User stories (Part 1)

Note
In all the test cases if an Overwrite warning popup appears when loading or creating machines simply click the “Continue” button. If another test has been run the current state of the previous machine will have been temporarily stored.

User story: Create new machine

Test:
Create new machine - from application launch

Rationale:
Users should be able to start a new, blank Turing Machine on launching the application.

Setup:
Start on the menu screen.

Test Procedure:
1. Launch the application
2. Click on “New Machine”
3. Enter the name “Test New” into the text field
4. Click the “Create” button
5. Click the “Ok” button
6. Type “2” to make the text field contain “_012”
7. Click the “Define” button
8. Click the “Menu” button
9. Click “Load Machine”

Expected Results:
1. Display a menu screen with the options: “New Machine”, “Load Machine”, “Import Machine” and “Delete Machine”
2. Display a popup for creating a New Turing machine with a text field for naming it and “Create” and “Cancel” buttons
3. Entered text appears in the text field
4. A popup saying the Machine was saved to “Test New.xml” will appear
5. Display a popup for defining the alphabet with a text field and “Define” and “Cancel” buttons
6. Entered text appears in the text field
7. Popup disappears and the build screen is enabled
8. Returns to the menu screen
9. A popup with a list of saved machines appears, “Test New” is on the list

Test:
Create new machine - while another is open

Rationale:
The user should be able to create a new, blank Turing Machine when they have been working on another.

Setup:
Follow the steps in the "Load Machine" test case

Test Procedure:

1. Click the “Menu” button on the toolbar
2. Click on “New Machine”
3. Enter the name “Test New 2” into the text field
4. Click the “Create” button
5. Click the “Ok” button
6. Add a “2” to the text field so it contains “_012”
7. Click the “Define” button
8. Click the “Menu” button
9. Click “Load Machine”

Expected Results:

1. Display the menu screen with the options: “New Machine”, “Load Machine”, “Import Machine” and “Delete Machine” and “Reopen: Test Machine”
2. Display a popup for creating a New Turing machine with a text field for naming it and “Create” and “Cancel” buttons
3. Entered text appears in the text field
4. A popup saying the Machine was saved to “Test New 2.xml” will appear
5. Display a popup for defining the alphabet with a text field and “Define” and “Cancel” buttons
6. Entered text appears in the text field
7. Popup disappears and the build screen is enabled
8. Returns to the menu screen
9. A popup with a list of saved machines appears, “Test New 2” is on the list

Test:
Create a new machine - using a name that's already in use

Rationale:
The simulator should detect when the user is trying to name a turing machine the same thing as another machine.

Setup:
Pre-tests setup - 1 complete.
Start on the menu screen. (Launch the application or tap the Menu button on the build screen.)

Test Procedure:

1. Click "New Machine"
2. Enter "Test Machine" into the text field
3. Click the Create button
4. Click the Create button again
5. Add "01" to the text to make it "Test Machine01"
6. Click the Create button again
7. Click the "Ok" button
8. Click the "Define" button
9. Click the "Menu" button
10. Click "Load Machine"

Expected Results:

1. Display a popup for creating a New Turing machine with a text field for naming it and "Create" and "Cancel" buttons
2. Entered text appears in the text field
3. The popup remains in place and text appears under it saying "Machine already exists"
4. Nothing happens
5. Entered text appears in the text field
6. A popup saying the Machine was saved to "Test Machine01.xml" will appear
7. Display a popup for defining the alphabet with a text field and "Define" and "Cancel" buttons
8. Popup disappears and the build screen is enabled
9. Returns to the menu screen
10. A popup with a list of saved machines appears, "Test Machine01" is on the list

Test:
Create a new machine - cancel

Rationale:
The user should be to cancel creating a machine if they change their mind.

Setup:
Start on the menu screen.

Test Procedure:

1. Click “New Machine”
2. Enter “Test New 3” into the text field
3. Click “Cancel”
4. Click “Load Machine”

Expected Results:

1. Display a popup for creating a New Turing machine with a text field for naming it and “Create” and “Cancel” buttons
2. Entered text appears in the text field
3. The popup disappears returning the user to the menu screen
4. A popup with a list of saved machines appears, “Test New 3” is not on the list

User story: Set the initial contents of the tape

Test:
Set the initial contents of the tape - set from blank

Rationale:
Setting the initial contents of the type will be vital to creating useful Turing Machines.

