SE 3XA3: Module Interface Specification

Random Flag Generator

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

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

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

Chapter 2

Class Index

2.1 Class List

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

Chapter 3

File Index

3.1 File List

Here is a list of all documented files with brief descriptions:

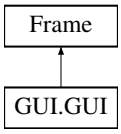
Chapter 4

Class Documentation

4.1 GUI.GUI Class Reference

Creates graphical user interface for random flag generator.

Inheritance diagram for GUI.GUI:



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 generate (self, event)

  Generate the flag using the input text.

Public Attributes

- **controller**
- **image_text_input**
- **image2**
- **image3**
- **image4**
- **resize_image_text_input**
- **resize_image2**
- **resize_image3**
- **resize_image4**
- **photo_text_input**
- **photo2**

- **photo3**
- **photo4**
- **label**
- **button1**
- **button2**
- **button3**
- **input_box**
- **generate**

4.1.1 Detailed Description

Creates graphical user interface for random flag generator.

Graphical user interface user to use the software.

4.1.2 Constructor & Destructor Documentation

4.1.2.1 __init__()

```
def GUI.GUI.__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.

4.1.3 Member Function Documentation

4.1.3.1 generate()

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

Generate the flag using the input text.

The flag is generated with this function

Parameters

self	Current object, common first parameter for any method of a class.
event	When generate button is clicked, it will call this function to generate the flag.

4.1.3.2 init_window()

```
def GUI.GUI.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, and a back button.

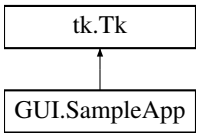
The documentation for this class was generated from the following file:

- GUI.py

4.2 GUI.SampleApp Class Reference

Creates graphical user interface app for user to use the random flag generator.

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**

4.2.1 Detailed Description

Creates graphical user interface app for user to use the random flag generator.

App for user to use the software.

4.2.2 Constructor & Destructor Documentation

4.2.2.1 __init__()

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

Constructor for new app GUI object.

Creates object for GUI

Parameters

self	Current object, common first parameter for any method of a class.
∗*args*	allows the function to accept an arbitrary number of arguments
∗∗*kwargs*	allows the function to accept an arbitrary number of keyword arguments.

4.2.3 Member Function Documentation

4.2.3.1 show_frame()

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

Shows the frame and switches between frames.

The frame is generated and switched.

Parameters

self	Current object, common first parameter for any method of a class.
page_name	Used to switch between frames in tkinter

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

- GUI.py

4.3 Settings.settings Class Reference

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

Public Member Functions

- def **set_flag_resolution** ()

4.3.1 Detailed Description

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

Settings graphical user interface for user to change the settings.

Sets flag resolution

The flag resolutions are determined, so the user needs to select the size they want

Returns

flag_resolution Selected flag resolution from the user, which the flag generator will use.

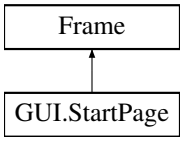
The documentation for this class was generated from the following file:

- Settings.py

4.4 GUI.StartPage Class Reference

Creates start page for random flag generator graphical user interface.

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_settings_button**
- **image_help_button**
- **resize_image_app_logo**
- **resize_image_start_button**
- **resize_image_settings_button**
- **resize_image_help_button**
- **photo_app_logo**
- **photo_start_button**
- **photo_settings_button**
- **photo_help_button**
- **app_logo**
- **start_button**
- **start_generate**
- **settings_button**
- **settings_generate**
- **help_button**
- **help_generate**

4.4.1 Detailed Description

Creates start page for random flag generator graphical user interface.

Start page for user to use the software.

4.4.2 Constructor & Destructor Documentation

4.4.2.1 __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, 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 descides what page will be visible.

4.4.3 Member Function Documentation

4.4.3.1 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, and a help button.

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

- GUI.py

Chapter 5

File Documentation

5.1 DecisionUtilities.py File Reference

@title 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, n)

  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 (a, b)

  Calculates the absolute difference of two float values.

Variables

- int **DecisionUtilities.COLOR_QUANTITY** = 5

  Number of colours to be generated for flag design options.
- int **DecisionUtilities.HEX_COLOR_LEN** = 6

  Length of a color in hex.
- int **DecisionUtilities.HEX_BASE** = 16

  Base of the hexadecimal number system.
- def **DecisionUtilities.MINIMUM_HASH_LEN** = COLOR_QUANTITY ∗ HEX_COLOR_LEN

  To generate unique colors, hashes need to contain at least this many characters.
- int **DecisionUtilities.ASPECT_CONTROL_LEN** = 6

  Length of hash characters to generate one option.
- int **DecisionUtilities.MAX_DECISION_VALUE** = 16777215

  Decimal representation of hexadecimal 'ffffff' as the maximum value for aspect decisions.
- bool **DecisionUtilities.DEBUG** = False

  Set True to generate debug output in this module.

5.1.1 Detailed Description

@title DecisionUtilities

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

Author

Akram Hannoufa

Date

2022-03-15

5.1.2 Function Documentation

5.1.2.1 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.

Returns

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

5.1.2.2 diff()

```
def DecisionUtilities.diff (
              a,
              b )
```

Calculates the absolute difference of two float values.

Parameters

a	first float value
b	second float value

Returns

The integer value of the absolute difference between float values.

5.1.2.3 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

Returns

RGB tuple, representing an RGB colour in integer values

5.1.2.4 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

Returns

decision, index of array to get decision from.

5.1.2.5 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.

Returns

modified hashcode, a padded version of the input hashcode.

5.1.2.6 split_sequence()

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

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
n	length of generated substrings

Returns

tokens, list of shorter substrings of length n

5.2 Display.py File Reference

@title Display

Functions

- def Display.display_flag (input_string)

  Displays the flag image file for the given input string and.

