

Dokumentacja końcowa projektu "Gra w życie"

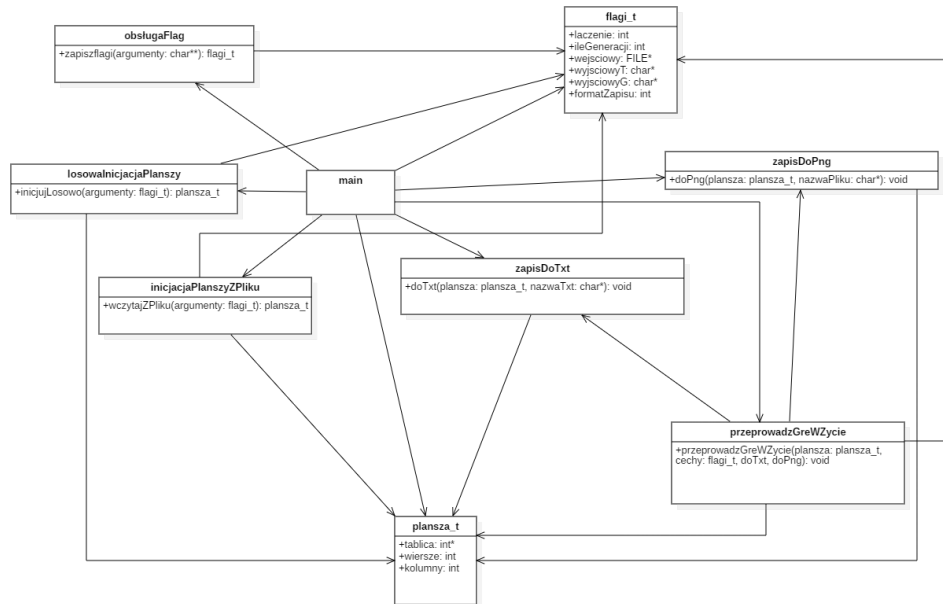
Karolina Czachorska, Katarzyna Stankiewicz

10 kwietnia 2019 r.

Spis treści

1	Ostateczny projekt modułów	2
2	Opis modyfikacji projektu	2
2.1	Moduł przeprowadzGreWZycie	2
2.2	Moduł zapisDoTxt	2
2.3	Moduł obsługaFlag	3
2.4	Struktura plansza t	3
3	Prezentacja działania	3
3.1	Prawidłowe wywołanie programu	3
3.1.1	Wczytywanie planszy z pliku	3
3.1.2	Generowanie losowej planszy	5
3.2	Przykładowe błędne wywołania i reakcja programu na błędy . . .	6
4	Podsumowanie testów modułów	7
4.1	Test modułu zapisDoPng	7
4.2	Test modułu zapisDoTxt	7
4.3	Test modułu przeprowadzGreWZycie	8
4.4	Test modułu inicjacjaPlanszyZPliku	8
4.5	Test modułu losowaInicjacjaPlanszy	9
4.6	Test modułu obsługi flag	9

1 Ostateczny projekt modułów



2 Opis modyfikacji projektu

Podczas pisania programu zostały przeprowadzone pewne modyfikacje zaplanowanych w początkowej fazie modułów. Zmianom uległy m.in. niektóre nazwy plików.

2.1 Moduł przeprowadzGreWZycie

Moduł generator został zastąpiony nazwą przeprowadzGreWZycie, podobnie funkcja generator znajdująca się w nim, ponieważ jest ona bardziej intuicyjna i lepiej oddaje jego działanie, czyli tworzenie wszystkich generacji planszy oraz zapisywanie ich do plików tekstowych lub graficznych. W module tym została dodana funkcja zapisz, która zajmuje się zapisem do .png oraz .txt. Ma to na celu podział całości na mniejsze funkcje, dzięki czemu kod jest czytelniejszy.

2.2 Moduł zapisDoTxt

Nazwa funkcji doPlikuTxt została zastąpiona przez doTxt. Zmieniona została także kolejność argumentów wywołania funkcji doTxt, tak, aby była analogiczna do kolejności argumentów w doPng.

