Zappy SERVER

Generated by Doxygen 1.9.6

1 Class Index 1

1.1 Class List . . . 1

2 File Index 3

2.1 File List . . . 3

3 Class Documentation 5

3.1 s_app Struct Reference . . . 5

3.2 s_client Struct Reference . . . 5

3.3 s_game Struct Reference . . . 5

3.4 s_gui Struct Reference . . . 6

3.5 s_ia Struct Reference . . . 6

3.6 s_incantation_info Struct Reference . . . 6

3.7 s_inventory Struct Reference . . . 7

3.8 s_list Struct Reference . . . 7

3.9 s_list_node Struct Reference . . . 7

3.10 s_node_data Union Reference . . . 8

3.11 s_parsing Struct Reference . . . 8

3.11.1 Detailed Description . . . 8

3.12 s_server Struct Reference . . . 9

3.13 s_team Struct Reference . . . 9

3.14 s_time_info Struct Reference . . . 9

3.15 s_vector2i Struct Reference . . . 10

3.16 tile_s Struct Reference . . . 10

3.16.1 Detailed Description . . . 10

4 File Documentation 11

4.1 ai.h . . . 11

4.2 command_ai.h . . . 12

4.3 look.h . . . 12

4.4 stuck.h . . . 13

4.5 team.h . . . 13

4.6 app.h . . . 14

4.7 game.h . . . 14

4.8 gui.h . . . 15

4.9 list.h . . . 15

4.10 type.h . . . 15

4.11 map.h . . . 16

4.12 parsing.h . . . 16

4.13 rules.h . . . 17

4.14 client.h . . . 17

4.15 server.h . . . 18

4.16 types.h . . . 19

Index **21**

Chapter 1

Class Index

1.1 Class List

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

s_app 5
s_client 5
s_game 5
s_gui 6
s_ia 6
s_incantation_info 6
s_inventory 7
s_list 7
s_list_node 7
s_node_data 8
s_parsing
Struct for parsing the arguments 8
s_server 9
s_team 9
s_time_info 9
s_vector2i 10
tile_s
Struct for a map's tile 10

Chapter 2

File Index

2.1 File List

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

/home/tjerome-rocher/Desktop/Tek2/Zappy/server/include/parsing.h 16
/home/tjerome-rocher/Desktop/Tek2/Zappy/server/include/rules.h 17
/home/tjerome-rocher/Desktop/Tek2/Zappy/server/include/types.h 19
/home/tjerome-rocher/Desktop/Tek2/Zappy/server/include/ai/ai.h 11
/home/tjerome-rocher/Desktop/Tek2/Zappy/server/include/ai/stuck.h 13
/home/tjerome-rocher/Desktop/Tek2/Zappy/server/include/ai/team.h 13
/home/tjerome-rocher/Desktop/Tek2/Zappy/server/include/ai/cmd/command_ai.h 12
/home/tjerome-rocher/Desktop/Tek2/Zappy/server/include/ai/cmd/look.h 12
/home/tjerome-rocher/Desktop/Tek2/Zappy/server/include/app/app.h 14
/home/tjerome-rocher/Desktop/Tek2/Zappy/server/include/game/game.h 14
/home/tjerome-rocher/Desktop/Tek2/Zappy/server/include/gui/gui.h 15
/home/tjerome-rocher/Desktop/Tek2/Zappy/server/include/list/list.h 15
/home/tjerome-rocher/Desktop/Tek2/Zappy/server/include/list/type.h 15
/home/tjerome-rocher/Desktop/Tek2/Zappy/server/include/map/map.h 16
/home/tjerome-rocher/Desktop/Tek2/Zappy/server/include/server/client.h 17
/home/tjerome-rocher/Desktop/Tek2/Zappy/server/include/server/server.h 18

Chapter 3

Class Documentation

3.1 s_app Struct Reference

Collaboration diagram for s_app:

3.2 s_client Struct Reference

Public Attributes

- size_t **fd**

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

- /home/tjerome-rocher/Desktop/Tek2/Zappy/server/include/server/client.h

3.3 s_game Struct Reference

Collaboration diagram for s_game:

