

API Documentation

API Documentation

April 11, 2011

Contents

Contents	1
1 Module camera	3
1.1 Variables	3
1.2 Class Camera	3
1.2.1 Methods	3
2 Module check_button	4
2.1 Variables	4
2.2 Class Check_button	4
2.2.1 Methods	4
3 Module check_stone	5
3.1 Functions	5
3.2 Variables	5
4 Module create_sgf	6
4.1 Variables	6
4.2 Class Create_sgf	6
4.2.1 Methods	6
5 Module cte	7
5.1 Variables	7
6 Module difference	9
6.1 Functions	9
6.2 Variables	9
7 Module function	10
7.1 Functions	10
7.2 Variables	10
8 Module interface_about	11
8.1 Variables	11
8.2 Class Interface_about	11
8.2.1 Methods	11
9 Module interface_configuration	12

9.1	Variables	12
9.2	Class Interface_configuration	12
9.2.1	Methods	12
10	Module interface_main	14
10.1	Variables	14
10.2	Class Interface_main	14
10.2.1	Methods	14
11	Module intersection	17
11.1	Functions	17
11.2	Variables	17
11.3	Class Intersection	17
11.3.1	Methods	18
12	Module luminosity	19
12.1	Functions	19
12.2	Variables	19
13	Module main	20
13.1	Variables	20
14	Module notation	21
14.1	Functions	21
14.2	Variables	21
15	Module tablero	22
15.1	Variables	22
15.2	Class Tablero	22
15.2.1	Methods	22
16	Module video	23
16.1	Variables	23
16.2	Class Video	23
16.2.1	Methods	23
	Index	24

1 Module camera

1.1 Variables

Name	Description
<code>--package--</code>	Value: None

1.2 Class Camera

Clase para trabajar con las cámaras que estén conectadas al ordenador

1.2.1 Methods

<code>--init--(self)</code>
<code>check_name_cameras(self)</code> Función que devuelve los nombres de las camaras conectadas
<code>open_camera(self, num)</code> Devuelve la captura de la cámara
<code>get_frame(self)</code> Devuelve un frame de la captura
<code>close_camera(self)</code> Cierra la cámara abierta

2 Module *check_button*

2.1 Variables

Name	Description
<code>--package--</code>	Value: None

2.2 Class *Check_button*

Clase que comprueba la salida de arduino. Devuelve True cuando encuentra Pulsado

2.2.1 Methods

<code>--init--(<i>self</i>)</code>
<code>check(<i>self</i>)</code>

3 Module *check_stone*

3.1 Functions

check_stone(*tablero*, *p*, *color*)

Funcion recursiva que comprueba si la piedra pasada por parámetro está muerta o esta viva. Ver reglas del go para más detalles.

3.2 Variables

Name	Description
BLACK	Value: 0
EJE_X	Value: 0
EJE_Y	Value: 1
INITSGF	Value: ('(;FF[4]GM[1]SZ[19]', '\nAP[Tablerogo]', '\nPB[player_b]...'
KEY_NOT_DOWN	Value: -1
NAME_FILE	Value: 'Inserte el nombre del archivo y pulse Intro: '
PLAYER_B	Value: 'player_b'
PLAYER_W	Value: 'player_w'
RANK_B	Value: '20k'
RANK_W	Value: '20k'
SIZE	Value: 19
SPACE	Value: -1
TEXT_INTERSECTION	Value: 'Intersecciones encontradas. \xc2\xbfEstan bien?(Y/N)'
THRESHOLD	Value: 0.7
WHITE	Value: 1
WITH_VIDEO	Value: 2
WITH_WEBCAM	Value: 1
__package__	Value: None
fecha_actual	Value: '11 Apr 2011'
tiempo	Value: 1800

4 Module *create_sgf*

4.1 Variables

Name	Description
<code>--package--</code>	Value: None

4.2 Class *Create_sgf*

Clase que nos genera un archivo de extensión *.sgf* con una cabecera predeterminada para guardar la partida que estamos capturando.

