Drone Tracker

Release 1.0.3

Hayley Eckert, Jonathan Westerfield, Donald Elrod, Ismael Rodrig

CONTENTS:

1	Program Controller	1
2	OpenCV	5
3	Exceptions	7
4	PhoneController	9
5	Export Flight	11
6	Import Flight	13
7	Loading Screen	15
8	Report Screen	17
9	Startup Screen	23
10	Tracking Screen	25
11	Verify Setup Screen	29
12	MatPlot Graph	33
13	Graph Test	35
14	Import File Test	39
15	Indices and tables	41
Py	thon Module Index	43
Inc	dex	45

ONE

PROGRAM CONTROLLER

class src.Controllers.Program_Controller.Controller(phoneControl):

Con-

trollers.PhoneController.PhoneControl)

Controller class for the application. Changes between application views based on user input. Run this file in order to begin the application.

${\tt cleanup}() \to {\tt None}$

Goes through our file structure and deletes all files (except anything in the Flight folder). This is for when we are done with the program and want to delete the raw footage from the phone, the output points from the opency processing and any other files that were used during the program execution. Should also be called before any time the user wants to fly.

Returns None

```
get_all_files(folderPath: str) \rightarrow list
```

Will get all of the file names in a folder. This will be used after the video footage is transferred from the phone to the laptop. We need to grab the 2 videos filenames in the folder in order to send them to the opency analysis to have the coordinates extracted. This will also be used when we need to splice together all of the coordinates output by the OpenCVController json files.

Parameters folderPath – The directory that we need to get all of the files from.

Returns A list of all of the file names in that directory

get_flight_info ($pilotName: str, instructorName: str, flightInstructions: str) <math>\rightarrow$ None Saves the pilot name, instructor name, and flight instructions once confirmed by the user.

Parameters

- pilotName String containing the pilot name
- **instructorName** String containing the instructor name
- **flightInstructions** String containing the flight instructions

Returns None

```
get in order (files: list) \rightarrow list
```

Will return the file names in order. This means that phone-1 will be first in the list and phone-2 will be second. Need this to ensure that a specific phone's coordinates are being used for the Z, X axis and the other is used for Z, Y

Parameters files - The list of files we get from the get_all_files() function

Returns The list of file names starting with phone-1 first and phone-2 second

$import_flight(flightPath: str) \rightarrow None$

Reads in a .flight file and displays the report view for it.

Parameters flightPath – String with file path of chosen file to import.

Returns None

$setupFileStructure() \rightarrow None$

Sets up the folders that need to exist before we can transfer footage and analyze. The file structure should be: drone-tracker > FTP, opency-output, Flights. This makes it easier to keep track of the files that we are working with during the application lifecycle.

Returns None

show home () \rightarrow None

Loads the home startup screen for the user.

Returns None

$show_loading_window() \rightarrow None$

Loads the loading screen for the user.

Returns None

 $show_report_window$ (previousFlight: str, usingPreviousFlight: bool, flightData: dict) \rightarrow None Loads the report screen for the user.

Parameters

- **previousFlight** String containing path to flight data. Should be .flight file if using-PreviousFlight is true, or empty if usingPreviousFlight is false.
- usingPreviousFlight Boolean representing if the report view is for an existing .flight file or a new analysis.
- **flightData** Dictionary containing the flight data. Should be populated with only coordinates if usingPreviousFlight is false, and empty if usingPreviousFlight is true.

Returns None

$\verb"show_tracking_window"() \to None$

Loads the tracking screen for the user.

Returns None

$\verb"show_verify_screen"() \to None$

Loads the verify setup screen for the user.

Returns None

$start_analysis() \rightarrow None$

Spawns the sub processes that will analyze the footage of the drone footage. We will need to have the files before hand so we can pass them into each OpenCVController process.

Returns None

$transfer_complete(flightData: dict) \rightarrow None$

Calls the report view using the flight data dictionary.

Parameters flightData - Dictionary of flight data

Returns none.

transfer_footage (phoneControl: Controllers.PhoneController.PhoneControl) \rightarrow None Transfers footage and calls DroneController to analyze the footage.

Parameters phoneControl – Phone Controller object for the active phone connection.

$updateFlightStatus() \rightarrow None$

Sets the status of the system verification test.

Returns none

$\textbf{wait_for_analysis} \, (\,) \, \to None$

Waits for the analysis of the footage to complete. Essentially is just a loop that checks to see if the file locks (*.lock) for our files are still in that directory. If they are, we wait, otherwise, we will exit the loop. From there, we need to get the output from those processes and splice the points together into a single 3D coordinate list.

Returns None

 $src. \texttt{Controllers.Program_Controller.close_conn} \ (phone Control: \ Controllers.Phone Controller.Phone Control) \\ \rightarrow \texttt{None}$

Closes the connection from the laptop to the phones.

Parameters phoneControl – PhoneControl object containing the active connection.

Returns None.

Parameters portNo – Port number we are going to be listening for signals from the phone over.

Returns PhoneControl object.

 $\texttt{src.Controllers.Program_Controller.main}\,() \to None$ Begins the main application.

TWO

OPENCY

class src.Controllers.OpenCVThreadedController.DroneTracker(videoFile)

This class is intended to track sUASs in video recorded from the smartphone app All containing code to track the drone and output the coordinates extracted from the recorded video is contained within this class, and this file is intended to be run as a separate process so that both of the recordings can be processed in parallel (don't try this on Windows though).

$is_light_on(frame) \rightarrow bool$

Takes in a video frame and returns the frame at which the light turns on.

Parameters frame – A single video frame to see if the light is on.

Returns True if the light is on, false otherwise.

read video() \rightarrow None

This function is to be threaded, and its purpose is to read in the video file all at once to improve performance

$rescale_frame (frame, percent=50)$

Resizes a frame as a percentage of the original frame size

Parameters

- **frame** the frame to be resized
- percent the percent value the frame needs to be rescaled to

$resize_bbox(bbox: tuple, factor=2) \rightarrow tuple$

Resizes the bounding box for translating it to the full size video. In order to be able to see enough of the footage on screen to draw the box around the drone, the video frame must be resized, so the drawn bounding box must be translated back into the coordinate system the full size video uses. For example, if the 4k footage is shrunk by 50% (to 1080p), the scale factor here must be 2 so the coordinates chosen in the 1080p frame will match up with the actual drone coordinates in the 4k frame.

Parameters

- **bbox** bounding box of selected drone, which is (x, y, box_width, box_height)
- factor the factor by which to scale the bounding box

Returns tuple

$\texttt{trackDrone}\,(\,)\,\rightarrow list$

Function that contains all code to track the drone, and is to be run as a thread. Will run much slower if 2 processes running this method are started and run on different videos at the same time.

Returns List of tuples of the extracted coordinates of the footage, in the format [(time, x_coord, y_coord, z_coord)].

 $\textbf{exception} \ \texttt{src.Controllers.OpenCVThreadedController.VideoCorruptedException} \ (\textit{message:} \\$

str)

This error is raised if the video being read is corrupted, or if the frames cannot be successfully extracted from the video files

exception src.Controllers.OpenCVThreadedController.VideoNotPresentException(message:

str)

This error is raised when the video for processing is not there, or if an incorrect path is given

src.Controllers.OpenCVThreadedController.get_phone_id (filename: str) \rightarrow str Gets the phone Id from the end of the file name so we can keep track of the json and lock files.

Parameters filename - The file name of the video file. Should have "phone-#.mp4" file names.

Returns The ID of the phone from the file name

 $src.Controllers.OpenCVThreadedController.main(filename: str) <math>\rightarrow None$

Will take the filename passed in and analyze the footage. All coordinates of the drone in the footage will be output to a json file.

Returns None

```
src.Controllers.OpenCVThreadedController.merge_data_points (phonelPoints: list, phone2Points: list)  \rightarrow \text{dict}
```

Takes the points outputted by the opency analysis and merges the points together to create the 3D coordinates needed to output the visual flight path.

