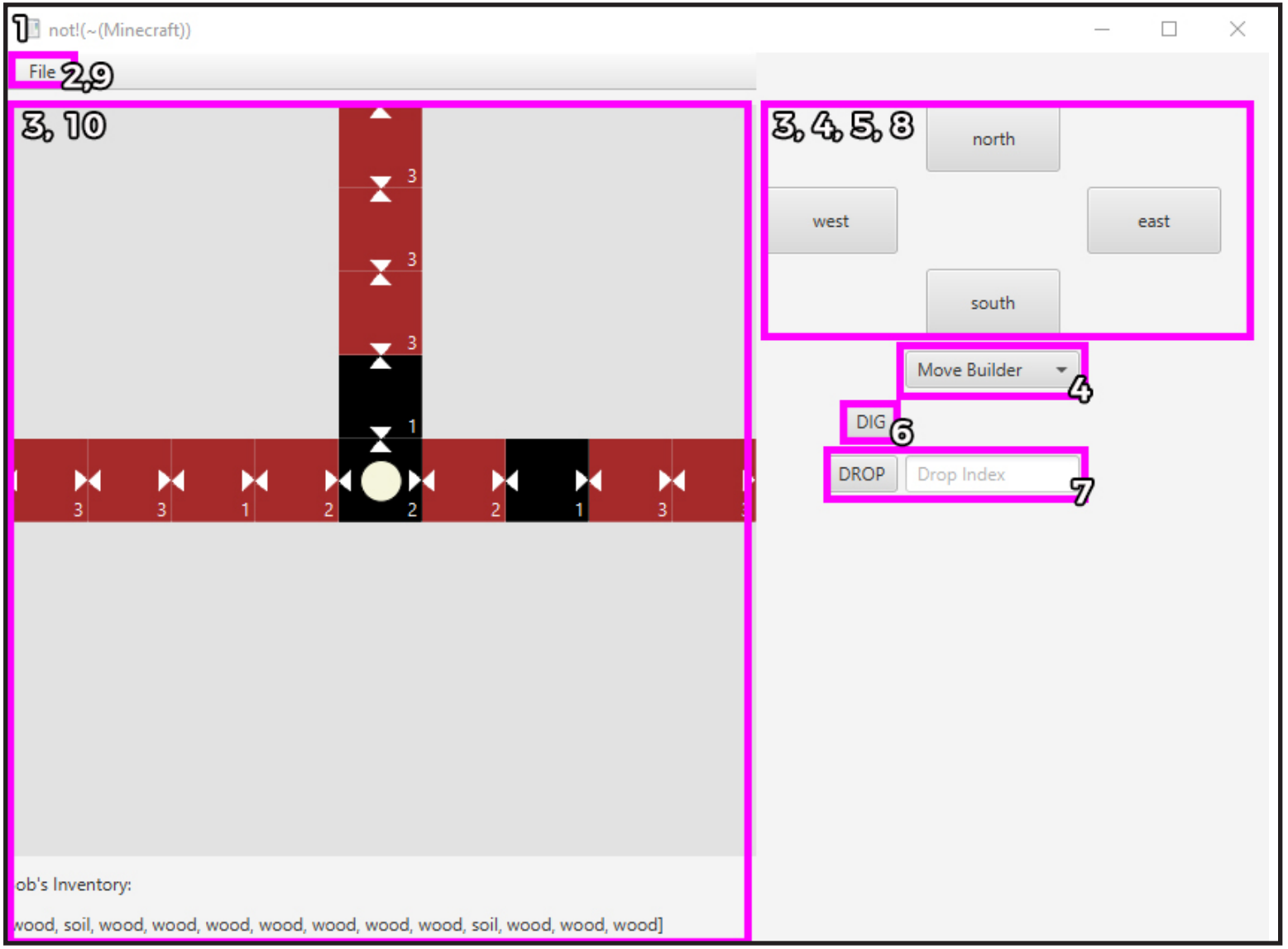
Hello and welcome to my super cool GUI. In this GUI I have included all functionality as well as displays for loaded game worlds. For each exception there is also a custom error box that will show. I hope you enjoy my memey documentation (all memes included in this documentation are original)
- Natalie Hong, some bored IT student who is going to switch to Software Engineering next year.

TABLE OF CONTENTS:



1. LOADING THE GUI

When the gui is compiled in IntelliJ it shows this screen. There is a title with a file menu being shown below. There is also a grid which shows the 9x9 boundaries of the game made using a GridPane. There is also a label below this 9x9 grid which shows the label inventory as well as an empty inventory. To the right, is the button panel. I made this using another GridPane. The buttons are disabled and will be disabled until a WorldMap is loaded (refer to figure 1).

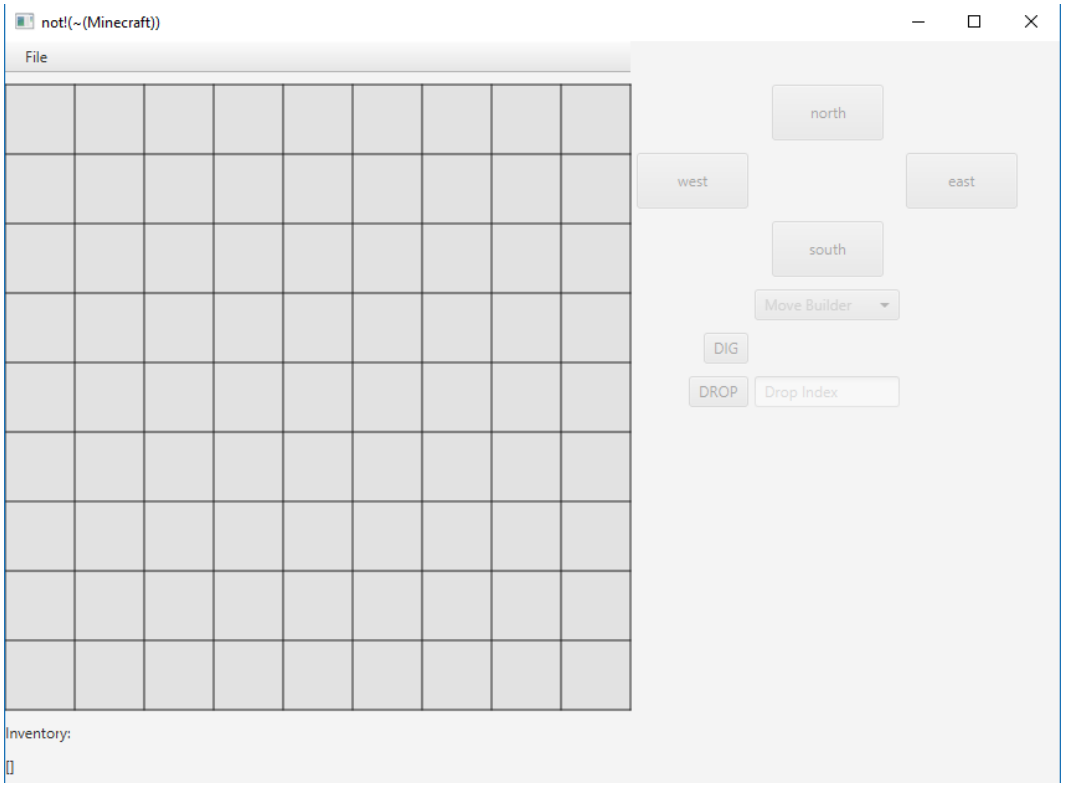


Figure 1 - The GUI Loaded and ready to go

Meme 1 - Never forget the sacrifices your IDE makes for you

2. LOADING A FILE

When clicking the file menu in my GUI (figure 2), you are given two options. For the meantime, I'm not going to worry about Save World until changes have been made to the loaded world. Anyway, clicking the Load Game World will bring up a file chooser pop up (figure 3). From there, you are free to choose a valid world map to load into the GUI.

