

BATTLESHIP

DOKUMENTACJA PROJEKTU

Daniel Jambor

Wydział Matematyki Stosowanej: Informatyka, semestr I, grupa 2C

Charakterystyka programu

Program jest konsolową interpretacją popularnej gry w statki. Dwóch graczy po wpisaniu swoich imion rozpoczyna rozmieszczanie swoich okrętów na polu o wymiarach 9x9. Każdy gracz ma w swoim arsenale pięć statków: jednoelementowy, dwuelementowy, trzelementowy, czteroelementowy i pięcioelementowy. Statki są ustawiane pionowo i poziomo, oraz nie mogą sąsiadować ze sobą. Po rozstawieniu statków, gracze zaczynają wybierać pola „do odstrzału”. Wygrywa gracz, który jako pierwszy zniszczy wszystkie okręty przeciwnika.

Instrukcja obsługi

Grę można uruchomić wyłącznie w systemie Windows. Oprócz podstawowych programów oferowanych przy instalacji systemu, możliwe, że będzie wymagane zainstalowanie „Visual C++ Redistributable”.

Wybory w menu głównym gry można podejmować poprzez naciśnięcie klawisza „1” bądź „0” na klawiaturze numerycznej. Po wybraniu opcji „START”, gracze będą kolejno proszeni o podanie swoich imion. Imiona nie mogą być dłuższe niż 15 znaków. Po wybraniu imion, kolejno „Gracz 1” i „Gracz 2” będą proszeni o rozmieszczenie swoich okrętów. Aby rozmieścić okręt, należy wpisać jego współrzędne (np. A1). Gra rozpoznaje duże i małe litery, w każdym innym przypadku poinformuje gracza o błędnie wprowadzonych danych. Po wpisaniu współrzędnych gra poprosi nas o wciśnięcie „H”, jeśli chcemy ustawić nasz statek poziomo, lub klawisza „V”, jeśli chcemy ustawić okręt pionowo. Kiedy „Gracz 1” ustawi swoje okręty, „Gracz 2” nie powinien w tym czasie patrzeć na ekran gry.

Turę zawsze rozpoczyna „Gracz 1”. Na ekranie zostaną wyświetlone dwie plansze – górna, która należy do przeciwnika, oraz dolna – należąca do gracza wykonującego swój ruch. Wybieranie celów następuje w ten sam sposób, co ustawianie okrętów. Jeśli gracz trafi okręt, zostanie wyświetlony stosowny komunikat o trafieniu (bądź zatopieniu). W takim przypadku gracz nie traci swojej tury i może oddać ponowny strzał. Przed każdą turą zostanie wyświetlony komunikat informujący czyja tura będzie następna. Gra w takim momencie nie wyświetla pola bitwy, co uniemożliwia graczom oszukiwanie oraz daje czas na zmianę gracza przy urządzeniu. Po ukończeniu gry, zostaje wyświetlony komunikat o zwycięstwie danego gracza, po czym możliwe jest wyjście do menu głównego gry.

Legenda pola bitwy:

- „O” – okręt
- „X” – strzał, który trafił w okręt
- „+” – strzał, który spudłował

Komunikaty, które mogą pojawić się w grze:

- < GRACZ X > Proszę podać imię (max 15 znaków)
- < KOMUNIKAT > Naciśnij dowolny klawisz...
- < KOMUNIKAT > Twoja nazwa jest za długa!
- < WYBÓR > Podaj koordynaty statku (np. A1)
- < KOMUNIKAT > Błędne Dane! Spróbuj jeszcze raz!
- < WYBÓR > Naciśnij 'H'/'V', jeśli chcesz ustawić statek poziomo/pionowo
- < KOMUNIKAT > Nie możesz tutaj postawić statku! Nachodzi on na inny!
- < KOMUNIKAT > Nie możesz tutaj postawić statku! Wychodzi poza planszę!
- < KOMUNIKAT > Nie możesz tutaj postawić statku! Sąsiaduje on z innym
- < KOMUNIKAT > Brawo! Trafiłeś (oraz zatopiłeś) statek!
- < KOMUNIKAT > Już tutaj strzelałeś!

