

# Specyfikacja Implementacyjna Projektu „Gra W Życie”

Andrzej Czechowski, Bartosz Zakrzewski

Data utworzenia: 05.03.2020

Data ostatniej modyfikacji: 10.03.2020

## 1 Cel Projektu

Celem projektu jest napisanie programu w języku C. Będzie on tworzył kolejne generacje komórek według „Gry w życie” Johna Conwaya. Staramy się utrzymać kod czytelny, wybór opcji mają zapewnić użytkownikowi argumenty wywołania, program będzie podzielony na pliki i funkcje.

## 2 Środowisko pracy z programem

Program w większości zostanie napisany na dwóch komputerach dwóch członków zespołu: Andrzeja Czechowskiego i Bartosza Zakrzewskiego.

Wersja języka C: GNU C Library (Ubuntu GLIBC 2.27-3ubuntu1) stable release version 2.27. Compiled by GNU CC version 7.3.0.



caption 1: Środowisko pracy Andrzej Czechowskiego



caption 2: Środowisko pracy Bartosz Zakrzewskiego

### 3 Zasady pracy z Gitem

Kolejne wersje programu będą umieszczane i zapisywane za pomocą systemu kontroli wersji - git. Kod oraz dokumentacja będzie trafiać na repozytorium o nazwie „Git\_proj”

Zasady tworzenia i działania na gałęziach:

- Na każdej gałęzi dodajemy funkcjonalność, która po przestawieniu trafia na mastera.
- Na gałąź master trafia tylko kod przetestowany i dający się skompilować.
- Prace testowe nad projektem mogą mieć miejsce na gałęzi danego członka zespołu (czechoa1, zakrzewb).

Zasady tworzenia komentarzy przy commitach:

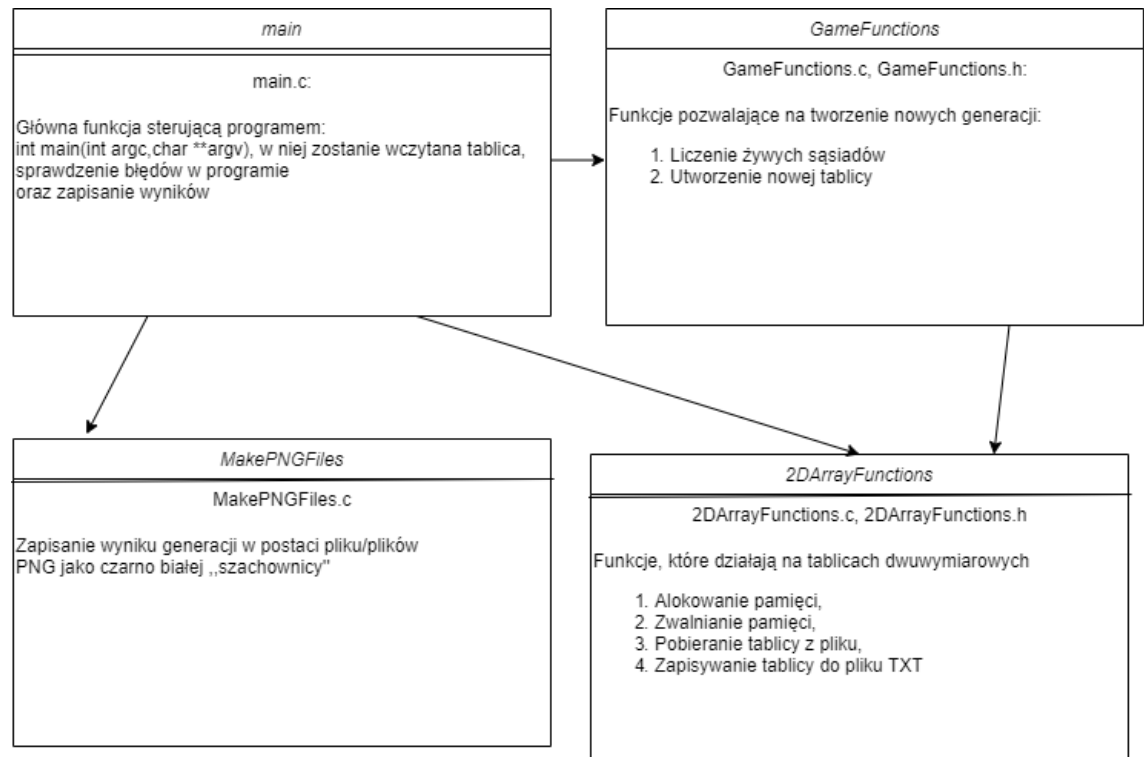
- Komentarz ma opisywać dodaną funkcjonalność, (opcjonalnie) możliwe rzeczy do poprawy oraz błędy.
- Nie będziemy używać numeracji/wersjonowania - opieramy się na dacie wysłania.
- Komentarze piszemy po polsku.

