OBJECT ORIENTED ANALYSIS AND DESIGN

TCP 2201 PROJECT

TRIMESTER 2, 2019/2020

BY

GROUP 5

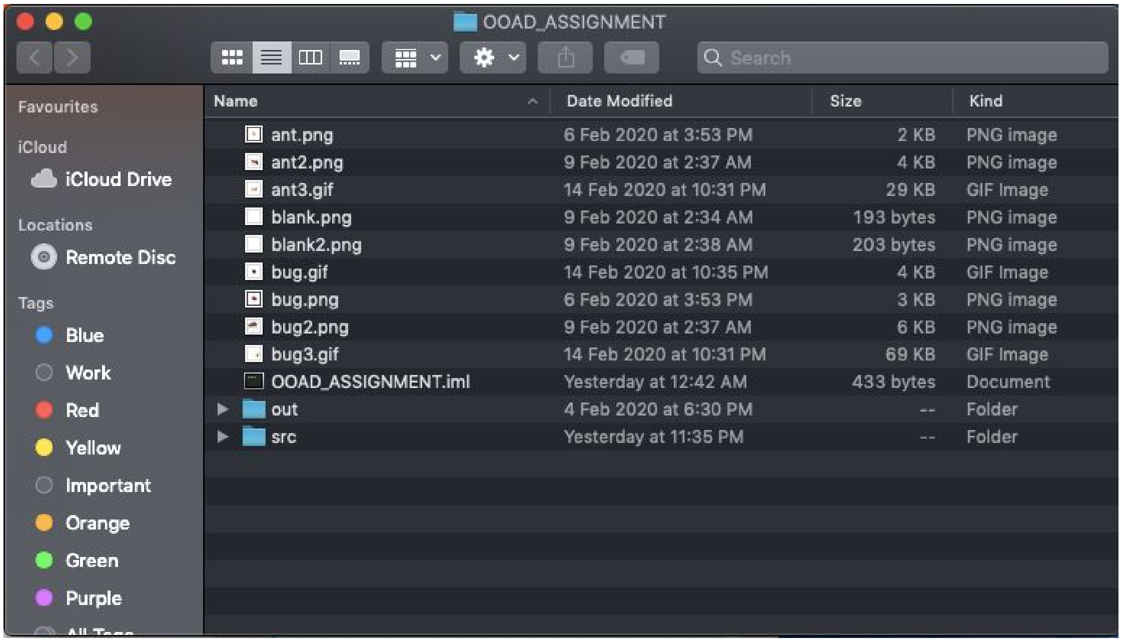
TEAM LEADER: MUHAMMAD AFHAM BIN MD AWAL-LUDIN

STUDENT ID	NAME	PHONE NUMBER	EMAIL
1161200646	MOHAMAD AMIRUL ARIFF BIN MOHAMAD YUSOF	011-65503506	1161200646@student.mmu.edu.my
1181302569	MUHAMMAD AFHAM BIN MD AWAL-LUDIN	017-4272800	1181302569@student.mmu.edu.my
1161202222	FATIN NABILAH BINTI MOHD SUHAIMI	019-2105168	1161202222@student.mmu.edu.my
1161101841	IZZAH NURHANA BINTI MOHAMMAD RASDAN	011-19710615	1161101841@student.mmu.edu.my

Table of Content

1.0 Instruction

Step 1: Ensure you have all the following files in your extracted folder.

