

Sprawozdanie końcowe

War Of Tanks

Aleksandra Michalska, Natalia Olszewska

08.06.2021

Spis treści

1	Informacje ogólne	2
1.1	O dokumencie	2
1.2	O programie	2
1.3	Środowisko implementacyjne i funkcjonalne	2
2	Zmiany w funkcjonowaniu programu	2
2.1	Uruchomienie programu	2
2.2	Domyślne wartości paramentów programu	2
2.3	Ograniczenia parametrów programu	2
2.4	Plik konfiguracyjny	3
2.5	Panel gry	3
2.5.1	Ruch czołgu	3
2.6	Interface gry	4
2.6.1	Początkowy ekran	4
2.6.2	Okno załączania pliku konfiguracyjnego	5
2.6.3	Ekran gry	5
2.6.4	Okno zasad gry	6
3	Zmiany w implementacji	6
3.1	Diagram klas	6
3.2	Rezygnacja z elementów	6
3.3	Nowe rozwiązania	6
3.3.1	Tank, Body, Barrel	6
3.3.2	Colony	7
3.3.3	SettingImages	7
3.3.4	Wątki	7
3.3.5	ConfFrame	7
3.4	Zmiany	7
3.4.1	StartFrame	7
3.4.2	HelpFrame	8
3.4.3	GameFrame	8
3.4.4	TankPanel, CellsPanel	8
4	Przeprowadzone testy	8
5	Uwagi końcowe	9
5.1	Osiągnięcie celu projektu	9
5.2	Elementy niedoskonałe i możliwe błędy	9
5.3	Ograniczenia programu	9
6	Podsumowanie współpracy	9

1 Informacje ogólne

1.1 O dokumencie

Dokument jest podsumowaniem pracy nad projektem gry WarOfTanks. Korresponduje z dokumentami *"Specyfikacja funkcjonalna"* oraz *"Specyfikacja implementacyjna"*.

1.2 O programie

Program jest przeznaczony dla dwóch użytkowników korzystających z jednego urządzenia. Ich zadaniem jest strzelanie do spadających komórek przy pomocy poruszających się czołgów. Wygrywa ten użytkownik, który uzyska większą liczbę punktów na koniec gry.

1.3 Środowisko implementacyjne i funkcjonalne

Program został zaimplementowany w języku programowania **Java 1.8** w systemie operacyjnym **Windows 10**.

W celu kontroli wersji został użyty **Projektor EE** (system kontroli wersji Politechniki Warszawskiej).

W potrzebie uzyskania więcej informacji implementacyjnych patrz *"Specyfikacja implementacyjna"* punkt 1.

2 Zmiany w funkcjonowaniu programu

2.1 Uruchomienie programu

Uruchomienie programu może odbywać się na dwa sposoby:

1. przy pomocy odpowiednich programów (np. IntelliJ IDEA, NetBeans)
2. przy pomocy konsoli systemowej. W tym sposobie należy przejść do katalogu `/out/production/Tanks` i z tego poziomu użyć polecenia:

```
java frames.MainOfWar
```

2.2 Domyślne wartości paramentów programu

W celu poprawy funkcjonowania programu, zostały ustalone nowe domyślne wartości parametrów programu (patrz *Tabela 1*).

2.3 Ograniczenia parametrów programu

W związku ze zmianą domyślnych wartości parametrów programu, zostały zmienione ich ograniczenia (patrz *Tabela 2*).

parametr	wartość	parametr	wartość
v1	30	x1	7
dv1	5	p1	3
v2	60	r1	20
dv2	10	dr1	5
t1	30	h1	70
t2	10	dh1	5
t3	3		

Tabela 1: Domyślne wartości parametrów programu

parametr	ograniczenie	parametr	ograniczenie
v1	$10 \leq v1 \leq 50$	x1	$0 \leq x1 \leq 25$
dv1	$0 \leq dv1 \leq 25$	p1	$1 \leq p1 \leq 9$
v2	$5 \leq v2 \leq 35$	r1	$5 \leq r1 \leq 25$
dv2	$0 \leq dv2 \leq 20$	dr1	$2 \leq dr1 \leq 10$
t1	$30 \leq t1 \leq 180$	h1	$20 \leq h1 \leq 70$
t2	$5 \leq t2 \leq 40$	dh1	$1 \leq dh1 \leq 10$
t3	$3 \leq t3 \leq 15$		

Tabela 2: Ograniczenia parametrów programu

2.4 Plik konfiguracyjny

Przekazanie pliku konfiguracyjnego do programu odbywa się przez menu **Add File**.

W przypadku podania (w pliku konfiguracyjnym) nieistniejących parametrów programu lub nieprawidłowych wartości parametrów programu, zostają one pominięte i użytkownik nie jest informowany o żadnych błędach. Do programu zostają przekazane tylko te wartości parametrów, które udało się poprawnie odczytać, a pozostałe zostają przypisane jak opisano w sekcji *"Domyślne wartości parametrów programu"*.

