**Daily Diary for Computer Challenges Module**

**11/01/2022**

* Received the email from John that confirms my first choice of the industry simulation project

“F.1.i Industry from Data Analysis course. A number of industries have been examined in detail, including interviewing experts from the industry, to identify what are the most important positive and negative events that occur within them. Also, data sources that provide real metrics that can be used to quantify the simulations are also available. These include Farming, Youtuber/Social media creator and many more.”

* John sent a few folders with an example simulation of a coffee shop which isn’t complete. Hard to get to grips with it as I’ve never used js before so finding it a bit difficult to follow what’s going on.
* Another folder that John sent to me included lots of data analysis, several documents talking about the beef farming industry
* Being from a beef and arable farm that is the industry I would like to create the simulation on however I know firsthand just how many variables come into the industry and I’m not sure I’d be able to quantify many of them in a meaningful way
* Currently thinking about doing some web scraping (which I’ve never successfully done) in order to gather data on average local weather conditions as this would help a simulation massively in its flexibility. Making it much more useful for others, which seems to be being stressed as important.
* However, I don’t feel like that’s a good place to start as I should get some of the backbones of the simulation in place. Just need to find out what the backbones might be…..

**12/01/22**

* Found a very useful python library called ‘simpy’ today, seems to do a lot of the heavy lifting for creating any simulation within python. Not sure if it would be allowed as it may take away some of the struggles of creating a program that can simulate something rather than just creating an algorithm that can accurately simulate the farming industry…. Haha ‘just’. But why invent the wheel right?
* I think I’m starting to find my starting point for the project. Need to define a rough idea of a simulation algorithm, figuring out which parameters I want to use to control the simulation. However to do this I need to narrow down what exactly I’m simulating as the entire industry wouldn’t be feasible. Needs to be something useful….
* One idea that has just came to me would be to simulate the net carbon output per unit of beef produced by a farm or something due to it being so relevant at the moment… will keep thinking of other ideas.

**17/01/2022**

* After deliberating over it for a few days the ideas of what to simulate within the industry of farming are:

Impact of fertiliser, different types of fertiliser, their impact on yield and the environment and profitability

Yield of different crop types/varieties dependant on different conditions which can be set by the user

Net carbon output per unit of beef produced by a farmer

**Outcome of First Friday Meeting**

* Meeting went well got a lot of issues ironed out
* In terms of the direction of the project we discussed that it was to become a model simulation of an entire farm, not a simulation of something to specific that occurs on the farm.
* It was also pointed out why java script would be by far the best language to develop the program in as it makes it extremely easy to use as a portfolio piece as all you need is a browser to run it within.
* Web scrapping was also immediately ruled out.
* The next stages that we discussed was to remove parts of the sample code that aren’t relevant to my simulation and start to add comments which denote an event that may happen on the farm
* These comments can then start to be turned into methods which can be called to simulate the events that occur within the working day.
* It was also pointed out that for this sort of simulation style program, procedural program would allow a much clearer approach that can be more easily modified and built upon by other users.

**24/01/22**

* As discussed in the meeting I spent some time today removing any code from the sample program that wasn’t relevant to my simulation
* I also inserted many comments referring to some of the events that occur on the farm from day to day.
* I also started to add some basic common variables to be used to help to implement functions later
* Plan is to start converting comments to rudimentary methods throughout the rest of the week.
* I’ll also continue to add many more comments as they come to me.

**25/01/22**

* Spent some time today learning some basic javascript syntax as I’ve only ever made one program in javascript and it was a mod for a game so I don’t know that it even counts!
* Had some coursework to do for webTechnologies so spent the rest of my time getting it done to enable me to spend more time later in the week on this module.

**26/01/22**

* Finally started programming properly. Feels like I’m actually making progress now
* I feel like I had been trying to plan everything out far too much before I got started as last term we were told over and over if we plan out the program properly before starting to code it will cause a lot less hassle in the long run. However for this type of project I’m realising it’s much better to get stuck in and actually have something started as then it becomes easier to see where you want to go.
* Have decided for simplicities sake I will leave all my data variables in the same js file for now, just to get something running. Perhaps will have to move them to a separate file in the future when the simulation becomes much bigger and one file just becomes impractical to read and work in.

