

# Specyfikacja funkcjonalna

## Projekt *Gra Galaxy Defender* w języku Java

Jakub Czajka (299239)

Daniel Daczko (299241)

25 kwietnia 2019

### Spis treści

<b>1</b>	<b>Opis ogólny</b>	<b>2</b>
1.1	Nazwa programu . . . . .	2
1.2	Poruszany problem . . . . .	2
1.3	Użytkownik docelowy . . . . .	2
<b>2</b>	<b>Opis funkcjonalności</b>	<b>2</b>
2.1	Jak korzystać z programu? . . . . .	2
2.2	Uruchomienie programu . . . . .	3
2.3	Możliwości programu . . . . .	3
<b>3</b>	<b>Format danych i struktura plików</b>	<b>3</b>
3.1	Pojęcia . . . . .	3
3.2	Struktura katalogów . . . . .	4
3.3	Przechowywanie danych w programie . . . . .	4
3.4	Dane wejściowe . . . . .	4
3.4.1	Plik konfiguracyjny . . . . .	5
3.4.2	Stan gry . . . . .	5
3.5	Dane wyjściowe . . . . .	5
<b>4</b>	<b>Scenariusz działania programu</b>	<b>5</b>
4.1	Scenariusz ogólny . . . . .	5
4.2	Scenariusz szczegółowy . . . . .	6
4.3	Ekrany działania programu . . . . .	8
<b>5</b>	<b>Testowanie</b>	<b>15</b>
5.1	Ogólny przebieg testowania . . . . .	15

## 1 Opis ogólny

### 1.1 Nazwa programu

Jako nazwę programu przyjęliśmy wyrażenie **Galaxy Defender**, które można przetłumaczyć z języka angielskiego jako *Obrońca Galaktyk*. Jest to ściśle nawiązanie do graficznego interfejsu naszej gry.

### 1.2 Poruszany problem

*Galaxy Defender* jest grą typu multiplayer shooter. Wykorzystuje ona grafikę dwuwymiarową. Gra inspirowana jest dwoma klasycznymi tytułami:

- *Space Invaders*, z których czerpie sposób eliminacji wrogów, poruszanie się gracza oraz zbieranie punktów,
- *Tetris*, z których czerpie kształt wrogów oraz ich sposób poruszania się.

Rozgrywka rozpoczyna się od pojawienia się na ekranie dwóch statków przypisanych do graczy. Gracze mogą poruszać się horyzontalnie i ustawiać kąt działka laserowego. Nie mogą oni jednak zamienić się miejscami (tzn. gracz, który zaczął grę po lewej stronie nie może przejść na prawo za gracza, który rozpoczął grę po prawej i na odwrót). Wraz z upływem czasu pojawia się pierwsza fala wrogów. Zadaniem graczy jest zestrzelenie klocków w jak najkrótszym czasie. Za każde zabicie użytkownik otrzymuje punkty. Po przekroczeniu pewnej ilości punktów gra ulega utrudnieniu. Klocki zaczynają poruszać się szybciej i tworzyć bardziej złożone formy.

Jeżeli wróg dotrze do linii statków, zostają odjęte punkty od wyników obydwu graczy. Gra kończy się automatycznie, kiedy jeden z wyników wskaże liczbę ujemną. Dodatkowo gracze w dowolnej chwili mogą zakończyć grę, zapisując jej wynik.

### 1.3 Użytkownik docelowy

Program dedykowany jest graczom w dowolnym wieku. Użytkownikami programu są dwie osoby, których celem jest rywalizacja i dobra zabawa. Każdy gracz chce zdobyć jak największą liczbę punktów i pokonać swojego przeciwnika. Udział w grze sprowadza się do sterowania własnym statkiem kosmicznym przy pomocy klawiatury.

## 2 Opis funkcjonalności

### 2.1 Jak korzystać z programu?

Program należy uruchomić za pomocą kliknięcia na odpowiednią ikonę pliku w formacie JAR.

## 2.2 Uruchomienie programu

Program posiada graficzny interfejs użytkownika. Po uruchomieniu można wybrać jedną z możliwych opcji wyświetlonych na ekranie menu głównego.

## 2.3 Możliwości programu

1. Rozpoczęcie nowej gry.
2. Wczytanie zapisanej gry z pliku tekstowego.
3. Wprowadzenie imion graczy.
4. Sterowanie statkami kosmicznymi.
5. Wyświetlanie na bieżąco zdobytych przez gracza punktów.
6. Wyświetlanie czasu gry.
7. Zatrzymanie gry.
8. Wznowienie gry.
9. Zapisanie aktualnego stanu gry do pliku tekstowego.
10. Obsługa błędów powstałych na skutek wczytania niepoprawnego pliku tekstowego.
11. Przeczytanie instrukcji do gry.