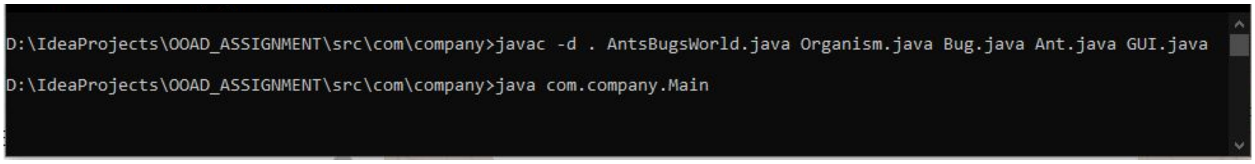


Name	Date Modified	Size	Kind
ant.png	6 Feb 2020 at 3:53 PM	2 KB	PNG image
ant2.png	9 Feb 2020 at 2:37 AM	4 KB	PNG image
ant3.gif	14 Feb 2020 at 10:31 PM	29 KB	GIF Image
blank.png	9 Feb 2020 at 2:34 AM	193 bytes	PNG image
blank2.png	9 Feb 2020 at 2:38 AM	203 bytes	PNG image
bug.gif	14 Feb 2020 at 10:35 PM	4 KB	GIF Image
bug.png	6 Feb 2020 at 3:53 PM	3 KB	PNG image
bug2.png	9 Feb 2020 at 2:37 AM	6 KB	PNG image
bug3.gif	14 Feb 2020 at 10:31 PM	69 KB	GIF Image
OOAD_ASSIGNMENT.iml	Yesterday at 12:42 AM	433 bytes	Document
out	4 Feb 2020 at 6:30 PM	--	Folder
src	Yesterday at 11:35 PM	--	Folder

Step 2: Open the Command Prompt and go to the folder path and proceed to type

D:\IdeaProjects\OOAD_ASSIGNMENT\src\com\company>**javac -d . AntsBugsWorld.java Organism.java Bug.java Ant.java GUI.java**

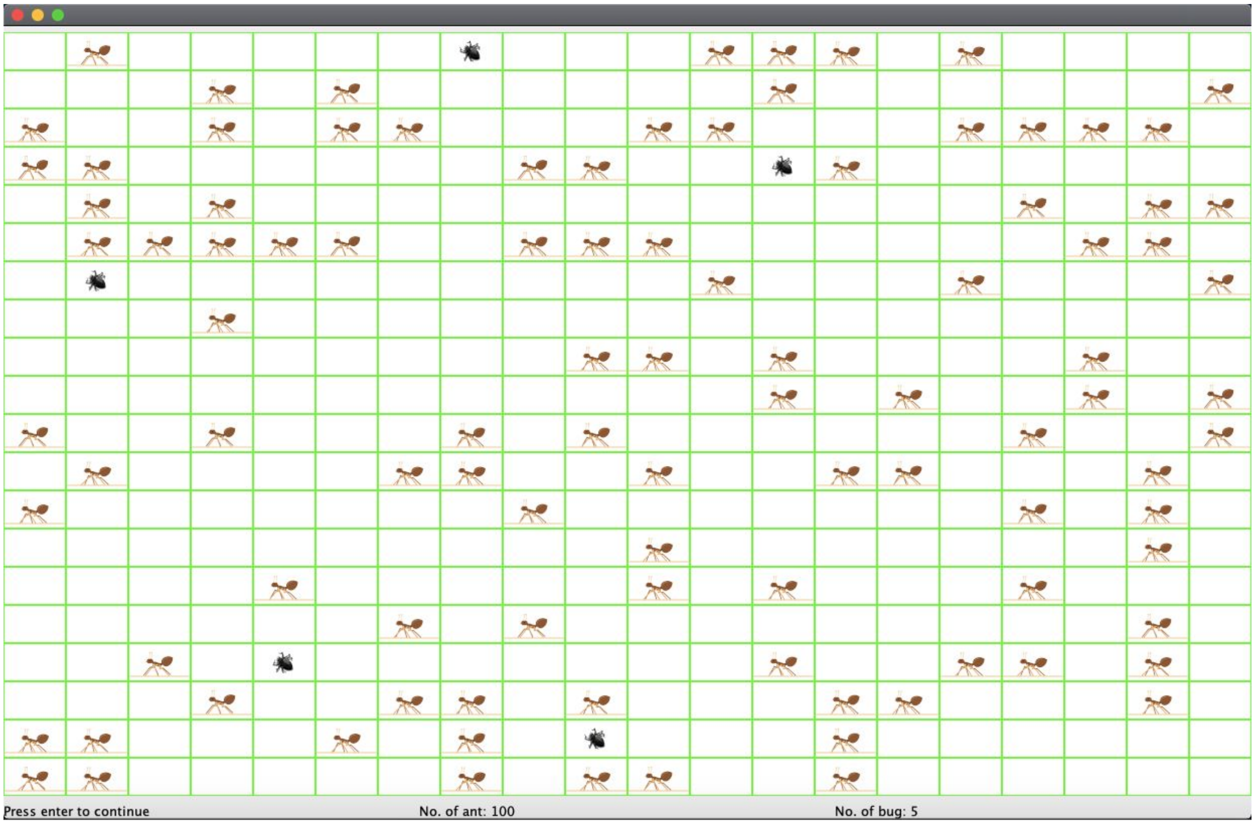
D:\IdeaProjects\OOAD_ASSIGNMENT\src\com\company>**java com.company.Main**



```
D:\IdeaProjects\OOAD_ASSIGNMENT\src\com\company>javac -d . AntsBugsWorld.java Organism.java Bug.java Ant.java GUI.java

D:\IdeaProjects\OOAD_ASSIGNMENT\src\com\company>java com.company.Main
```

