

---

PROJEKT WYKONAŁ: **MATEUSZ MOGIELNICKI**

GRUPA PS: **9**

INDEX: **110879**

---

---

## DOKUMENTACJA – BATTLESHIPS

---

---

### OPIS PROJEKTU

---

Głównym celem projektu jest stworzenie interaktywnej wersji gry "Battleships", znanej również jako "Statki". Gra będzie dostępna w formie aplikacji konsolowej, korzystającej z interfejsu tekstowego. Realizacja projektu odbyła się w języku Java, z wykorzystaniem biblioteki tekstowej Laterna. Wprowadzenie gry do świata interaktywnego pozwoli na przekształcenie klasycznej rozgrywki planszowej w emocjonujące doświadczenie wirtualne.

Główne założenia

- **Cel projektu:** Przeniesienie klasycznej gry planszowej "Battleships" do formy interaktywnej w aplikacji konsolowej z użyciem interfejsu tekstowego i biblioteki Laterna.
- **Istota aplikacji:** Umożliwienie graczom planowania taktyki, rozmieszczania statków oraz skutecznego strzelania w celu zniszczenia floty przeciwnika, przy jednoczesnym wykorzystaniu prostego i intuicyjnego interfejsu tekstowego.

---

### OPIS FUNKCJONALNOŚCI

---

1. Gra przeciwko Komputerowi:
  - Gracze mają opcję rozpoczęcia rozgrywki przeciwko komputerowi, zaprogramowanego poprzez programistę.
  - Gracze będą mieli możliwość interaktywnego rozmieszczania swoich statków na planszy przed rozpoczęciem rozgrywki. Za pomocą prostego interfejsu tekstowego będą wybierać pozycje i orientacje statków poprzez klikanie strzałek.
  - Komputer będzie prowadził swoją flotę, a gracz będzie musiał zastosować swoje umiejętności taktyczne, aby go pokonać.
  - Gra umożliwi na zmianę wykonywanie ruchów, strzelając w konkretne pola na planszy przeciwnika. Po każdym strzale, gra poinformuje gracza, czy trafienie było udane, czy też pole jest puste.
  - Gracz, który jako pierwszy zatopi wszystkie statki przeciwnika, zostanie uznany za zwycięzcę. Gra wyświetli stosowny komunikat o wyniku, a gracze będą mieli możliwość rozpoczęcia nowej rozgrywki.
  - Na koniec gry zostaną przydzielone punkty w rankingu(tylko w przypadku gdy gracz wygra), gdzie można potem zobaczyć swoje wyniki w odpowiedniej zakładce RANKING w menu.

- ## 2. Symulacja gry między komputerami:

- Gra umożliwia również symulację rozgrywki między dwoma komputerami. Algorytm zaimplementowany w programie rywalizuje z drugim takim samym algorytmem, a użytkownik może obserwować przebieg symulowanej bitwy.

### 3. Zasady Gry

- Dostęp do zasad gry w trakcie rozgrywki, umożliwiające graczom łatwe zrozumienie reguł oraz specyfiki rozgrywki w dowolnym momencie.

#### 4. Zakupy w Sklepie

- Gracze będą mieli możliwość zdobycia specjalnych statków lub barier poprzez zakupy w wirtualnym sklepie. To dodatkowe elementy taktyczne, które mogą wpłynąć na przebieg rozgrywki.

## 5. Ranking

- System rankingowy pozwala graczom śledzić swoje postępy i porównywać swoje umiejętności z innymi graczami. Wyświetlenie rankingu dostarcza dodatkowej motywacji do poprawy swoich wyników.