Parameters

- **phone1Points** The opency datapoints created from the main method of this class for the first phone
- **phone2Points** The opency datapoints created from the main method of this class for the second phone

Returns List of tuples of coordinates and time values that represent the flight path of the drone in the format [(time, x_coord, y_coord, z_coord)]

THREE

EXCEPTIONS

- **exception** src.Controllers.Exceptions.**FailedDisconnectException** (*message: str*)

 This error is for telling us that something went wrong when we tried to disconnect from the RPI.
- **exception** src.Controllers.Exceptions.**FailedRPIFlashException** (*message: str*) This error is for letting us know that the RPI did not flash the light.
- **exception** src.Controllers.Exceptions.**PhonesNotSyncedException** (*message: str*)

 This exception is for when the user tries to perform an operation with the phones without actually syncing the phones first.
- **exception** src.Controllers.Exceptions.RPINotConnectedException (*message: str*) This class is for letting us know that we had an issue connecting to the raspberry pi.
- exception src.Controllers.Exceptions.RecordingNotStartedException(message:

str)

This exception class is for alerting the user that the recording has not started and they are trying to access a function that requires the phone cameras to be rolling.

exception src.Controllers.Exceptions.TransferNotStartedException(message:

str)

This exception class is for alerting the user that the file transfer process has not been started and any actions that depend on it will fail.

FOUR

PHONECONTROLLER

class src.Controllers.PhoneController.PhoneControl (portNum: int)

This class is for communicating with and controlling the phones out in the field. It uses simple TCP connections with each phone in order to control them.

${\tt closeConn}$ () $\to {\tt None}$

Closes all of the connections and the socket. :return: None

$isTransferring() \rightarrow bool$

A flag for us to access to see if the system is still waiting for the video to finish the file transfer of the videos it recorded.

Returns True if the video has finished transferring, false otherwise.

$setupSocket() \rightarrow None$

Creates the socket that we will use to listen for incoming connections.

Returns None

$startFileTransfer(filepath: str) \rightarrow None$

Will send a signal to the phone that tells it to transfer the video files it recorded over to the laptop by opening an FTP connection.

Parameters filepath – The file path on our laptop that the phones will need to send their videos to over FTP.

Returns None

$startRecording() \rightarrow None$

Call this in order to send a signal to the phones that they need to start recording.

Returns None

$\textbf{stopRecording()} \rightarrow None$

Call this in order to send a signal to the phones that they need to stop tracking. Sends both phones a stop signal and the name of the file path that they will need to send their videos to over FTP.

Returns None

$sync() \rightarrow None$

This function will wait until both phones have been connected to this app.

Returns None

$synced() \rightarrow bool$

The getter function for seeing if the phones synced or not.

Returns True if they have been synced, false otherwise

threadSendSignal (conn: socket.socket, signal: str, sigMessage: str, sigAck: str) \rightarrow None

This function takes a signal and the expected output so that we don't have to rewrite the same code for every action we have with the phones.

Parameters

- **conn** A socket connection to a phone that has already been opened.
- signal The Signal we want to send to the phone. Valid options are: START, STOP, and START FTP
- **sigMessage** A message that we want to send alongside the signal for the phone to use.
- **sigAck** The Signal we expect to get back from the phone in response to our signal. Valid options are: START_ACKNOWLEDGE, STOP_ACKNOWLEDGE, START_FTP_ACKNOWLEDGE.

Returns None

$\textbf{threadWaitForFileTransfer} (\textit{conn: socket.socket}) \rightarrow None$

This is a thread for waiting for the signal from the phone that the file transfer to the filepath specified in the startFileTransfer() function arguments.

Parameters conn – A socket connection to a phone that has already been opened.

Returns None

$waitForFileTransfer() \rightarrow None$

Spawns threads that will wait for both phones to send a signal saying that the file transfer of the videos is complete.

FIVE

EXPORT FLIGHT

 $src. \texttt{Export.ExportFile.export_data} \ (\textit{pilotName}, \textit{instructorName}, \textit{flightDate}, \textit{flightLength}, \textit{flightInstructions}, \textit{xCoordinates}, \textit{yCoordinates}, \textit{yCoordinates}, \textit{velocityValues}, \textit{outPath}) \rightarrow \texttt{None} \\ \texttt{Exports} \ \text{the flight data to a JSON file stored with a ".flight" extension}.$

Parameters

- pilotName String containing the pilot name
- instructorName String containing the instructor name
- flightDate String containing the flight date
- flightLength String containing the flight length
- flightInstructions String containing the flight instructions
- xCoordinates Array of x coordinates
- yCoordinates Array of y coordinates
- zCoordinates Array of z coordinates
- velocityValues Array of velocity values
- outPath String containing the path to save the file. Should end in ".flight".

SIX

IMPORT FLIGHT

 $\verb|src.Export.ImportFile.importData| (inPath)| \to dict \\ Imports the flight data from a JSON file stored with a ".flight" extension.$

Parameters inPath – String containing the pilot name.

Returns Flight dictionary

SEVEN

LOADING SCREEN

class src.Views.View_LoadingScreen.LoadingWindow

The view for the loading page that is shown when the user presses the "Stop Tracking" button on the tracking window page.

Variables __btnHome - The class property for the 'Return to Home' button.

property BtnHome

The home for the view. Is used to return to home screen.

Returns None

property BtnTestReport

The test report for the view. Is used to switch to the test report screen.

Returns The reference to the test report button.

property LblStatus

Getter property for the timer label. We need to attach a QTimer to it so it can count the time the application has been tracking the drone.

Returns The timer label

property del_BtnHome

The home for the view. Is used to return to home screen.

Returns None

property del BtnTestReport

The test report for the view. Is used to switch to the test report screen.

Returns The reference to the test report button.

property del_LblStatus

Getter property for the timer label. We need to attach a QTimer to it so it can count the time the application has been tracking the drone.

Returns The timer label

$initView() \rightarrow None$

Sets up the view and lays out all of the components.

Returns None

$\texttt{returnHome}() \rightarrow None$

Sends a signal to the main controller that the Cancel and Return to Home button was pushed.

Returns none

setSubtitle() → PyQt5.QtWidgets.QLabel

Sets up the subtitle label.

Returns The subtitle label

$setTitle() \rightarrow PyQt5.QtWidgets.QLabel$

Sets up the title with the application title on top and the name of the screen just below it.

Returns Layout with the application title and screen title labels

property set_BtnHome

The home for the view. Is used to return to home screen.

Returns None

property set_BtnTestReport

The test report for the view. Is used to switch to the test report screen.

Returns The reference to the test report button.

property set_LblStatus

Getter property for the timer label. We need to attach a QTimer to it so it can count the time the application has been tracking the drone.

Returns The timer label

setupLoadingIcon() → PyQt5.QtWidgets.QLabel

Used for configuring the loading icon on the loading screen. Loading icon is a gif, so QMovie is used to animate the icon.

Returns The icon containing the loading label.

$\textbf{signalTestReport} \; () \; \rightarrow None$

Sends a signal to the main controller that the Test Report button was pushed. NOTE: ONLY USED FOR TESTING PURPOSES

Returns none

$\textbf{signalTransferFootage} \, (\,) \, \to None$

Sends a signal to the main controller that the button to transfer footage was pressed.

EIGHT

REPORT SCREEN

The view for the report page that is shown when the user opens the application.

Variables

- __btnExport The class property for the 'Export Results' button.
- __btnFlyAgain The class property for the 'Fly Again' button.
- __btnHome The class property for the 'Return to Home' button.
- __btnViewGraphVelocity The class property for the 'View Flight Path' button.
- <u>__btnViewGraphNoVelocity</u> The class property for the 'View Flight Path with Velocity Changes' button.
- __btnViewInstructions The class property for the 'View Flight Instructions' button.

property BtnExport

The export button for the view.

Returns None

property BtnFlyAgain

The fly again button for the view.

Returns None

property BtnHome

The home for the view. Is used to return to home screen.

Returns None

property BtnViewGraphNoVelocity

The home for the view graph without velocity button.

Returns None

property BtnViewGraphVelocity

The home for the view graph with velocity button.

Returns None

property LblFlightDate

Getter for the flight date label.

Returns Reference to the flight date label.

property LblFlightInstructions

Getter for the flight instructions label.

Returns Reference to the flight length label.

property LblFlightLength

Getter for the flight length label.

Returns Reference to the flight length label.

property LblInstructor

Getter for the instructor label.

Returns Reference to the instructor label.

property LblPilot

Getter for the Pilot label so we can set who the pilot is for the flight in child class.

Returns Reference to the pilot label.

$analyzeFlight(flightDict: dict) \rightarrow dict$

Analyzes the flight data to extract coordinates, velocity values, and statistics.

Parameters flightDict - Dictionary of flight data, with only coordinates populated.

Returns Updated dictionary, with legal points and flight statistics included.

$\textbf{createStatisticsTable} () \rightarrow PyQt5.QtWidgets.QTableWidget$

Creates a table containing flight statistics.

Returns QTableWidget containing flight statistics.

property del_BtnExport

The export button for the view.