Setup:
Follow the steps in the "Load Machine" test case

Test Procedure:

1. Click on the first cell in the tape (above the number 0)
2. Enter “0”
3. Click on the second cell in the tape
4. Enter “_”
5. Click on the seventh cell in the tape
6. Enter “2”
7. Enter “1”

Expected Results:

1. Typing cursor appears in the first cell, the keyboard appears and the tape is shifted upwards
2. First cell contains “0”, the keyboard disappears and the tape returns to the bottom of the screen
3. Typing cursor appears in the second cell, the keyboard appears and the tape is shifted upwards
4. Second cell is blank. (Blanks in the machine are displayed as empty.) The keyboard disappears and the tape returns to the bottom of the screen
5. Typing cursor appears in the seventh cell, the keyboard appears and the tape is shifted upwards
6. The number won’t appear as it is not in the alphabet
7. Seventh cell contains “1”, the keyboard disappears and the tape returns to the bottom of the screen

Test:
Set the initial contents of the tape - change contents

Rationale:
Changing the initial contents of the tape will be vital to creating useful Turing Machines.

Setup:
Follow the steps in the "Load Machine" test case.

Test Procedure:

1. Click on cell 0
2. Enter “1”
3. Click on cell 3
4. Enter “0”

Expected Results:

1. Typing cursor appears in the cell 0, the keyboard appears and the tape is shifted upwards. The contents of the current cell are selected.
2. Cell 0 contains “1”, the keyboard disappears and the tape returns to the bottom of the screen.
3. Typing cursor appears in the cell 3, the keyboard appears and the tape is shifted upwards. The contents of the current cell are selected.
4. Cell 3 contains “0”, the keyboard disappears and the tape returns to the bottom of the screen.

User story: Add a state

Test:
Add a state - new machine

Rationale:
Users need to be able to add states to build new Turing Machines.

Setup:
Follow the steps in “Create new machine - from application launch” use case.

Test Procedure:

1. Click on the Create State button in the toolbar
2. Click on a blank part of the canvas
3. Click on another blank part of the canvas

Expected Results:

1. The Create State button is highlighted
2. A new state appears where the user clicked with the name “0” and coloured red to indicate it is a start state
3. A new state appears where the user clicked with the name “1” and coloured white

Test:
Add a state - loaded machine

Rationale:
Users need to be able to add states to build new Turing Machines.

Setup:
Follow the steps in the "Load Machine" test case.

Test Procedure:

1. Click on the Create State button in the toolbar
2. Click on a blank part of the canvas
3. Click on another blank part of the canvas

Expected Results:

1. The Create State button is highlighted
2. A new state appears where the user clicked with the name “3” and coloured white
3. A new state appears where the user clicked with the name “4” and coloured white

Test:
Add a state - distance from another state

Rationale:
The machine should prevent the user from creating states that overlap to keep the machine clear and usable.

Setup:
Follow the steps in the "Load Machine" test case.

Test Procedure:

1. Click on the Create State button in the toolbar
2. Click on a blank part of the canvas
3. Click on the the new state
4. Click on the canvas directly next to the new state
5. Click on the canvas a distance greater than the diameter of a state from the new state

Expected Results:

1. The Create State button is highlighted
2. A new state appears where the user clicked with the name “3” and coloured red to indicate it is a start state
3. Nothing happens
4. Nothing happens
5. A new state appears where the user clicked with the name “4” and coloured white

Test:
Add a state - Rename a state

Rationale:
The user should be able to name states.

Setup:
Follow the steps in the "Load Machine" test case.

Test Procedure:

1. Click on the Create State button in the toolbar
2. Double click on state “0”
3. Enter “strtttt” into the text field
4. Click the “Set Name” button
5. Double click on state “1”
6. Clear the text field
7. Click the “Set Name” button
8. Enter “strt” into the text field
9. Click the “Set Name” button
10. Enter “End” into the text field
11. Click the “Set Name” button

Expected Results:

1. The Create State button is highlighted
2. A popup appears asking the user to enter a unique for the state, a text field and “Set Name” and “Cancel”
3. Only to letters “strt” appear in the text field (The simulator limits length of names to 4 characters to avoid overflowing the state.)
4. The popup disappears and the first state is now named “strt”
5. The State Name popup appears again
6. The text field is empty
7. Text appears under the text field saying “State name can not be blank”
8. The text field contains “strt”
9. Text appears under the text field saying “State name is not unique”
10. The text field contains “End”
11. The popup disappears and the state is now named “End”

User story: Add a transition

Test:
Add a transition - between two states

Rationale:
Users need to be able to add transitions to build new Turing Machines.

