



Documentation for the *octave* coffee machine project

Emre Arapcic-Uevak
Vedad Siljic

January 12, 2023

Abstract

This is a documentation file for the final project assignment for the course **ENS101**

Contents

1	Assignments	3
2	Design	3
2.1	UI component set up	3
2.2	Setting up Interacting drop down menu	4
2.3	Set up KeyPad buttons	5
2.4	Set up coffee menu	6
2.5	Save all UI data	6
3	Functionality	7
3.1	KeyPad Buttons Functionalities	7
3.2	Coffee Machien Interaction Functionality	8

1 Assignments

The following table will show all the assignments taken by the corresponding students:

	EMRE ARAPCIC-UEVAK	VEDAD SILJIC
Design	✓✓	x
Functionality	x	✓✓

2 Design

Lets take a look at the code that was used to generate all the ui components

2.1 UI component set up

```
1  % Set up all needed frames
2  mainFrame = figure(
3      'position', [0 0 800 600],
4      'Name', 'Coffee Machine',
5      'NumberTitle', 'off',
6      'color', primaryColor,
7      'toolbar', 'none',
8      'resize', 'off');
9
10 titleFrame = uipanel(
11     'Parent', mainFrame,
12     'position', [0 .85 1 .15],
13     'HighLightColor', primaryColor);
14
15 rightHalf = uipanel('Parent', mainFrame,
16     'position', [0.5 0 0.5 .85],
17     'backgroundcolor', primaryColor,
18     'HighLightColor', primaryColor );
19
20 keypadFrame = uipanel('Parent', rightHalf,
21     'position', [0.05 0.3 0.5 0.5],
22     'HighLightColor', primaryColor );
23
24 coffeeSlotFrame = uipanel('Parent', rightHalf,
25     'position', [0.6 0.3 0.35 0.5],
26     'HighLightColor', primaryColor );
27
28 leftHalf = uipanel('Parent', mainFrame,
29     'position', [0 0 .5 0.85],
30     'backgroundcolor', primaryColor ,
31     'HighLightColor', primaryColor );
32
33 coffeeMenu = uipanel('Parent', leftHalf,
34     'position', [0.025 0.05 0.95 .9],
35     'backgroundcolor', primaryColor,
36     'HighLightColor', primaryColor );
37
38 % Set up all elements
39 data.coffeeGifDisplayAxes = axes('Parent', coffeeSlotFrame,
40     'position', [0 0 1 1],
41     'xtick', [], 'ytick', [], 'xlim', [0 1], 'ylim', [0 1],
42     'Color', primaryColor);
43 data.consoleOutput = uicontrol(rightHalf, 'Style', 'edit', 'units', 'normalized' ...
44     , 'string', 'IUS Coffee Machine', 'position', [0.05 0.85 0.9 0.13], 'Max', 5, ...
45     'Min', 0, 'enable', 'off', 'backgroundcolor', '#000000');
46
47 % Set up coffee animation
48 axes(data.coffeeGifDisplayAxes)
49 frame = imread('./CoffeePercentBarFrames/finalCoffeeVideo-0.png');
50 imshow(frame, []);
51
52 % Set up title
53 uicontrol('parent', titleFrame,
54     'Style', 'text',
```

```

53         'units', 'normalized' ,
54         'string', 'IUS COFFEE',
55         'position', [0 0 1 1],
56         'backgroundcolor', primaryColor, 'fontsize', 18);

```

Quick note about about some of the functions used in the given code block:

- **Figure** is an function used for generating a new figure with the given properties.
- **Uipanel** is a function used for generating a panel(frame) which acts as a group for other ui components.
- **Axes** is a function used for generating a new axes with the given properties.

As we can see from line 2 to 8 we are generating a new figure that will not have a tool bar and will be called "Coffee Machine", it will as well not be resizable. it will start on the bottom left of the users screen with a fixed size of 800×600 pixels.

After that from line 10 all the way down to line 36 we will seperate the figure into multiple frames on on the far top for the title and then split it in 2 vertically with the rest of the space remaining, we called these spaces *rightHalf* and *leftHalf* in our program. Afterwards we will set up some *Axes* that will parented to the given frames which will be later used for image displaying. The *coffeeGifDisplayAxes* is an *Axes* that will be used for displaying a progress bar type gif, but unfortunately since *Octave* is really slow at handling image loading and does not have *GIF nor Video* support all the images had to be loaded one after another.

