
```

function varargout = Question1(varargin)
% QUESTION1 MATLAB code for Question1.fig
%     QUESTION1, by itself, creates a new QUESTION1 or raises the
%     existing
%     singleton*.
%
%     H = QUESTION1 returns the handle to a new QUESTION1 or the
%     handle to
%     the existing singleton*.
%
%     QUESTION1('CALLBACK',hObject,eventData,handles,...) calls the
%     local
%     function named CALLBACK in QUESTION1.M with the given input
%     arguments.
%
%     QUESTION1('Property','Value',...) creates a new QUESTION1 or
%     raises the
%     existing singleton*. Starting from the left, property value
%     pairs are
%     applied to the GUI before Question1_OpeningFcn gets called. An
%     unrecognized property name or invalid value makes property
%     application
%     stop. All inputs are passed to Question1_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

% Edit the above text to modify the response to help Question1

% Last Modified by GUIDE v2.5 31-Oct-2019 16:37:19

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @Question1_OpeningFcn, ...
                  'gui_OutputFcn',  @Question1_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 EDIT

% --- Executes just before Question1 is made visible.
function Question1_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 Question1 (see VARARGIN)

% Choose default command line output for Question1
handles.output = hObject;
% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Question1 wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = Question1_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;

% --- Executes on button press in Inpluse_button.
function Inpluse_button_Callback(hObject, eventdata, handles)
% hObject    handle to Inpluse_button (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% Set slide visible and text string.
    set(handles.Inpluse_posx, 'visible', 'on');
    set(handles.Inpluse_posy, 'visible', 'on');
    set(handles.Sin_angle, 'visible', 'off');
    set(handles.Sin_freq, 'visible', 'off');
    set(handles.Sin_phrase, 'visible', 'off');
    set(handles.Rect_angle, 'visible', 'off');
    set(handles.Rect_whrate, 'visible', 'off');
    set(handles.Rect_posx, 'visible', 'off');
    set(handles.Rect_posy, 'visible', 'off');
    set(handles.Rect_size, 'visible', 'off');
    set(handles.Gauss_var, 'visible', 'off');
    set(handles.text1, 'String', 'x#'+
string(floor(get(handles.Inpluse_posx, 'value') * 10 + 1)));
    set(handles.text2, 'String', 'y#'+
string(floor(get(handles.Inpluse_posy, 'value') * 10 + 1)));
    set(handles.text3, 'String', '');
    set(handles.text4, 'String', '');

```

```

        set(handles.text5, 'String', '');
% Display image
        displayInpluse(handles, hObject);

function displayInpluse(handles, hObject)
    I = zeros(256,256);
    x = floor(get(handles.Inpluse_posx, 'value') * 10 + 1);
    y = floor(get(handles.Inpluse_posy, 'value') * 10 + 1);
    I(x, y) = 1;
    axes(handles.axes1);imshow(I);
    I1 = DFT2D(I);
    axes(handles.axes2);imshow(log(abs(I1)),[]);
    axes(handles.axes3);imshow(angle(I1));
    step = 4;
    axes(handles.axes4);surf(I(1:1:end,1:1:end));
    axes(handles.axes5);surf(log(abs(I1(1:step:end,1:step:end)))));
    axes(handles.axes6);surf(angle(I1(1:step:end,1:step:end)))));

% --- Executes on button press in Sin_Button.
function Sin_Button_Callback(hObject, eventdata, handles)
% hObject      handle to Sin_Button (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
    set(handles.Inpluse_posx, 'visible', 'off');
    set(handles.Inpluse_posy, 'visible', 'off');
    set(handles.Sin_angle, 'visible', 'on');
    set(handles.Sin_freq, 'visible', 'on');
    set(handles.Sin_phrase, 'visible', 'on');
    set(handles.Rect_angle, 'visible', 'off');
    set(handles.Rect_whrate, 'visible', 'off');
    set(handles.Rect_posx, 'visible', 'off');
    set(handles.Rect_posy, 'visible', 'off');
    set(handles.Rect_size, 'visible', 'off');
    set(handles.Gauss_var, 'visible', 'off');
    set(handles.text1, 'String', '');
    set(handles.text2, 'String', '');
    set(handles.text3, 'String', '#
##'+string(get(handles.Sin_angle, 'Value')));
    set(handles.text4, 'String', '#
##'+string(get(handles.Sin_freq, 'Value') / 1000));
    set(handles.text5, 'String', '#
##'+string(get(handles.Sin_phrase, 'Value')));
    % Display image
    displaySin(handles, hObject);

function displaySin(handles, hObject)
    [X, Y] = meshgrid(1:256);
    a = get(handles.Sin_angle, 'Value') * pi / 180;
    freq = get(handles.Sin_freq, 'Value') / 1000;
    phrase = get(handles.Sin_phrase, 'Value');
    I = cos(2 * pi * freq * (cos(a) * X + sin(a) * Y) + phrase);
    axes(handles.axes1);imshow(I);
    I1 = DFT2D(I);
    axes(handles.axes2);imshow(log(abs(I1)),[-1,5]);

