Universal Turing Machine

About

The Universal Turing Machine is an application written in *C#(.NET 5)* in which you can simulate any Turing machine .
(**Important Note 1:** This app supports only deterministic Turing machines so make sure that your Turing machine is deterministic.)
(**Important Note 2:** To run the framework-dependent version of the application the .NET 5 Runtime is required or you can download the self-contained version which the .NET 5 is integrated into the application(See Releases) .)

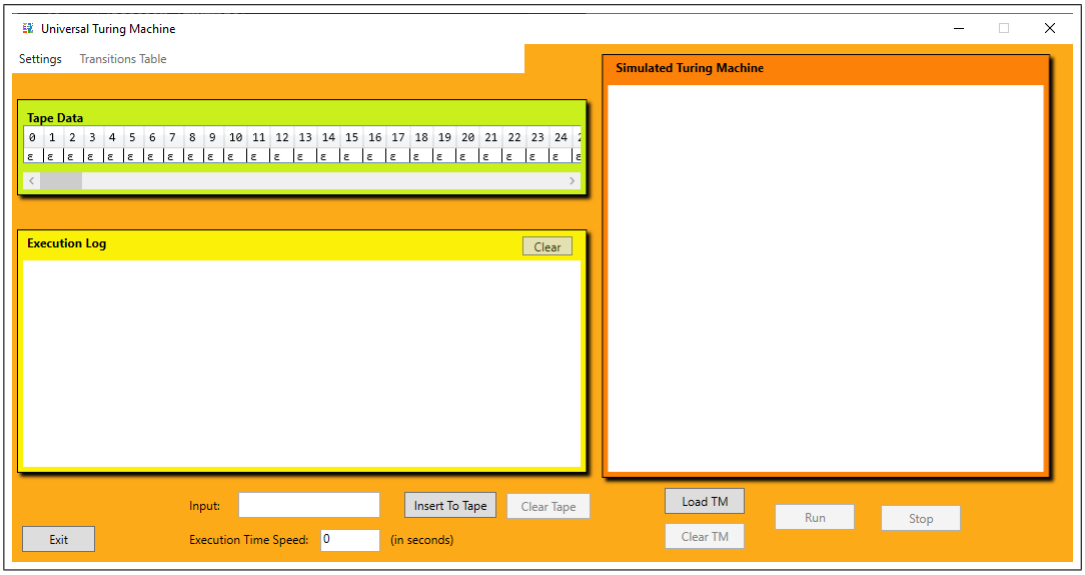

Figure 1: Main window of the application

The application's main window contains three fields :

- The **Tape Data** which simulates the tape that the Turing Machine uses .

- The **Simulated Turing Machine** which it shows the Turing Machine that is imported and will show any errors that may have occur.
- The **Execution Log** which shows the state of the simulation,whether your TM(Turing machine) has accepted or not the input of the tape and diagnostic messages about the input, the Turing Machine, the Tape e.t.c .
- And the menu that contains the **Settings** and the **Transitions Table** .

The application accept only files with *.uTM*(universalTuringMachine) extension. The form of this file format is explained below.

File Format

The format of the files with the *.uTM* extension must be exactly as mentioned or else the application will not function properly.
This file format supports single line comments(using the \\) everywhere inside the document but not next to the lines (ex. States:q1,q2,q3 comment). First you have to start with the **States** tag. In this line you declare the states of your Turing Machine for example: *States:q1,q2,q3*(**Note:** the q1,q2,... can be whatever you want ex. States:a,b,c,dd,ca,ww,... . After the states declaration you have to declare the accepting state by writing the **Accept** tag and then the name of the state(separated by :). Afterwards you have to define the Reject State of the Turing Machine by writing the **Reject** tag and then the name of the state(separated by :). Note that if your machine does not have a rejecting state you can create a new state and assign it as the rejecting state.
Last, you have to define the **Input** tag which defines the symbols that apart your input(separated by :). The symbols are supported are `[a-zA-Z0-9_]` (UTF-8 characters not fully tested.) and the special symbols `[!@#$%^&*]` .(Note: If you type the blank symbol in the input field or define it in the Input tag an error will occur. The same will happen if you type any symbol in the input field that you don't have defined in the Input .(ex. Input: a, b))
The **description** of your Turing Machine is the transitions between your states. The form of the description between two states, for example q1,q2 must be: *q1,<input>* $\rightarrow$ *<write2Tape>,<HeaderMove>,q2.*

- <input> : Is the symbol that the Turing machine must read in the tape.
- <write2Tape> : Is the symbol that the Turing machine will write to the tape. If you don't want to write anything ignore it. (ex. q1,a$\rightarrow$R,q2) .

