

Computer Games Development

Project Report

Year IV

Aaron O’Dea

C00217820

[03-05-2020]

**Faculty of Department of Computing and Networking**

**Open-Book and Remote Assessment Cover Page**

**Student Name: Aaron O’Dea**

**Student Number: C00217820**

**Lecturer Name: Ross Palmer**

**Module: 4th-year project**

**Stage/Year: 4th Year**

**Date: 03-5-20**

**Declaration**

This examination/assessment will be submitted using Turnitin as the online submission tool. By submitting my examination/assessment to Turnitin, I am declaring that this examination/assessment is my own work. I understand that I may be required to orally defend any of my answers, to the lecturer, at a given time after the examination/assessment has been completed, as outlined in the student regulations.

**Contents**

[**Acknowledgements**](#_heading=h.vk7jvt45irgo) **2**

[**Project Abstract**](#_heading=h.30j0zll) **3**

[**Project Introduction and Research Question**](#_heading=h.1fob9te) **3**

[**Background**](#_heading=h.gub8vkwbktl9) **4**

[**Literature Review**](#_heading=h.ubxkqgsrqhc3) **4**

[**Study**](#_heading=h.tyjcwt) **6**

[**Singlepoint.**](#_heading=h.jgnr1vsmyns3) **6**

[**Two-point CrossOver.**](#_heading=h.7plghtwuwpqh) **7**

[**Uniform CrossOver.**](#_heading=h.1s7mf8puzj6l) **7**

[**Whole Arithmetic Recombination**](#_heading=h.t2umh251ae2i) **7**

[**Davis’ Order Crossover (OX1)**](#_heading=h.flklq497t1p6) **8**

[Fitness Function](#_heading=h.fdrklzr6u7bc) 8

[TerminationFunction](#_heading=h.flmdnuxkxo7) 8

[**Project Description**](#_heading=h.10ujjyygcsqx) **9**

[**Project Milestones**](#_heading=h.1t3h5sf) **12**

[**Results and Discussion**](#_heading=h.4d34og8) **14**

[**Project Review and Conclusions**](#_heading=h.2s8eyo1) **17**

[**References**](#_heading=h.17dp8vu) **18**

# Acknowledgements

I would like to thank the Lecturers of the Games development course for all of the help During this project, especially Ross Palmer. I would also like Przemek Tomczyk for the support on user testing and reviews.

# Project Abstract

My project is about researching and testing the possibility of using a Genetic algorithm to produce a population of NPC which reproduces using multiple generations to create a more realistic Pattern of reproduction and development over time.

I am using this Genetic algorithm to target two NPCs from different generations or the same generation once these NPCs have been selected and are of a viable age for reproduction as well as able to reproduce and of opposite genders, the NPCs will breed and their DNA strings will influence the Child NPC.

Over time these children will all develop towards a preset global goal.

I want to do this because I would like to see the potential ability for a population to be more adaptive to a current environment.

I would also like to one day be able to use this algorithm to create more of a realistic group dynamic within a game instead of a group which have only aesthetic differences but similar backgrounds because they are all derived from one overall best adapted NPC.

With a Group that grows together, you should get a similar NPC but distinctive in their own Core statistics and backgrounds. and these core statistics develop through the lifetime of a population. Instead of having a group which simulates death and just spawns a new NPC,

I want to try to reproduce a child NPC that in theory will have a similar statistic with the parents of the child.

I want to compare what I do with a standard genetic algorithm that is well known.

I also want to have different mutations to compare with while using the algorithm to prove that it is not just forcing the statics. In theory, the mutation should have some sort of change within the Graph output.

# Project Introduction and Research Question

The project that I have chosen is a look into using a multi-objective genetic algorithm to produce a population that grows towards a specific goal.

The potential impact of My research project is that the using the algorithm I have created an Npc population will be able to better adapt to circumstance within an area this could be an event that requires a population to better in a certain area such as being stronger or being faster,

This could create a better adaptive population as a whole compared to some other methods.

The reason that I want to do this research project is that I was interested in genetic algorithms and how they can influence a population and what influences an NPC population within a game and what can be done to improve them for a better game experience.

“The use of a Genetic algorithm to create a growing NPC population with an overarching Goal “-research question

# Background

For my project, I would Like to Research the Possibility of using a genetic algorithm to create an optimal Population of NPCs that are suited to their current environment. It is expected when the NPCs are introduced to a random environment they will over time develop into an optimal state to best survive based on a larger overarching goal of the population.

I want to do this because I would like to see the potential ability for a population to be more adaptive to a current environment.

I would also like to one day be able to use this algorithm to create more of a realistic group dynamic within a game instead of a group which have only aesthetic differences but similar backgrounds because they are all derived from one overall best adapted NPC.

With a Group that grows together, you should get a similar NPC but distinctive in their own Core statistics and backgrounds. and these core statistics develop through the lifetime of a population. Instead of having a group which simulates death and just spawns a new NPC,

I want to try to reproduce a child NPC that in theory will have a similar statistic with the parents of the child.

I want to compare what I do with a standard genetic algorithm that is well known.I also want to have different mutations to compare with while using the algorithm to prove that it is not just forcing the statics. In theory, the mutation should have some sort of change within the Graph output.

# Literature Review

A genetic algorithm is a method used in computing which aims to solve both constrained and unconstrained optimization problems[1].

The method is based on the theory of natural selection which is derived from biological evolution, In which organisms better adapted to their current environment tend to live longer and produce more offspring.

These adaptations are created through a process of selective reproduction which affects its genotype through multiple generations[2,3].

In computing, a genetic algorithm repeatedly modifies a population of individual objects, By selecting objects at random and producing a child object, Deriving a new generation of objects. Through multiple cycles of this process, the optimal solution will be developed by tiny variations and improvements of the individual objects introduced during the “reproduction” section of the algorithm.

One of the issues that comes with using genetic algorithms to control the evolution of an NPC population

is how to control the reproduction of the NPCs in such a way as to not interfere with the NPC population in a non-natural way and a way that makes it obvious that certain NPCs are not being selected for reproduction.[4] The problem that occurs in the reproduction of the NPCs are the plateau of statistics, This is because of too many Npc with similar statistics/traits reproducing together within the group. To control this we can use a few methods one method is to assign each NPC an attractiveness to a certain statistic/Trait and a dislike towards another trait/statistic. This will randomize the population enough to get a varied population and should sufficiently give enough of a statistical change to help encourage more mutations of statistics

Another method that we could use is to simply stop using similar stated NPC When a reproduction phase begins between two NPCs, You can check the statistics of each of the NPCs against each other and if they are within a certain control difference then simply do not allow them to reproduce together

Some of the work that people have done in this area is the [5 ]exploration into was the use of a genetic algorithm to help give a more varied feel and randomness to a game to stop the game becoming too repetitive by the same characters coming up, again and again. They do this by taking a genetic sequence and dividing it into multiple seconds for each part of the card. This genetic sequence is then randomly generated After the sequence is generated it checks some conditions that randomise some of the DNA sequences again based on factors of the card like the type and element of the card.

