***Dokumentacja techniczna aplikacji „PUZZLE”***

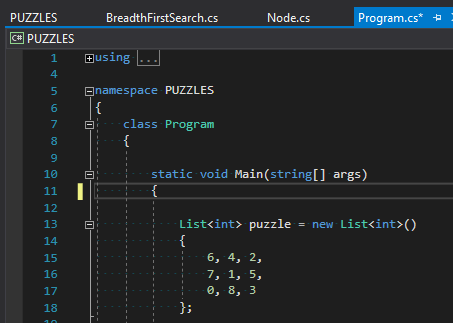
1. *Wstęp:*

„15 Puzzli” to przesuwana łamigłówka, składająca się z tabeli o wymiarach posiadająca ponumerowanych płytek z numerami od 1 do w dowolnym porządku. Ponadto tabela zawiera jedno puste pole. Celem układanki jest uszeregowanie puzzli w kolejności od numeru najmniejszego do największego, przesuwając puzzle przy wykorzystaniu pustego pola.

Kod programu został napisany w języku C# oraz IDE VisualStudio2017. Ponadto zostały utworzone dwie wersje oprogramowania: DEBUG oraz RELEASE.

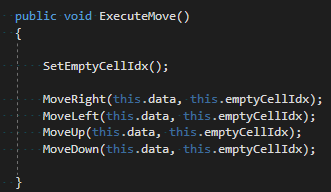
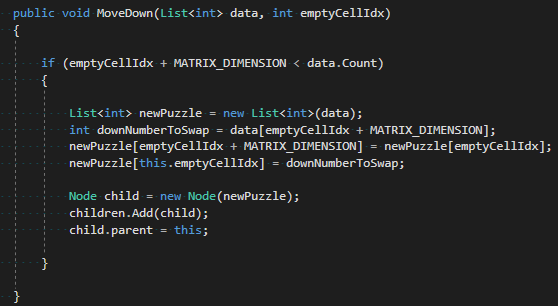
1. *Wersja DEBUG:*

W celu uruchomienia aplikacji w wersji DEBUG należy załadować solucję PUZZLES.sln do kompilatora (z wersją .NET przynajmniej 4.6.1). W tej wersji oprogramowania użytkownik może modyfikować układankę poprzez wprowadzenie danych do poniższej listy:



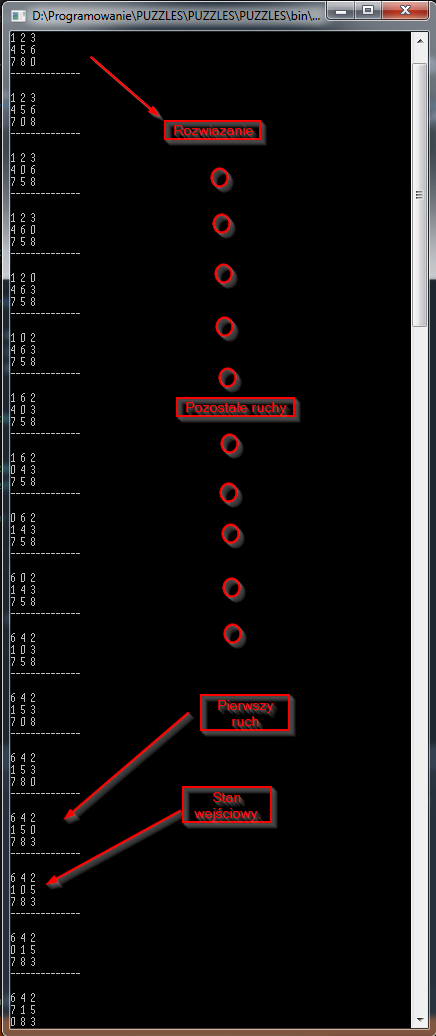
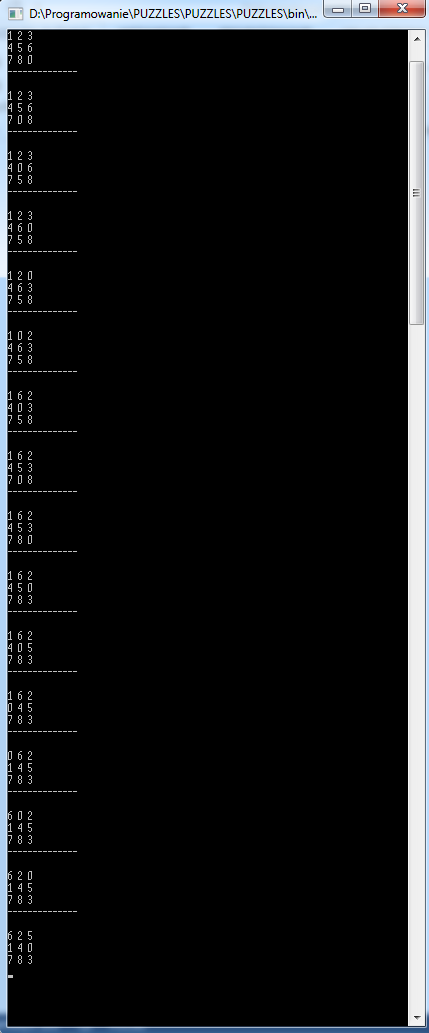
Zdj. 1. Lista z odpowiednimi puzzlami.

Następnie po zbudowaniu i uruchomieniu aplikacji, program poszukuje rozwiązania układanki. Wykorzystano algorytm przeszukiwania wszerz (breadth-first search, BFS). Najpierw ustalany jest indeks pustego puzzle (wartość 0). Następnie z zadanej pozycji można wykonać maksymalnie 4 ruchy (prawo, lewo, góra, dół). Jeśli przesuwany puzzle znajduje się przy krawędzi, ruch nie zostanie wykonany, w przeciwnym wypadku do zmiennej zostaną dodane możliwe ruchy.

Zdj. 2. Funkcja wywołująca odpowiednie ruchy w czterech kierunkach.

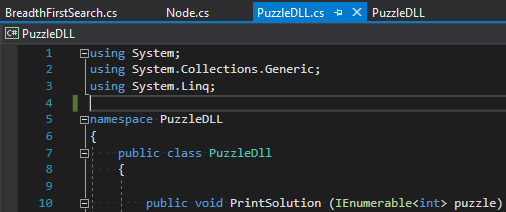
W ten sposób zostaje utworzone drzewo, gdzie każda możliwa kombinacja zostaje dodana, jako nowy węzeł, natomiast stan początkowy układanki zostaje korzeniem drzewa. Warunkami stopu algorytmu są: weryfikacja, czy wprowadzona kombinacja puzzli jest rozwiązywalna, znalezienie rozwiązania, przeszukanie wszystkich możliwych kombinacji. Po znalezieniu rozwiązania algorytm wyświetli w konsoli wszystkie wymagane kroki, jak poniżej:

Zdj. 3. Przykład 1. Zdj. 4. Przykład 2.

1. *Wersja RELEASE:*

W celu uruchomienia aplikacji w wersji RELEASE należy załadować solucję PuzzleDLL.sln do kompilatora. Po zbudowaniu i uruchomieniu programu zostanie wygenerowana biblioteka DLL, która ma jedną publiczną metodę . Metoda ta jest typu void i przyjmuje, jako parametr dowolną kolekcję danych implementująca interfejs . Po wywołaniu tej metody zostaje wywołana konsola i analogicznie znajdywane jest rozwiązanie.



Zdj. 5. Upubliczniona metoda w bibliotece.