

Report 1 Documentation

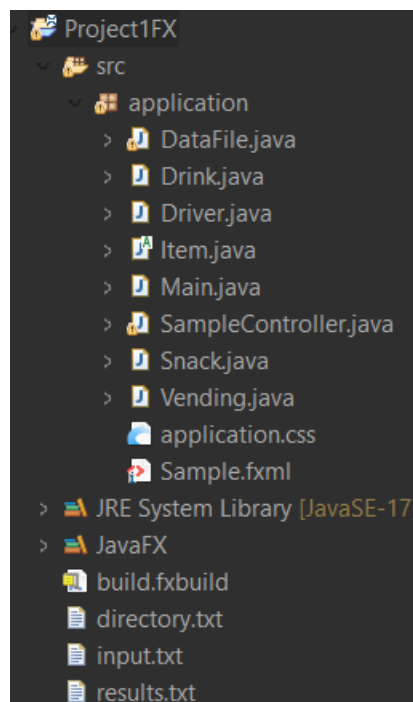
Michael Hager

CSCE 314 – 598

12/12/2022

How to run

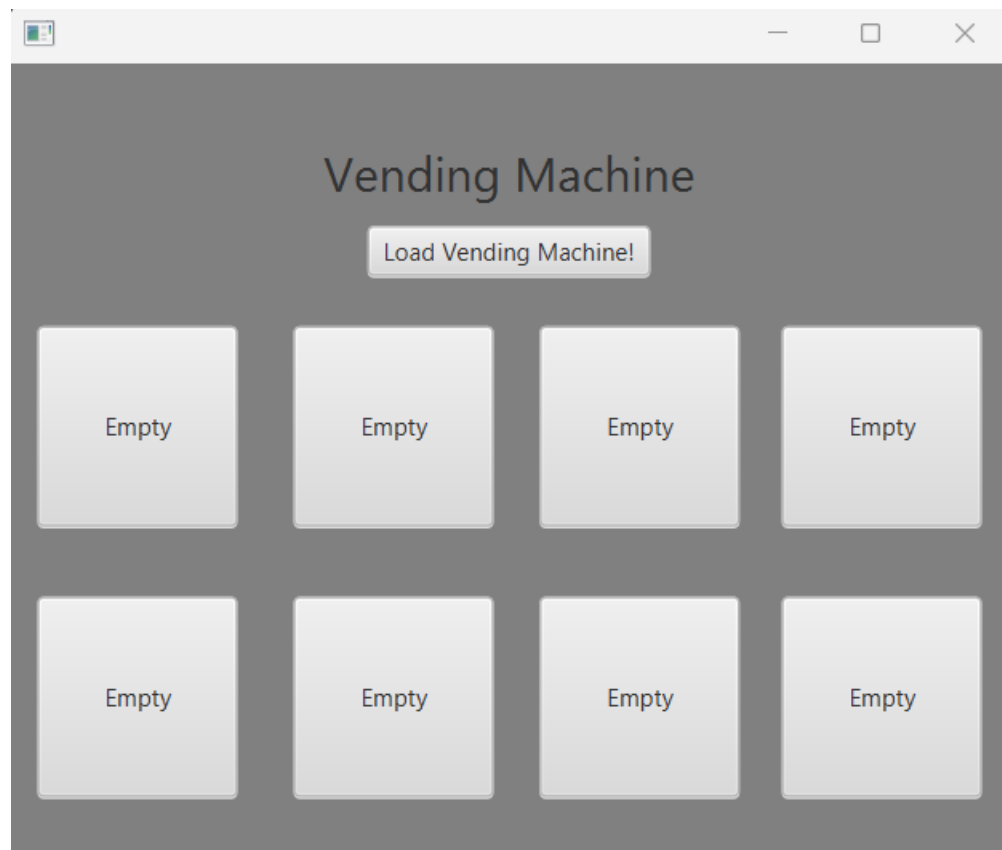
After unzipping all the files from the zipped folder “P1-code”, a JavaFX project should be created in eclipse. After doing so, we should delete “module-info” files if they are auto-created. After setting up the arguments properly in the run configurations section, we can now attempt to compile and run the code. First, let’s copy all the code from the zipped folder into the “application” tab which is inside the source folder. When done, it should look something like this.



