Casey Pei
CSCE 314-598

Documentation - Vending Machine

A basic vending machine Java Project with accompanying GUI (implemented with JavaFX and Scene Builder) for the Fall 2022 Semester of CSCE-314 at Texas A&M University.

Sections of this documentation;

1. How to run
2. How it looks running
3. Maintaining selection protocol

This project was made by Casey Pei

- (https://github.com/peicasey/)
- caseypei@tamu.edu

Documentation - How to run

Basic explanation of how to run the vending machine project. This will be most helpful for the graders of the project, as it assumes you have access to the canvas page.

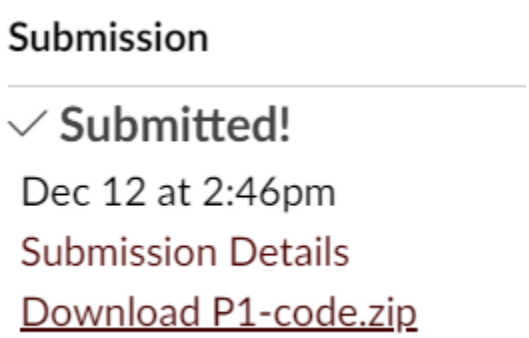
Requirements:

- Eclipse
  - any IDE could work; however this documentation will be specific to Eclipse users
- Java SE-11 or above
  - Windows: download (https://ninite.com/ selecting Eclipse and JDK x64 11)
  - MAC: download (https://www.eclipse.org/downloads/packages/)
- JavaFX SDK 19 or above
  - instructions on how to install (https://www.youtube.com/watch?v=bk28ytggz7E)

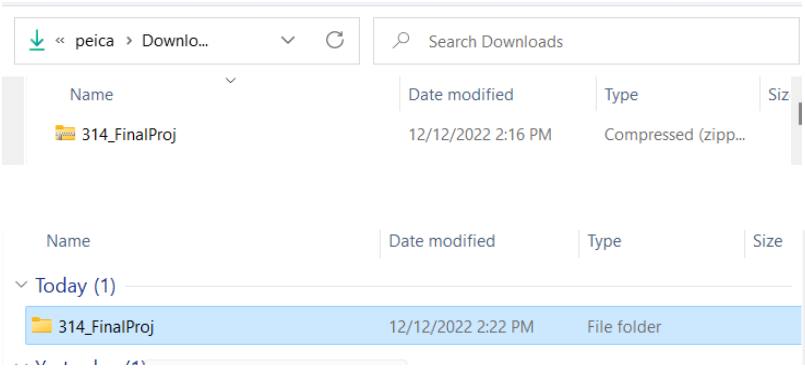
Casey Pei
CSCE 314-598

Steps:

1. Download .zip file submission from Canvas. (Show as *P1-code.zip* here but will be seen as *314_FinalProj.zip* for the rest of the documentation)