Setup:
Follow the steps in the "Load Machine" test case.

Test Procedure:

1. Click on the Create Transition button in the toolbar
2. Click and hold on state 0
3. Drag the mouse over to state 2 and release the mouse button
4. Select the first text field and enter “_”
5. Select the second text field and enter “5”
6. Enter “2” into the second text field
7. Select the third text field and enter “1”
8. Enter “R” into the third text field
9. Click the “Set Info” button

Expected Results:

1. The Create Transition button is highlighted
2. State 0 is outlined in yellow
3. A “Transition Details?” popup window opens with text fields for read, write and movement
4. The first text field now contains “_”
5. No text appears in the second text field, “5” is not in the alphabet definition for this machine
6. The second text field now contains “2”
7. No text appears in the third text field, “1” is not a valid direction
8. The third text field now contains “R”
9. A transition appears from state 0 to state 2 with the details “_/2/R”

Test:
Add a transition - from a state to itself

Rationale:
Users need to be able to add transitions from a state to itself to build many turing machines.

Setup:
Follow the steps in the "Load Machine" test case.

Test Procedure:

1. Click on the Create Transition button in the toolbar
2. Click on state 2
3. Select the first text field and enter “_”
4. Enter “2” into the second text field
5. Enter “R” into the third text field
6. Click the “Set Info” button

Expected Results:

1. The Create Transition button is highlighted
2. A “Transition Details?” popup window opens with text fields for read, write and movement
3. The first text field now contains “_”
4. The second text field now contains “2”
5. The third text field now contains “R”
6. A transition appears from state 1 to itself with the details “_/2/R”

Test:
Add a transition - reshape transition

Rationale:
Being able able to reshape transitions will be important to make complicated turing easier to understand.

Setup:
Follow the steps in the "Load Machine" test case.

Test Procedure:

1. Click on the Create Transition button in the toolbar
2. Click and hold on the green box for the transition from state 0 to itself with the details “1/1/R”. Drag it around to below the state
3. Click and hold on the green box for the transition from state 0 to 1 with the details “_/1/L”. Drag it up the canvas a short distance

Expected Results:

1. The Create Transition button is highlighted
2. The transition is rotated around so that it is below the state. This is important in machines that have multiple transitions leading from a single state to itself.
3. The transition will curve up in an arc. This is important when you have multiple transitions from one state to another, or a transition from one state to another and a returning transition from the second state to the first.

Test:
Add a transition - edit a transition

Rationale:
In case a user makes a mistake or wants to adjust a transition it is more convenient to be able to do so directly than having to delete and recreate transition.

Setup:
Follow the steps in the "Load Machine" test case.

Test Procedure:

1. Double tap the green square for the transition from state 0 to state 1
2. A “Transition Details?” popup window opens with text fields for read, write and movement
3. Change the text in the middle text field to “0”
4. Click the “Set Info” button

Expected Results:

1. The Create Transition button is highlighted
2. A “Transition Details?” popup window opens with text fields for read, write and movement
3. The middle text field contains “0”
4. The popup disappears and the transition now has the label “_/0/L”

User story: Set start state

Test:
Set the start state - new machine

Rationale:
The machine will automatically set the first state created as the start state

Setup:
Start on the menu screen.

Test Procedure:

1. Click “New Machine”
2. Enter “Start State Test”
3. Click the “Create” button
4. Click the “Ok” button
5. Click the “Define” button
6. Click on the Create State button
7. Click on the canvas
8. Click on a blank part of the canvas
9. Click the run button on the toolbar

Expected Results:

1. Display a popup for creating a New Turing machine with a text field for naming it and “Create” and “Cancel” buttons
2. Entered text appears in the text field
3. A popup saying the Machine was saved to “Test New.xml” will appear
4. Display a popup for defining the alphabet with a text field and “Define” and “Cancel” buttons
5. Popup disappears and the build screen is enabled
6. The Create State button is highlighted
7. A new state is created, coloured red to indicate it is the start state
8. A second state is created, coloured white to indicate it isn’t the start state
9. State 0 has the yellow outline marking it as the current state

Test:
Set the start state - existing machine

Rationale:
The user should be able to change the start state.

