SE 3XA3: Module Interface Specification

Random Flag Generator

Team #2, Team Jakriel
Akram Hannoufa, hannoufa
Ganghoon (James) Park, parkg10
Nathaniel Hu, hun4

April 11, 2022

Generated by Doxygen 1.9.3 (plus some manual editing)

Table 1 Revision History

Date	Version	Notes
March 15, 2022	1.0	Initial Document
March 16, 2022	1.1	Added/updated doxygen commenting to HashGenerator.py, FlagGenerator.py and JKAReader.py modules
March 16, 2022	1.2	Added doxygen commenting to DecisionUtilites.py, FlagAssetsLib.py and HashToFlag.py modules
March 17, 2022	1.3	Added doxygen commenting to GUI.py, Settings.py, Display.py and Help.py modules
March 18, 2022	1.4	Updated doxygen commenting to various modules
April 11, 2022	2.0	Updated Module Interface Specification documentation for Revision 1 submission

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 3

Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

Chapter 5

Namespace Documentation

5.1 DecisionUtilities Namespace Reference

Functions

- def pad_hashcode (hashcode)

  Generates a padded hashcode if it is not the minimum required length.
- def choose_from_list (source_list, index)

  Generates a selection from an array.
- def map_decision (max_digitsum, num_decisions, digitsum)

  Maps a number to an index of an array.
- def split_sequence (seq, length)

  Generates a list of shorter tokens from a given input string.
- def hex2rgb (hexvalue)

  Generates a tuple of an RGB colour from a hexadecimal number.
- def diff (num1, num2)

  Calculates the absolute difference of two float values.

Variables

- int COLOR_QUANTITY = 5
- int HEX_COLOR_LEN = 6
- int HEX_BASE = 16
- int MINIMUM_HASH_LEN = COLOR_QUANTITY $*$ HEX_COLOR_LEN
- int ASPECT_CONTROL_LEN = 6
- int MAX_DECISION_VALUE = 16777215
- bool DEBUG = False

5.1.1 Function Documentation

5.1.1.1 choose_from_list()

```
def DecisionUtilities.choose_from_list (
              source_list,
              index )
```

Generates a selection from an array.

Parameters

source_list	the list to make a selection from
index	the index to take the selection.

Exceptions

ValueError	if index value is negative

Returns

choice, the option from the source_list that matches the index.

5.1.1.2 diff()

```
def DecisionUtilities.diff (
              num1,
              num2 )
```

Calculates the absolute difference of two float values.

Parameters

num1	first float value
num2	second float value

Returns

The integer value of the absolute difference between float values.

5.1.1.3 hex2rgb()

```
def DecisionUtilities.hex2rgb (
              hexvalue )
```

Generates a tuple of an RGB colour from a hexadecimal number.

hexvalue is length 6, with every 2 characters being part of the RGB tuple

Parameters

hexvalue	hexadecimal value (length=6) to convert to RGB

Exceptions

TypeError	if hexvalue is not a string

Returns

RGB tuple, representing an RGB colour in integer values

5.1.1.4 map_decision()

```
def DecisionUtilities.map_decision (
              max_digitsum,
              num_decisions,
              digitsum )
```

Maps a number to an index of an array.

Parameters

max_digitsum	the maximum possible option
num_decisions	the number of possible decisions
digitsum	the digit to map within possible options

Exceptions

ValueError	if any param is negative

Returns

decision, float index of array to get decision from.

5.1.1.5 pad_hashcode()

```
def DecisionUtilities.pad_hashcode (
              hashcode )
```

Generates a padded hashcode if it is not the minimum required length.

Input string gets padded until it is long enough to generate all required flag options.

Parameters

hashcode	a string of the input string's corresponding hashcode value.

Exceptions

TypeError	if hashcode is not string type

Returns

modified hashcode, a padded version of the input hashcode.

5.1.1.6 split_sequence()

```
def DecisionUtilities.split_sequence (
              seq,
              length )
```

Generates a list of shorter tokens from a given input string.

Created strings are of a specified size, n

Parameters

seq	input string to break apart
length	length of generated substrings

Returns

tokens, list of shorter substrings of length n

5.1.2 Variable Documentation

5.1.2.1 ASPECT_CONTROL_LEN

```
int DecisionUtilities.ASPECT_CONTROL_LEN = 6
```

5.1.2.2 COLOR_QUANTITY

```
int DecisionUtilities.COLOR_QUANTITY = 5
```

5.1.2.3 DEBUG

```
bool DecisionUtilities.DEBUG = False
```

5.1.2.4 HEX_BASE

```
int DecisionUtilities.HEX_BASE = 16
```

5.1.2.5 HEX_COLOR_LEN

```
int DecisionUtilities.HEX_COLOR_LEN = 6
```

5.1.2.6 MAX_DECISION_VALUE

```
int DecisionUtilities.MAX_DECISION_VALUE = 16777215
```

5.1.2.7 MINIMUM_HASH_LEN

```
int DecisionUtilities.MINIMUM_HASH_LEN = COLOR_QUANTITY * HEX_COLOR_LEN
```

5.2 Display Namespace Reference

Classes

- class FlagDisplay

  FlagDisplay is a class that allows the user to generate and display the flag.

5.3 FlagAssetsLib Namespace Reference

Variables

- list BASE_STRIPE_STYLES
- list OVERLAY_STRIPE_STYLES
- list STRIPE_NUMBER = ['ONE', 'TWO', 'THREE', 'SIX', 'TWELVE']
- list SYMBOL_LOCATION = ['TOP_LEFT', 'CENTER', 'TOP_RIGHT']
- list SYMBOL_NUMBER = ['ONE', 'TWO']
- list SYMBOL_TYPES = ['NONE', 'MOON', 'ROUNDEL', 'SWORD']
- dictionary colours2rgb
- dictionary low_res_flag_assets
- dictionary mid_res_flag_assets
- dictionary high_res_flag_assets

5.3.1 Variable Documentation

5.3.1.1 BASE_STRIPE_STYLES

```
list FlagAssetsLib.BASE_STRIPE_STYLES
```

Initial value:

```
=  ['NONE', 'HORIZONTAL',
                      'VERTICAL', 'CROSS', 'SALTIRE', 'CROSS_SALTIRE']
```

5.3.1.2 colours2rgb

```
dictionary FlagAssetsLib.colours2rgb
```

Initial value:

```
=  {'RED': (255, 0, 0), 'GREEN': (0, 255, 0), 'BLUE': (
    0, 0, 255), 'YELLOW': (255, 255, 0), 'PURPLE': (127, 0, 255)}
```

5.3.1.3 high_res_flag_assets

```
dictionary FlagAssetsLib.high_res_flag_assets
```

Initial value:

```
=  {"VERTICAL": {"ONE": "jka/high_res/vstripe_1.jka",
                                     "TWO": "jka/high_res/vstripe_2.jka",
                                     "THREE": "jka/high_res/vstripe_3.jka",
                                     "SIX": "jka/high_res/vstripe_6.jka",
                                     "TWELVE": "jka/high_res/vstripe_12.jka",
                                     "ONE_THIN":
                                     "jka/high_res/vstripe_1_thin.jka"},
                        "HORIZONTAL": {"ONE": "jka/high_res/hstripe_1.jka",
                                       "TWO": "jka/high_res/hstripe_2.jka",
                                        "THREE": "jka/high_res/hstripe_3.jka",
                                        "SIX": "jka/high_res/hstripe_6.jka",
                                        "TWELVE": "jka/high_res/hstripe_12.jka",
                                        "ONE_THIN":
                                        "jka/high_res/hstripe_1_thin.jka"},
                         "SALTIRE": "jka/high_res/saltire.jka",
                         "CROSS": "jka/high_res/cross.jka",
                         "CROSS_SALTIRE": "jka/high_res/cross_saltire.jka",
                         "SALTIRE_THIN": "jka/high_res/saltire_thin.jka",
                         "CROSS_THIN": "jka/high_res/cross_thin.jka",
                         "CROSS_SALTIRE_THIN":
                         "jka/high_res/cross_saltire_thin.jka",
                         "MOON": "jka/high_res/moon.jka",
                         "SWORD": "jka/high_res/sword.jka",
                         "ROUNDEL": "jka/high_res/roundel.jka"}
```

