Survivor Buddy 3.0

Release 0.1.0

Ben Shiller, Joseph Duran, Philip Rettenmaier, Yara Mohamed, Ki

Apr 28, 2020

CONTENTS:

CHAPTER

ONE

APPLICATION

class Application.**Application**(*master*, ***kwargs*)

The main GUI class

__init__(*master*, ***kwargs*)

The constructor for the Application class

Parameters **master** – the Tk parent widget

create_widgets()

Creates the widgets seen in the GUI

close_app()

Closes the GUI application

create_menu(*root_menu*)

Creates the main GUI menu

Parameters **root_menu** – The root menu (self.menu_bar) that is instantiated in create_widgets()

refresh_devices()

Refreshes the Devices menu

connect(*dev*)

Connects to the given device

Parameters **dev** – The serial device to connect to

close()

Closes the active serial connection

hello()

A test function

Simply prints "Hello from Menu" to the console and the NotificationsFrame

CHAPTER

TWO

CONTROLBUTTONS

class ControlButtons.**ControlButtons**(*master*, *arm_controller*, *notifications*, ***kwargs*)

Buttons to control the Survivor Buddy 3.0 arm

__init__(*master*, *arm_controller*, *notifications*, ***kwargs*)

The constructor for ControlButtons

Parameters

- **master** – The Tk parent widget
- **arm_controller** – The SerialArmController being used
- **notifications** – The NotificationsFrame being used

create_buttons()

Creates the control buttons displayed in the GUI

open_arm()

Opens the arm using SerialArmController

close_arm()

Closes the arm using SerialArmController

portrait()

Changes the arm to portrait mode using SerialArmController

landscape()

Changes the arm to landscape mode using SerialArmController

tilt()

Tilts the arm using SerialArmController

nod()

Nods the arm using SerialArmController

shake()

Shakes the arm using SerialArmController

shutdown()

Shuts down the arm using SerialArmController

CHAPTER

THREE

NOTIFICATIONSFRAME

class NotificationsFrame.**NotificationFrame**(*master*, *_logFile*, ***kwargs*)

Box to display notification in the GUI

__init__(*master*, *_logFile*, ***kwargs*)

The constructor for NotificationsFrame

Parameters

- **master** – The Tk parent widget
- **_logFile** – The file handle for the output log file

append_line(*line*)

Prints a line to the notification box and a timestamped line to the log file

CHAPTER

FOUR

POSITIONFRAME

class PositionFrame.**PositionUpdater**(*dev*, *_pitch_control*, *_yaw_control*, *_roll_control*, *_yaw_queue*, *_pitch_queue*, *_roll_queue*, *_notifications*, ***kwargs*)

Updates UI elements based on arm position

__init__(*dev*, *_pitch_control*, *_yaw_control*, *_roll_control*, *_yaw_queue*, *_pitch_queue*, *_roll_queue*, *_notifications*, ***kwargs*)

Constructor for PositionUpdater

Parameters

- **dev** – The SerialArmController
- **_pitch_control** – The pitch LabelScaleSpinbox
- **_yaw_control** – The yaw LabelScaleSpinbox
- **_roll_control** – The roll LabelScaleSpinbox
- **_yaw_queue** – The yaw queue
- **_pitch_queue** – The pitch queue
- **_roll_queue** – The roll queue
- **_notifications** – The NotificationsFrame

run()

Continually checks for changes to arm position, then updates the UI based on these changes, checks for changes every 0.1 seconds. Runs as a thread separately from rest of UI, uses queue to update render and directly updates sliders and spinboxes.

class PositionFrame.**LabelScaleSpinbox**(*master*, *text=''*, *from_=0*, *to=10*, *axis=0*, *dev=None*, ***kwargs*)

A custom class to combine Tk Scale and Spinbox and keep them in sync

__init__(*master*, *text=''*, *from_=0*, *to=10*, *axis=0*, *dev=None*, ***kwargs*)

Constructor for LabelScaleSpinbox

Parameters

- **master** – The Tk parent widget
- **text** – The text to display next to the control
- **from** – The minimum valid value
- **to** – The maximum valid value
- **axis** – The axis that this LabelScaleSpinbox controls

- **dev** – The SerialArmController

sliderUpdate(*val*)
Sends command to arm based on slider value, sets spinbox based on slider

Parameters **val** – The value passed to this function when the slider is released

validate_spinbox(*val*)
Check that spinbox and slider are within valid range of values

