The Modeling and Simulation Of Cheating In An Examination Environment

Principle Investigator: Greg Pennefather

27 February 2013

1 Statement of the problem

Cheating in an examination environment is a problem that has no one definite solution. Potential cheaters are continuously creating either new methods for cheating or variations on old methods. Examiners respond using anti-cheating measures of their own, but it is difficult to see the effect of these measures.

A common response to the threat of cheating is to increase the number of invigilators in an exam. However it is preferable to use as few invigilators as possible as staff members are often unwilling to invigilate. How many invigilators are optimal? How would one measure the effectiveness of different numbers of staff members?

The effectiveness of an invigilator is a factor in decreasing the amount of cheating. An invigilator present in a venue, while being a discouraging presence, is not sufficient without the invigilator actively attempting to detect cheating. What active measures should be used to discourage and detect cheating? What will the effect of these different measures be on the amount of cheating?

The ability for an invigilator to detect cheating is dependent on the attentiveness of the invigilator. How attentive does an invigilator need to be to effectively discourage and detect cheating? How can we ensure that an invigilator stays attentive for the duration of the exam?

Creating models for these problems is a difficult task, due to the complexity of human nature. This makes anti-cheating measures unreliable until proven effective by testing. However due to the nature of the problem it is difficult to properly test the measures. Students will not admit they were discouraged from cheating or cheated successfully, so how can it be known if the measures were successful?

Through computer simulations of examination environment we can observe the exam in its entirety. Metrics can be gathered about the number of students detected, discouraged from, or successfully cheating. The parameters of the environment can be adjusted to observe changes in these metrics, allowing for the discovery of the most cost effective means for prevention of cheating.

2 Aim

The aim of this project is first to model the examination environment. The second aim is to create a tool for the simulation of the model. The simulation will have adjustable parameters to allow for the representation of any examination environment, with a summative report of different metrics that will indicate the effectiveness of the anti-cheating measures.

The simulator should be capable of visually representing the simulation if requested. The visual representation should provide useful information which will allow the viewer to observe events of interest.

3 Background

Computer simulation allows for the modeling and testing of systems and procedures that would otherwise have been difficult to test physically. Examples of procedures that are difficult to test are evacuation and emergency procedures. [2]

Test drills are expensive and time consuming to perform. They require work to be stopped in order to perform, making it infeasible to run multiple tests. With only one test performed, it is not possible to iteratively grow the procedures. [2]

Drills do not properly simulate the environment that they intend to. No sense of panic is felt during an emergency drill [2]. Instead they are often met with feelings of frustration as workers are being distracted from their work. It is lacks the ability to properly simulate the threat. In the case of fire emergency drills, without the presence of a fire it is not possible to know which paths will or will not be viable evacuation exits [3]. It also lacks the ability to simulate injuries to workers, as doing so would be unethical [2].

4 History

During his work on the Manhattan project, physicist Stanislaw Ulam created an algorithm for deterministic simulations. His algorithm attempted to overcome the randomness of certain systems and produce accurate information about the likelihood of outcomes [4]. Due to the randomness of the simulation, his colleague Nicholas Metropolis named the simulation a Monte Carlo Simulation. This was because Ulam's initial simulation was based off the card game solitaire [4]. John von Neumann, another colleague, saw the value in the algorithm, and used it to model the behavior of neutrons in a nuclear reaction [4]. During a reaction, electrons are subject to large amounts of scatter events. These events make it hard to predict the trajectory of the electron, as many possibilities exist based on the collection of compounding events [1].

Von Neumann's solution to the complexity of this problem was to create a tree like structure, where each branch of the node had a certain probability of occurring. The simulation would run through the tree from an initial state to one of the outcome states, generating random numbers at each node and following the branch that fitted the numbers generated. By accumulating a large number of outcomes, the simulation is able to describe the likelihood of the outcomes. [4]

This was the first use of the Monte Carlo Simulation. However many complex systems require the a large number of different interactive processes to be run in parallel with each other. This is a non-trivial task. It requires the of modeling of the system with consideration being paid to the interaction. In larger systems, this can become a complex balancing act. [4]

5 Approach

The following details the approach to the modeling of the system and the creation of the simulator. It makes special mention of considerations about each of the key aspects of the system, and the requirements that the simulator must fill.