## INTERESUJĄCE ZAGADNIENIA PROJEKTOWE

## 1. Wykorzystanie biblioteki Laterna

- Skomplikowana integracja biblioteki Laterna do obsługi interfejsu tekstowego, pozwalająca na interaktywność w grze, a jednocześnie zapewniająca czytelność i estetykę.
- Została ona wykorzystana do wyświetlania wszystkich widoków, ale dzięki jej funkcjom można sprawnie wyświetlać np. główny tytuł gry oraz zmieniać kolory.

```
public static void printTitle() {
    TextGraphics tg = screen.newTextGraphics();
    tg.setForegroundColor(TextColor.ANSI.RED_BRIGHT);
    tg.putString(1, 3, 1, s: "                                ", SGR.UNDERLINE);
    tg.setForegroundColor(TextColor.ANSI.YELLOW_BRIGHT);
    tg.putString(1, 6, 1, 2, s: " ---- - ----- - ----- - ----- - -----", SGR.BOLD);
    tg.putString(1, 6, 3, s: " | _ _ / \_ \_ _ _ | | | _ _ _ / _ _ | | | _ _ \_ \_ _", SGR.BOLD);
    tg.putString(1, 6, 4, s: " | _ \_ / _ \_ | | | | | | _ _ \_ \_ \_ | | | | | | \_ \_ \_ \n", SGR.BOLD);
    tg.putString(1, 6, 5, s: " | | _ / _ _ \_ | | | | | _ _ _ _ | _ | | | _ _ / _ _ | \n", SGR.BOLD);
    tg.putString(1, 6, 6, s: " | _ _ _ / _ \_ \_ | | | | _ _ _ _ | _ _ / | | | _ _ | | _ _ _ \n", SGR.BOLD);
    tg.setForegroundColor(TextColor.ANSI.RED_BRIGHT);
    tg.putString(1, 3, 8, s: "                                ", SGR.CROSSED_OUT);
}
```

## 2. Obsługa zdarzeń interaktywnych

- Zaimplementowanie obsługi interakcji z graczem poprzez interfejs tekstowy, umożliwiając jednocześnie płynną rozgrywkę i intuicyjne działanie.
- Do tego zostało użyte nasłuchiwanie na klawisze poprzez interfejs Laterny, gdzie po menu można się poruszać poprzez kliknięcie strzałki.
- Oto jedna z funkcji klasy UserInput.java gdzie można przechwytywać klawisze (w tym przypadku funkcja do wyświetlania zadanego pytania oraz pobrania odpowiedzi od użytkownika):

```
public static boolean question(Screen screen, Terminal terminal, String message) throws IOException, InterruptedException {
    char userResponse = '\0';
    TextGraphics tg = screen.newTextGraphics();
    KeyStroke keyStroke;

    do {
        tg.setForegroundColor(TextColor.ANSI.WHITE_BRIGHT);
        keyStroke = terminal.pollInput();

        if (keyStroke != null)
            if (keyStroke.getKeyType() == KeyType.Character) {...} else if (keyStroke.getKeyType() == KeyType.Escape) {
                try {...} catch (GameException e) {
                    throw new RuntimeException(e);
                }
            }

        tg.putString(10, 11, message, SGR.BOLD);
        tg.putString(16, 13, s: "Odpowiedź: ", SGR.BOLD);
        tg.setForegroundColor(TextColor.ANSI.GREEN_BRIGHT);
        tg.putString(17, 13, String.valueOf((userResponse == '\0' ? "" : userResponse)), SGR.BOLD);
        if (userResponse == '\0')
            tg.putString(17, 13, s: "| ", SGR.BOLD, SGR.BLINK);
        else
            tg.putString(18, 13, s: "| ", SGR.BOLD, SGR.BLINK);
        screen.refresh();
        Thread.sleep(10);
    } while (userResponse == '\0');
    return userResponse == 'y';
}
```

## 3. Struktura danych planszy

- Optymalne zorganizowanie struktur danych planszy gry, tak aby umożliwić efektywne rozmieszczanie statków, szybkie sprawdzanie rezultatów strzałów i obsługę dodatkowych elementów taktycznych
- Są to funkcje z klasy UserInput.java: readPositionToShot() oraz getMovedShipPosition(), są one bardzo zbliżone do funkcji question, lecz bardziej rozbudowana, ponieważ należy aktualizować pozycję kursora oraz rozmieszczanego statku.

