
SYMULATOR MIEJSKIEJ SIECI TRAMWAJOWEJ

DOKUMENTACJA

Projektant: Dominika Ferfecka
Nr albumu: 318650
Uczelnia: Politechnika Warszawska

Cel projektu:

Głównym założeniem projektowym było stworzenie symulacji jazdy tramwajów według wcześniej ustalonego rozkładu jazdy.

Opis projektu:

Symulacja pokazuje sieć tramwajową składającą się z licznych przystanków i linii tramwajowych. Tramwaje przemierzają trasę przystanków tramwajowych zgodnie z kolejnością według której są przypisane do danej linii. Czas wyruszenia pierwszego tramwaju jest określony dla każdej linii, a następne tramwaje wyruszają w ustalonym odstępie czasowym. Tramwaje z nieparzystym numerem przypisania do linii wyruszają z pierwszego przystanku, natomiast te z parzystym z ostatniego. Po przejechaniu trasy całej linii, zatrzymują się by następnie przejechać ją ponownie, w przeciwnym kierunku. Czas przejazdu między każdą parą sąsiednich przystankami jest ustalony i stały. Projekt zakłada brak opóźnień i kolizji tramwajowych, a każdy segment torów jest dwukierunkowy. Stan sieci tramwajowej odświeża się co sekundę, odzwierciedlającą minutę w czasie symulacji.

Struktura projektu:

Poszczególne etapy tworzenia projektu zostały podzielone na kilka plików źródłowych.

- **Baza danych:**
database.py – zawiera podstawowe klasy używane w projekcie:
 - *TramNetwork* – klasa sieci tramwajowej posiadająca listę wszystkich przystanków tramwajowych należących do sieci. Umożliwia pobranie czasu przejazdu tramwaju między sąsiednimi przystankami
 - *TramLine* – klasa linii tramwajowej przechowująca nazwę linii, listę kolejnych przystanków należących do tej linii, listę tramwajów poruszających się daną trasą, godzinę pierwszego odjazdu oraz czas pomiędzy odjazdami kolejnych tramwajów
 - *TramStop* – klasa przystanku tramwajowego zawierająca najważniejsze informacje o przystankach takie jak jego id, nazwę, współrzędne na których jest położony, oraz listę sąsiadujących przystanków
 - *Tram* – klasa tramwaju przechowująca aktualne współrzędne położenia tramwaju, linię do której należy, oraz numer pokazujący kolejność przypisania do linii

- **Odczyt danych z plików:**
`database_io.py` – zawiera trzy funkcje pozwalające kolejno:
 - Stworzyć bazę przystanków, poprzez pobranie ich id, nazw i współrzędnych, a następnie utworzenie obiektów klasy `TramStop`
 - Pobranie połączeń sąsiadujących przystanków tramwajowych i czasu przejazdu między nimi
 - Stworzenie bazy linii tramwajowych poprzez pobranie nazw linii i przystanków składających się na ich trasę
- **Graficzny Interfejs Użytkownika:**
`gui.py` – odpowiada za graficzne przedstawienie symulacji:
 - `TramSimulatorWindow` – to klasa odpowiadająca za utworzenie okienka symulacji, wraz ze wszystkimi elementami sieci tramwajowej. Przystanki tramwajowe są przedstawione za pomocą fioletowych kółek i nazw przystanków, na odpowiednich współrzędnych. Sąsiadujące przystanki są połączone liniami. Tramwaje, to niebieskie kółka, podpisane numerem linii do której należy tramwaj.
 - `Clock` – klasa obsługująca i wyświetlająca zegar. Czas zwiększa się z każdym taktem ustawionego stopera. Jedna sekunda naszego czasu odpowiada jednej minucie czasu w symulacji.
- **Uruchomienie:**
`setup.py` – odpowiada za podanie danych źródłowych, wywołanie funkcji tworzących bazę danych, a także uruchomienie symulacji
- **Testy:**
 - `test_database.py` – posiada testy jednostkowe, służące do przetestowania części programu odpowiedzialną za stworzenie bazy danych
 - `test_database_io.py` - posiada testy jednostkowe, służące do przetestowania części programu odpowiedzialną za pobranie danych źródłowych z podanych plików tekstowych

Konfiguracja projektu:

Projekt prezentowany jest na przykładowych danych testowych, przedstawiających 10 linii i 100 przystanków tramwajowych, sporządzonych inspirowanych komunikacją miejską w Krakowie.

