

SPRINT 1 SOFTWARE ENGINEERING

Pratham Shah

Ramazan Dzhekshem

Welcome page

Welcome to the Bug World!

Start!

After pressing start button you are taken to the settings page

Bug World

Please upload a world map file:

Choose Files No file chosen

Please upload a bug assembler source code file #1:

Choose Files No file chosen

Please upload a bug assembler source code file #2:

Choose Files No file chosen

Please input the number of iterations:

Do you want to log the results of the session?

☐

Next

Have not implemented bug source code 1 or 2 or ticks and logs hence any input on these buttons won't make a difference however they do work and will make progress on them in the next sprint

Bug World

Please upload a world map file:

Choose Files No file chosen

Please upload a bug assembler source code file #1:

Choose Files No file chosen

Please upload a bug assembler source code file #2:

Choose Files No file chosen

Please input the number of iterations:

Do you want to log the results of the session?

Next

However the main implementation that I have done is for the world map file where I have implemented ability to upload files and parse them all on the client-side these are the instructions in the world map file

- # obstacle
- . empty cell (ie, no bug)
- empty cell, black swarm nest
- + empty cell, red swarm nest
- 1..9 empty cell with number of food units

A # in one cell input in the world map file will give an obstacle output that looks like this in it's respective cell



A . input in a cell will give you an empty cell that looks like this



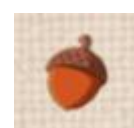
A - input in a cell will give you a black swarm nest which is indicated by a black dot that looks like this



A + input in a cell will give you a red swarm nest which is indicated by a red dot that looks like this



A 1...9 input in a cell which indicates food will give you a output that looks like this I was not able to implement number of food units but a number input in a particular cell position will give you a food output like the image above in that position



First I will show an example what successful test case looks like suppose I have the following input world map file which accepts any format(.txt, .world etc) doesn't matter as long as the instructions pass the error checks which I have worked on

```
10
10
# # # # # # # # # #
# 9 9 . . . . 3 3 #
# 9 # . - - - - - #
# . # - - - - - #
# . . 5 - - - - - #
# + + + + + 5 . . #
# + + + + + + # . #
# + + + + + . . 9 #
# 3 3 . . . . 9 9 #
# # # # # # # # # #
```

This is an example of a successful world map file and I will input this file on the following page like this

Bug World

Please upload a world map file:

Choose Files

Please upload a bug assembler source code file #1:

Choose Files

Please upload a bug assembler source code file #2:

Choose Files

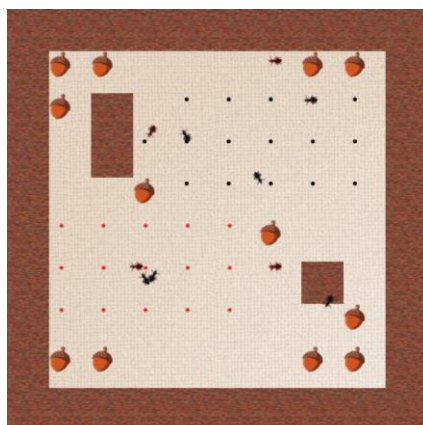
Please input the number of iterations:

Do you want to log the results of the session?

☒

Next

After pressing the next button we get the following output as it reads the instructions of the world map file all on the client side



Quit

As you can see all the instructions in the world map file are implemented as mention (Note: Have not implemented bug assembler the red and black bugs moving everywhere are just for display purposes which will be implemented ahead in a further sprint) After pressing quit button you are taken back to the welcome page

Now three error cases that I have implemented

Error case 1: incorrect dimensions

Example suppose I input the world map file below which clearly has incorrect dimension

```
10
12
# # # # # # # # # #
# 9 4 . . . . 3 3 #
# 9 # . . . . . #
# . # . . . . . #
# . . 5 . . . . #
# + + + + + 5 . . #
# + + + + + # . #
# + + + + + . # 9 #
# 3 3 . . . . 9 9 #
# # # # # # # # # #
```

After pressing next I will get the following error message

clabsql.clamv.jacobs-university.de says

Wrong Dimensions.

OK

Quit!

Error test case 2: border is not closed

Example suppose I input the world map file below which clearly has incorrect border

```
10
10
# # # # # # # # # #
# 9 9 . . . 3 3 #
# 9 # . - - - - - #
# . # - - - - - #
# . . 5 - - - - - #
# + + + + + . . #
# + + + + + # . #
# + + + + + . # 9 #
# 3 3 . . . 9 9 .
# # # # # # # # # #
```

After pressing next I will get the following error message

clabsql.clamv.jacobs-university.de says

Borders not closed.

OK

Quit!

Error test case 3: Invalid character in map

Example suppose I input the world map file below which clearly has an invalid character a

```
10
10
# # # # # # # # # #
# a 9 . . . . 3 3 #
# 9 # . - - - - - #
# . # - - - - - - #
# . . 9 - - - - - #
# + + + + + 5 . . #
# + + + + + + # . #
# + + + + + . # 9 #
# 3 3 . . . . 0 9 #
# # # # # # # # # #
```

After pressing next I will get the following error message

clabsql.clamv.jacobs-university.de says

Invalid character in the map

OK

Quit!

Any test case that does not defy the errors that I have worked on will be a successful test case and output a correct map file (there are other errors but these were the ones I worked on)

Link to the website: <http://clabsql.clamv.jacobs-university.de/~rdzhekshem/>