Sound Mixer and Equilizer

* *Youssef Mohamed Alshaarawi 5329*
* *Mostafa Ehab El-Zawawy 5311*
* *Kyrellos Naeem 5566*
* *Adham Mohamed Al-Sawaf 5355*
* *Mohamed Wahba 5648*

Summary:

*This program is made to fulfill one goal which is: Mixing between two or more audio tracks and also control the frequency of each track, it also can be used as an audio player which can play, pause, resume, plotting the sound graph while playing and control the volume of each track individually or together in a good graphical user interface (GUI).*

*Code:*

function varargout = untitled(varargin)

% UNTITLED MATLAB code for untitled.fig

% UNTITLED, by itself, creates a new UNTITLED or raises the existing

% singleton\*.

%

% H = UNTITLED returns the handle to a new UNTITLED or the handle to

% the existing singleton\*.

%

% UNTITLED('CALLBACK',hObject,eventData,handles,...) calls the local

% function named CALLBACK in UNTITLED.M with the given input arguments.

%

% UNTITLED('Property','Value',...) creates a new UNTITLED or raises the

% existing singleton\*. Starting from the left, property value pairs are

% applied to the GUI before untitled\_OpeningFcn gets called. An

% unrecognized property name or invalid value makes property application

% stop. All inputs are passed to untitled\_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 untitled

% Last Modified by GUIDE v2.5 12-May-2018 02:35:18

% Begin initialization code - DO NOT EDIT

gui\_Singleton = 1;

gui\_State = struct('gui\_Name', mfilename, ...

'gui\_Singleton', gui\_Singleton, ...

'gui\_OpeningFcn', @untitled\_OpeningFcn, ...

'gui\_OutputFcn', @untitled\_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 untitled is made visible.

function untitled\_OpeningFcn(hObject, ~, 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 untitled (see VARARGIN)

% Choose default command line output for untitled

handles.output = hObject;

% Update handles structure

guidata(hObject, handles);

% UIWAIT makes untitled wait for user response (see UIRESUME)

% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.

function varargout = untitled\_OutputFcn(~, ~, 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 slider movement.

function slider15\_Callback(hObject, ~, handles)

% hObject handle to slider15 (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

global y;

global Fs;

pause(handles.player);

newstart=get(handles.player,'CurrentSample')+1;

stop(handles.player);

y=y(newstart:end,:);

volume1=get(handles.slider15,'value');

handles.player=audioplayer(y\*volume1,Fs);

guidata(hObject,handles);

% save all data to the handles object

handles.y = y';

handles.Fs = Fs;

handles.timeSec = length(y)/Fs;

handles.atSample1 = 0;

% assign a periodic timer to fire during playback

set(handles.player,'TimerFcn',{@timerCallback5,handles.figure1}, 'TimerPeriod', 0.1);

% save the updated handles object

guidata(hObject,handles);

% reset the axes and set some properties

cla(handles.axes1);

hold(handles.axes1,'on');

xlim(handles.axes1,[1 length(y)]);

% play the music

play(handles.player);

function timerCallback5(hObject, ~, hFig)

% get the handles structure

handles = guidata(hFig);

% determine the current sample

currSample1 = get(hObject,'CurrentSample');

% get all of the sound data

data1 = handles.y(handles.atSample1+1:currSample1);

% plot the most recent set of data

plot(handles.axes1,handles.atSample1+1:currSample1,data1,'red');

% update the handles object

handles.atSample1 = currSample1;

guidata(hFig,handles);

% update the slider

if currSample1 > 1

sliderVal1 = min(1.0,currSample1/length(handles.y));

set(handles.slider1,'Value',sliderVal1);

end

% --- Executes during object creation, after setting all properties.

function slider15\_CreateFcn(hObject, ~, ~)

% hObject handle to slider15 (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 button press in play2.

function play2\_Callback(hObject, ~, handles1)

% hObject handle to play2 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

global z;

global Ts;

handles1.player1 = audioplayer(z, Ts);

% save all data to the handles object

handles1.z = z';

handles1.Ts = Ts;

handles1.timeSec = length(z)/Ts;

handles1.atSample = 0;

% assign a periodic timer to fire during playback

set(handles1.player1,'TimerFcn',{@timerCallback2,handles1.figure1}, 'TimerPeriod', 0.1);

% save the updated handles object

guidata(hObject,handles1);

