

## Virtualization of a Turing Machine Weekly Report: 2-21-2021

Outlined in this document details the work and progress toward the making of the Virtualization of a Turing Machine Project for the week of 2-21-2021.

Throughout the week, James Merenda focused on implementing the `machineanimationsdemo.html` assets into the main page. Unfortunately, the "drag-and-drop" method explained in a previous report did not pan out. As a result, minor changes to the CSS code was required to implement the assets properly. Among those features, a new feature to handle user input was also added. When the user enters an input into the input box above the Turing machine tape and clicks the *Enter Input* button, the input entered will be placed at the read-head's location. The speed slider was also adjusted to convey a better message by reversing the direction of the slider. By doing this, the user can expect a faster speed if the slider is moved all the way to the right and slower when moved to the left respectively. A new file was created called *animations.js* to hold all functionality for the tape animations from the original `machineanimationsdemo.html` page.

There was also progress made for the "infinite" tape concept. There will be two separate array objects: one for display and another for transition manipulation. These ideas are explored and somewhat implemented in a file called *mapper.js*.

The display array will handle the contents for each tape cell that is visible to the user. As the tape moves left and right, each position will be updated to compensate for elements not within the number of cells the user can see. For example, an input 8 characters long will not be visible from the initial state. To compensate for this, the display array only holds up to the physical number of cells on the tape, 11. By moving left or right, the goal is to pop and push elements not visible to the array to their respective locations. Then the display array references the main data array to remap and write its values to the tape to reflect a move left or right. It may

be necessary to store these pushed and popped values to a different structure, but more testing will be required to see if this is necessary.

The other array will hold the raw symbols that were provided by the user and modified by user-written transition functions. This structure is not modified by any of the *mapper.js* functions, rather this will be modified by a separate file where values in the tape will be analyzed and realized at runtime. Jisue Lee will focus on this portion of the Turing machine in the coming weeks.

Some problems the team faced during the week include the following. James Merenda did not realize how complicated the "infinite" tape concept would be to implement. Multiple attempts were made to create a solution to this problem and most failed, but by next week - a solution to these complications will be formalized. There was also some confusion with understanding code written for the compiler. But a brief explanation from Brett George helped us understand his programs better.

Plans for next week include the following. James Merenda will continue and finish implementing the "infinite" tape concept. He will also go through his files and begin writing detailed comments and descriptions. Jisue Lee will begin creating a recognizer of sorts to manipulate tape values based on transition functions.