4.2.1 Methods

```
--init--(self, path='sgf')
```

```
add_text(self, text, color)
```

```
end_file(self)
```

5 Module cte

Este módulo es para guardar las constantes que se usan en el programa.

5.1 Variables

Name	Description
PLAYER_B	Value: 'player_b'
RANK_B	Value: '20k'
PLAYER_W	Value: 'player_w'
RANK_W	Value: '20k'
SIZE	Value: 19
WITH_WEBCAM	Value: 1
WITH_VIDEO	Value: 2
BLACK	Value: 0
WHITE	Value: 1
SPACE	Value: -1
TEXT_INTERSECTION	Value: 'Intersecciones encontradas. \xc2\xbfEstan bien?(Y/N)'
NAME_FILE	Value: 'Inserte el nombre del archivo y pulse Intro: '
EJE_X	Value: 0
EJE_Y	Value: 1
KEY_NOT_DOWN	Value: -1
THRESHOLD	Value: 0.7
fecha.actual	Value: '11 Apr 2011'
tiempo	Value: 1800

continued on next page

Name	Description
INITSGF	<p>(section) CARACTERISTICAS DE LOS ARCHIVOS .SGF</p> <p>* AB: Add Black: locations of Black stones to be placed on the board prior to the first move. * AW: Add White: locations of White stones to be placed on the board prior to the first move. * AN: Annotations: name of the person commenting the game. * AP: Application: application that was used to create the SGF file (e.g. CGOban2,...). * B: a move by Black at the location specified by the property value. * BR: Black Rank: rank of the Black player. * BT: Black Team: name of the Black team. * C: Comment: a comment. * CP: Copyright: copyright information. See Kifu Copyright Discussion. * DT: Date: date of the game. * EV: Event: name of the event (e.g. 58th Honinbo Title Match). * FF: File format: version of SGF specification governing this SGF file. * GM: Game: type of game represented by this SGF file. A property value of 1 refers to Go. * GN: Game Name: name of the game record. * HA: Handicap: the number of handicap stones given to Black. Placement of the handicap stones are set using the AB property. * KM: Komi: komi. * ON: Opening: information about the opening (fuseki), rarely used in any file. * OT: Overtime: overtime system. * PB: Black Name: name of the black player. * PC: Place: place where the game was played (e.g.: Tokyo). * PL: Player: color of player to start. * PW: White Name: name of the white player. * RE: Result: result, usually in the format "B+R" (Black wins by resign) or "B+3.5" (black wins by 3.5 moku). * RO: Round: round (e.g.: 5th game). * RU: Rules: ruleset (e.g.: Japanese). * SO: Source: source of the SGF file. * SZ: Size: size of the board, non square boards are supported. * TM: Time limit: time limit in seconds. * US: User: name of the person who created the SGF file. * W: a move by White at the location specified by the property value. * WR: White Rank: rank of the White player. * WT: White Team: name of the White team.</p> <p>Value: ('(;FF[4]GM[1]SZ[19]', '\nAP[Tablerogo]', '\nPB[player_b]...'</p>
__package__	Value: None

6 Module difference

Devuelve la diferencia entre dos matrices/imagenes dadas, aplicándoles anteriormente el efecto Smooth

6.1 Functions

difference(*img1*, *img2*)

Devuelve la diferencia entre dos matrices/imagenes dadas, aplicándoles anteriormente el efecto Smooth

6.2 Variables

Name	Description
__package__	Value: None

7 Module function

7.1 Functions

convert_opencv2gtk(<i>image</i>)

Función que convierte una imagen de opencv en una imagen gtk
--

convert_opencv_gray2gtk(<i>image</i>)
--

Función que convierte una imagen de opencv en una imagen gtk
--

load_image_no_webcam()

7.2 Variables

Name	Description
__package__	Value: None

8 Module *interface_about*

8.1 Variables

Name	Description
<code>--package--</code>	Value: None

8.2 Class *Interface_about*

8.2.1 Methods

<code>__init__(self)</code>
<code>dialog_hide(self, widget)</code>

9 Module *interface_configuration*

9.1 Variables

Name	Description
<code>--package--</code>	Value: None

9.2 Class *Interface_configuration*

Clase utilizada para las interfaz de configuración.