% reset the axes and set some properties

cla(handles1.axes2);

hold(handles1.axes2,'on');

xlim(handles1.axes2,[1 length(z)]);

% play the music

play(handles1.player1);

function timerCallback2(hObject, ~, hFig)

% get the handles structure

handles1 = guidata(hFig);

% determine the current sample

currSample = get(hObject,'CurrentSample');

% get all of the sound data

data = handles1.z(handles1.atSample+1:currSample);

% plot the most recent set of data

plot(handles1.axes2,handles1.atSample+1:currSample,data,'green');

% update the handles object

handles1.atSample = currSample;

guidata(hFig,handles1);

% update the slider

if currSample > 1

sliderVal = min(1.0,currSample/length(handles1.y));

set(handles1.slider2,'Value',sliderVal);

end

% --- Executes on button press in Browse.

function Browse\_Callback(hObject, ~, handles)

% hObject handle to Browse (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

[filename, pathname] =uigetfile({'\*.mp3';'\*.wav'}, 'File Selector');

handles.fullpathname = strcat(pathname, filename);

set(handles.address, 'String', handles.fullpathname)

global y;

global Fs;

global player;

[y,Fs] = audioread(handles.fullpathname);

player = audioplayer(y, Fs);

guide(hObject, handles)

% --- Executes on button press in resume2.

function resume2\_Callback(hObject, eventdata, handles)

% hObject handle to resume2 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

resume(handles.player1);

% --- Executes on button press in play.

function play\_Callback(hObject, eventdata, handles)

% hObject handle to play2 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

global y;

global Fs;

handles.player = audioplayer(y, Fs);

% save all data to the handles object

handles.y = y';

handles.Fs = Fs;

handles.timeSec = length(y)/Fs;

handles.atSample1 = 0;

% assign a periodic timer to fire during playback

set(handles.player,'TimerFcn',{@timerCallback5,handles.figure1}, 'TimerPeriod', 0.1);

% save the updated handles object

guidata(hObject,handles);

% reset the axes and set some properties

cla(handles.axes1);

hold(handles.axes1,'on');

xlim(handles.axes1,[1 length(y)]);

% play the music

play(handles.player);

function timerCallback5(hObject, event, hFig)

% get the handles structure

handles = guidata(hFig);

% determine the current sample

currSample1 = get(hObject,'CurrentSample');

% get all of the sound data

data1 = handles.y(handles.atSample1+1:currSample1);

% plot the most recent set of data

plot(handles.axes1,handles.atSample1+1:currSample1,data1,'red');

% update the handles object

handles.atSample1 = currSample1;

guidata(hFig,handles);

% update the slider

if currSample1 > 1

sliderVal1 = min(1.0,currSample1/length(handles.y));

set(handles.slider1,'Value',sliderVal1);

end

% --- Executes on slider movement.

function volume2\_Callback(hObject, eventdata, handles)

% hObject handle to volume2 (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

global z;

global Ts;

pause(handles.player1);

newstart=get(handles.player1,'CurrentSample')+1;

stop(handles.player1);

z=z(newstart:end,:);

volume=get(handles.volume2,'value');

handles.player1=audioplayer(z\*volume,Ts);

play(handles.player1);

guidata(hObject,handles);

% --- Executes on button press in Browse2.

function Browse2\_Callback(hObject, eventdata, handles)

% hObject handle to Browse2 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

[filename pathname] =uigetfile({'\*.mp3';'\*.wav'}, 'File Selector');

handles.fullpathname = strcat(pathname, filename);

set(handles.address2, 'String', handles.fullpathname)

global z;

global Ts;

global p;

[z,Ts] = audioread(handles.fullpathname);

p = audioplayer(z, Ts);

guide(hObject, handles)

% --- Executes on button press in pause2.

function pause2\_Callback(hObject, eventdata, handles)

% hObject handle to pause2 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

pause(handles.player1)

% --- 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)

pause(handles.player);

% --- Executes on button press in resume.

function resume\_Callback(hObject, eventdata, handles)

% hObject handle to resume (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

resume(handles.player);

% --- Executes on button press in play3.

function play3\_Callback(hObject, eventdata, handles5)

% hObject handle to play3 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

global x;

global Ys;

handles5.player5 = audioplayer(x, Ys);

% save all data to the handles object

handles5.x = x';

handles5.Ys = Ys;

handles5.timeSec = length(x)/Ys;

handles5.atSample5 = 0;

