Artificial intelligence ACW   
  
Background:

What is a genetic algorithm? A genetic Algorithm is a for of search-based technique original developed by Professor John Holland in the 70s. these algorithms are based on human evolution and work by taking the best results and merging them together, essentially taking the best genes and trying to create the best kind of data like we do with DNA.  
The way a Genetic Algorithm works involves a number of recursive steps:

1. Initial Population – this is where we create our ancestors, the original data.
2. Fitness Calculation – this is where we evaluate the data before us and give it a score based on the desired outcome.
3. Tournament – this is where we select four random members of the population. Next, they are sorted by the fitness score to determine which to are the best. These become the new parents
4. Crossover/Mutation – this is the final stage; the two best data sets are merged into a child by taking the first half of parent 1 and the second half of parent 2.
5. Mutation – This is the final stage of the algorithm, here the function we randomly alter one value in the data of the child, this is to make sure that the data becomes the best it possible can down the line.

The algorithm will recursively go through these functions till it meets the desired requirements which tends to be when the variation in the population is less than 0.05.  
The algorithm in this project will be built by attempting to use a object- orientated style methodology, this is because it will allow for greater autonomy and will be a more secure system.   
The type of crossover in this project will be single-point crossover.

Experimental Design:

The variable that will be changed in this experiment will be the size of the population. The way to do this will be to change how many times the for loop which adds elements to list iterates, for example from 100 to 1000. The aim will be to see if this changes the results dramatically. Will having a higher amount of data provide us with better children and therefore closer and better results or will the amount of data skew the scores and throw everything off course.  
These parameters were chosen as it is the simplest way to manipulate the results but it could also provide the most interesting and significant results, this is because the more data the algorithm can process the more it can perfect the score. This is a similar concept to our race, the more humans there are the more DNA variants are which in turn means there are more desirable DNA combinations which help better our chances of survival.

Experimental Results:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Population Size = 100** | **Population 1** | **Population 2** | **Population 3** | **Population size = 200** | **Population 1** | **Population 2** | **Population 3** |
| **Gen 1** | 0.099514522 | 0.049305529 | 0.007932083 | **Gen 1** | 0.001822393 | 0.002722888 | 0.099737935 |
| **Gen 2** | 0.098059661 | 0.002422151 | 0.002722884 | **Gen 2** | 0.004374021 | 0.002663206 | 0.008104996 |
| **Gen 3** | 0.099514522 | 0.049305529 | 0.007932083 | **Gen 3** | 0.001822393 | 0.002722888 | 0.099737935 |
| **Gen 4** | 0.003179879 | 0.013878723 | 0.099831845 | **Gen 4** | 0.00811852 | 0.142802285 | 0.002540497 |
| **Gen 5** | 0.003066229 | 0.01770606 | 0.023074121 | **Gen 5** | 0.005485146 | 0.043469294 | 0.001641466 |
| **Gen 6** | 0.008697275 | 0.00965447 | 0.007579367 | **Gen 6** | 0.009705347 | 0.013905956 | 0.02151963 |
| **Gen 7** | 0.099906963 | 0.015545803 | 0.004224163 | **Gen 7** | 0.010523919 | 0.00283701 | 0.008256694 |
| **Gen 8** | 0.009274513 | 0.002700521 | 0.008377416 | **Gen 8** | 0.014385275 | 0.002717941 | 0.008132833 |
| **Gen 9** | 0.099902122 | 0.007956846 | 0.019610655 | **Gen 9** | 0.160125148 | 0.002722368 | 0.002690432 |
| **Gen 10** | 0.099912499 | 0.009959145 | 0.099955275 | **Gen 10** | 0.007391838 | 0.012785789 | 0.032316206 |
| **Gen 11** | 0.004443031 | 0.008132797 | 0.002566171 | **Gen 11** | 0.002495152 | 0.00246094 | 0.008577126 |
| **Gen 12** | 0.008105006 | 0.006322729 | 0.008471969 | **Gen 12** | 0.019854389 | 0.097174814 | 0.002711211 |
| **Gen 13** | 0.009512082 | 0.008131859 | 0.191941383 | **Gen 13** | 0.002721939 | 0.002722713 | 0.099890464 |
| **Gen 14** | 0.002722885 | 0.072553436 | 0.004847448 | **Gen 14** | 0.00269623 | 0.06286671 | 0.002547051 |
| **Gen 15** | 0.034580267 | 0.008126253 | 0.018054113 | **Gen 15** | 0.100301847 | 0.002703947 | 0.009309294 |
| **Gen 16** | 0.099888105 | 0.03967875 | 0.002686458 | **Gen 16** | 0.00269734 | 0.008131869 | 0.079844427 |
| **Gen 17** | 0.009828277 | 0.038473396 | 0.008132193 | **Gen 17** | 0.008143539 | 0.00834546 | 0.001977757 |
| **Gen 18** | 0.011363333 | 0.09989279 | 0.002554237 | **Gen 18** | 0.002722795 | 0.002687719 | 0.002721634 |
| **Gen 19** | 0.00988991 | 0.018136565 | 0.037964281 | **Gen 19** | 0.028378371 | 0.011204663 | 0.099891728 |
| **Gen 20** | 0.009196035 | 0.002651482 | 0.00783756 | **Gen 20** | 0.044272641 | 0.011989323 | 0.025607015 |
| **Gen 21** | 0.002719521 | 0.009124002 | 0.084687642 | **Gen 21** | 0.008542911 | 0.002413941 | 0.142802285 |
|  |  |  |  |  |  |  |  |
| **Lowest value** | 0.002719521 | 0.002422151 | 0.002722884 |  | 0.001822393 | 0.002413941 | 0.001977757 |
| **Highest Value** | 0.099912499 | 0.072553436 | 0.191941383 |  | 0.100301847 | 0.142802285 | 0.142802285 |
|  |  |  |  |  |  |  |  |
| **Average** | 0.039203649 | 0.023317087 | 0.030999207 |  | 0.021265769 | 0.021050082 | 0.036217077 |

Conclusion:  
These results show that a larger population gives us better results, I feel that if I managed to get the algorithm to work correctly then this would be even more prevalent than it already is. If I had the chance to re do this work, I would write a plan at the start on how I want to format it and go through. One of my initial problems was deciding how to design the code and in the end I didn’t do that I just went and built it. This caused me to waste a lot of time. To get the best results I think you would need a computer with a lot of power and to make the population size over 1000 as this would generate a lot of data.  
  
I feel in the future genetic algorithms will be heavily refined for use in CCTV and aiding policing with things such as tracking terrorists or organized crime. I think it would also be useful for tracking stolen cars and such. Other types of AI I have learnt about in the course will be useful in the medical field such as machine learning. And I think a combination of these two could give us a perfect self-driving car.