5.1 Method

The simulation will make use of the Monte Carlo Simulation algorithm. This will allow it to handle the uncertainty of human interaction, and produce cumulative data of the random outcomes of interaction.

5.2 Design

At time of writing, it is intended that the simulation be written in the C# programming language. This allows for use of libraries such as XNA for the visulisation of the simulation.

5.3 Students

The student is the first object in the system that must be modeled. A student should be capable of movement, should have a field of view and range of hearing, should have a basic emotional scale, and should be capable of making choices such as whether to cheat or not.

5.3.1 Characteristics:

- *Confidence:* A student should have a level of confidence in their ability to achieve a desired result an exam. That confidence should be able to fluctuate during the exam. Students with lower level of confidence are more likely to attempt to cheat.
- *Nervousness:* A student should be capable of being nervous. Nervousness should be effected by the actions of invigilators. A student who is more nervous is more likely to be discouraged from cheating.

5.3.2 Cheating techniques

Models should be created for multiple different methods of cheating. Some archetypes of these are:

- *Concealed notes*: A student should be able to conceal a note on him or her self that should not be detectable by an invigilator. However in order to view the note, the student must reveal it, making it detectable. This technique must be prepared for prior to the exam.
- *Communication methods*: Verbal or gesture based communication for transferring information from one student to another. This should be detectable through hearing and vision. It should be possible for an invigilator to become suspicious of a student. This is a technique that can be performed ad hoc, or can be prepared for.

- *Wandering eyes*: A student should be capable of applying his field of view to another student's paper in order to gather information. This should be detectable by vision. This technique can be preformed ad hoc.

Other possibilities for cheating exist, but have yet to be formally described. Attempts will be made to collect more techniques through anonymous surveys.

5.4 Invigilators

The invigilator object is the second object to be modeled in the system. An invigilator should have a field of view and range of hearing, should be capable of movement, should be capable of suspicion, should have measures of attentiveness and should have procedures to follow.

5.4.1 Characteristics:

- *Attentiveness*: This is a measure of how attentive a invigilator is. This number will influence the randomness of events, increasing the chance that a invigilator will detect cheating or follow procedures.
- *Suspicion*: A invigilator should be capable of being suspicious of individual students. This will influence the chance of an invigilator noticing gesture communication, and will increase the chance of detection for that individual student.

5.4.2 Procedures and Techniques

The invigilator should be capable of performing various techniques and procedures to detect and discourage cheating. These include:

- *Positional and patrol procedures:* It should be possible to create a procedure where invigilators patrol between certain points in the venue. It should also be possible to assign an invigilator a place to stand for a given duration. These two procedures can be chained together. The presence of the invigilator in this case aims to discourage cheating, and possibly detect it by coming upon an unsuspecting student.
- *Listening techniques:* An invigilator should be capable of listening to noises in the surrounding area. This will allow for the possibly detection of verbal cheating.
- *Observation techniques and procedures:* An invigilator should be capable of following procedures such as checking every deck for illegal items or accompanying a student to the bathroom. It should also be capable of adjusting its field of vision to scan its surrounding area for cheating.
- *Distraction techniques:* These are techniques an invigilator can perform to entertain his or her self. They include books, cell phones and daydreaming. These result in a loss of attentiveness, with the possibility of an increase upon their completion.

Other possibilities exist, but have yet to be formally described.

5.5 Environment

It should be possible to fully model and simulate an exam environment.

5.5.1 Exam Venue

- *Layout:* It should be possible to design a layout that matches a desired exam environment.
- *Physical objects:* It should be possible to create and place physical objects, such as desks and exam papers. Desks should be numbered, and students should be capable of being seated at a desk.
- *Bathrooms:* It should be possible to model the path from the exam venue to the bathrooms and procedures for signing in and out of the bathroom.

