

# Cel projektu

Celem projektu jest opracowanie strategii transportu jęczmienia i piwa w Shire tak, aby zminimalizować koszty naprawy dróg, utrzymując maksymalną ilość piwa dostarczanego do karczm.

### Problem wymaga:

- odpowiedniego odwzorowania pól jęczmienia, browarów, karczm i dróg w strukturach danych,
- obliczenia maksymalnego przepływu (jęczmień → browary → piwo → karczmy),
- uwzględnienia kosztów naprawy zniszczonych dróg,
- wprowadzenia zróżnicowanej wydajności pól w różnych ćwiartkach Shire,
- opracowanie sposobu wyszukiwania słów kluczowych w dokumentach (takich jak "piwo", "jęczmień", "browar").



# Podział na podproblemy

Znalezienie maksymalnej ilości piwa, którą można dostarczyć do karczm

Przy zachowaniu ilości przewożonego towaru koszt naprawy dróg był możliwie najmniejszy. Ilości jęczmienia wyrastające w różnych ćwiartkach Shire różnią się od siebie

Zapisanie rozwiązania na przyszłość. Wyszukiwać w rozwiązaniach słów.

### Założenia

- Cały transport odbywa się jednego dnia.
- Jedną krawędzią skierowaną mogą być przepuszczane rożne jednostki towaru na raz.
- Browar po wykorzystaniu pojemności nie przetwarza więcej jęczmienia, ale towar może być przepuszczany dalej.
- Koszt naprawy jest liczony dla każdej wykorzystanej drogi tylko raz.

Poniższe założenia wynikają ze sposobu implementacji projektu oraz ustaleń zespołu:

- Każde pole musi znajdować się wewnątrz ćwiartki.
- Wszystkie wartości zmiennoprzecinkowe ograniczone są do dwóch miejsc po przecinku.



**Język:** C++ (Backend), C# (interfejsy graficzne)

Środowisko: Visual Studio, Unity (GUI),

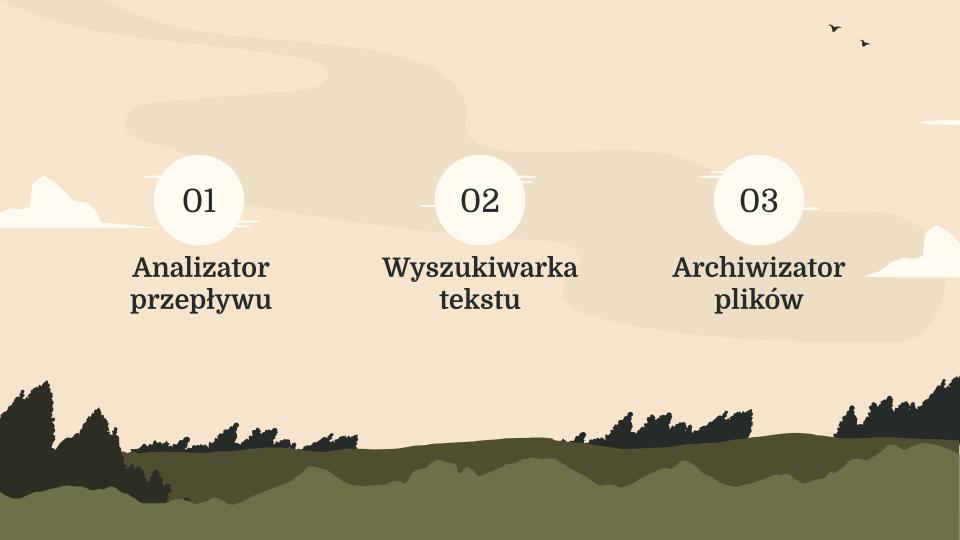
Windows Presentation Foundation (GUI generatora)

