

---

## Table of Contents

.....	1
-- L'execució quan es polsa el botó reset. Primer borra tota la .....	3
-- L'execució quan polses el botó quit. Crida a la funció salir que li .....	3
-- L'execució quan polses comprova que hi ha alguna cosa executant en el .....	4
-- L'execució quan polses Start, activa tots els botons de control i .....	4

```
function varargout = guigraf(varargin)
% GUIGRAF MATLAB code for guigraf.fig
%     GUIGRAF, by itself, creates a new GUIGRAF or raises the
%     existing
%     singleton*.
%
%     H = GUIGRAF returns the handle to a new GUIGRAF or the handle
%     to
%     the existing singleton*.
%
%     GUIGRAF('CALLBACK',hObject,eventData,handles,...) calls the
%     local
%     function named CALLBACK in GUIGRAF.M with the given input
%     arguments.
%
%     GUIGRAF('Property','Value',...) creates a new GUIGRAF or raises
%     the
%     existing singleton*. Starting from the left, property value
%     pairs are
%     applied to the GUI before guigraf_OpeningFcn gets called. An
%     unrecognized property name or invalid value makes property
%     application
%     stop. All inputs are passed to guigraf_OpeningFcn via
%     varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
%     only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% editor the above text to modify the response to help guigraf

% Last Modified by GUIDE v2.5 29-Aug-2016 08:20:32

% Begin initialization code - DO NOT EDITOR
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
    'gui_Singleton',  gui_Singleton, ...
    'gui_OpeningFcn', @guigraf_OpeningFcn, ...
    'gui_OutputFcn',  @guigraf_OutputFcn, ...
    'gui_LayoutFcn',  [] , ...
```

---

```

        'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDITOR

% --- Executes just before guigrafi is made visible.
function guigrafi_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin   command line arguments to guigrafi (see VARARGIN)

% Choose default command line output for guigrafi
%%Inicialitzem les Variables per a la interfície gràfica.
handles.output = hObject;
handles.start=false;
handles.carga=false;
handles.oldstate= pause('on');
set(handles.stop, 'Enable', 'off');
set(handles.pause, 'Enable', 'off');
set(handles.reset, 'Enable', 'off');
set(handles.kilobot, 'Enable', 'off');
set(handles.time, 'Enable', 'off');
set(handles.informa, 'Enable', 'off');
% Refresquem la estructura de dades.
guidata(hObject, handles);

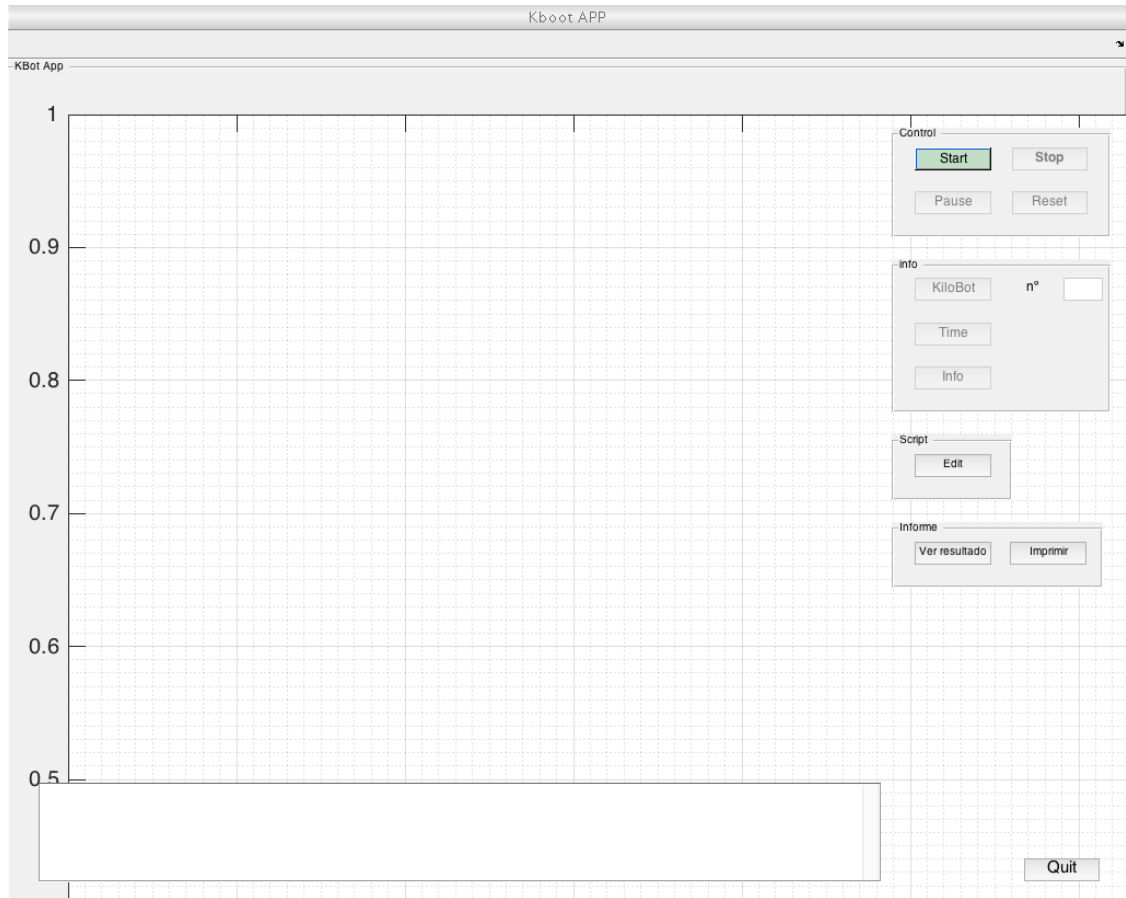
% --- Outputs from this function are returned to the command line.
function varargout = guigrafi_OutputFcn(hObject, eventdata, handles)

% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