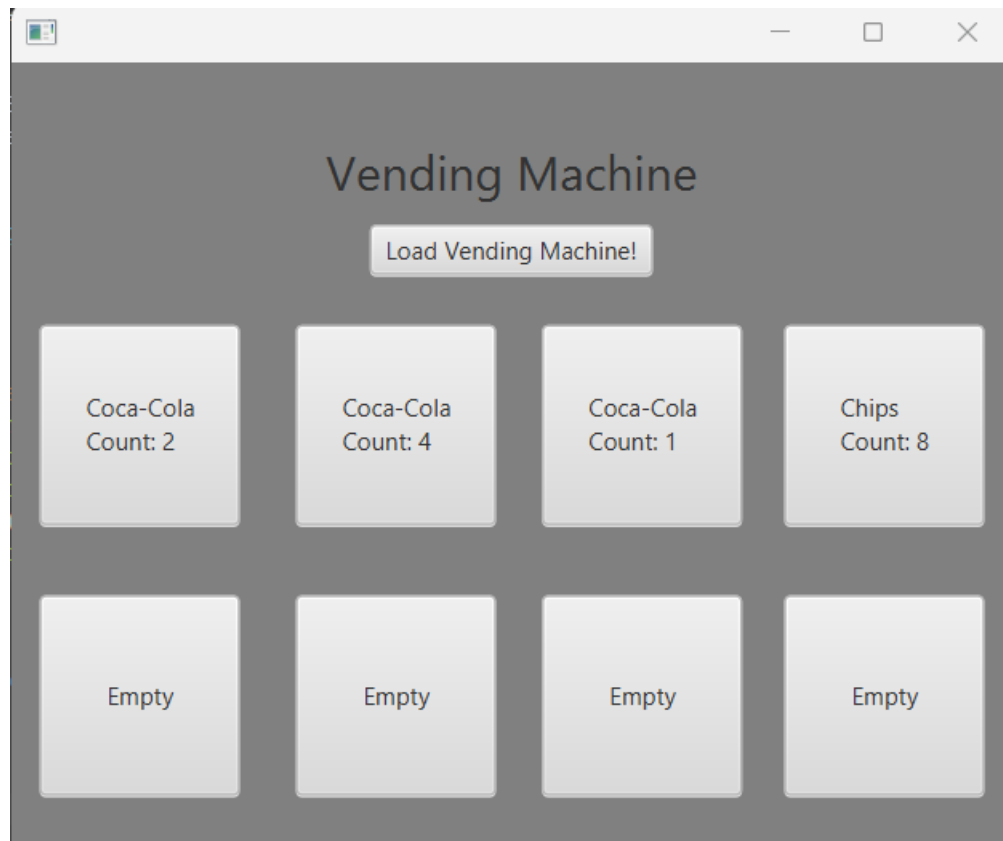
After doing so, we can run the “Driver.java” in the console by hitting the run button. In the console, there should be a one-line print. All the tests and prints are put into another file called “results.txt”, which should be generated after running the driver. This file shows the pretty output of all the original testing, as well as some extra test cases made for extra credit. To run the JavaFX part of the project, you can go to either “SampleController.java” or “Main.java” and click the run button. This should result in a GUI pop-up of a vending machine with eight empty slots. Once you click on the button that says, “Load Vending Machine!”, this will load the vending machine based on the “directory.txt”, which is the text file used to fill the vending machine.

NOTE: Running in the console takes a “directory.txt” and “input.txt” to run and print results, while the JavaFX vending machine just reads a “directory.txt” file because the input would be from the tester or user clicking the buttons on the vending machine.

