Project Proposal: Building a game around an evolutionary algorithm

Motivation and Rationale:

The video games industry has been rapidly changing and moving forward ever since its very origins in the early 1950's. As an avid gamer my self I have witnessed the rise to fame of many games and game franchises, and with them the rise in popularity of game genre's, structure's and mechanics. With computers becoming more powerful and more accessible, and barriers to entry for new developers dropping, we have never seen a more rich and diverse market for games.

One of the more broadly successful trends in the progression of games, is the tendency to move away from static content and towards dynamic content. This manifests its self in many ways. For example the huge success of a franchise like grand theft auto can no doubt be partially attributed to its original use of a large open world and the way in which that puts the power in the hands of the user to create their own experience (rather than following a linear path). This concept of user generated experience can be seen on a further level in the highly successful game Mine craft, which provides no story line what so ever and relies entirely on the game mechanics to provide intrigue. The concept of making a game dynamic and adaptive to the user can be seen in many ways across many genre's. But what's the next step for dynamic content?

Game developers these days will often try to give the illusion that the player is affecting the world by giving them a pre-set tree of options in certain situations that affect the narrative or the environment in some way. However, this effect is still some what faked because it is deterministic and hard coded. One thing that I would argue is missing from many games in terms of both realism, and potential for dynamically generated content, is the use of an ecosystem of NPC's (non player characters) within the game that the player has an affect on through their actions. What if the enemies in an MMORPG like World of Warcraft were generated via a genetic algorithm? What if killing loads of a particular species upset the balance of the ecosystem as we would expect in the real world? What if that species would then evolve and adapt in light of that culling. I would like to explore this game mechanic on a smaller scale, as a proof of concept.

I took my inspiration for this project from the many evolution enthusiasts out there who have developed evolution simulations, such as the creators of Gene Pool[1] or Box Car 2D[2]. I believe these simulations provide enough evidence to suggest that an evolutionary algorithm could be an effective solution space search algorithm for generating NPC's for a game.

Aim & Objectives:

Aim: To assess the effectiveness of evolutionary algorithms as a means of generating NPC's (non player characters) within a game.

Objectives:

- Use a set time to accumulate background research into the use of Bio-Computing algorithms in games, with a focus on genetic algorithms.
- Define 3 game ideas that could assist with my aim.

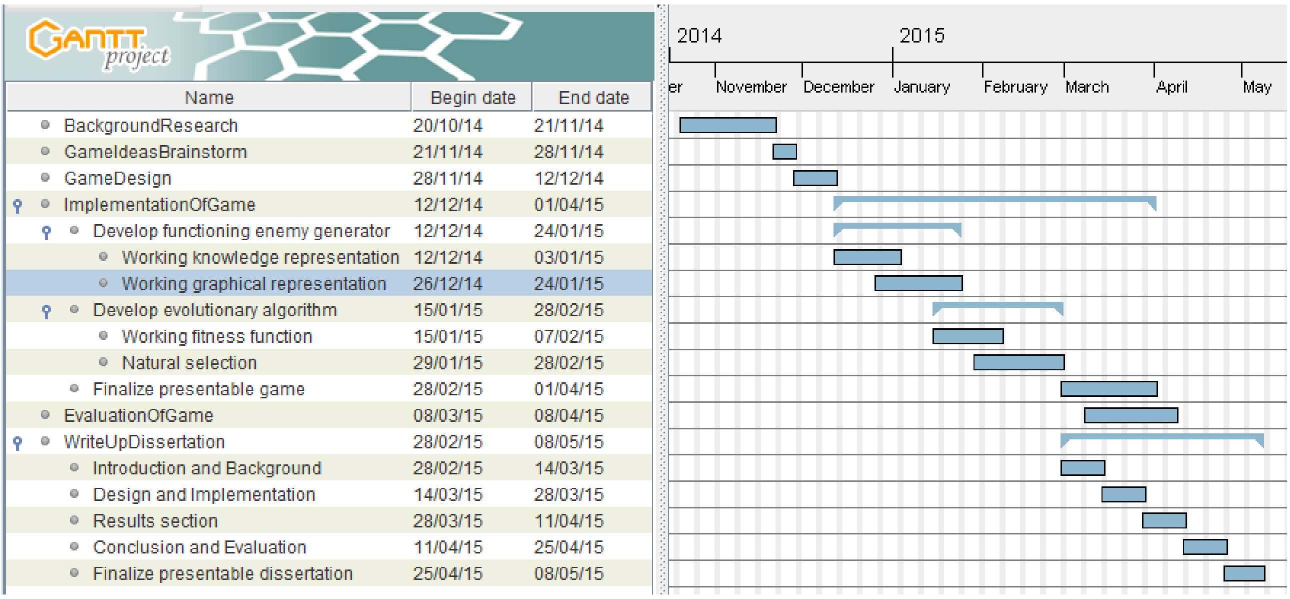
- Finalize a time realistic design for a game that will demonstrate “the effectiveness of evolutionary algorithms as a means of generating NPC species within a game.”
- Implement the game and tune the evolutionary algorithm.
- Reflect on the game and personally evaluate the strengths and weaknesses of the game mechanics and what this tells us about the potential for evolutionary algorithms in games.