Returns None

property del_BtnFlyAgain

The fly again button for the view.

Returns None

property del_BtnHome

The home for the view. Is used to return to home screen.

Returns None

property del_BtnViewGraphNoVelocity

The home for the view graph without velocity button.

Returns None

property del_BtnViewGraphVelocity

The home for the view graph with velocity button.

Returns None

property del_LblFlightDate

Getter for the flight date label.

Returns Reference to the flight date label.

property del_LblFlightInstructions

Getter for the flight instructions label.

Returns Reference to the flight length label.

property del_LblFlightLength

Getter for the flight length label.

Returns Reference to the flight length label.

property del_LblInstructor

Getter for the instructor label.

Returns Reference to the instructor label.

property del_LblPilot

Getter for the Pilot label so we can set who the pilot is for the flight in child class.

Returns Reference to the pilot label.

$handleEndSliderValueChange(value) \rightarrow None$

Listener that will updated the stop time label when the user moves the stop time slider.

Parameters value – The value that the stop slider has been moved to horizontally.

Returns None

$handleStartSliderValueChange(value) \rightarrow None$

Listener that will updated the start time label when the user moves the start time slider.

Parameters value – The value that the start slider has been moved to horizontally.

Returns None

initView (pilotName: str, instructorName: str, flightInstructions: str, previousFlight: str, usingPreviousFlight: bool, flightData: dict) → None
Sets up the view and lays out all of the components.

Parameters

- pilotName String containing pilot name
- instructorName String containing instructor name
- **flightInstructions** String containing flight instructions.
- **previousFlight** String containing path to flight data. Should be .flight file if using-PreviousFlight is true, or blank if usingPreviousFlight is false.
- usingPreviousFlight Boolean denoting if the report should be populated from the same file or a different one.
- **flightDict** Dictionary containing flight data. Should be empty if usingPrevious-Flight is true.

Returns None

setButtonLayout () → PyQt5.QtWidgets.QHBoxLayout

Lays out the 'Export Results', 'Fly Again' and 'Import Previous Flight' buttons into a horizontal layout to be put on screen.

Returns The horizontal layout containing the 3 buttons

$\textbf{setSubTitle} (\textit{text}) \rightarrow PyQt5.QtWidgets.QLabel$

Sets up a subtitle label for the window

Parameters text – String as name for label

Returns Subtitle label

property set_BtnExport

The export button for the view.

Returns None

property set_BtnFlyAgain

The fly again button for the view.

Returns None

property set_BtnHome

The home for the view. Is used to return to home screen.

Returns None

property set_BtnViewGraphNoVelocity

The home for the view graph without velocity button.

Returns None

property set_BtnViewGraphVelocity

The home for the view graph with velocity button.

Returns None

property set_LblFlightDate

Getter for the flight date label.

Returns Reference to the flight date label.

property set_LblFlightInstructions

Getter for the flight instructions label.

Returns Reference to the flight length label.

property set_LblFlightLength

Getter for the flight length label.

Returns Reference to the flight length label.

property set_LblInstructor

Getter for the instructor label.

Returns Reference to the instructor label.

property set_LblPilot

Getter for the Pilot label so we can set who the pilot is for the flight in child class.

Returns Reference to the pilot label.

$setupFlightInfo() \rightarrow PyQt5.QtWidgets.QGridLayout$

Sets up the flight info (pilot, instructor, date, length, and smoothness score) in a grid.

Returns Grid layout of the flight information

$setupGraph (flightData: dict, displayVelocity: bool) \rightarrow None$

Sets up the 3d plot for viewing upon click of button. displayVelocity is a boolean denoting if the graph should display colored segments for velocity.

Parameters

- flightData Dictionary of flight data
- **displayVelocity** Bool denoting if velocity should be plotted or not.

Returns None

setupSlider() → PyQt5.QtWidgets.QVBoxLayout

Setups the slider that will be used to adjust the times of the flight path that will be displayed on the graph. This lets us control the beginning and ending bounds of the time of the flight that we want to view. We

have to use 2 sliders. One for the start time and one for the end. We originally wanted to make this such that both sliders were on top of each other to make it more intuitive, but we ran into a bug that prevented us from doing that.

Returns A horizontal layout containing both sliders and the labels displaying the time that they represent.

$setupTitle() \rightarrow PyQt5.QtWidgets.QVBoxLayout$

Sets up the title with the application title on top and the name of the screen just below it.

Returns Layout with the application title and screen title labels

${\tt showWindow}\,() \to None$

Takes all of the elements from the view and displays the window.

Returns None

$\textbf{signalExportResults}\,(\,)\,\rightarrow None$

Sends a signal to the main controller that the Export Results button was pushed.

Returns none

$\texttt{signalReturnHome}\,(\,)\,\to None$

Sends a signal to the main controller that the Return Home button was pushed.

Returns none

$signalStartTracking() \rightarrow None$

Sends a signal to the main controller that the Fly Again button was pushed.

NINE

STARTUP SCREEN

class src.Views.View_StartupScreen.StartupWindow (flightModeEnabled: bool)

The view for the home Startup page that is shown when the user opens the application.

Variables

- __btnVerifySetup The class property for the 'Verify Setup' button.
- __btnStart The class property for the 'Start Tracking' button.
- __btnImport The class property for the 'Import Previous Flight' button.

property BtnImport

Getter for the Import Previous Flight button. Is used to import past flight files. Use to attach functionality.

Returns None

property BtnStart

Getter for the startTracking button. Use to attach functionality.

Returns None

property BtnVerifySetup

Getter for the verifySetup button. Use to attach functionality.

Returns The reference to the verifySetup button

property del_BtnImport

Getter for the Import Previous Flight button. Is used to import past flight files. Use to attach functionality.

Returns None

property del_BtnStart

Getter for the startTracking button. Use to attach functionality.

Returns None

property del_BtnVerifySetup

Getter for the verifySetup button. Use to attach functionality.

Returns The reference to the verifySetup button

$initView() \rightarrow None$

Sets up the view and lays out all of the components.

Returns None

${\tt openFileNameDialog()} \to None$

Allows user to select a .flight file from a file dialog window.

Returns Path to selected file as a string.

setButtonLayout () → PyQt5.QtWidgets.QHBoxLayout

Lays out the 'Test Config', 'Start' and 'Import' buttons into a horizontal layout to be put on screen.

Returns The horizontal layout containing the 3 buttons

setTeamMembers() → PyQt5.QtWidgets.QVBoxLayout

Sets up the team members label for the window

Returns Team members label of the application

$setTitle() \rightarrow PyQt5.QtWidgets.QVBoxLayout$

Sets up the title with the application title on top and the name of the screen just below it.

Returns Layout with the application title and screen title labels

property set_BtnImport

Getter for the Import Previous Flight button. Is used to import past flight files. Use to attach functionality.

Returns None

property set_BtnStart

Getter for the startTracking button. Use to attach functionality.

Returns None

property set_BtnVerifySetup

Getter for the verifySetup button. Use to attach functionality.

Returns The reference to the verifySetup button

$setupAMLogo() \rightarrow None$

Used for configuring the display for the A&M logo on the startup screen.

Returns None

$setupPicture() \rightarrow None$

Used for configuring the display for the logo on the startup screen.

Returns None

$signalImportFlight() \rightarrow None$

Calls function to allow user to select a file for import. Sends a signal to the main controller that the Import Previous Flight button was pushed.

Returns None.

$\textbf{signalStartTracking}\,(\,)\,\rightarrow None$

Sends a signal to the main controller that the Start Tracking button was pushed.

Returns none

$signalVerifySetup() \rightarrow None$

Sends a signal to the main controller that the Verify Setup button was pushed.

TEN

TRACKING SCREEN

class src.Views.View_TrackingScreen.TrackingWindow(phoneControl:

Con-

trollers. Phone Controller. Phone Control)

The view for the tracking view page that is shown when the user presses the "Start Tracking" button on the home page. Allows the user to enter in flight information and begin tracking the drone.

Variables

- __btnConfirm The class property for the 'Confirm' button.
- __btnClear The class property for the 'Clear' button.
- __btnStart The class property for the 'Start Tracking' button.
- __btnStop The class property for the 'Stop Tracking' button.

property BtnClear

Getter for the Clear button so we can attach functionality to it later.

Returns Reference to the clear button

property BtnConfirm

Getter for the Confirm button so we can attach functionality to it later.

Returns Reference to the confirm button

property BtnStart

Getter for the Start button

Returns Reference to the start button

property BtnStop

Getter for the Stop button

Returns Reference to the stop button

property LblInstructor

Getter for the Instructor label so we can attach functionality to it later.

