

Python wrapper for Gcompris

Ir. Olivier Samyn

<osamyn@ulb.ac.be>

Table of Contents

[Notes about this wrapper](#)

[Python board structure](#)

[GcomprisBoard structure mapping](#)

[gcompris.h functions and constants Mapping](#)

[Functions mapping](#)

[GComprisBarFlags enum mapping](#)

[Misc constants mapping](#)

[Colors constants mapping](#)

[Fonts constants mapping](#)

[Board font constants mapping](#)

[Cursor constants mapping](#)

[gameutil.h functions and constants Mapping](#)

[Functions mapping](#)

[Constants mapping](#)

[bonus.h functions and constants Mapping](#)

[Functions mapping](#)

[Constants mapping \(BonusStatusList\)](#)

[Constants mapping \(BonusList\)](#)

[Constants mapping \(BoardFinishedList\)](#)

[score.h functions and constants Mapping](#)

[Functions mapping](#)

[Constants mapping](#)

[sound.h functions and constants Mapping](#)

[Functions mapping](#)

[Constants mapping](#)

[timer.h functions and constants Mapping](#)

[Functions mapping](#)

[Constants mapping](#)

[anim.h structures Mapping](#)

[Animation](#)

[CanvasItem](#)

Notes about this wrapper

Generally speaking, all C gc_XXX functions are wrapped to a corresponding gcompris.XXX function; parameters and return types remains the same. Some structures have also been wrapped. See below particular notes about this.

The wrapped functions are divided into packages. Every package corresponds to a gcompris header file (gameutil.h, score.h, ...)

For each package/header not all functions have been wrapped. Only functions that are used/usefull for plugins are wrapped.

Notice also that not all functions have been tested (with time and new python plugins implemented they will be, but...). The wrappers have been written, they compile, but there is no warranty on their behaviour... I you find a bug report it!

Python board structure

Each board written in python must contains a class which name is "Gcompris_XXX" where XXX is the name of the board. Let's take an example: I made a gcompris board written in python and called "pythontest".

For this board there is a gcompris pythontest.xml file containing "pythontest" for the board name and "pythonboard" for the board type. There's also a pythontest.py file containing the board code. And to finish, this file contains a "Gcompris_pythontest" class definition which in turns contains the needed gcompris board functions.

The class defined in the python board must contains the following methods:

- `__init__(self, gcomprisBoard)`
- `start(self)`
- `end(self)`
- `ok(self)`
- `repeat(self)`
- `config(self)`
- `key_press(self, keyval)`
- `pause(self, pause)`
- `set_level(self, level)`

For the method role and use, please consult the sample pythontest.py board and other C boards.

GcomprisBoard structure mapping

An instance of this structure in Python will act like an instance of the C one. Here are some Python call samples:

```
...
gcomprisBoard.level = 1
gcomprisBoard.maxlevel = 1
...
gcompris.bar_start(gcomprisBoard.canvas)
...
```

Here are the Python structure member name and type, and a note indicating if the attribute is readable and/or writable.

Structure member	Type	Readable	Writable
type	string	Y	N
board_ready	boolean	Y	N
mode	string	Y	N
name	string	Y	N
title	string	Y	N
description	string	Y	N
icon_name	string	Y	N
author	string	Y	N
boarddir	string	Y	N
filename	string	Y	N
difficulty	string	Y	N
mandatory_sound_file	string	Y	N
section	string	Y	N
menuposotion	string	Y	N
prerequisite	string	Y	N
goal	string	Y	N
manual	string	Y	N
credit	string	Y	N
width	int	Y	N
height	int	Y	N

Structure member	Type	Readable	Writable
canvas	gnome.canvas.Canvas	Y	N
previous_board	GcomprisBoard	Y	N
level	int	Y	Y
maxlevel	int	Y	Y
sublevel	int	Y	Y
number_of_sublevel	int	Y	Y