This genetic algorithm is meant to help and improve the replayability of games because the research project I have chosen to do is not about replayability.

From what I researched to take what I have learned from this paper and uses it to help reproduce more realistically in the NPC group and to transfer statistics and traits in a more fair way this research will also be compounded with a mutation rate to change the group’s stats and traits to create a more realistic and varied group with the same overarching aim in improvement.

I will also take what I learned from this paper and use it as a basis for developing my genetic algorithm it goes into the different ways you can add two DNA together there are multiple ways I can do this I can split the DNA in two add each opposing sides together I can split the sequence in three and then add the ends and randomise the middle there are pros and cons to each of the methods one can be created a better matching child from the parents the other method tho creates more randomness within the NPC population

but this varied DNA sequence will cause the entire population to grow more slowly than the first method.[6]

I have also learned from my research about multiobjective genetic algorithms

this paper goes into detail about how to properly have a genetic algorithm affect multiple objects of a group for the benefit of the individual and the overall improvement of the group.[7]

From the Study of the papers above, I have also looked into the termination of NPC there are a few methods one method is to simply remove the wors NPC every cycle and create a new one base on the Top NPC but change the data with a mutation.

I will not be able to do this because I want to reproduce NPC in a realistic way so I will add an Age cap so when an NPC reaches this age they will die and hopefully, by this time they will have to reproduce and a mutation will affect the Childs statics to improve them.

# Study

For my study, I want to take multiple different groups of NPC populations with randomly generated statistics, and traits and put each of them into multiple predetermined Environments, Once in the environment, the NPCs will exist, reproduce, die based on the predetermined global variable,

From the statistics gathered from multiple groups. I want to then determine if there is an optimal arrangement of statistics and traits to provide.

1. The longest lifespan for the NPCs in their environment.

B. The largest possible population.

C. The longest possible lifespan of an individual NPC.

And from this Data be able to theoretically tailor an RTS game to a players playstyle. Improving the gaming experience.

The knowledge I want to gain from this research project is how the use of a genetic algorithm used to control the growing, maintenance and improvement of a randomly generated NPC population within an environment compares to the uses of a prescription NPC group with non-changing statistics or Traits.

Some of the research that I have done includes different methods of CrossOver of genes in DNA sequences for the genetic algorithm. The following are some of the Methods I researched

### Singlepoint.

A point for the CrossOver is selected on the first parent. The Data beyond this point is the swapped with the second parent. This new DNA string is then inherited by the child of the two parents

Starting DNA string & cross over chosen

1000|1010

1010|0101

New DNA sequence

1000|0101

**------------------------------------------------------------------------------------------------------------------------**

### Two-point CrossOver.

From the first parent’s DNA string two points are chosen. The DNA of the second parent is

Also broken up in the same manner. The centre section of the second parent or the two outside sections of the Second parent are swapped with the equivalent sections of the first parent.

This new DNA string is then inherited by the child.

Starting DNA string & cross over chosen

1000|1010|0101

1010|0101|0011

New DNA sequence

1000|0101|0101

Or

1010|1010|0011

**------------------------------------------------------------------------------------------------------------------------**

### Uniform CrossOver.

Each Genom in the DNA sequence is randomly selected using, for example, a coin flip technique the new randomly made DNA string is inherited by the child

Starting DNA string & cross over chosen

10001010

10100101

New DNA sequence

10000111

**------------------------------------------------------------------------------------------------------------------------**

### Whole Arithmetic Recombination

This technique is usually used for weighted integers. The new DNA sequence is created by comparing each corresponding Genome and applying this formula to get the result( (α+b)/2 ).

If the Genome is a whole number then a coin flip technique can be employed to determine if the floating-point number is rounded up or down

The new DNA sequence is then inherited by the child

Starting DNA string & cross over chosen

1,4,5,6,2,3,4

3,6,6,7,3,1,8

New DNA sequence

2,5,6,6,2,2,6

**------------------------------------------------------------------------------------------------------------------------**

### Davis’ Order Crossover (OX1)

Two random crossover points are selected from the first-parents DNA sequence the centre segment from the first parent is maintained for the new DNA sequence. From the end of the second cross over point, the state of the second parents DNA sequence is added once all of the Genomes are filled in after the second cross over point the sequence is filled in from the start of the new DNA sequence

Starting DNA string & cross over chosen

1,4|,5,6,2|,3,4

3,6|,6,7,3,|1,8

New DNA sequence

1,8|,5,6,2|,3,6

**------------------------------------------------------------------------------------------------------------------------**

### Fitness Function

The fitness section of genetic algorithms is the section which determines which of the population will be selected to reproduce. There are many ways to determine what you want using variables. The fitness function is an equation which takes these variables and boils them down into a singular number which then can be compared to others.

for example, you can have a fitness function which Takes into account the Highest level statistics that you want and then their reproductive chance these two numbers combine will give a single number which then can be used to determine which is the best possible NPC to reproduce with

Another use for the Fitness function is to determine when the population has reached a point where you are happy it has completed its goal. Because The fitness function is a representation of all of the variables that determine how good an object is this fitness function can be used as a gauge to see how well the object has improved over time as well as when it has reached its predetermined endpoint.

### TerminationFunction

Another aspect of genetic algorithms is the termination of an object each cycle; the worst of the group is terminated and replaced with an NPC that is based on current improved NPC this can be done by taking Two high fitness function NPC and giving the New NPC a mutated Genome from the parents.

# Project Description

To evaluate the project I am going to use a prescripted set of NPCs with multiple varied status and traits. This prescripted NPC set will be used as a baseline to make a comparison against the NPCs generated by the genetic algorithm.

These NPC in the system will act in the same way and reproduce with the same NPC and will be affected by things with non-varied endpoints and starting points.

Some of the Statistics that the NPCs Have are Speed, Strength, Intelligence and Size. Some of the Traits that are affected by these statistics are the chances an Npc will live past its age cap, How fast the NPC can go this is based on the size and speed, The chance of not taking damage by an event is affected by the intelligence statistic.

