Politechnika Śląska

Wydział Matematyk Stosowanej

Kierunek Informatyka

Gliwice, 24.01.2023

Programowanie I

**projekt zaliczeniowy**

**"Maszyna Turinga"**

**Mateusz Goik gr. lab. 2/3**

**1. Opis projektu.**

*Jest to program z interfejsem graficznym, który symuluje działanie maszyny Turinga. Umożliwia on tworzenie algorytmów za pomocą tablicy charakterystycznej i testowanie ich z ustawioną przez użytkownika taśmą znaków. Jest to nowa wersja podobnego programu, który do tej pory był używany na wydziale Matematyki Stosowanej Politechniki Śląskiej.*

**2. Wymagania**

* *Możliwość implementacji dowolnych algorytmów możliwych do realizacji za pomocą maszyny Turinga,*
* *Przejrzysty i prosty w obsłudze interfejs użytkownika,*
* *Możliwość zapisywania i odczytywania plików z tablicami charakterystycznymi*
* *Wsteczna kompatybilność z plikami starszej wersji symulatora,*
* *Zabezpieczenie przed zamknięciem programu bez zapisu pliku,*
* *Możliwość zmiany języka aplikacji,*
* *Możliwość zmiany motywu aplikacji,*
* *Kompatybilność z systemami operacyjnymi z rodziny Linux oraz Windows.*

**3. Przebieg realizacji**

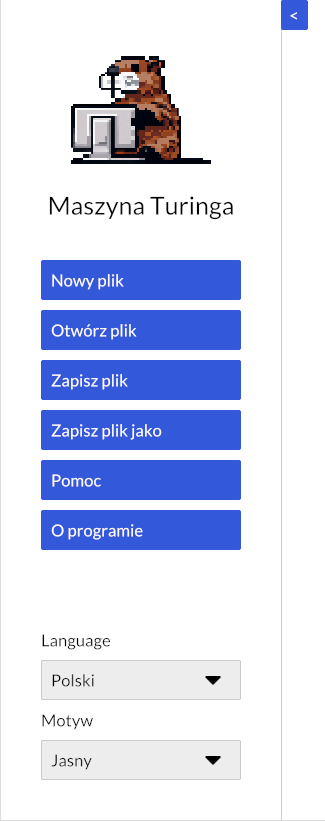
*Za zgodą prowadzącego zajęcia, projekt został wykonany przy użyciu języka programowania Rust, który podobnie jak C++ został stworzony do pisania programów o wysokiej wydajności. Do stworzenia interfejsu graficznego została użyta biblioteka iced, natomiast do obsługi natywnych okien do wyboru plików użyta została biblioteka rfd. Ikona programu została stworzona przez generator DALL·E 2.*

*Najpierw dodane zostały dodane elementy interfejsu takie jak tabela charakterystyczna, taśma ze znakami oraz kilka pól do wpisywania innych parametrów. Następnie zaimplementowane zostało symulowanie działania maszyny Turinga. Później dodana została możliwość otwierania plików z tablicami charakterystycznymi. Na koniec dodane zostały ustawienia języka i motywu aplikacji.*

**4. Instrukcja użytkownika**

Po otworzeniu programu widoczne są dwie części: lewa kolumna z ikoną i prawa z symulatorem.

**Lewa kolumna**



Kolumna po lewej stronie zawiera przyciski umożliwiające kolejno:

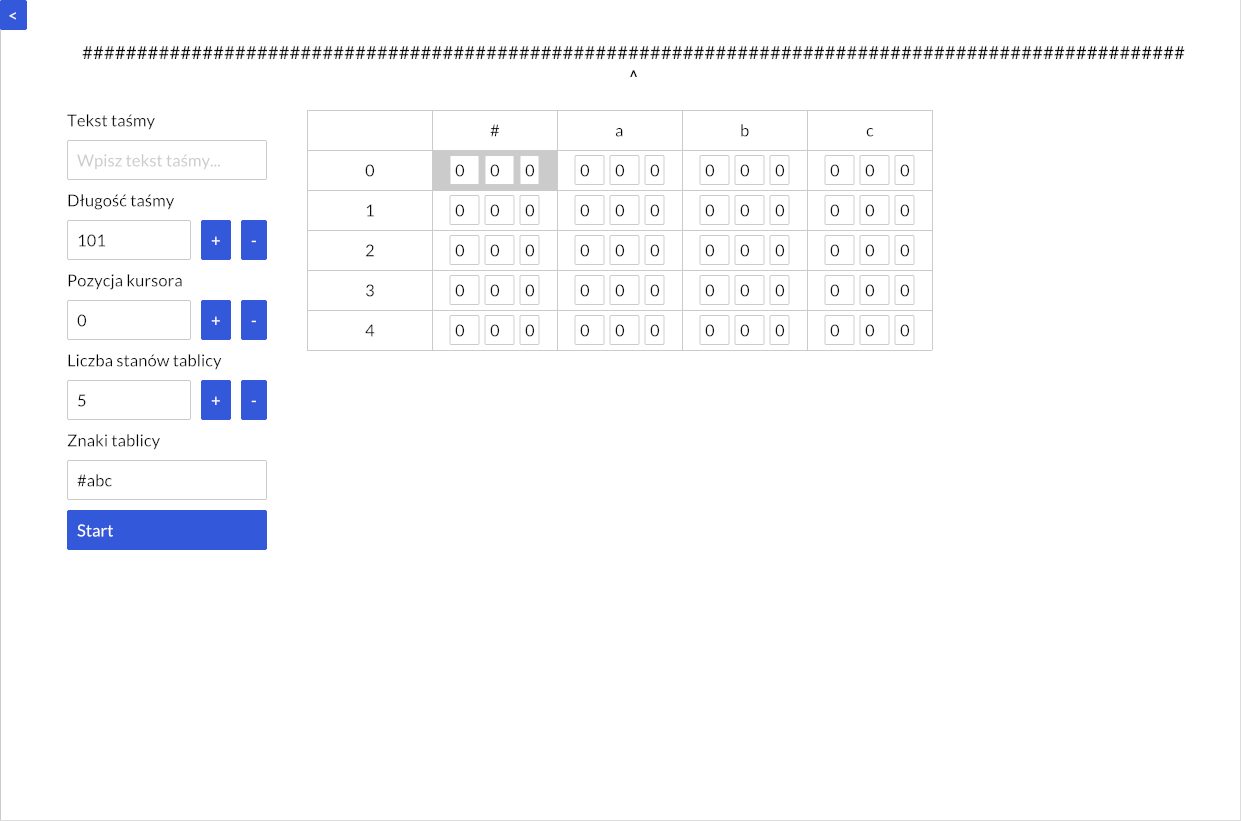
* utworzenie nowego pliku,
* otwarcie zapisanego wcześniej pliku,
* zapisanie pliku,
* zapisanie pliku jako nowy plik,
* otwarcie poradnika jak korzystać z programu (dostępny w języku polskim i angielskim),
* wyświetlenie informacji o programie.

Kolumna po lewej stronie umożliwia również dostosowanie ustawień takich jak:

* język aplikacji (dostępne są polski i angielski),
* motyw aplikacji.

Kolumnę po lewej stronie można otworzyć lub zamknąć za pomocą przycisku u góry po prawej stronie od linii kolumny.

**Symulator**



W trybie edycji (domyślnym) po lewej stronie można dostosować ustawienia takie jak:

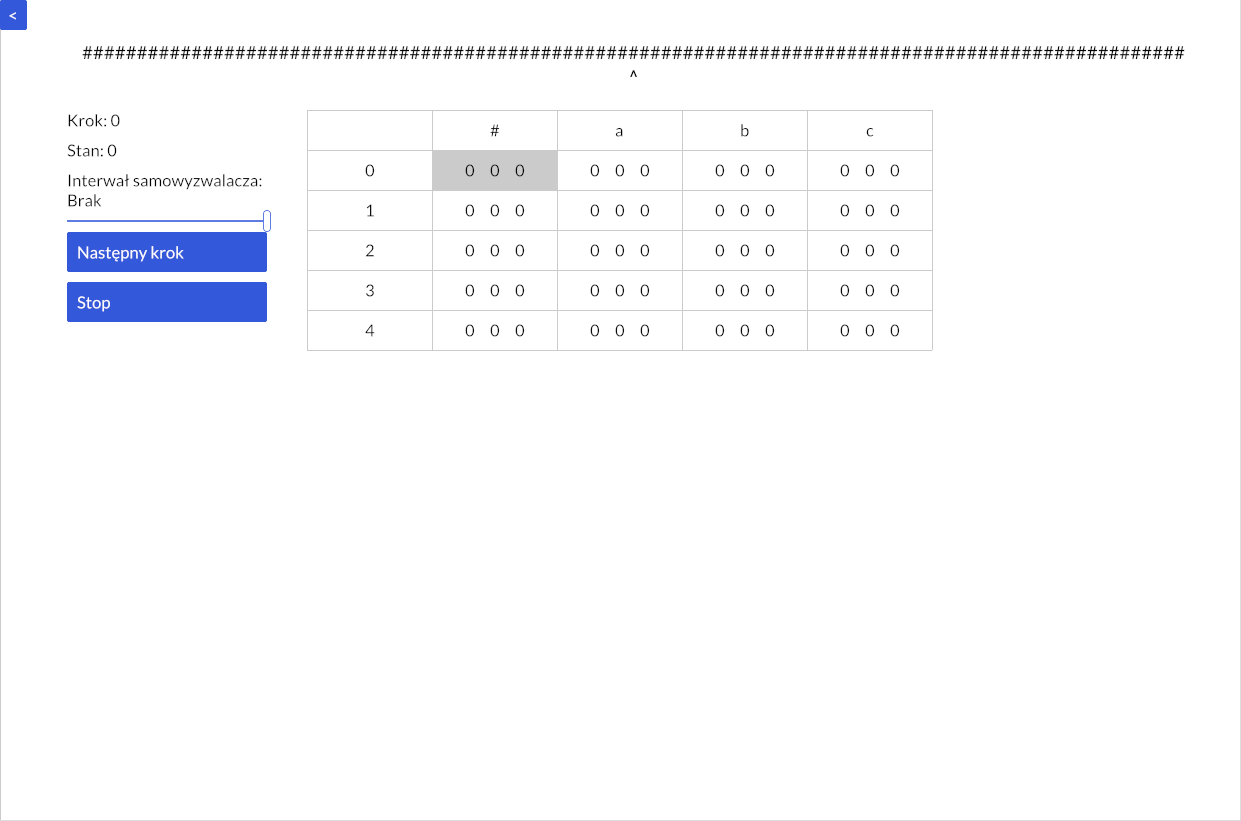
* tekst taśmy,
* długość taśmy,
* pozycja kursora (głowicy),
* liczba stanów tablicy,
* znaki tablicy.

Po prawej stronie znajduje się tabela w której można wpisywać wartości poszczególnych komórek. Wartości te są ustawione w następującej kolejności:

* nowy stan maszyny (od 0 do 99),
* nowy znak na taśmie (dowolny znak poza znakami białymi),
* kierunek ruchu głowicy (+, - lub 0).

U góry znajduje się podgląd początkowych znaków taśmy.

Po ustawieniu wszystkich parametrów można przejść do trybu symulacji klikając przycisk Start.



W tym trybie, po lewej stronie znajdują się:

* informacja o liczbie wykonanych kroków,
* informacja o wewnętrznym stanie maszyny,
* suwak do zmieniania interwału samowyzwalacza maszyny,
* przycisk do ręcznego zmieniania kroków,
* przycisk Stop do powrotu w tryb edycji.

Po prawej stronie znajduje się tabela w której wyświetlają się ustawione wcześniej wartości komórek.

U góry znajduje się podgląd obecnego stanu taśmy.

**5. Podsumowanie i wnioski.**

Projekt ten pozwolił mi w praktyce zastosować język programowania Rust. Pomógł mi on przyswoić bardziej zaawansowane mechanizmy tego języka jakimi są system „wypożyczania” zmiennej (borrowship) i czasu życia zmiennej (lifetimes). Dzięki nim możliwe jest pisanie programów bez obaw o błędy związane z dynamiczną alokacją pamięci, bez użycia systemów automatycznego zarządzania pamięcią (garbage collector).

Pomimo pewnych trudności związanych z młodym wiekiem biblioteki graficznej jak i samego języka, udało się zaimplementować wszystkie zaplanowane funkcje programu.

W ramach dalszego rozwoju programu można do niego dodać więcej języków i motywów do wyboru, a także można dodać do repozytorium więcej plików z przykładowymi tablicami charakterystycznymi.