Richard Jacobs - 170083964
Thanuja Jegatheesan – 190030250
Bolanle Oyetade - 180014367
Grace Sinclair - 190127985

Group 6

Coursework

Design Description

1. **Introduction**
   This design document for the Party Simulation will provide a detailed design description which will show how the requirements of the project will be met. This will include the user interface, the functionality, and the milestones.
2. **Problem**
   In this project we aim to develop a simulation of a party. The party is attended by hosts and guests. Guests can either be artists, scientist, or engineers. The room in which the party takes place is represented as two-dimensional grid(field). During this our team will intend for each person at the party to occupy a location in the room. Hosts and guests will move about in the room, socialising with different people, Guests have a certain preference as to which types of people they find interesting to socialise with.
3. **Scope**
   The scope of this design document is to provide information of the design procedure of the system. The design constraints, user interface, the functionality and the milestones. Also, the data design will be included. The intended audience is the ones who will be using the party simulation in the project and the marker; hence the document is made to be a guideline for us developers and the marker.

4. **Definitions, Acronyms and Abbreviations.**

Artist	Artists are in the room. Artists chat only to other artists
Engineer	Engineers are in the room. Engineers like all other engineers, scientists, and artists
Guests	Superclass for the people in the room
GUI	Graphical User Interface
Host	Host are in the room
Model Constant	Model constants stores constants to be shared across multiple locations
Person	Person is an abstract class which sets location
Scientist	Scientists are in the room. Scientists like other scientists and artists.
Simulator	Simulator is responsible for initialising, controlling, and executing the simulation

5. **Design Assumption**
   When designing the party simulation, we had to make some assumptions related to the software, hardware was not too much of our concern as it will run on most PC's and Mac OS. Our program is running on most computers that can run java ide.

6. **Time Constraints/delegation**
   The design and the implementation part of the project should be finished by the first week of December around the 3RD of December with a final product which should be finished. To make sure we finish before/on our deadline all group members should follow the waterfall method that we have chosen to follow. To fulfil the requirements, we will split the work amongst our groups, so that we can meet the deliverables. Richard will be writing up a detailed design description, which will show how the requirements of the project will be met. This will include the user interface, the functionality, and milestones. Bola will be responsible for creating the class diagrams, where the relationships between the classes, along with their fields and methods, will be shown. We already

made a basic class diagram before beginning the coding part of the project, nut these will be much more detailed. The implementation will be done collaboratively using blackboard collaborate, where we can code together. This has been split up by the different labs set on blackboard and one of these are completed each week so that we can meet our deadline. So far, we have completed two of these labs and are on track to complete the others. Thanuja will test the program and provide the experimental results. Grace will write up the team dynamics narrative and document the way in which the work was carried out by the individual members of the group and how they contributed.

7. User interface

In this program there are three buttons: Start simulation, end simulation user input. They are placed underneath the simulation using a layout manager.

The start simulation button is a JButton which starts the simulation.

The end simulation is a JButton which ends the simulation

Then there are buttons that allows the user to input information so that they can set:

- the creation probability for each type of person, at the start of the simulation
- the seed for the random number generator
- the distance (n) up to which each guest may look to find a person they would like to socialise with
- the dimensions of the party room.
- the number of steps that the simulation should be run for.

8. Functionality

How the simulation works

In the beginning of the simulation, the party room is populated with people according to a certain type of creation probability given to them entered by the user. The creation probability will signify the likelihood of a person of that type to occupy a certain location in the room.

The hosts and guests will move around the room engaging n conversations with people next to them. Guests will have certain preferences and will move to new location based off those. There are certain socialising preferences with each person in the room. Hosts have no socialising preferences and all guests (artist, engineer, scientists) find host interesting. Artists find other artists interesting and will only talk to them, Scientists like other Scientists and artists and engineers like all other engineers, scientist, and artists.

The happiness level of each guests would be determined by the number of interesting people which surround them.