Parameters **val** – The value from the spinbox

invalid_spinbox()
Function that runs when the spinbox has an invalid value

set_slider()
Set slider position based on spinbox value, send command to arm

send_command()
Sends a new position to the arm based on changed axis

class PositionFrame.**RenderDiagram**(*master*, *dev=None*, ***kwargs*)
Displays a basic render of arm, helps to show position when arm can not be seen by user

__init__(*master*, *dev=None*, ***kwargs*)
Constructor for RenderDiagram

Parameters

- **master** – The Tk parent widget
- **dev** – The SerialArmController

draw_axes()
Clear units from axes, display arm base, set axis limits

update_render(*master*, *new_yaw*, *new_pitch*, *new_roll*)
Update display of render based on new arm position

Parameters

- **master** – The Tk parent widget
- **new_yaw** – The new yaw value
- **new_pitch** – The new pitch value
- **new_roll** – The new roll value

class PositionFrame.**PositionFrame**(*master*, *arm_controller*, *_logFile*, ***kwargs*)
Creates the Render and Control Sliders in the GUI

__init__(*master*, *arm_controller*, *_logFile*, ***kwargs*)
Constructor for PositionFrame

Parameters

- **master** – The Tk parent widget
- **arm_controller** – The SerialArmController
- **_logFile** – The output log file handle

create_render(*master*)
Initializes render of arm

Parameters **master** – The Tk parent widget

create_controls(*master*)
Creates LabelScaleSpinbox controls

Parameters **master** – The Tk parent widget

create_updater()
Starts updater function to update GUI based on current position

process_queue()
Processes queue of position updates

Uses this queue data to print new position to log file, and to update render position

CHAPTER

FIVE

SERIALARMCONTROLLER

class SerialArmController.**Command**
A class to keep track of command numbers

PITCH = 0

YAW = 1

ROLL = 2

CLOSE = 3

OPEN = 4

PORTRAIT = 5

LANDSCAPE = 6

NOD = 7

SHAKE = 8

TILT = 9

SHUTDOWN = 16

class SerialArmController.**Position**(*pitch=0*, *yaw=0*, *roll=0*)
A class to store position data

__init__(*pitch=0*, *yaw=0*, *roll=0*)
Constructor for Position

Parameters

- **pitch** – The pitch value
- **yaw** – The yaw value
- **roll** – The roll value

class SerialArmController.**SerialArmController**(*_status_bar*, *_notifications*)
Send commands to the robot arm and receives data from the arm

__init__(*_status_bar*, *_notifications*)
Constructor for SerialArmController

Parameters

- **_status_bar** – StatusBar to use
- **_notifications** – NotificationsFrame to use

update_devs()
Updates the list of available devices

connect(*comport*)
Connects to the device at the desired COM port

Parameters **comport** – The name of the COM port to connect to

close()
Closes the current connection

send(*data*)
Sends data to the arm

Parameters **data** – Bytes to send

recv()
Receives data from the arm

Returns data - bytes from the arm

update_position()
Updates the current stored position

set_pitch(*val*)
Sends a command to the arm to go to the desired pitch

Parameters **val** – The pitch to go to

set_yaw(*val*)
Sends a command to the arm to go to the desired yaw

Parameters **val** – The yaw to go to

set_roll(*val*)
Sends a command to the arm to go to the desired roll

Parameters **val** – The roll to go to

close_arm()
Sends the CLOSE command to the arm

open_arm()
Sends the OPEN command to the arm

portrait()
Sends the PORTRAIT command to the arm

landscape()
Sends the LANDSCAPE command to the arm

tilt()
Sends the TILT command to the arm

nod()
Sends the NOD command to the arm

shake()
Sends the SHAKE command to the arm

_shutdown()
Sends the SHUTDOWN command to the arm

CHAPTER

SIX

STATUSBAR

Created on Thu Feb 27 14:17:02 2020

@author: shill

class StatusBar.**StatusBar**(*master*, ***kwargs*)

Displays the connection status of the GUI to the arm

__init__(*master*, ***kwargs*)

Constructor for StatusBar

Parameters **master** – The Tk parent widget

set_status(*status*)

Sets the status of the GUI to the arm

Parameters **status** – The status to set to

PYTHON MODULE INDEX

INDEX

Symbols

A

C

D

H

I

L

M

N

O

P

R

S

T

U

V

Y