```
%   Academic Integrity Statement:
%   We have not used source code obtained from
%   any other unauthorized source, either modified
%   or unmodified.  Neither have we provided access
%   to our code to other teams. The project we are
%   submitting is our own original work.


function varargout = openingGUI(varargin)
global count;
global winnerName;
global winnerScore;
global darkTheme;

count = 0;
% OPENINGGUI MATLAB code for openingGUI.fig
%      OPENINGGUI, by itself, creates a new OPENINGGUI or raises the
 existing
%      singleton*.
%
%      H = OPENINGGUI returns the handle to a new OPENINGGUI or the
 handle to
%      the existing singleton*.
%
%      OPENINGGUI('CALLBACK',hObject,eventData,handles,...) calls the
 local
%      function named CALLBACK in OPENINGGUI.M with the given input
 arguments.
%
%      OPENINGGUI('Property','Value',...) creates a new OPENINGGUI or
 raises the
%      existing singleton*.  Starting from the left, property value
 pairs are
%      applied to the GUI before openingGUI_OpeningFcn gets called.
 An
%      unrecognized property name or invalid value makes property
 application
%      stop.  All inputs are passed to openingGUI_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 openingGUI

% Last Modified by GUIDE v2.5 04-Dec-2017 16:34:49

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
```

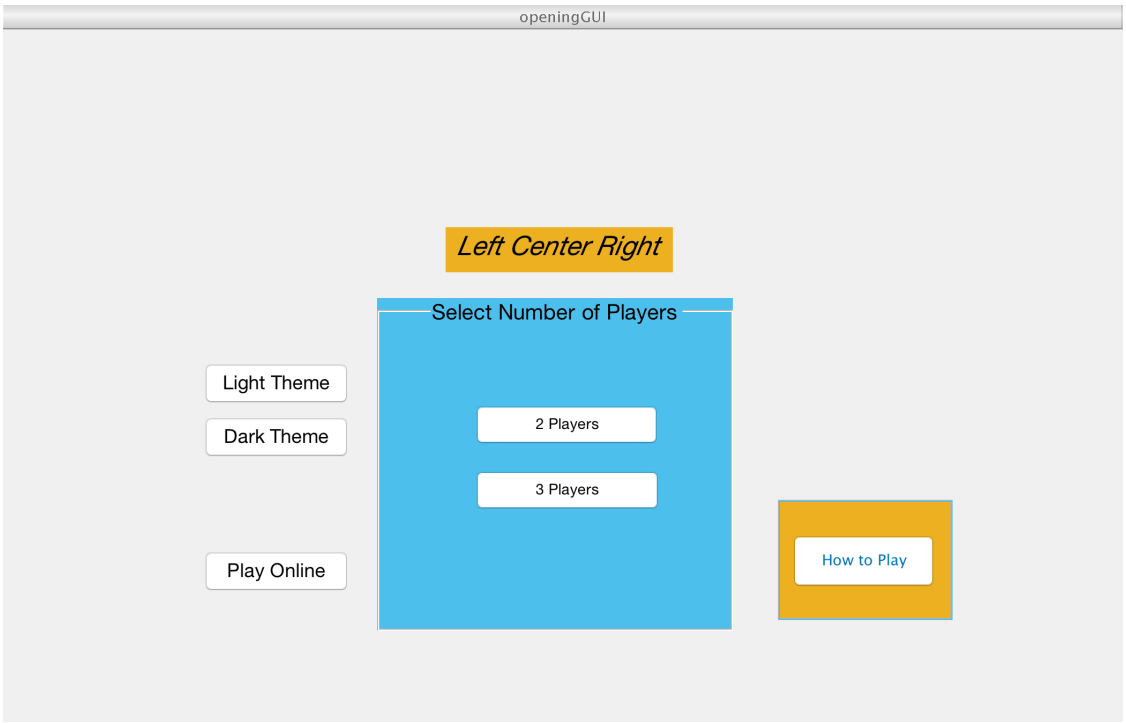
```
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @openingGUI_OpeningFcn, ...
                   'gui_OutputFcn',  @openingGUI_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   '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 openingGUI is made visible.
function openingGUI_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 openingGUI (see VARARGIN)

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

% Update handles structure
guidata(hObject, handles);
set(handles.howtoplaytext,'visible','off')



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



% --- Outputs from this function are returned to the command line.
function varargout = openingGUI_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 togglebutton2.
function togglebutton2_Callback(hObject, eventdata, handles)
% hObject    handle to togglebutton2 (see GCBO)
```