## 3 Format danych i struktura plików

### 3.1 Pojęcia

**Gra** – ustawienia oraz obecny stan rozgrywki.

**Gracz** – użytkownik gry.

**Wróg** – obiekt spadający, za którego zabicie gracz otrzymuje punkty (synonim: klocek).

**Plansza** – obszar gry, w którym wyświetlani są przeciwnicy, gracze oraz toczona jest rozgrywka.

**Statki** – kosmiczne obiekty, którymi gracze poruszają.

**Laser** – promienie wysyłane przez statki, które zbijają wrogów. Nie odbijają się one od krawędzi ekranu oraz znikają po kontakcie z wrogiem.

### 3.2 Struktura katalogów

Drzewo katalogów i plików prezentuje się w następujący sposób:

- doc (*miejsce przechowywania dokumentacji programu*)
  - img (*miejsce przechowywanie grafiki do dokumentacji*)
- src
  - main
    - java/pl/edu/pw/iem/ (*miejsce przechowywanie kodu*)
    - resources
      - fxml (*miejsce przechowywania plików FXML*)
      - styles (*miejsce przechowywania plików CSS*)
  - test
    - java/pl/edu/pw/iem/ (*miejsce przechowywania kodu testów*)
- target
  - classes
    - pl/edu/pw/iem/ (*miejsce przechowywania plików* )
    - fxml (*miejsce przechowywania plików FXML*)
    - styles (*miejsce przechowywania plików CSS*)
  - generated-sources
  - maven-archiver
  - maven-status
  - GalaxyDefender-1.0-SNAPSHOT.jar
  - config.txt (*plik przechowujący konfigurację programu*)
  - pom.xml
  - README.md

### 3.3 Przechowywanie danych w programie

Pliki przechowujemy w projektowym repozytorium kontroli wersji Git na serwerze wydziałowym.

Dane w programie znajdują się w zmiennych o różnych modyfikatorach dostępu.

### 3.4 Dane wejściowe

Program uwzględnia wczytywanie dwóch plików tekstowych. Pliku konfiguracyjnego podczas uruchomienia programu, co odbywa się niejawnie oraz opcjonalnego pliku stanu poprzedniej gry w celu jej kontynuacji.

### 3.4.1 Plik konfiguracyjny

Plik tekstowy przechowujący konfigurację programu postaci:

```
PrędkośćKlocków;WielkośćPoczątkowa;  
PrzyspieszenieKlocków;CoJakiCzasPrzyspieszenie;
```

### 3.4.2 Stan gry

Plik tekstowy przechowujący stan zapisanej gry postaci:

```
NazwaPierwszegoGracza;PunktyPierwszegoGracza;  
NazwaDrugiegoGracza;PunktyDrugiegoGracza;  
CzasGry;PrędkośćKlocków;WielkośćKlocków;  
PrzyspieszenieKlocków;CoJakiCzasPrzyspieszenie;
```

## 3.5 Dane wyjściowe

Program uwzględnia zapisanie do pliku tekstowego przerwanej gry w takiej samej postaci jak plik opisany w podpunkcie 3.4.2.

## 4 Scenariusz działania programu

### 4.1 Scenariusz ogólny

1. Uruchomienie programu.
2. Wybranie opcji:
  - (a) O grze – wyświetlany zostaje ekran z opisem gry.
  - (b) Wczytaj – wczytanie gry z podanego pliku i kontynuacja wczytanej rozgrywki.
    - i. Rozpoczęcie rozgrywki.
    - ii. Możliwość zapisania, spauzowania lub opuszczenia gry.
    - iii. Wyświetlenie zwycięzcy i możliwych opcji.
  - (c) Nowa gra – wczytanie domyślnych ustawień.
    - i. Podanie nazw graczy.
    - ii. Rozpoczęcie rozgrywki.
    - iii. Możliwość zapisania, spauzowania lub opuszczenia gry.
    - iv. Wyświetlenie zwycięzcy i możliwych opcji.
3. Zakończenie działania programu.

