Agent-Based Modelling and Simulation for Crowd Behaviour and Control Studies

Héctor C. Alzate Ramírez

October 2014

Abstract

Crowd control and behaviour studies are gaining importance as police departments around the world start working on ways to better train officers. For such task we offer Agent Based Modelling and Simulation as a tool to test and evaluate the possible outcomes of the interactions of different kinds of artificial agents conforming crowds in urban areas. Doing so with a realistic human-like crowd behaviour based on Sociological studies. We use the Vancouver Riot event as a test scenario with a virtual model of the area near to the stadium.

1 Introduction

A crowd is the agglomeration of a considerable amount of people with similar intentions in a specific zone. Sometimes the behaviour of such groups can get conflicting and nocif to the well-being of surrounding areas or citizens, or even for their own security. We are going to focus in the simulation of crowd behaviour in urban areas, one of the most common scenarios is when police officers have to handle huge groups of sport's fans that get angry because of the outcome of an important match or other sport's event. These situations create a state of collective hysteria that submerges individuals in an unconscious state where they feel powerful an behave in aggressive and barbaric ways.

Therefore, police officers have to intervene and control the behaviour of such individuals in order to reestablish order and mitigate the damage caused to public property, bystanders and the crowd itself. To do so, police officers are trained with different tactics and strategies to control crowds in urban areas; these strategies include the usage of human barricades and non-lethal weapons and are planned thinking on critical zones and objectives the officers should protect.

An agent-based model can be used as a tool to test different strategies to control crowds in dangerous situations without harming people nor causing any kind of damage to public property. In order to be use-full, such tool should properly simulate human-like crowd behaviour; which can be expressed as coherent path-finding, realistic interactions between agents and their surroundings. We

propose to use an heterogeneous environment where different types of individuals interact in the crowd, the different types we will use are: *instigator, guardian and by-stander*; each one of these classes of crowd react in different ways to external stimuli and will be discussed further in this document. These behaviours and taxonomy are based on prior sociological Studies on the field.

As a case study we are going to use a 3D model of the area surrounding the Vancouver Stadium. There, the crowd tries to get to a local commercial stores area that could get pillaged or damaged by the crowd.

2 Literature Review

Agent-based modelling and simulation have been widely used to explore social issues in a huge range of different areas (Balietti, 2012). These areas include but are not limited to: *group dynamics, agglomeration and segregation, crowd dynamics, social conflicts and collective intelligence.* Agent-Based models have shown to be of great value while handling situations that present complex human behaviour, because even though human way of decision making is hard to model accurately, sometimes agent-based modelling is the best tool we have to work with real world data in these situations. In other areas where agent-based models are being used, the main focus is to predict real world events based on the results; this is not necessarily common or true for the application of agent-based models in social sciences, a model can be used as a powerful tool to gain insight in real world situations.

Many different agent-based crowd models have been created using different techniques and approaches. There are 2 different approaches when modelling crowds: one could start with a really simple model with a limited set of rules in order to look for *emergent behaviours* in the model *(Keep It Simple Stupid)*, or start with a real and complex model that accurately reflects the behaviour showed by individuals, making simplifications wherever possible and justified (Keep It Descriptive Simple). The trade-off presented when both approaches are contrasted is *performance* vs *realism.*

Different ways to model crowds as agent systems have been proposed (Thalmann, 2007). The most common one is to gather information about real-world crowds and abstract different properties, behaviours and events that determine the way the crowd works. This approach relies in the specification of a system as a set of interacting pieces and the corresponding relationships between them. If done appropriately the model should consider different sociological aspects that regulates the way humans interact in crowds, e.g., personal space, social conventions and leader-follower relationships among others; that could lead to complex effects on the crowd, such as polarization, loss of individuality and dramatic changes in the group's structure. Other more complex approaches are the use of computer vision to find patterns in crowd movement, usage of fuzzy cognitive maps to find the path towards a direction or machine learning and evolutionary algorithms to create intelligent agents that create different sets of rules and have the ability to modify them while the program is running.