Specyfikacja techniczna

Rozwiązanie „*Battleship*” jest podzielone na 3 pliki *.cpp* oraz 2 pliki *.h*:

- *Battleship.cpp* – następuje tam główna inicjalizacja funkcji odpowiedzialnej za uruchomienie menu głównego, ukrycia kursora konsoli oraz rozmiarów okna.
- *Functions.h* / *Functions.cpp* – w plikach tych znajdują się wszystkie funkcje wykorzystywane w programie
- *Structs.h* / *Structs.cpp* – w plikach tych znajdują się wszystkie struktury wykorzystywane w programie oraz ich metody.

Program został napisany w języku C++ w środowisku Visual Studio. Zostały wykorzystane następujące biblioteki:

- *<iostream>*
- *<conio.h>*
- *<string>*
- *<windows.h>*

Szczegóły techniczne

Funkcje, które zostały napisane na potrzeby programu:

- *mainMenu()* – wyświetla główne menu gry i wszystkie inne funkcje potrzebne do wyboru opcji
- *mainMusicStart()* – odpowiada za odtwarzanie głównego motywu muzycznego w grze
- *mainMusicStop()* – odpowiada za wyłączenie głównego motywu muzycznego w grze

- *soundEffect()* – odpowiada za odtworzenie danego dźwięku w danej sytuacji (np. plusk wody przy spudłowaniu)
- *windowSize()* – odpowiada za odpowiednie rozmiary okna konsoli
- *hidePointer()* – chowa kursor konsoli
- *removeScrollbar()* – ukrywa suwak konsoli
- *getGUIOfMenu()* – wyświetlanie główne napisy menu głównego
- *setChoiceInMainMenu()* – odpowiada za podejmowanie decyzji w menu głównym
- *startGame()* – funkcja rozpoczynająca grę
- *startBattle()* – funkcja rozpoczynająca etap strzelania
- *isShipNearby()* – sprawdza, czy statki sąsiadują ze sobą
- *playerTrun()* – funkcja odpowiadająca za wszystkie akcje, jakie gracz może podjąć podczas swojej tury
- *setCoords()* – funkcja ustawiająca koordynaty planowanego ataku oraz rozmieszczenia statków
- *areCoordsCorrect()* – funkcja sprawdzająca poprawność wprowadzanych danych

Struktury znajdujące się w programie:

- *Ship* – struktura zawierająca wszystkie potrzebne dane na temat statku (koordynaty, długość statku)
- *Field* – struktura pola, zawierająca informacje o polu (graficzna reprezentacja, kolor reprezentacji graficznej, informację czy pole jest puste, czy znajduje się na nim statek oraz czy został na nie oddany strzał)
- *Battlefield* – główna struktura programu, zawierająca 5 obiektów struktury *Ship*, tablicę struktur *Field* rozmiarów 10x10, informację na temat punktów życia gracza, jego imię oraz numer

Metody struktury *Battlefield*:

- *setPlayerName()* – metoda odpowiedzialna za ustawienie imienia gracza
- *setDefaultFields()* – tworzy „czyste” pole bitwy, tzn. bez żadnych statków, wszystkie pola z tą samą reprezentacją graficzną, kolorem etc.
- *setShips()* – odpowiedzialna za ustawianie statków gracza na polu bitwy
- *updateFields()* – aktualizuje pole bitwy po rozstawieniu każdego statku oraz po każdym oddanym strzale
- *updateEnemyFields()* – aktualizuje pole bitwy przeciwnika z punktu widzenia danego gracza
- *getBattlefield()* – metoda odpowiedzialna za wyświetlanie pola bitwy
- *endGame()* – ekran końcowy gry, pozwalający również na powrót do menu głównego