Public Attributes

- size_t **height**
- size_t **width**
- tile_t ∗∗ **map**
- struct timeval **start**
- struct timeval **start_food**
- int **freq**

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

- /home/tjerome-rocher/Desktop/Tek2/Zappy/server/include/game/game.h

3.4 s_gui Struct Reference

Collaboration diagram for s_gui:

Public Attributes

- size_t **fd**
- list_t ∗ **list_messages**

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

- /home/tjerome-rocher/Desktop/Tek2/Zappy/server/include/gui/gui.h

3.5 s_ia Struct Reference

Collaboration diagram for s_ia:

Public Attributes

- size_t **fd**
- list_t ∗ **list_command**
- list_t ∗ **list_messages**
- vector2i_t ∗ **position**
- orientation_t **direction**
- inventory_t ∗ **inventory**
- incantation_info_t ∗ **incantation**
- size_t **level**
- time_info_t ∗ **time**

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

- /home/tjerome-rocher/Desktop/Tek2/Zappy/server/include/ai/ai.h

3.6 s_incantation_info Struct Reference

Public Attributes

- bool **status_incantation**
- size_t **target_level**

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

- /home/tjerome-rocher/Desktop/Tek2/Zappy/server/include/ai/ai.h

3.7 s_inventory Struct Reference

Public Attributes

- size_t **food**
- size_t **linemate**
- size_t **deraumere**
- size_t **sibur**
- size_t **mendiane**
- size_t **phiras**
- size_t **thystame**

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

- /home/tjerome-rocher/Desktop/Tek2/Zappy/server/include/ai/ai.h

3.8 s_list Struct Reference

Collaboration diagram for s_list:

Public Attributes

- list_node_t ∗ **first**
- list_node_t ∗ **last**
- size_t **len**

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

- /home/tjerome-rocher/Desktop/Tek2/Zappy/server/include/list/type.h

3.9 s_list_node Struct Reference

Collaboration diagram for s_list_node:

Public Attributes

- node_data_t **data**
- struct s_list_node ∗ **next**
- struct s_list_node ∗ **prev**

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

- /home/tjerome-rocher/Desktop/Tek2/Zappy/server/include/list/type.h

3.10 s_node_data Union Reference

Collaboration diagram for s_node_data:

Public Attributes

- ia_t ∗ **ai**
- gui_t ∗ **gui**
- client_t ∗ **client**
- team_t ∗ **team**
- char ∗ **message**
- char ∗ **command**
- vector2i_t ∗ **coord**

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

- /home/tjerome-rocher/Desktop/Tek2/Zappy/server/include/list/type.h

3.11 s_parsing Struct Reference

Struct for parsing the arguments.

```
#include <types.h>
```

Public Attributes

- int **port**
- int **width**
- int **height**
- int **clientsNb**
- int **freq**
- char ∗∗ **names**

3.11.1 Detailed Description

Struct for parsing the arguments.

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

- /home/tjerome-rocher/Desktop/Tek2/Zappy/server/include/types.h

3.12 s_server Struct Reference

Public Attributes

- fd_set **read_fds**
- fd_set **write_fds**
- int **fd**
- socklen_t **addrlen**
- struct sockaddr_in **addr**

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

- /home/tjerome-rocher/Desktop/Tek2/Zappy/server/include/server/server.h

3.13 s_team Struct Reference

Collaboration diagram for s_team:

Public Attributes

- list_t ∗ **list_ai**
- list_t ∗ **egg_position**
- char ∗ **name**
- size_t **max_place**

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

- /home/tjerome-rocher/Desktop/Tek2/Zappy/server/include/ai/team.h

3.14 s_time_info Struct Reference

Public Attributes

- bool **stuck**
- struct timeval **start_stuck**
- double **total_stuck**
- struct timeval **start_life**
- double **total_life**

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

- /home/tjerome-rocher/Desktop/Tek2/Zappy/server/include/ai/ai.h

3.15 s_vector2i Struct Reference

Public Attributes

- int **x**
- int **y**

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

- /home/tjerome-rocher/Desktop/Tek2/Zappy/server/include/list/type.h

3.16 tile_s Struct Reference

Struct for a map's tile.

```
#include <map.h>
```

Public Attributes

- size_t **food**
- size_t **linemate**
- size_t **deraumere**
- size_t **sibur**
- size_t **mendiane**
- size_t **phiras**
- size_t **thystame**

3.16.1 Detailed Description

Struct for a map's tile.