Setup:
Follow the steps in the “Load Machine” test case.

Test Procedure:

1. Click on the Set Start State button on the toolbar
2. Click on state 2
3. Click on state 1
4. Press the Run button on the toolbar

Expected Results:

1. The Set Start State button is highlighted
2. State 2 is coloured red to show it is the start state. State 0 is coloured white.
3. State 1 is coloured red and state 2 is reverted to white.
4. State 1 has the yellow outline marking it as the current state

User story: Set an accepting state

Test:
Set an accepting state - set a state to be accepting

Rationale:
The user must be able to set states as accepting states for the machine to halt with the answer “yes”.

Setup:
Follow the steps in the "Load Machine" test case

Test Procedure:

1. Click on the Set Accepting State button on the toolbar
2. Click on state 2

Expected Results:

1. The Set Start Accepting button is highlighted
2. State 2 has an extra circle inside it to show it is an accepting state

Test:
Set an accepting state - set an accepting state to not be accepting

Rationale:
The user must be able to change accepting states to normal states.

Setup:
Follow the steps in the "Load Machine" test case

Test Procedure:
1. Click on the Set Accepting State button on the toolbar
2. Click on state 1

Expected Results:
1. The Set Start Accepting button is highlighted
2. State 1 no longer has an extra circle inside it and is no longer an accepting state

User story: Run to completion

Test:
Run to completion - answer “no”

Rationale:
The turing machine simulator needs to be able to run turing machines to test them.

Setup:
Follow the steps in the "Load Machine" test case.

Test Procedure:

1. Click the Run button on the toolbar
2. Click the Run to Finish button on the Run toolbar

Expected Results:

1. The machine switches to Run Mode and displays the Run toolbar instead of the Build toolbar
2. A popup appears that says the machine halted with answer “no” in 9 steps

Test:
Run to completion - answer “yes”

Rationale:
The turing machine simulator needs to be able to run turing machines to test them.

Setup:
Follow the steps in the "Load Machine" test case.

Test Procedure:

1. Click the Set Final State button on the Build toolbar
2. Click on State 2
3. Click the Run button on the toolbar
4. Click the Run to Finish button on the toolbar

Expected Results:

1. The Set Final State button is highlighted
2. State 2 is set as a final state and has an inner circle
3. The machine switches to Run mode
4. A popup appears that says the machine halted with answer “yes” in 9 steps

User Stories (Part 2)

User story: Save machine

Test:
Save machine

Rationale:
For the turing machine simulator to be of long term use it will have to be able to save and load turing machines.

Setup:
Follow the steps in the “Create new machine - from application launch” Test Case but name the machine “Test Save”

Test Procedure:

1. Click on the Create State button on the toolbar
2. Click on the canvas to create a state
3. Click on another part of the canvas to create a second state
4. Click the Create Transition button on the toolbar
5. Click on state 0 and drag to state 1
6. Fill in the text fields with 0/1/R
7. Click the “Set Info” button
8. Click the Save button in the toolbar
9. Click Ok
10. Close the Application
11. To test that the Save worked, follow the “Load Machine” Test case but replace the filename “Test Machine” with “Test Save”

Expected Results:

1. The Create State button is highlighted
2. A state is added where the user clicked
3. A second state is added where the user clicked
4. The Create Transition button is highlighted
5. A “Transition Details?” popup window opens with text fields for read, write and movement
6. The text fields are filled in with the user’s input
7. The popup closes and the transition is added

8. The machine is saved, and a popup appears to tell the user it has saved and the filename

User story: Load machine

Test:
Load Machine

Rationale:
Users will need to be able to load saved turing machines.

Setup:
Start on the menu screen.

Test Procedure:

1. Launch the application
2. Click Load Machine
3. Click “Test Machine”

Expected Results:

1. Display a menu screen with the options: “New Machine”, “Load Machine”, “Import Machine” and “Delete Machine”
2. Open a list of saved turing machines, with their filename and size
3. The saved Turing Machine “Test Machine” is loaded and the Build screen is opened

Test:
Load Machine - while another is open

Rationale:
Users will need to be able to load saved turing machines.

