Modelling and Simulation of Swarm Behaviour using Particle Systems

Jonathan A. Z. Thomson, Sep. 2011

Department of Software and Computer Engineering
University of Canterbury, New Zealand
jat126@uclive.ac.nz; jonemode@gmail.com

Abstract

A Swarm Intelligence (SI) model was developed to simulate pedestrians under various conditions. This was accomplished using a Cellular Automaton (CA) approach with vector field based crowd guidance techniques. Individuals of a crowd have many behavioural parameters, which when altered can change the outcome of a given situation. A boarding and evacuation procedure of a Boeing 777-300 was simulated, as well as an evacuation of a building. Behavioural parameters such as the probabilities associated with a give-way situation were changed so that the resulting affect on the swarm behaviour could be analysed, and predictions about real scenarios could be made.

1. Introduction

Modelling swarm behaviour requires the identification of behavioural attributes which characterise the thing or life form being modelled. To accurately simulate a scenario, the most prominent behaviours need to be recognised first so that a realistic interpretation can be made about how the thing or life form would move in the real world. In this paper, a Swarm Intelligence (SI) model is designed and implemented using a Cellular Automaton (CA) approach to model people in urban environments. Several scenarios are tested with the model, not only to identify bottlenecking areas and durations times, but also to validate the model against real simulations.

1.1 Swarm Intelligence

Swarm Intelligence (SI) as a concept models a system of self-organized bodies and their combined behavioural attitudes. Interacting artificial or natural bodies at a micro level cause coherent functional macro level (global) patterns to emerge, and it is possible to investigate collective (or distributed) problems without centralised control or the provision of a global model using the basis SI provides.

Application Areas
The application area of this project is *crowd simulation*, or more specifically, *human simulation*. Swarm technology allows the simulation of other life forms as well, such as birds or fish. The flocking behaviour of birds was simulated by *Craig Reynolds* [1] using his *Boids* artificial life program. Boids is an example of emergent behaviour, as complexity arises due to the interaction of individual bodies obeying to a set of rules. The program developed in this project follows the same principle, except a different set of rules will be used, and only in two dimensions.

Packet routing is a common research area in SI. Known as *Ant-based routing*, a probabilistic routing table rewarding/reinforcing the route successfully navigated by each "ant" which flood the network is used. Airlines have used ant-based routing for simulating the allocation of aircraft to airport gates.

1.2 Cellular Automata

A Cellular Automaton (CA) is a discrete model using a collection of cells in grid architecture, whereas each cell has a finite number of states *k* it can be in, in a finite number of dimensions, according to a set of rules [2]. The number of states *k* is typically an integer with $k = 2$ (binary). Colours are usually assigned to each state so that there is a distinction between each.

Depending on the application, the evolution of a specified shape or specified initial conditions through a number of discrete time steps can be measured and analysed. Cellular automata come in a variety of shapes and forms, as there are many kinds of grid architecture, and many possible rule sets. The most typical grid architecture (used for this project) consists of two orthogonal sets lines forming a two dimensional grid of cells. Other two dimensional possibilities include a formation of hexagonal cells, or even triangular cells. A grid of squares provides eight neighbouring cells per cell (can depend on the rules defined), where a grid of hexagons provides six and a grid of triangles provides twelve neighbouring cells. Cellular automata may be constructed on Cartesian grids in an arbitrary number of dimensions, with the *d*-dimensional lattice $\mathbb{Z}^d$ being the most common choice. The choice of architecture will certainly be dependent on the application.

The immediate neighbourhood of each cell is an important design decision, but like the architecture it is also dependent on the application. The simplest neighbourhood choice consists of the "nearest neighbours", where the cells one discrete step away are chosen as the neighbourhood of that cell. Neighbourhoods like this are known as *Moore* neighbourhoods (a square neighbourhood), or the *von Neumann* neighbourhood (a diamond neighbourhood). These neighbourhoods are typical choices for a two dimensional grid of squares, as shown in Fig. 1 below.

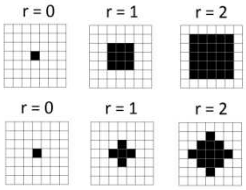


Figure 1. Moore neighbourhood (top) and von Neumann neighbourhood (bottom)

S. Wolfram studied the amazing properties of binary, nearest-neighbour, one-dimensional automaton, known as "elementary cellular automata". There are 256 of these automata; whereas each one is named after the decimal representation of the rules that govern it (refer to Fig. 2 in Appendix E for a demonstration of "Rule 90"). There are many other applications of cellular automata in areas of mathematics, physics and theoretical biology.

1.3 Micro/Macro Level Architecture

To implement the design requirements for the project, two models were needed; a small and large scale model of the swarm, those being the *Micro* and *Macro* level architectures respectively. As discussed in 1.6, a give-way convention must be followed for both models so that they are unified with one another. The give-way convention used is based on a "stay left, give way to right" principle.

Micro Model

SI systems and their underlying "intelligence" are highly dependent on the local behaviour of the individual bodies. This local behaviour is governed by a *Micro* model which defines the characteristics each individual body exhibits and the activities each body can perform at the smallest scale measurable.

A tri-nary cellular automata comprised of a two dimensional grid of square cells will be used for the simulations of human crowds. Using a *Moore* neighbourhood for each cell, eight transitions are possible for each individual (nine including a stationary transition), shown in Fig. 3 below.

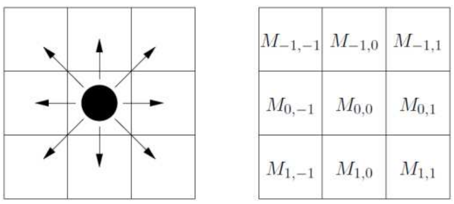


Figure 3. Grid architecture at the micro level and the possible transitions M_{ij} according to row and column variables [3].

The transitions available to each individual shown above are only possible if the requested cell is empty in the next time frame. When more than one individual requests access to a single cell, a selection is done based on probability (Fig. 4).

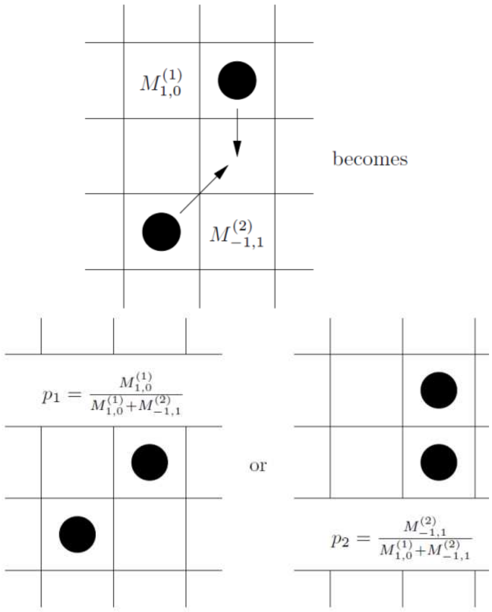


Figure 4. Probabilities associated with multiple access requests to a single cell, where $p_1 + p_2 = 1$ [3].

