codeSaver.js

saveButton.onClick(): This function saves the project locally with the filename of the user's choice. using JSON.stringify().

loadButton.onClick(): This function loads the local save file, processes it, and creates a flowchart identical to the saved one.

codeboxBackend.js

addBox(type): This function creates a new codeBox. A codeBox is a div that can be dragged around. As its children, it has a textarea, a receiver connector, a primary connector, and optionally a secondary connector. There are 4 types of codeBoxes ('f', 's', 'i', 'c').

'f' means 'first,' it is the box that functions as the beginning point of the flowchart. It leads to the box that is connected to its primary connector. There should always be exactly one of these in a project.

's' means 'simple.' These boxes lead to the boxes that are connected to their primary connectors.

'i' means 'input.' These boxes prompt the user to enter an expression that is assigned to the variable names they contain. They lead to the boxes that are connected to their primary connectors.

'c' means 'conditional.' If the boolean expression they contain is true, they lead to the boxes that are connected to their primary connectors. Otherwise, they lead to the boxes that are connected to their secondary connectors.

removeBox(index, override): The index is the id of the box that the user is deleting. The method deletes it and removes all the connections that the box had with other boxes in the flowchart. The override is necessary to delete first boxes, which the user can't delete.

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

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

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

codeMaker(): This function processes codeBoxArr, which is an array that contains all the necessary information about the project. It creates an array of code that is easy to interpret and run.

draggability.js

canvasResizer(): Resizes the project canvas, where flowcharts are created.

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

makeDraggable(box, savedDepth, depth): Makes a box draggable, and assigns an appropriate z-index to it. Then it calls addConnectorNodes(box);

deleteBox(e): if the user is dragging a box and presses backspace, that box is deleted.

orientAnchoredLine(myLine): Orients a line when one of the boxes that the line is connected to is moved.

loadLine(myLine, origin, target): Given two connectors (origin, target), creates a line that connects origin to target.

orientLine(myLine, x1, y1, x2, y2): Makes a line start at point (x1, y1) and end at point (x2, y2).

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

anchorLine(e): When the user drags a line to make a connection between two connectors, this function is called. The line is “anchored.”

startDrawing(myConnector): When a connector is clicked, this function creates a line that will have the mouse pointer as one of its endpoints. Such a line is not anchored, and be deleted if the mouse is released unless it can be anchored.

clickOnConnector(e): When a connector is clicked, this function is called.

clickOnBox(e): When a box is clicked, this function is called.

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

mouseUp(e): Listens to the mouseup event.

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

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

newmain.js

runner.onClick(): Runs the flowchart when the run button is clicked. It does so by spawning web workers, which have their own threads. They can communicate with the main program using a messaging system.

stopper.onClick(): If there is a running flowcharts, this kills it by terminating the worker.

worker.onmessage: This is the function that handles communication between the main program and the worker.

newworker.js

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

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

clear(): Cleans the console.

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

$: This is a variable that has a lot of properties important for the script. It is used to minimize the number of variable names that should not be used in the flowchart.

$.closer(): Sends a message to have the worker terminated.

$.inputPrompter(word, next): Makes a prompt() dialog box appear, where the user can enter input that will be the value of the variable word.

$.inputEvaluator(): Evaluates the code in a box using eval().

$.complexEvaluator(words): Evaluates the code in a box depending on the code type of the box. It is responsible for making sure that the next box that will be evaluated is selected correctly. There are 4 types of code types (‘s’, ‘c’, ‘i’, ‘e’).
‘s’ means ‘simple.’ This type belongs to boxes of type **‘s’** or **‘f’**, assuming that they are not terminal points.
‘c’ means ‘conditional.’ This type belongs to boxes of type **‘c’**.
‘i’ means “input.’ This type belongs to boxes of type **‘i’**, assuming that they are not terminal points.
‘e’ means ‘end.’ This type belongs to all boxes of type **‘s’**, **‘f’** or **‘i’**, assuming that they are terminal points. The code inside them is run, and then the worker is terminated.

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

$.creator(rawFunctions): Creates the array of functions that the program uses to run the flowchart.

This.onmessage: This is the function that handles communication between the main program and the worker.