

Politechnika Śląska
Wydział Matematyk Stosowanej
Kierunek Informatyka

Gliwice, 09.02.2021

Programowanie I
projekt zaliczeniowy

"Konsolowa gra w *Statki*"

Jakub Pietrasik gr. lab. 6

1. Opis projektu.

Aplikacja konsolowa umożliwiająca grę w statki dla dwóch osób lub dla jednej osoby przeciwko komputerowi.

2. Wymagania

- Menu główne z możliwością wyboru trybu gry, zapoznania się z instrukcją, wyłączenia muzyki lub zakończenia programu
- Możliwość ustawienia pseudonimów graczy
- Możliwość manualnego bądź pseudolosowego ustawienia masztowców przez gracza
- W przypadku gry przeciwko komputerowi dostępne są dwa poziomy trudności oraz możliwość podejrzenia planszy komputera przed rozpoczęciem gry
- Po każdej grze program wyświetla statystyki rozgrywki i pozwala na powrót do menu oraz rozpoczęcie nowej rozgrywki
- Podczas działania programu w tle gra muzyka, którą można wyłączyć w menu głównym
- Po oddaniu celnego strzału program odgrywa odpowiedni dźwięk (podobnie dla strzału nietrafionego)
- Kolorowe menu oraz plansze do gry

3. Przebieg realizacji

Program zacząłem od planu i rozpisania w punktach kolejnych jego funkcjonalności. Najpierw stworzyłem losowo generowane plansze do gry oparte o tablice dwuwymiarowe oraz możliwość manualnego rozstawienia statków na planszy. Następnie powstał główny szkielet: menu, uproszony tryb dla dwóch graczy, uproszczone AI. Po wykonaniu wstępnych testów zacząłem rozbudowywać kod, który namnażając się, stawał się coraz mniej przejrzysty, sprawiając trudności w znajdowaniu ewentualnych błędów. Zacząłem stosować komentarze do danych partii kodu, nad którymi aktualnie pracowałem. Zamieniałem początkowo wpisywane w dwóch miejscach koordynaty liczbowe na literowo-liczbowe, zapisywane w postaci c-stringa, a następnie za pomocą stworzonej specjalnie w tym celu funkcji transformowałem je na zrozumiałe dla programu współrzędne, nadające się do wpisania do tablicy dwuwymiarowej. Jak się często okazywało kod, który początkowo działał, okazywał się wadliwy w różnych szczególnych przypadkach. Wiele błędów wychodziło na jaw dopiero po wpisaniu odpowiednich sekwencji komend (np. Po strzale na współrzędne z linii 10 i zbitiu masztowca program wykrywał każdy następny strzał jako trafiony (problemem okazał się jeden błędnie postawiony znak). Sporo problemów sprawiło też stworzenie funkcji odpowiadającej za ruchy komputera, które do złudzenia miały przypominać te wykonywane przez człowieka. Przez nieodpowiednio napisaną bazę programu, wykonanie tej funkcji było dużym wyzwaniem, musiałem pójść na kilka kompromisów, ale ostatecznie funkcja działa dobrze. Najwięcej czasu zajęły mi zdecydowanie ostateczne testy i ewentualne poprawki w finalnym programie.

Program napisałem strukturalnie, jego działanie opiera się na funkcjach, których wywołanie w odpowiedniej kolejności umożliwia poprawne działanie programu (przykładowo: funkcja rysujP odpowiada za wyświetlanie na ekranie danej w argumencie planszy).

Opis plików:

- Źródło.cpp - plik zawierający program
- Instukcja.txt - plik zawierający instrukcję gry w statki
- Pliki dźwiękowe - program korzysta z nich, aby odtwarzać muzykę i dźwięki

Użyte biblioteki:

1. <fstream> - zawiera funkcje umożliwiające wczytywanie i odczytywanie danych z pliku
2. <cstdlib> - zawiera funkcję system("cls") umożliwiając "czyszczenie" konsoli
3. <locale.h> - dzięki funkcji zawartej w tej bibliotece program obsługuje polskie znaki
4. <SFML/Audio.hpp> - biblioteka umożliwiająca odtwarzanie plików audio
5. <string> - zawiera funkcję transform() umożliwiającą m.in. zamianę małych znaków na duże
6. <random> - zawiera funkcje służące do generowania liczb pseudolosowych
7. <conio.h> - zawiera funkcję getch(), która czeka na naciśnięcie przez użytkownika dowolnego znaku na klawiaturze
8. <windows.h> - zawiera funkcje potrzebne do użycia kolorów w konsoli
9. <cmath> - biblioteka matematyczna
10. <thread> - zawiera funkcję sleep_for umożliwiającą chwilowe "uśpienie" programu
11. <iostream> - mało znana biblioteka obsługująca operacje wejścia i wyjścia
12. <limits> - zawiera funkcje umożliwiające czyszczenie bufora wejścia

4. Instrukcja użytkownika

Program działa w oparciu o wpisywanie odpowiednich komend na klawiaturze oraz zatwierdzenie ich klawiszem "Enter". Instrukcja samej gry w statki jest możliwa do otwarcia z poziomu menu głównego.

5. Podsumowanie i wnioski.

Realizacja projektu nie przebiegła bezproblemowo. Program okazał się być znacznie bardziej rozbudowany i wymagający lepszego rozplanowania niż wyjściowo zakładałem. Wiele rozwiązań, które zastosowałem na początku, nie sprawdzało się przy późniejszych próbach modyfikacji lub usprawnienia działania programu, co skutkowało wieloma godzinami spędzonymi na próbie implementacji nowych partii kodu. Niestety po zapisaniu 1000 linijki na rozpoczęcie pisania kodu od nowa było już za późno. Finalnie udało mi się rozwiązać niemal wszystkie problemy napotkane po drodze, realizując wszystkie założenia projektu. Był to mój pierwszy większy projekt i koniec końców jestem zadowolony z ostatecznego rezultatu, nauczył mnie on cierpliwości, skrupulatności i niekonwencjonalnego podejścia do pisania programów komputerowych, jednak nie zamierzam nigdy więcej do niego wracać. W przyszłych projektach skupię się na dokładniejszym zaplanowaniu pracy, dobraniu odpowiednich struktur danych już na samym początku i zadbanie o komentarze oraz przejrzystość w każdej partii kodu.