9.2.1 Methods

<code>--init--(self, camera)</code> Inicializador. Parameters camera: instancia de <i>Camera</i> para trabajar con las cámaras. <i>(type=Camera)</i>
<code>button_aceptar_clicked(self, widget)</code> Callback que es llamado cuando hacemos click en el boton Aceptar, y que guarda todas las variables escritas en la configuracion
<code>button_close_window_clicked(self, widget)</code>
<code>hscale_camera_value_changed(self, widget)</code> Callback que se activa cuando movemos la barra de escala, la cual nos cambia al numero de cámara que seleccionemos.
<code>menu_language_1_activate(self, widget)</code> Callback que es llamado cuando se pulsa el botón del idioma 1 en la barra de menú de la interfaz y que nos cambia al idioma seleccionado.
<code>menu_language_2_activate(self, widget)</code> Callback que es llamado cuando se pulsa el botón del idioma 2 en la barra de menú de la interfaz y que nos cambia al idioma seleccionado.
<code>menu_help_about_activate(self, widget)</code> Callback que es llamado cuando se pulsa el botón del idioma 1 en la barra de menú de la interfaz y que nos cambia al idioma seleccionado.

menu_file_close_activate(*self*, *widget*)

Callback que es llamado cuando pulsamos la opción de cerrar ventana de la barra menú, la cual cierra la ventana

gtk_destroy_window(*self*, *widget*)

Callback que es llamado cuando pulsamos la X, pulsamos ALT+F4, o cuando llamamos a `window.destroy()`, esto no cierra la ventana.

update(*self*)

Función llamada por un bucle cada cierto tiempo que nos actualiza la cámara y el video mediante frame y los muestra por pantalla.

Return Value

Devuelve True siempre que exista frame de una cámara o un video, y False en caso de no encontrarlo.

(*type=bool*)

load_image(*self*, *image*)

Nos carga una imagen en la pantalla.

Parameters

image: imagen a cargar.

(*type=IplImage*)

10 Module *interface_main*

10.1 Variables

Name	Description
<code>--package--</code>	Value: None

10.2 Class *Interface_main*

Clase utilizada para las interfaz principal.

10.2.1 Methods

`--init--(self)`

`menu_file_new_activate(self, widget)`

Callback que es llamado cuando se pulsa el botón de Nuevo en la barra de menú de la interfaz y que nos abre una nueva interfaz de configuración.

`menu_help_about_activate(self, widget)`

Callback que es llamado cuando se pulsa el botón de Acerca de en la barra de menú de la interfaz y que nos abre una nueva ventana.

`menu_language_1_activate(self, widget)`

Callback que es llamado cuando se pulsa el botón del idioma 1 en la barra de menú de la interfaz y que nos cambia al idioma seleccionado.

`menu_language_2_activate(self, widget)`

Callback que es llamado cuando se pulsa el botón del idioma 2 en la barra de menú de la interfaz y que nos cambia al idioma seleccionado.

`menu_file_exit_activate(self, widget)`

Callback que es llamado cuando pulsamos la opción de salir de la barra menú.

