

Politechnika Świętokrzyska
Wydział Elektrotechniki, Automatyki i Informatyki

Dokumentacja projektu zespołowego
Wizualizacja wyszukiwania drogi w labiryncie

Filip Stępień Rafał Grot
Nr indeksu: 094117 Nr indeksu: 094046

Informatyka, grupa 3ID11B

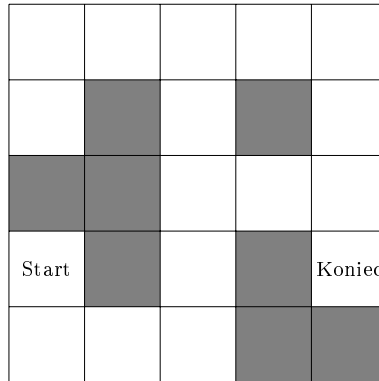
26 maja 2025

Spis treści

1	Wstęp	3
2	Generowanie labiryntu	4
2.1	Algorytm Prima	4

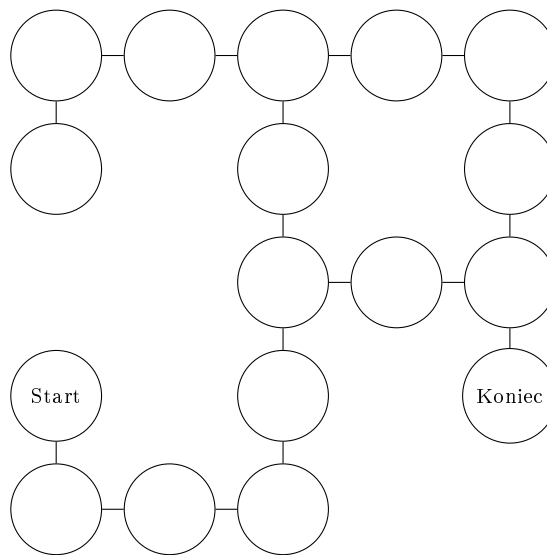
1 Wstęp

Celem projektu jest stworzenie aplikacji umożliwiającej generowanie dwuwymiarowego labiryntu oraz wizualizację procesu wyszukiwania ścieżki pomiędzy dwoma punktami. Labirynt w kontekście projektu to struktura siatki, gdzie każde pole może stanowić przejście lub ścianę.



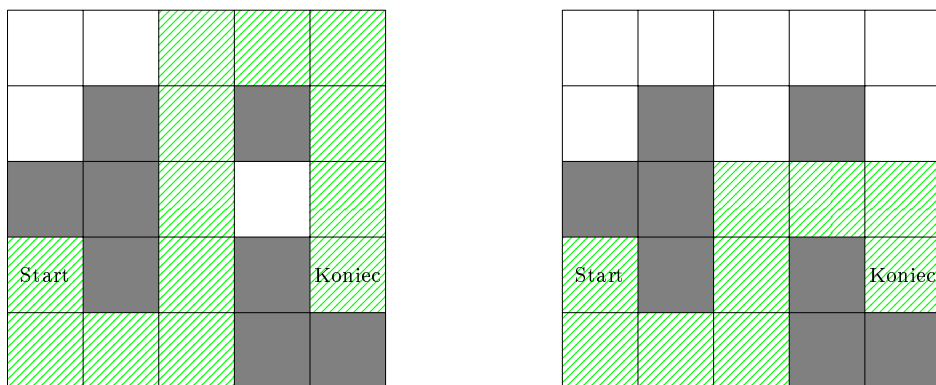
Rysunek 1: Przykładowy labirynt 5x5 z zaznaczonym startem i końcem, gdzie białe pole - przejście, czarne - ściana.

Można zauważyć, że taka struktura jest reprezentacją grafu, gdzie pola odpowiadają wierzchołkom, a krawędzie łączą sąsiadujące pola przejściowe.



Rysunek 2: Graf reprezentujący labirynt z rys. 1.

W projekcie istotne jest porównanie różnych algorytmów wyszukiwania ścieżki, które pozwalają znaleźć trasę między dwoma punktami. W najlepszym przypadku celem jest znalezienie ścieżki *optymalnej*, czyli takiej, która minimalizuje liczbę kroków, co w kontekście grafu o jednakowych wagach krawędzi sprowadza się do znalezienia drogi o minimalnej długości.



Rysunek 3: Porównanie ścieżek w labiryncie. Po lewej stronie – ścieżka nieoptymalna, po prawej – optymalna.