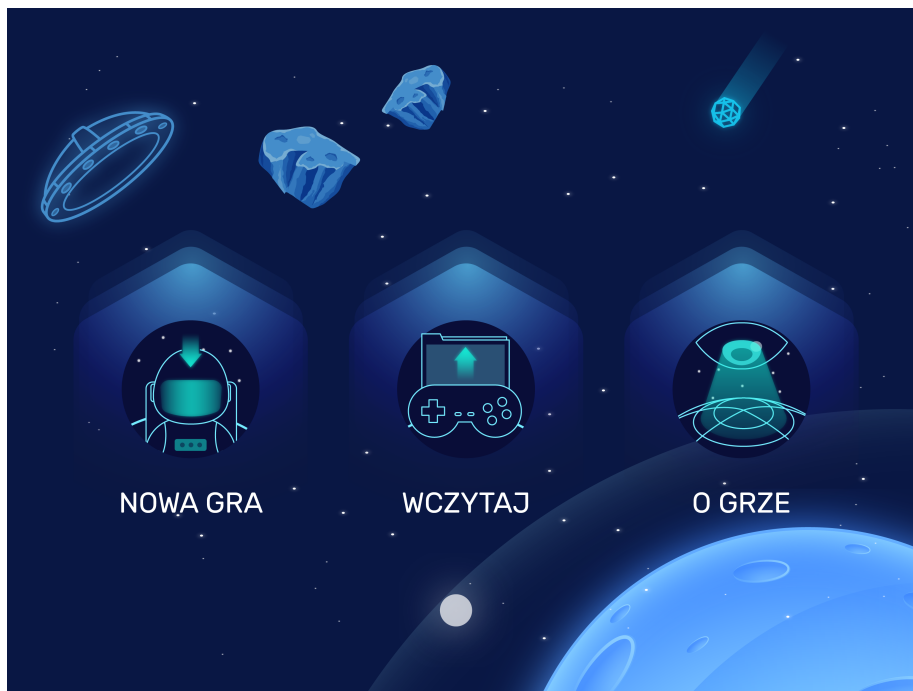
## 4.2 Scenariusz szczegółowy

1. Uruchomienie programu.
2. Wyświetlenie menu głównego.
3. Kliknięcie w jedną z 3 ikon.
4. Sprawdzenie jaka opcja została wybrana:
  - (a) O grze
    - i. Wyświetlenie ekranu z opisem i instrukcją gry (z możliwością powrotu do menu głównego).
  - (b) Wczytaj
    - i. Przejście do ekranu *Wczytaj*.
    - ii. Podanie przez użytkownika ścieżki do pliku.
    - iii. Sprawdzenie istnienia ścieżki i poprawności zawartości pliku (w tym ustawień oraz nazw zawodników). W razie błędu odmowa przejścia krok dalej.
    - iv. Przejście do ekranu gry.
    - v. Rozpoczęcie rozgrywki.
  - (c) Nowa gra
    - i. Wczytanie konfiguracji (domyślnych ustawień) z pliku przy jednoczesnym sprawdzaniu poprawności jego zawartości.
    - ii. Wyświetlenie ekranu *Gracze*.
    - iii. Podanie nazw graczy przez użytkowników.
    - iv. Sprawdzenie poprawności podanych nazw. W razie błędu odmowa dostępu do kolejnego kroku.
    - v. Przejście do ekranu gry.
    - vi. Rozpoczęcie rozgrywki.
5. W trakcie rozgrywki gracze sterują swoimi statkami przy pomocy klawiatury zgodnie z instrukcją zamieszczoną w programie.
6. Możliwe jest otworzenie menu pauzy w trakcie rozgrywki (przy pomocy klawisza *ESC* lub opcji *PAUZA*), kliknięcie w jedną z 3 opcji i sprawdzenie jaka opcja została wybrana:
  - (a) Wznów
    - i. Powrót do rozgrywki.
  - (b) Zapisz (tę opcję można również wywołać bezpośrednio przy pomocy przycisku *ZAPISZ* w trakcie rozgrywki)
    - i. Wyświetlenie ekranu pozwalającego na wskazanie ścieżki do katalogu, w którym ma się znaleźć plik.

- ii. Stworzenie pliku do zapisu gry.
  - iii. Sprawdzenie poprawności utworzenia pliku.
  - iv. Zapis gry do pliku.
  - v. Powrót do menu pauzy.
  - (c) Wyjdź
    - i. Wyświetlenie menu końca gry z opcjami:
      - A. Jeszcze raz – zresetowanie stanu rozgrywki i rozpoczęcie nowej gry.
      - B. Wyjdź – zakończenie pracy programu.
7. Gra może zakończyć się automatycznie, kiedy jeden z wyników graczy wskaże liczbę ujemną. Następuje wyświetlenie ekranu końca gry z opcjami:
- (a) Jeszcze raz – zresetowanie stanu rozgrywki i rozpoczęcie nowej gry.
  - (b) Wyjdź – zakończenie pracy programu.
8. Praca programu może zostać zakończona w każdym momencie jego działania poprzez zamknięcie okna systemowego.