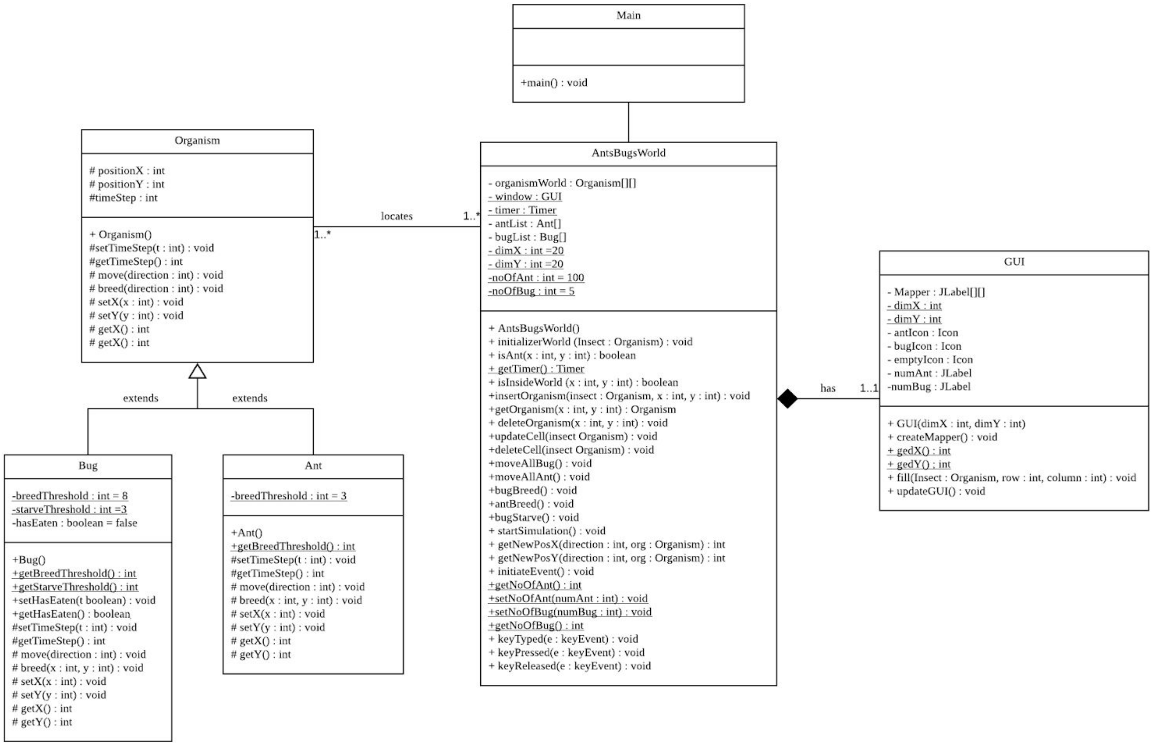
Step 3: The GUI will pop out. In order to start the simulation, you have to press enter. Repeat step 3 to continue the simulation.



```
Press enter to continue                    No. of ant: 100                    No. of bug: 5
```