Przykładowy commit: „dodanie argumentów wywołania, nieefektywny algorytm liczenia sąsiadów”

Tagi:

- Jako, że na mastera trafia kod przetestowany, to nie będziemy oznaczać wersji stabilnych.
- Wersja, którą będziemy chcieli oddać jako wersję finalną oznaczymy tagiem FINAL.

## 4 Diagram plików i funkcji



caption 3: Diagram plików

## 5 Szczegółowy opis plików w projekcie

### 5.1 main.c

Przykładowe funkcje:

- `int main(char **argv, int argc)` - funkcja sterująca programem. Jej głównym zadaniem jest wczytanie argumentów wywołania oraz kontrola błędów w programie.

Biblioteki, które mogą być potrzebne:

- `stdio.h`

## 5.2 2DArrayFunctions.c

Przykładowe funkcje:

- `int **allocate2DArray(int r, int c)` - tworzy tablice dwuwymiarową o wymiarach `r` na `c`, która zwraca.
- `void free2DArray(int **arr, int r, int c)` - zwalnia pamięć macierzy (tablicy) `arr` o wymiarach `r` na `c`.
- `void print2DArrayToFile(int **arr, int r, int c, int gen, FILE *out)` - funkcja zapisuje do pliku TXT tablicę dwuwymiarową (`arr` o wymiarach na `r, c`) zgodnie z wzorcem pliku wyjściowego, gdzie `gen` oznacza która jest to generacja.
- `int **download2DArrayFromFile(FILE *in, int *r, int *c, int *gen)` - zapisuje z pliku wejściowego `in`, planszę do macierzy dwuwymiarowej (o wymiarach `r` na `c`) oraz zapisuje rozmiary planszy i wczytaną generację (poprzez podanie adresu `r, c` i `gen` podczas wywołania funkcji np. w `main`ie). Jeśli plik `in` nie miał formatu pliku wejściowego funkcja zwróci `NULL`.

Biblioteki, które mogą być potrzebne:

- `stdlib.h`

## 5.3 GameFunctions.c

Jest to najważniejsza część projektu.

Przykładowe funkcje:

- `int countAliveNeighbours (int **arr, int r, int c, int i, int j)` - zwraca ilość żywych sąsiadów komórki `arr[i][j]`. Argumenty `r, c` określają rozmiar planszy, dzięki czemu wiemy ilu sąsiadów ma komórka `arr[i][j]` - wiemy czy jest w skrajnym położeniu.
- `int *makeNextGeneration2DArray(int **arr, int r, int c)` - tworzy nową generację tablicy `arr`, zapisując ją do nowej macierzy, a następnie ją zwraca.

## 5.4 MakePNGFiles.c

Przykładowe funkcje:

- void writePngFile(char\* fileName);
- void procesFile(void);

Do stworzenia tych funkcji posłużymy się wzorcem umieszczonym na ISO-DZIE ( plik pngexample.c ) Biblioteki, które mogą być potrzebne:

- unistd.h
- stdlib.h
- stdio.h
- string.h
- stdarg.h

## 6 Źródła

- Opis projektu „Automat Komórkowy” B. Chaber
- Stworzenie diagramu plików w programie: <https://www.draw.io>
- Tobiasz Siemiński - opis specyfikacji implementacyjnej <https://sortris.blogspot.com/2010/08/jak-napisac-specyfikacje.html?m=1>
- Ten dokument został utworzony w LaTeX'ie za pomocą strony <https://www.overleaf.com>
- Jest to drugi z kolei dokument dotyczący projektu „Gra w Życie”