# Project Proposal

### Sion Williams

## Introduction

### Rationale

Video games have always been designed and built to offer some form of single player experience. Although many advances have been made in AI, humans still prefer competing against other human players. Machine learning is a subclass of AI that is concerned with designing and constructing algorithms that allow the computer to learn. Although only implemented in a few games they have given interesting results.

## Literature Review

### Bayesian Networks (Belief Network)

Bayes Theorem, developed by the Rev. Thomas Bayes, an 18th century mathematician and theologian, was first published in 1763 . Mathematically it is expressed as:

* P(H|E) is the conditional probability of H, given E. It is also called the posterior probability as it’s derived from or dependant upon E.
* H is the hypothesis or theory of interest.
* E represents a new piece of evidence that either confirms or disconfirms the theory.
* P(H) is the prior probability or marginal probability of H.
* P(E|H) is the condition probability of E given H.
* P(E) is the prior or marginal probability of E, and acts as a normalizing or scaling factor.

The Bayes theorem offers a form of conditional probability by relating the conditional and marginal probabilities of events H and E, where E has a non-diminishing probability [1]. It is useful because there are many real world examples where the probability of one event is conditional on the probability of a previous one.

While the Bayes theorm can anticipate the factor of conditionality, in many cases calculations are nondeterministic polynomial-time hard [2]. It may be possible to manage a scenario with 5 discrete random variables but a system monitoring 37 variables would not be managable [2]. This is where a Bayesian network is useful. A Bayesian network is a graphical model that encodes probabilistic relationships among variables of interest [3] and can be used to learn casual relationships, and hence can gain understanding about a problem domain making it a very useful control method in A.I.

### Genetic Algorithms

Genetic algorithms (GAs) are search algorithms based on the mechanics of natural selection and genetics as observed in the biological world. Charles Darwin proposed the theory of evolution in his work titled “On the Origin of Species” in 1859. He proposed that those that are most able to survive in their environments are able to pass on their traits to the next generation. The individual traits are encoded in chromosomes. In the next generation, these chromosomes are combined in a process called crossover. Crossover is a recombination of the chromosomes in the offspring.

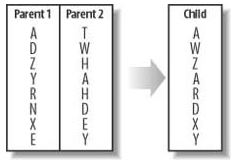


Figure 2 Crossover

shows each parent passes half of its genetic material on to the child. However, in the real world this crossover process might not be exact. Random mutations also can take place. Random mutations are nature's way of trying new things. If a random mutation improves the species, it gets passed on to future generations. However, if not it, the particular trait doesn't get passed on.

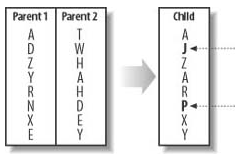


Figure RANDOM MUTATIONS

This constant recombination of chromosomes from the most successful members of the previous generation, combined with random mutations, creates future generations that are better adapted to survive and flourish in their environments.

GA’s use both “survival of the fittest” or crossover and randomisation to robustly explore a function. One example of their use in games is described by Nicholas Cole et al. Where they discuss their use for tuning parameters in a “Counterstrike” bot (AI controlled non-player character).

### Artificial Neural Networks

In 1943 Warren S. McCulloch and Walter Pitts released a paper “A logical calculus of the ideas immanent in nervous activity” where the article introduced the first mathematical model of a biological neuron. Later, in 1958 the first practical application of the artificial neural network was presented by the AI community , the perceptron. The perceptron was created by Rosenblatt to model the human vision system [6].

Neural networks have since seen many applications from aerospace technology as high performance autopilots [7] to truck brake diagnosis systems [8] in the transportation industry. Their uses are numerous but they have encountered many problems in video games. One of the biggest problems is the large number of different architectures and the time costly matter of debugging errors, many developers still believe neural networks are too resource intensive [9].

### Multi Layered Perceptron

The multi layered perceptron is a feed forward network. In its simplest form a multi layer network will have an input layer (which takes information from a given example), a second or hidden layer (which receives its inputs from input/first layer), and an output layer (which takes outputs from the hidden layer to produce the networks overall results). The input layer is defined by the number of inputs provided by the example and the outputs are defined by the number values to be predicted.

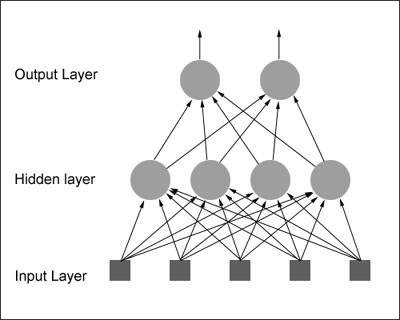


Figure MULTI LAYERED PERCEPTRON

### Supervised Learning

Each network undergoes training; the reson for this is to ensure that the correct response is obtained for a given input. Supervised learning is a learning paradigm for providing a network with a set of example data which provides input and expected outputs. These examples are known as training sets.

One form of supervised learning is known as the Backpropagation learning algorithm. The Backpropagation algorithm works by propagating the error back through the network, effectively changing the weights of the inputs and biases.

