November 1st - I have started thinking what I am going to do for my project and I have decided that I am going to do something along the lines of computing and code a program. Because of this I have made the decision that I am going to make an artifact instead of writing a full paper. But I have not decided what yet.

November 3rd - I have now made my decision about my project and I am going to make a self learning AI but I still have not made a complete decision on the title yet. So will do some research so that I can get a better idea on what I want to do. As well as getting some ideas of good titles.

November 7th - so after doing this research I have learned about the different ways computers can learn. I have now got a title of “To make a self learning artificial intelligence that can learn the simple and complex patterns found in board games using multiple learning techniques.” My next step is to start coding an emulator for noughts and crosses so that I have something for it to learn and so that I can prove that it learns over time.

November 9th - I have finished writing my code for the noughts and crosses emulator so I can now start designing the first of my self learn codes. My first self learn code will be a brute force method.

November 12th - I have now made a rough plan in flowchart and pseudocode of how I'm going to program the first self learning code (brute force) and I will now start programming it. One thing that I have learnt from this planning is that being able to search the files will be the slowest part of the program. So while I'm programming this I will be thinking of other ways to increase the speed of the program and decrease the time taken reading from the file.

November 15th - after a few days of programing. I have only just realised that I forgot to add in a part so that the Noughts and crosses program can talk to the AI. So I need to design a way for them to do this. I am thinking about making an exchange file for talking between programs.

November 18th - I have done some more programing and I am now ready to start making the exchange part of the program but as this is a large part of the code I feel I should test it in a smaller version first to make sure it works. So my next step is to make a small exchange program to test this part of the code.

November 19th - I have finished coding the small exchange code so I am going to implement it into the main code so that I can test it.

November 21st - I have now implemented the exchange code however I have seen that the program is crashing a lot so the next step is going to be to fix that as it is deleting all of the data it is learning. But before it crashes I have been able to look at what it has learned and it looks promising but it is difficult to tell at the moment.

November 25th - I took a break for a bit as I felt I wasn’t managing my time very well between this and my other work. But as a result of this I have now forgotten what each part of my code does. Because I did not leave any comments and all my variable names made no sense. Before I can fix the bugs I am going to go through it and rewrite it (renaming variables and adding comments). So that I can understand it to debug. Hopefully this will help me to find where all the bugs are as I am doing this.

November 27th - after spending a few days renaming and commenting the code. Doing the testing I have noticed some patterns of when the program crashes. The program is a collection of different codes working together with exchange files in between so that they can all talk to each other. To allow it to learn different games to prove that it is general purpose. It also helps to have them as different programs because it means that I can spread the work over multiple CPU cores and so I can later add AI vs AI to that it can learn faster without a human opponent. However this multiple program approach has caused a lot of problems as I have to pass the information between the different programs and this is the part that looks like it is crashing. This happens when one program is writing to it and the other is reading it at the same time and this makes the program crash as it deletes the info just passed to it and therefore it never gets the info that is sent and just stops.

November 28th - as I was trying to fix the problem with the exchange between programs I have found out the the programme overheats the computer if it is left for too long. I started a test and left it then came back later and it turned off from overheating. To prevent this I have added into the parts of the code where it does a lot of work for it to wait until for only a 50th of a second before it continues. But it has made a very big difference. However just in case I am going to to get it to back up itself every time it finishes a game so that it doesn’t lose any data from over heating.

December 1st - I have now added the part so that it backs itself up after every match but it is very slow. I have so started on making another small version of the code for exchange so that I can test it faster. I think I have fixed the problem by adding another file so that all info is passed between the programs; using one file and then in the other file a ‘tag’ is written to show that it has finished writing. As the other program reads the ‘tag’ file to see when the tag arrives then when it has seen the tag it looks in the other file to see what info has been passed across. In the tests I have done it is working very well. So the next task is to move it across to the other programs so that they all use the new way of passing information to each other.

December 2nd - I have moved the code across and to test it I am going to leave it to run for a hour to see if it crashes. If it does not crash then I am planning to leave it to run overnight to see how much data it can learn and how long it takes.

December 3rd - after leaving it to run for a hour it had slowed down a lot and was taking a lot longer to play each match. So I stopped it and gave it a time to cool down and then started it again to see if it was just because it was running for so long that it had thermal-throttled. But it was still very slow so the only thing that would make it slow down that much would be that it has learnt more and that makes the file a lot longer so it has to read a lot more lines of the file and reading from the file is the slowest part of the program so it slows down as it fills more of the file up. So I am going to start to plan a different way to read and write all of the data that the program learns.