For example, if an NPC is born it will live the exact same amount of time as the previous and future NPCs and will act in the same way as previous and future generations. All of the statistics that affect the NPC group as a whole will be standardised and will not vary if the population has a max population size it will not grow past that size. This will give a standardized base of what to compare the genetic algorithm too.

The algorithm that I want to implement is a genetic algorithm which takes a random set of generated NPCs within this group. There will be a set of randomly generated statistics and traits. These statistics will include (health, speed, strength intelligence, age, size), And the traits will include( a preferred trait, a disliked trait, children per cycle, reproductive cooldown).

The NPCs will also pair off for reproduction randomly like I previously stated this approach will anticipate enough of a pool of varied statistic and traits that when reproduction happens and controlled there will be a less likely chance of a plateau in the traits and statistics of a unique NPCs and of the NPC group as a whole leading to more realistic improvements.

For NPC Reproduction I will also use up to two different Algorithms and Two different Mutations for the algorithm I will use the one I design and then One that takes the highest NPC with the statistic that you want. the mutation I will use is a ½ split mutation that is described in the Study section as well as an average mutation which is also described in the study sections

To give the tests a more realistic real-world scenario I will have timed world events that will target specific NPCs with specific dominant traits. The effect I hope that these events have on the NPC group is that it will give them enough of a realistic variable that will affect the groups NPCs in such a way that it will remove enough NPCs in the group to again stop the genetic algorithm for plateauing. To determine if plateauing is occurring. These events will affect a certain statistic. If the NPC has a high static of this event type it will take a certain amount of damage that is based on how high that statistic is.



For the visualization of the project, I am going to make an SFML window with will represent the NPCs groups area the NPCs will be represented by little NPC images one for male and one for female I will have a standard background the NPCs will also have specific areas that they will go to reproduce the NPCs when in the reproduction will have an empty heart above them once they find a partner they will have full hearts above them they will go to a house where they will reproduce and after reproduction, the NPC will have two hearts above the head this will stay as long as they are in there reproduction cool down once they are out of the reproduction cool-down they will return to normal.

during normal events, the NPCs will wander around the map simulating tasks that should be done

NPCs that die during the simulation will go grey and then disappear

The data for all of these tests will be displayed on the sides of the screen the data will be displayed as simple text as well as some minor bar graphs mean age, amount of males amount of females etc

Some of the indicators that the genetic algorithm is improving the experience of the NPC group to be more realistic is that the NPCs will have more specific genetics based on a predetermined goal.

Another indicator is that the NPCs group might die out if the right conditions are met.

For my project, I would Like to research the possibility of using a genetic algorithm to create an optimal Population of NPCs that are suited to their current environment. It is expected when the NPCs are introduced to a random environment they will over time develop into an optimal state to best survive based on a larger overarching goal of the population that is predetermined by the user on the set up of the global variables.

To start the project I want to create a closed environment. Within this environment, I will be able to control all the starting variables of the environment and then after the start of the test, the environment becomes completely independent and continues on its own until it completes its overarching goal or until the tester stops the test due to complacency or a plateau in improvement.

# 

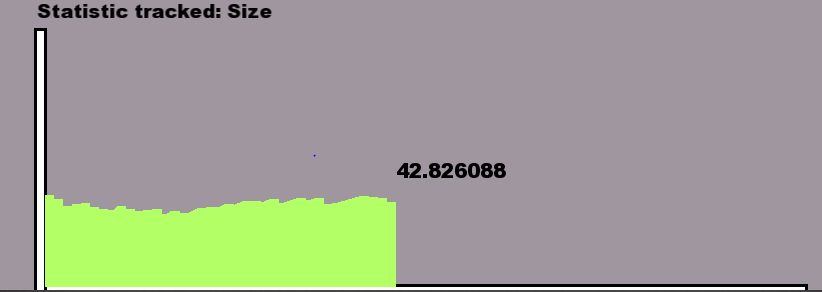
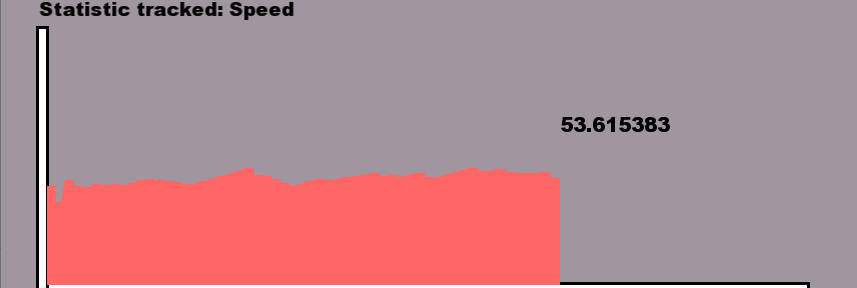
Some of the variables within this environment are the max Distance of breeding NPCs the Mutation rates of all the statistics. What statistic is Wanted, Event timer, Which Algorithm which mutation, Reproduction chance, End Goal

From this enclosed environment, I want to be able to monitor the NPC As they reproduce and grow. I will have a sidebar with relevant information such as total males total females total population of all time current population as well as run setting such as mutation rates and what statistics are the NPCs growing towards on this side panel I also want to display a timer that counts down to the next event as well as text which will display what the next event is and what the current event is as well as what percentage of the population has been affected by the current event.

I also want to indicate the affected NPC in the Environment. I will do this by highlighting them with a circle.



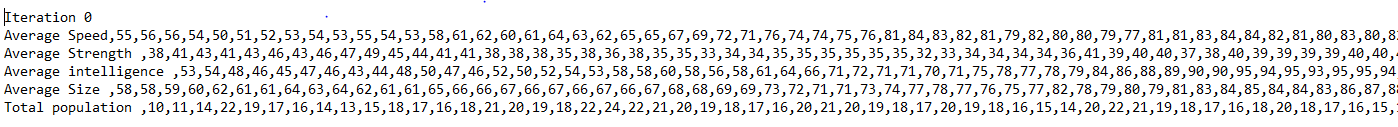
With the environment setup and relevant data being displayed on the side panel underneath, I also want to track statistics of the NPCs. to do this I am going to make a graph which the user can switch between With the left and right arrow keys. This graph will track the current selected average statistic. This graph will update with every change in the population.



With all this tracking of data I also want to keep a log of it in the background of the run I will log all of the Data for each statistic and the current total population at the end of a run it will output all of their data to a text file as well as a settings file which holds all of the current settings of the Run.

