Dokumentacja projektu *Labirynt* JIMP2

Norbert Gwiazda i Bartosz Piłat

16.04.2024

Spis treści:

1 Problem

W ramach projektu pt. "Labirynt" utworzyliśmy aplikacje w jezyku C, której celem jest rozwiazywanie labiryntów. Program ten sczytuje pliki w formie tekstowej lub binarnej, zawierajace odpowiadajace im reprezentacje labiryntów. Dla plików tekstowych znaki "X" wyznaczaja ściany labiryntu, spacje wolna przestrzeń, "P" poczatek, a "K" koniec. Natomiast dla plików binarnych, poczatek, koniec, znaki ścian oraz ścieżek opisane sa w nagłówku pliku, a ich ułożenie opisane poprzez powtarzajace sie słowa kodowe. Dla takich plików wczytanych na wejściu, program znajduje ścieżke bedaca rozwiazaniem ich labiryntu.

Oto przykładowy wygenerowany labirynt w formie pliku tekstowego:

```
XXXXXXXXXXXXXXXXXXXXX
P     X     X       X
X X XXX X X X XXX X X
X X X   X X X X X X X
X XXX XXX XXX X X XXX
X X   X X   X X X   X
X X XXX XXX X X XXX X
X X X     X   X     X
X X XXX X XXXXXXXXXX
X   X   X X       X X
X XXX XXX X XXX X X X
X X   X   X   X X X X
X XXX XXX XXXXX X X X
X   X   X       X X X
XXX X X XXXXXXXXX X X
X   X X X     X X X X
X XXX X X XXX X X X X
X X   X     X X X   X
X XXXXXXXXXX X XXX X
X             X     K
XXXXXXXXXXXXXXXXXXXXX
```

2 Algorytm rozwiazania

Program tworzy strukture znacznika, posiadajaca zmienne współrzednych X i Y oraz zmienna określajaca kierunek, w którym jest skierowany. Oś X oznacza kolumny labiryntu, a oś Y rzedy pliku tekstowego zawierajacego labirynt.

W celu uruchomienia programu użytkownik wybiera plik określajac jego nazwe po fladze "-f", a nastepnie wybiera tryb programu ("-d" lub "-k").
W przypadku wybrania "-d", należy podać plik w formie tekstowej, a program sam znajdzie rozwiazanie labiryntu.
W przypadku wybrania "-k", należy podać plik w formie binarnej, który program najpierw przetłumaczy na forme tekstowa, a nastepnie znajdzie rozwiazanie zakodowanego labiryntu.

Zasady działania programu sa nastepujace:

- Znacznik rozpoczyna ruch po labiryncie odczytanego z pliku tekstowego zaczynajac od znaku P i kończac na znaku K, przy założeniu, że labirynt posiada rozwiazanie
- Znacznik porusza sie stale wzdłuż prawej ściany
- Jeżeli znacznik natrafi na ślepy zaułek, rozpoczyna zalepianie. Zalepianie polega na wstawianiu w labirynt znaków X, aż do napotkania skrzyżowania. Uniemożliwia to ponowne wejście znacznika w dany ślepy zaułek.
- Po dojściu na koniec labiryntu, znacznik stawiany jest z powrotem na poczatek labiryntu. Robimy to ponieważ skoro znacznikowi udało sie dojść na koniec labiryntu, zalepiajac po drodze ślepe zaułki, oznacza to że znacznik przechodzacy ponownie labirynt, zaczynajac od jego poczatku pójdzie prosto do jego końca.
- W trakcie drugiego przejścia labiryntu, ruchy znacznika zapisywane sa do pliku, a na labiryncie wynikowym poszczególne kroki przejścia zaznaczone sa przy pomocy znaków "*".

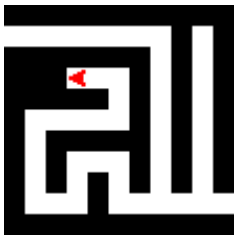
3 Graficzne przedstawienie działania programu

Znacznik zaczyna na poczatku labiryntu

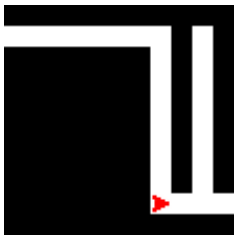
Znacznik porusza sie wzdłuż prawej ściany. Nagle napotyka ślepy zaułek.