9. Milestones

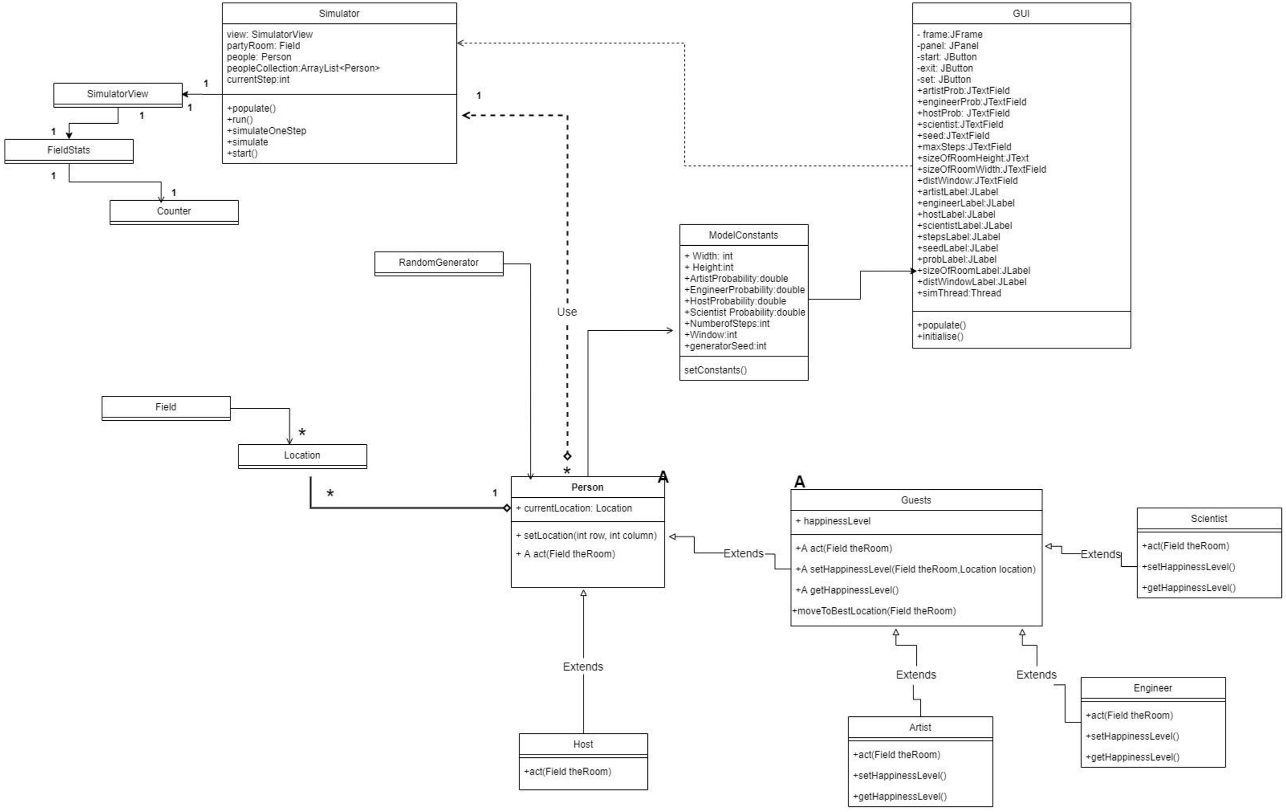
Start of Part Simulation, create a controlling class for the simulation with a class hierarchy – 02/11/20

Enhancing Simulator class and adding functionality to the people at the part – 08/11/20

Identifying the need to add Abstract classes and methods working on the host and guests socialising patterns -22/11/20

Creating components and arranging them using a layout manager while making them responsive to the user inputs while enhancing how the party simulations run by creating GUI for the simulation – 01/12/20

Class Diagram



Testing

Effect of creation probability:

The higher the creation probability of each person the more people there are in the room therefore less empty space in the party room. While changing the probabilities of all people I have found that when the guests creation probabilities are high compared to the hosts probabilities, there is more movement of guests than hosts as the hosts have less space to move to. However, when the host probability is higher than total guest probability guests still move around just as much as the hosts do.

Effect of distance n:

The distance n, the maximum distance a person can move away from their current position, affects how fast the people move around in the simulation. The lower the value of n the faster the people move around the simulation, this can be because there are less spaces to be checked before moving which reduces the time taken for a person to be moved. When the range of movement is lower, there is more space between people as they move around which separates the larger groups of people into smaller ones, meaning more interactions. This allows the attendees to meet more types of people other than their own types and have more intimate interactions in smaller groups.

