**Example: Adding the Camera Module**

This is a reflection on how the camera screens were added to the software (GitHub commit: [153a361](https://github.com/byu-crop-biomechanics-lab/FIELDAQ/commit/153a3611583ae599b12dd20bf0e2d1b27eac2ee1)). The emphasis on how the screens were managed rather than introducing the picamera python library. See appendix for resources about implementing the picamera library.

The first thing done to add the camera module was to draw a new flowchart depicting navigation between the screens. This is depicted in Figure 1.

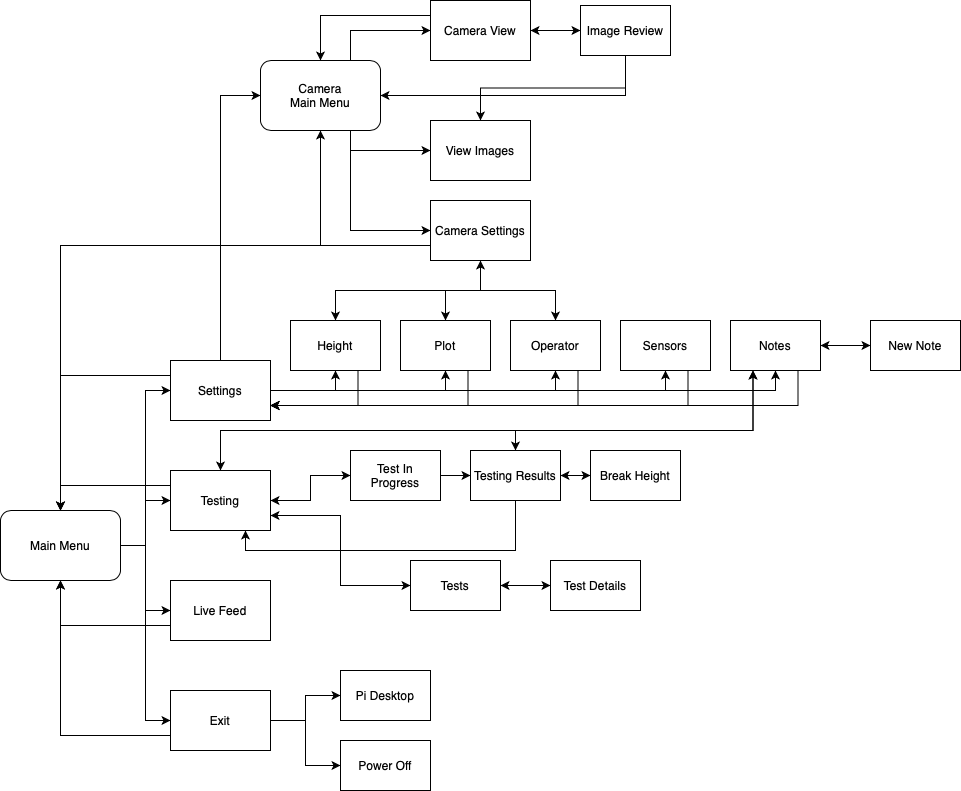


Figure 1. Screen shot of the updated flowchart featuring the Camera Main menu and 4 additional screens.

5 screens would be created, a main menu for the Camera interface, a screen for viewing the camera feed, an image review screen, a screen to see all images, and a settings screen. Once the flowchart had been thought through, several screens were identified that could be duplicated and modified in a simple way to achieve the new screen layout. The following pairs were identified:

• Camera Main Menu – Main Menu

• Camera View – Main Menu (looking back, the Exit Screen would be more appropriate due to the simple layout of the Exit Screen)

• Image Review – Main Menu (Exit Screen would also be more appropriate here as well for the same reasons as the Camera View Screen)

• View Images – Main Menu (Tests Screen would have been more appropriate here because of the use of a list)

• Camera Settings – Settings

The identified screen layout source files were duplicated and renamed. In addition, the 10 new files created (5 screens with a python and kivy file each) were moved into a new directory: view/screens/camera in order to help separate the camera screens from the rest of the software.

Defining the New Screens for the GUI

The “Kivy Screen Management” Document outlines 8 steps for defining a new screen in the GUI. The steps for adding the Camera Main Menu are used as an example here. The same steps were used for each of the 5 new screens added.

Step 1 had been completed and it was time to proceed with the rest of the steps.

Python Filename: CameraMainScreen.py

Kivy Filename: CameraMainScreen.kv

Step 2 was completed by default of duplicating files that already had the necessary libraries imported.

Steps 3-6 deal with the naming conventions used for each of the files and are closely related to each other.

Python file class (Step 3):

1. class CameraMainScreen(BaseScreen):

Python builder line of code (Step 4):

1. Builder.load\_file('view/screens/camera/CameraMainScreen.kv')

Outermost kivy node (Step 5):

1. <CameraMainScreen>:

Kivy node name attribute (Step 6):

1. name: 'cam\_main\_screen'

Now that the naming was consistent across each file, the kivy screen manager (src/main.kv) will be updated to know where each file is and what it was named.

Next, Steps 7 and 8 will import the new screens into the screen manager and allow the GUI to navigate throughout these files as they are referenced in other places of the GUI (to be completed later).

Step 7 is to add the following lines of code to src/main.kv in order to import the new screen files:

1. #:import CameraMainScreen view.screens.camera.CameraMainScreen
2. #:import CameraFeedScreen view.screens.camera.CameraFeedScreen
3. #:import ImageReviewScreen view.screens.camera.ImageReviewScreen
4. #:import ImagesViewScreen view.screens.camera.ImagesViewScreen
5. #:import CameraSettingsScreen view.screens.camera.CameraSettingsScreen

Note that these lines of code use a hash-colon notation, the kivy interpreter sees this as an instruction and ***not*** a comment.