5.3.1.4 low_res_flag_assets

```
dictionary FlagAssetsLib.low_res_flag_assets
```

Initial value:

```
=  {"VERTICAL": {"ONE": "jka/low_res/vstripe_1.jka",
                                   "TWO": "jka/low_res/vstripe_2.jka",
                                   "THREE": "jka/low_res/vstripe_3.jka",
                                   "SIX": "jka/low_res/vstripe_6.jka",
                                   "TWELVE": "jka/low_res/vstripe_12.jka",
                                   "ONE_THIN":
                                   "jka/low_res/vstripe_1_thin.jka"},
                      "HORIZONTAL": {"ONE": "jka/low_res/hstripe_1.jka",
                                     "TWO": "jka/low_res/hstripe_2.jka",
                                      "THREE": "jka/low_res/hstripe_3.jka",
                                      "SIX": "jka/low_res/hstripe_6.jka",
                                      "TWELVE":
                                      "jka/low_res/hstripe_12.jka",
                                      "ONE_THIN":
                                      "jka/low_res/hstripe_1_thin.jka"},
                       "SALTIRE": "jka/low_res/saltire.jka",
                       "CROSS": "jka/low_res/cross.jka",
                       "CROSS_SALTIRE": "jka/low_res/cross_saltire.jka",
                       "SALTIRE_THIN": "jka/low_res/saltire_thin.jka",
                       "CROSS_THIN": "jka/low_res/cross_thin.jka",
                       "CROSS_SALTIRE_THIN":
                       "jka/low_res/cross_saltire_thin.jka",
                       "MOON": "jka/low_res/moon.jka",
                       "SWORD": "jka/low_res/sword.jka",
                       "ROUNDEL": "jka/low_res/roundel.jka"}
```

5.3.1.5 mid_res_flag_assets

```
dictionary FlagAssetsLib.mid_res_flag_assets
```

Initial value:

```
=  {"VERTICAL": {"ONE": "jka/mid_res/vstripe_1.jka",
                                   "TWO": "jka/mid_res/vstripe_2.jka",
                                   "THREE": "jka/mid_res/vstripe_3.jka",
                                   "SIX": "jka/mid_res/vstripe_6.jka",
                                   "TWELVE": "jka/mid_res/vstripe_12.jka",
                                   "ONE_THIN":
                                   "jka/mid_res/vstripe_1_thin.jka"},
                      "HORIZONTAL": {"ONE": "jka/mid_res/hstripe_1.jka",
                                     "TWO": "jka/mid_res/hstripe_2.jka",
                                      "THREE": "jka/mid_res/hstripe_3.jka",
                                      "SIX": "jka/mid_res/hstripe_6.jka",
                                      "TWELVE": "jka/mid_res/hstripe_12.jka",
                                      "ONE_THIN":
                                      "jka/mid_res/hstripe_1_thin.jka"},
                       "SALTIRE": "jka/mid_res/saltire.jka",
                       "CROSS": "jka/mid_res/cross.jka",
                       "CROSS_SALTIRE": "jka/mid_res/cross_saltire.jka",
                       "SALTIRE_THIN": "jka/mid_res/saltire_thin.jka",
                       "CROSS_THIN": "jka/mid_res/cross_thin.jka",
                       "CROSS_SALTIRE_THIN":
                       "jka/mid_res/cross_saltire_thin.jka",
                       "MOON": "jka/mid_res/moon.jka",
                       "SWORD": "jka/mid_res/sword.jka",
                       "ROUNDEL": "jka/mid_res/roundel.jka"}
```

5.3.1.6 OVERLAY_STRIPE_STYLES

```
list FlagAssetsLib.OVERLAY_STRIPE_STYLES
```

Initial value:

```
=  ['NONE', 'HORIZONTAL_THIN', 'VERTICAL_THIN',
                        'CROSS_THIN', 'SALTIRE_THIN', 'CROSS_SALTIRE_THIN']
```

5.3.1.7 STRIPE_NUMBER

```
list FlagAssetsLib.STRIPE_NUMBER = ['ONE', 'TWO', 'THREE', 'SIX', 'TWELVE']
```

5.3.1.8 SYMBOL_LOCATION

```
list FlagAssetsLib.SYMBOL_LOCATION = ['TOP_LEFT', 'CENTER', 'TOP_RIGHT']
```

5.3.1.9 SYMBOL_NUMBER

```
list FlagAssetsLib.SYMBOL_NUMBER = ['ONE', 'TWO']
```

5.3.1.10 SYMBOL_TYPES

```
list FlagAssetsLib.SYMBOL_TYPES = ['NONE', 'MOON', 'ROUNDEL', 'SWORD']
```

5.4 FlagGenerator Namespace Reference

Classes

- class FlagGenerator

  FlagGenerator is a class that implements and encapsulates various attributes and methods needed to generate the flag image files.

5.5 Gallery Namespace Reference

Classes

- class FlagGallery

  FlagGallery is a class that opens the gallery so that the user can look at the generated flags.

5.6 GUI Namespace Reference

Classes

- class SampleApp

  SampleApp creates the graphical user interface and allows the window to switch between different frames.
- class StartPage

  StartPage is a start page that opens once the program starts.

Variables

- app = SampleApp()

5.6.1 Variable Documentation

5.6.1.1 app

```
GUI.app = SampleApp()
```

5.7 HashGenerator Namespace Reference

Functions

- def _get_hash_algo (hash_type)

  gets the hashing algorithm from the dictionary of available hashing algorithms using the given input hash type string
- def _get_hash_hex (hash_input, hash_algo)

  gets the hexidecimal representation of the hashing digest using the given input string and hashing algorithm
- def hash_generator (hash_input, hash_type='sha256')

  generates a hashing digest using the given input string and hashing algorithm

5.7.1 Function Documentation

5.7.1.1 _get_hash_algo()

```
def HashGenerator._get_hash_algo (
              hash_type )  [private]
```

gets the hashing algorithm from the dictionary of available hashing algorithms using the given input hash type string

the default hashing algorithm, SHA-256, is used if the input hash type is not in the dictionary of available hashing algorithms

Parameters

hash_type	a string representing the selected hashing algorithm

Returns

selected hashing algorithm if found; otherwise SHA-256 hashing algorithm returned

5.7.1.2 _get_hash_hex()

```
def HashGenerator._get_hash_hex (
              hash_input,
              hash_algo )  [private]
```

gets the hexidecimal representation of the hashing digest using the given input string and hashing algorithm

the byte encoding will be specified per the Python version used

Parameters

hash_input	a string that will be run through the given hashing algorithm to get a hexidecimal hashing digest
hash_algo	a hashing algorithm that will be used to turn the input string into a hexidecimal hashing digest

Returns

hexidecimal hashing digest obtained from the given input string using the given hashing algorithm

5.7.1.3 hash_generator()

```
def HashGenerator.hash_generator (
              hash_input,
              hash_type = 'sha256' )
```

generates a hashing digest using the given input string and hashing algorithm

the hashing algorithm SHA-256 will be used by default if none is specified

Parameters

hash_input	a string that will be run through the given hashing algorithm to get a hexidecimal hashing digest
hash_type	a string representing the selected hashing algorithm

Returns

hexidecimal hashing digest obtained from the given input string using the given hashing algorithm

5.8 HashToFlag Namespace Reference

Functions

- def grind_hash_for_colors (hashcode)

  Generates the array of colours to be used in the flag generation.
- def grind_hash_for_base_stripe_style (hashcode)

  Generates the base stripe style to be used in flag generation.