December 5th - So I have written out a plan of all the work that needs doing to make the program run faster and ways to improve it. But I am thinking of making a new version of the programs so that I can start from scratch. Learning from my mistakes of the last program I will make an improved version. So that I can add new features for example, decreasing the time taken to play each match and the ability to select a piece then move it (like chess). As well as getting rid of the bugs. To do this I am going to spend the next few days working on designing the new programs so that I don’t have to go back and fix it later. This was a big problem with the last program because I didn’t plan and test each part enough.

December 9th - So I have planned each step of the program and as I have learned from last time that the: exchange, reading and writing files where the biggest problems with it and making it slow down. So because of this I am going to plan these more as that will be the most important parts of the program. So I am going use the learned files from the last program to uses as test files so that I can test how well the new search algorithms work on large files in the same format; I will also make some more small files to see how it changes as it learns more.

The first thing I will try and do will be to split the data up across a collection of smaller files. I am thinking of doing this by having each move be a new database that will be linked with a single look up database that will be easier to read and hopefully this will help fix the long time it takes for the program to read the file.

December 11th - I have found that searching for the board in the lookup database line by line takes a long amount of time so because of that I am going to see if there is a way to skip to the line of the file that is needed and then use that line number to find all the other ones for that board. So the next step is to start making these small programs and find a way to test them to see how well they do compared to each other. One way I am planning to do this is to get the computer to try to different scenarios with each of the ways of searching and time itself to see how long it took each version.

December 13rd - I have been testing different ways that already are used to search a file so that I can see how to improve my way of searching the files.I will test the different ways by using the timing code I made before to test how long it takes to read all the moves for the board.

December 15th - I have been thinking through different ways to fix the problem with searching the files for the next board to pick and I have decided on having it split the moves that can be make on a board into the different files. For example O&X will be split into 9 different files that are for each move that can be made on a O&X board. However to be able to find the line number for all the files I will use the “subcontext” function that can tell me the line of the board. I will use that on a lookup file that just has the board to look up and then points to the line number in the other files. The next step is to test this as a full look up and see how well it works

December 17th - I have implemented the code so that I can see how fast it can look up from the file and it is a lot faster and works better than all the others however I still feel it could be improved later to run even faster. It is runing around 10 times faster and it only slows down by 0.01 seconds if I add all the lines needed to play O&X. But for now it works and I will add it into the main code and test it

December 18th - I have added it to the main code and tested it.

* Now that the code is done I have started running it so that it can go over night and I can come back tomorrow to see how well it has done.
* After leaving it to run over night it nearly fully learned how to play tic tac toe apart from a few boards. Because of this I am thinking that I need to look at a new approach to learning the game tic tac toe as the current way it is learning will not be able to properly learn these boards. So I am going to look into new ways to get the AI to learn.
* After thinking how to get it to learn I am going to look at different ways different people get the computers to learn.
* From my research I have found that one of the best ways to get a computer to learn is by using a neural network. This is a bit like simple model of the human brain. But that it can learn a lot faster than a human can as it is can run through it a lot faster than a human and that it can make bigger jumps in learning by using what is called back-propagation. So I am going to start by making my own version of someone's code that make can make a colour based on an input colour. As I felt this was a very good way to understand how it works and to make it more visual.
* After coding the basis network it looks very good and easy to follow. And I work with what is called feed forward pass this means you put in inputs and it outputs with the current weights what it thinks it should become, And it now creates the error of the network. Next step is to research backpropagation.

