Walking Penguins Crowd Simulation

Di Xiao

Master of Science,

Computer Animation and Visual Effects

NCCA 2013-2014

Bournemouth University, Talbot Campus

August, 2014

Content

Abstract 1
Acknowledgements 2
Chapter 1. Introduction 3
Chapter 2. Related Work 6
2.1 Craig Reynold's model 6
2.2 Massive 7
2.3 MPC Artificial Life Crowd Engine 9
Chapter 3. Technical background 11
3.1 Physics 11
3.2 Reynold's Flocking Algorithm 12
3.2.1 Separation 13
3.2.2 Alignment 13
3.2.3 Cohesion 14
3.2.4 Steering 14
3.3 Terrain Following 15
Chapter 4. Design and Implementation 17
4.1 Neighbor List 17
4.2 Seeking Food 18
4.3 Home 19
4.4 Finding Couple 20
4.5 Stick to Terrain 20
Chapter 5. Results 22
Chapter 6. Conclusions and Future Work 26
6.1 Summary 26
6.2 Future Work 27
Bibliography 28

Abstract

The activity of social animal is a very common phenomenon in nature. In the computer animation industry, the simulated animals not only have their individual behaviour, but also are restricted by the entire crowd. In many animation films and games, the technique of simulating the crowd animals has often been applied as well, such as Happy Feet (2006). The aim of this project is to simulate the crowd of penguins, and obtain independent animal's behaviour which also can influence each other, making the behaviours movement of a walking penguin closer to reality under the three-dimensional environment. And the method will be built on the Reynold's Flocking Algorithm. This program is based on C + + and OpenGL, and uses Qt Creator and NGL library to compile.

Acknowledgements

Completing my master project is a great challenge for me. Therefore, I would to take this chance to express thankfulness to my colleagues who helped me. Without those people it will be very tough for me to complete my project. At the beginning, I would like to thank Jonathan Macey and Mathieu Sanchez. They always answered my questions very patiently and taught me how to take the right direction when I had the problems and mistakes in my code. At the same time, they also helped me successfully to improve the quality of my project.

Secondly, I would like to thank my boyfriend and classmates to answer my questions, and gave me courage and confidence to finish my project. Additionally, thank everyone who gave me valuable advices because without them I probably could not successfully completed my project.

Finally, I would like to thank for my family willing to cultivate me to study my Master Degree. They always spent much time to concern about my situation. When I felt upset, they also listened to me and gave me the greatest encouragement to complete the target of my life.

Chapter 1

Introduction

Crowd behaviour is common in humans, animals, birds and insects (Figure 1). However, each group displays different behaviours. In addition, in specific environments, each individual member can behave as a separate entity, whilst their movements and behaviours continue to interact with each other (Reynolds1987). According to Reynolds's revolutionary idea (Thalmann and Musse 2007), members of the 'crowd' have a complex behaviour within the group, and will have different responsibilities or specific behaviours.

Figure 1: Birds

Recently, crowd simulation has become a very common technique in live-action films, animations and video games, as exampled in the cases of Troy (2004), Epic (2013), Mario & Sonic at the London 2012 Olympic Games (2011). In order to obtain the expected visual effect, the simulation should manage the behaviours or actions of the agents to achieve the crowd. In the film Troy, some of the scenes were shot in the real world, but many final shots, which appeared real were created using computer visual effects.

McClean (2007) states that visual effects such as crowd simulation should be invisible and not be perceptible to the audience and that to achieve this all shots must be based on the real physical world. In live-action films, crowd simulation can not only created the intended visual impact, such as a war scene, but can also fully grasp the attention of the audience. However, most of these scenes involve a great number of groups, containing at times innumerate numbers, which makes the process of creating the figures very complicated and time consuming.

Creating simulations are powerful tools to tell and enrich a story. For instance, it could be challenging to engender a sense of the size Agamemnon's army marching outside the city of Troy, or breathtaking apocalyptic scenes of millions of zombies attacking humans in films such as *World War Z.*

The Emperor penguin is one of the most social species in the world. Due to the extreme environment of the Antarctic, these penguins need to live together interdependently. In March of the Penguins (2005), the producers faithfully documented the details of the annual journey of Emperor penguins focusing on activities such as (Figure 2), such as finding food and taking care of their chicks.

In crowd simulation, many animation companies have created their own crowd simulation solutions. This is possible as independent companies often have their own program teams to develop the software separately, and use the developed technique as custom effects. For example, DNEG, MPC and Industrial Light and Magic, all use bespoke crowd simulation software, which offers them greater flexibility and convincing effects. Another reason for creating bespoke software is that the cost of buying a license can be prohibitive for smaller companies and individuals (Bielik

2004).

Figure 2. March of the Penguins (2005)

Figure 3. *Happy Feet*

In the animation film, Happy Feet (2006), there are many shots simulating penguin crowds. (Figure 3) those penguins live together and interact with each other. In this project, the main idea was to use the crowd simulation techniques to emulate penguin behaviours and create an animated crowd using the C++ program.

Chapter 2

Related Work

During the late 20th century, some researches about crowd simulation emerged and became increasingly of interest to animators. Moreover, researchers who were working on the crowd simulation came from different academic fields, such as computer graphics and physics. For researchers, there were limitations in terms of exchanging information and ideas, as according to Thalmann et al. (2004). They are not typically familiar with areas of activity other than their own.

Currently crowd simulation is commonly used in the entertainment industry. With the extensive development of computer games and films incorporating digital effects and it has become one of the essential techniques for animators. However, in this area, the artists place emphasis on visual realism and aesthetics, rather than the validity of computation and calculation.