As agent-based system gained complexity, the performance of such programs started to be an important issue when modelling and programming the simulations; because when we have a huge amount of agents each step of the simulation can take several seconds to calculate the decisions and actions each one of them makes within the system, these calculations would become more expensive if the population of agent grows or becomes heterogeneous (composed by different types of agents requiring independent calculations for each one). These concerns have been the topic for many important papers in this area. In (Xiong et al., 2010), a framework to take advantage of detailed modelling (microscopic) and simplified modelling (macroscopic) is proposed, the idea is that because individuals suffer a loss of individuality in the most dense parts of the crowd, it is of no use to make the detailed calculations there; therefore saving performance while keeping a realistic behaviour. However, the model should be provided with operations to determine whether an specific agent decision process should be executed or not, and change that state in order to get a smooth transition from one to another, this technique is called *hybrid modelling*.

Different psychological theories have been constructed to describe the behaviour of individuals inside a emotionally unstable crowd; as (Hoggett and Stott, 2010) shows, the most widely known and used are the Instigator and ESIM models. The *Instigator* model describes individuals who participate in crowds as people seeking to create mayhem and take advantage of the anonymity to hide their bad behaviour. Also depicts those individuals as malleable, people that can be easily manipulated or influenced to follow a smaller group of troublemakers. In the other hand, the *ESIM (Elaborated Social Identity Model)* explains that the behaviour of the crowd is the outcome of the interactions of a group of diverse individuals with different intentions and characteristics; they also react to the actions performed by external actors, such as the police officers dealing with them.

Behaviour-oriented Artificial Intelligence have been used through a diverse amount of approaches both in computer games and simulations. In (Mount, 2013) a brief of most widely used approaches is developed. Being *HFSM (Hierarchical Finite State Machines)* the most relevant to our problem; A state machine is a data structure that contains a set of different nodes (states), a set of transitions between them as well as the conditions that must occur in order for said transitions to be performed; each state can refer to different common behaviours that digital agents must perform in a simulation. Even further, each state inside a HFSM can hold other states within itself, being a Finite State Machine itself. This compound behaviour allows us to model sets of behaviours and transitions between them in a structured and coherent way.

Human movement has been an interesting challenge when realism is intended on virtual models. Even further, crowd movement offers it's own challenges; each agent must react to the movement of agents among itself, and even use it as a variable to calculate it's own movement. In (Reynolds, 1987), the movement of a group of agents called *boids* is described; such agents follow 3 simple rules to move in a group as referenced from:

1. Two boids that are closer than a threshold will move apart so as to avoid collision.

2. A boid will seek to match the velocity (including direction) of other boids nearby.

3. A boid always seeks to be in the crowd, at the centroid of the surrounding boids.

An applied version of this approach using a set of circumferences to define the different sets of boids that should be considered for each one of said rules can be found in (Chiang et al., 2009).

Another method for agent movement can be found in (van den Berg et al., 2011); each agent handles the others as obstacles that he must avoid using a geometrical approach, and by achieving a set of pre-conditions a collision-free movement can be ensured by all the agents until they arrive to their destinations.

3 Theoretical Model

We propose a model with 3 kinds of agents that interact with each other and their environment based on the current state of the system and their internal situation. We model their behaviours as HFSMs that can make a transition between states based on an *rage* level and a set of thresholds that can be easily configured. Each state can perform one or more from the following set of actions:

1. *Move:* Each agent is able to find a way to get to a given place inside the model; they do so keeping a collision-free movement coherent with their surroundings.

2. *Break:* Agents are able to damage objects in the model, such as cars and stores that they find in their way.

3. *Influence:* An agent can be influenced by agents close to him either in a good or a bad way, modifying it's rage levels.

The rage level of each agent can change randomly through time within a threshold, making each run of the simulation possibly different but similar in terms of outcomes and results. Also, the distance and rage level changes the way they influence agents near to them.

3.1 By-stander

A by-stander is somebody that just happens to be in the place where the crowd is occurring. They have no specific intention to damage anything, however they can be influenced by other agents to do so.

