|  |  |
| --- | --- |
| Name | Peter Maloney |
| Student Number | 13002995 |
| Project Title | Real-Time Crowd Simulation Pipeline with a Focus on Implementation in Video Games |

Project Description

My aim for this project is to create a system and/or pipeline for real time crowd simulation, with a focus on human crowds for use within RTS strategy games and for both passive and user controlled crowds, but aiming to create a system versatile enough to fulfil a wider functionality (e.g. other animals and their behaviours, movement in 3 dimensions such as for birds or spacecraft for example)

The system will produce dynamic and realistic behaviours to add depth/life to games (peripheral crowds, city simulation) and/or to create interesting, dynamic gameplay (players having to predict / control behaviours in an emergency situation simulator, realistic enemy behaviour)

Deliverables:

* A unity 'tech demo' to showcase the functionality of my final system.
* A unity asset pack containing all components for an end user to utilise in their own projects.
* User documentation to help users fully utilise the software
* "Stretch goal" : Repackaging the functionality into a c++ library for "plug and play" usability with other systems.

Research and background

Initial research was focused on existing examples of crowd simulation. I was already familiar with the Boids Model1which creates interesting and usually realistic swarm behaviours from 3 basic 'rules', and when optimised can be fairly 'cheap' to run making it good for real time applications. While I aim to produce slightly different behaviours to those of the Boids, the method of building up individual agent movements from a number of individual movements based of each behavioural rule is a method I would at least like to try and use to implement my own system.

During my research I came across a tool called *Miarmy2*which creates some very realistic human behaviours, and has a very wide range of functionality, much of which outside of the scope of what I hope to produce, however it is a Maya plug in and has been created for use in animation, and so is less suitable for real-time applications. However some of the systems it utilises to create behaviours may be useful for use in my own project, such as finite state machines to move between behaviours.

While I have also found a number of other systems available, due to them being commercial, and often designed for specialist industries (emergency services in many cases) it is difficult to get much more information about the functionality and specifics of these systems, however it is safe to assume that while these systems produce reliable and realistic output (due to the fact they are used by emergency services etc. where these factors, may, in cases make the difference between life and death if used for training for example), they are also highly bespoke, and "black box" systems, and so do not allow much re-usability or adaptability for users.

Besides existing full systems, I also looked into the range of possible methods and "sub systems" that I could adapt and combine to create my system. A number of different solutions have already been produced for a number of different problems, one of these core problems is collision avoidance.  Simple collision avoidance systems are evident in the Boids model, where a force/acceleration is applied to each model 'away' from any other close agents, however there are a range of other methods. One piece of research3 looks into modelling collision avoidance by finding the nearest velocity closest to the desired velocity that would not lead to a collision with any nearby agents, after taking into account these other agents would also move to avoid. This system produces desirable effects, runs efficiently even for fairly large crowds, and can be combined with other systems. Another piece of research4 looks into making collision avoidance "smoother" and more realistic, by damping the speed at which agents move at leading up to collisions, moving faster if there is a big space to move into, slower as they approach a collision, creating 'waves' in crowds of people moving in the same direction. This method only effects 'following' behaviour, but showed me how a realistic behaviour is made up of lots of smaller parts, that are combined to produce the overall effect, like the Boids model, but that each of these individual behaviours can be broken up again. It also showed, as many other examples I looked at, that relatively simple equations can produce complex behaviours.

Finally, there are a number of other sources I have looked at, simply for a better understanding of the task I have chosen to complete, and to better guide my progress with the project. I am yet to fully evaluate these sources, but I have collected them for future reference. These include a paper5which, in its introduction discussed the levels of detail expected/required from the sort of system I plan to produce for different end uses, something for me to consider, especially when I am keeping the real-time capability of my system, and also started me planning how I can implement dynamic levels of detail (LoD) to achieve this. I have also began looking into the social force model, animal behaviour models as the behaviours produced by these are often already suitable for human simulation, or can be easily tweaked to be so, and this blog6 dedicated to crowd simulation for a point of reference for crowd simulation in general.

Objectives

By the end of this project I hope to:

* Achieve the creation of a robust, versatile crowd simulation software for own use and for a portfolio.
* Improve my programming and software development practices.
* Develop my problem solving abilities.
* Gain experience in producing professional documentation and other piece of support work for my projects.