gcompris.h functions and constants Mapping

Functions mapping

Python function	C equivalent	Notes
gcompris.end_board()	void gc_board_end(void)	
gcompris.bar_start(gnomeCanvas)	void gc_bar_start(GnomeCanvas *theCanvas)	
gcompris.set_background(gnomeCanvasGroup, file)	GnomeCanvasItem *gc_set_background(GnomeCanvasGroup *parent, gchar *file)	return a gnome.canvas.CanvasItem object
gcompris.bar_set_level(gcomprisBoard)	void gc_bar_set_level(GcomprisBoard *gcomprisBoard)	
gcompris.bar_set_repeat_icon(pixmap)	void gc_bar_set_repeat_icon(GdkPixmap *pixmap)	
gcompris.bar_set(flags)	void gc_bar_set(const GComprisBarFlags flags)	
gcompris.bar_hide(hide)	void gc_bar_hide(gboolean hide)	
gcompris.get_canvas()	GnomeCanvas* gc_get_canvas()	return a gnome.canvas.Canvas
gcompris.get_window()	GtkWidget* gc_get_window(void)	return a gtk.Widget
gcompris.get_locale()	gchar* gc_locale_get(void)	return a string
gcompris.set_locale(locale)	void gc_locale_set(gchar *locale)	
gcompris.set_cursor(cursor_type)	void gc_cursor_set(guint gdk_cursor_type)	
gcompris.images_selector_start(gcomprisBoard, dataset, callback)	void gc_selector_images_start(GcomprisBoard *gcomprisBoard, gchar *dataset, void (*callback)(GcomprisBoard *gcomprisBoard, gchar *dataset))	The callback must be a callable object

Python function	C equivalent	Notes
callback)	*dataset, ImageSelectorCallBac k imscb)	
gcompris.images_selector_stop()	void gc_selector_images_stop(void)	
gcompris.exit()	void gc_exit(void)	
gcompris.log_set_reason (gcomprisBoard, expected, got)	gc_log_set_reason (GcomprisBoard *gcomprisBoard, gchar *expected, gchar *got)	
gcompris.log_end (gcomprisBoard, status)	gc_log_end (GcomprisBoard *gcomprisBoard, gchar *status)	
gcompris.file_selector_load(gcomprisBoard, rootdir, file_types, callback)	void gc_selector_file_load (GcomprisBoard *gcomprisBoard, gchar *rootdir, gchar *file_types, FileSelectorCallBac k fscb)	The callback must be a callable object
gcompris.file_selector_save(gcomprisBoard, rootdir, file_types, callback)	void gc_selector_file_save(GcomprisBoard *gcomprisBoard, gchar *rootdir, gchar *file_types, FileSelectorCallBac k fscb)	The callback must be a callable object
gcompris.file_selector_stop()	void gc_selector_file_stop(void)	
gcompris.get_database()	gchar* gc_db_get_filename(void)	return the filename (string) of the database (fullpath)

GComprisBarFlags enum mapping

Python constant	C equivalent
gcompris.BAR_LEVEL	GCOMPRIS_BAR_LEVEL
gcompris.BAR_OK	GCOMPRIS_BAR_OK
gcompris.BAR_REPEAT	GCOMPRIS_BAR_REPEAT
gcompris.BAR_REPEAT_ICON	GCOMPRIS_BAR_REPEAT_ICON
gcompris.BAR_CONFIG	GCOMPRIS_BAR_CONFIG
gcompris.BAR_ABOUT	GCOMPRIS_BAR_ABOUT

Misc constants mapping

Python constant	C equivalent
gcompris.BOARD_HEIGHT	BOARDHEIGHT
gcompris.BOARD_WIDTH	BOARDWIDTH
gcompris.BAR_HEIGHT	BAR_HEIGHT
gcompris.DEFAULT_SKIN	DEFAULT_SKIN

Colors constants mapping

Python constant	C equivalent
gcompris.COLOR_TITLE	COLOR_TITLE
gcompris.COLOR_TEXT_BUTTON	COLOR_TEXT_BUTTON
gcompris.COLOR_CONTENT	COLOR_CONTENT
gcompris.COLOR_SUBTITLE	COLOR_SUBTITLE

Fonts constants mapping

Python constant	C equivalent
gcompris.FONT_TITLE	FONT_TITLE
gcompris.FONT_TITLE_FALLBACK	FONT_TITLE_FALLBACK
gcompris.FONT_SUBTITLE	FONT_SUBTITLE
gcompris.FONT_SUBTITLE_FALLBACK	FONT_SUBTITLE_FALLBACK
gcompris.FONT_CONTENT	FONT_CONTENT
gcompris.FONT_CONTENT_FALLBACK	FONT_CONTENT_FALLBACK

Board font constants mapping

Python constant	C equivalent
gcompris.FONT_BOARD_TINY	FONT_BOARD_TINY
gcompris.FONT_BOARD_SMALL	FONT_BOARD_SMALL
gcompris.FONT_BOARD_MEDIUM	FONT_BOARD_MEDIUM
gcompris.FONT_BOARD_BIG	FONT_BOARD_BIG
gcompris.FONT_BOARD_BIG_BOLD	FONT_BOARD_BIG_BOLD