2.3 Moduł obsługaFlag

Istotną zmianą w module dotyczącym flag jest rezygnacja z możliwości wyboru opcji zawijania kolumn i wierszy. Zostały usunięte flagi „łączenie” a dodane „wyświetlanie” umożliwiające wyświetlenie plansz oraz „pomoc” która wyświetla wskazówki dotyczące wywołania programu. Ponadto zmienione zostały nazwy flag „ileGeneracji” na „generacje” oraz „formatZapisu” na „zapisz”. Na koniec również dodano funkcję zmieniającą typ zmiennej formatZapisu.

2.4 Struktura plansza t

Struktura Plansza t została zastąpiona nazwą plansza t, aby ujednolicić schemat nazw modułów, tak aby wszystkie zaczynały się z małej litery. Do plansza t została dodana funkcja wyświetlająca planszę.

3 Prezentacja działania

3.1 Prawidłowe wywołanie programu

3.1.1 Wczytywanie planszy z pliku

Wywołując program używamy krótkiej wersji flag i podajemy następujące parametry:

- plik wejściowy: "test.txt"
- nazwa pliku do zapisu w formacie .txt: "tekstowy"
- nazwa pliku do zapisu w formacie .png: "grafika"
- liczba generacji: 14
- format zapisu: oba formaty (.png i .txt)

W rezultacie otrzymujemy następujące wyniki:

```

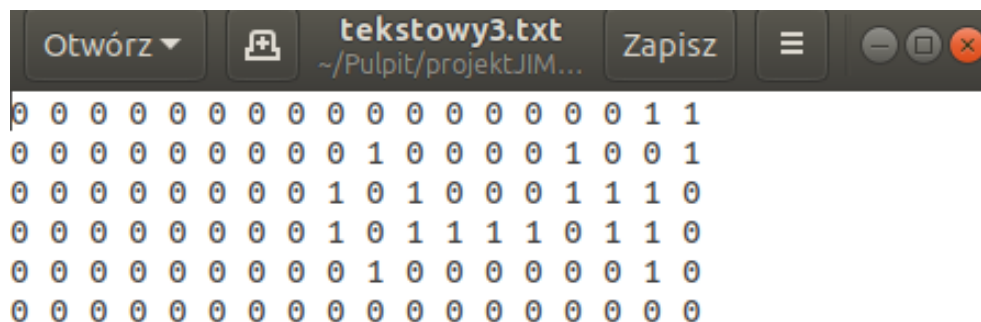
kws@kws-VirtualBox:~/Pulpit/projektJIMP/kod$ ls
flagi_t.h          losowaInicjacjaPlanszy.h  obslugaFlag.o        zapisDoPng.c
gra               losowaInicjacjaPlanszy.o  plansza_t.h          zapisDoPng.h
inicjacjaPlanszyZPliku.c  main.c                   przeprowadzGreWZycie.c  zapisDoPng.o
inicjacjaPlanszyZPliku.h  main.o                   przeprowadzGreWZycie.h  zapisDoTxt.c
inicjacjaPlanszyZPliku.o  Makefile                 przeprowadzGreWZycie.o  zapisDoTxt.h
losowaInicjacjaPlanszy.c  obslugaFlag.c            test.txt              zapisDoTxt.o
kws@kws-VirtualBox:~/Pulpit/projektJIMP/kod$ ./gra -p "test.txt" -t "tekstowy" -g "grafika"
-n 14 -z oba
testkws@kws-VirtualBox:~/Pulpit/projektJIMP/kod$ ls
flagi_t.h          grafika7.png              obslugaFlag.o            tekstowy4.txt
gra               grafika8.png              plansza_t.h              tekstowy5.txt
grafika10.png      grafika9.png              przeprowadzGreWZycie.c   tekstowy6.txt
grafika11.png      inicjacjaPlanszyZPliku.c  przeprowadzGreWZycie.h   tekstowy7.txt
grafika12.png      inicjacjaPlanszyZPliku.h  przeprowadzGreWZycie.o   tekstowy8.txt
grafika13.png      inicjacjaPlanszyZPliku.o  tekstowy10.txt           tekstowy9.txt
grafika14.png      losowaInicjacjaPlanszy.c  tekstowy11.txt           test.txt
grafika1.png       losowaInicjacjaPlanszy.h  tekstowy12.txt           zapisDoPng.c
grafika2.png       losowaInicjacjaPlanszy.o  tekstowy13.txt           zapisDoPng.h
grafika3.png       main.c                    tekstowy14.txt           zapisDoPng.o
grafika4.png       main.o                    tekstowy1.txt            zapisDoTxt.c
grafika5.png       Makefile                  tekstowy2.txt            zapisDoTxt.h
grafika6.png       obslugaFlag.c             tekstowy3.txt            zapisDoTxt.o

```

Rysunek 1: Wywołanie programu z planszą wczytaną z pliku.