```

---



## --- L'execució quan es polsa el botó reset. Primer borra tota la

informació que hi ha a axes1 (o siga a l'escenari de simulació). Després actualitza l'estructura de dades. I seguidament crida al Callback Start.

```
function reset_Callback(hObject, eventdata, handles)

cla(handles.axes1);
handles.start=false;
guidata(hObject,handles);
start_Callback(hObject, eventdata, handles);
```

## --- L'execució quan polses el botó quit. Crida a la funció salir que li

retorna un true o false, depenent si prems sí o no a la pregunta de vol tancar el programa?, en el cas que si, quit és true, aleshores és neteja tot el handles (la classe principal) i es tanca tot.

```
function quit_Callback(hObject, eventdata, handles)

quit=salir;
```

---

```
if quit == true
    clear all
    close all;
end
```

## --- L'execució quan polses comprova que hi ha alguna cosa executant en el

simulador, després crida a la funció paro, que pregunta si realment vol parar l'execució de la simulació, en el cas de ser true, neteja el escenari de simulació, actualitza la variable de simulació activa a false, actualitza l'estructura de dades i desactiva tots els botons de control. Mitjançant la consola de comandos, avisa que la simulació ha sigut avortada.

```
function stop_Callback(hObject, eventdata, handles)

if handles.start==true
    stoped=paro;
    if stoped == true
        oldtext=get(handles.consola, 'String');
        text='Simulacion abortada';
        set(handles.consola, 'String', [oldtext; {text}]);
        %close (handles.axes1);
        cla
        %clear handles.axes1;
        handles.start=false;
        guidata(hObject, handles);
        set(handles.stop, 'Enable', 'off');
        set(handles.pause, 'Enable', 'off');
        set(handles.reset, 'Enable', 'off');
    end
else

    guidata(hObject, handles);
end
```

## --- L'execució quan polses Start, activa tots els botons de control i

neteja tota l'àrea de dades dels kilobots. Comprova que no està executant-se res i posa la variable start a true. Mostra per consola que la simulació ha començat e inicia el comptador de temps, executa el script de simulació, i en acabar, para el rellotge. Activa els botons d'informació, ja que la informació no està disponible fins que no s'acabe d'executar el script (coses de matlab), torna la variable start a false, mostra en la consola que la simulació ha sigut finalitzada recarrega les dades. I desactiva els botons de control.

```
function start_Callback(hObject, eventdata, handles)
set(handles.stop, 'Enable', 'on');
set(handles.pause, 'Enable', 'on');
set(handles.reset, 'Enable', 'on');

cla
clear handles.kilobot;
```