Effect of different room sizes:

The size of the room affects the length of the party simulation. When the dimensions are smaller, the party room is smaller, as there are less spaces to be checked before moving each person this speeds up the simulation making the length of the simulation faster than if the dimensions of the party are bigger. This also affects the speed of the movement of the people around the party. When the dimension sizes are lower the people move around the party at a faster pace than when there are smaller dimensions.

What happens if only one type of guest attends the party?

When only one type of guest attends, all the guests group together and move in the groups together around the room and join other groups of the guests. The hosts move around individually joining the groups of guests or other individual or small groups of hosts.

What happens if only two types of guest attend the party?

If two types of guests attend the result is similar to one guest type attending. Each type of guest moves around in a group of themselves and join with other groups of a type of guest, while all moving around together. The hosts move around individually while joining different groups of guests or other individual hosts.

What happens if there are no hosts at the party?

When there are no hosts the guests move around in big groups of their own type joined with other groups which are also of one type. They all move around in a few big groups of guests.

What purpose do hosts serve?

The hosts help to break these groups down into smaller groups. They ensure that the guests are interacting with more people at the party rather than sticking in big groups of their own type. This means that they are meeting more people that are not just of their type and moving around the party more.

How does distance n affect the number and size of clusters that the guests form?

As n increases the size of the clusters increases which decreases the number of clusters there are. When n is lower the guests interact in a higher number of smaller groups scattered around the party.

Team Dynamics Narrative

How the work was distributed

During our first meeting as a team, the work for this project was distributed and each member of the team was given tasks to complete. For the report part of the project, there were four main aspects that had to be completed.

Richard was given the job of writing up the design description, which clearly shows how the requirements outlined in the problem description will be met.

Bolanle had to create a detailed class diagram in order to clearly outlining the classes that had been implemented in the code.

Thanuja was put in charge of testing the code and she did this by using different inputs and recording what the program outputted, she then recorded these experimental results by answering different questions about how the program reacted in different situations.

Grace wrote the team dynamics narrative, which is used to document how the individual members of the team carried out their tasks.

For the coding element of this project, both Grace and Thanuja worked together to make the code that populated the party and handled how each person could move around the room based on their socialising preferences. They did this by creating the Simulation class and the ModelConstants class.

The ModelConstants class was later adjusted, as Bola was put in charge of how the program would process input data. She created a graphical user interface with the help of Richard by combining buttons, labels, and text boxes, where the user could set the model constants by inputting their own numbers for the variables. The exit button allowed the user to stop the simulation from running and the start button initiated the simulation.

Whilst coding, everyone added comments to explain the functionality of each aspect of the code, which will be very helpful to any developers in the future, who need to add or adjust to the code we have already provided.

How much each member of the team contributed

Thanuja contributed 25% of this project by completing the first three labs worth of code with Grace and then writing up the results of the testing.

Grace contributed 25% of this project by completing the first three labs worth of code with Thanuja and writing up the team dynamics narrative.

Richard contributed 25% of this project by helping Bola in the final lab as well as writing up the design description.

Bolanle contributed 25% of this project by completing the last lab of the code and creating the class diagram.

What have we learned?

Thanuja:

Whilst completing the implementation of the simulation I've become more confident in using the NetBeans integrated development environment. I have learned how to use the debugger to step through the code and find errors. I have also developed my understanding of how to use up casting and down casting in Java while creating the actors in the party simulation.

Grace:

I have learnt the purpose of abstract classes, as well as how to show them in a class diagram and how to implement them in code. Whilst populating the simulation with different types of people, I learned how to create a ModelConstants class and use the variables in this. Finally, I have learnt how to import classes from other packages, as well as classes the java library, such as the random class.

Richard:

What I have learned in this experience is the use of file sharing through the means of GitHub I’ve also learnt the use of abstract classes and learning how to implement our work with components to make the GUI for the simulation so it’s more easier for the user to navigate the simulation.

Bolanle:

During this project I learnt the importance of the use of static and final when creating a user input simulation. I also learnt the use of inheritance in a java package as well. When creating the GUI I also learnt about the use of threads and how it can bring different classes in the package together to work as one.