Returns The instructor label

property LblPilot

Getter for the Pilot label so we can attach functionality to it

Returns The pilot label

property LblTimer

Getter property for the timer label. We need to attach a QTimer to it so it can count the time the application has been tracking the drone.

Returns The timer label

property TBInstructor

Getter for the Instructor textbox so we can attach functionality to it later.

Returns The instructor textbox

property TBPilot

Getter for the Pilot Textbox so we can attach functionality to it

Returns The pilot textbox

property TEInstructions

Getter for the instructions text edit box

Returns Reference to the instructions text edit box

$clearValues() \rightarrow None$

Clears the values in the text boxes.

Returns None

$confirmValues() \rightarrow None$

Confirms the values in the textboxes by displaying a pop up message of the values.

Returns None

property del_BtnClear

Getter for the Clear button so we can attach functionality to it later.

Returns Reference to the clear button

property del_BtnConfirm

Getter for the Confirm button so we can attach functionality to it later.

Returns Reference to the confirm button

property del_BtnStart

Getter for the Start button

Returns Reference to the start button

property del_BtnStop

Getter for the Stop button

Returns Reference to the stop button

property del LblInstructor

Getter for the Instructor label so we can attach functionality to it later.

Returns The instructor label

property del_LblPilot

Getter for the Pilot label so we can attach functionality to it

Returns The pilot label

property del_LblTimer

Getter property for the timer label. We need to attach a QTimer to it so it can count the time the application has been tracking the drone.

Returns The timer label

property del_TBInstructor

Getter for the Instructor textbox so we can attach functionality to it later.

Returns The instructor textbox

property del_TBPilot

Getter for the Pilot Textbox so we can attach functionality to it

Returns The pilot textbox

property del_TEInstructions

Getter for the instructions text edit box

Returns Reference to the instructions text edit box

$initView() \rightarrow None$

Initializes and lays out all of the controls and elements on the view.

Returns None

$returnHome() \rightarrow None$

Sends a signal to the main controller that the Return Home button was pushed.

Returns none

setClrConfirmBtns() → PyQt5.QtWidgets.QHBoxLayout

Sets the buttons for clearing and confirming the pilot, instructor, and flight instruction information.

Returns The confirmation button

$\texttt{setFlightInstructions}() \rightarrow PyQt5.QtWidgets.QVBoxLayout$

Sets the textbox that will allow the instructor to type in the flight instructions for the pilot to try to match.

Returns A vertical layout with the Instructions label on top of the text box

$\textbf{setInstructor} () \rightarrow PyQt5.QtWidgets.QVBoxLayout$

Sets up the instructor label and the textbox that will be used to set the instructor flying during this session.

Returns Returns a vertical layout with the instructor label over the instructor textbox

$\mathtt{setPilot}() \rightarrow PyQt5.QtWidgets.QVBoxLayout$

Sets up the Pilot label and the textbox that will be used to set the pilot flying during this session.

Returns Returns a vertical layout with the pilot label over the pilot textbox

setStartAndStopBtns() → PyQt5.QtWidgets.QHBoxLayout

Sets up the start and stop buttons for tracking the drones.

Returns

setStatusLabel (*text*) → PyQt5.QtWidgets.QLabel

Sets up a status label for the window

Returns Label of the application taken from the "text" parameter

setSubTitle (text) \rightarrow PyQt5.QtWidgets.QLabel

Sets up a subtitle label for the window

Returns Subtitle of the application taken from the "text" parameter

$\textbf{setTimerLabel} \ () \ \rightarrow PyQt5.QtWidgets.QVBoxLayout$

Sets the label that will continuously update and display the time that the application has been actively tracking. Need to attach a QTimer() to it.

Returns The label that will contain the timer.

$\texttt{setTitle}() \rightarrow PyQt5.QtWidgets.QVBoxLayout$

Sets up the title with the application title on top and the name of the screen just below it.

Returns Layout with the application title and screen title labels

property set_BtnClear

Getter for the Clear button so we can attach functionality to it later.

Returns Reference to the clear button

property set_BtnConfirm

Getter for the Confirm button so we can attach functionality to it later.

Returns Reference to the confirm button

property set_BtnStart

Getter for the Start button

Returns Reference to the start button

property set_BtnStop

Getter for the Stop button

Returns Reference to the stop button

property set_LblInstructor

Getter for the Instructor label so we can attach functionality to it later.

Returns The instructor label

property set_LblPilot

Getter for the Pilot label so we can attach functionality to it

Returns The pilot label

property set_LblTimer

Getter property for the timer label. We need to attach a QTimer to it so it can count the time the application has been tracking the drone.

Returns The timer label

property set_TBInstructor

Getter for the Instructor textbox so we can attach functionality to it later.

Returns The instructor textbox

property set_TBPilot

Getter for the Pilot Textbox so we can attach functionality to it

Returns The pilot textbox

property set_TEInstructions

Getter for the instructions text edit box

Returns Reference to the instructions text edit box

$\textbf{startTracking}\,(\,)\,\to None$

Sends a signal to the main controller that the Start Tracking button was pushed.

Returns none

$stopTracking() \rightarrow None$

Sends a signal to the main controller that the Stop Tracking button was pushed.

ELEVEN

VERIFY SETUP SCREEN

class src.Views.View_VerifySetupScreen.VerifySetupWindow(phoneControl: Con-

trollers. Phone Controller. Phone Control)

The view for the verify setup page that is shown when the user presses the "Verify Setup" button on the home page.

Variables

- __btnPhoneSync The class property for the 'Phone Sync' button.
- __btnTestLight The class property for the 'Test Light' button.
- __btnTestFull The class property for the 'Test Full Setup' button.
- __btnCheck The class property for the 'Check Status' button.
- __btnHome The class property for the 'Return to Home' button.

property BtnCheck

The check status button for the view so we can attach functionality to it later on. Is used to check the status of the set up procedures.

Returns None

property BtnHome

The home for the view. Is used to return to home screen.

Returns None

property BtnPhoneSync

The phone sync button so we can attach functionality to it later on.

Returns The reference to the phoneSync button

property BtnTestFull

The test full setup button for the view so we can attach functionality to it later on. Is used to import past flight files.

Returns None

property BtnTestLight

The test light button so we can attach functionality to it later on.

Returns None

$checkStatus() \rightarrow None$

Shows the status of the system setup.

property del_BtnCheck

The check status button for the view so we can attach functionality to it later on. Is used to check the status of the set up procedures.

Returns None

property del_BtnHome

The home for the view. Is used to return to home screen.

Returns None

property del_BtnPhoneSync

The phone sync button so we can attach functionality to it later on.

Returns The reference to the phoneSync button

property del_BtnTestFull

The test full setup button for the view so we can attach functionality to it later on. Is used to import past flight files.

Returns None

property del_BtnTestLight

The test light button so we can attach functionality to it later on.

Returns None

$initView() \rightarrow None$

Sets up the view and lays out all of the components.

Returns None

$\texttt{returnHome}() \rightarrow None$

Sends a signal to the main controller that the Return Home button was pushed.

Returns none

setButtonLayout() → PyQt5.QtWidgets.QHBoxLayout

Lays out the 'Phone Sync', 'Test Light' and 'Test Full Setup' buttons into a horizontal layout to be put on screen.

Returns The horizontal layout containing the 3 buttons

$\texttt{setTitle}() \rightarrow PyQt5.QtWidgets.QVBoxLayout$

Sets up the title with the application title on top and the name of the screen just below it. :return: Layout with the application title and screen title labels

property set_BtnCheck

The check status button for the view so we can attach functionality to it later on. Is used to check the status of the set up procedures.

Returns None

property set_BtnHome

The home for the view. Is used to return to home screen.

Returns None

property set_BtnPhoneSync

The phone sync button so we can attach functionality to it later on.

Returns The reference to the phoneSync button

property set_BtnTestFull

The test full setup button for the view so we can attach functionality to it later on. Is used to import past flight files.

Returns None

property set_BtnTestLight

The test light button so we can attach functionality to it later on.

Returns None

$\textbf{setupPicture} () \rightarrow PyQt5.QtWidgets.QLabel$

Used for configuring the display for the logo on the screen.

Returns Returns the label that contains our logo so it can be put on the main panel.

 $\textbf{syncPhone} \ (\) \ \to None$

Runs phone sync test.

Returns None

 $\texttt{testFull}() \rightarrow None$

Runs full system test.