2.1 Craig Reynold's model

The main idea of this project is based on Craig Reynold's (1987) flocking system. In this system, there is a group 'flock' or 'herd', and the individuals are called "Boids". The motion of each boid based on simple steering behavioural rules, including separation, alignment and cohesion. Their movements are based on simple decisions

to move and interact with their neighbours independently.

In this classic study, the researchers not only present a flocking algorithm, but also talk about creating a turning point, an individual virtual force, and affecting the movement of each agent. In this project employs an adapted version of Reynold's (1987) model.

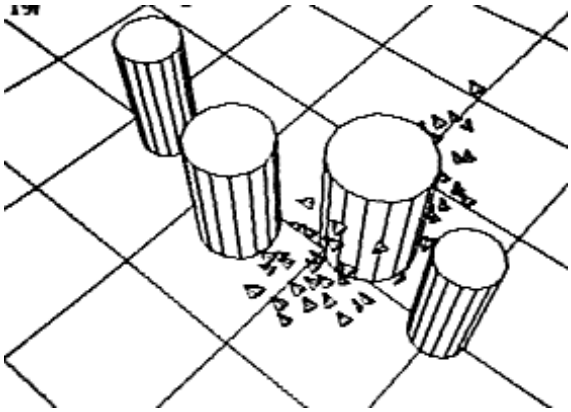

Figure 4. Flocking system (Reynolds, 1987)

2.2 Massive

In the visual effects industry, MASSIVE (Multiple Agent Simulation System in Virtual Environment) is the most popular available software for creating crowd simulation. It was first developed by Stephen Regelous. In Peter Jackson's films, the Lord of the Rings (2001; 2002; 2003) trilogy, the audience was attracted by the battle senses which were generated using Massive.

Figure 5. The battle shot of Lord of the Rings: Return of the King

Massive has contributed significantly to the production of visual effects. It is not only used in the real-life films, but also in many animation films, for example, The Mummy: Tomb of the Dragon Emperor (2008) and Happy Feet (2006) and Happy Feet Ⅱ(2011). In order to achieve the movement of the agents, Massive sets a 'brain' applying fuzzy logic rules, which controls the behaviours for each agent depending on the environment, and interactions with other agents, avoiding obstacles, tracking the goal or following a terrain.

Figure 6. ***The Mummy: Tomb of the Dragon Emperor*** **(2008)**

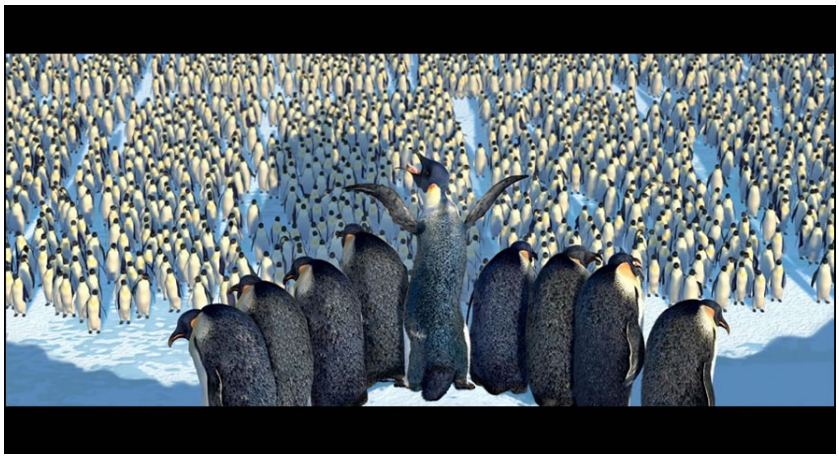

Figure 7. ***Happy Feet*** **Ⅰ(2006) or Ⅱ(2011)**

2.3 MPC Artificial Life Crowd Engine

Artificial Life Crowd Engine (ALICE) is another crowd simulation technology in MPC. In many films, ALICE has been used for creating complex crowd simulation scenes, and it was developed for the film Troy (2004).

Figure 8. The before and after effect in Tory (2004)

ALICE was originally a part of Maya. (Figure 9 and Figure 10) However, it has moved away from its dependence on Maya. Nevertheless, there is a MEL bridge for connecting with Maya. The artists can easily use the interface of ALICE to control the complex behavioural state in a node graph and build and manage the crowd graphs (Pieke 2008).

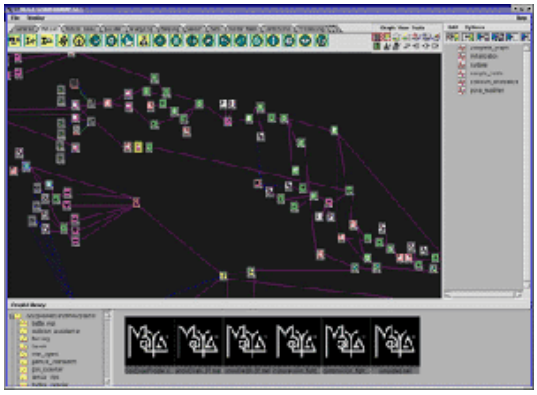

Figure 9. Crowd Simulation in Maya,(Kolve 2004)

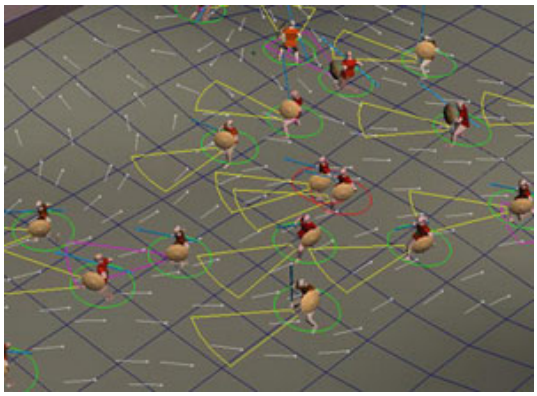

Figure 10. Crowd Simulation in Maya,(Kolve 2004)

Chapter 3

Technical background

At this point it is useful to discuss the background for researching the engineering and computer animation industry. Therefore, some physical and mathematical concepts are introduced in this chapter. This chapter includes a brief outline of all the ideas applied in crowd simulation directly or indirectly in order to better understand the technical methods for following approach.