4. Logika strzałów komputera w symulacji oraz komputer vs gracz:

- Komputer na początku tak jak gracz losowo strzela w planszę, ale gdy trafi w statek to zapisuje jego położenie.
- Następnie strzela do pól otaczających poprzedni trafiony strzał, do momentu gdy zatopi cały statek, a potem powtarza cały algorytm od początku szukając innych statków.

```
public Position ComputerShoot(BattleField defenderBattleField) throws GameException {
    if (shoots.isEmpty()) return Position.randPosition();
    else {
        nextShoots.addAll(defenderBattleField.getAdjacentValidPositions(getLastShoot()));

        if (nextShoots.isEmpty())
            return Position.randPosition();

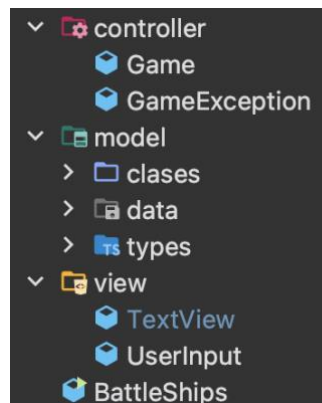
        Position nextPos = nextShoots.get(0);
        nextShoots.remove(index: 0);

        return nextPos;
    }
}
```

5. Użycie podziału na MVC(model, view, controller)

- Dzięki czemu został zachowany odpowiedni podział na klasy.
- Struktura projektu stała się czytelniejsza oraz nie ma problemu w znalezieniu szukanych klas.

6. Dzięki skrupulatnemu przemyśleniu całej logiki projektu, aby dodać widok graficzny należy tylko stworzyć odpowiednią klasę obsługującą graficzne widoki.



## INSTRUKCJE

---

### INSTRUKCJA INSTALACJI

---

1. Pobierz pliki źródłowe gry z CEZ2.
2. Uruchom dany projekt w swoim edytorze kodu.
3. Wejdź w plik BattleShips.java.
4. Uruchom plik.
5. Postępuj zgodnie z instrukcjami na ekranie(wybierz tryb wyświetlania w konsoli, działa jak na razie tylko tekstowe wyświetlanie).

### INSTRUKCJA KONFIGURACJI

---

Dodatkowe kroki konfiguracyjne nie są wymagane. Gra jest gotowa do użycia po instalacji.

### INSTRUKCJA UŻYTKOWNIKA

---

1. Uruchom aplikację.
2. Postępuj zgodnie z instrukcjami na ekranie.
3. Wprowadź swój NICK.
4. Poruszaj się po menu za pomocą strzałek.
5. W razie problemów wejdź w zakładkę zasady gry w menu.
6. Jeśli chcesz zagrać kliknij w zakładkę Rozpocznij Grę i rozmieść swoje statki na planszy.
7. Następnie gra się rozpocznie.
8. Nie daj się wyeliminować Wrogowi!

## WNIOSKI

---

Projekt nie tylko dostarcza wspaniałej rozrywki, ale również stanowi fascynujące wyzwanie programistyczne. Implementacja interaktywnego interfejsu tekstowego przy użyciu biblioteki Laterna pozwala na przeniesienie klasycznej rozgrywki "Battleships" do nowej, atrakcyjnej formy, dodając przy tym elementy taktyczne i symulacji gry komputer vs komputer.

## SAMOOCENA

---

Projekt oceniam jako udany. Udało się skutecznie zaimplementować wszystkie funkcjonalności postawione na początku projektu, a napisanie gry było bardzo pozytywnym wyzwaniem. Istnieje potencjał do rozszerzenia gry o dodatkowe funkcje w przyszłości oraz możliwość zmiany interfejsu na graficzny. Liczę na pozytywne rozpatrzenie projektu oraz Ocenę: 5.