

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 ## +X Report

Name: [Sheel Patel]

Email: [scpate16@ncsu.edu]

Section: [002]

Collaborators: [No Other Student]

Summary:

For my +X section of the problem set I decided to work on updating the GUI to be more of a debugger. The GUI setup right now when doing visual simulation just runs the and shows the robot traversing. When I was working on the implementation tested constantly using the visual simulation, I thought being able to stop the execution in a spot along with being able to move one step forward in time and one step backward using buttons would have been really helpful for gaining an understanding of what the robot is doing. This would also help understand the algorithm better. So I decided to implement those three features by being able to pause the execution (and timer), being able to go one step forward and backward.

"+X" Concept:

For the +X concept I went with a debugger that allows users to stop the execution of the simulation, step forward and step backward. I created this since it allows users a better understanding of how their algorithm works and what the robot is doing at each step. For instance, the user can map each step to their algorithm and understand where they might be going wrong or where they might be able to improve things. I took inspiration from our normal java debugger since it has helped out in various situations where I could proceed step by step and figure out where I was making a mistake.

I used online libraries and websites to get information on how to add buttons and add event listeners that will allow me to pause the timer and step forward and backward. This information is covered in the next section since it includes technical information as well.

[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.]

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:

To accomplish this I looked at the already written code to develop a basic layout of the buttons and used an online [website](#) [2] to get the panel information and how to get the buttons setup. The logic itself was pretty straightforward since pausing would pause the timer and step forward would call an already existing method that is called when `getAction` is called in `Environment.java`.

Lastly, I added custom logic in `Environment.java` to include a method that uses a [stack](#) [3] record actions which I can use to move backwards. This allowed me to pop off actions in reverse order and work on step backwards. I created a method similar to the `updateEnvironment` (this recorded actions from `getAction`) called `updateEnvironmentReverse` to accomplish this logic. Furthermore, in `updateEnvironment` when an action was made I pushed to the stack the opposite action so that when it was popped in the reverse method it would get another set of action that needed to be performed and that was backwards. Online libraries I used were [swing](#) [4] and [awt](#) [5].

Following includes some snippets of code I included.

`VisualSimulation.java`

```
private JButton stopExecutionButton = new JButton("Pause");
private JButton iterateForwardButton = new JButton("Step Forward");
private JButton iterateBackwardsButton = new JButton("Step Backward");

JPanel buttonPanel = new JPanel();
buttonPanel.setLayout(new FlowLayout());
buttonPanel.add(stopExecutionButton);
buttonPanel.add(iterateForwardButton);
buttonPanel.add(iterateBackwardsButton);

add(buttonPanel, BorderLayout.AFTER_LAST_LINE);
```

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.

```
        @Override
        public void actionPerformed(ActionEvent e) {
            if (isPaused) {
                timer.start();
                stopExecutionButton.setText("Pause");
                iterateForwardButton.setEnabled(false);
                iterateBackwardsButton.setEnabled(false);
            } else {
                timer.stop();
                stopExecutionButton.setText("Resume");
                iterateForwardButton.setEnabled(true);
                iterateBackwardsButton.setEnabled(true);
            }
            isPaused = !isPaused;
        }
    });

iterateForwardButton.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
        try {
            env.updateEnvironment();
        } catch (Exception e2) {
            System.out.println("Error: Cannot perform action");
        }
        repaint();
    }
});

iterateBackwardsButton.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
        try {
            env.updateEnvironmentReverse();
        } catch (Exception e3) {
            System.out.println("Error: Cannot perform action");
        }
        repaint();
    }
});
```

Environment.java

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.