Wywołanie programu z powyższymi parametrami spowodowało utworzenie 14 plików o rozszerzeniu ".txt" zawierających kolejne generacje planszy zapisane w formie tekstowej oraz 14 plików o rozszerzeniu ".png" zawierających zapis kolejnych generacji w formie graficznej.

poniżej przedstawiono przykładowe pliki:



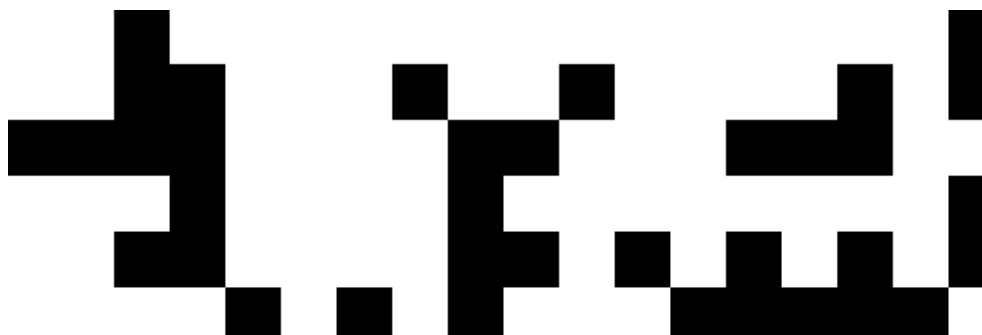
```

tekstowy3.txt
~/Pulpit/projektJIM...
Zapisz

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1
0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 1
0 0 0 0 0 0 0 0 1 0 1 0 0 0 1 1 1 0
0 0 0 0 0 0 0 0 1 0 1 1 1 1 0 1 1 0
0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

```

Rysunek 2: wyniki zapisane w formacie tekstowym



Rysunek 3: wyniki zapisane w formacie graficznym

3.1.2 Generowanie losowej planszy

Wywołując program używamy długiej wersji flag i podajemy następujące parametry:

- rozmiar planszy: 3x3
- liczba generacji: 3
- format zapisu: png
- wyświetlanie: włączone

W rezultacie otrzymujemy następujące wyniki:

```

kws@kws-VirtualBox:~/Pulpit/projektJIMP/kod$ ls
flagi_t.h          losowaInicjacjaPlanszy.h  obslugaFlag.o      zapisDoPng.c
gra               losowaInicjacjaPlanszy.o  plansza_t.h        zapisDoPng.h
inicjacjaPlanszyZPliku.c  main.c                  przeprowadzGreWZycie.c  zapisDoPng.o
inicjacjaPlanszyZPliku.h  main.o                  przeprowadzGreWZycie.h  zapisDoTxt.c
inicjacjaPlanszyZPliku.o  Makefile                przeprowadzGreWZycie.o  zapisDoTxt.h
losowaInicjacjaPlanszy.c  obslugaFlag.c           test.txt            zapisDoTxt.o
kws@kws-VirtualBox:~/Pulpit/projektJIMP/kod$ ./gra --rozmiar 3x3 --generacje 3 --zapisz png
--wyswietl
1 1 0
1 0 1
1 1 1

1 1 0
0 0 1
1 0 1

0 0 0
0 1 1
0 1 1

0 0 0
0 1 1
0 1 1

kws@kws-VirtualBox:~/Pulpit/projektJIMP/kod$ ls
flagi_t.h          inicjacjaPlanszyZPliku.o  obslugaFlag.c      zapisDoPng.c
gen1.png          losowaInicjacjaPlanszy.c  obslugaFlag.o      zapisDoPng.h
gen2.png          losowaInicjacjaPlanszy.h  plansza_t.h        zapisDoPng.o
gen3.png          losowaInicjacjaPlanszy.o  przeprowadzGreWZycie.c  zapisDoTxt.c
gra               main.c                    przeprowadzGreWZycie.h  zapisDoTxt.h
inicjacjaPlanszyZPliku.c  main.o                  przeprowadzGreWZycie.o  zapisDoTxt.o
inicjacjaPlanszyZPliku.h  Makefile                test.txt

```

Rysunek 4: Wywołanie programu z wyświetleniem losowo wygenerowanej planszy.