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

- /home/tjerome-rocher/Desktop/Tek2/Zappy/server/include/map/map.h

Chapter 4

File Documentation

4.1 ai.h

```c
/*
** EPITECH PROJECT, 2024
** Zappy Server
** File description:
** AI
*/

#pragma once

#include <stddef.h>
#include <stdlib.h>
#include <stdbool.h>

#include "list/list.h"

typedef struct s_app app_t;

typedef enum {
    NORTH,
    SOUTH,
    EAST,
    WEST,
} orientation_t;

typedef struct s_inventory {
    size_t food;
    size_t linemate;
    size_t deraumere;
    size_t sibur;
    size_t mendiane;
    size_t phiras;
    size_t thystame;
} inventory_t;

typedef struct s_time_info {
    bool stuck;
    struct timeval start_stuck;
    double total_stuck;
    struct timeval start_life;
    double total_life;
} time_info_t;

typedef struct s_incantation_info {
    bool status_incantation;
    size_t target_level;
} incantation_info_t;

typedef struct s_ia {
    size_t fd;
    list_t *list_command;
    list_t *list_messages;
    vector2i_t *position;
    orientation_t direction;
    inventory_t *inventory;
    incantation_info_t *incantation;
    size_t level;
    time_info_t *time;
} ia_t;
```

```c

ia_t *create_ia(app_t *app, int fd, team_t *team);

void add_ia(app_t *app, size_t fd, char *line);

ia_t *find_ia(app_t *app, size_t fd);

void check_die(app_t *app);
```

4.2 command_ai.h

```c
/*
** EPITECH PROJECT, 2024
** Zappy Server
** File description:
** AI Command
*/

#pragma once

#include "ai/ai.h"

#define BEGIN_INCANTATION 1
#define END_INCANTATION 2

void command_ai_handler(app_t *app, ia_t *ai, char *line);

bool move_command(app_t *app, ia_t *ai, char *line);

bool object_info_command(app_t *app, ia_t *ai, char *line);

bool other_command(app_t *app, ia_t *ai, char *line);

void eject_cmd(app_t *app, ia_t *ai);

void eject_player(app_t *app, ia_t *ai);

void eject_egg(app_t *app, ia_t *ai);

void dead_response(app_t *app);

void fork_cmd(app_t *app, ia_t *ai);

void connect_nbr_cmd(app_t *app, ia_t *ai);

void take_cmd(app_t *app, ia_t *ai);

void set_cmd(app_t *app, ia_t *ai);

bool broadcast_command(app_t *app, ia_t *ai, char *line);

bool incantation_command(app_t *app, ia_t *ai, char *line);

bool check_incantation(app_t *app, ia_t *ai, int status);

void update_status(app_t *app, ia_t *ai, int update_status);

void level_up(app_t *app, ia_t *ai);

void add_command_to_list(ia_t *ai, char *line);

void treat_command(app_t *app);

void destroy_command_list(list_t *command_list);
```

4.3 look.h

```c
/*
** EPITECH PROJECT, 2024
** Zappy Server
** File description:
** AI Command
*/

#pragma once

#include "ai/ai.h"

#define PLAYER_STRING " player"
```

```
#define EGG_STRING " egg"
#define FOOD_STRING " food"
#define LINEMATE_STRING " linemate"
#define DERAUMERE_STRING " deraumere"
#define SIBUR_STRING " sibur"
#define MENDIANE_STRING " mendiane"
#define PHIRAS_STRING " phiras"
#define THYSTAME_STRING " thystame"

void look_cmd(app_t *app, ia_t *ai);

void check_player(vector2i_t *pos, app_t *app, char **reply, ia_t *ai);

void check_egg(vector2i_t *pos, app_t *app, char **reply);

void check_resources(vector2i_t *pos, app_t *app, char **reply);

void look_north(app_t *app, ia_t *ai, int index_line, char **reply);

void look_east(app_t *app, ia_t *ai, int index_line, char **reply);

void look_south(app_t *app, ia_t *ai, int index_line, char **reply);

void look_west(app_t *app, ia_t *ai, int index_line, char **reply);
```