**27/01/22**

* Getting confused now, every time I try to test the code to understand it better I’m getting completely unexpected numbers for all of my variables.
* I’m pretty sure I’m not getting how the debugger works on chrome properly and that’s causing confusion. When I attempt to step through the code by putting breakpoints I assume it stops on the breakpoint in the first iteration. At which point no variables should be changed but yet all of them have been changed as though several cycles have been ran
* Going to take some time to teach myself how to use the debugger properly and then we’ll see if that helps.
* Should have started with that. Knowing how to actually use it helped a lot and now have some working code
* Even though we were told to keep it procedural and the benefits were pointed out, it’s hard to see how you could get an accurate simulation without modelling the cows as objects and being able to control variables for individual cows rather than an entire herd. However I’m sure there’s a perfectly logical reason why this wouldn’t be practical.
* So far most of the numbers I’ve used are fairly random. I could find out some more accurate figures by discussing more deeply with my family, the variable differentiation technique that was discussed in the last meeting sounds like a much more interesting way to go about this however it sounded a lot more like something for the end of the project, so maybe I just put up with less than accurate results for now?

**Outcomes of second meeting**

* We discussed the issue of creating the cows as objects, it was pointed out that every cow could have an individual set of vars within the json file, worrying about repetitiveness of writing jsons on a large scale for larger farms to use the program it was pointed out that a program can easily be written in order to help automate the productions of jsons if that were to become an issue
* We also confirmed the fact that probability variables etc aren’t important currently as they can easily be made more accurate with the use of software later in the process
* We also discussed the option of changing the tick time to simulate an entire day every tick rather than seconds of a day. This would result in a large simplification of the program without losing much accuracy or detail. This will be something I plan on utilising going forward.
* John talked to me about the basic structure of a json as they aren’t something I’ve ever worked with before
* Instead of tick through the actions of the entire herd of cows I should have a set of actions that a cow can carry out and then loop through all of the cows stored in the json and have some cows do some actions whilst other cows do other actions.
* The how to guide should be based around how to make a simulator of a farm, discussing what’s important to capture and what are the causes of the important events that take place
* The entire website should be contained within a folder with an index.html file that is the main file to open the website containing all the details of the how to guide.

**31/1/22**

* Spent some time today looking at other how to guides for software projects. Focusing especially on ones I’ve found useful in past projects
* Noticed that the ones I find easiest to follow are broken up into smaller sections which are in chronological order. This make the guide easier to follow as you work through the project as you can quickly find the stage that you’re at and continue from there rather than scanning the entire document in order to find your place.
* Trying to break up large chunks of text into smaller sections visually makes the how to guide seem a lot less daunting and made me much more likely to be able to follow the guide right to the end.

**1/2/22**

* Spent some time today setting up the basic layout of the website, getting the general colour scheme sorted etc
* Didn’t have much time to continue to work on the guide as I was preparing for my scholarship interview which was the next morning

**2/2/22**

* Started adding the content of the guide to the webpage, typing up all the different sections.
* Decided to add the sections focusing on the structure of the program, the adaptability of the program and the accuracy of the program.
* The structure of the program is used to give the user an indication of the way in which I suggest actually going about coding the simulator. I think this would be beneficial as it’s important to show specific methods of achieving the goals as well as the more high-level theory based parts of the guide.
* The adaptability of the program is an important section to mention as it allows the reader to start off on the right foot. So many guides can be followed through to the end, hardcoding variables in the main document, only to find they should really be in different files is you want your program to be useful. I have found this to be very frustrating in the past and therefore would want to make it easy to start as you intend to go on.
* The way in which to achieve accuracy within the simulation is a way that I would not initially have thought of (I’d have tried a trial and error method until it’s accurate) Therefore I felt it important to add this section in order to make it clear to others how this can be achieved theoretically.
* Added a box around the main headings in another attempt to break up the guide a bit more. Also helps to highlight that these will be the main talking points of the document.
* Added large headings with plenty of spacing to make clear breaks between sections in order to make the document a lot less daunting as previously discussed.