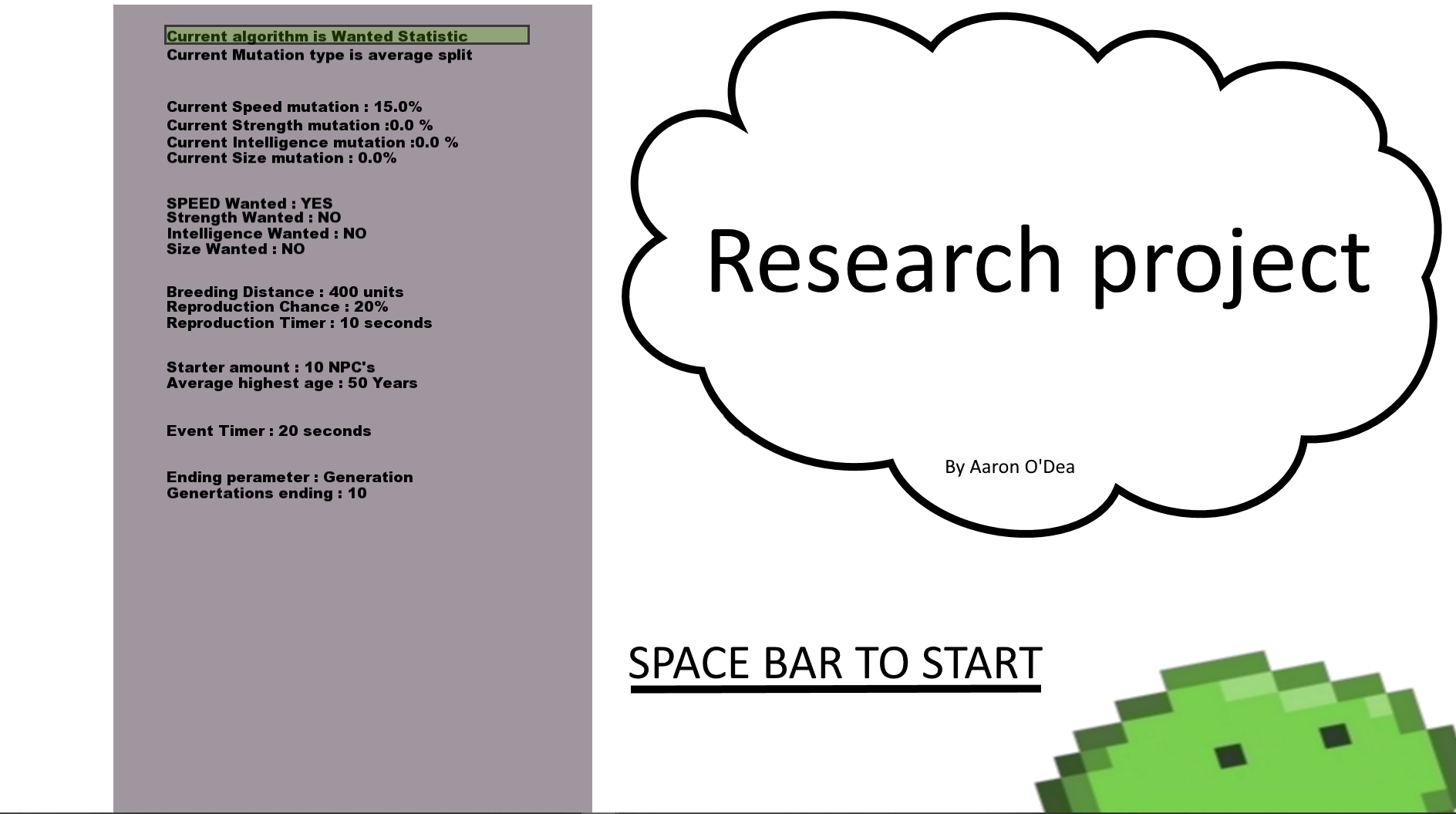
Because I will have to perform multiple runs of the system to gather data I will include a way to reset the current run and start a new run with the same setting.





with the outline described above, I will also have to have the main menu which will allow a user to choose all of the

Global statistics that have been previously described include Mutation Rates age caps, which statistic is wanted as well as other variables which affect the NPCs while they are in their environment.





From the above statement, slight alterations to the systems might have occurred such changes like traits which were removed and replaced with a better Statistic system for the NPCs.

From the project, I learned a lot about genetic algorithms and differences in genetic algorithms as well as what different mutations and how certain external variables can be used to alter them.

Some of the technical achievements that I have made in creating a genetic algorithm is that I created a genetic algorithm that takes more into account than the best NPC but instead takes a more realistic adaptive approach. My algorithm also improves a group rather than each individual NPC which is then copied to make a group.

# Project Milestones

If anything is not completed within the milestone the task is then assigned to the next milestone list.

**Milestone 1 (End of October 2019)**

For the first milestone of the project I wanted to get a basic structure of the project operational, this includes the initial game loop of the project which allows me to render updates and take keyboard inputs from the user.

I also want to create a basic NPC with a genome. This genome includes 4 sets of DNA that are specific values. These NPCs should have a texture and be able to wander around a specific area.

**Milestone 2 (End of December 2019)**

For the second milestone, I wanted to include my algorithm and a mutation which allows the NPC to reproduce based on variables such as age, distance, reproduction cycle, gender and wanted statistic this wanted statistic would become the base for my fitness assessment based on the strength of the state and how many wanted stats the current NPC has the chance of reproduction increase. I also want a basic mutation which takes each Genome sequence from each parent NPC and produces a new genome and adds a mutation to it which affects the child NPC.

**Milestone 3 (End of February 2020)**

For my third milestone, I would like to implement a graph which tracks the average statistic of the population. This graph should increase and decrease with each change in population. The graph should be able to switch between each statistic with a keyboard input as well as log the data in a background file for future use.

The highest NPC of the currently Tracked statistic should be highlighted on the screen by an indicator.

I also want to show specific statistics on the side panel of the screen E.g Total population female, male split and the highest reached population within the current run.

I also want to set up a basic menu with the ability to determine some statistics within the system such as age cap and reproduction time.

**Milestone 4 (End of March 2020)**

For my fourth milestone, I want to introduce a second mutation and a second algorithm. This will be my comparison for my algorithm.

I also want to add in events to the game that happen at set intervals to the NPCs. These events will affect the NPCs with high statistics of that event type (speed events target NPC’s with a high statistic in speed). I also want to show the percentage of the NPCs affected on the side panel as well as what is the current event and what is the next event. I also want to highlight each NPC that was affected by the current event.

I want to add more data to the main menu which the user can choose such as mutation rate, what statistic is wanted and other data that might affect the run.

