Documentation for the *octave* coffee machine project

Emre Arapcic-Uevak
Vedad Siljic

January 11, 2023

Abstract

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

Contents

1 Assignments

The following table will show all the assignemtns 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

```
% Set up all needed frames
 mainFrame = figure(
       'position', [0 0 800 600],
       'Name','Coffee Machine',
       'NumberTitle', 'off',
       'color', primaryColor,
       'toolbar','none',
       'resize', 'off');

 titleFrame = uipanel(
       'Parent', mainFrame,
       'position', [0 .85 1 .15],
       'HighLightColor', primaryColor);

 rightHalf = uipanel('Parent', mainFrame,
       'position', [0.5 0 0.5 .85],
       'backgroundcolor', primaryColor,
       'HighLightColor', primaryColor );

 keypadFrame = uipanel('Parent', rightHalf,
       'position', [0.05 0.3 0.5 0.5],
       'HighLightColor', primaryColor );

 coffeeSlotFrame = uipanel('Parent', rightHalf,
       'position', [0.6 0.3 0.35 0.5],
       'HighLightColor', primaryColor );

 leftHalf = uipanel('Parent', mainFrame,
       'position', [0 0 .5 0.85],
       'backgroundcolor', primaryColor ,
       'HighLightColor', primaryColor );

 coffeeMenu = uipanel('Parent', leftHalf,
       'position', [0.025 0.05 0.95 .9],
       'backgroundcolor', primaryColor,
       'HighLightColor', primaryColor );

 % Set up all elements
 data.coffeeGifDisplayAxes = axes('Parent', coffeeSlotFrame,
       'position', [0 0 1 1],
       'xtick', [], 'ytick', [], 'xlim', [0 1], 'ylim', [0 1],
       'Color', primaryColor);
 data.consoleOutput = uicontrol(rightHalf, 'Style', 'edit', 'units', 'normalized' ...
     ,'string', 'IUS Coffee Machine','position', [0.05 0.85 0.9 0.13], 'Max', 5, ...
     'Min', 0, 'enable', 'off','backgroundcolor', '#000000');

 % Set up coffee animation
 axes(data.coffeeGifDisplayAxes)
 frame = imread('./CoffeePercentBarFrames/finalCoffeeVideo-0.png');
 imshow(frame, []);

 % Set up title
 uicontrol('parent', titleFrame,
       'Style', 'text',
       'units', 'normalized' ,
       'string', 'IUS COFFEE',
       'position', [0 0 1 1],
       '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.

```
% Set up menubar
  interactions = uimenu('Text', 'Interactions');
  interactionsItem1 = uimenu(interactions, 'Text', 'Insert 5KM', 'callback',
        { @interactionPressed, '5         ' });

  interactionsItem2 = uimenu(interactions, 'Text', 'Insert 2KM', 'callback',
        { @interactionPressed, '2         ' });

  interactionsItem3 = uimenu(interactions, 'Text', 'Insert 1KM', 'callback',
        { @interactionPressed, '1         ' });

  interactionsItem4 = uimenu(interactions, 'Text', 'Insert 50F', 'callback',
        { @interactionPressed, '0.5       ' });

  interactionsItem5 = uimenu(interactions, 'Text', 'Insert 20F', 'callback',
        { @interactionPressed, '0.2       ' });

  interactionsItem6 = uimenu(interactions, 'Text', 'Insert 10F', 'callback',
        { @interactionPressed, '0.1       ' });

  interactionsItem7 = uimenu(interactions, 'Text', 'Insert 5F', 'callback',
        { @interactionPressed, '0.05      ' });

  interactionsItem8 = uimenu(interactions, 'Text', 'Take the coffee', 'callback',
        { @interactionPressed, 'takeCoffee' });

  interactionsItem9 = uimenu(interactions, 'Text', 'Take the change', 'callback',
        { @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 dispalyed 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.

```
% Set up KeyPad Buttons
  for i = 1:4
      for j = 1:3
          if (i != 4)
              uicontrol(keypadFrame,
              'Style', 'pushbutton',
              'units', 'normalized' ,
              'string', num2str(3*(i-1) + j - 1),
              'callback', { @buttonPressed, num2str(3*(i-1) + j - 1) },
              'position', [1/3*(j-1) 1-1/4*i 1/3 1/4],
              'Max', 5, 'Min', 0, 'enable', 'on', 'fontsize', 14, 'backgroundcolor', ...
                  secondaryColor);
          else
              if (j == 2)
                  uicontrol(keypadFrame,
                  'Style', 'pushbutton',
                  'units', 'normalized' ,
                  'string', '9',
                  'callback', { @buttonPressed, '9' },
                  'position', [1/3 0 1/3 1/4],
                  'Max', 5, 'Min', 0, 'enable', 'on',
                  'fontsize', 14,'backgroundcolor', secondaryColor);
              elseif (j == 1)
                  uicontrol(keypadFrame,
                  'Style', 'pushbutton',
                  'units', 'normalized' ,
                  'string', 'Yes',
                  'callback', { @buttonPressed, 'yes' },
                  'position', [0 0 1/3 1/4],
                  'Max', 5, 'Min', 0, 'enable', 'on',
                  'fontsize', 14,'backgroundcolor', secondaryColor);
              else
                  uicontrol(keypadFrame,
                  'Style', 'pushbutton',
                  'units', 'normalized' ,
                  'string', 'No',
                  'callback', { @buttonPressed, 'no ' },
                  'position', [2/3 0 1/3 1/4],
                  'Max', 5, 'Min', 0, 'enable', 'on',
                  'fontsize', 14,'backgroundcolor', secondaryColor);
              end
          end
      end
  end
```

The above mentioned code was used to generate all the buttons that can be found on the kaypad 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

```
% Set up for coffee menu
  for i = 1:3
      for j = 1:3
          if (3*(i-1) + j > length(data.coffeeNames)) break; end
          coffeeFrame = uipanel('Parent', coffeeMenu,
          'position', [(j-1)/3 1-i/3 1/3 1/3],
          'HighLightColor', primaryColor ,
          'backgroundcolor',primaryColor );

          axes('Parent', coffeeFrame,
          'position', [0 0.4 1 0.6],
          'xtick', [], 'ytick', [], 'xlim', [0 1], 'ylim', [0 1],
          'Color', primaryColor);

          imshow(strcat('./Images/', data.coffeeNames{3*(i-1) + j}, '.png'), []);

          uicontrol(coffeeFrame,
          'Style', 'text',
          'units', 'normalized' ,
          'string', num2str(3*(i-1) + j),
          'position', [0 0.2 1 0.2],
          'enable', 'on', 'fontsize', 14,
          'backgroundcolor',primaryColor );

          uicontrol(coffeeFrame,
          'Style', 'text',
          'units', 'normalized' ,
          'string', data.coffeeNames{3*(i-1) + j},
          'position', [0 0 1 0.2],
          'enable', 'on', 'fontsize', 14,
          'backgroundcolor',primaryColor );
      end
  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.

```
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.