

Download a copy of this report for your personal use this semester. Likewise, please remove this header and the square brackets before submitting. It shows attention to detail. You may use Google Documents, Microsoft Word, or Mac Pages to work on this document. You will submit this as a PDF. Corrupted PDFs will automatically be graded as 0 points.

CSC Problem Set 03 +X Report

Name: [Sheel Patel]

Email: [scpate16@ncsu.edu]

Section: [002]

Collaborators: [None]

Summary:

[Include a short 1 paragraph summary of what your +X implementation was about. Simply put, "what did you do?"]

For this +x implementation involved making updates to the GUI for better user experience when it came to running the simulations. More specifically, I created a button that allows users to pick an input file from the inputs before the simulation starts. Additionally, I have users enter a number "A" and then generate an AxA grid of random numbers to run the simulation on. The user must click start after either selecting an already existing file or click generate grid after entering the number.

"+X" Concept:

[Includes a more in-depth 1-2 paragraph explanation and justification for your implementation. If you use additional resources, like additional Java libraries, websites, videos, 'inspirations' (things that sparked ideas in your head), etc., please include hyperlinks to them. For the sake of getting practice, please use [IEEE citation formatting](#) [1] and include the citations in the Works Referenced section. See the [1] example in the Works Referenced Section for an example.]

These changes allow users to run the simulation as many times as they want with selecting different files without having to restart the application. Additionally, the random generator allows users to test their simulation against random numbers with grids that can get really large. All in all these changes make the user experience more smooth and allow for faster testing.

The main reason I included these changes was to ensure users can have an easier time testing along with making sure they have a chance to test more rigorously without having to create numbers themselves. I used online libraries to create these changes we mentioned in the next section.

Download a copy of this report for your personal use this semester. Likewise, please remove this header and the square brackets before submitting. It shows attention to detail. You may use Google Documents, Microsoft Word, or Mac Pages to work on this document. You will submit this as a PDF. Corrupted PDFs will automatically be graded as 0 points.

Technical Implementation of +X:

[Provide your technical approach in 1-3 paragraphs on implementing this "+X". Include code snippets, pseudocode, graphics, or an algorithm flowchart to demonstrate the complexity or novelty of the addition.]

Code wise I JPanel to create a panel for which I used JButton to create buttons for selecting a file, generate, and start simulation. I also added action listeners on each button to make sure the file is selected, numbers are generated, and the simulations starts. I also created a few methods to run each task. In the random grid generator method I generate random numbers between 0 and 99 inclusive and write them to a file random.txt. When the user presses generate the random.txt file will be selected automatically after generating the numbers. I also have a select file method that uses JFileChooser to make sure users have selected the file from the proper folder. Lastly, the start simulation method that allows users to start the simulation.

Online libraries I used were [awt](#) [2], [swing](#) [3], [util](#) [4], and [io](#) [5].

Some code changes:

```
public VisualizeSimulation() {  
  
    JPanel controlPanel = new JPanel(new FlowLayout());  
  
    selectMapButton = new JButton("Select Map File");  
    controlPanel.add(selectMapButton);  
  
    gridSizeField = new JTextField(5);  
    controlPanel.add(new JLabel("Grid Size:"));  
    controlPanel.add(gridSizeField);  
  
    generateGridButton = new JButton("Generate");  
    controlPanel.add(generateGridButton);  
  
    startButton = new JButton("Start");  
    startButton.setEnabled(false);  
    controlPanel.add(startButton);  
  
    selectMapButton.addActionListener(this::selectMapFile);  
    startButton.addActionListener(this::startSimulation);  
    generateGridButton.addActionListener(this::generateRandomGrid);  
  
    add(controlPanel, BorderLayout.NORTH);  
  
}
```

Download a copy of this report for your personal use this semester. Likewise, please remove this header and the square brackets before submitting. It shows attention to detail. You may use Google Documents, Microsoft Word, or Mac Pages to work on this document. You will submit this as a PDF. Corrupted PDFs will automatically be graded as 0 points.