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

% Hint: get(hObject,'Value') returns toggle state of togglebutton2


% --- Executes on button press in togglebutton3.
function togglebutton3_Callback(hObject, eventdata, handles)
% hObject    handle to togglebutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of togglebutton3


% --- Executes on button press in togglebutton4.
function togglebutton4_Callback(hObject, eventdata, handles)
% hObject    handle to togglebutton4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of togglebutton4


% --- Executes on button press in togglebutton5.
function togglebutton5_Callback(hObject, eventdata, handles)
% hObject    handle to togglebutton5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of togglebutton5


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

% Hint: get(hObject,'Value') returns toggle state of howtoplaypush
if get(hObject, 'Value') == 1
    set(handles.howtoplaytext,'visible','on')
else
    set(handles.howtoplaytext,'visible','off')
end




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

```
% Opens two player GUI for game play
close(openingGUI);
run('twoplayerGUI')


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

% Opens three player GUI for game play
close(openingGUI);
run('threeplayerGUI')


% % --- Executes on button press in fourplayerpush.
% function fourplayerpush_Callback(hObject, eventdata, handles)
% % hObject    handle to fourplayerpush (see GCBO)
% % eventdata  reserved - to be defined in a future version of MATLAB
% % handles    structure with handles and user data (see GUIDATA)
% close(openingGUI);
% run('fourplayerGUI')


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

% Hint: get(hObject,'Value') returns toggle state of playOnline
client_id = '968206513642-
aab2kc8d5m7roa0oreejddtc5rj3pu5a.apps.googleusercontent.com';
client_secret = 'rKpxIvO4xbFi_K7IeHrGKZcc';
spreadsheetID = '1XKkV4B15hJeu7fMXkjvqLBJsmOsX-_j1zB1i_0AnrTI';
sheetID = '0';


    RunOnce(client_id, client_secret); % connect once
    mat2sheets(spreadsheetID, sheetID, [1 1], {'player1'}); %player1
 always starts first
    mat2sheets(spreadsheetID, sheetID, [2 1], {0});  %initial scores
    mat2sheets(spreadsheetID, sheetID, [3 1], {0});  %initial scores

    %get user id and store it inside the variable playerName. Choices
 are
    %player1 one or player2
    if(playerID == 1)
        currentPlayerName = 'player1';
        otherPlayerName = 'player2';
    else
        currentPlayerName = 'player2';
```

```
        otherPlayerName = 'player1';
    end

% function throwDice_Callback(hObject, eventdata, handles)
%
%    msgbox('Waiting for other player.'); %display a message
%    while true
%         % find whose turn it is
%         gameData = GetGoogleSpreadsheet(spreadsheetID);
%         currentPlayer = gameData{1, 1};
%         if(strcmp(currentPlayer, currentPlayerName))  %current
 player turn
%            % get the most recent scores
%             player1Score = str2double(gameData{2, 1});
%             player2Score = str2double(gameData{3, 1});
%             %update your score panel (in the gui)
%             msgbox('Your turn!'); %display a message
%             break;
%         end
%         pause(3); %wait 3 seconds for the other player
%
%         %additions - to make the game more interactive
%         %sharing the dice throw results
%         %share more data if you want using the same idea
%         gameState = gameData{1, 2};
%         if(strcmp(gameState, 'diceThrown')) %did the other player
 threw the dice?
%         otherPlayerDice = gameData{1, 3}; %get the dice throw
 results
%         % animate the dice code
%         % update the dice in your gui to show what the other player
 got
%         end
%    end
%
%
% %proceed to your game throwDice steps ....
% %find the player score
%
% %....after your throwDice code
%
% %additions - to make the game more interactive
% mat2sheets(spreadsheetID, sheetID, [1 2], {'diceThrown'}); %share
 throwing dice results
% mat2sheets(spreadsheetID, sheetID, [1 3], {diceResults});
%
% if(strcmp('player1', currentPlayerName)) %update score
%     mat2sheets(spreadsheetID, sheetID, [2 1], {calculatedScore});
% else
%     mat2sheets(spreadsheetID, sheetID, [3 1], {calculatedScore});
% end
% %take turns
%  mat2sheets(spreadsheetID, sheetID, [1 1], {otherPlayerName});
```

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

% Hint: get(hObject,'Value') returns toggle state of darkbutton
% Sets background color to purple
global darkTheme;
set ( openingGUI, 'Color',[0.5,0,0.5] )
darkTheme = true;


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

% Hint: get(hObject,'Value') returns toggle state of lightbutton
% Sets background color to green
global darkTheme;
set ( openingGUI, 'Color',[0, 207, 0] ./ 255 )
darkTheme = false;
```



Published with MATLAB® R2017a