### 4.3 Ekran działania programu

Rysunek 1. prezentuje menu główne naszego programu. Widoczne są na nim trzy ikony pozwalające wybrać jedną z opcji działania.



Rysunek 1: Ekran startowy

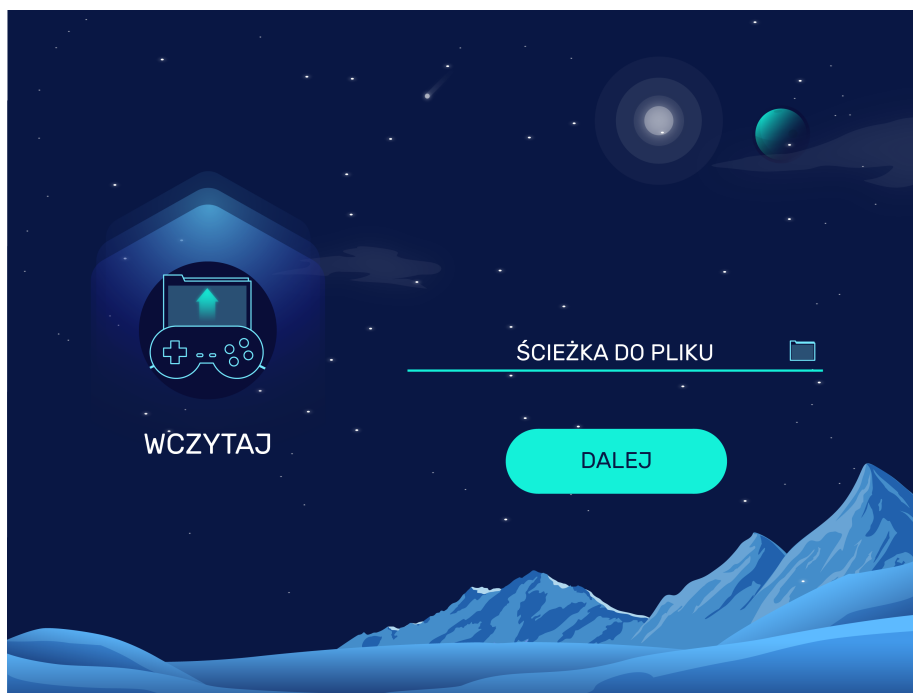


Na rysunku 2. możemy dostrzec ekran pozwalający nam na wprowadzenie nazw graczy i przejście do następnego kroku.



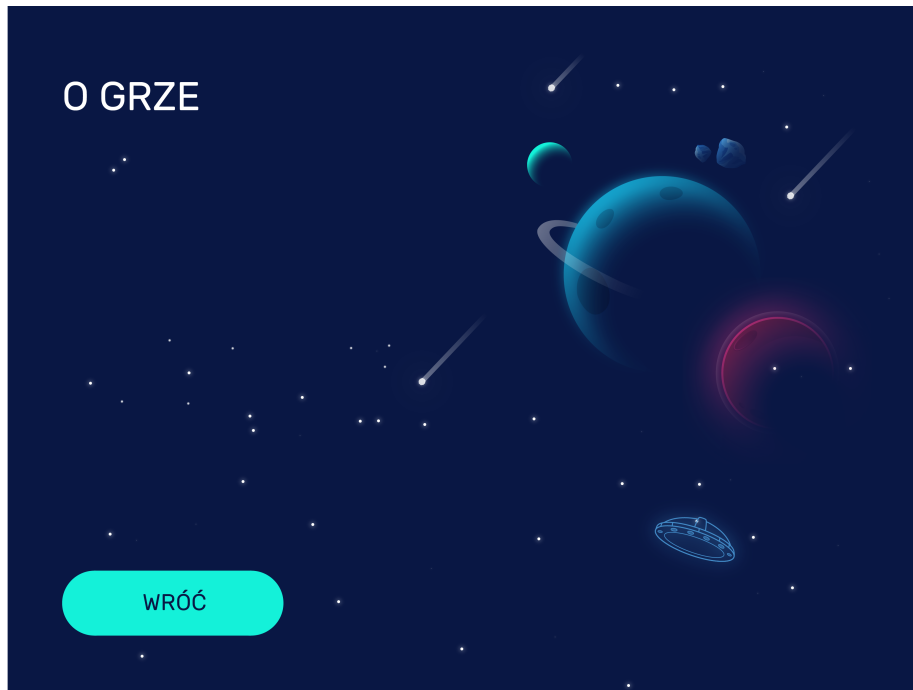
Rysunek 2: Ekran wprowadzania imion graczy

Kolejna grafika (rys. 3.) ukazuje nam ekran, przy pomocy którego możemy ręcznie wprowadzić ścieżkę do pliku w polu tekstowym. Ikona folderu przy końcu linii pozwala użytkownikowi otworzyć okno systemowe z wyborem pliku na dysku, co powinno ułatwić jego odnalezienie. Przycisk *DALEJ* umożliwia przejście do następnego kroku.