**3/2/22**

* Added a navigation bar to the page in order to make it easy to quickly jump to the section that the reader has got to in order to make it easier to follow as the reader progresses through their project as discussed previously.
* Had some fun adding in extra details like highlighting the option on the menu that is hovered over, just to increase the appeal of the webpage to the user, the little things that aid engagement all add up to help create a much better guide, providing the user with a better overall experience.
* Included images, where appropriate, depicting code to help the user visualise what is being talked about in the text. This further aids breaking up the text into smaller more manageable chunks whilst also keeping the reader engaged by providing a visual element to the guide.
* The use of images is very important for guides, too many and it becomes too large of a document to load quickly on slower internet speeds (something I’m very conscious of considering my internet speed) I tried to keep images minimal by mainly using them in order to aid descriptions rather than for aesthetics, with the one exception being the main image at the top of the page which helps prevent the user being thrown straight into reading large chunks of text.
* I got John to send me the code for optimising the variables within the program. Thinking that it would be a good code snippit to add to show how to improve the accuracy of the program.
* However, as it was sent as a github open source repository I thought it would be much more useful to link the repository itself in order to allow the user to reuse the code and adapt however they see necessary. I felt that as it seems to be openly available to the public it can just be something that people wanting to complete a similar project can reuse rather than having to recreate.

**4/2/22**

* I realised that most of the guides I follow are never for the exact project that I am doing, rather something that’s close enough for me to get the general idea.
* Therefore, I added a ‘primary goals’ section in an attempt to help the user to follow the guide whilst adapting it to their application by pointing out the main areas that would change in a different variation of the project.

**Outcomes of third meeting**

* Spent some time talking about the ways forward within the project, how we could shift the focus to the ui and how all the data of the simulation is presented in a meaningful way to the user.
* John showed me an example of a good stat sheet which will be helpful to use in the future when we get that far.
* We also discussed the possibility of focusing more on fitting the data in order to make the simulation more and more accurate.
* Another option was to continue to develop the complexity of the simulation in order to make it more realistic, possibly something I’d go back to after implementing some sort of ui and fitting some data.

**Feedback from how to guide submission**

* Find specific guides that are similar to my project and compare my work to theirs
* Don’t use <br> to centre text as it makes it difficult to read
* When talking about constant and probability variables need to explain the difference between them and what they are used for
* In my simulation I should have ‘agents’ which trigger events based on probability
* Straw etc needs to be brought to the farm by an agent then moved from where it’s stored to where it is used then moved out again, each action being performed by an agent.
* Probability of an agent performing a task should be a sum of causes of the agent performing the action
* Events should occur based on a probability calculation that is a sum of a set of ‘causes’ that make the agent perform this event.
* Put in comment form a comprehensive list of events and causes of those events so we can review which are the most important and how to represent them

**7/2/22**

* Got straight to work attempting to implement the use of the json
* Was able to write the json but couldn’t figure out how this is converted to the data.js file.
* Chose to write the data.js file by hand instead so as I could keep moving forward
* Worked out how to manipulate the data from the data.js file from within the simulation file
* From here I made a loop that looped through the dictionary of cows within the data.js file
* This loop then triggers a tick function for each cow, causing them to eat, gain weight and dung
* The amount of food consumed, weight gained and dung produced has a random variation from day to day per cow and also a variation from cow to cow as they each have different feed conversion rate variables and appetite variables which influence the amount of food consumed.
* From here I then moved on to creating a similar loop only for the employees on the farm.
* This calls a tick function for each employee, with a chance of each employee completing a task
* The probability of completing a task is dependant on their role on the farm (part time least chance, full time more chance, owner/manager 100% chance of doing it as if the task hasn’t been done by the time it gets to them they have to do it.
* This loop was for daily tasks which include scraping bedding and feeding
* A loop was also created for jobs that shouldn’t occur daily however I haven’t implemented these jobs yet.
* Added some validation to ensure the cows have ate enough silage to warrant needing fed. If they don’t the job is skipped that day. However it’s important that they are still checked for silage every day due to varied eating rates etc
* Implemented cows calving, if the cow is close to calving it is moved to a pen
* Once in the pen the cow has a greater chance of calving the closer she is to her due date/ the further past her due date she is
* Once calved a new cow(calve) is added to the dictionary and they both stay in the pen
* Still need to write functions for taking cow and calve out of pen and tagging calve etc etc
* Added the possibility of having different types of bedding with some having better values than others