3.1 Physics

In the real world, every motion of objects is based on physical rules. Newton's second law states the relationship between an object's mass (m), its acceleration (a) and the force F (the Physics Classroom).

$$F = m \times a \tag{1}$$

- Force. This physical quantity also is a vector, and has the own magnitude and direction. It is the interaction between objects and can alter an object's speed or line of motion. In this project, force is the main variable for an agent's movement.

- Position. The position coordinates of the specific point is determined based on the origin. Depend on the change of the position per unit time, the magnitude and the direction, the movement can be calculated during the process of moving.

$$p_{i+1} = p_i + v \times \Delta t \quad (2)$$

where,

p_{i+1} and p_i mean the position at the time instant t_{i+1} and t_i;

v means the velocity;

Δt means the time interval between the time instant t_{i+1} and t_i.

- Velocity. Velocity is a vector, and also has its own magnitude and direction. In physics, velocity, a physical quantity, defines the rate of change of distance over time, and represents the speed of the object which relatives to the movement of a reference object. Mathematical equation can be expressed as follows:

$$v = \frac{\Delta x}{\Delta t} \quad (3)$$

$$v_{i+1} = v_i + a * \Delta t \quad (4)$$

where,

v_{i+1} and v_i mean the velocity at the time instant t_{i+1} and t_i;

a means the acceleration;

Δt means the time interval between the time instant t_{i+1} and t_i.

- Acceleration. It is a vector which defines the rate of change in velocity with respect to time. And the direction of the acceleration is the direction of the net force. Mathematical equation of acceleration can be expressed as follows:

$$a = \frac{F}{m} \quad (5)$$

$$a = \frac{\Delta v}{\Delta t} \quad (6)$$

3.2 Reynold's Flocking Algorithm

Reynolds (1987) introduced a distributed model for schools, herds and flocks and simulated behaviours and movements by applying three simple rules to each agent. He called the agents "Boid". The three steering rules are cohesion, alignment and separation, which can modify the position and velocity of the boids to make them move as a unit at the same time, and also limit their velocity for avoiding collision with each other. Each boid's position and velocity can be based on their position and

the velocity of its neighbouring boids.

There are there main rules/behaviors for each flocking boid:

- Separation (avoid collision with the neighbours)
- Alignment (turn to the average heading of the neighbours)
- Cohesion (go to the average position of the neighbours)

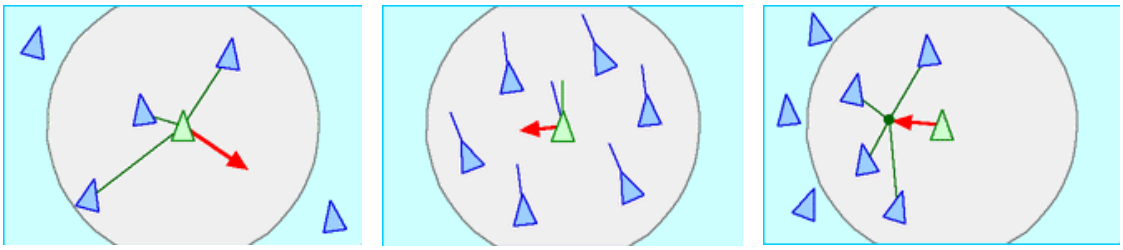

Figure 10. Alignment, Cohesion & Separation (Reynolds, 1987)

The flock members need to have a direction and keep a certain distance (safe distance) from their neighbours. These rules/behaviours describe the dynamic behaviour of interaction between each boid and their neighbours.

3.2.1 Separation

This force makes the boid to move away from its neighbors and remain to keep a certain distance.

$$ForceOfSeparation = \sum_{i=1}^{n} \frac{p - p_i}{d_i} \tag{7}$$

where,

p is the position of the current boid;

p_i is the position of the neighbor I;

d_i is the distance between p and p_i.

3.2.2 Alignment

This force makes that every boid aligns with their neighbors.

$$averageSteering = \frac{\sum_{i=1}^{n} v_i}{n} \tag{8}$$

$$ForceOfAlignment = averageSteering - v \tag{9}$$

where,

v_i is the velocity of the neighbors from i to n;

v is the velocity of the current boid.

3.2.3 Cohesion

The force of cohesion makes a boid cohere with its neighbour. From its neighbour list, if there are neighbours and the distance is larger than safe distance, they will have a positive vector to move towards the centre of the mass of the neighbours, and also preserve the safe distance.

$$centerOfNeighbourhood = \frac{\sum_{i=1}^{n} p_i}{n} \tag{10}$$

$$ForceOfCohesion = normalize(centerOfNeighbood - currentPosition) \tag{11}$$

where,

i is the number of neighbors from 1 to n.