Przykładowo, podanie następujących danych w pliku konfiguracyjnym:

```
-nieistniejący 8 -samParametr -v1 złaWartość -h1 60
```

skutkuje zmianą tylko początkowej wysokości komórek na 60 pikseli.

2.5 Panel gry

2.5.1 Ruch czołgu

Z powodu kwestii implementacyjnych został zmieniony sposób ruchu czołgu dla gracza po prawej stronie (niebieski czołg):

- ◇ ruch w górę: klawisz ↑
- ◇ ruch w dół: klawisz ↓
- ◇ obrót lufy w górę: klawisz →
- ◇ obrót lufy w dół: klawisz ←

2.6 Interface gry

2.6.1 Początkowy ekran

Ostateczny ekran początkowy zawiera 4 przyciski:

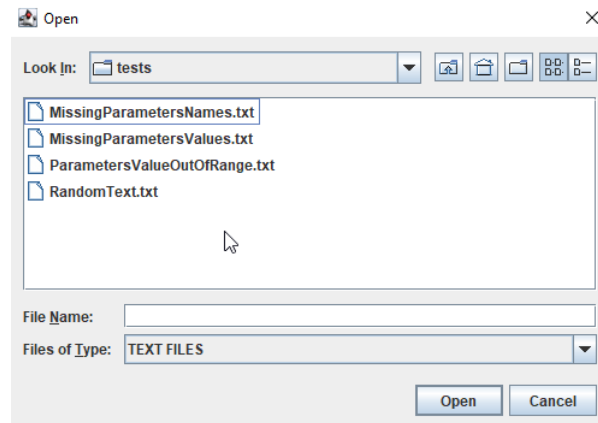
- **START** rozpoczynający grę
- **HELP** wyświetlający zasady gry
- **ADD FILE** wyświetlający okno do załączenia pliku konfiguracyjnego
- **RESET** resetujący dane wprowadzone za pomocą pliku konfiguracyjnego



Rysunek 1: Początkowy ekran

2.6.2 Okno załączania pliku konfiguracyjnego

Po naciśnięciu przycisku **ADD FILE** wyświetla się okno, w którym użytkownik może wybrać plik z danymi.



Rysunek 2: Okno załączania pliku konfiguracyjnego

2.6.3 Ekran gry

Rozmieszczenie paneli na ekranie gry pozostało niezmienione. Ostateczny wygląd ekranu gry przedstawiony jest na Rysunku 3.



Rysunek 3: Ekran gry

2.6.4 Okno zasad gry

Po naciśnięciu przycisku **HELP** wyświetla się okno, w którym użytkownik może przeczytać zasady gry.



Rysunek 4: Okno zasad gry

3 Zmiany w implementacji

3.1 Diagram klas

Ze względu na zmiany implementacyjne w klasach został zmieniony diagram klas. Z przyczyn wizualnych jest on dostępny pod danym [adresem](#).

3.2 Rezygnacja z elementów

Z powodu niewystarczającej ilości czasu komponent *SettingsFrame*, wraz z podłączeniem muzyki do projektu, nie został zaimplementowany.

3.3 Nowe rozwiązania

3.3.1 Tank, Body, Barrel

Zamiast utworzenia dwóch osobnych klas dla prawego i lewego czołgu, została utworzona jedna. Wykorzystuje ona dwie klasy, ponownie zaimplementowane w

pojedyncze, *Body* i *Barrel*. Przy tworzeniu nowego obiektu typu **Tank** należy podać literę 'r' lub 'l' odpowiadającą stronie czołgu do utworzenia.

Pozwoliło to zapobiec powtarzaniu się prawie tego samego kodu w celu utworzenia dwóch obiektów.

3.3.2 Colony

Zastępco do tworzenia jednej klasy abstrakcyjnej i z niej czterech osobnych klas kolonii, została utworzona klasa generująca wszelkie możliwe rozstawienia koloni w siatce 3 na 3.

3.3.3 SettingImages

Z powodu dużej liczby grafik użytych do projektu została utworzona klasa *SettingImages*, która pobiera grafiki z odpowiednich katalogów i zwraca ich wartości w odpowiednich metodach.

3.3.4 Wątki

Program został podzielony na kilka niezależnych wątków odpowiadających pewnym elementom. Zaliczają się do nich:

- ◇ poruszanie czołgiem lewym,
- ◇ poruszanie czołgiem prawym,
- ◇ strzelanie czołgów,
- ◇ zmiana parametrów.

