

Dokumentacja projektu "Gra w życie"

Damian Kolaska

Instrukcja uzytkowania

1. Wywołanie programu.

`./a.out i input_file o output_file d interval`

- input_file - nazwa pliku, z ktorego program ma pobrac plansze
- output_file - nazwa pliku, w ktorym program ma zapisywac stan planszy
- interval - co ile krokow ma zapisywac stan planszy

parametry sa opcjonalne!

2. Zamykanie programu

Aby zamknac program nalezy, zatrzymawszy uprzednio symulacje, wcisnac klawisz 'q'. Program zapyta sie nastepnie czy napewno chcemy wyjsc z gry, jesli nie zostanie uruchomiona nowa gra, w ktorej bedziemy mogli zmienic parametry mapy.

Struktura programu

- plik **main.c** – zawiera petle gry
- plik **menu.c** - zawiera funkcje obslugi interfejsu uzytkownika
- plik **input.c** – zawiera metody sluzace do pobrania danych od uzytkownika
- plik **map.c** - zawiera metody obslugi planszy

ZMIENNE GLOBALNE

volatile STOP_FLAG – flaga informujaca program o tym czy uzytkownik zdecydowal sie zatrzymac symulacje

int w – szerokosc planszy

int h - wysokosc planszy

int steps – ilosc krokow symulacji

volatile int interval – odstep pomiedzy krokami

struct Vector - wektor [x, y]

Vector * cursor - zmienna przechowujaca aktualna pozycje kursora

Vector * anchor - zmienna przechowujaca "punkt zaczepienia" planszy

char * input_file - wskaznik na obszar w pamieci, w ktorym przechowywana jest nazwa pliku wejscowego

char * output_file - analogicznie

int delta_steps - informacja o tym jak czesto zapisywac dane do pliku

char filetype - typ pliku wyjsciowego (.html lub .txt)

MAIN.C

FUNKCJE

void free_globals() - zwalnia zmienne globalne

void wait(double msec) – czeka ilosc sekund okreslona w parametrze msec

void init_vals(char ** map, int argc, char * argv[])

Przyjmuje

map - wskaznik podwojny na tablice char

argc - liczba argumentow main'a

argv - tablica argumentow main'a

void init_screen() - inicjalizuje okno i wszelkie parametry z nim zwiazane

void dw_output(FILE ** file) - zapytuje uzytkownika czy ten chce wygenerowac

plik wyjsciowy

Przyjmuje
file - wskaźnik do zmiennej plikowej pliku wyjściowego
void * game(void * map) – startuje grę na planszy.
Przyjmuje
wskaźnik void na tablicy char map
Zwraca
wskaźnik NULL
void * check_for_stop() - sprawdza czy użytkownik nie chce zatrzymać symulacji.
Zwraca wskaźnik NULL.

INPUT.C

int clear_buffer() - czyszczy bufor standardowego wejścia
Zwraca
1 - jeśli bufor był czysty
0 - jeśli bufor zawierał znaki różne od spacji i znaku nowej linii
int check_buffer(char * buf, int len)
Przyjmuje
buf - bufor w postaci tablicy char
len - długość bufora
Zwraca
1 - jeśli bufor zawierał prawidłowo zapisaną liczbę całkowitą
0 - w przeciwnym wypadku
void get_value(int * val, int min, int max, const char * msg, char * val_name) - pobiera od użytkownika wartość zmiennej val.
Przyjmuje
val - wskaźnik do zmiennej, której wartość ma pobrać
min - wartość minimalna
max - wartość maksymalna
msg - wiadomość, która należy wyświetlić w przypadku podania nieprawidłowej wartości
val_name - nazwa zmiennej
int ask(const char * question) - zadaje pytanie typu (T/N)
Przyjmuje
question - treść pytania
void parse_args(int argc, char * argv[]) - analizuje argumenty podane przy wywołaniu programu
void get_int_from_file(FILE * file, int * val, int len, int min, int max, const char * msg_nn, const char * msg_tl, char * val_name) - pobiera zmienną typu int z pliku
Przyjmuje
msg_nn - wiadomość, która należy wyświetlić jeżeli zmienna okaże się nie być liczbą
msg_tl - jeżeli zmienna będzie zbyt duża
void get_map(FILE * file, const char * map) - pobiera planszę z pliku
Przyjmuje
file - plik, z którego należy wczytać mapę
mode - tryb otwarcia pliku
void open_file(FILE ** file, const char * mode) - otwiera plik w trybie 'mode'
Przyjmuje
file - plik do otwarcia
mode - tryb otwarcia pliku
void get_game_values(char ** map, int argc, char * argv[]) – pyta użytkownika czy ten chce pobrać dane z pliku. Jeśli nie użytkownik powinien wprowadzić je ręcznie.

int check_extension(char * filename) - sprawdza rozszerzenie pliku 'filename'
Zwraca
1 - jeśli rozszerzenie było poprawne (.txt, .html)
0 - w przeciwnym wypadku

MENU.C

void print_menu(int if_restart) - wyświetla menu tekstowe
Przyjmuje
if_restart - flaga informująca o tym jaki typ menu wyświetlić
void move_cursor(int c) - przesuwa kursor
Przyjmuje
c - znak z klawiatury
int menu(char * map) - wyświetla plansze wraz z menu tekstowym
Zwraca
1 - jeśli użytkownik zdecydował się rozpocząć grę
0 - w przeciwnym wypadku

MAP.C

void map_randomize(char * map, int len) - losuje rozmieszczenie komorek na planszy
void map_update(char * map) - aktualizuje plansze (bez obramowania)
void map_draw(char * map) - aktualizuje całość planszy
void save_to_html(FILE * file, char * map) - zapisuje stan planszy do pliku html
void save_to_txt(FILE * file, char * map) - zapisuje stan planszy do pliku txt
void simulate(char * map) - symuluje jedną turę gry