I want to display the important information for the run on the side panel of the game screen. This includes mutation rates, statistics, wanted what type of algorithm and what type of mutation is being used.

**Final Milestone (29th of April 2020)**

For my final milestone, I want to output the collected data and have a way of restarting the Current run with the same settings.

During this milestone, I also want to address any problems that I have found such as overpopulation render lag. Population caping.

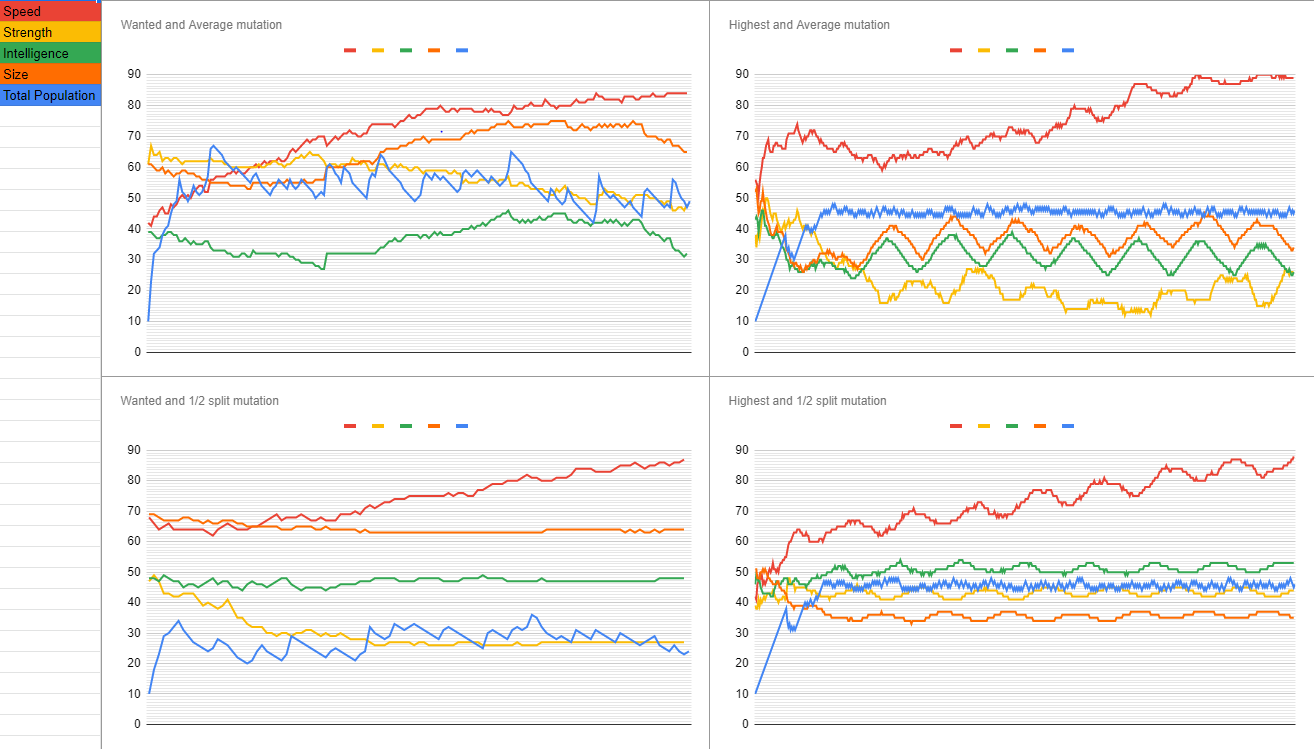
I also want to go back over the code and clean up anything that could be improved and reduce code size where appropriate as well as comment on anything that was missed.

I also want to add anything that might make the Simulation look better colours to indicate statistics and so on.

The final thing I want to do is run Doxygen on the code to produce a file which should make my code easier to understand.

# Results and Discussion

[Statistics](https://drive.google.com/open?id=1r7-gLxFz2D7vO-gM68zBUP6AkJWcbF9JqI8KnobQVUE) all of the results that are being referenced and conclusions that have been drawn are in reference to these results.



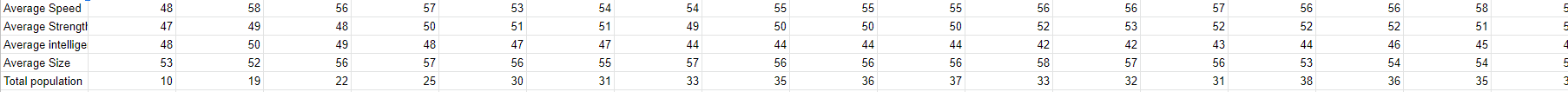
The Above graphs are a comparison of the four different combinations of Mutations and Algorithms.

To gather this data I made a total of nine runs per Combination, From these results I choose the one result that would be the most representative of the data gathered in total.

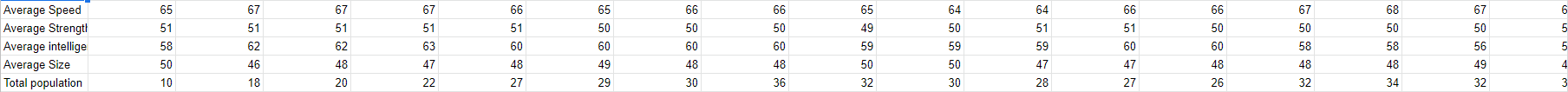
To gather the data I used my inbuilt Code which outputs a Text file after a run this file contains all of the averages for each statistic this included Speed, Strength, Intelligence, Size. as well as the current total population as the run progressed. The settings that I used also output to maintain clear governance of each run through.

Some of the raw data gathered is shown below (One run from each combination)

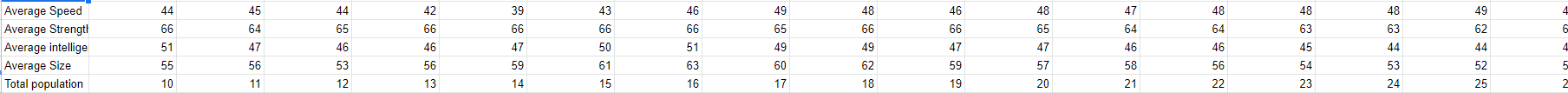
Wanted algorithm and Average mutation.



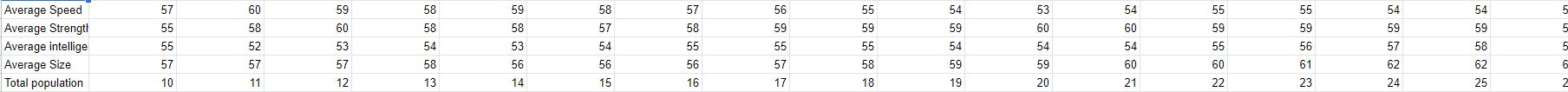
Wanted algorithm and ½ split mutation.