5.2.1 Detailed Description

@title Display

A display of the generated flag

Author

Ganghoon Park

Date

2022-03-17

5.2.2 Function Documentation

5.2.2.1 display_flag()

```
def Display.display_flag (
              input_string )
```

Displays the flag image file for the given input string and.

Generated flag gets displayed on the screen.

Parameters

input_string	The given input string for the generated flag. This is also the flag of the flag file.

5.3 FlagAssetsLib.py File Reference

@title FlagAssetsLib

Variables

- list **FlagAssetsLib.STRIPE_STYLE** = ['HORIZONTAL', 'VERTICAL', 'NONE']

  Possible stripe styles a flag can have.
- list FlagAssetsLib.STRIPE_NUMBER = ['ZERO', 'TWO', 'THREE', 'SIX', 'TWELVE']

  Possible number of stripes a flag can have.
- list **FlagAssetsLib.SYMBOL_LOCATION** = ['TOP_LEFT', 'CENTER', 'TOP_RIGHT']

  Possible locations a symbol can be placed on a flag.
- list **FlagAssetsLib.SYMBOL_NUMBER** = ['ZERO','ONE', 'TWO']

  Possible number of symbols a flag can have.
- list **FlagAssetsLib.SYMBOL_TYPES** = ['MOON', 'ROUNDEL', 'SWORD', 'CROSS', 'SALTIRE']
- dictionary **FlagAssetsLib.flag_assets** = {"SALTIRE": "jka/saltire.jka", "CROSS": "jka/cross.jka", "MOON": "jka/moon.jka", "SWORD": "jka/sword.jka", "ROUNDEL": "jka/roundel.jka"}

  Loads in all flag assets.

5.3.1 Detailed Description

@title FlagAssetsLib

A library of constants and symbol/design options

Author

Akram Hannoufa

Date

2022-03-15

5.3.2 Variable Documentation

5.3.2.1 STRIPE_NUMBER

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

Possible number of stripes a flag can have.

Possible symbol types.

5.4 FlagGenerator.py File Reference

@title FlagGenerator

Functions

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

5.4.1 Detailed Description

@title FlagGenerator

A library module for generating the flag 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

Date

2022-03-18

5.4.2 Function Documentation

5.4.2.1 generate_flag()

```
def FlagGenerator.generate_flag (
              hash_input,
              hash_type )
```

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

5.4.2.2 generate_flag_data()

```
def FlagGenerator.generate_flag_data (
              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

Returns

tuple containing the generated flag data consisting of the list of five colours, the stripe info and symbol info

5.5 Gallery.py File Reference

@title Gallery

Functions

- def Gallery.gallery ()

  Display a gallery of previously generate flags.

5.5.1 Detailed Description

@title Gallery

A gallery of the previously generated flags

Author

Ganghoon Park

Date

2022-03-17

5.5.2 Function Documentation

5.5.2.1 gallery()

```
def Gallery.gallery ( )
```

Display a gallery of previously generate flags.

The gallery will display the generated flags and have an option to go back to the main GUI screen.

5.6 GUI.py File Reference

@title GUI

Classes

- class GUI.SampleApp

  Creates graphical user interface app for user to use the random flag generator.
- class GUI.StartPage

  Creates start page for random flag generator graphical user interface.
- class GUI.GUI

  Creates graphical user interface for random flag generator.

Variables

- **GUI.app** = SampleApp()

5.6.1 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

2022-03-17

5.7 HashGenerator.py File Reference

@title 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

5.7.1 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

2022-03-18

5.7.2 Function Documentation

5.7.2.1 _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

5.7.2.2 _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

5.7.2.3 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.py File Reference

@title 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_stripe_style (hashcode)

  Generates the 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.

5.8.1 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

5.8.2 Function Documentation

5.8.2.1 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.

5.8.2.2 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.

5.8.2.3 grind_hash_for_stripe_style()

```
def HashToFlag.grind_hash_for_stripe_style (
              hashcode )
```

Generates the 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 stripe style option.

5.8.2.4 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.

5.8.2.5 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.

5.8.2.6 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.py File Reference

@title Help

Functions

- def Help.help_menu ()

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

5.9.1 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

2022-03-17

5.9.2 Function Documentation

5.9.2.1 help_menu()

```
def Help.help_menu ( )
```

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.

5.10 JKAReader.py File Reference

@title 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** = "#"

  exported type representing a filled pixel for a flag asset
- string **JKAReader.UNFILLED_PIXEL** = "."

  exported type representing an unfilled pixel for a flag asset

5.10.1 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

2022-03-18

5.10.2 Function Documentation

5.10.2.1 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

5.11 Settings.py File Reference

@title Settings

Classes

- class Settings.settings

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

Functions

- def Settings.set_hash_type ()

  Sets the hash type.
- def Settings.set_flag_colour ()

  Sets flag colour.
- def Settings.set_flag_symbol ()

  Sets flag symbol.

5.11.1 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

Date

2022-03-17

5.11.2 Function Documentation

5.11.2.1 set_flag_colour()

```
def Settings.set_flag_colour ( )
```

Sets flag colour.

The flag colour are determined, so the user needs to select the colour they want

Returns

flag_colour Selected flag colour from the user, which the flag generator will use.

5.11.2.2 set_flag_symbol()

```
def Settings.set_flag_symbol ( )
```

Sets flag symbol.

The flag symbols are determined by the jka files, so the user needs to select the symbol they want

Returns

flag_symbol Selected flag symfol from the user, which the flag generator will use.

5.11.2.3 set_hash_type()

```
def Settings.set_hash_type ( )
```

Sets the hash type.

The flag generator will use the selected hash type

Returns

hash_type Selected hash type from the user, which flag generator will use.

Index