`button_end_clicked(self, widget)`

Callback que es llamado cuando pulsamos el botón terminar

`button_search_intersection_clicked(self, widget)`

Callback que es llamado para buscar las intersecciones del tablero cuando pulsamos el boton Buscar Intersecciones.

button_intersection_ok_clicked(*self*, *widget*)

Callback que es llamado cuando pulsamos el boton Intersecciones OK, lo que hace es activar el foton de foto y el de terminar para que empecemos a buscar piedras.

button_foto_clicked(*self*, *widget*)

Callback que es llamado cuando pulsamos el boton Foto, el cual nos busca los cambios producidos en el tablero.

gtk_main_quit(*self*, *widget*)

Funcion que es llamada cuando queremos cerramos la ventana.

load_image_left(*self*, *image*)

Nos carga una imagen en la parte izquierda de la pantalla.

Parameters

image: imagen a cargar.

(*type=IplImage*)

load_image_right(*self*, *image*)

Nos carga una imagen en la parte derecha de la pantalla.

Parameters

image: imagen a cargar.

(*type=IplImage*)

check_configuration(*self*)

Comprueba si la ventana de configuración está abierta.

Return Value

True si la configuración se ha creado correctamente, False en caso contrario.

(*type=bool*)

end_file_sgf(*self*)

Función que cierra el archivo sgf y desactiva los botones hasta nueva configuración.

configuration_initial(*self*)

Funcion que vuelve a la configuración inicial.

configuration_applicate(*self*)

Función que aplica la configuración creada en la interfaz configuracion.

update(*self*)

Función que actualiza el video, cámara o imagen.

Return Value

Devuelve True siempre.

(*type=bool*)

new_thread(*self*, *method*)

Funcion que crea un nuevo hilo.

thread_progressbar(*self*)

Función hilo que muestra la barra de progreso.

thread_search_intersection(*self*)

Función hilo que busca las intersecciones del tablero.

11 Module intersection

11.1 Functions

<code>contador(lista, num, indice=0)</code>

11.2 Variables

Name	Description
BLACK	Value: 0
EJE_X	Value: 0
EJE_Y	Value: 1
INITSGF	Value: ('(;FF[4]GM[1]SZ[19]', '\nAP[Tablerogo]', '\nPB[player_b]...'
KEY_NOT_DOWN	Value: -1
NAME_FILE	Value: 'Inserte el nombre del archivo y pulse Intro: '
PLAYER_B	Value: 'player_b'
PLAYER_W	Value: 'player_w'
RANK_B	Value: '20k'
RANK_W	Value: '20k'
SIZE	Value: 19
SPACE	Value: -1
TEXT_INTERSECTION	Value: 'Intersecciones encontradas. \xc2\xbfEstan bien?(Y/N)'
THRESHOLD	Value: 0.7
WHITE	Value: 1
WITH_VIDEO	Value: 2
WITH_WEBCAM	Value: 1
--package--	Value: None
fecha_actual	Value: '11 Apr 2011'
tiempo	Value: 1800

11.3 Class Intersection

Contiene todos los datos de las intersecciones de un tablero y las funciones necesarias para encontrarlas. Parámetros necesarios: **image*: la imagen del tablero que queremos tratar. **template*: el template que vamos a pasarle a la imagen. **size*: el tamaño del tablero para saber cuantas intersecciones buscar.

11.3.1 Methods

__init__(self, image, template='images/tplPoint.png', size=19)

Inicializamos la imagen, el template y algunas variables.

Parameters

image: imagen donde se buscarán las intersecciones del tablero o cadena con el directorio donde se encuentra la imagen.

(type=IplImage or str)

template: imagen de la plantilla que buscaremos en la imagen.

(type=str)

size: el tamaño del tablero

(type=int)

Return Value

None

(type=None)

detect_intersections(self, threshold=0.7)

Función recursiva que detecta las intersecciones del tablero usando un template y un valor umbral, el cual iremos variando para hasta obtener el resultado deseado.

Parameters

threshold: Umbral para usar la búsqueda del template.

(type=float.)

paint_save_intersection(self)

Función que pinta y guarda las intersecciones para chequeos.

media(self)

Hallamos la media de todas las filas y columnas.

search_intersection(self, points)

Busca el punto donde se encuentra la piedra para añadirlo al archivo sgf.