- def grind_hash_for_overlay_stripe_style (hashcode)

  Generates the overlay stripe style to be used in flag generation.
- def grind_hash_for_stripe_number (hashcode)

  Generates the number of stripes to be used in flag generation.
- def grind_hash_for_symbol_locations (hashcode)

  Generates the symbol location to be used in flag generation.
- def grind_hash_for_symbol_number (hashcode)

  Generates the number of symbols to be used in flag generation.
- def grind_hash_for_symbol_types (hashcode)

  Generates the symbol type to be used in flag generation.

5.8.1 Function Documentation

5.8.1.1 grind_hash_for_base_stripe_style()

```
def HashToFlag.grind_hash_for_base_stripe_style (
              hashcode )
```

Generates the base stripe style to be used in flag generation.

Uses the first 6 characters of a hashcode to map to an array index ,ie. the option to use for the desired aspect.

Parameters

hashcode	a string of the input string's corresponding hashcode value.

Returns

A base stripe style option.

5.8.1.2 grind_hash_for_colors()

```
def HashToFlag.grind_hash_for_colors (
              hashcode )
```

Generates the array of colours to be used in the flag generation.

Hex values of the hashcode are converted to an RGB value for colour.

Parameters

hashcode	a string of the input string's corresponding hashcode value.

Returns

colors, an array of RGB values to be used by FlagGenerator.

5.8.1.3 grind_hash_for_overlay_stripe_style()

```
def HashToFlag.grind_hash_for_overlay_stripe_style (
              hashcode )
```

Generates the overlay stripe style to be used in flag generation.

Uses the first 6 characters of a hashcode to map to an array index ,ie.the option to use for the desired aspect.

Parameters

hashcode	a string of the input string's corresponding hashcode value.

Returns

An overlay stripe style option.

5.8.1.4 grind_hash_for_stripe_number()

```
def HashToFlag.grind_hash_for_stripe_number (
              hashcode )
```

Generates the number of stripes to be used in flag generation.

Uses the second 6 characters of a hashcode to map to an array index ,ie.the option to use for the desired aspect.

Parameters

hashcode	a string of the input string's corresponding hashcode value.

Returns

A stripe number option.

5.8.1.5 grind_hash_for_symbol_locations()

```
def HashToFlag.grind_hash_for_symbol_locations (
              hashcode )
```

Generates the symbol location to be used in flag generation.

Uses the third 6 characters of a hashcode to map to an array index ,ie.the option to use for the desired aspect.

Parameters

hashcode	a string of the input string's corresponding hashcode value.

Returns

A symbol location option.

5.8.1.6 grind_hash_for_symbol_number()

```
def HashToFlag.grind_hash_for_symbol_number (
              hashcode )
```

Generates the number of symbols to be used in flag generation.

Uses the fourth 6 characters of a hashcode to map to an array index ,ie.the option to use for the desired aspect.

Parameters

hashcode	a string of the input string's corresponding hashcode value.

Returns

A symbol number option.

5.8.1.7 grind_hash_for_symbol_types()

```
def HashToFlag.grind_hash_for_symbol_types (
              hashcode )
```

Generates the symbol type to be used in flag generation.

Uses the fifth 6 characters of a hashcode to map to an array index, ie.the option to use for the desired aspect.

Parameters

hashcode	a string of the input string's corresponding hashcode value.

Returns

A symbol type option.

5.9 Help Namespace Reference

Classes

- class Help

  Shows the help menu to teach the user how to use the software and how it works.

5.10 JKAReader Namespace Reference

Functions

- def parse_jka_file (filename)

  parses the input flag asset (.jka) file data into a pixel map

Variables

- string FILLED_PIXEL = "#"
- string UNFILLED_PIXEL = "."

5.10.1 Function Documentation

5.10.1.1 parse_jka_file()

```
def JKAReader.parse_jka_file (
              filename )
```

parses the input flag asset (.jka) file data into a pixel map

parses the file data by pixel and adds filled pixels to the pixel map

Parameters

filename	a string representing the name of the flag asset (.jka) file that contains the flag asset pixel map data

Returns

a list containing the (x, y) coordinates of all filled pixels for the given flag asset

5.10.2 Variable Documentation

5.10.2.1 FILLED_PIXEL

```
string JKAReader.FILLED_PIXEL = "#"
```

5.10.2.2 UNFILLED_PIXEL

```
string JKAReader.UNFILLED_PIXEL = "."
```

5.11 Settings Namespace Reference

Classes

- class Settings

  Creates settings graphical user interface app for user to use the random flag generator and set their settings.

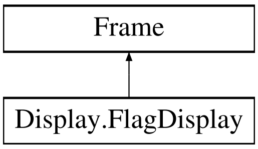
Chapter 6

Class Documentation

6.1 Display.FlagDisplay Class Reference

FlagDisplay is a class that allows the user to generate and display the flag.

Inheritance diagram for Display.FlagDisplay:



Public Member Functions

- def __init__ (self, parent, controller)

  Creates the next page after clicking the start button from the GUI main page.
- def init_window (self)

  Places the different widgets onto the second page after clicking on the start button on the GUI main page.
- def get_input_string (self)

  Gets the input string from the input text box.
- def generate (self, event)

  Generates the flag from the input string.
- def display (self, event)

  Displays the generated flag.
- def delete_image (self, event)

  Clears the displayed flag.

Public Attributes

- controller
- FG
- image_text_input
- image_generate_button
- image_display_button
- image_back_button
- image_clear_button
- resize_image_text_input
- resize_generate_button
- resize_image_display_button
- resize_image_back_button
- resize_image_clear_button
- photo_text_input
- photo_generate_button
- photo_display_button
- photo_back_button
- photo_clear_button
- label
- button_generate
- generate
- val
- button_display
- display
- button_back
- button_clear
- delete_image
- input_box
- image_display
- resized_image_display
- photo_display
- label_display
- image_empty
- resized_image_empty
- photo_empty
- label_empty

6.1.1 Detailed Description

FlagDisplay is a class that allows the user to generate and display the flag.

the FlagDisplay contains the text input box, generate button, back button. display button, and a clear button for the user to use.

6.1.2 Constructor & Destructor Documentation

6.1.2.1 __init__()

```
def Display.FlagDisplay.__init__ (
              self,
              parent,
              controller )
```

Creates the next page after clicking the start button from the GUI main page.

The GUI second page will have the text input box, generate button, display button, and a back button.

Parameters

self	Current object, common first parameter for any method of a class.
parent	A widget that acts as the parent of self, current object. All widgets in tkinter except the root window require a parent
controller	Other objects that are designed to act as a shared point, allowing several pages of widgets to interact. It decouples the different pages, making them independent. The controller descides what page will be visible.

6.1.3 Member Function Documentation

6.1.3.1 delete_image()

```
def Display.FlagDisplay.delete_image (
              self,
              event )
```

Clears the displayed flag.