Returns None

 $\textbf{testLight} \; () \; \rightarrow None$

Runs light test.

TWELVE

MATPLOT GRAPH

src.Views.Graph.checkCoordinates (flightDict: dict)

Reads the input dictionary of flight data from a .flight file.

Parameters flightDict - Dictionary containing flight data.

Returns Dictionary of flight data.

src.Views.Graph.checkLegalInput(x, y, z)

Checks if the inputted 3D coordinate is within legal bounds. Legal bounds are: x, y, z > 0 and x < 15 and y < 15 and z < 10.

Parameters

- \mathbf{x} x value to check.
- y y value to check.
- $\mathbf{z} \mathbf{z}$ value to check.

Returns A boolean denoting if legal or not (true if legal, false if outside bounds).

src. Views. Graph. computeVelocity (x1, y1, z1, x2, y2, z2, t1, t2)

Computes the velocity of the drone between two points (x_1,y_1,z_1) and (x_2,y_2,z_2) at respective times t1 and t2.

Returns A float value representing the velocity of the drone.

src.Views.Graph.computeVelocityStatistics(flightDict: dict)

Computes statistics on the velocity points.

Parameters flightDict - Dictionary of flight data

Returns Updated dictionary.

 $\mathtt{src.Views.Graph.dimensionless_jerk}$ (movement: list, fs: int) \rightarrow float

Calculates the dimensionless jerk for a 1 dimensional array of points.

Parameters

- movement The numpy array of points to calculate the jerk for. The array containing
 the movement speed profile. Doesn't need to be numpy array but it MUST at least be a 1
 dimensional list.
- **fs** The sampling frequency of the data points.

Returns The dimensionless jerk estimate of the given movement's smoothness.

src.Views.Graph.generateGraph (flightData: dict, displayVelocity: bool, t1: float, t2: float)
Driver function for generating the 3d graph of drone coordinates.

Parameters

- **flightData** Dictionary containing flight data.
- **displayVelocity** Boolean saying if velocity changes should be displayed on the graph.
- t1 Minimum time bound for plotting coordinates.
- t2 Maximum time bound for plotting coordinates.

Returns The figure to display as the 3d graph.

src.Views.Graph.log_dimensionless_jerk (movement: list, fs: int) \rightarrow float Calculates the smoothness metric for the given speed profile using the log dimensionless jerk metric.

Parameters

- movement The numpy array of points to calculate the jerk for. The array containing
 the movement speed profile. Doesn't need to be numpy array but it MUST at least be a 1
 dimensional list.
- **fs** The sampling frequency of the data points.

Returns The dimensionless jerk estimate of the given movement's smoothness.

src.Views.Graph.velocityColors(flightDict: dict)

Determines the color of the line segment between two points based on velocity values. The line color is determined by the change in velocity of the drone between two points. The color of the line should be green if the velocity point is greater than the previous velocity point, yellow if within 0.5 m/s, and red if less.

Parameters flightDict - Dictionary of flight data.

Returns An array of color values to use when plotting line segments between points.

src.Views.Graph.velocityPoints(flightData: dict)

Calculates the velocity of the drone between consecutive points for the entire flight.

Parameters flightData – Dictionary of flight Data.

Returns Modified flightData dictionary.

THIRTEEN

GRAPH TEST

class src.Tests.Graph_Test.Graph_Test (methodName='runTest')

This class is for testing the graphing functions that we are using to display the flight path onto the UI. We want to make sure that what is being displayed is correct.

$\texttt{test_checkLegalInput}\,(\,)\,\to None$

Test that legal and illegal coordinate points can be detected.

Returns None

$test_computeVelocity() \rightarrow None$

Test that the velocity is computed as expected between two points (1,1,1) and (3,3,1) with timeDiff = 1.

Returns None

$test_graphShows_noError() \rightarrow None$

Test that graph generates correctly with and with velocity changes shown for data set of 100, then 200, then 800, then 1200 data points. For each size, two tests are run. One test contains all legal inputs. Another test contains 80% legal inputs. Illegal coordinate points in file should not be included in "legalPoints" list in dictionary.

Returns None

$\texttt{test_readCoordinates_size100_allLegal}\:(\:) \: \to None$

Test that file containing 100 (x,y,z) points is read in correctly. This test contains all legal inputs. Illegal coordinate points in file should not be included in "legalPoints" list in dictionary.

Returns None

test readCoordinates size100 someLegal() \rightarrow None

Test that file containing 100 (x,y,z) points is read in correctly. This test contains 80% legal inputs. Illegal coordinate points in file should not be included in "legalPoints" list in dictionary.

Returns None

$\texttt{test_readCoordinates_size1200_allLegal}\:(\:) \: \to None$

Test that file containing 1200 (x,y,z) points is read in correctly. This contains all legal inputs. Illegal coordinate points in file should not be included in "legalPoints" list in dictionary.

Returns None

$\texttt{test_readCoordinates_size1200_someLegal}\:(\:) \: \to None$

Test that file containing 1200 (x,y,z) points is read in correctly. This contains 80% legal inputs. Illegal coordinate points in file should not be included in "legalPoints" list in dictionary.

Returns None

$test_readCoordinates_size200_allLegal() \rightarrow None$

Test that file containing 200 (x,y,z) points is read in correctly. This contains all legal inputs. Illegal coordinate points in file should not be included in "legalPoints" list in dictionary.

Returns None

$test_readCoordinates_size200_someLegal() \rightarrow None$

Test that file containing 200 (x,y,z) points is read in correctly. This contains 80% legal inputs. Illegal coordinate points in file should not be included in "legalPoints" list in dictionary.

Returns None

$\texttt{test_readCoordinates_size600_allLegal}\:(\:)\:\to None$

Test that file containing 600 (x,y,z) points is read in correctly. This contains all legal inputs. Illegal coordinate points in file should not be included in "legalPoints" list in dictionary.

Returns None

$\texttt{test_readCoordinates_size600_someLegal}\:(\:)\:\to None$

Test that file containing 600 (x,y,z) points is read in correctly. This contains 80% legal inputs. Illegal coordinate points in file should not be included in "legalPoints" list in dictionary.

Returns None

$\texttt{test_readCoordinates_small_illegal} \ () \ \to None$

Test that coordinates from a small data file are read in correctly.

Returns None

$test_readCoordinates_small_legal() \rightarrow None$

Test that coordinates from a small data file are read in correctly.

Returns None

$\texttt{test_smoothnessComputes}\,() \, \to None$

Test that smoothness function returns a number when inputted data set of 100, then 200, then 800, then 1200 data points. For each size, two tests are run. One test contains all legal inputs. Another test contains 80% legal inputs. Illegal coordinate points in file should not be included in "legalPoints" list in dictionary.

Returns None

$test_smoothnessValues() \rightarrow None$

Test that smoothness function returns expected number.

Returns None

$\textbf{test_velocityColors}\,(\,)\,\to None$

Test that the correct color is assigned to the graph segment between consecutive velocity values.

Returns None

$\verb|test_velocityColorsComputes_correctSize|() \rightarrow None$

Test that velocityColors returns array of correct size when inputted data set of 100, then 200, then 800, then 1200 data points. For each size, two tests are run. One test contains all legal inputs. Another test contains 80% legal inputs. Illegal coordinate points in file should not be included in "legalPoints" list in dictionary.

Returns None

$\texttt{test_velocityComputes_correctSize}\:(\:) \: \to None$

Test that velocityPoints returns array of correct size when inputted data set of 100, 200, then 800, then 1200 data points. For each size, two tests are run. One test contains all legal inputs. Another test contains 80% legal inputs. Illegal coordinate points in file should not be included in "legalPoints" list in dictionary.

Returns None

 $\mbox{\bf test_velocityPoints} \mbox{\rm ()} \rightarrow \mbox{None}$ Test that velocity between consecutive points is computed as expected.

Returns None

CHAPTER

FOURTEEN

IMPORT FILE TEST

class src.Tests.ImportFile_Test.ImportFileTests (methodName='runTest')

This test class is used the test the import and export functions to ensure we can successfully save and reload past flights.

$\textbf{test_import}\;(\,)\;\to None$

Test that .flight file generated from exporting flight data can be imported successfully. All data members should exist, and no extra keys should be in the file.

