

I used Co-Pilot to help get Maven working on vscode as well as to give me an idea of where to start on the assignment. The suggestions were very useful as I went from having virtually no progress with an application that wouldn't start, to having a successful level 1 project in ~20 minutes of troubleshooting and running commands through powershell.

Some of the suggestions were misleading due to duplicate code on classes such as `RectangleShape`, where I thought it'd be ok to remove the code before the AI got around to reading it. This messed up a lot of things but eventually it bounced back and finished the job itself. At one point the project had a lot of red errors and the suggestions weren't making much sense, but once it had reread the files it fixed a lot of its own issues and ran more commands. I accepted virtually every suggestion the AI made since it knew the code better than I did at that point. Once the AI had calmed down I made sure to know what I was looking at and thus I could make my own revisions if necessary. I started with revising a name and added some `JavaDoc`.

My AI prompt was,

"Build a Drawing App that allows users to:
Click to draw rectangles and circles on a canvas.
Choose shape type (e.g., `RadioButtons` in a `ToggleGroup` or a `ComboBox`).
Clear the canvas with a `Button`.
Main window shown in a `JavaFX Stage` with controls laid out cleanly.
Use the following `JavaFX` tools
`Stage`, `Scene`, and layout panes (e.g., `BorderPane`, `HBox`, `VBox`).
`Canvas` + `GraphicsContext` for drawing (or `JavaFX Shape` nodes if you prefer — pick one approach and stay consistent).
Event handling via `setOnMouseClicked`, `setOnAction`, etc.
`ObservableList` for tracking drawn shapes (recommended).
Design practices
Use at least 3 classes (e.g., `MainApp/App`, `DrawingController`, `DrawableShape` + concrete shape classes).
Proper access modifiers, constructors, and clear method separation."

And the returned code that followed was my entire Level 1 project, as well as installing files such as `Scoop`, fixing Maven to run `JavaFX`, and 5 files. It also added the maven wrapper so that there are multiple ways to start the file, whichever one is easiest.

When I went to tackle the Level 2 challenge with the prompt,

"Real-time updates using `JavaFX` bindings (e.g., bind a label to `shapes.sizeProperty()` or a computed `DoubleProperty` for total area).
Tooltips or status text for "current selected shape." "

It satisfied the requirements such as adding the tooltip. I was going to change the tooltip but soon realized that it was linked to the `ListView` and it actually fit quite nicely with relaying information to the user.

For level 3 my prompt was,

"Keyboard input: e.g., press `C` to clear, `R` to switch to `Rectangle`, `O` (or `C`) to switch to `Circle`.
Custom shape subclasses (e.g., `RectShape`, `CircleShape`) implementing a shared `DrawableShape` interface.
`JavaDoc`-style comments for all public classes and methods."

I made edits such as putting the keybinds on the buttons for the shapes and clear button so that you knew what to press without needing to see the code and going back to an earlier bug and fixing it so the canvas always started out as white instead of becoming so when you cleared it for the first time in addition to adding or standardizing the JavaDoc.

When I realized that I had used AI to write most of the assignment, however, I decided to recode the entire thing by hand. Since I was low on AI prompts and out of completions for my free trial until after Sept. 21st (the due date), I decided to use my original assignment as a huge inspiration (i.e. formatting and what packages to import) for my current version since there I was unable to get assistance from co-pilot and I had zero assistance past that as my only tool. The recode, while very similar, was done by me and I understand most of what I had written, there were a total of 3 copy-pastes (2 of which were from the recode to cut time, 1 of which for complicated string formatting). I also wrote the JavaDoc and README.txt by hand and troubleshooted the entire project by myself, making sure I had everything I needed for it to run. It took me ~4-5 hours but it is done and it works great if I do say so myself.