Alternative Definitions for

1.>robot.rkt

Used in the program is (make-robot

coordinate

coordinate

direction)

Pros: This data definition made the functions for movement and handle directions easily.

Cons: This data definition was apt for the functionality that were required to be provided to the robot, hence I do not see any disadvantage of using this approach

Alternate approach could be (make-robot

distance –from-origin

positive-angle-from-x-axis

direction)

Pro: This representation could make non-linear movement of an image easier.

Con: would be very difficult to render the image using this approach. Also robot movement would also be a bit tricky to implement as angle is involved.

2.>fsm.rkt

Used color of the fsm to represent the state of fsm

Pros: Since the state of FSM has one to one relationship with the color of the image, it made very easy to go from show state going from one state to another.

Con: The representation is very trivial to the user and not much information is conveyed in the image, as to what input the particular state can expect and what is the input string thus far etc.

Alternative representation would be a structure (make string string string) where the first string would keep the input keys that have been entered thus far, second string would represent the next valid input the current state expects and finally the third string would represent the color of the state.

Pros: The output would be more informative and user friendly

Cons: This would certainly make the program lengthier and more conditions should be handled than before.

3.>editor.rkt

Used paradigm is (make-editor String String), where first string represents the text before the cursor and the second string represents the text after the cursor.

Pro: Rendering the image gets easier as cursor position inherently exists in the editor-pre and editor-post. Tracking the Cursor position is minimum and will only be used for cursor position is at the boundary

Con: Every time Cursor position, which is actually required very few times in this approach, is required an explicit function that calculates the position is needed.

Alternate paradigm would be to use (make-editor string integer), where the string represents the whole text that would be present in the editor and the integer would represents the cursor position on the 0-based indexing of the string.

Pro: function call to calculate the cursor position is not required as it is wrapped in the editor structure.

Cons: Rendering the final image would be a little clumsy.