==================================================================

* I have got annoyed with the the way it calculates the derivatives so I am wanting to add a library so that I can calculate the derivatives for me so that I know that they are calculated properly. So the next step is to see if there is a library that can calculate derivatives. Next is to download and learn the library.
* I found a derivative calculator library (sumpy) but I now need to install the library and I need to learn how to as I have never been able to install them.
* So I have now learned how to install a library and I have installed sumpy. And because it was so easy I have now decided that I will also install the other library that I will probably use later so I have also installed: scipy , numpy and matlab as they can help speed up the process.
* I have now learned how sumpy works and what is needed to calculate the derivatives that are needed for my program. However from learning this and testing some long algorithms I have found that it is taking a long time. So I am going to try and speed it up. As I add it into the code. So the next step is to add in the part at the start of the code so that it makes the algorithm for the new network.
* I have added the part that makes the algorithm for the network at the start of the code. However I have seen that it will probably run very slow. But I will try it.so the next step it so replace the way it used to calculate the derivative with the new way of calculating them using the library extension sumpy.
* I have now add the derivative calculate function so that I calculate the derivative for each weight each time. However it looks like it is going to talk around 4 seconds. After testing it takes 2.9 seconds to run each time and this is too slow. So the next task to to speed it up.
* I have been thinking about the different ways to speed it up and have been testing the different parts that slow it down and have found that that derivative calc is the slowest part with string as the next slowest part so I am going to see if I can do all the derivative calc at the start when the network is set up.
* I have changed the way the difference is calculated so that it is done at the start of the program and in each weight info it stores the derivative with the variable that needs subbing in. However this is still using a lot of string manipulation and that took a lot of time before.
* I have tested it and as I thought it is taking even more time then before so I am going to investigate more ways to speed it up so that I can keep the accuracy of this library.
* I have not found any more ways that will be able to speed the code up enough to allow it to work fast enough for my needs so I am going to stop it there with that library and try again with the old method of it.
* After talking to one of my maths teachers about how I was calculating I made some changes so that it is a bit closer to the code but it is still wrong and I will try a improve on it more with what my maths teacher and I talked about.
* After trying to work out how to fix it I have decided that I am going to try evolution as I am not getting the drive to fix the maths with the ANN atm so I am going try and make another way to learn that can use the same processing structure to speed up the computing of it.
* So I am now going to start the planning of how I will try to code evolutions as this is a big code with everything needed to work more closely together I want to plan it more and think how to to make it better.
* After watching more videos to learn how natural evo works I am going to try and plan it out. I feel before I get anywhere near coding though I need to look at how others code computer evo to see how it compares. And then build upon that so I will plan my way then compared to the other ways looking at the strengths and weakness of both.
* I have look how other people code their evo learning programs and from learning I have planned how I wil code my code so I will now start coding it
* I have have coded most of the evo however it is not working at the moment. The thing that I have found the most difficult with the coding it is that there is a lot of lists and it is difficult to code it so they all work with each other.
* While I have been coding evo I have also created a tic tac toe code for it to play to try to learn but after finishing it I have realised that I am not happy with how it is structured at the moment and it is very slow. So I am going to completely rewrite the evo code to make it run faster so that I can run more generations. but this means I need to rewrite the tic tac toe game to work with it.
* After thinking about how to get tic tac toe to work with the new way of coding the evo so that I can run more generations. I have decided that I am going to make and design a new game so that I can test that is works more easy to see if it works while I am trying to quickly code it and see if it works.
* I have planned a simple multiplication game for evo to try and learn and I managed to fully code it in the a morning so I have now got to do the finishing touches to make it work with evo then I will test it to try and increase the performance that I can get from evo
* I have now fixed all the bugs that stopped the programs talking to each other so I am now going to try and get evo to learn how to beat the control (hand made).
* I have tried running evo on the new game that I have now called 2xgame (as the way to win is to multiply by 2). However it crashes after it has been run for between 7-10 moves so I need to see what is the problem but I have a feeling that it has something to do with some games finishing before others even though that should not be possible at the moment as I wanted to try it as simple as possible to start off to make testing easier.
* After debugging the code I found it was because I was stopping some of the games before earlier as they made invalid moves so I am now going to both:
* just in case I am going to start coding a part to make it cope with them stopping early
* stop them from stopping the game early even if they make invalid moves
* How that it can run I have found that all the fittness is the same so I am going to have to change how fitness is calculated so that I is collected from a different part of the game to make it easier to find the fitness of each DNA.
* I have now fixed this and it now learns the 2x game however I feel that it could be made faster so I am going to try and optimise it so that it can learn faster. But first I am going to see how other people how made their learning faster to see if that will help me.