Python constant	C equivalent
gcompris.FONT_BOARD_FIXED	FONT_BOARD_FIXED
gcompris.FONT_BOARD_TITLE	FONT_BOARD_TITLE
gcompris.FONT_BOARD_TITLE_BOLD	FONT_BOARD_TITLE_BOLD
gcompris.FONT_BOARD_HUGE	FONT_BOARD_HUGE
gcompris.FONT_BOARD_HUGE_BOLD	FONT_BOARD_HUGE_BOLD

Cursor constants mapping

Python constant	C equivalent
gcompris.CURSOR_FIRST_CUSTOM	GCOMPRIS_FIRST_CUSTOM_CURSOR
gcompris.CURSOR_LINE	GCOMPRIS_LINE_CURSOR
gcompris.CURSOR_FILLRECT	GCOMPRIS_FILLRECT_CURSOR
gcompris.CURSOR_RECT	GCOMPRIS_RECT_CURSOR
gcompris.CURSOR_FILLCIRCLE	GCOMPRIS_FILLCIRCLE_CURSOR
gcompris.CURSOR_CIRCLE	GCOMPRIS_CIRCLE_CURSOR
gcompris.CURSOR_DEL	GCOMPRIS_DEL_CURSOR
gcompris.CURSOR_FILL	GCOMPRIS_FILL_CURSOR
gcompris.CURSOR_SELECT	GCOMPRIS_SELECT_CURSOR
gcompris.CURSOR_DEFAULT	GCOMPRIS_DEFAULT_CURSOR

gameutil.h functions and constants Mapping

Functions mapping

Python function	C equivalent	Notes
gcompris.utils.image_to_skin(image_name)	gchar *gc_skin_image_get(gchar *image_name)	return a string
gcompris.skin.load_pixmap(pixmapfile)	GdkPixbuf *gc_skin_pixmap_load(char *pixmapfile)	return a gtk.gdk.Pixbuf
gcompris.utils.load_pixmap(pixmapfile)	GdkPixbuf *gc_pixmap_load(char *pixmapfile)	return a gtk.gdk.Pixbuf
gcompris.utils.find_file_absolute(file)	char *gc_file_find_absolute(char *file)	return a string

Python function	C equivalent	Notes
)		
gcompris.utils.set_image_focus(item, focus)	void gc_item_focus_set(GnomeCanvasItem *item, gboolean focus)	
gcompris.utils.item_event_focus(item, event, dest_item)	gint gc_item_focus_event(GnomeCanvasItem *item, GdkEvent *event, GnomeCanvasItem *dest_item)	return an integer
gcompris.utils.item_absolute_move(item, x, y)	void gc_item_absolute_move(GnomeCanvasItem *item, int x, int y)	
gcompris.utils.item_rotate(item, angle)	void gc_item_rotate(GnomeCanvasItem *item, double angle)	
gcompris.utils.item_rotate_relative(item, angle)	void gc_item_rotate_relative(GnomeCanvasItem *item, double angle)	
gcompris.utils.item_rotate_with_center(item, angle, x, y)	void gc_item_rotate_with_center(GnomeCanvasItem *item, double angle, int x, int y)	
gcompris.utils.item_rotate_relative_with_center(item, angle, x, y)	void gc_item_rotate_relative_with_center(GnomeCanvasItem *item, double angle, int x, int y)	
gcompris.utils.dialog(str, callback)	void gc_dialog(gchar *str, DialogBoxCallBack callback)	The callback must be a callable object
gcompris.utils.dialog_close()	void gc_dialog_close()	

Constants mapping

There is no constants in this package.

bonus.h functions and constants Mapping

Functions mapping

Python function	C equivalent	Notes
gcompris.bonus.display(BonusStatusList, BonusList)	void gc_bonus_display(BonusStatusList, BonusList)	

Python function	C equivalent	Notes
gcompris.bonus.board_finished(BoardFinishedList)	void gc_bonus_end_display(BoardFinishedList)	

Constants mapping (BonusStatusList)

Python constant	C equivalent
gcompris.bonus.LOOSE	BOARD_LOOSE
gcompris.bonus.WIN	BOARD_WIN
gcompris.bonus.DRAW	BOARD_DRAW

Constants mapping (BonusList)