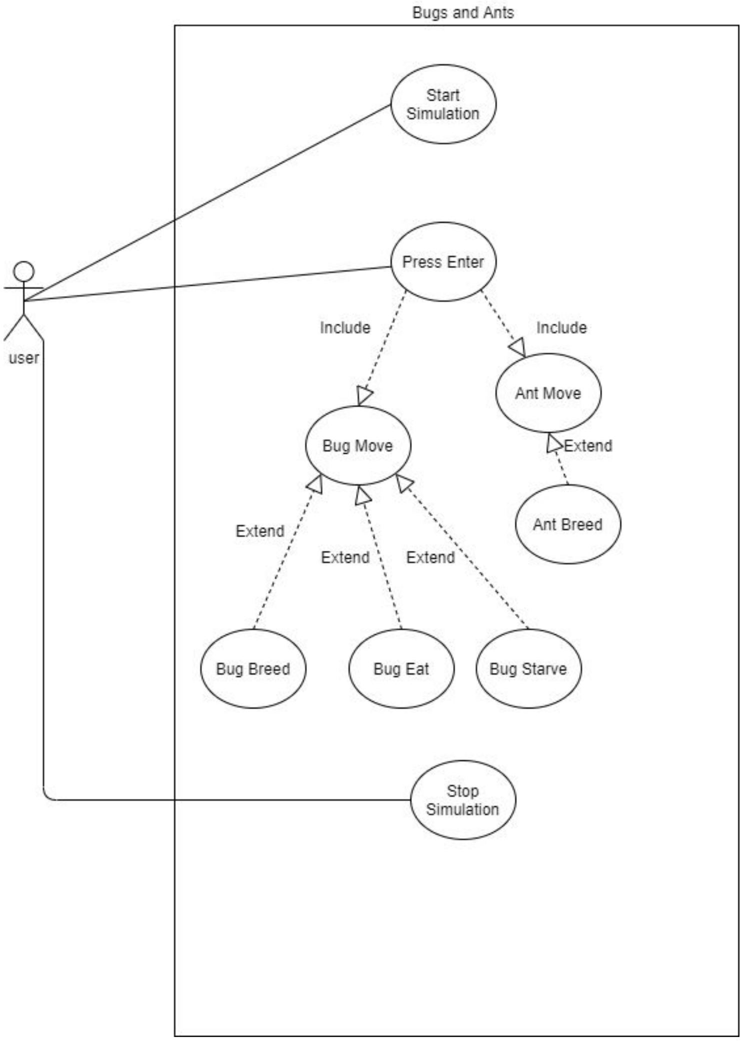
Step 4: To stop the simulation, click the red button on top left.

2.0 UML Class Diagram


Main
+main() : void
Organism
# positionX : int
# positionY : int
#timeStep : int
+ Organism()
#setTimeStep(t : int) : void
#getTimeStep() : int
# move(direction : int) : void
# breed(direction : int) : void
# setX(x : int) : void
# setY(y : int) : void
# getX() : int
# getX() : int
locates
1..*
1..*
AntsBugsWorld
- organismWorld : Organism[][]
- window : GUI
- timer : Timer
- antList : Ant[]
- bugList : Bug[]
- dimX : int =20
- dimY : int =20
-noOfAnt : int = 100
-noOfBug : int = 5
+ AntsBugsWorld()
+ initializerWorld (Insect : Organism) : void
+ isAnt(x : int, y : int) : boolean
+ getTimer() : Timer
+ isInsideWorld (x : int, y : int) : boolean
+insertOrganism(insect : Organism, x : int, y : int) : void
+getOrganism(x : int, y : int) : Organism
+ deleteOrganism(x : int, y : int) : void
+updateCell(insect Organism) : void
+deleteCell(insect Organism) : void
+moveAllBug() : void
+moveAllAnt() : void
+bugBreed() : void
+antBreed() : void
+bugStarve() : void
+ startSimulation() : void
+ getNewPosX(direction : int, org : Organism) : int
+ getNewPosY(direction : int, org : Organism) : int
+ initiateEvent() : void
+getNoOfAnt() : int
+setNoOfAnt(numAnt : int) : void
+setNoOfBug(numBug : int) : void
+getNoOfBug() : int
+ keyTyped(e : keyEvent) : void
+ keyPressed(e : keyEvent) : void
+ keyReleased(e : keyEvent) : void
has
1..1
GUI
- Mapper : JLabel[][]
- dimX : int
- dimY : int
- antIcon : Icon
- bugIcon : Icon
- emptyIcon : Icon
- numAnt : JLabel
-numBug : JLabel
+ GUI(dimX : int, dimY : int)
+ createMapper() : void
+ gedX() : int
+ gedY() ; int
+ fill(Insect : Organism, row : int, column : int) : void
+ updateGUI() : void
extends
extends
Bug
-breedThreshold : int = 8
-starveThreshold : int =3
-hasEaten : boolean = false
+Bug()
+getBreedThreshold() : int
+getStarveThreshold() : int
+setHasEaten(t boolean) : void
+getHasEaten() : boolean
#setTimeStep(t : int) : void
#getTimeStep() : int
# move(direction : int) : void
# breed(x : int, y : int) : void
# setX(x : int) : void
# setY(y : int) : void
# getX() : int
# getY() : int
Ant
-breedThreshold : int = 3
+Ant()
+getBreedThreshold() : int
#setTimeStep(t : int) : void
#getTimeStep() : int
# move(direction : int) : void
# breed(x : int, y : int) : void
# setX(x : int) : void
# setY(y : int) : void
# getX() : int
# getY() : int