Rysunek 3: Ekran wczytywania gry

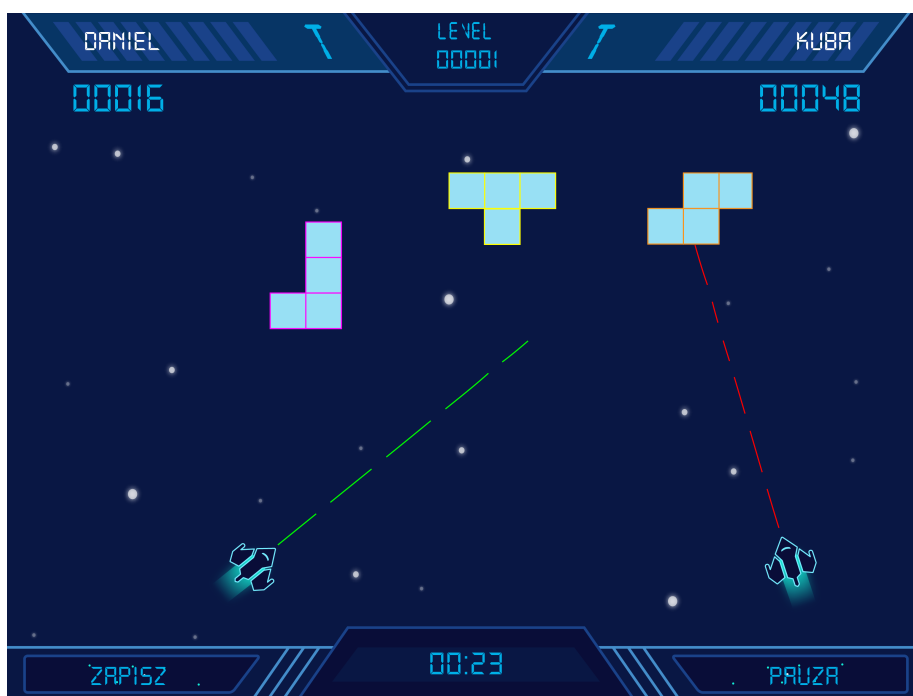
Na ekranie przedstawionym na rysunku 4. zostanie zamieszczona informacja o grze i instrukcja do niej. Istnieje możliwość powrotu do menu głównego przy pomocy przycisku *WRÓĆ*.



Rysunek 4: Ekran o grze

Rysunek 5. prezentuje:

- planszę,
- spadające klocki (wrogów),
- statki,
- lasery,
- zegar czasu gry,
- liczniki punktów,
- licznik poziomów,
- imiona graczy,
- przyciski pozwalające na przerwanie gry i/lub zapisanie jej do pliku.



Rysunek 5: Ekran gry

Na rysunku 6. możemy zaobserwować przyciski *WZNÓW*, *ZAPISZ*, *WYJDŹ*, których działanie dokładnie opisane jest w podrozdziale 4.2 pkt 6.



Rysunek 6: Ekran zatrzymania gry

Rysunki 7. i 8. reprezentują ekrany końca gry.

Rysunek 7. zostanie wyświetlony, gdy w momencie ukończenia gry jeden z graczy ma większą liczbę punktów niż drugi. Liczba ta musi być dodatnia.

Rysunek 8. zostanie wyświetlony, gdy w momencie ukończenia gry obaj gracze będą mieli ujemną liczbę punktów lub dojdzie do remisu.



Rysunek 7: Ekran wygranej



Rysunek 8: Ekran przegranej

*Pojedyncze elementy grafiki wektorowej i fonty użyte do przygotowania poglądowych rysunków pochodzą z otwartych źródeł. Licencje pozwalają na ich bezpłatne wykorzystanie (bez obowiązku podania autora) zarówno w projektach prywatnych jak i komercyjnych. Wszystkie powstałe do projektu wizualizacje są naszą autorską inwencją twórczą.*

## 5 Testowanie

### 5.1 Ogólny przebieg testowania

Do przetestowania kodu użyjemy narzędzi z biblioteki AssertJ. Natomiast GUI przetestujemy ręcznie podczas tworzenia aplikacji. Dodatkowo gotowy program przetestujemy sami, grając w niego oraz prześlemy go do testów znajomym.