Setup:
Follow the steps in the “Load machine” test case.

Test Procedure:

1. Click the Menu button
2. Click Load Machine
3. Click “Test Machine”

Expected Results:

1. Menu screen is loaded
2. Open a list of saved turing machines, with their filename and size
3. The saved Turing Machine “Test Machine” is loaded and the Build screen is opened

Test:
Load Machine - re-open

Rationale:
Users will need to be able to load saved turing machines.

Setup:
Follow the steps in the “Load machine” test case.

Test Procedure:

1. Click the Menu button
2. Click “Reopen: Test Machine” in the top right of the screen

Expected Results:

1. Menu screen is loaded
2. The saved Turing Machine “Test Machine” is loaded and the Build screen is opened

User story: Step through execution

Test:
Step through execution - step by step

Rationale:
Often it is very useful to be able to step through a Turing Machine to see how it functions.

Setup:
Follow the steps in the "Load Machine" test case

Test Procedure:

1. Click the Run button in the build toolbar
2. Click the Step button in the toolbar
3. Move the slider in the top right to the right
4. Click the step button in the toolbar
5. Repeat step 2 until the machine halts

Expected Results:

1. The simulator switches to the Run mode and toolbar
2. The simulator highlights the current state by surrounding it with a yellow circle, highlights the transition in yellow as it changes state, moves the window to show the path of the transition, updates the values in the tape at the bottom, highlights the current tape cell in yellow and scrolls the tape so the current cell is in the centre.
3. The slider adjusts the speed of execution for the machine
4. As in step 2, but the movement is faster
5. When the user clicks the Step button while the Turing Machine is on state 2 it will halt and a popup will appear to inform the user that the simulation halted with the answer ‘No’

Test:
Step through execution - reset

Rationale:
The user should be able to reset the machine, during or at the end of its execution.

Setup:
Follow the steps in the "Load Machine" test case

Test Procedure:

1. Click the Run button in the build toolbar
2. Click the Run button in the run toolbar
3. Click the Pause button before the machine halts
4. Click the Reset button

Expected Results:

1. The simulator switches to the Run mode and toolbar
2. The simulator executes the machine as in the step by step case but automatically continues as it completes each step.
3. The simulator pauses when it completes the current step
4. The simulator resets the tape and resets the current state to the start state

Test:
Step through execution - run step by step

Rationale:
The user can press the run button to automatically step through the execution at a speed determined by the slider.

Setup:
Follow the steps in the "Load Machine" test case

Test Procedure:

1. Click the Run button in the build toolbar
2. Click the Run button in the run toolbar
3. Click the Pause button
4. Drag the slider halfway across to the right
5. Click the Run button
6. Click the Reset button
7. Drag the slider all the way to the right
8. Hit the run button
9. Hit the “Ok” button

Expected Results:

1. The simulator switches to the Run mode and toolbar
2. The simulator executes the machine as in the step by step case but automatically continues as it completes each step.
3. The simulator pauses when it completes the current step
4. Slider moves
5. The simulator executes the machine as in step 2, but significantly faster
6. The machine halts and resets the tape and resets the current state to the start state
7. Slider moves
8. The simulator rapidly steps through the execution of the machine. When it completes a popup appears telling the user the answer.
9. The popup closes.

Test:

Step through execution - Step after the machine finishes

Rationale:

The simulator should handle the user trying to continue after the machine has halted

Setup:

Follow the steps in the “Run to completion - answer no: test case”

Test Procedure:

1. Click the Step Forward button
2. Click the “Ok” button
3. Click the Run button
4. Click the “Ok” button

Expected Results:

1. A popup appears telling the user the answer (no) and the number of steps (9).
2. The popup closes
3. A popup appears telling the user the answer (no) and the number of steps (9).
4. The popup closes

User story: Step backwards through execution

Test:
Step backwards through execution - mid execution

Rationale:
Being able to step backwards through the execution will be very useful for testing machines.

Setup:
Follow the steps in the “Load Machine” test case.