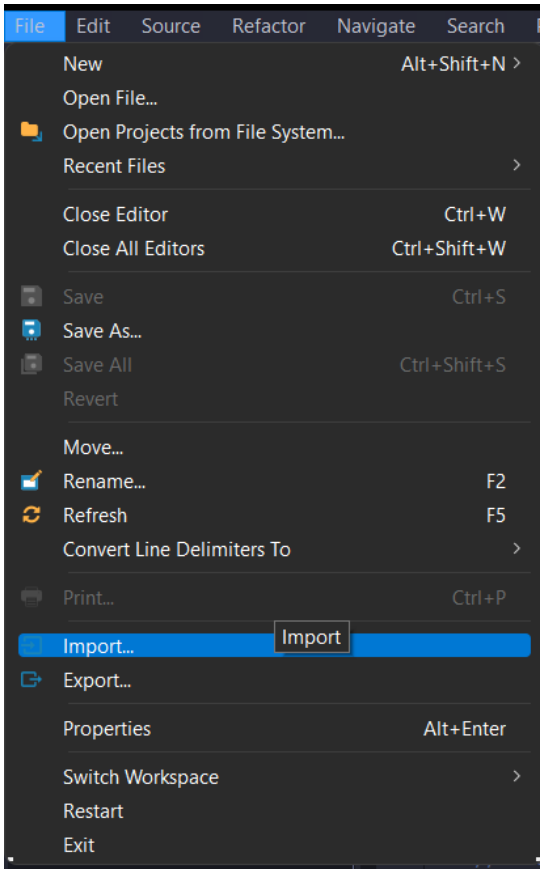


2. Locate the downloaded .zip file and unzip it.

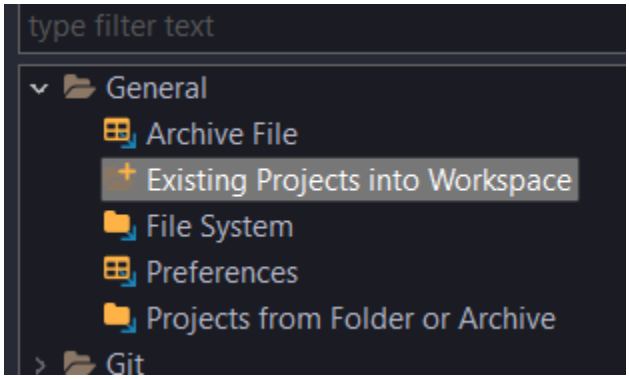


3. Open Eclipse. It may be helpful to open it in a new workspace in case the project has the same name as any of the previous students' submissions (the project is called 'JavaFX_vending') -- however if you are to start in a new workspace be careful as that may require re-adding JavaFX as a User Library (which is gone over in this tutorial: https://www.youtube.com/watch?v=bk28ytgqz7E).

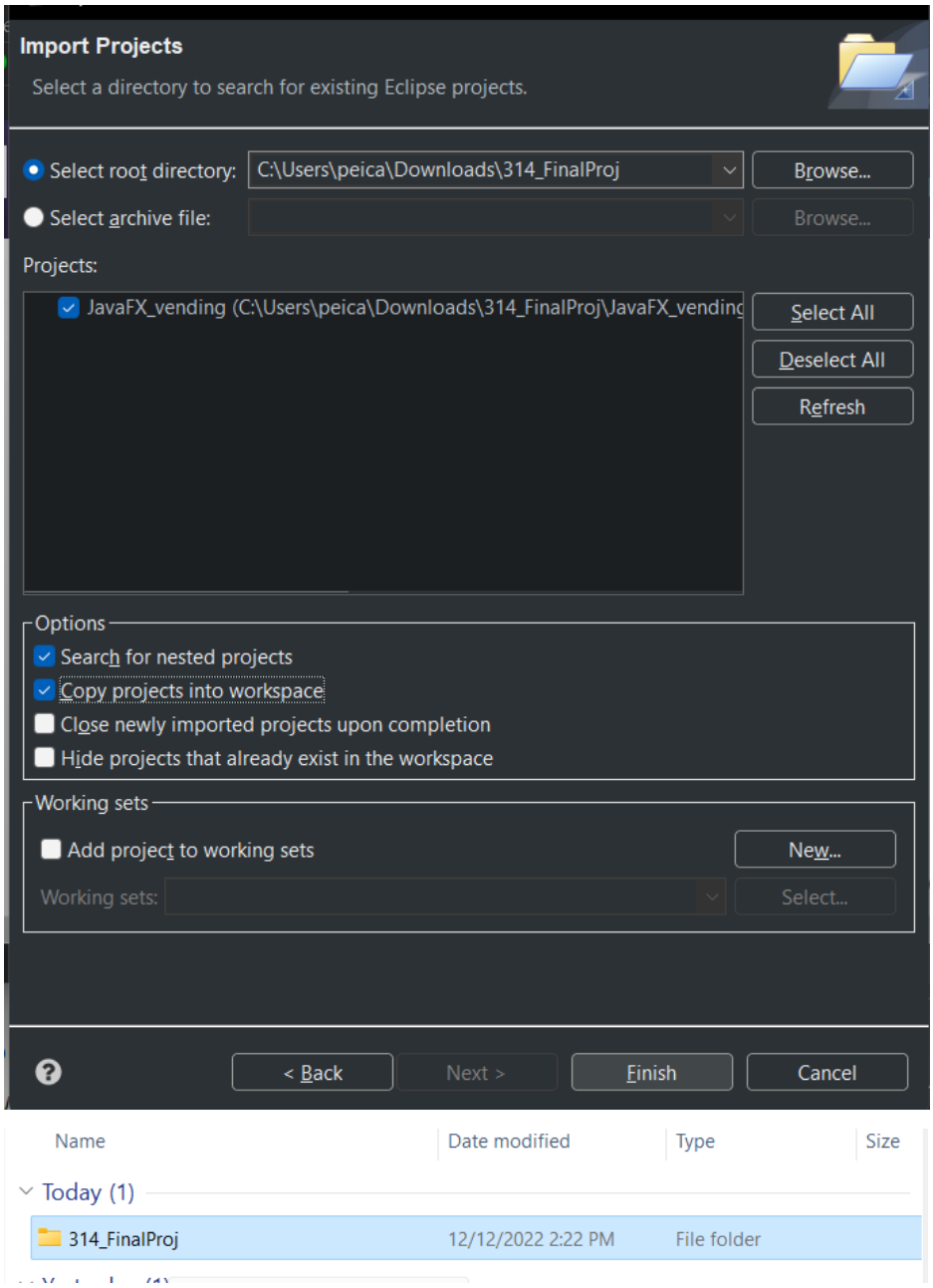
4. Click ‘Files’ → ‘Import’ in the top left corner.