3.2.4 Steering

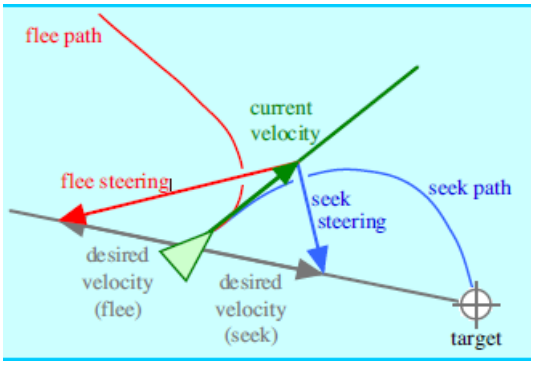

Figure 11. Seek (Reynolds 1999)

Reynolds (1999) noticed that “Seek (or pursuit of a static target) acts to steer the character towards a specified position in global space.” There are principally two ways to implement the sought behaviours. The first is to apply multi-points, in the

sought behaviour so that the agents only need to search the point when they are within the spatial area defined by the distance from the specified point. The second is to find a single point at any fixed position that is irrespective of the agent, this is very important when applying a constant steering behaviour in a specific condition. Both approaches will be implemented in this project.

3.3 Terrain Following

Height of a point in the triangle

For a grid, the geometry is created by several triangle faces, each face has three vectors which includes three values (x, y, z). After confirming which triangle the point is within, for example, in the triangle V0V2V1, it is possible to calculate the height of the point (Figure 12).

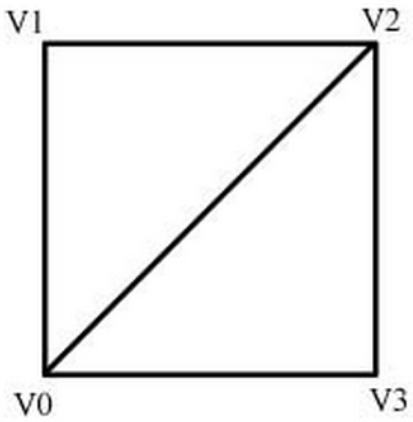

Figure 12. Sketch plot of the vertexes and triangles

Noticing that the point P should on the plane determined by the three vertexes of the triangle, we can obtain the relation as follows:

$$\overrightarrow{V_0P} \cdot \overrightarrow{N} = 0 \tag{12}$$

where,

$\overrightarrow{V_0P}$ is the vector determined by the point P and the point V_0;

$\overrightarrow{N}$ is the normal vector of the plane $V_0V_2V_1$, which can be calculated by:

$$\overrightarrow{N} = \left[N_x, N_y, N_z \right]^T = \overrightarrow{V_0 V_1} \times \overrightarrow{V_0 V_2} \tag{13}$$

Then we can obtain the expression of y of the point P:

$$y_P = y_{V_0} - \frac{N_x \left(x_P - x_{V_0} \right) + N_z \left(z_P - z_{V_0} \right)}{N_y} \tag{14}$$

Chapter 4

Design and Implementation

This chapter will discuss the design and implementation of the program. In nature, a Multi-Agent System would have some complex and unexpected behaviours, which is very interesting. In Antarctica, the adult penguins' behaviours are related to their chicks, which they repeat every year.

The project is based on real life of penguins, and simulates foraging female penguins and their return to their partner and chicks throughout the process. The crowd system would be based on practical algorithms for flocking system behaviours. Due to gender differences and the harsh environment, the penguin groups will have diverse behaviours. However, individual penguins will also have different behaviour. As proposed by McClean (2007), the simulation was based on reality. Meanwhile, it should be noted that this simulation can be adapted for other scenes, such as migrating animals.

4.1 Neighbour List

In the group every member will have their own neighbour list, depending on the position of the agent. The neighbour list will be stored as a vector list for each agent, which is set up in the Object class. As is shown in the Figure 13, during the process,

an agent will get its neighbours by a visibility radius, any agents outside of the radius will not affect this agent.

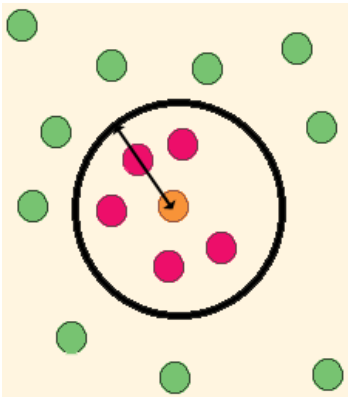

Figure 13. Neighbour List

At each frame, the neighbour list will be updated - based on the new position of the agent by the moving functions. Each rule of the flocking, such as cohesion, separation and alignment will provide a vector for each agent. Under those vectors, a new vector will be created. The direction of the agent will be based on this vector which will be normalized, the next position will be calculated by the direction.

Within the crowd, the agents may be from different groups and have different functions to drive their movements. So there are many radius values for each function which can create different movements.

4.2 Seeking Food

During the process of penguin's growth, one of the parent penguins will take care their child and the other will leave and collect food. Once this is achieved they return and the roles are switched. This phenomenon is the main idea of this function.

In this function, the agent is assigned a new vector. However, due to the position of each penguin, the direction of vectors is different. For this behaviour, the food is located in several f positions, each has a circle range. If an agent moves into the range of one food, it will be assigned a new force to the position, and move toward it. (Figure 14) at the same time, its neighbours will get a cohesion vector from the agent, and drive them to move into the range.

