 Speed 75 (%)  
42 Close Gripper  
43 Go to Position 1 Speed 75 (%)  
44 Go to Position 3 Speed 75 (%)  
45 Go to Position 4 Speed 75 (%)  
46 Open Gripper  
47 Go to Position 3 Speed 75 (%)  
48 Go to Position 1 Speed 75 (%)  
49 Go to Position 9 Speed 75 (%)  
50 Set Variable PIEZA_KOP = PIEZA_KOP+1  
51 If PIEZA_KOP == PIEZA_ESK Jump to MATLAB_HASI  
52 Jump to HASIERA  
53 POSIZIO1:  
54 Go to Position 9 Speed 75 (%)  
55 Go to Position 10 Speed 75 (%)  
56 Close Gripper  
57 Go to Position 1 Speed 75 (%)  
58 Go to Position 2 Speed 75 (%)  
59 Open Gripper  
60 Go to Position 1 Speed 75 (%)  
61 Go to Position 9 Speed 75 (%)  
62 Set Variable PIEZA_KOP = PIEZA_KOP+1  
63 If PIEZA_KOP == PIEZA_ESK Jump to MATLAB_HASI  
64 Jump to HASIERA
```



```
65 POSIZIO3:  
66 Go to Position 9 Speed 75 (%)  
67 Go to Position 10 Speed 75 (%)  
68 Close Gripper  
69 Go to Position 1 Speed 75 (%)  
70 Go to Position 5 Speed 75 (%)  
71 Go to Position 6 Speed 75 (%)  
72 Open Gripper  
73 Go to Position 5 Speed 75 (%)  
74 Go to Position 1 Speed 75 (%)  
75 Go to Position 9 Speed 75 (%)  
76 Set Variable PIEZA_KOP = PIEZA_KOP+1  
77 If PIEZA_KOP == PIEZA_ESK Jump to MATLAB_HASI  
78 Jump to HASIERA
```

Eskatutako pieza denak kokatuta daudenean argazkia aterako du aplikazioak eta erabiltzaileak sailkatu nahi dituen piezak adieraziko ditu. Orduan robotari informazioa iritsiko zaio eta sailkatzen hasiko da. Sailkapena piezak daukan kolorearen eta formaren arabera egingo du:

```
78 Jump to HASIERA  
79 MATLAB_HASI:  
80 Set Variable PIEZA_HARTU = 0  
81 Set Variable GORRI = 0  
82 Set Variable URDIN = 0  
83 Set Variable BERDE = 0  
84 Reset Timer  
85 MATLAB:  
86 If Input 1 On Call Subroutine HARTU_POSIZIO1  
87 If Input 2 On Call Subroutine HARTU_POSIZIO2  
88 If Input 3 On Call Subroutine HARTU_POSIZIO3  
89 If Input 4 On Call Subroutine HARTU_POSIZIO4  
90 Set Variable AMAITU to Timer  
91 If AMAITU > 250 Jump to AMAITU  
92 Jump to MATLAB  
93 AMAITU:  
94 Stop Conveyor Axis 7  
95 End
```

---

105 Set Subroutine HARTU\_POSIZIO1  
106 Go to Position 1 Speed 75 (%)  
107 Go to Position 2 Speed 75 (%)  
108 Close Gripper  
109 Go to Position 1 Speed 75 (%)  
110 If Input 5 On Call Subroutine GORRIA  
111 If Input 6 On Call Subroutine URDINA  
112 If Input 7 On Call Subroutine BERDEA  
113 Set Variable PIEZA\_HARTU = PIEZA\_HARTU+1  
114 Turn On Output 1  
115 Reset Timer  
116 Wait 20 (10ths of seconds)  
117 Return from Subroutine  
118 Set Subroutine HARTU\_POSIZIO2  
119 Go to Position 1 Speed 75 (%)  
120 Go to Position 3 Speed 75 (%)  
121 Go to Position 4 Speed 75 (%)  
122 Close Gripper  
123 Go to Position 3 Speed 75 (%)  
124 Go to Position 1 Speed 75 (%)  
125 If Input 5 On Call Subroutine GORRIA  
126 If Input 6 On Call Subroutine URDINA  
127 If Input 7 On Call Subroutine BERDEA  
128 Set Variable PIEZA\_HARTU = PIEZA\_HARTU+1  
129 Turn On Output 2  
130 Reset Timer  
131 Wait 20 (10ths of seconds)  
132 Return from Subroutine

133 Set Subroutine HARTU\_POSIZIO3  
134 Go to Position 1 Speed 75 (%)  
135 Go to Position 5 Speed 75 (%)  
136 Go to Position 6 Speed 75 (%)  
137 Close Gripper  
138 Go to Position 5 Speed 75 (%)  
139 Go to Position 1 Speed 75 (%)  
140 If Input 5 On Call Subroutine GORRIA  
141 If Input 6 On Call Subroutine URDINA  
142 If Input 7 On Call Subroutine BERDEA  
143 Set Variable PIEZA\_HARTU = PIEZA\_HARTU+1  
144 Turn On Output 3  
145 Reset Timer  
146 Wait 20 (10ths of seconds)  
147 Return from Subroutine  
148 Set Subroutine HARTU\_POSIZIO4  
149 Go to Position 1 Speed 75 (%)  
150 Go to Position 7 Speed 75 (%)  
151 Go to Position 8 Speed 75 (%)  
152 Close Gripper  
153 Go to Position 7 Speed 75 (%)  
154 Go to Position 1 Speed 75 (%)  
155 If Input 5 On Call Subroutine GORRIA  
156 If Input 6 On Call Subroutine URDINA  
157 If Input 7 On Call Subroutine BERDEA  
158 Set Variable PIEZA\_HARTU = PIEZA\_HARTU+1  
159 Turn On Output 4  
160 Reset Timer  
161 Wait 20 (10ths of seconds)  
162 Return from Subroutine

163 Set Subroutine GORRIA  
164 Set Variable GORRI = GORRI+1  
165 Go to Position 14 Speed 99 (%)  
166 Go to Position 11 Speed 75 (%)  
167 Open Gripper  
168 Return from Subroutine

---

169 Set Subroutine URDINA  
170 Set Variable URDIN = URDIN+1  
171 Go to Position 14 Speed 99 (%)  
172 Go to Position 12 Speed 75 (%)  
173 Open Gripper  
174 Return from Subroutine  
175 Set Subroutine BERDEA  
176 Set Variable BERDE = BERDE+1  
177 Go to Position 14 Speed 99 (%)  
178 Go to Position 13 Speed 75 (%)  
179 Open Gripper  
180 Return from Subroutine