The above diagram shows the UML class diagram that implements the MVC design pattern. The Organism class as the Model, the GUI class as the View while the AntsBugsWorld as the Controller. The Organism class extended by two other classes which are the Ant and Bug class that will have all the logic to update the controller if its data changes. While the GUI class will represent the visualization of the world grid cell of the simulation. The controller of the simulation which is the AntsBugsWorld will control the data flow in the model and view whenever the simulation starts. It keeps the view and model separate. The Main class ids the class will do the demonstration of the simulation once the user presses the enter button.

3.0 Use Case Diagram



Above is the use case diagram regarding Ants and Bugs Simulation. The main functions in the simulation are “Start Simulation”, “Press Enter”, and “Stop Simulation”. When the user presses enter, the following function must include to perform the function. The functions include “Bug Move” and “Ant Move”. As the “Bug Move” and “Ant Move” functions perform, there will be optional functions (extend) that might need to perform. Therefore, the extended use cases for “Bug Move” are “Bug Breed”, “Bug Eat”, and “Bug Starve”. As for “Ant Move” there will be one extended use case which is “Ant Breed”.

4.0 Sequence Diagram

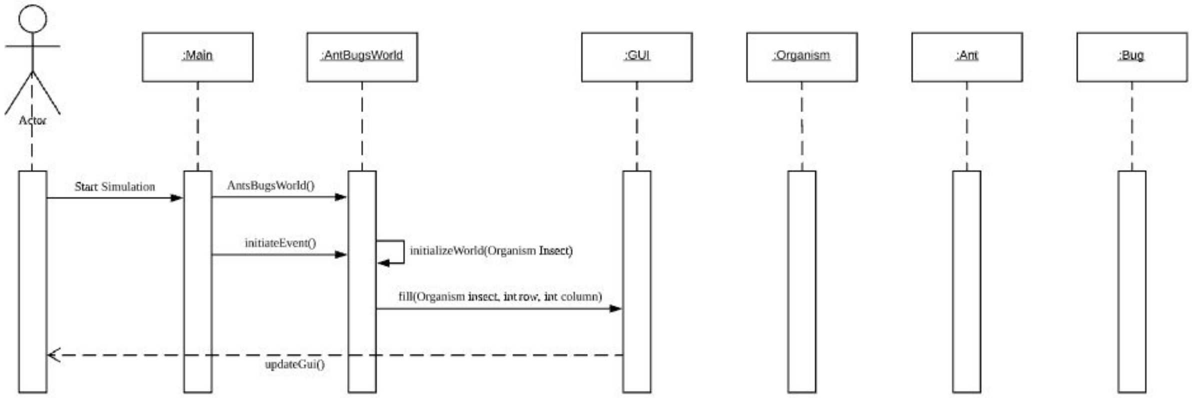


Diagram above is a sequence diagram for starting simulation.