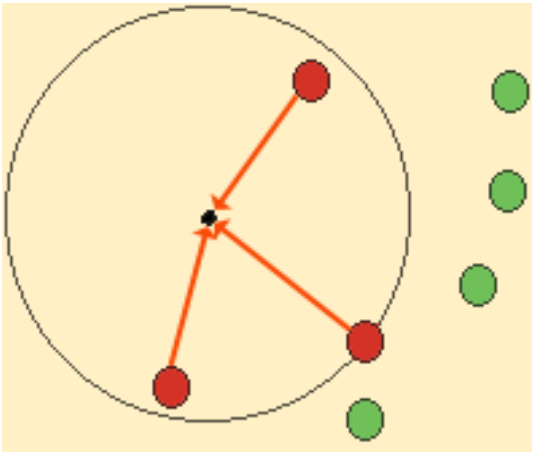

Figure 14. Finding food

4.3 Home

In this scenario, two groups of agents will find new habitat positions. Whilst moving, each chick agent will be assigned a new force to its own parent which can help them to locate and face to their parent. Apart from the basic forces, the adult agents will get a new force from the 'home' position, while the chicks will follow their parent's movement. The purpose of this function is to create a constant flow of movement amongst the crowd of penguins. For those two groups, the movement is dynamic and also allows 'start and stop' movements (Figure 15).

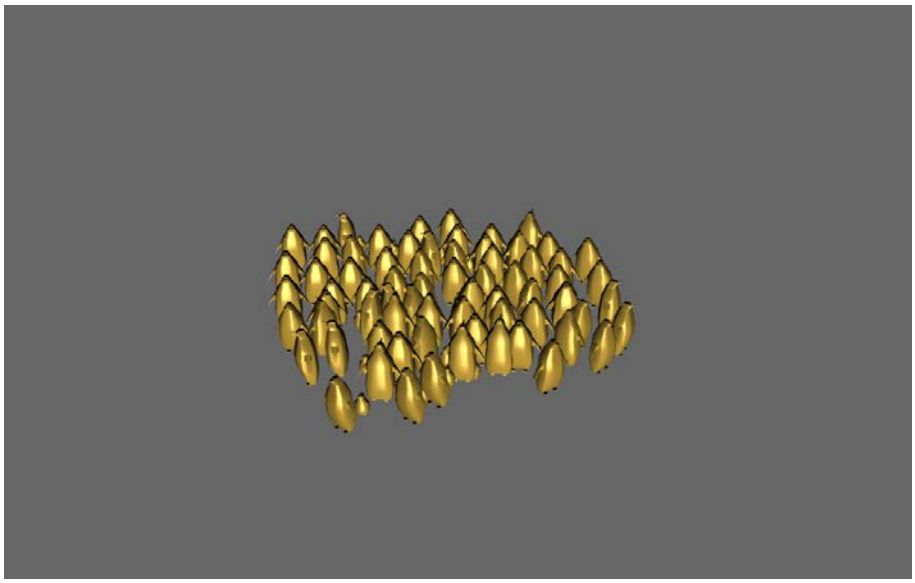

Figure 15. Finding Home

4.4 Finding Couple

In the real Antarctic environment penguins will return to their family after feeding. Because each penguin is an independent individual, the timing of their return will vary. Each penguin only has a single companion, so, creating an agent when foraging for food requires a new force, does finding their partner, while they also have some basic behaviours when they move.

As shown in Figure 16, penguins also behave as herds, for example, when they are returning from a feeding location to their partner they will move as a group. However, as they walk closer to their own family, they will separate, and find their family individually. After that, the agent will move as a family.

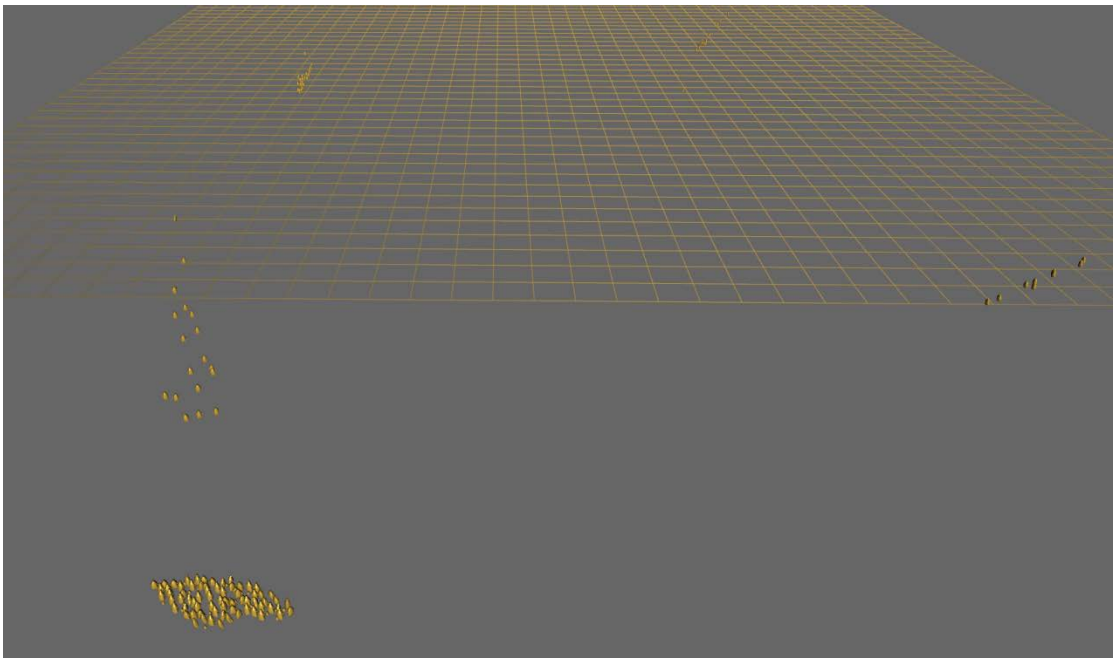

Figure 16. After foraging and finding their family

4.5 Stick to Terrain

In this project, all the agents will walk on a 3D terrain which is a triangulated mesh imported from Maya (Figure 17). There are many methods that can be used to locate the agents' position on the terrain. In this project, the agent will depend on the present position to find the nearest face in the triangle given, based on its three vertices. Each

triangle face will store the number of three vertexes. In order to make them follow the terrain, each agent can locate the three vertexes from the vertexes list for each face of the mesh, and then calculate the minimum distances of the face of a triangle's centre of mass, to find the triangle face in which the agent is present. For the 3D mesh, each X-Z coordinate has one Y value. To acquire the Y position, the agent also can use the three vertexes to obtain the height value for each time step. As discussed in the previous section.