2.2 Setting up Interacting drop down menu

```

1  % Set up menubar
2  interactions = uimenu('Text', 'Interactions');
3  interactionsItem1 = uimenu(interactions, 'Text', 'Insert 5KM', 'callback',
4      { @interactionPressed, '5' });
5
6  interactionsItem2 = uimenu(interactions, 'Text', 'Insert 2KM', 'callback',
7      { @interactionPressed, '2' });
8
9  interactionsItem3 = uimenu(interactions, 'Text', 'Insert 1KM', 'callback',
10     { @interactionPressed, '1' });
11
12 interactionsItem4 = uimenu(interactions, 'Text', 'Insert 50F', 'callback',
13     { @interactionPressed, '0.5' });
14
15 interactionsItem5 = uimenu(interactions, 'Text', 'Insert 20F', 'callback',
16     { @interactionPressed, '0.2' });
17
18 interactionsItem6 = uimenu(interactions, 'Text', 'Insert 10F', 'callback',
19     { @interactionPressed, '0.1' });
20
21 interactionsItem7 = uimenu(interactions, 'Text', 'Insert 5F', 'callback',
22     { @interactionPressed, '0.05' });
23
24 interactionsItem8 = uimenu(interactions, 'Text', 'Take the coffee', 'callback',
25     { @interactionPressed, 'takeCoffee' });
26
27 interactionsItem9 = uimenu(interactions, 'Text', 'Take the change', 'callback',
28     { @interactionPressed, 'takeChange' });

```

The above presented code is fairly simple to understand, firstly we use **Uimenu** function to create a new menu component that will be displayed on the top of the frame, afterwards we just create the wanted tabs, parents them to the given menu we just created, change their text, and a call back function to every single one of them.

2.3 Set up Keypad buttons

```
1 % Set up Keypad Buttons
2 for i = 1:4
3     for j = 1:3
4         if (i != 4)
5             uicontrol(keypadFrame,
6                 'Style', 'pushbutton',
7                 'units', 'normalized' ,
8                 'string', num2str(3*(i-1) + j - 1),
9                 'callback', { @buttonPressed, num2str(3*(i-1) + j - 1) },
10                'position', [1/3*(j-1) 1-1/4*i 1/3 1/4],
11                'Max', 5, 'Min', 0, 'enable', 'on', 'fontsize', 14, 'backgroundcolor', ...
                    secondaryColor);
12        else
13            if (j == 2)
14                uicontrol(keypadFrame,
15                    'Style', 'pushbutton',
16                    'units', 'normalized' ,
17                    'string', '9',
18                    'callback', { @buttonPressed, '9' },
19                    'position', [1/3 0 1/3 1/4],
20                    'Max', 5, 'Min', 0, 'enable', 'on',
21                    'fontsize', 14, 'backgroundcolor', secondaryColor);
22            elseif (j == 1)
23                uicontrol(keypadFrame,
24                    'Style', 'pushbutton',
25                    'units', 'normalized' ,
26                    'string', 'Yes',
27                    'callback', { @buttonPressed, 'yes' },
28                    'position', [0 0 1/3 1/4],
29                    'Max', 5, 'Min', 0, 'enable', 'on',
30                    'fontsize', 14, 'backgroundcolor', secondaryColor);
31            else
32                uicontrol(keypadFrame,
33                    'Style', 'pushbutton',
34                    'units', 'normalized' ,
35                    'string', 'No',
36                    'callback', { @buttonPressed, 'no ' },
37                    'position', [2/3 0 1/3 1/4],
38                    'Max', 5, 'Min', 0, 'enable', 'on',
39                    'fontsize', 14, 'backgroundcolor', secondaryColor);
40            end
41        end
42    end
43 end
```

The above mentioned code was used to generate all the buttons that can be found on the keypad frame, here we can see the use of nested loops to allow us to automatically add all the needed buttons, and we can also see that we have a special case if we are dealing with the last row since we do not only want numbers to be displayed there, we also want to display the words "Yes", "No". For the rest of the document the documentation about all of these functions will be linked at the end of the document

2.4 Set up coffee menu

```
1 % Set up for coffee menu
2 for i = 1:3
3     for j = 1:3
4         if (3*(i-1) + j > length(data.coffeeNames)) break; end
5         coffeeFrame = uipanel('Parent', coffeeMenu,
6             'position', [(j-1)/3 1-i/3 1/3 1/3],
7             'HighLightColor', primaryColor ,
8             'backgroundcolor',primaryColor );
9
10        axes('Parent', coffeeFrame,
11            'position', [0 0.4 1 0.6],
12            'xtick', [], 'ytick', [], 'xlim', [0 1], 'ylim', [0 1],
13            'Color', primaryColor);
14
15        imshow(strcat('./Images/', data.coffeeNames{3*(i-1) + j}, '.png'), []);
16
17        uicontrol(coffeeFrame,
18            'Style', 'text',
19            'units', 'normalized' ,
20            'string', num2str(3*(i-1) + j),
21            'position', [0 0.2 1 0.2],
22            'enable', 'on', 'fontsize', 14,
23            'backgroundcolor',primaryColor );
24
25        uicontrol(coffeeFrame,
26            'Style', 'text',
27            'units', 'normalized' ,
28            'string', data.coffeeNames{3*(i-1) + j},
29            'position', [0 0 1 0.2],
30            'enable', 'on', 'fontsize', 14,
31            'backgroundcolor',primaryColor );
32    end
33 end
```