```

```

axes(handles.axes3);imshow(angle(I1));
step = 4;
axes(handles.axes4);surf(I(1:step:end,1:step:end));
axes(handles.axes5);surf(log(abs(I1(1:step:end,1:step:end))));
axes(handles.axes6);surf(angle(I1(1:step:end,1:step:end)));

% --- Executes on button press in Rect_Button.
function Rect_Button_Callback(hObject, eventdata, handles)
% hObject    handle to Rect_Button (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
    set(handles.Inpluse_posx, 'visible', 'off');
    set(handles.Inpluse_posy, 'visible', 'off');
    set(handles.Sin_angle, 'visible', 'off');
    set(handles.Sin_freq, 'visible', 'off');
    set(handles.Sin_phrase, 'visible', 'off');
    set(handles.Rect_angle, 'visible', 'on');
    set(handles.Rect_whrate, 'visible', 'on');
    set(handles.Rect_posx, 'visible', 'on');
    set(handles.Rect_posy, 'visible', 'on');
    set(handles.Rect_size, 'visible', 'on');
    set(handles.Gauss_var, 'visible', 'off');
    set(handles.text1, 'String', 'x##
##'+string(floor(get(handles.Rect_posx, 'value') * 10)));
    set(handles.text2, 'String', 'y##
##'+string(floor(get(handles.Rect_posy, 'value') * 10)));
    set(handles.text3, 'String', '#
##'+string(get(handles.Rect_angle, 'Value')));
    set(handles.text4, 'String', '##
##'+string(get(handles.Rect_whrate, 'Value')));
    set(handles.text5, 'String', '####
##'+string(floor(get(handles.Rect_size, 'Value'))));
    % Display image
    displayRect(handles, hObject);

function displayRect(handles, hObject)
    center_x = 128 + floor(get(handles.Rect_posx, 'value') * 10);
    center_y = 128 + floor(get(handles.Rect_posy, 'value') * 10);
    a = get(handles.Rect_angle, 'Value');
    whrate = get(handles.Rect_whrate, 'Value');
    w = floor(get(handles.Rect_size, 'Value'));
    h = w / whrate;
    I = zeros(256,256);
    I(max(floor(center_x - h / 2), 1):min(floor(center_x + h / 2),
256), max(floor(center_y - w / 2), 1):min(floor(center_y + w / 2),
256)) = 1;
    I = imrotate(I, a, 'bicubic', 'crop');
    axes(handles.axes1);imshow(I);
    I1 = DFT2D(I);
    axes(handles.axes2);imshow(log(abs(I1)),[-1,5]);
    axes(handles.axes3);imshow(angle(I1));
    step = 4;
    axes(handles.axes4);surf(I(1:4:end,1:4:end));

```

```

        axes(handles.axes5);surf(log(abs(I1(1:step:end,1:step:end))));
        axes(handles.axes6);surf(angle(I1(1:step:end,1:step:end)));

% --- Executes on button press in Gauss_Button.
function Gauss_Button_Callback(hObject, eventdata, handles)
% hObject      handle to Gauss_Button (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see
GUIDATA)set(handles.Inpluse_posx, 'visible', 'off');
    set(handles.Inpluse_posx, 'visible', 'off');
    set(handles.Inpluse_posy, 'visible', 'off');
    set(handles.Sin_angle, 'visible', 'off');
    set(handles.Sin_freq, 'visible', 'off');
    set(handles.Sin_phrase, 'visible', 'off');
    set(handles.Rect_angle, 'visible', 'off');
    set(handles.Rect_whrate, 'visible', 'off');
    set(handles.Rect_posx, 'visible', 'off');
    set(handles.Rect_posy, 'visible', 'off');
    set(handles.Rect_size, 'visible', 'off');
    set(handles.Gauss_var, 'visible', 'on');
    set(handles.text1, 'String', '');
    set(handles.text2, 'String', '');
    set(handles.text3, 'String', '#
##'+string(get(handles.Gauss_var, 'Value')));
    set(handles.text4, 'String', '');
    set(handles.text5, 'String', '');
    % Display image
    displayGauss(handles, hObject);

function displayGauss(handles, hObject)
[X, Y] = meshgrid(-128:1:127);
sigma = get(handles.Gauss_var, 'Value');
I = exp(-(X .* X + Y .* Y) / (2 * sigma * sigma));
axes(handles.axes1);imshow(I);
I1 = DFT2D(I);
axes(handles.axes2);imshow(log(abs(I1)),[-1,5]);
axes(handles.axes3);imshow(angle(I1));
step = 4;
axes(handles.axes4);surf(I(1:4:end,1:4:end));
axes(handles.axes5);surf(log(abs(I1(1:step:end,1:step:end))));
axes(handles.axes6);surf(angle(I1(1:step:end,1:step:end)));

% --- Executes on slider movement.
function Inpluse_posx_Callback(hObject, eventdata, handles)
% hObject      handle to Inpluse_posx (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
    displayInpluse(handles, hObject);
    set(handles.text1, 'String', 'x#'+
string(floor(get(handles.Inpluse_posx, 'value') * 10 + 1)));