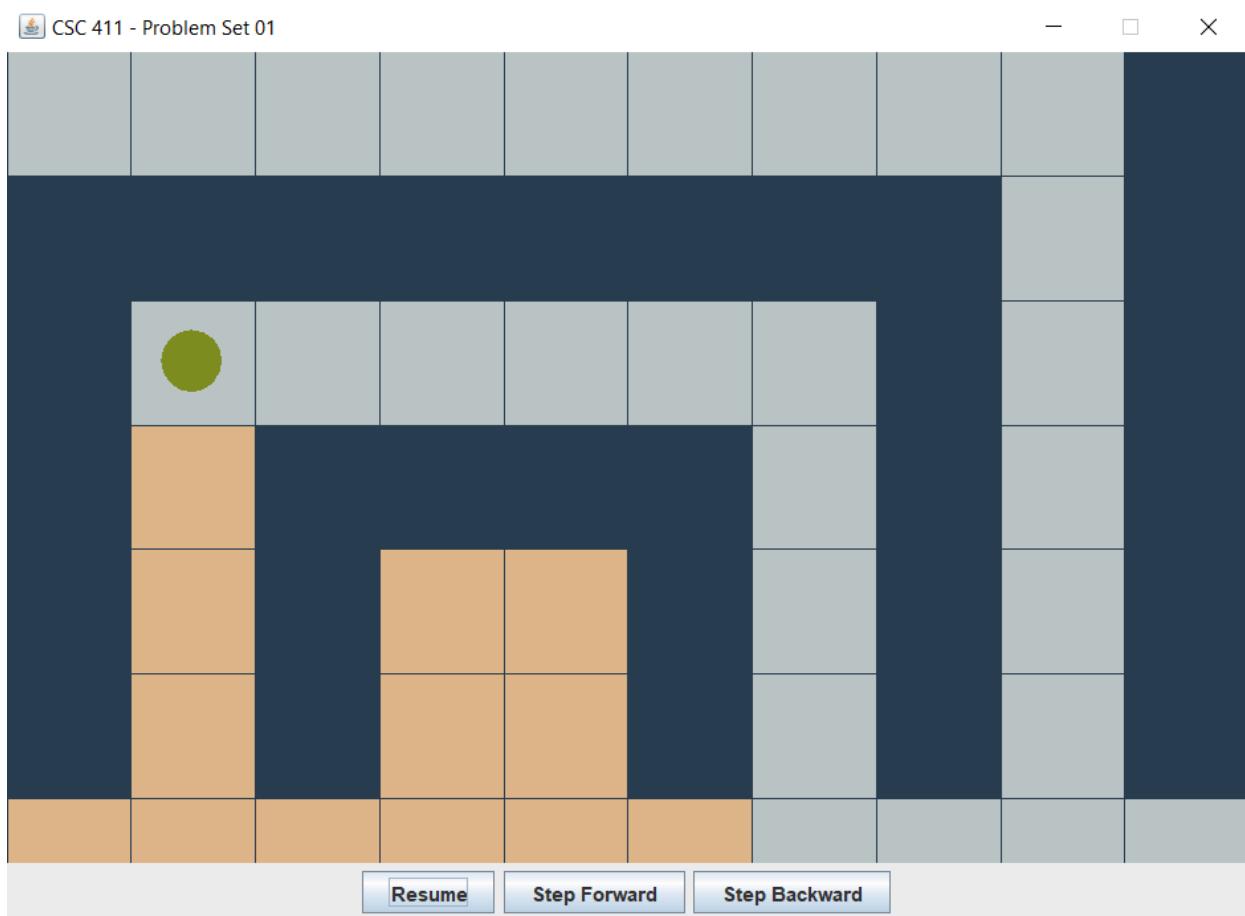
```
public void updateEnvironmentReverse() {
    for (Robot robot : robots) {
        Action action = actionStack.pop();
        Position robotPos = getRobotPosition(robot);
        int row = robotPos.getRow();
        int col = robotPos.getCol();
        switch (action) {
            case CLEAN:
                Tile t = new Tile(TileStatus.DIRTY);
                this.tiles.put(positions[row][col], t);
                break;
            case MOVE_DOWN:
                if (validPos(row + 1, col)) {
                    updateRobotPos(robot, row + 1, col);
                }
                break;
            case MOVE_LEFT:
                if (validPos(row, col - 1)) {
                    updateRobotPos(robot, row, col - 1);
                }
                break;
            case MOVE_RIGHT:
                if (validPos(row, col + 1)) {
                    updateRobotPos(robot, row, col + 1);
                }
                break;
            case MOVE_UP:
                if (validPos(row - 1, col)) {
                    updateRobotPos(robot, row - 1, col);
                }
                break;
            case DO NOTHING: // pass to default
            default:
                break;
        }
    }
}
```

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:

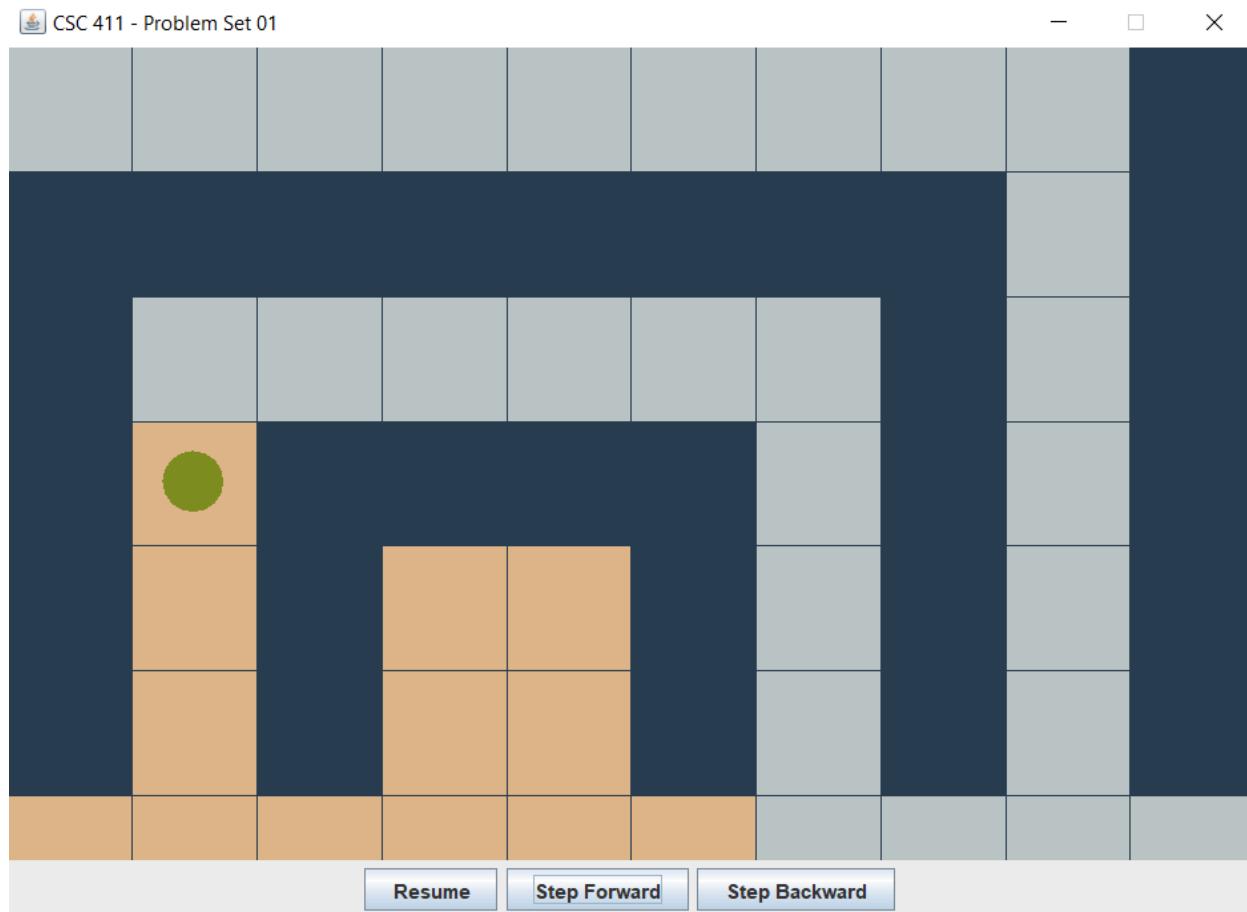
The outcome was quite successful since it allowed me to pause whenever I wanted and conduct each step. THis saved a lot of time as well since I did not have to restart everytime I wanted to see the algorithm at a specific position. This helped me analyze my algorithm as it showed me some steps my robot was taking and how I might be able to make it better. For instance, I had the robot going right, down, left, up but testing it through a lot of these simulations and being able to use the debugger feature, I found overall it was better to go above, right, down, left. Once I made the update the execution was slightly faster. This is due to different map styles, however, it is useful to know the testing grounds and understand how we can make small tweaks. The feature helped me quite a bit when working on this problem set. Some screenshots of its different stages are included below.