Dane testowe są wprowadzane w trzech plikach konfiguracyjnych:

- `tram_stops.txt` – zawiera podstawowe dane o przystankach tramwajowych takich jak id, nazwa, współrzędna x, współrzędna y. ID, nazwa muszą być jednoznaczne i unikalne. Różne przystanki nie mogą być na obu tych samych współrzędnych.

```
1,Teatr Bagatela,20,-70
2,Stary Kleparz,80,-110
```

Przystanek 1: Id: 1, Nazwa: Teatr Bagatela, Położenie: (20,-170)

Przystanek 2: Id: 2, Nazwa: Stary Kleparz, Położenie: (80,-110)

- `Tram_stops_connection.txt` - zawiera informacje o połączeniach między sąsiednimi przystankami: ID przystanków i czas przejazdu między nimi podany w minutach. Para sąsiadujących przystanków może wystąpić tylko raz, kolejność wpisania przystanków nie ma znaczenia.

```
1,2,4
1,10,5
```

Przystanek z ID 1, sąsiaduje z przystankiem o ID 2, a czas przejazdu pomiędzy nimi zajmuje 4 minuty

Przystanek z ID 1, sąsiaduje z przystankiem o ID 10, a czas przejazdu pomiędzy nimi zajmuje 5 minut

- Tram_line.txt – zawiera dane o liniach tramwajowych. Cztery pierwsze liczby rozdzielone przecinkami określają kolejno: numer linii, godzinę i minutę odjazdu pierwszego tramwaju, interwał – co ile odjeżdżają kolejne tramwaje. Następnie występują dowolna dodatnia ilość liczb oznaczająca ID kolejnych przystanków należących do danej linii tramwajowej.

```
1,5,0,20,14,13,12,4,3,8,9,15,96,97,98,99,100  
2,5,30,15,50,49,14,13,1,2,3,8,9,18,19,20
```

Linia 1: Numer: 1, Pierwszy odjazd: 5:00, Czas między odjazdami kolejnych tramwajów: 20 minut, Trasa linii to przystanki o ID: 14,13,12,4,3,8,9,15,96,97,98,99,100

Linia 2: Numer: 2, Pierwszy odjazd: 5:30, Czas między odjazdami kolejnych tramwajów: 15 minut, Trasa linii to przystanki o ID: 50,49,14,13,1,2,3,8,9,18,19,20

Uruchomienie symulacji:

W celu uruchomienia symulacji należy wpisać komendę uruchamiając plik odpowiedzialny za graficzny interfejs użytkownika: python3 gui.py.

Dodatkowo można podać ścieżkę do własnych plików źródłowych wprowadzając komendy opcjonalne:

- `--file-tramstop`
wprowadza ścieżkę do pliku zawierającego dane o przystankach tramwajowych
- `--file-connection`
wprowadza ścieżkę do pliku zawierającego dane połączeniach między przystankami
- `--file-tramline`
wprowadza ścieżkę do pliku zawierającego dane o liniach tramwajowych

Podsumowanie:

Projekt udało mi się ukończyć realizując wszystkie postawione wymagania. Symulacja tramwajowa działa zgodnie z ustalonym rozkładem jazdy, na wskazanej minimalnej ilości danych.

Uważam, że projekt wyszedł całkiem dobrze, poświęciłam na niego sporo czasu, starając się jak najlepiej go stworzyć. Jestem zadowolona z efektu końcowego, mimo że nie zrealizowałam wszystkich swoich pomysłów ulepszenia podstawowej wersji symulacji, takich jak wyświetlenie w wygodny sposób dla użytkownika rozkładu jazdy, głównie z powodu pracowitości projektu i ograniczonego czasu.

Projekt dał mi możliwość nauczenia się wielu rzeczy, mogłam jeszcze lepiej opanować przerobiony już materiał, a także rozszerzyć swoje umiejętności, poznając nowe rzeczy. Do tej pory nie miałam zbyt wiele okazji pracować z graficznym interfejsem użytkownika, więc było to ciekawe wyzwanie. Dodatkowo cieszę się, że poradziłam sobie z tworzeniem i zarządzaniem kodu, w bardziej rozbudowanym projekcie.

Pracowało mi się bardzo przyjemnie, pokonywanie kolejnych trudności dawało sporą satysfakcję, a widoczny postęp w projekcie i moich umiejętnościach dawał motywację do dalszej pracy i zwiększania swoich możliwości.

Podsumowując, jestem zadowolona z mojej pracy i stworzonego projektu, więc mam nadzieję, że zostanie to dostrzeżone, a projekt będzie wysoko oceniony :)