5.6 Simulator

The simulator should be runnable on a windows machine. It should be possible for the simulator to complete the required simulations in a reasonable amount of time.

5.6.1 Parameters

It should be possible to adjust various parameters about the simulation.

- *Number of examinations:* It should be possible to model multiple examinations of different lengths in a single venue.
- *Number of simulations:* It should be possible to set the number of simulations that should be run before generating summaries of the outcomes.

It should be possible to create exams for simulations with the following parameters:

- *Examination Length:* It should be possible to change the length of an examination.
- *Seating arrangements:* It should be possible to set up the seating arrangements for students.
- *Invigilators:* It should be possible to create and add invigilators to an exam. Invigilators can have different behaviors, techniques, procedures and characteristics from each other.
- *Students:* Using a given set of parameters, the simulator should generate a class of students. These students should be named and identifiable. They have predefined characteristics, behaviors and plans for cheating. It should be possible to edit individual students before the simulation is run.

5.6.2 Summaries

Two types of summaries should be generated. The first a summative report on the collection of outcomes, the second a summative report on each individual simulation.

The following should be displayed in the report:

- *Student and Invigilators:* A list of the students and invigilators present in the exam. It should be possible to expand each invigilator so that their characteristics can be viewed.
- *Discouraged student metrics*: Metrics detailing which students intended to cheat, but were discouraged from it.

- *Detected student metrics:* Metrics detailing which students were detected to be cheating.
- *Successful student metrics:* Metrics about the students who were successful in their attempts.
- *Method success rates:* Metrics about the effectiveness of methods created for the invigilators.
- *Cheating success rates:* Metrics about the success of each different type of cheating technique.
- *Other interesting metrics:* Such as the number of students who were not detected due to an inattentive invigilator.
- *List of simulations:* So that each individual simulation may be viewed.
- *Visulisation button:* Allowing the visulisation of an individual simulation.

5.7 Visualisation

The visulisation should take place in a two dimensional space. Students, invigilators and objects will be represented by different shapes allowing for easy identification. The following should also be features:

- *Events of Interest:* Indicators should be made when an event of interest is taking place, to draw the attention of the reviewer. These could include increases in suspicion, significant decreases in confidence, accessing of crib notes, communication attempts etc.
- *Colour coding of objects:* Colour coding should be used to differentiate the status of different students and invigilators. For example a student who is green has no interest in cheating, a student who is yellow is extremely nervous, a student who is orange has a lack of confidence, a student who is red is attempting to cheat.
- *Object selection:* It should be possible to select a single object, viewing the details about that object in an overlay. If this object is a student or invigilator it should also show that objects field of view and range of hearing.

Visualisation should focus on showing only the most necessary of information to prevent information overload. It should however allow the reviewer access to that information if they wish it.

6 Requirements

No special requirements are needed by this project.

A machine for running the simulations is needed, though it should be possible to run the simulation on any machine with a decent amount of processing power.

7 Progression Time-line

Weeks	Focus
2	Background reading
5	Modeling of environment, creation of Monte Carlo diagrams
7	Coding of simulator
3	Testing
2	Final write up

8 References

References

[1] C., J. D. An introduction to monte carlo simulations. *Scanning Microscopy International 5*, 2 (1991), 329–337.

[2] Gwynne, S., Galea, E., Owen, M., Lawrence, P., and Filippidis, L. A review of the methodologies used in the computer simulation of evacuation from the built environment. *Building and Environment 34*, 6 (1999), 741 – 749.

[3] Li, Y., Sun, X., Feng, X., Wang, C., and Li, J. Study on evacuation in subway transfer station fire by steps. *Procedia Engineering 45*, 0 (2012), 735 – 740.

[4] Stochastikon.com. Monte Carlo Simulation. Online. Available from: `http://132.187.98.10:8080/encyclopedia/en/monteCarlo.pdf`.