Test Procedure:

1. Click the Run button in the build toolbar
2. Step 5 steps forward into the execution using the Step forward button
3. Click the Step Backward button
4. Click the Step Backward button again

Expected Results:

1. The simulator switches to the Run mode and toolbar
2. The simulator highlights the current state by surrounding it with a yellow circle, highlights the transition in yellow as it changes state, moves the window to show the path of the transition, updates the values in the tape at the bottom, highlights the current tape cell in yellow and scrolls the tape so the current cell is in the centre.
3. The simulator behaves the same as in step 2 but in reverse
4. The simulator behaves the same as in step 2 but in reverse

Test:
Step backwards through execution - at the start

Rationale:
The simulator should prevent the user from stepping backwards past the beginning of the execution.

Setup:
Follow the steps in the “Load Machine” test case.

Test Procedure:

1. Click the Run button in the build toolbar
2. Click the Step Forward button
3. Click the Step Backward button

Expected Results:

1. The simulator switches to the Run mode and toolbar, the Step Backwards button is greyed out
2. The simulator highlights the current state by surrounding it with a yellow circle, highlights the transition in yellow as it changes state, moves the window to show the path of the transition, updates the values in the tape at the bottom, highlights the current tape cell in yellow and scrolls the tape so the current cell is in the centre.
3. The simulator behaves the same as in step 2 but in reverse, the Step Backwards button is greyed out

User story: Deploy on Pendo Pad

Test:
Deploy on Pendo Pad

Rationale:
The application needs to be deployed on a Pendo Pad to

Setup:

1. Run the virtual machine
2. Copy the KivyTuring folder in it’s entirety into the file system
3. Plug in the Pendo Pad
4. Open a terminal
5. Navigate to the KivyTuring folder
6. Open the buildozer.spec file
7. Change the path to be the specific path to the KivyTuring folder
8. Enter “buildozer android debug deploy run” and wait
9. When the prompt appears on the pendo pad to allow USB Debugging click Ok

Test Procedure:

1. Follow the instructions in the Pre-tests setup
2. Close the application
3. Reopen the application and follow the instructions in the “Load Machine” test case

Expected Results:

1. A machine will be created for testing purposes
2. ..
3. The same machine that was created in step 1 will be loaded into the simulator

User story: Undo & Redo

Test:
Undo & Redo - create state

Rationale:
The user should be able to undo and redo changes.

Setup:
Follow the steps in the “Load Machine” test case

Test Procedure:

1. Click on the Create State button in the toolbar
2. Click on a blank part of the canvas
3. Click the Undo button
4. Click the Redo button

Expected Results:

1. The Create State button is highlighted
2. A new state appears where the user clicked
3. The state disappears
4. The state reappears

Test:
Undo & Redo - rename state

Rationale:
The user should be able to undo and redo changes.

Setup:
Follow the steps in the “Load Machine” test case

Test Procedure:

1. Click on the Create State button in the toolbar
2. Double click on the first state
3. Enter “strt” into the text field
4. Click the “Set Name” button
5. Click the Undo button
6. Click the Redo button

Expected Results:

1. The Create State button is highlighted
2. A popup appears asking the user to enter a unique for the state, a text field and “Set Name” and “Cancel”
3. The text field contains the text
4. The popup disappears and the first state is now named “strt”
5. The state’s name changes back to 0
6. The state’s name changes to “strt” again

Test:
Undo & Redo - move state

Rationale:
The user should be able to undo and redo changes.

Setup:
Follow the steps in the “Load Machine” test case

Test Procedure:

1. Click on the Create State button in the toolbar
2. Click on state 0 and drag it up a short distance
3. Click the Undo button
4. Click the Redo button

Expected Results:

1. The Create State button is highlighted
2. The state moves up
3. The state moves back
4. The state moves up again

Test:
Undo & Redo - create transition

Rationale:
The user should be able to undo and redo changes.

Setup:
Follow the steps in the “Load Machine” test case

Test Procedure:

1. Click on the Create Transition button in the toolbar
2. Click on state 2
3. Fill in the details “0/0/R”
4. Click the “Set Info” button
5. Click the Undo button
6. Click the Redo button

Expected Results:

1. The Create Transition button is highlighted
2. A “Transition Details?” popup window opens with text fields for read, write and movement
3. Text fields are filled out with the user’s input
4. The popup disappears and the transition is added
5. The transition disappears
6. The transition reappears

Test:
Undo & Redo - edit transition

Rationale:
The user should be able to undo and redo changes.