The above mentioned code was used to generate all images for coffees that can be found in the coffee menu upon ordering, what the code does is it goes by the same principal as the generation for keypad buttons by using a nested loop to make frames that all take $\frac{1}{3}$ of width and height of the available space. Afterwards it makes an axis and 2 text components that is lays next to each other vertically and sets all of their properties.

2.5 Save all UI data

```
1 guidata(mainFrame, data);
```

And finally we will use this line of code to set up everything we have saved in the structure variables called "*data*" and link that variable to the figure. We do this because Octaves scope is function based meaning any variables made in a function is *only* visible in the said function. But by doing this we can use the function guidata again to read the structure we connected to the figure element.

3 Functionality

In this section we will take a look at the code responsible for functionality of the program.

3.1 KeyPad Buttons Functionalities

```
1 % For all buttons on the keypad
2 function buttonPressed(hObject, eventdata, value)
3
4     data = guidata(gcf());
5
6     if (data.enableButtons != 1)
7         return;
8     end
9
10    % Yes Button
11    if (value == 'yes')
12
13        if (data.askForExtraMilk == 1)
14            data.askForExtraMilk = 0;
15            data.askForExtraSugar = 0;
16            data.extraMilk = 1;
17            pause(1);
18            set(data.consoleOutput, 'string', strcat('The total is ', ...
19                num2str(data.coffeePrice{str2num(data.number)}), 'KM'));
20            data.enableButtons = 0;
21            data.enableInteractions = 1;
22        end
23
24        if (data.askForExtraSugar == 1)
25            data.askForExtraSugar = 0;
26            data.extraSugar = 1;
27            pause(1);
28            set(data.consoleOutput, 'string', 'Do you want extra milk?');
29            data.askForExtraMilk = 1;
30        end
31
32        if (data.numberEntered == 0)
33            if (str2num(data.number) == 1 || str2num(data.number) == 2 || ...
34                str2num(data.number) == 3 || str2num(data.number) == 4 || ...
35                str2num(data.number) == 5 || str2num(data.number) == 6 || ...
36                str2num(data.number) == 7 || str2num(data.number) == 8 || ...
37                str2num(data.number) == 9)
38                set(data.consoleOutput, 'string', ...
39                    strcat(data.coffeeNames{str2num(data.number)}, ' selected. ');
40                pause(1);
41                data.numberEntered = 1;
42                data.enableNumbers = 0;
43                set(data.consoleOutput, 'string', 'Do you want extra sugar?');
44                data.askForExtraSugar = 1;
45            else
46                set(data.consoleOutput, 'string', 'Wrong number entered. ');
47                pause(2);
48                data.number = '';
49                set(data.consoleOutput, 'string', data.number);
50            end
51        end
52
53    % No Button
54    elseif (value == 'no ')
55
56        if (data.askForExtraMilk == 1)
57            data.askForExtraMilk = 0;
58            data.askForExtraSugar = 0;
59            data.extraMilk = 0;
60            pause(1);
61            set(data.consoleOutput, 'string', strcat('The total is ', ...
62                num2str(data.coffeePrice{str2num(data.number)}), 'KM'));
63            data.enableButtons = 0;
64            data.enableInteractions = 1;
```

```

58     end
59
60     if (data.askForExtraSugar == 1)
61         data.askForExtraSugar = 0;
62         data.extraSugar = 0;
63         pause(1);
64         set(data.consoleOutput, 'string', 'Do you want extra milk?');
65         data.askForExtraMilk = 1;
66     end
67
68     if (data.numberEntered == 0)
69         data.number = '';
70         set(data.consoleOutput, 'string', data.number);
71     else
72
73     end
74
75     % Number Buttons
76     else
77         if (data.enableNumbers == 0)
78             return;
79         end
80         data.number = strcat(data.number, value);
81         set(data.consoleOutput, 'string', data.number);
82     end
83
84     guidata(gcf(), data);
85
86 end

```

This function is for all interactions with keypad of the coffee machine. It has every functionality for selecting which coffee you want and if you want extra sugar and milk.