Highest algorithm and Average mutation.



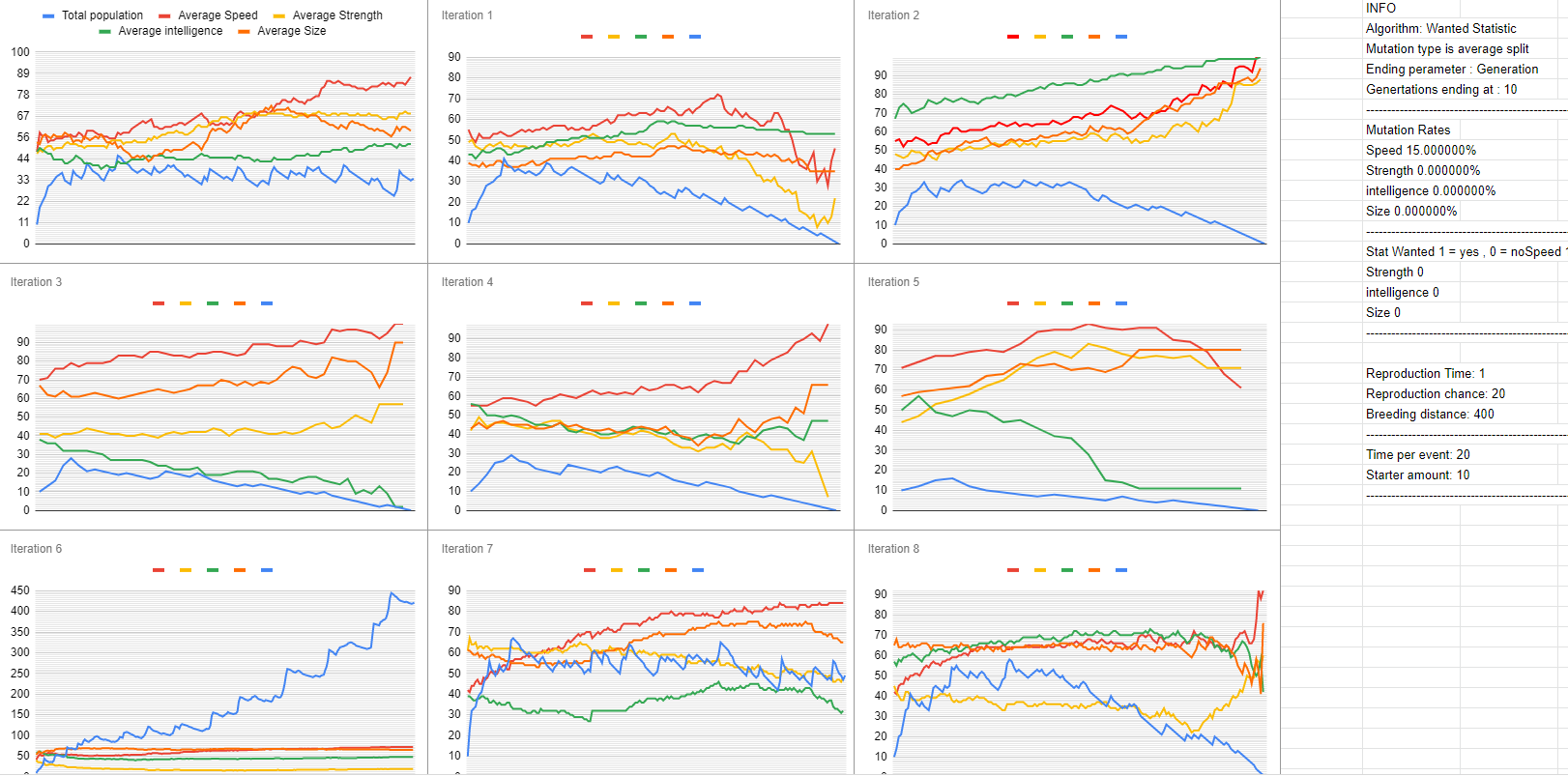
Highest algorithm and ½ split mutation.



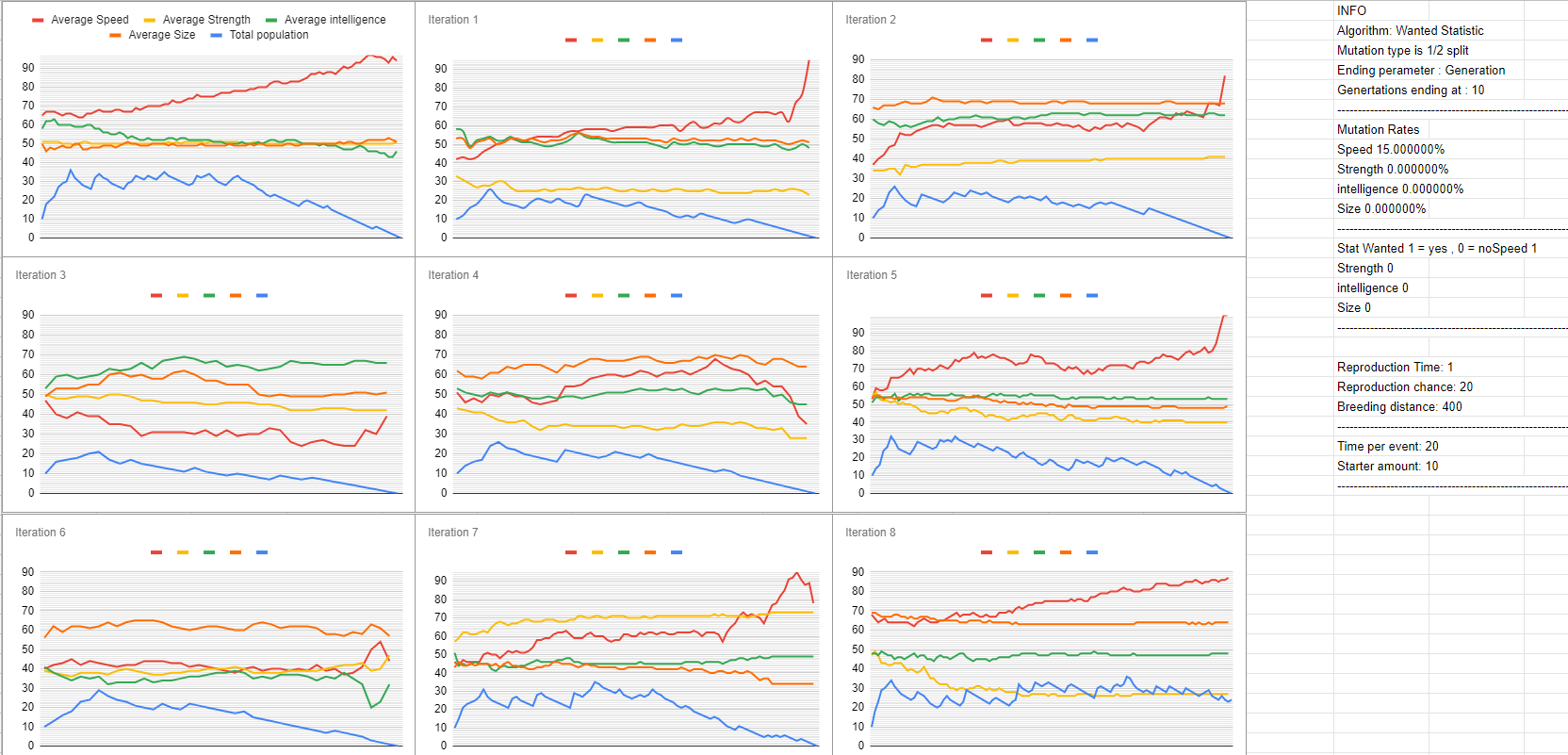
The data above only shows a section of the total Output.