The probabilities in Fig. 4 above show the probabilities of a transition with no gender biasing behaviour. The probability functions for p_1 and p_2 are modified to include gender biasing behaviour for the project as follows:

$$p_1 = \frac{M^{(1)}_{1,0}}{M^{(1)}_{1,0}+M^{(2)}_{-1,1}} \pm G \qquad \text{(EQ1.1)}$$

$$p_2 = \frac{M^{(2)}_{-1,1}}{M^{(1)}_{1,0}+M^{(2)}_{-1,1}} \mp G \qquad \text{(EQ1.2)}$$

where G is the *gender biasing factor* (GBF), chosen so that

$$0 \leq G \leq 1$$

and

$$0 \leq p_1 \leq 1$$
$$0 \leq p_2 \leq 1$$

For the unbiased case, G is zero. The unbiased case will be used in circumstances where both genders are the same in a multiple access request condition. When the individuals' genders are opposite, biasing towards a certain gender (female for this project) is done by adding the GBF to that individual's transition probability, and subtracting the GBF from the other individual whose gender is opposite. All transitions to an empty cell with no other access requests are given a probability of 1. However, because diagonal transitions have a larger discrete distance associated with them compared to the horizontal or vertical transitions (by a factor of √2), the probability associated with a diagonal transition is reduced by a factor of 1/√2 and the reduction is added to the stationary transition probability if travelling diagonally. Fig. 5 demonstrates the effect of this factor, but note that the result can vary due to the properties of statistical probabilities.

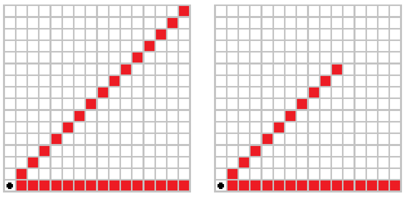


Figure 5. Reduction of diagonal transitions is done by applying a factor of 1/√2 to the probability of the transition (right) to keep travelling speeds equivalent in any direction.

For collision avoidance purposes discussed in 1.6, crowds need to be identified at the micro level and then processed at the macro level. For the purposes of this project, a crowd is defined as:

A set of two or more individuals whose positions form a body that is linked in one way or another so that no single individual is more than one discrete time step away from another in the set.

An example of a crowd and an example of a group of people not forming a crowd are shown in Fig. 6 below.

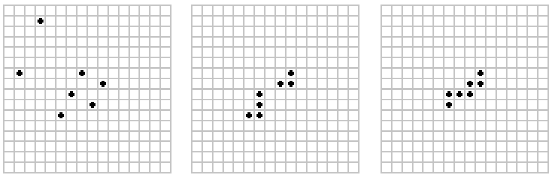


Figure 6. Crowd formation example.
From left to right: No crowds, 2 crowds, 1 crowd

The main purpose for the identification of these crowd formations is for individuals not part of a crowd to make path planning decisions based on how many people they have to navigate around to get to their destination. With no crowd model, individuals leave collision avoidance mechanisms up until one discrete step away from another individual. This is considered acceptable if the other person is not part of a group (crowd), as a simple step left (for convention) will avoid the collision. If the other person is part of a large group, the default collision avoidance mechanism will not be adequate, as there will most likely be other surrounding individuals and a collision is likely to happen. This is described in detail in 1.6.

Macro Model
The *Macro* model is the model of the system as a whole; it is the global representation of the swarm. The macro model used is not flexible enough to aid completely in the reaction mechanisms of individual bodies, which is the general case for most SI systems. Swarms are therefore primarily controlled by the behaviours of the individuals at the micro level, which cumulatively result in global patterns to emerge. However, there is one collision avoidance mechanism used by individual bodies that the macro model is used for. This involves an individual or crowd to scan the environment for a path to take. This collision avoidance mechanism is discussed in 1.6.

Using the Moore neighbourhood model, a crowd density plot for a region can be made. The crowd density is a floating point number ranging from 0 to 1, where ‘0’ represents full vacancy in the neighbourhood of a particular cell, and ‘1’ represents all cells in the neighbourhood being occupied. Only the state where a person occupies a cell is counted; environment occupying a cell does not increase the crowd density. Fig. 7 below demonstrates three different crowd density values. Environment-edge crowd densities are ignored as a full neighbourhood for each cell is not available.

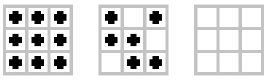


Figure 7. Crowd density example. Densities from left to right: 1.000, 0.667, 0.000

To identify bottlenecking occurrences without visually analysing the simulation results, the macro model can be used to identify densely populated regions with low travelling speeds which correspond to bottle-necking situations.

1.4 Behavioural Models

When considering computer programs that emulate life processes, life characterisation is the most prominent issue; things may be able to learn, reproduce and evolve but not need to resemble earthling life much at all. Lifelike behaviours are sometimes said to be features of life that resist entropy, as life seems to continually renew itself whilst entropy is supposedly increasing. Conway’s Game of Life generated the motivation for intense study of systems that exhibit lifelike behaviours using cellular automata (refer to Fig. 8 in Appendix E).

Wolfram’s Type 4 cellular automata seem to have the ability to perpetually renew themselves, and it is for this reason that they have become the symbol, if not the seed, for artificial life research [4].

The dynamics of the CA rule set used in the project are by no means evolutionary, but they resemble the behaviour human beings exhibit when moving in small and large groups. The behaviour models used in the project differ with various parameter adjustments. Because the global behaviour of a swarm is governed by the properties of the individual bodies it contains, mainly the properties of the individual bodies are adjusted for a change in swarm behaviour. The CA architecture used in the project supports both male and female genders, and so a focus for the project is to identify whether biasing toward one gender in a give way situation will affect the outcome, using the equations defined in 1.3 (EQ1.1) (EQ1.2). Males will generally have a lower priority of right-of-way with respect to the females; as this is inherent in the polite mannerisms most human beings exhibit. For example, if a male and female both want to use the same door (multiple access requests), the male will give-way and perform a stationary transition based on a probability which is favourable to giving way.

A collision avoidance mechanism discussed in 1.6 uses a scanning technique to find available paths to take. This technique scans the entire view frustum, which is defined by an angle. By limiting the angle at which individuals scan the environment at, different behaviours will result. The second behavioural model used in the project is dependent on the view frustum angle.

1.5 Path Planning

Path planning is a fundamental feature of a swarm's intelligence. To aid in the path planning process, vector fields have been incorporated in the CA. These fields exist in a subset of Euclidean space (the region within the environmental boundary), and have a uniform magnitude.

The idea is that a swarm particle can navigate an environment using these fields without any significant computation, as the vector fields are generated for only the swarm bodies' positions in each frame. They are generated by iteratively calculating each cell's field vector using the cell's position and a destination point. Fig. 9 below shows the field generated for a rectangular room with one exit point. Swarm particles will follow the general direction of the field and only alter their direction if it is more efficient to do so.

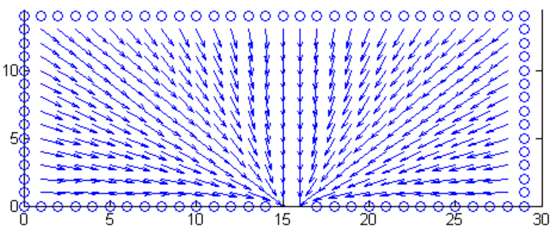


Figure 9. Example vector field generated for a room 30 x 15

The above field does not work for a discrete system, as the field leads particles through points in Euclidean space which aren't defined in the discrete model. To overcome this problem, an approximation to the field is made using eight possible directional vectors. Fig. 10 below shows the above field after an approximation has been applied.