3.2 Coffee Machien Interaction Functionality

```

1  % For all interactions on the menubar
2  function interactionPressed(hObject, eventdata, value)
3
4      data = guidata(gcf());
5
6
7      if (value == 'takeCoffee')
8          if (data.coffeeFinished == 1)
9              set(data.consoleOutput, 'string', 'Enjoy your coffee. ');
10             pause(2);
11             close all;
12             return;
13         else
14             return;
15         end
16     end
17
18
19     if (value == 'takeChange')
20         if (data.hasChange == 1)
21             data.coffeeFinished = 1;
22             data.hasChange = 0;
23             set(data.consoleOutput, 'string', 'Please wait while the coffee is being made. ');
24             pause(1);
25
26             % Coffee making sound and animation
27             [y, fs] = audioread('./Audio/coffeeMakingAudio.wav');
28             player = audioplayer(y, fs);
29             play(player);
30
31             axes(data.coffeeGifDisplayAxes)
32             for i = 1:49
33                 frame = imread(strcat('./CoffeePercentBarFrames/finalCoffeeVideo-', ...
34                                     num2str(i), '.png'));

```



```

35         pause(50/450);
36     end
37
38     if (data.extraSugar == 1 && data.extraMilk == 1)
39         set(data.consoleOutput, 'string', strcat('Here is your ', ...
40             data.coffeeNames{str2num(data.number)}, ' with extra sugar and extra milk.));
41     elseif (data.extraSugar == 1 && data.extraMilk == 0)
42         set(data.consoleOutput, 'string', strcat('Here is your ', ...
43             data.coffeeNames{str2num(data.number)}, ' with extra sugar.));
44     elseif (data.extraSugar == 0 && data.extraMilk == 1)
45         set(data.consoleOutput, 'string', strcat('Here is your ', ...
46             data.coffeeNames{str2num(data.number)}, ' with extra milk.));
47     else
48         set(data.consoleOutput, 'string', strcat('Here is your ', ...
49             data.coffeeNames{str2num(data.number)}, '.));
50     end
51     pause(2);
52     set(data.consoleOutput, 'string', 'Please take your coffee. ');
53     else
54         return;
55     end
56 end
57
58 guidata(gcf(), data);
59
60 if (data.coffeePaid == 1)
61     return;
62 end
63
64 numValue = str2num(value);
65 data.coffeePrice{str2num(data.number)} -= numValue;
66 strValue = num2str(data.coffeePrice{str2num(data.number)});
67
68 if (data.coffeePrice{str2num(data.number)} > 0)
69
70     set(data.consoleOutput, 'string', strcat(strValue, 'KM left to enter. ');
71
72
73 elseif (data.coffeePrice{str2num(data.number)} == 0)
74
75     set(data.consoleOutput, 'string', 'Please wait while the coffee is being made. ');
76     data.coffeePaid = 1;
77     pause(1);
78
79     % Coffee making sound and animation
80     [y, fs] = audioread('./Audio/coffeeMakingAudio.wav');
81     player = audioplayer(y, fs);
82     play(player);
83
84     axes(data.coffeeGifDisplayAxes)
85     for i = 1:49
86         frame = imread(strcat('./CoffeePercentBarFrames/finalCoffeeVideo-', ...
87             num2str(i), '.png'));
88         imshow(frame, []);
89         pause(50/450);
90     end
91
92     data.coffeeFinished = 1;
93     if (data.extraSugar == 1 && data.extraMilk == 1)
94         set(data.consoleOutput, 'string', strcat('Here is your ', ...
95             data.coffeeNames{str2num(data.number)}, ' with extra sugar and extra milk.));
96     elseif (data.extraSugar == 1 && data.extraMilk == 0)
97         set(data.consoleOutput, 'string', strcat('Here is your ', ...
98             data.coffeeNames{str2num(data.number)}, ' with extra sugar.));
99     elseif (data.extraSugar == 0 && data.extraMilk == 1)
100         set(data.consoleOutput, 'string', strcat('Here is your ', ...
101             data.coffeeNames{str2num(data.number)}, ' with extra milk.));
102     else

```

```

99     set(data.consoleOutput, 'string', strcat('Here is your ', ...
        data.coffeeNames{str2num(data.number)}, '.'));
100 end
101 pause(2);
102 set(data.consoleOutput, 'string', 'Please take your coffee.');
```

```

103
104
105 else
106     set(data.consoleOutput, 'string', 'Please take your change');
107     data.hasChange = 1;
108     data.coffeePaid = 1;
109 end
110
111 guidata(gcf(), data);
112
113 end

```

This function is for all the interactions options that can be found in the intercation drop down menu like:

- Inserting coins
- Taking change
- Taking coffee