

# Specyfikacja funkcjonalna - Gra w życie

Krzysztof Dąbrowski i Jakub Bogusz

27 lutego 2019

# Spis treści

<b>1</b>	<b>Cel projektu</b>	<b>2</b>
<b>2</b>	<b>Opis ogólny problemu</b>	<b>3</b>
<b>3</b>	<b>Działanie programu</b>	<b>4</b>
3.1	Komunikacja z użytkownikiem . . . . .	4
3.1.1	Tryb z argumentami z wiersza poleceń . . . . .	4
3.1.2	Tryb interaktywny . . . . .	5
3.2	Format pliku wejściowego . . . . .	5
<b>4</b>	<b>Wyniki działania programu</b>	<b>7</b>

# Rozdział 1

## Cel projektu

Celem projektu jest implementacja gry w życie w języku C. Gotowy program ma przeprowadzać symulację kolejnych pokoleń oraz generować na ich podstawie pliki graficzne przedstawiające etapy symulacji.

## Rozdział 2

# Opis ogólny problemu

Gra w życie jest automatem komórkowym wymyślonym przez brytyjskiego matematyka Johna Horton Conway w 1970 roku. Polega na symulacji kolejnych pokoleń życia komórek według następujących zasad.

**Stany** Komórka może znajdować się w jednym z dwóch stanów

- żywa
- martwa

**Reguły** Następne pokolenie generowane jest zgodnie z regułami:

- Jeżeli komórka była martwa i miała dokładnie 3 żywych sąsiadów, w następnym pokoleniu staje się żywa.
- Jeżeli komórka była żywa to pozostaje żywa jeśli miała dwóch lub trzech żywych sąsiadów. W przeciwnym razie staje się martwa.

## Rozdział 3

# Działanie programu

### 3.1 Komunikacja z użytkownikiem

#### 3.1.1 Tryb z argumentami z wiersza poleceń

##### Argumenty

- -h / -help  
Wyświetlenie pomocy
- -f [nazwa pliku] / --file plik=[nazwa pliku]  
Plik z wejściowym stanem planszy zgodny z [formatem](#).
- -o [ścieżka] / --output\_dest=[ścieżka]  
Ścieżka do folderu, w którym zostaną zapisane wyniki symulacji. Domyślnie brak generacji plików wynikowych i aktywna [flaga -d 1000](#)
- -t (gif — png) / --type (gif — png)  
Typ generowanych rezultatów. Domyślnie gif.
- -n [liczba] / --amount\_of\_generations=[liczba]  
Ilość pokoleń do wygenerowania. Domyślnie 15
- -p [liczba] / --step=[liczba]  
Wybór co który stan symulacji będzie zapisywany. Domyślnie 1
- -s [liczba] / --size=[liczba]  
Losowe generowanie planszy początkowej o podanym rozmiarze. Wyklucza się z -f
- -d [liczba] / --delay=[liczba]  
Podanie tego argumentu spowoduje wyświetlanie w konsoli kolejnych generacji symulacji. Wartość argumentu [liczba] oznacza czas w milisekundach między wyświetleniem poszczególnych pokoleń. Domyślnie 1000.

### 3.1.2 Tryb interaktywny

Program prowadzi dialog z użytkownikiem pozwalając na wybór wszystkich niezbędnych ustawień (odpowiedników flag). Prosi użytkownika o podawanie kolejnych parametrów zgodnie z którymi program będzie działać, na bieżąco kontrolując ich poprawność, nie dopuszczając danych mogących negatywnie wpłynąć na działanie programu.

## 3.2 Format pliku wejściowego

### Przykład

5 3	– rozmiar (x y)
1 0 0 1 1	– Wartości poszczególnych komórek
0 1 1 0 1	– 1 - żywa
0 0 0 1 1	– 0 - martwa

### Opis

Plik wejściowy pozwala na wczytanie planszy bez konieczności wprowadzania przez użytkownika informacji o komórkach jedna po drugiej. Jest to jedyna opcja wczytania planszy w trybie z argumentami z wiersza poleceń.

W trybie interaktywnym użytkownik ma możliwość skorzystania z pliku, jako źródła danych o planszy lub może ręcznie wskazać rozmiar i stany poszczególnych komórek.

### Format pliku

#### Kodowanie

Ponieważ plik powinien zawierać tylko liczby arabskie i odstępy możliwe jest dowolne kodowanie kompatybilne z ASCII.

#### Sugerowane kodowania to:

- ASCII
- UTF-8
- ISO 8859
- Windows-1250

### **Opis formatu**

Plik w pierwszej linii powinien zawierać 2 liczby. Pierwsza z nich oznacza rozmiar planszy w poziomie, druga w pionie.

Następnie plik powinien zawierać tyle linii jaki został podany rozmiar w pionie.

W każdej z tych linii powinno być tyle 0 lub 1 ile wynosi rozmiar w poziomie.

Zero oznacz komórkę martwą, a jeden komórkę żywą.

## Rozdział 4

# Wyniki działania programu

Wyniki działania programu będą zależeć od preferencji użytkownika. W trybie [interaktywnym](#) program zapyta o docelowy format wyniku, a w trybie z argumentami wiersza poleceń, wynik zależeć będzie od wartości [argumentu -t](#) oraz [argumentu -o](#). W obu przypadkach użytkownik będzie mógł:

- wyświetlić wybraną ilość generacji w konsoli,
- wygenerować wybraną ilość plików .png z reprezentacjami graficznymi kolejnych pokoleń
- wygenerować plik .gif przedstawiający życie cywilizacji