Setup:
Follow the steps in the “Load Machine” test case

Test Procedure:

1. Click on the Create Transition button in the toolbar
2. Double click on the green box for the transition between states 0 and 1
3. Fill in the details “0/1/R”
4. Click the “Set Info” button
5. Click the Undo button
6. Click the Redo button

Expected Results:

1. The Create Transition button is highlighted
2. A “Transition Details?” popup window opens with text fields for read, write and movement
3. Text fields are filled out with the user’s input
4. The popup disappears and the transition is added
5. The transition label changes back
6. The transition label changes again

Test:
Undo & Redo - reshape transition

Rationale:
The user should be able to undo and redo changes.

Setup:
Follow the steps in the “Load Machine” test case

Test Procedure:

1. Click on the Create Transition button in the toolbar
2. Click and hold on the green box for the transition from state 0 to 1 with the details “_/1/L”. Drag it up the canvas a short distance
3. Click the Undo button
4. Click the Redo button

Expected Results:

1. The Create Transition button is highlighted
2. The transition will curve up in an arc.
3. The transition straightens
4. The transition curves down again

Test:
Undo & Redo - Edit tape cell

Rationale:
The user should be able to undo and redo changes.

Setup:
Follow the steps in the “Load Machine” test case

Test Procedure:

1. Click the first tape cell
2. Enter “0”
3. Click the Undo button
4. Click the Redo button

Expected Results:

1. Typing cursor appears in the first cell, the keyboard appears and the tape is shifted upwards
2. First cell contains “1”, the keyboard disappears and the tape returns to the bottom of the screen
3. The cell contains “0”
4. The cell contains “1”

User story: Build interface

To test the build interface run the test cases:

- Add a state test cases
- Add a transition test cases
- Set start state test cases
- Set an accepting state test cases
- Save machine test cases
- Undo & redo test cases

Additional test cases:

Test:
Delete State

Rationale:
The user should be able to delete unwanted states.

Setup:
Follow the steps in the “Load Machine” test case

Test Procedure:

1. Click the Delete button
2. Click on state 2

Expected Results:

1. The Delete button is highlighted
2. State 2 and the transition attached to it are removed

Test:
Delete Transition

Rationale:
The user should be able to delete unwanted transitions.

Setup:
Follow the steps in the “Load Machine” test case

Test Procedure:

1. Click the Delete button
2. Click on the transition from state 1 to state 2

Expected Results:

1. The Delete button is highlighted
2. The transition is removed

Test:
Define alphabet

Rationale:
The user should be able to change the alphabet.

Setup:
Follow the steps in the “Load Machine” test case

Test Procedure:

1. Click the Define Alphabet button
2. Add a “2” to the text field
3. Click the “Define” button

Expected Results:

1. An “Alphabet Definition” popup opens with text, a text field, and Define and Cancel buttons
2. The text field contains “_012”
3. The popup closes and the alphabet changes

User story: Run interface

To test the run interface run the test cases:

- Run to completion test cases
- Step through execution test cases
- step backwards through execution test cases

Pre-tests setup - 1

Summary:

-Create a turing machine called “Test Machine” for use in many of the test cases. The completed machine should have the same states and transitions as the machine depicted in the “Test Machine example” image in the docs directory.

Procedure:

1. Click “New Machine”
2. Enter the name “Test Machine”
3. Click “Create”
4. Click “Define”
5. Click the Create State button
6. Create 3 states (Click the screen 3 times in different places.)
7. Click the Create Transition button
8. Click on state 0
9. Enter 1,1 & R into the text boxes
10. Click “Set Info”
11. Click on state 0
12. Enter 0,1 & R into the text boxes
13. Click “Set Info”
14. Click the green box above one of the transitions and drag it around to the side to rotate the transition so that both are clearly visible
15. Click on state 0 and drag to state 1, then release
16. Enter _, 1 & L into the text boxes
17. Click “Set Info”
18. Click on state 1 and drag to state 2, then release
19. Enter 1, 1 & L into the text boxes
20. Click “Set Info”
21. Click the Set Accepting State button
22. Click on state 1
23. Click on cell 0 in the tape and enter 0
24. Click on cell 1 in the tape and enter 0
25. Click on cell 2 in the tape and enter 0
26. Click on cell 3 in the tape and enter 1
27. Click on cell 4 in the tape and enter 0

28. Click on cell 5 in the tape and enter 1
29. Click on cell 6 in the tape and enter 1
30. Click the Save button