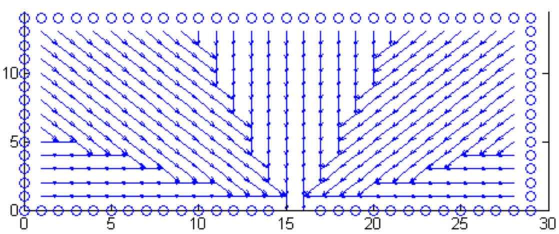


Figure 10. Approximation to the generated vector field

As discussed in 1.3, diagonal transitions will be reduced with a probability so that speeds are kept consistent in any direction (by a factor of $1/\sqrt{2}$). These diagonal transitions aren't visibly smaller in the vector fields produced, as the vector fields are primarily used for directional purposes. The reduction in diagonal transitions is done at a later step in the transition procedure.

1.6 Collision Avoidance

Collision avoidance mechanisms are a fundamental feature of a swarm particle. Without them, no recognition of other particles' behaviour would be taken and most likely, collisions would occur resulting in inefficient swarm behaviour. In this project, micro and macro level collision avoidance mechanisms are implemented so that the resulting global behaviour is efficient, but also keeping in mind the limits of human ability.

When the mechanisms are in operation, they must work in unison; meaning a convention must be adhered to for the mechanisms to work. The collision avoidance mechanisms would be of no use if, for example, one person tried to take a left to avoid a collision, while the other takes a right – a convention has to be followed. The convention is largely dependent on the country one lives in, but for the purposes of this project, a stay left convention is adopted.

The most important of the two collision avoidance mechanisms considered is the micro level mechanism. It serves as the basis for a person's intelligence, where it uses information about the closest cells to the person to make a decision (the cells in the Moore neighbourhood). A swarm is largely comprised of closely packed individuals, and so the micro collision avoidance mechanism is the dominant mechanism which affects the outcome. Individuals need to be able to judge potential collisions from a distance as well, not just within their immediate neighbourhood. The macro level collision avoidance mechanism serves this purpose, by iteratively scanning the environment from a person's perspective, identifying the most effective route to take.

2. Methodology

This section outlines which tools and methods were used to achieve the final result.

Language and tools

The project was primarily developed in Python 2.7.2 [5]; with data analysis performed using Matlab [6]. Wingware's Python IDE [7] was used for development, as it provided a user friendly text editor, code intelligence features, a graphical debugger and version control. In order to output graphical information, the TkInter module [8], a graphical user interface programming toolkit was used – otherwise known as the "Tk interface".

Cellular Automaton

In the project, the behaviour of people in crowds was modelled and simulated using a tri-nary cellular automaton with a Moore neighbourhood for each cell. The three possible states for each cell are empty, occupied by person, or occupied by environment. The basic rules for the cellular automaton used are as follows:

- A person is able to occupy an empty cell unless another person is trying to occupy that cell with higher priority

- A person cannot occupy a cell occupied by environment
- A person cannot occupy a cell occupied by a person unless the other person is moving away in the next time frame

Prioritisation of individuals will be primarily controlled according to gender for the purposes of this project, and will be varied for analysis.

Collision Avoidance

The micro level collision avoidance mechanism first takes into account the direction vector stored in the person object. This direction vector may not be pointing at a single cell, it could be pointing somewhere between one of two cells, depending on where the destination is located. The direction vector is then approximated to the nearest cell (as described in 1.5), and the path is taken. However, if another person is occupying that cell, some factors need to be considered in order to successfully avoid a collision. The main factor that determines the micro level outcome is where the other person is planning to go, because if that person is planning to move, the required cell in the next time frame could be free, but if that person is heading toward the current person, an alternative cell needs to be considered depending on the convention used (stay-left). In some cases, alternate routes are not available due to even more people occupying cells surrounding the person. In cases like these where no other alternate routes are available, a stationary transition is performed. Naturally, the surrounding cells will eventually clear and neighbouring cells will become vacant.

When a person has no obstacles in their immediate neighbourhood, the macro level collision avoidance mechanism takes over. This involves iteratively scanning the environment, so that other crowds can be identified and avoided. Crowds are formed and identified according to the definition explained in section 1.3. Each crowd has a (average) direction vector associated with it, and this vector along with the position of the crowd determines the choice of route an individual takes who is trying to avoid that crowd. An example iterative scanning process is shown in Fig. 11 below.

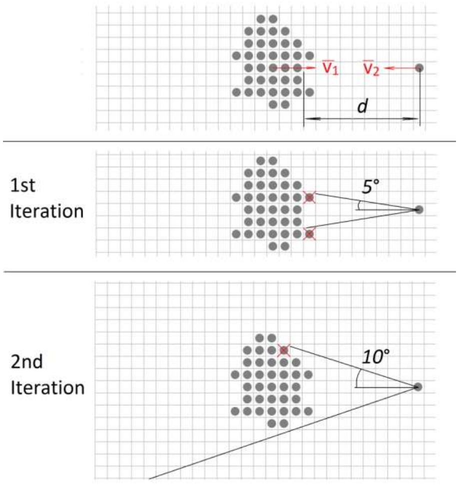


Figure 11. Example iterative process of the macro level collision avoidance mechanism

Vectors $\boldsymbol{v_1}$ and $\boldsymbol{v_2}$ in Fig. 11 above denote the directions of a crowd and individual respectively. In the case where the dot product of the two vectors is within the range -1 and -0.7, the iterative process is carried out. If the dot product of the two direction vectors is not in this range, the individual will continue heading in the original direction unless the distance $\boldsymbol{d}$ is not satisfied.

Event loop

The outcome of a scenario is dependent on the initial conditions and the CA rule set used. The event loop uses this information to determine the next frame's state according to the previous frame, and performs this operation until a desired result is obtained; such as the evacuation of a building. Two arrays are used to perform the movement operation for each frame, those being the "current" and "future" people arrays. The "current" array stores pointers to people who have not had a movement operation performed on them yet, and the "future" array stores pointers to people who have moved and are not at their final destination. For each frame, the "current" people array is looped through randomly and a movement operation is performed for each person. At the start of each new frame, the new "current" array is the previous frame's "future" array.

Data recording

Frames are generated and saved in a postscript data format, and once all frames are generated they may be collated so that they can be animated. Not only are the visuals saved, but also other information is saved such as crowd densities, potential collisions and crowd formations. This extra information is outputted to text files so that it can be processed later for analysis.

3. Results and Discussion

Two environments were chosen for testing and demonstrational purposes of the SI model developed, those being a passenger aircraft cabin and the first floor University of Canterbury engineering block.

3.1 Passenger Aircraft Boarding and Evacuation

Passenger aircraft boarding and evacuation procedures are common examples where bottlenecking of people occurs. The aircraft chosen for the simulation of these procedures was a Boeing 777-300, a large passenger aircraft seating 386 to 550 people depending on the seating arrangements [9]. For the proposed simulation, a two-class seating arrangement allowing up to a maximum of 417 passengers was used [10]. Filling every seat in a passenger aircraft of this size is considered "unhealthy", so a typical load factor of 80-85% is used and will be used for the proposed simulations, resulting in a passenger count of 335.