The generated flag is cleared on the screen with this function

Parameters

event	When clear button is clicked, it will call this function to clear the flag being displayed.

6.1.3.2 display()

```
def Display.FlagDisplay.display (
              self,
              event )
```

Displays the generated flag.

The generated flag is displayed on the screen with this function

Parameters

event	When display button is clicked, it will call this function to display the flag.

6.1.3.3 generate()

```
def Display.FlagDisplay.generate (
              self,
              event )
```

Generates the flag from the input string.

The flag is generated with this function

Parameters

event	When generate button is clicked, it will call this function to generate the flag.

6.1.3.4 get_input_string()

```
def Display.FlagDisplay.get_input_string (
              self )
```

Gets the input string from the input text box.

The input string is retrieved with this function

6.1.3.5 init_window()

```
def Display.FlagDisplay.init_window (
              self )
```

Places the different widgets onto the second page after clicking on the start button on the GUI main page.

The second page will include the text input input box, generate button, display button, clear button, and a back button.

6.1.4 Member Data Documentation

6.1.4.1 button_back

```
Display.FlagDisplay.button_back
```

6.1.4.2 button_clear

```
Display.FlagDisplay.button_clear
```

6.1.4.3 button_display

```
Display.FlagDisplay.button_display
```

6.1.4.4 button_generate

```
Display.FlagDisplay.button_generate
```

6.1.4.5 controller

```
Display.FlagDisplay.controller
```

6.1.4.6 delete_image

```
Display.FlagDisplay.delete_image
```

6.1.4.7 display

```
Display.FlagDisplay.display
```

6.1.4.8 FG

```
Display.FlagDisplay.FG
```

6.1.4.9 generate

```
Display.FlagDisplay.generate
```

6.1.4.10 image_back_button

```
Display.FlagDisplay.image_back_button
```

6.1.4.11 image_clear_button

```
Display.FlagDisplay.image_clear_button
```

6.1.4.12 image_display

```
Display.FlagDisplay.image_display
```

6.1.4.13 image_display_button

```
Display.FlagDisplay.image_display_button
```

6.1.4.14 image_empty

```
Display.FlagDisplay.image_empty
```

6.1.4.15 image_generate_button

```
Display.FlagDisplay.image_generate_button
```

6.1.4.16 image_text_input

```
Display.FlagDisplay.image_text_input
```

6.1.4.17 input_box

```
Display.FlagDisplay.input_box
```

6.1.4.18 label

```
Display.FlagDisplay.label
```

6.1.4.19 label_display

```
Display.FlagDisplay.label_display
```

6.1.4.20 label_empty

```
Display.FlagDisplay.label_empty
```

6.1.4.21 photo_back_button

```
Display.FlagDisplay.photo_back_button
```

6.1.4.22 photo_clear_button

```
Display.FlagDisplay.photo_clear_button
```

6.1.4.23 photo_display

```
Display.FlagDisplay.photo_display
```

6.1.4.24 photo_display_button

```
Display.FlagDisplay.photo_display_button
```

6.1.4.25 photo_empty

```
Display.FlagDisplay.photo_empty
```

6.1.4.26 photo_generate_button

```
Display.FlagDisplay.photo_generate_button
```

6.1.4.27 photo_text_input

```
Display.FlagDisplay.photo_text_input
```

6.1.4.28 resize_generate_button

```
Display.FlagDisplay.resize_generate_button
```

6.1.4.29 resize_image_back_button

```
Display.FlagDisplay.resize_image_back_button
```

6.1.4.30 resize_image_clear_button

```
Display.FlagDisplay.resize_image_clear_button
```

6.1.4.31 resize_image_display_button

```
Display.FlagDisplay.resize_image_display_button
```

6.1.4.32 resize_image_text_input

```
Display.FlagDisplay.resize_image_text_input
```

6.1.4.33 resized_image_display

```
Display.FlagDisplay.resized_image_display
```

6.1.4.34 resized_image_empty

```
Display.FlagDisplay.resized_image_empty
```

6.1.4.35 val

```
Display.FlagDisplay.val
```

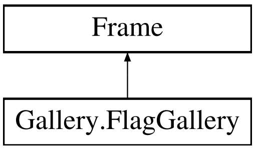
The documentation for this class was generated from the following file:

- Display.py

6.2 Gallery.FlagGallery Class Reference

FlagGallery is a class that opens the gallery so that the user can look at the generated flags.

Inheritance diagram for Gallery.FlagGallery:



Public Member Functions

- def __init__ (self, parent, controller)

  Constructor for FlagGallery GUI object.
- def init_window (self)

  method for placing buttons in FlagGallery.
- def open_gallery (self)

  method for open the gallery after clicking the open gallery button.
- def open_gallery_win (self)

  method for constructing the gallery.

Public Attributes

- controller
- image_back_button
- resize_image_back_button
- photo_back_button
- button_back
- image_open_gallery_button
- resize_image_open_gallery_button
- photo_open_gallery_button
- button_open_gallery

6.2.1 Detailed Description

FlagGallery is a class that opens the gallery so that the user can look at the generated flags.

the FlagGallery contains the open gallery button and a back button for the user to control.

6.2.2 Constructor & Destructor Documentation

6.2.2.1 __init__()

```
def Gallery.FlagGallery.__init__ (
              self,
              parent,
              controller )
```

Constructor for FlagGallery GUI object.

Creates object for GUI

Parameters

∗*args*	allows the function to accept an arbitrary number of arguments
∗∗*kwargs*	allows the function to accept an arbitrary number of keyword arguments.

6.2.3 Member Function Documentation

6.2.3.1 init_window()

```
def Gallery.FlagGallery.init_window (
              self )
```

method for placing buttons in FlagGallery.

the buttons will be placed onto the FlagGallery GUI.

6.2.3.2 open_gallery()

```
def Gallery.FlagGallery.open_gallery (
              self )
```

method for open the gallery after clicking the open gallery button.

it will close the FlagGallery frame and open the gallery window.

6.2.3.3 open_gallery_win()

```
def Gallery.FlagGallery.open_gallery_win (
              self )
```

method for constructing the gallery.

it will showcase all the generated flags by clicking buttons for next and previous flag.

6.2.4 Member Data Documentation

6.2.4.1 button_back

```
Gallery.FlagGallery.button_back
```

6.2.4.2 button_open_gallery

```
Gallery.FlagGallery.button_open_gallery
```

6.2.4.3 controller

```
Gallery.FlagGallery.controller
```

6.2.4.4 image_back_button

```
Gallery.FlagGallery.image_back_button
```

6.2.4.5 image_open_gallery_button

```
Gallery.FlagGallery.image_open_gallery_button
```

6.2.4.6 photo_back_button

```
Gallery.FlagGallery.photo_back_button
```

6.2.4.7 photo_open_gallery_button

```
Gallery.FlagGallery.photo_open_gallery_button
```

6.2.4.8 resize_image_back_button

```
Gallery.FlagGallery.resize_image_back_button
```

6.2.4.9 resize_image_open_gallery_button

```
Gallery.FlagGallery.resize_image_open_gallery_button
```

The documentation for this class was generated from the following file:

- Gallery.py

6.3 FlagGenerator.FlagGenerator Class Reference

FlagGenerator is a class that implements and encapsulates various attributes and methods needed to generate the flag image files.