From this data, I converted each of the raw data sections into a graph. Shown below.

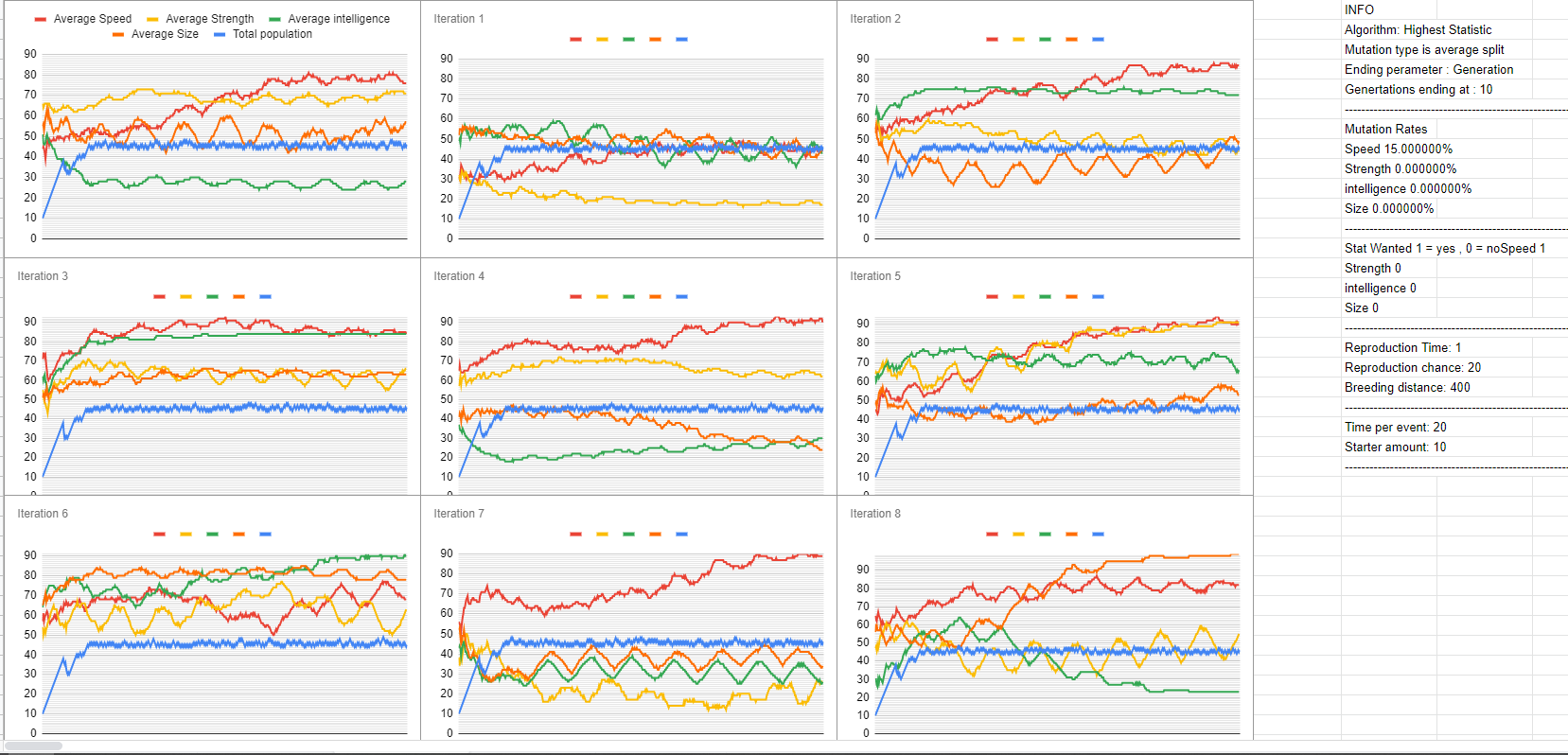
Wanted algorithm and Average mutation.