4.4 stuck.h

```
/*
** EPITECH PROJECT, 2024
** Zappy Server
** File description:
** Game struct
*/

#pragma once

#include <stddef.h>
#include <stdlib.h>
#include <stdbool.h>
#include <sys/time.h>

#include "list/list.h"

typedef struct s_app app_t;

double time_elapsed(struct timeval *time);

void set_time_stuck(ia_t *ai, double total_stuck);

void treat_stuck(app_t *app);
```

4.5 team.h

```
/*
** EPITECH PROJECT, 2024
** Zappy Server
** File description:
** Team struct
*/

#pragma once

#include <stddef.h>
#include <stdlib.h>
#include <stdbool.h>

#include "list/list.h"

typedef struct s_app app_t;

typedef struct s_team {
    list_t *list_ai;
    list_t *egg_position;
    char *name;
    size_t max_place;
} team_t;

team_t *create_team(app_t *app, char *name, size_t max_place);

```

```
void add_team(app_t *app, char *team_name, size_t max_place);

void display_egg_position(app_t *app);

void add_egg(list_t *eggs, int random_x, int random_y);

team_t *find_team(app_t *app, size_t fd);

void destroy_team(list_t *teams_list);
```

4.6 app.h

```
/*
** EPITECH PROJECT, 2024
** Zappy Server
** File description:
** App
*/

#pragma once

#include <time.h>
#include <stdio.h>
#include <stdlib.h>
#include <stdarg.h>
#include <string.h>
#include <signal.h>

#include "ai/ai.h"
#include "ai/team.h"
#include "gui/gui.h"
#include "parsing.h"
#include "ai/stuck.h"
#include "list/list.h"
#include "game/game.h"
#include "server/server.h"

typedef struct s_app {
    list_t *gui_list;
    list_t *teams_list;
    list_t *clients_list;
    server_t *server;
    game_t *game;
} app_t;

app_t *create_app(parsing_t *parsing);

void destroy_app(app_t *app);
```

4.7 game.h

```
/*
** EPITECH PROJECT, 2024
** Zappy Server
** File description:
** Game struct
*/

#pragma once

#include "map/map.h"

typedef struct s_game {
    size_t height;
    size_t width;
    tile_t **map;
    struct timeval start;
    struct timeval start_food;
    int freq;
} game_t;

game_t *create_game(int height, int width, int freq);

void spawn_ressources(app_t *app);

bool check_win(app_t *app);

void destroy_game(game_t *game);
```

4.8 gui.h

```c
/*
** EPITECH PROJECT, 2024
** Zappy Server
** File description:
** Gui struct
*/

#pragma once

#include <stddef.h>
#include <stdbool.h>

#include "list/list.h"

typedef struct s_gui {
    size_t fd;
    list_t *list_messages;
} gui_t;

gui_t *create_gui(int fd);

void add_gui(app_t *app, size_t fd, char *line);

gui_t *find_gui(app_t *app, size_t fd);

void destroy_gui(list_t *gui_list);
```

4.9 list.h

```c
/*
** EPITECH PROJECT, 2024
** Zappy Server
** File description:
** Function about list
*/

#pragma once

#include <stdbool.h>

#include "type.h"

list_t *list_new(void);

void list_free(list_t *list);

void list_delete(list_t *list, list_node_t *node);

void list_remove_front(list_t *list);

void list_remove_back(list_t *list);

bool list_add_back(list_t *list, node_data_t data);

bool list_add_front(list_t *list, node_data_t data);
```