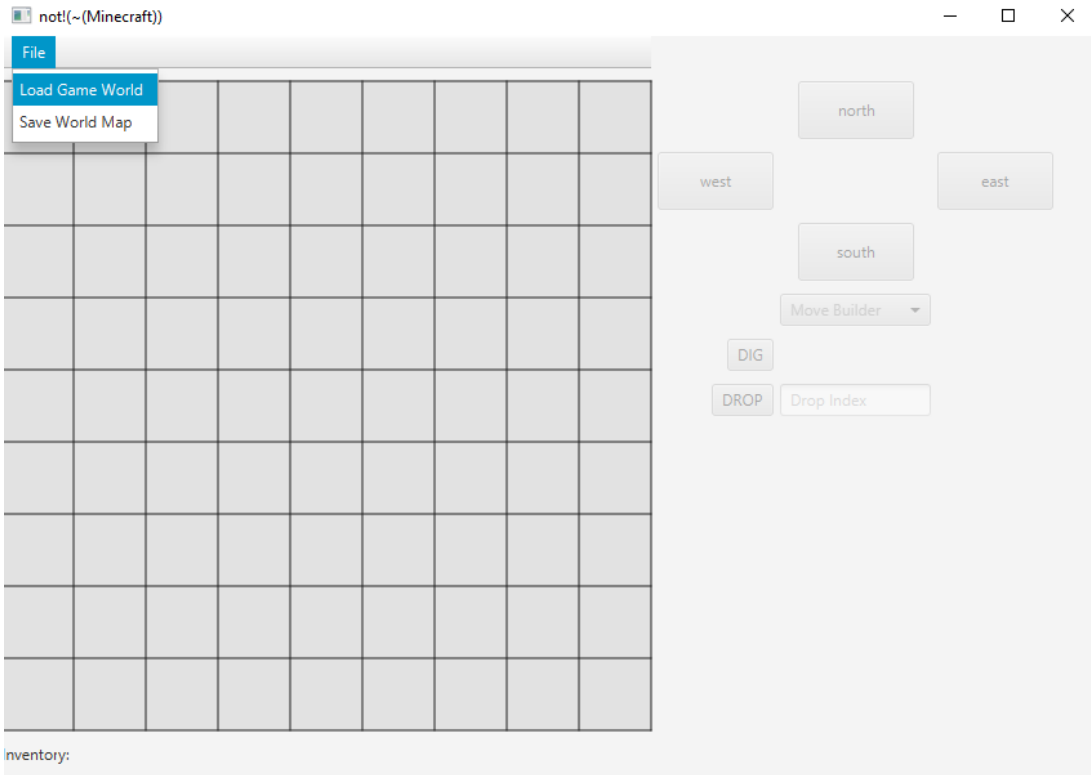


Figure 2 - The File Menu

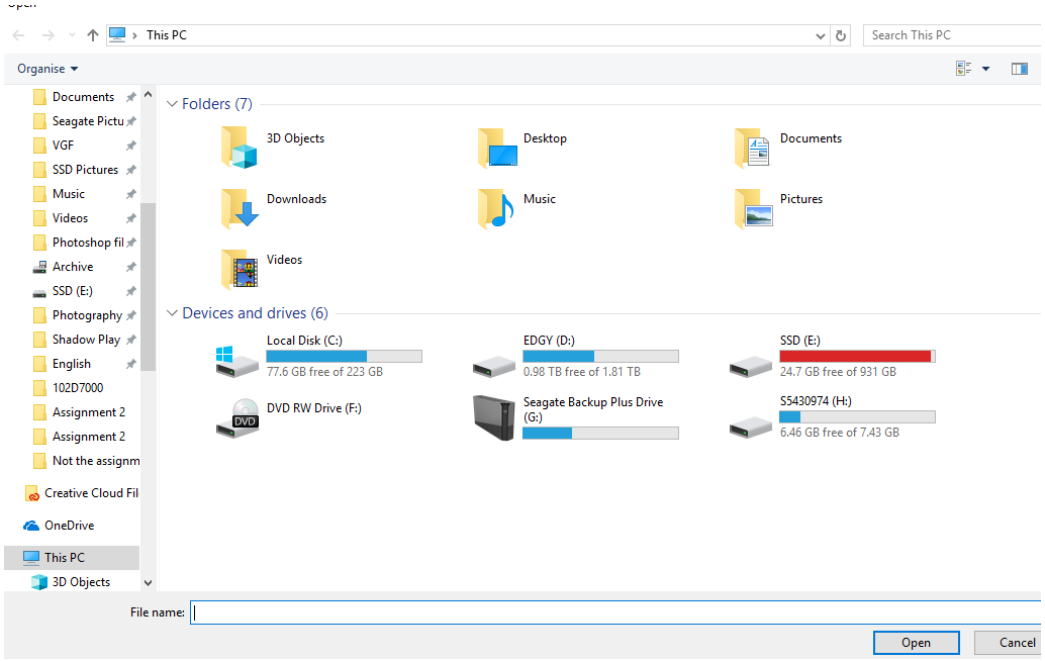
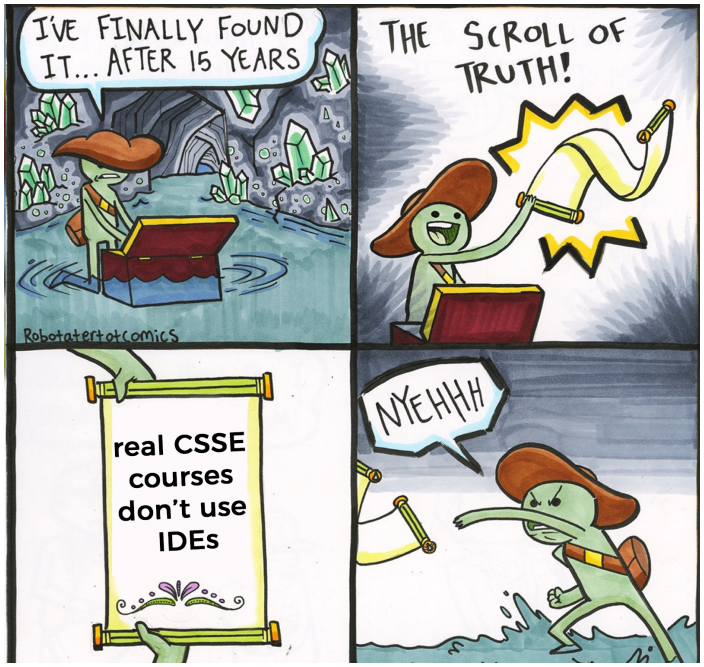


Figure 3 - File Chooser



Meme 2 - The CSSE truth

3. LOADED GAME WORLD

When a valid file is loaded into the GUI, the map will appear with each tile being loaded into the screen. A builder circle will also be placed in the center of the screen. For each tile there are also exits (shown by the white triangles with each direction indicating their corresponding exit directions). There are also corresponding colours respresenting the top block of the tile with brown being wood, black being. soil, grey being stone and green being grass. The number in the bottom right of each tile also represents the amount of blocks in each tile. The buttons on the GUI are also available to be pressed with them as well as the name of the builder being shown along with an inventory to be used (figure 4).