Znacznik rozpoczyna zalepianie ślepego zaułku.

Znacznik kontynuuje podróż wzdłuż prawej ściany.

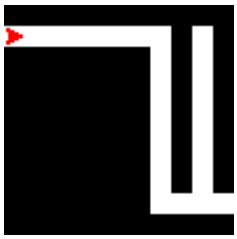


Znacznik ponownie napotyka ślepy zaułek.

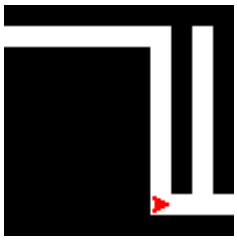


Zaułek zostaje zalepiony.

Znacznik dochodzi do końca labiryntu!



Znacznik zostaje przeniesiony na poczatek labiryntu w celu dokonania ponownego przejścia, tym razem wyznaczajac trase rozwiazania.



Znacznik dochodzi do końca, trasa została wyznaczona, a program kończy swoje działanie zapisujac wyniki w katalogu "wyniki"

```
XXXXXXXXXXX
P*******X X
X XXXXX*X X
X X   X*X X
XXXXX X*X X
X     X*X X
X XXXXX*X X
X X   X*X X
X X X X*X X
X   X  ***K
XXXXXXXXXXX
```

```
START
FORWARD 7
TURNRIGHT
FORWARD 8
TURNLEFT
FORWARD 3
STOP
```

Rozwiazanie labiryntu oraz zapis przejść utworzone w katalogu "wyniki"

4 Obsługa programu

Przed uruchomieniem programu należy go skompilować oraz przygotować do uruchomienia. Obie te czynności zawarte sa w pliku Makefile. Plik Makefile uruchamiamy poleceniem *make*. Po uruchomieniu Makefile wszystkie moduły programu sa kompilowane, a sam plik wykonywalny zapisywany w ścieżce bin/labirynt. Oprócz kompilacji plik Makefile czyści również katalogi *tmp* oraz *wyniki*
Po skompilowaniu programu możemy go uruchomić.Program uruchamiamy z poziomu katalogu Labirynt, wpisujac polecenie *./bin/labirynt*, po którym użytkownik musi podać plik z folderu *dane* zawierajacy labirynt określajac jego nazwe po fladze "-f", a nastepnie wybierać tryb programu ("-d" lub "-k") w zależności od rodzaju podanego pliku ("-d" - tekstowy; "-k" - binarny).

W przypadku wybrania "-d", należy podać plik w formie tekstowej. Program po wczytaniu takiego labiryntu poda jego: nazwe, wybrany tryb programu, czy udało sie odczytać plik, liczbe wierszy oraz kolumn labiryntu, umiejscowienie poczatku oraz końca labiryntu oraz poczatkowa charakterystyke znacznika z miejsca, z którego rozpoczyna sie wyszukiwanie rozwiazania.

Przykładowe wywołanie programu: *./bin/labirynt -f labirynt.txt -d*

W przypadku wybrania "-k", należy podać plik w formie binarnej, który program najpierw przetłumaczy na forme tekstowa. Program po odczytaniu takiego labiryntu poda jego: nazwe, wybrany tryb programu, czy udało sie odczytać plik binarny, ID pliku binarnego, znak escape, liczbe kolumn oraz rzedów labiryntu, współrzedne X i Y (liczone od 0) poczatku oraz końca labiryntu, liczbe słów kodowych, Solution_offset określajacy czy rozwiazanie labiryntu zawarte jest w pliku binarnym oraz określenie znaków separatora, ściany i ścieżki, które składaja sie na słowa kodowe. Po zdekodowaniu pliku binarnego użytkownik otrzyma komunikat, czy plik binarny zawiera już rozwiazanie.

Przykładowe wywołanie programu: *./bin/labirynt -f labirynt.bin -k*

Kiedy program zakończy swoje działanie otrzymujemy komunikat: *"Znaleziono rozwiazanie labiryntu"*, jest to potwierdzenie faktu, że program poprawnie odnalazł rozwiazanie podanego labiryntu przez użytkownika.