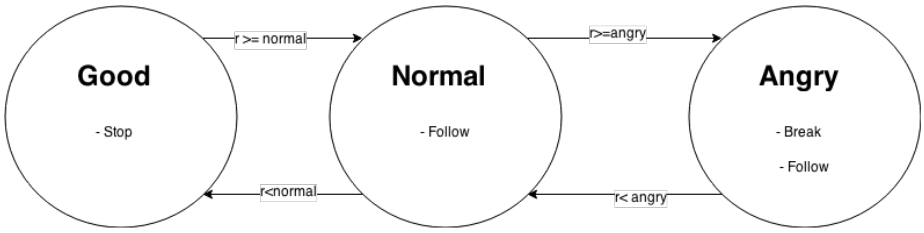

Figure 1: By-stander behaviour HFSM

3.2 Instigator

Instigators are chaos-making agents that influence by-standers to follow them towards a more vulnerable area and break things through their way there.

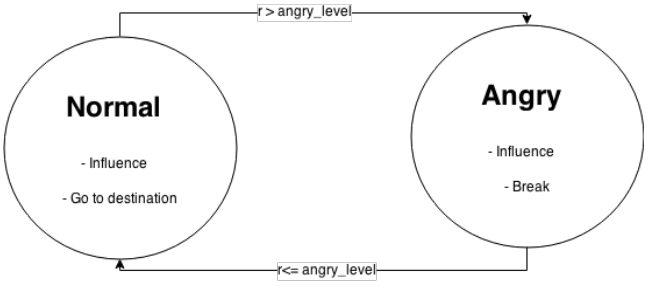

Figure 2: Instigator behaviour HFSM

3.3 Guardian

Guardians try to make by-standers to calm down and protect objects that can be broken from other agents by putting themselves in their way.

Figure 3: Instigator behaviour HFSM

4 Design

The proyect was developed with a behaviour oriented architecture, allowing similar behaviours to be shared by the agents. The following is a class diagram depicting the overall architecture of the simulation. We used the engine Unity3D, and therefore reference some of its components.

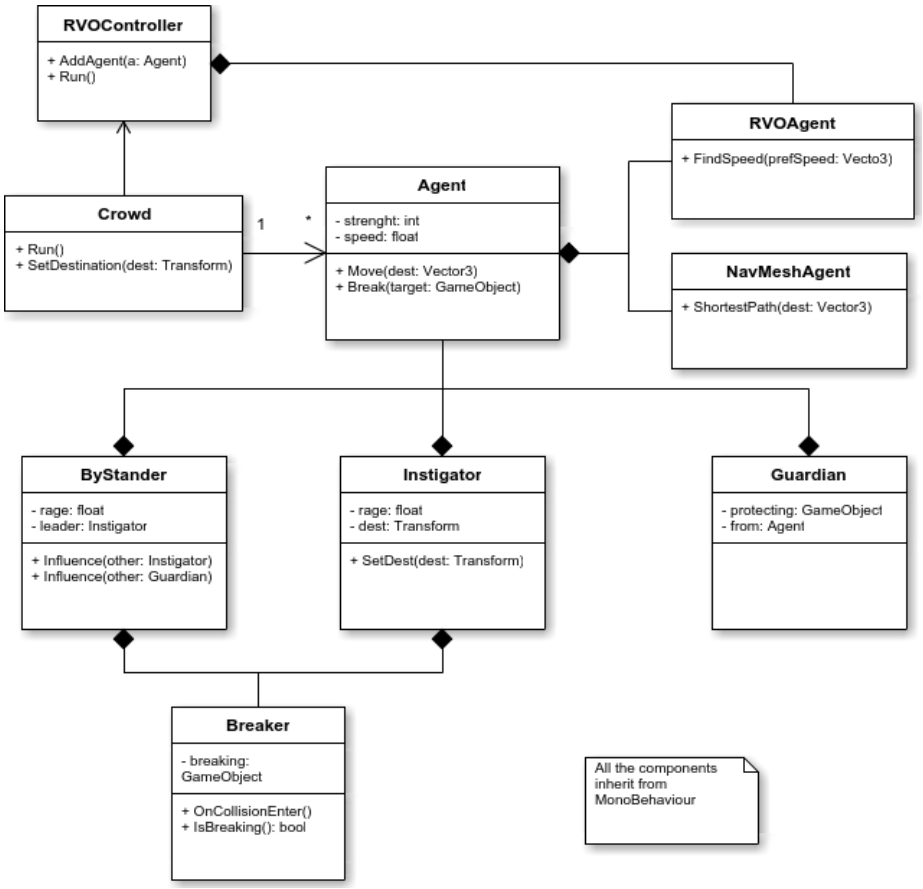

Figure 4: UML Class diagram

The following are some relevant pseudo-codes:

4.1 Movement

We begin by using Unity's path finding component to find the shortest path towards the agent destination. We use it in the rvo motion calculating algorithm as a parameter, which will find the collision-free velocity closer to the preferred one.

```
begin Move(dest)
    prefVel := navMeshAgent.FindPath(dest)
    vel := rvoAgent.FindVelocity(prefVel)
    controller.Move(vel)
```

4.2 Influence

The way each agent is influenced is changed by different variables: influencing agent rage, distance, agent influenciability.

```
begin Influence(leader)
    if(neutral || (angry && !isBreaking)) &&
        (!isFollowing || leader == following)) then

                setDestination(leader.destination)
                following := leader
    endif

    d := 1 - distance(leader) / maxD
    rage := clamp(rage + leader.rage * inf *
            d * leader.infl, 0, leader.rage)
```

If the agent is not already following a leader, or the leader it is following is the one influencing him at the time, he will change his destination to the leader's one.

5 Implementation and Technical Details

The simulation was developed in the game engine Unity3D using C-Sharp as scripting language, the engine can be downloaded from `http://unity3d.com/downloads`.

Also, agent movement using the Reciprocal Collision Avoidance algorithm detailed in (van den Berg et al., 2011) was found in `http://gamma.cs.unc.edu/RVO2/` and adapted to be used.

5.1 Movement

We rely in Unity's NavMesh component, which takes the environment 3D model and following a series of specifications, creates a set of mesh vertices's that the agents can walk on. Also, using the A* algorithm, they allow us to find the shortest path towards a destination.

However, in order to be able to use the RVO component. A interface was developed to deal with some inconsistencies; such as mapping 3D space to the 2D domain (which the RVO library uses), and the construction of 2D representation of the obstacles inside the model from the mesh's vertices before the simulation starts.

When a new agent is created, it is added to the RVO simulator and configured with a RVO agent, which will be used to calculate a collision free movement in each frame. The state of the RVO agent and the velocity used on the CharacterController component are synced in each FixedUpdate call.

The RVO library uses parallel threads to be able to compute the velocities individually for each agent.

5.2 Vision

In order to detect other agents as well as interactable objects in their surroundings, the agents must have a vision system. We used Unity3D physics engine to allow us to develop this behaviour.

Each agent has a Sphere collider attached to it, this collider calls the OnTriggerEnter and OnTriggerStay events in all MonoBehaviours attached to the agent. We difference between other agents and objects by the tags that each GameObject has. In order to avoid this collider to push other objects and agents, it is marked as a Trigger, disabling any physical behaviour beside the collision detection.

5.3 Animation

All the animations are controlled by the Animator component, it depicts a state machine with 2 states: idle and walking. The changes between these 2 states are triggered by the velocity each Agent has.

The models and animations used are available at `https://www.mixamo.com/`.

6 Tests

Qualitative tests to validate the appropriate behaviour of the model were designed.

6.1 Crowd Movement

In order to test the path-finding behaviour, we placed the final destination in an arbitrary location marked by a flag, after the simulation was run the view was switched to 3rd person camera in order to better see the agents behaviour.