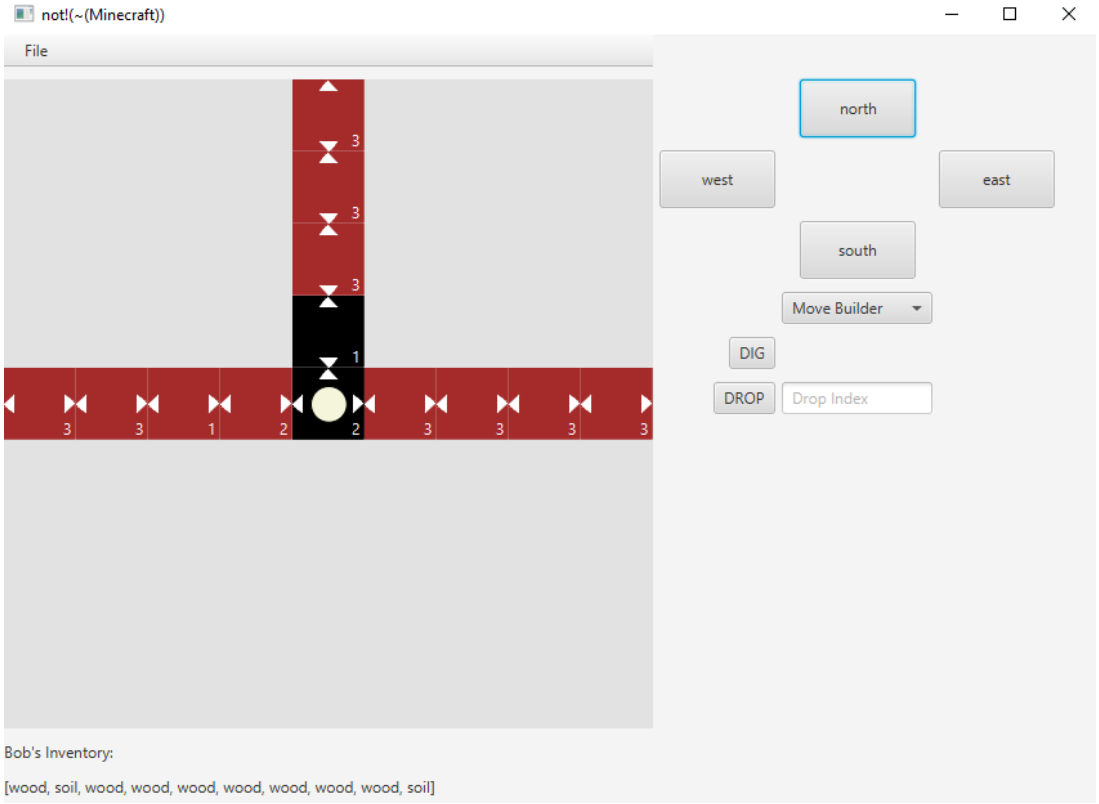


Figure 4 - Loaded Game World

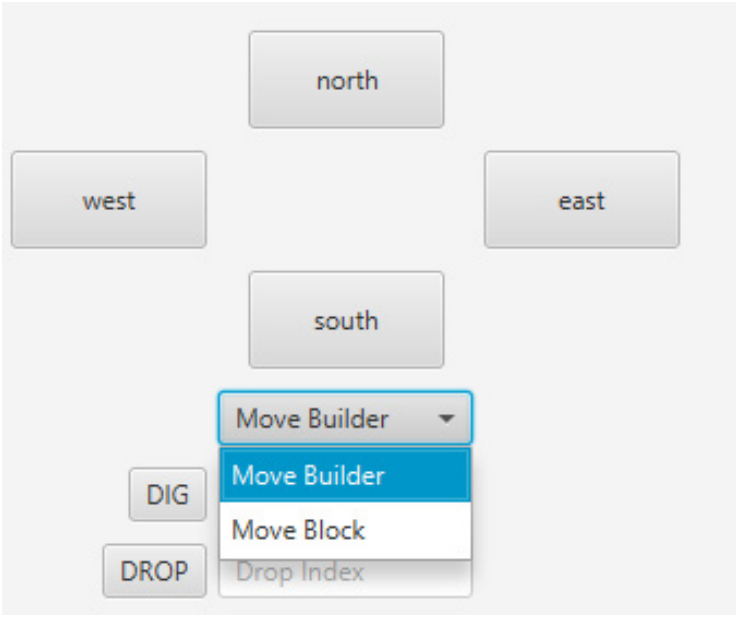


Figure 5 - Action Buttons and ComboxBox

4. ACTION BUTTONS AND THE COMBOBOX

The buttons for the GUI involve directions, a combobox, a dig button, a drop button and a textfield (figure 5). Depending on which value is selected in the comboBox determines whether the action being carried out by the builder is that of moving the builder or moving the current top block on the current tile.

5. MOVING THE BUIILDER/MOVING A BLOCK

When selecting 'Move Builder' from the combo box and pressing one of the directional buttons, if the builder can move to that tile, it will move to the pressed direction. As seen in figure 6, when the east button is pressed, the builder moves to the right as there is an available exit. Similarly, if the move block value is selected and a direction is inputted, the builder will move the top block from the current block in the direction specified if it can. In figure 7, I have inputted move block west and as you can see, the builder has moved the top block from the current block to the west

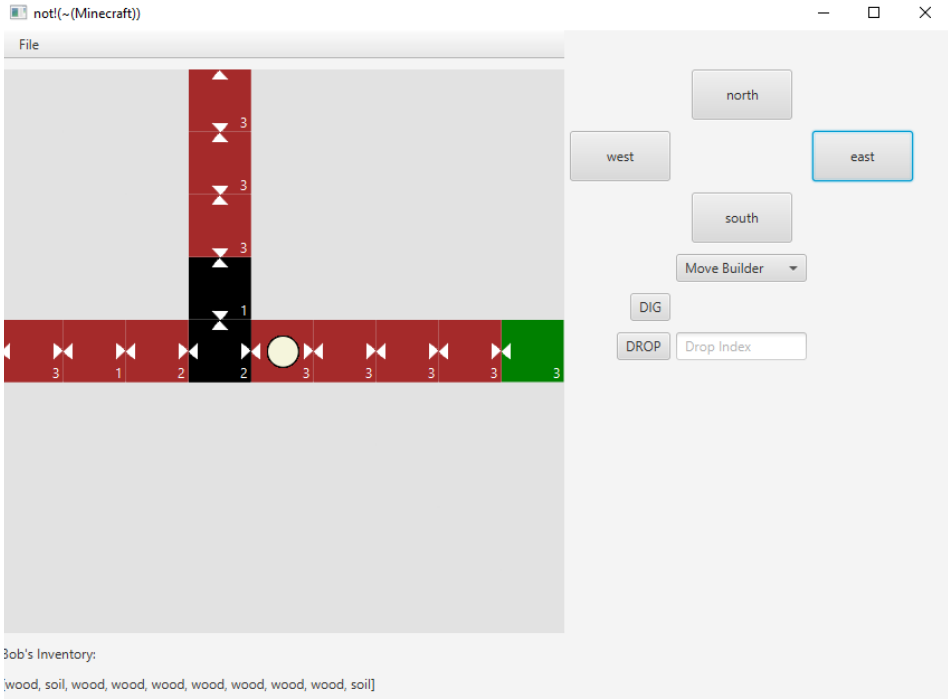


Figure 6 - Moving the builder east

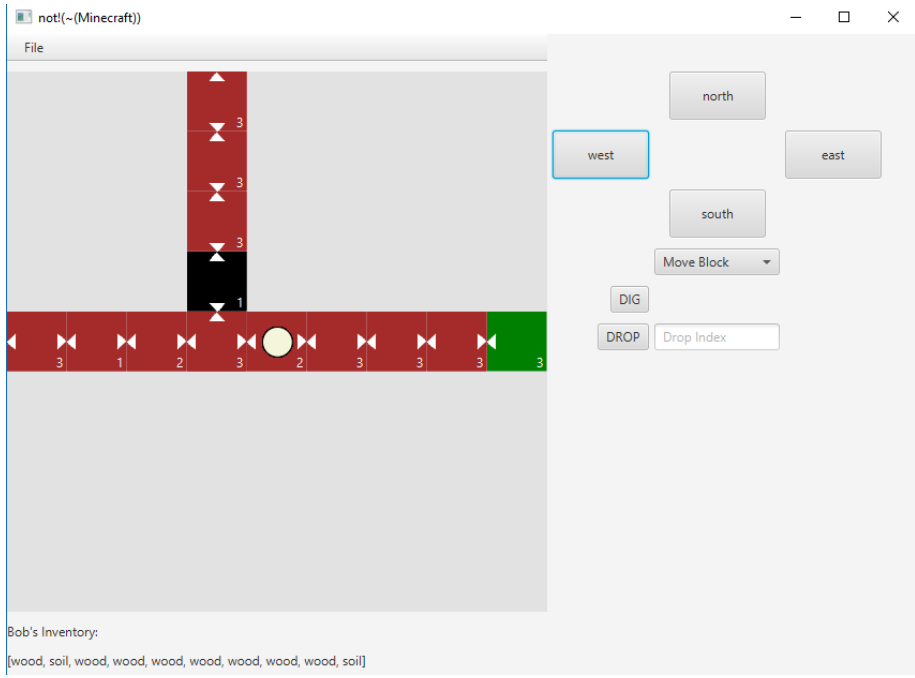
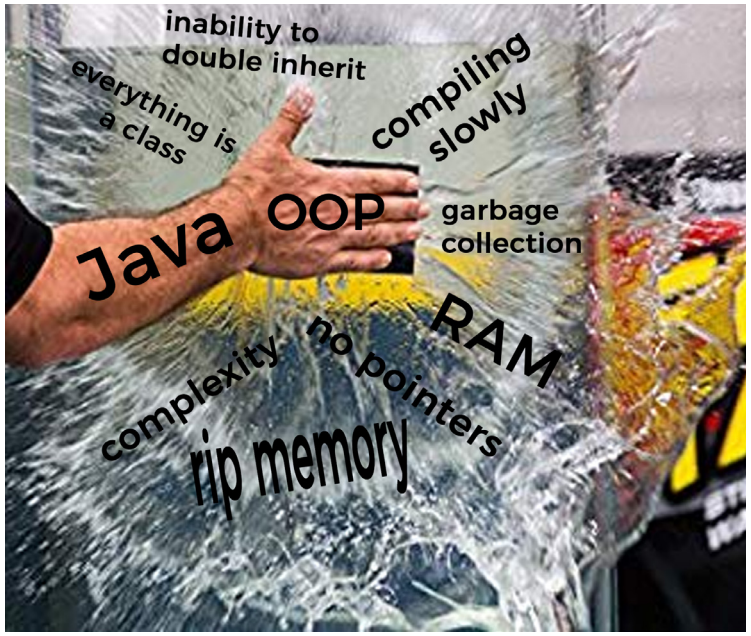


Figure 7 - Moving the builder east



Meme 3 - Java is an OOP language

6. DIGGING ON A TILE

As well as being able to move the builder and blocks, there are also the options of being able to dig on a current tile. When the dig button is pressed the builder then digs on the top block (top block in figure 8) and removes the block from the current tile (making the amount of blocks in the tile decrease by). The block is then added to the builder's inventory if it is carryable (figure 9).