Public Member Functions

- def __init__ (self)

  constructor method for FlagGenerator
- def generate_flag (self, hash_input, hash_type="SHA256", settings={"RESOLUTION":"HIGH", "BASE_↩ COLOUR":"RANDOM", "SYMBOL_COLOUR":"RANDOM", "SYMBOL_TYPE":"RANDOM"})

  generates the flag image file using the given hash input string and input hash type string
- def generate_flag_data (self, hash_input, hash_type)

  generates the flag data using the given hash input string and input hash type string

Public Attributes

- settings
- colours
- stripe_info
- symbol_info
- width
- height
- flag_image
- flag_pixels

Private Member Functions

- def _set_flag_dimensions (self)

  sets dimensions for the generated flag image
- def _set_flag_base_colour (self)

  sets base colour for the generated flag image
- def _set_flag_base_stripes (self)

  sets base stripes colour and style for the generated flag image
- def _set_flag_overlay_stripes (self)

  sets overlay stripes colour and style for the generated flag image
- def _set_flag_symbol (self)

  sets symbol colour and type for the generated flag image
- def _save_flag_image (self, hash_input)

  saves generated flag image

6.3.1 Detailed Description

FlagGenerator is a class that implements and encapsulates various attributes and methods needed to generate the flag image files.

the FlagGenerator contains the user settings and flag colours, stripe info, symbol info, dimensions, image and pixel data

6.3.2 Constructor & Destructor Documentation

6.3.2.1 __init__()

```
def FlagGenerator.FlagGenerator.__init__ (
              self )
```

constructor method for FlagGenerator

the attributes will be used to temporarily store the data needed to generate each flag image; overwritten with each flag subsequently generated

6.3.3 Member Function Documentation

6.3.3.1 _save_flag_image()

```
def FlagGenerator.FlagGenerator._save_flag_image (
              self,
              hash_input )  [private]
```

saves generated flag image

saves generated flag image as png image file to flag gallery directory, using given hash_input to name the image file

Parameters

hash_input	a string that will be used to name the generated flag image file

6.3.3.2 _set_flag_base_colour()

```
def FlagGenerator.FlagGenerator._set_flag_base_colour (
              self )  [private]
```

sets base colour for the generated flag image

sets flag image base colour based on set colour choice (or if set to RANDOM, then sets to randomly generated colour)

6.3.3.3 _set_flag_base_stripes()

```
def FlagGenerator.FlagGenerator._set_flag_base_stripes (
              self )  [private]
```

sets base stripes colour and style for the generated flag image

sets flag image base stripes colour to randomly generated colour and base stripes style to randomly chosen stripes style

6.3.3.4 _set_flag_dimensions()

```
def FlagGenerator.FlagGenerator._set_flag_dimensions (
              self )  [private]
```

sets dimensions for the generated flag image

sets flag image width and height (in pixels) based upon the set resolution

6.3.3.5 _set_flag_overlay_stripes()

```
def FlagGenerator.FlagGenerator._set_flag_overlay_stripes (
              self )  [private]
```

sets overlay stripes colour and style for the generated flag image

sets flag image overlay stripes colour to randomly generated colour and overlay stripes style to randomly chosen stripes style

6.3.3.6 _set_flag_symbol()

```
def FlagGenerator.FlagGenerator._set_flag_symbol (
              self )  [private]
```

sets symbol colour and type for the generated flag image

sets flag image symbol type based on set symbol type choice (or if set to RANDOM, then sets to randomly selected symbol type); sets flag image symbol colour based on set symbol colour choice (or if set to RANDOM, then sets to randomly generated colour)

6.3.3.7 generate_flag()

```
def FlagGenerator.FlagGenerator.generate_flag (
              self,
              hash_input,
              hash_type = "SHA256",
              settings = {"RESOLUTION": "HIGH", "BASE_COLOUR": "RANDOM",                                  "SYMB
_COLOUR": "RANDOM",                                  "SYMBOL_TYPE": "RANDOM"} )
```

generates the flag image file using the given hash input string and input hash type string

uses functions to get the flag data from the hash input string and input hash type string and stacks the flag asset layers to generate the final flag image saved to the gallery

Parameters

hash_input	a string that will be run through the given hashing algorithm to get a hexidecimal hashing digest
hash_type	a string representing the selected hashing algorithm; default hashing algorithm is sha256
settings	a dictionary representing the user selected settings; default resolution is high default colours and symbol type are randomly generated, i.e. taken from generated flag data

6.3.3.8 generate_flag_data()

```
def FlagGenerator.FlagGenerator.generate_flag_data (
              self,
              hash_input,
              hash_type )
```

generates the flag data using the given hash input string and input hash type string

uses external functions to generate the flag data from the hash input string and input hash type string

Parameters

hash_input	a string that will be run through the given hashing algorithm to get a hexidecimal hashing digest
hash_type	a string representing the selected hashing algorithm

6.3.4 Member Data Documentation

6.3.4.1 colours

```
FlagGenerator.FlagGenerator.colours
```

6.3.4.2 flag_image

```
FlagGenerator.FlagGenerator.flag_image
```

6.3.4.3 flag_pixels

```
FlagGenerator.FlagGenerator.flag_pixels
```

6.3.4.4 height

```
FlagGenerator.FlagGenerator.height
```

6.3.4.5 settings

```
FlagGenerator.FlagGenerator.settings
```

6.3.4.6 stripe_info

```
FlagGenerator.FlagGenerator.stripe_info
```

6.3.4.7 symbol_info

```
FlagGenerator.FlagGenerator.symbol_info
```

6.3.4.8 width

```
FlagGenerator.FlagGenerator.width
```

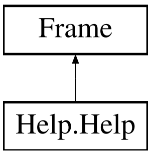
The documentation for this class was generated from the following file:

- FlagGenerator.py

6.4 Help.Help Class Reference

Shows the help menu to teach the user how to use the software and how it works.

Inheritance diagram for Help.Help:



Public Member Functions

- def __init__ (self, parent, controller)

  Creates the help page after clicking the help button from the GUI main page.
- def init_window (self)

  Places the different widgets onto the help page after clicking on the help button on the GUI main page.

Public Attributes

- controller
- image_back_button
- resize_image_back_button
- photo_back_button
- button_back
- image
- resized_image
- photo
- label

6.4.1 Detailed Description

Shows the help menu to teach the user how to use the software and how it works.

The help menu will have instructions and a button to go back to the main GUI screen.

6.4.2 Constructor & Destructor Documentation

6.4.2.1 __init__()

```
def Help.Help.__init__ (
              self,
              parent,
              controller )
```

Creates the help page after clicking the help button from the GUI main page.

The help page will have the instructions and a back button.

Parameters

self	Current object, common first parameter for any method of a class.
parent	A widget that acts as the parent of self, current object. All widgets in tkinter except the root window require a parent
controller	Other objects that are designed to act as a shared point, allowing several pages of widgets to interact. It decouples the different pages, making them independent. The controller descides what page will be visible.

6.4.3 Member Function Documentation

6.4.3.1 init_window()

```
def Help.Help.init_window (
              self )
```

Places the different widgets onto the help page after clicking on the help button on the GUI main page.

The help page will include the instructions and a back button.

6.4.4 Member Data Documentation

6.4.4.1 button_back

```
Help.Help.button_back
```

6.4.4.2 controller

```
Help.Help.controller
```

6.4.4.3 image

```
Help.Help.image
```

6.4.4.4 image_back_button

```
Help.Help.image_back_button
```

6.4.4.5 label

```
Help.Help.label
```

6.4.4.6 photo

```
Help.Help.photo
```

6.4.4.7 photo_back_button

```
Help.Help.photo_back_button
```

6.4.4.8 resize_image_back_button

```
Help.Help.resize_image_back_button
```

6.4.4.9 resized_image

```
Help.Help.resized_image
```

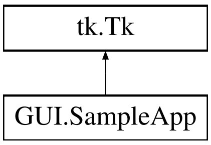
The documentation for this class was generated from the following file:

- Help.py

6.5 GUI.SampleApp Class Reference

SampleApp creates the graphical user interface and allows the window to switch between different frames.

