

Politechnika Śląska
Wydział Automatyki, Elektroniki i Informatyki

Programowanie Komputerów II

Statki

autor Bartosz Siwiaszczyk

rok akademicki 2023/2024

kierunek Teleinformatyka

semestr 3

sekcja 1

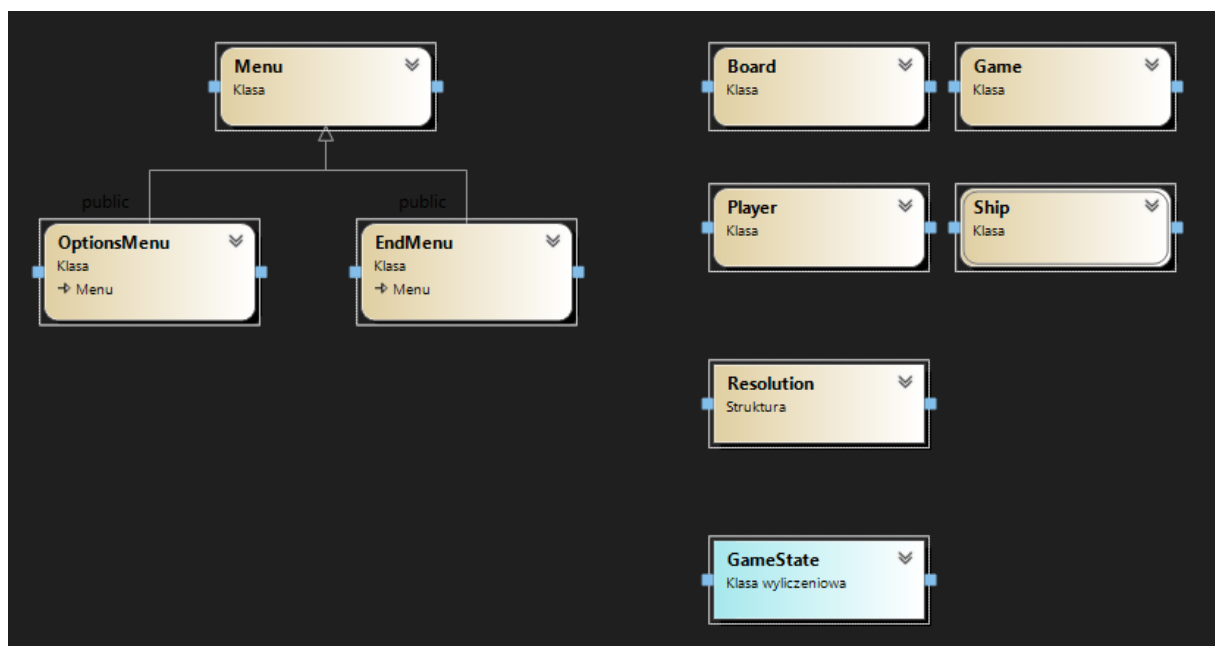
1. Treść zadania

Zadanie polegało na stworzeniu gry w statki w języku C++ z wykorzystaniem paradygmatu programowania obiektowego. Gra miała być odporna na błędy wprowadzanych danych, musiała obsługiwać niewielkie niezgodności formatu danych wejściowych, a także być poprawna pod względem czasu wykonania i zajętości pamięci.

2. Analiza zadania

2.1 Struktury danych

W moim projekcie użyłem różnych struktur danych do reprezentacji planszy, statków, strzałów itp. Na przykład, wykorzystałem tablice dwuwymiarowe do reprezentacji plansz oraz dynamiczne struktury danych do przechowywania informacji o statkach.



2.2 Algorytmy

Do implementacji gry zastosowałem algorytmy sprawdzające trafienia, obsługujące umieszczanie statków na planszy oraz algorytmy do generowania losowych pozycji statków.

3. Specyfikacja zewnętrzna

Specyfikacja zewnętrzna mojej gry obejmuje instrukcję dla użytkownika, opisującą, jak korzystać z programu. Zawiera informacje o sposobie rozpoczęcia gry, wprowadzania strzałów, zasadach zwycięstwa i przegranej.

4. Specyfikacja wewnętrzna

4.1 Typy zdefiniowane w programie

W programie zdefiniowałem różne typy reprezentujące planszę, statki, strzały, graczy itp. Każdy typ miał przemyślane zastosowanie zgodnie z zasadami obiektowości.

Board	Klasa reprezentująca planszę do gry w statki
EndMenu	Klasa reprezentująca menu końcowe gry
Game	Klasa reprezentująca główny obiekt gry
Menu	
OptionsMenu	Klasa reprezentująca menu opcji w grze
Player	
Resolution	Struktura reprezentująca rozdzielczość, składającą się z szerokości i wysokości	..
Ship	Klasa reprezentująca statek

4.2 Ogólna struktura programu

Program składa się z kilku klas, takich jak Game, Player, Board, które są logicznie powiązane. Zastosowałem dziedziczenie, aby wprowadzić polimorfizm w niektórych miejscach.

```
~/9c491b86-gr01-repo/Projekt/Statki/board.cpp .....
~/9c491b86-gr01-repo/Projekt/Statki/board.h .....
~/9c491b86-gr01-repo/Projekt/Statki/config.h .....
~/9c491b86-gr01-repo/Projekt/Statki/endMenu.cpp ..
~/9c491b86-gr01-repo/Projekt/Statki/endMenu.h .....
~/9c491b86-gr01-repo/Projekt/Statki/game.cpp .....
~/9c491b86-gr01-repo/Projekt/Statki/game.h .....
~/9c491b86-gr01-repo/Projekt/Statki/main.cpp .....
~/9c491b86-gr01-repo/Projekt/Statki/menu.cpp .....
~/9c491b86-gr01-repo/Projekt/Statki/menu.h .....
~/9c491b86-gr01-repo/Projekt/Statki/optionsMenu.cpp
~/9c491b86-gr01-repo/Projekt/Statki/optionsMenu.h ..
~/9c491b86-gr01-repo/Projekt/Statki/player.cpp .....
~/9c491b86-gr01-repo/Projekt/Statki/player.h .....
~/9c491b86-gr01-repo/Projekt/Statki/ship.cpp .....
~/9c491b86-gr01-repo/Projekt/Statki/ship.h .....
```

4.3 Szczegółowy opis implementacji funkcji

W implementacji funkcji zwróciłem szczególną uwagę na prawidłowy podział na klasy oraz zastosowanie paradygmatu programowania obiektowego. Komentarze w kodzie opisujące działanie poszczególnych funkcji zostały umieszczone w plikach źródłowych.

5. Testowanie

Przeprowadziłem testy jednostkowe, sprawdzające poprawność działania poszczególnych funkcji oraz testy integracyjne, weryfikujące współpracę między klasami.

6. Podsumowanie

Projekt został zrealizowany zgodnie z założeniami przedstawionymi w wymaganiach. Zastosowane struktury danych i algorytmy umożliwiają poprawne działanie gry, a zastosowane paradygmaty programowania obiektowego spełniają wymagania dotyczące struktury kodu.