```

```

        set(handles.text2, 'String', 'y#'+
string(floor(get(handles.Inpluse_posy, 'value') * 10 + 1)));

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

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'),
    get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

% --- Executes on slider movement.
function slider2_Callback(hObject, eventdata, handles)
% hObject    handle to slider2 (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

% --- Executes during object creation, after setting all properties.
function slider2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to slider2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
            called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'),
    get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

% --- Executes on slider movement.
function Inpluse_posy_Callback(hObject, eventdata, handles)
% hObject    handle to Inpluse_posy (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
    displayInpluse(handles, hObject);
    set(handles.text1, 'String', 'x#'+
string(floor(get(handles.Inpluse_posx, 'value') * 10 + 1)));

```

```

        set(handles.text2, 'String', 'y#'+
string(floor(get(handles.Inpluse_posy, 'value') * 10 + 1)));

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

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'),
    get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

% --- Executes on slider movement.
function Sin_angle_Callback(hObject, eventdata, handles)
% hObject    handle to Sin_angle (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
    displaySin(handles, hObject);
    set(handles.text3, 'String', '#
##'+string(get(handles.Sin_angle, 'Value')));
    set(handles.text4, 'String', '#
##'+string(get(handles.Sin_freq, 'Value') / 1000));
    set(handles.text5, 'String', '#
##'+string(get(handles.Sin_phrase, 'Value')));

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

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'),
    get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

% --- Executes on slider movement.
function Sin_freq_Callback(hObject, eventdata, handles)
% hObject    handle to Sin_freq (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
    displaySin(handles, hObject);
    set(handles.text3, 'String', '#
##'+string(get(handles.Sin_angle, 'Value')));
    set(handles.text4, 'String', '#
##'+string(get(handles.Sin_freq, 'Value') / 1000));
    set(handles.text5, 'String', '#
##'+string(get(handles.Sin_phrase, 'Value')));

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

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'),
    get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

% --- Executes on slider movement.
function Sin_phrase_Callback(hObject, eventdata, handles)
% hObject    handle to Sin_phrase (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
    displaySin(handles, hObject);
    set(handles.text3, 'String', '#
##'+string(get(handles.Sin_angle, 'Value')));
    set(handles.text4, 'String', '#
##'+string(get(handles.Sin_freq, 'Value') / 1000));
    set(handles.text5, 'String', '#
##'+string(get(handles.Sin_phrase, 'Value')));

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

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'),
    get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);

```

```

end

% --- Executes on slider movement.
function Rect_posx_Callback(hObject, eventdata, handles)
% hObject      handle to Rect_posx (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
    set(handles.text1, 'String', 'x##
##'+string(floor(get(handles.Rect_posx, 'value') * 10)));
    set(handles.text2, 'String', 'y##
##'+string(floor(get(handles.Rect_posy, 'value') * 10)));
    set(handles.text3, 'String', '#
##'+string(get(handles.Rect_angle, 'Value')));
    set(handles.text4, 'String', '##
##'+string(get(handles.Rect_whrate, 'Value')));
    set(handles.text5, 'String', '####
##'+string(floor(get(handles.Rect_size, 'Value'))));
    % Display image
    displayRect(handles, hObject);

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

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'),
    get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

% --- Executes on slider movement.
function Rect_posy_Callback(hObject, eventdata, handles)
% hObject      handle to Rect_posy (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
    set(handles.text1, 'String', 'x##
##'+string(floor(get(handles.Rect_posx, 'value') * 10)));
    set(handles.text2, 'String', 'y##
##'+string(floor(get(handles.Rect_posy, 'value') * 10)));
    set(handles.text3, 'String', '#
##'+string(get(handles.Rect_angle, 'Value')));
    set(handles.text4, 'String', '##
##'+string(get(handles.Rect_whrate, 'Value')));