**8/2/22**

* Added possibility of cow having twins
* Added possibility of cow requiring assistance during calving
* Added the function for checking if a cow is ready for the abattoir or not and sending it to the abattoir if it is
* Added wages being paid once per month only by manager/owner employees
* Added a cull variable for the cows, if true the cow has been marked to be culled, it won’t be served and will be fed meal to get it ready for the abattoir
* Added a counter for the number of calves a cow has, once the cow has had 8 (this variable may be changed within the data file) calves it is put into the cull herd, i.e. it won’t be served again and fed meal
* Have also been adding to the list of comments for what is to be added to the simulation. I feel like I have the major components commented now, just need to keep working on implementing them. Although I’m sure more will crop up as I progress.
* Just got asked to start powerwashing our calving sheds this week as calving season is nearly here…. Another job to add to the simulation…
* Added a dictionary for all the fields on the farm
* Added the variable to the cow that allows it to track which field each cow is in
* Added a grass growth function that occurs each day and is influenced by fertilizer and time of year
* Added the section to the main loop to check if any cows are in the fields
* If cows are in a field, it’ll check if the field has enough grass
* If the field doesn’t have enough grass all of the employees have to move the cows
* Added validation to check that cows are in the house before performing any of the winter daily jobs

**Outcomes of fourth meeting**

* It was pointed out how to make the cow tick function a lot more readable by passing in data.company.cow[number] to the function to prevent that having to be type out every line
* Noted that the main way forward is to continue adding more functionality to the simulation, including any events that have large capital expenditure
* There’s no issue with using fake data for optimization if privacy concerns are an issue. It is simply used to put the framework in place that enables new data to be used.
* Focus on the addition of content before optimization
* Also the conversion of a json file to the data.js file is not an important feature currently. Having the data.js file provided isn’t any different currently.

**15/02/22**

* Implemented the change within the cow\_tick function in order to make the code more readable as discussed within the meeting
* Added bedding the calving pens if any cows are in them to the bedding function

**16/02/22**

* Added a function which simulates a cow being served when provided with a cow number and a bull number provided criteria are met.
* Added automatic cull marking when a calf is born if there are no bulls in the herd that it’s not related to
* Implemented a check in the daily cow tick to make the cows eat grass if they are in the field instead of silage
* Also implemented the cow serving function in the daily tick
* Implemented cows being put out to the fields and brought into the houses at certain times of the year
* Implemented the possibility of a cow becoming lame whilst in the houses and the consequences of this
* Implemented small chance of cow throwing the calf at birth
* Completed some minor bug fixes from the last days code, although I’m sure there’ll be plenty more that creep up

**18/02/22**

* Implemented the spreading of fertiliser on fields that are for grazing
* Implemented the fertiliser more accurately affecting the growth rate of the field, effect slowly decaying until the field is refertilised
* Implemented the chance for managers to restock the farms supplies. Currently restocking straw for bedding and artificial fertiliser when the farms inventory is low
* Used constants to set prices of straw and fertiliser however in the future these could possibly be varied throughout the course of the simulation to add some variation
* Struggling to fix a bug in the code that allows the first two employees in the list to complete the jobs but the last employ gets passed to the functions incorrectly for some reason
* Started looping through the simulation multiple times and very occasionally when the money variable is changed it becomes nan instead of a number

**Outcome of fifth meeting**

* We talked about the goals of the blog post
* Communicate what success looks like for the project within the blog post
* Three key areas are to talk about what would be good to have in the project
* What is required within the project to make it work, referencing a paper to back up the areas of importance, ie a data analysis paper showing what affects the farm the most
* What has been implemented already out of the list of things that need to be done and how I’ve implemented them successfully
* At the end of the meeting I asked if John could help me identify what was causing a couple of bugs that I couldn’t solve as mentioned previously
* He described to me how to implement a function which validates that the data within the simulation is as expected after the execution of every function and if it isn’t then it flags that function as being the area in which the bug is contained
* John also mentioned that rather than creating a random number every time I need to within the program I should generate the random numbers based off a seed in order to make troubleshooting easier as then the bug is repeatable every time the program is run

**19/2/22**

* Successfully implemented the validation function as described within the meeting
* Currently checks that all the objects haven’t been corrupted by the function and flags if they have.
* More can be added to the validation function as time goes on in order to make it more sensitive and better at pointing out bugs within the program
* The function enabled me to find the typo that was causing the employees object to become corrupted every now and again

**21/2/22**

* Added tracking of the cows most recent calf to enable the cow and calf to be kept together until weaned
* Added the calving ease being changed in the data file according to the breed of cow
* This also affects the probability of assistance being required as well as the probability of the cow throwing the calf
* Updated the calve cow function to include all the new variables contained within the cow object
* Fixed a long list of outstanding bugs and omissions of functionality within functions in order to have the simulation working as it would be expected to