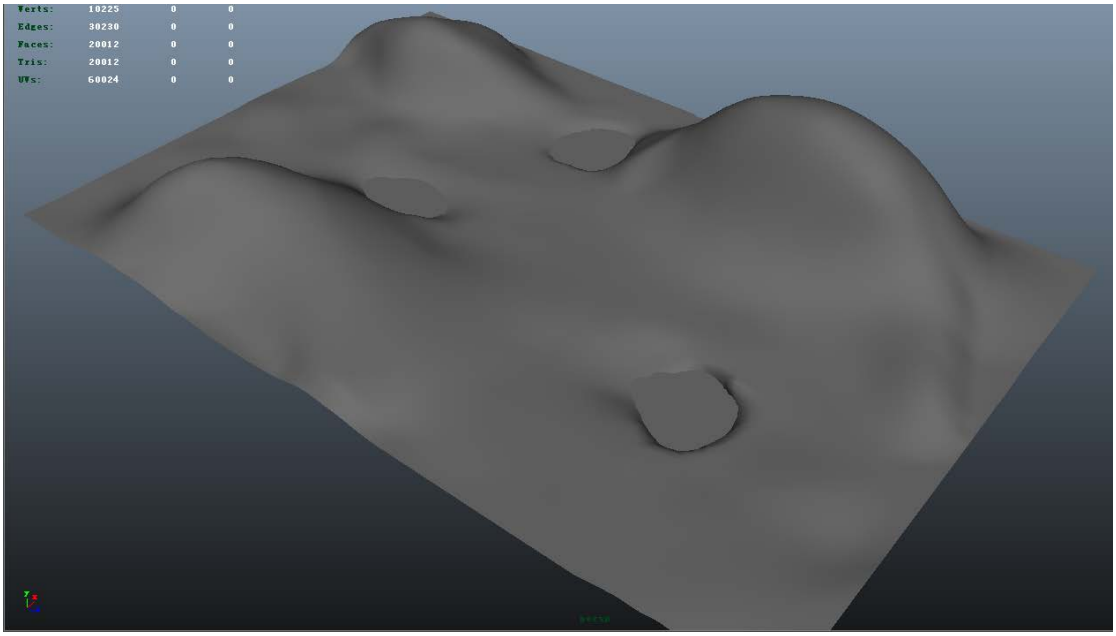

Figure 17. Terrain Mesh in Maya

Chapter 5

Results

For this project, every agent can walk and move properly on the terrain. At the beginning, one group of agents gets a random vector as their own velocity, and move by following this vector to find the “Food”. After they “saw” the water, they will get a new velocity to drive them move toward the water. During finding the “Food”, they can move as a group, but they also have the individual behaviours. After the foraging, the agents can not only find the “Home” positon, but also they can find their family accurately.

Another two groups of the agents, they will move around the location of the “Home”. The group of the baby, they are always getting a vector to point their parents, and follow their parents to move.

Also, giving the simple texture to each agent what can help to recognize the direction of the facing for agents and the rotation correctly. During the whole process, their next movement will depend on their neiborhours’ position and their own position to get, including the position and velocity.

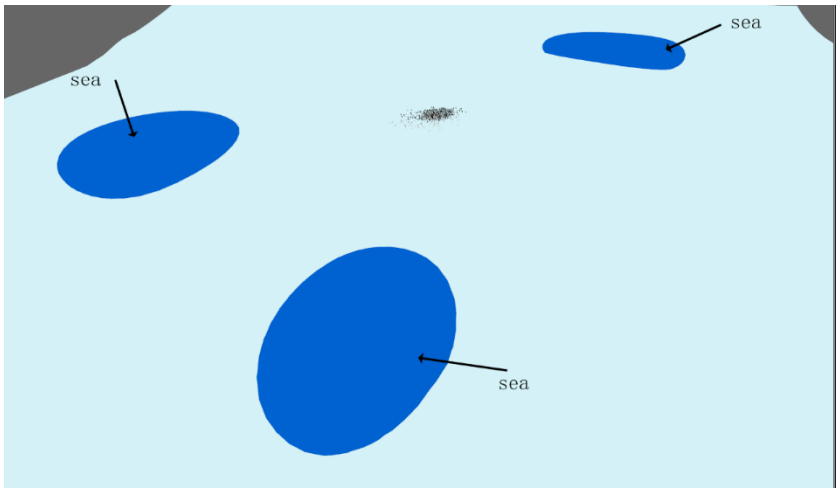

Figure 18. The close shot: The start shot on the default setting (1200 agents)