---

```

    text1=sprintf('Comienza simulación');
    if handles.start == false
        handles.start=true;
        set(handles.consola,'String',text1);
        guidata(hObject, handles);
        tic;
        handles.kbot=program;
        handles.t=toc;
        set(handles.kilobot,'Enable','on');
        set(handles.time,'Enable','on');
        set(handles.informa,'Enable','on');
        handles.start=false;
        text2=sprintf('Simulacion Finalizada');
        guidata(hObject, handles);
        set(handles.stop,'Enable','off');
        set(handles.pause,'Enable','off');
        set(handles.reset,'Enable','off');
        set(handles.consola,'String',[text1;{text2}]);
    end

% --- Executes on button press in selectfile.
function selectfile_Callback(hObject, eventdata, handles)
% hObject      handle to selectfile (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
clear handles.data;
%[handles.script]=uigetfile({'*.m'},'Select file','.m');
[ handles]= uigetfile({'*.m'; '*.txt'},'Select
    file','program_carga.m');
handles.carga= true;
guidata(hObject, handles);

% --- Executes on button press in pause.
function pause_Callback(hObject, eventdata, handles)
% hObject      handle to pause (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
if handles.start == true
    handles.oldstate
    pause(handles.oldstate)
    handles.pause = pausa;
    pause(handles.oldstate);
end
guidata(hObject, handles);

% --- Executes on button press in kilobot.
function kilobot_Callback(hObject, eventdata, handles)
% hObject      handle to kilobot (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
oldText=get(handles.consola,'String');
text= ['Kilobot numero ' num2str(handles.kbot_num) ' finalizo en: '
    num2str(handles.kbot(handles.kbot_num).iter/60) ' minutos'];

```

---

---

```

set(handles.consola, 'String', [oldText; {text}]);

% --- Executes on button press in time.
function time_Callback(hObject, eventdata, handles)
% hObject    handle to time (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
handles.totaltime = max([handles.kbot.iter])/60;
oldText=get(handles.consola, 'String');
text=sprintf(['Tiempo real: ' num2str(handles.totaltime) ' minutos']);
set(handles.consola, 'String', [oldText; {text}]);
guidata(hObject, handles);
% --- Executes on button press in informa.
function informa_Callback(hObject, eventdata, handles)
% hObject    handle to informa (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
oldtext=get(handles.consola, 'String');
text=['Tiempo de maquina: ' num2str(handles.t) ' segundos'];
set(handles.consola, 'String', [oldtext; {text}]);
% -----
% -----
function editor_Callback(hObject, eventdata, handles)
% hObject    handle to editor (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
if handles.carga == true
    edit(handles.ans);
else
    edit program;
end
% -----

% --- Executes during object creation, after setting all properties.
function consola_CreateFcn(hObject, eventdata, handles)
% hObject    handle to consola (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
    called

% --- Executes on button press in result.
function result_Callback(hObject, eventdata, handles)
% hObject    handle to result (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
resultado;

% --- Executes on slider movement.
function consola_Callback(hObject, eventdata, handles)

```

---

---

```

% hObject      handle to consola (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'Value') returns position of slider
%         get(hObject,'Min') and get(hObject,'Max') to determine range
%         of slider

function edit2_Callback(hObject, eventdata, handles)
% hObject      handle to edit2 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit2 as text
%         str2double(get(hObject,'String')) returns contents of edit2
%         as a double
handles.kbot_num=str2double(get(hObject,'String'));
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit2 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
%              called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
    get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% -----
function nuevo_Callback(hObject, eventdata, handles)
% hObject      handle to nuevo (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% --- Executes on button press in imprime.
function imprime_Callback(hObject, eventdata, handles)
% hObject      handle to imprime (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
time_cpu=num2str(handles.t);
maxim=num2str(max([handles.kbot.iter]));
time_max= num2str(max([handles.kbot.iter])/60);
minim= num2str(min([handles.kbot.iter]));
time_min= num2str(min([handles.kbot.iter])/60);

```

---

---

```
fid=fopen('archivo.txt','w');

fprintf(fid,'Tiempos CPU: %s\n ',time_cpu);
fprintf(fid,'Tiempo simulacion(tiempo real) %s minutos\n',time_max);
fprintf(fid,'Tiempo maximo: %s minutos\n',time_max);
fprintf(fid,'Tiempo minimo: %s minutos\n',time_min);
fprintf(fid,'Iteracion maxima: %s\n',maxim);
fprintf(fid,'Iteracion minima: %s\n',minim);
fclose(fid);
open archivo.txt;
```

*Published with MATLAB® R2015b*