

# Maze Master

Dokumentacja funkcjonalna

Jędrzej Gołaszewski i Szymon Stasiak

March 5, 2024

## 1 Opis główny

### 1.1 Nazwa programu

Maze master

### 1.2 Opis problemu

Problemem, który rozwiązuje program jest znalezienie optymalnej trasy pomiędzy punktem początkowym a końcowym w labiryncie. Program musi przeanalizować strukturę labiryntu, zidentyfikować przeszkody i wyznaczyć najkrótszą ścieżkę, umożliwiając efektywne nawigowanie od punktu startowego do celu. To narzędzie stworzone jest w celu automatyzacji procesu rozwiązywania labiryntów, eliminując konieczność ręcznego poszukiwania optymalnej drogi i umożliwiając szybkie, skuteczne rozwiązanie tego zadania

### 1.3 Użytkownik docelowy

Użytkownikiem docelowym programu jest osoba poszukująca rozwiązania labiryntu najkrótszą ścieżką. Program może mieć zastosowanie w badaniu algorytmów nawigacyjnych, grach komputerowych czy rozwiązywania łamigłówek

## 2 Opis funkcjonalności

### 2.1 Jak korzystać z programu

Aby skorzystać z programu należy wpisać komendę: `./MazeMaster [argumenty programu]`

### 2.2 Uruchamianie programu

Lista parametrów programu:

#### 2.2.1 Dane wejściowe [-i|- -in]

Podajemy nazwę pliku z labiryntem jeśli jest w katalogu in bądź pełną ścieżkę pliku

#### 2.2.2 Dane wyjściowe [-o|- -out]

Podajemy nazwę pliku który będzie zawierał nasze dane wyjściowe

#### 2.2.3 Pomoc [-h|- -help]

Flagi pomocy (wpisanie ich spowoduje że program wypisze pomoc ale nie uruchomi się)

#### 2.2.4 Verbose [-v|- -verbose]

Flagi które sprawią że program będzie wypisywał więcej informacji które mogą być przydatne

#### 2.2.5 Debug [-d|- -debug]

Flagi które sprawią że program będzie wypisywał informacje potrzebne do debugowania

### 2.3 Możliwości programu:

#### 2.3.1 Poprawnie wczytanie danych labiryntu

#### 2.3.2 Zapisanie danych wyjściowych w czytelnej formie

#### 2.3.3 Znalezienie najkrótszej drogi z miejsca A do B

#### 2.3.4 Znalezienie najkrótszego wyjścia z labiryntu

## 3 Format danych i struktura plików

- 3.1 **input/** Katalog zawierający pliki wejściowe z labiryntem.
- 3.2 **output/** Katalog zawierający pliki wyjściowe z wynikami działania programu.
- 3.3 **Dane wejściowe** Dane wejściowe programu są plikami tekstowymi opisującymi strukturę labiryntu. Każdy plik wejściowy zawiera informacje o labiryncie, gdzie poszczególne znaki reprezentują różne elementy labiryntu, takie jak ściany, wolne przestrzenie, punkt startowy i punkt końcowy.
- 3.4 **Dane wyjściowe** Dane wyjściowe programu są również plikami tekstowymi, w których zawarte są wyniki działania programu, takie jak znaleziona najkrótsza ścieżka z punktu startowego do punktu końcowego oraz ewentualne dodatkowe informacje diagnostyczne.

## 4 Scenariusz działania programu

Przedstawiony poniżej jest ogólny scenariusz działania programu:

1. Użytkownik uruchamia program z odpowiednimi parametrami, wskazując plik wejściowy z labiryntem.
2. Program wczytuje dane z pliku wejściowego, analizuje strukturę labiryntu oraz znajduje najkrótszą ścieżkę z punktu startowego do punktu końcowego.
3. Program zapisuje wyniki działania do pliku wyjściowego w czytelnej formie.
4. Opcjonalnie, jeśli użytkownik wybrał tryb debugowania lub verbose, program wypisuje dodatkowe informacje diagnostyczne.

## 5 Testowanie

### 5.1 Ogólny przebieg testowania

Testowanie programu będzie przeprowadzone z wykorzystaniem odpowiednich narzędzi, takich jak JUnit dla testów jednostkowych oraz testowanie ręczne podczas tworzenia aplikacji. Ważnym elementem testowania będzie także zapewnienie odporności programu na różne sytuacje wyjątkowe oraz błędy, aby zapewnić jego stabilność i niezawodność w działaniu.