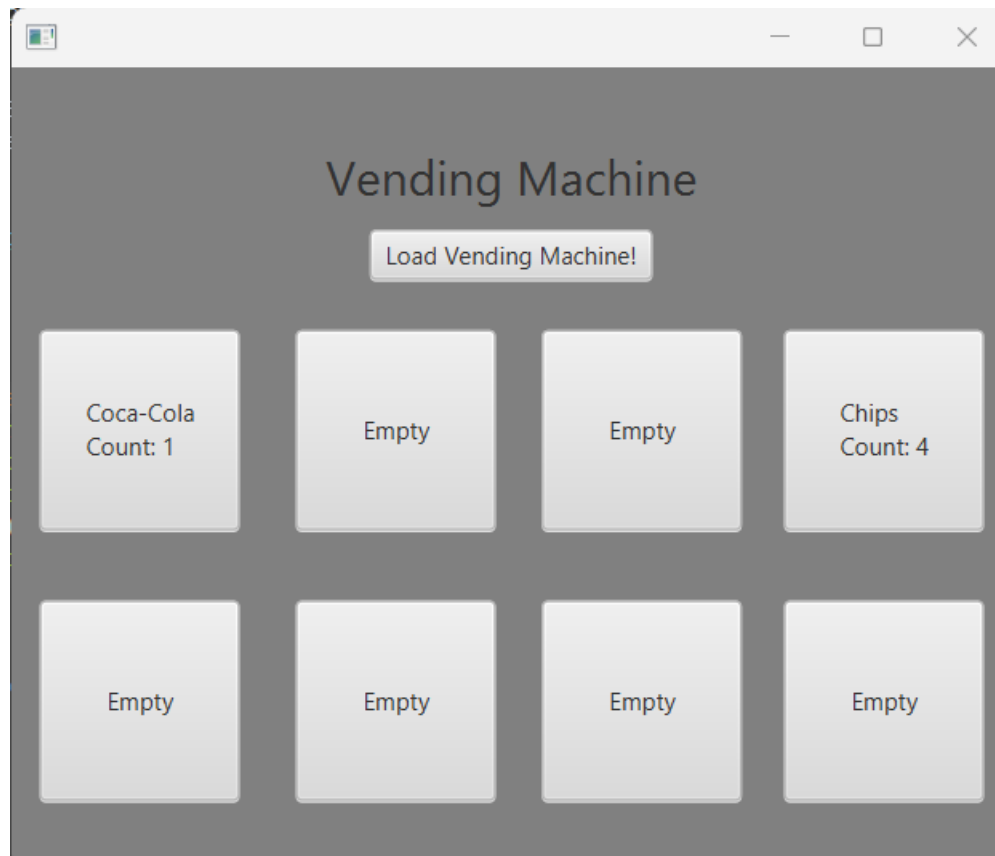
How it looks running in Main.java



After running the "Main.java", this GUI will come up. The first step to using the GUI is to click the button "Load Vending Machine!".



After clicking the load button, the vending machine is now loaded with whatever data is provided in the "directory.txt". From here, the user can click on any of the buttons until the counts reach 0, then they are empty again. To restart, you can always just push the loading button again to start over.