Paused State



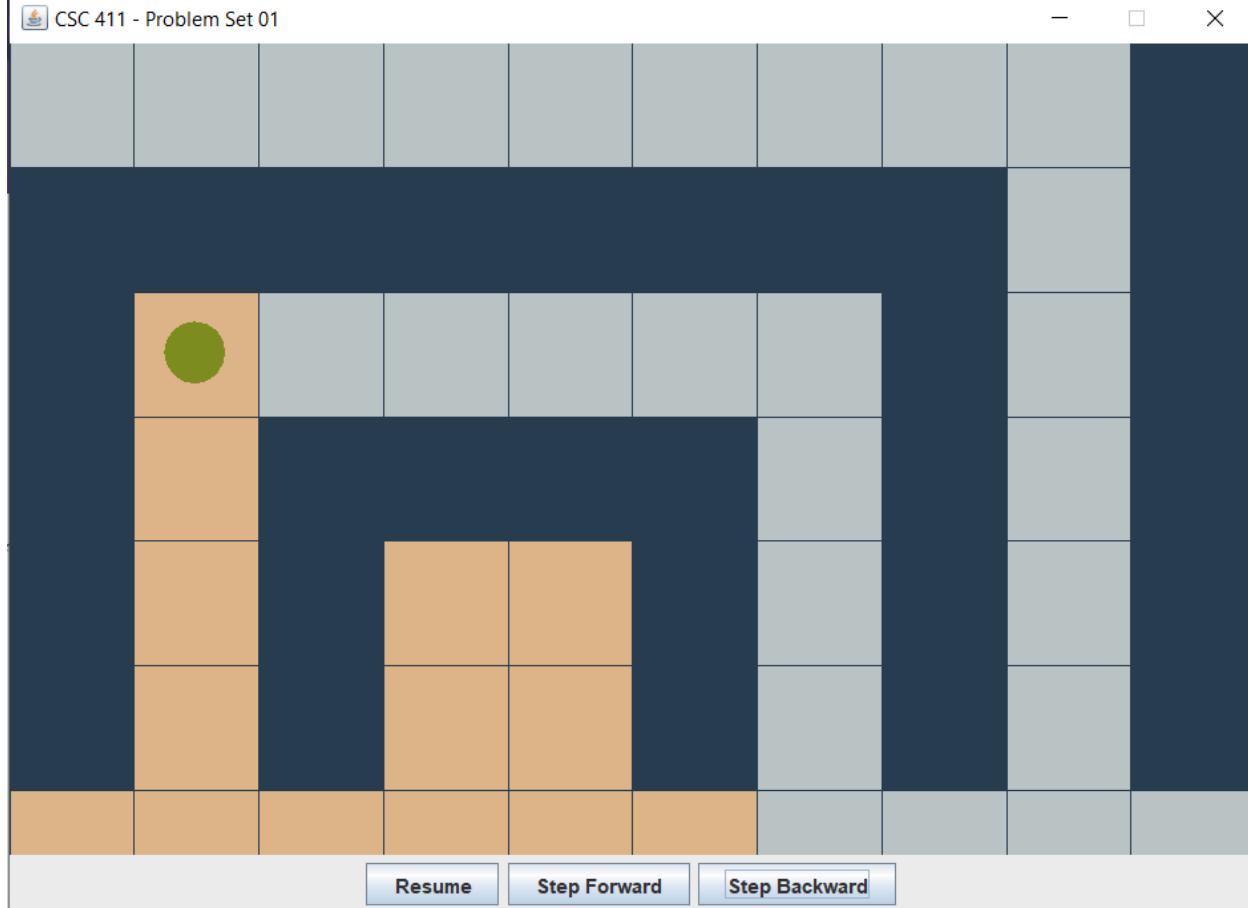
Step Forward

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.



Step Backwards x2

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.



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] D. Burnham, "Tutorial: Java Swing Buttons - Layout | CodeHS," codehs.com <https://codehs.com/tutorial/david/java-swing-buttons-layout> (accessed Jan. 17, 2025).
3. [3] Oracle, "Stack (Java Platform SE 8)," docs.oracle.com. <https://docs.oracle.com/javase/8/docs/api/java/util/Stack.html> (accessed Jan. 16, 2025).
4. [4] 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).

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] Oracle, “javax.swing (Java Platform SE 7),” [docs.oracle.com](https://docs.oracle.com/javase/7/docs/api/javax/swing/package-summary.html).
<https://docs.oracle.com/javase/7/docs/api/javax/swing/package-summary.html> (accessed Jan. 17, 2025).