Python constant	C equivalent
gcompris.bonus.TIME_CLICK_TO	TIME_CLICK_TO_BONUS
gcompris.bonus.RANDOM	BONUS_RANDOM
gcompris.bonus.SMILEY	BONUS_SMILEY
gcompris.bonus.FLOWER	BONUS_FLOWER
gcompris.bonus.TUX	BONUS_TUX
gcompris.bonus.GNU	BONUS_TUX
gcompris.bonus.LION	BONUS_LION

Constants mapping (BoardFinishedList)

Python constant	C equivalent
gcompris.bonus.FINISHED_RANDOM	BOARD_FINISHED_RANDOM
gcompris.bonus.FINISHED_TUXPLANE	BOARD_FINISHED_TUXPLANE
gcompris.bonus.FINISHED_TUXLOCO	BOARD_FINISHED_TUXLOCO
gcompris.bonus.FINISHED_TOOMANYERRORS	BOARD_FINISHED_TOOMANYERRORS

score.h functions and constants Mapping

Functions mapping

Python function	C equivalent	Notes
gcompris.score.start(style, x, y, max)	void gc_score_start(ScoreStyleList style, guint x, guint y, guint max)	
gcompris.score.end()	void gc_score_end()	
gcompris.score.set(value)	void gc_score_set(guint value)	

Constants mapping

Python constant	C equivalent
gcompris.score.STYLE_NOTE	SCORESTYLE_NOTE
gcompris.score.STYLE_LIFE	SCORESTYLE_LIFE

sound.h functions and constants Mapping

Functions mapping

Python function	C equivalent	Notes
gcompris.sound.play_ogg_list(list)	void gc_sound_play_ogg_list(GList* files)	The list argument is a GList
gcompris.sound.play_ogg(list)	void gc_sound_play_ogg(char *, ...)	The list argument is a python list containing strings
gcompris.sound.play_ogg_cb(list)	void gc_sound_play_ogg_cb((const gchar *file, GcomprisSoundCallback cb)	The list argument is a python string and a function callback. It will be called once the sound has been played.

Constants mapping

There is no constants in this package.

timer.h functions and constants Mapping

Functions mapping

Python function	C equivalent	Notes
<code>gcompris.timer.display(x, y, type, second, callback)</code>	<code>void gcompris_timer_display (int x, int y, TimerList type, int second, GcomprisTimerEnd gcomprisTimerEnd);</code>	The callback must be a callable object
<code>gcompris.timer.add(second)</code>	<code>void gc_timer_add(int second)</code>	
<code>gcompris.timer.end()</code>	<code>void gc_timer_end(void)</code>	
<code>gcompris.timer.get_remaining()</code>	<code>guint gc_timer_get_remaining()</code>	Return an integer
<code>gcompris.timer.pause(pause)</code>	<code>void gc_timer_pause(gboolean pause)</code>	

Constants mapping

Python constant	C equivalent
<code>gcompris.timer.TEXT</code>	<code>GCOMPRIS_TIMER_TEXT</code>
<code>gcompris.timer.SAND</code>	<code>GCOMPRIS_TIMER_SAND</code>
<code>gcompris.timer.BALLOON</code>	<code>GCOMPRIS_TIMER_BALLOON</code>
<code>gcompris.timer.CLOCK</code>	<code>GCOMPRIS_TIMER_CLOCK</code>

anim.h structures Mapping

The `gcompris.anim` module supplies two objects, `Animation` and `CanvasItems`, corresponding to `GcomprisAnimation` and `GcomprisAnimCanvasItem` respectively.

Animation

The `Animation` object represents a set of animation files loaded into memory; it has no methods or members, only a constructor and a destructor. The constructor takes a single argument, the name of a text file containing a space-separated list of animation files. Each animation file in the text file corresponds to an animation state; the animations states are numbered from 0 to n-1.

CanvasItem

The `CanvasItem` object is a bit more interesting than the `Animation` object. It represents an active instance of an

animation file. Its constructor takes two arguments, an `Animation` and a `GnomeCanvasGroup` (the parent of the desired active animation).

Members

Structure member	Type	Readable	Writable
<code>gnome_canvas</code>	<code>gnome.canvas.CanvasPixbuf</code>	Y	N
<code>num_states</code>	<code>int</code>	Y	N

Functions

Python function	C equivalent	Notes
<code>gcompris.anim.CanvasItem.setState(int)</code>	<code>gc_anim_set_state(GcomprisAnimCanvasItem*, int)</code>	