To assist in the predictions about how a crowd would behave when boarding and evacuating the aircraft, Air New Zealand and Boeing were contacted for more information about how the boarding and evacuation procedures are carried out for the particular aircraft [11]

[12]. Since this information is largely classified, no information regarding the evacuation of the cabin was obtained apart from in online sources [13], but boarding times for a normal boarding procedure and a random boarding procedure were given.

For the seating arrangement considered, airlines normally board passengers seated in business class first, followed by premium economy passengers, and then the rest of the economy passengers from the rear – all through one of the doors closest to the front of the aircraft. The seating arrangements can be seen in Fig. 12 below.

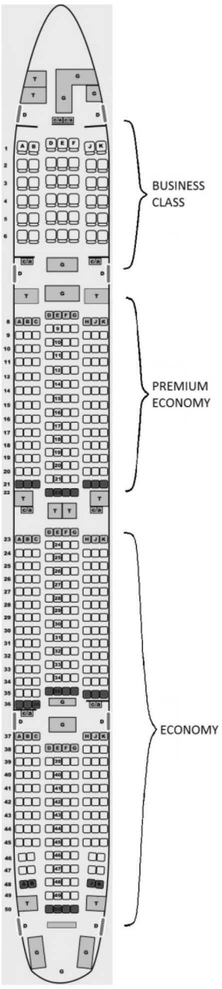


Figure 12. Seat configuration for a two-class Boeing 777-300

This boarding procedure usually takes 32 minutes with a load factor of 80-85%. A random boarding procedure allows passengers to board the aircraft when pleased and they do not have to wait until a certain class has boarded. When passengers board randomly, all passengers are usually boarded 2 minutes earlier (real recordings) [11].

Because no passenger evacuation times were obtainable, a simulation was carried out to make an estimation. In order to obtain accurate results, an SI model was needed which closely resembled the passengers in the real scenario. The SI model developed modelled peoples' swarm behaviours on their fundamental traits, such as their path planning and collision avoidance abilities. The behavioural model could be changed to match that of the passengers boarding/evacuating the aircraft by altering give way properties, mainly gender and view frustum related. Since only boarding duration times were available, the boarding procedures were simulated first so that the ratio between time and frames could be found. This is because the time taken to perform any boarding or evacuation procedure will be proportional to the number of frames that it takes.

A scale model of the Boeing 777-300 was constructed and a boarding simulation was performed. The main purpose of this simulation was to find out the number of frames required for a typical 32 minute boarding procedure. The results from the normal boarding simulation are shown in Table 1 below (see Appendix A for visuals).

Table 1: Normal Boarding Simulation Results

Time	(0 d.p.)	32m 0s
No. people		335
No. frames		595
Cell width	(3 d.p.)	0.446m
Seconds per frame	(2 d.p.)	3.23s
GBF	(3 d.p.)	0.000
View frustum	(0 d.p.)	90 degrees

The behavioural model used a GBF of zero, meaning a male was just as likely to give way to a female as a female was to a male; in addition, a view frustum of 90 degrees was used which was considered a default field of view. Only once an entire class was boarded were the next class allowed to board. For the real boarding duration times supplied, it is unknown whether this method of boarding was used. For example, the boarding method could be that once the last person for a specific class enters the aircraft, the next class may be allowed to board before the previous class has even gotten to their seats. This is discussed in further detail after the random boarding simulation results.

Next, the random boarding simulation was performed, with an expected boarding duration time of 30 minutes (2 minutes less) or 558 frames (595*30/32), according to the real data obtained. The results from the random boarding simulation are shown in Table 2 below (see Appendix B for visuals).

Table 2: Random Boarding Simulation Results

Time	(0 d.p.)	19m 54s
No. people		335
No. frames		370
Cell width	(3 d.p.)	0.446m
Seconds per frame	(2 d.p.)	3.23s
GBF	(3 d.p.)	0.000
View frustum	(0 d.p.)	90 degrees

The time taken to perform the simulation is based on the previous boarding result which gave the number of seconds per frame. Based on the number of frames taken to perform the procedure for this simulation, a time (duration) could be estimated as follows:

$$Time = 32 \cdot \frac{370}{595} = 19.899 \text{ minutes (3 d.p.),}$$
$$\text{or 19m 54s (0 d.p.)}$$

For the random boarding simulation, the same crowd model as in the first normal boarding simulation was used.

This meant the same GBF and view frustum was used and the same crowd behaviour was inherent. Surprisingly, the aircraft was boarded in approximately 20 minutes, which was about 10 minutes less than expected according to the real data obtained. This meant one of two things; either the crowd model didn't simulate the human behaviour realistically, or the procedure that the simulation used to board the passengers was different from the procedure which generated the real times given. From analysing the footage for the normal boarding simulation, it is clear that there could be massive differences in boarding duration times if classes were boarded straight after one another, as opposed to having the passengers wait until the class before them is seated. Because of this, it was suspected that airlines board classes as soon as, or soon after the previous class has entered the aircraft; this would account for the 10 minutes difference between the two simulations (was meant to be 2 minutes).

Although it was quickly assumed that the boarding procedure used was the source for the error, the GBF and view frustum used could have been a factor also. To eliminate the possibility of the GBF having an effect, the exact same simulation (random boarding procedure) was redone with a GBF favouring females in a give way situation (0.5 GBF maximum), keeping the view frustum the same (90 degrees). The results for the test are shown in Table 3 below, with the original simulation shown in bold.

Table 3. GBF Test using Random Boarding Procedure

GBF	No. People	No. Frames	Time	View Frustum
0.0	**335**	**370**	**19m 54s**	**90**
0.1	335	377	20m 17s	90
0.2	335	356	19m 9s	90
0.3	335	364	19m 35s	90
0.4	335	359	19m 18s	90
0.5	335	372	20m 0s	90

It was found that the crowd transition time was very similar to the original time obtained (19m 54s), with the ratio between males and females being approximately 1:1 after randomly generating the genders and positions of the individuals – where the same randomly generated initial conditions were used throughout the entire test. Females were expected to "lead the pack", as they would have been given way to more frequently; however this was not the case, possibly because the entire swarm travelled at the same speed and so females would not have been given the opportunity to overtake other individuals. It would take extensive testing to find any large dependencies on the GBF with regards to the boarding duration time, and so further testing regarding the behavioural model for this scenario was abandoned. It was concluded that GBF did not appear to affect the boarding duration time significantly. The only other variable that needed to be analysed was the view frustum for each individual, which was altered with GBF kept constant (0.0). Using the same randomly generated initial conditions as in the GBF test, the relation of view frustum to boarding time was analysed in Table 4 below.

Table 4. View Frustum Test using Random Boarding Procedure

View Frustum (degrees)	No. People	No. Frames	Time	GBF
0	335	373	20m 4s	0.0
45	335	360	19m 22s	0.0
90	**335**	**370**	**19m 54s**	**0.0**
135	335	375	20m 10s	0.0
180	335	389	20m 55s	0.0

From the results, no significant difference in boarding time can be seen with variations of the view frustum. Because each individual is relatively close to other individuals during the boarding procedure, the Moore neighbourhood collision avoidance mechanism is the primary method used and so the view frustum does not affect the outcome considerably.