Returns None

CHAPTER

FIFTEEN

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

S

44 Python Module Index

INDEX

A	BtnViev	vGraphVeloci	ity()		
analyzeFlight()(src.Views.View_ReportScreen.	ortWindow	(src.Views.View_ property), 17	_ReportScreen.R	eportWindow	
В	С				
BtnCheck() (src. Views. View_VerifySetupScreen. VerifySetupScreen. Ver		33			
BtnClear() (src. Views. View_Tracking Screen. Tracking W property), 25		33			
BtnConfirm() (src. Views. View_TrackingScreen. Tracking property) 25		method), 29			tupWindo
BtnExport() (src.Views.View_ReportScreen.ReportWin property), 17		method), 1			
BtnFlyAgain() (src. Views. View_ReportScreen.ReportV property), 17		method), 26	iews.View_Track		indow
BtnHome() (src.Views.View_LoadingScreen.LoadingWin property), 15		src.Controllers.I	(in Program_Contro		
BtnHome () (src.Views.View_ReportScreen.ReportWindow property), 17		method), 9			l
BtnHome () (src.Views.View_VerifySetupScreen.VerifySetupproperty), 29		33		.Views.Graph),	
BtnImport() (src.Views.View_StartupScreen.StartupWin		src.Views.Graph	ı), 33	(in module	
BtnPhoneSync() (src. Views. View_VerifySetupScreen. VerifySetupScreen.	'erif9SetilipW	gwadwes () (sro method), 26	c.Views.View_Tra	ackingScreen.Tracking	gWindow
BtnStart() (src.Views.View_StartupScreen.StartupWindproperty), 23	$d\delta$ wontrol	ller <i>src.Controllers.l</i>	(class Program_Contro	in eller), 1	
BtnStart() (src.Views.View_TrackingScreen.TrackingWproperty), 25	<i>Vindow</i> ateI	PhoneConnect src.Controllers.	tion() (i Program_Contro		
BtnStop() (src.Views.View_TrackingScreen.TrackingWingproperty), 25	<i>ndow</i> eateS	StatisticsTa (<i>src.Views.View</i> _	able() <i>_ReportScreen.R</i>	eportWindow	
BtnTestFull() (src.Views.View_VerifySetupScreen.Ver property), 29	Ь				
BtnTestLight() (src. Views. View_VerifySetupScreen. VerifySetupScreen.			Views.View_Verij	fySetupScreen.VerifyS	SetupWind
BtnTestReport() (src.Views.View_LoadingScreen.Loaproperty), 15			Views.View_Trac	ckingScreen.Tracking	Window
BtnVerifySetup() (src.Views.View_StartupScreen.StartupScre			rc.Views.View_T	rackingScreen.Tracki	ngWindow
BtnViewGraphNoVelocity() (src.Views.View_ReportScreen.ReportWindow property) 17	del_Btr	property), 26		portScreen.ReportWii	

```
del_BtnFlyAgain()
                                                             property), 19
        (src.Views.View ReportScreen.ReportWindow
                                                    del LblPilot() (src. Views. View Tracking Screen. Tracking Window
        property), 18
                                                             property), 26
del_BtnHome () (src.Views.View_LoadingScreen.LoadingWindow) lStatus () (src.Views.View_LoadingScreen.LoadingWindow)
        property), 15
                                                             property), 15
del BtnHome() (src.Views.View ReportScreen.ReportWindowLblTimer() (src.Views.View TrackingScreen.TrackingWindow
                                                             property), 26
        property), 18
del BtnHome() (src. Views. View Verify Setup Screen. Verify Setup Window ructor()
        property), 30
                                                             (src.Views.View TrackingScreen.TrackingWindow
del_BtnImport() (src.Views.View_StartupScreen.StartupWindowproperty), 26
                                                    del_TBPilot() (src.Views.View_TrackingScreen.TrackingWindow
        property), 23
del_BtnPhoneSync()
                                                             property), 26
        (src.Views.View_VerifySetupScreen.VerifySetupWindtedw_TEInstructions()
        property), 30
                                                             (src. Views. View_Tracking Screen. Tracking Window
del_BtnStart() (src. Views. View_StartupScreen. StartupWindow property), 27
        property), 23
                                                    dimensionless_jerk()
                                                                                      (in
                                                                                                module
del_BtnStart() (src. Views. View_Tracking Screen. Tracking Windowrc. Views. Graph), 33
                                                    DroneTracker
                                                                                 (class
        property), 26
                                                                                                    in
del_BtnStop() (src.Views.View_TrackingScreen.TrackingWindowsrc.Controllers.OpenCVThreadedController),
        property), 26
del_BtnTestFull()
        (src. Views. View_VerifySetupScreen. VerifySetupWindow
        property), 30
                                                    export data() (in module src.Export.ExportFile), 11
del BtnTestLight()
        (src.Views.View_VerifySetupScreen.VerifySetupWindow
        property), 30
                                                    FailedDisconnectException, 7
del_BtnTestReport()
                                                    FailedRPIFlashException, 7
        (src.Views.View_LoadingScreen.LoadingWindow
        property), 15
                                                    G
del_BtnVerifySetup()
                                                    generateGraph() (in module src. Views. Graph), 33
        (src. Views. View_StartupScreen.StartupWindow
                                                    get all files()(src.Controllers.Program Controller.Controller
        property), 23
                                                             method), 1
del_BtnViewGraphNoVelocity()
                                                    get_flight_info()
        (src.Views.View_ReportScreen.ReportWindow
                                                             (src.Controllers.Program_Controller.Controller
        property), 18
                                                             method), 1
del BtnViewGraphVelocity()
                                                    get_in_order() (src.Controllers.Program_Controller.Controller
        (src.Views.View ReportScreen.ReportWindow
                                                             method), 1
        property), 18
                                                    get_phone_id()
                                                                                                module
del_LblFlightDate()
                                                             src.Controllers.OpenCVThreadedController),
        (src.Views.View_ReportScreen.ReportWindow
        property), 18
                                                    Graph_Test (class in src.Tests.Graph_Test), 35
del LblFlightInstructions()
        (src.Views.View ReportScreen.ReportWindow
                                                    Н
        property), 18
                                                    handleEndSliderValueChange()
del_LblFlightLength()
                                                             (src.Views.View ReportScreen.ReportWindow
        (src.Views.View_ReportScreen.ReportWindow
                                                             method), 19
        property), 18
                                                    handleStartSliderValueChange()
del_LblInstructor()
                                                             (src.Views.View_ReportScreen.ReportWindow
        (src.Views.View_ReportScreen.ReportWindow
                                                             method), 19
        property), 19
del_LblInstructor()
        (src.Views.View_TrackingScreen.TrackingWindow
                                                    import flight()(src.Controllers.Program Controller.Controller
        property), 26
del LblPilot() (src. Views. View ReportScreen. ReportWindow
```

```
importData() (in module src.Export.ImportFile), 13
ImportFileTests
                                                                        (class
                                                                                                                       openFileNameDialog()
                   src.Tests.ImportFile Test), 39
                                                                                                                                           (src.Views.View_StartupScreen.StartupWindow
initView() (src.Views.View_LoadingScreen.LoadingWindow
                                                                                                                                           method), 23
                   method), 15
initView() (src. Views. View ReportScreen. ReportWindov P
                   method), 19
                                                                                                                                                                                           (class
                                                                                                                        PhoneControl
                                                                                                                                                                                                                                      in
initView() (src. Views. View Startup Screen. Startup Window
                                                                                                                                           src.Controllers.PhoneController), 9
                    method), 23
initView() (src. Views. View_Tracking Screen. Tracking Window) PhonesNotSyncedException, 7
                   method), 27
method), 30
is_light_on() (src.Controllers.OpenCVThreadedController.DroneTracker_
                                                                                                                         RecordingNotStartedException, 7
                   method), 5
ReportWindow isTransferring() (src.Controllers.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController.PhoneController
                                                                                                                                                                                           (class
                                                                                                                                                                                                                                      in
                                                                                                                                            src. Views. View_ReportScreen), 17
                    method), 9
                                                                                                                        rescale_frame()(src.Controllers.OpenCVThreadedController.DroneT
                                                                                                                                           method), 5
L
\verb|resize_bbox()| (src.Controllers.OpenCVThreadedController.DroneTracellers.OpenCVThreadedControllers.OpenCVThreadedController.DroneTracellers.OpenCVThreadedControllers.OpenCVThreadedControllers.OpenCVThreadedControllers.OpenCVThreadedControllers.OpenCVThreadedControllers.OpenCVThreadedControllers.OpenCVThreadedControllers.OpenCVThreadedControllers.OpenCVThreadedControllers.OpenCVThreadedControllers.OpenCVThreadedControllers.OpenCVThreadedControllers.OpenCVThreadedControllers.OpenCVThreadedControllers.OpenCVThreadedControllers.OpenCVThreadedControllers.OpenCVThreadedControllers.OpenCVThreadedControllers.OpenCVThreadedControllers.OpenCVThreadedControllers.OpenCVThreadedControllers.OpenCVThreadedControllers.OpenCVThreadedControllers.OpenCVThreadedControllers.OpenCVThreadedControllers.OpenCVThreadedControllers.OpenCVThreadedControllers.OpenCVThreadedControllers.OpenCVThreadedControllers.OpenCVThreadedControllers.OpenCVThreadedControllers.OpenCVThreadedControllers.OpenCVThreadedControllers.OpenCVThreadedControllers.OpenCVThreadedControllers.OpenCVThreadedControllers.OpenCVThreadedControllers.OpenCVThreadedControllers.OpenCVThreadedControllers.OpenCVThreadedControllers.OpenCVThreadedControllers.OpenCVThreadedControllers.OpenCVThreadedControllers.OpenCVThreadedControllers.OpenCVThreadedControllers.OpenCVThreadedControllers.OpenCVThreadedControllers.OpenCVThreadedControllers.OpenCVThreadedControllers.OpenCVThreadedControllers.OpenCVThreadedControllers.OpenCVThreadedControllers.OpenCVThreadedControllers.OpenCVThreadedControllers.OpenCVThreadedControllers.OpenCVThreadedControllers.OpenCVThreadedControllers.OpenCVThreadedControllers.OpenCVThreadedControllers.OpenCVThreadedControllers.OpenCVThreadedControllers.OpenCVThreadedControllers.OpenCVThreadedControllers.OpenCVThreadedControllers.OpenCVThreadedControllers.OpenCVThreadedControllers.OpenCVThreadedControllers.OpenCVThreadedControllers.OpenCVThreadedControllers.OpenCVThreadedControllers.OpenCVThreadedControllers.OpenCVThreadedControllers.OpenCVThreadedControllers.Open
