Code Visualization

Milestone 4

Progress Matrix

Task	Completion %	Curtice	Joshua	Catherine
1. PyQt main window	100%	50%	50%	0%
2. Implement code editor	100%	50%	50%	0%
3. Custom List/Map implementations	100%	0%	0%	100%
4. Modify Traceprinter compile-time options	100%	100%	0%	0%

1. PyQt Main Window

Curtice Gough
Joshua Hartzfeld

GUI Progress

- Main window has a 3 pane layout with primary focus on the text editor.

- The Text editor pane has 2 additional buttons “Execute” and “Step”,
  - The Execute button sends the current contents of the text box to traceprinter for compilation
  - The Step button sends a signal to trace printer to step through the program

2. Implement Code Editor

Curtice Gough

PyQt6 Widgets

- QVBoxLayout
- QPlainTextEdit
- QHBoxLayout
- QPushButton

```
20          # Code pane setup
21          code_execute_button = QPushButton("Execute")
22          code_execute_button.released.connect(self.code_execute)
23
24          code_pane_buttons = QWidget()
25          code_pane_buttons_layout = QHBoxLayout(code_pane_buttons)
26          code_pane_buttons_layout.addWidget(code_execute_button)
27          code_pane_buttons_layout.addWidget(QPushButton("Step"))
28
29          self.code_edit = QPlainTextEdit()
30          code_font = QFont("Monospace", 12)
31          self.code_edit.setFont(code_font)
32          self.code_edit.setTabStopDistance(40)
33
34          code_pane = QWidget()
35          code_pane_layout = QVBoxLayout(code_pane)
36          code_pane_layout.addWidget(self.code_edit)
37          code_pane_layout.addWidget(code_pane_buttons)
```

Linking Functions to Buttons

- main.py imports trace.py
- On button press: send code to traceprinter

```
59      def code_execute(self):
60          usercode = self.code_edit.toPlainText()
61          self.tracer.execute(usercode)
62          print(self.tracer.trace_dict)
```

- Still trying to fix decorative text effects
- Non-functional code not shown
    - Can be viewed on dev branch

GUI Demo

3. Custom List/Map Implementations

Catherine DiResta

Custom Classes

- Implemented custom classes for List and Map
- Coded test programs for the custom classes

Todo

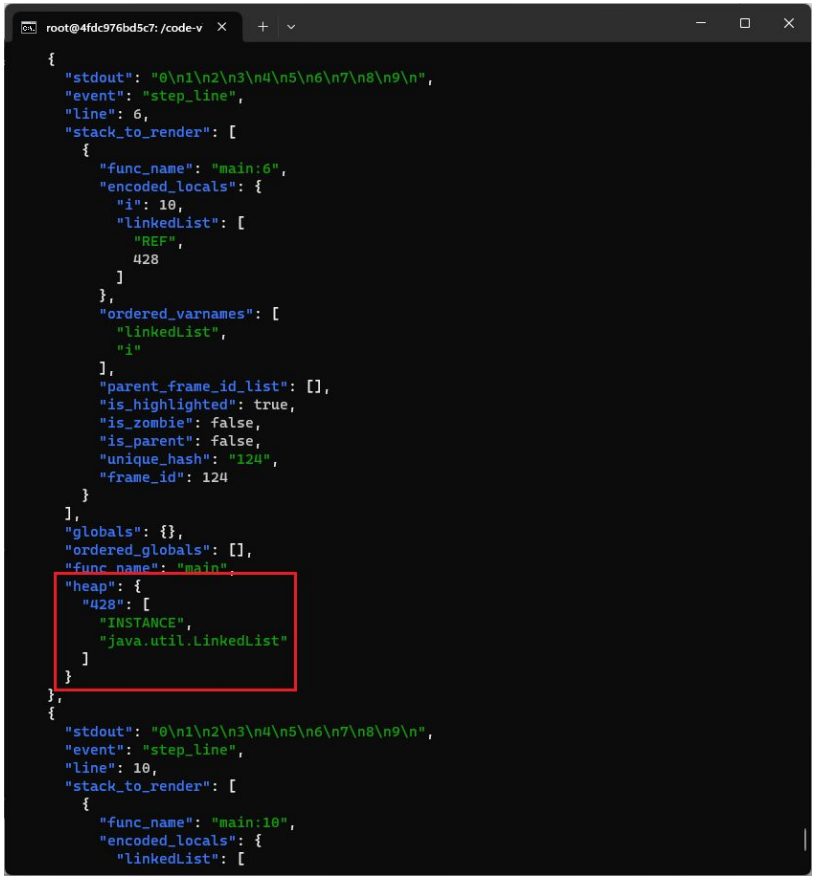
- Finish creating the custom classes for all supported data structures

4. Modify Traceprinter Compile-Time Options

Curtice Gough

Before

- Arbitrary classes are marked as “INSTANCE”
- Actual field values are not shown



Modified Traceprinter Code

- Iterate over every file in cp/codeviz
- Add file to source table
- Add usercode to source table
- Compile everything

```
        String[] cpFiles;
        try { // List all files in cp/codeviz
            cpFiles = Stream.of(new File("codeviz").listFiles())
                .filter(file -> !file.isDirectory())
                .map(File::getName)
                .collect(Collectors.toSet())
                .toArray(new String[0]);
        }
        catch(NullPointerException e) {
            System.err.println(System.getProperty("user.dir") + "codeviz directory contains no valid .java fi
les");
            e.printStackTrace();
            cpFiles = new String[0];
        }

        String[][] fileinfo = new String[cpFiles.length + 1][2];

        for (int i = 0; i < cpFiles.length; i++) { // Stage all java files for compilation
            String className = cpFiles[i].substring(0, cpFiles[i].indexOf('.')); // Remove ".java"
            fileinfo[i][0] = className;
            fileinfo[i][1] = getFileContents("codeviz/" + cpFiles[i]);
        }
        fileinfo[fileinfo.length - 1][0] = mainClass;
        fileinfo[fileinfo.length - 1][1] = usercode;

        bytecode = c2b.compileFiles(fileinfo); // Compile everything
```

Test class

- Person.java used for testing
- Each instance has three private member variables

File: **traceprinter_backend/cp/codeviz/Person.java**

```java
public class Person {
    private String name;
    private int age;
    private double height;

    public Person(String name, int age, double height) {
        this.name = name;
        this.age = age;
        this.height = height;
    }
}
```

After

- Arbitrary classes are marked as “INSTANCE”
- Actual field values are shown correctly

Milestone 5

Task Matrix

Task	Curtice	Josh	Catherine
1. Implement data structure diagrams	50%	50%	0%
2. Conduct evaluation and analyze results	0%	0%	100%
3. Create poster and ebook page for Senior Design Showcase	100%	0%	0%

Thank You