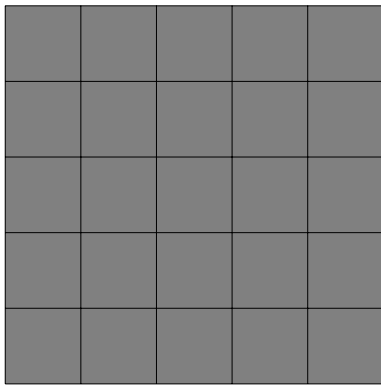
2 Generowanie labiryntu

2.1 Algorytm Prima

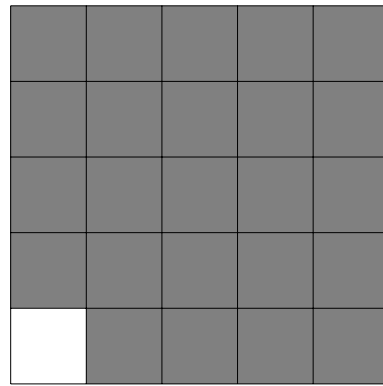
Algorytm Prima to metoda generowania labiryntów wykorzystująca technikę tworzenia minimalnego drzewa rozpinającego (MST) dla grafu reprezentującego planszę labiryntu. W kontekście generowania labiryntu działanie algorytmu przebiega następująco:

1. Na początku tworzona jest plansza, w której wszystkie pola są oznaczone jako ściany.
2. Następnie wybierane jest losowe pole i oznaczane jako przejście.
3. Do zbioru krawędzi dodawane są sąsiednie pola, do których można przejść z wybranego pola.
4. Losowana jest jedna krawędź ze zbioru. Jeśli prowadzi ona do nieodwiedzonego pola, to tworzy się przejście pomiędzy bieżącym polem a nowym (usuwana zostaje ściana między nimi), a nowe pole zostaje oznaczone jako przejście.
5. Do zbioru krawędzi dodawani są sąsiedzi nowego pola.
6. Proces powtarza się, dopóki zbiór krawędzi nie będzie pusty.

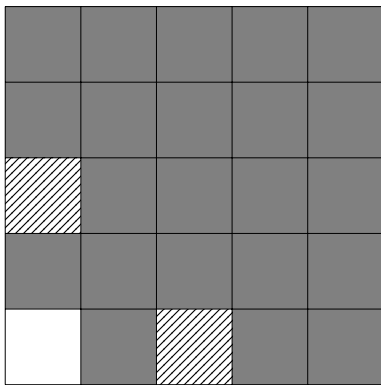
Algorytm ten gwarantuje, że wygenerowany labirynt nie będzie zawierał zamkniętych pętli, a pomiędzy dowolnymi dwoma punktami zawsze będzie istniała możliwa ścieżka. Charakteryzuje się ponadto dużą liczbą krótkich, ślepych zaułków, co w przypadku większych plansz znacząco utrudnia szybkie odnalezienie rozwiązania. Ilustracja działania algorytmu Prima została przedstawiona na rysunkach 4-9.



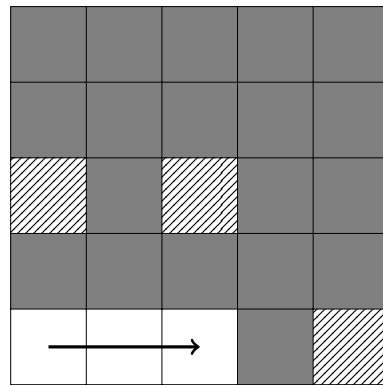
Rysunek 4: Krok 1. Cała plansza stanowi ściany.



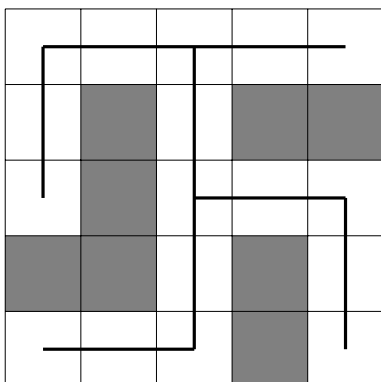
Rysunek 5: Krok 2. Wybierane jest losowe pole startowe.



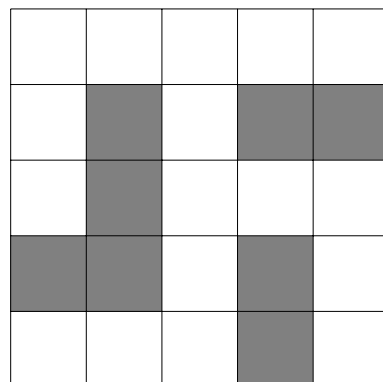
Rysunek 6: Krok 3. Wybór sąsiednich pól, do których można poprowadzić krawędzie.



Rysunek 7: Krok 4. Losowy wybór sąsiedniego pola i utworzenie nowej krawędzi.



Rysunek 8: Ilustracja kolejnych kroków wyznaczania krawędzi w algorytmie Prima.



Rysunek 9: Labirynt powstały na podstawie wyznaczonych krawędzi z Rysunku 8.