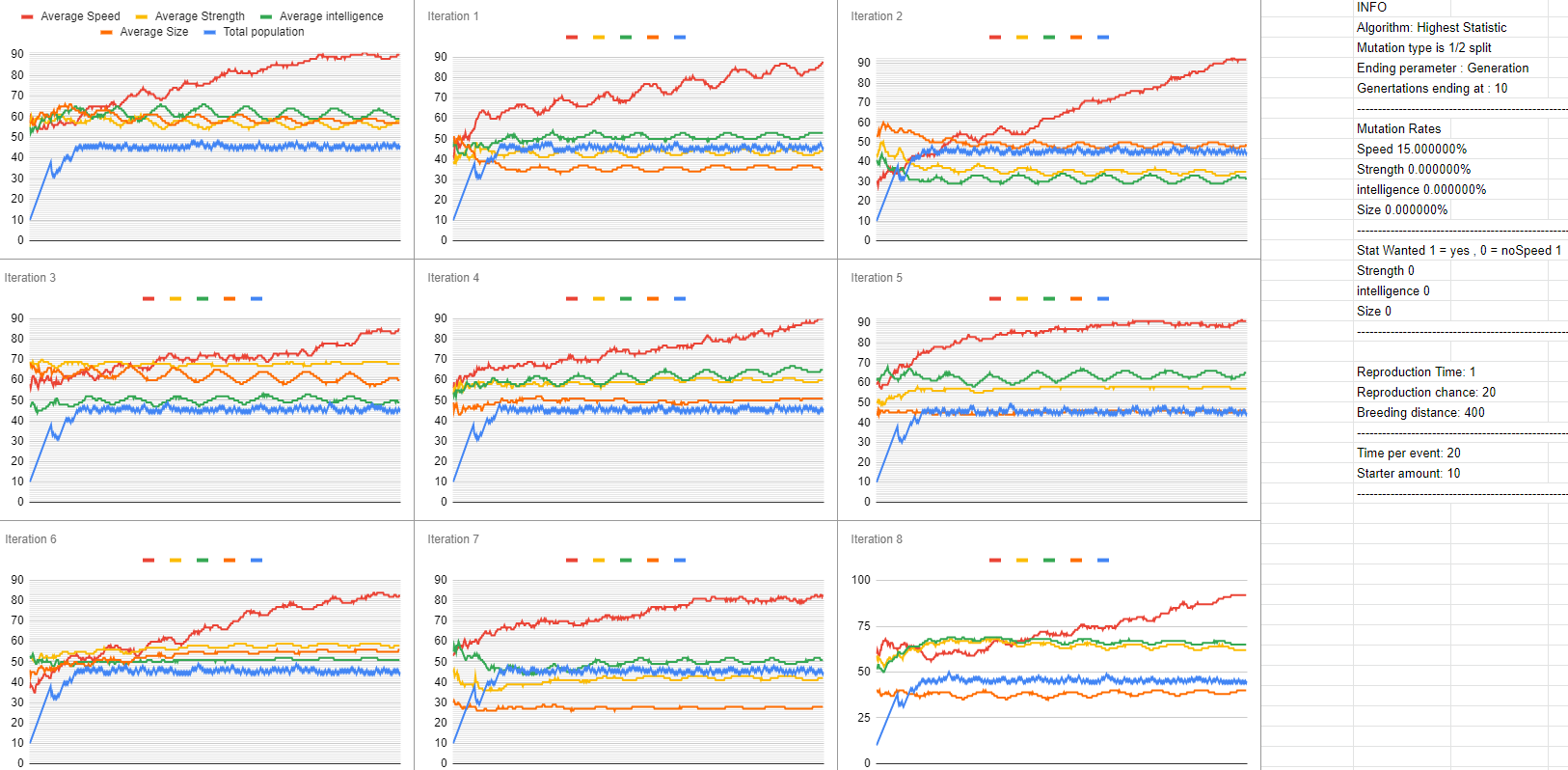
Wanted algorithm and ½ split mutation.



Highest algorithm and Average mutation.



Highest algorithm and ½ split mutation.



From the Data gathered I concluded that my algorithm (Wanted algorithm ) causes the NPC population to grow rapidly and then maintain an average that statistic that is wanted grows in a linear curve rapidly while the statistics that are unwanted maintain an average and fluctuate at the increase and decrease of the population.

while the Highest algorithm produces a steady linear curve a lot slower but produces a larger total population

The mutations influence each algorithm as well but this influence mostly affects the statistics to show how a ½ split causes a slight dip and a slow curve up in the statistic wanted while the average mutation causes deeps each time the older generations die off and the average of the newer NPCs are shown more prominently.

I determined that for a long term steady growth most optimal and a ½ split mutation is best suited.

While a quick growth in a short amount of time the most Appropriate and average mutation.

All of the settings that are determined before each run influence the data as well such as the highest population for age cap and growth in a state by the mutation applied.

# Project Review and Conclusions

In conclusion from the data that I have gathered I show a quicker Growth using either of the two different types of genetic algorithm mutation types.

My Algorithm which takes multiple different variables into account when picking another NPC’s to reproduce with.

This growth is specified to a certain statistic of an NPC’s which is predetermined in settings

I determined that for a long term steady growth most optimal and a ½ split mutation is best suited.

While a quick growth in a short amount of time the most Appropriate and average mutat

ion

The problems that occurred were how to determine what has improved and based on what statistics.

How to measure conclusions of runs in reference to other run and population explosions skewing the data as well as population dying too soon.

I would advise someone else who wants to do a similar problem to create a better environment for the NPCs and an environment which influences the NPCs in a better way. I would also advise the person to look into how to control a larger population of NPCs in a way to not lag the system.

if I was starting again I would do what I stated above as well as coming up with more variables which could affect the population

I feel that the technologies used were sufficient in displaying and outputting data in the future I would want to use XML for this so I could also maybe input data into the system

if a future student wants to undertake a project similar to this they might also want to see if they could introduce a neural network that would take the data from the previous run and change variables to try to find a most optimal set of settings for a certain high-level goal.

# References

|  |  |  |
| --- | --- | --- |
|  |  |  |
| website  [1] | 1. (MathWorks 2019) | MathWorks-(2019-10-6),  Genetic algorithm[online]  ( [https://www.mathworks.com/genetic-algorithm.html](https://www.mathworks.com/help/gads/what-is-the-genetic-algorithm.html))  (Accessed 2019-10-6) |
| website  [2] | 1. (Encyclopaedia 2. Britannica 2019) | 1. Encyclopaedia Britannica-(2019-10-6). 2. Natural selection[online]   (<https://www.britannica.com/science/natural-selection>)  (Accessed 2019-10-6) |
| website  [3] | (Encyclopaedia  Britannica 2019) | 1. Encyclopaedia Britannica-(2019-10-6).   Genotype[online]  ([www.britannica.com/science/genotype](https://www.britannica.com/science/genotype))  (Accessed 2019-10-6) |
| Report  [4] | (Evolutionary Algorithms — How to Cope With Plateaus of Constant Fitness and When to Reject Strings of The Same Fitness 2009) | Thomas Jansen, Ingo Wegener TJ IW. (2009). Evolutionary Algorithms — How to Cope With Plateaus of Constant Fitness and When to Reject Strings of The Same Fitness 2009. FB Informatik, LS 2, Univ. Dortmund, 44221 Dortmund, Germany. |
| Report  [5] | Monsters of Darwin: a strategic game based on ArtiﬁcialIntelligence and Genetic Algorithm |  |
| website  [6] |  | <https://towardsdatascience.com/introduction-to-genetic-algorithms-including-example-code-e396e98d8bf3> |
| Website  [7] | A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II |  |

# 