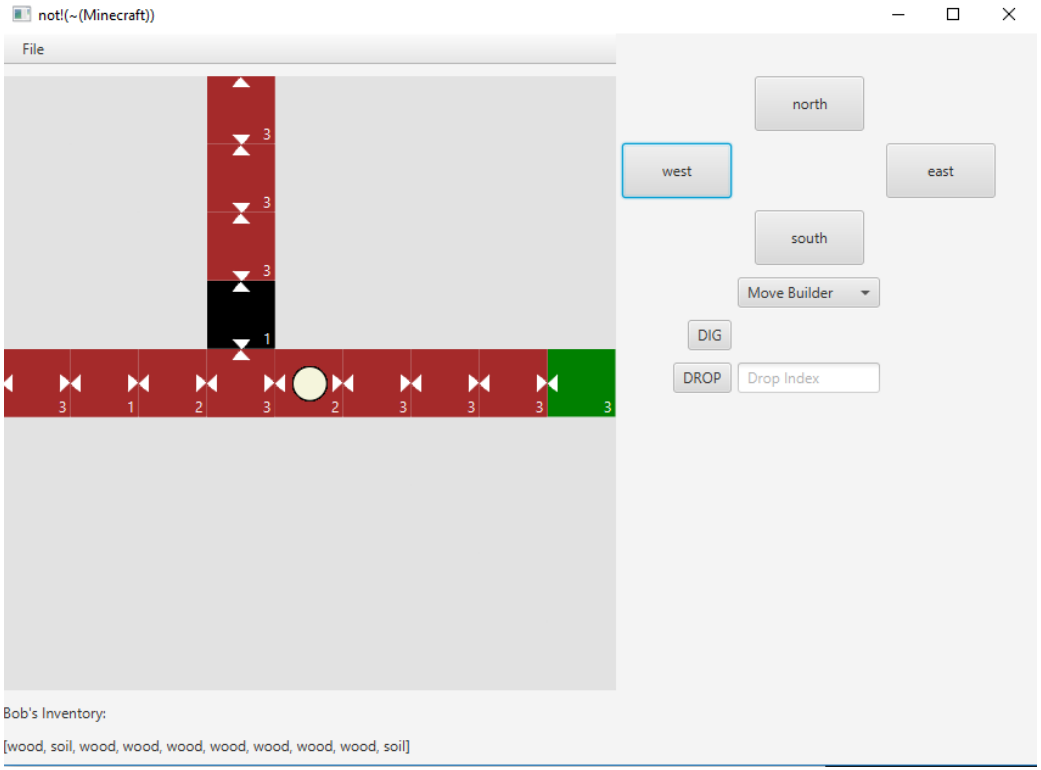
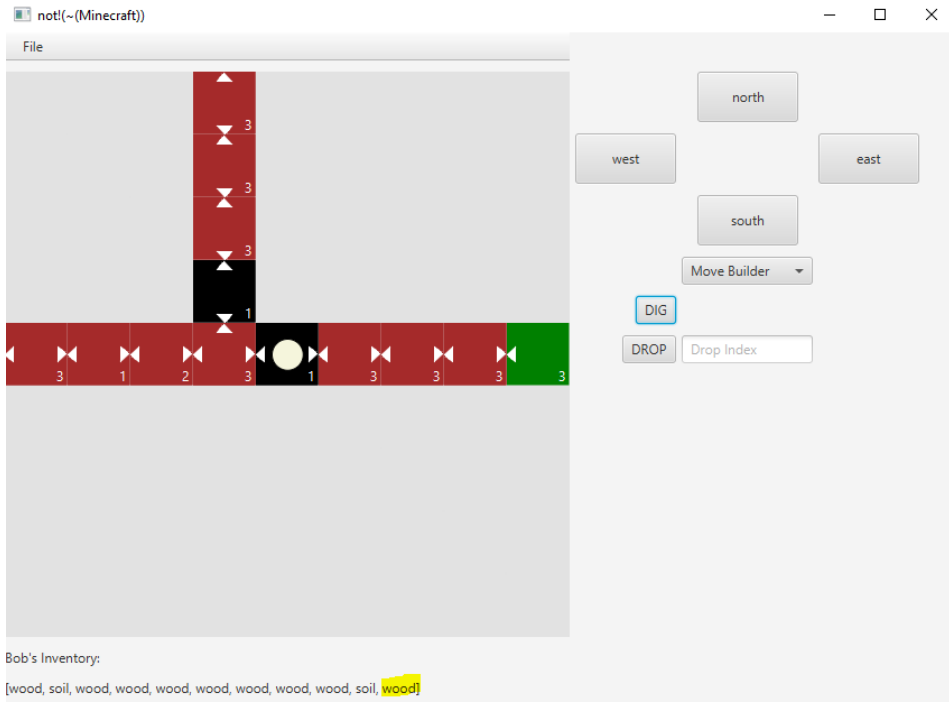


Figure 8 - Map prior to digging on current tile

Figure 9 - Map after to digging on current tile



Meme 4 - CSSE2002 Week 12

7. DROPPING FROM INVENTORY

As well as being able to dig, it is also possible to drop a block from the builder's inventory. To do this, the user must input an integer into the textfield and then press drop. In figure 10, I have inputted 1 which is the index of the second item in my inventory. I then press drop and the block is removed from the inventory and put into the current block. As you can see, the block has been added and the amount of blocks in the tile has increased (figure 11).

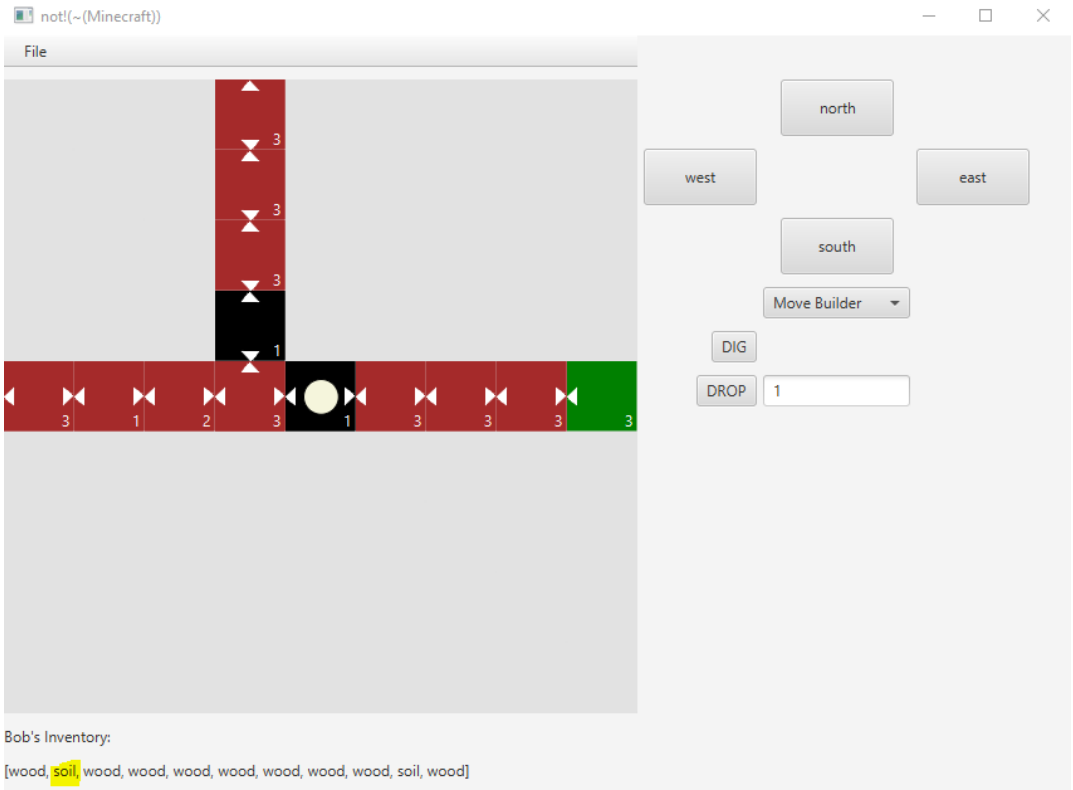


Figure 10 - Map before dropping from inventory

you vs. the guy she tells you not to worry about

```
import java.util.*;

public class CitySearch {

    static void depthFirstSearch(City startingCity) {

        Stack<City> nodesToVisit = new Stack<>();
        Set<City> alreadyVisited = new HashSet<>();

        nodesToVisit.push(startingCity);

        while (nodesToVisit.size() != 0) {
            City city = nodesToVisit.pop();

            if (!alreadyVisited.contains(city)) {

                alreadyVisited.add(city);

                // process city
                System.out.println(city);


                for (CityDistance neighbour : city.getNeighours()) {

                    nodesToVisit.push(neighbour.getCity());
                }
            }
        }
    }
```

```
import java.util.*;

public class CitySearch2 {

    static void breadthFirstSearch(City startingCity) {

        Queue<City> nodesToVisit = new LinkedList<>();
        Set<City> alreadyVisited = new HashSet<>();

        nodesToVisit.add(startingCity);

        while (nodesToVisit.size() != 0) {
            City city = nodesToVisit.remove();

            if (!alreadyVisited.contains(city)) {

                alreadyVisited.add(city);

                // process city
                System.out.println(city);


                for (CityDistance neighbour : city.getNeighours()) {

                    nodesToVisit.add(neighbour.getCity());
                }
            }
        }
    }
```