In order to complete the project I will need to find out about:

* The methods of existing simulations.
* Equations and other representations that can be used to create realistic human behaviours, as well as what is considered human behaviour in order to compare my results.
* Options in terms of "streamlining" my system, in order to make it as suitable for real-time applications and use in a wider system as possible.
* The importance of and use of crowd simulation in games, such as what "players" respond to most in terms of realism, levels of detail, interaction etc.

I personally intend to learn / gain:

* About general system programming, to improve my programming ability, especially for programming for larger projects involving many systems working together.
* About the possibilities and opportunities in terms of adapting existing systems to be better suited to use in video games.
* What it takes to create realistic simulations and whether it is possible to reduce computational costs of these simulations without reducing this realism
* A greater understanding of the needs and limitations of systems like the one I aim to produce, as well as of systems for video games in general.
* A greater understanding of progress in the 'world' of simulation and create something that utilises or helps towards this.

Methods and Tools

To complete my project I will utilise skills gained from my course this far, my own understanding and research, and use available resources to gain new skills I may need.

So far on my course I have gained experience using the Unity Games engine, which I plan to use as an environment to complete a majority of my project. My experience thus far has shown me the capabilities and limitations of the engine, one of the main capabilities being it’s handling of 3d visuals and UI's, allowing me to focus on the 'under the hood' development of the system. Through my course I have also gained experience of both c++ and c# in different environments and this has given me a good understanding of coding practices and this should assist me in producing versatile, clean and functional code. The experience of c++ is also the main reason I have set my stretch goal to produce a c++ library from my output from unity, as I feel I have enough understanding to effectively port the code.

|  |  |  |
| --- | --- | --- |
| **Risk** | **Mitigation** | **Contingency** |
| Unity's limitations may hinder progress. |  | Begin porting toc++earlier, and work within the DirectX framework to give more flexibility |
| Aspects of my project may require knowledge outside of my current scope. | Focused research throughout the project should provide me with specific knowledge, or at least relevant knowledge that allows me to continue work without a complete understanding of a subject. |  |

Risks and Issues

Required Specialist Resources

Input from those with a knowledge of crowd behaviours would be useful but not vital, to give feedback on the output of my project as it progresses to help me produce more realistic results.

If progress goes well, and the core system is finished with time to spare, it would be nice to begin adding animations / models to create a more polished system and better test LoD, which it would be nice to have someone with specific skills produce.

Monthly Project Plan

|  |  |  |
| --- | --- | --- |
| October | Proposal submitted by 15/10/2015  Implement basic agent behaviour  Begin implementing agent Interaction | 8 days  7 days  7 days |
| November | Polish agent interaction and test behaviours with varying group sizes  Begin work on pathfinding  Combine pathfinding with existing behaviour  Polish progress so far | 3 days  7 days  7 days  7 days |
| December | Work on research report and extra time to complete previous objectives that may be unfinished  Research Report hand in 10/12/15  Begin work on agent individuality (adding variables)  Utilise agent variables for behaviours | 10 days  7 days  10 days |
| January | Begin work on UI and polished debug system  Catch up for previous objectives, and prepare some scenarios for the demo | 7 days  Rest of Jan |
| February | Prototype demo in class 03/02/16 or – 05/02/16  Begin work on LoD systems  Work on feedback from demo | 7 days  7 – 14 days |
| March | If at a suitable stage, begin work on porting to c++ library | Unsure |
| April | Hand-in 14/04/2016 |  |

1 Reynolds, C  Boids Background and Update <http://www.red3d.com/cwr/boids/>                                        [Last updated 6 Sept 2001]

2 More information on Miarmy at <http://www.basefount.com/miarmy.html>

3Guy, S et al [2009] ClearPath: Highly Parallel Collision Avoidance for Multi-Agent Simulation  available at http://pcl.intel-research.net/publications/clearpath\_sca2009.pdf

4Lemercier, C et al Realistic following behaviors for crowd simulation  available                                          at https://spiral.imperial.ac.uk/bitstream/10044/1/21595/2/EG2012.pdf

5 Badler, Net al  Real Time Virtual  Humans(Abstract) Available                                                                 from <http://www.cis.upenn.edu/~badler/bcs/Paper.htm>

6http://crowdsimulation.blogspot.co.uk/