Assuming the SI model simulated passengers relatively realistically; a cabin evacuation time was predicted based on the number of frames taken to perform the operation. The result obtained has an error associated with it as not enough (boarding) information was available to accurately predict the behaviour. Using the same conditions as the boarding procedures, an evacuation simulation was performed with people populating the aircraft in a random seating arrangement. The results are shown below in Table 5 (see Appendix C for visuals).

Table 5. Cabin Evacuation Results

Time	(0 d.p.)	4m 33s
No. people		335
No. frames		85
Cell width	(3 d.p.)	0.446m
Seconds per frame	(2 d.p.)	3.23s
GBF	(3 d.p.)	0.000
View frustum	(0 d.p.)	90 degrees

Using the results from the normal boarding procedure to predict the time, the evacuation time is calculated as follows:

$$Time = 32 \cdot \frac{85}{595} = 4.571$$ minutes (3 d.p.), or 4m 33s (0 d.p.)

Because there was not enough information about boarding procedures to know if the SI model used is accurate, a maximum error percentage in the result was calculated as follows (where RBT is the Random Boarding Time):

$$error = \left(\frac{real\ RBT - simulated\ RBT}{simulated\ RBT}\right) \cdot 100$$

$$error = \left(\frac{30-19.899}{19.899}\right) \cdot 100 = 50.76\% \quad \text{(EQ3.1)}$$

(2 d.p.)

The prediction for the time taken to evacuate the entire cabin, with all doors accessible, is not unreasonable but can only be verified with more information about how long it takes people to traverse the aircraft; which was just not available.

3.2 UC Engineering Block Evacuation

The University of Canterbury engineering block is a multi-story building, and populates many people during the day. In the event of an emergency, bottlenecking could easily occur and could affect the evacuation procedure, and so it was decided to determine how long a section of this building would take to evacuate. Because no real data was available, approximations had to be made about how fast people would traverse the environment. The results are shown below in Table 6 (see Appendix D for visuals).

Table 6. UC Engineering Block Evacuation

Time	(0 d.p.)	7m 32s
No. people		200
No. frames		115
Cell width	(3 d.p.)	0.815m
Seconds per frame	(2 d.p.)	3.93s
GBF	(3 d.p.)	0.000
View frustum	(0 d.p.)	90 degrees

To calculate the time for the evacuation procedure to complete, relative speeds of individuals were used based on the speeds obtained from the aircraft boarding simulation results. It was assumed that individuals would travel approximately 1 ½ times faster, as the environment is not as dense and movement limiting as the aircraft cabin. The following calculations are done to calculate the duration of the evacuation procedure:

$$Cell\ Width\ Ratio = \frac{0.815}{0.446} = 1.827$$

$$Equivalent\ frames = CWR * frames = 1.827 * 115 = 210.146 \text{ frames}$$

$$Time = \frac{210.146*32}{595*1.5} = 7.535 \text{ minutes (3 d.p.), or 7m 32s}$$

Of course, this result has the same error (EQ3.1) associated with it as the aircraft evacuation result. The only way this error could be reduced was with a verification test of the SI model, which was attempted using the real duration times for normal and random boarding procedures. Unfortunately, alternative methods of boarding were identified which if used, would alter the result dramatically resulting in the uncertainty (EQ3.1).

3.3 Future Development

A robust SI model using a CA has been developed, but there is always room for improvement. Below a variety of improvements have been identified and are discussed briefly.

Accuracy of results
Despite the uncertain results obtained from the simulations, potentially very accurate results could be achieved if a real simulation is replicated and the SI model is verified to represent the real swarm behaviour. As mentioned, there was not enough procedural information about the real simulation that was replicated in attempt to achieve accurate results.

Modelling
In order to more accurately model the environment and people with a cellular automaton, the cell resolution could be increased. This would mean the Moore neighbourhood of each person would have to be increased in order to keep the scale correct. For example, if twice the resolution was to be used, the Moore neighbourhood would be increased two fold, and individuals would have to keep at least an extra cell between them and other individuals to avoid a collision.

Visuals
To improve visual quality, OpenGL, a cross-platform API for writing applications that produce 2D and 3D computer graphics is considered, allowing better observation of simulated scenarios.

Optimisations
The time taken to perform a simulation was proportional to the number of bodies in the simulated swarm. The majority of time taken, however, was processing the postscript data format with which the visuals were saved in. By using a different data format or by moving to a 2D/3D graphics API as mentioned above would further optimise the system.

4. Summary and Conclusion

Many obstacles have been overcome in order to develop a robust crowd simulation tool from scratch. The project supervisor, Dr. R. Mukundan, was especially helpful in the developmental process and provided great insights into the system architecture. The system produced is able to simulate large scale environments with potentially limitless individuals at high speeds due to the cellular automaton approach (also dependent on the computer hardware available). As a result, the potential is there for the extraction of statistical information about the system using a Monte Carlo method.

In order to predict times with accuracy, at least one real simulation needs to be verified with an artificial simulation, so that the SI model used is known to simulate people’s behaviours accurately. This was attempted with the normal and random boarding procedures of the Boeing 777-300, but not enough information was available so a large error was inherent in the results of proceeding simulations. Regardless of the accuracy (to be verified in future work), several simulations were performed to show the ability of the system. Boarding and evacuation procedures were performed on a Boeing 777-300 cabin, and the evacuation procedure of the University of Canterbury engineering block was performed.

The simulation tool developed may be readily used to verify crowd dynamics in any two dimensional scenario. Although the behavioural models used did not appear to affect the outcome of a simulation, future modifications such as speed variance of individuals could enable the influence the behavioural models have on a swarm.

5. References

[1] C. Reynolds. “Craig Reynolds”. Internet: http://www.red3d.com/cwr/, Dec. 7, 2010 [Sep. 1, 2011]

[2] E. Weisstein. “Cellular Automaton”. From MathWorld--A Wolfram Web Resource. Internet: http://mathworld.wolfram.com/CellularAutomaton.html, [Sep. 1, 2011]

[3] A. Schadshneider (2001), “Cellular Automaton Approach to Pedestrian Dynamics – Theory”, University of Koln, Germany.

[4] J. Kennedy, R. Eberhart (2001), “Swarm Intelligence”

[5] Python community, “About Python”. Internet: http://www.python.org/about/

[6] The MathWorks Inc., “MATLAB – The Language Of Technical Computing”. From Mathworks. Internet: http://www.mathworks.com/products/matlab/index.html, [Sep. 1, 2011]

[7] “Wingware Python IDE”, From Wingware. Internet: http://wingware.com/, [Sep. 1, 2011]

[8] “TkInter”, From Python. Internet: http://wiki.python.org/moin/TkInter, [Sep. 1, 2011]

[9] Aerospace Publications, “The Boeing 777-300”. From Airliners.net. Internet: http://www.airliners.net/aircraft-data/stats.main?id=107, [Jul. 11, 2011]

[10] “Emirates – Boeing B777 300ER (two class) Seatmap”. From seatmaestro. Internet: http://www.seatmaestro.com/airplanes-seat-maps/emirates-airlines-boeing-777-300-c2.html, [Jul. 12, 2011]

[11] D. Hiscotte. Airport Performance and Standards. Interview, August 2011.