Meme 5 - DFS vs. BFS

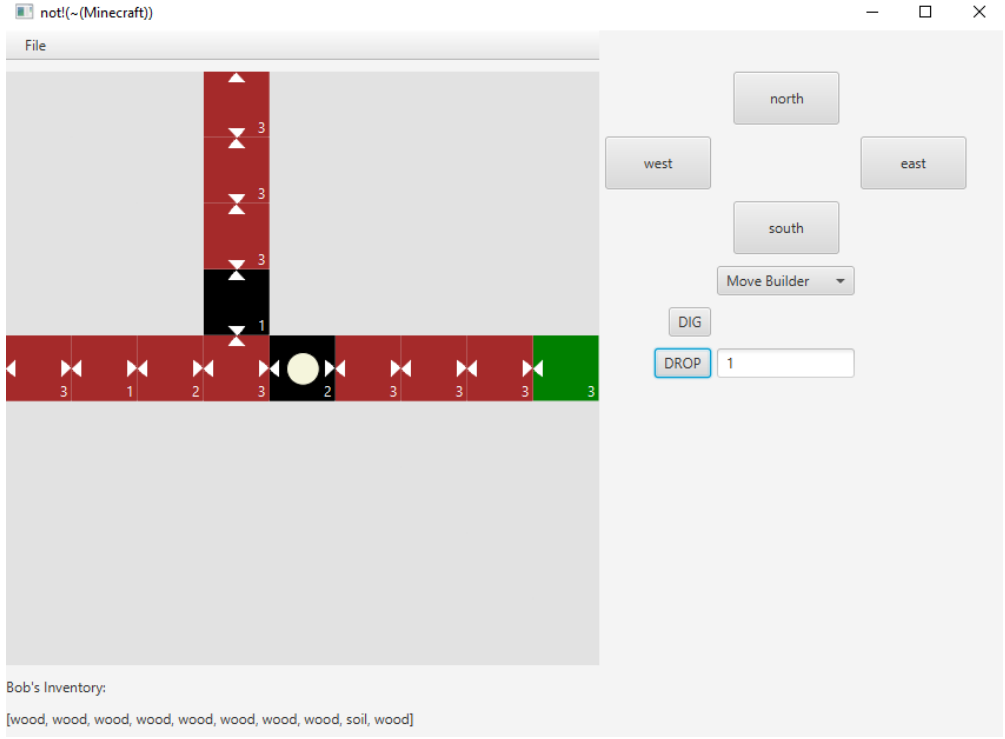


Figure 11 - Map before dropping from inventory

8. ERRORS

As stated in the beginning of this document, there are custom error messages that will show depending on the exception encountered. Some examples are as follows:

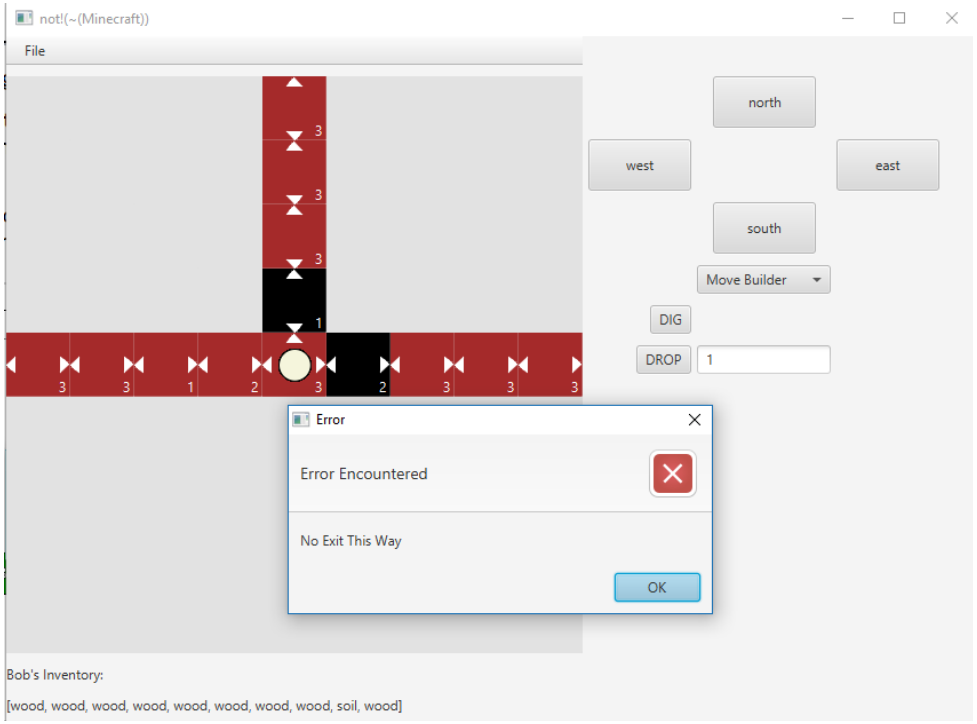


Figure 12.1 - Trying to Move builder north but the difference in tile blocks is too high resulting in a NoExitException.

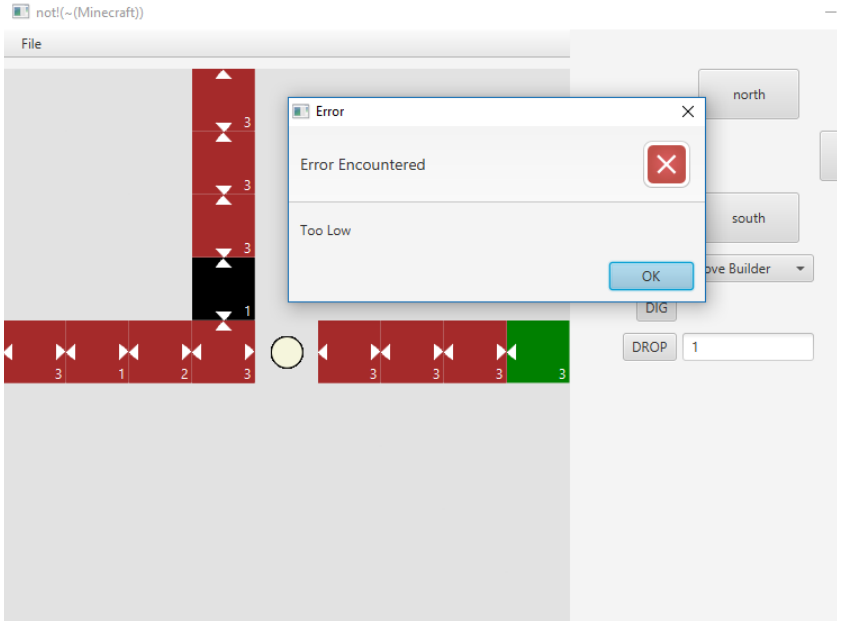


Figure 12.2 - Trying to dig below 0 blocks resulting in a TooLowException

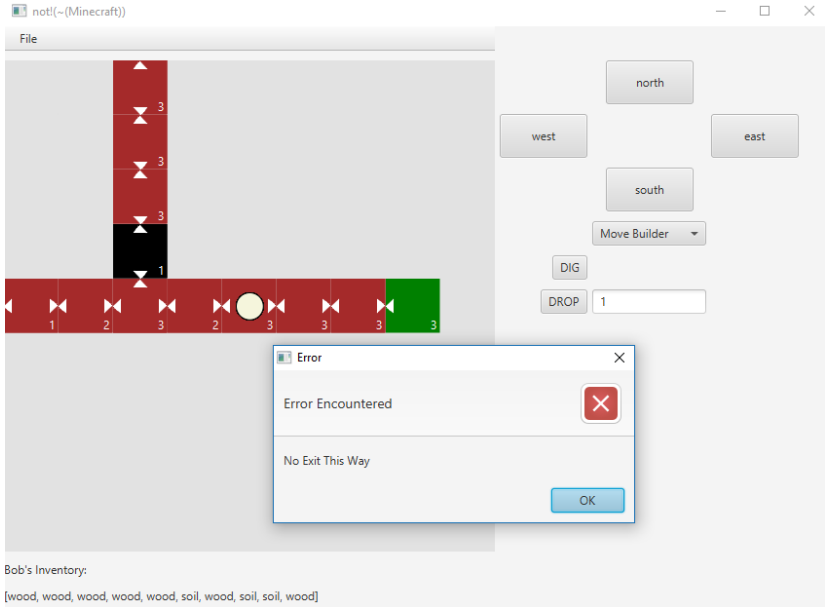


Figure 12.3 - Moving north to through an exit that does not exist resulting in a NoExitException.