Step 8 consists of appending the following lines of code to the end of the file under the GranuScreenManager node:

1. # Camera Interface
2. CameraMainScreen:
3. CameraFeedScreen:
4. ImageReviewScreen:
5. ImagesViewScreen:
6. CameraSettingsScreen:

Integrating the Newly Defined Screens to the Rest of the GUI

Now that the 8 steps for defining the screens in the GUI are completed, the next step in the process of integrating these screens fully into the GUI is to adjust the relationships between the new screens so the navigation can resemble the outlined flowchart. The changes for this step will be made completely in the kivy files for each of the new screens. (ADD SCREENSHOTS FOR CLARIFICATION HERE)

Beginning with the Main screen:

Rather than navigating to the Settings, Testing, Live Feed and Exit Screens; the new Camera Main Menu should navigate to the Camera Settings, Camera View, View Images and the Exit Screens. These changes are made by editing the text and on\_release attributes of the GranuSideButton elements.

The new code contained nested under the GranuSideArea looks like:

1. GranuSideArea:
2. GranuSideButton:
3. text: 'Settings'
4. on\_release:
5. root.move\_to('cam\_settings\_screen') # Move to camera settings screen
6. GranuSideButton:
7. text: 'Camera'
8. on\_release:
9. root.move\_to('cam\_feed\_screen') # Move to camera feed screen
10. GranuSideButton:
11. text: 'View\nImages'
12. on\_release:
13. root.move\_to('img\_viewer\_screen') # Move to image view screen
14. GranuSideButton:
15. text: 'Exit'
16. on\_release:
17. root.move\_to('exit\_screen') # Move to exit screen

Note that the root.move\_to() commands utilize the names of the screens created in Step 6 of defining the new screens.

A new title for this screen should also be used. This was changed by editing line 23 from ‘Main Menu’ to ‘Camera Main Menu’

1. GranuContent:
2. GranuTitle:
3. text: 'Camera Main Menu'

Similar processes were followed for the remaining 4 screens. If not all 4 side buttons were needed, a GranuNone element can be used instead of a GranuSideButton. An example is shown in CameraFeedScreen.kv:

1. GranuSideArea:
2. GranuSideButton:
3. text: 'Take\nPhoto'
4. on\_release:
5. root.captureImage() # Call the function captureImage in the CameraFeedScreen python class
6. GranuNone:
7. GranuNone:
8. GranuSideButton:
9. text: 'Back'
10. on\_release:
11. root.move\_to('cam\_main\_screen') # Move to main camera screen

Once all the navigation commands were adjusted, the next step was to add a way to navigate to these new screens from the original software. This was done by adding a SettingsButton element to the main Settings Screen. This would place a button on the settings screen in line with all the others and it would move the user to the new Camera Main Menu Screen.

To do this, a new SettingsButton element was added under the Grid Layout in the GranuContent portion of the screen. The SettingsButton behaves in the same way as the GranuSideButton but has slightly different formatting. The resulting code added to SettingsScreen.kv is as follows: (ADD VISUAL)

1. SettingsButton:
2. text: 'Switch to\nCamera'
3. on\_release:
4. root.move\_to('cam\_main\_screen')

A similar step was performed on the camera settings screen for navigation back to the Main Menu for push tests.

Finally, this code was [committed to the GitHub repository](https://github.com/byu-crop-biomechanics-lab/FIELDAQ/commit/153a3611583ae599b12dd20bf0e2d1b27eac2ee1) as a functioning way to navigate through the screens added. More work was still needed to get picamera functioning and integrated to the software.

**Appendix – GitHub Commits**

The following GitHub commit reference numbers are from the entire process of adding a camera interface to the GUI. Most commits correspond to introducing and adjusting settings in the picamera library. A few relate to the steps necessary to creating the screen layout for the camera GUI.

b94b3e2037045a361c083bacdd1c3b91ab8b49b5

**153a3611583ae599b12dd20bf0e2d1b27eac2ee1**

42d0f48efe83b1ba048048f84d4e9ce625446242

8c97848f8bfd95a503a01183c90860add473db36

ad5470b3f886762ab2fa284baceb448d8e697445

91474546b059fe3c1341b0d10b8fd6ce5b68d9a8

7c0072ddd34e6b07bd557428a1caae83b9103d57

53ede597e4f0dc924a34fb80b0aa05ae8b8aaeee

6f66fa05ee316f289fce5b841d0219dd0b614cec

6f71d4a4a0847b39393a5d1463893573b89583f9

fe22e930a228b0d537753bc5f11b5458fc403afe

e15383d523577d09f388e07e56ea6d760a1ef35d

170a3802bb82abf9d192aead9eff9ef02e57f373

6f6cc556901cdb44566afad40c06bea5b2038ff9

d99ea49d72321479ca6d247ca533f166eb156eb8

6434b298555fdba9d93417d92952415a135370e1

e3eb9dce3929be17b2052c14bbea0b52bdf9d4b6

d44bcbb137328fa76a07de5c2f722e83570334c7

a3ada4045244d44d5b644a12293cc0b44bc492ee

eda134962cbbf0bb86b31ba6722d5c94fde308b7

d4f6c27a0693778564320c858694706bc0636d7f

b830f616cb05111bcc85d4c8fa58c0719456bf4e

4e84b5c33865da2b2f334ffa7bb0e22542557ddf

7e7bd360304c55e5b48d21beda69a0047b54cda0