Inheritance diagram for GUI.SampleApp:



Public Member Functions

- def __init__ (self, ∗args, ∗∗kwargs)
  Constructor for new app GUI object.
- def show_frame (self, page_name)
  Shows the frame and switches between frames.

Public Attributes

- frames

6.5.1 Detailed Description

SampleApp creates the graphical user interface and allows the window to switch between different frames.

the SampleApp contains the the start page, display, gallery, settings, and help frames.

6.5.2 Constructor & Destructor Documentation

6.5.2.1 __init__()

```
def GUI.SampleApp.__init__ (
                 self,
                ∗ args,
                ∗∗ kwargs )
```

Constructor for new app GUI object.

Creates object for GUI

Parameters

∗*args*	allows the function to accept an arbitrary number of arguments
∗∗*kwargs*	allows the function to accept an arbitrary number of keyword arguments.

6.5.3 Member Function Documentation

6.5.3.1 show_frame()

```
def GUI.SampleApp.show_frame (
              self,
              page_name )
```

Shows the frame and switches between frames.

The frame is generated and switched.

Parameters

page_name	Used to switch between frames in tkinter. The frame it will show on the window.

6.5.4 Member Data Documentation

6.5.4.1 frames

```
GUI.SampleApp.frames
```

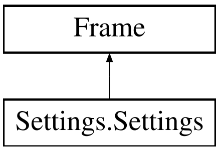
The documentation for this class was generated from the following file:

- GUI.py

6.6 Settings.Settings Class Reference

Creates settings graphical user interface app for user to use the random flag generator and set their settings.

Inheritance diagram for Settings.Settings:



Public Member Functions

- def __init__ (self, parent, controller)

  Creates the settings page after clicking the settings button from the GUI main page.
- def init_window (self)

  Places the different widgets onto the settings page after clicking on the settings button on the GUI main page.
- def get_input_string (self)

  Gets the input string from the input text box.
- def generate (self, event)

  Generates the flag from the input string.
- def display (self, event)

  Displays the generated flag.
- def delete_image (self, event)

  Clears the displayed flag.

Public Attributes

- controller
- FG
- image_text_input
- resize_image_text_input
- photo_text_input
- label_t_i
- input_box
- image_back_button
- resize_image_back_button
- photo_back_button
- button_back
- image_generate_button
- resize_image_generate_button
- photo_generate_button
- button_generate
- generate
- image_display_button
- resize_image_display_button
- photo_display_button
- button_display
- display
- image_clear_button
- resize_image_clear_button
- photo_clear_button
- button_clear
- delete_image
- image_p_c
- resize_image_p_c
- photo_p_c
- label1
- image_s_c
- resize_image_s_c
- photo_s_c
- label2
- image_f_s
- resize_image_f_s

- photo_f_s
- label3
- image_h_t
- resize_image_h_t
- photo_h_t
- label4
- image_r_t
- resize_image_r_t
- photo_r_t
- label5
- primary_colours
- secondary_colours
- flag_symbols
- hash_types
- res_types
- p_c
- s_c
- f_s
- h_t
- r_t
- image_display
- resized_image_display
- photo_display
- label_display
- image_empty
- resized_image_empty
- photo_empty
- label_empty

6.6.1 Detailed Description

Creates settings graphical user interface app for user to use the random flag generator and set their settings.

Settings graphical user interface for user to change the settings.

6.6.2 Constructor & Destructor Documentation

6.6.2.1 __init__()

```
def Settings.Settings.__init__ (
              self,
              parent,
              controller )
```

Creates the settings page after clicking the settings button from the GUI main page.

The settings page will have options for the user to change the flag setting and a back button.

Parameters

self	Current object, common first parameter for any method of a class.
parent	A widget that acts as the parent of self, current object. All widgets in tkinter except the root window require a parent
controller	Other objects that are designed to act as a shared point, allowing several pages of widgets to interact. It decouples the different pages, making them independent. The controller descides what page will be visible.

6.6.3 Member Function Documentation

6.6.3.1 delete_image()

```
def Settings.Settings.delete_image (
              self,
              event )
```

Clears the displayed flag.

The generated flag is cleared on the screen with this function

Parameters

event	When clear button is clicked, it will call this function to clear the flag being displayed.

6.6.3.2 display()

```
def Settings.Settings.display (
              self,
              event )
```

Displays the generated flag.

The generated flag is displayed on the screen with this function

Parameters

event	When display button is clicked, it will call this function to display the flag.

6.6.3.3 generate()

```
def Settings.Settings.generate (
              self,
              event )
```

Generates the flag from the input string.

The flag is generated with this function

Parameters

event	When generate button is clicked, it will call this function to generate the flag.

6.6.3.4 get_input_string()

```
def Settings.Settings.get_input_string (
              self )
```

Gets the input string from the input text box.

The input string is retrieved with this function

6.6.3.5 init_window()

```
def Settings.Settings.init_window (
              self )
```

Places the different widgets onto the settings page after clicking on the settings button on the GUI main page.

The settings page will include the options to change the flag settings, such as colours, symbols, resolutions, and a back button.