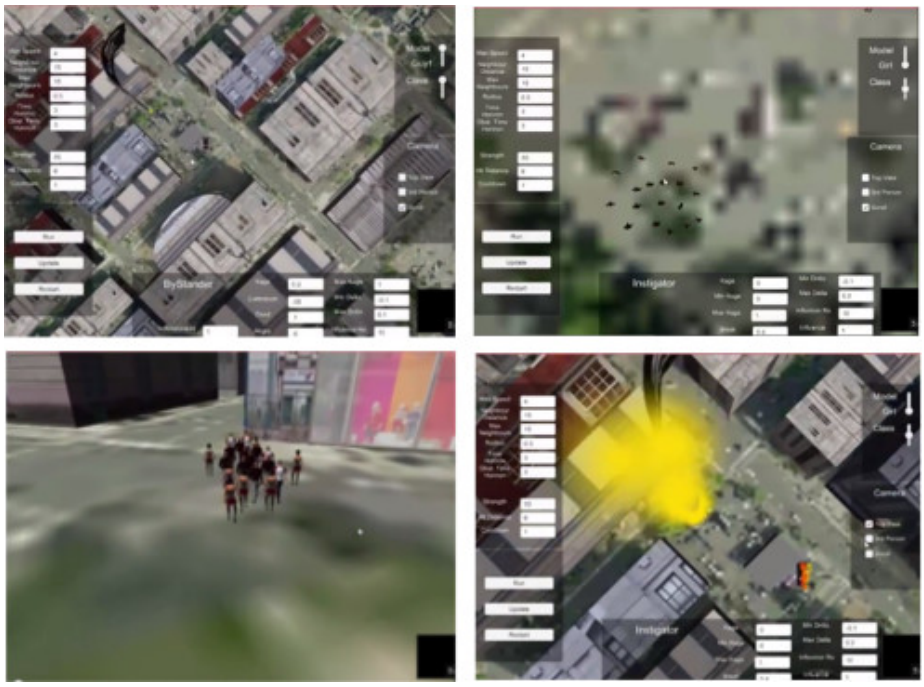

Figure 5: In the first image we can see where the agents intend to go, the second one shows the initial setup and location of the agents, third one shows a 3rd person view of the agents walking and in the last one we see the agents arrived to their location.

6.2 Breaking

A test on the object detection and breaking was made, the agents broke both cars and stores in their way to their destination.

Figure 6: In the first image the agents find the car, on the second one the car is broken by the agents.

6.3 Influence

Tests to ensure the appropriate influencing systems allowed us to observe the way the 3 kinds of agents interact with each other.

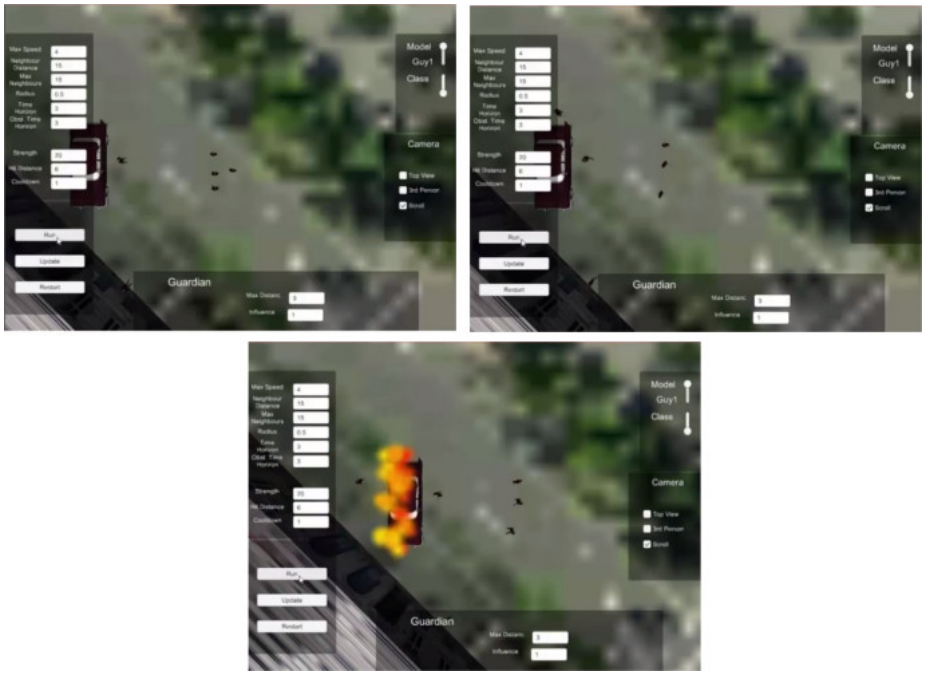

Figure 7: In the first image we see the initial setup with 3 by-standers, 1 instigator and 1 guardian. The second one shows the guardian calming down the by-standers and them staying where they started. The last ones shows that instigators are not influenced by guardians and breaks the car anyways.

