

```

function varargout = G_Tune(varargin)

g_S = 1;
graf_St = struct('gui_Name',      mfilename, ...
                 'gui_Singleton', g_S, ...
                 'gui_OpeningFcn', @G_Tune_Create, ...
                 'gui_OutputFcn',  @G_Tune_Out, ...
                 'gui_LayoutFcn',  [], ...
                 'gui_Callback',   []);
if nargin && ischar(varargin{1})
    graf_St.gui_Callback = str2func(varargin{1});%
end
if nargin
    [varargout{1:nargout}] = gui_mainfcn(graf_St,
varargin{:});
else
    gui_mainfcn(graf_St, varargin{:});
end
% конец инициализации
% перед инициализацией тюнера
function G_Tune_Create(QQ, ~, handles, varargin)
% командная линия по умолчанию
handles.output = QQ;
% обновление структуры handles
guidata(QQ, handles);
load play_rec_ico;
set(handles.Pl_b, 'CData', PlayOff);
set(handles.Record_b, 'CData', RecOff);
disp('Гитарный тюнер');
% выходные параметры данной функции возвращают значения в
командную строку
function varargout = G_Tune_Out(~, ~, handles)
varargout{1} = handles.output;
% Выполняется во время создания объекта, после установки всех
свойств
function freq1_CreateFcn(QQ, ~, ~)

if ispc && isequal(get(QQ, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
    set(QQ, 'BackgroundColor', 'white');
end

function MicFFT(Fs, update_rate, ff, handles)
df=30;
recObj = audiorecorder(Fs, 16, 2);
recordblocking(recObj, update_rate);

```

```

myRecording = getaudiodata(recObj);
L = length(myRecording);
NFFT = 2^nextpow2(L);
Y = fft(myRecording,NFFT)/L;
f = Fs/2*linspace(0,1,NFFT/2+1);
plot(f,2*abs(Y(1:NFFT/2+1)))

if ff~=8
liney=[0 1];
freq = str2double(get(handles.freq1,'String'));
axis([freq-df freq+df 0 0.01])
linex = [freq freq];
line(linex,liney,'Color','m');
line(linex+2,liney,'Color','green');
line(linex-2,liney,'Color','green');
end
if ff==8
axis([0 1600 0 0.01,]);
end;
set(gca,'YColor','w','YTick',[])
set(gca,'XColor','w')
drawnow;

```

% Выполняется во время создания объекта, после установки всех свойств

```
function popupmenu1_CreateFcn(QQ, ~, ~)
```

```

if ispc && isequal(get(QQ,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(QQ,'BackgroundColor','white');
end

```

```

% создание осей, которые охватывают весь графический интерфейс
ah= axes('unit', 'normalized', 'position', [0 0 1 1]);
% импортирование фонового изображения и его размещение по осям
bg= imread('background.jpg'); imagesc(bg);
% предотвращение построения за границами фона

```

```

set(ah,'handlevisibility','off','visible','off')
% фон располагается позади всех объектов
uistack(ah, 'bottom');

```

% кнопка плэй

```
function Pl_b_Callback(QQ, ~, handles)
```

```

load play_rec_ico;
Fd = 44100;
t = get(handles.popupmenu8, 'Value');
set(QQ, 'CData', PlayOn);
freq = str2double(get(handles.freq1, 'String'));
t=[0:1/Fd:t];
sig= sin(2*pi*freq*t);
soundsc(sig,Fd);
pause(t);
set(QQ, 'CData', PlayOff);

% функция для воспроизведения любой заданной вручную частоты
function freq1_Callback(QQ, ~, handles)

% кнопка записи
function Record_b_Callback(QQ, eventdata, handles)

load play_rec_ico;
if get(QQ, 'UserData')
    % если текущее состояние включена, выключить:
    set(QQ, 'UserData', 0);
    set(QQ, 'CData', RecOff)
else
    %если текущее состояние выключена, включить:
    set(QQ, 'UserData', 1);
    set(QQ, 'CData', RecOn)
    while get(QQ, 'UserData')==1
        ff=get(handles.popupmenu1, 'Value');
        MicFFT(16000, 1, ff, handles);
    end
end;

% меню
function popupmenu1_Callback(QQ, ~, handles)
tones = [82.41 110 146.83 196 246.94 329.63 ]; %выбор
ноты
selection = get(QQ, 'Value');
if selection<7,

set(handles.freq1, 'String', num2str(tones(selection), '%3.2f'));
%отображение частоты в окне
    set(handles.freq1, 'Enable', 'off'); %Dcgksdf.ott vty. ds
отключение активности окна частоты для ввода своих данных
else
    set(handles.freq1, 'Enable', 'on'); % включение активности
окна частоты для ввода своих данных

```

```

end;

% Метроном
function slide_Callback(hObject, ~, handles)
% hObject      handle to slide (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
filename='1.wav'; %чтение файла 1
[x,fs, bits]=wavread(filename);

filename2='2.wav'; %чтение файла 2
[y,fs, bits]=wavread(filename2);

while get(hObject,'Value')~=0

k=get(hObject,'Value'); % чтение данных слайдера
set(handles.bps,'String',round(get(hObject,'Value'))); % вывод
BPS в окно
wavplay(y, fs); %воспроизведение сильной доли
pause(60/k-0.07); % регулировка BPS
%прерывание
if get(hObject,'Value')==0
break;
end;
for i=1:1:get(handles.popup,'Value')
wavplay(x, fs); %воспроизведение слабой доли
pause(60/k-0.07);
%прерывание
if get(hObject,'Value')==0
break;
end
end;
    % вывод BPS в окно
set(handles.bps,'String',get(hObject,'Value'));
end;

% Функция закрытия окна
function off_button_Callback(~, ~, handles)
clc; % очистка командного окна
set(handles.slide,'Value',0); % сброс слайдера на 0

```

```

close(G_Tune);
function slide_CreateFcn(hObject, ~, ~)
if isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end
function bps_CreateFcn(hObject, ~, ~)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function popup_Callback(~, ~, ~)
function popup_CreateFcn(hObject, ~, ~)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function popupmenu8_Callback(hObject, eventdata, handles)

function popupmenu8_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```