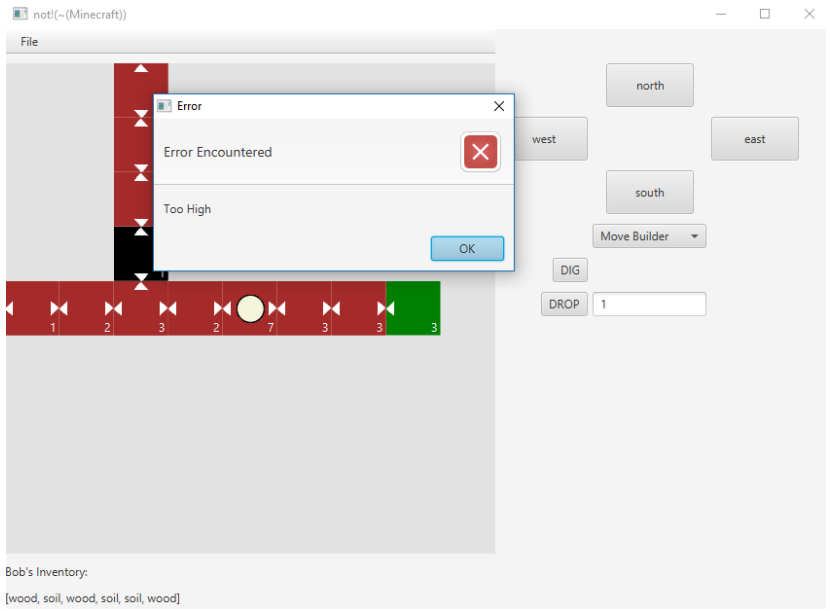


Figure 12.4 - Trying to drop a block on a tile that has 7 blocks resulting in a TooHighException.

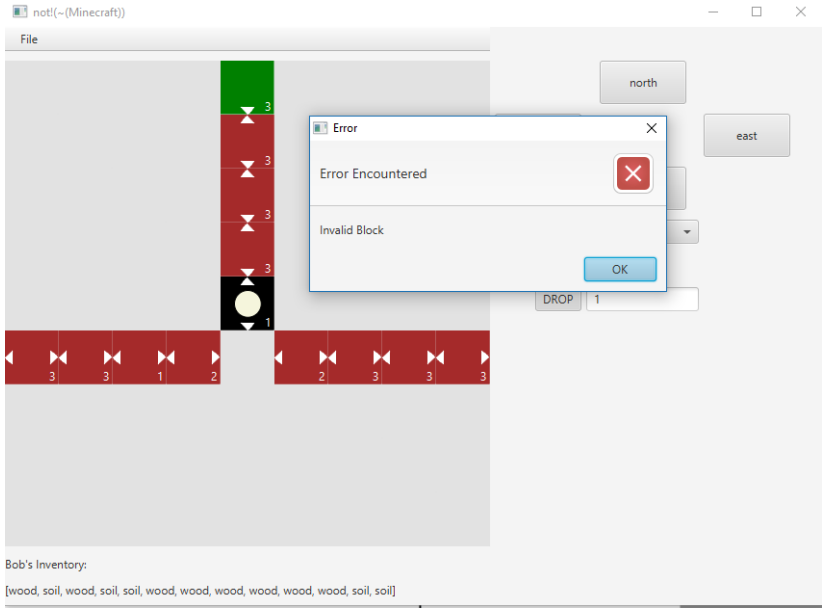


Figure 12.5 - Trying to move an unmoveable block resulting in an InvalidBlockException.

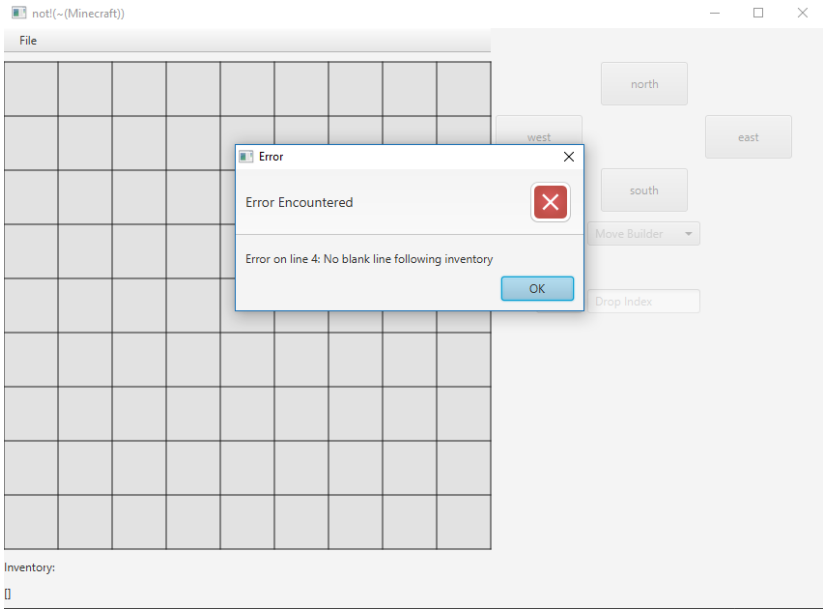


Figure 12.6 - Loading an invalid worldMap resulting in a WorldMapFormatException

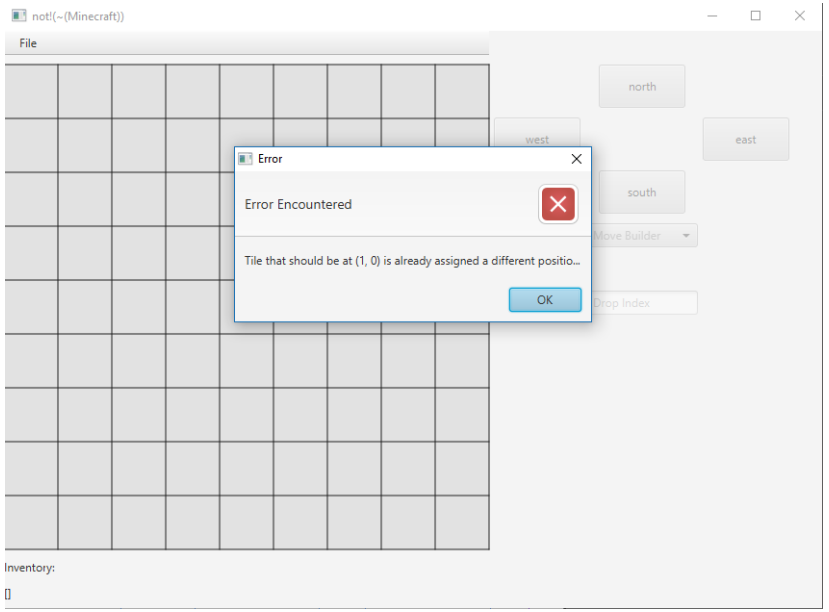


Figure 12.7 - Loading an inconsistent worldmap resulting in a WorldMapInconsistentException

9. SAVING THE MAP

After playing the game the user may want to save their progress. To do this they can choose the save world map option in the file menu (figure 13). The user will then be taken to a file chooser (figure 14) where they can choose where to save their files in any format they want. The saved file will then save all tiles, exits and the builder inventory and will differ from that of the original loaded file (figure 15).

