

---

# **Tiles**

***Release 1.0***

**Oliver Galvin**

**Jun 05, 2019**



**CONTENTS:**

<b>1</b>	<b>User Guide</b>	<b>3</b>
<b>2</b>	<b>Code Reference - main.c</b>	<b>5</b>
<b>3</b>	<b>Code Reference - modes.c</b>	<b>9</b>
<b>4</b>	<b>Credits</b>	<b>11</b>
<b>5</b>	<b>About Tiles</b>	<b>13</b>
<b>6</b>	<b>Indices</b>	<b>15</b>
	<b>Index</b>	<b>17</b>







**USER GUIDE**





## CODE REFERENCE - MAIN.C

### Defines

IMG\_DIR  
LEXICA\_EXT  
LEXICA\_DIR  
ALPHA\_EXT  
ALPHA\_DIR  
HANDY\_USE\_UNSTABLE\_API  
GDK\_DISABLE\_DEPRECATED  
GTK\_DISABLE\_DEPRECATED  
OPT (T)  
ROOT  
ROUND (x)

### Typedefs

```
typedef struct config_t config_t  
    Record of all configuration data.  
typedef struct letter_t letter_t  
typedef struct lang_t lang_t  
typedef struct root_t root_t
```

### Functions

```
void free_board (square_t ***board, int n)  
    Free the board array.  
static void free_config (config_t **config)  
    Free the config data.  
static void free_lang (lang_t **lang)  
    Free the lexicon/alphabet data.  
static void shutdown (GtkApplication *app ATTR_UNUSED, gpointer root)  
    Callback function ran when program ends - free everything.
```

```
static void quit_app(GSimpleAction *action ATTR_UNUSED, GVariant *parameter ATTR_UNUSED, g
    Ran when the 'quit' action is triggered, ends the main loop.

static void clear_board(square_t **board, int n)
    Clear/initialise the board - run before a new game starts.

static gboolean get_file(const char *filename, char **buf, gsize *buflen)
    Reads a file stored in DATADIR into a string.

static void draw_tile(cairo_t *cr, const char *letter, const char *lang, int x, int y, const config_t
    *config)
    Render the tile image for the given letter.

static void draw_text_in_square(GtkWidget *widget, cairo_t *cr, const char *text, const Pan-
    goFontDescription *font, int x, int y, const config_t *config)
    Render the text given, in the centre of the board square at the coordinates given.

static gboolean draw_board_elements(GtkWidget *widget, cairo_t *cr, gpointer root)
    Draw elements within the board: gridlines, coloured squares, and labels.

static GtkWidget *draw_board(root_t *root)
    Calculate the dimensions of the whole board, and tiles.

static gchar **list_langs(void)
    List available languages by searching the lexica directory.

static gchar **list_modes_for_lang(const gchar *lang)
    List available game modes for the given language.

static gchar **load_lexicon(const gchar *lang, const gchar *lang_var, gint *w_num)
    Load the given lexicon into an array of strings.

static letter_t *load_alphabet(const gchar *lang, const gchar *mode, gint *l_num)
    Load the given alphabet into an array of letter_t.

static root_t *setup_root(void)
    Create the initial root structure that gets passed everywhere.

static void show_prefs(GSimpleAction *action ATTR_UNUSED, GVariant *parameter ATTR_UNUSED,
    A preference dialog, for the user to set the config.

static void show_about(GSimpleAction *action ATTR_UNUSED, GVariant *parameter ATTR_UNUSED,
    A small 'About' dialog, with a description, license information and links.

static void setup_window(GtkApplication *app, gpointer root)
    Set up the window and draw the graphics.

static void setup_app(GtkApplication * app, gpointer root ATTR_UNUSED)
    Set up the GtkApplication, create the app menu and the keyboard shortcuts.

int main(int argc, char **argv)
    Main entry point: initialise libhandy, declare the GtkApplication, attach the callbacks and run the main loop.

struct config_t
    Record of all configuration data.
```

## Public Members

```
gint square_num
    Number of squares in a row/column.

gint border_size
    Thickness of border within and around the board, in pixels.
```

gint **board\_width**  
 Overall width of board, in pixels.

gint **tile\_width**  
 Width of each tile, in pixels.

double **margin**  
 Margin around the board, as a proportion of window size.

GdkRGBA **font\_colour**

GdkRGBA **border\_colour**

GdkRGBA **\*sq\_colour**

gchar **\*lang**  
 Current language used for word list.

gchar **\*lang\_var**  
 Current language variant, if applicable.

gchar **\*mode**  
 Current mode, i.e. rule set.

