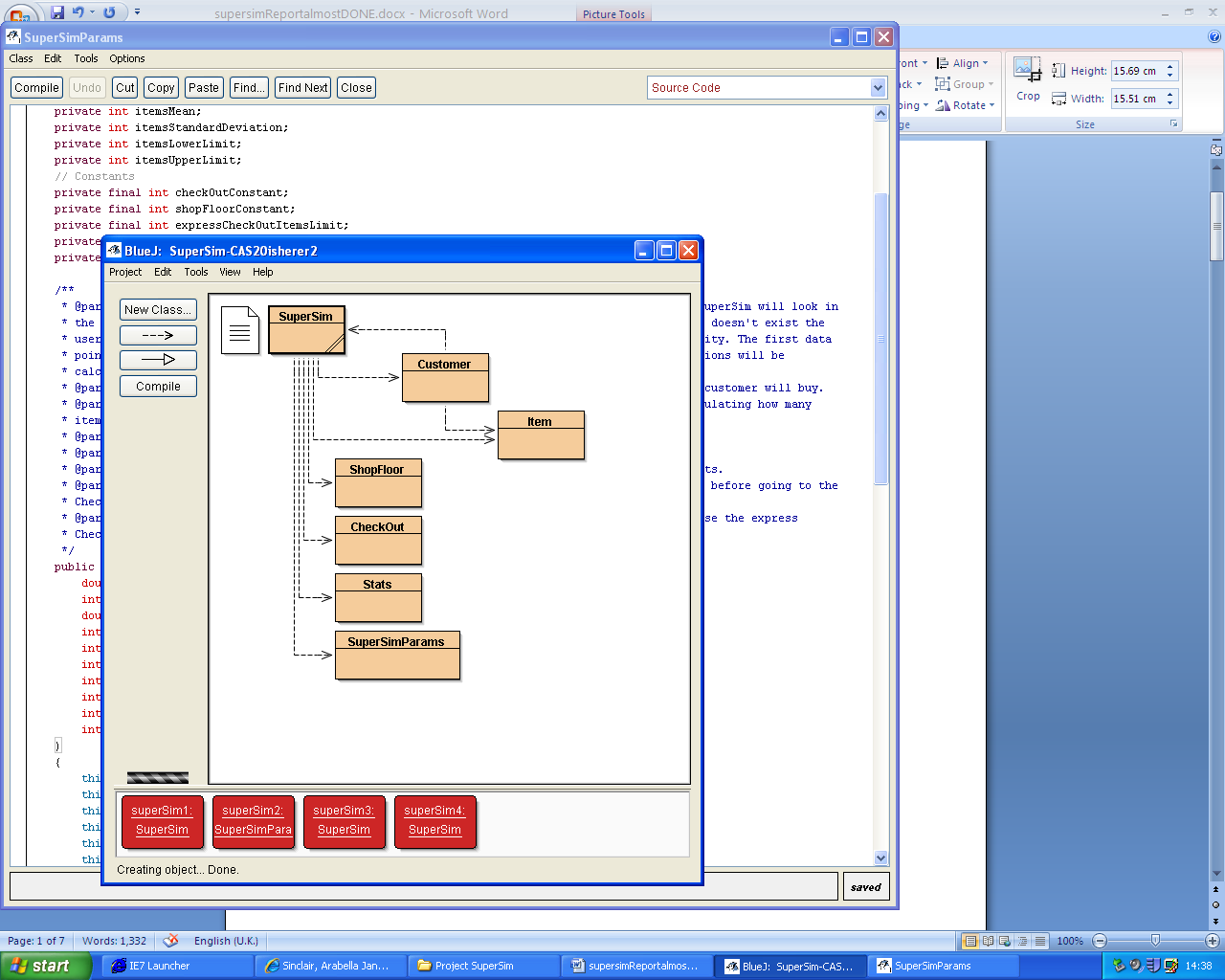
**SuperSim Project Report**

Group members: Fergal Hainey: 50907479

Yowanan Kuvaka 50905917

Chiedu Agborh 50907438

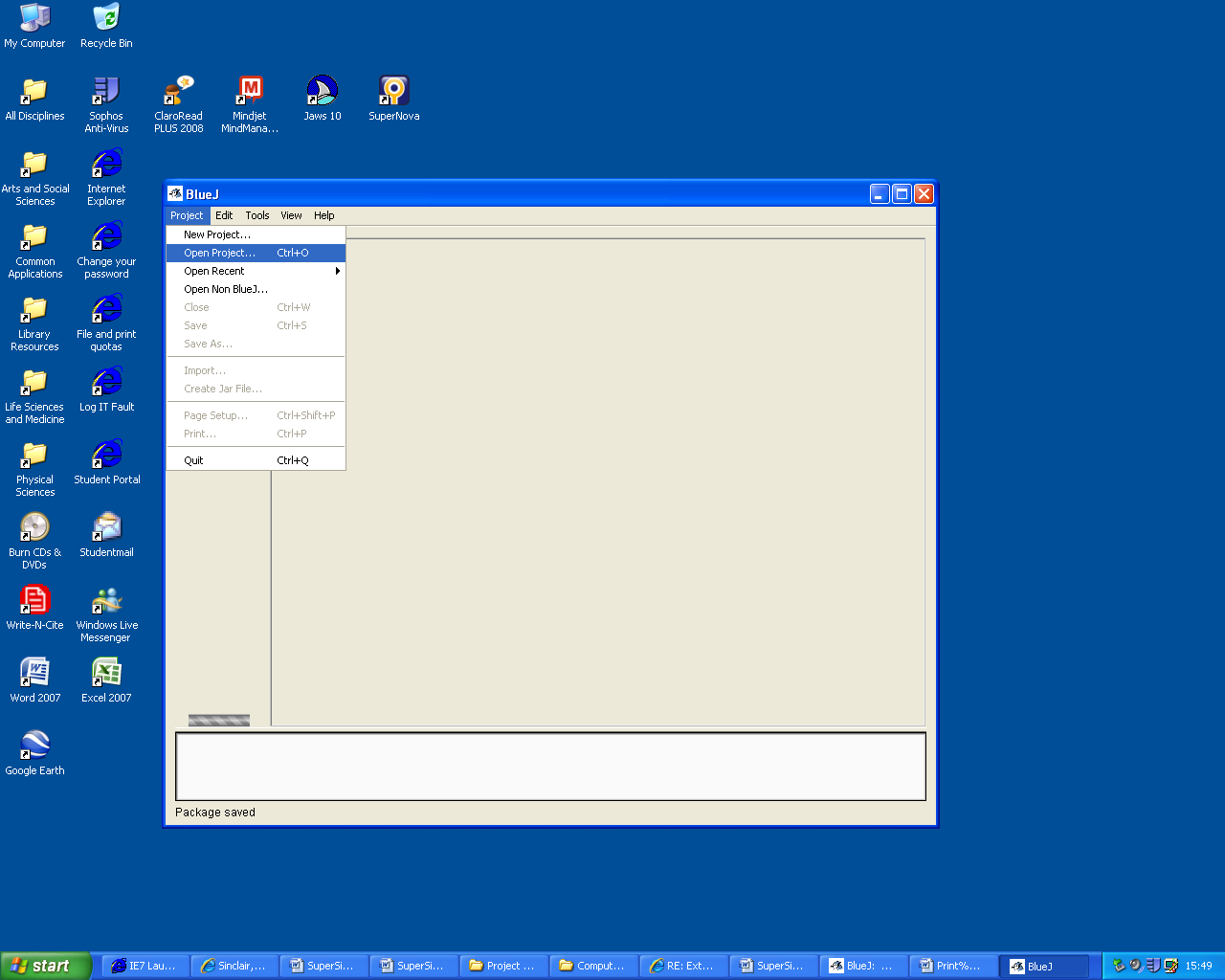