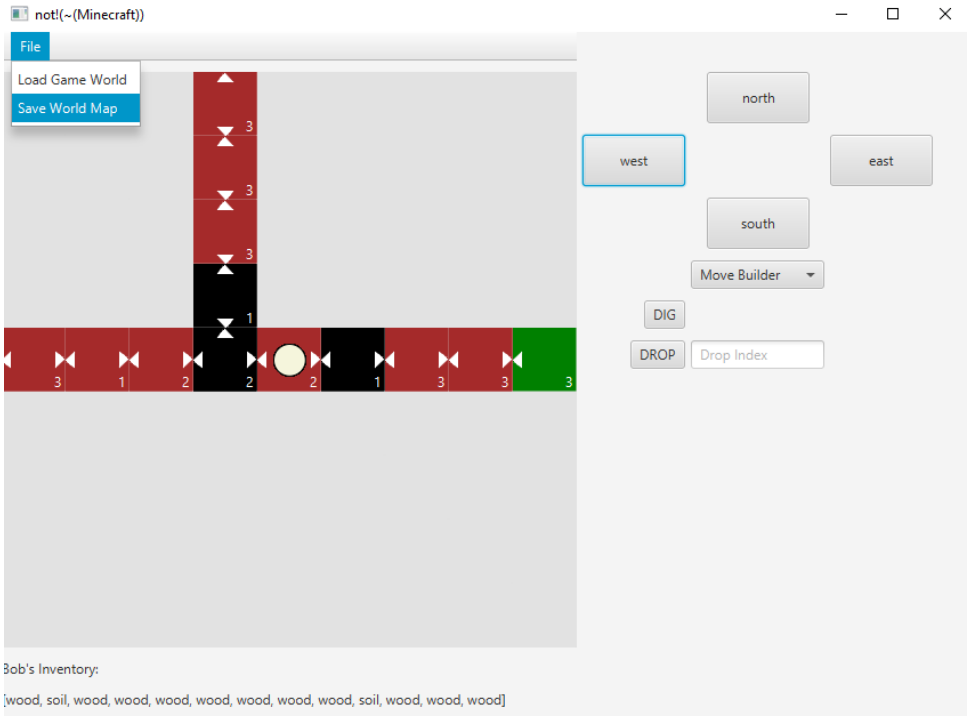


Figure 13 - Selecting the Save World Map option

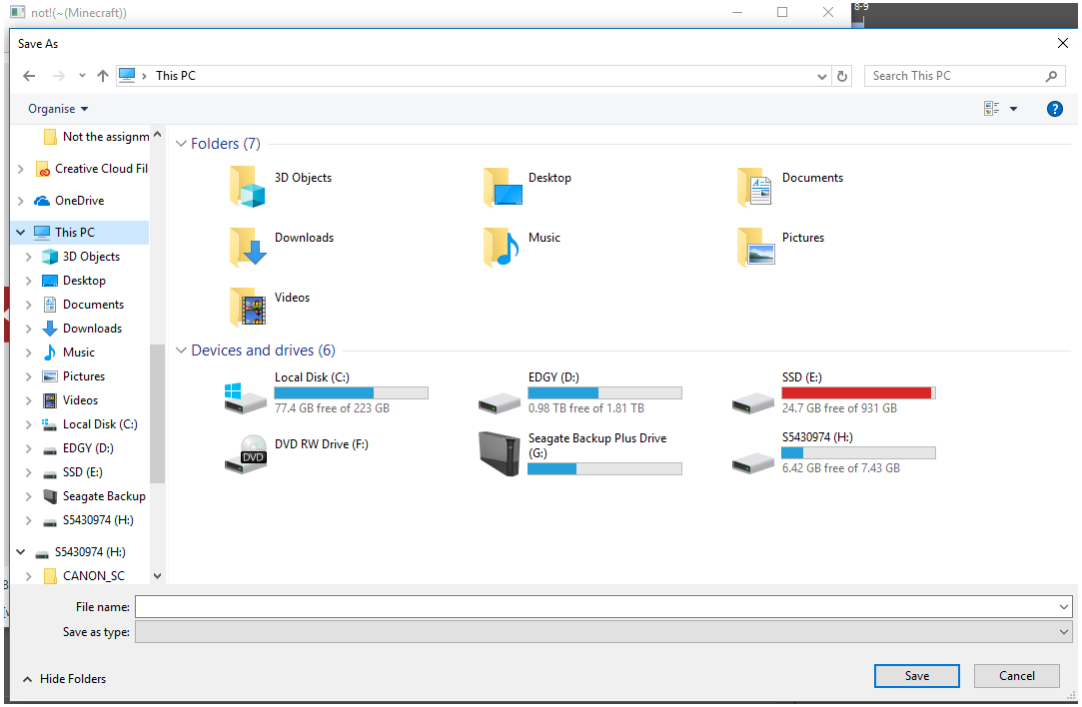


Figure 14 - Save World Map File Chooser

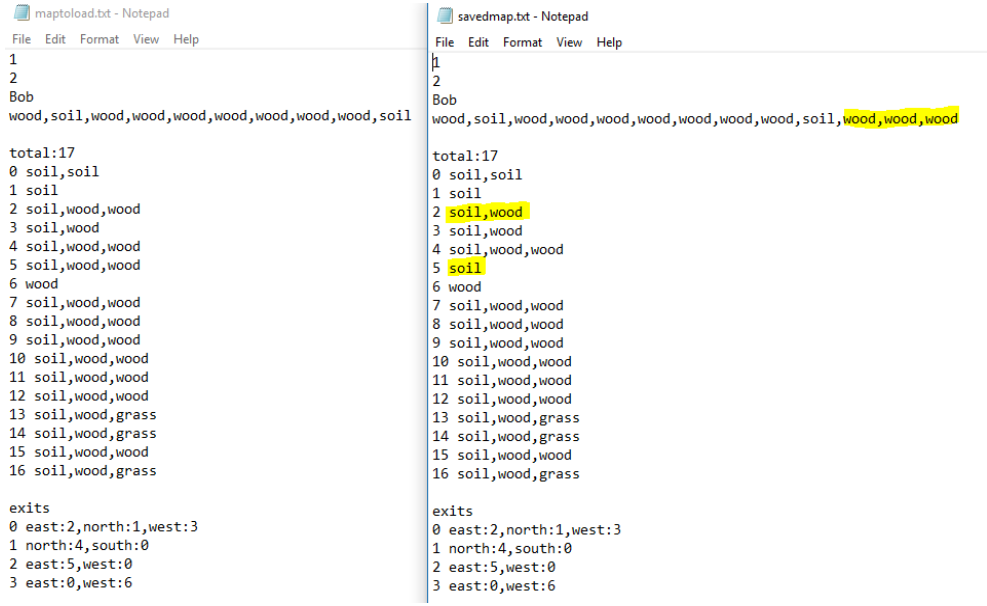


Figure 15 - Loaded File vs. Saved File

10.THE DYNAMIC MAP

As stated in the specifications for this assignment, the builder should be able to traverse through maps that are bigger than the boundaries of the map. To demonstrate this I created grass blocks at the end of the boundaries. As you can see from figures 16-18, as the builder moves further along the map, the map moves below and builder and shifts so that more tiles are shown as they move into the boundaries.