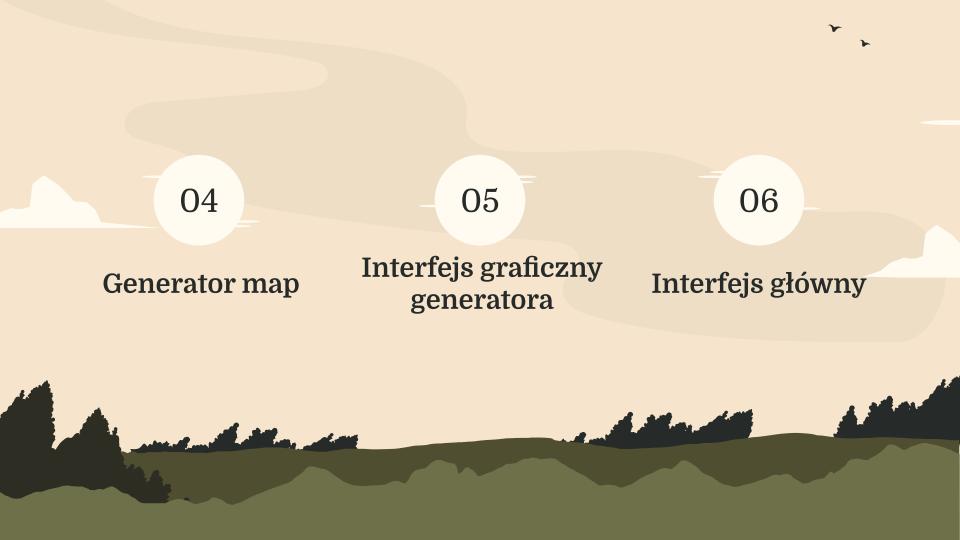






Rozwiązanie składa się z sześciu modułów:





## 01

# Analizator przepływu

Analizator wyznacza maksymalny przepływ o minimalnym koszcie, używając algorytmów
Edmonds'a-Karp'a oraz
Busacker'a-Gowen'a, na podstawie pliku z danymi wejściowymi.

## 02

### Wyszukiwarka tekstu

Wyszukiwarka tekstu – Program służy do wyszukiwania wzorca tekstowego w pliku tekstowym przy użyciu wybranego algorytmu wyszukiwania.

### Obsługiwane algorytmy wyszukiwania:

- Naiwny "Szukaj wszędzie po kolei"
- Rabin-karp "Szukaj po numerkach"
- KMP "Jak już raz coś nie pasuje, to nie wracaj tam"
- Boyer-Moore "Zaczynaj od końca i skacz dalej"

03

# 04

# Archiwizator plików

**Archiwizator plików** - umożliwia kompresję i dekompresję plików za pomocą **algorytmu Huffman'a.** 

### Generator map

Generator map – generuje pliki wejściowe, pozwala również na dostosowanie wyjściowego grafu miasta, różnorodności połączeń czy wyboru pomiędzy połączeniami jednokierunkowymi lub dwukierunkowymi. 05

### Interfejs graficzny generatora

Interfejs graficzny generatora –
pozwala na przyjazne dla użytkownika
wprowadzanie danych do generatora.
Dba o poprawność danych
wejściowych generatora oraz
proponuje rekomendowane wartości.

## Interfejs główny

Interfejs główny - aplikacja zintegrowana z pięcioma poprzednimi modułami, pozwalająca również na wygodne tworzenie i wizualizację danych wejściowych oraz wizualizację danych wyjściowych.



# Zespół

### Karol Ławicki

Koordynator, Programista GUI, integrator

### Dominika Karbowiak

Programistka, autorka dokumentacji, testerka

### Jakub Klonowski

Projektant danych wejściowych, programista, specjalista pomocniczy

### Mateusz Wójciak

Inżynier modelu grafowego, Programista grafów przepływowych

### Weronika Gburek

Programistka, testerka, autorka dokumentacji i prezentacji

Zespół 5 (LE)

# Podział obowiązków

Analizator przepływu ~Mateusz Wójciak

Generator map
Interfejs graficzny generatora
~Jakub Klonowski

Wyszukiwarka tekstu ~Weronika Gburek

~Dominika Karbowiak

05

Interfejs główny ~Karol Ławicki

O3 Archiwizator plików ~Dominika Karbowiak

06

Testy ~Wszyscy

# Szczegółowy podział zadań

#### Mateusz Wójciak

Implementacja struktur danych i algorytmu maksymalnego przepływu, testowanie i rozszerzenie analizatora o koszty (Min-Cost Max-Flow)

#### Dominika Karbowiak

Implementacja kompresji i dekompresji danych metodą Huffmana, eksport danych, testowanie i poprawki modułu wyszukiwania, pomoc przy rozbudowie analizatora przepływu, testy jednostkowe

#### Weronika Gburek

Implementacja algorytmów wyszukiwania wzorców (naiwny, Rabin-Karp, KMP, Boyer-Moore), import danych, przygotowanie prezentacji i dokumentacji, pomoc przy rozbudowie analizatora przepływu, testy wyszukiwarki

#### Jakub Klonowski

Projektowanie danych wejściowych, generowanie sieci residualnych, integracja danych wejściowych z pozostałymi modułami, implementacja obsługi ćwiartek (algorytmów geometrycznych), budowa drzewa Huffmana

#### Karol Ławicki

Planowanie i koordynacja prac projektowych, projekt i implementacja interfejsu graficznego, integracja frontendu z backendem, dokumentacja techniczna, pomoc przy algorytmach geometrycznych (m.in. wypukła otoczka), ogólna pomoc w różnych częściach projektu