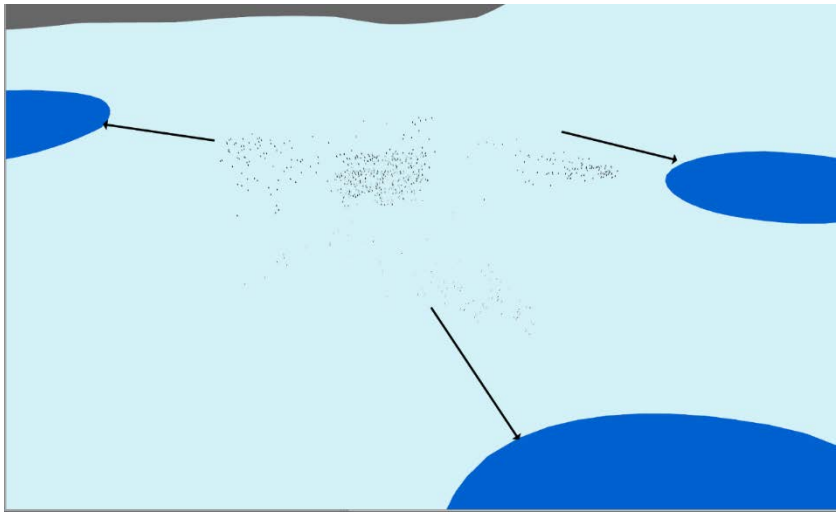

Figure 19. The close shot: A group of adult agents are going to forage themselves (1200 agents)

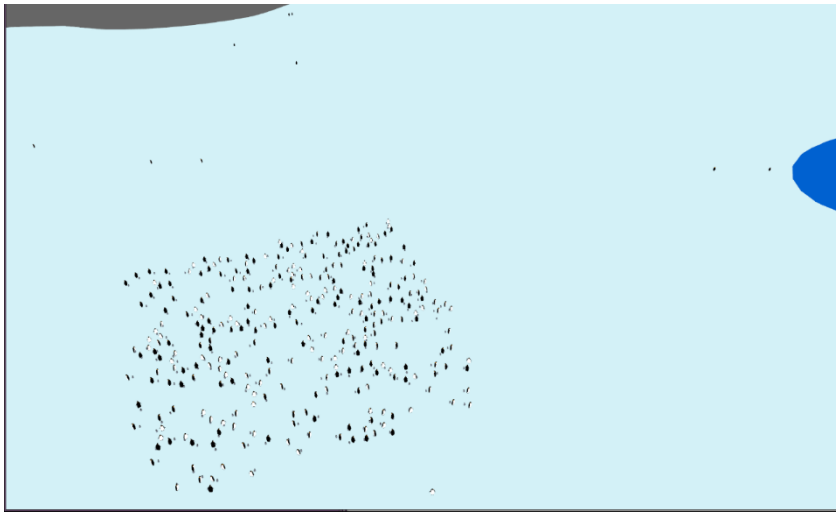

Figure 20. The shot of "Home": A group of adult agents are walking around the "Home" area, and the baby penguins are following their parent. (1200 agents)

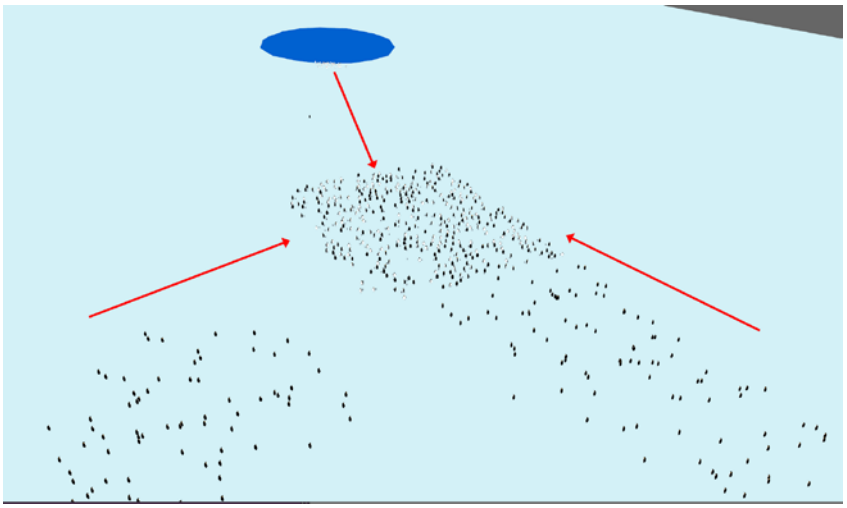

Figure 21. The penguins go back to their "Home" (1200 agents)

From the beginning of the simulation the agents will look for the 'food'. Once they find the food they return back to their home and find their partner. The entire process will take about 3050 frames. In Figure 19, the agents start to find the food position. After they found some 'food', they will not hesitate to directly move to the location of the 'food'. As the position of each agent is different, they will follow different paths and directions. However, each penguin will only return once they have found food.

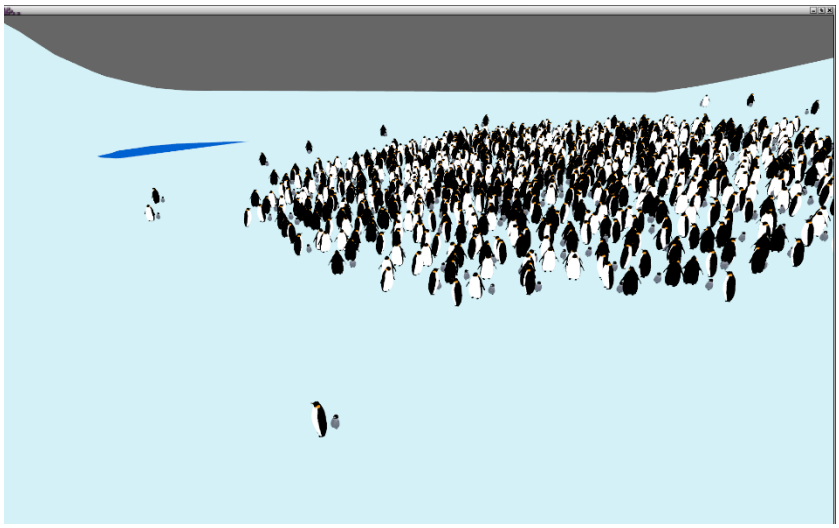

Figure 22. The small number of penguins stand outside

As shown in Figure 20, there are minority members of the searching food group who will not find the specific location of the 'food'. They will continue looking for food, and so not return to their home. The partner and baby will wait for the other penguin to return (Figure 22). Normally, most of the agents will return to the 'home', like

Figure 21. Then, they can move together.