                                                                                                                                          method), 5
                   property), 17
                                                                                                                        returnHome()(src.Views.View_LoadingScreen.LoadingWindow
LblFlightInstructions()
                                                                                                                                           method), 15
                    (src.Views.View_ReportScreen.ReportWindow
                                                                                                                       returnHome() (src. Views. View Tracking Screen. Tracking Window
                   property), 17
                                                                                                                                           method), 27
LblFlightLength()
                                                                                                                       returnHome() (src. Views. View VerifySetupScreen. VerifySetupWindow
                    (src.Views.View_ReportScreen.ReportWindow
                                                                                                                                            method), 30
                   property), 18
                                                                                                                       RPINotConnectedException, 7
LblInstructor() (src. Views. View Report Screen. Report Window
                   property), 18
LblInstructor() (src.Views.View_TrackingScreen.TrackingWindow_set_BtnCheck() (src.Views.View_VerifySetupScreen.VerifySetupWindow_set_BtnCheck()
                                                                                                                                           property), 30
LblPilot() (src.Views.View_ReportScreen.ReportWindow
                                                                                                                        "set_BtnClear()(src.Views.View_TrackingScreen.TrackingWindow
                   property), 18
                                                                                                                                           property), 27
LblPilot() (src. Views. View Tracking Screen. Tracking Window
                                                                                                                                   _BtnConfirm()(src.Views.View_TrackingScreen.TrackingWindow
                   property), 25
                                                                                                                                           property), 28
LblStatus() (src. Views. View_Loading Screen. Loading Window
                                                                                                                                   BtnExport() (src. Views. View_ReportScreen. ReportWindow
                   property), 15
                                                                                                                                           property), 19
LblTimer() (src.Views.View_TrackingScreen.TrackingWindow
                                                                                                                         set_BtnFlyAgain()
                   property), 25
                                                                                                                                           (src. Views. View ReportScreen. ReportWindow
LoadingWindow
                                                                     (class
                                                                                                              in
                                                                                                                                           property), 20
                   src. Views. View_LoadingScreen), 15
                                                                                                                        set_BtnHome()(src.Views.View_LoadingScreen.LoadingWindow
                                                                                                    module
log_dimensionless_jerk()
                                                                                   (in
                                                                                                                                           property), 16
                   src. Views. Graph), 34
                                                                                                                       set_BtnHome() (src.Views.View_ReportScreen.ReportWindow
                                                                                                                                           property), 20
M
                                                                                                                                  _BtnHome() (src. Views. View_VerifySetupScreen. VerifySetupWindow
main() (in module src. Controllers. OpenCVThreadedController),
                                                                                                                                           property), 30
                                                                                                                        set_BtnImport()(src.Views.View_StartupScreen.StartupWindow
main() (in module src. Controllers. Program Controller),
                                                                                                                                           property), 24
                    3
                                                                                                                        set_BtnPhoneSync()
                                                                                                    module
merge_data_points()
                                                                           (in
                                                                                                                                           (src. Views. View VerifySetupScreen. VerifySetupWindow
                    src.Controllers.OpenCVThreadedController),
                                                                                                                                           property), 30
                    6
                                                                                                                        set BtnStart() (src. Views. View Startup Screen. Startup Window
                                                                                                                                           property), 24
```

```
set_BtnStart() (src.Views.View_TrackingScreen.TrackingWindow(src.Views.View_ReportScreen.ReportWindow
        property), 28
                                                             method), 19
set_BtnStop() (src.Views.View_TrackingScreen.TrackingsWilledowtonLayout()
                                                             (src.Views.View_StartupScreen.StartupWindow
        property), 28
set_BtnTestFull()
                                                             method), 23
        (src. Views. View Verify Setup Screen. Verify Setup Winder Button Layout ()
        property), 30
                                                              (src. Views. View VerifySetupScreen. VerifySetupWindow
                                                             method), 30
set BtnTestLight()
        (src. Views. View Verify Setup Screen. Verify Setup Winderta ClrConfirmBtns()
                                                             (src.Views.View_TrackingScreen.TrackingWindow
        property), 31
                                                             method), 27
set_BtnTestReport()
        (src.Views.View_LoadingScreen.LoadingWindow setFlightInstructions()
        property), 16
                                                             (src.Views.View_TrackingScreen.TrackingWindow
set_BtnVerifySetup()
                                                             method), 27
                                                     setInstructor() (src.Views.View_TrackingScreen.TrackingWindow
        (src. Views. View\_Startup Screen. Startup Window
        property), 24
                                                             method), 27
                                                     setPilot() (src.Views.View_TrackingScreen.TrackingWindow
set_BtnViewGraphNoVelocity()
        (src.Views.View ReportScreen.ReportWindow
                                                             method), 27
                                                     setStartAndStopBtns()
        property), 20
                                                             (src.Views.View TrackingScreen.TrackingWindow
set BtnViewGraphVelocity()
        (src.Views.View_ReportScreen.ReportWindow
                                                             method), 27
        property), 20
                                                     setStatusLabel() (src.Views.View_TrackingScreen.TrackingWindow
set_LblFlightDate()
                                                             method), 27
        (src.Views.View ReportScreen.ReportWindow
                                                     setSubtitle()(src.Views.View LoadingScreen.LoadingWindow
        property), 20
                                                             method), 15
set_LblFlightInstructions()
                                                     setSubTitle() (src.Views.View_ReportScreen.ReportWindow
        (src.Views.View_ReportScreen.ReportWindow
                                                             method), 19
        property), 20
                                                     \verb|setSubTitle()| (src. Views. View\_Tracking Screen. Tracking Window)|
set_LblFlightLength()
                                                             method), 27
        (src.Views.View_ReportScreen.ReportWindow
                                                     setTeamMembers() (src. Views. View_Startup Screen. Startup Window
        property), 20
                                                             method), 24
                                                     setTimerLabel() (src.Views.View_TrackingScreen.TrackingWindow
set_LblInstructor()
        (src.Views.View_ReportScreen.ReportWindow
                                                             method), 27
                                                     setTitle() (src. Views. View_Loading Screen. Loading Window
        property), 20
set_LblInstructor()
                                                             method), 16
        (src. Views. View_TrackingScreen.TrackingWindowsetTitle() (src. Views. View_StartupScreen.StartupWindow
        property), 28
                                                             method), 24
set_LblPilot() (src.Views.View_ReportScreen.ReportWimtDwitle() (src.Views.View_TrackingScreen.TrackingWindow
        property), 20
                                                             method), 27
set_LblPilot() (src.Views.View_TrackingScreen.TrackingWindowe() (src.Views.View_VerifySetupScreen.VerifySetupWindow
        property), 28
                                                             method), 30
set_LblStatus() (src. Views. View_Loading Screen. Loading Winn How ogo () (src. Views. View_Startup Screen. Startup Window
        property), 16