Arabella Jane Sinclair 50900659

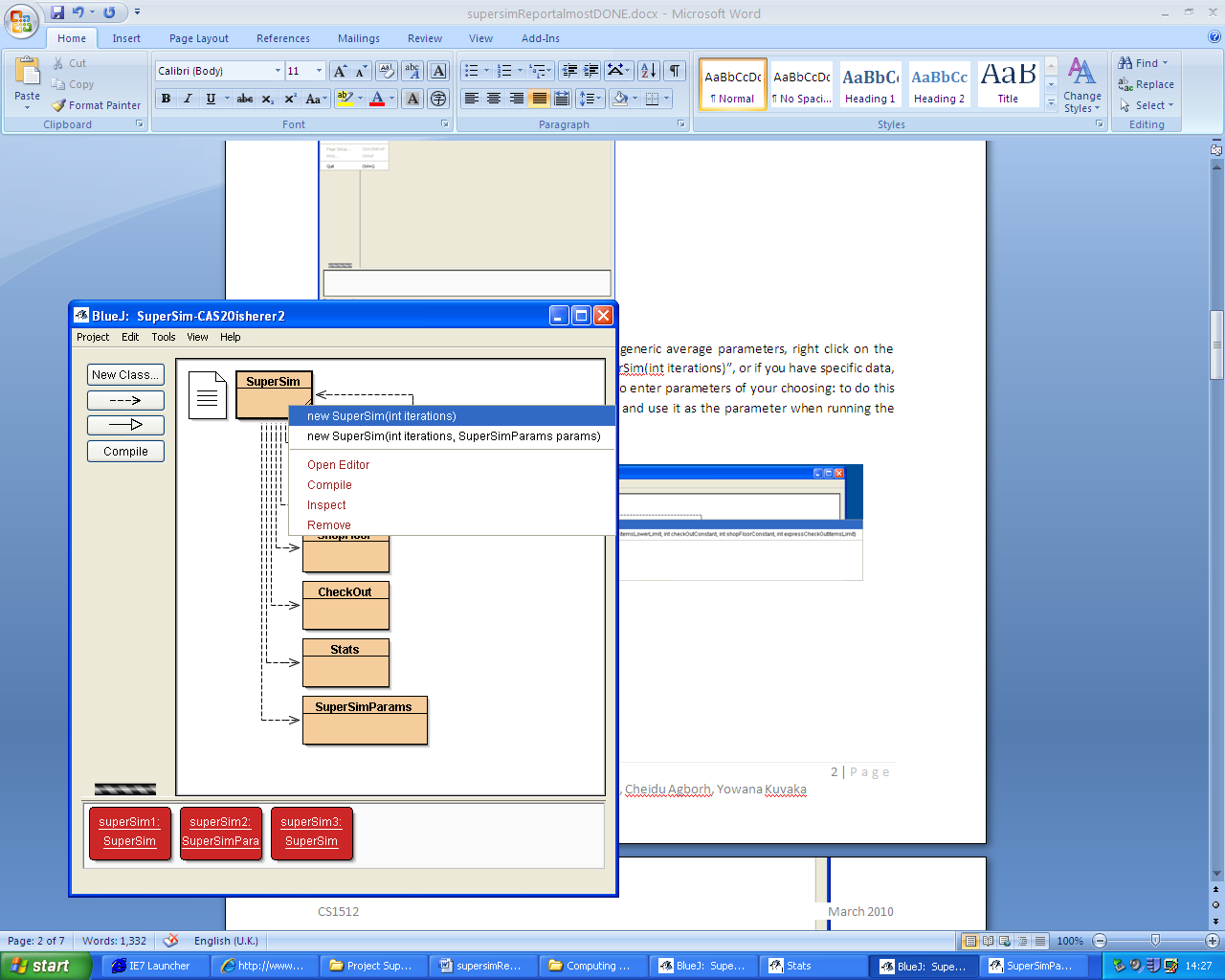


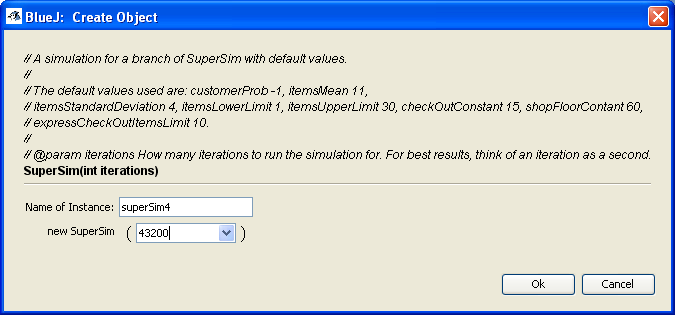
We acknowledge that this exercise is part of the formal assessment of the course, and declare that the work done is our own. Any contribution made by anyone outside the group is acknowledged as indicated in the section on Cheating & Plagiarism in the student handbook.

