

Dokumentacja projektu “Gra w życie”

Damian Kolaska

Struktura programu

- plik **main.c** – zawiera petle gry I funkcje obslugi interfejsu uzytkownika
 - plik naglowkowy **input.h** wraz z plikiem **input.c** – metody sluzace do pobrania danych od uzytkownika
-

main.c

MAKRA

- **DELTA** – reprezentuje wektor o jaki nalezy przesunac sie po tablicy ‘map’ aby wyluskac interesujacy nas element. Uzyte w funkcji **menu**.

FUNKCJE

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

void print_menu(int w, int h) – wypisuje menu. Jako parametr przyjmuje wysokosc planszy **h**.

Vector * menu(char * map, int w, int h, Vector * cursor, Vector * anchor, Vector * game_val) - zarzadza logika menu. Pobiera parametry gry: **ilosc krokow** I **czas pomiedzy krokami**. Wywoluje funkcje inicjalizujace ulozenie komorek, zawiera mechanizm umozliwiajacy reczne wprowadzenie ulozenia.

Zwraca wektor [ilosc krokow, odstep pomiedzy]

Parametry:

- **map** – wskaznik do tablicy char reprezentujaca mape
- **w, h** – szerokosc I wysokosc mapy
- **cursor** – wskaznik do wektora zawierajacego obecna pozycje kursora
- **anchor** – wskaznik do wektora reprezentujacego punkt zaczepienia lewego gornego rogu mapy
- **game_val** – wektor o postaci [ilosc krokow, odstep pomiedzy krokami]

void map_update(char * map, int w, int h, Vector * cursor, Vector * anchor) – rysuje kompletna plansze (z obramowaniami).

Jako parametry przyjmuje:

- **map** – wskaznik do tablicy char reprezentujaca mape
- **w, h** – szerokosc I wysokosc mapy
- **cursor** – wskaznik do wektora zawierajacego obecna pozycje kursora
- **anchor** – wskaznik do wektora reprezentujacego punkt zaczepienia lewego gornego rogu mapy

void map_draw(char * map, int w, int h, Vector * cursor, Vector * anchor) – rysuje mape (tylko czesc zawierajaca komorki)

Przyjmuje te same parametry co **map_update**

void map_reset(char * map, int len) – resetuje mape. Parametry to tablica char I jej dlugosc

void map_randomize(char * map, int len) – losuje ulozenie komorek na mapie.

void simulate(char * map, int w, int h) – symuluje jedna ture gry

void game(int w, int h) – startuje gre na planszy o szerokosci **w** I wysokosci **h**
input.c

input.c

struct Vector – wektor o wspolrzecznych [x, y]

void clear_buffer() - czysci bufor standardowego wejscia

void clear_str(char * str) – zeruje ciag znakow **str**

int ctod(char * str) – konwertuje ciag **str** na **int**. Zwraca przekonwertowana wartosc **int**

int getint(char * str, int n) – pobiera ciag **str** za pomoca funkcji biblioteki **ncurses getch()**

Parametry:

str – wskaznik do tablicy w ktorej zostanie zapisany ciag

n – maksymalna dlugosc ciagu

Zwraca ciag **str** przekonwertowany do typu **int**

void get_map_dimensions(int * w, int * h) – pobiera rozmiar planszy ze standardowego wejscia.

Parametry to wskazniki do zmiennych przechowujacych: szerokosc – **w** I wysokosc – **h** planszy.

void get_game_values(Vector * game_val) – pobiera **ilosc krokow** I **odstep** I zapisuje je w zmiennej **game_val**