Parameters

points: intersecciones donde puede encontrarse la piedra.

(type=list(tuples*))

Return Value

Tupla que contiene: Cadena con la intersección donde se encuentra la piedra.

Color de la piedra. Boleano para saber si hemos terminado la partida.

(type=tuple(str, int, bool # Los candidatos los hemos dividido en dos, los que estaban en # intersecciones libres, y los que estaban en ocupadas.)

12 Module luminosity

12.1 Functions

luminosity(*image*, *tablero*, *size*, *lum*=150)

Funcion que busca los puntos luminosos de la imagen y los devuelve. Esta función ya no se usa por que existian varios factores en contra. Factor tiempo: tardaba mucho en recorrer muchos puntos (aunque con alg. de busqueda se podía bajar ese tiempo). Factor error: como cogía todos los puntos luminosos, cuando aparecía una pequeña sombra se producían muchos errores y a la vez cientos de puntos mas que recorrer

search_circle(*image*, *tablero*, *size*, *dp*=1.6)

Función que trata la imagen luminosa con los filtros Canny y Smooth y busca circunferencias. Devuelve los centros de las circunferencias encontradas

12.2 Variables

Name	Description
<code>__package__</code>	Value: None

13 Module main

13.1 Variables

Name	Description
i_main	Value: Interface_main()
--package--	Value: None

14 Module notation

14.1 Functions

notation(*point*)

Función que tiene como parámetro de entrada un punto, y devuelve el punto en la notación del tablero de go

14.2 Variables

Name	Description
--package--	Value: None

15 Module tablero

15.1 Variables

Name	Description
<code>--package--</code>	Value: None

15.2 Class Tablero

Clase que nos crea una matriz tablero para poder guardar todos los datos de las intersecciones y las piedras puestas

15.2.1 Methods

<code>--init--(self, size=19)</code>

<code>get_x_y(self)</code>

Funcion que nos devuelve en que coordenadas se encuentra la intersección del tablero 1-1, y la 19-19
--

<code>get_square(self)</code>

Función que cambia el valor de la variable self.square, la cual contiene la información del tamaño de un cuadrado del tablero

<code>--str--(self)</code>

16 Module video

16.1 Variables

Name	Description
<code>--package--</code>	Value: None

16.2 Class Video

Clase para abrir videos

16.2.1 Methods

<code>--init--(<i>self</i>)</code>
<code>open_video(<i>self</i>, <i>path</i>)</code> Devuelve la captura del video
<code>get_frame(<i>self</i>)</code> Devuelve un frame de la captura
<code>close_video(<i>self</i>)</code> Cierra la cámara abierta