W przypadku niepodania pliku z którego ma być wczytana początkowa plansza, jest ona generowana losowo o rozmiarze jaki został podany w argumentach wywołania.

3.2 Przykładowe błędne wywołania i reakcja programu na błędy

Wywołując program używamy flagi zarówno z plikiem zawierającym początkową planszę jak i z rozmiarem, który chcielibyśmy uzyskać w przypadku losowej generacji.

```

kws@kws-VirtualBox:~/Pulpit/projektJIMP/kod$ ./gra -p "test.txt" -r 4
6x29 -t "tekstowy" -g "grafika" -n 14 -z oba
Podano juz plik wejsciowy, rozmiar nie zostanie uwzgledniony. Plansza
wygenerowana na podstawie danych zawartych w pliku.

```

Rysunek 5: Wywołanie programu z równoczesnym podaniem nazwy pliku wejściowego i rozmiaru planszy do losowej generacji.

Program prawidłowo wczytuje planszę z pliku, ale ignoruje podany rozmiar,

wyświetlając komunikat o nieuwzględnieniu parametrów podanych przy flagie dotyczącej rozmiaru.

4 Podsumowanie testów modułów

4.1 Test modułu zapisDoPng

Przeprowadzony test zakończył się pomyślnie.
Plansza początkowa:

```
0 0 0 1 0 0 0 1 1 1
1 1 1 1 1 0 0 1 0 0
1 1 0 1 1 1 1 0 1 1
0 1 0 0 0 0 0 1 1 0
0 1 1 1 0 0 0 0 1 0
1 0 1 1 0 0 0 1 0 1
1 1 1 1 1 1 1 0 0 0
0 1 1 1 0 0 1 0 0 0
1 1 0 0 1 0 1 1 0 1
1 1 0 0 0 0 1 1 0 0
```

Plik png:

4.2 Test modułu zapisDoTxt

Przeprowadzony test zakończył się pomyślnie.

Plansza początkowa:

```
0 0 1 0 1 0 0 1 0 0
1 0 0 0 0 1 0 1 1 0
1 0 0 1 1 0 0 1 0 1
0 1 1 0 1 0 0 1 1 0
1 0 0 1 1 1 0 1 0 0
1 1 0 0 1 1 0 1 0 0
0 1 1 1 1 0 1 1 1 0
0 1 0 0 0 1 1 1 0 0
1 0 1 1 0 0 1 0 1 1
0 1 0 0 0 1 0 1 1 0
```

Plansza zapisana do pliku:

```
0 0 1 0 1 0 0 1 0 0
1 0 0 0 0 1 0 1 1 0
1 0 0 1 1 0 0 1 0 1
0 1 1 0 1 0 0 1 1 0
1 0 0 1 1 1 0 1 0 0
```

```
1 1 0 0 1 1 0 1 0 0
0 1 1 1 1 0 1 1 1 0
0 1 0 0 0 1 1 1 0 0
1 0 1 1 0 0 1 0 1 1
0 1 0 0 0 1 0 1 1 0
```

4.3 Test modułu przeprowadzGreWZycie

Przeprowadzony test zakończył się pomyślnie.
Kolejne generacje planszy:

```
1 0 1 1
0 1 0 1
0 0 1 1
0 1 1 1
```

```
0 1 1 1
0 1 0 0
0 0 0 0
0 1 0 1
```

```
0 1 1 0
0 1 1 0
0 1 0 0
0 0 0 0
```

4.4 Test modułu inicjacjaPlanszyZPliku

Przeprowadzony test zakończył się pomyślnie. Program zapisał planszę odczytaną z pliku „test.txt” zawierającego następujące dane:

```
100101010100100101
010101010100100011
000011100001110010
110010101011010101
000101111101111111
110010100000000000
```

Wypisał je w następujący sposób, zliczając ile plik zawiera kolumn i wierszy:

```
kws@kws-VirtualBox:~/Pulpit/projektJIMP/testy/testInicjacjaPliku
$ ./zPliku "test.txt"
Liczba kolumn: 18
Liczba wierszy: 6
1 0 0 1 0 1 0 1 0 1 0 0 1 0 0 1 0 1
0 1 0 1 0 0 1 0 0 1 0 1 0 1 0 1 0 1
1 0 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 0
0 1 0 1 0 1 0 1 0 1 0 0 1 0 0 0 1 1
0 1 0 1 0 0 1 0 0 0 1 1 0 0 0 0 1 1
1 0 0 0 1 1 0 0 0 0 1 1 1 0 0 0 0 1
```