5. In the subsequent pop-up, select ‘General’ → ‘Existing Projects into Workspace’ then click ‘Next’ in the bottom right corner.

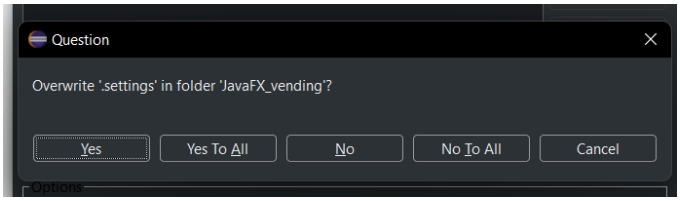


6. In the following screen, select the ‘Select root directory option’ which will bring up a pop-up of the file explorer, where you should navigate to where you have extracted the .zip file. Click ‘Open’ to select it.

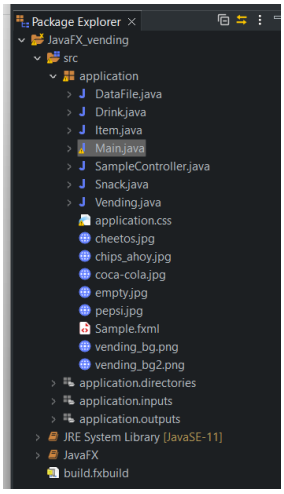


Check the boxes for ‘Search for nested projects’ and ‘Copy projects into workspace.’

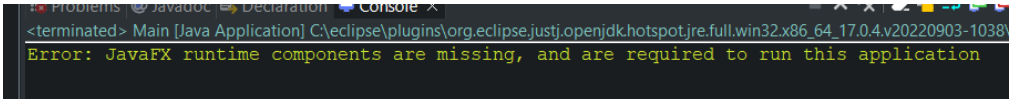
7. If you receive a prompt to overwrite settings, click ‘Yes.’



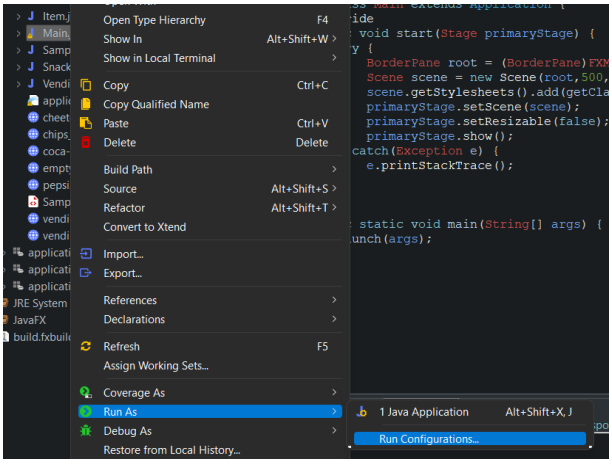
8. Select the ‘Main.java’ file and click the Green Run button. ( ).



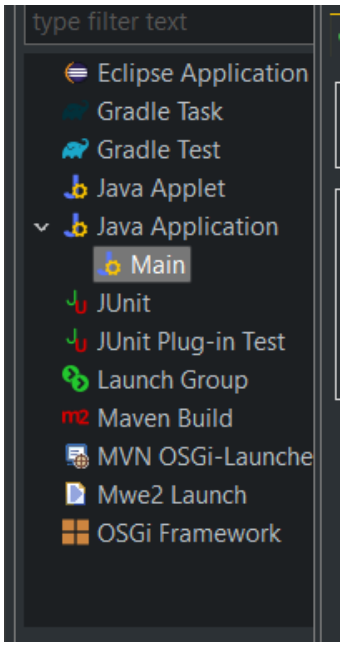
9. Receive this error message:



10. After receiving an error message, then right-click ‘Main.java’ and navigate through ‘Run As’ → ‘Run Configurations…’



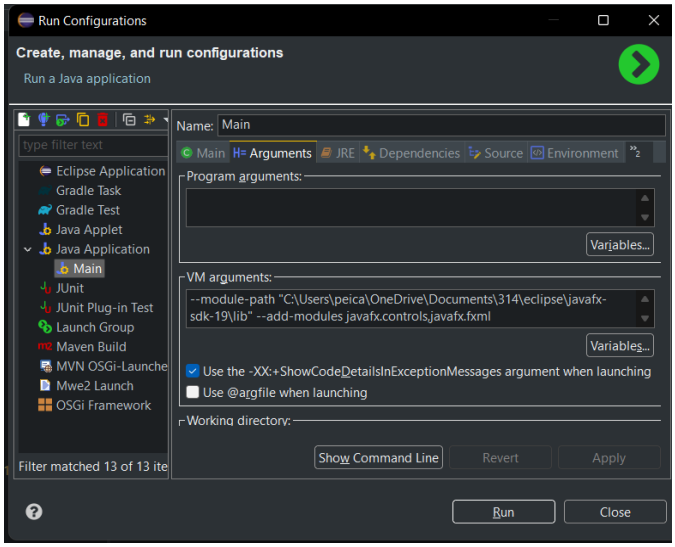
11. In the following pop-up, select the bottom-most ‘Main’ underneath the ‘Java Application’ tab. (In this image there was only one ‘Main’) from the left side-panel.



12. Then in the main area, select the ‘Arguments’ tab, and under the header ‘VM arguments:’ copy-paste the following text:

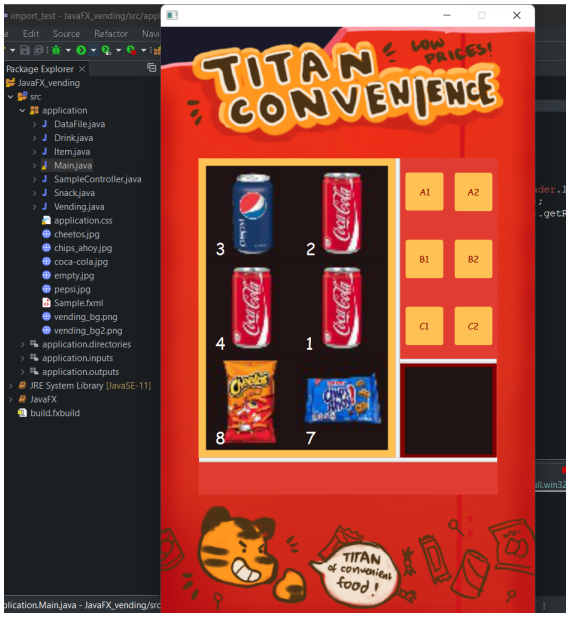
```
--module-path "YOUR\PATH\lib" --add-modules
javafx.controls,javafx.fxml
```

However, for the blue portion of text (aka. YOUR\PATH\lib), replace that with your own path to your javafx\lib folder. Example:



13. Then click ‘Apply’ in the bottom right corner.

14. Then re-click the run button ( ). This time the project GUI should display.



Documentation - How it looks running

This will show some screenshots of the GUI after it has received some output from the user.

NOTE: Some of the screenshots in this section and in the following section feature the machine before the ‘Reset’ button was added. However, the functionality displayed remains largely the same.

Casey Pei
CSCE 314-598

Start-up

GUI at start-up; shows amount of item in each slot, buttons corresponding to choosing each item.*

Clicking Button

Displaying the result after the user clicks 'A1.' The GUI decrements the amount of drink in the respective slot, and shows it 'dispensed.'

Casey Pei
CSCE 314-598

Empty Slots

GUI after all of one item has been dispensed; it now has 0 items in the slot and is empty with its picture. The last dispense item is still in the bottom right.