Chapter 6

Conclusions and Future Work

The aim of this project is to simulate the crowd of penguins, and obtain independent animal's behaviour which also can influence each other, making the behaviours movement of a walking penguin closer to reality under the three-dimensional environment. And the method will be built on the Reynold's Flocking Algorithm. And the code part is almost finished the design. However, the simulation should be export the data and import into the visual software for rendering.

6.1 Summary

The project highlighted that in order to achieve a more realistic simulation, the behaviour of a group is depended on every agent actions and decisions, and whether the result of the simulation is realistic or not, depends on how realistically each agent behaves.

This simulation program is based on and aims to represent group behaviours, and also includes the different individual behaviours of each agent using C++ program. The simulated movement of behaviours is only related to the individual penguin or interaction with neighbours. There is also has a relationship with the environment. The completed simulation was implemented in the compiled language C++, also using

the NGL library and some QT functions.

6.2 Future Work

Although the program can basically realize the crowd simulation of penguins, there still exist many parts that can be optimized for the whole project.

Rendering. Recently, the generation of an animation can be realized by QT based on OpenGL, by which the whole visual effect is not good enough such as the setting of the lights and the polygon's texture. In current program, the setting of the lights is under default option, which is not good for demonstrating the reflection and shadow. This can be modified by exporting the motion data of the agent, importing and then render it in some visual effect commercial software, such as Maya and Houdini, so that it can be obtain a better animation with higher quality.

Rigging. In the real world, the skeleton motion should exist in the motion of the penguins. Thus, in order to obtain a more realistic animation scene, the project can be added more detailed rigging data to its behaviours. For example, the adult penguins feed the young penguins or glide on the snow. Blending them altogether, it will generate smooth and realistic individual motions.

Collision Detection. In the future work, a data structure can also be added to divide the space by using the regular grids or Octree, which can accelerate the speed of the program to realize the crowd simulation in a more efficient way.

Bibliography

British Antarctic Survey, 2013. *Penguins and other birds* [online]. NERC Science Of The Environment, Available from: http://www.antarctica.ac.uk/about_antarctica/teacher_resources/information/primary/pschoolsq_penguins.php [Accessed on 06.07.2014].

Bielik, A., 2004, *'Troy': Innovative Effects on an Epic Scale* [online], Animation World Network, Available from: http://www.awn.com/vfxworld/troy-innovative-effects-epic-scale [Accessed on 06.07.2014]

Happy Feet Ⅰ (2006) [film, DVD]. Directed by George Miller. USA: Kennedy Miller Productions, Animal Logic Films

Happy Feet Ⅱ (2011) [film, DVD]. Directed by George Miller. USA: Village Roadshow Pictures, Kennedy Miller Mitchell, Dr. D Studios

Kolve, C., 2004. *VFX WORK*. Available from: http://www.kolve.com/vfxwork/vfxwork.htm [Accessed 12.07.2014].

March of the Penguins, 2005. [film, DVD]. Directed by Luc Jacquet. USA: Canal+, Warner Bros. Entertainment, Wild Bunch, National Geographic Society

Massive Software. *Driven character animation for Film.* [online] Available from: http://www.massivesoftware.com/film.html [Accessed on 16.08.2014]

Mcclean, S., 2008. *Digital storytelling: the narrative power of visual effects in film.* [Online] Cambridge, Mass.: MIT Press.

Pieke, R., 2008. *The Digital Eye: MPC's R&D Confronts a Changing Industry* [online]. Animation world Network. Available from:

http://www.awn.com/vfxworld/digital-eye-mpcs-rd-confronts-changing-industry [Accessed on 06.08.2014]

Reynolds, C., 1987, *'Flocks, Herds, and Schools: A Distributed Behavioral Model'*, in Computer Graphics, 21(4) (SIGGRAPH '87 Conference Proceedings), 25-34.

Reynolds, C., 1999, *'Steering Behaviours For Autonomous Characters', in Miller Freeman Game Group* [online], (Game Developers Conference proceedings), 763-782. Available from: http://www.red3d.com/cwr/steer/, [Accessed on 06.06.2014].

Seymour, M., 2008, *Massive Mummy Clay Armies* [online]. fxguide. Available from: http://www.fxguide.com/featured/massive_mummy_clay_armies/ [Accessed on 06.08.2014].

Tory, 2004. [film, DVD]. Directed by Wolfgang Petersen. USA: Helena Productions

The Mummy: Tomb of the Dragon Emperor, 2008. [film, DVD]. Directed by Rob Cohen. USA: Relativity Media, The Sommers Company, Alphaville Films

Thalmann, D. and Musse, s., 2007, *Crowd Simulation.* London: Springer

Thalmann, D., Hery, C., Lippman, S., Ono, H., Regelous, S., and Sutton, D.,2004. *`Crowd and group animation'*. In ACM SIGGRAPH 2004 Course Notes, SIGGRAPH '04, New York, NY, USA. ACM.