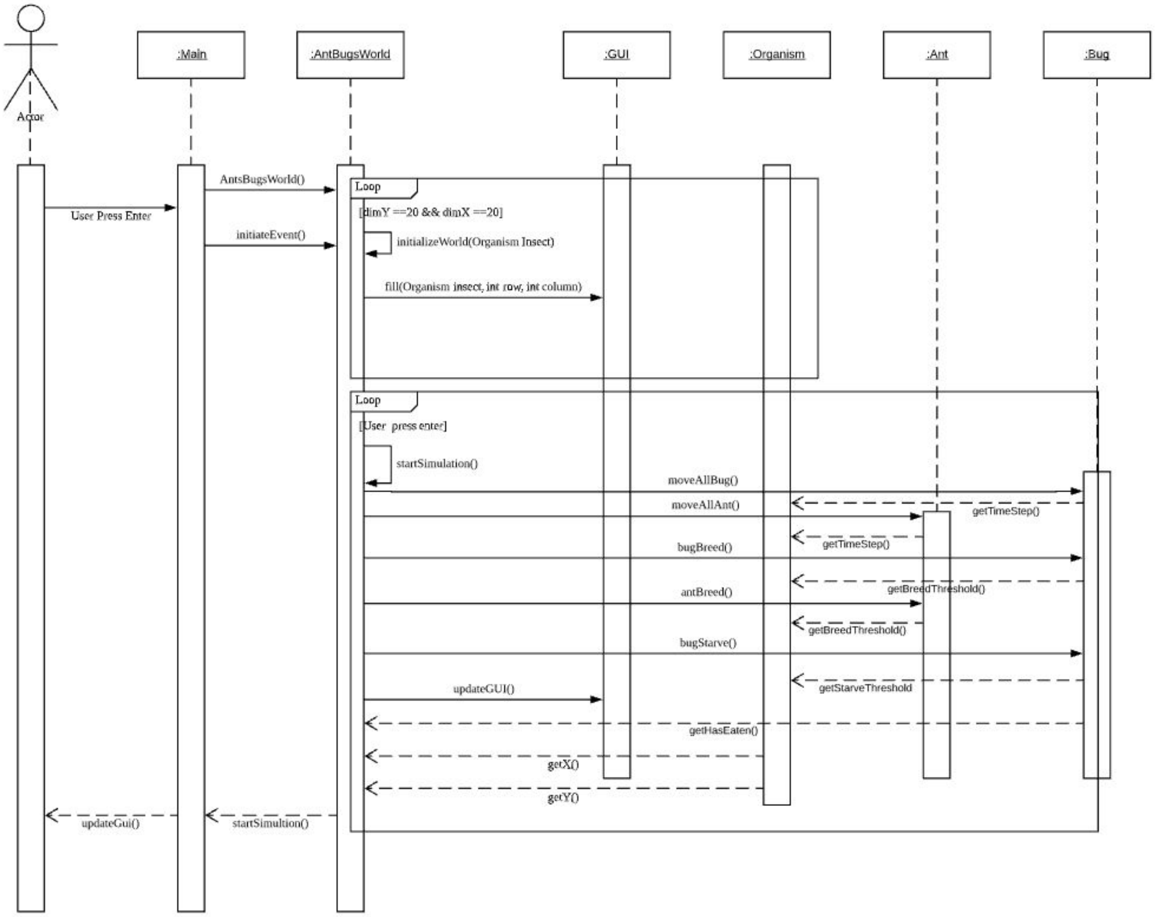


Diagram above is a sequence diagram for user press enter

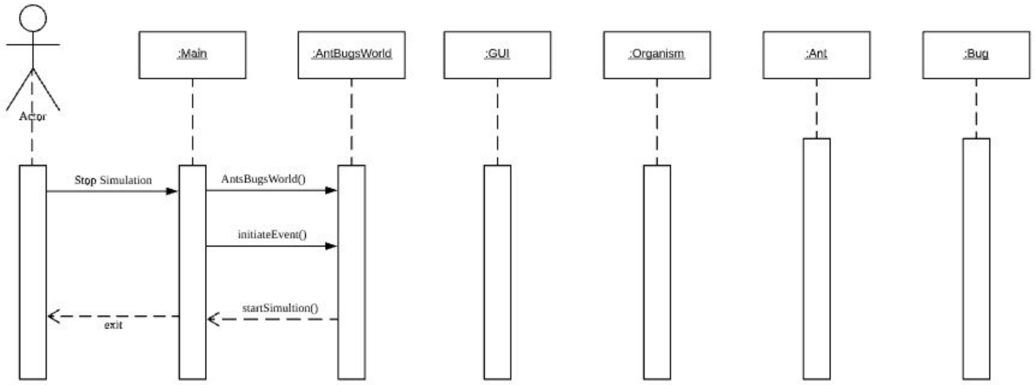


Diagram above is a sequence diagram for stop simulation

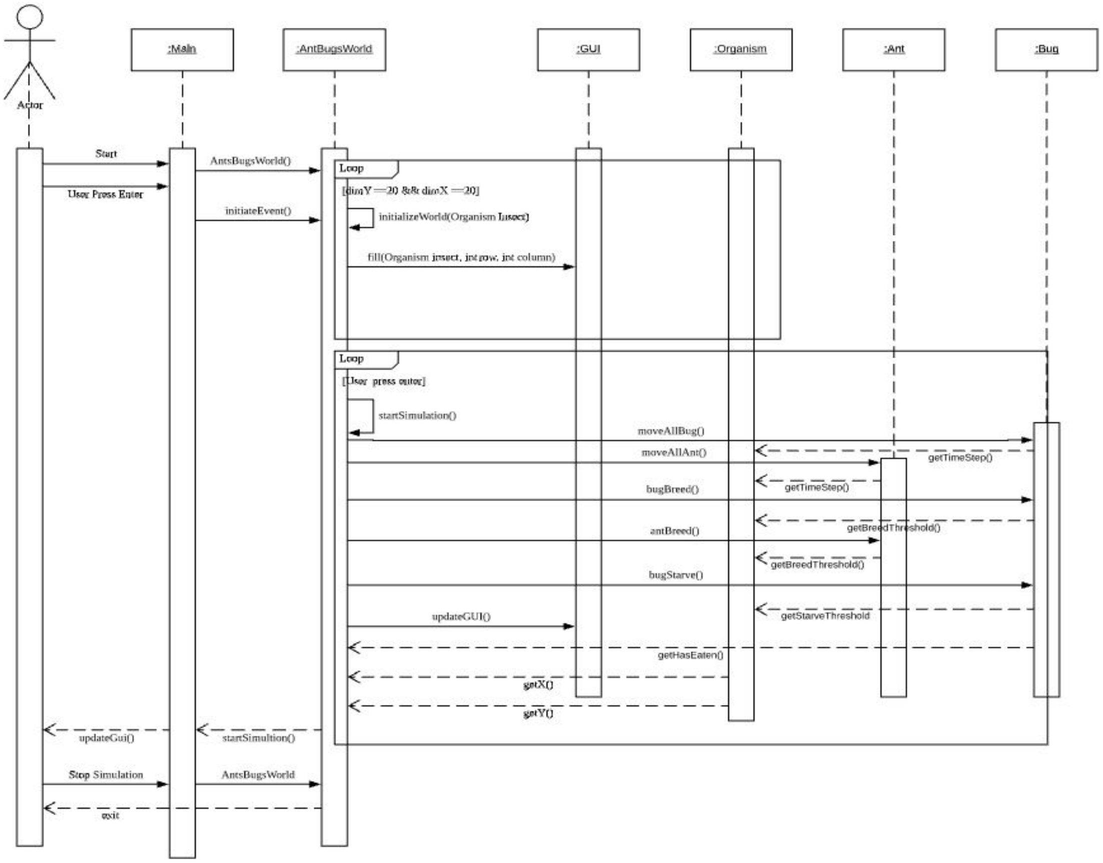


Diagram above is a sequence diagram for overall simulation

5.0 Project Evaluation

TCP2201 Project Evaluation Form (30%)

Tutorial Section	TT02
Group Name	Group 5
Team Leader	Muhammad Afham
Member Name	Amirul Ariff
Member Name	Fatin Nabilah
Member Name	Izzah Nurhana

Prototype and Presentation (20%)

Item	Maximum marks	Actual marks
Comments, indentation, following proper Java naming conventions, other Java style issues.	2	
Object-oriented concepts like subclassing, delegation, composition, aggregation, polymorphism, etc.	3	
Appropriate use at least one Design Pattern.	3	
User friendliness and appropriate GUI components used.	4	
Functional requirements fulfilled.	8	
Total	20	

Report (10%)

Item	Maximum Marks	Actual Marks
UML Class Diagram done and is coherent with the implementation	3	
Use Case Diagram done and is coherent with the implementation	2	
Sequence Diagrams for each use case done and is coherent with the implementation	3	
User Documentation done and is coherent with the implementation	2	
Total	10	

Note: Individual marks will be adjusted after the lecturer interviews you, based on how much work each person did.