Index

- camera (*module*), 3
 - camera.Camera (*class*), 3
 - camera.Camera.__init__ (*method*), 3
 - camera.Camera.check_name_cameras (*method*), 3
 - camera.Camera.close_camera (*method*), 3
 - camera.Camera.get_frame (*method*), 3
 - camera.Camera.open_camera (*method*), 3
- check_button (*module*), 4
 - check_button.Check_button (*class*), 4
 - check_button.Check_button.__init__ (*method*), 4
 - check_button.Check_button.check (*method*), 4
- check_stone (*module*), 5
 - check_stone.check_stone (*function*), 5
- create_sgf (*module*), 6
 - create_sgf.Create_sgf (*class*), 6
 - create_sgf.Create_sgf.__init__ (*method*), 6
 - create_sgf.Create_sgf.add_text (*method*), 6
 - create_sgf.Create_sgf.end_file (*method*), 6
- cte (*module*), 7–8
- difference (*module*), 9
 - difference.difference (*function*), 9
- function (*module*), 10
 - function.convert_opencv2gtk (*function*), 10
 - function.convert_opencv_gray2gtk (*function*), 10
 - function.load_image_no_webcam (*function*), 10
- interface_about (*module*), 11
 - interface_about.Interface_about (*class*), 11
 - interface_about.Interface_about.__init__ (*method*), 11
 - interface_about.Interface_about.dialog_hide (*method*), 11
- interface_configuration (*module*), 12–13
 - interface_configuration.Interface_configuration (*class*), 12–13
 - interface_configuration.Interface_configuration.__init__ (*method*), 12
 - interface_configuration.Interface_configuration.button_acceptar_clicked (*method*), 12
 - interface_configuration.Interface_configuration.button_close_window_clicked (*method*), 12
 - interface_configuration.Interface_configuration.gtk_destroy_window (*method*), 13
 - interface_configuration.Interface_configuration.hscale_camera_value_changed (*method*), 12
 - interface_configuration.Interface_configuration.load_image (*method*), 13
- interface_configuration.Interface_configuration.menu_file_close_all (*method*), 12
- interface_configuration.Interface_configuration.menu_help_about (*method*), 12
- interface_configuration.Interface_configuration.menu_language_1 (*method*), 12
- interface_configuration.Interface_configuration.menu_language_2 (*method*), 12
- interface_configuration.Interface_configuration.update (*method*), 13
- interface_main (*module*), 14–16
 - interface_main.Interface_main (*class*), 14–16
 - interface_main.Interface_main.__init__ (*method*), 14
 - interface_main.Interface_main.button_end_clicked (*method*), 14
 - interface_main.Interface_main.button_foto_clicked (*method*), 15
 - interface_main.Interface_main.button_intersection_ok_clicked (*method*), 14
 - interface_main.Interface_main.button_search_intersection_clicked (*method*), 14
 - interface_main.Interface_main.check_configuration (*method*), 15
 - interface_main.Interface_main.configuration_apply (*method*), 15
 - interface_main.Interface_main.configuration_initial (*method*), 15
 - interface_main.Interface_main.end_file_sgf (*method*), 15
 - interface_main.Interface_main.gtk_main_quit (*method*), 15
 - interface_main.Interface_main.load_image_left (*method*), 15
 - interface_main.Interface_main.load_image_right (*method*), 15
 - interface_main.Interface_main.menu_file_exit_activate (*method*), 14
 - interface_main.Interface_main.menu_file_new_activate (*method*), 14
 - interface_main.Interface_main.menu_help_about_activate (*method*), 14
 - interface_main.Interface_main.menu_language_1_activate (*method*), 14
 - interface_main.Interface_main.menu_language_2_activate (*method*), 14
 - interface_main.Interface_main.new_thread (*method*), 16
 - interface_main.Interface_main.thread_progressbar (*method*), 16

- interface_main.Interface_main.thread_search_intersection
(*method*), 16
 - interface_main.Interface_main.update (*method*),
15
- intersection (*module*), 17–18
 - intersection.contador (*function*), 17
 - intersection.Intersection (*class*), 17–18
 - intersection.Intersection.__init__ (*method*), 18
 - intersection.Intersection.detect_intersections (*method*),
18
 - intersection.Intersection.media (*method*), 18
 - intersection.Intersection.paint_save_intersection
(*method*), 18
 - intersection.Intersection.search_intersection (*method*),
18
- luminosity (*module*), 19
 - luminosity.luminosity (*function*), 19
 - luminosity.search_circle (*function*), 19
- main (*module*), 20
- notation (*module*), 21
 - notation.notation (*function*), 21
- tablero (*module*), 22
 - tablero.Tablero (*class*), 22
 - tablero.Tablero.__init__ (*method*), 22
 - tablero.Tablero.__str__ (*method*), 22
 - tablero.Tablero.get_square (*method*), 22
 - tablero.Tablero.get_x_y (*method*), 22
- video (*module*), 23
 - video.Video (*class*), 23
 - video.Video.__init__ (*method*), 23
 - video.Video.close_video (*method*), 23
 - video.Video.get_frame (*method*), 23
 - video.Video.open_video (*method*), 23