6.6.4 Member Data Documentation

6.6.4.1 button_back

```
Settings.Settings.button_back
```

6.6.4.2 button_clear

```
Settings.Settings.button_clear
```

6.6.4.3 button_display

`Settings.Settings.button_display`

6.6.4.4 button_generate

`Settings.Settings.button_generate`

6.6.4.5 controller

`Settings.Settings.controller`

6.6.4.6 delete_image

`Settings.Settings.delete_image`

6.6.4.7 display

`Settings.Settings.display`

6.6.4.8 f_s

`Settings.Settings.f_s`

6.6.4.9 FG

`Settings.Settings.FG`

6.6.4.10 flag_symbols

`Settings.Settings.flag_symbols`

6.6.4.11 generate

```
Settings.Settings.generate
```

6.6.4.12 h_t

```
Settings.Settings.h_t
```

6.6.4.13 hash_types

```
Settings.Settings.hash_types
```

6.6.4.14 image_back_button

```
Settings.Settings.image_back_button
```

6.6.4.15 image_clear_button

```
Settings.Settings.image_clear_button
```

6.6.4.16 image_display

```
Settings.Settings.image_display
```

6.6.4.17 image_display_button

```
Settings.Settings.image_display_button
```

6.6.4.18 image_empty

```
Settings.Settings.image_empty
```

6.6.4.19 image_f_s

```
Settings.Settings.image_f_s
```

6.6.4.20 image_generate_button

```
Settings.Settings.image_generate_button
```

6.6.4.21 image_h_t

```
Settings.Settings.image_h_t
```

6.6.4.22 image_p_c

```
Settings.Settings.image_p_c
```

6.6.4.23 image_r_t

```
Settings.Settings.image_r_t
```

6.6.4.24 image_s_c

```
Settings.Settings.image_s_c
```

6.6.4.25 image_text_input

```
Settings.Settings.image_text_input
```

6.6.4.26 input_box

```
Settings.Settings.input_box
```

6.6.4.27 label1

```
Settings.Settings.label1
```

6.6.4.28 label2

```
Settings.Settings.label2
```

6.6.4.29 label3

```
Settings.Settings.label3
```

6.6.4.30 label4

```
Settings.Settings.label4
```

6.6.4.31 label5

```
Settings.Settings.label5
```

6.6.4.32 label_display

```
Settings.Settings.label_display
```

6.6.4.33 label_empty

```
Settings.Settings.label_empty
```

6.6.4.34 label_t_i

```
Settings.Settings.label_t_i
```

6.6.4.35 p_c

```
Settings.Settings.p_c
```

6.6.4.36 photo_back_button

```
Settings.Settings.photo_back_button
```

6.6.4.37 photo_clear_button

```
Settings.Settings.photo_clear_button
```

6.6.4.38 photo_display

```
Settings.Settings.photo_display
```

6.6.4.39 photo_display_button

```
Settings.Settings.photo_display_button
```

6.6.4.40 photo_empty

```
Settings.Settings.photo_empty
```

6.6.4.41 photo_f_s

```
Settings.Settings.photo_f_s
```

6.6.4.42 photo_generate_button

```
Settings.Settings.photo_generate_button
```

6.6.4.43 photo_h_t

```
Settings.Settings.photo_h_t
```

6.6.4.44 photo_p_c

```
Settings.Settings.photo_p_c
```

6.6.4.45 photo_r_t

```
Settings.Settings.photo_r_t
```

6.6.4.46 photo_s_c

```
Settings.Settings.photo_s_c
```

6.6.4.47 photo_text_input

```
Settings.Settings.photo_text_input
```

6.6.4.48 primary_colours

```
Settings.Settings.primary_colours
```

6.6.4.49 r_t

```
Settings.Settings.r_t
```

6.6.4.50 res_types

```
Settings.Settings.res_types
```

6.6.4.51 resize_image_back_button

```
Settings.Settings.resize_image_back_button
```

6.6.4.52 resize_image_clear_button

```
Settings.Settings.resize_image_clear_button
```

6.6.4.53 resize_image_display_button

```
Settings.Settings.resize_image_display_button
```

6.6.4.54 resize_image_f_s

```
Settings.Settings.resize_image_f_s
```

6.6.4.55 resize_image_generate_button

```
Settings.Settings.resize_image_generate_button
```

6.6.4.56 resize_image_h_t

```
Settings.Settings.resize_image_h_t
```

6.6.4.57 resize_image_p_c

```
Settings.Settings.resize_image_p_c
```

6.6.4.58 resize_image_r_t

```
Settings.Settings.resize_image_r_t
```

6.6.4.59 resize_image_s_c

```
Settings.Settings.resize_image_s_c
```

6.6.4.60 resize_image_text_input

```
Settings.Settings.resize_image_text_input
```

6.6.4.61 resized_image_display

```
Settings.Settings.resized_image_display
```

6.6.4.62 resized_image_empty

```
Settings.Settings.resized_image_empty
```

6.6.4.63 s_c

```
Settings.Settings.s_c
```

6.6.4.64 secondary_colours

```
Settings.Settings.secondary_colours
```

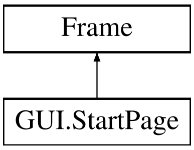
The documentation for this class was generated from the following file:

- Settings.py

6.7 GUI.StartPage Class Reference

StartPage is a start page that opens once the program starts.

Inheritance diagram for GUI.StartPage:



Public Member Functions

- def __init__ (self, parent, controller)

  Creates the start page for the GUI main page.
- def init_window (self)

  Places the different widgets onto the start page for the GUI main page.

Public Attributes

- controller
- image_app_logo
- image_start_button
- image_gallery_button
- image_settings_button
- image_help_button
- resize_image_app_logo
- resize_image_start_button
- resize_image_gallery_button
- resize_image_settings_button
- resize_image_help_button
- photo_app_logo
- photo_start_button
- photo_gallery_button
- photo_settings_button
- photo_help_button
- app_logo
- start_button
- start_generate
- gallery_button
- gallery_generate
- settings_button
- settings_generate
- help_button
- help_generate

6.7.1 Detailed Description

StartPage is a start page that opens once the program starts.

It displays the start page frame.

the StartPage contains buttons to start the flag generation, chnage the settings, open the gallery, and read the intructions.

6.7.2 Constructor & Destructor Documentation

6.7.2.1 __init__()

```
def GUI.StartPage.__init__ (
              self,
              parent,
              controller )
```

Creates the start page for the GUI main page.

The start page will have the logo, start button, settings button, gallery button, and a help button.

Parameters

self	Current object, common first parameter for any method of a class.
parent	A widget that acts as the parent of self, current object. All widgets in tkinter except the root window require a parent
controller	Other objects that are designed to act as a shared point, allowing several pages of widgets to interact. It decouples the different pages, making them independent. The controller decides what page will be visible.

6.7.3 Member Function Documentation

6.7.3.1 init_window()

```
def GUI.StartPage.init_window (
              self )
```

Places the different widgets onto the start page for the GUI main page.

The start page will include the logo, start button, settings button, gallery, and a help button.

6.7.4 Member Data Documentation

6.7.4.1 app_logo

```
GUI.StartPage.app_logo
```

6.7.4.2 controller

```
GUI.StartPage.controller
```

6.7.4.3 gallery_button

```
GUI.StartPage.gallery_button
```

6.7.4.4 gallery_generate

```
GUI.StartPage.gallery_generate
```

6.7.4.5 help_button

```
GUI.StartPage.help_button
```

6.7.4.6 help_generate

```
GUI.StartPage.help_generate
```

6.7.4.7 image_app_logo

```
GUI.StartPage.image_app_logo
```

6.7.4.8 image_gallery_button

```
GUI.StartPage.image_gallery_button
```

6.7.4.9 image_help_button

```
GUI.StartPage.image_help_button
```

6.7.4.10 image_settings_button

```
GUI.StartPage.image_settings_button
```

6.7.4.11 image_start_button

```
GUI.StartPage.image_start_button
```

6.7.4.12 photo_app_logo

```
GUI.StartPage.photo_app_logo
```

6.7.4.13 photo_gallery_button

```
GUI.StartPage.photo_gallery_button
```

6.7.4.14 photo_help_button

```
GUI.StartPage.photo_help_button
```

6.7.4.15 photo_settings_button

```
GUI.StartPage.photo_settings_button
```

6.7.4.16 photo_start_button

```
GUI.StartPage.photo_start_button
```

6.7.4.17 resize_image_app_logo

```
GUI.StartPage.resize_image_app_logo
```

6.7.4.18 resize_image_gallery_button

```
GUI.StartPage.resize_image_gallery_button
```

6.7.4.19 resize_image_help_button

```
GUI.StartPage.resize_image_help_button
```

6.7.4.20 resize_image_settings_button

```
GUI.StartPage.resize_image_settings_button
```

6.7.4.21 resize_image_start_button

```
GUI.StartPage.resize_image_start_button
```

6.7.4.22 settings_button

```
GUI.StartPage.settings_button
```

6.7.4.23 settings_generate

```
GUI.StartPage.settings_generate
```

6.7.4.24 start_button

```
GUI.StartPage.start_button
```

6.7.4.25 start_generate

```
GUI.StartPage.start_generate
```

The documentation for this class was generated from the following file:

- GUI.py

Chapter 7

File Documentation

7.1 DecisionUtilities.py File Reference

@title DecisionUtilities

Namespaces

- namespace DecisionUtilities

Functions

- def DecisionUtilities.pad_hashcode (hashcode)

  Generates a padded hashcode if it is not the minimum required length.
- def DecisionUtilities.choose_from_list (source_list, index)

  Generates a selection from an array.
- def DecisionUtilities.map_decision (max_digitsum, num_decisions, digitsum)

  Maps a number to an index of an array.
- def DecisionUtilities.split_sequence (seq, length)

  Generates a list of shorter tokens from a given input string.
- def DecisionUtilities.hex2rgb (hexvalue)

  Generates a tuple of an RGB colour from a hexadecimal number.
- def DecisionUtilities.diff (num1, num2)

  Calculates the absolute difference of two float values.