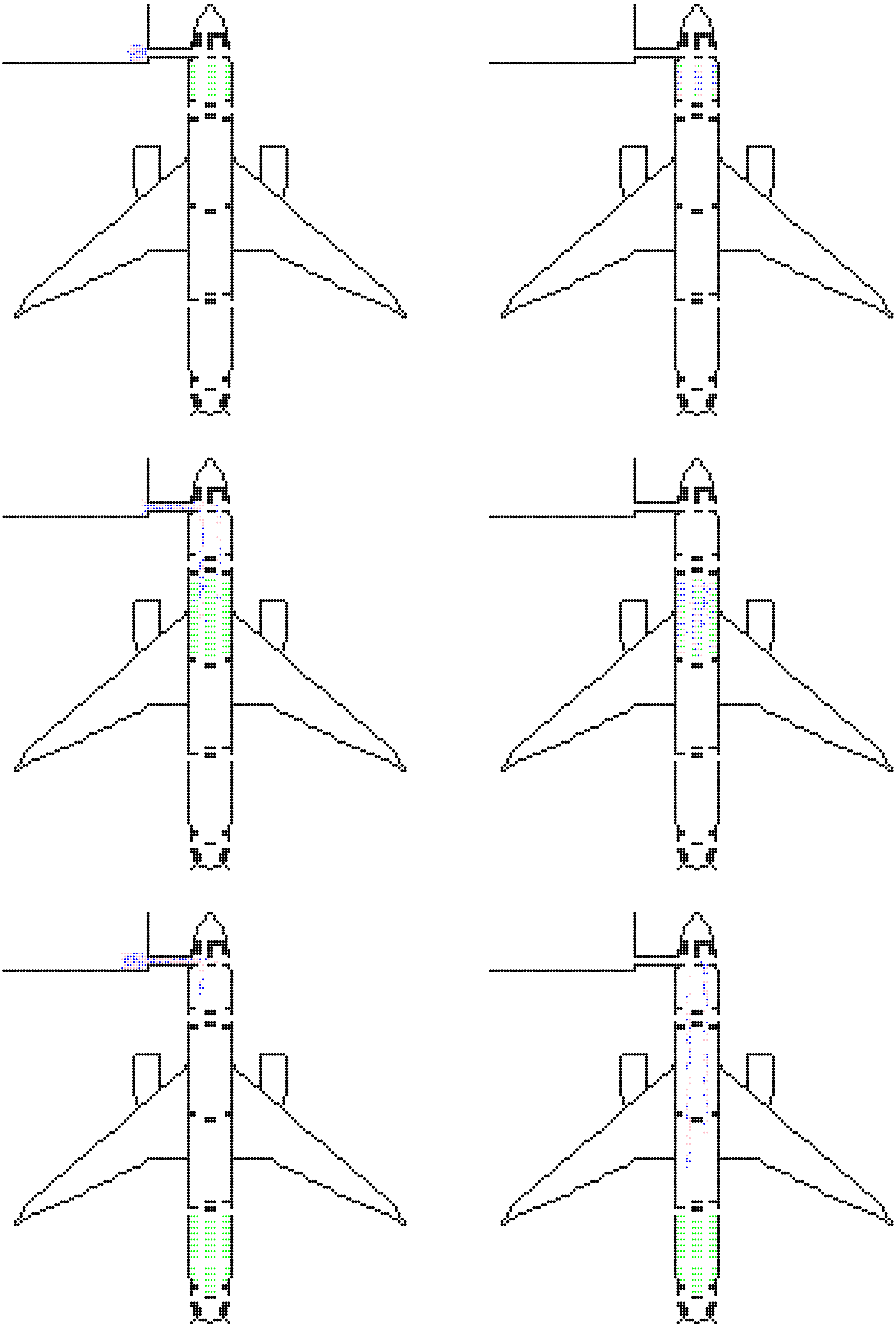
[12] J. Palmer, “Tests show fastest way to board passenger planes”, From BBC. Internet: http://www.bbc.co.uk/news/science-environment-14717695, Aug. 31, 2011 [Sep. 1, 2011]

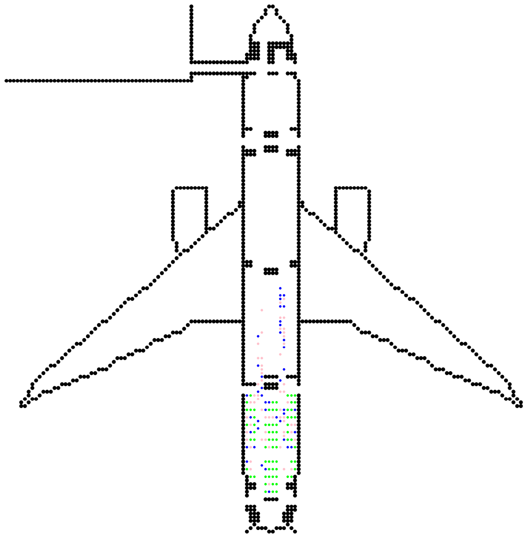
[13] A. Damski, L. Richardson, “New Evacuation Techniques”. Internet: http://www.warman.demon.co.uk/anna/boeing.html, [Aug. 14, 2011]

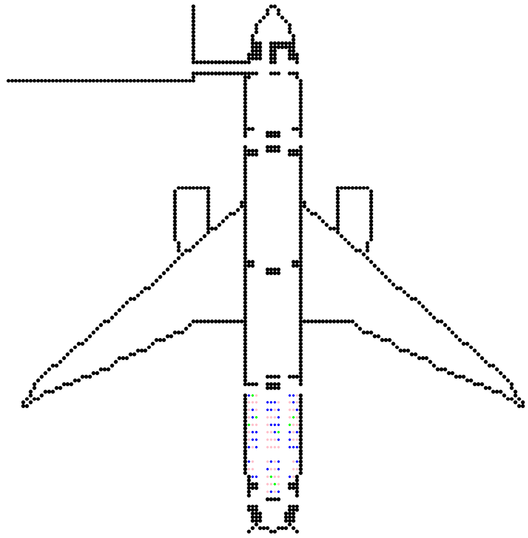
[14] “777 Family”. From Boeing. Internet: http://www.boeing.com/commercial/777family/pf/pf_300product.html, [Jul. 11, 2011]

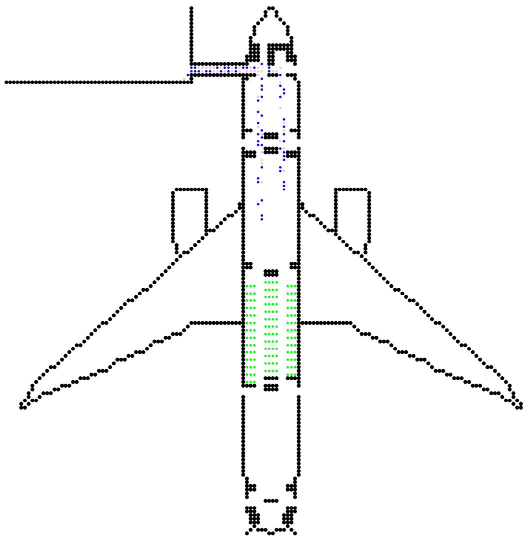
[15] T. Lightfoot, G. Milne, “Modelling Emergent Crowd Behaviour”, University of Western Australia, Perth, Australia

