codeSaver.js

saveButton.onClick(): This function gathers all the code boxes from the file and saves them in JSON format locally with the filename of the user’s choice.

loadButton.onClick(): This function grabs the local file, validates it, and recreates the flow chart to a similar state that it was saved in.

codeboxBackend.js

addBox(type): This function creates a new codeBox. A codeBox is a div that contains a textarea that can be dragged around, a textarea that is a child of the div, a type (f, s, i, c, or e), its primary connection, its secondary connection, an array of indices that hold all the boxes that have connections to this box, a x coordinate, a y coordinate, and a string that keeps the code of the box inside after saving. It also keeps a z index of the boxes.

‘f’ means ‘first,’ or starting block. It leads to the primary connection box. There should always be exactly one of these on screen.

‘s’ means a simple box that leads to its primary connection box.

‘i’ means ‘input.’ It prompts the user to enter an expression that is assigned to the variable that it contains. It leads to the primary connection box.

‘c’ means ‘conditional.’ It leads to its primary connection box if its condition is true. Else, it leads to its secondary connection box.

‘e’ means ‘end.’ The program halts when these boxes are reached. The code inside them is executed before this happens.

removeBox(index, override): The index is the id of the box that the user is deleting. The method deletes it and removes all other connections that the box had to ther boxes in the diagram.

primaryConnect(startIndex, endIndex): This function creates a primary connection between two boxes. The startIndex is the first box and the endIndex is the second box that the primary connection is being created between.

secondaryConnect(startIndex, endIndex): This function creates a secondary connection between two boxes. The startIndex is the first box and the endIndex is the second box that the secondary connection is being created between.

primaryUnconnectStart(startIndex): This function removes a primary connection from the box with the startIndex. The endIndex is already known.

secondaryUnconnectStart(startIndex): This function removes a secondary connection from the box with the startIndex. The endIndex is already known.

codeMaker(): This function creates code out of the boxes using the code inside their text areas and their connections.

Draggability.js

canvasResizer(): Resizes the canvas.

addConnectorNodes(box): Adds connector nodes to the box that is passed through as a parameter.

makeDraggable(): Makes a box draggable.

deleteBox(e): Deletes the box that is passed in as a parameter.

orientAnchoredLine(myLine): Orients a line when a box that is connected to another box is dragged around the screen.

loadLine(myLine, origin, target): Loads the lines when a file is loaded from local storage.

orientLine(myLine, x1, y1, x2, y2): Creates a line that begins at x1, y1, and ends at x2, y2.

destroyAnchoredLine(myLine): Destroys the line that is passed into the function.

anchorLine(e): Connects the line that is being created between two boxes.

startDrawing(myConnector): When a connector is clicked, this function creates a line.

clickOnConnector(e): Listens to a click on the connector button.

clickOnBox(e): Listens to a click on a code box.

startDragging(box): Starts dragging a box when the user wants to drag a box around a screen.

mouseUp(e): Listens to the mouse lifting.

boxLineOrient(box): When a box is being dragged around, all the lines that the box is connected to are oriented accordingly.

moveStuff(e): Deals with all the events that should happen when the mouse is moved.

newmain.js

runner.onClick(): Runs the file when the run button is clicked.

stopper.onClick(): Stops the the file when the stop button is clicked.

newworker.js

write(word): prints the parameter ‘word’ to the console.

writeln(word): print the parameter ‘word’ to the console on a new line.

clear(): Cleans the console.

sleep(milliseconds): sleeps the program for the number of milliseconds passed in as a parameter.

$.closer(): Sends a message to close the worker. The worker is the script that is running the code that makes the flow chart work.

$.inputPrompter(word, next): Asks the user to enter an input which will be the value of the variable word.

$.inputEvaluator(): Evaluates the code in the box.

$.complexEvaluator(words): Evaluates the code in the box depending on the boxes type and position in the flow chart.

$.runner(next, code): Starts running the program starting with id “next.”

$.creator(rawFuntions): Creates the array of functions that the program uses to run the flow chart.

This.onmessage: A function that handles the connection of the worker with the main program.