שם הפרוייקט

S-Emulator

מגישים

|  |  |  |
| --- | --- | --- |
| מייל ליצירת קשר | תז | שם |
| noadinabr@gmail.com | 211821525 | נועה דינבר |
| Luzon.shir@gmail.com | 208299057 | שיר לוזון |

בונוסים (אם מומשו)

|  |  |
| --- | --- |
| תיאור | שם |
| הוספנו 2 סקינים נוספים- יש אפשרות בחירה בצד ימין למעלה. | 1 |
| הוספנו אנימציות- בטבלת היסטוריה על כל שורה שנוספת, ובטבלת הפקודות בשינוי דרגה. | 2 |
| מימשנו stepback בדיבאג. | 3 |
| מימשנו breakpoint בדיבאג- על מנת להשתמש: צריך להתחיל ריצה כרגיל על לחיצת כפתור start ובחירה במצב debug כששורה כלשהי בטבלה מסומנת בעמודה השמאלית ביותר (לוחצים על עמודה זו בשורה המבוקשת ויופיע עיגול אדום). **בהמשך תיאור מפורט** | 4 |

# S-Emulator — Assignment 2 (JavaFX UI)

**Submitters:**

Shir Luzon — 208299057 — luzon.shir@gmail.com  
Noa Dinbar — 211821525 — noadinbar@gmail.com **GitHub repository:** https://github.com/noadinbar/S-emulator.git

## Overview

S-Emulator is split into modules with a strict Engine ↔ UI separation. The engine is passive and exposes only APIs that return deep DTOs. The UI (ui-fx) is responsible for user interaction, formatting, and presentation.

* engine - core model, validation, execution, expansion; depends on dto only.
* dto - deep data transfer objects used by both engine and UI; the UI never touches engine internals.
* ui-fx - JavaFX application: controllers, views (FXML), CSS files, and formatting.

The Assignment-1 console UI is still present in the repo for reference, but Assignment-2 focuses on the JavaFX front-end.

## Main UI (ui-fx) structure

* MainApp - loads program\_scene.fxml (size from FXML), ESC closes window.
* Header - top-right controls: Program/Function selector, Degree controls (− / + and direct input), Animations combo, Skin combo, Help (light-bulb) button, Highlight filter.
* Program table (center-left) - instructions with Line column shown.
* Ancestry/Chain table (bottom-left) - shows expansion origin chain for the selected instruction (Line), reversed order.
* Inputs / Outputs panes (center-right) – display and manage variables input and results.
* History (bottom-right) - past runs and debug sessions, supports Rerun and Show.

## Core flows & behaviors

### Load XML

* Until a file is loaded, all actions are disabled (greyed out).
* On following loads, the file chooser remembers the last directory used.
* The file chooser shows only \*.xml files.
* A valid load replaces the previous program, an invalid load leaves the previous program intact (with an error message).

### Program / Function selector

* Header combo lets you switch between the program and any function in the program.
* Switching selection refreshes the program table and resets the chain table appropriately.

### Degree selection

* Buttons: Expand (+1) / Collapse (−1).
* Direct entry: edit the degree text box and press Enter to apply. You can only put integar between 0-max degree.

### Execute

* Provide inputs and run at the selected degree.
* Outputs shows all variables in order: y → xᵢ → zᵢ, plus total cycles.
* Execute button is enabled only after clicking Start/Rerun button.
* Every successful run (on run mode) is appended to History with: index, degree, inputs, final y, cycles.
* Rerun: loads degree + inputs back.
* Show: opens a popup with the state of all variables at the end of the run/debug.

### Debug mode

* Step Over / Resume / Stop controls.
* While a debug session is active, Inputs are read-only. They are re-enabled when the session finishes or is stopped.
* Changed variables are highlighted after each step/resume tick + the relevant instruction in the table is selected.
* On any termination (stop, resume, reached to the end, selecting a different display in the program/function selector in the middle of a debug), the session is recorded in History automatically.

### Highlight filter

Choose EXIT, a label (L#), or a specific variable (y, xᵢ, zᵢ) to highlight all related instructions in the instruction table.

* The selection resets when changing degrees.

### Help (light-bulb)

A help button (light-bulb icon) opens a short in-app guide explaining the UI and the main flows.

## Notable UI details

* Inputs pane scrolls when many rows are present, first field gets focus when starting a new run or rerun.
* Instruction table shows Line, Chain table hides Line and shows the reversed origin chain for the selected row.
* Window content stays centered and scrollable when resizing.

## Build & Run

**Artifacts**: JARs - ui-fx.jar, engine.jar, dto.jar (packaged in the distribution).

**Run**: Double‑click run.bat.

**Bonuses (explanation)**

### Animations

A top-right combo box toggles UI animations on/off.

### Skins

A top-right combo box switches the app skin (CSS theme) at runtime.  
The default skin is plain and presented automatically.

**Breakpoint**  
If you choose to put a breakpoint on a line that eventually you won’t get to it, the program will just run execute as usual, and you can see that **a row in the history table has been added, also a message will be shown on the screen.**

**Stepback**  
Stepback only works on debug mode. So if you debug with a breakpoint, stepback will work only from the breakpoint and over.  
(We referred to each of the bonuses individually 😊)