TJHSST Senior Research Project Proposal: TJHSST Hallway Simulation 2008-2009

Paul Woods

February 13, 2009

Abstract

The project centers primarily around the creation of an agent-based traffic simulation tailored specifically to the study of movement around Thomas Jefferson High School for Science and Technology during the school day. The project incoroporates ideas and research from the following areas:

- Traffic Modeling: The movement of people through TJHSST during the day has similarities to many traffic simulations already created
- Computer Simulation: The project focuses on creating a simulation of TJHSST
- Multi-agent Systems: Each person's movement is determined on an individual basis, making the school a system with multiple agents.

Keywords: agent-based modeling, traffic

1 Introduction

1.0.1 Problem Statement

With an increased influx of students each year, the hallways in TJHSST have changed. The size of the student body continues to increase, yet unfortunately, our school building is unable to do the same. Over the course of four years at TJHSST, hallway traffic has changed. Normally, this is not an issue. The change in traffic simply causes students to arrive to class maybe a minute later. However, the effect of an emergency or major event could have changed as well. While experiments have been done in the past, up to this point, no one has investigated student movement patterns specifically at TJHSST, and no one has done an investigation quite this way before.

1.0.2 Project Aim

The aim of this project is to make it easier to view and understand what goes on in the hallways. The project will try to determine what would happen in the event of an unexpected occurrence, and find possible simple ways to relieve problematic areas. The research will be done several ways. A program will be created that simulates current student movement accurately and is tested by comparing its results to real-life information.

Collecting real-life data, applying logic, and changing program conditions will provide an increased insight into traffic patterns and how and what they are affected by. By modifying program code, possible solutions can be determined and analyzed. The goal of the project is to create an accurate model of the school, test the model against real-life occurrences to ensure its accuracy, tweak program variables to solve real-life dilemmas, use logic-based analysis to ensure that the results and possible proposed solutions are plausible and realistic, and suggest real-life changes and improvements that could be used to help solve problems.

2 Background

Before creating the experiment, background research involving similar experiments in the past and experiments that can be applied to this particular experiment were viewed. The purpose of this is to see what has already been done, how other people solve similar problems, and how this experiment can be carried out effectively.

1. Applied Research

    - Continuum Crowds was created by Adrian Trueille, Seth Cooper, Zoran Popovic in conjunction with the University of Washington, Electronic Arts. The project is designed to determine how to realistically model large crowd movement without collision detection. Collisions and movement were calculated together–as opposed to separately–because crowds tend to already know which areas are likely to be congested and adjust before reaching the congestion.

      They created several simulations, including one with 24 people in a hallway, a 2000 person army retreating from a 8001 person army, and a 16 square city block. The researchers concluded that the crowds actually moved more smoothly and realistically when calculating the collisions and movement together, as opposed to separately. The researchers mentioned future areas for continued research which included tighter packed areas, areas where people do not have a common goal in movement, and what they described as "posse chasing," in which certain people are avoided and others are looked for.

    - Finding Multi-Constrained Feasible Paths By Using Depth-First Search was carried out by Zhenjiang L and J.J. Garcia-Luna-Aceves The focus of the research was developing a depth first search algorithm that operated with an unusual number of restrictions. The algorithm was specifically designed for developing routing systems, which requires the consideration of several constraints (such as bandwidth, reliability, end-to-end delay, jitter, and cost).

- Washington and Lee Universitys Rance D. Necaises research paper titled Interactice Graphics Using OpenGL and the Graphix Windowing Toolkit is another source. The goal of the project was to analyze the Graphics Wnidowing Toolkit, which combines OpenGL and GLUT into an easier interface in order to make graphics easier for create. The result is to determine how to make graphics easier for students who do not have so much time to spend on graphics. The analysis involved primarily testing the new software, and it worked well for what they wanted to accomplish: finding an easier to use alternative to OpenGL and GLUT.

- Performance OpenGL: Platform Independence Techniques by Tom True, Brad Grantham, Bob Kuehne, and Dave Shreiner is an additional source. The part of the paper that I focused on involves errors. The paper explained that functions like glTexImage2d() are particularly vulnerable to errors. The paper explains what the errors are and how they affect performance. These are functions that are used in the code and such analysis will prove helpful in debugging and making display functions more efficient.