* After doing my research I have found a few things that will help me speed up the learning. Firstly I am going to square the fitness before I use it to calculate the selection chance so I am going to do this so that it picks the best one more likely than the small fittness. I am also going to look through each part and see what part of it is slowing it down the most and I will go through and speed that part up by optimizing the code. I am also going to add a config file so that I can tell it to only print at the end of each asked number of generations.
* I have made the program speed up however it is still taking a long time to learn so I feel there is a problem with one of my algorithms so I am going to try to and fix that by going through testing each part again against online examples of evo
* I have gone through the different parts of the evo code and found the problem and have fixed the bug and I have run the code and it has now fix the problem will is taking a lot less time now to learn.
* Because it now works and can learn quick enough I am going to run it on the 2xgame to see if it can learn it again and I feel that this time I have solved the all problems and it should work.
* I have tested evo on the 2xgame and it learned it and it did it very fast with no help needed to let it learn. And I looked through each step that it processed and from that I have seen that at each step it is taking the top from each of the population so because of this the next step is going to be to make 2xgame more difficult and see how this affects the learning speed.
* So I have planned the next level of 2xgame so that it can challenge more of the evo code so I am adding into the xgame these rules 1) an upper limit this is more just to make it easier for me to look at the data and to control how fast it increases the board number so I am going to set the limit to 3 times the current board to start of with. 2) if a move doesn’t fit into the rules that it is counted as invalid and that game is ended. 3) then instead of finishing on turn 4 it will now end when the board number is over 21.
* After adding in the new rules I found that it was still learning faster I feel that this was because it was limited more the population were all agreeing on what the best weighing is. As well as because the games are ending sooner it can run through more.
* The next thing I am going to do to improve the AI is going to be adding in a neural network so that it doesn’t just evolve a number it can evolve a neural network. This is the first time I am combining the two different learning types into one program so that I can use the benefits of both. To do this I am going to make an ANN library so that it is easy to code and can be used again. To make this easier I am going to use the ANN code I made before to make it faster to code.
* I have modified the ANN program from before to turn it into an easy to use library for the evo program to use an ANN runner. So the next step is to implement it into the evo program so that I can use neural networking to improve the learning it can do.
* I have added the new ANN runner libary to the evo program so the next step is to get rid of the old system that evo uses to use DNA and replace it with the ANN runnner.
* I have striped out the unneeded code and replaced the code for using the DNA with the ANN runner. The next step is to run evo on the 2xgame with the new code after some bug testing to see how it runs.
* After I tested for bugs and debugged the code I tested the new evo on the 2xgame and it worked very well only running about 0.0001 seconds slower per move which in this case can be counted as negligible as it speeds the evo program up in larger tasks. As well as making it easier to move to other problems without having to do more coding to make it work so it makes it a more general AI.
* The next steps I am going to take it to make it easier to analyse how well it has done by adding some more user interface so you can make it easy to understand what it is doing.
* I have now also added into the config load that I can miss off comments. This means it is easier to edit the config file for the evo code making even more general so that it can be used on more tasks without editing the code.
* The next step I am going to make to my code is to add in a new library so that it is easier to change the puzzle / game your training it on.
* Now that I have added in the new library I am going to start thinking of a new game for the evo code to learn how to play. So I am going to do some research to find a suitable game.
* After doing my research I have come to a new company called “OpenAI” they have a library that is called “gym” that allows you to easily get an AI to learn against different games for example I have found an environment on their website called “cartpole-v0” this is designed so that an AI can learn to balance a pole on a moving cart so that the cart doesn’t move off the screen too much and so that the pole doesn’t fall off the cart. Because this library is very easy to use and implement I am going to get evo to learn how to beat this pole balancing game.
* After a lot of problems trying to get the OpenAI library installed I am now going to start writing some code to run the cartpole puzzle. But as I was looking at how to install the library I have been looking through the different puzzles OpenAI have and it is making me want to test all of them on the new evo code.
* I have implemented the new code however it is crashing on rendering the cart so for now I am going to stop it from rendering the game but this will also speed up the learning of the the puzzle so I am fine with turning it off for now.
* With the rendering off it is now running a lot faster. But there are some bugs with the code so I am going to fix them and then I will run cartpole.
* I have fixed the bugs and have ran cartpole and in a matter of a few minutes it has learnt how to balance a pole on the cart! But because I got rid of the render it is still just numbers on a screen so I am going to add in rendering for the end so that I can show how it has got better over time. Because people will not understand just numbers. But people can very fast understand a video and see what is has learned.
* I have added in rendering so you can see how much it has learned. But I have realised that waiting for it to learn and watching the screen takes a lot of time so I am going to look into adding sound so that it can tell me when it is done so that I can continue with coding the next part while the old part is learning so that I can have a better work flow.