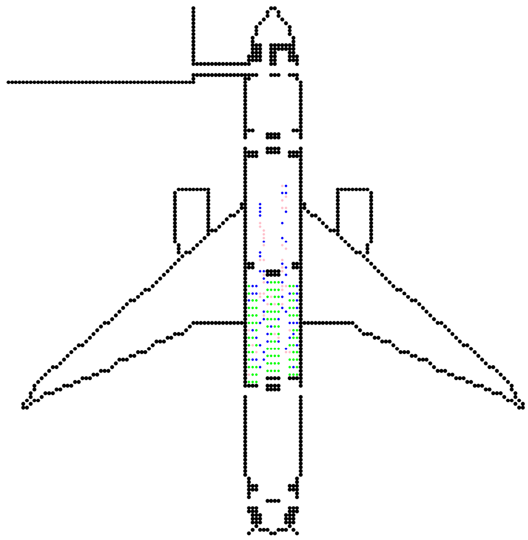
Appendix A

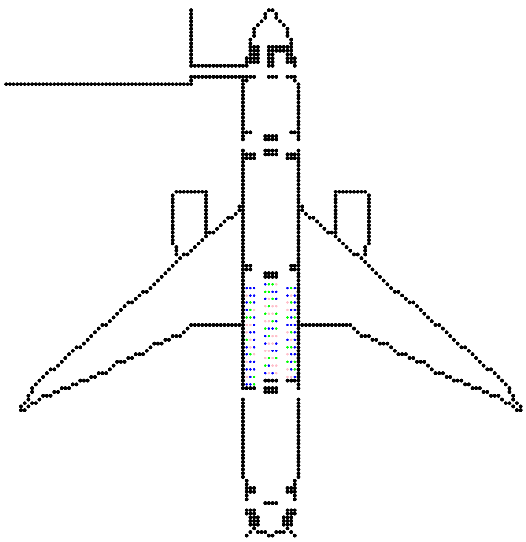




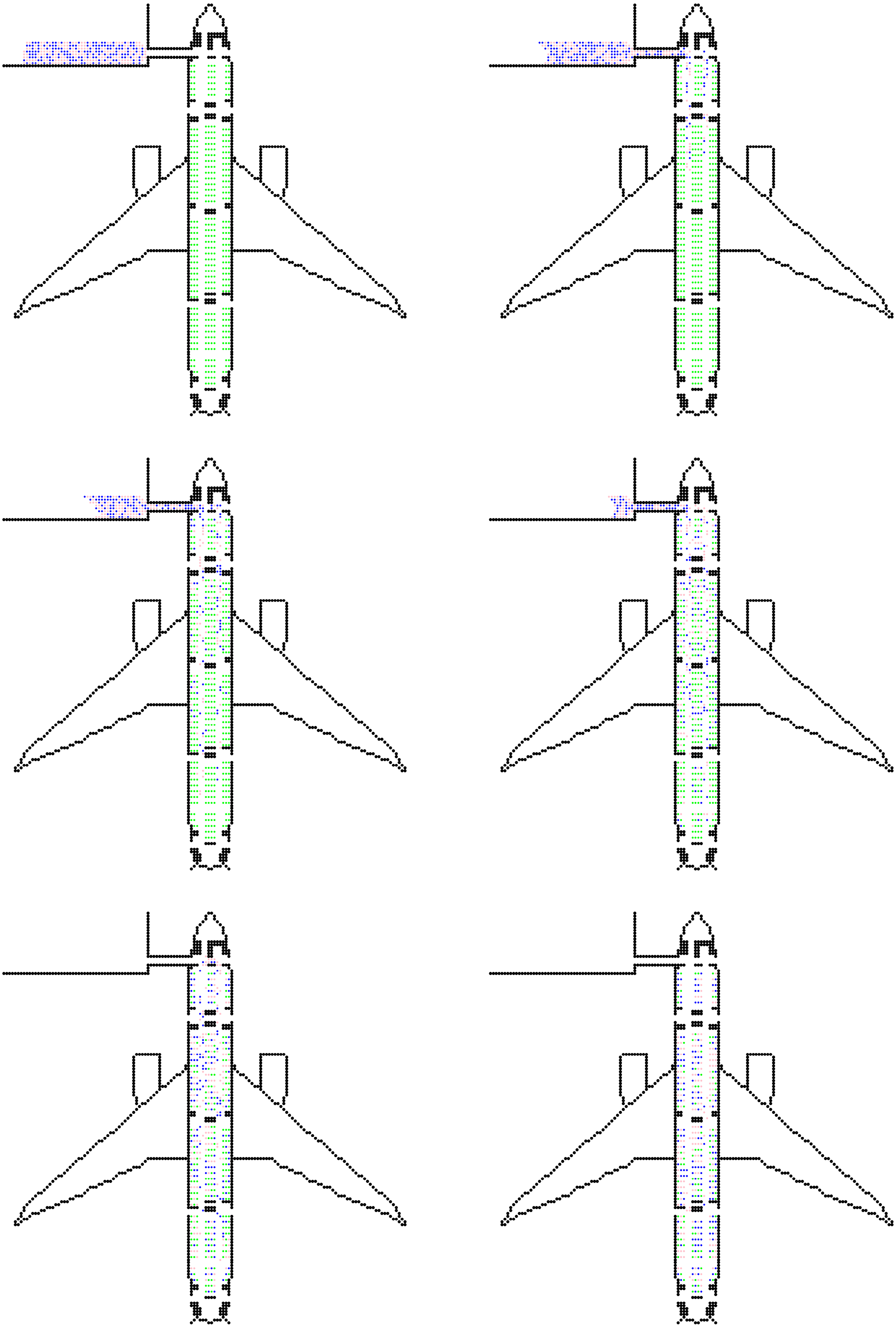


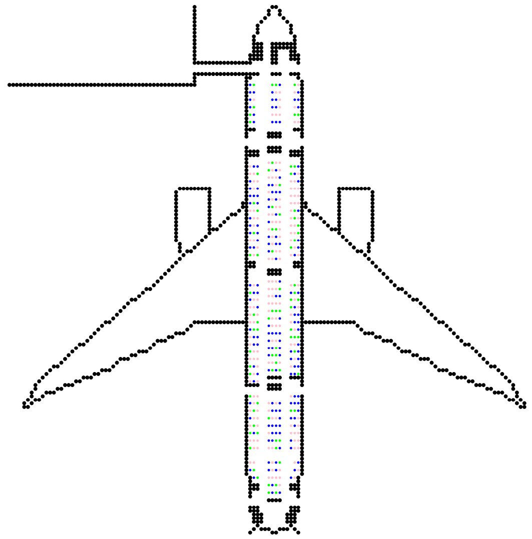




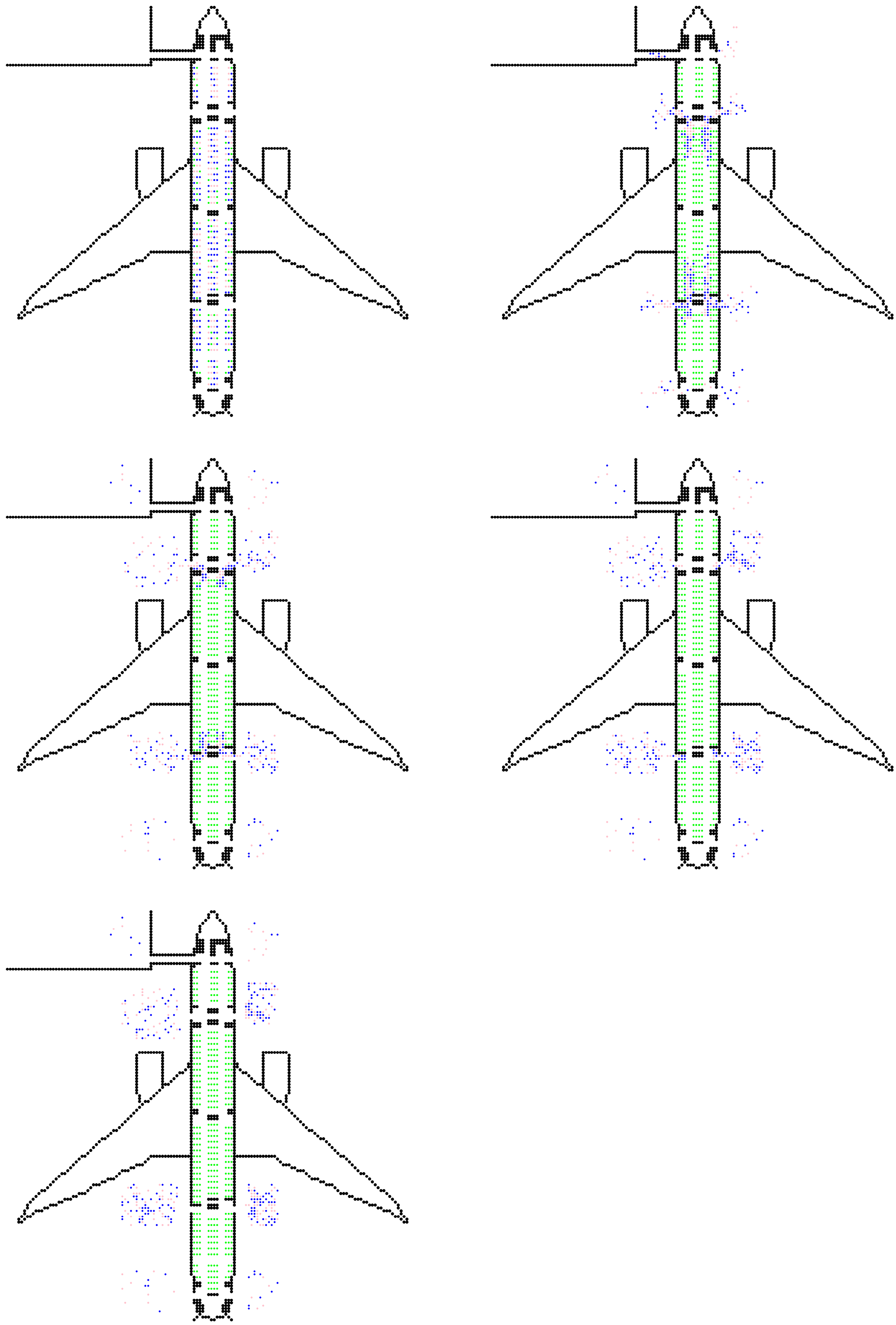


Appendix B





Appendix C



Appendix D

Appendix E