- <HeaderMove> : Shows the direction of the header. Accept the values ***L***: to move left , ***R***: to move right, ***N***: to stay still .

Example

Below is shown an example of a Turing machine which decides the language $L = \{0^{2^N} | N \geq 0\}$. The *.uTM* file that simulates the Turing machine of this language is:

```
\\States
States: q1, q2, q3, q4, q5, qy, qn
\\Accepting state
Accept:qy
\\Rejecting state
Reject:qn
Input: 0

\\Description
q1,0→ε,R,q2
q2,x→R,q2
q2,ε→R,qy
q2,0→x,R,q3
q3,0→R,q4
q3,ε→L,q5
q3,x→R,q3
q4,x→R,q4
q4,0→x,R,q3
q5,0→L,q5
q5,x→L,q5
q5,ε→R,q2
\\EOF
```

Other examples can be found at the folder *Examples* at the repository.

Settings

This section is about the application's settings which are shown and analyzed below. (Access it by clicking in the Settings at the menu bar shown at Figure 1.)

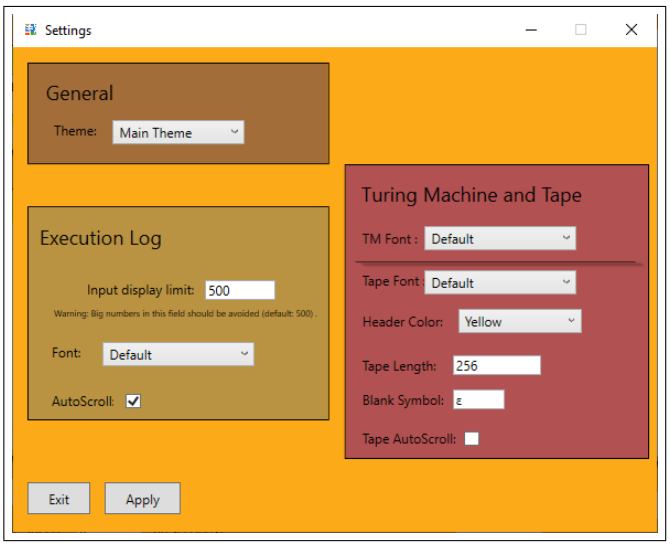

Figure 2: Settings Window

- **General**
  - **Theme** : From here you can change the theme of the application.
- **Execution Log**
  - **Input display limit** : This field sets the max lines will be displayed when a Turing Machine is running(Default Value: 500).
  - **Font** : Sets the font for the Execution Log (Default value : Segoe UI Font).
  - **AutoScroll** : If enabled the Execution log scrolls down automatically if needed .
- **Turing Machine and Tape**
  - **TM Font** : Sets the font for the Simulated Turing Machine (Default value: Consolas Font).
  - **Tape Font** : Sets the font for the Simulated Turing Machine (Default value: Consolas Font).
  - **Header Color** : Sets the header color of the tape.

- **Tape Length** : Sets the length of the tape (Default value: 256)(Note: Bigger numbers will require more memory).
- **Blank Symbol** : Sets the blank symbol in the Tapes Alphabet(Default value: ε).
- **Tape AutoScroll** : If enabled the Tape will scroll horizontally automatically if needed .

Transitions Table

This window shows the transitions of your Turing Machine with more detail. Also the transitions which lead to a final state is colored with green for the accepting state and red for the rejecting state. This is the transition table of the example which showed above .

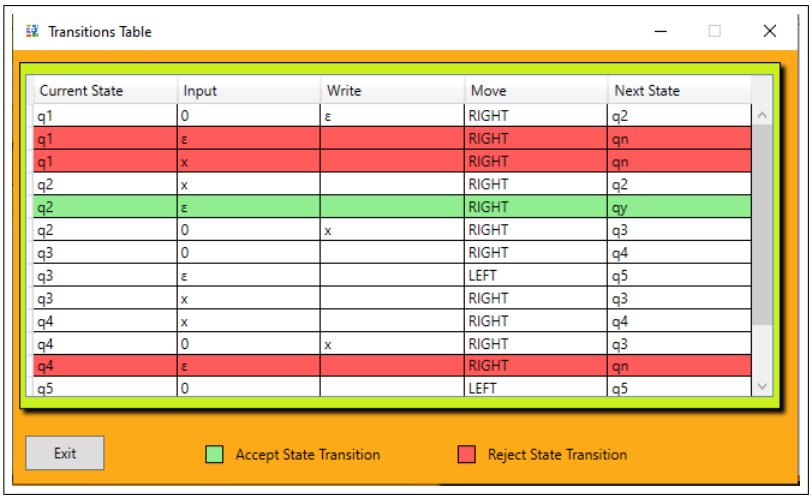

Figure 3: Transitions Window