**22/2/22**

* Worked on another list of bug fixes as well as logic errors that I found when testing
* Added a calf being flagged for culling from birth if herd limit has been reached
* Implemented the cows feed conversion rate being slowly decreased as they get older in order to prevent cows gaining excessive weight
* Implemented slurry being spread on silage ground
* Implemented silage being made 3 times per year if the grass has grown enough, charging for contractors, if excess silage produced it’s made into bales

**24/2/22**

* Implemented all random events being dependant on a set seed in the document in order to allow for aid with debugging whilst also producing consistently predictable runs due to a pre-determined seed.
* Completed the final bug fixes out of the known bugs, as far as I’m aware the program is now working as intended without glitches
* Spent some time tweaking some of the numbers to get the output values something close enough to be logical, used statista etc to get ball park figures for a lot of values
* Included the depreciation of farm machinery just as an average stat that is removed per year. This will give a very rough estimate figure for now however should be simulated in much more depth at a later date.
* Began changing the layout of the how to guide in preparation for next weeks submission

**Outcome of sixth meeting**

* We discussed the next deliverable. The blog post
* John mentioned it may include the how to guide but that the two pieces have a different audience
* The blog post should be aimed at someone just starting this module
* The reader should be able to understand the project, see where the project is going in the future and run the current code and explain what they should add to it to make it better
* The blog should talk about what success means for the project and talk about what would make the simulation a good simulation.
* We also talked about the next steps for the project asides from the blog post
* John mentioned that a calculations.js file should be made in order to help with the optimisation of the program, enabling more stats of the simulation to be recorded as the simulation progresses.

**28/02/22**

* Finished improving the how to guide. I attempted to make it easier for the reader to actually create the program just by reading the guide as per the suggestions from the feedback.
* I also split up the sections on to different web pages and added some extra styling to the pages to make them look much more appealing to the reader and engage them better. Making them more likely to read and appreciate the guide.
* Updated the guide to mention some of the additions I have made to the program since I wrote the last guide. E.g. the use of seeded randomness. As I think these additions are extremely important to be mentioned.
* I also included a link to a paper which describes how to create a simulation of a farm in theory as per the suggestion of linking similar work. I think adding this paper does help the reader to understand the thought process behind making a simulation a lot more.

**1/03/22**

* Got started on the blog post
* Decided on a general layout similar to the new layout of the how to guide with sections being different html pages that the reader can move to after finishing the previous section. This helps breakup the length of the document a lot and makes it more accessible to go back and find areas you left out or revisit something you read previously etc.
* Added links to the different sections in the nav bar at the top of every page to aid quicker navigation. This makes the reader more likely to use the blog as flicking from one page to the next to search for something will be tedious for anyone and many will simply stop looking. Instead they can now go straight to whichever section they are currently interested in.
* To further aid the quicker navigation, I added extra info appearing when the reader hovers over a link, this helps inform them what will be on that page. Allowing them to quickly see if that is the page they want or not.
* Added the first page of content including what success looks like for the project and a set of basic goals that I would deem necessary for the project to be a successful end product.
* Made the sections of the blog relevant to the criteria, informing the reader about the project so as they can understand it
* Inform the reader where I see the project going in the future, what the next steps are etc.
* Inform the reader how to use and run the program in it’s current state
* Evaluate the project where it’s at currently and after my immediate plans
* I believe all these sections encase what would be useful for a student about to start the module. Covering all they need to know to get going quickly and take the project on to completion.

**02/02/22**

* After talking with dad, I implemented the annual insurance cost of a company being included within the json and removed at the start of every year within the simulation. I chose to make it included within the json as it is such a variable cost from business to business.
* I filled out the progress so far section of the blog. Including all the features that I have implemented into the simulation
* I presented this in a series of long lists, splitting the events by ones that are carried out during the winter, summer and general features.
* This helps make it easier to come back and check if certain features have or have not been implemented by making it easier to find the features as they are in respective lists rather than one extremely large list.
* I decided to present the features in short bullet pointed format within a list in order to keep it brief and concise. If I were starting the module I wouldn’t want to be wading through paragraph after paragraph in order to see what a simulation can do. I’d much prefer a list that I can skim through and get the gist of each feature much quicker.
* I also filled out the ‘project future’ section
* I decided to do this by talking about the different features I’d like to see implemented individually
* I sectioned off the different sections, making it clear with titles what each block of text talks about. This makes it easy to flick through to find a particular feature that the reader is interested in implementing.
* Although text intense, I wanted to include detail about how I envision the feature being added in order to make it as easy as possible for the reader to see how to implement it. I decided that, even though I’ve been attempting to avoid large paragraph chunks of text so far in order to keep the reader more engaged, this was a trade-off I was willing to make to have the necessary details included.