2. Past Research

   - Past experiments specificic to TJHSST have also been done, such as Alex Kotkovas Traffic Dynamics in Scholastic Environments. The program works on finding a way to map traffic in an TJHSST-esque environment.

3 Development Sections

You should show that you clearly understand your task, have a logical time plan, say, by the research, design, programming, sub-testing and testing phases of your project.

3.1 Phase 1

The program was tested by implementing the proposed algorithm and recording the time required by the algorithm to find and select routing pathways. The results were that the algorithm can work very effectively, but works best with a smaller number of possible locations. The input the program will receive currently consists of students with a randomized schedule, and a hash containing a small-scale model room and hallway layout as shown by Figure 1. To test the program in its current state, I ran a simulation in which students would start in one area and be required to find their way to the designated path. This preliminary simulation assumed that each room was approximately three minutes apart and students would leave their classes at exactly 10:05 and would remain in their goal destination once they arrived. I then output student data and then combined it to get the data shown in Table 1.

The program is being made primarily in the C programming language, though it is possible other languages (such as C++ and Java) will be used later in the project to compensate for the weaknesses of the C programming language.

To test the program in its current state, I ran a simulation in which students would start positioned in a room and attempt to find their way to their destination: a different room. This preliminary simulation assumed the following:

1. Each room is located approximately three minutes apart from each other.

2. Students will attempt to take the quickest possible path to their destination.

3. Students will stay in their destination once they arrive.

4. Students move at the same speed, regardless of traffic conditions.

5. All students left their initial location at 10:05 and tried to make sure they arrived at their next location by 10:15.

The initial input into the program is as follows: Data containing the miniature school map, written directly into the program code* in the form of a hash. Randomly created student schedules

At this point, there was no point in inputting data by reading a file because the map size manageable. The time and processing used to open and read an input file did not make such a method useful.

3.2 Phase 2

Phase two of the program was tested and implemented in the C programming language. The program is tested by implementing the proposed algorithm, outputting program data to a text file as the program runs, reviewing the output, and comparing the output to logical results. In this version, recording the time was not necessary, because this new version of the program runs much more quickly and efficient than the old code.

This new program ads several new components and functions. OpenGL has been added as a critical component. This part of the code is currently in its developmental stages. Testing has been performed, and while the program is not yet at the stage where it can display all forms of student movement that are simulated, it does print the initial screen and window successfully, even though such screens are sparse on actual data.

While the program will receive information from alien sources in later stages of development, the program receives input from itself throughout the code execution. The current focus is not on efficiency or storage, but rather on functionality. In later stages, the entire school will be mapped out, not just a small previously input sample.

The input the program receives consists of several things. The program fills up data about several predetermined students. Such data consists of everything from their names and their schedules throughout the day. The program also contains information about hallways and rooms, which are now implemented as different entities. All of this is done in the C programming language with conjunction of OpenGL. A sample of the program output is listed in Appendix II.

The new project code is organized in a different way. Several layers of organization include hallways, rooms, and students. This new organization allows printing functions to be separated from other functions, though functions can still be called as a result of code written near the top of the program. This new organization is necessary, as the code has reached the point where it spans a length of over eight hundred lines. Confusion and coding inefficiency made this new organization necessary.

The new simulation works with regard to student data, hall data, room data, and other forms of crucial data. Most of this data is stored in hashes, and has been tested extensively several times, as shown by Code 2 located in the appendix. While these areas of the program work fine, there are still areas that are under construction. One such area is the graphics department.

The code currently has the necessary frameworks for graphics. Using OpenGL, the code generates the screen that is necessary to create further graphics. However, because the code is still under construction, critical graphical components that are necessary to discern and use the data in a visual sense are severely lacking. However, these problems do not impair the ability to analyze the data. Because data can be analyzed in text form, analyzing and testing the code is still easy, and such processes of testing have been and still are fully functioning. However, without visualization, presenting and explaining the code will be more difficult, which is why this area needs further improvement.

While the first phase code works well with regards to path determination in a very small and limited area, the second phase requires a broader and more powerful path determination algorithm. While such an algorithm is in development and improving, the algorithm falls short of providing the quick, accurate, and realistic student movement patterns the code is intended to render. Because of this, this algorithm is another area that will be worked on extensively.

