

# Dokumentacja projektu Mrówka Langtona

ver. 1.0



Mikołaj Kubik 291083, Karol Stasiak 291107

Grupa 15

3 kwietnia 2018

wersja	zmiany	kto	kiedy
0.1	Stworzenie dokumentu	KS	21.03.2018
0.2	Edycja sekcji	MK	02.04.2018
0.3	Edycja sekcji	KS	03.04.2018
1.0	Zatwierdzenie dokumentu	MK	03.04.2018



Politechnika Warszawska  
Wydział Elektryczny

## Spis treści

<b>1</b>	<b>Zasada działania mrówki</b>	<b>2</b>
<b>2</b>	<b>Założenia projektu</b>	<b>2</b>
<b>3</b>	<b>Domyślne ustawienia</b>	<b>3</b>
<b>4</b>	<b>Uruchamianie programu</b>	<b>3</b>
<b>5</b>	<b>Obsługa błędów</b>	<b>4</b>
<b>6</b>	<b>Budowa programu</b>	<b>4</b>
<b>7</b>	<b>Opis poszczególnych modułów</b>	<b>5</b>
7.1	ant.c . . . . .	5
7.2	matrix.c . . . . .	5
7.3	exporttopng.c . . . . .	5
7.4	error.c . . . . .	5
7.5	main.c . . . . .	5
<b>8</b>	<b>Przykładowe wywołanie</b>	<b>6</b>

## 1 Zasada działania mrówki

1. Jeśli znajdzie się na polu białym to obraca się w lewo (o kąt prosty), zmienia kolor pola na swój kolor i przechodzi na następną komórkę;
2. Jeśli znajduje się na polu zakolorowanym to obraca się w prawo (o kąt prosty), zmienia kolor pola na biały i przechodzi na następną komórkę;
3. Porusza się na nieskończonej planszy podzielonej na kwadratowe komórki (zakolorowane lub nie), o rozmiarze 1 px każda;
4. Mrówki startują z wylosowanego miejsca na planszy;

## 2 Założenia projektu

1. Program obrazuje symulację mrówki Langtona.
2. Program może zostać uruchomiony zarówno z ustawieniami domyślnymi, jak i podanymi przez użytkownika
3. Jeżeli użytkownik poda błędne argumenty, program zadziała zgodnie z argumentami domyślnymi dla danego parametru oraz wyświetli błąd.
4. Program może wyświetlić również animację ruchu mrówek w terminalu.
5. Użytkownik może podać dane zarówno z linii komend, jak i pliku tekstowego.
6. Maksymalne wymiary planszy wynoszą  $n \times m$ .
7. Program pozwala symulować zachowanie do  $x$  mrówek.
8. Mrówki mogą wykonać maksymalnie  $x$  kroków.
9. Stan planszy po wykonaniu zadania zostaje zapisany do pliku png.
10. Po dojściu do skraju planszy mrówka automatycznie przechodzi na jej przeciwległy brzeg.
11. Początkowe współrzędne mrówek są ustawiane losowo w przypadku podawania argumentów z linii komend.
12. Początkowe współrzędne mrówek są pobierane z pliku, w przypadku podawania argumentów z pliku tekstowego.

### 3 Domyślne ustawienia

- Wymiary planszy: 184 x 184
- Liczba mrówek: 1
- Ilość kroków: 11000
- Współrzędne początkowe: Losowe

### 4 Uruchamianie programu

Argumenty dla programu (rozmiar planszy oraz liczba mrówek) mogą zostać podane zarówno z terminalu, jak i pliku tekstowego. Zastosowanie podanych wyżej przedrostków x (x to dany przedrostek) pozwoli na wpisywanie danych bez konieczności zachowania odpowiedniej kolejności.

- Nazwa wywoływanego programu **./ant**
- Szerokość planszy **-w**
- Wysokość planszy **-h**
- Ilość mrówek **-q**
- Ilość kroków **-n**
- Czy wyświetlić animację w terminalu **-a**
- Nazwa pliku w którym ma zostać zapisany obraz **-o**
- Nazwa pliku z danymi **-i**

## 5 Obsługa błędów

- Jeżeli wśród argumentów znajdzie się zmienna niebędąca liczbą, program zakończy swoje działanie oraz powiadomi o błędzie.
- Jeżeli którykolwiek argument przekroczy zakres podany w założeniach, program zatrzyma się i poprosi o poprawę.
- Jeżeli liczba argumentów będzie zbyt duża, argumenty dodatkowe zostaną zignorowane.
- Jeżeli użytkownik poda pozycje startowe mrówek wykraczające poza planszę, program zakończy działanie oraz wyświetli informację o błędzie.
- Jeżeli ilość współrzędnych startowych nie będzie zgadzała się z ilością mrówek, mrówki zostaną rozmieszczone losowo.

## 6 Budowa programu

Program podzieliliśmy na 5 części, aby ułatwić czytelność, modyfikacje oraz testy.

- **ant.c, ant.h** - odpowiada za implementację struktur `ant_t` i `ants_t` oraz sposobu poruszania się mrówek

- `struct* ant_t`
- `struct* ants_t`
- `void move(ant_t, int)`
- `void mirror(ant_t, int ,int)`
- `ants_t ants_init(int)`
- `void free_ants(ants_t)`

- **matrix.c, matrix.h** - odpowiada za implementację struktury `mat_t`, odpowiednią alokację pamięci na nią oraz opcjonalną animację mrówki w terminalu
  - `struct* mat_t`
  - `mat_t init(int, int)`
  - `void animate(mat_t)`
  - `free_matrix(mat_t)`
- **exporttopng.c, exporttopng.h** - odpowiada za zapis planszy do pliku png
  - `void write_png_file(char*)`
  - `void process_file(mat_t)`
- **error.c, error.h** - odpowiada za obsługę błędów
  - Coś tu będzie jak zrobimy obsługę
- **main.c** - odpowiada za sterowanie programem. Działa jako reguła **main**.

Do programu dodatkowo załączyliśmy plik z regułą Makefile, mający na celu uproszczenie kompilacji oraz usuwania plików tymczasowych tworzonych przez program.

## 7 Opis poszczególnych modułów

### 7.1 ant.c

### 7.2 matrix.c

### 7.3 exporttopng.c

### 7.4 error.c

### 7.5 main.c

## 8 Przykładowe wywołanie

Zastosowane zostały domyślne wartości