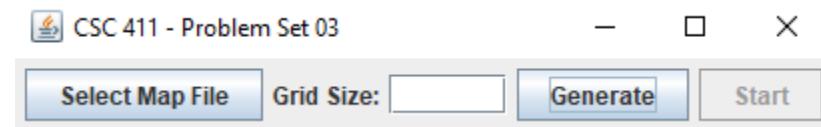
```
private void startSimulation(ActionEvent e) {  
  
    // Load configurations from the config file  
    Properties properties = ConfigurationLoader.loadConfiguration(configFile);  
    int ITERATIONS = Integer.parseInt(properties.getProperty("ITERATIONS", "200"));  
    int TILESIZE = Integer.parseInt(properties.getProperty("TILESIZE", "50"));  
    int DELAY = Integer.parseInt(properties.getProperty("DELAY", "200"));  
    boolean DEBUG = Boolean.parseBoolean(properties.getProperty("DEBUG", "true"));  
  
    // Load the selected map file  
    int[][] map = InputManager.loadMap(mapFile);  
  
    this.env = new Environment(map);  
    envPanel = new EnvironmentPanel(this.env, ITERATIONS, TILESIZE, DELAY, DEBUG);  
    scorePanel = new ScorePanel();  
    this.add(envPanel, BorderLayout.CENTER);  
    this.add(scorePanel, BorderLayout.SOUTH);  
    setResizable(false);  
  
    pack();  
    repaint();  
}  
  
private void selectMapFile(ActionEvent e) {  
    JFileChooser fileChooser = new JFileChooser(new File("inputs/public"));  
    fileChooser.setFileSelectionMode(JFileChooser.FILES_ONLY);  
  
    int returnValue = fileChooser.showOpenDialog(this);  
    if (returnValue == JFileChooser.APPROVE_OPTION) {  
        File selectedFile = fileChooser.getSelectedFile();  
        mapFile = selectedFile.getAbsolutePath();  
        selectMapButton.setText("Map: " + selectedFile.getName());  
        startButton.setEnabled(true);  
        gridGenerated = false;  
    }  
  
    try {  
        int size = Integer.parseInt(gridSizeField.getText());  
  
        int[][] randomGrid = new int[size][size];  
        Random rand = new Random();  
  
        // Generate grid with random values (0-99)  
        for (int i = 0; i < size; i++) {  
            for (int j = 0; j < size; j++) {  
                randomGrid[i][j] = rand.nextInt(100);  
            }  
        }  
  
        // Save the grid to random.txt  
        File file = new File("inputs/public/random.txt");  
        try (PrintWriter writer = new PrintWriter(new FileWriter(file))) {  
            for (int[] row : randomGrid) {  
                for (int num : row) {  
                    writer.print(num + " ");  
                }  
                writer.println();  
            }  
        }  
    }  
}
```

Download a copy of this report for your personal use this semester. Likewise, please remove this header and the square brackets before submitting. It shows attention to detail. You may use Google Documents, Microsoft Word, or Mac Pages to work on this document. You will submit this as a PDF. Corrupted PDFs will automatically be graded as 0 points.

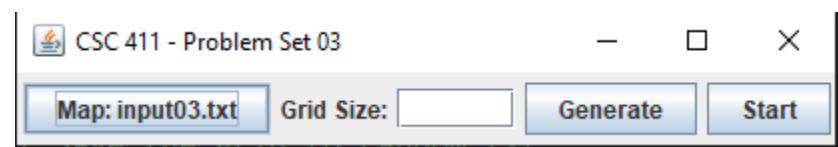
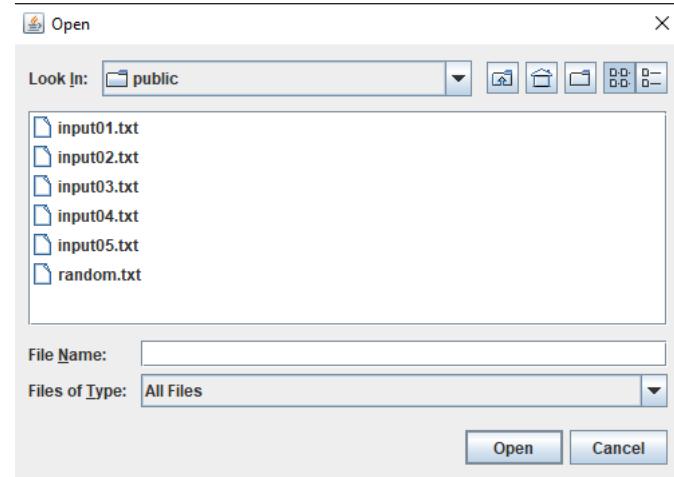
Evaluation and Results:

[Present the outcome or results of adding your "+X" in 1-3 paragraphs. Use tables, graphs, screenshots, or diagrams to compare the implementation's performance with and without the enhancement.]