**3/3/22**

* Filled in the running the simulation section of the blog today.
* I went heavy on the pictures with circles showing what to click and where to find things in order to make it as clear as possible to get the simulation running quickly.
* Although the number of pictures may be overkill to explain something simple, it is something I would have found useful at the start of the module. To me there is value in breaking down even simple processes into many, many steps.

**4/3/22**

* Today I worked on the evaluation section of the blog post.
* I approached this by addressing each of the goals set out at the beginning of the post.
* This method of evaluating the project helped to tie the entire blog post together and helped to really show the strengths and weaknesses of the project so far.
* I added a small conclusion to the end of the evaluation to give a brief summary that can give the reader a general understanding of where the project is currently at.
* I attempted to address each section as objectively as possible in order to give a fair summary of the project as I don’t want to impart any bias on the reader. Something which can be misleading and sometimes cause confusion, which is never helpful.

**Outcome of seventh meeting**

* We discussed the next step of the project being data fitting
* John described how to begin to approach this part of the project
* Start by gathering some real world data
* Create a calculations script which can record and keep track of data equivalent to that
* Create a function that compares the data gathered within the simulation to the real world data and create an error margin
* Use the optimiser to adjust the global variables and it will adjust the variables to produce the smallest error margin according to the function

**7/3/22**

* Today I began the data fitting process
* I gathered a small amount of real world data to begin with to get me started.
* I created a calculations.js file which helps to record the data needed whilst the simulation is running.
* It records the total number of cows that go lame, the total amount of grass silage that should be produced.
* The total feed cost in the simulation
* I then created an optimise function which takes the absolute value when you negate the real data from the simulated data. This is all then returned by the function as a totalError variable.
* I then attempted to use the optimiser John had given me the github link to. However upon running the optimiser it didn’t seem to change the variables.
* John explained that this was partly due to the fact that the optimiser was making very small incremental changes and seeing no difference in the error and therefore decided there was no way to make it better.
* Therefore I should either look into making the optimiser make larger changes in order to allow it to see a difference in the error margin, or make my own crude optimser that will change the variables randomly and keep changing them until the lowest number is recorded.

**8/3/22**

* Attempted to create my own optimiser program today
* Made it take in the parameters of the optimise function I made yesterday as an array
* It then loops through this array, adding a random number to the parameter, running the simulation to see if this reduced the error, if the error isn’t reduced then a number is taken away from the original parameter and the simulation is ran again to see if the error is smaller or larger. If the error is still larger then the parameter is set back to original and the loop moves on to the next parameter.
* I called this minimiser function from the optimisation program with an array of data, focusing on attempting to match the real world data for lame cows to the simulations data. I did manage to get it to work a little bit but it didn’t seem to have as large an affect as I had anticipated.
* For some reason running the minimiser function seemed to take a much longer period of time than I had anticipated. I must have programmed in some large inefficiencies unintentionally.

**9/3/22**

* Spent some time today starting to display the results of the simulation on the html page
* Just displayed the results as basic text on the page for now, updating them every time the simulation finishes.
* The run simulation button now increments the simulation by a year, allowing the user to see the changes from year to year.

**10/3/22**

* Spent more time working on the interface, still currently just displaying the results of the simulation after it has been done
* Have set it up that it only takes half the page so as I can implement running two simulations side by side. However I’m starting to realise that this is completely unnecessary as you can just have two instances of the program running side by side. Therefore I may scrap the half screen idea and just return to full screen.
* Added the ability to change the seed of the simulation
* Found a bug that allowed the surface quality to dip below zero very easily when the herd number became larger, this caused all cows to become lame and be culled.
* Fixed the bug by preventing the surface quality from dropping below 0
* Added the ability to change the herd size from the html page
* Note that the herdSize is the size of the breeding herd. The total size will normally be double this due to the breeding herd still having calves for beef.