Cannot dispense

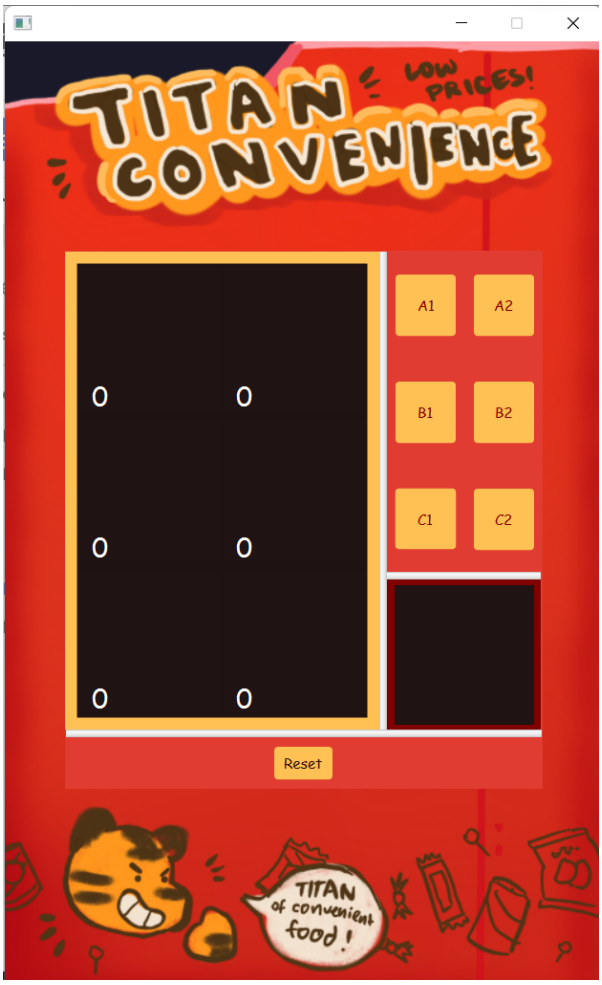
After dispensing all of the items in C2, clicking it again replaces the dispensed item picture with nothing as well.

* NOTE: Changing the directory.txt file will automatically change the number of items in each slot, as well as what the items themselves are called by the program. However, they do not automatically update the images or add more slots than 6.

The non-GUI version of the program is able to do this however. The user can access this functionality by commenting out the JavaFX portion of the main function and uncommenting out the vanilla Java portion.

Casey Pei
CSCE 314-598

All Items Dispensed



After clicking on all the buttons, the vending machine is entirely empty. All of the numbers in each slot are 0, all have no item, and there is nothing left to be dispensed.

Reset Machine

After clicking the 'Reset' button, the machine has all of its items returned and the output slot it empty.

Casey Pei
CSCE 314-598

Documentation - Maintaining Selection Protocol

This portion will show both the original non-GUI program and GUI program follow the vending machine selection protocol.

The logic is as expected:

1. Given an inputted slot from the user, the vending machine will dispense whatever item was in that slot -- however it will first dispense from:
    a. a same-item slot with the most items
    b. then from the lowest-index one of those same-item slots with the most items (if there are multiple same-item slots with the most items)

GUI Program Showing Expected Behavior:

GUI	Items per Slot	Button Clicked	How it fits behavior
	3 2 4 1 8 7	---	Start.

	3 | 2 3 | 1 8 | 7	A2	Removes it from the Coca-Cola slot with the most items (B1).
	3 | 2 2 | 1 8 | 7	B2	Removes it from the Coca-Cola slot with the most items (B1)

	3 | 1 2 | 1 8 | 7	B1	Since there was a tie for the Coca-Cola slot with the most items, it removed it from the one with the smallest index (A2)
	3 | 1 1 | 1 8 | 7	B1	It removed it from the Coca-Cola slot with the most items (B1).

Casey Pei
CSCE 314-598

	<table><tr><td>3</td><td>0</td></tr><tr><td>1</td><td>1</td></tr><tr><td>8</td><td>7</td></tr></table>	A2	All the Coca-Cola slots were tied for having the most items (1) so A2 was removed because it had the smallest index.
	<table><tr><td>3</td><td>0</td></tr><tr><td>1</td><td>1</td></tr><tr><td>7</td><td>7</td></tr></table>	C1	Removed the Cheetos from slot C1 because it was the only slot that had the same item, so it had the most.

Casey Pei
CSCE 314-598

Non-GUI Program Showing Expected Behavior:

Given the inputs:

directory.txt	inputs.txt

`Drink, Coca-Cola, 120, 16, soda, 2`
`Drink, Coca-Cola, 120, 16, soda, 4`
`Drink, Coca-Cola, 120, 16, soda, 1`
`Snack, Chips, 220, 4, false, 8`
	`0`
	`2`
	`1`