4.10 type.h

```c
/*
** EPITECH PROJECT, 2024
** Zappy Server
** File description:
** Type of List
*/

#pragma once

#include <stddef.h>

typedef struct s_ia ia_t;
typedef struct s_gui gui_t;
typedef struct s_team team_t;
typedef struct s_client client_t;

typedef struct s_vector2i {
    int x;
    int y;
} vector2i_t;

typedef union s_node_data {
    ia_t *ai;
    gui_t *gui;
    client_t *client;
    team_t *team;
    char *message;
    char *command;
    vector2i_t *coord;
} node_data_t;

typedef struct s_list_node {
    node_data_t data;
    struct s_list_node *next;
    struct s_list_node *prev;
} list_node_t;

typedef struct s_list {
    list_node_t *first;
    list_node_t *last;
    size_t len;
} list_t;
```

4.11 map.h

```
/*
** EPITECH PROJECT, 2024
** Zappy Server
** File description:
** map
*/

#pragma once

#include <stddef.h>
#include <stdbool.h>

#define FOOD_DENSITY 0.5
#define LINEMATE_DENSITY 0.3
#define DERAUMERE_DENSITY 0.15
#define SIBUR_DENSITY 0.1
#define MENDIANE_DENSITY 0.1
#define PHIRAS_DENSITY 0.08
#define THYSTAME_DENSITY 0.05

enum entity_type_e {
    EGG,
    FOOD,
    LINEMATE,
    DERAUMERE,
    SIBUR,
    MENDIANE,
    PHIRAS,
    THYSTAME,
    NONE
};

typedef struct tile_s {
    size_t food;
    size_t linemate;
    size_t deraumere;
    size_t sibur;
    size_t mendiane;
    size_t phiras;
    size_t thystame;
} tile_t;

tile_t **create_map(int width, int height);

void free_map(tile_t **map, int height);

void display_map(tile_t **map, int height, int width);
```

4.12 parsing.h

```
/*
** EPITECH PROJECT, 2024
** Zappy
** File description:
** Header file for parsing the arguments passed to the server.
*/

#pragma once

#define HELP_FLAG_LABEL "--help"
#define HELP_FLAG "-h"
#define PORT_FLAG "-p"
#define WIDTH_FLAG "-x"
#define HEIGHT_FLAG "-y"
#define TEAMS_NAMES_FLAG "-n"
#define CLIENTS_FLAG "-c"
#define FREQUENCY_FLAG "-f"

#define NB_ARGS_MIN 13
#define NB_ARGS_HELP 2

#include "types.h"

#include <stdbool.h>

parsing_t *parse_arg(int ac, char **av);

int handle_help(int ac, char **av);

int parse_positive_int_arg(char *arg);

int parse_client(char **arg, int *pos, parsing_t *parsing);

int parse_frequency(char **arg, int *pos, parsing_t *parsing);

int parse_height(char **arg, int *pos, parsing_t *parsing);

int parse_port(char **arg, int *pos, parsing_t *parsing);

int parse_width(char **arg, int *pos, parsing_t *parsing);

int parse_names(char **arg, int *pos, parsing_t *parsing);

void destroy_parsing(parsing_t *parsing);
```

4.13 rules.h

```
/*
** EPITECH PROJECT, 2024
** Zappy
** File description:
** Rules for the server
*/

#pragma once

enum RETURN_CODES {
    CODE_ERROR_MISSING_ARG = -1,
    CODE_ERROR_INVALID_ARG = -2,
    CODE_ERROR_WRONG_FLAG = -3,
    CODE_ERROR_INVALID_NUMBER = -4,
    CODE_ERROR_MALLOC_FAILED = -5,
    CODE_HELP_SUCCESS = 1,
    CODE_SUCCESS = 0,
    CODE_FAILLURE = 84
};
```

4.14 client.h

```
/*
** EPITECH PROJECT, 2024
** Zappy Server
** File description:
** Client
*/

#pragma once

#include <stddef.h>
#include <stdlib.h>
#include <stdbool.h>