The max ammount of agents that the simulation have been tested with was 100 in a Intel Core i7 with a 6Gb ram, however, the model should be able to handle more. Even a bigger number of agents can be used if the simulation realism is dimished.

7 Conclusion

The development of this model allowed us to see how individuals interact with each other in a crowd, and explore some of the variables that affect them. Such tool can be used to learn about crowd behaviour.

The emergent phenomena common in agent-based models leaded us to improve the rule specification on our model; a precedence rule was necessary between the following and breaking behaviours that a by-stander can perform.

A more realistic crowd movement method was necessary in order to create appealing behaviour, simple shortest path-finding algorithms don't fit in human behaviour without some modifications. There are many different parameters that can change the way a crowd moves, such as: personal space, max speed, need for collision-free movement, etc.

Crowds modelled following the ESIM model should be modelled by a heterogeneous set of agents with different behaviours, only one kind of agent won't be enough to reflect the complex behaviour of individuals in a crowd.

8 Improvement

In order to make this project better, the most important change is to research and implement police crowd-control tactics in order to be able to train police officers in the tactical planing of those.

Also, better human crowd movement could be achieved by enhancing the current movement component with position prediction approaches such as the one described in (Golas et al., 2013).

References

D. Balietti, S. Helbing. How to Do Agent-Based Simulations in the Future: From Modeling Social Mechanisms to Emergent Phenomena and Interactive Systems Design. Understanding Complex Systems. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. ISBN 978-3-642-24003-4. doi: 10.1007/978-3-642-24004-1. URL http://link.springer.com/10.1007/978-3-642-24004-1.

C.-s. Chiang, C. Hoffmann, S. Mittal, and W. Lafayette. Emergent Crowd Behavior. 6(6):865–875, 2009. doi: 10.3722/cadaps.2009.865-875.

A. Golas, R. Narain, S. Curtis, and M. C. Lin. Hybrid Long-Range Collision Avoidance for Crowd Simulation. *IEEE transactions on visualization and computer graphics*, 20(7):1022–1034, Sept. 2013. ISSN 1941-0506. doi: 1CD38DB9-D5F7-496A-932E-1A43A0B9F47F. URL http://www.ncbi.nlm.nih.gov/pubmed/24080711.

J. Hoggett and C. Stott. Crowd psychology, public order police training and the policing of football crowds. *Policing: An International Journal of Police Strategies & Management*, 33(2):218–235, 2010. ISSN 1363-951X. doi: 10.1108/13639511011044858. URL http://www.emeraldinsight.com/10.1108/13639511011044858.

D. Mount. CMSC 425: Lecture 20 Artificial Intelligence for Games: Decision Making. Technical report, 2013.

C. W. Reynolds. Flocks, herds and schools: A distributed behavioral model. *SIGGRAPH Comput. Graph.*, 21(4):25–34, Aug. 1987. ISSN 0097-8930. doi: 10.1145/37402.37406. URL http://doi.acm.org/10.1145/37402.37406.

S. Thalmann, D. Raupp Musse. *Crowd Simulation*. 2007. ISBN 9781447144496. URL http://onlinelibrary.wiley.com/doi/10.1002/9780470050118.ecse676/full.

J. van den Berg, S. J. Guy, M. Lin, and D. Manocha. Reciprocal n-body Collision Avoidance. pages 1–16, 2011.

M. Xiong, M. Lees, W. Cai, S. Zhou, and M. Y. H. Low. Hybrid modelling of crowd simulation. *Procedia Computer Science*, 1(1):57–65, May 2010. ISSN 18770509. doi: 10.1016/j.procs.2010.04.008. URL http://linkinghub.elsevier.com/retrieve/pii/S1877050910000098.