**11/3/22**

* Started to work on showing, graphically, data values each month throughout the year to give an idea of how the numbers change throughout the year.
* This made some new bugs clear, i.e. cows not calving as expected. It seems to work for the first year then it just stops calving any cows at all.
* Trying to debug this error, I found that cows aren’t getting pregnant after the first couple of years, which is causing the bug. The serve\_cow function isn’t being called.
* Have now noticed this is because the cows are staying in the pens, can’t be served if they’re still in a pen.
* Managed to stop the cows from staying in the pens all the time but now something is preventing the cow from going into the pen even when she should be.
* Have managed to narrow it down to a bug in the month detection being compared with the due month
* Noticed that the months on the graph weren’t correct which was skewing why bugs might be appearing
* Now noticing that every year calving moves forward a month until suddenly no cows calf and they slowly all get served
* Found this to be due to a small logical error when setting the due month after the cow is served
* This seems to have fixed the bug and now the graph output is as expected
* Added total cows culled to the monthly view
* This pointed out another bug where a cow that had previously been culled was being culled again and again. This was the reason for the money going so high so fast. It has now been fixed.
* Money never becomes positive now

**Outcome of eighth meeting**

* We largely talked about feedback from the blog post in this meeting.
* In my blog post my goals were more focused on what I would have time to do within the scope of the module rather than larger scales goals for the project with a vision of it continuing well beyond the module’s time limitations.
* More details on the existing code are necessary in order to make it easier for a student to understand how it was built and works.
* The clearly defined layout of goals and what I have achieved so far was good.
* We then discussed why the data fitment I had tried to do this week wasn’t successful. Mainly it was due to the change in variables not resulting in a smooth change in outcome. Which makes it difficult to optimise with a program.
* If I wanted to do this better I should run the program with several different seeds and take an average in order to achieve a smoother curve that could be optimised much easier.
* We then discussed the pros and cons of carrying on pursuing the data fitment goal or moving to focus on the visualisation aspect of the project for the remaining weeks

**14/03/22**

* After the meeting on Friday I have decided to continue focusing on the visualisation of the project much more before attempting to fit the data again.
* This is because it was clear whilst just implementing some very basic visualisation last week that more bugs were present in the program than I realised and the visualisation helped to find and remove these bugs. There would be no point in fitting the real world data to the simulation if there are a large number of bugs in the program that I’m unaware of. Hence I think it important to continue with visualisation.
* I also feel it is important to continue with visualisation before data fitment as it adds a lot more appeal to the project for an outsider looking in. If somebody wants to follow and carry on the project then having a nice graphical interface that the user can interact with makes the program feel much more polished and have a much higher level of appeal to a much more general audience.
* Therefore I will be focusing on visualisation in the coming weeks.
* Today I programmed a bar chart that shows the monthly outgoings of the farm. So far it is displaying the spending on fertiliser and straw.
* This presented another bug, where the farm was spending over £2million on fertiliser in the first month. Which is obviously illogical
* Found that this was partially due to the restock function ordering enough fertiliser to cover all fields including silage fields which shouldn’t need it anyway.
* It was also due to more being stocked than would ever be required.
* Farm still makes a small loss every year now but not as drastic as before.

**15/04/22**

* The visualisation shows that the simulation constantly makes silage, far more than it uses. This results in the farm constantly paying extremely large prices for silage bales to be made because their clamp is always over full. This would cause the farm to constantly loose money.
* Having looked into it I think this is more of a data fitment error as grass is growing more than it would in the real world due to growth rates being higher than realistic. Therefore this ‘bug’ should fix itself when the simulation is optimised.
* There seems to be a bug that fertiliser is only ordered in the first year and then never again. Could be due to still ordering more than is ever used or else it isn’t being used properly

**16/3/22**

* Changed the styling of the graphs to make them fully visible and more easily read by the user.
* Added a wages statistic to the monthly outgoings chart
* Added a statistic to track the contractors fees to the outgoings chart
* Adding this statistic showed a bug that contractors only seem to make silage once per year and they make the exact same amount every year. Which doesn’t seem like intended functionality.
* All the additional bugs I’m finding are reassuring that going down the visualisation route was definitely the right choice.
* Spent some time attempting to fix the fertiliser being ordered incorrectly bug.

**18/3/22**

* Noticed from adding the wages statistic that the wages peak during the summer. Although this looks like a bug I believe it is more to do with the simplicity of the program as it requires all employees to move cattle from one field to the next. Although in real life the contrast between winter and summer wages wouldn’t be as dramatic it would be present.