4 Results and Discussion

The results of the experiment were very positive. These results are based off of the first phase of testing, as the second phase of code, which is intended to be more powerful, more efficient, and work over a much larger area and paths, is still in development. Every student was able to locate his next period and arrive before the start of the next class (10:05). However, by looking at the data, it was clear

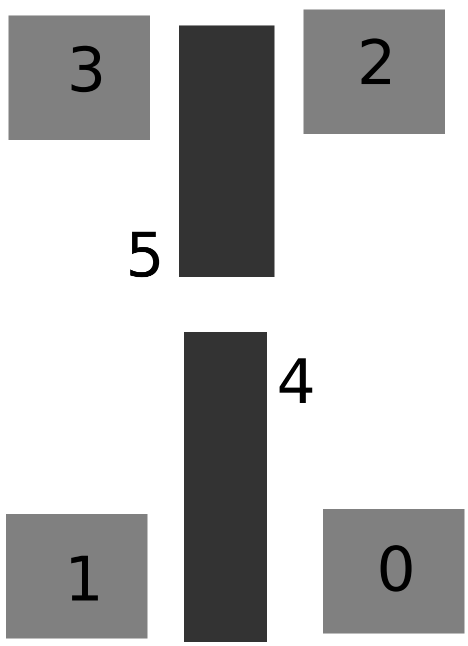

Figure 1: Model mini-building used in simulation phase.

that the hallways were not utilized efficiently. For instance, at 10:11, the Hall5 hallway had six times as much traffic as Hall4. Every student was not only able to locate his or her destination, but he or she was able to arrive before there before the start of the next class (10:15).

However, there were some problems. Looking at the data, it was clear that the hallways were not utilized efficiently. For instance, at 10:11, the traffic through Hall5 hallway was 600

This may have been a result of the fact that students did not take into account traffic when selecting movement. Also, the order of the schools could have been switched so to alleviate the traffic.

In the future, when the program is expanded to include the entire school, the program will likely be testing by collecting actual student movement data. There are several ways to do this, including counting the number of total students passing through certain hallways between classes, attaching tracking devices to willing participants, or asking a group of students to respond to survey questions about where they traveled and which areas they consider to be congested.

The results of the next testing was also promising. These results are based on the second phase of code indicated in the Appendices. The results of the testing were printed out in text form to a word file. These results indicated how effectively the program was able to collect and store data in the new program format.

The output and results were as expected. The program effectively stores all forms of

Hall Traffic Given Time:

Hall	10:08	10:11
4	60%	10%
5	40%	60%

Figure 2: Table showing results of initial experiment.

data, such as student name and hallway connections. These hashes and arrays are all necessary to implement movement algorithms, which will be refined.

5 Expected Results

For the third quarter, I plan on expanding the scope of the programs input to include the entire school and all students. Once this is in place, an accurate and full-fledged simulation can be developed and tested. The third phase of coding will likely involve improvements in the current code related to OpenGL, which will allow visualization of the movement and output. While visualization is not entirely necessary, it will be a huge step toward making the results of the program accessible.

Other improvements planned for the third quarter are also present. Such improvements include an improved text output functions in order to make data clear. This will be helpful not just building visualization but also testing the program. An improved algorithm for determining student movement is another planned improvement. This new algorithm will make the program run more quickly, efficiently, and accurately.

The third quarter will be a big step toward a complete simulation. These improvements will hopefully be the final steps of a fully functioning accurate model and simulation. While the current code works in several aspects, the third quarter expansions will make it applicable and testable in a real-life experiment. These changes are the next step.

References

[1] Alex Kotkova. "Traffic Dynamics in Scholastic Environments"

[2] Adrian Trueille, Seth Cooper, Zoran Popovi "Continuum Crowds"

[3] Zhenjiang L. J.J, Garcia-Luna-Aceves. "Finding Multi-Constrained Feasible Paths By Using Depth-First Search"

[4] Tom True, Brad Grantham, Bob Kuehne, and Dave Shreiner. "Performance OpenGL: Platform Independence Techniques"

[5] Rance D. Necaise. Computer Science Department Washington and Lee University. "Interactive Graphics Using OpenGL and the Graphix Windowing Toolkit"