Pattern	111	110	101	100	011	010	001	000
New state	0	1	0	1	1	0	1	0

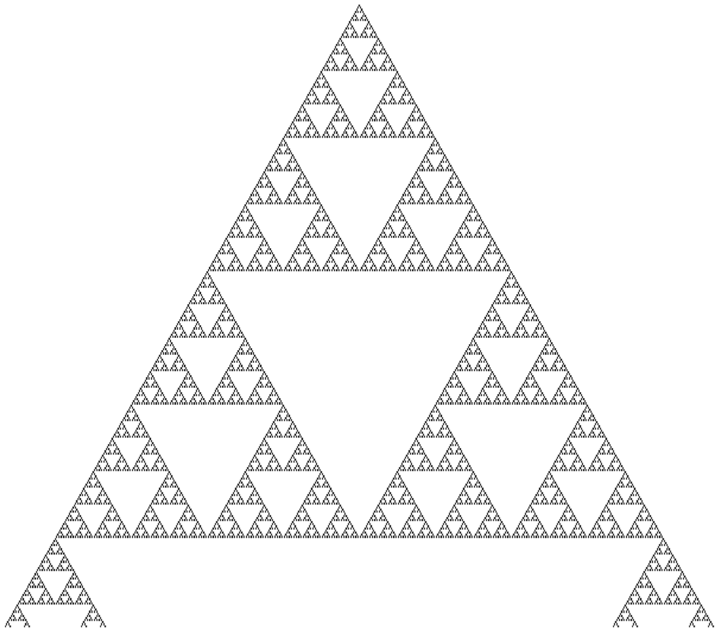


Figure 2. "Rule 90" generating the Sierpinski triangle

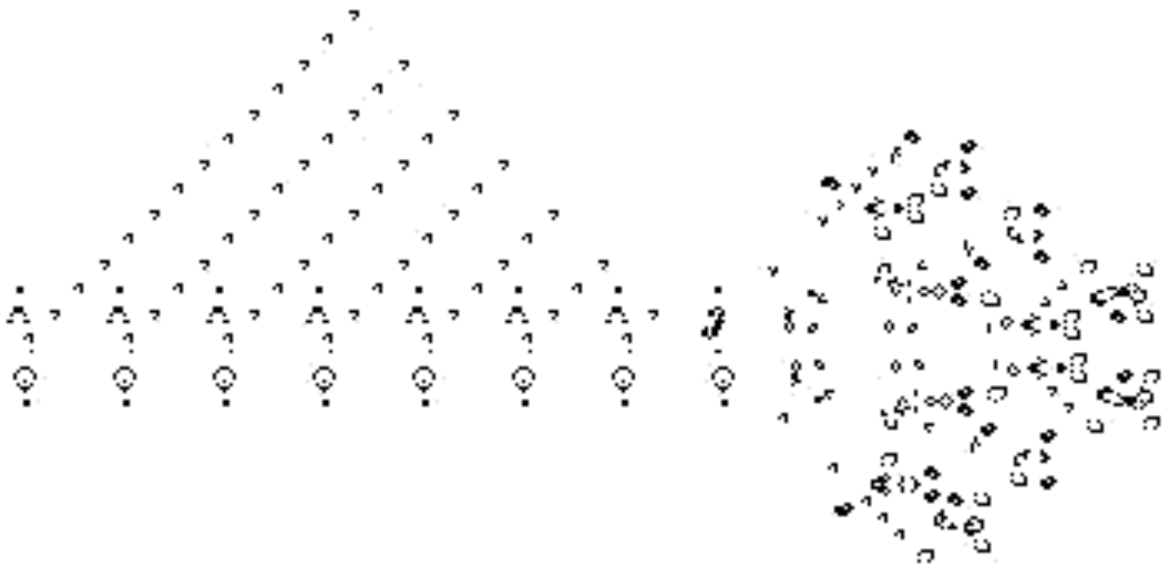


Figure 8. Evolution of a "breeder" that leaves "glider guns" in its wake (Conway's Game of Life)