Maintaining Selection Protocol

```
Items originally there:
Coca-Cola: (Drink): 2
Coca-Cola: (Drink): 4
Coca-Cola: (Drink): 1
Chips: (Snack): 8

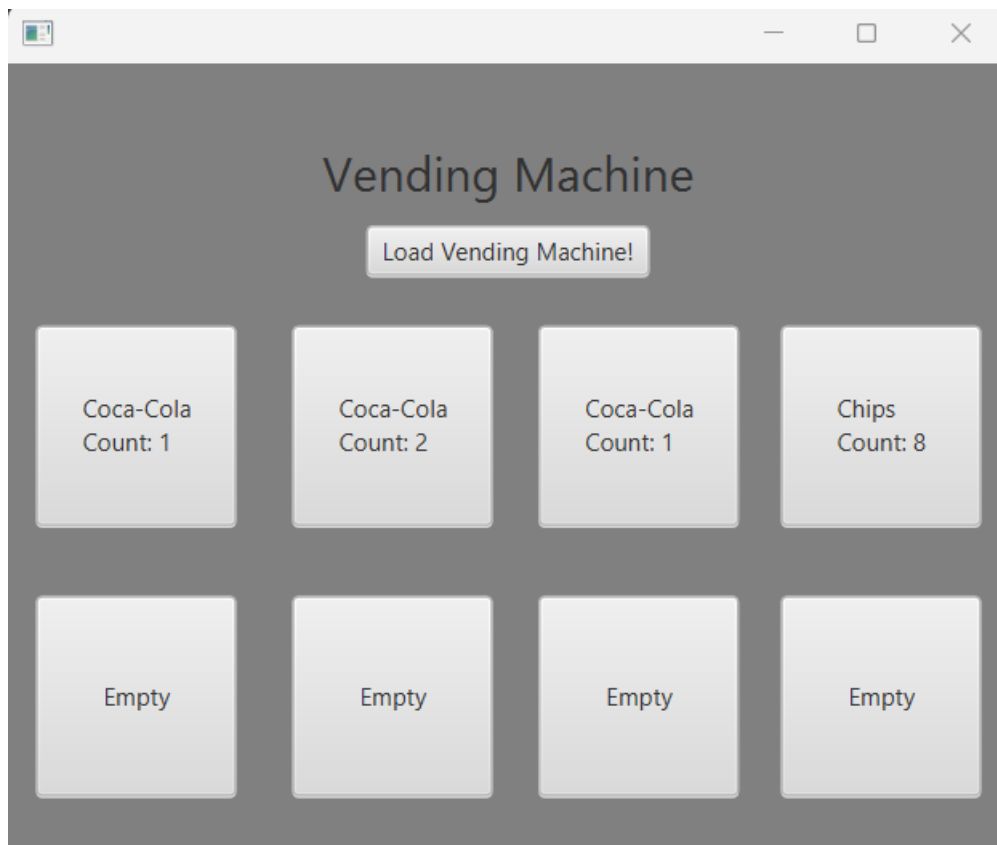
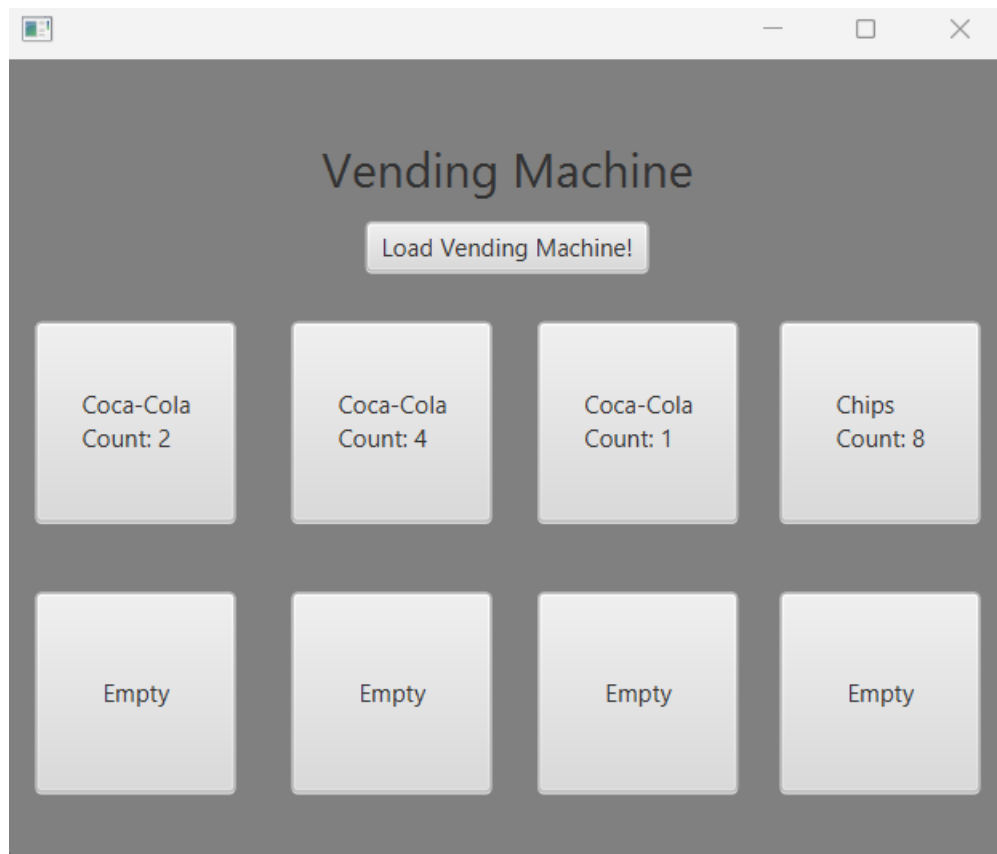
Items removed from list:
Coca-Cola: (Drink): 1
Coca-Cola: (Drink): 2
Coca-Cola: (Drink): 1
Chips: (Snack): 8

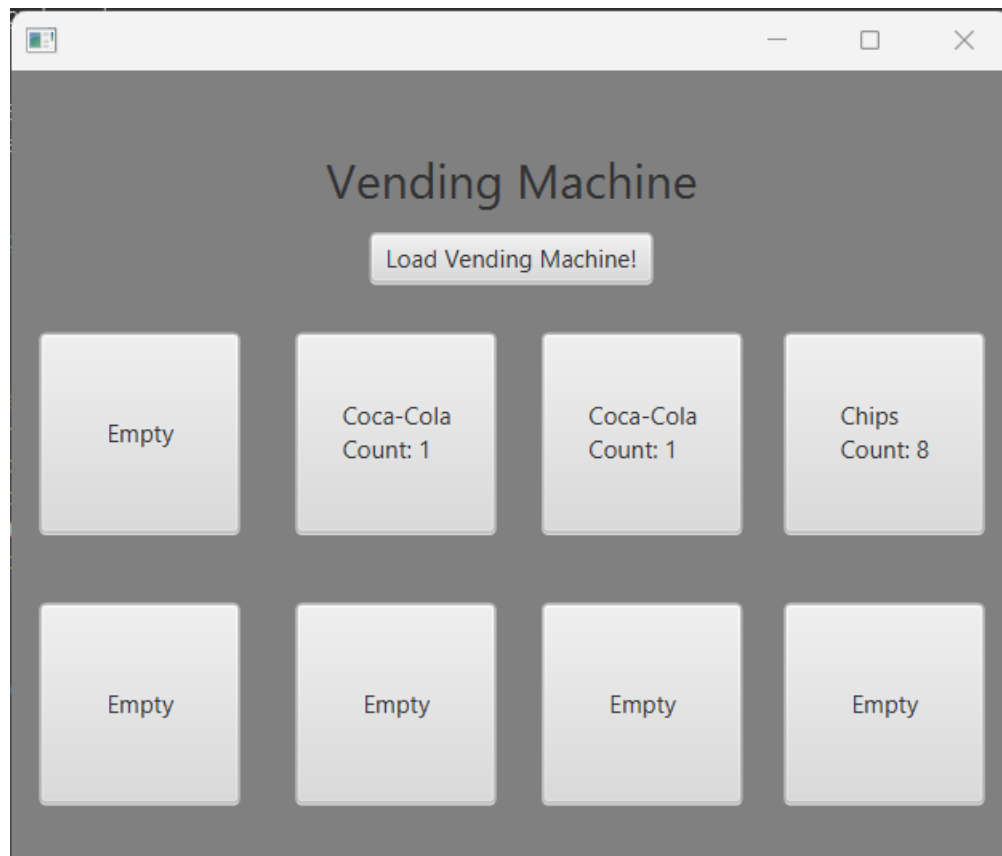
Items added by item:
Coca-Cola: (Drink): 1
Coca-Cola: (Drink): 2
Coca-Cola: (Drink): 1
Chips: (Snack): 9

Items added by index:
Coca-Cola: (Drink): 1
Coca-Cola: (Drink): 2
Coca-Cola: (Drink): 1
Chips: (Snack): 10

Items removed from row until empty:
Coca-Cola: (Drink): 1
Coca-Cola: (Drink): 1
Coca-Cola: (Drink): 1
Chips: (Snack): 10
Coca-Cola: (Drink/Snack): 0
Coca-Cola: (Drink): 1
Coca-Cola: (Drink): 1
Chips: (Snack): 10
Coca-Cola: (Drink/Snack): 0
Coca-Cola: (Drink): 1
Coca-Cola: (Drink): 1
Chips: (Snack): 10
```

This is the “results.txt” file created after running Driver.java. To see the results in the console instead, uncomment all the code above the try-catch and comment on all the try-catch code.





Here is a similar style of testing for the GUI created by JavaFX. The only negative thing about my GUI is that there currently is not any implementation that allows adding to the vending machine unless done so in the “directory.txt” file.