

Politechnika Śląska
Wydział Informatyki,
Elektroniki i Informatyki

Programowanie Komputerów

temat projektu
Sokoban

| | |
|-----------------------------|-------------------------|
| autor | Sławomir Put |
| prowadzący | Mgr inż. Maciej Długosz |
| rok akademicki | 2021/2022 |
| kierunek | Informatyka |
| rodzaj studiów | SSI |
| semestr | 2 |
| termin laboratorium | środa, 15:30 – 17:00 |
| sekcja | 13 |
| termin oddania sprawozdania | 2022-07-03 |

1 Treść zadania

Proszę napisać grę typu Sokoban. Jej protagonista jest magazynierem, którego zadaniem jest przesuwanie przedmiotów w taki sposób, aby znalazły się w swoich miejscach docelowych w jak najmniejszej liczbie ruchów oraz bez zakleszczenia się przedmiotu. Przesuwanie odbywa się po magazynie, który składa się z różnych pól (np. podłoga, ściana, miejsce docelowe), które różnie się zachowują, gdy zostanie na nie wepchnięty przedmiot. Program powinien zachowywać historię wyników, przy czym historia powinna być przechowywana niezależnie dla użytkowników, którzy podali swoją nazwę, oraz pragnących zachować anonimowość, użytkowników identyfikowanych przez wartość liczbową.

2 Analiza zadania

W ramach projektu wymagane jest napisać grę komputerową typu Sokoban. Z wszelką obsługą poruszania postacią, kolizji oraz interakcji z obiektami.

2.1 Struktury danych

W programie wykorzystano rodzaj danych typu Enum, zawiera ona listę wartości, którym można nadać zmiennej własnego typu enum. Powstała do przechowywania oraz przekazywania wartości dotyczących grup obiektów. Reszta danych jest w zmiennych bez struktur.

2.2 Algorytmy

W ramach projektu zostały zaimplementowane takie techniki jak:

- ☐ klasy wraz z dziedziczeniem,
- ☐ polimorfizm,
- ☐ szablony (wzorce),
- ☐ co najmniej jeden wzorzec projektowy.

Dodatkowo została wykorzystana technika:

- ☐ przeciążania operatorów

3 Specyfikacja zewnętrzna

Program jest uruchamiany przy pomocy biblioteki zewnętrznej SDL2 wraz z jej rozszerzeniami SDL_ttf oraz SDL_image. Biblioteki te pozwalają na rozszerzenie programu o sferę wizualną rozumianą poprzez wyświetlane grafiki oraz tekstu.

By poprawnie uruchomić program wymagane jest debugowanie aplikacji w wersji x86 oraz potrzebne jest dołączenie wymienionych wyżej bibliotek. By ułatwić ten proces biblioteki te są zawarte w folderach programu (..\Projekt\Sokoban\Libraries).

Instrukcja dołączania bibliotek we właściwościach programu.

1. Właściwości konfiguracji -> katalogi VC++ -> zewnętrzne katalogi dyrektyw include (wpisanie ścieżek)

- a. (odpowiednio dla użytkownika) ..\Sokoban\Libraries\SDL2\include
- b. (odpowiednio dla użytkownika) ..\Sokoban\Libraries\SDL2_image-2.0.1\include
- c. (odpowiednio dla użytkownika)
..\Projekt\Sokoban\Libraries\SDL2_ttf\include

2. Właściwości konfiguracji -> konsolidator -> ogólne -> dodatkowe katalogi biblioteki (wpisanie ścieżek)

- a. (odpowiednio dla użytkownika)
..\Projekt\Sokoban\Libraries\SDL2_ttf\include\Projekt\Sokoban\Libraries\SDL2_ttf\lib\x86
- b. (odpowiednio dla użytkownika)
..\Projekt\Sokoban\Libraries\SDL2_ttf\include\Sokoban\Libraries\SDL2_image-2.0.1\lib\x86
- c. (odpowiednio dla użytkownika)
..\Projekt\Sokoban\Libraries\SDL2_ttf\include\Sokoban\Libraries\SDL2\lib\x86

3. Właściwości konfiguracji -> konsolidator -> dane wejściowe -> dodatki zależności (wpisanie nazw plików)

- a. SDL2.lib
- b. SDL2main.lib
- c. SDL2_image.lib
- d. SDL2_ttf.lib

4 Specyfikacja wewnętrzna

Program został zrealizowany zgodnie z paradygmatem strukturalnym. W programie rozdzielono interfejs (komunikację z użytkownikiem) od logiki aplikacji (działania na wprowadzanych danych).

4.1 Ogólna struktura programu

W klasie głównej wywoływane są wszystkie funkcje obsługujące działanie jak i wprowadzanie danych z klawiatury. Plik **main.cpp** to podstawa generowania i wywoływania okna, funkcji odświeżających oraz renderujących wszelakie obiekty. Głównym plikiem działania programu jest **Game.cpp** który po kolei tworzy okno, obiekty oraz wywołuje funkcje dotyczące działania poszczególnych funkcji programu. Każda klasa nazywa się odpowiednio do jej zadania i działania. Po uruchomieniu programu pierwsza funkcja wywołująca się to funkcja z klasy menu. Prezentująca nam możliwości działań, do dyspozycji jest start gry, tablica wyników oraz zamknięcie programu. Odpowiednio od wyboru wykonują się kolejne funkcje i działanie programu.

4.2 Szczegółowy opis typów i funkcji

Szczegółowy opis typów i funkcji zawarty jest w załączniku.

5 Testowanie

Program został przetestowany pod względem czasu działania, wycieku pamięci oraz danych wprowadzanych po zakończeniu gry. Czas działania jest zależny od użytkownika, bez znaczenia, ile program będzie załączony. Wyciek pamięci nie występuje w opisywanym programie. Ilość pamięci jaką zużywa to stałe 32 MB. Dane wprowadzane przez gracza są sprawdzane i odpowiednio do wprowadzanej wartości wywoływana jest reakcja.

6 Wnioski

Program o tematyce Sokoban to niewątpliwie skomplikowane zadanie. Złożoność całokształtu jakim jest gra komputerowa pokazało mi skalę bardziej wymagających projektów, ile pracy, czasu oraz nerwów trzeba włożyć by stworzyć coś naprawdę dobrego. Największy problem podczas pisania rozwiązania do takowego programu było zrozumienie od podstaw założeń, działania oraz zaplanowanie całokształtu. Najbardziej wymagające okazało się stworzenie poprawnie działającej kolizji obiektów.

Literatura

[1] Dokumentacja C++ zawarta na stronie internetowej en.cppreference.com,

[2] Materiał audio-video

„https://www.youtube.com/watch?v=QQzAHcojEKg&list=PLhfAbcv9cehhkG7ZQK0nfIGJC_C-wSLrx&ab_channel=Let%27sMakeGames”

zawarty na stronie internetowej YouTube.

Dodatek

Szczegółowy opis typów i funkcji