4.5 Test modułu losowaInicjacjaPlanszy

W przypadku niepodania pliku wejściowego plansza zostaje wygenerowana losowo

```
Nie podano pliku. Plansza o wymiarach 10x10 wygenerowana losowo:
1 1 0 1 0 1 0 0 0 0
1 0 0 0 1 1 0 1 1 0
0 0 1 0 1 0 0 0 0 1
0 1 0 0 0 1 0 0 1 0
1 0 1 1 1 0 0 1 1 1
1 1 1 0 1 0 1 1 0 1
0 0 0 1 1 1 0 1 1 1
1 0 0 0 1 1 0 1 0 1
0 1 0 1 0 1 1 1 0 1
0 1 0 0 0 1 1 0 0 1
```

4.6 Test modułu obsługi flag

W przypadku prawidłowego podania wszystkich parametrów program zapisuje je w następujący sposób:

```
kws@kws-VirtualBox:~/Pulpit/projektJIMP/testy/testObsulgaFlag$ ./flagi --plikWejscowy "test.txt" --r
ozmiar 45x53 --wyjscowyT "tekstowy" --wyjscowyG "grafika" --generacje 15 --zapisz oba --wyswietl
-----Zapisane flagi-----
Plik wejscowy: test.txt
Liczba generacji: 15
Liczba wierszy: 53
Liczba kolumn: 45
Format zapisu: oba
Nazwa pliku do zapisu pliku tekstowego: tekstowy/
Nazwa pliku do zapisu grafiki: grafika/
Wyswietlanie: tak
```

W przypadku podania złych argumentów lub nie podania ich wcale, wyświetla wskazówkę dotyczącą wyświetlenia pomocy.

```

kws@kws-VirtualBox:~/Pulpit/projektJIMP/kod$ ./gra -h
Aby podac plik wejscowy do wygenerowania planszy, wpisz:
    --plikWejscowy [nazwa_pliku]
    lub
    -p [nazwa_pliku]
Aby podac rozmiar planszy wygenerowanej losowo, wpisz:
    --rozmiar [liczba_kolumn]x[liczba_wierszy]
    -r [liczba_kolumn]x[liczba_wierszy]
Aby podac nazwe pliku wyjsciowego, gdzie maja zostac wygenerowane plansze, wpisz:
dla zapisu w formacie tekstowym:
    --wyjscowyT [nazwa_pliku]
    lub
    -t [nazwa_pliku]
dla zapisu w formacie graficznym:
    --wyjscowyG [nazwa_pliku]
    lub
    -g [nazwa_pliku]
Aby podac liczbe generacji kolejnych planszy, wpisz:
    --generacje [liczba_generacji]
    lub
    -n [liczba_generacji]
Aby wybrac format zapisu, wpisz:
    --zapisz txt (aby zapisac w formacie tekstowym)
    --zapisz png (aby zapisac w formacie graficznym)
    --zapisz oba (aby zapisac w obu formatach)
    lub
    -z txt (aby zapisac w formacie tekstowym)
    -z png (aby zapisac w formacie graficznym)
    -z oba (aby zapisac w obu formatach)
Aby wyswietlic wygenerowane plansze, wpisz:
    --wyswietl
    lub
    -w

```

5 Sprawdzenie pamięci

Program został przetestowany przez program valgrind pod kątem wycieków pamięci. Program zwraca następujące wyniki:

```

kws@kws-VirtualBox:~/Pulpit/projektJIMP/kod$ valgrind ./gra -r 14x67 -n 2
==7789== Memcheck, a memory error detector
==7789== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==7789== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==7789== Command: ./gra -r 14x67 -n 2
==7789==
==7789==
==7789== HEAP SUMMARY:
==7789==     in use at exit: 0 bytes in 0 blocks
==7789==   total heap usage: 13 allocs, 13 frees, 16,918 bytes allocated
==7789==
==7789== All heap blocks were freed -- no leaks are possible
==7789==
==7789== For counts of detected and suppressed errors, rerun with: -v
==7789== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)

```

Rysunek 6: Wyniki sprawdzenia programu przez valgrind.