W celu odczytania wyników należy otworzyć katalog *"wyniki"*. Graficzne rozwiazanie oznaczone znakami *"*"* zapisane jest w pliku *"sciezka_rozwiazujaca_labirynt.txt"*, a zapis kolejnych kroków potrzebnych do rozwiazania labiryntu dla pliku tekstowego zapisany jest w pliku *"zapis_przejsc.txt"*, natomiast w przypadku pliku binarnego zapis_przejsc okreslony jest w sekcji Solution_offset poprzez liczbe kroków rozwiazania, na które składa sie kierunek (N,E,S,W) oraz liczba pól do przejścia przez znacznik.

5 Opis implementacji

Program składa sie z 8 modułów:

- labirynt.c
- analiza_labiryntu.c
- obsluga_argumentow.c
- odczyt_labiryntu.c
- odczyt_pliku_binarnego.c
- poruszanie_znacznika.c
- zapis_trasy.c
- znalezienie_dowolnego_przejscia.c

Moduł labirynt.c jest rdzeniem programu. To tam spotykaja sie ze soba funkcje wszystkich modułów. Funkcja main wywołuje moduł odczytu argumentów, na podstawie wyniku, którego określa co dalej powinien robić program. Określa również charakterystyke poczatkowa znacznika i sprawdza, czy ma ona sens wyświetlajac komunikat o błedny przypadku skonstruowania labiryntu, jeśli jest on niepoprawny.

Moduł analiza_labiryntu.c odpowiada za określenie liczby kolumn i rzedów pliku tekstowego, określenie punktu startowego i końcowego oraz utworzenie pliku pomocniczego, który bedzie edytowany wraz z przejściami wskaźnika.

Moduł obsluga_argumentow.c odpowiada za obsługe odpowiednich flag "-f", "-d" oraz "-k". Na podstawie podanych flag odczytuje nazwe podanego pliku oraz zapisuje wybrany przez użytkownika tryb odczytu i podaje go z powrotem do modułu main.

Moduł odczyt_labiryntu.c odpowiada za dwie funkcje, umożliwiajace kolejno: sczytywanie bloku przed znacznikiem i bloku, na którym obecnie znajduje sie wskaźnik. Funkcje maja zastosowanie w obsłudze ruchu wskaźnika. O ile odczytywanie aktualnego bloku wymaga znajomości jedynie położenia wskaźnika, sczytywanie bloku przed znacznikiem wymaga znajomości kierunku, w którym porusza sie wskaźnik.

Moduł odczyt_pliku_binarnego.c odkodowuje plik binarny, w przypadku wybrania flagi "-k" oraz przypisuje odpowiednim zmiennym wartości. Moduł tłumaczy również plik binarny na plik tekstowy możliwy do odczytania przez człowieka i zapisuje go jako *przetlumaczony_plik_binarny*.

Moduł poruszanie_znacznika.c odpowiada za szereg funkcji, które obsługuja ruch wskaźnika po pliku labiryntu. Funkcja *ile_przejść* sprawdza, czy wskaźnik znajduje sie obecnie na skrzyżowaniu, zwykłej drodze, czy w ślepym zaułku. Każdy z tych scenariuszy obsługiwany jest oddzielnie. Funkcja poruszanie po labiryncie jest główna funkcja całego modułu. Obsługuje ona przede wszystkim ruch wskaźnika po zwykłej drodze, ale wywołuje również funkcje odpowiedzialne za zalepianie labiryntu. Funkcja ruch do przodu odpowiada za poruszanie sie wskaźnika do przodu, a funkcja zmiana kierunku znacznika za, jak nazwa wskazuje, zmiane kierunku ruchu wskaźnika. Funkcja zalepianie obsługuje zalepianie ślepych zaułków znakami ścian.

Moduł zapis_trasy.c obsługuje utworzenie pliku wynikowego tj. labiryntu z zaznaczona ścieżka oraz pliku z kolejnymi przejściami wskaźnika na planszy.

Moduł znalezienie_dowolnego_przejscia.c przy pomocy funkcji *znalezienie_dowolnego_przejscia* obsługuje funkcje zwiazane z tworzeniem pliku pomocniczego oraz wynikowego, jak i określanie parametrów labiryntu oraz znacznika, znajdujace sie w innych modułach.