```

```

        set(handles.text5, 'String', '####
##'+string(floor(get(handles.Rect_size, 'Value'))));
        % Display image
        displayRect(handles, hObject);

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

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'),
    get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

% --- Executes on slider movement.
function Rect_angle_Callback(hObject, eventdata, handles)
% hObject    handle to Rect_angle (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
        set(handles.text1, 'String', 'x##
##'+string(floor(get(handles.Rect_posx, 'value') * 10)));
        set(handles.text2, 'String', 'y##
##'+string(floor(get(handles.Rect_posy, 'value') * 10)));
        set(handles.text3, 'String', '#
##'+string(get(handles.Rect_angle, 'Value')));
        set(handles.text4, 'String', '##
##'+string(get(handles.Rect_whrate, 'Value')));
        set(handles.text5, 'String', '####
##'+string(floor(get(handles.Rect_size, 'Value'))));
        % Display image
        displayRect(handles, hObject);

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

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'),
    get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

```

```

% --- Executes on slider movement.
function Rect_whrate_Callback(hObject, eventdata, handles)
% hObject      handle to Rect_whrate (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
    set(handles.text1, 'String', 'x##
##'+string(floor(get(handles.Rect_posx, 'value') * 10)));
    set(handles.text2, 'String', 'y##
##'+string(floor(get(handles.Rect_posy, 'value') * 10)));
    set(handles.text3, 'String', '#
##'+string(get(handles.Rect_angle, 'Value')));
    set(handles.text4, 'String', '##
##'+string(get(handles.Rect_whrate, 'Value')));
    set(handles.text5, 'String', '####
##'+string(floor(get(handles.Rect_size, 'Value'))));
    % Display image
    displayRect(handles, hObject);

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

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'),
    get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

% --- Executes on slider movement.
function Rect_size_Callback(hObject, eventdata, handles)
% hObject      handle to Rect_size (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
    set(handles.text1, 'String', 'x##
##'+string(floor(get(handles.Rect_posx, 'value') * 10)));
    set(handles.text2, 'String', 'y##
##'+string(floor(get(handles.Rect_posy, 'value') * 10)));
    set(handles.text3, 'String', '#
##'+string(get(handles.Rect_angle, 'Value')));
    set(handles.text4, 'String', '##
##'+string(get(handles.Rect_whrate, 'Value')));

```

```

        set(handles.text5, 'String', '####
##'+string(floor(get(handles.Rect_size, 'Value'))));
        % Display image
        displayRect(handles, hObject);

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

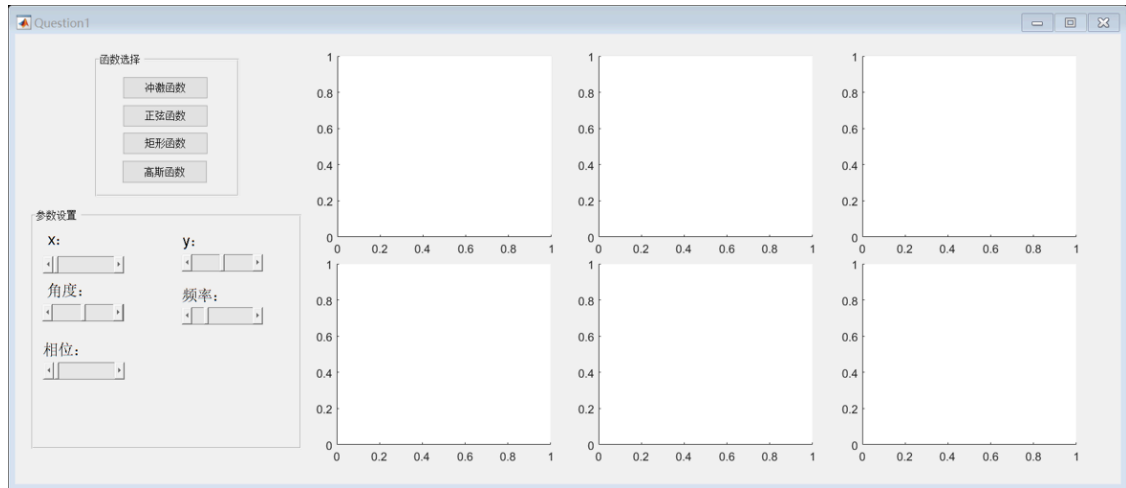
% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'),
    get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

% --- Executes on slider movement.
function Gauss_var_Callback(hObject, eventdata, handles)
% hObject    handle to Gauss_var (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
        set(handles.text1, 'String', '');
        set(handles.text2, 'String', '');
        set(handles.text3, 'String', '#
##'+string(get(handles.Gauss_var, 'Value')));
        set(handles.text4, 'String', '');
        set(handles.text5, 'String', '');
        % Display image
        displayGauss(handles, hObject);

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

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'),
    get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

```



Published with MATLAB® R2018b