Background:

“Evolutionary Artificial Intelligence in Video Games” [4]	This paper offers an interesting insight into the place of evolutionary algorithms (EA's) in games of the future. It argues that as building AI's becomes more challenging (with games becoming more complex), the task of building an AI by hand will become greater than the task of setting up an EA to generate an AI solution. The paper provides many interesting examples of games which use bio-computing concepts, One which was of particular interest to me was Nero[3], which was it's self a research project which I then followed up on.
“Evolving Neural Network Agents in the NERO Video Game” [5]	NERO is a game where by you train a set of soldier NPC's by giving them more and more complex tasks to do and then rewarding the behaviour that you want from them. The NPC's then evolve through a process called neuro-evolution, and you can battle your friends teams online. This paper primarily describes the process the developers went through to build NERO. It was particularly insightful to me because the entities that are being evolved in NERO are knowledge representations for AI's. What I want to achieve in my project is to evolve NPC's in general. These NPC's will need to be made up of a physical form and an AI to control them, so it is important I understand the knowledge representations for both.
“Team Blockhead Wars: Generating FPS Weapons in a Multiplayer Environment” [6]	Team blockhead wars(TBHW) was a game developed specifically for the purposes of this project. The study revolves around generating weapons for a first person shooter game using a genetic algorithm, the fitness function for which defines the best weapons as the most successful and most commonly chosen by the players. I found this paper particularly inspiring because it addresses the issue of the physical form of the weapons by generating 3D meshes for the weapons based on their genetics. Where NERO is a useful study for understanding the knowledge representation for the AI, TBHW is useful for understanding how to generate a physical representation of an NPC in the game world.
“GenePool: Exploring The Interaction Between Natural Selection and Sexual Selection” [7]	This is a very interesting chapter taken from a book called 'Artificial life models in software.' The reason I consider this such an important piece of background research is that it attempts to explain the mechanics of and the behaviours elicited by the NPC's in GenePool: an evolution simulation program which originally inspired me to undertake this project. GenePool helps to visualize

	how evolution works and serves as an education tool for understanding many of its nuances. This software and its explanation helps to visualize why we sometimes see unwanted or unexpected behaviour from evolved NPC's (such as extinction or symbiotic relationships.)
“Evolving 3D Morphology and Behavior by Competition” [8]	In this research paper, Karl Sims attempts to do something quite similar to what I want to achieve. He creates a 3D physical world and generates random NPC's out of box's whose behaviours and morphology are both determined by their genetics. He then pits them against each other in a physical contest and determines their fitness by where they place in a tournament among each other. Finally he uses this fitness to mate the NPC's and create a new generation. Karl helps to answer many of the questions that need to be asked when creating an evolution simulation for NPC's, hopefully leaving me with good enough guidance to wrap a game around a similar concept.

Work Plan:



Name	Begin date	End date
BackgroundResearch	20/10/14	21/11/14
GameIdeasBrainstorm	21/11/14	28/11/14
GameDesign	28/11/14	12/12/14
ImplementationOfGame	12/12/14	01/04/15
Develop functioning enemy generator	12/12/14	24/01/15
Working knowledge representation	12/12/14	03/01/15
Working graphical representation	26/12/14	24/01/15
Develop evolutionary algorithm	15/01/15	28/02/15
Working fitness function	15/01/15	07/02/15
Natural selection	29/01/15	28/02/15
Finalize presentable game	28/02/15	01/04/15
EvaluationOfGame	08/03/15	08/04/15
WriteUpDissertation	28/02/15	08/05/15
Introduction and Background	28/02/15	14/03/15
Design and Implementation	14/03/15	28/03/15
Results section	28/03/15	11/04/15
Conclusion and Evaluation	11/04/15	25/04/15
Finalize presentable dissertation	25/04/15	08/05/15

So far, the bulk of what I have achieved lies in my background research. Having taken a module in Bio-Computing, I've been able to gain a broad understanding of the differences between Bio-Computing algorithms. My coursework for that module was to compare the effectiveness of evolutionary algorithms with neural networks in solving a classification problem. Though not directly related to the problem I'm trying to solve in this project, just spending the time implementing and analysing both approaches gave me a good understanding of what kind of problems each are more suited to solving and how one might go about implementing and tuning an evolutionary algorithm. Further to this I have also been able to read many interesting papers for my background research (some of which are listed above) and gathered further understanding of what is being done in this field already, where the pit falls lie, and what isn't being done. I've had chance to work with two Java libraries for evolutionary algorithms: JGAP and ECJ. I've also spent time testing the ease with which I can use Unity as a development tool for the project. I've created a simple application which generates 3D NPC's out of sphere's and cylinders with random limb motions, to test how many I can safely simulate at once.