User Manual

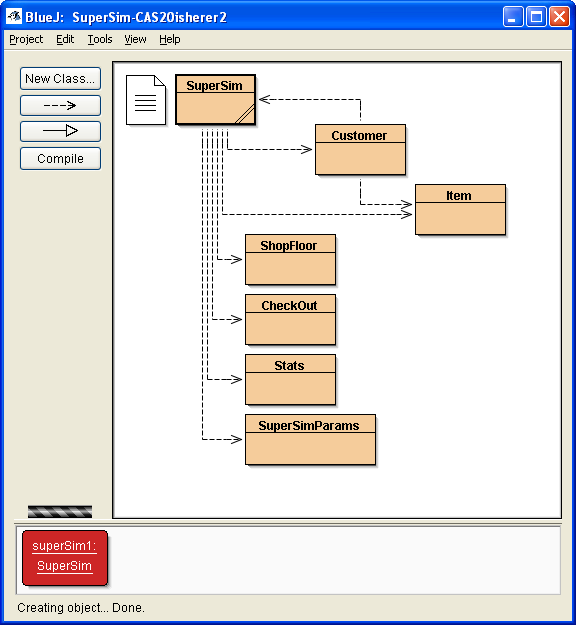
The SuperSim program simulates the operation of a supermarket and checkouts for a period specified by the user. At the end of the period the simulation runs for, the program will output a series of run-time statistics. The user has the option to use the supplied sample times for various customer processes within the simulation or they can enter their own details to tailor the stats to a specific supermarket setting.

To Run the Supermarket Simulation program first open the BlueJ environment and load/open the SuperSim project file.

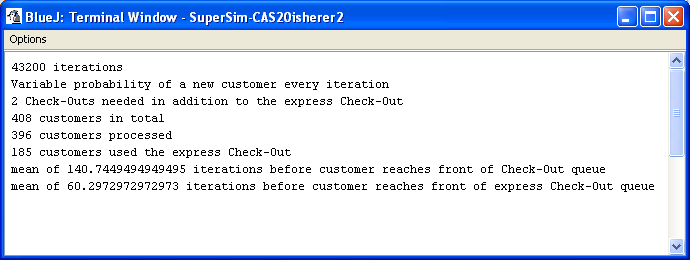
To run a simulation of a supermarket using some generic average parameters, right click on the SuperSim class and select from the menu “new SuperSim(int iterations)”, or if you have specific data, then choose the option below this that allows you to enter parameters of your choosing: to do this create a new instance of the SuperSimParams class and use it as the parameter when running the main simulation in the manner above.



The number 43200 used here (above) is the number of seconds the user wishes to run the simulation for (equates to 12 hours if an iteration is considered to be a second, which fits best with the default settings). Now you have an instance of this class which you can print statistical data about: this will return a list of details about the running of the simulation:



Simulation Information



This Summary stats information outlines the details of the simulation, iterations standing for seconds of time passing during the simulation. The probability of a new customer arriving every second varies according to the time of day inputted through the csd file.

To make a record of the customers processed and the details of their purchases in the store, select the “void writeLoyalty()” option on the menu, which will come up with a window asking you to select the file you wish to write the data to.

Maintenance Manual

See JavaDoc for more details:

Classes:

SuperSim

Methods:

Constructor

Determine stats, initialise variables

Populate stock list – hash map of Items

One Iteration:  
 Move customers out of queues

* Method to check customer departing checkout

Close checkout

* If the mean number of people in a checkout is less than a constant and there is a checkout empty.

Move customers from shop floor to checkouts, first checking which checkout to move to.

* ArrayList is filled with customers to be moved
* Loop: for each customer on the list, check if they qualify for processing in the express checkout: if not, then add them to the shortest queue.
* If all the queues are filled to their max limit then open a new checkout.

Check if there is a new customer

* Method: add the new customer to the shop floor

Start processing customer at front of queue if necessary

* For each checkout, start processing the customer at the front of the queue if this is their first iteration at the front of the queue.