## Aims

### Overview

This project aims to assess the competence of artificial intelligence techniques (for example neural networks) in controlling an agent in a video games related scenario. The agents behaviour must look realistic and avoid collisions. The algorithm should also be able to generalize from examples to cover all eventualities i.e. change of scenario.

## Objectives

* Research AI techniques applied to games for agent control
* Select appropriate technique
* Design a games related test scenario and suitable performance criterion
* Evaluate the control of the agent according to the selected criterion

### Scope

The autonomous agent will be tested in a simulated environment where variables can be stored for further analysis. A visual representation of the simulation will be provided by OpenGL. For evaluation purposes the track times and path of the vehicle will be recorded.

|  |  |
| --- | --- |
| In Scope | Out of Scope |
| Explore all methods available for controlling an agent in a game environment | Design and test a new form of AI |
| Outputs from the AI controller will control acceleration/deceleration and turning left/right | Vehicle dynamics |
| Collision detection (line intersection) to provide sensory input to AI controller | Physics resulting from collision |

## Resource

This project will be undertaken on a Windows XP personal computer installed with Visual studio 2005. The OpenGL graphics library will be employed to handle all graphics rendering and if required any vehicle dynamics will be obtained from a 3rd party source.

## Project Plan

Whilst I am aware that this timetable is unlikely to be very accurate past the first few months and subject to significant development, this table presents my proposed timetable to submission.

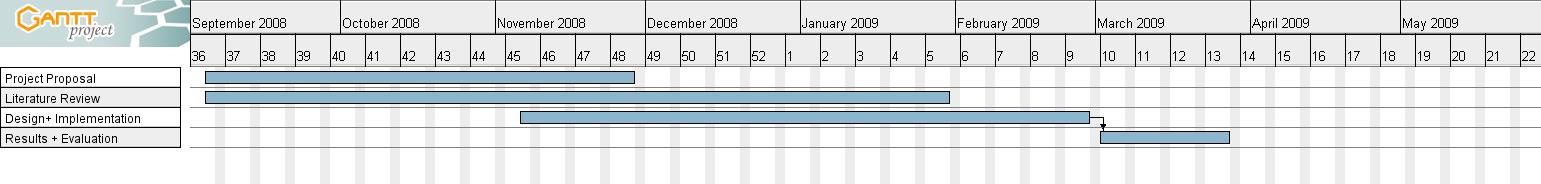


Figure Gantt Project Plan

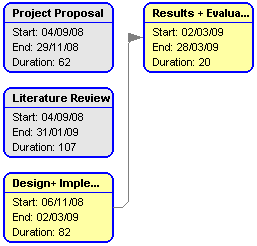


Figure 5 Pert Chart

# Bibliography

1. **J. Stutz, P. Cheeseman.** *A Short Exposition on Bayesian Inference and Probability.* s.l. : National Aeronautic and Space Administration Ames Research Centre: Computational Sciences Division, Data Learning Group, 1994.

2. **Friedman, N. & M. Goldszmidt.** Learning Bayesian Network from Data. [Online] 1998. [Cited: 11 2008, 18.] http://www.erg.sri.com/people/moises/tutorial/index.htm.

3. **Heckerman, David.** *A tutorial on learning with Bayesian networks.* Redmond : Microsoft, 1996. MSR-TR-95-06.

4. **Luger, George F.** *Artificial Intelligence, Structures and Strategies for Complex Problem Solving.* Harlow, England : Addison- Wesley, 2002. 978-0321545893.

5. *Using a genetic algorithm to tune first-person shooter bots.* **N. Cole, S. Louis, and C. Miles.** pages 139 to 145, Oregon : IEEE Press, June 2004, Vol. Proceedings of the 2004 IEEE Congress on Evolutionary Computatio.

6. *The perceptron: a probabilistic model for information storageand organization.* **Rosenblatt, F.** s.l. : Psychological Review, 1958, Vol. 65.

7. **NASA.** NASA NEURAL NETWORK PROJECT PASSES MILESTONE. *NASA.* [Online] 9 2, 2003. [Cited: 11 19, 2008.] http://www.nasa.gov/centers/dryden/news/NewsReleases/2003/03-49.html.

8. **Fausett, Laurene.** *Fundamentals of Neural Networks - Architectures, Algorithms and Applications.* s.l. : Pearson Education, Inc, 2006. 978-81-317-0053-2.

9. **Dewdney, A. K.** *Yes, We Have No Neutrons: An Eye-Opening Tour through the Twists and Turns of Bad Science.* s.l. : Wiley, 1997. 978-0471295860.

10. *Learning internal representation by error.* **D. E. Rumelhart, G. E. Hinton, and R. J. Williams.** Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Cambridge, MA : MIT Press, 1986, Vol. 1.

11. **Martin T. Hagan, Howard B. Demuth, Mark Beale.** *Neural Network Design.* s.l. : PWS Publishing, 1996. 81-315-0395-X.