* I have added in sound and it only took a few minutes and it has already been worth it. I have saved so much time and I am getting more work done now that I can train and code at the same time. The next thing I want to add is a new puzzle for it to learn. I am thinking of the second version of cart pole as it makes sure that evolution has learned it because it make the learn to Balance the Pole for longer amount of time.
* I have ran a test case for the second version of cart pole to test it works and as it learned it and it works I am going to now make a new code to run it properly.
* I have written the code for cartpole-v1 and it is doing a lot better and it proving it works however because it now runs for 500 ticks each generation takes a lot longer. So the sound is helping a lot. Because this is working so well I am planning to start writing my report soon as well as my mid protect review. But I will take my time on them as I want a very strong report but I am also about to start my exams so I want to put more focus on to them.
* I have added in a file picker so that it is faster to pick a different puzzle to learn
* I am now feel I need to start writing the brute force code so using the same format as evidence so that I can more easily combination them into one program.i am going to start by making a copy of my old brute force code and editing the parts I don't need and that won't work with the new way I am coding it.Then turn the tin to python code so that I can add more and edit it properly
* I have edited the original code and striped it down to the code needed to run brute force well I am now going to convert to python
* I am finding it is more difficult to turn small basic code into python then I though and this shows how different the two languages are but this is because it is designed to run faster. As well as not being designed as a starting code for people to learn
* I have code all the basis of brute force in python I am now needing to use a simple environment to test it to compete it to how it should work. To do this I will use the 2 X code as it is simple and easy to debug. So next step is to get brute force to be able to connect with the 2 X code
* I have got them to connect together and I have to now debug the different input and arrays they will have to deal with. How ever while I have been doing this I have found that I will probably need to give brute force code more info about each problem to say what numbers to try on each board. This will also speed it up when it get to the training Stage.
* I have done some maths of how many possible boards there are in tic tac toe and I need to change the code for tic tac toe as the current code as 512 possible moves on each board instead of 9 so I am going to change it so that it doesn't have to run all 512 as only 9 are valid
* I have recoded tic tac toe so that it only has 9 possible moves for each board instead of 512 these means that it is running a lot faster on brute force. As well as evo is working a bit faster.
* Brute force is now working on 2x and tic tac toe so I am going to now combine evo and brute force so that they can also play any environment. So it is easier to code. As I don't have to copy and paste all the code as they can all access the same environments through a Common user interface. I am going to call this full AI and the common user interface is going to be called the backbone as every thing will talk to each other through it.
* I have started making the full AI but I am struggling as I have not planned it before starting so I feel I will probably be changing it a few times as I didn't have a clear structure to the code.
* I have got as far as I can with the backbone part as I am unsure how I am going to call the different Ais and environments so I am going to copy their code across into it.
* I code for the different Ais have now been put in and I have found a few problems with how I coded the back bone but it should be easy to fix so i am now going to code all the file read in part so that it can run it all the different part without my input needed and so that It can use new environment without having to hardcode it this will make it easier down the road. As just to test it I have hard code it.
* It can now find each AI and evident for me then to pick which to use the next step is to make this automatic so it cycle through each AI and environments which plays each one in turn.
* I been looking for some new environments that are more visual so it makes more sense to people when I explain it. I am struggling to make these work as they need more complex downloads and require more stuff to work as well as some need a license to run.
* I have added in the sound to say when it has finished this will help a lot more when it is left to tech it's self all the environments. While testing this I realized that brute force has no way to know what number to output so I need to design an algorithm to get that to work I don't feel I have the knowledge of how to do this so I'm going to ask for someone's help to design the algorithm. As I am not good at designing these sorts

* I have been told that I have a presentation coming up and I am start to plan how I am going to give people the best understanding of my code possible and project possible. From my previous talks and presentations I have found that the more visual the better so I am going to record the rendered outputs so that I can include them in my presentation. I am also going to save all the data that is collected about fitness so that I can make some line graphs. To make it easier to understand but I am going to have a problem decided what information I want to tell people as there is so much I just don't know what to include in ain a short presentation
* I am adding in a part of the code so that it can render what it has learned so that it is easier to understand in the presentation. This however i need to add more environments.
* I have now been given a deadline for when all my work needs to be completed by andit is starting to feel like a lot of work that need to be done in a short amount of time. Because so far I apply have one environment that can render.
* I have tried the program i have found that it crashes on the neural network AI i don’t know if this is possible to fix and if i can fix it in the time i have been given. So i am going to crunch down and try to fix it in the time given.
* I have found a old version of my nurl network that is working but looks very different from what i need it to look like so i am going to start slowly change it to what i need it to be structured like so but step by step so i can see where it breaks.
* I have managed to get my old code to work in a structure that works with my current format of my project. So this now need to be moved over to the full AI 2.0 code to test it. In the real conditions.