                                                             method), 24
set_LblTimer() (src.Views.View_TrackingScreen.TrackingWipdtoixLeStructure()
                                                             (src.Controllers.Program_Controller.Controller
        property), 28
set_TBInstructor()
                                                             method), 2
        (src.Views.View_TrackingScreen.TrackingWindowsetupFlightInfo()
                                                             (src.Views.View_ReportScreen.ReportWindow
set_TBPilot() (src.Views.View_TrackingScreen.TrackingWindowmethod), 20
                                                     setupGraph() (src.Views.View_ReportScreen.ReportWindow
        property), 28
set_TEInstructions()
                                                             method), 20
        (src.Views.View_TrackingScreen.TrackingWindowsetupLoadingIcon()
        property), 28
                                                             (src.Views.View LoadingScreen.LoadingWindow
setButtonLayout()
                                                             method), 16
```

```
setupPicture() (src. Views. View StartupScreen. StartupWindow 9
        method), 24
                                                   src.Controllers.Program Controller (mod-
setupPicture() (src.Views.View VerifySetupScreen.VerifySetupWinhedpyW
                                                   src.Export.ExportFile (module), 11
        method), 31
setupSlider() (src. Views. View_ReportScreen. ReportWindow Export. ImportFile (module), 13
                                                  src.Tests.Graph Test (module), 35
        method), 20
setupSocket() (src.Controllers.PhoneController.PhoneControllers.ImportFile Test (module), 39
                                                   src. Views. Graph (module), 33
        method), 9
setupTitle()(src.Views.View_ReportScreen.ReportWindow.Views.View_LoadingScreen(module), 15
                                                   src.Views.View_ReportScreen (module), 17
        method), 21
show_home() (src.Controllers.Program_Controller.Controllers.Views.View_StartupScreen (module), 23
                                                  src.Views.View_TrackingScreen (module), 25
        method), 2
show_loading_window()
                                                   src.Views.View_VerifySetupScreen (mod-
        (src.Controllers.Program_Controller.Controller
                                                           ule), 29
        method), 2
                                                  start_analysis()(src.Controllers.Program_Controller.Controller
show_report_window()
                                                           method), 2
        (src.Controllers.Program_Controller.Controller startFileTransfer()
                                                           (src. Controllers. Phone Controller. Phone Control
        method), 2
show_tracking_window()
                                                           method), 9
        (src.Controllers.Program Controller.Controller startRecording() (src.Controllers.PhoneController.PhoneControl
        method), 2
                                                           method), 9
show_verify_screen()
                                                   startTracking() (src.Views.View_TrackingScreen.TrackingWindow
        (src. Controllers. Program\_Controller. Controller
                                                           method), 28
                                                  StartupWindow
                                                                                (class
                                                                                                 in
        method), 2
showWindow()(src.Views.View_ReportScreen.ReportWindow
                                                           src. Views. View StartupScreen), 23
                                                   stopRecording()(src.Controllers.PhoneController.PhoneControl
        method), 21
signalExportResults()
                                                           method), 9
        (src.Views.View_ReportScreen.ReportWindow
                                                  stopTracking() (src.Views.View_TrackingScreen.TrackingWindow
        method), 21
                                                           method), 28
                                                  sync() (src. Controllers. Phone Controller. Phone Control
signalImportFlight()
        (src. Views. View_StartupScreen.StartupWindow
                                                           method), 9
        method), 24
                                                  synced() (src.Controllers.PhoneControl
signalReturnHome()
        (src.Views.View_ReportScreen.ReportWindow
                                                  syncPhone() (src.Views.View_VerifySetupScreen.VerifySetupWindow
        method), 21
                                                           method), 31
signalStartTracking()
                                                  T
        (src.Views.View ReportScreen.ReportWindow
        method), 21
                                                  TBInstructor() (src.Views.View_TrackingScreen.TrackingWindow
signalStartTracking()
                                                           property), 25
        (src.Views.View_StartupScreen.StartupWindow
                                                  TBPilot() (src.Views.View_TrackingScreen.TrackingWindow
        method), 24
                                                           property), 26
signalTestReport()
                                                  TEInstructions() (src. Views. View_Tracking Screen. Tracking Window
        (src.Views.View LoadingScreen.LoadingWindow
                                                           property), 26
        method), 16
                                                  test_checkLegalInput()
signalTransferFootage()
                                                           (src.Tests.Graph Test.Graph Test
                                                                                            method),
        (src.Views.View_LoadingScreen.LoadingWindow
                                                           35
        method), 16
                                                  test computeVelocity()
signalVerifySetup()
                                                           (src.Tests.Graph_Test.Graph_Test
                                                                                            method),
        (src. Views. View_StartupScreen.StartupWindow
        method), 24
                                                  test graphShows noError()
src.Controllers.Exceptions (module), 7
                                                           (src.Tests.Graph Test.Graph Test
                                                                                            method).
src.Controllers.OpenCVThreadedController
        (module), 5
                                                  test_import() (src.Tests.ImportFile_Test.ImportFileTests
src.Controllers.PhoneController (module),
                                                           method), 39
```

```
test readCoordinates size100 allLegal()
                                                                                                            (src.Controllers.Program Controller.Controller
                (src.Tests.Graph_Test.Graph_Test method), 35
                                                                                                            method), 2
test_readCoordinates_size100_someLegal()TransferNotStartedException,7
               (src.Tests.Graph_Test.Graph_Test method), 35
test_readCoordinates_size1200_allLegal() \mathsf{U}
               (src.Tests.Graph Test.Graph Test method), 35
                                                                                            updateFlightStatus()
test readCoordinates size1200 someLegal()
                                                                                                            (src.Controllers.Program Controller.Controller
                (src.Tests.Graph_Test.Graph_Test method), 35
                                                                                                            method), 2
test readCoordinates size200 allLegal()
                (src.Tests.Graph_Test.Graph_Test method), 35
test_readCoordinates_size200_someLegal()velocityColors()(in module src.Views.Graph), 34
                (src.Tests.Graph_Test.Graph_Test method), 36
                                                                                             velocityPoints() (in module src. Views. Graph), 34
test_readCoordinates_size600_allLegal()
                                                                                            VerifySetupWindow
                                                                                                                                                       (class
               (src.Tests.Graph_Test.Graph_Test method), 36
                                                                                                            src. Views. View_VerifySetupScreen), 29
\verb|test_readCoordinates_size600_someLegal()| \\ VideoCorruptedException, 5
                (src.Tests.Graph_Test.Graph_Test method), 36
                                                                                            VideoNotPresentException, 6
test_readCoordinates_small_illegal()
                                                                                            W
               (src.Tests.Graph Test.Graph Test method), 36
test_readCoordinates_small_legal()
                                                                                             wait_for_analysis()
                (src.Tests.Graph Test.Graph Test method), 36
                                                                                                            (src.Controllers.Program Controller.Controller
test_smoothnessComputes()
                                                                                                            method), 3
               (src.Tests.Graph_Test.Graph_Test
                                                                           method),
                                                                                             waitForFileTransfer()
               36
                                                                                                            (src. Controllers. Phone Controller. Phone Control
test smoothnessValues()
                                                                                                            method), 10
                (src.Tests.Graph_Test.Graph_Test
                                                                           method),
test_velocityColors()
               (src.Tests.Graph\_Test.Graph\_Test
                                                                           method),
test_velocityColorsComputes_correctSize()
                (src.Tests.Graph_Test.Graph_Test method), 36
test_velocityComputes_correctSize()
                (src.Tests.Graph_Test.Graph_Test method), 36
test_velocityPoints()
               (src.Tests.Graph_Test.Graph_Test
                                                                           method),
testFull() (src. Views. View VerifySetupScreen. VerifySetupWindow
               method), 31
testLight()(src.Views.View_VerifySetupScreen.VerifySetupWindow
               method), 31
threadSendSignal()
                (src.Controllers.PhoneController.PhoneControl
               method), 9
threadWaitForFileTransfer()
               (src. Controllers. Phone Controller. Phone Controllers. Phone Contro
               method), 10
trackDrone() (src.Controllers.OpenCVThreadedController.DroneTracker
               method), 5
{\tt TrackingWindow}
                                                       (class
                                                                                     in
               src. Views. View_TrackingScreen), 25
transfer_complete()
               (src. Controllers. Program\_Controller. Controller
               method), 2
transfer_footage()
```