Get statistics

Update/increment data variables storing customer/shopfloor details

Print info

Print all the stats/variables

Write loyalty

Read all the customer/transaction details from the Hash map

Write to a user-chosen text file

Customer

Constructor

Method make ID

Record when arrived in shopfloor – through SuperSim

Method calculate number of items

Calculate when customers leave shopfloor using a constant

Create a shopping basket Array list of Items

Method to fill the basket

Calculate number of items

Get parameter means and standard deviations from SuperSim

Calculate limits

Generate a random number using Gaussian standard distribution between the two limiting parameters

Fill Basket

Arraylist of Barcodes

Add Items to Basket, these are indexed randomly by barcodes from SuperSm

Set unique customer number

Increment static ID counter

Process

Calculate arrive and depart times using the SuperSim iterations

SuperSimParams

Saves the user-input parameters in a class so that they can be entered on multiple simulations

* Has to be used when not using default parameters: making an instance of this class and passing it as the second parameter to the SuperSim main constructor

Stats

Constructor –used internally

Takes and stores multiple parameters

Item

Constructor

Parameters: name, price, barcode: initialise

ShopFloor

Extends Array list

CheckOut

Extends Array list

Simulation

Stats assumptions: for the time values given we estimated sample data and parameterised all the inputs so that the eventual program could be used in various different settings and would not be dependant on certain data.

A Gaussian Standard Distribution function was used to determine the number of items selected by a customer on entering the store: the mean and limit values are parameters so again they are flexible to situation.

The default parameters for the simulation are:

customerProb: (0.5/60.0)

itemsMean: 11

itemsStandardDeviation: 4

itemsLowerLimit: 1

itemsUpperLimit: 30

checkOutConstant: 15

shopFloorConstant: 16

expressCheckOutItemsLimit: 10

Group Activities

Meeting Log

9th March

1st meeting: discussion of classes, planning and roles within the group

10th March

Group discussion and coding of solution up to CAS 11

16th March

Allocation and planning of up to CAS 14

17th March

Combining individual solutions from last meeting and planning CAS 17

22nd March

Meeting to discuss CAS 17 part-coded solutions

23rd March

Finishing CAS17 code and writing of project report

24th March

CAS20 writing and planning, finalisation of Report

25th March

Tweaking

Research

For the research stage of the project we used our combined knowledge and experience of supermarket operation to estimate a sample average time for various tasks. To allow for flexibility in different situations, we didn’t place much importance on exact statistics for a certain supermarket, rather we created the program to operate around input parameters so that the result could be used to accurately simulate any particular supermarket system from a small convenience store to a large family super-store.

Problem Breakdown

The problem proposed was already broken into 4 main stages separated by the CAS marks awarded. We decided that the problem should therefore be tackled in these stages, and arranged our meetings around breaking down each stage, allocating the parts within the group then coding the individual solutions. The following meeting would see the combination of the parts, debugging and then following the same process for the next stage.

For up to CAS 11 the solution was discussed and coded as a group working together through one console. For up to CAS 14 the problem was discussed and planned as a group then split to be worked on for next team meeting. For up to CAS 17 again problem discussed and allocated. For up to CAS20 the coding was done by the most competent member, the rest of the group taking part in the discussion of various functionality aspects.

Team work

Within the team most of the main coding and planning issues were discussed as a group and agreed together. Our varying programming abilities were taken into account when allocating the tasks.

Tasks:

🡪CAS11 Group: Janie, Fergal, Chiedu and Yowana

🡪CAS14 Express Checkout – Chiedu

Shop floor Constant –Yowana

Create Shopfloor and move Customers in and out based on iterations passed – Janie

🡪CAS17 Text File –Yowana

Read file; assign Items randomly to a Customer –Janie

Calculate and write the money spent, customer details and items in basket to a text file –Chiedu and Fergal

Static customer number field incremented on each new customer – Fergal

🡪CAS20 The simulation runs over an extended period of time (12 hours) with the probability of a customer arriving at the store varying according to the time of day.

Stats and Debugging – Fergal

🡪Report User Manual –Janie, screenshots: Yowana

Maintenance Manual –Janie

Simulation Details –Janie

Group Activities Record –Janie

🡪JavaDoc documentation –Fergal

Future additions:

Running multiple instances of SuperSim simultaneously to take mean

Use above, specify input value to be changed, monitor an output stat, then write a file showing how certain inputs affect certain outputs, this file would have the potential to be used in Excel – Making the Program Useful in a real life context.