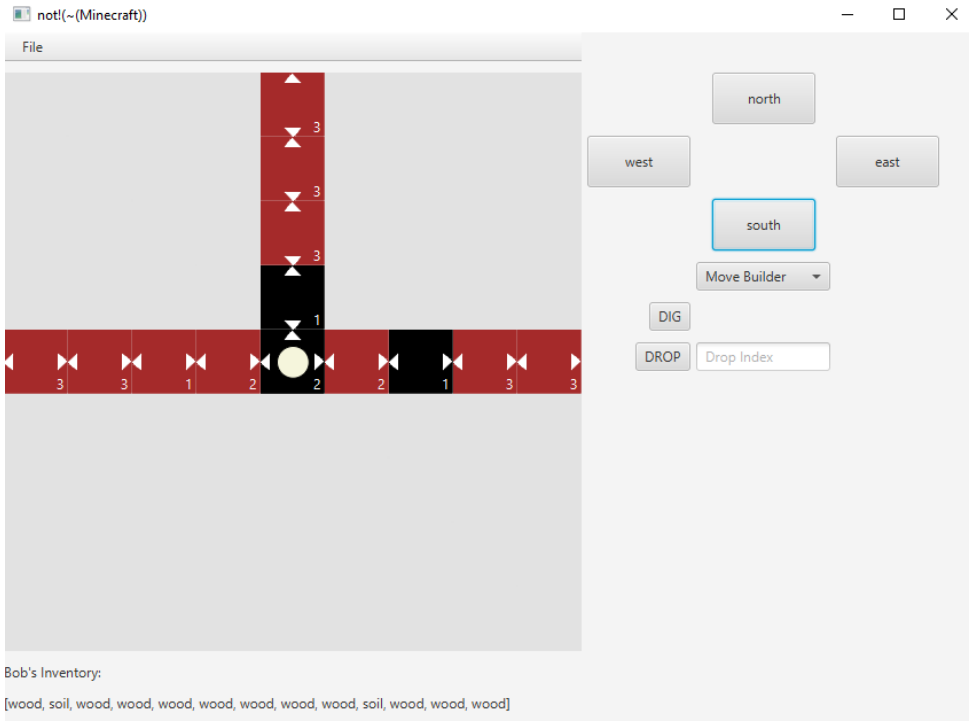


Figure 16 - Centered Map

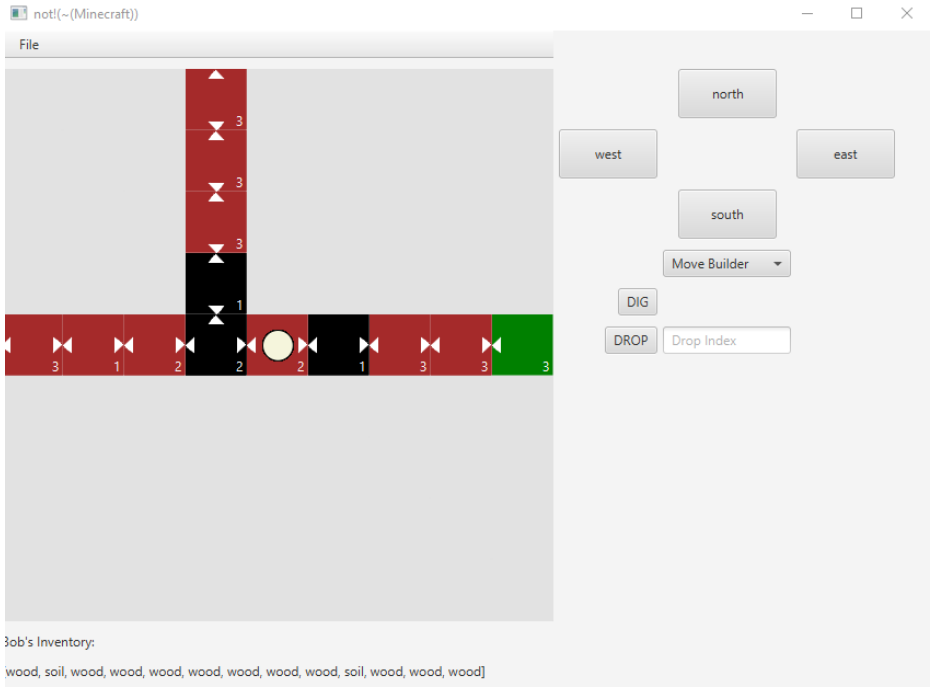


Figure 17 -Moving Map to the Left

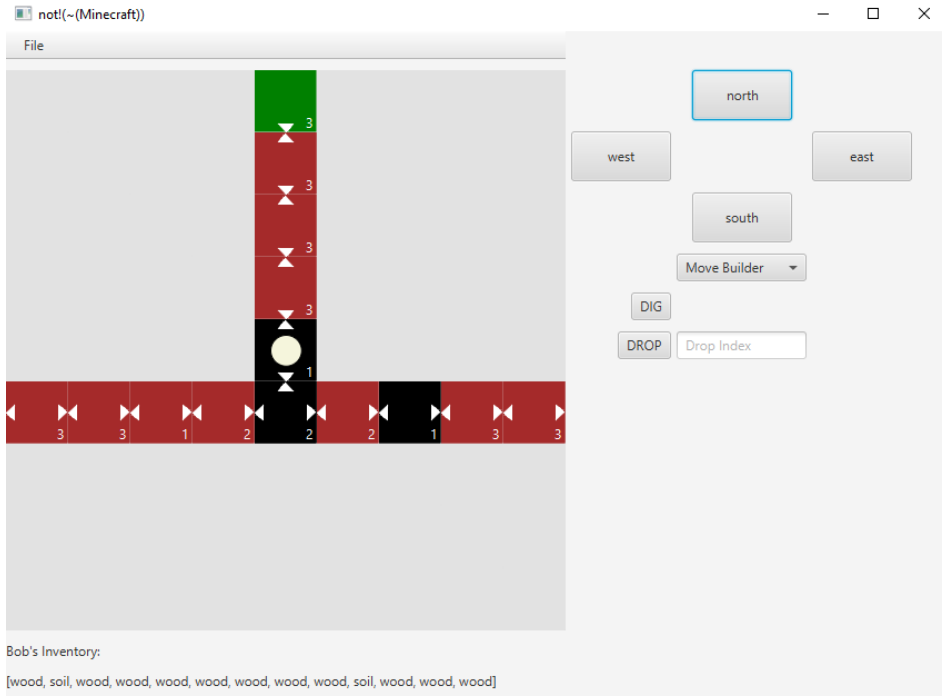
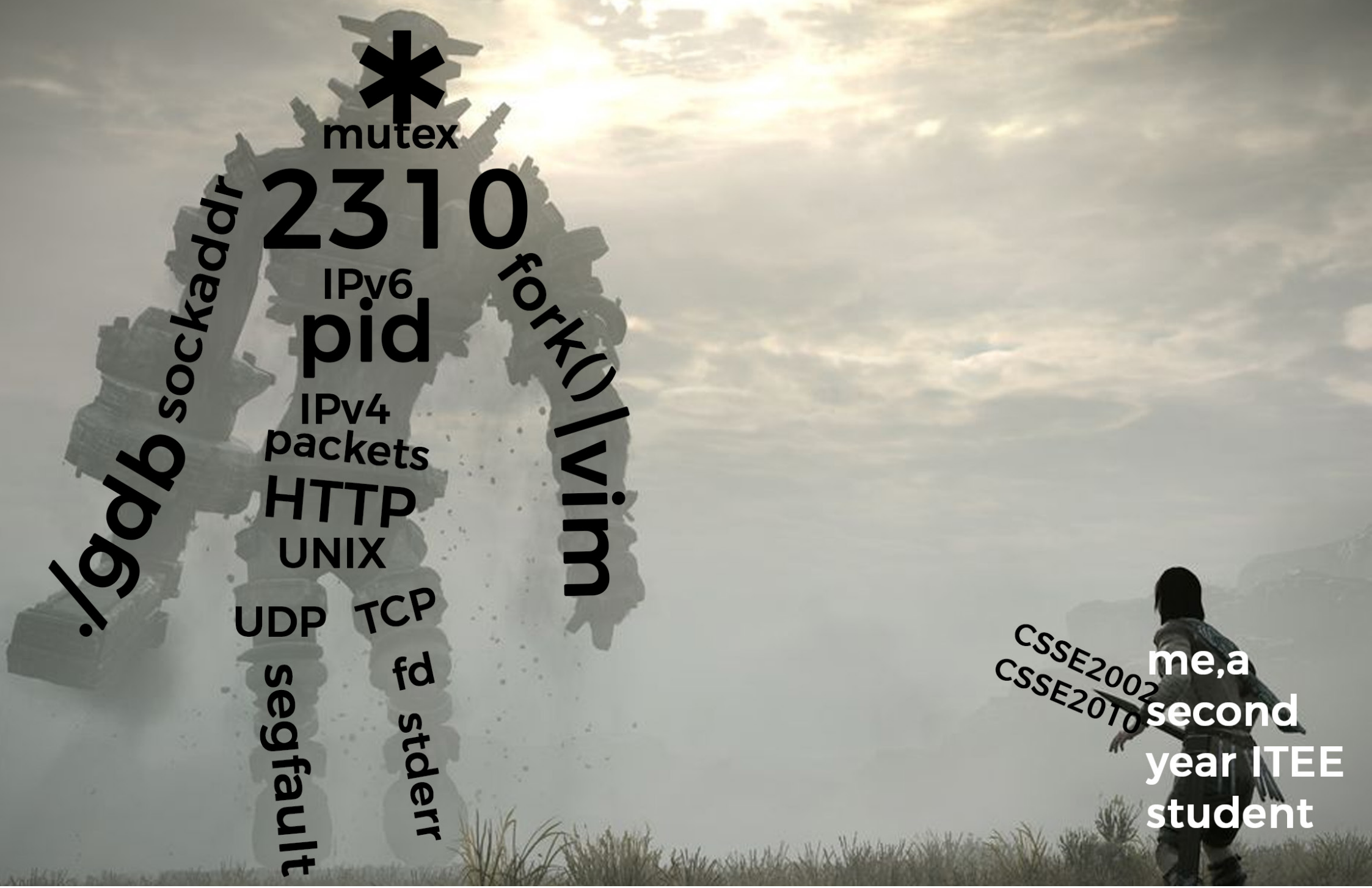


Figure 17 -Moving Map to the Bottom



Meme 6 - Thank you so much for reading my documentation, I'm really not looking forward to CSSE2310