% assign a periodic timer to fire during playback

set(handles5.player5,'TimerFcn',{@timerCallback,handles5.figure1}, 'TimerPeriod', 0.1);

% save the updated handles object

guidata(hObject,handles5);

% reset the axes and set some properties

cla(handles5.axes3);

hold(handles5.axes3,'on');

xlim(handles5.axes3,[1 length(x)]);

% play the music

play(handles5.player5);

function timerCallback(hObject, event, hFig)

% get the handles structure

handles5 = guidata(hFig);

% determine the current sample

currSample5 = get(hObject,'CurrentSample');

% get all of the sound data

data5 = handles5.y(handles5.atSample5+1:currSample5);

% plot the most recent set of data

plot(handles5.axes3,handles5.atSample5+1:currSample5,data5);

% update the handles object

handles5.atSample5 = currSample5;

guidata(hFig,handles5);

% update the slider

if currSample5 > 1

sliderVal5 = min(1.0,currSample5/length(handles5.y));

set(handles5.slider3,'Value',sliderVal5);

end

% --- Executes on slider movement.

function slider67\_Callback(hObject, eventdata, handles)

% hObject handle to slider67 (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

global x;

global Ys;

pause(handles.player5);

newstart=get(handles.player5,'CurrentSample')+1;

stop(handles.player5);

x=x(newstart:end,:);

volume2=get(handles.slider67,'value');

handles.player5=audioplayer(x\*volume2,Ys);

play(handles.player5);

guidata(hObject,handles);

% --- Executes on button press in browse3.

function browse3\_Callback(hObject, eventdata, handles)

% hObject handle to browse3 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

[filename pathname] =uigetfile({'\*.mp3';'\*.wav'}, 'File Selector');

handles.fullpathname = strcat(pathname, filename);

set(handles.address3, 'String', handles.fullpathname)

global x;

global Ys;

global p1;

[x,Ys] = audioread(handles.fullpathname);

p1 = audioplayer(x, Ys);

guide(hObject, handles)

% --- Executes on button press in pause3.

function pause3\_Callback(hObject, eventdata, handles)

% hObject handle to pause3 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

pause(handles.player5);

% --- Executes on button press in resume3.

function resume3\_Callback(hObject, eventdata, handles)

% hObject handle to resume3 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

resume(handles.player5);

% --- Executes on button press in play\_all.

function play\_all\_Callback(hObject, eventdata, handles)

% hObject handle to play\_all (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

play(handles.player5);

play(handles.player);

play(handles.player1);

% --- Executes on button press in pushbutton20.

function pushbutton20\_Callback(hObject, eventdata, handles)

% hObject handle to pushbutton20 (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 pause\_all.

function pause\_all\_Callback(hObject, eventdata, handles)

% hObject handle to pause\_all (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

pause(handles.player5);

pause(handles.player);

pause(handles.player1);

% --- Executes on button press in resume\_all.

function resume\_all\_Callback(hObject, eventdata, handles)

% hObject handle to resume\_all (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

resume(handles.player5);

resume(handles.player);

resume(handles.player1);

% --- Executes on slider movement.

function slider77\_Callback(hObject, eventdata, handles)

% hObject handle to slider77 (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

global y;

global Fs;

pause(handles.player);

newstart=get(handles.player,'CurrentSample')+1;

stop(handles.player);

y=y(newstart:end,:);

freq1=get(handles.slider77,'value');

handles.player=audioplayer(y,Fs\*freq1);

play(handles.player);

guidata(hObject,handles);

% --- Executes on slider movement.

function slider79\_Callback(hObject, eventdata, handles)

% hObject handle to slider79 (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

global x;

global Ys;

pause(handles.player5)

newstart2=get(handles.player5,'CurrentSample')+1;

stop(handles.player5);

x=x(newstart2:end,:);

freq3=get(handles.slider79,'value');

handles.player5=audioplayer(x,Ys\*freq3);

play(handles.player5);

guidata(hObject,handles);

% --- Executes on slider movement.

function slider78\_Callback(hObject, eventdata, handles)

% hObject handle to slider78 (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

global z;

global Ts;

pause(handles.player1)

newstart1=get(handles.player1,'CurrentSample')+1;

stop(handles.player1);

z=z(newstart1:end,:);

freq2=get(handles.slider78,'value');

handles.player1=audioplayer(z,Ts\*freq2);

play(handles.player1);

guidata(hObject,handles);

**Screenshot:**