**struct letter\_t**

### Public Members

gchar **\*label**

gint64 **freq**

gint64 **score**

**struct lang\_t**

### Public Members

gchar **\*\*words**

gint **w\_num**

*letter\_t* **\*letters**

gint **l\_num**

**struct root\_t**

### Public Members

GtkWidget **\*window**

square\_t **\*\*board**

*config\_t* **\*config**

*lang\_t* **\*lang**



## CODE REFERENCE - MODES.C

Allocating and setting up the board array, with a function for each game mode which is run on startup of when the board is reconfigured.

**Author** Copyright (C) 2019 Oliver Galvin

### Defines

**BOARD**

**REFLECT** (*x*, *y*, *T*)

### Functions

**static** square\_t \*\***alloc\_board** (int *n*)  
Allocate memory to, and initialise, the board.

int **build\_board\_std** (square\_t \*\*\**board*)  
Set up a standard scrabble-style board arrangement.



---

CHAPTER

**FOUR**

---

**CREDITS**





## **ABOUT TILES**

Tiles is a responsive app, written in C using GTK, which is a crossword game, inspired by the likes of Scrabble and Words With Friends.

This documentation aims to offer both a brief user guide, and a developer guide documenting the code.



## INDICES

- `genindex`
- `modindex`
- `search`



## A

alloc\_board (C++ *function*), 9  
 ALPHA\_DIR (C *macro*), 5  
 ALPHA\_EXT (C *macro*), 5

## B

BOARD (C *macro*), 9  
 build\_board\_std (C++ *function*), 9

## C

clear\_board (C++ *function*), 6  
 config\_t (C++ *class*), 6  
 config\_t (C++ *type*), 5  
 config\_t::board\_width (C++ *member*), 7  
 config\_t::border\_colour (C++ *member*), 7  
 config\_t::border\_size (C++ *member*), 6  
 config\_t::font\_colour (C++ *member*), 7  
 config\_t::lang (C++ *member*), 7  
 config\_t::lang\_var (C++ *member*), 7  
 config\_t::margin (C++ *member*), 7  
 config\_t::mode (C++ *member*), 7  
 config\_t::sq\_colour (C++ *member*), 7  
 config\_t::square\_num (C++ *member*), 6  
 config\_t::tile\_width (C++ *member*), 7

## D

draw\_board (C++ *function*), 6  
 draw\_board\_elements (C++ *function*), 6  
 draw\_text\_in\_square (C++ *function*), 6  
 draw\_tile (C++ *function*), 6

## F

free\_board (C++ *function*), 5  
 free\_config (C++ *function*), 5  
 free\_lang (C++ *function*), 5

## G

GDK\_DISABLE\_DEPRECATED (C *macro*), 5  
 get\_file (C++ *function*), 6  
 GTK\_DISABLE\_DEPRECATED (C *macro*), 5

## H

HANDY\_USE\_UNSTABLE\_API (C *macro*), 5

## I

IMG\_DIR (C *macro*), 5

## L

lang\_t (C++ *class*), 7  
 lang\_t (C++ *type*), 5  
 lang\_t::l\_num (C++ *member*), 7  
 lang\_t::letters (C++ *member*), 7  
 lang\_t::w\_num (C++ *member*), 7  
 lang\_t::words (C++ *member*), 7  
 letter\_t (C++ *class*), 7  
 letter\_t (C++ *type*), 5  
 letter\_t::freq (C++ *member*), 7  
 letter\_t::label (C++ *member*), 7  
 letter\_t::score (C++ *member*), 7  
 LEXICA\_DIR (C *macro*), 5  
 LEXICA\_EXT (C *macro*), 5  
 list\_langs (C++ *function*), 6  
 list\_modes\_for\_lang (C++ *function*), 6  
 load\_alphabet (C++ *function*), 6  
 load\_lexicon (C++ *function*), 6

## M

main (C++ *function*), 6

## O

OPT (C *macro*), 5

## R

REFLECT (C *macro*), 9  
 ROOT (C *macro*), 5  
 root\_t (C++ *class*), 7  
 root\_t (C++ *type*), 5  
 root\_t::board (C++ *member*), 7  
 root\_t::config (C++ *member*), 7  
 root\_t::lang (C++ *member*), 7  
 root\_t::window (C++ *member*), 7  
 ROUND (C *macro*), 5

## S

`setup_root (C++ function), 6`

`setup_window (C++ function), 6`