Bringing all this background together, I then spent time brainstorming game ideas, and settled on a design which I think will strike the optimum balance between being realistically achievable in the given time frame and powerful as a tool to assess my aim. The broad idea is to create a game similar to a classic Asteroids clone, except using randomly constructed spaceships (made of thrusters and other parts) instead of actual asteroids. These spaceships will then be the NPC's which I evolve.

The next thing I'm going to work on will be the implementation of my game. I intend to work in Unity to build this game, as I have a lot of experience in Unity and it is broadly considered one of the best development environments for getting a game up and running quickly. Since I want to focus on the evolutionary aspect of the game, I want to get something to work with as soon as possible, and for a simple Asteroids style 2D game, I don't need high end graphics. So here I am choosing ease of use over graphical capability.

I have broken my implementation phase down into two primary sub-tasks. First I will need to develop a functioning enemy generator. Since I have already done something similar to this in 3D, I know it can be done in 2D. However, this task is more complex than simply generating enemies. I will need to design a knowledge representation for the AI (e.g. a multi-layer perceptron) and the physical body that AI will control. In order for these NPC's to be suitable for evolution, I will need to be able to represent their DNA intelligently as a binary string (This helps to make other phases of the evolution cycle easier, such as crossover and mutation.) So I will need a generator that can convert between an NPC object and a binary string for its DNA. Once I have this, I will need to place a graphical representation over the top of it so that I can observe my NPC's in the game world.

The next sub-task is to develop the actual evolutionary algorithm. I am undecided at this time as to whether I will implement my own algorithm or find a C# library for this, but either way, there are some parts to this that I will have to write my self. The evolutionary algorithm will need a fitness function that can determine how good or “fit” a particular NPC is. To do this I will need to implement the game play and decide upon a good way to assign a fitness to each NPC in a particular generation. This is not a simple task, for example, it can't simply be based on the NPC's that survive the longest or cause the most damage, because then the player can simply pick off the better NPC's before attacking the weaker ones to try and warp the results. I will need to first deduce the optimum fitness function and then tie it into the game play so that the process of natural selection takes place in the game.

I have given my self time at the end to finalize a presentable game. It is my experience from working in the games industry, and on my own games, that they nearly always take more time to build than you schedule for them. This segment of time should hopefully give me enough lea-way if I get stuck during earlier stages, and also give me time to implement some simple interface for the game such as a menu system and/or an in-built explanation of how the game works.

The last two sections are time to evaluate the degree to which I met my aim and reflect on what I have learnt, and time to write up my findings in my dissertation. It is worth noting that I am not planning on using any kind of surveys or feedback outside that of academic colleagues. This is because I do not wish for my project to be judged based on the quality of the game or how “immersive” it is, but rather the potential it demonstrates in accordance with my aim. I believe that the evaluation of my game therefore should be done from more a technical stand point than a subjective one.

References:

[1] *Gene Pool (Swimbots) homepage*
http://www.swimbots.com/

[2] *Box Car 2D homepage*
http://boxcar2d.com/

[3] *NERO homepage*
http://nerogame.org/

[4] *Simpson, R. (2012). "Evolutionary Artificial Intelligence in Video Games."*
https://wiki.umn.edu/pub/UmmCSciSeniorSeminar/Fall2012PapersAndTalks/ReedSimpson.pdf

[5] *Stanley, K. O., et al. (2005). "Evolving neural network agents in the NERO video game." Proceedings of the IEEE: 182-189.*
http://nn.cs.utexas.edu/downloads/papers/stanley.cig05.pdf

[6] *McDuffee, E. and A. Pantaleev "Team Blockhead Wars: Generating FPS Weapons in a Multiplayer Environment."*
http://www.fdg2013.org/program/workshops/papers/PCG2013/pcg2013_7.pdf

[7] *Ventrella, J. (2005). Genepool: Exploring the interaction between natural selection and sexual selection. Artificial Life Models in Software, Springer: 81-96.*
http://www.ventrella.com/alife/genepool.pdf

[8] *Sims, K. (1994). "Evolving 3D morphology and behavior by competition." Artificial life 1(4): 353-372.*
http://www.karlsims.com/papers/alife94.pdf