AKADEMIA GÓRNICZO-HUTNICZA

Dokumentacja do projektu

BattleShip game

z przedmiotu

Języki programowania obiektowego

Elektronika 1 rok

Jakub Konior

czwartek 14:40

prowadzący: Rafał Frączek

28.05.2023

1. Opis projektu

Gra w statki w ramach tego projektu oferuje interaktywną planszę bitwy, na której gracze mogą rozmieszczać swoje statki i grać między sobą. Możliwa jest rozgrywka w 2 osoby. Środowiskiem gry jest konsola.

2. Project description

The project's Battleship game features an interactive battle board where players can place their ships and play against each other. It is possible to play with 2 people. The game environment is the console.

3. Instrukcja użytkownika

Gracze na przemiennie strzelają w planszę przeciwnika z nadzieją trafienia w statek co przybliży ich do wygranej. Wygrywa gracz który pierwszy pozbędzie się statków przeciwnika. Każde trafienie daje graczowi kolejną szansę. W pierszej fazie gracze ustawiają swoje statki na planszy od największego do najmniejszych. Ustawianie statków odbywa się poprzez wpisanie wyrażenia dla początku statku (litera , liczba , położenie). Litery na planszy powinny być małe i w zakresie od a – j. Liczby przyjmowane są od 1 – 10, a położenie to informacja czy chcemy ustawić statek w pionie czy w poziomie.

4. Kompilacja

Wystarczy standardowa kompilacja. Program działa tylko w systemie Windows.

5. Pliki źródłowe

Projekt składa się z następujących plików źródłowych:

- *BattleShip.h*, *BattleShip.cpp* – główny plik projektu odpowiedzialny za rozgrywkę,
- *Board.h*, *Board.cpp* – klasa odpowiedzialna za operacje na planszy gry,
- *BoardShips.h*, *BoardShips.cpp* – klasa odpowiedzialna za operacje na panszy logiki,
- *Player.h, Player.cpp – klasa odpowiedzialna za przechowywanie informacji o graczu,*
- *Ship.h, Ship.cpp – klasa odpowiedzialna za przechowywanie informacji o statkach.*

6. Zależności

W projekcie wykorzystano następujące dodatkowe biblioteki:

- Windows API – interfejs programistyczny systemu Microsoft Windows. Strona internetowa: https://learn.microsoft.com/en-us/windows/win32/apiindex/windows-api-list

7. Opis klas

W projekcie utworzono następujące klasy:

1. Klasa Board reprezentuje planszę gry:

- Konstruktor Board(int id, BoardShips boardShip, vector<vector<char>> board = {}, int dim = 10) - tworzy obiekt klasy Board.
- int getId() const - zwraca identyfikator planszy.
- int getDim() const - zwraca rozmiar planszy.
- BoardShips getBoardShip() - zwraca obiekt klasy BoardShips reprezentujący rozmieszczenie statków na planszy.

- vector<vector<char>> getBoard() - zwraca aktualną konfigurację planszy.
- void setId(int id) - ustawia identyfikator planszy na podaną wartość id.
- void setBoardShip(BoardShips boardShip) - ustawia rozmieszczenie statków na planszy na podany obiekt klasy BoardShips.
- void setBoard(vector<vector<char>> board) - ustawia konfigurację planszy na podaną tablicę dwuwymiarową board.
- vector<vector<char>> createBoard() - tworzy początkową konfigurację planszy, wypełniając ją znakami '#'.
- void changeFieldIcon(int positionX, int positionY) - zmienia ikonę pola na planszy na podstawie obecności statku w danym miejscu. Jeśli statek jest na polu (positionX, positionY), ikona zostaje zmieniona na 'X', w przeciwnym razie na 'o'.
- void printBoard() - wyświetla aktualną konfigurację planszy na ekranie w czytelnej formie.

2. Klasa Player - reprezentuje gracza:
    - Konstruktor Player(string name, int boardId, bool turn = false) - tworzy obiekt klasy Player.
    - int getBoardId() - zwraca identyfikator planszy gracza.
    - string getName() - zwraca nazwę gracza,
    - bool getTurn() - zwraca informację, czy jest tura gracza (true) czy nie (false).
    - void setBoardId(int boardId) - ustawia identyfikator planszy gracza na podaną wartość boardId.
    - void setName(string name) - ustawia nazwę gracza na podaną wartość name,
    - void setTurn(bool turn) - ustawia turę gracza na podaną wartość
3. Klasa BoardShips reprezentuje rozmieszczenie statków na planszy logiki:
    - Konstruktor BoardShips(int id, int dim = 10) - tworzy obiekt klasy BoardShips.
    - int getId() - zwraca identyfikator planszy.
    - int getDim() - zwraca rozmiar planszy.
    - vector<vector<int>> getBoardShips() - zwraca tablicę dwuwymiarową reprezentującą rozmieszczenie statków na planszy.
    - void setId(int id) - ustawia identyfikator planszy na podaną wartość id.
    - vector<vector<int>> createBoardShips() - tworzy początkowe rozmieszczenie statków na planszy, wypełniając ją zerami.
    - void addShip(Ship ship) - dodaje statek do planszy, aktualizując odpowiednie pola w tablicy reprezentującej rozmieszczenie statków.
    - int getFieldState(int positionX, int positionY) - zwraca stan pola na planszy o podanych współrzędnych (positionX, positionY).
    - bool isShipOnField(int positionX, int positionY) - sprawdza, czy na danym polu planszy znajduje się statek.
    - void changeFieldState(int positionX, int positionY) - zmienia stan pola na planszy o podanych współrzędnych (positionX, positionY), oznaczając trafienie statku lub pudło.

- bool isEnd() - sprawdza, czy na planszy nie ma już żadnych statków.
- void showBoardShips() - wyświetla aktualne rozmieszczenie statków na planszy w czytelnej formie.

4. Klasa Ship reprezentuje statek:
    - Konstruktor Ship(int id, int positionX, int positionY, int sizeX, int sizeY) - tworzy obiekt klasy Ship.
    - int getId() - zwraca identyfikator statku.
    - int getPositionX() - zwraca współrzędną X początkowej pozycji statku.
    - int getPositionY() - zwraca współrzędną Y początkowej pozycji statku.
    - int getSizeX() - zwraca rozmiar statku w kierunku X.
    - int getSizeY() - zwraca rozmiar statku w kierunku Y.
    - void setId(int id) - ustawia identyfikator statku na podaną wartość id.
    - void setPositionX(int positionX) - ustawia współrzędną X początkowej pozycji statku na podaną wartość positionX.
    - void setPositionY(int positionY) - ustawia współrzędną Y początkowej pozycji statku na podaną wartość positionY.
    - void setSizeX(int sizeX) - ustawia rozmiar statku w kierunku X na podaną wartość sizeX.
    - void setSizeY(int sizeY) - ustawia rozmiar statku w kierunku Y na podaną wartość sizeY.

8. Zasoby

- brak

9. Dalszy rozwój i ulepszenia

Dodanie interfejsu graficznego, żeby nie musieć wprowadzać pozycji z klawiatury,

Dodanie ochrony przed postawieniem statków obok siebie,

Dodanie ochrony przed postawieniem statków na sobie,

Dodanie możliwości rozgrywki z komputerem,

Dodanie historii rozgrywek.

10. Inne

brak