```

```c
#include "app/app.h"

typedef struct s_client {
    size_t fd;
} client_t;

client_t *create_client(int fd);

bool its_client(app_t *app, size_t fd);

list_node_t *find_client(list_t *clients_list, size_t fd);

void destroy_client(list_t *client_list);
```

4.15 server.h

```c
/*
** EPITECH PROJECT, 2024
** Zappy Server
** File description:
** Server struct
*/

#pragma once

#include <stdio.h>
#include <stddef.h>
#include <stdlib.h>
#include <unistd.h>
#include <stdbool.h>
#include <sys/select.h>
#include <netinet/in.h>

#include "list/list.h"

#define LISTEN_NUMBER 100
#define WELCOME_MESSAGE "WELCOME\n"
#define WELCOME_MESSAGE_LEN 8

typedef struct s_app app_t;

typedef struct s_server {
    fd_set read_fds;
    fd_set write_fds;
    int fd;
    socklen_t addrlen;
    struct sockaddr_in addr;
} server_t;

void destroy_server(server_t *server);

server_t *create_server(size_t port);

bool server_run(app_t *app);

bool server_connection_handler(app_t *app, size_t fd);

bool server_data_handler(app_t *app, size_t fd);

char *read_line(int fd);

void server_quit_handler(app_t *app, size_t fd);

bool write_message(list_t *list_messages, size_t fd);

void add_message(list_t *list, char *message);

char *format_string(const char *format, ...);

void server_reset_fd(app_t *app);

void handle_request(app_t *app, size_t fd, char *line);

char *append_char(char *line, char current_char);

void handle_client_read(app_t *app, int fd);

void handle_client_write(app_t *app, int fd);

vector2i_t *create_vector2i(int x, int y);

void concatenate_strings(char **str1, char *str2);

void destroy_message_list(list_t *message_list);
```

4.16 types.h

```
/*
** EPITECH PROJECT, 2024
** Zappy
** File description:
** Types for the server
*/

#pragma once

typedef struct s_parsing {
    int port;
    int width;
    int height;
    int clientsNb;
    int freq;
    char **names;
} parsing_t;
```

Index

/home/tjerome-rocher/Desktop/Tek2/Zappy/server/include/ai/ai.h, 11
/home/tjerome-rocher/Desktop/Tek2/Zappy/server/include/ai/cmd/command_ai.h, 12
/home/tjerome-rocher/Desktop/Tek2/Zappy/server/include/ai/cmd/look.h, 12
/home/tjerome-rocher/Desktop/Tek2/Zappy/server/include/ai/stuck.h, 13
/home/tjerome-rocher/Desktop/Tek2/Zappy/server/include/ai/team.h, 13
/home/tjerome-rocher/Desktop/Tek2/Zappy/server/include/app/app.h, 14
/home/tjerome-rocher/Desktop/Tek2/Zappy/server/include/game/game.h, 14
/home/tjerome-rocher/Desktop/Tek2/Zappy/server/include/gui/gui.h, 15
/home/tjerome-rocher/Desktop/Tek2/Zappy/server/include/list/list.h, 15
/home/tjerome-rocher/Desktop/Tek2/Zappy/server/include/list/type.h, 15
/home/tjerome-rocher/Desktop/Tek2/Zappy/server/include/map/map.h, 16
/home/tjerome-rocher/Desktop/Tek2/Zappy/server/include/parsing.h, 16
/home/tjerome-rocher/Desktop/Tek2/Zappy/server/include/rules.h, 17
/home/tjerome-rocher/Desktop/Tek2/Zappy/server/include/server/client.h, 17
/home/tjerome-rocher/Desktop/Tek2/Zappy/server/include/server/server.h, 18
/home/tjerome-rocher/Desktop/Tek2/Zappy/server/include/types.h, 19

s_app, 5
s_client, 5
s_game, 5
s_gui, 6
s_ia, 6
s_incantation_info, 6
s_inventory, 7
s_list, 7
s_list_node, 7
s_node_data, 8
s_parsing, 8
s_server, 9
s_team, 9
s_time_info, 9
s_vector2i, 10

tile_s, 10