Variables

- int DecisionUtilities.COLOR_QUANTITY = 5
- int DecisionUtilities.HEX_COLOR_LEN = 6
- int DecisionUtilities.HEX_BASE = 16
- int DecisionUtilities.MINIMUM_HASH_LEN = COLOR_QUANTITY $*$ HEX_COLOR_LEN
- int DecisionUtilities.ASPECT_CONTROL_LEN = 6
- int DecisionUtilities.MAX_DECISION_VALUE = 16777215
- bool DecisionUtilities.DEBUG = False

7.1.1 Detailed Description

@title DecisionUtilities

A collection of modules used by HashToFlag to grind the hashcode and map decisions to arrays.

Author

Akram Hannoufa

Date

2022-04-11

7.2 Display.py File Reference

@title Display

Classes

- class Display.FlagDisplay

  FlagDisplay is a class that allows the user to generate and display the flag.

Namespaces

- namespace Display

7.2.1 Detailed Description

@title Display

A dispay option to display the generated flag.

Author

Ganghoon Park

Date

2022-04-11

7.3 FlagAssetsLib.py File Reference

@title FlagAssetsLib

Namespaces

- namespace FlagAssetsLib

Variables

- list FlagAssetsLib.BASE_STRIPE_STYLES
- list FlagAssetsLib.OVERLAY_STRIPE_STYLES
- list FlagAssetsLib.STRIPE_NUMBER = ['ONE', 'TWO', 'THREE', 'SIX', 'TWELVE']
- list FlagAssetsLib.SYMBOL_LOCATION = ['TOP_LEFT', 'CENTER', 'TOP_RIGHT']
- list FlagAssetsLib.SYMBOL_NUMBER = ['ONE', 'TWO']
- list FlagAssetsLib.SYMBOL_TYPES = ['NONE', 'MOON', 'ROUNDEL', 'SWORD']
- dictionary FlagAssetsLib.colours2rgb
- dictionary FlagAssetsLib.low_res_flag_assets
- dictionary FlagAssetsLib.mid_res_flag_assets
- dictionary FlagAssetsLib.high_res_flag_assets

7.3.1 Detailed Description

@title FlagAssetsLib

A library of constants and symbol/design options

Author

Akram Hannoufa, Nathaniel Hu

Date

2022-03-31

7.4 FlagGenerator.py File Reference

@title FlagGenerator

Classes

- class FlagGenerator.FlagGenerator

  FlagGenerator is a class that implements and encapsulates various attributes and methods needed to generate the flag image files.

Namespaces

- namespace FlagGenerator

7.4.1 Detailed Description

@title FlagGenerator

A module for generating the flag image using a given input string and hashing algorithm

FlagGenerator module, uses HashGenerator, HashToFlag and JKAReader modules; no exported constants or types, no state or environment variables, no state invariant or assumptions

Author

Nathaniel Hu, Akram Hannoufa

Date

2022-04-07

7.5 Gallery.py File Reference

@title Gallery

Classes

- class Gallery.FlagGallery

  FlagGallery is a class that opens the gallery so that the user can look at the generated flags.

Namespaces

- namespace Gallery

7.5.1 Detailed Description

@title Gallery

A gallery to showcase all the generated flags.

Author

Ganghoon Park

Date

2022-04-11

7.6 GUI.py File Reference

@title GUI

Classes

- class GUI.SampleApp

  SampleApp creates the graphical user interface and allows the window to switch between different frames.
- class GUI.StartPage

  StartPage is a start page that opens once the program starts.

Namespaces

- namespace GUI

Variables

- GUI.app = SampleApp()

7.6.1 Detailed Description

@title GUI

A graphical user interface module uses all other modules to allow the user to communicate with the Random Flag Generator software

Author

Ganghoon Park

Date

2022-04-11

7.7 HashGenerator.py File Reference

@title HashGenerator

Namespaces

- namespace HashGenerator

Functions

- def HashGenerator._get_hash_algo (hash_type)

  gets the hashing algorithm from the dictionary of available hashing algorithms using the given input hash type string
- def HashGenerator._get_hash_hex (hash_input, hash_algo)

  gets the hexidecimal representation of the hashing digest using the given input string and hashing algorithm
- def HashGenerator.hash_generator (hash_input, hash_type='sha256')

  generates a hashing digest using the given input string and hashing algorithm

7.7.1 Detailed Description

@title HashGenerator

A library module for getting the hexidecimal hash of a given string

HashGenerator module, uses no other modules; no exported constants or types, no state or environment variables, no state invariant or assumptions

Author

Nathaniel Hu

Date

2022-03-31

7.8 HashToFlag.py File Reference

@title HashToFlag

Namespaces

- namespace HashToFlag

Functions

- def HashToFlag.grind_hash_for_colors (hashcode)

  Generates the array of colours to be used in the flag generation.
- def HashToFlag.grind_hash_for_base_stripe_style (hashcode)

  Generates the base stripe style to be used in flag generation.
- def HashToFlag.grind_hash_for_overlay_stripe_style (hashcode)

  Generates the overlay stripe style to be used in flag generation.
- def HashToFlag.grind_hash_for_stripe_number (hashcode)

  Generates the number of stripes to be used in flag generation.
- def HashToFlag.grind_hash_for_symbol_locations (hashcode)

  Generates the symbol location to be used in flag generation.
- def HashToFlag.grind_hash_for_symbol_number (hashcode)

  Generates the number of symbols to be used in flag generation.
- def HashToFlag.grind_hash_for_symbol_types (hashcode)

  Generates the symbol type to be used in flag generation.

7.8.1 Detailed Description

@title HashToFlag

A module with functions for taking a given hashcode input and generating the options for the flag to be generated.

Uses DecisionsUtilities and FlagAssetsLib

Author

Akram Hannoufa

Date

2022-03-15

7.9 Help.py File Reference

@title Help

Classes

- class Help.Help

  Shows the help menu to teach the user how to use the software and how it works.

Namespaces

- namespace Help

7.9.1 Detailed Description

@title Help

A help option teach the user how to use the software and giving more information of it.

Author

Ganghoon Park

Date

2022-04-11

7.10 JKAReader.py File Reference

@title JKAReader

Namespaces

- namespace JKAReader

Functions

- def JKAReader.parse_jka_file (filename)

  parses the input flag asset (.jka) file data into a pixel map

Variables

- string JKAReader.FILLED_PIXEL = "#"
- string JKAReader.UNFILLED_PIXEL = "."

7.10.1 Detailed Description

@title JKAReader

A library module for parsing .jka files for use in generating flags

JKAReader module, uses no other modules; exported constants FILLED_PIXEL and UNFILLED_PIXEL, no exported types, no state or environment variables, no state invariant, assumption that input .jka file exists in the flag assets directory

Author

Nathaniel Hu

Date

2022-04-07

7.11 Settings.py File Reference

@title Settings

Classes

- class Settings.Settings

  Creates settings graphical user interface app for user to use the random flag generator and set their settings.

Namespaces

- namespace Settings

7.11.1 Detailed Description

@title Settings

A settings option to select differnt flag size, set certain features, such as colour, symbols, stripes, and select different hash type

Author

Ganghoon Park, Nathaniel Hu

Date

2022-04-11

Index