The results include a better user experience with being able to load any file and also being able to generate any set of numbers for custom grid size. The visualization starts below:



User can select a file



User can then start simulation

Download a copy of this report for your personal use this semester. Likewise, please remove this header and the square brackets before submitting. It shows attention to detail. You may use Google Documents, Microsoft Word, or Mac Pages to work on this document. You will submit this as a PDF. Corrupted PDFs will automatically be graded as 0 points.



User can also enter grid size number

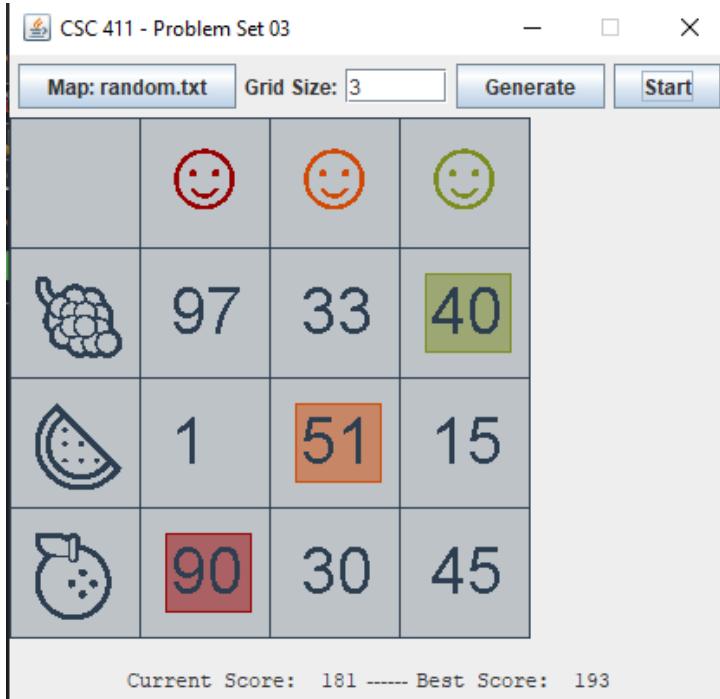


User can press generate and then start button becomes enabled



User can then start

Download a copy of this report for your personal use this semester. Likewise, please remove this header and the square brackets before submitting. It shows attention to detail. You may use Google Documents, Microsoft Word, or Mac Pages to work on this document. You will submit this as a PDF. Corrupted PDFs will automatically be graded as 0 points.



Once execution is completed the user can select another file or enter another number to generate numbers. This system allows for convenient ways of running our simulation as users can select their file at execution (users can also add their field to input if they want something custom) or generate some numbers to test against.

Works Referenced:

[Include links and citations to everything your reference within this document in IEEE citation format. Example for the [1] reference in the "+X" Concept section:

1. [1] IEEE Publication Operations. *IEEE Reference Style Guide for Authors*. (2023). Accessed: 01/12/2025. [Online]. Available: <https://journals.ieee.org/authors/style-guide>.
2. [2] Oracle, “java.awt (Java Platform SE 7),” docs.oracle.com. <https://docs.oracle.com/javase/7/docs/api/java/awt/package-summary.html> (accessed Jan. 17, 2025)
3. [3] Oracle, “javax.swing (Java Platform SE 7),” docs.oracle.com. <https://docs.oracle.com/javase/7/docs/api/javax/swing/package-summary.html> (accessed Jan. 17, 2025).
4. [4] “java.util (Java Platform SE 8),” Oracle.com, Jan. 06, 2020. <https://docs.oracle.com/javase/8/docs/api/java/util/package-summary.html> (accessed Feb. 14, 2025).

Download a copy of this report for your personal use this semester. Likewise, please remove this header and the square brackets before submitting. It shows attention to detail. You may use Google Documents, Microsoft Word, or Mac Pages to work on this document. You will submit this as a PDF. Corrupted PDFs will automatically be graded as 0 points.

5. [5] “java.io (Java Platform SE 8),” docs.oracle.com.

<https://docs.oracle.com/javase/8/docs/api/java/io/package-summary.html> (accessed Feb. 15, 2025).