3.3.5 ConfFrame

Do celów załączenia pliku konfiguracyjnego została utworzona klasa *ConfFrame*. Odbiera ona parametry i przetwarza ich zawartość oraz poprawność, modyfikując przy tym odpowiednio obiekt **ConfObject** towarzyszący rozgrywce.

3.4 Zmiany

3.4.1 StartFrame

Zamiast jednej klasy *StartFrame* jej zawartość została rozdzielona na kilka:

- ◇ *StartPanelTitle* - wyświetla tytuł gry w panelu startowym,
- ◇ *StartPanelBackground* - wyświetla tło gry w panelu startowym.
- ◇ *StartPanelButtons* - wyświetla panel przycisków i odpowiada za ich funkcjonowanie,
- ◇ *StartFrame* - odpowiada za wyświetlenie ramki panelu startowego.

3.4.2 HelpFrame

Do klasy *HelpFrame* zostały dodane dwie pomocnicze klasy:

- ◇ *HelpPanelTitle* - wyświetla tytuł w panelu pomocy,
- ◇ *GoBackButton* - wyświetla przycisk powrotu do panelu startowego.

3.4.3 GameFrame

Do klasy *GameFrame* została dopisana pomocnicza *GameTimer*. Odpowiada ona za wykonanie zrzutu ekranu po zakończeniu gry oraz pokazaniu zwycięzcy. Wyświetla również czas danej rozgrywki.

3.4.4 TankPanel, CellsPanel

Z klasy *GameFrame* została wyodrębniona główna struktura rozgrywki. Część zależna od graczy została zaimplementowana w klasie *TankPanel*. To właśnie tam znajdują się wątki odpowiadające poruszaniu i strzelaniu czołgów. Rozgrywka od drugiej strony została umieszczona w *CellsPanel*. Właśnie tam generują się losowo komórki i kolonie.

4 Przeprowadzone testy

Testy zostały podzielone na kilka procesów.

Pierwszymi były testy poprzez uruchomienia. Sprawdzanie na pierwszy i drugi rzut oka, czy program wykonuje się tak jak powinien. Dotyczyły one również likwidacji ostrzeżeń.

Następnym procesem było dostosowanie projektu do uruchomienia w różnych środowiskach. Testy zostały przeprowadzone w kompilatorze **IntelliJ Idea Project**, terminalu środowiska Windows CMD oraz w terminalu środowiska Linuxa.

Przed podejściem do testów ostatecznych program został przetestowany względem minimalnych i maksymalnych obciążeń oraz wartości parameterów wejściowych.

Ostatnim krokiem było przeprowadzenie testów jednostkowych JUnit. W tej części uwaga została poświęcona badaniu poprawności pojedynczych metod. Zaliczały się do nich m.in:

- ◇ *decreaseBombPoints* - z klasy **Bomb**,
- ◇ *setBulletCoordinate* - z klasy **Bullet**,
- ◇ *changeCellSide* - z klasy **CellsPanel**,
- ◇ *increaseCellPoints* - z klasy **Cell**,
- ◇ *decreaseCellPoints* - z klasy **Cell**,
- ◇ *getIsAlone* - z klasy **Cell**,
- ◇ *getWhichColony* - z klasy **Cell**,
- ◇ *changeSizeOfCellsInColony* - z klasy **Colony**,

- ◇ *setParametersTest* - z klasy **ConfFrame**,
- ◇ *changeTankParameters* - z klasy **TankPanel**.

5 Uwagi końcowe

5.1 Osiągnięcie celu projektu

Projekt został zakończony pomyślnie oraz wszelkie podstawowe elementy zostały dodane. Można zatem uznać, że cel projektu został osiągnięty.

5.2 Elementy niedoskonałe i możliwe błędy

Do elementów, które zdecydowanie nie zadowalają, należy brak implementacji muzyki w grze, wbrew początkowym zamiarom. Spowodowane to było zbyt małą ilością czasu, jednak jest to element, jaki można poprawić.

Innym przykładem niedoskonałości na progu błędu może być wykonywany zrzut ekranu po zakończeniu gry. Jeśli gracz, w momencie zakończenia gry, będą wyświetlać inny panel lub okno, na grafice zapisze się fragment właśnie tego obrazu. Pomimo problematyki tego aspektu, można go jak najbardziej poprawić w przyszłości.

5.3 Ograniczenia programu

Program posiada własne ograniczenia. Jedne ze względów technicznych, takich jak potrzebna pamięć do uruchomienia gry. Inne poprzez ograniczenia wartości parametrów wejściowych. Te drugie z kolei pełnią ważną rolę w poprawnym funkcjonowaniu gry. W momencie, gdy gracz podałiby dowolne parametry nieograniczone, program mógłby zwracać nielogiczne wartości lub nawet zakończyć się błędem.

6 Podsumowanie współpracy

Współpraca przebiegła bez komplikacji. Obie strony były równie zaangażowane w pracę nad projektem.